

**PRÉ-PROJETO: CALCULADORA PROGRAMÁVEL.**

**Objetivo:** O objetivo deste pré-projeto é apresentar alguns conceitos/estratégias de projeto úteis para o projeto final da disciplina.

**1. Preparo**

- a) Faça uma descrição funcional do microcomputador.
- b) Projete, com uso de portas *tri-states*, uma **memória ROM** de 8 palavras de comprimento igual a 3 bits D0D1D2 e cada palavra P<sub>i</sub> na memória é endereçável por 3 bits E0E1E2, conforme ilustra a Figura 1. Explique o funcionamento desta memória ROM. Liste a saída para todas as possíveis combinações E0E1E2.

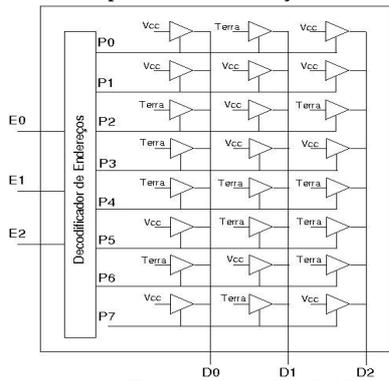


Figura 1: Memória ROM.

- c) Nas Experiências 4 e 5 você teve que coordenar manualmente, por meio de botoeiras, uma sequência de operações para realizar uma tarefa específica. Considere agora a possibilidade de armazenar os códigos de operação desta sequência numa memória ROM e projete uma máquina de estados que controle a execução na sequência programada. Em cada estado da máquina ocorre a decodificação e execução de uma operação, como ilustra a Figura 2. Essa máquina de estados é conhecida por **contador de programa (CP)**. No circuito mostrado, cada código de operação de 3 bits é armazenado em uma posição da memória endereçável pelo CP e os valores (A,B,C) são pré-definidos por um conjunto de 3 chaves.

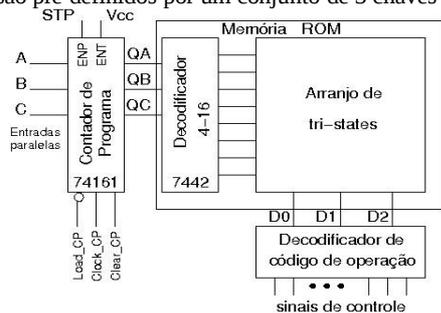


Figura 2: Contador de Programa.

Explique o funcionamento do circuito da Figura 2, ressaltando a tabela-verdade do CI 7442, a duração de cada estado em termos de um relógio do sistema CLK, e os valores dos sinais de controle Load\_CP, Clock\_CP, Clear\_CP, STOP para as seguintes situações:

- c.1) Como se pode inicializar o contador com o endereço 000?
- c.2) Quando o CP deve mudar de estado? Na borda de subida ou descida do CLK? E quando os sinais de controle devem ser executados dentro de um período do CLK?
- c.3) Uma instrução JMP (A,B,C) (*jump*) carrega o valor ABC no CP de forma que o próximo código de operação a ser executado seja o armazenado em ABC. Como?
- c.4) Como se pode parar a execução do circuito?
- d) Observe atentamente o circuito da Figura 3, no qual foram utilizados CI 74181 e CI 74182 para implementar um somador de 8 bits ( $S_3S_2S_1S_0=1001$  e  $M=0$ ). Explique o seu funcionamento, destacando funções lógicas que relacionam os sinais de saída G,P

- em relação às entrada X e Y do CI 74181 e funções lógicas que relacionam os sinais Cn+X e Cn+Y com as entradas do CI 74182.
- e) Simule os circuitos no ambiente Quartus II para verificar o seu funcionamento. Registre os testes realizados.

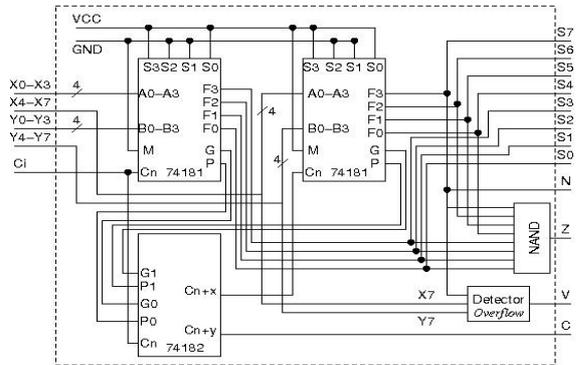


Figura 3: Somador.

<b>RA:</b>	<b>Visto:</b>	<b>Data:</b>
------------	---------------	--------------

**2. Calculadora Programável**

Considere a seguinte organização de entrada e saída da UA, provida de um contador de programa e um registrador para armazenar os bits de condição (C,V,Z,N). Substitua ainda o somador da Figura 1 da Experiência 5 pelo circuito da Figura 3.

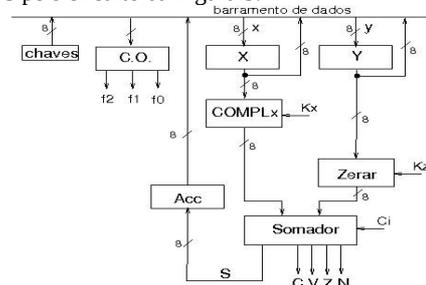


Figura 3: Circuito UA.

Projete um novo decodificador de sinais de controle, apropriado para esta nova organização. O formato dos códigos segue o mesmo padrão definido na Experiência 3. Reutilize os circuitos de periféricos projetados na Experiência 5.

Código de Operação	Operação	
	f2 f1 f0	
	ADD	Acc = x+y
	EXSUB	Acc = -x+y
	CMX	Acc=x'
	MOVX	X <- (Acc)
	MOV X,Y	y <- (x)
	LDX	x <- (chaves)
	JMP(A,B,C)	Desvio para (A,B,C)
	STP	Pára a execução

<b>RA:</b>	<b>Visto:</b>	<b>Data:</b>
------------	---------------	--------------

**3. Periféricos e Programação**

Programa a memória ROM para executar as seguintes tarefas: 15+15+24, (35-20+5)\*2; 16x11; led correspondente ao bit de condição Z fique piscando,

<b>RA:</b>	<b>Visto:</b>	<b>Data:</b>
------------	---------------	--------------