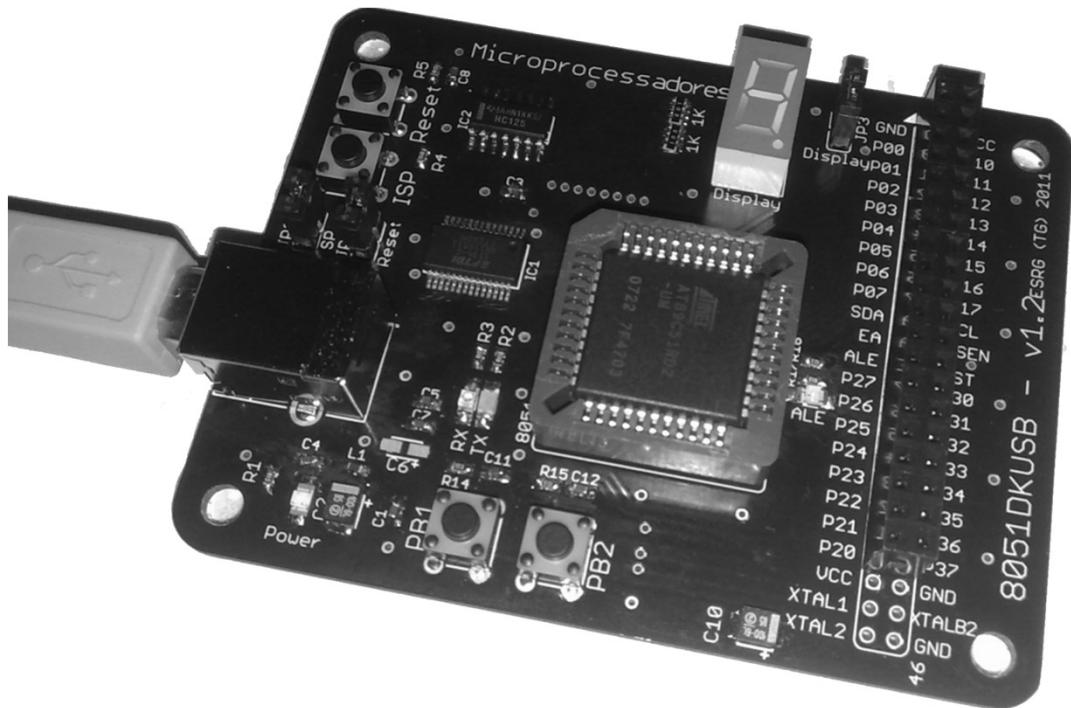
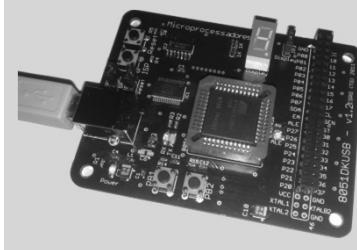


# Mestrado Integrado em Eng. Electrónica Industrial e Computadores



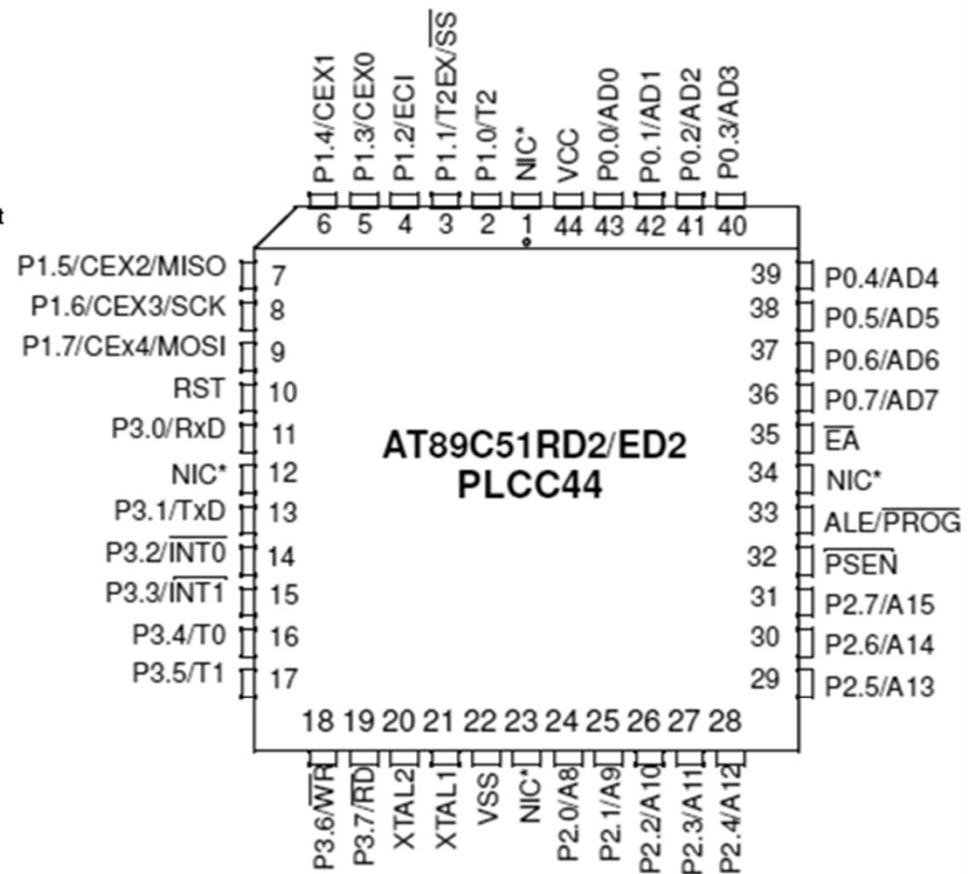
**Intel MCS-51**

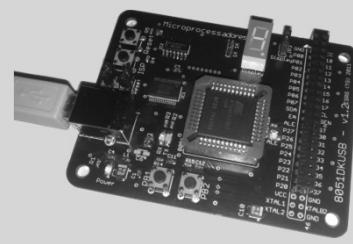
Microcontroladores  
2º Ano – A05



# AT89C51RD2

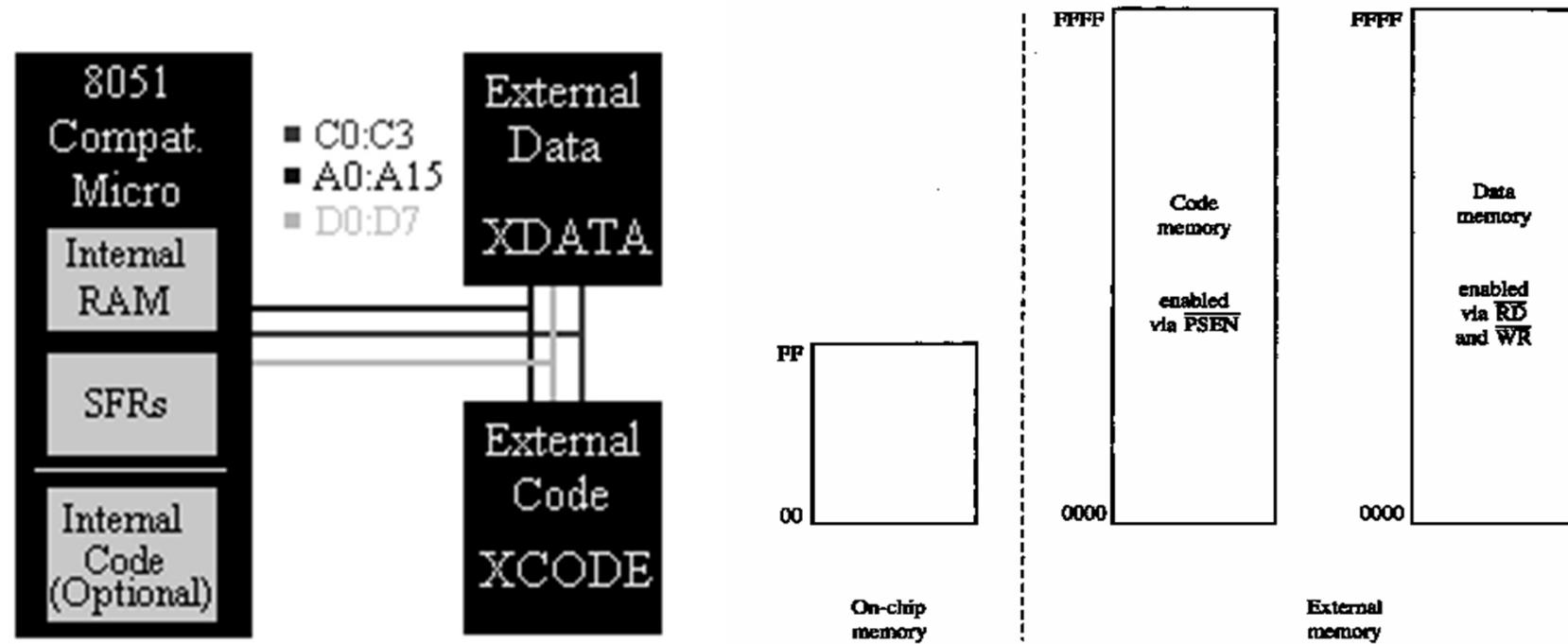
- 80C52 Compatible
  - 8051 Instruction Compatible
  - Six 8-bit I/O Ports (64 Pins or 68 Pins Versions)
  - Four 8-bit I/O Ports (44 Pins Version)
  - Three 16-bit Timer/Counters
  - 256 Bytes Scratch Pad RAM
  - 9 Interrupt Sources with 4 Priority Levels
- ISP (In-System Programming) Using Standard V<sub>cc</sub> Power Supply
- 2048 Bytes Boot ROM Contains Low Level Flash Programming Routines and a Default Serial Loader
- High-speed Architecture
  - In Standard Mode:
    - 40 MHz (Vcc 2.7V to 5.5V, both Internal and external code execution)
    - 60 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
  - In X2 mode (6 Clocks/machine cycle)
    - 20 MHz (Vcc 2.7V to 5.5V, both Internal and external code execution)
    - 30 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
- 64K Bytes On-chip Flash Program/Data Memory
  - Byte and Page (128 Bytes) Erase and Write
  - 100k Write Cycles
- On-chip 1792 bytes Expanded RAM (XRAM)
  - Software Selectable Size (0, 256, 512, 768, 1024, 1792 Bytes)
  - 768 Bytes Selected at Reset for T89C51RD2 Compatibility
- On-chip 2048 Bytes EEPROM Block for Data Storage (AT89C51ED2 Only)
  - 100K Write Cycles
- Keyboard Interrupt Interface on Port 1
- SPI Interface (Master/Slave Mode)
- 8-bit Clock Prescaler
- 16-bit Programmable Counter Array
  - High Speed Output
  - Compare/Capture
  - Pulse Width Modulator
  - Watchdog Timer Capabilities



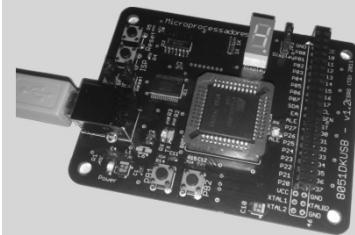


# Organização da memória

- Nesta arquitectura há dois tipos de memórias:
  - A memória de código/programa (tipo-ROM);
  - A memória de dados (tipo-RAM).
- Dependendo do microcontrolador, as memórias podem:
  - Estar implementadas internamente (dentro do *chip* CODE ou DATA);
  - Podemos implementá-las externamente usando barramentos (endereços, dados e controlo) e uma *latch* externa (XCODE e XDATA).



Espaço de memória do 8031



# Organização da memória de dados interna – IDATA/IRAM

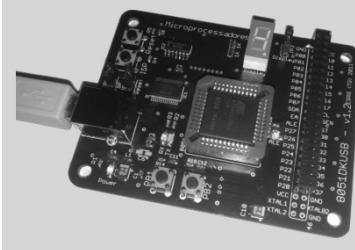
I RAM Addr	Description							
00	R0 R1 R2 R3 R4 R5 R6 R7							
08	R0 R1 R2 R3 R4 R5 R6 R7							
10	R0 R1 R2 R3 R4 R5 R6 R7							
18	R0 R1 R2 R3 R4 R5 R6 R7							
20	00 08 10 18 20 28 30 38							
28	40 48 50 58 60 68 70 78							
30	General User RAM & Stack Space (80 bytes, 30h-7Fh)							
7F	General IRAM							
80	Special Function Registers (SFRs) (80h - FFh)							
:								
:								
:								

- Os bancos de registos permitem uma forma rápida e eficiente de comutação de contexto em que parcelas de código usam um conjunto privado de registos independentemente.

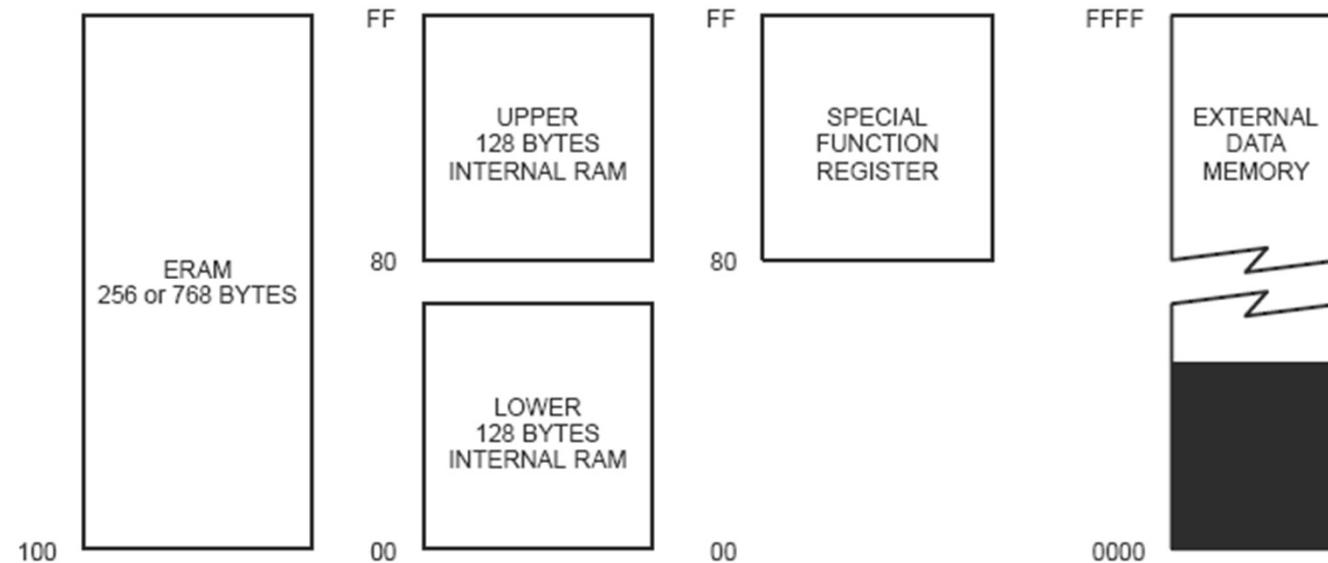
Byte address	Bit address								Byte address	Bit address								
7F	General purpose RAM								F9	F7 F6 F5 F4 F3 F2 F1 F0	B							
7E									E9	E7 E6 E5 E4 E3 E2 E1 E0	ACC							
7D									D9	D7 D6 D5 D4 D3 D2 - D0	PSW							
7C									C9	- - - BC BB BA B9 B8	IP							
7B									B9	B7 B6 B5 B4 B3 B2 B1 B0	P3							
7A									A9	AF - - AC AB AA A9 A8	IE							
79									A8	A7 A6 A3 A4 A3 A5 A3 A5	P2							
78									99	not bit addressable								SRUF
77									98	9F 9E 9D 9C 9B 9A 99 98	SCON							
76									90	97 96 95 94 93 92 91 90	P1							
75									8D	not bit addressable								TH1
74									8C	not bit addressable								TH0
73									8B	not bit addressable								TLL1
72									8A	not bit addressable								TLO
71									89	not bit addressable								TMOD
70									88	8F 8E 8D 8C 8B 8A 89 88	TCON							
6F									87	not bit addressable								PCON
6E									83	not bit addressable								DPH
6D									82	not bit addressable								DPL
6C									81	not bit addressable								SP
6B									80	87 86 85 84 83 82 81 80	PO							

RAM

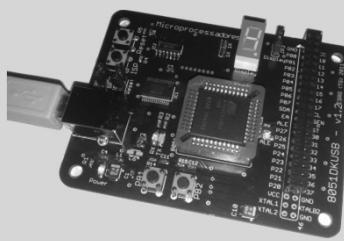
SPECIAL FUNCTION REGISTERS



# Organização da memória



1. Os primeiros 128 bytes da RAM interna podem ser acedidos directa ou indirectamente;
2. Os SFR's só podem ser acedidos por endereçamento directo;
3. Os segundos 128 bytes da RAM interna só podem ser acedidos indirectamente;
4. Os 768 bytes da RAM expandida (ERAM 00h-2FFh) são acedidos indirectamente pela instrução de MOVX e com o bit EXTRAM a zero.



# SFR – Special Function Registers

Table 1. Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE	
			MSB	E7	E6	E5	E4	E3	E2	E1	LSB	
ACC*	Accumulator	E0H										00H
AUXR#	Auxiliary	8EH		–	–	–	–	–	–	EXTRAM	AO	xxxxxx10B
AUXR1#	Auxiliary 1	A2H		–	–	ENBOOT	–	GF2	0	–	DPS	xxxxxx0B
B*	B register	F0H		F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH										xxxxxxxxB
CCAP1H#	Module 1 Capture High	FBH										xxxxxxxxB
CCAP2H#	Module 2 Capture High	FCH										xxxxxxxxB
CCAP3H#	Module 3 Capture High	FDH										xxxxxxxxB
CCAP4H#	Module 4 Capture High	FEH										xxxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH										xxxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH										xxxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH										xxxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH										xxxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH										xxxxxxxxB
CCAPM0#	Module 0 Mode	DAH		–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH		–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH		–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH		–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH		–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCON**#	PCA Counter Control	D8H		DF	DE	DD	DC	DB	DA	D9	D8	00x00000B
CH#	PCA Counter High	F9H		CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0	00H
CL#	PCA Counter Low	E9H										00H
CMOD#	PCA Counter Mode	D9H		CIDL	WDTE	–	–	–	CPS1	CPS0	ECF	0x000000B
DPTR:	Data Pointer (2 bytes)	83H										00H
DPH	Data Pointer High	82H										00H
DPL	Data Pointer Low											00H
IE*	Interrupt Enable 0	A8H		AF	AE	AD	AC	AB	AA	A9	A8	00H
				EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
				BF	BE	BD	BC	BB	BA	B9	B8	00H
IP*	Interrupt Priority	B8H		–	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
				B7	B6	B5	B4	B3	B2	B1	B0	x0000000B
IPH#	Interrupt Priority High	B7H		–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	
				87	86	85	84	83	82	81	80	
P0*	Port 0	80H		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
				97	96	95	94	93	92	91	90	FFH
P1*	Port 1	90H		CEX4	CEX3	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
				A7	A6	A5	A4	A3	A2	A1	A0	FFH
P2*	Port 2	A0H		AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
				B7	B6	B5	B4	B3	B2	B1	B0	FFH
P3*	Port 3	B0H		RD	WR	T1	T0	INTT	INT0	TxD	RxD	FFH
PCON#1	Power Control	87H		SMOD1	SMOD0	–	–	GF1	GF0	PD	IDL	00xxx000B

Table 1. Special Function Registers (Continued)

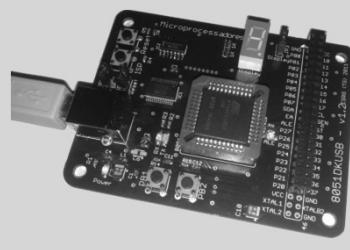
SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE	
			MSB	D7	D6	D5	D4	D3	D2	D1	LSB	
PSW*	Program Status Word	D0H		CY	AC	F0	RS1	RS0	OV	F1	P	0000000B
RCAP2H#	Timer 2 Capture High	CBH										00H
RCAP2L#	Timer 2 Capture Low	CAH										00H
SADDR#	Slave Address	A9H										00H
SADEN#	Slave Address Mask	B9H										00H
SBUF	Serial Data Buffer	99H		9F	9E	9D	9C	9B	9A	99	98	xxxxxxxxB
SCON*	Serial Control	98H		SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H										07H
TCON*	Timer Control	88H		8F	8E	8D	8C	8B	8A	89	88	00H
				TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
				CF	CE	CD	CC	CB	CA	C9	C8	00H
T2CON*	Timer 2 Control	C8H		TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/R/L2	00H
T2MOD#	Timer 2 Mode Control	C9H		–	–	–	–	–	–	T2OE	DCEN	00H
TH0	Timer High 0	8CH										00H
TH1	Timer High 1	8DH										00H
TH2#	Timer High 2	CDH										00H
TL0	Timer Low 0	8AH										00H
TL1	Timer Low 1	8BH										00H
TL2#	Timer Low 2	CCH										00H
TMOD	Timer Mode	89H		GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDTRST	Watchdog Timer Reset	A6H										

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

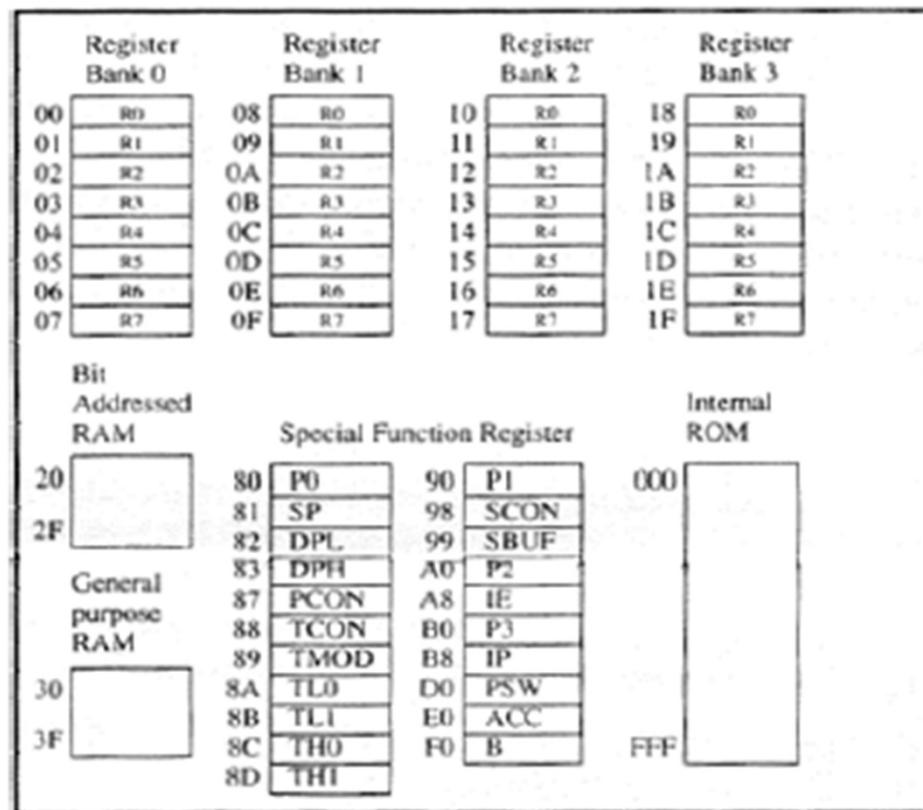
- Reserved bits.

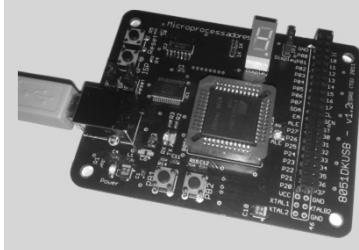
**P89C51RD2**



# Modelo de Programação

## 80C51 Program Model - Calcutt

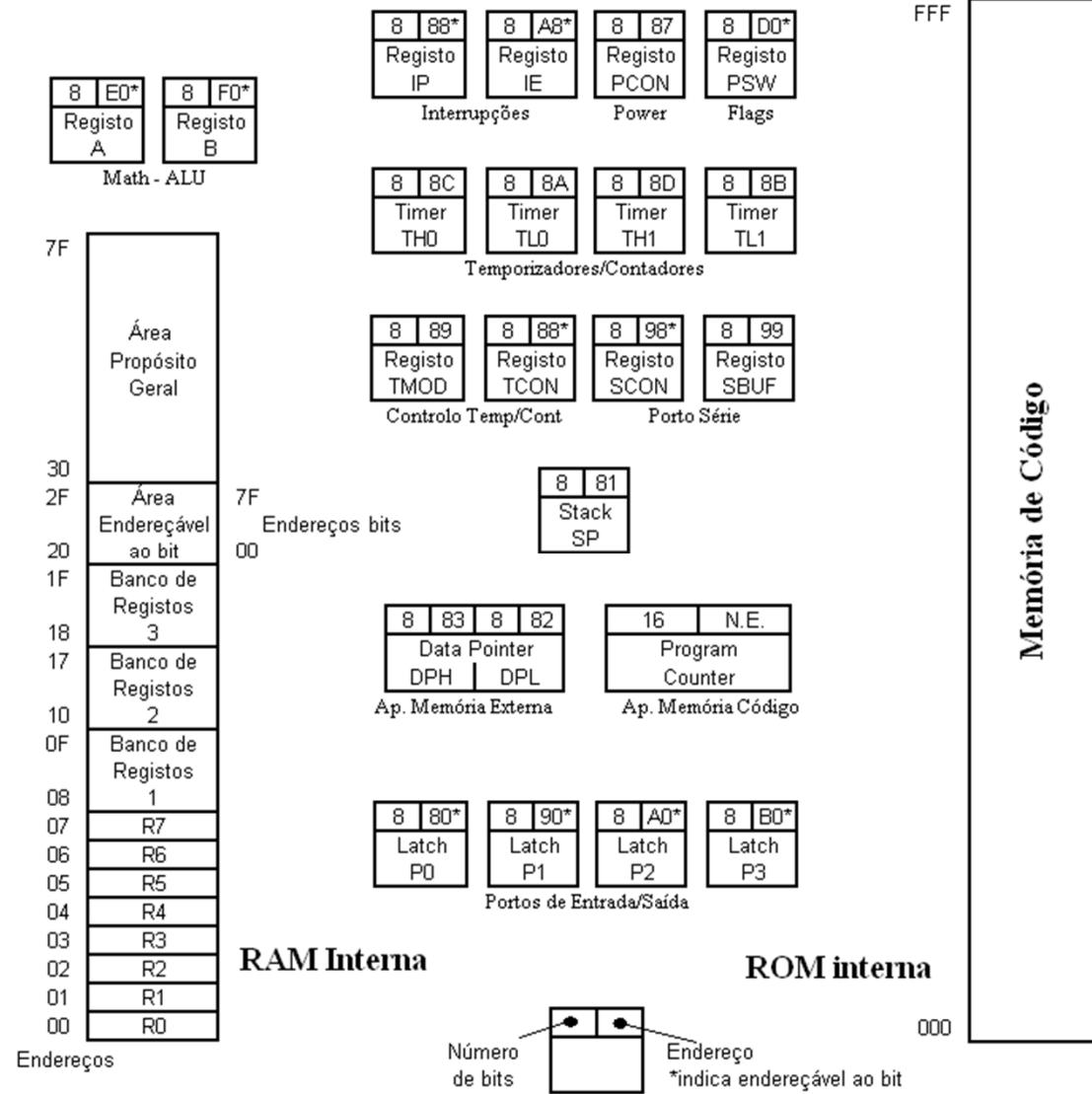


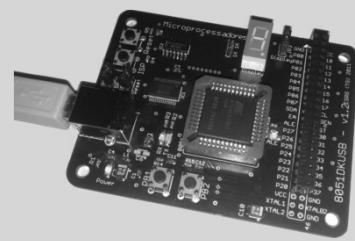


# Modelo de Programação

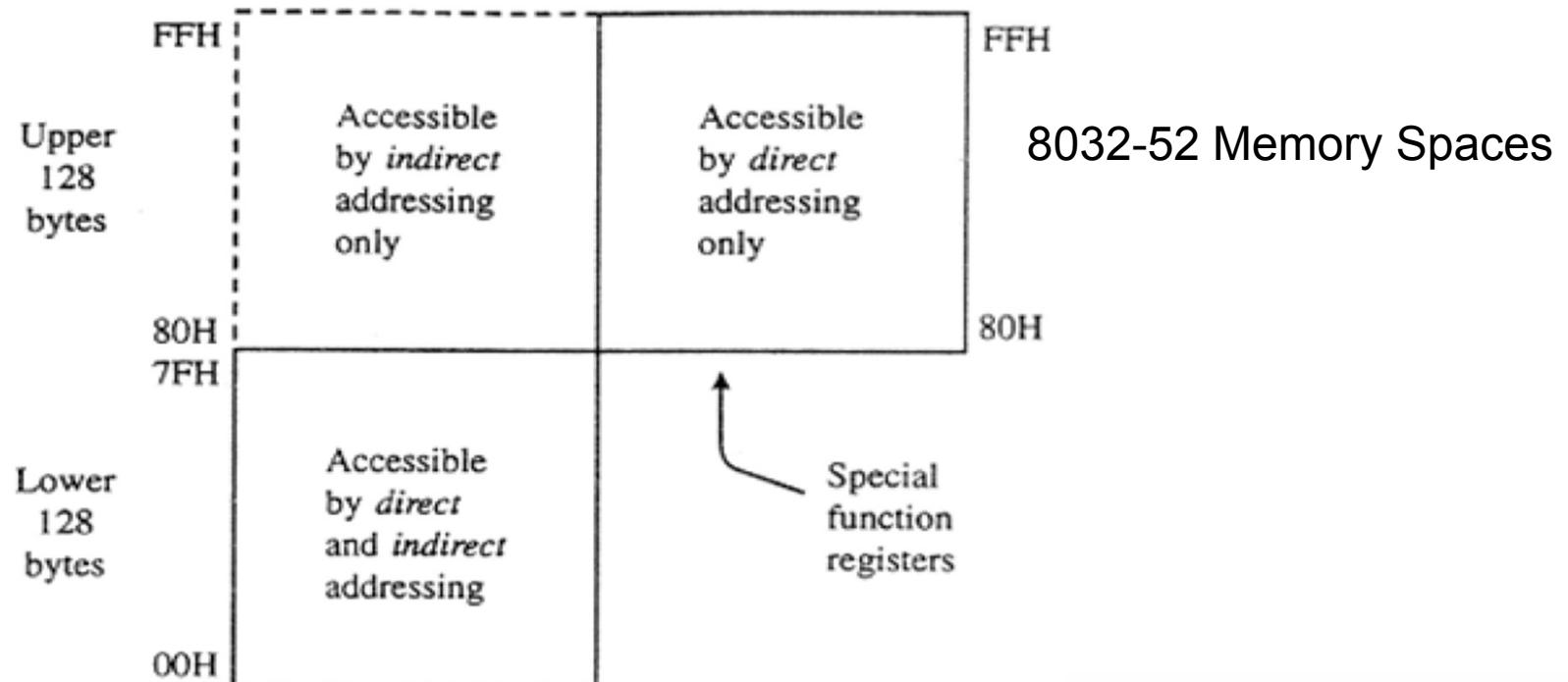
Adaptado de:

**AYALA**  
**“8051**  
*Microcontroller  
Architecture  
Programming  
and Applications”*



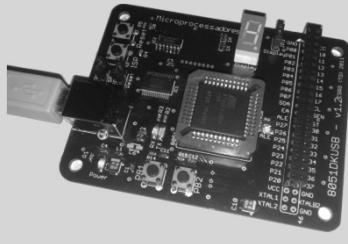


# Família i8052



8032-52 Memory Spaces (MacKenzie)

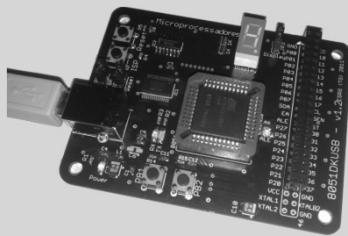
REGISTER	ADDRESS	DESCRIPTION	BIT-ADDRESSABLE
T2CON	C8H	Control	Yes
RCAP2L	CAH	Low-byte capture	No
RCAP2H	CBH	High-byte capture	No
TL2	CCH	Timer 2 low-byte	No
TH2	CDH	Timer 2 high-byte	No



# Estado do microcontrolador após *reset*

REGISTER(S)	CONTENTS
Program counter	0000H
Accumulator	00H
B register	00H
PSW	00H
SP	07H
DPTR	0000H
Ports 0-3	FFH
IP (8031/8051)	XXX00000B
IP (8032/8052)	XX000000B
IE (8031/8051)	0XX00000B
IE (8032/8052)	0X000000B
Timer registers	00H
SCON	00H
SBUF	00H
PCON (HMO8)	0XXXXXXXB
PCON (CMOS)	0X0X0000B

Registo(s)	Conteúdo
Program Counter	0000H
Accumulator, B register, PSW	00H
SP	07H
DPTR	0000H
Ports 0-3	FFH
IP	XXX00000B
Timer registers, SCON	00H



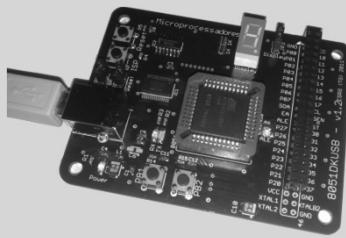
# Conjunto de Instruções

## Notas sobre o conjunto de instruções e modos de endereçamento

<b>Rn</b>	Registo (R0 a R7) do banco de registos seleccionado
<b>direct</b>	Endereço directo de 8-bit de uma posição da memória interna de dados. Pode-se tratar da RAM de dados interna (0-127) ou de um SFR (porto I/O, reg de estado, etc (128-255))
<b>@Ri</b>	Endereço de 8-bit de uma posição da RAM interna, endereçada indirectamente através de R0 ou R1.
<b>#data</b>	Constante de 8-bit incluída na instrução
<b>#data16</b>	Constante de 16-bit incluída na instrução
<b>addr16</b>	Endereço de destino de 16-bit. Usado por LCALL e LJMP. O salto pode ser para qualquer localização da ROM.
<b>addr11</b>	Endereço de destino de 11-bit. Usado por ACALL e AJMP. O salto pode ser para qualquer localização dentro da página de 2KB da ROM.
<b>rel</b>	Offset de destino de 8-bit com sinal. Usado por SJMP e todos os saltos condicionais. O alcance vai de -128 a 127 bytes relativamente ao endereço da próxima instrução.
<b>bit</b>	Bit da RAM de dados interna ou do SFR

## Instruções que afectam as flags da ALU

Instrução	Flags			Instrução	Flags			
	C	O	V		AC	C	O	V
ADD	X	X	X	CLRC	0			
ADDC	X	X	X	CPLC	X			
SUBB	X	X	X	ANLC,bit	X			
MUL	0	X		ANLC,/bit	X			
DIV	0	X		ORLC,bit	X			
DA	X			ORLC,/bit	X			
RRC	X			MOVC,bit	X			
RLC	X			CJNE	X			
SETBC	1							

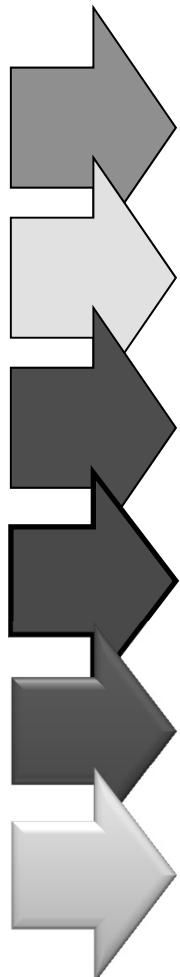


# Conjunto de Instruções

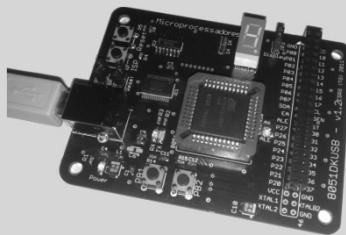
**Mneumónica:** Nome da instrução assembly

**Tamanho:** N° de bytes que a mneumónica ocupa na memória de programa

**Ciclos:** N° de ciclos máquina necessários para efectuar a operação. Um ciclo=12 impulsos de relógio



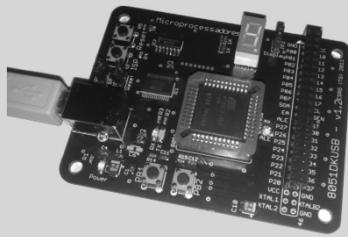
Operações Aritméticas				
Mneumónica	Tamanho	Ciclos	Descrição	
ADD A,Rn	1	1	Somar ao Acumulador o registo	
ADD A,direct	2	1	Somar ao Acumulador o endereço directo de RAM	
ADD A,@Ri	1	1	Somar ao Acumulador o endereço indirecto de RAM	
ADD A,#data	2	1	Somar ao Acumulador o byte de dados directo	
ADDC A,Rn	1	1	Somar ao Acumulador o registo com carry	
ADDC A,direct	2	1	Somar ao Acumulador o endereço directo de RAM com carry	
ADDC A,@Ri	1	1	Somar ao Acumulador o endereço indirecto de RAM com carry	
ADDC A,#data	2	1	Somar ao Acumulador o byte de dados directo com carry	
SUBB A,Rn	1	1	Subtrair ao Acumulador o registo com borrow	
SUBB A,direct	2	1	Subtrair ao Acumulador o endereço directo de RAM com borrow	
SUBB A,@Ri	1	1	Subtrair ao Acumulador end. indirecto de RAM com borrow	
SUBB A,#data	2	1	Subtrair ao Acumulador o byte de dados directo com borrow	
INC A	1	1	Incrementar Acumulador	
INC Rn	1	1	Incrementar Registo	
INC direct	2	1	Incrementar endereço directo de RAM	
INC @Ri	1	1	Incrementar endereço indirecto de RAM	
DEC A	1	1	Decrementar Acumulador	
DEC Rn	1	1	Decrementar registo	
DEC direct	2	1	Decrementar endereço directo de RAM	
DEC @Ri	1	1	Decrementar endereço indirecto de RAM	
INC DPTR	1	2	Incrementar DPTR (apontador para dados)	
MUL AB	1	4	Multiplicar A por B	
DIV AB	1	4	Dividir A por B	
DA A	1	1	Ajuste decimal ao Acumulador	



# Conjunto de Instruções



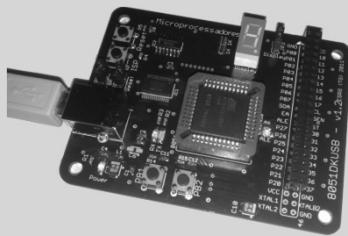
Operações Lógicas				
Mneumónica	Tamanho	Ciclos	Descrição	
ANL A,Rn	1	1	AND do Acumulador com registo	
ANL A,direct	2	1	AND do Acumulador com o endereço directo de RAM	
ANL A,@Ri	1	1	AND do Acumulador com endereço indirecto de RAM	
ANL A,#data	2	1	AND do Acumulador com o byte de dados directo	
ANL direct,A	2	1	AND do endereço directo de RAM com o Acumulador	
ANL direct,#data	3	2	AND do endereço directo de RAM com o byte de dados	
ORL A,Rn	1	1	OR do Acumulador com registo	
ORL A,direct	2	1	OR do Acumulador com o endereço directo de RAM	
ORL A,@Ri	1	1	OR do Acumulador com endereço indirecto de RAM	
ORL A,#data	2	1	OR do Acumulador com o byte de dados directo	
ORL direct,A	2	1	OR do endereço directo de RAM com o Acumulador	
ORL direct,#data	3	2	OR do endereço directo de RAM com o byte de dados	
XRL A,Rn	1	1	XOR do Acumulador com registo	
XRL A,direct	2	1	XOR do Acumulador com o endereço directo de RAM	
XRL A,@Ri	1	1	XOR do Acumulador com endereço indirecto de RAM	
XRL A,#data	2	1	XOR do Acumulador com o byte de dados directo	
XRL direct,A	2	1	XOR do endereço directo de RAM com o Acumulador	
XRL direct,#data	3	2	XOR do endereço directo de RAM com o byte de dados	
CLR A	1	1	Limpar Acumulador	
CPL A	1	1	Complementar Acumulador	
RL A	1	1	Rodar à esquerda Acumulador	
RLC A	1	1	Rodar à esquerda Acumulador através do carry	
RR A	1	1	Rodar à direita Acumulador	
RRC A	1	1	Rodar à direita Acumulador através do carry	
SWAP A	1	1	Trocar os nibbles do Acumulador	



# Conjunto de Instruções

Operações de Transferência de Dados

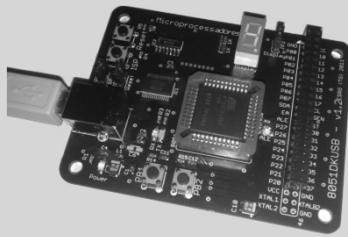
Mneumónica	Tamanho	Ciclos	Descrição
MOV A,Rn	1	1	Mover para o Acumulador do registo
MOV A,direct	2	1	Mover para o Acumulador do endereço directo de RAM
MOV A,@Ri	1	1	Mover para o Acumulador do endereço indirecto de RAM
MOV A,#data	2	1	Mover para o Acumulador do byte de dados directo
MOV Rn,A	1	1	Mover para o registo do Acumulador
MOV Rn,direct	2	2	Mover para o registo do endereço directo de RAM
MOV Rn,#data	2	1	Mover para o registo do byte de dados directo
MOV direct,A	2	1	Mover para o endereço directo de RAM do Acumulador
MOV direct,Rn	2	2	Mover para o endereço directo de RAM do Registo
MOV direct,direct	3	2	Mover para o endereço directo de RAM do end. directo de RAM
MOV direct,@Ri	2	2	Mover para o endereço directo de RAM do end. indirecto de RAM
MOV direct,#data	3	2	Mover para o endereço directo de RAM do byte de dados directo
MOV @Ri,A	1	1	Mover para o endereço indirecto de RAM do Acumulador
MOV @Ri,direct	2	2	Mover para o endereço indirecto de RAM do end. directo de RAM
MOV @Ri,#data	2	1	Mover para o endereço indirecto de RAM do byte de dados directo
MOV DPTR,#data16	3	2	Mover para o DPTR dois bytes de dados
MOVC A,@A+DPTR	1	2	Mover para o Acumulador o byte de código relativo a DPTR
MOVC A,@A+PC	1	2	Mover para o Acumulador o byte de código relativo a PC
MOVX A,@Ri	1	2	Mover para o Acumulador o endereço indirecto (8-bit) de XRAM
MOVX A,@DPTR	1	2	Mover para o Acumulador o endereço indirecto (16-bit) de XRAM
MOVX @Ri,A	1	2	Mover para o endereço indirecto (8-bit) de XRAM o Acumulador
MOVX @DPTR,A	1	2	Mover para o endereço indirecto (16-bit) de XRAM o Acumulador
PUSH direct	2	2	Colocar (Push) na stack do endereço directo de RAM
POP direct	2	2	Retirar (Pop) da stack do endereço directo de RAM
XCH A,Rn	1	1	Trocar entre Acumulador e registo
XCH A,direct	2	1	Trocar entre Acumulador e endereço directo de RAM
XCH A,@Ri	1	1	Trocar entre Acumulador e endereço indirecto de RAM
XCHD A,@Ri	1	1	Trocar dígito menor entre Acumulador e endereço indirecto de RAM



# Conjunto de Instruções

Operações de Manipulação sobre booleanos

Mneumónica	Tamanho	Ciclos	Descrição
CLR C	1	1	Limpar carry
CLR bit	2	1	Limpar bit directo da RAM
SETB C	1	1	Activar carry
SETB bit	2	1	Activar bit directo da RAM
CPL C	1	1	Complementar carry
CPL bit	2	1	Complementar bit directo da RAM
ANL C,bit	2	2	AND do carry com o bit directo da RAM
ANL C,/bit	2	2	AND do carry com o complemento do bit directo da RAM
ORL C,bit	2	2	OR do carry com o bit directo da RAM
ORL C,/bit	2	2	OR do carry com o complemento do bit directo da RAM
MOV C,bit	2	1	Mover para o carry o bit directo da RAM
MOV bit,C	2	2	Mover para o bit directo da RAM o carry
JC rel	2	2	Saltar (para end. relativo) se o carry estiver activo
JNC rel	2	2	Saltar se o carry estiver limpo
JB bit,rel	3	2	Saltar se o bit directo da RAM estiver activo
JNB bit,rel	3	2	Saltar se o bit directo da RAM estiver limpo
JBC bit,rel	3	2	Saltar se o bit directo da RAM estiver activo e limpar bit



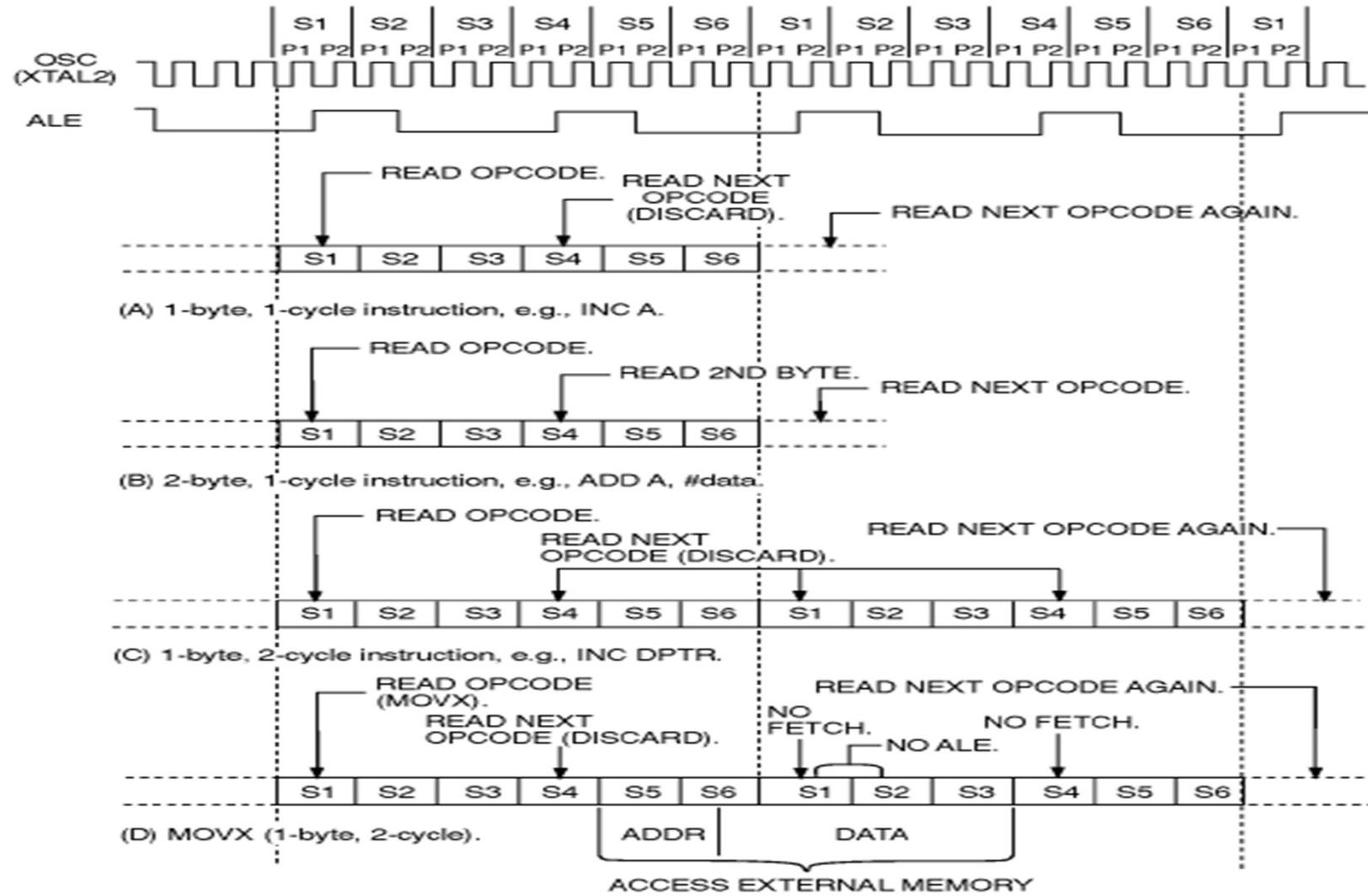
# Conjunto de Instruções

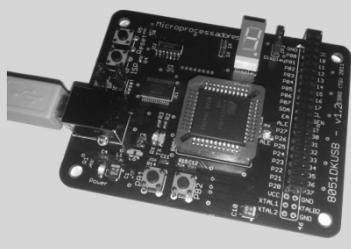
## Operações de Salto na Execução

Mneumónica	Tamanho	Ciclos	Descrição
ACALL	addr11	2	Invocação end. absoluto de subrotina
LCALL	addr16	3	Invocação end. longo de subrotina
RET		1	Retorno de subrotina
RETI		1	Retorno de interrupção
AJMP	addr11	2	Salto para endereço absoluto
LJMP	addr16	3	Salto para endereço longo
SJMP	rel	2	Salto curto para endereço relativo
JMP	@A+DPTR	1	Salto indirecto relativo a DPTR
JZ	rel	2	Salto (para end. relativo) se acumulador for zero
JNZ	rel	2	Salto se acumulador não for zero
CJNE	A,direct,rel	3	Comparar A com end. directo de RAM e saltar se diferente
CJNE	A,#data,rel	3	Comparar A com byte de dados directo e saltar se diferente
CJNE	Rn,#data,rel	3	Comparar reg. com byte de dados directo e saltar se diferente
CJNE	@Ri,#data,rel	3	Comparar end. indirecto de RAM com byte de dados e saltar se diferente
DJNZ	Rn,rel	2	Decrementar registo e saltar se registo não for zero
DJNZ	direct,rel	3	Decrementar end. Directo de RAM e saltar senão for zero
NOP		1	Nenhuma operação



# *Fetch do opcode*

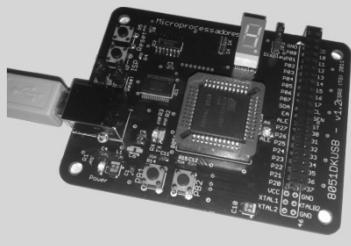




# Directivas assembly básicas

- **CSEG AT X**
  - coloca a próxima instrução no endereço X da memória de código/programa (ROM);
  - ex: CSEG AT 0H
- **END**
  - Indica ao assembler que o ficheiro fonte terminou.
- **“Etiquetas”**
  - Em vez de calcularmos o endereço de cada salto, podemos utilizar etiquetas ou *labels* para marcar esses endereços:
  - ex:

```
CSEG AT 0H
JMP MAIN ;dependendo da distância do salto, o assembler escolhe a instrução de salto ideal
...
MAIN:
    MOV R0,#25
```



# Características eléctricas

- Portos de Entrada/Saída

$V_{OL}$  – tensão de saída nível lógico baixo

	$V_{CC} = 4.5V \text{ to } 5.5V$
0.3 V	$I_{OL} = 100 \mu A^{(4)}$
0.45 V	$I_{OL} = 1.6 mA^{(4)}$
1.0 V	$I_{OL} = 3.5 mA^{(4)}$

	$V_{CC} = 4.5V \text{ to } 5.5V$
	$I_{OL} = 200 \mu A^{(4)}$
	$I_{OL} = 3.2 mA^{(4)}$
	$I_{OL} = 7.0 mA^{(4)}$

$V_{OH}$  – tensão de saída nível lógico alto

	$V_{CC} = 5V \pm 10\%$
$V_{CC} - 0.3 V$	$I_{OH} = -10 \mu A$
$V_{CC} - 0.7 V$	$I_{OH} = -30 \mu A$
$V_{CC} - 1.5 V$	$I_{OH} = -60 \mu A$

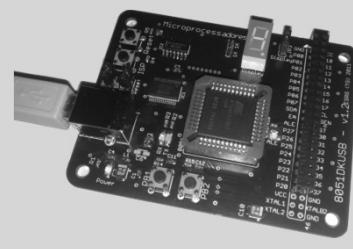
	$V_{CC} = 5V \pm 10\%$
	$I_{OH} = -200 \mu A$
	$I_{OH} = -3.2 mA$
	$I_{OH} = -7.0 mA$

Em condições de regime permanente (não transitórias),  $I_{OL}$  deve ser externamente limitada de modo a garantir:

$I_{OL}$  máxima por pino de porto: 10mA

$I_{OL}$  máxima por porto (8-bit): 15mA (P1, P2 e P3) e 26mA (P0)

$I_{OL}$  total máxima para todos os pinos de saída: 71mA

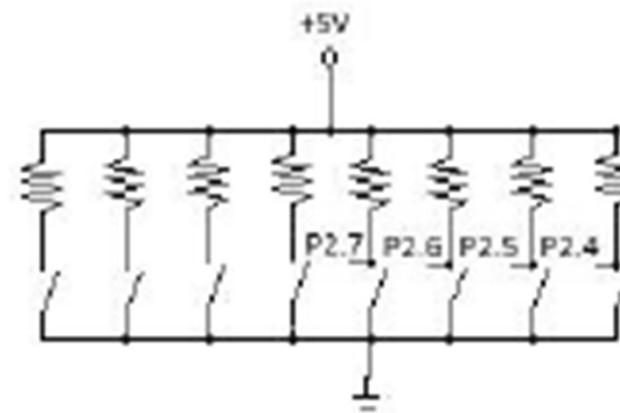
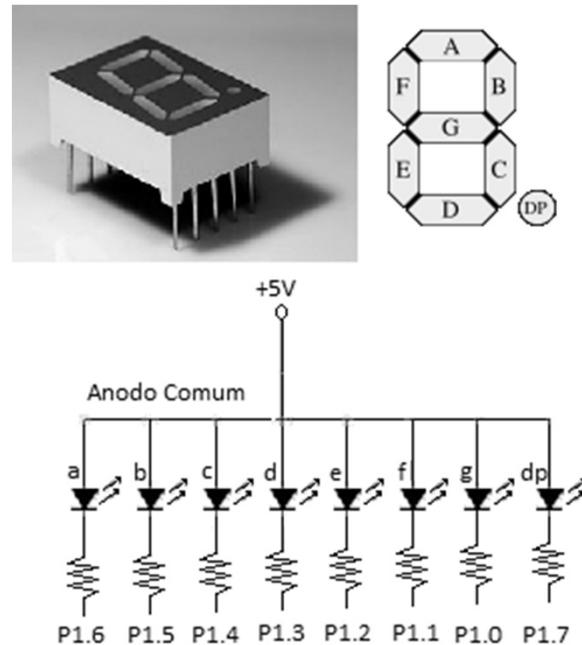


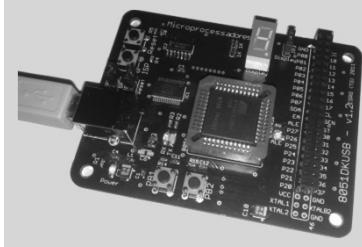
# Laboratórios - Interface

- **Problema:**

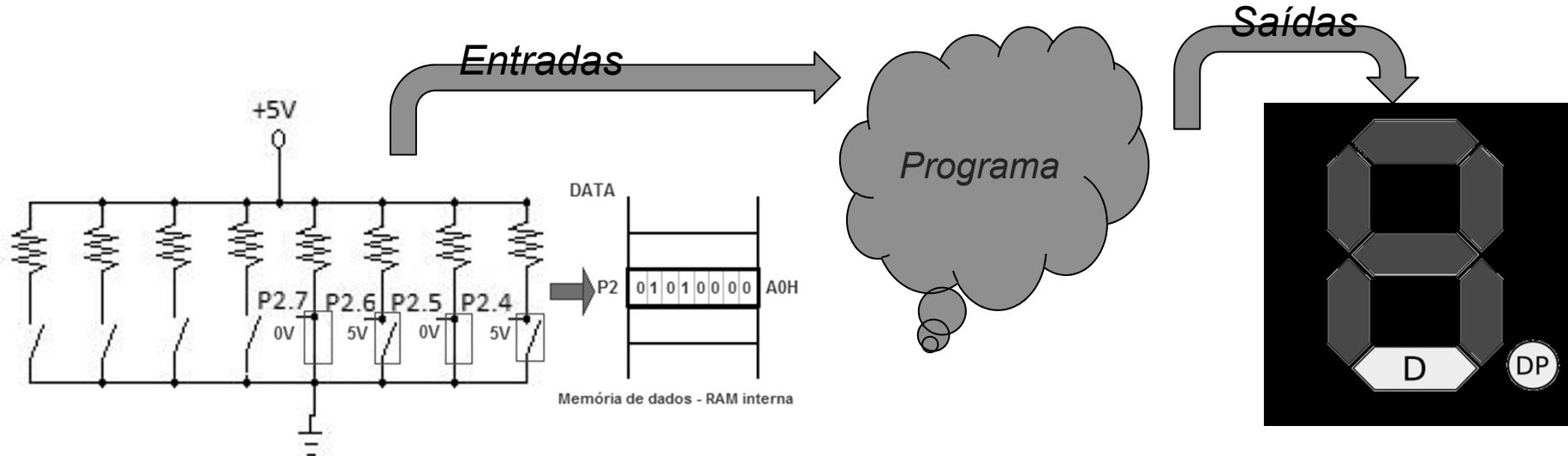
- Com base no valor de 4 pinos de entrada do porto 2 (P2.4 a P2.7), ou seja, do *nibble* (4-bit) mais significativo de P2, escrever no *display* de 7-segmentos o caracter hexadecimal correspondente ao valor do *nibble*.

- **Hardware:**

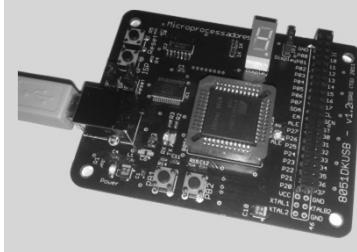




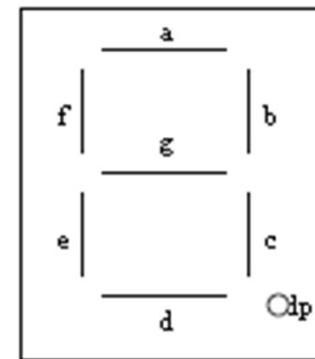
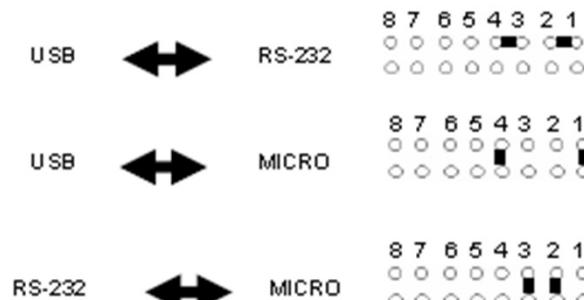
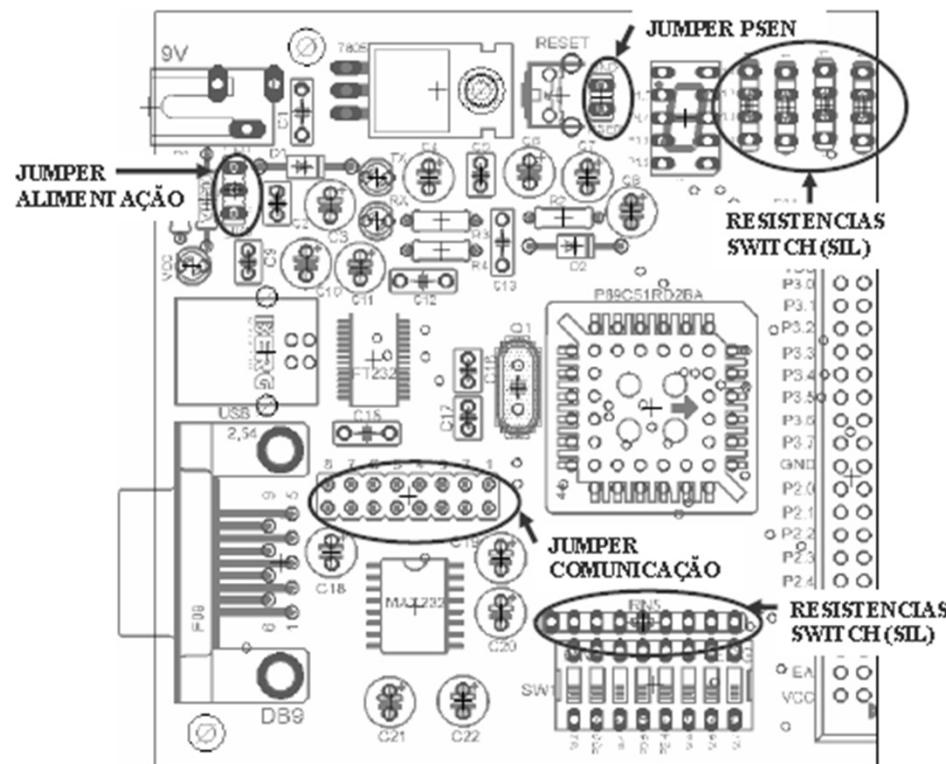
# Exemplo



- O microcontrolador automaticamente coloca no endereço A0H da memória de dados interna, a representação binária das tensões lidas nos pinos de entrada P2.7 a P2.4;
- No exemplo os bits P2.7 a P2.4 foram definidos como entradas digitais e os bits P2.3 a P2.0 como saídas digitais, usando a instrução: MOV P2,#0F0H;
- Reparar que os bits foram lidos em lógica negativa devido à configuração do hardware;
- O programa lê os bits do porto E/S para o acumulador: MOV A,P2
- O programa realiza operações sobre o acumulador;
- Após as verificações e conversões o programa tem no acumulador os bits (bit a 0 LED liga, bit a 1 LED desliga) a colocar nos pinos do porto de E/S (P1);
- MOV P1,A ;acendem-se os segmentos desejados.



# KIT8051- Esquema de Ligações



a	P1.3
b	P1.2
c	P1.0
d	P1.6
e	P1.5
f	P1.7
g	P1.4
dp	P1.1

*Escrever 0 em P1.3 liga o segmento a.  
Escrever 1 desliga*