Resumão de Arquitetura de Computadores

Autor: Adonai Estrela Medrado (adonaimedrado@hotmail.com)

Construído tendo como referência FERREIRA, Filipe de Castro. **Módulo de arquitetura de computadores**. Vitória: ESAB, 2007. Disponível apenas para alunos.
Este material é apenas um resumo disponibilizado para uso acadêmico e didático. Ele não pretende esgotar o assunto ou abordá-lo em sua integralidade. A sua utilidade é relembrar alguns conceitos e estimular o aprofundamento e a pesquisa.

Breve Histórico

Data de 1623 a primeira calculadora mecânica e da metade do século XIX o primeiro computador mecânico. Este último nunca foi construído, mas concebido por Charles Babbage (precursor do computador) e teve a colaboração de Ada Augusta Byron (precursora do software). Em 1853 George Boole publicou seu trabalho que daria origem a álgebra de Boole, fundamento da informática moderna. Os cartões perfurados foram inventados por Herman Hollerith, funcionário do censo norte-americano; eles agilizaram em cem vezes o processamento censitário. A companhia de Hollerith deu origem à IBM.

O período de 1943 a 1945 foi marcado pelo ENIAC, o primeiro computador programável universal conseguia realizar cinco mil operações por segundo e seu funcionamento era baseado em válvulas e no sistema decimal. A programação era manual através de operações de ligar/desligar e conectar/desconectar chaves e cabos. O resultado era em forma de luzes. A maioria dos programadores eram mulheres.

O período entre 1945 e 1952 foi marcado pelo EDVAC e as ideias de Von Neumann para adoção do sistema binário e alguma forma de armazenamento ordenado. O primeiro *bug* foi causado em uma mariposa num Mark II em 1945. O Mark I foi construído com a colaboração Alan Turing entre 1939 e 1944. Nele, em 24 de junho de 1948, foi executado o primeiro programa gravado.

Máquina de Von Neumann

A arquitetura base dos computadores atuais é a máquina de Von Neumann. Ela possibilita resolver qualquer problema exprimível em forma de algoritmo. Seus

componentes são a *memória principal* (guarda dados usando a representação binária), a *unidade central de processamento* (UCP; controle das operações e execução das instruções) e os *dispositivos de entrada e saída* (comunicação com o exterior). A UCP é composta pela *unidade lógica aritmética* (ULA; executa as operações lógicas e aritméticas) e pela *unidade de controle* (UC; controla as operações interna e externas à UCP). A UCP pode ser fixa ou programável e executa três ações: busca, decodificação e execução. As instruções de máquina podem envolver: operações aritméticas, lógicas, de controle de sequência de execução e de transferência entre a UCP e a memória central ou as unidades de entrada e saída.

Unidades, Sistemas Numéricos e Números Negativos

O bit só pode ter ou o valor zero ou o valor um. Agrupamentos de bits criam significados úteis. O byte é um grupo ordenado de oito bits. O conceito de palavra está relacionado ao processamento, armazenamento e transferência. A palavra é fixa e constante em um dado processador.

O sistema decimal trabalha com potências de 10, o binário com potências de dois, o hexadecimal com potências de 16.

Um *oveflow* ocorre quando o resultado de uma operação ultrapassa o limite máximo possível. É esperada uma sinalização do processador sobre este fato.

Há três formas convencionadas para representar números negativos: *sinal e magnitude* (bit mais a esquerda indica o sinal; 0 com positivo, 1 como negativo; o restante dos bits é a magnitude - o valor de fato; há duas representações do zero), *complemento de um* (números negativos representados pelo complemento de um do número positivo), *complemento de dois* (mais utilizado; números negativos representados por duas operações sucessivas com o número positivo: primeiro faz-se o complemento de um e depois ao resultado soma-se um; acarreta em faixas assimétricas - mais números negativos que positivos; possui uma única representação para o número zero).

Memória: tipos, características e classificação

Os diversos tipos de memória utilizadas no computador são interligadas de forma estruturada formando o subsistema de memória. As principais características da memória são custo (preço por byte), tempo de acesso (espaço de tempo entre a solicitação e a

disponibilização) e persistência do armazenamento (volatilidade e tempo de permanência da informação). As memórias podem ser ordenadas hierarquicamente começando-se das com menor custo, velocidade e capacidade da seguinte forma: memória secundária (ótica e magnética), memória principal, memória *cache*, registradores. Existem diversas tecnologias de fabricação dentre elas as memórias de semicondutores, as memórias de meio magnético e as memórias óticas. A volatilidade diz respeito à capacidade de reter a informação armazenada quando a energia elétrica é desligada. Os registradores e a memória RAM são voláteis. As memórias magnéticas, óticas, ROM, EPROM são não voláteis. Tempo de ciclo de memória é o espaço de tempo entre duas operações sucessivas na memória.

As memórias podem ser classificadas quanto à leitura e escrita em R/W (*Read and Write*; geralmente denominadas de RAM), ROM (*Read Only Memory*; uma vez gravada não pode ser apagada), PROM (*Programmable Read Only Memory*; processo de programação mais simples que os da ROM; não pode ser apagada), EPROM (*Erasable Programmable Read Only Memory*; o processo de apagar é feito com máquinas adequadas ou raios ultra-violeta) e EEPROM ou E2PROM ou EAROM (*Electrically Erasable Programmable Read Only Memory/Eletrically Alterable ROM*; reprogramavel por processo elétrico com equipamento e software adequado) .

A memória principal é controlada pelo sistema operacional e armazena programas e dados enquanto estes estão sendo necessários para a UCP. A célula é a menor unidade endereçável; seu tamanho depende da arquitetura da máquina. Cada célula tem um endereço único.

A função da memória *cache* é acelerar o processamento. A memória *cache* é mais rápida que a memória principal, sendo compatível com a velocidade da UCP. Seu funcionamento baseia-se no princípio da *localidade temporal* (probabilidade de um mesmo item ser referenciado novamente em um curso espaço de tempo) e *localidade espacial* (probabilidade de itens com endereços próximos serem referenciados em um curto espaço de tempo). Existem duas possibilidades quando o processador busca uma informação no *cache*: ou ela está lá (*cache hit*) ou ela não está (*cache miss*). Denomina-se tempo de acerto o tempo necessário para acessar uma informação que está no *cache*. Penalidade por falta é o tempo gasto substituindo informações que não estão no *cache*.

A quantidade de registradores depende do modelo e fabricante do processador. Poucos registradores resultam em mais referência à memória, entretanto um número muito grande não reduz significativamente a referência à memória. O tamanho do registrador de endereços deve ser suficiente para o endereçamento da arquitetura. O tamanho do registrador de dados deve atender à maioria dos tipos de dados disponíveis na arquitetura. Existem dois tipos de registradores: os *visíveis para os usuários* e os de *controle e estado* usados por programas privilegiados do sistema operacional. Os do primeiro tipo podem ser *de uso geral*, *de dados* e *de endereços*. Os registradores de endereço podem ser de segmento, de índice ou apontadores de topo de pilha. São quatro os registradores essenciais de controle de estado: contador de instrução (CI; indica a próxima instrução a ser executada), registrador de instrução (RI; contém a última instrução); registrador de endereçamento à memória (MAR; contém o endereço de uma posição de memória) e registrador de armazenamento temporário de dados (MBR; contém uma palavra de dados lida ou a ser escrita).

Programa e Linguagem

Um programa é uma sequência lógica de instruções com o objetivo de realizar determinada tarefa. Uma linguagem de programação permite escrever programas de computador. Linguagem de máquina é única que o computador pode entender diretamente; nela cada instrução corresponde a uma ação (relação um para um); sua especificidade é para uma família de computadores. Ela é um código binário totalmente vinculado ao conjunto de instruções da máquina. São disponibilizadas instruções para operações matemáticas, movimentação de dados, entrada/saída e controle (desvio de sequência de execução). Cada instrução é identificada unicamente por um código binário e é composta por um código de operação (OPCODE) e, se for o caso, parâmetros (ou operandos; OP 1...OP N). O ciclo de instrução é o processamento necessário para executar uma instrução. Um ciclo é composto pelas seguintes etapas: 1) busca e armazenamento no registrador de instrução da próxima instrução; 2) incremento do contador de instrução; 3) determinação do tipo de instrução, determinar, se for o caso, os endereços dos operandos; 4) buscar, se for o caso, operandos na memória; 5) executar instrução.

A linguagem de montagem (*Assembly Language*) substitui os códigos numéricos da linguagem de máquina por mnemônicos; ela é também dependente de uma família de

máquinas; o processo de transformação da linguagem de montagem para a linguagem de máquina é denominado de montador (*assembler*). As linguagens de alto nível são afastadas do nível da máquina; o processo de conversão da linguagem de alto nível para a linguagem de máquina chama-se compilação; uma linguagem de alto nível geralmente faz chamada a diversas funções pré-programadas (bibliotecas); o ligador (*linker*), dentre outras funções, decide como tratar a referências às bibliotecas; para um programa em linguagem de alto nível ser executado ele precisa passar pela compilação e pela linkedição (feito no momento da construção do programa) ou pela interpretação (feito no momento da execução); a interpretação é o processo que executa toda a tradução e resolução de referências externas em tempo de execução. Na técnica de compilação o tempo de execução é menor e a identificação de erros é mais difícil em tempo de execução do que na técnica de interpretação onde o consumo de memória é maior e o processo (de tradução e resolução de referências) tem que ser repetido todas às vezes quando o código for executado.

Máquina virtual é uma camada de emulação criada em um ambiente de interpretação. Este ambiente de emulação faz com que uma máquina se comporte como outra. Na linguagem Java o código é compilado para o bytecode (código intermediário) que é depois interpretado pela Máquina Virtual Java (JVM).

Modos de Endereçamento

A forma como os operandos indicam os dados a serem processados é chamada de modo de endereçamento. São quatro os principais modos de endereçamento: *modo imediato* (operando indica o valor do dado), *modo direto* (indica o endereço da memória), *modo indireto* (operando indica um ponteiro para o dado), *modo de endereçamento por registrador* (operando indica um registrador). O modo de endereçamento por registrador pode ser *direto* (registrador contém o dado a ser manipuado) ou *indireto* (registrador armazena o endereço de uma célula de memória).

Pipeline

O objetivo da técnica de *pipelining* é melhorar o desempenho do processador aumentando seu *throughput*. O tempo para executar uma operação em *pipeline* geralmente é ligeiramente maior que o tempo para executá-la sem *pipeline*, graças a *overheads* como, por exemplo, o de transferência de dados. A transferência de dados entre os estágios do

pipeline pode ser assíncrona (utiliza sinais de *handshaking* para indicar disponibilidade de dados) ou síncrona (utiliza *latches* para armazenar dados intermediários durante a transferência de estágios; controlado por sinal de relógio). O método assíncrono permite maior velocidade, mas o método síncrono é mais adotado devido à sua simplicidade.

Interface de Entrada e Saída, Barramento

A interface de entrada e saída (E/S) controla os periféricos de modo a proporcionar transparência para o processador. Nesta operação estão envolvidos no mínimo o registrador de dados (envio/recebimento de dados), o registrador de controle (envio de comandos) e o registrador de estado (indica os estados da operação e condições de erro).

Nas operações de E/S as três principais técnicas para transferência de dados são *polling* (técnica de implementação em *software*; utiliza o *done bit* do registrador de estado para sinalizar o fim da transferência de dados entre a memória e a interface E/S; exige dedicação exclusiva do processador), interrupção (requer suporte de *hardware*; não exige dedicação; interface sinaliza ao processador sobre fim de operação de E/S; o controlador de interrupção decide, com base em uma lista de prioridades, qual pedido de interrupção será atendido em caso de pedidos simultâneos; para decidir que rotina será executada o processador consulta a tabela de vetores de interrupção que contém ponteiros para as rotinas correspondentes a cada interrupção) e acesso direto à memória ou *Direct Memory Access* (DMA) (o processador não participa da transferência de dados; o controlador de DMA controla o processo de transferência de dados entre a memória e a interface de E/S e arbitra pedidos simultâneos; o controlador de DMA solicita o barramento ao processador com o sinal PDMA; o processador libera o barramento para a operação com o sinal LIVRE e entra no *hold state*; ao final do processo de DMA o processador é liberado para utilizar o barramento).

O barramento de entrada e saída forma um padrão de comunicação entre o processador e os dispositivos. Ele consiste em um caminho comum pelo qual os dados trafegam; seu tamanho determina quantos dados podem ser transmitidos por vez e sua largura de banda precisa ser compartilhada. Há três conjuntos de barramentos: barramento de endereço (unidirecional; os endereços de memória e dispositivos trafegam do processador), barramento de dados (bidirecional; os dados da memória e dos dispositivos de E/S trafegam do e para o processador) e barramento de controle (bidirecional; os sinais

de controle da memória e dos dispositivos E/S trafegam do e para o processador; indica estado ou determina operação de leitura/gravação). Os eventos de interação com o barramento ocorrem em sincronia com o sinal do *clock*.

O barramento ISA (Industry Standard Architecture) opera a 8Mhz com transferência de 8 ou 16 bits. O EISA (*Extended Industry Standard Architecture*) amplia o ISA para 32 bits e permite compartilhamento do barramento por processadores; compatível com o ISA. MCA (*MicroChannel Architecture*) é um barramento de 32 bits da IBM com suporte a multiprocessamento e desenvolvida para prevenir conflito; não é compatível com o ISA. VESA (*Video Electronic Standards Association*) Local Bus é um barramento de 32 bits compatível com ISA utilizado para principalmente para vídeo e controladores de disco. O PCI (*Peripheral Component Interconnect*) opera na faixa entre 25 e 33 Mhz, permite a transferência a 32 ou 64 bits com taxas de até 132 MB/s; incorpora recursos *Plug and Play*. O AGP (*Accelerated Graphic Port*) é exclusivo para placas de vídeo; suas taxas podem chegar a 1GB/s e dependem da placa de vídeo e da frequência do barramento da placa-mãe. O USB (*Universal Serial Bus*) permite a comunicação serial de até 127 periféricos por porto; também incorpora os recursos *Plug and Play*; dispositivos podem ser conectados e desconectados com a máquina ligada; capaz de fornecer até 5 V de energia. O Firewire (IEEE 1394) é um barramento serial que permite transferência de até 400 Mbps e aceita até 63 dispositivos por porto.

Portas de Comunicação

As portas de comunicação permitem a interligação física com os periféricos. Na comunicação serial o byte é desmembrado em bits e enviado separadamente um após o outro. Na comunicação paralela os bits do byte são enviados simultaneamente por meios físicos diferentes (cada um em seu fio). Na interface paralela pode ocorrer dos bits não chegarem todos ao mesmo tempo (*skew*).