

СӘТБАЕВ
УНИВЕРСИТЕТІ



SATBAYEV
UNIVERSITY

**Командалар жиынының архитектурасын тарату.
Өтулер. Ерекшеліктері және үзулер.
Архитектура деңгейлері. Микрокод және
командалар жиынының архитектурасының
байланыс. Шиналы байланыс құрылымдары.
Командалар конвейірін ұйымдастыру. Кэш.
Суперскалярлық.**

Оқытушы: Жекамбаева М.Н.
«Программалық инженерия»
кафедрасының профессоры
m.zhekambayeva@satbayev.university

Командалар жиынының архитектурасын тарату

Процессор архитектурасы және командалар жиыны аппаратура және программалақ қамтамалар арасындағы шекара болып табылады және ол тек компиляторды өңдегіштер және программистке ғана көрінеді, жоғары деңгейде жұмыс жасайтын программистердің көрмеуі мүмкін.

Командалар жиынының архитектурасының екі негізгілері болып CISC (командалар саны көптер) және RISC (командалар саны аздар). RISC- 200, ал CISC 300 астам командалар бар.

Командалар жиынының архитектурасын тарату

CISC (Complete Instruction Set Computer) Intel компаниясымен x86 сериялы компьютерлер жиыны жатады.

CISC-процессорларының сипаттамалары:

- Салыстырмалы жалпы міндетті регистрлердің саны аз
- Машиналық командалары көп, көп тактілер арқылы орындалады және жоғары деңгейдегі тілдерде қолданылатын операторларға пара пар болып келеді
- Адресациялау әдістерінің саны көп (6 - 12 әдіс)
- Әртүрлі разрядтылықтағы командалар форматтарының саны көп (бір байт, екі байт, үш байт және т.с.с командалар ұзындығы)
- Екі адресті командалар форматын пайдаланады
- Регистр-жады түріндегі өңдеу командаларын пайдаланады

Командалар жиынының архитектурасын тарату

RISC-процессорларының сипаттамалары:

- Фиксирленген командалар ұзындығы
- регистр-регистр архитектурасы
- қарапайым адресациялау әдістері (2 - 3 әдіс)
- қарапайым (интерпретацияны керек етпейтін) командалар
- регистрлік файлдар саны көп
- үшоперандты (бұзылмайтын) командалар форматы

RISC-жүйесін негізгі ұсынатын компаниялары: Hewlett-Packard (PA-RISC), Sun Microsystems Computers (SPARC), Digital Equipment (Alpha), Silicon Graphics, IBM және Motorola (PowerPC).

Командалар жиынының архитектурасын тарату

Кең нұсқау сөзі (VLIW, Very Long Instruction Word) бар процессорлар - бұл процессордың нұсқау жүйесі архитектурасының бір түрі, оның нұсқау сөздері әрбір атқарушы құрылғы үшін операцияны анықтайтын бірнеше буындардың жиыны болып табылады. Атаудың шығу тегі мұндай нұсқау сөзінің ұзындығы 512 бит немесе одан да көп болуы мүмкін екендігімен анықталады.

Командалар жиынының архитектурасын тарату

Командалардың ең аз жинағы, анг. MISC, Minimal Instruction Set Computer – нұсқаулардың өте аз саны (бірнеше ондаған) және олардың барлығы дерлік нөлдік операндты, жинақталған 5 - 8 бит. Аралық деректер әдетте стекте сақталады.

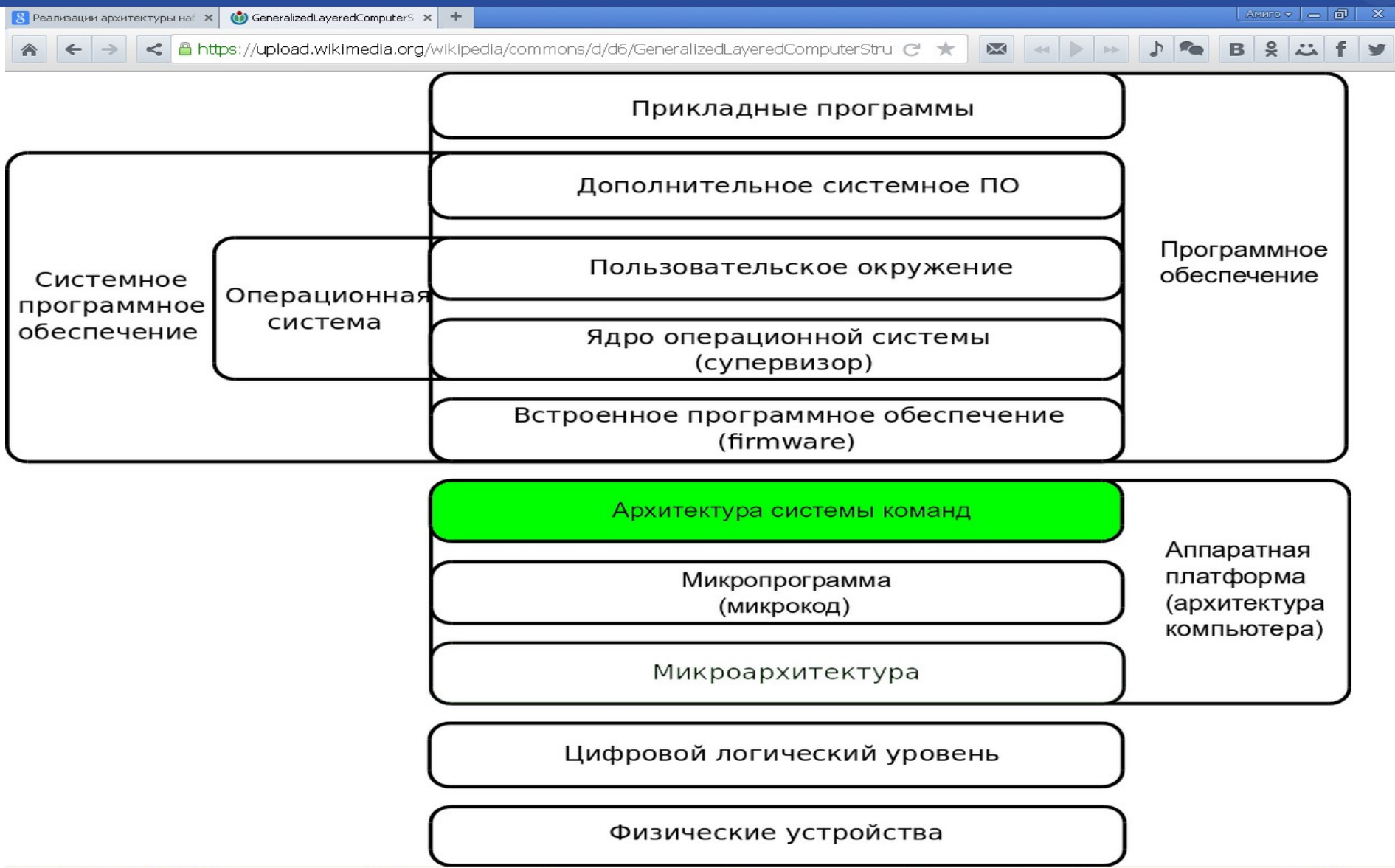
Командалар жиынының архитектурасын тарату

Өзгертілетін командалар жинағы, анг. WISC, Writable Instruction Set Computer — шешілетін тапсырма үшін нұсқаулар жинағын өзгерту арқылы өзіңізді қайта программалауға мүмкіндік беретін архитектура.

Командалар жиынының архитектурасы

Командалар жиынының архитектурасы — бұл компьютер МП ядросының программаланатын бөлігін анықтайтын архитектурасының бөлігі. Бұл деңгейде нақты типтегі МП таратылғандар анықталады:

Командалар жиынының архитектурасы



Программалық қамтама

Пайдаланушы, қолданбалы программалық қамтама, операциялық жүйе және аппараттық қамтамалар (құрал-жабдықтар) арасындағы қарым-қатынас.

Програ́ммалық қамтамалар — программа, процедура, тәртіптер және сәйкес құжаттамалар (*ISO/IEC 2382-1:1993*). Сондай-ақ компьютерлік программалар, компьютерді басқаратын программа немесе көптеген программалар, әйтпесе программалар жиыны.

ПҚ классификациясы

Программаларды орындау әдісі бойынша

- интерпретацияланатын
- компиляцияланатын

Переносимость ұтқырлық деңгейі бойынша

- платформаға тәуелді
- кроссплатформалы

Қодану және тарату әдісі бойынша

- бос емес (жабық)
- ашық
- бос

Программалардың тағайындалуы бойынша:

- жүйелік
- қолданбалы
- аспаптық

Программа түрлері бойынша:

компонентті: өзбетінше немесе құрама бөліктерде қолданыллатын аяқталған функцияны орындайтын бір бүтін ретінде қарастырылатын программа;

кешенді: өзбетінше немесе басқа құрама бөлік құрамаларында өзара байланысты функцияларды орындайтын бір немесе бірнеше компоненттерден немесе кешендерден тұратын программа

Жүйелік программалық қамтама

Компьютерлік жүйенің компоненттерін басқаруды қаматамсыз етеді, мысалы МП, жедел жады, енгізу-шығару құрылғысы, желілік қоршаулар, бір жағынан аппаратура, ал екінші жағынан пайдаланушы қосымшасы ретінде жұмыс жасайды. Тәртіп бойынша, ЖПҚ операциялық жүйелер, утилиттер, деректер қорын басқару жүйесі, және т.б. байланыстырушы ПҚ жатады.

Енгізілген программалар

Цифрлық электронды құрылғыларға тігілген программалар, мысалы, [BIOS IBM-PC сияқты](#).

Утилиттер

Қосымша компьютерлік программалар, арнайы типтік есептерді шешу үшін қолданылады, утилиттер кейбір мүмкіндіктерге жол ашып береді (параметрлер, күйге келтірулер, установкалар, олар ОЖ құрамына кіруі және арнайы құралдардың кешенінде болуы немесе жеке таратылуы мүмкін.

Қолданбалы программалық қамтамалар

Бұл жеке пайдаланушы есептерін анықтап шешу үшін қолданылады.

- **Әмбебап**
- **Арнайы**
- **Ғылыми программалық қамтамалар**
- **Ойын программалық қамтамалар**
- **Мультимедийлі программалық қамтамалар**
- **САПР**
- **Электронды құжаттау жүйесі**
- **Аспаптық программалық қамтамалар**

Компьютер архитектурасы

Компьютер архитектурасы — есептеу машинесінің концептуальды құрылымы, ол ақпараттарды өңдеу үшін техникалық құралдар және программалық қамтамалар арасындағы қарым-қатынасты орнатады.

Ядро (kernel) — ОЖ орталық бөлігі, компьютер ресурстарына қатынауды координациялайды.

Ядро сипаттамалары:

- [микроархитектура](#);
- Командалар жүйесі;
- Функциональды блоктар саны ([ALU](#), [FPU](#), конвейерлер және т.б.);
- Кірістірілген [кэш-жады](#) көлемі;
- интерфейс (логикалық және физикалық);
- такт жиілігі;
- қоректендіру кернеуі;
- максимальды және қалыпты жылу бөлуі;
- өндіріс технологиясы;
- кристалл көлемі.
- Жады архитектурасы,
- Сыртқы енгізу-шығару құрылғыларымен қарым-қатынасы
- [машиналық командалары](#),
- Өртүрлі ішкі деректер түрлері (например, [с плавающей запятой](#), [целочисленные](#) типы и т. д.),

Үзу өңдегіштер және ерекшеліктер

Микропрограмма кірістірілген программалық қамтамаларды білдіреді (*прошивка*):

Бұл негізінен ПЗУ-дағы интегральды микросұлбада жазылған және аппараттық қамтама жұмысын басқаратын компьютерлік программа. Егер процессордың микросұлбасын қосылып жазылатын болса онда ол арнайы аппараттық қамтамаларды басқаратын микроконтроллерге айналады.

Микрокод:

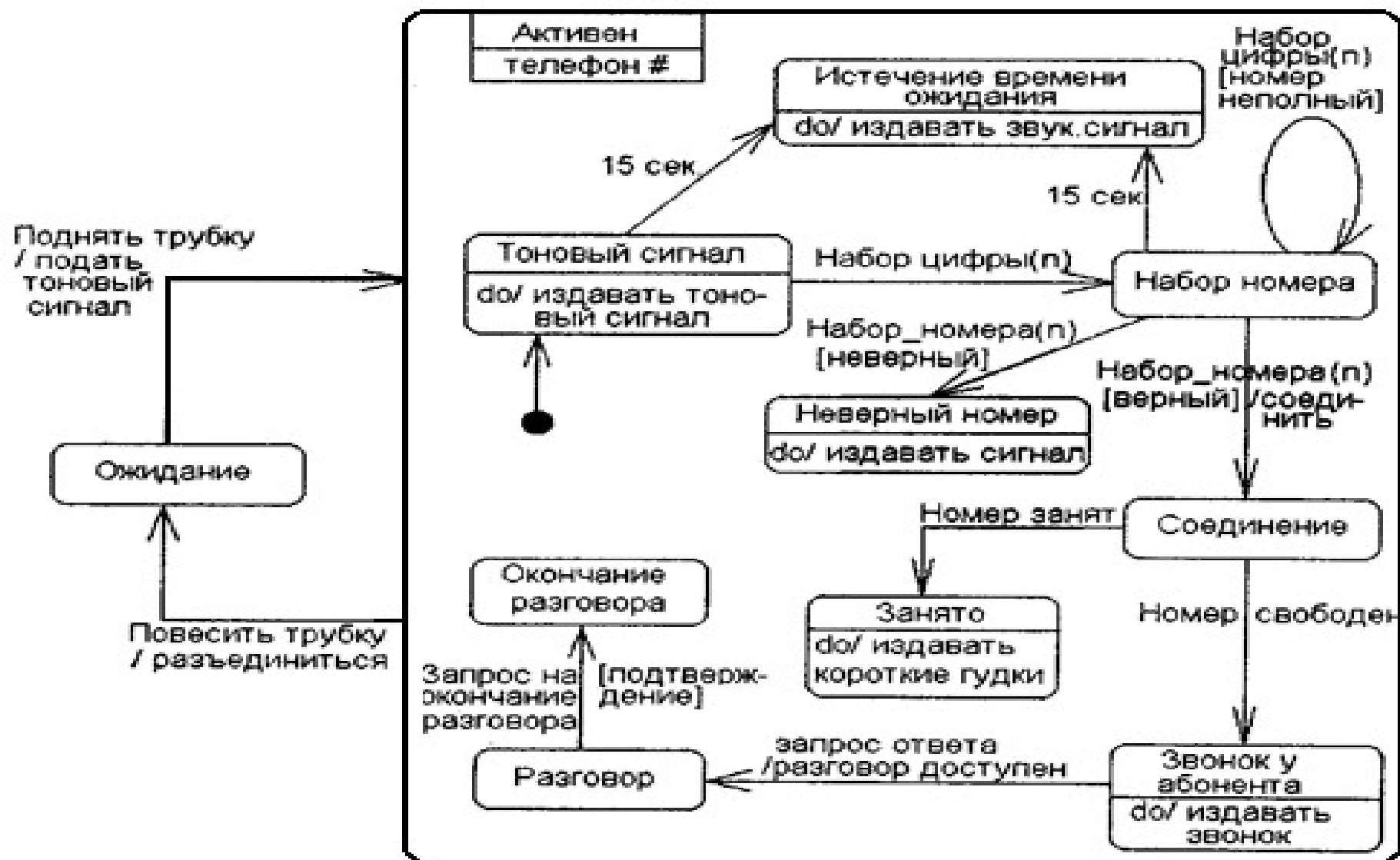
Такт бойынша есептеу құралдарының ресурстарын басқаратын программа (ALU, сдвигатели, мультиплексоры и др.). әдетте командалық сөздерде керекті құрылғыларды басқару үшін жеке биттер бөлінеді.

Микроархитектура

МП микросұлбасындағы моделін, топологиясын және ISA таратылуын сипаттайтын деңгей. Бұл деңгейде:

- Орталық МП негізгі блоктары және конструкциялары арасындағы қарым-қатынастар анықталады,
- АЛУ орындаушы құрылғыларының ядроларының құрылымы және олардың қарым-қатынастар анықталады,
- Өтулерді болжау блоктары,
- Конвейерлерді ұйымдастыру,
- Кеш-жадыны ұйымдастыру,
- Сыртқы құрылғылармен қарым-қатынастар анықталады.

Өтулер. Мысар ретінде телефонды көтеру өтулері



Компьютерлік инженериядағы микроархитектура

Компьютерлік инженерияда микроархитектура ([ағыл. *microarchitecture*](#)) — бұл процессорда таратылған командалар жиынының архитектурасының (ISA, КЖА) әдісі. Әрбір КЖА әртүрлі микроархитектуралардың көмегімен таратылуы мүмкін. Таратылулар берілген дизайнның мақсаты немесе технологияларды өзгерту нәтижелері бойынша түрленіп отырады. Компьютер [архитектура](#) [сы](#) микроархитектура, микрокод және КЖА комбинациялары болып табылады.

Микрокод және командалар жиынының архитектурасымен байланыс

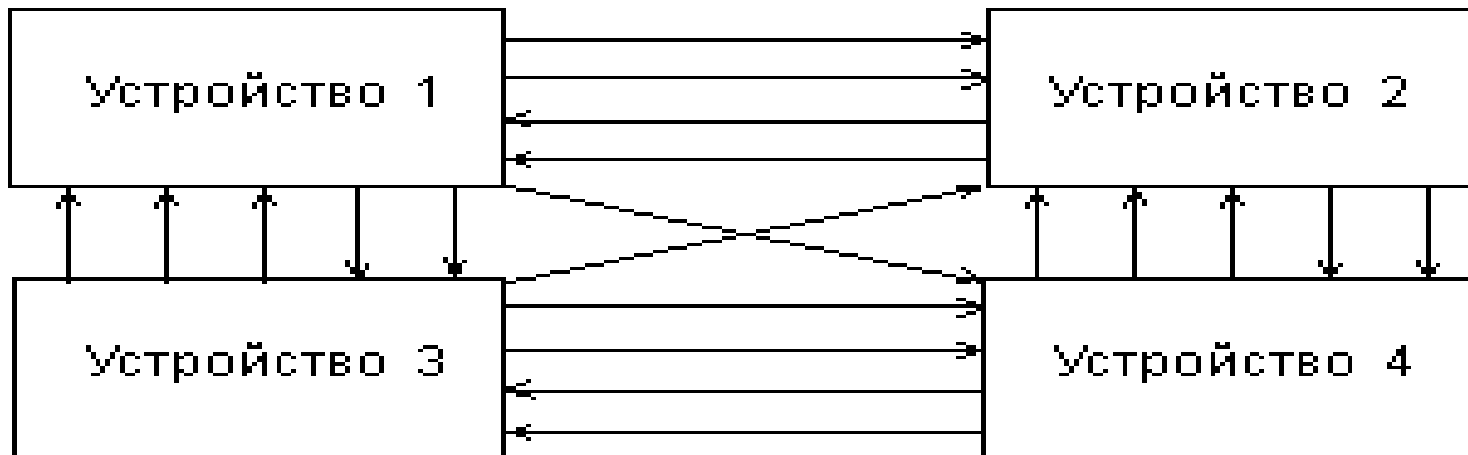
КЖА — бұл шамамен ассемблер тілідегі программалаушы немесе компилятор құрушының көзқарасы жағынан программалау моделі. КЖА, құрамына орындау моделі, процессор регистрлері, адрестер және деректер форматы, және сонымен қатар, процессордың құрама бөліктері және олардың байланыстары әрі КЖА таратуға арналған әрекеттер кіреді.

Бірақ көп жағдайларда микроархитектура элементтерінің жұмысы процессорға орнатылған микрокодтармен бақыланады. Процессор архитектурасында микрокод қабаты бар болса, онда ол өзіндік интерпретатор ретінде қызмет атқарады да, КЖА командалар деңгейін микроархитектура деңгейіндегі командаларға ауыстырады. Бұл кезде командалардың әртүрлі жүйелері бір микроархитектура базасында таратылуы мүмкін.

Машина микроархитектурасы әдетте азды, көпті бөлшекті диаграммаларды көрсетеді, олар әртүрлі микроархитектуралық элементтердің байланыстарын сипаттайды, мысалы: жеке вентилдерден бастап, регистрлерге дейін, АЛҚ және одан да үлкен элементтері. Бұл диаграммаларда әдетте деректер тракті (онда деректер орналасады) және басқару тракті (деректердің қозғалысын басқарады) бөлінеді.

Классикалық байланыс құрылымдары

Микропроцессорлық жүйелерде ақпараттарды алмасу протоколдарының жұмысын жеңілдету және максимальды әмбебаптыққа жету үшін жүйеге кірістірілген, жеке құрылғылар арасында шиналы байланыс құрылымдары қолданылады.



Байланыстың классикалық құрылымы

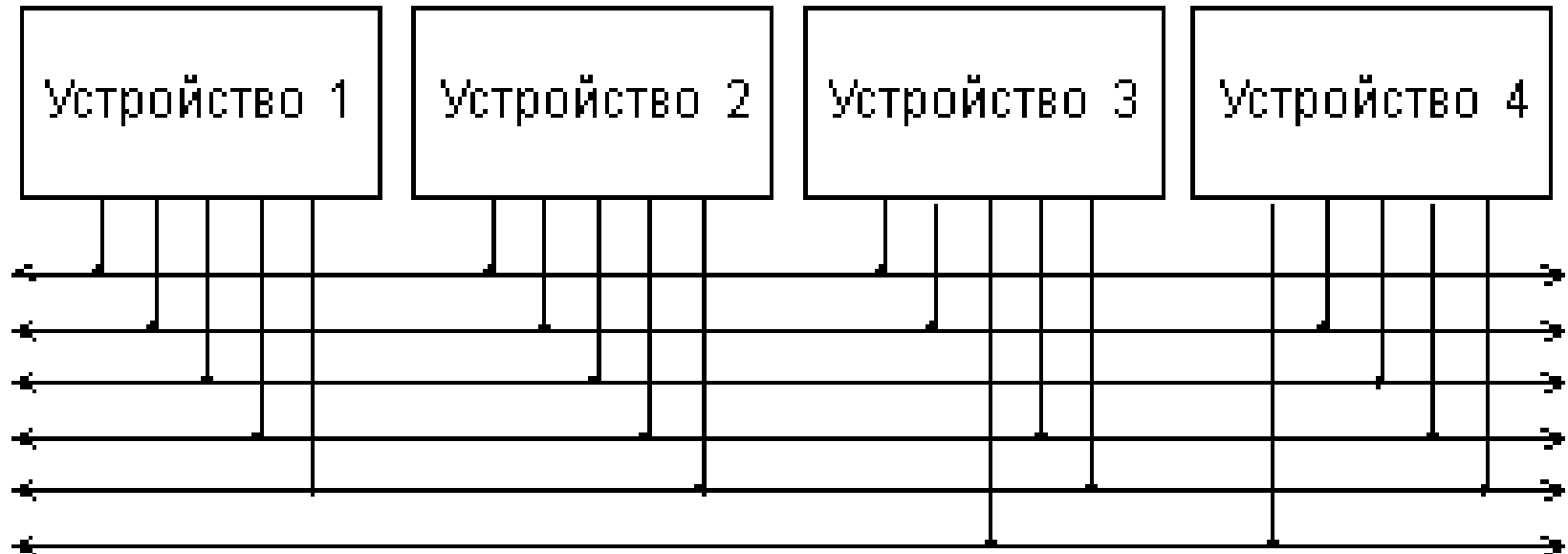
Классикалық байланыс құрылымдары

Классикалық байланыс құрылымы кезінде (суретте келтірілгендей) құрылғылар арасындағы барлық сигналдар және кодтар жеке байланыс сызықтары арқылы беріледі. Жүйеге енгізілген әрбір құрылғы өзінің сигналдарын және кодтарын басқа құрылғыларға тәуелсіз орындай береді. Бірақ жүйеде өте көптеген сызықтар және әртүрлі ақпараттарды *алмасу протоколдары* пайда болады.

Ал шиналық байланыс құрылымы кезінде (келесі суретте келтірілгендей) құрылғылар арасындағы барлық сигналдар бір ғана байланыс сызықтары арқылы әртүрлі уақыттарда жіберіледі (бұл мультиплексирленген алмасу деп аталады). Бұл кезде барлық байланыс сызықтары арқылы алмасулар екі бағытта да орындала береді (екібағытты алмасу деп аталады). Нәтижесінде байланыс сызықтарының саны азаяды және алмасу тәртіптері (протоколдары) жеңілдетіледі. Міне осы байланыс сызықтарының тобы немесе кодтары **шина** (англ. bus) деп аталады.

Шиналы байланыс құрылымдары

Шиналық байланыс құрылымы кезінде барлық ақпараттық ағындар бағыты жеңіл орындалатындығы түсінікті, мысалы, оларды бір процессор арқылы жіберуге болады, ол микропроцессорлық жүйелер үшін өте маңызды. Бірақ классикалық құрылымға қарағанда жылдамдығы аздау болады, себебі уақыт бойынша кезек күту керек.



Шиналық байланыс құрылымы

Шиналы байланыс құрылымдары

Бұл шиналық құрылымының артықшылығы, ол барлық шинаға қосылған құрылғылар бір тәртіппен жұмыс жасайтындығы. Сәйкесінше, барлық түйіндер, унифицирленген әрі біртекті болуы керек.

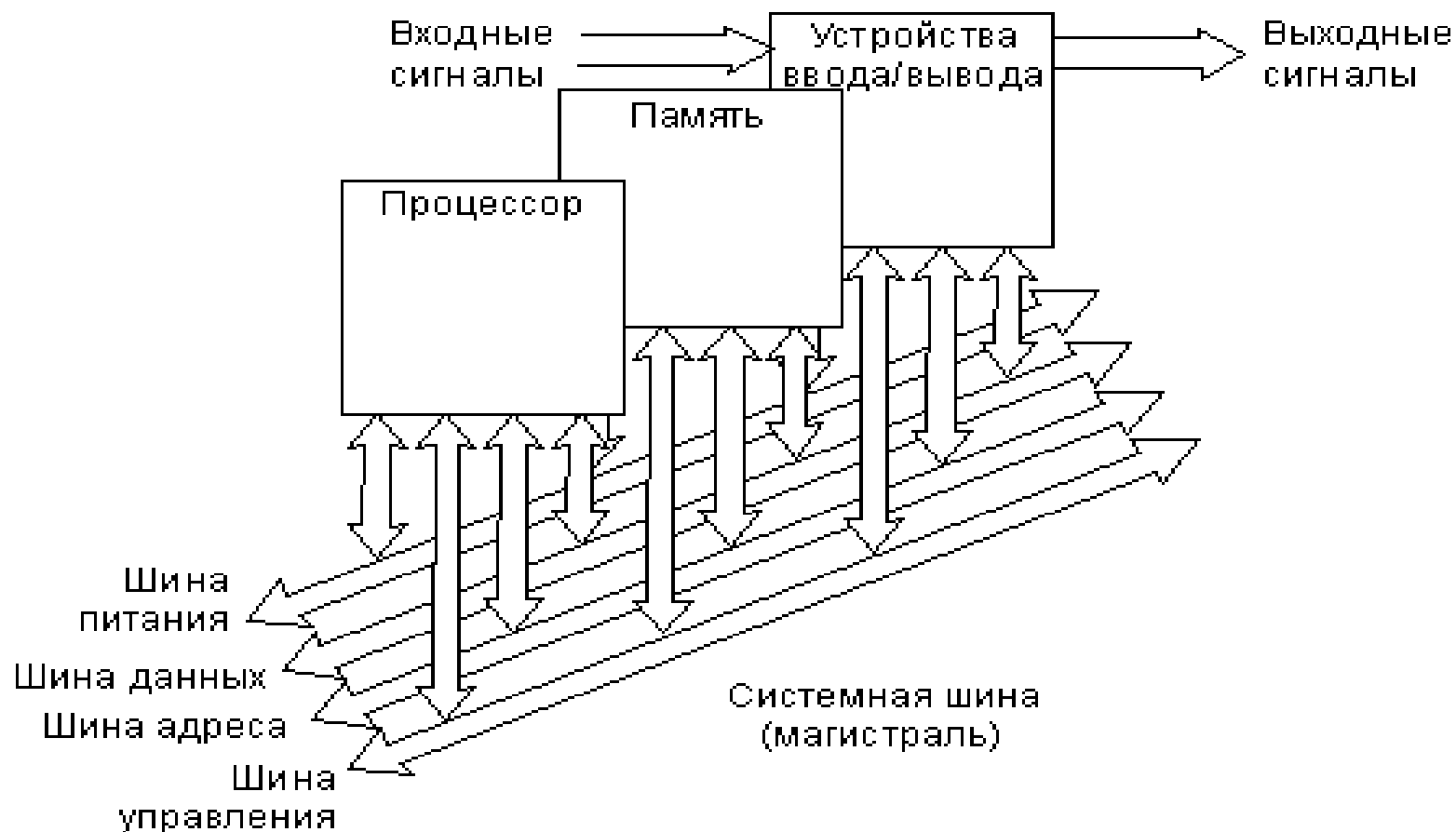
Ал кемшілігі, барлық құрылғылардың барлығы шинаға параллельді қосылған сондықтан қандай-да бір құрылымының дұрыс жұмыс жасамауы барлық жүйе жұмысын тоқтатады.

МПЖ құрылымы

Қарапайым микропроцессорлық жүйенің құрылымы келесі суретте көрсетілген. Ол негізгі үш құрылғы түрінен тұрады:

- процессор;
- *жады*, жедел *жады* (ОЗУ, RAM — *Random Access Memory*) және тұрақты *жады* (ПЗУ, ROM — *Read Only Memory*), тұрады, олар программалар және деректерді сақтау үшін керек;
- *енгізу/шығару құрылғысы* (ЕШҚ, I/O — *Input/Output Devices*), микропроцессорлық құрылғылардың сыртқы құрылғылармен байланыс үшін қолданылады, кіріс сигналдарын қабылдау үшін (енгізу, оқу, Read) және шығыс сигналдарын шығару үшін (шығару, жазу, Write) қолданылады.

МПЖ құрылымы



Микропроцессорлық жүйе құрылымы

Барлық микропроцессорлық жүйедегі құрылғылар жалпы жүйелік шина арқылы біріктіріледі (ол жүйелік магистраль немесе канал деп те аталады). Жүйелік *магистраль* төрт негізгі төменгі деңгейлі шиналардан тұрады:

- *адрес шинасы (Address Bus);*
- *деректер шинасы (Data Bus);*
- *басқару шинасы (Control Bus);*
- *қорек беру шинасы (Power Bus).*

МПЖ құрылымы

Адрестер *шинасы* берілген уақытта процессор ақпарат алмасатын құрылғы адресін (нөмірін) анықтау үшін қолданылады. Әрбір құрылғыға (процессордан басқа), микропроцессорлық жүйенің әрбір ұяшығында жеке адрес меншіктеледі. Егер процессормен қандай-да бір адрес, адрестер шинасына берілген болса, онда ол ақпарат алмасу керек екендігін түсінеді. Адрестер *шинасы* бірбағытты және екібағытты болуы мүмкін.

Деректер шинасы — бұл негізгі *шина*, ол микропроцессорлық жүйенің барлық құрылғылары арасында ақпараттық кодтарды алмасу үшін қолданылады. Әдетте дерекалмасу кезінде процессор қатысады, ол деректер кодын қандай-да бір құрылғыға немесе жады ұяшығына береді немесе керісінше қандай-да бір құрылғыдан немесе жады ұяшығынан қабылдайды. Бірақ процессордың қатысуынсыз да алмасулар орындала береді. Деректер *шинасы* барлық уақытта екібағытты болады.

Басқару шинасы адрес және деректер шинасынан ерекшелігі ол жеке басқару сигналдарынан тұрады. Бұл сигналдардың әрбірінің ақпараттар алмасу кезінде өзінің функциясы болады. Кейбір сигналдар берілетін немесе қабылданатын деректер үшін шешім қабылдау үшін пайдалануы мүмкін (мысалы уақыт моментін анықтау). Басқа басқару сигналдары деректерді қабылдауды бекіту, барлық құрылғыларды бастапқы күйге келтіру, барлық құрылғыларды тактовкалау үшін және т.б. қолданылады. Басқару шиналары бірбағытты және екібағытты болуы мүмкін.

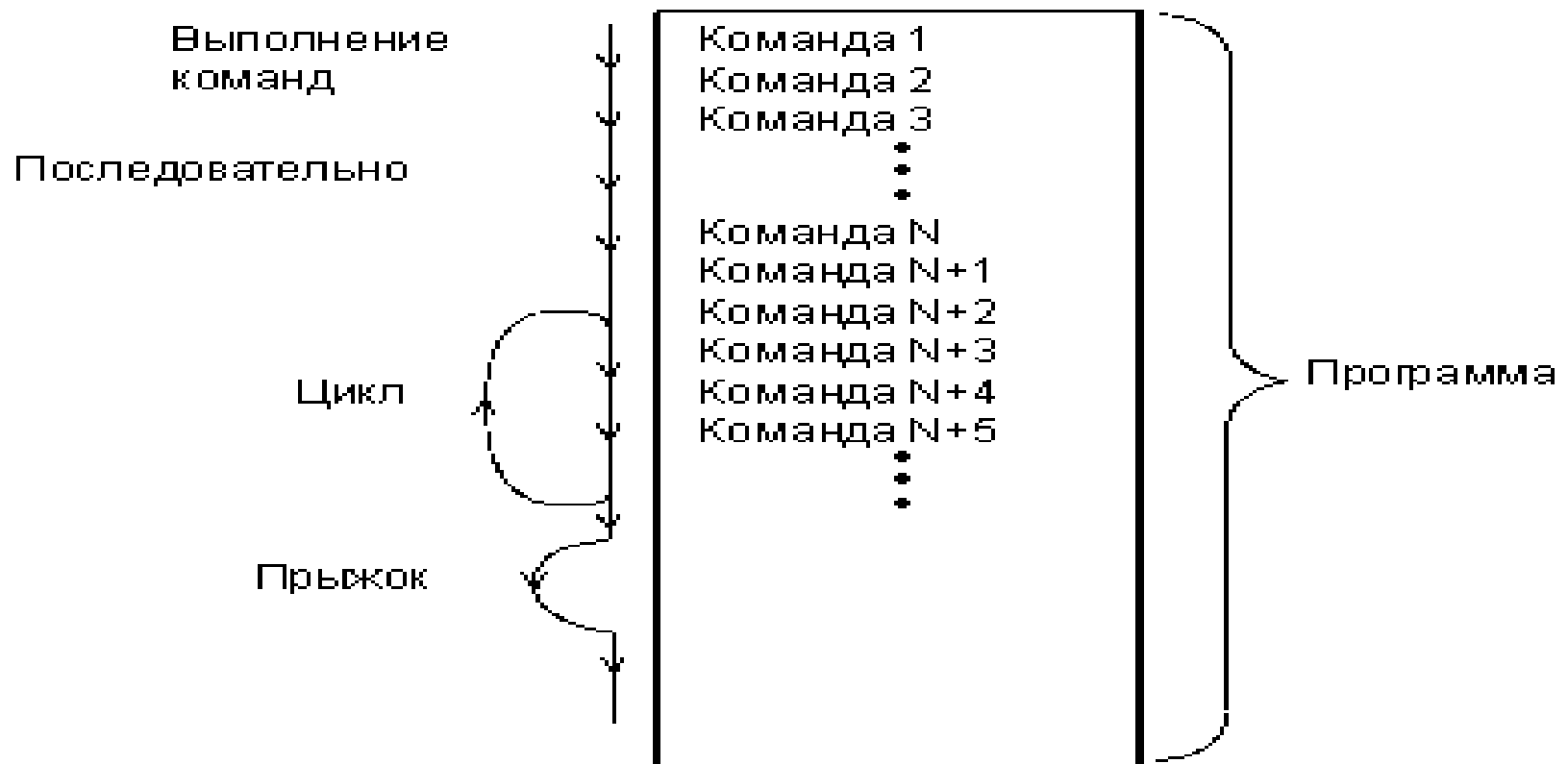
Және ең соңғысы, **қорек беру *шинасы*** ақпараттық сигналдарды беру үшін емес, ал жүйені қоректендіру үшін пайдаланылады. Ол қоректендіру сызықтары және жалпы сымдардан тұрады. Микропроцессорлық жүйеде бір қоректендіру көзі (жиі +5 В) немесе бірнеше (әдетте -5 В, +12 В және -12 В) болуы мүмкін. Әрбір қорек көзінің өзінің байланыс сызығы болады. Барлық құрылғылар бұл сызықтарға параллельді орналасытырылады.

Тәжірибе жүзінде барлық микропроцессорлық жүйелер үш магистраль бойынша алмасу режимдерін қолдайды:

- программалы ақпарат алмасу;
- үзулер арқылы алмасу (*Interrupts*);
- жадыға тікелей қатынау арқылы алмасу (*ПДП, DMA — Direct Memory Access*).

Программалы ақпарат алмасу

Магистраль бойынша барлық сиагналдар процессордың бақылауында болады.

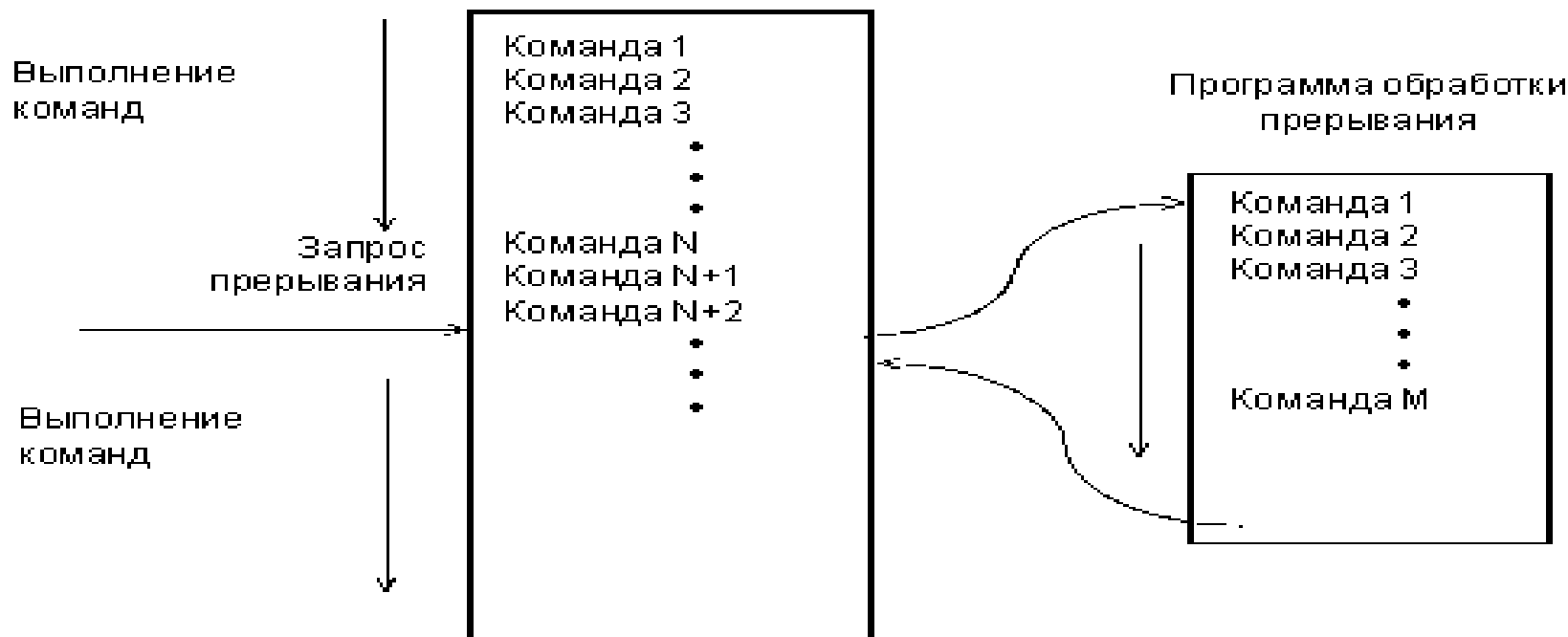


Программалы ақпарат алмасу

Үзулер арқылы алмасу

Магистраль бойынша барлық сигналдар контроллердің бақылауында болады, ал контроллер жұмысын процессор бақылайды.

Основная программа

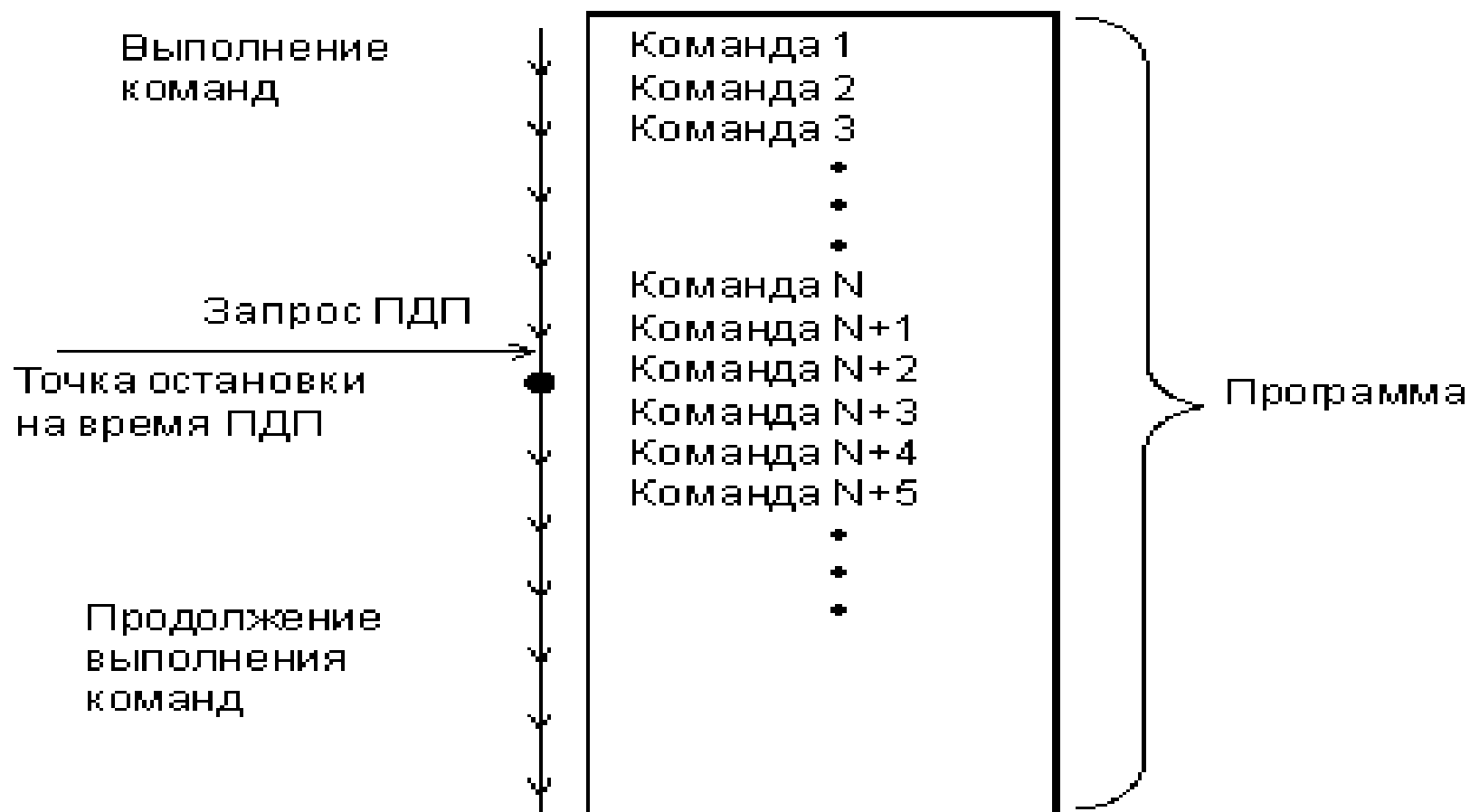


Үзулерді өңдеу

Жадыға тікелей қатынау арқылы алмасу

(ПДП, DMA) — бұл режим алдында көрсетілген режимдерден ерекшеленеді, себебі бұл кезде басқару процессордың қатысуынсыз орындалады. Яғни, бұл жағдайда жүйеге қосымша ЖТҚ контроллері құрылғысы енгізілу керек, ол процессордың бақылауынсыз магистраль бойынша алмасуды орындайды. Бірақ ол ақпаратты қайдан алуы керек екендігін және қайда орналастыруы керек екендігін процессор алдын ала айтып отырады. Сондықтан ЖТҚ *контроллерін* арнайы процессор ретінде қарауға болады (келесі сурет).

Жадыға тікелей қатынау арқылы алмасу



ЖКҚ өңдеу

Командалар конвейрін ұйымдастыру

Қазіргі заманғы микроархитектураларда конвейерлік деректер тракті кеңінен қолданылады деп есептеледі. Бұл техника көптеген қазіргі заманғы микропроцессорларда, микроконтроллерде және цифрлық сигнальды процессорларда қолданылады. Конвейерлік архитектура бірнеше инструкцияның орындалуы кезінде үзілуіне мүмкіндік береді, әрі жинау құрылымын қамтамасыздандырады.

Командалар жүйесін таңдау

Жылдар өте командалар жүйесі қарапайымнан, өте қиынға дейін дамыды. Соңғы жылдары ең кеңінен таратылғандардың қатарына RISC, VLIW, EPIC архитектураларын жатқызуға болады. Деректер деңгейінде параллельділікпен байланысты архитектураларға SIMD және векторлы процессорларды жатқызуға болады.

Инструкциялар конвейеризациясы

Өнімділікті арттырудың мықты техникаларының бірі ретінде конвейерлік инструкцияны алуға болады. Бұрынғы процессор моделдерінің барлығы қадамдардың барлығын орындап болғаннан кейін ғана инструкцияларын ауыстыра алатын. Ал конвейерлер бірнеше инструкцияларға бір уақытта бірнеше жол салуға мүмкіндіктер береді.

Кэш

Кэш жады және конвейерлер бір бірін жақсы толықтырады. Егер бастапқыда конвейерлер құрудың маңызы болмаса, кештің пайда болуымен оның маңызы өсті, кэш конвейерлер чиптегі жедел жадыға қатынаудың жылдамдығын арттырды. Нәтижесінде процессорлардың тактілік жиіліктерін арттыруға мүмкіндіктер пайда болды.

Суперскалярлық

Бастапқы кезде конвейерлік микроархитектурадағы процессорлардың өзі тек бір инструкцияны ғана орындай алатын. Сондықтан біруақытта бірнеше инструкцияны жіберу қарастырылды. Міне осы кезде суперскалярлы микроархитектуралар пайда болды, ол бірнеше бірдей функциональды блоктарды қолдану арқылы жүзеге асырылды, мысалы АЛҚ.

Қазіргі заманғы өңдеулерде екі таңдау құрылғысын және, бір сақтау құрылғысын, екі немесе одан көп бүтін сандар АЛҚ, екі немесе одан көп жылжымалы нүктелі сандар құрылғысын көруге болады.