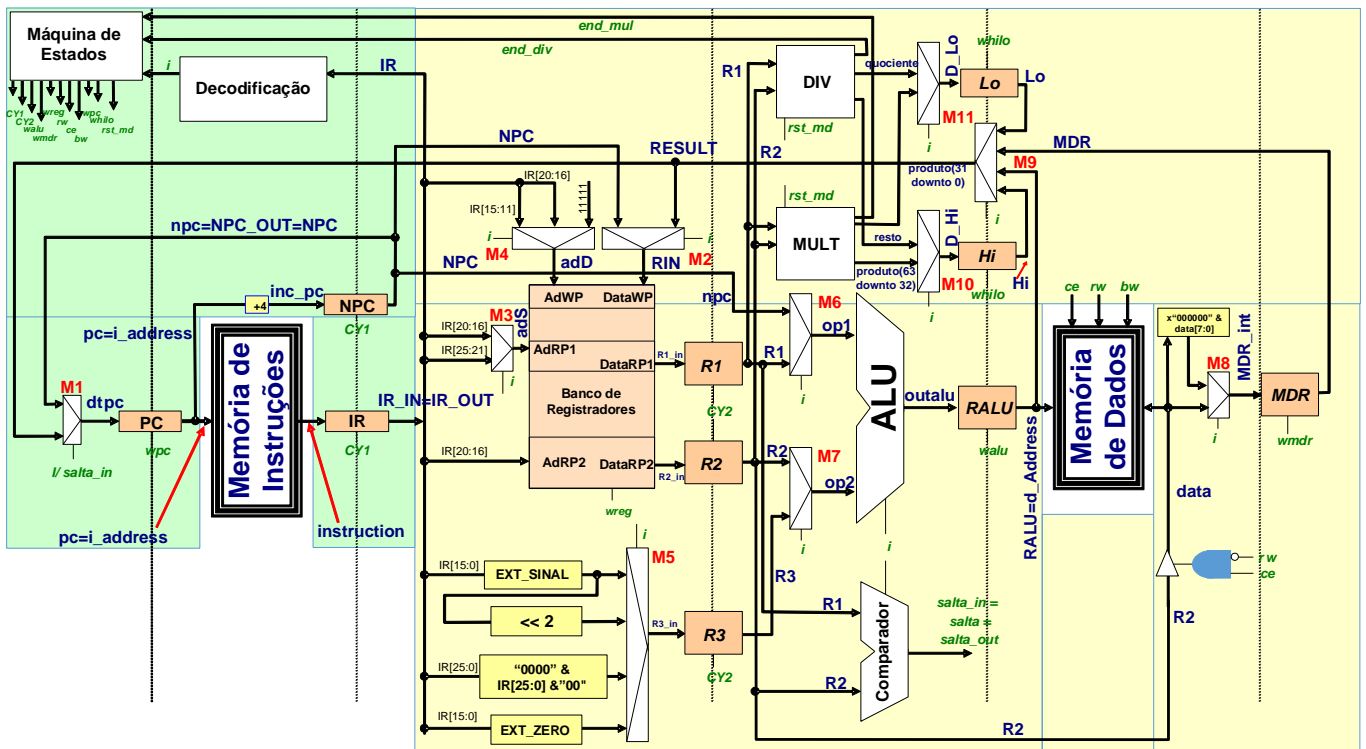


1. [3,0 pontos] No desenho da MIPS monociclo acima, note que existem 4 multiplexadores (muxes) nesta organização, denominados M1, M2, M3 e M4. Responda às questões a seguir sobre estes componentes e seu uso/efeito no processador:
  - (a) (1,5 pontos). Das 9 instruções da arquitetura MIPS 2000 que a MIPS monociclo pode executar, qual ou quais instruções não poderiam mais executar corretamente se o multiplexador M2 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a sua saída deixasse sempre passar o sinal **ext\_0** (entrada inferior do mux M2)?
  - (b) (1,5 pontos). 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(15 downto 11)** (marcada no desenho como a entrada direita do mux M1, rotulada **15..11**)?
  
2. [3,0 pontos] Considere o processador MIPS multiciclo a seguir (visto em aula) e marque sobre ele todos os caminhos de dados e de controle usados para executar a instrução **JALR**. Ressalte também todos os sinais de controle efetivamente usados para executar a instrução. Não esqueça de identificar qual ou quais dos 11 sinais de controle gerados pela máquina de estados de controle do processador são relevantes para a execução da **JALR**. Use, além do diagrama de blocos fornecido abaixo, a especificação da instrução **JALR** que consta no Apêndice A do livro texto, a descrição VHDL da MIPS multiciclo, bem como a especificação textual da MIPS multiciclo.



3. [4 pontos] Assuma uma frequência de relógio de 400 MHz para a organização **MIPS multiciclo** estudada em aula e diga:

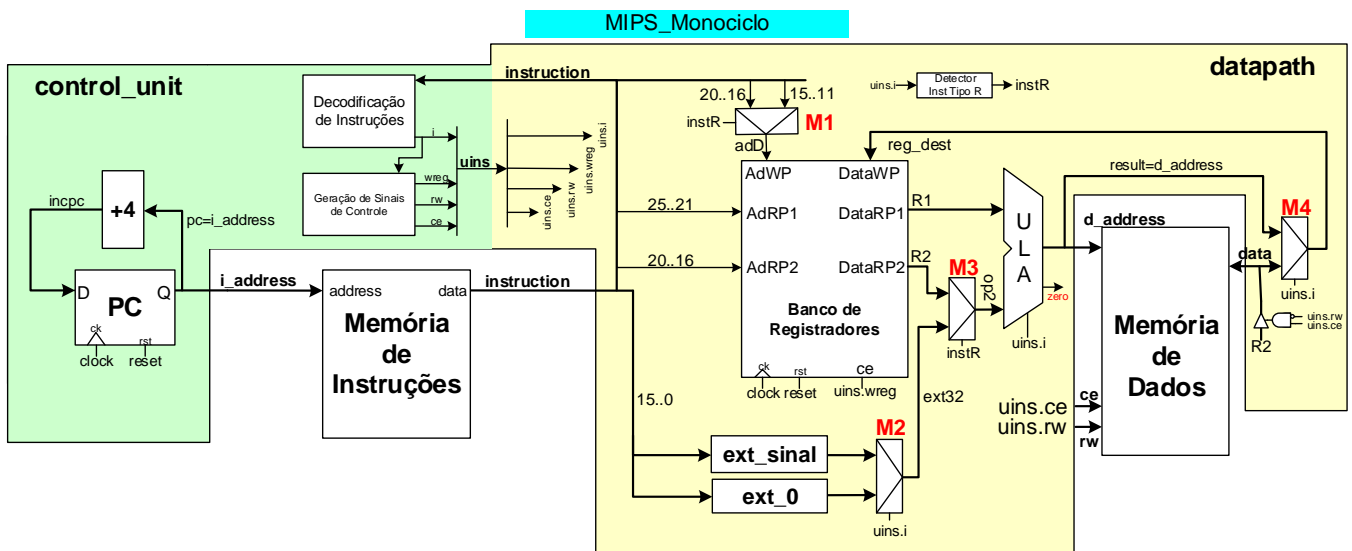
- Qual o número de ciclos de relógio consumidos para a execução do programa abaixo nesta organização (Considere a área de dados fornecida, assumindo que a pseudo-instrução **la** leva 8 ciclos de relógio para executar sendo equivalente a duas instruções, um **lui** seguido de um **ori**), e que a instrução **syscall** leva 4 ciclos de relógio para executar).
- Qual o tempo de execução do programa, em **segundos**.
- O que faz este programa.
- Se o programa contém alguma subrotina. Em caso afirmativo, identifique-a no código.

```

1.      .data
2.  ar:  .word      0x12 0xff 0x3 0x14 0x878 0x31  0x62 0x10 0x5 0x16 0x20
3.  s:   .word      11
4.
5.      .text
6.      .globl     main
7.  main: la       $t0,ar
8.        la       $t1,s
9.        lw       $t1,0($t1)
10. l:   beq      $t1,$zero,end
11.        lw       $t2,0($t0)
12.        multu   $t2,$t2
13.        mflo    $t2
14.        sw      $t2,0($t0)
15.        addiu   $t0,$t0,4
16.        addiu   $t1,$t1,-1
17.        j       1
18. end:  ori      $v0,$zero,10
19.      syscall

```

# Gabarito



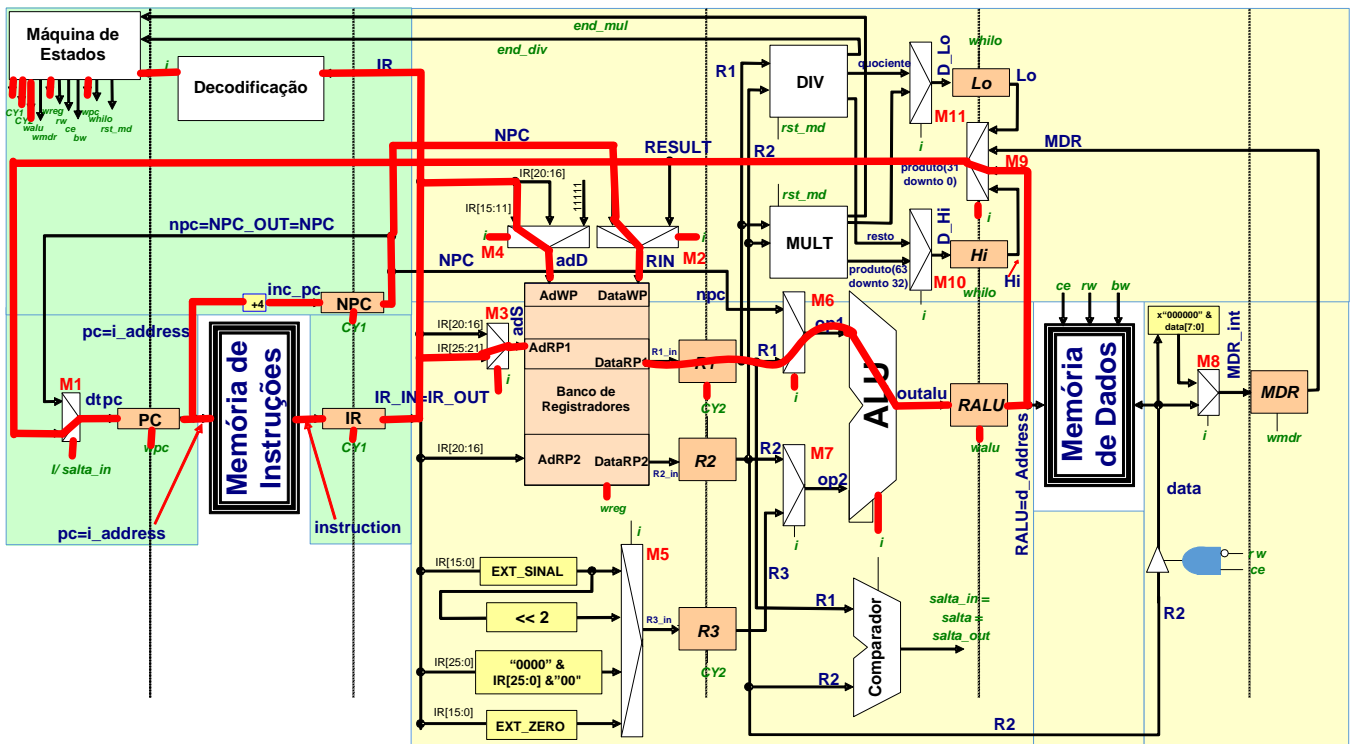
- [3,0 pontos] No desenho da MIPS monociclo acima, note que existem 4 multiplexadores (muxes) nesta organização, denominados M1, M2, M3 e M4. Responda às questões a seguir sobre estes componentes e seu uso/efeito no processador:
  - (1,5 pontos). Das 9 instruções da arquitetura MIPS 2000 que a MIPS monociclo pode executar, qual ou quais instruções não poderia(m) mais executar corretamente se o multiplexador M2 falhasse de tal forma que, independentemente do valor de seu sinal de controle, a sua saída deixasse sempre passar o sinal **ext\_0** (entrada inferior do mux M2)?
  - (1,5 pontos). 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(15 downto 11) (marcada no desenho como a entrada direita do mux M1, rotulada 15..11)?

**Solução:**

- O mux M2 controla a escolha do tipo de extensão de 16 bits usar para um dado imediato, seja ele uma constante ou um deslocamento para cálculo de endereços. Assim, apenas as instruções que necessitam alguma forma de extensão dos bits 15 a 0 da instrução são afetadas, ou seja, LW, SW e ORI seriam as instruções que **potencialmente** não poderiam operar mais corretamente com esta falha. Contudo, ORI usa extensão de 0 (gerada no sinal **ext\_0**) e continuaria então a funcionar normalmente. LW e SW, no entanto, precisam da extensão de sinal e não funcionariam mais corretamente. Resposta: **LW e SW**.
- O mux M1 controla a escolha do endereço do registrador a ser escrito como resultado da execução de uma instrução. A falha conecta sempre o campo Rd da instrução a este endereço. Logo, as instruções que falham são aquelas que necessitam escrever usando o campo Rt (bits 20 a 16 da instrução). Pelo Apêndice A, aprende-se que estas instruções são LW e ORI apenas (note-se que SW não escreve no banco, logo o endereço do registrador de escrita é irrelevante para ela). Resposta: **LW e ORI**.

- [3,0 pontos] Considere o processador MIPS multiciclo a seguir (visto em aula) e marque sobre ele todos os caminhos de dados e de controle usados para executar a instrução JALR. Ressalte também todos os sinais de controle efetivamente usados para executar a instrução. Não esqueça de identificar qual ou quais dos 11 sinais de controle gerados pela máquina de estados de controle do processador são relevantes para a execução da JALR. Use, além do diagrama de blocos fornecido abaixo, a especificação da instrução JALR que consta no Apêndice A do livro texto, a descrição VHDL da MIPS multiciclo, bem como a especificação textual da MIPS multiciclo.

**Solução:**



3. [4 pontos] Assuma uma frequência de relógio de 400 MHz para a organização MIPS multiciclo estudada em aula e diga:
- Qual o número de ciclos de relógio consumidos para a execução do programa abaixo nesta organização (Considere a área de dados fornecida, assumindo que a pseudo-instrução **la** leva 8 ciclos de relógio para executar sendo equivalente a duas instruções, um **lui** seguido de um **ori**), e que a instrução **syscall** leva 4 ciclos de relógio para executar).
  - Qual o tempo de execução do programa, em **segundos**.
  - O que faz este programa.
  - Se o programa contém alguma subrotina. Em caso afirmativo, identifique-a no código.

Solução:

```

1.      .data
2.  ar:  .word      0x12 0xff 0x3 0x14 0x878 0x31  0x62 0x10 0x5 0x16 0x20
3.  s:   .word      11
4.
5.      .text
6.      .globl     main          # ciclos
7.  main: la        $t0,ar        # 8 - $t0 recebe ponteiro para vetor ar
8.        la        $t1,s         # 8 -
9.        lw        $t1,0($t1)    # 5 - $t1 recebe tamanho do vetor
10. l:   beq        $t1,$zero,end  # 4 - Se tratamento do vetor acabou, fim
11.        lw        $t2,0($t0)    # 5 - Senão, lê elemento para $t2
12.        multu    $t2,$t2       # 67 - Gera quadrado do elemento
13.        mflo     $t2           # 4 - Busca resultado de lo para $t2
14.        sw        $t2,0($t0)    # 4 - Escreve resultado de volta
15.        addiu    $t0,$t0,4     # 4 - Avança ponteiro para próx elemto
16.        addiu    $t1,$t1,-1    # 4 - Decrementa contador dos que restam
17.        j        l             # 4 - Volta a testar fim do processo
18. end: ori        $v0,$zero,10  # 4 - Gera 10 (ident de syscall fim prog)
19.        syscall                # 4 - Termina o programa

```

- a) O programa é apenas um laço com três linhas preparatórias do laço (linhas 7 a 9) e duas instruções de fechamento do programa (linhas 18 e 19). Estas linhas executam em  $8+8+5 + 4+4=29$  ciclos. O laço ocupa as linhas 10 a 17 e é executado ou totalmente (para todo elemento do vetor **ar**) ou apenas a primeira linha (quando terminar o processamento). O laço executado totalmente gasta  $4+5+67+4+4+4+4+4=96$  ciclos e a última vez gasta 4 ciclos. Como a cadeia **ar** possui 11 valores, o laço é executado 12 vezes. Assim, o número de ciclos para executar este programa pode ser facilmente determinado:  $\text{num\_ciclos} = 29 + 11 \cdot 96 + 4 = 1089$  ciclos.

- b) Como a frequência de execução dada é de 400MHz, um ciclo dura  $(1/(400 * 10^6))$ s ou 2,5ns ou  $2,5 \times 10^{-9}$  s. Logo, o tempo total de execução do programa, que gasta 1089 ciclos de relógio (clock) para executar é  $1089 * (2,5 \times 10^{-9} \text{ s})$  ou seja, *tempo total de execução do programa* =  $2,7225 \times 10^{-6}$  s.
- c) Ele eleva cada elemento (número inteiro) do vetor **ar** ao quadrado.
- d) O programa não possui sub-rotinas, pois não existe nele uso de instruções jal/jalr e jr (\$ra).