

PROJETO: MICROCOMPUTADOR.

Objetivo:

O objetivo deste projeto é desenvolver um microcomputador de 8 bits, com execução sequencial automática de instruções (arquitetura Von Neumann) (Referência: Capítulo 8, parte A, do livro “Circuitos Lógicos: Teoria e Laboratório”, D. Camilo, J. B. T. Yabu-uti e Y. Yano. LTC, 1984. Uma cópia eletrônica deste texto está disponível na página da disciplina)

Especificação:

Projete, com uso das funções lógicas do circuito integrado ALU 74181, um microcomputador de 8 bits. Diferentemente da memória de 8x3 bits do Pré-projeto, este microcomputador dispõe de uma memória de 2⁴x8 bits para armazenar as instruções de um programa. As instruções são codificadas na seguinte forma

Bit	7	6	5	4	3	2	1	0
	0	INSTRUÇÃO						
	1	INST.			OPER/END			

A bit 7 indica a forma de codificação: quando é igual a 0, os restantes 6 bits contêm um código de instrução/operação, e quando é igual a 1, os restantes 6 bits são divididos em 1 campo de instrução (2 bits) e 1 campo de operando/endereço (de tamanho até 5 bits). O conjunto de instruções que o microcomputador suporta é listado na Tabela 1.

Código (8 bits)		Função	Descrição
0xxx	(0 + grupo) %16	ADD	(Ac) ← (A) + (B)
0xxx	(1 + grupo) %16	SUB	(Ac) ← (A) - (B)
0xxx	(2 + grupo) %16	ADDC	(Ac) ← (A) + (B) + Carry
0xxx	(3 + grupo) %16	INC	(Ac) ← (A) + 1
0xxx	(4 + grupo) %16	DEC	(Ac) ← (A) - 1
0xxx	(5 + grupo) %16	MOVA	(A) ← (Ac)
0xxx	(6 + grupo) %16	MOVB	(B) ← (Ac)
0xxx	(7 + grupo) %16	MOVC	(Ac) ← (A)
0xxx	(8 + grupo) %16	CLR	(Ac) ← 0
0xxx	(9 + grupo) %16	XOR	(Ac) ← (A^B)
0xxx	(10 + grupo) %16	NEG	(Ac) ← (~A)
0xxx	(11 + grupo) %16	STP	(CP) ← (CP)
1000	OPERANDO	LDA	(A) ← OPERANDO(3...0)
1001	OPERANDO	LDB	(B) ← OPERANDO(3...0)
1010	ENDEREÇO	JMP	(CP) ← (ENDEREÇO[3...0])
1011	OPERANDO	BZ	(CP) ← (CP) + (OPERANDO[3...0]) se (Ac) = 0
1100	OPERANDO	BEQ	(CP) ← (CP) + (OPERANDO[3...0]) se (A) = (B)

Tabela 1: Instruções.

O microcomputador possui três registradores de dados de 8 bits, A, B e Ac. Eles são, respectivamente, equivalentes aos registradores X, Y e Acc do Pré-projeto.

Um banco de displays de 7 segmentos é utilizado para visualizar o conteúdo, em decimal, do registrador Ac. Há ainda uma botoeira Reset que, quando acionado, limpa o CP, ou seja, (CP)←0.

Novos Problemas e Sugestão de Solução:

1. As instruções de desvio condicional, BZ e BEQ, requerem

uma soma entre o conteúdo de CP e o valor extraído do campo de instrução (saída do decodificador). Você pode aproveitar ALU para fazer esta soma, guardar o resultado em Ac e carregá-lo em 74161 (ver Figura 2 do Pré-projeto). Para isso, deve-se prever um caminho de dados do decodificador e do CP para ALU.

2. Os registradores A e B podem receber dados do acumulador Ac (MOVA e MOVB) ou da memória (LDA e LDB). Se for diretamente da “memória” via barramento, é necessário introduzir um circuito de controle de acesso da saída do decodificador de instrução ao barramento para evitar **contenção**. Outra alternativa seria transferir o valor para Ac, via ALU, e, em segundo momento, de Ac para A ou B. Neste caso, a entrada do Ac deve estar habilitada para receber ou os dados de saída da ALU ou os dados da memória.

Projeto de Decodificador:

Observe que há 4 grupos de instruções e cada grupo de instruções demandará uma estratégia diferente viabilizando sua implementação.

a. Instruções Lógico-Aritméticas: ADD, SUB, ADDC, INC, DEC, CLR, XOR, NEG → Derivar os sinais de seleção S₃S₂S₁S₀ e M para controlar ALU 74181. No Pré-projeto, o valor destes sinais foi 1001.

b. Instruções de Carregamento: MOVA, MOVB, LDA, LDB → Derivar os sinais de habilitação C₂, C₃, C₄, C₅, C₆ que você viu no Pré-Projeto e o sinal de habilitação de Ac.

c. Instrução de Parada: STP → CP deve ser desabilitado, como no Pré-projeto.

d. Instruções de Jump: JMP, BZ, BEQ → Conteúdo do CP deve ser alterado através do sinal Load_CP, similar ao Pré-projeto.

RA:	Visto:	Data:
------------	---------------	--------------

Programação:

Implemente um roteiro de testes considerando cada uma das instruções. Planeje os testes necessários para que cada instrução possa ter seu comportamento devidamente validado. Observe que algumas instruções (e.g. instruções condicionais) precisarão de mais de um teste para ter seu comportamento validado. Implemente dois programas demonstrativos do funcionamento do circuito, utilizando (considerando os dois programas) todas as instruções, de forma que um deles tenha pelo menos um loop, controlado pela instrução BZ e o outro pela instrução BEQ. Cada programa deve ter pelo menos 8 instruções diferentes.

RA:	Visto:	Data:
------------	---------------	--------------

Relatório

O relatório do projeto final deve conter descrição sucinta do desenvolvimento do projeto, incluindo diagrama de estados, os mapas de Karnaugh; esquema lógico-elétrico (impresso da captura esquemática, com a identificação dos pinos dos periféricos); simulação lógica (ilustrando as situações típicas); análise temporal; e esquema de testes. (descrição dos testes realizados).