

## ELC1011 – Organização de Computadores

### Trabalho 2

Prof. Giovanni Baratto

✉ Giovanni.Baratto@ufsm.br  
☎ (55) 98116-2420

Entregar as respostas do trabalho, usando a ferramenta Moodle, na data acordada pelo professor. Envie as soluções dos problemas em um arquivo compactado (tipo ZIP). Para cada problema, descreva detalhadamente a solução (use os arquivos no formato PDF), comentando sempre o resultado. Adicione sempre os arquivos fonte usados na solução.

1. Traduza o seguinte programa em C, para o *assembly* do processador MIPS. Compile e execute o seu programa *assembly* usando o programa MARS. Comente bem o seu código. Indique nos comentários o trecho do código em C que está sendo traduzido<sup>1</sup>.

```

1 int valor1 = 10; /** variáveis globais */
2 int valor2 = 20;
3
4 /* retorna a média de x e y */
5 int procedimento3(int x, int y) {
6     int tmp; /* variável local*/
7
8     if (x < y) {
9         tmp = x;
10        x = y;
11        y = tmp;
12    }
13
14    while (x > y) {
15        x = x - 1;
16        y = y + 1;
17    }
18
19    return x;
20 }
21
22 /* retorna a soma da média(x, 10) com média(x, 20) */
23 int procedimento2(int x) {
    vetorAC[i]

```

<sup>1</sup>Após a execução do programa, o valor da variável resultado é 750.

```

24  int resultado; /* variável local */
25  resultado = procedimento3(x, valor1) + procedimento3(x, valor2);
26  return resultado;
27 }
28
29 /* acumulamos o resultado de procedimento2 para x=i*9+20 , i = 0, 1, ..., 9
   */
30 int procedimento1(int x, int y) {
31     int vetorA[10]; /* variáveis locais */
32     int i;
33     int acumulador;
34
35     for (int i = 0; i < 10; i++) {
36         vetorA[i] = i * x + y;
37         vetorA[i] = procedimento2(vetorA[i]);
38     }
39     acumulador = 0;
40     for (int i = 0; i < 10; i++) {
41         acumulador = acumulador + vetorA[i];
42     }
43     return acumulador;
44 }
45
46 /* procedimento principal */
47 int main(void) {
48     int n; /* variáveis locais */
49     int m;
50     int resultado;
51
52     n = 9;
53     m = 20;
54     resultado = procedimento1(n, m);
55     return 0;
56 }

```

Listagem 1: Programa usado no problema 1.

- Sejam os números  $x = 00101_2 = 5$  e  $y = 00110_2 = 6$ . Use uma representação com 5 bits. Faça a multiplicação  $X \cdot Y$ , usando o segundo algoritmo da multiplicação. Apresente os resultados em cada um dos passos do algoritmo.
- Escreva um procedimento `double cos(double x)`, em assembly para o MIPS, para calcular o cosseno de um ângulo  $x$ , dado em radianos. O procedimento calcula o cosseno usando uma série de Taylor expandida em  $x = 0$  (veja a equação 1). No procedimento, trunque a série em  $n = 7$  (até o termo  $\frac{x^{14}}{14!}$ ).

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2 \cdot n)!} x^{2 \cdot n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \frac{x^{12}}{12!} - \frac{x^{14}}{14!} + \dots \quad (1)$$

Crie um programa em assembly para o MIPS. O programa permite a entrada de um ângulo  $x$  em graus ( $^\circ$ ), converte o ângulo para radianos, calcula o cosseno do ângulo usando o procedimento `cos()` e imprime o resultado. Use o programa para calcular o seno de  $57.23^\circ$ . Mostre a saída da execução do seu programa no programa MARS. Verifique se o resultado apresentado pelo seu programa está correto.

4. Represente o número  $x = 462.234\,38$  em ponto flutuante, precisão simples. Mostre os passos na solução deste problema.
5. Qual o valor decimal do número  $x = 0x78787800$ , representado em ponto flutuante, precisão simples. Mostre os passos na solução deste problema.
6. Explique detalhadamente, usando as tabelas 1 e 2 e o diagrama de blocos do processador na figura 1, como a instrução `sw $s0, 40($at)` é executada pelo processador monociclo. Converta a instrução para linguagem de máquina, apresentando os campos. Apresente na figura os sinais de controle. Escreva um texto explicando como a instrução é executada.
7. Explique detalhadamente, usando as tabelas 1 e 2 e o diagrama de blocos do processador na figura 1, como a instrução `bne $a0, $a1, loop` é executada pelo processador monociclo. O endereço desta instrução é `0x00400034` e `loop` é um rótulo para o endereço `0x00400018`. Converta a instrução para linguagem de máquina, apresentando os campos. Apresente na figura os sinais de controle. Escreva um texto explicando como a instrução é executada.
8. Explique detalhadamente, usando as tabelas 1 e 2 e o diagrama de blocos do processador na figura 1, como a instrução `j loop` é executada pelo processador monociclo. O endereço desta instrução é `0x00400034` e `loop` é um rótulo para o endereço `0x00400014`. Converta a instrução para linguagem de máquina, apresentando os campos. Apresente na figura os sinais de controle. Escreva um texto explicando como a instrução é executada.

Tabela 1: Instruções e valores dos sinais na unidade de controle do processador monociclo.

Controle	Sinal	Formato R (0)	lw (35)	sw (43)	beq (4)
Entradas	OP5	0	1	1	0
	OP4	0	0	0	0
	OP3	0	0	1	0
	OP2	0	0	0	1
	OP1	0	1	1	0
	OP0	0	1	1	0
Saídas	RegDst	1	0	X	X
	UALFonte	0	1	1	0
	MemParaReg	0	1	X	X
	EscReg	1	1	0	0
	LerMem	0	1	0	0
	EscMem	0	0	1	0
	DvC	0	0	0	1
	UALOp1	1	0	0	0
	UALOp0	0	0	0	1

Tabela 2: Operação da ULA para a combinação de UALOp e o campo de função.

UALOP		Campo de Função							
UALOp1	UALOp0	F5	F4	F3	F2	F1	F0	Operação da ULA	
0	0	X	X	X	X	X	X	0010	soma
X	1	X	X	X	X	X	X	0110	subtração
1	X	X	X	0	0	0	0	0010	soma
1	X	X	X	0	0	1	0	0110	subtração
1	X	X	X	0	1	0	0	0000	and
1	X	X	X	0	1	0	1	0001	or
1	X	X	X	1	0	1	0	0111	slt

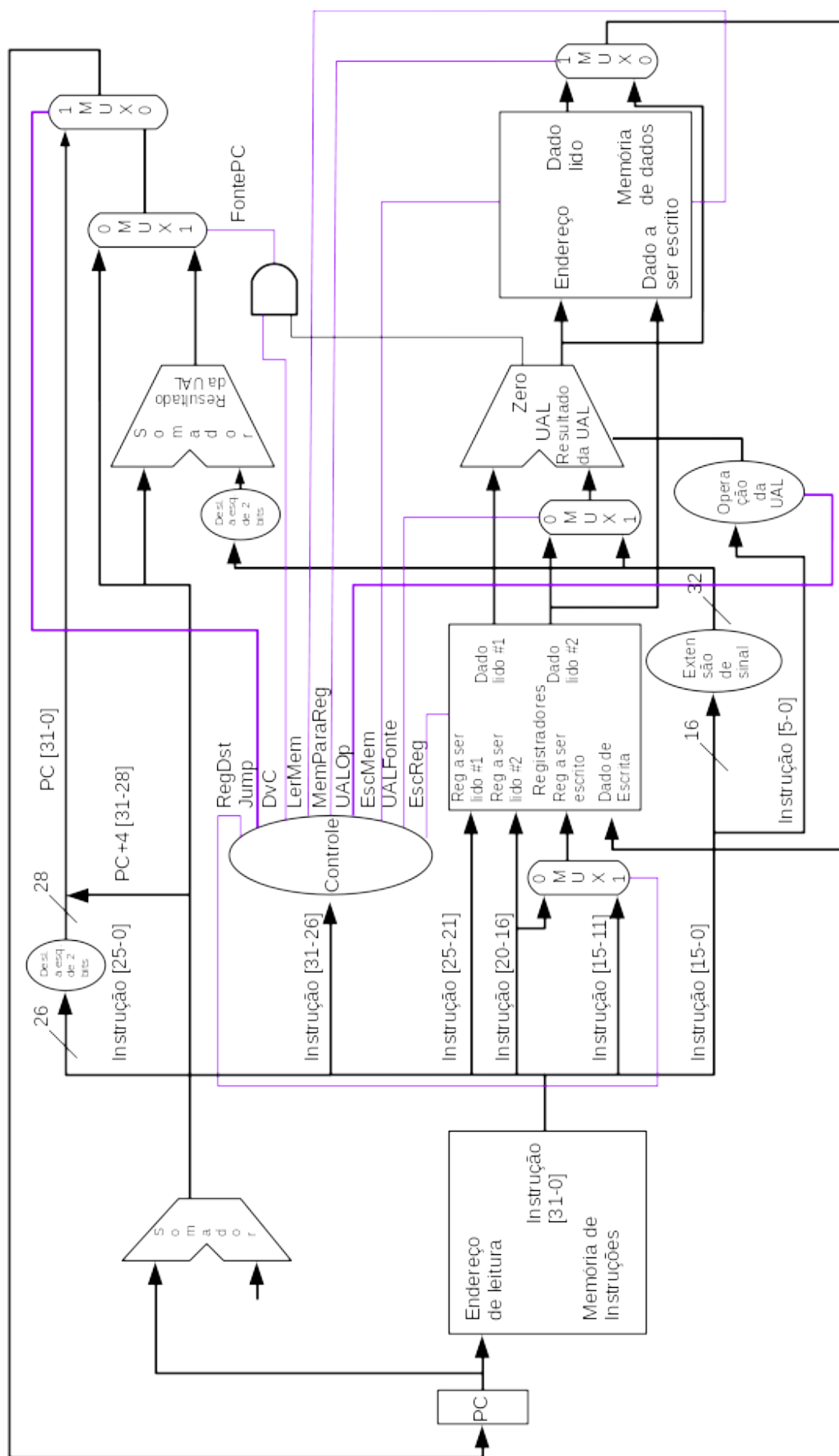


Figura 1: Diagrama de blocos do processador monociclo