Program & data are both in memory
- Program is executed one instruction at a time directed by control unit
  - Instruction is in memory
  - Processed in a step-by-step manner (instruction cycle)
Six phases of instruction cycle

1. Fetch instruction
2. Decode it (look at opcode IR[15:12])
3. Evaluate Address (if there is memory access)
4. Fetch operands from register or memory
5. Execute requested operation
6. Store result in register or memory

LC-3 Datapath
Fetch phase:

- Instruction is a set of bits
- Resides in memory
- Needs to go to IR backwards

IR ← MDR (read operation)

Where to read from?

MDR ← M[MAR]

PC holds the address to the instruction

MAR ← PC, PC ← PC+1
Σ \( \text{MAR} \leftarrow \text{PC} \), \( \text{PC} \leftarrow \text{PC} + 1 \)

**In Summary,**

**In fetch phase**

1. \( \text{MAR} \leftarrow \text{PC} \), \( \text{PC} \leftarrow \text{PC} + 1 \)
2. \( \text{MDR} \leftarrow M[\text{MAR}] \)
3. \( \text{IR} \leftarrow \text{MDR} \)

**Imp:** When LC3 executes an instruction, PC holds the address of next instruction (current value +1)

2. **Decode:** Examine the instruction \( \text{IR}[15:12] \) to figure out what to do.
LC-3 INSTRUCTION SET - PART I

Operate instructions

* NOT: bitwise complement of Source Register (SR), result stored in Destination Register (DR).
Set condition codes (setcc)

Example:

1001 011 101 11111

Op code (NOT) Unused

Diagram of ALU with inputs 16 16 and outputs B A and NOT.
* ADD (immediate mode): Source Register (SR1) is added to sign-extending bits IR[4:0], result is stored in DR.

Setcc

Example:

```
00001 001 100 1110
```

Opcodes (ADD)
Addressing mode

Question: what happens if IR[5] = 0?

* ADD (register mode): SR1 is added to SR2, result is stored in DR.

Setcc
Example:

```
0001 001 100 0 00 110
```

(ADD)

Addressing mode

* AND (immediate mode): $SR1$ is bitwise AND to sign-extending bits $IR[4:0]$, result is stored in $DR$.

Setcc
Example:

```
0101   001  100   1   1110
```

Operator
(AND)

*AND (register mode): SR1 is bitwise AND with SR2, result is stored in DR.*
Example:

0101 001 100 0 00 110

Opcode (AND)

Remember:

(1) Instructions with a '+' next to their names end with setting condition codes
as appropriate

(2) Not all 2's complement integers can be used as immediate operands—only \([-16, 15]\)

(3) \textbf{NOT R2 R2} \underline{111111} \\
Source & destination can be same register

(4) Clear register:

(5) Copying from one register to another:
(5) Copying from one register to another.
\[ R_2 \leftarrow R_1 \text{AND} x'FFFF \quad \text{or} \quad R_2 \leftarrow R_1 + x'0000 \]

(6) Incrementing value in register:
\[ R_2 \leftarrow R_2 + 1 \quad (\text{immediate mode}) \]

0001 010 010 1 00001

---

**Snapshot of LC-3**

Address
| R0 | 0000 0000 0000 0000 |
| R1 | bits |
| R2 | bits |
| R3 | 0001 1000 0000 0001 |
| R4 | bits |
| R5 | bits |
| R6 | bits |
| R7 | bits |

REG FILE

MAR \( x \ 30F6 \)
PC \( x \ 30F7 \)
MDR \( x \ 54AO \)
IR \( x \ 54AO \)

R3 \( x \ 1801 \)

0101 010 010 1 0001 011 011 1 0010 010 000 0000 0010 010 010 0000 0001 010 010 010 0000 0110 000 000 0000
R2 ← R2 AND 0
R3 ← R3 + 2
R2 ← M[PC + 1]
1. Which instruction is being executed?
2. What are reg. values after (2) is executed?

Fetch:
- \( \text{MAR} \leftarrow \text{PC} \), \( \text{PC} \leftarrow \text{PC} + 1 \)
- \( \text{MDR} \leftarrow M[\text{MAR}] \)
- \( \text{IR} \leftarrow \text{MDR} \)

\( x_{30F7} \)  \( x_{30F8} \)  \( x_{16E2} \)  \( x_{16E2} \)  \( x_{1803} \)