

# Arquitetura de Computadores

---

PROF. ISAAC

# Entradas e Saídas digitais

---

Os microcontroladores possuem uma série de pinos, sendo que boa parte deles possuem como função principal a **entrada** e/ou **saída** de dados **digitais**.



# Entradas e Saídas digitais

---

Em linhas gerais, dados digitais são informações processadas por dispositivos digitais e possuem apenas dois valores:

- Zero (0): correspondente à tensão de 0 V
- Um (1): correspondente, em geral, à tensão de 5 V.

# Entradas e Saídas digitais

---

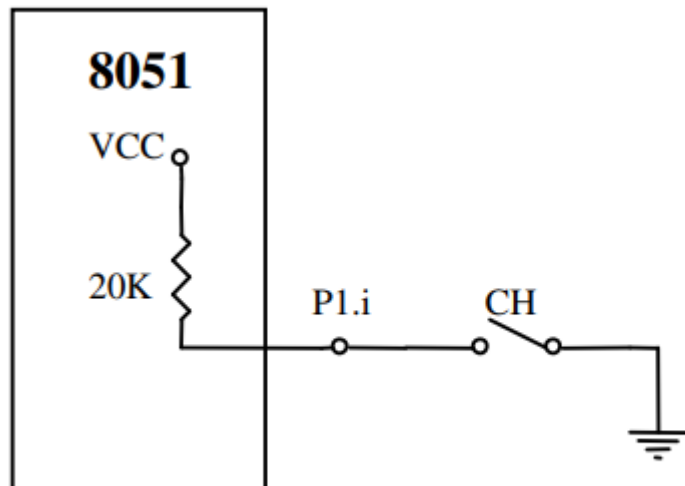
Deste modo, nos pinos de entrada e saída digitais de um microcontrolador podem existir apenas dois níveis de tensão entrando ou saindo deles: 0 V ou 5 V.

- **Nível lógico Baixo (LOW) = 0V**
- **Nível lógico Alto (HIGH) = 5V**

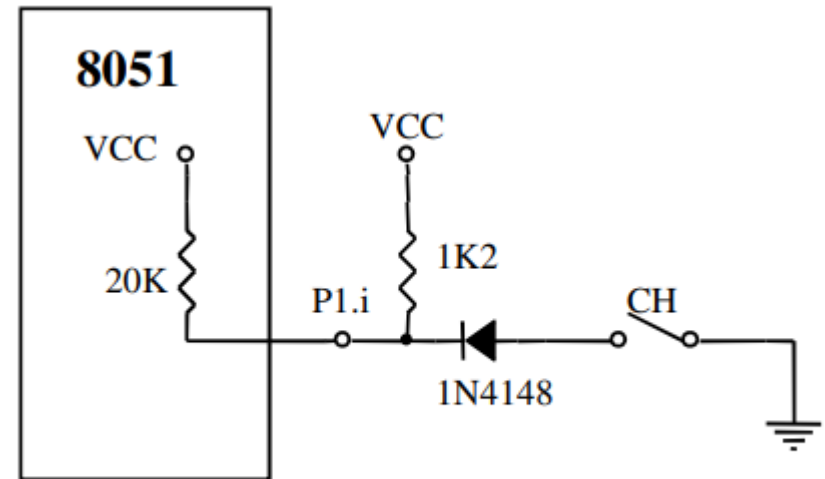
# Exemplo de entradas digitais

A adição de um resistor de externo reduz os rebotes.

A presença do diodo evita que rebotes da chave gerem uma tensão elevada sobre o pino da porta paralela.



(a) Chave ligada diretamente.

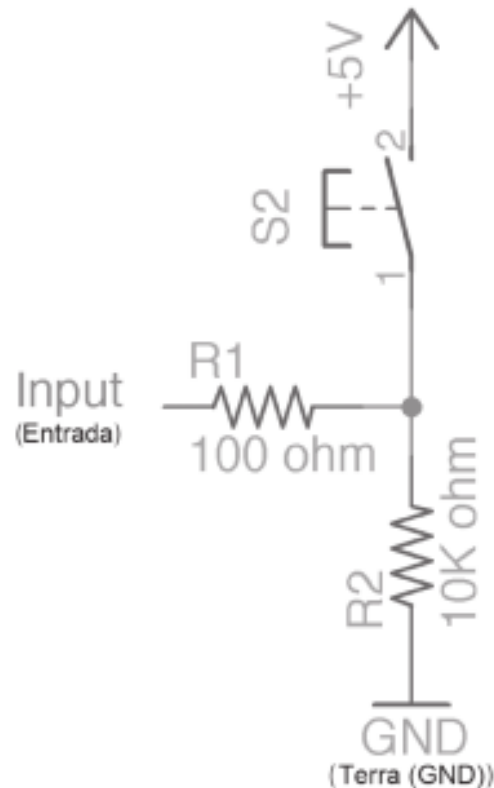


(b) Chave com proteção.

# Exemplo de entradas digitais

## Resistores pull-down:

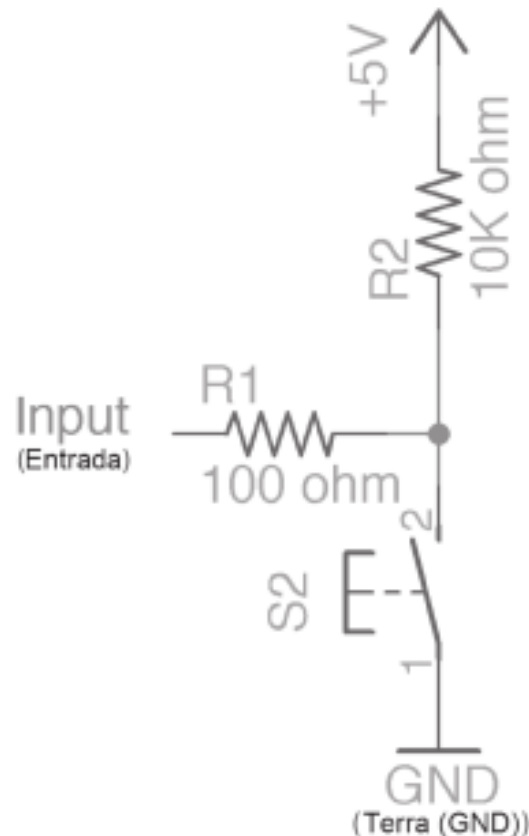
- A entrada fica sempre LOW (0V) enquanto o botão não for apertado.



# Exemplo de entradas digitais

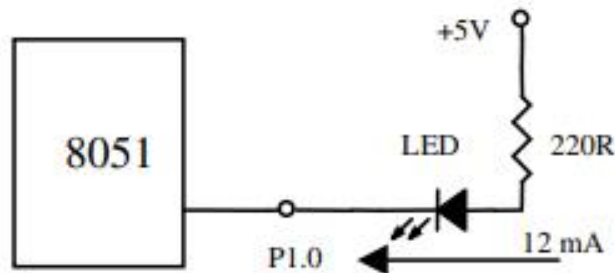
## Resistores pull-up:

- A entrada fica sempre HIGH (5V) enquanto o botão não for apertado.

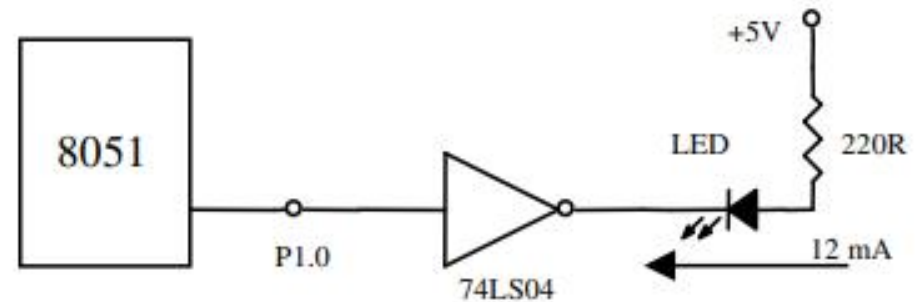


# Exemplo de saídas digitais

Quando utilizamos o microcontrolador com saídas digitais, devemos evitar que a corrente elétrica exigida no pino de saída não seja acima da corrente máxima do microcontrolador.



(a) Excesso de corrente em P1.0

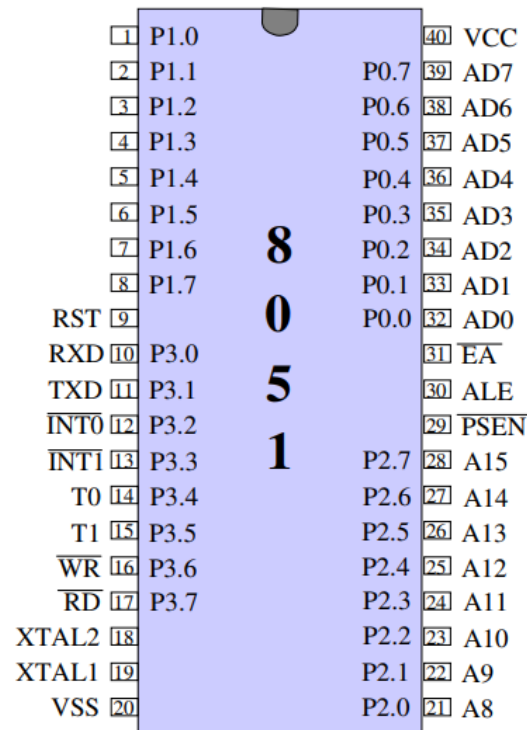


(b) TTL recebe a corrente de 12 mA.



# Portas do 8051

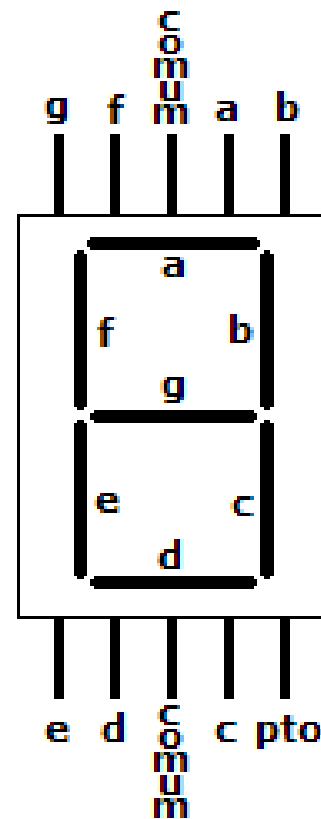
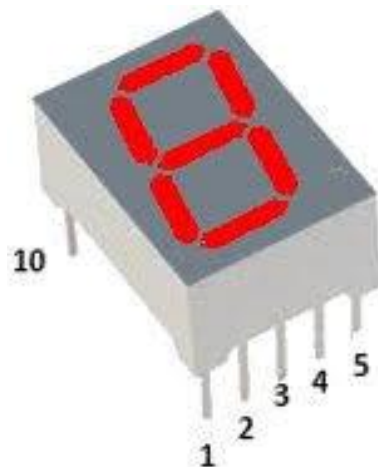
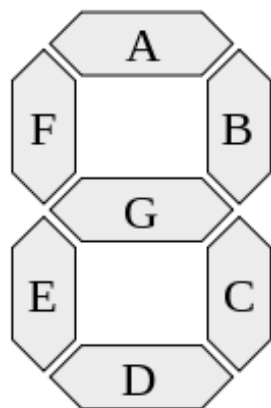
No 8051, estão disponíveis quatro portas paralelas, denominadas portas P0, P1, P2 e P3. Cada bit dessas portas corresponde a um pino do microcontrolador.



# Display de 7 Segmentos

---

# Display de 7 segmentos



Display anodo comum

# Decodificador para display de 7 segmentos

- O **Decodificador** recebe um código binário (BCD 8421) e escreve a sequencia de 0 a 9 em um display de 7 segmentos catodo comum.

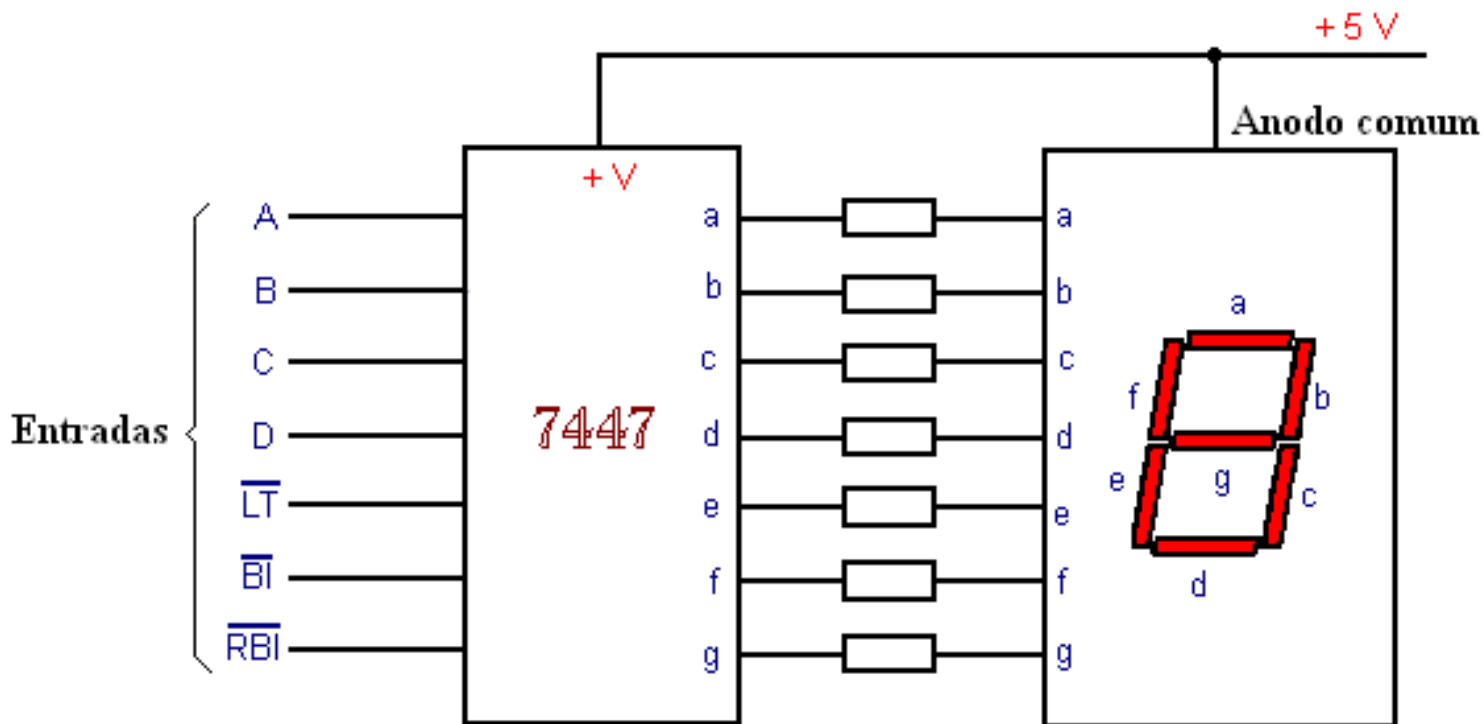


Figura 1 - Display de 7 segmentos comandado por um decodificador 7447.

# Decodificador para display de 7 segmentos

- Na disciplina de Sistemas Digitais vocês utilizaram o decodificador **74247** no projeto desenvolvido no programa Quartus Prime.

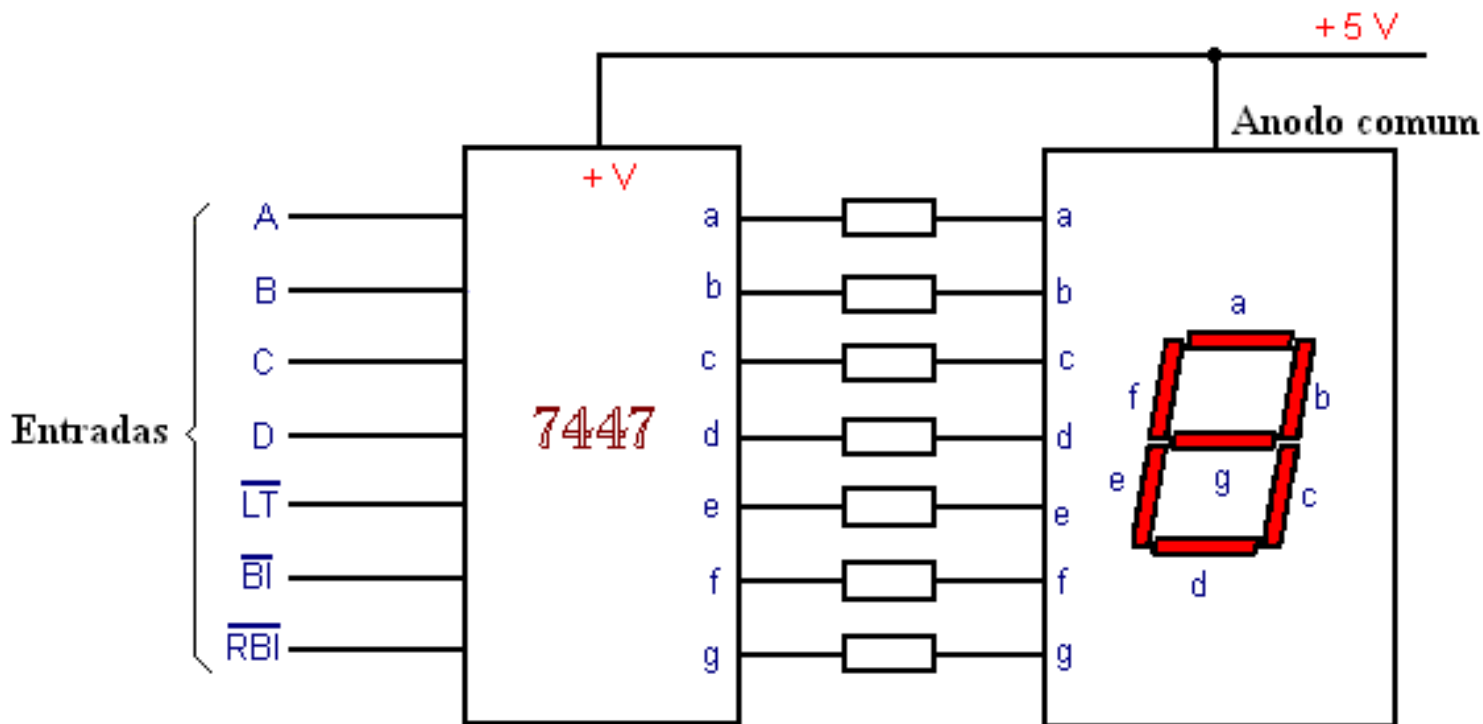
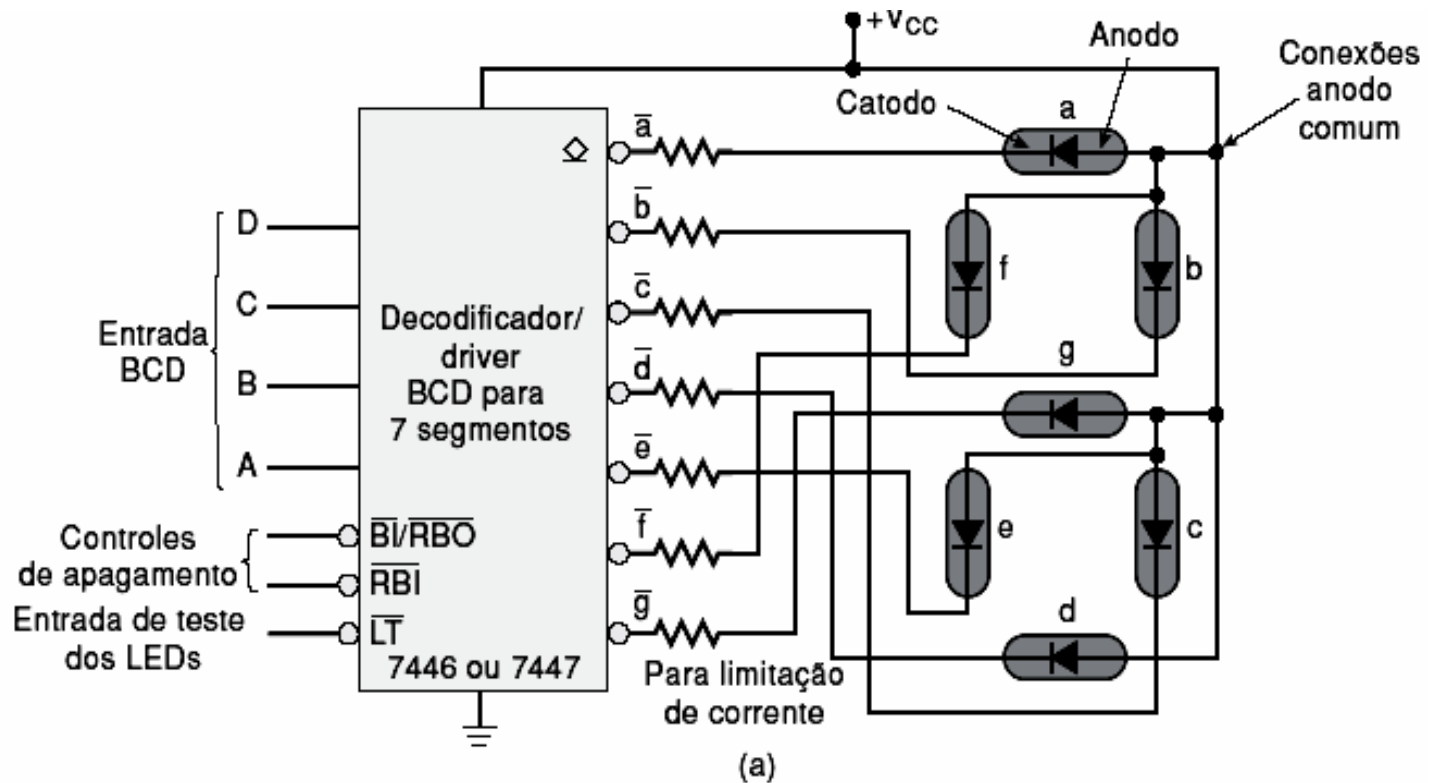
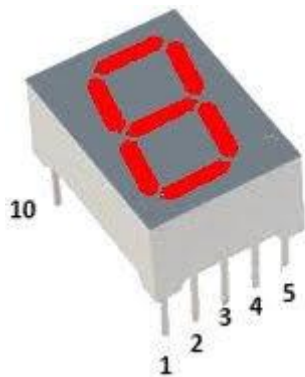
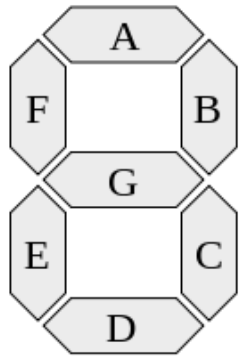


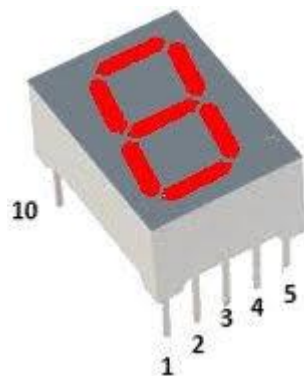
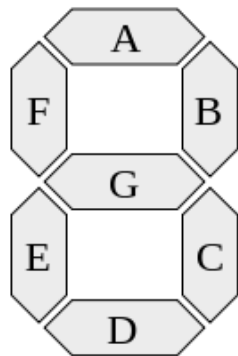
Figura 1 - Display de 7 segmentos comandado por um decodificador 7447.

# Decodificador para display de 7 segmentos



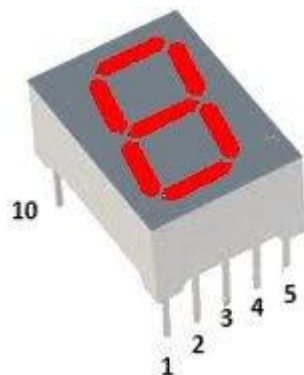
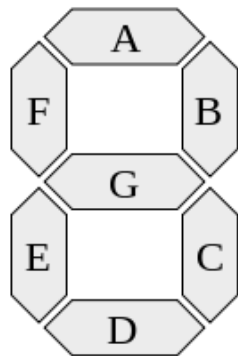
0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

# Decodificador para display de 7 segmentos de anodo comum



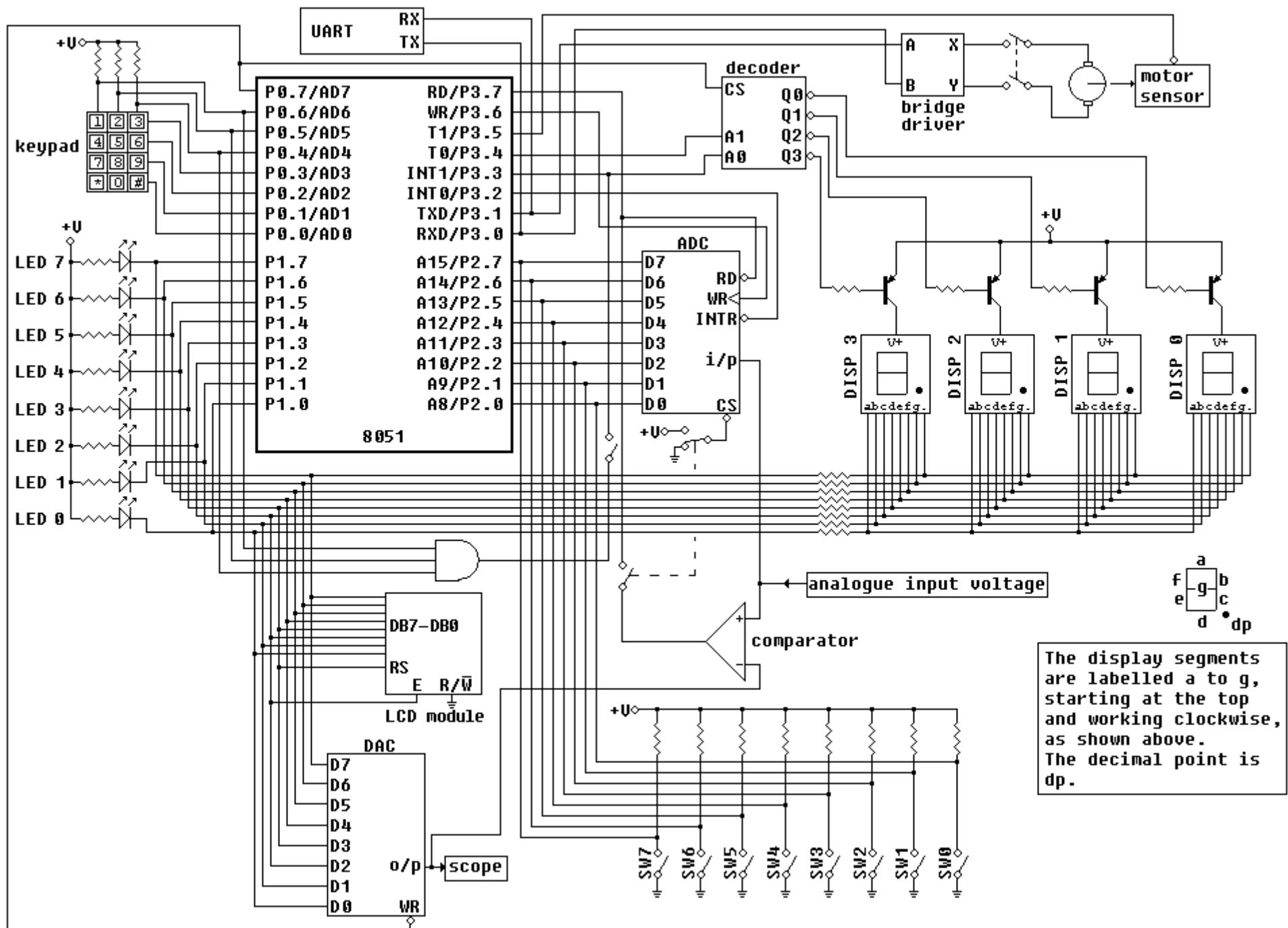
N°	Entradas				Saídas						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

# Decodificador para display de 7 segmentos de catodo comum

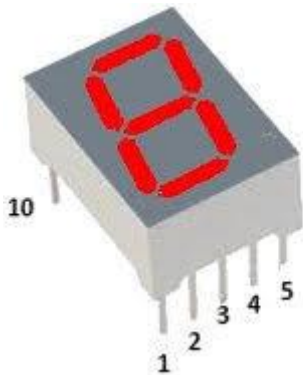
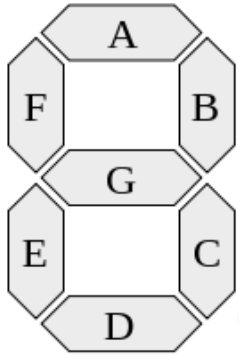


N°	Entradas				Saídas						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1





# Decodificador para display de 7 segmentos no programa edSim51



Nº	Saídas								
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1
	•	g	f	e	d	c	b	a	HEX
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	1	83
7	1	1	1	1	1	0	0	0	F8
8	1	0	0	0	0	0	0	0	80
9	1	0	0	1	1	0	0	0	98

# Exemplo 1

**start:**

<b>MOV P3, #1111111b</b>	<b>; HABILITA display 3</b>
<b>MOV P1, #0F9h</b>	<b>; Apresenta 1 no display</b>
<b>CALL delay</b>	
<b>MOV P3, #1110111b</b>	<b>; HABILITA display 2</b>
<b>MOV P1, #0A4h</b>	<b>; Apresenta 2 no display</b>
<b>CALL delay</b>	
<b>MOV P3, #1110111b</b>	<b>; HABILITA display 1</b>
<b>MOV P1, #0B0h</b>	<b>; Apresenta 3 no display</b>
<b>CALL delay</b>	
<b>MOV P3, #11100111b</b>	<b>; HABILITA display 0</b>
<b>MOV P1, #099h</b>	<b>; Apresenta 4 no display</b>
<b>CALL delay</b>	
<b>JMP start</b>	<b>; Salta para o start</b>

**delay:**

```
MOV R0, #250  
DJNZ R0, $  
RET
```

# Exemplo 2

**;Subrotina para atualizar o display de 7 segmentos**

**;RECEBE: A = número a ser mostrado no display**

**;**

**DEC7SEG:**

**MOV DPTR,#TABELA**

**;DPTR = início da tabela de códigos**

**MOVC A,@A+DPTR**

**;lê a tabela da memória de programa**

**MOV P1,A**

**;escreve código na porta P1**

**RET**

**;Tabela (memória de programa) com os códigos para o display**

**; 0 1 2 3 4 5 6 7 8 9**

**TABELA:**

**DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 83H, 0F8H, 80H, 98H**

# Exemplo 3

**start:**

**MOV R2, #10**

**ROT:**

**MOV A, R2**

**DEC A**

**CALL DEC7SEG** ; Chama a subrotina DEC7SEG

**CALL delay** ; Chama a subrotina delay

**DJNZ R2, ROT**

**JMP start** ; Salta para o start

**delay:**

**MOV R0, #250**

**DJNZ R0, \$**

**RET**

**DEC7SEG:**

**MOV DPTR, #TABELA** ;DPTR = início da tabela de códigos

**MOVC A, @A+DPTR** ;lê a tabela da memória de programa

**MOV P1, A** ;escreve código na porta P1

**RET**

**;Tabela (memória de programa) com os códigos para o display**

**; 0 1 2 3 4 5 6 7 8 9**

**TABELA:**

**DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 83H, 0F8H, 80H, 98H**

# Bibliografia

---

ZELENOVSKY, R.; MENDONÇA, A. Microcontroladores Programação e Projeto com a Família 8051. MZ Editora, RJ, 2005.

Gimenez, Salvador P. Microcontroladores 8051 - Teoria e Prática, Editora Érica, 2010.