



KRAKEN 2.0

Débora Bianca & Ewelly Fabiane



PROCESSADOR KRACKEN 2.0

- Possui 16 bits;
- É uniciclo;
- Baseado na arquitetura MIPS;
- Possui 16 registradores;

Nome	Número	Uso
\$0	0	Constante 0
\$v0	1	Resultados e avaliações de expressões
\$a0	2	Argumentos
\$a1	3	Argumentos
\$a2	4	Argumentos
\$a3	5	Argumentos
\$s0	6	Salvos
\$s1	7	Salvos
\$s2	8	Salvos
\$s3	9	Salvos
\$t0	10	Temporários
\$t1	11	Temporários
\$t2	12	Temporários
\$t3	13	Temporários
\$ra	14	Endereço de retorno
\$sp	15	Ponteiro para a pilha

FORMATOS DE INSTRUÇÃO

KRAKEN 2.0

- ◆ Tipo R (lógica e aritmética):

Opcode	rs	rd	Funct
4 bits	4 bits	4 bits	4 bits
0 - 3	4 - 7	8 - 11	12 - 15

- ◆ Tipo I (transferência de dados):

Opcode	rs	rt	Endereço
4 bits	4 bits	4 bits	4 bits
0 - 3	4 - 7	8 - 11	12 - 15

- ◆ Tipo J (desvio condicional):

Opcode	Endereço
4 bits	12 bits
0 - 3	4 - 15

INSTRUÇÕES KRAKEN 2.0

Opcode é igual a 4 bits, ou seja, temos um total de 16 bits, além do campo **Funct** que é composto por 4 bits no processador Kraken 2.0, dos 16 bits de opcode, 15 são distribuídos entre operações do tipo **I** e **J**, e 1 bit é voltado para operações do tipo **R**, que com auxílio do campo funct aumenta o número de operações do tipo R para 16.

INSTRUÇÕES KRACKEN 2.0

Opcode	Funct	Nome	Formato	Breve descrição
0000	0000	ADD	R	Soma
0000	0001	SUB	R	Subtração
0000	0010	MULT	R	Multiplicação
0000	0011	AND	R	Porta lógica AND
0000	0100	OR	R	Porta lógica OR
0000	0101	XOR	R	Porta lógica XOR
0000	0110	NOR	R	Porta lógica NOR
0000	0111	NAND	R	Porta lógica NAND
0000	1000	NOR	R	Porta lógica XOR
0000	1001	XNOR	R	Porta lógica XNOR
0000	1010	COMPARADOR	R	Porta lógica que compara se o primeiro valor é menor que o segundo
0000	1011	MOVE	R	Move um dado para o registrador
0000	1100	SLT	R	Set if less than
0000	1101	SRA	R	Shift aritmético para direita
0000	1111	<u>BoothMultiplier</u>	R	Realiza a multiplicação utilizando o algoritmo de Booth
0001	x	LW	I	Lê uma palavra da memória
0010	x	SW	I	Guarda uma palavra na memória
0011	x	BEQ	I	branch on equal
0100	x	BNE	I	branch on not equal
0101	x	MOVI	I	Move um dado imediato para o registrador
0110	x	SLTI	I	Set if less than immediate
0111	x	ADDI	I	Soma imediata
1000	x	ADDIU	I	Soma imediata sem overflow
1001	x	J	J	Salto
1010	x	JR	J	Jump register
1011	x	JAL	J	Jump and link

COMPONENTES

KRAKEN 2.0

- Implementados em VHDL na IDE Quartus Prime Lite Edition 17.0.
- ◆ And, Or, Not, Xor;
 - ◆ RightShift e LeftShift;
 - ◆ Comparador;
 - ◆ BoothMultiplier;
 - ◆ Multiplexador3_8;
 - ◆ Unidade Lógica Aritmética (ULA);
 - ◆ Somador;
 - ◆ Contador do PC;
 - ◆ Memória ROM;
 - ◆ Memória RAM;
 - ◆ Banco de Registradores;
 - ◆ Registrador Flip-Flop tipo D;
 - ◆ Extender de 4 para 16;
 - ◆ Unidade de Controle (UC);

ULA Kraken 2.0

Tabela de Instruções da ULA

INSTRUÇÃO	SELETOR 1	SELETOR 2	SELETOR 3	Negate
AND	0	0	0	0
NAND	0	0	0	1
OR	0	0	1	0
NOR	0	0	1	1
XOR	0	1	0	0
XNOR	0	1	0	1
ADD	0	1	1	0
SUB	0	1	1	1
Compara	1	0	0	x
ShiftLeft2	1	0	1	x
ShiftRigth2	1	1	0	x
Mul	1	1	1	x

Declaração da Unidade de Controle

```

ENTITY Unidade_de_controle is
    Port (
        opcode : in STD_LOGIC_VECTOR(3 downto 0);
        funct  : in STD_LOGIC_VECTOR(3 downto 0);
        ulaop  : out std_logic_vector(2 downto 0);
        jump, aluSrc, memWrite, memRead, memToReg, branch : out std_logic;
        regWrite, regDest : out std_logic
    );
END Unidade_de_controle;
```

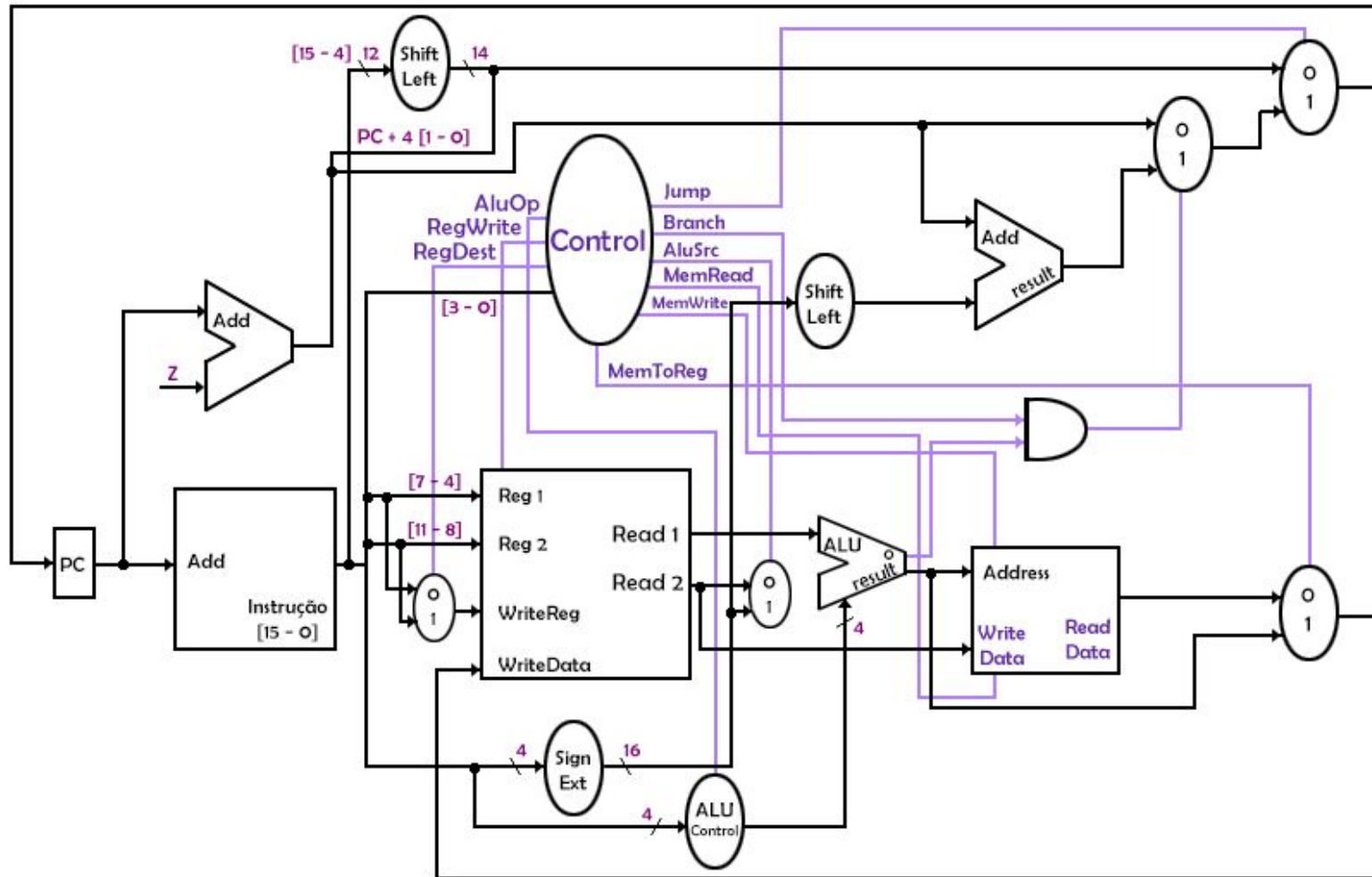
UC
Kraken 2.0

Flags da unidade de controle do Kraken 2.0

SINAIS DE CONTROLE UNICICLO										
Instrução	Opcode	RegWrite	ULAOp	ULANeg	ALUSrc	MemWrite	MemRead	MemParaReg	Branch	Jump
AND	0000	1	000	0	0	0	0	0	0	0
NAND	0000	1	000	1	0	0	0	0	0	0
OR	0000	1	001	0	0	0	0	0	0	0
NOR	0000	1	001	1	0	0	0	0	0	0
XOR	0000	1	010	0	0	0	0	0	0	0
XNOR	0000	1	010	1	0	0	0	0	0	0
ADD	0000	1	011	0	0	0	0	0	0	0
SUB	0000	1	011	1	0	0	0	0	0	0
COMPARADOR	0000	1	100	0	0	0	0	0	0	0
LeftShift2	0000	1	101	0	0	0	0	0	0	0
RigthShift2	0000	1	110	0	0	0	0	0	0	0
BoothMultiplier	0000	1	111	0	0	0	0	0	0	0
MOVE	0000	1	011	0	0	0	0	0	0	0
lw	0001	0	011	0	1	0	1	1	0	0
sw	0010	0	011	0	1	1	0	0	0	0
beq	0011	0	011	1	0	0	0	0	1	0
bne	0100	0	011	1	0	0	0	0	1	0
movi	0101	1	011	0	1	0	0	0	0	0
slti	0110	0	100	x	1	0	0	0	0	0
addi	0111	1	011	0	1	0	0	0	0	0
addiu	1000	1	011	0	1	0	0	0	0	0
j	1001	0	0	x	0	0	0	0	0	1
jr	1010	0	0	x	0	0	0	0	0	1
jal	1011	0	0	x	0	0	0	0	0	1



DATAPATH





SIMULAÇÕES & TESTES



PROF VOCÊ É
TOP

