

## Green Card da ISA:

Opcode	Tipo	Mnemonic	Nome	Operação
0000	B	blt	Branch Less Than	if ( $R[rs1] < R[rs2]$ ) $PC = PC + IMM$
0001	J	jal	Jump And Link	$R[rd] = PC + 1$ ; $PC = PC + IMM$
0010	I	ld	Load	$R[rd] = M[R[rs1] + IMM]$
0011	B	st	Store	$M[R[rs1] + IMM] = R[rs2]$
0100	J	addi	Add Immediate	$R[rd] = R[rd] + IMM$
0101	I	jalr	Jump And Link Register	$R[rd] = PC + 1$ ; $PC = R[rs1]$
0110	B	bge	Branch Greater Than or Equal	if ( $R[rs1] \geq R[rs2]$ ) $PC = PC + IMM$
0111	B	beq	Branch Equal	if ( $R[rs1] == R[rs2]$ ) $PC = PC + IMM$
1000	R	not	Not	$R[rd] = \text{not } R[rs1]$
1001	R	and	And	$R[rd] = R[rs1] \text{ and } R[rs2]$
1010	R	or	Or	$R[rd] = R[rs1] \text{ or } R[rs2]$
1011	R	xor	Xor	$R[rd] = R[rs1] \text{ xor } R[rs2]$
1100	R	add	Add	$R[rd] = R[rs1] \text{ add } R[rs2]$
1101	R	sub	Sub	$R[rd] = R[rs1] \text{ sub } R[rs2]$
1110	I	slli	Shift Left Logical Immediate	$R[rd] = R[rs1] \ll IMM$
1111	I	srl	Shift Right Logical Immediate	$R[rd] = R[rs1] \gg IMM$

Tipo R											
11	10	9	8	7	6	5	4	3	2	1	0
rs2		rs1			rd			OPCODE			

Tipo I											
11	10	9	8	7	6	5	4	3	2	1	0
IMM			rs1			rd			OPCODE		

Tipo B											
11	10	9	8	7	6	5	4	3	2	1	0
IMM		rs2			rs1			OPCODE			

Tipo J											
11	10	9	8	7	6	5	4	3	2	1	0
IMM					rd			OPCODE			

## Descrição da ISA:

É uma ISA de 12 bits, com 8 registradores, 16 operações e 4 tipos de formato para o dado, sendo eles Register(R), Branch(B), Jump(J) e Immediate(I). Informações notáveis:

- Nos formatos B e I os imediatos que serão inseridos devem estar entre o intervalo de 0 a 3.
- As memórias ROM e RAM são indexadas a cada 12 bits.
- O registrador X0 é uma constante zero.

## Registradores:

São oito registradores no total, seguindo a seguinte convenção:

- X0 ou Zero: constante zero ;
- X1 ou A0: argumento para função e valor de retorno ;
- X2 ou A1: argumento para função ;
- X3 ou T1: registrador temporário ;
- X4 ou T2: registrador temporário ;

- X5 ou T3: registrador temporário ;
- X6 ou T4: registrador temporário ;
- X7 ou RA: registrador salvo que guarda o endereço de retorno ;