

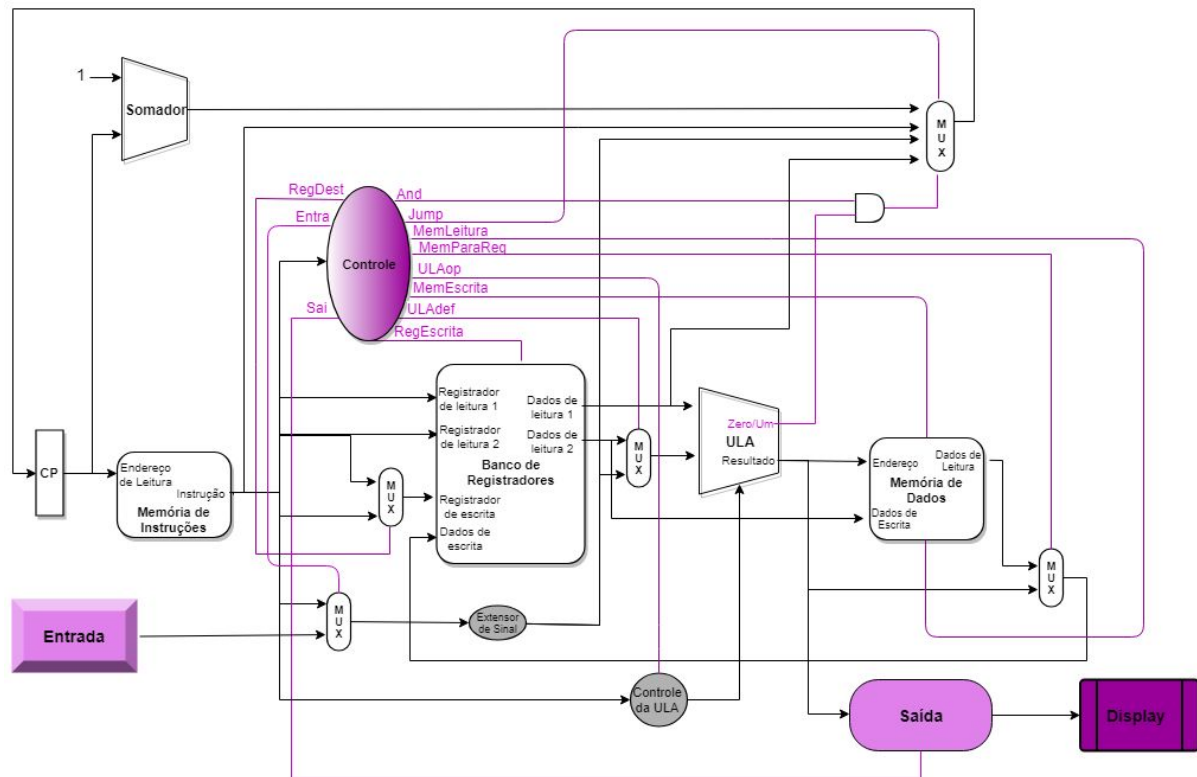
MODELAGEM EM SYSML DO PROCESSADOR BASEADO EM MIPS E PROGRAMADO EM UMA PLACA CYCLONE IV E FPGA

Este processador é baseado na arquitetura MIPS monociclo, com o processador RISC (Conjunto reduzido de instruções), contendo 32 registradores de 32 bits, sendo assim possui um conjunto de instruções mais simples, rápido, contendo:

- 3 formatos de instruções:
 - tipo R
 - tipo I
 - tipo J
- 5 modos de endereçamento:
 - Endereçamento Imediato
 - Endereçamento por Registrador
 - Endereçamento por Deslocamento
 - Endereçamento Direto
 - Endereçamento Relativo ao PC
- 38 instruções:

Opcode	Instrução	Tipo	Operação	Descrição
Aritméticas				
000000	add	R	$RD = RS + RT$	soma
000001	addi	I	$RT = RS + \text{IMEDIATO}$	soma com imediato
000010	sub	R	$RD = RS - RT$	subtração
000011	subi	I	$RT = RS - \text{IMEDIATO}$	subtração com imediato
000100	mul	R	$RD = RS * RT$	multiplicação
000101	muli	I	$RT = RS * \text{IMEDIATO}$	multiplicação com imediato
000110	div	R	$RD = RS / RT$	divisão
000111	divi	I	$RT = RS / \text{IMEDIATO}$	divisão com imediato
Lógicas				
001000	and	R	$RD = RS \& RT$	E
001001	andi	I	$RT = RS \& \text{IMEDIATO}$	E com imediato
001010	or	R	$RD = RS RT$	OU
001011	ori	I	$RT = RS \text{IMEDIATO}$	OU com imediato
001100	xor	R	$RD = RS \wedge RT$	ou exclusivo
001101	xori	I	$RT = RS \wedge \text{IMEDIATO}$	ou exclusivo com imediato
001110	not	I	$RT = \sim RS$	valor negado
Deslocamento				
001111	sll	R	$RD = RS \ll \text{shamt}$	Deslocamento a esquerda
010000	srl	R	$RD = RS \gg \text{shamt}$	Deslocamento a direita
Desvio Incondicional				
010001	jump	J	$CP = \text{novo CP}$	Pula para o endereço
010010	jal	J	$\$ra = CP; CP = \text{novo CP}$	Pula e armazena endereço de retorno
010011	jr	J	$CP = \text{endereço de } \ra	pula para registrador
Desvio Condicional				
010100	beq	I	se $RS == RT$, $CP = \text{desvio}$	Desvia se igual
010101	bnq	I	se $RS != RT$, $CP = \text{desvio}$	Desvia se diferente
010110	blt	I	se $RS < RT$, $CP = \text{desvio}$	Desvia se menor
010111	bgt	I	se $RS > RT$, $CP = \text{desvio}$	Desvia de maior
011000	bltz	I	se $RS < 0$, $CP = \text{desvio}$	Desvia se menor que zero
011001	bgtz	I	se $RS > 0$, $CP = \text{desvio}$	Desvia se maior que zero
011010	slt	R	$RD = RS < RT?$	RD igual a 1 se menor, 0 se maior ou igual
011011	slet	R	$RD = RS \leq RT?$	RD igual a 1 se menor ou igual, 0 se maior
011100	sgt	R	$RD = RS > RT?$	RD igual a 1 se maior, 0 se menor ou igual
011101	sset	R	$RD = RS \geq RT?$	RD igual a 1 se maior ou igual, 0 se maior
Acesso a Memória				
011110	li	I	$RT = \text{Imediato}$	igual a imediato
011111	lw	I	$RT = \text{Mem}[RS + \text{offset}]$	Carrega uma palavra
100000	sw	I	$\text{Mem}[RS + \text{offset}] = RT$	Armazena uma palavra
100001	move	I	$RT = RS$	igual ao valor do registrador
Controle				
100010	nop	J	-	Sem operação para executar
100011	hlt	J	-	Parar processamento
Entrada e Saída				
100100	in	-	-	Sinal de entrada
100101	out	-	-	Sinal de saída

O caminho de Dados mostrado abaixo foi construído baseado no caminho de dados da arquitetura MIPS, feito pequenas modificações na implementação.



O projeto do processador foi implementado e testado em uma placa Cycloce IV E FPGA, contendo 18 switches de entrada e 8 displays de 7 segmentos. Nesta disciplina, o projeto 1 tem o objetivo principal de colocar em prática, de forma integrada, os modelos diagramáticos da SysML. Para isso deverá ser feita a modelagem do processador comentado acima. Como este projeto trata-se da modelagem de um processador, no qual é um sistema complexo e difícil, não foi possível mostrar todas suas atividades e utilidades, mas sim um resumo delas.

O modelo diagramático da SysML foi montado no software Visual Paradigm contendo 1 diagrama de pacotes, 5 diagramas de casos de uso, 4 diagramas de atividade e 5 diagramas de requisitos. Para modelar o diagrama de pacotes, foi dividido em 5 pacotes, sendo eles: Entrada/Saída, Geral, Memórias, Operações e PC.