

## 1.Introdução

Este relatório tem como intenção relatar a confecção de um processador Saguí vetorial baseado em uma arquitetura de 8 bits, load-store endereçada por byte, com instruções escalares e vetoriais, projetado no software Logisim Evolution.

O processador tem como seus principais componentes uma Memória de Programa, um Program Counter que a controla, uma Memória de Controle para controlar muxes e bits de enable, 5 Bancos de Registradores(4 X 8b), sendo 4 desses vetoriais(r0 uma constante ID, e os outros normais) e um escalar(r0 uma constante 0, e os outros 3 normais), 5 ULAs(4 vetoriais e 1 escalar) e 5 Memórias de Dados(4 vetoriais e uma escalar).

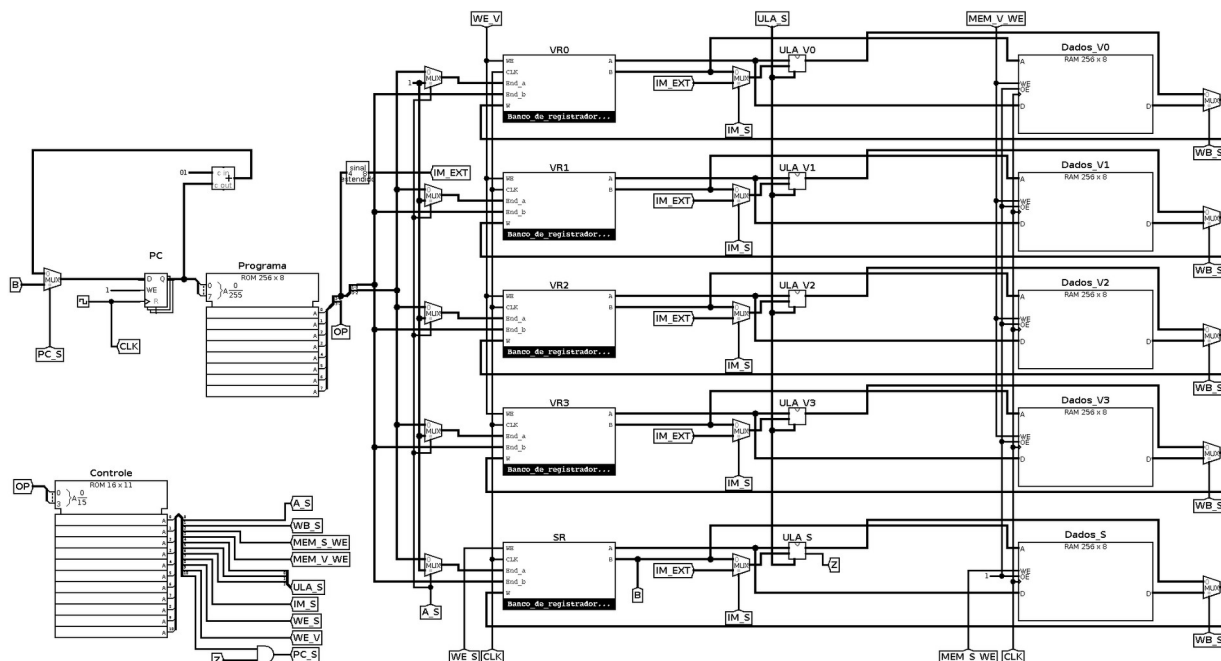
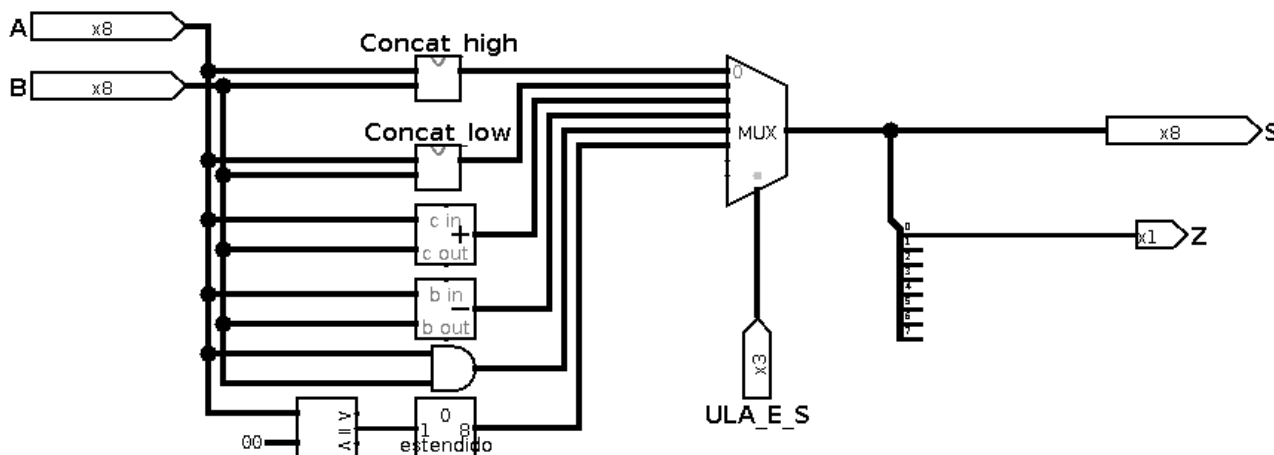


Figura 1: Diagrama de caixas do saqui vetorial

## 2.ULAs

Duas ULAs foram projetadas para esse processador, sendo que a única diferença entre elas é que a ULA escalar(Figura 2) consegue realizar uma comparação entre o registrador A e a constante 0, enquanto a ULA vetorial(Figura 3) pode fazer um OR lógico entre os registradores A e B. Além disso, os dois projetos podem fazer: concatenação dos bits do imediato para a parte alta ou baixa do registrador, soma, subtração e AND lógico.



*Figura 2 : ULA Escalar*

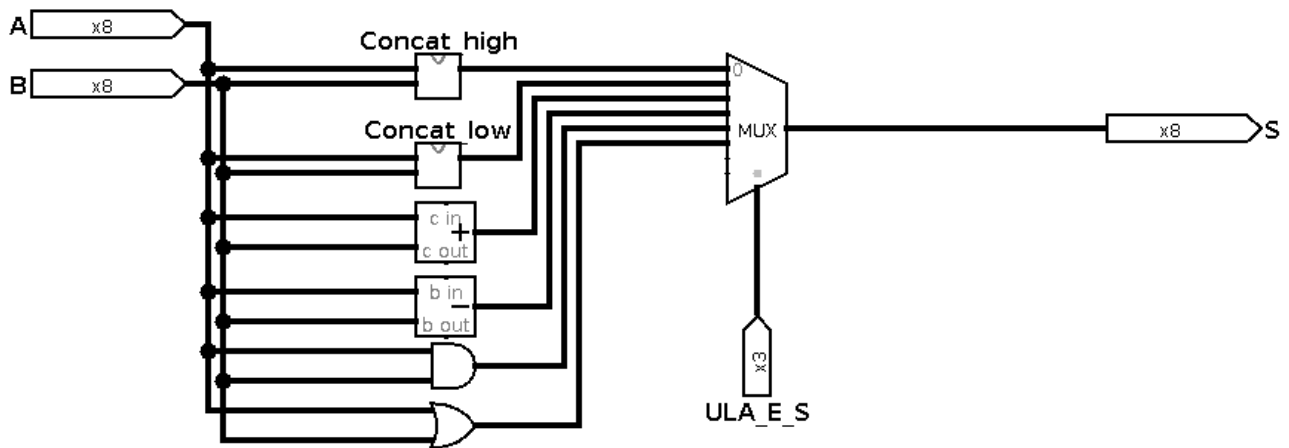


Figura 3: ULA Vetorial

### 3. Sinais de controle

Foi montada a seguinte tabela para organizar os sinais de controle do processador:

Assembl	PC_	WE_	WE_	IM_	ULA_	MEM_V_	MEM_S_W	WB_	
y	OP	S	V	S	S	WE	E	S	A_S
lds	0000	0	1	0	000	0	0	1	0
sts	0001	0	0	0	000	0	1	0	0
movhs	0010	0	1	1	000	0	0	0	1
movls	0011	0	1	1	001	0	0	0	1
adds	0100	0	1	0	010	0	0	0	0
subs	0101	0	1	0	011	0	0	0	0
ands	0110	0	1	0	100	0	0	0	0
brzrs	0111	1	0	0	101	0	0	0	0
ldv	1000	0	1	0	000	0	0	1	0
stv	1001	0	0	0	000	1	0	0	0
movhv	1010	0	1	0	1	000	0	0	1
movlv	1011	0	1	0	1	001	0	0	1
addv	1100	0	1	0	0	010	0	0	0
subv	1101	0	1	0	0	011	0	0	0
andv	1110	0	1	0	0	100	0	0	0
orv	1111	0	1	0	0	101	0	0	0

### 4. Programa

Um programa de teste foi feito em assembly e transcrito em base hexadecimal para a Memória de Programa do Sagui para que fosse possível analisar o funcionamento do circuito no Logisim Evolution.

Instruções terminadas em s são escalares e terminadas em v são vetoriais.

Para instruções do tipo R os 4 bits mais importantes representam o opcode, os 2 próximos representam o endereço do registrador a, e os últimos 2 representam o endereço do registrador B.

Para instruções do tipo I os 4 bits mais importante representam o opcode, e os outros 4 representam um imediato.

PC	Tipo	Assembly	Binario
	0 Tipo I	movls 5	00110101
	1 Tipo R	adds r2, r1	01001001
	2 Tipo R	ands r1, r2	01100110
	3 Tipo I	movls 5	00110101
	4 Tipo I	movhs 1	00100001
	5 Tipo R	adds r3, r1	01001101
	6 Tipo R	subs r3, r2	01011110
	7 Tipo R	sts r3, r2	00011110
	8 Tipo R	lds r1, r3	00000111
	9 Tipo I	movhs 0	00100000
	10 Tipo I	movls 13	00111101
	11 Tipo R	brzrs r0, r1	01110001
	12 Tipo I	movhs 1	00100001
	13 Tipo I	movlv 5	10110101
	14 Tipo R	addv r2, r1	11001001
	15 Tipo R	andv r1, r2	11100110
	16 Tipo I	movlv 5	10110101
	17 Tipo I	movhv 1	10100001
	18 Tipo R	addv r3, r1	11001101
	19 Tipo R	subv r3, r2	11011110
	20 Tipo R	stv r3, r2	10011110
	21 Tipo R	ldv r1, r3	10000111
	22 Tipo R	orv r2, r1	11111001
	23 Tipo R	subs r3, r3	01011111
	24 Tipo R	subv r2, r2	11011010
	25 Tipo R	subv r3, r3	11011111
	26 Tipo R	subv r1,r1	11010101
	27 Tipo I	movhs 0	00100000
	28 Tipo I	movls 1	00110001
	29 Tipo R	adds r3, r1	01001101
	30 Tipo I	movhs 2	00100010
	31 Tipo I	movls 6	00110110
	32 Tipo R	brzrs r2, r1	01111001
	33 Tipo R	stv r1, r1	10010101
	34 Tipo R	addv r1, r0	11000100
	35 Tipo R	subs r2, r3	01011011
	36 Tipo I	movhs 1	00100001
	37 Tipo I	movls 14	00111110
	38 Tipo R	brzrs r0, r1	01110001

#### 4.Conclusão

Ao fim do processo temos um processador Sagui funcional com uma ROM programável, e uma memória RAM conectada ao processador, permitindo que um programador possa programá-lo uma vez utilizando as 16 instruções nativas.