

MICROPROCESSORS

INPUT/OUTPUT PORTS, MEMORY



SFR

2

- 128 B in the address range of 80H to FFH

Symbol	Name	Symbol	Name
* ACC	Accumulator	* IP	Interrupt priority register
* B	B register	* IE	Interrupt enable register
* PSW	Program state word	TMOD	Timer mode register
SP	Stack pointer	* TCON	Timer control register
DPTR	Data pointer	TH0	Higher byte of Timer 0
DPL	Lower byte of DPTR	TL1	Lower byte of Timer 0
DPH	Higher byte of DPTR	TH1	Higher byte of Timer 1
* P0	port 0	TL1	Lower byte of Timer 0
* P1	port 1	* SCON	Serial control register
* P2	port 2	SBUF	Serial buffer
* P3	port 3	PCON	Power control register

Adres	Symbol	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F0	B								
E0	ACC								
D0	PSW	CY	AC	FO	RS1	RS0	OV		P
B8	IP	—	—	—	PS	PT1	PX1	PT0	PX0
B0	P3	/RD	/WR	T1	T0	/INT1	/INT0	TxD	RxD
A8	IE	EA	—	—	ES	ET1	EX1	ET0	EX0
A0	P2								
99	SBUF								
98	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
90	P1								
8D	TH1								
8C	TH0								
8B	TL1								
8A	TL0								
89	TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
		Timer 1				Timer 0			
88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
87	PCON	SMOD	—	—	—	GF1	GF0	PD	IDL
83	DPH								
82	DPL								
81	SP								
80	P0								



Ports

3

- ▶ 8051 has four 8-bit ports P0, P1, P2, P3.
- ▶ Ports can be addressed as whole bytes or by each bit of the port.
- ▶ 32 I/O lines are available - each bit of a port corresponds to different I/O line
- ▶ In the DSM-51 System only P1 and P3 (6 lines) can be used.
- ▶ The ports are placed within the Special Function Registers (SFR) area
- ▶ Notable connections:
 - ▶ P1.5 – buzzer, P1.7 – TEST diode,
 - ▶ P3.4 – Timer 0, P3.5 – Timer 1



General Purpose Registers

4

- ▶ 128 bytes of RAM in the address range of 00 - 7FH.
- ▶ First 32 bytes form 4 banks of registers.
- ▶ Each bank consists of 8 registers: R0 - R7.
- ▶ Selection of a register bank is performed by modifying the value of the RS0 and RS1 bits located in the PSW register.

RS1	RS0	bank	address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH



Registers

- ▶ The remaining RAM memory (20H to 7FH) is divided into two parts
- ▶ 20H - 2FH – this registers are treated as single bits.
- ▶ 30H – 7FH – general purpose address range.

Assembler

6

► Syntax

Label	Instruction	Arguments	Commentary
	LJMP	START	;Jump to START label
	ORG	100H	;Set the begining of the program to address100H
START:			
	CLR	P1.7	; Turn on the LED on line 7 of port 1
	SJMP	\$;Stay in this line. End of program

Number	Number format		
	Decimal	Binary	Hexadecimal
1	1	1B	01H
2	2	10B	02H
15	15	1111B	0FH
65	65	1000001B	41H



Assembler - help

7

Ex. 2

► Diode control

► Diode ON

CLR P1.7

► Diode OFF

SETB P1.7

► Diode state toggle

CPL P1.7

► Delay

► MOV A,#1

► LCALL DELAY_100MS

The subroutine DELAY_100MS modifies accumulator register. Each call of the DELAY_100MS requires setting the accumulator value that will be used as a multiplier.

Ex. 4

► Buzzer

► Line 5 port P1: P1.5

► Bytes operations:

► MOV P1,#01111111B – Diode ON

► MOV P1,#11011111B – Buzzer ON

► MOV P1,#01011111B – Diode and Buzzer ON

► XRL P1,#10100000B – Toggle the state of Diode and Buzzer



Memory - Addressing

8

- ▶ Register

MOV A,#10

MOV R0,#10

- ▶ Direct

MOV ACC,#10 ;ACC is the direct address of the
;accumulator

MOV 00H,#10 ; R0 register of bank 0

- ▶ Indirect

MOV A,@R0 ;

C/C++ analogy

MOV A,#10 – int A = 10;

MOV R0,#3 – int R0 = 3;



LOOPS

- ▶ Selected loop commands
 - ▶ LJMP,AJMP,SJMP – unconditioned jumps
 - ▶ JZ – jump if A=0
 - ▶ JNZ – jump if A != 0
 - ▶ CJNE – **C**ompare , **J**ump if **N**ot **E**qual
 - ▶ DJNZ – Decrease, **J**ump if **N**ot **Z**ero

Loops - examples

10

- ▶ Infinite loop (while(1))

LOOP:

...

LJMP LOOP

- ▶ Conditional loop

- ▶ DJNZ

MOV R0,#10

LOOP:

...

DJNZ R0,LOOP; stay in loop while R0 > 0

- ▶ CJNE

LOOP:

...

CJNE R0,#10,LOOP; stay in loop while R0 != 10



Assembler - help

11

Ex. 5

- ▶ The **Rx** symbols can be linked with 4 different memory addresses:
 - ▶ Eg., R7 - 07H,0FH,17H, 1FH
- ▶ The Register bank selection is made by changing RS1 and RS0 bits:
 - ▶ Setting bank 3:
SETB RS1
SETB RS0
- ▶ Change of the register bank changes the address of the Rx symbol.
- ▶ **Subroutines LCD_CLR i WRITE_HEX modifies the accumulator!**

Ex. 8

- ▶ Use loops
- ▶ Only symbols R0 and R1 can be used for indirect addressing
- ▶ Indirect addressing is denoted with @ symbol
- ▶ Example:

```
MOV R0,#07H;
```

```
MOV @R0,#1
```

In this example a value of 1 is written into register 07H „pointed” by R0.

