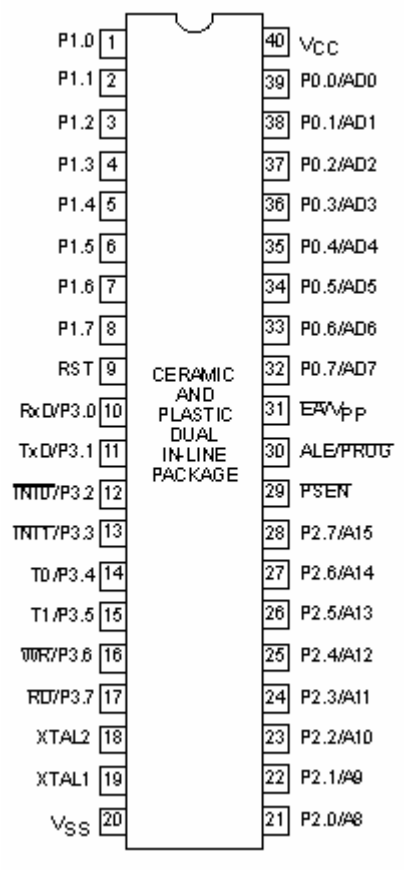
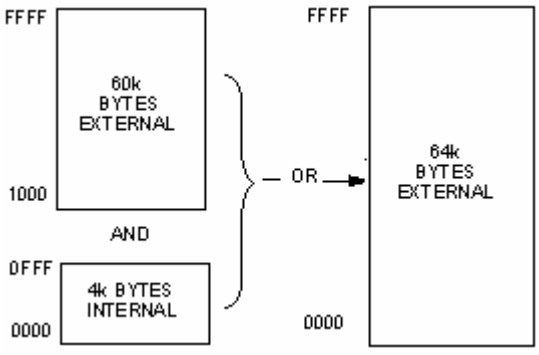
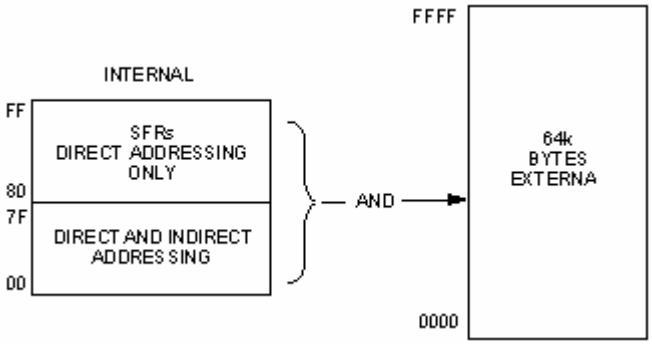
Piedinatura



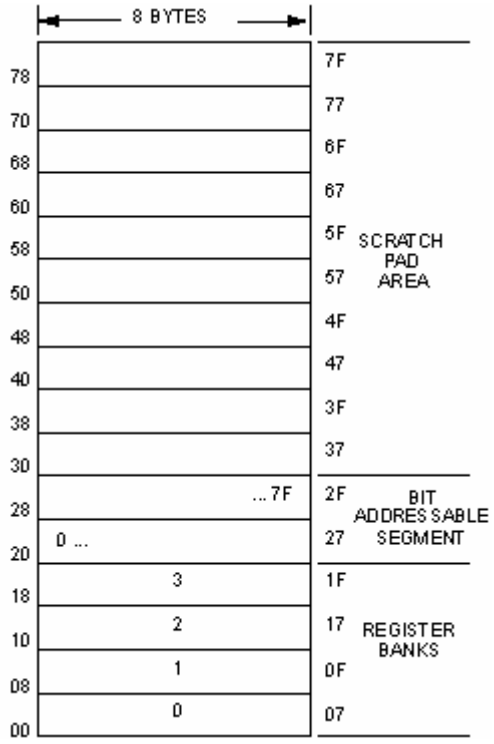
Memoria programma



Memoria dati



Memoria dati



Registri (SFR)

Simbolo	Nome	Indirizzo
ACC	Accumulatore	(*) OE0H
B	Registro B	(*) OF0H
PSW	Registro di stato (FLAG)	(*) OD0H
SP	Stack pointer	81H
DPTR	Puntatore dati (2 byte)	
DPL	parte bassa	82H
DPH	parte alta	83H
PO	Porta O	(*) 80H
P1	Porta 1	(*) 90H
P2	Porta 2	(*) 0A0H
P3	Porta 3	(*) OB0H
IP	Registro priorità interrupt	(*) OB8H
IE	Registro abilitazione interrupt	(*) OA8H
TMOD	Modo contatore/timer	89H
TCON	Controllo contatore/timer	(*) 88H
THO	Contatore/timer 0 byte alto	8CH
TLO	Contatore/timer 0 byte basso	8AH
THI	Contatore/timer 1 byte alto	8DH
TL1	Contatore/timer 1 byte basso	8BH
SCON	Controllo porta seriale	(*) 98H
SBUF	Buffer dati porta seriale	99H
PCON	Controllo alimentazione	97H

Arithmetic Operations

Mnemonic	Description	Osc. Period
ADD A,Rn	Add register to Accumulator	12
ADD A,direct	Add direct byte to Accumulator	12
ADD A,@Ri	Add indirect RAM to Accumulator	12
ADD A,#data	Add immediate data to Accumulator	12
ADDC A,Rn	Add register to Acc. with Carry	12
ADDC A,direct	Add direct byte to Acc. with Carry	12
ADDC A,@Ri	Add indirect RAM to Acc. with Carry	12
ADDC A,#data	Add immediate data to Acc. / Carry	12
SUBB A,Rn	Subtract reg. from Acc. with borrow	12
SUBB A,direct	Sub. direct byte from Acc. / borrow	12
SUBB A,@Ri	Sub. indirect RAM from Acc./ borrow	12
SUBB A,#data	Sub. imm. data from Acc. / borrow	12
INC A	Increment Accumulator	12
INC Rn	Increment register	12
INC direct	Increment direct byte	12
INC @Ri	Increment indirect RAM	12
DEC A	Decrement Accumulator	12
DEC Rn	Decrement register	12
DEC direct	Decrement direct byte	12
DEC @Ri	Decrement indirect RAM	12
INC DPTR	Increment Data Pointer	24
MUL AB	Multiply A and B	48
DIV AB	Divide A by B	48
DA A	Decimal adjust Accumulator	12

Logical Operations

Mnemonic	Description	Osc. Period
ANL A,Rn	AND register to Accumulator	12
ANL A,direct	AND direct byte to Accumulator	12
ANL A,@Ri	AND indirect RAM to Accumulator	12
ANL A,#data	AND immediate data to Accumulator	12
ANL direct,A	AND Accumulator to direct byte	12
ANL direct,#data	AND immediate data to direct byte	24
ORL A,Rn	OR register to Accumulator	12
ORL A,direct	OR direct byte to Accumulator	12
ORL A,@Ri	OR indirect RAM to Accumulator	12
ORL A,#data	OR immediate data to Accumulator	12
ORL direct,A	OR Accumulator to direct byte	12
ORL direct,#data	OR immediate data to direct byte	24
XRL A,Rn	Exc-OR register to Accumulator	12
XRL A,direct	Exc-OR direct byte to Accumulator	24
XRL A,@Ri	Exc-OR indirect RAM to Accumulator	12
XRL A,#data	Exc-OR immediate data to Acc.	12
XRL direct,A	Exc-OR Accumulator to direct byte	12
XRL direct,#data	Exc-OR imm. data to direct byte	24
CLR A	Clear Accumulator	12
CPL A	Complement Accumulator	12
RL A	Rotate Accumulator left	12
RLC A	Rotate Acc. left through Carry	12
RR A	Rotate Accumulator right	12
RRC A	Rotate Acc. right through Carry	12
SWAP A	Swap nibbles within the Accumulator	12

Data Transfer

Mnemonic	Description	Osc. Period
MOV A,Rn	Move register to Accumulator	12
MOV A,direct	Move direct byte to Accumulator	12
MOV A,@Ri	Move indirect RAM to Accumulator	12
MOV A,#data	Move immediate data to Accumulator	12
MOV Rn,A	Move Accumulator to register	12
MOV Rn,direct	Move direct byte to register	24
MOV Rn,#data	Move immediate data to register	12
MOV direct,A	Move Accumulator to direct byte	12
MOV direct,Rn	Move register to direct byte	24
MOV direct,direct	Move direct byte to direct byte	24
MOV direct,@Ri	Move indirect RAM to direct byte	24
MOV direct,#data	Move immediate data to direct byte	24
MOV @Ri,A	Move Accumulator to indirect RAM	12
MOV @Ri,direct	Move direct byte to indirect RAM	24
MOV @Ri,#data	Move immediate data to indirect RAM	12
MOV DPTR,#data16	Load Data Pointer with 16-bit const	24
MOVC A,@A+DPTR	Move Code byte rel. to DPTR to Acc.	24
MOVC A,@A+PC	Move Code byte rel. to PC to Acc.	24
MOVX A,@Ri	Move Ext. RAM (8-bit addr.) to Acc.	24
MOVX A,@DPTR	Move Ext. RAM (16-bit addr) to Acc.	24
MOVX @Ri,A	Move Acc. to Ext. RAM (8-bit addr.)	24
MOVX @DPTR,A	Move Acc. to Ext. RAM (16-bit addr)	24
PUSH direct	Push direct byte onto stack	24
POP direct	Pop direct byte from stack	24
XCH A,Rn	Exchange register with Accumulator	12
XCH A,direct	Exchange direct byte with Acc.	12
XCH A,@Ri	Exchange indirect RAM with Acc.	12
XCHD A,@Ri	Exchange low order digit indirect	
	RAM with Accumulator	12

Program Branching

Mnemonic	Description - Program Branching	Osc. Period
ACALL addr11	Absolute subroutine call	24
LCALL addr16	Long subroutine call	24
RET	Return from subroutine	24
RETI	Return from interrupt	24
AJMP addr11	Absolute jump	24
LJMP addr16	Long jump	24
SJMP rel	Short jump (relative address)	24
JMP @A+DPTR	Jump indirect relative to the DPTR	24
JZ rel	Jump if Accumulator is zero	24
JNZ rel	Jump if Accumulator is not zero	24
CJNE A,direct,rel	Compare direct byte to Accumulator	
	and jump if not equal	24
CJNE A,#data,rel	Compare immediate data to	
	Accumulator and jump if not equal	24
CJNE Rn,#data,rel	Compare immediate data to register	
	and jump if not equal	24
CJNE @Ri,#data,rel	Compare immediate data to indirect	
	RAM and jump if not equal	24
DJNZ Rn,rel	Decr. register and jump if not zero	24
DJNZ direct,rel	Decrement direct byte and jump if	
	not zero	24
NOP	No operation	12

Notes on Instruction Set and Addressing Modes

- **Rn** = Register R0 - R7 of the currently selected register bank. direct = 8-bit internal data location's address.This could be an internal Data RAM location (0-127) or a SFR.
- **@Ri** = 8-bit internal Data RAM location addressed indirectly through register R0 or R1.
- **#data** = 8-bit constant included in instruction.
- **#data16** = 16-bit constant included in instruction.
- **Addr11** = 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K byte page of Program Memory as the first byte of the following instruction.
- **addr16** = 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64K byte Program Memory address space.
- **rel** = Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
- **bit** = Direct addressed bit in internal Data RAM or SFR.

Bit Operations

Mnemonic	Description	Osc. Period
CLR C	Clear Carry	12
CLR bit	Clear direct bit	12
SETB C	Set Carry	12
SETB bit	Set direct bit	12
CPL C	Complement Carry	12
CPL bit	Complement direct bit	12
ANL C,bit	AND direct bit to Carry	24
ANL C,/bit	AND complement of dir. bit to Carry	24
ORL C,bit	OR direct bit to Carry	24
ORL C,/bit	OR complement of dir. bit to Carry	24
MOV C,bit	Move direct bit to Carry	12
MOV bit,C	Move Carry to direct bit	24
JC rel	Jump if Carry is set	24
JNC rel	Jump if Carry not set	24
JB bit,rel	Jump if direct bit is set	24
JNB bit,rel	Jump if direct bit is not set	24
JBC bit,rel	Jump if dir. bit is set & clear bit	24

PSW: il registro di stato

CY	AC	F0	RS1	RS0	OV	-	P

I bit del registro di stato sono indirizzabili separatamente:

- **CY** (PSW.7) : CarrY flag, *ossia flag di riporto*. Questo bit viene utilizzato come accumulatore dalle istruzioni che operano sui singoli bit. In tal caso il bit suddetto viene indicato con la lettera C.
- **AC**(PSW.6) : Auxiliary Carry *ovvero flag di riporto ausiliario*; utilizzato nelle operazioni in BCD, rappresenta il riporto che si può verificare tra il 4° e il 5° bit di un dato da 1 byte.
- **FO** (PSW.5) : Flag 0 *utilizzabile dall'utente*.
- **RS1** (PSW.4) : Register bank Selector bit 1.
- **RSO** (PSW.3) : Register bank Selector bit 0. Insieme a RS1 *seleziona il banco di 8 registri*. R0 - R7 nella memoria RAM interna da 128 byte:

RS1	RS0	REGISTER BANK ADDRESS
0	0	00H-07H
0	1	08H-0FH
1	0	10H-17H
1	1	18H-1FH

- **OV**(PSW.2) : OVerflow; flag che *indica il superamento dei limiti dell 'operando*. - (PSW. 1) : flag *definibile dall'utente*.
- **P** (PSW.O) : Parity flag *flag di parità*. I1 suo valore rispecchia direttamente il numero di 1 presenti nell'accumulatore. P = 1 se il numero di bit ad 1 nell'accumulatore è dispari; P = O se il numero di bit ad 1 nell'accumulatore è pari.

Tabella vettori di interruzione

Indirizzo di salto	Sorgente interruzione	Indirizzo di memoria scheda DB8051
0003H(3)	INTO	80A1H
000BH(11)	Timer0	80A4H
0013H (19)	INT1	80A7H
001BH(27)	Timerl	80AAH
0023H (35)	Porta seriale	80ADH

Esempio di inizializzazione vettore di interrupt:

```
MOV DPTR,#80A7H
MOV A,#85H
MOVX @DPTR,A
INC DPTR
MOV A,#00
MOVX @DPTR,A
```

il servizio di interruzione per INT1 di trova alla 8500h