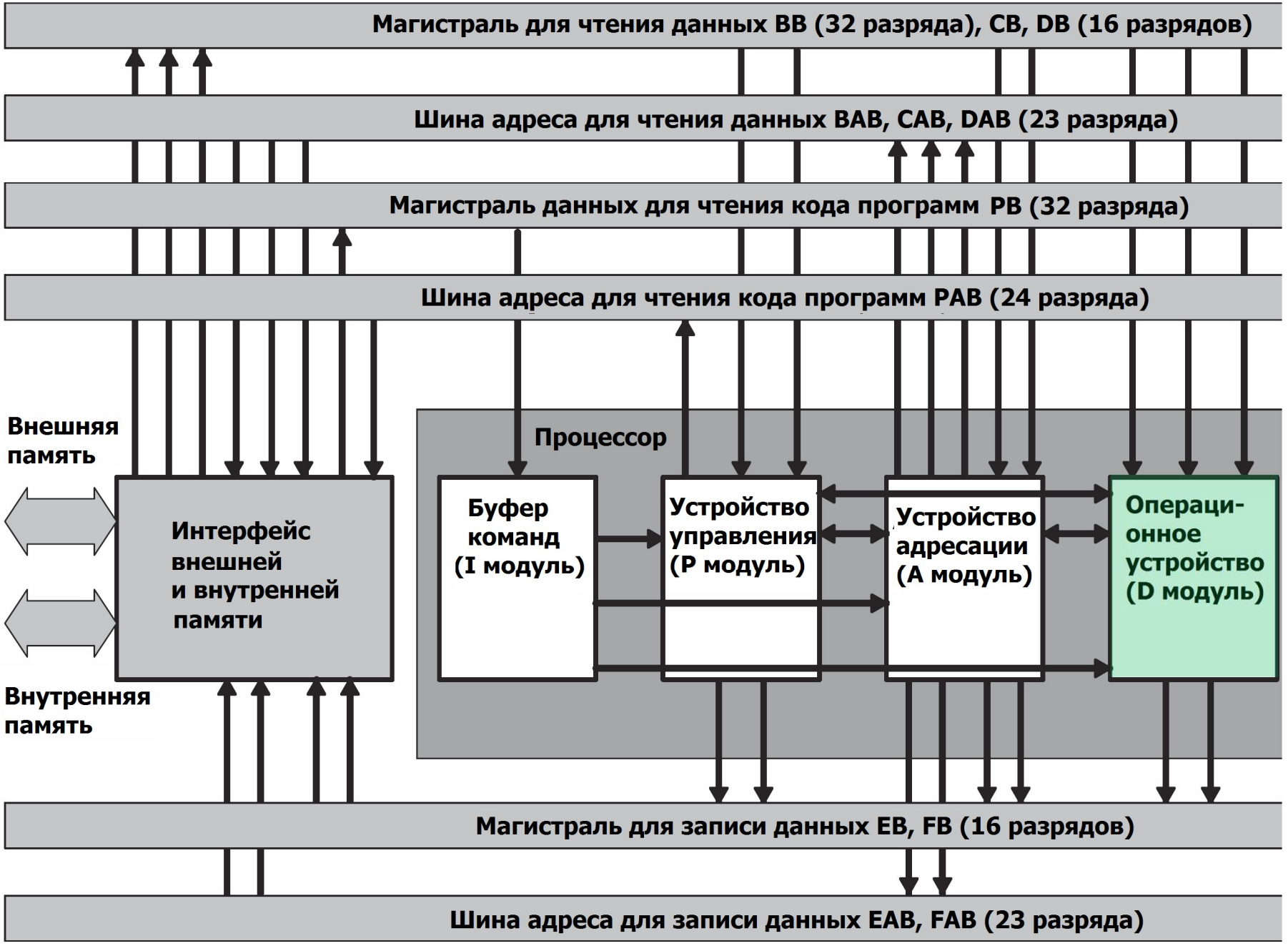
Микропроцессорные устройства обработки сигналов

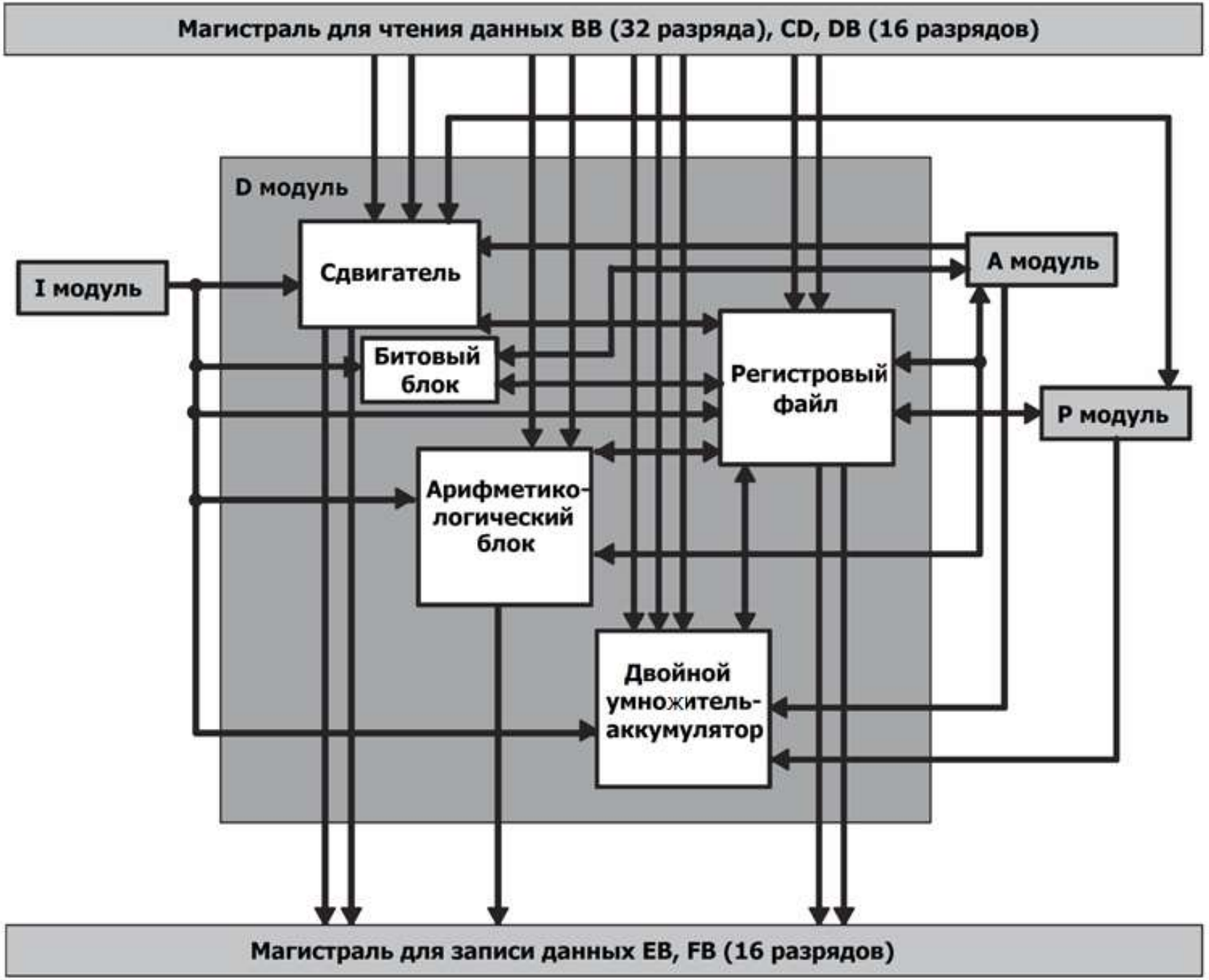
Лекция L06
«Операционное устройство»

http://vykhovanets.ru/course67/

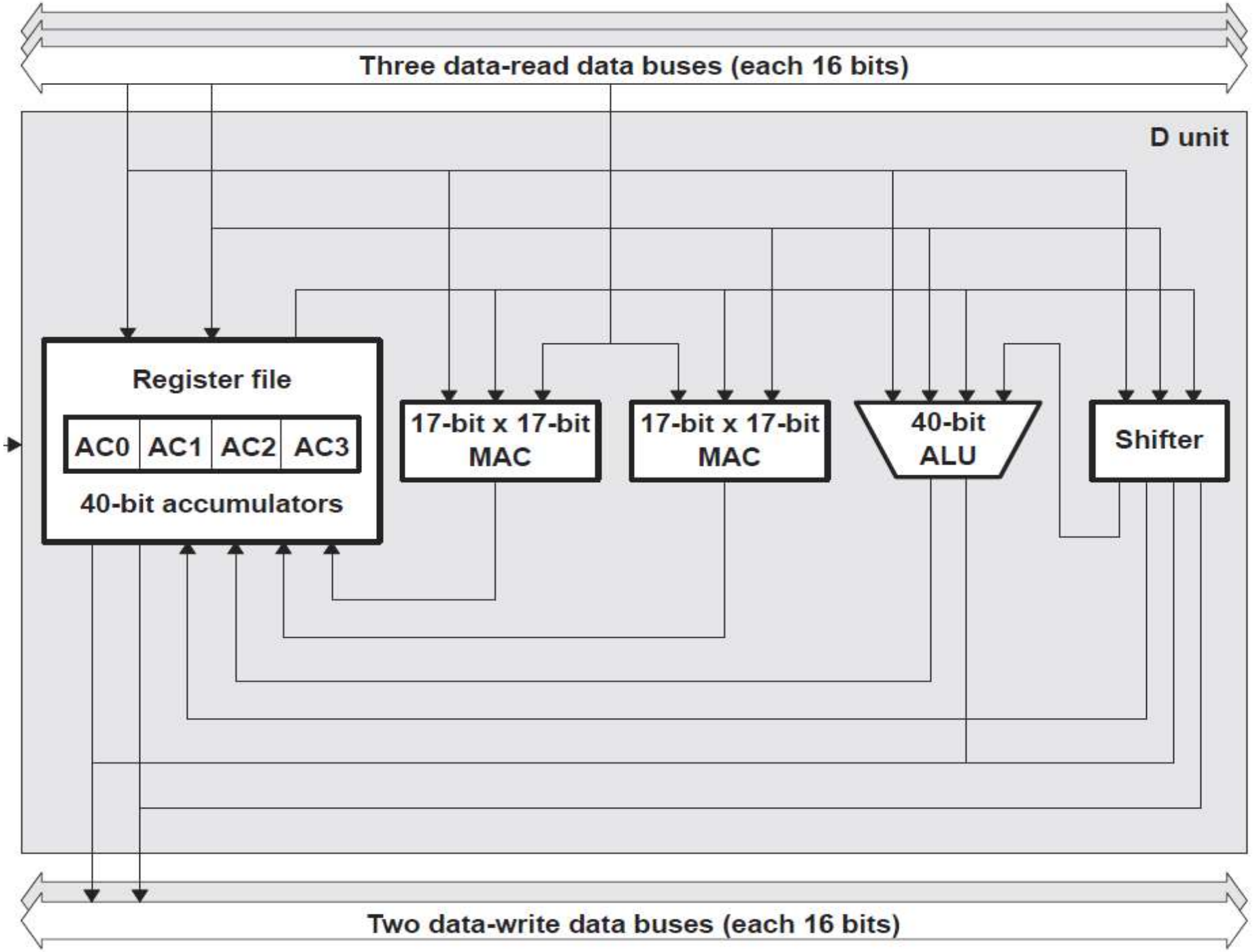
Ядро микропроцессора



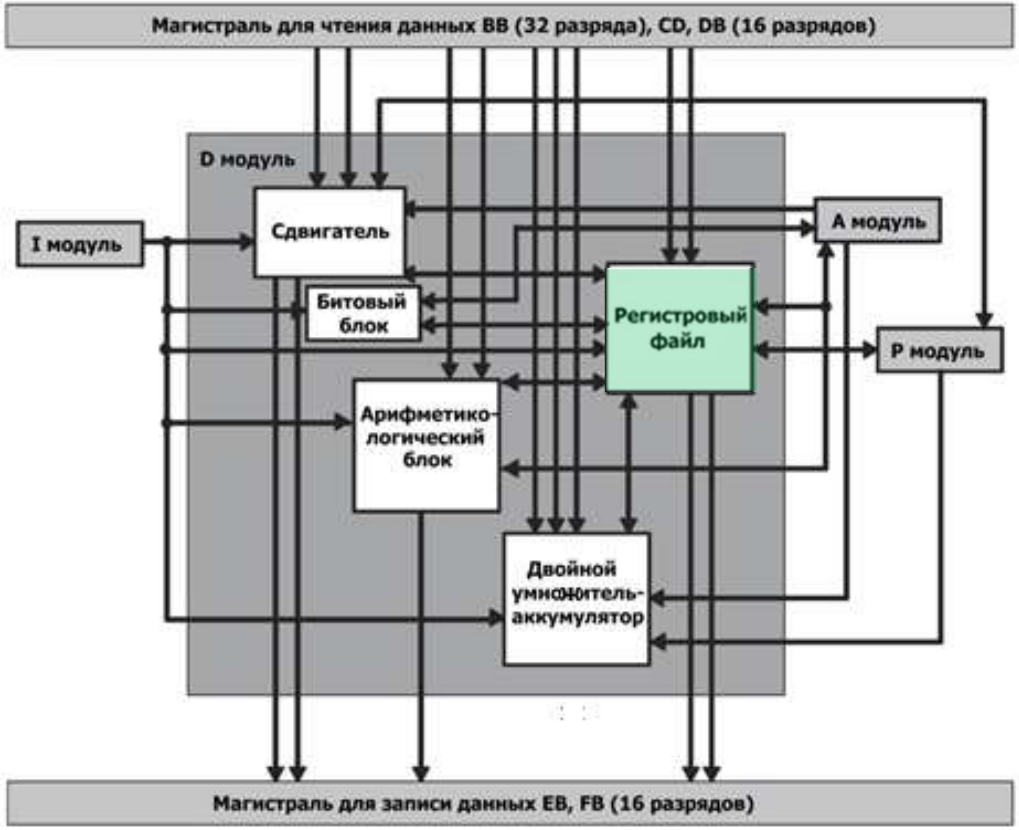
Операционное устройство



Соединения блоков



Регистровый файл D



- Хранение промежуточных данных D модуля
- Двухсторонний обмен данными с блоками D модуля
- Запись непосредственных данных из кода команды I модуля
- Обмен промежуточными данными с модулями I, P, A и памятью

Регистры-аккумуляторы:
AC0-AC3

Переходные регистры:
TRN0, TRN1

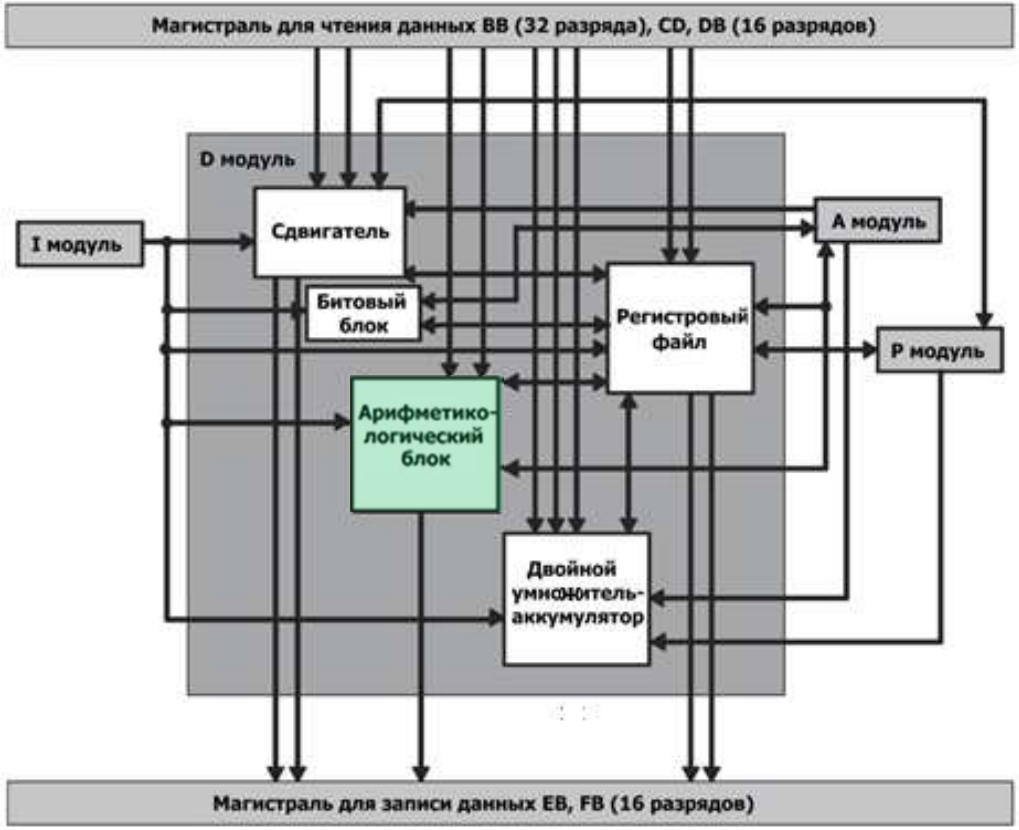
Регистры D модуля

	39–32	31–16	15–0
AC0	AC0G	AC0H	AC0L
AC1	AC1G	AC1H	AC1L
AC2	AC2G	AC2H	AC2L
AC3	AC3G	AC3H	AC3L

	15–0
TRN0	
TRN1	

- Регистры-аккумуляторы AC0, AC1, AC2, AC3 (Accumulators, Low, High, Guard)
- Переходные регистры TRN0, TRN1 (TRaNsition registers)

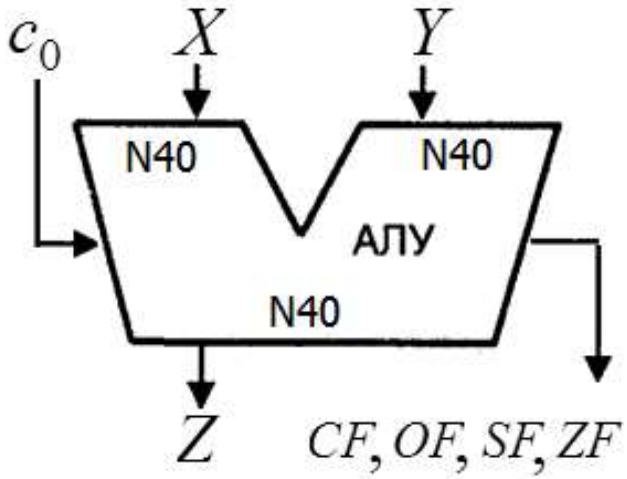
Арифметико-логический блок D



- Получение непосредственных данных от I модуля
- Двухсторонний обмен данными с регистрами модуля A и D, а также с ячейками памяти
- Арифметические операции с данными в формате N40, Z40, Q9.31
- Многоразрядные логические операции

Операции АЛБ

Форматы: N40, Z40, Q9.31

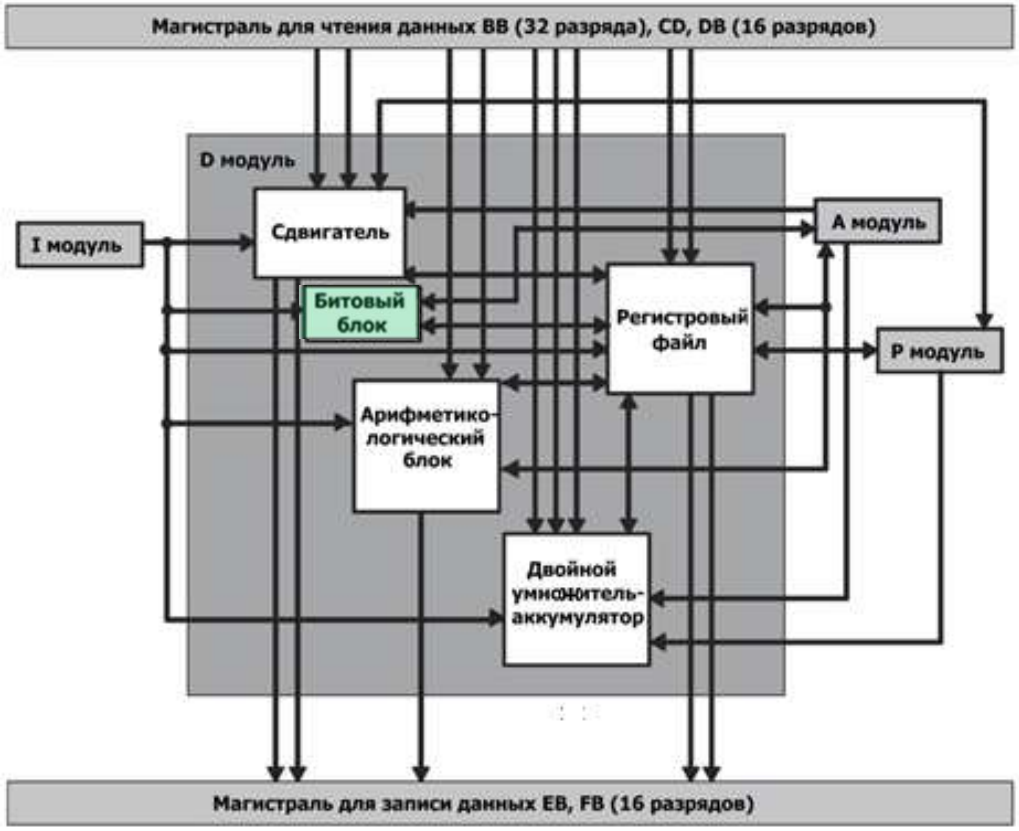


Арифметические:
изменение знака NEG;
абсолютное значение ABS;
сложение ADD;
вычитание SUB;
вычитание и сложение SUBADD;
сложение и вычитание ADDSUB;
сравнение CMP;
сравнение с конъюнкцией CMPAND;
сравнение с дизъюнкцией CMPOR;
максимум MAX, MAXDIFF;
минимум MIN, MINDIFF.

Поразрядные логические:
отрицание – NOT;
конъюнкция – AND;
дизъюнкция – OR;
неэквиваленция – XOR.

Округления:
округление – ROUND;
насыщение – SAT.

Битовый блок D



Адресация бит в регистрах:
Используются те же методы, что и для адресации слов в памяти, например: #3 – номер бита 3, *AR0+ – номер бита в регистре AR0 с увеличением на 1.

- Выполнение операций манипуляции двоичными разрядами (битами): сброс, установка, инверсия, проверка.
- Подсчет числа бит
- Выполнение операций над битовыми полями
- Нормализация чисел в формате с фиксированной запятой

Битовые операции D

Битовые операции:

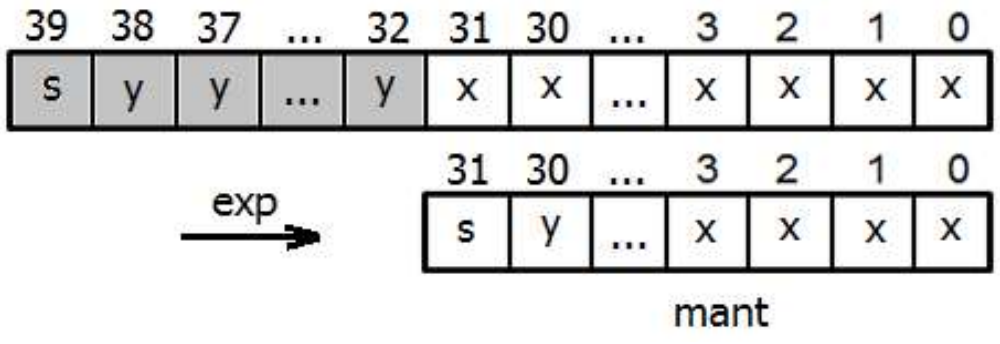
очистка бит – BCLR;
инверсия бит – BNOT;
установка бит – BSET;
проверка бит – BTST;
проверка битовой пары – BTSTP;
установка бит с проверкой – BTSTSET;
очистка бит с проверкой – BTSTCLR;
инверсия бит с проверкой – BTSTNOT.

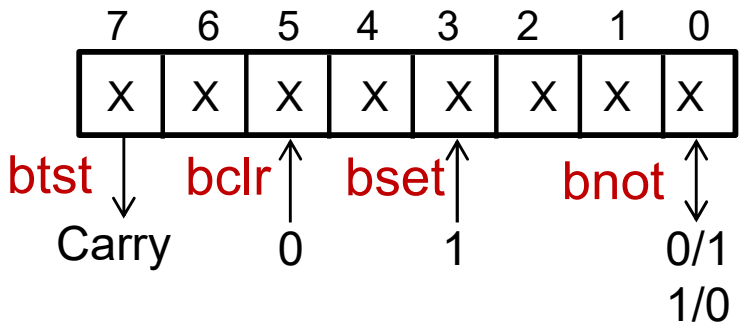
Операции с битовыми полями:

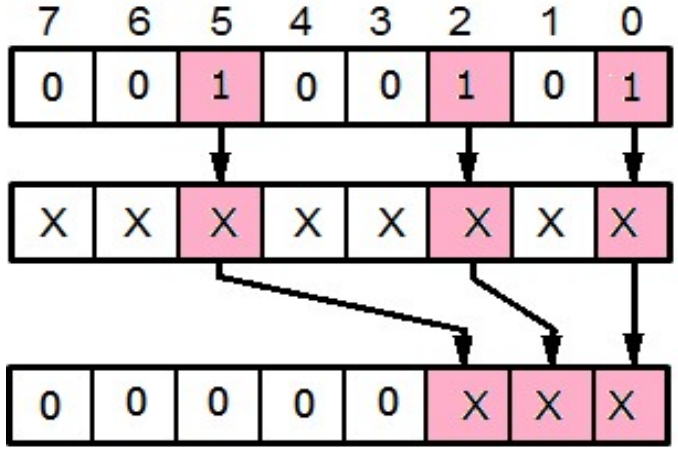
извлечение бит BFXTR;
расширение бит BFXPA;
число бит – BCNT.

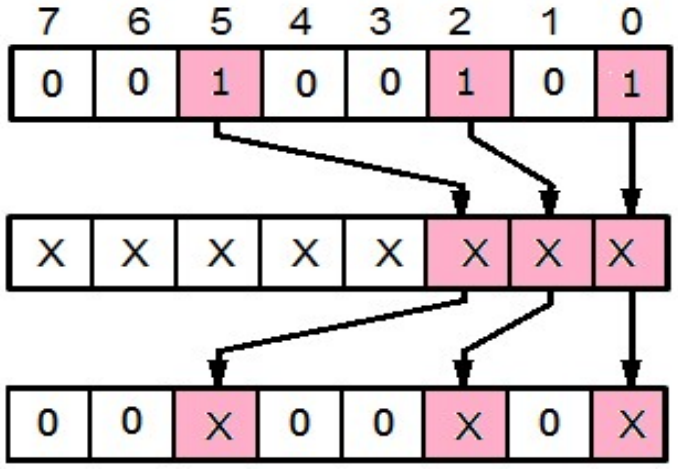
Операции нормализации:

порядок EXP;
мантисса и порядок MANT::NEXP.

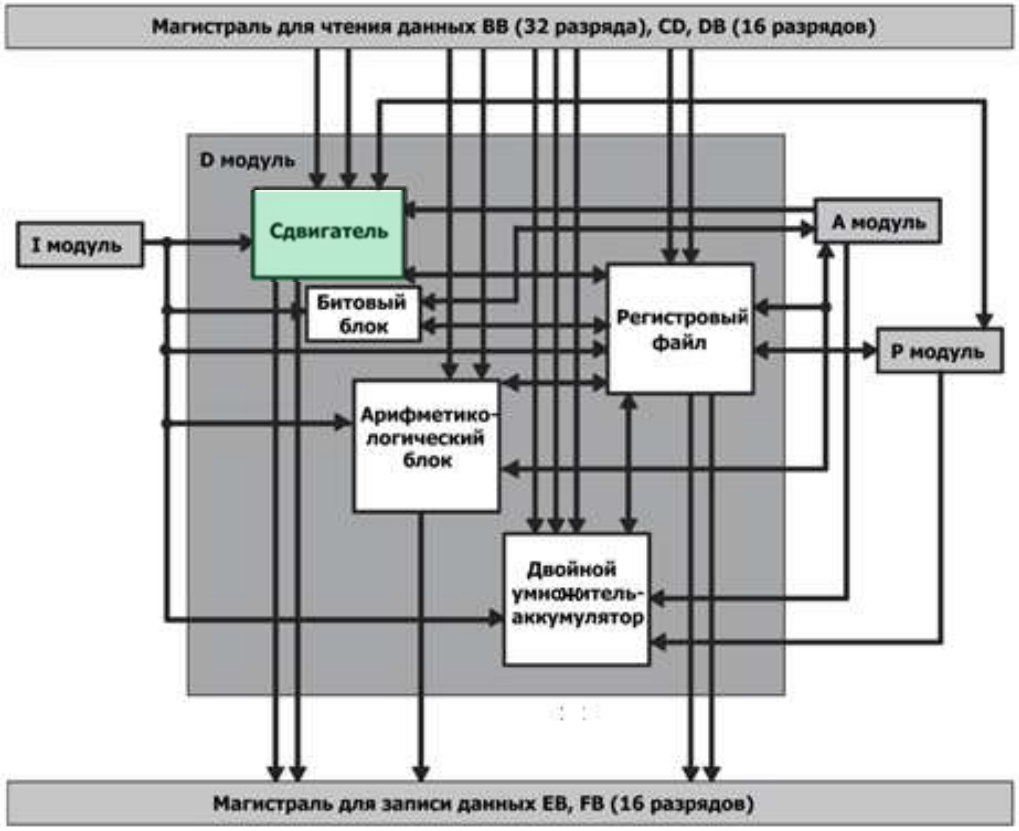








Сдвигатель D



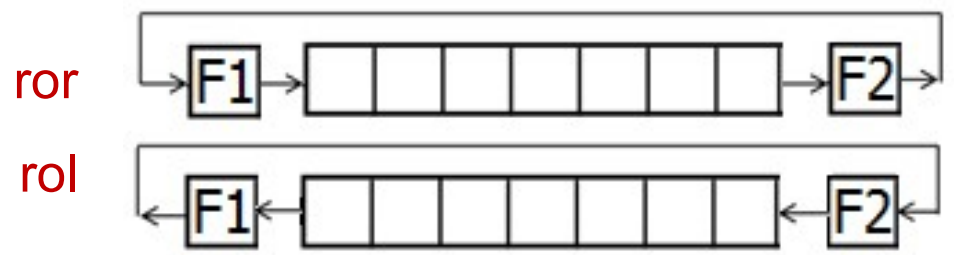
- Арифметические сдвиги 40-разрядных данных в форматов N40, Z40, Q9.31 от -31 до 32 разряда.

- Арифметические сдвиги 16-разрядных данных форматов N16, Z16, Q1.15 от -31 до 32 разряда и сохранение результата в аккумуляторе.

- Циклические сдвиги 16- и 40-разрядных данных в регистрах AC0-AC3 и TRN0, TRN1

Сдвиговые операции D

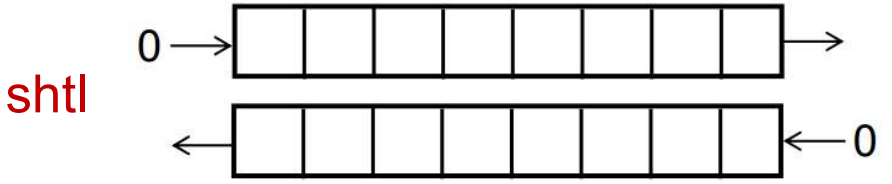
Циклические сдвиги:
влево ROL;
вправо ROR.



F1, F2: CARRY, TC1, TC2

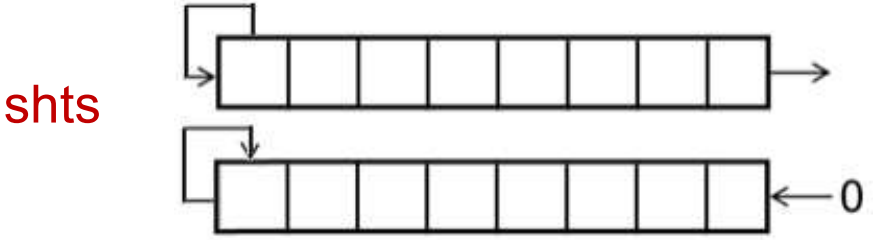
```
unsigned int x, y;
x = (x & 0x1?0x8000:0) | y>>1;
y = (y & 0x8000?1:0) | y<<1;
```

Логические сдвиги:
влево-вправо SFTL.



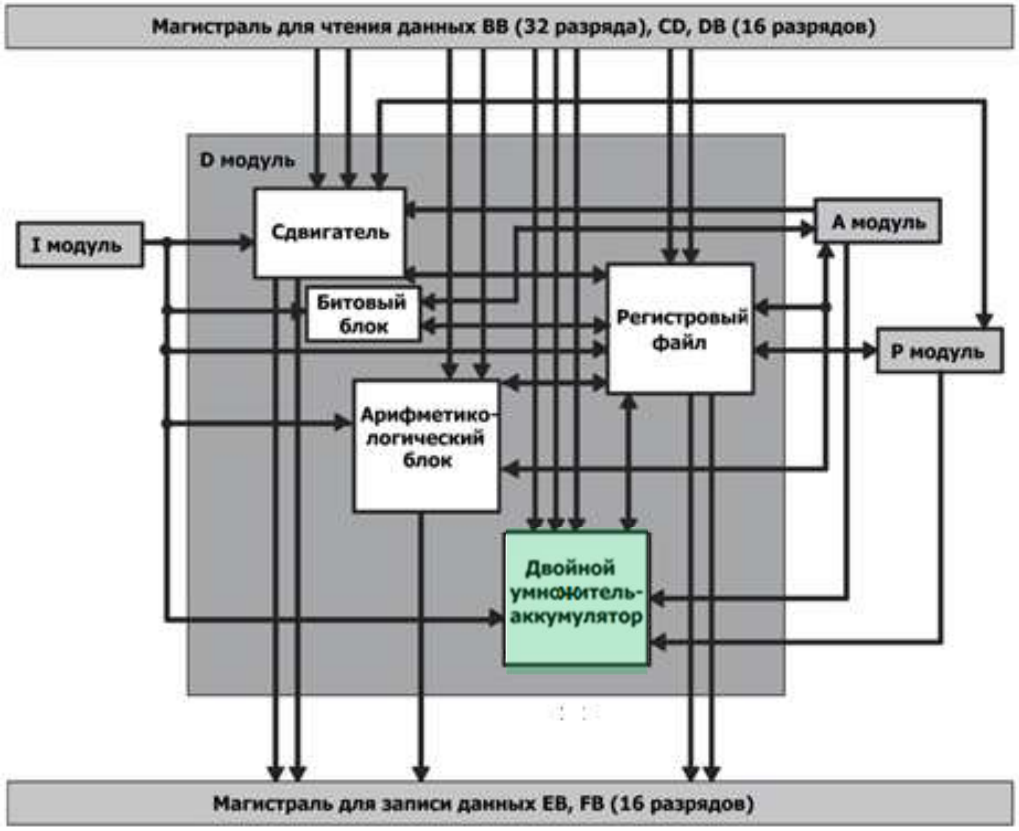

```
unsigned int x, y;
x = x<<3; y = y>>2;
```

Арифметические сдвиги:
влево-вправо SFTS.




```
signed int x, y;
x = y<<3; y = y>>2;
```

Двойной умножитель D

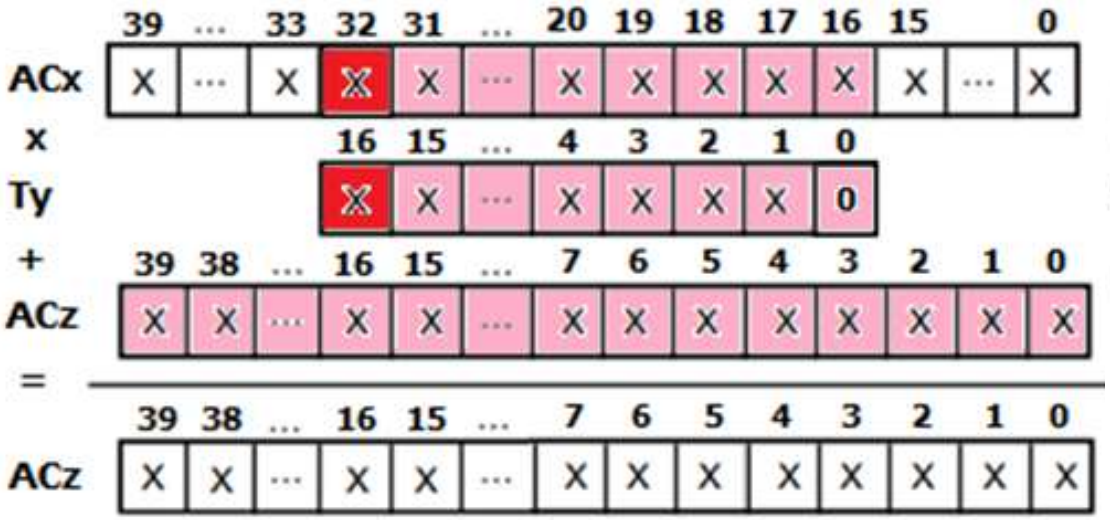


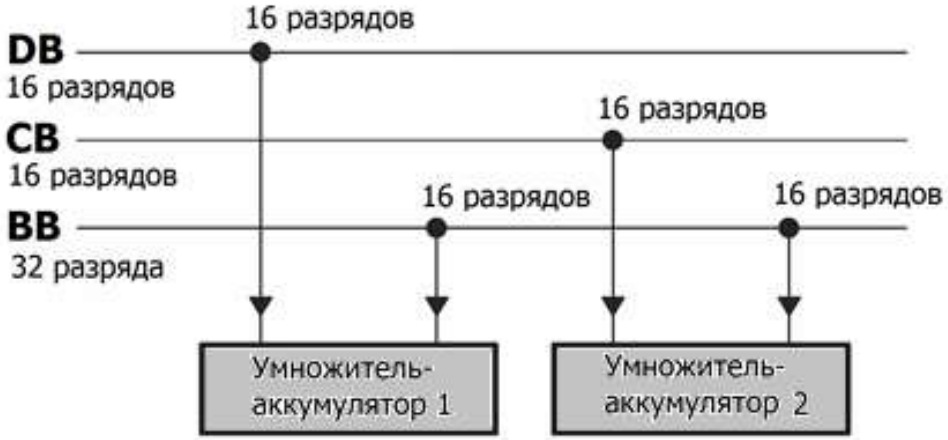
- Предварительное преобразование множителей в 17-разрядный формат (16 разрядов числа и один разряд знака).

- Выполнение одинарных и двойных операций умножения со сложением (вычитанием) целых со знаком (SXMD=1) и без знака (SXMD=0), а также и дробных чисел (SXMD=1, FRC=1).

Примечание. 17-ти разрядный умножитель требуется для умножения целых чисел со знаком (флаг расширения знака SXMD равен нулю).

Умножитель-аккумулятор





ACx, ACz,(40 разрядов): AC0, AC1, AC2, AC3

Ty, Tz (16 разрядов): T0, T1, T2, T3

mac: ACx = ACy + (ACx * Tz)

mac: ACy = (ACy * Tz) + ACx

mack: ACy = ACx + (Tz * k8)

mack: ACy = ACx + (Tz * k16)

Встроенный параллелизм:
MAC::MAC, MAC::MAS, MAC::MPY;
MAS::MAC, MAS::MAS, MAS::MPY;
MPY::MAC, MPY::MAS, MPY::MPY.

- Умножение – **mpy** (multiply)
- Умножение со сложением (вычитанием) – **mac** (**mas**) (multiply and accumulate (subtract))

Подстановочные функции

```
int         _sadd(int, int)
long        _lsadd(long, long)
long long   _llsadd(long long, long long)
int         _ssub(int, int)
long        _lssub(long, long)
long long   _llssub(long long, long long)
int         _smpy(int, int)
long        _lsmpy(int, int)
long        _smac(long, int, int)
long        _smas(long, int, int)
int         _abss(int)
long        _labss(long)
long long   _llabss(long long)
int         _sneg(int)
long        _lsneg(long)
long long   _llsneg(long long);
long        _smpyr(int, int)
long         _smacr(long, int, int)
long        _smasr(long, int, int)
```

```
int     _norm(int)
int     _lnorm(long)
long    _rnd(long)
int     _sshl(int, int)
long    _lsshl(long, int)
int     _shrs(int, int)
long    _lshrs(long, int)
```

```
_sadd:
    BSET    SATA
    ADD     T1, T0
    BCLR    SATA
    RET
```

Среда разработки