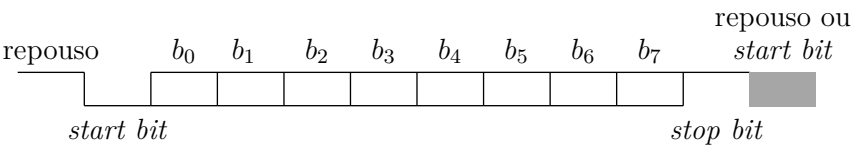
UFPR – Bacharelado em Ciência da Computação
CI210 – Projetos Digitais e Microprocessadores — 2015-2

Primeira Prova

1. Projete o circuito que implementa a parte de transmissão de uma interface serial. A linha de comunicação fica em repouso em 1. Cada octeto é precedido de um *start bit* (bit em 0), o octeto é transmitido iniciando pelo bit menos significativo, e o octeto é seguido por um *stop bit* (bit em 1).

Além do circuito de dados, seu projeto deve conter o circuito de controle, implementado como um microcontrolador (com uma memória ROM).

A transmissão se inicia depois que o registrador de transmissão é carregado com um octeto e o sinal inicia é ativado pelo circuito externo à interface serial. Quando o *stop bit* é enviado, o sinal pronto deve ser ativado pelo controlador da interface serial. [20 pontos]



2. Projete o circuito que implementa a parte de recepção de uma interface serial. A linha de comunicação fica em repouso em 1. Cada octeto é precedido de um *start bit* (bit em 0), o octeto é recebido iniciando pelo bit menos significativo, e o octeto é seguido por um *stop bit* (bit em 1).

Além do circuito de dados, seu projeto deve conter o circuito de controle, implementado como um microcontrolador (com uma memória ROM).

A recepção inicia quando é detectado um *start-bit* após um intervalo de repouso. Quando o *stop bit* é recebido, o sinal pronto deve ser ativado pelo controlador da interface serial. [20 pontos]

3. Abaixo está o código com modelos para dois multiplexadores implementados com comandos de VHDL. Explique as diferenças entre os circuitos combinacionais sintetizados (gerados) a partir destes comandos. [5 pontos]

```
entity mux4x8 is
  port(a,b,c,d: in  bit_vector(7 downto 0);
       s:       in  bit_vector(1 downto 0);
       z:       out bit_vector(7 downto 0));
end mux4x8;

architecture when_else of mux4x8 is
begin
   z <= a when (s = "00") else   -- atribuição condicional
        b when (s = "01") else
        c when (s = "10") else
        d;
end when_else;

architecture with_select of mux4x8 is
begin
  with s select                  -- atribuição selecionada
    r <= a when "00",
         b when "01",
         c when "10",
         d when "11";
end with_select;
```

4. Em VHDL, quais as diferenças entre *sinais* e *variáveis*? [5 pontos]

5. Explique a diferença entre um processo com lista de sensibilidade e um processo sem lista de sensibilidade. Escreva o código para um exemplo de cada tipo de processo. [10 pontos]

Segunda Prova

6. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [15 pontos]
Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como `ri`, `rn`, *etc.*

```
int A[1024]; int B[2048];
...
x = fun(16*i, s=A[16*i], r=B[i], s*r, r%32);    // % = MOD
...
int fun(int g, int h, int i, int j, int k) {
      int f;
      f = (g+h)-(i+j)*k;
      return (f*4);
}
```

7. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [15 pontos]

```
typedef elem {
   elem *nxt;
   int   val;
} elemType;
elemType *lista;
...
  total = reduz(lista);
...
int reduz(elemType *e) {
  if (e->nxt != NULL)
     return( e->val + reduz(e->nxt) );
  else
     return( e->val );
}
```

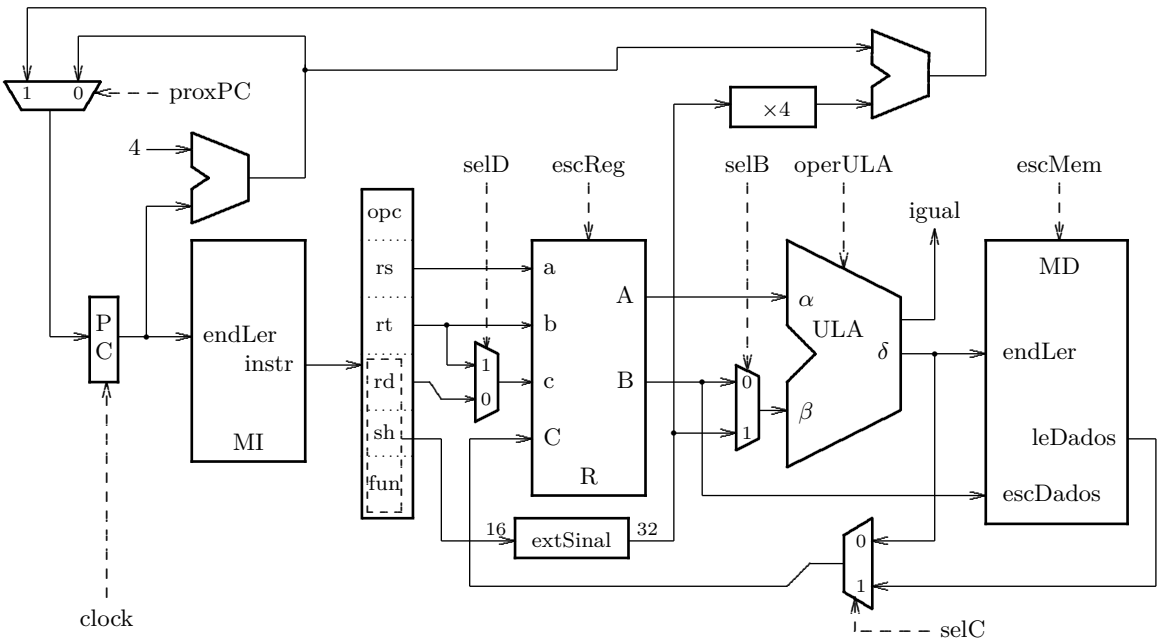
8. Considere um novo modo de endereçamento a ser adicionado ao conjunto de instruções do MIPS. Neste novo modo, o primeiro operando e o resultado são o mesmo registrador e o segundo operando é buscado da memória. A instrução `addm` é definida como

```
addm rt, desloc(rs)   # R[rt] <- R[rt] + Mem[ extSin(desloc) + R[rs] ]
```

(a) Mostre como implementar a instrução `addm`. Adicione quaisquer circuitos necessáios e indique todos os sinais que controlam a execução desta instrução; (b) desenhe um diagrama de tempo para a execução desta instrução no processador com suas adições. [15 pontos]

9. Mostre como implementar a instrução ABSOLUTE VALUE definida abaixo. Sua resposta deve conter: (i) indicação clara de quaisquer modificações necessárias ao circuito do processador, *que podem ser desenhadas na folha de perguntas*; (ii) uma tabela de sinais de controle indicando todos os sinais ativos durante a execução desta instrução; e (iii) um diagrama de tempos completo da execução desta instrução . [15 pontos]

`abs rd, rs   # R[rd] ← ( R[rs] >= 0 ? R[rs] : compl2(R[rs]) )` (formato R)



Exame Final

10. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [40 pontos]

```
typedef elem {
   elem *next;
   int   vet[3];
} elemType;
elemType *x;
elemType strut[256];
   ...
   x = insert( strut, j );
   x->vet[2] = 512;
   ...
elemType * insert(elemType *p, int i) {
  while (p != NULL) {
     p = p->next;
  }
  p->next = &(strut[i]);
  (p->next)->next = NULL;
  return p->next;
}
```

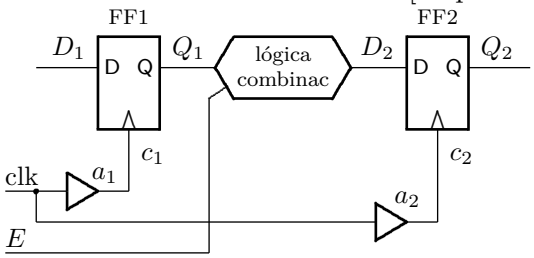
11. Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. [40 pontos]

```
   char fte[N]="abc␣...␣";
   char dst[N];
   ...
   s = reverte(fte, dst, N);
   ...
int reverte(char *f, char *d, int n) {
   int i;
   d = (char *)( (int)d+n );
   for (i=0; (i < n) && (*f != '\0') ; i++, d--, f++)
     *d = *f;
   return i;
}
```

12. Considere o circuito da figura abaixo. A especificação temporal do registrador de estado é: $T_{FF} = 3\,\text{ns}$, $T_{C,F} = 1\,\text{ns}$, $T_s = 2\,\text{ns}$, $T_h = 2\,\text{ns}$. [30 pontos]

A especificação temporal do circuito combinacional é: $T_{LC} = 10\,\text{ns}$, $T_{C,C} = 3\,\text{ns}$.

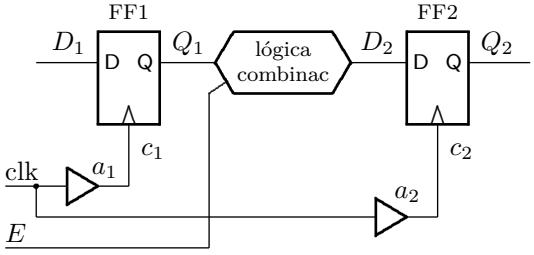
(a) Qual é o período mínimo do relógio?
(b) Há folga no tempo de *hold* do registrador?
(c) Qual deve ser o *setup* mínimo nas entradas?
(d) Qual deve ser o *hold* mínimo nas entradas?



13. Considere o circuito da figura abaixo. A especificação temporal do registrador de estado é: $T_{FF} = 4\,\text{ns}$, $T_{C,F} = 1\,\text{ns}$, $T_s = 2\,\text{ns}$, $T_h = 2\,\text{ns}$. O *skew* é $T_{skew} = -3\,\text{ns}$. [30 pontos]

A especificação temporal do circuito combinacional é: $T_{LC} = 12\,\text{ns}$, $T_{C,C} = 4\,\text{ns}$.

(a) Qual é o período mínimo do relógio?
(b) Há folga no tempo de *hold* do registrador?
(c) Qual deve ser o *setup* mínimo nas entradas?



14. Mostre como implementar um somador completo usando somente uma memória ROM de 8x2bits. A especificação do somador completo é: [30 pontos]

$$s = a + b + vem \qquad vai = a \wedge b \vee a \wedge vem \vee b \wedge vem$$

15. O circuito da figura abaixo é uma *porta de transmissão*. Mostre como implementar um multiplexador de 4 entradas com estes circuitos. [30 pontos]