

лаба 3 - исполнение циклических программ

Полезный гайд - <https://github.com/Zerumi/OPD-guide-RU->

Организация одномерных массивов данных в памяти. Организация и обработка массивов с числом измерений, больше чем одно

Массивы располагают элементы данных в памяти. Перебираем элементы массива с помощью LOOP и косвенной относительной адресации.

В БЭВМ как и в PDP-8 и PDP-11 память организована как линейное пространство. Это значит, что независимо от размерности массива данные хранятся в последовательных ячейках памяти. Мы должны учитывать организацию массива (по строкам или столбцам), чтобы правильно вычислять адреса элементов

По строкам: Адрес $A[i][j]$ = базовый адрес + i *кол-во столбцов + j

По столбцам: Адрес $A[i][j]$ = базовый адрес + j *кол-во строк + i

Сравнение значений в БЭВМ. Команды условного и безусловного переходов

Инструкция сравнения команды CMP

AC + (- operand) → признаки результата NZVC

CMP не записывает результат в аккумулятор, но выставляет вспомогательные флаги NZVC

Как сравнивать числа A и B?

$A \leq B$

LD A

CMP B

Разветвления в программах организуются с помощью **команд перехода**. Эти команды при выполнении заданного признаками NZVC условия

осуществляют переход по указанному адресу(запись в IP нового адреса выполнения в программе). Если условие не выполняется, то выполняется команда, которая следует за командой ветвления.

Нужно различать использование ветвлений по признакам, которые подходят для беззнаковых, знаковых чисел или обоих сразу:

Для беззнаковых: после установки NZVC можно проверить командами BCC, BCS

Для знаковых: BLT и BGE

Равенство/Неравенство: CMP или SUB / команды ветвления BEQ и BNE / AND и BEQ

е) Команда ветвления

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
КОП=1111				Расш. КОП				Смещение							

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

Организация циклических вычислений

JUMP M (BR) - переход, к команде, расположенной в ячейке с адресом M

Циклы со счетчиком в БЭВМ организуются при помощи команды "Декремент и пропуск" (**LOOP**). Данная команда уменьшает заданную ячейку памяти,

определяющую число повторений, и проверяет, что в ячейке еще находится положительное число. Если оно действительно положительное, то выполняется следующая после LOOP команда, если число отрицательное или равно нулю, то к счетчику команд добавляется единица, и следующей будет выполнена команда через одну после LOOP

Цикл исполнения

DR - 1 → DR

DR - 1 → BR, DR → MEM(AR)

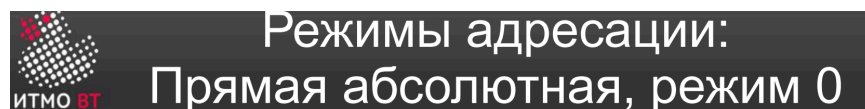
IF (BR 15) == 0; break; IP+1 → IP (переход к следующей команде)

Таким образом, мы уменьшаем значение на 2 и проверяем, не стало ли оно отрицательным. Это выполняется, т.к не установлены признаки результата при данной операции

Режимы адресации БЭВМ

1. Прямая абсолютная(0)

В 11 бите всегда 0. Указывает на ячейку, где хранится необходимое значение. При выполнении операции команда обращается по заданному адресу выбирая или записывая операнд



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

ADD 0x10

0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Адрес полностью кодируется в младших 11 битах
- Непосредственно загружается в AR из кода команды

2. Прямая относительная(со смещение относительно IP), (E)

Код 0xE. Указывает на адрес ячейки. Адрес операнда получается сложением закодированного в команде смещения со счетчиком команд

Adress Fetch:

CR(0...7) → BR

BR + IP → DR



3. Косвенная относительная(8)

Код 0x8. Косвенная адресация означает, что в ячейке памяти, которая вычисляется из адресной части команды через сложение смещения со счетчиком команд, хранится адрес операнда

(В результате после вычисления ячейки, где хранится адрес, ее значение снова используется в качестве адреса, вычисляя расположение операнда в памяти)

Adress Fetch

CR(0-7) → BR

BR + IP → AR, MEM(AR) → DR

4. Косвенная автоинкрементная(A)

Аналогична косвенной относительной, но после загрузки операнда из памяти значение адреса в ячейке памяти увеличивается на 1

Address Fetch

CR(0-7 биты) → BR

BR + IP → AR

MEM(AR) → DR

DR + 1 → DR

DR → MEM(AR)

DR - 1 → DR

5. Косвенная автодекрементная(B)

Перед загрузкой операнда из памяти значение адреса в ячейке памяти уменьшается на 1

Address Fetch

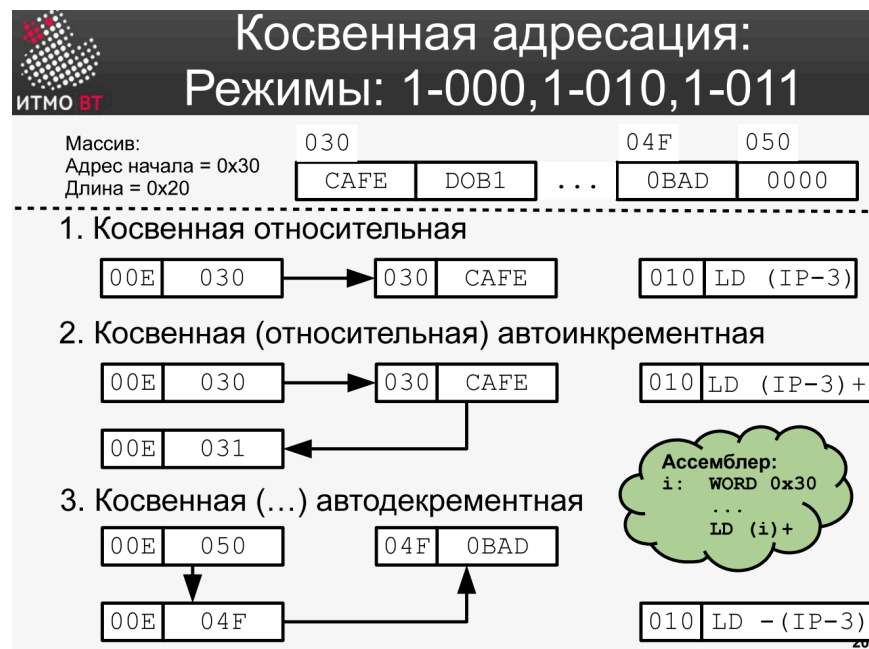
CR(0-7) → BR

BR + IP → AR

MEM(AR) → DR

DR - 1 → DR

DR → MEM(AR)



Код				Мнемоника	Описание	Реализация машинных циклов Address Fetch, Operand Fetch
11	10	9	8			
0	M	M	M	ADD 0ADDR ADD \$L	Прямая абсолютная	$DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	0	0	ADD (L)	Косвенная относительная	$SXT_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	0	1		Резерв	
1	0	1	0	ADD (L)+	Косвенная автоинкрементная (постинкремент)	$SXT_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR, DR + 1 \rightarrow DR,$ $DR \rightarrow MEM(AR), DR - 1 \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	0	1	1	ADD -(L)	Косвенная автодекрементная (преддекремент)	$SXT_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow AR, MEM(AR) \rightarrow DR, DR - 1 \rightarrow DR,$ $DR \rightarrow MEM(AR),$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	0	0	ADD &N ADD (SP+N)	Косвенная относительная, со смещением (SP)	$SXT_CR(0..7) \rightarrow BR,$ $BR + SP \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	0	1		Резерв	
1	1	1	0	ADD L ADD (IP+N)	Прямая относительная	$SXT_CR(0..7) \rightarrow BR,$ $BR + IP \rightarrow DR,$ $DR \rightarrow AR; MEM(AR) \rightarrow DR$
1	1	1	1	ADD #N	Прямая загрузка	$SXT_CR(0..7) \rightarrow BR, BR \rightarrow DR$

Рисунок В.10 Циклы выборки адреса и операнда для различных режимов адресации

Важно помнить, что КОП + режим отнимают у нас 8 бит. Значит, на сам адрес остается всего 8, старший бит отвечает за знак, поэтому остаётся уже **7 бит для смещения**, тогда $[-2^7; 2^7-1] = [-128; 127]$ (исключаем значение 1000 0000)

6. Прямая загрузка в AC(F)

Код 0xF. Для такого формата биты 8-11 установлены в единицы. Команда с режимом адресации «прямая загрузка» по факту не является адресной, а только использует формат адресной команды. Она берет число в битах 0-7 команды в качестве операнда и рассматривает его как знаковое, расширяя знак байта (7 бит копируется в биты 8-15 старшего байта)