LOGIC REG2REG:
ADD, SUB, OR, XOR, NOR

2:0 = op_code (R-type)
5:3 = rs1
8:6 = rs2
11:9 = rd
15:12 = instruction

ADDI, SUBI, ORI, XORI, NORI
Logic IMMEDIATE (double word)
2:0 = op_code (I-type)
5:3 = rs1
11:9 = rd
15:12 = instruction
31:16 = IMM;

BRANCH
BEQ, BNE, BGE, BLT, BLTE

2:0 = B-type
15:13 = instruction
5:3 = rs1
8:6 = rs2
9:12 = branch offset

JUMP:
2:0 = J-type
5:3 = rd
6 = JAL/JALR
7:15 = offset (2 byte aligned so final offset is 8’b {offset, 0}

MEM (double word):
2:0 = mem-type (opcode)
5:3 = rs1
8:6 = rs2
11:9 = rd
15:12 =instr
16 = ldst
{[16:31]} = offset
LW, LB, LH, SW, SB, SH

SYS_CALL:
2:0 = sys_type