EA773 Laboratório de Circuitos Lógicos

FEEC-UNICAMP - 2º Semestre de 2011 – Turmas C, D, S, T, U e W

Turma:_____ Grupo:_____ Data:_____

RA:_____ Nome:_____

RA:_____ Nome:_____

PROJETO: CALCULADORA PROGRAMÁVEL.

revisada pela Profa.Ting

Objetivo: O objetivo deste projeto é desenvolver uma calculadora com execução sequencial automática de instruções (arquitetura Von Neumann). As instruções são de tamanho de 12 *bits*. Com isso, pode-se "personalizar" um circuito digital de propósito-geral para executra uma tarefa específica, ou seja, programá-la.

2. Especificação

Projete, com uso das funções lógicas dos componentes 74181 e 74182, uma calculadora programável operando a uma frequência de 1.5 Hz. Esta calculadora dispõe de uma memória de 2^4x12 *bits* para armazenar as instruções de um programa, três registradores de dados de 8 *bits*, X, Y e Acc, e um registrador de instrução de 12 *bits*. *Displays* de 7 segmentos são utilizados para visualizar o conteúdo, em decimal, do registrador Acc, e um *led* para visualizar o *bit* menos significativo do registrador X. Há ainda uma botoeira *Reset* que, quando acionado, limpa o contador de programa CP, ou seja, (CP)←0, e 4 *leds* para visualizar o conteúdo do CP. O conjunto de instruções que ele suporta é listado na Tabela 1.

Tabela 1: Código de Operação.

Código (12 bits)		Mnemô-nica	Descrição
MSB	LSB		
10000000	(Nº.grupo+11)%16	ADD	Acc ← (X) + (Y)
10000000	(Nº.grupo+10)%16	SUB	Acc ← (X) - (Y)
10000000	(Nº.grupo+9)%16	CMX	Acc ← (X)'
10000000	(Nº.grupo+8)%16	INC	Acc ← (X)+1
10000000	(Nº.grupo+7)%16	DEC	Acc ← (X)-1
10000000	(Nº.grupo+6)%16	OR	Acc ← (X)v(Y)
10000000	(Nº.grupo+5)%16	AND	Acc ← (X)^(Y)
10000000	(Nº.grupo+4)%16	XOR	Acc ← (X)x(Y)
10001111	(Nº.grupo+3)%16	CLR	Acc ← 0
10001111	(Nº.grupo+2)%16	MY	Y ← (Acc)
10001111	(Nº.grupo+1)%16	MX	X ← (Acc)
1010	OPERANDO	LDX	X ← OPERANDO
1011	OPERANDO	LDY	Y ← OPERANDO
11010011	(Nº.grupo)%16	STP	CP ← (CP)
11010000	ENDEREÇO	JMP	CP ← ENDEREÇO
11010001	ENDEREÇO	JZ	CP ← ENDEREÇO se Z=1
11010010	ENDEREÇO	JV	CP ← ENDEREÇO se V=1
1111		NOP	nenhuma ação

Novos módulos devem, portanto, ser integrados à organização da calculadora da Experiência 5 para suportar as novas funcionalidades, conforme os destaques na Figura 1.

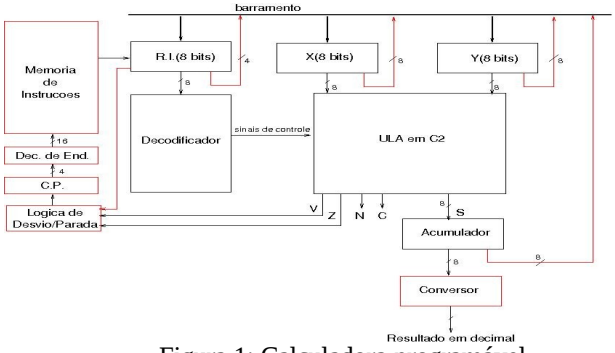

Figura 1: Calculadora programável

3. Controlador Programável

Figura 2 ilustra os componentes básicos que viabilizam a execução automática de uma sequência de instruções previamente armazenadas na memória. Tipicamente existe um **contador de programa** (CP), que especifica o endereço $A_3A_2A_1A_0$ da próxima instrução a ser processada, e um **registrador de instruções** (RI), onde a instrução lida é temporariamente armazenada. Para teste do circuito, codifique o seguinte programa de 7 instruções, desenvolvido pelo André Luiz Garcia Reis, armazene-a na "memória" e verifique se os sinais nos pinos D_{11} ...$D_6D_5D_4D_3D_2D_1D_0$ condizem com os programados à medida que o contador mude do estado 0000 ao estado 0111:

LDX 0, LDY 1, ADD, JV 9, MX, JMP 2, STP

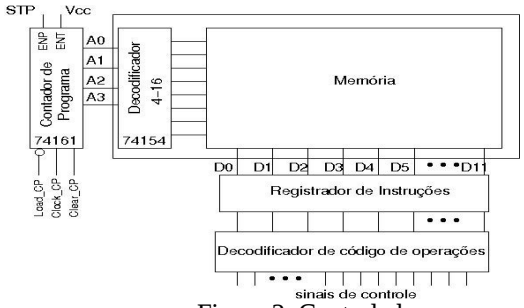

Figura 2: Controlador

Dica: Utilize VHDL para descrever o conteúdo da "memória" ROM. O componente seria um decodificador que tem como entrada 4 *bits* de endereço e saída 12 *bits* de dados.

RA:	Visto:	Data:

4. ULA

Utilize componentes 74181 e 74182 para implementar a unidade lógio-aritmética. Determine, quando pertinente, os sinais de controle $S_3S_2S_1S_0$, M para cada instrução do repertório da Tabela 1. Simule e verifique se o comportamento do seu circuito está condizente com a expectativa para todas as instruções lógico-aritméticas da Tabela 1. Não se esqueça de incluir um registrador para armazenar os valores dos *bits* de condição CNVZ.

RA:	Visto:	Data:

5. Sinais de Controle e Decodificador

Observe que há 4 grupos de instruções:

a. Instruções Lógico-Aritméticas: ADD, SUB, ADDC, INC, DEC, CLR, XOR, AND, OR, NEG → Derivar os sinais de seleção $S_3S_2S_1S_0$ e M para controlar ALU 74181.

b. Instruções de Carregamento: MX, MY, LDX, LDY → Derivar os sinais de habilitação dos registradores conforme Exp. 5..

c. Instrução de Parada: STP → CP deve ser desabilitado.

d. Instruções de Jump: JMP, JZ, JV → Conteúdo do CP deve ser alterado através do sinal Load_CP.

Projete os sinais de controle associados a cada instrução do repertório da Tabela 1.

Implemente, simule e valide o comportamento do decodificador.

RA:	Visto:	Data:

6. Ajustes finos nos sinais de controle

As instruções de carregamento são as que utilizam o barramento para transferência de dados. Ajuste os sinais de habilitação do "transmissor" e os de "receptor", de tal forma que a duração do primeiro seja maior que a duração do segundo. Simule o circuito ajustado.

RA:	Visto:	Data:

7. Integração

Integre todos os módulos desenvolvidos e programe a sua calculadora com os seguintes 3 programas (em 3 "memórias") para testá-la:

•Um pisca-pisca operando a 0.5 Hz.

•Uma sequência de números ímpares até estouro da capacidade de representação.

•Uma sequência de números em que cada um é o resultado de XOR dos dois antecessores, decrementado de 1, ou um programa seu.

RA:	Visto:	Data: