

Aula 5 - Interrupções

Interrupções por software

- O 8085 tem **8** instruções de **recomeço por software** (restart), que são: RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6 e RST 7.
- Estas instruções são para chamadas de subrotinas que são usadas com frequência.
- O efeito de uma instrução RST é o mesmo de uma chamada de subrotina, com uso da pilha para guardar o endereço de retorno.

Interrupções por software

- Cada instrução RST desvia o processamento para um endereço predeterminado
- A chamada de uma RST é feita com apenas 1 byte de código, ao contrário de uma chamada de subrotina normal, que é feita com 3 bytes de código
- Seu efeito é o mesmo que no caso de uma subrotina comum, quando uma instrução RET é encontrada o processamento volta para o programa principal.

Interrupções por software

Instrução	Efeito	Código de Operação	Posição do Vetor
RST 0	CALL 0000h	C7	0000h
RST 1	CALL 0008h	CF	0008h
RST 2	CALL 0010h	D7	0010h
RST 3	CALL 0018h	DF	0018h
RST 4	CALL 0020h	E7	0020h
RST 5	CALL 0028h	EF	0028h
RST 6	CALL 0030h	F7	0030h
RST 7	CALL 0038h	FF	0038h

Interrupções por software

- Há apenas 8 posições (8 bytes) disponíveis para cada instrução de recomeço.
- Normalmente essas posições vetoradas são usadas apenas para desviar para uma subrotina que é frequentemente usada e que não caberia em apenas 8 posições.
- Uma instrução de desvio incondicional **JMP addr** é então usada.
- A chamada da subrotina através de uma dessas instruções de recomeço é mais eficiente que a chamada direta da subrotina.

Interrupções por hardware

- O 8085 possui 4 instruções de **recomeço por hardware (Interrupções Externas)** e **1 instrução para expansão das interrupções de hardware**.
- Essas instruções são: **TRAP** (pino 6), **RST 7.5** (pino 7), **RST 6.5** (pino 8) e **RST 5.5** (pino 9).
- Quando qualquer desses pinos é ativado, os circuitos internos do 8085 produzirão uma chamada de subrotina (**CALL**) por hardware que desviarão o processamento para o endereço predeterminado.

Interrupções por hardware

Instrução	Efeito	Código de Operação	Posição do Vetor	Pino
TRAP	CALL 0024h	não tem	0024 h	6
RST 5.5	CALL 002Ch	não tem	002C h	9
RST 6.5	CALL 0034h	não tem	0034 h	8
RST 7.5	CALL 003Ch	não tem	003C h	7

Interrupções por hardware

- As operações de entrada e saída por interrupção usam uma das interrupções por hardware dadas na tabela, de forma que, **quando o periférico está pronto para a transferência de dados a interrupção selecionada é acionada**, permitindo a transferência de dados de uma forma mais rápida que a transferência Programada.

Interrupções por hardware

- As interrupções externas têm uma **ordem de prioridade**, de forma que, no caso de dois ou mais pedidos simultâneos de interrupção, a ordem de atendimento é:

Interrupção	Prioridade	Posição do Vetor
TRAP	1	0024 h
RST 7.5	2	003C h
RST 6.5	3	0034 h
RST 5.5	4	002C h
INTR	5	Nenhuma

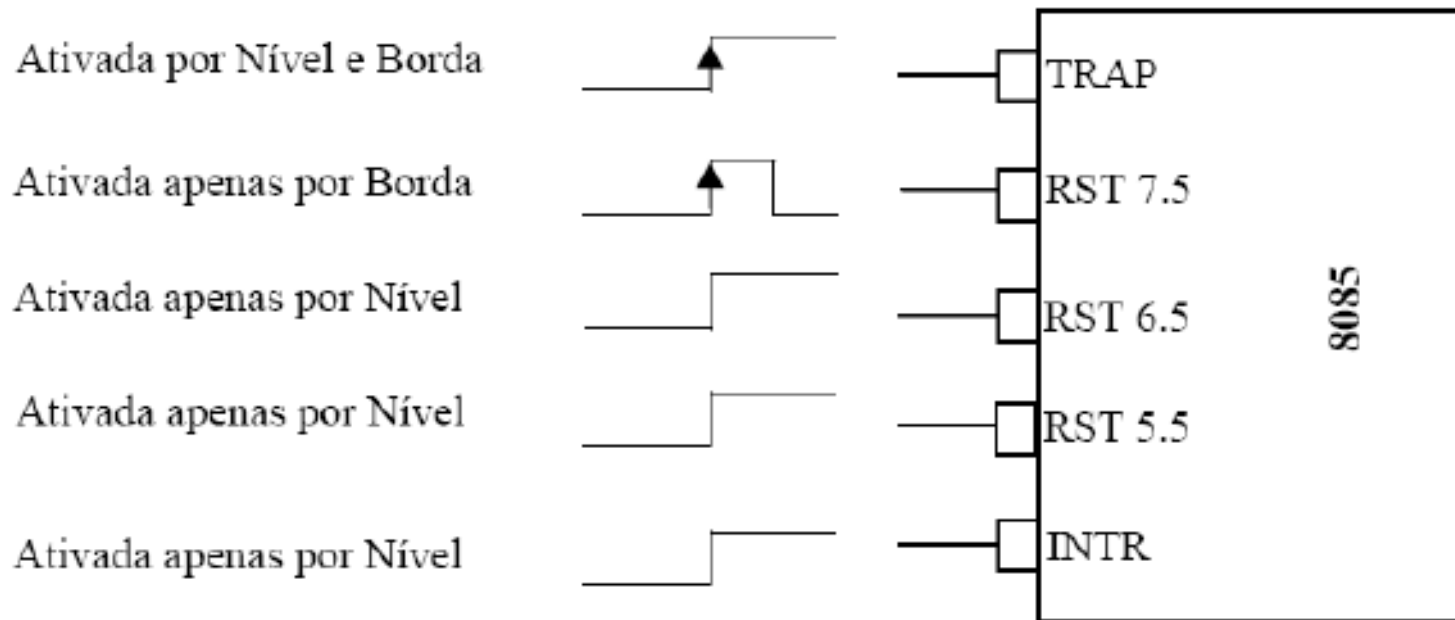
Interrupções por hardware

- Quando um pedido de interrupção é feito, o microprocessador termina a execução da instrução em curso para somente depois atender a interrupção pendente.
- A interrupção **INTR** é um canal que permite expandir a capacidade de interrupção (usado um CI extra para essa expansão).

Interrupções por hardware

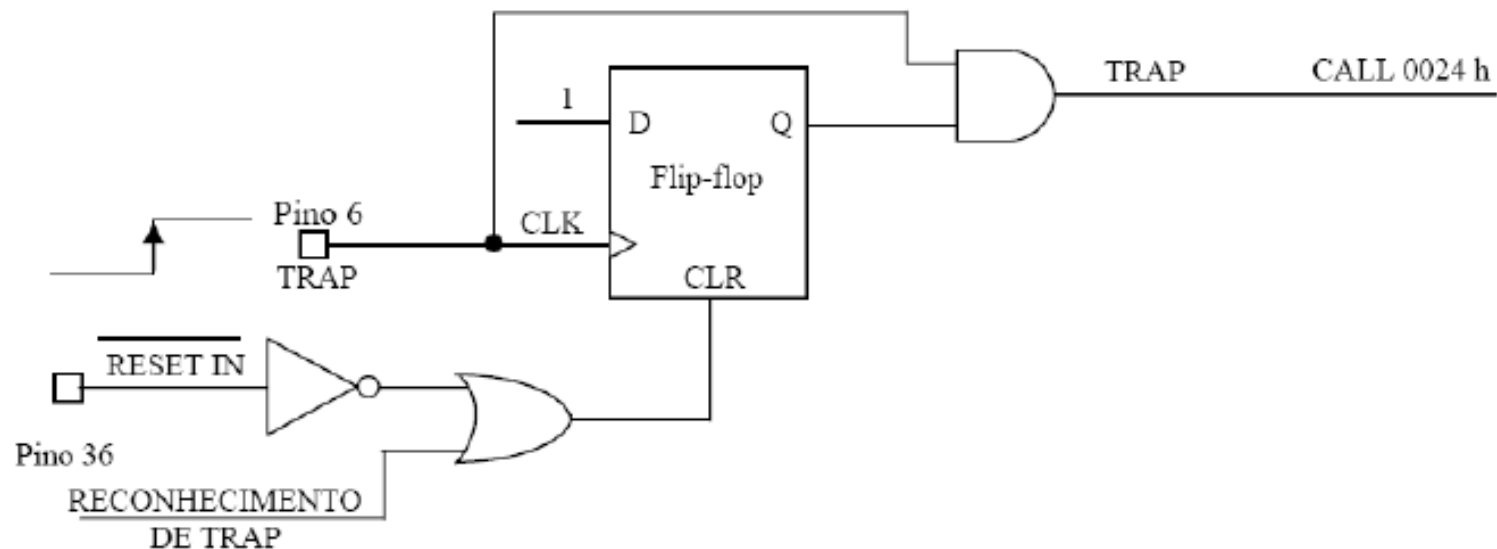
- As interrupções são acionadas por sinais externos nos pinos correspondentes.
- Esses sinais de acionamento de interrupção podem ser nível alto no pino (acionamento por nível) ou transição do nível baixo para o nível alto (acionamento por borda, ou por transição).

Interrupções por hardware



Interrupções por hardware

- Circuito de Interrupção

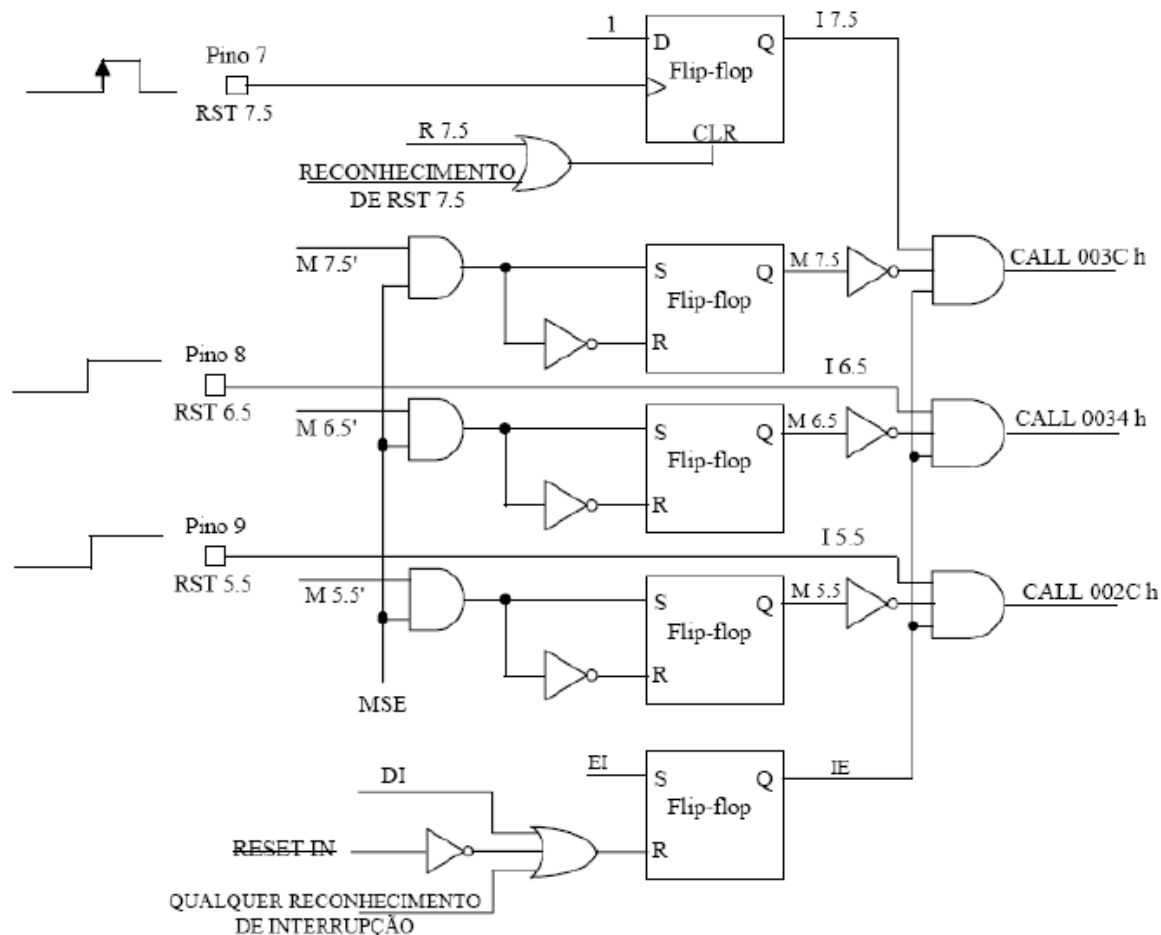


Interrupções por hardware

- O **sinal alto de *clock* habilita a porta AND** na saída do Flip-Flop, enquanto a **transição do *clock* do nível baixo para o nível alto faz com que a saída do Flip-Flop assumo o valor 1** e, conseqüentemente, a saída geral do circuito da interrupção **TRAP** assume o valor 1, fazendo com que o processamento seja desviado para o endereço 0024 h.
- Um sinal baixo no pino RESET IN ou um sinal alto de RECONHECIMENTO DE TRAP limpa o pedido de interrupção.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5**, **RST 6.5** e **RST 5.5**

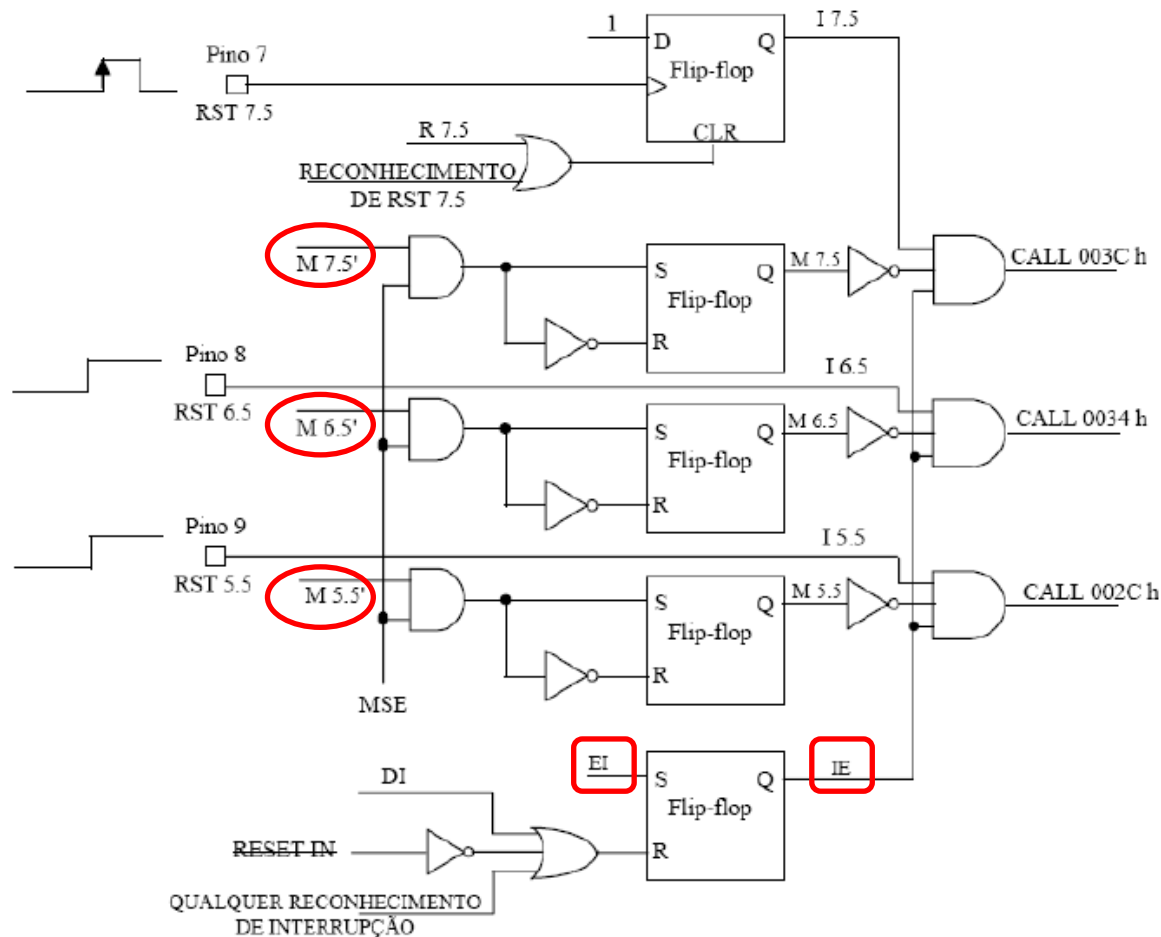


Interrupções por hardware

- As portas AND de saída dos circuitos das interrupções tem três entradas.
- Se o sinal IE esta alto, todas as interrupções estão habilitadas. Esse sinal fica alto quando a instrução EI (entrada do flip-flop) é executada.
- Se nenhuma das interrupções estiver com máscara, então os sinais M7.5, M6.5 e M5.5 permanecem em nível baixo e, conseqüentemente as respectivas entradas das portas AND estarão altas. Dessa forma, todas as interrupções estarão aptas a serem acionadas.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5**, **RST 6.5** e **RST 5.5**

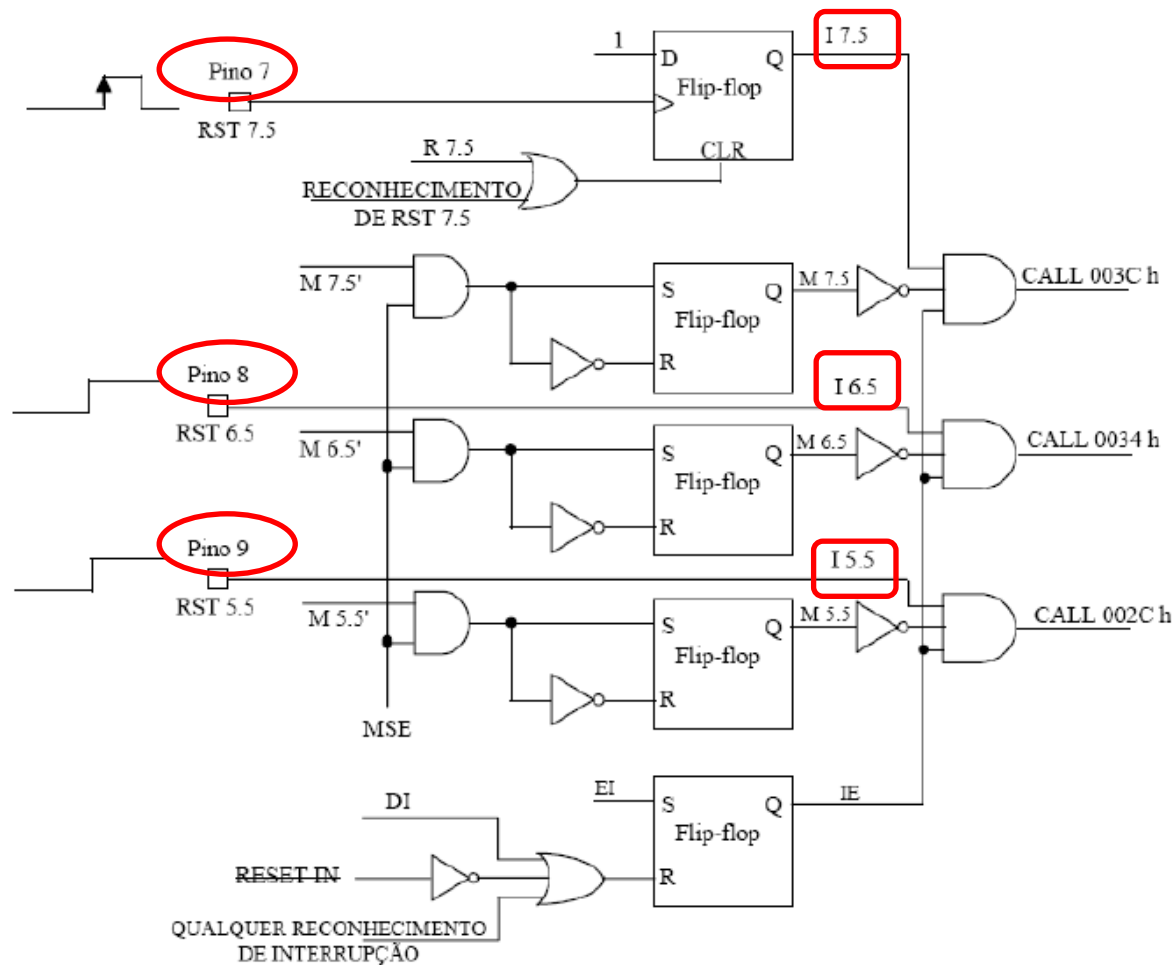


Interrupções por hardware

- Uma **borda de subida no pino 7** faz com que a **saída 17.5** do flip-flop correspondente à interrupção RST 7.5 **fique alta**, acionando essa interrupção.
- Um **nível alto no pino 8** faz com que a interrupção **RST 6.5 seja executada** e um nível alto no **pino 9** faz com que a interrupção **RST 5.5 seja executada**.
- Os sinais **17.5, 16.5 e 15.5** quando estão em **nível alto** indicam que **há interrupção pendente**.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5, RST 6.5 e RST 5.5**

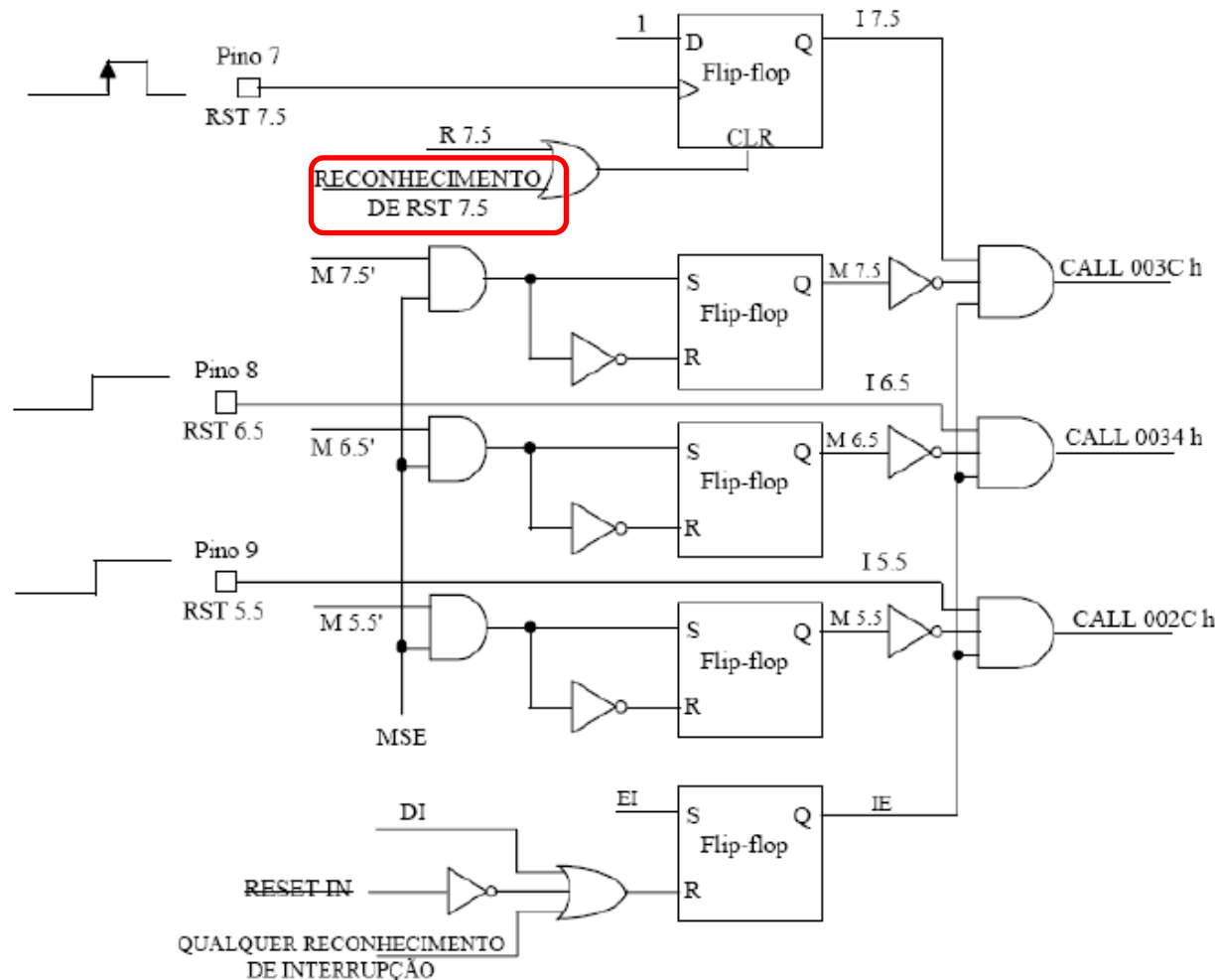


Interrupções por hardware

- Um sinal **R7.5** alto limpa uma interrupção **RST 7.5** pendente. Um sinal de **Reconhecimento de interrupção RST 7.5** também faz com que a saída do flip-flop seja zerada, **limpando qualquer interrupção RST 7.5** pendente.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5, RST 6.5 e RST 5.5**

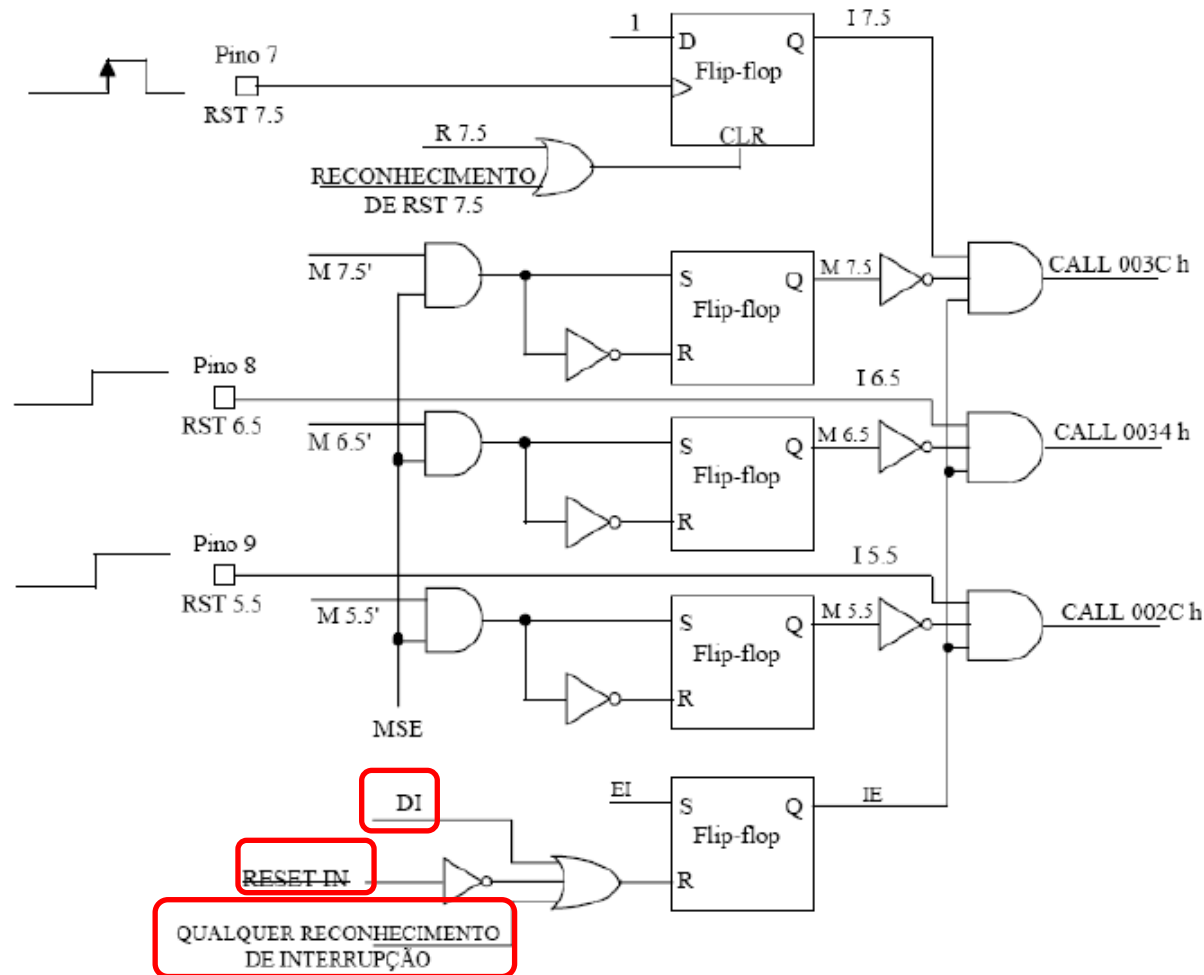


Interrupções por hardware

- O **bloqueio total das interrupção** pode ser feito de três maneiras distintas:
- (a) **através da instrução DI (Desable Interrupt)**, que desabilita todas as interrupções, com exceção da TRAP;
- (b) através de um **sinal baixo no pino 36 (RESET IN)**;
- (c) através de um **sinal alto de reconhecimento de qualquer interrupção**.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5**, **RST 6.5** e **RST 5.5**

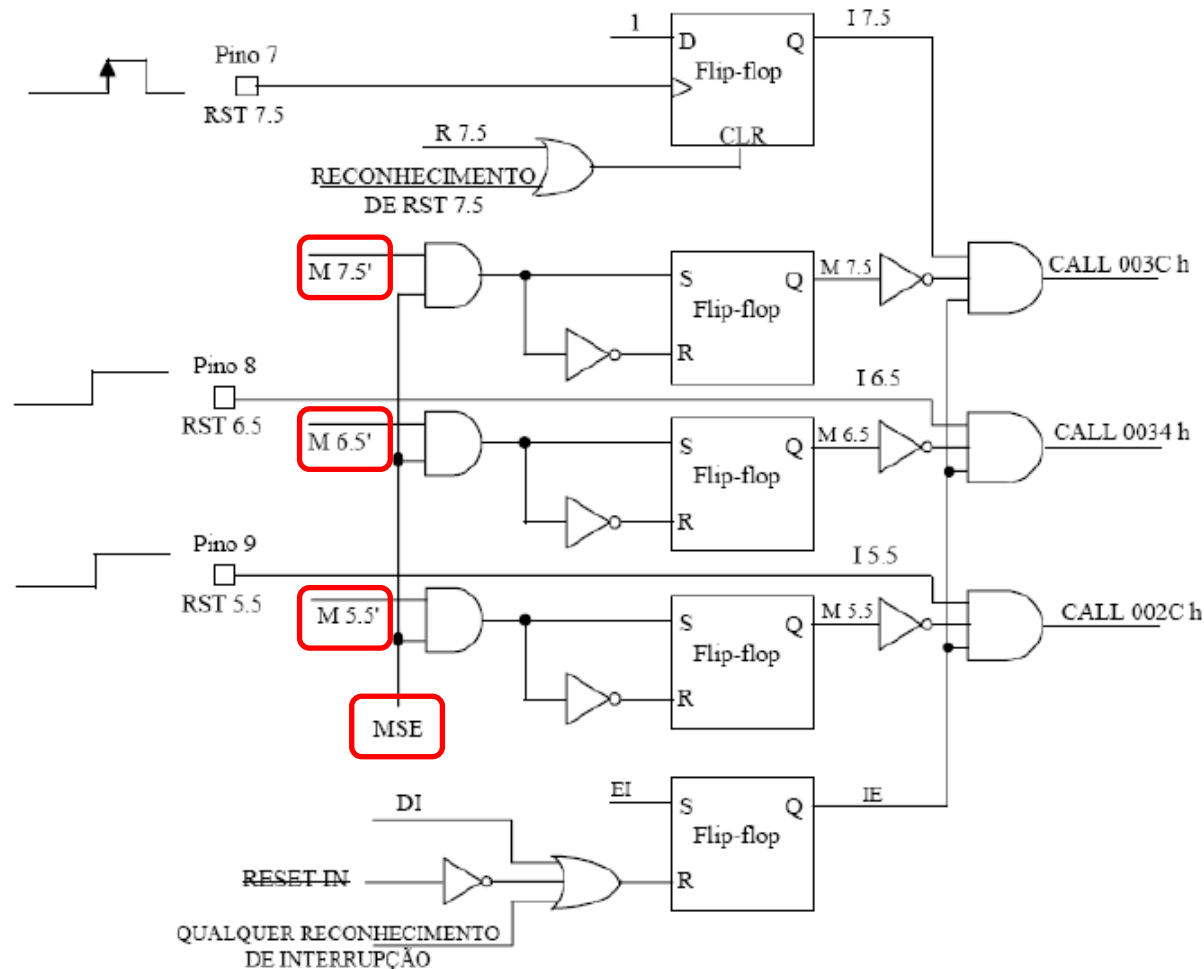


Interrupções por hardware

- Um sinal alto **MSE (Mask Set Enable)** é necessário para habilitar qualquer das máscaras de interrupção.
- Desejando-se mascarar (não permitir a execução) a interrupção RST 7.5, o sinal M7.5' deve ser levado ao nível alto.
- Desejando-se mascarar a interrupção RST 6.5 o sinal M6.5' deve estar em nível alto.
- Se a interrupção que se deseja mascarar é a RST 5.5, o sinal M5.5' deve ficar em nível alto.

Interrupções por hardware

- Circuito para as interrupções **RST 7.5**, **RST 6.5** e **RST 5.5**



Máscara de Interrupção

- A **máscara de interrupção** é usada para que as interrupções sejam usadas de forma seletiva, onde as **interrupções não desejadas sejam bloqueadas**.
- A máscara de uma interrupção mascarável só é **reconhecida após a execução do comando **SIM**** (Set Interrupt **M**ask = Setar a Máscara de Interrupção).
- Antes da execução deste comando alguns valores do circuito de interrupção devem ser carregados no acumulador de forma que a instrução SIM faz a transferência desses valores para os pontos necessários.

Máscara de Interrupção

Acumulador	SOD	SOE	×	R7.5	MSE	M7.5'	M6.5'	M5.5'
	Serial Output Data - Dado da Saída Serial	Serial Output Enable - Habilita Saída Serial	Irrelevante	Zera flip-flop da RST 7.5	Mask Set Enable - Habilita Setagem de Máscara	Seta máscara da RST 7.5	Seta máscara da RST 6.5	Seta máscara da RST 5.5

Máscara de Interrupção

- Exemplo: Desejando-se habilitar apenas a interrupção **RST 5.5** e mascarar as demais, o conteúdo do acumulador antes da instrução SIM é dado a seguir:

	SOD	SOE	×	R7.5	MSE	M7.5'	M6.5'	M5.5'
A:	0	0	0	0	1	1	1	0

Máscara de Interrupção

- A execução do comando **RIM** (**R**ead **I**nterrupt **M**ask) faz com que o estado das máscaras de interrupção sejam transferidos para o acumulador.

	SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
A:	0	1	0	0	1	1	0	0

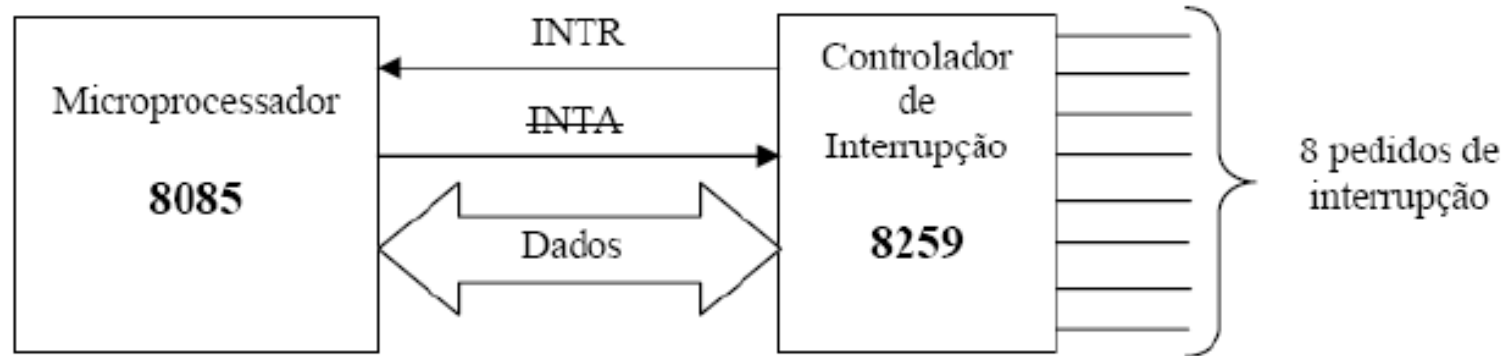
Máscara de Interrupção

A:

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	1	0	0	1	1	0	0

- Significado:
- Há uma interrupção RST 7.5 pendente (I7.5 = 1)
- As interrupções estão habilitadas (IE = 1)
- A máscara da interrupção RST 7.5 está setada, significando que ela não será atendida

Ampliação das interrupções



- O Controlador de interrupção (8259) permite expandir a capacidade de interrupção do microcontrolador 8085.
- Cada CI 8259 permite acrescentar em 8 as linhas de interrupções, permitindo, portanto a conexão de um número maior de periféricos ao microprocessador.

Ampliação das interrupções

- O 8259 contém os endereços iniciais de oito subrotinas de atendimento de interrupção, um para cada dispositivo periférico.
- Para atender a interrupção ele envia uma *CALL endereço* para o 8085, o que faz que o 8085 interrompa o processamento normal para o atendimento da interrupção solicitada.

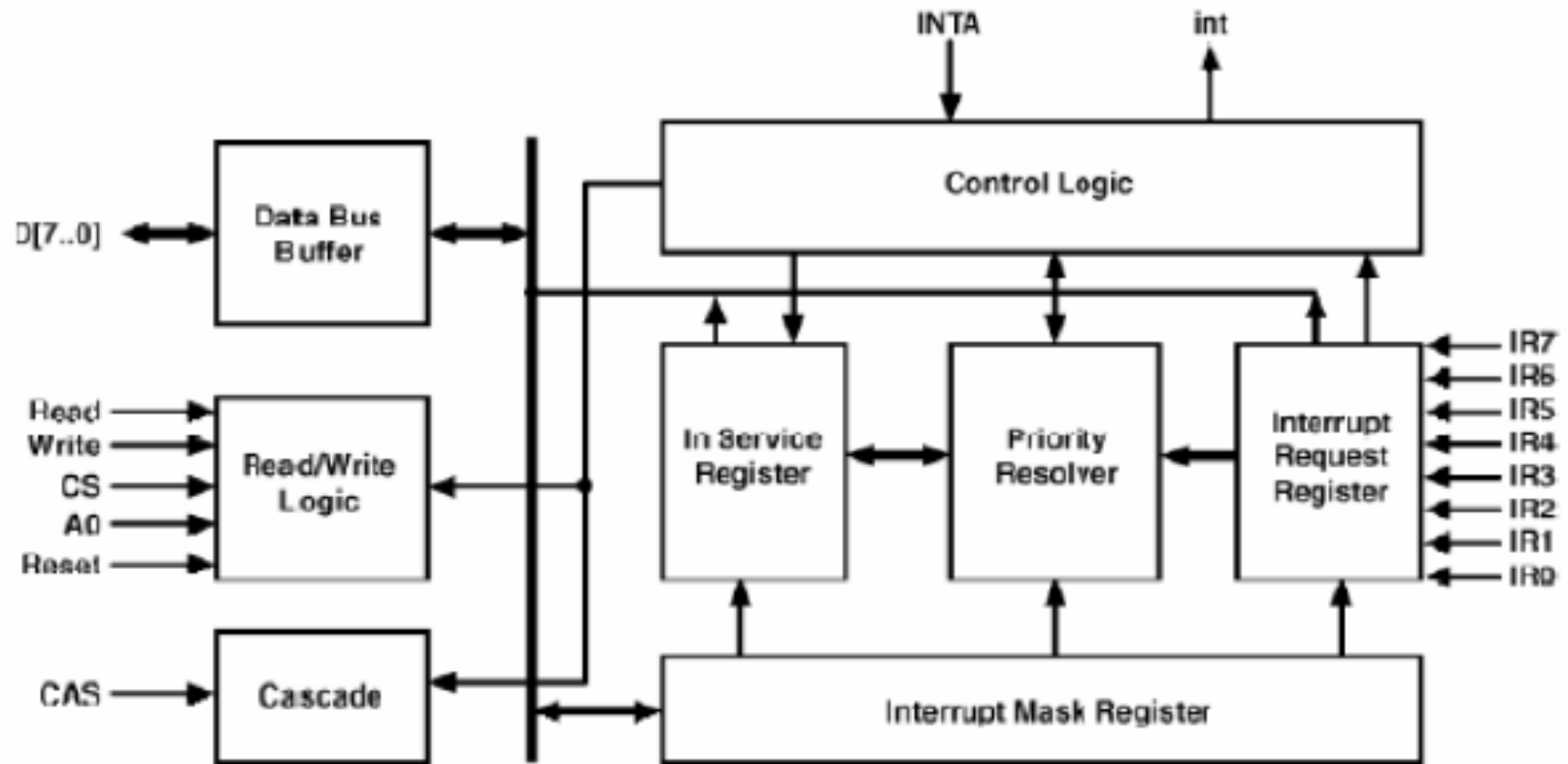
Ampliação das interrupções

- A cada solicitação de interrupção o controlador de interrupção envia um pulso alto para o 8085 através do pino INTR.
- O 8085 envia de volta um sinal baixo INTA (Interrupt Acknowledge) de reconhecimento de interrupção.
- O 8259 envia o código de operação da subrotina, ao qual o 8085 responde com outro sinal INTA baixo solicitando o byte inferior do endereço da subrotina.

Ampliação das interrupções

- Após o envio do byte inferior de endereço um outro sinal INTA baixo é enviado pelo 8085 solicitando o byte superior do endereço da subrotina.
- A seguir, o conteúdo do contador de programa (PC) é enviado para a pilha e o processamento é desviado para o endereço indicado

Ampliação das interrupções



Ampliação das interrupções

- Procedimento para uso de interrupções através do 8259:
- 1 - O nível lógico de um ou mais dos pinos IR (IR0 a IR7) é levado a 1, setando o correspondente registrador de pedido de interrupção IRR;
- 2 - O 8259 envia um pulso alto para o 8085 através do pino INTR (pino 10);
- 3 - O 8085 envia de volta um sinal baixo INTA (Interrupt Acknowledge) de reconhecimento de pedido de interrupção;
- 4 - Após receber o sinal de OK, o 8259 recebe o bit ISR (In Service Register) do pedido de maior prioridade, limpa o registro de interrupção IRR correspondente e envia para o 8085 o código de chamada de subrotina;

Ampliação das interrupções

- 5 - O 8085 responde com outro sinal INTA baixo solicitando o byte inferior do endereço da subrotina;
- 6 - O 8259 envia o byte inferior de endereço correspondente ao pedido de interrupção;
- 7 - Um outro sinal INTA baixo é enviado pelo 8085 solicitando o byte superior do endereço da subrotina;
- 8 - O 8259 envia o byte superior do endereço da subrotina para o 8085 e, em seguida, limpa o bit ISR correspondente;
- 9 - A seguir, o conteúdo do contador de programa (PC) é enviado para a pilha e o processamento é desviado para o endereço indicado

Dispositivos de Entrada e Saída

- Dispositivos de Entrada e Saída (ou dispositivos de I/O) são necessários para possibilitar a comunicação entre a CPU e o resto do sistema.
- Há basicamente duas formas de comunicação entre o computador e um periférico: comunicação serial e comunicação paralela.

Dispositivos de Entrada e Saída

- **Comunicação Serial:** quando é enviado um bit de cada vez.
- Exemplo: comunicação via linha telefônica (através de um modem), no mouse, câmara fotográfica digital, impressora serial, instrumentos eletrônicos, agendas eletrônicas.
- A taxa de transmissão é medida em bit por segundo (bps ou bit/s).

Dispositivos de Entrada e Saída

- **Comunicação Paralela:** quando é enviado um grupo de bits, simultaneamente.
- Exemplo: comunicação entre um computador e uma impressora paralela.
- A taxa de transmissão é, em geral, medida em bytes por segundo.
- A transmissão paralela é utilizada em substituição à transmissão serial quando a distância entre transmissor/receptor é pequena e deseja-se maior velocidade de transferência.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Técnicas de hardware e software utilizadas para controlar o fluxo de dados entre o computador e seus periféricos.
- Os três métodos mais frequentemente encontrados são varredura, interrupção e DMA.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- **Varredura** (ou E/S por consulta ou *Pooling*):
- Técnica de comunicação na qual o **processador interroga periodicamente o periférico para determinar seu estado**;
- Pode ser síncrona ou assíncrona;
- Desvantagem: perda de tempo: o processador abandona o programa principal para fazer a varredura mesmo se nenhum periférico deseja serviço.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

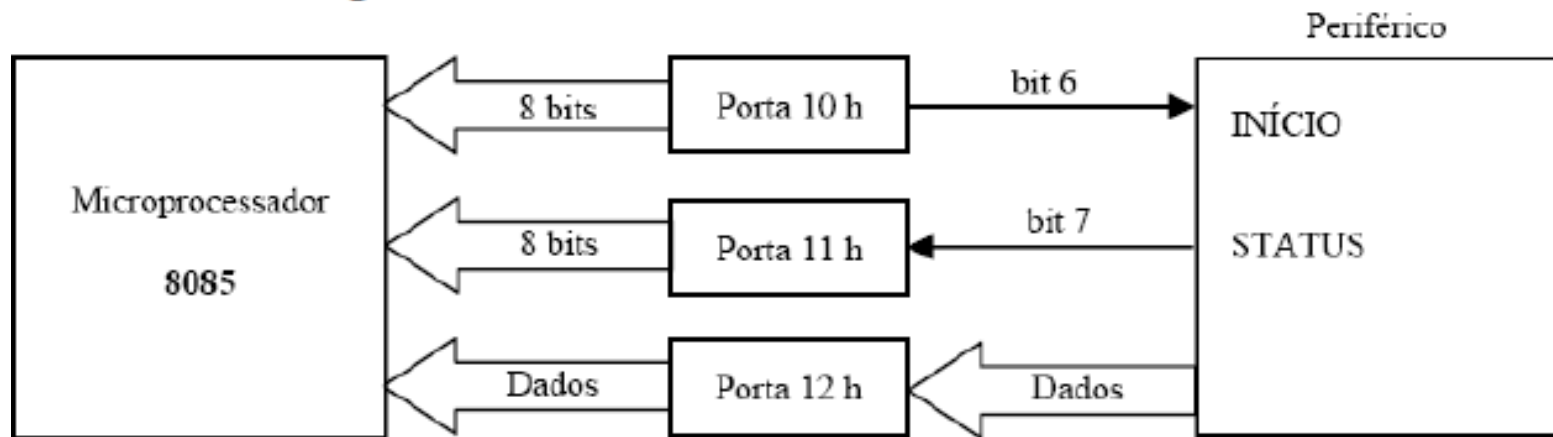
- **Interrupção:**
- Técnica de comunicação na qual o **processador somente é ocupado se há pedido de serviço de algum periférico;**
- Mecanismo assíncrono;
- Transferência de dados é feita por software (mais lento que DMA);

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- **DMA** (*Direct Memory Access*):
- Técnica de comunicação na qual **não se interrompe o processamento do microprocessador (se a instrução em curso não faz acesso ao barramento)**;
- O DMAC (controlador de DMA) envia o sinal HOLD requerendo o barramento e a CPU responde com HLDA. O DMAC passa a controlar as linhas de dados, linhas de endereço, e sinais de controle (RD, WR, etc.);
- A transferência de dados é feita por hardware: é particularmente interessante quando necessita-se transferência de dados em alta velocidade.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Exemplo 1: Entrada Programada

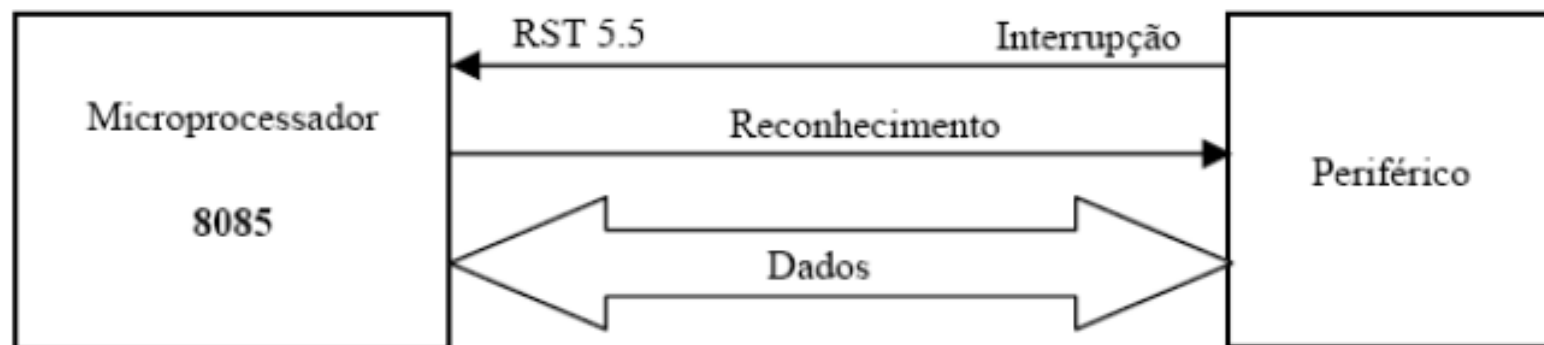


MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Quando a CPU está pronta para receber dados ela envia um bit alto de INÍCIO (bit 6) para periférico.
- Quando os dados estão prontos, um bit alto de STATUS (bit 7) é enviado de volta à CPU.
- Em seguida os dados são transferidos do periférico para a CPU.
- Problema: a CPU fica aguardando o bit alto de STATUS, o que pode significar um tempo considerável, uma vez que a maior parte do tempo é gasto esperando o periférico terminar o processamento de dados.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Exemplo 2: Transferência por interrupção

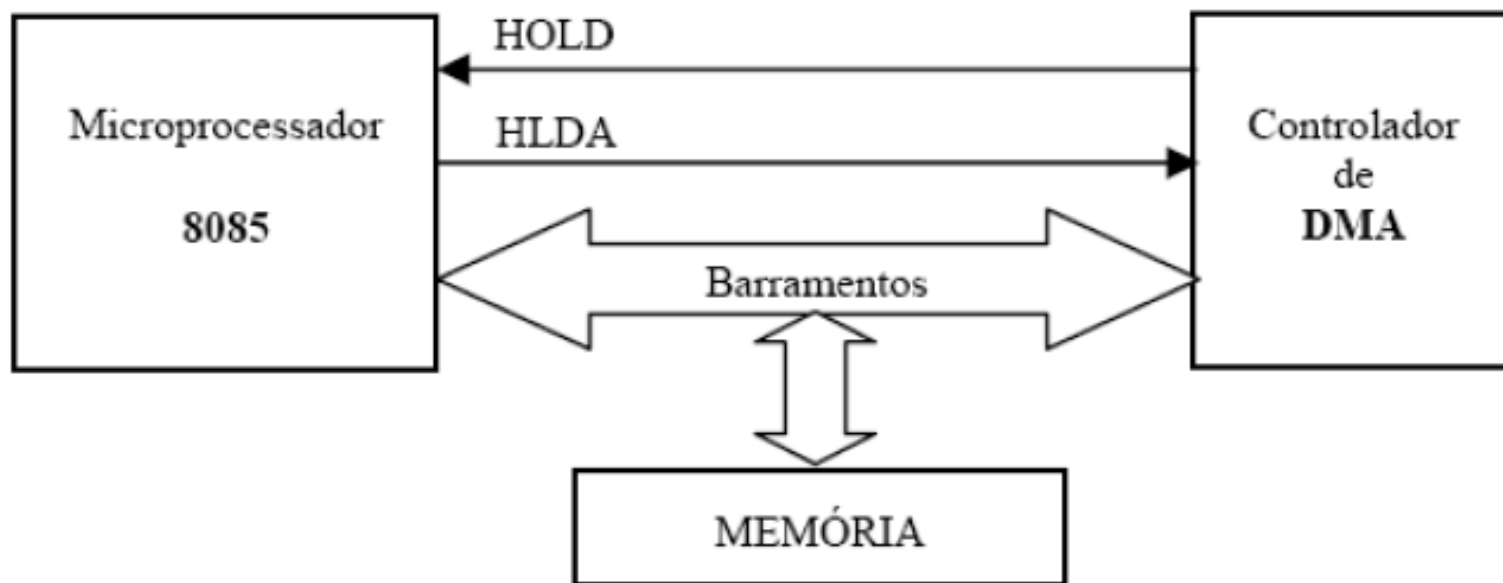


MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- A CPU não precisa aguardar o processamento do periférico.
- Quando o periférico está pronto para transferir ou receber os dados um bit de INTERRUPÇÃO é enviado à CPU.
- O processamento normal da CPU é interrompido para atender o pedido externo.
- Um bit de RECONHECIMENTO de interrupção é enviado para o periférico, havendo então a transferência dados para a CPU.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Exemplo 3: Transferência por DMA



MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- As transferências de dados são mais rápidas porque o **acumulador não é utilizado e não há necessidade de leituras da memória ROM.**
- Os **dados vão diretamente da memória para o periférico, ou vice-versa.**
- O controlador de DMA é fisicamente conectado aos barramentos de dados, endereços e controle, dependendo apenas da liberação dos mesmos pela CPU.

MÉTODOS DE CONTROLE DE ENTRADA E SAÍDA

- Quando o **periférico está pronto para a transferência de dados**, um sinal de HOLD alto é enviado à CPU solicitando a liberação do barramentos.
- A **CPU responde com um sinal HLDA**, informando que o processamento foi interrompido e os **barramentos liberados**.
- O controlador de **DMA realiza então a transferência de dados em alta velocidade** e **devolve o controle dos barramentos para a CPU**, enviando um **sinal de HOLD baixo**.