



**«ТҰРАН» УНИВЕРСИТЕТІ**

**Г.Б. КАШАГАНОВА**

**ЕСЕПТЕУ ЖҮЙЕЛЕРІНІҢ АРХИТЕКТУРАСЫ  
ЖӘНЕ ҰЙЫМДАСТЫРЫЛУЫ**

**Оқу-әдістемелік құрал**

**Алматы  
2020**

ӘОЖ 004  
КБЖ 32.973  
К31

Баспаға «Тұран» университетінің Ғылыми кеңесі ұсынған  
(09.04.2020 ж. № 11 хаттама)

Рецензенттер:

И.Т.Утепбергенов, «Тұран» университеті «Компьютерлік және бағдарламалық инженерия» кафедрасының профессоры, техника ғылымдарының докторы

О.Ж. Мамырбаев, PhD, қауымдастырылған профессор, ҚР БҒМ ҒК «Ақпараттық және есептеу технологиялары институты» РМК бас директорының орынбасары

Кашаганова Г.Б.

К31 Есептеу жүйелерінің архитектурасы және ұйымдастырылуы: Оқу-әдістемелік құрал/ Г.Б. Кашаганова – Алматы: «Тұран» Университеті, 2020. – 55 б.

Оқу-әдістемелік құралында қазіргі әлемде ерекше рөл атқаратын есептеу машиналары мен есептеу жүйелерінің негіздері қарастырылған. Материалды баяндау студентті модельдеу мен жобалаудың қазіргі заманғы жүйелерін жүйелі және өзіндік зерттеуге дайындау үшін де көлемде жасалған.

«Компьютерлік және бағдарламалық инженерия» кафедрасында 6В06102 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығының білім беру бағдарламасы негізінде әзірленген және «Есептеу жүйелерінің архитектурасы және ұйымдастырылуы» пәндерінің оқу бағдарламасына сәйкес келеді.

© Кашаганова Г.Б., 2020  
© «Тұран» Университеті, 2020

## МАЗМҰНЫ

Дәріс 1.	Компьютер тарихы	4
Дәріс 2.	Логикалық негіздер	9
Дәріс 3.	Мультипроцессорлы есептеу жүйесінің жұмысының мақсаты, көлемі және әдістері	13
Дәріс 4.	Есептеу жүйелерінің архитектурасы. Параллель мәліметтерді өңдеуге арналған архитектуралардың жіктелуі	17
Дәріс 5.	Есептеу жүйелерінің архитектурасы. SMP және MPP сәулеті. Гибридтік сәулет (NUMA). Көп деңгейлі иерархиялық жадтың үйлесімділігін ұйымдастыру	19
Дәріс 6.	Есептеу жүйелерінің архитектурасы. ПВП архитектурасы. Кластер сәулеті	22
Дәріс 7.	Байланыс құралдарын құру принциптері	28
Дәріс 8.	Жоғары өнімді процессорларды ұйымдастыру жолдары. Ассоциативті процессорлар. Конвейерлік процессорлар. Матрицалық процессорлар	32
Дәріс 9.	Жоғары өнімді процессорларды ұйымдастыру жолдары. Жасушалық және ДНҚ процессорлары. Байланыс процессорлары	34
Дәріс 10.	Жоғары өнімді процессорларды ұйымдастыру жолдары. Мәліметтер базасының процессорлары. Ағын процессорлары. Нейрондық процессорлар. Көп мағыналы (анық емес) логикасы бар процессорлар	35
Дәріс 11-12.	I / O құрылғылары.	40
Дәріс 13.	Ақпаратты өңдеу жүйелері. ВК мультипроцессоры. Деректерді тарату жүйелері (ДТЖ)	42
Дәріс 14.	Компьютерлік желілер	48
Дәріс 15.	Есептеу жүйелері мен желілерін жобалау негіздері	50

## **Дәріс 1. Компьютер тарихы**

*Аннотация:* Дәріс компьютерлердің даму тарихын талқылайды, компьютерлердің ұрпақтарын, әртүрлі буындардың компьютерлерінің параметрлерін, компьютерлердің өзіндік құнын анықтайды. Ақпараттық технологиялардың 3 кезеңі, сонымен қатар компьютер жұмысының негізгі принциптері ұсынылған.

Арифметикалық есептеулерді автоматты түрде орындайтын құрылғыны құру үшін бағдарламалық басқаруды қолдану идеясын алғаш рет 1833 жылы ағылшын математигі С.Барж айтқан. Алайда оның компьютермен басқарылатын механикалық есептеу құрылғысын жасау әрекеттері сәтсіз болды.

Бірінші жұмыс істейтін әмбебап автоматты басқарылатын «Марк - 1» есептік-механикалық машинасы болып саналады (АҚШ, 1944). Машинаның тоқтап қалуы көбінесе құрады. Жақсартылған релелік конструкцияға негізделген Марк-2, төмен өнімділікке ие болды.

Бірінші ENIAC компьютерінің жобасын Дж.Моули (АҚШ, 1942 ж.) Жасаған; 1946 жылы автокөлік іске қосылды. Бұл машинада 18000 электр шамдары, 1500 электромеханикалық реле бар. Шамдарды пайдалану «Марк - 1» құрылғысымен салыстырғанда жұмыс жылдамдығын 1000 есе арттырды.

Компьютерлер дәуірінің анықтамалығы үшін 1946 жылы Пенсильвания университетінде басталған ENIAC машинасының сынақ жұмысының сессиялары алынады.

Бұл компьютердің тағы бірнеше техникалық сипаттамалары: жалпы салмағы - 30 тонна, өнімділігі - секундына 5000 операция. Алғашқы компьютер іске қосылғаннан 40 жыл өткен соң, ВТ компоненттерінің жылдық өндірісі 1985 жылға қарай есептелді. 1014 белсенді логикалық элементтерде (белсенді элементтер тобы), бұл жердің әр тұрғынына 1 ENIAC-қа тең. Салыстыру үшін: 1962 жылға қарай типографияның 500 жылдан астам дамуы. барлық басылымдардың жалпы тиражы Жердің әр тұрғыны үшін 2 кітап деңгейіне жетті.

Электрондық түтіктер бірінші буындағы ВМ-нің элементтік базасына айналды. Негізгі схема - симметриялы триггер 1918 жылы құрылды. Кеңес ғалымы Бонч-Бруевич М.А. 1919 жылы Осыған ұқсас схеманы американдық ғалымдар Иклес пен Иордания да жасаған.

Отандық компьютерлердің алғашқы жобаларын S.A. ұсынған. Лебедев, Б.И. Рамеев 1948 ж 1949-51 жж. жоба бойынша S.A. Лебедева MESM (шағын электронды компьютер) салынды. Бірінші буын компьютеріне сонымен бірге S.A. жетекшілігімен әзірленген BESM-1 (үлкен электронды компьютер) кіреді. Лебедева 1952 жылы салынып бітті, оның құрамында 5 мың шам бар және 10 сағат бойы тоқтаусыз жұмыс істеді. Өнімділік

секундына 10 мың операцияға жетті. Strela компьютері бір уақытта дерлік Ю.Я. Базилевский, 1953 ж. ол өндіріске енгізілді. Кейіннен «Орал-1» компьютері пайда болды, ол Б.И. жетекшілігімен дамып, өндіріске енгізілген көптеген Орал машиналарының негізін қалады. Рамеева. 1958 жылы бірінші буын М-20 сериялық өндіріске шығарылды (жылдамдығы 20 мыңға дейін).

1950 жылдардың ортасында транзисторлардың пайда болуымен бірінші буын компьютерлерді жартылай өткізгіш құрылғыларда құрылған екінші буын компьютерлерімен алмастырды.

Біздің елде әртүрлі мақсаттағы жартылай өткізгіш компьютерлер құрылды: «Наири» және «Мир» сериялы шағын компьютерлер, жылдамдығы 5-30 мың жылдамдықтағы орташа компьютерлер - Минск-22 және Минск-32, Храздан-2 «,» Храздан - 3 «, BESM - 4, М - 220 және ең жақсы екінші буындағы машиналар - BESM - 6 жылдамдығы 1 миллион опера / с дейін.

60-жылдардың басында электроникада жаңа бағыт пайда болды - біріктірілген электроника. Компьютерлерді құру үшін интегралды схемаларды қолдану VT-дегі революция болды және 3-ші буын машиналарының пайда болуына ықпал етті.

1972 жылдан бастап ЕО компьютерлерінің бірінші кезеңінің (социалистік елдермен бірге) модельдерін шығара бастады. Жол - 1: ЕС - 1010, 1020, 1022, 1030, 1033, 1040, 1050, 1052. Екінші кезең (Жол - 2): ЕС - 1015, 1025, 1035, 1045, 1055, 1060, 1065 заманауи электр тізбегін, конструкторлық және технологиялық база, соның арқасында олардың өнімділігі артып, функционалдығы кеңейе түсті.

4-буын ЭЕМ-нің бір ерекшелігі - интегралды функционалды тізбектерден біріктірілген компьютерлік ішкі жүйелерге көшу. LSI енгізу сенімділікті кем дегенде 10 есе арттырады деген болжам бар. Отандық компьютерлерден бастап 4-ші буындағы машиналарға, ең алдымен, Эльбрус отбасына жатады.

1.1-кестеде электр тізбегінің негізгі параметрлері мен компьютер буындарының арасындағы байланыс көрсетілген. Орындау бір элементар элемент (конъюнктор, дезуктор және т.б.) енгізген сигналдың таралуының кідірісімен сипатталады. Маңызды индикатор орамның тығыздығы, 1 см<sup>3</sup>-ге элементтердің саны.

Кесте 1.1

Қол кою, компьютер параметрі	Ұрпақтар				4-ші кейін 80ж.
	1-ші 1946- 1955	2-ші 1955- 1965	3-е 965-1970	п кейін 70ж.	
Негізгі элементтер	Эстафеталар, электронды лампарлар	Жартылай өткізгіштер	ИС	БИС	СБИС
Орындау	1мс	1мкс	10нс	1нс	< 1нс
Қаптама тығыздығы, элементтер / см <sup>3</sup>	0,1	2-3	10-20	1000	> 10000

30 жылдан кейін компьютерлік индустрия өтеді, суретте көрсетілгендей. Жалпы қаржылық салмақтың 1,1 миллиардтан бір бөлігі, ол АҚШ-тың барлық өндіріс салаларында сатылымның ең жоғары өсу қарқынын сақтап келеді.

Ақпараттық технологияның үш кезеңі: критерийлер эволюциясы.

1953 жылы Ақпарат теориясының негізін қалаушы, американдық математик Клод Шеннон: «Біздің ВМ схоластикалық ғалымдарға ұқсайды. Арифметикалық амалдардың ұзақ тізбегін есептеу кезінде цифрлық компьютерлер адамды айтарлықтай басып озады. Сандық компьютерлерді арифметикалық емес операцияларды орындауға бейімдеуге тырысқанда, олар ыңғайсыз және мұндай жұмысқа жарамсыз» деп жазды.

**1 кезең:** машина қорлары. Шеннон атап өткен функционалдық шектеулер, сондай-ақ алғашқы компьютерлердің қорқынышты құны 1950-1960 жж. Ақпараттық технологияның негізгі міндетін толығымен анықтады. - дайындалған немесе оңай рәсімделген алгоритмдерді қолданып мәліметтерді өңдеу тиімділігін арттыру.

Содан кейін негізгі мақсат оны шешу үшін қажет машина сағаттарының жалпы санын, сондай-ақ жұмыс істейтін жедел жад көлемін азайту болды. Деректерді өңдеудің негізгі шығындары сол кезде олар жұмыс істейтін уақытқа тікелей байланысты болды.

**2 кезең:** бағдарламалау. 60-жылдардың ортасында ақпараттық технологиялар дамуының 2-ші кезеңі басталды, ол 80-жылдардың басына дейін жалғасты. Бағдарламаны тиімді орындау технологиясынан бастап тиімді бағдарламалау технологиясына дейін, осы кезеңдегі өнімділік критерийлерін өзгертудің жалпы бағытын анықтауға болады. Бағдарламалау технологиясының өлшемдерін алғашқы түбегейлі қайта

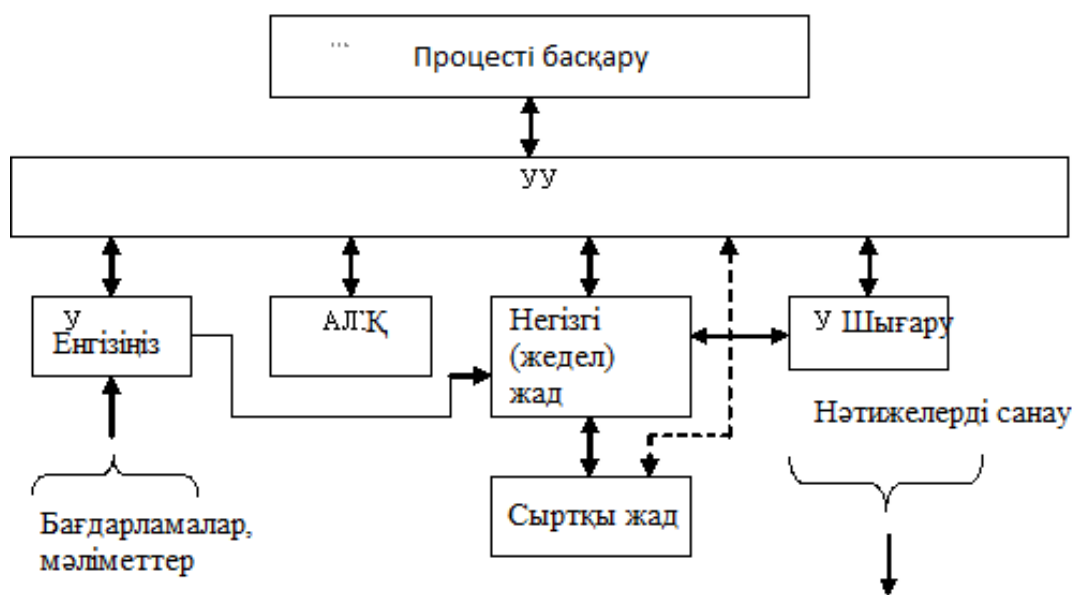
қараудың ең танымал нәтижесі 70-ші жылдардың басында құрылған UNIX ОС болды. Ең алдымен бағдарламашылардың жұмысының тиімділігін арттыруға бағытталған UNIX операциялық жүйесін Bell Labs қызметкерлері К.Томпсон және Д.Ритчи әзірледі, олар қарабайыр партиялық режимге бағытталған бағдарламалық жасақтама құралдарымен толығымен қанағаттанбады. 80-ші жылдардың басында UNIX классикалық ОС үлгісі ретінде қарастырылды - ол 70-ші жылдардың ортасында PDP-11 сериялы шағын компьютерлерде салтанатты шеруді бастады.

**3 кезең:** білімді ресімдеу. Әдетте, дербес компьютер, жаңадан келген пайдаланушыны қашықтан басқару құралында жұмыс істеуге өзін-өзі даярлау құралдарын, оның қателіктерінен қорғаудың икемді құралдарын ойлап тапты, ең бастысы, мұндай компьютердің барлық аппараттық және бағдарламалық жасақтамалары бір «супер тапсырмаға» бағынады - машинаның кез-келгеніне «достық реакциясын» қамтамасыз етеді. пайдаланушының орынсыз әрекеттерін қоса. Жеке есептеудің негізгі міндеті - кәсіби білімді ресімдеу, әдетте, бағдарламалаушы емес қолданушы өздігінен немесе бағдарламашының минималды техникалық қолдауымен жүзеге асырылады.

### **Компьютер жұмысының принциптері**

Адам қызметінің кез-келген формасы, техникалық объектінің жұмыс істеуінің кез-келген процесі ақпараттың берілуімен және өзгеруімен байланысты. Ақпарат белгілі бір табиғи құбылыстар, қоғамдық өмірдегі оқиғалар және техникалық құрылғылардағы процестер туралы ақпаратты білдіреді. Материалдық түрде енгізілген және жазылған ақпарат хабарлама деп аталады. Хабарламалар үздіксіз және дискретті (сандық) болуы мүмкін. Үздіксіз (аналогтық) хабар физикалық шама (электр кернеуі, ток және т.б.) арқылы ұсынылады, уақыт өте келе өзгерістер осы процесстің барысын көрсетеді.

Дискретті хабарлама белгілі бір уақыт аралығында белгілі бір нүктелерде әртүрлі реттіліктер пайда болатын элементтер жиынтығының болуымен сипатталады. Компьютерлер немесе компьютерлер - бұл ақпарат түрлендіргіші. Оларда мәселенің бастапқы деректері оны шешудің нәтижесіне айналады. Ақпаратты ұсыну үшін қолданылатын пішінге сәйкес машиналар 2 классқа бөлінеді: үздіксіз жұмыс - аналогты және дискретті жұмыс - сандық. Біз компьютерлерді (сандық) зерттейміз.



Сурет 1.3. Компьютердің классикалық блок-схемасы

Арифметикалық логикалық құрылғы (АЛҚ) – машина сөздерін түрлендіреді

Жад - негізгі немесе жедел (ішкі) жад; сыртқы жад

Жад ұяшықтары нөмірленеді, ұяшық нөмірі адрес деп аталады.

Компьютерлерде жад функциясын орындайтын жад құрылғыларында басқа құрылғыларға беру және басқа құрылғылардан ақпаратты жазу үшін сақталған ақпаратты оқу операциялары орындалады.

Есепті сандық әдіспен шешу алгоритмі - бұл мәселенің шешімін алу үшін бастапқы деректер мен аралық нәтижелерде орындалуы қажет арифметикалық және логикалық амалдардың реттілігі. Алгоритмді қандай операцияларды, қандай ретпен және қай сөзбен орындау керектігін көрсету арқылы анықтауға болады. Алгоритмді компьютер қабылдайтын формада сипаттау бағдарлама деп аталады.

Бағдарлама жеке командалардан тұрады. Әрбір команда белгілі бір әрекетті тағайындайды және осы әрекеттің қай сөздерде (операндалар) орындалатынын көрсетеді. Бағдарлама дегеніміз - компьютердегі мәселені шешуді қамтамасыз ететін белгілі бір ретпен жазылған командалар жиынтығы.

Мысалы, есептеу керек

$$F = (a - x)/(ax + c),$$

a, c, x сандық мәндері үшін. Есептеу бағдарламасын F келесі командалар тізбегімен ұсынуға болады:

a - x ;

a\*x ;



$$ax + c ;$$
$$(a - x)/(ax + c).$$

Басқару құрылғысы командаларды қабылдауы үшін олар сандық түрде кодталуы керек.

Мәселені шешу процесін автоматты түрде басқару компьютерлердің басты ерекшелігі болып табылатын бағдарламалық басқару принципі негізінде жүзеге асырылады.

Тағы бір маңызды қағида - жадта сақталатын бағдарламаның принципі, соған сәйкес сандық түрде кодталған бағдарлама жадта сандармен қатар сақталады. Команда операцияларға қатысатын сандарды өздері көрсетпейді, бірақ олар орналасқан ұяшықтардың мекен-жайын және операция нәтижесі орналастырылған ұяшықтың мекен-жайын көрсетеді.

Екілік тізбектерді, бағдарламаны басқару принциптерін және жадта сақталатын бағдарламаны қолдану есепті циклдардан тұратын есептерді шығаруда есептерді шығарудағы командалардың санын осы бағдарламаларды орындау кезінде орындайтын операциялардың санымен салыстырғанда бірнеше есе азайтуға мүмкіндік берді. Пәрмендер осы тәртіпті сөзсіз өзгертетін шартты емес және шартты ауысу командаларын қоспағанда, кезекті жад ұяшықтарында орналасқан орындарына сәйкес ретпен орындалады немесе белгілі бір шарт, әдетте нөлге теңдік ретінде белгіленген, алдыңғы команданың оң немесе теріс нәтижесі немесе  $<$ , пәрменмен көрсетілген сандар үшін  $=, >$ . Шартты ауысу командаларының болуына байланысты компьютер автоматты түрде жүретін процестің барысын өзгерте алады және күрделі логикалық есептерді шеше алады.

Кіріс құрылғыны қолдана отырып, бағдарлама және бастапқы деректер оқылады және ОР-ге жіберіледі.

## **Дәріс 2.      Логикалық негіздер**

*Аннотация:* Дәріс логикалық алгебраның тұжырымдамасын береді, талдау және синтездеу мәселелерін сипаттайды. Бір және екі айнымалы элементар функциялардың сипаттамасы келтірілген. Негізгі эквиваленттер ұсынылған.

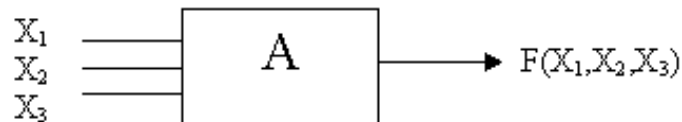
### **Логика алгебрасы**

Кәдімгі алгебрадан басқа, арнайы негіз бар, оның негізін 19 ғасырдағы ағылшын математигі Дж.Бул салған. Бұл алгебра пропозициялық есептеулер деп аталады.

Оның ерекшелігі - автоматика мен компьютерлік құрылғылардың бүкіл класын қамтитын дискретті құрылғылардың жұмысын сипаттауға болатындығы.

Сонымен қатар, алгебраның өзі құрылғының моделі ретінде әрекет етеді. Бұл көрсетілген типтегі ерікті құрылғының жұмысы тек кейбір жағынан осы алгебраның конструкцияларын қолдану арқылы сипатталуы мүмкін дегенді білдіреді. Нақты нақты құрылғы физикалық түрде жұмыс істемейді, өйткені оны логика алгебрасы сипаттайды. Алайда, осы теорияның ережелерін қолдану практикалық тұрғыдан бірқатар пайдалы жалпылама тұжырымдар жасауға мүмкіндік береді.

Белгілі бір схеманы қарастырып, оны «қара» қорап түрінде ұсыныңыз.



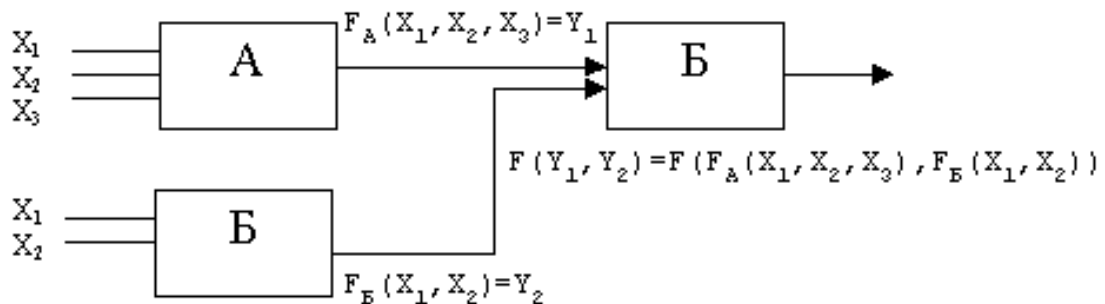
Қораптың ішкі мазмұны белгісіз деп санаймыз.

$X_1, X_2, X_3$  - кіріс сигналдары,  $F$  - шығу сигналы.

Сонымен қатар, біз  $A$  тізбегі қарапайым деп санаймыз, яғни.  $A$ -дан кіші басқа  $B$  схемасы жоқ, олар  $A$ -да болады.

Біз қарапайым құрылғылардан абстрактілі құрылғыны, мысалы,  $A, B, C$ , т.б. Күрделі құрылысты қарапайым жолдардан жасауға болатыны анық:

- элементтерді тізбектей қосу;
- параллель қосылу;
- элементтердің енгізілуін есептеу.



Содан кейін  $B$  екінші элементі үшін  $Y_1$  рөлі ойнайды:

$$Y_1 = F_A(X_1, X_2, X_3)$$

$$Y_2 = F_B(X_1, X_2)$$

$$F = F(Y_1, Y_2) = F(F_A(X_1, X_2, X_3), F_B(X_1, X_2))$$

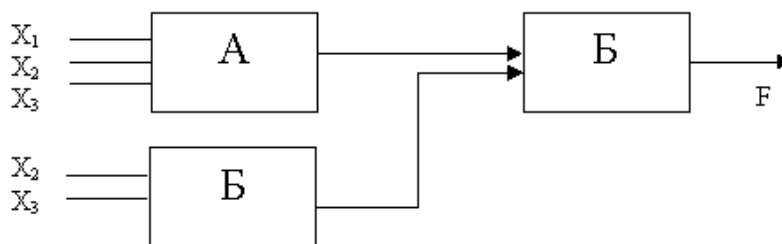
Элементтердің параллель қосылуы функцияны өзгертпейді, сондықтан логика тұрғысынан бұл байланыстың түрі қолданылмайды. Физикалық

тұрғыдан кейде олар әлі де элементтердің параллель байланысын қолданады, бірақ негізінен, мысалы, сигналды күшейту үшін.

Осыған байланысты логика алгебрасындағы элементтердің параллель қосылуы қарастырылмайды.

Элемент орындайтын функция, әдетте, кіріске берілетін айнымалыларға байланысты.

Сондықтан, дәлелдердің пермутациясы функцияның сипатына әсер етеді.



$$F = F(F_A(X_1, X_2, X_3), F_B(X_2, X_3)) \neq F(F_B(X_2, X_3), F_A(X_1, X_2, X_3))$$

Осылайша, қисынды мағынадағы еркін, еркін түрдегі сызбаларды екі әдіспен құруға болады:

- элементтерді тізбектей қосу;
- элементтердің енгізілуін есептеу.

Логика алгебрасындағы осы екі физикалық әдіс сәйкес келеді:

Суперпозиция принципі (басқа функцияларды функцияға оның дәлелдерінің орнына ауыстыру);

- аргументтерді ауыстыру (функция аргументтерін жазу тәртібін өзгерту немесе бір функция дәлелін басқасына ауыстыру).

Сонымен, күрделі құрылғыны құру және оның жұмысын талдаудың физикалық міндеті логика алгебрасының тиісті функцияларын синтездеудің және талдаудың математикалық есептерімен алмастырылады.

### Логика алгебрасының элементар функциялары

Логика алгебрасының функцияларына қатысты бірнеше синонимдер бар:

- логика алгебрасының функциялары (ЛАФ);
- коммутациялық функциялар;
- Логикалық функциялар;
- екілік функциялар.

Біз осы синонимдердің барлығын қажет болған жағдайда қолданамыз.

Бірнеше дәлелдер жинағын қарастырайық:

$$\langle X_1, X_2, X_3, \dots, X_i, \dots, X_n \rangle$$

және дәлелдердің әрқайсысы басқаларына қарамастан мүмкін болатын екі мәннің біреуін ғана алады деп санаймыз

Әр түрлі жиындардың саны қандай?

$$X_i = \{0, 1\}$$

Біз әр жиынға екілік нөмір береміз:

$$X_1, X_2, \dots, X_n$$

0, 0, ..., 0 нөлдік жиынтық

0, 0, ..., 1 бірінші жиын

0, 0, ..., 1, 0 екінші жиын

.....

1, 1, ..., 1  $(2^n - 1)$ - жиынтығы

Позициялық екілік жүйеде әртүрлі  $X_1, X_2, \dots, X_n$   $n$ -биттік сандардың саны  $2^n$  болатыны анық.

Осы жиындарда  $F(X_1, X_2, \dots, X_n)$  кейбір функциялар анықталды делік және олардың әрқайсысында ол '0' немесе '1' th мәнін алады.

Мұндай функция логика алгебрасының немесе коммутацияның функциясы деп аталады.

'N' дәлелдерінің әртүрлі ауысу функциясының саны қанша?

Себебі Әрбір жиынтықтағы функция '0' немесе '1' мәнін қабылдауы мүмкін, және барлығы  $2^n$  әр түрлі жиындардан тұрады, сондықтан 'n' аргументінің әртүрлі функцияларының жалпы саны:  $2^{(2^n)}$ .

Үздіксіз аргументтің аналитикалық функциясымен салыстырғанда, тіпті бір аргумент үшін де көптеген түрлі функциялар бар.

Дәлелдер саны	1	2	3	4	5	10
Әр түрлі қосқыш саны. функциялары	4	16	256	65536	$\sim 4 \cdot 10^9$	$\sim 10300$

Әр түрлі компьютерлік құрылғыларда ондаған және жүздеген айнымалылар (дәлелдер) бар, сондықтан бір-бірінен ерекшеленетін әр түрлі құрылғылардың саны шексіз болатыны анық.

Сонымен, сіз осы күрделі функцияларды (және, демек, құрылғыларды) қалай құруды үйренуіңіз керек, сонымен қатар оларды талдаңыз.

Неғұрлым күрделі функцияларды синтездеу міндеті - оларды суперпозиция және аргументті ауыстыру операцияларына негізделген қарапайымдар арқылы ұсыну.

Осылайша, олардың негізінде күрделі функцияларды құру үшін алдымен осы қарапайым функцияларды зерттеу қажет.

ЛАФ бір дәлел

Сұрау ЛАФ, оның мәндерін барлық дәлелдер жиынтығына қою керек.

$f_0(x) = 0$ -бірдей НОМ (тұрақты 0)

$f_1(x) = x$ - сәйкестендіру функциясы

$f_2(x) = \bar{x}$  теріске шығару  $x$  (инверсия)

$f_3(x) = 1$ - бірдей бірлік (тұрақты 1)

$x_1, x_2$	$f_0$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$f_0(x_1, x_2) = 0$ - жеке басы нөл (константа 0),

$f_1(x_1, x_2) = x_1 \cdot x_2 = x_1 \& x_2 = x_1 \wedge x_2$  Бұл функция көбінесе логикалық өнім немесе конъюнктура деп аталады;

$f_2(x_1, x_2) = X_1 \Delta X_2$  Тыйым салу  $X_2$

$f_3(x_1, x_2) = x_1$  – қайталану  $x_1$ ;  $f(X_1, X_2) = X_1$

$f_4(x_1, x_2) = X_2 \Delta X_1$  Тыйым салу  $X_1$

$f_5(x_1, x_2) = x_2$ - қайталану  $x_2$ ;

$f_6(x_1, x_2) = x_1 + x_2$ - модуль 2 немесе mod2 қосу (қосу)

$f_7(x_1, x_2) = x_1 \vee x_2$  көбейту немесе логикалық қосу;

$f_8(x_1, x_2) = x_1 \cdot x_2$ - Webb функциясы (Пирс көрсеткі)

$f_9(x_1, x_2) = x_1 x_2$ - эквивалент;

$f_{13}(x_1, x_2) = x_1 \rightarrow x_2$  әсер ету;

$f_{14}(x_1, x_2)$ - Шеффердің соққысы;

$f_{15}(x_1, x_2) = 1$ - сәйкестендіру бірлігі (тұрақты 1)

Логикалық алгебра үшін үш аксиома жарамды:

Коммутация заңы -  $x \vee y = y \vee x$ ,  $x \cdot y = y \cdot x$ ;

Ассоциация заңы -  $(x \vee y) \vee z = x \vee (y \vee z)$ ,  $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ ;

Таралу заңы -  $x \cdot (y \vee z) = x \cdot y \vee x \cdot z$ ,  $x \vee y \cdot z = (x \vee y) \cdot (x \vee z)$ ;

### Дәріс 3. Мультипроцессорлы есептеу жүйесінің жұмысының мақсаты, көлемі және әдістері

*Аннотация:* Бұл дәрісте мультипроцессорлы есептеу жүйесінің түрлері мен түрлері сипатталған. Есептеу жүйесінің шыңы мен нақты жұмысының анықтамалары, сонымен қатар оларды бағалау әдістері келтірілген.

Қазіргі уақытта мультипроцессорлы есептеу жүйелерін (MVS) қолдану аясы үнемі кеңейіп келеді, ол ғылымның, бизнестің және өндірістің әртүрлі салаларында барлық жаңа бағыттарды қамтиды. Кластерлік жүйенің қарқынды дамуы экономиканың нақты секторында мультипроцессорлы есептеулерді қолдануға жағдай жасайды.

Егер дәстүрлі түрде MVS негізінен ғылыми салада қуатты есептеу ресурстарын қажет ететін есептеу мәселелерін шешуде қолданылса, қазіргі уақытта бизнестің қарқынды дамуына байланысты компьютерлік технологияларды және электрондық құжат айналымын қолдануға басты рөлді беретін компаниялар саны күрт өсті. Осыған байланысты транзакцияларды өңдеуге, дерекқорды басқаруға және телекоммуникация қызметтеріне байланысты миссиялық-сыни қосымшалар үшін орталықтандырылған есептеу жүйелерін құру қажеттілігі үнемі артып келеді. Сипатталған жүйелерді қолданудың екі негізгі бағытын бөліп көрсетуге болады: нақты уақыттағы транзакцияны өңдеу (OLTP, on-line транзакция) және шешім қабылдауды қолдау жүйесін құру үшін деректер қоймасын құру (Data Mining, Data Warehousing, шешімдерді қолдау жүйесі).

MVS жетілдірілуімен масштабтың кеңеюімен қатар, өнімділігі жоғары есептеулерді дәстүрлі түрде қолданатын салалардағы міндеттердің күрделілігі мен көбеюі байқалады. Қазіргі уақытта тек қана өте қуатты есептеу ресурстарын қолдану арқылы тиімді шешуге болатын іргелі және қолданбалы проблемалар шеңбері бөлінді. «Үлкен міндеттер» тұжырымдамасымен белгіленген бұл үйірме келесі міндеттерді қамтиды:

- ауа-райының, климаттың және атмосферадағы ғаламдық өзгерістердің болжамдары;

- материалтану;
- жартылай өткізгіш құрылғылардың құрылысы;
- өткізгіштік;
- құрылымдық биология;
- Фармацевтикалық препараттардың дамуы;
- Генетика;
- кванттық хромодинамика;
- астрономия;
- көлік міндеттері;
- гидро және газ динамикасы;
- басқарылатын термоядролық синтез;
- отынды жағу жүйесінің тиімділігі;
- географиялық ақпараттық жүйелер;
- пайдалы қазбаларды барлау;
- мұхиттар туралы ғылым;
- сөйлеуді тану және синтездеу;
- кескінді тану.

Мультипроцессорлық есептеу жүйелері әртүрлі конфигурацияларда болуы мүмкін. MVS ең көп таралған түрлері:

- жоғары сенімділік жүйелері;
- жоғары өнімділікті есептеу жүйелері;
- Көп ағынды жүйелер.

Осы типтегі MVS арасындағы шекаралар белгілі бір дәрежеде бұлдыр болатынына назар аударыңыз, көбінесе жүйеде аталған түрлерден асып түсетін осындай қасиеттер немесе функциялар болуы мүмкін. Сонымен қатар, жалпы мақсаттағы жүйе ретінде қолданылатын үлкен жүйені конфигурациялау кезінде жоғарыда аталған барлық функцияларды орындайтын блоктарды бөлу қажет.

MVS – бұл ақпараттық және есептеу жүйелерінің сенімділігін арттырудың тамаша схемасы. Жеке көріністің арқасында MVS-тің жеке компоненттері немесе компоненттері пайдаланушы үшін ақаулы элементтерді біртіндеп алмастыра алады, тіпті мәліметтер базасы сияқты күрделі қосымшалардың үздіксіздігі мен үзіліссіз жұмысын қамтамасыз етеді.

Катастрофиялық тұрақты шешімдер мультипроцессорлық жүйелік тораптардың жүздеген шақырымдық аралықтарының негізінде құрылады және осындай түйіндер арасындағы деректерді ғаламдық үндестіру тетіктерін қамтамасыз етеді.

Жоғары өнімділікті есептеу үшін MVS параллель есептеулерге арналған. Көптеген операциялардың бір уақытта жүргізілуін қамтамасыз ететін бірнеше қымбат емес процессорлардың параллель жұмысы негізінде орындалған ғылыми есептеулердің көптеген мысалдары бар.

Жоғары өнімді есептеуге арналған MVC әдетте көптеген компьютерлерден жиналады. Мұндай жүйелердің дамуы көптеген компьютерлерді орнату, пайдалану және бір уақытта басқару, сол жүйелік файлға (немесе файлдарға) параллельді және жоғары қол жетімділікке, түйіндер арасындағы интерпроцесс байланысын және параллельді жұмыстарды үйлестіру сияқты мәселелерді үнемі үйлестіруді талап ететін күрделі процесс. режимін таңдаңыз. Бұл мәселелер барлық кластерге операциялық жүйенің бір кескінін беру арқылы оңай шешіледі. Дегенмен, мұндай схеманы жүзеге асыру әрдайым мүмкін емес және әдетте ол тек шағын жүйелер үшін қолданылады.

Көп ағынды жүйелер уақыт өте келе көбейетін (немесе азаятын) бірқатар ресурстарға біртұтас интерфейсті қамтамасыз ету үшін қолданылады. Әдеттегі мысал - веб-серверлер тобы.

Мультипроцессорлық есептеу жүйесінің басты ерекшелігі - оның өнімділігі, яғни. уақыт бірлігімен жүйе жасаған операциялардың саны. Шың мен нақты өнімді ажыратыңыз. Шың дегеніміз бір процессордың берілген машинадағы осындай процессорлардың саны бойынша ең жоғары өнімділікке тең шамасы. Барлық компьютерлік құрылғылар ең өнімді

режимде жұмыс істейді деп болжанады. Компьютердің ең жоғары өнімділігі бір мәнді есептеледі, және бұл сипаттама жоғары өнімділігі жоғары есептеу жүйелерімен салыстырылатын негізгі сипаттама болып табылады. Ең жоғары өнімділік неғұрлым көп болса, пайдаланушы теориялық тұрғыдан тезірек өз мәселесін шеше алады. Ең жоғары көрсеткіштер теориялық және жалпы айтқанда, нақты қолданбаны бастау кезінде қол жетімді емес. Бұл қосымшаның нақты қол жетімділігі қолданбалы бағдарлама жұмыс істейтін машинаның сәулеттік ерекшеліктерімен орындалатын бағдарламалық модельдің өзара әрекеттесуіне байланысты.

Компьютердің жоғары өнімділігін өлшеу үшін екі әдіс бар. Олардың бірі уақыт бірлігінде орындалатын компьютердің нұсқауларына тәуелді. Өлшем бірлігі - бұл әдетте MIPS (секундына миллион нұсқаулық). MIPS-де көрсетілген өнімділік компьютердің өз нұсқауларын орындау жылдамдығын көрсетеді. Бірақ, біріншіден, белгілі бір бағдарламада неше нұсқаулық көрсетілетіні алдын-ала анық емес, екіншіден, әр бағдарламаның өзіндік ерекшелігі бар және бағдарламадан бағдарламаға дейінгі командалардың саны өте өзгеруі мүмкін. Осыған байланысты, бұл сипаттама компьютердің жұмысы туралы ең жалпы идеяны ғана береді.

Өнімділікті өлшеудің тағы бір әдісі - компьютердің уақыт бірлігіне жасаған маңызды операцияларының санын анықтау. Өлшем бірлігі - бұл Flops (Секундына өзгермелі нүктедегі операциялар) - компьютер бір секундта орындайтын өзгермелі нүкте операцияларының саны. Бұл әдіс қолданушы үшін анағұрлым қолайлы, өйткені ол бағдарламаның есептеу күрделілігін біледі және осы сипаттаманы қолдана отырып, қолданушы оны орындау уақытының төменгі бағасын ала алады.

Дегенмен, шыңның өнімділігі идеалды жағдайларда ғана алынады, яғни. барлық құрылғыларды біркелкі жүктеу арқылы жадқа қол жеткізу кезінде қақтығыстар болмаған жағдайда. Нақты жағдайда нақты бағдарламаның жұмысына берілген компьютердің аппараттық және бағдарламалық сипаттамалары әсер етеді, мысалы: процессор құрылымының ерекшеліктері, командалық жүйелер, функционалды құрылғылардың құрамы, енгізу-шығару және компиляторлардың тиімділігі.

Анықтайтын факторлардың бірі жадпен өзара әрекеттесу уақыты болып табылады, ол құрылымымен, көлемімен және жадқа қол жеткізудің ішкі жүйелерінің сәулетімен анықталады. Қазіргі компьютерлердің көпшілігінде көп деңгейлі иерархиялық жад деп аталады, олар жадқа ең тиімді қол жетімділікті ұйымдастырады. Пайдаланылатын деңгейлер - регистрлер мен регистрлер жады, негізгі жад, кэш-жад, виртуалды және қатты дискілер, таспалы роботтар.

Нақты есептер бойынша есептеу жүйесінің жұмысын бағалау үшін белгіленген тесттер жинағы жасалды. Олардың ішіндегі ең танымалы - LINPACK - қатардағы негізгі элементті таңдай отырып, тығыз матрицасы бар сызықтық алгебралық теңдеулер жүйесін шешуге арналған бағдарлама.



LINPACK Top500 - әлемдегі ең қуатты бес жүз компьютердің тізімін жасау үшін қолданылады. Алайда, LINPACK-те айтарлықтай кемшілік бар: бағдарлама параллельді, сондықтан суперкомпьютердің байланыс компонентінің тиімділігін бағалау мүмкін емес.

Қазіргі уақытта модельдік немесе нақты өнеркәсіптік қосымшаларды ұсынатын әр түрлі пәндерден алынған тестілік бағдарламалар кеңінен қолданылады. Мұндай сынақтар компьютердің жұмысын нақты тапсырмалар бойынша бағалауға және нақты қосымшаның көмегімен компьютердің тиімділігі туралы толық ақпарат алуға мүмкіндік береді.

Осы қағидаға негізделген ең көп таралған сынақтар: Liverpool 24 циклдарының жиынтығы (Ливермор Фортран ядросы, LFK) және NAS Parallel Benchmarks (NPB) жиынтығы, ол сұйықтықтың динамикалық есептеушінің есептеуіш бағдарламасының әр түрлі аспектілерін көрсететін екі тест тобын қамтиды. NAS тесттері LINPACK-қа балама болып табылады, өйткені олар салыстырмалы түрде қарапайым және сонымен бірге, мысалы, LINPACK немесе LFK-ге қарағанда айтарлықтай көп есептеулерді қамтиды.

Алайда, тестілеу бағдарламалары өзінің әртүрлілігіне байланысты әртүрлі режимдердегі компьютердің толық бейнесін бере алмайды. Сондықтан мультипроцессорлы есептеу жүйелерінің нақты өнімділігін анықтау міндеті шешілмеген күйінде қалып отыр.

#### **Дәріс 4. Есептеу жүйелерінің архитектурасы. Параллель мәліметтерді өңдеуге арналған архитектуралардың жіктелуі**

*Аннотация:* Бұл дәріс жоғары тиімділік жүйесінің архитектурасы ұғымын анықтайды, нұсқаулық ағымдары мен мәліметтер ағындарының санын ескере отырып, сәулеттің классификациясын қамтамасыз етеді.

Мультипроцессорлы есептеу жүйелері туралы неғұрлым толық көрініс беру үшін жоғары өнімділіктен басқа, басқа да ерекшеліктерді атап өту керек. Біріншіден, бұл өнімділікті арттыруға бағытталған ерекше архитектуралық шешімдер (векторлық операциялармен жұмыс істеу, процессорлар арасында жылдам хабар алмасуды ұйымдастыру немесе мультипроцессорлық жүйелерде ғаламдық жадыны ұйымдастыру және т.б.).

Жоғары өнімді жүйенің архитектурасы туралы түсінік өте кең, өйткені архитектурада жүйеде қолданылатын мәліметтерді параллель өңдеу әдісі, жадыны ұйымдастыру, процессорлар арасындағы байланыс топологиясы және жүйенің арифметикалық амалдарды орындау әдісі ретінде түсінуге болады. Көптеген сәулеттерді жүйелеуге талпыныстар алғаш рет 60-шы жылдардың соңында жасалды және осы күнге дейін жалғасуда.

1966 жылы М.Флинн компьютерлік жүйелер архитектурасын жіктеуге өте ыңғайлы тәсілді ұсынды. Ол ағын тұжырымдамасына негізделді, ол процессор өндеген элементтердің, командалардың немесе деректердің реттілігі ретінде түсініледі. Тиісті жіктеу жүйесі оқыту ағындары мен деректер ағындарының санын ескеруге негізделген және төрт сәулет класын сипаттайды:

SISD = Single Instruction Single Data

MISD = Multiple Instruction Single Data

SIMD = Single Instruction Multiple Data

MIMD = Multiple Instruction Multiple Data

SISD (single instruction stream / single data stream) – бір командалық ағын және бірыңғай деректер ағыны. Бұл сыныпқа бір орталық процессоры бар, тізбектелген орындалатын нұсқаулықтардың тек бір ағынын өңдей алатын сериялық компьютерлік жүйелер кіреді. Қазіргі уақытта барлық тиімділігі жоғары жүйелерде бірнеше орталық процессор бар, бірақ олардың әрқайсысы әртүрлі мәліметтер кеңістігінде жұмыс істейтін SISD жүйелерінің осындай жүйелік кешендерін құрайтын өзара байланысты емес ағындарды орындайды.

MISD (бірнеше нұсқаулық ағындары / жалғыз деректер ағыны) - бірнеше нұсқаулық ағындары және бірыңғай мәліметтер ағыны. Теориялық тұрғыдан алғанда, машинаның бұл түрінде көптеген нұсқаулар бірыңғай деректер ағынында орындалуы керек. Осы уақытқа дейін осы классқа енетін бірде-бір нақты машина жасалған жоқ.

SIMD (бір нұсқаулық ағын / бірнеше деректер ағыны) - бір командалық ағын және бірнеше мәліметтер ағыны. Бұл жүйелерде әдетте 1024-тен 16384-ге дейінгі процессорлардың үлкен саны бар, олар қатты конфигурациядағы әртүрлі мәліметтерге қатысты бірдей нұсқауларды орындай алады. Бір нұсқаулық көптеген мәліметтер элементтеріне параллель орындалады. SIMD машиналарының мысалдары - CPP DAP, Gamma II және Quadrics Apemille. SIMD жүйелерінің тағы бір ішкі класы - векторлық компьютерлер. Векторлық компьютерлер осындай массивтердің жекелеген элементтерін скалярлық машиналар өңдейтін ұқсас мәліметтер массивтерін басқарады.

MIMD (бірнеше нұсқаулық ағыны / бірнеше мәліметтер ағыны) - бірнеше нұсқаулық ағыны және бірнеше мәліметтер ағыны. Бұл машиналар бір мезгілде әртүрлі ағындарда бірнеше нұсқаулық ағындарын орындайды. Жоғарыда аталған SISD мультипроцессорлы машиналардан айырмашылығы, командалар мен мәліметтер бір-бірімен байланысты, өйткені олар бір тапсырманың әртүрлі бөліктерін ұсынады.

Есептеу жүйелерінің архитектурасын классификациялау белгілі бір архитектураның ерекшеліктерін түсіну үшін қажет, бірақ MVS құру кезінде оған сенім арту жеткіліксіз, сондықтан әртүрлі компьютерлік

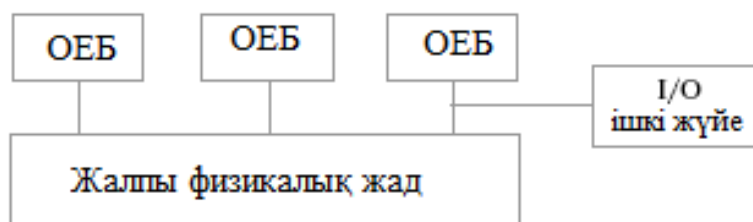
архитектуралармен және онымен байланысты егжей-тегжейлі жіктеу енгізілуі керек. жабдық.

### **Дәріс 5. Есептеу жүйелерінің архитектурасы. SMP және MPP сәулеті. Гибридтік сәулет (NUMA). Көп деңгейлі иерархиялық жадтың үйлесімділігін ұйымдастыру**

*Аннотация:* Бұл дәрісте есептеу жүйелерінің жаппай параллельді және гибридік архитектурасы, симметриялы мультипроцессордың сипаттамасы берілген. Әр архитектураның негізгі артықшылықтары мен кемшіліктері, сонымен қатар сәйкес бағдарламалау парадигмалары келтірілген.

#### **SMP архитектурасы**

SMP (симметриялық мультипроцессинг) - симметриялық мультипроцессорлық сәулет. SMP архитектурасы бар жүйелердің басты ерекшелігі - барлық процессорлар ортақ физикалық жадының болуы.



Сурет 5.1. SMP архитектурасының схемалық көрінісі

Жад, атап айтқанда, процессорлар арасында хабарлама алмасуға қызмет етеді, ал оған қол жеткізген кезде барлық есептеу құрылғылары барлық жад ұяшықтары үшін бірдей адреске ие. Сондықтан SMP архитектурасы симметриялы деп аталады. Соңғы жағдай басқа есептеу құрылғыларымен өте тиімді мәліметтер алмасуға мүмкіндік береді. SMP жүйесі жоғары жылдамдықты автобустың негізінде жасалған (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), олардың ұяшықтарына типтердің функционалды блоктары қосылған: процессорлар (процессорлар), кіріс / шығыс ішкі жүйесі (I / O) және т.б. I / O модульдері баяу автобустарды қолданады (PCI, VME64).

SMP жүйелерінің негізгі артықшылықтары:

- бағдарламалаудың қарапайымдылығы мен әмбебаптығы. SMP архитектурасы қосымшаны құру кезінде пайдаланылатын бағдарламалау моделіне шектеулер қоймайды: параллель тармақ моделі, әдетте, барлық процессорлар бір-бірінен тәуелсіз жұмыс жасағанда қолданылады.

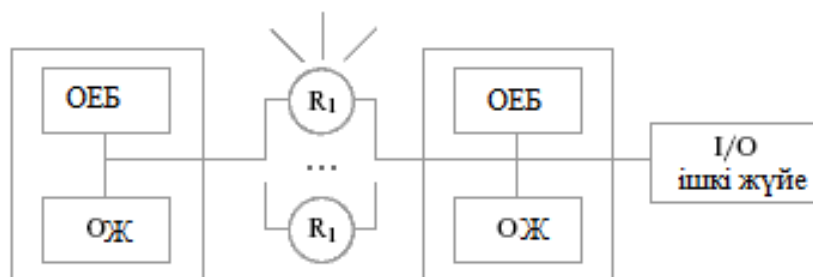
- жұмыстың қарапайымдылығы. Әдетте SMP жүйелері ауаны салқындатуға негізделген ауаны пайдаланады, бұл оларды ұстауды жеңілдетеді;

- салыстырмалы түрде төмен баға.

Кемшіліктері: ортақ жад жүйесі жақсы мөлшерлемейді.

SMP-жүйелердің бұл кемшілігі оларды шынымен перспективалы деп санауға мүмкіндік бермейді. Нашар масштабтаудың себебі қазіргі уақытта автобус тек бір транзакцияны орындай алады, нәтижесінде бірнеше процессорлар ортақ физикалық жадтың бірдей аймақтарына қол жеткізе отырып, қақтығыстарды шешуде проблемалар туындайды. Есептеу элементтері бір-біріне кедергі жасай бастайды. Мұндай қақтығыс туындаған кезде, бұл байланыс жылдамдығына және есептеу элементтерінің санына байланысты. MPP-архитектура

MPP (massive parallel processing) – массивтік сәулет. Бұл архитектураның басты ерекшелігі - жадтың физикалық бөлінуі. Бұл жағдайда жүйе процессордан, операциялық жадтың жергілікті банкінен (OP), байланыс процессорларынан (маршрутизаторлардан) немесе желілік адаптерлерден, кейде қатты дискілерден және / немесе басқа енгізу / шығару құрылғыларынан тұратын жеке модульдерден тұрады. Шын мәнінде, мұндай модульдер - бұл толық мүмкіндігі бар компьютерлер (5.2 суретті қараңыз). Осы модульден OP банкке қол жетімділік тек сол модульден процессорларға (процессорларға) қол жетімді.



Сурет 5.2. Жадтың ортақ архитектурасының схемалық көрінісі

Бөлек жады бар жүйелердің басты артықшылығы - оның жақсы ауқымдылығы: SMP жүйелерінен айырмашылығы, бөлек жады бар машиналарда әр процессор тек өзінің жергілікті жадына қол жеткізе алады, сондықтан процессорларды сағат үнімен синхрондаудың қажеті жоқ. Бүгінгі күні барлық дерлік рекордтар бірнеше мың процессорлардан (ASCI Red, ASCI Blue Pacific) тұратын осындай архитектураның машиналарында жинақталған. Кемшіліктері:

- ортақ жадтың жоқтығы процессорлармен алмасу жылдамдығын едәуір төмендетеді, өйткені процессорлар арасында алмасуға арналған

мәліметтерді сақтаудың ортақ ортасы жоқ. Процессорлар арасындағы хабарламаны жүзеге асыру үшін арнайы бағдарламалау әдісі қажет;

- әр процессор жергілікті жад банкінің шектеулі мөлшерін ғана қолдана алады;
- Көрсетілген архитектуралық кемшіліктерге байланысты жүйелік ресурстарды барынша пайдалану үшін көп күш қажет. Бұл бөлек жады бар жаппай параллельді жүйелерге арналған бағдарламалық қамтамасыз етудің жоғары бағасын анықтайды.

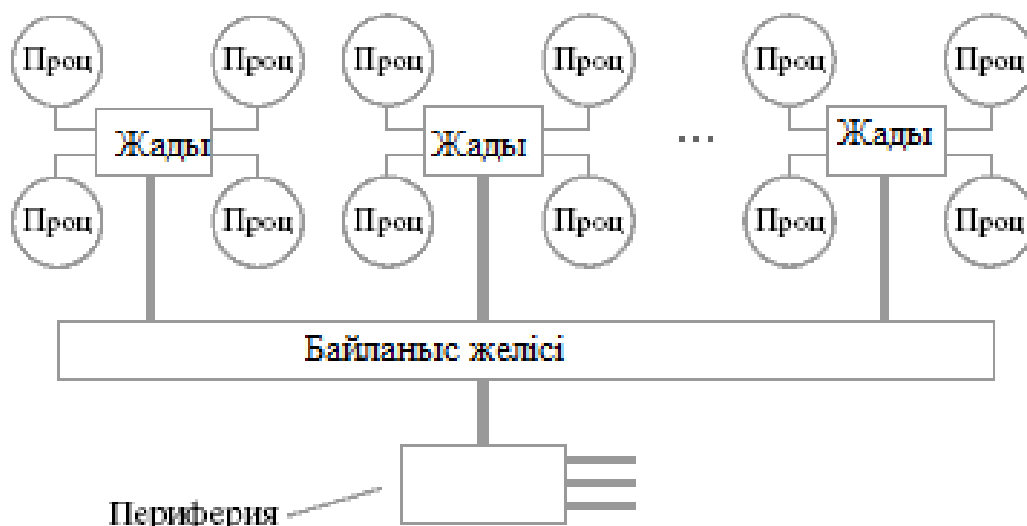
Бөлінген жад жүйелері - MVS-1000, IBM RS / 6000 SP, SGI / CRAY T3E, ASCI, Hitachi SR8000, Parsytec суперкомпьютерлері.

SGI-дің ең жоғарғы өнімі 1200 Mflops (CRAY T3E-1200) болатын Dec Alpha 21164 процессорларына негізделген CRAY T3E-нің соңғы сериясы 2048 процессорға дейін тарата алады.

MPP жүйелерімен жұмыс кезінде олар Massive Passing Programming Paradigm деп аталады - мәліметтерді берумен (MPI, PVM, BSPlib) бағдарламалау парадигмасы.

NUMA гибридік архитектурасының басты ерекшелігі (жадқа біркелкі емес қол жетімділік) - жадқа гетерогенді кіру.

Гибридік архитектура ортақ жады бар жүйелердің артықшылықтарын және ортақ жады бар жүйелердің салыстырмалы арзандығын біріктіреді. Бұл архитектураның мәні есте сақтаудың арнайы ұйымында, атап айтқанда: жад физикалық түрде жүйенің әртүрлі бөліктері бойынша таратылады, бірақ логикалық түрде бұл жалпы, сондықтан қолданушы бір мекенжай кеңістігін көреді. Түйіндер көбелектің типтік желісімен қосылады:



Сурет 5.3. Гибриді желілік компьютердің блок-схемасы

Гибриді сәулет идеясын алғаш Стив Воллох ұсынған, ол оны Exemplar сериялы жүйесінде енгізген. Воллох нұсқасы - сегіз SMP түйінінен тұратын

жүйе. HP бұл идеяны сатып алды және SPP сериялы суперкомпьютерлерде іске асырды. Сеймур Р.Край идеяны қабылдады және жаңа элементті - кэш үйлесімділігін қамтамасыз ете отырып, біркелкі емес жадқа қол жетімділікті білдіретін cc-NUMA (Cache Coherent Non-Uniform Memory Access) архитектурасын құра отырып, үйлесімді кэштi қосты. Ол оны шығу сияқты жүйелерде іске асырды.

#### **Көп деңгейлі иерархиялық жадтың үйлесімділігін ұйымдастыру**

Кэш үйлесімділік ұғымы барлық орталық процессорлар кез-келген уақытта бірдей айнымалылардың бірдей мәндерін алатындығымен сипатталады. Шынында да, кэш жады тұтас алғанда бүкіл мультипроцессорлық жүйеге емес, жеке компьютерге тиесілі болғандықтан, бір компьютердің кэшіне түсетін мәліметтер басқасына қол жетімді болмауы мүмкін. Бұған жол бермеу үшін процессор кэшінде сақталған ақпаратты синхрондау керек.

Кэш үйлесімділігін қамтамасыз етудің бірнеше нұсқалары бар:

- кез-келген орталық процессорға берілетін айнымалыларды қадағалайтын және қажет болған жағдайда осындай айнымалылардың өздерінің көшірмелерін өзгерте алатын кнопки автобусының хаттамасын бақылау механизмін қолданыңыз;

Жадтың барлық пайдаланылған айнымалы көшірмелерінің сенімділігін бақылауға арналған арнайы бөлімді бөлу.

Ең танымал cc-NUMA архитектуралық жүйелері: SCA конфигурациясындағы HP 9000 V-класс, SGI Origin3000, Sun HPC 15000, IBM / Sequent NUMA-Q 2000. Бүгінгі таңда cc-NUMA жүйелеріндегі процессорлардың максималды саны 1000 (сериядан асуы мүмкін) Шығу3000). Әдетте бүкіл жүйе SMP сияқты бірыңғай ОЖ-мен жұмыс істейді. Жүйенің динамикалық «бөлінуінің» нұсқалары жүйенің бөлек «бөлімдері» әртүрлі ОЖ-де жұмыс істеген кезде де мүмкін. NUMA-жүйелермен, сондай-ақ SMP-мен жұмыс істеу кезінде олар ортақ жад парадигмасын қолданады.

### **Дәріс 6. Есептеу жүйелерінің архитектурасы. ПВП архитектурасы. Кластер сәулеті**

*Аннотация:* Бұл дәрісте векторлық процессорлармен (ПВП) параллельді архитектураның сипаттамасы, сонымен қатар мультипроцессорлы есептеу жүйелерінің кластерлік архитектурасы берілген. Әр архитектураның негізгі артықшылықтары мен кемшіліктері, сонымен қатар сәйкес бағдарламалау парадигмалары келтірілген. Сәулет деректері бар ең танымал есептеу жүйелері сипатталған.

PVP (Parallel Vector Process) - векторлық процессорлармен параллельді архитектура. ПВП-жүйелерінің басты ерекшелігі - конвейердің функционалды құрылғыларында тиімді орындалатын дербес векторлардың бірдей түрін өңдеуге нұсқаулар беретін арнайы векторлық-конвейер процессорларының болуы. Әдетте, бірнеше бірнеше процессорлар (1-16) мультипроцессорлық конфигурациялар аясында бір уақытта ортақ жадпен (SMP-ге ұқсас) жұмыс істейді. Бірнеше түйіндерді коммутатордың көмегімен біріктіруге болады (MPP-ге ұқсас). Сәулет өнерінің ең танымал үш машинасы:

CRAY X1, SMP -архитектура. Стандартты конфигурациядағы жүйенің ең жоғары өнімділігі ондаған терафлоп болуы мүмкін.



Сурет 6.1. CRAY SV-2

NEC SX-6, NUMA -архитектура. Жүйенің ең жоғары өнімділігі 8 TFlops-қа жетуі мүмкін, бір процессордың өнімділігі - 9,6 Gflops. Жүйе бір операциялық жүйенің кескіні бар 512 процессорға таралады. Fujitsu-VPP5000 (vector parallel processing), MPP -архитектура. Бір процессордың өнімділігі - 9,6 Gflops, жүйенің ең жоғары өнімділігі - 1249 Гллуп, ал жадтың максималды көлемі - 8 ТБ.



Сурет 6.2. Fujitsu-VPP5000

ПВП жүйелеріндегі бағдарламалау парадигмасы ілмектерді векторландыруды (бір процессордың тиімді жұмысына қол жеткізу үшін)

және олардың параллелизациясын (бірнеше процессорды бір қосымшамен бір уақытта жүктеу үшін) қамтамасыз етеді.

Іс жүзінде келесі процедуралар ұсынылады:

- тапсырманы матрица формасына аудару үшін қолмен векторлаңыз. Сонымен қатар, вектордың ұзындығына сәйкес матрицаның өлшемдері көбейтілген 128 немесе 256 болуы керек;
- виртуалды кеңістіктегі векторлармен жұмыс істеу, қатардағы қажетті функцияны кеңейту және 128 немесе 256 сериялары мүшелерінің санын қалдыру.

Физикалық жады үлкен болғандықтан (терабайттың фракциялары), тіпті PVC жүйелеріндегі нашар векторланған тапсырмалар скалярлық процессорлары бар машиналарда тезірек шешіледі.

Кластер екі немесе одан да көп компьютерлерден тұрады (көбінесе тораптар деп аталады), олар автобус архитектурасына немесе коммутаторға негізделген желілік технологияларды қолданумен біріктіріліп, пайдаланушыларға бірыңғай ақпараттық және есептеу ресурсы ретінде ұсынылады. Кластерлік тораптар ретінде серверлерді, жұмыс станцияларын және тіпті қарапайым дербес компьютерлерді таңдауға болады. Кластерлеу компьютерлік жүйенің әртүрлі деңгейлерінде, оның ішінде аппараттық, операциялық жүйелерде, қызметтік бағдарламаларда, басқару жүйелерінде және қосымшаларда орын алуы мүмкін. Кластер технологиясымен біріктірілген жүйенің деңгейлері неғұрлым көп болса, кластердің сенімділігі, масштабталуы және басқарылуы соғұрлым жоғары болады.

### **Кластерлер түрлері**

Сабақтарға шартты түрде бөлуді Джазек Радаевский мен Дуглас Адлайн ұсынды:

**Класс I.** Машиналар класы толығымен стандартты бөлшектерден тұрады, оларды көптеген компьютерлік жабдықтаушылар сатады (төмен баға, техникалық қызмет көрсету оңай, аппараттық құрал-жабдықтар әртүрлі көздерден алуға болады).

**Класс II.** Жүйеде эксклюзивті немесе өте кең таралмаған бөліктер бар. Осылайша өте жақсы өнімділікке қол жеткізуге болады, бірақ қымбатырақ болады.

Жоғарыда айтылғандай, кластерлер әртүрлі конфигурацияларда болуы мүмкін. Кластерлердің ең көп кездесетін түрлері:

- жоғары сенімділік жүйелері;
- жоғары өнімділікті есептеу жүйелері;
- Көп ағынды жүйелер.

Осы типтегі кластерлердің шекаралары белгілі бір дәрежеде бұлдырланып, кластердің осы типтер шеңберінен шығатын қасиеттері немесе функциялары болуы мүмкін екенін ескеріңіз. Сонымен қатар, жалпы мақсаттағы жүйе ретінде пайдаланылатын үлкен кластерді



конфигурациялау кезінде жоғарыда аталған барлық функцияларды орындайтын блоктарды бөлу қажет.

Жоғары өнімді есептеу кластерлері параллельді есептеулерге арналған. Бұл кластерлер әдетте көптеген компьютерлерден жиналады. Мұндай кластерлердің дамуы - бұл әр кезең сайын көптеген компьютерлерді орнату, пайдалану және бір уақытта басқару сияқты мәселелерді үйлестіруді, бірдей жүйелік файлға (немесе файлдарға) параллельді және жоғары қол жетімділікке техникалық талаптарды, түйіндер арасындағы процессаралық байланысты және үйлестіруді қажет ететін күрделі процесс. қатар жұмыс жасау. Бұл мәселелер барлық кластерге операциялық жүйенің бір кескінін беру арқылы оңай шешіледі. Алайда, мұндай схеманы жүзеге асыру әрдайым мүмкін емес және әдетте ол тек үлкен емес жүйелер үшін ғана қолданылады.

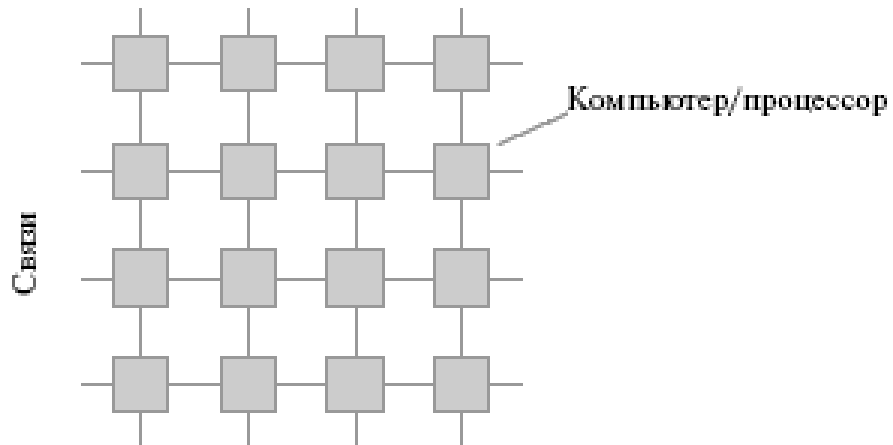
Көп ағынды жүйелер уақыт өте келе көбейетін (немесе азаятын) бірқатар ресурстарға біртұтас интерфейсті қамтамасыз ету үшін қолданылады. Әдеттегі мысал - веб-серверлер тобы.

1994 жылы Томас Стерлинг пен Дон Бекер қайталанатын арналары бар 10 Мбит / с Ethernet желісіне қосылған 16 Intel DX4 процессорларының кластерін құрды. Олар оны ескі эпопеяның атауымен «Беовульф» деп атады. Кластер NASA Goddard ғарыштық ұшу орталығында «Жер және ғарыш туралы ғылым» жобасының қажетті есептеу ресурстарын қолдау үшін пайда болды. Дизайн жұмыстары тез арада Веовульф жобасы ретінде белгілі болды. Жоба параллельді есептеулерде сәтті қолданыла алатын мультипроцессорлық архитектураны сипаттайтын параллель кластерлік компьютерлерді құрудың жалпы тәсілінің негізі болды. Веовульф кластері, әдетте, бір сервер торабынан (әдетте басты түйін деп аталады), сондай-ақ стандартты компьютерлік желі арқылы қосылған бір немесе бірнеше бағынышты (есептеу) түйіндерден тұратын жүйе.

### **Кластерлік жүйеде процессорлардың байланыс желісін құру мәселелері**

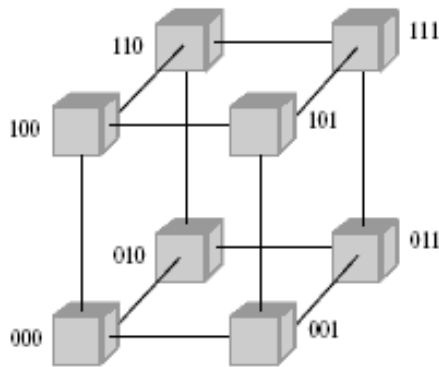
Кластерлік жүйенің архитектурасы (процессорларды бір-бірімен байланыстыру тәсілі) онда қолданылатын процессорлар түріне қарағанда көбінесе оның өнімділігін анықтайды. Мұндай жүйенің жұмысына әсер ететін маңызды параметр - бұл процессорлар арасындағы қашықтық. Сонымен, 10 дербес компьютерді біріктіріп, біз жоғары өнімді есептеулер жүйесін аламыз. Мәселе стандартты құралдарды бір-бірімен байланыстырудың тиімді әдісін табу болып табылады, өйткені әр процессордың өнімділігін 10 есе арттырған кезде жүйенің жалпы өнімділігі 10 есе өспейді.

Мысалы, симметриялық 16 процессорлық жүйені құру міндетін қарастырыңыз, онда барлық процессорлар тең болады. Табиғи болып жалпақ тор байланысы көрінеді, мұнда сыртқы ұштары сыртқы құрылғыларды қосу үшін қолданылады.



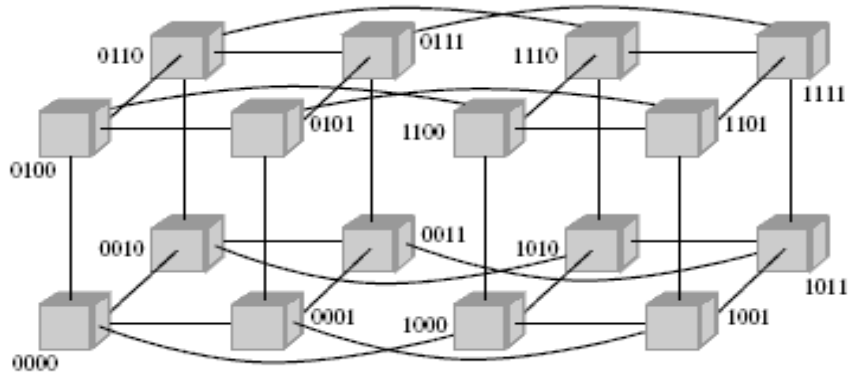
Сурет 6.3. Жазық тор түрінде процессорды қосу жоспары

Қосылымның бұл түрімен процессорлар арасындағы ең үлкен қашықтық 6-ға тең болады (ең жақын процессорды ең алыс бөлетін процессорлар арасындағы қосылыстар саны). Теория көрсеткендей, егер жүйеде процессорлар арасындағы ең үлкен қашықтық 4-тен көп болса, онда мұндай жүйе тиімді жұмыс істей алмайды. Сондықтан 16 процессорды бір-біріне жалғаған кезде жалпақ тізбек іс жүзінде мүмкін емес.

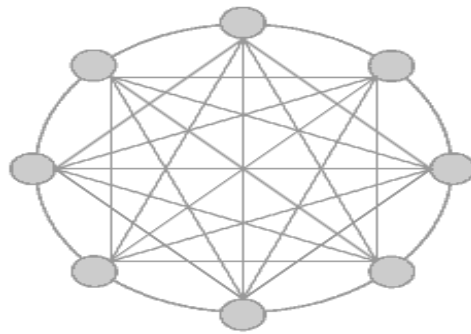


Сурет 6.4. Байланыс топологиясы, үшөлшемді гиперкуба

Гиперкуб архитектурасы екінші тиімді, бірақ ең көрнекі. Байланыс желілерінің басқа топологиялары да қолданылады: үш өлшемді торус, сақина, жұлдыз және басқалар.

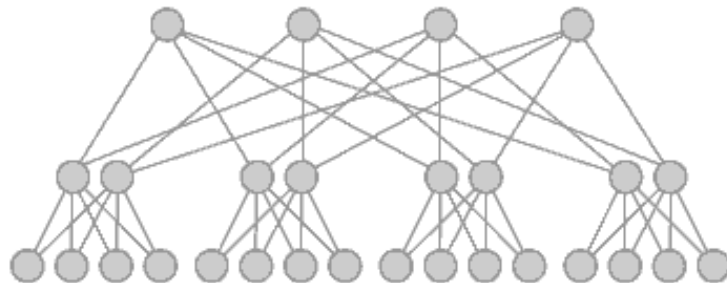


Сурет 6.5. Байланыс топологиясы, 4 өлшемді гиперкуба



Сурет 6.6. Хордаль сақинасының сәулеті

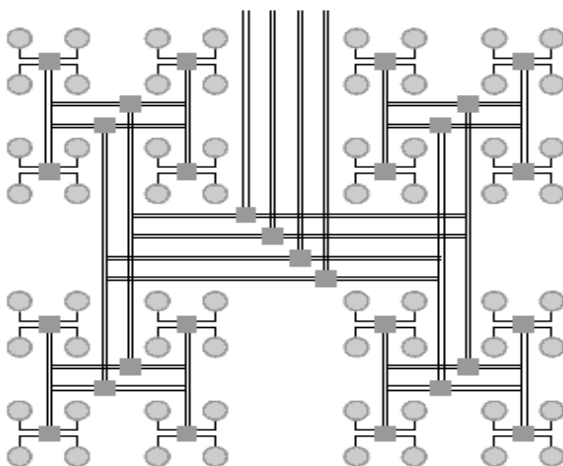
Ең тиімдісі - «қалың ағаш» (майлы ағаш) топологиясы бар сәулет. «Майлы ағаш» (гипертрофиялық) архитектураны 1985 жылы Чарльз Э.Лейсерсон ұсынған. Процессорлар ағаштың жапырақтарында локализацияланған, ал ағаштың ішкі түйіндері ішкі желіде орналасқан. Төменгі желілер жоғары деңгейге әсер етпестен бір-бірімен байланыса алады.



Сурет 6.7. «Майлы ағаш» кластерлік сәулеті

Процессорларды бір-біріне қосу әдісі онда қолданылатын процессорлардың түріне қарағанда кластердің өнімділігіне әсер

ететіндіктен, арзан компьютерлердің санынан гөрі арзан компьютерлердің көп санынан жүйені құру орынды болады. Кластерлер, әдетте, жұмыс станциялары үшін стандартты, көбінесе еркін таратылатын (Linux, FreeBSD) операциялық жүйелерді, параллель бағдарламалауды және жүктемелерді теңестіруді қолдау үшін арнайы құралдармен бірге пайдаланады.



Сурет 6.8. «Май ағашы» кластерлік архитектурасы  
(алдыңғы диаграмманың жоғарғы көрінісі)

## Дәріс 7. Байланыс құралдарын құру принциптері

*Аннотация:* Дәрісте SCI және Myrinet интерфейсі негізінде коммуникациялық медиа құрудың принциптері мен мысалдары сипатталған.

Жалпы алғанда компьютерлік архитектураны компьютерлерді бір-бірімен, жадымен және сыртқы құрылғылармен байланыстыру әдісі ретінде анықтауға болады. Бұл байланысты жүзеге асыру әртүрлі жолдармен жүруі мүмкін. Мұндай қосылудың нақты іске асырылуы компьютерлік байланыс ортасы деп аталады. Қарапайым енгізулердің бірі - процессорлар мен жады қосылатын ортақ шинаны пайдалану.

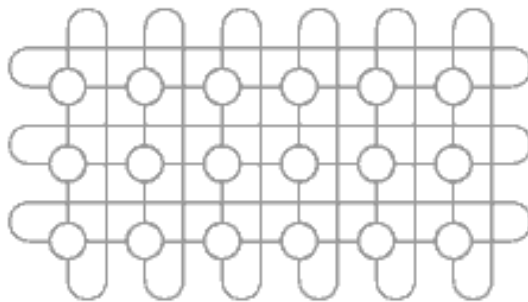
Бұл жағдайда алынған байланыс жүйелерінің әртүрлі конфигурациялары мүмкін. Сонымен, Cray T3D / T3E жанұясының компьютерлерінде барлық процессорлар арнайы жылдамдықты арналармен үшөлшемді торуста біріктірілді, онда әрбір есептеу түйіні алты көршімен тікелей байланыс орнатылды. IBM SP / 2 компьютерлерінде процессордың өзара әрекеті иерархиялық ауыстырып қосу жүйесі арқылы жүреді, ол сонымен қатар әр процессорды басқаларына қосу мүмкіндігін береді. Бұл ерекше ерекше шешімдер компьютерлердің бағасын айтарлықтай арттырады.

Ethernet негізіндегі байланыстарды қолдану Xerox әзірлеген әдіс әлдеқайда қарапайым және арзан болды. Бастапқыда 10 мегабиттік тұрақты желі қолданылды, содан кейін олар Fast Ethernet-ті, ал жақында, кейде Gigabit Ethernet-ті қолдана бастады. Бірақ Fast Ethernet жоғары кідіріспен сипатталады (деректерді беру кідірісі), ол 160-180 микросекундқа бағаланады, және гигабиттік Ethernet өзінің қымбаттығымен ерекшеленеді. Сонымен қатар, ол нүкте-нүктені қосқан кезде ғана тиімді, бірнеше түйінді қосқанда оның тиімділігі күрт төмендейді, ал 5-6 түйінді қосқан кезде ол тіпті жылдам Ethernet-тен де асып кетпейді. Сондықтан мультипроцессорлы есептеу жүйелерін құру кезінде көбінесе SCI, Myrinet немесе Raceway технологияларына артықшылық беріледі.

SCI интерфейсі негізінде байланыс ортасын құру мысалдары

SCI (Scalable Coherent Interface) 1992 жылы стандарт ретінде қабылданды (ANSI/IEEE Std 1596-1992). Ол қысқа кідіріс уақытымен жоғары беріліс жылдамдығына қол жеткізуге арналған және сонымен қатар көптеген блоктардан тұратын жүйелерді құруға мүмкіндік беретін кеңейтілген архитектураны ұсынады. SCI - бұл SCI хостында орналастырылған кэш үйлесімділігін таратылған каталогтың механизмі арқылы қамтамасыз ететін автобус пен локальдық желі жиынтығы, ол бөлектелген ортақ жад моделінде қашықтағы деректерге қол жеткізу құнын жасыру арқылы өнімділікті жақсартады.

SCI-нің тағы бір артықшылығы - өткізу қабілеттілігін арттыруды қамтамасыз ететін RISC сияқты қарапайым протоколдарды қолдану. SCI адаптері бар тораптар қосылуға немесе сақинада қосылуға болады. Әдетте әр түйін екі сақинаға енеді (7.1-сурет).



Сурет 7.1. SCI желісінің кластерлік түйін матрицасы

НІРРІ-ден айырмашылығы, бұл технология динамикалық трафикпен жұмыс істеу үшін оңтайландырылған, бірақ үлкен деректер блоктарымен жұмыс істегенде тиімсіз болуы мүмкін. Деректерді жіберу протоколы кепілдендірілген жеткізуді және аяқталудың болмауын қамтамасыз етеді.

SCI-дің дәстүрлі қолданылуы мультипроцессорлық жүйелердің коммуникациялық ортасы болып табылады. Осы технологияның негізінде, атап айтқанда, Siemens-тен келген hpcLine сериялы компьютерлер немесе

бұрын Sequent ретінде белгілі IBM-ден NUMA-Q модульдік серверлері жасалған.

Дельфиннің модульдік SCI коммутаторлары тұтынушыларға Windows NT / 2000 / XP, Linux, Solaris, VxWorks, LynuxWorks және NetWare жүйелерінде стандартты аппараттық және бағдарламалық жасақтаманы қолдана отырып, кластерлік кластерлік шешімдерді құруға мүмкіндік береді.

Кесте 7.1. SCI технологиясы

Жабдық өндірушілер	Dolphin Interconnect шешімдері және басқалары
Өнімділік көрсеткіштері	Дельфин өнімдері үшін: ең жоғары өткізу қабілеті - 667 Мбит / с. Аппараттық кідіріс - 1,46 мкс
Дельфин өнімдері үшін: ең жоғары өткізу қабілеті - 667 Мбит / с. Аппараттық кідіріс - 1,46 мкс	Драйверы для Linux, Windows NT, Solaris. ScaMPI – реализация MPI компании Scal Computer SCI негізіндегі жүйелер үшін. SISCO API - төменгі деңгейлі бағдарламалау интерфейсі
Пікірлер	SCI (ANSI / IEEE 1596-1992) - стандартталған технология. Стандартты желілік ортаға қосымша, SCI жад пен кэш үйлесімділік жүйелерінің құрылысын қолдайды. SCI байланыс технологиясы SCALI Computer кластерлік жүйелеріне, Siemens компаниясының hrcLine тұқымдас жүйелеріне, сонымен қатар Data General and Sequent cc-NUMA серверіне негізделген. SCI технологиясы HP / Convex Exemplar X класты жүйелеріндегі гипер тораптарды қосу үшін қолданылды.

### **Байланыс ортасы MYRINET**

Myrinet желілік технологиясын Myricom енгізді, ол өзінің коммуникациялық технологиясын алғаш 1994 жылы енгізген және бүгінде бүкіл әлемде 1000-нан астам қондырғыларға ие. Миринет технологиясы тораптар мен коммутациялық порттардың арасындағы байланыс ұзындығы шектеулі көп портты қосқыштарды қолдануға негізделген. Миринеттегі

түйіндер бір-біріне коммутатор арқылы қосылады (128 портқа дейін). Байланыс желілерінің максималды ұзақтығы нақты іске асырылуына байланысты өзгереді.

Коммутаторлар көмегімен қосылған Ethernet сегменттеріне ұқсас коммутацияланған желі ретінде Myrinet бір уақытта бірнеше пакетті жібере алады, олардың әрқайсысы 2 Гб / с жылдамдықпен келеді. Орташа ортаны бөлісетін қосылмаған Ethernet және FDDI желілерден айырмашылығы, Myrinet желісінің жалпы өткізу қабілеті машиналар санына қарай артады.

Кесте 7.2. Миринет технологиясы

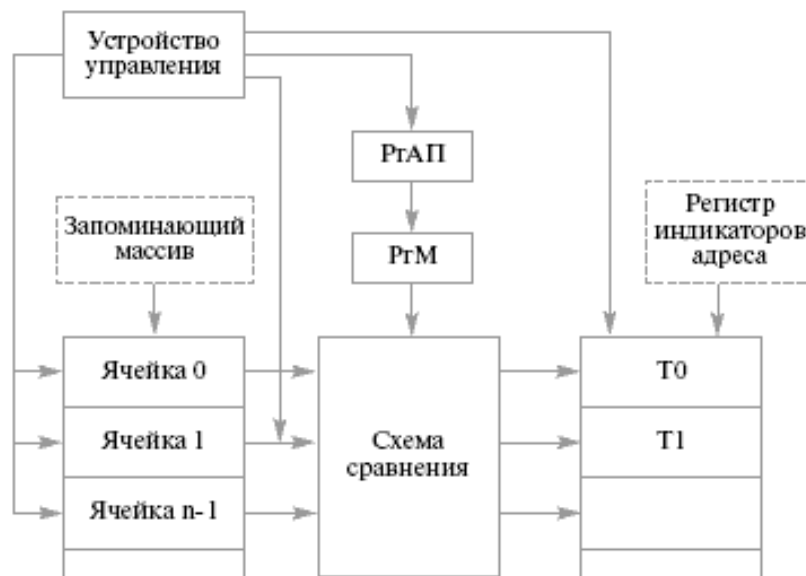
Жабдық өндірушілер	Myricom
Өнімділік көрсеткіштері	Пиковая пропускная способность – 2 Гбит/с, полный дуплекс. Латентность – порядка 4 мксек.
Бағдарламалық қамтамасыз ету	Жүргізушілер үшін Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) және Tru64 UNIX. GM – төменгі деңгейдегі бағдарламалау интерфейсі. HPVM пакеттері (MPI-FM, Myrinet-ке MPI енгізу), VIP-MPI және т.б.
Пікірлер	Миринет - бұл ашық стандарт. Myricom салыстырмалы түрде арзан бағамен желілік жабдықтардың кең таңдауын ұсынады. Физикалық қабат SAN (System Area Network), LAN (CL-2) және талшықты қолдайды. Myrinet технологиясы желінің масштабталуының кең нұсқаларын ұсынады және қазір жоғары өнімді кластерлерді құруда жиі қолданылады.

Физикалық деңгейде Миринет сілтемелері 9 өткізгіштен тұрады: ақпаратты беру үшін 8 бит пайдаланылады, ол тоғызыншы биттің күйіне байланысты мәліметтер байтына немесе басқару сипатына қарай түсіндіріледі; сонымен қатар әр сілтемеде ағынды бақылау және қателерді басқару қарастырылған.

## Дәріс 8. Жоғары өнімді процессорларды ұйымдастыру жолдары. Ассоциативті процессорлар. Конвейерлік процессорлар. Матрицалық процессорлар

*Аннотация:* Бұл дәрісте ассоциативті, конвейерлі және матрицалық процессорлардың математикалық негіздері, ұйымдастыру әдістері және дизайн ерекшеліктері қарастырылады.

Қазіргі уақытта қолданыстағы қосымшалар алгоритмдері, бағдарламалық қамтамасыз ету және бағдарламалық қамтамасыз ету негізінен мекен-жай бойынша деректерді өңдеуге бағытталған. Деректер шектеулі форматтағы форматтар түрінде ұсынылуы керек (мысалы, массивтер, тізімдер, жазбалар), деректер элементтері арасындағы қатынастар құрылымы жад элементтерінің мекен-жайларына бағыттағыштарды қолдана отырып нақты жасалуы керек, осы деректерді өңдеу кезінде мәліметтерге қол жеткізуді қамтамасыз ететін операциялар жиынтығы жасалуы керек. бағыт белгілері. Ассоциативті процессорлар. Деректерді өңдеудің ассоциативті әдісі бірнеше іріктеу критерийлерін белгілеу және қажетті өзгерістерді тек осы өлшемді қанағаттандыратын мәліметтер бойынша орындау арқылы жадқа қол жетімділікке қатысты көптеген шектеулерді жеңуге мүмкіндік береді.



Сурет 8.1. Ассоциативті жүйенің диаграммасы

### Конвейерлік процессорлар

Заманауи компьютерлердің процессорлары бір уақытта бірнеше команда өңдеуге мүмкіндік беретін арнайы технологияны - құбырларды пайдаланады.



Пәрменді өңдеуді бірнеше негізгі кезеңдерге бөлуге болады, оларды микрокомандалар деп атаймыз. Біз негізгі бес микрокоманданы бөлеміз:

- алып келу тобы;
- команданың шифрын шешу;
- қажетті операндтарды таңдау;
- командаларды орындау;
- нәтижелерді сақтау.

### Матрицалық процессорлар

Класс жүйелерінің ең көп тарағаны - бірыңғай нұсқаулық ағындары - бірнеше деректер ағындары (SIMD) - бұл тәуелсіз нысандардың немесе мәліметтердің параллелизмімен сипатталатын мәселелерді шешуге өте қолайлы матрицалық жүйелер. Алғашқы матрицалық процессорлардың бірі SOLOMON (60-шы жылдар) болды.



Сурет 8.2. Матрицалық есептеу жүйесінің құрылымы SOLOMON

SOLOMON жүйесінде матрица түрінде қосылған 1024 процессор элементі бар: 32x32. Матрицаның әрбір процессор элементіне тізбектелген арифметикалық және логикалық операцияларды, сонымен қатар 16 Кбайт жедел жад сыйымдылығын қамтамасыз ететін процессор кіреді. Сөздің ұзындығы 1-ден 128-ге дейін өзгереді. Бит тереңдігі бағдарламалық түрде орнатылады. Мультимодализм идеясы әр процессор элементінде 4 мемлекет үшін арнайы регистр - сән регистрі болады.

Матрицалық процессорлардың одан әрі дамуы BURROUGHS жасаған ILLIAC-4 жүйесі болды. Бастапқыда жүйеде топтарға бөлінген 256 процессор элементтері болуы керек еді, олардың әрқайсысы арнайы

процессормен басқарылуы керек. Алайда, әртүрлі себептерге байланысты процессор элементтерінің бір тобы мен басқару процессорынан тұратын жүйе құрылды.

80-жылдардың басында КСРО-да матрицалық жүйе болып табылатын PS-2000 жүйесі құрылды. Бұл жүйенің негізі шешім өрісі мен мультипроцессордың басқару құрылғысынан тұратын PS-2000 мультипроцессоры болып табылады. Шешуші өріс бір, екі, төрт немесе сегіз өңдеу құрылғыларынан құралған, олардың әрқайсысында 8 процессор элементі бар. 64 процессор элементінің мультипроцессоры қысқа операциялар үшін секундына 200 миллион операция жылдамдығын қамтамасыз етеді.

### **Дәріс 9. Жоғары өнімді процессорларды ұйымдастыру жолдары. Жасушалық және ДНҚ процессорлары. Байланыс процессорлары**

*Аннотация:* Бұл дәрісте ұялы, ДНҚ және байланыс процессорларының математикалық негіздері, ұйымдастыру әдістері және дизайн ерекшеліктері қарастырылады.

#### **Жасушалық және ДНҚ процессорлары.**

Қазіргі уақытта жаңа есептеу жүйелерін құрудың жартылай өткізгіш технологияларына нақты балама іздеуде ғалымдар биотехнологияға, яғни ақпарат, молекулалық технологиялар, сонымен қатар биохимия гибриді болып табылатын биомехнологияға көп көңіл бөлуде. Биокomпьютер биохимия мен молекулалық биологияда қабылданған әдістерді қолдану арқылы күрделі есептеулерді шешуге, тірі ұлпаларды, жасушаларды, вирустар мен биомолекулаларды қолдана отырып есептеулерді ұйымдастыруға мүмкіндік береді.

#### **ДНҚ процессорлары**

Кез-келген процессор сияқты ДНҚ процессоры құрылымы мен нұсқаулар жиынтығымен сипатталады. Біздің жағдайда процессор құрылымы - ДНҚ молекуласының құрылымы. Пәрмендер жиынтығы - молекулалармен жасалатын биохимиялық операциялардың тізімі. Компьютерлік ДНҚ жад құрылғысының жұмыс істеу принципі төрт нуклеотидтің (ДНҚ тізбегінің негізгі кірпіштері) тізбекті қосылуына негізделген.

ДНҚ молекулаларын есептеулерді ұйымдастыру үшін қолдану өте жаңа идея емес. Бұл мүмкіндіктің теориялық негіздемесі өткен ғасырдың 50-жылдарында жасалды (Р.П. Фейман). Егжей-тегжейлі, бұл теорияны 70-ші жылдары К.Беннетт және 80-жылдары М. Конрад жасаған.

ДНҚ негізіндегі алғашқы компьютерді 1994 жылы американдық ғалым Леонард Адлеман жасаған. Ол пробиркаға бастапқы мәліметтер кодталған ДНҚ молекуласын және арнайы таңдалған ферменттерді араластырды.

Адлеманның жұмысынан кейін басқалары ілесті. Висконсин университетінің қызметкері Ллойд Смит ДНҚ-ны қолдана отырып, төрт мекен-жайға пиццаның төрт түрін жеткізу мүмкіндігіне жауап берді. Принстон университетінің ғалымдары шахматтың комбинаторлық мәселесін шешті: РНҚ көмегімен олар тоғыз жасушадан тұратын тақтадан шахмат рыцарының дұрыс қозғалуын тапты (барлығы 512 опция).

Принстондағы Ричард Липтон ДНҚ-ны пайдаланып, екілік сандарды кодтауды және логикалық өрнекті қанағаттандыру мәселесін қалай шешетінін бірінші болып көрсетті. Оның мәні мынада, логикалық айнымалыларды қамтитын логикалық өрнек болған кезде, сіз өрнекті шындыққа айналдыратын айнымалы мәндердің барлық комбинацияларын табуыңыз керек. Мәселені тек  $2n$  комбинациясын санау арқылы шешуге болады. Барлық осы комбинацияларды ДНҚ көмегімен оңай кодтауға болады, содан кейін Адлеман әдісімен жалғастыруға болады. Липтон сонымен қатар логикалық өрнектің бір түрі ретінде түсіндірілетін DES шифрін (американдық криптографиялық) бұзудың әдісін ұсынды.

Биокомпьютердің алғашқы моделі, алайда, пластмассадан жасалған механизм түрінде, 1999 жылы Вейзмани жаратылыстану институтының өкілі Ихуд Шапиро жасаған. Ол ДНҚ мен ақуыздың арасындағы делдал ретінде РНҚ-ны пайдаланып, ДНҚ ақпаратынан ақуыз молекулаларын жинайтын тірі жасушадағы «молекулалық машинаның» жұмысына еліктеді.

2001 жылы Шапиро адамның араласуынсыз жұмыс істей алатын ДНҚ негізіндегі есептеу құралын іске қосты. Жүйе компьютерлік технологияның негізгі тұжырымдамаларының бірі - Тюринг машинасына еліктейді. Тюринг машинасы деректерді кезең-кезеңімен оқиды және олардың мәндеріне байланысты әрі қарайғы әрекеттер туралы шешім қабылдайды.

## **Дәріс 10. Жоғары өнімді процессорларды ұйымдастыру жолдары. Мәліметтер базасының процессорлары. Ағын процессорлары. Нейрондық процессорлар. Көп мағыналы (анық емес) логикасы бар процессорлар**

*Аннотация:* Бұл дәріс базалық процессорлардың, ағынды процессорлардың, нейрондық процессорлардың және көп мағыналы (анық емес) логикасы бар процессорлардың математикалық негіздерін, ұйымдастыру әдістерін және дизайн ерекшеліктерін қарастырады.

Мәліметтер базасының процессорлары. Мәліметтер базасының процессорлары (машиналары) қазіргі уақытта мәліметтер базасын басқару

жүйелерінің (ДҚБЖ) барлық немесе кейбір функцияларын орындауға арналған аппараттық және бағдарламалық жүйелер деп аталады. Егер бір кездері дерекқорды басқару жүйелері негізінен мәтіндік және сандық ақпаратты сақтауға арналған болса, қазір олар әртүрлі деректерді, соның ішінде графикалық, аудио және бейнелерді форматтауға арналған.

### **Ағын процессорлары**

Ағынды процессорлар деп көптеген процедураларды бір команда арқылы өңдеуге негізделген процессорлар деп атайды. Флинн классификациясына сәйкес, олар SIMD (бір нұсқаулық ағын / бірнеше деректер ағыны) архитектурасына жатады. SIMD технологиясы бір уақытта бірнеше сандар жиынтығында бірдей әрекетті орындауға мүмкіндік береді, мысалы, алу және қосу. Бұл бөлек ағынды процессор (Бір ретті ағынды процессор - SSP) және көп ағынды процессор (Multi-Streaming Processor - MSP) болуы мүмкін.

Ағынды процессорлардың жарқын өкілі - ағымдық SIMD кеңейтімдеріне негізделген Sentium III (SSE, «бір команда - көптеген деректер» ағынды өңдеу) негізіндегі Pentium III бастап Intel процессорлар тобы. Бұл технология Интернет ғасырында сөйлеу, бейне және аудио деректерді кодтау және декодтау, үш өлшемді графика мен кескіндерді өңдеу сияқты күрделі және қажетті тапсырмаларды орындауға мүмкіндік береді.

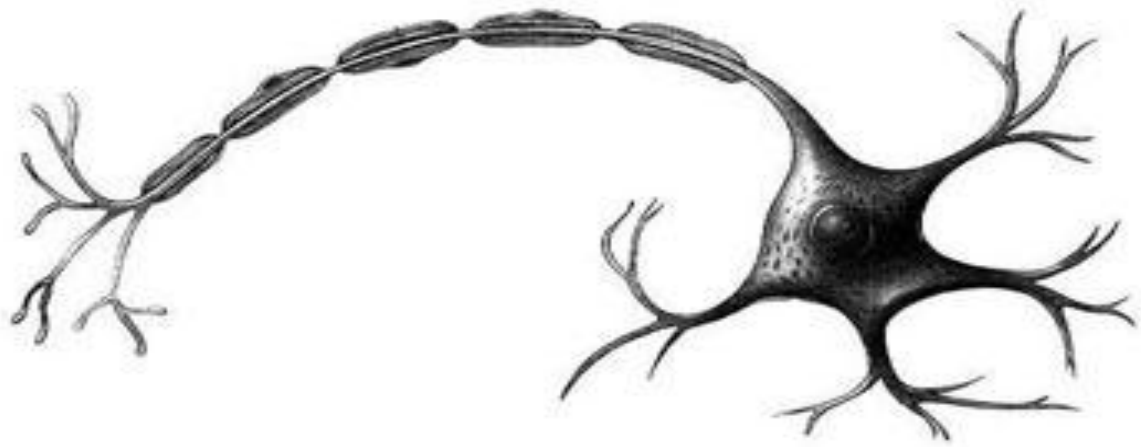
SIMD класының өкілдері процессор матрицалары болып саналады: ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine 1 және т.б. Мұндай жүйелерде бір басқару құрылғысы көптеген процессор элементтерін басқарады. Әрбір процессор элементі басқару приборынан бірдей белгіленген периодты белгілі бір уақытта алады және оны жергілікті деректерде орындайды.

SIMD класының басқа өкілдері - векторлық процессорлар, олар векторлық мәліметтерді өңдеуге негізделген. Векторлық өңдеу процестің өнімділігін арттырады, өйткені барлық мәліметтер жиынтығын (векторын) бір команда бойынша өңдеу жүзеге асырылады. Векторлық компьютерлер осындай массивтердің жекелеген элементтерін скалярлық машиналар өңдейтін ұқсас мәліметтер массивтерін басқарады. Бұл жағдайда вектордың әрбір элементін деректер ағынының жеке элементі ретінде қарастыру керек.

Нейрондық процессорлар. Есептеу жүйелерінің түбегейлі жаңа архитектураларын дамытудың перспективалық бағыттарының бірі жасанды нейрондық желілерде (NS) енгізілген ақпаратты өңдеу принциптеріне негізделген жаңа буын компьютерлерінің құрылуымен тығыз байланысты.

Жасанды нейрондық желілер мен нейрокомпьютерлердегі алғашқы практикалық жұмыс 40-50 жж басталды. Нейрондық желі деп әдетте ақпарат алмасу арналары - «синаптикалық байланыстар» арқылы бір-бірімен белгілі бір түрде байланысқан «нейрондар» деп аталатын қарапайым ақпарат түрлендіргіштер жиынтығы түсініледі.

Нейрон, шын мәнінде, кіріс және шығыс күйімен, берілу функциясымен (активация функциясы) және жергілікті жадпен сипатталатын қарапайым процессор.

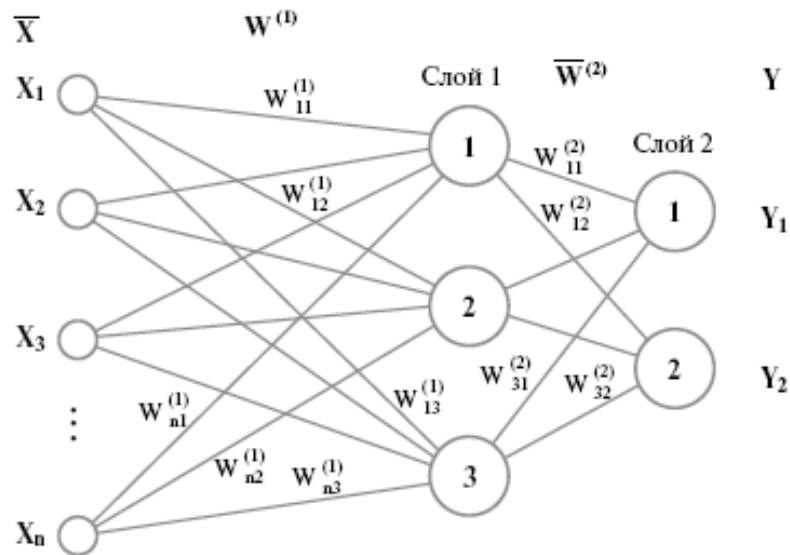


Сурет 10.1.

Нейрондардың күйлері жұмыс кезінде өзгереді және нейрондық желінің қысқа мерзімді есте сақтауын құрайды. Әр нейрон синапстарда оған түскен сигналдардың салмақты қосындысын есептеп, ондағы сызықты емес түрлендіруді жүзеге асырады. Синапстар арқылы жіберген кезде сигналдар белгілі бір салмақтық коэффициентке көбейтіледі. Салмақтарды бөлу кезінде Ұлттық ассоциацияның жадында сақталған ақпарат болады. Желіні жобалаудың негізгі элементі - оны оқыту. Ұлттық Ассамблеяны оқыту және қайта даярлау кезінде оның салмақтық коэффициенттері өзгереді. Алайда олар нейрондық желінің жұмыс істеуі кезінде тұрақты болып қалады, ұзақ мерзімді жадты қалыптастырады.

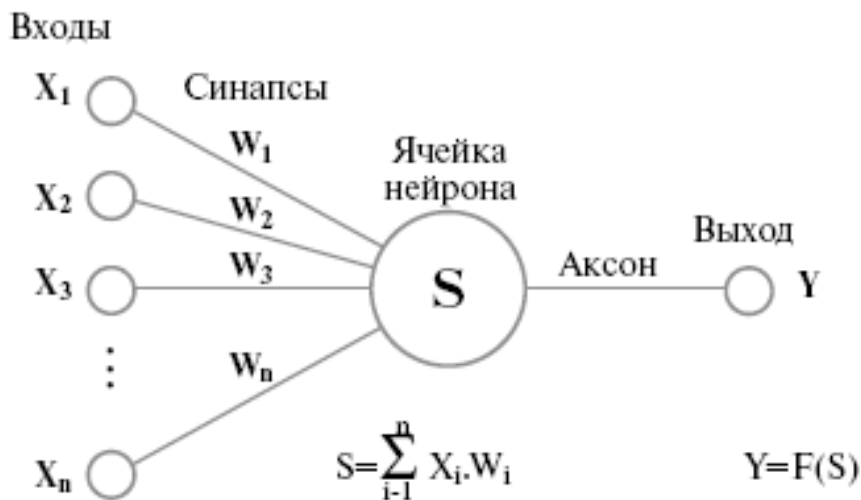
NS бір қабаттан, екіден, үштен немесе одан да көп қабаттардан тұруы мүмкін, алайда, практикалық есептерді шешу үшін NS-де үштен көп қабат талап етілмейді.

NS енгізу саны гипер кеңістіктің өлшемін анықтайды, онда кіріс сигналдары жақын орналасқан нүктелерден нүктелермен немесе гиперрегиондармен көрсетілуі мүмкін. Желілік қабаттағы нейрондардың саны гиперсферадағы гиперпланттардың санын анықтайды. Өлшенген сомаларды есептеу және сызықты емес түрлендіруді орындау гиперпланнның гиперпланктегі кіріс сигналының нүктесі екенін анықтауға мүмкіндік береді.



Сурет 10.2.

Классикалық үлгіні тану мәселесін шешіңіз: нүкте екі кластың біріне жататындығын анықтаңыз. Мұндай тапсырма табиғи түрде бір нейронды қолдану арқылы шешіледі. Бұл гиперсфераны екі дисконтталған және кірмеген гиперрегиондарға бөлуге мүмкіндік береді. Нейрондық желілер көмегімен шешілетін есептердегі кіріс сигналдары қатты ендірілген немесе гипер кеңістігінде бір-біріне нейронды бөліп алуға болатын аймақтарды құрайды.



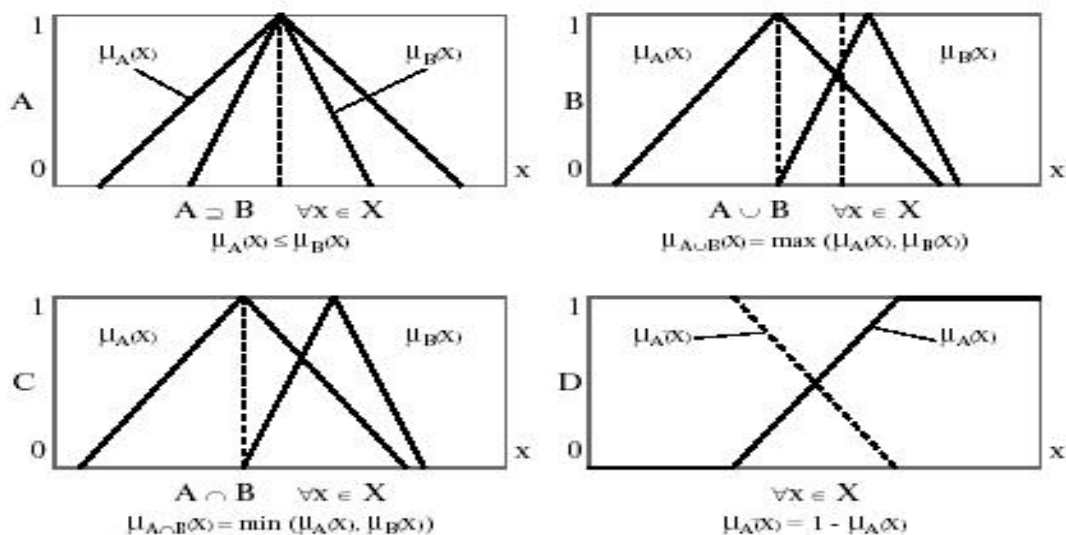
Сурет 10.3.

Жасанды нейрондық желілер таңғажайып қасиеттерге ие. Олар бағдарламалық жасақтаманы егжей-тегжейлі әзірлеуді қажет етпейді және шешім алгоритмін анықтайтын теориялық модельдер немесе эвристикалық ережелер жоқ есептерді шығару үшін мүмкіндіктер ашады. Мұндай желілер

жұмыс жағдайының өзгеруіне, оның ішінде бұрын болжанбаған факторлардың пайда болуына бейімделу қабілетіне ие. Өз табиғаты бойынша NS - бұл параллелизмнің өте жоғары деңгейі бар жүйелер.

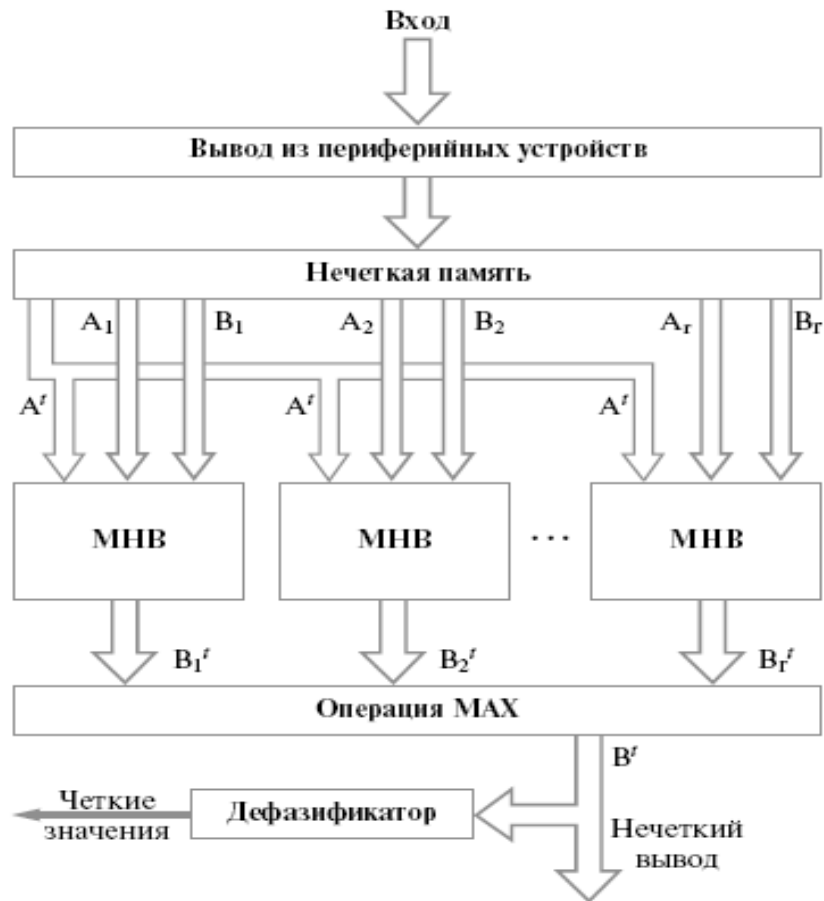
Бұлыңғыр математика тәсілдері уақыт өте келе өзгеріп отыратын мәліметтермен және бір мәнді орнатылмайтын мәндермен жұмыс істеуге мүмкіндік береді, мысалы, статистикалық зерттеулер нәтижелері. Аристотель дәуірінен бері белгілі және ақиқат пен жалған, иә және жоқ, нөл және біреу сияқты нақты және айқын ұғымдармен жұмыс істейтін дәстүрлі ресми логикадан айырмашылығы, анық емес логика белгілі бір (үздіксіз немесе дискретті) диапазонда орналасқан құндылықтармен айналысады.

Берілген жиынға жататын элементтердің функциясы «тиесілі - тиесілі» деген қатаң шекті білдірмейді, бірақ нөлден бір мәнге дейінгі барлық мәндер арқылы өтетін тегіс сигмоид. Бұлыңғыр логика теориясы осындай шамаларда логикалық операциялардың барлық спектрін - біріктіру, қиылысу, теріске шығаруды және т.б. орындауға мүмкіндік береді.



Сурет 10.4. Қосылу (A), одақ (B), қиылыс (C) және N (D) қосымшалары

Cosco дәлелдеген әйгілі FAT (Fuzzy approximation Theorem) сәйкес кез-келген математикалық жүйені анық емес логикаға негізделген жүйеге жақындатуға болады. Бұлыңғыр логика теориясы сексенінші жылдардың басында, бірнеше зерттеушілер тобы (негізінен АҚШ пен Жапонияда), бақылаудың алгоритмдерін қолдана отырып, әртүрлі қосымшалар үшін электрондық жүйелер құра бастаған кезде аман қалды. Маңызды презентацияның қарапайымдылығынан тұратын анық емес логиканың артықшылықтарын қолдана отырып, мәселені жеңілдетуге, оны қол жетімді түрде ұсынуға және жүйенің жұмысын жақсартуға болады.



Сурет 10.5. Анық емес компьютерлік архитектура (ЕОМ анық емес енгізу механизмі)

Еуропада және АҚШ-та анық емес командаларды ендірілген құрылғылардың өнеркәсіптік контроллерлерін құрастырушыларға біріктіру бойынша қарқынды жұмыс жүргізілуде (Motorola 68HC11.12.21 чиптері). Мұндай аппараттық құрал қарапайым ядрода орындалғанға қарағанда бағдарламаны орындау жылдамдығын және кодтың ықшамдылығын бірнеше есе арттыруға мүмкіндік береді.

### Дәріс 11-12. І / О құрылғылары.

Ақпаратты енгізу-шығару құрылғылары (І / О) сыртқы объектілермен немесе компьютерлік жүйемен адам операторы арасындағы байланысты қамтамасыз етуге арналған.

Компьютерлермен және қоршаған ортамен байланысты ауа жарылысын екі топқа бөлуге болады:

- адам-компьютерлік байланыс құралдарын;
- басқару объектілерімен байланысқа арналған құрылғылар.



Компьютер тек екілік сандармен кодталған физикалық күйлердің тілін түсінетіндіктен, әуедегі жарылыс, әдетте, компьютерге жіберілген ақпаратты (кодты алады) кодтайды.

Әуе жаруының электрлік тізбектерге қарағанда сенімділігі аз механикалық элементтері көп, сонымен қатар қателіктер электр шуылына байланысты және процесс (жад) мен ауа-жарылыс (ауа-үрлеу контроллері) арасында деректерді беру кезінде пайда болуы мүмкін. Сондықтан, әртүрлі шуылға қарсы кодтарды (паритет, тексеру кестесі, Хэмминг коды және т.б.) қолдана отырып, ақпаратты жіберген кезде қателерді анықтау немесе оларды жартылай түзету қажет.

Тиісті шифрланған мәліметтер байланыс жолдары арқылы екі жолмен беріледі.:

- параллель;
- дәйекті.

Компьютер қашықтықтағы әуе жарылыстарымен немесе басқа компьютерлермен мәліметтер алмасуы керек болған кезде, объектілер арасындағы салыстырмалы үлкен қашықтық деректермен алмасуды анықтайды. Байланыс құралы ретінде телефон желілері немесе радиоарналар жиі қолданылады.

Эфирлік жарылыс пен компьютер арасында екі сымды байланыс желісін немесе радиоарнаны пайдалану кезінде құрылғы параллель кодтан сериялық кодқа және керісінше мәліметтерді түрлендіру үшін қолданылады.

Байланыс желілері бойынша мәліметтерді беру кезінде үш режимнің біреуі қолданылады:

- симплекс;
- жартылай дуплекс;
- дуплексті.

Шығару құрылғылары

1. Экранның қашықтан басқару пульті:

– әріптік-сандық дисплейлер - әріптік-сандық ақпаратты шығару үшін;

– CRT графикалық дисплейлері - мәтіндік және графикалық;

– Кескінді қалпына келтіру режимі бар векторлық дисплейлер және экранда графикалық кескіннің контурын алудың параметрлік әдісі нүктелер жиынтығы, векторлар мен символдар сегменті түрінде статикалық және динамикалық суреттерді алуға арналған.;

– растрлық дисплейлер - нүктелер, сегменттер, векторлар, символдар мен толтырылған аймақтар жиынтығы түрінде кескін алуға арналған (нүктелік кескін элементтерінің матрицасын қолдану - пиксельдер);

1. плоттерлер - қағазға графикалық ақпаратты шығаруға арналған құрылғылар;

2. баспа құрылғылары - принтерлер, плоттерлер;

3. ВЗУ.

Кіріс құрылғылар

1. Пернетақта - электромеханикалық типтегі енгізу құрылғылары;

2. Сканерлер - бағдарламалық қамтамасыздандыруды цифрландыру мүмкіндігімен графикалық түрдегі ақпаратты енгізу құрылғылары;

3. Графикалық планшеттер - графиктерді, диаграммаларды, сызбаларды енгізуге арналған құрылғылар, содан кейін бағдарламаларды өңдеу мүмкіндігі - цифрландырғыштар;

4. Кіріс құрылғыларын басқару:

– тышқан;

– джойстик;

– жеңіл қалам.

**Тест сұрақтары:**

1. I / O құрылғылары не үшін арналған?

2. I / O құрылғыларын қандай екі топқа бөлуге болады?

3. Интерференцияға қарсы кодтар дегеніміз не?

4. Байланыс желілері бойынша беруді ұйымдастыру кезінде қандай режимдер қолданылады?

5. Кіріс құрылғылары қандай құрылғылар?

6. Шығу құрылғылары қандай құрылғылар?

7. Қандай құрылғылар кіріс және шығыс функцияларын орындай алады?

### **Дәріс 13. Ақпаратты өңдеу жүйелері. ВК мультипроцессоры. Деректерді тарату жүйелері (ДТЖ)**

Параллельді есептеулер мен үлестірілген өңдеу үшін үлкен қажеттілік деректер мәселелерді шешкен кезде пайда болады:

– күрделі технологиялық жүйелерді модельдеу және жобалау;

– ядролық реакторларды басқару;

– ғарыштық зерттеулер нәтижелерін өңдеу;

– геофизикалық зерттеулердің мәселелерін шешу;

– үлгіні тану және кескінді өңдеу;

– жасанды интеллект т.б.

ДТЖ дегеніміз дисперсті есептеу процестерін компьютердегі сияқты бірнеше процессорларды бір уақытта қолдану арқылы ұйымдастыру әдісі. Жіктеу:

1. функционалдық мақсатта:

– ДТЖ өндірісті басқаруда қолданылады: кәсіпорындар, технологиялық және өндірістік процестер және т.б ;

– өндірістік емес объектілерді басқаруда қолданылатын ДТЖ: мекемелер, бөлшек сауда желілері, кітапханалар, жинақ кассалары және т. Б.;

– ДТЖ ғылыми зерттеулерді басқаруда қолданылады: ғылыми-зерттеу институттарында, университеттерде, жобалау және технологиялық институттарда немесе бюроларда және т.б.

2. басқаруды орталықтандыру дәрежесіне сәйкес:

- орталықтандырылған басқарумен ДТЖ;
- орталықтандырылмаған басқарумен ДТЖ;
- аралас басқарумен ДТЖ.

3. ДТЖ жұмыс режимі бөлінеді:

- пакеттік режимдегі жүйелер;
- интерактивті режимдегі жүйелер;
- уақыт бөлу режимі бар жүйелер.

4. компьютер мен ОЖ түрі бойынша:

- біртекті ДТЖ;
- ішінара гетерогенді ДТЖ;
- гетерогенді ДТЖ.

5. аумақтық бөліну дәрежесі бойынша:

- жергілікті деректерді өңдейтін жүйелер;
- таратылған мәліметтерді өңдейтін жүйелер.

6. мәліметтерді тарату әдістері бойынша:

– бір немесе бірнеше компьютерлері бар орталықтандырылған мәліметтер жүйелері;

– тәуелді мәліметтердің иерархиясы бар жүйелер - басқару жүйесі бір компьютерде сақталады, ал төменгі деңгейдің мәліметтері жоғарғы деңгейдегі ЭЕМ деректерімен қосылады;

– барлық дербес компьютерлер мәліметтерді өңдеудің жабық жүйелері болып табылатын, таратылған басқаруымен дербес мәліметтер иерархиясы бар жүйелер;

– ДТЖ құрамына кіретін барлық компьютерлердегі және құрылымы бірдей болатын мәліметтер бөлінген жүйелер;

– таратылған мәліметтері бар жүйелер, мұнда әр компьютердің өзіндік құрылымы болады;

– кейбір компьютерлерде бірдей мәліметтердің көшірмелері бар жүйелер.

Есептеуіш кешендер мен жүйелер

Есептеу жүйесінің архитектурасы – бұл жүйенің функционалды модульдерінің (блоктарының) әрқайсысында жүйелік функциялардың көп деңгейлі орындалуын қамтамасыз ететін өзара байланысты аппараттық және бағдарламалық құралдар жиынтығының сипаттамасы.

Есептеу жүйесінің архитектурасы өзара байланысты құрылымдардың үш түрін қамтиды:

- физикалық;
- логикалық;
- бағдарламалық қамтамасыз ету.

Құрылымдардың әрқайсысында жеке элементтер бар:

- физикалық құрылым (ФҚ):
- негізгі құрылым;
- есте сақтауды ұйымдастыру;
- университетпен өзара әрекеттесуді ұйымдастыру;
- басқару принциптері.

Физикалық құрылымды зерттеу қарастырылатын жүйенің физикалық ресурстарын анықтауға, процессорлардың немесе компьютерлердің, жадтың және т.б. қажетті санын есептеуге мүмкіндік береді; олардың техникалық сипаттамалары қандай болуы керек екенін анықтаңыз. Сонымен бірге, барлық функционалды модульдердің, соның ішінде әртүрлі жад деңгейлері мен басқару блогының барлық жиынтығының өзара әрекеті Қарулы Күштер қабылдаған басқару, күй және модульдік байланыстар қағидаттарына сәйкес жүзеге асырылады.

Қажетті құрылымды таңдау кезінде бірқатар мәселелерді шешу қажет:

- есептеу жүйесінің құрылымын алгоритмдер мен бағдарламалардың осы класын жақсы іске асыратындай етіп таңдаңыз;
- есептеу жүйесінің мүмкіндіктерін тиімді пайдалануға мүмкіндік беретін бағдарламалық жасақтаманың (құрылымның) құрамын анықтау;
- алгоритмдер мен бағдарламаларды құру әдістерін, сонымен қатар берілген есептеу жүйесіне арналған машиналық тілді таңдау;
- берілген есептеу жүйесінде ең жақсы орындалатын есептеу есептері класын анықтау.

Логикалық құрылым (ЛҚ):

- көптеген ерекшеліктер;
- командалар жүйесі;
- мәліметтер форматтары;
- командалардың уақытша сипаттамалары.

ЛҚ есептеу жүйесінің есептеу мүмкіндіктерін анықтайды. Сонымен бірге программалар мен деректер форматын құруда қолдануға болатын командалар тізімі жүйенің белгілі бір тапсырмалар класына бағдарлануын бағалауға мүмкіндік береді.

Бағдарлама құрылымы (ПС):

- языки программирования;
- ОЖ
- ДҚБЖ;
- ППП.

Есептеу жүйесінің бағдарламалық құрылымы – бұл жүйелік аппараттық құралдарды тиімді пайдалануға мүмкіндік беретін өзара байланысты бағдарламалар кешені.

Сонымен, Есептеу жүйесінің дизайнері үш негізгі мәселені шешуде таңдауды негіздеуі керек:

- команданың жүйесіне, олардың орындалу уақытына, сипаттамаларына, бағдарламалау тілдеріне, қолданбалы бағдарламаларға, ОЖ және т.с.с тәуелді күн программасында түсіндіру ережелерін және түсіндіру ережелерін;

- жадыны ұйымдастыру және есептеулерді басқару принциптеріне тәуелді бағдарламалардағы мәліметтерді адресациялау әдістерін құру;

- мәліметтер форматын, жадыны ұйымдастыру тәсілдерін және сыртқы құрылғылармен өзара әрекеттесуді анықтайтын мәліметтерді ұсыну формасын орнатыңыз.

Компьютерлік жүйелер мен жүйелердің жіктелуі

1. функционалдық мақсатта:

- әмбебап – кең ауқымды міндеттерді шешуге арналған;
- мамандандырылған - белгілі құрылымы бар міндеттерді алдын-ала шешуге арналған.

Мамандану деңгейлері:

- пайдаланушы – қосымшаның ішкі жүйелері мен қолданушы бағдарламалары деңгейінде;

- тіл – қолданушымен тығыз байланысты, өйткені проблемалық бағдарлау бағдарламалау тілдерінің шектеулі санын қолдануды қарастырады;

- жүйе – ОЖ және аппараттық құрал арасында функциялардың бөлінуімен байланысты;

- функционалды – тәуелсіз модульдер деңгейіндегі мамандандыру, яғни Қарулы Күштер шеңберіндегі машиналар, процессорлар немесе ішкі жүйелер;

- архитектуралық - бір жүйенің, машинаның, процессордың немесе ішкі жүйенің құрамындағы құрылғылар мен модульдерге жатады. Сонымен қатар, олардың барлығына бағдарламашылар қол жеткізе алады, олар өздерінің ұйымдастырылу ерекшеліктерін, уақытша уақытша жағдайларды және т.б. білуі және ескеруі керек;

- аппараттық құрал – авиациялық техниканың мамандандырылу деңгейі, олар қол жетімді және бағдарламалаушыға белгісіз болуы мүмкін;

- схема – бұл деңгей жеке тізбектерге немесе құрылымдық модульдерге жатады.

2. әуе кемесін басқаруды орталықтандыру дәрежесі бойынша мыналарға бөлінеді:

- орталықтандырылған басқару жүйелері;

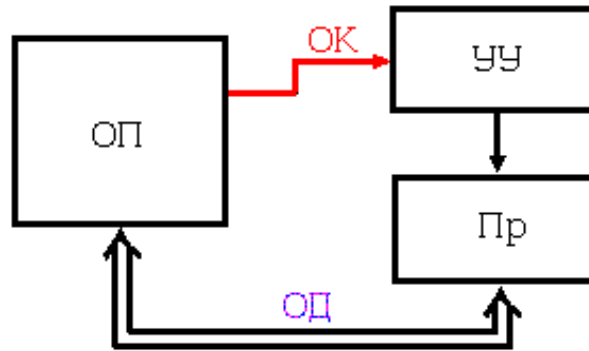
- орталықтандырылмаған таратылған басқару жүйелері;

- аралас басқару.

3. Әуе кемесінің жұмыс режиміне сәйкес:

- пакеттік режиммен;

- «сұраныс - жауап» режимі;
  - нақты уақыт режимі.
4. Командалар мен мәліметтер ағындарының саны бойынша (М. Флинин бойынша) әуе кемелері әуе кемелеріне (ВК) бөлінеді:
- деректердің бірыңғай ағыны және командалардың бірыңғай ағымы – көп емес процессорлар (13.1-сурет)

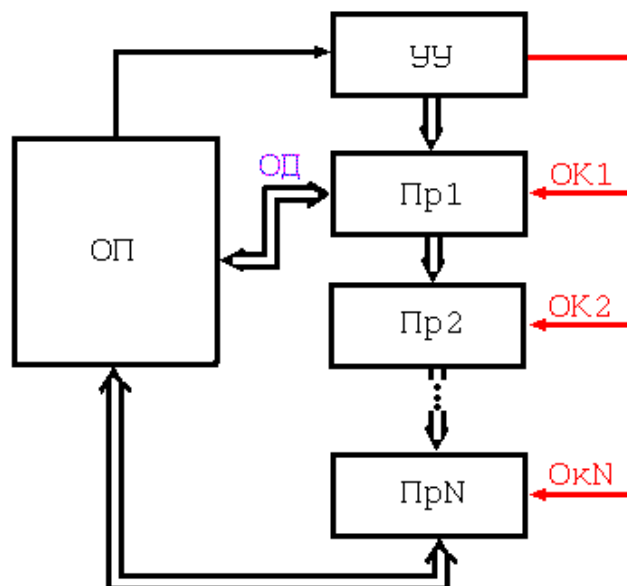


Сурет 13.1. Деректер мен командалардың бірыңғай ағымы

- командалардың бірыңғай ағымы және мәліметтердің бірнеше ағымы – матрица, вектор, ассоциативті, қалдық класстардың авиациясы (13.1-сурет).

Бірыңғай командалық ағын және бірнеше мәліметтер ағыны

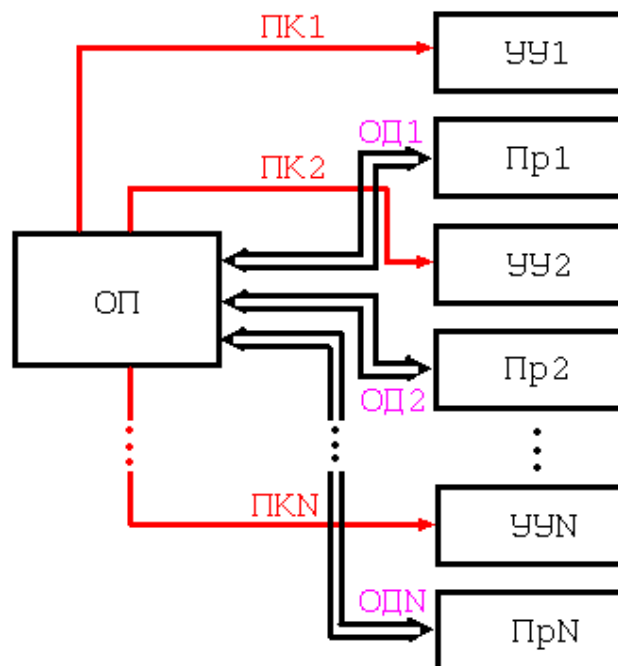
- командалардың бірнеше ағымы және бірыңғай мәліметтер ағыны – құбырланған (13.2-сурет).



Сурет 13.2 Бірнеше нұсқаулық ағындар және бірыңғай деректер ағыны

- командалардың бірнеше ағымы және мәліметтердің бірнеше ағымы (МК МД) - көп машиналы ВК, МРVS, ағынды ұшақтар, динамикалық

құрылымы бар ұшақтар, гибридтік ұшақтар, нейро-компьютерлік ұшақтар (13.3-сурет)



Сурет 13.3 Пәрмендер мен мәліметтердің бірнеше ағымы

5. Әуе кемесін жинақтау әдісімен (ШК) мыналарға бөлінеді:

- төмен жалғанған;
- байланысы жоғары.

Сонымен қатар, байланыс деңгейі Қарулы Күштерді (ВК) құруға қатысатын интеграция деңгейлерінің санына байланысты:

- университет деңгейіндегі интеграция;
- арналар деңгейіндегі АСС адаптерін қолдана отырып канал деңгейінде интеграциялау;
- деректерді беру жабдықтары деңгейінде интеграция (ADF);
- ортақ жедел жады деңгейінде интеграция;
- процессор деңгейінде интеграция.

Интеграцияның алғашқы үш деңгейін пайдалану бір-біріне жұпталған жүйелер береді. Соңғы екі деңгей күрделілігі жоғары қарулы күштерді (ШК) құрайды.

6. компьютер компоненттерінің түрлері бойынша мыналарға бөлінеді:

- біртекті - бағдарламалық және аппараттық үйлесімді компьютерлерден тұрады;

- гетерогенді.

7. ішкі қатынастарды ұйымдастыру әдісімен:

- жалпы автобуспен;
- кросс коммутациясымен;
- көп кіріс жадымен;

- ағынды - бағдарламалық қамтамасыз етуді есептеуді басқару мәліметтердің болуына сәйкес жүзеге асырылады;
  - бағдарламаланатын немесе динамикалық сәулетімен;
  - нейрокомпьютерлер;
  - матрица және вектор;
  - ассоциативті;
  - конвейер;
  - қалдық класстардың ұшақтары;
  - гибридті.
8. өңдеу түйіндерінің құрамы:
- жалпы мақсаттағы мультипроцессор;
  - мамандандырылған, есептеу есептері құрылымдарының белгілі бір түрлеріне бағытталған.

### **Тест сұрақтары:**

1. Қандай мәселелерді шешуде параллельді есептеулер мен таратылған мәліметтерді өңдеудің қажеттілігі көбірек?
2. ДТЖ функционалдық мақсатына, басқаруды орталықтандыру дәрежесіне, жұмыс режимдеріне, компьютер мен ОЖ типіне, аумақтық бөліну дәрежесіне, мәліметтерді тарату әдісіне жіктеуге мысалдар келтіріңіз.
3. Қарулы Күштердің архитектурасын қамтитын өзара байланысты құрылымдардың үш түрі қандай?
4. Физикалық құрылым деңгейінде қандай мәселелер шешіледі?
5. Логикалық құрылымды жүзеге асыру қандай функцияларды қамтамасыз етеді?
6. Бағдарламалық құрылым деңгейінде қандай мәселелер шешіледі?
7. Компьютерлік жүйелер мен жүйелердің мамандану деңгейлері қандай?
8. Компьютерлік жүйелер мен жүйелер қалай жіктеледі?

## **Дәріс 14. Компьютерлік желілер.**

### **Негізгі ақпарат**

Жергілікті желі дегеніміз – кабельдермен жалғанған компьютерлер, перифериялық құрылғылар (принтерлер және т.б.) және коммутациялық құрылғылар жиынтығы. Кабель ретінде «қалың» коаксиалды кабель, «жұқа» коаксиалды кабель, бұралған жұп кабель, талшықты-оптикалық кабель қолданылады. «Қалың» кабель негізінен өткізгіштік қабілеті жоғары қажеттіліктері бар алыс қашықтықтарда қолданылады. Бастапқыда желілер жұқа Ethernet принципін қолдана отырып құрылды. Ол коаксиалды кабельмен қатар қосылған желілік адаптері бар бірнеше компьютерге



негізделген және барлық желілік адаптерлер бір уақытта оған сигнал береді. Бұл принциптің кемшіліктері кейінірек белгілі болды.

Желілік өлшемдердің өсуімен бір компьютерде көптеген компьютерлердің қатар жұмыс істеуі іс жүзінде мүмкін болмады: бір-біріне өзара әсер өте жақсы болды. Коаксиалды кабельдің кездейсоқ өшуі (мысалы, өзектің ішкі сынуы) бүкіл желіні біржола өшірді. Ал үзіліс орнын немесе желіні «жалғанған» бағдарламалық жасақтама жұмысының бұзылуын анықтау мүмкін болмады.

Сондықтан компьютерлік желілерді одан әрі дамыту құрылымдау принциптеріне негізделген. Бұл жағдайда әр желі өзара байланысты бөлімдер жиынтығынан тұрады - құрылымдар.

### **Жергілікті желі**

Батыс әлеміндегі компьютерлердің басым көпшілігі белгілі бір желіге біріктірілген. Пайдалану желілерінің тәжірибесі көрсеткендей, желі арқылы жіберілген барлық ақпараттың 80% -ы бір кеңседе құлыпталған.

Компьютерлік желілердің екі түрі бар: тең-теңімен және арнайы сервермен желілер. «Тең-теңімен» желісі желіні ұйымдастыратын арнайы компьютерлерді бөлуді қарастырмайды.

### **Ғаламдық желілер**

Телефон желілері қашықтағы компьютерлік желілерге қосылу үшін қолданылады. Деректерді телефон желілері арқылы беру процесі электрлік тербелістер түрінде болуы керек - дыбыстық сигналдың аналогы, ал компьютерде ақпарат код түрінде сақталады. Компьютерден ақпаратты телефон желісі арқылы беру үшін кодтарды электрлік тербелістерге айналдыру керек.

### **INTERNET халықаралық желісі**

ИНТЕРНЕТ-тің алғашқы нұсқаларының бірі жетпісінші жылдары АҚШ Қорғаныс министрлігімен сол кездегі қорғаныс үшін ерекше маңызды мәселелер бойынша жұмыс істейтін ғылыми-зерттеу институттарына ақпарат алмасуға мүмкіндік берді.

Әрбір ИНТЕРНЕТ пайдаланушысының жеке желілік мекен-жайы болады. Пайдаланушылар арасында екі бірдей адрес пайда болмас үшін INTERNET мекенжайларын бақылайтын (Вирджиния штатында) компания бар.

### **Тест сұрақтары:**

1. Жергілікті желі дегеніміз не?
2. Коммутациялық құрылғылар ретінде желіні құру кезінде не қолданылады?
3. Желілік адаптерлердің мақсаты?
4. Жергілікті желінің қандай ерекшелігі бар?
5. «Тең-теңімен» желісі дегеніміз не?
6. Клиент-сервер архитектурасы дегеніміз не?

## 7. Желілік адрес дегеніміз не?

### **Дәріс 15. Есептеу жүйелері мен желілерін жобалау негіздері**

Қазіргі жағдайда компьютерлік жүйелер мен желілерді жобалау компьютерлік технологияны қолдану арқылы жүзеге асырылады.

Жалпы, компьютерлік технологияны жобалауды екі кезеңге бөлуге болады:

– пайдалану талаптарын және осы компьютерлік құралды сипаттайтын барлық материалдарды зерттеу.

– мәселелерді шешу, жобалық құжаттама түріндегі дизайны және оның дизайнын егжей-тегжейлі зерделеу.

#### **Жобалау және өндіріс кезеңдері**

Дизайн және өндіріс кезеңдері 13.4 суретте көрсетілген

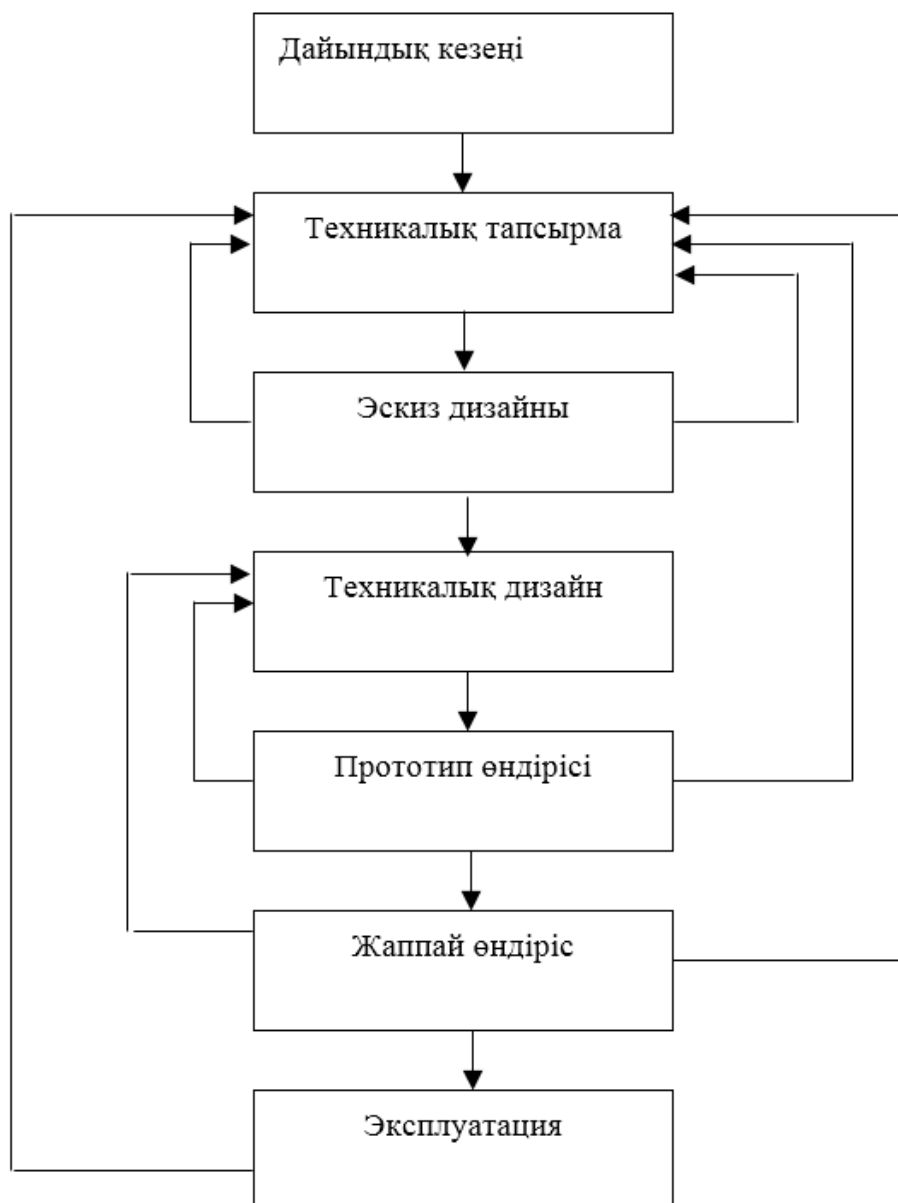
**Дайындық кезеңі.** Жобалау дайындық кезеңінен басталады, оның негізгі мәселесі компьютерлік техниканың жұмыс істеу жағдайлары мен міндеттерін зерттеу болып табылады. Бұл кезеңнің мақсаты - техникалық шарттарды дайындау.

Техникалық тапсырма мыналарды көрсететін құжат болып табылады:

– компьютерлік технологияларды тағайындау,  
– пайдалану шарттары, сақтау және тасымалдау,  
– негізгі техникалық сипаттамалары: ақпаратты өңдеудің жылдамдығы мен дәлдігі, дизайнның параметрлері, сенімділігі, құны, жалпы техникалық талаптар (температура, ылғалдылық шегі, кернеудің ауытқуы, шуылға қарсы иммунитет және т.б.), электрмен жабдықтауға қойылатын талаптар, мүмкін пайдалану үшін ұсынылған тізбек элементтері.

**Контурдың дизайны.** Концептуалды дизайнның негізгі мақсаты – белгіленген техникалық талаптарға сәйкес келетін компьютер құралын жасау мүмкіндіктерін анықтау. Алдын ала жобалау кезеңінде келесі міндеттер шешіледі:

– компьютерлік құрал салынатын техникалық негізді (яғни физикалық элементтер мен бөлшектерді) анықтау;



Сурет 13.4. Компьютерлік техниканы жобалау және өндіру кезеңдері

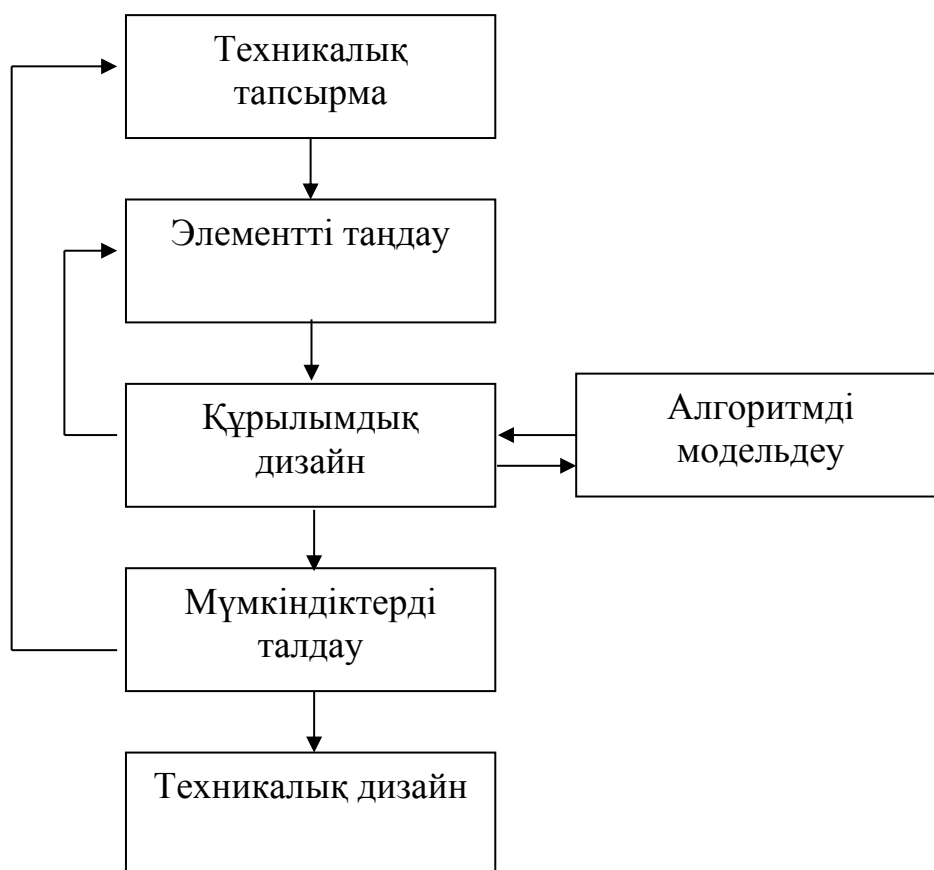
– функционалды, мамандандырылған және көмекші элементтер жүйесін анықтау,

Жабдықтардың есептік мөлшерін есептеу

– есептеу техникасының құрылымын дамыту;

Осы мәселелерді шешу реті және олардың бір-біріне әсері суретте келтірілген (сурет 13.5)

Бұл кезеңде шешілетін міндеттердің алгоритмдерін талдауға көп көңіл бөлінеді. Алгоритмдер көбінесе компьютер құралының логикалық құрылымын анықтайды, яғни. арифметикалық және логикалық амалдардың реттілігі.



Сурет 13.5 Эскиз дизайны

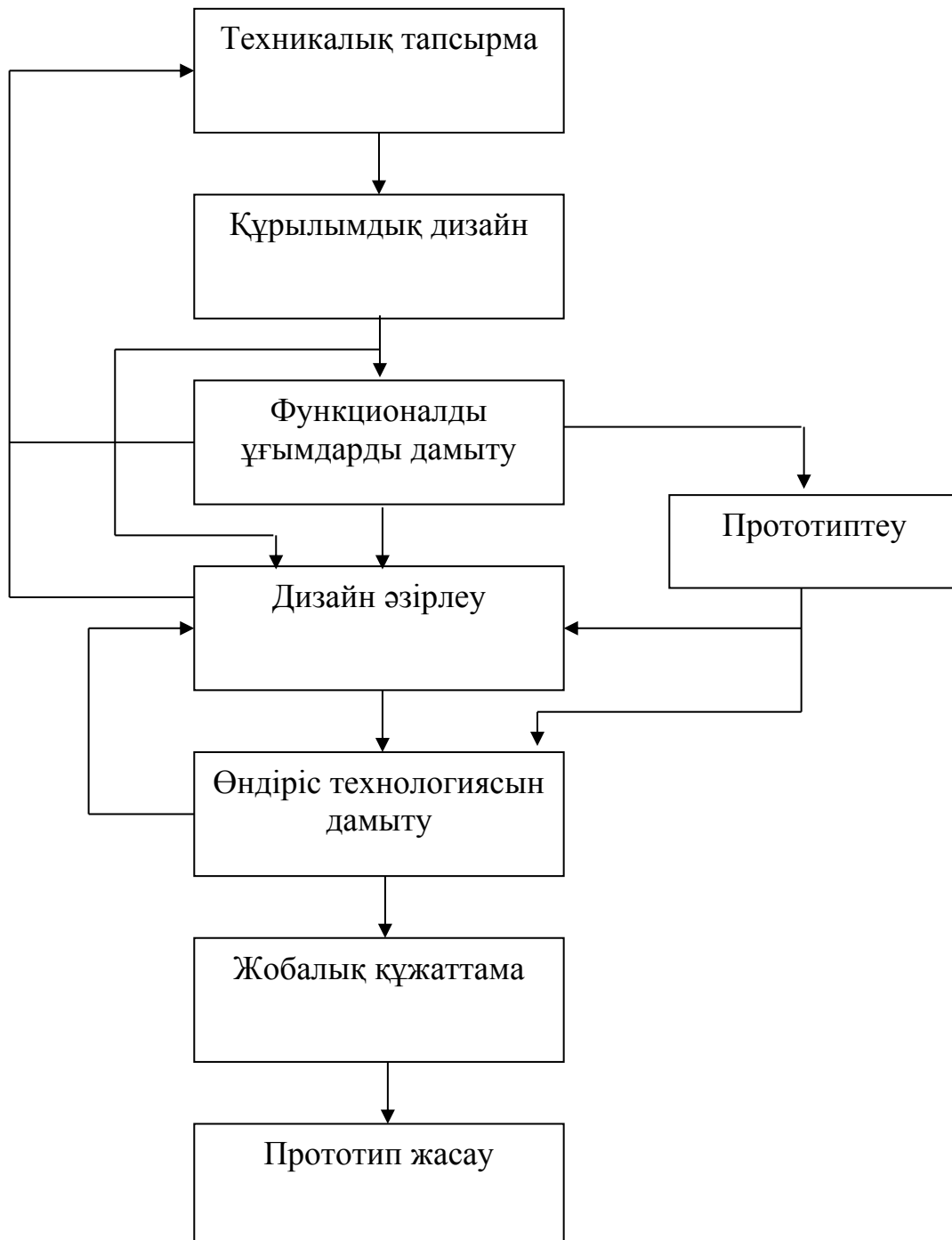
Алгоритмдерді талдау нәтижесінде негізгі құрылғылардың құрамы және қажетті техникалық сипаттамалары анықталады. Егер мәлімделген талаптарға жауап беретін компьютер құралын құру мүмкін емес екені белгілі болса, онда барлық техникалық тапсырма өңделеді. Алгоритмдерді талдау оларды қолданыстағы компьютерлерде модельдеу арқылы жүзеге асырылады, бұл біркатар дизайн кезеңдерін автоматтандыруға мүмкіндік береді.

**Техникалық дизайн.** Техникалық жобалау сатысында келесі міндеттер шешіледі:

- есептеу техникасы мен оның барлық құраушы блоктарының жұмыс принципін, техникалық сипаттамаларын егжей-тегжейлі әзірлеу;
- компьютерлік технологияның блоктарын, тораптары мен құрылғыларының дизайнын жасау, дизайн сипаттамаларын алу;
- компьютерлік технология құралдарынан тұратын барлық құрылғылар үшін өндіріс технологиясын жасау;
- барлық құрылғыларды үйлестіру;
- компьютерлік техниканы құрастыру, іске қосу және сынау әдістерін анықтау.

Техникалық жобалау кезеңдерінің кезектілігі 13.6 суретте келтірілген.

Электр схемаларын әзірлеу – бұл компьютерлік технологияны жобалаудың көп еңбегін қажет ететін процестердің бірі. Қиындық осы кезге дейін сызықты емес электр тізбектерін талдаудың (немесе есептеудің) әдістері жеткіліксіз болып табылады. Есептеулер, әдетте, жеңілдетіліп, схеманың жеке жұмыс режимдеріне арналған.



Сурет 13.6. Техникалық дизайн және осы параметрлерге төзімділік.

Математикалық модельдеу үшін электр тізбектерінің цифрлық модельдері кеңінен қолданылады. Сандық модель дегеніміз - берілген элементтің қабылданған физикалық моделінің математикалық сипаттамасы. Компьютерлік технологияның тізбек элементтеріне арналған сандық модельдерді құрастырудағы негізгі қиындық - сызықтық емес элементтердің көп болуы.

Зертханалық прототиптеу негізгі компоненттердің жұмысын тексеруге және монтаж тізбегінің мүмкіндіктерін анықтауға арналған.

Дизайнердің жұмысы электр тізбектерінің массалық элементтерін талдаумен байланысты. Осы талдау негізінде типтік құрылымдық элемент немесе типтік ауыстыру элементі - модуль немесе ұяшық таңдалады.

**Тест сұрақтары:**

1. Компьютерлік құрылғыларды жобалаудың қандай кезеңдерін білесіз?
2. Компьютерлік техниканы жобалау және өндіру кезеңдері қандай?
3. Компьютерлік техниканы жобалау мен өндірудің дайындық кезеңінде қандай міндеттер шешіледі?
4. Алдын ала жобалау сатысында не жүзеге асырылады?
5. Техникалық жобалау сатысында қандай міндеттер шешіледі?
6. Неліктен іс-әрекеттегі қадамдардың реттілігі жиі бұзылады?





















Кашаганова Г.Б.

Есептеу жүйелерінің архитектурасы және ұйымдастырылуы

Оқу-әдістемелік құрал

Пішімі 60x84, 1/16.  
Тығыздығы 80 г/м<sup>2</sup>.  
Көлемі 3.2