Arquitetura de Computadores

Controlador de interrupções

João Pedro Patriarca (jpatri@cc.isel.ipl.pt)

ISEL, ADEETC, LEIC

Conceito

- Uma interrupção serve para “parar” o fluxo de execução normal do CPU para dar resposta célere a um ou mais eventos externos
  - A alternativa é estar a testar continuadamente no programa principal a ocorrência de evento[s] externo[s] complicando a lógica do mesmo
- O CPU disponibiliza uma ou mais entradas através das quais um dispositivo externo solicita interrupção (IRQ – *Interrupt Request*)
- O CPU avalia o estado da entrada IRQ no final da execução de cada instrução
- O CPU atende um pedido de interrupção executando a rotina de interrupção (ISR – *Interrupt Service Routine*)
- O CPU quando conclui a execução da ISR retorna para o programa onde foi interrompido

Exemplos de aplicação

- Implementação de relógios de sistema:
  - Num sistema multiprocesso/multithread, impede que um determinado processo monopolize o CPU
- Resposta em tempo real a eventos urgentes
- Retirar o CPU de estados de baixo consumo (num estado onde tipicamente o CPU não está a executar programa)

Implicações de um sistema de interrupções

- Capacidade de inibir ou permitir o atendimento de um pedido de interrupção
- No processo de atendimento de um pedido de interrupção é necessário reter algum estado interno do CPU para poder retornar posteriormente ao programa interrompido (com o mesmo estado)
- Impedir o atendimento de novos pedidos de interrupção durante o processamento da interrupção em curso
- Vetorizar a[s] ISR[s] como rotina[s] de interrupção

Permissão/inibição de um pedido de interrupção

- Tipicamente é controlado por uma *flag* do CPU
- Após *reset*, esta *flag* é iniciada desativa (inibindo o CPU de atender pedidos de interrupção)
- No processo de atendimento de um pedido de interrupção, esta *flag* é colocada desativa automaticamente para impedir o CPU de atender o mesmo ou outros pedidos de interrupção durante a execução da ISR
  - No retorno da ISR, o valor da *flag* deve ser reposto
  - O retorno da ISR e a reposição do valor da *flag* tem de ser feito de forma indivisível (atómica)

- P16: a *flag* de permissão/inibição é a *flag* I presente no registo CPSR

Salvaguarda do estado do CPU aquando a interrupção

- Depois de retornar da ISR, todos os registos do CPU devem ter os mesmos valores que tinham aquando o atendimento do pedido de interrupção
- Tipicamente, a retenção do valor do PC é realizado com o mesmo critério usado na chamada de funções
- Para os restantes registos, ou ficam ao cuidado do programador (empilhados no *stack*) ou existe um banco de registos correspondente à duplicação dos mesmos que é comutado automaticamente durante o processo de atendimento da interrupção

- P16:
  - O registo PC é retido no registo LR (iLR)
  - O registo CPSR é retido no registo SPSR
  - Os restantes ficam ao cuidado do programador

Vetorização para a ISR

- Consiste em afetar o PC com o endereço da ISR correspondente ao pedido de interrupção
- Pode ser realizado pelas alternativas seguintes:
  - Valor constante definido pela arquitetura
  - Leitura do barramento de uma instrução (Intel 8085)
  - Leitura do barramento de um vetor de interrupção (Intel x86)

- P16:
  - Existe apenas uma única entrada de IRQ e o P16 define o endereço 2 como o ponto de entrada para a ISR

Retorno da ISR

- De notar que o retorno e a reposição da *flag* de permissão/inibição tem de ser feito de forma atómica
- Pode ser feito através de uma instrução de reposição de *flags* com efeito retardado para possibilitar o retorno para o programa interrompido ainda com a *flag* inibida
- Pode ser feito por uma instrução específica que realiza as duas ações em simultâneo

- P16:
  - Instrução específica: movs pc, lr

Interrupções no P16

- Uma única entrada externa (IRQ) – ativa a 0. O dispositivo que solicita o pedido de interrupção retira o pedido quando esta for atendida
- Um único vetor de interrupção (endereço 2)
- Uma flag I para permissão/inibição
- Uma flag M para indicar o modo de execução (0 – programa normal / 1 – programa correspondente à ISR) presente no registo CPSR
- Adição de dois registos ao banco de registos:
  - LR (de interrupção)
  - SPSR (*Saved Program Status Register*)
- Retorno por instrução específica: movs pc, lr

Ações realizadas durante o atendimento de um pedido de interrupção no P16

- Copia o valor do registo CPSR para o registo SPSR
- Coloca a 1 a flag M do registo CPSR mudando o P16 para o modo interrupção. Tem como consequência a comutação da utilização do registo LR normal para o registo LR da interrupção
- Coloca a 0 a flag I do registo CPSR
- Copia o valor do registo PC para o registo LR (do modo interrupção)
- Coloca o valor 2 no registo PC

Definição da rotina de interrupção

```
    .section startup
    b _start            ; Entry point após reset
    ldr   pc, addr_isr  ; Entry point de interrupção
    …
addr_isr:
    .word isr           ; Endereço da ISR
    .text
main:
    …
    ; Habilita o atendimento de interrupções
    mov   r0, #IE_ENA
    msr   cpsr, r0
    …
; Rotina de interrupção
isr:
    …
    movs  pc, lr        ; Retorno da rotina de interrupção: repõe PC e CPSR
```

Convenções no âmbito de uma rotina de interrupção

Rotina de interrupção folha

```
isr:
   ; Preserva apenas os registos
   ; alterados na ISR
   ; (ex: R0, R1 e R4)
   push   r0
   push   r1
   push   r4
   …
   pop    r4
   pop    r1
   pop    r0
   movs   pc, lr
```

Rotina de interrupção não folha

```
isr:
   ; Preserva sempre registos R0-3 e outros
   ; alterados na ISR (ex: R0, R1 e R4)
   push   lr
   push   r0
   push   r1
   push   r2
   push   r3
   push   r4
   …
   bl     func1
   …
   pop    r4
   pop    r3
   pop    r2
   pop    r1
   pop    r0
   pop    lr      ; Retorno desempilhado sempre
   movs   pc, lr  ; para LR para retornar com MOVS
```