



UNIVERSIDADE FEDERAL DE RORAIMA  
CENTRO DE CIÊNCIA E TECNOLOGIA  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO



DISCIPLINA: ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES  
PROFESSOR: HERBERT OLIVEIRA ROCHA

# PROCESSADOR XLA-8

Lucas Bessa Façanha Pereira  
Rafael Nóbrega de Lima

Boa Vista/RR  
2021

## Sobre o Projeto

- Processador RISC XRL-8 monociclo de 8 bits.
- Baseado na Arquitetura do processador MIPS.



# Formato de Instruções

Tipo R		
Opcode	Reg1	Reg2
4 bits	2 bits	2 bits
7-4	3-2	1-0

Tipo I		
Opcode	Reg1	Endereço/Constante
4 bits	2 bits	2 bits
7-4	3-2	1-0

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

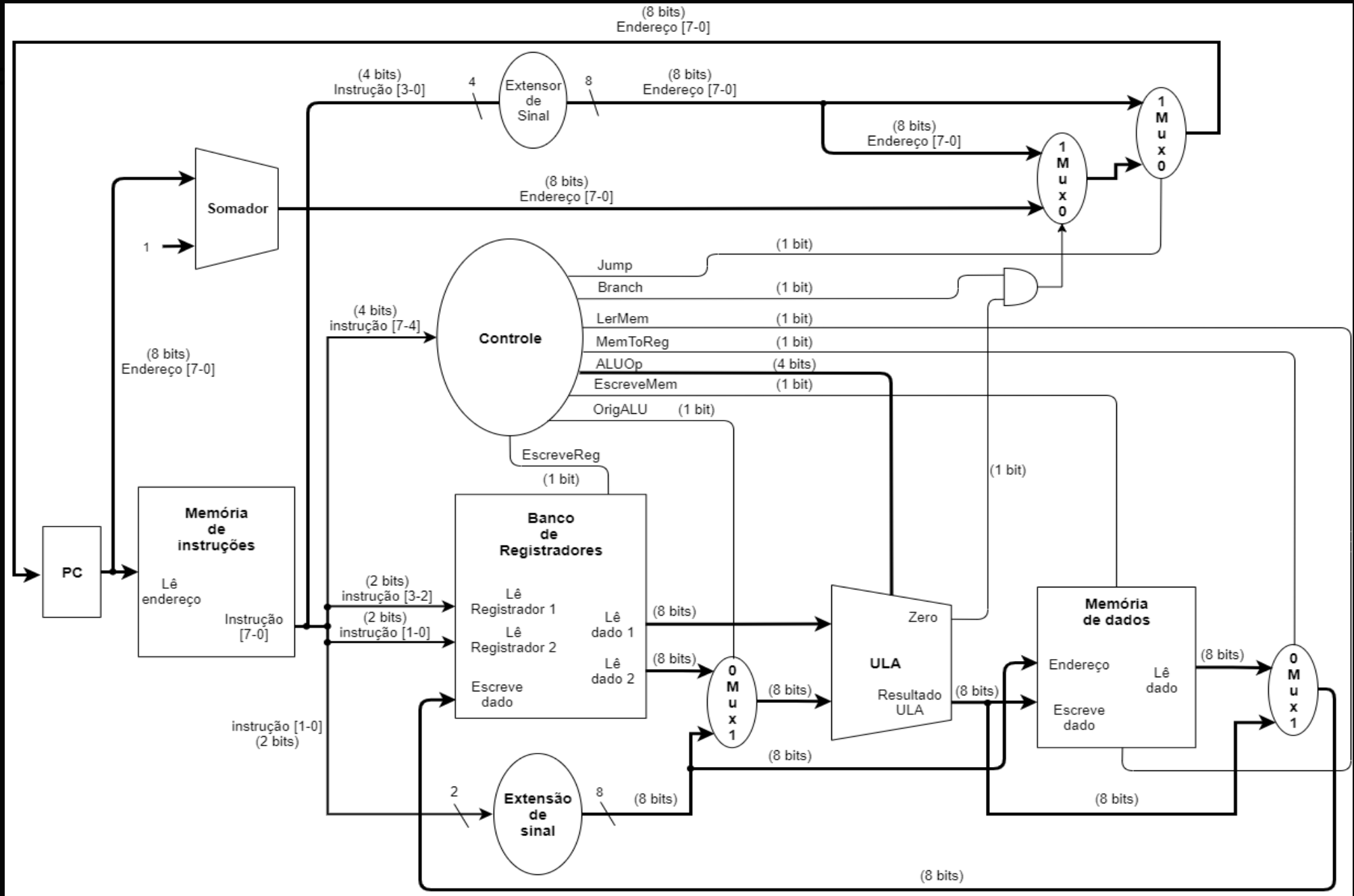
# Opcodes

Instrução	Tipo	Opcode	Sintaxe	Registradores
Add	R	(0000)	Add \$s0,\$s1	2
Sub	R	(0001)	Sub \$s0,\$s1	2
Lw	I	(0010)	Lw \$s0, address	2
Sw	I	(0011)	Sw \$s0, address	2
J	J	(0100)	J address	0
Beq	J	(0101)	Beq address	0
Bne	J	(0110)	Bne address	0
Addi	R	(0111)	Addi \$s0,valor	1
mul	R	(1000)	Mul \$s0,\$s1	2
And	R	(1001)	And \$s0,\$s1	2
Or	R	(1010)	Or \$s0,\$s1	2
Not	R	(1011)	Not \$s0	1
Li	R	(1100)	Li \$s0,value	2
Move	R	(1101)	Move \$s0,\$s1	2
JumpC	R	(1110)	JumpC \$s0,\$s1	2

# Unidade de Controle

Funcionamento da Unidade de Controle								
Instruções	Orig Alu	Escreve Reg	Escreve Mem	Alu Op	Mem To Reg	Ler mem	Branch	Jump
Add	0	1	0	0000	1	0	0	0
Sub	0	1	0	0001	1	0	0	0
Lw	X	1	0	0010	0	1	0	0
Sw	X	0	1	0011	1	0	0	0
J	X	0	0	0100	1	0	0	1
Beq	0	0	0	0101	1	0	1	0
Bne	0	0	0	0110	1	0	1	0
Addi	1	1	0	0111	1	0	0	0
Mul	0	1	0	1000	1	0	0	0
And	0	1	0	1001	1	0	0	0
Or	0	1	0	1010	1	0	0	0
Not	X	1	0	1011	1	0	0	0
Li	1	1	0	1100	1	0	0	0
Move	0	1	0	1101	1	0	0	0
JumpC	0	0	0	1110	1	0	1	0

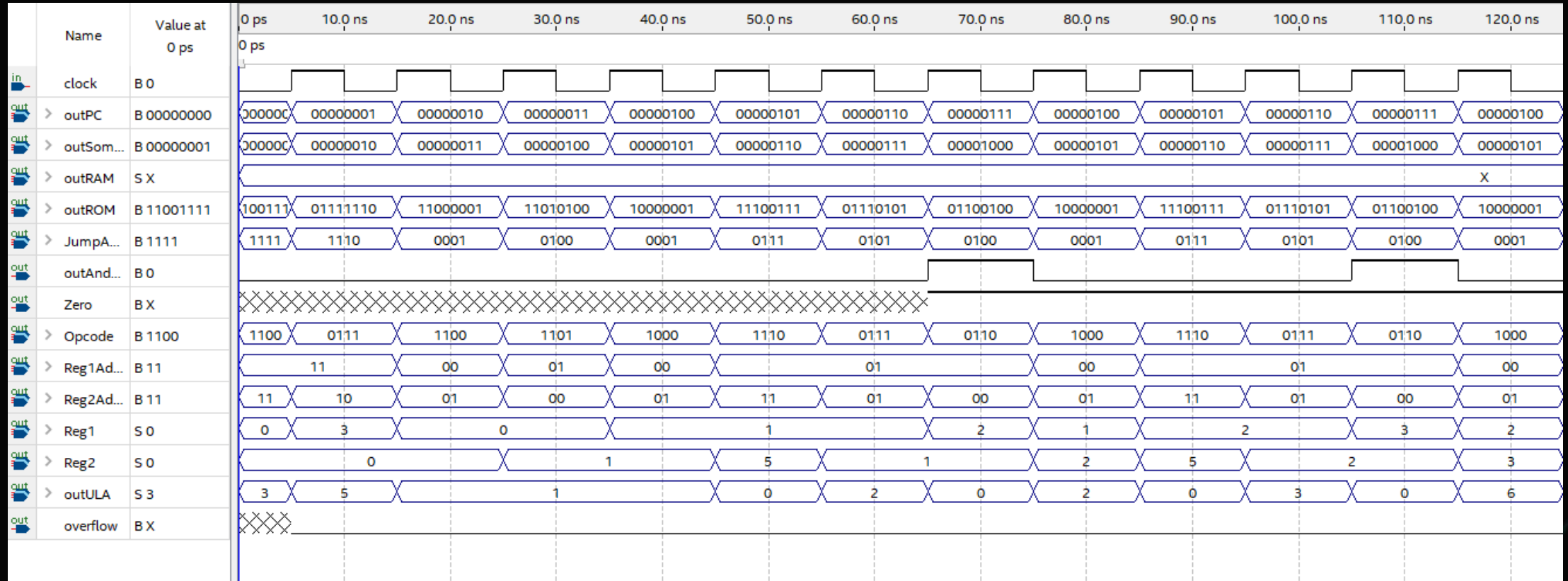
# DataPath



# Teste de Fatorial

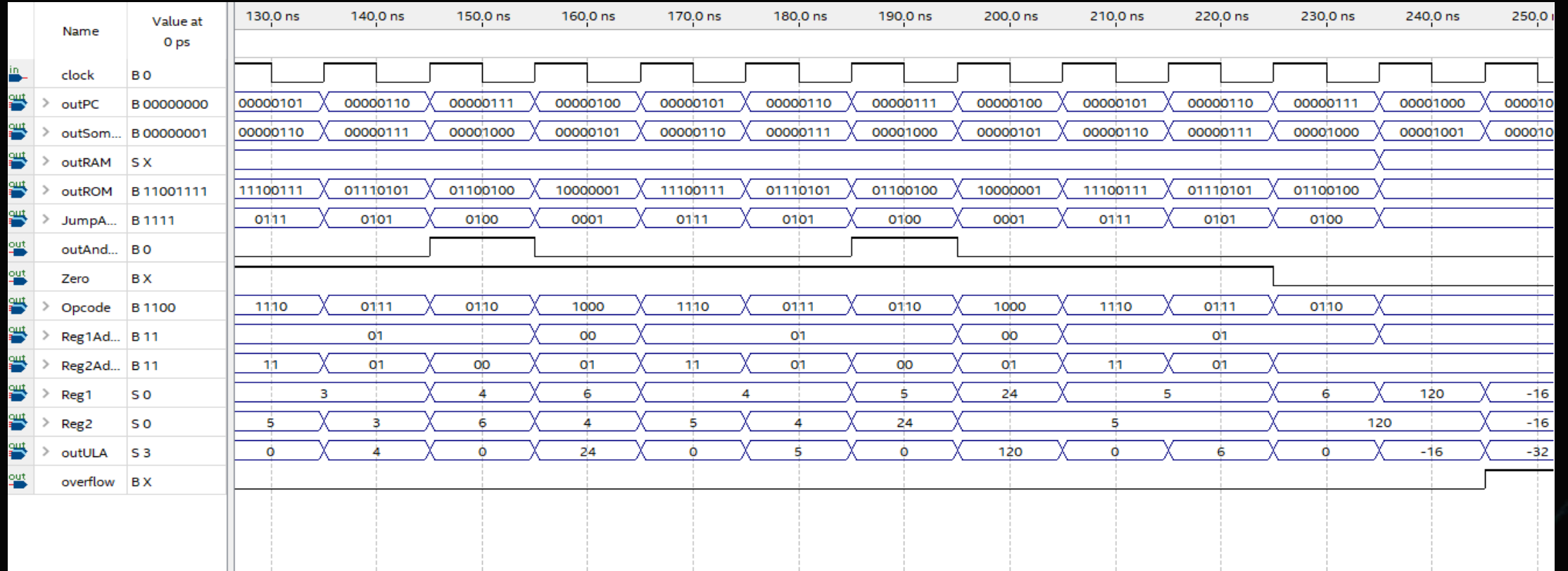
Endereço	Instrução(Omni)	Instrução (Binário)		
		Opcode	Reg1	Reg2
			Endereço	
0	Li \$s3, 3	1100	11	11
1	Addi \$s3, 2	0111	11	10
2	Li \$s0, 1	1100	00	01
3	Move \$s1, \$s0	1101	01	00
4	Mul \$s0, \$s1	1000	00	01
5	JumpC \$s1, \$s3	1110	01	11
6	Addi \$s1, 1	0111	01	01
7	Bne 4	0110	0100	

# Teste de Fatorial





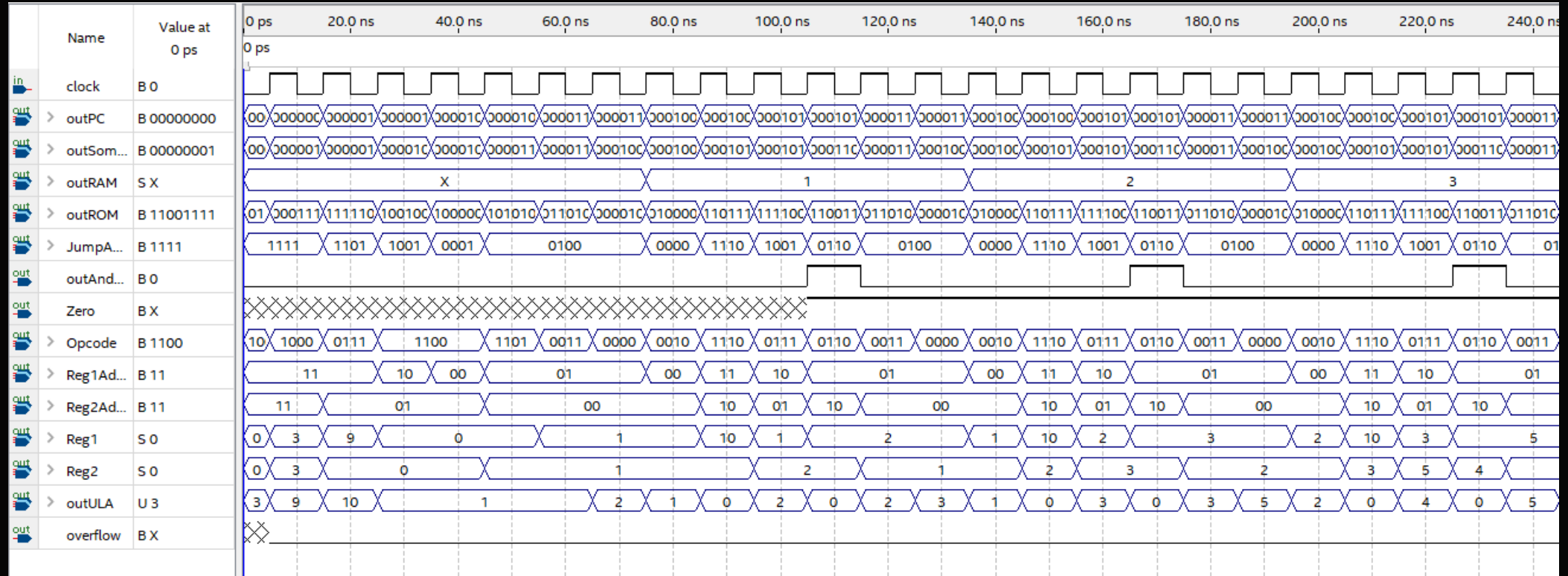
# Teste de Fatorial



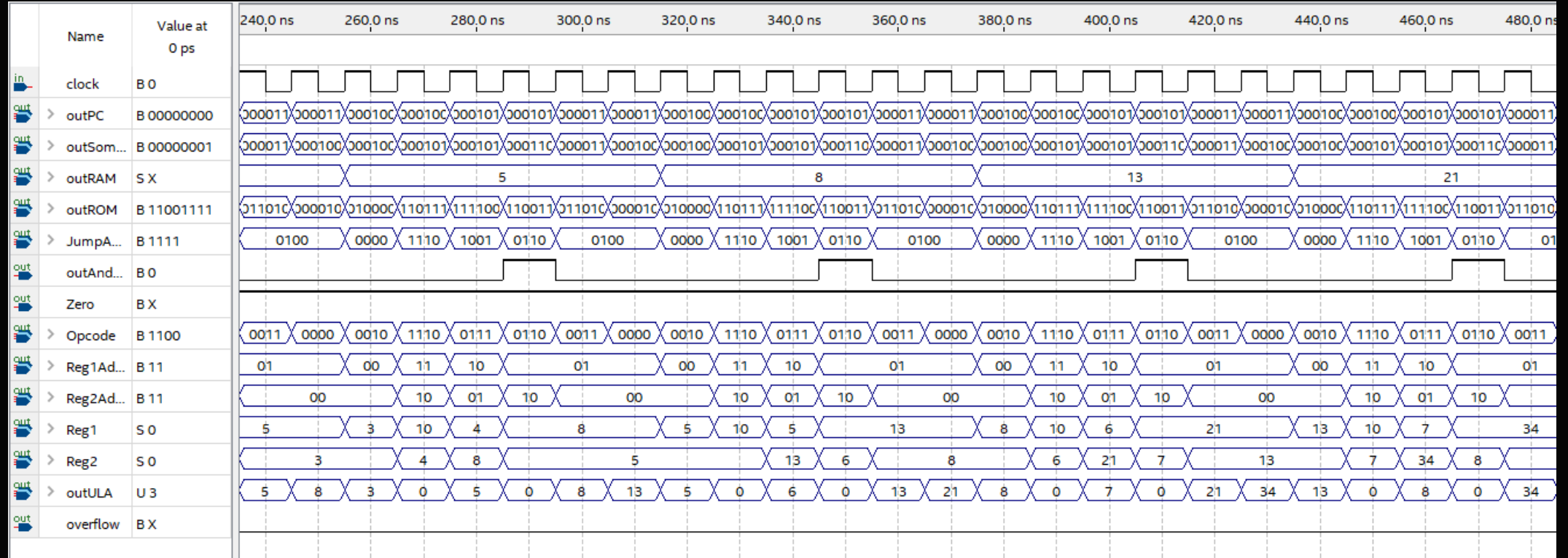
# Teste de Fibonacci

Endereço	Instrução(Octal)	Instrução(Binário)		
		OpCode	Reg1	Reg2
			Endereço	
0	Li \$s3, 3	1100	11	11
1	Mult \$s3, \$s3	1000	11	11
2	Addi \$s3, 1	0111	11	01
3	Li \$s2, 1	1100	10	01
4	Li \$s0, 1	1100	00	01
5	Move \$s1, \$s0	1101	01	00
6	Sw \$s1, 0	00110	01	00
7	Add \$s1, \$s0	0000	01	00
8	Lw \$s0, 0	0010	00	00
9	JumpC \$s3, \$s2	1110	11	10
10	Addi \$s2, 1	0111	10	01
11	Bne 6	0110	0110	

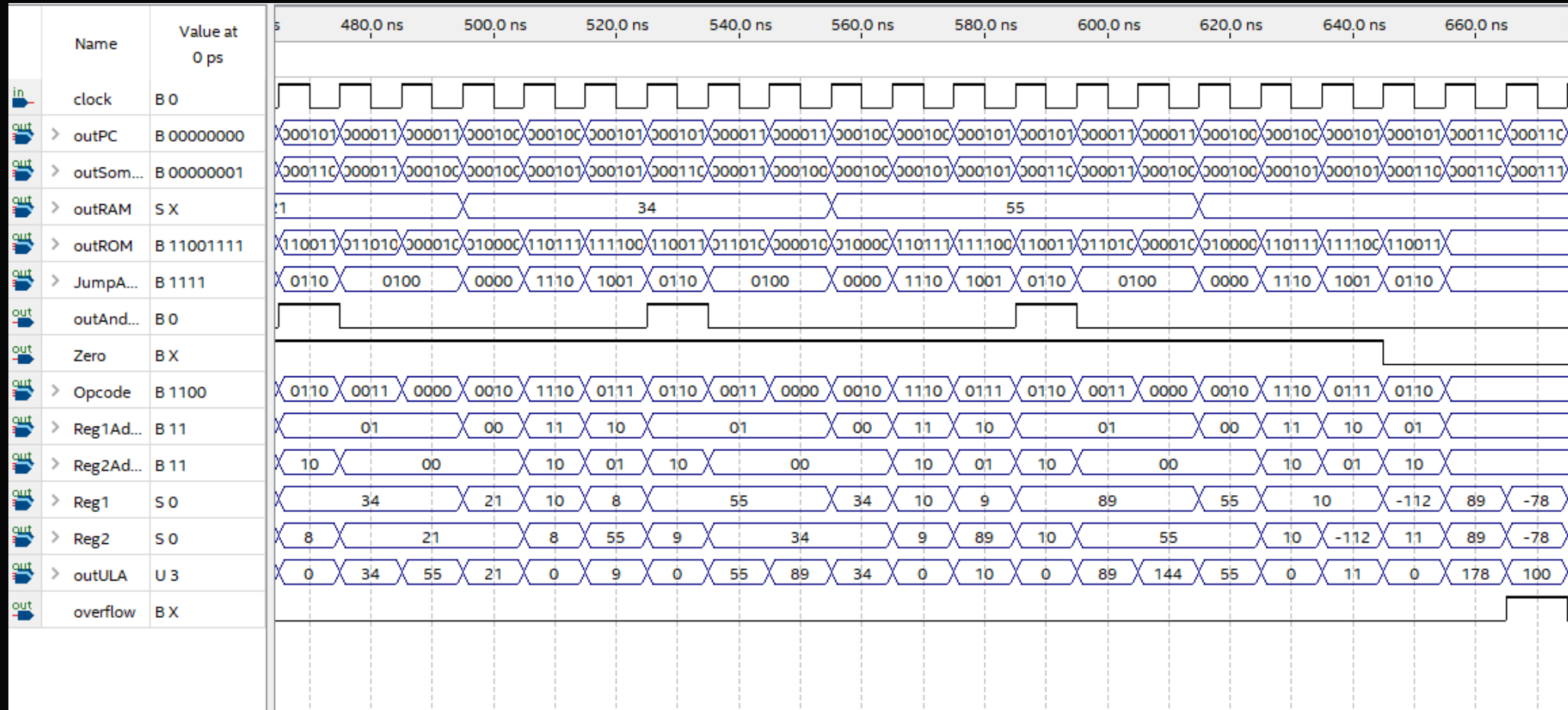
# Teste de Fibonacci



# Teste de Fibonacci



# Teste de Fibonacci



# Referências Bibliográficas

- A. PATTERSON, David; HENNESSY, John. Organização e Projeto de Computadores: A interface Hardware/software. 4<sup>a</sup>. ed. [S. l.]: Elsevier, 2013. 736 p.



O Cara não faz o curso  
e quer entender a  
matéria KKKKKKKK