

## Справка по микрокоду Базовой ЭВМ (v 1.45.09 / 2023):

Что такое микрокоманда?

Микрокоманда -- часть исполнения команды.

Что такое команда?

Команда -- это способ управления Базовой ЭВМ. Тривиальный пример команд: CLA (0200), ADD 0x042 (4042). Команды хранятся в памяти БЭВМ.

Микрокоманды складываются в некую цепочку (обязанностей), образуя так называемый интерпретатор (*в нашем случае это интерпретатор Базовой ЭВМ*). Когда мы отдаём любой приказ Базовой ЭВМ, первое, что она делает -- **обращается к интерпретатору**, где заданы **простейшие потактовые действия**, последовательное исполнение которых **гарантирует выполнение одной команды**. Как только исполнение одной команды будет завершено, Базовая ЭВМ попадет в начало интерпретатора, таким образом, будет готова исполнить еще одну команду.

Микрокод по своей сути состоит из битов **0/1**, которые отвечают за *открытие того или иного вентиля* в Базовой ЭВМ. Именно из таких закрытий/открытий вентилей и состоит вся логика управления Базовой ЭВМ. Исполнение одной команды может включать в себя исполнение ~20-40 микрокоманд. Весь микрокод (код микрокоманды) хранится в *памяти микрокоманд*, внутри Устройства Управления (*Control Unit*) Базовой ЭВМ.

Память микрокода состоит из **256** (адрес 8-битный/8-разрядный) **40-разрядных ячеек**. Существует **2** типа микрокоманд: **операционные** (*оперируют регистрами/памятью Базовой ЭВМ*) и **управляющие** (*управляют логикой вычисления, то есть выбирают нужную операционную микрокоманду*). За выбор этого типа отвечает **39-й бит**.

Устройство микрокода:

0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0  
0 . 0 0 0 0  
39 38 37 36 . 35 34 33 32 . 31 30 29 28 . 27 26 25 24 . 23 22 21 20 . 19 18 17 16 .  
15 14 13 12 . 11 10 9 8 . 7 6 5 4 . 3 2 1 0

Состоит из **10** тетрад (блоков по **4** двоичных бита) -> **40** бит.

Устройство управляющей микрокоманды:

1XXX.XXXC.JUMP.ADDR.CBIT.MASK.COMM.RALU.READ.REGS

Биты X - не используются.

Шестнадцатеричная система:

8CJABMCARR

- Первая тетрада (**8**) -- всегда 8\*
- Вторая тетрада (**C** - Compare) -- либо 1, либо 0 -- однобитовое поле сравнения.
- Третья и четвертая тетрады (**JA** - Jump Address): адрес перехода в память микрокоманд (напоминаю, адрес 8-битовый)
- Пятая и шестая тетрада (**BM** - Bit Mask) -- Битовая маска для проверки значения с коммутатора
- Седьмая тетрада (**C** - Commutator) -- Правила для коммутатора
- Восьмая тетрада (**A** - ALU) -- Правила для АЛУ
- Девятая и десятая тетрада (**RR** - Read Registers) -- Правила для чтения регистров

*N/B! \* - Первая тетрада обязательно должна иметь вид: 1XXX (39-й бит = 1), при этом биты с 38-36 не используются, поэтому вполне может быть и число, большее 0x8, например 0x9 (1001). То есть, 9000000000 -- тоже управляющая микрокоманда.*

Рассмотрим это на примере микрокоманды: 80C4101040 - GOTO INT @C4

- Первая тетрада свидетельствует о том, что это *управляющая микрокоманда*.
- Вторая тетрада свидетельствует о том, что *сравнение будет с нулем*.
- Третья и четвертая тетрада: *если сравнение будет удачным, мы перейдем на микрокоманду по адресу C4*.
- Пятая и шестая тетрада: *Битовая маска: 0001.0000* - мы будем смотреть *только на 4й бит* из тех, что нам придут из коммутатора.
- - Напоминаю, что из коммутатора приходят только младшие (первые) 8 разрядов (с 0 по 7 - это называется *младший байт*)
- Седьмая тетрада: Правила коммутатора: 0001
  - - -> 12-й бит = 1 (правила коммутатора это биты с 12 по 15) -> LTOL (см. Шпаргалку вентильных схем в третьей части презентации по ОПД)
  - - LTOL: Lower To Lower - младший байт неизменно пройдет через коммутатор, останется младшим байтом
  - - // вопрос для опытных: *что будет со старшим байтом?*
- Восьмая тетрада: Правила АЛУ: 0000 -> классическая сумма (бит SORA 11-й = 0, остальные биты говорят о том, что ничего не инвертируется, и +1 сигнал тоже не подается)
- Девятая и десятая тетрада: 0100.0000 -> байт = 1 -> *читаем регистр PS*.

Итого, что мы имеем:

Содержимое регистра PS направляется в АЛУ с правилом сложения с нулем (потому что на другом входе АЛУ ничего нет), младший байт из АЛУ пройдет через коммутатор, и направиться в Устройство Управления. В Устройстве Управления в данный момент проверяется соответствие 4-го бита на 0, и если это условие соблюдается, мы перейдем по адресу C4.

*N/B! Так как PS(4) всегда равен 0 (см. Лекцию 4 части 3), то это де-факто означает безусловный переход. - GOTO INT @C4*

Рассмотрим еще пример:

800C404002 - IF CR(14)=0 GOTO CHKABS @0C

- Первая тетрада - все как обычно
- Вторая тетрада - сравним с нулем
- Третья и четвертая - Переход на 0C, если сравнение будет удачным
- Пятая и шестая - Битовая маска: 0100.0000 -> проверим 6-й бит, который придет с коммутатора (почему не 14й???)
- Седьмая - Правила коммутатора: 0100 -> HTOL = 1 (14-й бит микрокоманды) -> передадим старший байт в младший.
- Восьмая - Правила АЛУ: 0000 -> сложение

- Девятая и десятая: 0000.0010 -> RDCR (2-й бит), читаем CR.

Итого, что мы имеем:

Содержимое регистра CR направляется в АЛУ с правилом сложения с нулем, старший байт из АЛУ пройдя через коммутатор, перепишется в младший байт, младший байт традиционно направляется в Устройство Управления. Там на данный момент проверяется, чтобы 6-й бит пришедшего из коммутатора значения было равен 0, в этом случае будет сделан переход на микрокоманду 0C.

Рассмотрим последний пример: 815C081040 - IF PS(N) = 1 GOTO BR @5C

В отличие от примеров выше, тут уже проверка на 1. Переход будет произведен в адрес 5C, битовая маска говорит о том, что проверяется 3-й бит младшего байта, на коммутаторе стоит правило 0001 (LTOL), на АЛУ стоит правило 0000 (сумма), а смотреть мы будем содержимое регистра PS.

Так как 3-й бит PS это флаг N -> метка для нашей микрокоманды будет выглядеть так: IF PS(N) = 1 GOTO BR @5C

*N/B! Никакая управляющая микрокоманда не меняет значения ни одного регистра, поэтому вы можете смело в таблице трассировки, видя управляющую микрокоманду переписывать значения регистров IP, CR, AR, DR, SP, BR, AC, NZVC те, что были в прошлой строчке.*

С управляющими микрокомандами все довольно просто и понятно, перейдем к операционным микрокомандам.

Устройство операционной микрокоманды:

0НXX.ΙΟΜΜ.WRTE.REGS.COMM.COMM.COMM.RALU.READ.REGS

Биты X - не используются.

Шестнадцатеричный код: 0IWRCRR

- Первая тетрада (0): 4 для команды Halt (остановка Базовой ЭВМ), **во всех остальных случаях - 0.**
- Вторая тетрада (I - IO/MEM): Приказы для IO + Memory
- Третья и четвертая тетрада (WR - Write Registers): Приказы для записи значений в регистры.
- Пятая, шестая тетрады (CC - Commutator): Расширенные правила для коммутатора
- Седьмая тетрада (C - Commutator): Тоже правила для коммутатора (но уже Вам знакомые)
- Восьмая тетрада (A - ALU): Правила для АЛУ (Здесь все тоже самое, что и было раньше)
- Девятая и десятая тетрада (RR - Read Registers): Правила чтения регистров (Вы это уже знаете)

Подробнее про каждый бит в операционной микрокоманде:

Блок чтения регистров (9 и 10 тетрады):

- 00 - RDDR
- 01 - RDCR
- 02 - RDIP
- 03 - RDSP
- 04 - RDAC
- 05 - RDBR

- 06 - RDPC
- 07 - RDIR

*N/B! Сначала перечислены правые регистры сверху вниз, потом левые регистры сверху вниз. (А сейчас откроите БЭВМ...)*

Блок правил АЛУ (8 тетрада):

- 08 - COMR
- 09 - COML
- 10 - PLS1
- 11 - SORA

Блок правил Коммутатора (5, 6 и 7 тетрады):

- Общий блок (7 тетрада / и те, и другие это используют):
  - 12 - LTOL
  - 13 - LTOH
  - 14 - HTOL
  - 15 - HTOH
- Операционные микрокоманды (5 и 6 тетрады):
  - 16 - SEXT
  - 17 - SHLT
  - 18 - SHL0
  - 19 - SHRT
  - 20 - SHRF
  - 21 - SETC
  - 22 - SETV
  - 23 - STNZ

Блок записи в регистры (3 и 4 тетрады):

- 24 - WRDR
- 25 - WRCR
- 26 - WRIP
- 27 - WRSP
- 28 - WRAC
- 29 - WRBR
- 30 - WRPS
- 31 - WRAR

*N/B! Записывать в IR нельзя, вместо него записывать предлагается в AR. То же самое, что и читать из AR нельзя, вместо него предлагается читать из IR.*

Блок IO + MEM (2 тетрада):

- 32 - LOAD
- 33 - STOR

- 34 - IO
- 35 - INTS

*N/B! Отдельно 38 - HALT Микрокоманда остановки БЭВМ: 4000000000*

*N/B! Чтобы проще все это запомнить, запоминайте сначала управляющие микрокоманды, потом посмотрите на сходство управляющих микрокоманд с операционными, дальше запомните порядок расположения блоков в операционных микрокомандах, а уже потом взгляните на другие N/B, которые подчеркивают закономерности для запоминания уже конкретных блоков микрокода.*

Рассмотрим на примере: 00A0009004 - IP -> AR, BR

- Первая тетрада сигнализирует о том, что это операционная команда
- Вторая тетрада сигнализирует о том, что никаких операций с памятью и вводом/выводом не происходит
- Третья и четвертая тетрады: 1010.0000; Запись будет произведена в AR (31-й бит) и в BR (29-й бит).
- Пятая и шестая тетрада говорит о том, что флаги простираются не будут, сдвига не будет, расширение знака тоже не произойдет.
- Седьмая тетрада: 1001: НТОН & LTOL, оба байта передаются в прямом порядке, ничего не меняется друг с другом.
- Восьмая тетрада: Правила АЛУ: 0000 -> классическая сумма
- Девятая и десятая тетрада: 0000.0100 -> считываем IP.

Итого мы имеем:

Содержимое регистра IP пройдя через АЛУ сложится с 0, пройдя через коммутатор сохранит расположение старших и младших байтов и запишется в регистры AR и BR. Признаки результата не устанавливаются.

*N/B! Вспомните таблицу системы команд Базовой ЭВМ.*

*В ней \* \* 0 -устанавливаются для всех команд, результат которых записывается в аккумулятор и не выходит за пределы 16 разрядов*

*\* \* \* \* - для команд, которые могут получить результат в AC, выходящий за пределы 16-разрядной сетки,*

*а - - - - указано для всех команд, которые вообще не трогают аккумулятор.*

*Здесь все работает по точно такой же логике. Напоминаю, за установку признаков результата отвечают старшие 3 бита пятой тетрады. // Вопрос для опытных: почему N и Z ставятся неразрывно друг от друга?*

**Немного разобравшись в этом всем деле, поговорим про рубежную работу.**

Почему важно понимать то, что я расписал выше?

Скорее всего знание, описанное выше, вам не пригодиться (написано студентами Афанасьева для студентов Афанасьева). Но если без шуток, вам могут выдать на рубежке таблицу интерпретатора Базовой ЭВМ без меток, и там уже не так очевидно, что 0001009201 это ~0 + DR -> DR. Это придется считать самому, а когда вы не дружите с микрокодом, разобрать это будет довольно непросто

(Ex: 4-я тетрада тут говорит о записи в DR, 7-я тетрада о сохранении порядка старшего и младшего байтов, 8-я тетрада 0010 -> инвертировать левый вход ALU, а 9 и 10-я тетрады говорят о том, что чтение будет из DR)

Что меня ждет на рубежке?

Предлагаю вам прочитать официальный текст задания на рубежную работу:

Запишите последовательность микрокоманд для выполнения заданной команды до машинного цикла прерывания. Заполните таблицу значениями регистров после выполнения каждой микрокоманды. Если происходит обращение к памяти, укажите адрес памяти, значение и операцию. Содержимое памяти по адресу 0x570:

24A3, 319B, 9B0D, F7B7, 20BD, 2542, 057B, 8985, D81A, 612B, 307E, BA30, 2FE0, 28F8, 6957, 194B. Исходная команда располагается по адресу 0x57D.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after
			57D	DBA6	57D	9AD5	057D	D605	0000	

Что здесь вообще написано? Вообще задание гласит о том, что надо сделать один пробег интерпретатора БЭВМ с исходными данными, которые изначально даны в таблице + нам дан кусочек памяти, наличие которого необходимо и достаточно для выполнения рубежной работы.

Вот давайте с этого куска памяти и начнем.

Адрес	Значение
570	24A3
571	319B
572	9B0D
573	F7B7
574	20BD
575	2542
576	057B
577	8985
578	D81A
579	612B
57A	307E
57B	BA30
57C	2FE0
57D	28F8
57E	6957
57F	194B

Нам это еще не раз пригодиться, поверьте...

Вернемся к таблице. Каждую новую выполняемую микрокоманду вместе со значениями регистров после ее исполнения надо писать на новой строчке таблицы трассировки. Собственно говоря это и

называется трассировка; вы пишете микрокоманду и значения регистров Базовой ЭВМ после выполнения каждой микрокоманды.

*N/B! Как вы уже могли догадаться, мы трассируем только одну команду! Но много микрокоманд.*

Ну давайте же вместе с вами и сделаем эту рубежку:

Для честности эксперимента, мы будем пользоваться [только интерпретатором без меток](#).

Начинается все с первой микрокоманды:

00A0009004

План:

- 1-я тетрада = 0? Да -> Следующий адрес микрокоманды (MP after) всегда будет на 1 больше текущего. Запишем сразу это, пока не забыли:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01										02	00A0009004

Далее, оценим, что это за микрокоманда и какие регистры будут изменены:

Микрокоманда операционная (0000)

Операций ввода/вывода и с памятью отсутствуют (0000)

Запись будет произведена в регистры AR, BR (1010.0000)

LTOL НТОН (0000.0000.1001)

Правила АЛУ: Классическая сумма (0000)

Чтение из IP (0000.0100).

Итого: IP -> ALU (+0) -> COMM: LTOL; НТОН -> AR, BR.

Так и запишем:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004

Поздравляю, мы заполнили первую строчку, теперь мы можем заполнять вторую строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02											0104009420

Бойцы рубежки продолжают сражаться. Дальше мы смотрим на следующую микрокоманду (*Откуда мы ее берем? Из интерпретатора!*) 0104009420

Внимательно посмотрим на нее:

Операционная микрокоманда (0000) -> следующая микрокоманда будет на 1 больше предыдущей (03)

Обратите внимание, здесь произойдет операция чтения из памяти (2-я тетрада: 0001)

*N/B! Запись в память происходит строго в Регистр Данных (DR)! Адрес берется из AR, из прошлой строчки таблицы мы можем сделать вывод о том, что обращение произойдет по адресу AR = 57D. Содержимое памяти нам дано в условии задания!*

Запись произойдет в регистр IP (0000.0100)

LTOL, НТОН (0000.0000.1001)

Произойдет операция PLS1 на АЛУ (0100)

Чтение произойдет из BR (0010.0000)

Итого: MEM(AR) -> DR; BR -> ALU (+1) -> COMM: LTOL, НТОН -> IP

Запишем строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	<b>57D</b>	<b>28F8(L)</b>	<b>57E</b>	DBA6	57D	<b>28F8</b>	<b>057D</b>	D605	0000	03	0104009420
<b>03</b>											<b>0002009001</b>

Продолжаем. Следующая микрокоманда (тоже операционная): 0002009001

Вторая тетрада гласит о том, что операции ввода/вывода и использования памяти отсутствуют.

Запись будет осуществляться в CR (0000.0010)

LTOL, НТОН (0000.0000.1001)

Классическая сумма (0000)

Чтение из DR (0000.0001)

Итого: DR -> ALU +0 -> COMM: LTOL, НТОН -> CR

Запишем строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	<b>28F8</b>	57D	<b>28F8</b>	057D	D605	0000	04	0002009001
<b>04</b>											<b>8109804002</b>

Ура! Первая управляющая микрокоманда!

Рассмотрим этого покемона попороднее (кто еще помнит, что там было в начале этого документа?)

8109804002

Управляющая микрокоманда (1-я тетрада = 8)

*N/B! Управляющая микрокоманда = переписываем все значения регистров IP, CR, AR, DR, BR, AC, NZVC; они меняться не будут:*

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000		8109804002

Проверяется бит на 1

В случае успешного сравнения прыжок будет на адрес 09

Битовая маска: 1000.0000. Смотрим 7-й бит, пришедший с коммутатора

Правило коммутатора: 0100: (HTOL) (проверка 7-го бита превращается в проверку 15-го бита)

Правило АЛУ: сумма (0000)

Чтение происходит из CR (0000.0010)

Итого: IF CR(15) = 1 then GOTO 09

Н/В! Посмотрим на содержимое CR: 28F8

Двоичное представление: 0010.1000.1111.1000

Как можем увидеть, 15-й разряд CR = 0, что удовлетворяет нашему условию.

Условие не выполнено. Вместо перехода на адрес 09 мы перейдем на следующую микрокоманду.

Н/В! Не обязательно додумывать, что же там будет дальше. Просто следуйте алгоритму.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
<b>05</b>											<b>810C404002</b>

Управляющих команд будет больше, чем операционных, поэтому привыкаем к ним (не зря мы их разбирали первыми)

810C404002

Проверяем на 1

Переходим на адрес 0C

Битовая маска: 0100.0000 -> проверяем 6-й бит, пришедший с коммутатора.

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)

Читаем с CR (0000.0010)

Итого: IF CR(14) = 1 then GOTO 0C

14-й бит CR не равен 1, поэтому мы просто идем дальше, записав в MP after значение 06.

Остальные регистры просто переписываем:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
<b>06</b>											<b>810C204002</b>

Продолжаем серию беспощадных проверок, которые нам даровали управляющие микрокоманды.  
**810C204002**

Проверяем на 1

Переходим на адрес 0C

Битовая маска: 0010.0000 -> проверяем 5-й бит, пришедший с коммутатора.

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)

Читаем с CR (0000.0010)

Итого: IF CR(13) = 1 then GOTO 0C

13-й бит CR действительно удовлетворяет нашему условию. Переход на 0C будет выполнен.

Обратите внимание на сходство с предыдущей микрокомандой...

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
<b>0C</b>											<b>8024084002</b>

Новый день, новый микрокод.

**8024084002**

Управляющая микрокоманда (вопрос: что мы делаем с регистрами IP, CR, AR, DR, BR, AC, NZVC?)

Проверяем на 0

В случае успешной проверки переходим на адрес 24

Битовая маска: 0000.1000 -> проверяется 3-й бит

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)  
Читаем регистр CR (0000.0010)

Итого: IF CR(11) = 0 then GOTO 24

*N/B! Если вы так и не поняли, почему именно 10-й бит, (это нормально, мы тоже ничего не поняли,) то вот вам простая формула, что делать, если правило коммутатора оказалось HTOL: Проверяется (3 + 8)-й бит: прибавляем 8.*

Почему? Давайте посмотрим сначала на битовую маску:  
0000.1000 (проверим 3-й бит)

Потом посмотрим на то, что сделал коммутатор:

HTOL - операция переноса старшего байта (разрядов с 8 по 15-й) в младший (с 0 по 7-й)

Рассмотрим старший байт:

x x x x . x x x x  
15 14 13 12 . 11 10 9 8

Наложив на все это битовую маску, получаем, что де-факто проверяется 11-й бит, а выражается все это по формуле (3 + 8)-й бит.

Ну и потому что 8 это длина байта, а мы как раз скинули младший байт.

10-й бит CR не равен нулю, проверка не пройдена. Это означает, что счетчик микрокоманд просто увеличится на 1.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
<b>0D</b>											<b>0020011002</b>

Так-с, мы наткнулись на операционную микрокоманду... Грядут большие изменения в регистрах, которые уже как 4 микрокоманды никак не менялись.

Вспоминаем, что читали буквально недавно (например то, что следующая микрокоманда будет  $0D + 1 = 0E$ ).

**0020011002**

Операций ввода-вывода и доступа к памяти отсутствует (2-я тетрада = 0)

Значение будет записано в BR (0010.0000)

А вот с правилами коммутатора тут интереснее, чем 0000.0000.1001:

Они вот такие: 0000.0001.0001 (0x011 - тетрады 5-я, 6-я и 7-я)

С 7-й тетрадой все понятно: LTOL

А вот в 6-й тетраде младший бит отвечает за операцию расширения знака (SEXT)

Правила АЛУ: сумма (0000)  
Чтение происходит из CR (0000.0010)

Итого: CR > ALU (+0) -> COMM: SEXT(7 -> 8...15) -> BR

M P	Адре с	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	<b>28F8</b>	57D	28F8	<b>FFF8</b>	D605	0000	0E	0020011002
<b>0E</b>											<b>811C044002</b>

Ну, недолго операционные команды радовали нас, поехали проверять дальше (кстати, без меток тут довольно легко ошибиться, будьте внимательны)

**811C044002**

Проверяем на 1

Переход на 1С

Битовая маска: 0000.0100 -> проверяем 2-й бит.

Правила коммутатора: 0100: HTOL (-> 2 + 8 -> 10-й бит)

Правила АЛУ: 0000 / сумма

Читаем содержимое CR (0000.0010)

Итого: IF CR(10) = 1 then GOTO 1С

Это неправда, напоминаю содержимое CR:

28F8 = 0010.1000.1111.1000

Значит, мы просто пойдем дальше по микрокоду (но вы верьте, что когда-нибудь это закончится):

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
<b>0F</b>											<b>0080009024</b>

Операционная микрокоманда подошла, откуда ее не ждали, а значит далее мы перейдем на микрокоманду **10**.

**0080009024**

Посчитаем значения регистров:

Операции чтения/записи/ввода-вывода не происходит.

Запись произойдет в AR (**1000.0000**).

Правила коммутатора просты: LTOL, НТОН (**0000.0000.1001**)

Правила АЛУ: **0000** / сумма

Чтение произойдет из регистров BR и IP (**0010.0100**).

N/B! Чтение может проходить из двух регистров. Просто до этого АЛУ суммировала содержимое регистра с нулем, а сейчас на левый вход подается содержимое BR, на правый -- IP.

Итого: BR, IP -> ALU (BR + IP) -> COMM: НТОН, LTOL -> AR.

Посчитаем BR + IP:

**FFF8 + 57E = 0x10576**.

В AR дойдут только **0x576**.

M P	Адре с	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			<b>57E</b>	28F8	<b>576</b>	28F8	<b>FFF8</b>	D605	0000	<i>10</i>	0080009024
<b>10</b>											<b>0100000000</b>

Вот бы все микрокоманды были такие, как та, что вы увидите сейчас...  
**0100000000**

Тут все просто. Мы произведем операцию чтения из памяти (2-я тетрада: **0001**)

И все. Вопрос на понимание: чему будет равна следующая микрокоманда? Не бойтесь ответить даже самый глупый, на ваш взгляд, ответ.

Кто помнит, что было в ячейке по адресу 576??

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11											8114024002

Вернемся к управляющим микрокомандам.

8114024002

Проверка на 1

Переход в случае успешной проверки на 14 (не очень далеко кнш)

Битовая маска: проверяем 1-й бит, пришедший с коммутатора. (0000.0010)

Правило коммутатора: 0100 / HTOL

Правило АЛУ: 0000 / сумма

Чтение CR (0000.0010)

Итого: IF CR(9) = 1 then GOTO 14 (!!! bug fix by @O\_skull32)

Но у нас это не так, поэтому переход будет на 1 микрокоманду вперед.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
<b>12</b>											<b>81E0014002</b>

Вечер управляющих микрокоманд в самом разгаре!  
**81E0014002**

Проверка на 1

Переход на E0 в случае успешной проверки.

Битовая маска: проверяем 0-й бит, пришедший с коммутатора (0000.0001)

Правило коммутатора уже до боли знакомое нам: HTOL (0100)

Правило АЛУ тоже: 0000 (кто помнит, что это такое?)

Читаем содержимое все еще CR (0000.0010)

Итого: IF CR(8) = 1 then GOTO E0

У нас это не так, поэтому двигаемся дальше на 13-ю микрокоманду.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
<b>13</b>											<b>8024101040</b>

А кто-то говорил, что это легко?  
**8024101040**

Управляющая микрокоманда, если кто забыл (все еще держу в курсе, что содержимое регистров IP, CR, AR, DR, BR, AC, NZVC не изменяется)

Проверка на 0

Переход на адрес 24, если проверка удачная

Битовая маска: 0001.0000 -> проверяем 5-й бит

Правило коммутатора: LTOL (0001)

Правило АЛУ: сумма (0000)

Читаем содержимое PS (0100.0000)

*N/B! Я напоминаю, что 5-й бит PS всегда равен 0. То самое, о чем говорил Клименков на лекции.*

Ну итого нас вынуждают прыгнуть на 24-й адрес получается...

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24											8026804002

Быстро расправимся с этой управляющей микрокомандой!

8026804002

Проверяем на 0

Прыжок на 26 в случае удачной проверки

Проверяем 7-й бит (1000.0000)

Правило коммутатора: HTOL (0100)

Правило АЛУ: 0000 / сумма

Читаем CR (0000.0010).

Итого: IF CR(15)=0 then GOTO 26

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
<b>26</b>											<b>0080009001</b>

Настало время операционной микрокоманды:  
**0080009001**

Чтение/запись/ввод-вывод не производится.

Запись производится в AR (1000.0000)

Правило коммутатора: 0000.0000.1001: LTOL, НТОН

Правило АЛУ: 0000 / сумма

Чтение происходит из DR (0000.0001).

Итого: DR -> ALU (+0) -> COMM: LTOL, НТОН -> AR

Просто перенесли содержимое из одного регистра в другой. Классика!

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	<b>57B</b>	<b>057B</b>	FFF8	D605	0000	27	0080009001
<b>27</b>											<b>0100000000</b>

Кажется, мы уже видели эту микрокоманду! Поэтому, чтобы сэкономить ваше и мое время, воспользуемся стараниями из 10-й микрокоманды

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	<b>57B</b>	<b>BA30(L).</b>	57E	28F8	<b>57B</b>	<b>BA30</b>	FFF8	D605	0000	28	0100000000
<b>28</b>											<b>813C804002</b>

Поздравляю, все выборки закончились! В DR хранится тот самый заветный операнд, который сделает логическое И (КОП: 0x2) с содержимым аккумулятора... Ждем этот момент с нетерпением. А пока -- еще одна управляющая микрокоманда.  
813C804002

Проверка на 1

Переход на ЗС (я уже буду опускать, что в случае успешной проверки)

Битовая маска: проверка 7-го бита. (1000.0000)

Правило коммутатора: HTOL (0100)

Правило АЛУ: 0000 / сумма

Читаем CR (0000.0010).

Итого: IF CR(15) = 1 then GOTO 3C

Условие не верно. Идем дальше.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29											8130404002

Устали? Я готов подождать.

8130404002

Проверка на 1

Переход на 30

Проверка 6-го бита (0100.0000)

HTOL (0100)

Сумма (0000)

Чтение CR (0000.0010).

Итого: IF CR(14) = 1 then GOTO 30

У нас это не так, поэтому продолжаем.

*N/B! The biggest mistake in your life! :c: owl from hogwarts: 0x29 + 0x1 = 0x30*

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
<b>2A</b>											<b>812D104002</b>

Спидран по управляющей микрокоманде, поехали  
**812D104002**

Проверка на 1  
Переход на 2D  
Проверка 4-го бита (0001.0000)  
HTOL (0100)  
Сумма (0000)  
Чтение CR (0000.0010)

Итого: IF CR(12) = 1 then GOTO 2D

Неправда, поэтому мы туда не пойдем.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
<b>2B</b>											<b>0010C09811</b>

Финишная прямая, господа. Это последняя операционная команда (спойлер, упс)  
**0010C09811**

Операций чтения/записи/ввода-вывода отсутствуют.  
Запись произведется в AC (**0001.0000**)

*N/B! Будьте начеку и помните, что как только вы видите запись в AC, где-то рядом надо быть готовым увидеть установку флагов NZVC...*

Правила коммутатора: **0xC09 (1100.0000.1001)**. С **0x9** все просто: LTOL, HTOL.  
А вот биты тетрады **0xC (1100)** отвечают за установку флагов N, Z и V (STNZ - 23-й разряд, SETV - 22-й).

Правила АЛУ: **1000. SORA = 1** -> Операция логического умножения (И)  
Читаем регистры AC и DR. (**0001.0001**)

Итого: AC, DR -> ALU (AC & DR) -> COMM: LTOL, HTOH + N, Z, V -> AC

Здесь главное очень аккуратно переписать все в таблицу.

**D605 & BA30 = 9200 (N = 1; Z = 0; V = 0)**

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
----	-------	------------------------	----	----	----	----	----	----	------	-------------	-----------

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
2B			57E	28F8	57B	BA30	FFF8	<b>9200</b>	<b>1000</b>	2C	0010C09811
<b>2C</b>											<b>80C4101040</b>

Вроде бы как мы все сделали. Наверное, это должна быть последняя управляющая микрокоманда.  
**80C4101040**

Проверка на 0

Переход на C4

Проверка 5-го бита (0001.0000)

Правило коммутатора: LTOL (0001)

Правило АЛУ: 0000 / сумма

Чтение PS (0100.0000)

Кажется, 5-й бит PS всегда 0, поэтому мы "безусловно" переходим на C4.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
2B			57E	28F8	57B	BA30	FFF8	9200	1000	2C	0010C09811
2C			57E	28F8	57B	BA30	FFF8	9200	1000	C4	80C4101040

Как только мы увидели, что произошел переход на C4, мы **останавливаемся**. Потому что *C4 -- это начало цикла прерывания*, а нас просили написать ровно до момента этого цикла.

Выдыхаем, открываем шампанское, рубежка написана, а перед вами полная трассировка микрокоманд, которые в сумме составляют исполнение команды 28F8. Сверить ответ вы можете в методичке, вариант взят оттуда. А также используя [режим запуска БЭВМ cli/dual](#) + консольной команды `c1`, которая включит потактовую трассировку (см. [Трассировка в БЭВМ](#)).

! Чтобы закрепить полученные знания решений рубежки, откройте, например, вот эти [табличные варианты](#) (и кстати, там есть интерпретатор с метками! Первые пару вариантов рекомендую прорешать именно пользуясь меточным интерпретатором, а уже потом приступать к более сложным вариантам, если конечно понимаете и опасаетесь, что вам дадут трассировку микрокоманд с [интерпретатором без меток](#))

*Вопрос опытным людям: Я не использовал интерпретатор с метками во время решения этого варианта. А вы?*

*Древние русы собирались и решали на механических ЭВМ эти рубежки без калькуляторов, и с первого раза. Так решите и вы!*

Благодарю за прочтение.

---

Если вы нашли техническую проблему в данном тексте, писать сюда: [@ze](#)

## Справка по микрокоду Базовой ЭВМ (v 1.45.09 / 2023):

Что такое микрокоманда?

Микрокоманда -- часть исполнения команды.

Что такое команда?

Команда -- это способ управления Базовой ЭВМ. Тривиальный пример команд: CLA (0200), ADD 0x042 (4042). Команды хранятся в памяти БЭВМ.

Микрокоманды складываются в некую цепочку (обязанностей), образуя так называемый интерпретатор (*в нашем случае это интерпретатор Базовой ЭВМ*). Когда мы отдаём любой приказ Базовой ЭВМ, первое, что она делает -- **обращается к интерпретатору**, где заданы **простейшие потактовые действия**, последовательное исполнение которых **гарантирует выполнение одной команды**. Как только исполнение одной команды будет завершено, Базовая ЭВМ попадёт в начало интерпретатора, таким образом, будет готова исполнить еще одну команду.

Микрокод по своей сути состоит из битов **0/1**, которые отвечают за *открытие того или иного вентиля* в Базовой ЭВМ. Именно из таких закрытий/открытий вентилей и состоит вся логика управления Базовой ЭВМ. Исполнение одной команды может включать в себя исполнение ~20-40 микрокоманд. Весь микрокод (код микрокоманды) хранится в *памяти микрокоманд*, внутри Устройства Управления (*Control Unit*) Базовой ЭВМ.

Память микрокода состоит из **256** (адрес 8-битный/8-разрядный) **40-разрядных ячеек**. Существует **2** типа микрокоманд: **операционные** (*оперируют регистрами/памятью Базовой ЭВМ*) и **управляющие** (*управляют логикой вычисления, то есть выбирают нужную операционную микрокоманду*). За выбор этого типа отвечает **39-й бит**.

Устройство микрокода:

0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0  
0 . 0 0 0 0  
39 38 37 36 . 35 34 33 32 . 31 30 29 28 . 27 26 25 24 . 23 22 21 20 . 19 18 17 16 .  
15 14 13 12 . 11 10 9 8 . 7 6 5 4 . 3 2 1 0

Состоит из **10** тетрад (блоков по **4** двоичных бита) -> **40** бит.

Устройство управляющей микрокоманды:

1XXX.XXXC.JUMP.ADDR.CBIT.MASK.COMM.RALU.READ.REGS

Биты X - не используются.

Шестнадцатеричная система:

8CJABMCARR

- Первая тетрада (8) -- всегда 8\*
- Вторая тетрада (C - Compare) -- либо 1, либо 0 -- однобитовое поле сравнения.
- Третья и четвертая тетрады (JA - Jump Address): адрес перехода в памяти микрокоманд (напоминаю, адрес 8-битовый)

- Пятая и шестая тетрада (*BM* - Bit Mask) -- Битовая маска для проверки значения с коммутатора
- Седьмая тетрада (*C* - Commutator) -- Правила для коммутатора
- Восьмая тетрада (*A* - ALU) -- Правила для АЛУ
- Девятая и десятая тетрада (*RR* - Read Registers) -- Правила для чтения регистров

*N/B! \* - Первая тетрада обязательно должна иметь вид: IXXX (39-й бит = 1), при этом биты с 38-36 не используются, поэтому вполне может быть и число, большее 0x8, например 0x9 (1001). То есть, 9000000000 -- тоже управляющая микрокоманда.*

Рассмотрим это на примере микрокоманды: 80C4101040 - GOTO INT @C4

- Первая тетрада свидетельствует о том, что это *управляющая микрокоманда*.
- Вторая тетрада свидетельствует о том, что *сравнение будет с нулем*.
- Третья и четвертая тетрада: *если сравнение будет удачным, мы перейдем на микрокоманду по адресу C4*.
- Пятая и шестая тетрада: *Битовая маска: 0001.0000* - мы будем смотреть только на 4<sup>й</sup> бит из тех, что нам придут из коммутатора.
- - Напоминаю, что из коммутатора приходят только младшие (первые) 8 разрядов (с 0 по 7 - это называется *младший байт*)
- Седьмая тетрада: Правила коммутатора: 0001
- - -> 12-й бит = 1 (правила коммутатора это биты с 12 по 15) -> LTOL (см. Шпаргалку вентильных схем в третьей части презентации по ОПД)
- - LTOL: Lower To Lower - младший байт неизменно пройдет через коммутатор, останется младшим байтом
- - // вопрос для опытных: что будет со старшим байтом?
- Восьмая тетрада: Правила АЛУ: 0000 -> классическая сумма (бит SORA 11-й = 0, остальные биты говорят о том, что ничего не инвертируется, и +1 сигнал тоже не подается)
- Девятая и десятая тетрада: 0100.0000 -> 6<sup>й</sup> бит = 1 -> читаем регистр PS.

Итого, что мы имеем:

Содержимое регистра PS направляется в АЛУ с правилом сложения с нулем (потому что на другом входе АЛУ ничего нет), младший байт из АЛУ пройдет через коммутатор, и направиться в Устройство Управления. В Устройстве Управления в данный момент проверяется соответствие 4-го бита на 0, и если это условие соблюдается, мы перейдем по адресу C4.

*N/B! Так как PS(4) всегда равен 0 (см. Лекцию 4 части 3), то это де-факто означает безусловный переход. - GOTO INT @C4*

Рассмотрим еще пример:

800C404002 - IF CR(14)=0 GOTO CHKABS @0C

- Первая тетрада - все как обычно
- Вторая тетрада - сравним с нулем
- Третья и четвертая - Переход на 0C, если сравнение будет удачным

- Пятая и шестая - Битовая маска: 0100.0000 -> проверим 6-й бит, который придет с коммутатора (почему не 14й???)
- Седьмая - Правила коммутатора: 0100 -> HTOL = 1 (14-й бит микрокоманды) -> передадим старший байт в младший.
- Восьмая - Правила АЛУ: 0000 -> сложение
- Девятая и десятая: 0000.0010 -> RDCR (2-й бит), читаем CR.

Итого, что мы имеем:

Содержимое регистра CR направляется в АЛУ с правилом сложения с нулем, старший байт из АЛУ пройдя через коммутатор, перепишется в младший байт, младший байт традиционно направляется в Устройство Управления. Там на данный момент проверяется, чтобы 6-й бит пришедшего из коммутатора значения был равен 0, в этом случае будет сделан переход на микрокоманду 0C.

Рассмотрим последний пример: 815C081040 - IF PS(N) = 1 GOTO BR @5C

В отличие от примеров выше, тут уже проверка на 1. Переход будет произведен в адрес 5C, битовая маска говорит о том, что проверяется 3-й бит младшего байта, на коммутаторе стоит правило 0001 (LTOL), на АЛУ стоит правило 0000 (сумма), а смотреть мы будем содержимое регистра PS.

Так как 3-й бит PS это флаг N -> метка для нашей микрокоманды будет выглядеть так: IF PS(N) = 1 GOTO BR @5C

*N/B! Никакая управляющая микрокоманда не меняет значения ни одного регистра, поэтому вы можете смело в таблице трассировки, видя управляющую микрокоманду переписывать значения регистров IP, CR, AR, DR, SP, BR, AC, NZVC те, что были в прошлой строчке.*

С управляющими микрокомандами все довольно просто и понятно, перейдем к операционным микрокомандам.

Устройство операционной микрокоманды:

0HXX.IOMM.WRTE.REGS.COMM.COMM.COMM.RALU.READ.REGS

Биты X - не используются.

Шестнадцатеричный код: 0IWRCCCCARR

- Первая тетрада (*0*): 4 для команды Halt (остановка Базовой ЭВМ), **во всех остальных случаях - 0**.
- Вторая тетрада (*I* - IO/MEM): Приказы для IO + Memory
- Третья и четвертая тетрада (*WR* - Write Registers): Приказы для записи значений в регистры.
- Пятая, шестая тетрады (*CC* - Commutator): Расширенные правила для коммутатора
- Седьмая тетрада (*C* - Commutator): Тоже правила для коммутатора (но уже Вам знакомые)
- Восьмая тетрада (*A* - ALU): Правила для АЛУ (Здесь все тоже самое, что и было раньше)
- Девятая и десятая тетрада (*RR* - Read Registers): Правила чтения регистров (Вы это уже знаете)

Подробнее про каждый бит в операционной микрокоманде:

Блок чтения регистров (9 и 10 тетрады):

- 00 - RDDR

- 01 - RDCR
- 02 - RDIP
- 03 - RDSP
- 04 - RDAC
- 05 - RDGR
- 06 - RDPC
- 07 - RDIR

*N/B! Сначала перечислены правые регистры сверху вниз, потом левые регистры сверху вниз. (А сейчас откройте БЭВМ...)*

Блок правил АЛУ (8 тетрада):

- 08 - COMR
- 09 - COML
- 10 - PLS1
- 11 - SORA

Блок правил Коммутатора (5, 6 и 7 тетрады):

- Общий блок (7 тетрада / и те, и другие это используют):
- 12 - LTOL
- 13 - LTOH
- 14 - HTOL
- 15 - HTOH
- Операционные микрокоманды (5 и 6 тетрады):
- 16 - SEXT
- 17 - SHLT
- 18 - SHL0
- 19 - SHRT
- 20 - SHRF
- 21 - SETC
- 22 - SETV
- 23 - STNZ

Блок записи в регистры (3 и 4 тетрады):

- 24 - WRDR
- 25 - WRCR
- 26 - WRIP
- 27 - WRSP
- 28 - WRAC
- 29 - WRBR
- 30 - WRPS
- 31 - WRAR

*N/B! Записывать в IR нельзя, вместо него записывать предлагается в AR. Тоже самое, что и читать из AR нельзя, вместо него предлагается читать из IR.*

Блок IO + MEM (2 тетрады):

- 32 - LOAD
- 33 - STOR
- 34 - IO
- 35 - INTS

*N/B! Отдельно 38 - HALT Микрокоманда остановки БЭВМ: 4000000000*

*N/B! Чтобы проще все это запомнить, запоминайте сначала управляющие микрокоманды, потом посмотрите на сходство управляющих микрокоманд с операционными, дальше запомните порядок расположения блоков в операционных микрокомандах, а уже потом взгляните на другие N/B, которые подчеркивают закономерности для запоминания уже конкретных блоков микрокода.*

Рассмотрим на примере: 00A0009004 - IP -> AR, BR

- Первая тетрада сигнализирует о том, что это операционная команда
- Вторая тетрада сигнализирует о том, что никаких операций с памятью и вводом/выводом не происходит
- Третья и четвертая тетрады: 1010.0000; Запись будет произведена в AR (31-й бит) и в BR (29-й бит).
- Пятая и шестая тетрада говорят о том, что флаги проставлены не будут, сдвига не будет, расширение знака тоже не произойдет.
- Седьмая тетрада: 1001: НТОН & LTOL, оба байта передаются в прямом порядке, ничего не меняется друг с другом.
- Восьмая тетрада: Правила АЛУ: 0000 -> классическая сумма
- Девятая и десятая тетрада: 0000.0100 -> считываем IP.

Итого мы имеем:

Содержимое регистра IP пройдя через АЛУ сложится с 0, пройдя через коммутатор сохранит расположение старших и младших байтов и запишется в регистры AR и BR. Признаки результата не устанавливаются.

*N/B! Вспомните таблицу системы команд Базовой ЭВМ.*

*В ней \* \* 0 -устанавливаются для всех команд, результат которых записывается в аккумулятор и не выходит за пределы 16 разрядов*

*\* \* \* \* - для команд, которые могут получить результат в AC, выходящий за пределы 16-разрядной сетки,*

*а - - - - указано для всех команд, которые вообще не трогают аккумулятор.*

*Здесь все работает по точно такой же логике. Напоминаю, за установку признаков результата отвечают старшие 3 бита пятой тетрады. // Вопрос для опытных: почему N и Z ставятся неразрывно друг от друга?*

**Немного разобравшись в этом всем деле, поговорим про рубежную работу.**

Почему важно понимать то, что я расписал выше?

Скорее всего знание, описанное выше, вам не пригодиться (написано студентами Афанасьева для студентов Афанасьева). Но если без шуток, вам могут выдать на рубежке таблицу интерпретатора Базовой ЭВМ без меток, и там уже не так очевидно, что 0001009201 это  $\sim\theta + DR \rightarrow DR$ . Это придется считать самому, а когда вы не дружите с микрокодом, разобрать это будет довольно непросто

(Ex: 4-я тетрада тут говорит о записи в DR, 7-я тетрада о сохранении порядка старшего и младшего байтов, 8-я тетрада 0010 -> инвертировать левый вход АЛУ, а 9 и 10-я тетрады говорят о том, что чтение будет из DR)

Что меня ждет на рубежке?

Предлагаю вам прочитать официальный текст задания на рубежную работу:

Запишите последовательность микрокоманд для выполнения заданной команды до машинного цикла прерывания. Заполните таблицу значениями регистров после выполнения каждой микрокоманды. Если происходит обращение к памяти, укажите адрес памяти, значение и операцию. Содержимое памяти по адресу 0x570:

24A3, 319B, 9B0D, F7B7, 20BD, 2542, 057B, 8985, D81A, 612B, 307E, BA30, 2FE0, 28F8, 6957, 194B. Исходная команда располагается по адресу 0x57D.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after
			57D	DBA6	57D	9AD5	057D	D605	0000	

Что здесь вообще написано? Вообще задание гласит о том, что надо сделать один пробег интерпретатора БЭВМ с исходными данными, которые изначально даны в таблице + нам дан кусочек памяти, наличие которого необходимо и достаточно для выполнения рубежной работы.

Вот давайте с этого куска памяти и начнем.

Адрес	Значение
570	24A3
571	319B
572	9B0D
573	F7B7
574	20BD
575	2542
576	057B
577	8985
578	D81A
579	612B
57A	307E
57B	BA30
57C	2FE0
57D	28F8
57E	6957
57F	194B

Нам это еще не раз пригодиться, поверьте...

Вернемся к таблице. Каждую новую выполняемую микрокоманду вместе со значениями регистров после ее исполнения надо писать на новой строчке таблицы трассировки. Собственно говоря это и называется трассировка; вы пишете микрокоманду и значения регистров Базовой ЭВМ после выполнения каждой микрокоманды.

*N/B! Как вы уже могли догадаться, мы трассируем только одну команду! Но много микрокоманд.*

Ну давайте же вместе с вами и сделаем эту рубежку:

Для честности эксперимента, мы будем пользоваться [только интерпретатором без меток](#).

Начинается все с первой микрокоманды:

00A0009004

План:

1. 1-я тетрада = 0? Да -> Следующий адрес микрокоманды (MP after) всегда будет на 1 больше текущего. Запишем сразу это, пока не забыли:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01										02	00A0009004

Далее, оценим, что это за микрокоманда и какие регистры будут изменены:

Микрокоманда операционная (0000)

Операций ввода/вывода и с памятью отсутствуют (0000)

Запись будет произведена в регистры AR, BR (1010.0000)

LTOL НТОН (0000.0000.1001)

Правила АЛУ: Классическая сумма (0000)

Чтение из IP (0000.0100).

Итого: IP -> ALU (+0) -> COMM: LTOL; НТОН -> AR, BR.

Так и запишем:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			<b>57D</b>	DBA6	<b>57D</b>	<b>9AD5</b>	057D	D605	0000	02	00A0009004

Поздравляю, мы заполнили первую строчку, теперь мы можем заполнять вторую строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
<b>02</b>											<b>0104009420</b>

Бойцы рубежки продолжают сражаться. Дальше мы смотрим на следующую микрокоманду (*Откуда мы ее берем? Из интерпретатора!*) 0104009420

Внимательно посмотрим на нее:

Операционная микрокоманда (0000) -> следующая микрокоманда будет на 1 больше предыдущей (03)

Обратите внимание, здесь произойдет операция чтения из памяти (2-я тетрада: 0001)

*N/B! Запись в память происходит строго в Регистр Данных (DR)! Адрес берется из AR, из прошлой строчки таблицы мы можем сделать вывод о том, что обращение произойдет по адресу AR = 57D. Содержимое памяти нам дано в условии задания!*

Запись произойдет в регистр IP (0000.0100)

LTOL, HTOH (0000.0000.1001)

Произойдет операция PLS1 на АЛУ (0100)

Чтение произойдет из BR (0010.0000)

Итого: MEM(AR) -> DR; BR -> ALU (+1) -> COMM: LTOL, HTOH -> IP

Запишем строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	<b>57D</b>	<b>28F8(L)</b>	<b>57E</b>	DBA6	57D	<b>28F8</b>	<b>057D</b>	D605	0000	<i>03</i>	0104009420
<b>03</b>											<b>0002009001</b>

Продолжаем. Следующая микрокоманда (тоже операционная): 0002009001

Вторая тетрада гласит о том, что операции ввода/вывода и использования памяти отсутствуют.

Запись будет осуществляться в CR (0000.0010)

LTOL, HTOH (0000.0000.1001)

Классическая сумма (0000)

Чтение из DR (0000.0001)

Итого: DR -> ALU +0 -> COMM: LTOL, HTOH -> CR

Запишем строчку:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	<b>28F8</b>	57D	<b>28F8</b>	057D	D605	0000	<i>04</i>	0002009001
<b>04</b>											<b>8109804002</b>

Ура! Первая управляющая микрокоманда!

Рассмотрим этого покемона попороднее (кто еще помнит, что там было в начале этого документа?)

8109804002

Управляющая микрокоманда (1-я тетрада = 8)

*N/B! Управляющая микрокоманда = переписываем все значения регистров IP, CR, AR, DR, BR, AC, NZVC; они меняться не будут:*

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000		8109804002

Проверяется бит на 1

В случае успешного сравнения прыжок будет на адрес 09

Битовая маска: 1000.0000. Смотрим 7-й бит, пришедший с коммутатора

Правило коммутатора: 0100: (HTOL) (проверка 7-го бита превращается в проверку 15-го бита)

Правило АЛУ: сумма (0000)

Чтение происходит из CR (0000.0010)

Итого: IF CR(15) = 1 then GOTO 09

N/B! Посмотрим на содержимое CR: 28F8

Двоичное представление: 0010.1000.1111.1000

Как можем увидеть, 15-й разряд CR = 0, что удовлетворяет нашему условию.

Условие не выполнено. Вместо перехода на адрес 09 мы перейдем на следующую микрокоманду.

N/B! Не обязательно додумывать, что же там будет дальше. Просто следуйте алгоритму.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
<b>05</b>											<b>810C404002</b>

Управляющих команд будет больше, чем операционных, поэтому привыкаем к ним (не зря мы их разбирали первыми)

**810C404002**

Проверяем на 1

Переходим на адрес 0C

Битовая маска: 0100.0000 -> проверяем 6-й бит, пришедший с коммутатора.

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)

Читаем с CR (0000.0010)

Итого: IF CR(14) = 1 then GOTO 0C

14-й бит CR не равен 1, поэтому мы просто идем дальше, записав в MP after значение 06.

Остальные регистры просто переписываем:

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
<b>06</b>											<b>810C204002</b>

Продолжаем серию беспощадных проверок, которые нам даровали управляющие микрокоманды.  
**810C204002**

Проверяем на 1

Переходим на адрес 0C

Битовая маска: 0010.0000 -> проверяем 5-й бит, пришедший с коммутатора.

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)

Читаем с CR (0000.0010)

Итого: IF CR(13) = 1 then GOTO 0C

13-й бит CR действительно удовлетворяет нашему условию. Переход на 0C будет выполнен.

Обратите внимание на сходство с предыдущей микрокомандой...

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
<b>0C</b>											<b>8024084002</b>

Новый день, новый микрокод.

**8024084002**

Управляющая микрокоманда (вопрос: что мы делаем с регистрами IP, CR, AR, DR, BR, AC, NZVC?)

Проверяем на 0

В случае успешной проверки переходим на адрес 24

Битовая маска: 0000.1000 -> проверяется 3-й бит

Правило коммутатора: HTOL (0100)

Правило АЛУ: сумма (0000)

Читаем регистр CR (0000.0010)

Итого: IF CR(11) = 0 then GOTO 24

*N/B! Если вы так и не поняли, почему именно 10-й бит, (это нормально, мы тоже ничего не поняли,) то вот вам простая формула, что делать, если правило коммутатора оказалось HTOL: Проверяется (3 + 8)-й бит: прибавляем 8.*

Почему? Давайте посмотрим сначала на битовую маску:

0000.1000 (проверим 3-й бит)

Потом посмотрим на то, что сделал коммутатор:

HTOL - операция переноса старшего байта (разрядов с 8 по 15-й) в младший (с 0 по 7-й)

Рассмотрим старший байт:

X X X X . X X X X  
15 14 13 12 . 11 10 9 8

Наложив на все это битовую маску, получаем, что де-факто проверяется 11-й бит, а выражается все это по формуле (3 + 8)-й бит.

Ну и потому что 8 это длина байта, а мы как раз скинули младший байт.

10-й бит CR не равен нулю, проверка не пройдена. Это означает, что счетчик микрокоманд просто увеличится на 1.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
<b>0D</b>											<b>0020011002</b>

Так-с, мы наткнулись на операционную микрокоманду... Грядут большие изменения в регистрах, которые уже как 4 микрокоманды никак не менялись.

Вспоминаем, что читали буквально недавно (например то, что следующая микрокоманда будет 0D + 1 = 0E).  
**0020011002**

Операций ввода-вывода и доступа к памяти отсутствует (2-я тетрада = 0)

Значение будет записано в BR (0010.0000)

А вот с правилами коммутатора тут интереснее, чем 0000.0000.1001:

Они вот такие: **0000.0001.0001** (0x011 - тетрады 5-я, 6-я и 7-я)

С 7-й тетрадой все понятно: LTOL

А вот в 6-й тетраде младший бит отвечает за операцию расширения знака (SEXT)

Правила АЛУ: сумма (**0000**)

Чтение происходит из CR (**0000.0010**)

Итого: CR → ALU (+0) → COMM: SEXT(7 → 8...15) → BR

M P	Адре с	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	<b>28F8</b>	57D	28F8	<b>FFF8</b>	D605	0000	0E	0020011002
<b>0E</b>											<b>811C044002</b>

Ну, недолго операционные команды радовали нас, поехали проверять дальше (кстати, без меток тут довольно легко ошибиться, будьте внимательны)

**811C044002**

Проверяем на 1

Переход на 1с

Битовая маска: **0000.0100** → проверяем 2-й бит.

Правила коммутатора: **0100**: HTOL (→ 2 + 8 → 10-й бит)

Правила АЛУ: **0000** / сумма

Читаем содержимое CR (**0000.0010**)

Итого: IF CR(10) = 1 then GOTO 1с

Это неправда, напоминаю содержимое CR:

**28F8 = 0010.1000.1111.1000**

Значит, мы просто пойдем дальше по микрокоду (но вы верьте, что когда-нибудь это закончится):

MP	Адресс	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
<b>0F</b>											<b>0080009024</b>

Операционная микрокоманда подошла, откуда ее не ждали, а значит далее мы перейдем на микрокоманду 10.

**0080009024**

Посчитаем значения регистров:

Операции чтения/записи/ввода-вывода не происходит.

Запись произойдет в AR (1000.0000).

Правила коммутатора просты: LTOL, НТОН (0000.0000.1001)

Правила АЛУ: 0000 / сумма

Чтение произойдет из регистров BR и IP (0010.0100).

N/B! Чтение может проходить из двух регистров. Просто до этого АЛУ суммировала содержимое регистра с нулем, а сейчас на левый вход подается содержимое BR, на правый -- IP.

Итого: BR, IP -> ALU (BR + IP) -> COMM: НТОН, LTOL -> AR.

Посчитаем BR + IP:

FFF8 + 57E = 0x10576.

В AR дойдут только 0x576.

M P	Адре с	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			<b>57E</b>	28F8	<b>576</b>	28F8	<b>FFF8</b>	D605	0000	<i>10</i>	0080009024
<b>10</b>											<b>0100000000</b>

Вот бы все микрокоманды были такие, как та, что вы увидите сейчас...

**0100000000**

Тут все просто. Мы произведем операцию чтения из памяти (2-я тетрада: 0001)  
 И все. Вопрос на понимание: чему будет равна следующая микрокоманда? Не бойтесь ответить даже самый глупый, на ваши взгляд, ответ.

Кто помнит, что было в ячейке по адресу 576???

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11											8114024002

Вернемся к управляющим микрокомандам.

8114024002

Проверка на 1

Переход в случае успешной проверки на 14 (не очень далеко кнш)

Битовая маска: проверяем 1-й бит, пришедший с коммутатора. (0000.0010)

Правило коммутатора: 0100 / HTOL

Правило АЛУ: 0000 / сумма

Чтение CR (0000.0010)

Итого: IF CR(9) = 1 then GOTO 14 (!!! bug fix by @O\_skull32)

Но у нас это не так, поэтому переход будет на 1 микрокоманду вперед.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
<b>12</b>											<b>81E0014002</b>

Вечер управляющих микрокоманд в самом разгаре!  
**81E0014002**

Проверка на 1

Переход на E0 в случае успешной проверки.

Битовая маска: проверяем 0-й бит, пришедший с коммутатора (0000.0001)

Правило коммутатора уже до боли знакомое нам: HTOL (0100)

Правило АЛУ тоже: 0000 (кто помнит, что это такое?)

Читаем содержимое все еще CR (0000.0010)

Итого: IF CR(8) = 1 then GOTO E0

У нас это не так, поэтому двигаемся дальше на 13-ю микрокоманду.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
<b>13</b>											<b>8024101040</b>

А кто-то говорил, что это легко?  
**8024101040**

Управляющая микрокоманда, если кто забыл (все еще держу в курсе, что содержимое регистров IP, CR, AR, DR, BR, AC, NZVC не изменяется)

## Проверка на 0

Переход на адрес 24, если проверка удачная

Битовая маска: 0001.0000 -> проверяем 5-й бит

Правило коммутатора: LTOL (0001)

Правило АЛУ: сумма (0000)

Читаем содержимое PS (0100.0000)

*N/B! Я напоминаю, что 5-й бит PS всегда равен 0. То самое, о чем говорил Клименков на лекции.*

Ну итого нас вынуждают прыгнуть на 24-й адрес получается...

Быстренько расправимся с этой управляющей микрокомандой!

8026804002

## Проверяем на 0

Прыжок на 26 в случае удачной проверки

Проверяем 7-й бит (1000.0000)

Правило коммутатора: HTOL (0100)

Правило АЛУ: 0000 / сумма

Читаем CR (0000.0010).

Micro. II CR(15), 3 min 5015-25

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
----	-------	------------------------	----	----	----	----	----	----	------	-------------	-----------

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
<b>26</b>											<b>0080009001</b>

Настало время операционной микрокоманды:  
**0080009001**

Чтение/запись/ввод-вывод не производится.

Запись производится в AR (**1000.0000**)

Правило коммутатора: **0000.0000.1001**: LTOL, НТОН

Правило АЛУ: **0000** / сумма

Чтение происходит из DR (**0000.0001**).

Итого: DR -> ALU (+0) -> COMM: LTOL, НТОН -> AR

Просто перенесли содержимое из одного регистра в другой. Классика!

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	<b>57B</b>	<b>057B</b>	FFF8	D605	0000	27	0080009001
<b>27</b>											<b>0100000000</b>

Кажется, мы уже видели эту микрокоманду! Поэтому, чтобы сэкономить ваше и мое время, воспользуемся стараниями из 10-й микрокоманды

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	<b>57B</b>	<b>BA30(L).</b>	57E	28F8	<b>57B</b>	<b>BA30</b>	FFF8	D605	0000	28	0100000000
<b>28</b>											<b>813C804002</b>

Поздравляю, все выборки закончились! В DR хранится тот самый заветный операнд, который сделает логическое И (КОП: 0x2) с содержимым аккумулятора... Ждем этот момент с нетерпением. А пока -- еще одна управляющая микрокоманда.  
**813C804002**

Проверка на 1

Переход на 3С (я уже буду опускать, что в случае успешной проверки)

Битовая маска: проверка 7-го бита. (1000.0000)

Правило коммутатора: HTOL (0100)

Правило АЛУ: 0000 / сумма

Читаем CR (0000.0010).

Итого: IF CR(15) = 1 then GOTO 3С

Условие не верно. Идем дальше.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
<b>29</b>											<b>8130404002</b>

Устали? Я готов подождать.

**8130404002**

Проверка на 1

Переход на 30

Проверка 6-го бита (0100.0000)

HTOL (0100)

Сумма (0000)

Чтение CR (0000.0010).

Итого: IF CR(14) = 1 then GOTO 30

У нас это не так, поэтому продолжаем.

*N/B! The biggest mistake in your life! :c: owl from hogwarts: 0x29 + 0x1 = 0x30*

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
<b>2A</b>											<b>812D104002</b>

Спидран по управляющей микрокоманде, поехали  
**812D104002**

Проверка на 1

Переход на 2D

Проверка 4-го бита (0001.0000)

HTOL (0100)

Сумма (0000)

Чтение CR (0000.0010)

Итого: IF CR(12) = 1 then GOTO 2D

Неправда, поэтому мы туда не пойдем.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
<b>2B</b>											<b>0010C09811</b>

Финишная прямая, господа. Это последняя операционная команда (спойлер, упс)  
**0010C09811**

Операций чтения/записи/ввода-вывода отсутствуют.

Запись произведется в AC (**0001.0000**)

*N/B! Будьте начеку и помните, что как только вы видите запись в AC, где-то рядом надо быть готовым увидеть установку флагов NZVC...*

Правила коммутатора: **0xC09 (1100.0000.1001)**. С **0x9** все просто: LTOL, HTOL.

А вот биты тетрады **0xC (1100)** отвечают за установку флагов N, Z и V (STNZ - 23-й разряд, SETV - 22-й).

Правила АЛУ: **1000. SORA = 1** -> Операция логического умножения (И)

Читаем регистры AC и DR. (**0001.0001**)

Итого: AC, DR -> ALU (AC & DR) -> COMM: LTOL, HTOH + N, Z, V -> AC

Здесь главное очень аккуратно переписать все в таблицу.

**D605 & BA30 = 9200 (N = 1; Z = 0; V = 0)**

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
2B			57E	28F8	57B	BA30	FFF8	<b>9200</b>	<b>1000</b>	2C	0010C09811
<b>2C</b>											<b>80C4101040</b>

Вроде бы как мы все сделали. Наверное, это должна быть последняя управляющая микрокоманда.  
**80C4101040**

Проверка на 0

Переход на C4

Проверка 5-го бита (0001.0000)

Правило коммутатора: LTOL (0001)

Правило АЛУ: 0000 / сумма

Чтение PS (0100.0000)

Кажется, 5-й бит PS всегда 0, поэтому мы "безусловно" переходим на C4.

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
			57D	DBA6	57D	9AD5	057D	D605	0000		
01			57D	DBA6	57D	9AD5	057D	D605	0000	02	00A0009004
02	57D	28F8(L)	57E	DBA6	57D	28F8	057D	D605	0000	03	0104009420

MP	Адрес	Значение (Операция)	IP	CR	AR	DR	BR	AC	NZVC	MP after	Microcode
03			57E	28F8	57D	28F8	057D	D605	0000	04	0002009001
04			57E	28F8	57D	28F8	057D	D605	0000	05	8109804002
05			57E	28F8	57D	28F8	057D	D605	0000	06	810C404002
06			57E	28F8	57D	28F8	057D	D605	0000	0C	810C204002
0C			57E	28F8	57D	28F8	057D	D605	0000	0D	8024084002
0D			57E	28F8	57D	28F8	FFF8	D605	0000	0E	0020011002
0E			57E	28F8	57D	28F8	FFF8	D605	0000	0F	811C044002
0F			57E	28F8	576	28F8	FFF8	D605	0000	10	0080009024
10	576	057B(L)	57E	28F8	576	057B	FFF8	D605	0000	11	0100000000
11			57E	28F8	576	057B	FFF8	D605	0000	12	8114024002
12			57E	28F8	576	057B	FFF8	D605	0000	13	81E0014002
13			57E	28F8	576	057B	FFF8	D605	0000	24	8024101040
24			57E	28F8	576	057B	FFF8	D605	0000	26	8026804002
26			57E	28F8	57B	057B	FFF8	D605	0000	27	0080009001
27	57B	BA30(L)	57E	28F8	57B	BA30	FFF8	D605	0000	28	0100000000
28			57E	28F8	57B	BA30	FFF8	D605	0000	29	813C804002
29			57E	28F8	57B	BA30	FFF8	D605	0000	2A	8130404002
2A			57E	28F8	57B	BA30	FFF8	D605	0000	2B	812D104002
2B			57E	28F8	57B	BA30	FFF8	9200	1000	2C	0010C09811
2C			57E	28F8	57B	BA30	FFF8	9200	1000	C4	80C4101040

Как только мы увидели, что произошел переход на C4, мы **останавливаемся**. Потому что *C4 -- это начало цикла прерывания*, а нас просили написать ровно до момента этого цикла.

Выдыхаем, открываем шампанское, рубежка написана, а перед вами полная трассировка микрокоманд, которые в сумме составляют исполнение команды 28F8. Сверить ответ вы можете в методичке, вариант взят оттуда. А также используя [режим запуска БЭВМ cli/dual](#) + консольной команды `c1`, которая включит потактовую трассировку (см. [Трассировка в БЭВМ](#)).

! Чтобы закрепить полученные знания решений рубежки, откройте, например, вот эти [табличные варианты](#) (и кстати, там есть интерпретатор с метками! Первые пару вариантов рекомендую прорешать именно пользуясь меточным интерпретатором, а уже потом приступать к более сложным вариантам, если конечно понимаете и опасаетесь, что вам дадут трассировку микрокоманд с [интерпретатором без меток](#))

*Вопрос опытным людям: Я не использовал интерпретатор с метками во время решения этого варианта. А вы?*

*Древние русы собирались и решали на механических ЭВМ эти рубежки без калькуляторов, и с первого раза. Так решите и вы!*

Благодарю за прочтение.

---

Если вы нашли техническую проблему в данном тексте, писать сюда: [@zerumi](#)

Ну и если как бы Вы это до конца дочитали, вам это помогло и было полезным, то купите, пожалуйста, автору кофе, он старался: +7 (937) 653-50-50 / Сбер, Тинькофф

Большое спасибо!

---

Используемая литература:

1. Методические указания к лабораторным работам по курсу "Основы профессиональной деятельности" / В. В. Кириллов, А. А. Приблуда, С. В. Клименков, Д. Б. Афанасьев, - Универститет ИТМО - 52 с. 2019г.
2. Основы профессиональной деятельности. Часть третья (не последняя) / Презентация [Электронный ресурс], 2022г.

rumi

Ну и если как бы Вы это до конца дочитали, вам это помогло и было полезным, то купите, пожалуйста, автору кофе, он старался: +7 (937) 653-50-50 / Сбер, Тинькофф

Большое спасибо!

---

Используемая литература:

1. Методические указания к лабораторным работам по курсу "Основы профессиональной деятельности" / В. В. Кириллов, А. А. Приблуда, С. В. Клименков, Д. Б. Афанасьев, - Универститет ИТМО - 52 с. 2019г.
2. Основы профессиональной деятельности. Часть третья (не последняя) / Презентация [Электронный ресурс], 2022г.