Chapter 4

Register Transfer and Micro operation

A digital system is an interconnection of digital hardware modules that accomplish a specific information-processing task.

Digital systems vary in size and complexity from a few integrated circuits to a complex form of interconnected and interacting digital computers

The operations executed on data stored in registers are called micro operations. A micro operation is an elementary operation performed on the information stored in one or more registers.

Examples of micro operation are shift, count, clear, and load.

Register Transfer Language

The symbolic notation used to describe the micro operation transfers among registers is called a *register transfer language*.

Cont.........

The term “register transfer” implies the availability of hardware logic circuits that can perform a micro operation and transfer the result of operation to some other register.

Register Transfer

Computer registers are designed by capital letters (some time followed by numerals) to denote the functions of the registers. For example, Memory address register is designated by MAR, PC (program counter), IR (instruction register) and R1 (process register).

- Information transfer from one register to another is designed in symbolic form by means of a replacement operator.
- **Example, R2←R1**,
- denotes the transfer of the content of register R1 into register R2.

Cont.....

- A comma is used to separate two or more operations that are executed at the same time.
- **T: R2←R1, R1←R2**, exchanges of the contents of two registers during common clock pulse (**T=1**).

Symbol	Description	Examples
Letters (and numerals)	Denotes a register	MAR, R2
Parentheses ()	Denotes a part of a register	R2(0–7), R2(L)
Arrow ←	Denotes transfer of information	R2←R1
Comma ,	Separates two microoperations	R2←R1, R1←R2

Basic Symbols for Register Transfers

Cont…..

Bus and Memory Transfers

A typical computer has *many registers*, and *paths must be provided* to transfer information from one register to another.

An efficient way for transferring information between registers in a multiple-register configuration is a *common bus system.* A bus structure consists of a set of *common lines*, one for each bit of a register.

Two ways of constructing, a common bus system is with **multiplexers & three-State Bus Buffers**

- The symbolic statement that includes *bus transfer* is:
- **BUS ←C, R1←BUS**

  R1←C

Three-State Bus Buffers AND, NAND & Buffer gates

The bus system can be constructed with three-state gates instead of multiplexers. A three-state gate is a digital circuit that exhibits three states.

Cont……

Memory Transfer

The transfer of information from a memory word to the outside environment is called *read* operation. The transfer of new information to be stored into the memory is called a ***write*** operation.

Read: DR←M[AR]

- This causes a transfer of information into DR (Data Register) from the memory word M selected by the address in AR **Write: M[AR]←R1**

This causes a transfer of information from R1 into the memory word M selected by the address in AR.

Arithmetic and Logic Micro operations

Arithmetic Micro operations

- The basic arithmetic micro operations are ***addition***, *substruction*, *increment* and *decrement*. It Is defined by the statement
- **R3←R1 + R2**

Cont…..

- specifies an add microoperation.
- *Substruction* is most often implemented through *complementation* and *addition*.
- **Example,** $R3 \leftarrow R1 + \overline{R2} + 1$

Adding the contents of R1 to the 2’s complement of R2 is equivalent to R1−R2.

Symbolic Designation	Description
$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow \overline{R2}$	Complement the contents of R2 (1’s complement)
$R2 \leftarrow \overline{R2} + 1$	2’s complement the contents of R2 (negative)
$R3 \leftarrow R1 + \overline{R2} + 1$	R1 plus the 2’s complement of R2 (substruction)
$R1 \leftarrow R1 + 1$	Increment the contents of R1 by one
$R1 \leftarrow R1 - 1$	Decrement the contents of R1 by one

Arithmetic Microoperation

Cont…..

The increment and decrement micro operations are symbolized by plus one and minus- one operation, respectively.

Logic Micro operations

- Logic micro operations specify binary operations for strings of bits stored in registers.
- P: R1←R1 EX-OR R2

1010	Contents of R1
<u>1100</u>	Contests of R2
0110	Contents of R1 after P=1

- The contents of R1, after the execution of the micro operation, is equals to bit-by-bit exclusive- OR operation on pairs of bits in R2 and previous values of R1.
- Special symbols will be adopted for the logic micro operation OR (v), AND (∧), and complement to distinguish them from Boolean functions symbols.

Cont…..

For example,

$P + Q: R1 \leftarrow R2 + R3, R4 \vee R5$

- the + between P and Q is an OR operation between two binary variables of a control function. The + between R1 and R2 specifies an add micro operation. The OR micro operation is designated by V between registers R5 and R6.
- Logic micro operations are very useful for manipulating individual bits or a portion of a word stored in a register. They can be used to change bit values, delete a group of bits, or insert new bit values into a register

Some Applications

- Logic micro operations are very useful for manipulating individual bits or a portion of a word stored in a register. They can be used to change bit values, delete a group of bits, or insert new bit values into a register.

and also additional applications of logical micro operation are:

- Selective- set
- Selective-Compliment
- Selective-Clear

Shift Micro operations

- Shift micro operation are used for serial transfer of data. They are also used in conjunction with *arithmetic*, *logic*, and other *data processing*. The content of a register is shifted to the left or the right.
- There are three types of shifts: *logical*, *circular*, and *arithmetic*.

Cont…..

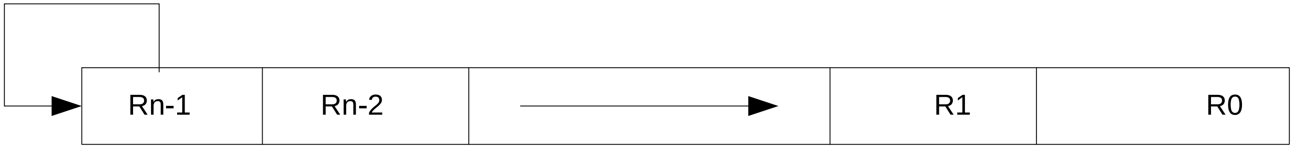
- **Example, R1←shl R1**
- **R2←shr R2**
- are two micro operations that specifies a 1-bit shift to the left of the content of register R1 and a shift to the right of the content of register R2. The register symbol must be the same on both sides of the arrow.
- The *circular shift (also known as a rotate operation)* circulates the bits of the register around the two ends without loss of information. This is accomplished by connecting the serial output of the shift register to its serial input.
- A *logical shift* is one that transfers 0 through the serial input.

Cont…..

Symbolic Destination	Description
R←shl R	Shift-left register R
R←shr R	Shift-right register R
R←cil R	Circular shift-left register R
R←cir R	Circular shift-right register R
R←ashl R	Arithmetic shift-left register R
R←ashr R	Arithmetic shift-right register R

Shift Micro operations

An arithmetic shift is a micro operation that shifts a signed binary number to the left or right. An arithmetic shift-left multiplies a signed bit binary number by 2. An arithmetic shift-right divides the number by 2.



Sign bit

- The leftmost bit in a register holds the sign bit, and the remaining bits hold the number. The sign bit is 0 for positive and 1 for negative.
- The arithmetic shift-left inserts a 0 into $R_{0,}$ and shifts all other bits to the left. The initial bit of R_{n-1} is lost and replaced by the bit from R_{n-2}.

  $V_s = R_{n-1}\ R_{n-2}$
- If Vs=0, there is no overflow, but if Vs=1, there is an overflow and a sign reversal after the shift. V_s must be transferred into the overflow flip-flop with the same clock pulse that shifts the register.