

Конвейерлік ұйымдастыру және конвейерлік өңдеу принциптері:

**Конвейердегі конфликт түсінігі
және оларды шешу жолдары:**

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

Конвейерлеу

Конвейерлеу (немесе конвейерлік өңдеу) жалпы жағдайда баспалдақ деп аталатын орындалуы керек функциялардың кіші бөліктерге бөлінуіне негізделген, әрі олардың арқайсысына жеке аппаратура блогын ерекшелеуі. Осылайша кез келген машиналық команданың орындалуын бірнеше кезеңге немесе баспалдаққа бөлуге болады, ол кезде деректерді жіберу бір кезеңнен екінші кезеңге жалғасып отырады. Ескерте кетейік конвейерлік өңдеу кезінде сондай ақ әртүрлі командалардың орындалу кезеңдерін де біріктіруге болады. Себебі өнімділік бір уақытта бірнеше кезеңде бірнеше команданы орындай алу мүмкіндіктерінің берілгендігімен өседі. Мұндай конвейерлік өңдеулер қазіргі кездегі процессорларда кеңінен қолданылады.

Қарапайым команда конвейері және оның тиімділігін бағалау

Процессорларды құрудың негізгі принциптерін суреттеу үшін біз 32 жалпы мақсаттағы бүтін регистрлер ($R_0, \dots, R_{31}$), 32 өзгермелі нүкте регистрлері ($F_0, \dots, F_{31}$) және ДК командаларының есептегішін қамтитын қарапайым архитектураны қолданамыз. Біздің процессордың командалар жинағы типтік арифметикалық және логикалық операцияларды, өзгермелі нүкте операцияларын, деректерді беру операцияларын, команда ағынын басқару операцияларын және жүйе операцияларын қамтитынын қарастырайық. Арифметикалық нұсқаулар RISC процессорларына тән үш адрестік форматты пайдаланады, ал жадыға қол жеткізу жадыға жүктеу және жазу регистрінің операцияларын пайдаланады.

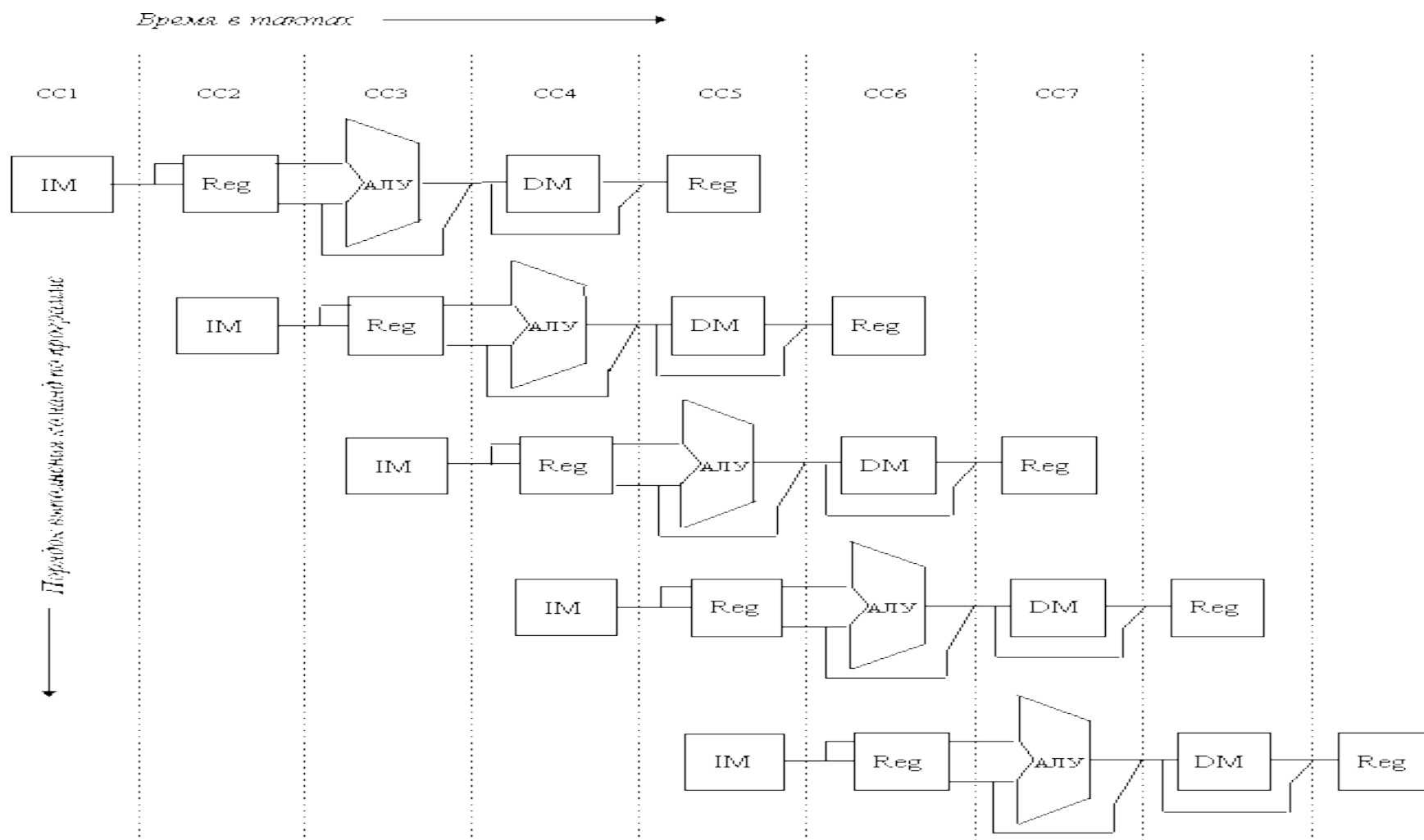
MIPS ассемблері бес кезеңнен тұратын классикалық RISC конвейер үлгісін пайдаланады:

1. **Fetch** (алу): Жадтан нұсқауды қабылдау және оны процессордың командалар регистріне жүктеу.
2. **Decode** (декодтау): нұсқауды декодтау және регистрлерді оқу.
3. **Execute** (орындау): операцияны орындау немесе мекенжайды есептеу.
4. **Memory** (жад): жадқа оқу немесе жазу рұқсаты.
5. **Write back** (write): Нәтижені регистрге жазыңыз.

Қарапайым команда конвейері және оның тиімділігін бағалау

Әдеттегі команданың орындалуын келесі қадамдарға бөлуге болады:

- команданы таңдау – **IF** (бағдарлама есептегіші көрсеткен мекенжай бойынша команда жадыдан алынады);
- команданы декодтау/регистрлерден операндтарды таңдау – **ID**;
- операцияны орындау / тиімді жады мекенжайын есептеу – **EX**;
- жадыға қатынау – **MEM**;
- нәтижені есте сақтау – **WB**



Конвейер жұмысын уақыт диаграммасы түрінде көрсету

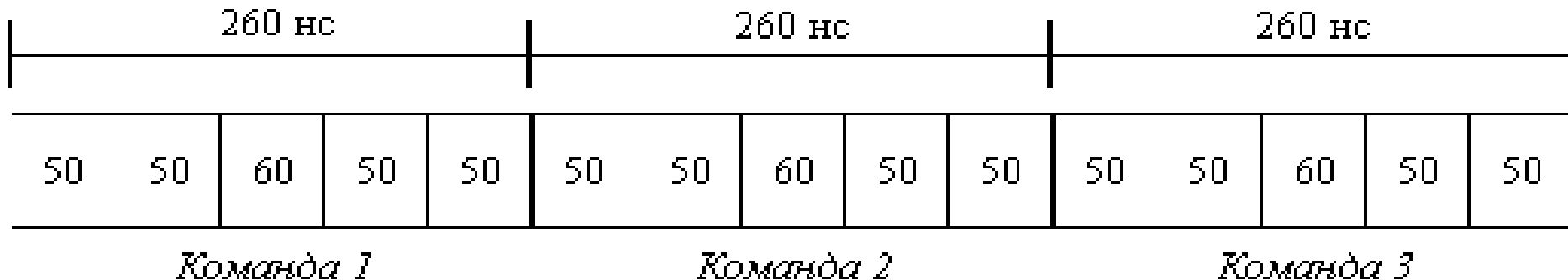
Команда нөмірі	Такт нөмірі								
	1	2	3	4	5	6	7	8	9
Команда i	IF	ID	EX	MEM	WB				
Команда i+1		IF	ID	EX	MEM	WB			
Команда i+2			IF	ID	EX	MEM	WB		
Команда i+3				IF	ID	EX	MEM	WB	
Команда i+4					IF	ID	EX	MEM	WB

Конвейерлік өңдеу тиімділігі

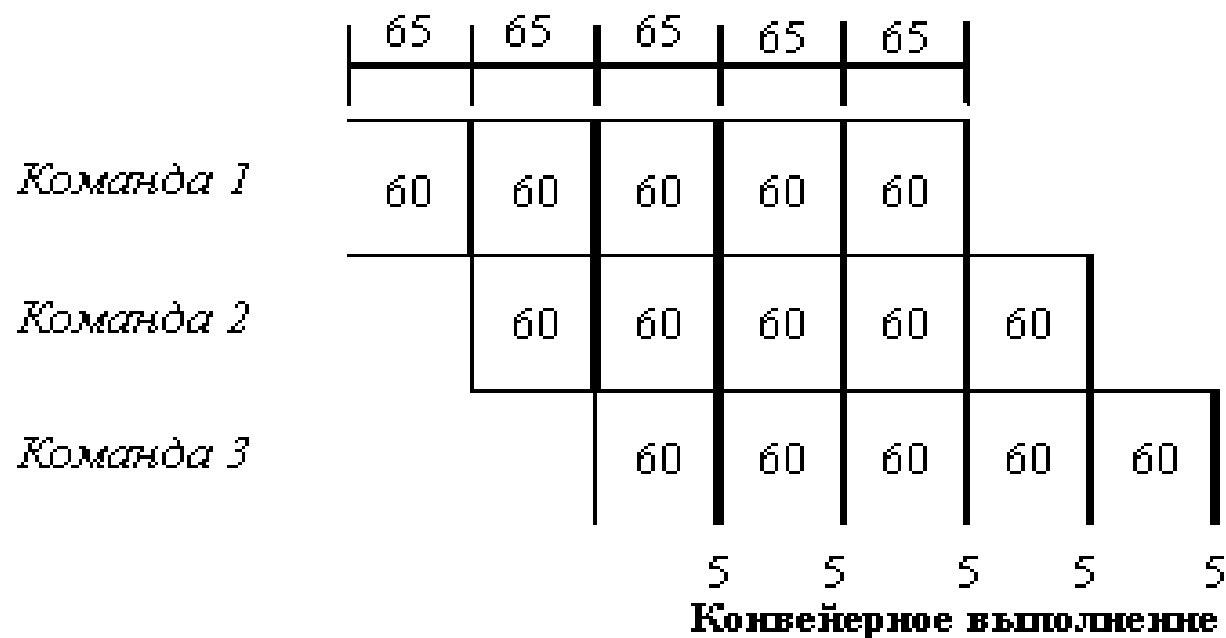
Мысал ретінде, сәйкесінше ұзақтығы 50, 50, 60, 50 және 50 нс болатын операциялардың бес кезеңі бар конвейерсіз машинаны қарастырайық. Конвейердің үстіңгі жағы 5 нс болсын. Сонда конвейерсіз машинада команданың орташа орындалу уақыты 260 нс болады. Егер конвейерді ұйымдастыру пайдаланылса, цикл ұзақтығы өңдеудің ең баяу кезеңінің ұзақтығына және үстеме шығындарға тең болады, яғни. 65 нс Бұл жолы конвейердегі команданың орташа орындалу уақытына сәйкес келеді. Осылайша, конвейер пайдалану нәтижесінде алынған үдеу $260/65=4$ қатынасына тең болады. Бұл сәйкесінше:

Конвейерсіз өңдеу уақыты: 260 нс
Конвейерлік өңдеу уақыты: 65 нс

Конвейерлік жылдамдату



Неконвейерное выполнение



Конвейерлеу

Конвейерде өңдеудің маңызды сипаттамасы конвейердегі нұсқауларды орындау үшін тактілердің орташа саны (CPI) болып табылады:

Конвейердің $CPI = \text{Идеалды конвейер CPI} +$
 $RAW \text{ тиіндегі конфликт үшін тоқтатылуы} +$
 $WAR \text{ тиіндегі конфликт үшін тоқтатылуы} +$
 $WAW \text{ тиіндегі конфликт үшін тоқтатылуы} +$
Басқару бойынша конфликт үшін тоқтатылуы да әсер етеді

Конвейерлеу деңгейлері

- Макроконвейер – процессорлар деңгейіндегі конвейеризация
- Команда конвейері – процессор командаларының конвейеризациясы
- Арифметикалық конвейер – процессор командаларын орындау деңгейіндегі конвейеризация

Конвейердегі конфликтілер түсінігі

Оларды жою жолдары

1. Аппараттық құралдардың қабаттасумен бір уақытта орындау режимінде командалардың барлық мүмкін комбинацияларына қолдау көрсете алмаған кезде ресурс конфликтілерінен туындайтын құрылымдық конфликтілер.
2. Бір команданың орындалуы алдыңғы команданың нәтижесіне байланысты болған кезде пайда болатын деректер конфликтілері.
3. Команда санағышының мәнін өзгертетін конвейеризация кезіндегі бір командадан екінші бір командаға өту арқылы пайда болатын басқару конфликтілері.

Конвейердегі конфликтілер түсінігі

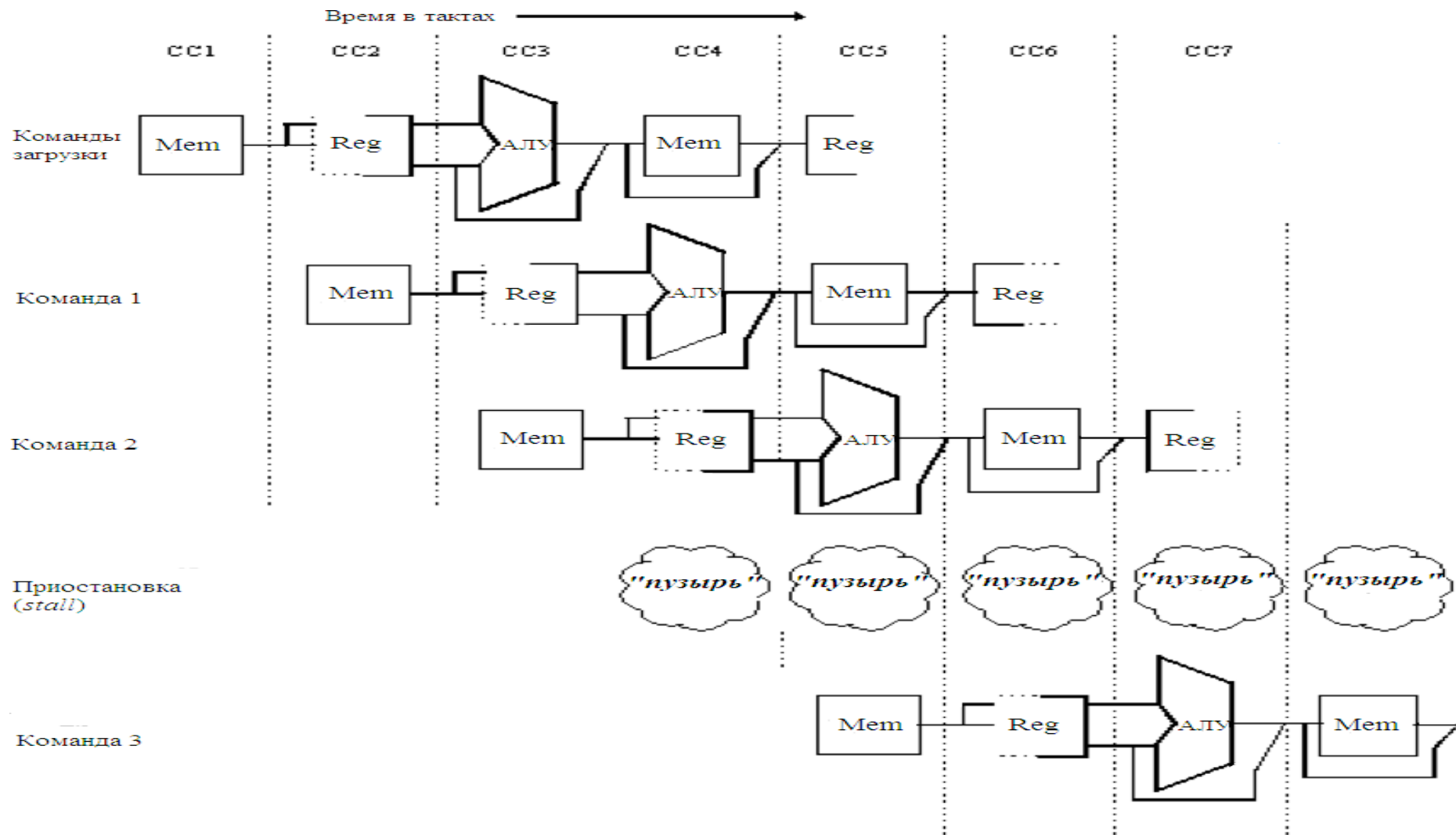
Оларды жою жолдары

Конвейердегі қайшылықтар командалардың орындалуын тоқтата тұру қажеттілігіне әкеледі (pipeline stall).

Әдетте қарапайым конвейерлерде кез келген команда тоқтатылған болса, одан кейінгі барлық командалар да тоқтатылады.

Бұл кезде тоқтатылған команданың алдындағы команда өз жұмысын жалғастыра алады, бірақ тоқтатылған кезде ол командалар шығарылмай қалып калады.

Бір портты жадыны ұйымдастыру кезіндегі құрылымдық конфликт мысалы



Құрылымдық конфликт кезіндегі конвейер жұмысының диаграммасы

Команда	Такт нөмірі									
	1	2	3	4	5	6	7	8	9	10
Жүктеу командасы	IF	ID	EX	MEM	WB					
Команда 1		IF	ID	EX	MEM	WB				
Команда 2			IF	ID	EX	MEM	WB			
Команда 3				stall	IF	ID	EX	MEM	WB	
Команда 4						IF	ID	EX	WB	
Команда 5							IF	ID	EX	WB
Команда 6								IF	ID	EX

Деректер бойынша конфликтілер

Деректер конфликті конвейерді өңдеуді пайдаланған кездегі операндтардың қатынау тәртібін өзгерте алатын кезде қолданылады, осылайша бұл тәртіп конвейерсіз машинада командалар тәртібі бір тізбекпен орындалған кездегіден ерекшеленеді. Келесі суреттерде осы ерекшеліктерді қарастырайық.

Мұнда барлық ADD командасынан кейінгі командалар осы команданың нәтижесіне байланысты орындалуы керек болсын. Сол себепті ADD командасы нәтижені R1 регистріне жазады, ал SUB командасы ол мәнді оқиды. Егер бұл жағдайда пайда болатын конфликтіні алдын алмаса онда SUB командасы дұрыс емес деректі оқып оны орындауы мүмкін.

Деректер бойынша конфликтілер

ADD	R1, R2, R3	IF	ID	EX	MEM	WB				
SUB	R4, R4, R5		IF	ID	EX	MEM	WB			
AND	R6, R1, R7			IF	ID	EX	MEM	WB		
OR	R8, R1, R9				IF	ID	EX	MEM	WB	
OR	R10, R1, R11					IF	ID	EX	MEM	WB

Деректер бойынша конфликтілер

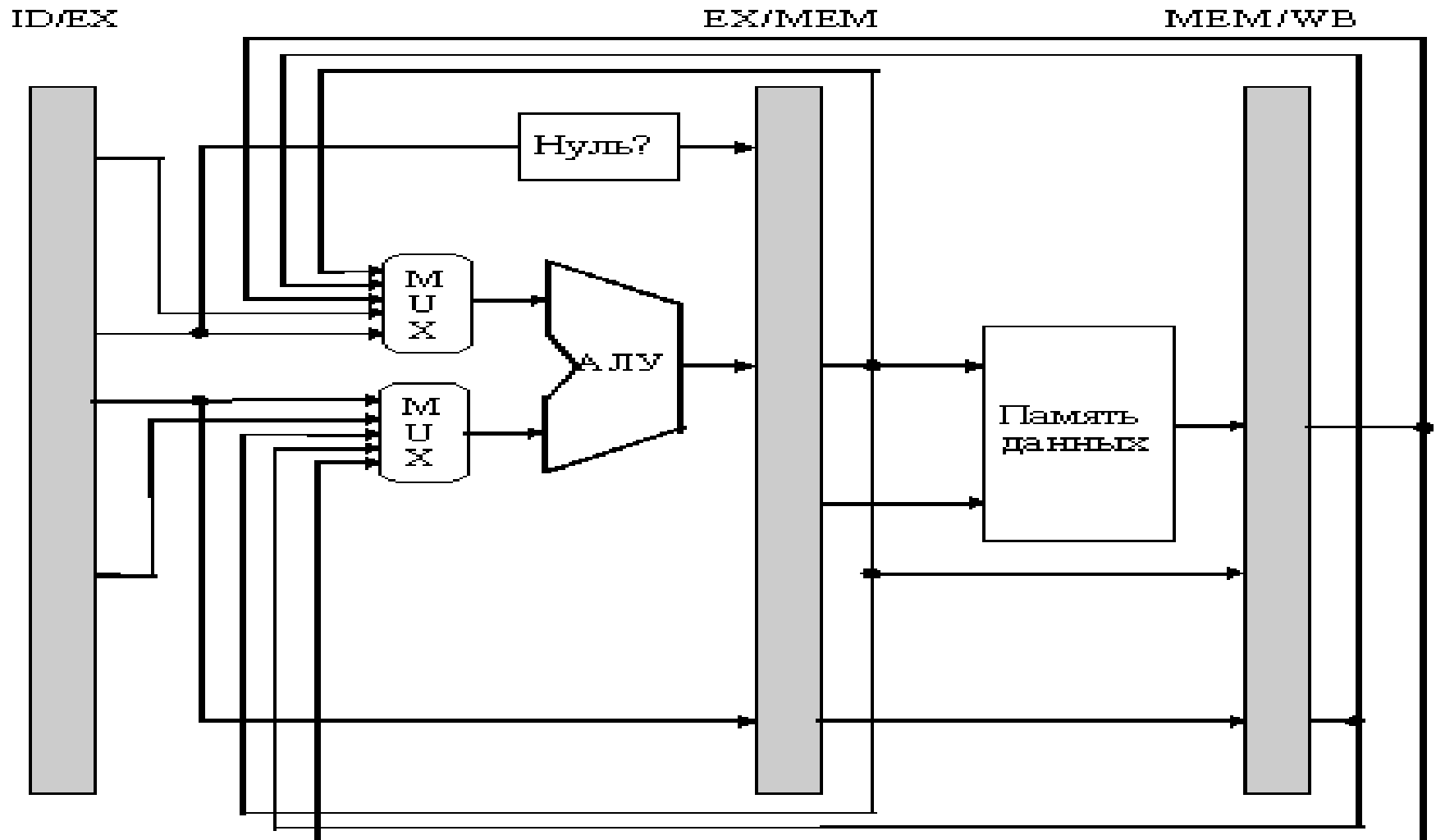
ADD	R1, R2, R3	IF	ID	EX	MEM	WB				
			R			W				
SUB	R4, R4, R5		IF	ID	EX	MEM	WB			
				R			W			
AND	R6, R1, R7			IF	ID	EX	MEM	WB		
					R			W		
OR	R8, R1, R9				IF	ID	EX	MEM	WB	
						R			W	
OR	R10, R1, R11					IF	ID	EX	MEM	WB
							R			W

Деректер бойынша конфликтілерді жою

Бұл құрылғы келесідей жұмыс істейді:

- ALU операциясының нәтижесі оның шығыс регистрінен әрқашан ALU кірістеріне қайта жіберіледі.
- егер аппараттық құрал алдыңғы ALU операциясының нәтижені келесі ALU операциясы үшін операндтың өзіне сәйкес регистрге жазғанын анықтаса, онда басқару логикасы сұлбасы нәтижені ALU емес, кіріс ретінде «айналып өту» тізбегінен, регистр файлынан оқылатын мәннен тандайды.

Деректер бойынша конфликтілерді жою



Деректер бойынша конфликтілердің классификациясы

Оқу және жазу амалдарының орындалу ретіне қарай келесі үш топқа бөлу әдетке айналған. *j* алдында *i* келетін *i* және *j* екі командасын қарастырайық.

- RAW (жазғаннан кейін оқу) – жазғаннан кейін оқу; *j* деректер көзінің операнды мен оған жазбас бұрын оны оқуға тырысады. Осылайша, *j* ескі мәнді қате алуы мүмкін. Бұл қақтығыстардың ең көп тараған түрі, оларды «айналып өту» механизмінің көмегімен жеңу жолы бұрын талқыланған болатын.
- WAR – оқығаннан кейін жазу; *j* нәтижені *i* пәрмені арқылы оқуға дейін тағайындалған жерге жазуға тырысады, сондықтан мен жаңа мәнді дұрыс ала алмауым мүмкін. Әсіресе мұндай қақтығыстар командалардың программа кодында орналасу ретінен тыс орындалуына мүмкіндік беретін жүйелерде жиі кездеседі.
- WAW – жазудан кейін жазу; *j* *i* нұсқауының нәтижесі жазылмай тұрып операндты жазуға тырысады, яғни. жазбалар қабылдағыштағы мәнді қалдырып, қате тәртіппен аяқталады *j* емес, *i* командасы арқылы жазылады. Қақтығыстың бұл түрі тек көптеген кезеңдерден тұратын конвейерлерде болады.

Басқару бойынша конфликтілер

Деректер конфликтінен де жоғары конвейер өнімділігінің жоғарылауын тудыруы мүмкін. Шартты көшу командасы орындалғанда, ол команда санағышының мәнін өзгертуі немесе өзгертпеуі мүмкін. Егер шартты тармақталған нұсқау команданың санағышының мәнін ауыстырса, онда тармақтың орын алғаны айтылады; әйтпесе, ол орындалмайтын деп аталады. Шартты тармақтармен жұмыс істеудің ең қарапайым әдісі - шартты тармақты нұсқау кездескен бойда, ол жаңа программа санағышының мәнін есептейтін конвейер сатысына жеткенше конвейерді кідірту.

Басқару бойынша конфликтілер

Ауысу командасы	IF	ID	EX	ME M	WB					
Келесі команда		IF	stall	stall	IF	ID	EX	ME M	WB	
Келесі команда +1			stall	stall	stall	IF	ID	EX	ME M	WB
Келесі команда +2				stall	stall	stall	IF	ID	EX	ME M
Келесі команда +3					stall	stall	stall	IF	ID	EX
Келесі команда +4						stall	stall	stall	IF	ID
Келесі команда +5							stall	stall	stall	IF

Басқару бойынша конфликтілер болдырмау жолдары

Егер конвейер шартты өтудің командасында үш тактіге тоқтатылған болса онда ол машинаның өнімділігіне қатты әсер етеді. Сол себепті шартты өтулер кезіндегі жоғалтуларды азайту критикалық шешілмей тұрған маңызды мәселе болып табылады.

Конвейердің тоқтатылуын азайтатын бірнеше әдіс бар, олар:

- Күту әдісі;
- Қайтару әдісі;
- Орындалатын өтулерді болжау;
- Өтулерді кідірту.

Қарапайым конвейердегі ұзу себептері

Конвейер кезеңі	Ұзу себептері
IF	Команданы таңдау кезіндегі жады бетіне қатынау кезіндегі қатесі; жадыға теңестірілмей қатынау; жадыны қорғаудың бұзылуы
ID	Анықталмаған немесе қорғалған операция коды
EX	Арифметикалық ұзулер
MEM	Команданы таңдау кезіндегі жады бетіне қатынау кезіндегі қатесі; жадыға теңестірілмей қатынау; жадыны қорғаудың бұзылуы
WB	Жоқ

Ұзын конвейерлер

Бұған дейін қарастырған мысалдарымызда біз тек (**EX**) командасының орындалуы тек бір такт болатындарын қарастырдық. Бірақ жылжымалы нүктелі сандармен жұмыс жүргізген кезде тактілер саны көбірек болуы қажет. Ол логикалық сұлбалардың көп болғандығын немесе конвейер синхронизациясының тактілерінің артуына әкелер еді. Қарапайым тілмен аутар болсақ сол конвейр бірақ келесі өзгертулерімен қолданылады десек болады:

біріншіден, **EX такті** операцияны орындау үшін қанша қажет болса сонша рет қайталанып орындалады;

екіншіден, процессорда жылжымалы нүктелі сандармен жұмыс үшін бірнеше функциональды құрылғылар болады. Бірақ конвейер тоқтатылуы алдыңғы айтып өткеніміздей құрылымдық немесе деректер бойынша конфликтілердің пайда болуына байланысты болуы мүмкін.

Мұндай қосымшалары бар конвейерлерді **ұзын конвейерлер** деп атайды.

Ұзын конвейерлерді мысалмен қарастырып көрейік.

Ұзын конвейерлер

Айталық біздің таратылуымыздағы процессорда төрт жеке функциональды құрылғылар бар болсын:

- Негізгі бүтін санды құрылғы .
- Бүтін санды операндтар мен жылжымалы нүктелі сандарды көбейту құрылғысы.
- Жылжымалы нүктелі қосу құрылғысы.
- Бүтін санды операндтар мен жылжымалы нүктелі сандарды бөлу құрылғысы.

Бүтін санды құрылғылар барлық жүктеу және жадыға жазу командаларын, барлық бүтін санды операцияларды және барлық өту командаларын өңдейді.

Қосымша функциональды құрылғылары бар конвейер

