

ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

имени И.И. Воровича —

Архитектура компьютера и операционные системы

Лекция 9. Уровень микроархитектуры. Оптимизация

Андреева Евгения Михайловна

доцент кафедры информатики и вычислительного эксперимента



План лекции

- Микроархитектура
 - Микропрограммное управление Mic-1
 - Ускорение и оптимизация
 - Mic-2
 - Mic-3
- Домашнее задание

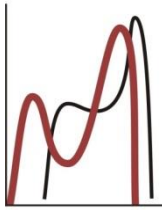


IJVM (повторение)

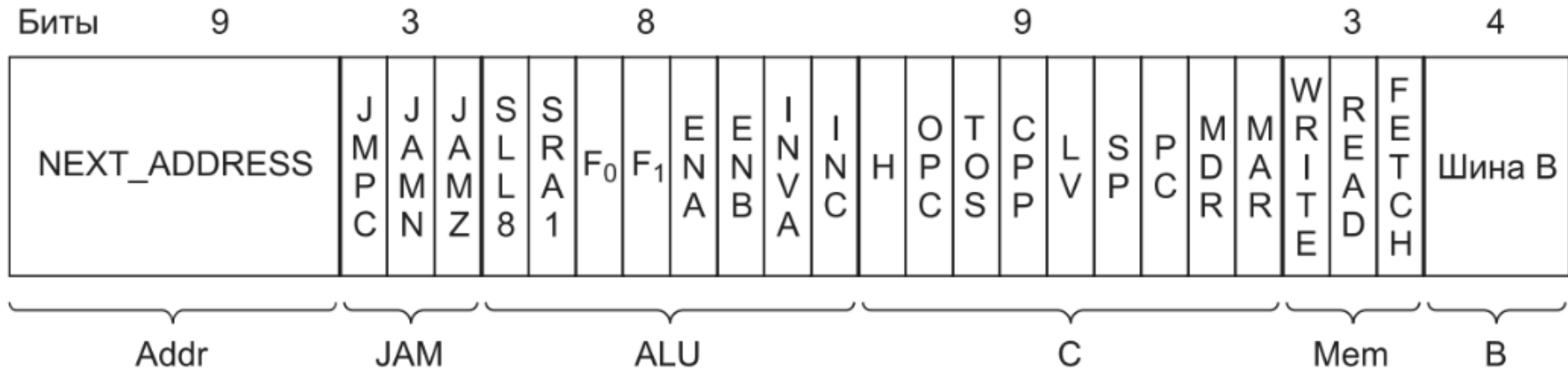
- IJVM является 32-разрядной архитектурой:
 - и числа, над которыми выполняются арифметические действия,
 - и адреса в памяти занимают 32 бита.
- Коды микрокоманд занимают 1 байт.
- IJVM – стековая машина



Полная диаграмма микроархитектуры Mic-1



Формат микрокоманды



- **Addr** — адрес следующей потенциальной микрокоманды;
- **JAM** — определение того, как выбирается следующая микрокоманда;
- **ALU** — функции АЛУ и схемы сдвига;
- **C** — выбор регистров, которые записываются с шины C;
- **Mem** — функции памяти;
- **B** — выбор источника для шины В (закодированный номер)

Регистры шины В

0 — MDR	5 — LV
1 — PC	6 — CPP
2 — MBR	7 — TOS
3 — MBRU	8 — OPC
4 — SP	9–15 — нет



Микропрограмма для Mic-1

- Main1 $PC = PC + 1$; fetch; goto (MBR)

- ILOAD k

iload1 $H = LV$

iload2 $MAR = MBRU + H$; rd

iload3 $MAR = SP = SP + 1$

iload4 $PC = PC + 1$; fetch; wr

iload5 $TOS = MDR$; goto Main1



Ускорение работы Mic-1

■ Параметры

- Число циклов на одну команду (другая шина В)
- Время выполнения одного цикла (декодер)
- Параллельность исполнения

■ Возможная реализация

- Блок предварительной выборки инструкций (ILU)
- 3-шинная архитектура
- конвейеризация



Дополнительные возможности ускорения

- Спекулятивное исполнение и прогнозирование ветвлений
- Введение доп. регистров (для спекулятивного исполнения)
- Изменение последовательности инструкций
- Кэш-память



Слияние заголовка цикла и микропрограммы

Main1 PC = PC + 1; fetch; goto (MBR)

...

...; goto Main1

■ Старая реализация POP

pop1 MAR = SP = SP - 1; rd

pop2

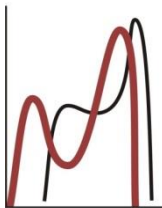
pop3 TOS = MDR; goto Main1

■ Новая реализация POP

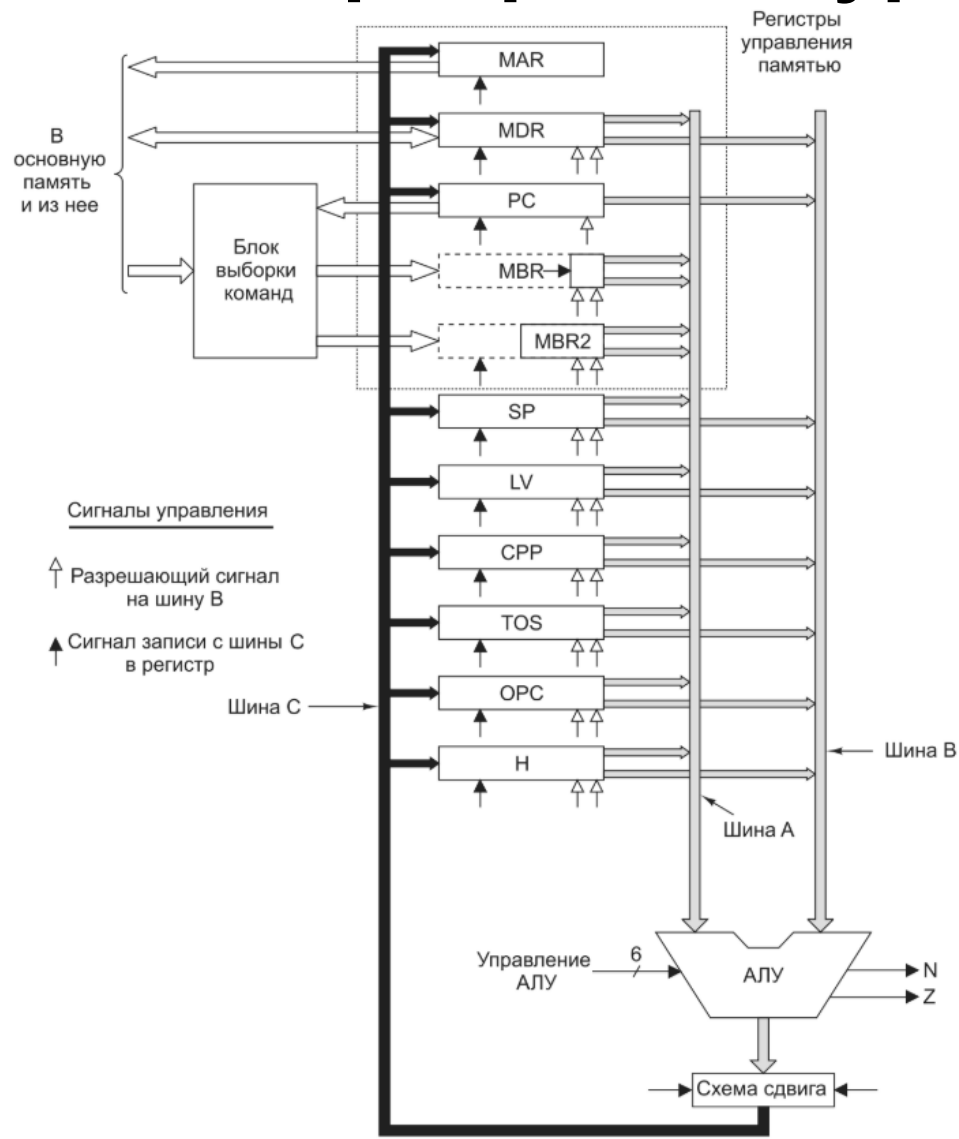
pop1 MAR = SP = SP - 1; rd

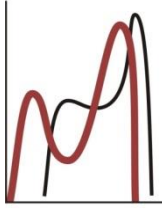
pop2 PC = PC + 1; fetch

pop3 TOS = MDR; goto
(MBR)



Трёхшинная микроархитектура Mic-2



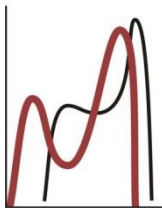


3-шинная микроархитектура

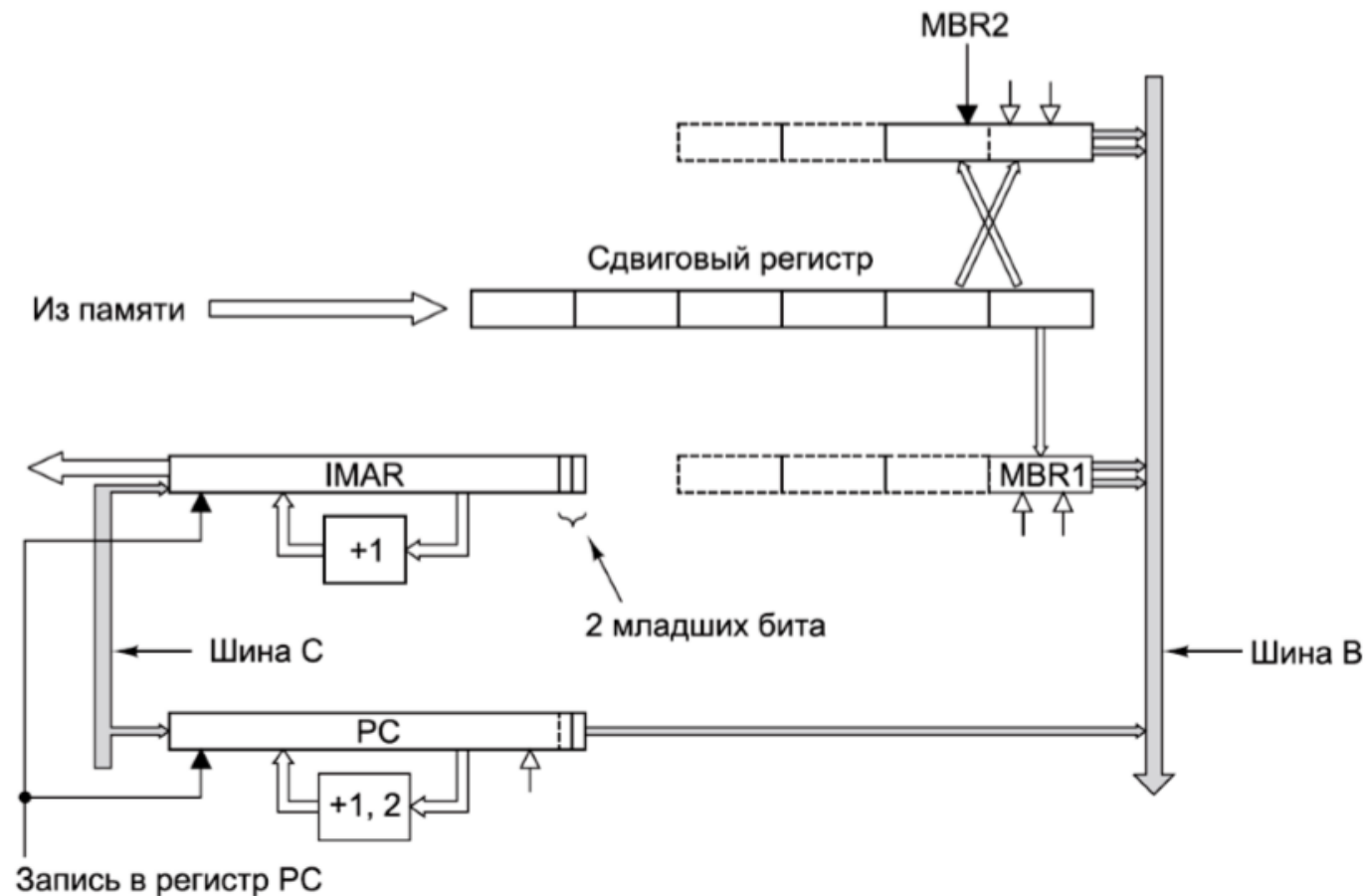
- реализация ILOAD на Mic-1
- реализация ILOAD на Mic-2

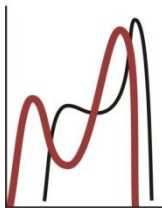
- 1 $H = LV$
- 2 $MAR = MBRU + H; rd$
- 3 $MAR = SP = SP + 1$
- 4 $PC = PC + 1; fetch; wr$
- 5 $TOS = MDR; goto Main1$

- 1 $MAR = MBRU + LV; rd$
- 2 $MAR = SP = SP + 1$
- 3 $PC = PC + 1; fetch; wr$
- 4 $TOS = MDR; goto Main1$

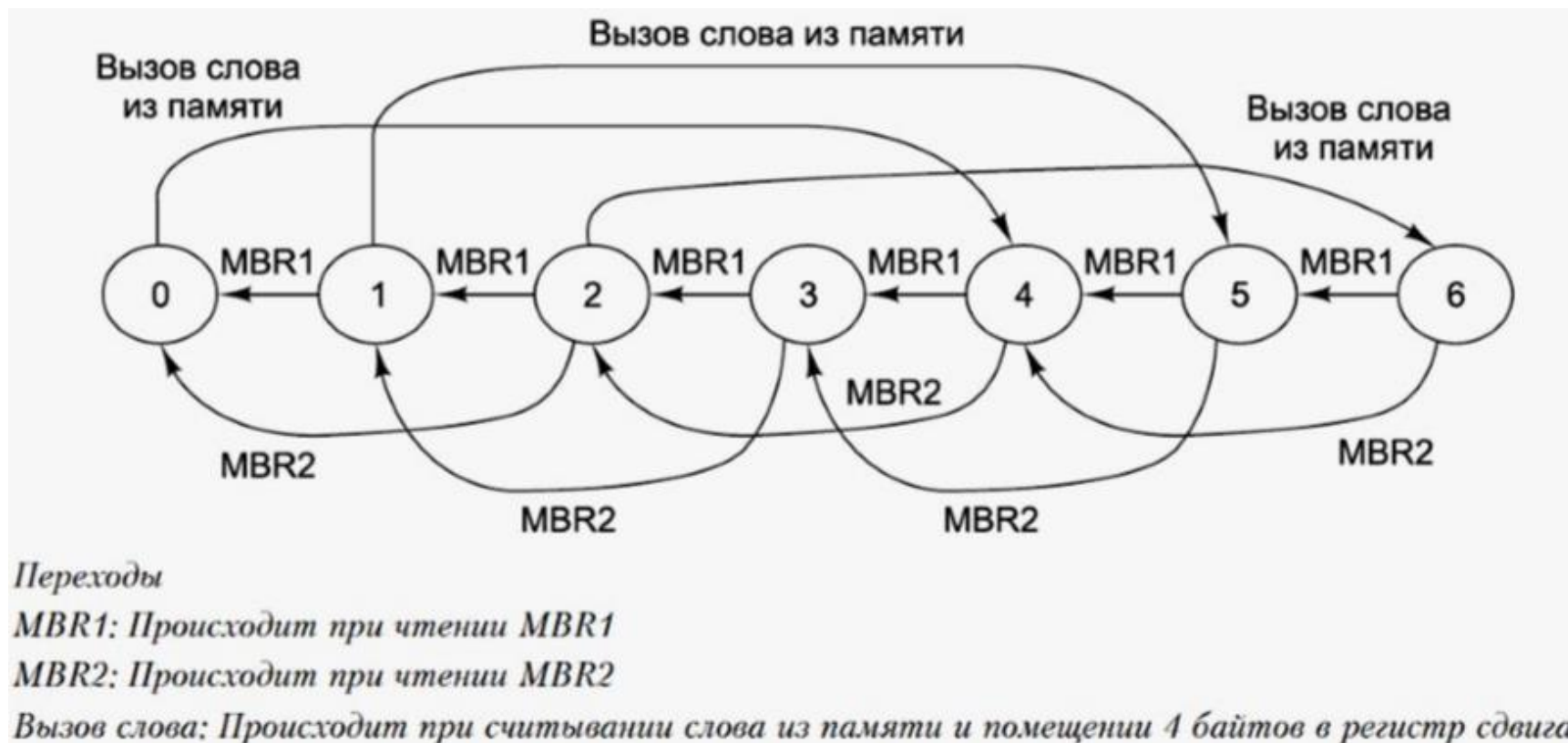


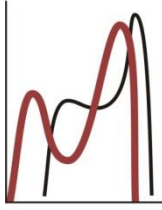
Логическая схема блока IFU





Конечный автомат для регистра сдвига в IFU





Сравнение реализаций ILOAD k

■ Старая реализация ILOAD k

- 1 $MAR = MBRU + LV; rd$
- 2 $MAR = SP = SP + 1$
- 3 $PC = PC + 1; fetch; wr$
- 4 $TOS = MDR; goto Main1$

■ Новая реализация ILOAD k

- 1 $MAR = MBR1U + LV; rd$
- 2 $MAR = SP = SP + 1$
- 3 $TOS = MDR; wr; goto (MBR1)$

■ Выполнение сократилось на 2 цикла тракта данных

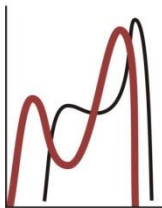


Сравнение реализаций IINC k n

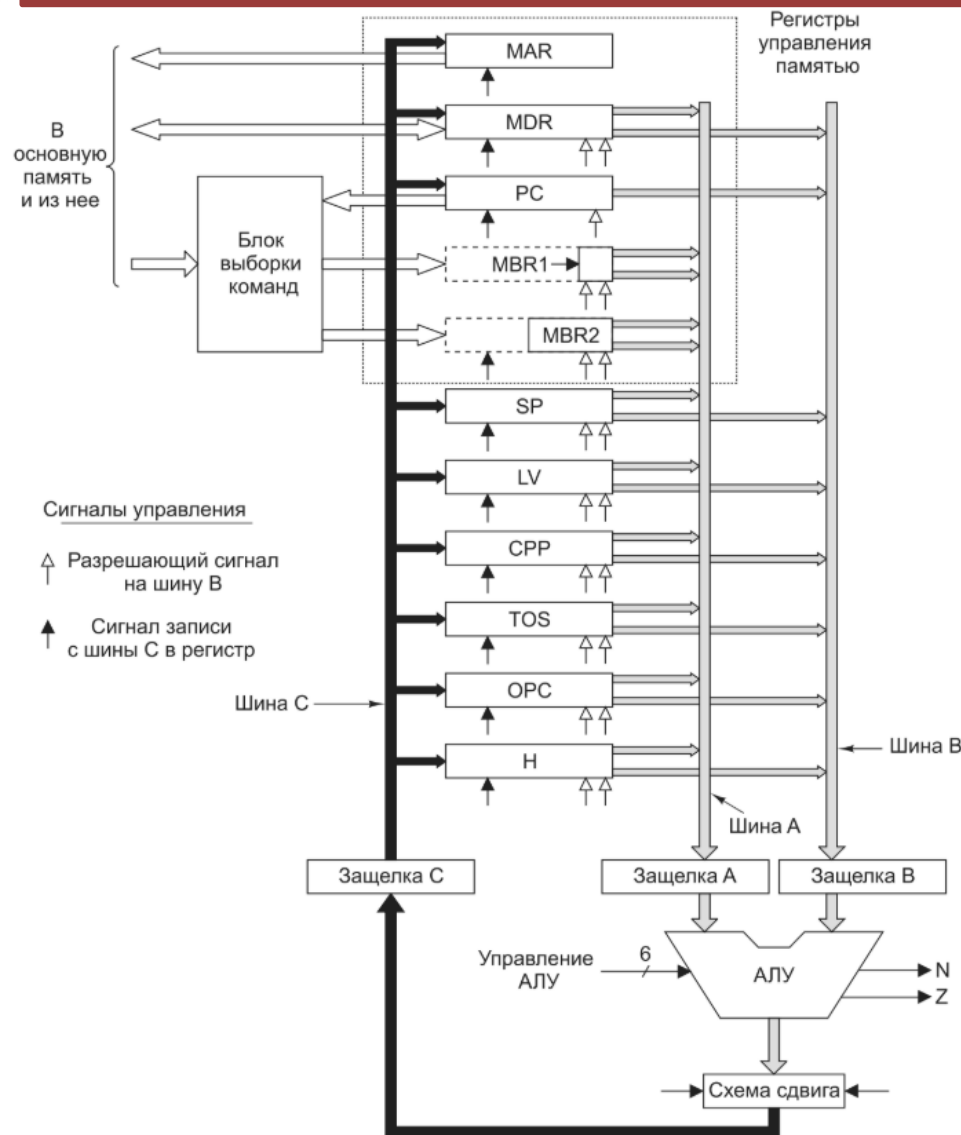
- Старая реализация IINC k n
- Новая реализация IINC k n

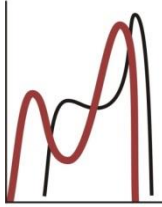
- 1 $H = LV$
- 2 $MAR = MBRU + H; rd$
- 3 $PC = PC + 1; fetch$
- 4 $H = MDR$
- 5 $PC = PC + 1; fetch$
- 6 $MDR = MBR + H; wr; goto$
Main1

- 1 $MAR = MBR1U + LV; rd$
- 2 $H = MBR1$
- 3 $MDR = MDR + H; wr; goto$
(MBR1)

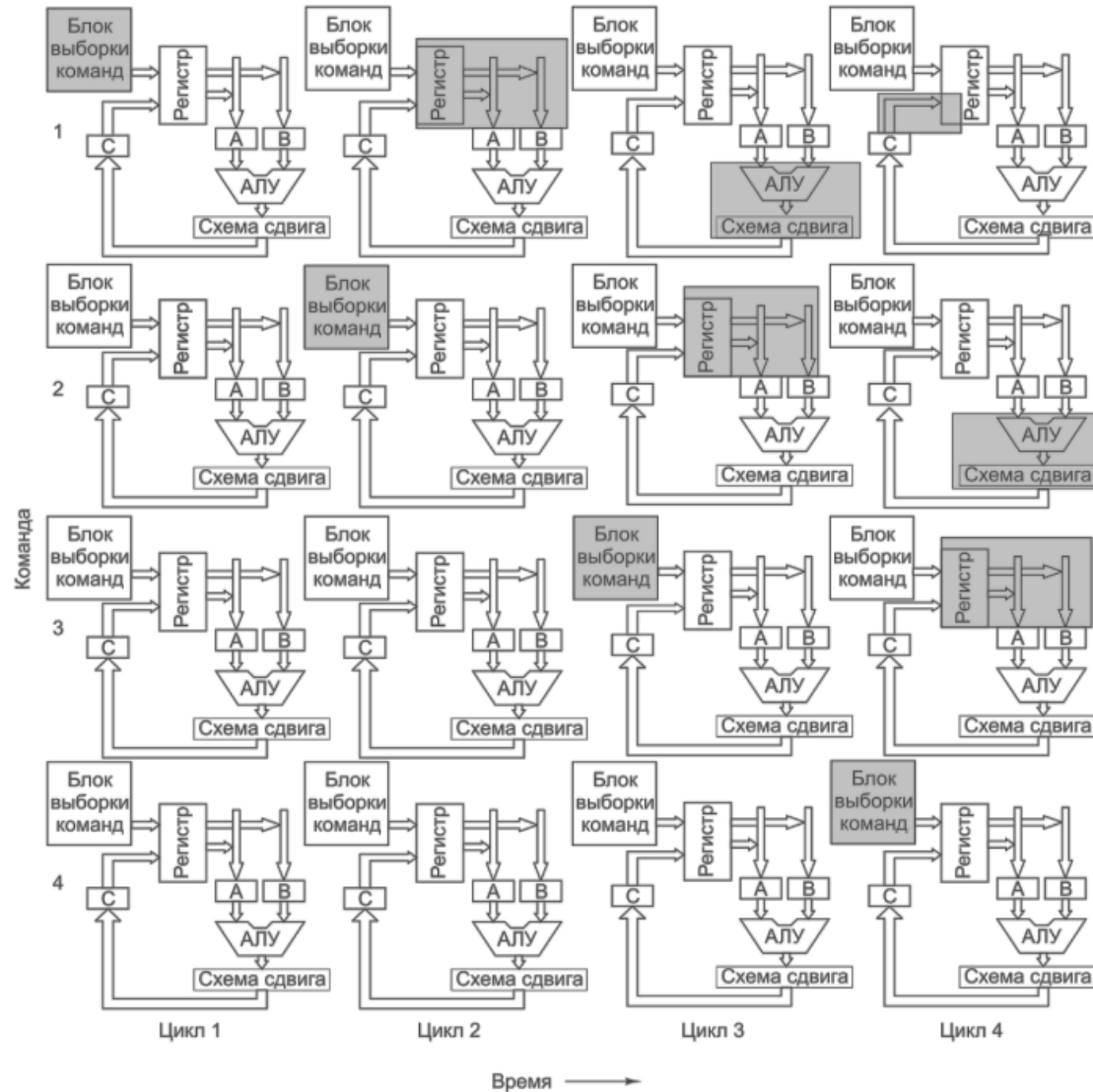


Микроархитектура Мис-3





Конвейер Мис-3

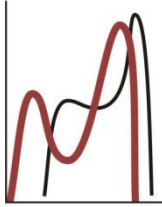




Команда SWAP

■ Старая реализация для MIC-1

swap1	$MAR = SP - 1; rd$
swap2	$MAR = SP$
swap3	$H = MDR; wr$
swap4	$MDR = TOS$
swap5	$MAR = SP - 1; wr$
swap6	$TOS = H; goto Main1$



Конвейерная реализация

	Swap1	Swap2	Swap3	Swap4	Swap5	Swap6
Цикл	MAR = SP - 1; rd	MAR = SP	H = MDR; wr	MDR = TOS	MAR = SP - 1; wr	TOS = H; goto (MBR1)
1	B = SP					
2	C = B - 1	B = SP				
3	MAR = C; rd	C = B				
4	MDR = Mem	MAR = C				
5			B = MDR			
6			C = B	B = TOS		
7			H = C; wr	C = B	B = SP	
8			Mem = MDR	MDR = C	C = B - 1	B = H
9					MAR = C; wr	C=B
10					Mem = MDR	TOS = C
11						goto (MBR1)



Домашнее задание

- Подготовка к тесту по лекции
 - раздел книги Таненбаума и Остина, пример архитектуры набора команд
 - IJVM (глава 4).