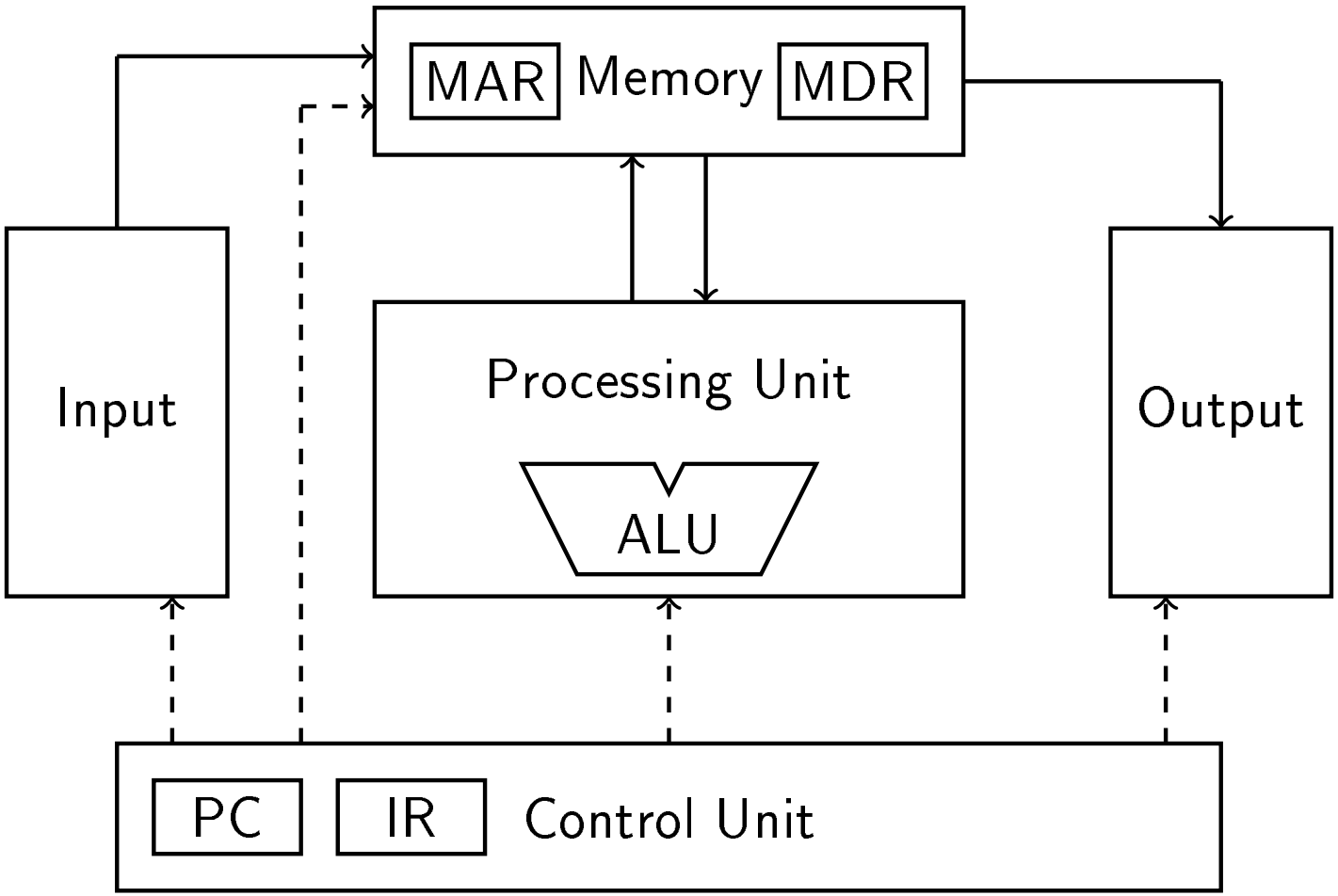
The Von Neumann Model

The Stored Program Computer

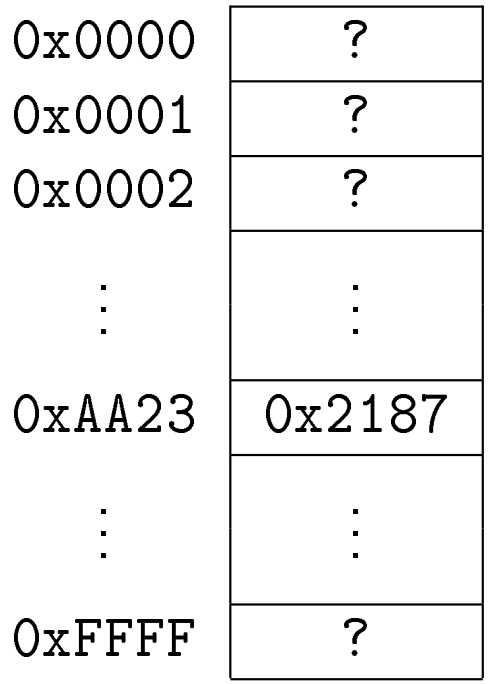
- **1822** — Charles Babbage's Difference Engine, a mechanical computer, not built until 1991.
- **1936** — Alan Turing's Universal Turing Machine, the model of a general purpose computer, reads programs stored on tapes.
- **1943-'46** — ENIAC, among the earliest general purpose electronic computers, is programmed manually.
- **1944-'49** — EDVAC uses binary, reads input from magnetic tape, and stores programs internally in mercury delay lines.
- **1945** — In *First Draft of a Report on the EDVAC*, John von Neumann describes the architecture of EDVAC.

The Von Neumann Model



Memory

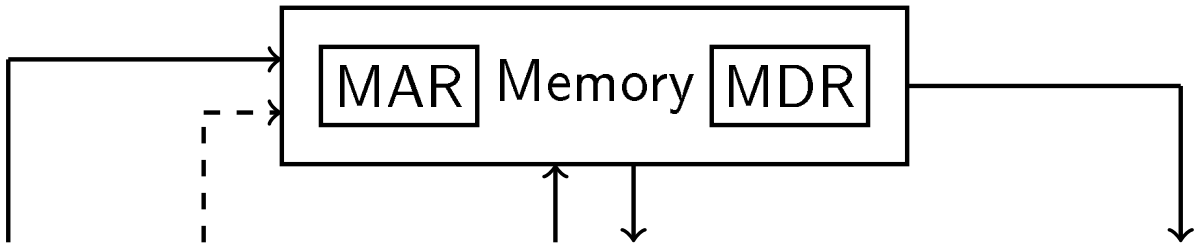
- Memory is a $2^k \times m$ array of stored bits.
- Every location in memory has a unique k-bit **address**.
- Every location contains an m-bit value.



Example

- The LC-3's memory (diagrammed above) is $2^{16} \times 16$.
- The memory location at address `0xAA23` has contents `0x2187`.

Memory



- Memory has two basic operations:
  - **load** (read)
  - **store** (write)
- Memory has two special registers: the Memory Address Register (**MAR**) and the Memory Data Register (**MDR**).

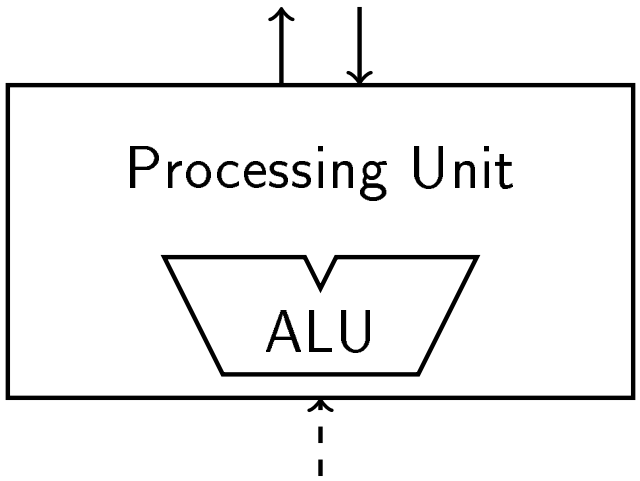
To load a value from memory:

1. Write address to MAR.
2. Send a "read" signal.
3. Read value from MDR.

To store a value in memory:

1. Write value to MDR.
2. Write address to MAR.
3. Send a "write" signal.

Processing Unit

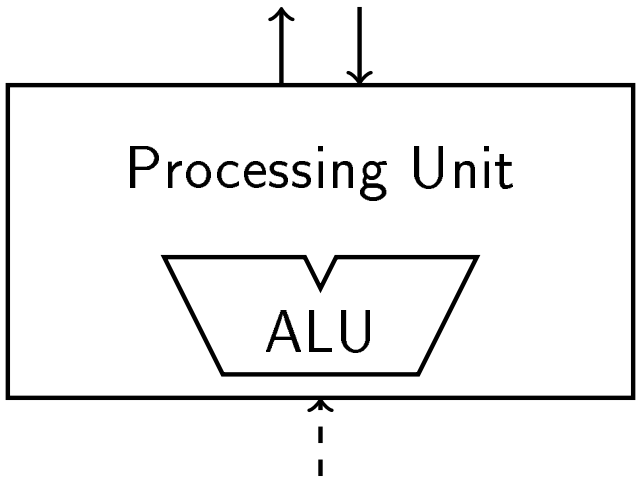


- Contains functional units for performing computations, including the Arithmetic and Logic Unit.
- Contains **registers**, small, fast temporary storage.

Definition

The **word size** is the number of bits handled at once by the Processing Unit, and is typically the size of registers.

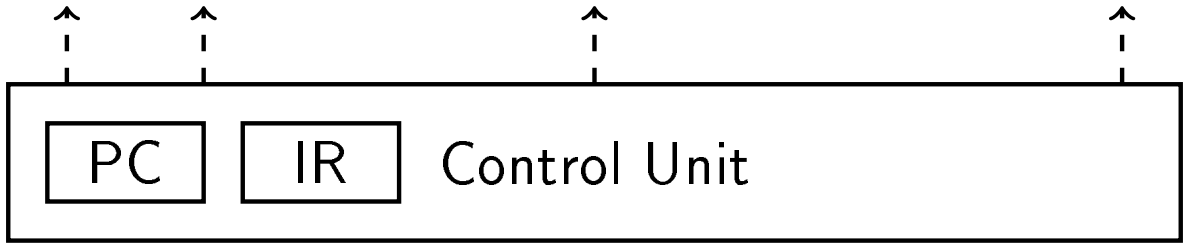
Processing Unit



Example

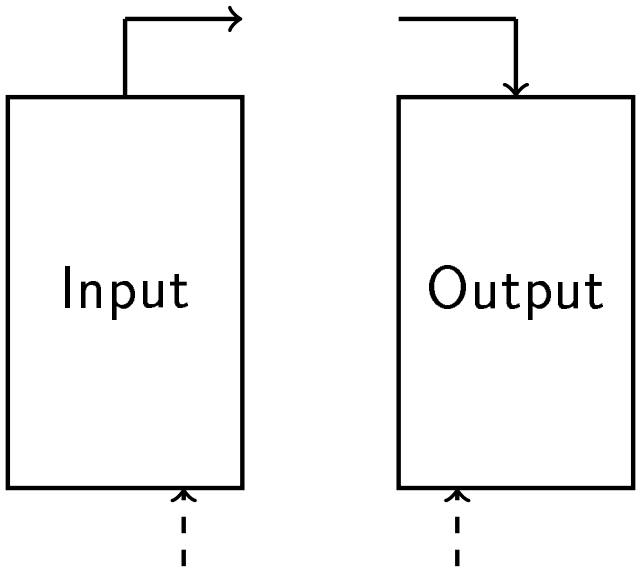
- LC-3's ALU performs three operations: `ADD`, `AND`, and `NOT`.
- It has eight registers, `R0` through `R7`.
- It has a word size of 16 bits.

Control Unit



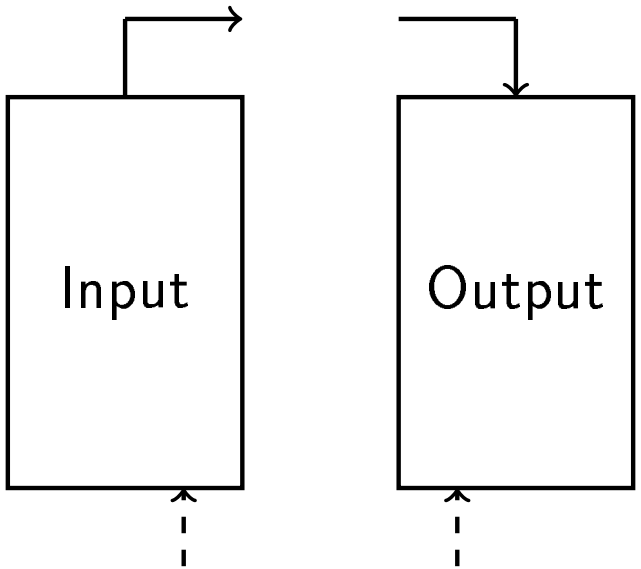
- The Control Unit interprets instructions and sends signals to other components to coordinate the execution of an instruction.
- The Instruction Register (**IR**) contains the current instruction being executed.
- The Program Counter (**PC**) contains the *address* of the *next* instruction to be executed. Where is the next instruction?

Input and Output



- I/O are devices for transferring data into and out of memory.
  - Disk, network, keyboard, mouse, monitor, printer, ...
- Each device has its own hardware interface, typically a set of specialized registers.
- A program controlling access to I/O devices is called a:

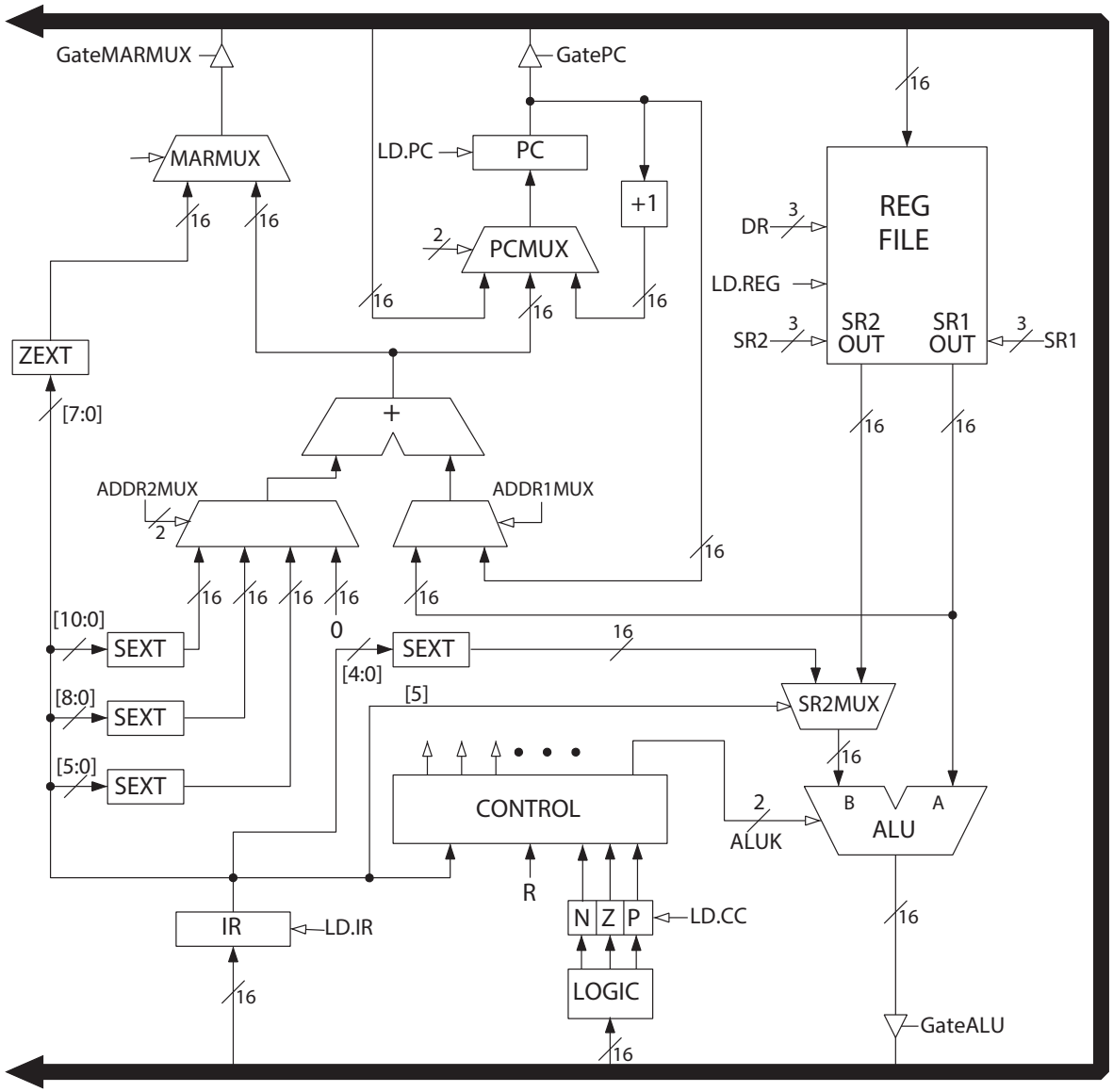
Input and Output



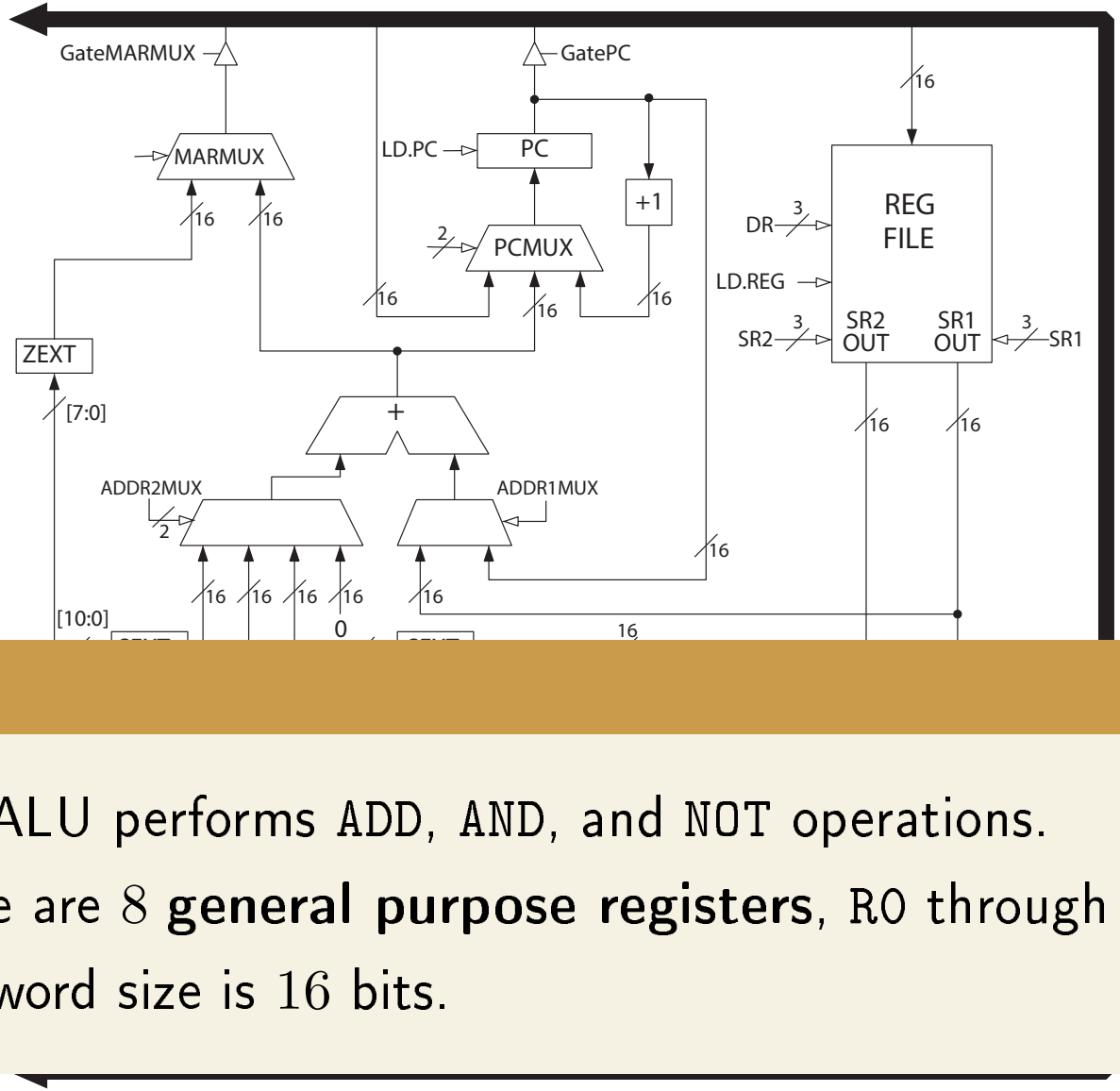
Example

- LC-3 supports keyboard input and terminal output.
- LC-3 I/O is controlled using four specialized registers: KBSR, KBDR, DSR, and DDR.

The LC-3


GateMARMUX
GatePC
16
MARMUX
LD.PC
PC
+1
16
16
2
PCMUX
16
16
16
DR
3
REG
FILE
LD.REG
SR2
3
SR2
OUT
SR1
OUT
3
SR1
ZEXT
[7:0]
+
16
16
ADDR2MUX
2
ADDR1MUX
16
16
16
16
16
16
[10:0]
SEXT
0
SEXT
16
[4:0]
[5]
SR2MUX
[8:0]
SEXT
16
[5:0]
SEXT
CONTROL
2
B
A
ALU
ALUK
R
IR
LD.IR
N
Z
P
LD.CC
16
LOGIC
16
GateALU

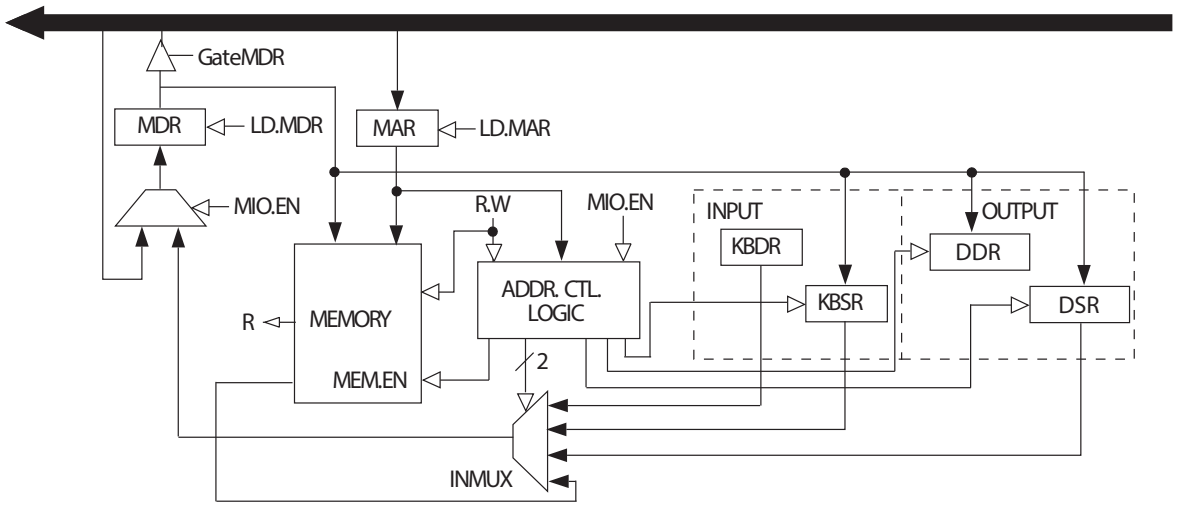
The LC-3's Processing Unit



Example

- The ALU performs `ADD`, `AND`, and `NOT` operations.
- There are 8 **general purpose registers**, `R0` through `R7`.
- The word size is 16 bits.

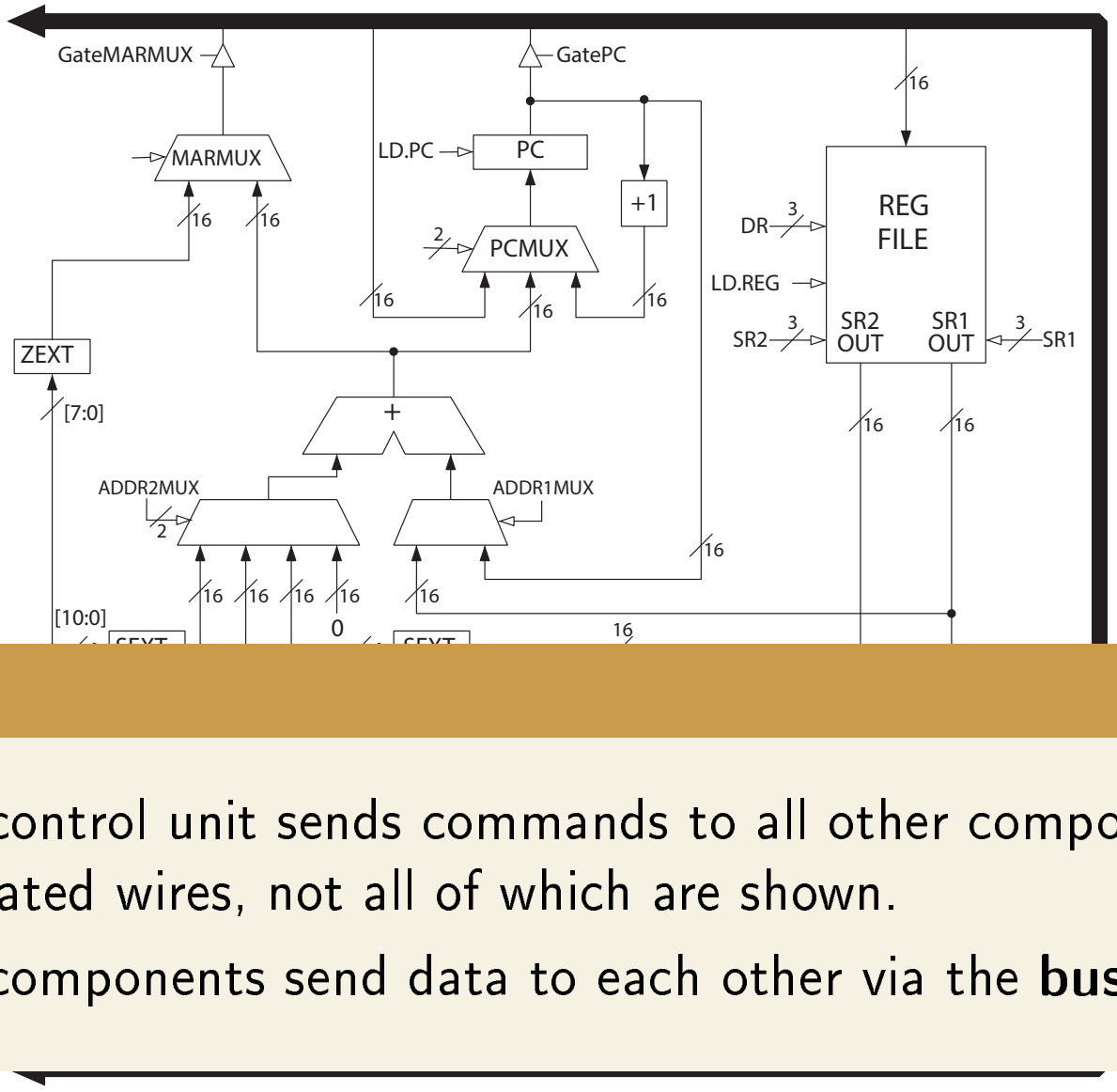
The LC-3's Memory



Example

- There are $2^{16} \times 16$ bits of memory.
  - Every location has a 16-bit address and a 16-bit value.
  - The **address space** is 2^{16}; the **addressability** is 16 bits.
- Addresses range from `0x0000` to `0xFFFF`.

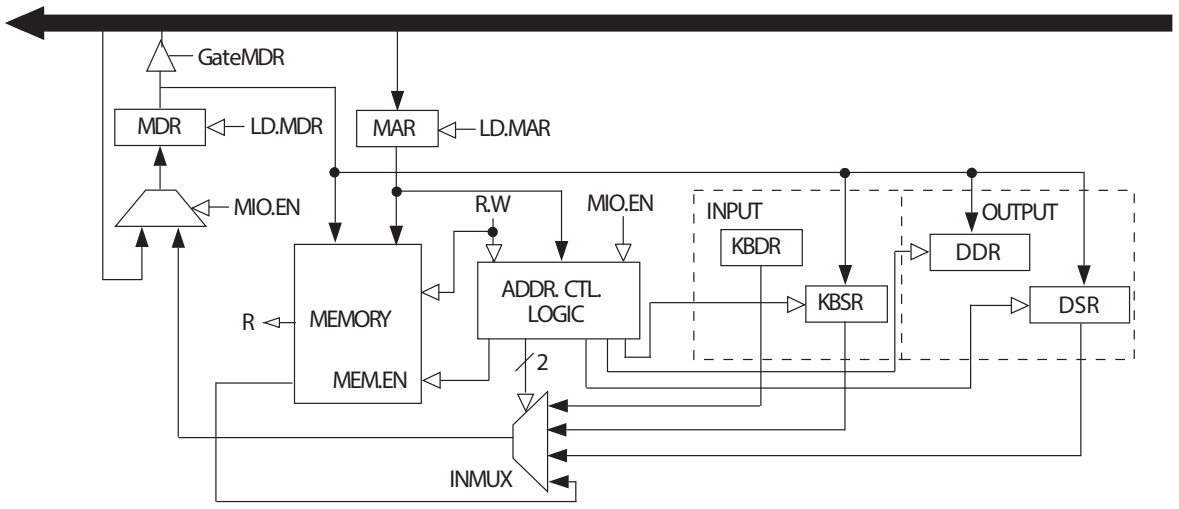
The LC-3's Control Unit



Example

- The control unit sends commands to all other components via dedicated wires, not all of which are shown.
- The components send data to each other via the **bus**.

The LC-3's Input and Output



Example

- Keyboard input is supported via the Keyboard Status Register, or **"KBSR"**, and Keyboard Data Register, or **"KBDR"**.
- Terminal output is supported via the Display Status Register, or **"DSR"**, and Display Data Register or **"DDR"**.