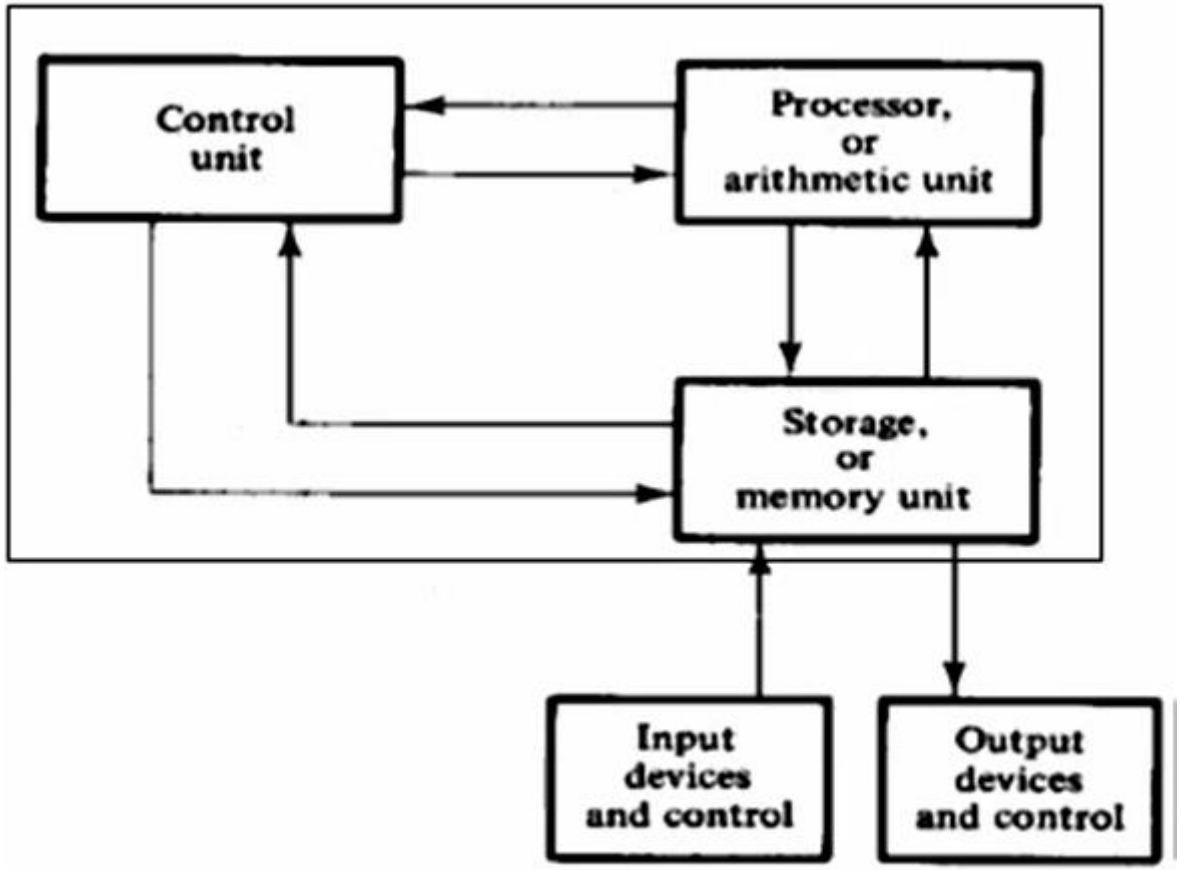
Computer System Architecture

DR. Howida Youssry

REGISTER TRANSFER LANGUAGE

- A **digital system** is an interconnection of digital hardware ***modules*** that accomplish a specific information processing task.
- The various modules are *interconnected* with **common data** and **control paths** to form a digital computer system.



General Computer organization

Hardware Organization of Digital Computer

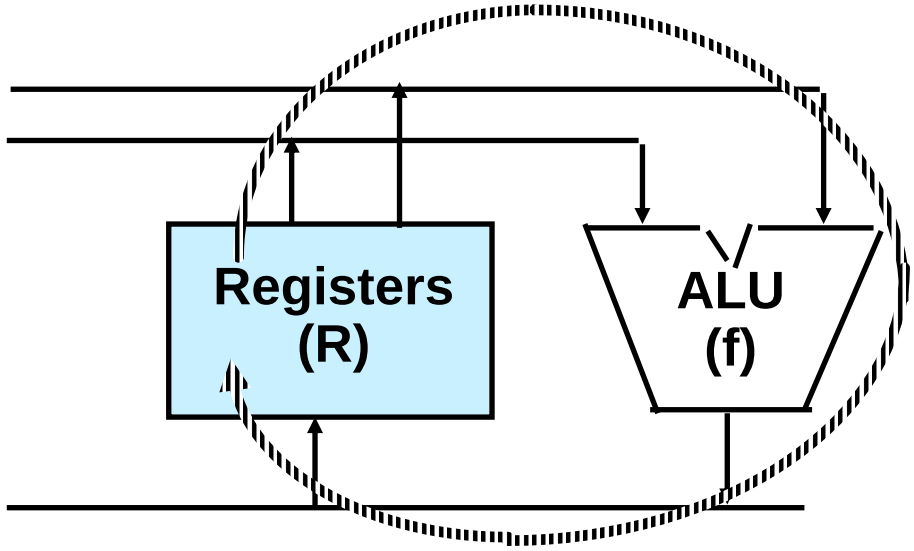
Hardware organization of a digital computer is defined by specifying:

- The **set of register** it contains and their function.
- The **sequence of micro-operations** performed on the binary information stored in the registers.
- The **control** that initiates the sequence of micro-operations

Digital Computer = Registers + Microoperations Hardware + Control Functions

MICROOPERATION

An elementary operation performed during one clock pulse, on the information stored in one or more registers



1 clock cycle

$R \leftarrow f(R, R)$

f: shift, count, clear, load, add,...

MICRO-OPERATIONS

Four types of microoperations

- **Register transfer microoperations**
- **Arithmetic microoperations**
- **Logic microoperations**
- **Shift microoperations**

REGISTER TRANSFER LANGUAGE (RTL)

- This is a **hardware description language** for digital system design.
- It allows you to specify the **data transfers and data manipulation** that can take place in a digital system.
- The **symbolic notation** used to describe the *microoperation* transfer among registers provides an organized and concise manner

Register Transfer

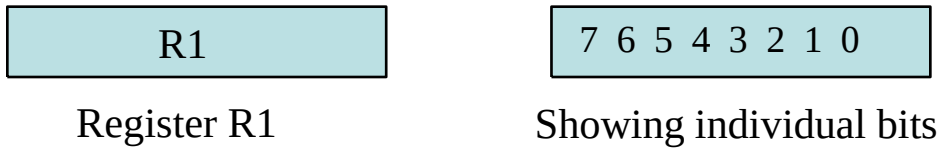
BasicSymbolsforRegisterTransfers		
Symbol	**Description**	**Examples**
Letters& numerals	Denotesaregister	MAR,R2
Parenthesis()	Denotesapartofa register	R2(0-7),R2(L)
Arrow←	Denotestransferof information	R2←R1
Comma,	Separatestwo microoperations	R2←R1,R1←R2

Register Transfer (our first microoperation)

- Computer registers are designated by capital letters (sometimes followed by numerals) to denote the function of the register
  - R1: processor register
  - MAR: Memory Address Register (holds an address for a memory unit)
  - PC: Program Counter
  - IR: Instruction Register
  - SR: Status Register

Register Transfer (our first microoperation)

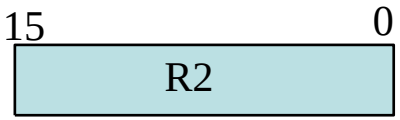
- The individual flip-flops in an n-bit register are numbered in sequence from 0 to n-1 (from the right position toward the left position)



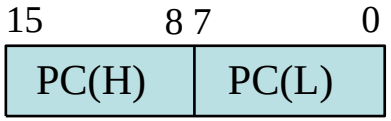
A block diagram of a register

Register Transfer (our first microoperation)

Other ways of drawing the block diagram of a register:



Numbering of bits

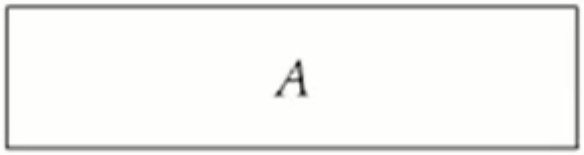


Upper byte

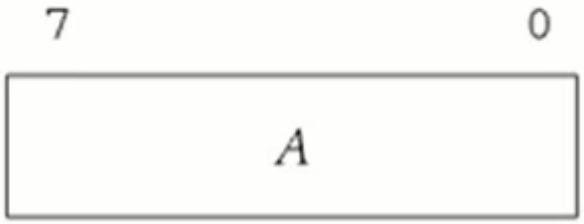
Lower byte

Partitioned into two parts

Register Notation

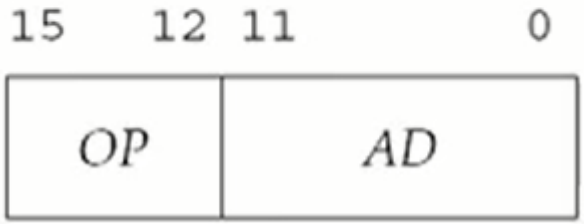


A simple register



Register showing bit positions

A *Register showing individual bits*



IR *Register showing named portions*

e.g. A(AD), A(Low), A(H),A(High)

Register Transfer (our first microoperation)

- Information transfer from one register to another is described by a replacement operator: **R2 ← R1**
- This statement denotes a transfer of the content of register R1 into register R2
- The transfer happens in one clock cycle
- The content of the R1 (source) does not change
- The content of the R2 (destination) will be lost and replaced by the new data transferred from R1

Register Transfer (our first microoperation)

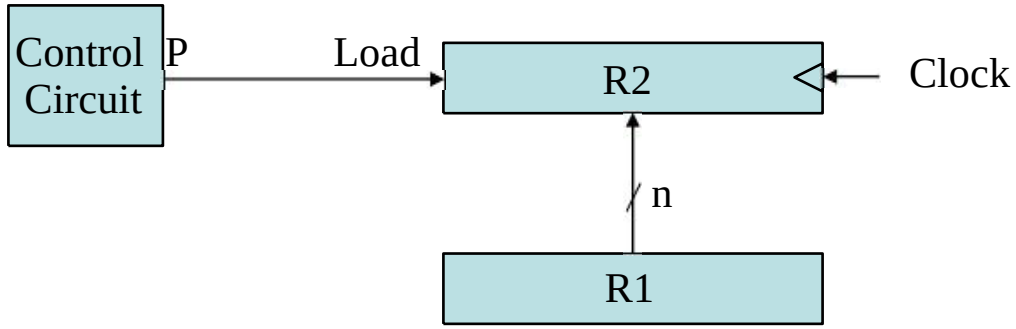
- Conditional transfer occurs only under a control condition
- Representation of a (conditional) transfer

  If (P=1) then R2 ← R1
- A binary condition (P equals to 0 or 1) determines when the transfer occurs
- The content of R1 is transferred into R2 only if P is 1

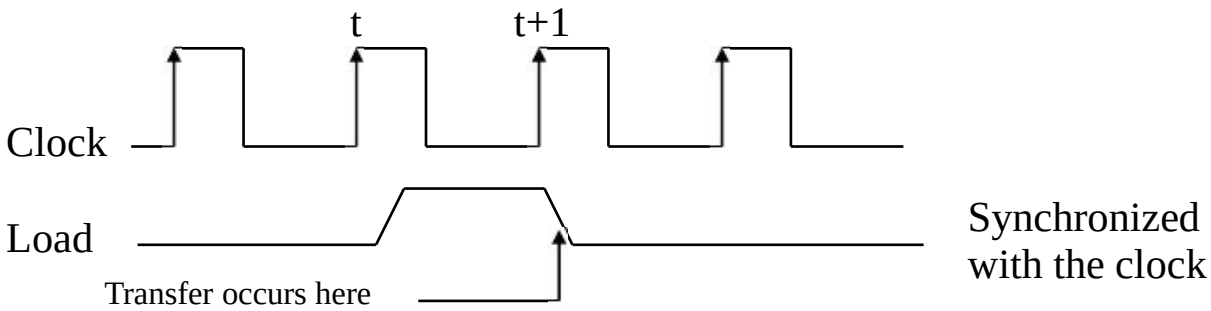
Register Transfer (our first microoperation)

Hardware implementation of a controlled transfer: P: R2 ← R1

Block diagram:



Timing diagram



Register Transfer (our first microoperation)

Basic Symbols for Register Transfers		
Symbol	*Description*	*Examples*
Letters& numerals	Denotesaregister	MAR,R2
Parenthesis()	Denotesapartofa register	R2(0-7),R2(L)
Arrow←	Denotestransferof information	R2←R1
Comma,	Separatestwo microoperations	R2←R1,R1←R2

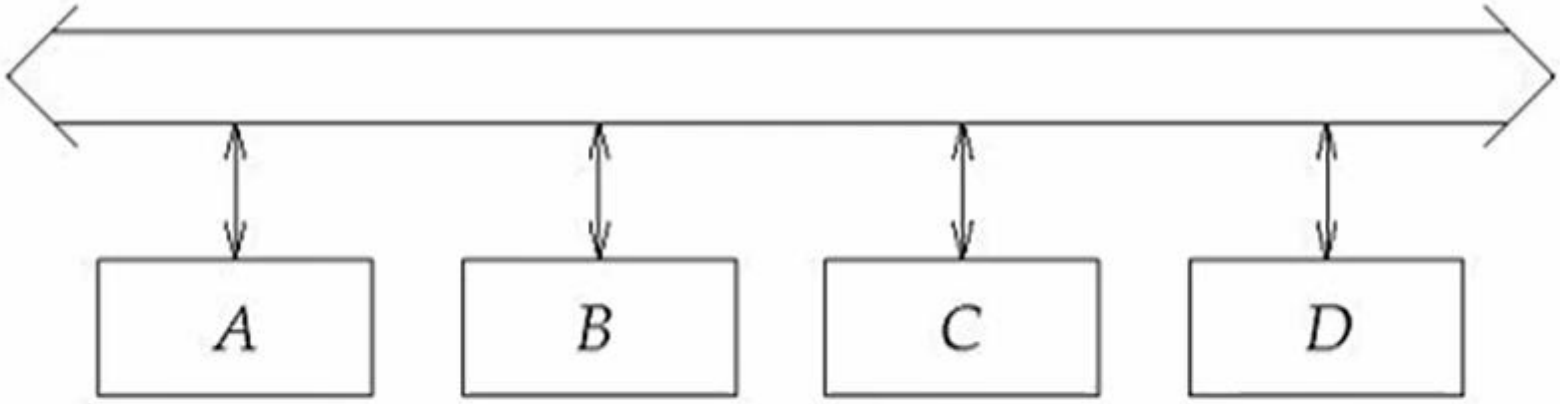
Bus and Memory Transfers

- Paths must be provided to transfer information from one register to another
- A Common Bus System is a scheme for transferring information between registers in a multiple-register configuration
- A bus: set of common lines, one for each bit of a register, through which binary information is transferred one at a time
- Control signals determine which register is selected by the bus during each particular register transfer

Solution: Common Bus Solution

Common bus: is a set of common lines ,one for each bit of a register.

Control signals determine which register is the source, and which is the destination



B← A

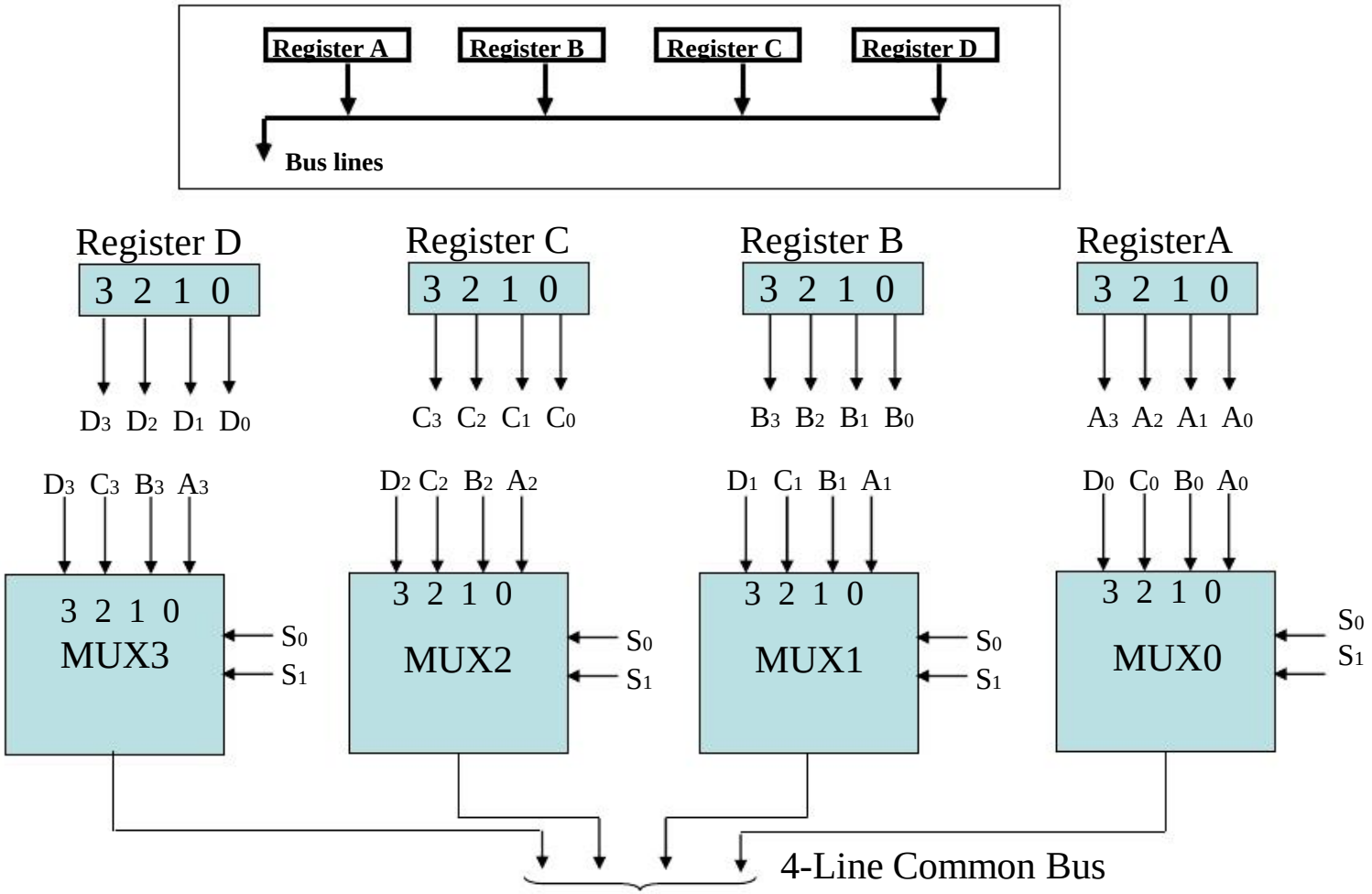
Dest ← Source

means that: Bus ← Source, then Dest ←Bus

Bus ← A

B ← Bus

Bus and Memory Transfers



Bus and Memory Transfers

- Bus selection : two selection lines S1 and S0 are connected to the selection inputs of all four multiplexers.

S_1	S_0	Registerselected
0	0	A
0	1	B
1	0	C
1	1	D

Bus and Memory Transfers

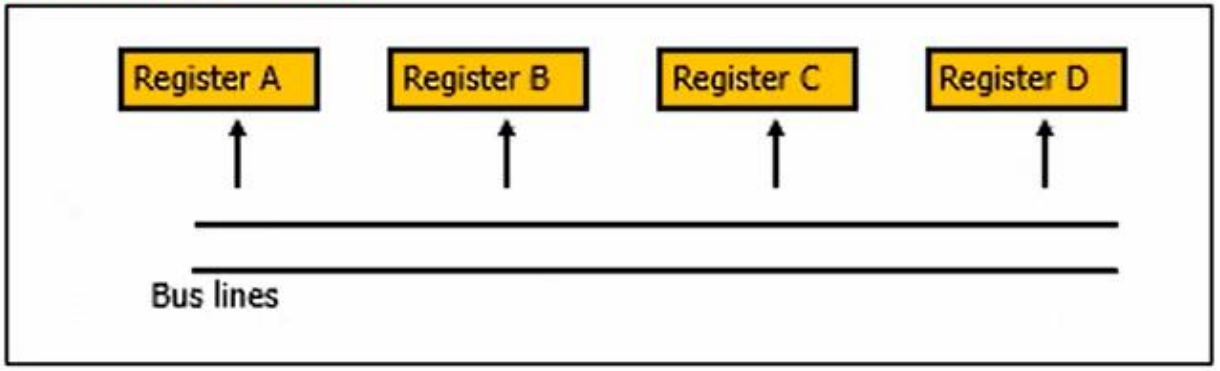
- The transfer of information from a bus into one of many destination registers is done:
  - By connecting the bus lines to the inputs of all destination registers and then:
  - activating the load control of the particular destination register selected

$$BUS \leftarrow C\ ,\ R1 \leftarrow BUS$$

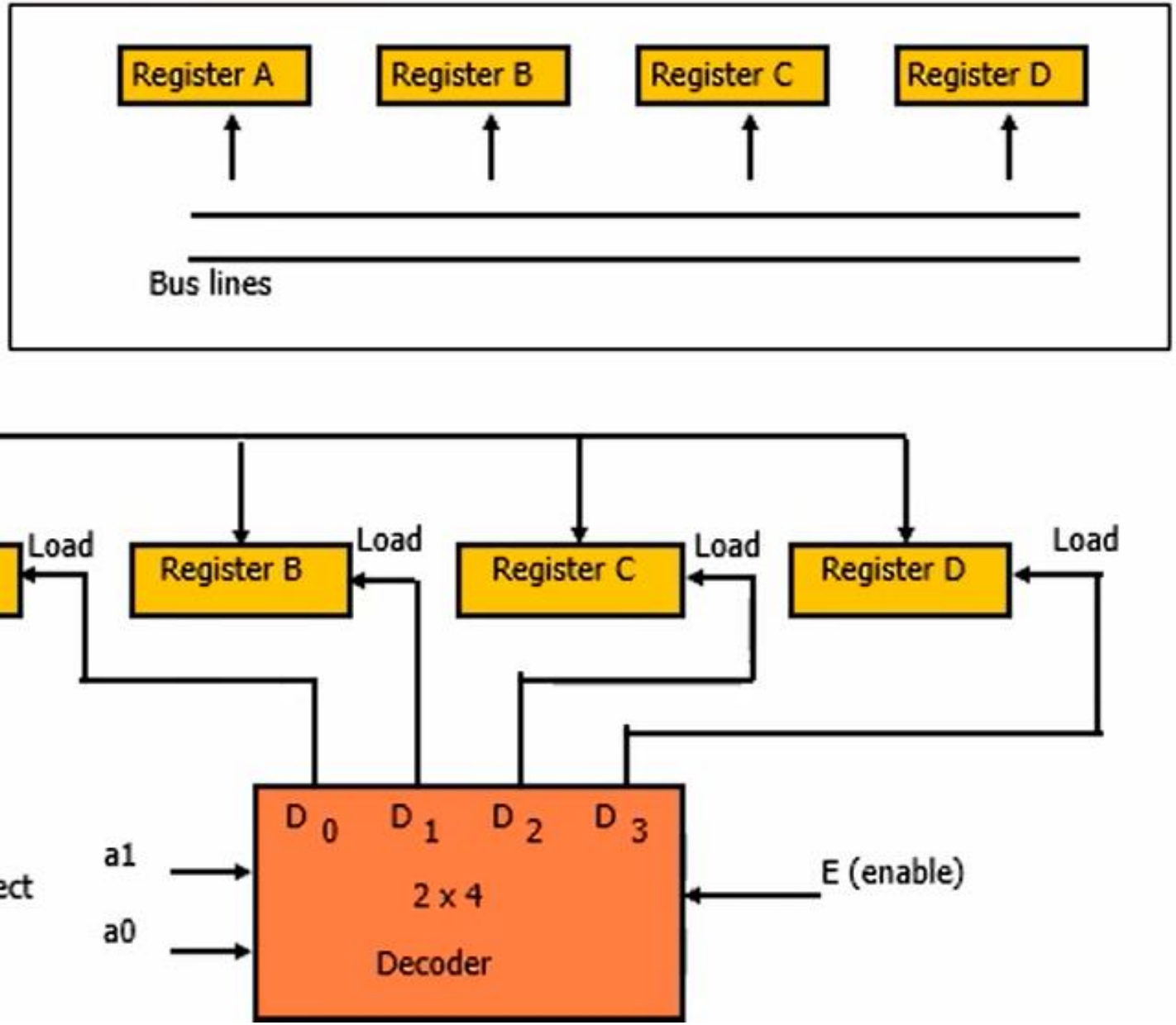
- The content of register C is placed on the bus
- content of bus is loaded into register R1
- It is equivalent to: R1 ← C

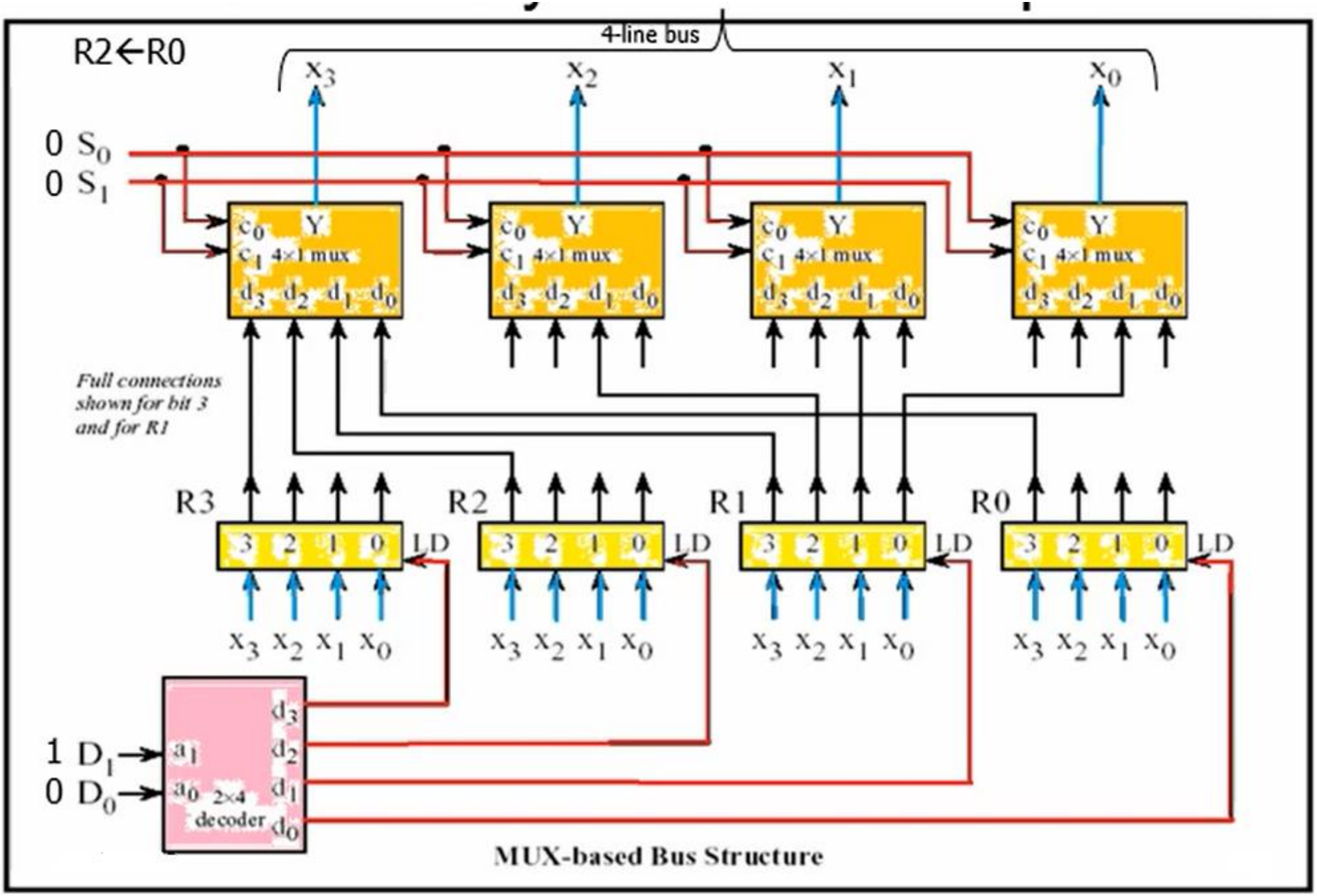
Transfer from bus to a destination register.

From bus to register : R ← bus



From bus to register : R ← bus




R2←R0
4-line bus
x_3 x_2 x_1 x_0
0 S_0
0 S_1
c_0 c_1 Y 4×1 mux d_3 d_2 d_1 d_0
Full connections shown for bit 3 and for R1
R3 R2 R1 R0
3 2 1 0 LD
x_3 x_2 x_1 x_0
1 D_1
0 D_0
a_1 a_0 2×4 decoder d_3 d_2 d_1 d_0
MUX-based Bus Structure

Memory Transfer

- Read operation: The transfer of information from a memory word to outside environment.

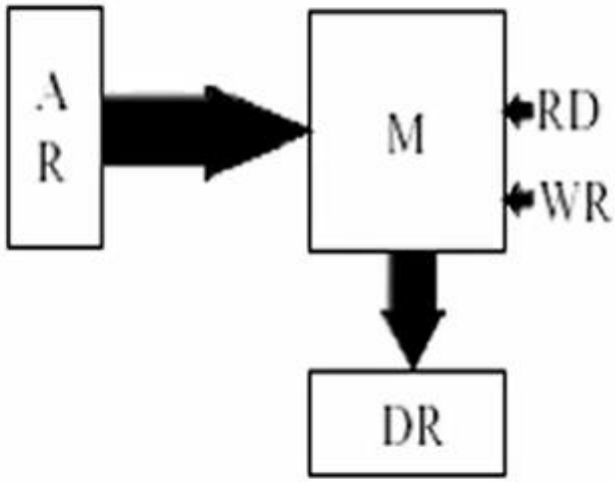
  Read : DR ← M[AR]

  (Bus ← M[AR], DR ← Bus)

- Write operation: The transfer of new information to be stored in memory.

  Write : M[AR] ← R1

  (Bus ← R1, M[AR] ← Bus)



Note:
AR: address register
DR: data register

4-3 Bus and Memory Transfers: Memory Transfer

- Memory read : Transfer from memory
- Memory write : Transfer to memory
- Data being read or wrote is called a memory word (called M)
- It is necessary to specify the address of M when writing /reading memory
- This is done by enclosing the address in square brackets following the letter M
- Example: M[0016] : the memory contents at address 0x0016

Bus and Memory Transfers: Memory Transfer

- Assume that the address of a memory unit is stored in a register called the Address Register AR
- Lets represent a Data Register with DR, then:
  - Read: DR ← M[AR]
  - Write: M[AR] ← DR

4-3 Bus and Memory Transfers: Memory Transfer