

CSE 140 Lecture 16

System Designs

Methodology.

CK Cheng

CSE Dept.

UC San Diego

GPU

CPU

Video image proc

⋮

1

System Designs

- Introduction
 - Methodology and Framework
- Components
- Specification
- Implementation

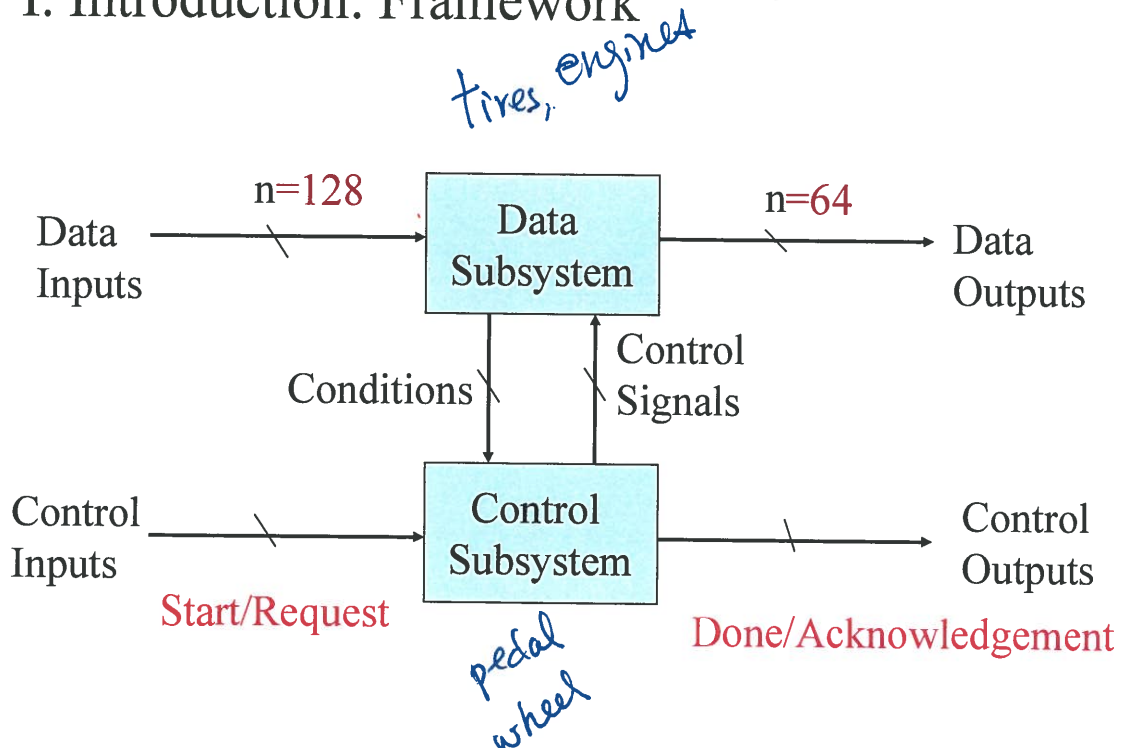
2

Introduction

- Methodology
 - Approach with success stories.
 - Hierarchical designs with interface between the modules (BSV).
- Data Subsystem and Control Subsystem
 - For n-bit data, each operation takes n times or more in hardware complexity.
 - Data subsystem carries out the data operations and transports.
 - Control system sequences the data subsystem and itself.

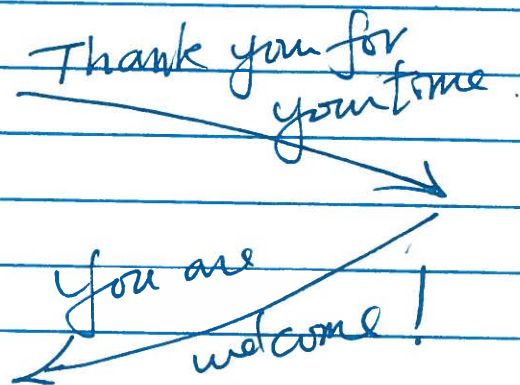
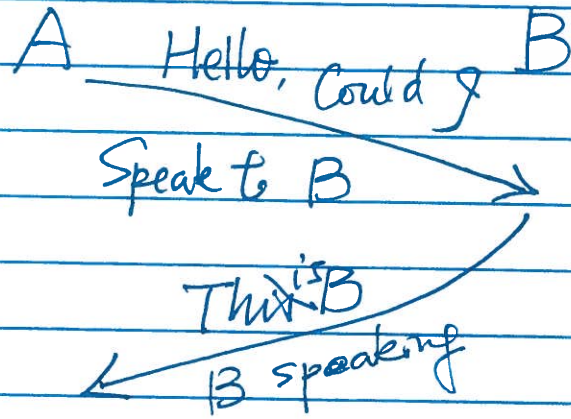
3

I. Introduction: Framework

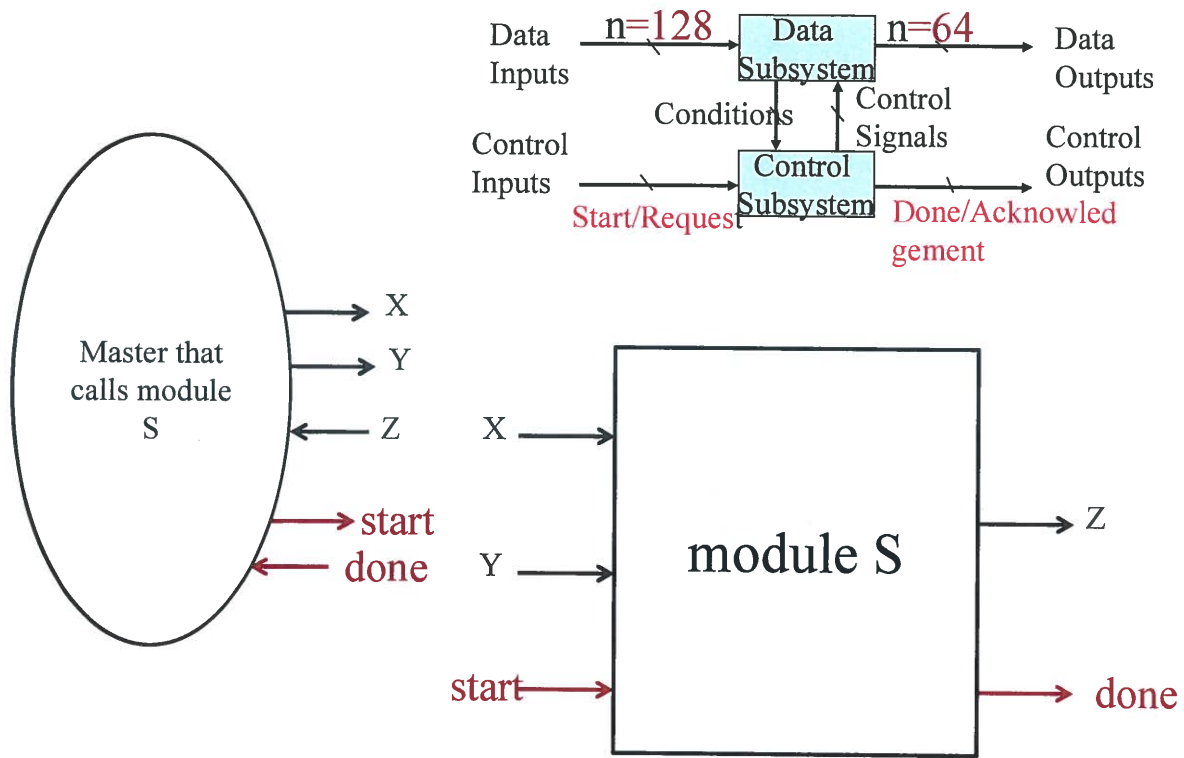


4

4-Phase Handshaking

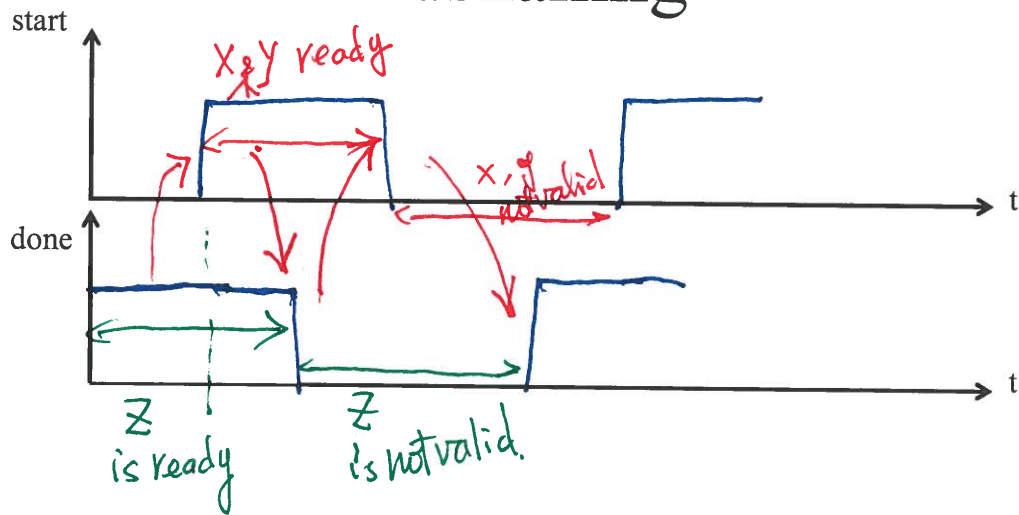


I. Introduction: Handshaking



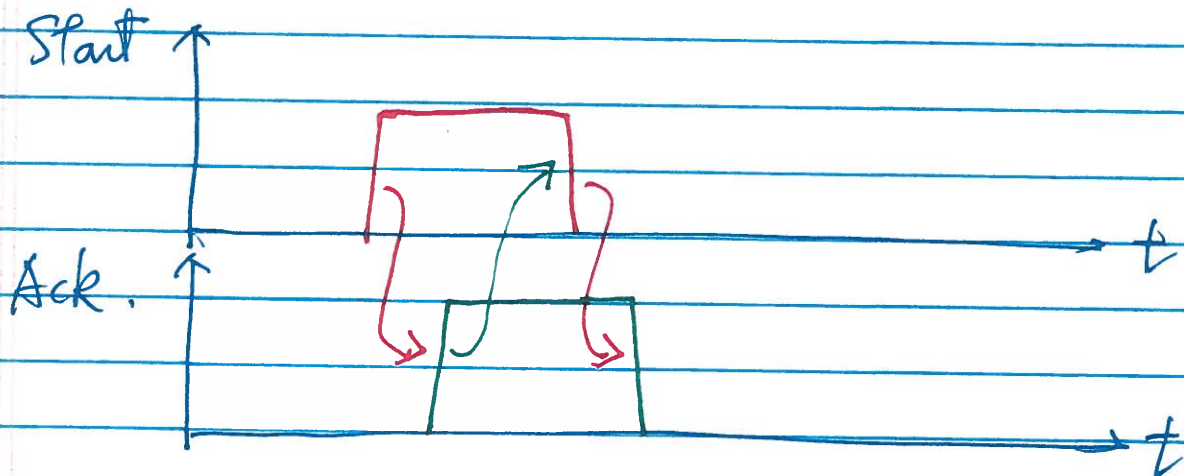
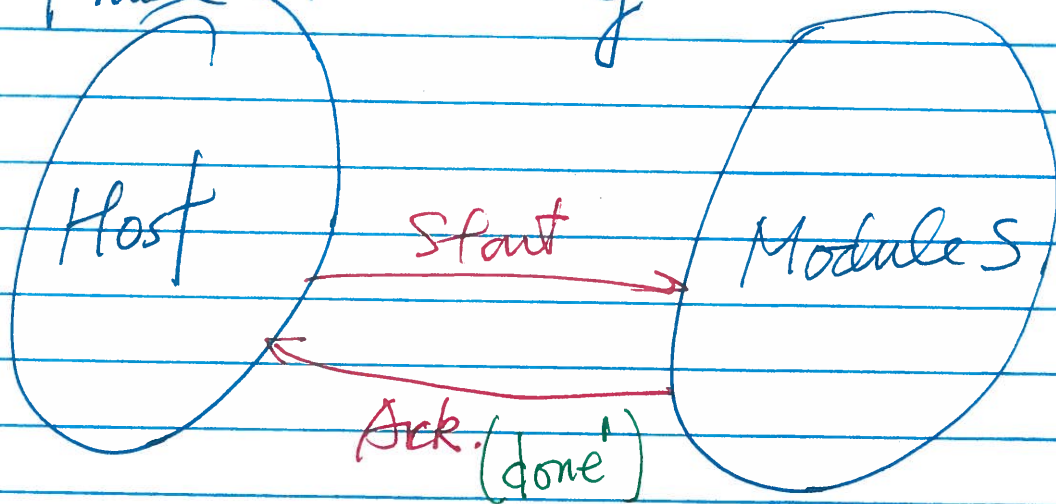
5

Handshaking



6

4-Phase Handshaking



Host Sender wants to start the operation

raises $Start = 1$

Mod S Receiver detects $Start = 1$,

raises $Ack = 1$ $done = 0$

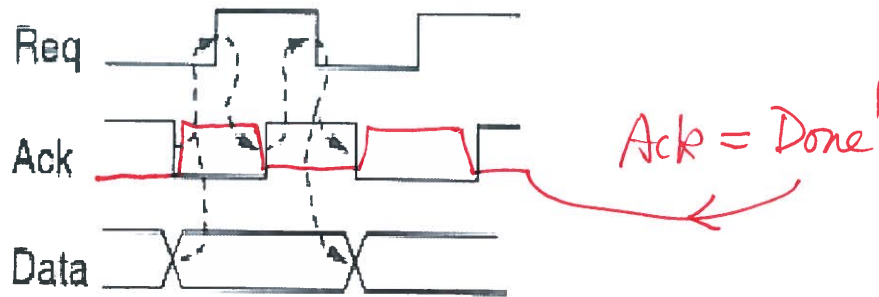
Host Sender detects $Ack = 1$

lowers $Start = 0$

Mod S Receiver detects $Start = 0$

lowers $Ack = 0$

Handshaking



7

Handshaking: Example

The master module that calls module S controls the following signals

- A. Signal **start**
- B. Signal **done**
- C. Signals **start** and **done**
- D. None of the above

8

II. Components

	Components	Functions
Data Subsystem	Storage Modules	Data storage
	Operators <i>ALU, INC, ...</i>	Data operations <i>(Conditions)</i>
	Interconnections	Data transport
	<i>Wires, Decoders, Muxes.</i>	<i>Compare, True, False, Overflow</i>
Control Subsystem	<i>Conditions</i> ↓ Sequential machines	<i>control signals.</i> <i>If, then, else, while, until, ...</i>
		Control of data operations
		Control of data transports
		Control of the sequential system

9

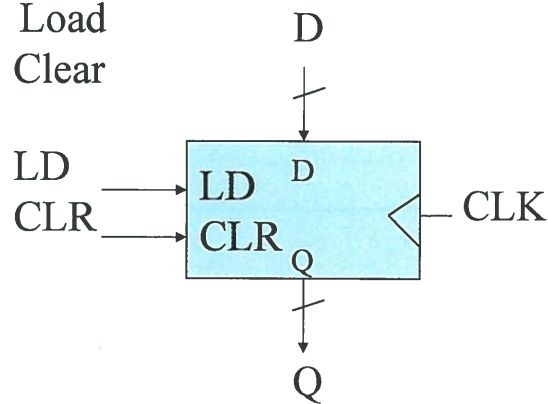
II. Data Subsystem Components

- Storage: Register, RAM, FIFO, LIFO, Counter, Shifter
- Operator: ALU, Floating Point Operators
- Interconnect: Wire, Buses, Crossbars

II. Components: Storage Modules, Register

LD: Load

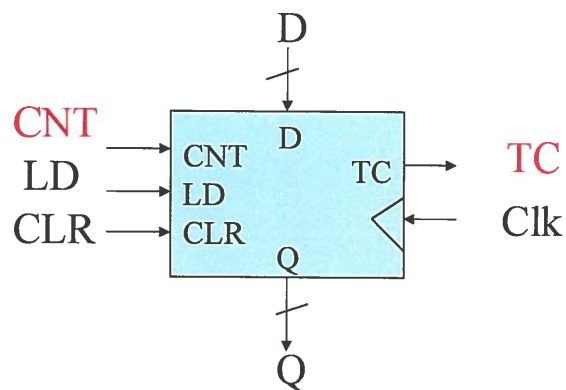
CLR: Clear



$$\begin{aligned}
 Q(t+1) &= (0, 0, \dots, 0) && \text{if } CLR = 1 \\
 &= D && \text{if } LD = 1 \text{ and } CLR = 0 \\
 &= Q(t) && \text{if } LD = 0 \text{ and } CLR = 0
 \end{aligned}$$

11

Modulo-n Counter



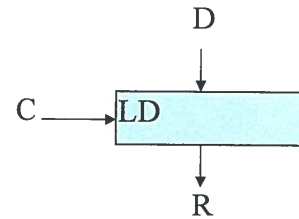
$$\begin{aligned}
 Q(t+1) &= (0, 0, \dots, 0) && \text{if } CLR = 1 \\
 &= D && \text{if } LD = 1 \text{ and } CLR = 0 \\
 &= (Q(t)+1) \bmod n && \text{if } LD = 0, CNT = 1 \text{ and } CLR = 0 \\
 &= Q(t) && \text{if } LD = 0, CNT = 0 \text{ and } CLR = 0
 \end{aligned}$$

$$\begin{aligned}
 TC &= 1 && \text{if } Q(t) = n-1 \text{ and } CNT = 1 \\
 &= 0 && \text{otherwise}
 \end{aligned}$$

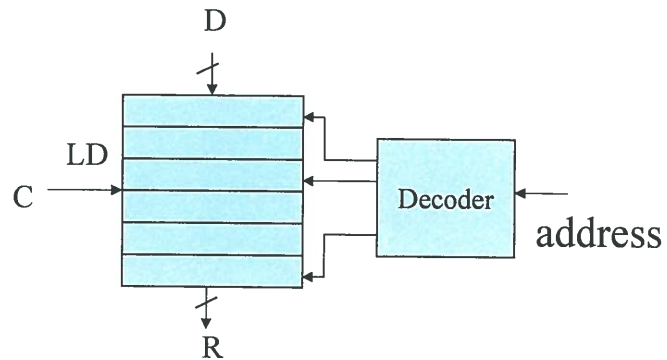
12

Storage Component: Registers, Array of Registers

Registers: If C then $R \leftarrow D$



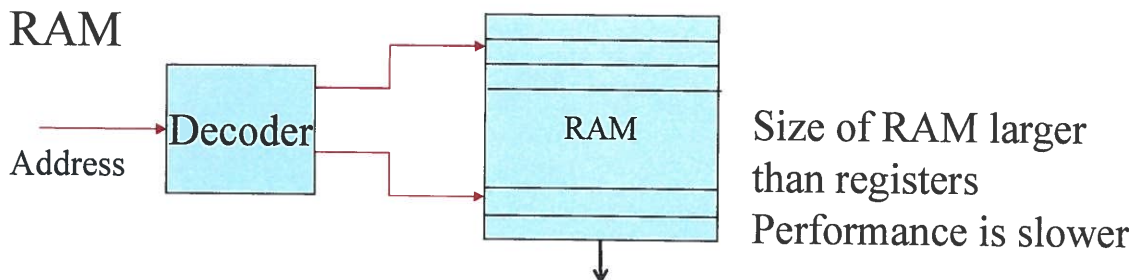
Register Array: If C then $R_{\text{address}} \leftarrow D$
Sharing connections and controls



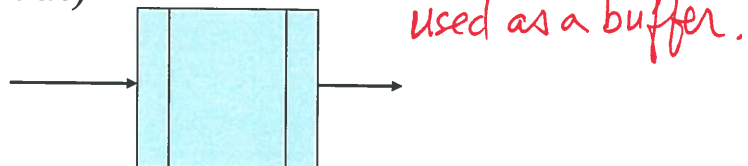
13

Storage Components: RAM, FIFO, LIFO

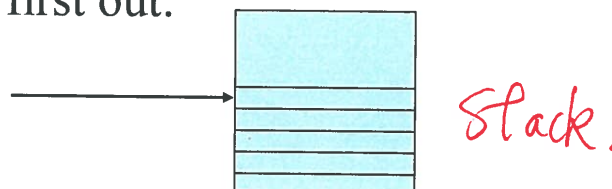
RAM



FIFO (First in first out)

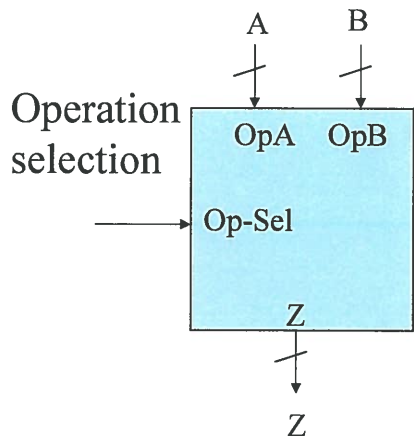


LIFO (Last in first out:
Stack)



14

Functional Modules



```

CASE Op-Sel Is
  When F1, Z <= A op1 B
  When F2, Z <= A op2 B
  .
  .
End CASE
  
```

Example:

```

CASE Op-Sel Is
  Z <= (A + B)mod 2n if Op-Sel=addition,
  Z <= (A - B)mod 2n if Op-Sel=subtraction
End CASE
  
```

15

Interconnect Modules (Wires and Switches)

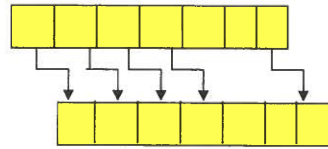
- Single Lines
- Band of Wires
- Shared Buses
- Crossbar

1. Single line (shifting, time sharing)

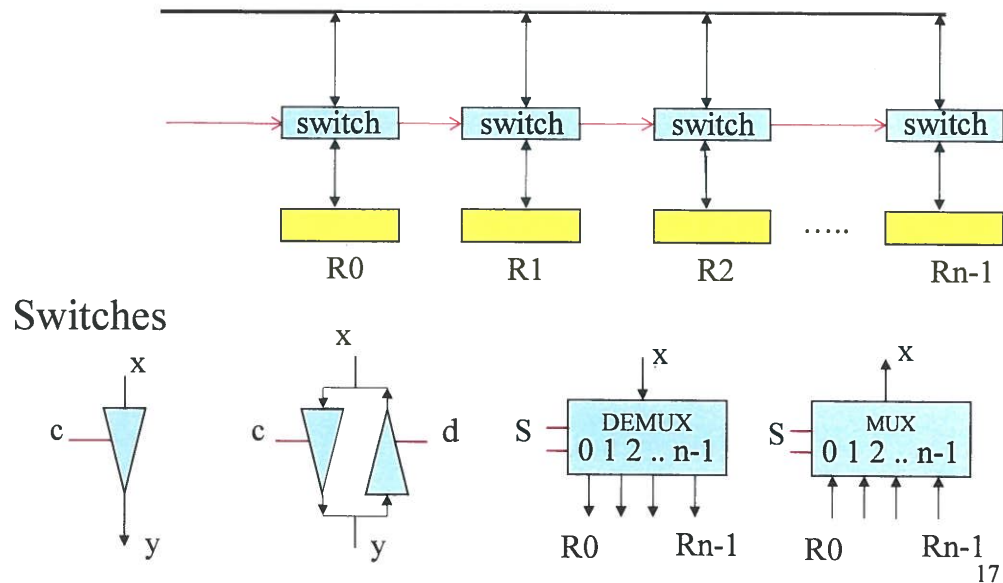


16

2. Band of Wires (BUS)

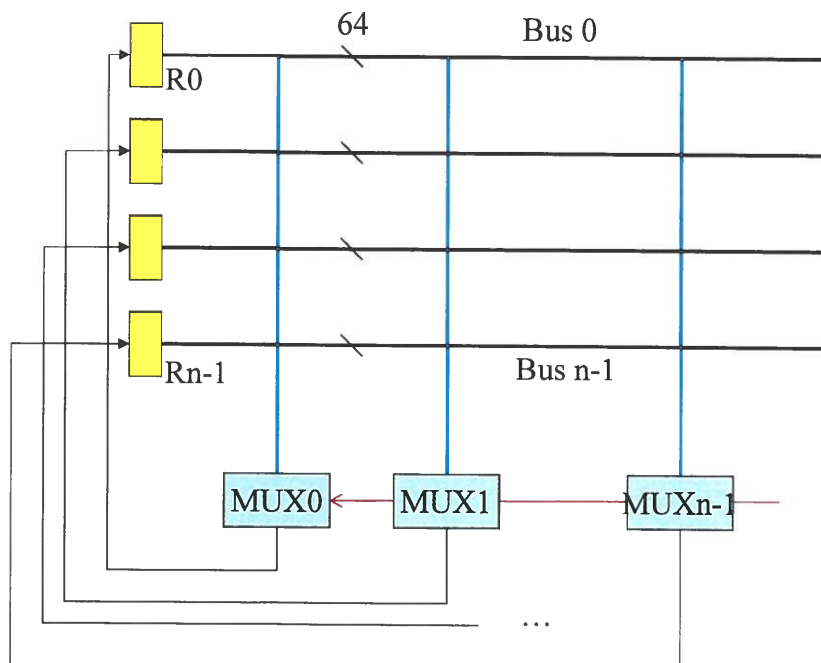


3. Shared Bus

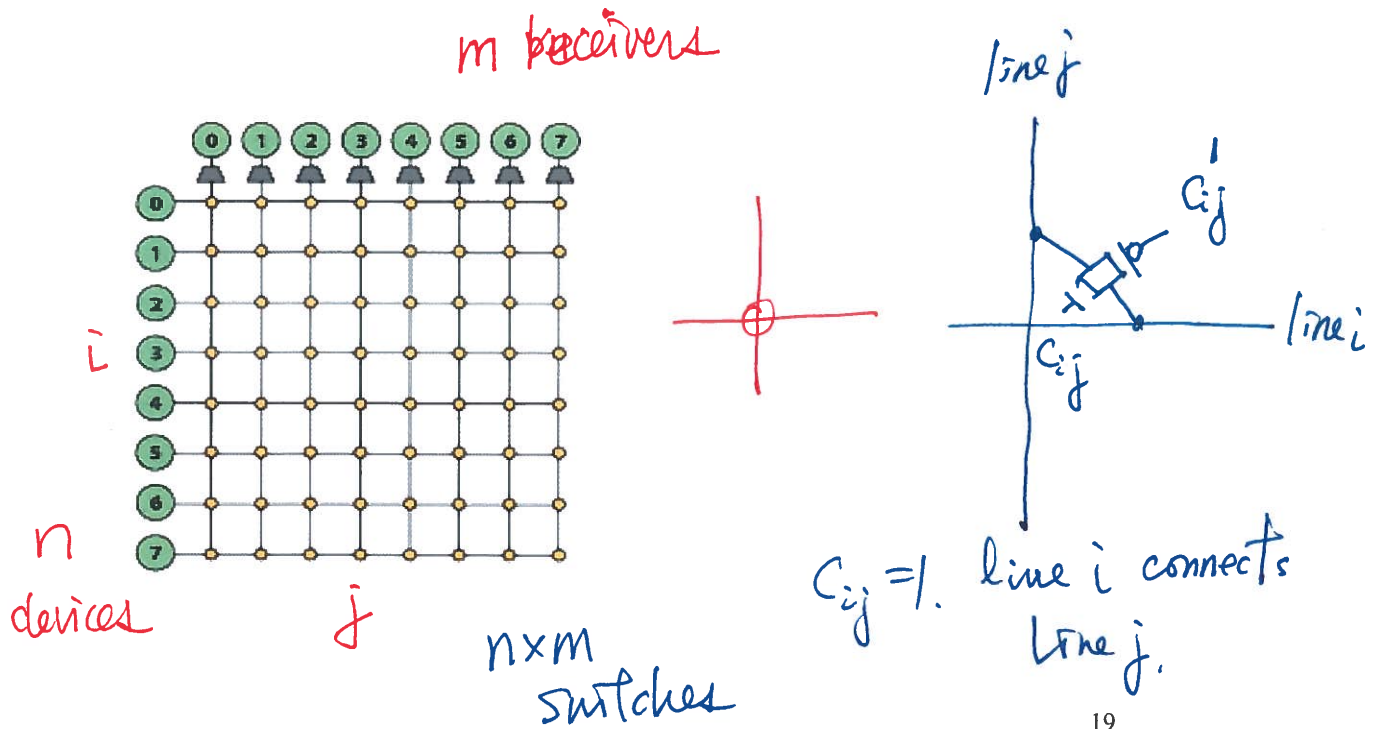


4. Crossbar (Multiple buses running horizontally)

m simultaneous transfers are possible, but more expensive.

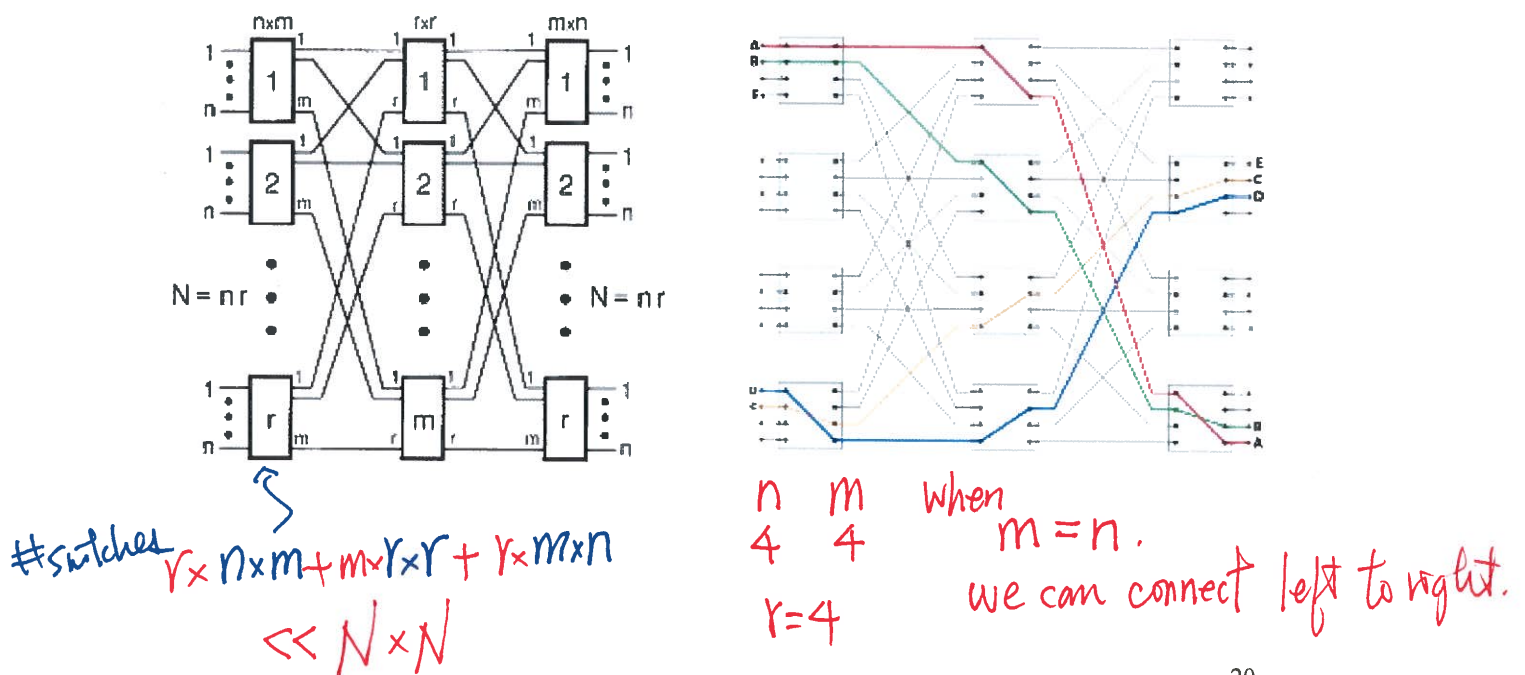


Crossbar Switches: VLSI Design



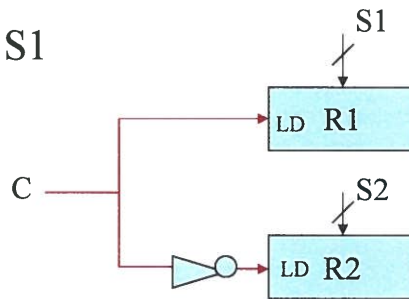
19

Multistage Network: Clos Network

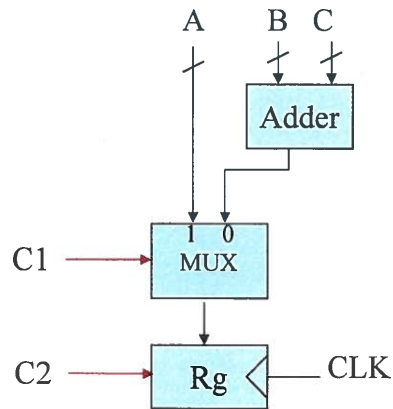


20

Ex. If C then $R1 \leftarrow S1$
 Else $R2 \leftarrow S2$
 Endif;



If C1 then $X \leftarrow A$
 Else $X \leftarrow B + C$
 Endif
 If C2 then $Rg \leftarrow X$
 Endif



23

System Designs

- Introduction
 - Methodology and Framework
- Components
- Specification
- Implementation

24

VI. Implementation

- Example
- Handshaking
 - Request and Acknowledgement
- Datapath Subsystem
 - Data Operators
 - Data Transporters
- Control Subsystem
 - One Hot Machine Design

25

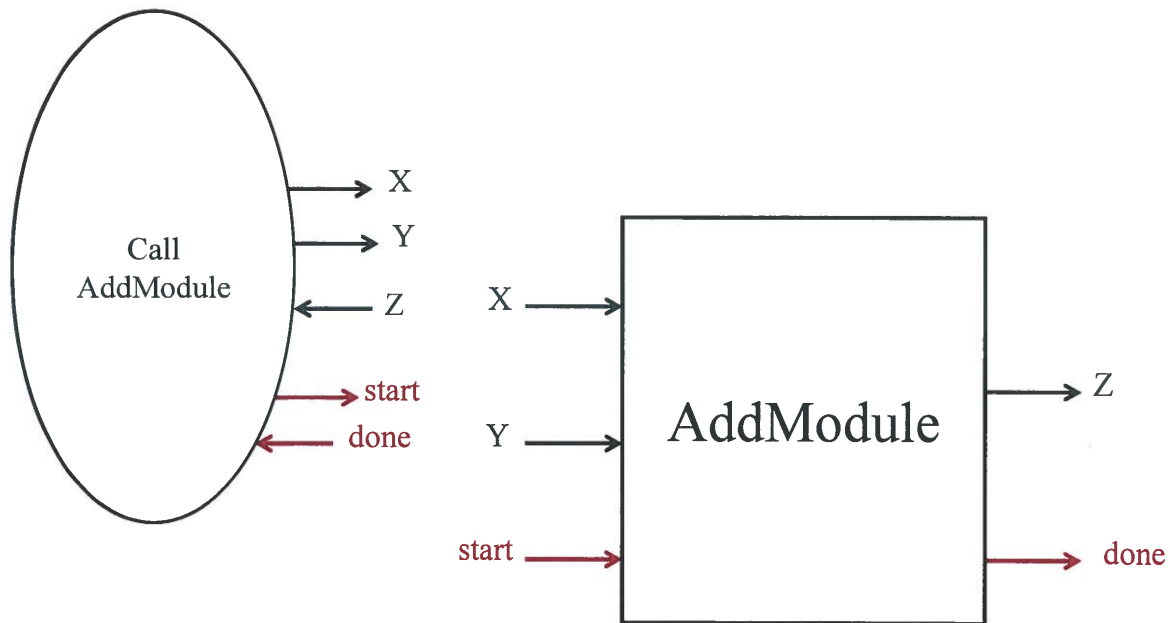
VI. Implementation: Example

```
AddModule(X, Y, Z, start, done)
{ Input X[15:0], Y[15:0] type bit-vector,
  start type Boolean;
  Local-Object A[15:0], B[15:0] type bit-vector;
  Output Z[15:0] type bit-vector,
  done type Boolean;
S0: If start' goto S0 || done ← 1;
S1: A ← X || B ← Y || done ← 0;
S2: Z ← Add(A, B) || goto S0;
}
```

Exercise: Go through the handshaking, data subsystem and control subsystem designs.

26

AddModule(X,Y,start,done)



27

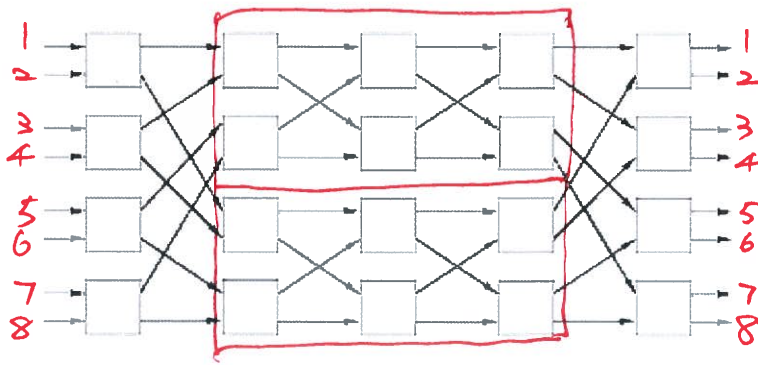
Example

Suppose that each step (S_i) takes one clock cycle.
How many clock cycles does the AddModule take to complete one handshaking iteration?

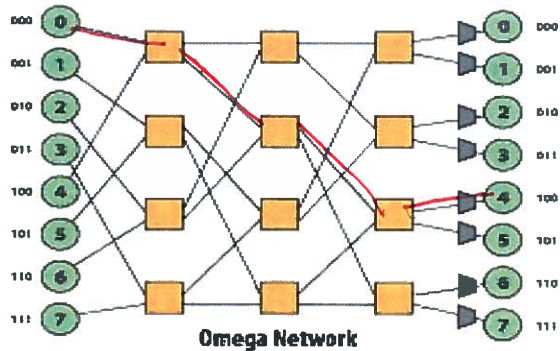
- A. One cycle
- B. Two cycles
- C. More than two cycles

28

Multistage Networks: Benes and Omega Networks



*rearrangeable
non-blocking*



Blocking: Previous assignment can block the next route.

*0 → 4
4 → 5 is blocked.²¹*

III. Specification: Program

1. Objects (Registers, Outputs of combinational logic)
2. Operation (Logic, Add, Multiplication, DSP, and etc.)
3. Assignment
4. Sequencing

Example:

Signal S1, S2, R[15:0]: FFs, Registers, wires
 $Z \leftarrow A + B$: Registers, Adder, Interconnect
 $R1 \leftarrow R2$: Registers and Interconnect
 Begin, End: Control
 if () then (), ENDIF: Control