

**UFPR - DInf - Bacharelado em Ciência da Computação
CI210 - Projetos Digitais e Microprocessadores**

Primeira Prova (duas turmas)

- 1) Projete um circuito combinacional que efetua a comparação de magnitude de dois números positivos de 8 bits. Este circuito tem duas entradas P e Q, de 8 bits cada, e três saídas que são "menor", "igual" e "maior".

```
P,Q: B8
menor,igual,maior: B
menor = num(P) < num(Q)
igual = num(P) = num(Q)
maior = num(P) > num(Q)
```

- 2) Projete um circuito seqüencial síncrono que aceita uma seqüência de números positivos de 8 bits, e a cada novo valor recebido as duas saídas, chamadas de MAX e MIN, mostram os valores máximos e mínimos já observados na seqüência de entrada. [10 pontos]

```
IN: N -> B8
MAX,MIN: B8
all t | MAX >= IN[t]
all t | MIN <= IN[t]
```

- 1) Projete um circuito seqüencial síncrono, com uma entrada de inicialização (init) e 8 saídas, que produza a seguinte seqüência nas saídas. [10 pontos]

```
init 10111111111111111111
z0   x0111111101111111...
z1   x0011111100111111...
z2   x0001111100011111...
z3   x0000111100001111...
z4   x0000011100000111...
z5   x0000001100000011...
z6   x0000000100000001...
z7   x1000000010000000...
```

- 2) A medida em que cresce a densidade dos CIs de memória, maior é a probabilidade de ocorrerem erros internos ao CI de memória (inversão dos bits armazenados). Usando paridade, projete um circuito que seja capaz de detectar a ocorrência de um bit em erro em uma memória. O CI de memória tem largura de 9 bits, mas o processador acessa a memória em palavras de 8 bits. Note que o intervalo entre a escrita em um certo endereço da memória, e a leitura do conteúdo deste mesmo endereço é arbitrariamente longo. [10 pontos]

Segunda Prova

- 1) Deseja-se adicionar um novo modo de endereçamento aos modos disponíveis no Mico. O novo modo é chamado de "indireto a memória" e nele o endereço efetivo é obtido a partir de um indexador armazenado em memória. A instrução LDM r1, [r2(desl)] é definida por
r1 <- M[M[r2+desl]].

a) Desenhe um diagrama de blocos do Mico mostrando claramente as adições necessárias para a execução da instrução LDM. [5 pontos]

b) Desenhe um diagrama de estados desta instrução, indicando *claramente* as ações efetuadas em cada estado. [10 pontos]

- 2) Mostre como implementar um barramento assíncrono para interligar um processador a periféricos. Seu projeto deve definir e descrever as três transações mais importantes que são (1) a transferência de endereços pelo barramento, (2) a leitura, e (3) a escrita pelo processador.

a) Desenhe os diagramas de tempo das três transações. [10 pontos]

b) Desenhe as máquinas de estado, do processador e de um periférico, que controlam as transações no barramento. [15 pontos].

- 1) Deseja-se aumentar a flexibilidade do Mico através da adição de instruções que permitam a manipulação de bytes. Para tanto, são necessárias versões para byte das instruções LOAD e STORE, além de uma instrução similar a SETI que permite carregar uma constante de 8 bits sem estender o sinal.
 - a) Defina cuidadosamente a sintaxe e a semântica das instruções LB e SB (load-byte e store-byte). [5 pontos]
 - b) Desenhe dois diagramas de tempo, um para uma carga do byte no endereço 1024 [LB r1,0(r2), r2=1024], e outro para uma atualização do byte no endereço 1025 [SB r2,0(r3), r3=1025]. [5 pontos]
 - c) Desenhe um diagrama de blocos do sistema de memória do Mico mostrando claramente as modificações necessárias para a execução das instruções LB e SB.
 - d) Qual/quais problema/s com o acesso a palavras de 2-bytes (instruções LD e ST para 16 bits) que podem ocorrer por causa das modificações necessárias para as instruções LB e SB? [5 pontos].
- 2) Um controlador de Acesso Direto a Memória é um dispositivo que efetua cópias entre endereços, como a cópia do conteúdo de uma área em disco para uma área em memória. Quando o processador necessita que uma cópia seja efetuada entre um periférico e memória, o processador deve programar o controlador de ADM com o endereço fonte (periférico), o endereço destino em memória, e o número de palavras a serem transferidas. Projete um controlador de ADM. Seu projeto deve conter um diagrama de blocos do controlador e uma descrição textual do seu algoritmo de funcionamento. [15 pontos].

Exame Final

- 1) Mostre como implementar as instruções CALL e RET no Mico. [50 pontos]

```
CALL ender # Mem[sp--] := PC; PC := ender
RET       # PC := Mem[++sp]
```

 - a) Desenhe um diagrama de blocos completo do Mico acrescentando todas as ligações ou circuitos adicionais que julgar necessário.
 - b) Defina e descreva as seqüências de operações para implementar estas DUAS instruções.
- 2) Projete um circuito combinacional com seis entradas e uma saída que computa a função maioria. A saída fica em 1 sempre que houver maioria de entradas em 1. [20 pontos]
- 3) Descreva o mecanismo de busca antecipada e indique as modificações necessárias no Mico para suportá-lo. Liste as instruções em que a busca antecipada não é viável e justifique cada caso. [30 pontos]
- 1) Projete um circuito que executa a divisão inteira de dois números inteiros positivos de quatro bits. O circuito deve implementar a divisão por subtrações repetidas. Sua resposta deve conter o circuito do divisor e uma descrição completa do seu funcionamento. [30 pontos]
- 2) Mostre como implementar as instruções PUSH e POP no Mico. [50 pontos]

```
PUSH r # Mem[sp--] := r
POP r  # r := Mem[++sp]
```

 - a) Desenhe um diagrama de blocos completo do Mico acrescentando todas as ligações ou circuitos adicionais que julgar necessário.
 - b) Defina e descreva as seqüências de operações para implementar estas DUAS instruções.
- 3) Mostre como implementar um contador Johnson com 4 flip-flops tipo D. Desenhe o diagrama de tempo do contador. [20 pontos]