ISA 24bits

Instruction Formats

R (register):

unused	rs2	rs1	funct3	rd	opcode
7b	4b	4b	3b	4b	2b

I (immediate):

imm[10:0]		rs1	funct3	rd	opcode
11b		4b	3b	4b	2b

V (void):

imm[10:4]	rs2	rs1	funct3	imm[3:0]	opcode
7b	4b	4b	3b	4b	2b

L (load):

imm[15:0]			unused	rd	opcode
16b			2b	4b	2b

Instructions

Format R

```
R [00 000] add – addition
R [00 001] sub – subtraction
R [00 010] mul – multiplication
R [00 011] div – division
R [00 100] mod – remainder

R [00 101] and – logical and
R [00 110] or – logical or
R [00 111] xor – logical xor
```

Format I

```
I [01 000] addi – add immediate

I [01 001] lw – load word
I [01 010] jalr – jump and link register

I [01 011] slli – shift left logical immediate

I [01 100] modi – remainder immediate
I [01 101] andi – logical and immediate
I [01 110] ori – logical or immediate
I [01 111] xori – logical xor immediate
```

Format V

```
V [10 000] sw - store word

V [10 001] halt - halt the program execution

V [10 100] beq - branch if equal
V [10 101] bne - branch if not equal
V [10 110] bge - branch if greater or equal
V [10 111] blt - branch if less than
```

Format L

```
L [11] li - load immediate
```

Registers

```
r0       : r0       (zero)
r1       : rad      (return address)
r2.      : rbp      (base pointer)
r3       : rsp      (stack pointer)
r4..r7   : rt0..rt3 (temporaries)
r8..r11  : rs0..rs3 (saved)
r12..r15 : ra0..ra3 (fn arguments / return value)
```