| Instruction | Action | Opcode | Clk | Rising Edge | Falling Edge | Control Signals | Select Signals |
|---|---|---|---|---|---|---|---|
| Fetch | fetch instruction from memory | | 1 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 2 | read opcode from memory to bus | load opcode to IR; decode to MS | RD,Lir,Lms | |
| nop | does literally nothing | 00 | 3 | | | End | |
| adi xx | add immediate | 01 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | put output of ALU in AR | Eor,Ear,Lar,End | Salu->ADD |
| sbi xx | subtract immediate | 02 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | put output of ALU in AR | Eor,Ear,Lar,End | Salu->SUB |
| xri xx | xor immediate | 03 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | put output of ALU in AR | Eor,Ear,Lar,End | Salu->XOR |
| ani xx | bitwise add immediate | 04 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | put output of ALU in AR | Eor,Ear,Lar,End | Salu->AND |
| ori xx | bitwise or immediate | 05 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | put output of ALU in AR | Eor,Ear,Lar,End | Salu->OR |
| cmi xx | compare immediate | 06 | 3 | put address to read from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value from bus to OR | RD,Lor | |
| | | | 5 | put previous AR and OR values in ALU | | Eor,Ear,End | Salu->CMP |
| stop | shuts down the processor | 07 | 3 | | | End,StopClock | |
| ret<FL> | returns to address on top of stack if flag is true | 08-0F | 3 | if <FL> is 0, skip 4 and 5 | | Efl,End if <FL>' | Sfl-><FL> |
| | | | 4 | put address of top of stack on the bus I.e. SP | load bus to MR; increment SP | Esp,Lmr,Isp | |
| | | | 5 | read address from memory to bus | load address into PC | RD,Lpc,End | |
| add <R> | add AR to <R> | 10-1F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | load value from ALU to AR | Ear,Lar,End | Salu->ADD |
| sub <R> | subtract <R> from AR | 20-2F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | load value from ALU to AR | Eor,Ear,Lar,End | Salu->ADD |
| xor <R> | xor AR with <R> | 30-3F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | load value from ALU to AR | Eor,Ear,Lar,End | Salu->ADD |
| and <R> | and AR with <R> | 40-4F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | load value from ALU to AR | Eor,Ear,Lar,End | Salu->ADD |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| or <R> | or AR with <R> | 50-5F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | load value from ALU to AR | Eor,Ear,Lar,End | Salu->ADD |
| cmp <R> | compare AR with <R> | 60-6F | 3 | put value from <R> onto bus | load value from bus to OR | Erg,Lor | Srg-><R> |
| | | | 4 | put value from AR to ALU | | Eor,Ear,End | Salu->ADD |
| movs <R> | moves value from <R> to AR | 70-7F | 3 | put values from <R> onto bus | load value from ALU to AR(<R> value is passed) | Erg,Lar,End | Srg-><R>, Salu->PASS0 |
| movd <R> | moves value from AR to <R> | 80-8F | 3 | put value from AR onto bus | load value from bus to <R> | Ear,Lrg,End | Srg-><R> |
| movi <R> xx | move value xx to <R> | 90-9F | 3 | put address to red from into bus from PC | put address into MR; increment PC | Epc,Lmr,Ipc | |
| | | | 4 | read value from memory to bus | load value into <R> | RD,Lrg,End | Srg-><R> |
| stor <R> | writes value from <R> to memory in location given by AR | A0-AF | 3 | put address from AR onto bus | load address into MR | Ear,Lmr | |
| | | | 4 | put value from <R> onto bus | write value from bus onto memory | Erg,WR,End | Srg-><R> |
| load <R> | reads value from memory to <R> in location given by AR | B0-BF | 3 | put address from AR onto bus | load address into MR | Ear,Lmr | |
| | | | 4 | read value from memory to bus | load value into <R> | RD,Lrg,End | Srg-><R> |
| push <R> | pushes value from <R> to top of stack | C0-CF | 3 | | decrements SP to location of top of stack | Dsp | |
| | | | 4 | put address of top of stack on the bus i.e. SP | load address into MR | Esp,Lmr | |
| | | | 5 | put value from <R> onto bus | write value from bus onto memory | Erg,WR,End | Srg-><R> |
| pop <R> | pops value from top of stack to <R> | D0-DF | 3 | put address of top of stack on the bus i.e. SP | loads address into MR; increment SP | Esp,Lmr,Isp | |
| | | | 4 | read value from top of stack to bus | load value from bus onto <R> | RD,Lrg,End | Srg-><R> |
| jumpd<FL> xx | if flag value is 1, program jumps to given address xx | E0-E7 | 3 | if <FL> is 0, skip 4; put address from PC to bus | load address into MR; increment PC | Epc,Lmr,Ipc, Efl,End if <FL>' | Sfl-><FL> |
| | | | 4 | put address from memory onto bus | load address onto PC | RD,Lpc,End | |
| jumpr<FL> | if flag value is 1, program jumps to address in AR | E8-EF | 3 | if <FL> is 0, skip 4 | | Efl,End if <FL>' | Sfl-><FL> |
| | | | 4 | put address from AR onto bus | load address onto PC | Ear,Lpc,End | |
| cd<FL> xx | if flag value is 1, program stores current address in stack and jumps to given address xx | F0-F7 | 3 | if <FL> is 0, skip 4,5,6,7; put address from PC to bus | load address into MR; increment PC | Epc,Lmr,Ipc, Efl,End if <FL>' | Sfl-><FL> |
| | | | 4 | put address from memory onto bus | load address into OR; decrement SP to location on top of stack | RD,Lor,Dsp | |
| | | | 5 | put address of top of stack on the bus i.e. SP | load address into MR | Esp,Lmr | |
| | | | 6 | put address from PC to the bus | write address from PC to the top of stack | Epc,WR | |
| | | | 7 | put address from OR(taken from memory) onto bus | load address onto PC | Eor,Lpc,End | |
| cr<FL> | if flag value is 1, program stores current address in stack and jumps to address in AR | F8-FF | 3 | if <FL> is 0, skip 4,5,6,7 | | Efl,End if <FL>' | Sfl-><FL> |
| | | | 4 | | decrements SP to location of top of stack | Dsp | |
| | | | 5 | put address of top of stack on the bus i.e. SP | load address into MR | Esp,Lmr | |
| | | | 6 | put address from PC to the bus | write address from PC to the top of stack | Epc,WR | |
| | | | 7 | put address from AR onto bus | load address onto PC | Ear,Lpc,End | |