



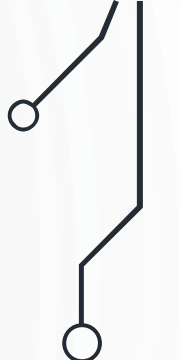
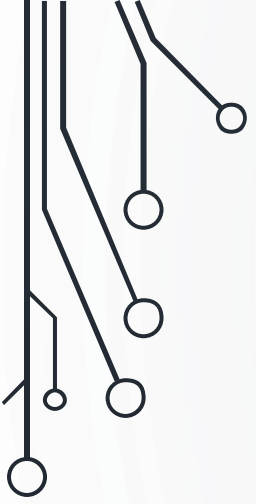
СиФО ВМ Л.Р. №4

Стракович Андрей Иванович

БГУИР, кафедра ЭВМ

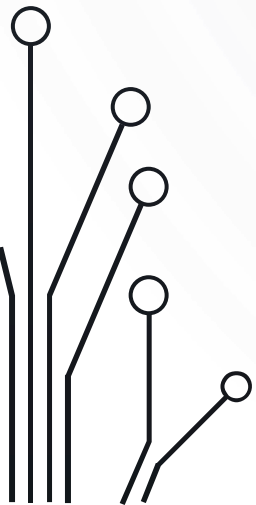
а. 508-5

2023



СЧИТЫВАНИЕ, ДЕКОДИРОВАНИЕ И ВЫПОЛНЕНИЕ КОМАНД. СПОСОБ АДРЕСАЦИИ ОПЕРАНДОВ В КОММАНДАХ.

SIFO LAB PRACTICUM PART 1 – Л.Р. 4



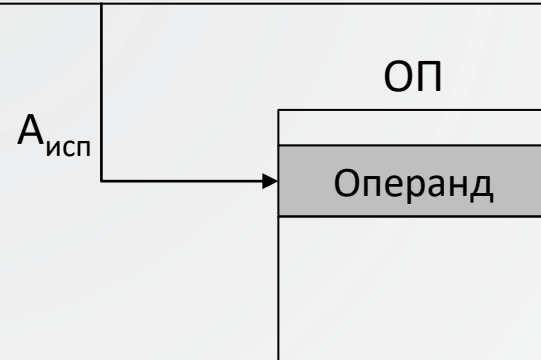
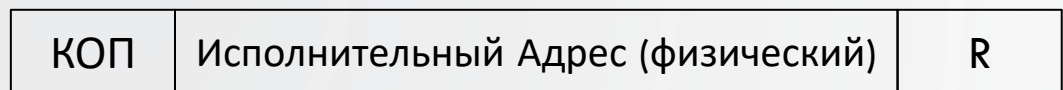
ДАННЫЕ ПО ВАРИАНТУ

- Разрядность ШД
- Разрядность ША
- Команда пересылки данных и адресация операнда-источника в ней:
 - Mem \rightarrow Reg или Reg \rightarrow Mem
- Команда условного или безусловного перехода и адресация исполнительного адреса в ней
 - JMP, JZ, JS

ВАРИАНТЫ КОМАНД

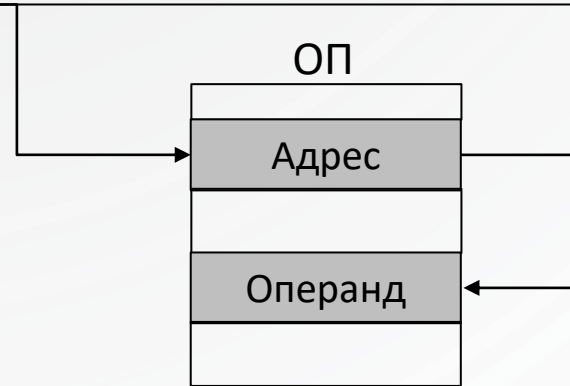
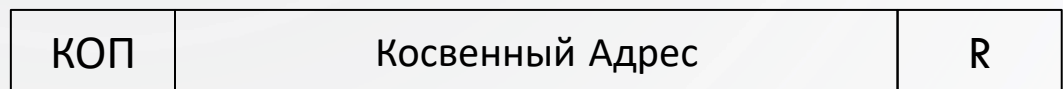
$M \rightarrow R$

Прямая адресация



$M \rightarrow R$

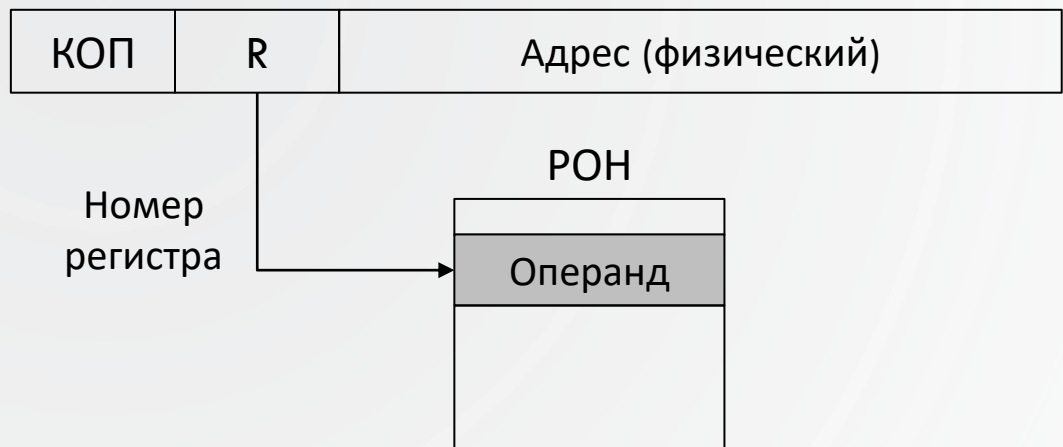
Косвенная адресация



ВАРИАНТЫ КОМАНД

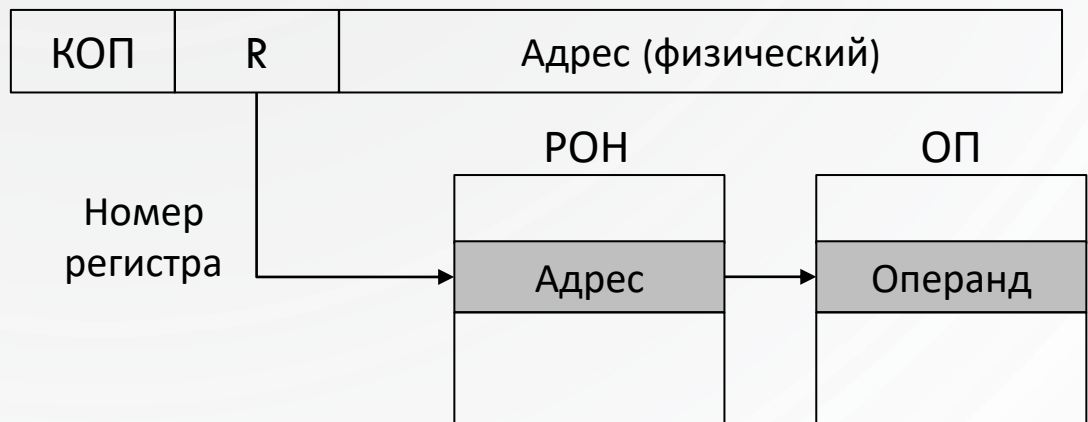
$R \rightarrow M$

Прямая регистровая



$R \rightarrow M$

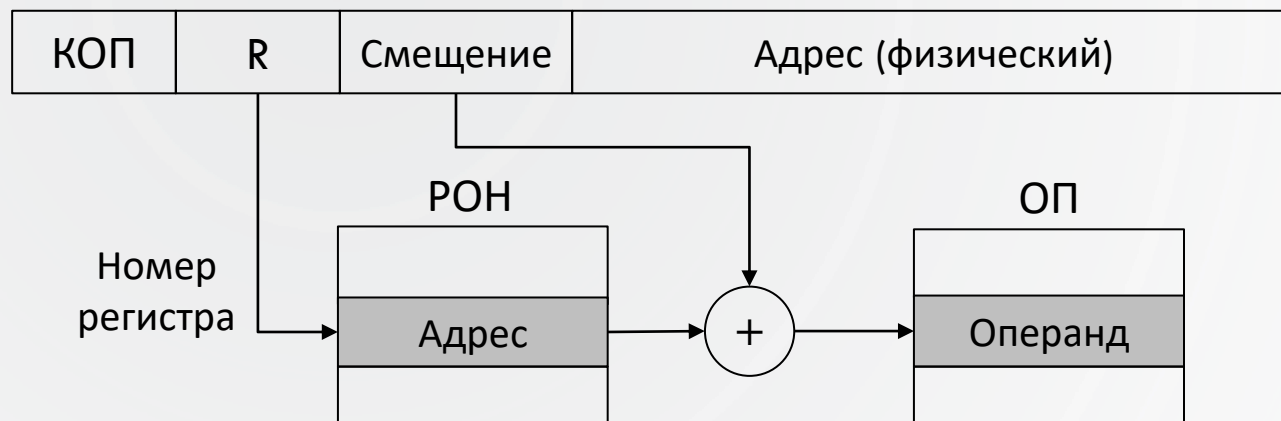
Косвенная регистровая



ВАРИАНТЫ КОМАНД

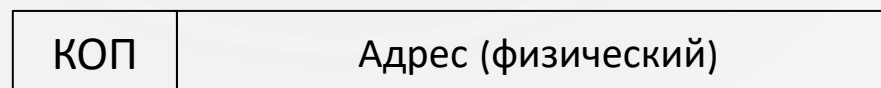
$R + C \rightarrow M$

Со смещением



$R \rightarrow M$

Неявная



При неявной адресации адресное поле отсутствует, а адрес операнда подразумевается кодом операции

ВАРИАНТЫ КОМАНД

Непосредственная

CONST \rightarrow R

КОП	Операнд	R
-----	---------	---

CONST \rightarrow M

КОП	Операнд	Адрес
-----	---------	-------

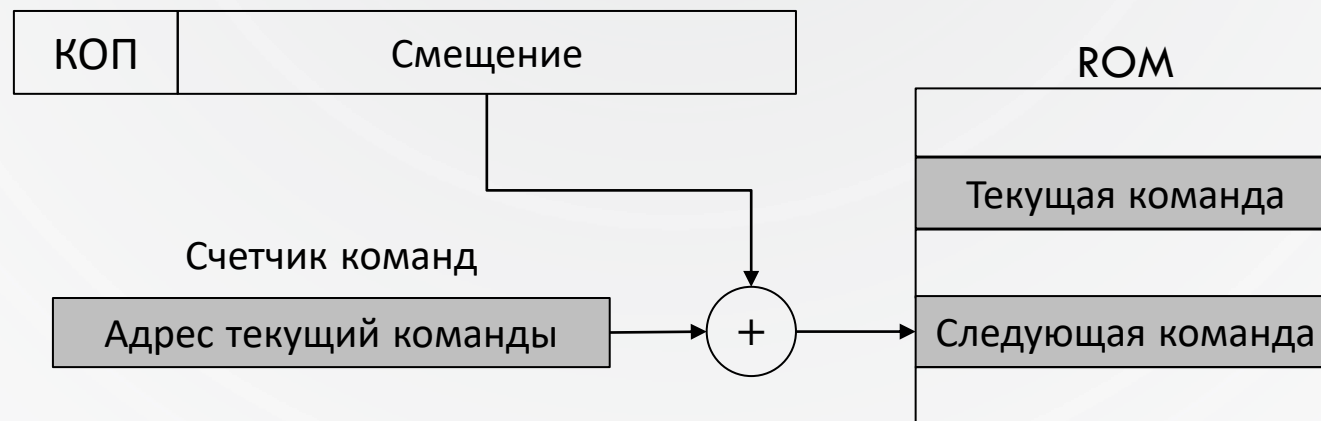
Jmp, Js, Jz

КОП	Исполнительный Адрес
-----	----------------------

ВАРИАНТЫ КОМАНД

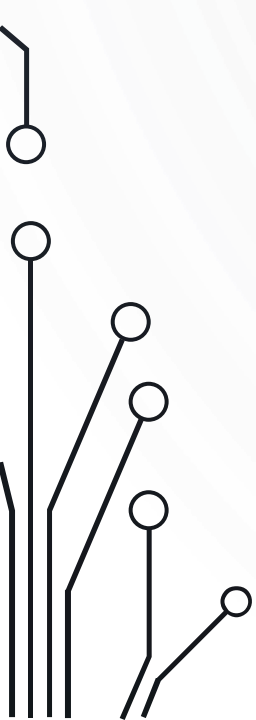
Jmp, Js, Jz

Относительная со смещением





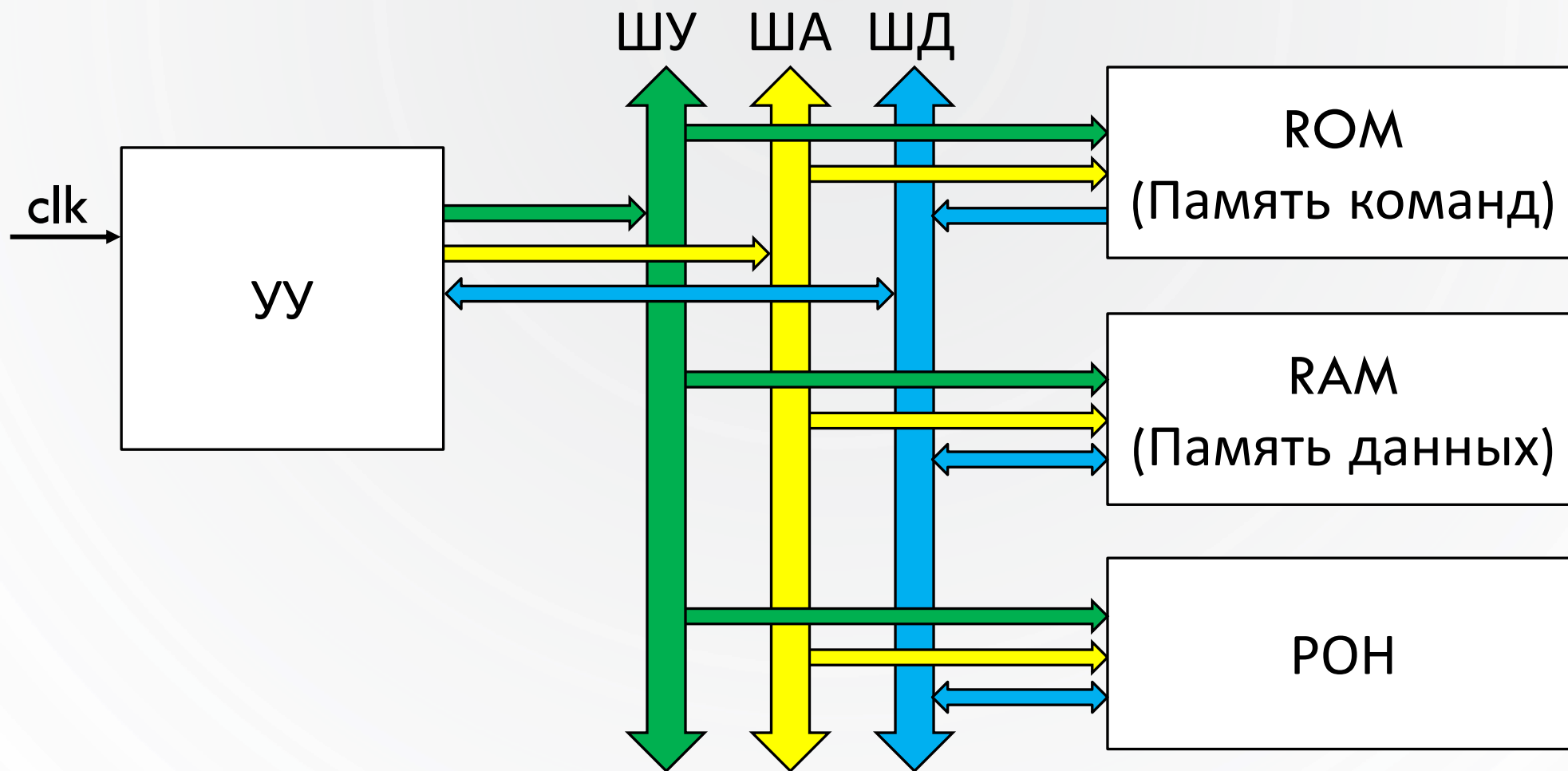
ЗАДАНИЕ

1. Разработать архитектуру системы команд (АСК) для команд выданных по варианту
 2. Ввести шину адреса (ША), шину данных (ШД) и шину управления (ШУ)
 3. Разделить память на память данных (блок RAM) и память команд (блок ROM). На адресные входы завести ША. Ввод и вывод данных осуществлять через ШД
 4. Ввести блок регистров общего назначения (РОН) и управляющую логику для него. Кол-во регистров 4-8 (на выбор)
 5. Написать микропрограмму (4-6 вызовов команд) в которой указать конкретные адреса памяти или регистров (пример будет)
- 

ЗАДАНИЕ

6. Записать микропрограмму в память команд (ROM) (в файл *.hex или *.mif)
7. Записать необходимые данные для микропрограммы в память данных (RAM) (в файл *.hex или *.mif)
8. Разработать устройство управления (УУ) которое будет считывать, декодировать и выдавать управляющие сигналы для выполнения полученной команды
9. Ввести специальные регистры, разрядность которых определяется разрядностью ШД. Физически разместить их в блоке управления
10. Промоделировать работу схемы

СТРУКТУРНАЯ СХЕМА

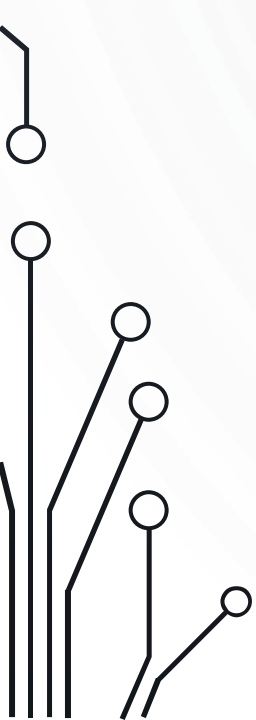
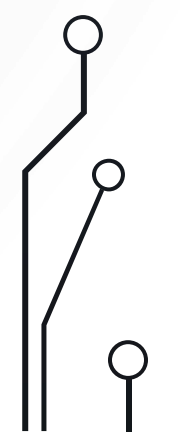


СПЕЦИАЛЬНЫЕ РЕГИСТРЫ

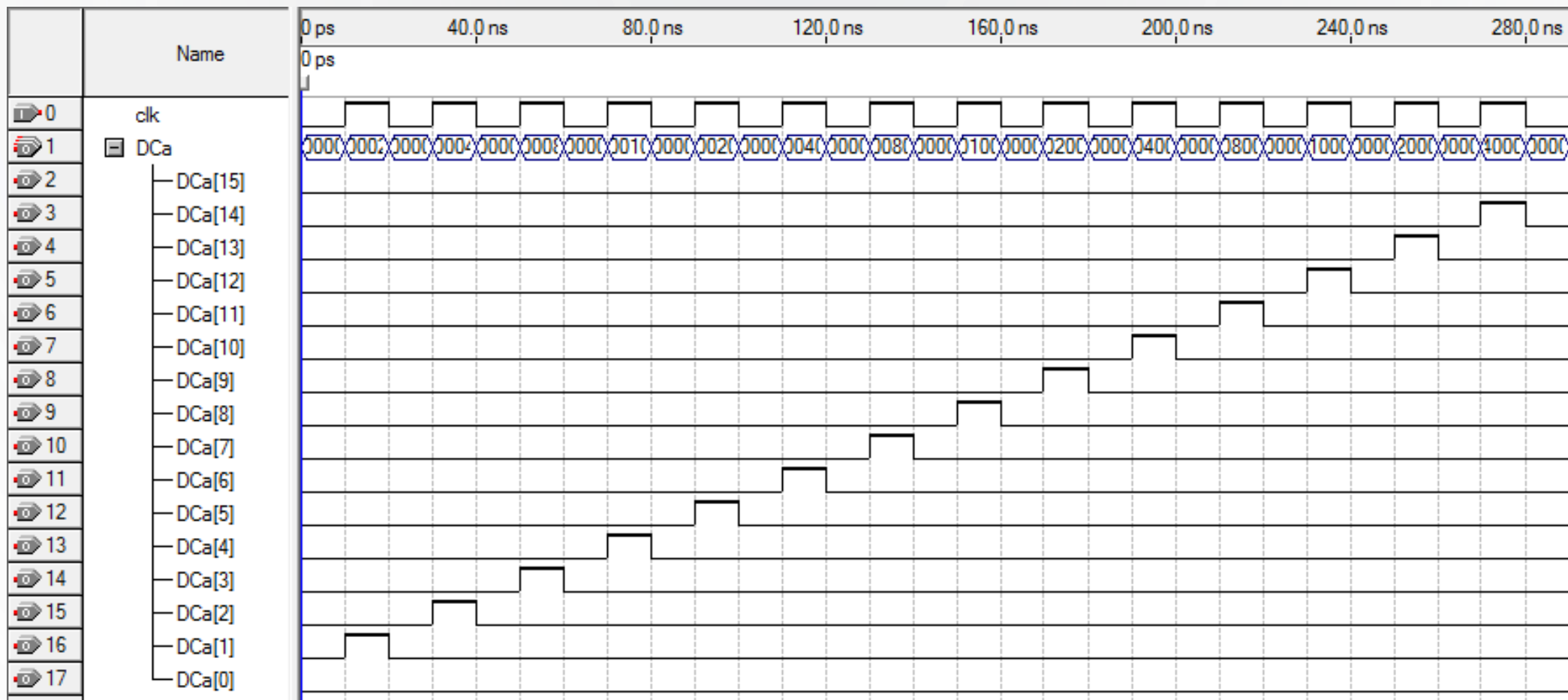
- Счетчик команды IP (Instruction Pointer) или PC (Program Counter)
- Регистр команд IR (Instruction Register)
- Регистр адреса операнда AR (Address Register)
- Регистр данных DR (Data Register)
- Регистр флагов FR (Flag Register)

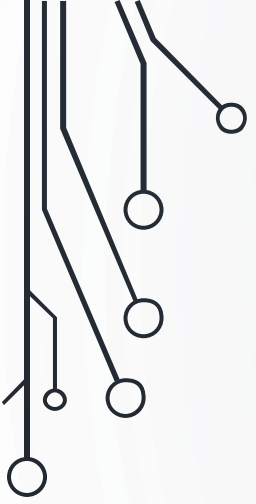


ЭТАПЫ ВЫПОЛНЕНИЯ КОМАНДЫ

- Выборка команды и формирование адреса следующей команды
 - Декодирование команды
 - Формирование исполнительных адресов операндов
 - Выборка операндов
 - Исполнение операции
 - Запись результата
- 
- 

ИСПОЛЬЗОВАНИЕ БЛОКА ОПОРНЫХ СИГНАЛОВ

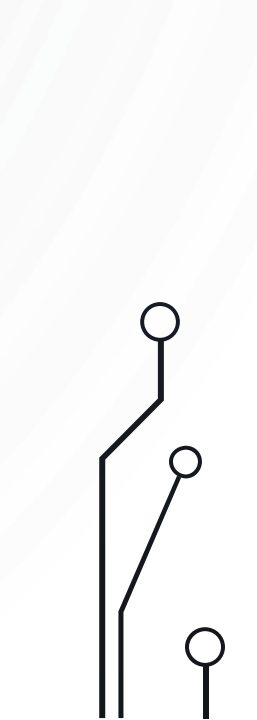
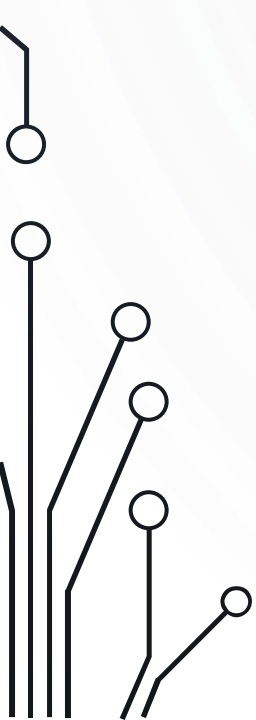




БАЛЛЫ ЗА Л.Р.

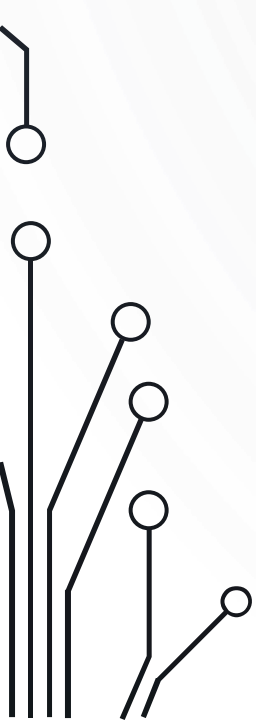
4-8 – за выполнение.

1 – за скорость сдачи.





СОДЕРЖАНИЕ ОТЧЁТА

- Титульный лист
 - Задание согласно варианту
 - АСК
 - Код программы в символьном и числовом виде
 - Содержание памяти до моделирования и после
 - Основные схемы
 - Моделирование схемы
- 

МИКРОПРОГРАММА

- Mov
- Mov
- Jmp
- Mov
- Jmp \$0

- Mov
- Jz (перехода нет)
- Jz (переход есть)
- Mov
- Jz \$0 (переход есть)

-
- A decorative graphic consisting of several vertical and diagonal lines of varying lengths, some ending in small circles, resembling a circuit board or a stylized tree structure.

РОН – 4 шт.

-
- A decorative graphic consisting of several vertical and diagonal lines of varying lengths, some ending in small circles, resembling a circuit board or a stylized tree structure.

РОН – 4 шт.

МИКРОПРОГРАММА

- Mov \$12, R1
- Mov \$79, R3
- Jmp \$52
- Mov \$43, R2
- Jmp \$0

ША – 7

ШД – 6

РОН – 4 шт.

КОП		х	REG		ADR						
11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	1	1	0	0
0	1	0	1	1	1	0	0	1	1	1	1
1	0	0	0	0	0	1	1	0	1	0	0
0	1	0	1	0	0	1	0	1	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0

01 – Mov

10 – Jmp