Prova P4 **Disciplina: Organização e Arquitetura de Computadores I**
T148 **Professor: Ney Laert Vilar Calazans**
Aluno: 03/Julho/2006

1. **[2 pontos] Responda se as afirmações abaixo são verdadeiras ou falsas. Todas são relativas ao processador Cleópatra e sua organização, conforme descrito na documentação de aula.**

( ) **Uma microinstrução leva de 1 a 3 ciclos de relógio para executar.**
( ) **Uma microinstrução é um conjunto de microoperações, todas as quais são executadas simultaneamente em um único ciclo de relógio.**
( ) **A cada ciclo de relógio, os flip-flops qualificadores N, Z, C, e V são escritos com algum valor.**
( ) **Os registradores MAR, MDR, AC, IR, RS e PC são elementos seqüenciais, enquanto a ALU ou ULA é um elemento combinacional.**
( ) **Todas as maneiras possíveis de realizar a busca de uma instrução em qualquer organização da arquitetura Cleópatra implicam gastar pelo menos três ciclos de relógio para esta ação.**

2. **[5 pontos] Considere o programa abaixo, escrito em linguagem de montagem do processador Cleópatra. Considere que o programa é executado sobre a organização proposta na descrição do processador mostrada na disciplina. Produza o código objeto do programa, mostrando a área de programa e a área de dados (1,5 pontos). Indique claramente os endereços em que cada palavra do código objeto se encontra. Considere que uma implementação do processador utiliza um relógio de 25MHz e que executa este programa. Calcule quanto tempo leva a execução do programa com a área de dados fornecida (1,5 pontos). Note que para calcular o tempo de execução, o passo mais importante é calcular quantos ciclos de relógio leva para executar o programa inteiro. Descreva claramente o desenvolvimento dos cálculos, de forma que erros triviais que produzam um resultado final incorreto não impeçam a correção pelo professor. Cuidado com os modos de endereçamento usados no programa. Diga o que faz este programa. (1 ponto) Existe pelo menos uma condição dos dados de entrada que faz este programa funcionar incorretamente. Descreva uma destas condições (1 ponto).**

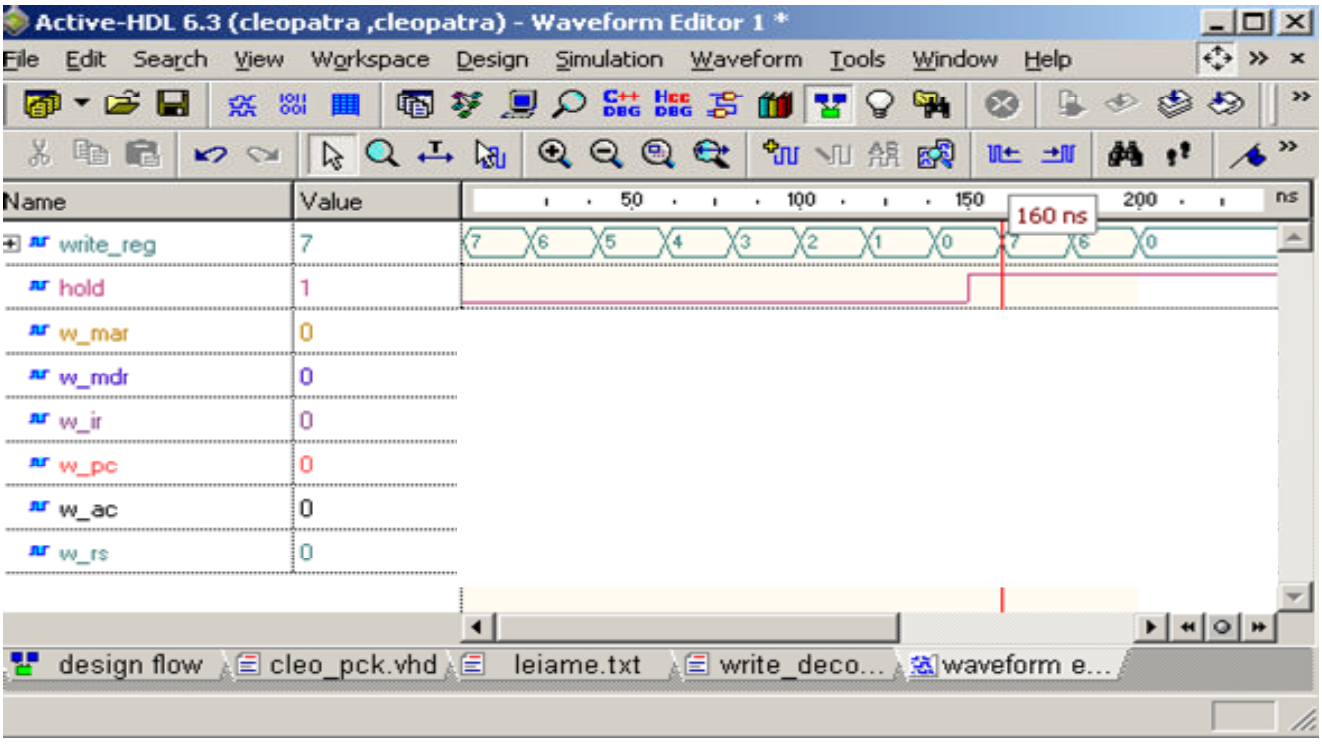
Cleópatra	Código Objeto
.CODE	
repete: lda n,R	
jz fim	
lda c	
not	
add #1	
add pvet,I	
sta pvet,I	
lda n	
add #0ffh	
sta n	
lda pvet,R	
add #01h	
sta pvet	
jmp repete	
fim: hlt	
.ENDCODE	
.DATA	
INI: db #03h	
db #08h	
db #3bh	
db #0f5h	
db #0f6h	
n: db #05h	
pvet: db INI	
c: db #01h	
.ENDDATA	

3. **[3 pontos] O código abaixo representa a implementação VHDL do decodificador de escrita da organização Cleópatra vista em aula. Implemente um testbench em VHDL que seja capaz de gerar a forma de onda dada abaixo do código (2 pontos). Complete as formas de onda, simulando manualmente o circuito com o seu testbench e desenhando as formas de onda resultantes (1 ponto).**

```
----------------------------------------------------------------------
-- Decodificador de Escrita
-- Este bloco combinacional gera sinais de habilitação de escrita nos
-- registradores do Bloco de Dados.
----------------------------------------------------------------------
library IEEE;
use IEEE.Std_Logic_1164.all;
use work.cleo.all;

entity write_decoder is
    port(write_reg :in std_logic_vector(2 downto 0); hold :in
std_logic;
     w_mar, w_mdr, w_ir, w_pc, w_ac, w_rs :out std_logic);
end write_decoder;

architecture write_decoder of write_decoder is
begin
    -- Códigos: 0:MAR, 1:MDR, 2:IR, 3:PC, 4:AC, 5:RS, 6:PC/MDR, 7:nada
    w_mar <= '1' when hold='0' and write_reg="000"   else '0';
    w_mdr <= '1' when hold='0' and
               (write_reg="001" or write_reg="110")  else '0';
    w_ir  <= '1' when hold='0' and write_reg="010"   else '0';
    w_pc  <= '1' when hold='0' and
               (write_reg="011" or write_reg="110")  else '0';
    w_ac  <= '1' when hold='0' and write_reg="100"   else '0';
    w_rs  <= '1' when hold='0' and write_reg="101"   else '0';
end write_decoder;
```



GABARITO

1. **QUESTÃO 1:**

(F) **Uma microinstrução leva de 1 a 3 ciclos de relógio para executar.**

Errado, por definição uma microinstrução dura exatamente 1 ciclo de relógio.

(V) **Uma microinstrução é um conjunto de microoperações, todas as quais são executadas simultaneamente em um ciclo de relógio.**

(F) **A cada ciclo de relógio, os flip-flops qualificadores N, Z, C, e V são escritos com algum valor.**

Errado, os sinais Inz e Icv existem para dizer em que ciclo os flagas devem ou não ser escritos com o valor que a ULA gera a cada ciclo de relógio.

(V) **Os registradores MAR, MDR, AC, IR, RS e PC são elementos seqüenciais, enquanto a ALU ou ULA é um elemento combinacional.**

(F) **Todas as maneiras possíveis de realizar busca de instruções em qualquer organização da arquitetura Cleópatra implicam gastar pelo menos três ciclos de relógio para esta ação.**

Errado, vimos em aula mais de uma organização em que se pode fazer a busca em 2 ou até mesmo em apenas 1 ciclo de relógio (ver, por exemplo, gabarito da P2)

2. **QUESTÃO 2:**

a. Código Objeto/Ciclos de relógio

.CODE		Ciclos	Endereço	Código
repete:	lda n,R	8	00	4C 1F
	jz fim	5/6	02	B4 1B
	lda c	8	04	44 23
	not	4	06	00
	add #1	6	07	50 01
	add pvet,I	10	09	58 22
	sta pvet,I	9	0B	28 22
	lda n	8	0D	44 21
	add #0ffh	6	0F	50 FF
	sta n	7	11	24 21
	lda pvet,R	8	13	4C 0D
	add #01h	6	15	50 01
	sta pvet	7	17	24 22
	jmp repete	6	19	84 00
fim:	hlt	4	1B	F0
.ENDCODE				
.DATA				
INI:	db #03h		1C	03
	db #08h		1D	08
	db #3bh		1E	3B
	db #0f5h		1F	F5
	db #0f6h		20	F6
n:	db #05h		21	05
pvet:	db INI		22	1C
c:	db #01h		23	01
.ENDDATA				

b. Tempo de execução

Desenvolvimento:

Laço: (8+5+8+4+6+10+9+8+6+7+8+6+7+6)*N + 8+6+4

Fórmula: 98 * N + 8 + 6 + 4=98 *n + 18

Para n=5 número de ciclos de relógio=508

Se o clock é de 25MHz, o período é de 1/(25*106)= 40ns para 1 ciclo de relógio. O tempo de execução do programa é então 508*40ns = 20320ns ou 20,32 microssegundos, ou 0,02032ms uu 0,00002032 segundos.

c. Funcionalidade/Erros possíveis

O programa subtrai a constante contida no endereço c de todos os elementos de um vetor

Os erros possíveis decorrem da forma como se faz a subtração, invertendo o sinal da constante e adicionando o resultado a cada elemento do vetor. Se a constante for o valor hexa 80H, o processo de inversão do sinal produz um resultado errado, pois não existe positivo correspondente a 80H em complemento de 2 com 8 bits. Outras situações que podem gerar erros são o valor de n não ser o tamanho exato do vetor.

3. QUESTÃO 3: Em preto, parte adicionada pelo usuário, caso utilize o gerador automático de TBs

```
library ieee;
use work.cleo.all;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity write_decoder_tb is
end write_decoder_tb;
architecture TB_ARCHITECTURE of write_decoder_tb is
      component write_decoder
      port(write_reg : in std_logic_vector(2 downto 0); hold : in std_logic;
w_mar : out std_logic;
            w_mdr : out std_logic; w_ir : out std_logic;    w_pc : out std_logic;
            w_ac : out std_logic;   w_rs : out std_logic );
      end component;
      signal write_reg : std_logic_vector(2 downto 0):="111";
      signal hold : std_logic;
      signal w_mar : std_logic;
      signal w_mdr : std_logic;
      signal w_ir : std_logic;
      signal w_pc : std_logic;
      signal w_ac : std_logic;
      signal w_rs : std_logic;
begin
      UUT : write_decoder port map (write_reg => write_reg,
                  hold => hold,     w_mar => w_mar, w_mdr => w_mdr,
                  w_ir => w_ir,     w_pc => w_pc, w_ac => w_ac, w_rs => w_rs);
      hold <= '0', '1' after 150ns;
      process
      begin wait for 20ns;
       if write_reg="000"
       then write_reg <= (others=>'1'); else write_reg <= write_reg-1; end if;
      end process;
end TB_ARCHITECTURE;
```