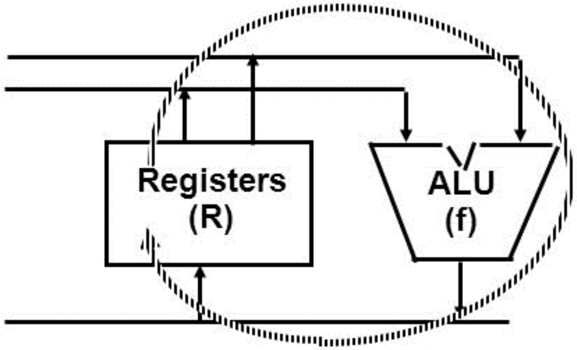
معماری کامپیوتر

فصل دو

زبان انتقال ثبات



1 clock cycle

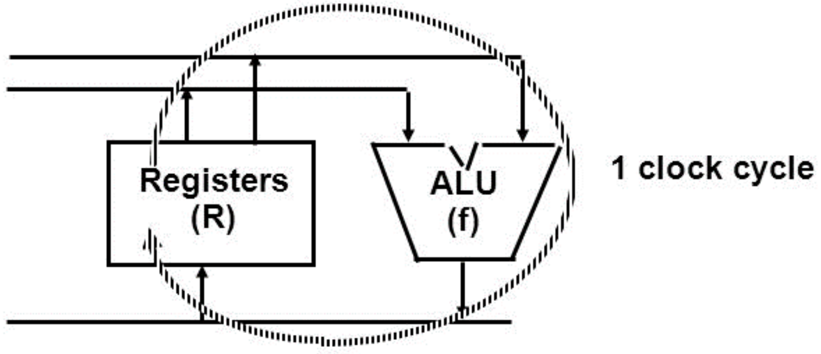
$R \leftarrow f(R, R)$

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

Computer Architecture

Chapter Two

Register Transfer Language (RTL)



$R \leftarrow f(R, R)$

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

Copyright Notice

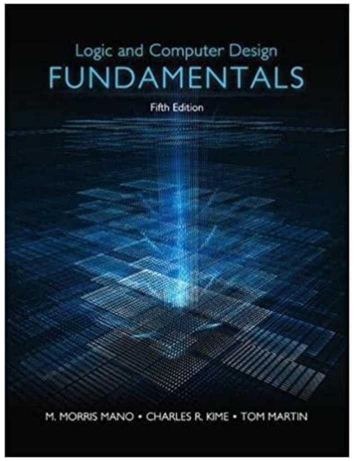
Parts (text & figures of this lecture are adopted from:

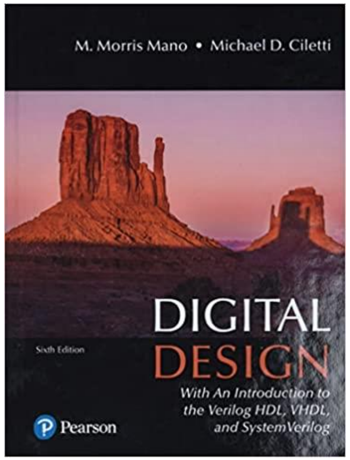
- M. M. Mano, C. R. Kime & T. Martin, “Logic & Computer Design Fundamentals”, 5th Ed., Pearson, 2015
- M. Morris Mano, “Computer System Architecture”, Pearson, 1999

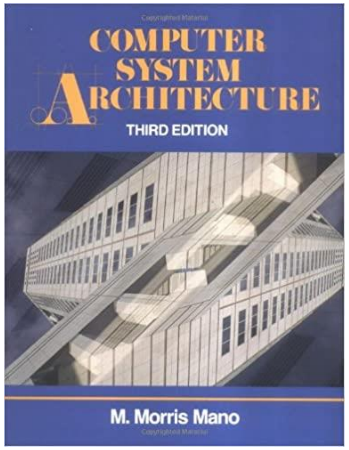
M. Morris Mano

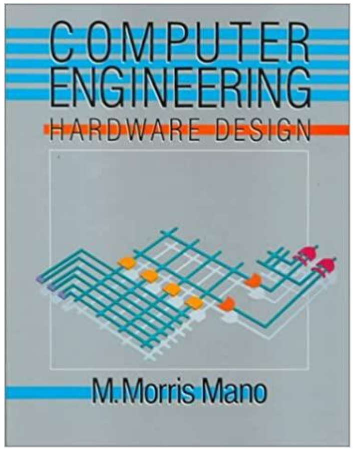
M. Morris Mano is an Emeritus Professor at California State University, Los Angeles. His subject is computer engineering. He is the author of many books pertaining to Digital Circuits, which deals with the primary digital logic circuits in an accessible manner. His books are popular for the basic-level learning.

Political theory, political theory by Rajeev Bhargava, modern political theory books, modern political theory, introduction of political theory


Logic and Computer Design
FUNDAMENTALS
Fifth Edition
M. MORRIS MANO • CHARLES R. KIME • TOM MARTIN


M. Morris Mano • Michael D. Ciletti
DIGITAL
DESIGN
Sixth Edition
With An Introduction to the Verilog HDL, VHDL, and SystemVerilog
Pearson





Contents

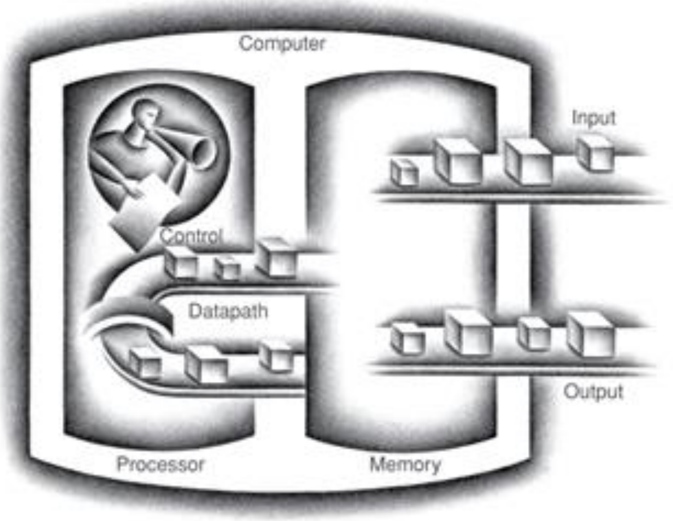
- Introduction
- RTL notifications
- Micro-operations:
  - Bus & memory transfers
  - Arithmetic, Logic, Shift
- Bus Design

Digital System

- An interconnection of digital hardware modules, that are constructed from digital components, such as:
  - Registers
  - Decoders
  - Arithmetic elements
  - Control logics

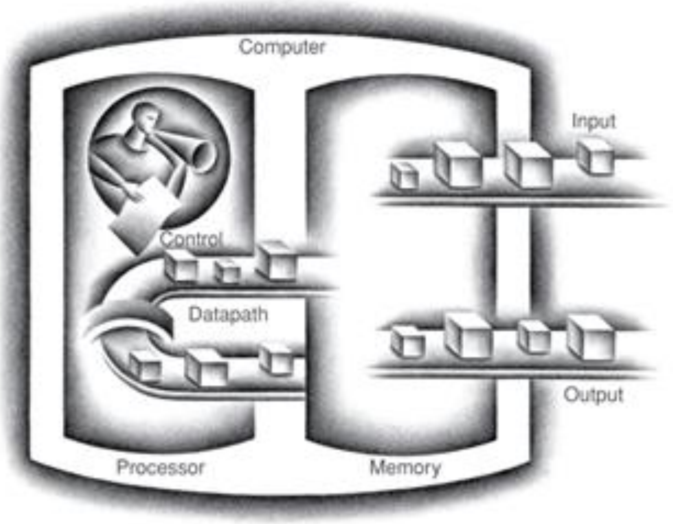
Computer Architecture

- A high-level description of the hardware from the ISA perspective
- Typically divided into
  - a datapath
  - a control unit



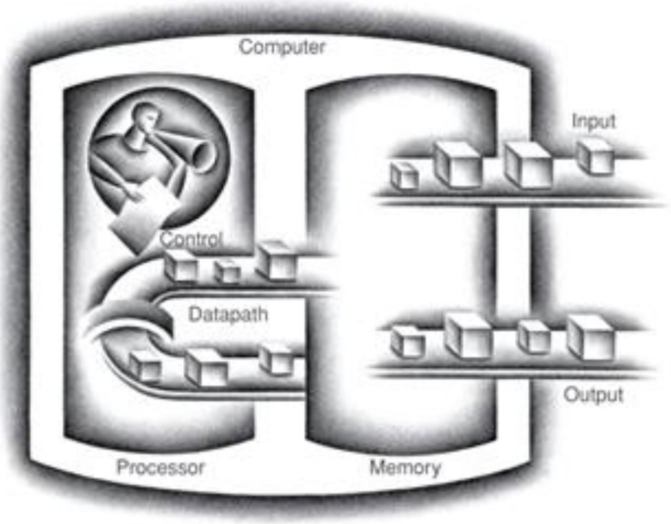
Datapath Basic Components

- A set of registers,
- The micro-operations performed on data stored in the registers
- The control interface.

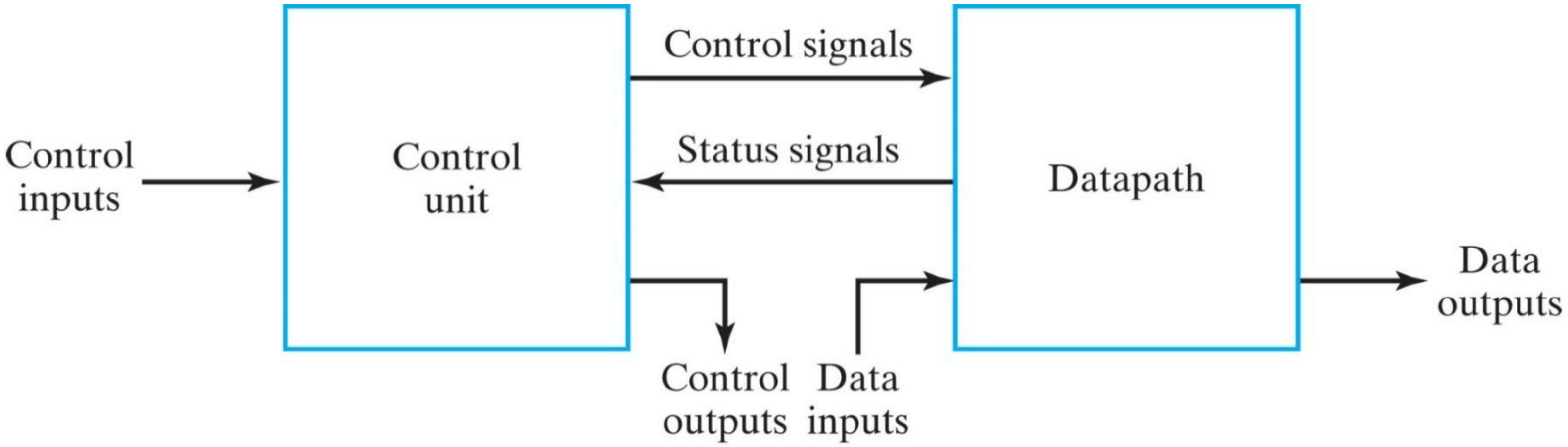


Control Unit

- Provides signals that control the micro-operations performed in the datapath and in other components of the system, such as memories



Interaction of Data Path & Control Unit



Instructions vs. Micro-operations

?

Instructions

- A group of bits that instructs a digital system (computer) to perform a specific action
- Includes:
  - Opcode (Operation code)
  - Operand(s)
- For every operation code the control unit issues a sequence of control signals to initiate μoperations in internal computer registers

Micro-Operations

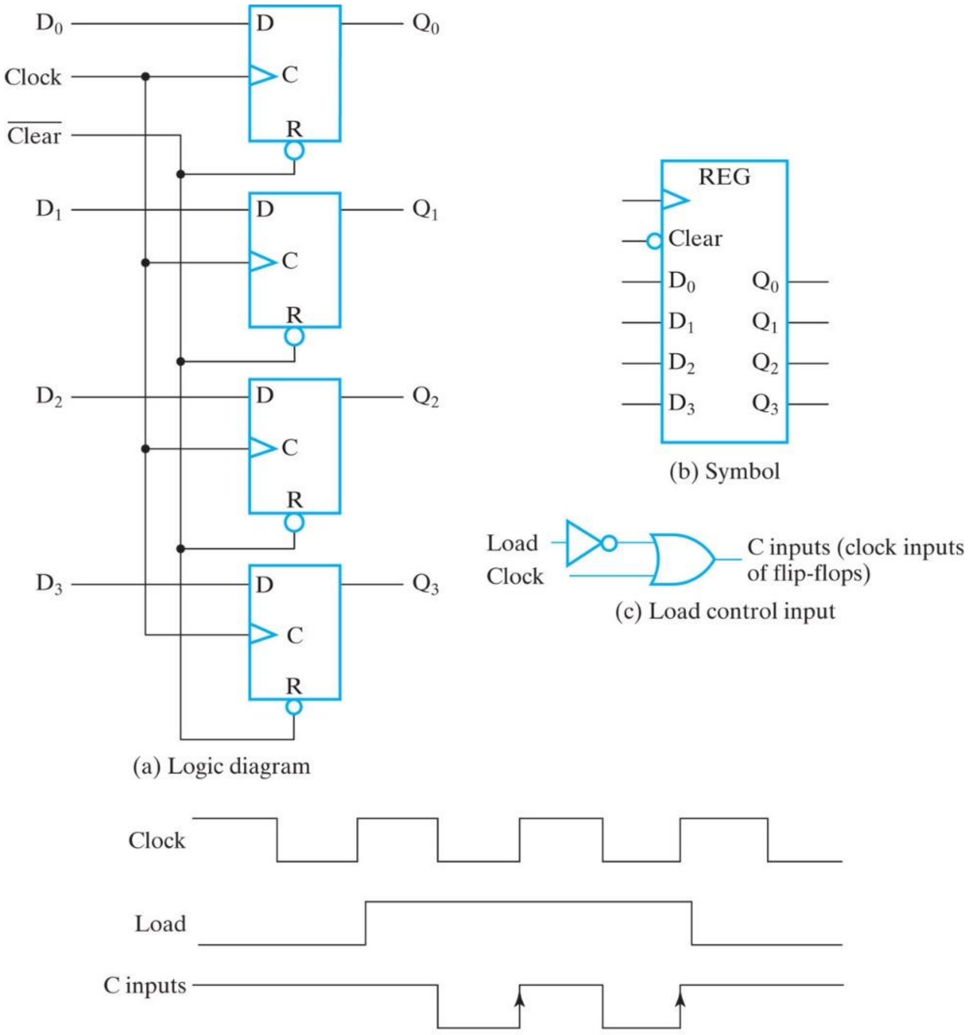
- Elementary operations executed on data stored in registers
  - Shift, count, clear, load, …
- The result of the operation may
  - Replace the previous data of the register
  - Be transferred to another register

Register Transfer Language

A system for expressing in symbolic form, the μoperation sequences among the registers of a digital module

Every RTL statement implies a hardware construction for implementing the transfer in exactly one clock cycle

4-Bit Register



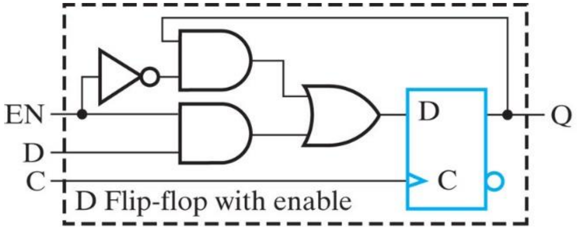
(a) Logic diagram

(b) Symbol

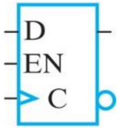
(c) Load control input

(d) Timing diagram

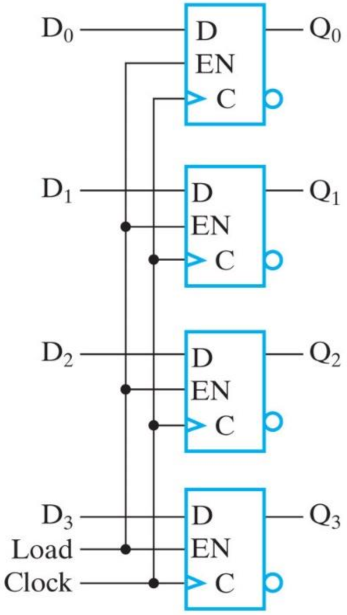
4-Bit Register with Parallel Load



(a)

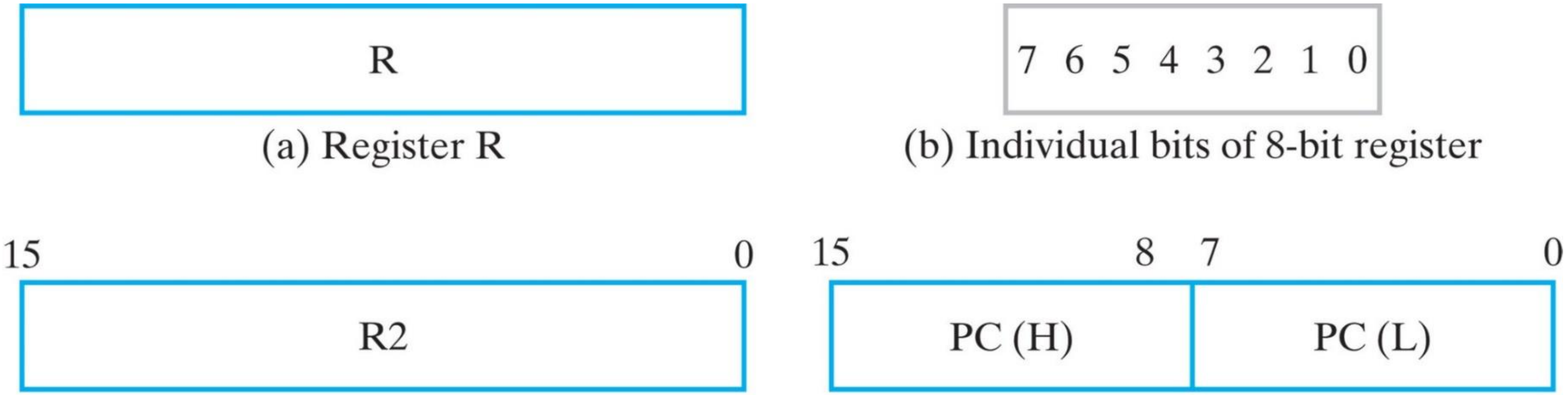


(b)



(c)

Block Diagrams of Registers



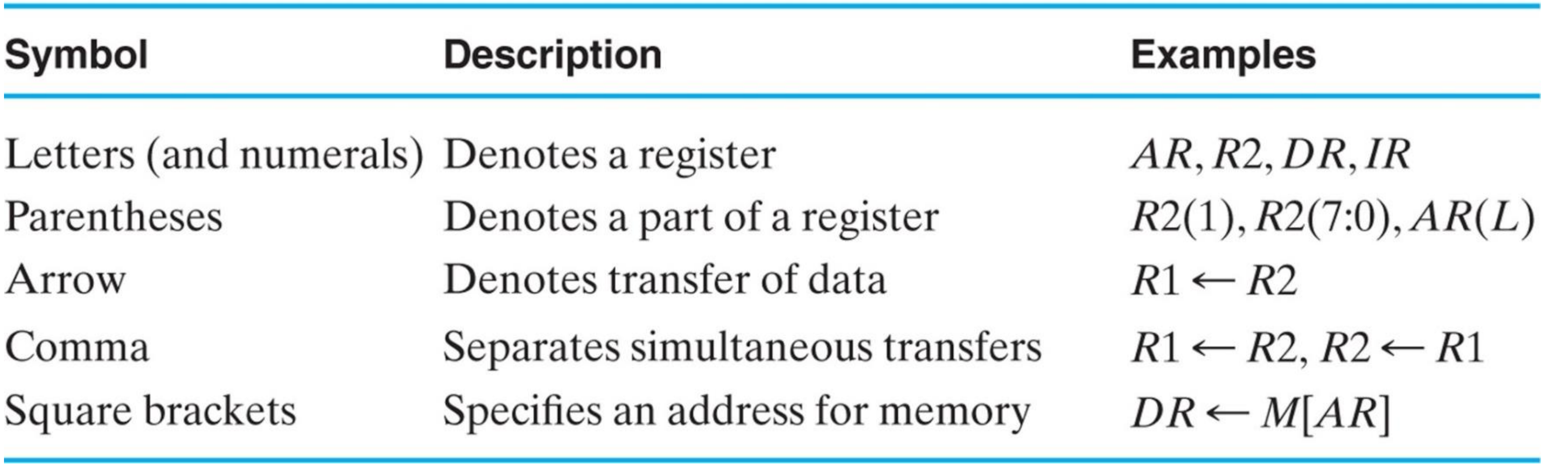
(a) Register R

(b) Individual bits of 8-bit register

(c) Numbering of 16-bit register

(d) Two-part 16-bit register

Basic Symbols for Register Transfers



Symbol	Description	Examples
Letters (and numerals)	Denotes a register	$AR, R2, DR, IR$
Parentheses	Denotes a part of a register	$R2(1), R2(7{:}0), AR(L)$
Arrow	Denotes transfer of data	$R1 \leftarrow R2$
Comma	Separates simultaneous transfers	$R1 \leftarrow R2,\ R2 \leftarrow R1$
Square brackets	Specifies an address for memory	$DR \leftarrow M[AR]$

Control Function

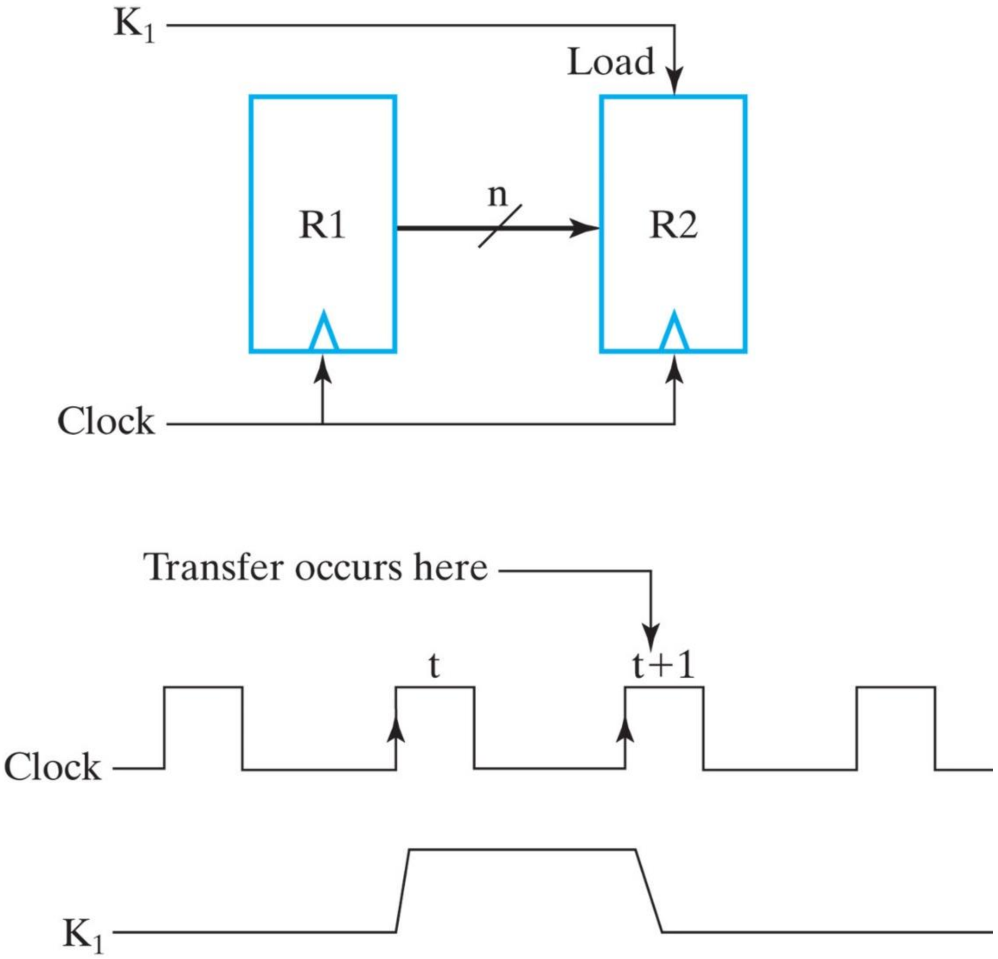
if ($K_1=1$) then ($R2 \leftarrow R1$)

is designated as:

$$K_1 : R2 \leftarrow R1$$

A Boolean variable that states the condition on which the transfer should be performed

Register Transfer



Remark!

What are the contents of R1 and R2 after this μoperation:

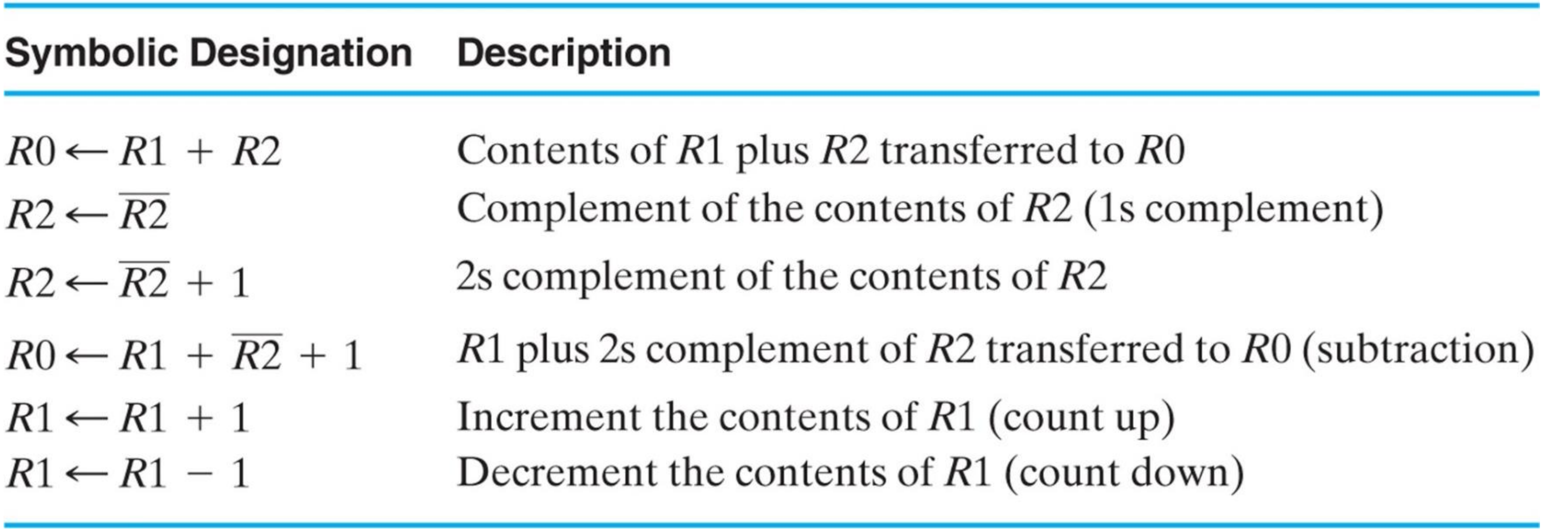
P: R2 ← R1, R1 ← R2

?

Micro-operation Types

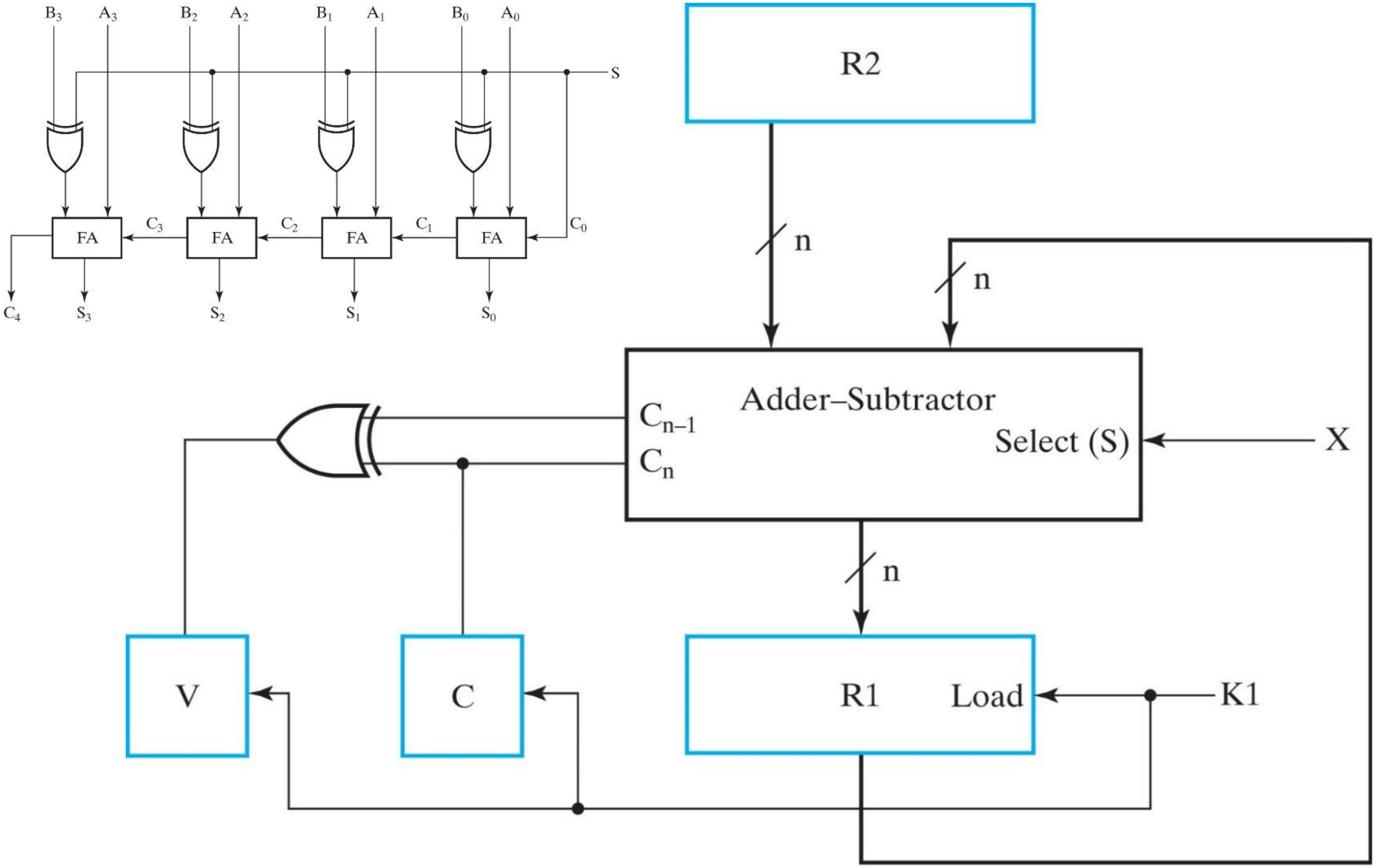
- Arithmetic μoperations
- Logic μoperations
- Shift μoperations
- Bus & memory transfers

Arithmetic Micro-Operations

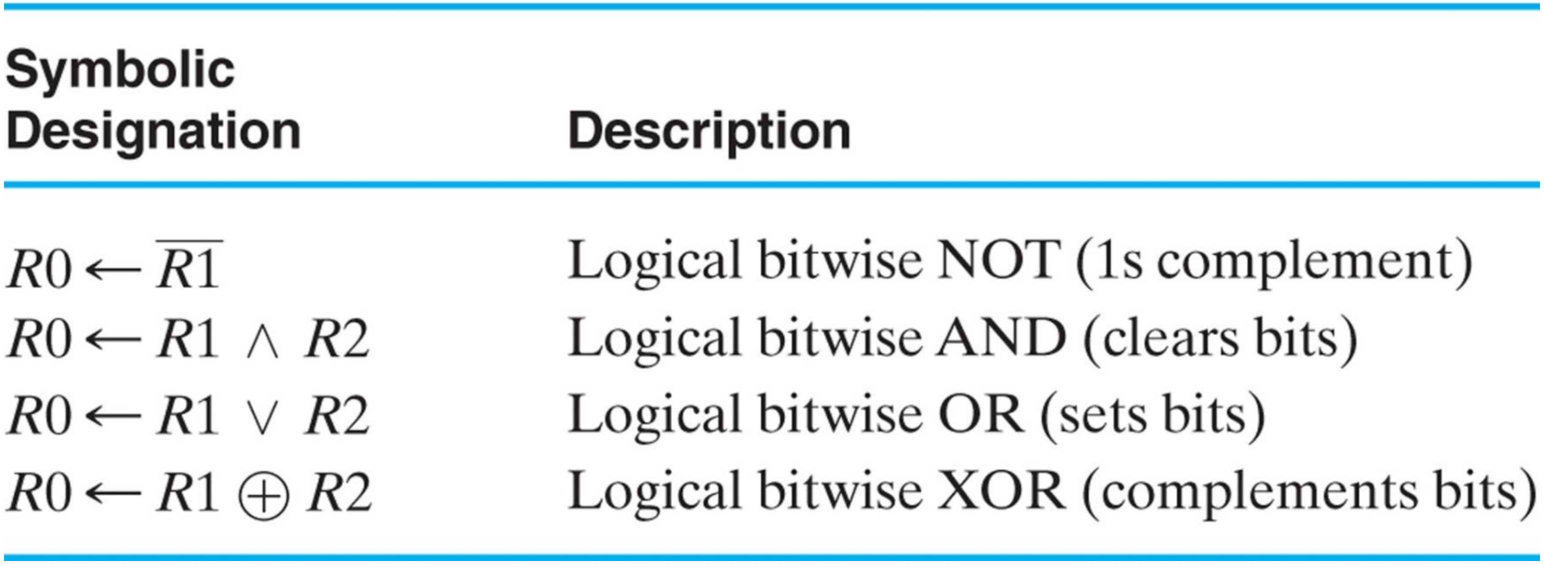


Symbolic Designation	Description
$R0 \leftarrow R1 + R2$	Contents of $R1$ plus $R2$ transferred to $R0$
$R2 \leftarrow \overline{R2}$	Complement of the contents of $R2$ (1s complement)
$R2 \leftarrow \overline{R2} + 1$	2s complement of the contents of $R2$
$R0 \leftarrow R1 + \overline{R2} + 1$	$R1$ plus 2s complement of $R2$ transferred to $R0$ (subtraction)
$R1 \leftarrow R1 + 1$	Increment the contents of $R1$ (count up)
$R1 \leftarrow R1 - 1$	Decrement the contents of $R1$ (count down)

Implementation of Add & Subtract µOps

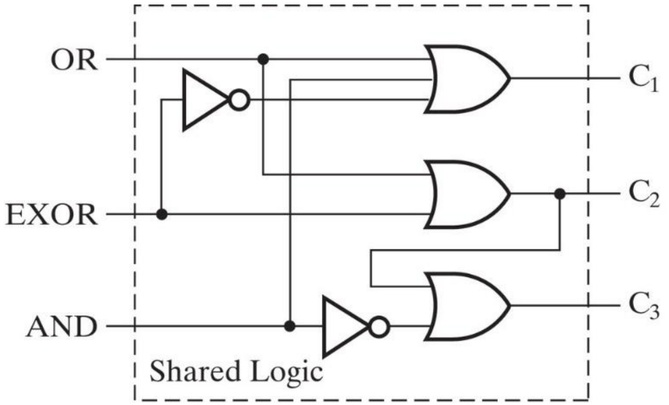


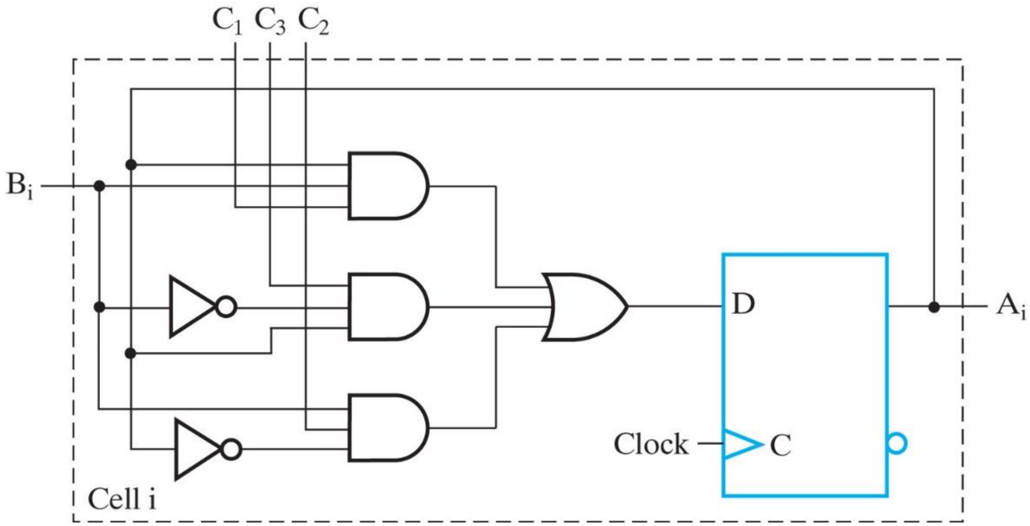
Logic Micro-Operations



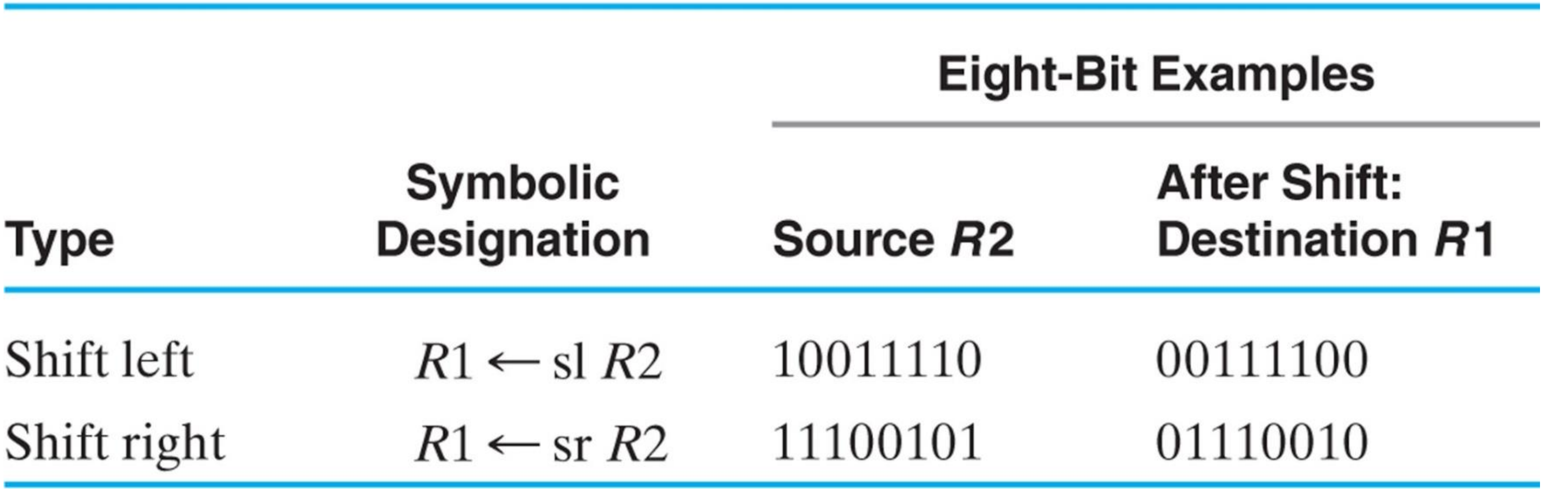
Symbolic Designation	Description
$R0 \leftarrow \overline{R1}$	Logical bitwise NOT (1s complement)
$R0 \leftarrow R1 \wedge R2$	Logical bitwise AND (clears bits)
$R0 \leftarrow R1 \vee R2$	Logical bitwise OR (sets bits)
$R0 \leftarrow R1 \oplus R2$	Logical bitwise XOR (complements bits)

Example: Register Cell Design



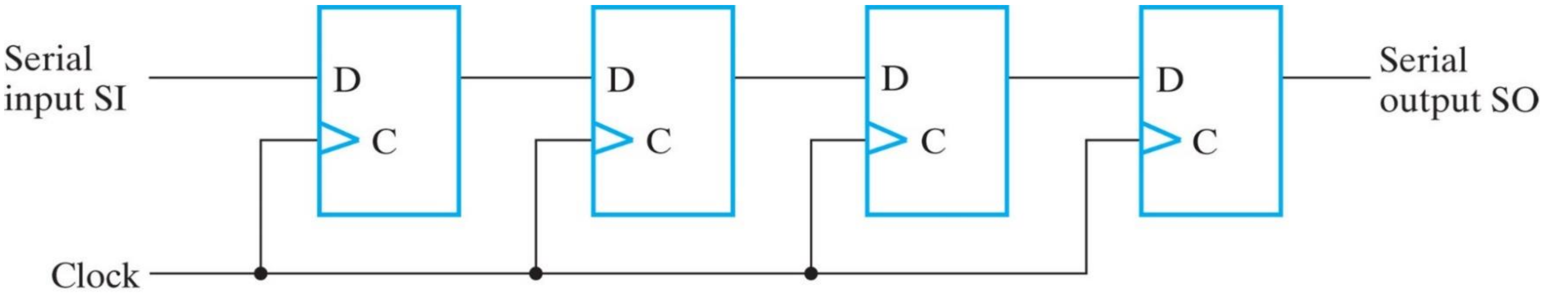


Examples of Shifts

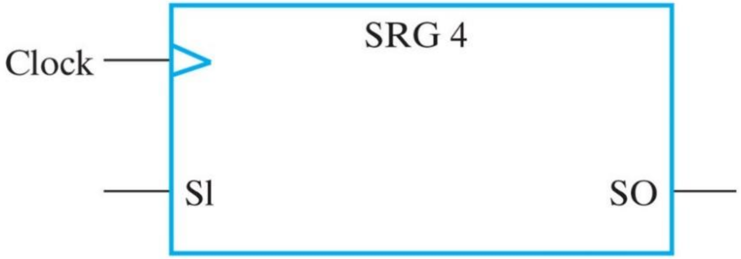


		Eight-Bit Examples	
Type	Symbolic Designation	Source *R*2	After Shift: Destination *R*1
Shift left	$R1 \leftarrow \text{sl } R2$	10011110	00111100
Shift right	$R1 \leftarrow \text{sr } R2$	11100101	01110010

4-bit Shift Register

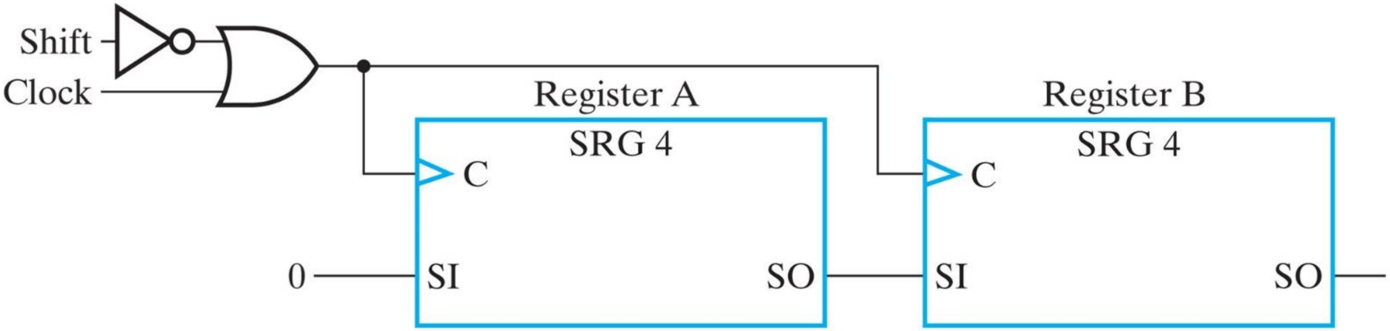


(a) Logic diagram

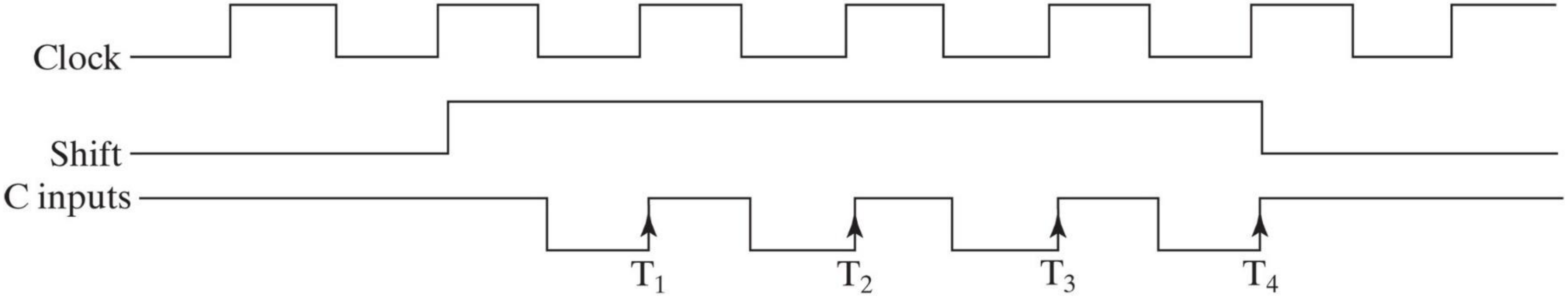


(b) Symbol

Serial Transfer

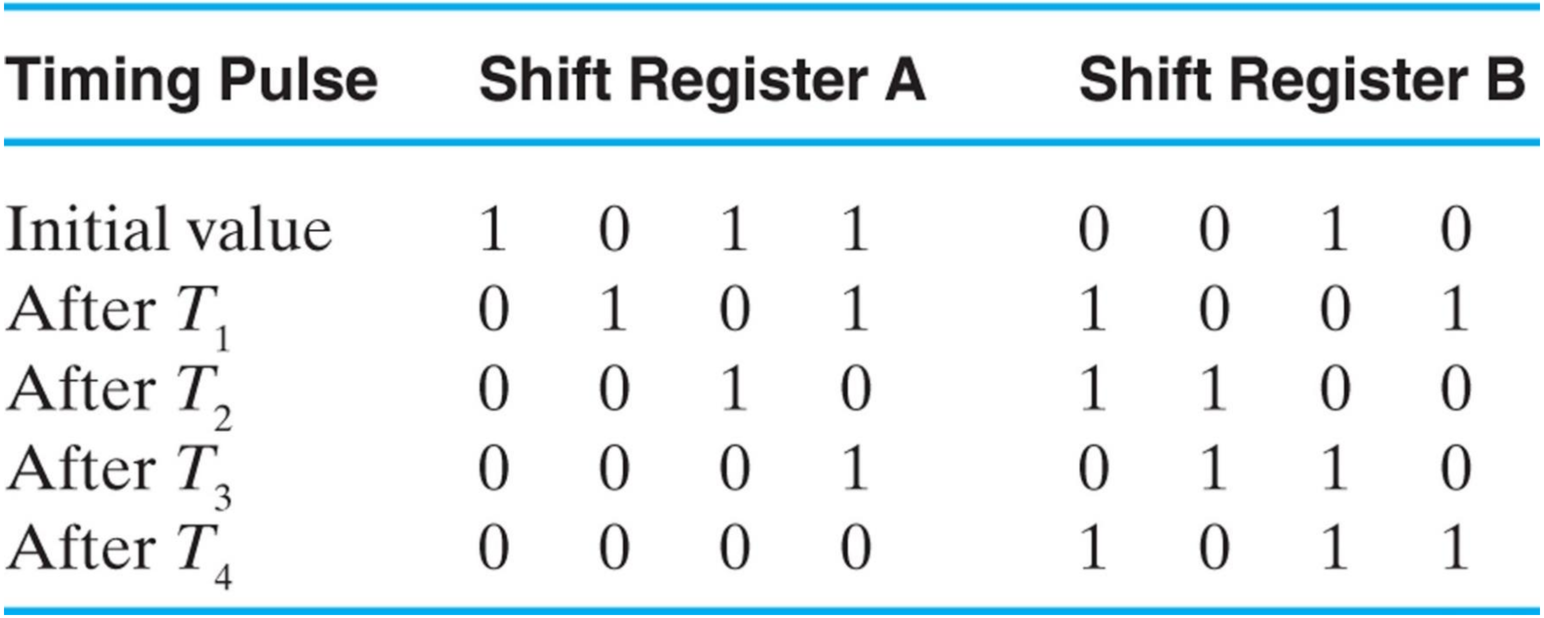


(a) Block diagram



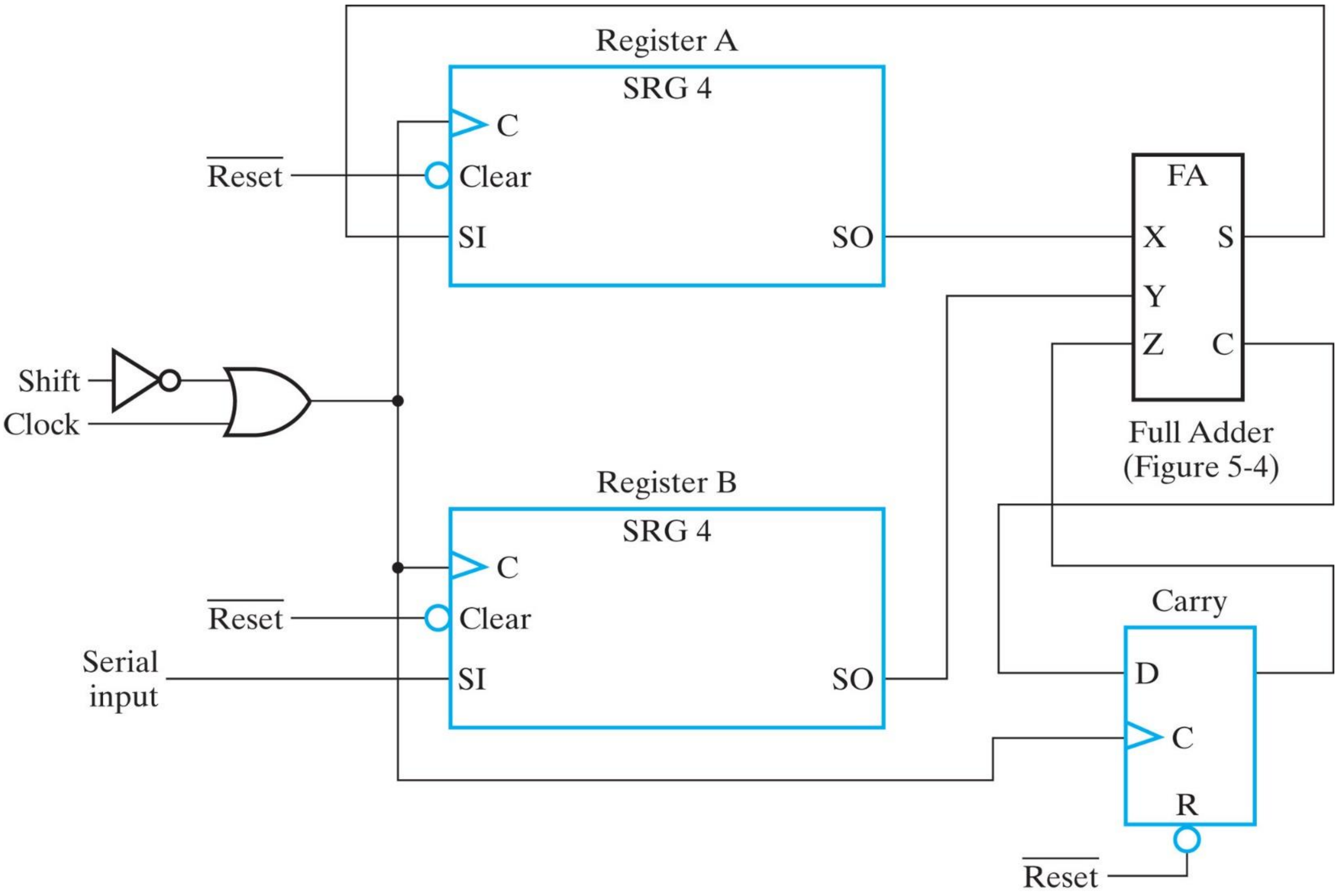
(b) Timing diagram

Example of Serial Transfer

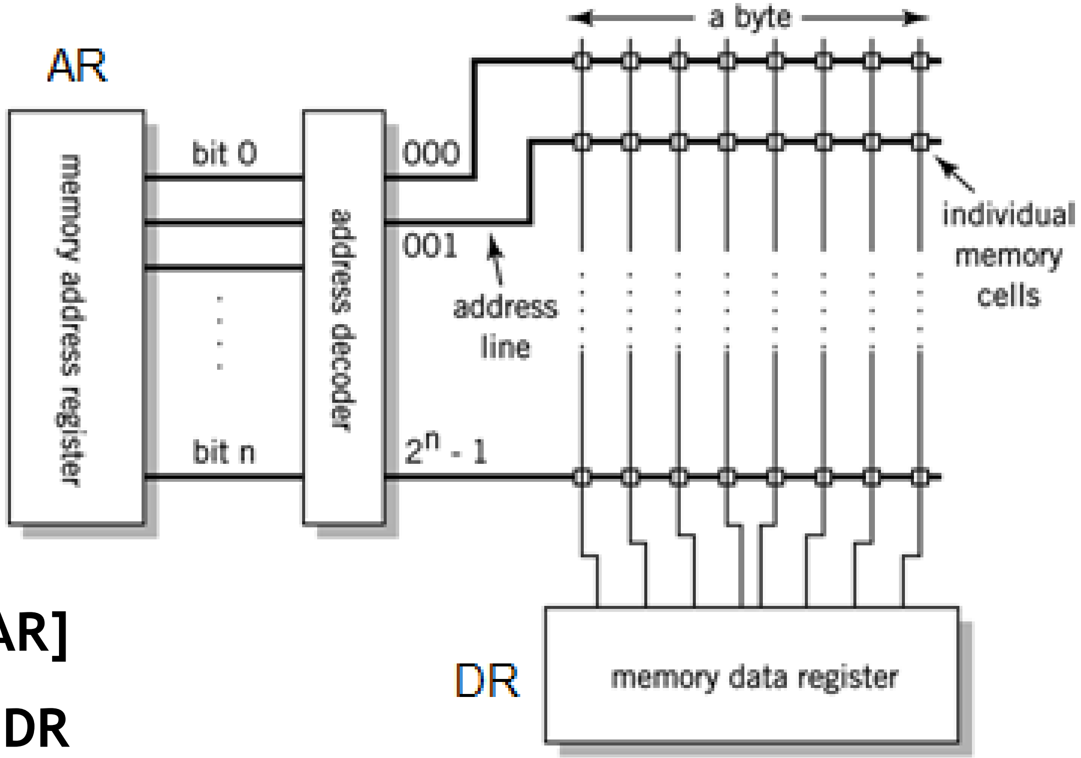


Timing Pulse	Shift Register A				Shift Register B			
Initial value	1	0	1	1	0	0	1	0
After T_1	0	1	0	1	1	0	0	1
After T_2	0	0	1	0	1	1	0	0
After T_3	0	0	0	1	0	1	1	0
After T_4	0	0	0	0	1	0	1	1

Serial Addition



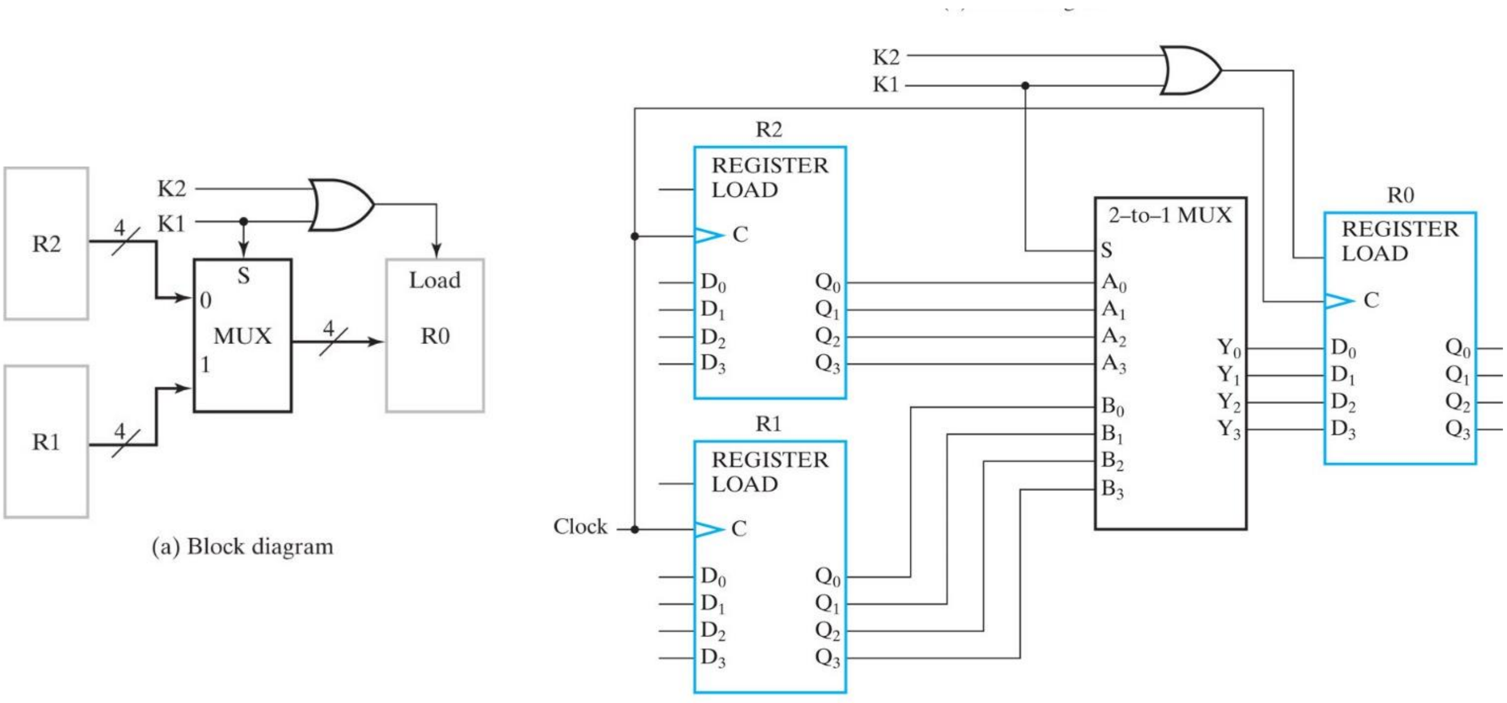
Memory Transfer



Read: DR ← M[AR]

Write: M[AR] ← DR

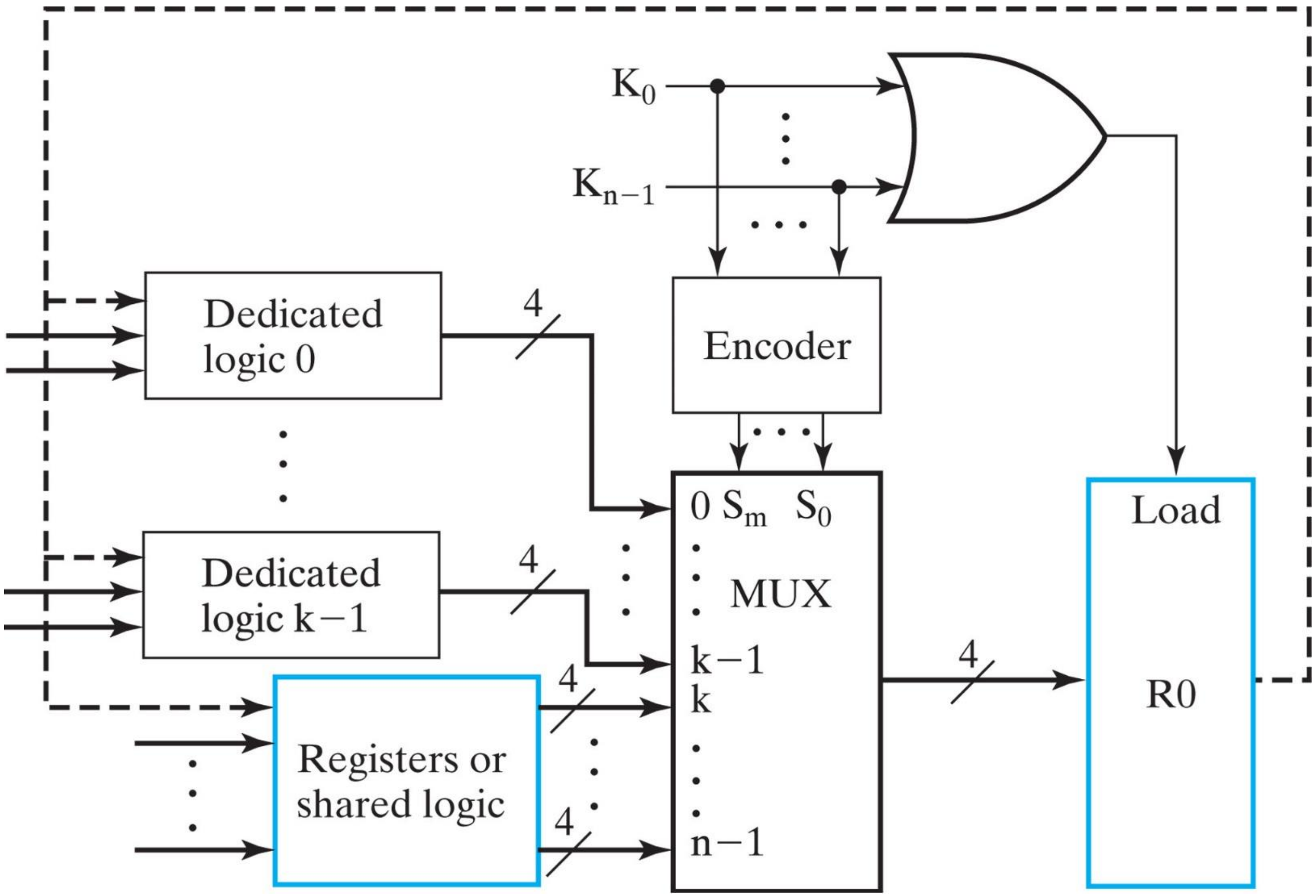
Use of Multiplexers to select between two registers



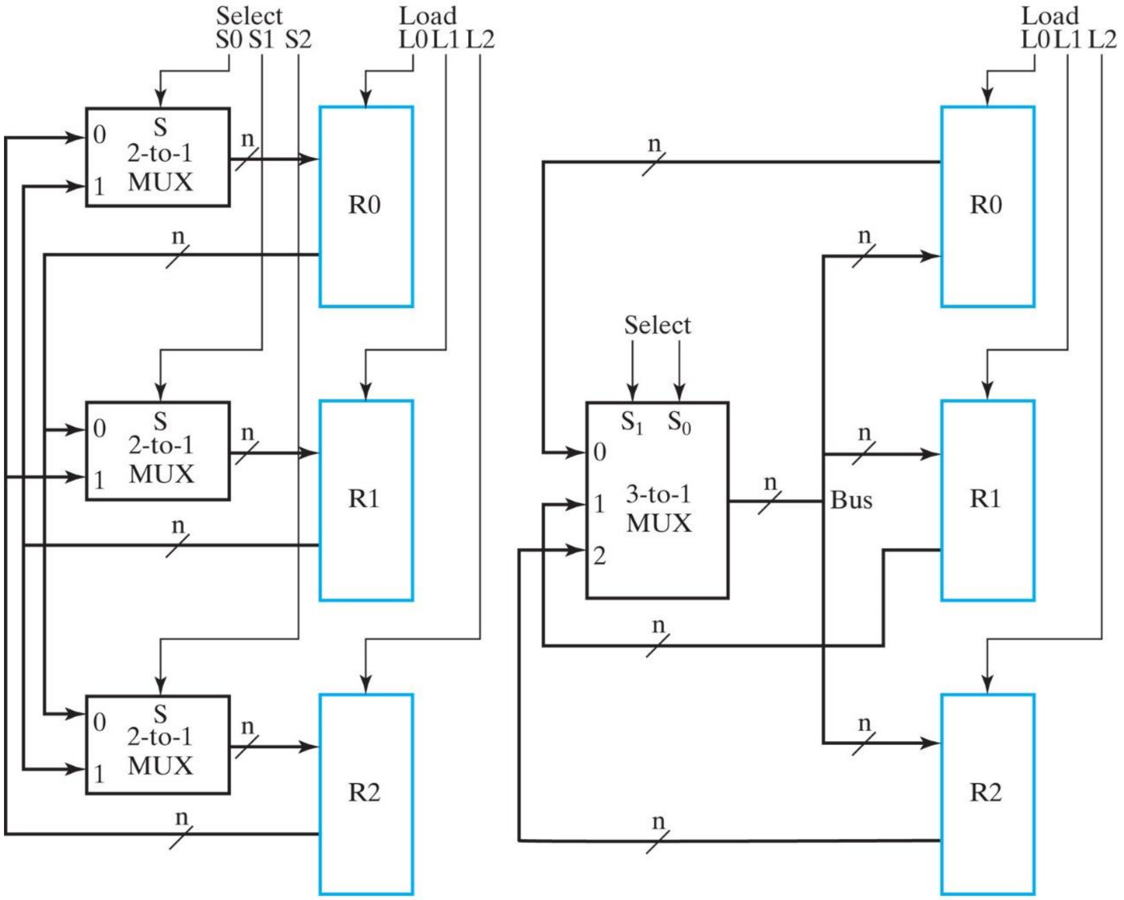
(a) Block diagram

(b) Detailed logic

Generalization of Multiplexer Selection



Single Bus vs. Dedicated Multiplexers

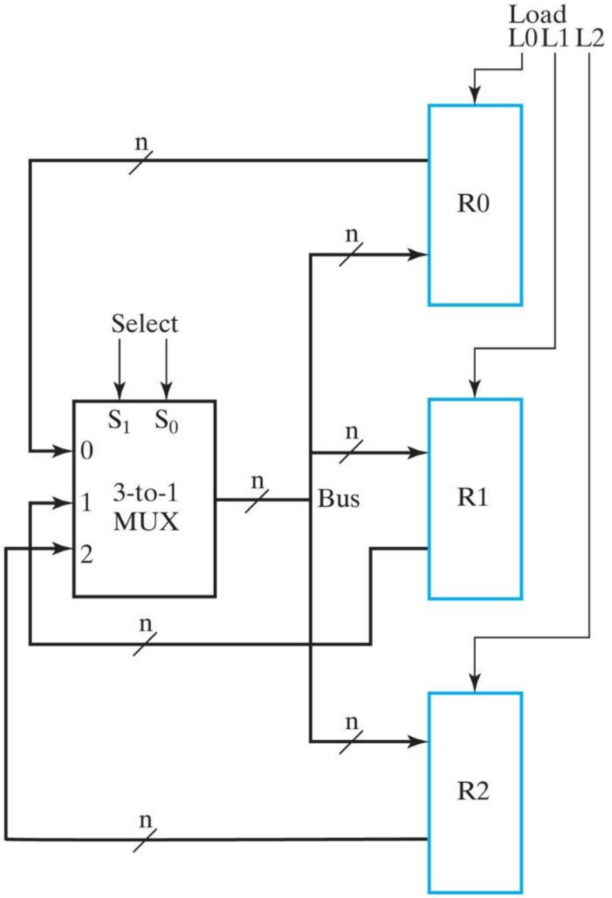


(a) Dedicated multiplexers

(b) Single bus

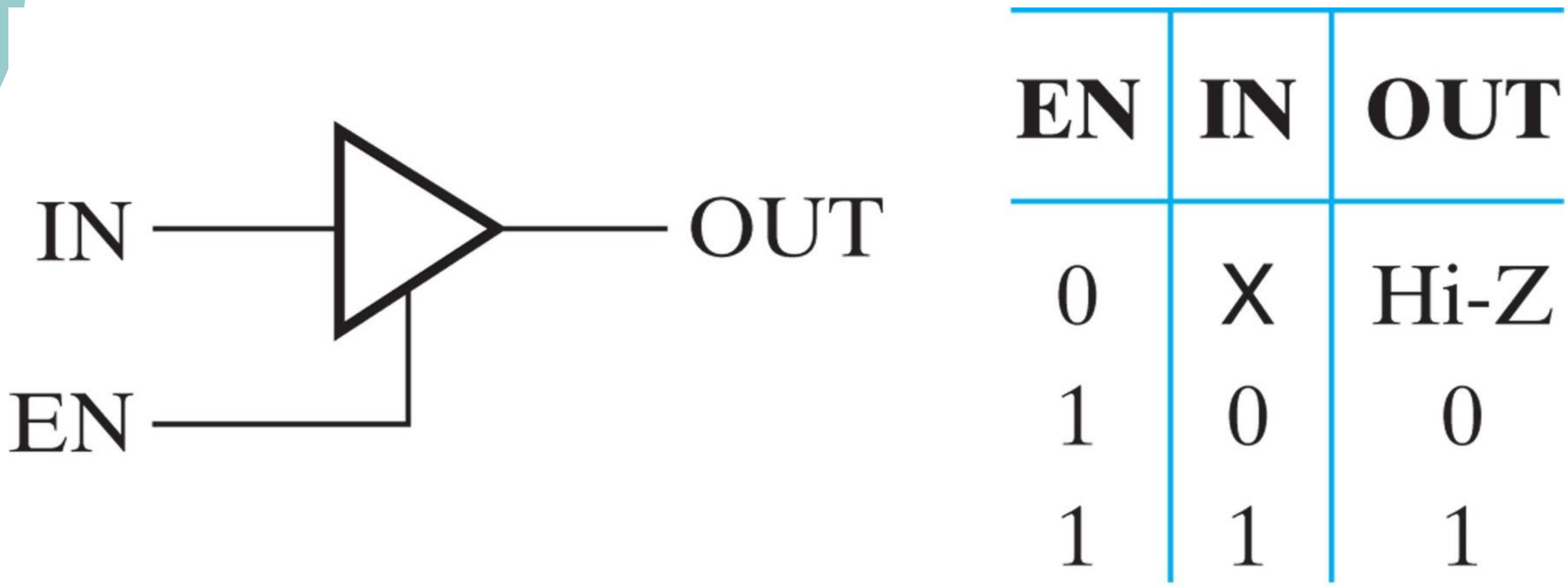
Examples of Register Transfers using the single bus

	Select		Load		
Register Transfer	S1	S0	L2	L1	L0
$R0 \leftarrow R2$	1	0	0	0	1
$R0 \leftarrow R1, R2 \leftarrow R1$	0	1	1	0	1
$R0 \leftarrow R1, R1 \leftarrow R0$	Impossible				



(b) Single bus

Three-State Buffer

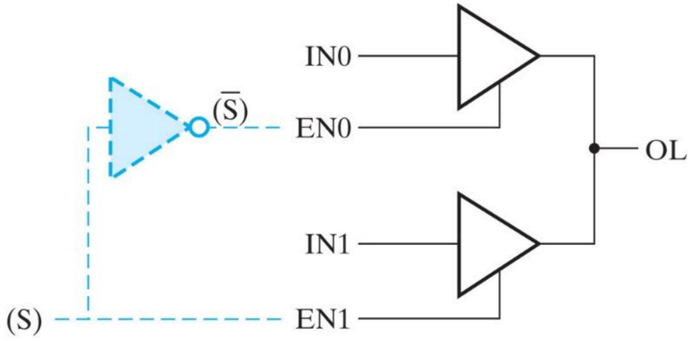


(a) Logic symbol

EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

(b) Truth table

Three-State Buffers
forming a multiplexed line OL

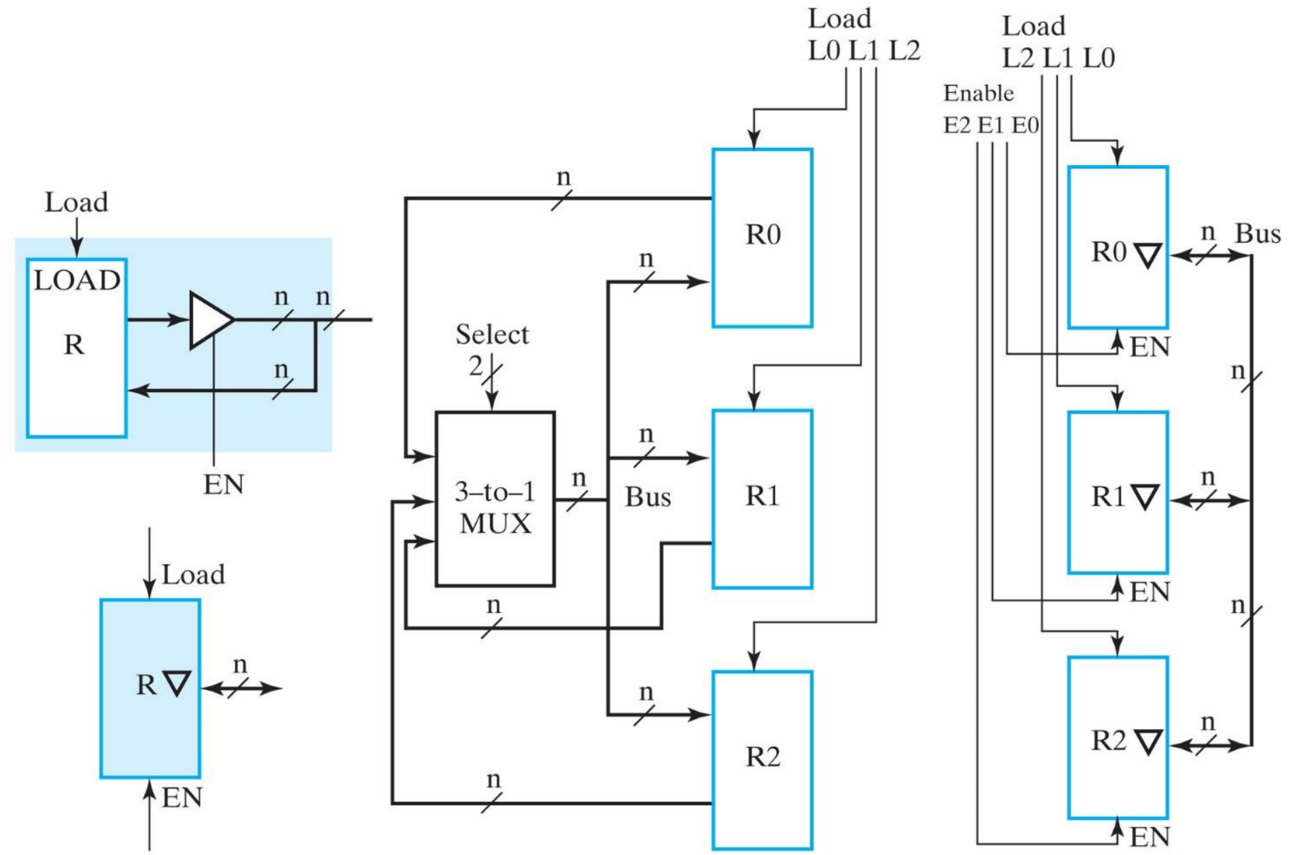


(a) Logic Diagram

EN1	EN0	IN1	IN0	OL
0	0	X	X	Hi-Z
(S) 0	($\overline{S}$) 1	X	0	0
0	1	X	1	1
1	0	0	X	0
1	0	1	X	1
1	1	0	0	0
1	1	1	1	1
1	1	0	1	💨
1	1	1	0	💨

(b) Truth table

Three-State Bus vs. Multiplexer Bus



(a) Register with bidirectional input–output lines and symbol

(b) Multiplexer bus

(c) Three-state bus using registers with bidirectional lines

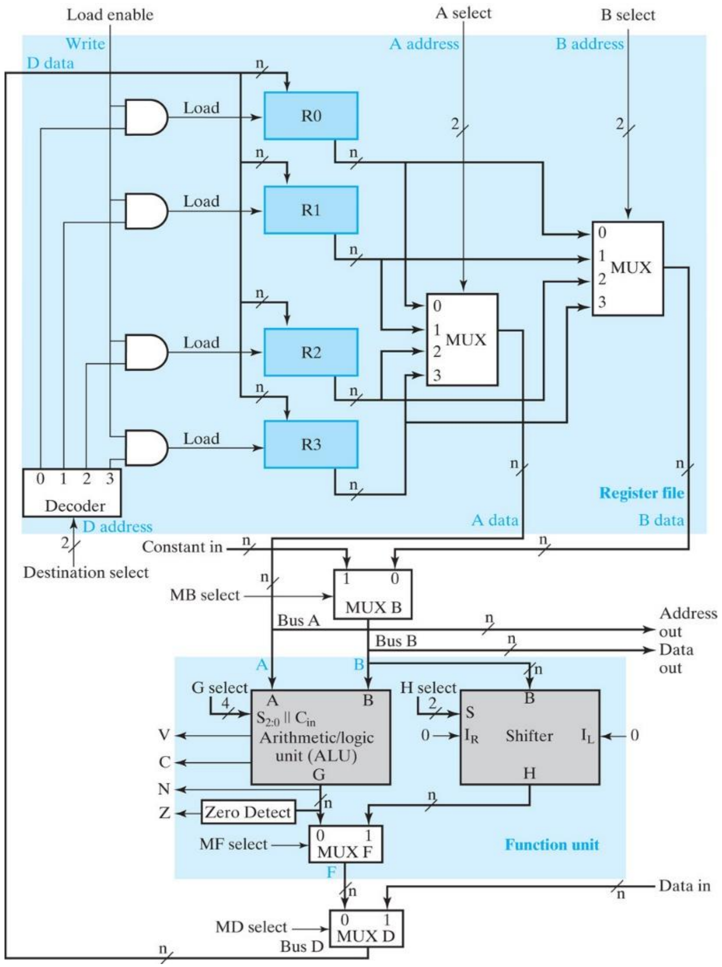
Final Overview

- Register Transfer Language (RTL):
  - The symbolic notation used to describe the μoperation transfers among registers
- Micro-operations:
  - Elementary operations executed on data stored in registers
  - Performed in exactly one clock cycle
    - Bus & memory transfers
    - Arithmetic
    - Logic
    - Shift

Internal HW Organization (in short)

- Set of registers & their functions
- Sequence of μoperations performed on them
- The control that initiates the sequence of μoperations

Block Diagram of a Generic DataPath


Load enable
A select
B select
Write
A address
B address
D data
n
Load
R0
R1
R2
R3
2
MUX
0
1
2
3
Register file
A data
B data
Decoder
D address
Destination select
Constant in
MB select
MUX B
Bus A
Bus B
Address out
Data out
A
B
G select
4
$S_{2:0}$ || C_{in}
Arithmetic/logic unit (ALU)
G
H select
S
I_R
Shifter
I_L
H
V
C
N
Z
Zero Detect
MF select
MUX F
F
Function unit
Data in
MD select
MUX D
Bus D