Relatório do Projeto: ISA 12 BITS  
  
Pedro Lucas Alves de Oliveira   
GRR: 20232366  
  
16 de junho de 2025

**1. Especificações da ISA**

1.1. Convenção e declaração dos registradores

|  |  |  |  |
| --- | --- | --- | --- |
| Registrador | Convenção | Endereço Binário | ---------------------------------- |
| r0 | x0 | 0 0 0 0 | ZERO |
| r1 | ra | 0 0 0 1 | Return Adress |
| r2 | sp | 0 0 1 0 | Stack Pointer |
| r3 | t0 | 0 0 1 1 | Registrador temporário. |
| r4 | t1 | 0 1 0 0 | Registrador temporário. |
| r5 | t2 | 0 1 0 1 | Registrador temporário. |
| r6 | t3 | 0 1 1 0 | Registrador temporário. |
| r7 | t4 | 0 1 1 1 | Registrador temporário. |
| r8 | s0 | 1 0 0 0 | Registrador salvo. |
| r9 | s1 | 1 0 0 1 | Registrador salvo. |
| r10 | s2 | 1 0 1 0 | Registrador salvo. |
| r11 | s3 | 1 0 1 1 | Registrador salvo. |
| r12 | a0 | 1 1 0 0 | Argumento/retorno da função. |
| r13 | a1 | 1 1 0 1 | Argumento. |
| r14 | a2 | 1 1 1 0 | Argumento. |
| r15 | a3 | 1 1 1 1 | Argumento. |

1.2 Tabela/ordem das instruções

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ordem | Instrução | Opcode | Tipo | Instrução' |
| 1 | add | 0 0 0 0 | P | X = X + Y |
| 2 | sub | 0 0 0 1 | P | X = X - Y |
| 3 | mul | 0 0 1 0 | P | X = X \* Y |
| 4 | div | 0 0 1 1 | P | X = X / Y |
| 5 | sll | 0 1 0 0 | P | X = X << Y |
| 6 | and | 0 1 0 1 | P | X = X & Y |
| 7 | or | 0 1 1 0 | P | X = X or Y |
| 8 | xor | 0 1 1 1 | P | X = X xor Y |
| 9 | addi | 1 0 0 0 | L | X = X + imm |
| 10 | lw | 1 0 0 1 | L | X = Mem[imm] |
| 11 | sw | 1 0 1 0 | L | Mem[imm] = X |
| 12 | jal | 1 0 1 1 | L | r1 = PC + 1; PC = imm |
| 13 | jalr | 1 1 0 0 | L | r1 = PC + 1; PC = X + imm |
| 14 | blt | 1 1 0 1 | L | if (X < imm) PC = PC + imm |
| 15 | bge | 1 1 1 0 | L | if (X >= imm) PC = PC + imm |
| 16 | ebreak | 1 1 1 1 | L | STOP |

1.3 Formatos de instrução

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO P | Bits 11-8 | Bits 7-4 | Bits 3-0 |
|  | OPCODE | Y | X |
|  |  |  |  |
| TIPO L | Bits 11-8 | Bits 7-4 | Bits 3-0 |
|  | OPCODE | IMM | X |

**2. Especificações da Memória**

2.1 Assembly

addi R3, 5 fat:

addi R4, 2 addi R2, -1

mul R5, R3, R4 sw R1, 0

div R5, R5, R4 addi R2, -1

sll R5, R5, R0 sw R12, 0

and R5, R3, R4 addi R3, R12, 0

or R5, R3, R4 addi R5, 1

xor R5, R3, R4 sub R4, R3, R5

blt R3, R4, 2 bge R4, 0, 3

addi R5, 1 addi R12, 1

jalr R0, 2 jalr R0, 4

addi R5, 0 addi R12, R4, 0

addi R6, 3 jal fat

sw R6, 0 mul R12, R3

addi R12, 0 lw R12, 1

add R12, R6 addi R2, 1

jal fat lw R1, 0

addi R7, R12, 0 addi R2, 1

sw R7, 1 jalr R1, 0

ebreak

2.2 C

int fatorial(int n) {

if (n <= 1) {

return 1;

} else {

return n \* fatorial(n - 1);

}

}

void main() {

int numero = 3;

int resultado;

int a = 5;

int b = 2;

int c;

c = a \* b;

c = c / b;

c = c << 1;

c = a & b;

c = a | b;

c = a ^ b;

if (a < b) {

c = 0;

} else {

c = 1;

}

resultado = fatorial(numero);

}

2.3 Hex

0x835

0x842

0x253

0x354

0x450

0x553

0x653

0x753

0xD32

0x850

0xC02

0x850

0x863

0xA60

0x8C0

0x0C6

0xB07

0x87C

0xA71

0xF00

0x82F

0xA10

0x82F

0xAC0

0x83C

0x851

0x143

0xE43

0x8C1

0xC04

0x8C4

0xB0A

0x2C3

0x9C1

0x821

0x910

0x821

0xC10  
  
  
2.4 Binário

100000110101

100001000010

001001010011

001101010100

010001010000

010101010011

011001010011

011101010011

110100110010

100001010000

110000000010

100001010000

100001100011

101001100000

100011000000

000011000110

101100000111

100001111100

101001110001

111100000000

100000101111

101000010000

100000101111

101011000000

100000111100

100001010001

000101000011

111001000011

100011000001

110000000100

100011000100

101100001010

001011000011

100111000001

100000100001

100100010000

100000100001

110000010000

**3. Unidade de Controle**  
  
3.1 Hex

0x80

0x82

0x84

0x86

0x88

0x8A

0x8C

0x8E

0xC0

0xA0

0x10

0x200

0x600

0x500

0x501

0x0