



Universidade Federal de Roraima
Departamento de Ciência da Computação
Arquitetura e Organização de Computadores

Processador Uniciclo de 8 Bits

Por Ilem Lima dos Santos e Paulo César Pereira Belmont

Maio de 2021

Características

O processador implementado possui arquitetura de 8 bits, operação uniciclo e se assemelha a um processador de arquitetura MIPS.

Pode trabalhar com -2^7 até 2^7-1 valores numéricos inteiros.

Especificamente, os valores vão de -128 a 127

Existem 8 componentes e 4 tipos de subcomponente na unidade como um todo.

Formato das Instruções

TIPO R		
Opcode	Reg 1	Reg 2
4 bits	2 bits	2 bits
7-4	3-2	1-0

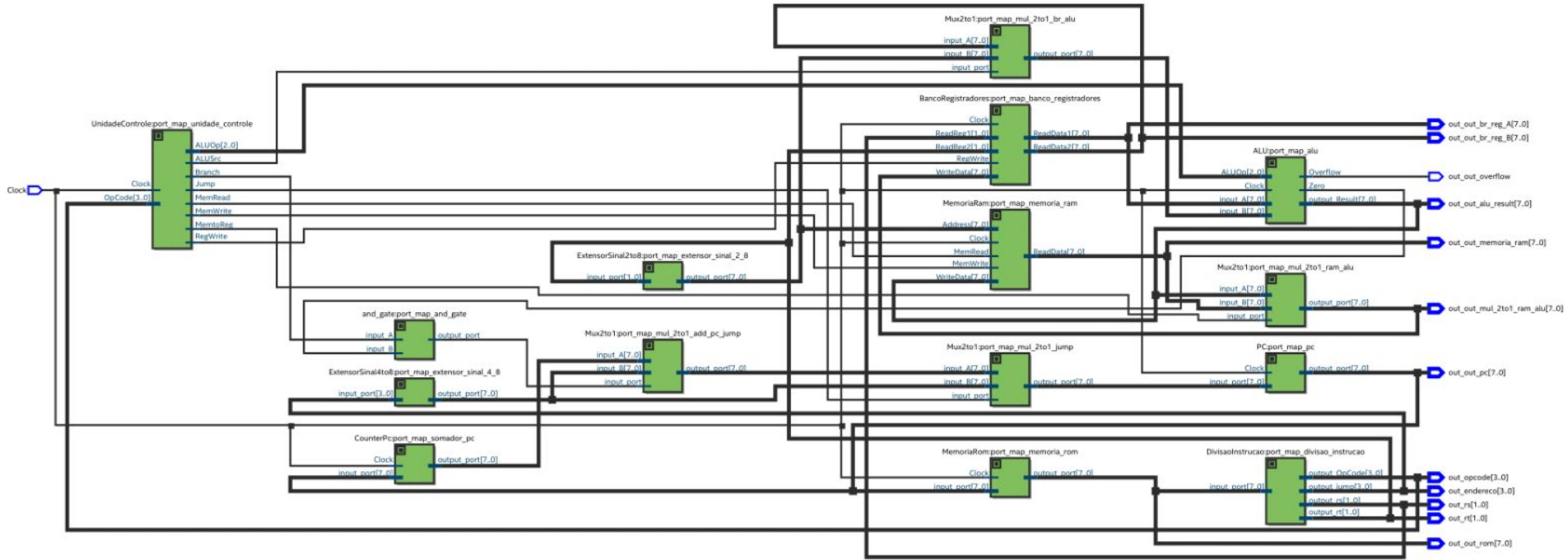
TIPO I		
Opcode	Reg 1	Imediato
4 bits	2 bits	2 bits
7-4	3-2	1-0

TIPO J	
Opcode	Endereço
4 bits	4 bits
7-4	3-0

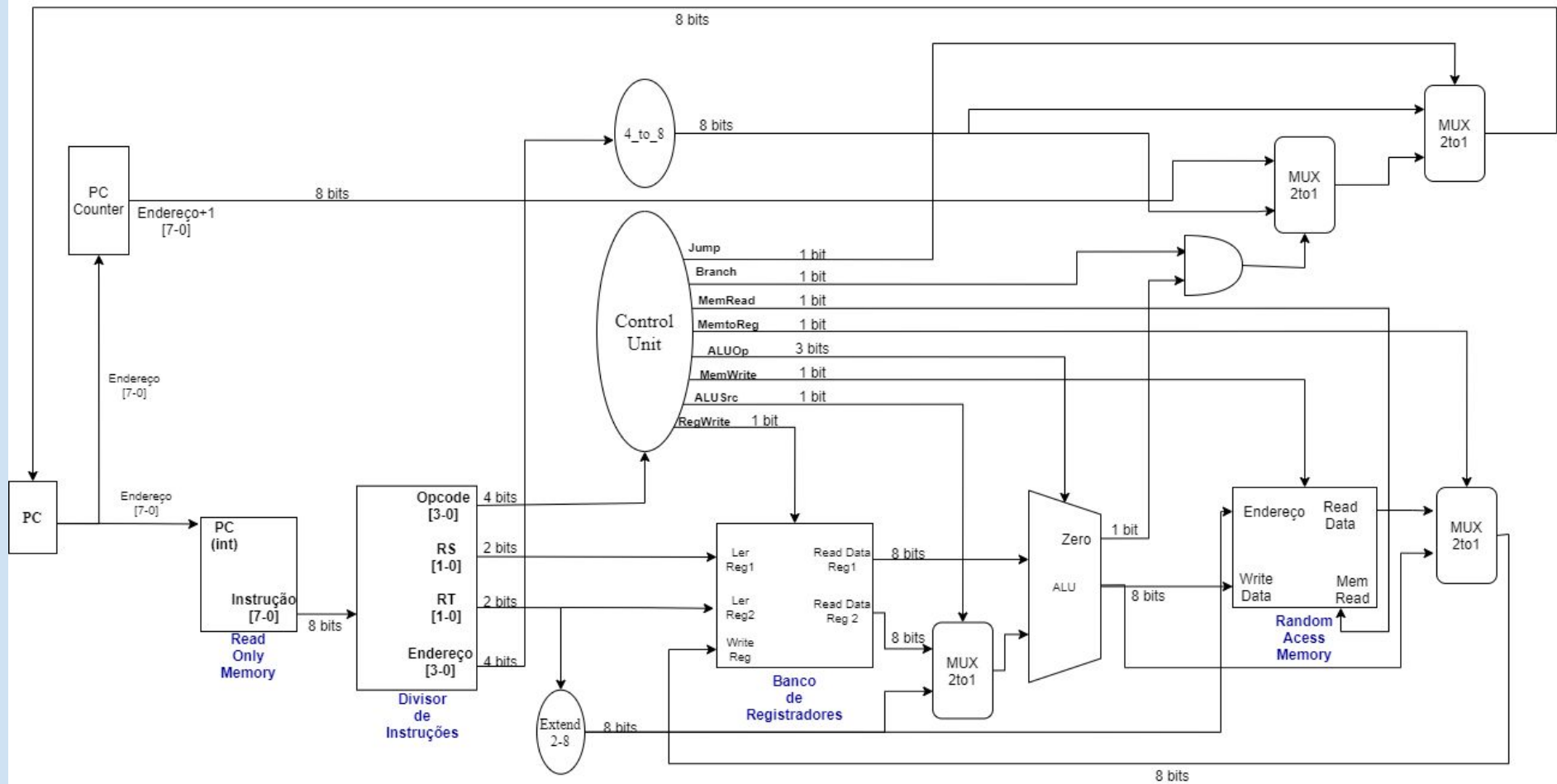
Operações Implementadas

Opcode	Nome	Formato	Breve Descrição	Exemplo
0000	LW	R	Load	lw \$s0,\$s1
0001	SW	R	Store	sw \$s0,\$s1
0010	ADD	R	Soma	add \$s0 \$s1 - \$s0 = \$s0+\$s1
0011	SUB	R	Subtração	sub \$s0 \$s1 - \$s0 = \$s0-\$s1
0100	ADDI	I	Soma imediata	addi \$s0 10 - \$s0 = \$s0 + 2
0101	SUBI	I	Subtração imediata	subi \$s0 11 - \$s0 = \$s0 - 3
0110	MOVE	R	Move	move \$s0 \$s1 - \$s0 = \$s1
0111	LI	I	Load Imediato	li \$s0,2 - \$s0 = ram(02)
1000	BEQ	J	Branch if equal	beq \$s0,\$s1, 0000
1001	BNE	J	Branch if not equal	bne \$s0,\$s1, 0000
1010	CMP	R	Comparação	cmp \$s0,\$s1
1011	JUMP	J	Salto incondicional	jump 0000 -> PC= 0000

RTL Viewer



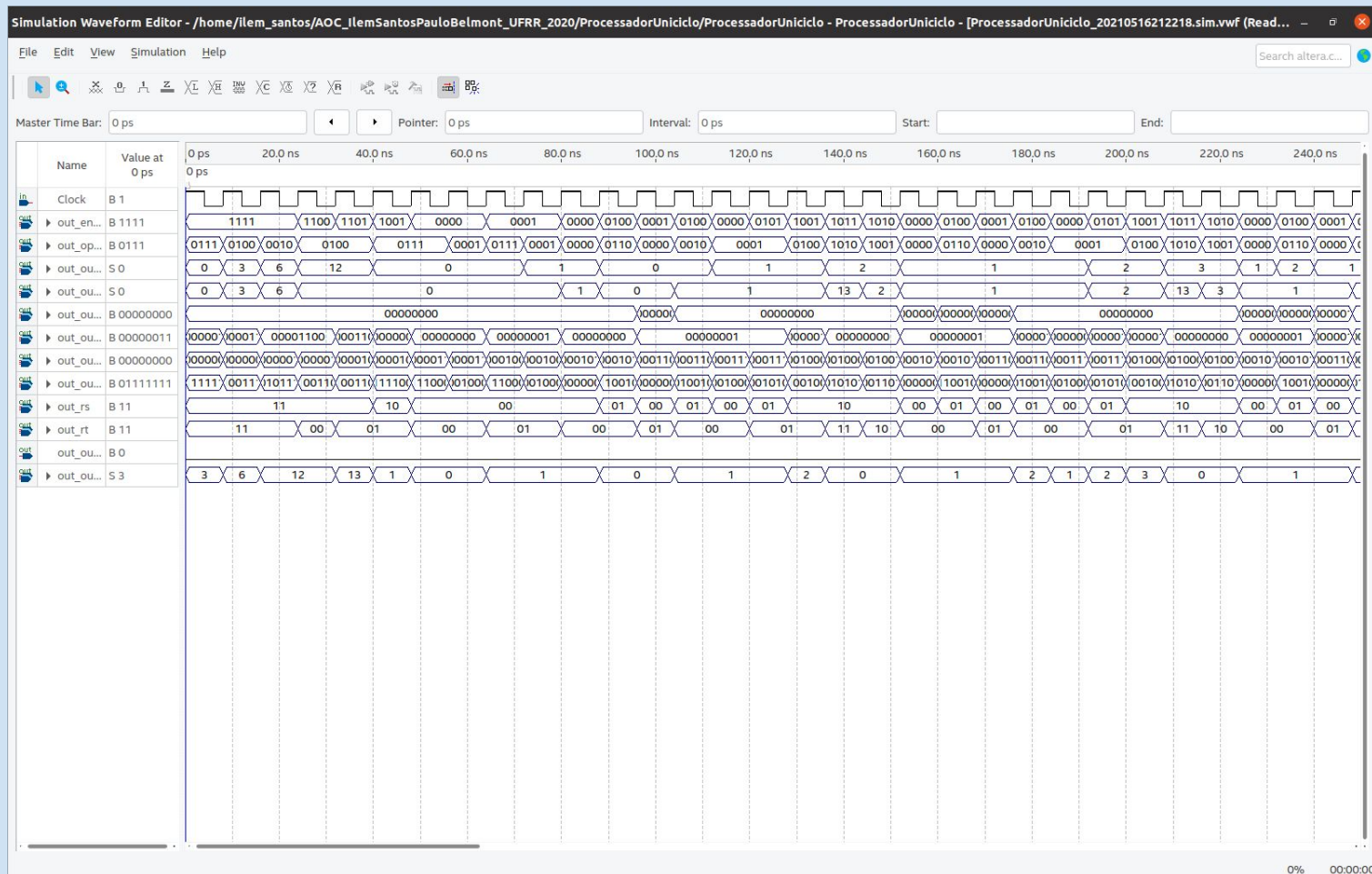
Datapath



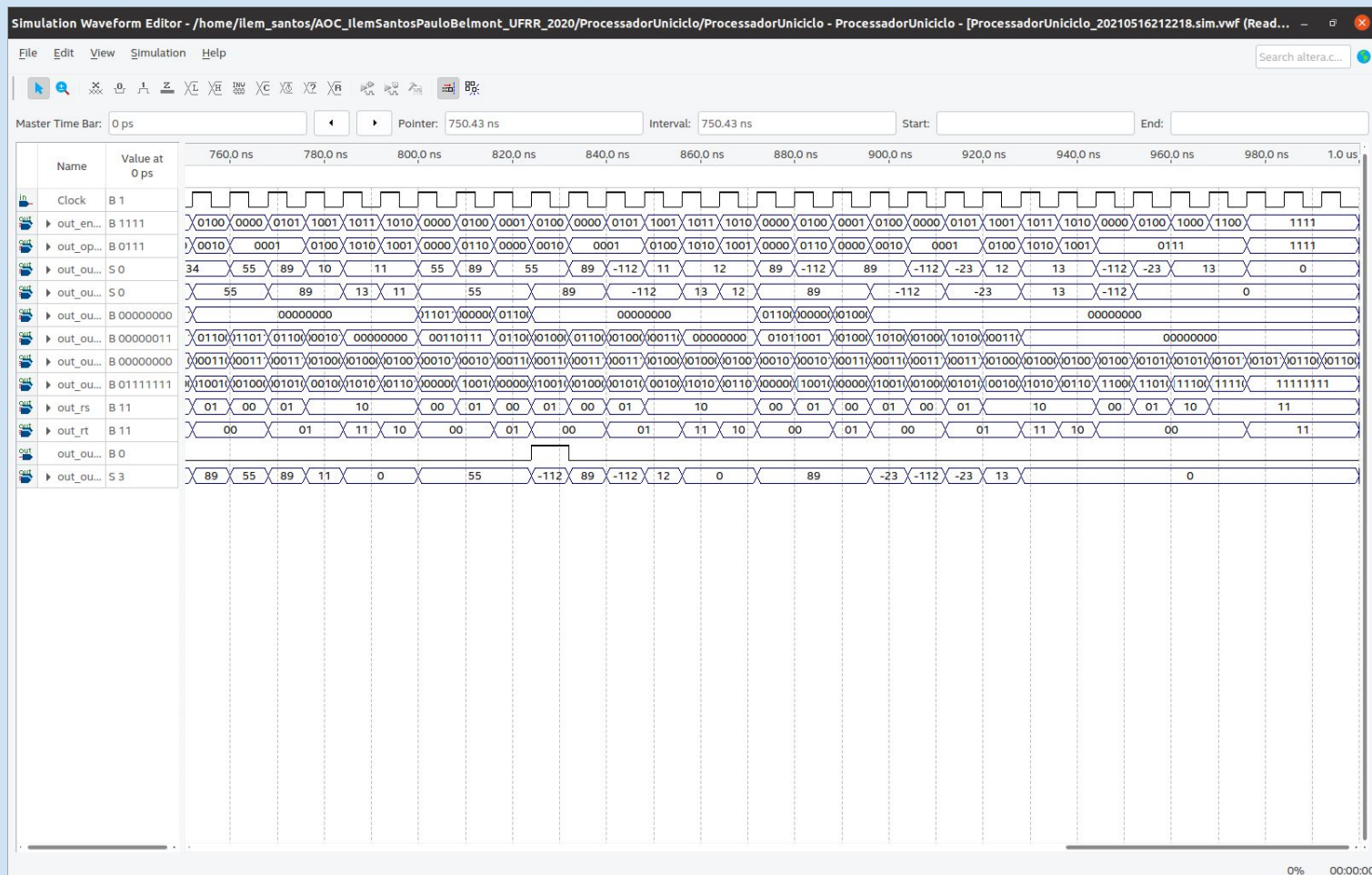
Programa Fibonacci

Endereço	Instrução	Alto Nível	Binário		
			Opcode	Reg2	Reg1
				Imediato	Endereço
0	01111111	li S3 3	0111	11	11
1	01001111	addi S3 3 == 6	0100	11	11
2	00101111	add S3 S3 == 12	0010	11	11
3	01001100	addi S3 0 == 12	0100	11	00
4	01001101	addi S3 1 == 13	0100	11	01
5	01111001	li S2 1	0111	10	01
6	01110000	li S0 0	0111	00	00
7	00010000	sw S0 ram(00) == fib1 = 0	0001	00	00
8	01110001	li S0 1	0111	00	01
9	00010001	sw S0 ram(01) == fib2 = 1	0001	00	01
10	00000000	lw S0 ram(00)	0000	00	00
11	01100100	move S1 S0	0110	01	00
12	00000001	lw S0 ram(01)	0000	00	01
13	00100100	add S1 S0	0010	01	00
14	00010000	sw S0 ram(00) == fib1 = fib2	0001	00	00
15	00010101	sw s1 ram(01) == fib2 = aux_soma	0001	01	01
16	01001001	addi s2 1	0100	10	01
17	10101011	cmp S2 == S3	1010	10	11
18	10011010	bne S2 != S3 jump 1010	1001	10	10
19	01110000	li S0 0	0111	00	00
20	01110100	li S1 0	0111	01	00
21	01111000	li S2 0	0111	10	00
22	01111100	li S3 0	0111	11	00

Waveforms notáveis no Fibonacci



Waveforms notáveis no Fibonacci



Conclusão

Devido ao pequeno tamanho máximo de bits por instrução, as operações do processador são muito limitadas, sendo notáveis tais limitações:

- Os passos acessíveis pelo Jump vão de 0 a 15
- Apenas números inteiros são utilizados como informação.
- Uma informação pode armazenar apenas números entre -128 e 127, havendo um pino de overflow na waveform para acusar o “estouro”.

Todavia, as instruções, se executadas dentro dos limites impostos pelo componente, devolvem os valores corretos.