

# Microprocessadores e Aplicações

## *Acetatos de apoio às aulas teóricas*

Ana Cristina Lopes  
Dep. Engenharia Electrotécnica  
<http://orion.ipt.pt> [anacris@ipt.pt](mailto:anacris@ipt.pt)

1. Introdução aos Microcontroladores da família MCS51;
  - (a) Temporização;
  - (b) Organização da Memória Interna;
  - (c) Modos de endereçamento;
2. Tipos de Instruções:
  - (a) Aritméticas;
  - (b) Lógicas;
  - (c) Transferência de dados;
  - (d) Booleanas;
  - (e) Controlo de Fluxo;

- Introdução aos Microcontroladores

Introdução à família MCS51

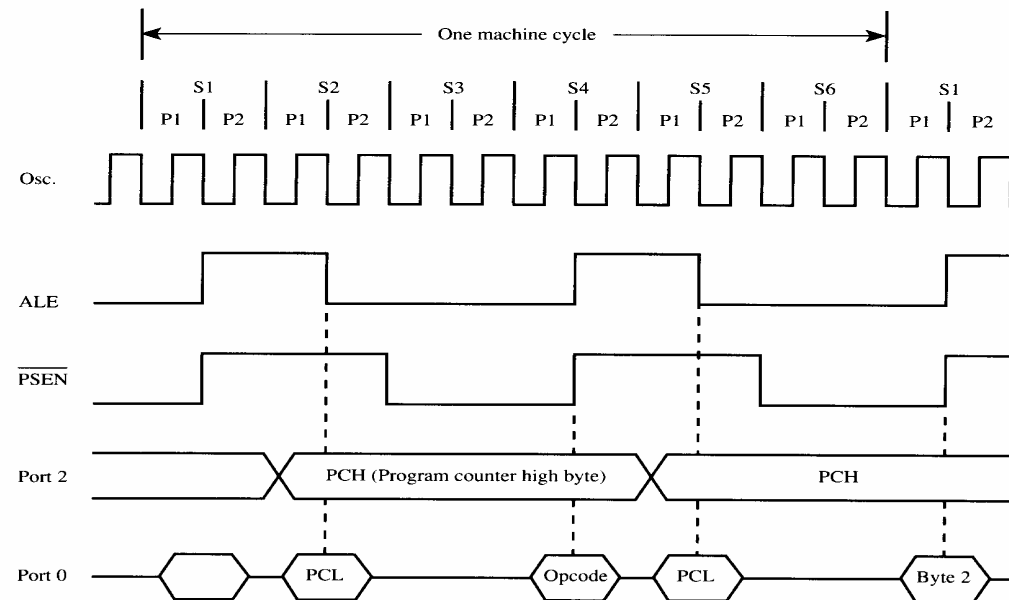
- Temporização

- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

- Um ciclo de máquina do 8051 consiste na sequência de 6 estados, numerados de S1 a S6.
- Um ciclo de máquina demora 12 períodos de relógio - 1 microsegundo.
- A frequência do oscilador é 12 MHz.
- Cada estado é composto por dois períodos de relógio ou duas fases, denominadas por P1 e P2.
- Durante um ciclo de máquina típico, o nível da ALE varia duas vezes, o que corresponde à leitura de dois bytes da memória externa do programa.
- O número de estados por ciclo de máquina varia consoante o processador.

A Figura seguinte mostra a temporização do acesso à memória externa.



- Notar que o PC (Program counter) = PCH (High) + PCL (Low), dois registos de oito bits.
- As frequências de ALE e \*PSEN são iguais a 1/6 da frequência de relógio.
- Duty Cycle (ALE) = 33% e o Duty Cycle (\*PSEN) = 50%;

- A leitura e a escrita na memória de dados externa necessita do dobro do tempo necessário para aceder à memória de programa externa.
- $f(ALE) \neq \frac{1}{6} f(XTAL)$  devido à ausência de 1 pulso de ALE;
- $f(*PSEN) \neq \frac{1}{6} f(XTAL)$  devido à ausência de 2 pulsos de \*PSEN;
- No acesso à memória de dados externa é utilizado um endereço de 16 bits. Esse endereço pode ser:
  - ◆ Registo de 16 bits - ponteiro de dados - DPTR = DPH+DPL;
  - ◆ R0 ou R1 e o conteúdo da SFRP2;

	MSB	LSB	Exemplo Leitura	Exemplo Escrita
Com DPTR	DPH	DPL	MOVX A,@DPTR	MOVX @DPTR, A
Com P2 e Ri	SFRP2	Ri (i = 0, 1 )	MOVX A,@Ri	MOVX @Ri, A

# Acesso à memória de dados

● Introdução aos Microcontroladores

Introdução à família MCS51

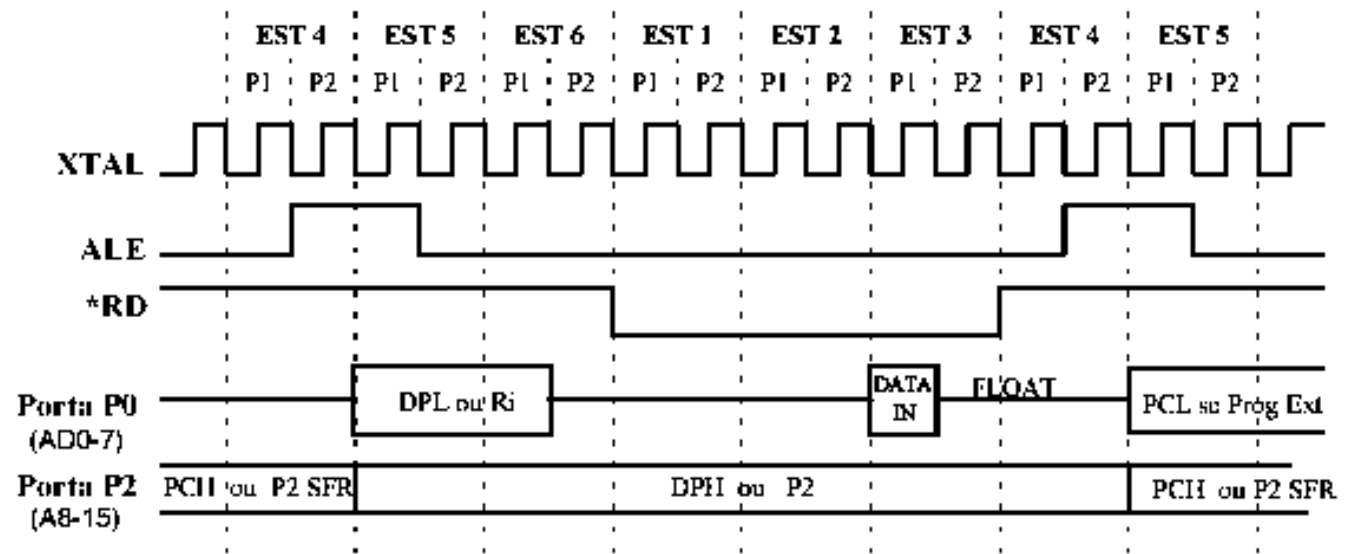
● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

- Notar que quando se usa DPTR para endereçar a memória externa, os 16 bits dos endereços são emitidos por P0 e P2;
- Quando se usa o Ri, os 16 bits são formados pelo conteúdo de Ri (LSB) e pelo que se encontra no SFR P2, ou seja, não se altera o conteúdo de P2.



■ A CPU responde gerando um Reset interno de acordo com a temporização mostrada na figura seguinte.

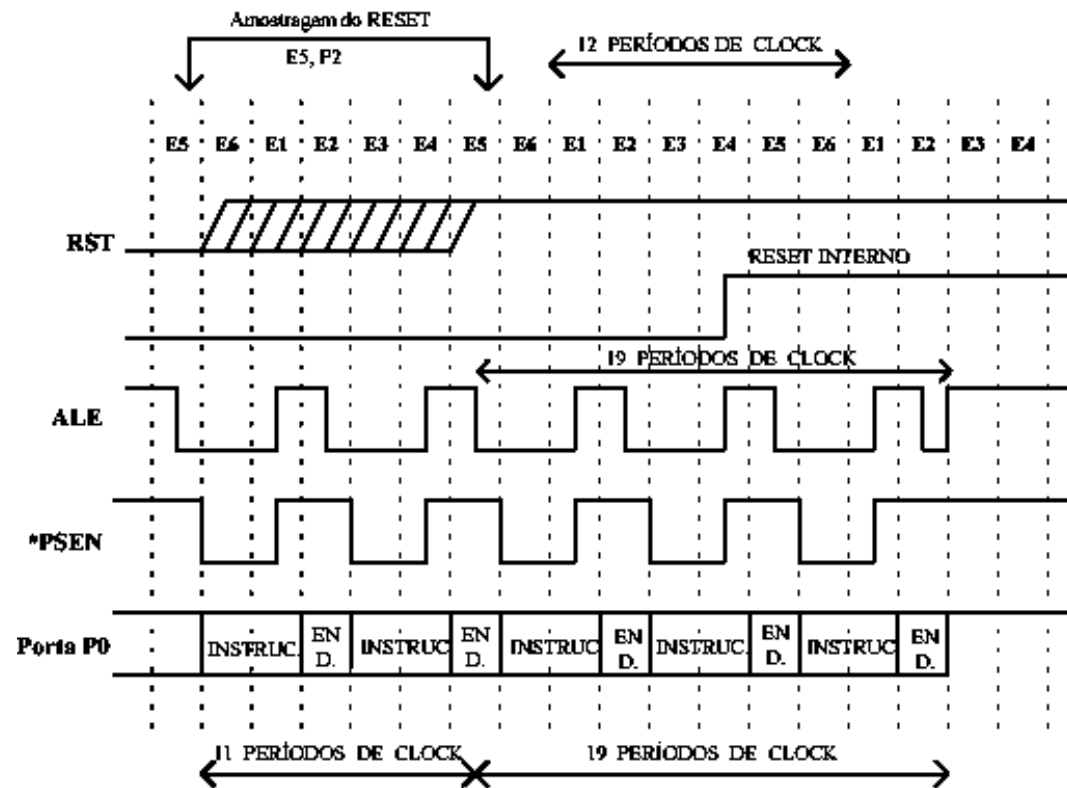


Figura 3.6. Temporização do RESET.

- A entrada de RESET corresponde ao pino RST do 8051;
- O Reset do sistema é realizado quando este pino se mantém num nível alto durante pelo menos 2 ciclos de máquina;
- A entrada RST é assíncrona e é amostrada durante a fase P2 do estado 5 (S5);
- Após a entrada RST ser amostrada como alta, os pinos ainda mantêm a actividade durante 19 períodos de relógio;
- Os pinos cessam a sua actividade 19 a 31 pinos de relógio depois do RST ser activada;
- Após o RST, os pinos ALE e PSEN ficam a 1;
- Após o RST voltar a zero (desactivado), são necessários 1 a 2 períodos de relógio para que ALE e PSEN reajam; razão pela qual outros dispositivos externos não podem ser sincronizados com o RESET do 8051;
- A RAM interna não é afectada pelo Reset;
- Após o Reset a CPU é actualizada com os seguintes valores:  $SP = 7$ ;  $SBUF = ?$ ;  $P0, P1, P2, P3 = FFh$  e os outros SFR são iguais a zero.



- Introdução aos Microcontroladores

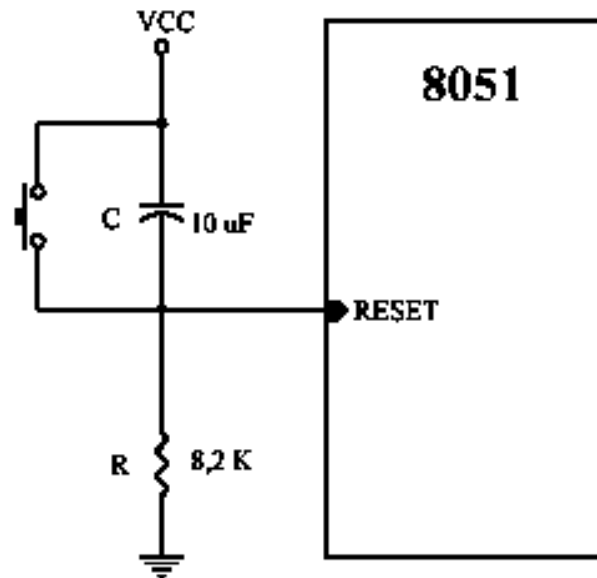
Introdução à família MCS51

● Temporização

- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

Para famílias HMOS:



- Quando é pressionado o botão de pressão, o circuito RC assegura no pino RST um nível alto durante uma quantidade de tempo que depende dos valores R e C;
- O esquema da figura é apenas válido para circuitos HMOS, para circuitos CHMOS a resistência de 8,2K pode ser omitida, porque há uma resistência de pull-down interna e o condensador deve ser reduzido para  $1 \mu F$ .

● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

- a memória interna do 8051 consiste numa ROM de 4K e uma RAM de 256 bytes. Esta última permite um acesso rápido aos dados e para muitas aplicações pode eliminar a necessidade de RAM externa.
- Para além da velocidade de acesso, a RAM interna tem a vantagem de possuir áreas que podem ser acedidas bit a bit (útil para operações booleanas);
- O espaço de endereçamento para acesso à RAM interna é de 8 bits, o que implica um máximo de 256 bytes endereçáveis.

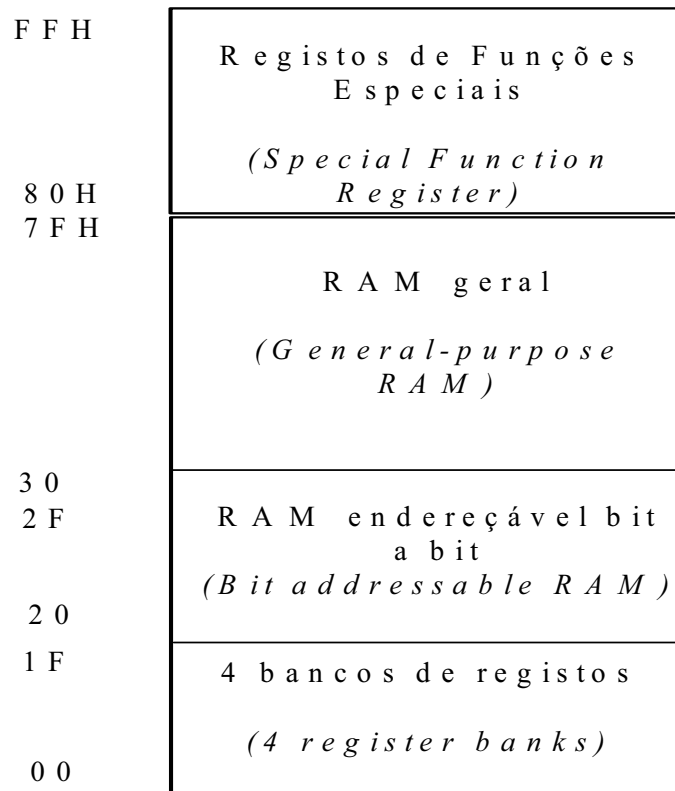
- Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

A Figura seguinte mostra a organização da RAM interna do 8051.



**Exemplo:** Tendo em conta que não há RAM externa e sabendo que se pretende utilizar os 4 bancos de registos, os 16 bytes endereçáveis bit a bit e 10 variáveis de 1 byte, quanto espaço fica disponível para a possibilidade de ocorrência de interrupções e chamadas de subrotinas?

**Interrupções:** Se forem utilizadas interrupções, para cada uma são guardadas: PC (2 bytes), PSW, ACC e B  $\Rightarrow$  5 bytes. Como só podem ocorrer duas interrupções em simultâneo  $\Rightarrow$  10 bytes.

**Subrotinas:** Para cada Call são guardados: PC (2 bytes), PSW, ACC e B  $\Rightarrow$  5 bytes. Tal implica que há disponibilidade para 12 Call aninhadas.

● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

- Os primeiros 32 bytes da RAM estão reservados para os bancos de registos;
- O conjunto de instruções do 8051 suporta 8 registos - R0 a R7 - que por defeito (após Reset) se encontram de 00h a 07h;
- Exemplo: A próxima instrução lê o conteúdo do registo R5 para o acumulador: *MOVA, R5*;

## Como utilizar os registos de outro banco de dados?

- O banco pretendido tem de ser seleccionado nas flags da PSW (Program Status Word).

### Exemplo:

*MOVPSW, #0 %* selecciona o Banco 0

*MOVA, R5 %* coloca no acumulador o valor de R5 do banco 0.

## Porquê utilizar bancos de registos?

Porque permite uma mudança rápida e eficaz de contexto, onde secções separadas de software utilizam um conjunto privado de registos, o qual é independente de outras secções de software.

● Introdução aos Microcontroladores

Introdução à família MCS51

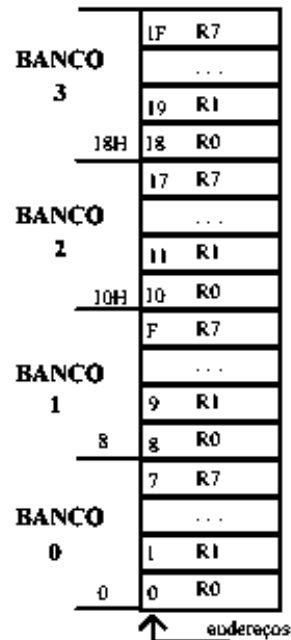
● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

A Figura seguinte mostra como estão organizados os bancos de dados.



- Notar que mesmo que um banco de dados não esteja seleccionado, é possível aceder aos seus registos através do seu endereço.

# Special Function Registers - SFR

● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

128 bytes da RAM (80h a FFh) são reservados para as SFRs. A tabela seguinte apresenta a lista das SFR, com o respectivo endereço.

Byte address	Bit address	
FF		
F0	F7 F6 F5 F4 F3 F2 F1 F0	B
E0	E7 E6 E5 E4 E3 E2 E1 E0	ACC
D0	D7 D6 D5 D4 D3 D2 - D0	PSW
B8	- - - BC BB BA B9 B8	IP
B0	B7 B6 B5 B4 B3 B2 B1 B0	P3
A8	AF - - AC AB AA A9 A8	IE
A0	A7 A6 A5 A4 A3 A2 A1 A0	P2
99	not bit addressable	SBUF
98	9F 9E 9D 9C 9B 9A 99 98	SCON
90	97 96 95 94 93 92 91 90	P1
8D	not bit addressable	TH1
8C	not bit addressable	TH0
8B	not bit addressable	TL1
8A	not bit addressable	TL0
89	not bit addressable	TMOD
88	8F 8E 8D 8C 8B 8A 89 88	TCON
87	not bit addressable	PCON
83	not bit addressable	DPH
82	not bit addressable	DPL
81	not bit addressable	SP
80	87 86 85 84 83 82 81 80	P0



- **A - Accumulator:** Acumulador dos valores resultantes das diversas operações (notar que A é a designação para a utilização em instruções e Acc é a designação para o local na zona de memória das SFRs;
- **B:** Utilizado nas operações de divisão e multiplicação; nos restantes casos pode ser utilizado como auxiliar;
- **SP - Stack Pointer:** Ponteiro de pilha - incrementado antes de instruções PUSH e CALL, decrementado após instruções POP e RET;
- **DPTR - Data Pointer:** Ponteiro para a memória de dados externa; é um registo de 16 bits formado por DPH+DPL;
- **P0, P1, P2, P3 e P4:** Portos de Entrada/ Saída do 8051;
- **IP - Interrupt Priority:** Prioridade das interrupções;

# Special Function Registers - SFR

● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

- **IE - Interrupt Enable:** Permissão de interrupções;
- **TMOD - Timer/Counter Mode:** Modo de operação dos temporizadores e contadores;
- **TCON - Timer/Counter Control:** Registo de controlo associado aos temporizadores e contadores;
- **SCON - Serial Control:** Registo de controlo da porta série;
- ;
- **SBUF - Serial Buffer:** Dois registos: o de leitura que recebe os dados a partir da porta série e o de escrita que envia a partir da porta série.

# Special Function Registers - SFR

● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

■ **PSW - Program Status Word:** Registo de flags de estado;

A tabela seguinte mostra como se encontra organizado o registo de flags de estado:

7	6	5	4	3	2	1	0
CY	Ac	F0	RS1	RS0	OV	-	P

■ **PSW.7 - CY - Carry Flag:** Flag de transporte;

■ **PSW.6 - Ac - Auxiliar Flag:** Flag de transporte auxiliar para operações com BCD - *Binary Coded Decimal*;

■ **PSW.5 - F0 - Flag 0:** Flag de uso geral;

■ **PSW.4 - RS1 - Range Selector:** Seleccionador de banco, bit 1;

■ **PSW.3 - RS0 - Range Selector:** Seleccionador de banco, bit 0;

■ **PSW.2 - OV - Overflow Flag:** Flag de transbordo;

■ **PSW.1** - Flag a definir pelo utilizador;

■ **PSW.0 - P -Parity Flag** - Flag de paridade (1 = ímpar);

## Como funciona a selecção de bancos?

A tabela seguinte mostra a selecção do banco de registos de acordo com os valores de RS1 e RS0:

RS1	RS0	Banco Seleccionado
0	0	Banco 0
0	1	Banco 1
1	0	Banco 2
1	1	Banco 3

- Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

A Figura seguinte mostra o mapa de RAM interno.

SFR									
FFH	-	-	-	-	-	-	-	-	F8H
F7H	-	-	-	-	-	-	-	B	F0H
EFH	-	-	-	-	-	-	-	-	E8H
E7H	-	-	-	-	-	-	-	Acc	E0H
DFH	-	-	-	-	-	-	-	-	D8H
D7H	-	-	-	-	-	-	-	PSW	D0H
CFH	-	-	-	-	-	-	-	-	C8H
C7H	-	-	-	-	-	-	-	-	C0H
BFH	-	-	-	-	-	-	-	IP	B8H
B7H	-	-	-	-	-	-	-	P3	B0H
AFH	-	-	-	-	-	-	-	IE	A8H
A7H	-	-	-	-	-	-	-	P2	A0H
9FH	-	-	-	-	-	-	SBUF	SCON	98H
97H	-	-	-	-	-	-	-	P1	90H
8FH	-	-	TH1	TH0	TL1	TL0	TMOD	TCON	88H
87H	PCON	-	-	-	DPH	DPL	SP	P0	80H
128 LOWER									
7FH	7FH	7EH	...						78H
77H									70H
6FH									68H
67H									60H
5FH									58H
57H									50H
4FH									48H
47H									40H
3FH									38H
37H						...	31H	30H	30H
2FH	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H	28H
27H	27H	26H	25H	24H	23H	22H	21H	20H	20H
BK3	1FH	R7	R6	R5	R4	R3	R2	R1	R0
BK2	17H	R7	R6	R5	R4	R3	R2	R1	R0
BK1	0FH	R7	R6	R5	R4	R3	R2	R1	R0
BK0	07H	R7	R6	R5	R4	R3	R2	R1	R0

- Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

A Figura seguinte mostra o mapa de RAM interno.

MAPA DOS 256 BITS												
END	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	END	END	NOM
BIT	7	6	5	4	3	2	1	0	BIT	BYTE	BYTE	
FFH	-	-	-	-	-	-	-	-	FFH	FFH		
F7H	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	FOH	FOH	B	
EFH	-	-	-	-	-	-	-	-	EOH	EOH		
E7H	Acc.7	Acc.6	Acc.5	Acc.4	Acc.3	Acc.2	Acc.1	Acc.0	E0H	E0H	Acc	
DFH	-	-	-	-	-	-	-	-	D8H	D8H	-	
D7H	CY	AC	F0	RS1	RS0	OV	-	-	DOH	DOH	PSW	
CFH	-	-	-	-	-	-	-	-	COH	COH	-	
C7H	-	-	-	-	-	-	-	-	COH	COH	-	
BFH	-	-	-	PS	PT1	PX1	PT0	PX0	B8H	B8H	IP	
B7H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0H	B0H	P3	
AFH	EA	-	-	ES	ET1	EX1	ET0	EX0	A8H	A8H	IE	
A7H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0H	A0H	P2	
9FH	SM0	SM1	SM2	REN	TBB	RBB	TI	RI	98H	98H	SCON	
97H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90H	90H	P1	
8FH	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H	88H	ICON	
87H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80H	80H	P0	
7FH	7FH	7EH	7DH	7CH	...				78H	7FH		
77H									70H	7EH		
6FH									68H	6EH		
67H									60H	6CH		
5FH									58H	5EH		
57H									50H	5AH		
4FH									48H	4EH		
47H									40H	4EH		
3FH									38H	3EH		
37H									30H	3EH		
2FH									28H	2EH		
27H									20H	2EH		
1FH									18H	1EH		
17H									10H	1EH		
0FH				...	0EH	0AH	09H	08H	08H	21H		
07H	07H	06H	05H	04H	03H	02H	01H	00H	00H	20H		

● Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

De acordo com o "data-sheet" do 8051 existem 6 modos possíveis de endereçamento, designadamente:

- Imediato;
- Directo;
- Indirecto;
- Registos;
- Específico de registos;
- Indexado;

## Modo Imediato:

- Um dos operandos é uma constante;
- Exemplo MOV A, #100;
- Com esta instrução, o valor 100 é carregado para o acumulador;
- Note-se que o byte 100 é um dado imediato;
- A presença do sinal # indica uma operação imediata;



● Introdução aos Microcontroladores

Introdução à família MCS51

● Temporização

● Memória Interna

● Modos de Endereçamento

Tipos de Instruções

## Modo Directo:

- Neste caso um dos operandos especifica um endereço de 8 bits da RAM interna;
- Exemplo MOV A, 20;
- Com esta instrução, o conteúdo do endereço 20 da RAM interna é transferido para o acumulador;
- Todo o endereçamento directo usa apenas a RAM interna;

## Modo Indirecto:

- Neste caso especifica-se um registo onde está o endereço do operando;
- Exemplo MOV A, @R0;
- Com esta instrução, o conteúdo do endereço que está em R0 é transferido para o acumulador;

● Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

## Modo de Registos:

- Neste caso, a instrução especifica um registo R0 a R7 como operando;
- Trata-se de operações codificadas com apenas 1 byte, em que o registo do operando é especificado por três bits desse byte;
- Esta forma de endereçamento evita a utilização de 1 byte adicional para especificar o registo;
- Exemplo MOV A, R0;
- Com esta instrução, o conteúdo que está em R0 é transferido para o acumulador;

● Introdução aos Microcontroladores

Introdução à família MCS51

- Temporização
- Memória Interna
- Modos de Endereçamento

Tipos de Instruções

## Modo Específico de Registos:

- Algumas operações são específicas para certos registos;
- A execução de instruções, nas quais é utilizado este modo de endereçamento, afecta o conteúdo de determinado registo (por exemplo: o acumulador, o registo B ou o DPTR);
- Neste caso não é necessário nenhum byte de endereço na instrução;
- Exemplo INC B;

## Modo Indexado:

- Este modo de endereçamento é utilizado para proporcionar o acesso a tabelas de valores que são armazenadas na memória de programa (*look-up tables*);
- O endereço base da tabela tem 16 bits é escrito no DPTR ou no PC;
- O endereço relativo é armazenado no acumulador;
- Exemplo: Supor que DPTR = FF00h e A = 10h;
- A instrução `MOVC A, @A+DPTR` transfere para o acumulador o conteúdo da posição de memória de programa com endereço FF10h.

● Introdução aos Microcontroladores

○ O 8051 possui 5 tipos de instruções fundamentais:

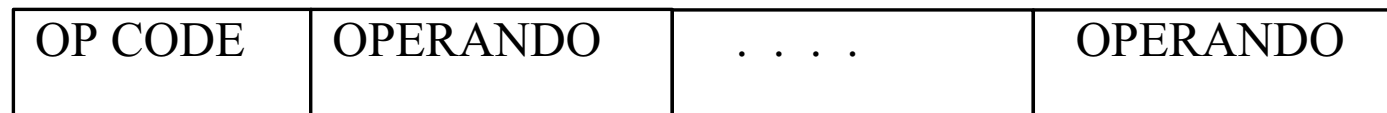
Introdução à família MCS51

Tipos de Instruções

● Tipos de Instruções

1. Aritméticas;
2. Lógicas;
3. Transferência de dados;
4. Booleanas;
5. Controlo de Fluxo;

- Uma instrução é um comando executado pelo processador;
- Um programa consiste numa sequência de instruções;
- Instruções expressas em números binários designam-se por código de máquina (para facilitar a leitura o código de máquina) é normalmente expresso em hexadecimal;
- Normalmente cada instrução integra: a operação, normalmente designada por "OPCODE" (*Operation Code*), operando fonte dos dados e o operando destino do resultado;



Formato Geral das Instruções

**OPCODE - (*Operation Code*):** Indica qual a operação que o processador tem de realizar. O OPCODE do 8051 tem 8 bits o que possibilita a existência de 256 instruções.

**Operandos:** Indicam quais os dados que o processador necessita para executar a operação especificada pelo OPCODE. Uma instrução pode ter vários operandos. A existência de muitos operandos requer mais espaço de memória para representar a instrução. Por esse motivo, cada instrução está normalmente limitada a dois operandos.

Com uma instrução de dois operandos, um deles é re-escrito; Veja-se o exemplo seguinte:

**24 09 corresponde a ADD A, #9H**

Isto significa que '24H' é o opcode que instrui o 8051 a adicionar um número ao conteúdo do acumulador A. '09H' é o dado imediato que é adicionado ao acumulador. O resultado da adição é colocado no acumulador (o valor que lá estava anteriormente é apagado). Note-se '09H' é o operando fonte e o acumulador A é o operando destino (local onde fica armazenado o resultado).



De seguida apresentam-se as instruções do 8051 e alguns exemplos para cada uma delas. Para cada instrução serão apresentados o número de bytes do OP-CODE, o número de ciclos de máquina (MC=12 períodos de relógio) e as flags que lhe estão associadas. De seguida apresenta-se a lista das instruções aritméticas:

1. Soma de 8 bits;
2. Soma de 8 bits com carry;
3. Subtração de 8 bits com borrow;
4. Incremento de 8 bits;
5. Decremento de 8 bits;
6. Incremento de 16 bits;
7. Multiplicação e divisão de 8 bits;
8. Ajuste decimal.

A Tabela seguinte é referente à instrução soma de 8 bits. O acumulador é o operando destino, onde se guarda o resultado (para todos os exemplos R0=25H e R1=37H).

Instrução	Op.	Bytes	MC	CY	AC	OV	Exemplos
ADD A,	Rn	1	1	X	X	X	$ADDA, R1 \Leftrightarrow A = A + R1$
	Directo	2	1				$ADDA, 20H \Leftrightarrow A = A + C(20H)$
	@Ri	1	1				$ADDA, @R0 \Leftrightarrow A = A + C(25H)$
	#data	2	1				$ADDA, \#data \Leftrightarrow A = A + data$

Nota:  $C(X)$  representa o conteúdo do endereço X.

A Tabela seguinte é referente à instrução soma de 8 bits com carry. O acumulador é o operando destino, onde se guarda o resultado.

Instrução	Op.	B	MC	CY	AC	OV	Exemplos Anteriores
ADDC A,	Rn	1	1	X	X	X	$A = A + R1 + CY$
	Directo	2	1				$A = A + C(20H) + CY$
	@Ri	1	1				$A = A + C(25H) + CY$
	#data	2	1				$A = A + data + CY$

# Subtracção de 8 bits e incremento de 8 bits

A Tabela seguinte é referente à instrução subtracção de 8 bits com borrow. O acumulador é o operando destino, onde se guarda o resultado.

Instrução	Op.	B	MC	CY	AC	OV	Exemplos
SUBB A,	Rn	1	1	X	X	X	$SUBB A, R1 \Leftrightarrow A - R1 - CY$
	Directo	2	1				$SUBB A, 20H \Leftrightarrow A - C(20H) - CY$
	@Ri	1	1				$SUBB A, @R0 \Leftrightarrow A - C(25H) - CY$
	#data	2	1				$SUBB A, \#data \Leftrightarrow A - data - CY$

A Tabela seguinte é referente à instrução incremento de 8 bits. Existem 4 instruções para o incremento de 8 bits.

Instrução	Op.	B	MC	CY	AC	OV	Exemplos
INC,	A	1	1	X	X	X	$INCA \Leftrightarrow A = A + 1$
	Rn	1	1				$INCR0 \Leftrightarrow R0 = 1 + R0(20H)$
	Directo	2	1				$INC25H \Leftrightarrow C(25H) = 1 + C(25H)$
	@Ri	1	1				$INC@R1 \Leftrightarrow C(37H) = 1 + C(37H)$

A Tabela seguinte é referente à instrução decremento de 8 bits, a qual é semelhante à instrução incremento de 8 bits. Existem 4 instruções para o decremento de 8 bits.

Instrução	Op.	B	MC	CY	AC	OV	Exemplos
DEC,	A	1	1	X	X	X	$DECA \Leftrightarrow A = A + 1$
	Rn	1	1				$DEC R0 \Leftrightarrow R0 = 1 + R0(20H)$
	Directo	2	1				$DEC 25H \Leftrightarrow C(25H) = 1 + C(25H)$
	@Ri	1	1				$DEC @R1 \Leftrightarrow C(37H) = 1 + C(37H)$

A Tabela seguinte é referente à instrução incremento de 16 bits, para qual apenas existe uma instrução.

Instrução	Operando Fonte	Bytes	MC	CY	AC	OV
INC,	DPTR	1	2	-	-	-

# Multiplicação e divisão de 8 bits

Na multiplicação e na divisão são sempre utilizados os registos A e B. Estas são as instruções que necessitam de mais tempo de execução.

- Multiplicação:  $A \times B \Rightarrow A = LSB; B = MSB$
- Divisão:  $A \div B \Rightarrow A = quociente; B = resto$

A Tabela seguinte é referente às instruções multiplicação e divisão de 8 bits.

Instrução	Operandos	Bytes	MC	CY	AC	OV
MUL,	AB	1	4	0	-	X
DIV,	AB	1	4	0	X	0

Esta permite operações com representação em BCD. O ajuste é válido apenas para as instruções de soma. A Tabela seguinte apresenta a única instrução respeitante ao ajuste decimal.

Instrução	Operando	Bytes	MC	CY	AC	OV
DA ,	A	1	1	X	X	-

### Observações:

- DA A Não converte simplesmente um número hexadecimal no acumulador para BCD;
- Esta instrução não se pode aplicar à subtracção;

As instruções lógicas são as que realizam as operações AND, OR e XOR. Note-se que neste caso o resultado não têm de obrigatoriamente ficar no acumulador. O 8051 tem as seguintes instruções lógicas:

1. AND de 8 bits;
2. OR de 8 bits;
3. XOR de 8 bits;
4. Operações lógicas com o acumulador.

# AND, OR e XOR de 8 bits

Estas instruções são semelhantes e afectam as mesmas flags. Por esse motivo apresenta-se apenas uma tabela para os três tipos de instruções. A instrução lógica designar-se-á por IL que deverá ser substituída da seguinte forma:

- IL=ANL no caso de se pretender um AND lógico;
- IL=ORL no caso de se pretender um OR lógico;
- IL=XRL no caso de se pretender um XOR lógico;

Instrução	Op.	B	MC	Exemplos
IL A,	Rn	1	1	$ILA, R1 \Leftrightarrow A = AILR1$
	Directo	2	1	$ILA, 20H \Leftrightarrow A = AILC(20H)$
	@Ri	1	1	$ILA, @R0 \Leftrightarrow A = AILC(25H)$
	#data	2	1	$ILA, \#data \Leftrightarrow A = AILdata$
IL Directo	A	2	1	$IL25H, A \Leftrightarrow C(25H) = AILC(25H)$
	#data	3	2	$IL25H, \#data \Leftrightarrow C(25H) = \#dataILC(25H)$



É possível realizar as seguintes operações lógicas com o acumulador:

- **CLR A:** Inicializa a zero o acumulador;
- **CPL A:** Realiza o complemento de 1 ao acumulador (inverte os bits);
- **RL A:** Roda o acumulador à esquerda;
- **RLC A:** Roda o acumulador à esquerda com carry;
- **RR A:** Roda o acumulador à direita;
- **RRC A:** Roda o acumulador à direita com carry;
- **SWAP A:** Troca os nibbles do acumulador.

# Operações lógicas com o acumulador

● Introdução aos Microcontroladores

Introdução à família MCS51

Tipos de Instruções

● Tipos de Instruções

A Tabela seguinte mostra as instruções e quais as flags afectadas pelas instruções lógicas com o acumulador.

Instrução	Op.	B	MC	CY	AC	OV
CLR	A	1	1	-	-	-
CPL				-	-	-
RL				-	-	-
RLC				X	-	-
RR				-	-	-
RRC				X	-	-
SWAP				-	-	-

# Operações lógicas com o acumulador

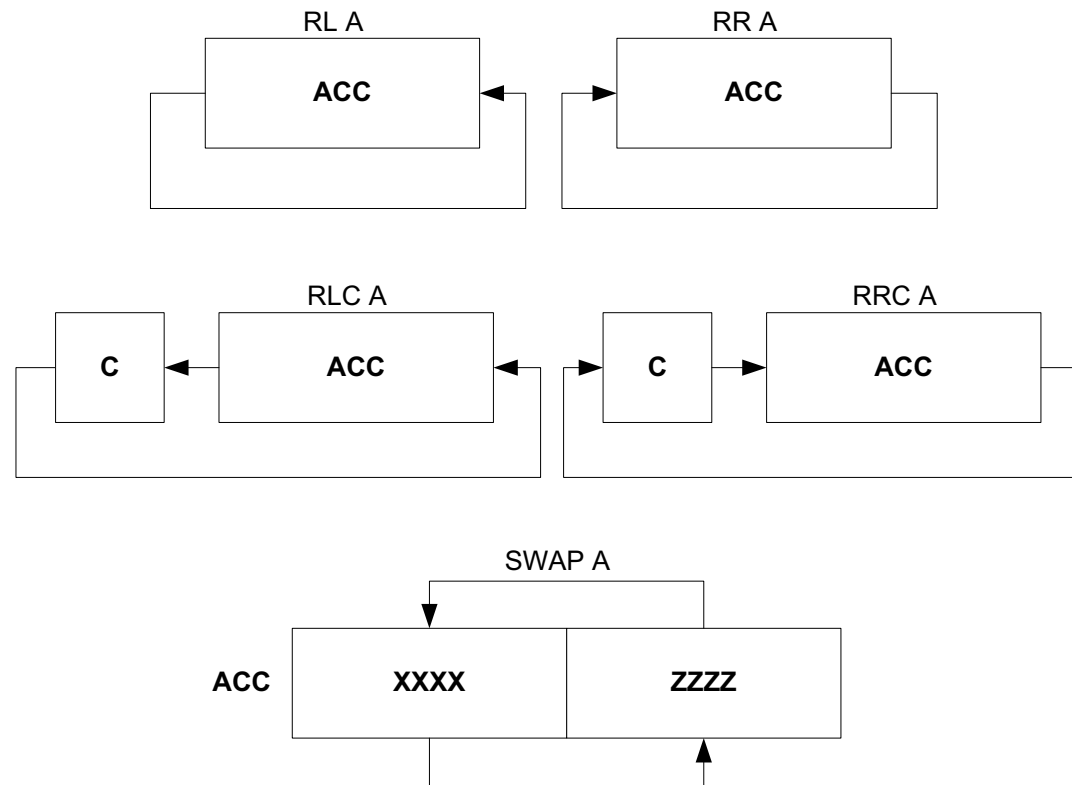
● Introdução aos Microcontroladores

Introdução à família MCS51

Tipos de Instruções

● Tipos de Instruções

A Figura seguinte apresenta os esquemas referentes a algumas das instruções lógicas com o acumulador.



Existem três tipos de instruções de transferência de dados:

1. As que trabalham com a RAM interna (22 instruções);
2. As que trabalham com a memória de dados externa (4 instruções);
3. As que trabalham com a memória de programa (2 instruções);

Regra geral as instruções de transferência de dados trabalham na RAM interna a menos que o contrário seja indicado.

A Tabela mostra as instruções de transferência de dados da RAM interna.

Instrução	Operando Fonte	Bytes	MC
MOV A,	Rn	1	1
	Directo	2	1
	@Ri	1	1
	#data	2	1
MOV Rn,	A	1	1
	Directo	2	2
	#data	2	1
MOV Directo,	A	2	1
	Rn	2	2
	Directo	3	2
	@Ri	2	2
	#data	3	2
MOV @Ri	A	1	1
	Directo	2	2
	#data	2	1

Note-se que não existem todas as combinações entre operandos. As seguintes instruções não são aceites pelo 8051:

- MOV A,A;
- MOV Rn,Rn;
- MOV Rn,@Ri;
- MOV @Ri,@Ri;
- MOV @Ri,Rn;

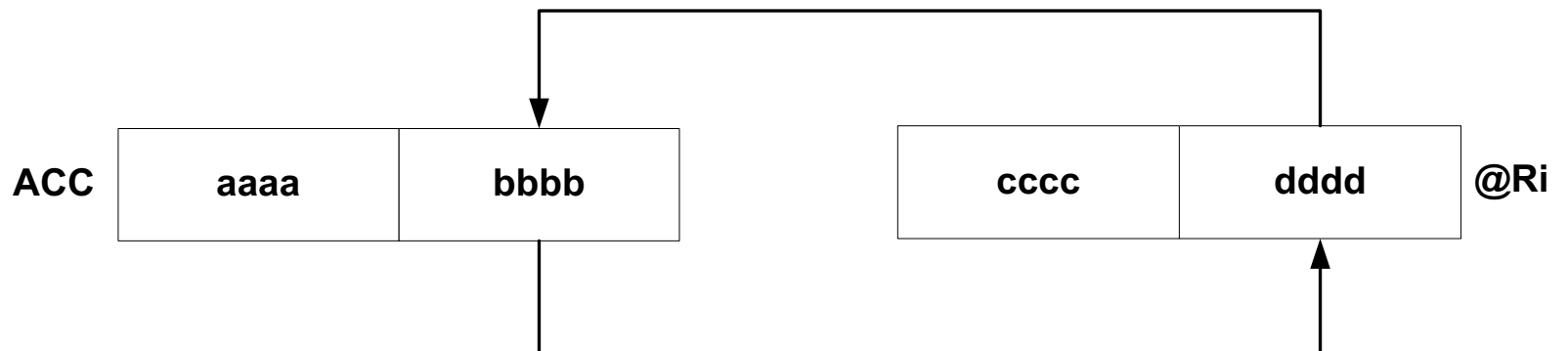
**Permutação de Bytes:** a tabela seguinte é referente às três instruções de permutação de bytes. Estas instruções são muito úteis, pois permitem a troca de conteúdo entre dois registos, sem que haja a necessidade de usar um registo auxiliar.

Instrução	Operando	Bytes	MC
XCH A,	Rn	1	1
	Directo	2	1
	@Ri	1	1

**Permutação de Nibbles:** a tabela seguinte é referente à instrução de permutação de nibbles. A instrução utiliza sempre o acumulador como um dos operandos e endereçamento indirecto. Esta instrução é útil para trabalhar em BCD e em conversões de hexadecimal para ASCII.

Instrução	Operando	Bytes	MC
XCHD A,	@Ri	1	1

A figura seguinte mostra como se processa a permutação de nibbles referente à instrução XCHD.





**Operações com a pilha:** a pilha opera exclusivamente com a RAM interna. A tabela seguinte é referente às duas instruções de pilha que trabalham com endereçamento directo. Note-se que se utilizam sempre dois bytes para o OPCODE e a instrução deve-se escrever PUSH Acc e não PUSH A. Lembra-se que Acc é o label para o endereço do acumulador.

Com a instrução PUSH primeiro incrementa-se o ponteiro (SP) e em seguida escreve-se a informação na pilha.

Exemplo: Se SP=7, a instrução PUSH Acc colocará o conteúdo do acumulador no endereço 8. A instrução POP funciona de forma inversa.

Instrução	Operando	Bytes	MC
PUSH	Directo	2	2
POP	Directo	2	2

Existem instruções de leitura e escrita que utilizam o acumulador com operando fonte ou destino dos dados. Na transferência de dados com a RAM externa é possível fazer dois tipos de acesso:

- Com endereçamento de 16 bits, utilizando o DPTR;
- Com endereçamento de 8 bits, utilizando @Ri, neste caso também se utiliza o porto P2 para os 8 bits mais significativos do endereço.

A tabela seguinte é referente à leitura de dados na memória externa (o X indica que se está a trabalhar com a memória de dados externa):

Instrução	Operando	Bytes	MC
MOVX A,	@Ri	1	2
	DPTR	1	2

A tabela seguinte é referente à escrita de dados na memória externa:

Instrução	Operando	Bytes	MC
MOVX @Ri,	A	1	2
MOVX @DPTR,	A	1	2

Quando se trabalha com Memória de Programa separada da Memória de Dados Externa, a primeira só pode ser lida (sinal \*PSEN). Existem duas instruções para realizar esta operação. Note-se que estas são as únicas que utilizam endereçamento indexado. Estas instruções são muito convenientes para construção de tabelas de consulta.

A instrução designa-se por MOVC onde o C indica Memória de Códigos. A tabela seguinte é referente às duas instruções de leitura da memória de programa externa.

Instrução	Operando	Bytes	MC
MOVC A,	@A+@DPTR	1	2
	@A+@PC	1	2

As instruções booleanas são aquelas que trabalham só com bits. O Carry é neste caso um dos operandos e tem a mesma funcionalidade do acumulador. Os bits são endereçados directamente. Porém o carry é endereçado implicitamente.

1. Colocar a zero, um e complementar um bit;
2. AND e OR booleanos;
3. Transferência de bits;
4. Desvios baseados em bits.

# Colocar a zero, um e complementar um bit

- Instrução CLR: Coloca o bit a zero;
- Instrução SETB: Coloca o bit a um;
- Instrução CPL: Complementa o bit.

A tabela seguinte apresenta estas funções:

Instrução	Operando	Bytes	MC	CY	AC	OV
CLR	C	1	1	0	-	-
	bit	2	1	-	-	-
SETB	C	1	1	1	-	-
	bit	2	1	-	-	-
CPL	C	1	1	X	-	-
	bit	2	1	-	-	-

Existem duas instruções AND e duas instruções OR Booleanas. Para estas instruções o carry (C) funciona como acumulador, ou seja, é o operando destino onde é colocado o resultado. Quando se utiliza a nomenclatura "/bit" significa que se está a realizar um AND ou um OR, consoante o caso, com o complemento do bit.

A tabela seguinte apresenta estas funções:

Instrução	Operando	Bytes	MC	CY	AC	OV
ANL C,	bit	2	1	X	-	-
	/bit	2	1	X	-	-
ORL C,	bit	2	1	X	-	-
	/bit	2	1	X	-	-

Existem duas instruções de transferência de bits, as quais usam sempre o acumulador C como origem ou destino.

Instrução	Operando	Bytes	MC	CY	AC	OV
MOV C,	bit	2	1	X	-	-
MOV bit,	C	2	2	-	-	-

Existem cinco instruções desvio. Duas delas utilizam o carry C e as outras três operam com qualquer bit.

- As instruções JC e JB realizam o desvio se o bit estiver activo;
- As instruções JNC e JNB realizam o desvio se o bit não está activo;
- A instrução JBC realiza o desvio se o bit está activo e depois complementa este bit.

A tabela seguinte apresenta estas funções:

Instrução	Operando	Bytes	MC	CY	AC	OV
JC	rel	2	2	X	-	-
JNC		2	2	X	-	-
JB	bit,rel	3	2	X	-	-
JNB		3	2	X	-	-
JBC		3	2	X	-	-

Exemplo:

ADD A, # 88H; A = A + 88H

JC THERE ; se o resultado de A + 88H for maior que FFH, o carry fica a um, então salta para a instrução com o label THERE.

As instruções de desvio incluem:

1. Chamada de subrotinas (CALL), estas podem ser de 11 ou 16 bits;
2. Saltos (JMP), estes podem ser de 11 ou 16 bits;
3. Saltos condicionais (JZ, JNZ e CJNE);
4. LOOPS (DJNZ);
5. Instruções de retorno (RET e RETI).



São duas as instruções de chamada de subrotinas. A instrução LCALL utiliza um endereço de 16 bits. e por isso permite o acesso a qualquer posição nos 64KB de programa. A instrução ACALL utiliza apenas 11 bits, ou seja, permite acesso apenas dentro de bloco de 2 KB da memória de programa. Neste último caso, os 11 bits menos significativos do PC são substituídos pelos 11 bits especificados na instrução. A vantagem do ACALL é que necessita apenas de 2 bytes para o OP-CODE, enquanto que a LCALL necessita de 3 bytes.

A tabela seguinte apresenta estas funções:

Instrução	Endereço	Bytes	MC	CY	AC	OV
LCALL	ADDRESS16	3	2	-	-	-
ACALL	ADDRESS11	2	2	-	-	-

Existem quatro instruções de desvio (saltos). Tal como na chamada de subrotinas, o salto AJMP é de 11 bits e permite saltos dentro de um bloco de 2 KB da memória de programa. Por outro lado, o salto de 16 bits LJMP permite saltos nos 64 KB da memória de programa. Os outros dois saltos são relativos.

A tabela seguinte apresenta estas funções:

Instrução	Endereço	Bytes	MC	CY	AC	OV
LJMP	ADDRESS16	3	2	-	-	-
AJMP	ADDRESS11	2	2	-	-	-
SJMP	REL	1	2	-	-	-
JMP	@A+DPTR	1	2	-	-	-

Exemplo:

JMP @A+DPTR; Salta para a zona de memória definida por  $A=A+DPTR$  ( $PC=A=A+DPTR$ ); ou seja, se  $A=04H$  e  $DPTR=1000H$ , então o  $PC=1004H$ . Note-se que os desvios relativos têm um alcance de 127 bytes para a frente e 128 bytes para trás.

Há dois desvios que são feitos com os valores do acumulador: JZ, que salta se o conteúdo do acumulador é zero, e JNZ, que salta se o conteúdo do acumulador é diferente de zero. Há outras 4 instruções que utilizam a notação CJNE, que significa "compare dois elementos e realize o desvio se forem diferentes". Todos estes desvios são relativos e por isso têm um alcance de 127 bytes para a frente e 128 bytes para trás.

A tabela seguinte apresenta estas funções:

Instrução	Endereço ou Operando	Bytes	MC	Condição
JZ	rel	2	2	se Acc=0
JNZ	rel	2	2	se Acc≠0
CJNE	A, Directo,REL	3	2	se Acc≠ Directo
CJNE	A ou Rn ou @Ri, #data, REL	3	2	se A ou Rn ou @Ri≠ #data

As instruções de loop são do tipo "decremente e desvie se for diferente de zero"(DJNZ). Há duas instruções deste tipo, as quais são muito úteis para a repetição de determinadas partes do programa.

Estas instruções utilizam como contador um registo ou um byte da RAM Interna e por isso têm um limite de até 256 repetições. Como são baseadas em desvios relativos, têm um alcance de 127 bytes para a frente e de 128 bytes para trás.

A tabela seguinte apresenta estas funções:

Instrução	Operando	Endereço	Bytes	MC	CY	AC	OV
DJNZ	Rn	REL	2	2	-	-	-
DJNZ	Directo	REL	2	2	-	-	-

**NOP:** É uma instrução que não faz nada, consumindo apenas ciclos da CPU. Consome 1 byte de OP-CODE e tem a duração de 1 ciclo de máquina.