

opcodes

Bits of the instruction from left to right

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Description | syntax |
|--|---|---|--------------------|---|------------------|---|---|---|---|-------------------------------|---------------------------------|-------------------|
| 1 * * * * * * * * * * Load/Store | | | | | | | | | | | | |
| 1 | 1 | 0 | Address (8 bits) | | | | | | | Load A with content of Addr | mov A, [addr] | |
| 1 | 1 | 1 | Address (8 bits) | | | | | | | Load B with content of Addr | mov B, [addr] | |
| 1 | 0 | 0 | Address (8 bits) | | | | | | | Store Addr with content of A | mov [addr],A | |
| 1 | 0 | 1 | Address (8 bits) | | | | | | | Store Addr with content of B | mov [addr],B | |
| 0 0 * * * * * * * * * Load with Immediate value | | | | | | | | | | | | |
| 0 | 0 | 0 | Immediate (8 bits) | | | | | | | Load A with Immediate | mov A, Imm | |
| 0 | 0 | 1 | Immediate (8 bits) | | | | | | | Load B with Immediate | mov B, Imm | |
| 0 1 x 0 * * * x x x x Tests of values into W | | | | | | | | | | | | |
| 0 | 1 | x | 0 | 0 | 0 | 0 | x | x | x | x | <=0 | test <= 0 |
| 0 | 1 | x | 0 | 0 | 0 | 1 | x | x | x | x | XX | |
| 0 | 1 | x | 0 | 0 | 1 | 0 | x | x | x | x | <0 | test < 0 |
| 0 | 1 | x | 0 | 0 | 1 | 1 | x | x | x | x | != 0 | test != 0 |
| 0 | 1 | x | 0 | 1 | 0 | 0 | x | x | x | x | XX | |
| 0 | 1 | x | 0 | 1 | 0 | 1 | x | x | x | x | >0 | test > 0 |
| 0 | 1 | x | 0 | 1 | 1 | 0 | x | x | x | x | >=0 | test >=0 |
| 0 | 1 | x | 0 | 1 | 1 | 1 | x | x | x | x | =0 | test = 0 |
| 0 1 * 1 0 1 0 * * * * UAL work | | | | | | | | | | | | |
| 0 1 * 1 0 1 0 1 0 0 0 Load W | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | x | 0 | 0 | Load W with A | mov W, A |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | x | 0 | 0 | Load W with B | mov W, B |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | x | 0 | 1 | Load A with W | mov A, W |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | x | 0 | 1 | Load B with W | mov B, W |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | x | 1 | 0 | IO from A | mov IO, A |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | x | 1 | 0 | IO from B | mov IO, B |
| 0 | 1 | x | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | or A,B | or A, B |
| 0 | 1 | x | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | and A,B | and A, B |
| 0 | 1 | x | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | xor A,B | xor A, B |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | not A | not A |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | not B | not B |
| 0 | 1 | x | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | add A, B | add A, B |
| 0 | 1 | x | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | sub A,B | sub A, B |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | shl A | shl A |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | shl B | shl B |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | shr A | shr A |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | shr B | shr B |
| 0 1 x 1 * * * * * * IP management | | | | | | | | | | | | |
| 0 | 1 | x | 1 | 0 | 1 | 1 | 0 | x | x | x | Skip next instruction if Status | sis (skip if set) |
| 0 | 1 | x | 1 | 0 | 1 | 1 | 1 | x | x | x | Return from subroutine | ret |
| 0 | 1 | 0 | 1 | 1 | Address (6 bits) | | | | | Relative jump (extended addr) | jmp rel | |
| 0 | 1 | 1 | 1 | 1 | Address (6 bits) | | | | | Relative Call (extended addr) | call rel | |

opcodes

schema

Remark

[redacted] * Mask of important values
A <- [addr] x Don't care
B <- [addr]
[addr] <- A
[addr] <- B

[redacted]

A <- Imm
B <- Imm

[redacted]

Status <- W <= 0
Illegal
Status <- W < 0
Status <- W != 0
Illegal
Status <- W > 0
Status <- W >= 0
Status <- W = 0

[redacted]

W <- A
W <- B
A <- W
B <- W
IO port <- A
IO port <- B
W <- A or B
W <- A and B
W <- A xor B
W <- not A
W <- not B
W <- A + B
W <- A - B
W <- A << 1
W <- B << 1
W <- A >> 1
W <- B >> 1

Not commutative. B-A is not accepted

[redacted]

if(status): IP <- IP + 1
IP <- saved_ip + 1
IP <- IP + sign_extend(addr) rel is relative to current ip.
saved_ip <- IP; IP <- IP + sign_extend(addr); so, jmp 0 will loop forever.