# Encoding

m.kim-4

October 2025

## 1 MAP to custom ISA

| **Instruction Type** | **Field Breakdown (bits)** |
|---|---|
| R-type | `[ op(4) \| rs(6) \| rt(6) \| rd(6) \| 00(2) ]` |
| I-type | `[ op(4) \| rs(6) \| rt(6) \| immediate(8) ]` |
| J-type | `[ op(4) \| target address(20) ]` |

Table 1: Instruction Format Summary for the Custom ISA

| **Mnemonic** | **Type** | **Opcode (4-bit)** | **Description** |
|---|---|---|---|
| `HALT` | J-type | 0000 | Stop program execution |
| `ADD` | R-type | 0001 | Add $R_s + R_t \rightarrow R_d$ |
| `MUL` | R-type | 0011 | Multiply $R_s \times R_t \rightarrow R_d$ |
| `LI` | I-type | 0100 | Load 8-bit immediate into register |
| `LOAD` | I-type | 0101 | Load word from memory $\rightarrow$ register |
| `STORE` | I-type | 0110 | Store register $\rightarrow$ memory |
| `BEQ` | I-type | 0111 | Branch if equal |
| `JMP` | J-type | 1000 | Jump to target address (relative) |
| `LUI` | I-type | 1001 | Load upper 8 bits into register |
| `ORI` | I-type | 1010 | OR immediate with register |

Table 2: Opcode Assignments for Custom ISA (Verilog-Compatible)