

## Projeto de um Computador

### PREPARATÓRIO

Projete um computador com as seguintes características:

- palavra de 8 bits: barramento de dados (D0-D7).
- 8 sinais de endereço: barramento de endereço (A0-A7).
- espaço de endereçamento direto de 256 posições de memória
- instrução ocupa 1 palavra ou  $n$  palavras consecutivas de 8 bits na memória. Formato de instrução:

<b>opcode</b>
<b>operando 1</b>
....
<b>operando n</b>

- as instruções implementadas devem ser no mínimo 8 (das 256 possíveis) e incluir as instruções mostradas abaixo:

Inst.	Código	Operand	Operação	Comentário
<b>load</b>	0000	<i>addr</i>	$[AC] \leftarrow [MS(addr)]$	load do acumulador c/ conteúdo da memória
<b>store</b>	0001	<i>addr</i>	$[MS(addr)] \leftarrow [AC]$	armazena conteúdo do acumulador na memória
<b>jz</b>	0010	<i>addr</i>	$PC \leftarrow addr, \text{ if } Z=1$	salto condicional
<b>ldi</b>	0011	<i>data</i>	$[AC] \leftarrow \text{data (8bits)}$	load imediato do acumulador
<b>dec</b>	0100	<i>addr</i>	$[MS(addr)] \leftarrow [MS(addr)] - 1$	decrementa conteúdo da memória
<b>jp</b>	0101	<i>addr</i>	$PC \leftarrow addr$	salto incondicional
<b>call</b>	0110	<i>addr</i>	empilha PC, $[PC] \leftarrow addr$	chamada de sub-rotina
<b>ret</b>	0111		$[PC] \leftarrow \text{topo da pilha}$	retorno de sub-rotina
.....				

- 1- Faça um esboço da arquitetura interna da CPU especificando seus componentes (registros, ALU e data path).
- 2- A CPU deve possuir portas de entrada e saída para **conectar teclado e display ASCII. Inclua instruções de I/O** (memory mapped ou explícitas) de modo a testar o computador com uma aplicação simples.
- 3- Usando a linguagem RTL, detalhe as operações de controle para a execução do fetch e de cada instrução. A partir daí, especifique os sinais de controle necessários.
- 4- Implemente a Unidade de Controle com lógica aleatória (controle fixo) ou microprogramação.
- 5- Caso utilize um "ripple counter" para implementar o gerador dos estados  $T_i$ , evite a geração de estados espúrios utilizando o sinal clock="0" (p/contador sensível a transição positiva) ou clock="1" (...transição negativa), como "enable" do decodificador de estados.
- 6- Crie o sinal externo RESET que coloca o computador em um estado conhecido (ex: PC=0000 etc).
- 7- Projete e acrescente um **painel de operação** computador contendo a chave RESET, o display ASCII, e um display hexadecimal de 2 dígitos para as seguintes entidades de sua máquina: PC, IR, AC, bus Di, bus Ai, SP. Este painel será usado para a apresentação do projeto.

### EXECUÇÃO

- 1- Simule e teste o projeto do computador executando todas as instruções da máquina isoladamente.
- 2- Programe o computador para este funcionar como uma máquina que, ao se acionar uma tecla alfa-numérica qualquer, comanda a visualização no display do texto (32 letras) pré-armazenado em memória (PROM), a partir da primeira ocorrência do código ASCII da tecla acionada, até o final do texto. Se o código não é encontrado no texto, o programa deve mostrar a mensagem: ??
- 3- Utilize PROM nas primeiras posições da memória principal do computador para armazenar, de modo permanente, o programa de teste, evitando digitá-lo cada vez que executar o simulador. Áreas de dados devem ser implementadas em RAM.