

Кодирование микрокоманд

Содержимое различных ROM формируется в виде файлов с расширением .mif на этапе кодирования и загружается в ПЛИС при компиляции проекта в MaxPlus и Quartus.

Файлы создаются в системе микропрограммирования по текстам структурных микрокоманд после отладки функциональных микрокоманд.

Ниже приведены таблицы кодирования микрокоманд по блокам

- коды полей микрокоманд представлены порядковыми номерами 0,1,2,...,7,8, ...

Если 0-код указан, то он обозначает выборку по умолчанию соответствующего источника данных в мультиплексоре.

Если первый код 1, то все коды трактуются как унитарные 1 в соответствующих разрядах поля Unicond.

Блок имя файл кодирования разрядность

=====

Блок 1 16mem Dcm1.mif 7 бит

```
char *Selpc= "Acall,Pc+a,Vect,Call ";
           0   1   2   3                бит 2
char *Unicond16="Wrpc,Wrxda,Incdptr,Incpc ";
           1   2   3   4                бит 4
char *Selaxa ="Dptr, P2wrk ";
           0   1                бит 1
Code.mif программный код -тест
```

блок 3 Bit Dcm3.mif

```
char *Unibit="Bitor,Bitand,Nebit,Newbit,Setbit,Wlocpsw,Selpsw,Clrb ";
           1   2   3   4   5   6   7   8 бит 8
```

блок 4 RALU Dcm4.mif бит 19

```
char *Mop="L,Suba,Subb,Add,Or,Xor,And,H ";
           0 1   2 3   4 5 6 7                бит 3
char *Selacc="BusB,Lalc,Rarc,Xdata,F,Mulow,Quot ";
           0   1   2   3   4   5   6                бит 3
char *SelB="Busc,Mulhigh,Remain ";
           0   1           2                бит 2
char *Uniralu ="Wrb,Wrpb,Wrpa,Ci,Mul,Div,Wrwrk,IncWrk,Count ";
           1   2           3 4 5 6   7   8   9                бит 9
char *Selpa = "Pa,Acc";
           0   1                бит 1
_char *Selpb = "Pb,Pcl";
           0   1                бит 1
```

блок 5 ports -

```
char *Uniport="outp0,outp1,outp2,outp3,cntsp,incsp,altf0,altf2,altf3 ";
           1   2   3   4   5   6   7   8   9
```

блок 6 Interrupt - управление в Control

блок 7 Bus8 Dcm.7.mif бит 17

char *SelbusB=

="Code,Wrk,Pch,Pcl,P0in,P1in,P2in,P3in,Ram,Sp,F,Psw,Acc,Ntcon ";

0 1 2 3 4 5 6 7 8 9 10 11 12 13 бит 4

char *SelbusA ="Riram,Wrk,Asfr,Sp,Abitwrk ";

0 1 2 3 4 бит 3

char *Unibus8 ="Ari,Wram,WramNsfr ";

1 2 3 бит 3

char *Asfr ="Acc ,Sp, B, Psw, Dpl,Dph, P0, P1, P2, P3, Tcon, Ip, Ie" бит 7

адрес 7 бит 0xe0 0x81 0xf0 0xd0 0x82 0x83 0x80 0x90 0xa0 0xb0 0x88

0xb8 0xa8 **адрес 6 бит** 0x60 0x01 0x70 0x50 0x02 0x03 0x00 0x10 0x20

0x30 0x08 0x38 0x28

блок 8 Control Dcm.8.mif бит 5

char *Unicon="Clramk,Ramk1,Wrir,Eintra,Clre,Clrinta ";

1 2 3 4 5 6

ADC.mif декодер 256 кодов команд в 128 адресов микропрограмм/ clk1

Адрес МК ADC[Cop].cnt

ROMM **ROMM.mif** 10 бит

микропрограмма с максимальным кодированием, clk2

Условная микрокоманда ROMM[ADC] - промежуточный код формирует адрес
Мк в декодере

DC по условию if(усл) DC[ROMM], else DC[0]

char *Selif="L,Cc,Zacc,Bitwrk,Wrk7,PswC,Intra,Neiff ";

0 1 2 3 4 5 6 бит 3 + Neiff

char dcmicro; //код общий для всех блоков бит 6

Wrsfr.mif кодирование сигналов записи в Sfr с прямым доступом

"wracc,wrsp,wrb,wrspw,wrdbl,wrdbph,wrp0,wrp1,wrp2,wrp3,wrtcon,wrip,wrie

1 2 3 4 5 6 7 8 9 10 11 12 13

psw acc tcon p0 p1 p2 p3 sp b ip ie dpl dph

adr 0xe0 0x81 0xf0 0xd0 0x82 0x83 0x80 0x90 0xa0 0xb0 0x88 0xb8 0xa8

0x60 0x01 0x70 0x50 0x02 0x03 0x00 0x10 0x20 0x30 0x08 0x38 0x28

VECTOR.mif вектора прерываний

0x03 0x0B 0x13 0x1B 0x23 вектор

1 2 4 8 адрес

Int0 Tf0 Int1 Tf1 T1vR1 прерывание

DCM1-4,7,8,Мсrom формируются для заданных структурных микрокоманд

ADC, Asfr, VECTOR, Wrsfr, Code - различные типы ROM, формируются безусловно

В регистр Sfr записывается новое значение в команде по умолчанию (mov a, #55.) - в микрокоманде блока 7 задается сокращенный адрес регистра **Asfr** и Wram, который передается и распознается на адресной шине BusA. Формируются сигналы записи в теневой регистр Ram[Sfr] и в неявно доступный регистр Sfr,

Регистр Sfr может быть задан адресом в команде (mov Acc, #55, push Acc, ..) – адрес передается на шину BusA и далее интерпретируется. Данные одновременно записываются в Ram[Sfr] и в регистр Sfr с прямым доступом. Таким образом, обеспечивается когерентность значений Sfr в Ram и неявно(прямо) доступных.

Все передачи {Pb, wrk} → busc → {Sfr, Ram}

При чтении значение читается из Ram по адресу на шине BusA с прямым адресом в команде и коммутируется через шину BusB для записи в рабочие регистры Pa, Pb, Wrk.

Все передачи {Code, Xdata, Pcl, Pch} → busb → {Pa, Pb, Wrk}

В регистры Sfr (в частности, Acc, B и тд) передача Ram → Ram всегда двухтактная через промежуточный рабочий регистр.

Прямое обращение при чтении доступно ко всем регистрам Sfr