| Bits index | What bits represent |
|---|---|
| 15 | Instruction words count |
| 14 | ALU out write back en |
| 13 | Data memory enable |
| 12 | Data out write back en |
| 11 | Rd / Wr |
| 10 : 6 | ALU operation |
| 5 : 3 | Src |
| 2 : 0 | Dst |

## Special Cases:

1- Mul : write back high ALU out to Src
2- LDD: bits 13 and 11 : 3 contain effective address so control signals are implicit
3- STD: bits 13, 11 affective address so control signals are implicit
4- Bits 2 : 0 = 7 means out.port
5- Bits 5 : 3 = 7 means in.port
6- ALU operation may specified by the means of the rest of instruction
7- Bits 5 : 3 = 6 means push PC

## Notes

1- Processor doesn’t support nested interrupts
2- Processor is in complete freeze while pipeline is stalled (doesn’t respond to interrupt signal until pipeline returns to ordinary state)
3- Load case doesn’t cause data hazards because data memory writes data in rising edge (data is passes at once to execution stage, decode stage and to fetch stage(return case fetch needs loaded PC))
4- Stalling happens due to control hazards and in return case
5- Interrupt pushed an instruction in the pipeline and this instruction is executed normally
6- R6 is SP
7- PC is not included in register file fetch stage only has control on it

Opcodes:

| instr | 15 | 14 | 13 | 12 | 11 | 10 : 6 | 5 : 3 | 2 : 0 |
|---|---|---|---|---|---|---|---|---|
| nop | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| mov | 0 | 1 | 0 | 0 | 0 | F = A | Rsrc | Rdst |
| add | 0 | 1 | 0 | 0 | 0 | F = A + B | Rsrc | Rdst |
| mul | 0 | 1 | 0 | 0 | 0 | F = A * B | Rsrc | Rdst |
| sub | 0 | 1 | 0 | 0 | 0 | F = A - B | Rsrc | Rdst |
| and | 0 | 1 | 0 | 0 | 0 | F = A &B | Rsrc | Rdst |
| or | 0 | 1 | 0 | 0 | 0 | F = A \| B | Rsrc | Rdst |
| rlc | 0 | 1 | 0 | 0 | 0 | F = rlcA | Rsrc | Rdst |
| rrc | 0 | 1 | 0 | 0 | 0 | F = rrcA | Rsrc | Rdst |
| shl | 0 | 1 | 0 | 0 | 0 | F = shlA | Rsrc | Rdst |
| shr | 0 | 1 | 0 | 0 | 0 | F = shrA | Rsrc | Rdst |
| setc | 0 | 0 | 0 | 0 | 0 | carry | 0 | 0 |
| clrc | 0 | 0 | 0 | 0 | 0 | carry | 0 | 1 |
| push | 0 | 1 | 1 | 0 | 1 | Dec A | 6 | Rdst |
| pop | 0 | 1 | 1 | 1 | 0 | Inc A | 6 | Rdst |
| out | 0 | 0 | 0 | 0 | 0 | F = A | Rdst | 7 |
| in | 0 | 1 | 0 | 0 | 0 | F = A | 7 | Rdst |
| not | 0 | 1 | 0 | 0 | 0 | F = ~A | Rsrc | Rdst |
| inc | 0 | 1 | 0 | 0 | 0 | Inc A | Rsrc | Rdst |
| dec | 0 | 1 | 0 | 0 | 0 | Dec A | Rsrc | Rdst |
| jz | 0 | 0 | 0 | 0 | 0 | jmp | 0 | Rdst |
| jn | 0 | 0 | 0 | 0 | 0 | jmp | 1 | Rdst |
| jc | 0 | 0 | 0 | 0 | 0 | jmp | 2 | Rdst |
| jmp | 0 | 0 | 0 | 0 | 0 | jmp | 3 | Rdst |
| call | 0 | 1 | 1 | 0 | 1 | call | 6 | 6 |
| ret | 0 | 1 | 1 | 0 | 0 | Inc A | 6 | 0 |
| rti | 0 | 1 | 1 | 0 | 0 | Inc A | 6 | 1 |
| ldm | 1 | 1 | 0 | 0 | 0 | F = A | 0 | Rdst |
| ldd | 1 | 0 | EA | 1 | EA | EA | EA | Rdst |
| std | 1 | 0 | EA | 0 | EA | EA | Rsrc | EA |
| interrupt | 0 | 1 | 1 | 0 | 1 | Dec A | 6 | 6 |