

**NOTE: ALL HAVE AN INPUT FOR CLOCK**

IF/ID:

* Input:
  + Instruction (32 bits)
* Output:

1. Op code: first 6 bits (31-26)
2. Read Register 1 (rs): next 5 bits (25-21)
3. Read Register 2 (rt): next 5 bits (20-16)
4. Immediate Value: last 16 bits (15-0)
5. Write Register for “I” instruction: same as register 2 (20-16)
6. Write Register for “R” Instruction (rd): next 5 bits after register 2 (15-11)

ID/EX:

* Input:

1. ALUop
2. regDst
3. memRead
4. memToRegister
5. memWrite
6. ALUsrc
7. regWrite
8. Read data 1
9. Read data 2
10. Sign extend
11. Write Register for “I” instruction
12. Write Register for “R” Instruction (rd)

* Output:

1. ALUop
2. regDst
3. memRead
4. memToRegister
5. memWrite
6. ALUsrc
7. regWrite
8. Read data 1
9. Read data 2
10. Sign extend
11. Write Register for “I” instruction
12. Write Register for “R” Instruction (rd)
    * 11 and 12 will be sent to a multiplexer

EX/MEM

* Input:

1. memRead
2. memToRegister
3. memWrite
4. regWrite
5. Read data 2 (write data for memory)
6. ALU result
7. Value from multiplexer between IF/ID Out 11 and 12

* Output:

1. memRead
2. memToRegister
3. memWrite
4. regWrite
5. Read data 2 (write data for memory)
6. ALU result
7. Value from multiplexer between IF/ID Out 11 and 12

MEM/WB

* Input:

1. regWrite
2. memToRegister
3. ReadData
4. ALUResult
5. Value from multiplexer between IF/ID Out 11 and 12

* Output:

1. regWrite
2. memToRegister
3. ReadData
4. ALUResult
5. Value from multiplexer between IF/ID Out 11 and 12

