

# Module 7 – Register Transfers and Datapaths

**CSE-103 Digital Computer Logic**

**Athar Mahboob**

**WWW:** <http://www.atharmahboob.com>

**Email:** [athar@atharmahboob.com](mailto:athar@atharmahboob.com)

# Module Outline

---

- Datapaths and Operations
- Register Transfer Operations
- Microoperations
- Datapaths
- ALUs

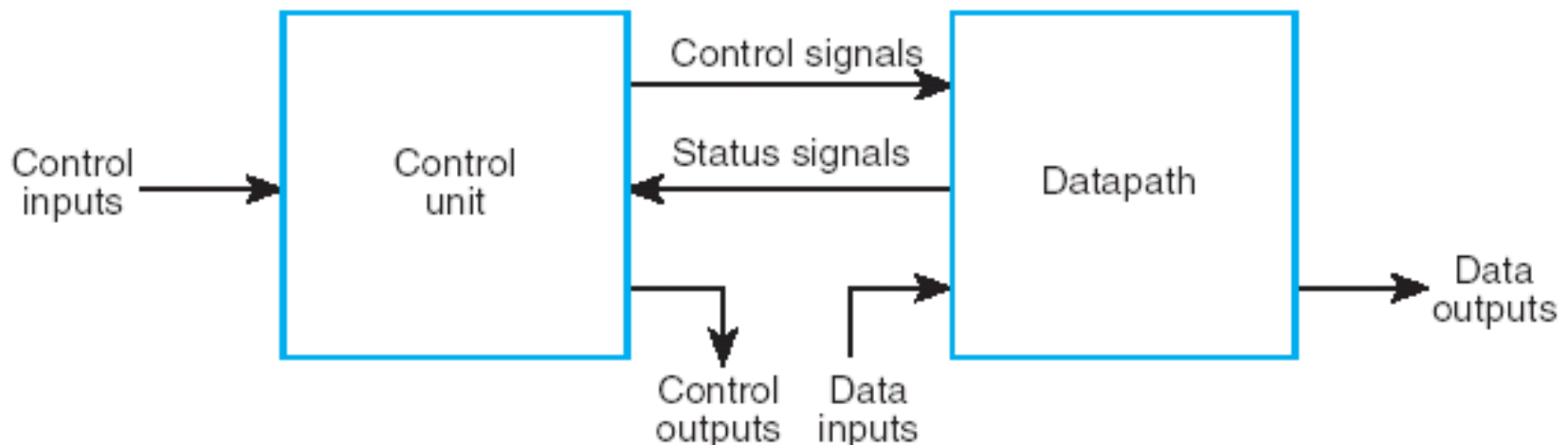
# Interaction between Datapath and Control Unit

---

A general digital system is a sequential circuit – defined by a state table – for a large digital system the state table can become unwieldy

Solution is to work with modules – design systems using well tested modules – generally two main modules datapath and control unit

What is Datapath? Performs data processing functions under the control of the Control unit



# Register Transfers and Microoperations

Registers are the basic component of a digital system

The movement of data into and out of registers and the processing of data stored in the registers are known as **Register Transfer Operations**

The elementary operations performed on the data stored in a register are called **microoperations**



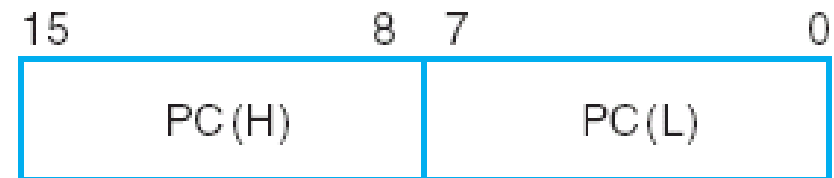
(a) Register R



(b) Individual bits of 8-bit register



(c) Numbering of 16-bit register

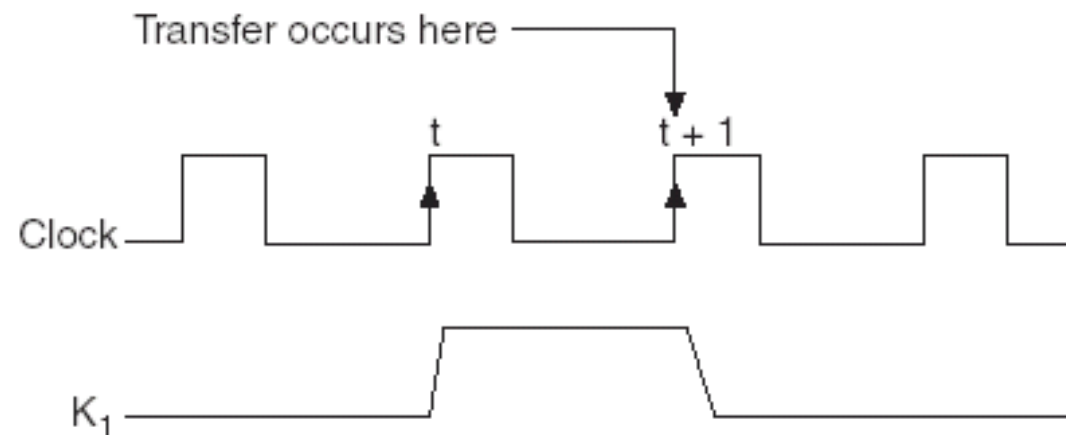
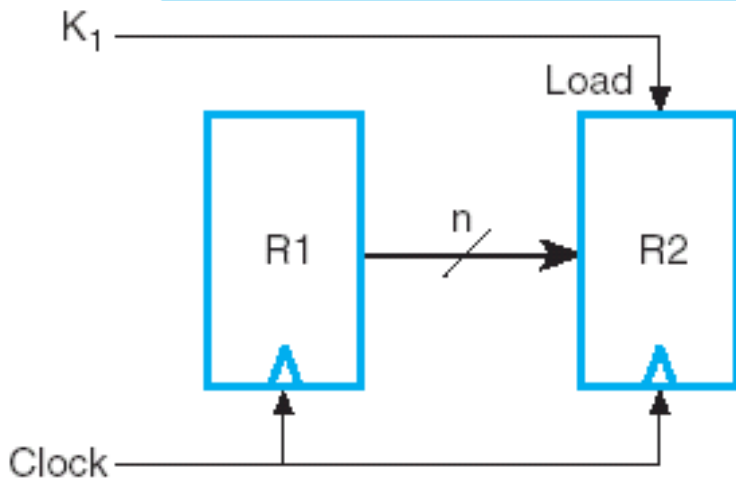


(d) Two-part 16-bit register

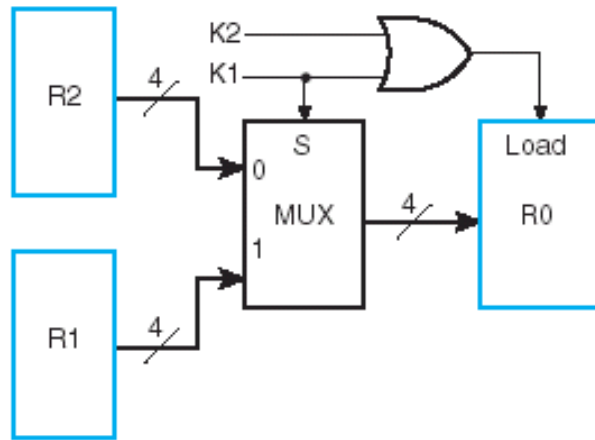
# Transfer from R1 to R2 when $K_1 = 1$

□ TABLE 7-1  
Basic Symbols for Register Transfers

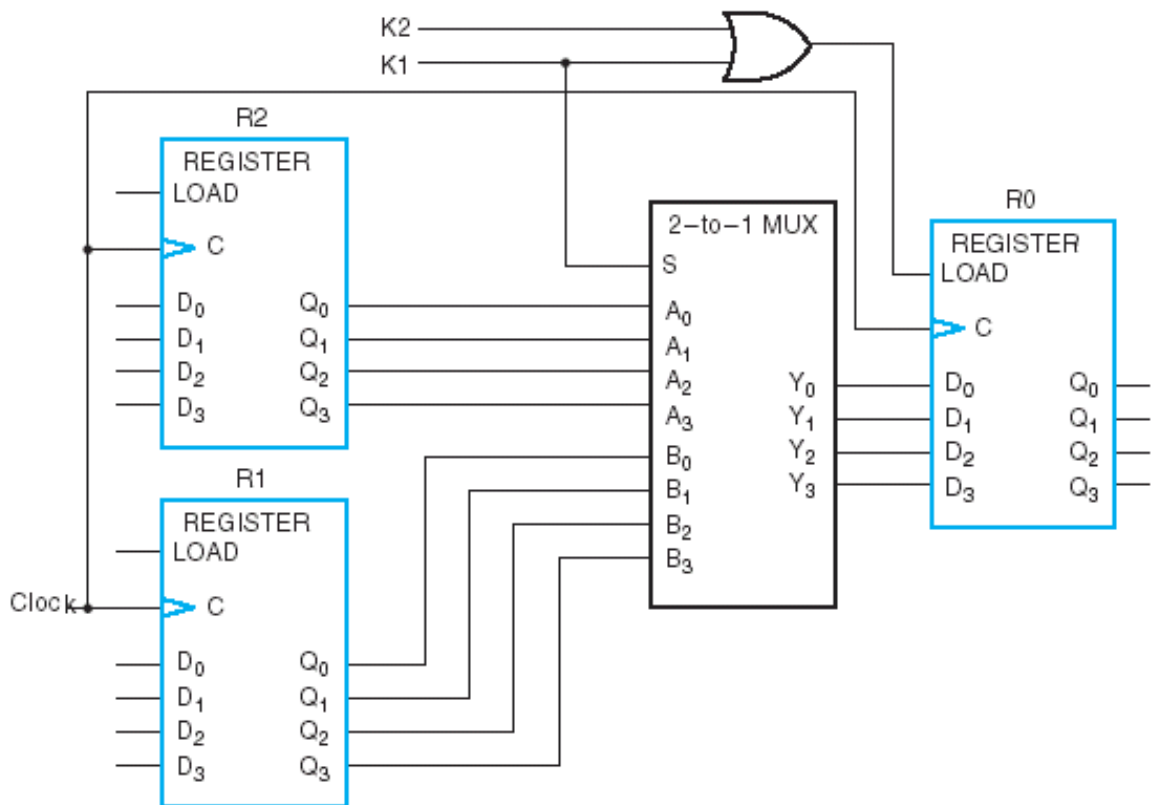
Symbol	Description	Examples
Letters (and numerals)	Denotes a register	$AR, R2, DR, IR$
Parentheses	Denotes a part of a register	$R2(1), R2(7:0), AR(L)$
Arrow	Denotes transfer of data	$R1 \leftarrow R2$
Comma	Separates simultaneous transfers	$R1 \leftarrow R2, R2 \leftarrow R1$
Square brackets	Specifies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



# Use of Multiplexers to Select between Two Registers



(a) Block diagram



(b) Detailed logic

# Textbook RTL, VHDL, and Verilog Symbols for Register Transfers

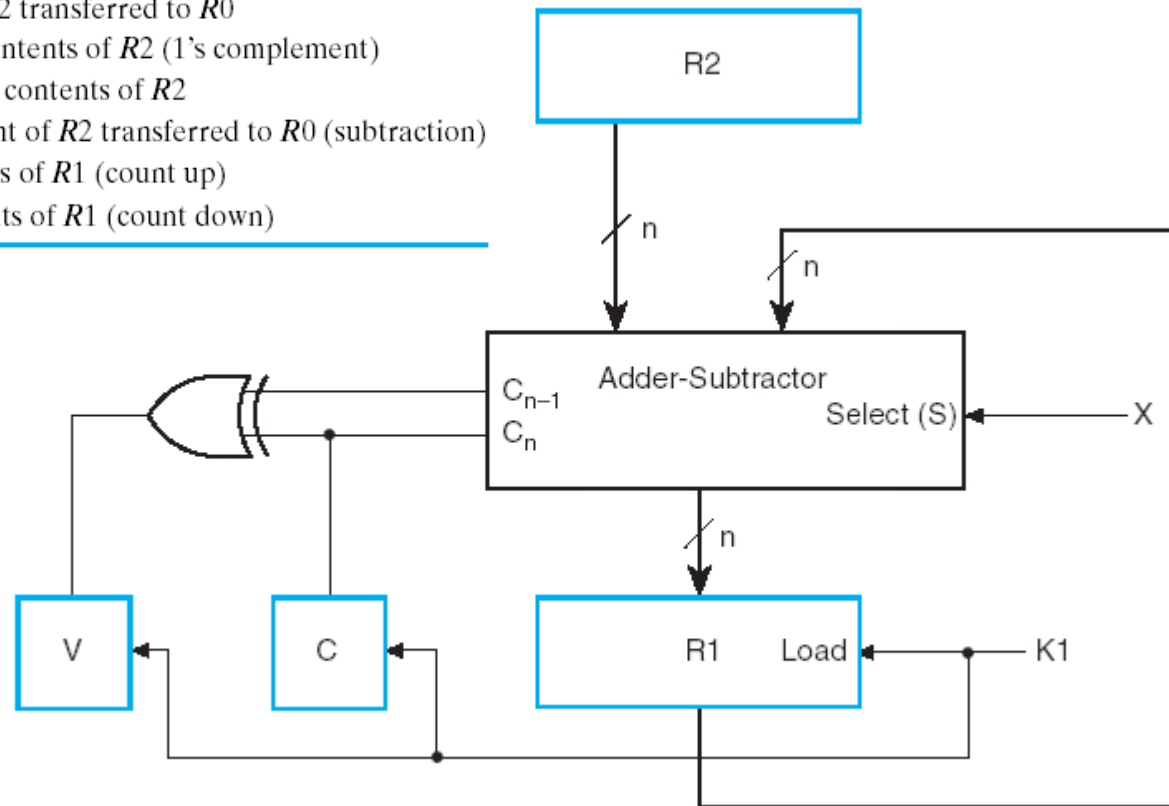
□ **TABLE 7-2**  
**Textbook RTL, VHDL, and Verilog Symbols for Register Transfers**

Operation	Text RTL	VHDL	Verilog
Combinational Assignment	=	<= (concurrent)	assign = (non-blocking)
Register Transfer	←	<= (concurrent)	<= (non-blocking)
Addition	+	+	+
Subtraction	-	-	-
Bitwise AND	^	and	&
Bitwise OR	∨	or	
Bitwise XOR	⊕	xor	^
Bitwise NOT	¯	not	~
Shift left (logical)	sl	sll	<<
Shift right (logical)	sr	srl	>>
Vectors/Registers	A(3:0)	A(3 downto 0)	A[3:0]
Concatenation		&	{, }

# Implementation of Add and Subtract Microoperations

□ TABLE 7-3  
Arithmetic Microoperations

Symbolic designation	Description
$R0 \leftarrow R1 + R2$	Contents of $R1$ plus $R2$ transferred to $R0$
$R2 \leftarrow \overline{R2}$	Complement of the contents of $R2$ (1's complement)
$R2 \leftarrow \overline{R2} + 1$	2's complement of the contents of $R2$
$R0 \leftarrow R1 + \overline{R2} + 1$	$R1$ plus 2's 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)



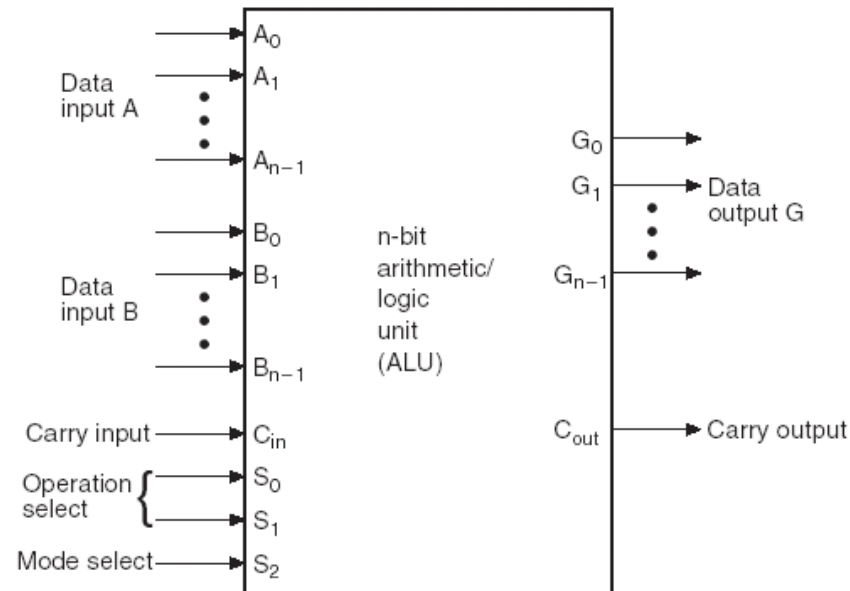
# What is an ALU?

## What is an ALU?

Instead of having each register perform its own microoperations, a shared operation unit called arithmetic/logic unit is used. ALU is a combinational circuit.

**TABLE 7-8**  
Function Table for ALU

Operation Select				Operation	Function
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	C <sub>in</sub>		
0	0	0	0	$G = A$	Transfer $A$
0	0	0	1	$G = A + 1$	Increment $A$
0	0	1	0	$G = A + B$	Addition
0	0	1	1	$G = A + \overline{B} + 1$	Add with carry input 0
0	1	0	0	$G = A + \overline{B}$	$A$ plus 1's complement
0	1	0	1	$G = A + \overline{B} + 1$	Subtraction
0	1	1	0	$G = A - 1$	Decrement $A$
0	1	1	1	$G = A$	Transfer $A$
1	0	0	X	$G = A \wedge B$	AND
1	0	1	X	$G = A \vee B$	OR
1	1	0	X	$G = \underline{A} \oplus B$	XOR
1	1	1	X	$G = A$	NOT (1's complement)



Symbol for an n-Bit ALU

# Bit Slice Architecture

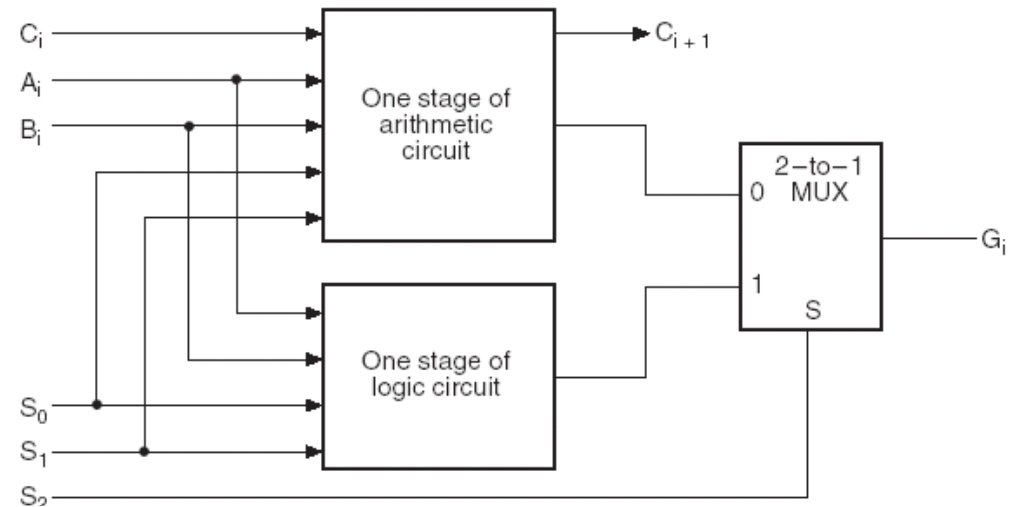
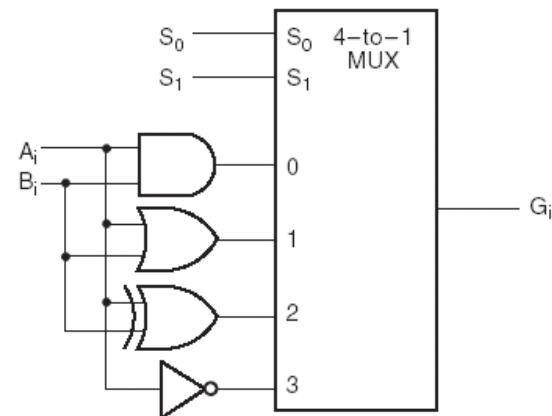


Fig. 7-15 One Stage of ALU



(a) Logic Diagram

$S_1$	$S_0$	Output	Operation
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \bar{A}$	NOT

(b) Function Table

Fig. 7-14 One Stage of Logic Circuit

# Datapath Using the Register File and Function Unit

Register file – a set of register having common microoperations performed on them

Register file is made from a special type of fast memory that allows one or more words to be read out or written into simultaneously

