



Organização de Computadores Digitais

Operação da Unidade de Controle
e Controle Micro Programado



Unidade de Controle

- Até agora falamos do modo como são constituídas as instruções contendo um opcode e um modo de endereçamento próprio
- Se conhecermos o conjunto de instruções máquina e se soubermos quais os registradores que estão visíveis para o usuário, sabemos também quais as funções que a CPU pode executar
- No entanto é necessário conhecer as interfaces externas, geralmente através do bus e o modo como as instruções são tratadas



Unidade de Controle

- Para especificar as funções do sistema temos que levar em conta:
 - 1 - Operações (opcodes)
 - 2 - Modos de endereçamento
 - 3 - Registradores
 - 4 - Interfaces de entrada e saída
 - 5 - Interfaces
 - 6 - Estrutura de processamento de interrupções
- Os itens 1 ao 3 são definidos pela definição de instruções, os itens 4 e 5 são geralmente definidos pelo bus de sistema, o item 6 é definido parcialmente pelo bus do sistema e parcialmente pelo tipo de suporte que a CPU oferece ao sistema operacional.



Unidade de Controle

- o Estes seis termos determinam o que é que a CPU faz.
- o A partir daqui, vamos abordar como é que estas operações são executadas ou mais propriamente, como é que os vários elementos do CPU são controlados para realizar essas funções.



Micro Operações

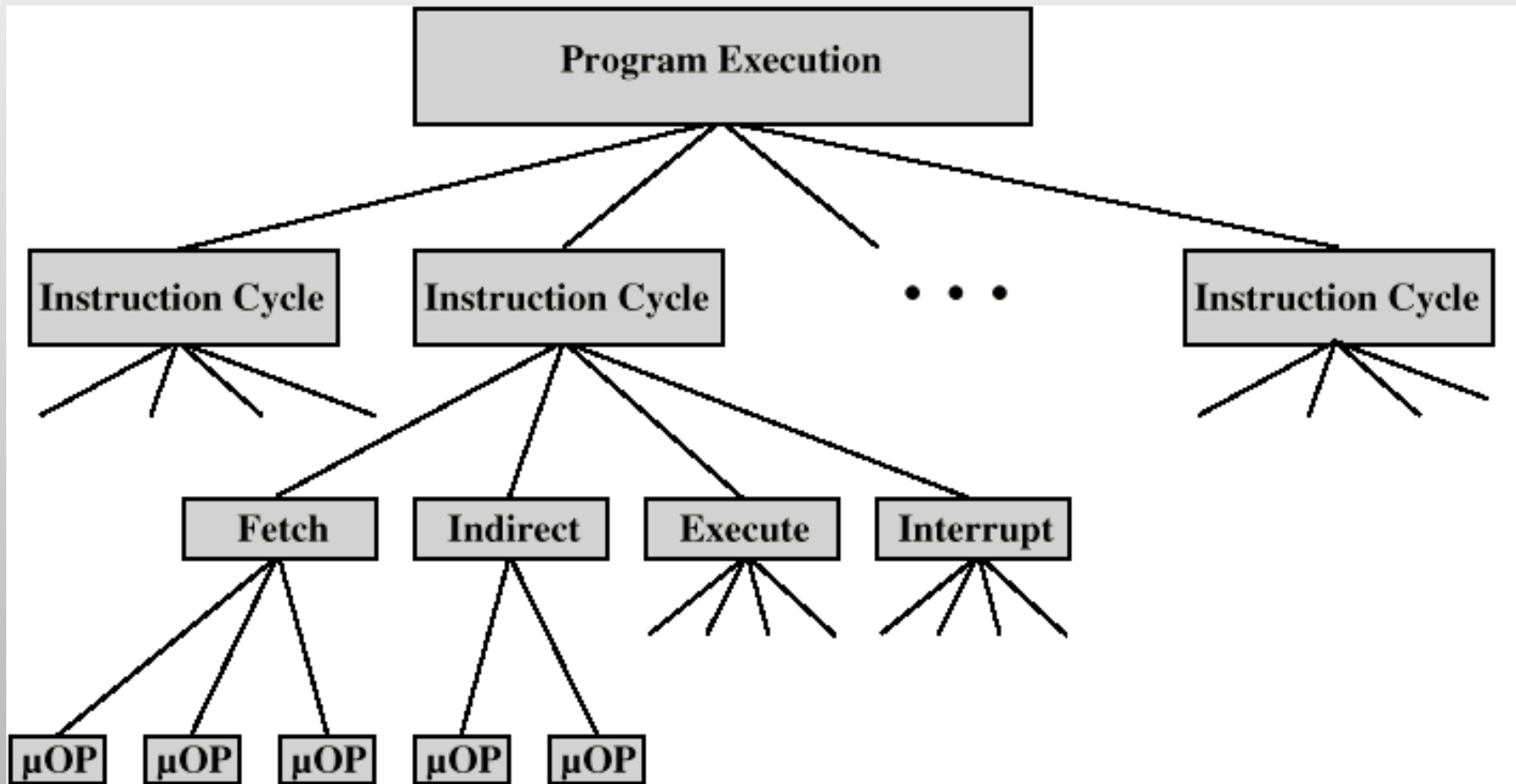
- A função do computador é executar programas.
- Como já vimos anteriormente, a execução de um programa consiste na sequência de ciclos de instruções, ao ritmo de uma instrução máquina por ciclo.
- A execução de um programa torna-se numa execução temporal de instruções.
- Vimos também que cada ciclo de instrução pode ser considerado como um número menor, de pequenas unidades.
- Para desenhar uma unidade de controle, será necessário analisar esta descrição.



Micro Operações

- Cada simples ciclo envolve um conjunto de vários passos, cada um dos quais envolvendo registradores da CPU.
- Podemos referirmo-nos a estas operações como microoperações.
- O prefixo micro refere-se ao fato de que cada passo é na verdade muito simples.
- Cada instrução é executada durante um ciclo de instrução constituindo vários subciclos (busca, execução, interrupção).
- O desempenho de cada subciclo envolve uma ou mais pequenas operações que são as micro-operações.

Elementos da Execução de Programa

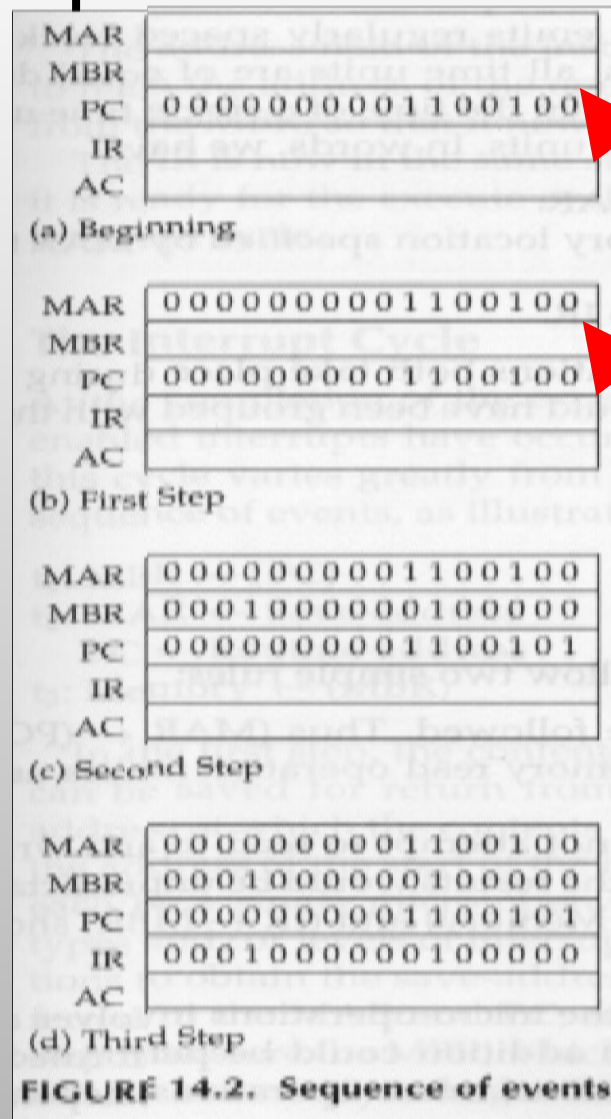




Ciclo de Busca - 4 Registradores

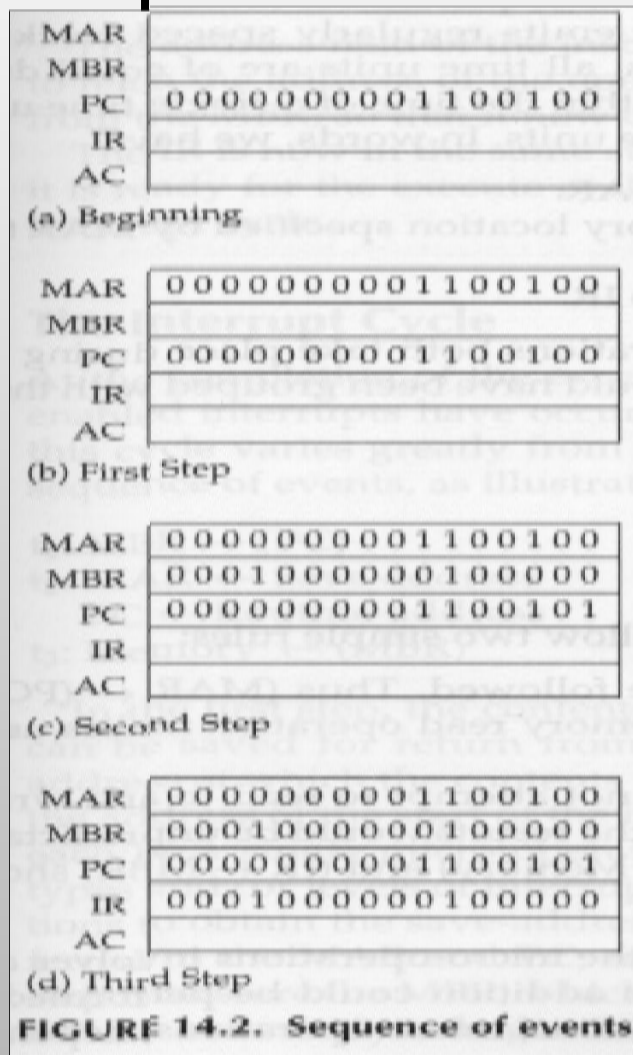
- Memory Address Register (MAR)
 - Conectado às linhas de endereço do bus do sistema
 - Especifica endereçamento na memória para operação de leitura ou escrita
- Memory Buffer Register (MBR)
 - Conectado às linhas de dados do bus
 - Contém dados para escrita na mem ou os últimos dados lidos
- Program Counter (PC)
 - Contém endereço da próxima instrução a ser buscada
- Instruction Register (IR)
 - Contém a última instrução buscada

Sequência do Ciclo de Busca



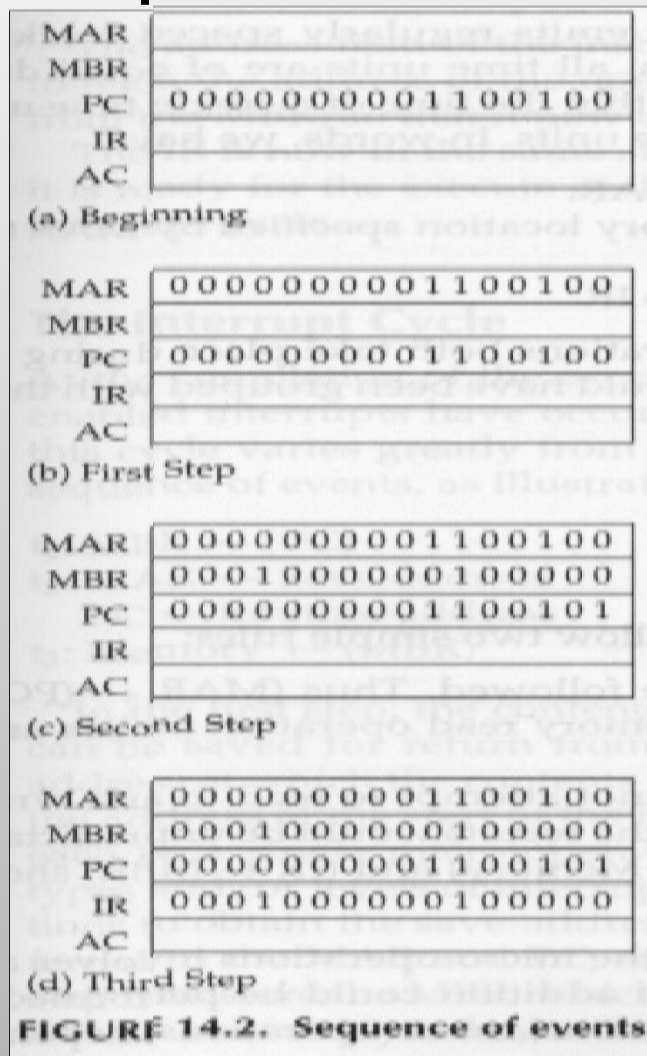
- No início do ciclo, o endereço da próxima instrução a ser executada é colocado no PC
- Neste caso é 00000001100100
- O primeiro passo é mover esse endereço para o MAR.
- A partir daqui este é o único registrador ligado às linhas de endereço do bus do sistema.

Sequência do Ciclo de Busca



- O passo seguinte é adquirir a instrução.
- O endereço desejado 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, é então copiado para o MBR.
- É necessário também incrementar o PC
- uma unidade para obter a próxima instrução.
- Como estas operações (ler da

Sequência do Ciclo de Busca



- O terceiro passo, consiste em mover o conteúdo do MBR para o IR.
- Isto libera o MBR para utilização durante um possível ciclo de indireto.
- Assim sendo, um possível ciclo de busca consiste em ter 3 passos e 4 micro-operações.
- Cada micro-operação envolve o movimento de dados de ou para um registrador.



Sequência do Ciclo de Busca

- Assim sendo, um possível ciclo de aquisição consiste em ter 3 passos e 4 micro-operações.
- Cada micro-operação envolve o movimento de dados de ou para um registrador.
- Desde que os movimentos de dados não interfiram uns com os outros, mais movimentos podem decorrer no mesmo passo conseguindo uma poupança de tempo.



Sequência do Ciclo de Busca

Simbolicamente:

- A1: $MAR \leftarrow PC$ (mover os dados do PC para o MAR)
- A2: $MBR \leftarrow \text{Memória}$, $PC \leftarrow PC + 1$ (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 de Busca

- Nestas circunstâncias, cada pulso de relógio define uma unidade de tempo. Assim sendo, todas as unidades de tempo têm a mesma duração.
- Embora seja possível fazer o agrupamento da microoperação é necessário ter em conta as seguintes regras:
 - A sequência de passos tem de ser respeitada. Assim (MAR \leftarrow PC) tem de que ser antes de (MBR \leftarrow Memória) desde que a execução de leitura da memória utilize o endereço de MAR.
 - Os conflitos devem ser evitados, não devemos tentar ler e escrever do/no mesmo registrador ao mesmo tempo, pois os resultados são imprevisíveis.



Ciclo de Busca

- o Logo após uma busca de uma instrução, o próximo passo é buscar os operandos fonte.
- o Continuando o exemplo simples, assume-se uma instrução de um endereço, que permita endereçamento direto e indireto.
- o Se a instrução especificou um endereço indireto, então o ciclo indireto tem de preceder o ciclo de execução.



Ciclo Indireto

- o As micro-operações associadas são as seguintes:
 - o T1: $MAR \leftarrow (IR \text{ (Endereço)})$
 - o T2: $MBR \leftarrow \text{Memória}$
 - o T3: $IR \text{ (Endereço)} \leftarrow MBR \text{ (Endereço)}$
- o O campo endereço de uma instrução é transferido para o MAR. É então usado para adquirir o endereço do operando. Finalmente, o campo endereço do IR é atualizado para o MBR.
- o O IR tem o mesmo estado como se fosse usado um ciclo direto e está pronto para executar o ciclo.



Ciclo de Interrupção

- Sempre e logo após se complementar o ciclo de execução é feito um teste para determinar se ocorreu alguma interrupção.
- Em caso afirmativo damos início ao ciclo de interrupção.
 - T1: $MBR \leftarrow (PC)$
 - T2: $MAR \leftarrow \text{Save ADDRESS}$
 $PC \leftarrow \text{Routine ADDRESS}$
 - T3: $\text{Memória} \leftarrow MBR$



Ciclo de Interrupção

- o No primeiro passo, o conteúdo do PC é transferido para o MBR que assim pode ser gravado e retomado logo depois da interrupção.
- o O MAR é carregado com o valor que está no PC e o PC é atualizado com o endereço da rotina de processamento da interrupção.
- o O passo final é guardar o MBR que contem o velho valor do PC na memória.
- o A CPU está assim pronta para começar o próximo ciclo de instrução.



Ciclo de Execução

- Os ciclos de busca, indireto e interrupção são simples e previsíveis.
- Cada um envolve um pequeno conjunto, ou sequência de micro-operações e em cada caso, as mesmas operações são repetidas.
- Quanto ao ciclo de execução, uma máquina com N opcodes possui N sequências diferentes de microoperações que podem ocorrer.
- O endereço do PC, no início da instrução é o endereço da próxima instrução em sequência.

0 0 0

Ciclo de Execução (ADD)

- ADD R1,X - adiciona o conteúdo da posição X ao registrador 1 , resultado colocada em R1
- t1: $MAR \leftarrow (IR_{\text{address}})$
- t2: $MBR \leftarrow (\text{memory})$
- t3: $R1 \leftarrow (R1) + (MBR)$

0 0 0

Ciclo de Execução (ISZ)

- o ISZ X - incrementa e salta se for zero
 - o t1: $MAR \leftarrow (IR_{\text{address}})$
 - o t2: $MBR \leftarrow (\text{memory})$
 - o t3: $MBR \leftarrow (MBR) + 1$
 - o t4: $\text{memory} \leftarrow (MBR)$
 - o if $(MBR) == 0$ then $PC \leftarrow (PC) + 1$

0 0 0

Ciclo de Execução (BSA)

- o BSA X - Branch and save address
 - o Endereço da instrução seguinte a BSA é salvo em X
 - o Execução continua de X+1
 - o t1: $MAR \leftarrow (IR_{\text{address}})$
 - o $MBR \leftarrow (PC)$
 - o t2: $PC \leftarrow (IR_{\text{address}})$
 - o $memory \leftarrow (MBR)$
 - o t3: $PC \leftarrow (PC) + 1$



Requisitos Funcionais

- o Requisitos funcionais - são as funções que a unidade de controle tem de executar.
- o A definição destes requisitos funcionais é a base para o projeto e implementação da unidade de controle.
- o A Unidade de Controle é caracterizada por:
 - 1 - Definir os elementos básicos da CPU
 - 2 - Descrever as micro-operações às operações que a CPU pode executar
 - 3 - Determinar as funções que a unidade de controle tem de executar para que as micro-operações sejam executadas



Elementos Básicos do Processador

- ALU - é a componente funcional de um computador.
- Registradores - são usados para guardar dados internos na CPU. Alguns registradores contêm informação de estado, necessário para controlar sequências de informação (ex.: Program Status Word). Outros contêm dados que são enviados de e para a ALU, memória, módulos de I/O.
- Path de dados internas - são usadas para mover dados entre registradores e entre registradores e a ALU.
- Path de dados externas - ligam os registradores à memória e módulos de I/O.
- Unidade de controle - controla as operações da CPU.



Tipos de Micro Operações

Todas as micro-operações serão classificadas numa das seguintes categorias:

- 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 entrada e saída.

0 0 0

Funções da Unidade de Controle

- o Sequenciamento

- o UC é quem controla a sequência certa das micro-instruções de programas a serem executadas.

- o Execução

- o a UC origina micro-instruções para serem
 - o executadas.

- o Os sinais de controle são a chave de como funciona a Unidade de controle.

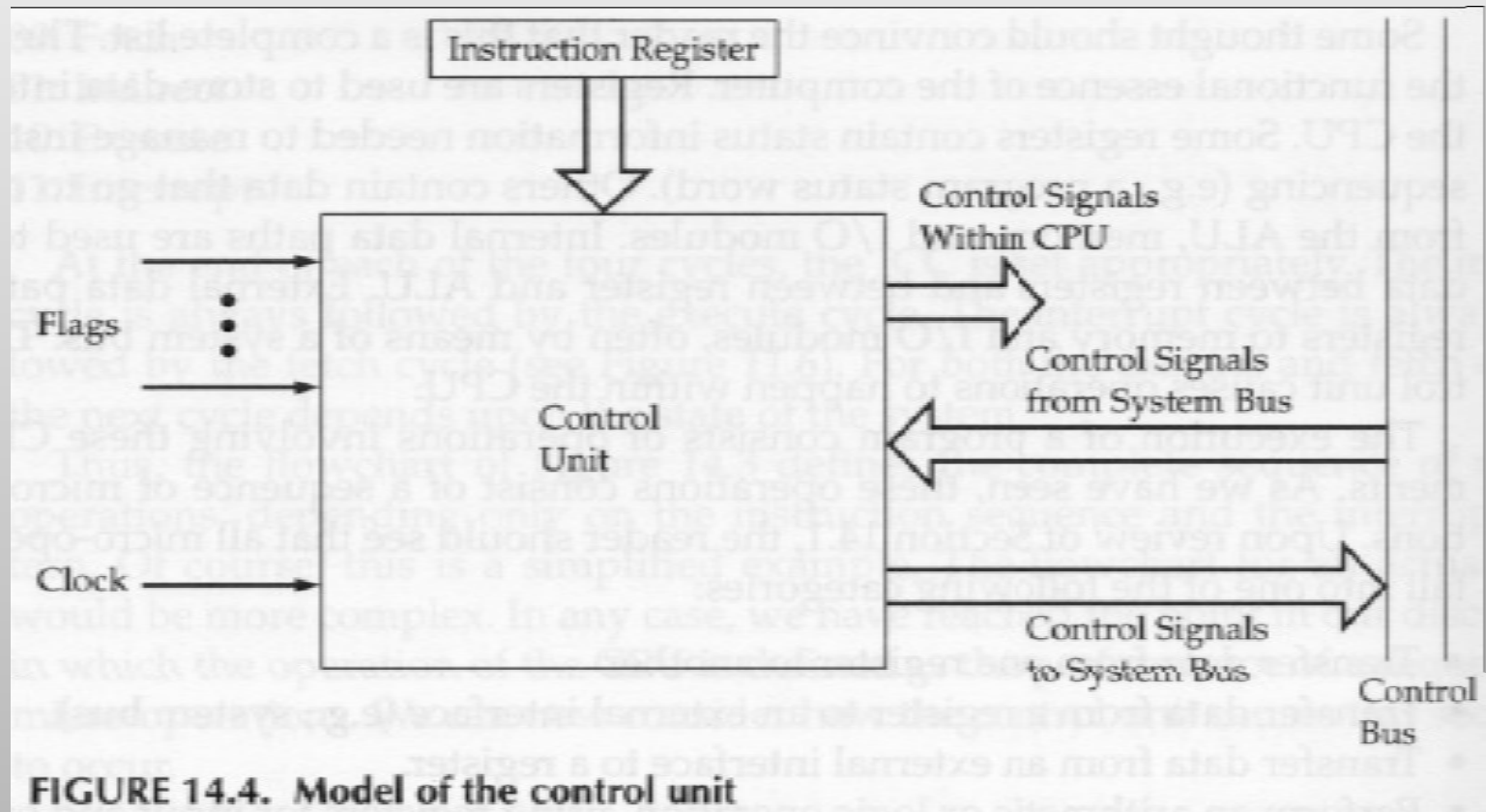


Sinais de Controle

- o Para além das partes constituintes do CPU, para que a unidade de controle possa executar as suas funções, é necessário ter *inputs*, que permitam avaliar o estado do sistema e vários *outputs* que permitam controlar o comportamento do sistema.

0 0 0

Sinais de Controle





Sinais de Controle - inputs

- Os vários *inputs* são:
- Relógio**
 - A unidade de controle executa uma micro-operação em cada pulso de relógio.
 - É muitas vezes referido como o ciclo de tempo do processador ou o ciclo de tempo do relógio;
- Instruction Register**
 - O *OPCODE* da atual instrução é usado para determinar qual a microoperação que vai ser realizada durante um ciclo de execução

0 0 0

Sinais de Controle - inputs

- o **Flags**

- o São necessárias para determinar o estado atual da CPU e o estado atual das operações da ALU;

- o **Sinais de Controle de bus de controle**

- o O bus de controle do bus do sistema, envia sinais para a unidade de controle, p.e. sinais de interrupção e acknowledge.



Sinais de Controle - output

- o Sinais de controle com a CPU
- o Existem 2 tipos, uns que causam movimentação de dados de um registrador para outro e outros que ativam funções específicas da ALU;
- o Sinais de controle para o bus de controle
- o Existem também 2 tipos: sinais de controle à memória e sinais de controle aos módulos de I/O



Exemplo - ciclo de busca (1)

A sequência de sinais enviados simultaneamente no caso do ciclo de aquisição é a seguinte:

- Um sinal de controle abre a porta colocando o conteúdo do MAR no bus de endereços
- É feita uma leitura dos sinais de controle do bus de controle por parte da memória
- Os sinais de controle abrem as portas lógicas, permitindo que o conteúdo do bus de dados possa ser arquivado no MBR
- Os sinais de controle lógicos adicionam 1 ao conteúdo do PC e guardam de novo o resultado do PC



Exemplo - ciclo de busca (2)

- Após esta sequência, a unidade de controle envia sinais de controle que abrem as portas entre o MBR e o IR.
- Para decidir o ciclo seguinte (indireto ou de execução) a UC examina o IR para ver se é necessário fazer uma referência indireta à memória.



Organização Interna

- o A ALU e todos os registradores do CPU estão ligados a um simples bus interno.
- o Portas e sinais de controle proporcionam movimento de dados para, e do bus para cada um dos registradores.
- o Sinais de controle adicionais controlam de, e para o bus de sistema (externo) as operações da ALU. Dois novos registradores, Z e Y são adicionados.
- o Estes registradores são necessários à organização da ALU.



Formas de Implementação

- o Até agora temos visto a unidade de controle no ponto de vista dos seus *inputs*, *outputs* e funções.
- o Vamos analisar como é feita a sua implementação.
- o Existem essencialmente duas categorias:
 1. Implementações por Hardware
 2. Implementações Microprogramadas



Implementação por Hardware

- o No caso das implementações por Hardware a UC é essencialmente um circuito em que os seus sinais lógicos são transformados no conjunto de sinais lógicos na saída que são os sinais de controle.
- o Entradas da Unidade de controle
- o Já vimos que os *inputs* da UC podem ser: os registradores, o relógio, as *flags* e os sinais do bus de controle.
- o Através do IR, a UC utiliza o *opcode* e vai executar várias operações para as diferentes instruções.

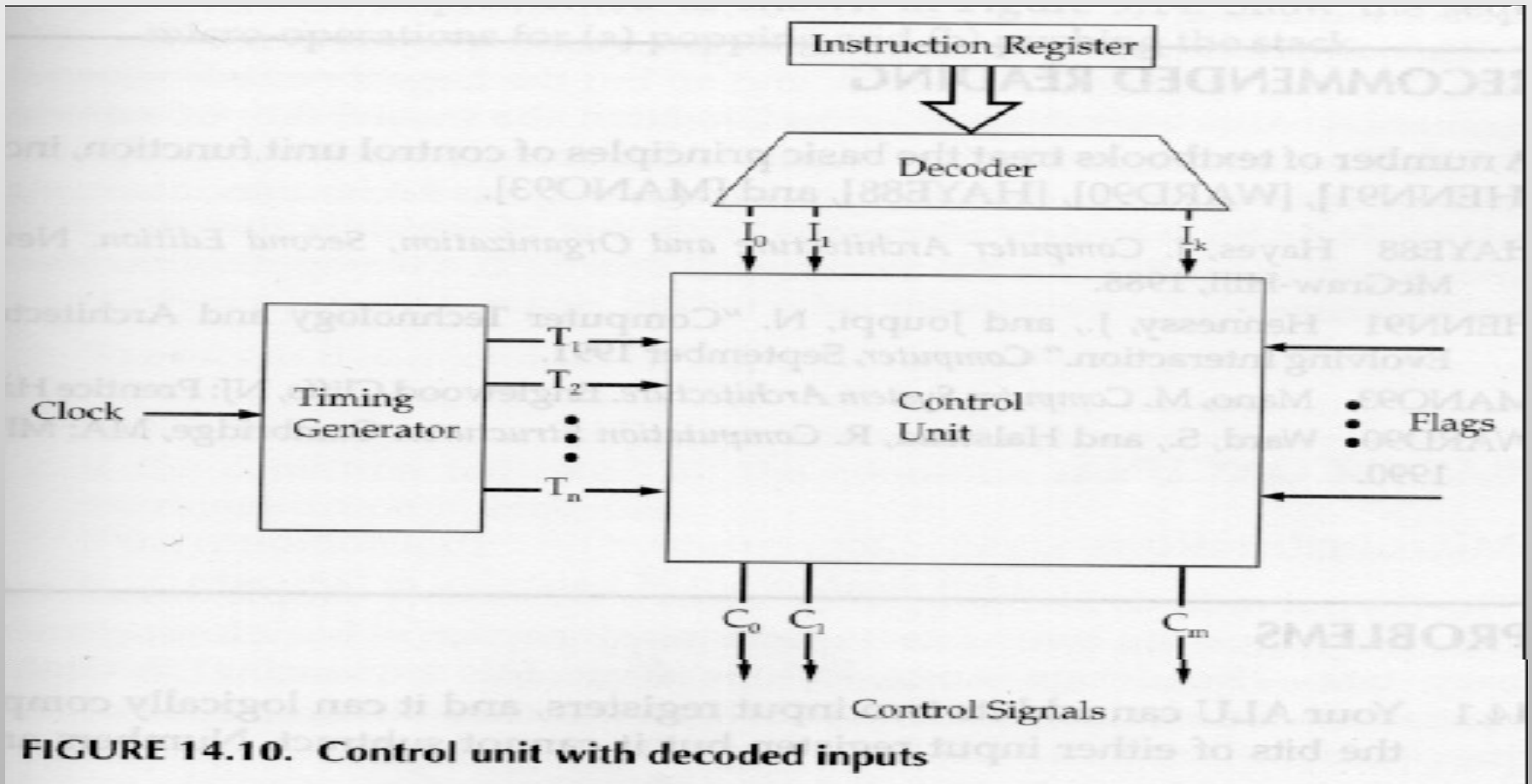


Implementação por Hardware

- Para simplificar a lógica da UC terá de haver uma única entrada lógica para cada *opcode*.
- Esta operação pode ser executada por um decodificador que recebe um *input* codificado e produz um simples *output*.
- A contribuição do relógio proporciona uma sequência de pulsos.
- São importantes para a medição da duração das micro-operações.
- O período de pulso de relógio terá de ser suficiente para permitir a propagação dos caminhos de dados (*Data path*) ao longo dos circuitos da CPU.

0 0 0

Implementação por Hardware





Problemas - Implement. por HW

- Lógica complexa de sequenciamento e micro-operação
- Difícil de projetar e testar
- Inflexível
- Difícil de adicionar novas instruções
- A solução para isto é a utilização da microprogramação para a implementação da UC.



Implementação Microprogramada

- Na utilização de sinais de controle, cada microoperação é descrita numa notação simbólica.
- A notação assemelha-se a uma linguagem de programação.
- Cada linha descreve um conjunto de micro-operações a correr em cada unidade de tempo denominada de microinstrução.
- A sequência de microinstruções é conhecida como microprograma ou firmware



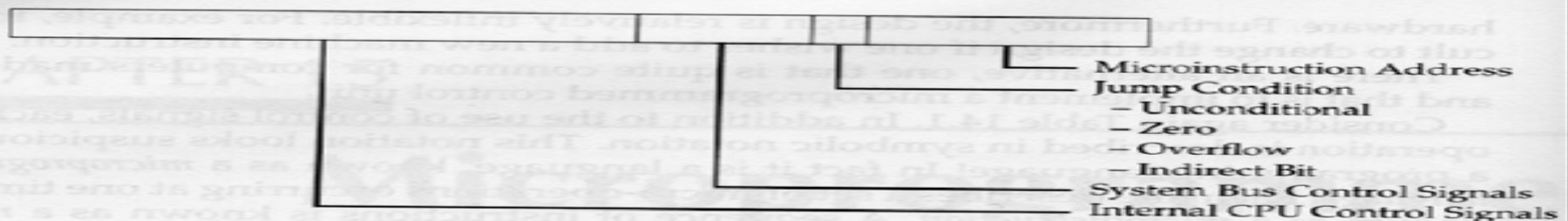
Implementação Microprogramada

- Se olharmos para uma microinstrução, cada linha de controle reflete-se num sinal on ou off por parte da unidade de controle.
- Esta condição pode obviamente ser representada por um dígito binário para cada linha de controle.
- Sendo assim podemos construir uma palavra de controle, em que cada uma representa uma linha de controle.
- Mais, cada microinstrução poderia também ser representada por uma sequência diferente de 1s e 0s na palavra de controle.

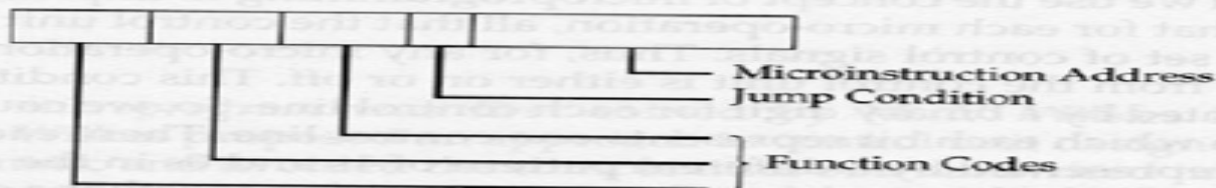
0 0 0

Implementação Microprogramada

- O conjunto das microinstruções (Fig15.1a) é denominado por microinstruções horizontais.



(a) Horizontal Microinstruction



(b) Vertical Microinstruction

FIGURE 15.1. Typical microinstruction formats



Implementação Microprogramada

- o Cada microinstrução é constituída por um bit, por cada linha interna de controle do CPU e por um outro bit por cada linha do controle do bus de sistema.
- o Existe ainda um campo de condição, indicando a condição sob a qual deverá haver rotinas de salto e um campo com o endereço da microinstrução a ser executada quando esse salto ocorrer.



Implementação Microprogramada

Uma microinstrução pode ser executada do seguinte modo:

- Para executar uma microinstrução, ativar todas as linhas de controle indicadas por um 1; ignorar todas as linhas de controle indicadas por 0. Os sinais de controle resultantes irão causar uma ou mais microinstruções a serem executadas
- Se a condição indicada pelos bits de condição for falsa, a próxima microinstrução é executada
- Se a condição indicada pelos bits de condição for verdadeira, a próxima microinstrução a ser executada esta indicada no campo de endereço.

0 0 0

Implementação Microprogramada

- o A memória de controle expressa na figura seguinte contém um programa que descreve o comportamento da unidade de controle.
- o Isto significa que poderíamos implementar uma unidade de controle simplesmente executando esse programa.

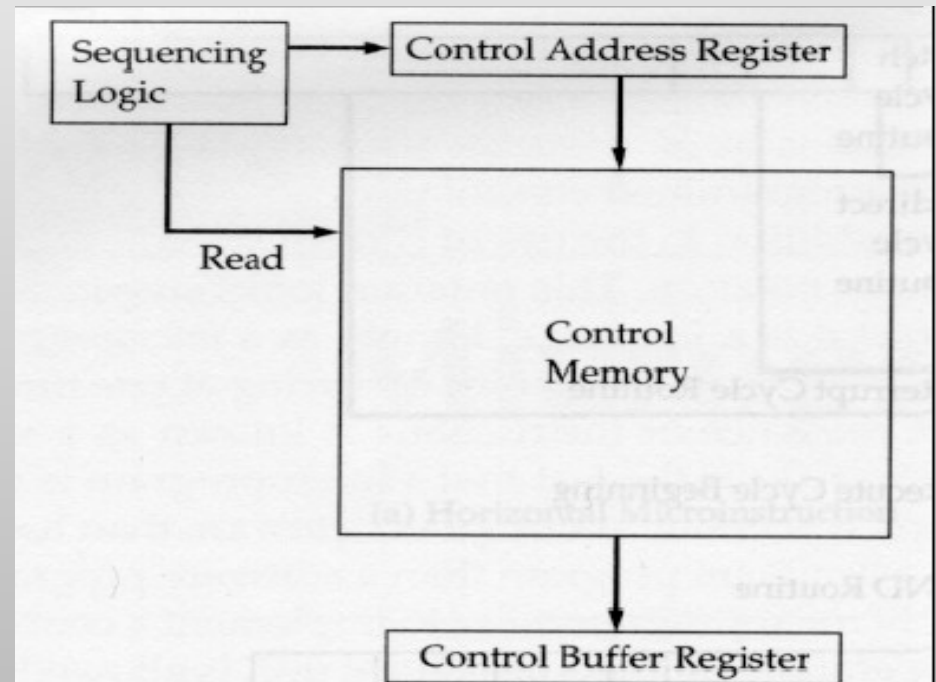


FIGURE 15.3. Control unit microarchitecture



Implementação Microprogramada

- O conjunto de microinstruções é guardado na memória de controle.
- O registrador de controle de acesso contém o endereço da próxima microinstrução a ser lida.
- Uma vez lida essa microinstrução, a mesma é transferida para o registrador de controle do *buffer*. A parte mais significativa deste registrador liga-se à unidade de controle.
- Sendo assim, ler a microinstrução da memória de controle é o mesmo que executar essa microinstrução!
- O terceiro elemento indicado na figura é a unidade de sequenciamento que carrega em memória o registrador de controle de endereço e detém um comando de leitura .



Implementação Microprogramada

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 ALU.



Implementação Microprogramada

- Tudo isto acontece durante um pulso de relógio.
- 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 ALU e do registrador de controle do *buffer*, uma das três decisões é tomada:



Implementação Microprogramada

1. Adquirir a próxima instrução: adicionar 1 ao registrador de controle de endereço;
2. Saltar para uma nova rotina baseada numa microinstrução de salto: Carregar o campo de endereço do registrador de controle do *buffer*, para o registrador de controle de endereço;
3. Saltar para a próxima instrução máquina: carregar para o registrador de controle de endereço baseado no *opcode* do IR.



Implementação Microprogramada

- o O decodificador identificado na figura 15.4, traduz o *opcode* do IR para um endereço de controle de memória.
- o O decodificador de baixo não é usado nas microinstruções horizontais mas é usado nas microinstruções verticais.

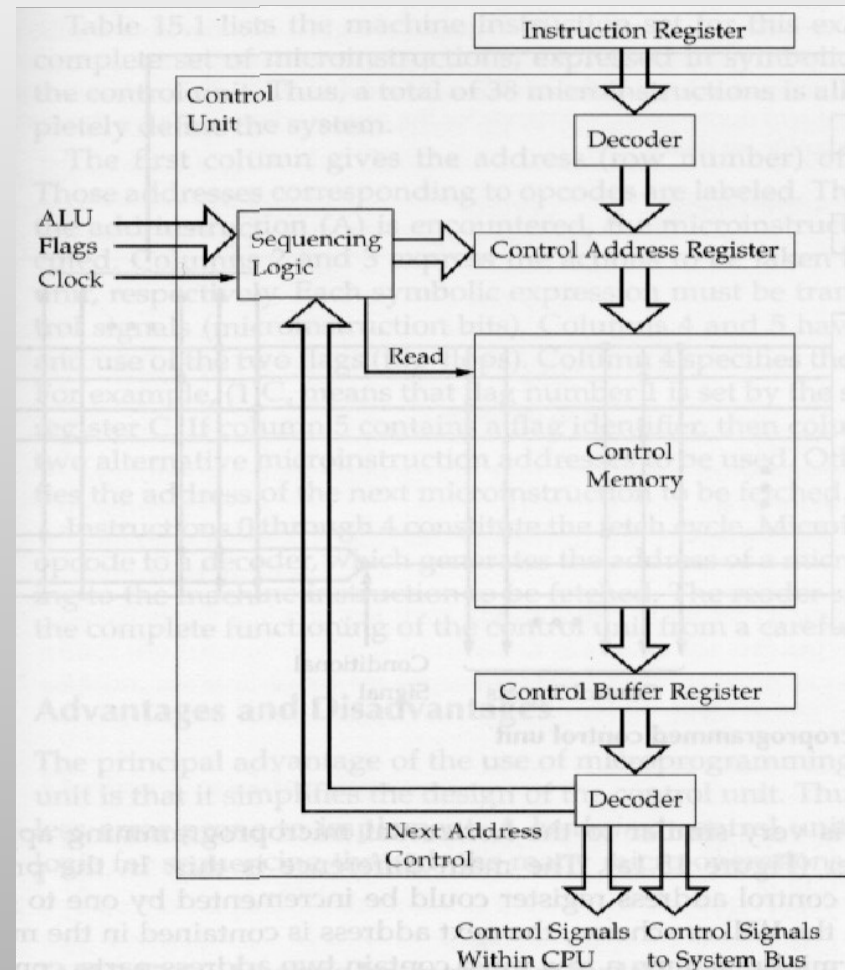


FIGURE 15.4. Functioning of microprogrammed control unit



Implementação Microprogramada

- o Numa microinstrução horizontal cada bit no campo de controle tem uma correspondência a uma linha de controle.
- o No caso de uma microinstrução vertical, é usado um código por cada ação realizada, p.ex $MAR \leftarrow (PC)$, e o decodificador traduz este código num sinal de controle individual.
- o A vantagem das microinstruções verticais é que são mais compactas (usam menos bits) com a consequente diminuição de complexidade lógica e de atrasos envolvidos.



Implementação Microprogramada - Vantagens

- o A principal vantagem do uso da micro programação para a implementação de uma UC é que isso simplifica o seu projeto.
- o Este torna-se mais barato e menos sujeito a erros.
- o A implementação baseada em hardware tem de incluir lógica complexa e um elevado número de micro-operações do ciclo de instrução.



Implementação Microprogramada - Desvantagens

- Por outro lado os decodificadores e as unidade lógicas de sequenciamento baseiam-se em lógica muito simplificada.
- A principal desvantagem das unidades de microprogramação são por vezes muito mais lentas que as UC baseadas em hardware, comparando em termos tecnológicos.
- Apesar disso, a microprogramação é a técnica dominante para a implementação de UC nos computadores contemporâneos.