UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Arquitetura de Comptuadores II

Arquitetura Básica de um Processador

Marcelo Zamith
zamith.marcelo@gmail.com

Conceitos Básicos

- Um processador é organizado em duas unidades:
  - Seção de Processamento
  - Seção de Controle

Conceitos Básicos

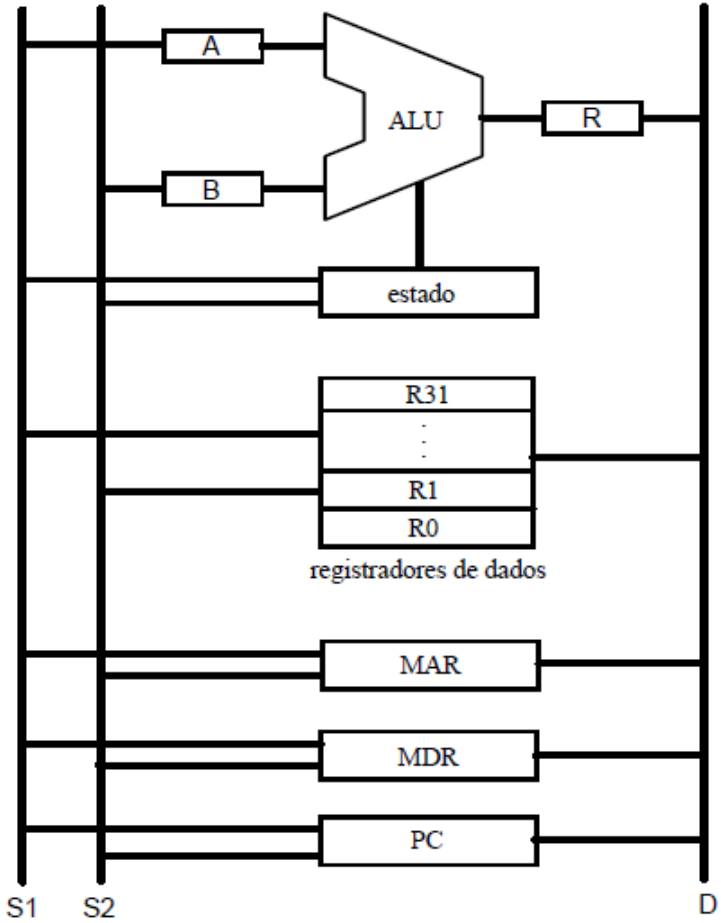
- Um processador é organizado em duas unidades:
  - Seção de Processamento
  - Seção de Controle

Seção de Processamento

- A seção de processamento é formada principalmente:
  - Unidade Lógica e Aritmética (ALU)
  - Registradores

Seção de Processamento

- Componentes da seção de processamento



Seção de Processamento

- ALU realiza as operações
  - Aritméticas
  - Lógicas
- Registradores são usados para armazenar informações internamente no processador
  - Leitura
  - Escrita

Seção de Processamento

- Os diversos registradores possuem um uso bem definido na arquitetura
- De uma maneira em geral podem ser classificados em três tipos:
  - Registradores de uso geral
  - Registradores de uso específico
  - Registradores auxiliares

Registradores de Uso Geral

- Os registradores de uso geral normalmente são usados para armazenar dados que serão processados ou produzidos pela ALU

- Coletivamente, estes registradores são chamados de **conjunto de registradores de dados** (*data register file*)
  - Na figura são os registradores R0, ... , R31

Registradores de Uso Específico

- O **registrador de estado** (*status register*) associado à ALU é um registrador de uso específico e contém informações sobre o resultado produzido pela ALU

- Possui bits sinalizadores que são ativados ou desativados dependendo do resultado produzido pela ALU. Por exemplo:
  - Resultado nulo
  - Resultado negativo

Registradores de Uso Específico

- O **contador de programa** (*program counter*) contém o endereço da locação de memória onde se encontra a próxima instrução a ser executada pelo processador

Registradores Auxiliares

- Os registradores auxiliares normalmente são usados para armazenamento temporário

- Antes de cada operação da ALU, os operandos são transferidos dos registradores de dados ou da memória principal para registradores auxiliares

Registradores Auxiliares

- O resultado da ALU é temporariamente armazenado em um registrador auxiliar até ser transferido para o seu destino
  - Registrador de dados
  - Memória principal

Registradores Auxiliares

- O registrador MAR (*Memory Address Register*) armazena o endereço da locação de memória onde será feito o acesso

- O registrador MDR (*Memory Data Register*) armazena temporariamente a informação transferida de ou para a locação de memória endereçada por MAR

Registradores Auxiliares

- Em geral, registradores auxiliares não são visíveis, no sentido que um programador de baixo nível não dispõe de instruções para acessá-los diretamente

Barramentos Internos

- A interligação entre os registradores e a ALU é feita através de três vias, denominadas de **barramentos internos**
- Os barramentos internos S1 e S2 permitem a transferência de dados dos registradores para a ALU
- O barramento D permite a transferência do resultado da ALU, armazenado no registrador auxiliar para outro registrador

Arquitetura de Processador

- Em uma arquitetura de *n* bits, todas as operações da ALU podem ser realizadas sobre operandos de até *n* bits.

- Normalmente, os registradores de dados e os barramentos internos também são de *n* bits, de forma a permitir que os dados sejam armazenados e transferidos de forma eficiente

Execução de Instruções

- A execução de uma instrução envolve a realização de uma sequência de passos, que pode ser denominada de **passos de execução**.

Execução de Instruções

- No passo de **busca**, o processador realiza o acesso ao código binário da instrução, que está armazenado na memória principal

- No passo de **decodificação** da instrução, as informações contidas no código da instrução são interpretadas

Execução de Instruções

- O terceiro passo é a **execução**, onde a operação indicada pela instrução é efetuada. Por exemplo, uma operação na ALU

- O último passo é chamado de **resultado**. Neste passo, o resultado produzido pela instrução é armazenado em um registrador ou na memória principal

Execução de Instruções

- Cada passo de execução envolve a realização de várias operações básicas

- As operações básicas acontecem dentro da seção de processamento, sob a coordenação da seção de controle

Execução de Instruções

- Existem quatro tipos principais de operações básicas
  - Transferência de dados entre os registradores e a ALU
  - Transferência de dados entre os registradores
  - Transferência de dados entre os registradores e a memória
  - Operações aritméticas e lógicas realizadas pela ALU

Execução de Instruções

- Cada passo de execução não possui necessariamente o mesmo número de operações básicas em todas as instruções

Exemplos de Instruções

Tipo	Exemplo	Descrição
lógica/aritmética	`ADD R1,R2,Rd`	O conteúdo dos registradores R1 e R2 devem ser somados e o resultado armazenado em Rd
desvio incondiconal	`JMP dst`	A próxima instrução a ser executada deve ser a que se encontra na locação de memória com endereço dst
desvio condicional	`JZ dst`	A próxima instrução a ser executada deve ser a que se encontra no endereço dst, caso o *bit Z* do registrador de estado esteja ativado
acesso à memória	`LOAD end,R1` `STORE end,R1`	O dado armazenado na locação de memória com endereço end deve ser transferido para o registrador R1 (LOAD)

Execução de Instruções

	Aritméticas e Lógicas	**Desvios Incondicionais**	**Desvios Condicionais**	**Acessos à Memória**
Busca	MAR ← PC MDR ← M[MAR] IR ← MDR PC++	MAR ← PC MDR ← M[MAR] IR ← MDR PC++	MAR ← PC MDR ← M[MAR] IR ← MDR PC++	MAR ← PC MDR ← M[MAR] IR ← MDR PC++
Decodificação	decod A ← Rs1 B ← Rs2	decod	decod	decod
Execução	R ← A op B	PC ← destino	cond se (cond) PC ← destino	MAR ← end MDR ← Rs (E) M[MAR] ← MDR (E) MDR ← M[MAR] (L)
Resultado	Rd ← R			Rd ← MDR (L)

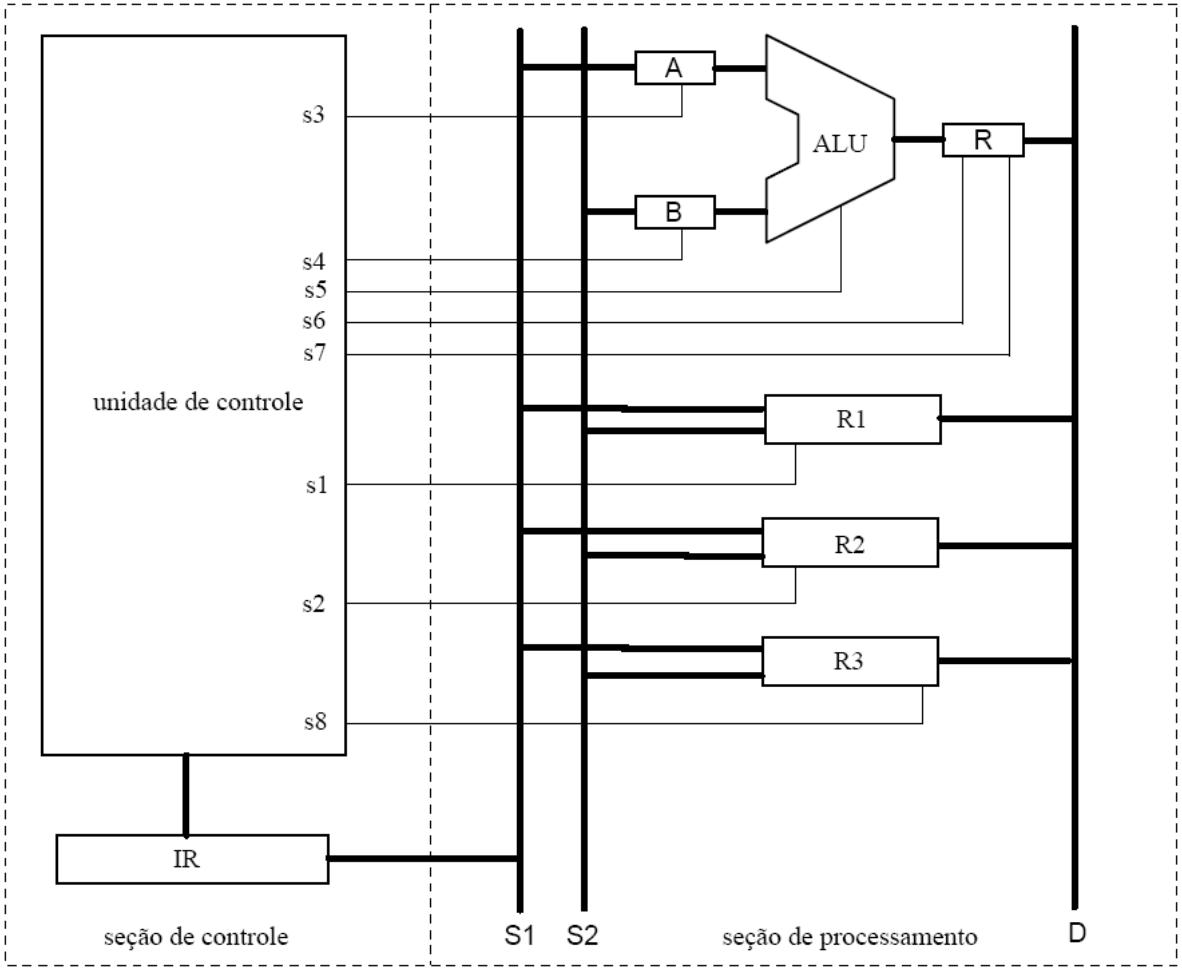
Seção de Controle

- As operações básicas que ocorrem dentro da seção de processamento são todas comandadas pela seção de controle

- A seção de controle é formada basicamente pela unidade de controle e pelo registrador de instrução ou IR (*Instruction Register*)

Seção de Controle

- A seção de controle e a parte de processamento



Seção de Controle

- Ao efetuar a busca da instrução, a **unidade de controle** interpreta a instrução
  - Identifica quais as operações básicas que devem ser realizadas
  - Ativa os sinais de controle que fazem a operação básica de fato acontecer

- Os sinais de controle que são ativados, bem como a sequência com que são ativados, depende de cada instrução em particular

Seção de Controle

- Suponha a execução da instrução
  add R1, R2, R3
- Essa e todas as demais instruções são coordenadas pela unidade de controle
- A seção de processamento foi representada apenas com os três registradores de dados envolvidos na execução desta instrução

Seção de Controle

- A execução da instrução add R1, R2, R3 requer quatro operações básicas:

  (1) Transferência do conteúdo do registrador de dados R1 para o registrador temporário A

  (2) Transferência do conteúdo do registrador de dados R2 para o registrador temporário B

Seção de Controle

(3) Adição dos dados armazenados nos registradores A e B e armazenamento do resultado no registrador R

(4) Transferência do conteúdo do registrador R para o registrador R3

Sinais de Controle

- A sequência de ativação dos sinais de controle com as operações básicas apresentadas anteriormente:

operação básica	sinal de controle	descrição da operação básica
(1) (2)	s1, s2	coloca o conteúdo de R1 e R2 nos barramentos S1 e S2, respectivamente.
	s3,s4	armazena a informação presente nos barramentos S1e S2 em A e B, respectivamente.
(3)	s5	seleciona a operação de soma na ALU.
	s6	armazena o resultado produzido pela ALU em R.
(4)	s7	coloca o conteúdo de R no barramento D.
	s8	armazena a informação presente no barramento D em R3.

Sinais de Controle

- Para cada registrador existem sinais que controlam a leitura e a escrita do registrador

- Para a ALU existem sinais que indicam a operação aritmética ou lógica que deve ser realizada

Sinais de Controle

- Para qualquer componente na seção de processamento existem os sinais de controle necessários

- Para a execução de uma operação básica, a unidade de controle simplesmente ativa os sinais apropriados na sequência correta

Sinal de *Clock*

- A ordem na qual os sinais de controle são ativados é crítica
- Alguns sinais devem obrigatoriamente preceder outros (Exemplo: s1 antes de s3)
- Outros sinais podem ser ativados simultaneamente (Exemplo: s1 e s2)

Sinal de *Clock*

- Em alguns casos deve ser observado um intervalo de tempo mínimo entre a ativação de dois sinais para que seja garantido um tempo suficiente para a transmissão da informação através dos barramentos internos

Sinal de *Clock*

- Com o objetivo de atender as relações de tempo requeridas na ativação dos sinais de controle, a unidade de controle opera em sincronismo com um **sinal de *clock***

- A execução de uma instrução consome um determinado número de ciclos de *clock*

Sinal de *Clock*

- O número de ciclos de *clock* por instrução não é o mesmo para todas as instruções, visto que cada instrução pode envolver um número diferente de operações básicas em cada passo de execução

Sinal de *Clock*

- O tamanho do ciclo de *clock* é um dos fatores que determinam diretamente o desempenho de um processador

- Quanto menor o tamanho do ciclo de *clock*, menor será o tempo de execução das instruções, e assim maior será o número de instruções executadas por unidade de tempo

Sinal de *Clock*

- Nas décadas de 70 e 80, procurava-se diminuir o tamanho do ciclo de *clock* com novas tecnologias que permitissem velocidades de operação cada vez maiores

- A aproximação das tecnologias utilizadas com os limites impostos pela própria física, fez com que esta evolução se tornasse mais lenta e os custos mais elevados

Sinal de *Clock*

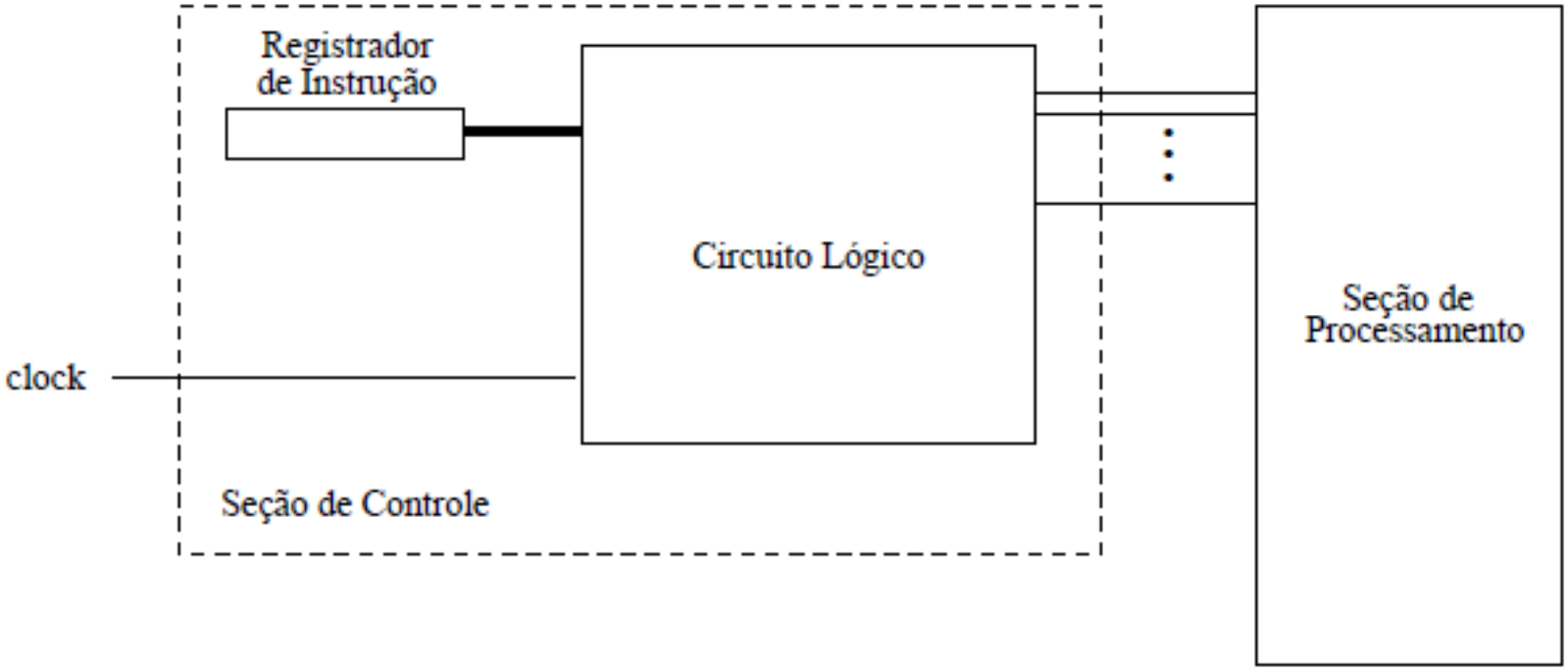
- Assim, a redução do ciclo de *clock* passou a ser considerada sob o ponto de vista arquitetural
  - Através de simplificações na arquitetura, de modo que ela possa ser implementada através de circuitos mais simples e naturalmente mais rápidos

Implementação da Unidade de Controle

- Existem basicamente duas maneiras de implementar uma unidade de controle
  - **Lógica Aleatória** (*hardwared control*)
  - **Microprogramação**

Lógica Aleatória

- Organização de uma unidade de controle implementada com lógica aleatória



Lógica Aleatória

- A unidade de controle é formada por um único circuito lógico
  - A entrada é o código de instrução armazenado no registrador de instrução
  - As saídas são os próprios sinais de controle que comandam as operações básicas na seção de processamento

Lógica Aleatória

- A desvantagem desta implementação está na complexidade do circuito de controle

- O número de dispositivos lógicos aumenta rapidamente com o número de instruções oferecidas pela arquitetura e com o número de operações básicas que devem ser realizadas na execução de uma instrução

Lógica Aleatória

- Em arquiteturas com instruções funcionalmente complexas, o projeto de uma unidade de controle com lógica aleatória torna-se muito difícil e mais propenso a erros

Microprogramação

- Na técnica de microprogramação, cada instrução oferecida pela arquitetura, denominada de **macroinstrução**,é executada por uma sequência de instruções primitivas, extremamente simples, denominadas **microinstruções**

Microprogramação

- Os próprios bits das microinstruções são usados para ativar e desativar os sinais de controle que comandam as operações básicas

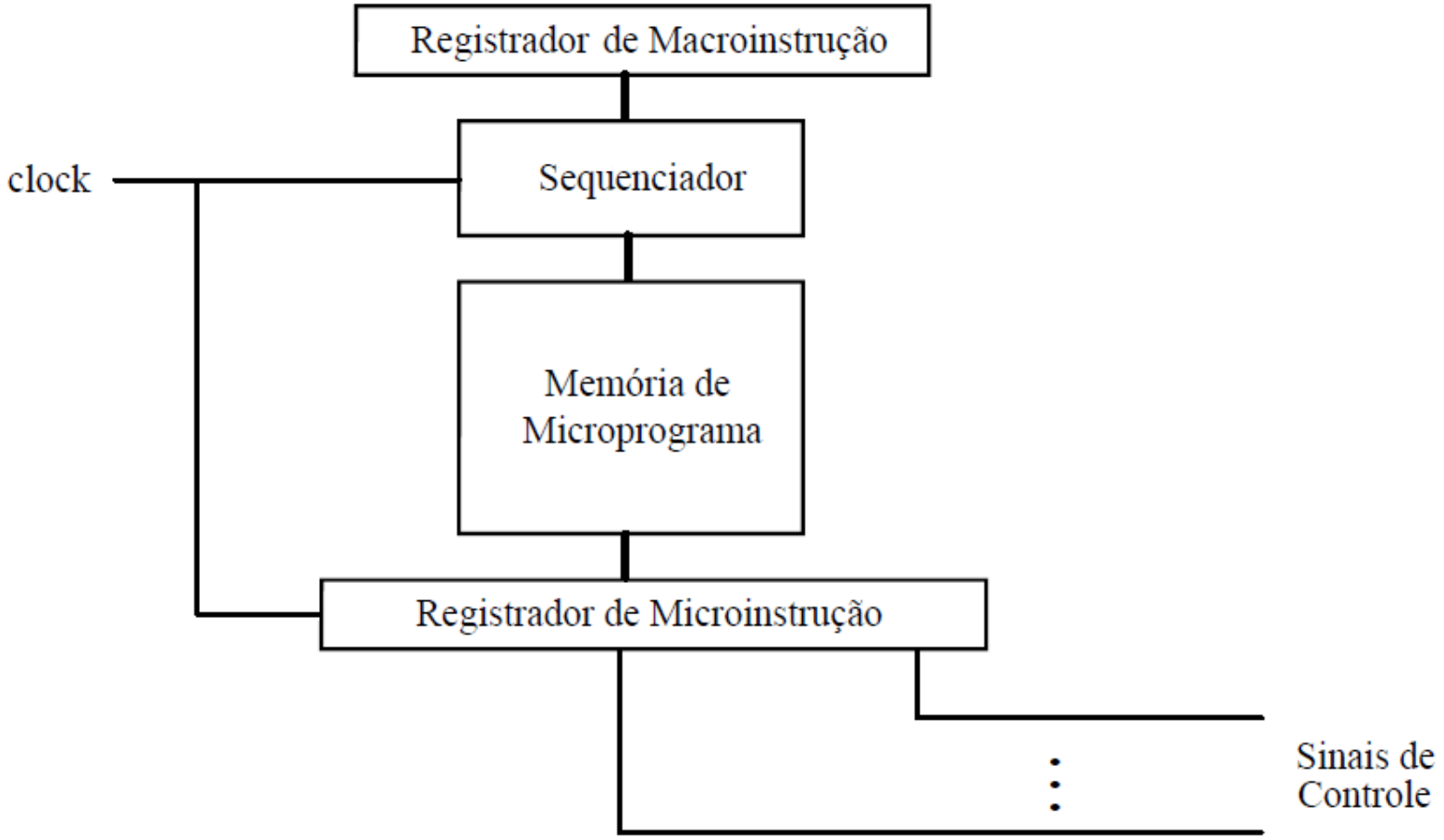
- A sequência de microinstruções que executa uma macroinstrução forma uma **microrotina**

Microprogramação

- As microinstruções da microrotina executam as operações básicas associadas à macroinstrução

Microprogramação

- Unidade de controle microprogramada



Microprogramação

- As microinstruções são armazenadas na **memória de microprograma**
- O código da macroinstrução é armazenado no registrador de (macro)instrução
- O **sequenciador** interpreta este código e determina o endereço de entrada da microrotina que executa aquela macroinstrução

Microprogramação

- A cada ciclo de *clock*, o sequenciador fornece o endereço da próxima microinstrução a ser executada

- Após o acesso à microinstrução, ela é armazenada no **registrador de microinstrução**

Microprogramação

- Alguns bits da microinstrução são usados diretamente para comandar os sinais de controle para a seção de processamento

- Outros bits são utilizados pelo sequenciador para determinar a próxima microinstrução a ser executada

Microprogramação

- A execução de uma microinstrução envolve
  - Acesso da microinstrução na memória de microprograma
  - Armazenamento no registrador de microinstrução
  - Realização das operações básicas comandadas pelos bits da microinstrução
- Uma nova microinstrução é executada a cada ciclo de *clock*

Microprogramação

- Quando a execução de uma microrotina é concluída
  - Uma nova macroinstrução é acessada na memória principal e armazenada no registrador de instrução
  - Iniciada a execução de uma nova microrotina

Microprogramação

- Vantagem da microprogramação
  - A implementação das instruções reduz-se basicamente à escrita das microrotinas que serão gravadas na memória de microprograma
- Esta vantagem é mais significativa quando a arquitetura fornece um grande número de instruções e estas instruções são complexas
  - Circuito lógico complicado substituído pela escrita das microrotinas

Exceções e Interrupções

- Uma das partes mais difíceis do controle é a implementação das exceções e das interrupções
  - Mudam o fluxo normal da execução das instruções

Exceções e Interrupções

- Uma exceção é um evento inesperado gerado dentro do próprio processador
  - Exemplo: *overflow* aritmético

- Uma interrupção também é um evento inesperado só que gerado externamente ao processador
  - Exemplo: interrupção de um dispositivo de E/S

Exceções e Interrupções

- A detecção das condições excepcionais faz parte do caminho crítico de uma máquina

- Elas estão relacionadas a temporização, influenciam na determinação do período de *clock* e consequentemente no desempenho da máquina

Exceções e Interrupções

- As ações básicas que a máquina precisa realizar são:
  - Salvar o endereço da instrução envolvida no evento
  - Transferir o controle para o sistema operacional num determinado endereço
- O sistema operacional pode então realizar as ações apropriadas
  - Conhecendo a instrução e a razão da exceção

Exceções e Interrupções

- Existem dois métodos conhecidos para comunicar ao sistema operacional a razão de uma exceção:
  - Um registrador de causa com um campo adicional para indicar o motivo da exceção
  - Um vetor de interrupções em que o endereço para o qual o controle é transferido é determinado pela causa da exceção

Perguntas ?

Marcelo Zamith
zamith.marcelo@gmail.com