

Máquina Multinível

Um programa pode ser definido como uma seqüência de instruções que descrevem como executar uma determinada tarefa.

Uma instrução pode ser definida como um comando para o processador.

Linguagem de máquina corresponde ao conjunto de instruções primitivas do computador. Devem ser simples, reduzindo a complexidade e custo dos circuitos.

Tradução é o método pelo qual um programa escrito numa linguagem L2 é substituído por um outro programa escrito em L1, então executado pela máquina M1, cuja linguagem de máquina é L1.

Máquina Multinível

Interpretação é o método pelo qual um programa escrito em L1 executa cada instrução do programa escrito em L2, através de uma seqüência de instruções L1 equivalentes.

Máquina virtual compreende um computador hipotético para uma determinada linguagem, tendo esta como linguagem de máquina.

Para uma melhor compreensão da arquitetura de um computador, podemos dividi-lo em níveis de hierarquia. Nível compreende uma máquina M com sua linguagem de máquina L, dentro de um computador.

Uma máquina define uma linguagem, assim como uma linguagem define uma máquina.

Máquina Multinível

Um computador com n níveis pode ser visto como n diferentes máquinas virtuais, cada uma com a sua linguagem de máquina. 

Um programador de nível n não precisa conhecer os níveis inferiores.

Nível 0 ou nível de lógica digital é o *hardware* verdadeiro da máquina, cujos circuitos executam os programas em linguagem de máquina de nível 1. Não existe aqui o conceito de programa como uma seqüência de instruções a serem executadas. Neste nível, os objetos são denominados *portas*.

Máquina Multinível

Nível 1 ou nível de microprogramação é o verdadeiro nível de máquina, havendo um programa denominado *microprograma*, cuja função é interpretar as instruções de nível 2. A instrução neste nível é denominada *microinstrução*.

Nível 2 ou nível convencional de máquina é o primeiro nível de máquina virtual. A linguagem de máquina deste nível é comumente denominada *linguagem de máquina*. As instruções são executadas interpretativamente pelo microprograma. Em máquinas que não tenham o nível de microprogramação, as instruções de nível convencional de máquina são executadas diretamente pelos circuitos.

Máquina Multinível

Nível 3 ou nível de sistema operacional apresenta a maior parte das instruções em linguagem de nível 2, um conjunto de novas instruções, organização diferente da memória, capacidade de execução de dois ou mais programas em paralelo. As novas facilidades são realizadas por um interpretador denominado *sistema operacional*, em execução no nível 2. As instruções de nível 3 idênticas às de nível 2 são executadas diretamente pelo microprograma.

Máquina Multinível

Nível 4 ou nível de linguagem de montagem consiste de uma forma simbólica para uma linguagem de nível inferior. Os programas em linguagem de montagem são traduzidos para uma linguagem de nível 2 ou 3, e, então, interpretados pela máquina apropriada. O programa que executa a tradução é denominado *montador*.

Nível 5 ou nível de linguagem orientada para problemas consiste de linguagem de alto-nível. Os programas escritos nessas linguagens são, normalmente, traduzidos para o nível 3 ou 4 por tradutores conhecidos como *compiladores*.

Máquina Multinível

Os níveis 2 e 3 são sempre interpretados, enquanto os níveis 4 e 5 são, geralmente, traduzidos.

As linguagens de máquina dos níveis 1, 2 e 3 são numéricas, ao passo que as dos níveis 4 e 5 são simbólicas, contendo palavras e abreviaturas.

O *hardware* é constituído pelos circuitos eletrônicos e o *software* é constituído pelos programas. O *firmware* consiste no *software* embutido em dispositivos eletrônicos durante a fabricação. Em muitos computadores, o microprograma está em *firmware*.

Máquina Multnível

Hardware e *software* são logicamente equivalentes, isto é, qualquer operação efetuada pelo *software* pode também ser implementada pelo *hardware* e qualquer instrução executada pelo *hardware* pode também ser simulada pelo *software*.

A decisão de colocar certas funções em *hardware* e outras em *software* baseia-se em fatores tais como: custo, velocidade, confiabilidade e possíveis modificações.

Nível de Microprogramação

O nível de microprogramação tem uma função específica: *executar interpretadores para outras máquinas virtuais*.

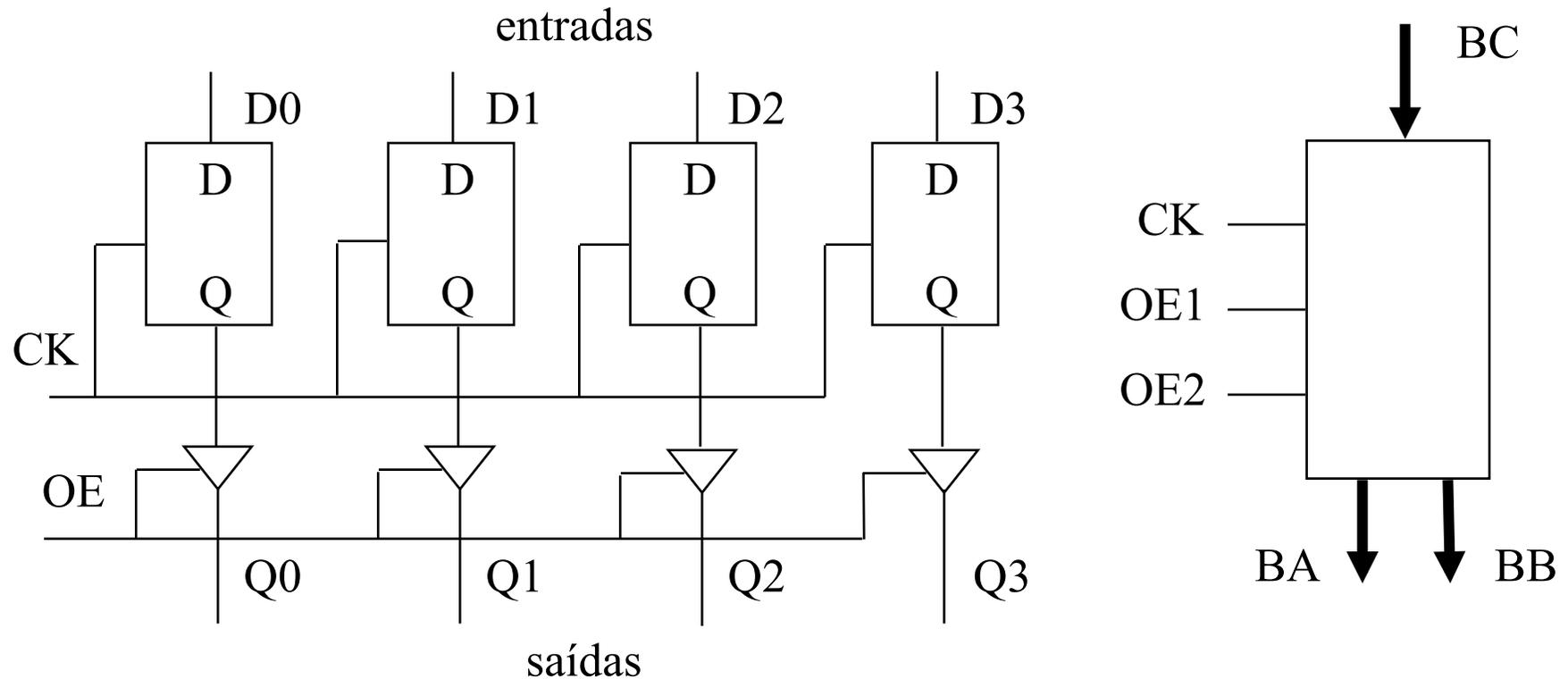
Um *microprograma* compreende um programa que controla os registradores, os barramentos, a ULA, as memórias e outros componentes do *hardware*.

Os registradores estão localizados fisicamente dentro do processador.

Um *barramento* é uma coleção de fios usados para transmitir sinais em paralelo. Pode ser *unidirecional* ou *bidirecional*.

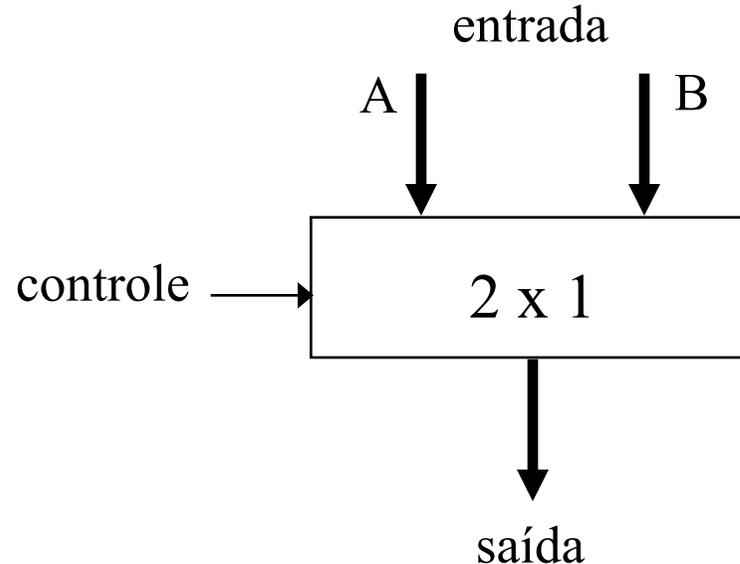
Nível de Microprogramação

Um barramento *tri-state* possui dispositivos capazes de apresentar na saída 0, 1 ou alta impedância. Utilizados quando há muitos dispositivos ligados a um mesmo barramento.



Nível de Microprogramação

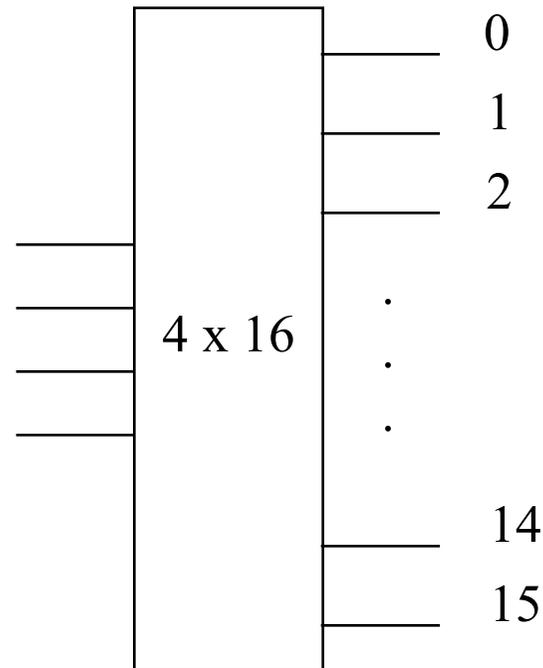
Um multiplexador tem 2^n entradas, uma saída da mesma largura da entrada e uma entrada de controle de n bits, que seleciona uma das entradas e a direciona para a saída.



Um *demultiplexador* é o inverso de um multiplexador, direcionando a entrada para uma dentre 2^n saídas, de acordo com as n linhas de controle.

Nível de Microprogramação

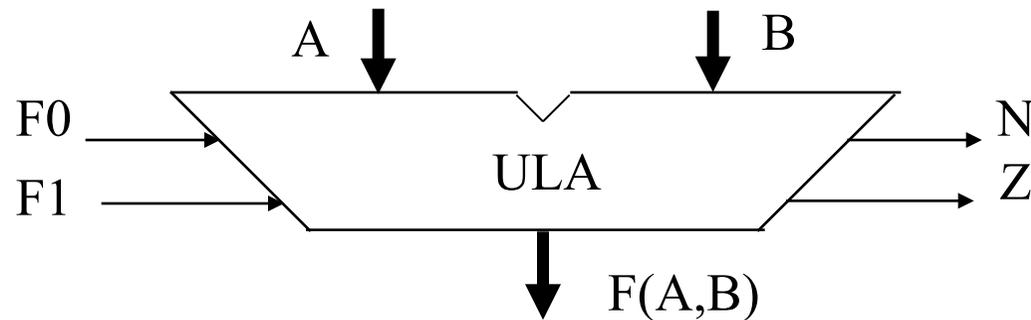
Um decodificador tem n linhas de entrada e 2^n linhas de saída. De acordo com o código binário da entrada, uma das saídas é ativada.



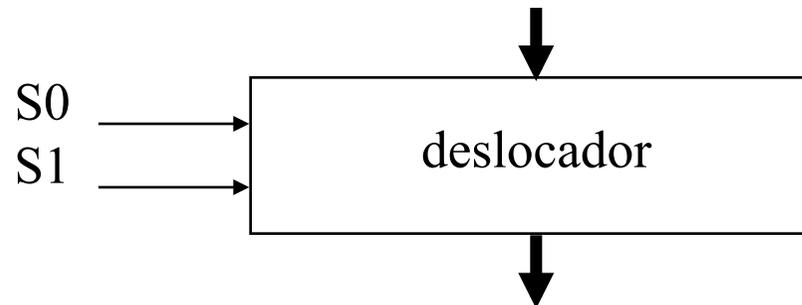
Um codificador é o inverso de um decodificador, possuindo 2^n entradas e n saídas. Somente uma das entradas estará ativa.

Nível de Microprogramação

A Unidade Lógica e Aritmética possui duas entradas e uma saída para dados, havendo outras entradas e saídas de controle.



Um deslocador é um circuito com capacidade para deslocar à direita ou à esquerda, ou mesmo não deslocar.



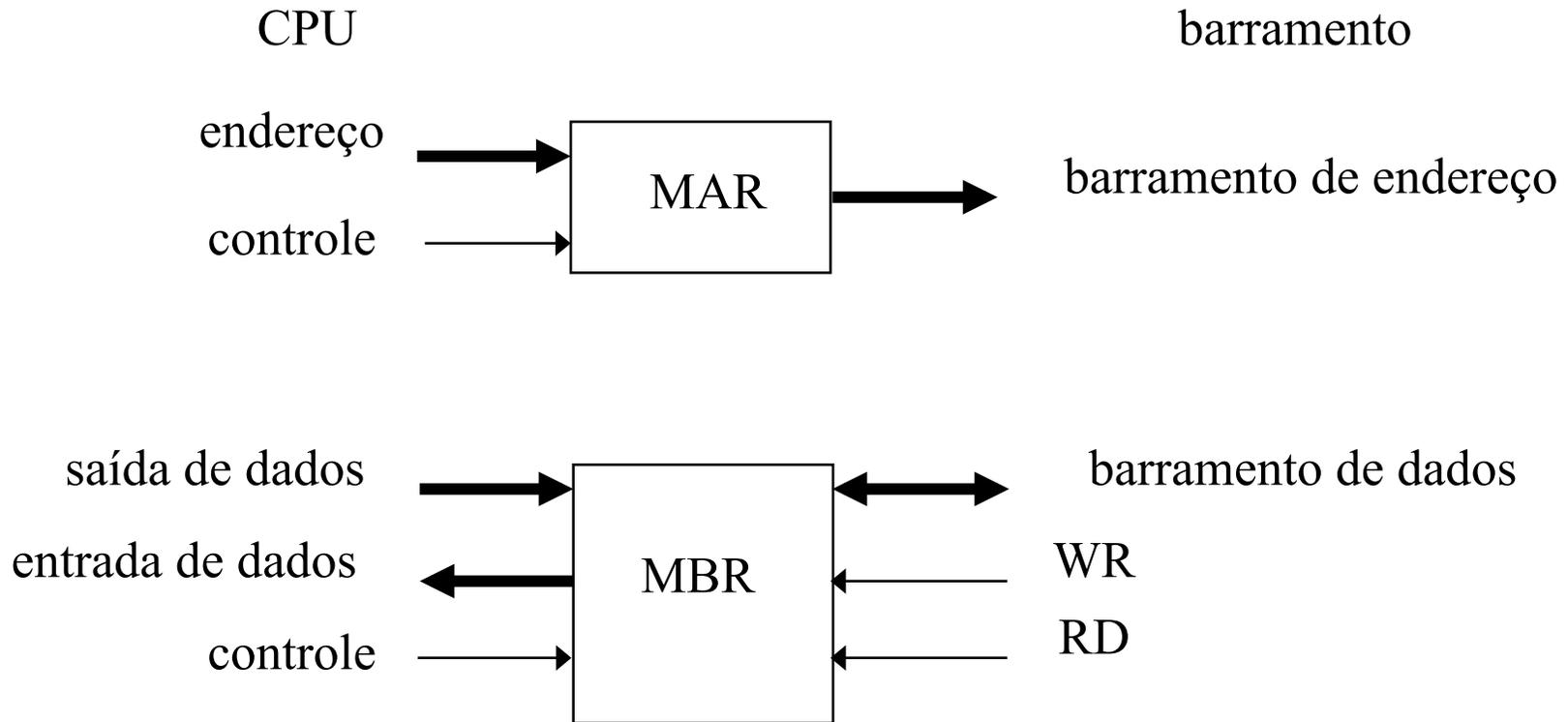
Nível de Microprogramação

A maioria dos computadores tem um barramento de endereço, um barramento de dados e sinais de controle para a comunicação entre a UCP e os demais componentes do sistema.

Um acesso à memória é quase sempre consideravelmente mais demorado que o tempo necessário para executar uma única microinstrução.

O registrador MAR é responsável pelo armazenamento do endereço da memória. O registrador MBR é responsável pelo armazenamento do dado.

Nível de Microprogramação



A linha de controle de MBR permite carregar o registrador com dado da UCP. O sinal RD carrega o registrador com dado do barramento. O sinal WR libera o conteúdo do registrador no barramento. ▶

Nível de Microprogramação

Um formato de microinstrução, contendo alguns campos codificados, pode ser:

1	2	2	2	1	1	1	1	1	4	4	4	8
A	C	A	S	M	M	R	W	E				
M	O	L	H	B	A	D	R	N	C	B	A	ADDR
U	N	U		R	R			C				
X	D							C				

AMUX : 0 = latch A; 1 = MBR

COND: 0 = não salta; 1 = salta se N=1;

2 = salta se Z=1; 3 = salta sempre

ALU: 0 = A+B; 1 = A.B; 2 = A; 3 = NOT A

SH: 0 = não desloca; 1 = desloca 1 bit à direita;

2 = desloca 1 bit à esquerda; 3 = x

Nível de Microprogramação

Um ciclo básico consiste em colocar os valores nos barramentos A e B, armazená-los nos dois *latches*, passá-los pela ALU e pelo deslocador, e armazená-los na memória local ou no MBR. O seqüenciamento dos eventos compreende:

- 1 - carregar a próxima microinstrução no registrador de microinstrução (MIR);
- 2 - colocar o conteúdo dos registros nos barramentos A e B, e guardá-los nos *latches* A e B;
- 3 - dar tempo à ALU e ao deslocador para produzirem um resultado e carregar o MAR, se necessário;
- 4 - armazenar o valor existente no barramento C, na memória local ou no MBR.

Nível de Microprogramação

O microprograma, para a arquitetura proposta, deve realizar a busca, decodificação e execução da instrução do programa de nível convencional de máquina. ▶

O registrador AMASK é a máscara de endereço (Ox007777) usada para separar os bits de endereço do restante da instrução.

O registrador SMASK é a máscara de pilha (Ox000377) usada para separar a constante, associada às instruções INSP e DESP, do restante da instrução.

Para realizar uma subtração utiliza-se complemento a dois:

$$x - y = x + \bar{y} + 1$$

Nível de Microprogramação

A microlinguagem de montagem consiste dos seguintes comandos:

- 1 - atribuição: $AC := A;$
- 2 - aritmética: $PC := PC + 1;$
- 3 - lógica: $A := \text{band} (IR, SMASK);$
 $B := \text{inv} (C);$
- 4 - deslocamento: $TIR := \text{lshift} (TIR);$
 $D := \text{rshift} (A + B)$
- 5 - desvios: $\text{goto } 0;$
 $\text{if } N \text{ then goto } 50;$

Nível de Microprogramação

Em muitos computadores, a microarquitetura tem suporte de hardware para extrair código de operação da macroinstrução e colocá-lo diretamente no MPC.

Não há instruções de E/S, utilizando E/S mapeada em memória.

(4092) - dado a ser lido

(4093) - o bit de sinal indica dado para leitura (=1)

(4094) - dado a ser escrito

(4095) - o bit de sinal indica dispositivo pronto (=1)

Microprogramação horizontal: microinstrução com campos muito pouco codificados.

Microprogramação vertical: microinstrução com campos mais codificados.

Um microprograma é mais vertical quanto maior for o grau de codificação da microinstrução.

Uma microinstrução extremamente vertical poderia ter um código de operação e operandos. 

