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

Primeira Prova

1. Arya é valente e não tem escrúpulos em matar seus piores inimigos, ainda mais depois do treinamento com os Sem Rosto. Mindinho pensa que está em vantagem no julgamento frente aos Lordes do Norte, em Winterfell. Contudo, seu destino depende do resultado da implementação da máquina de estados construída com FFs tipo D e equações de próximo estado como especificado a seguir. Os FFs possuem saídas Q e $\overline{Q}$. Mindinho só usa portas de duas entradas. Arya usa portas com o máximo de entradas possível. Ambos usam o menor número de portas possível, mas sem simplificar expressões dadas e sem aproveitar pedaços de uma em outra. Quem fizer o projeto que suportar maior frequência de *clock* sobrevive. Calcule e responda o que aconteceu, indicando a frequência máxima de *clock* de cada projeto. [10 pontos]

$$
\begin{aligned}
D_0 &= \overline{Q_0} \\
D_1 &= Q_0 \vee \overline{Q_2} \vee Q_3 \\
D_2 &= Q_0 \wedge Q_1 \wedge \overline{Q_2} \wedge Q_3 \\
D_3 &= Q_0 \wedge Q_2
\end{aligned}
$$

	T_{prop}	T_{cont}	T_{setup}	T_{hold}
FF	8	3	6	2
and-2	4	1		
and-3	5	2		
or-2	4	1		
or-3	5	2		

Tempos em nanosegundos.

2. Seis anos haviam se passado após a extinção da Ordem dos Cavaleiros Jedi, o desaparecimento do jovem Anakin Skywalker e o surgimento do cruel Lorde Sith Darth Vader. A galáxia agora se encontra dominada pela ditadura, escravidão e opressão. O rebelde Jyn elabora um plano para hackear o banco de dados da Estrela da Morte e roubar as plantas elétricas dela. Para isso precisa projetar um sistema digital que verifique se, em uma mensagem de oito bits gerada pelo sistema de vigilância, existe a sequência "11011", o que indica uma janela de tempo de alarme desativado. Os bits são apresentados serialmente, um de cada vez, por um sinal "input". O início é sinalizado por um sinal "start", disponível junto com o primeiro bit da sequência. Não importa se o primeiro bit é o lsb ou msb. Construa o circuito com um *datapath* e uma unidade de controle, implementada com uma Máquina de Moore. Utilize os recursos que julgar necessários, como ULA (especifique suas operações), somador, registrador de deslocamento, *flip-flops* e registradores. Não se pode usar componentes mágicos com os quais se obtenha o resultado diretamente, sem projetar coisa alguma. Desenhe um diagrama claro com os componentes utilizados e suas conexões, bem como diagrama de estados do controle. Não é necessário implementar a máquina de estados do controle, apenas especificá-la, desenhando seu diagrama de estados, com entradas, saída e transições. [10 pontos]

3. Thomas desperta em um elevador sem lembrança do que ocorreu. Ao sair encontra outros meninos, que explicam que única maneira de sair do local é cruzando um labirinto cheio de monstros. A sobrevivência à noite no labirinto é um grande desafio. As sessões numeradas do labirinto abrem e fecham em uma sequência desconhecida, mas regular. Para detectar um determinado padrão, Thomas projetou um circuito com a seguinte expressão Booleana:

$$Detectado = (S1 \wedge S2) \vee (S3 \wedge S4)$$

Thomas conhece lógica mas não CMOS. Projete o circuito com transistores CMOS N e P, com rede *pull-down* e *pull-up*, que implemente a equação acima. Avalie se implementar a função negada com um inversor na saída (com dois transistores no inversor) utiliza mais ou menos transistores que a implementação da função direta. Detalhe sua resposta com o projeto e a rede de transistores utilizados em cada opção. Considere que sinais de entrada, quando negados, necessitam um inversor. [5 pontos]

Segunda Prova

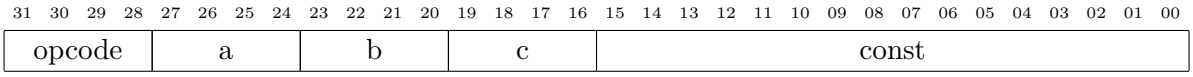
4. Projete um sistema de memória com capacidade total de 16Kbytes, cujo barramento de dados tem largura de 32 bits. Você dispõe de um estoque ilimitado de CIs de memória. Cada CI é uma matriz com 1Kbytes (1024×8).

Sua resposta deve conter um diagrama completo, claro e limpo do projeto. [5 pontos]

5. A codificação das instruções do Mico XII é indicada abaixo. Sua tarefa é mostrar como implementar, no Mico XII, as instruções *"jump indirect through memory"* (salta para a instrução indicada em uma variável em memória, como um *pointer* para uma função em C) e *"jump indirect through memory indexed"* (salta para uma instrução indicada pelo elemento de uma tabela de endereços contida na memória), assim definidas:

(i) a instrução **`ji`** (*jump indirect*) faz: `IP ← M[E];` (opcode α)

(ii) a instrução **`jini`** (*jump indirect indexed*) faz: `IP ← M[E + R(a)];` (opcode β)



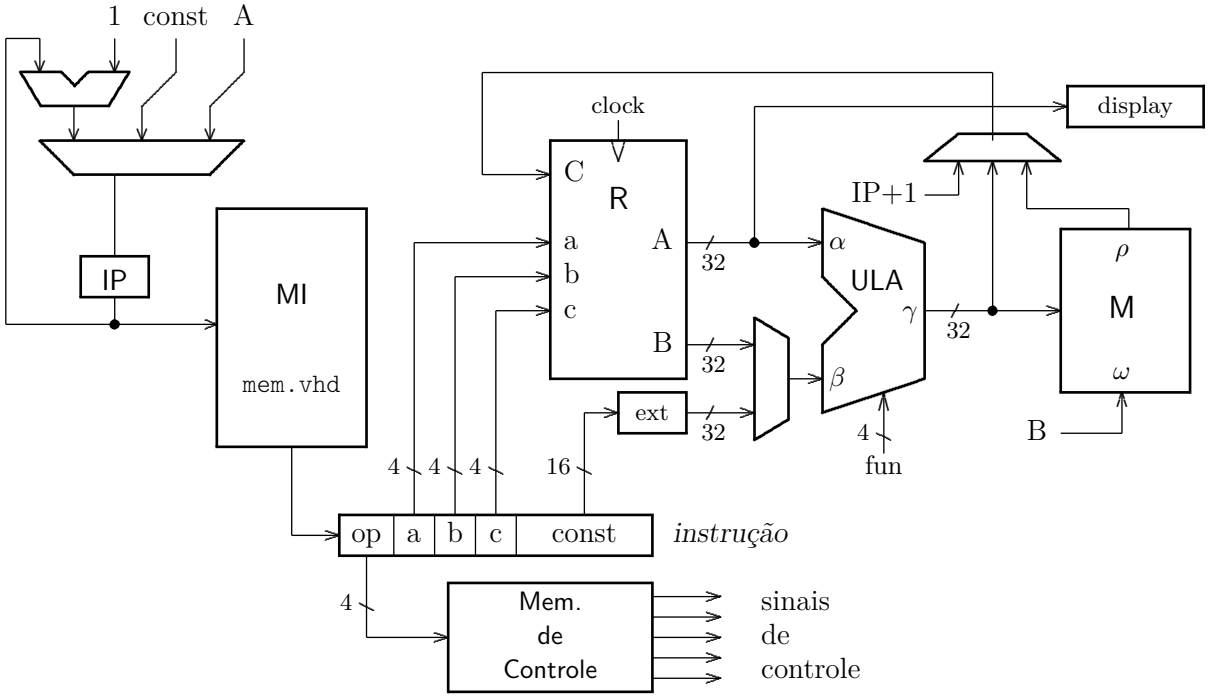
Suponha que existam opcodes disponíveis para as novas instruções.

Sua resposta deve conter: [10 pontos]

(a) a codificação das novas instruções;

(b) alterações no circuito do Mico XII – você pode usar a folha de perguntas para indicar as modificações; não esqueça de colocar seu nome na folha;

(c) o conteúdo da memória de controle para as instruções novas.



6. 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 ao lado. 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 páginas 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

7. Sua tarefa é mostrar como implementar, no MIPS32, as instruções *"jump indirect through memory"* (salta para a instrução indicada em uma variável em memória, como um *pointer* para uma função em C), assim definida: [30 pontos]

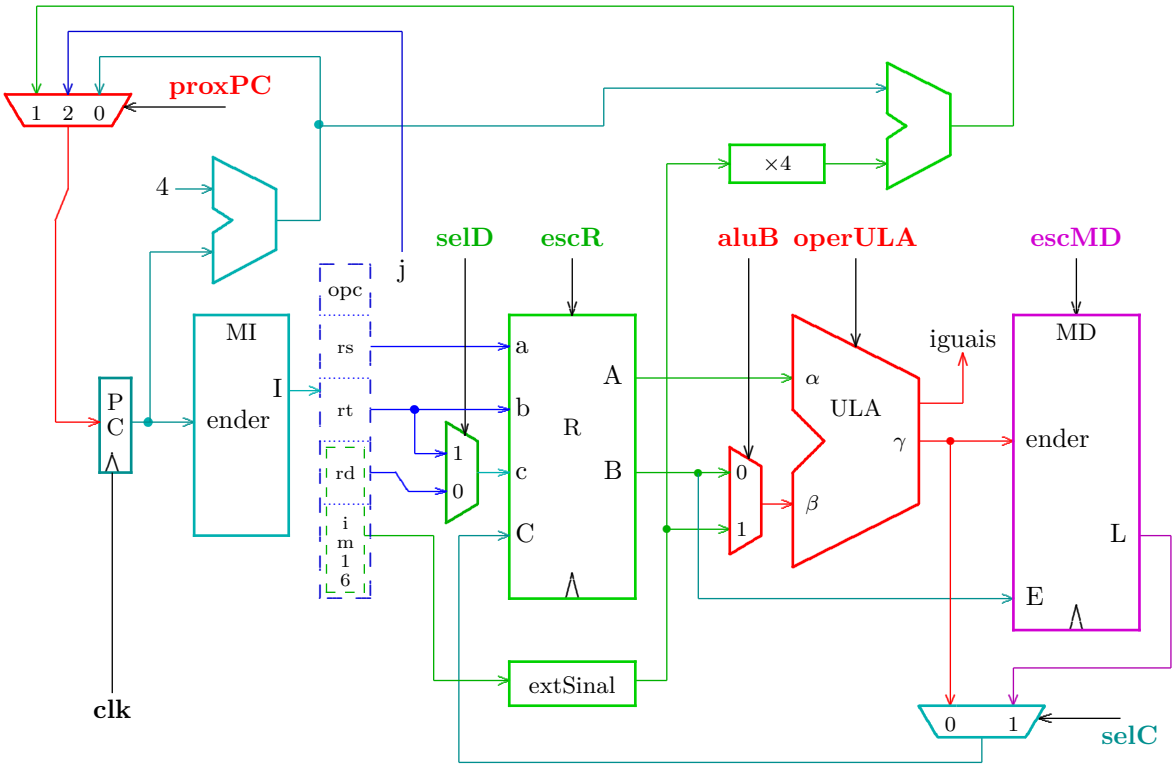
```
ji imed(rs)   #  PC ← M[ R(rs) + extSin(imed) ].
```

Sua resposta deve conter:

(a) a codificação da nova instrução;

(b) alterações no circuito do MIPS – você pode usar a folha de perguntas para indicar as modificações; não esqueça de colocar seu nome na folha;

(c) o conteúdo da memória de controle para a instrução nova.



8. Desenhe uma diagrama de tempos detalhado para o processador MIPS da questão anterior, ao executar a instrução **`ji`**. Compute a velocidade máxima de operação do processador, com os tempos de propagação indicados abaixo. Todos os tempos estão em picosegundos. [30 pontos]

componente	T_{prop}	T_{setup}
PC	40	20
somador	100	
Mem. Instr (MI)	250	
mux-4 de 5 bits	10	
registradores (R)	80	20
mux-2 de 32 bits	20	
mux-4 de 32 bits	40	
ULA (soma)	150	
Mem. Dados (MD)	250	40

9. Traduza para *assembly* do MIPS o trecho de programa ao lado. 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 páginas internas para escrever o programa em *assembly*.

```
int main(void) {
   ...
   x = power(y,z);
   ...
}

int power(int n,int exp) {
   if (exp > 1)
      return (n * power(n,exp-1));
   else
      return (n);
}
```