Truth Table for 19-bit CPU Architecture

Instruction	Opcode	RegWrite	AluSrc	MemWrite	ResultSrc	Branch	Jump	Call	Ret	ImmSrc	AluOp
R-type	00000	1	0	0	0	0	0	0	0	00	00
I-type (Load)	00001	1	1	0	1	0	0	0	0	01	00
S-type (Store)	00010	0	1	1	0	0	0	0	0	01	00
BEQ	00011	0	0	0	0	1	0	0	0	01	00
BNE	00100	0	0	0	0	1	0	0	0	01	00
JMP	00101	0	0	0	0	0	1	0	0	10	00
CALL	00110	0	0	0	0	0	1	1	0	10	00
RET	00111	0	0	0	0	0	1	0	1	00	00

PICT, PUNE

PICT, PUNE

Signals:-

1. RegWrite: Asserted (1) when instruction writes into a result (R type and Load)

2. ALUSrc:
- For I-type and S-type instructions, the second ALU operand comes from the sign extended immediate. (set to 1)
- For R-type and branch instructions, it comes from a register (set to 0)

3. MemWrite: Asserted (1) only from S-type instructions

4. ResultSrc: For Load type Instr. (I-type), the result is taken from memory (set to 1). Otherwise, the ALU result is used (set to 0).

5. Branch: Asserted (1) for BEQ & BNE to indicate branch.

6. Jump, Call, Ret: These signals are asserted when the respective instructions are called. They are used to update PC.

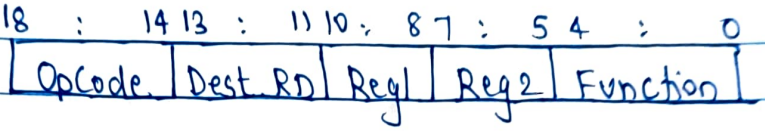
7. ImmSrc:
- "00" is used when an immediate is not used (R-type & RET)
- "01" is used for 8-bit sign-extended immediates (I, S, B type)
- "10" is used for 14 bit immediate for JMP & CALL.

8. ALUop: Currently ALUop is set to 0 for all above instruction as function field is used for conditioning.

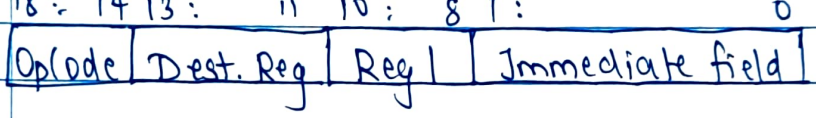
PICT, PUNE

Types of Instructions:-

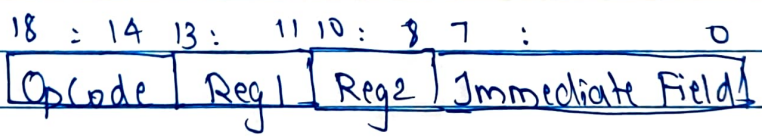
1. R-type (Arithmetic / Logical / Custom)



2. I type (Load)



3. S/B type



4. J-type (Jump / Call / Ret)