

CAPÍTULO 9

ELEMENTOS DE INSTRUÇÕES DE MÁQUINAS

Elementos de instruções:

- Código de Operação (OPCODE): especifica a operação a ser efetuada, através de um código binário
- Referência a Operando Fonte: uma operação pode envolver um ou mais operandos fontes, que são os dados de entrada (inputs) da operação*
- Referência a Operando de Destino: uma operação pode produzir um resultado*
- Endereço da próxima instrução: indica onde a CPU deve buscar a próxima instrução. Pode estar localizada na memória principal ou na secundária (disco).

*Podem estar localizados na memória principal ou virtual, registrador da CPU, dispositivo de E/S

Representação de Instruções:

- Mnemônicos:
 - ADD: Adição
 - SUB: Subtração
 - MPY: Multiplicação
 - DIV: Divisão
 - LOAD: Carregar dados da memória
 - STOR: Armazenas dados na memória

Tipos de Instruções de máquina:

- Processamento de dados: instruções lógicas e aritméticas
- Armazenamento de dados: instruções de memória
- Movimentação de dados: instruções de e/s
- Controle: instruções de teste e desvio

* *Operações aritméticas* tem a capacidade de processamento de dados numéricos. *Instruções*

lógicas (booleanas) operam sobre bits de uma palavra, ou seja, processa qualquer outro tipo de dado. Essas operações são executadas em dados armazenados em registradores, para isso são necessárias *instruções de memória* para mover os dados entre a memória e os registradores. Para passar dados para a memória são necessárias as *instruções de E/S*. As *instruções de teste* são usadas para testar o valor de uma palavra de dados. Dependendo do resultado desse teste, são utilizadas as *instruções de desvio* para desviar a execução do programa a uma nova instrução.

Número de Endereços:

- Idealmente o número de endereços que uma instrução possui seriam 4: 2 para operandos, uma para o resultado e outra para o endereço da próxima instrução
- Poucos endereços por instrução requerem menor complexidade da CPU e tem instruções menores. Por outro lado, resulta em maior tempo de execução e em programas mais complexos

Projeto do Conjunto de Instruções

- Questões relativas ao projeto de conjunto de instruções
 - Repertório de operações: quantas e quais são as operações que devem ser fornecidas e quão complexas elas devem ser
 - Tipos de dados: quais tipos de dados as operações são efetuadas
 - Formatos de instrução: qual o tamanho das instruções, número de endereços por instrução, tamanho dos campos...
 - Registradores: qual o número de registradores da CPU que podem ser utilizados pelas instruções e qual o propósito de cada um
 - Endereçamento: qual a forma que o endereço de um operando pode ser especificado

Tipos de Operandos:

- Endereços
- Números: inteiro, ponto flutuante, decimal
- Caracteres
- Dados Lógicos

Tipos de Operações

- Operações de transferência de dados:

- Deve especificar o endereço dos operandos fonte e de destino, o comprimento de dados a transferir, o modo de endereçamento de cada operando

- São o tipo mais simples de operação, em termos de ação tomada pela CPU.

- Se os dois operandos são registradores, a CPU transfere dados entre registradores, é uma operação interna da CPU.

- Se um ou ambos os operandos estão na memória deve realizar algumas ou todas as seguintes ações: calcular endereço de memória (com base no modo de endereçamento especificado), se o endereço for da memória virtual, traduzir para um endereço de memória real, determinar se o item está na memória cache, se não estiver, emitir um comando para o módulo de memória.

- Operações aritméticas:

- A maioria das máquinas fornece operações aritméticas básicas para soma, subtração, multiplicação e divisão.

- A execução de uma instrução aritmética pode envolver transferência de dados, para fornecer os valores dos operandos como entrada para a ULA e para armazenar na memória o valor obtido como saída da ULA.

- Operações lógicas:

- Variedade de operações para manipular bits de uma palavra ou de uma unidade endereçável, e são baseadas em operações booleanas: NOT, AND, OR, XOR, EQUAL

- Operações de conversão:

- São aquelas que mudam ou operam sobre o formato de dados. Ex: conversão de número decimal para binário

- Operações de E/S:

- Existe uma variedade de abordagens: E/S programada, E/S mapeada na memória, DMA e uso de processadores de E/S

- Operações de controle de sistema:

- São executadas apenas quando o processador está no estado privilegiado ou está executando um programa carregado em área especial da memória, que é privilegiada.

- Geralmente são reservadas para uso do sistema operacional

- Operações de transferência de controle:

- Quando altera a sequência de execução de instruções, a CPU atualiza o contador de programa com o endereço de alguma outra instrução armazenada na memória.

- Essas operações são requeridas pelas razões: poder executar um conjunto de instruções mais de uma vez; tomada de decisões, executar uma sequência de operações dada determinada condição; dividir o programa em partes.

- Operações de desvio: tem como um de seus operandos o endereço da próxima instrução. Pode ser um desvio condicional.

- Operações de salto: incluem um endereço de desvio implícito. Indica que a execução de uma instrução da sequência de instruções deve ser omitida.

- Operações de chamada de procedimento: é um subprograma autocontido, que é incorporado em um programa maior. Ele pode ser chamado ou invocado em qualquer ponto do programa. Uma chamada a um procedimento instrui o processador a executar todo o procedimento e depois retornar ao ponto em que ocorreu a chamada. Envolve duas instruções básicas: uma instrução de chamada, que desvia a execução da instrução corrente para o início do procedimento, e uma instrução de retorno, que provoca o retorno da execução do procedimento para o endereço em que ocorreu a chamada. Ambas são formas de instruções de desvio

CAPÍTULO 10

- Modo de Endereçamento: quase todos os computadores possuem mais de um modo de endereçamento. A questão é como a unidade de controle escolherá o modo a ser usado em uma instrução. Pode ser decidido por aproximação através do OPCODE ou então um ou mais bits podem ser usados no formato de instruções.

- Técnicas de Endereçamento:

* A: conteúdo do campo de endereço da instrução

* EA: endereço real (efetivo) da posição que contém o operando

* X: conteúdo da posição de endereço x

- Imediato:

- Operando = campo de endereço

- Ex: ADD 5 (5 é o operando)

- É a forma mais simples de endereçamento e pode ser utilizado para definir e utilizar constantes e inicializar variáveis. Rápido.

- Vantagem: não necessita de qualquer acesso à memória para obter o operando, economizando assim um ciclo de acesso à memória

- Desvantagem: o tamanho do operando fica limitado ao tamanho do campo de endereço, o qual na maioria das vezes é menor que o tamanho de uma palavra

- Direto:

- O campo de endereçamento contém o endereço efetivo do operando $EA = A$

- Técnica muito utilizada nas primeiras gerações de computadores

- Ex: ADD A (A está na memória). Adiciona o conteúdo da célula A ao acumulador. Procura no endereço de memória A pelo operando.

- Vantagem: implica apenas um acesso à memória

- Desvantagem: limita o espaço do endereçamento

- Indireto:

- Procura solucionar o problema do endereçamento direto, colocando no campo de endereço da instrução uma referência a memória que contém o endereço completo da operação. Ex: $EA = (A)$

- Vantagem: para um campo da palavra n está disponível um espaço de endereçamento 2^n

- Desvantagem: são necessário dois acessos para buscar o operando

- Registrador:

- Semelhante ao endereçamento direto, só que ao invés de o campo referenciar uma posição na memória, referencia um registrador.

- Vantagem: apenas um pequeno campo de endereço é necessário e não é preciso referências à memória. Execução muito rápida

- Desvantagem: número limitado de registradores

- Indireto por Registrador:

- Semelhante ao endereçamento indireto, só que ao invés da referência do campo de memória ser de uma posição na memória é um registrador. Ex: $EA = R$

- Vantagens e desvantagens são as mesmas do endereçamento indireto

- Vantagem: usa uma referência à memória a menos que o endereçamento indireto

- Deslocamento (Indexado):

- Combina as técnicas de endereçamento direto e endereçamento indireto por registrador. Ex: $EA = A + (R)$

- Obriga que a instrução tenha dois campos de endereço. Pelo menos um deve ser explícito (valor A usado diretamente). O outro campo referencia implicitamente um registrador cujo conteúdo é adicionado a A.

- 1) Endereçamento Relativo: utiliza o Program Counter. O campo de endereço da instrução atual é um deslocamento em relação ao PC. Explora o princípio da localidade permitindo poupar bits de endereço na instrução

- 2) Endereçamento via Registrador Base: o registrador de referência contém a posição da memória e o endereço da instrução contém o deslocamento a partir de uma posição de memória.

- 3) Endereçamento Indexado: o campo de endereço referencia uma memória principal e o registrador de referência contém um deslocamento positivo a partir dessa posição de memória. É contrária ao endereçamento baseado em registrador e tem utilidade na execução de instruções iterativas através de alterações sucessivas do registrador de referência. Ex: $EA = A + (R)$
-> $R = R + 1$.

- Pilha:

- A pilha é LIFO e existe um apontador associado Pa pilha que tem o endereço do topo da pilha. O operando está (implicitamente) no topo da pilha.

- Ex: ADD - remove dois elementos da pilha, adiciona-os e coloca o resultado na pilha.

- Formato das Instruções: define a forma como os campos são distribuídos. Cada instrução deve ter um código de operação (opcode), e um ou mais operandos. Cada operando explícito tem de ser referenciado utilizando um dos métodos de endereçamento.

- Tamanho das Instruções: é o fator mais básico do desenho do conjunto de instruções e é afetada por: tamanho da memória, organização da memória, estrutura do bus, velocidade da CPU. O compromisso mais óbvio é entre ter um repertório de instruções mais poderoso e a necessidade de poupar espaço. Além disso, existem outras considerações, como: o tamanho da instrução deve ser igual ou múltiplo do bus do sistema, senão as instruções não ficariam completas; o tamanho da instrução deve ser múltiplo do tamanho de caracteres e dos números da vírgula fixa para não ter

desperdício de bits nos cálculos a efetuar.

- Alocação de Bits: Fatores que determinam a utilização dos bits de endereçamento:

- nº de modos de endereçamento: os modos de endereçamento indicados explicitamente ocupam mais bits que os indicados implicitamente

- nº de operandos: menos endereços tornam os programas mais longos e complexos.

- registrador vs memória: quanto mais registradores puderem ser utilizados para referenciar operandos, menos bits serão necessários

- nº de conjuntos de registradores: para um número fixo de conjuntos de registradores, um menor número de bits será utilizado

- alcance do endereçamento: está relacionado ao número de bits para endereçamento na instrução

CAPÍTULO 11

- Arquitetura de Microprocessadores: A CPU é quem vai exercer o controle entre os registradores da memória e calcular as operações tendo em conta vários sinais de controle. Os computadores contemporâneos são baseados em 3 conceitos:

- Os dados e as instruções são armazenados em uma memória de leitura/escrita

- O conteúdo dessa memória é endereçado por localização, sem preocupação com o tipo de dados

- A execução ocorre de uma forma sequencial (a não ser que explicitamente modificada) de uma instrução para outra

- Organização Interna da CPU:

- Funções básicas:

- Buscar instruções: a CPU tem de ler as instruções da memória

- Interpretar instruções: as instruções tem de ser decodificadas para determinar a ação a executar

- Buscar dados: a execução de uma instrução pode precisar da leitura de dados da memória ou do módulo de E/S

- Processar dados: a execução de uma instrução pode precisar de operações lógicas ou aritméticas nos dados

- Escrever dados: o resultado de uma instrução pode precisar escrever dados na memória ou em um módulo de E/S

- O computador precisar ter um lugar onde guardar os dados

- Precisa saber qual o endereço fixo da última instrução para que ele saiba qual a próxima a executar

- Precisa guardar temporariamente os dados durante a execução de uma instrução. Por isso a CPU precisa ter uma pequena memória interna

- A CPU é constituída por uma Unidade Lógica Aritmética (ULA) e uma Unidade de Controle, além dos registradores internos no processador (instruções e dados são alocados temporariamente)

- A ULA processa e calcula dados

- A Unidade de Controle controla o fluxo de dados e as instruções enviadas e recebidas da CPU, além de controlar as operações da ULA

- A ULA e todos os registradores da CPU estão interligados por um bus interno

- Estrutura e Funções da CPU: a espaço que a CPU tem para trabalhar é constituído por registradores, que constituem o topo da hierarquia da memória. O número e funções dele varia de processador para processador.

- Organização dos Registradores:

- Organizam-se em dois grupos: registradores visíveis ao usuário e registradores de controle de estado

- Registradores Visíveis ao Usuário: permitem ao programador minimizar as referências à memória principal, e possuem as seguintes categorias:

- Registradores de Uso Geral: podem ser atribuídos a uma variedade de funções pelo programador, normalmente podem conter operandos para qualquer código de operação

- Registradores de Dados: só podem ser utilizados para guardar dados e não podem ser utilizados em operações de cálculo de endereços

- Registradores de Endereço: podem ser de uso geral ou estar direcionado para um modo de endereçamento particular (Ex: Stack pointer)

- Códigos de Condição: também conhecidos como flags, geralmente agrupados em

um ou mais registradores que só são alterados pela CPU, dependendo da última operação lógica ou aritmética. Ex: Conjunto de bits individuais (resultado da última operação foi zero), podem ser lidos implicitamente por programas (salta se zero), não pode ser estabelecido por programas

- Registradores de Controle de Estado: são utilizados pela unidade de controle para controlar a operação da CPU e por programas privilegiados (SO) para controlar a execução de outros programas. Exemplos:

- Program Counter(PC): contém o endereço da instrução a ser buscada
- Instruction Register (IR): contém a última instrução buscada
- Memory Access Register (MAR): contém o endereço de uma posição de memória
- Memory Buffer Register (MBR): contém uma palavra de dados a ser escrita na memória ou a palavra lida mais recentemente

- Fazem parte dos registradores internos da CPU: MAR e MBR.

- Os registradores I/O AR e o I/O BR especificam o módulo de E/S usado para a troca de dados entre módulos de E/S

- Todas as CPUs possuem um registrador ou um conjunto deles normalmente chamado de Program Status Word (PSW) que contém informações de estado:

- sign: contém o sinal da última operação aritmética
- zero: ativo quando o resultado é zero
- carry: ativo se uma comparação lógica for verdadeira
- overflow: indica overflow aritmético
- interrupt enable/disable: liga e desliga interrupções
- supervisor: indica se a CPU está executando em modo privilegiados

- Ciclo de Instrução: inclui os seguintes subciclos:

- Busca: lê a próxima instrução da memória para a CPU
- Execução: interpreta o opcode e executa a respectiva operação
- Interrupção: no caso de haver uma interrupção, a informação é salva no estado em que se encontra
- Fluxo de Dados/Ciclo de Busca: no ciclo de busca, uma instrução é lida da memória. O PC tem o endereço da próxima instrução a ser buscada, esse endereço é movido para o MAR e colocado no

barramento (bus) de endereço. A Unidade de Controle requisita uma leitura à memória e o resultado é colocado no barramento de dados e copiado no MBR, e depois movido para o IR. O PC é incrementado em 1. Quando o ciclo de busca termina, a Unidade de Controle verifica se a instrução que está no IR contém operandos que possuem endereçamento indireto. Se tiver, um ciclo indireto é executado.

- Ciclo Indireto: a busca de um endereço indireto começa um novo subciclo de instrução. A principal linha de atividade nessa situação é a alternância entre atividades de busca de instruções e execução. Quando a Unidade de Controle examina o conteúdo do IR, se for um operando com endereçamento indireto, inicia-se o ciclo indireto. Os n bits mais significativos do MBR que contém o endereço de referência, são transferidos para o MAR. Dessa forma a Unidade de Controle faz uma leitura à memória para obter o endereço desejado do operando para o MBR.

- Ciclo de Interrupção: o valor do PC é transferido para o MBR e escrito na memória. O local reservado em memória para este feito é carregado no MAR a partir da Unidade de Controle. O PC é carregado com o endereço da rotina de interrupção.

CAPÍTULO 14

- Para especificar as funções do sistema, temos que levar em conta: operações (opcodes), modos de endereçamento, registradores, interfaces de E/S, interfaces e estrutura de processamento de interrupções. Esses termos determinam o que a CPU faz.

- Micro Operações: cada ciclo envolve um conjunto de vários passos, cada um envolvendo registradores da CPU. Cada um desses passos são micro-operações.

- Ciclo de Busca: 4 registradores estão envolvidos:

- MAR (Memory Address Register): conectado às linhas de endereço do bus do sistema. Especifica endereçamento na memória para operação de leitura ou escrita

- MBR (Memory Buffer Register): conectado às linhas de dados do bus. Contém dados para escrita na memória ou os últimos dados lidos

- PC (Program Counter): contém o endereço da próxima instrução a ser buscada

- IR (Instruction Register): contém a última instrução buscada

- Sequência do Ciclo de Busca:

- O endereço da próxima instrução a ser executada é colocado no PC

- Mover esse endereço para o MAR

- O endereço é colocado no bus de endereços
- A Unidade de Controle envia um comando READ para o controlador do bus e o resultado aparece no bus de dados
- Copia o dado para o MBR
- Incrementa o PC em uma unidade para obter a próxima instrução
- Mover o conteúdo do MBR para o IR
- Um ciclo de busca tem 3 passos e 4 micro-operações (***cada micro-operação envolve o movimento de dados de ou para um registrador***)

- Simbolicamente:

* A1: $MAR \leftarrow PC$ (mover os dados do PC para o MAR)

* A2: $MBR \leftarrow MEMÓRIA$, $PC \leftarrow PC + 1$ (mover o conteúdo da memória especificada do MAR para o MBR e incrementar o PC)

* A3: $IR \leftarrow MBR$ (mover o conteúdo do MBR para o IR)

- Ciclo Indireto:

- O campo endereço de uma instrução é transferido para o MAR. É então usado para adquirir o endereço do operando e passado para o MBR. Finalmente, o campo endereço do IR é atualizado para o contido no MBR.

- Micro-operações:

* T1: $MAR \leftarrow (IR(ENDEREÇO))$

* T2: $MBR \leftarrow MEMÓRIA$

* T3: $IR(ENDEREÇO) \leftarrow MBR(ENDEREÇO)$

- Ciclo de Interrupção:

- Depois de completado o ciclo de execução é feito um teste para saber se houve alguma interrupção. Em caso afirmativo é dado início ao ciclo de interrupção:

* T1: $MBR \leftarrow (PC)$

* T2: $MAR \leftarrow \text{Endereço de Salvamento}$

$PC \leftarrow \text{Endereço de Rotina}$

* T3: $MEMÓRIA \leftarrow MBR$

- O conteúdo do PC é transferido para o MBR, para que possa ser salvo para o retorno da interrupção

- O MAR é carregado com o endereço onde o conteúdo do PC deve ser salvo e o PC é atualizado com o endereço da rotina de tratamento de interrupção

- Guardar o MBR que contém o velho valor do PC na memória

- A CPU está pronta para começar o próximo ciclo de instrução

- Ciclo de Execução:

ADD

- ADD R1, X - adiciona o conteúdo da posição X ao registrador 1, resultado coloca em R1

* T1: MAR \leftarrow (IR ADDRESS)

* T2: MBR \leftarrow (MEMÓRIA)

* T3: R1 \leftarrow (R1) + (MBR)

ISZ

- ISZ X - incrementa 1 e salta se for zero

* T1: MAR \leftarrow (IR ADDRESS)

* T2: MBR \leftarrow (MEMÓRIA)

* T3: MBR \leftarrow (MBR) + 1

* T4: MEMÓRIA \leftarrow (MBR)

IF (MBR) == 0 THEN PC \leftarrow (PC) + 1

BSA (BRANCH AND SAVE ADDRESS)

- BSA X - endereço de instrução seguinte a BSA é salvo em X e a execução do programa reinicia a partir da instrução X+1

* T1: MAR \leftarrow (IR ADDRESS)

MBR \leftarrow (PC)

* T2: PC \leftarrow (IR ADDRESS)

MEMÓRIA \leftarrow (MBR)

* T3: PC \leftarrow PC + 1

- Requisitos Funcionais: são as funções que a unidade de controle tem que executar.

- A Unidade de Controle é caracterizada por:

- Definir os elementos básicos da CPU

- Descrever as micro-operações às operações que a CPU pode executar

- Determinar as funções que a unidade de controle tem de executar para que as micro-operações sejam executadas

- Elementos Básicos do Processador:

- ULA - componente funcional de um computador

- Registradores - guardar dados internos na CPU

- Path de dados internos - mover dados entre registradores e a ULA e entre registradores

- Path de dados externos - ligam registradores à memória e a módulos de E/S

- Unidade de Controle - controla as operações da CPU

- Tipos de Micro Operações:

- transferência de dados de um registrador para outro

- transferência de dados de um registrador para uma interface externa

- transferência de dados de uma interface externa para um registrador

- execução de uma operação lógica ou aritmética, utilizando registradores de E/S

- Funções da Unidade de Controle:

- Sequenciamento: é quem controla a sequência certa das micro-instruções a serem executadas

- Execução: origina micro-instruções para serem executadas

- Sinais de Controle: são a chave de como funciona a Unidade de Controle. Para que a UC possa executar suas funções é necessário ter *inputs* (avaliar o estado do sistema) e *outputs* (controlar o comportamento do sistema).

- **Input**

- Relógio - a UC executa uma micro-operação em cada pulso de relógio

- Instruction Register - o OP-CODE da instrução atual é usado para determinar a

micro-operação que será realizada durante o ciclo de execução

- Flags - determinar o estado atual da CPU e das operações da ULA

- Sinais de Controle de Bus de Controle - envia sinais para a UC, sinais de interrupção e de reconhecimento

- Output

- Sinais de Controle com a CPU - esses sinais são de dois tipos: os que causam a movimentação de dados de um registrador para outro e os que ativam funções específicas da ULA

- Sinais de Controle para o Bus de Controle - existem dois tipos: sinais de controle para a memória e para os módulos de E/S

- Organização Interna:

- Ula e os registradores da CPU estão ligados a um simples bus interno

- Portas e sinais de controle proporcionam o movimento de dados para e do bus para cada um dos registradores

- Formas de Implementação: UC tem duas categorias de implementação:

- Implementações por Hardware

- a UC é um circuito em que seus sinais lógicos são transformados no conjunto de sinais lógicos na saída, que são os sinais de controle

- a lógica da UC deverá ter uma única entrada lógica para cada opcode. Pode ser executada por um decodificador que recebe um input codificado e produz um simples output

- Problemas:

- * Lógica complexa de sequenciamento e micro-operação

- * Difícil de projetar e testar

- * Inflexível

- * Difícil de adicionar novas instruções

- * Solução: utilizar a microprogramação para a implementação da UC

- Implementações Microprogramadas

CAPÍTULO 15 - IMPLEMENTAÇÃO MICROPROGRAMADA

- Na utilização de sinais de controle, cada micro-operação é descrita em uma anotação simbólica, que se assemelha a uma linguagem de programação
- Sequência de microinstruções é conhecida como microprograma ou firmware
- O conjunto de microinstruções é denominado *microinstruções horizontais*
- Cada microinstrução é constituída por um bit, por cada linha interna de controle da CPU e por um outro bit por cada linha de controle do bus do sistema. Existe ainda um campo de condição, indicando a condição a qual deverá ocorrer um salto, e um campo com o endereço da microinstrução a ser executada quando esse salto ocorrer

Execução de uma instrução:

1. Para executar uma instrução, a unidade lógica de sequenciamento envia um comando de leitura (READ) para a memória de controle
 2. A palavra na qual o endereço é especificado no registrador de controle de endereço é lida para o registrador de controle do buffer
 3. O conteúdo do registrador de controle do buffer gera um sinal de controle e a informação em relação ao endereço para a unidade lógica de sequenciamento
 4. A unidade lógica de sequenciamento carrega o novo endereço no registrador de controle de endereço, baseado na informação do próximo endereço do registrador de controle do buffer e das flags da ULA
- No final de cada microinstrução, a unidade lógica de sequenciamento carrega o novo endereço para o registrador de controle de endereço.
 - Dependendo do valor das flags da ULA e do registrador de controle do buffer, umas das três decisões é tomada:

- Adquirir a próxima instrução
- Saltar para uma nova rotina baseada numa microinstrução de salto
- Saltar para a próxima instrução de máquina

- Vantagens:

- Simplificação do projeto, mais barato e menos sujeito a erros
- A implementação em hardware tem de incluir lógica complexa e um elevado número de micro-operações do ciclo de instrução

-Desvantagens:

- Decodificadores e unidades lógicas de sequenciamento baseiam-se em lógicas muito simplificadas
- São mais lentas que as UC baseadas em hardware