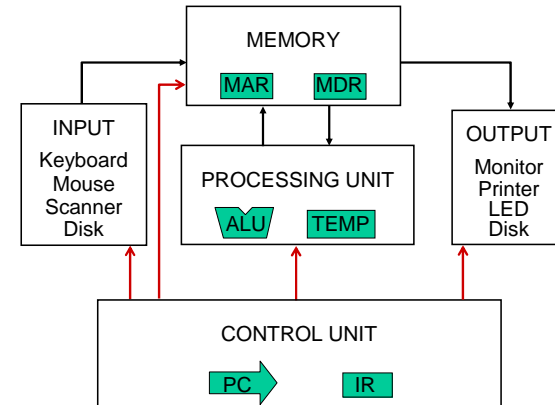


## Processor Data Path & LC3 Overview

CIT 595  
Spring 2008

## Von Neumann Model



CIT 595

2

## Instruction Set Architecture

- ISA = *Programmer-visible* components & operations
  - Memory organization
    - Address space -- how many locations can be addressed?
    - Addressability -- how many bits per location?
  - Register set
    - How many? What size?
  - Instruction set
    - Opcodes
    - Data types
    - Addressing modes
- All information needed to write/generate *machine language* program

CIT 595

3

## Instruction

- Fundamental unit of work
- Constituents
  - *Opcode*: operation to be performed (e.g. ADD, LD)
  - *Operands*: data/locations to be used for operation
    - Source: location that contains the data/instruction
    - Destination: location that will store the result of computation
    - Immediate: data values not contained at a particular location

CIT 595

4

## LC-3 Overview: Memory and Registers

- **Memory**
  - Address space:  $2^{16}$  locations (16-bit addresses)
  - Addressability: **16 bits**
- **Registers**
  - Temporary storage (Memory access takes longer)
  - Eight general-purpose registers: **R0 - R7** (each **16 bits wide**)
  - Other registers: Not directly addressable, but used by and affected by instructions
    - E.g. **PC** (program counter), **condition codes** (N/Z/P)
- **Word Size**
  - Number of bits normally processed by ALU in one instruction
  - Also width of registers
  - LC-3 word size is 16 bits

CIT 595

5

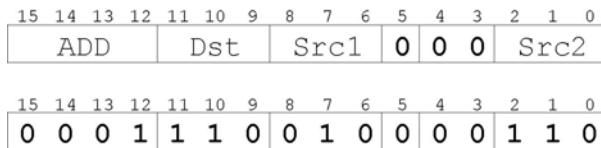
## LC-3 ISA: Overview

- **Opcodes**
  - 16 opcodes ([15:12] of instruction =  $2^4 = 16$  possible values)
  - Types of instructions:
    - **Operate** instructions: ADD, AND, NOT
    - **Data movement** instructions: LD, LDI, LDR, LEA, ST, STR, STI
    - **Control** instructions: BR, JUMP, TRAP JSR, JSRR, RET, RTI
  - Operate and Data movement instructions (except Store) set/clear *condition codes*, based on result
    - N = negative (<0), Z = zero (=0), P = positive (> 0)
- **Addressing Modes**
  - How is the location of an operand (data to acted upon) specified?
  - Non-memory addresses: *register, immediate (literal)*
  - Memory addresses: *base+offset, PC-relative, indirect*
- **Data Types**
  - 16-bit 2's complement integer

CIT 595

6

## Example: ADD Instruction Format

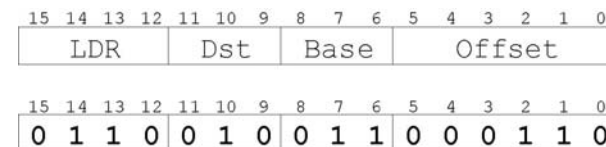


**LC-3 ADD** : Add the contents of R2 to the contents of R6, and store the result in R6.

CIT 595

7

## Example: LDR Instruction Format



*Add the value 6 to the contents of R3 to form a memory address. Load the contents of memory at that address and place the resulting data in R2*

CIT 595

8



## Data Path Components (cont..)

### Control Unit

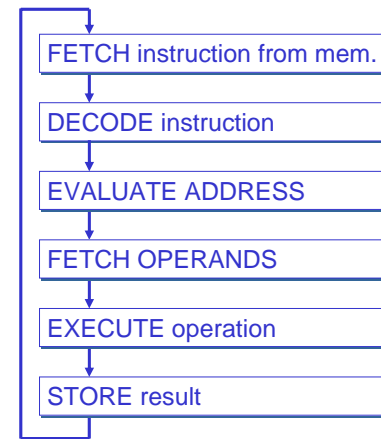
- Using the opcode information, it decides for each stage in instruction
  - > Who drives the bus?
  - > Which registers are write enabled?
  - > Which operation should ALU perform?

- Recap Instruction Processing Cycle

CIT 595

13

## Instruction Processing Cycle



CIT 595

14

## Instruction Processing Cycle (continued)

### Fetch

- Get instruction from PC and supply it to decode stage
  - Simple datapath version, the instruction memory is read by default (CU is not controlling)
  - PC is incremented by 1 (default)

### Decode

- Identify opcode and send this information to CU
  - Bits[15:12] bits of instruction
  - Split remaining instruction information and send it to appropriate components



CIT 595

15

## Instruction Processing

### Evaluate Address

- Compute address loads, stores & control-flow instructions (BR instruction)
  - Add offset to base register (as in LDR)
  - Add offset to PC (as in LD and BR)
- CU signals appropriate mux for operands to ALU

### Fetch Operands

- Get source operands for operation
- Examples
  - Read data from register file (ADD)
  - Load data from memory (LDR)



CIT 595

16

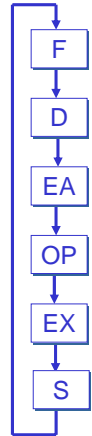
## Instruction Processing: EXECUTE

### Execute

- Actually perform operation
- Do nothing for e.g. loads and stores
- Control Unit
  - sends operands to ALU
  - Asserts appropriate function (AND, NOT, OR)

### Store

- Write results to destination (register or memory)
- Examples
  - Result of ADD is placed in destination reg.
  - Result of load instruction placed in destination reg.
- Control unit
  - Sends the result of
  - asserts WRITE signal to register or memory



CIT 595

17

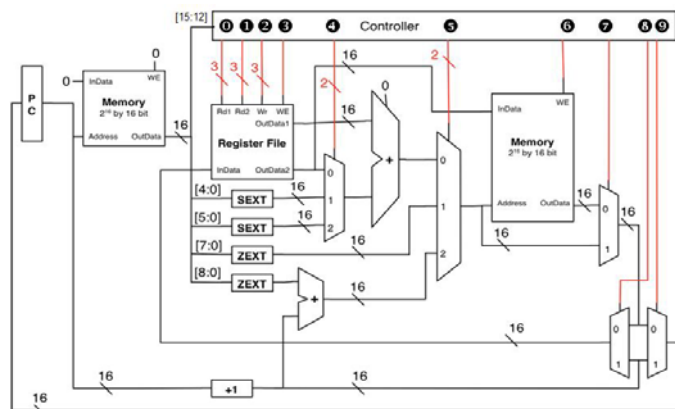
## Next

- How does the control unit work ?
  - **Two approaches:**
    - Hardwired Control
    - Microprogrammed Control (FSM)

CIT 595

18

## Simple LC-3 Datapath



CIT 595

19