

EA078 Mini e Microcomputadores: Hardware

Prof. Alice M. Tokarnia

Campinas, 13 de abril de 2010.

Prova 2

Nome/RA:

Questão 1

- 1.1 Compare as duas reduções no número de estados a seguir: (0,6 pt)
 - (a) redução do número de estados feita no projeto de processadores (RT) através da combinação de estados com operações independentes
 - (b) redução do número de estados feita no projeto de circuitos sequenciais.
- 1.2 Trabalhar com números inteiros permite simplificar os componentes de várias formas. Forneça os limites de superior e inferior para a representação complemento de 2's $b_{31}b_{30}\dots b_1b_0$ quando cada uma das condições a seguir é satisfeita: (0,4 pt)
 - (a) $b_{31}.b_{30}.b_{29}.b_{28} = 1$
 - (b) $b_{31} + b_{30} + b_{29} + b_{28} = 0$
- 1.3 Qual o papel das variáveis de status de operação no projeto de processadores dedicados? (0,5 pt)
- 1.4 Por que implementação pipeline de funções e datapaths é frequentemente utilizada? (0,5 pt)

Escolha uma das duas questões a seguir.

Questão 2

Processadores dedicados ou “aceleradores de hardware” são usados para executar funções que necessitam ser executadas rapidamente e com frequência. Divisão de números não negativos é um exemplo.

2.1. Apresente o diagrama ASM com o menor número possível de estados (2 pt) e um datapath correspondente (4 pt) para o algoritmo de divisão de números inteiros não-negativos a seguir. No projeto do datapath, minimize o número de registradores, barramentos e arquivos de registradores.

2.2. Apresente o projeto da unidade de controle. (2pt)

dd e resto tem 64 bits;

divisor e dividendo tem 32 bits;

sinal-erro, sd tem 1 bit;

```
1. while (1) {
2.   while (!go );
3.   ready = 0;
4.   dividendo = IN (dividendo);
5.   divisor = IN (divisor);
6.   If (divisor == 0)
7.     sinal-erro = 1;
8.   else
9.   {
10.    sinal-erro = 0;
11.    dd = divisor << 32; resto = dividendo; quo = 0; sd = 1;
12.    If !(dividendo == 0)
13.    {
14.      For (i=0, 32, i++)
15.      {
16.        If (sd == 1)
17.          resto = resto - dd;
18.        else resto = resto + dd;
19.        quo = quo << 1;
20.        If !(resto < 0 )
21.        { quo = quo + 1; sd = 1 }
22.        else sd = 0;
23.        dd = dd >> 1;
24.      }
25.      ready = 1; saída_quo = OUT (quo); saída_resto = OUT (resto); saída_erro = OUT
        (sinal-erro)
26.    }
```

Questão 3

Você deve implementar um processador dedicado correspondente ao Controlador PID descrito pelo código a seguir (adaptado de Vahid, 2002), considerando que as unidades funcionais a serem usadas são um multiplicador com duas entradas e uma saída (latência = 16 μ s) e um somador/subtrator com duas entradas e uma saída (latência = 2 μ s)

3.1. Apresente o diagrama ASM com o menor número possível de estados (1 pt) e um datapath correspondente (3 pt), minimize o número de registradores, barramentos e arquivos de registradores.

3.2. Apresente o diagrama de tempo com as saídas da unidade de controle. (2 pt).

3.3. Apresente o diagrama de tempo para uma implementação pipeline do datapath com 2 estágios (1,5 pt).

3.4. Compare a produtividade, isto é, o número de valores que podem ser processados por unidade de tempo do projeto pipeline com o projeto original. Compare também os tempos de reposta das duas implementações (0,5 pt).

Double Pgain, Dgain, Igain, PID;

Double reference_value, sensor_value, sensor_value_previous;

Double error, error_sum, difference;

```
1. While (1) {
2.   while (!go );
3.   ready = 0;
4.   If (reset==1)
5.     error_sum = 0;
6.   Pgain = IN (Pgain);           /* ganhos do controlador */
7.   Dgain = IN (Dgain);
8.   Igain = IN (Igain);

9.   reference_value = IN (reference_value); /* valor de referência */
10.  sensor_value = IN (sensor_value);      /* valor */

11.  error = reference_value – sensor_value; /* erro */
12.  Pterm = Pgain * error;

13.  error_sum = error_sum + error;          /* erro acumulado – integral */
14.  Iterm = Igain * error_sum;

15.  difference = sensor_value – sensor_value_previous; /* difference de valores */
16.  sensor_value_previous = sensor_value;
17.  Dterm = Dgain * difference;

18.  PID = Pterm + Iterm + Dterm;           /* saída do controlador */
19.  Out = OUT (PID); ready = 1;
20. };
```