

Nome do aluno:

$$\text{Nota das questões: } \cancel{101951} \times \cancel{110010} / 100 = \frac{295}{500} = 5,9$$

Primeira Avaliação - S72 - 24/06/2024 - Duração: 100 min

Solução à lápis ou caneta, em qualquer ordem, nas folhas à parte

Resolva apenas cinco das questões da prova! Deixe três de fora.

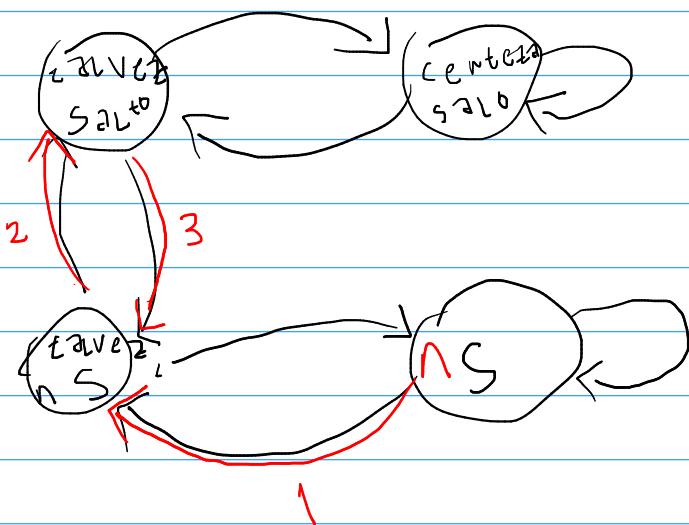
- [1] Suponha um preditor bimodal como visto em sala. Considere que determinado branch executou 3 vezes, resultando na sequência SALTO - SALTO - NÃO SALTO. Suponha que o preditor para este branch estava inicialmente em um estado que, considerando esta sequência de 3 instruções, era o pior estado inicial possível. Qual a sequência de previsões? Quantos acertos o preditor terá?
- [2] Dado processador superescalar possui uma CPI=0,6. Considerando que ele possui 9 unidades de execução, qual é o percentual médio de utilização destas UE's? (Ou seja, quantas UE's estão ocupadas executando instruções, ao invés de estarem ociosas, na média?).
- [3] Considere que um vetor v de 128 bytes deverá ter todos os seus dados zerados. Suponha que a opção (a) zera todos os bytes, um a um, caminhando de trás para a frente do vetor (ou seja, $v[127]$ em direção a $v[0]$) e a opção (b) usa um stride de passo 4 de cada vez (isso significa zerar primeiramente as posições 0, 4, 8, ..., 124, uma a uma, seguidas das posições 1, 5, 9, ..., 125 então 2, 6, 10, ..., 126 e finalmente 3, 7, 11, ..., 127). Pergunta um: qual das duas tem melhor localização espacotemporal de dados? Pergunta dois: haverá diferença de velocidade de execução entre (a) e (b) devido à cache?
- [4] Suponha a seguinte sequência de acessos a um endereço de memória x em um sistema SMP quadcore: (a) $\mu P1$ lê x, (b) $\mu P2$ lê x, (c) $\mu P1$ altera x, (d) $\mu P1$ lê x, (e) $\mu P2$ lê x. Considerando o protocolo MESI de coerência de cache, dê o estado das linhas cache associadas a x para o $\mu P1$ e para o $\mu P2$ em cada um dos 5 passos listados.
- [5] Rodamos um programa P num $\mu P1$ sem multithreading e também rodamos P num $\mu P2$ com multithreading temporal. Supondo as demais especificações de $\mu P1$ e $\mu P2$ idênticas, qual dos processadores deverá provavelmente indicar uma maior CPI? Justifique brevemente.
- [6] Imagine que um determinado processador está atolado por conta de inúmeros pedidos de interrupção que devem ser tratados por ele. Se dobrarmos a frequência de clock, será que isso pode resolver o problema? Responda se sim ou se não tem nada a ver, e justifique com uma frase bem curta.
- [7] Considere 3 modelos dos sistemas big.LITTLE da ARM, os 3 tendo demais especificações idênticas, sendo cada um com 4 núcleos big e 4 núcleos LITTLE. Qual deles deve apresentar o maior desempenho? Justifique sumariamente.
- [8] Rodamos um programa P num $\mu P1$ com $f_{clock1} = 1,5$ GHz e rodamos o mesmo binário do programa P num $\mu P2$ com $f_{clock2} = 1,0$ GHz. Verificamos que ele roda mais rápido no $\mu P2$. Sugira alguma razão plausível para este resultado aparentemente anti-intuitivo.

Boa Prova



[1] Suponha um preditor bimodal como visto em sala. Considere que determinado branch executou 3 vezes, resultando na sequência SALTO - SALTO - NÃO SALTO. Suponha que o preditor para este branch estava inicialmente em um estado que, considerando esta sequência de 3 instruções, era o pior estado inicial possível. Qual a sequência de previsões? Quantos acertos o preditor terá?

nenhum



[2] Dado processador superescalar possui uma CPI=0,6. Considerando que ele possui 9 unidades de execução, qual é o percentual médio de utilização destas UE's? (Ou seja, quantas UE's estão ocupadas executando instruções, ao invés de estarem ociosas, na média?).

$$CPI \rightarrow \text{clock} \cdot P_{\text{ot}} \cdot \text{instructions}$$

$$\frac{m}{s} \quad 0,6 \rightarrow C/I \quad \frac{1}{0,6} \rightarrow \frac{T}{C}$$

$$(1/0,6)/9 = \%$$

[3] Considere que um vetor v de 128 bytes deverá ter todos os seus dados zerados. Suponha que a opção (a) zera todos os bytes um a um, caminhando de trás para a frente do vetor (ou seja, de $v[127]$ em direção a $v[0]$) e a opção (b) usa um stride de passo 4 de cada vez (isso significa zerar primeiramente as posições 0, 4, 8, ..., 124, uma a uma, seguidas das posições 1, 5, 9, ..., 125 então 2, 6, 10, ..., 126 e finalmente 3, 7, 11, ..., 127). Pergunta um: qual das duas tem melhor localização espacotemporal de dados? Pergunta dois: haverá diferença de velocidade de execução entre (a) e (b) devida à cache?

[4] Suponha a seguinte sequência de acessos a um endereço de memória x em um sistema SMP quadcore: (a) $\mu P1$ lê x , (b) $\mu P2$ lê x , (c) $\mu P1$ altera x , (d) $\mu P1$ lê x , (e) $\mu P2$ lê x . Considerando o protocolo MESI de coerência de cache, dê o estado das linhas cache associadas a x para o $\mu P1$ e para o $\mu P2$ em cada um dos 5 passos listados.

[5] Rodamos um programa P num $\mu P1$ sem multithreading e também rodamos P num $\mu P2$ com multithreading temporal. Supondo as demais especificações de $\mu P1$ e $\mu P2$ idênticas, qual dos processadores deverá provavelmente indicar uma maior CPI? Justifique brevemente.

Nome do al:

Nota das questões: $75/25/50/40/\cancel{X}/\cancel{X}/\cancel{X}/15 = \frac{205}{500} = 4,1$

Primeira Avaliação - 572 - 22/04/2024 - Duração: 100 min

Solução à lápis ou caneta, em qualquer ordem, nas folhas à parte

Resolva apenas cinco das questões da prova! Deixe duas de fora.

- ✓ [1] O que é a ISA de um processador? Responda sucintamente.
- ✓ [2] Consulte as tabelas no verso para fazer as conversões abaixo. Quero hexadecimal, não binário.
[a] Qual é o valor hexadecimal de 32 bits do código de máquina correspondente à instrução sub s0,r1,s9 em assembly RISC-V?
[b] Qual é o valor hexadecimal de 32 bits do código de máquina correspondente à instrução addi s3,s3,-5 em assembly RISC-V?
- ✓ [3] Consulte as tabelas no verso para fazer as conversões abaixo. Se houver constante negativa, expresse como número negativo; se houver salto, identifique o endereço de memória destino em 32 bits (ex.: beq s1,s2,0x0040 1234), supondo que a instrução sendo executada está no endereço 0x0040 0008.
[a] Qual é a instrução completa em assembly RISC-V que é codificada para o código de máquina de 32 bits 0x00C06393?
[b] Qual é a instrução completa em assembly RISC-V que é codificada para o código de máquina de 32 bits 0x00938D33?
- ✓ [4] No circuito multiciclo do verso, se o sinal ALUSrcA ficar travado em 1 por algum bug, haverá um comportamento incorreto no processador. Qual é este comportamento?
- ✗ [5] Mostre o circuito decodificador que identifica se a instrução lida é uma instrução bne. Dito de outra forma, mostre a portona lógica que é parte da UC e que gera nível 1 se e somente se a instrução é bne.
- ✗ [6] Faça um programa assembly RISC-V que conta quantos números 0x0000 0000 estão presentes na faixa de memória entre 0x1001 0000 e 0x1002 0000 (exclusive). Considere as palavras de 32 bits alinhadas na memória.
- ✗ [7] Faça um programa assembly RISC-V que determina se a sequência de dois números 0x12345678 e em seguida 0x87654321 estão presentes em dois lugares subsequentes na faixa de memória entre 0x1001 0000 e 0x1002 0000 (exclusive). Considere as palavras de 32 bits alinhadas na memória.
- ✓ [8] Dado o programa abaixo, quantos clocks ele leva para ser executado no RISC-V com o pipeline visto em sala de aula? Considere que não há forwarding/bypass ou seja, utilizamos bolhas quando necessárias.

- A: add s2,zero,zero
B: addi s3,zero,65
C: addi s4,s2,31
D: add s2,s3,s3
E: add s3,s3,s3
F: add s2,s2,s2

N [1] O que é a ISA de um processador? Responda sucintamente.

Instruction

Set

Architecture

Conjunto de instruções, funcionamentos, registradores, etc.

resumindo é o comportamento do processador

N [2] Consulte as tabelas no verso para fazer as conversões abaixo. Quero hexadecimal, não binário.
[a] Qual é o valor hexadecimal de 32 bits do código de máquina correspondente à instrução sub \$0, \$1, \$9 em assembly RISC-V?
[b] Qual é o valor hexadecimal de 32 bits do código de máquina correspondente à instrução addi \$3, \$3, -5 em assembly RISC-V?

0100000 11001 00110 000 01000 0110011

0x41930433

0100 0001 1001 0011 0000 0100 0011 0011

4 1 9 3 0 4 3 3

B1 FEFOP0

N [3] Consulte as tabelas no verso para fazer as conversões abaixo. Se houver constante negativa, expresse como número negativo; se houver salto, identifique o endereço de memória destino em 32 bits (ex.: beq \$1, \$2, 0x0040 1234), supondo que a instrução sendo executada está no endereço 0x0040 0008.

[a] Qual é a instrução completa em assembly RISC-V que é codificada para o código de máquina de 32 bits 0x00C06393?

[b] Qual é a instrução completa em assembly RISC-V que é codificada para o código de máquina de 32 bits 0x00938D33?

0000|0000|1100|0000|0110|00011|1001|0011

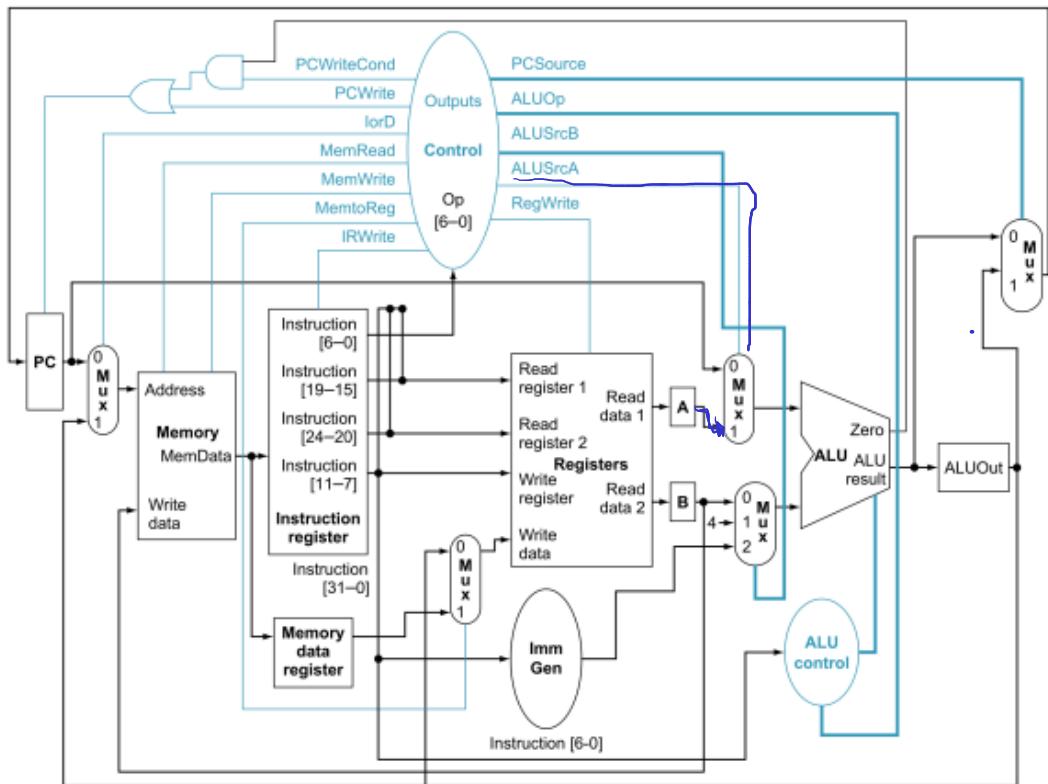
~~~~~ 110 00000 110 00111 001 0011

constante rs1 f3 rd

12

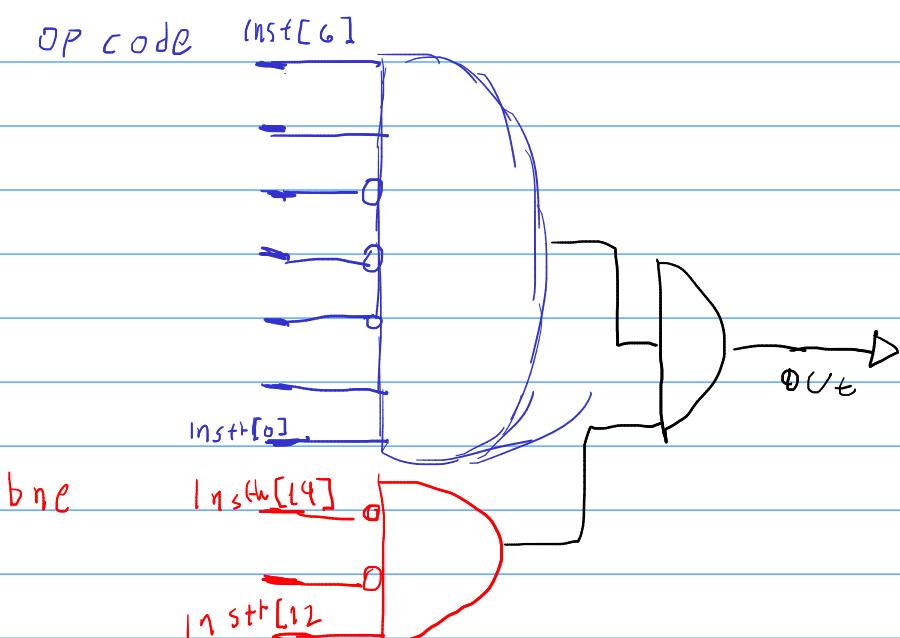
0 h I t2, Zeto, 12

Q[4] No circuito multiciclo do verso, se o sinal ALUSrcA ficar travado em 1 por algum bug, haverá um comportamento incorreto no processador. Qual é este comportamento?



Ele vai mudar o PC para um valor de um registrador A, e isso pode tudo tendo instruções não previstas

Q[5] Mostre o circuito decodificador que identifica se a instrução lida é uma instrução bne. Dito de outra forma, mostre a portona lógica que é parte da UC e que gera nível 1 se e somente se a instrução é bne.



[6] Faça um programa assembly RISC-V que conta quantos números 0x0000 0000 estão presentes na faixa de memória entre 0x1001 0000 e 0x1002 0000 (exclusive). Considere as palavras de 32 bits alinhadas na memória.

```
.text
.globl _start

_start:
    li t0, 0x10010000      # t0 = endereço inicial
    li t1, 0x10020000      # t1 = endereço final (exclusivo)
    li t2, 0x00000000      # t2 = valor a ser comparado
    li t3, 0                # t3 = contador de zeros encontrados

loop:
    beq t0, t1, end        # se chegou ao fim, sai do loop
    lw t4, 0(t0)            # carrega palavra da memória para t4
    bne t4, t2, skip        # se não for zero, pula incremento
    addi t3, t3, 1           # incrementa contador

skip:
    addi t0, t0, 4          # avança para próxima palavra (32 bits)
    j loop

end:
    # t3 contém o resultado (número de palavras 0x00000000)
    # fim do programa – em sistemas reais, poderia armazenar o valor ou usar syscall
    nop
```

[7] Faça um programa assembly RISC-V que determina se a sequência de dois números 0x12345678 e em seguida 0x87654321 estão presentes em dois lugares subsequentes na faixa de memória entre 0x1001 0000 e 0x1002 0000 (exclusive). Considere as palavras de 32 bits alinhadas na memória.

|      |                 |          |
|------|-----------------|----------|
| addi | \$0, zero, 0    | # fl26   |
| li   | t1, 0x12345678  | *Value 1 |
| li   | t2, 0xb7654321  | #Value 2 |
| li   | \$2, 0x10010000 | # Init   |
| li   | \$3, 0x10020000 | # Init   |

|     |               |      |
|-----|---------------|------|
| LW  | \$5, 0(\$2)   | -123 |
| beq | \$2, \$3, end | -723 |
|     |               | 876  |

bkz \$0, zero, flag activated  
 beq zero, zero, flag deactivated

flag activated:

beq t1, \$5, devB0m

flag deactivated:

beq t1, \$5, lig2Flag

addi \$6, zero, 0

beq, zero, zero, semLig2

lig2Flag:

addi \$0, zero, 1

sem Lig2:

[8] Dado o programa abaixo, quantos clocks ele leva para ser executado no RISC-V com o pipeline visto em sala de aula? Considere que não há forwarding/bypass ou seja, utilizamos bolhas quando necessárias.

- A: add s2,zero,zero
- B: addi s3,zero,65
- C: addi s4,s2,31
- D: add s2,s3,s3
- E: add s3,s3,s3
- F: add s2,s2,s2

| CLK | if | ID | exec | mem | wb  |
|-----|----|----|------|-----|-----|
| 1   | a  |    |      |     |     |
| 2   | b  | a  |      |     |     |
| 3   | c  | b  | a    |     |     |
| 4   | d  | c  | b    | a   |     |
| 5   | D  | C* | Nop  | b   | a   |
| 6   | E  | D  | C    | Nop | b   |
| 7   | F  | E  | D    | C   | Nop |
| 8   |    | F  | E    | D   | C   |
| 9   | F* |    | Nop  | E   | D   |
| 10  |    | F  | Nop  | E   |     |
| 11  |    |    | F    | Nop |     |
| 12  |    |    |      | F   |     |
| 13  |    |    |      |     |     |