

Microprocessadores

Hugo Marcondes
hugo.marcondes@ifsc.edu.br

Aula 12

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

Interrupções

- Uma interrupção é um sinal responsável por interromper o fluxo de execução de um programa em um microprocessador
 - Associado com eventos assíncronos de entrada/saída
 - Ou condições de exceção do microprocessador
- O tratamento de interrupções é realizado geralmente por um suporte interno ao microprocessador e em conjunto com um Controlador de Interrupções (IC)

2 IFSC - Departamento Acadêmico de Eletrônica

Interrupções

- De forma geral, ao ocorrer uma interrupção, o microprocessador deve:
 - Salvar o Program Counter (PC) na pilha. Alguns processadores salvam outros registradores além do PC.
 - Executar um ciclo de reconhecimento da interrupção, de forma a recuperar informações sobre a interrupção ocorrida
 - Desviar o PC para um endereço específico associado a interrupção reconhecida, geralmente uma ISR - Interrupt Service Routine.

3 IFSC - Departamento Acadêmico de Eletrônica

Interrupções

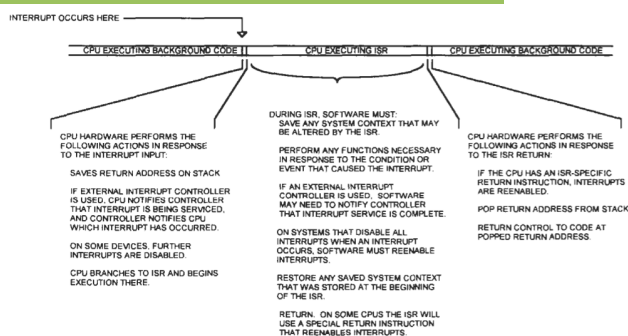
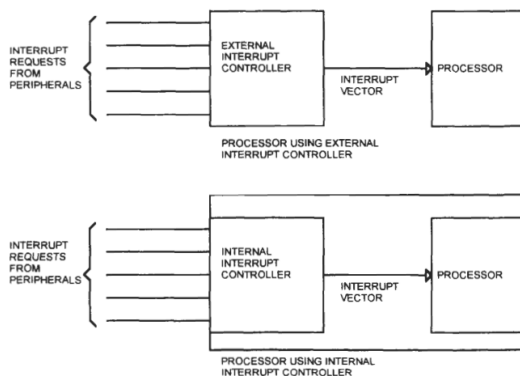


Figure 5.1
Interrupt Processing.

4 IFSC - Departamento Acadêmico de Eletrônica

Interrupções



5 IFSC - Departamento Acadêmico de Eletrônica

Tabela de Interrupções

- O microprocessador deve implementar uma tabela de interrupções, também conhecida como vetor de interrupções.
- Esta tabela pode conter:
 - Instruções que são executadas
 - Dados que são interpretados como endereços para as rotinas de tratamento de interrupção
- A tabela de interrupções pode estar localizada em um endereço fixo do mapa de memória do processador, ou ser programada.

6 IFSC - Departamento Acadêmico de Eletrônica

Sensibilidade das Interrupções

- As interrupções podem ser
- Ativas por borda:
 - A interrupção é sinalizada quando ocorre a transição de estado lógico do sinal, podendo ser:
 - Ativa por borda de subida
 - Ativa por borda de descida
- Ativas por nível
 - A interrupção é sinalizada quando ocorre um estado de ativação no sinal, podendo ser:
 - Ativo em nível baixo
 - Ativo em nível alto

7 IFSC - Departamento Acadêmico de Eletrônica

Interrupções

- A configurabilidade da sensibilidade das interrupções dependem muito do hardware e microprocessador em questão
- Em alguns microprocessadores é possível também estabelecer prioridades no atendimento das interrupções
 - Neste caso, as interrupções também podem ser encadeadas, ou seja, uma interrupção de maior prioridade pode interromper uma de menor prioridade

8 IFSC - Departamento Acadêmico de Eletrônica

Interrupções no MIPS



- As interrupções na arquitetura MIPS são tratadas pelo Coprocessador 0
- Também conhecido como **System Control Coprocessor**
- Trata as principais funcionalidades requeridas pelo sistema operacional
- Possui uma série de registradores para controle destas funcionalidades
- No MARS, apenas os registradores relacionados com as exceções/interrupções estão implementados

9 IFSC - Departamento Acadêmico de Eletrônica

Coprocessador 0



- Os registradores são:
 - \$8 - BadVAddr: Endereço da instrução de violação
 - \$12 - Status: Possui o **bit para a ativação** de interrupções
 - \$13 - Cause: Indica a **fonte de interrupção**
 - \$14 - Exception Program Counter (EPC) - Indica o valor do PC no momento em que a exceção/interrupção ocorreu

10 IFSC - Departamento Acadêmico de Eletrônica

Interrupções no MIPS



- Uma vez habilitada, quando uma interrupção ocorre:
 - O PC atual é armazenado no registrador EPC
 - O processador é desviado para o endereço **0x80000180**
- Neste endereço (que está dentro da área de memória destinada ao SO) deverá conter o tratador de interrupção que deve:
 - Salvar o estado do processador (na pilha)
 - Verificar a origem da interrupção e executar o seu tratador.
 - Retornar ao ponto de parada através do EPC

11 IFSC - Departamento Acadêmico de Eletrônica

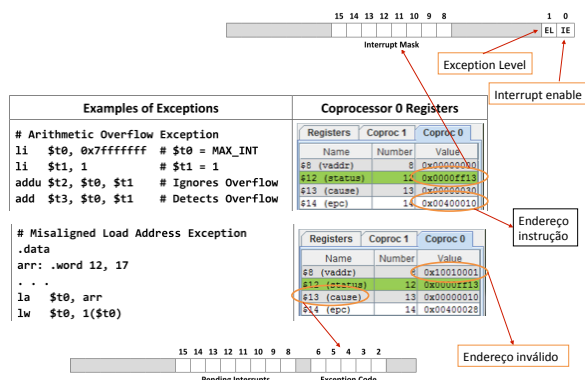
Registrador "Cause"



- O registrador \$13 do coprocessador 0 indica a causa da exceção / interrupção
- Bits 2-6 - Código de causa de exceções
- Bits 8-15 - Código de interrupções externas.
 - No MARS:
 - O bit 8 representa a interrupção do teclado
 - O bit 9 representa a interrupção do display

12 IFSC - Departamento Acadêmico de Eletrônica

System Control Coprocessor



13 IFSC - Departamento Acadêmico de Eletrônica

Habilitando Interrupções



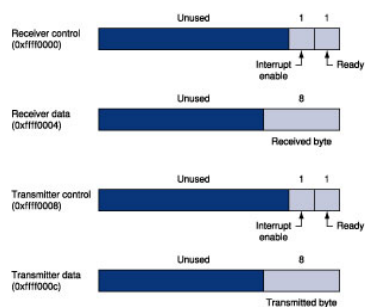
- O registrador Status do coprocessador 0, tem em seu bit menos significativo, a flag para controle das interrupções
- Desabilitar as interrupções no coprocessador 0
- Habilitar as interrupções nos periféricos
- Habilita as interrupções no coprocessador 0

14 IFSC - Departamento Acadêmico de Eletrônica

Interrupções do Teclado e Display



- As interrupções do teclado e display são configuradas através do bit 1 dos respectivos registradores de controle



15 IFSC - Departamento Acadêmico de Eletrônica

Vamos praticar



- Modifique o programa que utiliza o teclado / display realizando a programação por "pooling" para utilizar as interrupções.

16 IFSC - Departamento Acadêmico de Eletrônica

Gerenciamento de Entrada/Saída



- Polling vs Interrupções
- Dispositivos compartilhados entre processos ?
 - Recursos compartilhados
- Uso de interrupções prevê a implementação de buffers circulares para o gerenciamento de dados
 - Problema do produtor x consumidor
- Quais implicações temos em relação a programação de sistemas microprocessados?

Produtor x Consumidor



Producer:

```
shared int counter;
shared char buf[N];

int main()
{
    const int n = N;
    int in = 0;

    while (1) {
        while (counter == n);
        buf[in] = produce();
        in = ++in % n;
        counter++;
    }
}
```

Consumer:

```
shared int counter;
shared char buf[N];

int main()
{
    const int n = N;
    int out = 0;

    while (1) {
        while (counter == 0);
        consume (buf[out]);
        out = ++out % n;
        counter--;
    }
}
```

Condições de Corrida



Producer:

```
counter++;
load R1,[counter]
inc R1
store R1,[counter]
```

Consumer:

```
counter--
load R2,[counter]
dec R2
store R2,[counter]
```

	R1	R2	[counter]
0) P: load R1,[counter]	5	-	5
1) P: inc R1	6	-	5
2) C: load R2,[counter]	6	5	5
3) C: dec R2	6	4	5
4) C: store R2,[counter]	6	4	4
5) P: store R1,[counter]	6	4	6

Tratando condições de corrida



- As regiões críticas devem ser “protegidas”, evitando que haja concorrência de acesso aos dados compartilhados
- Mecanismos de sincronização de processos
- Travamento das interrupções

Gerenciamento de I/O



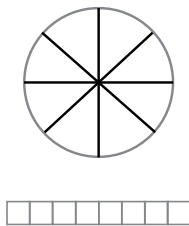
- Interrupções de modo geral devem ser tratadas da forma mais rápida possível
- Interrupções não reentrantes
- Tratar o mínimo possível em estado de interrupção
 - Sinalizar o hardware sobre o tratamento
 - Copiar os dados para buffers de longo prazo
 - Em geral buffers circulares (RingBuffers)

21 IFSC - Departamento Acadêmico de Eletrônica

RingBuffer



- Buffer de fácil gerenciamento para buffers de leitura/escrita para processo produtor/consumidor



```
write(10)
write(15)
read()
write(4)
read()
read()
write(23)
read()
```

22 IFSC - Departamento Acadêmico de Eletrônica

RingBuffer



- Implementar as funções para manipulação de um RingBuffer e utilizar o mesmo no exercício de entrada e saída com memória mapeada e interrupções

```
#define MAX_SIZE 16

typedef struct ringbuffer {
    int size;
    int rd;
    int wr;
    char buf[MAX_SIZE];
} t_ringbuffer;

void init(t_ringbuffer * rbuf){
    rbuf->size = 0;
    rbuf->rd = 0;
    rbuf->wr = 0;
}

bool rbuf_empty(t_ringbuffer * rbuf){
    if (rbuf->size == 0) {
        return 1;
    } else {
        return 0;
    }
}
```

```
bool rbuf_full(t_ringbuffer * rbuf){
    if (rbuf->size == MAX_SIZE)
        return 1;
    else
        return 0;
}

char read(t_ringbuffer * rbuf){
    char tmp = 0;
    if(!rbuf_empty(rbuf)){
        rbuf->size--;
        tmp = rbuf->buf[rbuf->rd];
        rbuf->rd = (rbuf->rd + 1) % MAX_SIZE;
    }
    return tmp;
}

bool write(t_ringbuffer * rbuf, char byte){
    if(!rbuf_full(rbuf)){
        rbuf->size++;
        rbuf->buf[rbuf->wr] = byte;
        rbuf->wr = (rbuf->wr + 1) % MAX_SIZE;
    }
    return 1;
}
```

23 IFSC - Departamento Acadêmico de Eletrônica