UFPR – Bacharelado em Ciência da Computação
CI210 – Projetos Digitais e Microprocessadores – 2018-2

Primeira Prova

1. Quiron lhe apresentou a especificação abaixo, e você dispõe dos componentes listados na tabela. Os FFs possuem saídas Q e $\overline{Q}$. Você projetou o circuito da maneira mais imediata, utilizando portas *and* com quantas entradas fossem necessárias para implementar cada equação, tentando minimizar a quantidade de portas. Assim, para a equação D_1 utilizou uma porta *and* de duas entradas e para D_2 uma de três entradas. Quiron, que tentava torná-lo um herói de projetos digitais, mostrou um exemplo utilizando somente portas *and* com duas entradas, conectando-as em cascata e aproveitando as partes de outras equações sempre que possível. Ele argumentou que ficaria mais barato assim, mas você contra-argumenta que sua opção permite uma frequência de clock maior. Calcule a frequência máxima de cada projeto e veja se você superou o seu mestre neste aspecto. [10 pontos]

$$
\begin{aligned}
D_0 &= \overline{Q_0} \\
D_1 &= Q_0 \wedge \overline{Q_2} \\
D_2 &= Q_0 \wedge \overline{Q_2} \wedge Q_3 \\
D_3 &= Q_0 \wedge \overline{Q_2} \wedge \overline{Q_3}
\end{aligned}
$$

	T_{prop}	T_{cont}	T_{su}	T_h
FF	6	3	4	2
and-2	3	1		
and-3	4	2		

Tempo em nanosegundos.

2. Dédalo era um inventor que estava preso no labirinto do rei Minos, e ele criou um circuito sequencial que a cada bifurcação dizia para onde ir, buscando alcançar a saída. Cada opção, em bifurcações, tinha um identificador de 16 bits; o circuito de Dédalo empregava outro circuito, que calculava o logaritmo base 2 do identificador a fim de poder decidir o caminho. Você deve implementar o circuito que calcula o logaritmo base 2 dos indicadores de 16 bits. Como fazer isto? Conte o número de zeros do identificador, a partir da esquerda, até encontrar o primeiro 1. O logaritmo buscado é igual 16 menos a contagem dos zeros à esquerda. Cuidado para não contar um zero a mais ou a menos, em função do processo de decisão de quando parar. Se contar um a mais ou a menos, você pode ajustar na subtração da constante. Considere que $\log_2(0) = 1$. Este algoritmo fornece uma aproximação aceitável para o caso. Construa o circuito com um bloco operacional (*datapath*) e uma unidade de controle que é uma Máquina de Moore. Considere que o identificador é carregado em um registrador denominado *ID*, de 16 bits, no estado inicial da máquina de estados de controle. Utilize os recursos que julgar necessários, como ULA (especifique operações, desde que não use a função *log*), somador, registrador de deslocamento, flip-flops e registradores. Desenhe um diagrama claro com os componentes utilizados e suas conexões, bem como diagrama de estados do controle. Não é preciso implementar a máquina de estados do controle, apenas especificá-la. [10 pontos]

3. Projete um circuito CMOS que computa a maioria dentre 4 bits. Seu circuito deve produzir saída em 1 quando a maioria absoluta dos bits no quarteto é 1 (4/4 ou 3/4). [10 pontos]

Segunda Prova

4. Você deseja criar um sistema de tradução automática da linguagem Dothraki para Português, a fim de entender o que Qhono fala durante as aulas de projetos. A linguagem tem raízes nos idiomas Russo e Árabe. Para implementar o algoritmo principal de conversão vários laços são necessários. Sua equipe de programação está cansada de montar esquemas de decrementar um contador, testar se o mesmo atingiu zero e, caso negativo, desviar para um determinado local do programa (label do loop, no caso). Para ajudar os programadores, você decide que o processador MIPS terá uma nova instrução, denotada por **djnz rs, label**. Esta instrução decrementa de uma unidade o registrador rs; se após isto ele não valer zero, o fluxo de execução é dirigido para label, indicado por valor imediato 16 bits em complemento de dois; caso contrário, segue para a próxima instrução. [10 pontos]

(a) desenhe a alteração necessária na arquitetura do MIPS, mostrando componentes e sinais envolvidos. Faça isto de forma a ficar claro como estas alterações se enquadram na arquitetura original. Você é o criador aqui, mas não esqueça de explicar em um ou dois parágrafos sua ideia e o que cada elemento faz. Pode acrescentar, remover ou alterar componentes da arquitetura;

(b) indique todos sinais de controle com seus valores para a execução desta nova instrução.

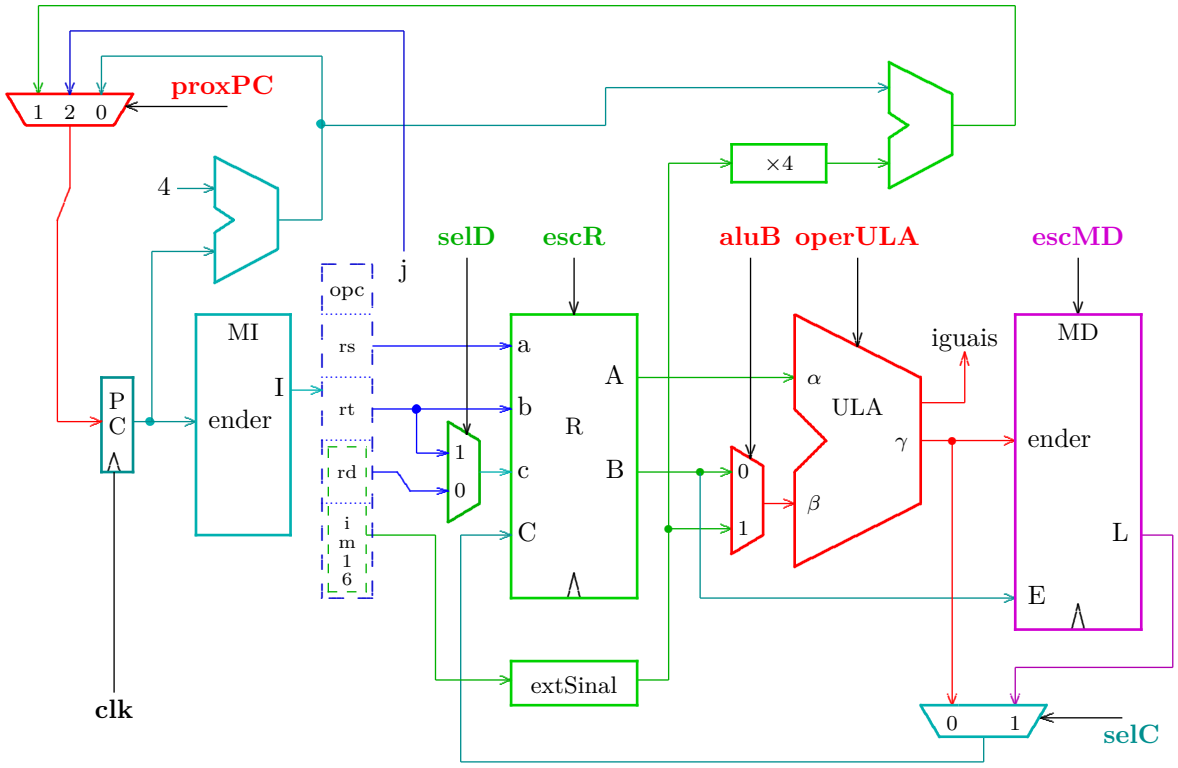

Figura 1: MIPS monociclo com sinais de controle.

5. A função sum() computa a soma de todos os elementos de um vetor. Seus argumentos são o endereço inicial do vetor e o número de elementos. Traduza para assembly do MIPS o trecho de programa abaixo. Seu código assembly deve empregar as convenções de programação do MIPS. [10 pontos]
Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como ri, rn, etc.
Use as folhas internas para escrever o programa em *assembly*.

```
  ...
  int X [1024];
  ...
  s = sum (\&( X [16]), 256);
  ...

int sum (int vet [ ], int num) {
  int s , i ;
  s = i = 0;
  while (i < num) {
    s = s + vet [i];
    i = i + 1;
  }
  return (s);
}
```

Exame Final

6. Esta questão tem dois itens: (a) projete um somador completo de 1 bit como uma função única, usando *transistores CMOS*. Este somador possui duas entradas, A e B, com os bits *a0*, *b0*, e vem-um, *ci0*, e como saída gera *s0* e *co0*. Sua resposta deve conter as expressões, simplificação e circuito com transistores. (b) Mostre como usar este bloco para construir um somador completo de 4 bits. [30 pontos]

7. Projete um circuito que efetua a divisão inteira de dois números positivos de 16 bits, como especificado na Equação 1. O circuito deve implementar a divisão por subtrações repetidas. Seu projeto deve conter o circuito de dados e a máquina de estados que o controla. Os dois operandos são carregados nos registradores *ddndo* e *dvsor* quando o sinal de entrada da interface externa *inicia* é ativado. Os valores finais do quociente e do resto são carregados nos registradores *quoc* e *resto* quando o sinal *pronto* for ativado pela máquina de estados do controlador. Descreva brevemente o funcionamento do seu circuito de tal forma que não seja necessário adivinhar nada na correção da sua resposta. [40 pontos]

$$
\begin{aligned}
&inicia, pronto : \mathbb{B} \\
&ddndo, dvsor : \mathbb{B}^{16} \\
&quoc, resto : \mathbb{B}^{16} \\
&div16 : \mathbb{B} \times (\mathbb{B}^{16} \times \mathbb{B}^{16}) \mapsto (\mathbb{B}^{16} \times \mathbb{B}^{16}) \times \mathbb{B} \\
&div16(inicia, ddndo, dvsor, quoc, resto, pronto) \equiv \\
&\qquad inicia \Rightarrow pronto \Rightarrow \\
&\qquad\quad [\, num(ddndo) = num(dvsor) \cdot num(quoc) + num(resto) \,]
\end{aligned}
\tag{1}
$$

8. A função `log2()` computa o logaritmo na base 2 de um inteiro. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [40 pontos]

Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como `ri`, `rn`, *etc.*

Use as folhas internas para escrever o programa em *assembly*.

```
int log2(int n) {
   if (n < 2) then
      return 0;
   else
      return (1 + log2(n/2)); // divisões por deslocamento
}
...
x = log2(96000);
...
```