

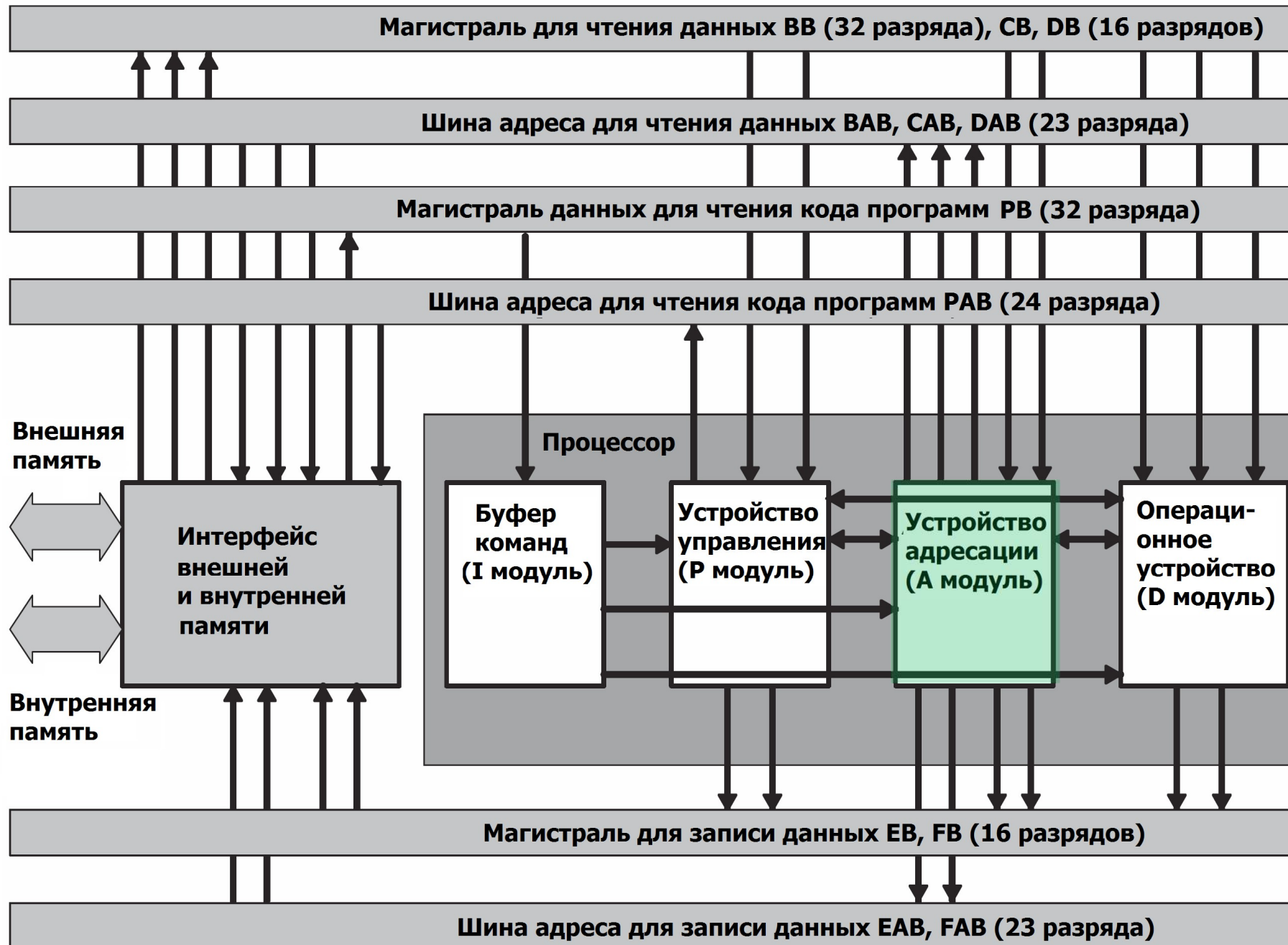


Микропроцессорные устройства обработки сигналов

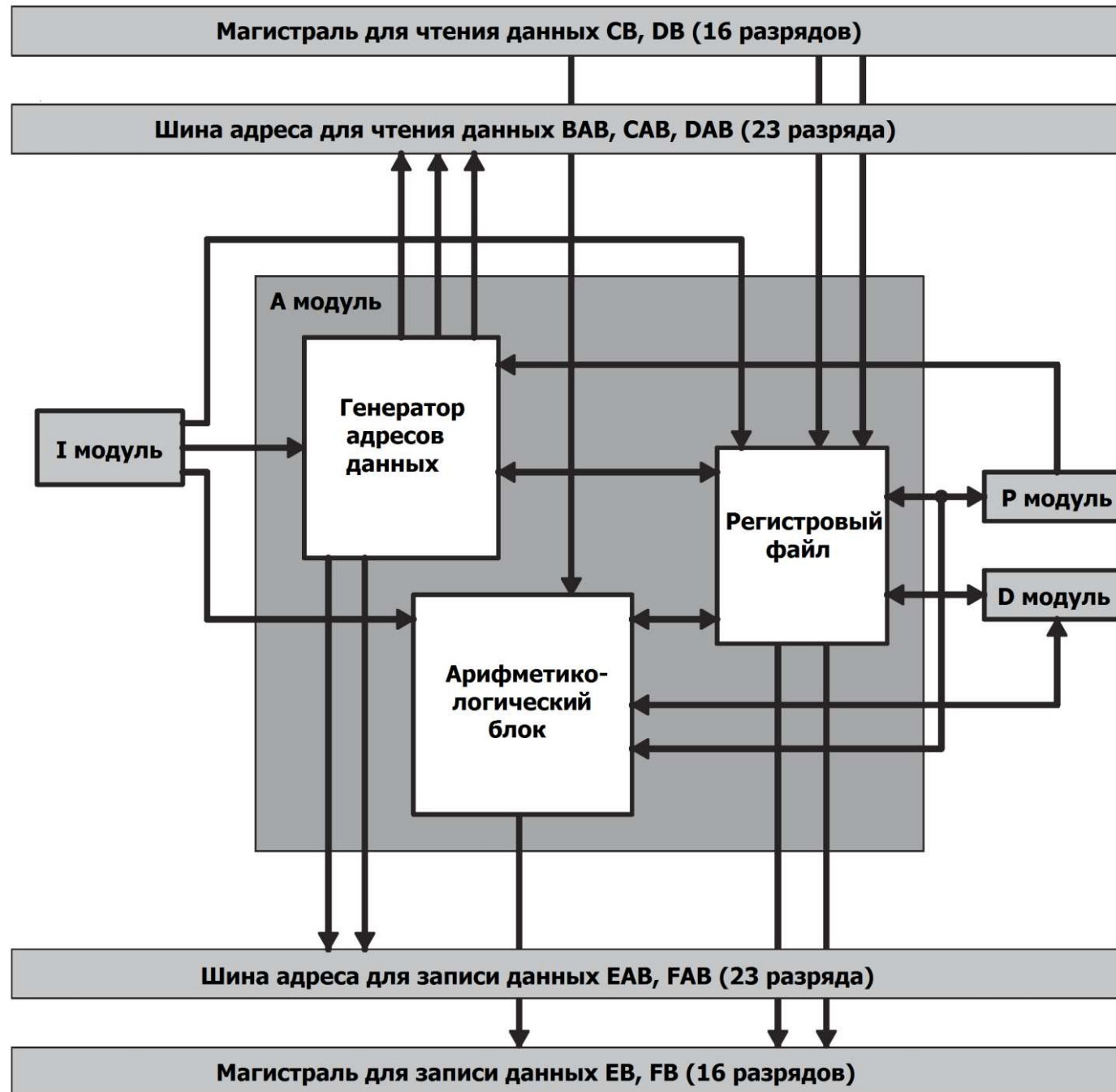
Лекция L07 «Устройство адресации»

<http://vykhovanets.ru/course67/>

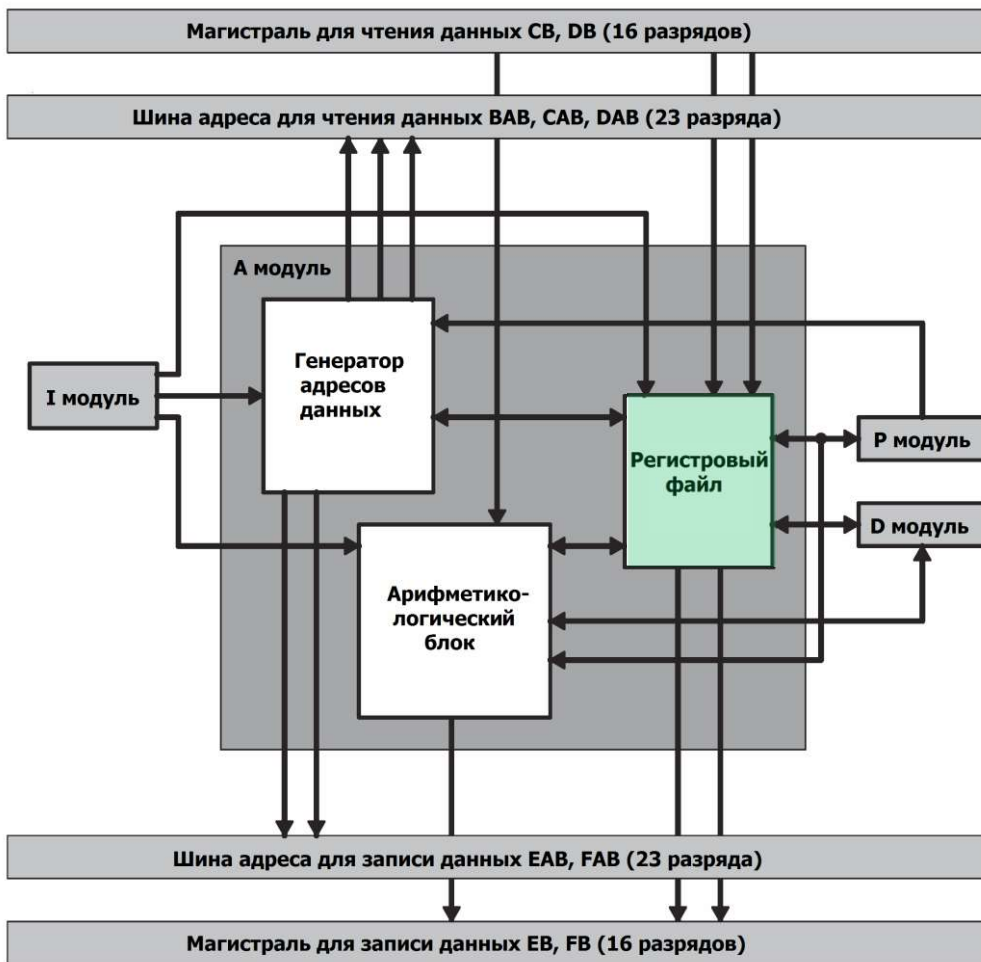
Ядро микропроцессора



Устройство адресации



Регистровый файл А



- Хранение промежуточных данных
- Предоставление операндов и получение результатов операций
- Обмен промежуточными данными с другими модулями и памятью

Регистры страниц данных:

DPH (7 бит), DP (16 бит), PDP (9 бит).

Регистры-указатели:

XSP (23 бита), **XSSP** (23 бита),
XARx (8*23 бита), **XCDP** (16 бит).

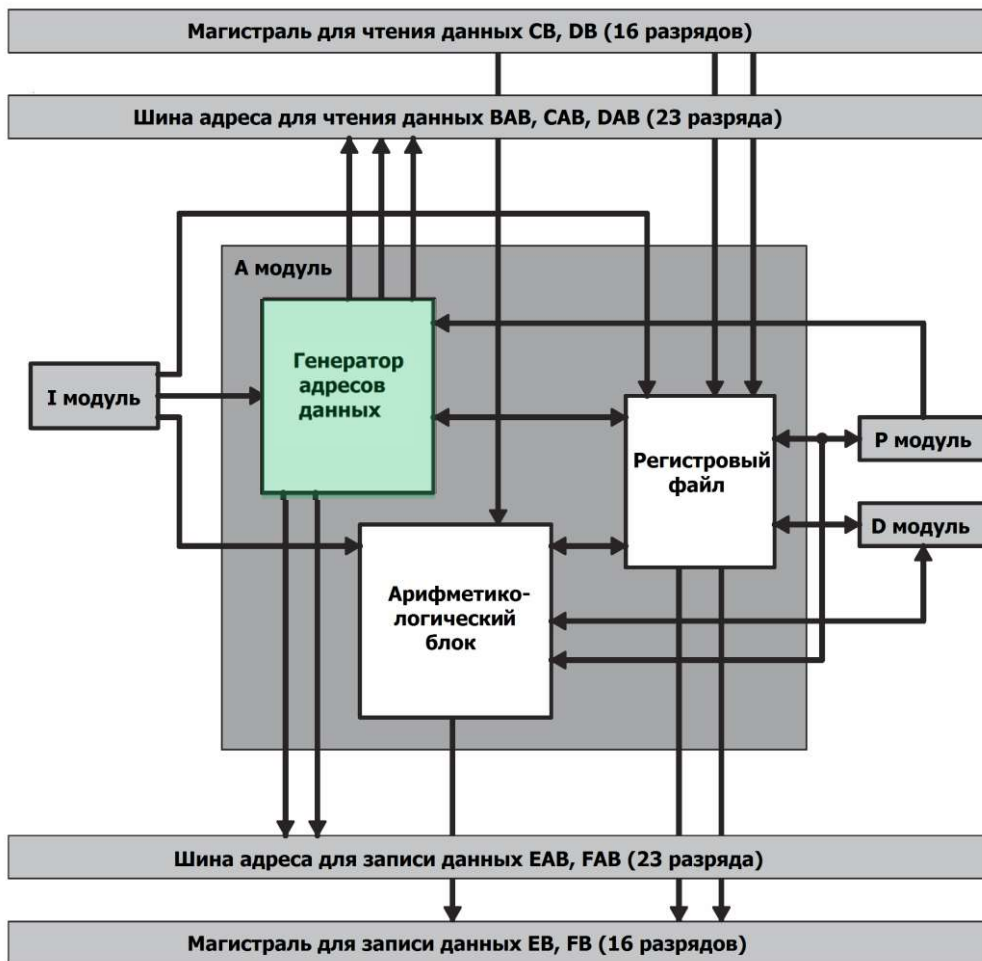
Регистры циклического буфера:

BSA_{xx} (5*16 бит), **BK_{xx}** (3*16 бит).

Временные регистры:

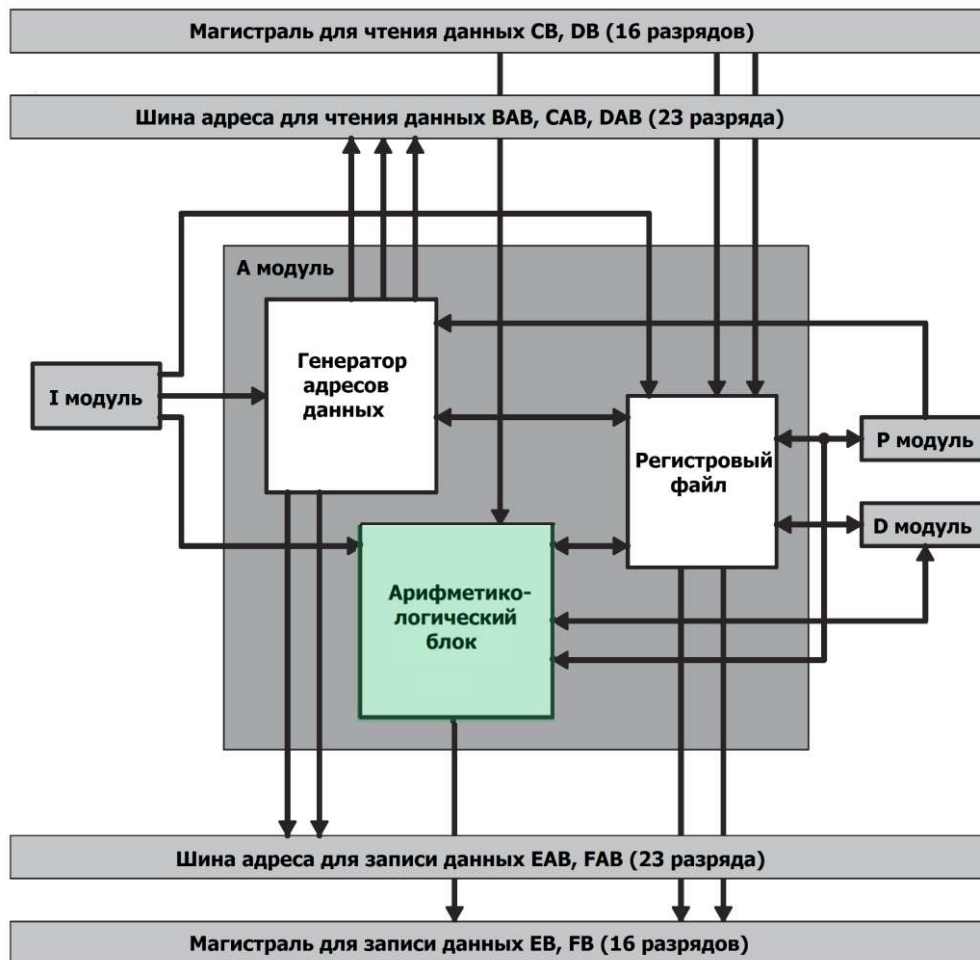
Тх (4*16 бит).

Генератор адресов



- Формирование физических адресов операндов
- Передача физических адресов на шины адреса для чтения и для записи данных

Арифметико-логический блок А



- Сложение, вычитание, сравнение адресов
- Логические и битовые операции с адресами
- Операции арифметических, логических и циклических сдвигов адресов
- Операции модификации регистров в косвенных методах адресации

Регистры модуля А

	22-16	15-0
XAR0	AR0H	AR0
XAR1	AR1H	AR1
XAR2	AR2H	AR2
XAR3	AR3H	AR3
XAR4	AR4H	AR4
XAR5	AR5H	AR5
XAR6	AR6H	AR6
XAR7	AR7H	AR7

	22-16	15-0
XCDP	CDPH	CDP

	22-16	15-0
XDP	DPH	DP

	22-16	15-0
XSP	SPH	SP

XSSP	SPH	SSP
------	-----	-----

	15-9	8-0
	Резерв	PDP

	15-0
T0	
T1	
T2	
T3	

	15-0
BSA01	
BSA23	
BSA45	
BSA67	
BSAC	

	15-0
BK03	
BK47	
BKC	

Регистры страниц данных (23 и 9 бит)

DPH, DP - **D**ata **P**age (**H**igh)

PDP - **P**eripheral **D**ata **P**age

Регистры-указатели (23 бита)

CDPH, CDP - **C**oefficient **D**ata **P**ointer (**H**igh)

SPH, SP, SSP - (**S**ystem) **S**tack **P**ointer (**H**igh)

XAR0–XAR7 - **E**xtended **A**uxiliary **R**egisters

Регистры циклического буфера (16 бит)

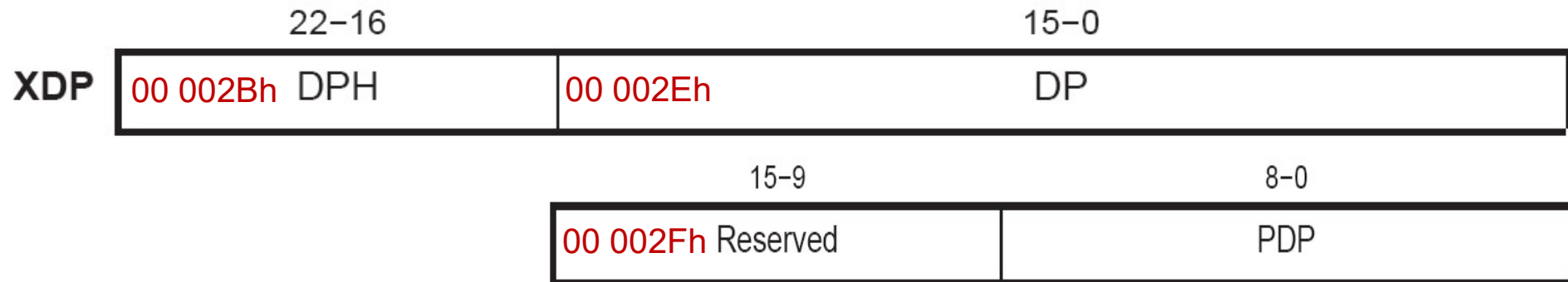
BK03, BK47, BKC - **B**uffer size registers

BSA01, BSA23, BSA45, BSA67, BSAC
- circular **B**uffer **S**tart **A**ddress registers

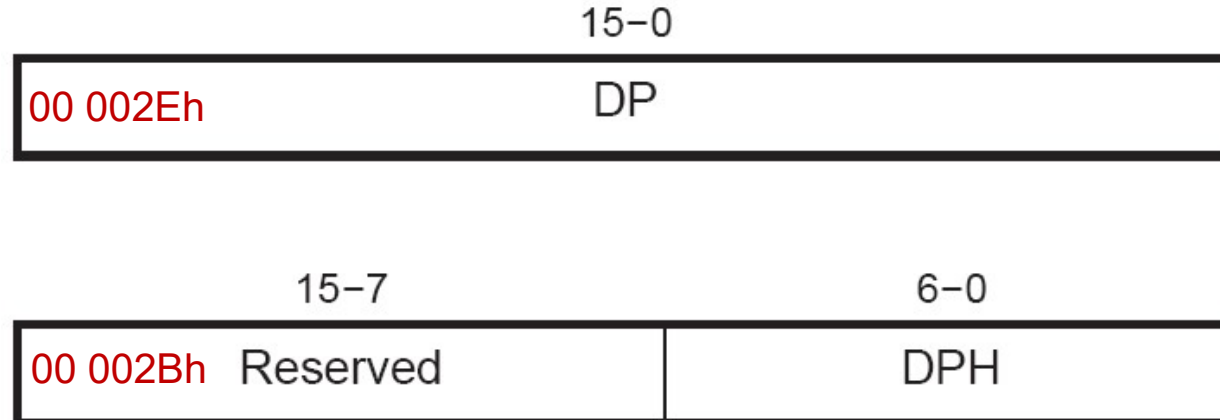
Временные регистры (16 бит)

T0–T3 - **T**emporary registers

Регистры страниц данных



Отображение регистров в адресное пространство памяти



00 002Bh – 23-разрядный адрес 16-разрядного слова данных (BAB, CAB, DAB)

00 002Bh – 0000 0000 0000 0000 0010 1011 – адрес слова

Регистры-указатели

	22-16	15-0
XCDP	00 004Fh CDPH	00 0027h CDP

	22-16	15-0
XAR0	AR0H	00 0010h AR0
XAR1	AR1H	00 0011h AR1
XAR2	AR2H	00 0012h AR2
XAR3	AR3H	00 0013h AR3
XAR4	AR4H	00 0014h AR4
XAR5	AR5H	00 0015h AR5
XAR6	AR6H	00 0016h AR6
XAR7	AR7H	00 0017h AR7

	22-16	15-0
XSP	00 004Eh SPH	00 004Dh SP
XSSP	00 004Eh SPH	00 004Ch SSP

Временные регистры

	15-0
T0	00 0020h
T1	00 0021h
T2	00 0022h
T3	00 0023h

Операционное устройство

39	32 31	16 15	0
G uard	H igh	L ow	

	39-32	31-16	15-0
AC0	000Ah AC0G	00 0009h AC0H	00 0008h AC0L
AC1	000Dh AC1G	00 000Ch AC1H	00 000Bh AC1L
AC2	0026h AC2G	00 0025h AC2H	00 0024h AC2L
AC3	002Ah AC3G	00 0029h AC3H	00 0028h AC3L

	15-0
TRN0	00 000Fh
TRN1	00 0038h

Регистры циклического буфера

15-0

AR0H:, AR1H: BSA01	00 0032h	+ AR0, AR1
AR2H:, AR3H: BSA23	00 0033h	+ AR2, AR3
AR4H:, AR5H: BSA45	00 0034h	+ AR4, AR5
AR6H:, AR7H: BSA67	00 0035h	+ AR6, AR7
CDPH: BSAC	00 0036h	+ CDP

15-0

BK03	00 0019h	AR0 – AR3
BK47	00 0030h	AR4 – AR7
BKC	00 0031h	CDP

```
#define H 131
#define BK 1024
int h[H], x[BK]; y[BK], s;
for(t = 0, t++, t < BK) {
    s = 0;
    for(i = 0; i++; i < H) {
        s += ((long)h[i]*x[(t-i+BK)%BK])>>15;
    }
    y[t] = s;
}
```

$$y(\tau) = \sum_{i=0}^{H-1} h(i) \times x(\tau - i)$$

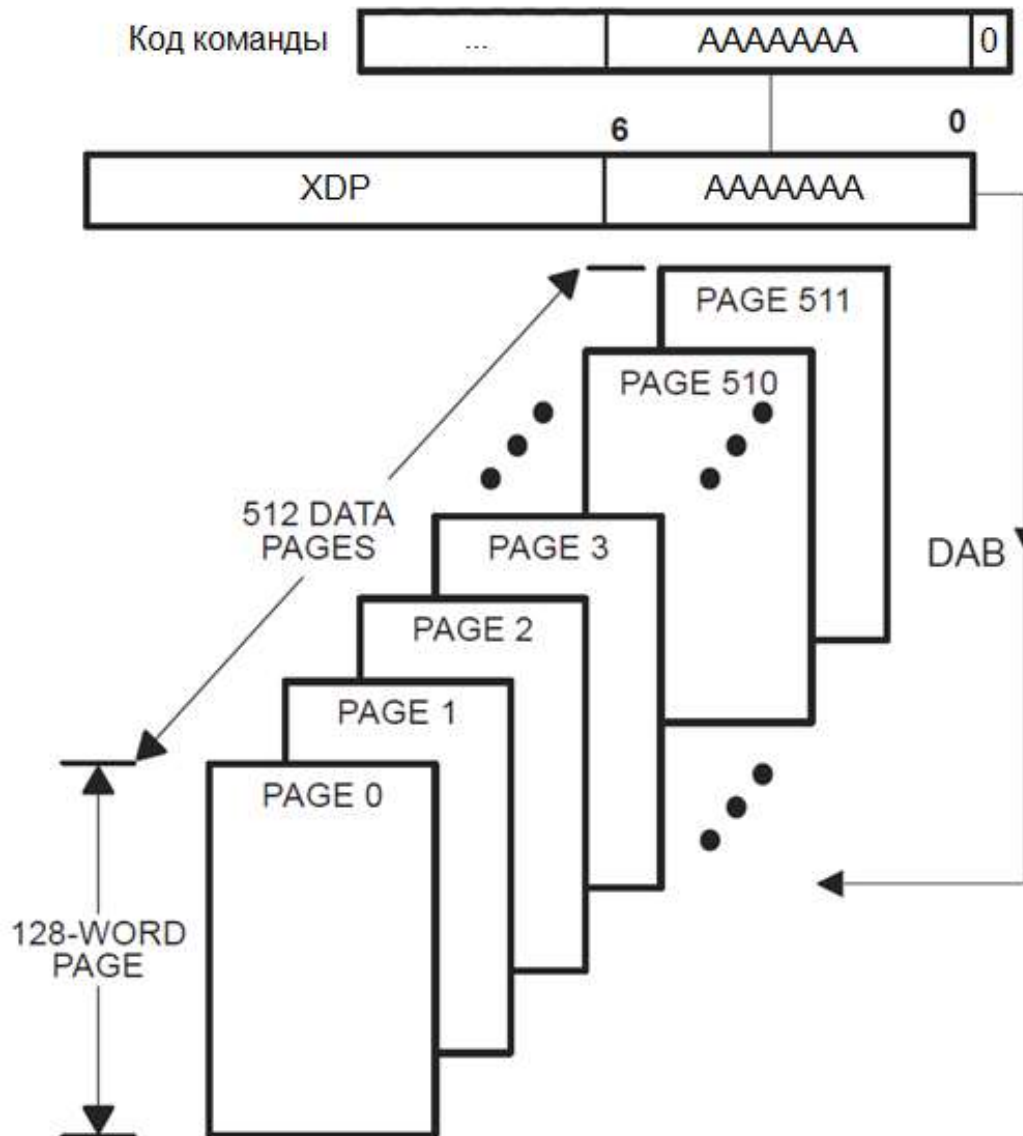


Методы адресации

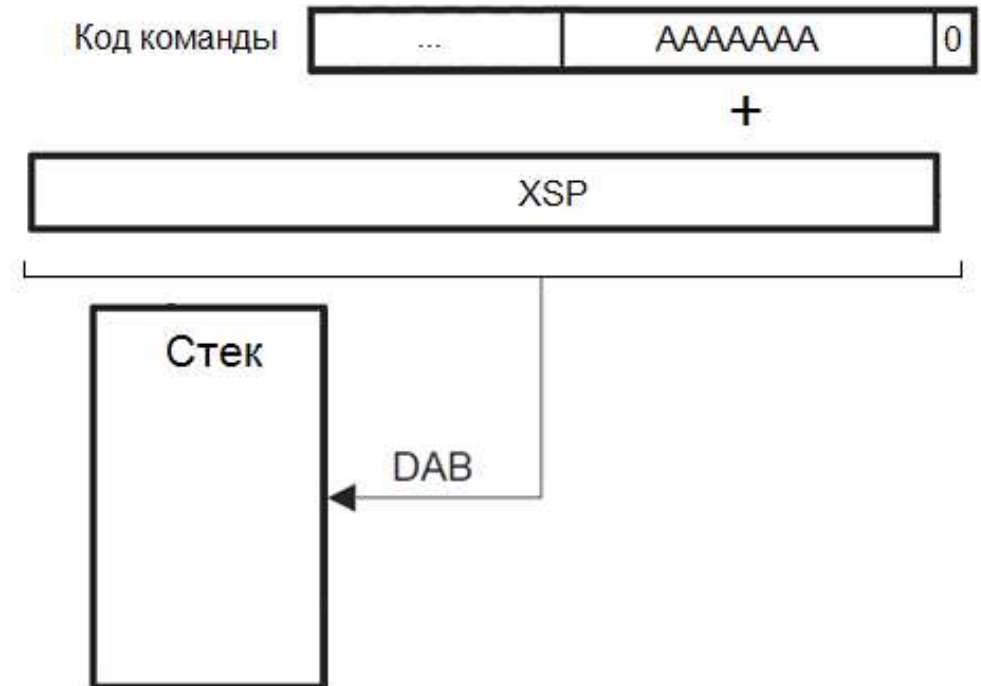
Адресация	Команда	Регистр	Память
Непосредственная	Операнд		
Абсолютная	Адрес		Операнд
Регистровая	Регистр	Операнд	
Прямая	Смещение	Адрес $\rightarrow +$	Операнд
Косвенная	Регистр	Адрес	Операнд
Базовая	Регистр Смещение	Адрес $\rightarrow +$	Операнд
Индексная Бит-реверсивная	Регистр Регистр	Адрес $\rightarrow +$ Смещение	Операнд
Циклическая	Регистр	Размер Смещение $\rightarrow \oplus$ Адрес $\rightarrow +$	Операнд

Прямая адресация

CPL=0



CPL=1



CPL=0: MOV @2, T0 -> MOV *DP(2), T0

CPL=1: MOV @2, T0 -> MOV *SP(2), T0

Косвенная адресация

Операнд	Модификация
*ARx, *CDP	ARx, CDP не изменяется
*ARx+, *CDP+	XARx, CDP увеличивается на 1 (2) ¹ после генерации адреса
*ARx–, *CDP–	XARx, CDP уменьшается на 1 (2) ¹ после генерации адреса
*+ARx	XARx увеличивается на 1 (2) ¹ перед генерацией адреса
*–ARx	XARx уменьшается на 1 (2) ¹ перед генерацией адреса
*(ARx + T0), *(ARx + T0B) ²	XARx увеличивается на T0 после генерации адреса
*(ARx – T0), *(ARx – T0B) ²	XARx уменьшается на T0 после генерации адреса
*ARx(T0)	XARx не изменяется, T0 используется как базовый ³
*(ARx + T1)	XARx увеличивается на T1 после генерации адреса
*(ARx – T1)	XARx уменьшается на T1 после генерации адреса
*ARx(T1)	XARx не изменяется, T1 используется как базовый ³
*ARx(#k16), *CDP(#k16)	XARx, CDP не изменяется, k16 используется как смещение ⁴
*+ARx(#k16), *+CDP(#k16)	XARx, CDP увеличивается на k16 перед генерацией адреса

Примечания:

¹ Изменение регистра на 1 происходит, если операнд слово, на 2 – если двойное слово;

² Режим адресации с обратным порядком бит, при котором 0 бит – старший, а 15 (31, 39) – младший;

³ При базовой адресации адрес операнда равен XARx плюс T0, расширенный со знаком до 23 бит.

⁴ Адрес операнда равен XARx плюс k16, расширенная со знаком до 23 бит.

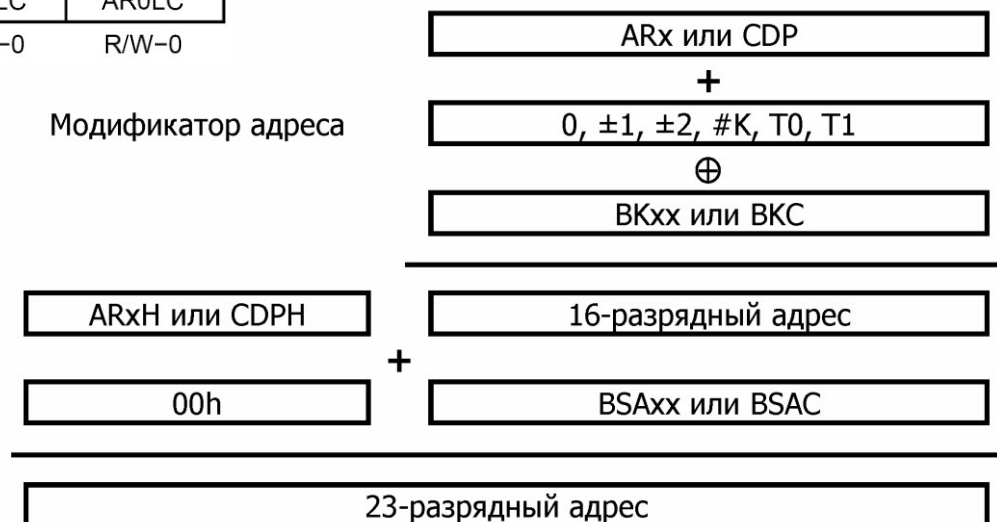
Циклическая адресация

ST2_55 - регистр состояния 2 микропроцессора 00 004Bh

15	14	13	12	11	10	9	8
ARMS	Reserved		DBGM	EALLOW	RDM	Reserved	CDPLC
R/W-0	R-11b		R/W-1	R/W-0	R/W-0	R-0	R/W-0
7	6	5	4	3	2	1	0
AR7LC	AR6LC	AR5LC	AR4LC	AR3LC	AR2LC	AR1LC	AR0LC
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Страница AR1H
Адрес BSA01
Смещение AR1
Текущее слово AC0
Размер BK03

Модификатор адреса

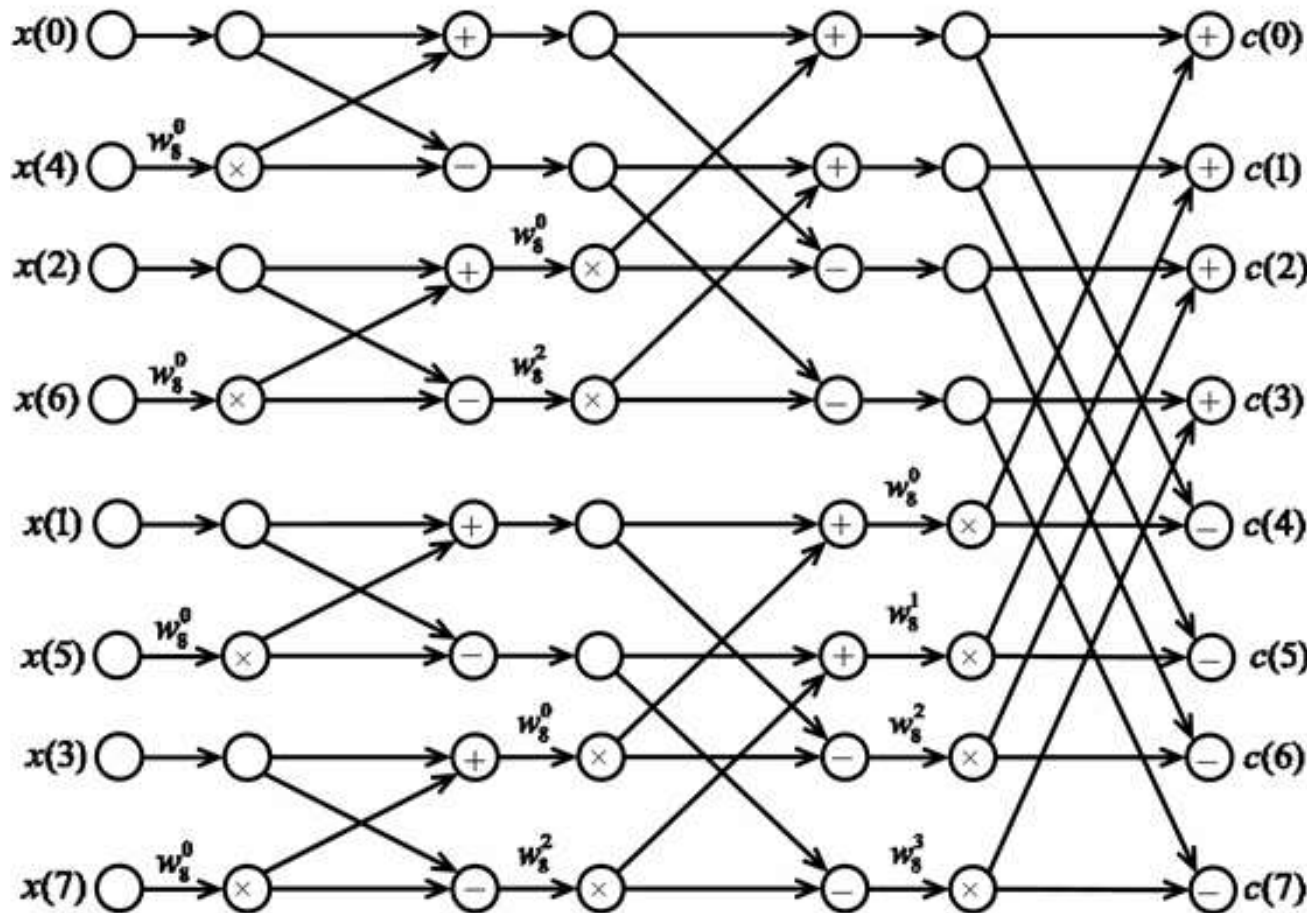


```
MOV #3, BK03
BSET AR1LC
AMOV #010000h, XAR1
MOV #0A02h, BSA01
MOV *AR1+, AC0
MOV *AR1+, AC0
MOV *AR1+, AC0
MOV *AR1+, AC0
```

; Размер циклического буфера в BK03
; Настройка циклической адресации через AR1
; Загрузка номера страницы в AR1H и очистка AR1
; Загрузка стартового адреса буфера в BSA01
; Пересылка в AC0 из 01 0A02 + AR1 = 01 0A02h
; Пересылка в AC0 из 01 0A02 + AR1 = 01 0A03h
; Пересылка в AC0 из 01 0A02 + AR1 = 01 0A04h
; Пересылка в AC0 из 01 0A02 + AR1 = 01 0A02h

Бит-реверсивная адресация

$$c(i) = \sum_{\tau=0}^{N-1} x(\tau) \times \omega_N^i(\tau), \quad \omega_N^i = \exp(-2\pi j i \tau / N), \quad w_N^i = \exp(-2\pi j i / N).$$



$c(*) - x(*)$
 000 – 000 (0)
 001 – 100 (4)
 010 – 010 (2)
 011 – 110 (6)
 100 – 001 (1)
 101 – 101 (5)
 110 – 011 (3)
 111 – 111 (7)

*(ARx -TxB)

*(ARx+TxB)

AR0= **01100000**

T0 = 00000100

01100000 (+0)

+ 00000100

= **01100100 (+4)**

+ 00000100

= **01100010 (+2)**

+ 00000100

= **01100110 (+6)**

+ 00000100

= **01100001 (+1)**

+ 00000100

= **01100101 (+5)**

+ 00000100

= **01100011 (+3)**

+ 00000100

= **01100111 (+7)**

Примечание. Реализована путем сложения (вычитания) с обратным распространением переноса (заема).

Режимы адресации

Базовая		Постфиксная		Префиксная	
Сигнальный режим (ARMS = 0)					
*ARn	PPP0 0001	*ARn+	PPP0 0011	*+ARn	PPP1 1001
*ARn(T0)	PPP0 1011	*ARn-	PPP0 0101	*-ARn	PPP1 1011
*ARn(T1)	PPP1 0111	*(ARn + T0)	PPP0 0111	*+ARn(#K16)	PPP0 1111
*ARn(#K16)	PPP0 1101	*(ARn - T0)	PPP0 1001		
		*(ARn + T0B)	PPP1 1101		
		*(ARn - T0B)	PPP1 1111		
		*(ARn + T1)	PPP1 0011		
		*(ARn - T1)	PPP1 0101		
Режим управления (ARMS = 1)					
*ARn	PPP0 0001	*ARn+	PPP0 0011	*+ARn(#K16)	PPP0 1111
*ARn(T0)	PPP0 1011	*ARn-	PPP0 0101		
*ARn(#K16)	PPP0 1101	*(ARn + T0)	PPP0 0111		
*ARn(short(#k3))	PPP1 xxx1	*(ARn - T0)	PPP0 1001		

ARMS – **AR M**ode **S**witch (переключатель режимов косвенной адресации: ARMS=0 – интенсивная обработка сигнала, ARMS=1 – обычные приложения).

Кодирование адресов

- **Непосредственная:** #k3, ..., #K16, #k16, #k23.
- **Абсолютная:** *abs(#a16) [DPH], *#a23, port(#a16).
- **Регистровая:** ACx, TRNx, Tx, ARx, CDP, ...
- **Прямая:** @a7 [XDP, XSP], port(@a7) [PDP].
- **Косвенная:** *ARx, *CDP, *ARx \pm , *CDP \pm , * \pm ARx.
- **Базовая:** *ARx(k16), *CDP(k16), *+ARx(k16).
- **Индексная:** *ARx(Tx), *(ARx \pm Tx), *+CDP(k16).
- **Циклическая:** [CR.] *ARx \pm , * \pm ARx, *(ARx \pm Tx), *CDP \pm , *+ARx(k16), *+CDP(k16) [BSAxx, BKxx].

Примеры адресации

- **Непосредственная** адресация **#**:
 - константа (k3, k4, k5, k6);
 - байт (K8, k8);
 - слово (k9, k12, k16, K16);
 - полуторное слово (k23).

```
MOV #1Fh, DPH
```
- **Регистровая** адресация **R**.

```
BCLR AR0, AC0
```
- **Абсолютная** адресация ***#**:
 - *abs(#a16);
 - *#a23;
 - port(#a16).

```
PSH *abs16(#1234h)
```
- **Прямая** адресация **@a7**:
 - через DP @a7 (CPL=0);
 - через SP @a7 (CPL=1);
 - через PDP port(@a7);
 - битовая @k5.

```
MOV @#-12, T0  
AND #FFEFh, port(@#1232h)  
BTST @30, AC3
```
- **Косвенная** адресация ***R, *±R, *R±**:
 - не модифицирующая *R;
 - постфиксная *R+, *R-;
 - префиксная *+R, *-R.

```
MOV *CDP+, T2  
BSET *-AR2, AC1
```
- **Базовая** адресация ***+R(k16)**:
 - не модифицирующая *R(k16);
 - модифицирующая *+R(k16).

```
MOV AR3, high_byte(*CDP(#4))  
BNOT *+AR4(#3), AC2
```
- **Индексная** адресация ***(R±R), *R(R)**:
 - знаковая *R(R);
 - беззнаковая *(R±R);
 - реверсивная *(R±RB).

```
MOV *AR1(T1) << #16, AC0  
MAS *(AR6-T0), *CDP, AC2  
BNOT *(AR2+T0B), AC3
```
- **Циклическая** адресация.

```
ADD.CR dual(*CDP+), AC0, AC1  
BSET *(AR7+T0), AR5
```

Мнемоника команд

BTSTCLR k4, Smem, TCx
MOV HI(ACy << T2), Ymem
MOV [uns()]high_byte(Smem)[], dst
MOV high_byte(Smem) << #SHIFTW, ACx
ADD.CR dual(Lmem), ACx, Acy
MPY[R][40] [uns()]Ymem[], [uns()]Cmem[], Acx
MOV [uns()] [rnd()]HI[(saturate)(ACx << Tx)[()]]], Smem

Smem, Lmem, Cmem, Xmem, Ymem – метод адресации операнда в памяти

Baddr – метод адресации бита в регистре

kx (Kx) – непосредственная беззнаковая (знаковая) константа разрядности x

U, uns – беззнаковый операнд

low_byte (high_byte) – младший (старший) байт

#SHIFTW – знаковая константа числа сдвигов

mmap() – память отображения регистров

port() – адресное пространство ввода-вывода

dbl – 32-разрядный операнд в памяти

dual – двойной доступ к памяти

HI (LO) – старшие (младшие) 16 разрядов аккумулятора

pair – регистровая пара

R, rnd – округление, **saturate** - насыщение

<instruction>.**CR** (.**LR**) – принудительная циклическая (линейная) адресация

Кодирование команд

MOV Smem, dst

mov dst Smem
1010 FDDD | AAAA AAAl

AAAA	AAAI	Smem	AAAA	AAAI	Smem
0001	0001	ABS16(#k16)	PPP0	0111	*(ARn + T0)
0011	0001	*(#k23)	PPP0	1001	*(ARn – T0)
0101	0001	port(#k16)	PPP0	1011	*ARn(T0)
0111	0001	*CDP	PPP0	1101	*ARn(#K16)
1001	0001	*CDP+	PPP0	1111	*+ARn(#K16)
1011	0001	*CDP–	PPP1	0011	*(ARn + T1)
1101	0001	*CDP(#K16)	PPP1	0101	*(ARn – T1)
1111	0001	*+CDP(#K16)	PPP1	0111	*ARn(T1)
PPP0	0001	*ARn	PPP1	1001	*+ARn
PPP0	0011	*ARn+	PPP1	1011	*–ARn
PPP0	0101	*ARn–	PPP1	1101	*(ARn + T0B)
			PPP1	1111	*(ARn – T0B)

FSSS FDDD	dst
0000	Accumulator 0 (AC0)
0001	Accumulator 1 (AC1)
0010	Accumulator 2 (AC2)
0011	Accumulator 3 (AC3)
0100	Temporary register 0 (T0)
0101	Temporary register 1 (T1)
0110	Temporary register 2 (T2)
0111	Temporary register 3 (T3)
1000	Auxiliary register 0 (AR0)
1001	Auxiliary register 1 (AR1)
1010	Auxiliary register 2 (AR2)
1011	Auxiliary register 3 (AR3)
1100	Auxiliary register 4 (AR4)
1101	Auxiliary register 5 (AR5)
1110	Auxiliary register 6 (AR6)
1111	Auxiliary register 7 (AR7)

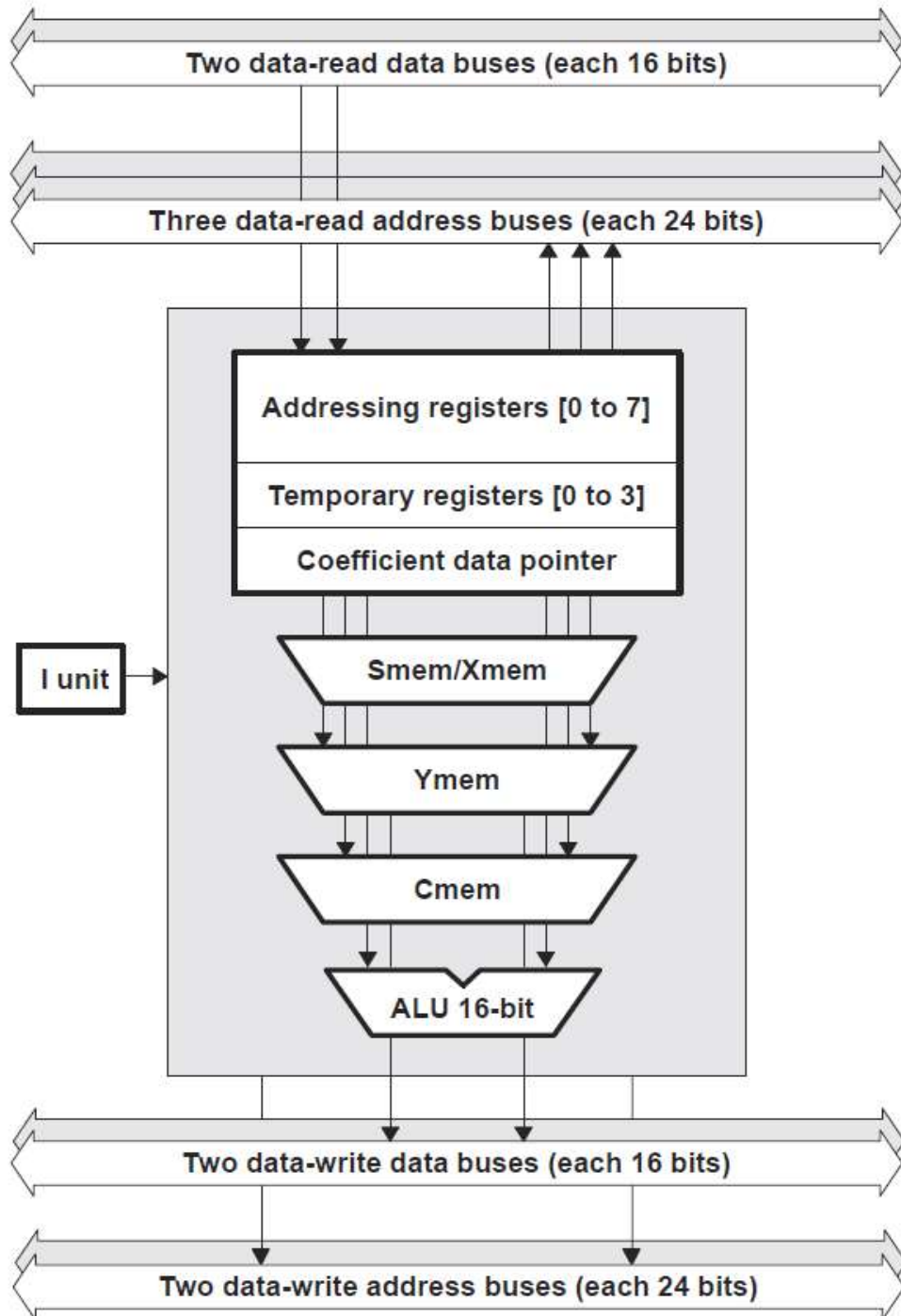
PPP – код дополнительного регистра ARx (000-111).

B – признак бит-реверсивной модификации регистра.

I – флаг методов адресации:

0 – прямая адресация, 1 – не прямая адресация.

Вычисление адресов



Xmem, Ymem – косвенный метод адресации: $*ARx$, $*ARx\pm$, $*(ARx\pm Tx)$, $*ARx(T0)$.

Cmem – косвенная адресация коэффициентов: $*CDP$, $*CDP\pm$, $*(CDP+T0)$.

Smem – прямой или косвенный метод адресации (поле AAAAAAAAAI): $@A...A$, $*abs16(\#k16)$, $*(\#k23)$, $port(\#k16)$, $*CDP$, $*CDP+$, $*CDP-$, $*CDP(\#K16)$, $*+CDP(\#K16)$, $*ARx$, $*ARx\pm$, $*\pm ARx$, $*(ARx\pm Tx)$, $*(ARx\pm T0B)$, $*ARx(T0)$, $*ARn(\#K16)$, $*+ARn(\#K16)$, $*ARn(k8)$.