



Resumo dos formatos de codificação e opcodes

| | | | | | |
|---------------|--------|-------|-------|-----|-----|
| Field | opcode | Rm | shamt | Rn | Rd |
| Bit positions | 31:21 | 20:16 | 15:10 | 9:5 | 4:0 |

a. R-type instruction

| | | | | | |
|---------------|--------------|---------|-------|-----|-----|
| Field | 1986 or 1984 | address | 0 | Rn | Rt |
| Bit positions | 31:21 | 20:12 | 11:10 | 9:5 | 4:0 |

b. Load or store instruction

| | | | |
|---------------|-------|---------|-----|
| Field | 180 | address | Rt |
| Bit positions | 31:24 | 23:5 | 4:0 |

c. Conditional branch instruction

| | | |
|--------|---------|---|
| 31 | 26 25 | 0 |
| opcode | address | |
| 6 bits | 26 bits | |

d. Branch

| Instrução | Opcode | | |
|-----------|--------|------|------|
| ADD | 100 | 0101 | 1000 |
| SUB | 110 | 0101 | 1000 |
| AND | 100 | 0101 | 0000 |
| ORR | 101 | 0101 | 0000 |
| LDUR | 111 | 1100 | 0010 |
| STUR | 111 | 1100 | 0000 |
| CBZ | 101 | 1010 | 0 |
| B | 000 | 101 | |

ALU trabalha em 3 contextos diferentes.

1. instruções lógico-aritméticas: $ALUOp[1:0]=10$
2. cálculo de endereços: $ALUOp[1:0]=00$
3. comparação: $ALUOp[1:0]=01$