University Of Diyala
College Of Engineering
Computer Engineering Department

Digital System Design II

Dr. Yasir Amer Al-Zubaidi

Third stage

2021

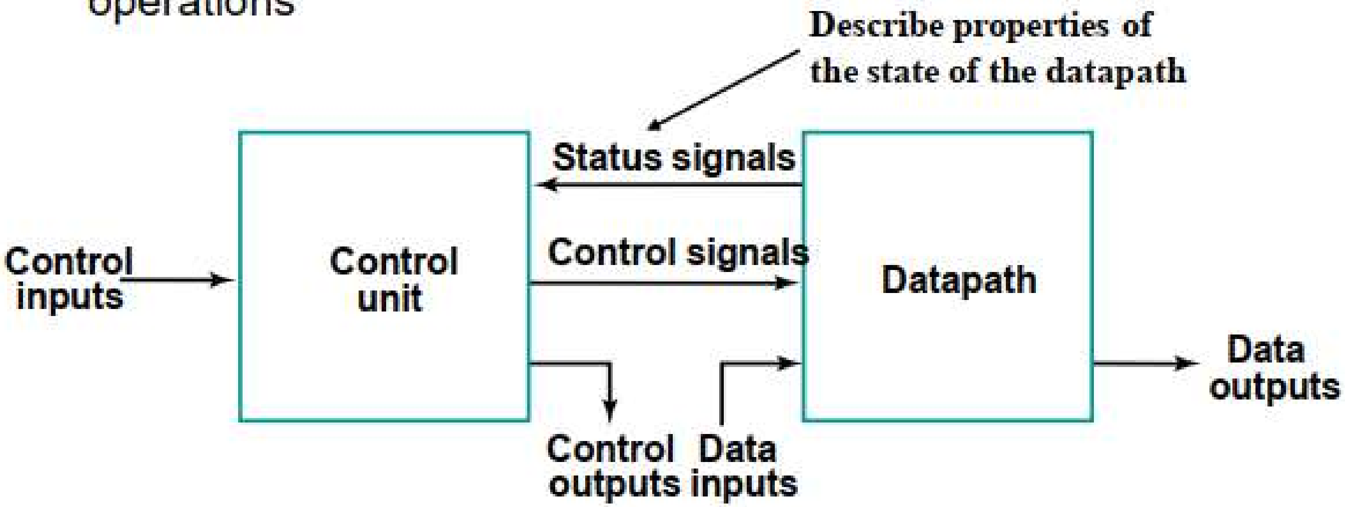
Design of Digital Sequential Circuits Using New Methods

Microprogramming Overview

- **Data path and Control**
- **Microoperations**
- **Sequencing and control**

Datapath and Control

- Datapath - performs data transfer and processing operations
- Control Unit - Determines the enabling and sequencing of the operations



- The control unit receives:
  - External control inputs
  - Status signals
- The control unit sends:
  - Control signals
  - Control outputs

Overview

- Datapath and control
- Microoperations
  - Register transfer operations
  - Microoperations - arithmetic, logic, and shift
  - Register cell design
  - Serial transfers and microoperations
- Sequencing and control

Register Transfer Operations

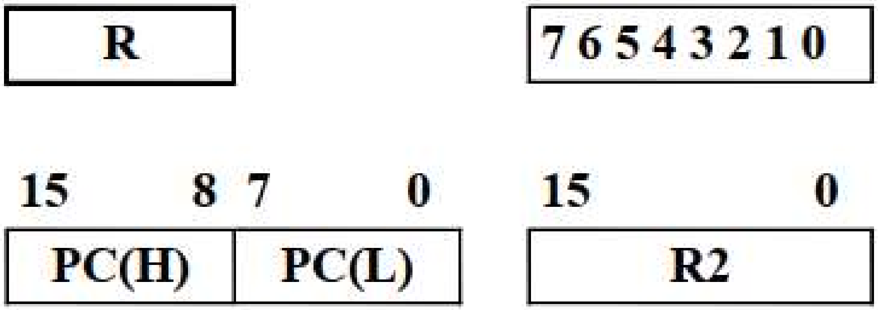
- Register Transfer Operations – the movement and processing of data stored in registers

- Three basic components:
  - A set of registers (operands)
  - Transfer operations
  - Control of operations

- Elementary operations -- called *microoperations*
  - load, count, shift, add, bitwise "OR", etc.

Register Notation

- Letters and numbers – register (e.g. R2, PC, IR)
- Parentheses ( ) – range of register bits (e.g. R1(1), PC(7:0), AR(L))



R

7 6 5 4 3 2 1 0

15 8 7 0

PC(H) PC(L)

15 0

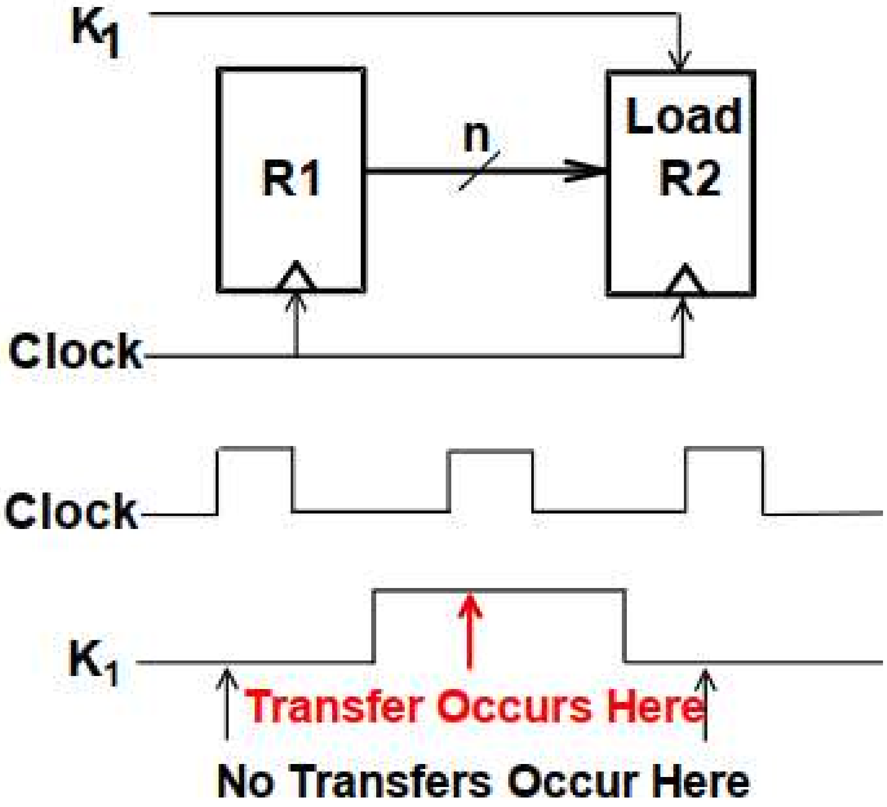
R2

- Arrow (←) – data transfer (ex. R1 ← R2, PC(L) ← R0)
- Brackets [ ] – Specifies a memory address (ex. R0 ← M[AR], R3 ← M[PC])
- Comma – separates parallel operations

Conditional Transfer

- If (K_1 =1) then (R2 ← R1) ⇔ K_1: (R2 ← R1)

  where K_1 is a control expression specifying a conditional execution of the microoperation.



Microoperations

- Logical groupings:
  - Transfer - move data from one set of registers to another
  - Arithmetic - perform arithmetic on data in registers
  - Logic - manipulate data or use bitwise logical operations
  - Shift - shift data in registers

Arithmetic operations

+ Addition
– Subtraction
* Multiplication
/ Division

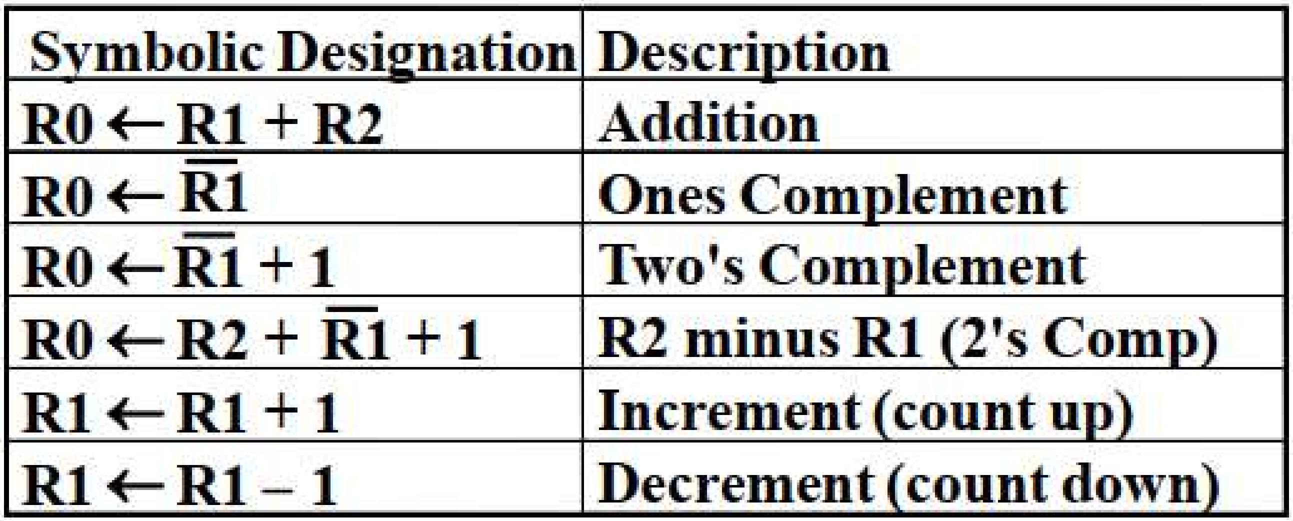
Logical operations

∨ Logical OR
∧ Logical AND
⊕ Logical Exclusive OR
‾ Not

Example Microoperations

- $R1 \leftarrow R1 + R2$
  - Add the content of R1 to the content of R2 and place the result in R1.
- $PC \leftarrow R1 * R6$
- $R1 \leftarrow R1 \oplus R2$
- $(K1 + K2)\colon\ R1 \leftarrow R1 \vee R3$
  - On condition K1 OR K2, the content of R1 is Logic bitwise Ored with the content of R3 and the result placed in R1.
  - NOTE: "+" (as in $K_1 + K_2$) means “OR.” In $R1 \leftarrow R1 + R2$, + means “plus.”

Arithmetic Microoperations



Symbolic Designation	Description
$R0 \leftarrow R1 + R2$	Addition
$R0 \leftarrow \overline{R1}$	Ones Complement
$R0 \leftarrow \overline{R1} + 1$	Two's Complement
$R0 \leftarrow R2 + \overline{R1} + 1$	R2 minus R1 (2's Comp)
$R1 \leftarrow R1 + 1$	Increment (count up)
$R1 \leftarrow R1 - 1$	Decrement (count down)

- Any register may be specified for source 1, source 2, or destination.
- These simple microoperations operate on the whole word

Logical Microoperations

Symbolic Designation	Description
$R0 \leftarrow \overline{R1}$	Bitwise NOT
$R0 \leftarrow R1 \vee R2$	Bitwise OR (sets bits)
$R0 \leftarrow R1 \wedge R2$	Bitwise AND (clears bits)
$R0 \leftarrow R1 \oplus R2$	Bitwise EXOR (complements bits)

Shift Microoperations

- Let R2 = 11001001

Symbolic Designation	Description	R1 content
R1 ← sl R2	Shift Left	10010010
R1 ← sr R2	Shift Right	01100100

- **Note: These shifts "zero fill". Sometimes a separate flip-flop is used to provide the data shifted in, or to "catch" the data shifted out.**
- **Other shifts are possible (rotates, arithmetic)**