

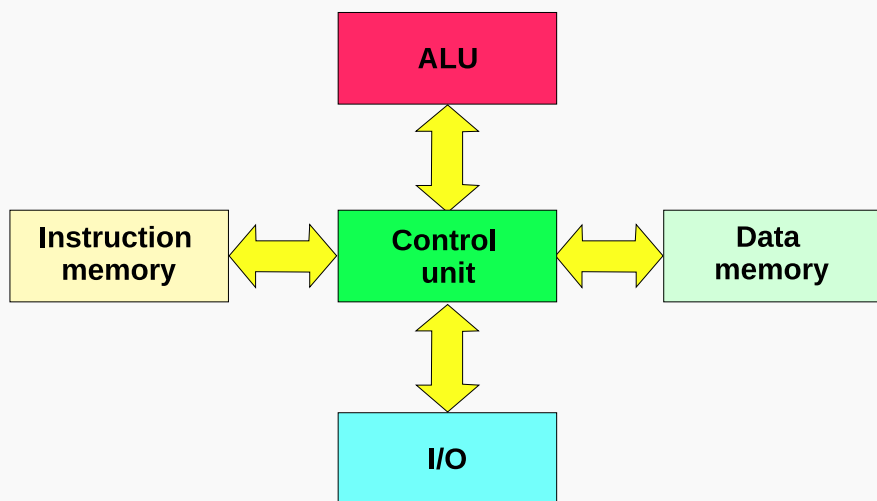
Общие сведения о БЭВМ

4

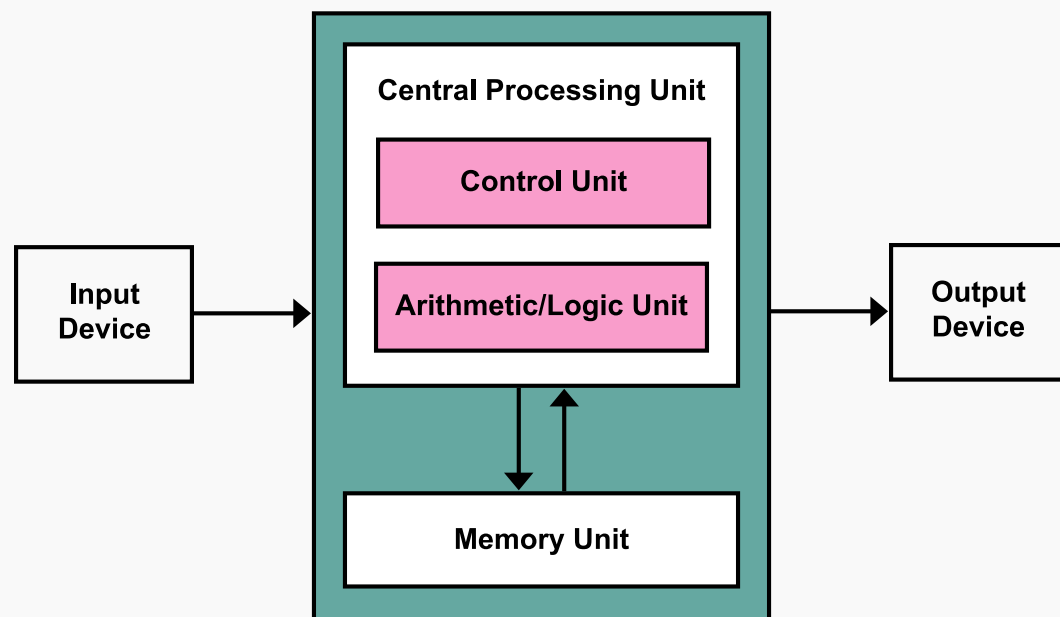


Архитектура ЭВМ

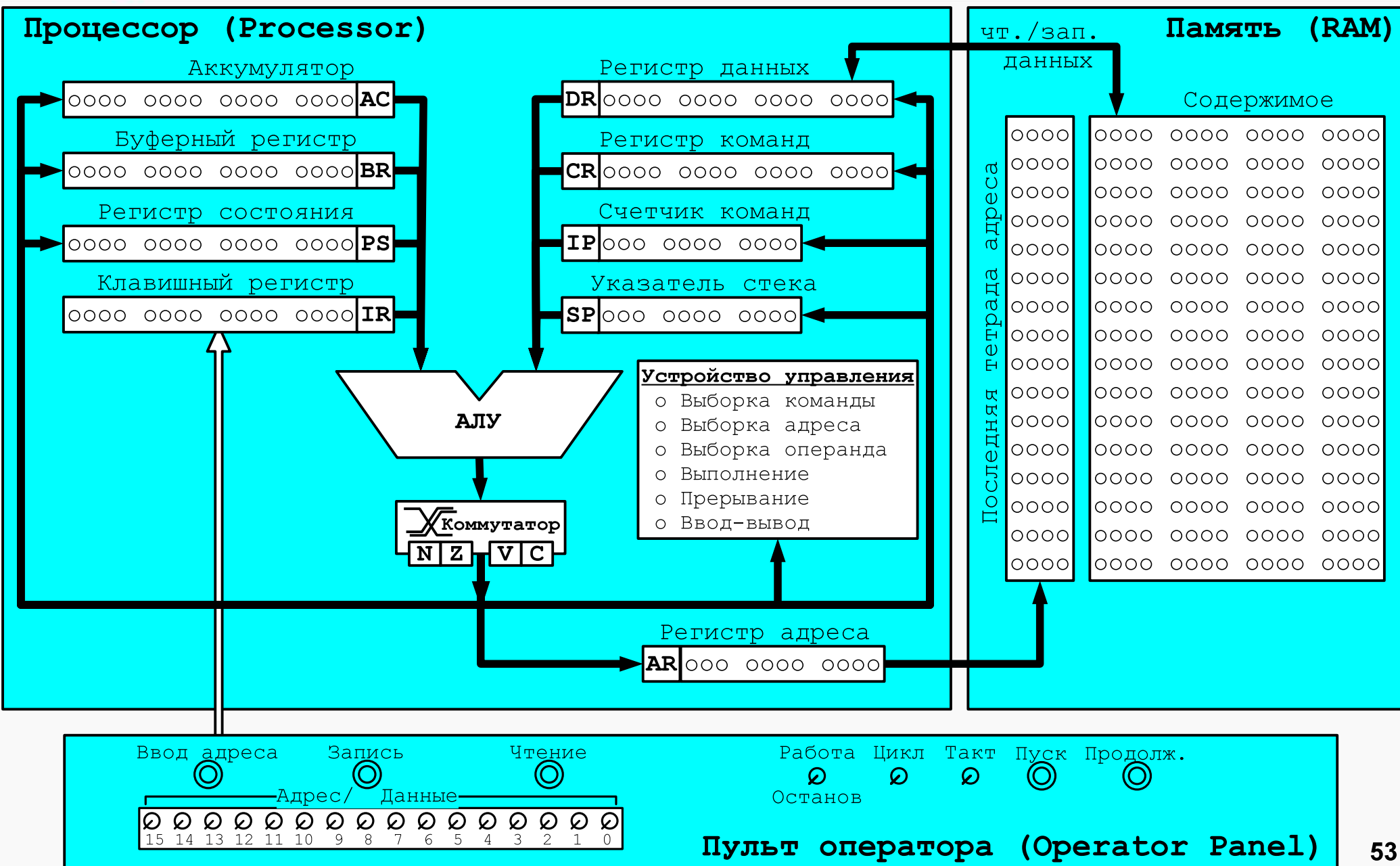
Гарвардская архитектура



Архитектура фон Неймана



БЭВМ-NG



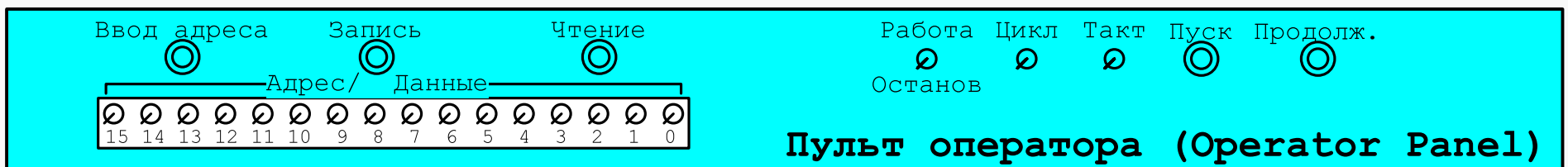
Устройство Управления

Цикл команды

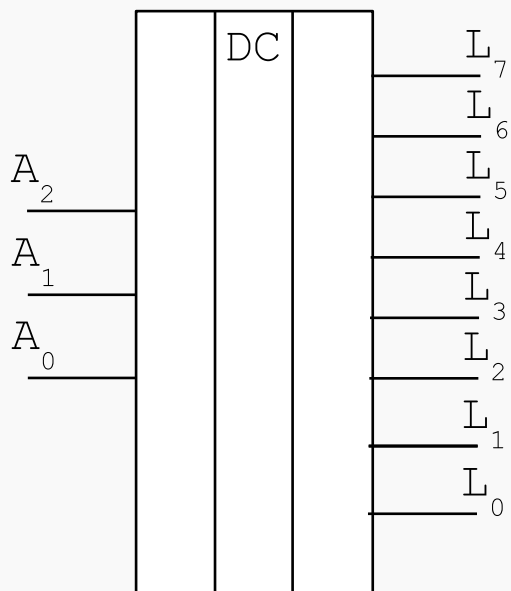
- ➔ 1. Цикл выборки команды (Instruction Fetch, IF)
- ➔ 2. Цикл выборки адреса (Address Fetch, AF)
- ➔ 3. Цикл выборки операнда (Operand Fetch, OF)
- ➔ 4. Цикл исполнения (Execution, EX)
- ➔ 5. Цикл прерывания (Interrupt, INT)

Циклы пультовых операций

- ➔ Ввод адреса (Set Instruction Pointer, SIP)
- ➔ Чтение (Read, RD)
- ➔ Запись (Write, WR)
- ➔ Пуск (Start, ST)



Отступление: Дешифратор

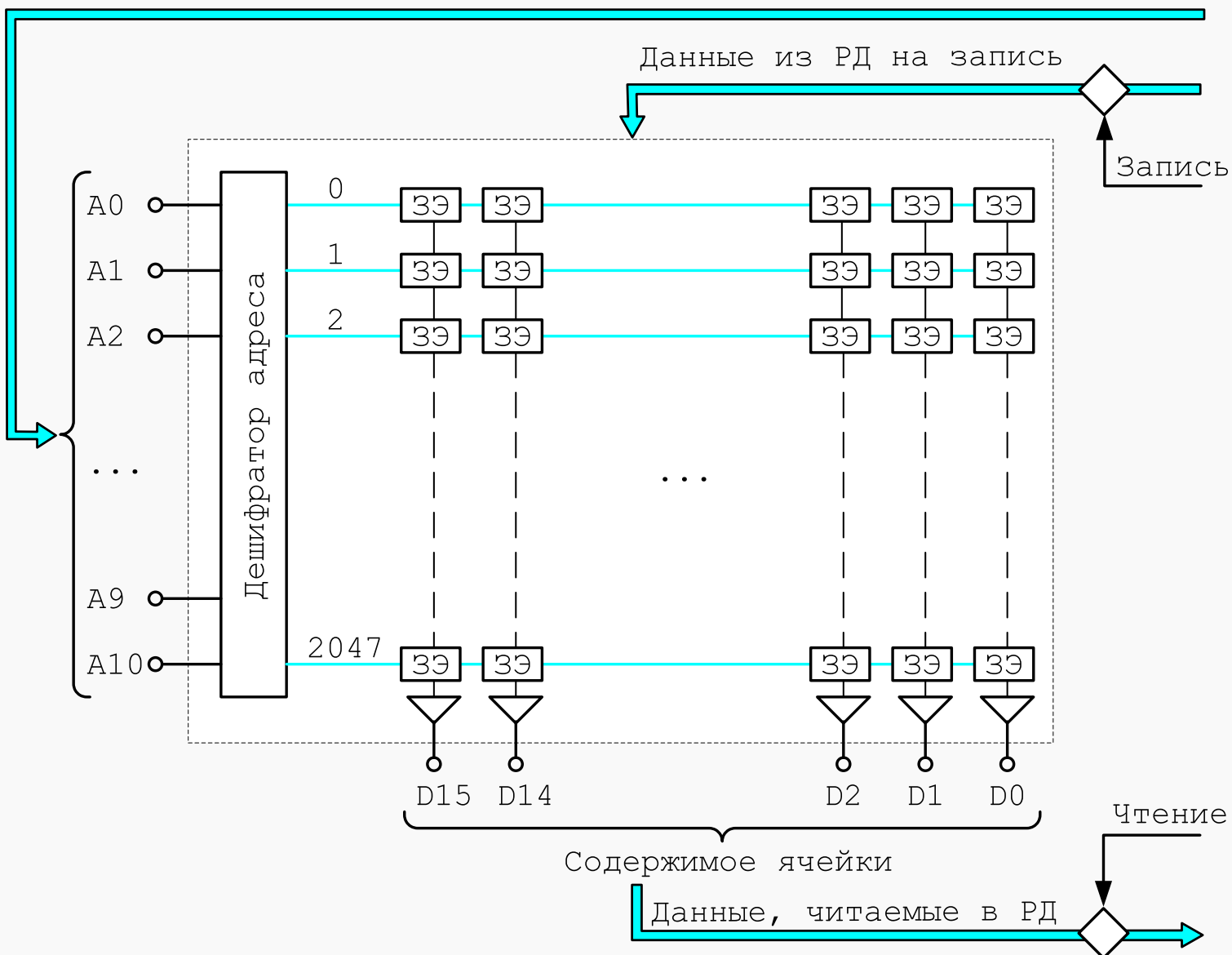


Адрес			Строка							
A_2	A_1	A_0	L_7	L_6	L_5	L_4	L_3	L_2	L_1	L_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

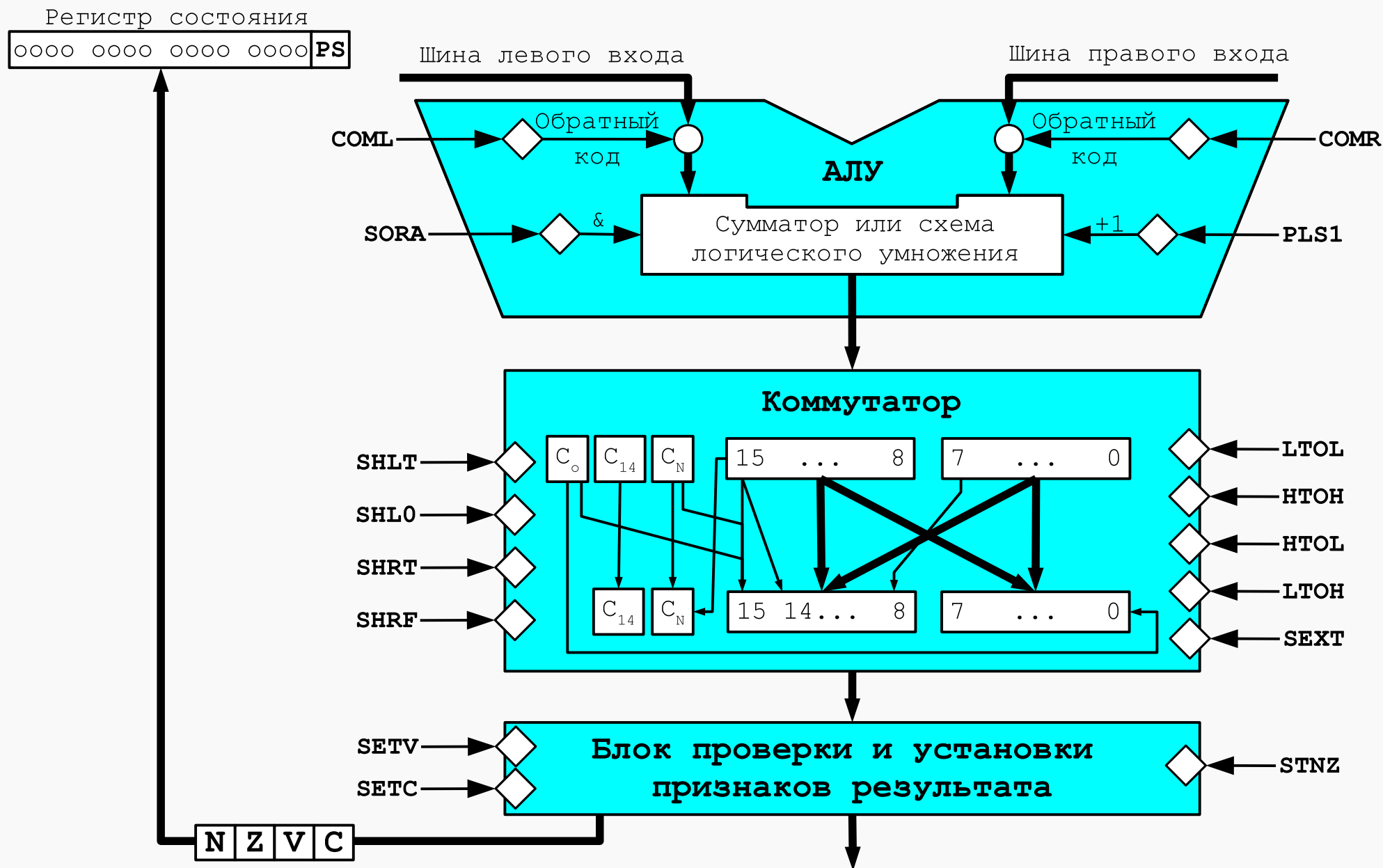
Адресуемая память БЭВМ

- 2048 16-ти разрядных ячеек

Адрес ячейки из РА



АЛУ, коммутатор, блок признаков результата



Адресная команда ...

... с прямой абсолютной адресацией

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				0	Адрес										

... с относительной адресацией

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				1	Режим			Смещение							

... с непосредственной загрузкой операнда

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП				1	1	1	1	Число							

Форматы команд

Безадресная команда

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Расширение КОП											

Команда ввода-вывода

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	Приказ				Устройство							

Команда ветвления

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	Расш. КОП				Смещение							

Адресные команды

Наименование	Мнемон.	Код	Описание
Логическое умножение	AND M	2XXX	$M \& AC \rightarrow AC$
Логическое или	OR M	3XXX	$M AC \rightarrow AC$
Сложение	ADD M	4XXX	$M + AC \rightarrow AC$
Сложение с переносом	ADC M	5XXX	$M + AC + C \rightarrow AC$
Вычитание	SUB M	6XXX	$AC - M \rightarrow AC$
Сравнение	CMP M	7XXX	Установить флаги по результату $AC - M$
Декремент и пропуск	LOOP M	8XXX	$M - 1 \rightarrow M$; Если $M \leq 0$, то $IP + 1 \rightarrow IP$
Резерв		9XXX	
Загрузка	LD M	AXXX	$M \rightarrow AC$
Обмен	SWAM M	BXXX	$M \leftrightarrow AC$
Переход	JUMP M	CXXX	$M \rightarrow IP$
Вызов подпрограммы	CALL M	DXXX	$SP - 1 \rightarrow SP$, $IP \rightarrow (SP)$, $M \rightarrow IP$
Сохранение	ST M	EXXX	$AC \rightarrow M$

Безадресные команды

Наименование	Мнемон.	Код	Описание
Нет операции	NOP	0000	Место для точек отладки, «патч» программы
Останов	HLT	0100	Отключение ТГ, переход в пультовый режим
Очистка аккумулятора	CLA	0200	$0 \rightarrow AC$
Инверсия аккумулятора	NOT	0280	$\neg AC \rightarrow AC$
Очистка рег. переноса	CLC	0300	$0 \rightarrow C$
Инверсия рег. переноса	CMC	0380	$\neg C \rightarrow C$
Циклический сдвиг влево	ROL	0400	AC и C сдвигается влево. $AC_{15} \rightarrow C, C \rightarrow AC_0$
Циклический сдвиг вправо	ROR	0480	AC и C сдвигается вправо. $AC_0 \rightarrow C, C \rightarrow AC_{15}$
Арифметический сдвиг влево	ASL	0500	AC сдвигается влево. $AC_{15} \rightarrow C, 0 \rightarrow AC_0$
Арифметический сдвиг вправо	ASR	0580	AC сдвигается вправо. $AC_0 \rightarrow C, AC_{15} \rightarrow AC_{14}$
Расширение знака байта	SXTB	0600	$AC_7 \rightarrow AC_{15} \dots AC_8$
Обмен ст. и мл. байтов	SWAB	0680	$AC_7 \dots AC_0 \leftrightarrow AC_{15} \dots AC_8$

Безадресные команды (2)

Наименование	Мнемон.	Код	Описание
Инкремент	INC	0700	AC + 1 → AC
Декремент	DEC	0740	AC - 1 → AC
Изменение знака	NEG	0780	^AC + 1 → AC
Чтение из стека	POP	0800	(SP)+ → AC
Чтение флагов из стека	POPF	0900	(SP)+ → PS
Возврат из подпрограммы	RET	0A00	(SP)+ → IP
Возврат из прерывания	IRET	0B00	(SP)+ → PS, (SP)+ → IP
Запись в стек	PUSH	0C00	AC → -(SP)
Запись флагов в стек	PUSHF	0D00	PS → -(SP)
Обмен вершины стека с аккумулятором	SWAP	0E00	AC ↔ (SP)

Команды ветвления

Наименование	Мнемон.	Код	Описание
Переход, если равенство	BEQ D	F0XX	IF Z==1 THEN IP+D+1 → IP
Переход, если неравенство	BNE D	F1XX	IF Z==0 THEN IP+D+1 → IP
Переход, если минус	BMI D	F2XX	IF N==1 THEN IP+D+1 → IP
Переход, если плюс	BPL D	F3XX	IF N==0 THEN IP+D+1 → IP
Переход, если выше или равно /перенос	BCS D BHIS D	F4XX	IF C==1 THEN IP+D+1 → IP
Переход, если ниже/нет переноса	BCC D BLO D	F5XX	IF C==0 THEN IP+D+1 → IP
Переход, если переполнение	BVS D	F6XX	IF V==1 THEN IP+D+1 → IP
Переход, если нет переполнения	BVC D	F7XX	IF V==0 THEN IP+D+1 → IP
Переход, если меньше	BLT D	F8XX	IF $N \oplus V$ ==1 THEN IP+D+1 → IP
Переход, если больше или равно	BGE D	F9XX	IF $N \oplus V$ ==0 THEN IP+D+1 → IP
Безусловный переход	BR D JUMP D	CEXX	IP+D+1 → IP

Команды ввода-вывода

Наименование	Мнемон.	Код	Описание
Запрет прерываний	DI	1000	
Разрешение прерываний	EI	1100	
Ввод	IN REG	12XX	REG → AC
Вывод	OUT REG	13XX	AC → REG
Прерывание	INT NUM	18XX	Програмное прерывание с вектором NUM
Возврат из прерывания	IRET	0B00	(SP)+ → PS, (SP)+ → IP

Как выполняются эти команды?

- Ответы на ВСЕ вопросы потактового выполнения команд:

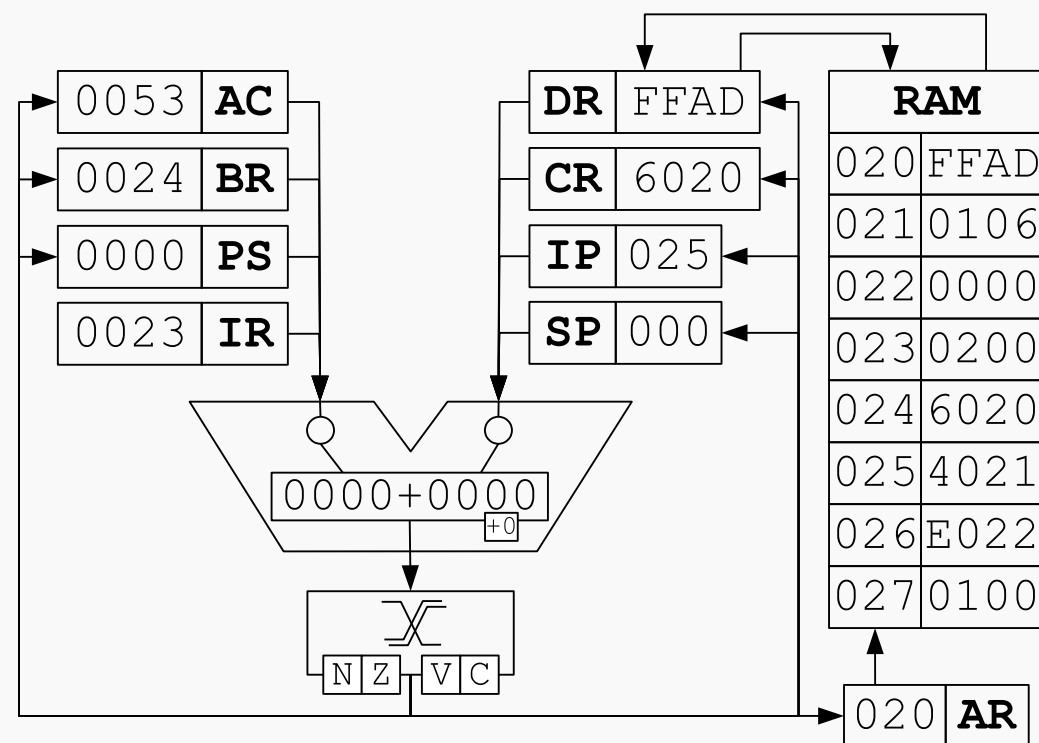
Методические указания к лабораторным работам

Приложение В, табл. В.10

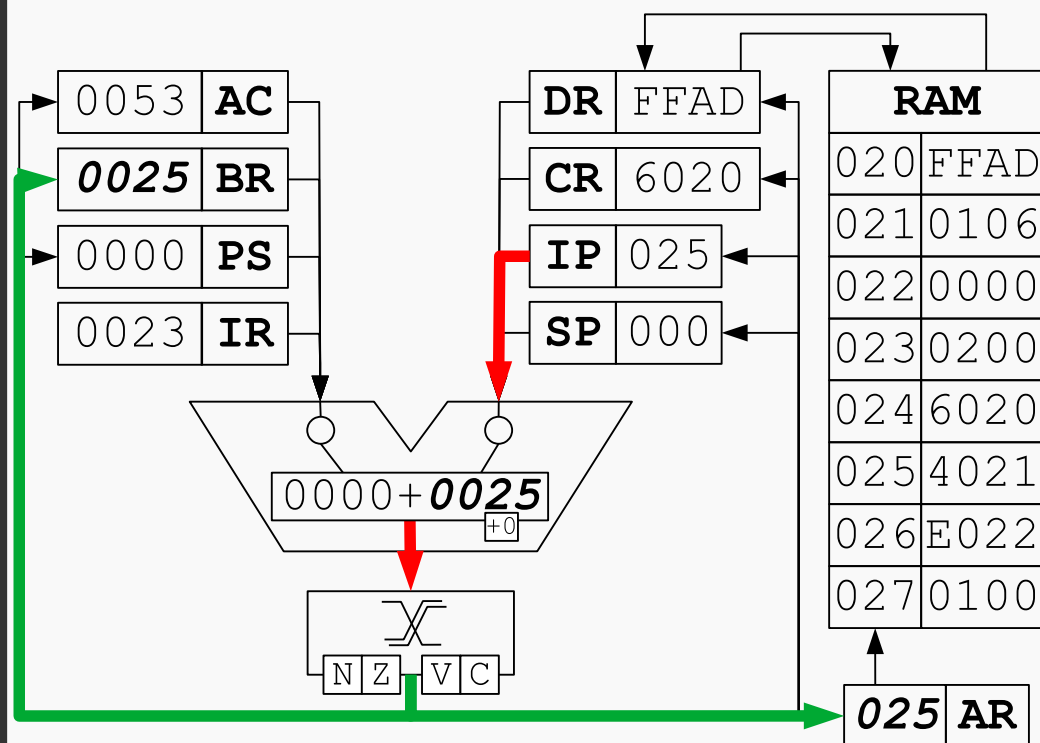
или `java -Dmode=decoder -jar bcomp-ng.jar`

- Используйте БЭВМ в режиме ТАКТ!

Цикл выборки команды: ADD 21

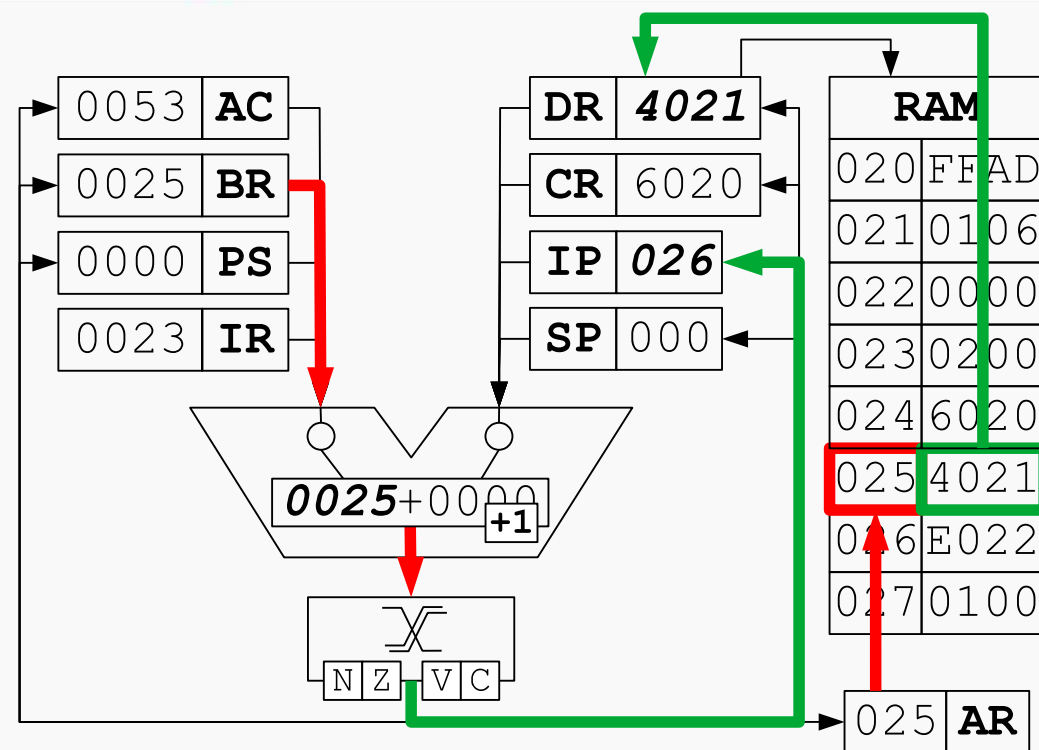


0) Исходное состояние до начала цикла выборки команды

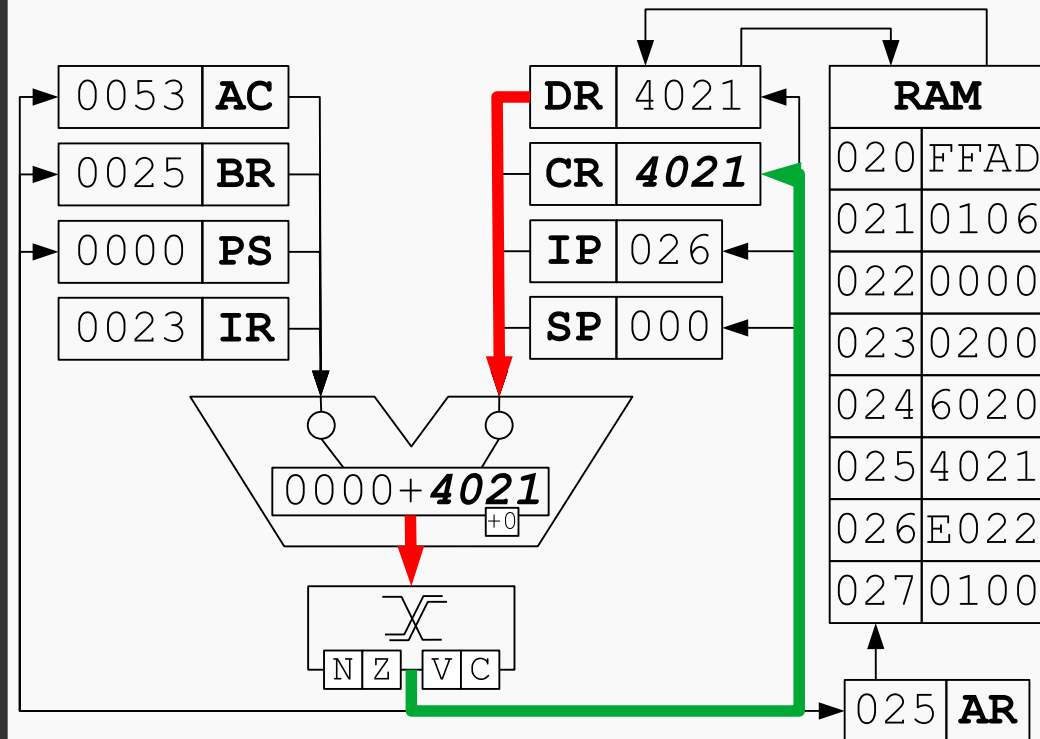


1) IP -> BR, AR
Содержимое IP через АЛУ записывается в BR и AR

Цикл выборки команды: ADD 21

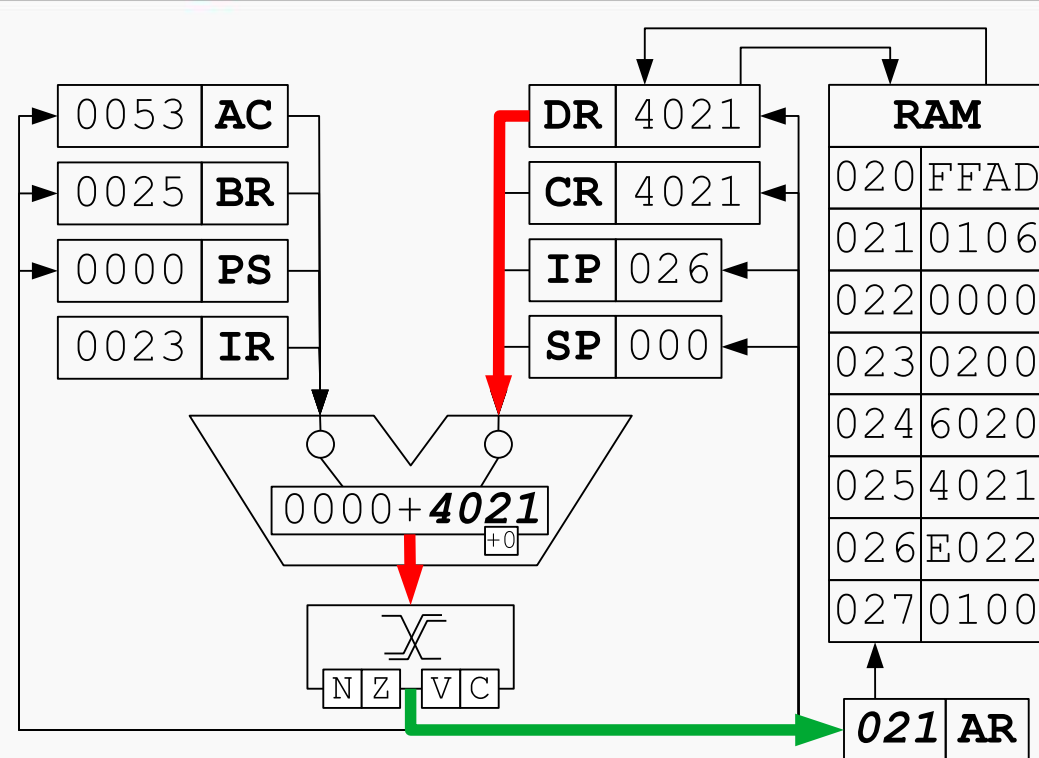


2) $BR + 1 \rightarrow IP$, $MEM(AR) \rightarrow DR$,
Содержимое BR увеличивается на 1 и записывается в IP, одновременно с этим по 25 адресу содержимое читается в DR



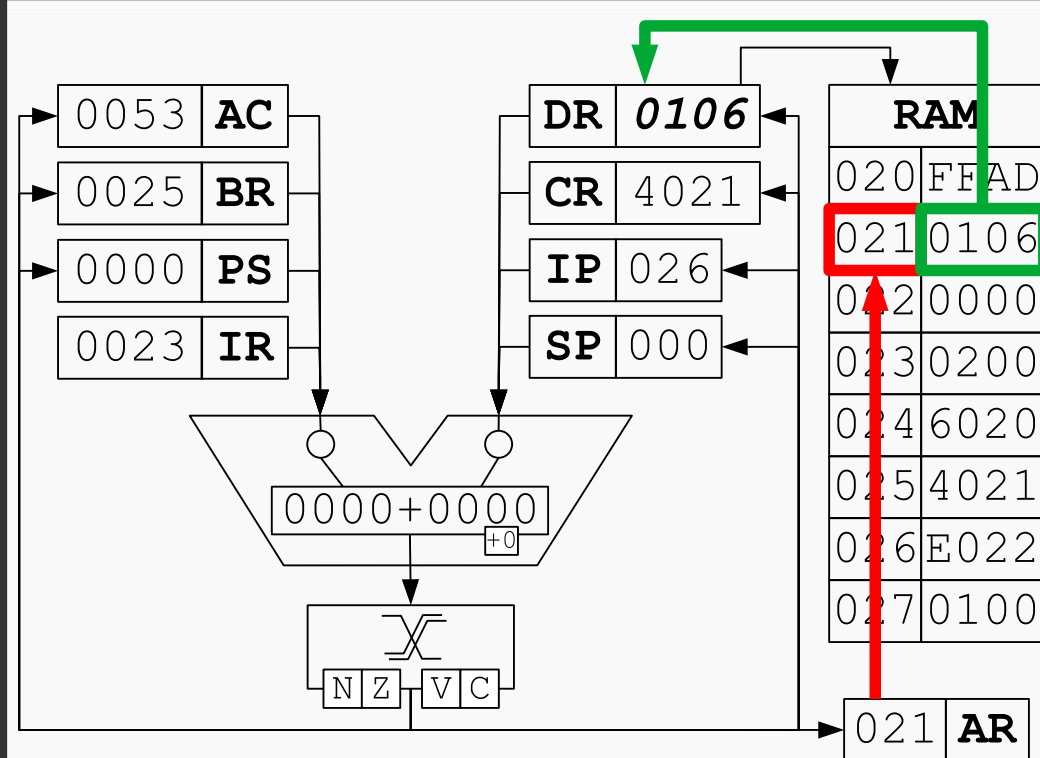
3) $DR \rightarrow CR$
Содержимое DR через АЛУ записывается в CR

Цикл выборки операнда: ADD 21



1) DR -> AR

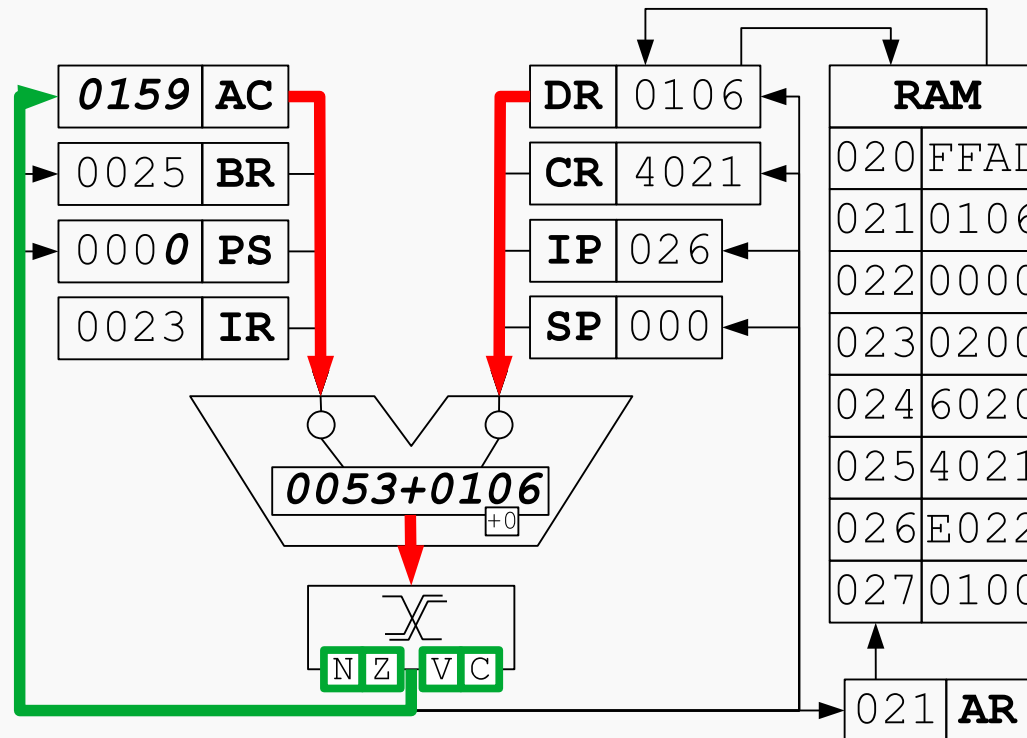
Младшие 11 разрядов
DR (адрес операнда из
команды)
пересылаются в AR



2) MEM(AR) -> DR

Загрузка из памяти по
адресу 21 значения в
DR

Цикл исполнения: ADD 21



1) $AC + DR \rightarrow AC, N, Z, V, C$

Содержимое DR на правом входе АЛУ складывается с содержимым AC на левом входе АЛУ и записывается в AC. Признаки результата N,Z,V,C обнуляются