

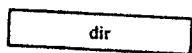
XRL A,dir

Bytes: 2
Cycles: 1
Encoding:

0110	0101
------	------

PSW: P
Operation: $(A) \leftarrow (A) \nabla (dir)$

Exclusive OR logical



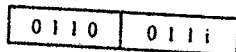
XRL A,@R_i

Bytes: 1
Cycles: 1
Encoding:

0110	011i
------	------

PSW: P
Operation: $(A) \leftarrow (A) \nabla ((R_i))$

Exclusive OR logical



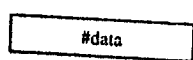
XRL A,#data

Bytes: 2
Cycles: 1
Encoding:

0110	0100
------	------

PSW: P
Operation: $(A) \leftarrow (A) \nabla \#data$

Exclusive OR logical



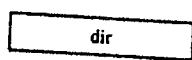
XRL dir,A

Bytes: 2
Cycles: 1
Encoding:

0110	0010
------	------

PSW: —
Operation: $(dir) \leftarrow (dir) \nabla (A)$

Exclusive OR logical



XRL dir,#data

Bytes: 3
Cycles: 2
Encoding:

0110	0011
------	------

PSW: —
Operation: $(dir) \leftarrow (dir) \nabla \#data$

Exclusive OR logical



Приложение E

СИСТЕМА ИНСТРУКЦИИ - РЕЗЮМЕ

Арифметични инструкции

Mnemonic	Description	Code	Bytes	Cycles
ADD A, R_n	Add	28-2F	1	1
ADD A, dir	Add	25	2	1
ADD A, @R_i	Add	26,27	1	1
ADD A, #data	Add	24	2	1
ADDC A, R_n	Add with carry	38-3F	1	1
ADDC A, dir	Add with carry	35	2	1
ADDC A, @R_i	Add with carry	36,37	1	1
ADDC A, #data	Add with carry	34	2	1
SUBB A, R_n	Subtract with borrow	98-9F	1	1
SUBB A, dir	Subtract with borrow	95	2	1
SUBB A, @R_i	Subtract with borrow	96,97	1	1
SUBB A, #data	Subtract with borrow	94	2	1
INC A	Increment	04	1	1
INC R_n	Increment	08-0F	1	1
INC dir	Increment	05	2	1

Mnemonic	Description	Code	Bytes	Cycles
INC GR_i	Increment	06,07	1	1
DEC A	Decrement	14	1	1
DEC R_n	Decrement	18-1F	1	1
DEC dir	Decrement	15	2	1
DEC GR_i	Decrement	16,17	1	1
INC DPTR	Increment	A3	1	2
MUL AB	Multiply	A4	1	4
DIV AB	Divide	84	1	4
DA A	Decimal adjust accumulator	D4	1	1

Mnemonic	Description	Code	Bytes	Cycles
XRL dir, A	Exclusive OR logical	62	2	1
XRL dir, #data	Exclusive OR logical	63	3	2
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate left	23	1	1
RLC A	Rotate left through carry	33	1	1
RR A	Rotate right	03	1	1
RRC A	Rotate right through carry	13	1	1
SWAP A	Swap nibbles	C4	1	1

Логически инструкции

Mnemonic	Description	Code	Bytes	Cycles
ANL A, R_n	AND logical	58-5F	1	1
ANL A, dir	AND logical	55	2	1
ANL A, GR_i	AND logical	56,57	1	1
ANL A, #data	AND logical	54	2	1
ANL dir, A	AND logical	52	2	1
ANL dir, #data	AND logical	53	3	2
ORL A, R_n	OR logical	48-4F	1	1
ORL A, dir	OR logical	45	2	1
ORL A, GR_i	OR logical	46,47	1	1
ORL A, #data	OR logical	44	2	1
ORL dir, A	OR logical	42	2	1
ORL dir, #data	OR logical	43	3	2
XRL A, R_n	Exclusive OR logical	68-6F	1	1
XRL A, dir	Exclusive OR logical	65	2	1
XRL A, GR_i	Exclusive OR logical	66,67	1	1
XRL A, #data	Exclusive OR logical	64	2	1

Инструкции за обмен на данни

Mnemonic	Description	Code	Bytes	Cycles
MOV A, R_n	Move	E8-EF	1	1
MOV A, dir	Move	E5	2	1
MOV A, GR_i	Move	E6,E7	1	1
MOV A, #data	Move	74	2	1
MOV R_n , A	Move	F8-FF	1	1
MOV R_n , dir	Move	A8-AF	2	2
MOV R_n , #data	Move	78-7F	2	1
MOV dir, A	Move	F5	2	1
MOV dir, R_n	Move	88-8F	2	2
MOV dir, dir	Move	85	3	2
MOV dir, GR_i	Move	86,87	2	2
MOV dir, #data	Move	75	3	2
MOV GR_i , A	Move	F6,F7	1	1
MOV GR_i , dir	Move	A6,A7	2	2
MOV GR_i , #data	Move	76,77	2	1

Mnemonic	Description	Code	Bytes	Cycles
MOV DPTR, #data16	Move	90	3	2
MOVC A, @A+DPTR	Move code	93	1	2
MOVC A, @A+PC	Move code	83	1	2
MOVX A, @R _i	Move external	E2, E3	1	2
MOVX A, @DPTR	Move external	E0	1	2
MOVX R _i , A	Move external	F2, F3	1	2
MOVX @DPTR, A	Move external	F0	1	2
PUSH dir	Push onto stack	C0	2	2
POP dir	Pop from stack	D0	2	2
XCH A, R _n	Exchange	C8-CF	1	1
XCH A, dir	Exchange	C5	2	1
XCH A, @R _i	Exchange	C6, C7	1	1
XCHD A, @R _i	Exchange digit	D6, D7	1	1

Инструкции за управление на програмата

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute call	♦	2	2
LCALL addr16	Long call	12	3	2
RET	Return from subroutine	22	1	2
RETI	Return from interrupt	32	1	2
AJMP addr11	Absolute jump	♦♦	2	2
LJMP addr16	Long jump	02	3	2
SJMP rel	Short jump	80	2	2
JMP @A+DPTR	Jump	73	1	2
JZ rel	Jump if accumulator is zero	60	2	2
JNZ rel	Jump if accumulator is not zero	70	2	2
JC rel	Jump if carry	40	2	2
JNC rel	Jump if no carry	50	2	2
JB bit, rel	Jump if bit	20	3	2
JNB bit, rel	Jump if no bit	30	3	2
JBC bit, rel	Jump if bit and clear	10	3	2
CJNE A, dir, rel	Compare and jump if not equal	B5	3	2
CJNE A, #data, rel	Compare and jump if not equal	B4	3	2
CJNE R _n , #data, rel	Compare and jump if not equal	B8-BF	3	2
CJNE @R _i , #data, rel	Compare and jump if not equal	B6, B7	3	2
DJNZ R _n , rel	Decrement and jump if not zero	D8-DF	2	2
DJNZ dir, rel	Decrement and jump if not zero	D5	3	2
NOF	No operation	00	1	1

♦ 11, 31, 51, 71, 91, B1, D1, F1

♦♦ 01, 21, 41, 61, 81, A1, C1, E1

Инструкции за работа с битове

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry	C3	1	1
CLR bit	Clear bit	C2	2	1
SETB C	Set carry	D3	1	1
SETB bit	Set bit	D2	2	1
CPL C	Complement carry	B3	1	1
CPL bit	Complement bit	B2	2	1
ANL C, bit	AND logical	82	2	2
ANL C, /bit	AND logical	B0	2	2
ORL C, bit	OR logical	72	2	2
ORL C, /bit	OR logical	A0	2	2
MOV C, bit	Move carry	A2	2	1
MOV bit, C	Move bit	92	2	2

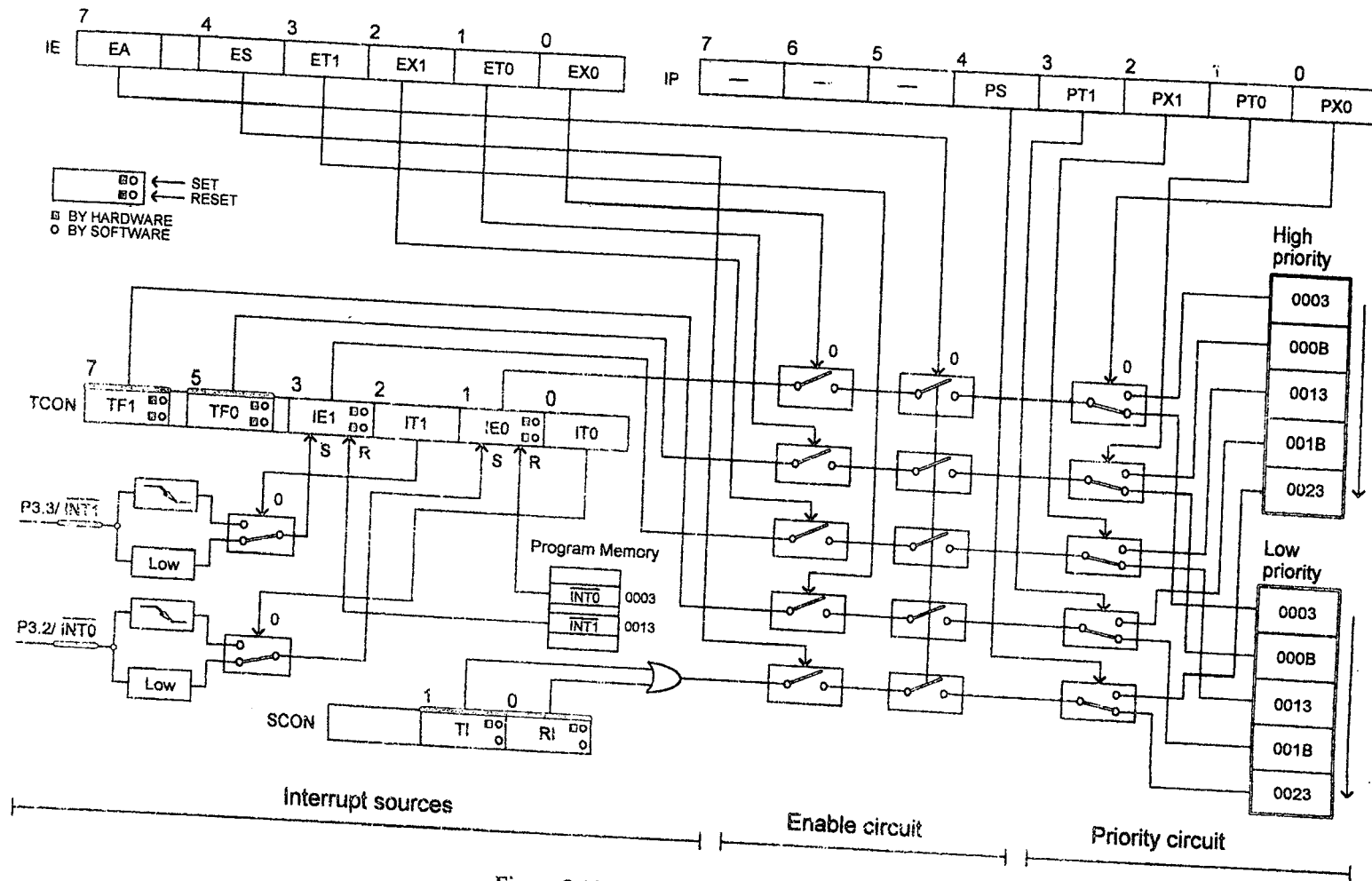
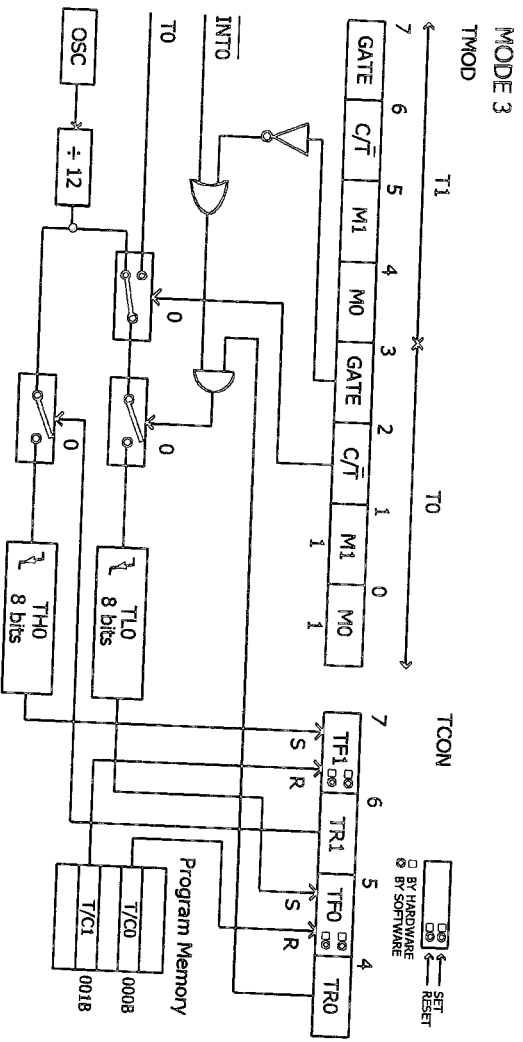
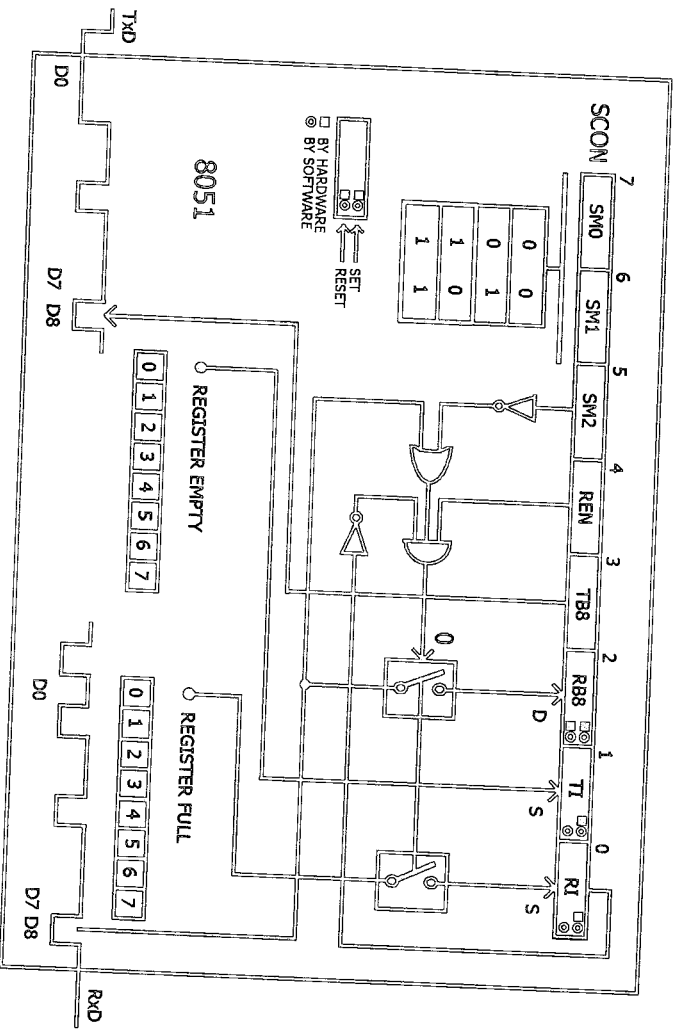


Figure 2.19 The 8051 interrupt system.



Фиг. 13.3



Фиг. 13.4