# RTL Instructions

| A-type instructions | C-type instructions | Load word/Store word | Branches | Jump | Jump Register | Jump and Link | Load Upper Immediate | Jump ‘n’ Link |
|---|---|---|---|---|---|---|---|---|
| I = Mem[PC]<br>PC = PC + 1 | | | | | | | | |
| A = Reg[I[11:8]]<br>B = Reg[I[7:4]]<br>ALUOut = PC + sign-extend*(I[3:0]) | | | | | | | | |
| ALUOut = A op B | ALUOut = A op sign-extend**(I[7:0]) | ALUOut = A + sign-extend*(I[3:0]) | if (A == B): PC = ALUOut | PC = PC[15:12] \|\| I[11:0] | PC = Reg[I[11:8]] | ALUOut= PC + 0 | Reg[I[11:8]] = extend(I[7:0]) | PC = A |
| Reg[I[3:0]] = ALUOut | Reg[I[11:8]] = ALUOut | load:<br>MDR = Mem[ALUOut]<br>store:<br>Mem[ALUOut] = Reg[I[7:4]] | | | | PC = PC[15:12] \|\| (I[11:0])<br>Reg[0xb] = ALUOut | | |
| | | load: Reg[I[7:4]] = MDR<br>store: no op | | | | | | |

sign-extend* will sign-extend a 4-bit integer to a 16-bit integer.

sign-extend** will sign-extend an 8-bit integer to a 16-bit integer.

sign-extend*** will sign-extend a 12-bit integer to a 16-bit integer .

extend will create a 16-bit value where the most significant 8-bits are the given immediate and the least significant are 0s.

| mv | clear | read | display |
|---|---|---|---|
| I = Mem[PC]<br>PC = PC + 1<br>A = Reg[I[11:8]]<br>B = Reg[I[7:4]]<br>ALUOut = PC + sign-extend*(I[3:0]) | | | |
| Reg[IR[11:8]]=<br>InterruptRegister | PC = EPC<br>InterruptRegister[SELECT] = 0 | Mem[decodeOut] = lcdData | Mem[decodeOut] |
| | | blank | lcdDisplay = Mem[decodeOut] |
| | | | blank |

| interrupt = 1 |
|---|
| EPC = PC<br>PC = Int_handler_master_address |