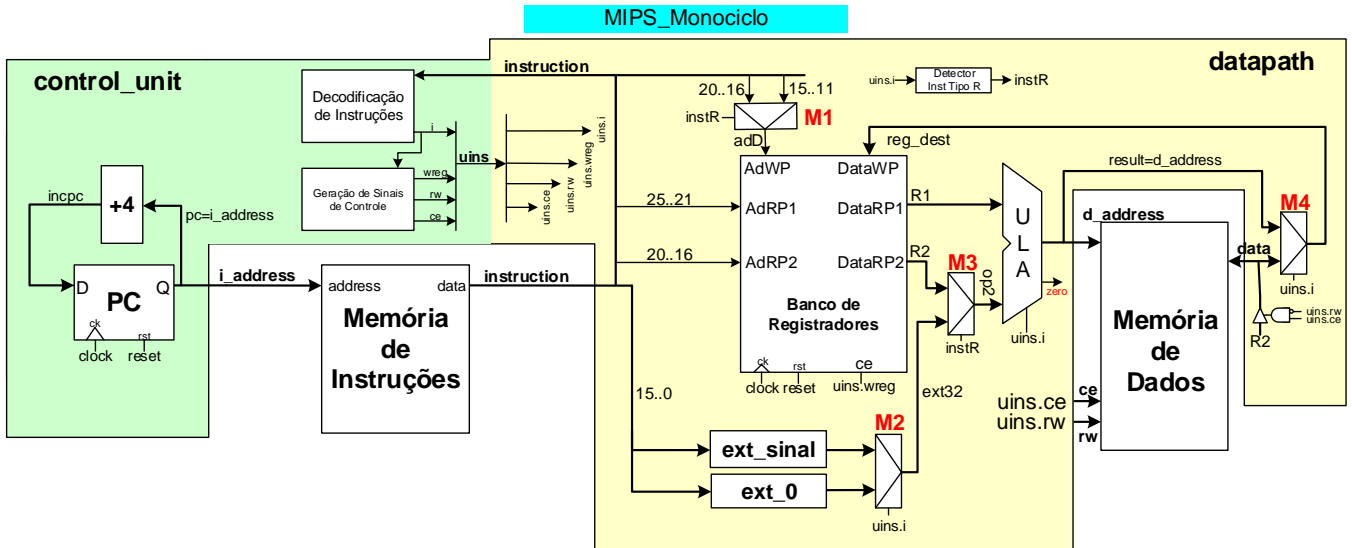


Aluno:

20/novembro/2018

1. [3,0 pontos] Assumindo uma frequência de relógio de 200MHz para a organização MIPS monociclo estudada em aula (e com o diagrama de blocos dado abaixo) calcule, para o programa desta questão, as informações pedidas nos itens a) e b). Assuma que a organização original foi alterada para dar suporte a todas as instruções do programa abaixo, mantendo a característica monociclo.

- a) (2 pontos) Calcule o número de ciclos de relógio consumidos para executar o programa, considerando a área de dados fornecida;
- b) (1 ponto) Compute o tempo de execução do programa, medido em segundos.



```

.text
.globl main
main:
    la    $t0, TI
    lw    $t0, 0($t0)
    la    $t1, Q
    lw    $t1, 0($t1)
    la    $t2, N
    lw    $t2, 0($t2)
    la    $t3, SOMAN
    beq   $t2, $zero, fim
    sw    $t0, 0($t3)
    addiu $t2, $t2, -1
loop:  beq   $t2, $zero, fim
    multu $t0, $t1
    mflo  $t0
    lw    $t4, 0($t3)
    addu  $t4, $t4, $t0
    sw    $t4, 0($t3)
    addiu $t2, $t2, -1
    j     loop
fim:   j     fim

.data
TI:    .word 0x1
Q:     .word 0x5
N:     .word 0x4
SOMAN: .word 0
    
```

2. [2,0 pontos]. No desenho da MIPS monociclo da questão anterior, note que existem 4 multiplexadores (muxes) nesta organização, denominados M1, M2, M3 e M4. Responda às questões a seguir sobre estes componentes:

- a) (1,0 ponto). Das 9 instruções da arquitetura MIPS 2000 que o processador MIPS monociclo pode executar, qual ou quais instruções não poderiam mais executar corretamente se o multiplexador

M3 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a sua saída deixasse sempre passar o sinal R2 (entrada superior do mux M3)?

- b) (1,0 ponto). Das 9 instruções da arquitetura MIPS 2000 que o processador MIPS monociclo pode executar, qual ou quais instruções não poderiam mais executar corretamente se o multiplexador M1 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a saída deixasse sempre passar a sua entrada instruction(20 downto 16) (marcada no desenho como a entrada rotulada **20..16**)?

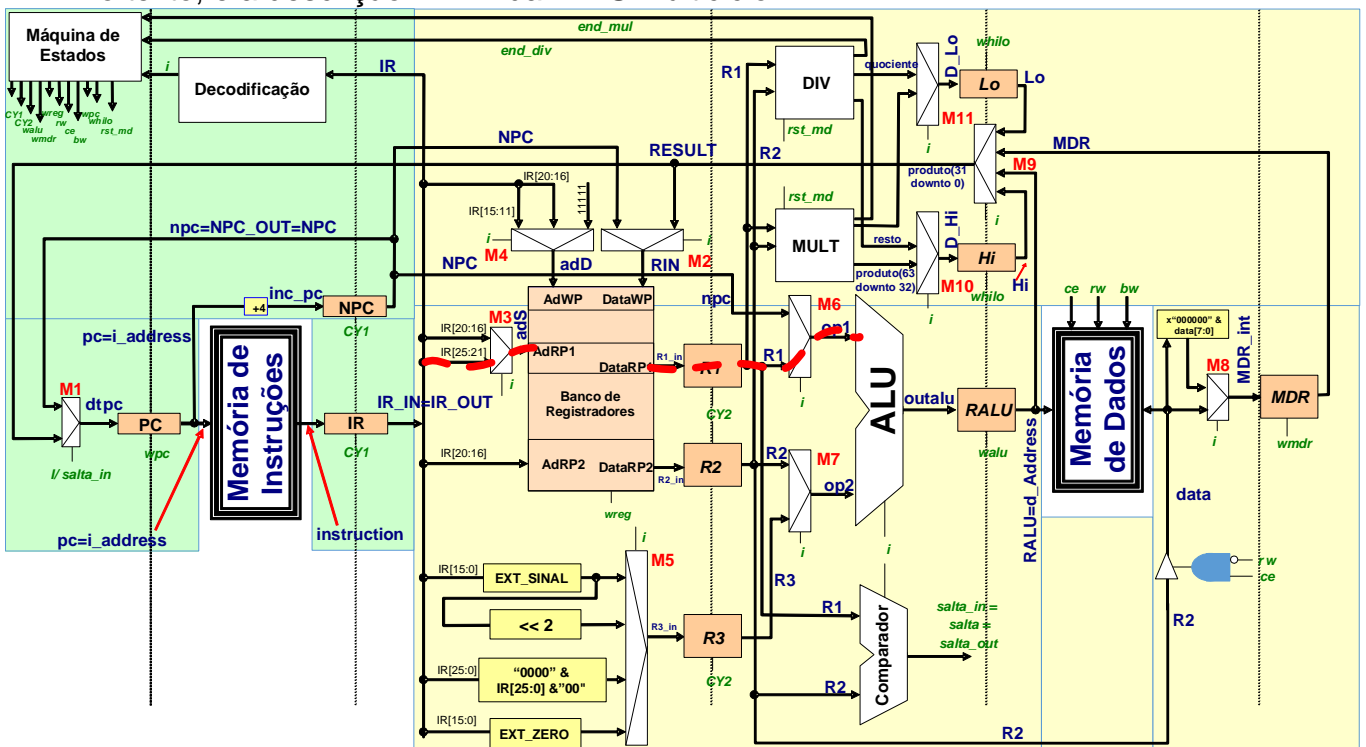
3. [2,0 pontos] O código VHDL dado abaixo descreve o hardware que gera os dois multiplexadores que recebem as saídas dos módulos de multiplicação e divisão, usados para executar as instruções MULTU e DIVU na MIPS multiciclo, e produz as entradas dos registradores Hi e Lo, usados para armazenar o resultado de cada uma destas instruções ao final da execução das mesmas.

```
D_Hi <= produto(63 downto 32) when uins.i=MULTU else
    resto;
D_Lo <= produto(31 downto 0) when uins.i=MULTU else
    quociente;
```

O objetivo desta questão é que o aluno mostre o hardware gerado por estes comandos, partindo dos pressupostos dados a seguir. Abaixo mostra-se o código que define o tipo `inst_type`, que pode identificar simbolicamente cada uma das 37 instruções e também uma instrução inválida (`invalid_instruction`). Assuma que o sinal `uins.i` é codificado com a quantidade mínima possível `n` de bits (determine o valor de `n`), iniciando com o valor 0 (representado em `n` bits) associado à instrução `ADDU`, o valor 1 associado à instrução `SUBU` e assim por diante para todas as instruções e para `invalid_instruction`. Assumindo esta codificação, defina o circuito que gera os sinais de controle dos dois multiplexadores e desenhe os multiplexadores com todas entradas e saídas corretamente identificadas.

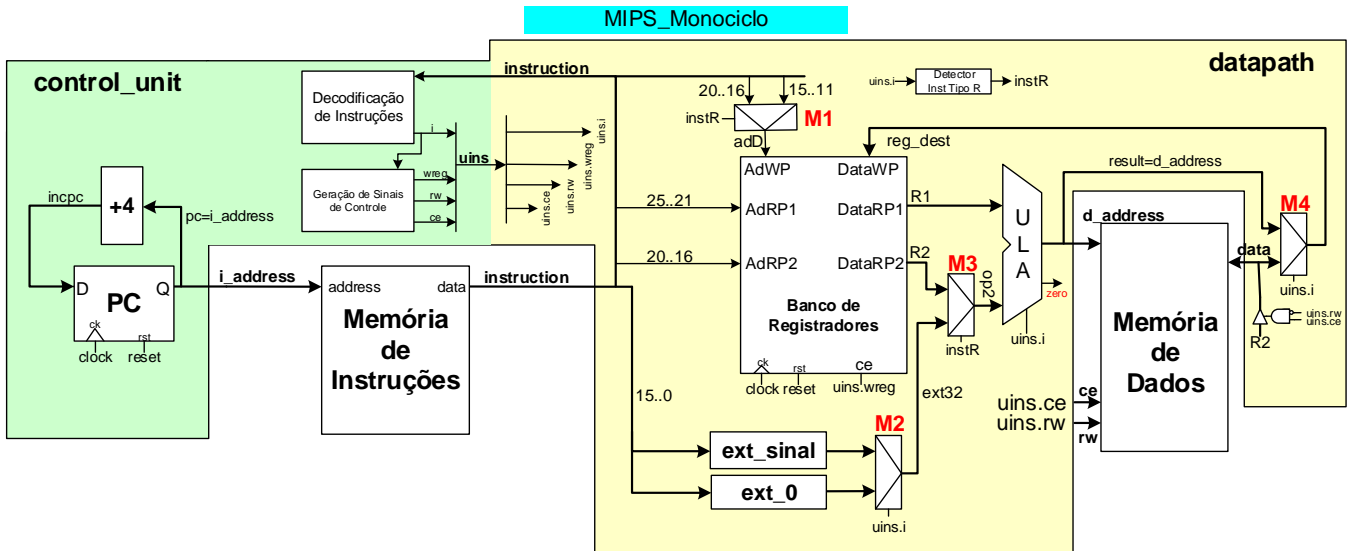
```
type inst_type is (ADDU, SUBU, AAND, OOR, XXOR, NNOR, SLL, SLLV, SSRA, SRAV,
    SSRL, SRLV, ADDIU, ANDI, ORI, XORI, LUI, LBU, LW, SB, SW, SLT,
    SLTU, SLTI, SLTIU, BEQ, BGEZ, BLEZ, BNE, J, JAL, JALR, JR,
    MULTU, DIVU, MFHI, MFLO, invalid_instruction);
```

4. [3,0 pontos] Considere o processador multiciclo abaixo (visto em aula) e marque sobre ele todos os caminhos de dados usados pela instrução `SLLV`. Ressalte também todos os sinais de controle ativados para executar a instrução. A título de dica um dos caminhos de dados usados pela instrução já está marcado no desenho, com linhas tracejadas. Use, além do diagrama de blocos dado, a especificação da instrução `SLLV` que consta no Apêndice A do livro texto, e a descrição VHDL da MIPS multiciclo.



# Gabarito

1. [3,0 pontos] Assumindo uma frequência de relógio de 200MHz para a organização MIPS monociclo estudada em aula (e com o diagrama de blocos dado abaixo) calcule, para o programa desta questão, as informações pedidas nos itens a) e b). Assuma que a organização original foi alterada para dar suporte a todas as instruções do programa abaixo, mantendo a característica monociclo.
- a) (2 pontos) Calcule o número de ciclos de relógio consumidos para executar o programa, considerando a área de dados fornecida;
- b) (1 ponto) Compute o tempo de execução do programa, medido em segundos.



Solução:

a)

		ciclos
.text		0
.globl main		0
main:		0
la \$t0, TI		2
lw \$t0, 0(\$t0)		1 \$t0 recebe o valor do Termo Inicial TI
la \$t1, Q		2
lw \$t1, 0(\$t1)		1 \$t1 recebe a razão Q da PG
la \$t2, N		2
lw \$t2, 0(\$t2)		1 \$t2 recebe o número de termos a somar, N
la \$t3, SOMAN		2 \$t3 recebe endereço do resultado
beq \$t2, \$zero, fim		1 testa se N=0. Se sim, fim
sw \$t0, 0(\$t3)		1 Inicializa resultado com o valor de TI
addiu \$t2, \$t2, -1		1 Decrementa contador. Até aqui, 14 ciclos
loop: beq \$t2, \$zero, fim		1 Aqui, testa contador que começa com valor N-1
multu \$t0, \$t1		1 senão, multiplica valor atual pela razão
mflo \$t0		1 Recupera para \$t0 o resultado
lw \$t4, 0(\$t3)		1 Recupera soma parcial para \$t4
addu \$t4, \$t4, \$t0		1 Atualiza a soma
sw \$t4, 0(\$t3)		1 Armazena de volta na memória
addiu \$t2, \$t2, -1		1 atualiza contador
j loop		1 Loop = Cada volta 8 ciclos, última 1 ciclo
fim: j fim		1 Trava
		Total = 14 + (N-1)*8 + 1 + 1 = 40 ciclos para N=5
.data		
TI:	.word 0x1	
Q:	.word 0x5	
N:	.word 0x4	
SOMAN:	.word 0	

- b) Um relógio de 200MHz implica um período de relógio de  $(1/(200 \cdot 10^6))$ s, ou seja, 5ns. Assim, para a área de dados dada, o tempo de execução é  $5\text{ns} \cdot 40\text{ciclos} = 200\text{ns}$  ou  $0,2\mu\text{s}$  ou  $0,0000002\text{s}$ .

2. [2,0 pontos]. No desenho da MIPS monociclo da questão anterior, note que existem 4 multiplexadores (muxes) nesta organização, denominados M1, M2, M3 e M4. Responda às questões a seguir sobre estes componentes:
- (1,0 ponto). Das 9 instruções da arquitetura MIPS 2000 que o processador MIPS monociclo pode executar, qual ou quais instruções não poderiam mais executar corretamente se o multiplexador M3 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a sua saída deixasse sempre passar o sinal R2 (entrada superior do mux M3)?
  - (1,0 ponto). Das 9 instruções da arquitetura MIPS 2000 que o processador MIPS monociclo pode executar, qual ou quais instruções não poderiam mais executar corretamente se o multiplexador M1 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a saída deixasse sempre passar a sua entrada instruction(20 downto 16) (marcada no desenho como a entrada rotulada **20..16**)?

**Solução:** As 9 instruções da MIPS mono são ADDU, SUBU AND, OR, XOR, NOR, LW, SW e ORI.

- Uma falha como descrita no mux M3 implica que nenhuma instrução que use algum dado imediato pode executar corretamente, pois elas precisam que o sinal de extensão (ext32) passa para a entrada da ULA. Isto significa que as instruções LW, SW e ORI não executam corretamente.
  - Uma falha como descrita no mux M1 implica que nenhuma instrução que precise endereçar o registrador de destino usando os bits 15-11 a instrução (campo Rd) pode mais executar, pois elas precisam que este sinal passe para a entrada da porta de endereçamento de escrita do Banco de Registradores (sinal adD, pinos AdWP do Banco). As instruções que usam este campo são exatamente as instruções tipo R, ou seja: ADDU, SUBU AND, OR, XOR e NOR, que não executam corretamente.
3. [2,0 pontos] O código VHDL dado abaixo descreve o hardware que gera os dois multiplexadores que recebem as saídas dos módulos de multiplicação e divisão, usados para executar as instruções MULTU e DIVU na MIPS multiciclo, e produz as entradas dos registradores Hi e Lo, usados para armazenar o resultado de cada uma destas instruções ao final da execução das mesmas.

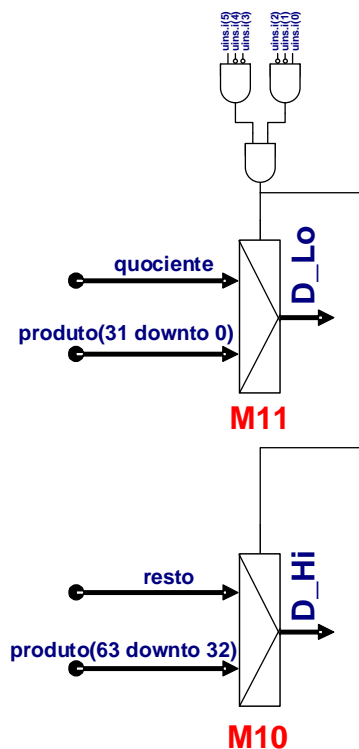
```
D_Hi <= produto(63 downto 32) when uins.i=MULTU else
      resto;
D_Lo <= produto(31 downto 0) when uins.i=MULTU else
      quociente;
```

O objetivo desta questão é que o aluno mostre o hardware gerado por estes comandos, partindo dos pressupostos dados a seguir. Abaixo mostra-se o código que define o tipo `inst_type`, que pode identificar simbolicamente cada uma das 37 instruções e também uma instrução inválida (`invalid_instruction`). Assuma que o sinal `uins.i` é codificado com a quantidade mínima possível **n** de bits (determine o valor de **n**), iniciando com o valor 0 (representado em **n** bits) associado à instrução `ADDU`, o valor 1 associado à instrução `SUBU` e assim por diante para todas as instruções e para `invalid_instruction`. Assumindo esta codificação, defina o circuito que gera os sinais de controle dos dois multiplexadores e desenhe os multiplexadores com todas entradas e saídas corretamente identificadas.

```
type inst_type is (ADDU, SUBU, AAND, OOR, XXOR, NNOR, SLL, SLLV, SSRA, SRAV,
                  SSR, SRLV, ADDIU, ANDI, ORI, XORI, LUI, LBU, LW, SB, SW, SLT,
                  SLTU, SLTI, SLTIU, BEQ, BGEZ, BLEZ, BNE, J, JAL, JALR, JR,
                  MULTU, DIVU, MFHI, MFLO, invalid_instruction);
```

**Solução:**

Como existem 37 instruções e o símbolo `invalid_instruction` é necessário codificar 38 informações distintas o que implica um código de no mínimo 6 bits. Com a convenção estabelecida, `ADDU` será codificada 000000, `SUBU` com 000001, etc e `invalid_instruction` com 100101 (37 em decimal). Observando o teste nas linhas que definem `D_Hi` e `D_Lo`, nota-se que apenas se testa pelo valor da instrução `MULTU`, cujo código é 33 em decimal ou 100001. Assim o sinal de controle de ambos multiplexadores é o mesmo, a saída de um detetor do código 100001. O desenho abaixo mostra então o circuito completo, solução do problema.



4. [3,0 pontos] Considere o processador multiciclo abaixo (visto em aula) e marque sobre ele todos os caminhos de dados usados pela instrução SLLV. Ressalte também todos os sinais de controle ativados para executar a instrução. A título de dica um dos caminhos de dados usados pela instrução já está marcado no desenho, com linhas tracejadas. Use, além do diagrama de blocos dado, a especificação da instrução SLLV que consta no Apêndice A do livro texto, e a descrição VHDL da MIPS multiciclo.

Solução:

