

8051

AULA 8

Prof. Adilson Gonzaga

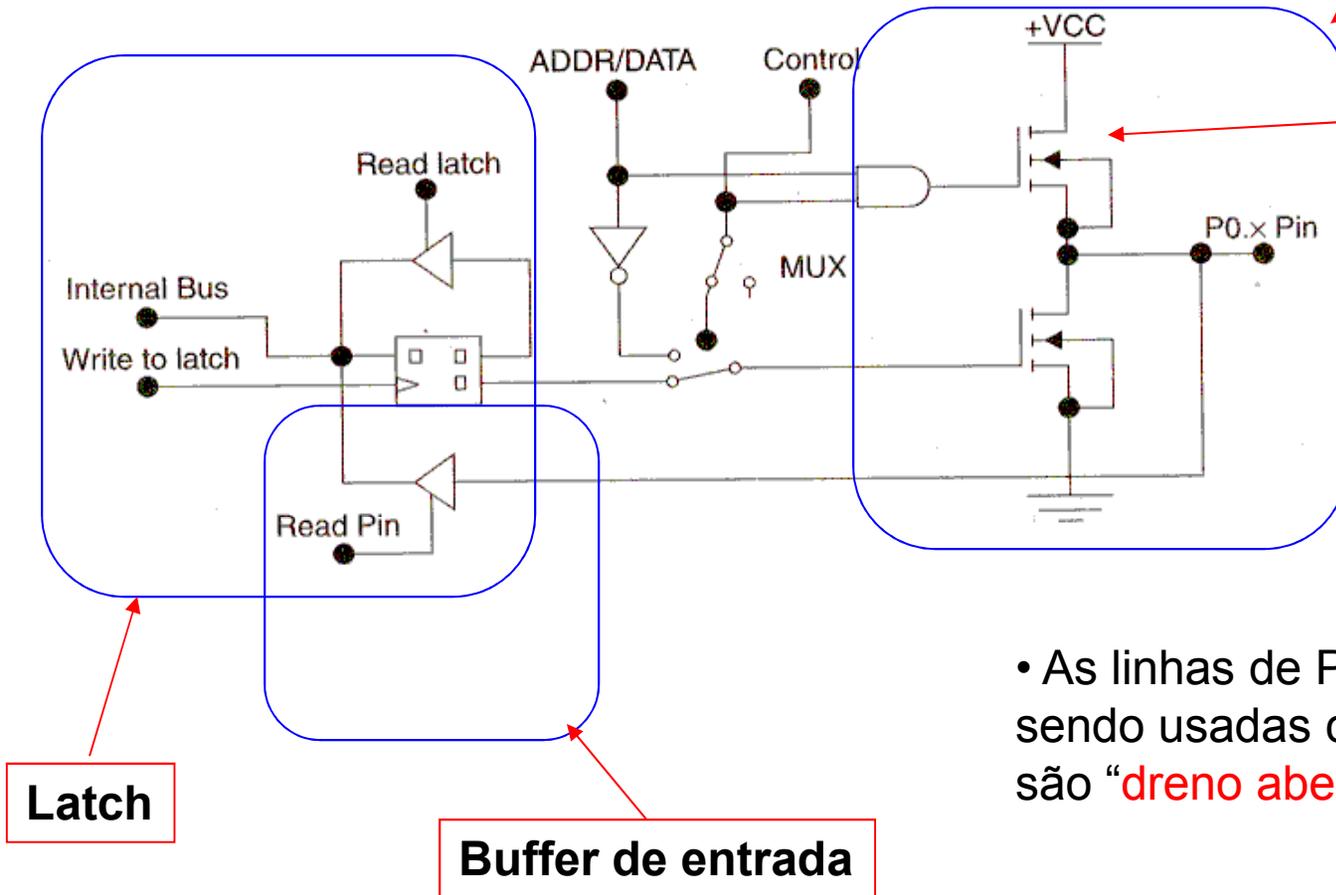
Técnicas de Interface

Técnicas de Interface com Microcontroladores (8051)

Estrutura das Portas do 8051:

- Todas as quatro Portas (P0,P1,P2,P3) da família de Microcontroladores MCS-51 são bi-direcionais.
- Cada uma delas consiste de um **Latch**, um **Driver** de saída e um **Buffer** de entrada.
- Os Drivers de saída e os Buffers de entrada da Porta P3 são multifuncionais - podem também assumir funções alternativas.

Arquitetura interna da Porta P0



Driver

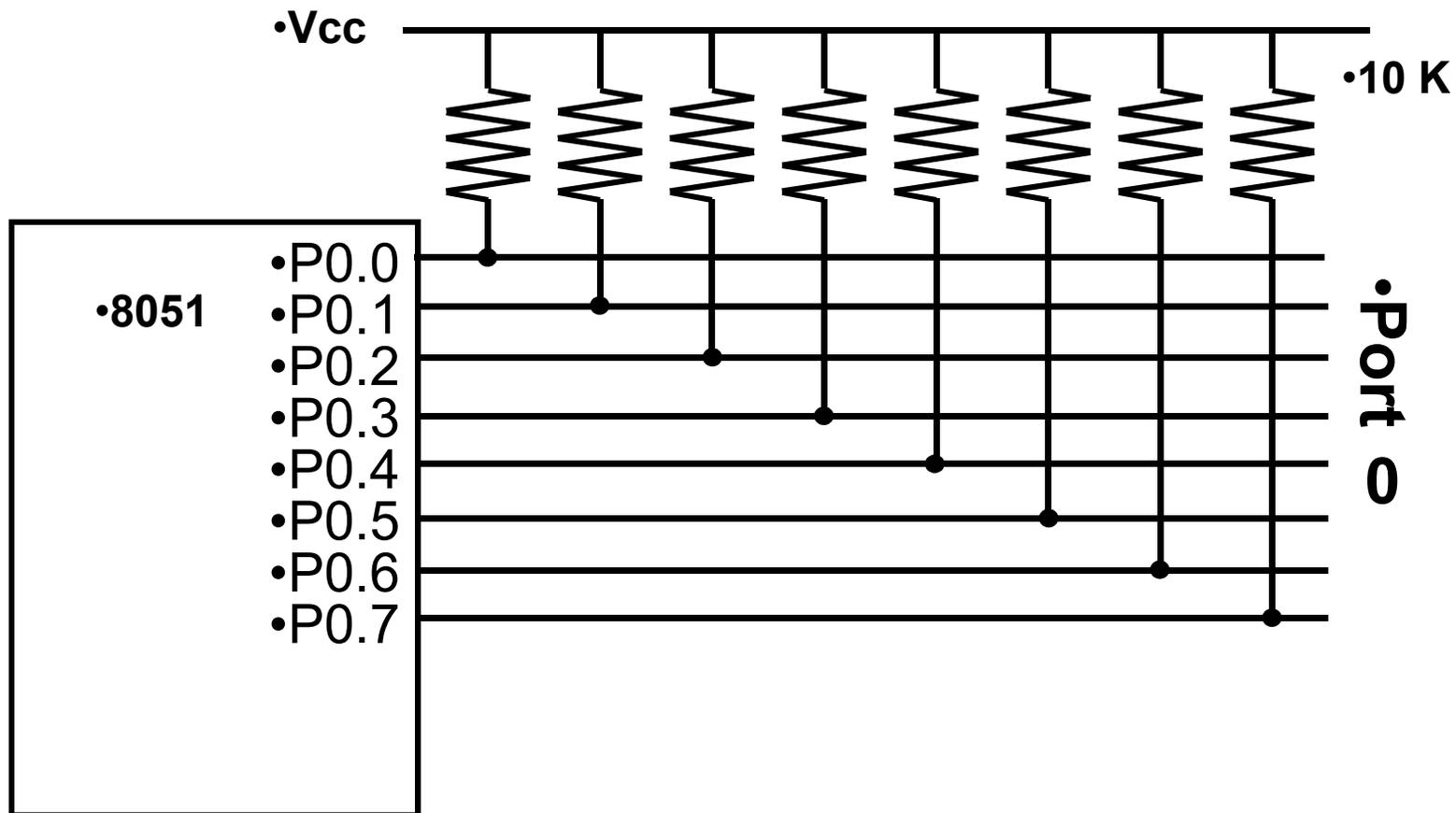
- O FET do Driver só está ativo quando a porta está fornecendo 1 lógico durante o acesso à memória externa

- As linhas de P0 que estão sendo usadas como saídas são “dreno aberto”.

- Escrever nível lógico 1 em qualquer pino da P0 desativa os FETs de saída e assim o pino flutua.

- Porta Bi-direcional verdadeira

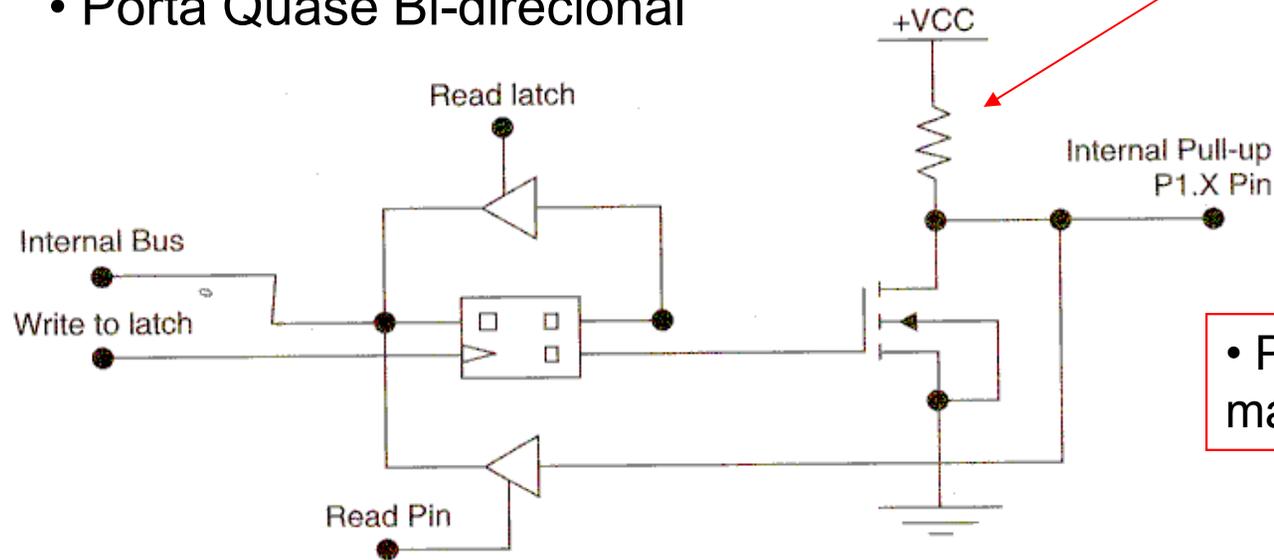
Porta 0 com Resistores de Pull-Up



Para utilizar a Porta 0 como Porta comum, deve-se adicionar Resistores de pull-up

Arquitetura interna da Porta P1

- Porta Quase Bi-direcional



- O Resistor de Pull-up fixo, fornece corrente (**IIL = Corrente de Input Low**) quando externamente é aplicado um nível lógico 0.

• Para o 89S52 o valor máximo de **IIL = - 50 μ A**

- Quando o Microcontrolador escreve nível lógico zero na saída, os Máximos Valores de **IOL (Corrente de Output Low)** para o 89S52 são:

Máxima IOL por pino: **10 mA**

Máxima IOL para 8-bits: P0 = **26 mA**

P1, P2 e P3: **15 mA**

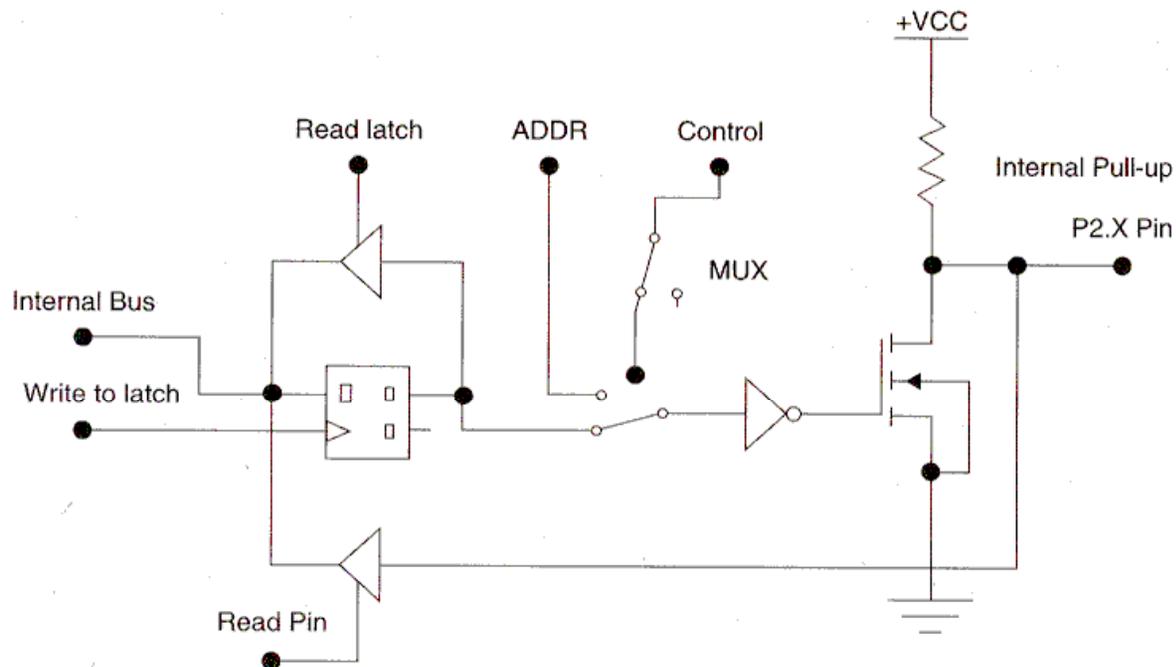
Máxima IOL - todos os pinos (P0, P1,P2,P3): **71 mA**

- A máxima corrente de saída em nível alto (**IOH = Corrente de Output High**) para o 89S52 é = **- 50 μ A**

Arquitetura interna da Porta P2

- Porta Quase Bi-direcional

- Os valores de IOL, IOH, IIL e IIH são os mesmos da Porta P1



- Após um Reset, todas as saídas dos Latches vão para nível lógico 1, programando todas as Portas (P0, P1, P2, P3) como entradas.

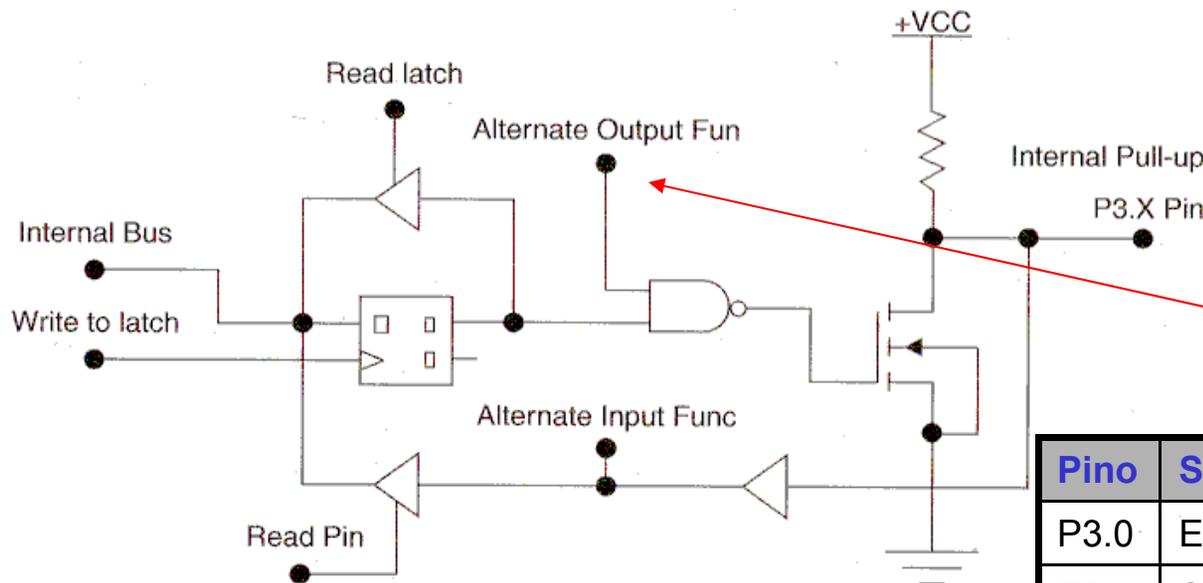
- A Porta P2, como a P0, não pode ser usada como entrada/saída se estiver sendo usada como Barramento de Endereços.

Arquitetura interna da Porta P3

- Porta Quase Bi-direcional

• Os valores de IOL, IOH, IIL e IIH são os mesmos da Porta P1

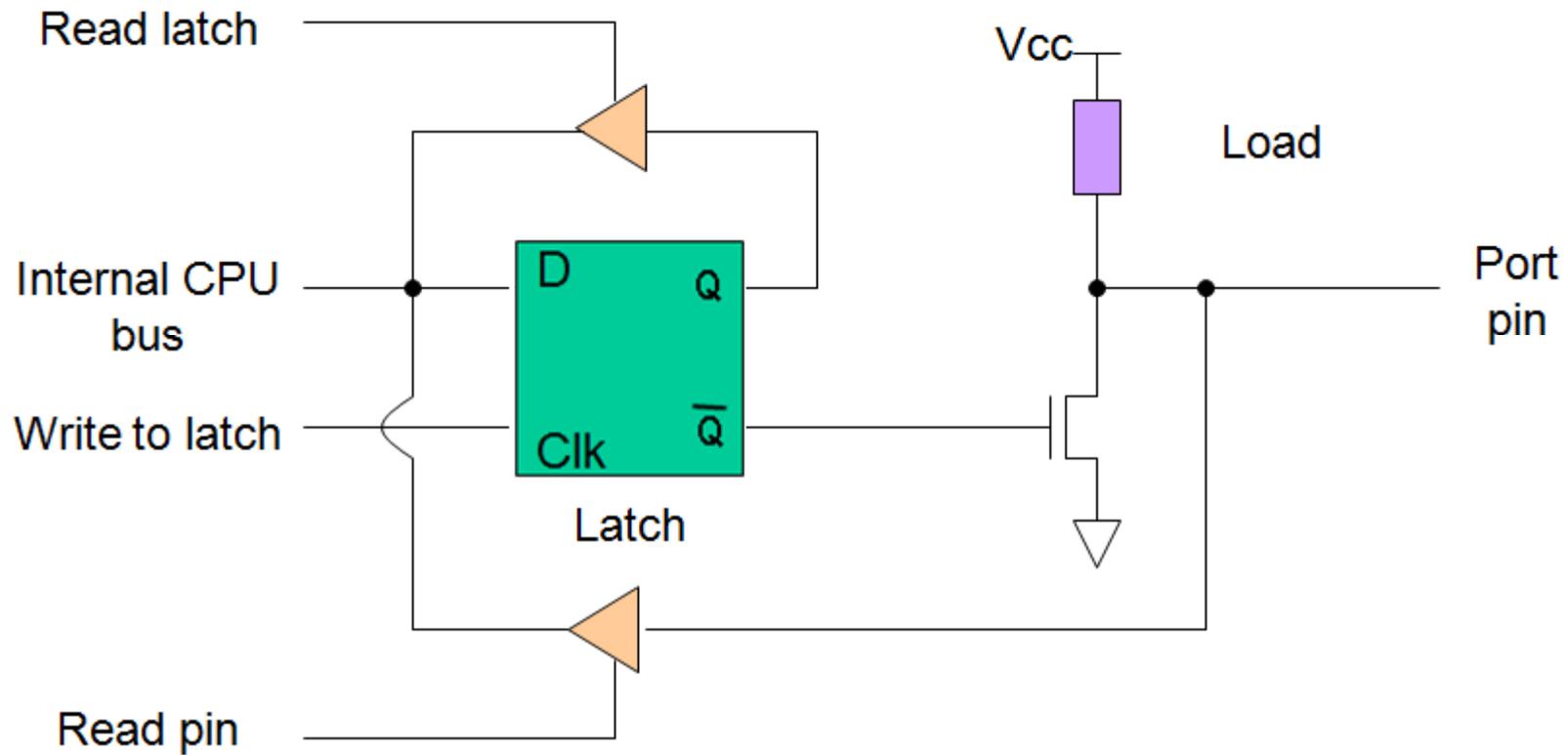
• Se o Latch da Porta P3 está em nível lógico 1, a saída é controlada pelo sinal de **Função de Saída Alternativa**



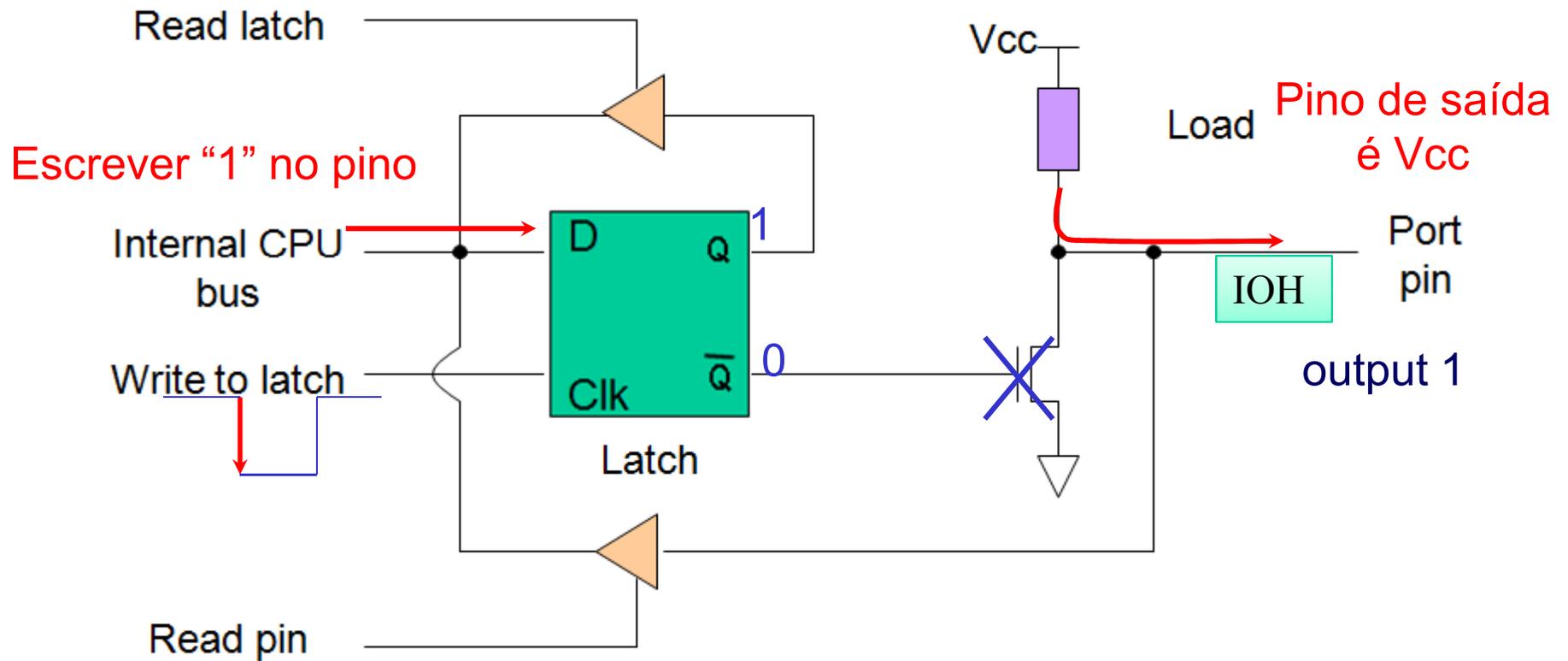
- Os Pinos da Porta P3 estão sempre disponíveis como Entrada das Funções Alternativas

Pino	Sentido	Função Alternativa
P3.0	Entrada	RxD (Entrada Serial)
P3.1	Saída	TxD (Saída Serial)
P3.2	Entrada	INT0 (Interrupção 0)
P3.3	Entrada	INT1 (Interrupção 1)
P3.4	Entrada	T0 (Contador 0)
P3.5	Entrada	T1 (Contador 1)
P3.6	Saída	WR (sinal de escrita)
P3.7	Saída	RD (Sinal de Escrita)

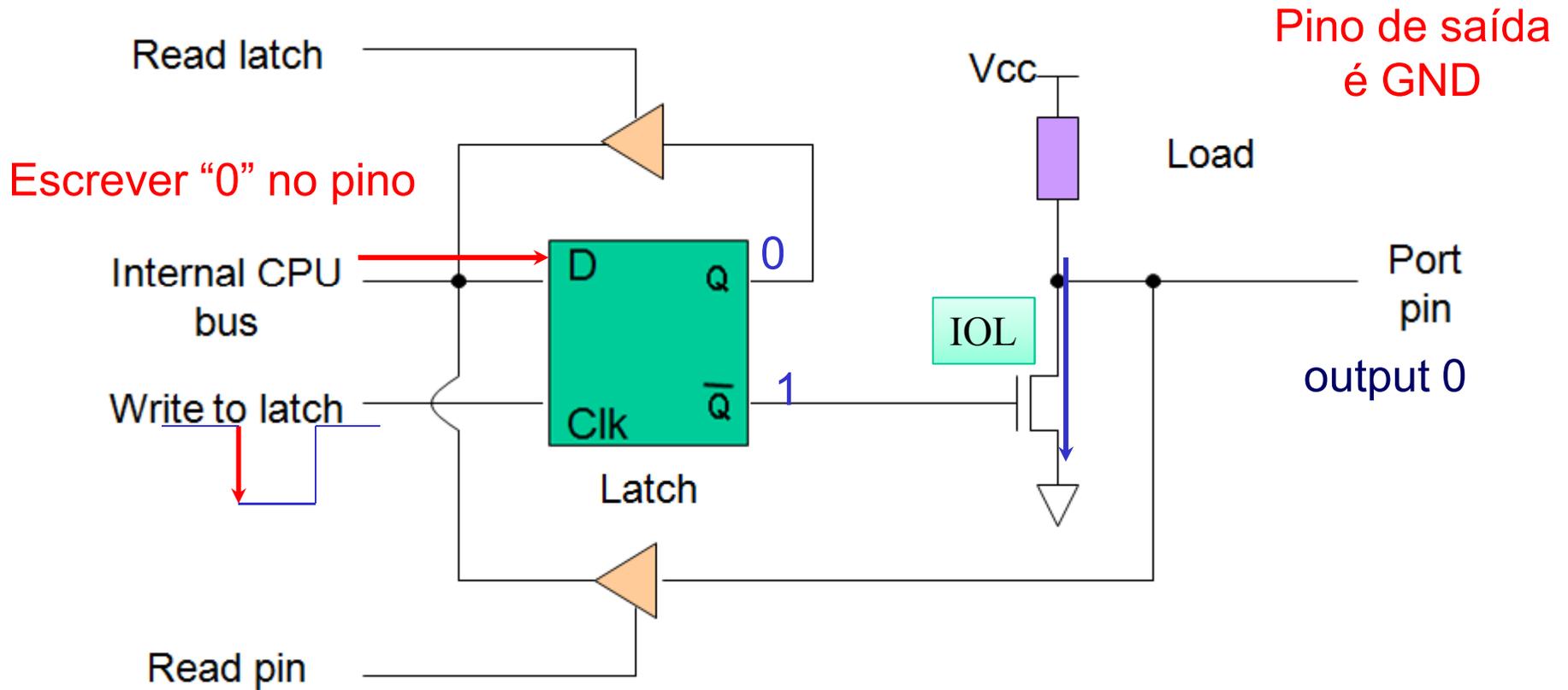
Exemplo de Operação das Portas do 8051: Circuito Genérico de um Bit de Porta



Escrevendo "1"

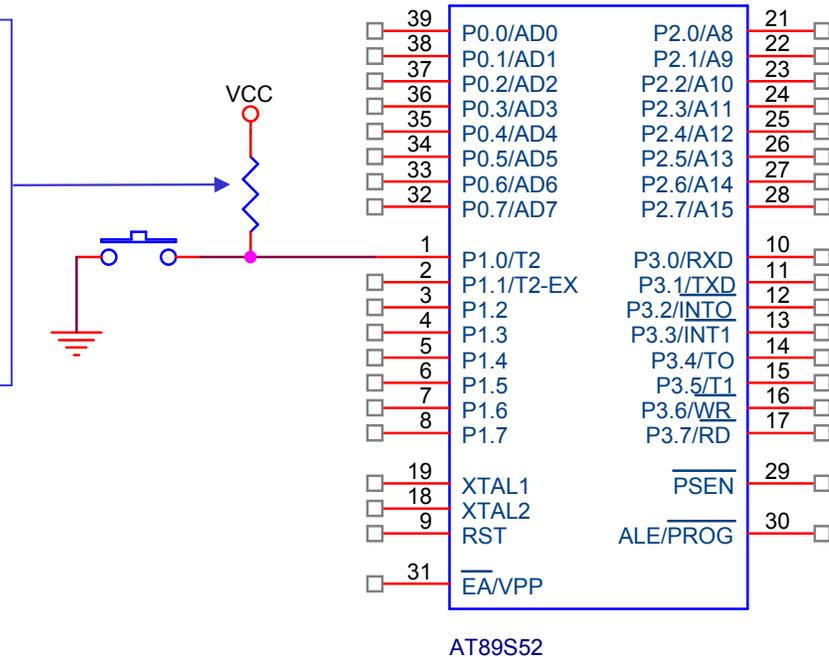


Escrevendo “0”



Conectando Chaves mecânicas às entradas do Microcontrolador

É possível utilizar, mas não obrigatoriamente, um Resistor de **10K ohms** que consumirá uma corrente da fonte de aproximadamente 500 μA quando a chave for acionada.



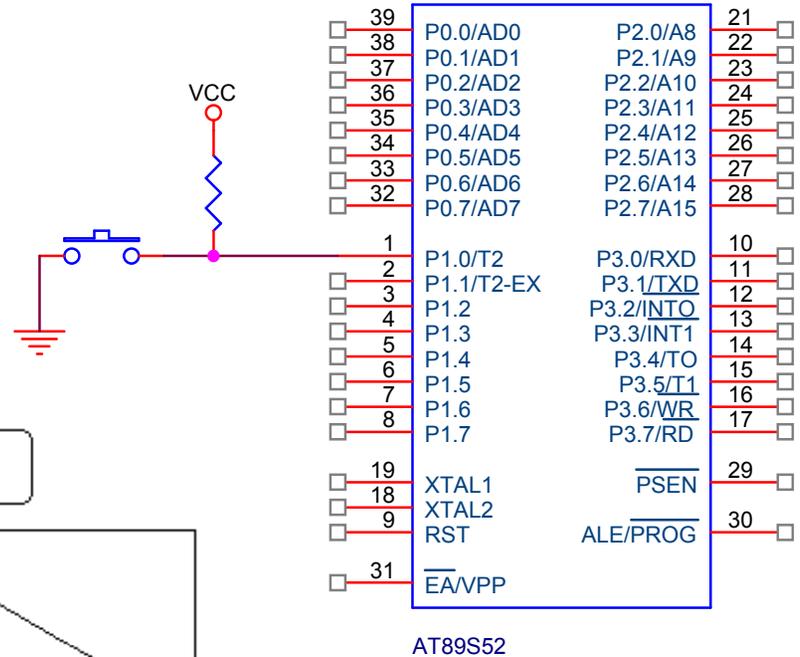
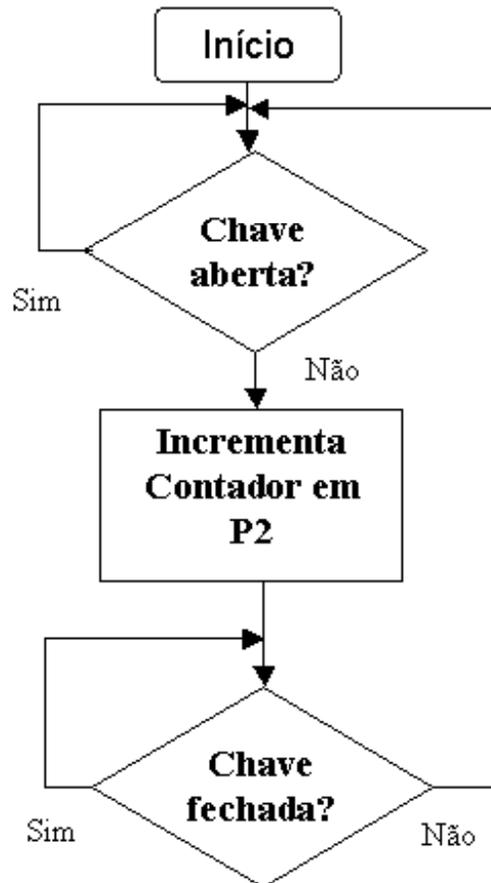
- É importante ressaltar que quando a chave for acionada, o resistor de Pull-up da porta fornece também no máximo $I_{IL} = 50 \mu\text{A}$.

Exemplo de Aplicação

- Considerando o esquema, implementar um contador binário de 8 Bits cujo valor deve ser incrementado de uma unidade a cada toque na chave e enviado à Porta P2

• Primeira Solução

- Testando-se a chave duas vezes verifica-se se o usuário já completou a operação de um toque, ou seja, um pulso completo.

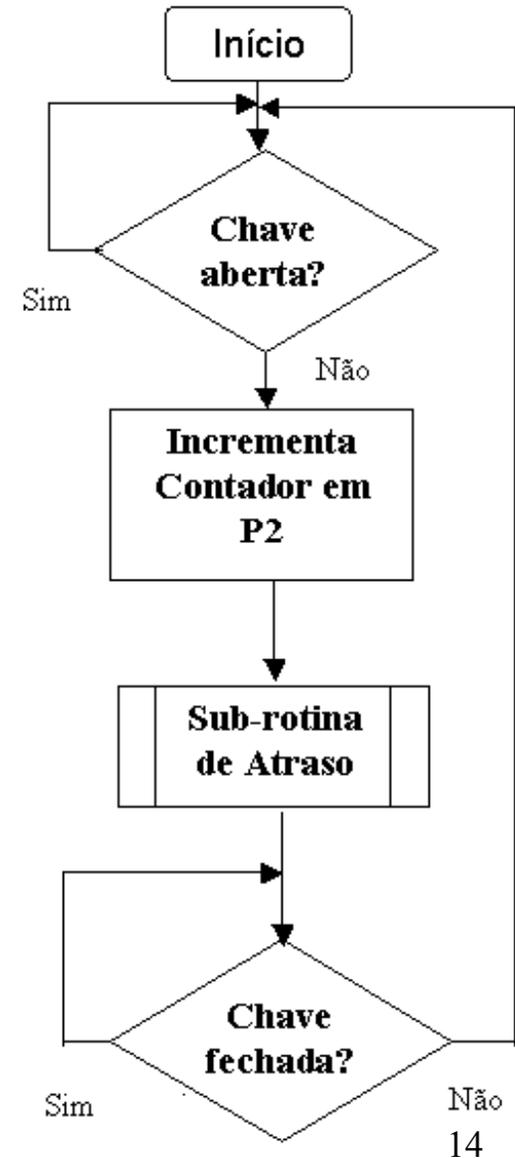
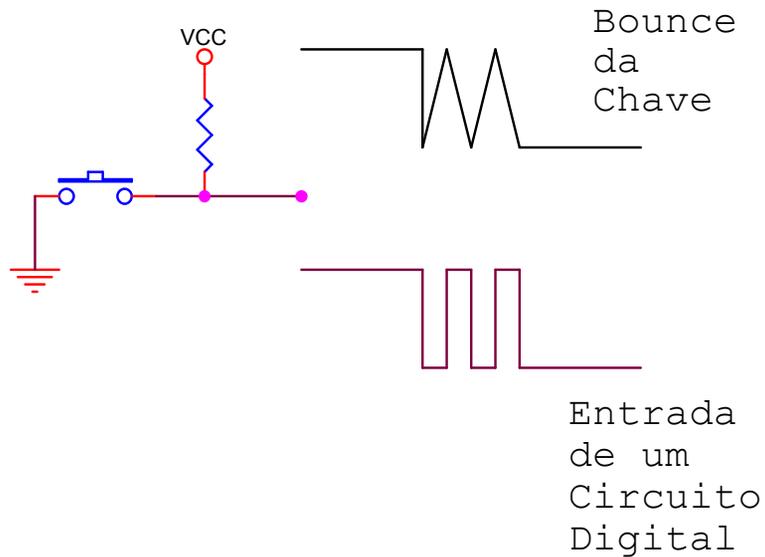


- Esta solução funcionaria na Prática?

NÃO

- As chaves mecânicas possuem inércia ao fechar que geram ruídos que poderão ser detectados pelo software ao se proceder a leitura da mesma.

- Este ruído, chamado de Bounce da chave deve ser eliminado, ou por hardware (capacitor, FF tipo D) ou por software inserindo um atraso antes da leitura da chave fechada.



```

;*****
; Programa de debounce de chave mecânica
; Uma sub-rotina de atraso deve ser inserida para garantir o tempo de
; estabilização mecânica da chave. Para a maioria das chaves um atraso de
; 500 us é suficiente.
;*****

```

```

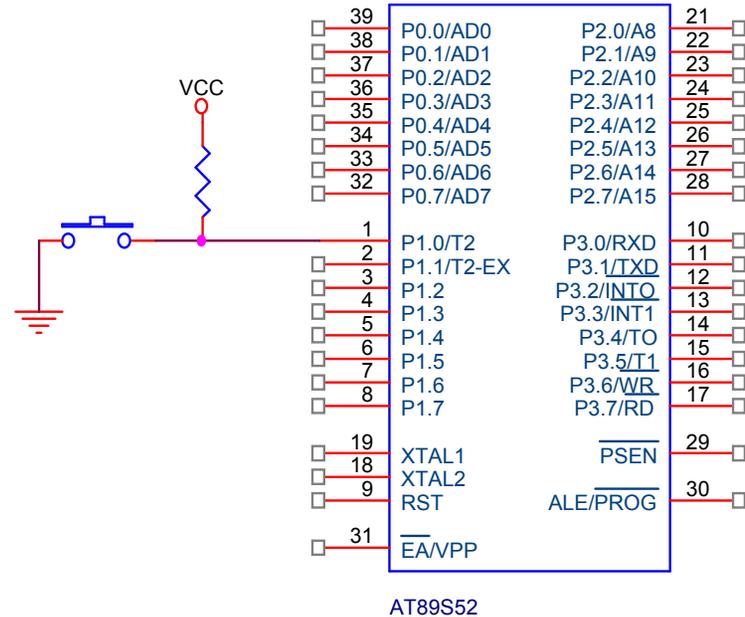
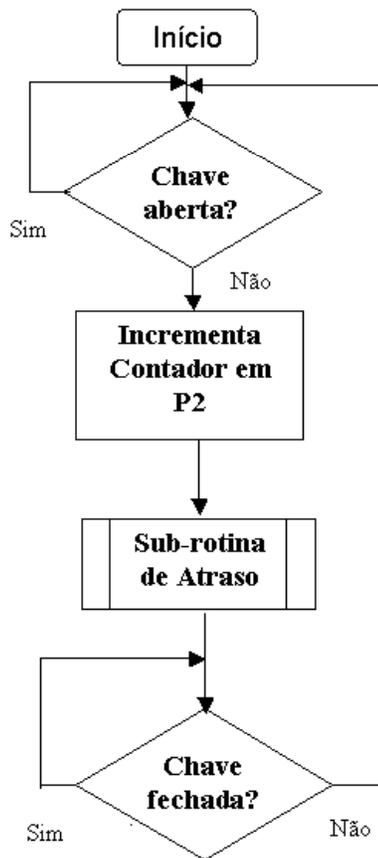
ORG 0
CLR A           ; Inicializa o contador em zero
LOOP: JB P1.0,$ ; Testa se a chave está aberta
      INC A     ; Incrementa o contador
      MOV P2,A  ; Mostra o valor do contador na Porta P2
      CALL ATRASO ; Realiza o debounce da chave
      JNB P1.0,$ ; Verifica se a chave continua fechada
      SJMP LOOP ; Retorna para nova contagem de pulso

```

```

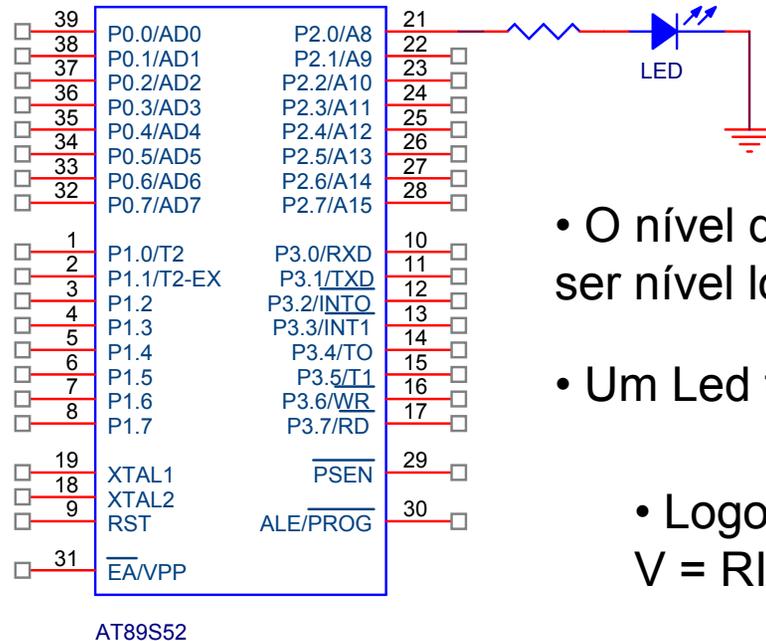
;*****
ATRASO: MOV R0,#OFFH ; Valor do atraso para debounce da chave de
        DJNZ R0,$   ; aproximadamente 500 us
        RET
;*****

```



Conectando Leds às saídas do Microcontrolador

- Qual o valor do Resistor e da corrente necessária para acender o Led no esquema abaixo?

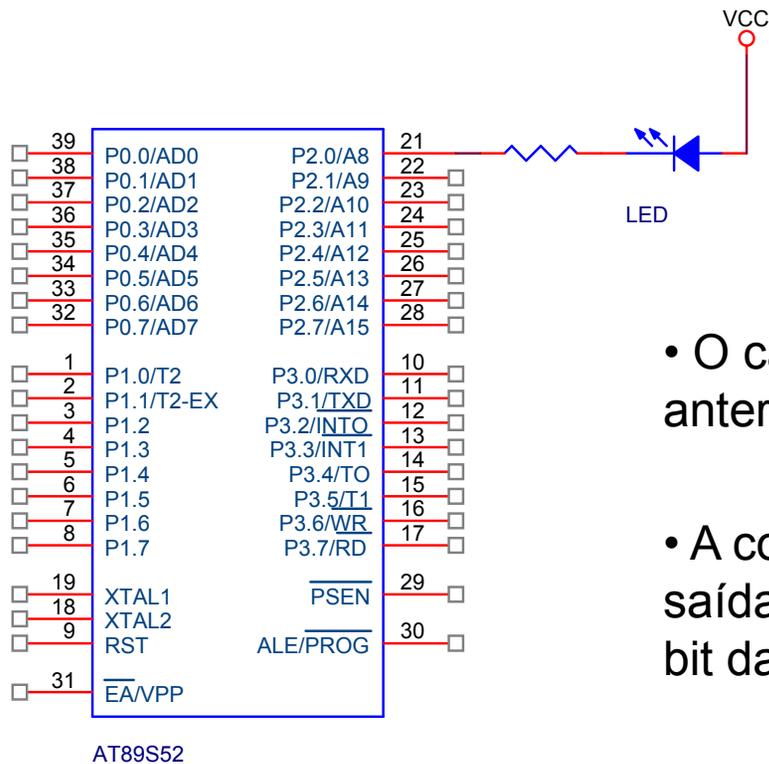


- O nível da porta para acionar o Led deve ser nível lógico 1, ou seja = 5v
- Um Led tem uma queda de tensão de 1,4 v
- Logo,

$$V = RI + V_{led} \rightarrow R = (5 - 1,4) / I$$

- Como um Led necessita de aproximadamente 10 mA para uma boa luminosidade $\rightarrow R = 3,6 / 10\text{mA} = 360$ ohms
- Como a corrente de saída em nível alto IOH é no máximo 50 uA, esta ligação é inviável

- E se alterarmos o esquema de conexão, qual o valor do Resistor e da corrente necessária para acender o Led ?



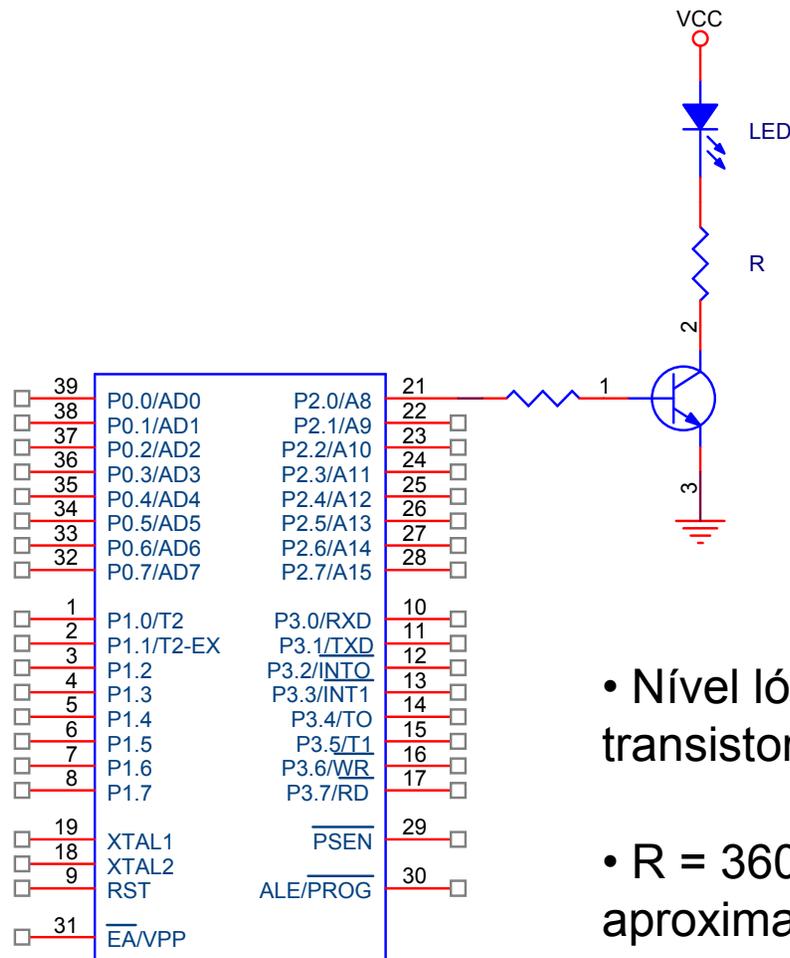
- O nível da porta para acionar o Led deve ser nível lógico 0, ou seja = 0v

- O cálculo do Resistor é o mesmo realizado anteriormente, ou seja, $R = 360$ ohms

- A corrente a ser avaliada agora é a de nível de saída baixo, ou seja, IOL é no máximo 10 mA por bit da Porta P2.

- O valor máximo de IOL para os 8 Bits da Porta P2 é 15 mA, ou seja, a solução é viável para um Bit mas não deve ser adotada para a porta toda, pois o nível de corrente ultrapassa o máximo recomendado.

Bufferizando as saídas do Microcontrolador



AT89S52

- Utilizando-se um Buffer/driver com transistores, um para cada bit da porta, fornece-se a corrente necessária à operação do Led sem afetar os valores máximos permitidos para o Microcontrolador.

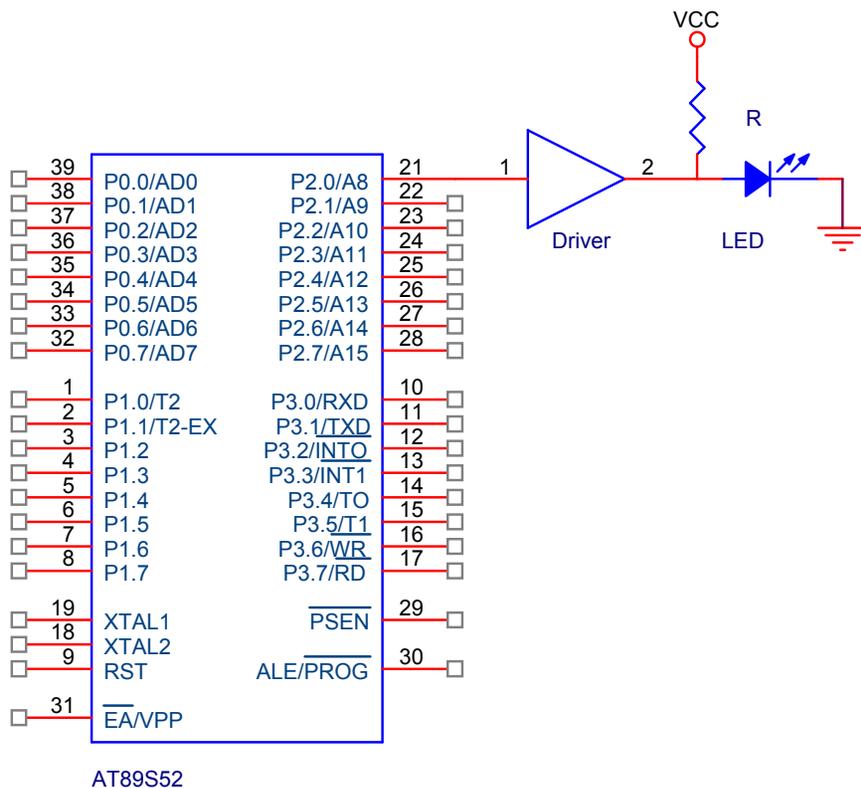
- Nível lógico 1 em P2.0 conduz o transistor acendendo o Led

- $R = 360 \text{ Ohms}$ fornece corrente de aproximadamente 10 mA ao Led.

- O resistor de base deve ser calculado conforme especificações do transistor com IOH máxima de 60 uA

Bufferizando as saídas do Microcontrolador

- A solução mais indicada é utilizar um Circuito “**Led Driver**” que fornecerá a corrente necessária aos Leds em cada um dos Bits da Porta não permitindo ultrapassar os valores máximos recomendados.



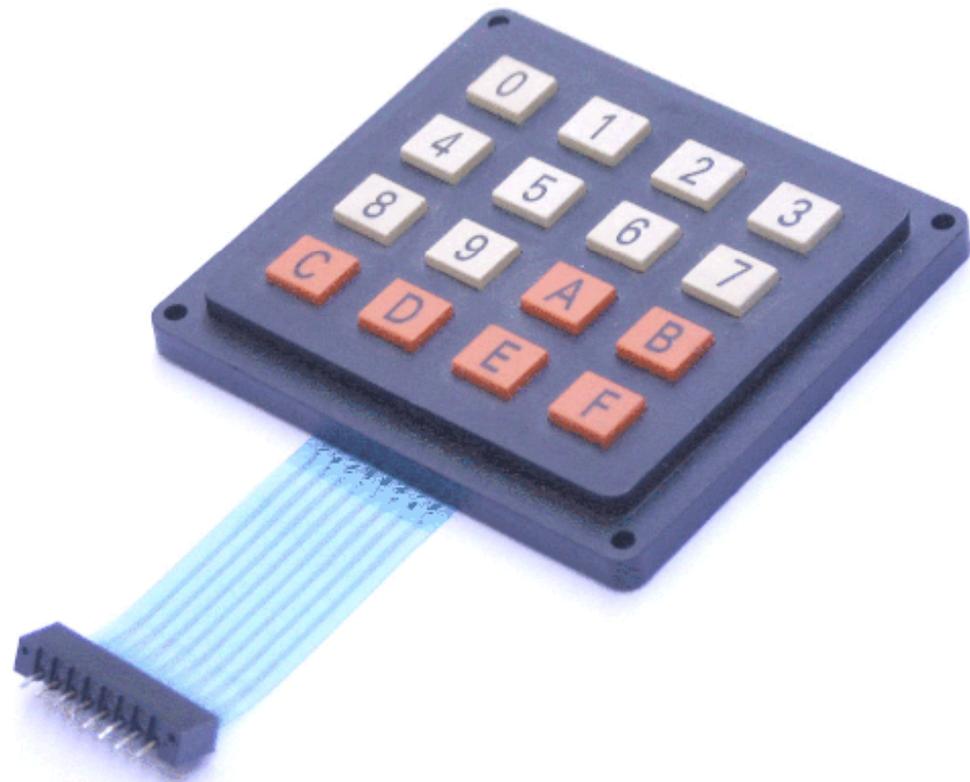
Alguns CI's de Led Driver

- 7407 (4 por chip)
- 74244 (8 por chip)
- ULN2803 (8 por chip)
- outros.....

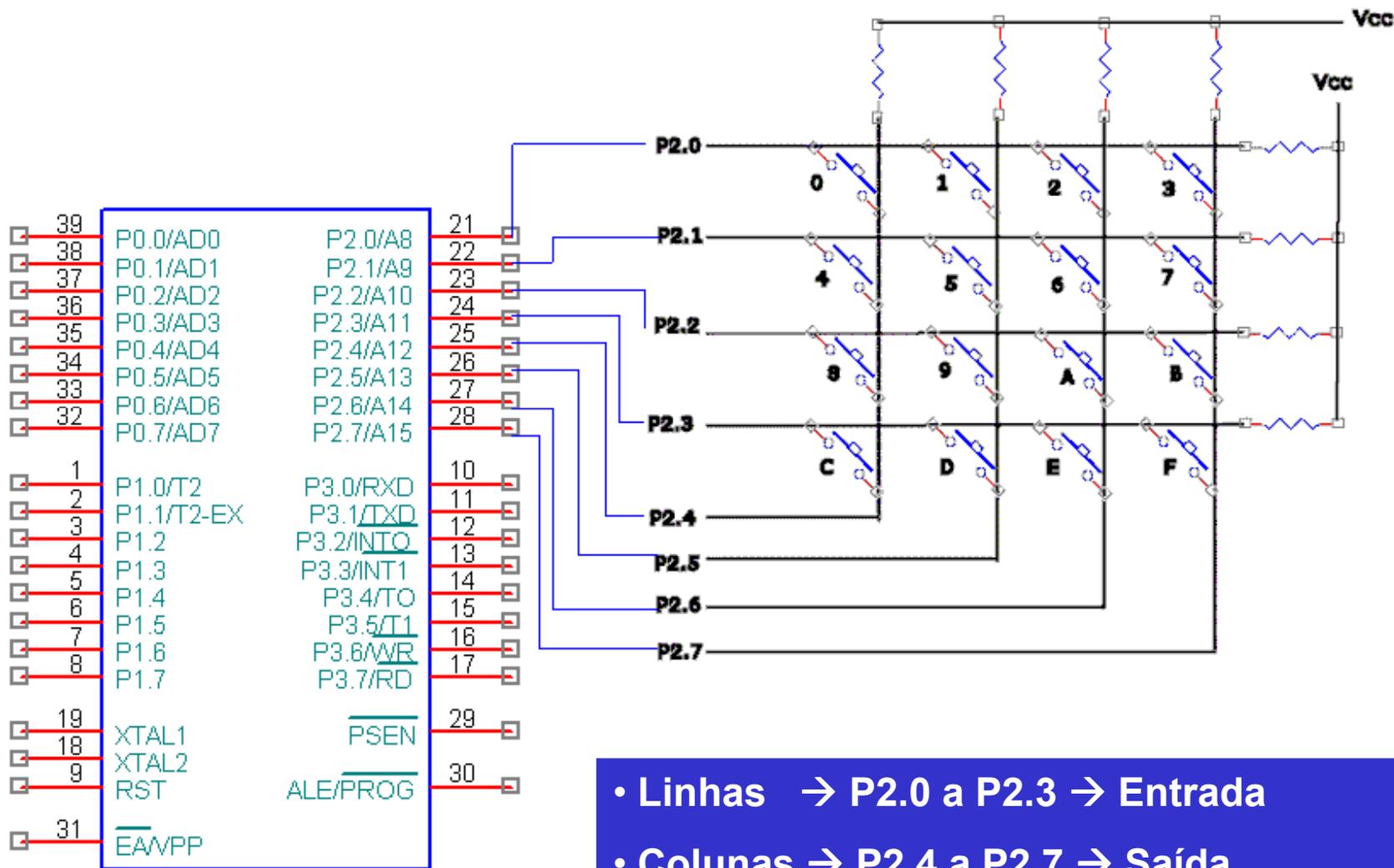
Interface do Microcontrolador com Teclado Matricial

- Um Teclado Matricial de 4 linhas por 4 colunas permite a geração de 16 códigos independentes utilizando-se apenas 8 linhas de uma Porta do Microcontrolador.

- Dessa maneira é possível interfacear um Teclado Hexadecimal Matricial ao Microcontrolador para entrada de dados binários



Interface direta do Microcontrolador com Teclado Matricial

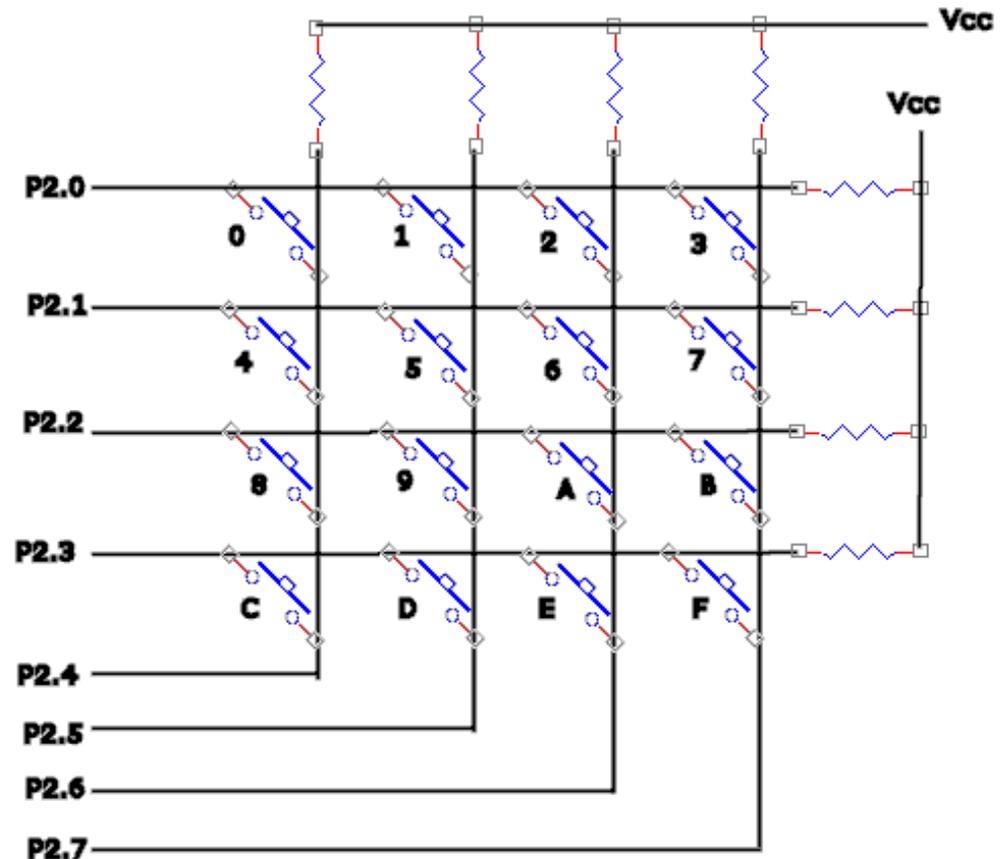


AT89S52

- Linhas → P2.0 a P2.3 → Entrada
- Colunas → P2.4 a P2.7 → Saída

Valores aplicados às colunas(P2.4 a P2.7) e lidos nas linhas (P2.0 a P2.3), se for acionada uma tecla presente na coluna que contém zero.

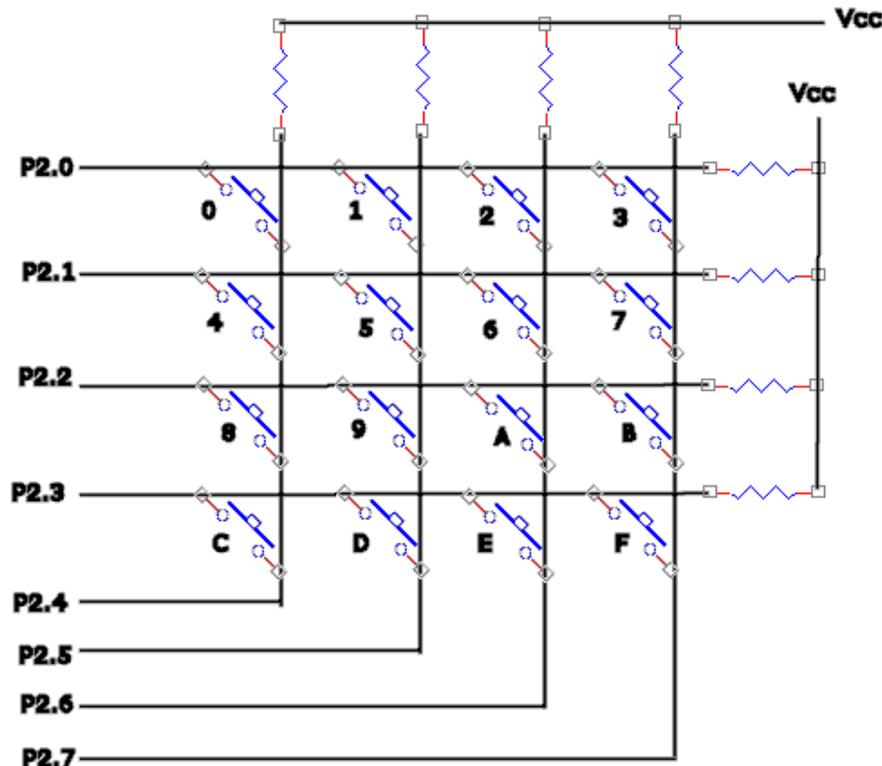
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	TECLA
1	1	1	0	1	1	1	0	0
1	1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	0	2
0	1	1	1	1	1	1	0	3
1	1	1	0	1	1	0	1	4
1	1	0	1	1	1	0	1	5
1	0	1	1	1	1	0	1	6
0	1	1	1	1	1	0	1	7
1	1	1	0	1	0	1	1	8
1	1	0	1	1	0	1	1	9
1	0	1	1	1	0	1	1	A
0	1	1	1	1	0	1	1	B
1	1	1	0	0	1	1	1	C
1	1	0	1	0	1	1	1	D
1	0	1	1	0	1	1	1	E
0	1	1	1	0	1	1	1	F



Caso nenhuma tecla seja acionada, o valor das linhas é = F

ORG 0

```
LOOP:      MOV P2,#0FFH      ; Garante nível lógico 1 em todos os bits de P2
LOOP1:    CLR P2.4          ; Zera a coluna das chaves 0,4,8,C
          JNB P2.0,ZERO
          JNB P2.1,QUATRO
          JNB P2.2,OITO
          JNB P2.3,CCC
          MOV P2,#0FFH      ; Garante nível lógico 1 em todos os bits de P2
LOOP2:    CLR P2.5          ; Zera a coluna das chaves 1,5,9,D
          JNB P2.0,UM
          JNB P2.1,CINCO
          JNB P2.2,NOVE
          JNB P2.3,DDD
          MOV P2,#0FFH      ; Garante nível lógico 1 em todos os bits de P2
LOOP3:    CLR P2.6          ; Zera a coluna das chaves 2,6,A,E
          JNB P2.0,DOIS
          JNB P2.1,SEIS
          JNB P2.2,AAA
          JNB P2.3,EEE
          MOV P2,#0FFH      ; Garante nível lógico 1 em todos os bits de P2
LOOP4:    CLR P2.7          ; Zera a coluna das chaves 3,7,B,F
          JNB P2.0,TRES
          JNB P2.1,SETE
          JNB P2.2,BBB
          JNB P2.3,FFF
          MOV A,#0FFH
          SJMP LOOP
```



```
ZERO:     MOV A,#00H
          SJMP LOOP1
UM:       MOV A,#01H
          SJMP LOOP2
DOIS:     MOV A,#02H
          SJMP LOOP3
TRES:     MOV A,#03H
          SJMP LOOP4
QUATRO:   MOV A,#04H
          SJMP LOOP1
CINCO:    MOV A,#05H
          SJMP LOOP2
SEIS:     MOV A,#06H
          SJMP LOOP3
SETE:     MOV A,#07H
          SJMP LOOP4
OITO:     MOV A,#08H
          SJMP LOOP1
NOVE:     MOV A,#09H
          SJMP LOOP2
AAA:      MOV A,#0AH
          SJMP LOOP3
BBB:      MOV A,#0BH
          SJMP LOOP4
CCC:      MOV A,#0CH
          SJMP LOOP1
DDD:      MOV A,#0DH
          SJMP LOOP2
EEE:      MOV A,#0EH
          SJMP LOOP3
FFF:      MOV A,#0FH
          SJMP LOOP4
```

Interface de um Teclado Matricial mapeado em Memória

Um Teclado Matricial é um dispositivo de entrada de dados.

Caso as portas do Microcontrolador estejam ocupadas com o mapeamento de memória (P0 e P2) e outros periféricos utilizam P1 e P3, pode-se mapear um Teclado Matricial na Memória RAM Externa do Microcontrolador.

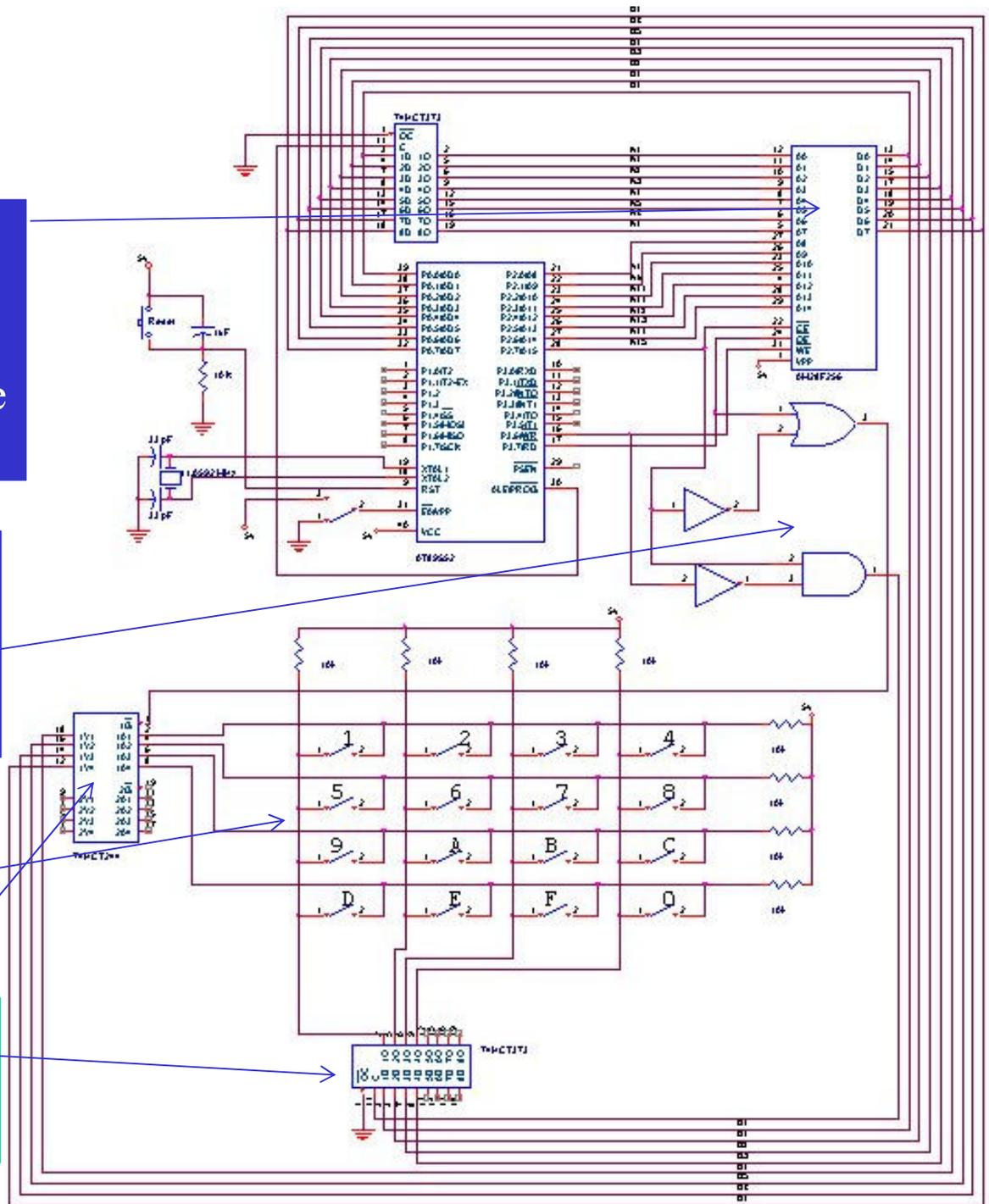
Exemplo:

Neste caso, existe uma RAM externa ocupando o espaço de endereço de 0000 a 7FFFh, ou seja, os primeiros 32 k Bytes de memória.

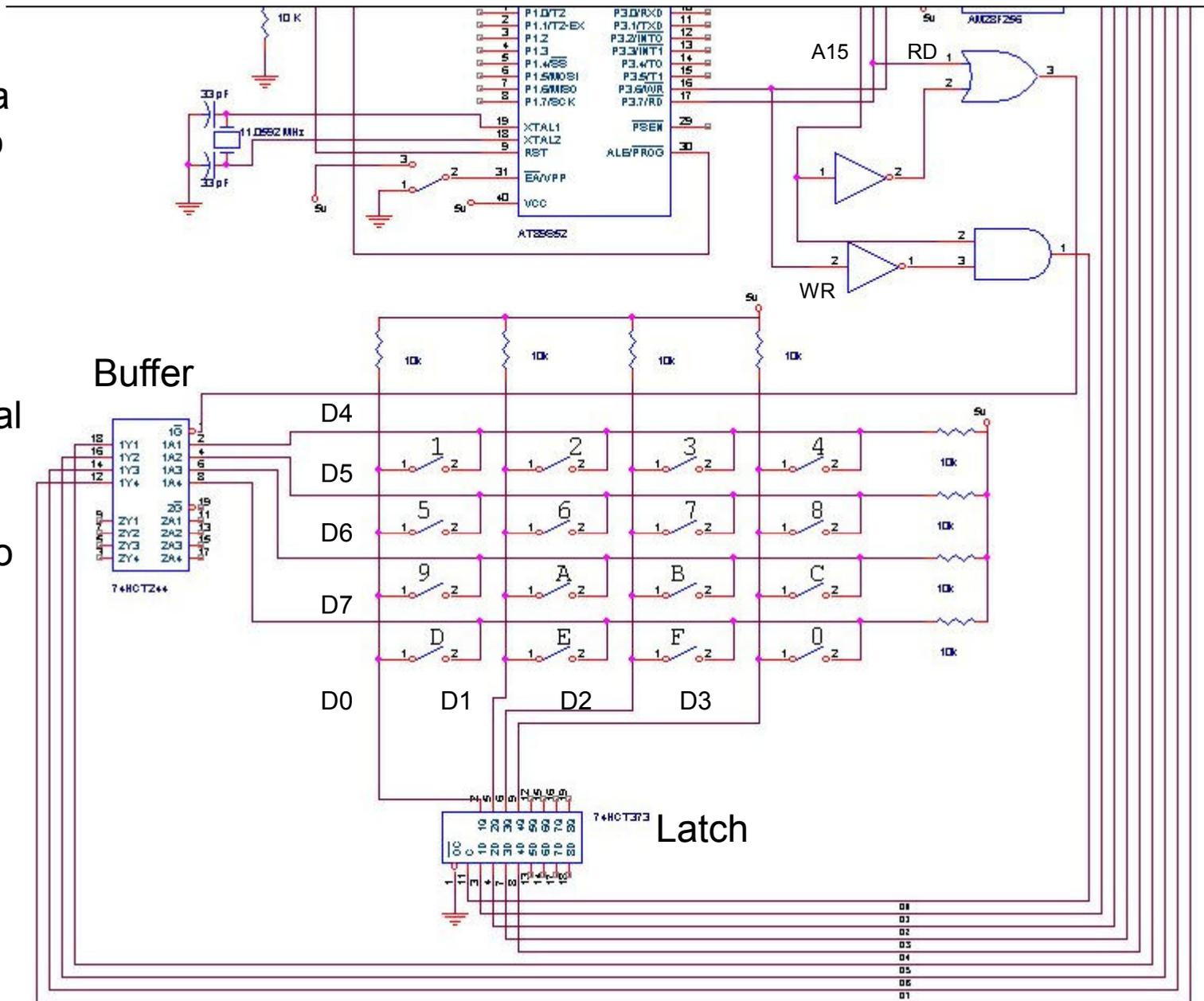
O circuito de mapeamento permite endereçar os 32 k Bytes superiores, ou seja, de 8000h a FFFFh

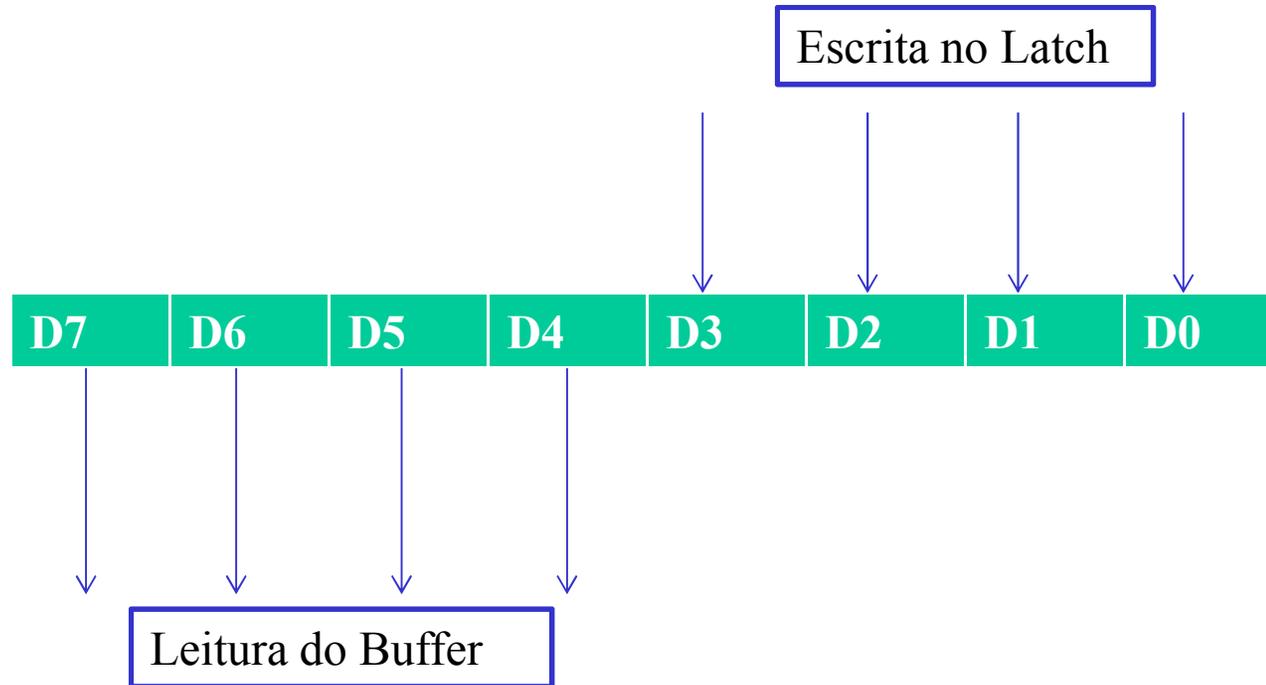
Um Teclado Hexadecimal foi mapeado nestes endereços.

Deve-se escrever o valor zero na coluna e ler o retorno na linha



Qualquer endereço acima de 8000h tem o A15 = 1, logo, serve para acessar na escrita (RW) a coluna do Teclado Matricial (Latch) e na leitura (RD) a linha do Teclado (Buffer).





```

Para Ler o Buffer
MOV  DPTR,#8000H
MOV  A,@DPTR

```

```

Para Escrever no Latch
MOV  DPTR,#8000H
MOVX @DPTR,A

```

D7	D6	D5	D4	D3	D2	D1	D0	HEXA	TECLA
0	1	1	1	0	1	1	1	77	0
1	1	1	0	1	1	1	0	EE	1
1	1	1	0	1	1	0	1	ED	2
1	1	1	0	1	0	1	1	EB	3
1	1	1	0	0	1	1	1	E7	4
1	1	0	1	1	1	1	0	DE	5
1	1	0	1	1	1	0	1	DD	6
1	1	0	1	1	0	1	1	DB	7
1	1	0	1	0	1	1	1	D7	8
1	0	1	1	1	1	1	0	BE	9
1	0	1	1	1	1	0	1	BD	A
1	0	1	1	1	0	1	1	BB	B
1	0	1	1	0	1	1	1	B7	C
0	1	1	1	1	1	1	0	7E	D
0	1	1	1	1	1	0	1	7D	E
0	1	1	1	1	0	1	1	7B	F

Sub-rotina para ler o Teclado Matricial mapeado em memória

```
LCALL          ORG      0
               VARRE_TECLA
               SJMP     $
; *****
; SUBROTINA DE VARREDURA DO TECLADO
; *****
VARRE_TECLA:   MOV      A,#0FEH          ; (11111110) D0 = 0
               MOV      R0,A
               MOV      DPTR,#8000H
               MOVX     @DPTR,A          ; ESCREVE NO LATCH
               ; ZERO SOMENTE NA COLUNA DE D0
               MOVX     A,@DPTR          ; LE O BUFFER (BBBBXXXX)
               ORL      A,#0FH           ; MÁSCARA (BBBB1111)
               CJNE     A,#0FFH,TECLA
               RET
               ; SE RETORNAR FF --> NENHUMA TECLA
               ; FOI APERTADA
; *****
```

TECLA:

```
JNB ACC.4,TECLA1 ; SE D4=0 --> TECLA_1
JNB ACC.5,TECLA5 ; SE D5=0 --> TECLA_5
JNB ACC.6,TECLA9 ; SE D6=0 --> TECLA_9
JNB ACC.7,TECLAD ; SE D7=0 --> TECLA_D
MOV A,R0
RLC A ; (11111101) D1 = 0
MOV R0,A
JNB ACC.4,TECLA2 ; SE D4=0 --> TECLA_2
JNB ACC.5,TECLA6 ; SE D5=0 --> TECLA_6
JNB ACC.6,TECLAA ; SE D6=0 --> TECLA_A
JNB ACC.7,TECLAE ; SE D7=0 --> TECLA_E
MOV A,R0
RLC A ; (11111011) D2 = 0
MOV R0,A
JNB ACC.4,TECLA3 ; SE D4=0 --> TECLA_3
JNB ACC.5,TECLA7 ; SE D5=0 --> TECLA_7
JNB ACC.6,TECLAB ; SE D6=0 --> TECLA_B
JNB ACC.7,TECLAF ; SE D7=0 --> TECLA_F
MOV A,R0
RLC A ; (11110011) D3 = 0
MOV R0,A
JNB ACC.4,TECLA0 ; SE D4=0 --> TECLA_0
JNB ACC.5,TECLA4 ; SE D5=0 --> TECLA_4
JNB ACC.6,TECLA8 ; SE D6=0 --> TECLA_8
JNB ACC.7,TECLAC ; SE D7=0 --> TECLA_C
L JMP VARRE_TECLA
```

```

, ****
TECLA1:      MOV      A,#01H
              RET
TECLA2:      MOV      A,#02H
              RET
TECLA3:      MOV      A,#03H
              RET
TECLA4:      MOV      A,#04H
              RET
TECLA5:      MOV      A,#05H
              RET
TECLA6:      MOV      A,#06H
              RET
TECLA7:      MOV      A,#07H
              RET
TECLA8:      MOV      A,#08H
              RET
TECLA9:      MOV      A,#09H
              RET
TECLAA:      MOV      A,#0AH
              RET
TECLAB:      MOV      A,#0BH
              RET
TECLAC:      MOV      A,#0CH
              RET
TECLAD:      MOV      A,#0DH
              RET
TECLAE:      MOV      A,#0EH
              RET
TECLAF:      MOV      A,#0FH
              RET
TECLA0:      MOV      A,#00H
              RET

```

Exercício

Fazer um programa em Assembly do 8051 que interfaceado a um Teclado Matricial

- a) Diretamente ao Microcontrolador**
- b) Mapeado em Memória conforme esquema apresentado em aula**

Comande um Motor de Passo conectado à Porta P1 (P1.0 = Clock, P1.1 = Direção).

- 1. O Número de passos (de 01 a 99) deve entrar pelo Teclado Matricial seguido da direção (A = Anti-horária e B = Horária). Após os dados de número de passos e direção terem sido inseridos, o programa aguarda uma tecla de início, ou Enter (usar a letra F).**
- 2. Quando o Motor terminar de dar os passos solicitados no item 1, deve parar e aguardar novos comandos de Número de Passos e Direção.**
- 3. Simular colocando um contador em paralelo com o clock do motor para verificar a contagem dos pulsos.**

Read-Modify-Write Instructions

- Algumas instruções que leem as Portas do 8051 leem os Latches e outras leem os Pinos.
- As instruções que leem os Latches ao invés dos Pinos são as que leem um valor, possivelmente alteram e então reescrevem no Latch.

Instruções que leem o Latch

- ANL P1,A
- ORL P2,A
- XRL P3,A
- JBC P1.1,LOOP
- CPL P3.0
- INC P2
- DEC P1
- DJNZ P3,LOOP2

- MOV P1.0,C
- CLR P2.3
- SETB P3.0

• Estas instruções são chamadas de Read-Modify-Write Instructions

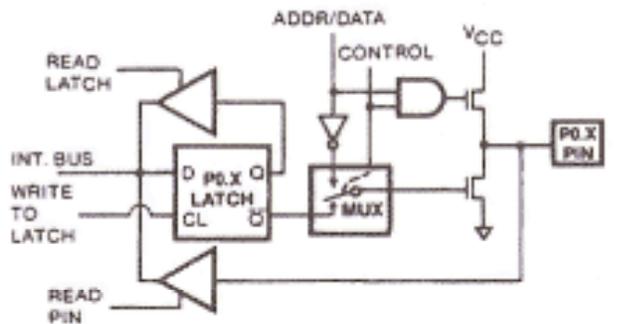
• Outras instruções, tais como,

JB P2.0, LABEL

JNB P1.4, LABEL

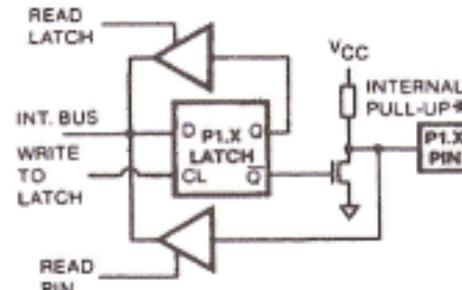
Testam diretamente o pino em questão.

Latches e Pinos das Portas



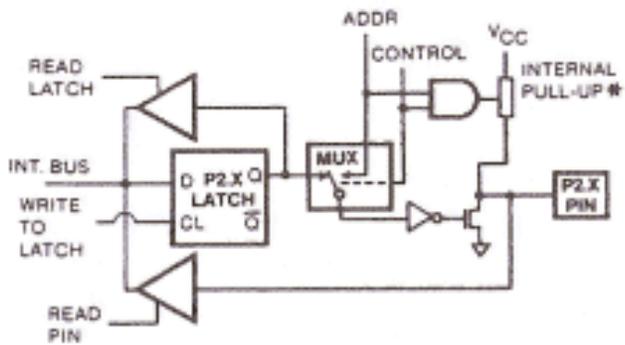
270252-2

A. Port 0 Bit



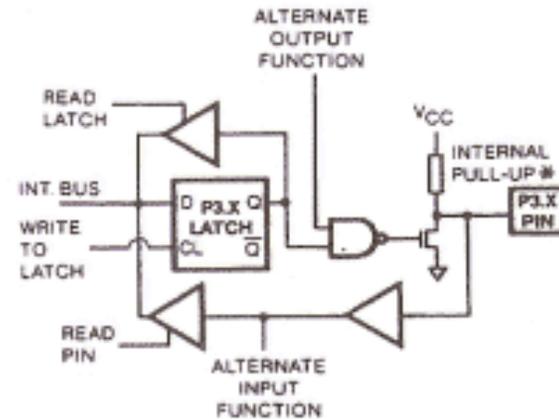
270252-3

B. Port 1 Bit



270252-4

C. Port 2 Bit



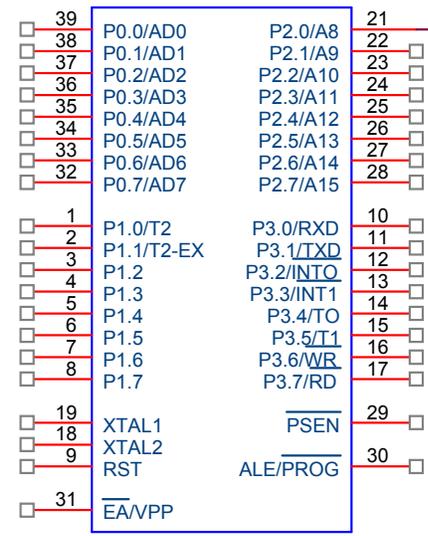
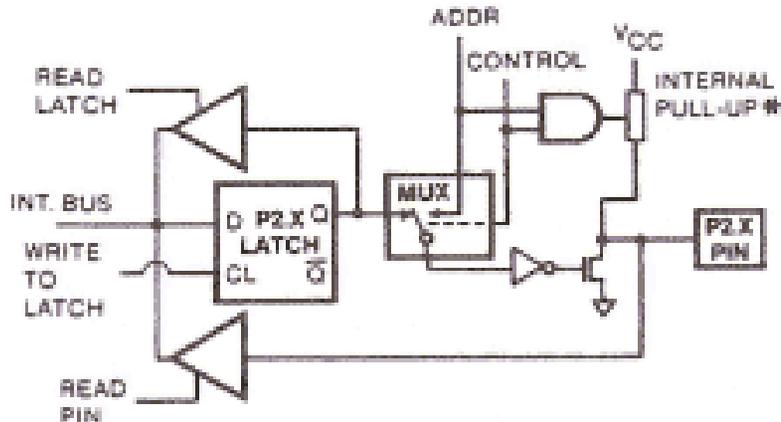
270252-5

D. Port 3 Bit

- Por exemplo: Se no Bit 0 da Porta P2 tiver um transistor.

- A instrução SETB P2.0 faz o Pino 21 do 8051 ir para nível lógico 1, conduzindo o transistor.

- Se for executada posteriormente a instrução JB P2.0, LABEL
- O valor que a instrução veria se fosse lido o Pino 21 seria nível lógico Zero, devido à condução do transistor (VBE).



AT89S52

