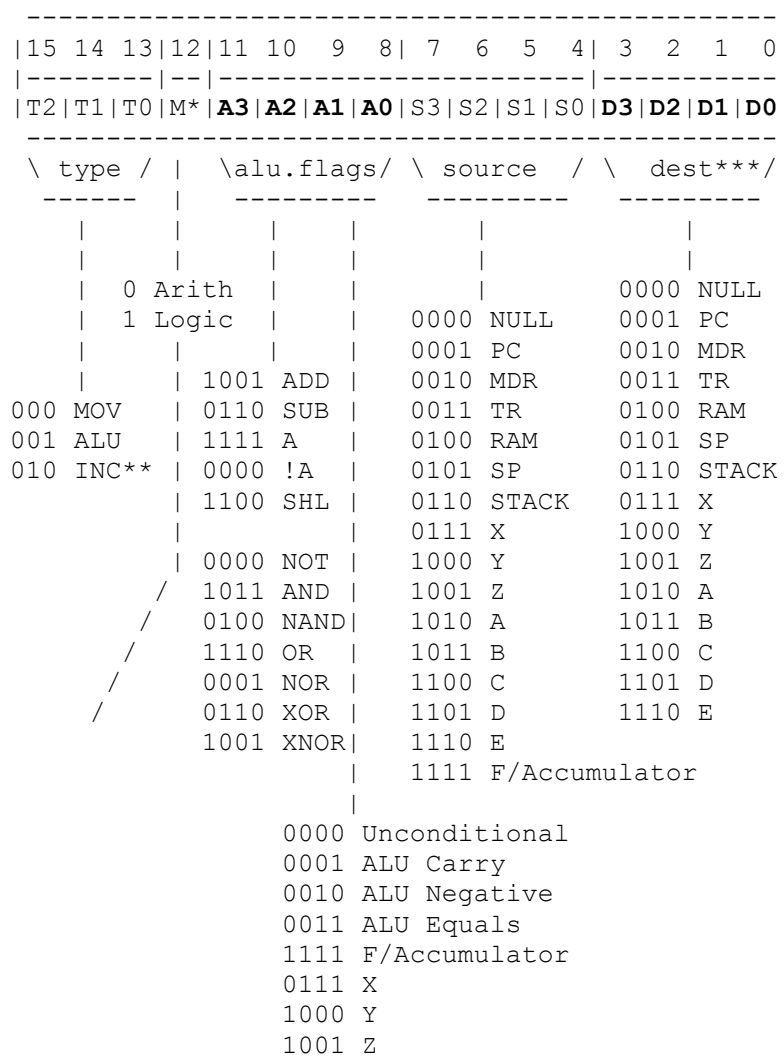
Instruction Breakdown Version 0.17



* The Mode bit is required for the ALU, but it is available as a toggle for other instructions. When the ALU executes an instruction, the destination device is actually the "right operand" input, and the output always stores to the F-Register.

** Because each PCB will have three sets of "1 of 16" enables, one instruction can increment, decrement, and reset devices. The format will be 0100 {reset device} {increment device} {decrement device}

*** The destination can be any device except the F-Register, as the inputs are tied to the results of ALU operations. If it needs to be loaded, the BUF instruction can pass a value from another device through the ALU and into the F-Register.

Control Signals:	INC, DEC, WE, OE, Reset
Register Flags:	F, X, Y, Z
ALU Operation Flags:	Carry, Negative, Equals

Addressing Modes:

Implied:	BUF
Register:	ADD A,B
Absolute/Immediate:	JMP $4032
Indirect:	JMP ($1000)

Instructions:

```
NOP         $00 00          ; null device to null device
MOV         $00 dd          ; device to device

JMP         $00 d1          ; device to program counter
BRC         $01 d1          ; device to program counter if alu carry
BRN         $02 d1          ; device to program counter if alu nergative
BRE         $03 d1          ; device to program counter if alu equal
BRF         $04 d1          ; device to program counter if accumulator (device f)
BRX         $05 d1          ; device to program counter if x
BRY         $06 d1          ; device to program counter if y
BRZ         $07 d1          ; device to program counter if z

ADD         $29 dd          ; left plus right to f
SUB         $26 dd          ; left minus right to f
BUF         $2F d0          ; left to f
NOT         $20 d0          ; not left to f
SHL         $3C d0          ; shift left left to f
AND         $3B dd          ; left and right to f
NAND        $34 dd          ; left nand right to f
OR          $3E dd          ; left or right to f
NOR         $31 dd          ; left nor right to f
XOR         $36 dd          ; left xor right to f
XNOR        $39 dd          ; left xnor right to f

INC         $4 0d0          ; increment device
DEC         $4 00d          ; decrement device
RESET       $4 d00          ; reset device
```

Examples:

```
NOP             %0000000001110001       $0000         move null to null

MOV X,PC        %0000000001110001       $0071         move X to PC
MOV X,$0000     %0000000001110100       $0074 $0000   move X to RAM[$0000]
MOV $2002,A     %0000000001001010       $0047 $2002   move $2002 to A

MVC X,PC        %0000000101110001       $0171         move X to PC if alu carry set
MVE A,X         %0000001110100111       $03A7         move A to X if alu equals set

ADD A,B         %0010100110101011       $29AB         A plus B
NOT Z           %0011000010010000       $3090         NOT Z
OR X,Y          %0011111001111000       $3E78         X OR Y

INC X           %0100000001110000       $6070         increment X
DEC X           %0100000000000111       $6007         decrement X
RESET X         %0100011100000000       $6700         reset X to a zero value
```