

Laboratório 4 - CPU

Projete um computador com as seguintes características:

- palavra de 8 bits
- espaço de endereçamento de 256 posições de memória
- instrução ocupa **1 ou 2 palavras consecutivas** de 8 bits na memória. Formato de instrução:

Opcode (8 bits)
Operando (8 bits)

- Implemente “um mínimo” de instruções que julgar necessário, sejam estas da tabela abaixo ou outras a sua escolha:

	Inst.	Opcode	Operand	Comentário
1	adi	00000000	<i>data</i>	Soma imediata de data ao acumulador
2	bra	00000001	<i>addr</i>	Salto incondicional para addr
3	brz	00000010	<i>addr</i>	Salto para addr se Z=1
4	ret	00000011		Retorno de subrotina
5	call	00000100	<i>addr</i>	Chamada de sub-rotina
6	move +	00000101		Carrega acumulador com o conteúdo da memória apontada pelo registro A0; incrementa A0
7	store +	00000110		Armazena o conteúdo do acumulador na memória apontada pelo registro A0; incrementa A0
8	out	00000111	<i>#port</i>	Transfere conteúdo do acumulador para a porta de saída #port
9	in	00001000	<i>#port</i>	
10	lea A0	00001001	<i>addr</i>	Carrega A0 com o endereço imediato
11	load	00001010	<i>addr</i>	Carrega acumulador com o conteúdo da memória apontada por addr
12	store	00001011	<i>addr</i>	Armazena o conteúdo do acumulador na memória apontada por addr
13	00001100		

A arquitetura do data path da CPU é fornecida.

- 1- Todas as entidades abaixo são de **8 bits**:
 - Acumulador (**AC**)
 - Programa Counter (contador) (**PC**)
 - Stack Pointer (**SP**), se necessário
 - Instruction Register (**IR**)
 - Registro de endereçamento (contador) (**A0**), se necessário
 - Programa Counter (contador) (**PC**)
 - Portas de **Entrada e Saída** (não necessita registro): **#PORT = 0H de saída** cujo conteúdo deve ser visualizado no display LCD da FPGA, **#PORT = 1H de entrada** que dá acesso ao teclado 4x4 da FPGA.
 - Address Register (**A0**), se necessário
 - Registro Interno Auxiliar **MBR** auxiliar (não visível ao usuário), se necessário
- 2- **Barramentos**: 2 barramentos de 8 bits (**D0-D7**) e (**B0-B7**).
- 3- **ALU**: ALU ou FULL-ADDER de 8 bits: Entrada 1 = Acumulador; Entrada 2 = (D0-D7); Saída = (B0-B7).
- 4- **Acesso aos barramentos**: todos os registros **podem** ser acessados para leitura e/ou escrita pelos 2 barramentos; **implemente apenas os acessos que forem necessários no seu projeto.**
- 5- **Memória RAM** de 8 bits **para código, dados e pilha**. Ligação aos barramentos: Dados (I/O) = (D0-D7); Endereço = (B0-B7).
- 6- O sinal de **RESET** deve fazer **PC=00H, AC=00H e SP=FFH**.
- 7- Utilizando uma linguagem de alto nível qualquer (ex: Pascal), esboce o programa do sistema de acesso com senha. A partir deste programa, determine um conjunto mínimo de **instruções de máquina necessárias para a sua CPU realizar o programa esboçado.**
- 8- Faça um **diagrama do Data Path** da CPU, conforme já especificado acima, incluindo: registros, memória, barramentos, ligação com a unidade de controle, 3-states e sinais de enable e clocks de registros, necessários ao seu projeto.
- 9- Usando a linguagem RTL (Register Transfer Language), detalhe as operações de controle para a execução do **fetch e das instruções necessárias a serem implementadas, indicando os estados Ti e os sinais acionados**. Coloque esses sinais de controle nas colunas e os estados Ti nas linhas de uma planilha Excel que será usada mais tarde no projeto da Unidade de Controle Microprogramada.
- 10- Usando **apenas o simulador**, reduza a **palavra da CPU para 4 bits na versão simulada para dar menos trabalho**, e implemente o **Data Path** (memória, registros, barramentos e circuitos de acesso) no simulador.
- 11- Agora, realize na FPGA, o **Data Path** (memória, registros, barramentos e circuitos de acesso).