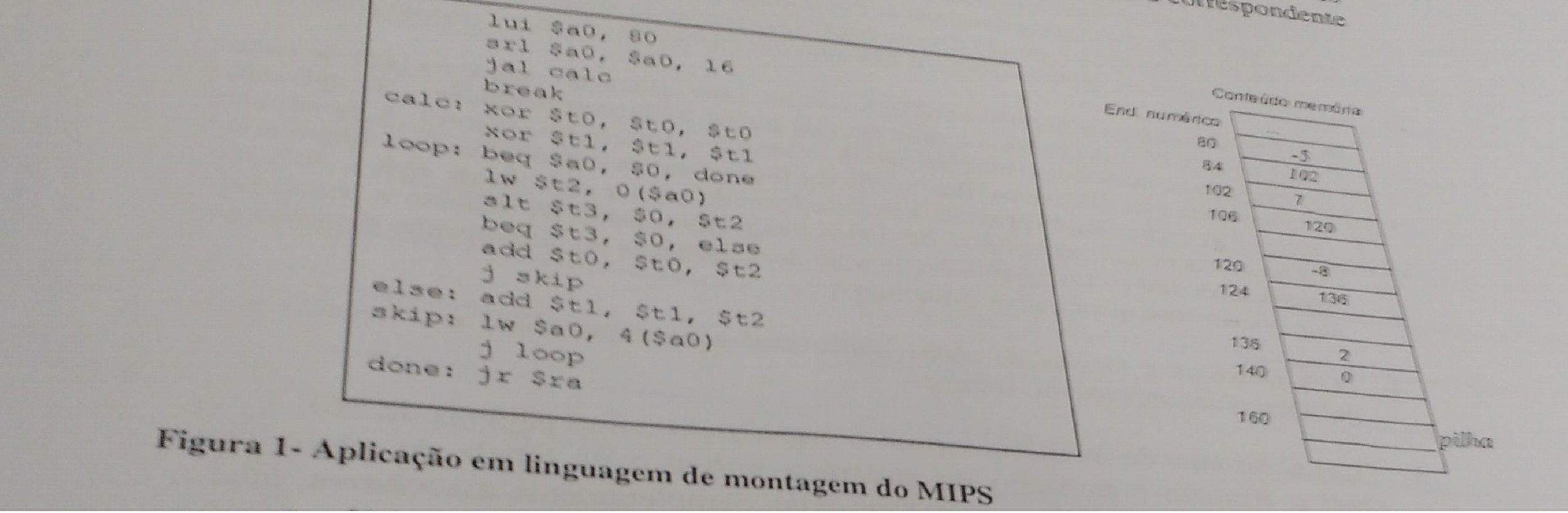
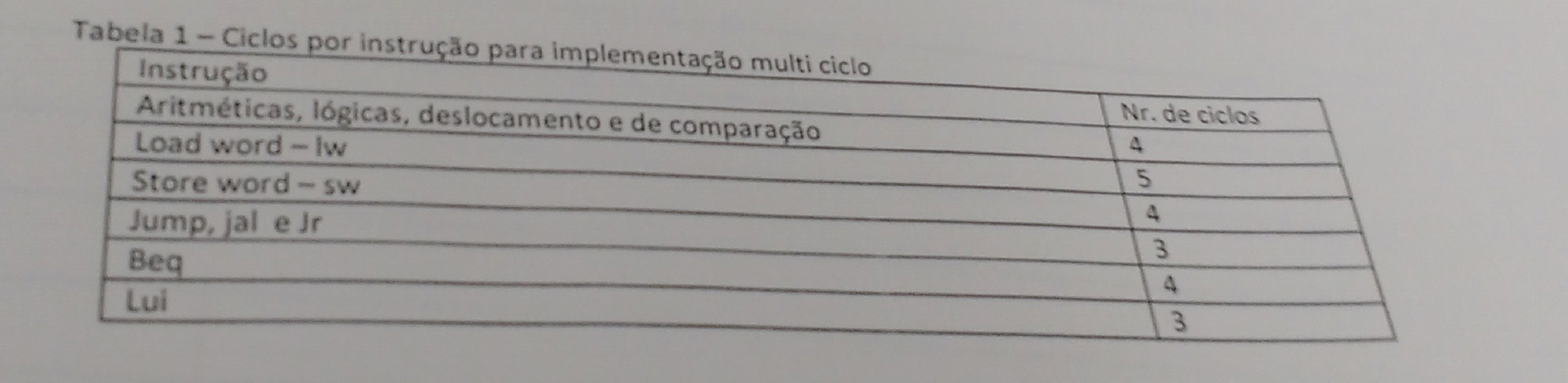
FINAL - 2017.1 - Infra-estrutura de Hardware

⦁ Um sistema computacional com clock 2,2 Ghz baseado no processador MIPS.Para definir este processador foram realizadas implementações multiciclo e pipeline. Para avaliação das implementações mencionadas considere o programa descrito abaixo. O registrador $29 possui o valor 160. A localização das variáveis na memória podem ser visualizada na figura 1, a qual contém os valores numéricos e simbólicos de cada endereço bem como o correspondente de conteúdo de memória.



a - [1,0] Considere que o programa da figura 1 está sendo executado em uma implementação multiciclo da CPU, onde o número de ciclos de cada instrução é dado pela tabela abaixo. Calcule o CPI para executar o programa na implementação multiciclo.



R - Em uma implementação multiciclo, o cálculo do CPI é feito por:

*CPI = ⅀ instrução\*ciclos/ Número de Instruções*

E como o loop vai de 80 -> 136 de 4 em 4, ele se repete 14 vezes. Dentro do loop, a operaçao de beq só será desviado em 2 casos(-5 e -8), em todos os outros, o desvio nao é executado. Fazendo os calculos, temos

1\*lui + 3\*aritmeticas + 2\*jumps + 2\*loop(2\*beq + 2\*lw + 2\*jump + 2\*aritmeticas) + 2\*(2\*beq + 2\*lw + 1\*jump + 2\*aritmeticas) + 1\*beq = 116 ciclos.

CPI = 147/37 = 4,0

R2-

Em uma implementação multiciclo, o cálculo do CPI é feito por:

*CPI = ⅀ instrução\*ciclos/ Número de Instruções*

O programa executa as instruções na seguinte ordem: LUI, SRL, JAL, XOR, XOR, BEQ, LW, SLT, BEQ, ADD, LW, J, BEQ, LW, SLT, BEQ, ADD, J, LW, J, BEQ, LW, SLT, BEQ, ADD, LW, J, BEQ, LW, SLT, BEQ, ADD, J, LW, J, BEQ, JR.

Ao todo são 37 instruções.

1 - LUI

11 - Aritméticas, lógicas e de comparação

8 - J/JAL/JR

9 - BEQ

8 - LW

b- [1,5] Considere agora o programa da figura 1 está sendo executado em uma implementação pipeline de 5 estágios. Os conflitos de dados e de controle devem ser resolvidos com a inserção de NOPs, cálculo o nome CPI e o speed up obtido. O desempenho melhorou? Justifique sua resposta.

R - Para resolver os conflitos de dados, temos 2 nops após o lui, 2 nops no primeiro lw e 2 nops no slt(Os do lw e slt fazem parte do loop). Já os conflitos de controle sao necessários 3 nops para saber do desvio, logo temos os nops nas instruçoes: jal, beq1, beq2 , j1 e j2(Os 4 ultimos estao dentro do loop).

Para fazer o calculo do CPI num processador pipeline, basta calcular a quantidade de instruçoes + nops + tempo da ultima instruçao acabar/Numero de instruçoes

CPI = 37 + 5nops + 1\*16nops 2\*13 + 3 /37 = 2,6.

c- [1,5] Considere agora uma implementação em pipeline, onde os conflitos de dados foram resolvidos com adiantamento e detecção de conflitos e os de controle com antecipação do desvio, qual o novo CPI para esta implementação? Justifique o CPI obtido.

R - Com o adiantamento, nao se faz mais necessaria a inserçao de nops nos casos de conflitos de dados, que sao 6 nops. Porém para os casos de conflitos de controle, com a antecipaçao dos desvio terá sera necessario apenas 1 ciclo para corrigir caso o desvio ocorra. Entao o novo CPI será:

CPI = 37 + 1\*4 + 2\*3 + 3/37 = 1,37.

d- [1,0] Para explorar paralelismo de thread, os processadores atuais possuem mais de um núcleo (multi-core). Qual o principal desafio destes processadores com relação ao custo de comunicação por memória compartilhada e quais as técnicas para se resolver estes desafios?

R - O principal desafio dos processadores multi-core é manter a coerência de cache, ou seja, fazer com que as caches dos diversos processadores estejam operando de acordo com as alterações que foram feitas nos outros processadores. As técnicas para resolver esse desafio são:

Snooping (Ideal para o modelo centralizado de memória) - Os controladores de cache ficam supervisionando o barramento até que algo relevante ocorra, daí ela manda um sinal invalidando ou atualizando as demais cópias. Se for uma cache Write-through ele vai buscar na memória e invalidar, caso seja Write-back a cache recebe o valor válido.

Diretório (Ideal para o modelo descentralizado de memória) - Cada cache tem um diretório responsável por um determinado número de dados. Toda vez que a cache for requisitar um dado é necessário pedir ao diretório responsável, que vai gravar quem tem cópias de que dado para poder gerenciar.

2 - [1,5] O processador possui uma espaço de endereçamento de 4096 G bytes, organizado como páginas de 8K bytes. A memória principal possui 1/512 da capacidade de endereçamento virtual e é endereçado por bytes. O sistema ainda possui uma cache de instruções cuja capacidade de armazenar informações corresponde a 1/512 da capacidade da memória principal. Esta cache é four-way associativa write - through. A cache de dados possui metade da capacidade e o mesmo grau de associatividade da cache de instruções, porém é write back. O tamanho do bloco em ambas as caches é de 16 palavras. Qual o layout e o tamanho das caches (instruções e dados) em bytes?

R - (n sei se tá certo)

Processador

Espaço de Endereçamento virtual - 4096 Gb ou

Espaço de Endereçamento físico - 4096/512 Gb = 8Gb =

Cache de instrução:

Tamanho - 8/512 Gb = 16 Mb = = bytes

Four-way associativa = = 22 bits (índice)

Write-through

Tamanho do bloco - 16 palavas = bytes = 4 bits (Off-set)

Layout do endereço da cache de instrução:

Tamanho do endereço da cache - 33 bits

Tag = 33 - 22 - 4 =

|  |  |  |
| --- | --- | --- |
| TAG  (5) | ÍNDICE  (22) | OFF-SET  (6) |

Layout da cache de instrução:

|  |  |  |
| --- | --- | --- |
| Bit de Validade  (1) | TAG  (5) | DADO |

Tamanho da cache de instrução em byte:

Cache de Dados:

Tamanho -

Four-way associativa =

Write-back

Tamanho do bloco - 16 palavras =

Layout do endereço de cache de dados:

Tamanho do endereço da cache - 33 bits

Tag = 33 - 25 = 8 bits

|  |  |  |
| --- | --- | --- |
| TAG  (6) | ÍNDICE  (21) | OFFSET  (6) |

Layout da cache de dados:

|  |  |  |  |
| --- | --- | --- | --- |
| Bit de Validade  (1) | Dirty Bit  (1) | TAG  (6) | DADO |

Tamanho da cache de dados:

3- [1,0] O sistema da questão anterior executou uma aplicação com uma taxa de cache hit igual a 95% para dados e uma taxa de hit de 97% para instruções. Na aplicação mencionada, 40% das instruções são de acesso à memória(load/store). A penalidade da memória é de 100 ciclos. Para melhorar o desempenho foi duplicado o tamanho do bloco e as caches duplicaram de tamanho levando a uma redução de 45% e 50% nas faltas da cache de instruções e de dados, respectivamente.No entanto a penalidade passou a ser 160 ciclos. Qual a implicação das mudanças no desempenho da máquina, faça uma análise quantitativa e qualitativa.

R -

Como 40% das instruções são de acesso à memória, apenas 40% estão sujeitas ao Miss Rate de 5% da cache de dados. No caso da cache de instruções os 100% estão sujeitos ao Miss Rate de 3%.

Após as alterações, o Miss Rate da cache de dados reduziu 50% e a da cache de dados reduziu 45%.

Logo, percebe-se uma melhora no CPI após as alterações.

4- [1,0] Descreva a técnica de cache multi nível e explique como esta técnica pode melhorar o desempenho do sistema?

R -

**Cache Multinível:** Adicionar mais camadas de cache, de forma a reduzir, no primeiro nível, o tempo de acesso, usando Early restart e Critical Word First e, no segundo nível, reduzir a taxa de **miss**, já que a cache de segundo nível é maior que a cache de primeiro nível (embora um pouco mais lenta).  
  
5- [1,5] O sistema de memória virtual é organizado como paginas de 16K bytes. Qual o formato e tamanho do endereço virtual e da tabela de paginas? O sistema possui uma TLB de 8192 entradas completamente associativa, qual o tamanho da TLB e quais as vantagens deste mecanismo?

R -

Memoria Virtual:

Espaço de Endereçamento =

Offset da Pagina =

Layout do Endereço Virtual

|  |  |
| --- | --- |
| Numero da Pagina Virtual | Offset da Pagina |
| 28 bits | 14 bits |

Tabela de Tradução:

Espaço de Endereçamento da Memoria = 2^33 bits.

Layout do endereço da Memoria:

|  |  |
| --- | --- |
| Pagina Fisica = 19 bits | Offset da Pagina = 14 bits |

Layout da Tabela de Paginas

|  |  |  |
| --- | --- | --- |
| Bit de Presença = 1bit | Dirty Bit = 1bit | Pagina Fisica = 19bits |

Tamanho da Tabela de Paginas = 2^28 \* 21 / 2^3 bytes

TLB:

Layout da TLB(Duvida, quais os bits que se usa na tlb? So o bit de presença ou existem outros?)

Tag = Pagina Virtual - Indice(Porem como é completamente associativa, nao tem indice)

|  |  |  |  |
| --- | --- | --- | --- |
| Bit de Presença = 1bit | Dirty bit = 1bit | Tag = 28bits | Pagina Fisica = 19bits |

Tamanho da TLB = 2^13\*48 / 2^3 bytes.

