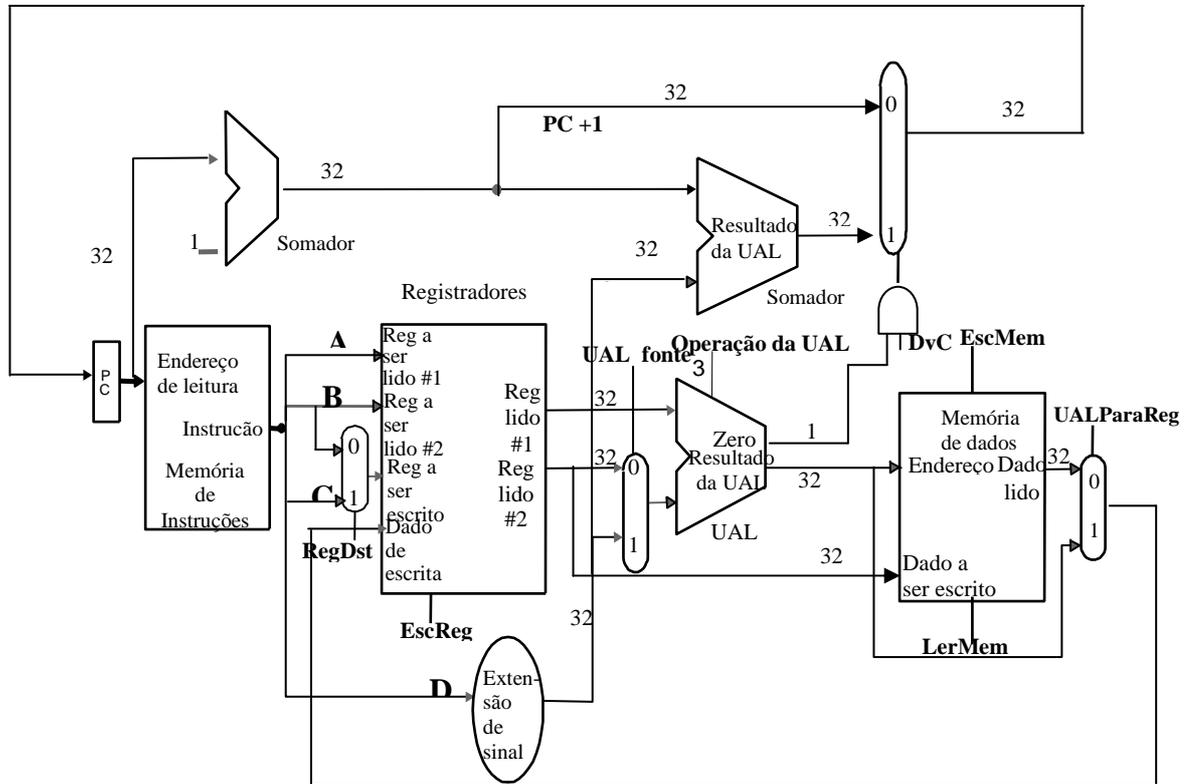


**UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

TCC04.070-Organização de Computadores I – Turma :A1 – Lista 3

1. A figura abaixo mostra o diagrama da implementação para a UCP estudada em sala.



a) Indique quais os bits da instrução devem estar conectados nas ligações A, B, C e D.

Resposta:

Consultando a tabela de instruções temos:

- A: bits 21 a 19
- B: 18 a 16
- C: 2 a 0
- D: 15 a 0

a) Considere a instrução `lw regA regB regC`, que utiliza o mesmo formato que a instrução `add`. Esta instrução, quando executada, deve armazenar no registrador `regC` o conteúdo de memória cujo endereço é a soma do conteúdo do registrador `regA` com o conteúdo do registrador `regB`. Complete a tabela com o valor que os sinais de controle devem ter (0, 1 ou X) para esta instrução, explicando suas escolhas:

Instr.	RegDst	EscReg	EscMem	LerMem	UAL fonte	DvC	UALParaReg
<code>lw regA regB regC</code>	1	1	0	1	0	0	0

Resposta:

- RegDst tem que ser igual a 1 porque os bits que indicam o registrador a ser escrito (`regC`) são os bits 2 a 0.
- EscReg tem que ser 1 porque deve haver escrita no registrador `regC`.
- EscMem tem que ser 0 porque a instrução não escreve na memória de dados.
- LerMem tem que ser 1 porque a instrução lê da memória de dados.

UAL fonte deve ser 0 porque o valor a ser somado ao conteúdo de regA deve ser o conteúdo do regB.

DvC deve ser 0 porque a próxima instrução a ser executada deve ser aquela que se encontra no endereço PC+1.

UAL Para Reg deve ser 0 porque o valor a ser carregado no registrador a ser escrito regC deve ser a saída da memória de dados.

2. Considere o esquema de pipeline mostrado para a arquitetura vista em sala de aula, onde temos 5 estágios de pipeline: Busca (B), Decodificação/Leitura de regs (DL), Execução da instrução (EX), Acesso à memória de dados (MD), Escrita nos registradores (ER). Analise o trecho de código abaixo e indique as instruções onde poderá existir conflito estrutural, ou de controle ou de dados, quando este código for executado. Suponha que a única maneira de resolver os conflitos é atrasar a execução dos estágios através da inserção de instruções noops. Indique o número de instruções noops que deverão ser inseridas para resolver cada conflito.

```

addi 3 3 1
lw 3 4 0
add 4 2 2
add 5 6 5
addi 7 7 1
addi 3 3 1
beq 2 5 if
addi 1 3 1
if addi 1 1 1
halt

```

Resposta:

```

addi 3 3 1 /* conflito de dados porque lw posterior precisa
do conteúdo do registrador 3 */
lw 3 4 0 /* conflito de dados porque add posterior
precisa do conteúdo do registrador 4 */

add 4 2 2
add 5 6 5
addi 7 7 1
addi 3 3 1
beq 2 5 if /* conflito de dados e de controle */
addi 1 3 1
if addi 1 1 1
halt

```

Inserindo noops para remover os conflitos:

```

addi 3 3 1 B DL EX MD ER
lw 3 4 0 B DL EX MD ER - 3 instruções noops
add 4 2 2 B DL EX MD ER - 3 instruções noops
add 5 6 5 B DL EX MD ER
addi 7 7 1 B DL EX MD ER
addi 3 3 1 B DL EX MD ER - 1 instrução noop
beq 2 5 if B DL EX MD ER
addi 1 3 1 B DL EX MD ER-2
addi 1 1 1 B DL EX MD ER
halt B DL EX MD ER

```

O código deve ficar com mostrado abaixo:

```

addi 3 3 1
noop
noop
noop
lw 3 4 0
noop
noop
noop
add 4 2 2

```

```

add 5 6 5
addi 7 7 1
addi 3 3 1
noop
beq 1 4
noop
noop
addi 1 3 1
if    addi 1 1 1
halt

```

3. Explique o que são e como funcionam os processos de compilação, montagem e ligação.

Resposta:

Compilação:

Processo que consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de montagem ou em linguagem de máquina (programa objeto). Processo este realizado pelo compilador. A Análise feita pelo compilador consiste em 3 partes:

- A análise léxica onde o programa fonte é decomposto em seus elementos individuais (comandos, operadores, variáveis, etc), gerando erros se for encontrada alguma incorreção.
- A análise sintática onde são criadas as estruturas de cada comando e verificação de acordo com as regras gramaticais da linguagem.
- A análise semântica onde são verificadas as regras semânticas estáticas, podendo produzir mensagens de erros.

Montagem:

Processo que consiste em traduzir um programa em linguagem de montagem (assembly) para seu equivalente em binário. Processo este realizado pelo montador. Esta tradução consiste em substituir os códigos de operação simbólicos por valores numéricos, nomes simbólicos de endereços por valores numéricos e converter valores de constantes para valores binários.

Ligação:

Processo onde é feita a identificação de chamadas a rotinas e respectiva conexão entre o código objeto principal e o código das rotinas. Este processo é executado pelo ligador. O ligador examina o código-objeto, procura referências externas não resolvidas e suas localizações nas bibliotecas substituindo a linha de chamada pelo código da rotina e emitindo mensagem de erro em caso de não encontrar a rotina.

4. Considere os seguintes tipos de interface de E/S: por programa, por interrupção e por acesso direto à memória.
- a) Descreva, em termos gerais, a operação de cada uma delas.

Resposta:

E/S por programa: O processador tem controle direto sobre a operação de E/S, incluindo a detecção do estado do dispositivo, o envio de comandos de leitura ou escrita e transferência de dados. Para realizar uma transferência de dados, o processador envia um comando para o módulo de E/S e fica monitorando o módulo para identificar o momento em que a transferência pode ser realizada. Após detectar que o módulo está pronto, a transferência de dados é realizada através do envio de comandos de leitura ou escrita em registradores do controlador do dispositivo de E/S pelo processador. Se o processador for mais rápido que o módulo de E/S, o procedimento realizado para identificar quando o módulo está pronto representa um desperdício de tempo de processamento.

E/S por interrupção: Neste caso, o processador envia um comando para o módulo de E/S e continua a executar outras instruções, sendo interrompido pelo módulo quando ele estiver pronto para realizar a transferência de dados, que é executada pelo processador através da obtenção dos dados da memória principal, em uma operação de saída, e por armazenar dados na memória principal, em uma operação de entrada.

E/S por acesso direto à memória (DMA): Nesse caso a transferência de dados entre o módulo de E/S e a memória principal é feita diretamente sem envolver o processador. Existe um outro módulo denominado controlador de DMA que realiza a transferência direta de dados entre a memória e o módulo de E/S. Quando o processador deseja efetuar a transferência de um bloco de dados com um módulo de E/S, ele envia um comando para o controlador de DMA indicando o tipo de operação a ser realizada (leitura ou escrita de dados), endereço do módulo de E/S envolvido, endereço de memória para início da operação de leitura ou escrita de dados e número

de palavras a serem lidas ou escritas. Depois de enviar estas informações ao controlador de DMA, o processador pode continuar executando outras instruções. O controlador de DMA executa a transferência de todo o bloco de dados e ao final envia um sinal de interrupção ao processador, indicando que a transferência foi realizada.

- b) Considere um sistema onde o número de ciclos de relógio para realizar uma operação por programa é igual a 400, o processador utiliza um relógio de 500 MHz para executar as instruções e nenhuma transferência de dados pode ser perdida. Determine o overhead -em termos de fração de tempo de CPU consumida- que ocorre quando se utiliza a interface por programa para os seguintes dispositivos:
- b.1) Um mouse que deve ser interrogado pelo sistema 30 vezes por segundo para garantir que nenhum movimento dele seja perdido.

Resposta:

Como o mouse deve ser interrogado 30 vezes por segundo e em cada atendimento são gastos 400 ciclos, teremos 12000 ciclos sendo gastos em um segundo para realizar operações de E/S por programa. A frequência do relógio desta máquina é 500 MHz, ou seja, 500×10^6 ciclos de relógio são realizados em um segundo. O overhead é calculado como a razão do número de ciclos utilizados em um segundo pelo método de E/S por programa dividido pelo número total de ciclos que são fornecidos em um segundo $12000/(500 \times 10^6)=0,0024 \%$.

b.2) Um disco flexível que transfere dados para o processador em blocos de 16 bits e possui uma taxa de transferência de dados de 50 KB/segundo.

Resposta:

Como a taxa de transferência é igual 50 KB/s e a unidade de transferência é 16 bits, ou seja, 2 bytes, temos que a taxa de atendimento deve ser igual a $50 \text{ KB}/2\text{B}=25 \text{ K}$ vezes por segundo. Então serão gastos $25\text{K} \times 400$ ciclos de relógio em um segundo para realizar operações de E/S por programa. O "overhead" é igual a $25 \times 10^3 \times 400/(500 \times 10^6)=2\%$

b.3) Um disco rígido que transfere dados para o processador em blocos de 16 bytes e possui uma taxa de transferência de 4MB/segundo.

Resposta:

Como a taxa de transferência é igual 4 MB/s e a unidade de transferência é 16 bytes, temos que a taxa de atendimento deve ser igual a $4 \text{ MB}/16\text{B}=250 \text{ K}$ vezes por segundo. Então serão gastos $250\text{K} \times 400$ ciclos de relógio em um segundo para realizar operações de E/S por programa. O "overhead" é igual a $250 \times 10^3 \times 400/(500 \times 10^6)=20\%$

- c) Discuta a vantagem que a interface por interrupção possui em relação à interface por programa. Ilustre sua resposta calculando a fração de tempo de CPU consumida pelo disco rígido conforme descrito no item b.3 assumindo que o overhead de cada transferência, incluindo a interrupção, é 500 ciclos de relógio e que o disco está ativo em 5 % do tempo total em que a CPU está sendo utilizada. Porque a porcentagem do tempo que um dispositivo está ativo é importante para comparar as interfaces por programa e por interrupção?

Resposta:

Como a taxa de transferência do disco rígido é igual 4 MB/s e a unidade de transferência é 16 bytes, temos que a taxa de atendimento deve ser igual a $4\text{MB}/16\text{B}=250 \text{ K}$ vezes por segundo. Então serão gastos $250\text{K} \times 500$ ciclos de relógio em um segundo para realizar operações de E/S por interrupção. Caso o dispositivo estivesse ativo todo o tempo, o "overhead" seria igual a $250 \times 10^3 \times 500 / (500 \times 10^6)=25\%$ que é maior que o "overhead" da E/S por programa. Mas como o dispositivo só está ativo 5% do tempo total, neste caso os ciclos de relógio só serão utilizados quando o dispositivo interromper o processador, ou seja, o "overhead" será $25\% \times 5\% = 1,25\%$.

A porcentagem do tempo que um dispositivo está ativo é importante para comparar estes dois métodos, porque no método por E/S por programa o processador é obrigado a gastar ciclos de relógio para monitorar o dispositivo independentemente do fato de ele estar sendo acionado ou não. No caso do método por interrupção, os ciclos só serão gastos quando o dispositivo estiver efetivamente sendo acionado para realizar transferência de dados.

Neste caso, observe que na operação por programa, de cada 100 tentativas em que o processador acessa o dispositivo apenas em 5 este estará disponível para transferir informações (5% ativo). Na operação de interrupção, o processador apenas será interrompido quando o dispositivo de E/S estiver em atividade.

- d) Considere agora outro cenário, onde o disco rígido está sendo controlado por um controlador de DMA, cada transferência entre o disco e a memória ocorre em blocos de 8KB e ocorrem transferências em 100 % do tempo total que a CPU está sendo utilizada. Calcule a fração de tempo de CPU que é consumida, caso necessite-se, em cada transferência, de 1000 ciclos de relógio do processador para inicializar o controlador de DMA e o tratamento da interrupção gerada pela finalização da operação do controlador do DMA necessite de 500 ciclos de relógio do processador. Ignore qualquer impacto que possa ocorrer devido à contenção do barramento entre o processador e o controlador de DMA.

Resposta:

O número de ciclos de inicialização é igual a 1000 e de interrupção é igual a 500. Logo o número total de ciclos por operação de DMA é igual a 1500. Como descrito no item b.3, o disco possui uma taxa de transferência de 6 MB/s. Neste caso, são transferidos 8 KB por operação de transferência de dados, então cada operação de DMA terá a duração de $8KB/4 MB/s = 2 \times 10^{-3}$ segundos. Se as transferências ocorrem 100 % do tempo, teremos que o número de ciclos gasto para transferência de dados será $1500 / (2 \times 10^{-3}) = 750 \times 10^3$ ciclos de clock/s. O “overhead” será $(750 \times 10^3) / (500 \times 10^6) = 0,15 \%$

5. Descreva cinco principais características da arquitetura RISC e compare-as com a arquitetura CISC.

Resposta:

RISC	CISC
Pequeno número de instruções de linguagem de máquina	Grande número de instruções de linguagem de máquina
Instruções com tempos de execução semelhantes e pequenos	Podem existir instruções com tempo de execução muito longo
Todas as instruções possuem um único tamanho	Instruções possuem tamanhos variáveis
Operações lógicas e aritméticas são aplicadas a operandos que estejam somente nos registradores	Operações lógicas e aritméticas podem ser aplicadas a operandos que estejam em registradores e na memória
Passagem de parâmetros para procedimentos é realizado preferencialmente através de registradores	Passagem de parâmetros é realizada através da memória