



Universidade Federal da Paraíba
Ciência da Computação – Campus I

Trabalho de Arquitetura II

Xor, Interrupção e Exceção na Mic-1

Discentes:

André Lopes;

Augusto Miguel;

Elias Victor;

Matheus Eduardo.

XOR na Mic-1

; XOR

```
xor1 | MAR = SP = SP - 1 | Carregar TOS para MDR
xor2 | MDR = NOT MDR | Inverter bits de MDR
xor3 | SP = SP - 1 | Carregar o próximo valor da pilha para MDR
xor4 | MDR = TOS AND MDR | Realizar AND entre TOS e MDR invertido
xor5 | SP = SP - 1 | Carregar o próximo valor da pilha para MDR
xor6 | MDR = MDR OR TOS | Realizar OR entre o resultado do AND e TOS
xor7 | TOS = MDR | Armazenar o resultado na pilha
```

- **Manipulação da Pilha:**

A primeira coisa a notar é que o algoritmo utiliza a pilha para armazenar valores temporários durante a execução das operações XOR. Isso é feito através da manipulação do ponteiro da pilha (SP). Antes de cada operação de leitura ou escrita na pilha, o ponteiro da pilha é decrementado ($SP = SP - 1$). Isso indica que o algoritmo está reservando espaço na pilha para armazenar valores temporários.

- **Operação XOR:**

O algoritmo executa operações XOR (exclusivo ou) em diferentes etapas:

- xor2: Inverte os bits do valor armazenado no topo da pilha (TOS), que é carregado em um registrador temporário (MDR).
- xor4: Realiza uma operação AND entre o valor atual do topo da pilha (TOS) e o valor invertido (resultado de xor2). O resultado é armazenado novamente em MDR.
- xor6: Realiza uma operação OR entre o resultado do xor4 e o valor original do topo da pilha (TOS).
- O resultado da operação XOR é armazenado de volta no topo da pilha (TOS) na instrução xor7.

Interrupção na Mic-1

; Endereços de memória simbólicos para representar o tratamento da interrupção ou exceção.

END_INT .data 0x1234

END_EXC .data 0x5678

; Interrupção

interrupt1 | IR = MDR | Carrega a instrução do endereço de memória apontado por MDR

interrupt2 | if (IR == INT) goto handle_interrupt | Verifica se é uma instrução de interrupção

interrupt3 | goto fetch | Caso contrário, continua o fetch normalmente

handle_interrupt: ; Rotina para tratar a interrupção

interrupt4 | SP = SP - 1 | Decrementa o ponteiro da pilha

interrupt5 | M[SP] = PC | Salva o endereço atual do PC na pilha

interrupt6 | PC = END_INT | Pula para o endereço de tratamento de interrupção

interrupt7 | goto fetch | Retorna ao fetch após o tratamento

- **Tratamento da Interrupção:**

- Quando uma interrupção ocorre (representada por INT), o programa salva o endereço atual do PC na pilha e pula para o endereço de tratamento da interrupção (END_INT). Isso é feito nas instruções interrupt4, interrupt5 e interrupt6.
- O SO (Sistema Operacional) é responsável por detectar e lidar com as interrupções. Ele garante que o contexto atual do programa seja preservado ao salvar o endereço do PC na pilha e controla o fluxo do programa para o tratamento adequado da interrupção.
- Após o tratamento da interrupção, o programa retorna ao fluxo normal de execução. Isso é feito com a instrução goto fetch, que retorna ao ciclo de busca e execução de instruções normalmente.

Exceção na Mic-1

; Endereços de memória simbólicos para representar o tratamento da interrupção ou exceção.

END_INT .data 0x1234

END_EXC .data 0x5678

; Exceção

exception1 | if (EXC) goto handle_exception | Verifica se há exceção

exception2 | goto fetch | Caso contrário, continua o fetch normalmente

handle_exception: ; Rotina para tratar a exceção

exception3 | SP = SP - 1 | Decrementa o ponteiro da pilha

exception4 | M[SP] = PC | Salva o endereço atual do PC na pilha

exception5 | PC = END_EXC | Pula para o endereço de tratamento de exceção

exception6 | goto fetch | Retorna ao fetch após o tratamento

- **Tratamento da Exceção:**

- Quando uma exceção é detectada (representada por EXC), o programa segue um processo semelhante ao tratamento da interrupção. Ele salva o endereço atual do PC na pilha e pula para o endereço de tratamento da exceção (END_EXC). Isso é feito nas instruções exception3, exception4 e exception5.
- Assim como no caso da interrupção, o SO é responsável por gerenciar as exceções e garantir que o estado atual do programa seja preservado ao salvar o endereço do PC na pilha e redirecionar o fluxo de execução para o tratamento apropriado.
- Após o tratamento da exceção, o programa retorna ao fluxo normal de execução. Isso é feito com a instrução goto fetch, que retorna ao ciclo de busca e execução de instruções normalmente.