

CSE 140 Lecture 16

System Designs

CK Cheng
CSE Dept.
UC San Diego

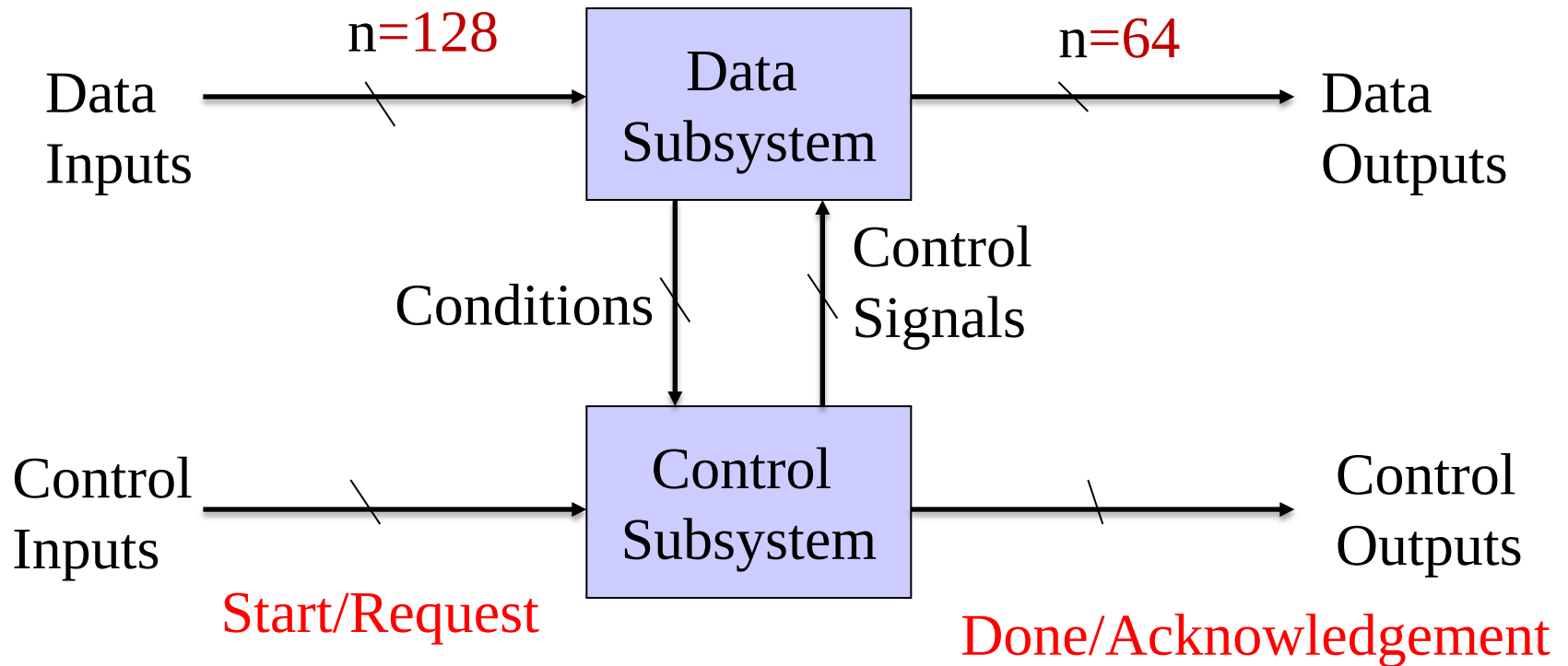
System Designs

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

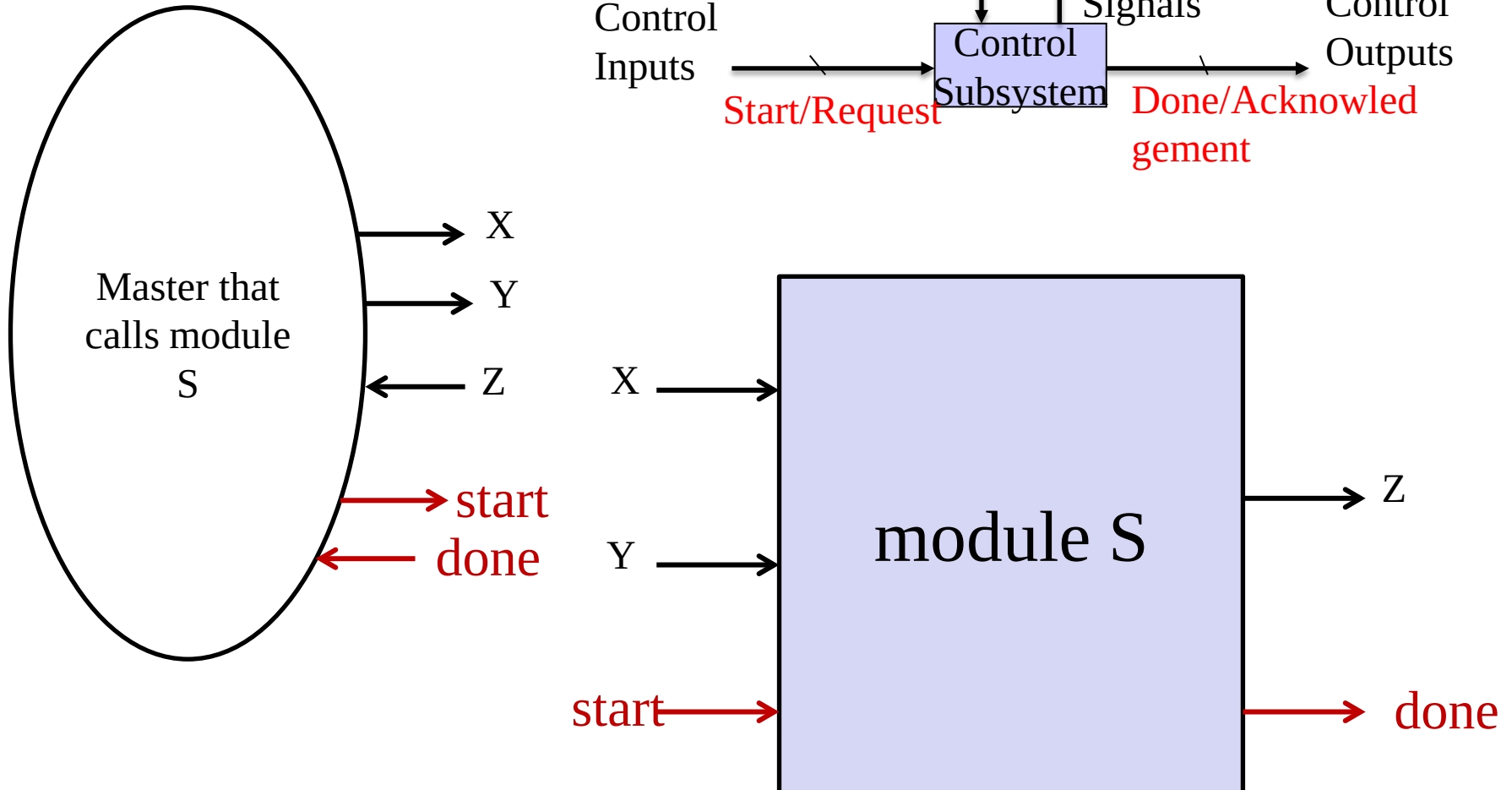
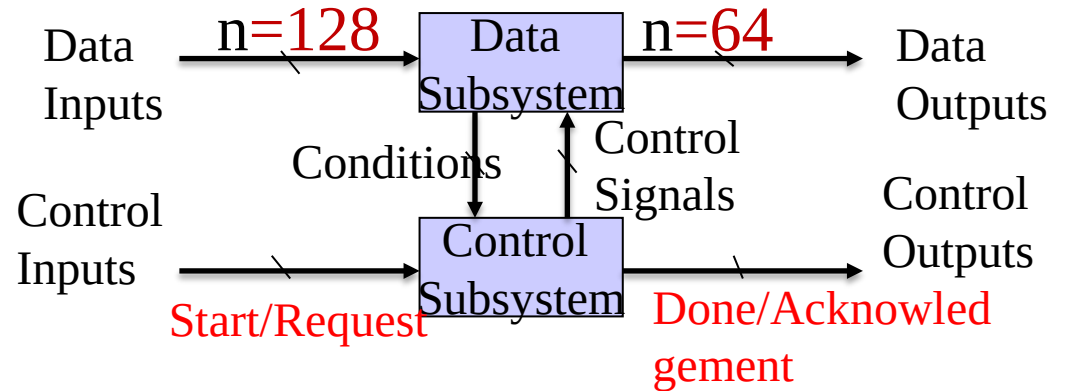
Introduction

- 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.

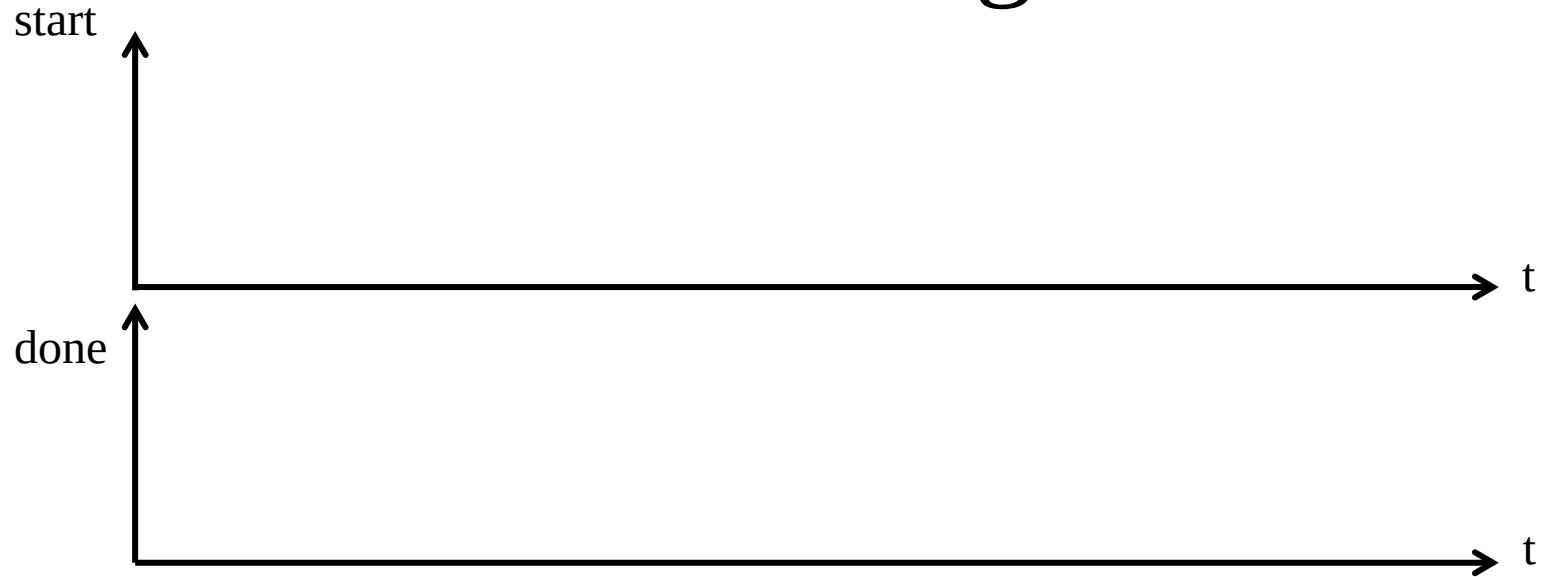
I. Introduction: Framework



I. Introduction: Handshaking



Handshaking



Handshaking: iClicker

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

II. Components

	Components	Functions
Data Subsystem	Storage Modules Operators Interconnections	Data storage Data operations Data transport
Control Subsystem	Sequential machines	Control of data operations Control of data transports Control of the sequential system

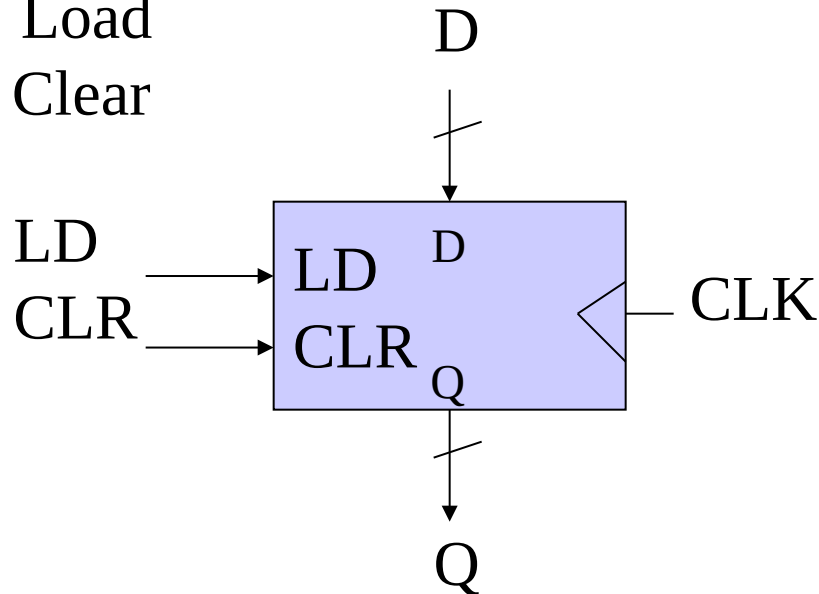
II. Data Subsystem Components

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

II. Components: Storage Modules, Register

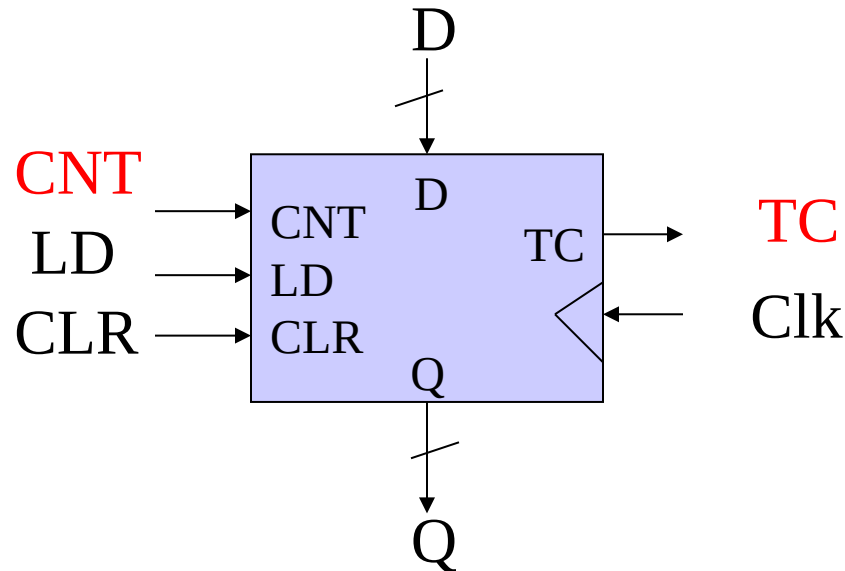
LD: Load

CLR: Clear



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

Modulo-n Counter



$$\begin{aligned}
 Q(t+1) &= (0, 0, \dots, 0) \\
 &= D \\
 &= (Q(t)+1) \bmod n \\
 &= Q(t)
 \end{aligned}$$

if CLR = 1

if LD = 1 and CLR = 0

if LD = 0, CNT = 1 and CLR = 0

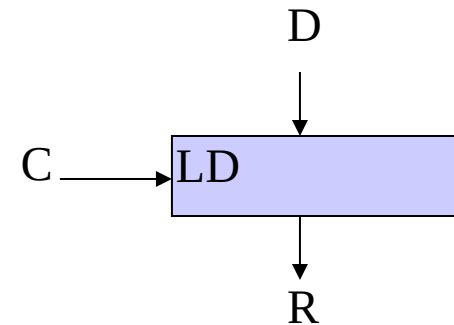
if LD = 0, CNT = 0 and CLR = 0

$$\begin{aligned}
 TC &= 1 \\
 &= 0
 \end{aligned}$$

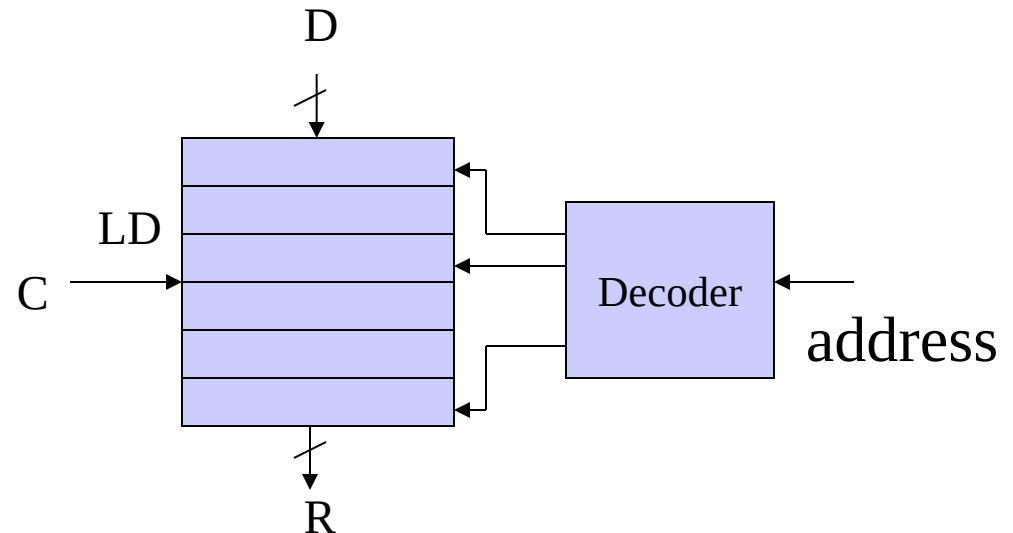
if $Q(t) = n-1$ and CNT = 1
otherwise

Storage Component: Registers, Array of Registers

Registers: If C then $R \leftarrow D$

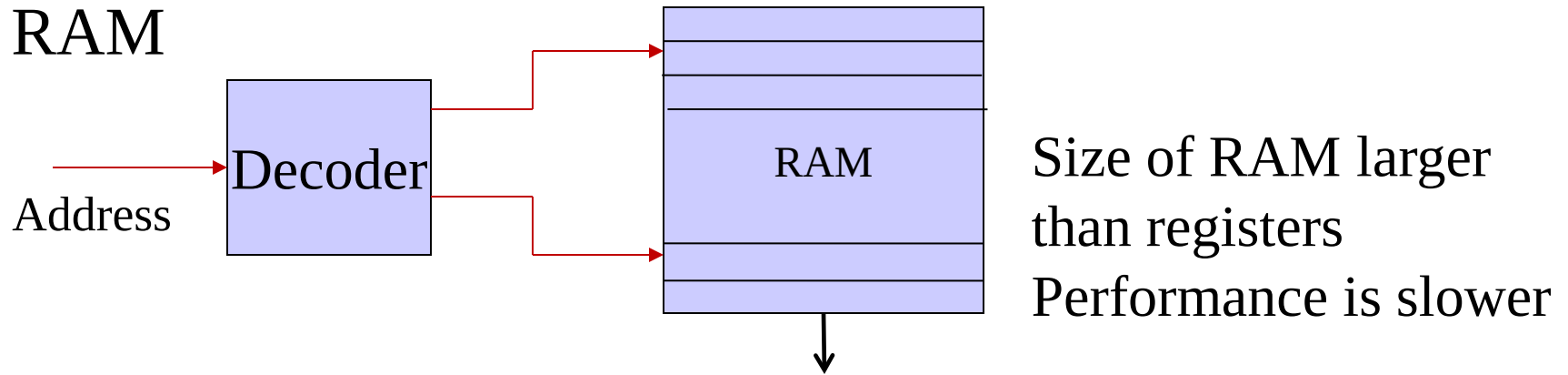


Register Array: If C then $R_{\text{address}} \leftarrow D$
Sharing connections and controls

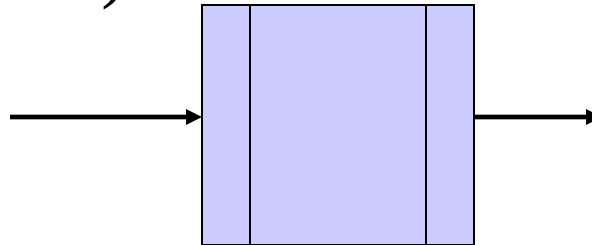


Storage Components: RAM, FIFO, LIFO

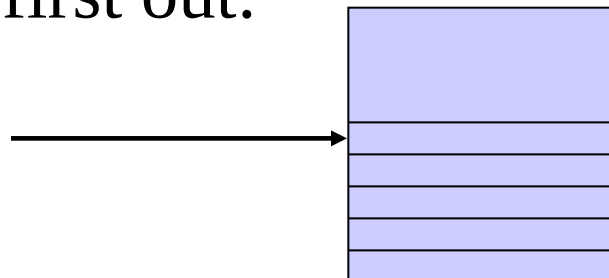
RAM



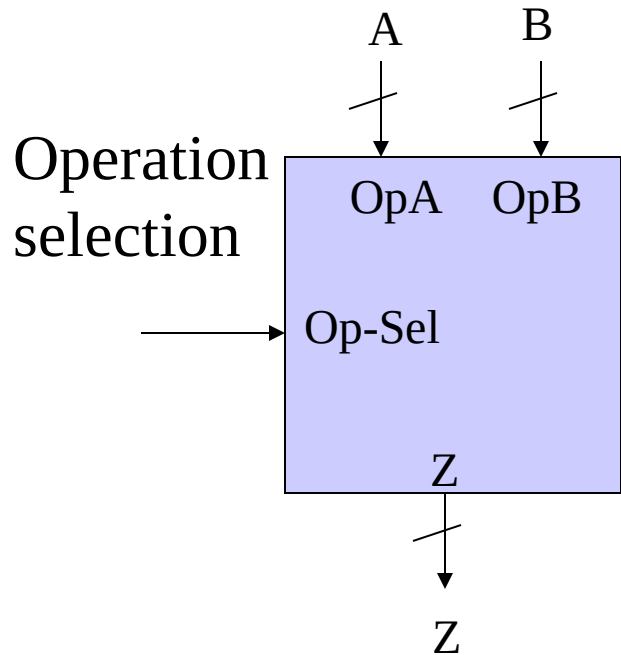
FIFO (First in first out)



LIFO (Last in first out:
Stack)



Functional Modules



CASE Op-Sel Is

When F1, $Z \leq A \text{ op1 } B$

When F2, $Z \leq A \text{ op2 } B$

.

.

End CASE

Example:

CASE Op-Sel Is

$Z \leq (A + B) \bmod 2^n$ if Op-Sel=addition,

$Z \leq (A - B) \bmod 2^n$ if Op-Sel=subtraction

End CASE

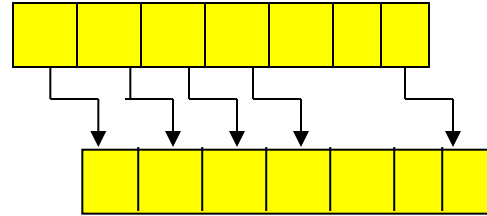
Interconnect Modules (Wires and Switches)

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

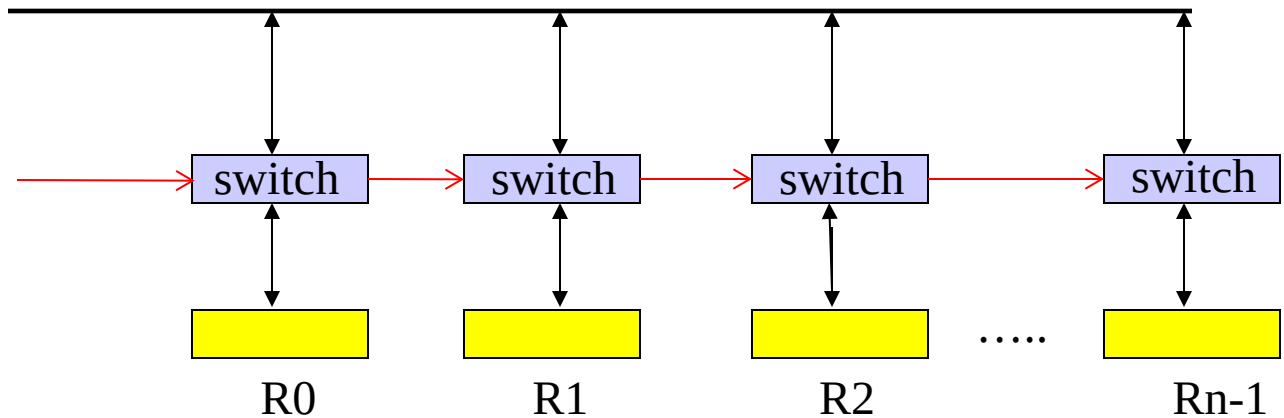
1. Single line (shifting, time sharing)



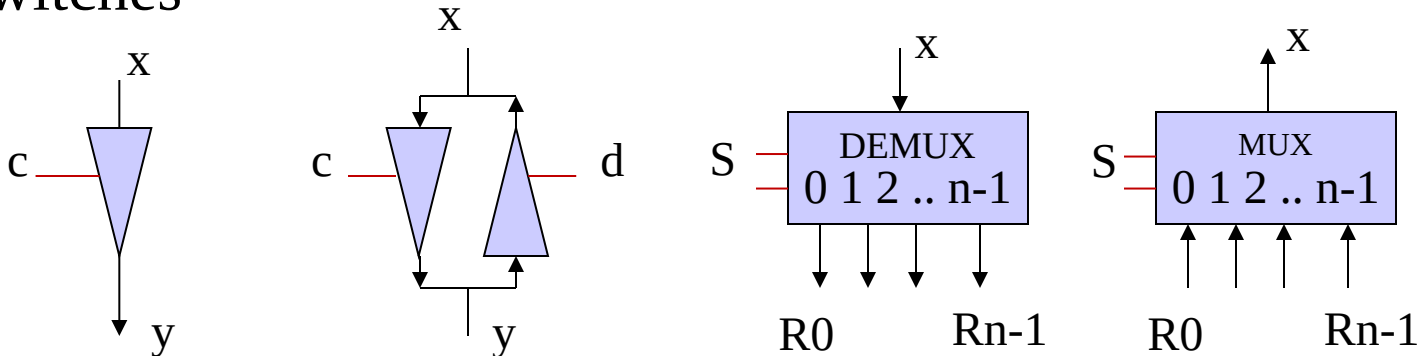
2. Band of Wires (BUS)



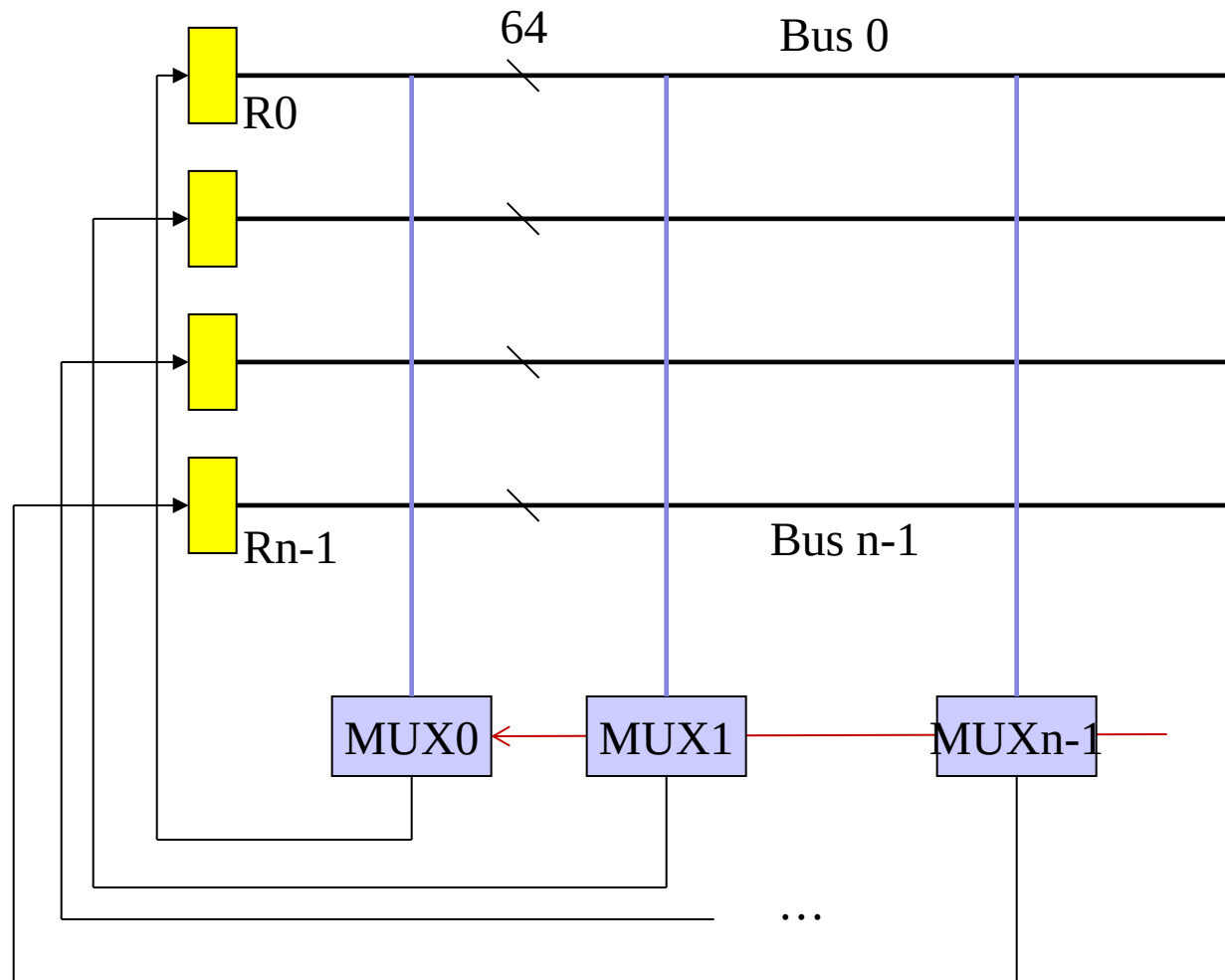
3. Shared Bus



Switches



4. Crossbar (Multiple buses running horizontally)
m simultaneous transfers are possible, but more expensive.



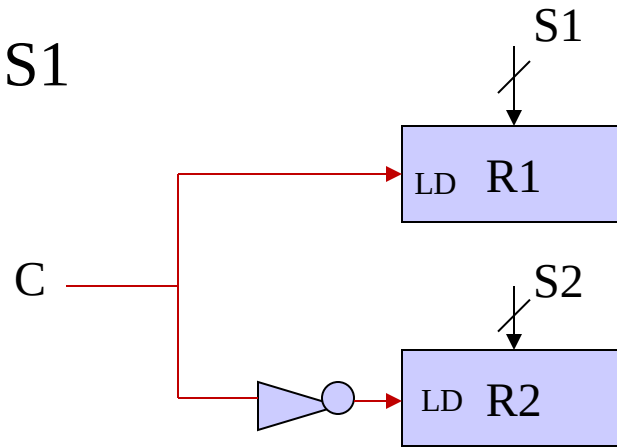
III. Specification: Program

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

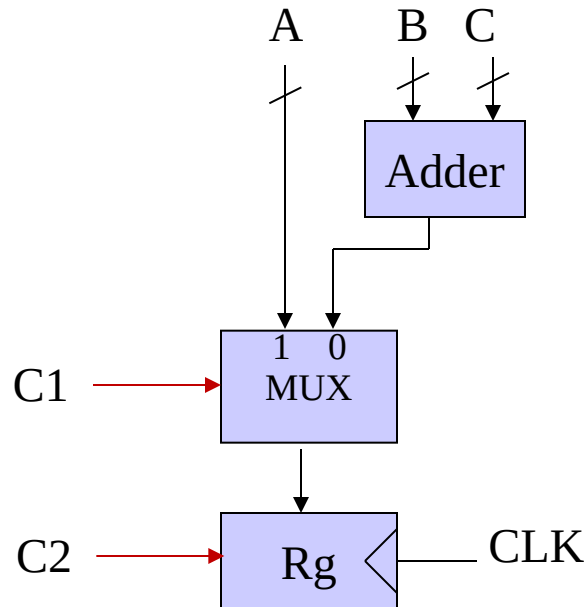
Example:

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

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



VI. Implementation

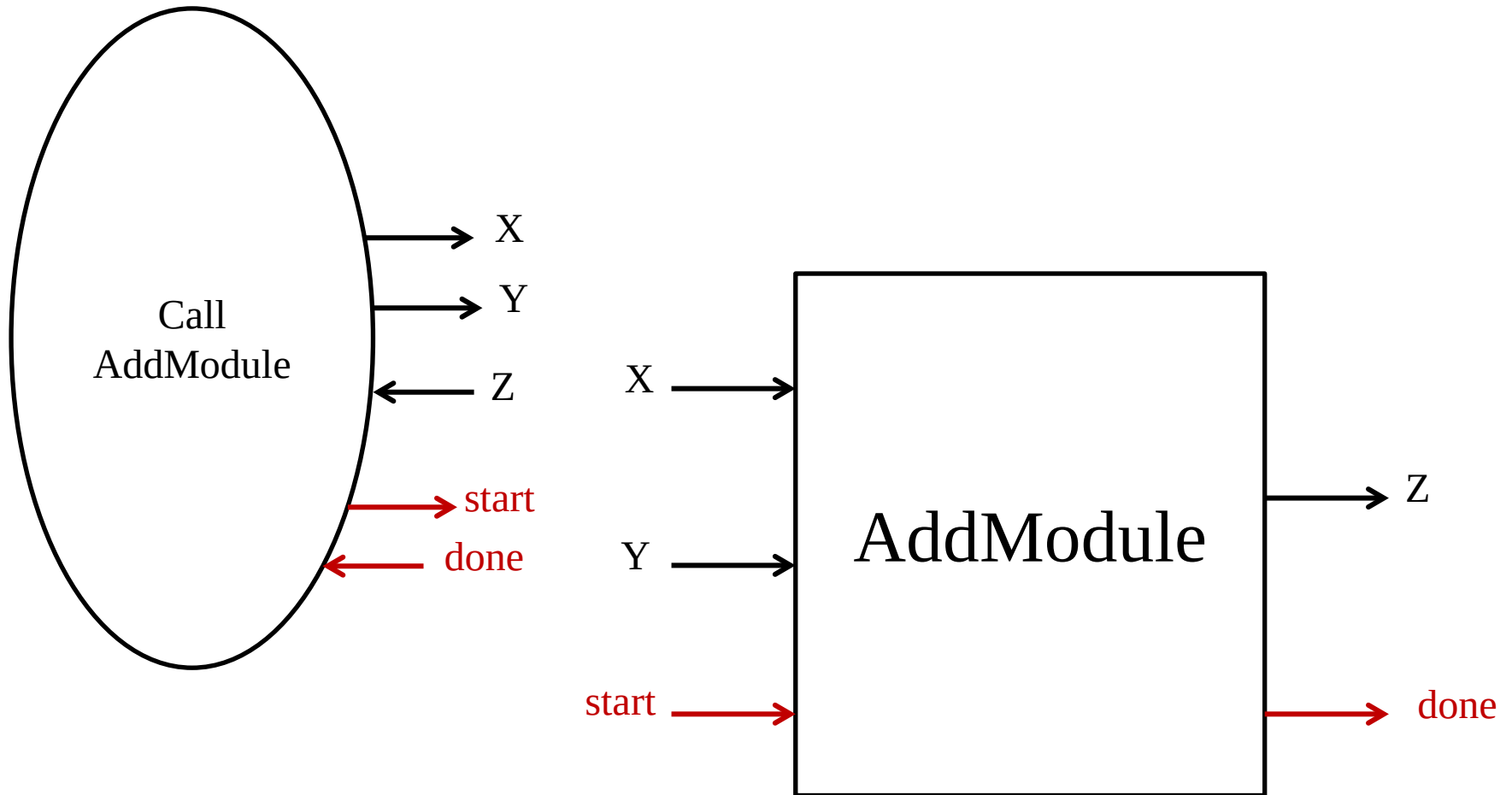
- Example
- Handshaking
 - Request and Acknowledgement
- Datapath Subsystem
 - Data Operators
 - Data Transporters
- Control Subsystem
 - One Hot Machine Design

VI. Implementation: Example

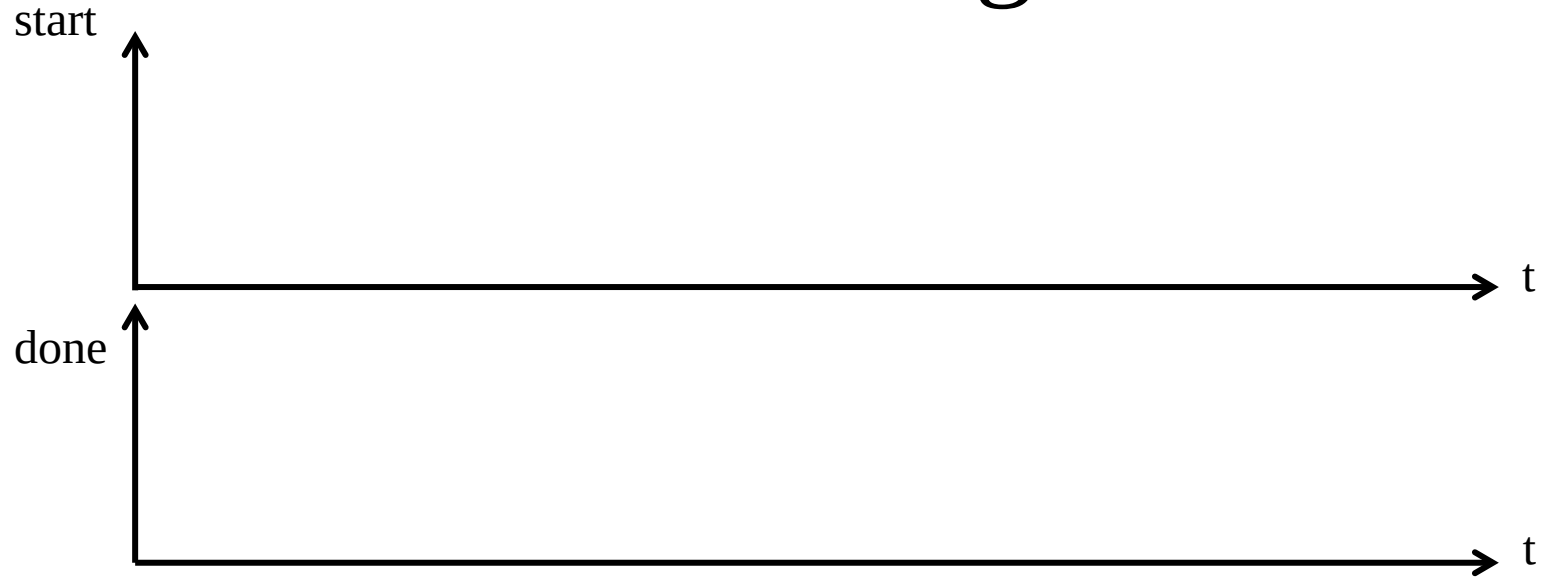
```
AddModule(X, Y, Z, start, done)
{ Input X[15:0], Y[15:0] type bit-vector,
  start type Boolean;
  Local-Object A[15:0], B[15:0] type bit-vector;
  Output Z[15:0] type bit-vector,
  done type Boolean;
S0: If start' goto S0 || done  $\leftarrow$  1;
S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || done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

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

AddModule(X,Y,**start,done**)



Handshaking



iClicker

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