



PROCESSADOR ONE PIECE



Fernando Souza Rodrigues
Paulo Ferreira da Silva Júnior

RESUMO

- Microprocessador em MIPS utilizando uma arquitetura alternativa voltada para 8 bits em consonância com a sua versão mais utilizada de 32 bits.
- Todas suas operações são realizadas utilizando números inteiros



FORMATOS DAS INSTRUÇÕES

INSTRUÇÕES DO TIPO R

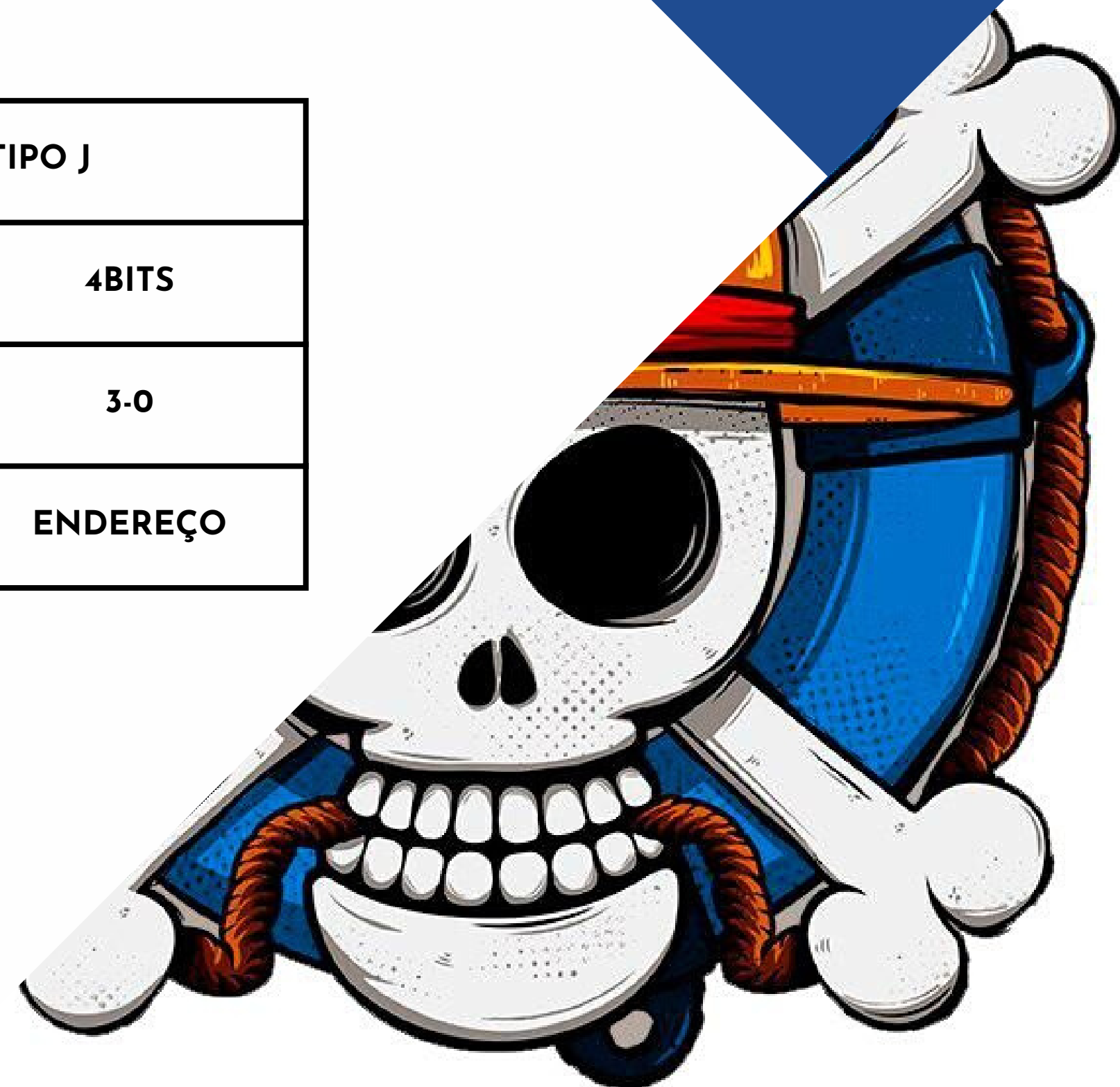
4BITS	2BITS	2BITS
7-4	3-2	1-0
OPCODE	REG1	REG2

INTRUÇÕES DO TIPO I

4BITS	2BITS	2BITS
7-4	3-2	1-0
OPCODE	REG1	REG2

INSTRUÇÕES DO TIPO J

4BITS	4BITS
7-4	3-0
OPCODE	ENDEREÇO





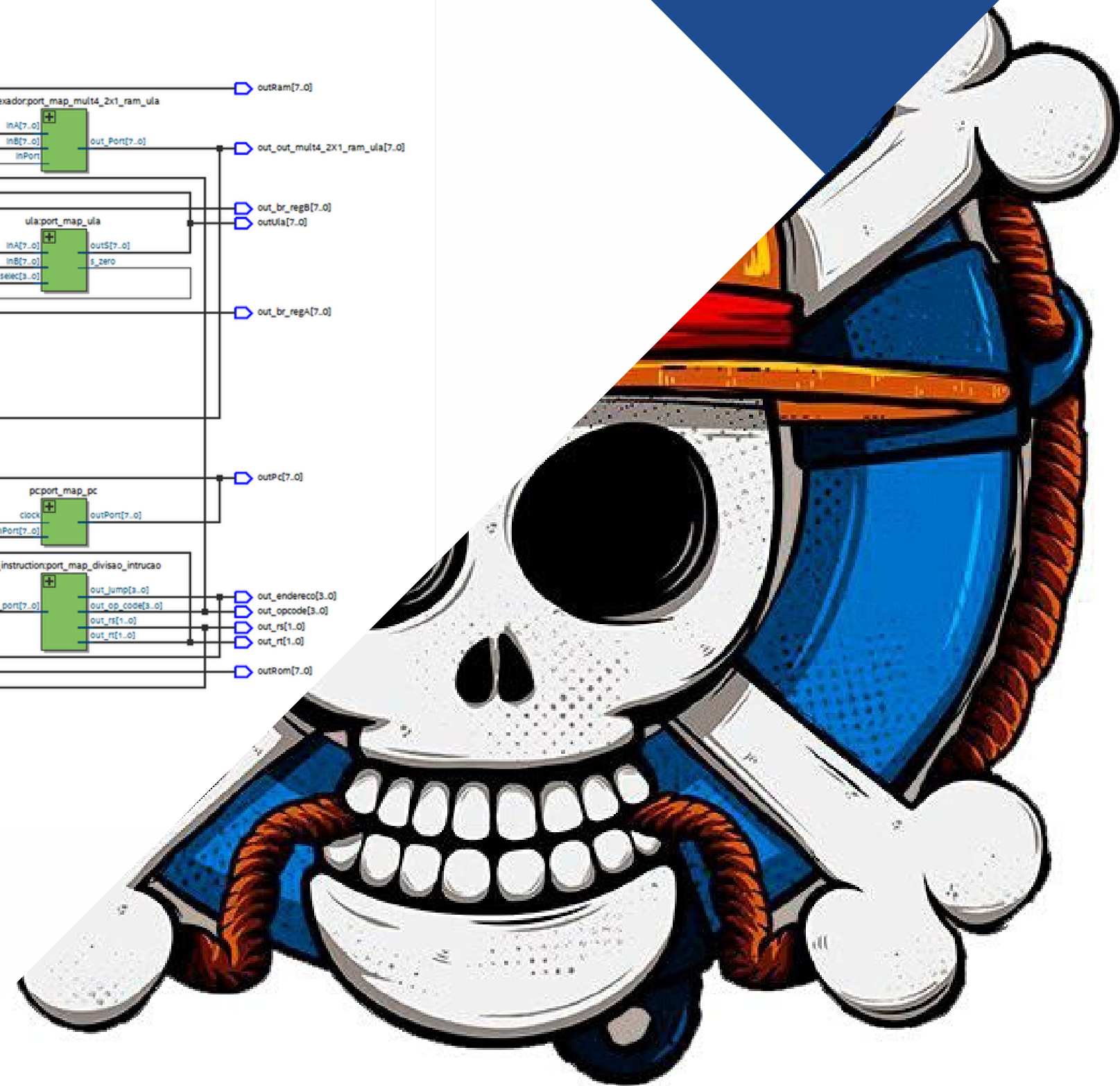
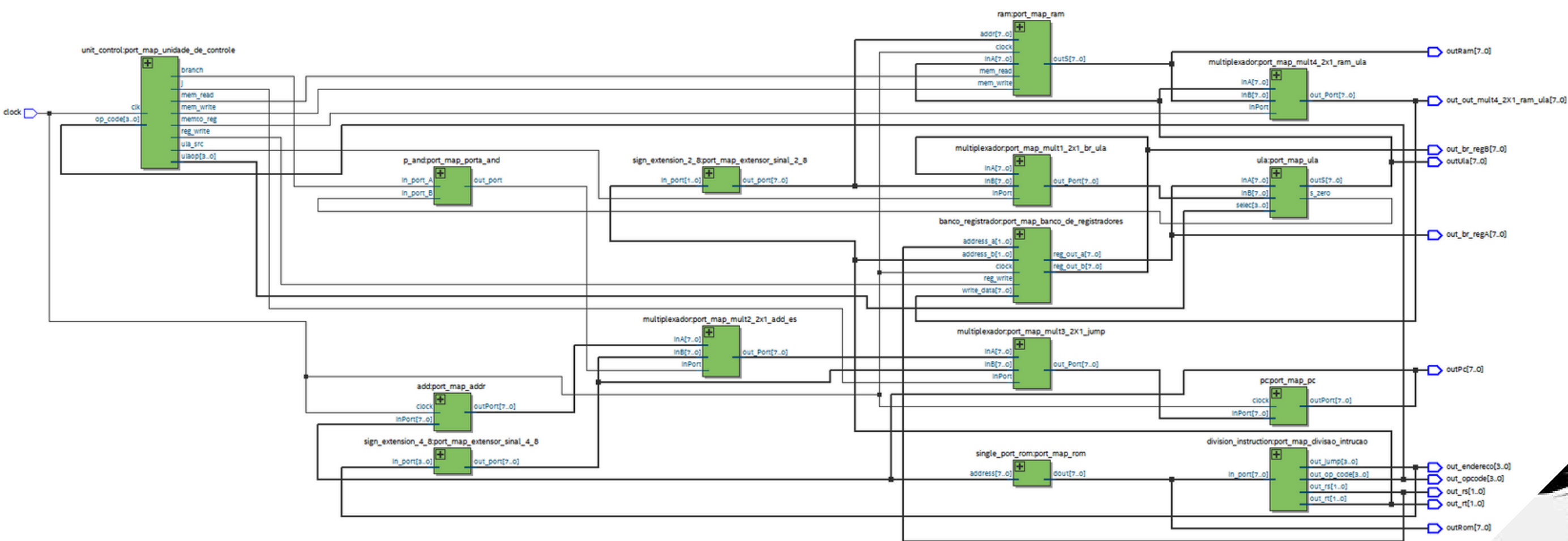
CONJUNTO DE INSTRUÇÕES



OPCODE	NOME	FORMATO	INSTRUÇÃO	EXEMPLO
0000	ADD	R	SOMA	ADD \$S0, \$S1
0001	ADDI	I	SOMA IMEDIATA	ADDI \$S0, 2
0010	SUB	R	SUBTRAÇÃO	SUB \$S0, \$S1
0011	SUBI	I	SUBTRAÇÃO IMEDIATA	SUBI \$S0, 5
0100	MUL	R	MULTIPLICAÇÃO	MUL \$S0, \$S1
0101	LW	I	LOAD WORD	LW \$S0 MEM (00)
0110	SW	I	STORE WORD	SW \$S0 MEM (00)
0111	MOVE	R	MOVER	MOVE \$S0, \$S1
1000	li	I	LOAD IMEDIATO	LI \$S0 1
1001	BEQ	J	DESVIO CONDICIONAL	BEQ ENDEREÇO
1010	BNE	J	DESVIO CONDICIONAL	BNE ENDEREÇO
1011	IF BEQ E BNE	R	CONDIÇÃO PARA DESVIO	IF \$S0 \$S1
1100	J	J	DESVIO INCONDICIONAL	J ENDEREÇO (0000)

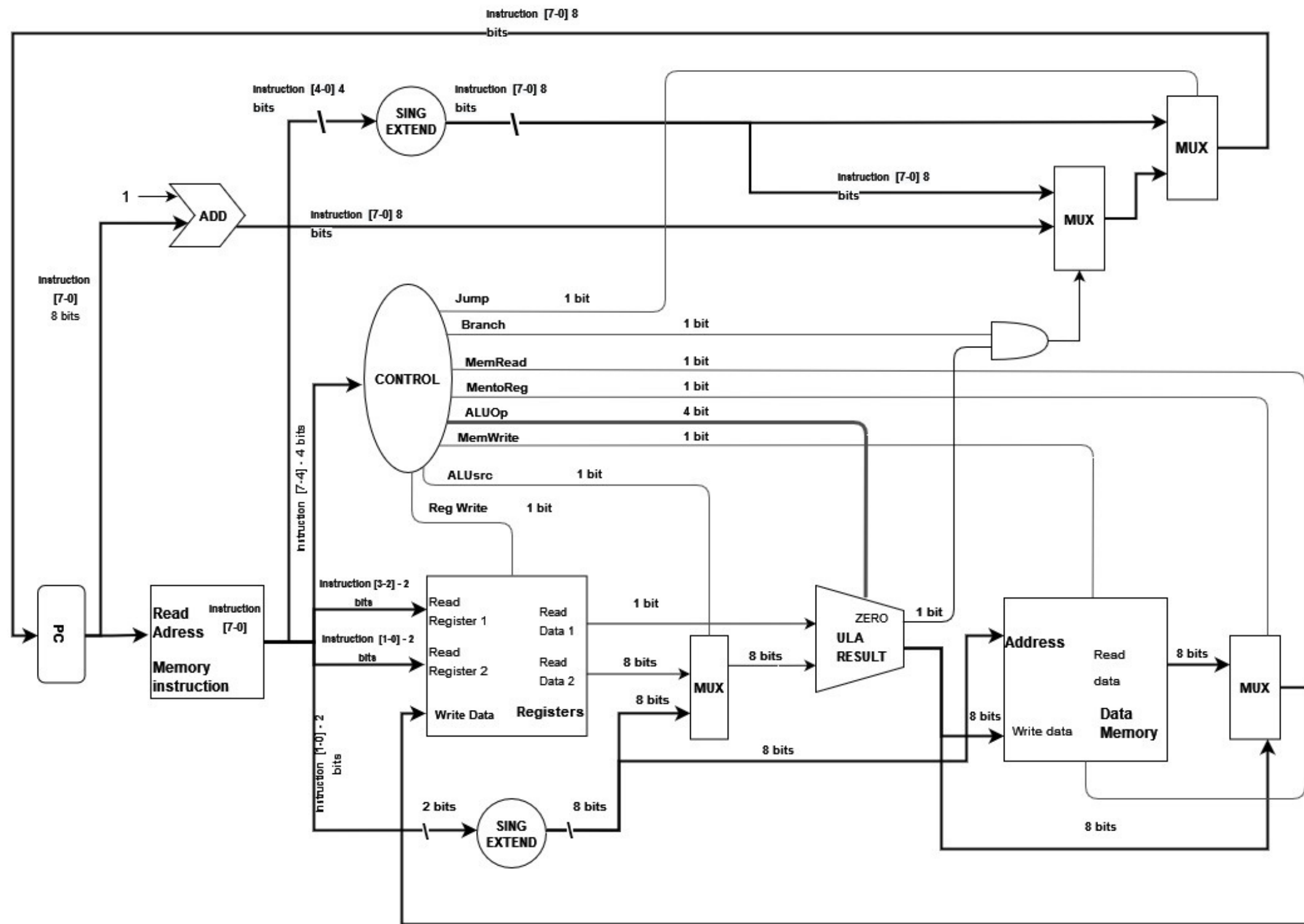


RTL VIEWER DO PROCESSADOR





DATAPATH





UFRR

CÓDIGO 1 - TESTE



1		0 => "10001111", --li S3 3
2		1 => "01001111", --mul S3 S3
3		2 => "00011101", --addi S3 1
4		3 => "00011110", --addi S3 2
5		4 => "00011110", --addi S3 2
6		5 => "10001000", --li S2 0
7		6 => "10000000", --li S0 0
8		7 => "01100000", --sw S0 RAM(00)
9		8 => "10000001", --li S0 1
10		9 => "01100001", --sw S0 RAM(01)
11		10 => "01010000", --lw S0 RAM(00)
12		11 => "00000100", --add S1 S0
13		12 => "01010001", --lw S0 RAM(01)
14		13 => "00000100", --add S1 S0
15		14 => "01100000", --sw S0 RAM(00)
16		15 => "01100101", --sw S1 RAM(01)
17		16 => "00011001", --addi S2 1
18		17 => "10111011", --if S2 S3
19		18 => "10101010", -- bne 1010



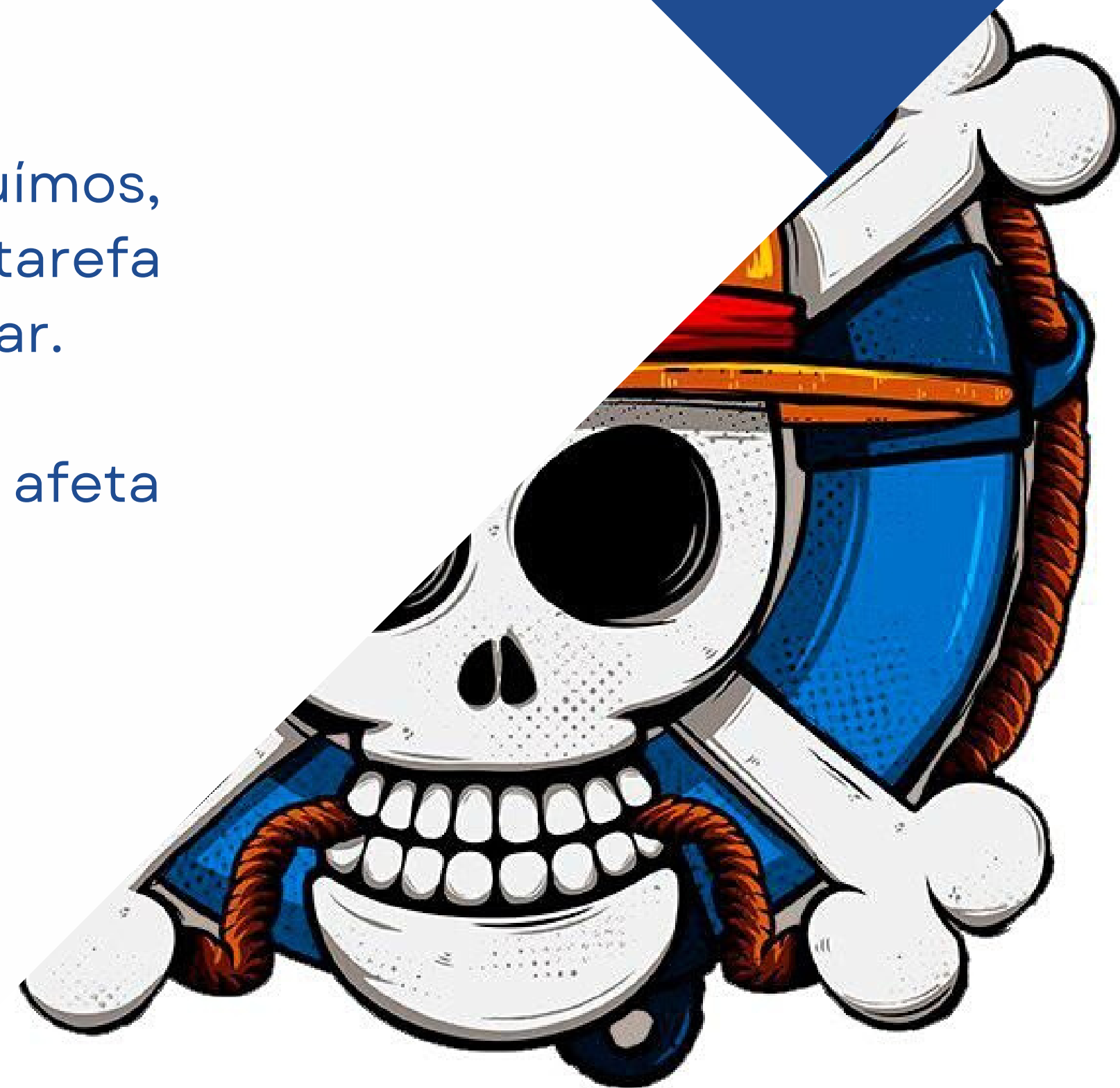
CÓDIGO 2 - TESTE

1		-- Fatorial
2		0 => "10001111", -- li S3 3
3		1 => "00011111", -- addi S3 3 == 6
4		2 => "10001001", -- li S2 1
5		3 => "10000001", -- li S0 1
6		4 => "01000010", -- mul S0 S2
7		5 => "10111011", -- if S2 == S3
8		6 => "00011001", -- addi S2 1
9		7 => "10100100", -- bne S2 != S3 jump 0100
10		8 => "10000000", -- li S0 0
11		9 => "10000100", -- li S1 0
12		10=> "10001000", -- li S2 0
13		OTHERS=> "00000000",



LIMITES DO PROCESSADOR

- Dada a pouca quantidade de Bits que possuímos, trabalhar com divisão acaba se tornando uma tarefa difícil, por este motivo decidimos por não adicionar.
- Operações de multiplicação têm um limite e isso afeta nos testes de fatorial, no caso indo apenas até 5



REFERÊNCIAS

- Livro: Arquitetura e Organização de Computadores - 8ª Edição - Autor: William Stallings
- Livro: Organização de Projetos de Computadores - 4ª Edição - Autor: David A. Patterson e John L. Hennessy

