LC3 Instruction Diagrams
NOT (Register)

Note: Src and Dst could be the same register.

Assembly Ex: NOT R3, R2
ADD/AND (Register)

**ADD**

```
+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 1 | Dst | Src1 | 0 | 0 | 0 | Src2 |
```

**AND**

```
+---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 1 | Dst | Src1 | 0 | 0 | 0 | Src2 |
```

This zero means “register mode.”

Assembly Ex:
Add R3, R1, R3
ADD/AND (Immediate)

ADD
0 0 0 1 Dst Src1 1 Imm5

AND
0 1 0 1 Dst Src1 1 Imm5

Note: Immediate field is sign-extended.

this one means “immediate mode”

Assembly Ex:
Add R3, R3, #1
LD (PC-Relative)

Assembly Ex:
LD R1, Label1

LD 0 0 1 0  | Dst | PCoffset9
ST (PC-Relative)

Assembly Ex:
ST R1, Label2
LDI (Indirect)

LDI 1 0 1 0 Dst PCoffset9

Assembly Ex: LDI R4, Adr
STI (Indirect)

Assembly Ex:
STI  R4, Adr
LDR (Base+Offset)

Assembly Ex:
LDR  R4, R1, #1
STR (Base+Offset)

Assembly Ex:
STR R4, R1, #1
LEA (Immediate)

LEA 1110 Dst PC_offset9

Assembly Ex:
LEA R1, Lab1
What happens if bits [11:9] are all zero? All one?
JMP (Register)

Jump is an unconditional branch -- *always* taken.

- Target address is the contents of a register.
- Allows any target address.
NOTE: PC has already been incremented during instruction fetch stage.
NOTE: PC has already been incremented during instruction fetch stage.
NOTE: PC has already been incremented during instruction fetch stage.