# Sistemas Reconfiguráveis Eng. de Computação

Profs. Francisco Garcia e Antônio Hamilton Magalhães

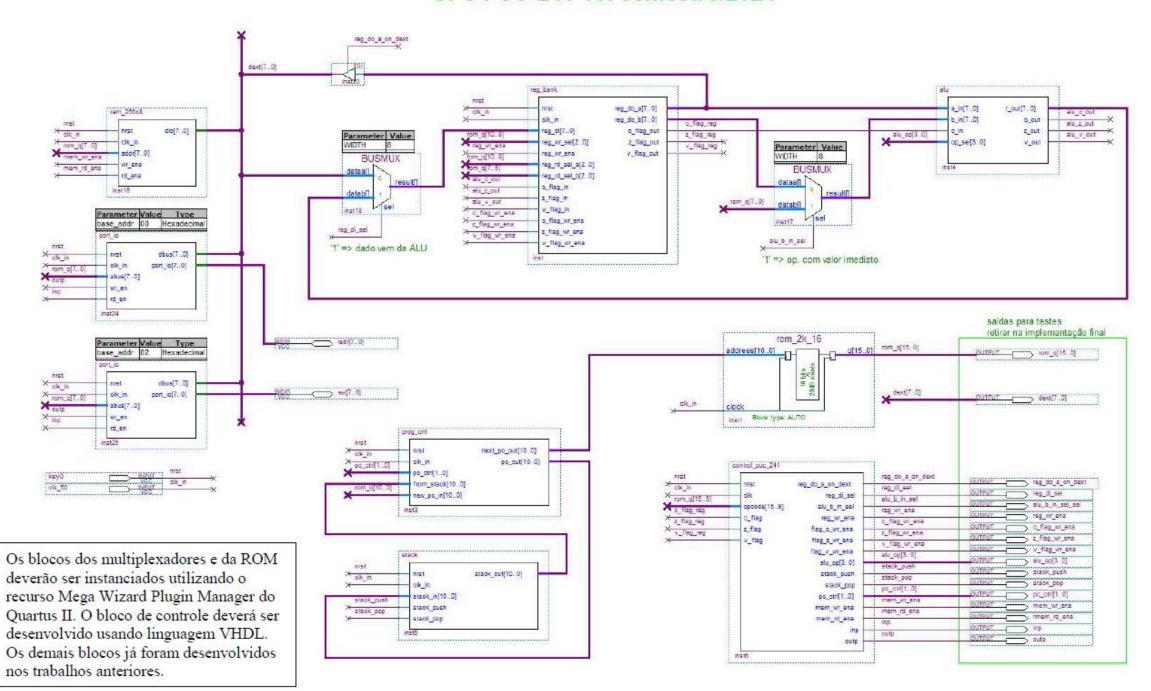
Aula 16 – Especificação do trabalho 4 – Processador PUC-124

## Especificações gerais



- O microcontrolador PUC-124 é um controlador destinado a fins educacionais.
- Sua CPU é similar a um processador de 8 bits usada em aplicações embutidas de baixa complexidade, mas o seu conjunto de instruções é mais parecido com o de um processador RISC.
- O processador do PUC-124 usa memórias separadas para instruções e dados (arquitetura Harvard).
- A memória de instruções (programa) tem uma capacidade de 2k instruções, e a memória de dados é de 256 x 8 bits.
- O processador também endereça 256 portas de entrada e 256 portas de saída.
- Possui oito registradores de propósito geral de 8 bits cada, r0 a r7, sendo que o registrador R7 armazena o *status* do processador: **c** (*carry*), **z** (zero), **v** (*overflow*).
- Todas as instruções, exceto as instruções de desvio (JMP, CALL, SKIPC, SKIPZ, SKIPV e RET) são executadas em um único ciclo de *clock*. As instruções de desvio necessitam de dois ciclos de *clock*.
- O topo da hierarquia deverá ser feito usando diagrama esquemático. Deverão ser instanciadas duas portas de entrada/saída. Uma delas (port\_A) deverá ser configurada, por software, para funcionar como saída. A outra (port\_B) deverá ser usada como entrada. Todas as entradas e saídas deverão ser do tipo std\_logic ou std\_logic\_vector.
- Para esse trabalho não será necessário o relatório. Deverá ser entregue um arquivo compactado (.zip), com todos os arquivos dos projetos gerados no ambiente Quartus, incluindo o arquivo de simulação e o arquivo com o programa da aplicação.
- Assim como os outros trabalhos, a versão do Quartus a ser utilizada deve ser a 9.1sp2.

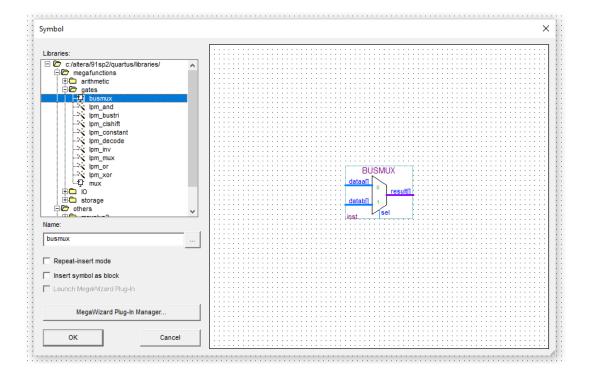
#### CPU PUC-241 1o. semestre/2024

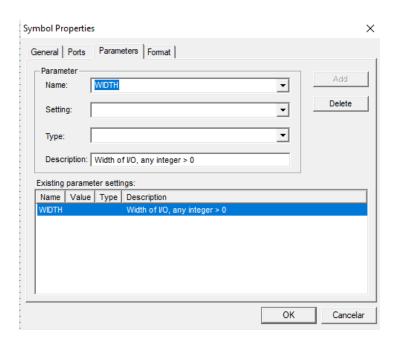


### BUSMUX



O componente BUSMUX corresponde a um elemento da bliblioteca do Quartus, inserido no arquivo "Block Diagram/Schematic File" (libraries->megafunction->gates). O parâmetro WIDTH do componente devem se configurado (click-duplo sobre o box de parâmetro).



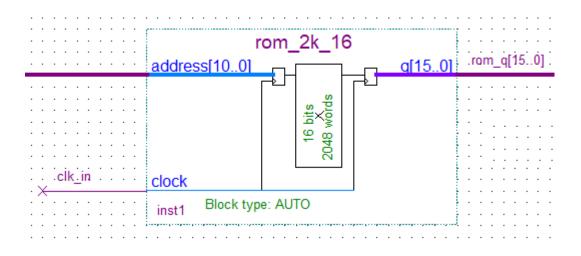


## ROM



A memória de programa (ROM) será inserida no "Block Diagram/Schematic File" por meio do menu TOOLS-> Megawizard Plu-gin Manager

Os parâmetros de tamanho da memória e largura do barramento devem ser configurados de acordo com o documento de arquitetura do projeto.



## ROM



Esse componente carrega um arquivo .mif que deve ser editado no quartus para conter o programa de testes em linguagem assembly.

#### Programa de teste inicial para o processador PUC-241

v0 - 01/06/26

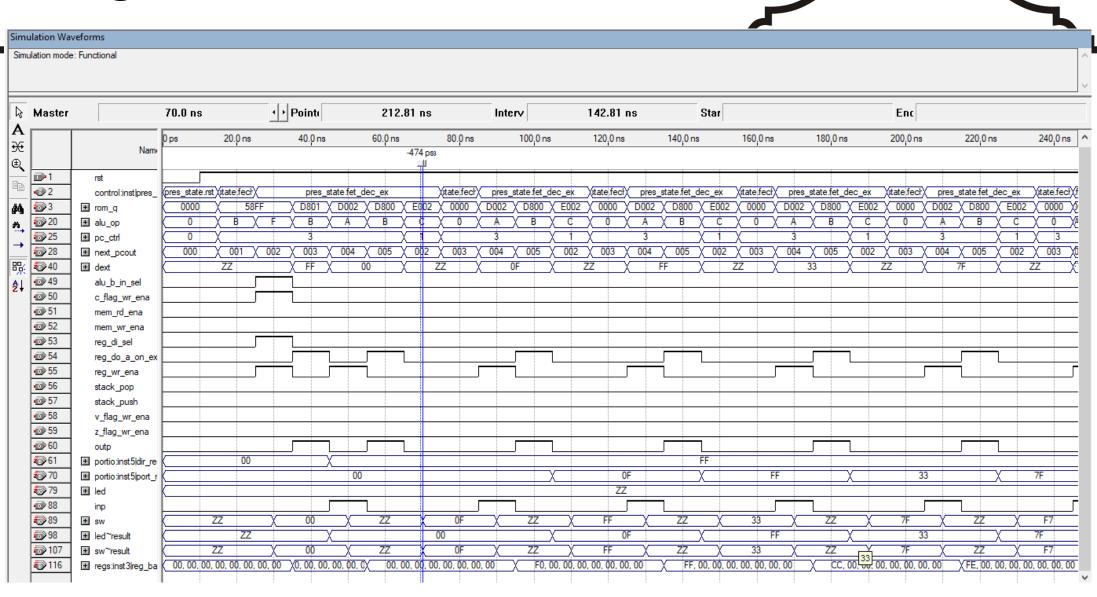
Esse programa configura a port A como saída e mantém a port B como entrada (*default*). As entradas da port B são enviadas às saídas da port A (bit a bit). O programa fica em loop.

#### Inicialização

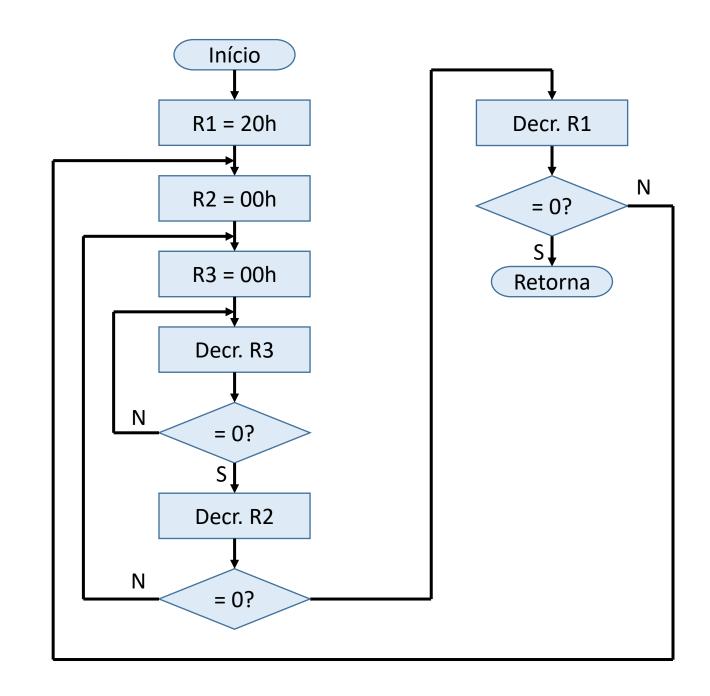
End. (hexa)	Intr.	Bin	Hexa	Obs.		
000	MOVI RO, FFh	01 011 000 1111 1111	58FF	Ajusta todos os pinos da <u>port A</u> como saída		
001	OUT 01, R0	11 0 11 000 0000 0001	D801			
002	INP RO, 02	11 0 10 000 0000 0010	D002	Lê a port B		
003	OUT 00, R0	11 0 11 000 0000 0000	D800	Escreve na port A		
004	JMP 002h	11 1 0 0 000 0000 0010	E002	Jumper (loop)		
			*			
- 1			2	8		

Addr	+0	+1	+2	+3	+4	+5	+6	+7
000	58FF	D801	D002	D800	E002	0000	0000	0000
800	0000	0000	0000	0000	0000	0000	0000	0000
010	0000	0000	0000	0000	0000	0000	0000	0000
018	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000
028	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000
038	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000
048	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000
058	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000

## Programa de testes inicial



### Rotina de atraso





### Programa – aplicação simples

- Deverá ser elaborado um programa para um teste simples de funcionamento.
- Esse programa deverá acionar sequencialmente os bits da porta de saída a intervalos regulares.
- Essas saídas serão ligadas, no módulo DE2, a LEDs.
- Um bit da porta de entrada (que será ligado a uma chave no módulo) deverá alterar a sequência de acendimento dos LED's, (do mais significativo para o menos significativo e vice-versa).
- A temporização para acionamento dos LEDs deverá ser baseada no tempo de execução de instruções.
- Cada estado dos leds deve ter duração de 0,1 segundos.

## Entregas do projeto

- Enviar arquivo .zip como tarefa no Canvas, com a árvore de diretórios contendo todos os arquivos do projeto, de forma que possa ser compilado no QUARTUS II e executado no módulo DE2.
- Demonstrar a execução do programa de testes no módulo DE2.



# Fim