

Operação da Unidade de Controle

Arquitetura e Organização de Computadores

Prof. Lucas de Oliveira Teixeira

UEM

Introdução

Introdução

- A unidade de controle de um processador efetua duas tarefas:
 - Faz com que o processador execute uma série de micro-operações na sequência correta. Com base no programa que está sendo executado.
 - Gera os sinais de controle que fazem com que cada micro-operação seja executada.
- Os sinais de controle gerados pela unidade de controle causam a abertura e o fechamento de portas lógicas, resultando em transferência de dados e/ou operações da ULA.

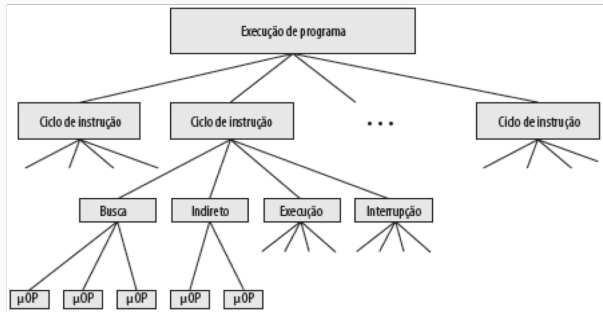
Micro-operações

Micro-operações

- A execução de uma instrução envolve a execução de uma sequência de subpassos.
- Por exemplo, uma execução geralmente consiste de ciclos de busca (1), indireto (2), execução (3) e interrupção (4).
- Cada ciclo é, por sua vez, feito de uma sequência de operações mais básicas chamadas micro-operações.
- Uma micro-operação geralmente envolve:
 - Uma transferência entre registradores;
 - Uma transferência entre um registrador e um barramento externo; ou
 - Uma simples operação da ULA.

- Um computador executa um programa.
- Por exemplo, cada instrução é executada por meio do ciclo de busca, decodificação e execução (com possibilidade de pipelining).
- Cada ciclo tem um número de passos.
- Estes passos são chamados de micro-operações. Cada passo faz muito pouco, é uma operação atômica da CPU.

Micro-operações



Tipos:

- Transferência de dados entre registradores.
- Transferência de dados de registrador para interface externa.
- Transferência de dados da interface externa para registrador.
- Realizar operações aritméticas ou lógicas.

Ciclo de busca:

- Busca a próxima instrução a ser executada na memória.
- Ocorre no início de cada ciclo de instrução.

Registradores envolvidos no ciclo de busca:

- Registrador de endereço de memória (MAR): Conectado ao barramento de endereço. Especifica um endereço para ler ou escrever um operando.
- Registrador de buffer de memória (MBR): Conectado ao barramento de dados. Mantém dados para escrever ou últimos dados lidos.
- Contador de programa (PC): Mantém endereço da próxima instrução a ser lida.
- Registrador de instrução (IR): Mantém última instrução lida.

Micro-operações

Micro-operações do ciclo de busca:

- Endereço da próxima instrução está no PC.
- Conteúdo de PC é colocado no MAR.
- Endereço (MAR) é colocado no barramento de endereço.
- Unidade de controle emite comando de READ.
- Resultado (instrução) aparece no barramento de dados.
- Os dados (instrução) são copiados para o MBR.
- PC é incrementado (em paralelo com comando acima).
- Os dados (instrução) são movidos do MBR para IR.
- MBR agora está livre para mais buscas de dados.

Micro-operações do ciclo de busca:

- | | |
|---|---------------------|
| 1 | t1: MAR = (PC) |
| 2 | t2: MBR = (memoria) |
| 3 | PC = (PC) +1 |
| 4 | t3: IR = (MBR) |

Regras para o agrupamento:

- Sequência apropriada deve ser seguida:
 - MAR = (PC) deve preceder MBR = (memória).
- Conflitos não podem acontecer:
 - MBR = (memória) deve preceder IR = (MBR).
- O incremento de PC envolve adição, logo pode demorar mais do que um ciclo de clock.

Micro-operações do ciclo indireto:

- | | |
|---|---|
| 1 | t4: MAR = (IR — endereço) — Campo de endereço do IR |
| 2 | t5: MBR = (memoria) |
| 3 | t6: IR — endereço = (MBR — endereço) |

- No ciclo indireto, o campo endereço da instrução (IR - endereço) não contém o endereço do operando, Contém o endereço do endereço do operando.
- Após estas micro-operações IR agora está no mesmo estado como se o endereçamento direto fosse usado.

Ciclo de execução:

- Obviamente, cada instrução possui um conjunto de micro-operações diferentes.
- Por exemplo, ADD R1, X
- Soma o conteúdo do local X de memória ao registrador R1, guardando resultado em R1 novamente.

1	t7: MAR = (IR — endereco)
2	t8: MBR = (memoria)
3	t9: R1 = R1 + (MBR)

Ciclo de execução:

- Por exemplo, ISZ X
- Incrementa X e pula a próxima instrução se X for zero.

1	t7: $MAR = (IR - \text{endereco})$
2	t8: $MBR = (\text{memoria})$
3	t9: $MBR = (MBR) + 1$
4	t10: $\text{memoria} = (MBR)$
5	t11: Verifica a flag de zero, se for zero $PC = (PC) + 1$

Ciclo de execução:

- Por exemplo, CALL X
- Salva o endereço de retorno e desvia o fluxo de execução para X.

1	t7: MBR = (PC)
2	RSP = RSP - 8 (considerando 64 bits)
3	t8: MAR = Stack Pointer (RSP) (topo da pilha)
4	PC = X
5	t9: memória = (MBR).

Ciclo de interrupção:

- Ao completar o ciclo de execução, um teste é feito para determinar se há interrupção habilitada.
- Se sim, ocorre o ciclo de interrupção.
- Interrupção habilitada é significa um sinal em algum barramento na unidade de controle.

Ciclo de interrupção:

- 1 t10: MBR = (PC)
- 2 RSP = RSP - 8 (considerando 64 bits)
- 3 t11: MAR = Stack Pointer (RSP) (topo da pilha)
- 4 PC = Endereco da rotina de interrupcao
- 5 t12: memoria = (MBR).

- Importante: O salvamento de contexto (valor atual dos registradores) é feito pela rotina de tratamento de interrupção.

União dos ciclos:

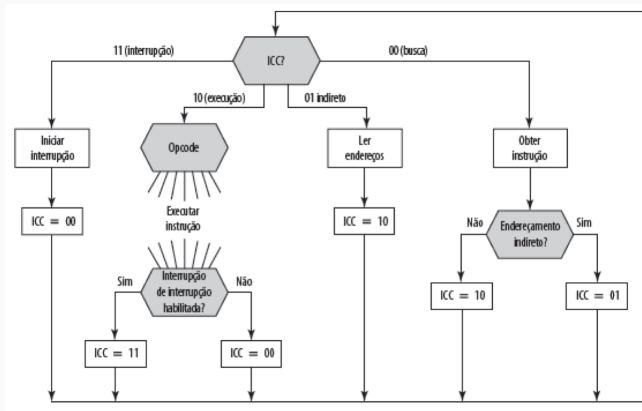
- Cada fase é decomposta em uma sequência de micro-operações elementares.
- Exemplo: Ciclos de busca, indireto e de interrupção.
- Ciclo de execução: Uma sequência de micro-operações para cada opcode.
- Na prática, precisamos unir essa sequência.

União dos ciclos:

- Para a união, existe um outro registrador chamado de Código de Ciclo de Instrução (CCI) designa em qual parte do ciclo o processador se encontra:
 - 00: Busca.
 - 01: Indireto.
 - 10: Execução.
 - 11: Interrupção.
- Isso considerando o nosso exemplo com quatro ciclos.

Micro-operações

Fluxograma do ciclo de instrução



Unidade de Controle

- A unidade de controle é responsável por fazer o sequenciamento das micro-operações, isso é determinar a ordem correta de execução.
- Além disso, é responsável por executar cada micro-operação por meio de sinais de controle para todos os componentes internos e externos do processador.

Sinais de entrada:

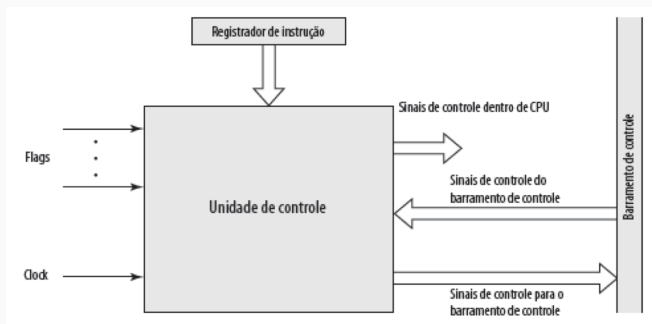
- Clock:
 - Comanda a operação da unidade de controle.
 - Uma micro-operação (ou conjunto de micro-operações paralelas) por ciclo de clock.
- Registrador de instrução:
 - Código de operação da instrução atual.
 - Determina quais micro-operações serão realizadas.
- Flags:
 - Estado da CPU.
 - Resultados das operações anteriores.
- Sinais do barramento de controle:
 - Interrupções.
 - Confirmações.

Sinais de saída:

- CPU:
 - Causam movimentação de dados de um registrador para outro.
 - Ativam funções específicas da ULA.
- Barramento de controle:
 - Para memória.
 - Para módulos de E/S.

Unidade de Controle

Sinais:



Unidade de Controle

Exemplos de sinais de controle:

1 MAR = (PC):

- Unidade de controle ativa sinal para abrir portas entre PC e MAR.

1 MBR = (memória):

- Abrir portas entre MAR e barramento de endereço.
- Sinal de controle de leitura de memória (READ).
- Abrir portas entre barramento de dados e MBR.

Implementação:

- Hardware: A unidade de controle é basicamente um circuito que implementa uma máquina de estados.
- Microprogramada: A unidade de controle é um circuito de uso geral que possui instruções que são usadas para criar o programa de controle.

Unidade de Controle por Hardware

Unidade de Controle por Hardware

- A unidade de controle implementada em hardware é basicamente um circuito digital com entradas e saídas.
- Os bits dos sinais no barramento de controle e as flags possuem um significado único, assim cada um deve ser considerado individual.
- Agora, para o opcode é necessário um circuito exclusivo, então o circuito tende a ficar grande.

Problemas:

- A sequenciação das micro-operações é complexa de ser realizada em hardware.
- O circuito é inflexível, é complexo incluir novas instruções.

Unidade de Controle Microprogramado

- Vamos ver na próxima aula!