

R.P. Abduraxmanov

# O'RNATILGAN TIZIMLAR



**O‘ZBEKISTON RESPUBLIKASI AXBOROT  
TEKNOLOGIYALARI VA KOMMUNIKASIYALARINI  
RIVOJLANTIRISH VAZIRLIGI  
MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT  
AXBOROT TEKNOLOGIYALARI UNIVERSITETI**

**R.P. Abduraxmanov**

# **O‘RNATILGAN TIZIMLAR**

**(O‘quv qo‘llanma)**

O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligi  
tomonidan tavsiya etilgan. Ta’lim yo‘nalishlari:  
5350100 - Telekommunikatsiya texnologiyalari  
5330500 - Kompyuter engineering

**TOSHKENT – 2021**

UO‘K:

KBK:

R. P. Abdurahmanov-O‘rnatilgan tizimlar. O‘quv qo‘llanma. –T.:  
«Nihol Print» OK, 2021, – 248 b.

ISBN 978–9943–

O‘quv qo‘llanmada o‘rnatilgan tizimlarning asosiy tushunchalari va asosiy xususiyatlari, asosiy ma‘lumotlar bazasi o‘rnatilgan tizimlarni qurish uchun ishlatiladigan apparat va dasturiy vositalar, ko‘milgan tizimlarni loyihalash masalalari ko‘rib chiqiladi, o‘rnatilgan tizimlarni ishlab chiquvchilarning asosiy muammolari, shuningdek, o‘rnatilgan tizimlarning yuqori darajadagi dizayni holati va istiqbollari tahlil qilinadi. O‘quv qo‘llanma o‘rnatilgan tizimlarning tizimli, oraliq va amaliy dasturlarini tashkil etish va o‘rnatilgan tizimlarni loyihalash, sinovdan o‘tkazish va disk raskadrovka qilishning instrumental vositalarini qamrab oladi.

O‘quv qo‘llanma talabalar tomonidan ta‘lim yo‘nalishlari uchun o‘quv jarayonida foydalanish uchun mo‘ljallangan:

5350100-telekommunikatsiya texnologiyalari  
(Telekommunikatsiyalar, telekommunikatsiya)

5330500 - kompyuter injiniringi (kompyuter injiniringi, AT-servis, Multimedia texnologiyalari).

UO‘K:

KBK:

### **Taqrizchilar:**

A.X.Abduqodirov – «Bitel Servis» MCHJ direktori;

X.E.Xujamatov – TATU, «MUT va T» kafedrası mudiri, PhD,  
dotsent.

O‘quv qo‘llanma Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Ilmiy-uslubiy kengashi tomonidan nashrga tavsiya etiladi (2021 yil 29.04.2021yil. №9(711) bayonnoma)

ISBN 978–9943–

© «Nihol Print» OK Nashriyoti, 2021.

# **Ustoz Yanovskiy Gennadiy Grigoryevich xotirasiga bag'ishlanadi**

## **Kirish**

Bugungi kunda O'rnatilgan tizimlar keng tarqalgan va ular maishiy texnika (televizorlar, muzlatgichlar, kir yuvish mashinalari, mikroto'lqinli pechlar), avtoulovlarda, zamonaviy aloqa, sanoat, harbiy va aerokosmik texnologiyalarda qo'llaniladi.

Yigirmanchi asrning saksoninchi yillarida O'rnatilgan tizimlar bozorida iqtisodiy investitsiyalarning keskin o'sishi tendentsiyasi kuzatildi. Bu ham sanoat, ham harbiy texnikalar uchun va doimiy ravishda tobora kengayib boradigan aqlli maishiy texnika bozoriga taalluqlidir. O'rnatilgan tizimlariga katta talab yarimo'tkazgichlarni ishlab chiqarish texnologiyasida jiddiy yutuqlarga olib keldi.

Umumiy maqsadli tizimlardan farqli o'laroq, har xil turdagi o'rnatilgan tizimlarini qurish qo'shimcha mas'uliyatni talab qiladi. O'rnatilgan tizim dasturini ishlab chiqishda ishonchlilik, xavfsizlik, real vaqt, xizmat ko'rsatish qobiliyati, va hokazolarni hisobga olish kerak. O'rnatilgan tizimning dasturiy ta'minotini apparat taminotidan ajratib ko'rib bo'lmaydi, shuni tushunish kerakki, u ishlab chiqilayotgan tizimning qismi emas, balki butun tizim.

Qo'llanma materiallari umumiy, kontseptual rejada keltirilgan, boshqacha qilib aytganda, o'rnatilgan tizimlarning arxitekturasi va dasturiy ta'minotni loyihalashning asosiy printsiplariga e'tibor qaratilgan.

# **1-BOB. O‘RNATILGAN TIZIMLAR, TA'RIFI, XUSUSIYATLARI VA TASNIFI.**

## **1.1 O‘rnatilgan tizimlarning asoslari**

Biz texnologiyalarning g‘ayrioddiy o‘zgarishi davridamiz. Tez orada elektron qurilmalarning ko‘plab turlari Internet orqali taqdim etiladi. Bular nafaqat kompyuterlari va mobil telefonlar, balki deyarli barcha funktsiyalar to‘plamining dasturiy ta‘minotiga ega bo‘lgan elektronika. Boshqacha aytganda, elektronika sinflarining aksariyati o‘rnatilgan kompyuter tizimlariga aylanadi va bu o‘tish muhim bo‘ladi.

Ushbu tendentsiya har xil turdagi mahsulotlar uchun doimiy ravishda o‘tib boradi. Ko‘plab yangi mahsulot ishlab chikish liniyalari Internetga ulanishni ta‘minlaydigan integral mikrosxemalar asosida quriladi. Quyidagi misollarni ko‘rib chiqing.

- Iste'molchi elektronikasi. Elektron iste'molchilarning eng raqobatbardosh va innovatsion toifalari bu yuqori aniqlikdagi televizorlar, raqamli kameralar, o‘yin pristavkalari va media-pleerlar bo‘lib, ular chip-tizimiga asoslangan va Internetga ulanishni ta‘minlovchi protsessorlar asosida ishlab chiqiladi.

- Telefoniya. Birinchi navbatda uyali aloqa tarmoqlariga ulanish uchun mo‘ljallangan mobil telefonlar. VoIP telefonlari nafaqat ovozli qo‘ng‘iroqlarni uzatish, balki tezkor xabar almashish, video chat va ijtimoiy tarmoqlarda tarmoq kontaktlari bilan birlashishni ta‘minlaydi. Zamonaviy telefoniya platformalarining aksariyati qo‘shimcha xizmat turlarini kengaytirish uchun mo‘ljallangan.

- Avtomobildagi axborot tizimlari. Avtomobildagi zamonaviy avtomatlashtirilgan axborot tizimlari jozibali dasturlarni taqdim etadi. Bular GPS-navigatsiya, orqani ko‘rish kameralari, ovozli aloqa va audio / video ijrosi.

- O‘rnatilgan tizimlarning dasturchilari uchun ushbu tendentsiyaning ahamiyati katta. Texnik nuqtai nazardan, O‘rnatilgan tizimlarni loyihalash usullari o‘zgarib bormoqda. Buning sababi turli xil multimediya interfeyslari orqali bir-biriga ulanishi kerak bo‘lgan ko‘plab tizimlar. O‘rnatilgan tizimlar dunyosi juda katta va tez o‘zgarib turadi.

O‘rnatilgan tizim bu shaxsiy kompyuterlar (kompyuterlar) yoki superkompyuterlar kabi boshqa kompyuter tizimlaridan farqli o‘laroq amaliy kompyuter tizimidir. Shu bilan birga, siz "O‘rnatilgan tizim" ta‘rifi moslashuvchan va aniqlash qiyinligini bilib olasiz, chunki u doimo texnologiya rivojlanishi va turli xil dasturiy va dasturiy komponentlarni

joriy qilish narxining keskin pasayishi bilan rivojlanib bormoqda. So'nggi yillarda ushbu mintaqa o'zining an'anaviy tasvirlarining aksariyatidan ustun keldi. O'rnatilgan tizimning eng keng tarqalgan tavsiflari quyidagilar:

O'rnatilgan tizimlarni shaxsiy kompyuterga (shaxsiy kompyuterga) qaraganda apparat va dasturiy funksiyalarda cheklangan. Bu O'rnatilgan tizimlarning kompyuter tizimlari oilasining ko'p qismi uchun to'g'ri keladi. Uskuna cheklashlari nuqtai nazaridan, bu ishlov berish samaradorligi, quvvat sarfi, xotira va apparat xususiyatlarining cheklanishini anglatishi mumkin. Dasturiy ta'minotda bu odatda shaxsiy kompyuterda cheklovlarni anglatadi - kamroq dasturlar, kichik dasturlar, operatsion tizim (OT) yoki cheklangan OS yoki mavhumlik darajasi uchun kamroq kod.

O'rnatilgan tizim maxsus funktsiyani bajarish uchun mo'ljallangan. O'rnatilgan qurilmalarning aksariyati birinchi navbatda bitta aniq funktsiyani bajarish uchun mo'ljallangan. Biroq, endi biz o'rnatilgan tizimlarga ega va turli funktsiyalarni bajarish uchun mo'ljallangan qurilmalarni ko'rmoqdamiz. Bundan tashqari, so'nggi raqamli televizorlar turli xil interfaol dasturlarni taqdim etadi. Ushbu ilovalar elektron pochta, veb-saytlarni ko'rish va o'yinlar kabi keng tarqalgan xususiyatlarni bajaradi.

O'rnatilgan tizim - bu boshqa kompyuter tizimlariga qaraganda yuqori sifat va ishonchlilik talablariga ega bo'lgan kompyuter tizimidir. O'rnatilgan qurilmalarning ayrim oilalari sifat va ishonchlilik talablari uchun juda yuqori darajaga ega.

O'rnatilgan tizimlar deb nomlangan ba'zi qurilmalar, masalan, Pocket PC yoki veb-planshetlar, aslida O'rnatilgan tizimlar emas.

O'rnatilgan tizimlar cheklangan kompyuterlardir. O'rnatilgan tizim (o'rnatilgan tizim, inglizcha o'rnatilgan tizim) - bu maxsus mikroprotsesor boshqarish, nazorat va monitoring tizimi. Bunday tizimni ishlab chiqish kontseptsiyasi shundan iboratki, bunday tizim to'g'ridan-to'g'ri uni boshqaradigan qurilmada o'rnatiladi.

Boshqarish tizimi yanada murakkab qurilma ichida joylashganligi sababli, uning rivojlanishida quyidagi omillar asosiy rol o'ynaydi:

- minimal energiya iste'moli (ehtimol avtonom elektr ta'minoti);
- minimal o'lchamlari va vazni;
- o'zining himoyasi minimal, bu strukturaning va qo'llaniladigan elementlarning mustahkamligi va qat'iyiligi bilan ta'minlanadi;

- issiqlikni o'chirish (sovutish) funksiyalari issiqlik sharoitlarining minimal talablarini ta'minlaydi. Agar issiqlik oqimining zichligi  $0,5 \text{ mVt} / \text{sm}^2$  dan oshmasa, qurilma yuzasining atrof-muhitga nisbatan haddan tashqari qizishi  $0,5 \text{ }^\circ \text{C}$  dan oshmaydi, bunday uskunalar issiqlik yuklanmagan deb hisoblanadi va maxsus sovutish sxemalarini talab qilmaydi.

Iloji bo'lsa, mikroprosessor va tizim mantig'i, shuningdek asosiy mikrosxemalar bitta chipda birlashtiriladi. Radiatsiya va elektromagnit qarshilikka, vakuumdagi ish qobiliyatiga, kafolatlangan ish vaqtiga, bozorda echimlarning mavjudligiga va hokazolarga oid maxsus harbiy-kosmik talablar. Oddiy O'rnatilgan tizimlarni qurish uchun ko'pincha bitta chipli kompyuterlar, ixtisoslashgan yoki universal mikroprosessorlar, FPGAlar kiradi. O'rnatilgan tizimlarning ba'zi turlarini qurish uchun ARM arxitekturasi mikroprosessorlari keng qo'llaniladi. Eskirgan qurilmalar va interfeyslar bilan to'g'ridan-to'g'ri foydalanish yoki ma'lum darajada muvofiqlikni ta'minlash ma'lum bir echimni ishlab chiqarishning arzonligi tufayli ARM mikroprosessorlari keng ishlatiladi. O'rnatilgan tizimlar cheklangan kompyuterlardir. Umumiy maqsadlar uchun mo'ljallangan kompyuterlar hisoblash tizimlarining eng keng tarqalgan sinfidir, shuning uchun ularning cheklovlari O'rnatilgan tizimlar bilan taqqoslanishi mumkin. Darhaqiqat, O'rnatilgan tizimlar loyihalash talablari va cheklovlari tufayli bir-biridan va umumiy maqsadli tizimlardan farq qiladi. Xususan, O'rnatilgan kompyuter tizimlari mo'ljallangan dasturlar, fizik o'lchamlari va shakillari, quvvat sarfi, tizim resurslari, funksiyalar va foydalanuvchilarning harakati to'g'risidagi taxminlarga nisbatan aniq cheklovlarga ega.

### **Foydalanish xususiyatlari va fizik o'lchamlari va shakillari**

Umumiy maqsadli mashinalardan farqli o'laroq, O'rnatilgan tizimlar odatda bitta maqsadli dastur yoki maqsadli dasturlar sinfiga mo'ljallangan. O'rnatilgan kompyuter tizimidan maqsadli foydalanish ko'plab loyihalash cheklovlari va murosaga olib keladi. Hajmi va fizik o'lchamlari ko'pincha tizimdan maqsadli foydalanishning tabiiy natijasidir. Masalan, bemorlar tomonidan doimiy ravishda foydalaniladigan tibbiy asboblarning fizik o'lchamlari va vazni cheklangan. Ushbu cheklovlar inson tanasining xususiyatlari va uning harakatlaridan kelib chiqadi. Uyali telefonlar cho'ntagining o'lchamiga (ba'zan katta) mo'ljallangan. Simsiz ulanish nuqtalari kabi O'rnatilgan

tarmoq qurilmalari o'zlarining o'lchamlari va shakllari bo'yicha cheklovlarga ega. Tizimning maqsadli ishlatilishi o'lchov jihatidan uning jismoniy cheklovlarini belgilaydi va boshqa ko'plab tarkibiy cheklovlar ulardan kelib chiqadi. Shunisi e'tiborga loyiqki, O'rnatilgan kompyuter tizimlari tizim darajasida yuqori darajadagi integratsiyani talab qiladi, ya'ni tizimning cheklangan hajmini, narxini va quvvat sarfini va boshqa ko'plab loyihalash cheklovlarini qondirish uchun tizimning funktsional imkoniyatlarini bir yoki bir nechta yarimo'tkazgichli qurilmalarda birlashtirishni.

### **Quvvat sarfi**

O'rnatilgan va universal tizimlarda energiya iste'moli ustuvor loyihalash chekloviga aylandi. Noutbuklar va serverlar kabi umumiy maqsadli tizimlarda quvvatni tarqatish chegaralari mos ravishda o'nlab va yuzlab vattlarga teng edi. O'rnatilgan tizimlar bilan taqqoslaganda, umumiy maqsadlar uchun mo'ljallangan platformalar nisbatan kam sonli quvvat diapazonini qo'llab-quvvatlaydi. Ushbu diapazon vaqt o'tishi bilan har bir alohida platformaning ishlash rejimlarini hisobga olgan holda rivojlandi. Ko'pgina hollarda O'rnatilgan tizimlar bir necha vatt dan mikrovatgacha bo'lgan loyihalash nuqtai nazaridan ancha past. Tarmoq marshrutizatorlari va telekommunikatsiya uskunalari kabi yuqori samarali ba'zi tizimlar quvvat va quvvat tarqalish chastotasi serverlar bilan bir xil yoki kamroq darajada bo'lishiga qaramay, O'rnatilgan tizimlarning aksariyati kichik xonalarda, ko'pincha batareya quvvatida va ventilyatordan foydalanmasdan yoki faol sovutishning boshqa turlarida ishlashga mo'ljallangan. Natijada, O'rnatilgan tizimlar agressiv dinamik quvvatni boshqarish mexanizmlari bilan ishlab chiqilgan. Umumiy maqsadlar uchun mo'ljallangan tizimlarga qaraganda, O'rnatilgan tizimlar energiya loyihalashi istiqbollarning yanada xilma-xil spektrlarini namoyish etadi. O'rnatilgan tizimlar turli xil dasturlarni qamrab oladi: yuqori samarali tizimlardan tortib batareyali tizimlarga qadar, ko'p yillar davomida bitta batareya bilan simsiz sensorli ilovalarda ishlash uchun mo'ljallangan. Shunday qilib, O'rnatilgan tizimlar quvvat rejaları va dasturlarga qarab farq qiladigan xususiyatlarga tayanadi.

### **Tizim manbalari va vazifalari**

Umumjahon va o'rnatilgan tizimlar miqdori emas, balki tizim resurslari va funktsiyalarining o'zgaruvchanligidan farq qiladi.



Oʻrnatilgan tizimlar odatda tizim resurslari va funktsiyalarining nisbatan statik va oldindan belgilangan toʻplamiga ega boʻlgan holda ishlab chiqiladi va joylashtiriladi. Bu haqiqat tizim dasturini va baʼzi tizim jarayonlarini, masalan, tizimni yuklash yoki muammolarni tashxislash jarayonini soddalashtiradi. Ushbu manbaning noaniqligi koʻplab Oʻrnatilgan tizimlarda mavjud emas, shuning uchun Oʻrnatilgan tizimni yuklash jarayonlari qisqaroq va sodda.

### **Foydalanuvchilar talablari**

Umumiy maqsadlar uchun moʻljallangan kompyuterlar foydalanuvchilarning xatti-harakati haqida saxiy taxminlarni keltirib chiqaradi. Aslida, elektronikaning bir nechta sinflari foydalanuvchi kompyuteriga qaraganda qulayroq boʻlgan foydalanuvchi profiliga ega. Mikrotoʻlqinli pechlar va audio / video pristavkalari tashqarisida bir nechta elektron mahsulotlar tizim muammolari va samaradorligini hal qilish uchun foydalanuvchiga juda katta yukni yuklaydi. Soʻnggi yillarda foydalanuvchi tajribasi sezilarli darajada yaxshilanganiga qaramay, odatdagi kompyuter foydalanuvchisi tizimning ishdan chiqishiga, dasturiy taʼminot paketlarini qayta oʻrnatishga yoki yangilashga, oʻrnatish va yangilanishlardan soʻng muvofiqlik muammolariga duch kelishga tayyor. Oʻrnatilgan tizimlarning aksariyati, oʻz navbatida, foydalanuvchilarning xatti-harakatlari haqidagi taxminlarning tor doirasiga ega. Koʻpgina Oʻrnatilgan tizimlar qurilmalar yoki infratuzilmaning qismlari boʻlib, ular ishonchli va aniq ishlaydi. Foydalanuvchilar media pleyerlar va telefonlar kabi qurilmalar hech qanday kechiktirmasdan javob berishlarini kutishadi. Infratuzilma va sanoat uskunalari koʻpincha xizmat koʻrsatish darajasi toʻgʻrisidagi kelishuvlarni ishonchli tarzda amalga oshirish uchun ishlab chiqilgan. Bu yanada qattiq isteʼmol talablari tizim dasturlariga va tizim funktsiyalarining umumiy tanloviga taʼsir qiladi.

### **Oʻrnatilgan tizimlarga oʻtish zarurati**

Albatta, bir necha yil oʻtgach, koʻplab foydalanuvchilar barcha elektronika va dasturiy taʼminot asosida ishlashini osonlikcha sezadilar. Ammo nima uchun bu shunday? Chiroq tugmachasi IP manziliga muhtojmi? Bepul dasturlarni yozish qiyin emasmi? Xoʻsh, nima uchun oldindan aytib boʻlmaydigan dasturiy xatolar bilan anʼanaviy elektronikaning kundalik ishonchliligini murakkablashtiradi? Ehtimol, eng ishonchli tushuntirish texnik nuqtai nazardan emas, balki elektron

mahsulotning rivojlanishiga ta'sir qiluvchi biznes dinamikasining ba'zi jihatlarini ko'rib chiqishdan kelib chiqadi. Haqiqiy tijorat raqobati mavjud bo'lgan paytdagi texnologik yutuqlar vaqt o'tishi bilan yangi mahsulotlar ushbu mahsulotlarning narxini ushlab turish yoki pasaytirishda o'sishni talab qiladi. Masalan, kompyuter biznesida o'rtacha narxlar vaqt o'tishi bilan samaradorlik oshishi bilan pasayishini ko'rish mumkin. Yangi mahsulotlar kamroq resursga ko'proq ish qilishlari kerak. chunki, agar siz yangi funktsiyalarni arzon narxlarda taklif qilmasangiz, sizning raqobatchilaringiz sizni ishingizdan chiqarib yuboradilar.

Agar yangi imkoniyatlarni ishlab chiqish va foydalibroq amalga oshirish mumkin bo'lsa, bu tendentsiya kamroq narsaga ko'proq vaqt sarflash bilan birga iqtisodiy jihatdan barqarordir. Yangi funktsiyalar ishlab chiqilishi va qo'shilishi kerak, ular yangi xususiyatdan kelib chiqadigan xarajatlarning o'sishiga qaraganda kamroq. Biznes muhitida rentabellik iqtisodiy barqarorlik bilan sinonimdir. Qanday qilib narxlarni pasaytirish sharoitida yangi xususiyatlarni ishlab chiqish mumkin? Oldingi rivojlanish natijalaridan kelajakdagi rivojlanish xarajatlarini kamaytirish yoki yo'q qilish uchun foydalanish. Va hech qanday texnologiya dasturiy ta'minot kabi qayta ishlatishni imkonini bermaydi. Agar oldingi rivojlanish investitsiyalari kelajakda, mahsulotning kelajak avlodlariga turtki bo'lishi mumkin bo'lsa, marjinal investitsiyalar funktsiyalarni rivojlantirishni moliyalashtirishi mumkin. Dastur bunga ideal sharoitda imkon beradi. Dasturiy ta'minot va kengaytiriladigan mahsulotga asoslangan mahsulot yadrosi yangi xususiyatlar va xizmatlarni iqtisodiy rivojlantirishni ta'minlab, kelajakdagi takliflarda o'tmishdagi o'zgarishlarni amortizatsiya qilishga imkon beradi. Dasturiy ta'minotga yo'naltirilgan xususiyatlar to'plamlari elektronikaning barcha turlariga apparat xususiyatlari to'plamlariga qaraganda ancha yuqori tejashni namoyish etish imkonini beradi. Yangi funktsiyalarni tarqatish haqida nima deyish mumkin? Bu sizning Internet aloqangizning muhim jihati. Internetga ulanish imkoniyati va dasturiy ta'minotga asoslangan funktsiyalar to'plami tufayli uskunani ishlatishdan oldin barcha bunday mexanizmlarni ishlab chiqishning hojati yo'q; bunday shartnomalar biznes oqimiga muvofiq tuzilishi va qabul qilinishi mumkin, agar platforma Internetga ulanishi bilan dasturiy ta'minot bilan belgilangan funktsiyalar to'plamiga ega bo'lsa. Albatta, Internetga ulanish nafaqat funktsiyalarni, balki qayta ishlash bilan bog'liq boshqa tadbirlarni ham amalga oshirishga imkon beradi. Kameralar ma'lumotlarni to'g'ridan-

to'g'ri onlayn saqlash hisoblariga yuklashlari mumkin. Video o'yinlar yangi tarkibga kirish yoki ko'p o'yinchi rejimlariga kirish kabi onlayn o'ynash xususiyatlarini ta'minlash uchun tuzilishi mumkin. Tibbiy asboblardan, sanoat uskunalari va bino avtomatlashtirish tizimlari nazorat buyruqlarini olishlari va vaqti-vaqti bilan diagnostika va diagnostika ma'lumotlarini olishlari mumkin. Kengroq ma'noda, Internetga ulanish sizga O'rnatilgan qurilmalarni boshqarishga imkon beradi. Ko'p hollarda, bu resurslarni boshqarish mexanizmini sifat jihatidan yaxshilashga imkon beradi. Masalan, yorug'lik, iqlim va xavfsizlik kabi tizimlarni masofadan turib boshqarish mumkin bo'lgan bino avtomatizatsiyasini ko'rib chiqiladi. Yoritgichlarni o'chirib qo'yish mumkin, va xonada jismoniy kirishga ehtiyoj sezmasdan, xonalar va binolarda harorat sozlanishi. Elektronikaning ko'plab turlarida keng qo'llanilganda, masofadan turib boshqarish va resurslarni nazorati, resurslarni boshqarishga katta ta'sir ko'rsatishi mumkin.

## **1.2. O'rnatilgan tizimlarning qo'llash ko'lami.**

Texnologiyalar bozorining deyarli har bir segmentidagi elektron qurilmalar O'rnatilgan tizimlar sifatida tasniflanadi (1.1-jadval). Muxtasar qilib aytganda, "kompyuter tizimlarining turlari" bundan mustasno, O'rnatilgan tizimlarning keng doirasi uchun yagona o'ziga xos xususiyat shundan iboratki, ularning barchasini aks ettiradigan yagona ta'rif mavjud emas. O'rnatilgan tizimlarni amalga oshirish doirasi haqiqatan ham juda katta. U uy taymeri darajasidagi eng oddiy qurilmalarni va ulkan hududlardagi muhim ob'ektlarni boshqaradigan eng murakkab taqsimlangan ierarxik tizimlarni o'z ichiga oladi:

- Telekommunikatsiya tizimlari, tarmoq uskunalari (komutatorlar, marshrutizatorlar, ADSL modemlar va boshqalar);
- Iste'molchi elektronikasi (Uyali telefonlar, PDA, O'yin pristavkalari, raqamli kameralar, elektr choynaklar, mikroto'lqinli pechlar, idish yuvish mashinalari va boshqalar);
- Zamonaviy tibbiy va sport anjomlari;
- Transportni avtomatlashtirish (avtomashinadan aviatsiya tizimlarigacha), avionika, shaharda harakatlanishni boshqarish tizimlari;
- Telemexanika tizimlari (tashqi yoritishni boshqarish tizimlari, elektr energiyasi va boshqa energiya manbalarini boshqarish va hisobga olish, energiya ob'ektlari monitoringi);

•Monitoring, navigatsiya, kuzatuv tizimlari, bort tizimlari uchun harbiy va kosmik ilovalar;

•Texnologiyalarga asoslangan "aqli uy" ("aqli bino") sensor tarmoqlari.

Ilova qilish uchun 1.1 va 1.2 jadvallarda O‘rnatilgan tizimlarga va ularga tegishli tavsiflarga ba’zi misollar keltirilgan.

Zamonaviy O‘rnatilgan tizimlar noyob cheklovlarga ega kompyuterlardir. O‘rnatilgan kompyuter tizimlari asosida elektron qurilmalar loyihalashi bo‘yicha davom etayotgan sanoat tendentsiyasi dasturiy ta‘minot to‘plamlari bilan Internetga ulangan qurilmalar sonining barqaror va sezilarli o‘shishiga olib keladi. Ushbu tizimlarni ishlab chiquvchilar va dasturchilar yangi imkoniyatlar va qiyinchiliklarga duch kelmoqdalar. O‘rnatilgan tizimlarni loyihalashda, ishlab chiqaruvchi tayyor va yangi yaratilgan echimlarning o‘zaro bog‘liqlik darajasidan qat’i nazar har doim ixtisoslashgan hisoblash tizimini yaratadi. Uning tahlil doirasi tizimni tashkil etishning barcha darajalarini o‘z ichiga oladi. Bu kuchli va qulay vositalar bilan tayyor operatsion tizim sharoitida dasturni yaratish bilan bog‘liq emas, balki juda qattiq cheklovlari sharoitida yangi ixtisoslashtirilgan tizimni yaratishga bog‘liq.

### 1.3. O‘rnatilgan tizimlarning ta’riflari va xususiyatlari

Hozirgi bosqichda boshqaruv tizimlariga tobora ortib borayotgan ehtiyoj kuchaymoqda, kompyuter texnologiyalarini ishlab chiquvchilar o‘zlarining vositalarini faol ravishda takomillashtirishlari kerak. Funktsional va konstruktiv maqsadlar bo‘yicha boshqarish yoki boshqarish ob’ekti bilan chambarchas bog‘liq bo‘lgan O‘rnatilgan tizimlar va tarmoqlar (O‘rnatilgan tizimlar va tarmoqlar) axborotni boshqarish tizimlarning katta qismini tashkil etadi. Bunday tizimlar o‘rnatilayotgan yoki O‘rnatilgan deb nomlanadi, biz "OT" qisqartirilgan belgisi bilan ushbu atamalarni sinonim sifatida ko‘rib chiqamiz.

1.1-jadval.

O‘rnatilgan tizimga misollar

O‘rnatilgan ilovalar	Misollar
Raqamli imzo	Ommaviy video displeylar, birinchi navbatda reklama va o‘yin-kulgi uchun

Raqamli saqlash	Korxonalar, kichik korxonalar va uylar uchun tarmoq qurilmasi, energiya va hududni samarali saqlash
Ishlab chiqarish jarayonlarini avtomatlashtirish	Aktivlarni nazorati va boshqarishi
Qimor	Video o'yinlar va boshqa o'yin platformalari
Qurilish avtomatizatsiyasi	Iqlim, yorug'lik, havo sifati va xavfsizlik kabi turar-joy va tijorat binolari uchun foydalanuvchi tomonidan boshqariladigan boshqaruv elementlari
O'yin-kulgi	GPS-navigatsiya, A / V ko'ngilochar, iqlim nazorati va tashqi aloqalar uchun birlashtirilgan platforma
IP kamera	Kengaytirilgan video xavfsizlik xususiyati
IP media telefoni	Konvergent xabarlar, kontaktlar, audio va video aloqalar, shuningdek veb / media-displeyli telefon apparati
Tibbiy	Tibbiy ko'rish va bemorlarning yozuvlarini boshqarish uchun platformalar
Harbiy / Aerokosmik	Qattiqlashtirilgan, standartlashtirilgan kompyuter, aloqa vositalari va saqlash elementlari birgalikda va keng miqyosli integratsiya tizimlari uchun qulaydir
Savdo nuqtasi	Ulangan kassa apparatlari, bankomatlar va tranzaksion terminallar
Printerlar	Qo'llab-quvvatlash ulanishi va faksni yaqinlashtirish, skanerlash va konversiya xususiyatlari

robototexnika	Sanoat va savdo robotlarini boshqarish va ishlatish
Tarmoq infratuzilmasi	Routerlar, xavfsizlik devori, bostirib kirishni aniqlash va profilaktika tizimlari
datchiklar	Atrof-muhit va sanoat jarayonlari sifatini nazorat qilish uchun sensorlar
Transport	Masalan, poezdlarda quvvatni boshqarish, kabinada iqlim nazorati, ma'lumot uzatish va media display
Simsiz infratuzilma	Kengaytiriladigan simsiz infratuzilma uchun modulli hisoblash, aloqa va saqlash elementlari

1.2-jadval.

*O'rnatilgan tizimlar va ularning bozorlariga misollar*

Bozor	O'rnatilgan qurilmalar
Avtomobi Isozlik	Avtomobilni yoqish tizimi
	Dvigatelni boshqarish
	Tormoz tizimi (ya'ni qulflashga qarshi tormoz tizimi)
Uy xo'jaligi elektronik asi	Raqamli va analog televizorlar
	Pristavkalar (DVD, videomagnitafon, kabel qutilari va boshqalar)
	Shaxsiy ma'lumotlar yordamchilari (PDA)
	Oshxona jihozlari (muzlatgichlar, tostlar, mikroto'lqinli pechlar)
	Avtomobillar
	O'yinchoqlar / o'yinlar
	Telefonlar / Uyali telefonlar / Peyjerlar
	Kameralar
Global joylashishni aniqlash tizimlari (GPS)	

Bozor	Oʻrnatilgan qurilmalar
Sanoat nazorati	Robototexnika va boshqaruv tizimlari (ishlab chiqarish)
Tibbiy	Tibbiy infuzion nasoslar
	Dializ mashinalari
	Tish protezlari
	Yurak monitorlari
Tarmoq	yoʻriqnomalar
	uyalar
	shlyuzlar
Avtomatlashtirish Ishlab chiqarish	Faks
	Suratni koʻchirish uskunasi
	Printerlar
	Monitorlar
	Skanerlar

Oʻrnatilgan tizimlar va tarmoqlar (yoki shunchaki Oʻrnatilgan tizimlar, OT) maishiy elektronika, sanoat avtomatikasi, transport, telekommunikatsiya tizimlari, tibbiy uskunalar, harbiy va aerokosmik muhandislik va boshqa sohalarida keng qoʻllaniladi. Oʻrnatilgan tizimlarlar koʻlami doimiy ravishda kengayib boradi va u yoki bu shaklda ushbu tizimlar yaqin orada inson faoliyatining barcha sohalariga kirib boradi.

Avtomatlashtirishning turli xil vazifalari va ularni hal qilish usullari oʻrnatilgan tizimlarning koʻp sonli imkoniyatlarini keltirib chiqaradi. Mavjud texnik cheklovlar va ajratilgan moliyaviy va vaqt byudjetlarini hisobga olgan holda, amalga oshirish variantini ishlab chiqaruvchi uchun murakkab ilmiy va texnik vazifaga aylanishi mumkin. Ishlab chiquvchi uchun loyihalash mavzusi, uni yaratishning mavjud usullari va vositalari toʻgʻrisida aniq tasavvurga ega boʻlish, yaqin prototiplarni olish yoki yaratish qobiliyati juda muhimdir.

Umuman olganda, oʻrnatilgan tizimlar kompyuter ishlab chiquvchilari uchun eng murakkab loyihalash ob'ektlaridan biridir. Oʻrnatilgan tizimlarni yaratishda hisobga olinishi kerak boʻlgan odatiy talablar va cheklovlarni hatto yuzaki tahlil ham buni tasdiqlaydi.

Real vaqt rejimidagi reaktiv tizimlarning xususiyatlari:

- atrof-muhit holatiga javob berish;
- Atrof-muhit bilan doimiy o‘zaro aloqada bo‘lish;
- Ideal maqsadda cheksiz maqsad algoritmini bajarish;
- tashqi vaqt cheklovlarini hisobga olish kerak (real vaqt).

O‘rnatilgan mobil tizimlarining xususiyati:

- murakkab funktsiyalar to‘plami;
- real vaqt rejimida ishlash;
- mahsulotning arzonligi;
- kam quvvat sarfi;
- qisqa vaqt ichida ko‘pincha kichik ishchi guruhlar tomonidan ishlab chiqilgan;

•“Cheksiz manbalar” yondashuvidan farqli o‘laroq “hisoblash resurslarini hisobga olgan holda” model doirasidagi dasturlash.

"Tizimdagi chip" (SOC) texnologiyasi bo‘yicha o‘rnatilgan xususiyatli tizimlar:

- ko‘pincha ishlab chiqaruvchilardan sotib olinadigan ("intellektual mulk") "tayyor bo‘lmagan komponentlar"ni yig‘ish;
- "qora qutilar" ierarxiyasi;
- loyihalash va tekshirish ko‘proq tizim darajadan amalga oshiriladi;
  - tarkibiy qismlarning o‘zaro ta‘siriga e‘tibor berish;
- dasturiy ta‘minotning ahamiyati.

Ko‘p jihatdan, o‘rnatilgan tizimlarni yaratishning murakkabligi bu kompyuter tizimlari klassi uchun adabiyotda aniq ta‘rifning yo‘qligi bilan tasdiqlanadi. Bu erda ba‘zi ta‘riflarga misollar keltirish mumkin.

"O‘rnatilgan tizim bu har qanday kompyuter, ko‘chma kompyuter (noutbuk) yoki katta universal kompyuter bo‘lmagan hisoblash tizimi."

"Dasturlashtiriladigan kompyuterni o‘z ichiga olgan, ammo umumiy maqsadlar uchun mo‘ljallangan kompyuter emas."

"Aniqlash qiyin. Stol kompyuteri bo‘lmagan deyarli har qanday kompyuter tizimi. "

" Qurilmaga kiritilgan ma'lumotlarga ishlov berish tizimi."

O‘rnatilgan tizimlarni amalga oshirish doirasi haqiqatan ham juda katta. U uy taymeri darajasidagi eng oddiy qurilmalarni va murakkab ob'ektlarni boshqaradigan eng murakkab taqsimlangan ierarxik tizimlarni o‘z ichiga oladi. O‘rnatilgan tizimlarni loyihalashda, ishlab chiqaruvchi tayyor va yangi yaratilgan echimlarning o‘zaro bog‘liqlik darajasidan qat‘i nazar har doim ixtisoslashgan hisoblash tizimini yaratishi muhimdir. U tizimni tashkil etishning barcha darajalarini tahlil qilishi kerak. Bu kuchli va qulay vositalar bilan tayyor operatsion



muhitda dasturni yaratish bilan bog‘liq emas, balki mutlaqo boshqa rejaning qattiq cheklovlari oldida yangi ixtisoslashtirilgan o‘rnatilgan tizimni yaratish bilan.

Albatta, o‘rnatilgan tizimlarni yaratish sohasidagi ba’zi vazifalarni, ayniqsa tayyor tizimni ishlab chiqish yoki o‘zgartirish haqida gap ketganda, stereotiplar yordamida hal qilinishi mumkin. Ammo bu holda ham, yuqori sifatli hisoblash platformasidan, kuchli ixtisoslashgan vositalardan foydalanish, mahsulotni sinchkovlik bilan tekshirish va sinovdan o‘tkazish talab etiladi. Ichki tizimlarni yaratish vazifalari, biron sababga ko‘ra yoki boshqa sabablarga ko‘ra shablon echimlari doirasiga to‘g‘ri kelmaydi, doimiy ravishda loyihalash usullari va vositalarini takomillashtirishni talab qiladi. O‘rnatilgan tizimlarni murakkablashtirish tendentsiyasi birinchi navbatda aksariyat tizimlar ko‘p protsessor taqsimlangan o‘rnatilgan tizimlar yoki boshqaruvchi tarmoqlar sifatida amalga oshirilishida namoyon bo‘ladi. Bu loyihalasherning vazifasini yanada murakkablashtiradi.

Zamonaviy taqsimlangan o‘rnatilgan tizimlarning asosiy xususiyatlari:

- Ko‘p o‘zaro ta’sirli tugunlar: ikkitadan ko‘p; Bugungi kunda minglab o‘zaro ta’sirli o‘rnatilgan kompyuterlarning tizimlari qiziqish uyg‘otmoqda.

- Odamlarning aralashuvisiz boshqaruv tizimlarining bir qismi sifatida ishlash. Bunday tizimlarda operator bo‘lishi mumkin, u ma’lumot olishi va qisman tizimning ishlashiga ta’sir qilish qobiliyatiga ega bo‘lishi mumkin, ammo boshqaruvning asosiy qismi taqsimlangan o‘rnatilgan tizimlar tomonidan amalga oshiriladi. Menejmentning funksional va fazoviy markazsizlashtirish darajasi juda katta farq qilishi mumkin.

- O‘rnatilgan kompyuterlarning hisoblash elementlari umumiy hisoblash va aloqa vazifalaridan tashqari vazifalarni bajaradilar.

- Taqsimlangan o‘rnatilgan tizimlar yirik texnik ob’ektlarning bir qismi sifatida ishlatiladi (masalan, muhandislik tuzilishi, energiya tuzilishi ob’ekti, transport tizimi, O‘rnatilgan tizimlar) yoki tabiiy ob’ektlar (masalan, atrof-muhit monitoringi komplekslari) bilan o‘zaro aloqada.

- Taqsimlangan o‘rnatilgan tizimlar energiya sarfi cheklangan tugunlar bilan tavsiflanishi mumkin, sobit yoki moslashuvchan topologiyaga ega, inson hayoti uchun zarur bo‘lgan funktsiyalarni

bajaradi, yuqori texnologiyalarni tatbiq etishni talab qiladi yoki prototipi yaratilishi mumkin.

Oʻrnatilgan tizimlarning yuqoridagi xususiyatlarini sarhisob qilganda quyidagilarni taʼkidlash kerak. Bular jismoniy dunyo obʼektlari bilan "chuqur birlashtirilgan" tizimlardir. Ularning elementlari deyarli har doim manbalarda cheklangan. Oʻrnatilgan tizimlar uzoq umr koʻrish tizimlari, koʻpincha avtonomdir. Oʻrnatilgan tizimlarning oʻlchamlari va murakkabligi koʻlami har xil. Ushbu tizimlar koʻpincha professional boʻlmagan foydalanuvchilar uchun moʻljallangan. Oʻrnatilgan tizimlar koʻpincha muhim funktsiyalarni bajaradilar. Mana bu tizimlar Oksford hisoblash lugʼatida qanday aniqlangan:

- Haqiqiy vaqt tizimi: misollar: jarayonlarni boshqarish, oʻrnatilgan hisoblash tizimlari, kassa savdo tizimlari va boshqalar.

- Oʻrnatilgan kompyuter tizimi (oʻrnatilgan tizimlar): kompyuterni element sifatida ishlatadigan, ammo asosiy vazifasi kompyuter funktsiyasi boʻlmagan har qanday tizim. Oʻrnatilgan tizimlarga misollar: DVD pleer, svetofor, bankomat, toʻxtash joyi va boshqalar.

- Axborot tizimlari va Real-Time tizimi oʻrtasidagi asosiy farq bu kirish-chiqish javob parametrining talqini.

Albatta, bugungi kunda "oʻrnatilgan tizimlar" tushunchasi evolyutsiyasini kompyuter texnologiyalarining rivojlanishi davomida kuzatish muhimdir:

1. Axborotni boshqarish tizimlari, 60-yillar.

2. Oʻrnatilgan tizimlar (ES), 70-yillarning oxirida.

3. Taqsimlangan oʻrnatilgan boshqarish tizimlari (tarmoq oʻrnatilgan oʻrnatilgan boshqarish tizimlari / tarqalgan axborot boshqaruv tizimlari) - 90-yillarning oxirlari.

4. Kiber-fizik tizimlar - (CPS), 2006 yildan beri.

Bugungi kunda oʻrnatilgan tizimlarning aksariyati aslida tarqatilgan. Oʻrnatilgan tizimlarning taqsimlanganligi yoki ushbu toifaga nazorat qiluvchi tarmoqlarning yoki tarqatilgan axborot va boshqaruv tizimlarining qoʻshilishi koʻplab mutaxassislar uchun haligacha munozarali boʻlib kelmoqda. Chet el ilmiy-texnik adabiyotlarida, aksincha, oʻrnatilgan tizimlar nafaqat kichik oʻlchamdagi yoki monoblokli, bitta modulli qurilmalar, balki keng koʻlamli obʼektlar bilan bevosita bogʻlangan, fazoviy va / yoki arxitektura taqsimlangan tizimlar ekanligi aniq koʻrsatilgan.

Oʻrnatilgan tizimlarning yuqoridagi talqinlarini umumlashtirib, bunday tizimlar uchun mumkin boʻlgan keng koʻlamli arxitektura echimlarini hisobga olgan holda, biz oʻrnatilgan tizimlarni boshqarish yoki boshqarish ob'ekti bilan bevosita taʼsir oʻtkazadigan maxsus hisoblash tizimlari (oʻrnatilgan tizimlar) deb belgilaymiz.

Bu sizga quyidagilarga imkon beradi:

- Asosiy umumiy xususiyatga koʻra hisoblash tizimlarining koʻplab sonlarini birlashtirish.

- «Hisoblash mohiyati» ga nisbatan ikkilamchi omillar taʼsiri muammosini bartaraf etish (oʻlchovlar, loyihalash, topologiya, aniq maqsad va boshqalar).

- Amalga oshirishda erkin tanlashni taʼminlash (ilgari mumkin boʻlgan koʻpgina arxitektura echimlari "bekor qilingan").

- Loyihalash texnikasini birlashtirish.

- Birinchi navbatda yuqori darajadagi loyihalashda yangi "faoliyat" larni ishlab chiqing (masalan, platformaga yoʻnaltirilgan loyihalash, aspekt loyihalashi va hk).

#### **1.4. Oʻrnatilgan tizimlarni tasniflash**

Bugungi kunda oʻrnatilgan tizimlar sinfining aniq taʼrifining yoʻqligi ularning loyihalashida ikkala oʻrnatilgan tizimlar va arxitektura paradigmalarni tasniflash, ulardagi hisoblash jarayonini tashkil etish variantlari, ularni loyihalash, dasturlash va sozlash texnologiyalari va boshqa bir qator muhim jihatlar bilan namoyon boʻladi. Oʻrnatilgan tizimlarni loyihalash, ushbu sohadagi mutaxassislar uchun yagona aloqa tilini yaratish tajribasini samarali toʻplash va uzatish "ishchi" tasnifisiz mumkin emas. Taniqli umumiy tasniflardan kompyuter texnologiyalarining hozirgi holati uchun oʻrnatilgan tizimlar Devid Pattersonning eng muvaffaqiyatli tasnifi boʻlib tuyuladi, unga koʻra kompyuter tizimlarining uch toifasi ajralib turadi. Tasniflash oʻrnatilgan tizimlarni ulardan foydalanish xarakteriga koʻra ajratadi. Ushbu tasniflash oʻrnatilgan tizimlar sinfining eng muhim xususiyatlarini taʼkidlaydi va, birinchi navbatda, barcha oʻrnatilgan tizimlar oʻrtasidagi murakkablikdagi oʻzgarishlarning eng keng doirasini taʼkidlaydi. Bu oʻrnatilgan tizimlarni tasniflash boʻyicha qiyin vaziyatni ochib beradi.

Ichki tizimlarning boshqaruv ob'ekti bilan integratsiya darajasini aks ettiruvchi tasniflash:

- axborotni boshqarish tizimlari;

- taqsimlangan axborotni boshqarish tizimlari;
- oʻrnatilgan tizimlar (ES);
- tarmoq ichiga oʻrnatilgan tizimlar (NES);
- kiberfizik tizimlar (CPS).

An'anaga koʻra, oʻrnatilgan tizimlar funktsional atribut bilan tasniflashni boshlaydi.

Siz oʻrnatilgan tizimlarni maqsadlariga koʻra quyidagilarga boʻlishingiz mumkin:

1. Avtomatik boshqarish tizimlari;
2. Sensorlardan ma'lumot toʻplash uchun oʻlchash tizimlari va tizimlari (oʻlchash, boshqarish bilan birga aniq oʻlchash funktsiyalari xarakterlidir);
3. Real vaqt rejimida soʻrovlarga javob beradigan axborot tizimlari (toʻlov tizimlari, chiptalarni bron qilish va boshqalar) Ular umumiy maqsaddagi axborot tizimlari bilan chegara pozitsiyasini egallaydi;
4. Raqamli ma'lumotlarni uzatish tizimlari (telekommunikatsiya tizimlari);
5. Real vaqt rejimidagi murakkab ierarxik tizimlar (murakkab, shu jumladan keng tarqalgan ob'ektlarni boshqarish va boshqarishni ta'minlaydi);
6. Harakatlanadigan ob'ektlarni boshqarish tizimlari;
7. Umumiy maqsadlar uchun oʻrnatilgan ichki quyi tizimlar;
8. Multimedia tizimlari.

Funktsional tasnif sizga har bir guruh oʻrnatilgan tizimlarning xarakteristikasini bilvosita Rasmlantirishga imkon beradi. Biroq, loyihalash shablonlari xususiyatlarining oʻzgarishi bu holda juda keng boʻlib, ishlab chiqaruvchi uchun bunday tasnifning ahamiyatini kamaytiradi.

Maqsadlar boʻyicha tasniflashdan tashqari, oʻrnatilgan belgilarni ajratish mumkin:

- tizimning murakkabligi (katta, oʻrta, kichik);
- tizim topologiyasi (kontsentrlangan, taqsimlangan);
- oʻrnatilgan tizimlarning asosi boʻlgan oʻrnatilgan tizim turi (bitta va koʻp protsessorli, homogen va heterogen, kuchli va erkin ulangan, OS dan foydalanadigan va foydalanmayotgan);
- real vaqtda amalga oshirishning oʻziga xos xususiyati (yumshoq va qattiq real vaqt);

• konstruksiya (monoblokli, modulli, oʻrnatilgan, kengaytiriladigan va kengaytirilmaydigan, xizmat koʻrsatiladigan va texnik xizmat koʻrsatilmagan);

• Oʻrnatilgan tizimlar uchun amalga oshiriladigan ishonchlilik, xavfsizlik, axborot xavfsizligi va boshqalar;

• boshqa funktsional va funktsional boʻlmagan tavsiflar.

Hayotiy tsikl fazalarida, ish vaqtidagi oʻrnatilgan tizimlarni ajratib koʻrsatish muhim (RunTime):

• tizimning instrumental imkoniyatlari;

• imkoniyatlar nuqtai nazaridan platforma turi (qisman yashirin imkoniyatlar; oʻrnatilgan tizimning yangi namunalari paydo boʻlganda uning imkoniyatlari aniqlanadigan platforma).

Rivojlanish bosqichida oʻrnatilgan tizimlar arxitekturiy darajada tasniflash uchun koʻplab mezonlarga ega. Quyidagi asosiy tarkibiy xususiyatlarni ajratib koʻrsatish mumkin:

Ierarxiya darajalari soni;

• tizimdagi kalkulyatorlar soni;

• tizimdagi kalkulyatorlarning heterojenlik darajasi;

• kompyuterdagi protsessorlar soni;

• kompyuterlardagi protsessorlar oʻrtasidagi aloqa turi;

• kompyuterdagi protsessorlarning bir xilligi darajasi;

• protsessor tengligi darajasi.

Integratsiya printsipiga koʻra, oʻrnatilgan tizimlarni toʻrt toifaga boʻlish mumkin: toʻliq tayyor, blokli, maxsus, yarim maxsus yoki aralash.

Toʻliq tayyor tizim bu maqsad vazifasini bajaradigan yoki maqsad vazifasini bajarish uchun mehnatkashlikni ozgina takomillashtirishni talab qiladigan tizimdir. Qoida tariqasida, tayyor tizimlarga asoslangan oʻrnatilgan tizimlar maksimal ortiqcha va shunga mos ravishda xarajatlarga ega.

Keyingi eng qiyin variant - bu tayyor bloklar tizimi. Bunday tizim maqsadli funktsiyasini bajarishi uchun sotib olingan dasturiy va apparat bloklarini birlashtirish yoki birlashtirish kerak. Tizimning zaxira qiymati birinchi holatga qaraganda ancha past. Rivojlanish narxi ham past.

Uchinchi va toʻrtinchi variantlarda maxsus tadqiqotlar, ishlab chiqarishni tayyorlash, yangi texnologiyalarni ishlab chiqish talab etiladi.

Ishlab chiquvchi uchun muhim tasnif mezonlari:

• ma'lumotlarni qayta ishlash nisbati - boshqarish;

- tizimning dasturlashtirilish darajasi;

- loyihalash usuli;

Platformani amalga oshirish usuli;

- loyihalashning turli jihatlarini bo'yicha loyihaga ta'sir kuchi.

O'rnatilgan tizimlar quyidagicha tasniflanishi mumkin:

- ko'lami / maqsadi bo'yicha;

- ma'lumot va boshqarish funksiyalarining har xil nisbati bo'yicha, ya'ni. asosan axborot tizimi (ma'lumotlarni yig'ish tizimi) yoki boshqarish (avtomatik boshqarish tizimi);

- apparat bloklarining fazoviy lokalizatsiyasi bo'yicha:

- a) fazoviy lokalizatsiya qilingan;

- b) fazoviy ravishda tarqalib ketgan.

- hisoblash (ma'lumotlarga ishlov berish) va aloqa (ma'lumotlarni kiritish-chiqarish funksiyasi) tarkibiy qismlarining har xil nisbatida;

- insonning ishtirok etish darajasiga ko'ra:

- a) avtomatik tizimlar - operator faqat tizim sozlamalari va ish rejimlarini dastlabki sozlash va operatsion sozlash funksiyalarini bajaradigan tizimlar. Ma'lumotlar to'plash, uzatish va boshqarish vazifalari, boshqaruv guruhlarining tezkor rivojlanishi inson aralashuvisiz amalga oshiriladi;

- b) avtomatlashtirilgan tizimlar - operator qisman yoki to'liq ravishda operatsion ma'lumotlarni qayta ishlashni va ijrochi qurilmalar uchun boshqarish buyruqlarini Rasmlantirishni ta'minlaydi (masalan, masofadan boshqarish).

- ma'lumotlarni qayta ishlash / hisoblash ishlarini tashkil etish to'g'risida (markazlashtirilgan / markazlashtirilmagan);

- tizimning fizik / mantiqiy modullari o'rtasida vazifalar va / yoki funksiyalar darajasida parallelizatsiya.

Tarqalgan ichki tizim - fazoviy ravishda tarqaladigan o'rnatilgan tizim. Bunday tizimlar tarkibiy qismlar o'rtasida zaif bog'lanish mavjudligi bilan tavsiflanadi. O'rnatilgan tizimlarni murakkablashtirish tendentsiyasi birinchi navbatda aksariyat tizimlarning ko'p protsessorli taqsimlangan hisoblash tizimlari yoki boshqaruvchi tarmoqlari Rasmida amalga oshirilishida namoyon bo'ladi. Bu loyihalashning vazifasini yanada murakkablashtiradi. Zamonaviy taqsimlangan o'rnatilgan tizimlarning asosiy xususiyatlarini ko'rib chiqing. O'zaro ta'sirli tugunlar juda ko'p (ikkita dan ko'p). Bugungi kunda minglab o'zaro ta'sirlangan o'rnatilgan kompyuterlardan iborat tizimlar qiziqish uyg'otmoqda. Inson aralashuvisiz boshqaruv tizimlarining bir qismi

sifatida ishlang. Bunday tizimlarda operator bo'lishi mumkin, u ma'lumot olishi va qisman tizimning ishlashiga ta'sir qilish qobiliyatiga ega bo'lishi mumkin, ammo boshqaruvning asosiy qismi taqsimlangan o'rnatilgan tizimlar tomonidan amalga oshiriladi.

Menejmentning funktsional va fazoviy markazsizlashtirish darajasi juda katta farq qilishi mumkin. O'rnatilgan hisoblash elementlari umumiy hisoblash va aloqa vazifalaridan tashqari vazifalarni bajaradilar. Tarqalgan o'rnatilgan tizimlar yirik texnik ob'ektlarning (masalan, O'rnatilgan tizimlar yoki bino) bir qismi sifatida ishlatiladi yoki tabiiy tabiat ob'ektlari (masalan, atrof-muhit monitoringi komplekslari) bilan o'zaro aloqada bo'ladi. Taqsimlangan o'rnatilgan tizimlar cheklangan energiya iste'moli bo'lgan tugunlar bilan tavsiflanishi mumkin, sobit yoki moslashuvchan topologiyaga ega, inson hayoti uchun zarur bo'lgan funktsiyalarni bajaradi, yuqori texnologiyalarni tatbiq etishni talab qiladi yoki prototip sifatida yaratilishi mumkin.

## **1.5. O'rnatilgan tizimlarda real vaqt mexanizmlari**

### **1.5.1. Real vaqt rejimidagi ta'riflar**

Real vaqt tizimi (RVT) - voqealarga kafolatlangan javob vaqti bo'lgan kompyuter tizimi. Real vaqt tizimi - chiqish effektini shakllantirish vaqti muhim bo'lgan har qanday kompyuter tizimi. RVTga misollar: jarayonlarni boshqarish, o'rnatilgan hisoblash tizimlari, kassa savdo tizimlari va boshqalar. O'rnatilgan tizimlarning xususiyatlari ishonchlilik, xavfsizlik va kafolatli javob vaqtini ta'minlash zarurligini o'z ichiga oladi. Kafolatlangan javob vaqtlariga rioya qilish odatda real vaqtda ish deb ataladi, o'rnatilgan tizim sensorlar orqali boshqarish ob'ekti haqida ma'lumot oladi. Olingan ma'lumotlarga javoban, o'rnatilgan tizim nazorat harakatini keltirib chiqaradi va uni ob'ekt bilan juftlashtirish uchun qurilma orqali boshqaruv ob'ektiga uzatadi. Boshqarish ob'ektidan ma'lumot olish va o'rnatilgan tizimdan boshqarish signalini berish o'rtasidagi vaqtga javob vaqti deyiladi. Real vaqtda ishlaydigan tizim tezkor bo'lishi shart emas. Real vaqt rejimida sensorlar tomonidan kafolatlangan vaqt oralig'ida olingan ma'lumotlarga javoban boshqarish signallari ta'minlanishi kerak. Reaktsiya vaqtiga rioya qilmaslik oqibatlarining ahamiyatiga ko'ra, real vaqt tizimlarining odatda ikkita guruhi farqlanadi:

- Real vaqt rejimida yumshoq tizim;

- Real vaqt rejimidagi qattiq tizim.

Real vaqt rejimida yumshoq dastur - kechikishlar o'rtacha qiymatlar bilan belgilanadi. Bu biznes jarayonlarini tashkil qilishda va savdoda sodir bo'ladi.

Real vaqt rejimidagi qattiq tizim bu real vaqt tizimidir, unda vaqt chegaralariga rioya qilmaslik tizimning ob'ektiv funktsiyasi uchun halokatli oqibatlarga olib keladi. Harbiy va kosmik maqsadlarda foydalanish paytida yuzaga keladi.

### **1.5.2. Real vaqt mexanizmlari**

#### **Taymer**

Taymer sizga ma'lum vaqt davomiyligini hisoblash imkonini beradi. Taymerning ishlash printsiplari asl qiymatni oldindan yozib olish qobiliyatiga ega bo'lgan ikkilik hisoblagichga asoslangan. Har bir soat tsiklidan keyin hisoblagich o'z qiymatiga birni qo'shadi yoki ayiradi. Nolga yetganda (ya'ni, u toshib ketganda), hisoblagich chiqishda faol darajani hosil qiladi. Odatda, taymerning chiqishi mikroprosessor yoki uzilishni boshqarish moslamasining so'rovini kiritishda ishga tushiriladi. Ko'pgina zamonaviy o'rnatilgan tizimlarda taymerlar vazifalarni almashtirishga asoslangan vaqtni taqsimlash tizimini tashkil qilish uchun asos sifatida ishlatiladi. Bunday holda, taymer kesish mexanizmi bilan birgalikda ishlatiladi.

#### **Qo'lga olish-taqqoslash qurilmasi**

Qo'lga olish-taqqoslash qurilmasi impulslarning davri va davomiyligini, navbati aylanishini o'lchash, shuningdek impulslarni hosil qilish uchun mo'ljallangan. Ushbu modul yordamida keng-impulslu (PWM) va fazali modulyatsiyani (FM) olish mumkin.

#### **Kuzatuvchi taymer**

Aytaylik, bizning tizimimizda ba'zi foydali vazifalarni bajaradigan ma'lum bir jarayon mavjud. Biz buni amaliy yoki kuzatiladigan jarayon deb ataymiz. Agar jarayon muhim maqsadli funktsiyani bajarsa, unda uning faoliyatini to'xtatish butun o'rnatilgan tizimning uzilishiga, ya'ni ishdan chiqishiga olib kelishi mumkin. Agar siz bunday vaziyatlarni aniqlash uchun maxsus choralarni ko'rmasangiz, o'rnatilgan tizim bir muncha vaqt to'g'ri ishlamasligi mumkin. Qo'riqchi taymeri odatda tizimlarni ishlamay qolgan holatdan chiqarish va normal holatga



keltirish uchun ishlatiladi. Shunday qilib, qo‘riqchi taymeri tizimni buzilishlardan himoya qilish mexanizmidir.

Qo‘riqchi taymerining ishini batafsilroq ko‘rib chiqamiz. Ishda uchta jarayon qatnashadi: kuzatilgan dastur jarayoni, qo‘riqchi taymeri va favqulodda vaziyatlardan tizimni himoya qilish mexanizmini amalga oshiruvchi xizmat ko‘rsatish jarayoni. Qo‘riqchi mexanizmining mohiyati kuzatilayotgan jarayon normal ishlashini aniqlash mumkin bo‘lgan mezonni tekshirishdir. Agar qo‘riqchi hamma kuzatilayotgan jarayon to‘g‘ri ishlayshini aniqlasa, unda hech narsa bo‘lmaydi. Agar qo‘riqchi kuzatilayotgan jarayonda biron bir narsa noto‘g‘ri ekanligini aniqlasa, ma‘lumot favqulodda vaziyatlarni hal qilish tizimiga uzatiladi, bu esa o‘z navbatida kuzatilayotgan jarayonning taqdirini hal qiladi. Eng sodda versiyada oddiy ayirish hisoblagichi qo‘riqchi taymeri vazifasini bajaradi. Ishga tushirish vaqtida hisoblagichga qiymat yoziladi. Agar ish paytida hisoblagichga vaqti-vaqti bilan yangi konstantra qo‘shilsa, hech narsa bo‘lmaydi. Agar dastur jarayonida konstanta yozish uchun vaqt bo‘lmasa va hisoblagich nolgacha hisoblashni boshlasa, apparatni qayta ishga tushirish signali hosil bo‘ladi va protsessor qayta ishga tushiriladi. Tabiiyki, qo‘riqchi taymerining sodda bajarilishi tizimning muvaffaqiyatsiz holatidan chiqishiga 100% kafolat emas.

Shuni ta‘kidlash kerakki, ko‘p mikrokontrolörlarda qo‘riqchi taymerining eng sodda sxemasi amalga oshiriladi. Uning normal ishlashi uchun qo‘shimcha, murakkab dasturiy ta‘minotga ehtiyoj bor. Oddiy qo‘riqchi taymerining sxemasida yana bir kamchilik - bu qurilmani qayta ishga tushirish signalini yaratish (RESET). Ko‘pgina hollarda, tizimning bir yoki bir nechta bir-biriga bog‘langan qismlarini qayta ishga tushirish yoki biron bir halokatli nosozliklar bo‘lsa to‘liq qayta ishga tushirish to‘g‘ri bo‘ladi.

### **Uzilishlar tizimi**

Klassik talqinda uzilish - joriy buyruqni yoki hodisalarni qayta ishlash uchun buyruqlar ketma-ketligini to‘xtatuvchi ishlov beruvchi tomonidan to‘xtatib qo‘yish, keyin esa to‘xtatilgan dasturga qaytish. Uzilishni ma‘lumotlar uzatish mexanizmlari (uzilishlar turi) va boshqarish (joriy jarayonni boshlash / to‘xtatish), shu jumladan protsesslararo aloqa mexanizmi sifatida aniqlash mumkin. Uzilishlar tizimi har qanday hisoblash tizimining ajralmas qismi bo‘lib, protsessorning e‘tiborni talab qiladigan bir qator holatlarga tezkor reaksiyasini ta‘minlash uchun ishlab chiqilgan, bu holatlar protsessor

ichida ham, undan tashqarida ham sodir bo'lishi mumkin. Uzilishlar nafaqat g'ayritabiiy vaziyatlarga reaksiya sifatida emas, balki zarur mexanizmlarning, masalan, virtual xotira, kirish / chiqish va boshqalar aksariyatini qo'llab-quvvatlaydigan tabiiy jarayon sifatida qaralishi kerak. Piter Nortonning iborasiga ko'ra: "Uzilishlar-bu kompyuterning harakatlantiruvchi kuchi."

Uzilishlar tizimi bu apparat va dasturiy ta'minot kompleksidir. Uzilishlar tizimning apparat vositalari odatda blok yoki uzilishlar controlleri deb ataladi. Kompyuterda bu PIC (Programmable Interrupt Controller), ya'ni alohida chip 8259A. Ba'zi hollarda uzilishlar controlleri mikroprotessor chipiga integratsiya kilinadi. Uzilishlarni boshqarish registrlar orqali boshqariladi. Har bir uzilishga tadbirning ahamiyatini raqamli ravishda belgilaydigan ustuvorlik berilishi mumkin. Maxsus uzilishlar ishlovchilari (interrupt handler) dasturi uzilish tizimlarining dasturiy vositalari hisoblanadi. Qoida tariqasida, ishlov beruvchilarning manzillari uzilishlar vektorlari deb nomlangan maxsus jadvalda joylashgan. Uzilish tizimining maqsadi dasturni amalga oshirish uchun protsessorni to'xtatib, tegishli vaziyatga xizmat qiladigan boshqa dasturga o'tish orqali ma'lum hodisalarga javob berish. Muayyan voqea (sabab) yuzaga kelgan paytda protsessorga kirib, maxsus operatsiyani - bir dasturni to'xtatib, protsessorni boshqa dasturga o'tkazadigan uzilish signali hosil bo'ladi.

Uzilishlar signalining manbasiga qarab, ular quyidagilarga bo'linadi:

- **asinxron yoki tashqi (apparat)** - tashqi manbalardan keladigan hodisalar (masalan, periferik qurilmalar) va har qanday o'zboshimchalik bilan sodir bo'lishi mumkin: taymer, tashqi interfeys va boshqalar;

- **ichki** - protessorning o'zida mashina kodini bajarishda ma'lum shartlarning buzilishi natijasida sodir bo'ladigan hodisalar: nolga bo'linish yoki ortiqcha to'ldirish, noto'g'ri manzillarga kirish yoki yaroqsiz operatsion kod. Ushbu turdagi uzilishlar istisnolar deb ham ataladi;

- **dastur uzilishlari (maxsus uzilishlar holati)** - dastur kodidagi maxsus ko'rsatmani bajarish bilan boshlangan. Dasturiy ta'minotning uzilishi odatda drayverlarning va operatsion tizimning dasturiy ta'minotiga kirish uchun ishlatiladi.

Apparat uzilishlari o'zboshimchalik bilan sodir bo'lishi mumkin va bajarilayotgan dasturga nisbatan asinxron bo'ladi. Apparat uzilishlaridan foydalangan holda protessor periferik qurilmalar bilan o'zaro aloqa

qiladi va turli xil apparat xatolari (masalan, xotira xatosi, shinada uzatish xatosi va boshqalar) yoki favqulodda elektr uzilishi haqida xabar beriladi. Ikkinchi holda, ba'zida bunday uzilishlar istisnolar deb ataladi. Apparat uzilishlariga javoban protsessor o'z manbasini aniqlab olishi, uzilgan dasturning minimal kontekstini saqlab qo'yishi va protsedura yoki vazifa sifatida bajarilishi mumkin bo'lgan maxsus dasturga - uzilishni qayta ishlov beruvchiga o'tishi kerak. Uzilishlar xizmati deb nomlangan Uzilishlarga ishlov beruvchisining harakati ma'lum bir manbaning uzilishiga to'g'ri javob berishdir (masalan, bosilgan kalitning xarakterini tamponga qo'yish, tizim soatini ko'paytirish va hk). Uzilish xizmatini tugatgandan so'ng, protsessor uzilgan dasturga qaytadi va u uzilishlar bo'lmagandek davom etishi kerak.

Protsessor tashqarisida yuzaga keladigan va uzilishlarga olib keladigan asosiy holatlar:

- boshqariladigan ob'ektdan so'rovlar (ular boshqarish tizimlari uchun xos), ya'ni tashqi qurilmalar (TQ) so'rovlari:

- a) almashish uchun tayyor bo'lgan TQ, dastur tomonidan boshqariladigan ma'lumotlarni uzatishni tashkil etish uchun protsessor reaksiyasini talab qiladi;

- b) TQ ma'lumot uzatish bo'yicha ishlarining yakunlanishi;

- c) TQ dagi maxsus (favqulodda) vaziyat.

- boshqa protsessorlardan uzilishlarni so'ralgan holda, ko'p protsessor tizimida sodir bo'ladigan hisoblash jarayonlarini sinxronlashtirishni ta'minlash.

Dasturiy uzilishlar, apparat uzilishlaridan farqli o'laroq, bajarilayotgan dasturga nisbatan sinxron ravishda paydo bo'ladi. Dasturni uzilishining sabablari dasturni bajarish paytida ro'y beradigan va dasturni normal davom ettirishga to'sqinlik qiladigan va maxsus ishlov qilishni talab qiladigan (vaziyatni to'ldirish, xotirani himoya qilishni buzish, operativ xotirada kerakli sahifaning yo'qligi va boshqalar), shuningdek INTn maxsus buyruqlar bo'lishi mumkin. - dasturiy uzilishlarining generatorlari bo'lgan uzilishlar. Ushbu buyruqlar odatda ma'lum bir OT funksiyalarini chaqirish uchun ishlatiladi.

Istisno vaziyatlarni qayta ishlash (exception handling) - dasturni amalga oshirishda yuzaga kelishi mumkin bo'lgan ish vaqti xatolariga va boshqa mumkin bo'lgan muammolarga (asosiy algoritm dasturi bilan keyingi ishlashning mumkin emasligiga olib keladigan istisnolar) javob berish uchun mo'ljallangan dasturlash tillari mexanizmi. Dasturni amalga oshirish jarayoni davomida ma'lumotlar holati, portlar yoki

kompyuter tizimi umuman asosiy algoritmgaga muvofiq qo'shimcha hisob-kitoblarni amalga oshira olmaydigan yoki ma'nosiz bo'lgan holatlar yuzaga kelishi mumkin. Bunday holatlarning klassik misollarlari:

- Butun sonni ajratish operatsiyasini bajarishda maxrajning nol qiymati. Amaliyot natijaga olib kelishi mumkin emas, shuning uchun keyingi hisob-kitoblar yoki bo'linish natijasidan foydalanishga urinish muammoning echimiga olib kelmaydi.

- Tashqi qurilmadan ma'lumotlarni o'qish paytida xatolik yuz berdi. Agar ma'lumotlar kiritib bo'lmaydigan bo'lsa, ular bilan boshqa biron bir rejalashtirilgan operatsiyalar ma'nosiz bo'ladi.

- Mavjud xotiraning yo'qolishi. Agar biron bir vaqtda tizim amaliy dastur uchun etarli tezkor xotira ajratolmasa, dastur normal ishlay olmaydi.

- Tizimning favqulodda o'chirish signalining paydo bo'lishi. Katta ehtimol bilan, dastur muammosini hal qilishning iloji bo'lmaydi, eng yaxshi holatda (agar biron bir quvvat zaxirasi mavjud bo'lsa), ilova dasturi ma'lumotlarni saqlash haqida g'amxo'rlik qilishi mumkin.

- Zudlik bilan o'qishni talab qiladigan ma'lumotlarning aloqa kanalining kirish qismida paydo bo'lishi. Hozirgi vaqtda dastur nima bilan shug'ullangan bo'lsa ham, olingan ma'lumotlarni yo'qotmaslik uchun ma'lumotlarni o'qishni davom ettirishi kerak.

Uzilishl tizimining funktsiyalari:

1. Ko'p manbalardan so'rovlarni qabul qiling va saqlang.
2. Qabul qilingan to'plamdan eng katta ustuvor so'rovni ajratish.
3. Markaziy protsessor tomonidan so'rovlarni qayta ishlash imkoniyatini tekshirish (niqoblangan so'rovlarni tekshirish yoki so'rovlarning ustuvorlik darajasini "uzilishlar chegarasi" bilan taqqoslash).
4. To'xtatilgan dasturning holatini (kontekstini) saqlash.
5. Uzilishlarga ishlov beruvchi vositasini chaqirish.
6. Haqiqatan ham ishlov berishda uzilishlar (uzilishlarga ishlov berish dasturining bajarilishi).
7. To'xtatilgan dasturning holatini (kontekstini) tiklash va uning amalga oshirilishini tiklash.

1-5-qadamlar uzilish so'rovi paydo bo'lganda kompyuterni apparat qurilmalari tomonidan avtomatik ravishda amalga oshiriladi. 7-bosqich ham apparat tomonidan amalga oshiriladi. Eng muhim ustuvorliklarni

ajratish uchun uzilish manbalarini so‘roq qilish ham apparat, ham dastur darajasida amalga oshirilishi mumkin.

Odatda birinchi o‘rnatilgan bitni qidirish uchun yagona registrga birlashtiriladi, so‘rovlar triggerlarini navbat bilan so‘rab oladigan maxsus dastur (**Dastuy poling**) tomonidan amalga oshiriladi.

**Apparat poling** ikkilik hisoblagichga asoslangan yoki odatda bir taktili elektron sxema yordamida amalga oshirilishi mumkin.

Protessorning kirish so‘rovlariga nisbati ikkita mexanizmdan irini qo‘llash orqali ifodalanishi mumkin:

- niqob mexanizmi;
- uzilishlar chegarasi.

### **Real vaqt soati**

Ko‘p kuchli mikrokontroller o‘rnatilgan real vaqt soat (RTC) blokiga ega. Soatlar sizga daqiqalar, soatlar, kunlar chegarasini kesib o‘tishni avtomatik ravishda kuzatishga, pog‘ona yillarini kuzatishga va avtomatik ravishda kunduzgi vaqtga o‘tishga imkon beradi. Qoida tariqasida, RTC bloke quvvat sarfini kamaytiradigan elementar asosda yaratiladi. RTCda vaqtni hisoblash uchun 32,768 kHz chastotali maxsus kvarts rezonatorlari qo‘llaniladi. Mikrokontroller odatda qo‘shimcha quvvatni (masalan, lityum batareyani) ulash imkoniyatiga ega.

Real vaqt soatlari aniq astronomik vaqtni ta‘minlashi uchun quyidagi shartlarga rioya qilish kerak:

1. Soat o‘zining avtonom elektr ta‘minotiga ega bo‘lishi kerak, shunda qisqa yoki uzoq elektr uzilishlari astronomik vaqtning qayta tiklanishiga olib kelmaydi.

2. Aniqlikni ta‘minlash uchun soatni sozlash kerak, chunki kvarts rezonatorlarni parametrlarida bir qator tarqalishlar bor.

3. Soatning aniqligi atrof-muhitga bog‘liq. Eng muhimi, urishning aniqligiga harorat ta‘sir qiladi, chunki harorat o‘zgaranda, kvarts rezonatorining chastotasi biroz o‘zgaradi. Aniqlikni ta‘minlash uchun har xil harorat oralig‘ida kalibrblash qiymatlarini berish kerak. Shuni esda tutish kerakki, real vaqt soati mutlaqo ishonchli qurilma emas. Muammolar soat bilan interfeys darajasida, kvarts rezonatorida yoki soatning o‘zida paydo bo‘lishi mumkin. Agar tizimning maqsadli funktsiyasi astronomik vaqtga bog‘liq bo‘lsa, ko‘pchilikni tashkil qilish uchun aniq vaqtning bir nechta manbalarini taqdim etish kerak, va real vaqt soatlaridan ma‘lumotlarni o‘qiyotganda, o‘qishlar oralig‘ining chegaralarini tekshiring va (masalan, taymer yordamida) soat ishlayaptimi yoki yo‘qmi tekshish kerak. Apparatdagi nosozliklar tufayli

real vaqtda soat nafaqat to'xtab qolishi mumkin, balki har xil muammolar tufayli soat chastotasi jiddiy ravishda o'zgarishi mumkin, bu soatning tezligi jiddiy pasayishiga yoki tezlashishiga olib keladi. Shuni ta'kidlash kerakki, real vaqtda soatning aniqligiga bosilgan elektron kartadagi o'tkazgichlarning topologiyasi ta'sir qiladi. Muammo shundaki, Real vaqt soatlari, odatda, kam quvvatli qurilma shaklida amalga oshiriladi. Kam quvvat iste'moli chip kontaktlarning orasidagi yuqori empedansning natijasidir va bunday sxemalar aralashuvlarga sezgir. Bundan tashqari, bosma montajning parazit qobiliyati soatning aniqligiga ta'sir qiladi. Parazitik quvvatning ta'sirini kamaytirish uchun elektron plataning kabelini maxsus tarzda tashkil qilish kerak.

### **Quvvatni boshqarish tizimi**

Quvvat tizimi har qanday elektron sxemaning asosidir. Afsuski, o'rnatilgan tizimlarda avtonom energiya manbalaridan (batareyalar, kimyoviy batareyalar va boshqalar), bort tarmog'idan (masalan, avtoulodan) foydalanganda, katta miqdordagi shovqin (masalan, ishlab chiqarishda) yuqori sifatli ikkinchi darajali elektr ta'minotiga erishish har doim ham mumkin emas. Quvvatni boshqarish tizimi kuchlanish barqaror bo'lmagan sharoitida mikrokontrollerni ishonchli ishlashini ta'minlash uchun mo'ljallangan. Qurilma yoqilgandan so'ng, chipda darhol o'tuvchilar jarayonlari boshlanadi. Kuchlanishning oshishi bir zumda yoki chiziqli ravishda yuz bermaydi, barqaror ta'minot kuchlanishini o'rnatish vaqti cxemaga bog'liq va odatda o'ndan yuzlab millisekundlarga teng. Hozirgi vaqtda, quvvatni boshqarish tizimi mikrokontrollerni ishga tushirishni kechiktiradi. Agar ishga tushirish kechiktirilmasa, barqaror bo'lmagan quvvat oladigan mikrokontrollerni nosozliklariga olib kelishi mumkin va ko'pincha qayta ishga tushiriladi. Ish paytida quvvatni boshqarish tizimi doimiy ravishda kuchlanish davridagi kuchlanish darajasini tekshiradi. Agar sath belgilangan qiymatdan chetga chiqsa, quvvatni boshqarish tizimi uzilishni keltirib chiqaradi. Uzilish ishlov beruvchisi o'rnatilgan tizimni, masalan, to'satdan elektr ta'minoti uzilib qolganda to'g'ri o'chirib qo'yishi mumkin.

Elektr uzilib qolganda, uzilishni qayta ishlaydigan uskunada nima qilish kerak? Quvvat manbaini qayta ishga tushirgandan so'ng, uzilib qolgan joydan ishlashni davom ettirish uchun siz dasturiy dasturining joriy holatini o'zgaruvchan bo'lmagan yoki oddiy xotirada saqlashga harakat qilishingiz mumkin. Tabiiyki, elektr ta'minotidagi uzilishlardan himoya qilishning bunday mexanizmini amalga oshirish uchun siz o'zingizning dasturingizni shunday amalga oshirishingiz kerakki, unda u

ish holatini aniq ko'rsatilishi kerak. Boshqacha qilib aytganda, bunday dasturlarni loyihalashda cheklangan avtomatlardan foydalanish juda foydali. Energiya kontrollerining vazifasi ta'minot zo'riqishining darajasini kuzatish va agar darajasi belgilangan belgilarga javob bermasa, raqamli signal berishdir. Ta'minot kuchlanishining o'zgarishiga nima olib kelishi mumkin? O'chirish sxemasi sof faol yuk emasligi sababli, o'tish davri deb ataladigan jarayonlar quvvat yoqilganda boshlanadi. Quvvat darajasi darhol o'rnatilmaydi, lekin bir necha o'nlab - yuzlab millisekundlarda va ba'zi hollarda tartibsiz o'zgarishi mumkin. Natijada, algoritmning boshlang'ich qismini ishlab chiqishda mikroprosessor va dasturlashtiriladigan mantiq o'nlab marta yoqilishi va o'chirilishi mumkin. O'chirish sxemani bunday xatti-harakati qurilmaning o'zgaruvchan bo'lmagan xotirasiga, shuningdek kirish-chiqish tizimiga salbiy ta'sir ko'rsatishi mumkin. Shunga o'xshash o'tishlar, odatda, qurilma o'chirilgan yoki ta'minot kuchlanishining to'satdan va qisqa muddatli yo'qolishi paytida (masalan, yomon aloqa tufayli) sodir bo'ladi. Ushbu hodisa bilan kurashishning eng oson usuli - bu vaqtinchalik o'tish davri mobaynida RESET signalini faol ushlab turadigan quvvat kontrolleridan foydalanish. Odatda, o'tish davrlaridan himoyalani uchun kuch kontrollerlari chiqish signalining sozlanishi yoki doimiy kechikishini tashkil qilish uchun Schmidt triggeri, hisoblagich, shuningdek hisoblash taymerini bilan ta'minlanadi.

Chipga bir nechta kuchlanishni etkazib berish bu juda jiddiy muammo. Qanday muammolar paydo bo'lishi mumkin? Birinchidan, kuchlanishni noto'g'ri etkazib berish bilan elektron komponentlarning ishonchligi pasayadi, bu esa o'z navbatida chipning umrini qisqartiradi. Buning sababi, kuchlanish yoqilgandan keyin bir muncha vaqt o'tgach, elektron komponentlar nomaqbul sharoitlarda ishlashga majbur (masalan, uchta o'rniga bitta quvvat manbai bilan). Ikkinchidan, sxemaga bir nechta kuchlanishni noto'g'ri etkazib berish, mikrosxemaning terminallarida ruxsat etilgan oqimlarning yoki kuchlanishning haddan tashqari ko'payishiga olib keladigan va halokatli oqibatlarga olib keladigan holatlar mavjud. Bir nechta kuchlanishni noto'g'ri etkazib berish tufayli tiristor effekti paydo bo'lishi mumkin, bu vaqtincha quvvatni ko'payishiga yoki qurilmaning ishdan chiqishiga olib keladi. Tizimli shinalarning to'qnashuvi ham mumkin, chunki tizim shinalarning bir qismi bir-birining oxiriga qarab harakatlanishi (ikkala shina bir vaqtning o'zida ishlash uchun ishlaydi). Bir nechta ta'minot kuchlanishni elektr kontroller bilan hal qilish mumkin.

## **Nazorat savollari**

1. O'rnatilgan tizim nima?
2. O'rnatilgan tizim qanday funksiyalarni ta'minlaydi?
3. Qanday dasturlar va omillari mavjud?
4. Qanday tizim funksiyalari va manbalari mavjud?
5. Foydalanuvchilar o'rnatilgan tizimlardan nimani kutmoqdalar?
6. Mur qonuni nima va u qayerda qo'llaniladi?
7. O'rnatilgan tizimlarda aloqa nima?
8. Maxfiylik qanday ta'minlanadi?
9. Tjoriylashtirishning sababi nimada?
10. O'rnatilgan tizimlarda energiya samaradorligi va xavfsizligini qanday ta'minlash kerak?
11. O'rnatilgan tizimlarning uchta an'anaviy yoki unchalik an'anaviy bo'lmagan ta'riflari qaysilar?
12. An'anaviy taxminlar qanday qo'llaniladi va keyinchalik murakkab o'rnatilgan loyihalarda qo'llanilmaydi?
13. O'rnatilgan tizimlarning arxitektura darajasida tasnifi.
14. O'rnatilgan tizimlar keng foydalaniladigan besh xil bozorni nomlang va tavsiflang.
15. Har bir bozorda to'rtta qurilmaga misollar keltiring.
16. Ko'plab o'rnatilgan loyihalar asoslangan to'rtta rivojlanish modelini nomlang va tavsiflang.
17. O'rnatilgan tizimlarni loyihalash va rivojlantirish uchun hayot tsiklining modeli nima?
18. O'rnatilgan tizimni ishlab chiqishda odatda duch keladigan beshta muammo nima?
19. O'rnatilgan tizimlarning funksional darajadagi tasnifi.
20. Zamonaviy taqsimlangan o'rnatilgan tizimlarning asosiy xususiyatlari
21. Real vaqt mexanizmlarini sanab bering.
21. Haqiqiy soat aniq astronomik vaqtni ajratib turishi uchun qanday shartlarni bajarish kerak?
22. Uzilish tizimining asosiy funksiyalarini sanab bering.
23. Signal manbasiga qarab uzilish signallari qanday guruhlarga bo'linadi?
24. Real vaqtning yumshoq tizimi va real vaqtning qattiq tizimi o'rtasidagi farq nima?
25. Qo'riqchi taymerining asosiy funksiyalarini tushuntiring.



## **2-BOB. O‘RNATILGAN TIZIMLARNI LOYIXALASHDA TIZIMLI YONDASHUV**

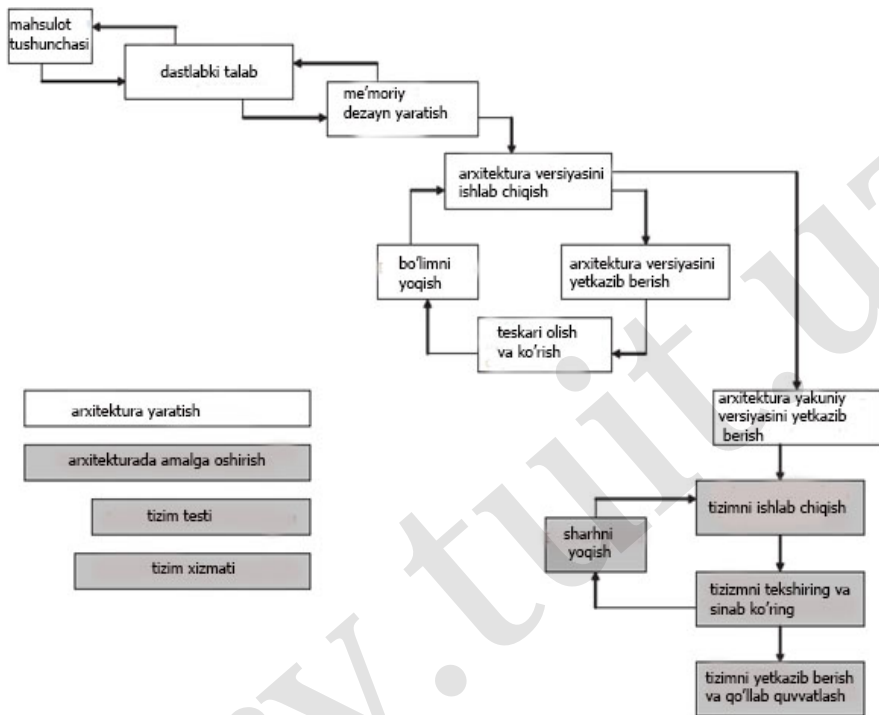
### **2.1. O‘rnatilgan tizilmarni loyihalashtirish**

Tizimli loyihalashtirish nuqtai nazaridan o‘rnatilgan tizimlarning arxitekturasini loyihalashda, o‘rnatilgan tizimlarning loyihalash tsiklini tavsiflash uchun bir nechta modellardan foydalanish mumkin. Ushbu modellarning aksariyati quyidagi rivojlanish modellarining bir yoki bir nechta kombinatsiyasiga asoslangan:

- Tizimni loyihalashdan oldin va uning davomida rejalashtirish yoki jarayonlarni amalga oshirmaydigan katta portlash modeli.
- Mahsulotga bo‘lgan ehtiyojni belgilaydigan, ammo ishlab chiqishdan oldin rasmiy jarayonlar mavjud emas kod va tuzatishlar modeli.
- Bir bosqich natijalari keyingi bosqichga o‘tadigan tizimni bosqichma-bosqich jarayoni amalga oshiriladigan sharshara modeli.
- Spiral modeli, unda tizimni bosqichma-bosqich ishlab chiqish jarayoni mavjud bo‘lib, turli bosqichlarda qayta aloqa jarayonga qaytishga imkon beradi.

Ushbu bo‘limda 2.1-rasmda keltirilgan, O‘rnatilgan tizimlarni loyihalash va rivojlantirish hayot tsiklining modeli deb nomlangan model taklif qilingan. Ushbu model mashhur sharshara va spiral modellarning kombinatsiyasiga asoslangan. Muvaffaqiyatsiz loyihalar yoki texnik va / yoki biznes talablariga javob beradigan ko‘plab qiyinchiliklarga duch kelgan loyihalar, shuningdek, barcha muvaffaqiyatli loyihalar muammoli loyihalarda etishmayotgan kamida bitta umumiy omilni o‘z ichiga oladi. Ushbu omil 2.1-rasmda ko‘rsatilgan jarayondir va shuning uchun ushbu model tizimning loyihalash jarayonini tushunishning muhim vositasi sifatida taqdim etilgan.

2.1-rasmda ko‘rsatilgandek, O‘rnatilgan tizimni loyihalash va ishlab chiqish jarayoni to‘rt bosqichga bo‘linadi: arxitekturani yaratish, arxitekturani amalga oshirish, tizimni sinash va tizimni qo‘llab-quvvatlash.



2.1-rasm. O‘rnatilgan tizmlarni ishlab chiqarish va loyihalashni hayot sikl modeli.

Ushbu matnda 1-bosqich olti bosqichdan iborat deb ta'riflangan: mustahkam texnik poydevor mavjudligi (1-bosqich), arxitektura biznes tsiklini tushunish (2-bosqich), arxitekturiy modellar va modellarni aniqlash (3-bosqich), arxitektura tuzilishini aniqlash (4-bosqich) va arxitektura hujjatlari (5-bosqich), shuningdek arxitekturani tahlil qilish (6-bosqich).

## 2.2. O‘rnatilgan tizim arxitekturasi ga kirish

O‘rnatilgan tizim arxitekturasi - bu O‘rnatilgan qurilmaning mavhumligi, ya'ni bu odatda dasturning manba kodi yoki apparat sxemasi kabi batafsil amalga oshiriladigan ma'lumotni ko'rsatmaydigan tizimning umumlashtirilishini anglatadi. Arxitektura darajasida O‘rnatilgan tizimning apparat va dasturiy ta'minotlari o‘zaro ta'sir

qiluvchi elementlarning biron bir tarkibi sifatida taqdim etiladi. Elementlar bu bajarilish tafsilotlari mavhumlashtirilib, faqat xatti-harakatlar va munosabatlar haqida ma'lumot qoldiradigan apparat va / yoki dasturiy ta'minotning namoyishlari. Arxitektura elementlari o'rnatilgan qurilmaga ichki qismi ravishda qo'shilishi yoki o'rnatilgan tizimdan tashqarida mavjud bo'lishi va ichki elementlar bilan o'zaro ta'sir qilishi mumkin. Qisqasi, o'rnatilgan arxitektura ichiga o'rnatilgan tizim elementlari, o'rnatilgan tizim bilan o'zaro aloqada bo'lgan elementlar, har bir alohida elementning xususiyatlari va elementlar orasidagi interfaol munosabatlar kiradi.

Arxitektura darajasidagi ma'lumotlar jismonan tuzilmalar sifatida taqdim etiladi. Tuzilish - bu arxitekturaning taqdim etilishi mumkin bo'lgan elementlari, xususiyatlari va munosabatlar haqida ma'lumotlarini o'z ichiga olgan mumkin bo'lgan tasvirlaridan biridir. Shuning uchun, tuzilish o'ziga xos muhit va ushbu elementlar to'plamini hisobga olgan holda ishlab chiqish va / yoki ish vaqtida ishlaydigan vaqtda tizimning apparat va dasturiy ta'minotining "surati" dir. Bitta "surat" uchun tizimning barcha murakkabliklarini qamrab olish juda qiyin bo'lgani uchun arxitektura odatda bir nechta tuzilmalardan iborat bo'ladi. Arxitekturadagi barcha tuzilmalar bir-biri bilan uzviy bog'liq bo'lib, bu qurilmaning o'rnatilgan arxitekturasi bo'lgan ushbu tuzilmalarning yig'indisidir. 2.1-jadvalda arxitektura o'rnatilishi mumkin bo'lgan eng keng tarqalgan tuzilmalarning qisqacha tavsifi keltirilgan va umuman, ma'lum bir struktura elementlari nima ekanligini va bu elementlarning o'zaro bog'liqligini ko'rsatadi. 2.1 jadval keyinchalik aniqlanishi va muhokama qilinishi kerak bo'lgan tushunchalarni taqdim etadi, shuningdek, o'rnatilgan tizimni namoyish etish uchun mavjud bo'lgan turli xil arxitektura inshootlarini namoyish etadi. Arxitekturalar va ularning tuzilmalari - ular bir-biri bilan qanday bog'langan, arxitekturani qanday yaratish kerak va hokazo - keyingi boblarda batafsil muhokama qilinadi.

Ushbu bo'limda o'rnatilgan tizimlarga arxitekturali tizimni yondoshuvi muhokama qilinadi, chunki u O'rnatilgan tizimlarning loyihalashini tushunishda yoki yangi tizimni loyihalashda yuzaga keladigan muammolarni hal qilishda ishlatilishi mumkin bo'lgan eng kuchli vositalardan biridir. Ushbu muammolarning eng keng tarqalgani quyidagilar:

- tizimni loyihalash usullarini aniqlash;
- narx bo'yicha xarajatlarni cheklash;

- ishonchlilik va xavfsizlik kabi tizim yaxlitligini aniqlash;
- kirish mumkin bo'lgan elementar funktsional imkoniyatlar doirasida ishlash (ya'ni hisoblash kuchi, xotira, batareya quvvati va boshqalar);
- bozor va bozorga chiqish qobiliyati;
- belgilangan talablar.

2.1-jadval.

Arxitektura inshootlariga misollar

Loyihalash turlari	Ta'rif
Modul	Modul elementlari (modul deb ataladigan o'rnatilgan qurilmaning ichida turli funktsional komponentlar (tizim to'g'ri ishlashi uchun zarur bo'lgan apparat va /yoki dasturiy ta'minot) sifati belgilanadi. Marketing va tijorat arxitektura sxemalari odatda modulli tuzilmalar ko'rinishida taqdim etiladi, chunki dasturiy ta'minot yoqo' shimcha qurilmalar odatda modullar Rasmio'tirilishi mumkin (ya'ni, operatsion tizim, protsess JVM va boshqalar).
Quyi tizim va komponent	Modullarning ishlatilishi bilan bir-biri bilan bog'liq bo'lgan (masalan, qaysi modul qaysi boshqa moduldan foydalanganligi) ish vaqti tizimini ifodalovchi modulli tuzilish turi.
Qatlamlar	Modullar qatlamlarga bo'linadigan (ya'ni ierarxik) ishlatiladigan tuzilish turi. Yuqori darajadagi modullar quyi darajadagi modullardan foydalanadi (talab qiladi).
Yadro	Tuzilma operatsion tizim yadrosining modullaridan (xizmatlaridan) foydalanadigan yoki yadro tomonidan boshqariladigan modullarni aks ettiradi.
Kanal arxitekturasi	Tuzilma modullarni ketma-ketlikda taqdim etadi, ulardan foydalanish orqali modullarning o'zgarishini namoyish
Virtual mashina	Tuzilma virtual mashina modullaridan foydalanadigan modullarni anglatadi.

Parchalanish	Ba'zi modullar aslida boshqa modullarning bo'linmalari (parchalangan birliklar) bo'lgan o'zaro aloqalar shunday modulli tuzilish turi. Resurslarni taqsimlash, loyihalarni boshqarish (rejalashtirish), ma'lumotlarni boshqarish (kapsulasyon, xususiylashtirish va boshqalar) ni aniqlash uchun keng qo'llaniladi.
O'quv mashg'uloti	Bu modullar sinflar deb ataladigan dasturiy ta'minotni ifodalaydigan modulli tuzilish turidir va munosabatlar ob'ektlarga yo'naltirilgan yondashuvga muvofiq belgilanadi, bunda sinflar boshqa sinflarga meros bo'lib o'tadi yoki ota-ona sinfining haqiqiy namunalari hisoblanadi
Komponent va ulagich	Ushbu tuzilmalar tarkibiy qismlardan iborat bo'lgan yoki ulagichlardan
Mijoz / server	Ishlash vaqtida tizimning tuzilishi, bu erda komponentlar mijozlar yoki serverlar (yoki ob'ektlar) va ulagichlar mijozlar va serverlar (yoki ob'ektlar) o'rtasida aloqa qilishda foydalaniladigan mexanizmlar (protokollar, xabarlar, paketlar va boshqalar).
Jarayon (aloqa jarayonlari deb ham ataladi)	Ushbu tuzilma operatsion tizimni o'z ichiga olgan tizimning SW tuzilishidir. Komponentlar bu jarayonlar va / yoki iplar va ularning ulagichlari protsesslararo aloqa mexanizmlari (umumiy ma'lumotlar, kanallar va boshqalar), rejalashtirish va ishlashni tahlil qilish uchun foydali.
Parallelizm va manba	Ushbu tuzilma tarkibiy qismlar parallel oqimlar orqali ulanadigan OSni o'z ichiga olgan tizimning suratidir. Aslida, ushbu tuzilma resurslarni boshqarish va umumiy manbalar bilan bog'liq muammolar mavjudligini aniqlash va parallel ravishda ishlashi mumkinligini aniqlash uchun ishlatiladi.
Uzilishlar	Uzilishlar tuzilishi tizimdagi uzilishlarni boshqarish mexanizmlarini aks ettiradi.
Rejalashtirish (EDF, ustuvorlik, dumaloq)	Tuzilma - bu OT rejalashtiruvchisining odilligini namoyish etuvchi oqimlarni rejalashtirish mexanizmi.

Xotira	Ushbu ish vaqtida xotirani ajratish va taqsimlash sxemalari (ulagich) bilan jihozlangan xotira va ma'lumotlar qismlaridan iborat - asosan tizim xotirasini boshqarish sxemasi
Axlatni olib tashlash	Ushbu tuzilma axlatni yo'q qilish sxemasini anglatadi
XotiraYni tarqatish	Ushbu tuzilma tizim xotirasini ajratish sxemasini (statik yoki dinamika, o'lcham va boshqalar) ifodalaydi.
Xavfsizlik va ishonchlilik	Bu tizimning tuzilishi bo'lib, unda ortiqcha komponentlar (HW va SW elementlari) va ularning o'zaro ta'sir mexanizmlari muammo yuzaga kelganda (uning turli xil muammolardan xalos bo'lish qobiliyati) tizimning ishonchliligi va xavfsizligini namoyish etadi.
Munosabatlar ni tarqatish	Turli muhitlardagi sw va hw elementlari va tashqi elementlar o'rtasidagi munosabatlarni ifodalovchi tizim.
Ish taqsimoti	Ushbu tizim modulning vazifalarini turli xil ishlab chiquvchilar va loyihalashlarning jamoalari o'rtasida taqsimlaydi. Loyihani boshqarishda keng qo'llaniladi.
Amalga oshirish	Bu SW tuzilishining fayl tizimida SW joylashgan joyni bildiradi
Joylashtirish	Ushbu tuzilma tizim bo'lib, bu strukturadagi elementlar HW va SW ni tashkil qiladi va SW qo'shimcha qurilmada ko'rsatiladigan elementlar orasidagi bog'liqlik.

Ushbu muammolarni loyihaning dastlabki bosqichlarida hal qilish uchun o'rnatilgan tizimlarning arxitekturasiidan foydalanish mumkin. Ichki amalga oshirish tafsilotlarini aniqlamasdan yoki bilmasdan, o'rnatilgan qurilmaning arxitekturasi tahlil qilinadigan va loyiha infratuzilmasi, loyihalashning mumkin bo'lgan variantlari va cheklovlarini belgilaydigan daraja rejasi sifatida ishlatilishi mumkin bo'lgan birinchi vosita bo'lishi mumkin.

Me'moriy yondashuvni bunchalik kuchli qiladigan narsa nima? Bu uning texnik tajribaga ega yoki tajribasiz har xil odamlarga loyihani norasmiy va tezkor ravishda taqdim etish qobiliyatidir, hatto loyihani

rejalashtirish yoki qurilmaning haqiqiy loyihalashi uchun asos bo'lib xizmat qiladi. Tizim talablarini aniq belgilab qo'yganligi sababli, arxitektura har xil sharoitlarda qurilmaning sifatini va uning ishlashini tahlil qilish va sinovdan o'tkazish uchun ishonchli asos bo'lib xizmat qilishi mumkin. Bundan tashqari, to'g'ri tushunish va xarajatlarni kamaytirish uchun arxitekturani to'g'ri ishlatish va samarali foydalanish mumkin. Bunga turli xil elementlarni amalga oshirish bilan bog'liq bo'lgan xatarlarni namoyish etish va shu bilan ushbu xatarlarni kamaytirish orqali erishiladi. Bundan tashqari, kelajakdagi mahsulotlarni o'xshash xususiyatlarga ega loyihalashda uchun turli xil arxitektura tuzilmalaridan foydalanish mumkin, bu esa dizayn bilimlarini qayta ishlatishga imkon beradi va natijada kelajakda loyihalash va ishlab chiqarish xarajatlari past bo'ladi. O'rnatilgan tizim arxitekturasini aniqlash va tushunish yaxshi tizim loyihalashining muhim tarkibiy qismidir. Buning sababi yuqorida sanab o'tilgan afzalliklarga qo'shimcha ravishda:

1. Har bir o'rnatilgan tizim hujjatlashtirilgan yoki yozilmaganligidan qat'iy nazar arxitekturaga ega, chunki har bir o'rnatilgan tizim o'zaro ta'sir qiluvchi elementlardan (apparat yoki dasturiy ta'minot bo'lsin) iborat. Arxitektura, ta'rifi ko'ra, ushbu elementlarning tasvirlari va ularning o'zaro aloqalari to'plamidir.

2. O'rnatilgan arxitektura tizimning vakili bo'lgan turli xil tasvirlarni o'z ichiga olganligi sababli, bu barcha asosiy elementlarni, nima uchun har bir tarkibiy qism mavjudligini va nima uchun elementlar o'zlarini qanday tutishini tushunish uchun foydali vositadir. Qurilmadagi har bir element qandaydir tarzda boshqa element bilan o'zaro ta'sir qiladi. Elementning funktsionalligi va ishlashining "nima uchun" ekanligini tushunmasdan, tizim haqiqiy dunyoda turli sharoitlarda o'zini qanday tutishini aniqlash qiyin bo'lar edi.

Arxitektura qandaydir tarzda loyihaning muhim tarkibiy qismlarini va ularning bir-biri bilan o'zaro bog'liqligini etkazsa ham, u loyiha ishtirokchilariga qurilmaning talablariga javob bera oladimi yoki qanday qilib bunday tizim muvaffaqiyatli qurilishi haqida muhim ma'lumotlarni beradi.

### **2.3. O'rnatilgan tizim modeli**

O'rnatilgan tizimning texnik tushunchalari va asoslari bilan tanishish uchun turli xil arxitektura tuzilishlaridan foydalaniladi. Yuqori

darajada, oʻrnatilgan tizim tarkibidagi asosiy elementlarni ifodalash uchun ishlatiladigan asosiy arxitektura vositasi 2.2. rasmda taqdim etilgan.



### Rasm.2.2. Oʻrnatilgan tizim modeli

Oʻrnatilgan tizim modeli barcha oʻrnatilgan tizimlarning eng yuqori darajada bitta oʻxshashlikka ega ekanligini koʻrsatadi; yaʼni ularning barchasi kamida bitta darajaga (apparat) yoki barcha tarkibiy qismlarga kiradigan barcha darajalarga (apparat, tizim dasturlari va amaliy dasturiy taʼminot) ega. Apparat darajasi oʻrnatilgan chipda joylashgan barcha asosiy jismoniy qismlarni oʻz ichiga oladi. Tizim va amaliy dasturiy taʼminot darajalarida oʻrnatilgan tizim tomonidan joylashtirilgan va ishlov berilgan barcha dasturlar mavjud.

Ushbu etalon modeli modulli arxitektura tuzilishi oʻrnatilgan tizimlar arxitekturasining koʻp darajali (modulli) ifodasidir. 2.2-jadvalda koʻrsatilgan qurilmalar orasidagi farqlardan qatʼi nazar, ushbu qurilmalar tarkibiy qismlarini qatlam sifatida vizual ravishda va guruhlash orqali ushbu tizimlarning barcha arxitekturasini tushunishingiz mumkin. Garchi koʻp darajali joylashtirish kontseptsiyasi oʻrnatilgan tizimlarning loyihalashi uchun oʻziga xos boʻlmasa ham, oʻrnatilgan tizimni ishlab chiqish uchun ishlatilishi mumkin boʻlgan yuzlab, minglab apparat va dasturiy komponentlarning kombinatsiyasini vizualizatsiya qilish uchun foydali vositadir. Ichki tizim arxitekturasining ushbu modulli koʻrinishi asosiy tuzilma sifatida ikkita asosiy sababga koʻra tanlanadi:

1.Asosiy elementlar va tegishli funksiyalarning vizual taqdimoti. Koʻp darajali yondashuv oʻquvchilarga oʻrnatilgan tizimning turli tarkibiy qismlari va ularning oʻzaro munosabatlarini tasavvur qilish imkonini beradi.

2.Modulli arxitektura koʻrinishlari odatda oʻrnatilgan loyihani tuzishda ishlatiladigan tuzilmalardir. Bu asosan ushbu turdagi turli xil



modullar (elementlar) funksional jihatdan mustaqil bo'lganligi sababli sodir bo'ladi. Ushbu elementlar yuqori darajadagi o'zaro ta'sirga ega, shuning uchun ushbu turdagi elementlarni qatlamlarga bo'lish tizimning tarkibiy tuzilishini yaxshilaydi, murakkab o'zaro munosabatlarini soddalashtirish yoki zarur funktsiyalarni e'tiborsiz qoldirmasdan.

## **2.4. O'rnatilgan tizimlarning yuqori darajadagi loyihash holati va istiqbollari**

### **2.4.1. O'rnatilgan tizimlarini loyihalash**

Umumiy va o'rnatilgan tizimlarda hisoblash tizimlarini loyihalashtirishning hozirgi amaliyoti, xususan, yuqorida ta'kidlab o'tilganidek, dasturiy qo'shimchalar yordamida hal etiladigan kanonik hisoblash platformalaridan birini (HP) tanlashdan iborat. Vazifa ikki qismga bo'linadi: tayanch (platforma) tanlangan, kerakli o'rnatilgan tizimni olish uchun tayanch (keng ma'noda dasturlash tufayli) qurilgan. Ushbu loyihalash usuli uchun texnologik usullar va vositalar mavjud. Masalan, yakuniy vazifani, tavsiflovchi dasturlash tillari, aktuatorlar (kompyuterlar) va tarjimonlar. Ikkinchi variant ham qo'llaniladi: HP tanlandi va kengaytma bilan bir qatorda pastga o'zgartirish ham amalga oshiriladi. Bunday holda, tayanch platforma ham prototip vazifasini bajaradi. Ushbu usul juda murakkabligi sababli kamroq qo'llaniladi.

O'rnatilgan tizimlarni ishlab chiquvchilarning **birinchi muammosi** quyidagilar:

- mavjud dasturlash tillari ideallashtirilgan virtual (til) mashina uchun vazifaning tavsifini talab qiladi;
- tarjimon ushbu til mashinasini ma'lum bir cheklovlarni hisobga olgan holda va ishlaydigan mashinaning ko'plab muhim texnik xususiyatlarini (masalan, kirish-chiqish tizimining xususiyatlari, himoya mexanizmlari) hisobga olmagan holda haqiqiy mashinaga joylashtiradi;
- ushbu cheklashlar va o'ziga xosliklarni hisobga olish uchun dasturchi tilni bilishdan tashqari, tarjimon va ijrochi mashinani bilishi kerak.

Bugungi kunda ushbu uchta komponentni tavsiflash uchun yagona tizim mavjud emas, ular turli mavzularda va til uslublarida tasvirlangan.

**Ikkinchi muammo** - bu dasturlarni amalga oshiradigan qo'shimchalari bo'lgan kanonik HPlar asosida amalga oshirish sxemasiga mos kelmaydigan juda ko'p vazifalar. Ushbu yondashuvda

echimlar iqtisodiy jihatdan maqbul emas yoki muammo zamonaviy texnik vositalar doirasida umuman hal qilinmaydi. Ixtisoslashgan HP-larni loyihalashtirish kerak, masalan, yuqori darajadagi parallelizm va ixtisoslagan operatsion birliklarni. Maqsadli oʻrnatilgan tizimlarni yuqorida tavsiflangan ommaviy an'anaviy texnologiyadan foydalangan holda loyihalashtirishga urinish til mashinasi, tarjimonlar va HPlarning aniq talablariga nomuvofiqlik muammosini yanada kuchaytiradi. Agar an'anaviy loyihalash sxemasi ma'lum bir vazifa sinfi uchun maqbul bo'lsa, u holda ixtisoslashtirilgan tizimlarning loyihalashi samaradorligi nolga tushirilishi mumkin (parallelizm bilan bog'liq katta muammolar, mudofaa mexanizmlari, real vaqt cheklovlari, sinov va kayta sozlash va boshqalar).

**Uchinchi muammo-** oʻrnatilgan tizimlardagi loyihalash hajimlarinig doimiy oʻsib borishi bilan bog'liq. Yuqorida keltirilgan tavsiflarning parchalanishi ishlab chiqilgan tarkibiy qismlardan (apparat bloklari, dasturlar, algoritmlarni amalga oshirish va boshqalar) qayta foydalanishni oldini oladi. Loyihalash samaradorligini oshirishda asosiy yoʻnalish, etakchi mutaxassislar ta'kidlaganidek, ishlanmalar natijalarini amalga oshirish uslubiga mos kelmaydigan mavhum texnik echimlar shaklida birlashtirishdir. Bunday darajada oʻrnatilgan tizimlarni loyihalashga oʻtish uchun tegishli metodologiya, til bazasi va vositalarning etishmasligi toʻsqinlik qilmoqda.

Bugungi kunda bozorda oʻrnatilgan tizimlarni loyihalashning barcha variantlari talabga ega:

- Dasturiy ta'minot (dasturiy ta'minot);
- Ilova va tizim dasturlari;
- Ob'ektga ulanish uchun moslamalar (OUM), dastur, tizim dasturlari;
- "Markaz" uskunalari, aloqa vositalari, OUM, amaliy va tizimli dasturlar.

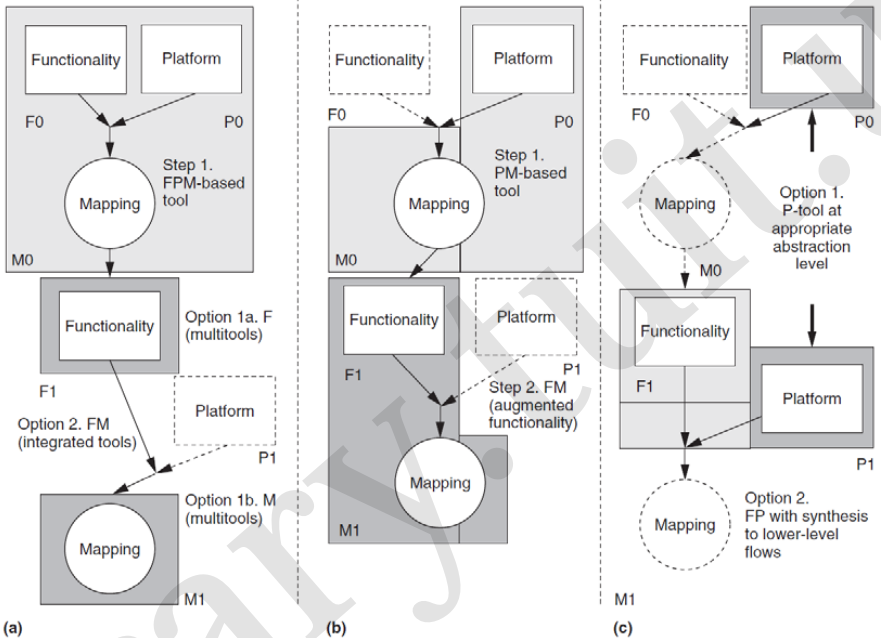
Loyihalash uchta tipik stsensariy doirasida amalga oshiriladi (2.3-rasm):

- maqsadli oʻrnatilgan tizimlar ;
- platforma oʻrnatilgan tizimlar ;
- mavjud oʻrnatilgan tizimlarni oʻzgartirish.

Oʻrnatilgan tizimlarning an'anaviy loyihalash jarayonini tahlil qilish quyidagi kamchiliklarni ta'kidlashga imkon beradi:

- dastlabki bosqichda dasturiy va dasturiy ta'minotning norasmiy ajralishi;

- asbob-uskunalar va dasturlarning izchil loyihalashi;
- uskunalar va dasturlarni alohida modellashtirish;
- loyihaning apparat va dasturiy qismlarini qo‘lda birlashtirish;
- dasturga kiritilgan o‘zgartirishlar tufayli aniqlangan xatolarni qoplash.



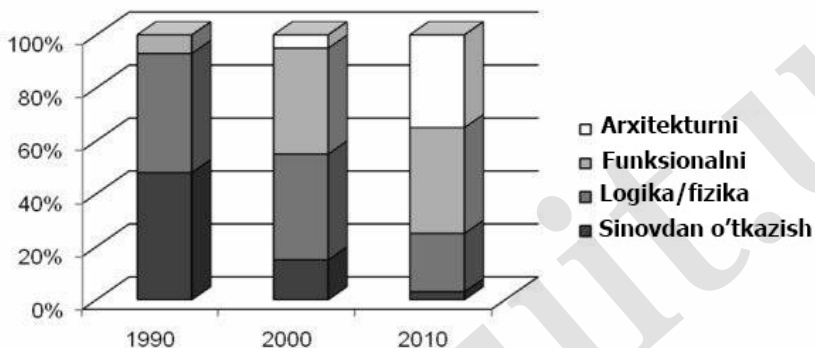
Rasm 2.3. O‘rnatilgan tizimlarni oddiy loyihalash senarisi

O‘rnatilgan tizimlar va jarayonlar loyihalashi vositalarining rivojlanishidagi asosiy tendentsiyalar o‘ziga xos tortishish hajmining oshishi va arxitekturiy, funktsional, mantiqiy loyihalash bosqichlarining ahamiyatini kuchaytirish bilan bevosita bog‘liqdir:

- loyihalashni abstraksiya qilish darajasini oshirish;
- modellashtirish, rasmiy tahlil va modelni tekshirish usullaridan keng foydalanish;
- hisoblash jarayonining mavhum vakillik darajasini ta’kidlash;
- o‘rnatilgan dasturlarni (ODT) yaratish uchun texnologiyalarni ajratish.

Loyihalashning yuqoridagi bosqichlari bugungi kunda o‘rnatilgan tizimlarning umumiy ishlab chiqarish hajmining 70 foizini tashkil etadi

(2.4-rasm) va yuqori darajadagi loyihalash sohasiga tegishli (HLD), uning markaziy tushunchasi mavhum deb hisoblanishi kerak - oʻrnatilgan tizimlarni tahlil qilish va sintez qilish jarayonida ularning muhim va doimiy xususiyatlarini ajratib koʻrsatish.



Rasm 2.4. Oʻrnatilgan tizimlarni loyihalash bosqichlarining nisbiy murakkabligi

Loyihalashlar hisoblash jarayonining mavhum modellari va butun loyihalash yoʻnalishidagi oʻrnatilgan tizimlar bilan ishlash doirasini kengaytirishi oʻrnatilgan tizimlarning loyihalashi sifatiga bevosita bogʻliqligini koʻrish mumkin.

Bugungi kunga kelib, yuqori darajadagi loyihalash bosqichining vazifalari, oʻrnatilgan tizimlar loyihalash bosqichini qamrab oluvchi va amalga oshirish bosqichiga taʼsir qilmasdan oltita katta bloklarga guruhlangan:

1. Maqsadli muammoni hal qilish tushunchasi, oʻziga xos xususiyatlar.
2. Hisoblash jarayonini tashkil qilish (hisoblash modeli).
3. Arxitektura va mikroarxitektura avlodlari.
4. Me'moriy echimlarni baholash va tanlash.
5. Arxitektura echimlarini tekshirish.
6. Amalga oshirish bosqichi uchun chiqish xususiyatlari.

Ushbu bosqichlar an'anaviy oʻrnatilgan Y-diagramma loyihalashining tizim darajasiga tegishli boʻlib, uni quyidagicha taqsimlaydi:

- **Arxitektura darajasi** –amalga oshirish usulidan qat'iy nazar tizim arxitekturasi: funktsional va funktsional boʻlmagan talablar / cheklolarga muvofiqligini arxitektura tahlili.

• **Mikroarxitektura darajasi** yoki **funksional** yoki **Electronic System Level (ESL)** - tizimni amalga oshirish bosqichida arxitektura modelining tarkibiy qismlari, algoritmlari, interfeyslari ishlab chiqiladi, spetsifikatsiyalar va tekshirish muhiti tayyorlanadi.

HLDning eng muhim muammosi HLD tavsiflari va spetsifikatsiyalari o'rtasidagi, shuningdek amalga oshirish uchun ham, xususiy arxitekturalarni (mikroarxitekturalar) yaratishda ham semantik bo'shliqni bartaraf etishdir. O'rnatilgan tizimlar uchun samarali loyihalash texnologiyalarini yaratishga qaratilgan tadqiqotlar ko'plab sohalarda olib borilmoqda. Oldingi bo'limlarda ko'rsatilgan muammolarni bartaraf etish bir qator muammolarni hal qilish natijasida mumkin. Ularning asosiylari ko'rib chiqilishi kerak:

- dasturiy ta'minot va dasturiy ta'minot komponentlarini loyihalash sohasini haqiqiy integratsiyalash;

- arxitektura loyihalashi bosqichini rasmiylashtirish masalasini ilgari surish;

- dasturiy va dasturiy mahsulotlarni tekshirish va sinovdan o'tkazish texnologiyalarini o'zgartirish;

- apparat vositalarini yaratish bilan bir vaqtda dasturiy mahsulotni samarali loyihalash uchun vositalarni yaratish;

- barcha darajadagi o'rnatilgan tizimlarning loyihalash mahsulotlari uchun qayta foydalanish texnologiyalarini yaratish va joriy etish;

- real vaqt tizimlarining algoritmik modellarini ishlab chiqish;

- o'rnatilgan tizimlarning ishonchligi mexanizmlarini takomillashtirish.

O'rnatilgan tizimlarning yuqori darajadagi loyihalashida eng muhim o'rin asboblari (SAPR), shu jumladan ulardan foydalanish usullariga ega. Tadqiqot manzarasi hozirgi paytda asosan qismlarga bo'linmagan va turli tadqiqotchilar guruhlariga muayyan muammolarni echishga harakat qilmoqdalar va tavsiflangan muammoni hal qilishda o'z yondashuvlarini taklif qilmoqdalar. Yondashuvlar turli darajadagi murakkabliklarga, rasmiylashtirishga va keng tarqalishga ega. Shubhasiz, bugungi kunda HLD vositalarini faol rivojlantirish zarur.

#### **2.4.2. O'rnatilgan tizimlarning HLD vositalarining holati va muammolari**

Asosiy vositalar sifatida biz vositalarning ikkita toifasini ko'rib chiqamiz (2.2-jadval):

- **Sanoat mahsulotlari** (tizimli SAPR tizimlarining etakchi ishlab chiqaruvchilaridan, loyihalash va dasturlashning oʻrta va quyi darajalarini qisman qamrab oladi; "sanoat" tillari va cheklangan miqdordagi MoS-dan foydalanadi).

- **Akademik mahsulotlar** (Berкли, Kanzas, Grenobl universitetlari, turli xil MoS, maxsus tillardan foydalanadilar).

Oʻrnatilgan tizimlar uchun tizim loyihalashi sohasidagi tadqiqotlarni rivojlantirishni ragʻbatlantiradigan mavjud metodologiyalar, vositalar va loyihalash muhiti (ramkalar) ning keng tarqalgan muammolari :

- Loyihalashning alohida (ayniqsa, funktsional boʻlmagan) jihatlari tizimda alohida ishlab chiqilgan modellar doirasida koʻrib chiqiladi: energiya sarfi, ishonchlilik, axborotni himoya qilish va boshqalar. Arxitektura tarkibiy qismlarining tuzilishi va tavsif rasmi ishlab chiqilishi kerak, bu arxitekturiy toʻplam kontseptsiyasida aks ettirilgan tomonlarini (ogʻirliklarini) aniq taklif qiladi.

- Me'moriy darajada funktsional jihatning aniq ustuvorligi saqlanib qolmoqda. Funktsional boʻlmagan jihatlar yoki qoʻllab-quvvatlanadigan loyiha mezonlarining ichki tizimlari rolini oʻynaydi yoki ularning aksariyati umuman hisobga olinmaydi. Ammo umumiy holda, oʻrnatilgan tizimlar uchun funktsionallik har doim ham ustuvor talab emas. Tizimni arxitekturiy tavsiflash vositasini ishlab chiqishda ushbu muammolarni hisobga olish kerak.

- Koʻp mikroarxitekturani amalga oshirish imkoniyatlari ish vaqtining HW/SW fazasi bilan cheklangan. Arxitektura virtualizatsiyasining turli darajalarini hisobga olgan holda, loyihalash echimlarini izlab, koʻp hollarda tizim loyihalashi doirasida instrumental tarkibiy qismni birlashtirish amalga oshirilmaydi.

Tadqiqotchilarning turli guruhlari muayyan muammolarni echishga harakat qilmoqdalar va tavsiflangan muammoni hal qilish uchun oʻzlarining yondashuvlarini taklif qilmoqdalar. Yondashuvlar turli darajadagi murakkabliklarga, rasmiylashtirishga va keng tarqalishga ega. Asosiy asosiy tendentsiyalar va muhim tadqiqot yoʻnalishlari quyida qisqacha tavsiflangan.

## 2.2-jadval.HLD vositalari

<b>Provider</b>	<b>Tools</b>	<b>Focus</b>	<b>Abstraction</b>	<b>Category</b>
Mentor Graphics	SystemVision	Mixed-signal and high-level simulation	VHDL-AMS, Spice, C	F: Industrial
National Instruments	LabView	Test, measurement, and control application development	LabView Programming language	F: Industrial
Univ. of California Berkeley	Ptolemy II	Modeling, simulation, and design of concurrent, real-time, embedded systems	All MoCs	F: Academic
Univ. of Kansas	Rosetta	Compose heterogeneous specifications in a single declarative semantic environment	All MoCs	F: Languages
Mentor Graphics	Nucleus	Family of real-time operating systems and development tools	Software	P: Industrial
Open SystemC Initiative	SystemC	Provide hardware-oriented constructs within the context of C++	Transaction level to RTL	FP: Languages
Object Management Group	Unified Modeling Language	Specify, visualize and document software system models	Object-oriented diagrams	FP: Languages
Cadence	Incisive	Integrated tool platform for verification, including simulation, formal methods, and emulation	RTL and SystemC assertions	PM: Industrial

Synopsys	System Studio	Algorithm and architecture capture, performance evaluation	SystemC	FPM: Industrial
Univ. of California, Berkeley	Metropolis	Operational and denotational functionality and architecture capture, mapping, refinement, and verification	All MoCs	FPM: Academic
Univ. of California, Berkeley	Mescal	Programming of application-specific programmable platforms	Extended Ptolemy II, network processors	FPM: Academic

## 2.5. O‘rnatilgan tizimlarni loyihalash usullari

So‘nggi 25 yil ichida yuqori darajadagi loyihalash texnikasining o‘rnatilgan tizimlari faol ishlab chiqilgan. Ularning asosiy yo‘nalishlari:

- Ob'ektga yo‘naltirilgan loyihalash (OYL).
- Parallel apparat va dasturiy ta‘minot loyihalashi (Hardware / Software CoDesign):
  - Komponent va platformaga yo‘naltirilgan loyihalash.
  - Aktyorga qaratilgan loyihalash.
  - Ko‘p tilli loyihalash.
  - Aspektli loyihalashi.

Mutaxassislar turli xil matematik modellarni, rasmiy tavsiflarni va algoritmlarni taklif qilishadi va tahlil qilishadi. Biz avtomat yondoshuvlarini (Petri tarmoqlari, asinxron va sinxron avtomatlar), ko‘p tilli tavsiflarni, qo‘shma apparat-dasturiy modellashtirish algoritmlarini va tekshirish, loyihalash natijalarini qayta ishlatish texnologiyasi, aspektli dasturlarni ko‘rib chiqamiz.

### 2.5.1. Hamkorlikagi apparat va dasturiy ta‘minotlarni loyihalash

An'anaviy mikroprosessor texnologiyalaridan keng foydalanish va integrallashgan dasturlashtiriladigan mantiqning tez rivojlanishi adabiyotda " Hardware–Software CoDesign " deb nomlangan apparat va



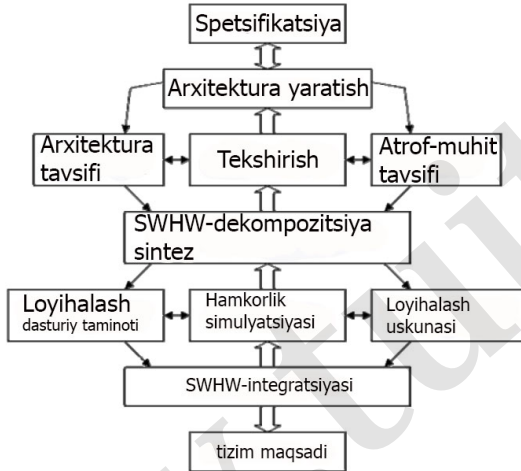
dasturiy ta'minotni birgalikda loyihalashda yangi falsafaning paydo bo'lishiga olib keldi. O'rnatilgan tizimlarni loyihalashning yaxlit muhitini ishlab chiquvchilarning loyihalash tilining bazasini integratsiyalash, hisoblash murakkabligini oldindan baholash va keyinchalik tekshirish va sinov jarayonlari tizimining simulyatsiya modellari ierarxiyasini yaratish istagida aks etadi. Hardware-Software CoDesign bugungi kunda o'rnatilgan tizimlarni loyihalashda istiqbolli yondashuvlardan biri sifatida qaraladi. Ushbu texnologiya nisbatan yosh, u to'qsoninchi yillarning birinchi yarmida paydo bo'lgan. Hozirgi vaqtda dunyoning bir qator universitetlarida ishlab chiqilgan bir nechta notijorat SAD tizimlari (Polis, Ptolemey, Chinook va boshqalar) mavjud. Zamonaviy o'rnatilgan tizimlarga nisbatan hisoblashlarni parallellashtirish va taqsimlash tezligini ko'pgina ishlab chiquvchilar loyihalashning murakkablashuviga qaramasdan, ularning arxitekturiy tashkil etilishining asosi sifatida ko'rib chiqadilar. Bu muqobil variantlarga nisbatan yoki boshqa texnik echimlarning tubdan etishmasligi fonida tayyor mahsulot xususiyatlarining sezilarli yaxshilanishi bilan bog'liq. Bunday tizimlarni loyihalashning murakkabligining oshishi ob'ektiv jarayon bo'lib, taqsimlangan va parallel o'rnatilgan tizimlarning elementlarining ishlashining natijasida muammoning o'lchamining keskin oshishi bilan belgilanadi. Yuqorida ko'rsatilgandek, bugungi kunda amalda qo'llaniladigan loyihalash usullari va vositalari bizga loyihalashning quyi darajalarini - elektron, loyihalash, dasturlash (kodlash) ni rasmiylashtirish va avtomatlashtirishga imkon beradi. Ma'lum darajada, yuqori darajalarni rasmiylashtirilgan deb hisoblash mumkin - tizim spetsifikatsiyasini va arxitekturiy loyihalashni yaratish (kamroq darajada). Apparat va dasturiy ta'minotning yuqori darajadagi tavsifidan bajarishga o'tish rasmiylashtirilmagan. Loyihani avtomatik ravishda apparat va dasturiy qismlarga bo'linishini da'vo qiladigan CoDesign, taniqli dasturlarda ushbu muammoni bir qator kanonik tuzilmalar va cheklangan elementlar bazasi uchun hal qilishga imkon beradi.

### **CoDesign texnologiyasi**

CoDesign loyihalash texnologiyasining quyidagi bosqichlarini taklif etadi:

- Talablarni ishlab chiqish;
- Rasmiy yondashuv asosida tizim xatti-harakatlarining modelini ishlab chiqish (Petri to'rlari, CFSM va boshqalar);

- Modelni rasmiy tahlil qilish, tekshirish, simulyatsiya, talablarni aniqlashtirish;
- Apparat va dasturiy ta'minotga ajratish;
- Dasturiy ta'minot va apparat tarkibiy qismlarini ishlab chiquvchilar yoki avtomatik kod yaratish uchun texnik shartlar berish.



Rasm 2.5. Codesign doirasida loyihalash jarayoni

Dasturiy ta'minot va apparat tarkibiy qismlarini ajratishga o'tish bir qator usullarga ko'ra sodir bo'lishi mumkin:

- Simulyatsiyalangan yumshatish algoritmi (Henkel, Ernst);
- Barcha apparat echimlari (Gupta, De Micheli);
- Jantsch algoritmiga asoslangan dinamik dasturlash algoritmi;
- PACE (LYCOS) dinamik dasturlash algoritmi.

Umuman olganda, ushbu metodologiyalar ajratish variantlarini sintez qilish va tahlil qilish ketma-ketligidir. Ba'zi bir metodologiyalarda dasturiy ta'minotdan apparat vositalarda amalga oshirishga o'tish (masalan, simulyatsiyalangan yumshatish algoritmi), ba'zilarida apparat vositalaridan dasturiy ta'minotga o'tish (masalan, barcha apparat echimlarida) mavjud.

CoDesign tizimlarida dasturiy va dasturiy qismlarga bo'lingandan so'ng, kod avtomatik ravishda FSM dan C va VHDL (yoki shunga o'xshash) ga yaratiladi. Bir qator tizimlarda dasturiy ta'minot tizimining ibtidoiy elementlari sifatida RTOS-da ishlatiladigan klassik mexanizmlar qo'llaniladi.

Tegishli loyihalash vositalariga misollar:

- Cierto VCC - Cadence loyihalash tizimidan CODESIGN texnologiyasida integratsiyalashgan loyihalash tizimi;

- COSYMA - dasturiy va dasturiy komponentlarni baholash va sintez qilishning universal tizimi, Germaniyaning Barunshve universiteti;

- LOCOS - Daniya texnik universiteti apparat-dasturiy ta'minot tizimining tarkibiy qismlari uchun grafik ma'lumotlari oqimini sintez qilish va dekompozitsiya qilish tizimi;

- POLIS - turli xil apparat va dasturiy qismlarni loyihalashtirish uchun universal tizim, Kaliforniya universiteti, Berkli (AQSh) va Turin politexnika universiteti (Italiya) qo'shma loyihasi;

PTOLEMY - heterojen tizimlarni modellashtirish uchun universal tizim, Kaliforniya universiteti, Berkli (AQSh);

- CoWare - sozlanadigan tuzilishga ega bo'lgan dasturlashtiriladigan mikrosxemalarga asoslangan heterojen bitta chipli tarkibiy qismlarni loyihalash tizimi. Leuven, Belgiya.

### **2.5.2. Platformaga yo'naltirilgan loyihalash tushunchasi**

PBD (Platform Based Design) metodologiyasi bir necha yil oldin o'rnatilgan dasturiy ta'minot va dasturiy ta'minotlarning yaxlit tizimli loyihalashi kontekstida paydo bo'lgan va loyihalash muammolarini hal qilish uchun istiqbolli texnologiyalar orasida alohida o'rin egallaydi. Kontseptsiya mualliflari ta'kidlashicha, bu ishlab chiqilgan tizimlarning samaradorligini oshiradi, ularning narxini, ishlab chiqarish xarajatlarini kamaytiradi va eng muhimi - ishlab chiqilgan tizimlarning ikkala apparat va dasturiy komponentlaridan qayta foydalanish imkonini beradi.

Berkli (Kaliforniya shtati) universiteti tadqiqotchilari guruhi tomonidan xalqaro hamjamiyat tomonidan ilgari surilgan ushbu loyihalash kontseptsiyasi asosan "apparat loyihalashi" tomonidan ishlab chiqilgan: guruh rahbarlari Cadence, Mentor Graphics, Synopsis kabi etakchi kompaniyalarning texnik mafkurachilari. PBD ning afzalliklari tizimning dasturiy va apparat tarkibiy qismlariga qarashni birlashtirish, arxitekturiy loyihalash bosqichining aniq ustuvorligi, tizim darajasidagi modellarni yaratish va tahlil qilish jarayonini matematik rasmiylashtirishga e'tibor berish ("hisoblash modellari", MoS).

Mualliflar o'zlarining ishlarida o'rnatilgan tizimlarni loyihalashning barcha bosqichlarida qayta foydalanish koeffitsientining sifat jihatidan o'sishini ta'kidlamogdalar. Bozor sharoitida tizim tarkibiy qismlarini (platalarr yoki ularning qismlari, drayverlar, interfeyslar va boshqalar) qayta ishlatishdir. Qayta foydalanish uchun muhim bo'lgan muqobil echimlarni yaxshiroq o'rganish uchun tizimni ikki nuqtai nazardan va har xil nuqtai nazardan ko'rib chiqish taklif etiladi.

- funktsionallik (tizim nima qiladi) va arxitektura (tizim buni qanday amalga oshiradi);
- o'zaro aloqa (ma'lumotlar almashinuvi) va hisoblash (tizimning ob'ektiv funktsiyasini bajaruvchi).

Funktsionallikni ishlab chiquvchi ham, buyurtmachi ham aniqlashi mumkin. Birinchi holda, funktsional loyihalash bosqichi rivojlanish jarayoniga kiritiladi. Arxitektura interfeysni belgilaydi va amalga oshirishning funktsional imkoniyatlarini tavsiflaydi. Biroq, bu amalga oshirishga bog'liq bo'lmasligi kerak. Arxitekturadan tashqari, mualliflar **m-arxitektura** yoki **mikroarxitektura** tushunchalarini taqdim etadilar. **M-arxitektura** - bu hisoblash tarkibiy qismlarining funktsional jihatdan to'liq to'plami (mikroprosessor, atrof-muhit, dasturlashtiriladigan mantiq, xotira). Loyihalashda ishlab chiquvchi quyidagi printsiplarga amal qilishi kerak:

- qaror qabul qilish jarayonini ishlab chiqish vaqti va qiymati;
- loyihalash yuqori mavhumlikda bajarilishi kerak;
- ishonchli, barqaror tizimlar yaratilishi kerak;
- tizim bir nechta murakkab va ko'plab oddiy apparat qismlariga ega bo'lishi kerak;
- ushbu komponentlarni dasturlashtirish turli darajalarda amalga oshiriladi.

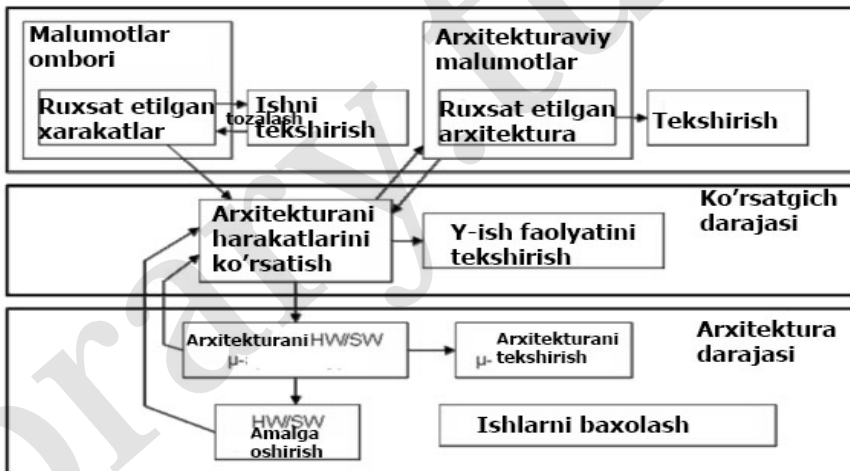
Ko'pgina bunday tushunchalar singari, PBD kontseptsiyasi ishlab chiquvchiga nisbatan o'ziga xos loyihalash retseptlarini taklif qilish uchun etarli emas. Taklif etilayotgan metodologiyaning tezislari haqiqatdir va ko'plab ishlab chiquvchilarga ma'lum. Muammo shundaki, metodologiya qanday qilib loyihalashi, ya'ni uni qanday qilish kerakligini va nima qilish kerakligini aytmaydi.

Platformaga yo'naltirilgan tizim loyihalashi kontseptsiyasida asosiy tushuncha platforma tushunchasidir. Mualliflar dasturiy va dasturiy platformalarni ajratib ko'rsatishadi, ular har bir loyihada ishtirok etadi va ularning kombinatsiyasi tizim platformasini tashkil etadi.

**Apparat platformasi** - bu kompyuter vazifalarini hal qilishga imkon beradigan kompyuter arxitekturalari to'plamidir. Loyihalashda arxitekturiy cheklolvar odatda ishlash va hajm jihatidan aniqlanadi. Qoida tariqasida, apparat platformasi mo'ljallangan tizimdan talab qilinadiganidan ko'proq xususiyatlarga ega. Bitta vazifani hal qilish uchun mos jihozlardan foydalanishning ma'nosi yo'q.

**Dasturiy ta'minot platformasi** - mavhum o'zaro ta'sir darajasi uskunarlar bilan dasturlar. Asosiy g'oya platformadan mustaqil API yaratishdir. Ya'ni, dastur va uskuna o'rtasida dasturning ishlashini ma'lum jihozlar to'plami bilan birlashtiradigan dastur qatlami qo'yiladi. Dasturiy platforma quyidagilarni o'z ichiga oladi:

- apparat resurslarini tarqatuvchi operatsion tizim (odatda RTOS);
- kirish / chiqish quyi tizimini ta'minlovchi qurilma drayverlari;
- hisoblash tizimining tarkibiy qismlarining o'zaro ta'sirini ta'minlaydigan tarmoq quyi tizimi.



Rasm 2.6. Platformaga yo'naltirilgan loyihalash jarayoni

Dasturiy ta'minot platformalariga misollar ISI, WindRiver, Microsoft (Windows CE), QSSL (QNX), POSIX mahsulotlari. Platformaga yo'naltirilgan tizim loyihalashi strategiyasi rasm. 2.6. Tizim platformasi, yuqorida aytib o'tilganidek, apparat va dasturiy platformalarni birlashtiradi. loyihalash boshida ikkala platforma ham mavhumdir. Bundan tashqari, ushbu abstraktsiya qanchalik baland bo'lsa, shuncha yaxshi, tizimni konkretlashtirish bo'yicha qarorlarni

tanlashda erkinlik bo'ldi. Funktsiyalarni apparat va dastur o'rtasida oldindan ajratish istalmagan. Cheklovlar (tezlik, o'lchovlar, ishonchlilik, energiya iste'moli, mavjud bo'lgan API, eng mos RTOS va boshqalar) qo'shilgan holda ishlab chiqilgan tizim takomillashtirilmoqda (yangilanmoqda) va uni amalga oshirish uchun kamroq va kamroq imkoniyatlar mavjud. Natijada bitta echim bo'ldi: uskunalarni aniq tanlash va ma'lum dasturiy model.

Eng kuchli va keng qamrovli PBD vositalari bugun Metropolis Framework (Center for electronic system design, University of California at Berkeley) vositalari hisoblanadi. Metropolis Framework funktsional va arxitektura darajasiga modellarni qo'llab-quvvatlaydi. Jarayonlar bilan bog'liq resurslarni boshqaradigan maxsus "miqdor menejeri" ob'ektlari ko'rinishidagi funktsional bo'lmagan talablarni qo'llab-quvvatlash mexanizmi taqdim etilgan.

Xuddi shu universitetdagi PBD toifasidagi yana bir taniqli loyiha Ptolemey II tizimi. Ptolemey II cheklangan tizimlarni rivojlantirishga emas, balki turli xil o'rnatilgan tizimlarning arxitekturasi uchun asos bo'lishi kerak bo'lgan turli xil hisoblash modellarini o'rganishga qaratilgan. Ptolemey II ochiq, u tayyor modellar va arxitektura tarkibiy qismlarining katta kutubxonasini taqdim etadi. Shunday qilib, Ptolemey II tadqiqot vositasi, ammo loyihalash vositasi emas.

Tijorat vositalaridan MLDesigner (Mission Level Designer, [www.mldesigner.com](http://www.mldesigner.com)) eng yaxshi tanilgan. MLDesigner ko'p jihatdan vazifa va arxitektura bosqichlarini (pastki) qo'llab-quvvatlaydi, funktsional bo'lmagan talablarni boshqarish vositalariga ega (MLDesigner atamasida - resurslar), ammo paketning ichki dasturiy ta'minotni ishlab chiqishga yo'naltirilganligi aniq kuzatilgan, dasturiy ta'minot / apparat bo'linishi va arxitektura tarkibiy qismlarini tatbiq etish vositalari yo'q.

### **2.5.3. MDD metodologiyasi**

MDD metodologiyasi (Model Driven Design, modelga yo'naltirilgan) real vaqt tizimlari va o'rnatilgan tizimlarni (MDD RTES) loyihalash uchun OMG tomonidan tavsiya etilgan tizim. Ushbu yo'nalish Evropaning ilmiy va tijorat tashkilotlari tomonidan, xususan, ARTEMISIA (Advanced Research & Technology for Embedded Intelligence and Systems Industrial Association) tomonidan ishlab chiqilgan Evropa texnologik platformasi GENESYS (GENeric

Embedded SYSTEM platformasi). Ushbu yo‘nalishning afzalligi - ko‘p sonli va keng tasdiqlangan til (UML) va MDD vositalariga ishonish, turli xil amaliy va texnologik sohalar uchun tizim arxitekturasi uchun turli jihatlarni (shu jumladan funksional bo‘lmaganlarni) tavsiflash uchun metamodel vositasidan foydalanish.

MDD RTESning kamchiliklari uning kelib chiqishining bevosita natijasidir - dasturiy ta‘minot yaratishga yo‘naltirish, apparat loyihalashi, qo‘shma loyihalash muammolariga zaif yoki sun‘iy moslashish.

MDD RTES infratuzilmasi jihatidan yuqorida aytib o‘tilgan GENESYS texnologik platformasi doirasidagi uslubiy va instrumental infratuzilma eng rivojlangan hisoblanadi. OMG UML profillari bu erda asosiy til vositasi sifatida ishlatiladi.

Vositalar nuqtai nazaridan GENESYS juda keng tanlovni taklif qiladi: Rational Software-dan UML dizayn paketlaridan Papirus (Grafical UML2 modellashtirish), Times (O‘rnatilgan tizimlarni modellashtirish va amalga oshirish vositasi), MAST (Real Vaqtni qo‘llash uchun modellashtirish va tahlil to‘plami) kabi bepul vositalargacha. GENESYS metodologiyasidagi bunday xilma-xil to‘plamlarning salbiy tomoni - bu katta mehnat sarflari va ushbu vositalarni bitta asbob-uskuna infratuzilmasi doirasida juftlashtirish samaradorligining pastligi.

Hozirgi vaqtda ARTEMISIA uyushmasi muvofiqlashtirilgan, erkin tarqatiladigan dasturiy ta‘minot to‘plamini yaratish bo‘yicha ish boshlanganligini e‘lon qildi.

GENESYSning sezilarli kamchiligi bu SW / HW Codesign vositalari va mikroarxitektura darajasidagi loyihalashning aniq ko‘rinadigan ikkilamchi roli, ya‘ni apparat bilan bog‘liq bo‘lgan barcha narsalar.

MDD RTES shuningdek, tijorat vositalarini ham taklif etadi bu CoFluent Studio va Mentor Graphics BridgePointUMLSuite tizimlar. CoFluent Studio MCSE yoki CoMES (Elektron tizimlar uchun Kodlar metodologiyasi) deb nomlangan o‘z metodikasiga tayanadi. CoMESning o‘ziga xos xususiyati ESL darajasidagi SystemC modelini yaratish va keyinchalik qo‘shimcha qurilmalar loyihalashiga chiqishdir. Kamchilik yuqori darajadagi arxitekturiy tahlilning juda kuchli vositasi emas. BridgePointUML Suite to‘plami, aksincha, o‘rnatilgan dasturni yaratishga (avtomatik generatsiyalashga) qaratilgan bo‘lsa-da, uni RTL-kodni ishlab chiqish bo‘yicha ish olib borilgan.

#### **2.5.4. O‘rnatilgan tizimlarni loyihalash texnikasini rivojlantirish**

Yuqorida faqat tizim darajasini loyihalash sohasini rivojlantirish uchun eng qiziqarli va ahamiyatli bo‘lgan vositalar keltirilgan. Ushbu munozaradan tashqarida SIMULINK (MathWorks) ga asoslangan ko‘plab vositalar qoldi, chunki bu sohada loyihalash uslublarini ishlab chiqishdan ko‘ra ko‘proq tijorat va amaliy yo‘nalish mavjud. Shunisi e‘tiborga loyiqli, elektronika va o‘rnatilgan tizimlar sohasidagi etakchi SAPR ishlab chiqaruvchilari ushbu sohada juda kam texnik va vositalarga ega - ular asosan ( sub ) arxitektura darajasidagi vositalarga e‘tibor berishadi: TLM / RTL-loyihalash, SystemC HW / SW qo‘shma tasdiqlash ( masalan, System Studio (Synopsys), Palladium Series (Cadence), Vista / Visual Elita seriyalari (Mentor Grafika). Buning sabablari, asosan, tijorat samaradorligi sohasida bo‘lishi mumkin : ushbu firmalar tayyor mahsulotni kutish jarayonida ilmiy-tadqiqot ishlarini qo‘llab-quvvatlamogda.

Mavjud metodologiyalar, vositalar va loyihalash muhitining umumiy kamchiliklari o‘rnatilgan tizimlar uchun tizimlarni loyihalash sohasidagi tadqiqot ishlarini muayyan tanlash va rivojlantirishni talab qiladi.

Loyihalashning alohida (ayniqsa funktsional bo‘lmagan) jihatlari umuman tizim uchun tegishli tizim modellari doirasida ko‘rib chiqiladi : energiya iste‘moli, ishonchlilik, axborotni himoya qilish va boshqalar. Bu, albatta, tizimning individual arxitekturiy tarkibiy qismlarining ichki murakkabligi va kuchli o‘zaro bog‘liqligi bilan izohlanadi. Boshqa tomondan, ma‘lum bir komponentni amalga oshirishning aspekt ko‘rsatkichlariga (xususiyatlariga) ta‘sirini baholash imkoniyatlari cheklangan.

Arxitektura tarkibiy qismlarining tuzilishi va tavsifi aniq ishlab chiqilgan bo‘lib, u intellektual baholarni (og‘irliklarni) aniq taklif qiladi. Ushbu yondashuv arxitektura agregati (AA) kontseptsiyasida aks etadi.

Ko‘rib chiqilgan barcha metodologiyalarda arxitektura darajasida funktsional tomonning aniq ustuvorligi namoyon bo‘ladi. Funktsional bo‘lmagan jihatlar loyihaning yordamchi (ikkilamchi) mezonlari rolini o‘ynaydi yoki ularning aksariyati umuman hisobga olinmaydi. Shunday qilib, instrumental va konstruktiv-texnologik jihat e‘tiborga olinmadi. Ustuvor jihatlariga kelsak, funktsionallik har doim ham (hamma dasturlar uchun ham) asosiy vazifa bo‘lmasligi mumkin. Masalan, real vaqt rejimidagi talablar tarmoq orqali ma‘lumotlarni majburiy etkazib



berishning funktsional talablarini bekor qilishi mumkin; yoki ommalashib borayotgan ehtimoliy mantiqiy kontseptsiya, energiya sarflash jihatidan funktsiyalarni hisoblashning aniqligiga nisbatan ustuvorlikni oshirishi mumkin. Tizimni arxitekturiy tavsiflash vositasini ishlab chiqishda ushbu muammolarni hisobga olish kerak.

Arxitektura tarkibiy qismlarini amalga oshirish variantlari ko'lam asosan HW / SW ish davri davomida qolmoqda. Turli darajadagi arxitekturani virtualizatsiya qilish variantlari tizim loyihalashi doirasida hisobga olinmaydi.

## **2.6. O'rnatilgan tizim arxitektura tillarini tavsifi**

Kompyuter texnologiyasida protsessor axborotni qayta ishlash moslamasini anglatadi. Turli xil hisoblash modellari zamonaviy protsessorlar ostida bo'lishi mumkinligiga qaramasdan, Von Neumann mashinasiga asoslangan model eng ko'p ishlatilishini tarixan rivojlantirgan.

### **2.6.1. IRSYD geterogen tizimlarini tuzilish metodologiyasi**

O'rnatilgan tizimlarni loyihalash texnologiyalarining asosiy vazifalaridan biri bir qator qarama-qarshi talablarni qondiradigan samarali tizim tavsifini yaratishdir. Ushbu muammoga taklif etilayotgan yondoshuvlardan biri bu heterogen tizimlarning ichki vakolatxonasi tili (IRSYD – An Internal Representation for System Description Version 0.1). Yondashuv mualliflari quyidagi tushunchalarni aniqlaydilar:

- **Funksionallik** - tizimning maqsadli vazifalarini amalga oshirish uchun javob beradigan modelning ideologik parametrlari ;

- **Strukturasi** - ma'lum bir funktsional qismning tashuvchisi bo'lgan moddiy yoki virtual ob'ektlar bilan ifodalangan tizim modelining elementlari. Tarkibiy komponenti bo'lmagan modellar mavjud bo'lishi mumkinligi taxmin qilinadi.

- **Xulqi** - bu dinamik vakillik va barcha model parametrlarining o'zgarish dinamik qonunlari. Biz modelning strukturaviy va funktsional harakati haqida gapirishimiz mumkin.

Murakkab heterogen tizimlarning ichki vakili uchun usul (metodologiya) taklif qilingan. Bunday taqdimot uchun mezonlar sifatida quyidagi talablar e'lon qilingan: statik va dinamik tarkibiy va funktsional modullik va ierarxiya, turli quyi tizimlar o'rtasidagi aloqa,

mavhum ma'lumotlarning turlarini taqdim etish, "funktSIONAL bo'lmagan" axborotni taqdim etish mexanizmlari, ishlab chiqish shablonlaridan qayta foydalanish.

IRSYD tili doirasida murakkab tizimning tarkibiy, funkSIONAL va vaqtinchalik harakatlarini tavsiflash uchun vositalar taklif etiladi. Ushbu tavsifning asosiy g'oyasi tizimni boshqarish va ma'lumotlar oqimini ifodalovchi umumlashtirilgan grafik ko'rinishida taqdim etishdir. Shu bilan birga, funkSIONAL modullik va ierarxiya joriy etiladi, bu tizimning tuzilishi.

IRSYD metodologiyasi funkSIONAL tarkibiy qismga qaratilgan. Aynan ushbu komponent tizimni modellashtirish va loyihalashida birinchi o'ringa qo'yilgan. Shuni aytishim kerakki, IRSYD ishlab chiquvchilari ba'zi atributlar ko'rinishida ba'zi "funktSIONAL bo'lmagan" elementlarni modelga kiritish imkoniyatini qoldiradilar. Tizimni loyihalash va modellashtirish jarayonida atributlarni ko'rib chiqish huquqi ishlab chiquvchining ehtiyojlariga qarab har bir aniq loyihalash jarayoni uchun ishlab chiqilgan ba'zi vositalarga beriladi. IRSYD-da bunday vositalarni va atributlarni aniqlash mezonlarini yaratish uchun umumiy retseptlar mavjud emas.

Apparat / dasturiy ta'minotning dekompozitsiya muammosi IRSYD-da hal qilinadi. Muammoni hal qilish kerak va mualliflarning fikriga ko'ra, jarayonlarni strukturaviy elementlarda aks ettirish darajasida nisbatan oson. Afsuski, IRSYDda rasmiy usullar va HW / SW ajralish mezonlari taklif etilmaydi.

## **2.6.2. AADL o'rnatilgan tizim arxitekturasini tavsiflash tili**

AADL 2001 yilda Xalqaro avtomobil muhandislari uyushmasi tomonidan standart sifatida ishlab chiqila boshlandi. Dastlab u aviatsiya elektron tizimlari (Avionics Architecture Description Language) arxitekturasini tavsiflash uchun til sifatida joylashtirilgan, ammo amalda nafaqat aviatsion tizimlarni, balki barcha o'rnatilgan (va boshqa ba'zi) tizimlarni qamrab oladi. Shuning uchun endi AADL, tom ma'noda arxitekturiy tahlil va loyihalashning (Architecture Analysis and Design Language) tili hisoblanadi. Standart AADL matnli yozuvini belgilaydi, unga ilova arxitekturani grafik usulida tasvirlash usulini ham tavsiflaydi.

Bunday standartning mavjudligi ishlab chiquvchilarga quyidagi afzalliklarni beradi:

- yozuvning umumiy shakli aniq belgilangan, bu talqinning noaniqligini yo‘q qiladi ;
- tahlil qilingan xususiyatlar bilan to‘ldiriladigan yagona arxitekturiy model qo‘llaniladi ;
- arxitektura modellarini qayta ishlatish va birlashtirish mumkin;
  - arxitektura modellashtirish vositalarining o‘zaro ta’siri va kengayishi uchun imkoniyatlar yaratadi ;
  - tilning imkoniyatlari hisoblash tizimlarining arxitekturasini loyihalash va tahlil qilishning keng tarqalgan usullariga mos keladi.

AADL birinchi navbatda real vaqt cheklovlari mavjud bo‘lgan ichki tizimlarni modellashtirish va tahlil qilishga qaratilgan. Muvaffaqiyatli ishlashi uchun bunday tizimlar odatda xatolarga chidamlilik, xavfsizlik, ishlash va hokazolar uchun maxsus talablarga ega. AADL ob’ektlari va konstruk-tsiyalarining aniq belgilangan semantikasi yuqorida keltirilgan barcha jihatlar bo‘yicha kompyuter tizimlarini tahlil qilishga imkon beradi. Ushbu tilning odatiy dasturlari maishiy elektronika tizimlaridan, tibbiy asboblardan motor va aerokosmik boshqaruv tizimlarigacha bo‘lgan axborotni boshqarish tizimlarini loyihalashdir. Biroq, bu har qanday boshqa hisoblash tizimlarini loyihalashda AADL dan foydalanishga to‘sqinlik qilmaydi. AADL tilidagi har qanday arxitektura tavsifi deyarli butunlay uning individual tarkibiy qismlarining tavsiflaridan iborat. Har bir tarkibiy qism quyidagi toifalardan biriga tegishli:

**Xotira, xotira qurilmasi (memory)**- bu ikkilik ma'lumotlar va dastur kodini saqlash uchun xizmat qiluvchi apparat komponenti;

**Qurilma (device)** - bu "qora quti" tamoyiliga binoan ma'lum bir funksional xususiyatga ega, ichki tuzilishi oshkor qilinmaydigan ma'lum bir faol apparat bloki ;

**Protssessor (protssessor)** - dastur kodini rejalashtirish va bajarish bilan shug‘ullanadigan asosiy kompyuter ;

**Shina (bus)** - protssessorlar, saqlash moslamalari va qurilmalarini ulaydigan jismoniy interfeys (odatda aloqa protokoli bilan birgalikda);

**Ma'lumotlar (data)** - dastur kodidagi ma'lumotlar turini mavhumlashtirish, ehtimol murakkab ichki tuzilishga ega;

**Subdastur (subprogram)** - dastur kodidagi kirish nuqtasi, unga oqim yoki qo‘shimcha dasturni boshqarish paytida boshqarish mumkin.

**Oqim, boshqarish ipi (thread)** - bu protssessor nuqtai nazaridan yagona jo‘natish ob’ekti deb hisoblanadigan buyruqlarning bajariladigan ketma-ketligi ;

**Oqim guruhi (thread group)** -bir yoki bir nechta mavzularga mantiqiy ravishda bog'lanish uchun ishlatiladi;

**Jarayon, vazifa (process)** - alohida virtual manzillar maydoniga to'g'ri keladi va agar u kamida bitta boshqaruv oqimini o'z ichiga olsa, dasturning to'liq birligi hisoblanadi;

**Tizim (system)** - dasturiy va apparat tarkibiy qismlarining kombinatsiyasi, shuningdek boshqa tizimlar, ehtimol o'zaro bog'liq, to'liq yoki qisman ma'lum bo'lgan ichki tuzilishga ega.

Dastlabki to'rtta toifa hisoblash tizimining apparat ta'minotga (jihozlariga), keyingi beshta dasturiy ta'minotga mos keladi. Shunday qilib, AADL sizga dasturiy tizimlarni va ularning apparat platformalarini yaxshi aniqlangan semantika va munosabatlar bilan o'zaro ta'sir qiluvchi komponentlar sifatida tavsiflashga imkon beradi.

Tizim, maxsus toifaning tarkibiy qismi sifatida, ierarxik ravishda joylashtirilishi mumkin bo'lgan apparat va dasturiy ta'minot elementlarining tarkibi. Tizim mavhum asbobdan farqli o'laroq, "oq quti" tamoyilini ifoda etadi va murakkab arxitektura elementlarini o'zboshimchalik bilan modellashtirishga imkon beradi.

Qismlarning o'zaro ta'sirining uch turi qo'llab-quvvatlanadi:

- har xil turdagi portlar (ulanish porti, voqea porti, voqea ma'lumot porti) o'rtasidagi ulanish orqali modellashtirilgan yo'nalishli ma'lumotlar va boshqarish oqimlari ;
- keyinchalik qaytib keladigan pastki dasturlarning qo'ng'iroqlari;
- birgalikda ishlatiladigan ma'lumotlarga bir nechta tarkibiy qismlardan kirish (ma'lumotlarga kirish).

Aksariyat komponentlar uchun bir nechta ish rejimlarini o'rnatishga ruxsat beriladi, ularning har birida komponentning harakati boshqacha. Bir rejimdan boshqasiga o'tish qoidalari ham o'rnatildi. Har bir komponent ikki bosqichda AADL-da aniqlanadi. Birinchisi, turi tavsifi deb nomlangan, tashqi tomondan ko'rinadigan va foydalanish uchun mavjud bo'lgan komponentning funktsional interfeysini belgilaydi. Ikkinchi bosqich komponentni amalga oshirishni (amalga oshirishni), ya'ni uning ichki tuzilishini, shu jumladan subkomponentlarni, xususiyatlarni, aloqalarni va boshqalarni tavsiflaydi. Har bir komponent turi bir nechta bajarishga ega bo'lishi mumkin.

Shunday qilib, ob'ektga yo'naltirilgan yondashuvning (OYY) asosiy printsiplaridan birini qo'llab-quvvatlash aniq kuzatiladi: inkapsulasya. Bu tizim loyihalashida juda ko'p erkinlikni qoldiradi:

masalan, bir xil komponentning turini va bajarilishini tavsiflash turli loyihalash bosqichlarida, ehtimol turli muhandislar tomonidan amalga oshirilishi mumkin.

OYYning yana bir asosiy printsipti - merosxo'rlik - AADL tilida ham o'z aksini topgan. Komponent turini (boshqa turidan) va uning bajarilishini (mos ravishda, boshqa amalga oshirishdan) meros qilib olish mumkin. Merosning mavjudligi o'xshash tarkibiy qismlarning tavsifining mos keladigan qismlarini takrorlash zaruratini yo'q qiladi. Har bir AADL komponenti uchun, shuningdek tarkibiy qismlar, aloqalar, ma'lumotlar oqimi va boshqarish, rejimlar va qo'ng'iroqlar uchun istalgan miqdordagi xususiyatlarni aniqlash mumkin. Har bir mulk o'ziga xos nomga, ma'lumot turiga va aslida qiymatga ega. AADL-da xususiyatlar til tuzilmalarining semantik kengayishi uchun universal mexanizmdir, chunki foydalanuvchi ularni deyarli hech qanday cheklolrsiz yaratilgan modelning abstraksiyalariga muvofiq kiritishi mumkin. Bundan tashqari, standart boshidanoq mavjud bo'lgan asosiy xususiyatlar to'plamini belgilaydi. Ushbu mexanizm uchun mumkin bo'lgan foydalanish doirasini ko'rsatadigan oldindan belgilangan xususiyatlarning ba'zi misollari:

- **Latency** - ma'lumotlar uzatilishining kechikishi;
- **Memory Protocol** - xotiraga kirish huquqi (o'qish va / yoki yozish);
- **Queue Size** - kiritish portni buferi hajmi;
- **Source Code Size** - xotira hajmi.

Xususiyatlar tarkibidagi ma'lumotlardan foydalanish usuli foydalanuvchi tomonidan belgilanadi va simulyatsiya qilingan tizimning individual xususiyatlariga bog'liq. Bu sizga arxitektura loyihalashi uchun juda muhim bo'lgan tizim xususiyatlarining ko'pchiligini maqsadli ravishda o'rganishga imkon beradi, garchi ular asosiy tuzilish to'plamiga kiritilmagan bo'lsa ham. AADL-ning kuchayib borishiga qaraganda kuchliroq nuqtasi har qanday abstraksiya darajasidagi tizimlarni tavsiflash qobiliyatidir. Bu shuni anglatadiki, siz tizim arxitekturasini eng umumiy g'oyadan boshlab va mavhumlik darajasini asta-sekin pasaytirib, tarkibiy qismlarning ichki tartibini aniqlab, qo'shimcha xususiyatlar va cheklolrarni kiritgan holda bosqichma-bosqich qurishingiz mumkin. Ushbu loyihalash uslubi bilan har bir bosqichda biz AADL nuqtai nazaridan ham, qo'llanmada keltirilgan arxitekturaning ta'rifiga muvofiq to'liq arxitekturaga ega bo'lamiz. Bu yoki boshqa ko'rib chiqish darajasi imkon qadar tahlil qilinishi mumkin

va shu bilan mumkin bo'lgan muammolarni aniqlash va loyihalashning dastlabki bosqichlarida yuqori darajadagi eng yaxshi echimlarni tanlash.

## **2.7. O'rnatilgan tizimlarini loyihalash metodologiyasini rivojlantirish istiqbollari**

Mutaxassislarining ta'kidlashicha, hozirgi sharoitda loyihalashning an'anaviy usullaridan foydalanish O'rnatilgan tizimlarni yaratish sohasidagi tizimli inqirozga olib keldi. Buqun kunda elementlar bazasining potentsial imkoniyatlari, elektron va dasturiy ta'minot loyihalashi samaradorligi, tekshirish va sinovning zarur hajmlari, arxitektura loyihalashi o'rtasida tobora katta farq mavjud. Umumiy loyihalash texnologiyalaridan foydalanish ishlab chiqaruvchilarni "qurilish g'ishtlari" darajasini doimiy ravishda oshirishga majbur qiladi, bu esa har tomonlama optimallashtirish va tekshirish vositalari yo'q bo'lganda yakuniy mahsulotning erishilishi mumkin bo'lgan murakkabligi va sifatiga "to'yinganlik" ta'sirini keltirib chiqaradi. Muammo shundaki, mavjud bo'lgan potentsial hajmdagi O'rnatilgan tizimlarni ushbu sohada ishlaydigan jamoalar an'anaviy loyihalash texnologiyalari doirasida, kerakli vaqtda yoki etarli darajada sifatli amalga oshira olmaydilar. Inqiroz rivojlanish sifatiga ta'sir qiladi (bozorda "xom" mahsulotlarning ommaviy ko'rinishi, birinchi navbatda chiplar va dasturiy mahsulotlar uchun), ishlab chiqarilgan O'rnatilgan tizimlarning mavjud murakkabligi cheklangan va rivojlanish elementlarini qayta ishlatish texnologiyasi deyarli qo'llanilmaydi.

Bularning barchasi birinchi navbatda sanoat, energetika, transport, tibbiyot, mudofaa bilan bog'liq mas'uliyatli O'rnatilgan tizimlarning rivojlanish sohasiga taalluqlidir, bu erda past sifatli ishlanmalarni joriy etish mumkin emas, uzoq muddatli foydalanishni ta'minlash talab etiladi. Natijada, ko'p holatlarda, eng so'nggi texnologik yutuqlarni qo'llash taqiqlanadi. An'anaga ko'ra, dasturiy va apparat tarkibiy qismlarining alohida loyihalashi, shuningdek, olingan resurslarga mos kelishini kafolatlash uchun, dasturiy ta'minotga bo'lgan talabni sun'iy ravishda qayta baholashga olib keladi. Yana bir muammo - alohida texnologiyalar yordamida yaratilgan tizimlarning ishonchliligi va sinovlarini baholash qiyinligi. Tizimni dasturiy ta'minot / apparat vositalarini amalga oshirishga va standart komponentlarni tanlashga ajratish vazifasi hozirda rasmiy echimga ega emas va har safar ishlab chiqaruvchi o'z tajribasi va mavjud elektron tarkibiy qismlar asosida qaror qabul qiladi.

Kompyuterlarning kuchi o'smoqda, O'rnatilgan tizimlarlar bozori rivojlanmoqda va tobora qisqa vaqt ichida tobora murakkab tizimlarni talab qilmoqda. Tizim qanchalik sodda bo'lsa, uni rivojlantirish uchun kamroq vaqt ketishi aniq. Shu bilan birga, turli xil mualliflar yaqin kelajakda har qanday O'rnatilgan tizimlarni murakkabligiga qaramay rivojlantirish uchun aniq vaqt ajratilishini taxmin qilishmoqda. Ushbu tomirda noldan rivojlanayotgan tizimlar nafaqat samarasiz, balki muvaffaqiyatsiz bo'ladi. Loyihalar muqarrar ravishda o'z vaqtida uzaytiriladi. Aks holda, yaratilgan tizimlarning sifati zarar ko'radi. Shunday qilib, hisoblash sanoati uchun O'rnatilgan tizimlar uchun sifat jihatidan yangi texnika, texnologiyalar va loyihalash vositalarini yaratish juda muhimdir.

Bugungi kunda O'rnatilgan tizimlarni yaratishda ishlatiladigan turli xil texnik va texnologiyalar loyihalash mavzusini amalga oshirishning murakkabligi va noaniqligini namoyish etadi. Oldingi material tizimning umumiy ko'rinishi rolini kuchaytirish, turli xil abstraktsiyalardan faol foydalanish va loyihalash va amalga oshirish bosqichlarini aniq ajratish bilan bog'liq bo'lgan yuqori darajadagi loyihalash bosqichlari (ko'pincha boshlang'ich bosqichlar deb yuritiladi) sohasiga yo'naltirilgan harakatlanish ahamiyatligi va samaradorligini ishonchli ravishda namoyish etadi. Har xil toifadagi va yo'nalishdagi mutaxassislarining qurolli kuchlarini yaratishda tabiiy ishtirok etish ularning rol ustuvorliklari, bajarilgan ishlarining ahamiyatini baholash, ishlab chiqilgan tizimlarning asosiy modellari, lug'atlar va faoliyatni rasmiylashtirish usullari bilan bir nechta "dominant" mafkuralarning mavjudligini belgilaydi. Odatda, hal qilinayotgan muammoning mohiyati bilan bevosita bog'liq bo'lmagan omillar bilan belgilanadigan O'rnatilgan tizimlar loyihasidagi mavjud vaziyatga qarab, ko'pincha "amaliy" (masalan, avtomatik boshqarish nazariyasi sohasida, jarayonlarni boshqarish tizimlari, kommunikatsiyalar), "apparatchiklar" mutaxassislari etakchi rol o'ynashi mumkin. (raqamli texnologiyalar sohasidagi mutaxassislar) yoki "dasturchilar" (tizim yoki amaliy dasturlash sohasidagi mutaxassislar).

Yuqoridagi misollarning barchasi loyihalashtirish tashkilotida "tarafkashlik" ga olib keladi, bu yaratilayotgan tizim va umuman loyihaning barcha asosiy xususiyatlariga ta'sir qiladi. Informatika va kompyuter texnologiyalari sohasidagi mutaxassisning faoliyat sohasiga tegishli bo'lgan masalalarning murakkabligi va hajmining tez sur'atlarda o'sishi quyidagi vaziyatni keltirib chiqaradigan vaziyatga olib keldi.

Tabiiy umumiy qarashlarga ega bo'lgan integral fan sifatida kompyuter texnologiyalarining dastlabki rivojlanishi o'rnini yaratuvchi uskunalari (o'zining "apparat" arxitekturasi kontseptsiyasi bilan) va dasturiy ta'minotni yaratish (bir tomondan, "sof" matematik abstraksiyalardan kelib chiqadigan, ikkinchi tomondan, unga zaif ta'sir ko'rsatgan holda "apparatshiklar" mahsulotiga moslashishga majbur qilish). Bunday tafovut ushbu sohani yagona tasavvurga ega mutaxassislarining etishmasligini aniqladi. Bunday tizim muhandislarining faoliyat sohasi har qanday yaratilayotgan axborot va hisoblash tizimining axborot infratuzilmasi arxitekturasi.

Ta'kidlash joizki, bugungi kunda O'rnatilgan tizimlar qo'llash sohalari juda ko'p, ularda dasturiy ta'minot sohasidagi mutaxassisi loyihaning mafkurasi sifatida ishlatish maqbuldir. Zamonaviy standart qurilmalar bilan ta'minlangan hisoblash resurslarining zaxirasi va tizim dasturiy ta'minotini ishlab chiqish darajasi ushbu sohalarda amaliy muammolarni maqbul sifat va ishonch bilan muvaffaqiyatli hal qilishga imkon beradi.

Loyiha mafkurasining roli kompyuter uskunalari bo'yicha mutaxassis tomonidan bajarilganda vaziyat boshqacha ko'rinadi. Odatda, tizimdagi hisoblash jarayonini tashkil qilishning ahamiyatsiz past darajadagi qismi uning ko'rish maydoniga kiradi. Bu talab qilinadigan apparat resurslarini ham, loyihani amalga oshirish uchun jami ish haqini ham etarli darajada baholay olmaslikka olib keladi. Juda tez-tez, bunday mutaxassisning asosiy loyihalash paradigmasi quyidagicha rasmlantiriladi: "Asosiysi dasturlashtiriladigan jihozlarni yaratishdir va hamma narsa dasturlashtirilishi mumkin." Bunday loyihalashning ijobiy natijasi ahamiyatsiz "ma'lumot-hisoblash" komponentiga ega bo'lgan eng oddiy tizimlardagina mumkin.

O'rnatilgan tizimlar loyihalashi hududida jiddiy cheklovlar mavjud tizimning texnik resurslarining ko'payishi ishonchlilik ko'rsatkichlari, loyihalarni bajarish muddati va byudjetlar nuqtai nazaridan juda muhimdir. Shunday qilib, tizim arxitekturalari rolini samarali o'ynay oladigan haqiqiy "integral" mutaxassislar - hisoblagichlarga ega bo'lish kerak. Bunday tizim texniklarining rasmlanishi va samarali ishlashi uchun shart-sharoitlar tegishli loyihalash va loyihalash mavzusining "ajralmas" nuqtai nazariga asoslangan O'rnatilgan tizimlar loyihalashi texnologiyalari bo'lishi mumkin.



Zamonaviy kompyuter texnologiyalari amaliy masalalarni hal qilishda hisoblash jarayonini tashkil etishning bir qator asosiy tamoyillarini va har xil darajadagi, sifat va umumiy darajadagi ko'plab texnik echimlarni ifodalaydi, ular O'rnatilgan tizimlar ko'lamini va loyihalash jarayonining barcha qismlarini qamrab oladi.

Mavjud O'rnatilgan tizimlarning xilma-xilligi, ayniqsa O'rnatilgan tizimlar sohasida, cheklangan miqdordagi standart echimlardan foydalanish samarasizligini isbotladi. Hatto asosiy hisoblash arxitekturalarining tasniflash darajasida aniq bo'lgan muammolar aniq ko'rib chiqilgan. Siz bir xil tamoyillarni turli xil nomlarga ega bo'lgan kompyuter texnologiyalarining turli sohalarida texnik echimlarda ko'p jihatdan misollar keltirishingiz mumkin. Kompyuter texnologiyasi sohasidagi mavjud texnik tasniflash, birlashtirish va mavhumlik darajasi qoniqarli emas. Bundan tashqari, fan sifatida kompyuter va kompyuter texnologiyalarining rivojlanishiga (texnik jihatdan) jiddiy zararlar, texnik bo'lmagan sabablarga ko'ra ma'qul keladigan texnik echimlarning mohiyatini (terminologiyadan boshlab) qasddan buzib ko'rsatish va yashirish bilan bog'liq.

Yagona koordinatalar tizimi zarur bo'lib, unda O'rnatilgan tizimlarni ishlab chiquvchilar o'z echimlarini taqdim etishlari, qayta foydalanish elementlarini to'plashlari, amaliy hisoblash jarayonlarini tashkil etishga e'tibor berishlari va buning uchun samarali bajarishni tanlashlari mumkin. O'rnatilgan tizimlar sohasida yagona loyiha makonini yaratishda hisobga olinishi kerak bo'lgan bir qator omillarni ta'kidlash mumkin. Bunday omillar yaratilayotgan tizimning ixtisoslashuvi, "hisoblashsiz" cheklovlarning murakkab tabiati, elementlar bazasining xilma-xilligi (shu jumladan dasturiy texnologiyalarning o'ziga xos xususiyatlarini o'z ichiga olgan) va yangi ishlanmalarga tobora ortib borayotgan talabni o'z ichiga oladi.

### **Nazorat savollari:**

1. O'rnatilgan tizimning arxitekturasi qanday?
2. O'rnatilgan tizim arxitekturasining elementi nima?
3. Arxitektura elementlariga to'rtta misol keltiring.
4. Arxitektura tuzilishi nima?
5. Besh turdagi konstruksiyalarni nomlang va aniqlang.
6. O'rnatilgan tizimlarni ishlab chiqish uchun minimal uchta maqsad nima?
7. Arxitektura ushbu muammolarni qanday hal qilishi mumkin?

8. Oʻrnatilgan tizim modeli nima?
9. Oʻrnatilgan tizimlar modelida qanday tarkibiy yondashuv qoʻllaniladi?
10. Oʻrnatilgan tizimlar modelning qatlamlarini chizing va aniqlang.
11. Modulli arxitektura koʻrinishi nima uchun foydali?
12. Oʻrnatilgan tizimlarning loyihalashi haqida maʼlumot toʻplash uchun qanday oltita manbalar mavjud?
13. Oʻrnatilgan tizimlarning loyihalash siklini tavsiflash uchun qanday modellar mavjud?
14. Yangi tizimni loyihalashda uchraydigan asosiy muammolarni sanab bering.
15. Oʻrnatilgan tizmlarni ishlab chiquvchilar duch keladigan muammolarni sanab bering.
16. Loyihalash samaradorligini oshirishning asosiy yoʻnalishlari.
17. Tizimning apparat-dasturiy taʼminotida "choʻktirish chuqurligi" shkalasi boʻyicha Oʻrnatilgan tizimlar uchun loyihalash variantlari qanday?
18. Oʻrnatilgan tizmlar loyihalash jarayonlari va vositalari rivojlanishining asosiy tendentsiyalari nimada?
19. Yuqori darajadagi Oʻrnatilgan tizmlarni loyihalash texnikasining asosiy yoʻnalishlarini sanab bering.
20. Oʻrnatilgan tizmlarni loyihalashning odatiy senariylari.
21. Oʻrnatilgan tizmlarni loyihalashtirishning samarali texnologiyalarini yaratishga qaratilgan tadqiqotlarning asosiy yoʻnalishlarini sanab bering.
22. Platformaga yoʻnaltirilgan loyihalash tushunchasining afzalliklari va kamchiliklari.
23. Oʻrnatilgan tizimlarning arxitekturasini tavsiflash uchun tillarni sanab bering.
24. Oʻrnatilgan tizmlarni loyihalash rivojlanish tendentsiyalarini sanab bering.

## **3-BOB. O‘RNATILGAN TIZIMLARNING ARXITEKTURAVIY LOYIHASH**

### **3.1. O‘rnatilgan tizimlarni loyihalash platformalari**

Kompyuter texnologiyasida intuitiv darajada platforma tushunchasi so‘nggi 10 yil ichida juda keng qo‘llanildi. Masalan, O‘rnatilgan tizim loyihalash va «Platform-Based Design» (PBD, Platformaga asoslangan loyihaning) etakchi yo‘nalishlaridan biri bo‘lgan hisoblash platformasi ba’zi umumiy shartlarni qondiradigan loyihalar to‘plami sifatida belgilangan..

Hisoblash platformasining eng muhim xususiyati, spetsifikatsiyalar to‘plamidan qayta foydalanish uchun belgilanganidek - amalga oshirish spetsifikatsiyalaridan ma‘lum bir mavhumlikni ta‘minlash qobiliyatidir. Xulosa tizimida hisoblash platformasi (HP) o‘rnatilgan tizim tarkibidagi funksional jihatdan tugalangan va funksional jihatdan ahamiyatli ob'ektning "tashqi" va "ichki" ko‘rinishlarining birligi sifatida qaraladi. O‘rnatilgan tizimlar platformasi doirasida- platforma qayta foydalanish uchun loyihada qayd etilgan texnik echimdir. HP - bu arxitekturiy darajadagi asosiy qayta foydalanish vositasi.

Bugungi amaliyotda, afsuski, "ikkinchi darajali" tanlangan, tayyor hisoblash platformalarini (apparat, dasturiy ta‘minot, instrumental, konstruktiv) asosida loyihalashdan iborat. Qo‘llanmada ko‘rib chiqilgan loyihalash metodologiyasi

sizga vakolatli tanlovni kuchaytirish va qo‘shish yoki loyiha uchun hisoblash maydonchalarini yaratish imkonini beradi.

Aspekt modeli doirasida arxitekturiy loyihalash bosqichining markaziy tushunchasi arxitekturiy platformadir. Ushbu kontseptsiya maqsadli tizimning arxitektura modelini yaratish, spetsifikatsiya qilish va tekshirishni amalga oshirishga, shuningdek, amalga oshirish bosqichi uchun spetsifikatsiyalarni shakllantirish. Shunday qilib, arxitekturiy platformani quyidagi loyiha elementlarining kombinatsiyasi sifatida ko‘rib chiqish mumkin :

- loyiha jarayonining aspekt maydoni ( loyiha jihatlari ro‘yxati );
- hisob-kitoblar modeli (modellari);
- individual jihatlar (loyiha mezonlari) o‘rtasidagi maqbul munosabatlarni belgilaydigan tashqi omillar ;
- qayta foydalanish uchun belgilangan namunalar ro‘yxati;
- elementlar bazasi (keng talqinda).

Aslida, arxitektura platformasi loyihalashtirish bosqichida ham, ko'p jihatdan amalga oshirish bosqichida ham loyiha faoliyatini boshqaradigan global echimlar, cheklovlar va ustuvorliklar to'plami sifatida ishlaydi. Loyihaning kontseptual qarorlarini aks ettiradi. O'rnatilgan tizimlarni loyihalash usullari va vositalarini rivojlantirishning hozirgi bosqichining eng muhim vazifasi loyihalash jarayonining dastlabki (yuqori darajadagi) bosqichlarida olingan natijalarni qayta ishlatish darajasini oshirishdir. Arxitektura platformasi kontseptsiyasini O'rnatilgan tizimlarlar sohasida kontseptual echimlarni aspekt loyihasi modeli doirasida qayta ishlatish vositasi sifatida foydalanish taklif etiladi.

Platformaning kontseptsiyasi loyihaning ushbu sohasidagi juda qulay tasniflash parametriga o'xshaydi. Loyiha platformasi kontseptsiyasi asosida (O'rnatilgan tizimlarning yaxlit yoki xususiy loyihasi uchun asos sifatida qabul qilingan hisoblash yoki boshqa platforma) O'rnatilgan tizimlarni tasniflash daraxtini qurish mumkin, bu tizimning umumiy xususiyatlariga yoki uning tarkibiy qismlariga, O'rnatilgan tizimlarning xususiyatlari, texnologiyalari va amalga oshirilishining butun maydoniga e'tibor berishga imkon beradi. loyiha platformasi asosida tasniflashga misol quyida keltirilgan:

- Hisoblash platformalari (protessorlar, OS, MC va boshqalar);
- Sanoat kompyuterlari va PLC platformalari;
- Multimedia sanoatining yarim tayyor mahsulotlari;
- Mikrokontrollerlar va DSP;
- FPGA ;
- ASIC, ASSP, ASIP, SOC, NOC;
- " Erkin" silikon kompilyatsiyasi;
- Tizimli dasturiy ta'minot (DT) platformalari, tarmoq, interfeys, tarkibiy, instrumental va boshqalar.

### **3.2. O'rnatilgan tizimlar arxitekturasini loyihalash**

#### **Arxitektura platformasi va arxitekturani loyihalash mezonlari**

Arxitekturani loyihalash doirasida savollar chiqadi: tizimning element bazasi qanday yaratilganligi, ishlab chiqaruvchi loyihaning etakchi tomonlarini qanday tanlashi va loyihaning tashqi cheklovlarini qaerdan oladi ( loyiha mezonlari ). Shubhasiz, har bir loyihani noldan rivojlantirish juda samarasiz va deyarli imkonsiz bo'ladi. Turli ishlab chiquvchilar ba'zi tayyor elementlar, tarkibiy qismlar, echimlar va

boshqalardan foydalanadilar. Ular qaerdan paydo bo'ldi? Ushbu savollarga javob platforma yoki arxitekturiy platforma tushunchasi. Arxitektura platformasini (A- platforma) boshqa funktsional plaginni yaratish uchun o'rnatilgan virtual mashinaning ba'zi bir umumlashmasi sifatida ko'rib chiqishga harakat qilishingiz mumkin. Bu holatda, ko'p mavjud platformalarini, o'tkazib yuborilgan Biroq, bu umuman muvaffaqiyatli emas, chunki bu holda funktsional imkoniyatlardan ko'proq narsani aniqlaydigan ko'plab mavjud platformalar o'tkazib yuborilgan. Yuqorida ta'kidlab o'tilganidek, arxitektura platformasi loyiha jarayonining quyidagi elementlarining birikmasidir:

- loyiha jarayonining aspekt maydoni ( loyiha jihatlar ro'yxati );
- xisob-kitoblar modeli (modellari);
- individual jihatlar (loyiha mezonlari) o'rtasidagi maqbul munosabatlarni belgilaydigan tashqi omillar ;
- qayta foydalanish uchun belgilangan namunalar ro'yxati;
- elementar bazasi.

Umumiy holda, ma'lum bir **A- platformani** tanlashda, ishlab chiqaruvchi darhol tanlangan platforma talab qiladigan doirada bo'ladi. **A-platforma** loyiha jarayonining aspektlari tarkibi va o'zaro muhimligini belgilaydi. U elementar bazaning ma'lum xususiyatlarini aytib beradi va shu bilan individual jihatlar o'rtasidagi yo'l qo'yiladigan munosabatlarni aniqlaydi. A-platforma, xususan, amalga oshiriladigan yoki virtual A-model bo'lib, qayta foydalanish usullarini amalga oshirishning ba'zi usullarini taklif qiladi yoki nazarda tutadi. A-platforma uni amalga oshirish jarayonida "amalga oshiradigan" bir yoki bir nechta MC ni belgilaydi. Ushbu parametrlarning nisbati ishlab chiquvchiga ma'lum bir muammoni hal qilish uchun ma'lum A-platformalarni tanlash imkoniyatini beradi.

Agar biz loyihaning xulq-atvor jihatlarini ko'rib chiqishga qaytsak, bu erda A-platforma MCni belgilab (o'rnatib) juda katta rol o'ynaydi. Xulq-atvor jihatlarini O'rnatilgan tizimlarni loyihalashda juda muhimdir, ammo u yagona variantdan uzoqda, shuning uchun tanlangan A platformasi deyarli har jihatdan ishlab chiqaruvchini qondiradigan holatlar bo'lishi mumkin, ammo u taqdim etgan MC muammoni hal qilish uchun mutlaqo mos emas. A-platforma uchun bunday holatlarning tarqalishi sababli konfiguratsiya kabi xususiyat aniqlanadi.

A-platformasining qayta tuzilishi MoSni amalga oshirishda "amalga oshirilgan" o'zgarishlarni o'zgartirish qobiliyati sifatida tavsiflanadi. Albatta, bunday xususiyatlarning mavjudligi "ishlab

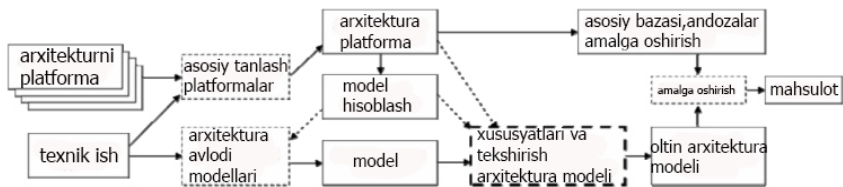
chiqaruvchining hayotini sezilarli darajada murakkablashtiradi", chunki A-platfomasi bilan ishlashda qo‘shimcha harakatlar talab etiladi, lekin ko‘pincha aniq foyda keltiradi. Umuman olganda, qayta konfiguratsiya boshqa xarakterga ega bo‘lishi mumkin (yoki amalga oshirish usuli):

- loyihalash bosqichida, ba’zi uslublar konfiguratsiya qilinishi kerak bo‘lgan hollarda ;

- amalga oshirish bosqichida, muayyan kompilyatsiya parametrlari o‘rnatilganda yoki element bazasining turli qismlaridan foydalanilganda;

- ijro etilish bosqichida, tizim funktsionallikka moslashuvchan moslashishga imkon berganda.

A-platfomaning qayta konfiguratsiyasi elementlar bazasining turli xil elektron tarkibiy qismlarini kiritish orqali ta‘minlanishi mumkin, funktsional xususiyatlari va fizik parametrlari jihatidan har xil, ammo xulosalarda bir-biriga mos keladigan, yuqori darajadagi tillardan foydalanish va amalga oshirish shablonlarini tavsiflash uchun shartli kompilyatsiya, platformaga dasturlashtiriladigan mantiqiy elementlarni kiritish va boshqalar. Ba’zi hollarda, A-platfomada MCni o‘zgartirish ichki qobiliyati bo‘lmaganida, ishlab chiqaruvchining o‘zi bunday o‘zgarishlarni amalga oshirishi kerak. Odatda, buning uchun tanlangan A-platfomani ba’zi bir amalga oshirish amalga oshiriladi, shundan so‘ng u bajarilgan vazifaga mos keladigan "yuqoridan" kerakli MC ning "mujassamlanishi" darajasigacha bajariladi. Keyingi loyihalash davomida yaratilgan ustki tuzilma asosiy A-platfomasi bilan birgalikda tanlov mezonlariga javob beradigan yangi platforma sifatida qaraladi. A-platfoma MC-ni o‘zgartirish yoki sozlash maqsadida yaratilgan A-platfoma qo‘shimchasi ish muhiti deb ataladi. Shuni alohida ta’kidlash kerakki, A platfomasining MC-ni nafaqat o‘ziga xos xususiyatlarga ko‘ra tanlangan vazifaga mos kelmasligi tufayli, balki A platfomasining o‘zi elementlaridagi xatolar tufayli ham o‘zgartirish kerak bo‘lishi mumkin. Shunday qilib, katta integral sxemalarni va asosiy hisoblash platfomasiga kiritilgan dasturiy ta‘minotning barcha aniqlangan xatolarini tuzatish uchun operatsion muhiti mo‘ljallangan. Aspekt loyihalash doirasida muayyan hisoblash modelini tanlashning umumiy mezonni sifatida tanlangan hisoblash modelidagi muammoni hal qilishning murakkabligi arxitekturiy platfomaning ish sharoitini amalga oshirish murakkabligiga nisbati deb hisoblanadi. A-platfomadan foydalangan holda O‘rnatilgan tizimlarlarni loyihalash jarayonining umumlashtirilgan sxemasi rasm. 3.1.



Rasm 3.1. Arxitektura platformasining Oʻrnatilgan tizimlarni loyihalash jarayonida tutgan oʻrni va ahamiyati

Rivojlanish tajribasiga asoslanib, ekspert fikridan foydalangan holda, dastlabki A-platforma loyihaning dastlabki bosqichlarida tanlanadi. Kelajakda tanlangan A platformasi maqsadli tizimni loyihalashning deyarli barcha bosqichlarida hal qiluvchi rol oʻynaydi. To -ning talablarini hisobga olgan holda A-platformasi tomonidan amalga oshirilgan MC asosida, ishlab chiqaruvchi tizimning A-modelini yaratadi. A platformasi tomonidan belgilangan loyiha mezonlari modelni aniqlashtirish uchun ishlatiladi, natijada tizimning "oltin" modeli Rasmlanadi. "Oltin" modelni amalga oshirish elementlar bazasi va A-platformasi tomonidan taqdim etilgan qayta foydalanish shablonlariga asoslanadi. Yuqorida aytilganlardan xulosa qilish mumkinki, puxta ishlab chiqilgan A-platforma Oʻrnatilgan tizimlarni loyihalash jarayonida kontseptual echimlarni qayta ishlatishda kuchli vosita hisoblanadi.

Arxitektura loyahasining bir qismi sifatida, qabul qilingan qarorlar sifatini va umuman loyiha samaradorligini rasmiy ravishda baholashga imkon beradigan mezonlarni Rasmlantirish kerak. Arxitektura loyihasi uchun samaradorlik mezoni yakuniy mahsulotning narxi, rivojlanish davri, ishlab chiqish guruhining bir xilligi, loyihalashtirilayotgan tizimning sifati va ishonchliligi, ishlanmalardan foydalanish darajasi va boshqalar boʻlishi mumkin. Belgilangan shartlar tizimning aniq "hisoblash" xususiyatlarini tizimning qolgan parametrlari bilan optimallashtirish talabi boʻlishi mumkin. Har bir muayyan vaziyatda (har bir aniq loyiha uchun) ma'lum bir mezon (mezonlar tizimi) etakchi rol oʻynaydi. Umumiy holda, oʻrnatilgan tizimlarni loyihalash bilan bogʻliq muammolarni hal qilishda multitriterial optimallashtirishni amalga oshirish kerak.

## O‘rnatilgan tizimlar loyihalash mezonlari

Multitritriy optimallashtirish nazariyasidan quyidagilar keladi :

• agar qabul qilinadigan parametrlarning qiymatlari to‘plami yopiq bo‘lsa va sifat ko‘rsatkichlari parametrlarning doimiy funksiyalari bo‘lsa samarali echimlar mavjud ;

• echim samarali bo‘ladi, agar u bilan taqqoslanadiganlar orasida undan yaxshiroq echim topilmasa;

• ikkita echim taqqoslanishi mumkin, agar bitta echim boshqasidan yomon bo‘lmasa (yoki yaxshiroq) ;

• taqqoslanadigan ikkita echimdan, shubhasiz, kamida bitta mezonga ega bo‘lgan echim ikkinchisining tegishli mezoniga qaraganda ancha yaxshi bo‘lsa;

• alternativ mezonlar orasidagi samarali echimlar uchun faqat bitta mezonning yaxshilanishi, hech bo‘lmaganda boshqasining yomonlashuviga olib keladigan munosabatlar mavjud;

• bunday o‘zaro bog‘liqlikni olish uchun boshqalarning sobit qiymatlari uchun har qanday mezonning yakuniy yaxshilanishini aniqlash zarur, bu esa, o‘z navbatida, belgilangan chegaralarda o‘zgarib turadi;

• qaysi mezon boshqalarning ma‘lum bir qiymati uchun yaxshilangan mezon sifatida tanlangan bo‘lsa, optimallashtirish natijasi mezonlar o‘rtasidagi o‘zaro bog‘liqlik bo‘ladi.

Matematik jihatdan O‘rnatilgan tizimlarning optimal loyihasi muammosi arxitektura loyihasi jarayonining maqbullik mezonini (ob‘ektiv funksiyasi) sifatida ko‘rib chiqiladigan ma‘lum bir funktsional shartli ekstremalni (maksimal yoki minimal) topish demakdir. Yuqorida aytib o‘tilganidek, loyiha mezonlari A-platforma tomonidan belgilanadi va ularning o‘ziga xos ko‘rinishi loyihadan loyihaga qarab farq qilishi mumkin. Arxitektura loyihasi mezonining umumiy ko‘rinishi quyidagi formulada keltirilgan:

$$C = \{c : \bigcup_i f_i \rightarrow R, N, Z\},$$

qaerda C - arxitekturiy loyihalarni bir necha mezonlari, c - xususiy mezon, i - loyiha jihatlari soni, fi - tegishli xarakterli vazifalarga majmui jihatini, va R, N va Z real, tabiiy va butun sonlarning majmuasi o‘z navbatida,.



Mezonlarning koʻpligi va loyihash jarayoni shartlariga bogʻliqligi sababli aspekt loyihalash uchun mutlaqo barcha mezonlarni sanab oʻtish mumkin emas. Bundan tashqari, loyiha jarayonining aspektli modeli hozirda turli xil loyiha vazifalari va maqsadlari uchun belgilangan loyiha namunasini taklif qilish uchun etarli darajada rasmiylashtirilmagan. Shu munosabat bilan, aniq funktsiyalarni belgilashning oʻrniga, loyiha mezonlarining eng keng tarqalgan turlarini sanab oʻtish mumkin.

### **3.3. Oʻrnatilgan tizimlarning arxitekturiy modellarini amalga oshirish**

Ommaviy ishlab chiqaruvchi ongida oʻrnatilgan tizimlarni loyihalash kabi ustuvorliklar va diqqatni oʻzgartirish kerak, masalan, loyihalash bosqichlarining ahamiyati va resurslarning zichligini, kanonik arxitekturalarning toʻplamini, ishonchlilik va xavfsizlik kontseptsiyasini, apparat va dasturiy komponentlarning nisbati, hisoblash platformasining qatʼiy darajasi. Birlashtirilgan hisoblash mexanizmlari bazasida oʻrnatilgan tizimlarni loyihalashning muhim sharti sifatida ishlab chiquvchilar amalda foydalanadigan cheklangan nomenklaturasini koʻrib chiqish kerak.

Istiqbolli metodologiyaning markaziy nuqtasi bu dasturiy taʼminot, dasturiy taʼminot yoki amalga oshirishning aralash usuliga nisbatan oʻzgarmas boʻlgan loyiha ishlarining maksimal miqdorini bajarish talabidir. Bu oʻrnatilgan tizimining umumlashtirilgan modelining mavjudligini anglatadi, bu turli xil amalga oshirish usullarining bir-birlariga taqqoslanishini yoʻq qiladi va mavhum maqsadli tizimni minimal murakkablikdagi aniq hisoblash platformalari toʻplamiga xaritalash imkonini beradi. Koʻrsatilgan arxitekturiy model raqamli Oʻrnatilgan tizimlarni loyihalash jarayonida tub oʻzgarishlar tufayli har tomonlama muhokama qilishni talab qiladigan bir qator tushunchalarga asoslanadi. Bular asosan quyidagilardan iborat:

- hisoblash qurilmasini apparat va dasturiy taʼminotda tatbiq etish;
- protsessor va uni tashkil etish variantlari;
- hisoblash platformasi;
- boshqariladigan va dasturlashtiriladigan xususiyatlar;
- virtual mashina;
- hisoblash modeli.

Bir qarashda tushunarli va ravshan bo'lgan "apparat" (hardware) va "dastur" (software) atamaları yanada aniqroq va ehtimol noan'anaviy ta'rifga muhtoj.

Loyihalash, amalga oshirish, ishlash bosqichlarida dasturlash printsiptidan foydalangan holda jarayon yoki mahsulotga kiritilgan sifatni anglash ham muhimdir. Apparat va dasturiy ta'minot terminlarini kengroq talqin qilish mumkin - mos ravishda O'rnatilgan tizimlarning o'zgarmas va o'zgaruvchan qismlari. Ammo bu holda ham, terminologik noqulayliklar namoyon bo'ladi: jarayon va o'zgarish ehtimoli nimani anglatadi? Kamroq tarqalgan atamalar dasturiy ta'minoti O'rnatilgan tizimlarning o'zgaruvchan qismini anglatadi, ammo u qanday amalga oshirilishini aniqlamaydi. Apparat va dasturiy atamalardan keng ma'noda foydalanish arxitektura loyihasining ko'rib chiqilgan metodologiyasi doirasida qulaydir.

Ko'rib chiqilayotgan metodologiya nuqtai nazaridan eng muhim narsa O'rnatilgan tizimlarning muayyan qismini amalga oshirish usuli bo'yicha tasniflash muammosidir. Va birinchi navbatda, bu protsessorni aniqlashda namoyon bo'ladi. Tuzilish usuliga ko'ra, hisoblagich, masalan, tartibsiz mantiqning "chalkashligi" yoki operatsion va boshqarish qismi bo'lgan tuzilgan qurilma sifatida amalga oshirilishi mumkin. O'z navbatida, boshqaruv qismi o'zgaruvchan darajada tuzilishi mumkin. Shuni ta'kidlash kerakki, yuqorida ko'rsatilgan amaliyotlar uchun kirish tilining darajasi har xil bo'lishi mumkin va variantlarning "vilkasi" ahamiyatli bo'lib chiqishi mumkin. O'rnatilgan tizimlarni loyihalashda tizimlarning algoritmik tashkil etilishini muhim rol o'ynaydi. Ishlab chiquvchi tomonidan tanlangan yoki elementlar bazasi tomonidan belgilab qo'yilgan algoritmnining aniq talqini loyiha usullari va vositalari va tizimning asosiy xususiyatlarini belgilaydi. Quyidagi talqinlar haqida gaplashishimiz mumkin:

- parallel ravishda apparat talqini (apparat ishlashi);
- dasturiy mikroprotessor sharhlash (o'rnatilgan protsessor

RISC / CISC guruhlari, shu jumladan dasturiy ta'minotni amalga oshirish, VLIW va EPIC arxitekturalari);

- virtual dasturiy ta'minot bilan ishlaydigan mashinani sharhlash (operatsion tizim, ajratilgan tarmoq aloqa tizimi, ma'lumotlar bazasini boshqarish tizimi, til mashinasi va boshqalar).

Amalga oshirishni tasniflash xususiyati sifatida buyruqning rasmiy ta'rifidan qat'iy belgilangan formatga ega boshqaruv tuzilmasi sifatida foydalanish taklif etiladi. Keyin, apparatni amalga oshirish orqali biz

"format" boshqaruv tuzilmalari - mikrokomandalar, buyruqlar mavjud bo'lganda, algoritmning bevosita bajarilishini tushunishimiz mumkin. Dasturni amalga oshirish deganda "formatli" boshqaruv tuzilmalari - ibtidoiyalar va ularning ketma-ketligi - dasturlar mavjud bo'lganda ko'p darajali talqin tushunilishi mumkin.

Tizimning apparat va dasturiy tarkibiy qismlariga taqdim etilgan qism arxitektura modelidan O'rnatilgan tizimlarni jismoniy amalga oshirishga o'tish usullari va vositalari uchun samarali asos yaratishga imkon beradi.

### **3.4. O'rnatilgan tizimlarni arxitekturaviy loyihalashda modellashtirishning roli**

Ko'rinib turibdiki, o'rnatilgan tizimlarni loyihalashga yuqorida keltirilgan yondashuv turli darajadagi abstraktsiya va yo'naltirilganlik modellaridan keng foydalanishga asoslangan. Aslida, loyiha jarayonining o'zi tashqi cheklolarga duch kelgan ko'plab modellarning boshqariladigan jarayonlar evolyutsiyasiga aylanadi. loyiha jarayonida tizimni modellashtirish tekshirish bilan chambarchas bog'liq bo'lgan ancha murakkab va xilma-xil jarayondir. Loyihalashtirilgan O'rnatilgan tizimlarni turli xil abstraktsiya darajalarida taqlid qilish va turli xil vositalar va modellashtirish tillaridan foydalanish mumkin. O'rnatilgan tizimlarni loyihalash jarayonida dasturiy ta'minot tizimlari yoki ma'lumotlar bazasi dasturlari loyihasidan ma'lum farqlar mavjud. Ushbu farqlar asosan O'rnatilgan tizimlarni loyihalashda har doim tizimning apparat tarkibiy qismlarini ishlab chiqish zarurligiga asoslanadi. Ushbu tarkibiy qismlarni ishlab chiqishning murakkabligi va mas'uliyati, ko'p hollarda, O'rnatilgan tizimlarning dasturiy komponentining murakkabligidan ko'proq emas.

O'rnatilgan tizimlarni loyihalashda modellashtirish quyidagi muammolarni echishga mo'ljallangan : tekshirish (maqсадli va ekvivalent), virtual prototiplash, tizim tarkibiy qismlarini ( apparat va dasturiy ta'minot ) amalga oshirish, ishlab chiqilgan tarkibiy qismlarning spetsifikatsiyasi va hujjatlari. Hozirgi vaqtda O'rnatilgan tizimlar modelini tavsiflashning bir necha usullari va darajalari mavjud. Muvaffaqiyatli rivojlanish uchun har xil mavhumlik va tafsilotlarda tizim modellariga ega bo'lish kerak. Modellar tizimni tizimning yaxlit yoki individual hisoblash tuzilmalari sifatida tavsiflashlari mumkin.

## **3.5. O‘rnatilgan platforma arxitekturasi**

### **3.5.1. O‘rnatilgan platforma xususiyatlari**

O‘rnatilgan barcha tizimlarning muhim xususiyati shundaki, ular ma‘lum bir vazifa yoki funktsiyani bajarish uchun yaratilgan. O‘rnatilgan tizimlar oddiy alohida avtoulodvan tortib, ko‘plab alohida tarmoq integratsiyalashgan elementlaridan tashkil topgan tizim uchun tarmoq kartalariga qadar bo‘lishi mumkin. Ularning barchasi umumiy tizim maqsadlariga erishish uchun birgalikda harakat qilishadi. Bunday tizimlar o‘rnatilgandan so‘ng asosan avtonom ishlaydi. Shassi tizimi telekommunikatsiya uskunalari va ma‘lumotlar uzatish uskunalarning umumiy konfiguratsiyasidir.

Markaziy ishlov berish birligi (protssessor; shuningdek, markaziy ishlov berish birligi - CPU; inglizcha markaziy protssessor birligi, protssessor, so‘zma-so‘z - markaziy ishlov berish birligi) - bu mashina ko‘rsatmalarini (dastur kodi) bajaradigan elektron blok yoki o‘rnatilgan protssessor (dastur kodi), kompyuter texnikasining asosiy qismi. yoki dasturlashtiriladigan mantiq tekshiruvi. Ba‘zan mikroprocessor yoki shunchaki protssessor deb ataladi.

An‘anaviy shaxsiy kompyuterlar ko‘p yillar davomida 32 bitli protssessorlardan foydalangan bo‘lsa-da, yaqinda 64-bit standart ish stoli platformasi bo‘lib, ko‘plab oddiy boshqarish dasturlari uchun juda ko‘p sonli 8-bitli va 16-bitli protssessorlar mavjud. Masalan, Atmel kompaniyasi Atmel AVR mikrokontrollerlari deb nomlanuvchi 8 bitli kontrollerlar uchun mahsulotlar ishlab chiqaradi. Ushbu mikrokontrollerlari 20-30 MGts diapazonda ishlaydi va o‘rnatilgan faqat doimiy xotirasi, tasodifiy operativ xotirasi va periferik qurilmalar bilan jihozlangan. 16 bitli PI C mikrokontrollerlari muhitida mikrokontrollerlar eng ko‘p qo‘llaniladi. Platforma juda past kuchga ega va ko‘pincha uzoq vaqt batareyada ishlashi mumkin. 8-bit controller taqdirda, ip va Ethernet orqali ham ulanish imkoniyatlari va ma‘lumotlarni qayta ishlash uchun qo‘shimcha qurilmalar talab qiladi. Ko‘pgina hollarda, ushbu qurilmalar hech qanday operatsion tizimdan foydalanmaydi, garchi freertos kabi oddiy real vaqtda operatsion tizimlar (rtos) mavjud.

8-bitli va 16-bitli qurilmalar bir muncha vaqt foydalanishda qolsa ham, 32 bitli qurilmalarga katta ehtiyoj mavjudligi aniq. Ular ko‘p hollarda 8/16-bitli mikrokontrollarni taklif qilishadi, shuningdek

ularning mahsulot qatorining 32-bitli mikrokontroller variantlarini taqdim etishadi. Yuqorida aytib oʻtilgan qurilmalar odatda mikrokontroller sifatida tasniflanadi. Odatda murakkab RTOS yoki toʻliq Linux distributivi kabi operatsion tizimga bogʻliq boʻlgan murakkab ulanish va grafikali yuqori unumli qurilmalar mavjud. Oʻrnatilgan tizimlarga boʻlgan taxminlar eksponent sifatida oʻsib borar ekan, dasturiy taʼminotning ushbu taxminlarga mos kelishini taʼminlash qobiliyati juda muhimdir. Bunday hollarda, Linux kabi toʻliq xususiyatli operatsion tizimlar katta imkoniyatlarni taqdim etadi. Ushbu yuqori sifatli qurilmalar odatda mikroprotsessor deb ataladi, ular mikrokontrollerlardan farqli oʻlaroq, ular 32-bit protsessor arxitekturasiga egadir.

Baʼzi hollarda protsessor tizim uchun nazorat funktsiyasini taʼminlaydi, boshqa holatlarda, u dasturning butun ish yukini boshqaradi. Turli xil dasturlarni hisobga olgan holda, protsessorning tezligi muhim xususiyatdir, Oʻrnatilgan mikroprotsessor tizimining diapazoni bir necha yuz megagerts (200 MGts) dan 1 Gigagertsgacha. Takt chastotasi muhim xususiyatdir, chunki bu ishlashning birinchi koʻrsatkichidir, ammo, umumiy protsessor arxitekturasi, shu jumladan kesh (va hajm), tizimning umumiy ishlashiga hissa qoʻshadigan muhim jihatlardir. Aslida, protsessor qanchalik tez boʻlsa, shunchalik tezroq xotira saqlashi (ishlatishi) kerak.

### **Integratsiya darajasi.**

Arzonroq narxlar, yuqori zichlik va kichik platformalarga boʻlgan talab oʻsib borishi, oʻrnatilgan platformani tashkil etuvchi har bir moslama uchun integratsiyalashuv darajasini oshirishni talab qildi. Dastlab, oʻrnatilgan platformalar alohida diskret qismlardan iborat edi. Protsessor xotira shinasining interfeysi bilan alohida tarkibiy qism edi va barcha qismlar ushbu shinaga ulangan edi. Integratsiya darajasi oshgani sayin protsessor birligiga koʻproq mantiq qoʻshilib, muayyan dasturlar uchun xizmat protsessorlari oilalarini yaratadi. Chipdagi tizim (SOC) atamasi koʻpincha yuqori darajada integratsiyalashgan protsessorlarni tavsiflash uchun ishlatiladi. Ushbu SOClar bir qator aniq dasturlar va maqsadlar uchun zarur boʻlgan mantiq va interfeyslarning aksariyatini oʻz ichiga oladi. Ushbu SOC qurilmalarini ishlab chiqaradigan kremniy ishlab chiqaruvchilari koʻpincha bir xil protsessor yadrosidan foydalangan holda SOC oilalarini yaratadilar, ammo keng qamrovli integratsiya imkoniyatlariga ega.

## **Quvvat sarfi.**

Qurilmalar tomonidan sarflanadigan quvvat har xil usullar bilan o'lanadi. O'rnatilgan qurilmalar uchun odatiy quvvat, odatda qurilmaning quvvat sarfini ta'minlaydi. Bu dasturni I / O qobiliyatlarining bir qismini bajaradigan protsessorida ishga tushirish bilan o'lanadi. Ko'pgina hollarda, aslida bir nechta turli xil kuchlanish manbalari qo'llaniladi, shuning uchun bu barcha quvvat manbalarida umumlashtirilishi kerak.

Aksariyat sotuvchilar asboblarning quvvat diagrammasini (TDP) taqdim etadi. TDP - bu sovutish tizimini tarqatish uchun zarur bo'lgan maksimal quvvat. Bu tizim faol sovutishga muhtoj degani emas. Ko'plab o'rnatilgan tizimlarda issiqlik tarqalishi hali ham zarur. TDP umumiy issiqlik loyhasining bir qismi sifatida ishlatiladi, bu CPU / SOCning qizib ketmasligini ta'minlashi kerak. Quvvat ko'rsatkichlari ko'pincha tizimning faoliyat darajasiga bog'liq. Ko'p protsessor tizimlarida juda kam quvvat sarfi mavjud. Tizimning kuchi, shuningdek, tizimning barcha tarkibiy qismlari tomonidan talab qilinadigan asosiy yagona quvvat manbaidan quvvat transformatsiyada yo'qotishlarni o'z ichiga olishi kerak; 60-80% ni samarali transformatsiya qilish normaldir.

O'rnatilgan tizimlarning o'lchamlari xilma-xil bo'lib, o'rnatilgan foydalanish holatlarida bo'lgani kabi. O'rnatilgan ko'plab tizimlar bitta bosilgan elektron platadan iborat va ko'pincha bitta platali kompyuterlar (SBC) deb nomlanadi. Platforma bitta kirish voltajini (12 V kabi) quvvat bilan ta'minlaydi va SATA/SDIO ommaviy saqlash, USB va displey kabi qurilmalar uchun ulagichlarni ta'minlaydi. Ushbu ulagichlar standart kompyuter platformalari kabi platformalarda bo'lishi shart emas (bu ayniqsa displey uchun juda muhim). Bitta platali platformasi maqsadli foydalanish uchun o'rnatilgan platformani yaratishning eng tejimli usuli hisoblanadi, ammo vaqt o'tishi bilan platformani yangilab bo'lmaydi.

### **3.5.2. O'rnatilgan tizimlarning element bazasi**

#### **Protsessor (CPU)**

**Protsessor** - bu hisoblash tizimining elementi, xotiradan buyruqlar olish va buyruqlarda ko'rsatilgan harakatlarni bajarish uchun asbob, ma'lumotlarni qayta ishlaydigan qurilma. Ba'zi hollarda protsessor ma'lumotni qayta ishlashga mo'ljallangan dasturlar deb ham ataladi (masalan, so'z protsessorlari, til protsessorlari).

Protsessorlar (qurilmalar ma'nosida) turli xil mezonlarga ko'ra tasniflanishi mumkin, masalan, ularni tashkil qilish (konveyer, matritsali), ishlov berilayotgan ma'lumotlarning tabiati, ularning maqsadi va boshqalar. Protsessor adres maydonlarining hajmi va soni, ichki va tashqi ma'lumotlar shinalarining kengligi, buyruqlar tizimi, uzilishlarni qayta ishlash usuli, tashqi qurilmalarni ulash uchun manzil tanlagichining mavjudligi, xotiraga bevosita kirish mavjudligi va quvvatni boshqarish tizimi bilan tavsiflanadi. Yuqori samarali yadrolarda, konveyer, kesh xotirasi, uzilishlarni bashorat qilish moslamasi va bir nechta ALU mavjudligi kabi mexanizmlar mavjud bo'lishi mumkin. Protsessorning dasturiyligi ixtiyoriydir.

Protsessorlar quyidagilar bo'lishi mumkin:

1. Dasturlashtirilmaydigan, dasturiy ta'minotsiz ;
2. Dasturlashtiriladigan va dasturiy ta'minotlangan ;
3. Dasturlashtirilmaydigan, ammo dasturiy ta'minotlangan;
4. Dasturlashtirilishi mumkin, ammo dasturiy ta'minlanmagan.

Dasturlashtiriladigan protsessor bu buyruqlar tizimiga ega bo'lgan protsessor. Uni ma'lum bir muammoni hal qilish uchun sozlash mumkin. Dasturlashtirilmaydigan protsessor funktsiyalari bir marotaba va umuman belgilanadi. Protsessorlar apparat bloklari sifatida yoki dastur tomonidan boshqariladigan qurilmalar printsipi bo'yicha qurilishi mumkin.

### **Protsessor tasnifi**

Protsessorlarni funktsional yo'nalish, funktsional moslashuvchanlik, amalga oshirish usuli bo'yicha tasniflash tavsiya etiladi. Birinchi holda, ular universal (umumiy maqsadlar uchun) va ixtisoslashtirilgan (kirish/chiqish protsessoslari, grafik, matematik, signallarni qayta ishlash) protsessorlar haqida gapirishadi. Birinchisi turli muammolarni echishga mo'ljallangan va keng doiraga ega, ikkinchisi esa tor doiradagi muammolarni echishga qaratilgan. Universal protsessorlar quyidagilar bilan tavsiflanadi: ko'p sonli buyruqlarni qayta ishlash qobiliyati; buyruqlar tizimi-agar buyruqlar tizimi har qanday muammoni hal qilishga imkon bersa, protsessor universaldir. Buyruq tizimini uning balansi, ma'lumot tuzilishi, manzilni boshqarish usullarini tahlil qilish kerak. Lekin protsessorni ko'p qirralikligi nisbiy savol.

Shuningdek, markaziy, periferik va xizmat protsessorlarga ajratish mumkin. Markaziy protsessor hisoblash tizimining umumiy boshqaruvini amalga oshiradi, ma'lumotlarni asosiy qayta ishlashni amalga oshiradi, ularni O'rnatilgan tizimlarning boshqa elementlari

bilan almashadi, shuningdek O'rnatilgan tizimlar elementlarining ishlashini nazorat qiladi. Periferik protsessor hisoblash tizimining funksiyalarining faqat bir qismini bajaradi, ma'lumotlarni kirish / chiqish qurilmalari (kirish / chiqish protsessorlari ) bilan boshqaradi va almashadi va hisoblash jarayonida (ma'lumotlarning ishlov berish qismi) qatnashishi mumkin. Xizmat protsessor, odatda, asosiy hisoblash jarayonida ishtirok etmaydi va amalga nazorat va texnik vazifalari bajaradi, amalga instrumental funksiyasi (etkazib berish va dasturiy ta'minot nosozliklarni yo'q qilish, apparat konfiguratsiyasi), nazorat tizimining to'g'ri ishlashini, atrof-muhitni parametrlari (harorat, namlik) o'lchash, kuchlanish ta'minoti va boshqalar. O'rnatilgan tizimlarda bir xil protsessor periferik va xizmat protsessorlarining funksiyalarini bajarishi mumkin.

Funksional moslashuvchanlik darajasi yoki protsessorni istalgan vaqtda ruxsat etilgan to'plamdan ma'lum bir funktsiyani bajarishi uchun sozlash qobiliyati, uni dasturlash qobiliyati va samaradorligi bilan belgilanadi. Agar ishlash paytida funktsiyani qayta tuzish mumkin bo'lsa, unda bunday protsessor programlanadigan protsessor deb ataladi, aks holda biz dasturlanmaydigan, "qattiq" qurilma bilan ish olib boramiz (ajratilgan protsessor yaqin, ammo aniq atama emas, boshqa mumkin bo'lgan variant - bu apparat tezlatgich). Dasturni boshqarish mexanizmining murakkabligi juda keng doirada o'zgarishi mumkin.

Amalga oshirish uslubiga ko'ra, protsessorlar, avvalambor, apparat tomonidan amalga oshiriladigan va dasturiy ta'minotga bo'linishi kerak. Ushbu guruhlarining har biri, o'z navbatida, ko'p variantlarga ega. Protsessorning dasturiy ta'minotini amalga oshirish mezonlari, agar u kamida bitta dasturiy ta'minot tomonidan amalga oshirilgan qismdan iborat bo'lsa, amaliy foydalanish uchun juda mos deb hisoblanishi kerak. Ko'pincha dasturiy ta'minotni amalga oshirish qurilmani loyihalashda dasturiy ta'minotni boshqarish printsiptan foydalanish bilan bog'liq. Shuningdek, qurilma boshqaruv ma'lumotlarini dasturiy ta'minot sifatida tasniflash uchun uni izohlashi kerak degan fikr keng tarqalgan.

Qurilmani (funktsiyani) apparat yoki dasturiy ta'minotni amalga oshirishning umumiy mezonini qurilmada mavjud bo'lgan doimiy strukturaning ortiqcha (doimiy yoki tasodifiy kirish xotirasi bloki, mantiqiy matritsa, operatsion elementlar va boshqalar) darajasini ko'rib chiqish uchun taklif etiladi. Ushbu ta'rifdan kelib chiqadiki, uni amalga oshirish yo'lini nafaqat qurilmaning boshqarish qismini tashkil qilish



belgilaydi. Masalan, jadval funksiyalari o'zgartiruvchi dasturiy ta'minot qurilmasi deb ham ajratish mumkin. Jihozni amalga oshirishning aniq hodisasi - bu qurilmada umuman muntazam tuzilmalarning yo'qligi (tartibsiz "qattiq" mantiq yoki boshqacha qilib aytganda "zanjirlar chigalligi"). Amalda, bugungi kunda, bir qator sabablarga ko'ra, ortiqcha ortiqcha muntazam tuzilmalar asosida qurilma yaratiladi, so'ngra u hosil bo'lgan shaklda o'rnatiladi yoki yana bir qadam qo'yiladi - odatdagi strukturaning ishlatilmaydigan elementlari olib tashlanadi. Birinchi holda, biz dasturiy ta'minot bilan amalga oshiriladigan qurilma bilan shug'ullanamiz, ikkinchi holda, loyihalash bosqichida dasturiy ta'minot tomonidan amalga oshirilgan versiya apparatni amalga oshirish bilan almashtiriladi ("qattiq mantiq").

Shunday qilib, to'rtta asosiy protsessor variantlari haqida gapirishimiz mumkin :

- " qattiq" (dasturlashtirilmaydiladigan) protsessor ( dedicated hardware protsessor );
- apparat ta'minoti bilan dasturlanadigan protsessor ( programmable hardware processor);
- dasturiy ta'minotni amalga oshiradigan "qattiq" protsessor (dedicated software processor)
- dasturlashtiriladigan dasturiy protsessor (programmable software processor).

### **Mikroprosessor va mikrokontroller**

An'anaga ko'ra, u integral dasturlashtiriladigan protsessor sifatida talqin etiladi. Biroq, ASIC, ASSP, PLD (FPGA) texnologiyasining rivojlanishi bilan bog'liq holda, protsessor elementlarining to'rt guruhini ham ular birlashtirilgan bo'lsa, ularni mikroprotsessor deb atash mantiqan to'g'ri keladi.

Ko'rib chiqishning eng yuqori darajasidagi hisoblash tizimlarini uchta elementlar guruhi ifodalashi mumkin: ishlovchilar, xotira qurilmalari, interfeyslar. Protsessor va boshqaruvchi atamaları ishlov beruvchilar guruhi bilan bog'liq. Amaliy vazifa doirasida funksiyalari hali aniqlanmagan protsessorni ishlov berish elementi deb atash taklif etiladi. Belgilangan dastur funksiyasiga qarab (umuman tizimga emas, balki berilgan elementga nisbatan) protsessor boshqaruvchi (ya'ni boshqaruv moslamasi), multiplikator, nazoratchi, dispetcher, va boshqalar. O'z navbatida, kontrollerlar turli maqsadlarda bo'lishi

mumkin: xotira, printer, ketma-ket interfeys, texnologik jarayon va boshqalar.

Yana bir muhim atama - "mikrokontroller" ni mikroprotessor elementlari bazasi asosida qurilgan kontroller deb tushunish kerak. Mikrokontrollerlar bitta chipli, bitta taxtali, programlanadigan, mantiqiy, sanoat, universal va hk. Bitta chipdagi mikrokontroller tarkibiga mikroprotessor va periferik qurilmalar va kontrollerlar to'plami kiradi: uzilish tekshiruvi, taymerlar, tarmoq boshqaruvchisi, ketma-ket kanal boshqaruvchisi, xotira boshqaruvchisi, DMA boshqaruvchisi va boshqalar.

Mikrokontrollerlarni tasniflashning ko'plab usullari mavjud:

- 8, 16 va 32 bitli mikrokontrollerlar bit birligi bilan ajralib turadi;
- signalni qayta ishlash imkoniyatlarini ko'rib chiqish mumkin an'anaviy mikrokontrolörler va DSP mikrokontrolerlari;
- qo'llanish sohasiga ko'ra quyidagi mikrokontrollerlar ajratiladi: avtomobil, sanoat, boshqaruvchi tarmoqlari, dvigatelni boshqarish, simsiz tarmoqni boshqarish.

Mikrokontrollerlarni hisoblash resurslari miqdori bo'yicha shartli ravishda to'rtta guruhga ajratish mumkin:

- Periferik Protessorlar - Microchip Pic 10, PIC12, PIC16, PIC18, PIC 24, Atmel AT 90 xxxx ;
- Universal 8 va 16-bit-Intel MCS51, Siemens Sab 5 xx, Atmel Mega 10 x ;
- Universal 16- va 32-bitli - Fujitsu FR-50, ARM7 va boshqalar;
- Universal bitta-chip 32-bit Microcontrollerlar va Protessorlar MPC 560 xx, ARM 9, ARM 11.

**Birinchi** tarkib quyidagi xususiyatlar bilan tavsiflanadi:

- kichik hajmdagi ma'lumotlar xotirasi (o'nlab - yuzlab baytlar);
- oz miqdordagi dastur xotirasi (birlik - o'nlab kiloslovlar );
- nisbatan yuqori tezlik ;
- RISC buyruqlar tizimi;
- kam quvvat sarfi;
- kam sonli chiqishlar;
- tashqi xotirani ulashning iloji yo'qligi;

Mikrokontrollerlarning eski modellari tarmoq kontrollerlarini o'z ichiga olishi mumkin. Ushbu kontrollerlarning asosiy g'oyasi kam quvvat sarflaydigan va bortda minimal miqdordagi tarkibiy qismlarning mavjudligini ta'minlashdir.

**Ikkinchi** toifali protsessor tashqi ( chipsiz ) xotiradan foydalanish imkoniyati bilan tavsiflanadi. Ushbu turkum past narxlar va kichik hisoblash resurslari bilan ajralib turadi. Bunday mikrokontrollerning ishlash darajasi, odatda, birinchi toifaga qaraganda ancha past. Ushbu turdagi kontroller asosan yuqori ishlash va quvvat talablariga ega bo‘lmagan, ammo dastur kodi miqdori va kerakli ma'lumotlar xotirasi uchun yuqori talablarga ega oddiy va arzon qurilmalarda qo‘llaniladi.

**Uchinchi** toifali protsessorlar birinchi va ikkinchi toifalarga nisbatan ancha kuchli markaziy kompyuter va ma'lumotlarni qayta ishlashni yaxshilash uchun qo‘shimcha qurilmalarni ishlab chiqdilar. Ushbu toifadan boshlab ishlab chiqaruvchilar mikrokontroller tarkibiga faol ravishda tarmoq kontrollerlarini kiritishni boshlaydilar. Hozirgi vaqtda ular eng keng tarqalgan mikrokontrollerlardir.

**To‘rtinchi** toifadagi protsessorlar xotirani muhofaza qilish mexanizmlari va katta manzil maydonidan foydalanish bilan ajralib turadi, bu esa real vaqtda operatsion tizimlardan muammosiz foydalanishga imkon beradi. Ular uchinchi toifadan yuqori mahsuldorligi bilan ham ajralib turadi.

### **Dasturlashtiriladigan mantiqiy integral mikroxxemalar**

**Dasturlashtiriladigan mantiqiy qurilma (PLD, DMIM)** - bu mantiqiy kataklardan va sozlanishi o‘tkazgich simlaridan tashkil topgan elektron komponent. Asosiy maqsad - qayta tiklanadigan raqamli davrlarni qurish. Odatiy integral mikroxxemalardan farqli o‘laroq, ishlab chiqarish bosqichida funktsionallik bir marotaba aniqlanmaydi va oxirgi foydalanuvchi (muhandis) tomonidan ularning ehtiyojlariga qarab yaratilishi va o‘zgartirilishi mumkin.

PLD, birinchi yaqinlashishda, maxsus kommutatsiya matritsalarini yordamida ulangan bir xil turdagi mantiqiy elementlarning to‘plamidir. Elementlarni ulash va ishga tushirish PLD konfiguratsiya xotirasiga yuklangan ikkilik rasm yordamida amalga oshiriladi. Konfiguratsiya fayllarni (ikkilik tasvirlar) g eneriruyutsya SAPR ishlab xos PLD foydalanib, uning intellektual mulki hisoblanadi. FPGA apparat bazasida tizimlarni chipda amalga oshirishingiz mumkin. Loyihani tavsiflash strukturaviy-funktsional tavsiflash tillarida va uskunalarda ( Verilog, VHDL ), shuningdek, SystemC kabi yuqori darajadagi tillardan foydalanilgan holda amalga oshirilishi mumkin.

PLD ni qo‘llash doirasi:

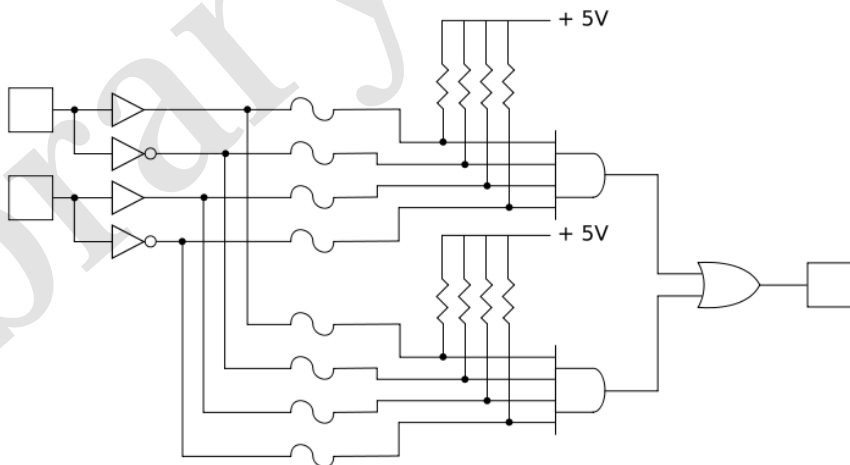
•Bog'lovchi mantiq ( elim mantiq'i ). PLD bir nechta interfeyslarning mosligini ta'minlash vositasi sifatida ishlaydi. Dastlab, bunga 74 va 40-qatorlarning mantiq'idan foydalangan holda va yanada murakkab holatlarda CPLD va FPGA-dan foydalangan holda erishildi. Misollar: manzil avtobus dekoderi, port ekspanderi; Elektron zanjirlarning intellektual mulkini shifrlash;

- IP-ni va kichik ishlab chiqarishda prototiplash uchun ;
- Signallarni, tasvirlarni raqamli qayta ishlash;
- Kriptografiya;
- Yuqori samaradorlik bilan hisoblash.

### Dasturlashtiriladigan mantiq matritsasi

**PAL ( Programmable Array Logic )** dasturlashtiriladigan mantiqiy matritsadir (DMM). Bu oddiy dasturlashtiriladigan mantiqiy integral elektron (DMIM).

PALLar ROM texnologiyasi asosida ishlab chiqilgan, ya'ni ular bir marta dasturlashtirilgan. Ichkarida, PAL bir nechta mantiqiy va elementlar bo'lib, ular kirishlari OR elementlariga sobit tarzda ulanadi. Dasturlash jarayoni istalgan mantiqiy funktsiyasini olish uchun yondirgichlarni yoqishdan iborat (3.2-rasm).



Rasm 3.2. PAL fragmenti: kirish va chiqish buferi, matritsa AND (2 ta shart), matritsa OR

## CPLD

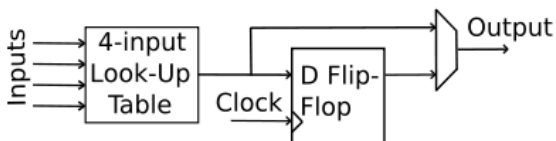
### CPLD ( murakkab bir dasturlashtiriladigan mantiqiy qurilma )

- elektron qurilma, dasturlashtiriladigan elektron tutashuv (DMIM) bir sinfga tegishli va FPGA va PAL orasida murakkabligi joylashgan. CPLDlar kommutatsiya matritsasiga dasturlashtiriladigan mantiq eshiklari bloklaridan iborat. FPGA-dan farqli o'laroq, CPLD davrlari odatda o'zgaruvchan bo'lmagan xotiraga asoslangan. So'nggi paytlarda CPLD va FPGA o'rtasidagi farqlar asta-sekin xiralashmoqda.

## FPGA ( Field-Programable Gate Array )

**FPGA ( Field-Programable Gate Array )** - bu bir xil turdagi mantiqiy elementlarning to'plami bo'lib, ularni dasturlashtiriladigan kommutatsiya zanjiri yordamida ulash mumkin. FPGA-larni yaratishning asosiy maqsadi laboratoriyada nisbatan sodda texnologiyalardan foydalangan holda loyihaerga chipda ancha murakkab va ishlaydigan raqamli moslamani olish imkonini berishdir. Shunday qilib, FPGA texnologiyasidan foydalanib, biz nimaga erishamiz? Birinchidan, biz integral mikrosxemalarni murakkab ishlab chiqarilishidan xalos bo'lamiz. Ikkinchidan, kristall loyihasi mikrosxemaning topologiyasi va turli tugunlarning o'zaro ta'siri tufayli muammolar yo'qligi sababli soddalashtirilgan. Biz nimani yo'qotmoqdamiz? Biz ish faoliyatini yo'qotamiz, ishonchlilikni pasaytiramiz, shovqinlarga sezgirlikni oshiramiz, kristalning energiya sarfini ko'paytiramiz va klapanlar sonini 20.. 30 baravar ko'payishi sababli kristal maydonidan kamroq foyda olamiz.

FPGA qo'llanilishining asosiy yo'nalishlari quyidagilardan iborat: kichik va o'rta qurilmalarni prototiplash, simulyatsiya qilish va ishlab chiqarish. Qurilmalarning etarlicha katta partiyalari bilan ASIC ishlab chiqarish iqtisodiy jihatdan samarali bo'ladi. Sxemalar mantiqiy elementlar asosida qurilgan. Har bir FPGA mantiqiy elementi ikkita asosiy qismdan iborat: dasturlashtiriladigan mantiqiy element (LUT - izlash jadvali deb nomlangan ) va chiqish tetiklari (3.3-rasm).



3.3-rasm. FPGA mantiqiy elementi

Bitta mantiqiy element 2 va-NOT yoki 2 OR-NOT asoslari asosida yaratilgan bir necha o'nlab mantiqiy eshiklardan iborat. Dasturlashtiriladigan mantiqiy element har qanday kombinatsiyalangan zanjirni amalga oshirishga imkon beradigan haqiqat jadvalini o'z ichiga oladi. Qoida tariqasida, bunday elementlar xotira registrlari to'plami sifatida amalga oshiriladi. FPGA deb nomlangan konfiguratsiya xotirasi LUT uchun mantiqiy elementlar va haqiqat jadvalini ulash uchun zarur bo'lgan ma'lumotlarni o'z ichiga oladi. FPGA'larni tashkil etadigan mantiqiy elementlar RTL qatlamiga tegishli. Ularning aloqasi tarkibiy VHDL va Verilog tomonidan yaxshi tavsiflangan. Amaldagi hisoblash modeli diskret hodisa modelidir. Integral mikrosxemalarni ishlab chiqarish texnologiyasi yaxshilangani sayin, FPGA'larning asosiy elementlarini kengaytirish tendentsiyasi mavjud. Ko'pgina zamonaviy modellar xotirani, arifmetik mantiqiy asboblarning elementlarini, ko'paytirgichlarni va butun protsessor yadrolarini qo'shadi.

### **Tizim kristallari**

**Tizim kristallari ( System-on-chip, SoC )** - bir tizimining umumiy holda Bir yonga ustida ba'zi xotira, periferik qurilmalar va interfeyslar bir qator integratsiya protsessor (mutaxassisligi jumladan Protsessorlar,) - deb, bir maksimal, tizim oldiga qo'yilgan vazifalarni hal qilish uchun zarur. "Chipdagi tizim" iborasi, aniq ma'noda, atama emas. Ushbu kontseptsiya funktsiyalarning integratsiyasi tufayli integratsiya darajasini oshirishning umumiy tendentsiyasini aks ettiradi. " Tizimdagi chip" sinfidagi qurilmalarning ishlashi ko'p jihatdan barcha tarkibiy qismlarning o'zaro ta'sirining samaradorligiga va tashqi, tashqi ko'rinishi bilan, qurilmaga nisbatan ta'sirlanishiga bog'liq. Bu, birinchi navbatda, ichki qismlarning tezligi farqiga bog'liq, ayniqsa interfeyslarni tashkil qilish.

Chipdagi tizimlar odatda uchta asosiy raqamli tizim birliklaridan iborat: protsessor, xotira va mantiq. CPU asosiy asbob bir nazorat oqimi qachon odnozn har nazorat dasturi achno butun mumkin algoritmlarini biri belgilash uchun imkon beradi operatsiyalarni qayta ishlash majmui natija ma'lumotlar integratsiya elektron qilingan. Xotira belgilangan maqsadda ishlatiladi - protsessor yadrosi va ma'lumotlarning dastur kodini saqlash. Va nihoyat, mantiqiy ma'lumotlarni qayta ishlash va uzatish uchun ixtisoslashgan apparat vositalarini amalga oshirish uchun ishlatiladi, ularning tarkibi va maqsadi oxirgi dastur - ma'lumotlar oqimi tomonidan belgilanadi.

Chipdagi haqiqiy tizim, ushbu bloklarning kamida uchtasini o‘z ichiga oladi, bu ko‘plab alohida integral mikrosxemalarni ishlatishni va ular orasidagi aloqa interfeyslarini amalga oshirishni yo‘q qiladi. Bitta chipli, sozlanishi yoki dasturlashtiriladigan yechim o‘zining ichki apparat tuzilishini va ishlab chiqarish bosqichida ham, maydonda ham, to‘g‘ridan-to‘g‘ri loyihada tezkor o‘zgartirishga imkon beradi.

SoC asosidagi tipik o‘rnatilgan tizim quyidagi interfeys va kontrollerlarning turli xil to‘plamlarini o‘z ichiga oladi :

LPC / ISA, PCI, PC MCIA tizimli avtobus va avtobus kontrollerlari ;

- Kontrollerlar nazorat NOR / NAND Flash, SDRAM, SRAM, DDR;

- Ethernet kontrolleri ;

- Serial interfeyslarni UART, SPI / SSP / uWire, RS-232, RS-422 / RS-485, JON;

- Simsiz interfeysi WiFi /IEEE802.11, ZigBee, Bluetooth, IrDA;

- Flash xotira kartasini qo‘llab-quvvatlash interfeyslari : SD / MMC, CompactFlash, MemoryStick ;

- Controller LCD STN / TFT / OLED;

- matritsa klaviaturasi boshqaruvchisi;

- GSM / GPRS, CDMA simsiz uzatish modullari;

- GPS yo‘ldosh navigatsiya tizimlaridan signallarni qabul qilish uchun modullar,

- GLONASS ;

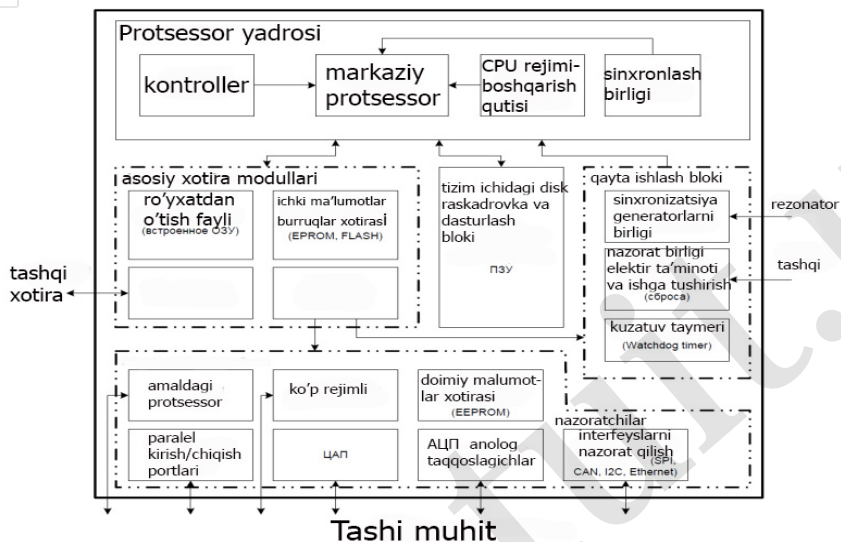
- Suzuvchi, shifrlash, DRM va boshqalarni texnik qo‘llab-quvvatlash;

- Audio va video interfeyslari.

### **3.5.3. O‘rnatilgan tizim tashkil etishning modulli printsiplari**

#### **O‘rnatilgan tizimlar uchun odatiy protsessor tuzilishi**

Hozirgi vaqtda ko‘milgan tizimlarda foydalanish uchun turli xil tuzilish va funktsiyalarga ega bo‘lgan juda ko‘p sonli protsessorlar mavjud. Ushbu nomenklatura turli xil amaliy dasturlarda muayyan muammolarni hal qilish uchun doimiy ravishda kengayib bormoqda. Qisqa vaqt ichida yangi modellarni ishlab chiqish va ishlab chiqarish qobiliyati tarkibiy tashkilotning modulli printsiplari ta‘minlaydi (3.4-rasm).



3.4-rasm. O‘rnatilgan tizimlar uchun odatiy protsessor tuzilishi.

Modulli qurilish prinsipi bilan bir xil oilaning barcha protsessorlari bir xil asosiy funksional blokni - protsessor yadrosini va o‘zgaruvchan funksional blokni o‘z ichiga oladi. Asosiy blok (protsessor yadrosi) quyidagilarni o‘z ichiga oladi:

- Markaziy protsessor;
- Manzil, ma'lumotlar va boshqarishning ertalabki qatorlari;
- Turli fazalarga ega bo‘lgan va ko‘p signallarni hosil qilish uchun blok

markaziy protsessor va ichki magistrallarni sinxronlashtirish uchun chastotalar;

- Protsessorni faol rejimga, ishlamay qolish rejimiga, kam quvvat sarfi rejimiga va qayta yoqish rejimiga o‘rnatishi mumkin bo‘lgan protsessor ish rejimini boshqarish bloki.

Protsessor yadrosi ma‘lum protsessorlar oilasi arxitekturasining asosiy ajralib turadigan xususiyati, shuning uchun u (yadro) oila nomi bilan ataladi. Masalan, MCS-51 yadrosi yoki PIC16 yadrosi. O‘zgaruvchan funktsiya bloki quyidagilarni o‘z ichiga oladi:



- Turli xil turdagi xotira modullari: SRAM tipidagi operativ xotira, ROM, EPROM yoki FLASH turidagi buyruqlar (dasturlar) uchun faqat o'qish, EEPROM turidagi energiyadan mustaqil ma'lumotlar xotirasi;
- Periferik qurilmalar modullari.

### **Boshqarish va sinxronizatsiya modullari.**

Turli xil chiplarida o'zgaruvchan funktsional blokning modullari boshqacha bo'lishi mumkin. Bitta oilaning mikrosxemalarida amalga oshirilgan modullarning umumiy to'plamiga ushbu oilaning periferik modullari kutubxonasi deyiladi. Ushbu kutubxonada aytilganidek, nafaqat periferik, balki xotira modullari, o'rnatilgan sinxronizatsiya generatorlari, nosozlik yoki "tashqi qayta tiklash" holatida elektr ta'minoti va tizimni qayta ishga tushirish signallarini boshqarish bloki, shuningdek, nosozliklarni tuzatish va dasturiy modullar mavjud. So'nggi paytlarda " System-On-Chip " yo'nalishi faol rivojlanmoqda, bunda oxirgi foydalanuvchi o'zi taqdim etilgan periferik modullar kutubxonasidan maxsus protsessor tarkibini tuzishi, shuningdek yangi modullarni mustaqil ravishda ishlab chiqishi mumkin.

### **Protsessor yadrosi**

Protsessor yadrosining texnik echimi quyidagi parametrlar bilan belgilanadi:

- Arxitektura parametrlari - registrlar to'plami, xotirani tashkillashtirish, operandalarni xotirada murojaat qilish usullari, ushbu ma'lumotlarni qayta ishlash bo'yicha ko'rsatmalar tizimi.

- Sxematehnik echimlari – registrlar majmui, ALU, shinalarni boshqarish sxemalari va boshqalar. Sxematehnik moslamasi ichki ishlash diagrammasini ham belgilaydi - registrlar, xotira va ALU o'rtasidagi shinalar bo'ylab ma'lumotlar harakati ketma-ketligi.

- Ishlab chiqarish texnologiyasi - sxemaning ruxsat etilgan murakkabligini, maksimal o'tish chastotasini, quvvat sarfini aniqlaydi. O'rnatilgan tizim uchun zamonaviy CPU sifatida amalga CISC-arxitektura( Motorola HC11, Intel. MCS-51, AMD Am186) va RISC mashinalari -arxitektura ( MicrochipPIC, Atmel AVR, Triscend E 7-ARM haqida ) tizimlari ishlatiladi. Protsessor yadrosining ishlashi omillar kombinatsiyasi bilan aniqlanadi :

- Shinaning modullararo manzili va ma'lumotlar uzatish liniyasining soat tezligi. Sinxronizatsiya generatorining chastotasidan Fxclk har bir protsessor yadrosi uchun nisbati individualiga qarab aniqlanadi.

• Vaqt birligi uchun ro'yxatdan o'tkaziladigan-ro'yxatdan o'tkazmalar soni. RISC protsessorlari uchun bu bitta shina tsikli uchun bitta, CISC uchun - 1..3 o'tkazmalari.

• Muayyan boshqaruv algoritmidagi eng ko'p ishlatiladigan operatsiyalarni bajarishda ishlash.

• Qo'ng'iroq qilish vaqti / vaqti-vaqti bilan uzilish xizmati. Ushbu parametr real vaqtda qattiq ishlash uchun muhimdir va qayta ishlangan hodisalarning maksimal intensivligini aniqlaydi.

### **Boshqarish protsessorlarida uzilishlarni tashkil qilish**

Buzilishlarning manbalari bo'lishi mumkin:

1. **Tashqi manbalar.** So'rov kirishdagi kuchlanish pasayishi bilan ("1" dan "0" gacha yoki "0" dan "1" gacha) yoki ma'lum bir kuchlanish darajasi ("0" yoki "1") tashqi so'rovning tashqi kirishida yuboriladi.

2. **Ichki manbalar** - o'rnatilgan xotira modullari (odatda EEPROM modulidan ) yoki periferik modullar:

a) Taymerlar / hisoblagichlar. So'rov toshib ketganda hosil bo'ladi;

b) Qo'lga olish / taqqoslash bloklari. Kirishni taqqoslash uchun kirish yoki qo'lga olish hodisasini so'rash.

c) ARO'. O'zgartirishni yakunlashni talab qiling.

d) Analog taqqoslagichlar. Kirish signali darajalarining nisbatlarini o'zgartirish to'g'risida so'rov.

e) Qabul qiluvchi serial interfeysi (RS-232, SPI, I2C, USB, CAN, Ethernet, HDLC ).

So'rov hosil qilinadi:

• Bayt yoki paketni olgandan so'ng va yangi olingan ma'lumotlar mavjud bo'lganda;

• Baytni yoki paketni uzatishni tugatgandan so'ng va uzatuvchini bo'shatish.

### **3. Dastur bilan uzilishlar.**

Boshqarish tizimlari uchun protsessorlarda uzilishlarni tashkil qilish universal protsessorlardan tubdan farq qilmaydi. Turli xil boshqaruv protsessorlari oilalarida turli xil uzilishlar mexanizmlari mavjud :

1. Qattiq ustuvor bo'lgan vektor (ST7, AVR, Am186).

2. Dasturlashtiriladigan ustuvorlikka ega bo'lgan vektor (MCS-51, M16C, i386EX).

3. Vektorlarning dinamik jadvali (M16C) bo'lgan vektor.

4. Umumiy vektor bilan ( poling mexanizmi ) (PIC).

## **Uzilish so'rovlarni qayta ishlash uchun blokning kengaytirilgan diagrammasi (niqoblar mexanizmi)**

Niqob mexanizmi har bir so'rov uchun maxsus bitdan foydalanishga asoslangan, uning yordamida ushbu so'rov bilan bog'liq bo'lgan uzilishlarni qayta ishlashga ruxsat beriladi yoki o'chiriladi. Intel oilasining protsessorida IF biti niqob funksiyasini bajaradi, uning yordamida ( $IF=1$ ) yoki tashqi aralashuv so'rovlarini qayta ishlashni taqiqlaydi ( $IF=0$ ) (odatda quldan). IF bayrog'i o'chirilganda, uzilishlar niqoblangan deb aytish odatiy holdir. O'rnatilgan tizimlardan uzilish so'rovlari protsessorga so'rov liniyasi bilan ulangan PICyordamida hosil bo'ladi. PIC-dan so'rovlar CPU ni tashqi INTR kirishga yuboriladi.

Uzilishlarni boshqarish uchun uzilishlar niqobidan foydalaniladi, bu  $M = m_1, m_2, \dots, m_k$  ikkilik sonidir va uzilishning maskalanadigan sabablari soniga teng. Agar niqobni darajasi  $mk=0$  bo'lsa,  $k$  tufayli uzilish taqiqlanadi (niqoblangan), agar niqob darajasi  $mk=1$  bo'lsa,  $k$  tufayli uzilish yoqilgan (niqoblanmagan). Uzilishlar niqobi protsessorida saqlanadi, u erda INSTALL MASK A buyrug'i yordamida yuklanadi, bu erda A - manzil. Ushbu buyruq bilan A manzili bo'lgan so'z protsessorga niqob sifatida yuklanadi va protsessorning signallarni uzishga nisbati aniqlanadi. Agar barcha niqob bitlari nolga teng bo'lsa, protsessor biron bir uzilish sababiga javob bermaydi.

Eng oddiy protsessorlar quyidagi uzilishlarni maskalash usulidan foydalanadilar. Kompyuter buyruqlar tizimiga ikkita tizim buyrug'i kiritilgan:

### **• O'ZGARTIRISHNING CHIROQLARI**

### **• INTERRUPTIONLARGA RUXSAT BERING**

uning bajarilishi bir vaqtning o'zida barcha sabablarga ko'ra uzilishlarning taqiqlanishi va ruxsatiga olib keladi. Uzilishlarni maskalash buyruqlari imtiyozli buyruqlar guruhiga tegishli. Tashqi uzilish so'rovi darajasi / chekkasi tanlovchisi tashqi uzilish so'rovi keltirib chiqaradigan hodisani tanlaydi ("tashqi uzilish"). Quyidagi sozlamalar mumkin: signal chekkasi yoki tushishi bilan yoki daraja bo'yicha. Uzilishlar so'rovi paydo bo'lganda, IF bayroqlar registrida manba biti o'rnatiladi. Uzilishlarni maskalash mantig'i ma'lum manbalardan yoki barcha manbalardan so'rovlarni bir vaqtning o'zida ishlab chiqarishga imkon beradi yoki o'chiradi. Uzilishlarni yoqish uchun Interrupt Enable (IE) registri va Global Interrupt Enable (GIE) bitining mos biti 1 ga o'rnatilishi kerak. Maskalash mantig'ining

maskalanmaydigan uzilishlarga ta'siri yo'q. Ustuvorlik mantig'i (Interrupt Priority, IP) eng yuqori ustuvor so'rov uchun vektor hosil qiladi va uni uzilish so'rovi sinxron ravishda hisoblash yadrosiga o'tkazadi; so'rovni qayta ishlashda ustuvorligini nazorat qiladi va agar ko'proq ustuvorlikka ega boshqa so'rov kelib tushsa, ma'lumotlar so'rovini oldindan ko'rib chiqadi (to'xtatadi).

### **Xotira modullari**

Xotira - dasturlarni, qayta ishlangan ma'lumotlarni (ma'lumotlarni), hisob-kitoblarning oraliq yoki yakuniy natijalarini saqlash uchun mo'ljallangan qurilmalar to'plami. Xotiraning eng muhim xususiyatlari - bu hajmi, tezlik va qiymati. Xotira hajmi xotirada saqlanadigan maksimal ma'lumot miqdori bilan belgilanadi va kilo, mega va gigabaytlarda hisoblanadi. Xotira qurilmasining tezligi xotira qurilmasiga kirishda ma'lumotni o'qish va yozish uchun sarflanadigan vaqt bilan tavsiflanadi. Xotira qiymati - bu xotira hajmi bilan aniqlangan ma'lumotlarning butun hajmini saqlash uchun pul bilan ifodalangan mablag' xarajatlari. Xotiraning har xil turlarining sifatini taqqoslash uchun birlik qiymati deb nomlangan va xotira sig'imiga bo'linadigan xotira narxiga teng bo'lgan xarakteristikadan foydalaniladi. Birlik narxi o'lchovga ega, masalan, dollar / MB.

Xotira qurilmalarining maqsadi va amalga oshirish xususiyatlariga qarab, ularni tasniflash masalalariga ham turlicha yondashiladi.

#### **Tasnif mezonlari:**

1. Maqsad bo'yicha;
2. Jismoniy vosita turi bo'yicha (ishlab chiqarish texnologiyasi);
3. Kirishni tashkil etish bo'yicha (manzil: ixtiyoriy, to'g'ridan-to'g'ri, ketma-ket; assotsiativ kirish);
4. Yozib olish va qayta yozish ilojisi bo'yicha;
5. Quvvatga qaramlik / quvvatga qaram bo'lmaslik bo'yicha ;
6. Interfeys turi bo'yicha;
7. Manzil maydonini tashkil etish turi bo'yicha;
8. Markaziy protsessor uchun masofaviyligi va mavjudligi (asosiy, ikkilamchi, uchinchi darajali xotira) bo'yicha.

"Xotira moduli" atamasi xotira kataklarining haqiqiy qatorlarini yozishni o'chirish uchun maxsus analog va raqamli boshqaruv zanjirlari, quvvat manbai (va ba'zan manbalar) bilan, rejimlarni boshqarish registrlari bilan birlashtirishni anglatadi.

Misol sifatida, o'zgaruvchanlik mezonini bo'yicha xotira modullarining tasnifini beramiz. Bunday holda, xotira modullari ROM(doimiy) va RAM (operativ)ga bo'linadi.

### **ROM modullari:**

- **Mask ROM (MaskROM)** - zavodda yozilgan va foydalanuvchi tomonidan o'zgartirilishi mumkin emas. Ular yuqori saqlash sifatiga ega. ROMning eng arzon turi. Ular bir necha o'n minglab dona katta miqdorda ishlab chiqarilgan mahsulotlar uchun ishlatiladi.

- **ROM**, foydalanuvchi tomonidan bir martalik dasturlashtirilishi mumkin (Bir martalik dasturlashtiriladigan ROM,OTPRM). Foydalanuvchi dasturlashtirilishi mumkin. Zavodda chiqarilganda barcha hujayralar FFh qiymatlariga ega. Bitlarga kuchlanish impulslarini qo'llash orqali "0" yozilishi mumkin, ammo teskari yozuv - "1" ga endi mumkin emas. Ular dasturlash rejimlariga (kuchlanish darajalari, vaqt diagrammasi, sinov rejimlari) qat'iy rioya qilgan holda yuqori saqlash sifatiga ega, aks holda, bir muncha vaqt (oylar yoki yillar) o'tgach, bitlar o'z-o'zidan "dasturlashtirilmasligi" mumkin - "1" holatiga o'tishi mumkin. Arzon ROM. Mahsulotlarning kichik partiyalari uchun ishlatiladi.

- Ultra ultrabinafsha yoki rentgen nurlarini o'chirishda foydalanuvchi tomonidan dasturlashtiriladigan ROM (**EPROM**). Ko'p marta dasturlash (bir necha o'n marta) ruxsat etiladi. Dasturlash texnologiyasi OTPROMga o'xshaydi, ammo "0" da dasturlashtirilgan barcha hujayralarni ultrabinafsha nurlar ostida "1" holatiga olib tashlash mumkin. Buning uchun tanada maxsus kvarts oynasi mavjud. Dasturni o'chirish rejimlarining buzilishi qayta dasturlash davrlari va saqlash vaqtining keskin qisqarishiga olib keladi. Juda qimmat xotira (OTPRMga qaraganda kattaroq buyurtma haqida). Nosozliklarni tuzatish namunalari ishlatiladi.

- Foydalanuvchi tomonidan elektr o'chirilishi bilan dasturlashtiriladigan ROM (**Elektr bilan o'chiriladigan dasturlashtiriladigan ROM - EEPROM** yoki E2PROM). Har qanday katak ustiga yozish mumkin. Bunday holda, o'chirish avtomatik ravishda amalga oshiriladi, foydalanuvchi uchun shaffof. 1.000.000 donagacha qayta yozish tsikllari soni. Saqlashning sezilarli vaqti (yillar.. 10 yil). Biroq, EEPROM bloklari cheklangan hajmga ega (bayt... o'nlab kB) va shuning uchun ular deyarli har doim ma'lumotlar xotirasi sifatida ishlatiladi.

• Elektr bilan o'chirilgan **Flash ROM** - bu hajmi sezilarli darajada oshgan EEPROM modifikatsiyasi. Ovoz balandligini oshirish uchun har bir bit uchun o'chirish sxemalari alohida olib tashlandi va o'chirish o'nlab baytdan o'nlab KB gacha bo'lgan sahifalarda amalga oshiriladi. Bundan tashqari, bir vaqtning o'zida bir nechta sahifadagi bloklarni yoki barcha xotirani o'chirish mumkin. Ushbu ishlash tartibi (sahifani o'chirish) ma'lumotlarni saqlash uchun noqulay, ammo dasturlarni yozish uchun maqbuldir. Shuning uchun FLASH xotirasi dastur xotirasi sifatida ishlatiladi. O'rnatilgan FLASH-xotira hajmi o'ndan yuzlab kBgacha. Qayta dasturlash tsikllarining soni 100000 gacha, saqlash muddati 10 yilgacha. Vaqt o'nlab o'chirish Milodiy, kilobayt dasturiy vaqtini - o'nlab mikro bayt boshiga. Ta'minot kuchlanishi 1,8 V dan iborat. FLASH tipidagi ROM hozirgi vaqtda ROM xotirasining ichki va tashqi modullari (mikrosxemalari) sohasida etakchi mavqega ega.

O'rnatilgan RAM sifatida, aksariyat hollarda statik xotira modullari ( Static Random Access Memory, SRAM ) ishlatiladi. Statik tasodifiy kirish xotirasi chipining yadrosi triggerlar to'plamidir - ikkita barqaror holatga ega mantiqiy qurilmalar, ulardan biri shartli ravishda mantiqiy nolga, ikkinchisi mantiqiy birlikka to'g'ri keladi. Boshqacha aytganda, har bir trigger bir oz ma'lumot saqlaydi.

Triggerning dinamik RAMdagi kondansator bilan solishtirganda afzalliklari quyidagilarni o'z ichiga oladi:

- tetik holatlari barqaror va quvvat mavjud bo'lganda cheksiz davom etishi mumkin, kondansator esa davriy yangilanishni talab qiladi;

- ozgina inertlikka ega bo'lgan tetik bir necha gigagertsgacha chastotalarda muammosiz ishlaydi, kondansatorlar esa 75-100 MGts da "qulab tushadi".

Triggerlarning kamchiliklari orasida ularning yuqori narxlari va ma'lumotlarni saqlashning past zichligi mavjud.

Agar dinamik xotira xujayrasini yaratish uchun faqat bitta tranzistor va bitta kondansator etarli bo'lsa, unda statik xotira xujayrasi kamida to'rttadan va o'rtacha oltita sakkizta tranzistordan iborat, shuning uchun statik xotira qurilmalari dinamikadan 4... 5 baravar qimmatroq bir xil va taxminan bir vaqtning o'zida kamroq ma'lumot hajmi. Ularning afzalligi - bu yuqori ishlashi va odatdagi dastur maydoni - bu kesh xotirasi davrlari.

**Dinamik RAM** (DRAM) da ma'lumotlar MOS tuzilmalari elementlari tomonidan hosil qilingan kondansator zaryadlari ko'rinishida saqlanadi. Kondansatorlarning o'z-o'zidan tushishi ma'lumotlarning

yo‘q qilinishiga olib keladi, shuning uchun ular vaqti-vaqti bilan (har bir necha millisekundlarda) yangilanishi kerak. Shu bilan birga, dinamik xotira elementlarining o‘rash zichligi statik RAMda erishilgan o‘rash zichligidan bir necha baravar yuqori. Dinamik xotirada ma‘lumotlarni tiklash maxsus kontrollerlar yordamida amalga oshiriladi. Shuningdek, ichki o‘rnatilgan regeneratsiya tizimiga ega bo‘lgan dinamik xotira elementlari bo‘lgan xotiralar ishlab chiqilgan bo‘lib, unda boshqaruv signallariga nisbatan tashqi xatti-harakatlar statik xotiralarning xatti-harakatlariga o‘xshash bo‘ladi. Bunday xotiralar kvasistatik deb nomlanadi.

Dinamik xotira qurilmalari eng katta ma'lumot sig‘imi va arzonligi bilan ajralib turadi, shuning uchun ular kompyuterlarning asosiy xotirasi sifatida ishlatiladi.

**Statik RAM**ni dinamik bo‘lmagan asosiy xotiraning moduli sifatida tanlash protsessor to‘liq to‘xtaguncha chastotani pasaytirganda ma‘lumotlarni saqlash qobiliyati bilan belgilanadi. Ushbu rejim energiyani tejash uchun ishlatiladi, masalan, batareya quvvatidan foydalanganda.

Hozirgi bosqichda o‘rnatilgan RAM modullarida sezilarli miqdor yo‘q - ba‘zida o‘nlab kilobayt birliklar. Katta hajmdagi RAM talab qiladigan kuchli tizimlar holatida tashqi xotira chiplari ulanadi. E bir yonga statik RAM va dinamik RAM (DRAM, SDRAM) bo‘lishi mumkin. ikkinchi holda, (masalan, AMD Am186ED, protsessor Mitsubishi M16C boshq.), bir dinamik RAM interfeysi moduliga ega ekanligini chip klassik interfeysi qo‘llab-quvvatlaydi m DRAM, yoki SDRAM, lekin, shuningdek, dinamik xotira regeneratsiyasi.

Zamonaviy RAM modullarining ikkinchi muhim xususiyati past, taxminan 1 V, ma‘lumotni saqlash uchun kuchlanish. Bu elektr uzilishlari paytida ma‘lumotlarni saqlashga imkon beradi. Xuddi shu maqsadda, ba‘zida batareya statik RAM moduliga kiritilgan bo‘lib, u ma‘lumotlarni yillargacha saqlashga imkon beradi ( DS 5000 Dallas Semiconductor oilasi).

## **Energiyaga qaram bo‘lmagan E2PROM xotirasi**

EEPROM Don Frohman tomonidan o‘sha Intelda ixtiro qilingan suzuvchi eshik tranzistoriga asoslangan. Va kelajakda, texnologik davrlarning o‘zgarishiga qaramay, doimiy xotira xujayrasi qurilmasining

printsipi o'zgarishsiz qoldi - o'chirish va yozishning qanday usuli ishlatilgan.

EEPROM xotirasidan foydalanishning asosiy afzalligi shundaki, uni doskadan olib tashlamasdan uni ko'p marta qayta dasturlash mumkin. Ushbu dasturlash usuli "Tizimda dasturlash" yoki "ISP" deb nomlanadi. Shu bilan birga, dasturlash xarajatlari kamayadi. Qayta yozish jarayonida oksid qatlami tutilgan elektronlarni asta-sekin to'playdi, natijada ular suzuvchi eshikka o'tishi mumkin. Bu "1" va "0" mos keladigan chegara voltajlari orasidagi farqni kamaytiradi. Etarlicha qayta yozish davrlaridan so'ng, farq tanib bo'lmaydigan darajada kichik bo'ladi. Qoida tariqasida, qayta yozilganlarning eng kam soni yuz minglab va million marta ko'payadi. Yozib olish paytida suzuvchi eshikka qilingan elektronlar izolyatordan o'tishi mumkin, ayniqsa harorat ko'tarilganda va bu holda zaryad yo'qolishi mumkin, ya'ni hujayra haqidagi ma'lumotlar o'chiriladi. Odatda ishlab chiqaruvchilar ma'lumotlarning kamida 10 yil saqlanib qolishiga kafolat berishadi.

### **Kirish / chiqish portlari**

O'rnatilgan dasturlar uchun har bir protsessorda tashqi portlar deb nomlangan mikrosxemaning tashqi pinlariga ulangan bir qator tashqi I / U chiziqlari mavjud. Yagona (1 bitli, bitta qatorli) I / U portlari guruhlangan bo'lib, odatda 4, 8 yoki 16 qatordan iborat bo'lib, parallel portlar deb nomlanadi. Parallel portlarning bit kengligi nostandart bo'lishi mumkin, masalan, PIC16F84 mikrokontrolleridagi 5-bitli port. Portlar orqali protsessor yadrosi turli xil tashqi qurilmalar bilan o'zaro aloqada bo'ladi - u kirish signallarining qiymatlarini o'qiydi va chiqish signallarining qiymatlarini o'rnatadi.

O'rnatilgan tizimlarda sensorlar, aktuatorlar, operatorning kirish-chiqarish moslamalari va tashqi xotira qurilmalari ko'pincha tashqi qurilmalar hisoblanadi.

Portlar signal turi bo'yicha ajralib turadi:

1. **Diskret** (raqamli) - mantiqiy "0" yoki "1" ning diskret qiymatlarini kiritish-chiqarish uchun ishlatiladi. O'rnatilgan dasturlarning aksariyat zamonaviy protsessorlari har ikkala parallel port chizig'ini mustaqil boshqarish va barcha bitlarni guruh nazoratini qo'llab-quvvatlaydi. Bitta 4, 8 yoki 16-bitli port doirasidagi alohida chiziqlarning sxemasi bir xil bo'lganligi sababli, qurilma va bitta razryadning ishlashi ko'rib chiqiladi;



2. **Analog** - ular orqali signallar ADC yoki boshqa analog davrlarning kirish qismiga kiritiladi va DAC yoki boshqa analog davrlarning chiqish signallari chiqariladi. Analog portlar (yoki analog rejimdagi sozlanishi portlar) - tashqi signallarni DAC, ADC yoki analog taqqoslagichlarga, o'ralgan qabul qiluvchi-uzatgichlarga ulash uchun ishlatiladi. DAC, ADC yoki komparator bilan ishlash rejimida portlar odatda 0V- dan Usup + gacha bo'lgan signallarni kiritishga imkon beradi (indekslar + va - biroz kattaroq va biroz kamroq degani, taxminan 200,.. 300mV). Transceiver rejimida signal parametrlari ma'lum bir interfeys bilan belgilanadi. Ko'pgina hollarda transduserlarga ulanish uchun analog yoki raqamli liniyalar umuman portlar deb nomlanmaydi, garchi ular sxemaga yaqin va protsessor tarkibida universal I / U portlariga ega bo'lsa. Kirish va chiqish bosqichlarini amalga oshirish ADC sxemasiga, taqqoslagichga, DAC yoki qabul qilgichga bog'liq;

3. **Sozlanuvchi** - analog yoki raqamli ishlash uchun sozlangan.

Signal uzatish yo'nalishi bo'yicha quyidagilar mavjud:

1. Faqatgina kirish uchun mo'ljallangan bir tomonlama portlar (kirish portlari, kirish portlari) yoki faqat chiqish (chiqish portlari, chiqish portlari).

2. Etkazish yo'nalishi sxemaning dastur tomonidan boshqariladigan konfiguratsiyasi paytida aniqlanadigan ikki tomonlama portlar.

3. Muqobil funktsiyaga ega portlar. Ushbu portlarning alohida chiziqlari taymer, ketma-ket qabul qilgich-nazorat qilish moslamalari kabi tashqi qurilmalarga ulangan. Agar tegishli periferik modul jalb qilinmasa, u holda chiziqlar odatdagi portlar sifatida ishlatilishi mumkin, agar modul yoqilgan bo'lsa, u bilan bog'liq bo'lgan chiziqlar avtomatik ravishda yoki "qo'lda" (dasturiy ta'minot) funktsional maqsadga muvofiq tuzilgan va ulardan foydalanish mumkin emas universal I / O portlari sifatida... Ba'zi hollarda, portlar faqat atrof-muhit moduli bilan aloqa qilish uchun ishlatilishi mumkin (masalan, ba'zi protsessorlarda ADC kirishlari).

Almashish algoritmiga ko'ra portlar ajratiladi:

1. Dasturiy ta'minot bilan boshqariladigan (dasturiy ta'minot) kirish-chiqarish bilan - ma'lumotlarni sozlash va o'qish faqat hisoblash jarayoni davomida aniqlanadi. Chiqishdagi bir xil (o'zgarmagan) qiymatni takroriy o'qish va chiqishda vaqtinchalik jarayon davomida o'qish-yozishdan himoya yo'q.

2. Strob bilan - har bir kirish / chiqish jarayoni signal manbasidan sinxronizatsiya pulsi (strobi) bilan tasdiqlanadi (chiqish paytida -

protsektor, kirish paytida - tashqi qurilma). Qabul qilgich ma'lumotni faqat strob orqali o'qiydi, bu esa kirish signalining vaqtinchalik jarayoni davomida ma'lumotlarni qabul qilishdan himoya qilishga imkon beradi.

3. To'liq e'tirof bilan. Ushbu rejim ko'pincha parallel kompyuter orqali boshqa kompyuter tizimi bilan ma'lumotlar almashish uchun ishlatiladi. Transmitter tomondan sinxronizatsiya signallaridan tashqari, qabul qiluvchi tomondan tasdiqlash signallari (keyingi almashishga tayyor) ishlatiladi. Bu sizga o'zaro ta'sir qiluvchi tomonlarning valyuta kursini boshqarish imkonini beradi va ulardan bittasi haddan tashqari yuklanganda ma'lumotlar yo'qolishini oldini oladi. Handshake portiga misol sifatida shaxsiy kompyuterning LPT portini olish mumkin. O'rnatilgan protsektor modullarida ushbu rejim ko'pincha dasturiy ta'minot va apparatda qo'llaniladi.

Chiqish portlari:

- ikki taktli chikish zanjiri bilan;
- bitta tsikliki chiqish zanjiri va ichki yuk bilan;
- ochiq chiqish bilan (ochiq kollektor yoki drenaj).

### **Ikki taktli chikish zanjiri bilan chiqish portlari**

Atmel AVR, Microchip PICmicro, AMD AM186, Motorola HC08, HC11 va boshqa ko'plab oilalarda ikki tartli chiqish portlari keng tarqalgan va amalga oshiriladi.

**Afzalliklar:**

• Sezilarli darajada maksimal kirish ("0" holatida) va chiquvchi ("1" holatida) chiqish oqimi;

• Normal ko'tarish qobiliyatiga ega bo'lgan bosqichlar uchun 2,.. 6mA (masalan, Fujitsu MB90) va 5,.. 30mA uchun yuk ko'tarish hajmi oshgan bosqichlar (masalan, PICmicro, AVR). Ultra yuqori yuk ko'tarish qobiliyatiga ega bo'lgan individual mikrosxemalar mavjud - 60,.. 90mA gacha (masalan, PIC17). Katta chiqish oqimi to'g'ridan-to'g'ri oyoqdan, kuchaytiruvchi va signalni konditsionizatsiya qilmasdan, etarlicha kuchli yukni boshqarishga imkon beradi;

• LEDlar, o'rni, kuchli elektron kalit (tranzistor, tiristor). Bu qurilma diagrammasini ancha soddalashtiradi.

**Kamchiliklari:**

• Dasturlash paytida qo'shimcha ravishda "output enable" bit registrini boshqarish kerak;

- Quvvatni sezilarli darajada iste'mol qilish va almashtirish shovqini. Ikkinchisi, ayniqsa, o'tish tezligiga bog'liq. Kommutatsiya vaqtida oqimlarni cheklash uchun ba'zida maxsus sönümleme davrlari ishlatiladi. Biroq, ular port ishlashini pasaytiradi. Söndürme davrlari, ayniqsa yuqori tezligi tufayli FPGA portlarida eng ko'p ishlatiladi;

- Nisbatan murakkab ichki zanjir, umuman mikrosxemaning murakkabligi va narxini oshiradi. Biroq, hozirgi bosqichda, mikrosxemalar ishlab chiqarish texnologiyasining muvaffaqiyati tufayli, bu endi muammo emas.

### **Bitta taktlli chiqish zanjiri va ichki yukli portlar**

Bitta taktlli chiqish zanjiri va ichki yukli portlar, masalan, MCS-51 oilasida qo'llaniladi. Ular oddiyroq ichki sxemaga ega.

#### **Afzalliklari:**

- Faqat bitta registrni boshqarish kerak;
- Oddiy sxema;
- Bir nechta tashqi chiqindilarni qo'shimcha sxemalarsiz bitta tashqi shinaga ulashni tashkil qilish imkoniyati. Yarim ikki yo'nalishli I / U portini qurish oson (quyida ko'rib chiqing).

**Kamchiliklari:** kichik qarshilik oqimi ("1" holatida), RL qarshiligi bilan cheklangan - yuzlar uA. Bu qo'shimcha kuchaytirish bosqichlarisiz nisbatan kuchli yuklarni boshqarishga imkon bermaydi yoki "0" ("nol boshqaruv") qiymatiga ega bo'lgan signal faolligini ta'minlashni talab qiladi.

**Ochiq chiqishi bilan chiqish portlari** (ochiq kollektor yoki drenaj).

Ular ko'plab mikroprotsessorlar oilalarida qo'llaniladi, masalan, AMD Am186, PICmicro. Chiqish bosqichi tashqi yuk bilan bitta uchli sxema bo'yicha qurilgan. Amaliyot printsipti bir martalik chiqish bosqichiga o'xshaydi.

#### **Afzalliklari:**

- Vcc ext tashqi quvvat manbai har xil bo'lishi mumkin - mikroprotsessor quvvat manбайдan yuqori yoki past. Bu 3.3V va 5V kabi har xil mantiqiy "1" darajadagi interfeyslarni o'chirish uchun qulay bo'lishi mumkin. Agar tashqi kuchlanish etarlicha yuqori bo'lsa, yuqori kuchlanish yuki to'g'ridan-to'g'ri boshqarilishi mumkin.

- Faqat bitta registrni boshqarish kerak;
- Oddiy sxema;

• Bir nechta tashqi chiqindilarni qo‘shimcha sxemalarsiz bitta tashqi avtobusga ulashni tashkil qilish imkoniyati. Bunday holda siz talab qilinadigan qarshilik RL ni tanlashingiz mumkin, masalan, I2C standarti qarshilikni 2,2 kOm bo‘lishini talab qiladi. Kvazi ikki yo‘nalishli I / U portini qurish oson (pastga qarang).

#### **Kamchiliklari:**

- Tashqi yukni talab qiladi;
- Tashqi tortishish qarshiligi bilan cheklangan past oqim oqimi ("1" holatida).

#### **Ikki tomonlama portlar va alternativ funktsiyali portlari**

Ikki tomonlama portning eng oddiy sxemasi-bir taktli chiqish kaskadi bilan chiqish portiga o‘xshash sxemaga ega bo‘lgan kvazidvun-yo‘naltirilgan port.

MCS-51 oilasida yo‘naltirilgan port sxemasi ishlatiladi. Qo‘shimcha chiqish kaskadli ikki tomonlama portli kontaktlarning sxemasi tez-tez ishlatiladi. Kirish porti va chiqish portining kontakt zanjirini push-pull chiqish pallasida birlashtiradi. Portni I / O rejimiga o‘tkazish "kirish / chiqish" registriga "1" yozib qo‘yish orqali amalga oshiriladi. Bunday holda (chiqish portining tavsifida ko‘rsatilganidek) ikkala tranzistor ham yopiq holatga aylanadi va chiqish porti kirish signaliga ta'sir qilmaydi. qarshilik, ikki tomonlama portlari -up torting va pull- pastga faqat kiritish nazorat qilish davri tomonidan kiritish rejimiga Ro‘yxatdan o‘tish »Input / output" ("1" - Kirish) tegishli ishlab chiqarish bilan bog‘laydi bog‘liq.

I/O portining funktsiyasini bajarish bilan bir qatorda, mikrosxemaning tashqi chiqishi mikroprosessorning ichki periferik modullari bilan, shuningdek protsessor yadrosi, xotira va boshqarish zanjirlarining quyi tizimlari (uzilish kontrolleri, tashqi xotira interfeysi bloki va boshqalar) bilan aloqa qilish uchun ishlatilishi mumkin. Ushbu funktsiyalar alternativ deb ataladi. Odatda, alternativ funktsiyani bajarish uchun port pimi ishlatilganda, asosiy kontaktlarning zanglashiga kirish holati qo‘yiladi yoki umuman o‘chirib qo‘yiladi.

#### **Taymer hisoblagichlari**

Taymer hisoblagichlari quyidagilar uchun mo‘ljallangan:

- Vaqt oralig‘ini hisoblash (taymer rejimi);
- Maxsus tashqi kirish (taymer rejimi) da impulslar sonini ("tashqi hodisalar") hisoblash.

Taymer rejimida taymerni ushlab turishi protsessorining ichki sinxronlash signalidan amalga oshiriladi. Odatda bu asosiy generator tomonidan ishlab chiqarilgan protsessor tsikllarining chastotasi. Vaqt oralig'ini hisoblash Fint signal davrida amalga oshiriladi. Qabul qilgich taymerga berilgan soat tezligini kamaytirish uchun ishlatiladi. Bu sizga uzoq vaqt oralig'ida hisoblash imkonini beradi, lekin namuna olish bosqichini oshiradi va shuning uchun aniqlikni pasaytiradi. Ajratuvchi sobit yoki programlanadigan bo'linish koeffitsienti bilan bo'lishi mumkin. Dasturlashtiriladigan ajratuvchi odatda 1, 2, 4, 8, bir qator bo'linish koeffitsiyenti tanlangan ro'yxatga olishi mumkin. Ro'yxatdan o'tish hisoblagichining kengligi butun taymer hisoblagichining kengligini aniqlaydi.

Taymer rejimidan farqli o'laroq, hisoblagich rejimi INcnt kirishiga qo'llaniladigan tashqi impuls signalidan soatni tanlaydi. Bunday holda, tashqi signalning impulslari hisoblanadi. Hisoblagichning o'sishi yoki kamayishi signalning chetida sodir bo'ladi. Bu holda signalning old tomoni "tashqi hodisa" deb nomlanadi. Qirralarning kutupliligi dasturlashtirilishi mumkin. Aks holda, hisoblagich va taymer rejimlarida ishlash bir xil.

### **Analog-raqamli ozgartirgich**

**Analog-raqamli konvertor (ADC)** moduli fizik kattalikdagi sensorlardan analog signallarni protsessorga kiritish va ushbu signallarning kuchlanish qiymatlarini dasturiy ta'minotni qayta ishlash uchun ikkilik kodga aylantirish uchun mo'ljallangan.

Eng oddiy bir bitli ikkilik ADC - bu komparator. Xususiyatlari:

- ADC o'lchamlari - ushbu ADC tomonidan konvertatsiya qilinishi mumkin bo'lgan analog signal qiymatining minimal o'zgarishi. Odatda volt bilan o'lchanadi, chunki kuchlanish ko'pgina ADClar uchun kirish signalidir;
- ADC quvvati konvertor chiqishi mumkin bo'lgan diskret qiymatlar sonini tavsiflaydi;
- diskretlash chastotasi;
- aniqlik;
- konversiya tezligi.

Analog signal vaqtning doimiy funktsiyasi bo'lib, ADCda u raqamli qiymatlar ketma-ketligiga aylanadi. Shuning uchun analog signaldan raqamli qiymatlarni tanlash tezligini aniqlash kerak. Raqamli qiymatlarni ishlab chiqarish chastotasi ADC diskretlash chastotasi deb

ataladi. Cheklangan spektral diapazonga ega doimiy ravishda o'zgarib turadigan signal raqamlashtiriladi (ya'ni signal qiymatlari vaqt oralig'ida T - diskretlash davri davomida o'lchanadi) va asl signal vaqt oralig'ida diskret qiymatlardan aniq qayta tiklanishi mumkin. Qayta qurish aniqligi kvantlash xatosi bilan cheklanadi.

Biroq, Kotelnikov-Shannon teoremasiga ko'ra, aniq qayta qurish faqat diskretlash chastotasi signal spektridagi maksimal chastotaning ikki baravaridan yuqori bo'lgan taqdirdagina mumkin bo'ladi. Haqiqiy ADClar bir zumda analog-raqamli konversiyani amalga oshira olmasligi sababli, analog kirish qiymati kamida konversiya jarayonining boshidan oxirigacha doimiy ravishda ushlab turilishi kerak (bu vaqt oralig'i konversiya vaqti deb ataladi). Ushbu muammo ADC kirishida maxsus namuna olish-saqlash moslamasi (OSM) sxemani qo'llash orqali hal qilinadi.

OSM, qoida tariqasida, analog voltaj orqali kirishga ulangan kondansatkichda kirish voltajini saqlaydi:

- kalit yopilganda, kirish signali tanlanadi (kondansator kirish voltajiga qadar quvvatlanadi);
- ochilgandan so'ng - saqlash.

Integral mikrosxemalar shaklida ishlab chiqarilgan ko'plab ADClarda ichki OSM mavjud. Konvertatsiya natijasida olingan qiymat ma'lumotlar registriga (RD) yoziladi. Protsektor kristaliga birlashtirilgan ADClar odatda ketma-ket taxminiy sxemada quriladi. Konversiya vaqtlari odatda ADC soat tezligiga qarab bir necha o'n mikrosaniyadir. Konvertatsiya jarayonining tugashi Fatsp bayrog'ini o'rnatish bilan belgilanadi va (agar u yoqilgan bo'lsa) uzilish so'rovi paydo bo'ladi.

Zamonaviy boshqaruv protsektorlari va mikrokontrollerlarda eng keng tarqalgan ADC 8, 10, kamroq 12 va juda kamdan-kam hollarda 14 va 16 bit tashkil qiladi. Analog kalit, mumkin bo'lgan analog kirish (chiqish) dan birini tanlaydi va konvertatsiya qilish uchun uni ichki ADC kirishiga ulaydi. Kanallarning ketma-ket tanlovi ko'p kanalli ADC taqlidini yaratadi. Haqiqiy ko'p kanalli ADC-lardan foydalanish protsektorning quvvat sarfini va narxini keskin oshiradi va odatda foydalanilmaydi (agar bir nechta kanal va yuqori konversiya tezligi talab etilsa, u holda tashqi ADC chipidan foydalaniladi). Kanallarni tanlash kodini dasturiy ta'minot yaratishi mumkin, ya'ni dasturchi "qo'lda" kanallarni almashtiradi yoki qo'shimcha (avtomatik), kanallarni ketma-ket saralash (ko'rish rejimi).

ADC modulidan skanerlash rejimida foydalanish uchun yanada qulayroq bo'lish uchun har bir kanal uchun bir nechta ma'lumotlar registrilarini (Fujitsu MB90, Intel 8051GB) amalga oshirish mumkin. Dasturchi uchun registrdan kerakli kanalga mos keladigan ma'lumotlarni o'qish kifoya qiladi. Bunday holda kanalni tanlash kodi ma'lumotlar registri blokining manzil yozuvlariga parallel ravishda beriladi.

Yo'naltiruvchi kuchlanish  $V_{ref}$  analog kirishlardagi kuchlanish qiymatlari oralig'ini aniqlaydi va ADCning o'lchamlari  $V_{ref} / 2^n$  ga teng, bu erda  $n$  ADC ning hajmi o'lchamlari. Agar kirish voltajining qiymati katta bo'lmasa,  $V_{ref}$ ni pasaytirish orqali konversiya aniqligini oshirish mumkin.  $V_{ref}$ - ning ruxsat etilgan qiymatlari oralig'i odatda protsessorning kuchlanish qiymatiga kiradi.

Ma'lumot manbalarining quyidagi turlaridan foydalanish mumkin:

1. Maxsus mikrosxemaviy pinlar orqali ulangan tashqi manbalar;
2. Ichki yoki dasturlashtiriladigan ( o'rnatilgan DAC- dan foydalangan holda ).

$V_{ref}$  kaliti yordamida tashqi yoki ichki manbalar ADC ga ulanadi. ADC-ni ishga tushirish signalini o'zgartirish tugmachasi ishlash jarayonini boshlash usulini tanlashga imkon beradi, shuningdek, mumkin bo'lgan ADC ish rejimlaridan birini aniqlaydi:

1. O'zgarish davri. Ushbu rejimda ADC asosiy soat yoki o'rnatilgan taymerdan davriy signal bilan ishga tushiriladi.

2. Agar boshlang'ich signal analogli o'tish moslamasi va kirish registrleri bloklari ma'lumotlarini ro'yxatga olish blokining manzil satrlariga ulangan ikkilik hisoblagichga qo'llanilsa, kanallarni ketma-ket ko'rish rejimini amalga oshirish juda oson.

3. Tashqi boshlanish. Tashqi signal orqali amalga oshiriladi, bu analog kuchlanish qiymatini o'qishdan o'qish momentini aniq aniqlashga imkon beradi.

4. Maxsus bitni o'rnatish orqali dasturiy ta'minot tomonidan boshqariladigan ishga tushirish. ADC modulini boshqarish bloki boshqa (yuqoridagi) birliklarning ishlashini sozlaydi va sinxronizatsiya qiladi, dasturiy ta'minot yordamida, maxsus registrlar orqali boshqariladi.

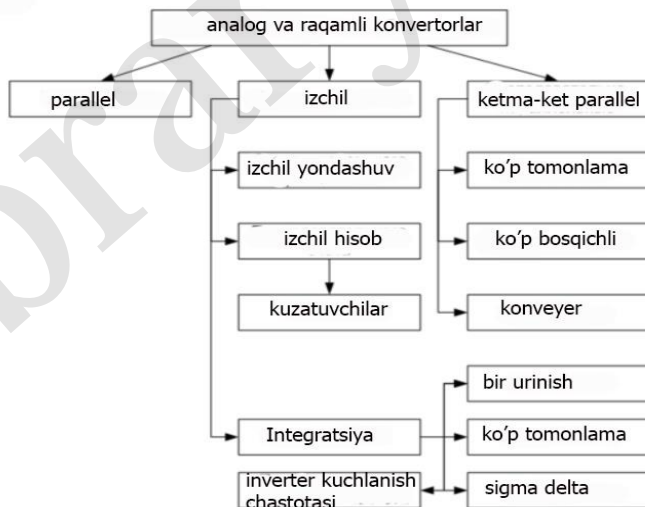
### **Analog taqqoslagich**

Analog taqqoslagich ikki tashqi analog signalning kuchlanishini taqqoslash yoki tashqi analog signalning kuchlanishini protsessor ichida ishlab chiqarilgan namunali kuchlanish bilan solishtirish uchun ishlatiladi. Namuna kuchlanishining turli darajalari dasturlashtirilishi

mumkin. Taqqoslash natijasi maxsus maqsadli registrda bit bilan kodlangan, masalan, "1" – kirish A B dan kattaroq yoki teng, "0" – kirish A B dan kam nisbati o'zgarganda, bitning qiymati o'zgaradi va bayroq o'rnatilishi va uzilish so'rovi ishlab chiqilishi mumkin. Analog kirish kaliti taqqoslash uchun analog signallarni tanlaydi. Bir signal tashqi AIN1 usulidan olinadi, ikkinchisi AIN 2 tashqi kirish signalidan yoki DAC tomonidan ishlab chiqarilgan namunali ichki kuchlanishdan olinadi.

### ADC tasnifi

Hozirgi vaqtda voltajdan kodga o'tkazish usullarining ko'pligi ma'lum. Ushbu usullar bir-biridan potentsial aniqligi, konversion tezligi va apparatni amalga oshirishning murakkabligi bilan sezilarli darajada farq qiladi. ADC tasnifi (3.5-rasm) analog qiymatni raqamli qiymatga aylantirish jarayoni vaqt o'tishi bilan qanday rivojlanishini ko'rsatuvchi belgiga asoslangan. Namuna olingan signal qiymatlarini raqamli ekvivalentlarga aylantirish kvantlash va kodlash operatsiyalariga asoslanadi. Ular raqamli ekvivalentni konvertatsiya qilingan qiymatga yaqinlashtirish uchun ketma-ket yoki parallel yoki ketma-ket parallel protseduralar yordamida amalga oshirilishi mumkin.

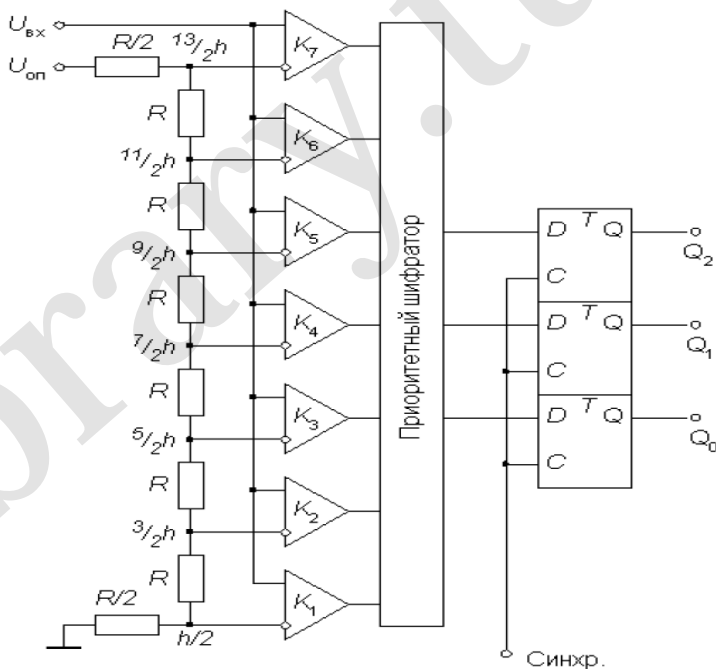


Rasm 3.5. Konversiya usullari bo'yicha analog-raqamli o'zgartirgich (ARO) tasnifi



### Parallel ADC lar

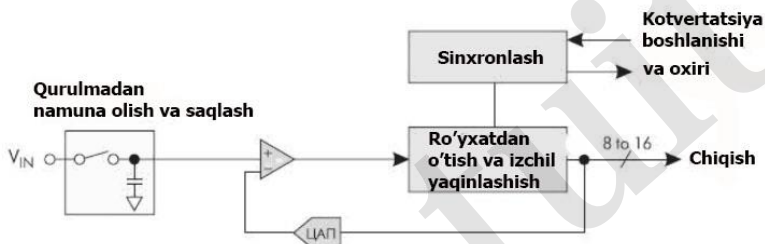
Ushbu turdagi ADC signalni kirish signal manbasiga parallel ravishda ulangan taqqoslagichlar to'plamidan foydalangan holda bir vaqtning o'zida o'lchaydi. 3.6-rasmda 3-bitli raqam uchun ADC-ni parallel ravishda o'tkazish usulining bajarilishi ko'rsatilgan. Uchta ikkilik raqamlardan foydalanib, nolni o'z ichiga olgan sakkiz xil raqamlarni ko'rsatish mumkin. Shuning uchun ettita taqqoslagich kerak. Rezistiv bo'linuvchi yordamida ettita mos keladigan mos yozuvlar zo'riqishida hosil bo'ladi. Agar qo'llaniladigan kirish voltaji  $5/2h$  dan  $7/2h$  oralig'iga tushmasa, bu erda  $h = U_{op} / 7$  - ADC ning eng past tartibli birlikiga mos keladigan kirish voltajining kvanti bo'lsa, unda 1 dan 3 gacha bo'lgan taqqoslovchilar o'rnatiladi 1 holatiga va 4-dan 7-gacha bo'lgan taqqoslovchilar 0 holatiga. Ushbu kodlar guruhini uch xonali ikkilik raqamiga aylantirish ustuvor kodlovchi deb nomlangan mantiqiy qurilma tomonidan amalga oshiriladi.



3.6-rasm. Parallel ADC shemasi

## Ketma-ket yaqinlashish ADC

Keyingi ketma-ket yaqinlashish ADC yoki bit tomon muvozanatli analog-raqamli konvertor (ADC) tarkibida taqqoslagich, yordamchi DAC va ketma-ket yaqinlashish registrarlari mavjud. ADC analog signalni  $N$  bosqichida raqamli signalga o'zgartiradi, bu erda  $N$  - ADC bit chuqurligi (hajmi). Har bir qadamda eng muhim bit (MSB) dan boshlab va eng kam bit (LSB) bilan tugaydigan kerakli raqamli qiymatning biti aniqlanadi. Keyingi bitni aniqlash bo'yicha harakatlar ketma-ketligi quyidagicha.



3.7-rasm. Tarkibiy ARO‘

Yordamchi DACda oldingi bosqichlarda allaqachon aniqlangan bitlardan hosil bo‘lgan analog qiymat o‘rnatiladi; ushbu bosqichda aniqlanishi kerak bo‘lgan bit 1 ga, pastki bitlar 0 ga o‘rnatiladi. Yordamchi DACda olingan qiymat kirish analog qiymati bilan taqqoslanadi. Agar kirish signalining qiymati yordamchi DAC qiymatidan katta bo‘lsa, aniqlangan bit 1 qiymatini oladi, aks holda 0. Shunday qilib, yakuniy raqamli qiymatni aniqlash ikkilik qidiruvga o‘xshaydi. Ushbu turdagi ADClar ham yuqori tezlikda, ham yaxshi piksellar soniga ega. Biroq, saqlash namunasini olish moslamasi bo‘lmagan taqdirda, xato ancha katta bo‘ladi Tegishli yaqinlashish ADC xarakteristikasi: past konversiya tezligi, past narx va kam quvvat iste‘moli.

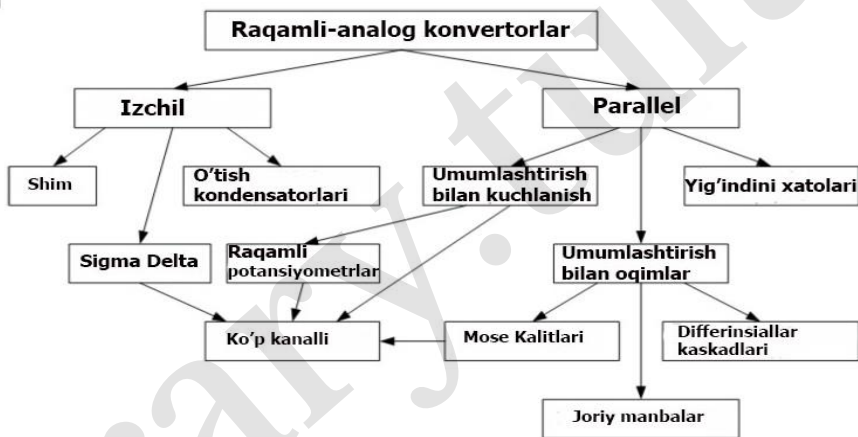
## Raqamli-analog o‘zgartirgich (DAC)

Raqamli-analog o‘zgartirgich (konvertor) raqamni, odatda, ikkilik kod shaklida, ushbu raqamga mutanosib voltajga yoki oqimga aylantirish uchun mo‘ljallangan. Raqamli-analogli konvertorlarning sxemasi juda xilma-xildir. 3.8-rasmda kirish kodini va chiqish signalini

hosil qilish davrlarini konvertatsiya qilish usullari bo'yicha DAC ning umumiy tasnifi ko'rsatilgan.

Raqamli-analogli konvertorlarni keyingi tasniflash bir qator o'ziga xos xususiyatlarga ko'ra amalga oshirilishi mumkin, masalan :

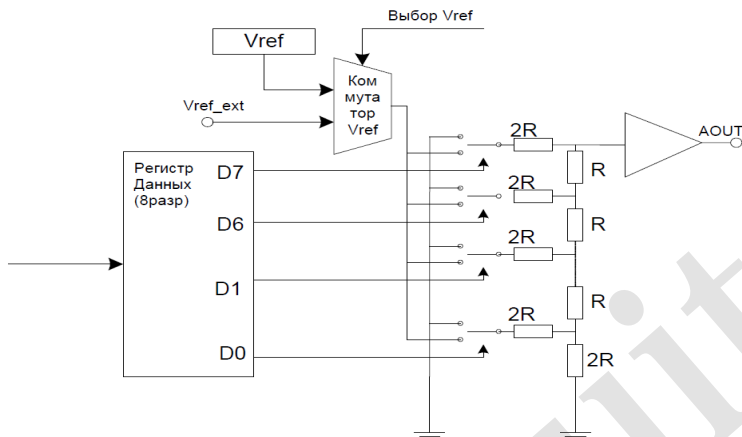
- chiqish signalining turi bo'yicha: oqim chiqishi yoki voltaji bo'lgan konvertorlar ;
- raqamli interfeys sifatida: ketma-ket kirish bilan yoki parallel kirish;
- chipdagi DAC soni bo'yicha: bitta kanalli va ko'p kanalli;
- tezlik: past, o'rta va yuqori tezlik;
- bit chuqurligi bo'yicha.



3.8-rasm. Umumiy DAC tasnifi

**R-2R** matritsasi eng keng tarqalgan raqamli-analogga o'tkazish usuli hisoblanadi. Matritsa kirish voltajini kirimlarga bo'lish printsipli asosida ishlaydi. Matritsada ma'lumotlar registrining bit soniga ko'ra kirish soni mavjud. Vref yoki 0V mos yozuvlar kuchlanishi kalit orqali har bir kirishga qo'llanilishi mumkin. Kalitlar ma'lumotlar registrining bitlari tomonidan boshqariladi: "1" - Vref matritsaga, "0" - 0V beriladi. Vref mos yozuvlar kaliti tashqi yoki ichki kuchlanish moslamasini tanlashga imkon beradi. Ma'lumotlar registriga raqamli kod yoziladi.

Ma'lumotlar registri DAC ning bit kengligini aniqlaydi.



3.9-rasm. DAC moduli "Matrix R-2R" transformatsiya turi bilan

Amalda, DAC turli xil aktuatorlar va tizimlarni boshqarish uchun ishlatiladi: o'zgaruvchan tezlikka ega motorlari, boshqariladigan kuchlanishli quvvat manbalari, turli xil ko'rsatkichlar va boshqalar. DAC-dan foydalanib, siz har xil analog signallarni sintez qilishingiz mumkin.

### **Ketma-ket interfeys kontrollerlari**

Ketma-ket tekshirgichlar quyidagi vazifalarni hal etishga yo'naltirilgan:

- O'rnatilgan mikroprotsessordagi yuqori darajadagi boshqaruv tizimi bilan aloqasi: sanoat yoki ofis kompyuterlari, programlanadigan kontroller. Ushbu maqsadlar uchun ko'pincha RS-232C, RS-422, USB, IrDA interfeyslari ishlatiladi.

- Mikroprotsessorga tashqi (EEPROM xotirasi, real vaqt soati (RTC) va boshqalar) tashqi, shuningdek ketma-ket raqamli chiqishi bo'lgan turli xil sensorlar bilan aloqa qilish. Ushbu maqsadlar uchun SPI, I2C, MicroWire, uLAN va boshqalar eng ko'p ishlatiladigan interfeyslardir.

- Tarqatilgan axborot va boshqaruv tizimlarida mahalliy tarmoq bilan aloqa interfeysi. Ushbu sohada RS-232C, RS-485, I2C, uLAN, CAN, Ethernet interfeyslaridan foydalaniladi.

- Ichki dasturlar uchun protsessorlarda doimiy dastur xotirasi (OTPROM, EPROM, FLASH) yoki ma'lumotlarning (EEPROM) tizimdagi dasturlash. Buning uchun odatda RS-232C interfeysi (ADuC (Analog Devices), MB90Fxxx (Fujitsu), MSP430 (Texas Instruments)) yoki SPI (AVR (Atmel)) ishlatiladi.

Hozirgi vaqtda deyarli barcha o'rnatilgan protsessorlarda o'rnatilgan ketma-ket (seriyali) tekshirgichlar mavjud, eng oddiy 8-16 pinli mikrosxemalar bundan mustasno. Ko'pgina protsessorlarda bir xil yoki har xil turdagi ushbu modullarning bir nechitasi mavjud. Seriyali almashinuv tekshirgichlari orasida amalda standart Universal Synchronous / Asynchronous Receiver and Transmitter (USART) moduli hisoblanadi.

UART ketma-ket portining xususiyatlari bosilgan elektron kartadan tashqarida ma'lumotlarni qabul qilish va uzatish imkoniyatini bermaydi. Boshqa qurilmalar bilan aloqa qilish uchun UART dan signal standartlardan birida ishlaydigan qabul qiluvchi-uzatgich orqali uzatilishi kerak:

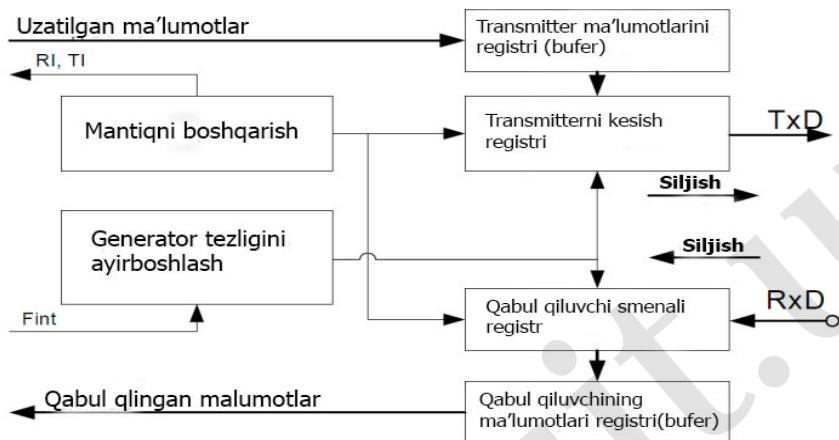
- RS-232;
- RS-485;
- RS-422.

Odatda, asenxron rejimdagi UART modullari RS-232 interfeysi (8N1 yoki 9N1) uchun aloqa protokolini qo'llab-quvvatlaydi; sinxron rejimda - nostandart sinxron protokollar, ba'zi hollarda - SPI protokoli. Transceiver - qoida tariqasida, ajralmas dizaynda yaratilgan darajadagi konvertor. Elektr signallarini TTL darajasidan ma'lum bir standartning jismoniy darajasiga mos keladigan darajaga o'tkazish uchun mo'ljallangan.

UART kontrolleri odatda quyidagilarni o'z ichiga oladi:

- Soat manbai.
- Kirish va chiqishni almashtirish registrlari.
- Ma'lumotlarni qabul qilish / uzatishni boshqarish registrlari; o'qish / yozish.
- Qabul qilish / uzatish buferlari.
- Buferlarni uzatish / qabul qilish uchun parallel ma'lumotlar shinasini.
- FIFO xotira buferlari (ixtiyoriy).

UART qabul qilgichining soddalashtirilgan tuzilishi 3.10-rasmda keltirilgan.



3.10-rasm. UART ning soddalashtirilgan tuzilishi

Tezlik generatori - bu Fint protsessorining ichki soat chastotasini bir tekis yoki pog'onali (diskret ravishda) bo'linish nisbati bilan ajratuvchi. "Silliq" dasturlash yordamida siz Fint chastotasidan qat'i nazar (ma'lum chegaralar ichida) kerakli tezlikni sozlashingiz mumkin. Buning uchun avtomatik qayta yuklash rejimida standart yoki maxsus ajratilgan taymer-hisoblagich ishlatiladi. "Ruxsat etilgan" bo'linish stavkalari holatida standart tezlikni saqlash uchun ma'lum bir protsessorning soat chastotasini tanlash kerak. Tezlik generatorining chiqishidan sinxronizatsiya signali qabul qiluvchi va uzatuvchi smenali registrnlarning soat kiritilishiga beriladi, ular ketma-ket ma'lumotlar bitlarini ma'lum tezlikda uzatadi / qabul qiladi. To'liq qabul qilingan bayt registrga kiradi - qabul qiluvchining ma'lumot buferi. Etkazish uchun bayt transmitter buferidan siljish registriga joylashtirilgan. Asenkron UART rejimida qabul qilish va uzatish jarayonlari mustaqil ravishda sodir bo'ladi. Shunday qilib, to'liq dupleks aloqa saqlanib qoladi. Shu bilan birga, qabul qilgich va uzatgich bir xil tezlikda o'rnatilishi talab qilinadi.

Sinxron rejimda ishlash osonroq. Bu erda har bir qabul qilingan / uzatilgan bit maxsus signal bilan yopiladi va qabul qilgich va uzatuvchining tezligini aniq moslashtirishga hojat yo'q.

UART xatolari:

- Haddan tashqari xato (uzatish tezligining oshishi, qabul qilish buferining oshib ketishi sababli xato). Ushbu xato UART qabul qiluvchisi kanaldan keladigan belgilarni qayta ishlashga ulgurmaganida, ya'ni bufer toshib ketganda yuzaga keladi.

- Kadr tuzishda xato. Ushbu xato, boshlang'ich yoki to'xtash bitini uzatishda ma'lumotlar liniyasining noto'g'ri holati qayd etilganda paydo bo'ladi. Masalan, 8 bitli ma'lumotlarni uzatgandan so'ng, qabul qilgich chiziqning to'xtash holatiga kirishini kutadi, ammo bu sodir bo'lmaydi.

- Tanaffus holati (uzatishni to'xtatish signali, aloqani uzish). Ushbu signal, kirish ma'lumotlari liniyasi bir belgining uzatilishidan uzoqroq vaqt davomida doimiy nol holatida bo'lganligi haqida xabar beradi. Qabul qilish buferi nol baytni o'z ichiga oladi. Ba'zi qurilmalar ushbu ketma-ketlikni transmitterga xabar berish uchun ishlatadi, masalan, tezlikni uzatish tezligini o'zgartirish uchun.

SPI rejimida ishlash yanada soddalashtirilgan: qabul qilgich va uzatuvchi sinxron ravishda ishlaydi: bitta bitni qabul qilish bir bitning uzatilishiga to'g'ri keladi, baytni uzatishni boshlanishi qabul qilish boshlanishiga to'g'ri keladi, bitta bayt qabul qilinadi va almashinuv sessiyasi davomida bitta bayt uzatiladi.

Ko'pgina hollarda transceiverlar TTL kirish va chiqish signallari bilan ishlaydi. Amalga oshirilgan interfeysga mos keladigan kuchlanish va oqim darajalariga ega bo'lgan jismoniy signallarni shakllantirish maxsus mikrosxemalar - transduserlar yoki fizik interfeys adapterlari yordamida amalga oshiriladi.

O'rnatilgan protsessorlarda ko'rib chiqilgan USART transversiyalaridan tashqari, boshqa interfeyslar, masalan, USB, CAN keng qo'llaniladi.

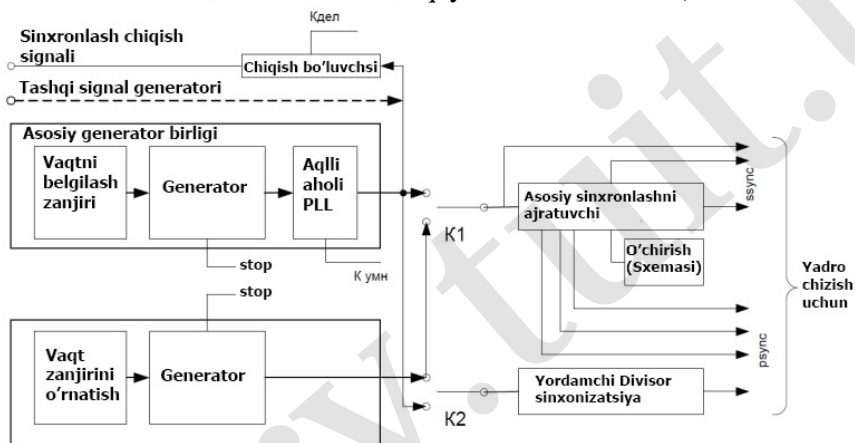
### **Sinxronizatsiya quyi tizimi**

Sinxronizatsiya quyi tizimi protsessorning ichki bloklari va ushbu protsessorga o'rnatilgan boshqaruv hisoblash tizimlarining tashqi sxemalarini (bloklarini) barqaror sinxronizatsiya signallarini Rasmlantirish uchun javobgardir.

Ichki birliklarga quyidagilar kiradi:

- Hisoblash yadrosi;
- Tashqi qurilmalar (taymerlar / hisoblagichlar, KNN, ARO', DAC, transversiya qurilmalari va boshqalar);
- Qayta boshlash sxemalari ("tiklash").

Tashqi shemalarni sinxronizatsiyasi uchun (periferik kontrolorler, interfeys chiplari, dasturlashtiriladigan mantiq bloklari va boshqalar) sinxronizatsiya signali protsessordan maxsus oyoqqa chiqariladi. Sinxronizatsiya quyi tizimi 3.11 ko'rsatilgan. Sinxronizatsiya quyi tizimiga soat generatorlari bloki (asosiy va yordamchi generatorlar, sinxronizatsia chiqish signalini shakllanish sxemasi) va protsessorni ichki sinxronizatsia chiqish signalini shakllanish soat generatori (K 1 va K2 kommutatorlari, bo'linuvchilar, "qayta o'rnatish" davri) kiradi.



Rasm 3.11. Sinxronizatsiya quyi tizimi

Asosiy generator bloki hisoblash yadrosi, ko'pgina periferik qurilmalar va qayta ishga tushirish pallasida sinxronizatsiya signallarini ishlab chiqaradi. Blok haqiqiy generator sxemasini, generator uchun tashqi yoki o'rnatilgan vaqt zanjirini va (har doim ham emas) chastota ko'paytirgich sxemasini o'z ichiga oladi.

Asosiy sinxronizatsiya chastotasi juda keng diapazonda o'zgarishi mumkin - o'nlab kilogertsdan o'nlab va yuzlab megagertslarga. Past chastotali (1 MGtsgacha) kam quvvatli tizimlarda ishlatiladi.

Tanlangan ish chastotasi, sinxronizatsiya signali parametrlarining aniqligi va barqarorligiga qarab, turli xil vaqt zanjirlari ishlatilishi mumkin, ular o'z navbatida generatorlarning ish rejimlarini qayta sozlashni talab qiladi. Generatorning ishlash rejimini va vaqt zanjiri turini tanlash mikroprotessor dasturlarining ichki xotirasida maxsus konfiguratsiya bitlarini dasturlash orqali amalga oshiriladi. Vaqtinchalik



zanjirlar maxsus mikroprosessor chiqishlariga ulanadi. Ba'zi modellarda o'rnatilgan zanjirlar mavjud.

Vaqt zanjirlarining eng ko'p ishlatiladigan turlari:

1. **Kvarts rezonatori:** o'nlab kilohertsdan o'nlab megagertsgacha bo'lgan chastotalar, yuqori chastotali barqarorlik (xato - foizning yuzdan / mingdan bir qismi), nisbatan yuqori narx;

2. **Pyezoseramik rezonator:** chastotalar o'nlab kilohertsdan megaherts birliklarga, o'rtacha chastota barqarorligi (xato - foizning o'ndan bir qismi), past narx;

3. **LC-kontur:** yuz kiloherts chastotalar, o'rtacha chastotaning barqarorligi (xato - o'ndan / foiz), past narx;

5. **RC-sxemasi:** bir yuz kiloherts chastotalari, past chastotali barqarorlik (xatolik bir necha foiz), arzon narx, ko'pincha o'rnatilgan vaqt zanjiri sifatida amalga oshiriladi;

Yuqori chastotali (30 MGts dan yuqori) hollarda o'rnatilgan generatorni to'liq ajratib, tashqi generatorini ulash tavsiya etiladi. Xuddi shu narsa, agar protsessorni o'z ichiga olgan bir nechta sxemalar bitta tashqi generatordan sinxronlashtirilsa. Generatörning yuqori chastotada ishlashi quyidagi qiyinchiliklarga olib keladi: o'rnatilgan generatorni "ishga tushirish" ning murakkabligi (30 MGts dan yuqori chastotalarda), elektron platalarning marshrutlashi va sifatiga qo'yiladigan maxsus talablar, yuqori darajadagi tashqi yuqori chastotali davrlarning shovqinlari (masalan, kvarts rezonatorini ulash uchun sxemalar), real vaqtda chastotani sozlashning iloji yo'qligi.

Ushbu muammolarni oldini olish uchun deyarli barcha 16/32 bitli protsessorlar programlanadigan raqamli chastotali multiplikatorlardan foydalanadilar. Bugungi kunda eng keng tarqalgan multiplikator bu fazali qulflangan pastadir sintezatoridir (PLL). Asosiy generatordan soat signali mikrosxemaning oyog'iga chiqadi va tashqi zanjirlarni soatlash uchun ishlatilishi mumkin. Chiqish signalining chastotasini kamaytirish uchun ushbu sxemada boshqariladigan yoki qattiq ajratuvchi ishlatilishi mumkin. Yordamchi generator bloki ba'zi periferik qurilmalarning, odatda hisoblagichlarning ( PICmicro, ATmega, Fujitsu MB90) vaqtlarini ta'minlaydi va ba'zi rejimlarda u asosiy generatorning funktsiyalarini bajarishi mumkin (yadroni sinxronizatsiya qilish, tashqi qurilmalarni sinxronlashtirish, "qayta o'rnatish" davrlari). Yordamchi generator odatda 1 MGts gacha chastotalarda ishlaydi. Agar real vaqt soatlari asosi sifatida foydalansa - 32768 Gts chastotada. Yordamchi generator asosiy generatorga o'xshash, ammo chastota multiplikatori

deyarli ishlatilmaydi. Asosiy va yordamchi generatorlarning chastota taqsimotlari ichki signal chizig'ining yadrosi hisoblanadi. Ularning chiqishlaridan asosiy soat va sinxronizatsiya sozlamalari va psync periferik modullarining sinxronizatsiya signallari olinadi. Odatda, barcha ichki signallar generatorlarning chastotasini sobit koeffitsientga bo'lish yo'li bilan olinadi, ammo ba'zi protsessorlarda bo'linish koeffitsientlari dasturlashtirilishi mumkin, masalan, agar kam quvvat sarflash rejimlarida yadro soat chastotasini kamaytirish kerak bo'lsa. K1 va K2 soat tugmachalari ichki protsessor sxemalarining soat manbaini tanlash uchun ishlatiladi: asosiy yoki yordamchi generator. Oddiy rejimda, quyi tizimlarning aksariyati asosiy generatordan sinxronlashtiriladi va yordamchi taymerlar, real vaqt soatlari yoki soatlar vaqtinchalik bazasi sifatida ishlatiladi. Asosiy generatorning ishlashida beqarorlik bo'lsa yoki kerak bo'lsa, past ish chastotalariga o'tish, masalan, energiyani tejash rejimlarida, siz asosiy generator bo'linuvchisining kirishini yordamchi generatorning chiqishiga ulashingiz mumkin.

### **Ichki xotirani ishga tushirish mexanizmlari**

Dastlabki ishga tushirish (yuklash) mexanizmlari dastur kodini, ma'lumotlarni yoki konfiguratsiya parametrlarini protsessor yoki bitta chipli mikrokompyuterning doimiy o'zgarmas xotirasiga yozishni ta'minlaydi. Dastlabki ishga tushirish jarayoni "yalang'och" uskunalar bilan ishlashni nazarda tutadi, ya'ni, protsessorning ish rejimida ishlaydigan biron bir dasturi yordamisiz. Yozilgan dastur kodi yuqori darajadagi yuklovchi yoki dasturning o'zi. Ma'lumotlar, odatda, ish parametrlarining dastlabki qiymatlari (o'rnatish nuqtalari).

Konfiguratsiya parametrlari protsessor apparati ish rejimlarini sozlaydi. Bular quyidagilardan iborat bo'lishi mumkin:

- generator turi (kvarts, piezoelektrik rezonator, LC loy va RC);
- elektr ta'minoti uzilib qolgan taqdirda, qayta tiklash quyi tizimlaridan foydalanilgan, quvvatni avtomatik ravishda tiklash ( Power On Reset );
- qo'riqchi taymeridan foydalanish ( Watch Dog Timer );
- ichki xotirani ruxsatsiz nusxalashdan himoya qiladigan bayroqlar ;
- tashqi xotira avtobusining ishlatilishi va hajmi;
- dasturni ishga tushirish manzili (vektorni tiklash).

Boshlang'ich ishga tushirish mexanizmlarining tasnifi 3.12-rasmda keltirilgan. O'rnatilgan yuklagich dasturi (Bootsrap loader) - bu protsessorni ROM-ning o'rnatilgan dastur xotirasining maxsus blokiga

tayyorlash paytida yozilgan maxsus dastur. Amalga oshirilgandan so‘ng, bootstrap loader ketma-ket port (odatda UART porti) orqali yoziladigan dasturni yoki ma'lumotlarni qabul qiladi va protsessor xotirasiga yozadi.



Rasm 3.12. Ichki xotirani boshlash mexanizmlari

Yuklab olingan dasturni saqlash uchun bir nechta variant mavjud:

1. Dastur RAMga yuklanadi va shu zahotiy oq boshqaruv unga o‘tkaziladi. Bu yuklash vositasi bo‘lishi kerak, u o‘z navbatida dasturni qabul qiladi va ichki yoki tashqi ROMga yozadi (odatda bu FLASH xotirasi). Qayta ishga tushirilgandan so‘ng boshqaruv dasturga o‘tkaziladi.

2. Dastur to‘g‘ridan-to‘g‘ri ichki ROMga yoziladi va normal rejimda qayta ishga tushirgandan so‘ng bajarishni boshlaydi. Yuklangan dastur sifatida yuklovchi yoki maqsad kodi harakat qilishi mumkin.

### Nazorat savollari:

1. O‘rnatilgan platformaning asosiy xususiyatlarini sanab bering.
2. O‘rnatilgan tizimlar uchun mikroprotsessor texnologiyasining elementlar bazasiga nimalar kiradi.
3. Protsessorni qaysi mezonlariga ko‘ra tasnifni amalga oshirilishi mumkin?
4. Mikrokontrolorlarni tasniflash mezonlari qanday?

5. Dasturlashtiriladigan mantiqiy integral mikroshemalarning xususiyatlari.
6. Dasturlashtiriladigan mantiqiy matritsalarining afzalliklarini sanab bering.
7. Qurilmaning apparat yoki dasturiy ta'minotini amalga oshirish uchun umumiy mezonlar?
8. Tizimning asosiy raqamli tizim birliklarini.
9. SoC asosida qurilgan o'rnatilgan tizimning interfeyslari va boshqaruvchilarini sanab o'ring.
10. O'rnatilgan tizimlar uchun tipik protsessor tuzilishi.
11. Protsessorning almashtiriladigan funktsional blokiga nimalar kiradi ?
12. Maqsad va amalga oshirish xususiyatlariga qarab xotira qurilmalarini tayinlashni bajaring.
13. Portlarning asosiy turlari va ularning xususiyatlarini sanab bering.
14. O'rnatilgan tizimlar tarkibiy qismlarini qurishda qanday tushunchalardan foydalanish kerak?
15. Murakkab O'rnatilgan tizimlar uchun istiqbolli loyiha jarayonining afzalliklari va kamchiliklari.
16. Taymerlarining asosiy funktsiyalarini sanab bering.
17. O'rnatilgan tizimlarni loyihalashda modellashtirish qanday vazifalarni hal qilishga mo'ljallangan?
18. ADC va DACni qanday mezonlar asosida tasniflash mumkin?
19. Loyiha jarayonining qaysi elementlari arxitektura platformasi bilan birlashtirilgan.
20. O'rnatilgan tizimlar arxitekturasining loyiha mezonlarini sanab bering.
21. O'rnatilgan tizimlarni loyihalash jarayonlarining shablonlarini sanab bering.

## **4-BOB. O‘RNATILGAN TIZIMLARNING TARMOQ INTERFEYSLARI**

Ushbu bo‘lim zamonaviy o‘rnatilgan tizimlarda o‘rnatilgan tizimning o‘zi yoki yuqori boshqaruv darajasidagi boshqa axborot tizimlari o‘rtasida aloqa kanalini tashkil qilish uchun ishlatiladigan simli va simsiz, mahalliy va global interfeyslarni ko‘rib chiqishga bag‘ishlangan.

### **4.1. I<sup>2</sup>C seriyali interfeysi**

Maishiy texnika, telekommunikatsiya uskunalari va sanoat elektronikalarida shunga o‘xshash echimlar ko‘pincha bir-biriga bog‘liq bo‘lmagan ko‘rinadi. Masalan, deyarli har bir tizim quyidagilarni o‘z ichiga oladi:

- Ba’zi "aqlli" boshqaruv tugunlari, odatda bitta chipli mikrokompyuterdir.
- LCD buferlar, kirish / chiqish portlari, RAM, EEPROM yoki ma’lumot konvertorlari kabi umumiy maqsadlar uchun tugunlar.
- Raqamli sozlash va radio va video tizimlari uchun signallarga ishlov berish davrlari yoki telefoniya uchun ovozli terish generatorlari kabi maxsus tugunlar.

Ushbu umumiy echimlardan dizaynerlar va ishlab chiqaruvchilar (texnologlar) foydasiga foydalanish, shuningdek apparat samaradorligini oshirish va elektron tizimni soddalashtirish uchun Philips 1980 yilda samarali "inter-IC" boshqaruv uchun oddiy ikki yo‘nalishli ikki simli shina ishlab chiqardi. Shina Inter-Integrated Circuit, yoki IIC (I<sup>2</sup>C) shinasi deb nomlanadi. Hozirgi vaqtda Philips mahsulot portfelida 150 ta CMOS va elektron uskunalarining yuqoridagi barcha uch toifalarida ishlash uchun ishlab chiqilgan bipolyar I<sup>2</sup>C mos keluvchi qurilmalar mavjud. I<sup>2</sup>C-ga mos keladigan barcha qurilmalar o‘rnatilgan interfeysga ega, bu ularga I<sup>2</sup>C shinasi orqali bir-biri bilan aloqa o‘rnatishga imkon beradi. Ushbu loyiha echimi odatda raqamli tizimni loyihalashda yuzaga keladigan ko‘plab qurilmalar interfeysi muammolarini hal qiladi.

I<sup>2</sup>C shinaning asosiy ishlash tartibi 100 kbit / s; tezlikni pasaytirish rejimida 10 kbit / s. Shuni esda tutingki, standart nolgacha bo‘lgan chastota bilan soatni taqsimlashga imkon beradi. I<sup>2</sup>C qurilmalariga murojaat qilish uchun 7 bit ishlatiladi. 1992 yilda standartni qayta ko‘rib chiqqandan so‘ng, bitta shinaga yanada ko‘proq qurilmalarni ulash

mumkin (10 bitli manzilga kirish imkoniyati tufayli), shuningdek yuqori tezlikda ishlaydigan rejimda 400 kbit / s gacha yuqori tezlik. Shunga ko'ra, bo'sh tugunlarning mavjud soni 1008 taga etdi. Bitta shingaga ulangan mikroshemalarning maksimal soni 400 pF maksimal shingaga sig'imi bilan cheklangan.

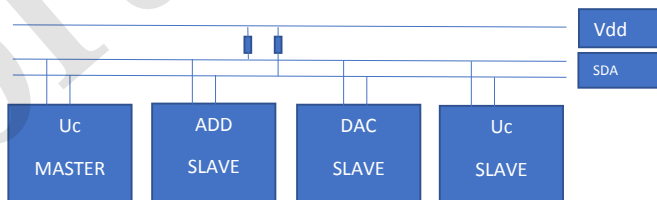
1998 yilda ishlab chiqarilgan standartning 2.0 versiyasida energiya sarfi kamaygan holda 3,4 Mbit / s gacha tezlikda yuqori tezlikda ishlash rejimi joriy etildi.

### I<sup>2</sup>C ni ishlatish mumkin bo'lgan ro'yxat:

- xotira modullariga kirish (RAM, EEPROM, Flash va boshqalar);
- past tezlikli DAC / ADC-ga kirish;
- real vaqt soati (RTC) bilan ishlash;
- monitorlarning kontrasti, to'yinganligi va ranglar muvozanatini sozlash;
- aqlli ovoz chiqargichlarni (karnaylarni) boshqarish;
- LCD-ni boshqarish, shu jumladan uyali telefonlarda;
- uskunalarni, masalan, markaziy protsessorning termostatini yoki protsessor sovutish foniyining aylanish tezligi sensori monitoringi va diagnostikasi uchun sensorlardan ma'lumotni o'qish;
- mikrokontrolörler o'rtasida ma'lumot almashish.

I<sup>2</sup>C ikkita ikki yo'nalishli drenaj liniyalaridan foydalanadi - ketma-ket ma'lumotlar liniyasi (SDA) va ketma-ket soat chizig'i (SCL), ikkalasi ham rezistorlar bilan to'ldirilgan.

### I<sup>2</sup>C shingaga kontseptsiyasi



Rasm.4.1. I<sup>2</sup>C shingadagi ulanish moslamalariga misol: bitta etakchi - mikrokontroller, uchta qul moslama - ADC, DAC, MK

Maksimal kuchlanish + 5V, + 3.3V ko'pincha ishlatiladi, ammo boshqa kuchlanishlarga ham ruxsat beriladi (kamida + 2V). I<sup>2</sup>C shingasi

har qanday IC texnologiyasini qoʻllab-quvvatlaydi. Har bir qurilma oʻziga xos manzil - mikrokontroller, LCD buferi, xotira yoki klaviatura interfeysi bilan tan olinadi va qurilmaning maqsadiga qarab transmitter yoki qabul qiluvchi vazifasini bajarishi mumkin. Odatda, LCD bufer faqat qabul qiluvchidir va xotira ma'lumotlarni qabul qilishi va uzatishi mumkin. Bundan tashqari, ma'lumotlarni uzatishda qurilmalar ustalar va qullar deb tasniflanishi mumkin. Etakchi bu ma'lumotlarni uzatishni boshlaydigan va sinxronizatsiya signallarini yaratadigan qurilma. Bundan tashqari, har qanday murojaat qilingan usta xoʻjayinga nisbatan qul deb hisoblanadi. Klassik manzillash 16 ta zaxiralangan manzilga ega boʻlgan 7 bitli manzil maydonini oʻz ichiga oladi. Bu bitta shinaga qoʻshimcha qurilmalarni ulash uchun 112 tagacha bepul manzillarni anglatadi.

Shinaga bir nechta mikrokontrollerlarni ulash imkoniyati bir nechta master bir vaqtning oʻzida yoʻnaltirishni boshlashga urinishi mumkinligini anglatadi. Bunday holda yuzaga kelishi mumkin boʻlgan tartibsizliklarni bartaraf etish uchun hakamlik muhokamasi ishlab chiqilgan. Ushbu protsedura barcha I<sup>2</sup>C qurilmalari shinaga I oʻrnatish qoidasiga muvofiq ulanganligiga asoslanadi.

Soatni yaratish har doim xoʻjayinning javobgarligi hisoblanadi; ma'lumotni shinaga yuborishda har bir usta oʻz sinxronizatsiya signalini yaratadi. Sinxronizatsiya signalini faqat sekin qul tomonidan tortib olinsa (chiziqni past ushlab turish bilan) yoki toʻqnashuv yuzaga kelganda boshqa usta tomonidan oʻzgartirilishi mumkin.

Shinaga ulanishi mumkin boʻlgan turli xil chip texnologiyalari (CMOS, bipolyar) tufayli, mantiqiy nol (LOW) va mantiqiy birlik (HIGH) darajalari oʻrnatilmagan va tegishli Vdd darajasiga bogʻliq. Har bir uzatiladigan bit uchun bitta sinhroimpuls ishlab chiqariladi.

SDA liniyasida ma'lumotlar HIGH sinxronizatsiya impulsi soat davomida barqaror boʻlishi kerak. Sinxronizatsiya liniyasi LOW holatida boʻlsa, ma'lumotlar liniyasining HIGH yoki LOW holati oʻzgarishi kerak. SDA liniyasida ma'lumotlar baytlarda uzatiladi, har bayt tasdiqlash biti bilan tugaydi. Har bir sessiya uchun oʻtkaziladigan baytlar soni cheklanmagan. Ma'lumotlar eng katta bitdan boshlab uzatiladi. Agar qabul qilgich boshqa funktsiyani bajarmaguncha boshqa butun baytni ololmasa (masalan, u ichki uzilishni amalga oshirsa), u SCL chizigʻini LOW holatida ushlab turishi mumkin, transmitterni kutish holatiga qoʻyib. Qabul qilgich keyingi baytga tayyor boʻlganda va SCL chizigʻini chiqarganida ma'lumotlarni uzatish davom etadi.

I<sup>2</sup>C shina orqali ma'lumot almashish jarayoni START holatini yaratuvchi usta bilan boshlanadi - master SCL liniyasida SDA liniyasining signalini HIGH-dan LOW-ga HIGH-dan HIGH-ga o'tishni ta'minlaydi. Ushbu o'tish shinaga ulangan barcha qurilmalar almashinuv tartibining boshlanishining belgisi sifatida qabul qilinadi. Almashtirish jarayoni STOP holatini shakllantiruvchi usta bilan yakunlanadi - SCL liniyasi HIGH bo'lsa SDA liniyasi holatining LOW dan HIGH holatiga o'tish. START va STOP holatlari har doim etakchi tomonidan yaratiladi. Shina START holatini o'rnatgandan keyin band bo'lganiga ishoniladi. Shina STOP holati o'rnatilgandan keyin biroz vaqt o'tgach bo'shatilgan deb hisoblanadi. Agar kerakli sxemalar o'rnatilgan bo'lsa, shinaga ulangan qurilmalar orqali START va STOP signallarini aniqlash juda oson. Shu bilan birga, bunday sxemalarga ega bo'lmagan mikrokontrollerlar SDA liniyasining qiymatini sinxronizatsiya davrida kamida ikki marta o'qishlari kerak.

Ma'lumotni uzatish paytida tasdiqlash majburiydir, qul tomonidan uzatishni tugatgandan tashqari. Mos keladigan sinxronizatsiya pulsini usta yaratadi. Transmitter tasdiqlash soati davomida (HIGH) SDA liniyasini chiqaradi. Qabul qilgich SDA chizig'ini HIGH tasdiq soatida barqaror ravishda LOW holatida ushlab turishi kerak. Albatta, o'rnatish va saqlash vaqtlari (elektr va vaqt parametrlari) ham hisobga olinishi kerak.

Shunday qilib, 8 ma'lumot bitini transmitterdan qabul qiluvchiga uzatish qo'shimcha tsikl bilan yakunlanadi (SCL chizig'ining 9- tsikl davrni shakllanishi), unda qabul qiluvchi baytni muvaffaqiyatli qabul qilish belgisi sifatida SDA liniyasida LOW signal darajasini o'rnatadi. Agar qabul qiluvchi qul o'z manzilini tasdiqlay olmasa (masalan, hozirgi vaqtda real vaqtda biron bir funktsiyani bajarayotgan bo'lsa), ma'lumotlar liniyasi HIGH holatida qoldirilishi kerak. Shundan so'ng, usta ma'lumot uzatishni to'xtatish uchun STOP signalini berishi mumkin. Agar uzatishda usta-qabul qiluvchi ishtirok etgan bo'lsa, u oxirgi baytni tasdiqlamasdan, uzatuvchi qul-uzatuvchiga xabar berish kerak. Magistrga STOP signalini berishga yoki START signalini takrorlashga ruxsat berish uchun transmitter quli ma'lumotlar qatorini tozalashi kerak.

Paketlarni I<sup>2</sup>C shina orqali yuborishda, har bir usta SCL chizig'ida o'z soat signalini yaratadi. Ma'lumotlar faqat soatning yuqori holati paytida haqiqiydir.



Sinxronizatsiya SCL liniyasiga ulanish yordamida I oʻrnatish qoidasiga muvofiq amalga oshiriladi, bu SCL chizigʻining LOW dan HIGH darajasiga oʻtishini boshqarish mutlaq huquqiga ega emasligini anglatadi. Agar qabul qilingan bitni qayta ishlash uchun qulga qoʻshimcha vaqt kerak boʻlsa, u keyingi bitni olishga tayyor boʻlgunga qadar SCL chizigʻini past darajada ushlab turish imkoniyatiga ega. Shunday qilib, SCL chizigʻi soat signallarining eng uzun LOW davri uchun LOW holatida boʻladi.

Qisqa LOW davrga ega boʻlgan qurilmalar uzoq muddat tugaguncha kutish vaqtiga kiradi. Ishlayotgan barcha qurilmalar LOW soat davrini tugatgandan soʻng, SCL liniyasi HIGH ga oʻtadi. Barcha qurilmalar soat signallarining yuqori vaqtini oʻtkazishni boshlaydilar. Ushbu davrni tugatgan birinchi qurilma yana SCL chizigʻini LOW ga oʻrnatadi. Shunday qilib, SINHrolinii SCL LOW davri barcha taʼsirlangan qurilmalarning eng uzoq vaqtini sinxronizatsiya davrini aniqladi va HIGH davri sinxronizatsiya qurilmasining eng qisqa davri bilan belgilanadi.

Sinxronizatsiya mexanizmini qabul qiluvchilar bayt va bit darajalarida maʼlumotlarni uzatishni boshqarish vositasi sifatida ishlatishlari mumkin. Bayt darajasida, agar qurilma maʼlumotlar baytlarini yuqori tezlikda qabul qilishi mumkin boʻlsa, lekin qabul qilingan baytni saqlash yoki keyingisini tayyorlash uchun maʼlum vaqt talab qilsa, u baytni qabul qilib, tasdiqlagandan soʻng SCL chizigʻini LOW ushlab turishi mumkin va shu bilan transmitterni kutish holatiga qoʻyadi.. Bit darajasida, I<sup>2</sup>C apparat zanjirlari yoki cheklangan zanjirlari boʻlmagan mikrokontroler kabi qurilma, LOW davrini uzaytirib, soat chastotasini pasaytirishi mumkin. Shunday qilib, har qanday usta uzatish tezligi sekin qurilmaning tezligiga moslashadi.

START signalidan soʻng, qul manzili yuboriladi. Manzilning 7 bitidan keyin maʼlumotlar yoʻnalishi biti (R / W), "nol" uzatish (yozish) degan maʼnoni anglatadi, "bitta" esa olish (oʻqish) degan maʼnoni anglatadi. Maʼlumot uzatish har doim usta yaratgan STOP signali bilan yakunlanadi. Ammo, agar usta bundan keyin shinada qolishni istasa, u START signalini va keyingi qurilmaning manzilini berishi kerak. Ushbu yuborish formatida oʻqish va yozishni turli xil kombinatsiyalari mumkin.

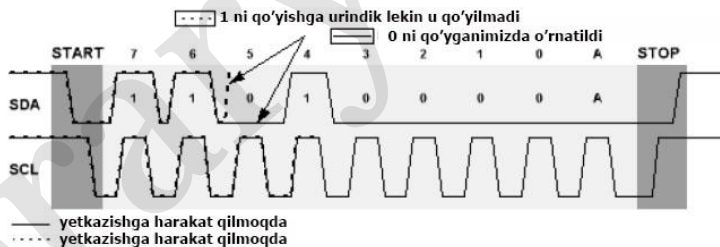
Arbitraj maʼlumotlari bir nechta etakchi ( multimaster rejimi ) mavjud boʻlganda IC orqali maʼlumot uzatishda nizolarni hal qilishga yordam beradi. Magistr maʼlumot uzatishni faqat shina ozod boʻlsa

boshlashi mumkin. Agar bitta usta ma'lumot chizig'iga LOW darajani yuborsa, ikkinchisi HIGH darajasida bo'lsa, oxirgisi satrdan uziladi, chunki SDA (LOW) holati uning ichki ma'lumotlar liniyasining HIGH holatiga to'g'ri kelmaydi.



4.2-rasm. START holatidan keyingi birinchi bayt (manzil bayt)

Hakmlik faqat raqobatlashayotgan rahbarlarning manzillari va ma'lumotlariga bog'liq bo'lganligi sababli, markaziy rahbar yo'q, shuningdek shinaga ustuvor kirish imkoniyati mavjud emas.



Rasm. 4.3. Ikki taqdimotchi o'rtasidagi hakamlik (bir vaqtning o'zida ma'lumotlar uzatish holati)

Ikkala taqdimotchi bir vaqtning o'zida uzatishni boshlaganda nima bo'ladi? Bu erda yana I- montaj xususiyati yordam beradi : ikkala usta ham asta-sekin qulning manzilini uzatishadi, so'ngra ushbu ziddiyatli vaziyatda "0" qatoriga birinchi kiritgan ma'lumotlar g'alaba qozonadi (4.3-rasm). Shunisi aniqki, eng muhim manzil noldan boshlanishi kerak, shunda unga murojaat qilganlar har doim hakamlikni yutib olishlari kerak. Yo'qotilgan tomon shinalar bo'shaguncha kutishga majbur.

Shunday qilib, hakamlik muhokamasi manzil oxirigacha davom etishi mumkin, va agar ma'ruzachilar bir xil qurilmaga murojaat qilsalar, ma'lumotlar hakamlik sudida ham qatnashadi. To'qnashuvda ushbu hakamlik sxemasi tufayli ma'lumotlar yo'qolmaydi. Arbitrajni yo'qotgan usta, kirish yo'qotilgan bayt oxirigacha SCL shinaga soat impulslarini berishga ruxsat etiladi. Agar qulning funktsiyalari shuningdek usta uskunasi birlashtirilgan bo'lsa va u manzilni uzatish bosqichida hakamlikni yo'qotsa, u darhol qul rejimiga o'tishi kerak, chunki arbitrajni yutgan rahbar unga murojaat qilishi mumkin.

I<sup>2</sup>C shinasining afzalliklari:

- Faqat ikkita liniy kerak - ma'lumotlar liniyasi (SDA) va sinxronizatsiya liniyasi (SCL) Shinaga ulangan har bir qurilma dasturiy ravishda noyob manzilga murojaat qilinishi mumkin. Vaqtning har bir nuqtasida oddiy usta va qul o'rtasidagi munosabatlar mavjud : magistrlar usta-uzatuvchi va usta-qabul qiluvchi sifatida ishlashlari mumkin.

- Shina bir vaqtning o'zida ikki yoki undan ortiq magistr bir vaqtning o'zida ma'lumotlarni uzatishni boshlagan vaziyatda ma'lumotlar buzilishining oldini olish uchun to'qnashuvlarni va hakamlik muhokamasini belgilash uchun vositalarni taqdim etadigan bir nechta magistr larga ega bo'lishga imkon beradi. Standart rejimda 8 bitli seriyali ma'lumotlar 100 kbit / s gacha va "tez" rejimda 400 kbit / s gacha tezlikda uzatiladi.

- O'rnatilgan mikrosirkulyator filtri portlashni bostiradi va ma'lumotlarning yaxlitligini ta'minlaydi.

- Bitta shinaga ulangan chiplarning maksimal soni shinning maksimal sig'imi bilan cheklangan.

Bular faqat ba'zi foydali narsalar. Bundan tashqari, I<sup>2</sup>C-ga mos keladigan mikrosxemalar tizimning moslashuvchanligini oshiradi, bu esa uskunalarni osonlikcha loyihalashtirish va yangilanishni osonlashtirishga imkon beradi. Shunday qilib, jihozlarning butun oilasini tayanch model asosida ishlab chiqish mumkin. Uskunani yangilash yoki uning funktsiyalarini kengaytirish (masalan, qo'shimcha xotira, masofadan boshqarish pulti va boshqalar) shunchaki tegishli mikrosxemani shinaga ulash orqali amalga oshirilishi mumkin. Agar sizga kattaroq ROM kerak bo'lsa, unda bu katta ROM hajmiga ega mikrokontrolderni tanlash masalasidir. Yangi mikrosxemalar eskilarini o'rnini bosishi mumkinligi sababli, eskirgan mikrosxemani uzib, yangisini shinaga ulab, uskunaga yangi xususiyatlar qo'shish yoki ularning ish faoliyatini oshirish oson.

## 4.2. RS -485 interfeysi.

**RS- 485** ( Recommended Standard 485, Electronics Industries Association 485, EIA-485) ma'lumotni ikki simli yarim dupleks ko'p nuqtali ketma-ket aloqa kanali orqali uzatish uchun standartdir.

RS-485 standarti ikkita uyushma tomonidan birgalikda ishlab chiqilgan: Elektron sanoati uyushmasi (EIA) va Aloqa vositalari sanoati uyushmasi (TIA). Avvalroq, EIA o'zining barcha standartlarini "RS" prefiksi bilan ( etakchi standart - tavsiya etilgan standart) etiketlagan edi. Ko'pgina muhandislar ushbu belgidan foydalanishda davom etishmoqda, ammo EIA/TIA o'zlarining standartlarining kelib chiqishini aniqlash maqsadida rasmiy ravishda "RS" ni "EIA/TIA" ga almashtirdilar. Bugungi kunda RS-485 standartining turli xil kengaytmalari turli xil dasturlarni qamrab oladi, ushbu standart sanoat avtomatlashtirishida keng qo'llaniladigan sanoat tarmoqlarining butun oilasini yaratish uchun asos bo'ldi. RS-485 standartida ma'lumot uzatish va qabul qilish uchun ko'pincha bitta burilgan juft sim ishlatiladi. Ma'lumot uzatish differentsial signallar yordamida amalga oshiriladi. Bir qutbning o'tkazgichlari orasidagi kuchlanish farqi mantiqiy birlik degan ma'noni anglatadi, boshqa qutblarning farqi nolga teng.

### **RS-485 quyidagi xususiyatlarga ega:**

- Assimetrik va nosimetrik zanjirlarni birlashtirish qobiliyati,
- signal sifati parametrlari, buzilish darajasi (%),
- aloqa liniyalariga kirish usullari,
- almashish protokoli,
- apparat konfiguratsiyasi (almashish vositasi, kabel),
- ulagichlar, rozetkalar, blokklar, kontaktlarning raqamlanishi,
- elektr ta'minoti sifati (barqarorlik, to'liqin, bardoshlik),
- uzun chiziqlardagi akslar.

### **RS-485 interfeysining elektr va vaqt xususiyatlari:**

• Ko'p nuqtali tarmoq konfiguratsiyasiga ega 32 uzatgich (bitta segmentda, bitta tarmoq segmentida maksimal liniya uzunligi: 1200 metr).

- Faqat bitta transmitter ishlaydi.
- Tarmoqdagi maksimal tugunlarning soni asosiy kuchaytirgichlarni hisobga olgan holda 250 ga teng.

### **Almashinuv xususiyatlari:**

- 62,5 kbit / s 1200 m (bitta buralgan juftlik)
- 375 kbit / s 300 m (bitta buralgan juftlik)

- 500 kbit / s
  - 1000 kbit / s
- 2400 kbit / s 100 m (ikkita o‘ralgan juft)
- 10 000 kbit / s 10 m

62,5 kbit / s, 375 kbit / s, 2400 kbit / s tezligi RS-485 standarti bilan belgilanadi. 500 kbit / s dan yuqori tezlikda ekranlangan o‘ralgan juftliklar tavsiya etiladi.

Transversiya turi - differentsial, potentsial. A va B chiziqlaridagi kirish va chiqish voltajining o‘zgarishi :  $U_a ( U_b ) -7V$  dan  $+ 12V (+ 7V)$ .

Buralgan juftlik va yuqori uzatish tezligi bilan ulangan qurilmalar orasidagi katta masofalarda uzoq chiziqlar deb nomlanadigan effektlar paydo bo‘la boshlaydi. Buning sababi o‘tkazgichlarda elektromagnit to‘lqinlarning tarqalishining cheklangan tezligidir. Bu tezlik vakuumdagi yorug‘lik tezligidan ancha past va 200 mm / ns dan bir oz ko‘proq.

Elektr signali, shuningdek, elektr uzatish liniyasining va uning shoxlarining ochiq uchlaridan ham aks ettirish xususiyatiga ega. Bir uchida hosil bo‘lgan elektromagnit to‘lqin chiziq bo‘ylab yuradi va oxirida devor orqali aks etadi, orqaga qaytadi va yana susayadi. Qisqa chiziqlar va past uzatish tezligi uchun bu jarayon shunchalik tez sodir bo‘ladiki, u beparvo bo‘ladi. Biroq, qabul qiluvchilarning javob vaqti o‘nlab / yuzlab ns. Bir necha o‘nlab metrlardagi bu vaqtda elektr uzatish bir zumda bo‘lmaydi. Va agar masofa etarlicha katta bo‘lsa, chiziqning oxirida aks ettirilgan va orqaga qaytarilgan old tomon joriy yoki keyingi signalni buzishi mumkin. Bunday hollarda, qandaydir tarzda aks ettirish effektini bostirish kerak. Har qanday aloqa liniyasida  $Z_v$  to‘lqin empedansi kabi parametr mavjud. Bu ishlatiladigan kabelning xususiyatlariga bog‘liq, ammo uning uzunligiga emas.  $Z_v = 120$  Ohm aloqa liniyalarida tez-tez ishlatiladigan burilgan juftliklar uchun. Aniqlanishicha, agar chiziqning eng chekka qismida, o‘ralgan juft o‘tkazgichlar o‘rtasida, chiziqning to‘lqin empedansiga teng qiymatga ega bo‘lgan rezistor yoqilsa, u holda "o‘lik oxiriga" etgan elektromagnit to‘lqin bunday rezistorga singib ketadi. Shuning uchun uning nomi - tugatish rezistori yoki "terminator".

Rezistorlardagi muvofiqlashtirishning katta minusi transmitterdan oqimning ko‘payishi hisoblanadi, chunki past empedans yuk liniyaga kiritilgan. Shuning uchun uzatgichni faqat posilkani yuborish vaqtida yoqish tavsiya etiladi. Hozirgi iste'molni kamaytirish usullari, shu

jumladan ketma-ket turadigan rezistorlar bilan bir qatorda, shaharni izolyatsiyalash uchun kondansatör mavjud. Biroq, bu usulning kamchiliklari bor. Qisqa chiziqlar (bir necha o'nlab metr) va past tezliklar (38,400 donadan kam) uchun solishtirishni umuman tashlab yuborish mumkin.

Ko'zgu effekti va muvofiq muvofiqlashtirish zarurati aloqa liniyalari konfiguratsiyasini cheklaydi. Aloqa liniyasi bitta o'ralgan juft juft kabel bo'lishi kerak. Barcha qabul qiluvchilar va uzatgichlar ushbu kabelga ulangan. Tarmoqdan RS-485 interfeysi chiqlarigacha bo'lgan masofa iloji boricha qisqa bo'lishi kerak, chunki uzun novdalar bir-biriga mos kelmaydi va akslarni keltirib chiqaradi. Kabelning ikkala eng tashqi uchlari ( $Z_v = 120 \text{ Om}$ ) har biri  $R_t 120 \text{ Om}$  ( $0,25 V_t$ )  $R_t$  rezistorlarini o'z ichiga oladi. Agar tizimda faqat bitta transmitter bo'lsa va u chiziqning oxirida bo'lsa, unda chiziqning qarama-qarshi uchida bitta tugatish rezistori kifoya qiladi.

### 4.3. CAN interfeysi

**CAN ( Controller Area Network )** bu yuqori darajadagi xavfsizlik bilan real vaqtda taqsimlangan boshqaruvni samarali qo'llab-quvvatlaydigan ketma-ket aloqa protokoli. BOSCH tomonidan ishlab chiqilgan. Uzatish rejimi - ketma-ket, keng translyatsiya, paket.

Standart fizik qatlamni ta'riflamaydi, lekin ko'p hollarda differentsial juftlikka asoslangan ISO 11898 standarti asosida shina topologiyasi qo'llaniladi va uzatish barcha tarmoq tugunlari tomonidan qabul qilingan ramkalar orqali amalga oshiriladi. To'qnashuvlarni hal qilish bilan ma'lumotlar tashuvchisiga kirish usuli xabarni uzatishda foydalanishga ustuvorlik beradi. Kadrdagi foydali ma'lumotlar 11 bit uzunlikdagi identifikatordan (standart format) yoki 29 bitdan (kengaytirilgan format) va uzunligi 0 dan 8 baytgacha bo'lgan ma'lumotlar maydonlaridan iborat. Identifikator paketning tarkibini ko'rsatadi va bir vaqtning o'zida bir nechta tugunlarni uzatishda ustuvorlikni aniqlash uchun xizmat qiladi.

Qo'llanish sohasi - yuqori tezlikda ishlaydigan tarmoqlardan arzon multipleks shinalarga qadar. Avtomatlashtirishda, boshqarish moslamalarida, CAN datchiklari 1 Mbit / s gacha tezlikda ishlatiladi.

Loyihaning shaffofligi va amalga oshirishning moslashuvchanligiga erishish uchun, ISO / OSI modeliga muvofiq, turli darajalarga bo'lindi :

- Ma'lumotlarni uzatish qatlami ( Data Link Layer ).
- Mantiqiy boshqarish liniyasini qatlam osti (LLC).
- Uzatish muhitiga ulanishini boshqarish qatlam osti (MAC).
  - Fizik qatlam ( Physical Layer)

Mantiqiy boshqarish liniyasini qatlam ostini qo'llanish sohasi:

• ma'lumotlarni uzatish va masofaviy so'rovlar bo'yicha xizmatlarni taqdim etish.

• LLC pastki qismidan qaysi xabarlarini aslida qabul qilish kerakligini hal qilish.

• Qayta tiklash va tiqilib qolishni boshqarish uchun vositalarni taqdim etish.

Uzatish muhitiga ulanishini boshqarish qatlam osti asosan uzatish protokoli, ya'ni arbitraj, xatolarni tekshirish, signalizatsiya va xatolarni terish. MAC qatlam ichida shina yangi uzatishni boshlashi mumkinmi yoki faqat ma'lumot olish mumkinmi degan qarorga kelindi. MAC qatlami, shuningdek, ba'zi vaqt elementlarini ham o'z ichiga oladi. Bularning barchasi MAC qatlami ichida joylashgan va uni o'zgartirish imkoniyati yo'q. Jismoniy qatlamning maydoni barcha elektr qoidalariga muvofiq turli tugunlar orasidagi bitlarning haqiqiy uzatilishi. Bitta tarmoq ichida, fizik qatlam barcha tugunlar uchun bir xil. Biroq, jismoniy darajani tanlashda erkinlik mavjud.

Asosiy xususiyatlari:

- xabarlarning ustuvorligi;
- kafolatlangan javob vaqti;
- konfiguratsiya moslashuvchanligi;
  - vaqtni sinxronlashtirish bilan guruhni qabul qilish;
- ma'lumotlarning muvofiqligi tizimi;
- multimaster ;
- xatolarni aniqlash va signalizatsiya qilish;
- shina yana bo'shagach, buzilgan xabarlarini avtomatik ravishda uzatish;
- doimiy bo'lmagan xatolar va tugunlarning doimiy ishdan chiqishi va nuqsonli tugunlarning avtonom yopilishi o'rtasidagi farq.

OSI modeliga muvofiq CAN-tarmoqni qatlamli arxitekturasi:

• Fizik qatlam signallarning haqiqatan ham qanday uzatilishini aniqlaydi va shuning uchun bit sinxronizatsiyasi va bit kodlash tavsifi bilan shug'ullanadi. Ushbu spetsifikatsiya doirasida, fizik qatlamni uzatuvchi / qabul qiluvchining xususiyatlari uzatuvchi vosita va signal

darajasining bajarilishini muayyan tizimlar uchun optimallashtirishga imkon beradigan aniqlanmagan.

- MAC qatlami CAN protokolining yadrosidir. LLC pastki qatlamidan olingan xabarlarni uzatadi va LLC pastki qatlamiga yuboriladigan xabarlarni oladi. MAC qatlami arbitraj, tan olish, xatolarni aniqlash va signalizatsiya uchun javobgardir.

- LLC pastki qatlami xabarlarni filtrlash, tiqilib qolish to'g'risida bildirish va tiklashni boshqarish bilan bog'liq.

#### **4.4. Sanoat Ethernet**

**Sanoat Ethernet (Industrial Ethernet)** – sanoatda foydalanish uchun Ethernet standartlashtirilgan (IEEE 802.3 va 802.11) versiyasi. CSMA / CD-ga kirish protsedurasiga ega tarmoq. Sanoat Ethernet odatda dasturlashtiriladigan kontrollerlar va inson-mashina interfeysi tizimlari o'rtasida ma'lumotlarni almashish uchun ishlatiladi, kontrollerlar o'rtasida kamroq ma'lumot almashish va ahamiyatsiz ravishda masofadan turib jihozlarni (sensorlar va aktuatorlarni) boshqaruvchilarga ulash uchun ishlatiladi. So'nggi vazifalarda Ethernetning keng qo'llanilishiga CSMA / CD usulining mohiyati to'sqinlik qilmoqda, bu oz miqdordagi ma'lumotni (bayt birliklari) yuqori chastotali (millisekundlik almashinuv tsikllari) almashinishini kafolatlamaydi.

Yaqinda bu eng keng tarqalgan sanoat tarmoqlaridan biri. Qurilishni avtomatlashtirishda va yuqori darajadagi ishonchlilikni talab qilmaydigan sohalarda keng qo'llaniladi.

Kafolatlangan reaksiya vaqtini ta'minlash uchun real vaqt protokollaridan foydalaniladi :

- Profinet
- EtherCAT
- Ethernet Powerlink
- Ether / IP

Ushbu protokollar TCP / IP standart stekini turli xil darajalarga o'zgartiradi, unga yangi funktsiyalarni qo'shib:

- sinxronizatsiya funktsiyalari;
- yangi tarmoq almashish algoritmlari;
- diagnostika funktsiyalari;
- o'z-o'zini sozlash usullari.



Kanal va fizik Ethernet qatlamlari o'zgarishsiz qoladi. Bu standart tarmoq uskunalari yordamida mavjud Ethernet tarmoqlarida real vaqt rejimida protokollardan foydalanishga imkon beradi.

Aloqa kanallarini bitta nosozlikdan himoya qilishni ta'minlash uchun ularni zaxiralash kerak. Shartnomaning qisqarishi muqarrar ravishda tarmoqning halqali bo'limlari - yopiq marshrutlar paydo bo'lishiga olib keladi. Ethernet standarti faqat daraxtlarning topologiyasini ta'minlaydi va halqalarga ruxsat bermaydi, chunki bu paketlarning aylanishiga olib keladi.

Zamonaviy kommutatorlar odatda qo'shimcha Spanning Tree Protocol (STP, IEEE 802.1d) protokolini qo'llab-quvvatlaydi, bu sizga Ethernet tarmoqlarida halqali yo'nalishlarini yaratishga imkon beradi. Tarmoq konfiguratsiyasini doimiy ravishda tahlil qilib, STP avtomatik ravishda zaxiraga ortiqcha aloqa liniyalarini uzatib, daraxtlar topologiyasini quradi. Shu tarzda qurilgan (yaxlitlik, masalan) tarmoq yaxlitligi buzilgan taqdirda, STP bir necha soniya ichida tarmoqning daraxt tuzilishini tiklab, zarur zaxira liniyalariga o'tadi. Ushbu protokol dastlabki konfiguratsiyani talab qilmaydi va avtomatik ravishda ishlaydi. Bu protokolni yanada kuchli versiyasi- Rapid Spanning Tree protokoli (RSTP, IEEE 802.1w) tarmoqni tiklash vaqtini bir necha millisekundlarda kamaytirish imkonini beradi.. STP va RSTP protokollari ko'p sonli aloqa liniyalarini yaratishga imkon beradi va rezerv tarmoqlarda ishlatiladigan sanoat kommutatorlari uchun majburiy xususiyatdir.

Oddiy Ethernet- dan farqlar :

- Sohaning aniq talablariga javob beradigan kabellar va ulagichlar uchun standartlar: yaxshilangan ekranlash, tajovuzkor muhitga qarshilik va boshqalar.

- Harakatlanadigan ob'ektlar bilan aloqa qilish uchun maxsus standartlar va moslamalar: moslashuvchan kabellar, simsiz qurilmalar

- TCP / IP protokollari to'plamini RFC 1006 bilan to'ldirish tarmoq kontrollerlari o'rtasida ma'lumot almashish uchun xos bo'lgan oz miqdordagi ma'lumotlarni tarmoq orqali muntazam va tez-tez uzatishni ta'minlaydi.

- Maxsus kommutatorlardan foydalanib, uzukli topologiyani tashkil qilish mumkin, bu tanaffusda aloqani tiklaydi, ya'ni an'anaviy tarmoqlarda ishlatiladigan "ortiqcha daraxt algoritmi" ga qaraganda tezroq ma'lumotlarni uzatishning yangi usulini topadi; TCP / IP protokollar stekti bilan birga maxsus protokollar to'plamidan tez-tez foydalanish. ISO transport protokoli

## 4.5. LIN interfeysi

**LIN ( Local Interconnect Network )** - bu Audi AG, BMW AG, Daimler Chrysler AG, Motorola Inc., Volcano Communications Technologies AB, Volkswagen AG va VolvoCar korporatsiyasi singari Evropaning avtomobil ishlab chiqaruvchilari va boshqa taniqli kompaniyalar konsorsiumi tomonidan ishlab chiqilgan tarmoq standarti. LIN protokoli qisqa masofalarga ma'lumot almashish uchun arzon narxlardagi mahalliy tarmoqlarni yaratishga mo'ljallangan. U kirish harakatlarini uzatishga, boshqaruv panellarini yoqishga va hokazolarga, shuningdek, "inson" deb ataladigan vaqt oralig'ida (yuzlab millisekundlarda) yuzaga keladigan LIN orqali bir tizimga ulangan turli xil qurilmalarning javob harakatlariga xizmat qiladi. Evropa avtomobil ishlab chiqaruvchilarining konsorsiumi tomonidan LINGa yuklatilgan asosiy vazifalar avtomobil ichki tizimlari va tarkibiy qismlarini (eshik qulflari, tozalagichlar, elektr oynalari, radio va iqlim nazorati, elektr O'rnatilgan tizim nuri va boshqalar) yagona elektron tizimga birlashtirish. LIN protokoli yuqori darajadagi ishonchli CAN protokoliga arzon qo'shimcha sifatida Evropa avtomobil konsorsiumi tomonidan tasdiqlangan.

LIN va CAN bir-birini to'ldiradi va barcha elektron avtoulov moslamalarini yagona ko'p funktsiyali bort tarmog'iga birlashtirishga imkon beradi. Bundan tashqari, CAN doirasi - o'ta ishonchlilik va tezlikni talab qiladigan joylar; LINning qo'llanish sohasi - bu qisqa masofalardagi past ma'lumot uzatish tezligi bilan ishlaydigan va ayni paytda ko'p qirrali, universal rejimida ishlaydigan arzon narxlardagi tugunlarning birlashmasi. LIN standarti protokollar va ma'lumotlar tashuvchisi uchun texnik talablarni o'z ichiga oladi. Ketma-ket aloqa protokoli sifatida, LIN A sinf shinasi (ikki tomonlama yarim dupleks) bilan avtoulov tizimlarida elektron qismlarni boshqarishni samarali qo'llab-quvvatlaydi, bu tizimda bitta usta va bir nechta qul tugunlarining mavjudligini anglatadi.

## 4.6. PLC texnologiyasi

**PLC ( Power Line Communication / Carrier )** - nisbatan yangi so'nggi mil telekommunikatsiya texnologiyasidir. "Tarmoqda joylashgan Internet" yuqori tezlikda ma'lumot almashish uchun xonadonda va xonadonda elektr tarmoqlaridan foydalanishga

asoslangan. Ushbu texnologiyada signalning chastotasini bo'linishiga asoslanib, yuqori tezlikda ishlaydigan ma'lumotlar oqimi bir necha past tezlikda ishlaydigan ma'lumotlar oqimlariga bo'linadi, ularning har biri alohida chastotada uzatilib, keyinchalik ularni bitta signalga birlashtiradi. Shu bilan birga, PLC qurilmalari ma'lumotni «ko'rishlari» va dekodlashlari mumkin, garchi an'anaviy elektr asboblari - lampalar, motorlar va h.k.lar ham tarmoq trafik signallari borligini «bilishmaydi» va normal rejimda ishlaydi. Power Line texnologiyasining asosi-signalning chastotaga ajratishidan foydalanish hisoblanadi, bunda yuqori tezlikda ishlaydigan ma'lumotlar oqimi bir necha nisbatan past tezlikda oqimlarga bo'linadi, ularning har biri alohida tashuvchi osti chastotasida uzatiladi va keyinchalik ularni bitta signalga birlashtiradi. Aslida Power Line texnologiyasi 4- 21 MGts diapazonida 84 tashuvchi osti chastotasidan foydalanadi.

PLC-larga ma'lumotlar uzatish tezligi 1 Mbit / s dan yuqori bo'lgan BPL (Power Lines over Broadband) va ma'lumotlar uzatishning ancha past tezligini ta'minlovchi NPL (Power Lines overrowband) kiradi.

Uy elektr tarmog'i orqali signallarni uzatishda ma'lum chastotalarda uzatish funksiyasida katta susayishi mumkin, bu esa ma'lumotlar yo'qolishiga olib kelishi mumkin. PowerLine texnologiyasi ushbu muammoni hal qilishning maxsus usulini taqdim etadi – dinamik ravishda signalni uzatishni yoqish va o'chirish (dynamically turning off and on data-carrying signals). Ushbu usulning mohiyati shundan iboratki, qurilma spektrning maydonini aniqlash uchun uzatish kanalining doimiy monitoringini amalga oshiradi, bu esa ma'lum bir pasayish chegarasidan oshib ketadi. Agar ushbu fakt aniqlansa, normal chastotani tiklash tiklanmaguncha ushbu chastotalardan foydalanish vaqtincha to'xtatiladi. Elektr manbalari bilan jihozlangan kuchli maishiy texnikani yoqish va o'chirish bilan bir qatorda impulsli shovqin (1 mikrosaniyagacha) paydo bo'lishi muammosi mavjud.

**PDSL** - bu xDSL oilasining texnologiyasi bo'lib, u uzatilayotgan elektr energiyasiga parallel ravishda quvvat simlari (4-20 kV ) orqali 2 Mbit/s gacha tezlikda ma'lumotlarni simmetrik ravishda uzatishni ta'minlaydi. PDSL uskunalari transformator shkaflariga o'rnatilgan interfeys moslamalari orqali yuqori voltli liniyalarga ulanadi.

#### **Afzalliklari:**

- Hech qanday kabel yo'q, uni qutilarga o'rab qo'yish, devorlar va qo'llab-quvvatlovchi tuzilmalarni burg'ulash;
- Foydalanish qulayligi;

- Oʻrnatish tezligi.

Wi-Fi dan PLC-ning afzalliklari :

- Sozlamalarni talab qilmaydi;
- Yanada barqaror aloqa;
- Axborot xavfsizligini ta'minlash;
- Ko'p tarmoqli trafik uchun mos, masalan, IPTV;
- Aloqa sifatiga kvartirada material va devor qalinligi ta'sir qilmaydi;

#### **Kamchiliklari:**

- PLC modemlari ishlaydigan xonalarda, ayniqsa oʻrta va qisqa toʻlqinlarda, modemdan 3-5 metr masofada juda qisqa masofada radio qabul qilishni buzish.

- Tarmoq oʻtkazish qobiliyati uning barcha qatnashchilari oʻrtasida taqsimlanadi. Masalan, agar bitta PowerLine tarmogʻida ikkita juft adapter faol ravishda ma'lumot almasha, har bir juftlik uchun almashinuv tezligi umumiy oʻtkazish qobiliyatining taxminan 50% ni tashkil qiladi.

- PLC barqarorligi va tezligiga simlarning sifati, turli materiallardan (masalan, mis va alyuminiy oʻtkazgich) boʻgʻinlar, shuningdek oʻtkazgich ulanishlarining soni ta'sir qiladi.

- Maxsus "PLC READY" rozetkalari bilan jihozlanmagan kuchlanishdan himoya qiluvchi va UPS orqali ishlamaydi.

- Aloqa sifatiga arzon energiya tejaydigan lampalar, tiristor dimmerlari, kommutatsiya quvvat manbalari va zaryadlovchilar ta'sir qilishi mumkin. Ruxsat etilgan qurilmalar PLC modemiga bevosita ulanganda tarmoq tezligiga maksimal darajada ta'sir qiladi.

#### **4.7. M2M texnologiyasi**

**M2M ( Machine-to Machine )** - mashina-mashina ora aloqa. Odatda, M2M tizimlari uyali GPRS modemlari asosida quriladi. Aloqa modem moduliga oʻrnatilgan TCP / IP steki orqali amalga oshiriladi.

Uyali aloqa tarmoqlari orqali almashish tezligi juda katta emasligi sababli, signalning sezilarli kechikishi va aloqa uzilishlari mavjud boʻlib, bunday tizimlarning asosiy ishlatilishi hisoblanadi:

- Katta hududlarda sekin jarayonlarni boshqarish (tashqi yoritishni boshqarish tizimlari, aqlli uy tizimi, uy-joy kommunal xoʻjaligini avtomatlashtirish tizimlari);

- Hududiy tarqalgan tizimlardan telemetriya ma'lumotlarini to'plash;
- Statsionar va harakatlanuvchi ob'ektlar uchun signalizatsiya tizimlari;
- Harakatlanadigan ob'ektlarning joylashuvi to'g'risida ma'lumot to'plash ( avtoulavlarning yurishini hisobga olish va boshqalar).

#### **4.8. ARINC 429 standarti**

**ARINC 429** - bu avionikada foydalanish uchun kompyuter shina uchun standart. ARINC firmasi tomonidan ishlab chiqilgan. Standart o'rnatilgan tizimning raqamli axborot tizimining asosiy funksiyalari va zarur fizik va elektr interfeyslarini tavsiflaydi. Bugungi kunda, ARINC 429 eng yaxshi jihozlangan samolyotlar uchun ustun bo'lgan aviatsiya shinalari.

ARINC 429 - bu ikki simli ma'lumotlar shinasini. Ulanish o'tkazgichlari - o'ralgan juftliklar. So'z hajmi 32 bitni tashkil etadi va aksariyat xabarlar bitta ma'lumot so'zidan iborat. Spetsifikatsiya elektr, aloqa va protokol xususiyatlarini belgilaydi.

ARINC 429 bir yo'nalishli ma'lumotlar shina standartidan foydalanadi (uzatish va qabul qilish liniyalari jismonan ajratilgan). Xabarlar uchta tezlikning bittasida uzatiladi: 12,5, 50 yoki 100 kbit / s. Transmitter har doim faol bo'lib, u 32 bitli ma'lumot so'zlarini uzatadi yoki "bo'sh" darajani beradi. Shinada 20 dan ortiq qabul qiluvchilar va bittadan ortiq transmitter ruxsat etilmaydi.

#### **4.9. MIL-STD-1553 standarti**

**MIL-STD-1553 (MIL-STD-1553B)** AQSh Mudofaa vazirligi standartidir, u elektron modul tizimida ishlatiladigan markazlashtirilgan ketma-ket magistral interfeysda qo'llaniladi. U dastlab AQSh Mudofaa vazirligining buyrug'i bilan harbiy avionikada foydalanish uchun ishlab chiqilgan, ammo keyinchalik uning qo'llanilish doirasi ancha kengaydi, standart fuqarolik tizimlarida ham qo'llanila boshladi. Interfeysning funksiyasi bu ikki baravar ortiqcha ma'lumot liniyasi, yarim dupleks protokoli va 31 tagacha masofadagi abonentlar (terminal qurilmalari). Har bir satr o'z kanal boshqaruvchisi tomonidan boshqariladi.

Ushbu standart quyidagi talablarni belgilaydi:

- Interfeys texnik vositalarining tarkibi;

- Ma'lumotlarning uzatilishini nazorat qilishni tashkil etish;
  - Axborot uzatish liniyasining tavsifi (LPI);
- Interfeys qurilmalarining xususiyatlari;
- Zaxira bilan interfeys.

Bitta shina 1 MGts chastotada 70-85 Ohm to'liq qarshiligi bo'lgan bir juft simdan iborat. Ulanish uchun dumaloq konnektor ishlatiladi, uning markaziy qismi bo'ylab Manchester kodi bilan kodlangan signal uzatiladi. Qabul qiluvchi va uzatuvchi terminal qurilmalari shinaga transformator izolyatsiyasidan foydalangan holda ulanadi va jalb qilinmagan ulanishlar transformator orqali izolyatsiya qilingan izolyatsiya qilingan rezistor juftligi yordamida ajratiladi. Bu qisqa tutashuvning ta'sirini kamaytiradi va samolyot korpusi orqali shina orqali oqim yo'qligiga ishonch qo'shadi. Manchester kodi ma'lumot uzatish va sinxronizatsiya signalini bir juft o'tkazgich orqali uzatish, shuningdek, transformator almashinuvi bilan kechiktiriladigan har qanday doimiy komponentni istisno qilish uchun ishlatiladi. Kanalning tarmoqli kengligi 1 Mbit / s dir. Xatolikka va uzoq muddatli uzatishga bardoshlik 0,1% ni tashkil qiladi, soat pulsining qisqa muddatli barqarorligi 0,01% chegarada bo'lishi kerak. O'tkazgichning kirish kuchlanishining amplitudasi 18-27 V bo'lishi kerak.

Axborot uzatish tizimidagi xabarlarining ko'pligiga shinadagi barcha qurilmalar ulangan ikkita yoki uchta mustaqil kanal (o'tkazgich) yordamida erishish mumkin. Ushbu ehtiyot choralari, joriy nosozlik yuzaga kelgan taqdirda takroriy shina boshqaruvchisini yoqish uchun qilingan.

Bundan tashqari, standartning MIL-STD-1773 deb nomlanuvchi ikkinchi versiyasi mavjud, unda optik tolalar kam vazn va yaxshiroq elektromagnit moslashuvchanlikka ega bo'lgan ma'lumot uzatish kanali sifatida ishlatiladi.

Odatda MIL-STD-1553B shina quyidagilardan iborat bo'lishi mumkin:

- Ikki kanal (asosiy va zaxira);
- Shina boshqaruvchisi;
- Terminal qurilmalar;
- Kanal monitoridan.

Istalgan vaqtda bitta shinada bitta boshqaruvchi bo'lishi mumkin. U ushbu shinadagi barcha xabarlarining tashabbuskori.

Nazoratchi:

- Ichki xotirada joylashgan buyruqlar bilan ishlaydi;

- Terminal qurilmalariga xabarlarini yuborish yoki qabul qilishni buyuradi;
  - Terminal qurilmalaridan kelib tushgan soʻrovlarga xizmat qiladi;
  - Xatolarni tuzatadi va tiklaydi;
  - Xatolar tarixini qoʻllab-quvvatlaydi.
- Terminal qurilmalari quyidagilar uchun ishlatiladi:
- Shina va unga ulangan quyi tizim oʻrtasida oʻzaro munosabatlarni tashkil etish;

- Ikki shinalar orasidagi koʻprikn tashkil qilish.

Kanal monitori terminal qurilmadan farq qiladi, chunki u shinada xabarlarini uzata olmaydi. Uning vazifasi shinadagi tranzaktsiyalarni nazorat qilish va qayd etishdir, bu nazorat qiluvchi va terminal qurilmalarining oʻzaro taʼsiriga xalaqit bermaydi. Ushbu yozuv keyingi tahlil uchun ishlatilishi mumkin.

### **Nazorat savollari:**

1. I<sup>2</sup>C mumkin boʻlgan dasturlarning roʻyxatini tuzing.
2. I<sup>2</sup>C ma'lumot almashish jarayoni qanday boshlanadi?
3. Ma'lumot almashish uchun I<sup>2</sup>C-da nechta qator ishlatiladi?
4. I<sup>2</sup>C shinasini asosiy ish rejimlarini tasvirlab bering.
5. Ikkala taqdimotchi bir vaqtning oʻzida I<sup>2</sup>C shina orqali uzatishni boshlaganda nima boʻladi?
6. I<sup>2</sup>C shinasining kuchli va zaif tomonlarini aytib bering.
7. RS-485 ning xususiyatlarini bering.
8. RS-485 interfeysining elektr va vaqtinchalik xususiyatlari.
9. ISO / OSI modeliga koʻra CAN nima darajalarga boʻlinadi?
10. Sanoat Ethernet nima uchun ishlatiladi ?
11. Industrial Ethernet va an'anaviy Ethernet oʻrtasidagi farqlar.
12. LIN protokoli qayerda ishlatiladi?
13. Power Line Communication / Carrier texnologiyasining mohiyati.
14. Power Line texnologiyasining asoslari nimadan iborat?
15. Elektr uzatish / tashuvchi yordamida elektr uzatish liniyasining afzalliklari va kamchiliklarini sanab bering.
16. M2M tizimlarini qoʻllash sohalari.
17. ARINC 429 xabarlar qanday tezlikda uzatiladi?
18. MIL-STD-1553 talablarini sanab bering.
19. MIL-STD-1553B tipik shinasining tarkibi?
20. MIL-STD-1553 standartida qaysi kod ishlatiladi?

## 5-BOB. O‘RNATILGAN TIZIMLARNI DASTURIY TAMINOTI

Ushbu bob ichiga o‘rnatilgan tizim modelidan foydalangan holda dasturiy ta‘minotni muhokama qiladi. O‘rnatilgan tizimda mavjud bo‘lishi mumkin bo‘lgan dasturiy ta‘minotning pastki qatlamlarini qayta joylashtirishni muhokama qiladi. Asosan, dasturiy ta‘minotni ikkita umumiy sinfga bo‘lish mumkin: tizim dasturiy ta‘minoti va amaliy dasturiy ta‘minot.

**Tizim dasturiy ta‘minoti** - bu qurilma drayverlari, operatsion tizimlar va vositachilik dasturlari kabi dasturlarni qo‘llab-quvvatlovchi har qanday dasturiy ta‘minot.

**Amaliy dasturiy ta‘minot** - bu o‘rnatilgan qurilmaning faoliyati va maqsadini belgilaydigan va foydalanuvchilar va ma‘murlar bilan o‘zaro aloqalarning ko‘p qismini boshqaradigan eng yuqori darajadagi dasturiy ta‘minot.

### 5.1. O‘rnatilgan tizimlarni tizimli va amaliy dasturiy ta‘minotni tashkil qilish

O‘rnatilgan tizimni tashkil etish va undan foydalanish xususiyatlari dasturiy ta‘minotni tashkil etishga ma‘lum cheklavlarni yuklaydi. Ushbu xususiyatlarga quyidagilar kiradi:

- kichik apparat resurslari;
- real vaqtda ko‘lami;
- ishonchliligi va xavfsizligi uchun ortib borayotgan talablarga muvofiqligi.

O‘rnatilgan tizimni loyihalash paytida dastur kodini yaratish usuliga ko‘ra dastur shaklida dasturiy ta‘minotni amalga oshirish variantlarini ajratib ko‘rsatish mumkin:

- monolit kod;
- bir nechta modullardan iborat tizim (drayverlar, amaliy dastur);
- real vaqt rejimidagi operatsion tizim (RTOS) va amaliy dastur kombinatsiyasi ;
- RTOS, virtual mashina, virtual mashinaning bir qismi sifatida ishlaydigan amaliy dastur (PLC, Embedded Java ).

Nomidan ko‘rinib turibdiki, monolitik dasturiy ta‘minot tizimi aniq qismlarga ega bo‘lmagan bir xil koddir. Bunday tizimning afzalliklari-



ichki aloqalarni tashkil etishning soddaligi va dasturchi harakatlarida (ayniqsa, rivojlanishning dastlabki bosqichlarida) erkinlik. Afsuski, kamchiliklar juda katta. Dasturning turli qismlari va individual komponentlarning (o'zgaruvchilar va funktsiyalarning) to'liq ko'rinishi o'rtasidagi aloqalardagi erkinlik tufayli xatolar ehtimoli juda yuqori. Murakkab monolitik tizimni qurish oson emas va murakkab vazifani qismlarga ajratish mumkin emas. Monolitik tizimining kattalashishi bilan xatolar tobora ko'payib boradi. Uning rivojlanishining ba'zi bosqichlarida xatolar soni va o'zgarishning murakkabligi shunday bo'ladiki, tizimni yanada rivojlantirish, undan foydalanish va sozlash qilish deyarli mumkin emas.

Monolit tizimlar dasturiy ta'minot sohasida juda kam uchraydi. Odatda, bunday tizimlar yangi boshlanuvchilar tomonidan amalga oshiriladi yoki tajribali mutaxassislar tomonidan tezda juda murakkab bo'lmagan va tizimning bir martalik prototipini tezkor qilish kerak bo'lganda.

Modulli tizimlar juda murakkab bo'lmagan o'rnatilgan tizimlarni qurish uchun keng tarqalgan variantdir. Manba darajasida yoki kutubxona darajasida modullik tizimni aniq belgilangan qismlarga ajratish, ularni jismoniy shaxslar uchun tuzatish, sinovdan o'tkazish va keyin tekshirishga imkon beradi.

Bunday tizimlardagi muammo murakkablikdir. Afsuski, hamma dasturchilar etarli darajada rivojlangan tizimli fikrlashga ega emas va ularning hammasi ham kodning qiyin bo'limlarini amalga oshira olmaydi. Ingliz tilida "framework" deb ataladigan arxitekturiy shablon deb nomlanadigan tizim skeleti kerak. Murakkablik muammosini hal qilish uchun ular odatda yuqori darajadagi ishlab chiquvchilar tomonidan amalga oshiriladigan ba'zi yadrolarni ajratadilar. Biz bunday skeletlari yoki arxitekturiy shablonlari uchun ikkita variantni ko'rib chiqamiz : real vaqtda operatsion tizim (RTOS) va virtual mashina.

RTOS -dan foydalanganda nima imkon beradi? Birinchidan, RTOS bu amaliy jarayonlar o'rtasida resurslarni taqsimlash vositasi, shuningdek, ushbu jarayonlarni tashkil qilish vositasi. Ikkinchidan, RTOS -bu yaxshi tashkil etilgan dastur kodi. Uchinchidan, RTOS, ma'lum bo'lgan afzalliklari va kamchiliklari bo'lgan arxitekturadir. To'rtinchidan, bu keng miqdordagi qo'shimcha qurilmalar (turli xil boshqarish moslamalari, portlashlar va boshqalar) bilan aloqani tashkil etish vositasidir. Ko'p almashinuv protokollari, turli xil protsessorlar va boshqaruvchilarning o'zini o'zi qo'llab-quvvatlash ko'pincha ABT

ishlab chiqishda ishtirok etadigan kompaniyalar uchun zararli emas va ishlab chiqaruvchidan tashqari ishlanmalardan foydalanish osonroq.

RTOS-dan foydalanishning kamchiliklari qanday? Tabiiyki, bozorda mavjud bo'lgan RTOS-ning aksariyati nisbatan universal tizim sifatida ishlab chiqilgan. Universallik bu funktsiyalarning ko'payishini va shuning uchun ushbu funktsiyalarni qo'llab-quvvatlash uchun qo'shimcha apparat manbalariga ehtiyojni anglatadi. Agar biron-bir loyihada tayyor RTOS ishlatilsa, yopiq tizimni olish mumkin, ya'ni, yashirin ichki tuzilishga ega tizimlar. Bunday qora qutini ishlatishga qarshi ko'plab dalillar mavjud. Ulardan biri bu tizimni tekshirishning iloji yo'qligi yoki jiddiyligi (masalan, sertifikatlash paytida), jiddiy xatolar va turli xil josuslarga qarshi dastur kodi hisobga olinmaganligi.

So'nggi paytlarda naqshlar asosida tizimlarni loyihalashtirish usuli ommalashmoqda. Shunday qilib, xususan, CoDesign loyihalarida RT operatsion tizimining oldindan o'rnatilgan sozlamalari (rejalashtiruvchilar, protsessorlar, IPC va boshqalar) ishlatiladi. Ushbu shablonlar arxitekturiy loyiha bosqichida qo'llaniladi va loyiha tizimining chiqishida ishlab chiqaruvchi monolit kodni oladi. Ushbu yondashuv universal (yoki sotib olingan) RT operatsion tizimidan foydalanishga xos bo'lgan ko'pgina kamchiliklardan mahrum.

Shunday qilib, RT OS-ni ishlatishga majbur qiluvchi sabablarga quyidagilar kiradi:

- tayyor, ishonchli va bashorat qilinadigan narsalardan foydalanish zarurati platformalar;
- qo'llaniladigan jarayonlarni parallel ishlashni ta'minlash zarurati
- jarayonlarning bir-biridan himoyasini ta'minlash zaruriyati;
- tayyor tarmoq drayverlariga, kirish/chikish qurilmalarga ehtiyoj (ko'pgina murakkab protokollar, murakkab qurilmalar mavjud, odam amalga oshirishi uchun ko'p yillar talab etiladi).

Murakkablik bilan kurashishning yana bir usulini ko'rib chiqing - virtual mashina. Virtual mashina sizga uskunaning barcha xususiyatlarini va real vaqt o'lchovini kuzatishda oxirgi dasturchidan ajratib olishga imkon beradi. Ushbu qulaylik yuqori narxga erishiladi.

**Birinchidan**, virtual ishlashni amalga oshirish zarurati tufayli tizimning ishlashi pasayadi. **Ikkinchidan**, dasturlash erkinligi torayib bormoqda.

Virtual mashinalar sanoatda, dasturlashtiriladigan mantiqiy boshqaruvchilar deb nomlanadigan narsalarda juda faol qo'llaniladi.

Dasturlash kontrollerlari uchun tillar allaqachon standartlashtirilgan. Ushbu standartlardan biriga misol IEC-61131-3.

## 5.2. O‘rnatilgan tizim apparat drayverlari

### 5.2.1. Aparat ta'minotiga bog‘liq dasturni amalga oshirish xususiyatlari

Hech kimga sir emaski, apparat ta'minotga bog‘liq dasturiy ta'minot yaratish odatiy dastur dasturlarini yozishdan ko‘ra ancha qiyin.

**Birinci sabab**, dasturchilar foydalanadigan hisoblash modelida apparat umuman ishlamaydi. Afsuski, o‘qitish tizimi shundayki, aksariyat dasturchilar hech qachon Von Neumannga asoslangan hisoblash modelidan chetga chiqmaydilar. Uskuna bilan ishlash parallelizm, uzilishlar tushunchasi, kombinatsion sxemalar va cheklangan avtomat nazariyasini emas, balki amalda tushunishni talab qiladi.

**Ikkinchi sabab** shundaki, vakolatli dasturiy ta'minotni yozish uchun loyihalashtirilayotgan tizimning apparati sxemasini tushunish kerak. Uskunani tushunish darajasi shunday bo‘lishi kerakki, dasturchi, agar kerak bo‘lsa, sxemani o‘zi tuzishi yoki operator tomonidan taklif qilingan sxemani tushunishi va tuzatishi mumkin. Yaxshiyamki, haqiqiy qurilmalarni uzoq vaqt ishlab chiqish va prototipini modellar bilan ishlashga almashtirish orqali qo‘shimcha qurilmalarni o‘rganish jarayoni ancha soddalashtirilishi mumkin. Endi Multisim yoki Proteus kabi kuchli zamonaviy zamonaviy dizayn va kosimulyatsiya vositalari mavjud.

**Uchinchi sabab**, apparatga bog‘liq bo‘lgan dastur hal qilinayotgan dastur muammosi bilan chambarchas bog‘liqdir. Dasturiy ta'minotni "umuman" haqiqiy muammodan ajratib dasturlash mumkin emas. To‘liq tizimli dasturiy ta'minotni yozish uchun sinov stendlari va atrof-muhit simulyatorlarini yaratish kerak. Bunday vositalarni yaratishda bizga MATLAB simulyatsiya to‘plami va LABView virtual laboratoriyasi yordam berishi mumkin.

**To‘rtinchi muammo** - bu nisbatan kichik hisoblash resurslari. Bu haqiqat bizni oddiy dasturchilarga tanish vositalarning ko‘pchiligidan voz kechishga majbur qiladi.

Yuqoridagilarga shuni qo‘shimcha qilish kerakki, tizim dasturlari boshqa dasturlar uchun platforma hisoblanadi. Bu haqiqat bizni

ishonchlilik va xavfsizlikning turli mexanizmlarini amalga oshirish to'g'risida jiddiyoq o'ylashga majbur qiladi.

Uskuna mavhumligi darajasi, HAL ( Hardware Abstraction Layer ) - bu turli xil usullarda ishlab chiqarilgan yoki turli ishlab chiqaruvchilar tomonidan chiqarilgan uskunalarning amaliy xususiyatlarini yashiradigan dasturiy ta'minot bilan ta'minlangan platforma. Bunday birlashtirishning narxi qo'shimcha darajani amalga oshirish uchun qo'shimcha xarajatlardir. Qoida tariqasida, agar siz osonlikcha ko'chma tizimni yaratish vazifangiz bo'lsa, HALni nisbatan kuchli mikrokontrolörlarda tanishtirish mantiqiy bo'ladi.

HAL tizimning barcha apparat qurilmalarining barcha interfeyslarini birlashtirishga imkon beradi. Oddiy tizimlarda bunday birlashtirish drayver darajasida amalga oshiriladi.

HAL quyidagi hollarda rivojlanishni mantiqiy qiladi:

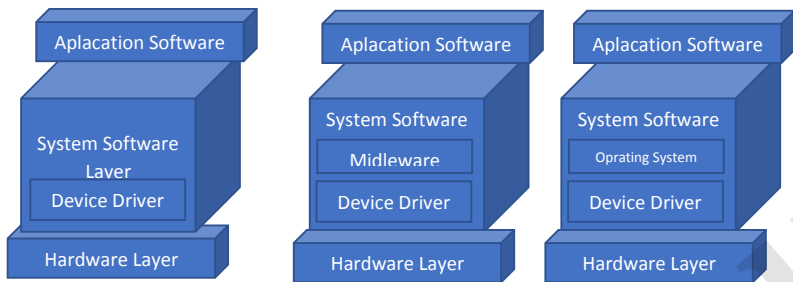
- Uskuna ko'pincha yangilanadi. Bunday yangilanishlar kichik partiyalar ishlab chiqarishda juda katta ehtimol.
- Bir qator apparat platformalari uchun dasturiy ta'minotni chiqarish kerak.
- Qurilma drayverlari murakkab va ularni joylashtirish juda katta resurslarni talab qiladi.

HALning nochorligi bu qo'shimcha darajaning kiritilishi tufayli tizimning ishlashidagi pasayishdir. Ba'zi hollarda, masalan, apparat platformalari bir-biridan tubdan farq qilganda, HAL-dan foydalanish amaliy bo'lmaydi.

## 5.2.2. Qurilma drayverlari

O'rnatilgan qurilmalarning aksariyati dasturiy ta'minotning ayrim turlarini ishga tushirishni va boshqarishni talab qiladi (5.1-rasm).

Ushbu uskunaga to'g'ridan-to'g'ri ta'sir o'tkazadigan va boshqaradigan dastur qurilmalar drayveri deb ataladi. Dasturiy ta'minotni talab qiladigan barcha ichki tizimlar kamida dasturiy ta'minot darajasida qurilma drayveri dasturiga ega. **Qurilma drayverlari** - bu yuqori darajadagi dasturiy ta'minot yordamida apparatni ishga tushiradigan va qo'shimcha qurilmalarga kirishni boshqaradigan dasturiy ta'minot kutubxonalari. Qurilma drayverlari - bu apparat va operatsion tizim, o'rta dastur va dastur qatlamlari o'rtasidagi munosabatlar.



Rasm 5.1. O‘rnatilgan model va qurilma drayverlari

Qurilma drayverini qo‘llab-quvvatlashni talab qiladigan qo‘shimcha qismlarning turlari har bir platada farq qiladi, ammo ularni fon Neymanning model uslubiga ko‘ra tasniflash mumkin. Fon Neumann modeli ma‘lum platformada qaysi qurilma drayverlari kerakligini aniqlash uchun ham dasturiy ta‘minot modeli, ham apparat modeli sifatida ishlatilishi mumkin. Bunga protsessorning asosiy arxitektura funktsiyalari, xotira va xotirani boshqarish, avtobusni ishga tushirish va tranzaksiya drayverlari, hamda ishga tushirish va kiritish-chiqarishni boshqarish drayverlari ham kengash darajasida, ham protsessor darajasida bo‘lishi mumkin (masalan, tarmoqlar, grafikalar, kirish qurilmalari, saqlash moslamalari). Ham plata darajasida, ham CPU darajasida.

Qurilma drayverlari odatda arxitektura va umumiy deb hisoblanadi. Arxitektura uchun mos keladigan qurilma drayveri asosiy protsessorga o‘rnatilgan apparatni boshqaradi. Asosiy protsessoridagi tarkibiy qismlarni ishga tushiradigan va ishga tushiradigan arxitektura drayverlarining namunalari o‘rnatilgan xotirasi, o‘rnatilgan xotira menejerlari (MMU) va suzuvchi nuqtali apparati kiradi. Umumiy bo‘lgan qurilma drayveri, platada bo‘lgan va asosiy protsessorga qo‘shilmagan apparatni boshqaradi. Umumiy drayverda odatdagi manba kodlari arxitekturaga murojaat qiladi, chunki asosiy protsessor markaziy boshqaruv bloki bo‘lib, kengashdagi har qanday narsaga kirish odatda asosiy protsessoridan o‘tishni anglatadi. Shu bilan birga, umumiy drayver, shuningdek, ushbu protsessorga xos bo‘lmagan plata apparatini boshqaradi, ya‘ni umumiy drayver haydovchi yozilgan tegishli apparat platalarini o‘z ichiga olgan turli xil arxitekturalarda ishlash uchun sozlanishi mumkin. Umumiy drayverlarga plataning shinalari (I2C, PCI, PCMCIA), tashqi xotira (kontrollerlar, IIC, kesh, Flash va boshqalar),

shu jumladan qolgan asosiy plataning boshqa qismlariga kirishni boshlaydigan va boshqaradigan kod va I/U chip kiradi (Ethernet, RS-232, display, sichqoncha va boshqalar).

Qurilma drayveri turidan yoki u boshqaradigan apparatdan qat'i nazar, barcha qurilmalar drayverlari odatda quyidagi funktsiyalarning bir qismidan yoki bir nechtasidan iborat:

- uskunani ishga tushirish, quvvat yoqilganda yoki qayta o'rnatishda apparatni ishga tushirish;

- uskunani o'chirish, jihozni o'chirish holatiga keltirish;

- uskuna o'chirilgan, bu boshqa dasturiy ta'minotni tezkor ravishda apparatni o'chirishga imkon beradi;

- boshqa dasturiy ta'minotning o'z vaqtida apparatni yoqishiga imkon beradigan apparat;

- boshqa dasturiy ta'minotga qo'shimcha qurilmalarga singular (to'sib qo'yuvchi) kirish huquqini beradigan apparat;

- boshqa dasturiy ta'minotga apparatni chiqarishga (qulfini ochishga) imkon beradigan apparat chiqarilishi;

- "O'qish" apparati, bu boshqa dasturiy ta'minotga apparatdan ma'lumotlarni o'qish imkonini beradi;

- boshqa dasturiy ta'minotga qo'shimcha ma'lumotlarga ma'lumotlarni yozish imkoniyatini beruvchi apparatni yozib olish;

- qurilmani o'rnatish, boshqa dasturlarga yangi apparatni tezda o'rnatishga imkon beradi;

- qurilmani o'rnatish, boshqa dasturlarga yangi apparatni tezda o'rnatishga imkon beradi;

- uskunani olib tashlash, boshqa dasturiy ta'minotga o'rnatilgan uskunani zudlik bilan olib tashlashga imkon beradi.

Albatta, qurilma drayverlari qo'shimcha funktsiyalarga ega bo'lishi mumkin, ammo yuqorida ko'rsatilgan ba'zi yoki barcha funktsiyalar qurilma drayverlariga tegishli. Ushbu funktsiyalar dasturiy ta'minotni aniq idrok etishga asoslangan bo'lib, bu har qanday vaqtda - faol emas, band yoki tugagan holda apparat uch holatda bo'lishidan iborat. Ishlamaydigan holatda uskuna ulanmagan (masalan, o'rnatish funktsiyasi zarurati), quvvatisiz (demak, boshlang'ich protsedurasi zarurati) yoki o'chirilgan deb talqin qilinadi (demak, protsedura zarurati). Band bo'lgan va tugallangan holatlar uskunaning faol holatidir, aksincha; shuning uchun funktsiyani o'chirishingiz, o'chirishingiz yoki o'chirishingiz kerak. Band bo'lgan apparat ma'lum turdagi ma'lumotlarni faol ravishda qayta ishlaydi va faol emas, shuning

uchun ozod qilish mexanizmining ba'zi turlari talab qilinishi mumkin. Uskunalar tayyor holatda, masalan, qabul qilish, o'qish yoki yozish uchun so'rovlarni qabul qilishga imkon beradigan kutish holatida.

Shunga qaramay, qurilma drayverlari ushbu funktsiyalarning barchasiga yoki bir qismiga ega bo'lishi mumkin va ba'zi funktsiyalarni bitta katta funktsiyalarga birlashtirishi mumkin. Ushbu drayver funktsiyalarining har birida odatda yuqori darajadagi dasturiy ta'minot bilan o'zaro aloqada bo'lgan apparat va kod bilan bevosita bog'liq bo'lgan kod mavjud. Ba'zi hollarda, bu darajalar orasidagi farq aniq va boshqa drayverlarda kod mahkam o'rnatilgan.

Asosiy protsessorga qarab, har xil turdagi dasturlar har xil rejimlarda ishlashi mumkin, eng keng tarqalgani monitoring va foydalanuvchi rejimlari. Ushbu rejimlar tizim tarkibiy qismlariga dasturiy ta'minotga kirishga ruxsat berilganligi jihatidan sezilarli darajada farq qiladi va dispetcherlik rejimida ishlaydigan dastur foydalanuvchi rejimida ishlaydigan dasturlarga qaraganda ko'proq imkoniyatlarga (imtiyozlarga) ega. Qurilma drayveri kodi odatda dispetcher rejimida ishlaydi. Keyingi bir nechta bo'limlarda qurilma drayverlari funktsiyalari qanday yozilishi va ular qanday ishlashi mumkinligini ko'rsatadigan qurilma drayverlari misollari keltirilgan. Ushbu misollarni o'rganib chiqib, o'quvchi har qanday taxtani ko'rib chiqishi va ushbu tizimga qaysi qurilma drayverlarini qo'shilishini aniqlab, von Neumann modelidan foydalanib, drayverlarga kerak bo'lishi mumkin bo'lgan uskunalarni turini kuzatib borish imkoniyatini beradi.

### **Uzilishlarni qayta ishlash qurilma drayverlari**

Yuqorida aytib o'tilganidek, uzilishlar - bu asosiy protsessor tomonidan ko'rsatmalar oqimini bajarish paytida voqea sodir bo'lgan signallar. Bu shuni anglatadiki, uzilishlar asenkron ravishda ishga tushirilishi mumkin, tashqi apparat qurilmalari, qayta tiklash, elektr uzilishi. Yoki tizim qo'ng'iroqlari yoki noqonuniy ko'rsatmalar kabi ko'rsatmalarga tegishli harakatlar uchun sinxron ravishda. Ushbu signallar asosiy protsessorga joriy ko'rsatmalar oqimining bajarilishini to'xtatishga va uzilish xizmatini ishga tushirishga olib keladi. Asosiy protsessorida uzilishlarni boshqaruvchi va apparatning uzilish mexanizmlarini boshqaradigan dastur (ya'ni uzilishlarni boshqarish) uzilishlarni boshqarish uchun qurilma drayverlaridan iborat. Qurilma

drayverlari ro'yxatidagi kamida o'nta funksiyadan to'rttasini to'xtatuvchi qurilmalar qo'llab-quvvatlaydi, shu jumladan:

- To'xtatishni boshlash, qurilmani yoqish yoki asl holatiga qaytarish bilan boshlash.

- Ishlov berish paytida uzilishni o'chirib qo'yish, uskunaning uzilib qolishini sozlash.

- Interrupt-Handling Disable, bu boshqa dasturlarga o'chirib bo'lmaydigan uzilishlar bo'lgan parvozidagi faol uzilishlarni o'chirib qo'yish imkonini beradi.

- Interrupt-Handling Enable boshqa dasturlar chivin harakatsiz uzilishni hal qilish imkonini beruvchi va ishlov berishga xalaqit beradigan yana bir qo'shimcha xususiyat.

- Interrupt-handler-ishlov berish - bu bajarilishning asosiy yo'nalishi uzilishidan keyin bajariladigan kodning o'zi.

Dasturiy ta'minotni ishga tushirish, o'chirish, yoqish va xizmat ko'rsatish funksiyalari qanday amalga oshiriladi, odatda, quyidagi mezonlarga bog'liq:

- Uzilishlarning turlari, soni va ustuvorlik darajasi mavjud (chipdagi apparat uzilish mexanizmlari bilan belgilanadi).

- Shovqinlar qanday ishlaydi.

- Tizimdagi uzilishlarga olib keladigan komponentlar va asosiy uzilish protsessor tomonidan taqdim etiladigan xizmatlar uchun uzilishlar siyosati.

Uchta asosiy uzilishlar - bu dasturiy ta'minot, ichki apparat va tashqi apparat uzilishlari.

**Dasturiy ta'minot uzilishlari** asosiy protsessor tomonidan bajarilgan ko'rsatmalar oqimidagi ba'zi ko'rsatmalar tomonidan aniq ichki ravishda qo'zg'atiladi. Boshqa tomondan, ichki apparatdagi uzilishlar noqonuniy matematik operatsiyalar (toshib ketish, nolga bo'linish), sozlash (bir bosqichli) kabi apparat xususiyatlari (yoki cheklovlar) tufayli mezon protsessor tomonidan bajarilayotgan joriy o'qitish oqimi bilan bog'liq muammolar natijasida vujudga kelgan voqea (uzilish nuqtalari), noto'g'ri yo'riqnomalar (ish kodlari) va boshqalar.

Ba'zi bir ichki hodisalar tomonidan asosiy protsessor tomonidan qo'zg'atilgan (so'ralgan) uzilishlar, asosan dasturiy ta'minot va ichki apparat uzilishlari, odatda istisnolar yoki tuzoqlar deb nomlanadi. Istisnolar - bu dasturiy ta'minotni bajarish paytida asosiy protsessor tomonidan aniqlangan xatolar natijasida yuzaga keladigan ichki apparat uzilishlari, masalan, yaroqsiz ma'lumotlar yoki nolga bo'lish.



Istisnolarning ustuvorliklari va ustuvorliklari me'morchilik tomonidan qanday aniqlanadi. Tuzoqlar - bu maxsus ko'rsatma orqali dastur tomonidan yaratilgan dasturiy ta'minotning uzilishlari.

Va nihoyat, **tashqi apparat uzilishlari** - bu asosiy protsessor platasi va I / U platalaridan tashqari qo'shimcha qurilmalar tomonidan uzilishlar. Tashqi hodisalar tufayli yuzaga keladigan uzilishlar uchun asosiy protsessor IRQ so'rov pini yoki port (uzilish so'rovi darajasi) deb nomlangan kirish pini orqali tashqi oraliq apparatga (ya'ni uzilish tekshirgichlariga) yoki to'g'ridan-to'g'ri taxtadagi boshqa komponentga uzilishni ko'tarishni xohlaganlarida asosiy protsessorga signal beradigan uzilish portlari. Ushbu turdagi uzilishlar ikki usuldan biri bilan boshlanadi: darajani olish yoki chekka ishlamay qolish. Darajadan kelib chiqadigan uzilish, uzilish so'rovi (IRQ) signalining ma'lum bir darajasida (ya'ni HIGH yoki LOW) bo'lganda amalga oshiriladi. Ushbu uzilishlar protsessor IRQ chizig'ini olish paytida, masalan, har bir buyruqni qayta ishlash jarayonida, darajadagi tetiklash so'rovini aniqlaganda qayta ishlanadi.

Ikkala turdagi uzilishlar ham o'zlarining kuchli va zaif tomonlariga ega. Darajadagi uzilishlarda, agar so'rov bajarilayotgan bo'lsa va keyingi namuna olish davrigacha o'chirilmagan bo'lsa, protsessor yana o'sha uzilishni bajarishga urinadi. Boshqa tomondan, agar protsessorni ishga tushirish davridan oldin qatlam tomonidan boshlangan uzilish ishga tushirilsa va o'chirib qo'yilgan bo'lsa, CPU uning mavjudligini hech qachon sezmaydi va shuning uchun uni qayta ishlamaydi. Agar chekka uzilishlar bir xil IRQ liniyasidan foydalansalar, agar ular xuddi shu vaqtda bir xil vaqtda boshlangan bo'lsa, natijada protsessor faqat bitta uzilishni aniqlay oladi.

### **Xotira qurilmasi drayverlari**

Aslida jismoniy xotiraning barcha turlari noyob qator va ustun bilan yo'naltirilgan kataklardan tashkil topgan ikki o'lchovli massiv (matritsalar) bo'lsa ham, asosiy protsessor va dasturchilar xotirani odatda bitta xotira kartasi deb ataladigan katta o'lchamli qator deb bilishadi. Xotira kartasida qatorning har bir uyasi baytlardan iborat (8 bit) va har bir satrda bayt soni ma'lumotlar shina kengligiga bog'liq (8 bit, 16 bit, 32 bit, 64 bit). Bu, o'z navbatida, bosh arxitektura registrlarining kengligiga bog'liq. Fizik xotira dasturiy ta'minot nuqtai nazaridan qaralganda, u odatda mantiqiy xotira deb nomlanadi va uning

asosiy bazasi baytdir. Mantiqiy xotira butun oʻrnatilgan tizimdagi barcha jismoniy xotiralardan (registrlar, ROM va RAM) iborat.

Dastur tizimdagi protsessorlarni xotira kartasining turli qismlariga kirish imkoniyatini taʼminlashi kerak. Asosiy protsessorida va kartada xotirani boshqaradigan va xotira apparat mexanizmlarini boshqaradigan dastur umumiy xotira quyi tizimini boshqarish uchun qurilmalar drayverlaridan iborat. Xotira quyi tizimi xotirani boshqarish komponentlari va MMU kabi xotira boshqaruv qismlarining barcha turlarini, shuningdek, registrlar, kesh, ROM, DRAM kabi xotira kartasidagi xotira turlarini oʻz ichiga oladi. Odatda, yoki qurilma drayveri funksiyalari roʻyxatidagi oʻnta drayver funksiyalaridan oltitasining har bir kombinatsiyasi amalga oshiriladi, shu jumladan:

- Xotira quyi tizimini ishga tushirish, apparatni ishga tushirish, yoqish yoki qayta tiklash.

- Xotira quyi tizimini oʻchirish, apparatni oʻchirish holatiga oʻrnatish.

- Memory Subsystem Disable, boshqa dasturiy taʼminotni tezkor ravishda apparatni oʻchirishga imkon beradi (keshni oʻchirib qoʻyish).

- Memory Subsystem Enable, boshqa dasturiy taʼminotni tezkor ravishda apparatni yoqishga imkon beradigan xotira quyi tizimini yoqing (keshni yoqing).

- Yozish xotira quyi tizimi, xotirada bayt yoki baytlar toʻplamini saqlash (keshda, ROM va asosiy xotirada).

- Oʻqish xotira quyi tizimi, xotiradan bayt yoki baytlar toʻplamidagi ("kesh", "ROM" va asosiy xotirada) "nusxa" ni olish.

Qaysi turdagi maʼlumotlar oʻqilgan yoki yozilmasin, xotiradagi barcha maʼlumotlar baytlar ketma-ketligi sifatida boshqariladi. Bitta xotiraga kirish maʼlumot shina hajmi bilan cheklangan boʻlsa ham, baʼzi arxitekturalar segmentlar deb ataladigan maʼlumotlarning katta bloklariga (qoʻshni baytlar toʻplamiga) kirishni nazorat qiladi va shu bilan dasturiy taʼminot orqali taqdim etilgan mantiqiy manzil bir nechta raqamlardan tashkil topgan manzillarni tarjima qilish sxemasini amalga oshiradi, xotira katakchasining fizik manzilini aniqlash uchun segment rakami (segment boshining manzili) va siljishdan (segment ichidagi) foydalanadilar. Baytlarni olish yoki xotirada saqlash tartibi arxitekturaning baytlarni buyurtma qilish sxemasiga bogʻliq.

Va nihoyat, dasturiy taʼminot xotirasiga haqiqatan kirish, oxirida dasturni yozish uchun ishlatiladigan dasturlash tiliga bogʻliq boʻladi. Masalan, assembler tilida arxitektura xos boʻlgan turli xil arxitektura

manzillari mavjud va Java sizga ob'ektlar orqali xotirani o'zgartirishga imkon beradi.

### **O'rnatilgan shina drayverlari**

Har bir shina bilan ba'zi protokollar turlari bog'langan bo'lib, ular qurilmalarning shinaga (arbitrajga) qanday kirishini, turli shina liniyalari bilan bog'liq signallarni aniqlaydi.

Shina protokoli quyidagilarni o'z ichiga oladi:

- Shinani ishga tushirish, avtobusni ishga tushirish yoki qayta yoqish.
- Shinani to'xtatish, avtobusni o'chirish holatiga o'tkazish.
- Bus Disable, boshqa dasturlarning shinani tezda o'chirib qo'yishga imkon beradi.
- Shinani yoqish, boshqa dasturiy ta'minot shinani tezda faollashtirishga imkon beradi.
- Bus Acquire, bu boshqa dasturiy ta'minotdan shinaga bir xil (blokirovka) kirish huquqini beradi.
- Shinani chiqarish, shinani chiqarish (ochish) uchun boshqa dastur.
- Bus Read- shinadagi ma'lumotlarni o'qish uchun shinani o'qing.
- Bus Write- boshqa dasturlarga shinaga ma'lumot yozish imkoniyatini beradi.
- Shinalarni o'rnatish.

Amaldagi dasturlarning qaysi biri amalga oshiriladi va qanday amalga oshiriladi, bu haqiqiy shinaga bog'liq.

### **I / O plata drayverlari**

Dasturiy ta'minotni boshqarishning bir turini talab qiladigan boshqaruv blokining tarkibiy qismlariga asosiy protsessor bilan birlashtirilgan komponentlar, shuningdek, agar mavjud bo'lsa, I/O boshqaruvchisi ham kiradi. Kirish / chiqish kontrollerlarida protsessorni boshqarish va uning holatini tekshirish uchun ishlatiladigan holat va nazorat registrlari to'plami mavjud. I/O quyi tizimiga qarab, odatda ushbu bobning boshida keltirilgan qurilma drayveri funktsiyalari ro'yxatidagi barcha 10 funktsiyalarning barchasi yoki ba'zi kombinatsiyalari odatda I / O drayverlarida amalga oshiriladi.

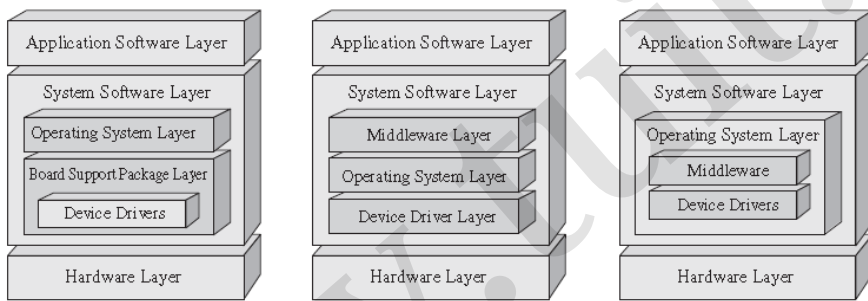
### **Nazorat savollari:**

1. Qurilma drayveri nima?
2. Muayyan arxitektura uchun qurilma drayveri va universal qurilma drayveri o'rtasidagi farq nima?
3. Qurilma drayveri funktsiyalarining besh turini sanab bering va tavsiflang.
4. Uzilish nima?
5. Qanday qilib uzilishlar boshlanishi mumkin?
6. Shovqinlarni boshqarish uchun bajarilishi mumkin bo'lgan qurilma drayveri funktsiyalarining to'rtta misolini nomlang va tasvirlab bering.
7. Uchta asosiy uzilishlar qanday?
8. Har bir tur ishga tushirilgan misollarni sanab bering.
9. Sath tomonidan qo'zg'atilgan uzilish va chet tomonidan qo'zg'aladigan uzilish o'rtasidagi farq nima?
10. Har birining kuchli va zaif tomonlari qanday?
11. ISR uzilishni boshlashdan oldin amalga oshiriladi.
12. Avtomatik va intervalgacha zanjirlar o'rtasidagi farq nima?
13. Xotirani boshqarish uchun bajarilishi mumkin bo'lgan qurilma drayveri funktsiyalarining to'rtta misolini nomlang va tasvirlab bering.
14. Bayt tartibi nima?
15. Mumkin bo'lgan baytlarni buyurtma qilish sxemalarini nomlang va tavsiflang.
16. Shina protokollari uchun bajarilishi mumkin bo'lgan qurilma drayveri funktsiyalarining to'rtta misolini nomlang va tasvirlab bering.
17. I/O uchun bajarilishi mumkin bo'lgan qurilma drayveri funktsiyalarining to'rtta misolini nomlang va tasvirlab bering.
18. OSI modelida Ethernet drayverlari va seriyali qurilmalar qayerda xaritaga olingan ?

## 6-BOB. O‘RNATILGAN OPERATSION TIZIMLAR

### 6.1. Operatsion tizimning tuzilishi

Operatsion tizim (OT) o‘rnatilgan qurilmaning tizim dasturiy ta‘minotining ixtiyoriy qismi bo‘lib, u barcha o‘rnatilgan tizimlarda mavjud emasligini anglatadi. Operatsion tizimlar operatsion tizim yuborilgan har qanday protsessorida (ISA) ishlatilishi mumkin. Rasmda ko‘rsatilgandek (6.1-rasm), OT qurilmaning yuqori qismida, qurilma drayveri darajasidan yuqori yoki BSP-ning ustida joylashgan.



6.1-rasm. OT va o‘rnatilgan tizim modeli

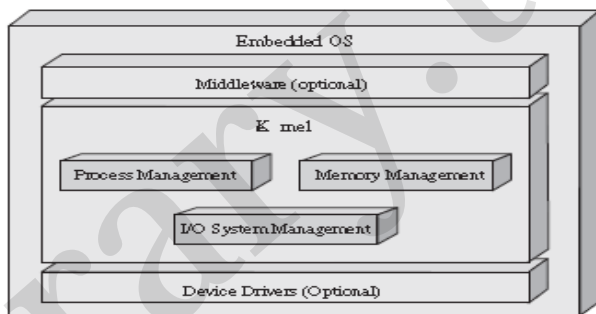
OT o‘rnatilgan tizimda ikkita asosiy maqsadga xizmat qiluvchi dasturiy kutubxonalar to‘plamidir: OT ning yuqori qismida dasturiy ta‘minotga mavhumlik darajasini ta‘minlash, shu bilan u apparat, OT tepasida joylashgan o‘rta dastur va dasturlarni ishlab chiqishga imkon beradi. Bu sodda va turli xil tizim apparat va dasturiy resurslarini boshqaradi, shunda butun tizim samarali va ishonchli ishlaydi. O‘rnatilgan OT lar qanday tarkibiy qismlarga ega bo‘lishlari bilan farq qilsa-da, barcha operatsion tizimlar hech bo‘lmaganda yadroga ega. Yadro - bu OT ning asosiy funktsional imkoniyatlarini, ya‘ni 6.2 (a-e) rasmda ko‘rsatilgan funktsiyalarning barcha yoki ba‘zi birlashmalarini va ularning o‘zaro bog‘liqligini o‘z ichiga oladigan tarkibiy qism, shu jumladan:

• **Jarayonlarni boshqarish.** OT o‘rnatilgan dasturdagi boshqa dasturlarni qanday boshqaradi va ko‘radi. Jarayonlarni boshqarishda keng tarqalgan funktsiya bu uzilishlar va xatolarni boshqarish hisoblanadi. Turli jarayonlar natijasida hosil bo‘lgan bir nechta uzilishlar

va/yoki tanaffuslar samarali boshqarilishi kerak, shunday qilib ular to'g'ri ishlov berilishi va ularni boshlagan jarayonlar to'g'ri kuzatilishi kerak.

• **Xotirani boshqarish.** O'rnatilgan tizimning xotira maydoni har xil jarayonlar tomonidan baham ko'riladi, shuning uchun siz xotira makonining qismlariga kirish va taqsimlashni nazorat qilishingiz kerak. Xotirani boshqarishning bir qismi sifatida, xavfsizlikni boshqarish kabi boshqa funktsiyalar, tizimning yopilishiga olib keladigan buzilishlarga moyil bo'lgan tizimning qismlarini dushman yoki yuqori darajadagi yomon yozilgan dasturlardan himoyalashga imkon beradi.

• **Kirish/chiqish tizimni boshqarish.** Kirish/chiqish qurilmalari, shuningdek, turli xil jarayonlar o'rtasida taqsimlanishi kerak va shuning uchun, xotira bilan bo'lgani kabi, kirish / chiqish qurilmasining kirish va tarqalishini boshqarish kerak. Kirish / chiqish tizimini boshqarish orqali fayl tizimini boshqarish fayllar shaklida ma'lumotlarni saqlash va boshqarish usuli sifatida ham ta'minlanishi mumkin.



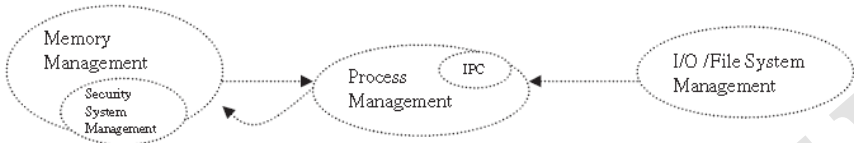
Rasm. 6.2 a. OSI umumiy modeli



Rasm. 6.2 b. Yadro quyi tizimiga bog'liqligi

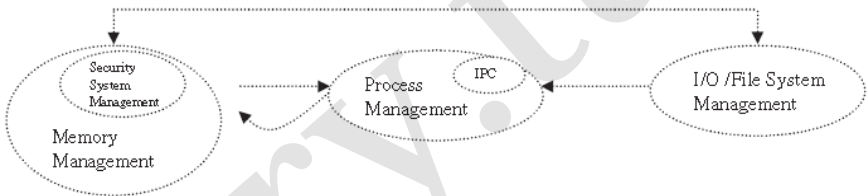
Operatsion tizim dasturlarni dasturiy ta'minotni jarayonlardan foydalangan holda boshqarishi sababli, jarayonni boshqarish

komponenti OTdagi eng markaziy quyi tizim hisoblanadi. Boshqa barcha OT quyi tizimlari jarayonni boshqarish blokiga bog‘liq.



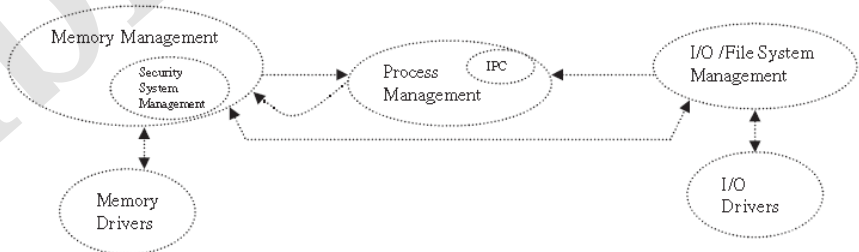
Rasm. 6.2 c. Yadro quyi tizimida bog‘liqliklar

Asosiy protsessorni ishga tushirish uchun barcha kod asosiy xotiraga (RAM yoki kesh) yuklanishi kerak va yuklash kodi va ma'lumotlar uchuvchi bo‘lmagan xotirada joylashgan (ROM, Flash va boshqalar), jarayonni boshqarish quyi tizimi quyi tizimga ham bog‘liq. xotirani boshqarish.



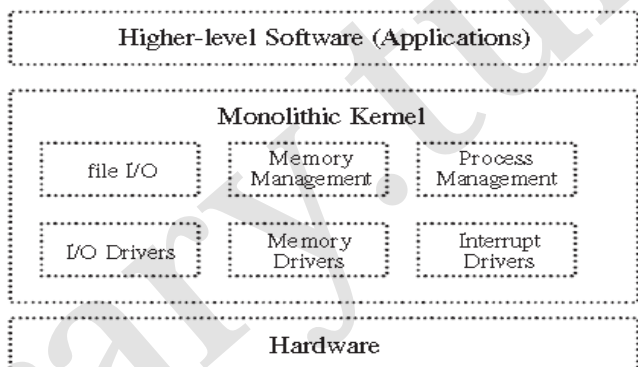
Rasm. 6.2.d. Yadro quyi tizimida bog‘liqliklar

Masalan, tarmoq / fayl tizimi (NFS) holatida xotira menejeri bilan o‘zaro ishlash uchun kirish/chiqish boshqarish tarmog‘ini o‘z ichiga olishi mumkin.



Rasm. 6.2 e. Yadro quyi tizimiga bog‘liqliklar

Yadro tashqarisida, xotira boshqaruvi va kirish/chiqishquyi tizimlari, keyinchalik qurilmaga kirish uchun qurilma drayverlaridan va aksincha foydalaning. OT yadrosi ichida yoki tashqarisida bo‘lishidan qat’i nazar, OT ham tizim dasturiy ta’minotining boshqa tarkibiy qismlari, masalan, qurilma drayverlari va o‘rta dasturlarni o‘z ichiga olganligiga qarab o‘zgaradi. Aslida, o‘rnatilgan OT larning aksariyati odatda uchta modeldan biriga asoslangan: monolit, ko‘p darajali yoki mikrokernel (mijoz-server). Umuman olganda, ushbu modellar OT yadrosining ichki loyihasida, shuningdek, boshqa tizim dasturlarining OT ga kiritilganligi bilan farq qiladi. Monolitik OT-da, o‘rta dastur va qurilma drayveri funksiyalari odatda yadro bilan birga OT ga birlashtirilgan. Ushbu turdagi OT bu barcha tarkibiy qismlarni o‘z ichiga olgan bitta bajariladigan fayldir (6.3-rasm).



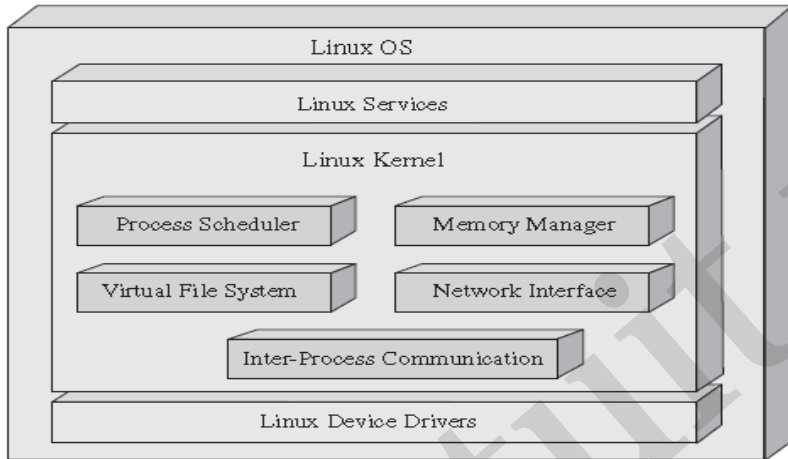
Rasm. 6.3. Monolit OT sxemasi

Monolit OT, odatda, boshqa OT arxitektura hamkasblariga qaraganda katta, yaxlit, o‘zaro bog‘liq bo‘lganligi sababli kamaytirish, o‘zgartirish yoki disk raskadrovka qilish qiyinroq. Shunday qilib, monolitik loyihaga asoslangan eng mashhur algoritm, monolitik-modulli algoritm deb nomlangan, oddiy monolitik yondashuv bilan solishtirganda sozlash, masshtablash va soddalashtirishni soddalashtirish uchun OTda amalga oshirildi.

Monolit-modulli OT da funktsionallik bitta bajariladigan faylga birlashtirilgan, bu modullardan, OT ning turli funksiyalarini aks ettiruvchi kodning alohida qismlaridan iborat. O‘rnatilgan Linux operatsion tizimi monolitik OTga misol bo‘lib, asosiy modullari



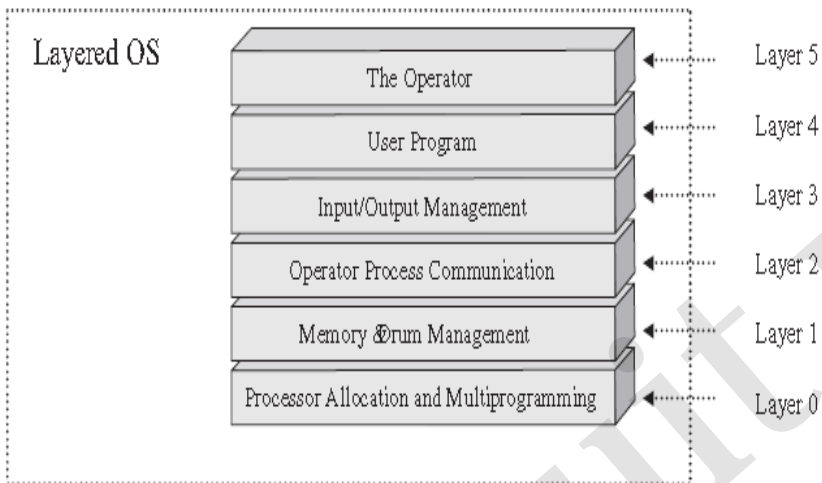
6.4.rasmda berilgan. JOC RTOS, MicroC / OC-II va PDOS o'rnatilgan monolitik OT larning namunalari.



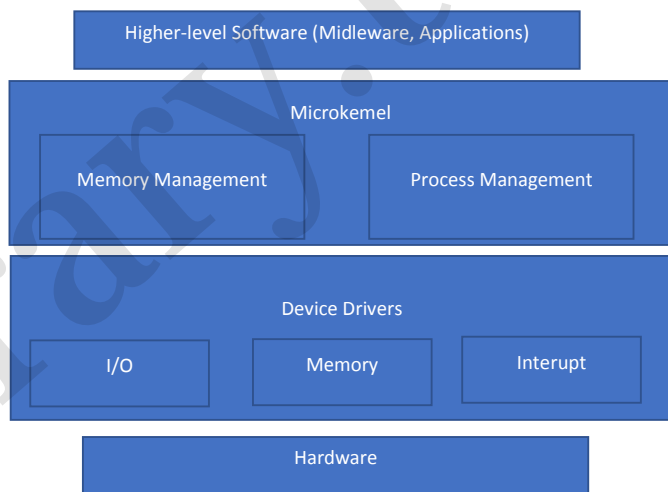
Rasm. 6.4. Linux OT jadvallari

Ko'p darajali strukturada OT ierarxik darajalarga bo'linadi (0... N), bu erda yuqori darajalar quyi darajalar tomonidan ta'minlangan funktsiyalarga bog'liq. Monolitik loyiha singari, ko'p darajali OTlar qurilma drayverlari va o'rta dasturlarni o'z ichiga olgan bitta katta fayldir (6.5-rasm). Darajali operatsion tizim monolitik tizimga qaraganda sodda va osonroq bo'lsa-da, har bir darajadagi taqdim etilgan API-lar hajmi va ishlashiga ta'sir qiladigan qo'shimcha xarajatlarni keltirib chiqaradi. DOS-C (FreeDOS), DOS / eRTOS va VRTX ko'p darajali OTga misoldir.

Rasmda ko'rsatilgandek, minimal ishlashga ega operatsion tizim, odatda faqat operatsion va xotira boshqaruv bloklari mavjud. 6.6 mijoz-server OT yoki mikrokernel deb nomlanadi. Boshqa yadro algoritmlariga xos bo'lgan qolgan funktsiyalar yadrodan mavhumlashtiriladi, shu bilan birga, qurilma drayverlari, masalan, 6.6-rasmda ko'rsatilgandek, mikrokerneldan butunlay chiqarib tashlanadi. Mikroyadro, odatda, OTning boshqa turlariga nisbatan jarayonlarni boshqarishni amalga oshirish bilan farq qiladi.



Rasm. 6.5. Ko‘p darajali OTning strukturaviy sxemasi



Rasm.6.6. Mikroyadroli OT sxemasi

Mikroyadroli OT, qoida tariqasida, yanada kengaytiriladigan (modulli) va tuzatilgan loyihadir, chunki unga qo‘shimcha komponentlar dinamik ravishda qo‘shilishi mumkin. Bundan tashqari, bu yanada

xavfsizroq, chunki funktsionallikning aksariyati hozirda OTdan mustaqil va mijoz uchun alohida xotira maydoni va server funktsiyalari mavjud. Shuningdek, yangi arxitekturalarni topshirish osonroq. Shu bilan birga, ushbu model boshqa OT arxitekturalariga qaraganda, masalan monolit, sekinroq bo'lishi mumkin, chunki mikroyadro komponentlari va boshqa "yadroga o'xshash" komponentlar o'rtasidagi ulanish paradigmasi. Yadro va boshqa OT komponentlari va OTdan tashqari tarkibiy qismlar (nisbatan ko'p darajali va monolit OT loyihalari) o'rtasida almashinishda qo'shimcha xarajatlar ham qo'shiladi. Alohida chiqarilgan va kamida yuzta o'rnatilgan operatsion tizimlarning ko'pida mikrokernellar toifasiga kiradigan yadrolar mavjud, ular orasida: OS-9, C Executive, vxWorks, CMX-RTX, Nucleus Plus va QNX.

## **6.2. Operatsion tizimlardagi jarayonlar**

OT o'rnatilgan qurilmaning apparat va dasturiy ta'minotini qanday boshqarishini tushunish uchun o'quvchi avval OT tizimga qanday qarashini tushunishi kerak. OT dastur va dasturning bajarilishini farqlaydi. Dastur shunchaki passiv statik ko'rsatmalar ketma-ketligidir, ular tizimning apparat va dasturiy resurslarini namoyish eta oladilar. Dasturning amalda bajarilishi - bu faol dinamik hodisa bo'lib, unda turli xil xususiyatlar vaqt va bajarilish bo'yicha o'zgaradi. Jarayon (odatda ko'pgina o'rnatilgan OTlarda vazifa deb ataladi) OT tomonidan dasturni bajarishda ishtirok etuvchi barcha ma'lumotlarni (ya'ni stack, kompyuter, manba kodi va ma'lumotlar va hokazo) inkapsulyatsiya uchun yaratilgan.

O'rnatilgan operatsion tizimlar barcha o'rnatilgan dasturlarni vazifalar yordamida boshqaradi va ular multitask yoki multitask bo'lishi mumkin. Istalgan vaqtda bitta vazifani bajaradigan OT muhitida faqat bitta vazifa bo'lishi mumkin, ko'p vazifali OT da esa bir vaqtning o'zida bir nechta vazifalar mavjud bo'lishi mumkin. Bir vazifali operatsion tizimlar odatda ko'p vazifali operatsion tizim kabi murakkab vazifalarni boshqarish tizimini talab qilmaydi. Ko'p tarmoqli muhitda bir nechta mavjud vazifalarni hal qilishning qo'shimcha murakkabligi har bir jarayon boshqalardan mustaqil bo'lib qolishini va maxsus dasturlashsiz boshqalarga ta'sir qilmasligini talab qiladi. Ushbu multitasking modeli har bir jarayonni ko'p darajali muhitda talab qilinmaydigan katta xavfsizlik bilan ta'minlaydi. Multitasking aslida murakkab o'rnatilgan tizimning ishlashini yanada uyushgan tarzda ta'minlashi mumkin. Ko'p

ishlaydigan muhitda tizim harakatlari sodda individual komponentlarga bo'linadi yoki bir xil jarayonlar bir vaqtning o'zida bir nechta jarayonlarda bajarilishi mumkin.

Ba'zi bir multitasking operatsion tizimlari, shuningdek, dastur inventarizatsiyasining qo'shimcha alternativ vositasi sifatida iplarni (engil jarayonlar) ta'minlaydi. Mavzular vazifa kontekstida yaratiladi (ya'ni, mavzu vazifa bilan bog'langan) va OT ga qarab, vazifa bir yoki bir nechta ipga ega bo'lishi mumkin. Ip bu o'z vazifasining bir qismi sifatida ketma-ket bajariladigan ip. O'ziga xos mustaqil xotira zonalariga ega bo'lgan boshqa vazifalarga kirish huquqiga ega bo'lmagan vazifalar uchun farqli o'laroq, vazifalar oqimi bir xil manbalarni taqsimlaydi (ishchi kataloglar, fayllar, kirish / chiqish qurilmalari, global ma'lumotlar, manzil maydoni, dastur kodi va boshqalar) va ko'rsatmalarning mustaqil bajarilishini ta'minlaydi. Iplar bir xil vazifa kontekstida yaratilgan va bir xil xotira maydonini ulashishi mumkinligi sababli, ular vazifalar bo'yicha osonroq o'zaro ta'sir va muvofiqlashtirishni ta'minlaydilar. Buning sababi, vazifa kamida bitta manzilni bitta dasturda bajaradigan bitta dasturni bajarishi mumkin yoki bitta manzilli maydonda bitta dasturning turli qismlarini vazifalar orasidagi o'zaro ta'sir mexanizmlarini talab qilmasdan bajaradigan ko'plab mavzularni o'z ichiga olishi mumkin. Bundan tashqari, umumiy manbalar mavjud bo'lsa, bir nechta mavzular bir xil ishni bajarish uchun bir nechta vazifalarni yaratishga qaraganda ancha arzon.

Odatda, dasturchilar tizimdagi har bir alohida harakat uchun bir-birining ustiga chiqadigan voqealar majmuasini emas, balki ushbu harakatning barcha harakatlarini bitta voqea oqimiga soddalashtirish uchun alohida vazifani (yoki ipni) belgilaydilar. Ammo, odatda, tizim dasturini namoyish qilish uchun qancha topshiriq ishlatilishini va agar mavzular mavjud bo'lsa, ular qanday qilib va qanday qilib vazifalar kontekstida ishlatilishini hal qilish dasturchiga topshiriladi.

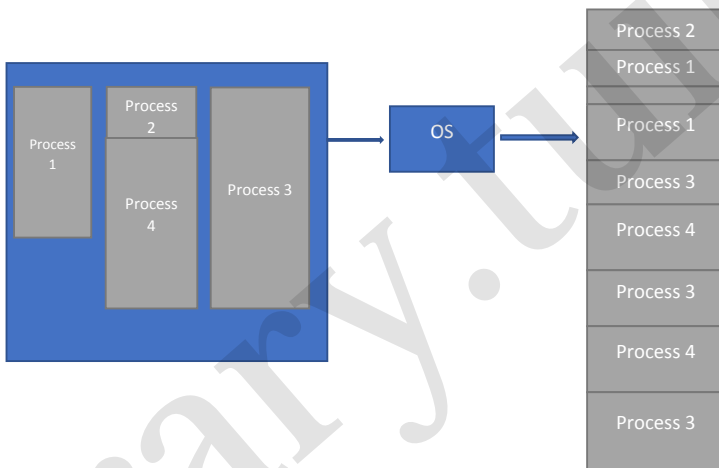
### **6.3. Ko'p vazifalilik va jarayonni boshqarish**

#### **6.3.1. O'rnatilgan OT da ko'p vazifalilik**

Ko'p vazifali OTlar uchun bir vaqtning o'zida mavjud bo'lishi mumkin bo'lgan vazifalarni boshqarish va sinxronlashtirish uchun bitta vazifali OTlarga nisbatan qo'shimcha mexanizm talab qilinadi. Buning sababi shundaki, hatto OT bir nechta vazifalarni birgalikda bajarishga

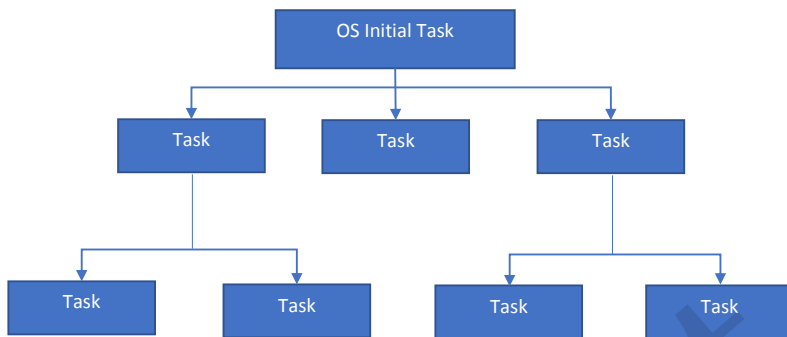
imkon bergan bo'lsa ham, o'rnatilgan taxtdagi bitta asosiy protsessor istalgan vaqtda faqat bitta vazifani bajarishi mumkin. Natijada, ko'p o'lchamli ko'milgan OT har bir vazifaga asosiy protsessorni ishlatish va asosiy protsessorni turli xil vazifalar o'rtasida almashtirish uchun ma'lum vaqt berishning biron bir yo'lini topishi kerak. Buning yordamida

vazifalarni bajarish, rejalashtirish, sinxronizatsiya va vazifalar o'rtasidagi o'zaro ta'sir orqali OT bitta protsessor bir vaqtning o'zida bir nechta vazifalarni bajarishi haqidagi illyuziyani muvaffaqiyatli yaratadi (6.7-rasm).



Rasm.6.7. Vazifalar tartibi

Ko'p funktsiyali o'rnatilgan OTlarda vazifalar ota-onalar va bolalarning vazifalari ierarxiyasi sifatida tuzilgan va o'rnatilgan yadroni ishga tushirganda bitta vazifa bo'ladi (6.8-rasm). Aynan shu birinchi vazifadan boshlab hamma yaratilgan.



Rasm. 6.8. Vazifalar ierarxiyasi

O‘rnatilgan OTlarda vazifalarni yaratish asosan ikkita modelga asoslanadi: fork/exec (IEEE / ISO POCIX 1003.1 standartidan olingan) va spawn (fork / exec-dan olingan). Spawn modeli fork/exec modeliga asoslanganligi sababli, ikkala modelda ham vazifalarni yaratish usullari o‘xshashdir. Barcha vazifalar fork/exec yoki spawn tizimiga qo‘ng‘iroqlar orqali o‘zlarining bolalar vazifalarini yaratadilar. Tizim qo‘ng‘irog‘idan so‘ng, OT nazoratni oladi va ba‘zi bir OTlarda jarayonni boshqarish birligi (PCB) deb ataladigan vazifalarni boshqarish birligini (TCB) yaratadi, unda vazifani aniqlash, vazifa holati, vazifa ustuvorligi va xato holati va kontekst kabi OT ni boshqarish haqidagi ma‘lumotlar mavjud. Ushbu aniq vazifani bajarish uchun protsessor ma‘lumotlari, masalan, registrlar. Ushbu nuqtada, yangi bola vazifasi uchun xotira ajratiladi, shu jumladan uning TCB, tizim qo‘ng‘irog‘idan o‘tgan har qanday parametrlar va bola vazifasi bajarishi kerak bo‘lgan kod. Vazifa bajarilgandan so‘ng, tizim qo‘ng‘irog‘i qaytib keladi va OT nazoratni asosiy dasturga qaytaradi.

Fork/exec va spawn modellari o‘rtasidagi asosiy farq shundaki, yangi bola vazifasi uchun xotira qanday ajratilgan. Fork/exec modelida chaqirish uchun forklar ota-ona vazifasi xotira bo‘shlig‘ining nusxasini bolalar vazifasi uchun ajratilgan narsada yaratadi, bu esa ota-ona vazifasidan turli xil xususiyatlarni, masalan dastur kodi va o‘zgaruvchilarni meros qilib olishga imkon beradi. Ota-ona vazifasining butun xotira maydoni bola vazifasi uchun takrorlanganligi sababli, ota-ona vazifasining dastur kodining ikki nusxasi xotirada, biri ota-ona

uchun, ikkinchisi esa bolaga tegishli. "Exec" qo'ng'irog'i ota-ona dasturiga berilgan har qanday havolalarni bolalar vazifasi xotirasidagi bo'sh joydan aniq o'chirish uchun ishlatiladi va bola vazifasiga tegishli yangi dastur kodini o'rnatadi.

Boshqa tomondan, avlod modeli, bola vazifasi uchun mutlaqo yangi manzillar maydonini yaratadi. Spawn tizimining chaqiruvi sizga yangi dastur va bola vazifasi uchun dalillarni aniqlash imkonini beradi. Bu sizga bola vazifasi dasturini uni yaratishda darhol yuklab olish va bajarish imkonini beradi.

Jarayonni yaratishda ikkala model ham kuchli va zaif tomonlariga ega. Urug'lantirish usuli bilan, xotira maydonlarining takrorlanadigan joylari yaratilmaydi va yo'q qilinadi, keyin fork / exec modelida bo'lgani kabi, yangi joy ajratiladi. Shu bilan birga, fork / exec modelining afzalliklari ota-ona vazifasining xususiyatlarini meros qilib oladigan va keyinchalik bola vazifasining muhitini o'zgartirish uchun moslashuvchanlikka ega bo'lgan bola vazifasi tomonidan olingan samaradorlikni o'z ichiga oladi.

### **6.3.2. Jarayonni rejalashtirish**

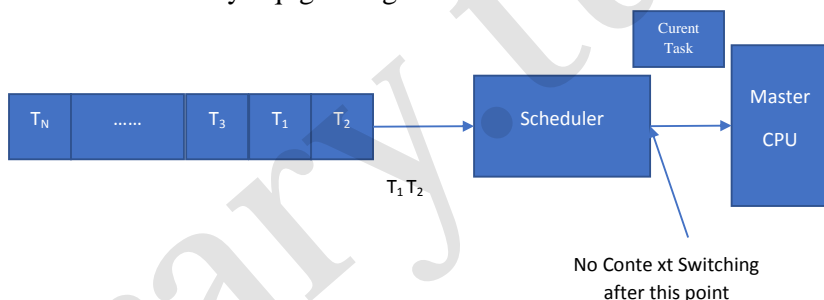
Ko'p ishlaydigan tizimda protsessor deb nomlangan mexanizm protsessorida bajariladigan topshiriqlar tartibi va muddatini belgilash uchun javobgardir. Rejalashtiruvchi qaysi vazifalar qaysi holatda bo'lishini (tayyor, ishlaydigan yoki bloklangan) tanlaydi va har bir vazifa uchun TCB ma'lumotlarini yuklab oladi va saqlaydi. Ba'zi bir operatsion tizimlarda bir xil rejalashtirgich protsessorni xotiraga yuklangan va boshlashga tayyor bo'lgan jarayonga ajratadi, boshqa operatsion tizimlarda esa protsessorni protsessorga taqsimlash uchun dispatcher (alohida rejalashtiruvchi) javob beradi.

O'rnatilgan OTda ko'plab rejalashtirish algoritmlari mavjud va har bir loyihaning afzalliklari va kamchiliklari mavjud. Rejalashtirish algoritmining samaradorligi va samaradorligiga ta'sir qiluvchi asosiy omillar uning javob vaqti (rejalashtiruvchining kontekstni tugallangan vazifaga o'tish vaqti va tugagan navbatdagi vazifani kutish vaqtini o'z ichiga oladi), burilish vaqti (jarayonni bajarish uchun zarur bo'lgan vaqt), qo'shimcha xarajatlar (zarur bo'lgan vaqt va ma'lumotlar keyingi qaysi vazifalar bajarilishini aniqlash) va adolat (qaysi omillar qaysi jarayonlar boshlanishini aniqlaydi). Rejalashtiruvchi tizim resurslaridan foydalanishni muvozanatlashi kerak - protsessorni qo'llab-quvvatlaydi,

imkon qadar ko‘proq quvvat olaman vazifalarni o‘tkazish bilan, ma'lum vaqt ichida iloji boricha ko‘p vazifalarni qayta ishlash. Ayniqsa, rejalashtiruvchi vazifaning maksimal darajasiga erishilganda, vazifa hech qachon boshlanmasa, och qolmasligini ta'minlashi kerak.

O‘rnatilgan OT bozorida, o‘rnatilgan OTlarda amalga oshiriladigan rejalashtirish algoritmlari odatda ikkita yondoshishga duchor bo‘ladi: siqilmasdan va siqilmasdan. Oldindan rejalashtirilmagan holda, vazifalarga asosiy protsessor protsessorlari bajarilish vaqti tugaguncha, vaqt qancha davom etishidan yoki boshqa kutilayotgan vazifalarning ahamiyatidan qat’iy nazar nazorat qilinadi.

Preventiv bo‘lmagan yondashuv asosida rejalashtirish algoritmlari quyidagilarni o‘z ichiga oladi: First-Come-First-Serve (FCFS)/Run-To-Completion, bu erda READY navbatdagi vazifalar navbatga kelgan tartibda bajariladi va bu vazifalar bajarilgunga qadar bajariladi va keyin tugatiladilar (6.9-rasm). Bunday holda, bu FCFS rejalashtirish loyihasida BLOCKED navbat yo‘qligini anglatadi.

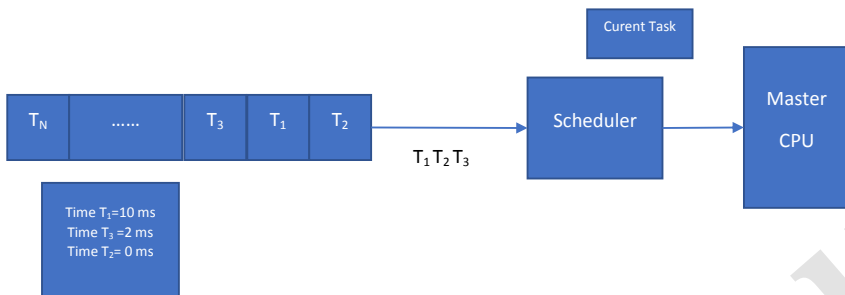


Rasm.6.9. Birinchi bo‘lib xizmat qilish jadvali

FCFS algoritmining javob vaqti odatda boshqa algoritmlarga qaraganda sekinroq (ya'ni, agar navbatda boshqa jarayonlar navbatda turishini talab qiladigan uzoqroq jarayonlar bo‘lsa), bu esa adolat muammosiga aylanadi, chunki navbat oxiridagi qisqa jarayonlar ko‘proq uchun jarima oladi. oldida uzoq. Biroq, ushbu loyiha bilan ochlikdan qutulish mumkin emas.

Qisqa jarayon bajarilishdan oldin bajariladi (SPN), bu erda READY navbatidagi topshiriqlar birinchi navbatda bajarilish muddati eng kam bo‘lgan tartibda bajariladi. (6.10-rasm).

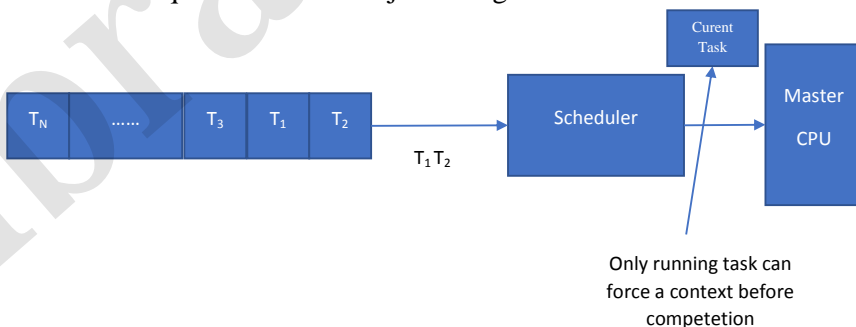




Rasm 6.10. Qisqa jarayonni eng qisqa vaqti

SPN algoritmidagi qisqa jarayonlar uchun tezkor javob vaqtlari mavjud. Biroq, uzoqroq jarayonlar navbatdagi barcha qisqaroq jarayonlar boshlanguncha kutish bilan jazolanadi. Bu holda ochlik uzoqroq jarayonlar bilan yuz berishi mumkin, agar tugagan navbat doimiy ravishda qisqa jarayonlar bilan to'ldirilsa. Xarajatlar FCFS narxidan yuqori, chunki tayyorlik navbatida jarayonni bajarish vaqtini hisoblash va saqlash kerak.

Vazifalar bajarilganda, ular OTga kontekstni o'zgartirishi mumkinligini aytguncha (ya'ni, kirish/kirish uchun va boshqalar). Ushbu algoritmda FCFS yoki SPN algoritmlari yordamida amalga oshirish mumkin, bajarilguncha bajariladigan skript emas, ammo ochlik hali ham SPN bilan ro'y berishi mumkin, masalan, agar qisqaroq jarayonlar "o'zaro ta'sir qilish" uchun mo'ljallanmagan bo'lsa.



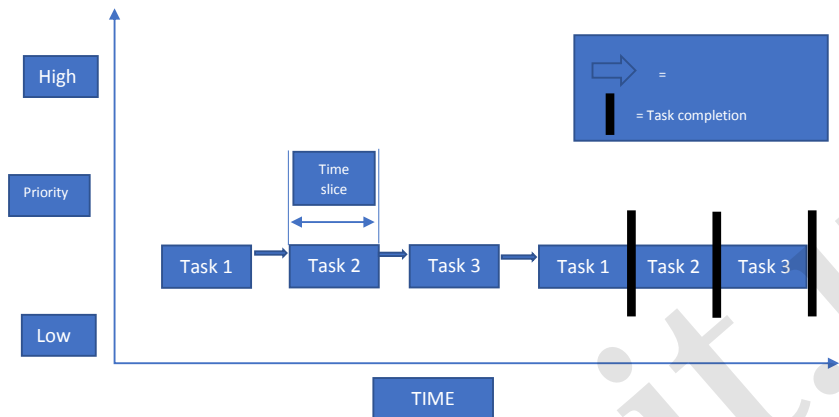
Rasm 6.11. Birgalikda rejalashtirish

Faol bo'lmagan algoritmlar qo'llab-quvvatlash uchun ko'proq xavf tug'dirishi mumkin, chunki asosiy protsessoridan qolgan barcha

vazifalarni yopib, hech qanday vazifa cheksiz pastadir bilan bajarilmaydi degan taxmin qilish kerak. Ammo, oldingi algoritmlarni qo'llab-quvvatlaydigan operatsion tizimlar vazifa tayyor bo'lgunga qadar majburiy kontekstni almashtirishga olib kelmaydi va to'liq bajarilmagan vazifalar o'rtasida almashish paytida vazifa haqida aniq ma'lumotni saqlash va tiklash bilan bog'liq xarajatlar muammodir. agar ustuvor rejalashtiruvchi qo'shma rejalashtirish mexanizmini amalga oshirsa. Proaktiv rejalashtirishda, boshqa tomondan, OT bajarilayotgan ishning tugatilishi yoki kontekstni almashtirish bilan bog'liq bo'lishidan qat'i nazar, vazifa uchun kontekstli almashtirishni keltirib chiqaradi.

### **Robin-FIFO algoritmi (birinchi kirish, birinchi chiqish)**

FIFO algoritmi tayyor jarayonlarni (bajarishga tayyor bo'lgan jarayonlar) saqlaydigan FIFO navbatini amalga oshiradi. Jarayonlar navbatning oxirida qo'shiladi va navbatning boshidan boshlash uchun olinadi. FIFO tizimida, barcha ish jarayonlari, ularning ish yuki yoki o'zaro ta'siridan qat'i nazar, bir xil tarzda amalga oshiriladi. Bu, asosan, boshqa protsessorlarning ishlashiga ruxsat berish uchun protsessor boshqaruvini qo'llab-quvvatlaydigan, hech qachon bloklanmaydigan yagona jarayonning mavjudligi bilan bog'liq. Davrni rejalashtirishda, FIFO navbatidagi har bir jarayonga teng vaqt oralig'i (har bir jarayon tugashi kerak bo'lgan vaqt) ajratiladi, bu erda har bir vaqt oxirida tanaffus jarayoni boshlanadi. (Vaqt oraliqlarini ajratadigan jadvalni tuzish algoritmlari vaqtning taqsimlash tizimlari deb ham ataladi.) Keyin, rejalashtiruvchi FIFO navbatidagi jarayonlar orasida aylanadi va navbatni boshidan boshlab jarayonlarni ketma-ket bajaradi. FIFO navbatining oxiriga yangi jarayonlar qo'shiladi va agar hozirda ishlayotgan jarayon ajratilgan vaqt oralig'ining oxiriga qadar tugamasa, u to'xtatiladi va keyingi safar kelganda bajarish uchun navbat oxiriga qaytadi. Agar jarayon belgilangan vaqt oralig'ining oxiriga qadar davom etsa, protsessor ixtiyoriy ravishda protsessorni bo'shatadi va keyin rejalashtiruvchi keyingi jarayonni protsessorga FIFO navbatini tayinlaydi (6.12-rasm).



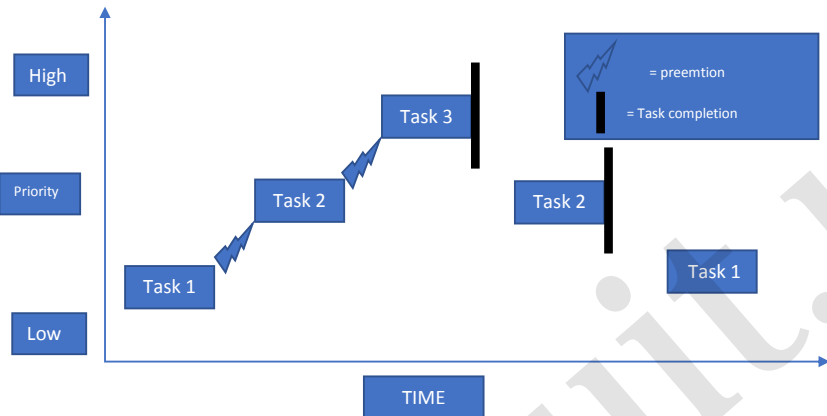
Rasm 6.12. FIFO davriy qidiruvini rejalashtirish

Round Robin/FIFO-ni rejalashtirish jarayonlarni teng ravishda qayta ishlashni ta'minlasa-da, kamchiliklar turli xil jarayonlarning yuqori yuklanishiga va doimiy uzilishlarga olib keladi, bu esa qo'shimcha kommutatsiya xarajatlarini keltirib chiqaradi. Navbatdagi jarayonlar boshqa jarayonlar bilan o'zaro ta'sirlashganda (masalan, boshqa jarayon tugashini kutayotganda) va boshqa navbat jarayoni tugamaguncha ularga biron bir ishni tugatish har doim taqiqlanadi. O'tkazish vaqti vaqt oralig'iga bog'liq. Agar vaqt oralig'i juda kichik bo'lsa, unda juda ko'p kontekstli kalitlar mavjud, juda katta vaqt oralig'i FCFS kabi siqilishsiz yondashuvdan farq qilmaydi. Dumaloq robinadan foydalanganda ochlik bo'lmaydi.

### Ustuvor (proaktiv) rejalashtirish

Rejalashtirish algoritmi ustuvor rejalashtirish algoritmi jarayonlarni bir-biriga va tizimga nisbatan muhimligiga asoslanib belgilaydi. Har bir jarayonga ustuvorlik beriladi, bu tizimdagi ustuvor buyurtmalar ko'rsatkichi bo'lib xizmat qiladi. Eng yuqori ustuvorlikdagi jarayonlar har doim ular boshlamoqchi bo'lgan paytlarda ustuvorligi pastroq bo'lgan jarayonlardan ustun turadi, ya'ni rejalashtiruvchi ustuvor vazifasi yuqori bo'lgan vazifani bajarishga tayyor bo'lsa, u davom etayotgan vazifani majburan bloklab qo'yishi mumkin. Rasmda 6.13 uchta vazifa ko'rsatilgan (1, 2, 3 - bu erda 1-vazifa eng past ustuvor vazifa, va 3-

topshiriq eng yuqori) va 3-topshiriq 2-vazifani, 2-vazifa - 1-vazifani almashtiradi.



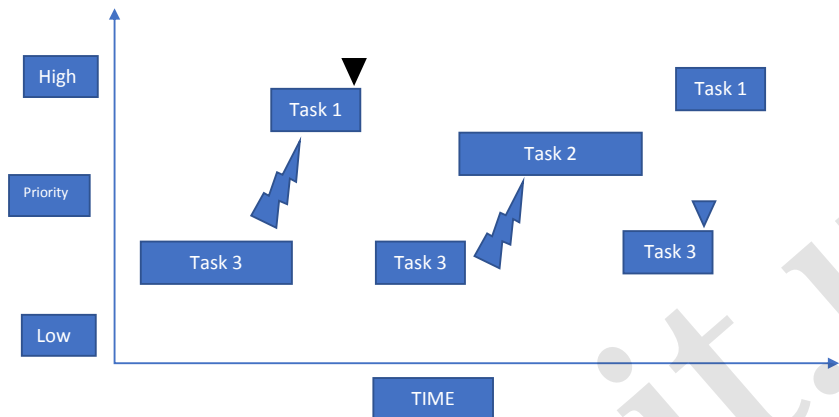
Rasm 6.13. Ustuvorlikni rejalashtirish

Ushbu rejalashtirish usuli o‘zaro ishlaydigan yoki turli xil yuklarga ega bo‘lgan jarayonlar bilan ishlashda rejalashtirishni rejalashtirish bilan bog‘liq ba‘zi muammolarni hal qilsa ham, ustuvor rejalashtirishda yangi muammolar paydo bo‘lishi mumkin, shu jumladan:

- ustuvor jarayonlarning uzluksiz oqimi ustuvorligi past bo‘lgan jarayonlarning boshlanishiga xalaqit beradigan jarayonlarning tugashi. Odatda bu eskirgan jarayonlar kamroq ahamiyatli echimlar bilan hal qilinadi (chunki bu jarayonlar navbatda ko‘proq vaqt sarflaydi, ularning ustuvorlik darajasini oshiradi);

- ustuvorliklar inversiyasi, bunda ustuvorligi past bo‘lgan jarayonlar bajarilishini kutish paytida to‘sib qo‘yilishi mumkin va boshlang‘ichda oraliq ustuvorlikli jarayonlar yuqori ustuvorlikka ega, shuning uchun ham ustuvorligi pastroq bo‘lgan jarayonlar, ham ustuvorroq bo‘lgan jarayonlar bajarilmaydi. ustuvorlik (6.14-rasm).

Qoida tariqasida, vazifa qanchalik muhim bo‘lsa, uning ustuvorligi shunchalik yuqori bo‘ladi. Bir xil darajada muhim bo‘lgan vazifalar uchun vazifalarni ustuvorlashtirish uchun ishlatilishi mumkin bo‘lgan usullardan biri bu monoton tezlikni rejalashtirish (RMS) sxemasi bo‘lib, unda tizimda bajarilish vaqtiga qarab vazifalar ustuvorlik qilinadi. Ushbu modelning sababi shundaki, proaktiv rejalashtiruvchini va to‘liq mustaqil bo‘lgan vazifalarni (umumiy



Rasm 6.14. Ustuvor inversiya

ma'lumotlar yoki manbalar mavjud emas) va vaqti-vaqti bilan bajariladigan (ya'ni muntazam ravishda bajariladigan) berilgan holda, vazifa qanchalik ko'p bajarilsa va uning ustuvorligi shunchalik yuqori bo'lishi kerak. RMS teoremasida aytilishicha, agar yuqoridagi taxminlar rejalashtiruvchi uchun belgilangan "n" vazifalar bajarilgan bo'lsa, agar barcha tengsizliklar aniqlansa, barcha muddatlar bajariladi.  $\sum E_i/T_i = n$  ( $21/n - 1$ ), bu erda  $i$  = davriy topshiriq,  $n$  = davriy topshiriqlar soni,  $T_i$  = vazifa  $i$  bajarish davri,  $E_i$  = yomon holat  $i$  vazifani bajarish vaqti,  $E_i / T_i$  = protsessor vaqtining ulushi, vazifani bajarish uchun zarur bo'lgan  $i$ .

Shunday qilib, eng kichik davrga ega bo'lgan vazifaga katta ustuvorlik berilgan davrlarga ko'ra ustuvor bo'lgan ikkita vazifani hisobga olsak, " $n(21/n - 1)$ " tengsizlikning qismi taxminan 0.828 bo'ladi, ya'ni protsessor. Barcha qat'iy belgilangan muddatlarga rioya qilish uchun ushbu vazifalardan foydalanish taxminan 82,8% dan oshmasligi kerak. Qisqa muddatlarga ega bo'lgan vazifalar yuqori ustuvorliklarga ega bo'lgan davrlarga ko'ra ustuvorliklarga ega bo'lgan 100 ta vazifalar uchun protsessorlardan foydalanish taxminan 69,6% dan oshmasligi kerak ( $100 * (21/100 - 1)$ ). barcha muddatlarni bajarish.

### Belgilangan ustuvorlik bilan OT.

OT vazifalari uchun ustuvorlik algoritmlari odatda vazifalar ishlab chiqish jarayonida ustuvorlik berilganida ustuvorliklar bilan sobit deb

tasniflanadi va ular vazifalar hayotiy tsiklida o'zgarmaydi, ustuvorliklar ish vaqtida ustuvorliklarni belgilashda dinamik ustuvorlik bilan yoki ikkala algoritmning kombinatsiyasi bilan. Ko'pgina tijorat operatsion tizimlari odatda faqat belgilangan ustuvor algoritmlarni qo'llab-quvvatlaydi, chunki bu amalga oshirish uchun eng kam murakkab sxema. Ruqsat etilgan ustuvorlik sxemasidan foydalanishning kaliti:

- topshiriqlarni o'z davrlariga ko'ra ustuvor deb belgilang, shunda qisqa muddat, ustuvorliklar shunchalik yuqori bo'ladi;

- belgilangan vazifalarni belgilash uchun belgilangan ustuvor algoritmdan (masalan, monoton tezlik algoritmi, RMS ramkasi) ustunlik qilish, shuningdek vazifalar to'plamini rejalashtirish mumkinligini tezda aniqlash vositasi;

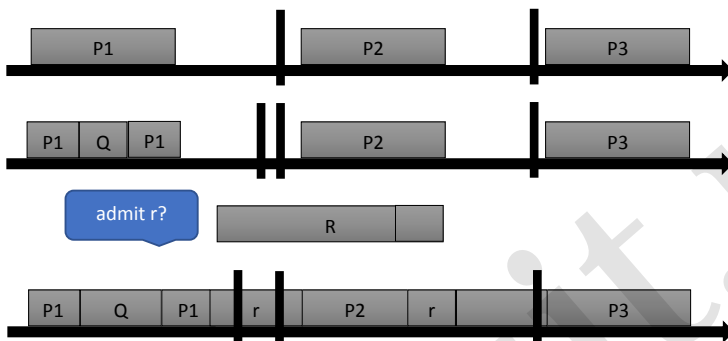
- RMS kabi muhim ustuvor algoritmning tengsizligi bajarilmasa, muayyan vazifalar to'plamini tahlil qilish talab etiladi. RMS bu ko'p hollarda protsessorni ishlatish chegarasidan past bo'lsa, ko'p hollarda belgilangan muddatlar bajarilishini taxmin qilishimizga imkon beradigan vositadir ("Ko'pchilik" holatlarda biron bir ustuvor sxemadan foydalanib rejalashtirish mumkin bo'lmagan vazifalar mavjudligini anglatadi). Protsessorning umumiy ishlatilishi tengsizlik tomonidan belgilangan chegaradan oshganiga qaramay, bir qator vazifalarni rejalashtirish mumkin. Shunday qilib, har bir topshiriqni bajarish davri va bajarilish vaqti tahlil qilinishi kerak, bunda to'plam kerakli sanalarni bajara oladimi yoki yo'qligini aniqlash kerak;

- belgilangan ustuvorlik bilan rejalashtirishning asosiy cheklanishi, asosiy protsessorni har doim ham 100% ishlatish imkoni bo'lmashligini tushunish. Agar maqsad belgilangan ustuvorliklardan foydalanganda protsessor protsessidan 100% foydalanish bo'lsa, u holda vazifalarga harmonika = davrlar tayinlanishi kerak, ya'ni vazifa davri qolgan barcha vazifalarning qisqaroq qismiga teng bo'lishi kerak.

### **EDF / Vaqt bilan ishlaydigan algoritm**

Rasmda ko'rsatilgandek (6.15 rasm), EDF / Clock-Driven algoritm uchta parametrga muvofiq jarayonlar uchun ustuvorliklarni belgilaydi: chastota (jarayonni boshlash vaqti), muddati (jarayon tugashi kerak bo'lgan vaqt) va davomiylik (jarayonni yakunlashi kerak bo'lgan vaqt). EDF algoritmi vaqt chegaralarini (asosan barcha topshiriqlar uchun kafolatlangan vaqt jadvalini) tekshirish va qo'llash imkonini beradi, ammo qiyinchilik turli xil jarayonlarning aniq muddatini belgilashda.

Odatda, oʻrtacha baho har bir jarayon uchun amalga oshiriladigan eng yaxshisidir.

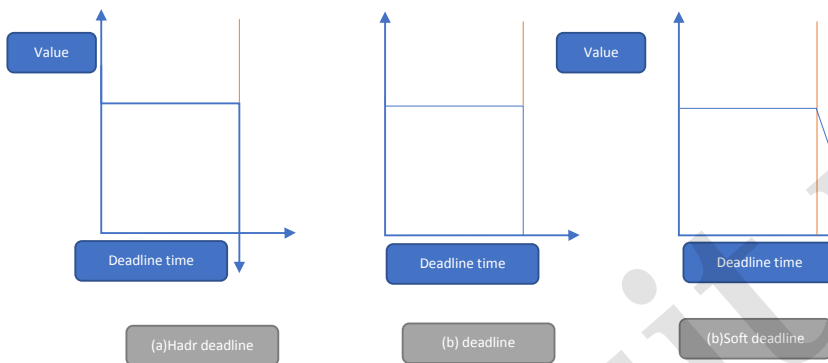


Rasm 6.15. EDF rejalashtirish algoritmi

### Faol rejalashtirish va real vaqtda ishlaydigan operatsion tizim (RTOS)

Oʻrnatilgan operatsion tizimlarda amalga oshiriladigan rejalashtirish algoritmlari oʻrtasidagi asosiy farqlardan biri bu algoritm oʻz vazifalarini belgilangan muddatlarga mos kelishini taʼminlashmi. Agar vazifalar har doim ularning belgilangan muddatlariga toʻgʻri keladigan boʻlsa (6.16-rasm) va bajarilish vaqti bilan aniqlik qilinadigan (deterministik) boʻlsa, OT real vaqtda operatsion tizim (RTOS) deb nomlanadi.

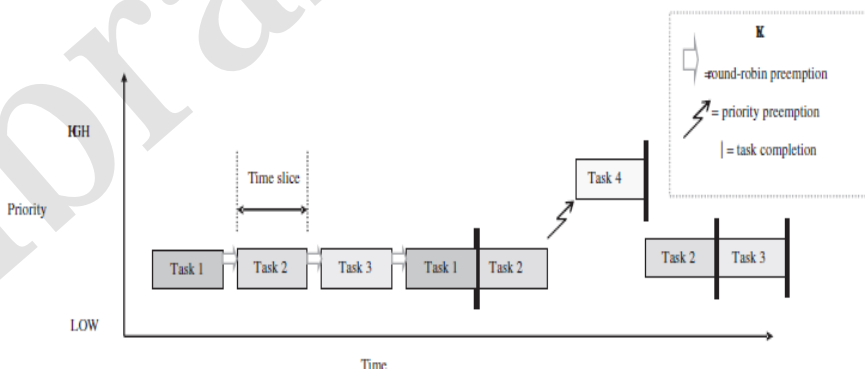
Faol rejalashtirish RTOS jadvallarida amalga oshiriladigan algoritmlardan biri boʻlishi kerak, chunki real vaqtda talablar qoʻyilgan vazifalar boshqa vazifalarga nisbatan ustunlikka ega boʻlishi kerak. RTOS rejalashtiruvchilari shuningdek, oʻzlarining qatʼiy muddatlarini boshqarish va bajarish uchun tizim soatlariga asoslangan oʻzlarining taymerlaridan foydalanadilar. U RTOS boʻladimi yoki real vaqt rejimida emasmi, rejalashtirish nuqtai nazaridan, hamma uning amalga oshirilayotgan rejalashtirish sxemalarida farq qiladi. Masalan, vxWorks (Wind River) ustuvor va doiraviy sxema, Jbed (Esmertec) - EDF sxemasi, Linux (Timesys) esa ustuvorlikka asoslangan sxema.



Rasm 6.16. Operatsion tizimlar va vaqtlari

### vxWorksni rejalashtirish

Wind rejalashtiruvchisi ikkala ustuvor algoritmlarga va real vaqtda reja tuzish algoritmlariga asoslanadi. 6.17-rasmda ko'rsatilgandek, tsiklni rejalashtirish ustuvor rejalashtirish bilan birlashtirilishi mumkin, bir xil ustuvorlikdagi vazifalarni asosiy protsessorni almashish uchun, shuningdek ustuvor vazifalar protsessorni yukdan tushirishga imkon beradi.



Rasm 6.17. Davriy rejalashtirish bilan to'ldiriladigan ustuvorlikni rejalashtirish

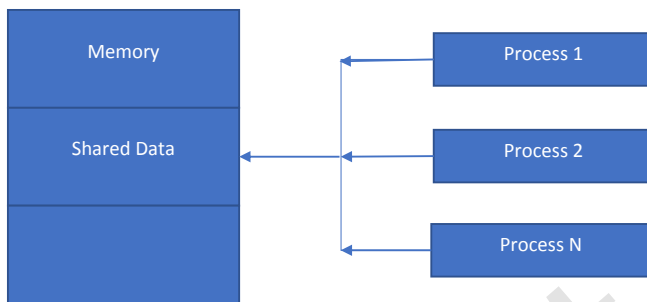


Davrni rejalashtirishsiz, vxWorks-da bir xil ustuvorlikka ega bo'lgan vazifalar hech qachon bir-birini ustun qo'ymaydi, agar dasturchi ushbu vazifalardan birini cheksiz tsiklda bajarish uchun ishlab chiqsa, muammo bo'lishi mumkin. Shu bilan birga, ustuvor ustuvor rejalashtirish vxWorksga real vaqtda o'z imkoniyatlaridan foydalanishga imkon beradi, chunki vazifalar belgilangan muddatni hech qachon o'tkazib yubormaslik uchun dasturlashtirilishi mumkin va ularga boshqa vazifalarni oldindan ko'rish uchun ustunliklar beriladi. Vazifalarni yaratish jarayonida taskSpawn buyrug'i yordamida vazifalar ustuvorlashtiriladi.

### **6.3.3. O'zaro amallar faoliyati va sinxronizatsiya**

O'rnatilgan tizimdagi turli xil vazifalar odatda bir xil apparat va dasturiy resurslarni almashishi kerak yoki to'g'ri ishlashi uchun bir-biriga ishonishi mumkin. Shu sabablarga ko'ra, o'rnatilgan OTlar ko'p vazifali tizimdagi vazifalarni o'zlarining funksiyalarini muvofiqlashtirish, muammolardan qochish va vazifalarni bir vaqtning o'zida uyg'unlikda ishlashiga imkon berish uchun o'zaro munosabat va sinxronlashtirishni ta'minlaydigan turli xil mexanizmlarni ta'minlaydi. Bir nechta o'zaro ta'sir jarayonlariga ega o'rnatilgan operatsion tizimlar, odatda, xotira almashish, xabarlarini uzatish va signalizatsiya mexanizmlarining bir yoki bir nechta kombinatsiyasiga asoslangan interrossess aloqa (IPC) va sinxronizatsiya algoritmlarini amalga oshiradilar. 6.18-rasmda ko'rsatilgan ma'lumotlar almashish modelidan foydalanib, bitta jarayon orqali o'zgartirilgan o'zgaruvchilar barcha jarayonlar uchun mavjud bo'lgan umumiy xotiraga kirish orqali ma'lumotlar almashinuvini amalga oshiradi.

Aloqa vositasi sifatida umumiy ma'lumotlarga kirish oddiy yondashuv bo'lsa-da, jiddiy musobaqada irqi muammosi paydo bo'lishi mumkin. Musobaqa sharti umumiy o'zgaruvchilarga kirish jarayoni o'zgarishga kirishni tugatishdan oldin to'xtatilganida yuzaga keladi, bu umumiy o'zgaruvchilarning yaxlitligiga ta'sir qiladi. Ushbu muammoni hal qilish uchun, tanqidiy bo'limlar deb nomlangan umumiy ma'lumotlarga kiradigan jarayonlarning qismlari o'zaro eksklyuziv tarzda



Rasm. 6.18. Xotirani kelishilgan holda ishlatish

ishlab chiqilishi mumkin (yoki qisqasi Mutex).

**Mutex mexanizmlari** umumiy xotirani unga kirish protsessi tomonidan blokirovka qilishiga imkon beradi va bu jarayonga umumiy ma'lumotlarga eksklyuziv kirish huquqini beradi. Birgalikda ishlatiladigan turli xil mexanizmlar nafaqat umumiy xotiraga kirishni muvofiqlashtirish uchun, balki boshqa umumiy tizim resurslariga kirishni muvofiqlashtirish uchun ham amalga oshirilishi mumkin. Birgalikda ishlatiladigan ma'lumotlarga bir vaqtning o'zida kirishni talab qiladigan vazifalarni sinxronlashtirish uchun o'zaro istisno qilish usullari quyidagilarni o'z ichiga olishi mumkin:

- CPU boshqa hech qanday vazifani oldinroq bajara olmaydigan tarzda rejalashtirilgan umumiy ma'lumotlarga kirish huquqini beradigan vazifalarni qulflaydi;

- Kontekstni almashtirishga olib keladigan boshqa yagona mexanizmlar uzilishlardir. Agar tanqidiy qismda kodni bajarishda uzilishlar o'chirilsa, agar to'xtatuvchiga ishlov beruvchilar bir xil ma'lumotlarga ega bo'lishsa, poyga shartlari stsenariysining oldini oladi.

Protsessor tomonidan blokirovka qilinishi mumkin bo'lgan yana bir narsa bu "**sinov va o'rnatish bo'yicha ko'rsatma**" mexanizmi (shartli o'zgaruvchan sxemasi deb ham ataladi). Ushbu mexanizmga muvofiq, ro'yxatdan o'tish bayrog'ini (holatini) sozlash va tekshirish elementar funktsiya bo'lib, jarayonni to'xtatib bo'lmaydi va bu bayroq tanqidiy qismga kirishni istagan har qanday jarayon tomonidan tekshiriladi. Muxtasar qilib aytganda, to'xtatish sxemasi ham, shartli o'zgaruvchan blokirovka sxemalari ham xotiraga alohida kirish huquqini kafolatlaydi,

bu erda hech narsa umumiy ma'lumotlarga kirishga to'sqinlik qilmaydi va tizim kirish paytida boshqa biron-bir hodisaga javob bera olmaydi.

Birgalikda ishlatiladigan xotiraga kirishni blokirovka qilish (o'zaro chiqarib tashlash), shuningdek, ishlaydigan jarayonlarni tashqi hodisalar bilan muvofiqlashtirish (sinxronizatsiya) uchun ishlatiladigan semaforalar. Semafora funktsiyalari atom funktsiyalari bo'lib, odatda tizim qo'ng'iroqlari orqali jarayon orqali chaqiriladi.

### **VxWorks semaforlari**

VxWorks semaforalarning uch turini belgilaydi:

• **Ikkilik semaforalar** mavjud yoki mavjud bo'lmagan, o'rnatilishi mumkin bo'lgan ikkilik (0 yoki 1) bayroqlardir. O'zaro o'chirish mexanizmi sifatida ikkilik semafora ishlatilganda (masalan, protsessor tomonidan blokirovka qilinsa, tizimdagi boshqa bog'liq bo'lmagan manbalar ham ta'sir qilishi mumkin) o'zaro bog'liqlik faqat tegishli manbaga ta'sir qiladi. Ikkilik semafora dastlab manbaning mavjudligini ko'rsatadigan = 1 (to'liq) ga o'rnatiladi. Vazifalar kirish zarur bo'lganda manbaning ikkilik semaforini tekshiradi va agar mavjud bo'lsa, manbaga kirishda bog'liq semafori oling (ikkilik semafori o'rnatish = 0) va keyin manba bilan ishlashni tugatgandan so'ng uni qaytaring (ikkilik semafori o'rnatish = 1). Ikkilik semafora vazifalarni sinxronlashtirish uchun foydalanilganda, dastlab 0 (bo'sh) qilib o'rnatiladi, chunki u boshqa vazifalar kutgan voqea sifatida ishlaydi. Muayyan ketma-ketlikda bajarilishi kerak bo'lgan boshqa vazifalar, keyin ikkilik semafori 1 vazifani kuting (voqea sodir bo'lgunga qadar) semafori asl topshiriqdan olib, uni 0 ga qaytaring. Quyidagi vxWorks psevdokod misolida binar qanday qilib ko'rsatilgan. semaforlardan vazifalarni sinxronlashtirish uchun vxWorks-da foydalanish mumkin.

• **O'zaro istisno semaforalari** bu vxWorks dasturini rejalashtirish modelida yuzaga kelishi mumkin bo'lgan o'zaro echim muammolari uchun ishlatilishi mumkin bo'lgan ikkilik semaforalar, masalan: ustuvor inversiya, o'chirish xavfsizligi (tanqidiy qismga kiradigan va boshqa vazifalarni blokirovka qiladigan ishlarni ta'minlamaydi). kutilmagan tarzda o'chiriladi) va manbalarga rekursiv kirish. Quyida vazifa usullari tomonidan rekursiv ravishda ishlatiladigan semafore psevdokodining o'zaro eksklyuzivligiga misol keltirilgan.

• **Hisoblash semaforlari** ikkita bog'liq funktsiyaga ega bo'lgan musbat butun son hisoblagichlardir: ortish va kamayish. Semaphore hisoblash odatda resurslarning ko'p nusxalarini boshqarish uchun

ishlatiladi. Resurslarga kirish kerak bo'lgan vazifalar semaforini qisqartiradi. Vazifalar resursni bo'shatganda, semaforning qiymati oshadi. Semafora "0" qiymatiga yetganda, tegishli kirishni kutadigan har qanday vazifa boshqa vazifa semaforini qaytarib bermaguncha bloklanadi.

Xulosa qilib shuni ta'kidlaymizki, o'zaro algoritmlar yordamida faqat bitta jarayon istalgan vaqtda umumiy xotiraga kirish huquqiga ega bo'lishi mumkin, asosan xotiraga kirish qulfi mavjud. Agar bir nechta protseduralar bloklangan bo'lsa, ular birgalikda xotiraga kirish uchun navbatni kutib, bir-birining ma'lumotlariga tayanib, muammoga duch kelishi mumkin (masalan, ustuvor rejalashtirishda ustuvor inversiya). Shunday qilib, o'rnatilgan OTlar to'siqni oldini olish mexanizmlarini, shuningdek, inqirozdan qutulish mexanizmlarini ta'minlashi kerak. Yuqoridagi misollardan ko'rinib turibdiki, vxWorks to'siqlarning oldini olish va oldini olish uchun semaforlardan foydalanadi. Xabarlar orqali o'zaro ishlash - bu algoritim bo'lib, unda xabarlar (ma'lumotlar bitlaridan iborat) jarayonlar o'rtasida xabarlar navbatiga yuboriladi. OT manzillarga yuborish va autentifikatsiya qilish uchun protokollarni jarayonlarga ishonchli etkazib berilishini ta'minlash uchun, shuningdek navbatga yozilishi mumkin bo'lgan xabarlar soni va xabarlarning o'lchamlarini belgilaydi. Ushbu sxema bo'yicha OT vazifalari xabarlarni navbatga yuboradi yoki ma'lumot almashish uchun navbatdan xabar oladi. Mikrokernelga asoslangan operatsion tizimlar odatda sinxronizatsiya asosiy mexanizmi sifatida xabarlar sxemasidan foydalanadilar.

#### **6.4. Xotirani boshqarish**

Ushbu bobda ilgari aytib o'tilganidek, yadro o'rnatilgan tizim ichidagi kodni vazifalar orqali boshqaradi. Yadro, shuningdek, tizimda qandaydir yuklash va vazifalarni bajarishga ega bo'lishi kerak, chunki protsessor faqat kesh yoki RAMdagi vazifa kodini bajaradi. Xuddi shu xotira maydonini taqsimlaydigan bir nechta vazifalardan foydalanishda OT, vazifa kodini boshqa mustaqil vazifalardan himoya qilish uchun xavfsizlik mexanizmini talab qiladi. Bundan tashqari, operatsion tizim boshqariladigan vazifalar bilan bir xil xotira maydonida bo'lishi kerakligi sababli, xavfsizlik mexanizmi xotirada mahalliy kodni boshqarish va uni boshqaradigan vazifa kodidan himoya qilishni o'z ichiga olishi kerak. Aynan shu funktsiyalar va boshqa ko'p narsalar OT

xotirasini boshqarish komponentlari uchun javobgardir. Umuman olganda yadro xotirasini boshqarish vazifalariga quyidagilar kiradi:

- Mantiqiy (jismoniy) xotira va vazifalar xotirasi aloqalari o'rtasidagi xaritalashni boshqarish.
- Mavjud xotira maydoniga qaysi jarayonlar yuklanishini aniqlash.
- Tizimga kiradigan jarayonlar uchun xotirani ajratish va bo'shatish.
- Xotirani ajratish va bo'shatish kod so'rovlarini qo'llab-quvvatlash (jarayonning bir qismi sifatida), masalan, C ajratish va ajratish funksiyalari yoki maxsus buferni ajratish va tarqatish protseduralari.
- Tizim komponentlari tomonidan xotira sarfini kuzatish.
- Kesh tutarliligini ta'minlash (kesh bilan ishlaydigan tizimlar uchun).
- Jarayon xotirasini himoya qilishni ta'minlash.

Fizik xotira ikki o'lchovli massivlardan iborat bo'lib, unda noyob qator va ustun bilan murojaat qilingan kataklardan iborat bo'lib, unda har bir katak 1 bit saqlashi mumkin. Shunga qaramay, OT xotirani xotira kartasi deb nomlangan bitta katta o'lchovli qator sifatida ko'radi. Asosiy protsessorga yoki doskaga o'rnatilgan apparat komponenti mantiqiy va jismoniy manzillar (masalan, MMU) o'rtasida ishlashni amalga oshiradi yoki OT orqali qayta ishlanishi kerak.

Operatsion tizimlar mantiqiy xotira maydonini boshqarish tizimidan farqli o'laroq farq qiladi, ammo yadrolar odatda yadro kodini yuqori darajadagi jarayonlardan alohida xotira maydonida boshqaradi. Ushbu xotira bo'shliqlarining har biri (yadro kodini o'z ichiga olgan yadro va yuqori darajadagi jarayonlarni o'z ichiga olgan foydalanuvchi) boshqacha boshqariladi.

Aslida, aksariyat OT jarayonlari odatda ikkita rejimning birida ishlaydi:

- 1) yadro rejimi;
- 2) maxsus rejim (bajarilayotgan tartiblarga qarab).

Yadro dasturlari yadro rejimida (shuningdek, nazoratchi rejimi deb ataladi), dasturiy ta'minot yoki dasturlar kabi yuqori darajadagi dasturlardan farqli ravishda bo'shliq va xotira darajasida ishlaydi. Odatda, ushbu yuqori darajadagi dasturiy ta'minot foydalanuvchi rejimida ishlaydi va faqat yadro rejimida ishlaydigan har qanday kishiga tizim qo'ng'iroqlari, yuqori darajadagi interfeyslarni yadro dasturlari orqali kirishi mumkin. Yadro o'zi uchun ham, foydalanuvchi jarayonlari uchun ham xotirani boshqaradi.

## Foydalanuvchi xotira maydoni

Ishlov berish uchun operativ xotiraga yuklashda bir nechta jarayonlar bir xil jismoniy xotiradan foydalanganligi sababli, himoya mexanizmlari ham bo'lishi kerak, shuning uchun jarayonlar bir xil jismoniy xotira joyidan joy almashishda bir-biriga ta'sir eta olmaydi. Ushbu muammolar odatda operatsion tizim tomonidan "xotira" xotirasini almashish orqali, ish vaqtidagi xotira bo'limlari o'zgartirilganda va xotiradan tashqarida bo'lganda hal qilinadi. Almashishda ishlatiladigan eng keng tarqalgan xotira bo'limlari segmentlar (jarayonlarni qismlarga ajratish) va sahifalar. Segmentatsiya va peyjing nafaqat xotirani ajratishni soddalashtiradi va xotirada bo'sh joyini bo'shatadi, balki kod va xotira himoyasini qayta ishlatishga imkon beradi, shuningdek virtual xotira uchun asos yaratadi. Virtual xotira - bu OC tomonidan boshqariladigan mexanizm bo'lib, u cheklangan qurilma xotirasi maydonini bir nechta raqobatdosh "foydalanuvchi" vazifalari bilan birlashtirishga imkon beradi, bu esa qurilmaning jismoniy xotirasi maydonini "virtual" xotira maydoniga sezilarli darajada oshiradi.

## Segmentatsiya

Ushbu bobning oldingi qismida aytib o'tilganidek, jarayon dasturni bajarishda ishtirok etadigan barcha ma'lumotlarni, shu jumladan dastlabki kodni, stekni, ma'lumotlarni va boshqalarni qamrab oladi. Jarayondagi barcha turdagi ma'lumotlar o'zgaruvchan o'lchamlarga ega "mantiqiy" xotira birliklariga bo'linadi, bu segmentlar deb nomlanadi. **Segment** - bu bir xil turdagi ma'lumotlarni o'z ichiga olgan mantiqiy manzillar to'plami. Segment manzillari 0 dan boshlanadigan mantiqiy manzillar bo'lib, segmentning bazaviy manzilini va segmentning siljishini (ofsetini) ko'rsatadigan, jismoniy xotiraning haqiqiy manzilini aniqlaydigan segment raqamidan iborat. Segmentlar mustaqil ravishda himoyalangani, ya'ni ular umumiy (masalan, boshqa segmentlar ushbu segmentga kira oladigan holatlarda), Faqat o'qish yoki o'qish / yozish kabi erkinlik xususiyatlariga ega bo'lishlarini anglatadi.

Ko'pgina operatsion tizimlar, odatda, jarayonlarga segmentlarning arkibidagi besh turdagi ma'lumotlarning barchasini yoki bir qismini birlashtirishga imkon beradi: matn segmenti (yoki kod), ma'lumotlar segmenti, bss segment (belgilar bilan boshlanadigan blok), suyak segmenti va to'plangan segment. **Matn segmenti** - bu original (dastlabki) kodni o'z ichiga olgan xotira maydoni. **Ma'lumotlar segmenti** - bu dastlabki kodning boshlang'ich parametrlari

(ma'lumotlari) bo'lgan xotira maydoni. **Bss segment** bu statik ajratilgan xotira maydoni bo'lib, unda boshlang'ich kodlanmagan o'zgaruvchi (ma'lumotlar) mavjud. Ma'lumotlar, matnlar va bss segmentlar hammasi ham kompilyatsiya vaqtida o'lchamlari bo'yicha o'rnatiladi va shunday statik segmentlardir; aynan shu uch segment, odatda bajariladigan faylning bir qismi. Amalga oshiriladigan fayllar, ular tuzilgan segmentlarida farq qilishi mumkin, ammo umuman olganda sarlavha va segmentlar turlarini, shu jumladan segment bir yoki bir nechta bo'limlardan iborat bo'lishi mumkin bo'lgan nomi, ruxsatlarini o'z ichiga olgan turli bo'limlarni o'z ichiga oladi. OT bajariladigan fayl tarkibini aks ettiruvchi xotiradan vazifaning tasvirini yaratadi, bu bajariladigan faylda ko'rsatilgan segmentlarni (bo'limlarni) xotiraga yuklash va izohlashni anglatadi. Ichki o'rnatilgan OT tomonidan qo'llab-quvvatlanadigan bir nechta bajariladigan fayl formatlari mavjud, ularning eng keng tarqalgani:

- **ELF** (Bajariladigan va bog'langan format): UNIX-ga asoslangan bo'lib, ELF sarlavhasi, dastur sarlavhalari jadvali, bo'lim sarlavhalari jadvali, ELF bo'limlari jadvallari va ELF segmentlari mavjud. Linux (Timesys) va vxWorks (WRS) - bu ELF-ni qo'llab-quvvatlaydigan OClarning namunalari.

- **Sinf (Java bytecode)**: sinf fayli 8 bitli bayt oqimi sifatida bitta Java sinfini batafsil tavsiflaydi ("bytecode" nomi). Segmentlar o'rniga sinf faylining elementlari element deb ataladi. Java sinf fayl formati sinfning tavsifini, shuningdek, ushbu sinf boshqa sinflar bilan qanday bog'liqligini o'z ichiga oladi. Sinf faylining asosiy tarkibiy qismlari - bu ramzlar jadvali (konstantalar bilan), maydonlarning deklaratsiyasi, usullarni amalga oshirish (kod) va ramziy havolalar (boshqa sinflarning havolalari joylashgan joylarda). Jbed RTOS Java bayt kod formatini qo'llab-quvvatlovchi misoldir.

- **COFF** (umumiy fayl formati); Sinf fayl formati (boshqa narsalar qatorida) fayl imzolari, COFF sarlavhasi, ixtiyoriy sarlavha va faqat COFF sarlavhasini o'z ichiga olgan ob'ekt fayllarini o'z ichiga olgan rasm faylini belgilaydi.

Boshqa tomondan, stack va yig'ish segmentlari kompilyatsiya vaqtida o'rnatilmaydi va ish vaqtida hajmi o'zgarishi mumkin, shuningdek dinamik tarqatish komponentlari. **Stek segmenti** - bu LIFO navbati sifatida tuzilgan (birinchi tartibda oxirgi) xotira bo'limi bo'lib, unda ma'lumotlar stekka "joylashtirilgan" yoki stekdan "tushirilmagan" (push va pop faqat stek bilan bog'liq bo'lgan ikkita operatsiyadir).

Dasturlar ichida aniqlik qilinadigan ma'lumotlarga (masalan, mahalliy parametrlarga, o'tish parametrlariga va hokazo) bo'sh joy ajratish va bo'shatish uchun dasturlardan oddiy va samarali usul sifatida foydalaniladi. Stekda barcha ishlatilgan va bo'shatilgan xotira maydoni ketma-ket xotira maydoniga joylashtiriladi. Biroq, "push" va "pop" stek bilan bog'liq bo'lgan ikkita operatsiya bo'lgani uchun, stek o'z maqsadlari uchun cheklanishi mumkin.

**To'plangan segment** - bu ish vaqti davomida bloklarga ajratilishi mumkin bo'lgan xotira bo'limi va odatda bepul ulangan xotira qismlari ro'yxati sifatida o'rnatiladi. Bu erda xotirani boshqarish vositalari «malloc» C funksiyasini (masalan) yoki OT buferini taqsimlash funksiyalarini qo'llab-quvvatlash uchun ajratish uchun ishlatiladi. Xotirani ajratishning odatiy sxemalariga quyidagilar kiradi:

- **FF** (birinchi yondashuv) - bu birinchi katta "teshik" uchun ro'yxat boshidan skanerdan o'tkaziladigan algoritmi.

- **NF** (keyingi fitna), bu erda ro'yxat oxirgi "teshik" uchun oxirgi qidiruv tugagan joydan skanerlanadi, etarlicha katta.

- **BF** (eng yaxshi moslik), bu erda barcha ro'yxat yangi ma'lumotlarga eng mos keladigan teshikni qidiradi.

- Ma'lumotlar mavjud bo'lgan eng katta "teshikka" joylashtiradigan **WF** (yomon holat).

- **QF** (tez moslash), bu erda xotira o'lchamlari ro'yxati saqlanadi va ushbu ma'lumotlardan tanlash amalga oshiriladi.

- Blok daraja 2 darajasida taqsimlanadigan do'st tizim. Blok bo'shatilganda u qo'shni blok bilan birlashtiriladi.

Xotirani bo'shatish usuli, endi uyumda kerak emas, operatsion tizimga bog'liq. Ba'zi bir operatsion tizimlar foydalanilmagan xotirani avtomatik ravishda tiklaydigan axlat yig'uvchi vositani ta'minlaydi. Boshqa operatsion tizimlar dasturchi tizim qo'ng'irog'i orqali aniq xotirani bo'shatishni talab qiladilar. Eng yangi texnologiyalardan foydalangan holda, dasturchi xotira etishmovchiligining mumkin bo'lgan muammolari haqida xabardor bo'lishi kerak, bu erda xotira yo'qoladi, chunki u ajratilgan, ammo endi ishlatilmagan va unutilgan, axlat yig'uvchida kamroq bo'ladi.

Boshqa bir muammo, ajratilgan va bo'shatilgan xotira xotira bo'linishiga olib kelganda yuzaga keladi, to'plangan bo'sh xotira bir qator teshiklarga joylashtirilgan va bu kerakli hajmdagi xotirani ajratishni qiyinlashtiradi. Bunday holda, ajratish / bo'shatish algoritmlari juda ko'p qismlarga bo'lishga olib keladigan bo'lsa, xotirani siqish



algoritmi amalga oshirilishi kerak. Ushbu muammoni axlat yig'ish algoritmlarini o'rganish orqali namoyish etish mumkin. Axlat yig'ish algoritmi xotiraning boshqa qismiga bog'langan narsalardan nusxa ko'chirish va keyinchalik asl xotira maydonini bo'shatish orqali ishlaydi. Ushbu algoritm ishlashda xotiraning katta maydonidan foydalanadi va nusxa ko'chirish paytida odatda uni to'xtatib bo'lmaydi (bu tizimlarni bloklaydi). Shu bilan birga, yangi xotira maydonida ob'ektlarni siqish orqali qanday turdagi xotiradan samarali foydalanilishini ta'minlaydi. Yorliqlarni yig'ish va axlatni belgilash algoritmi barcha ishlatilgan ob'ektlarni "markirovka qilish" bilan, so'ngra belgilanmagan narsalarni "supurish" (chiqarish) bilan ishlaydi. Ushbu algoritm odatda bloklanmagan, shuning uchun tizim kerak bo'lganda boshqa funktsiyalarni bajarish uchun axlat yig'uvchilarni to'xtatishi mumkin. Ammo, bu nusxa ko'chirish uchun axlat yig'ish vositasi kabi ixcham xotira emas, bu esa foydalanilmagan ob'ektlar ilgari ishlatilgan joyda mavjud bo'lishi mumkin bo'lgan kichik ishlatilmagan teshiklari bilan xotiraning parchalanishiga olib keladi. Belgilash va markalash uchun axlat yig'ish vositasidan foydalanib, qo'shimcha xotira siqish algoritmi amalga oshirilishi mumkin, bu esa uni yorliq va ixcham algoritmgaga aylantiradi. Va nihoyat, avlod axlat yig'ish algoritmi ob'ektlarni xotira uchun ajratilgan vaqtga qarab avlodlar deb nomlangan guruhlariga ajratadi. Ushbu algoritm tanlangan ob'ektlarning aksariyati qisqa umr ko'rishini anglatadi, shuning uchun qolgan ob'ektlarni uzoqroq xizmat qilish muddati bilan nusxalash yoki ixchamlashtirish vaqtni yo'qotishdir. Shunday qilib, yosh avlod guruhidagi narsalar eski avlod guruhlaridagi narsalarga qaraganda tez-tez tozalanadi. Ob'ektlarni yosh avloddan katta avlod guruhiga o'tkazish ham mumkin. Avlod uchun har bir axlat yig'uvchi shuningdek yuqorida tavsiflangan nusxa ko'chirish algoritmi yoki ajratish va tozalash algoritmlari kabi har bir avlod guruhidagi ob'ektlarni tarqatish uchun turli xil algoritmlardan foydalanishi mumkin. Parchalanish muammolaridan qochish uchun ikkala avlodda ham siqish algoritmlari kerak.

### **Peyjing va virtual xotira**

Segmentatsiya bilan yoki bo'lmasdan ham, ba'zi bir operatsion tizimlar mantiqiy xotirani bloklar, ramkalar, sahifalar yoki bir nechta yoki ularning barchasining birikmasidan iborat bo'lgan sobit o'lchamdagi bir qator qismlarga bo'lishadi. Masalan, xotirani freymlarga ajratuvchi OTlarda mantiqiy adres kadr raqami va ofset bilan

buziladi. Keyin foydalanuvchining xotira maydonini sahifalarga ajratish mumkin, bu erda sahifalar o'lchamlari odatda kvadrat o'lchamlariga teng bo'ladi.

Jarayon to'liq xotiraga (sahifalar ko'rinishida) yuklanganda, uning sahifalari doimiy kadrlar ichida joylashtirilmaligi mumkin. Har bir jarayonda uning sahifalari va har bir sahifaning tegishli xotiralarini kuzatib boradigan u bilan bog'liq jarayon jadvali mavjud. Yaratilgan mantiqiy manzil bo'shliqlari har bir jarayon uchun noyobdir, hatto bir nechta jarayonlar bir xil jismoniy xotirada bo'lsa ham. Mantiqiy manzil bo'shliqlari, odatda, ushbu sahifaning boshini ko'rsatadigan sahifa ramkasi raqamidan va ushbu sahifadagi haqiqiy xotiraning joylashgan joyidan iborat. Aslida, mantiqiy manzil bu sahifa raqami va ofsetning yig'indisidir.

OT ishga tushirish uchun zarur bo'lgan sahifalarni tayyorlash yoki yuklash bilan boshlashi mumkin, so'ngra jarayon xotirasida sahifalar bo'lmagan sahifalar va sahifalar ishlamay qolganda faqat RAM-ga yuklangan talab sahifasini o'rnatish sxemasini amalga oshirish mumkin (agar siz RAM-da bo'lmagan sahifaga kirishga urinayotganingizda xato yuzaga kelsa). ) Sahifada xato yuzaga kelganda, OC kerakli sahifani egallaydi va xotiraga yuklaydi, sahifalar jadvalarini yangilaydi va keyin sahifani yangilashga olib keladigan ko'rsatmalarni yana boshlaydi. Ushbu sxema Knutning kommutatsiya nazariyasiga asoslanib, tizim vaqtining 90 foizini kodning atigi 10 foizini qayta ishlashga sarflashini taxmin qiladi.

Mantiqiy xotirani sahifalarga ajratish operatsion tizimga almashtirish jarayonida xotira ierarxiyasidagi turli xil xotiralarga ko'chirilgan va undan kelgan vazifalarni osonroq boshqarishga yordam beradi. Qaysi sahifalar almashtirilganligini aniqlash uchun umumiy sahifani tanlash va almashtirish sxemalari quyidagilardan iborat:

- Optimal, kelajakda hisoblashni ishlatib, yaqin kelajakda foydalanilmaydigan sahifalarni almashtirish.

- Yaqinda ishlatilgan sahifalarni buzadigan Least Used (LRU).

- Uning nomidan ko'rinib turibdiki, tizimdagi eng eski (ularga qancha kirish kerak bo'lishidan qat'i nazar) sahifalarni almashtirgan FIFO. Oddiy LRU algoritmi bo'lsa-da, FIFO unchalik samarasiz.

- Yaqinda ishlatilmadi (NRU), ma'lum vaqt davomida ishlatilmagan sahifalarni siqib chiqaradi.

- Ikkinchi imkoniyat, agar "0" o'zgartirilsa, mos yozuvlar bitli FIFO pallasi (boshqarish biti kirish paytida "1" ga o'rnatiladi va tekshiruvdan so'ng "0" holatiga qaytariladi).

- Soat blokirovkasi, sahifalar soat bo'yicha (ular xotirada qancha vaqt bo'lgan bo'lsa), agar ular mavjud bo'lmaganda, xushmuomalalik bilan almashtirildi (kirish amalga oshirilganda boshqaruv tugmasi "1" ga o'rnatildi va "0" tasdiqlangandan keyin).

Har bir OT o'z almashtirish algoritmiga ega bo'lsa-da, har bir kishi tizimning resurslari OT tomonidan yo'q bo'lib, doimiy ravishda xotiradan ma'lumotlarni o'zgartiradigan vaziyatni qiyshayish ehtimolini kamaytirishga harakat qilmoqda. Ishqalanishni oldini olish uchun yadro doimiy ravishda ishlov berish sahifalarining doimiy sonini xotirada saqlaydigan ishlaydigan to'plam modelini amalga oshirishi mumkin. Ushbu ishchi to'plamni tashkil etadigan qaysi sahifalar (va sahifalar soni) OTga bog'liq, ammo odatda bu so'nggi paytlarda kirilgan sahifalardir. Jarayonni tayyorlamoqchi bo'lgan yadro, shuningdek, jarayon sahifalari xotira bilan almashtirilishidan oldin ushbu jarayon uchun belgilangan ishchi to'plamga ega bo'lishi kerak.

### **Virtual xotira**

Virtual xotira, odatda, talabni segmentatsiyalash (oldingi bo'limda muhokama qilinganidek, jarayonlarni ichki qismdan qismlarga ajratish) va / yoki peyj so'rovi (foydalanuvchining mantiqiy xotirasini umuman) xotira qismlarini ajratish usullari orqali amalga oshiriladi. Ushbu "talab" texnologiyalaridan foydalangan holda virtual xotira amalga oshirilsa, bu faqat foydalanilayotgan sahifalar va / yoki segmentlar RAMga yuklanganligini anglatadi. Virtual xotira tizimida OT mantiqiy manzillar asosida virtual manzillarni yaratadi va virtual manzillarni tarjima qilishda mantiqiy manzillar to'plamlari jadvallarini saqlab turadi. OT (apparat bilan bir qatorda) oxir-oqibatda har bir jarayon (fizik, mantiqiy va virtual) uchun bir nechta turli xil manzil maydonlarini boshqarishi mumkin. Dasturiy ta'minot RAM tomonidan bitta zich joylashgan xotira maydoni shaklida boshqariladi, ayni paytda yadroni xotirani bir necha qismlarga ajratish va paginatsiya qilish, segmentlarga ajratish va nafaol bo'lish, ajratish va tushirish yoki ajratib olinmagan va ekranlashsiz boshqarish mumkin.

## **Yadro xotirasi hajmi**

Yadro xotirasi bu yadro kodi joylashgan xotira qismidir, ularning ba'zilari yuqori darajadagi dasturiy ta'minot jarayoni orqali tizim qo'ng'iroqlari orqali olinadi va CPU ushbu kodni bajaradi. Yadro xotirasida joylashgan kod kerakli IPC mexanizmlarini o'z ichiga oladi, masalan, xabarlarini uzatish navbati uchun. Boshqa bir misol, vazifalar fork/exec yoki spawn tizimi qo'ng'iroqlarining ba'zi turlarini yaratganda. Vazifalarni yaratish tizimini chaqirgandan so'ng, OT nazoratni oladi va ba'zi bir operatsion tizimlarda yadro xotirasi bo'shlig'ida ushbu boshqaruv uchun ma'lumot va protsessor kontekstiga oid ma'lumotlarni o'z ichiga olgan vazifalarni boshqarish birligi (TCB) ni yaratadi. Oxir oqibat, yadro xotirasi maydonida boshqariladigan narsa, foydalanuvchi maydonidan farqli o'laroq, apparat, shuningdek, OT yadrosida amalga oshiriladigan haqiqiy algoritmlar bilan belgilanadi.

Yuqorida ta'kidlab o'tilganidek, foydalanuvchi rejimida dasturiy ta'minot tizim yadrosi rejimida ishlaydigan har kimga faqat tizim qo'ng'iroqlari orqali kira oladi. **Tizim chaqiruvlari** bu yadro protseduralari uchun yuqori darajadagi interfeyslar (foydalanuvchi rejimi) (yadro rejimida ishlaydi). OT va foydalanuvchi rejimida ishga tushirilgan tizim o'rtasida uzatilishi kerak bo'lgan tizim qo'ng'iroqlari bilan bog'liq parametrlar registrlar, ustunlar yoki asosiy xotiraning to'planishi orqali o'tadi. Tizimli qo'ng'iroqlar turlari odatda OT tomonidan qo'llab-quvvatlanadigan funktsiyalar turiga kiradi, shuning uchun ular fayl tizimlarini boshqarish (masalan, fayllarni ochish/o'zgartirish), jarayonlarni boshqarish (ya'ni jarayonlarni boshlash/to'xtatish), kirish/chiqish ma'lumotlarini almashish va yaqinda o'z ichiga oladi. Muxtasar qilib aytganda, yadro rejimidagi operatsion tizim foydalanuvchi rejimida qanday jarayon ishlayotganiga qarasa, dastur foydalanuvchi rejimida ishlaydi va OT ni tizim qo'ng'iroqlari orqali aniqlaydi.

## **6.5. Kirish/chiqish va fayl tizimini boshqarish**

Ba'zi o'rnatilgan operatsion tizimlar Flash, RAM yoki qattiq disk kabi turli xil xotira qurilmalarida fayl tizimini vaqtincha yoki doimiy saqlash uchun xotira boshqaruvini qo'llab-quvvatlaydi. Fayl tizimlari, aslida, boshqarish protokollari bilan birga fayllar to'plamidir (6.1-jadval). Fayl tizimining algoritmlari - bu saqlash moslamasiga o'rnatish

joyiga (joylashishiga) oʻrnatilgan (oʻrnatilgan) oʻrta dastur va / yoki dasturiy taʼminot.

Fayl tizimlari uchun yadro odatda fayl tizimini boshqarish bilan taʼminlaydi, kamida:

- Fayllarni ikkinchi darajali saqlashga, Flash yoki RAMga (masalan).

- Fayllar va kataloglarni boshqarish uchun ibtidoiy vositalarni qoʻllab-quvvatlash.

- Fayl taʼriflari va atributlari: nom protokoli, turlari (masalan, bajariladigan fayl, manba, manba, multimedia va boshqalar), oʻlchamlar, kirishni himoya qilish (oʻqish, yozish, bajarish, qoʻshish, oʻchirish va hk), oʻng mulk.

- Fayl operatsiyalari: yaratish, yoʻq qilish, oʻqish, yozish, ochish, yopish va hk.

- Faylga kirish usullari: ketma-ket, toʻgʻridan-toʻgʻri va boshqalar.

- Kataloglarga kirish, yaratish va oʻchirish.

OT fayllarni boshqarish uchun ishlatiladigan ibtidoiy (masalan, nomlash, maʼlumotlar tuzilmalari, fayl turlari, atributlar, operatsiyalar va boshqalar) jihatidan farq qiladi, qaysi xotira asboblari fayllari xaritaga olinishi va qaysi fayl tizimlari qoʻllab-quvvatlanishi mumkin. Aksariyat operatsion tizimlar fayl tizimi va xotira qurilmasi drayverlari oʻrtasida standart kirish/chiqishinterfeysidan foydalanadilar. Bu bir yoki bir nechta fayl tizimlarini operatsion tizim bilan birgalikda ishlatishga imkon beradi.

Oʻrnatilgan operatsion tizimlarda kirish / chiqish nazorati apparat va qurilma drayverlaridan uzoqda boʻlgan qoʻshimcha mavhumlikni taʼminlaydi (yuqori darajadagi dastur uchun). OC mavjud yadro tizim qoʻngʻirolari orqali koʻplab funktsiyalarni bajaradigan, kirish / chiqish qurilmalarini himoya qiladigan kirish/chiqishqurilmalari uchun yagona interfeysni taʼminlaydi, chunki foydalanuvchi jarayonlari faqat ushbu tizim qoʻngʻirolari orqali kirish/chiqishga kirish va halol va samarali kirish almashish sxemasini boshqarishi mumkin. - Bir nechta jarayonlar oʻrtasidagi natija. Bundan tashqari, OT protseduralarni sinxron va asinxron aloqani, asosan, voqealarga asoslangan holda boshqarish, ikkala tomonning (yuqori darajadagi jarayonlar va past darajadagi uskunalar) soʻrovlariga javob berish va maʼlumotlarni uzatishni boshqarish kerak. Ushbu maqsadlarga erishish uchun OT kirish/chiqishboshqaruv davri odatda foydalanuvchi jarayonlari va

qurilma drayverlari, shuningdek ba'zi bir bufer keshlash mexanizmining umumiy drayveri-qurilma interfeysidan iborat.

Jadval 6.1 Fayl tizimining oraliq standartlari

Fayl tizimi	Xulosa
FAT32 (fayllarni taqsimlash jadvali)	Xotira mumkin bo'lgan eng kichik qismga bo'linadi (sektorlar deb ataladi). Bir guruh tarmoqlar klaster deb ataladi. OC har bir klaster uchun o'ziga xos raqamni tayinlaydi va qaysi fayllardan foydalanishini hisobga oladi. FAT32 32 bitli adreslarni, shuningdek oldingi FAT (FAT, FAT16 va boshq.) Dan kichikroq klasterlarni manzillarni qo'llab-quvvatlaydi,
NFS (tarmoq fayl tizimi)	RPC (Remote Protocol Call) va XDR (Advanced Data Presentation) asosida NFS ishlab chiqilgan bo'lib, tashqi qurilmalarga tizimni bo'limni mahalliy xotirada bo'lgani kabi o'rnatishga imkon beradi. Bu sizga fayllarni tez va muammosiz tarmoq orqali almashish imkonini beradi.
FFS (flesh-fayl tizimi)	Flash xotira uchun mo'ljallangan.
Dosf	Bloklash moslamalarini (disklarni) real vaqt rejimida ishlatish uchun mo'ljallangan va MS-DOC fayl tizimiga mos keladi.
RawFS	Asosan butun diskni bitta katta fayl sifatida ko'rib chiqadigan oddiy xom fayl tizimini ta'minlaydi.
Tapefs	Lentadagi standart fayl yoki katalog tuzilishini ishlatmaydigan lenta qurilmalari uchun mo'ljallangan. Lenta hajmiga, asosan, katta hajmli fayl bo'lgan xom ashyo sifatida qaraladi
Cdromfl ar	Ilovalarga standart ISO 9660 fayl tizimiga muvofiq formatlangan kompakt-disklardan ma'lumotlarni o'qish uchun ruxsat beradi.

Qurilma drayveri kodi taxtaning kirish/chiqish apparatini boshqaradi. Kirish/chiqish -ni boshqarish uchun OT barcha qurilma drayver kodi ma'lum funktsiyalarni, masalan, boshlash, o'chirish, yoqish, o'chirish va hokazolarni talab qilishi mumkin. Keyin yadro

kirish/chiqishqurilmalarini, shuningdek ba'zi OT fayl tizimlarida nazorat qiladi. Yuqori darajadagi jarayonlar tufayli odatiy API-larning ma'lum to'plamiga kiradigan "qora qutilar". OT yuqori darajadagi qatlamlarni ta'minlash uchun qaysi turdagi kirish / chiqish interfeyslariga bog'liq ravishda katta farq qilishi mumkin. Masalan, Jbed yoki har qanday Java sxemasida barcha manbalar (shu jumladan, kirish/chiqish) skanerdan o'tkazilib, ob'ekt sifatida tuzilgan. O'z navbatida, VxWorks vxWorks kirish/chiqishquyi tizimidan foydalanish uchun quvurlar deb nomlangan aloqa mexanizmini ta'minlaydi. VxWorks-da, quvurlar bu kanal bilan bog'liq bo'lgan asosiy xabarlar qatorini o'z ichiga olgan virtual kirish/chiqishqurilmalaridir. Kanal orqali kirish/chiqish kirish baytlar oqimi (blokga kirish) yoki istalgan vaqtda bitta bayt (belgi kirish) sifatida ishlov beradi.

Ba'zi hollarda, kirish/chiqish apparati ma'lumot uzatilishini boshqarish uchun OT tamponlarini talab qilishi mumkin. Bir necha sabablarga ko'ra kirish/chiqish qurilmalarini boshqarish uchun tamponlar talab qilinishi mumkin. Asosan, ular blokka kirish orqali uzatiladigan ma'lumotlarni olishlari uchun zarurdir. OT buferda baytlar oqimini, qurilma bilan aloqa qilishni boshlagan yoki qilmaganligidan qat'i nazar, kirish/chiqishqurilmasiga uzatadi. Ishlash muammoga duch kelganda, tamponlar odatda sekinroq xotirada emas, balki keshda saqlanadi.

## **6.6. OT ishlashi bo'yicha tavsiyalar**

OT ishlashiga eng ko'p ta'sir qiladigan va bitta OT ishlashini boshqasidan ajratib turadigan ikkita OT quyi tizimlari xotirani boshqarish sxemasi (xususan, amalga oshirilgan almashinuv modeli) va rejalashtiruvchidir. Bitta virtual xotirani almashtirish algoritmining ishlashini boshqasiga solishtirganda, ular yaratgan sahifalardagi xatolar soni bilan bir xil xotira aloqalarini, ya'ni bir xil jarayon uchun har bir jarayon uchun ajratilgan sahifalar sonini hisobga olgan holda taqqoslash mumkin. ikkala OT ham. Bitta algoritmi har xil xotira ma'lumotlari bilan ta'minlash va har bir jarayon konfiguratsiyasi uchun har xil miqdordagi sahifalar uchun xatolar sonini qayd etish orqali ishlash uchun yana bir sinovdan o'tish mumkin.

Rejalashtirish algoritmining maqsadi umumiy ishlashni maksimal darajada oshiradigan sxemada bajarilishi kerak bo'lgan jarayonlarni tanlash bo'lsa ham, OT dispatcherlarining vazifasi shundaki, bir qator

ko'rsatkichlar mavjud. Bundan tashqari, algoritmlar indikatorga, xuddi shu jarayonlar bilan ham, teskari ta'sir ko'rsatishi mumkin. Rejalashtirish algoritmlarini bajarish uchun asosiy ko'rsatkichlar:

• **Tarmoq kengligi**, bu har qanday vaqtda protsessor tomonidan bajariladigan jarayonlar soni. OTni rejalashtirish darajasida, kichik jarayonlar kamroq o'tkazish qobiliyatiga ega bo'lishdan oldin, katta hajmdagi jarayonlarni bajarishga imkon beradigan algoritmlar mavjud. SPN sxemasida (quyida eng qisqa jarayon), hatto bir xil tizimda o'tkazish qobiliyati hozirgi vaqtda ishlayotgan jarayonlar hajmiga qarab o'zgarishi mumkin.

• **Bajarish vaqti**, davom etayotgan jarayonni (boshidan oxirigacha) bajarish uchun o'rtacha vaqt. Bu erda ushbu jarayonning hajmi ushbu ko'rsatkichga ta'sir qiladi. Biroq, rejalashtirish darajasida, jarayonni doimiy ravishda tushirishga imkon beradigan algoritmlar bajarilish vaqtini sezilarli darajada oshirishi mumkin. Bunday holda, xuddi shu jarayonni hisobga olgan holda, taqsimlanmagan va profilaktik rejalashtirgichni taqqoslash ikki xil bajarilish vaqtiga olib keladi.

• **Kutish vaqti**, jarayonni kutish kerak bo'lgan umumiy vaqt. Shunga qaramay, bu rejalashtirish algoritmi sekinroq jarayonlardan oldin katta jarayonlarga ruxsat berishiga bog'liq. Katta miqdordagi jarayonlar (ba'zi sabablarga ko'ra) ishlayotganligini hisobga olsak, keyingi har qanday jarayonlar yuqori kechikishga ega bo'ladi. Ushbu ko'rsatkich birinchi navbatda qaysi jarayonni tanlashni qaysi mezonlarga bog'liq bo'lishiga bog'liq - bitta sxemadagi jarayon boshqa rejalashtirish sxemasida joylashtirilganidan pastroq yoki yuqoriroq kutish vaqtiga ega bo'lishi mumkin.

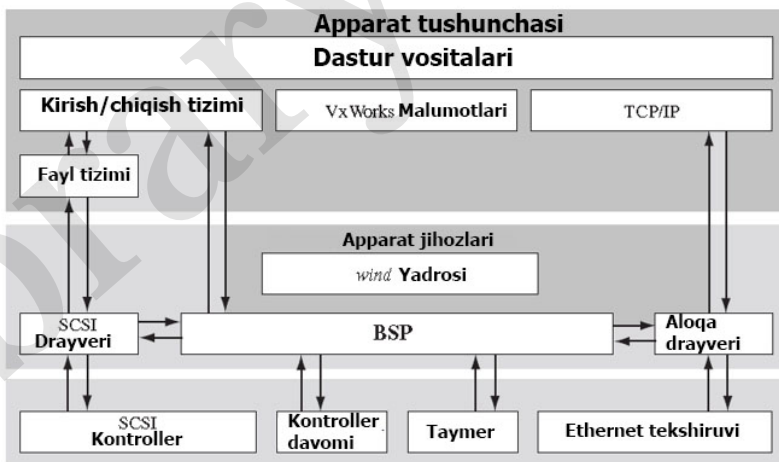
Xulosa qilib aytganda, rejalashtirish va xotirani boshqarish operatsion tizimning ishlashini yanada aniqroq tahlil qilish uchun ishlashga ta'sir etuvchi etakchi tarkibiy qismlardan biri bo'lsa-da, operatsion tizimda har ikkala turdagi algoritmlarning ta'sirini, shuningdek, OTga javob berish vaqti omilini (aslida, qaysi vaqtni) aniqlash kerak. foydalanuvchi jarayoni tizim so'rovni qayta ishlashni boshlaganda tizim qo'ng'irog'ini chaqiradi). Garchi OT qanchalik yaxshi ishlayotganligini birorta ham omil aniqlamasa ham, umuman OS ning ishlashi tizimdagi apparat resurslaridan (protsessor, xotira va kirish / chiqish qurilmalari) turli xil jarayonlar uchun qanday foydalanilayotganligi bilan aniq baholanishi mumkin. To'g'ri jarayonlar hisobga olingan holda, resurs bo'sh vaqtni emas, balki kodni bajarishga ko'proq vaqt sarflashi samaraliroq OTni ko'rsatishi mumkin.



## 6.7. OT va platani qo‘llab-quvvatlash paketlari (BSP)

Platalani qo‘llab-quvvatlash paketi (BSP) - bu OT sotuvchisi tomonidan ta'minlanadigan ixtiyoriy tarkibiy qism bo‘lib, uning asosiy maqsadi shunchaki operatsion tizim va universal drayverlar o‘rtasida mavhumlik darajasini ta'minlashdir.

BSP operatsion tizimni yangi apparat muhitiga o‘tkazishni osonlashtiradi, chunki u apparatga bog‘liq va apparat mustaqil manbalar kodi tizimida integratsiya nuqtasi bo‘lib xizmat qiladi. BSP dasturiy ta'minotni sozlashi va kompilyatsiya paytida moslashuvchanlikni ta'minlaydigan yuqori darajadagi dasturlarni ta'minlaydi. Ushbu muolajalar tizimning boshqa dasturiy ta'minotidan alohida kompilyatsiya qilingan qurilma drayver kodiga ishora qilganligi sababli, BSP-lar generator vaqtida ishlaydigan vaqtda kodni ko‘chma imkoniyatini ta'minlaydi. Rasmda ko‘rsatilgandek. 6.19, BSP arxitekturaga asoslangan qurilma drayverlarini konfiguratsiyani boshqarish va umumiy qurilma drayverlariga kirish uchun OT (yoki undan yuqori dasturiy ta'minot darajalari) uchun API-ni ta'minlaydi. BSP, shuningdek, tizim drayverini (apparat) va tizimdagi OTni ishga tushirishni boshqarish uchun javobgardir.



Rasm 6.19. BSP o‘rnatilgan tizimlar modelida

BSP qurilmasining konfiguratsiyani boshqarish qismi arxitekturaga tegishli qurilmalarning funksiyalarini o‘z ichiga oladi, masalan, mavjud

manzilni cheklash, foydalanish imkoniyati va ishlov berish rejimini cheklash (vektor jadvalini buzish uchun ISR ulanishi, o'chirish/yoqish, boshqarish registrlari va boshqalar). Umumjahon qurilma drayverlarini yangi arxitekturaga asoslangan arxitekturaga, turli xil spetsifikatsiyalari, uzilish sxemasi va arxitekturaga xos bo'lgan boshqa funktsiyalarga o'tkazishda maksimal moslashuvchanlik.

### **Nazorat savollari:**

1. Operatsion tizim nima?
2. Operatsion tizim nima qiladi?
3. Operatsion tizim ko'milgan tizimlar modeliga mos kelishini ko'rsatadigan diagramma tuzing.
4. Yadro nima? Kamida ikkita yadro funktsiyasini nomlang va tavsiflang.
5. OT arxitekturasi.
6. Jarayon va ip o'rtasidagi farq nima?
7. Jarayon va vazifa o'rtasidagi farq nima?
8. Vazifalarni yaratishda eng keng tarqalgan shablonlar qaysilar? Har bir sxemadan foydalanadigan OTga bitta misol keltiring.
9. Umuman olganda, vazifa qaysi davlatlarda turishi mumkin? OT va uning mavjud holatlariga, shu jumladan davlat diagrammalariga bitta misol keltiring.
10. Proaktiv va boshqarilmaydigan jadval o'rtasidagi farq nima? Proaktiv va boshqarilmagan rejalashtirishni amalga oshiradigan operatsion tizimlarga misollar keltiring.
11. Real vaqt rejimidagi operatsion tizim (RTOS) nima? RTOS-ga ikkita misol keltiring.
12. OT interfeysining eng keng tarqalgan aloqa va sinxronizatsiya mexanizmlarini nomlang va tavsiflang.
13. Musobaqa shartlari nima? Musobaqa shartlarini hal qilish usullari qanday?
14. Odatda uzilishlar uchun ishlatiladigan operatsion tizimlarning o'zaro ta'siri mexanizmi.
15. Yadro rejimida va foydalanuvchi rejimida ishlaydigan jarayonlar o'rtasidagi farq nima? Har bir rejimda ishlaydigan kod turiga misol keltiring.
16. Segmentatsiya nima? Segmentlarning manzillari qanday? Segmentda qanday ma'lumotlarni topish mumkin?
17. Stek, FIFO navbatida tuzilgan xotira segmentimi?

18. Peyjing nima? Xotirada va tashqarida sahifalarni almashtirish uchun bajarilishi mumkin bo'lgan to'rtta OT algoritmlarini nomlang va tavsiflang.

19. Virtual xotira nima? Nima uchun virtual xotiradan foydalanish kerak?

20. Nega ba'zi OTlarda POCIX standarti joriy qilingan? POCIX tomonidan belgilangan to'rtta OT API-larini sanab bering va aniqlang. Haqiqiy ichki o'rnatilgan uchta POCIX misollar keltiring.

21. OT ishlashiga qaysi ikkita quyi tizim eng ko'p ta'sir qiladi?

22. Har bir ta'sir ko'rsatkichidagi farqlar qanday farq qiladi?

23. BSP nima?

24. BSP ichida qanday elementlar joylashgan?

25. BSP-ni o'z ichiga olgan haqiqiy ichki o'rnatilgan OSlarga ikkita misol keltiring.

## 7-BOB. ORALIQ VA AMALIY DASTURIY TA'MINOTI

Oraliq dastur va amaliy dasturiy ta'minot o'rtasidagi chegaralar xiralashgan, shuning uchun ikkalasi ham ushbu bobda keltirilgan. Oraliq dastur - bu turli sabablarga ko'ra dastur sathidan olib tashlangan dastur. Buning sababi shundaki, u allaqachon tugagan OS to'plamiga kiritilishi mumkin. Uni ilova sathidan olib tashlashning boshqa sabablari quyidagilardan iborat: uni boshqa dasturlar bilan qayta ishlatish, uchinchi tomon sotuvchisi orqali sotib olish yoki dastur kodini soddalashtirish orqali xarajatlarni qisqartirish yoki vaqtni kamaytirish. Ushbu bobning qolgan bo'limlari o'rta dastur va amaliy dasturiy ta'minot nima ekanligini yoki yo'qligini aniqlaydi va o'rta dastur va amaliy dasturiy ta'minot uchun haqiqiy dunyoda soxta kodning namunalarini taqdim etadi.

### 7.1. Oraliq dasturiy ta'minot

Eng umumiy ma'noda o'rta dastur - bu OT, qurilma drayverlari yoki amaliy dasturlarning yadrosi bo'lmagan har qanday tizim dasturidir (7.1-rasm). E'tibor bering, ba'zi bir operatsion tizimlar o'rta dasturlarni OT ishlaydigan tizimga qo'shishi mumkin. Qisqacha aytganda, o'rnatilgan tizimda o'rta dasturiy ta'minot odatda qurilma drayverlarida yoki OT-ning tepasida joylashgan va ba'zida OT-ning ichiga qo'shilishi mumkin bo'lgan tizim dasturidir.



Rasm 7.1. O'rnatilgan tizim modelidagi oraliq dasturiy ta'minot

**Oraliq dasturiy ta'minot** odatda dasturiy ta'minot va yadro yoki qurilma drayveri o'rtasida vositachilikni ta'minlovchi dasturdir. Oraliq dasturiy ta'minot turli xil amaliy dasturlarni taqdim etadigan va

ularga xizmat ko'rsatadigan dasturdir. Xususan, oraliq dasturiy ta'minot - bu moslashuvchanlikni, xavfsizlikni, harakatchanlikni, moslashuvchanlikni, moslashuvchanlikni va / yoki ilovalar orasidagi o'zaro ta'sir mexanizmlarini ta'minlash uchun ikkita yoki undan ko'p dasturlarga ega bo'lgan o'rnatilgan qurilmalarda keng tarqalgan abstraksiya qatlami. Oraliq dasturlardan foydalanishning asosiy afzalliklaridan biri shundaki, u dasturiy ta'minot infratuzilmasini markazlashtirish orqali dasturlarning murakkabligini kamaytirish imkonini beradi, bu odatda an'anaviy ravishda dastur darajasida qayta topiladi. Biroq, tizimga oraliq dasturlarni kiritish qo'shimcha xarajatlarni talab qiladi, bu esa miqyosi va ish samaradorligiga sezilarli ta'sir ko'rsatishi mumkin. dasturiy ta'minot barcha darajadagi o'rnatilgan tizimga ta'sir qiladi.

Oraliq dasturiy ta'minotning turli xil turlari mavjud, ular orasida xabarga yo'naltirilgan oraliq dastur (MOM), ob'ektni so'rash vositachilari (ORB), masofaviy protsedura qo'ng'iroqlari (RPC), ma'lumotlar bazasi / ma'lumotlar bazasiga kirish va qurilma drayveri darajasidan yuqori tarmoq protokollari mavjud. dastur qatlamlari ostida OSI modeli joylashgan. Biroq, oraliq dasturlarning ko'p turlari odatda ikkita umumiy toifadan biriga tushadi:

- umumiy maqsad, ya'ni ular odatda turli xil qurilmalarda amalga oshiriladi, masalan, qurilma drayveri darajasidan yuqori bo'lgan tarmoq protokollari va OSI model dastur darajasidan past, fayl tizimlari yoki JVM kabi ba'zi bir virtual mashinalar;

- bozorga xos, ya'ni ular OT yoki JVM-da joylashgan standart raqamli televizion dasturlar kabi o'rnatilgan tizimlarning ma'lum bir oilasiga xosdir.

Oraliq qatlamning odatiy yoki bozorga xos elementi sifatida uni keyinchalik mulkiy deb tasniflash mumkin, ya'ni uni boshqalarga foydalanish uchun yoki ochiq ravishda litsenziyalashgan kompaniya tomonidan qo'llab-quvvatlanadigan xususiy dasturiy ta'minot bo'lib, u sanoat qo'mitasi tomonidan standartlashtirilgan va amalga oshirilishi mumkin. va / yoki har qanday manfaatdor tomon tomonidan litsenziyalangan.

Murakkab o'rnatilgan tizimlarda odatda bir nechta oraliq dastur elementlari mavjud, chunki ushbu amaliy talablarning barchasini qo'llab-quvvatlaydigan bitta texnologiyani topish odatiy emas. Bunday holda, oraliq dasturlarning individual elementlari odatda keyinchalik integratsiya muammolarining oldini olish uchun bir-biri bilan o'zaro

ta'siriga qarab tanlanadi. Ko'pgina o'rnatilgan OT ishlab chiquvchilari, shuningdek, tegishli OT va apparat platformalari bilan birga ishlaydigan o'rnatilgan dasturiy ta'minot paketlarini taqdim etadilar.

## 7.2. Ilova (amaliy) dasturiy ta'minot

O'rnatilgan tizimlardagi dasturlarning oxirgi turi-amaliy dasturiy ta'minotdir. 7.2-rasmda ko'rsatilgandek, dasturiy ta'minot tizim dasturiy ta'minoti darajasining yuqori qismida joylashgan va tizim dasturiy ta'minotiga bog'liq, boshqariladi va ishga tushiriladi. Bu dastur darajasida o'rnatilgan dastur bo'lib, u o'rnatilgan tizimning qurilmasini turini aniqlaydi, chunki dasturning funktsional imkoniyatlari ushbu o'rnatilgan tizimning eng yuqori darajasidadir va agar mavjud bo'lsa, ushbu qurilma foydalanuvchilari yoki ma'murlari bilan ko'p o'zaro aloqalarni amalga oshiradi. O'rnatilgan standartlar singari, o'rnatilgan ilovalar ham bozor xususiyatlariga qarab taqsimlanishi mumkin (faqat ma'lum bir turdagi qurilmalarda, masalan, interaktiv raqamli televidenie orqali talab qilingan video-ilovalar kabi) yoki umumiy maqsadga qarab (har xil qurilmalar turlarida amalga oshirilishi mumkin).



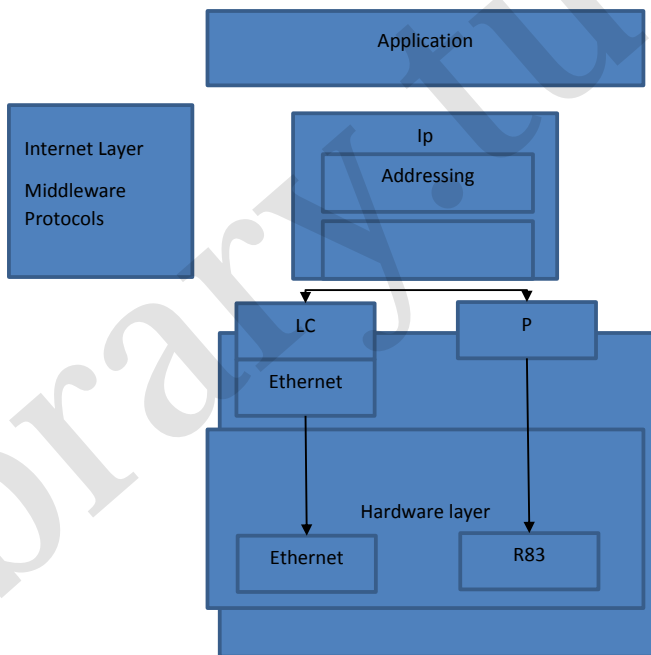
Rasm 7.2. O'rnatilgan tizimlarni ilova sathi modeli

## 7.3. Oraliq dasturiy ta'minot misollari

Internet qatlamining oraliq qatlamiga misol: Internet protokoli ( IP ). Internet protokoli yoki IP deb nomlangan tarmoq sathining protokoli DARPA RFC 791 standartiga asoslangan va asosan adreslash va bo'laklash funktsiyalarini amalga oshirish uchun javobgardir (7.3-rasm).

IP qatlam ma'lumotni yuqori qavat va pastki qavatdagi freymlardan paketlar ko'rinishida olganda, IP qatlam aslida ma'lumotlarni datagrammlar ko'rinishida ko'rib chiqadi va ishlov beradi, ularning formati 7.4. rasmda berilgan.

Butun IP datagramma bu quyi qatlamlardan olingan IP-ma'lumotlardir. Ma'lumotlar jadvalidagi oxirgi maydon, ma'lumotlar maydoni, IP orqali qayta ishlanganidan keyin yuqori qatlamlarga yuboriladigan paket. Qolgan maydonlar IP-ga ishlov berish tugagandan so'ng ma'lumotlar maydonidagi ma'lumotlarning yo'nalishiga qarab bo'linadi yoki qo'shiladi. Ushbu maydonlar IP manzilin yuborish va archalanish funksiyalarini qo'llab-quvvatlaydi. IP manzili va manzili maydonlari IP-qatlam tomonidan ishlov berilgan Internet yoki IP-manzil deb ataladigan tarmoq manzillari.



Rasm 7.3 IP tarmoq sathining protokoli

Aslida, bu erda IP-manzilning asosiy maqsadlaridan biri, manzilga murojaat qilish amalga oshiriladi. IP-manzillar "o'nlik aniqlikdagi o'nlik tizimda" 32 bitdan iborat bo'lib, "nuqta" bilan to'rt oltetaga bo'linadi

(jami 32 bit uchun 0-255 oralig'idagi to'rtta 8 bitli o'nlik raqamlar). IP manzili va manzili manzili maydonlari IP-qatlam

Version	IHL	TOS	Total	Length	
ID			Flags	Fragment	Offset
TTL		Protocol	Header	Checksum	
			Source Address		
			Destination Address		
			Option		Padding
			Data		

#### Rasm 7.4. IP- diagramma formati

tomonidan ishlov berilgan Internet yoki IP-manzil deb ataladigan tarmoq manzillari. Aslida, bu erda IP-manzilning asosiy maqsadlaridan biri, manzilga murojaat qilish amalga oshiriladi. IP-manzillar "o'nlik aniqlikdagi o'nlik tizimda" 32 bitdan iborat bo'lib, "nuqta" bilan to'rt oktetaga bo'linadi (jami 32 bit uchun 0-255 oralig'idagi to'rtta 8 bitli o'nlik raqamlar).

IP-manzili sinflar deb nomlangan guruhlariga bo'linadi, bu segmentlar hamma uchun WorldWide Internet yoki Internet kabi kattaroq tarmoqning soyabonida chalkashmasdan aloqa qilishlariga imkon beradi. RFC791-da ko'rsatilgandek, ushbu sinflar 7.1-jadvalda ko'rsatilgandek IP-adreslar oralig'ida joylashgan.

7.1-jadval: IP- manzillar oralig'i

Class	IP Address Range	
<b>A</b>	0.0.0.0	127.255.255.255
<b>B</b>	128.0.0.0	191.255.255.255
<b>C</b>	192.0.0.0	223.255.255.255
<b>D</b>	224.0.0.0	293.255.255.255
<b>E</b>	244.0.0.0	255.255.255.255

Sinflar (A, B, C, D va E) IP manzilidagi birinchi oktetning qiymatiga qarab ajratiladi. 7.5-rasmda ko'rsatilgandek, agar oktetdagi yuqori buyurtma biti "0" bo'lsa, unda IP-manzil sinfning "A" manzili. Agar eng yuqori buyurtma biti "1" bo'lsa, keyingi bit "0" uchun tekshiriladi - agar bo'lsa, u "B" sinf manzili va boshqalar.



A, B va C sinflarida, sinf bitlari yoki bitlar to'plamidan so'ng, bu tarmoq identifikatori. Tarmoq identifikatori Internetga ulangan har bir segment yoki qurilma uchun noyobdir va Internet Ma'lumot Markazi (InterNIC) tomonidan tayinlanadi. Keyin IP-manzilning xost identifikatori qismi qurilma yoki segment ma'murlarida qoladi. D sinfidagi manzillar tarmoq guruhleri yoki mezbon guruhlar deb nomlangan qurilmalarga tayinlanadi va ularga InterNIC yoki Internet Assigned Numbers Authority (IANA) tayinlanishi mumkin. 7.5 rasmda ta'kidlanganidek, E sinfidagi manzillar kelajakda foydalanish uchun ajratilgan.

Class A	0	Net ID (7 bits)		Host id (24 bits)	
Class B	1	0	Net ID (14 BITS)	Host id (16 bits)	
Class C	1	1	0	Net id (21 bits)	Host id (8 bits)
Class D	1	1	1	0	Multicast group ID (28 bits)
Class E	1	1	1	1	Reserved (28 bits)

Rasm 7.5. IP-manzil sinflari

### IP parchalanish mexanizmi

IP ma'lumotlar sxemasini qismlarga ajratish har qanday vaqtda faqat oz miqdordagi tarmoq ma'lumotlarini qayta ishlay oladigan qurilmalar uchun amalga oshiriladi. Datagrammalarni qismlarga ajratish va qayta yig'ish uchun IP protsedurasi tarmoq uzatilishida oldindan aytib bo'lmaydiganlikni qo'llab-quvvatlaydigan loyihadir. Bu shuni anglatadiki, IP tasodifiy ravishda qayta yig'ish uchun kelgan ma'lumotlarning tarkibiy qismlarini o'z ichiga olgan o'zgaruvchan miqdordagi ma'lumotlar jadvallarini qo'llab-quvvatlaydi va bu ularning parchalanmagan tartibida bo'lishi shart emas. Siz hatto turli xil datagrammalarning bo'laklarini qayta ishlashingiz mumkin. Parchalanish holatida, ma'lumotlar sarlavhasi deb nomlangan dastlabki

20 baytdagi maydonlarning ko'p qismi qismlarga ajratish va qayta yig'ish jarayonida qo'llaniladi.

Versiya maydoni uzatilgan IP versiyasini (ya'ni, IPv4 - versiya 4) ko'rsatadi. IHL (Internet Header Length) maydoni - bu IP ma'lumotlar sxemasi sarlavhasining uzunligi. Umumiy uzunlik maydoni bu sarlavha, parametrlar, ajratmalar va ma'lumotlarni o'z ichiga olgan butun ma'lumotlar datagrammalarning oktetsidagi haqiqiy uzunlikni aniqlaydigan 16 bitli maydon. To'liq uzunlikdagi maydon hajmining ta'siri shundan iboratki, ma'lumotlar jadvali 65,536 (216) sakkiztagacha bo'lishi mumkin.

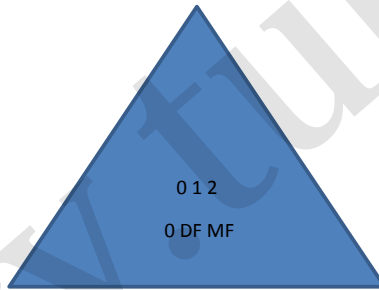
Ma'lumotlar jadvali bo'laklanganda, manba moslamasi "N" yo'llarini ajratadi va asl ma'lumot sathining sarlavhalarini mazmunini kichikroq sarlavhalarga ko'chiradi. «Internet-identifikatsiyasi» (ID) maydoni, qaysi parchalarni ma'lumotlar jadvaliga tegishli ekanligini aniqlash uchun ishlatiladi. IP protokoli bo'yicha kattaroq ma'lumotlar sxemasining ma'lumotlari bo'laklarga bo'linishi kerak, ulardan bittasi, qolgan sakkiz oktet blokdan (64 bit) iborat butun sonlardan iborat bo'lishi kerak.

Parchalarni ofset maydoni - bu 13 bitli maydon bo'lib, bu butun datagrammada fragment aslida qaerga tegishli ekanligini ko'rsatadi. Ma'lumotlar 8192 (213) gacha bo'lgan 8 oktetadan iborat bo'laklarga bo'linadi (har biri 64 bit), bu umumiy uzunlik maydoni 65.536 oktetni 8 oktet guruhlar uchun 8 ga bo'lingan = 8192. Birinchi bo'lak uchun fragment ofset maydoni bo'ladi. "0", lekin ayni o'sha datagrammaning boshqa bo'laklari uchun bu sxemaning ushbu qismining umumiy uzunligi (maydoni) va 8 oktetli bloklar soniga teng bo'ladi.

Bayroqcha maydonchalari (7.6-rasm), datagrammaning kattaroq qismning bo'lagi yoki yo'qligini bildiradi. Bayroq maydonining MF (Yana qismlar) bayrog'i, parcha ma'lumotlar jadvalining oxirgi (oxirgi elementi) ekanligini ko'rsatish uchun o'rnatiladi. Albatta, ba'zi tizimlarda parcha-parcha ma'lumotlarni yig'ish imkoniyati yo'q. Bayroq maydonidagi DF bayrog'i (parcha emas), bu qurilmaning parcha-parcha olingan datagrammalarni yig'ish uchun resurslarga egaligini ko'rsatadi. U bitta qurilmaning IP qatlami tomonidan boshqasiga unga berilgan ma'lumotlarning qismlarini qayta yig'ish imkoniyati yo'qligi to'g'risida xabar berish uchun ishlatiladi. Qayta yig'ish shunchaki identifikator, manba manzili, manzil manzili va protokol maydonlariga ega bo'lgan ma'lumotlar jadvallarini yig'ishni va parchalarni ofset maydonini va MF

bayroqlaridan foydalanib, bu ma'lumotlarning qaysi qismga tegishli ekanligini aniqlashni o'z ichiga oladi.

Version	HL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
	Protocol			Header Checksum
Padding				



Rasm.7.6. Bayroqlar

IP-datagrammada qolgan maydonlar quyidagicha umumlashtiriladi:

- Umr vakti (bu datagrammani umr muddatini bildiradi).
- Tekshirish summasi (ma'lumotlarning butunligini tekshirish).
- Variantlar maydoni (ba'zi hollarda zarur yoki foydali bo'lgan, lekin eng keng tarqalgan xabarlar uchun kerak bo'lmagan boshqarish funksiyalarini taqdim etadi).
  - Xizmat turi (kerakli xizmatning sifatini ko'rsatish uchun ishlatiladi. Xizmat turi - bu Internetni tashkil etuvchi tarmoqlarda taqdim etiladigan xizmatlarni tanlashni tavsiflovchi mavhum yoki umumlashtirilgan parametrlar to'plami.).
    - Padding (Internet sarlavhasini to'ldirish Internet sarlavhasi 32 bitli chegarada tugashini ta'minlash uchun ishlatiladi).
    - Protocol (Internet ma'lumotlar sxemasining ma'lumotlar qismida ishlatiladigan keyingi darajadagi protokolni bildiradi. Turli protokollar uchun qiymatlar RFC790 tomonidan berilgan (7.2-jadval).

Quyida IP ma'lumotlar sxemasini qayta ishlash protseduralarini yuborish va qabul qilish uchun psevdokodga misollar keltirilgan. Quyi sath protokollari (masalan, PPP, Ethernet, SLIP va boshqalar) bu qavatni demontaj qilish uchun ma'lumotlar sxemasini olish uchun bildirish

7.2-jadval.  
Sakkizlik protokollar sonlari

0	0	Reserved
1	1	ICMP
2	2	Unassigned
3	3	Gateway-to-Gateway
4	4	CMCC Gateway Monitoring Message
5	5	ST
6	6	TCP
7	7	UCL
8	10	Unassigned
9	11	Secure
10	12	BBN RCC Monitoring
11	13	NVP
12	14	PUP
13	15	Pluribus
14	16	Telenet
15	17	XNET
16	20	Chaos
17	21	User Datagram
18	22	Multiplexing
19	23	DCN
20	24	TAC Monitoring
21-62	25-76	Unassigned
63	77	any local network
64	100	SATNET and Backroom EXPAK
65	101	MIT Subnet Support
66-68	102-104	Unassigned
69	105	SATNET Monitoring

71	107	Internet Packet Core Utility
72-75	110-113	Unassigned
76	114	Backroom SATNET Monitoring
77	115	Unassigned
78	116	WIDEBAND Monitoring
79	117	WIDEBAND EXPAK
80-254	120-376	Unassigned
255	377	Reserved

uchun «IPReceive» protsedurasini chaqiring, yuqoriroq sath protokollari (masalan, TCP yoki UDP kabi) " Ma'lumotlar jadvalini uzatish uchun IPSend "(masalan.

```

ipReceive (ma'lumotlar sxemasi,...) {
....
parseDatagram (Versiya, InternetHeaderLength, TotalLength,
Flags,...);
....
if (InternetHeaderLength " OR " TotalLength = OutOfBounds) OR
(FragmentOffset = yaroqsiz) YOKI
(Versiya = qo'llab-quvvatlanmaydi), keyin {
... to'g'ri ma'lumotlar diagrammasi sifatida qayta ishlanmaydi...;
} else {
VerifyDatagramChecksum (HeaderChecksum...);
agar {HeaderChecksum = Yaroqli} bo'lsa, unda
....
agar (IPDestination = bu qurilma) bo'lsa, {
....
agar (qurilma tomonidan qo'llab-quvvatlanadigan protokol), keyin {
protokolga nuqta / o'tish, ma'lumotlar to'plami kutmoqda...;
qaytish;
}
....
} else {
... ma'lumotlar grafigi ushbu ishlov berish moslamasi uchun emas...;
} // tugatish if-then-else Ipdestination...
} else {
... checksumi GEÇERSÝZ uchun ishlash datagramlari...;
} // end -then-else headerchecksum...

```

```

} // tugatish, agar headerchecksum haqiqiy bo'lsa
ICMP (ma'lumotlar jadvalini qayta ishlash xatosi); // ko'rsatish
uchun ishlatiladigan Internet-ni boshqarish protokoli
// datagram ushbu qurilma tomonidan muvaffaqiyatli ishlanmadi
} // if-then-else (InternetHeaderLength...) tugatish
}
ipSend ( paket,...) {
....
CreateDatagram ( paket, versiya, InternetHeaderLength,
TotalLength, bayroqlar,...)
sendDatagramToLowerLayer (ma'lumotlar jadvallari);
....
}

```

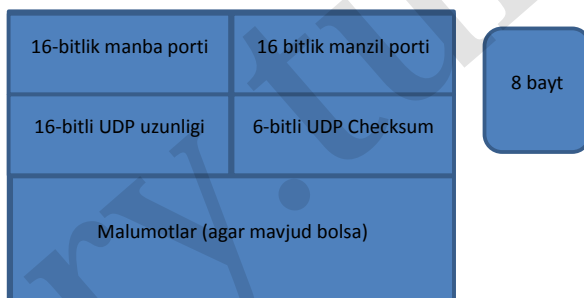
### **Transport sathi namunasi: foydalanuvchi datagramma protokoli (UDP)**

Ikki eng keng tarqalgan transport sathining protokollari Transmit Control Protocol (TCP) va User Datagram Protocol (UDP). Ikki protokol o'rtasidagi asosiy farqlardan biri bu ishonchlilikdir. TCP ishonchli deb hisoblanadi, chunki u paketlarini qabul qiluvchilardan tasdiqlashni talab qiladi. Agar ularni qabul qilmasa, TCP keyin tasdiqlanmagan ma'lumotlarni uzatadi. O'z navbatida, UDP, bu ishonchsiz transport qatlami protokoli, chunki u paketlarini qabul qiluvchisi haqiqatan ham ma'lumot oladimi yoki yo'qligini hech qachon bilmaydi. Qisqasi, bu misol RFC768-ga asoslangan oddiy, ishonchsiz, ma'lumotlar sxemasiga yo'naltirilgan UDP-ni qamrab oladi. UDP paketi 7.7-rasmda keltirilgan.

UDP (User Datagram Protocol) kabi transport sathining protokollari Internet qatlam protokollarining yuqori qismida joylashgan (IP kabi) va odatda ikkita maxsus qurilma o'rtasida aloqani o'rnatish va tarqatish uchun javobgardir. Ushbu turdagi aloqa "nuqta-nol" deb nomlanadi. Ushbu darajadagi protokollar qurilmada ishlaydigan bir nechta yuqori darajadagi dasturlarga nuqtalarni boshqa qurilmalarga ulash imkonini beradi. Ba'zi bir transport sathining protokollari ma'lumotlarning nuqtadan nuqtaga ishonchli uzatilishini ta'minlansa ham, UDP ulardan biri emas.

Garchi server tomonidagi aloqa mexanizmlari mijoz qurilmasining mexanizmlaridan farq qilishi mumkin bo'lsa-da, mijoz va server mexanizmlari transport sathining soketiga asoslangan. Transport

protokoli tomonidan ishlatilishi mumkin boʻlgan bir nechta rozetkalar mavjud, ular oqim, ma'lumotlar diagrammasi, xom va seriyali paketlar kabi bir nechtasini nomlash uchun ishlatiladi. UDP ma'lumotlar socketlarini ishlatadi, xabarga yoʻnaltirilgan socket bir vaqtning oʻzida bitta xabarni qayta ishlaydi (masalan, TCP tomonidan ishlatiladigan socket oqimi tomonidan qoʻllab-quvvatlanadigan simvollarning doimiy oqimidan farqli oʻlaroq). Bir-biridan-biriga ulanadigan aloqa kanalining har uchida rozetka mavjud va boshqa qurilma bilan aloqa oʻrnatmoqchi boʻlgan har bir dastur buni rozetkani oʻrnatish orqali amalga oshiradi. Socketlar ushbu qurilmadagi maxsus portlarga ulanadi, bu erda kirish raqami kirish dasturini aniqlaydi. Ushbu ikkita qurilma (mijoz va server) oʻzlarining rozetkalari orqali ma'lumotlarni yuboradilar va qabul qiladilar.



Rasm.7.7. UDP sxemasi

Umuman olganda, server dasturi server tomonida ishlaydi, socketni tinglaydi va ulanishni soʻrashni kutadi. Mijoz asosan oʻz porti orqali server bilan oʻzaro ishlaydi. Portlar 16 bitli butun sonlarni bildiradi, yaʼni har bir moslamada 65536 (0-65535) port mavjud. Baʼzi portlar maʼlum dasturlarga birlashtirilgan (yaʼni FTP = 20-21 portlar, HTTP = 80 port va boshqalar). UDP asosan qabul qiluvchining IP- manzili va uzatilayotgan paketdagi port raqamini oʻz ichiga oladi, maʼlumotlarning toʻgʻri tartibda yoki hatto umuman qabul qilinganligini tekshirish uchun qoʻl berib koʻrish mumkin emas. Server qabul qilingan paketning IP- manzili va port raqamini chiqarib olish orqali olingan maʼlumotlarning oʻz dasturlaridan biri uchun qabul qilinganligini aniqlaydi. Muvaffaqiyatli ulanish oʻrnatilgandan soʻng, mijoz ilovasi aloqa uchun

rozetkani o'rnatadi, so'ngra server boshqa mijozlarning kirish so'rovlarini tinglash uchun yangi rozetkani o'rnatadi.

Quyidagi psevdakod kiruvchi ma'lumotlar jadvalini qayta ishlash uchun soxta kodlangan UDP algoritmini ko'rsatadi. Ushbu misolda, agar qabul qilingan ma'lumotlar sxemasi uchun socket topilsa, ma'lumotlar sxemasi stakka (dastur darajasiga) yuboriladi, aks holda xato xabari qaytariladi va ma'lumotlar diagrammasi o'chiriladi.

```
demuxDatagram (ma'lumotlar diagrammasi ) {
...
verifyDatagramChecksum ( datagram. Cheksum );
if (datagram.Length <= 1480 && datagram.Length>= 8) {
...
if (datagram.Checksum VALID) bo'lsa, {
findSocket (ma'lumotlar jadvali, DestinationPort);
if (rozetka FOUND) {
sendDatagramToApp (manzilPort, datagram.Data); // ma'lumotlar
jadvalini dasturga yuborish
qaytish
} else {
Icmp.send( datagram, socketNotFound); // Internet darajasini
belgilang, qaysi
// ma'lumotlar mo'ljallangan dasturga etib bormaydi
qaytish }
}
}
discardInvalidDatagram ();
}
```

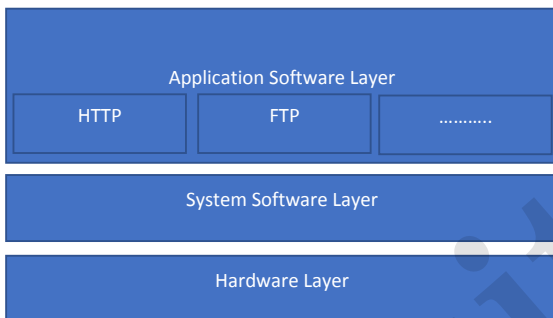
#### **7.4 Ilova darajasidagi dasturiy ta'minotga misollar**

Ba'zi hollarda, ilovalar, rasmda ko'rsatilganidek, umumiy qabul qilingan standartlarga asoslanishi mumkin 7.8. rasm.

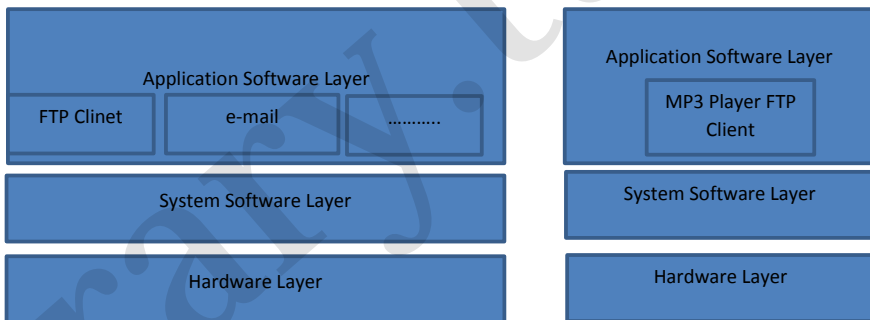
Masalan, ma'lumotlarni uzatish uchun boshqa qurilmalarga ulanishi yoki uzoqdan ishlaydigan qurilmalar buyruq bilan funktsiyalarni bajarishi uchun dasturning tarmoq protokoli ma'lum bir Rasmda dastur darajasida bajarilishi kerak. Ilova darajasidagi tarmoq protokollari ular amalga oshiriladigan dasturlardan mustaqil, ya'ni dastur protokoli mustaqil dasturda bajarilishi mumkinligini anglatadi, uning vazifasi faqat ushbu protokolni bajarishdir yoki u kengroq dasturning pastki



birliqi sifatida bajarilishi mumkin. 7.9-rasmda ko'rsatilgandek ko'p funktsiyalar.



Rasm 7.8. Ilova dasturlari va tarmoq protokollari



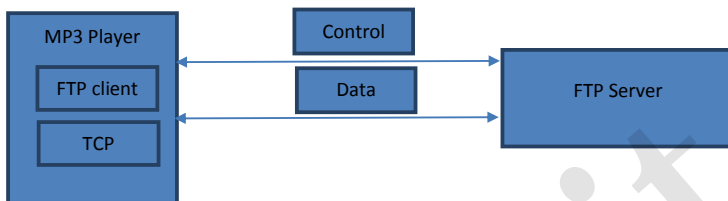
Rasm.7.9. Dastur va tarmoq protokoli dasturlari

Quyidagi misollar umumiy maqsadlarda qo'llaniladigan tarmoq protokollarini namoyish etadi.

### Fayl uzatish protokoli (FTP)

**FTP (fayl uzatish protokoli)** - faylni tarmoq orqali xavfsiz almashish uchun ishlatiladigan eng oddiy protokollardan biri. FTP RFC959-ga asoslangan va faqat tarmoq qurilmalari o'rtasida yoki brauzerlar va MP3 dasturlari kabi dasturlarda fayllarni uzatish uchun mo'ljallangan mustaqil dastur sifatida amalga oshirilishi mumkin.

7.10-rasmda ko'rsatilgandek, FTP protokoli FTP mijozni yoki foydalanuvchi protokoli tarjimoni (foydalanuvchi PI) deb nomlanuvchi va FTP serveri yoki FTP sayti deb nomlangan FTP ulanishini qabul qiladigan qurilma bilan uzatishni boshlaydigan qurilma o'rtasidagi aloqa mexanizmlarini belgilaydi.



7.10-rasm. FTP ulanishi

FTP mijozni va serverlari o'rtasida ikki xil ulanish bo'lishi mumkin: boshqarish ulanishi, bunda qurilmalar o'rtasida buyruqlar uzatiladi va fayllar uzatiladigan ma'lumotlar ulanishi. FTP seansi FTP mijozni maqsadli qurilmaning 21 porti bilan TCP ulanishini o'rnatish orqali boshqarish ulanishini boshlash bilan boshlanadi. FTP protokoli uning asosidagi transport protokoli TCP kabi ishonchli, soddalashtirilgan ma'lumotlar oqimi kanali bo'lishini talab qiladi (9.10-rasm). Eslatma. FTP ulanish mexanizmi qisman RFC854, Telnet protokoli (terminal emulyatsiyasi) ga asoslangan.

FTP mijozni o'z buyruqlarini uzatgandan so'ng, boshqarish ulanishi orqali FTP saytidan javob kodi bilan javob berilishini kutadi; ushbu kodlar RFC 959-da belgilangan va 7.3-jadvalda keltirilgan.

7.3-jadval. FTP Buyruq kodlari

Code	Definition
110	Restart marker reply
120	Service ready in "x" minutes
150	Data connection already open
200	File status ok
202	Command not implemented
211	System help
...	.....

Agar FTP saytidan javob ijobiy bo'lsa, FTP mijozni 7.4 jadvalda ko'rsatilgan buyruqlarni yuboradi, masalan foydalanuvchi nomi yoki

parol, uzatish mezonlari (masalan, ma'lumotlar porti, uzatish rejimi, kirish parametrlari) kabi kirishni boshqarish parametrlarini belgilaydigan buyruqlar. ko'rish turi va fayl tuzilishi va boshqalar), shuningdek tranzaksiya (saqlash, olish, qo'shish, o'chirish va hk).

Quyidagi soxta kod, FTP saytiga kirishni boshqarish buyruqlari yuboriladigan FTP mijoz dasturida mumkin bo'lgan boshlang'ich FTP ulanish mexanizmini namoyish etadi.

7.4-jadval. FTP buyruqlari

Code	Definition
USER	Username – access control command
PASS	Password – access control command
QUIT	Logout – access control command
PORT	Data port – transfer parameter command
TYPE	Representation Type – transfer parameter command
MODE	Transfer mode – transfer parameter command
DELET	Delete – FTP service command

### USER va PASS-ga kirishni boshqarish buyruqlari uchun psevdokod FTP mijoz

```

FTPConnect (simli tugun, satrga kirish, parol) {
    TCPSocket s = yangi TCPSocket (FTPServer, 21); // 21 port bilan
    TCP aloqasini o'rnatish
    // maqsadli qurilma
    Kutish vaqti = 3 soniya; // ulanishni o'rnatish uchun kutish vaqti 3
    soniya
    FTP muvaffaqiyatli = FALSE;
    Vaqt = 0;
    Xayr (vaqt < kutish vaqti) {
        JAVOB o'qing;
        Agar javob oluvchidan bo'lsa, {
            // FTP-ga kirish
            serverga uzatish ("USER" + login + "\r\n");
            serverga uzatish ("PASS" + parol + "\r\n");
            JAVOB o'qing;
            // 230 javob davom etish uchun foydalanuvchi tizimga kirganligini
            anglatadi
        }
    }
}

```

```
agar REPLY 230 bo'lmasa {
TCP ulanishini yoping
vaqt = kutish vaqti;
} else {
vaqt = kutish vaqti;
```

```
FTP muvaffaqiyatli = TRUE;
}
} else {
vaqt = vaqt + 1;
} // qabul qiluvchidan if-then-else javobini tugatish
} // tugash vaqti (vaqt < kutish vaqti)
}
```

Aslida, quyidagi soxta kodda ko'rsatilgandek, FTP mijosi FTP orqali kirish mumkin bo'lgan turli xil buyruqlarni uzatuvchi (7.4-jadval) foydalanuvchini qo'llab-quvvatlovchi mexanizmlarni taqdim qilishi kerak (7.3 jadval).

### **FTP mijozining psevdokodi**

```
// "QUIT" kirish buyruqlar qatori
FTPQuit () {
serverga uzatish ("QUIT");
JAVOB o'qing;
// javob 221 serverni yopish bilan bog'lanishni anglatadi
agar REPLY 221 bo'lmasa {
// serverga ulanishni yopish xatosi
}
TCP ulanishini yoping
}
// FTP-server buyruq operatsiyalari
// "O'CHIRISH"
FTPDelete (fayl nomi nomi) {
serverga uzatish ("DELE" + fayl nomi);
JAVOB o'qing;
// answer 250 Ok faylining talab qilingan harakatini bildiradi
agar REPLY 250 bo'lmasa
// faylni o'chirishda xato
}
```

```

}
// "RNFR"
FTPRenameFile ( string oldfilename, string newfilename) {
serverga uzatish ("RNFR" + oldfilename);
o'qing: REPLY; // javob 350 so'ralgan fayl harakatini anglatadi,
qo'shimcha ma'lumot kutilmaydi
agar REPLY 350 bo'lmasa {
// faylni qayta nomlashda xato
}
serverga o'tkazish (" RNTO " + yangifild nomi );
JAVOB o'qing;
// answer 250 Ok faylining talab qilingan harakatini bildiradi
agar REPLY bo'lmasa 250 {
// faylni qayta nomlashda xato
}
}
}

```

.....  
FTP serveri ma'lumot ulanishini va FTP mijozi tomonidan belgilangan buyruqlarga muvofiq har qanday uzatishni amalga oshiradi.

### **Dasturlash tillari va dasturiy ta'minot haqida sharh**

Yuqori darajadagi dasturlash tillarining ba'zi turlari amaliy darajadagi arxitekturaga ta'sir qilishi mumkin. Shunday tillardan biri Java; Java dasturi uning ichidagi JVM-ni birlashtirishi mumkin. Bunga misol sifatida Palm Pilot PDA yoki PocketPC platformasidagi.exe faylida.prc fayliga (Palm OS-da bajariladigan fayl) kompilyatsiya qilingan J2ME CLDC / MIDP JVM kiradi. Qisqasi, JVM apparat darajasida amalga oshirilishi mumkin, chunki oraliq dastur OTga yoki operatsion tizimning yuqori qismida joylashgan yoki dasturning bir qismi sifatida.

Dasturlash tillari amaliy darajadagi arxitekturaga ta'sir qilishi mumkin bo'lgan yana bir holat, agar dastur skriptlash tilida yozilgan dastlabki kodni qayta ishlashi kerak bo'lsa, masalan HTML (HyperText Markup Language, keng fayllar yoki sahifalar keng veb-sahifada joylashgan til) yoki Javascript (veb-sahifalarda interfaol funktsiyalarni amalga oshirish uchun foydalaniladi, shu jumladan kursorni urish paytida chivinlardagi tasvirlarni o'zgartirish, Rasmlarning o'zaro ta'siri, bajarilgan hisoblar va boshqalar). Keyin skript tilining tarjimoni amaliy dasturiy ta'minotga qo'shiladi (masalan, veb- brauzer, HTTP mijozining o'zi, bu erda bir nechta skript tillari tarjimonlari birlashtirilgan).

### **Nazorat savollari:**

- 1.Oraliq dastur nima?
- 2.Umumiy maqsadli oraliq dastur va sotuvga yo'naltirilgan oraliq dastur o'rtasidagi farq nima?
- 3.Tarmoq oraliq dasturlari OSI modelida qaerda joylashgan?
- 4.OSI modeliga nisbatan TCP / IP modelining qatlamlarini chizish.
- 5.TCP ostida qanday darajaga tushadi?
- 6.RS-232 bilan bog'liq dastur oraliq dasturmi?
- 7.PPP ma'lumotlarni kim boshqaradi?
- 8.PPP dasturini tashkil etuvchi to'rtta komponentni nomlang va tavsiflang. Qanday RFC har biri bilan bog'liq?
- 9.PPP holati va PPP voqeasi o'rtasidagi farq nima? Har birining uchta misolini sanab bering va tavsiflang.
- 10.IP-manzil nima? IP-manzillarni qaysi tarmoq protokoli boshqaradi?
- 11.UDP va TCP o'rtasidagi asosiy farq nima?
- 12.Oraliq dasturlarda bajarilishi mumkin bo'lgan uchta o'rnatilgan JVM standartlari nima? Ushbu standartlarning API o'rtasidagi farq nima?
- 13.NET ixcham ramkasi o'rnatilgan tizim modelining oraliq dasturi darajasida amalga oshirilganmi?
- 14.Ilova dasturlari nima?
- 15.Ichki tizim modelida odatda amaliy dasturiy ta'minot qayerda joylashgan?
- 16.Mustaqil dastur sifatida bajarilishi mumkin bo'lgan amaliy dastur protokollarining ikkita namunasi nima, ularning yagona vazifasi protokol hisoblanadi yoki katta funktsional dasturning pastki komponenti sifatida amalga oshiriladi.
- 17.FTP mijoz va FTP serveri oraliqsidagi farq nima? Har biri o'rnatilgan qurilmalarning qaysi turini amalga oshiradi?
- 18.SMTP protokol amalga oshiradi?
- 19.SMTP odatda ishlash uchun TCP oraliq dasturlariga tayanadi.
- 20.HTTP nima? HTTP mijoz yoki serverini qaysi ilovalar o'z ichiga oladi?
- 21.Ilovalar darajasida komponent qanday dasturlash tillarini kiritishi kerak?

## **8-BOB. O‘RNATILGAN TIZIMLARNI LOYIHALASH VOSITALARI**

### **8.1. Dasturlash tillari**

#### **8.1.1 Asosiy ta'riflar**

**Til** - bu birlashish qoidalari bilan birlashtirilgan va kommunikativ maqsadlarga xizmat qiladigan belgilar tizimidir. Til insonning fikrlari va tasvirlarini uzatish vositasidir. Rasmiy til ta'rifni birma-bir anglashga imkon beradi, norasmiy til esa noaniqlikka imkon beradi.

**Dasturlash tili** - bu ma'lum bir hisoblash modeli doirasida hisoblash moslamasi tomonidan aniq qabul qilinadigan, odamning fikrlari va tasvirlarini rasmiy shaklda ifodalashga imkon beradigan til.

Til tarkibiy qismlariga quyidagilar kiradi: leksika, sintaksis va semantika.

**Leksika** - til elementlarining tavsifi (uning elementar konstruktsiyalari). Masalan, C tokenlari: uchun, agar, int, char, while, +, -,...

**Sintaksis** - til elementlarini yozish uchun qoidalar to'plami (tokenlar).

**Semantika** - til tuzilmalarini semantik to'ldirish, til elementlari va ularning ma'nosi o'rtasidagi bog'liqlik.

Leksikalar va sintaksis juda yaqin bo'lgan va semantika bir-biridan farq qiladigan tillar mavjud. Odatda tillar turli xil hisoblash modellariga asoslangan bo'lsa, bu sodir bo'ladi. Dasturlash tillarining semantikasini aniqlashga bir nechta yondashuvlar mavjud. Quyidagi uchta eng keng tarqalgan navlar : operatsion, denotatsion (matematik) va lotin (aksiomatik). Operatsion yondashuv doirasida semantikani tavsiflashda, dasturlash tilidagi tuzilishlarning bajarilishi odatda ba'zi bir xayoliy (mavhum) kompyuter yordamida izohlanadi. Derivatsion semantika mantiqiy tildan foydalangan holda va pre- va postkonditsiyalarni o'rnatish orqali til konstruktsiyalarini bajarish oqibatlarini tavsiflaydi. Denotatsion semantika matematikaga xos bo'lgan tushunchalar - to'plam, yozishmalar, shuningdek hukmlar, bayonotlar va boshqalar bilan ishlaydi.

**Statik tiplash** - bu dasturlash tillarida keng qo'llaniladigan usul bo'lib, unda o'zgaruvchi, pastki dastur parametrlari, funktsiyaning qaytarilish qiymati deklaratsiya paytida ushbu tur bilan bog'lanadi va keyinchalik turni o'zgartirish mumkin emas (o'zgaruvchi yoki parametr

qabul qilinadi va funktsiya faqat shu turning qiymatlarini qaytaradi). Statik tiplangan tillarga misollar Ada, C ++, Paskal.

**Dynamic tiplash** - bu dasturlash tillarida va spetsifikatsiya tillarida keng qo'llaniladigan usul bo'lib, unda o'zgaruvchi qiymat tayinlanganda tip bilan bog'liq bo'ladi, ammo bu o'zgaruvchini e'lon qilganda emas. Shunday qilib, dasturning turli qismlarida bir xil o'zgaruvchi turli xil qiymatlarni qabul qilishi mumkin. Dinamik terish mavjud bo'lgan tillarga misollar - Smalltalk Python, Ruby, PHP, Perl, JavaScript, Object Pascal, Lisp, xBase.

Tillarni quyidagi mezonlarga ko'ra tasniflash mumkin:

- loyihaning hayotiy tsiklidagi o'rni (dasturlash tili, spetsifikatsiya tili);
  - rasmiyatchilik darajasi (rasmiy til, norasmiy til);
  - ishlatilgan hisoblash modeli;
  - tyuring to'liqligi;
- amalga oshirish usuli (tarjimon yoki tuzuvchi);
- yozish usuli (dinamik yoki statik);
  - ishlatiladigan semantikaning turi (operativ, denotativ lotin).

### 8.1.2. Tilni xususiyatlari va dasturlash tillari

O'rnatilgan tizimlarini dasturlash bu juda ko'p darajali jarayon bo'lib, unda til vositalari asosiy rol o'ynaydi. Foydalanish bosqichiga va xususiyatiga qarab, tillar odatda spetsifikatsiya tillari (dizayn bosqichi) va dasturlash tillari (amalga oshirish bosqichi) ga bo'linadi, ammo har qanday rasmiy til, asosan, ishlatilishi mumkin degan fikr mavjud. ushbu bosqichlarning har qandayida. Umumiy maqsadli samolyotlardan farqli o'laroq, ma'lum bir ob'ektning til tavsifini yaratish bosqichida, ushbu ob'ekt qaysi shaklda amalga oshirilishini aytish mumkin emas - dasturlashtiriladigan protsessor uchun an'anaviy dastur sifatida, FPGA konfiguratsiyasi sifatida, yoki ixtisoslashtirilgan apparat birligi sifatida.

O'rnatilgan tizimlarni loyihalashda tillardan foydalanishning ushbu xususiyatidan kelib chiqqan holda, har xil maqsadlarda va turli darajadagi taqdimotlarda o'rnatilgan tizimlarining vositalarini belgilashda ulardan foydalanish nuqtai nazaridan eng umumiy va qulay tilni ko'rib chiqish kerak. Aynan ana shu yondashuv taklif etilgan tillarni tasniflashga, bu o'rnatilgan tizimlarning ko'p tilli loyihasi kontseptsiyasidan kelib chiqadi. Quyida biz o'rnatilgan tizimlarni



yaratishda ishlatiladigan tillarni aniq ko'rib chiqamiz. Asosiy e'tibor tizim darajasida loyiha uchun ishlatiladigan vositalarga qaratiladi.

Tillar tizimni loyihalashdagi eng muhim bosqichlardan biri - tizimni aniqlashtirish bosqichida qo'llaniladi. Ko'p spetsifikatsiya tillari mavjud. Ularning har biri o'ziga xos afzalliklarga ega, ammo cheklangan dastur sohasida boshqalardan ustundir.

Tilni tanlash, odatda, tilning ifoda etuvchi kuchi, tilning asosiy modeli tomonidan taqdim etilgan avtomatlashtirish imkoniyatlari va tilni qo'llab-quvvatlovchi vositalar va usullarning mavjudligi kabi bir necha mezon o'rtasida murosaga keladi. Ba'zi hollarda bitta loyihaning turli xil modullarini ko'rsatish uchun bir nechta tillardan foydalanish foydalidir.

Turli xil qismlar turli xil dastur toifalariga tegishli bo'lgan heterojen tizimlarni loyihalash uchun ko'p tilli echimlar talab qilinadi, masalan boshqarish / ma'lumotlar yoki doimiy / diskret.

Tizimni loyihalashning barcha vositalari kirish malumotlari sifatida tillardan foydalaniladi. Ular odatda dastlabki spetsifikatsiyani qayta ishlash va o'zgartirishni amalga oshirish uchun oraliq shakldan foydalanadilar. Til vakili hisoblash modellariga asoslangan bo'lib, ularning mohiyati avvalgi bobda keltirilgan. Hisoblash modellari ma'lumotlarga yo'naltirilgan yoki boshqaruvga yo'naltirilgan bo'lishi mumkin. Ikkala holatda ham ular sinxron va asenkron bo'lishi mumkin.

Hisoblanadigan har bir funktsiya uchun uni hisoblash elementi (masalan, Turing mashinasi) yoki ijrochi uchun dastur mavjud va ko'plab hisoblagichlar tomonidan hisoblangan barcha funktsiyalar hisoblash funktsiyalari (ehtimol kirish va chiqish ma'lumotlarini ba'zi kodlash bilan ). Ism Alan Turingdan kelib chiqadi, u mavhum hisoblagichl - Turing mashinasi bilan tanishgan va Turing mashinalari yordamida hisoblab chiqilishi mumkin bo'lgan ko'p funktsiyalarni aniqlagan. Eng ko'p ishlatiladigan dasturlash tillari turing-complete. Bu Paskal kabi funktsional (Haskell) va mantiqiy dasturlash tillari (Prolog) kabi ikkala imperativ tillarga ham tegishli.

**Hisoblash modeli** (model of computation, MOC) - hisoblash tizimi elementlarining o'zaro ta'siri qonunlari to'plami.

**Hisoblash modeli** - hisoblash jarayonini tashkil qilish uchun qoidalar to'plami, uning doirasida rasmiy tahlil qilish mumkin.

**Hisoblash modeli** - bu ba'zi bir hisoblash tizimi modelining atomar qismlarining ko'p qismlarining o'zaro bog'liqligi va harakati tartibga solinadigan rasmiy qoidalar to'plami.

**Hisoblash modeli** - bu aniqlangan paradigma (qoidalar to'plami), bu hisoblash jarayoni, ma'lumotlar almashish usullari va individual funksional elementlarning o'zaro ta'sirini tavsiflaydi.

**Hisoblash modeli** - bu loyiha spetsifikatsiyalari va loyiha qarorlarini taqdim etish uchun aniq bir rasmiyatchilikdir.

**Hisoblash modeli** - foydalanuvchiga hisoblagichning hisoblash imkoniyatlarini va ulardan foydalanish qoidalarini namoyish etadigan matematik model.

Hisoblash nazariyasi va hisoblashning murakkabligi nazariyasi hisoblash modeliga nafaqat hisoblash uchun ishlatiladigan ruxsat etilgan operatsiyalar to'plamini belgilash, balki ularni qo'llashning nisbiy xarajatlari sifatida qaraydi. Hisoblashning ma'lum bir modeli tanlangan bo'lsa, zarur hisoblash manbalarini, ish vaqtini, xotira hajmini, shuningdek algoritmlar yoki kompyuterning cheklanishini tavsiflash mumkin. Modelga yo'naltirilgan muhandislikda hisoblash modeli va uning tanlovi, agar uning alohida qismlarining xatti-harakatlari ma'lum bo'lsa, butun tizim qanday ishlashi haqidagi savolga javob beradi. Hisoblashlarning murakkabligini asimptotik baholashda hisoblash modeli har birining narxi ma'lum bo'lgan yo'l qo'yiladigan ibtidoiy operatsiyalar orqali aniqlanadi. Amaldagi operatsiyalar to'plamiga va ularning hisoblash murakkabligiga qarab, bir qator hisoblash modellari ma'lum. Ular quyidagi keng toifalarga bo'linadi: algoritmik muammolarni echish uchun hisoblashning murakkabligini pastki chegarasini olish uchun ishlatiladigan algoritm va qaror modellarining hisoblash qobiliyatini isbotlash va hisoblash murakkabligining yuqori chegarasini olish uchun ishlatiladigan mavhum mashinalar (mavhum kalkulyatorlar).

Turli xil hisoblash modellariga tegishli tillarga misollar :

- C, Paskal, Ada - imperativ model yoki Fon Neumann modeli;
- VHDL, Verilog - diskret hodisalar modeli;
- Prolog, Refal - sentimental model;
- XML - ma'lumotlar ierarxik modeli;
- SQL - relyatsion model;
- Lisp - bu funksional model.

### 8.1.3. Dasturlash uslubi

**Dasturlash uslubi** - bu dasturlarni tuzishda ba'zi mantiqlarga asoslangan ichki kontseptual mos keladigan vositalar to'plami. Dasturlash uslubi kodlash uslubidan ajralib turishi kerak, bu esa

dastlabki matnni formatlashning o‘ziga xos usuliga o‘tadi. Dasturlash va dasturlash atamalarini aniqroq aniqlash kerak, shunda noaniqliklar yuzaga kelmasligi kerak. Hozirgi vaqtda dastur deganda Fon Neumann mashinasi bajaradigan harakatlar ketma-ketligi tushuniladi va Von Neumann mashinasi asosida dasturlashning strukturaviy uslubi dasturlash uslubi deb hisoblanadi. Shu nuqtai nazardan, dasturni VHDL yoki Verilogda yozilgan matn, shuningdek boshqa hisoblash modellarida ishlaydigan tizimlarni tavsiflovchi matn sifatida ko‘rib chiqish mumkinmi, aniq emas. Nepeyvoda dasturlash uslublarining quyidagi tasnifini beradi:

- sentimental dasturlash (**Refal, Prolog**);
- funktsional dasturlash (**Lisp**);
- avtomatik dasturlash;
  - voqealarni dasturlash;
- tizimli dasturlash (**C**);
- parallel dasturlash;
- ob'ektga yo‘naltirilgan dasturlash.

Dasturlash uslubi hisoblash modelining ustida joylashgan, chunki hisoblash modeli ma'lum bir uslub amalga oshiriladigan qoidalar to‘plamini belgilaydi. Muammoning echilishi, dasturlash uslubi va hisoblash modeli o‘rtasida kontseptual qarama-qarshiliklar bo‘lmasligi kerak. Aks holda, ishlab chiqilayotgan tizimning murakkabligida keskin sakrash bo‘ladi, xatolar va muddatlar soni ko‘payadi va loyiha byudjeti tezda tugaydi.

Afsuski, aksariyat dasturchilar ushbu muammoni sezmaydilar, bu odatda loyiha byudjetining ko‘payishiga va ko‘plab xatolarga olib keladi. Dasturlash uslubi va hisoblash modeli o‘rtasidagi uyg‘unlikning buzilishiga misol qilib, oqimni hisoblash modeli sun‘iy ravishda yaratilgan va tuzilgan dasturlash (ko‘pincha C tilida) dasturlash uslub sifatida ishlatiladigan RT operatsion tizimini keltirish mumkin. Shunga o‘xshash muammolar zamonaviy umumiy mo‘ljallangan operatsion tizimlarda (Microsoft Windows 2000 / XP / Vista, Linux, FreeBSD, Mac OS va boshqalar) va ular uchun dasturlash tizimlarida (**Java, C #, C ++** va boshqalar) mavjud.

Qarama-qarshiliklarning oldini olish uchun ular platforma asosidagi loyiha usulidan foydalanadilar, bunda har bir qavatda o‘z hisoblash modeli va u bilan uyg‘unlashgan dasturlash uslubi mavjud. Bunday holda, tizimning har bir qatlami keyingi qatlamlar uchun

poydevor bo‘lib, zamonaviy **RTOS**da uchraydigan tushunchalarning aralashishi sodir bo‘lmaydi.

Kolorado universiteti professori Robert Cebest o‘zining “Dasturlash tillarining asosiy tushunchalari” nomli kitobida dasturlash tillarini baholashning asosiy mezonlarini keltiradi:

- o‘qish;
- dasturlarni yaratish qulayligi;
  - ishonchlilik.

Dasturlash tillarining asosiy xususiyatlari quyidagilardan iborat:

- soddaligi (minimal til tuzilmalari);
  - ortogonallik (tilning yangi til tuzilmalarini, masalan, oz sonli elementar tuzilmalardan foydalangan holda ma’lumotlar tuzilmalarini yaratish qobiliyati );
    - abstraktsiyani qo‘llab-quvvatlash (asosiyini ajratib ko‘rsatish);
    - ekspressivlik (aniq va ixcham).

**O‘qish qobiliyati (Oson o‘qish )** - dasturlash tilida yozilgan dasturlarni o‘qish va tushunish qulayligi. Dastur ma'lum bir mavzu doirasiga mos tilda yozilgan bo‘lsa, tushunarli va sodda bo‘lishini tushunish kerak. Bunday holda, biz hisoblash modeli va dasturlash uslubi kabi tushunchalar haqida gapiramiz.

Dasturlarning oson o‘qilishi tilning soddaligi ta'sir qiladi. Til qancha turli xil tuzilishga ega bo‘lsa, undan foydalanish shunchalik qiyin bo‘ladi. Dasturlashning katta tillarini (masalan, **C ++**) ishlatadigan dasturchilar ko‘pincha faqat til konstruktsiyalarining pastki qismidan foydalanadilar. Shuni ta'kidlash kerakki, tilning haddan tashqari soddaligi ham dasturning o‘qilishiga salbiy ta'sir qiladi. Masalan, assembler - bu juda sodda til, ammo assemblerda yozilgan dasturni tushunish **C** yoki Paskalda yozilgan dasturga qaraganda bir necha baravar qiyinroq.

Tilning o‘qilishini pasaytiradigan yana bir xususiyati - bu xususiyatlarning ko‘pligi, ya'ni har qanday harakatni bajarishning bir necha usullarining mavjudligi.

Uchinchi muammo - operatorning haddan tashqari yuklanishi, ya'ni bir xil operatsiya belgisi uchun bir nechta ma'no mavjudligi. Agar dasturchi hech qanday asosli sababsiz operatorlarni ortiqcha yuklasa, dasturni tushunish juda qiyin bo‘lishi mumkin.

**Dasturlarni yaratish qulayligi** - ma'lum bir sohada dasturlarni yaratish qulayligini tavsiflaydi. Ko‘pincha, turli xil til konstruktsiyalarini o‘z ichiga olgan tillardan foydalanganda, dasturchilar

noto'g'ri til konstruktsiyalarini unutganda yoki ishlatganda vaziyat yuzaga keladi. Natijada kamroq oqlangan va samarasiz dasturlar mavjud.

Bir tomondan oz sonli elementlardan va ulardan foydalanish uchun izchil qoidalardan foydalanish (ortogonallik) juda ko'p miqdordagi ibtidoiylardan yaxshiroqdir. Boshqa tomondan, juda ortogonal tuzilish ham tildan foydalanishni murakkablashtiradi. Abstraktsiyani qo'llab-quvvatlash sizga kichik tuzilmalarni e'tiborsiz qoldirishga imkon beradigan murakkab tuzilmalar va operatsiyalarni aniqlash va undan keyin foydalanishga imkon beradi.

Abstraktsiyaning oddiy namunasi - bu subdastur. Tilning ekspresivligi sizga yanada kengroq va ixcham til konstruktsiyalaridan foydalangan holda dasturlar yozishga imkon beradi.

Agar dastur har qanday sharoitda o'z maqsadiga muvofiq bo'lsa, u ishonchli deb hisoblanadi. Tilning ishonchligiga ta'sir ko'rsatadigan eng muhim omillardan biri bu turni tekshirish. Tekshiruv mavjud bo'lmagan (yoki yo'q) tillar dasturchilarga xatolarga yo'l qo'yishga imkon beradi.

Tillarga bo'lgan talablar umumiy talablar asosida shakllantiriladi:

- ishonchlilik;
- real vaqt;
- oddiylik;
- Ekspressivlik;
- ortogonallik.

## **8.1.4 O'rnatilgan tizimlarini loyihalashda ishlatiladigan tillarning umumiy tahlili**

### **C dasturlash tili**

C - bu majburiy dasturlash tili yoki Fon Neyman hisoblash modeliga asoslangan til. Avtomatlashtirish piramidasiining deyarli barcha darajalarida qo'llaniladi. Eng keng tarqalgani piramidaning pastki (3 va 4) darajalarida bo'lgan.

Kompilyatorlarni turli xil apparat platformalariga ko'chirishning soddaligi va qulayligi, ko'rsatgichlar va bit operatsiyalarining mavjudligi tizim dasturlash, ko'milgan tizimlar va mikrokontrollerlarni dasturlash uchun C tilidan foydalanishni qulaylashtiradi. Bir tomondan C tilining zohidligi, boshqa tomondan funktsionalligi va kuchliligi KISS4 printsiyiga yaxshi mos keladi. C dasturlash tili 1970-yillarning

boshlarida Bell Labs xodimlari Ken Tompson va Denis Ritchi tomonidan B tilining rivojlanishi sifatida ishlab chiqilgan. **C UNIX** operatsion tizimida foydalanish uchun yaratilgan. O‘shandan beri u ko‘plab boshqa operatsion tizimlarga yuborildi va eng ko‘p ishlatiladigan dasturlash tillaridan biriga aylandi.

**C** tilini uning samaradorligi uchun qadrlanadi; tizimli dasturlarni yaratish uchun eng mashhur til. Bundan tashqari, ko‘pincha amaliy dasturlarini yaratish uchun ishlatiladi. **C** yangi boshlanuvchilar uchun mo‘ljallanmagan bo‘lsa-da, u dasturlashda dars berish uchun faol ishlatiladi. Keyinchalik, **C** tilining sintaksisi ko‘plab boshqa tillar uchun asos bo‘ldi.

**C** tili lakonizm bilan, oqimlarni boshqarish inshootlarining zamonaviy to‘plamlari, ma‘lumotlar tuzilmalari va keng ko‘lamli operatsiyalar bilan tavsiflanadi.

**C** tili elementlarining ko‘pi potentsial xavflidir va ushbu elementlarni suiiste‘mol qilish oqibatlari ko‘pincha oldindan aytib bo‘lmaydi. Kernigan shunday deydi: "**C** - bu o‘tkir o‘tkir asbob: siz u bilan nafis dastur va yaramas dastur yarata olasiz." Tilning nisbatan past darajasi tufayli xavfli elementlarni suiiste‘mol qilish holatlari aniqlanmaydi va ularni kompilyatsiya paytida ham, ish vaqtida ham aniqlab bo‘lmaydi. Ular ko‘pincha dasturni oldindan aytib bo‘lmaydigan xatti-harakatlariga olib keladi. Ba‘zan, til elementlaridan savodsiz foydalanish natijasida xavfsizlikning zaifliklari paydo bo‘ladi. Shuni ta‘kidlash kerakki, ushbu elementlarning ko‘pchiligidan foydalanish mumkin emas.

Xatolarning eng keng tarqalgan manbai mavjud bo‘lmagan element elementiga kirishdir. **C** tili to‘g‘ridan-to‘g‘ri statik massivlarni qo‘llab-quvvatlaganiga qaramay, uning massiv indekslarini tekshirish uchun vositalari yo‘q (chegarani tekshirish). Masalan, oltinchi elementga besh elementdan iborat qator yozish mumkin, bu, albatta, oldindan aytib bo‘lmaydigan natijalarga olib keladi. Bunday xatoning maxsus holati buferni to‘ldirish xatosi deb nomlanadi. Ushbu turdagi xatolar xavfsizlikning ko‘p muammolariga olib keladi. Xavfli vaziyatlarning yana bir mumkin bo‘lgan manbai ko‘rsatgich mexanizmi. Ko‘rsatkich xotiradagi mutlaqo istalgan ob‘ektga ishora qilishi mumkin.

**C** tili dasturchilariga ushbu va boshqa ko‘plab muammolarni hal qilishda yordam berish uchun kompilyatorlardan alohida ko‘plab vositalar yaratilgan. Bunday vositalar dastlabki kodni qo‘shimcha tekshirish va keng tarqalgan xatolarni qidirish uchun dasturlar,

shuningdek standart qatorga kirmaydigan qo‘shimcha funksiyalarni ta‘minlaydigan kutubxonalar, masalan, cheklovlarni yoki axlat yig‘ishning cheklangan shaklini tekshirish.

Nisbatan kam miqdordagi mikroprotessor ishlab chiqaruvchilari o‘z kompilyatorlarini bepul taklif qilishadi (masalan, Atmel- dan AVR Studio va Fujitsu- dan Softune Workbench ).

Tijorat kompilyatorlari quyidagi maxsus xususiyatlarga ega:

- rasmiy qo‘llab-quvvatlash;
- yaxshi hujjatlar;
- yaratilgan kodda kam xatolar (afsuski, har doim ham emas va umuman ham bo‘lmaydi);
- kodni optimallashtirish takomillashtirilgan;
- ko‘proq foydalanuvchilar bilan do‘st interfeysi.

Qoida tariqasida, boshlang‘ich ishlab chiquvchilar tijorat vositalarini tushunish osonroq. O‘rnatilgan tizimlarda ishlatiladigan bir qator kompilyatorlar GNU GPL litsenziyasi ostida chiqariladi. GNU GPL-ning maqsadi foydalanuvchiga dasturlarni nusxalash, o‘zgartirish va tarqatish (shu jumladan tijorat asosida) dasturlarini (mualliflik huquqi to‘g‘risidagi qonun bilan taqiqlangan) sukut saqlash, shuningdek barcha hosilaviy dasturlarning foydalanuvchilari yuqoridagi huquqlarga ega bo‘lishlarini ta‘minlashdir.

### **C ++ dasturlash tili**

C++ - bu kompilyatsiya qilingan, kuchli yozilgan, umumiy maqsadli dasturlash tili. U turli xil dasturiy paradigmalarni qo‘llab-quvvatlaydi : protsessual, umumlashtirilgan, funksional; Ob'ektga yo‘naltirilgan dasturlashni qo‘llab-quvvatlashga katta e‘tibor beriladi. 90-yillarda bu til eng keng tarqalgan umumiy maqsadli dasturlash tillaridan biriga aylandi. C ++ ni yaratishda ular C tili bilan moslikni saqlashga intilishdi. Ko‘pgina C dasturlari C ++ kompilyatori bilan yaxshi ishlaydi. C ++ da C sintaksisi asosida sintaksis mavjud. C ++ ning C bilan solishtirganda yangiliklari:

- sinflar orqali ob'ektga yo‘naltirilgan dasturlashni qo‘llab-quvvatlash ;
- andozalar orqali umumlashtirilgan dasturlashni qo‘llab-quvvatlash;
- standart kutubxonaga qo‘shimchalar;
- qo‘shimcha ma‘lumotlar turlari;
- istisnolar;

- ism maydonlari;
- oʻrnatilgan funktsiyalar;
- operatorning ortiqcha yuklanishi;
- funktsiyalarni ortiqcha yuklash;
- havolalar va xotirani bepul boshqarish operatorlari.

C++ tili 1980-yillarning boshlarida, Bell Laboratories xodimi B.Stroustrup oʻz ehtiyojlarini qondirish uchun C tilini takomillashtirish boʻyicha bir qator takliflarni qabul qilganida paydo boʻlgan. Rasmiy standartlashtirish boshlanishidan oldin, C++ tili asosan dasturiy hamjamiyatning soʻrovlariga javoban Stroustrup kuchlari tomonidan ishlab chiqilgan. 1998 yilda C++ tilining xalqaro standarti tasdiqlandi : ISO/IEC14882:1998 " Standard for the C++ Programming Language "; 2003 yilda standartga texnik tuzatishlar kiritilgandan soʻng, ushbu standartning joriy versiyasi ISO / IEC 14882: 2003 hisoblanadi. "C++" nomi C dan keladi, unda ++ unary operatori bu oʻsishni bildiradi. "C++ loyihasi va rivojlanishi" kitobida B.Stroustrup C++ loyihasida foydalangan ba'zi qoidalarni tasvirlaydi. Ushbu qoidalarni bilish C++ nima uchun paydo boʻlganligini tushunishga yordam beradi. Mana bu qoidalaridan ba'zilari. C++:

- Statik ma'lumotlarning turlari, C tilining tezkorligi va portativligi bilan universal til sifatida yaratilgan.

- Dasturlashning koʻpgina turlarini (protsessual dasturlash, ma'lumotlarning mavhumligi, ob'ektga yoʻnaltirilgan dasturlash va umumlashtirilgan dasturlash) bevosita va har tomonlama qoʻllab-quvvatlash uchun yaratilgan.

- Dasturchiga tanlash erkinligini berish uchun yaratilgan, garchi u unga boshqasini tanlash imkoniyatini bergan boʻlsa.

- C bilan moslikni maksimal darajada oshirish uchun ishlab chiqilgan va shu bilan dasturlashdan C ga osongina oʻtish mumkin.

- Platformaga bogʻliq yoki universal boʻlmagan xususiyatlardan qoching.

- Hech qanday xususiyatlardan foydalanmaydigan dasturga ortiqcha yuk yuklamaydi.

- Juda murakkab dasturlash muhitini talab qilmaydigan qilib yaratilgan.

Oʻrnatilgan tizimlar uchun C++ tilining soddalashtirilgan lahjasi ishlab chiqilgan - **embedded C++**. Asosiy farqlar qator xususiyatlarning yoʻqligidir.

- koʻp meros;



- virtual baza sinflari;
- ish vaqtidagi turlari to'g'risida ma'lumot (typid);
- Yangi uslubi turi konvertatsiya (static cast, dynamic cast, reinterpret cast, const cast);
- ismlarning bo'sh joylari ;
- istisnolar;
- shablonlar.

## **Java platformasi**

**Java** - 1991 yildan beri Sun Microsystems tomonidan ishlab chiqilgan va 1995 yil 23 mayda rasmiy ravishda chiqarilgan ob'ektga yo'naltirilgan dasturlash tili. Dastlab, yangi dasturlash tili Oak (James Gosling) deb nomlangan va iste'molchi elektronikasi uchun ishlab chiqilgan, ammo keyinchalik Java deb o'zgartirilgan va appletlar, dasturlar va server dasturlarini yozish uchun ishlatilgan. Java dasturlarini virtual Java mashinasida (JVM) bajarilgan bayt kodiga tarjima qilish mumkin - bu bayt kodni qayta ishlaydigan va ko'rsatmalarni tarjimon sifatida uskunaga etkazadigan dastur, ammo farqli o'laroq, bayt kodi, matndan farqli o'laroq, tezroq ishlov beradi. Dasturlarni ishga tushirishning ushbu usulining afzalligi - bu virtual mashinani qo'llab-quvvatlaydigan har qanday qurilmada Java dasturlarini ishga tushirishga imkon beradigan OT va apparat vositalaridan to'liq mustaqillik.

Java texnologiyasining yana bir muhim xususiyati dasturning bajarilishi virtual mashina tomonidan to'liq nazorat qilinishi sababli moslashuvchan xavfsizlik tizimidir. Belgilangan dastur ruxsatnomasidan oshgan har qanday operatsiyalar (masalan, ma'lumotlarga ruxsatsiz kirishga urinish yoki boshqa kompyuterga ulanish ) darhol uzilishlarga olib keladi. Bu foydalanuvchilarga Java-da yozilgan dasturlarni o'zlarining kompyuterlariga (yoki boshqa qurilmalarga, masalan, mobil telefonlarga) noma'lum manbalardan, virus infeksiyasidan, qimmatli ma'lumotlarning etishmasligidan va boshqalardan yuklab olish imkonini beradi.

Ko'pincha ushbu yondoshuvning kamchiliklari bayt kodni virtual mashina tomonidan bajarilishi Java tilida amalga oshirilgan dasturlar va algoritmlarning ishlashini kamaytirishi mumkinligini o'z ichiga oladi. Ushbu bayon Java virtual mashinasining birinchi versiyalari uchun to'g'ri, ammo yaqinda u deyarli o'z dolzarbligini yo'qotdi. Bunga bir qator yaxshilanishlar yordam berdi : JITs (Just-In-Time compiler)

texnologiyasidan foydalanish, dasturni bajarishda bytekodni mashina kodiga sinf versiyalarini mashina kodida saqlash imkoniyati, standart kutubxonalarda ona kodidan keng foydalanish, shuningdek dasturiy ta'minot. tezlantirilgan baytlarni qayta ishlashni ta'minlaydigan vositalar (masalan, ba'zi ARM protsessorlari tomonidan qo'llab-quvvatlanadigan Jazelle texnologiyasi ). Java ichida uchta asosiy texnologiyalar oilasi mavjud:

- J2EE yoki Java EE (v1.5 dan boshlab) - korxonada darajasidagi dasturlarni yaratish uchun Java Enterprise Edition ;
- J2SE yoki Java SE (v1.5 dan boshlab) - Java Standard Edition, birinchi navbatda ish stoli tizimlari uchun maxsus dasturlarni yaratish uchun ;
- J2ME, Java ME yoki Java Micro Edition, cheklangan hisoblash quvvatiga ega qurilmalarda, shu jumladan uyali telefonlar, PDA, o'rnatilgan tizimlarda foydalanish uchun.

Microsoft JVM JVM-ning oldingi versiyasi (SUN JVM v.1.1.3-ga o'xshash) ko'p jihatdan Windows platformasi uchun xususiy qo'llab-quvvatlash maqsadida Sun Microsystems tomonidan taklif qilingan til standartlaridan chetga chiqadi. Microsoft ma'lumotlariga ko'ra, SUN-JVM J2SE spetsifikatsiyasiga mos keladigan MS-J # spetsifikatsiyasi qo'llab-quvvatlanadi.

## **Java 2 Micro Edition**

**Java 2 Micro Edition (J2ME)** - bu Java platformasi kontsepsiyasiga asoslangan va mobil telefonlar, shaxsiy tashkilotchilar, raqamli televidenie qabul qiluvchilar va boshqalar kabi iste'molchi elektronika qurilmalarida Java dasturlarini ishlashga mo'ljallangan Sun Microsystems texnologiyalarining quyi qismi.

J2ME Java bayt kodini bajarishga qodir bo'lgan virtual mashinaga asoslangan. J2ME iste'molchi elektronikasi qurilmalarida Java dasturlarini samarali bajarilishini ta'minlash uchun ishlab chiqilgan bo'lib, ular cheklangan ishlov berish kuchi, cheklangan xotira, kichik displey hajmi, akkumulyator quvvati va past tezlik va nomuvofiq aloqa qobiliyatlari bilan ajralib turadi. Odatdagi zamonaviy uyali telefonda chastotasi 50 MGts bo'lgan 32 bitli RISC protsessori mavjud bo'lib, taxminan 4 Mb tezkor xotira, 2 dyuymli rangli displey va Internetga tezlikda GPRS bilan ulanish qobiliyatiga ega 172 Kb / s gacha, bu juda ishonchsiz, chunki ma'lumotlar uzatish tezligi kutilmaganda pasayishi yoki ulanish butunlay yo'qolishi mumkin.

J2ME ikkita virtual konfiguratsiyani belgilaydi, ular virtual mashinaga talablarni belgilaydilar, yoki boshqacha qilib aytganda, virtual mashina ishlashga qodir bo'lgan standart Java tilining pastki qismini, shuningdek, asosiy sinflarning minimal to'plamini belgilaydilar:

- **CLDC** (Connected Limited Device Configuration- ulangan cheklangan qurilmaning konfiguratsiyasi)
- **CDC** (Connected Device Configuration- ulangan qurilma konfiguratsiyasi).

J2ME shuningdek yuqorida aytib o'tilgan konfiguratsiyalarni to'ldiradigan va kengaytiradigan bir nechta deb nomlangan profillarni aniqlaydi, xususan, dastur modelini, grafik interfeys imkoniyatlarini, shuningdek, aloqa funksiyalarini ( masalan, Internetga kirish) va boshqalarni aniqlaydi.

Hozirgi kunda eng keng tarqalgan konfiguratsiya CLDC bo'lib, u uchun MIDP (Mobile Information Device Profile) profili ishlab chiqilgan - bu mobil funksiyalar uchun ma'lumot funksiyalari). MIDP MIDlet-ni belgilaydi - bu ixcham Java dasturi, uni tarmoq orqali uzatish va mobil qurilmaga o'rnatish uchun qulay qiladi. J2ME / CLDC uchun yana bir mashhur profil - bu o'z iMode xizmati uchun NTT DoCoMo tomonidan ishlab chiqilgan DoJa. CLDC konfiguratsiyasi aksariyat zamonaviy mobil telefonlarda va ko'chma tashkilotchilarda muvaffaqiyatli qo'llanilmoqda. Sun Microsystems ma'lumotlariga ko'ra, 2004 yil oxiriga qadar dunyo bo'ylab ushbu Java konfiguratsiyasiga ega 570 milliondan ortiq mobil qurilmalar etkazib berildi. Bu J2ME-ni dunyodagi dominant Java texnologiyasiga aylantiradi. Uyali telefonlar ishlab chiqarish hajmi Java dasturlarini (masalan, shaxsiy kompyuterlar) ishlashga qodir bo'lgan boshqa kompyuter qurilmalari sonidan sezilarli darajada oshib ketadi.

### **.NET platformasi**

**.NET Framework** - bu Microsoft-dan umumiy dasturlar va veb-lovalarni yaratish uchun dasturiy ta'minot texnologiyasi. Microsoft.NET-ning asosiy g'oyalardan biri bu turli tillarda yozilgan turli xil xizmatlarning o'zaro muvofiqligi. Masalan, Microsoft.NET uchun C++ da yozilgan xizmat Delphi-da yozilgan kutubxonadan sinf usuliga kirishi mumkin; C# da Visual Basic.NET-da yozilgan sinfdan meros qoldiradigan sinfni yozishingiz mumkin va C# da yozilgan usul bilan chiqarilgan istisno Delphi-da ushlanib qolinishi mumkin..NET-

dagi har bir kutubxona (yig'ilish) o'z versiyasi haqida ma'lumotga ega, bu esa assambleyalarning turli xil versiyalari o'rtasida yuzaga kelishi mumkin bo'lgan ziddiyatlarni bartaraf etishga imkon beradi.

.NET - Microsoft korporatsiyasining xususiy texnologiyasi. Biroq, Novell bilan tuzilgan shartnomadan so'ng, Mono texnologiyasi Unix-ga o'xshash tizimlarda (GNU/Linux, Mac OS X).NET dasturini amalga oshirish sifatida tan olindi. Shu bilan birga, shartnoma Novell va Novell mijozlariga, shuningdek ASP.NET, ADO.NET va Windows.Forms texnologiyalari ECMA/ISO tomonidan standartlashtirilmagan va ulardan Mono-da foydalanish tashqaridan da'vo qilish xavfi ostida. Mono ASP.NET, ADO.NET va Windows.Forms dasturlarini taqdim etadi, ammo ushbu APIlardan chetlab o'tishni tavsiya qiladi.

Ilovalar matn muharririda ham ishlab chiqilishi va konsol kompilyatoridan foydalanishlari mumkin. Java texnologiyasi singari.NET rivojlanish muhiti virtual mashinada bajarish uchun baytekod yaratadi. Ushbu mashinaning.NET-ga kirish tili MSIL (Microsoft Intermediate Language) yoki CIL (Common Intermediate Language, keyinchalik) yoki oddiygina IL deb nomlanadi. Bytecode-dan foydalanish, loyihalashtirilgan loyiha darajasida (.NET: montaj nuqtai nazaridan) o'zaro faoliyat platformani olish imkoniyatini beradi va nafaqat manba matni darajasida, masalan, C da. O'rnatishni CLR ish vaqtida ishlatishdan oldin, bayt kod o'rnatilgan JIT kompilyatori tomonidan maqsad protsessorning mashina kodiga aylantiriladi..NET Framework bilan ta'minlangan NGen.exe yordam dasturidan foydalanib, tanlangan platforma uchun mahalliy kodga assambleyani kompilyatsiya qilish ham mumkin.

### **ADA dasturlash tili**

**Ada tili**, 1979-1980 yillarda AQSh Mudofaa vazirligi tomonidan amalga oshirilgan loyiha natijasida, o'rnatilgan tizimlar (ya'ni real vaqtda ishlaydigan avtomatlashtirilgan komplekslarni boshqarish tizimlari) uchun yagona dasturlash tilini yaratish maqsadida yaratilgan. Bu, birinchi navbatda, harbiy ob'ektlarni (kemalar, o'rnatilgan tizimlar, tanklar, raketalar, qobiqlar va boshqalarni) bortda boshqarish tizimlari edi. Ishlab chiquvchilarga universal til yaratish vazifasi yuklanmagan, shuning uchun Ada mualliflari tomonidan qabul qilingan qarorlar tanlangan mavzu doirasining xususiyatlari nuqtai nazaridan qabul qilinishi kerak.

**Ada** - bu parallel jarayonlarni amalga oshirish uchun yuqori darajadagi dasturlash vositalarini o'z ichiga olgan tarkibiy, modulli, ob'ektga yo'naltirilgan dasturlash tili. Ada sintaksisi Algol yoki Paskal kabi tillardan meros bo'lib, kengaytirilgan, shuningdek, yanada qat'iy va mantiqiy bo'lgan. Ada - bu juda kuchli yozilgan til, u hech qanday turga ega bo'lmagan ob'ektlar bilan ishlashni o'z ichiga olmaydi va avtomatik turdagi konversiyalar mutlaq minimal darajaga tushiriladi.

Ishonchlilik talablariga javob berish uchun til tuzilgan bo'lib, kompilyatsiya bosqichida imkon qadar xatolar aniqlanadi. Bundan tashqari, tilni rivojlantirishning talablaridan biri dastur matnlarini, hatto yozishni osonlashtiradigan darajada osonlikcha o'qish edi. Ushbu yondashuvning natijasi, ba'zi bir "og'ir" sintaksis va ko'plab cheklovlar bo'lib, ular ko'pincha professional dasturchilar tomonidan "ahmoq" va "keraksiz" deb qabul qilinadi. Bu jahannam tushunchasini tilni ishlatish uchun murakkab, noaniq va noqulay sifatida shakllanishiga olib keldi. Bu ko'rinish faqat qisman haqiqatdir: Ada-da oddiy dastur yozish boshqa rasmiy tillarga qaraganda ko'proq vaqt talab qiladi, masalan C, lekin dasturlarni moslashtirish va xizmat ko'rsatish, ayniqsa katta va murakkab bo'lganlar juda soddalashtirilgan. Ada-da dasturiy ta'minotni ishlab chiqarish odatda 60% arzonroq va ishlab chiqilgan dastur C tilidan foydalanishdan 9 baravar kam kamchiliklarga ega.

Aga tili AQSSh va Evropada murakkab harbiy loyihalarni ishlab chiqishda, asosan o'rnatilgan tizimlarda va nafaqat harbiy maqsadlarda qo'llaniladi.

### **Esterel dasturlash tili**

**Esterel** bu aniq belgilangan rasmiy bazaga va to'liq amalga oshiriladigan imperativ va parallel til. ESTERELning asosiy tushunchasi- bu voqea. Hodisa ma'lumotlarni uzatuvchi signallarni yuborish yoki qabul qilishga mos keladi. Esterel sinxron modelga asoslangan. Ushbu sinxronizm vaqtini oqlashni soddalashtiradi va determinizmni kafolatlaydi.

Xozirgi kunda Esterel on - chip tizimlarini loyihalash uchun mo'ljallangan Esterel Studio (Synfora) dasturiy tizimida qo'llaniladi.

### **Luster dasturlash tili**

**Luster** SDF (sinxron ma'lumot uzatish) modeliga asoslangan va reaktiv tizimni dasturlash uchun mo'ljallangan deklarativ dasturlash tili. Til 1984 yilda Frantsiyada ishlab chiqila boshlandi. Yorug'lik tili

aerokosmik sohada muhim dasturlarni amalga oshirish uchun faol ishlatiladi.

## **8.2. O‘rnatilgan tizimlarni dasturiy taminotini nosozliklarni tuzatish**

### **8. 2.1. O‘rnatilgan tizim dasturiy taminotini nosozliklarni tuzatish xususiyatlari**

**Nosozliklarni tuzatish** - tizimni loyihalash jarayonida uni ish holatiga keltirish. **Nosozliklarni tuzatish** - bu ma'lum bo‘lgan xatoning aniq mohiyatini aniqlashga, so‘ngra ushbu xatoni tuzatishga qaratilgan faoliyat. **Nosozliklarni tuzatuvchi** - bu xato topishga yordam beradigan vosita. O‘rnatilgan tizimlarning asosiy xususiyati shundaki, tuzatilgan dastur ishlab chiqish muhitida emas, balki uzoqdagi mashinada ishga tushiriladi. Bundan tashqari, o‘rnatilgan dastur atrofdagi apparat bilan chambarchas bog‘liq.

O‘rnatilgan tizim dasturiy ta'minotini tuzatishni ikki yo‘li mavjud:

- nosozliklarni tuzatish uchun dasturiy ta'minotni simulyatsiya muhitiga kiritish;
- nosozliklarni tuzatish agentini maqsadli tizimga kiritish.

Ikkala yondashuv ham ularning afzalliklari va kamchiliklariga ega. Haqiqiy qurilmalarni aniq taqlid qiladigan simulyatorni yaratish juda muammoli, shuning uchun model va haqiqiy uskunalar har doim boshqacha bo‘ladi. Bundan tashqari, ishlab chiqarilgan mikrokontrollerlar va turli xil periferik qurilmalar (masalan, xotira chiplari, DAClar, ADC‘lar, tarmoq kontrollerlari, LCD va boshqalar) juda katta, chunki simulyator ishlab chiqaruvchilari shunchalik ko‘p kutubxona elementlarini ko‘paytirishga jismonan qodir emaslar.

Nosozliklarni tuzatish agentlarini amalga oshirish ham o‘z muammolariga ega. Boshlash uchun ikkinchi turdagi nosozliklarni tuzatish tizimini tashkil qilish variantlarini ko‘rib chiqing.

- Ayrim interfeys (masalan, RS-232 yoki Ethernet) orqali nosozliklarni tuzatuvchi bilan o‘zaro aloqada bo‘lgan dasturiy tuzatish vositasi.
- Nosozliklarni tuzatish agenti sifatida maxsus mikrokontrollerni almashtirishga imkon beruvchi maxsus emulyator boshi ishlaydi. Odatda, bunday tizimlar elektron davriy emulyator deb ataladi.

- Nosozliklarni tuzatish agenti mikrokontrollerga oʻrnatilgan qoʻshimcha qurilmadir. Hozirgi vaqtda bunday agentning eng keng tarqalgan varianti - JTAG.

Uchala variant ham bir xil natijani beradi:

- manbada xatolarni tuzatish imkoniyati;
- registrlari, steklar va xotiralarni koʻrish;
- uzilish nuqtalari bilan ishlash;
- bosqichma-bosqich xatolarni tuzatish.

Dasturiy taʼminotni tuzatish agenti eng sodda va arzon. Bundan tashqari, bu eng sekin va eng ishonchsiz. Oʻchirish emulyatorlari va JTAG yordamida yaxshiroq natijalarga erishiladi. Tarmoq ichidagi emulyator - bu eng qimmat, ammo ayni paytda eng kuchli sozlash vositasi. JTAG interfeysi ancha arzon va koʻp qirrali, chunki har bir mikrokontroller uchun emulyator sotib olishning hojati yoʻq. Afsuski, JTAG ish tezligi boʻyicha elektron emulyatorlardan sezilarli darajada pastdir.

## 8.2.2. Nosozliklarni tuzatish vositalari

### Simulyator

**Simulyator** - bu ob'ektning xatti-harakati va tuzilishini toʻliq yoki qisman taqlid qiladigan tizim. Simulyator hisoblash tizimining dasturiy va apparat tarkibiy qismlarini nosozliklarni tuzatish, sinovdan oʻtkazish va tekshirish vositalariga tegishli.

Dasturni amalga oshirishda simulyator asboblarni dastgohida bajarilishi mumkin. Oʻrnatilgan tizimlar va SoClarni dasturlashda protsessor simulyatori eng koʻp ishlatiladi. Tashqi tomondan, bunday simulyator odatdagi tuzatuvchiga oʻxshaydi.

Turli xil ASIC-larni taqlid qilishda FPGA asosidagi protsessorlar, grafik tezlatgichlar, dasturiy va apparat simulyatorlari qoʻllaniladi. Ushbu yondashuv sof dasturiy simulyatsiya bilan solishtirganda sinov va tekshirish vaqtini sezilarli darajada kamaytirishi mumkin. Dasturiy-apparat simulyatorining narxi dasturiy taʼminot narxidan ancha yuqori.

Simulyatorning afzalliklari oʻrganilayotgan tizim ichidagi kuzatuvchi nuqtai nazaridan real vaqt rejimida ishlashga imkon beradigan oʻrganilayotgan tizimning muhitini taqlid qilish qobiliyatidir. Afsuski, simulyatorlarda har doim modellash, mavhumlashtirish, amalga oshirishda xatolar va hokazolarning noaniqligi tufayli yuzaga keladigan instrumental xato mavjud.

## **Elektronli emulyator**

**Elektronli emulyator** - bu sozlash uchun ishlatiladigan, odatda qo‘shimcha kontaktlarga ega mikroprotessor shaklida bajariladigan qurilma. Elektron emulyatorlar maqsadli mikroprotessor yoki mikrokontroler o‘rniga diskka o‘rnatilgan yoki sinovdan o‘tgan tizimga ulanadi va sozlash jarayonida tizimning holatini moslashuvchan boshqarishga, turli ob'ektlarning holati to‘g‘risida ma'lumot to‘plashga, turli xil rejimlarda foydalanuvchi dasturlarini bajarishga imkon beradi: real vaqt rejimida (berilgan manzildan dasturning uzluksiz bajarilishi), qadam-baqadam rejimida, ishlayotgan rejimda ma'lum bir shartga ko‘ra to‘xtaydi. Ko‘pincha ular sizga nafaqat maqsadli protsessorni, balki xotira, soat, kirish-chiqish qurilmalarini ham taqlid qilishga imkon beradi.

Elektron emulyatorlardan foydalanish dasturiy va apparat vositalarini tuzatish va sinovdan o‘tkazish bilan bog‘liq deyarli barcha muammolarni hal qilishga imkon beradi. Afsuski, electron emulyatorlarning muhim kamchiliklari ularning juda yuqori narxidir. Elektron emulyatorlari maqsad darajasidagi tizimning bir qismini apparat darajasida almashtirish vositasidir. Hozirgi vaqtda ikkita asosiy variant keng tarqalgan:

- protessor emulyatori;
- ROM emulyatori.

So‘nggi paytlarda, faqat o‘qish uchun mo‘ljallangan xotira (FLASH yoki OTP shaklida) bo‘lgan kompyuterlarning paydo bo‘lishi bilan emulyatorlarning ikkinchi versiyasi asta-sekin ishlay qoldi. Amalda, elektron emulyatori dasturiy simulyator kabi funktsiyalar to‘plamiga ega. Bu erda asosiy farqlar:

- nosozliklarni haqiqiy uskunada o‘rnatish mumkin (bu dasturiy muhitni simulyatsiya qilish imkoniyatini istisno qilmaydi);
- nosozliklarni tuzatish real vaqtda amalga oshiriladi.

Elektron emulyator uchun qiziqarli variant - JTAG. Klassik protessor emulatorlarda, nosozliklarni uchun, markaziy protessor (yangi protessorlari bir qator uchun, bir emulyatsiya boshi bilan almashinadi emulyatsiya bosh lehimli bevosita ulanishi mumkin billurday). Bu taxtada protessor ostiga panelni qo‘yish zarurligiga olib keladi, bu tizimning ishonchligini pasaytiradi. JTAG texnologiyasidan foydalanganda emulyator doskaga maxsus texnologik ulagich orqali ulanadi. Bunday holda, protessor olinmaydi. JTAG sizga protessor yadrosini o‘chirish va manzil, ma'lumot va boshqaruv shinasini



to'g'ridan-to'g'ri boshqarish imkonini beradi. Afsuski, barcha protsessor pinlarini boshqarish uchun JTAG porti orqali ko'p ma'lumot uzatilishi kerak. Shuning uchun, real vaqtda sozlash ( protsessor chastotasida ) mumkin emas. JTAGning katta afzalligi - simulyator apparati soddaligi. Masalan, eng oddiy holatda, JTAG portini oddiy kompyuterning parallel LPT portiga ulash kifoya.

JTAG quyidagi asosiy vazifalar ro'yxatini hal qilishga mo'ljallangan:

- ishlab chiqarishdagi nuqsonlarni aniqlaydigan dastlabki sinov ;
- dasturlashtiriladigan komponentlar uchun kerakli konfiguratsiyani etkazib berish;
- turli xil tuzatish mexanizmlarini (statik yoki dinamik) va monitoring rejimini qo'llab-quvvatlash.

Tekshiruvning chegaraviy mexanizmidan foydalanishning moslashuvchanligiga standartning bir qator xususiyatlari erishiladi, ularning asosiylari:

- ushbu standartni qo'llab-quvvatlaydigan bir nechta qurilmalarning parallel muvofiqlashtirilgan ishlashi (raqam interfeysning elektr parametrlari bilan cheklangan );
- mexanizmning o'zini kengaytirish imkoniyati (qo'shimcha buyruqlar va ma'lumotlar formatlarini kiritish).

Birinchi bir jinsli yoki heterojen tuzilishga ega bo'lgan ko'p protsessorli tizimlarni ishlab chiqishda chegara ko'rish mexanizmidan foydalanish imkoniyatini aniqlaydi. Ikkinchisi mexanizmni samarali qo'llash mumkin bo'lgan vazifalar doirasini kengaytiradi, bu, masalan, disk raskadrovka, tashxis qo'yish va monitoring muammolarini hal qilishga imkon beradi.

### **JTAG to'plamini amalga oshirish**

O'rnatilgan tizimning loyihalashda mavjud muammolarni hal qilishda loyihachelar dastlabki sinov va elektron sxemani ishga tushirish vositalarini ishlatadilar. Shu maqsadda instrumental va texnologik xususiyatga ega ixtisoslashgan qurilma yoki xizmat ko'rsatish mexanizmi ishlab chiqilmoqda. Uskunaning funktsionalligi va tarkibi loyihaning xususiyatlari bilan belgilanadi. O'zaro bog'liq vositalar - bu ishlab chiqarish dasturi va maqsadli vositalarga asbobga kirishni ta'minlovchi dasturiy interfeyslarning kombinatsiyasi.

Instrumental mashinaning resurslari odatda qisman boshlash va dastlabki disk raskadrovka bosqichlarida tizimni masxara qilish bilan

ishlashning ko'p qismida chegara ko'rish mexanizmining til tavsifini tarjimonini amalga oshirish uchun etarli. Vaziyat tizimning prototiplari bilan farq qiladi, chunki loyiha vaqtida ular ajratilgan instrumental kanallarni kamaytirishga intilib, xizmat ko'rsatishning yaxlit mexanizmlarini shakllantiradilar. Bunday holda, initsializatsiya maqsadli funktsionallik va instrumental rejimning qo'llab-quvvatlashini birlashtiradigan ilgari sinov qilingan hisoblash komponentlaridan foydalangan holda amalga oshiriladi. Masalan, asboblarning kanali va chekka ko'rish mexanizmini qo'llab-quvvatlaydigan kanal birlashtirilishi mumkin. Bunday holda, tarjimonni instrumental mashinaga joylashtirish kanal cheklovlari tufayli ish samaradorligini pasayishiga olib keladi.

Tarjimonni (interpretator) chegaralarni tekshirish mexanizmini qo'llab-quvvatlovchi rezident dasturini amalga oshiradigan maqsadli tizimning bir qismi sifatida "xizmat" nazorati vositasi yordamida amalga oshirish muhimdir. Tarjimonni bunday rezident foydasiga hal qilish boshqaruvchining ishlov berish kuchiga va tarjima qilingan tilning murakkabligiga bog'liq bo'ladi.

Bugungi kunda ko'plab ishlab chiqaruvchilar turli xil apparat platformalari uchun manba kodida ushbu turdagi tarjimonlarni taqdim etadilar. Skanerlash texnologiyasidan foydalanishning samaradorligi ko'p jihatdan ikkita masalani hal qilishga bog'liq: JTAG tuzilishini tavsiflash va tasvirlangan zanjir bilan ishlash algoritmini shakllantirish. JTAG ( scan path ) ketma-ket bir necha komponentlar TDI va TDO signallari ulab olingan to'liq standart sinov avtobus anglatadi. Ushbu tushuncha quyidagi muammolarni hal qilish uchun ishlatiladi:

- chegara brauzerlari ko'rish mexanizmi qo'llab-quvvatlash individual mikrosxemalar ta'rifidir;
- jihatidan maqsadli tizimning tuzilishi tavsifi chegara brauzerlari ko'rish mexanizmi ;
- muayyan algoritm maqsadida ta'rifidir (ajratish).

Odatda haqiqiy tizimda bitta JTAG zanjiriga ulangan bir nechta mikrochiplar mavjud. Yuqorida ta'kidlab o'tilganidek, zanjirlar va ularning ierarxiyasini tavsiflash uchun BSDL va HSDL tillari qo'llaniladi. Ta'rif vositasi sifatida ular samarali, ammo amalda mavjud vositalar tomonidan qo'llab-quvvatlanmaydi. Natijada, ishlab chiquvchi JTAG asboblari zanjirini biron bir tarzda yoki norasmiy tarzda o'rnatadi, bu esa chegaralarni tekshirish mexanizmidan foydalanganda mehnat intensivligining keskin oshishiga olib keladi. Maqsadni taqsimlash

vazifasi JTAG zanjirining tuzilishini tavsiflashning standart vazifalari qatoriga kirmaydi. Biroq, amalda BSR-larning butun zanjirining birlashishiga teng bo'lgan BSR (chegaralarni tekshirish ro'yxati) bilan ishlash juda kam uchraydi. Odatda, ma'lum bir test yoki algoritm uchun BSR-dan o'ndan ortiq kataklar yoki bitta zanjirda bitta chip kerak bo'ladi. Ko'rinishidan, bunday tavsif uchun standart vosita mavjud emas. Bu, ehtimol, JTAG zanjirining tavsifining xususiyatlari (aslida) va JTAG tomonidan hal qilingan vazifalar qatoriga bog'liq. JSR-dan foydalanib, dasturlash vositalariga, masalan, flesh-xotiralarga havolalar mavjud, bu BSR-da manzillar, ma'lumotlar va boshqaruvlarni ajratishni talab qiladi. Biroq, taniqli dasturlarda bu norasmiy tarzda amalga oshiriladi. Adolatlilikda ta'kidlash kerakki, HSDL standarti alohida BSR kataklarini ularga o'zboshimchalik bilan nom berish orqali "ajratib ko'rsatish" xususiyatiga ega. Ammo bu maqsadni aniq tanlash deb nomlab bo'lmaydi, chunki ushbu tavsifga binoan tuzilgan algoritmlar to'liq BSR bilan ishlashni davom ettiradi. Ishlab chiquvchiga kelsak, u mikrosxemalardan bittagina tugmachasi ishlatilgan bo'lsa ham, u butun JTAG zanjirini (aniq yoki yashirin) ko'rib chiqishi kerak.

### 8.2.3. Dastur samaradorligini o'lchash

**Profilga tushirish** (profilanish) - bu "dasturiy ta'minot" ni bajarish uchun eng ko'p vaqt sarflaydigan qismlarni topish uchun butun dasturning ishlashini, shuningdek uning alohida qismlarini o'lchash.

**Profiler**- dastur optimizatorining asosiy vositasidir. Dastur kodi mashhur maqolda karvonning tezligini aniqlaydigan eng sekin tuyaga o'xshaydi, ya'ni dasturning ishlashi uning eng tor bo'limi bilan belgilanadi. Dasturchilarga o'z dasturlarini tahlil qilish va dasturning muhim bo'limlarini aniqlash uchun vositalar kerak.

Profilchilar sizga dasturning ba'zi qismlari qancha vaqt ishlashini, qancha ishlashini yoki qo'ng'iroq grafigini yaratishga yordam beradi. Odatda, ushbu ma'lumot dasturning eng ko'p ishlaydigan qismlarini aniqlash uchun ishlatiladi. Ushbu vaqt sarflaydigan qismlarni tezroq ishlash uchun optimallashtirish mumkin. Bu disk raskadrovka uchun umumiy uslub.

Profillashtirishning asosiy maqsadi dasturning barcha nuqtalarida xatti-harakatlarini o'rganishdir. Tafsilotlar darajasiga qarab "nuqta" ostida uni alohida mashina buyrug'i yoki yuqori darajadagi tilning butun

qurilishi (masalan, funktsiya, pastadir yoki manba matnning bitta qatori) sifatida tushunish mumkin.

Ko'pgina zamonaviy profillar quyidagi asosiy operatsiyalar to'plamini qo'llab-quvvatlaydilar :

- dasturdagi har bir punktning umumiy bajarilish vaqtini aniqlash;
- har bir dastur punktining aniq bajarilish vaqtini aniqlash;
- nizolar va jazolarning sabablarini va/yoki manbalarini aniqlash;
- dasturdagi ma'lum bir nuqtaning qo'ng'iroqlar sonini aniqlash;
- dasturning qamrov darajasini aniqlash.

### **Umumiy bajarilish vaqti**

Dasturning har bir punktini bajarishga sarflangan vaqt haqidagi ma'lumot uning eng "issiq" bo'limlarini aniqlashga imkon beradi. To'g'ri, bu erda bitta aniqlik kerak. To'g'ridan-to'g'ri o'lchash shuni ko'rsatadiki, barcha bajarilgan vaqtning kamida 99,99 foizi profilli dastur asosiy funktsiya ichida sarflanadi, shu bilan birga "issiq" deb ataladigan asosiy narsa asosiy emas, balki chaqiradigan funktsiyalardir. Dasturchilar orasida chalkashliklarni keltirib chiqarimaslik uchun, profillar odatda bola funktsiyalarini bajarishga sarflangan vaqtni dasturning har bir funktsiyasining umumiy bajarilish vaqtidan ajratadilar.

### **Muayyan bajarilish vaqti**

Agar dasturda ma'lum bir nuqtaning bajarilish vaqti doimiy bo'lmasa, lekin turli xil chegaralarda o'zgarib tursa (masalan, qayta ishlanadigan ma'lumot turiga qarab ), unda profil natijalarini talqin qilish noaniq bo'lib qoladi va natijaning o'zi ishonchsiz bo'ladi. Keyinchalik ishonchli tahlil quyidagilarni talab qiladi: bunday "suzuvchi" fikrlar dasturda haqiqatan ham mavjud yoki yo'qligini aniqlash va agar shunday bo'lsa, eng yaxshi, yomon va o'rtacha holatlarda ularni bajarilish vaqtini aniqlash.

### **Chaqiruvlar soni**

Biror nuqtaning haroratini nafaqat uni bajarilish vaqti bilan, balki qo'ng'iroqning chastotasi bilan ham baholash mumkin. Masalan, bizda ikkita "issiq nuqta" mavjud, deylik, protsessor bir vaqtni sarflaydi, lekin birinchisi yuz marta, ikkinchisi esa yuz ming marta deyiladi. Ikkinchisini kamida 1% ga optimallashtirish orqali biz katta samaraga erishamiz deb taxmin qilish oson, birinchisining bajarilish vaqtini yarmiga qisqartirish bilan biz dasturimizni atigi chorakka tezlashtiramiz. Shunday qilib, ko'p

hollarda chaqiriladigan funktsiyalar ko'p hollarda "inline" degan ma'noni anglatadi, ya'ni to'g'ridan-to'g'ri chaqiriladigan funktsiyalar tanasiga ularning kodini kiritish, bu vaqtni tejashga imkon beradi.

### **Qoplamani aniqlash**

**Qoplash** - bu uni yaratish paytida amalda bajarilgan dastur kodini foizi. Bunday ma'lumotlar, birinchi navbatda, sinov dasturchilari dasturning barcha kodlari to'liq sinovdan o'tganligini va unda "qorong'i" joylar yo'qligiga ishonch hosil qilishlari uchun kerak. Boshqa tomondan, dasturni optimallashtirishda uning qaysi qismlari profilli va qaysi biri bo'lmaganligini bilish juda muhimdir. Aks holda, dasturning tegishli filiallari hech qachon nazoratni olmaganligi sababli ko'pgina "issiq joylar" ni payqamaslik mumkin.

#### **8.2.4. Kod analizatori tahlili**

**Kod analizatori** (kod analizatori, kodni ko'rib chiquvchi dastur) - dasturning dastlabki kodidagi xatolarni (zaifliklar) aniqlashga yordam beradigan dastur (dasturiy ta'minot).

**Kod tahlili** - bu kodlarni ko'rib chiqishning yaqin qarindoshi.

**Kodlarni o'rganish** - bu rivojlanish bosqichida yo'l qo'yilgan xatolarni aniqlash, dasturning sifati va ishlab chiquvchilarning ko'nikmalarini yaxshilash maqsadida amalga oshiriladigan dastlabki kodni muntazam ravishda tekshirish.

Odatda, kodni tekshirish qatnashishni o'z ichiga oladi:

- kodni yozgan shaxs;
- ushbu kodni o'qiy oladigan va uning umumiy va o'ziga xos belgilarga qanchalik mos kelishini tushunadigan kishi (yoki odamlar).

Umumiy mezonlar kodlash standartidir. Maxsus mezonlar bu kod yozilgan talablar to'g'risidagi ma'lumotni anglatadi. Kodni tahlil qilish jarayoni sinovdan farq qiladi. Sinov paytida, dasturning haqiqiy harakatlaridagi nomuvofiqliklarni aniqlash uchun dastur ma'lum ma'lumotlarning to'plamida tekshiriladi. Shu bilan birga, spetsifikatsiya dasturning ishlashini faqat mumkin bo'lgan barcha kiritish ma'lumotlari to'plamida belgilashi mumkin. Shunday qilib, barcha xatolarni test orqali aniqlab bo'lmaydi. Buning uchun kodni tahlil qilish kerak, bu kabi xatolarni yoki aksincha, dastlabki koddagi zaifliklarni aniqlashga imkon beradi: bufer toshib ketishi, unifikatsiya qilinmagan xotira, null ko'rsatkich, xotira sizintisi, poyga holati va boshqalar. Kod bu dastur

haqida ma'lumotni uning dastlabki kodidan yoki ob'ekt kodidan olish jarayoni. Dastur kodi bu avtomatik ravishda bajariladigan kodga kompilyatsiya qilinishi mumkin bo'lgan kompyuter dasturining statik, matnli, inson tomonidan o'qiladigan, bajariladigan tavsifi.

Dastur kodini tahlil qilish uchta tarkibiy qismni o'z ichiga oladi:

**1.Parser**, dastur kodini tahlil qiladigan va tahlil natijalarini ichki vakillikning bir yoki bir nechta shakliga o'zgartiradigan tahlil qiluvchi. Ko'pincha tahlil qiluvchilar ( parsers) kompilyatorlarga asoslanadi.

**2.Ichki vakillik**, dasturning o'ziga xos tomoni mavjud va uni avtomatik tahlil qilish uchun mos shaklda taqdim etadi. Masalan, o'zgaruvchilar mos keladigan ma'lumotlar turlariga almashtiriladi. Ba'zi ichki vakilliklar to'g'ridan-to'g'ri tahlil qiluvchilar tomonidan yaratiladi, boshqalari oldingi tahlil natijalarini talab qiladi. Bunday vakilliklarning klassik misollari boshqaruv oqimining grafigi (boshqaruv oqimi grafigi, CFG), qo'ng'iroqlar daraxti (qo'ng'iroq grafigi), mavhum sintaksis daraxti (mavhum sintaksis daraxti, AST), statik yagona topshiriqni ( statik bitta topshiriq, SSA ) tashkil etadi.

**3.Ichki ish faoliyatini tahlil qilish.** Tahlil statik yoki dinamik bo'lishi mumkin. Statik kod tahlili dasturlarni amalda bajarmasdan amalga oshiriladi. Ushbu tahlil natijalari tegishli va barcha dasturlarning bajarilishi uchun bir xil. Statik tahlildan farqli o'laroq, dinamik kodni tahlil qilish real yoki virtual protsessorda dasturlarni bajarish orqali amalga oshiriladi. Ushbu tahlil natijalari aniqroq, ammo faqat ma'lum ma'lumotlar uchun kafolatlangan. Dinamik tahlil yordam dasturlari maxsus kutubxonalarni yuklashni yoki hatto dastur kodini qayta to'ldirishni talab qilishi mumkin.

Bundan tashqari, kod analizatorlarini quyidagicha tasniflash mumkin:

- Dastlabki qoidalar to'plamiga muvofiq dastlabki kodni tekshiradigan va tekshirish natijalari bo'yicha hisobotlarni tuzadigan avtomatlashtirilgan kontrollerlar.
- Dastur tuzilishini (arxitekturasi) vizual ravishda namoyish etadigan turli xil brauzerlar, uni yaxshiroq tushunishga yordam beradi.

## **8.2.5. Nosozliklarni tuzatish vositalariga misollar**

### **Keil dasturiy vositalari**

UVision to'plamiga kiritilgan Keil dasturiy vositalari simulyator va nosozliklarni tuzatish vositasi yordamida mikrokontroller tizimlarini

tuzatishga imkon beradi. Tuzatuvchi mikrokontrollerlarga JTAG interfeysi orqali (agar mavjud bo'lsa) va RS232 orqali ulanishi mumkin (bu holda dastur tuzatuvchisiga modul mikrokontrolderga o'rnatilishi kerak).

Simulyatorning asosiy xususiyati shundaki, ishlab chiqaruvchi disk raskadrovka jarayonini dasturlash, tashqi ta'sirlarni yaratish va maxsus skript tili yordamida natijalarni chiqarish qobiliyatiga ega. Aks holda, simulyator va disk raskadrovchining imkoniyatlari hozirgi kun uchun mutlaqo standartdir:

- dastlabki matnlarda disk raskadrovka;
- o'zgaruvchilarni ko'rish;
- ustunni ko'rish;
- ko'rish registrlari va xotiralar;
- o'tish nuqtalari.

Adapted Solutions tomonidan PONTiFLEX dasturidan foydalanib, uVision paketiga kiritilgan simulyator yoki nosozliklarni tuzatuvchini MATLAB / Simulink bilan birlashtirish mumkin.

### **Elektron sxema simulyatori**

National Instruments Electronics Workbench Group-dan Multisim va Labcenter Electronics-dan Proteus kabi elektron elektron simulyatorlari nafaqat raqamli va analog mikrosxemalarni disk raskadrovka qilish, balki asboblardan birgalikda dasturlarni tuzatishga qodir.

Dasturiy ta'minot to'plami turli xil mikrokontrollerlarning bir nechta modellarini o'z ichiga oladi, ular yordamida siz dasturlarni yuklab olishingiz va ularni doimiy, bosqichma-bosqich yoki to'xtash joyida bajarishingiz mumkin. Voltmetr, ampermetr, I2C kontroller, osiloskop, mantiqiy analizator va signal generatori kabi virtual vositalardan foydalanib, siz sinov signallarini yaratishingiz va kontaktlarning zanglashiga olib keladigan natijalarni ko'rishingiz mumkin.

### **PB eCos Nosozliklarni tuzatish vositalari**

RV eCos dasturni tuzatish uchun Redboot-dan foydalanadi. Redboot ( Red Hat o'rnatilgan Debug va Bootstrap uchun qisqartma ) - o'rnatilgan dasturiy ta'minot tizimida dasturiy ta'minot yoki dasturiy ta'minotni yuklab olish uchun eCos real vaqt operatsion tizimining apparat abstraksiyasi qatlamini (HAL) ishlatadigan ochiq manba dastur. Redboot GPL-ga mos eCos litsenziyasi bo'yicha taqdim etiladi.

Redboot maqsadli oʻrnatilgan tizimlarda dasturlarni yuklab olish va bajarish uchun ketma-ket aloqa yoki Ethernet ulanishi orqali oʻrnatilgan Linux va eCos dasturlarini, shuningdek maqsadli tizim parametrlarini boshqarish vositalarini keng qamrovli vositalarini taqdim etadi. Redboot shuningdek, bajariladigan rasmlarni yuklash uchun ishlatilishi mumkin boʻlgan flesh-modullar uchun oddiy fayl tizimini taqdim etadi. U mahsulotni ishlab chiqarishda (nosozliklarni tuzatish uchun), shuningdek yakuniy mahsulotda (tarmoq yoki flesh-xotiradan dasturlarni yuklab olish uchun) ishlatilishi mumkin. Redboot-ning xususiyatlari:

- 1.yuklash skriptlarini qoʻllab-quvvatlash;
- 2.redboot-ni boshqarish va sozlash uchun oddiy buyruq satri interfeysi, telnet protokoli orqali ketma-ket kanal yoki chekilgan ulanish orqali;
- 3.belgilangan oʻrnatilgan tizimda dasturlarni disk raskadrovka qilish uchun ketma-ket yoki chekilgan tarmoq interfeysi ( mahalliy tarmoq bilan cheklangan ) orqali asosiy kompyuterdagi nosozliklarni tuzatuvchiga ulanish uchun oʻrnatilgan GDB stublari ;
- 4.atribut konfiguratsiyasi - foydalanuvchini boshqarish va tizim vaqti va sanasi (agar ishlatilgan boʻlsa ), statik IP manzili va h.k. kabi jihatlarni oʻzgartirish imkoniyati ;
- 5.moslashuvchan va kengaytiriladigan, ayniqsa maqsad platformasiga moslashish uchun ;
- 6.BOOTP, DHCP va TFTP orqali oʻrnatish va yuklashni oʻz ichiga olgan tarmoq yuklanishini qoʻllab-quvvatlash ;
- 7.XModem va YModem protokollari yordamida ketma-ket interfeys orqali dasturlarni yuklab olishni qoʻllab-quvvatlash ;
- 8.startapning oʻzini sinab koʻrish;
- 9.Redboot eCos-ning bir tarmogʻi boʻlsa ham, uni har qanday oʻrnatilgan tizimlar va operatsion tizimlar uchun dasturiy taʼminotni yuklab olishni nazorat qilish va nazorat qilishning umumiy tizimi sifatida foydalanish mumkin.

### **Armulyator simulyatori**

**Armulyator** - bu ARM protsessor yadrolari uchun simulyator. Armulyator sizga xotirani, registrlarni, uzilishni boshqarish moslamasini, taymerni taqlid qilishga imkon beradi. Armulyatorning asosiy xususiyatlaridan biri ochiq arxitektura boʻlib, u sizga kerakli periferik modullarning zaruriy modellarini mustaqil ravishda amalga oshirishga imkon beradi.



### 8.3. O'rnatilgan tizimlarni sifatini baholash

#### 8.3.1. Asosiy ta'riflar

**Sinov** - bu tizim sifatini baholash jarayoni.

**Sinov** - tizimning belgilangan funksiyalarga muvofiqligini tekshirish. Uni loyiha, ishlab chiqarish va foydalanish bosqichida ajratish ( bajarish) mumkin. Tizimning funktsionalligi (texnik topshiriqning muhim qismi), ish sharoitlariga (mexanik, iqlimiy, elektromagnit ta'sirlar) muvofiqligi tekshiriladi.

**Defekt** (lat. Defectus) - nuqson, nuqson. Shuningdek, defekt atamasi quyidagicha aniqlanadi: nuqson - har bir alohida mahsulot belgilangan talablarga mos kelmasligi. Kamchiliklar doimiy yoki vaqtinchalik bo'lishi mumkin. **Muvaffaqiyatsizlik** -vaqtinchalik nuqson. **Kamchilik** - bu doimiy nuqson.

**Tasdiqlash** (verifikasiy) - belgilangan talablar bajarilganligi to'g'risida ob'ektiv dalillar taqdim etish asosida tasdiqlash. Tekshirish atamasi sinov, tadqiqot, sinov, ko'rib chiqish kabi tushunchalarni o'z ichiga oladi. Ba'zida tekshirish atamasi tekshirish bilan chalkashadi, ammo bu butunlay boshqacha narsalar. **Tasdiqlash** - bu sinov yoki simulyatsiya qilingan muhitda dasturni bajarish bilan xatolarni topishga urinish.

**Tasdiqlash (validasiy)** - bu aniq foydalanish yoki qo'llash uchun mo'ljallangan talablar bajarilganligi to'g'risida ob'ektiv dalillarni taqdim etish asosida tasdiqlash. Tasdiqlashning asosiy maqsadi ilgari olingan ob'ektiv dalillarga asoslanib, taklif qilingan echim asl vazifa uchun mosligini tasdiqlashdir. Tizimda hech qanday tadqiqot va sinov o'tkazilmaydi.

Tekshirishning maqsadi amalda muammoning echimi belgilangan talablarga javob berishini tekshirishdir. Dasturiy ta'minot nuqtai nazaridan tekshirish, bu haqiqiy muhitda dasturni bajarish orqali xatolarni qidirish.

#### 8.3.2. O'rnatilgan tizimlarini sifatini baholashni umumiy tamoyillari

Test quyidagi savollarga javob berishga imkon beradi:

- tizim etarlicha ishonchli va xavfsizmi;
- tizimning funktsionalligi mavjud xususiyatlarga mos keladimi;

- tizim real vaqt talablariga javob beradimi.

Sinov natijasi xatolar ro'yxati. Xatoni quyidagicha talqin qilish mumkin:

- to'plangan bilimlarga asoslanib, oldindan hisoblash va oldindan aytib bo'lmaydi;

- harakatlar, xatti-harakatlar, hukmlar, fikrlardagi xato.

Boshqacha qilib aytadigan bo'lsak, **xato** - bu tashqi ko'rinish, biz tizimimizda biron bir narsa noto'g'ri ekanligini sezamiz va tushunamiz. Texnologiyada nosozlikning namoyon bo'lishi odatda xato deb hisoblanadi. Shuni ta'kidlash kerakki, xato va nosozlik butunlay boshqa narsalar.

**Nosozlik** yoki nosozlik holati ob'ektning shunday holati bo'lib, bunda u normativ- texnik va (yoki) loyiha (loyiha) hujjatlarining kamida bittasiga mos kelmaydi. Nosozlik tizimimizning tarkibiga tegishli va xato faqat shakl. Shunga asoslanib, biz bir xil nosozlik ko'plab xatolarni keltirib chiqarishi mumkinligini tushunamiz ( sizning manbangiz matnidagi yagona sintaktik xato tufayli C kompilyatori qancha xato va ogohlantirishlarni keltirib chiqarishini eslang ). Tibbiy nuqtai nazardan, xato - bu alomat, nosozlik esa davolanishni talab qiladigan kasallikdir. Xatoning nomuvofiqligi va uni keltirib chiqargan nosozlik ko'plab muammolarni keltirib chiqaradi va sinov tizimini qurishda ehtiyotkorlik bilan murojaat qilishni talab qiladi. Barcha testlar nosozlik aniq tashxis qo'yishga imkon bermaydi. Muayyan sinov nosozlikni mahalliyalashtirishi mumkin bo'lgan aniqlik rezolyutsiya deb nomlanadi. Sinovni nafaqat mumkin, balki etarli darajada samarali qilish uchun tizimni sinov uchun mos keladigan tarzda ishlab chiqish kerak (DFT, sinov uchun loyiha).

Sinov ostidagi tizimga qo'yiladigan talablar:

- Ishlab chiqilayotgan tizim tez va samarali sinovdan o'tkazilishi uchun ishlab chiqilishi kerak. Bunday holda, biz dasturiy ta'minot kabi aniq bir narsa haqida gapirmaymiz, ya'ni butun tizimni bir butun sifatida.

- O'qish uchun vositalarni qanday tizimga joylashtirishingiz kerakligini va kirish ta'sirining simulyatorlarini tizimning qaysi joylarida bilishingiz kerak.

- Berilgan kirish ta'sirini sinab ko'rish uchun tizimning chiqishida nimani ko'rishimiz kerakligini oldindan bilishingiz kerak.

- Mumkin bo'lgan maksimal qamrovni ta'minlaydigan bunday test to'plamlarini qilish imkoniyati bo'lishi kerak.

## **Dasturiy ta'minotni sinovdan o'tkazishning xususiyatlari**

O'rnatilgan tizimlar, qoida tariqasida, tanqidiy dasturlar uchun ishlatiladi va real dunyo (boshqaruv ob'ekti) bilan bevosita aloqada bo'lgani uchun testlarga alohida tabellar qo'yiladi. Dasturiy ta'minotni sinovdan o'tkazishning xususiyatlari :

- Umumiy maqsadlar uchun mo'ljallangan dasturlar bilan taqqoslaganda, o'rnatilgan tizim dasturiy ta'minoti hisoblash jarayoni bitta emas, balki buyruqlarning faqat ketma-ket bajarilishini nazarda tutadigan odatiy Von Neumann mashinasidan farqli ravishda ko'p hisoblash modellariga asoslanganligi sababli ancha murakkabroq.

- Qurolli kuchlar umumiy maqsadlardagi tizimlarga qaraganda ancha kam hisoblash resurslariga ega.

- O'rnatilgan tizimlarning taqsimlanishi. O'rnatilgan tizimni sinovdan o'tkazishda eng keng tarqalgan variant - maqsad va instrumental tizimlarning bo'linishi. Keyinchalik murakkab holatlarda murakkab topologiya, heterojen tarmoq tugunlari va turli xil interfeyslarga ega bo'lgan maqsadli tizimlarning heterojen tarmog'i bilan ishlash mumkin.

### **8.3.3. loyiha jarayonlarini sinovdan o'tkazish**

Loyiha jarayoni sinovdan o'tkazilishi uchun, iloji boricha soddaligiga intilish kerak (KISS printsipli), tizimning arxitekturasini tushunish va uni loyihalashga harakat qilish kerak, shunda bu tizimga nafaqat sinovlarni kiritish mumkin, balki testlarni kiritish jarayoni ham sodda va uyg'undur. Shuni esda tutish kerakki, umumiy maqsadli tizimlardan farqli o'laroq, biz izolyatsiya qilingan ideal tizim bilan emas, balki boshqarish ob'ekti, apparat, loyiha, maqsadli va instrumental dasturlarni o'z ichiga olgan murakkab, kompozitsion tizim bilan ishlamaymiz.

Yuqorida aytilganlarga asoslanib, yirik dasturiy mahsulotlarni yaratishda ishlatiladigan klassik sinov vositalaridan faqat qisman foydalanish mumkinligi aniq bo'ladi. Bundan tashqari, bizning holatlarimizda bunday vositalar barcha muammolarni hal qilmaydi. Masalan, o'rnatilgan tizim uchun dasturni ishlab chiqaruvchiga quyidagilar aytilishi mumkin: "sizning dasturingiz 57 daraja haroratda ishlashni to'xtatadi" yoki "tebranish paydo bo'lganda dastur ishlamay qoladi".

An'anaviy dasturlash uchun odatiy bo'lmagan sinov vositalari orasida quyidagilarni ta'kidlash mumkin:

- osiloskop va mantiq analizatorlari;
- JTAG;
- apparat va dasturiy ta'minotlarni nosozliklarni tuzatish stendlari (testbench);
  - asbob-uskunalar;
  - iqlim qurilmalari;
  - elektromagnit shovqinlarni keltirib chiqaradigan qurilmalar;
  - tebranish stendlari.

Sinovning qiyinligi nimada? Afsuski, testlar instrumental xato deb ataladigan tizimni kuzatishda qo'shimcha shovqinlarni keltirib chiqaradi. Bundan tashqari, maqsadli tizimga kiritilgan sinov tizimi, odatda, juda qimmat resurslarga ega, ular bilasizki, doimo etishmayapti. Shuning uchun, siz doimiy ravishda sinovning qulayligi, testlarning maqsadli tizim va mavjud resurslarning ishlashiga ta'siri o'rtasidagi o'zaro kelishuvga murojaat qilishingiz kerak. Shuni tushunish kerakki, sinov uchun tayyor bo'lgan ichki tizimni yaratish murakkab, ko'p qirrali vazifa bo'lib, uni hal qilishga jiddiy yondashish kerak.

Sinov tizimini yaratish uchun zarur bo'lgan minimal miqdorni ko'rib chiqing. **Birinchidan**, sizga yozuvchiga o'xshash yozuv tizimi kerak. Haqiqat shundaki, ko'plab sinovlar bir necha kun davom etishi mumkin va bunday sharoitda xatolarni qo'lda tuzatishning imkoni yo'q. U bir cholg'u foydalanish qulay bir ma'lumotlar bazasi saqlash test natijalari bilan kompyuter, ro'yxatdan o'tishingiz tizimi sifatida. **Ikkinchidan**, o'rnatilgan tizimlarni sinash uchun sizga sinov dastgohi kerak. Sinov dastgohi simlarning oddiy to'plami, almashtirish tugmachalari, lampochkalar va o'tish moslamalari bo'lishi mumkin yoki boshqarish ob'ektining turli xil simulyatorlari bilan jihozlangan murakkab apparat-dasturiy ta'minot majmui bo'lishi mumkin va agar u jismonan mumkin bo'lsa, boshqarish ob'ektining o'zi ham bo'lishi mumkin. **Uchinchidan**, sinovdan o'tkazish uchun ishlab chiqilayotgan qurilma uchun noqulay muhitni yaratish imkoniyati bo'lishi kerak. Buning uchun ishlab chiqilgan tizimning normal sharoitda, issiqda, sovuqda, namlik, chang va tebranish ta'siri ostida qanday ishlashini tekshirish kerak. Oddiy holatda, siz elektr isitgichi, maishiy muzlatgich va shovqin generatoridan foydalanishingiz mumkin, murakkab vaziyatda esa bunday sinovlarni o'tkazadigan ixtisoslashgan kompaniyalardan sanab o'tilgan uskunalarni sotib olishingiz yoki ijaraga olishingiz kerak bo'ladi.

## **Interfaol (dialog) rejimida testlash**

Interfaol test sinov hajmi kichik bo'lgan yoki tizim hali to'liq tuzatilmagan holatlarda qo'llaniladi. Sinov tizimida foydalanuvchi interfeysi va test ishlarini tashkil qilish uchun vositalar to'plami bo'lishi shart.

Ushbu yondashuvning kamchiliklari quyidagilardir:

- test natijalariga ta'sir qiluvchi inson omilining mavjudligi;
- nisbatan uzoq sinov muddati;
- yuqori murakkablik va yuqori malakali sinovchilarga ehtiyoj.

Yondashuvning afzalliklari quyidagilarni o'z ichiga oladi:

- amalga oshirish qulayligi (KISS printsipiga amal qiling);
- sotishning arzonligi;
- yechim moslashuvchan.

Ko'p mikrokontrolörler UART-ni o'z ichiga olganligi sababli, o'rnatilgan tizim loyihaerlari orasida eng keng tarqalgan yondashuv - sinov tizimini tizimli kanal va terminal emulyatori orqali amalga oshirish.

Sinov uchun mo'ljallangan tizimda sizda quyidagilar bo'lishi kerak:

- RS-232 instrumental kompyuter bilan aloqa qilish uchun ketma-ket kanal ;
- anketa asosida ishlaydigan oddiy kanallarning drayversi;
- sinovlari va ularning tashkil chaqirish uchun oddiy matn menyuda tizimini konfiguratsiyani;
- RS-232-ket kanalli instrumental kompyuter;
- modem kabeli RS-232;
- terminal emulyatori.

O'rnatilgan amaliy dasturlar uchun (masalan, o'rnatilgan Linux) RTOS yoki o'rnatilgan OS asosida yaratilgan nisbatan murakkab tizimlarda nosozliklarni tuzatish va sinovdan o'tkazish uchun TCP / IP protokollari yordamida ketma-ket kanal va Ethernet orqali ulanadigan konsol ishlatiladi. IP va telnet yoki ssh protokollari.

## **Avtomatlashtirilgan sinov**

Avtomatik (inson ishtirokisiz) va avtomatlashtirilgan (inson ishtirokida) sinovdan o'tkazish, agar etarlicha katta miqdordagi qurilmalar bilan ishlash zarurati tug'ilsa yoki inson omilining ta'siri sinov jarayoniga juda salbiy ta'sir ko'rsatsa, iqtisodiy jihatdan maqsadga muvofiqdir.

Avtomatlashtirilgan sinov tizimining murakkabligini sinov ostidagi tizimning murakkabligi bilan taqqoslash mumkin. Avtomatik sinov tizimida test ta'sirini yaratadigan vositalar bilan jihozlangan sinov stendi bo'lishi kerak. Bundan tashqari, sinov tizimida voqealarni qayd qiluvchi jurnal mavjud bo'lishi kerak. Agar sinov ostidagi tizim taqsimlansa, avtomatik sinov jarayonining murakkabligi ko'p marta ortadi.

#### **8.4. O'rnatilgan tizimlarda quvvat sarfini kamaytirish usullari**

Mikrokontrollarning quvvat sarfini kamaytirish ichki tizimlarni loyihalash jarayonida juda muhim ahamiyatga ega. Energiya sarfini kamaytirish avtonom elektr manbalaridan (batareyalar, akkumulyatorlardan) ishlaydigan kontrollerlarning ishlash muddatini ko'paytirishga imkon beradi, passiv sovutish tizimlarining harorat rejimini yaxshilaydi (yopiq joyda (bortli elektronika)), elektr ta'minoti pallasini soddalashtiradi.

Qabul qilinadigan vazifaga muvofiq energiya sarfini amaytirishning bir necha usullari mavjud, ular birgalikda va alohida ishlatilishi mumkin:

- bortida tashqi qurilmalarning off quvvat;
- mikrokontrollarning soat chastotasini pasaytirish;
- uyqu rejimi;
- hozirda foydalanilmayotgan mikrokontroller bloklarini o'chirib qo'yish ;
- kirish /chiqish portlarini minimal quvvat iste'moli holatiga o'tkazish.

Shuni ta'kidlash kerakki, quyidagi tavsiyalardan foydalanish etarli emas. Ular umumiy xarakterga ega bo'lib, faqat energiya sarfini kamaytirishning asosiy printsiplarini namoyish etadi. Har holda, u batafsil sizning diagrammada o'rganish uchun zarur bo'lgan qurilma, shuningdek, foydalanuvchi qo'llanmasini va qo'shimcha dastur eslatmalarni mikro va boshqa ishlab chiqaruvchi tomonidan taqdim etilgan, elektron komponentlar. Bunga qo'shimcha ravishda, kerakli effektga erishish uchun multimetr yordamida taxtaning quvvat sarfini doimiy ravishda o'lchash va yozib qo'yishingiz kerakligini yodda tutishingiz kerak.

## Tashqi qurilmalarni o'chirish

Barcha kontaktlarning zanglashiga olib keladigan quvvatni kamaytirish uchun mikrokontrollerdan tashqarida bo'lgan bir qator mikroshemalar uchun quvvatni o'chirish qobiliyati bilan kontaktlarning zanglashini ta'minlash mumkin. Ushbu kontaktlarning zanglashiga olib borish mikrokontrolörning chiqishi orqali amalga oshiriladi.

## Mikrokontrollerni taktik chastotasini pasaytirish

Protessorning soat tezligini kamaytirish quvvat sarfini kamaytirishning eng oson usullaridan biridir. Mikrokontrolör turiga qarab, quvvat sarfini birdan o'n baravarga kamaytirish mumkin. Quvvat sarfining pasayishi mikrokontrolordagi tranzistor kaskadlari tomonidan ishlab chiqarilgan kommutatsiya operatsiyalari sonining kamayishi bilan izohlanadi. Ba'zi davrlarning quvvat sarfi chastotaga ko'proq bog'liq, boshqalari kamroq. Shuning uchun energiya sarfini chastotaga bog'liqlik tabiati mikrokontrolör turiga, unga kiritilgan birliklar soniga, kirish-chiqish portlarining parametrlariga va tashqi yuklarga bog'liq.

8.1-jadval. Ikki xil soat chastotasi uchun taxminiy quvvat sarfi qiymatlari

MIKROKONTROLLER	f1/P	f2/P
Motorola 68HC705KJ1	1,0 MГц / 4,0 мВт	2,1 MГц / 4,6 мВт
Microchip PIC 18	32 кГц / 3 мВт	40 MГц / 60 мВт
Philips LPC 9xx (MCS51)	12 MГц / 25 мВт	18 MГц / 40 мВт
Philips LPC 2292 (ARM7)	10 MГц / 13 мВт	60 MГц / 90 мВт

Ko'pgina hollarda, bu yondashuv qo'llanilmaydi, chunki soat chastotasi pasayganda mikrokontrolatorning ishlashi mutanosib ravishda pasayadi. Uyqu rejimidan foydalanish afzalroqdir. Soat chastotasi tegishli kvarts rezonatorlarini o'rnatish orqali boshqariladi. Bundan tashqari, zamonaviy mikrokontrolörlarning aksariyat qismida chastota taqsimlagichi va chastota multiplikatori davri amalga oshiriladi.

## Uyqu rejimi

Kutish rejimi mikrokontrollerning quvvat sarfini kamaytirishning eng kuchli usullaridan biridir. Mikrokontrolör turiga va uxlash rejimiga qarab, siz quvvat sarfini bir necha bor kattalikdagi bir necha buyurtmaga

qadar olishingiz mumkin. Kutish rejimida mikrokontrollerning soati to'xtaydi, dasturning bajarilishi to'xtaydi, quvvat iste'moli minimal bo'ladi. Ushbu rejimdan chiqish quyidagi hollarda mumkin:

- o'chirganda va quvvatni yoqishda;
- qo'riqchi (watchdog) taymeri tomonidan qo'shilgan RESET signalida;
- to'sqinliksiz uzilishlar yuzaga kelganda.

Mikrokontrolör turiga qarab, uyqu rejimi soat, registrlar, xotira va atrof-muhit holatiga qanday ta'sir qilishi mumkin. Hozirgi vaqtda uxlash, kutish, to'xtash, kutish va hokazo atamalar ingliz adabiyotida turli xil uyqu rejimiga murojaat qilish uchun ishlatiladi.

Odatda, o'rnatilgan tizimning uyqu rejimi quyidagicha:

- biron bir tashqi voqea bo'lmaganda, tizim uxlaydi, energiya iste'moli minimal bo'ladi;
- voqea sodir bo'lganda, mikrokontroller uyg'onadi,
- hodisani qayta ishlash uchun kerakli harakatlarni amalga oshiradi va yana uxlab qoladi.

8.2-jadval.

Oddiy rejim va uyqu rejimlaridan biri uchun quvvat sarfining taxminiy qiymatlari

<b>Mikrokontroller</b>	<b>O'rtacha rejim</b>	<b>Rejim SNA</b>
Motorola 68HC705KJ1	4,0 mBT	1,0 mBT
Atmel mega 128	33 mBT	75 mkBT
Microchip PIC	20 mBT (20 MΓII)	10 mkBT
TI MSP 430	3 mBT	15 mkBT
STMicroelectronics ARM7	150 mBT (48 MΓII)	33 mkBT

Uyqudan foydalanganda davriy va aperiodik rejimda ishlash mumkin. Vaqti-vaqti bilan mikrokontroller taymerning uzilishi bilan muntazam ravishda uxlaydi. Yilda aperiodic rejimida, uyqu tiklash bir, alohida chiqish kiritish portiga bo'yicha signal paydo ustiga, masalan, tashqi signal kelishi, ustiga sodir UART yoki men 2 C kiritish. Yuqorida aytib o'tilganidek, energiya sarfini kamaytirishning turli usullari birlashtirilishi mumkin. Ko'pgina hollarda, uyqu rejimidan foydalanganda protsessor soatining tezligini tushirishning ahamiyati yo'q. Yuqori tezlik bilan, vazifa tezroq amalga oshiriladi va mikrokontroler ilgari uxlab qoladi, bu esa ko'proq energiya tejashga olib keladi. Shuni ta'kidlash kerakki, juda ko'p turli xil mikrokontrolerlar mavjud va ularning barchasi boshqacha tarzda amalga oshiriladi.



Shuning uchun, har bir holda, tajriba o'tkazish va energiya sarfini kamaytirishning turli usullarining maqbul nisbatlarini izlash kerak.

### **Mikrokontroller bloklarini o'chirish**

Hamma biladi, agar siz kvartirangizning xonalarida qo'shimcha chiroqlarni o'chirsangiz, bu energiya sarfini kamaytiradi. Elektr energiyasi uchun ko'p pul to'lamaslik uchun odamlar foydalanilmayotgan elektr jihozlarini o'chirishga moyildirlar. Shunga o'xshash mexanizm zamonaviy mikrokontrolörlarning aksariyatida mavjud. Mikrokontrolör juda ko'p birliklardan iborat bo'lib, ularning ko'pini yoqish va o'chirish mumkin ( UART, CAN, I 2 C, DAC, ADC, taymerlar va boshqalar). Agar siz foydalanilmagan bloklarni o'chirsangiz, quvvat sarfi kamayadi.

### **I / O portlarini sozlash**

Shuni esda tutish kerakki, nafaqat quvvat sarfi, balki kontaktlarning zanglashiga olib kirish portlari konfiguratsiyasiga ham bog'liq. RESET signalidan so'ng barcha kirish / chiqish portlarining konfiguratsiyasi tiklanadi. Birinchi narsa, portlarning dastlabki holatini qurilmangizning sxemasiga va quyida keltirilgan quvvat sarfini kamaytirish bo'yicha tavsiyalarga muvofiq to'g'ri dasturlashdir.

### **Raqamli kiritish portlari**

Agar kirish voltaji nolga yoki ta'minot kuchlanishiga yaqin bo'lsa, mikrokontrolörning raqamli kirishlarida minimal iste'mol bo'ladi. Raqamli kirishdagi kuchlanish doimiy ravishda o'zgarib tursa yoki nol kuchlanish nuqtalari o'rtasida bo'lsa, agar kontaktlarning zanglashiga imkon beradigan bo'lsa, portni chiqish joyiga o'tkazish yaxshiroqdir. Shuni esda tutish kerakki, agar chiqishda ishlaydigan ikkita chiqish bir-biriga ulangan bo'lsa, bu portlarning ishdan chiqishiga olib kelishi mumkin. Faqat ochiq kollektor chiqishlarini bir-biriga ulash mumkin.

### **Raqamli chiqish portlari**

Raqamli chiqish portining iste'moli unga bog'liq bo'lgan yuk tufayli kelib chiqadi. Shuning uchun, agar bu kontaktlarning zanglashiga olib keladigan loyihasi bilan imkoni bo'lsa, quvvat sarfini minimallashtirish uchun, port orqali chiqadigan oqim minimal bo'lishi uchun bunday signalni port chiqishiga qo'llash kerak.

## **Analog kirish porti**

Analog kirish portiyuqori kirish empedansiga ega, shuning uchun analog kirishlarning joriy iste'moli juda past. Minimal energiya sarfi kirish voltaji nol va besleme zo'riqishining o'rtasida bo'lsa, kuzatiladi. Ba'zi hollarda, quvvat sarfini kamaytirish uchun kirish portlarini analog ish rejimiga o'tkazish foydali bo'ladi.

### **Nazorat savollari:**

- 1.Xost va nishon o'rtasidagi farq nima?
- 2.Yuqori darajadagi qanday toifalarga odatda rivojlanish vositalari kiradi?
- 3.IDE host-tizim bilan o'zaro aloqada bo'lish uchun ishlatiladi.
- 4.Simulator turlari?
- 5.Kod analizatorlarini vazifalari.
- 6.O'rnatilgan tizimlarda va boshqa kompyuter tizimlarida kompilyatsiya ehtiyojlaridan qanday xususiyatlar farq qiladi?
- 7.Ob'ekt fayli nima?
- 8.Yuklash vositasi va bog'lovchi o'rtasidagi farq nima?
- 9.Elektron emulyator nima?
- 10.Tarjimomni talab qiladigan real dunyoning qaysi uchta tili.
- 11.Nosozliklarni tuzatish nima?
- 12.Nosozliklarni tuzatish vositalarining asosiy turlari qanday?
- 13.Nosozliklarni tuzatish vositalarining har bir turidagi hayotiy to'rtta misollarning ro'yxati va tavsifi.
- 14.Nosozliklarni tuzatish uchun eng arzon besh usul qanday?
- 15.Yuklash kodi nima?
- 16.Nosozliklarni tuzatish va sinovdan qanday farq bor?
- 17.Sinov usullari mos bo'lgan to'rtta modelni sanab bering va aniqlang.
- 18.Ushbu modellarning har birida qanday besh xil sinov bo'lishi mumkin?
- 19.O'rnatilgan tizimlarda quvvat sarfini minimallashtirish usullari.
- 20.Tekshirishni sinab ko'rish va qobiliyatsiz sinov o'rtasidagi farq nima?
- 21.Qaysi tillarga ajratish mumkin, bo'lgan mezonlarni sanab bering?
- 22.Spesifikatsiya va dasturlash tillari o'rtasidagi farq nima?
- 23.Dasturlash uslublarining tasnifi.

24.Dasturlash tillarining asosiy xususiyatlariga nimalarni kiritish mumkin?

25.Oʻrnatilgan tizimni loyihalashda ishlatiladigan tillarning tahlilini oʻtkazing.

26.Oʻrnatilgan tizimda nosozliklarni tuzatish vositalarini sanab bering va tavsiflang.

27.Dasturiy taʼminotni sinashning xususiyatlari qanday ?

28.Kod analizatorining asosiy funktsiyalarini sanab bering.

29.Energiya sarfini kamaytirishning asosiy usullarini sanab bering?

## ADABIYOTLAR RO‘YXATI

1. Мирзиёев Ш.М. Буюк келажакимизни мард ва олижаноб халкимиз билан бирга қураимиз. Тошкент. «Ўзбекистон», НМИУ, 2017. – 488 б.

2. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Тошкент. «Ўзбекистон», НМИУ, 2017. – 48 б. 3. Мирзиёев Ш.М. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Тошкент. «Ўзбекистон», НМИУ, 2016. – 56 б.

4. Embedded Systems Architecture A Comprehensive Guide for Engineers and Programmers By Tammy Noergaard 2012.

5. Real-Time Concepts for Embedded Systems by Qing Li and Carolyn Yao ISBN:1578201241.

6. Dr. K.V.K.K. Prasad “Embedded/Real-time systems: Concepts, design and Programming”.

7. Wolf, Wayne Hendrix. Computers as components: principles of embedded computing system design /byWayneWolf – 2nd ed. ELSEVIER 2008.

8. М.М.Мусаев. Компьютер тизимлари ва тармоқлари. Т.: «Алоқачи» 2013б 394 б.

9. Laplante, Phillip A. Real-time systems design and analysis : tools for the practitioner / Phillip A. Laplante, Seppo J. Ovaska.—4th ed. Willey 2012

10. Ключев А.О., Кустарев П.В., Ковязина Д.Р., Петров Е.В. Программное обеспечение встроенных вычислительных систем. – СПб.: СПбГУ ИТМО, 2009.

11. Wolf, Wayne Hendrix. Computers as components: principles of embedded computing system design /byWayneWolf – 2nd ed. ELSEVIER 2008

12.Raj Kamal: Embedded Systems- Architecture, Programming and Design, TMH, New Delhi.

13.Microprocessor, theory and applications by A.V.Deshmukh, TMH Publication. 14.Advanced microprocessor and peripherals (architecture, programming and interfacing) by A.K.Roy & K.M.Bhurchandi, TMH Publication.

15.N.Senthil Kumar, M.Saravanan, S.Jeevananthan, „Microprocessors and Microcontrollers“, Oxford,2013.

16.Krishna Kant, “Microprocessor and Microcontrollers”, Eastern Company Edition, Prentice Hall of India, New Delhi, 2007.

17. R.S. Gaonkar, „Microprocessor Architecture Programming and Application“, with ii8085AA, Wiley Eastern Ltd., New Delhi, 2013.
18. Heath, S. Embedded systems design. EDN series for design engineers  
/ S. Heath. – 2nd ed. – Oxford: Elsevier Science, 2003. – 430 p. – ISBN 07506 5546 1
- 19 Hennessy, J.L., Patterson, D.A., Goldberg, D. Computer architecture: aquantitative approach / J.L. Hennessy, D.A. Patterson, D. Goldberg. –  
3rd ed. – San Francisco: Morgan Kaufmann, 2003. – 883 p. – ISBN 1-55860-596-7
- 20.Koenig, D. Computer-Integrated Manufacturing: Theory and Practice / D. Koenig. – Hemisphere: Taylor & Francis, 1990. – 248 p. – ISBN 0891168745
- 21.Koren, Y. Computer Control of Manufacturing Systems / Y. Koren. –  
McGraw-Hill Education (ISE Editions), 1984. – 304 p. – ISBN 0-07-035341-7
- 22.Waldner, J.-B. CIM: Principles of Computer-Integrated Manufacturing / Jean-Baptiste Waldner. – John Wiley & Sons Inc., 1992. – 206 p. –ISBN 047193450X
- 23.Wolf, W.H. Computers as Components: Principles of Embedded Computing Systems Design / W.H. Wolf. – San Francisco: Morgan Kaufmann, 2005. – 656 p. – ISBN 978-0-12-369459-1
- 24.Zurawski, R. Embedded systems handbook. Industrial information technology series / R. Zurawski. – Boca Raton: CRC Press, 2006. – 1160 p. – ISBN 0-8493-2824-1
- 25.Гук, М.Ю. Аппаратные средства IBM PC. Энциклопедия / М.Ю. Гук. – 3-е изд. – СПб.: Питер, 2006. – 1072 с.: ил. – ISBN 5-469-01182-8
- 26.Непейвода, Н.Н. Стили и методы программирования: Курс лекций: Учеб. пособие / Н.Н. Непейвода. – М.: Интернет-Университет информационных технологий (ИНТУИТ), 2005. – 320 с. – ISBN 5-95560-023-0

27. Себеста, Роберт У. Основные концепции языков программирования / Роберт У. Себеста; пер. с англ. – 5-е изд. – М.: Вильямс, 2001.

– 672 с. – ISBN 5-8459-0192-8 (рус.), ISBN 0-201-75295-6 (англ.)

28. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум. – 5-е

изд. – СПб.: Питер, 2007. – 844 с: ил. – ISBN 5-469-01274-3.

29. Embedded Microcomputer Systems, Valvano, p. 3.; Embedded Systems Building Blocks, Labrosse, p. 61.

30. Based on the software architectural brainchildren of the Software Engineering Institute (SEI); Software Architecture in Practice, Bass, Clements, and Kazman, 2003 (go to <http://www.sei.cmu.edu/>).

31. Advanced Telecommunications Computing Architecture. <http://www.picmg.org/>

32. Anthony J. Massa Embedded Software Development with eCosT Prentice Hall PTR November 25, 2002 ISBN : 0-13-035473-2.

33. Hatley D.J., Pirbhai I.A. Strategies for Real-Time System Specification. - N.Y. Dorset House Publishing, 1988.

34. Edwards S., Lavagno L., Lee E.A., Sangiovanni-Vincentelli A. Design of embedded systems: Formal models, validation, and synthesis. // Proceedings of the IEEE. March 1997

35. Lee E.A. Embedded Software. // Technical Memorandum UCB/ERL

M01/26, University of California. Berkeley. November 1, 2001

36. Maciel P., Barros E., Rosenstiel W. A Petri Net Model for Hardware/Software Codesign. // In Design Automation for Embedded Systems. Vol. 4. October 1999. P. 243–310

37. Patterson D.A., Hennessy J.L. Computer Organization and Design: The Hardware /software Interface. // Morgan Kaufmann. 2005.

38. Intelligent transport systems. <http://www.its.dot.gov/>

39. Baron, R.J. и Higbie L., Computer Architecture, Addison-Wesley, 1992. Kane, G. and Heinrich, J., MIPS RISC Architecture, Prentice-Hall, 1992.

40. Barr, M., Massa, A.N. Programming embedded systems: with C and GNU development tools [Text] / M. Barr, A.N Massa. – 2nd ed. – Sebastopol: O'Reilly Media Inc., 2006. – 301 p. – ISBN 0596009836

## Mundarija

<b>1-BOB. O‘rnatilgan tizimlar, ta’rifi, xususiyatlari va tasnifi...</b>	<b>4</b>
<b>2-BOB. O‘rnatilgan tizimlarni loyixalashda tizimli yondashuv .....</b>	<b>32</b>
<b>3-BOB. O‘rnatilgan tizimlarning arxitekturaviy loyihash .....</b>	<b>66</b>
<b>4-BOB. O‘rnatilgan tizimlarning tarmoq interfeyslari .....</b>	<b>116</b>
<b>5-BOB. O‘rnatilgan tizimlarni dasturi taminoti.....</b>	<b>135</b>
<b>6-BOB. O‘rnatilgan operatsion tizimlar .....</b>	<b>148</b>
<b>7-BOB. Oraliq va amaliy dasturiy ta’minoti.....</b>	<b>187</b>
<b>8-BOB. O‘rnatilgan tizimlarni loyihalash vositalari.....</b>	<b>206</b>

**R.P. Abduraxmanov**

# **O‘RNATILGAN TIZIMLAR**

**(O‘quv qo‘llanma)**

**Toshkent – «NIHOL PRINT» OK – 2021**

Muharrir: Q.Matqurbonov  
Tex. muharrir: A.Tog‘ayev  
Musavvir: B.Esanov  
Musahhiha: O.Muxammadiyeva  
Kompyuterda  
sahifalovchi: G.Tog‘ayeva

9323



№ 7439-765f-47f1-7ea1-a683-4648-1314.  
Bosishga ruxsat etildi: . Bichimi 60x841 /16.  
Shartli bosma tabog‘i 15,75. Nashr bosma tabog‘i 15,5.  
Adadi 100. Buyurtma № .



Library.tuit.uz

«Nihol print» Ok da chop etildi.  
Toshkent sh., M. Ashrafiy ko'chasi, 99/101.