

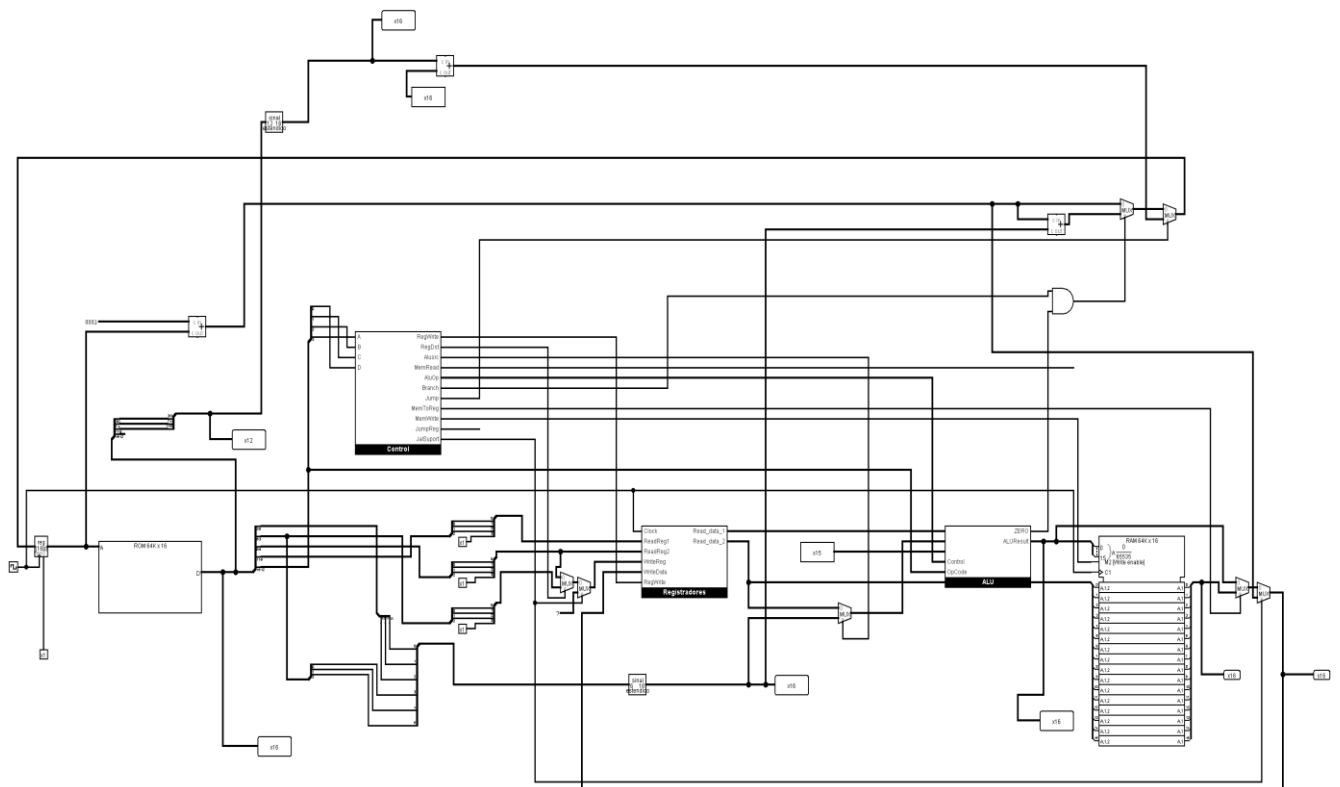
Relatório

Equipe:

João Edson De Sousa

Isac Peixoto Costa

Nome do processador: RAMEDICS



(16-Bit Instruction format)

	Opcode	Rs	Rt	Rd	Shamt	Funct
R-type	0000	000	000	000	0	00

	Opcode	Rs	Rt	Immediate value
I-type	0000	000	000	000000

	Opcode	Immediate value
J-type	0000	000000000000

(Opcode for the instructions)

Instruction	Opcode
Add	0010
Addi	0011
Slt	0000
Slti	0001
Beq	0100
Sll	0101
Lw	1000
Sw	1001
J	1111
Jal	1011
Jr	1110

(AluOP)

Operation	AluOP
Add	00
Sub	01
Sll	10

(Registers)

Register	number
\$Zero	000
\$a0	001
\$a1	010
\$t0	011
\$t1	100
\$t2	101
\$t3	110
\$ra	111

(Code)

Assembly:

.data

input: .word 1,2,30,4,5

length: .word 16

.text

lw \$a1, length(\$0) #length -> j

lw \$a0, input(\$0) #current element

beq \$a1, \$t0, end1

addi \$t0, \$0, 4 # current -> i

lw \$t1, input(\$t0) #next element

jal sort

end1:

li \$v0, 10

syscall

sort:

addi \$t7, \$0, 1

slt \$t6, \$a1, \$t0

beq \$t6, \$t7, end

slt \$t2, \$a0, \$t1

addi \$t3, \$0, 1

beq \$t2, \$t3, gte

beq \$a1, \$t0, end

addi \$t0, \$t0, 4

lw \$t1, input(\$t0)

j sort

gte:

addi \$a0, \$t1, 0

addi \$t0, \$t0, 4

lw \$t1, input(\$t0)

j sort

end:

li \$v0, 1

syscall

jr \$ra

Hexadecimal | Binary | Compiler:

8086 -> 1000 000 010 000110 -> lw \$a1, 6(\$0)

8040 -> 1000 000 001 000000 -> li \$a0, 0(\$0)

44d1 -> 0100 010 011 010001 -> beq \$t0, \$a1, 010001

30c1 -> 0011 000 011 000001 -> addi \$t0, \$0, 1

8700 -> 1000 011 100 000000 -> lw \$t1, 0(\$t0)

b006 -> 1111 000000000110 -> j 000000000110

31c1 -> 0011 000 111 000001 -> addi \$ra, \$0, 1

04c1 -> 0000 010 011 000 0 01 -> slt \$0, \$a1, \$t0

41FF -> 0100 000 111 111111 -> beq \$0, \$ra, 111111

0329 -> 0000 001 100 101 0 01 -> slt \$t2, \$t1, \$a0

3181 -> 0011 000 110 000001 -> addi \$t3, \$0, 1

4b85 -> 0100 101 110 000100 -> beq \$t2, \$t3, 000100

44FF -> 0100 010 011 111111 -> beq \$a1, \$t0, 111111

36c1 -> 0011 011 011 000001 -> addi \$t0, \$t0, 1

8700 -> 1000 011 100 000000 -> lw \$t1, 0(\$t0)

f006 -> 1111 000000000110 -> j 000000000110

3840 -> 0011 100 001 000000 -> addi \$a0, \$t1, 0

36C1 -> 0011 011 011 000001 -> addi \$t0, \$t0, 1

8700 -> 1000 011 100 000000 -> lw \$t1, input(\$t0)

f006 -> 1111 000000000110 -> j 000000000110

Observações gerais:

-Foi utilizado um vetor de 5 posições para encontrar o maior elemento.

-Apesar de possuir suporte para Jal, Jr, sw e Slti não foram utilizadas nenhuma dessas instruções.