CSE 140 Lecture 14
System Designs

CK Cheng

CSE Dept.

UC San Diego

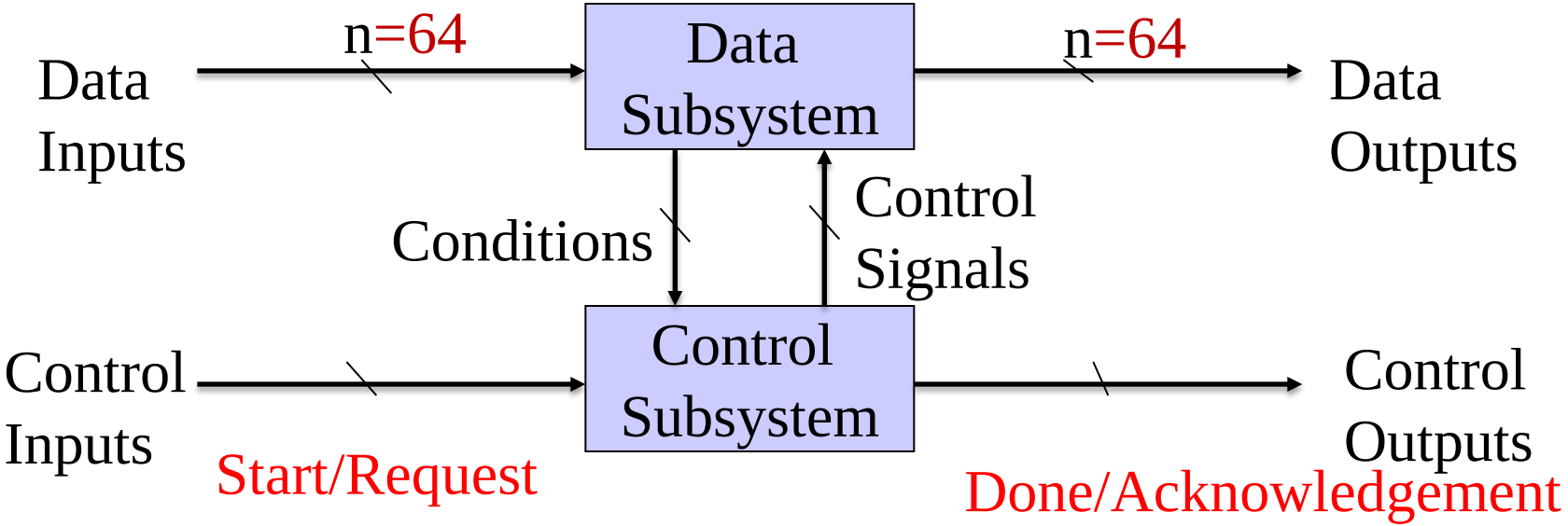
System Designs

- Introduction
- Components
- Spec
- Implementation

Introduction

- Methodology
  - Approach with success stories.
  - Hierarchical designs with interface between the levels.
- 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



Introduction

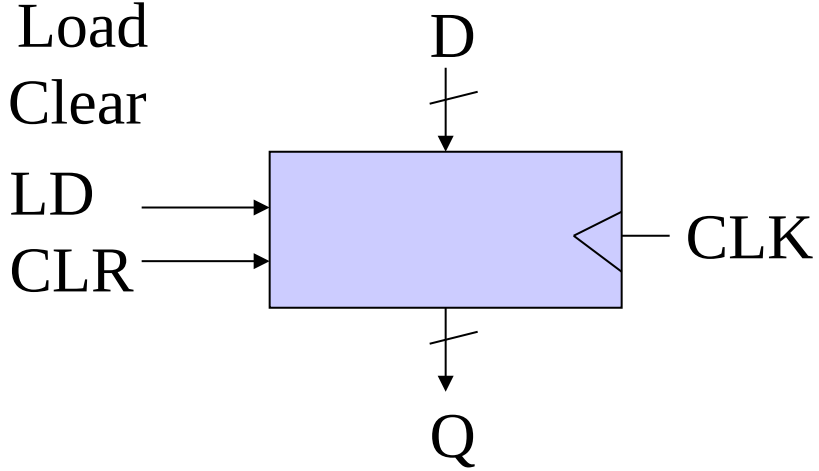
	Components	Functions
Data Subsystem	Storage Modules	Data storage
	Operators	Data operations
	Interconnections	Data transport
Control Subsystem	Sequential machines	Control of data operations
		Control of data transports
		Control of the sequential system

Data Subsystem Components

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

Components: Storage Modules, Register

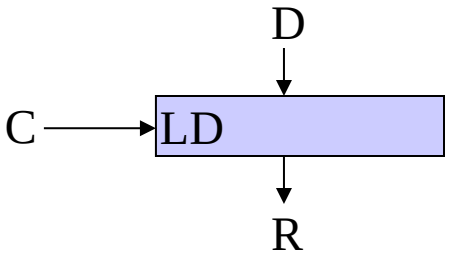
LD: Load
CLR: Clear



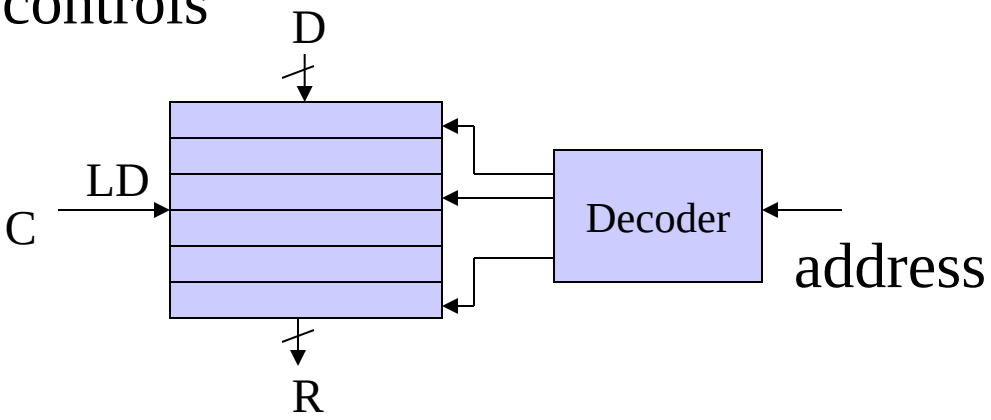
$$
\begin{aligned}
Q(t+1) &= (0, 0, \ldots, 0) && \text{if CLR = 1} \\
&= D && \text{if LD = 1 and CLR = 0} \\
&= Q(t) && \text{if LD = 0 and CLR = 0}
\end{aligned}
$$

Storage Component: Registers, Array of Registers

Registers: If C then R ← D

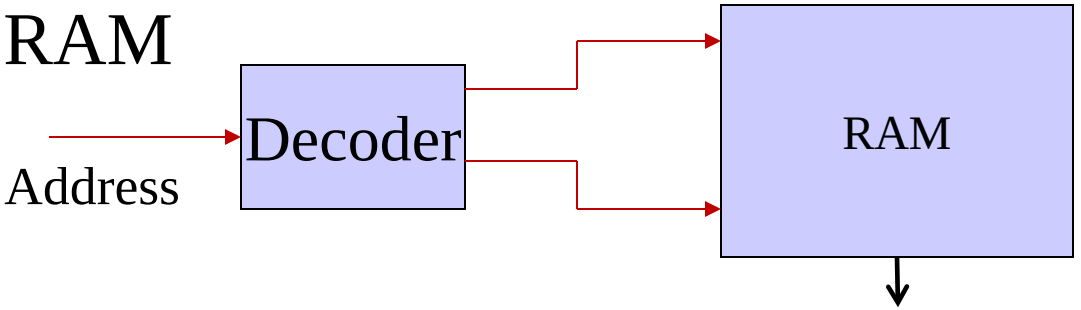


Register Array: If C then $R_{address}$ ← D
Sharing connections and controls



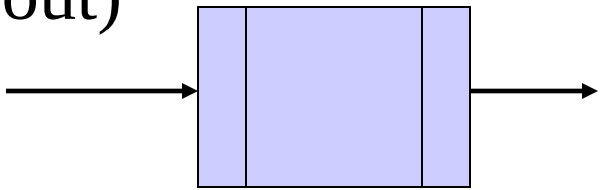
Storage Components: RAM, FIFO, LIFO

RAM

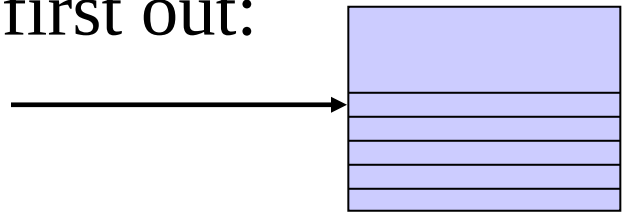


Size of RAM larger than registers
Performance is slower

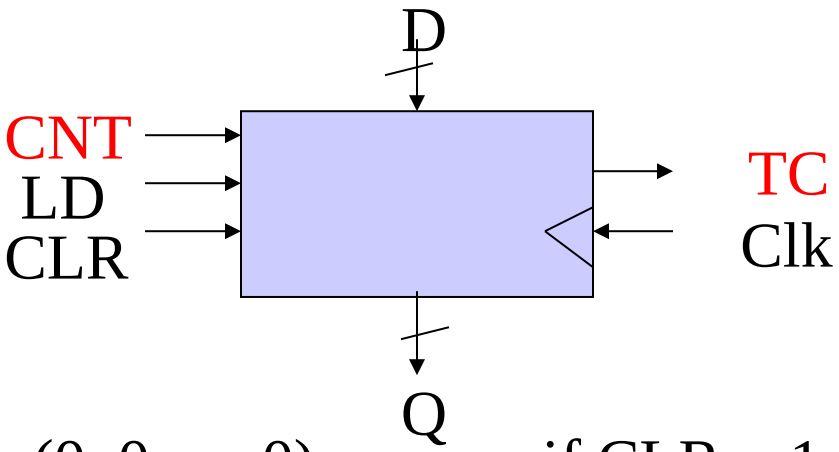
FIFO (First in first out)



LIFO (Last in first out: Stack)



Modulo-n Counter



$Q(t+1) = (0, 0, \ldots, 0)$ if CLR = 1

$= D$ if LD = 1 and CLR = 0

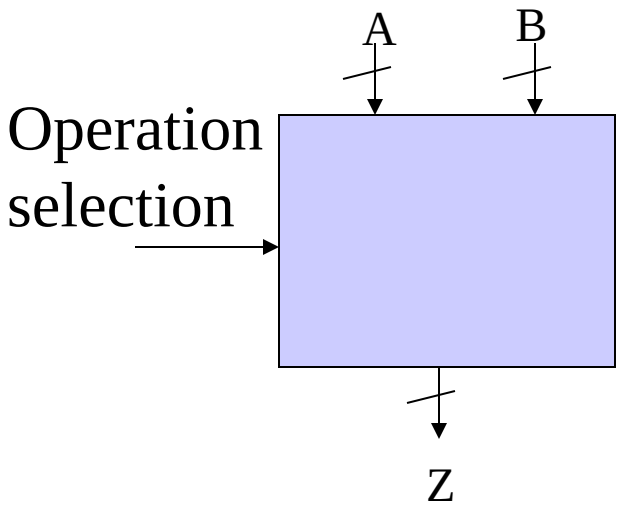
$= (Q(t)+1) \bmod n$ if LD = 0, CNT = 1 and CLR = 0

$= Q(t)$ if LD = 0, CNT = 0 and CLR = 0

TC = 1 if Q (t) = n-1 and CNT = 1

= 0 otherwise

Functional Modules




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

Example:

CASE Op-Set Is

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

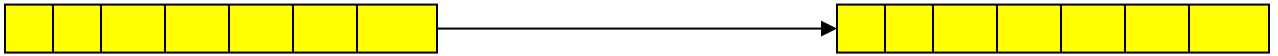
$Z <= (A - B) \bmod 2^n$ if Op-Sel=subtraction

End CASE

Interconnect Modules (Wires and Switches)

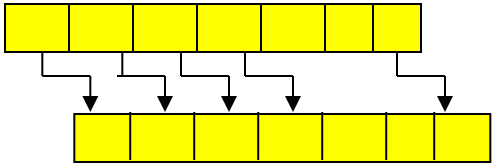
- Single Lines
- Band of Wires
- Shared Buses
- Crossbar

1. Single line (shifting, time sharing)

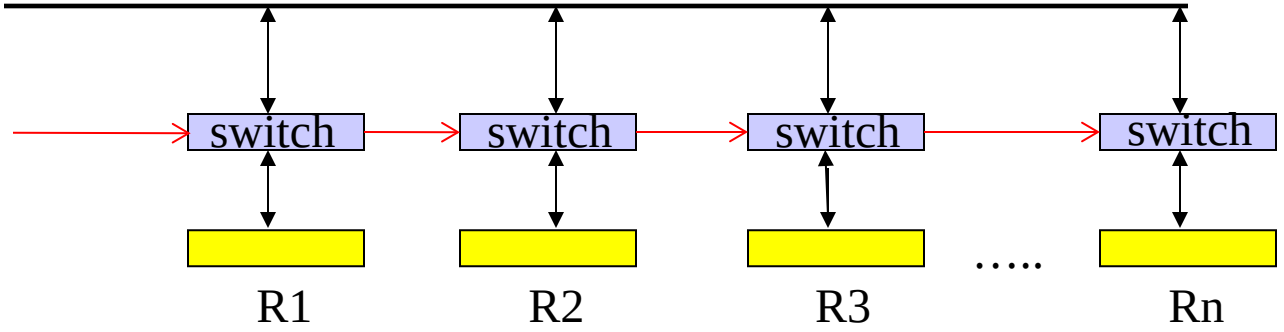


Shift Register Shift Register

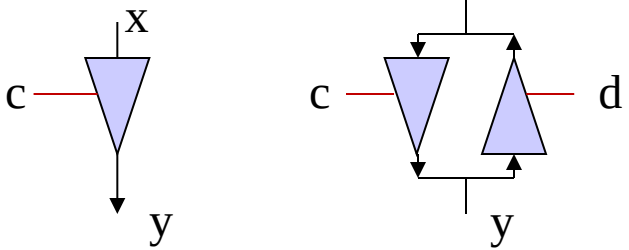
2. Band of Wires (BUS)

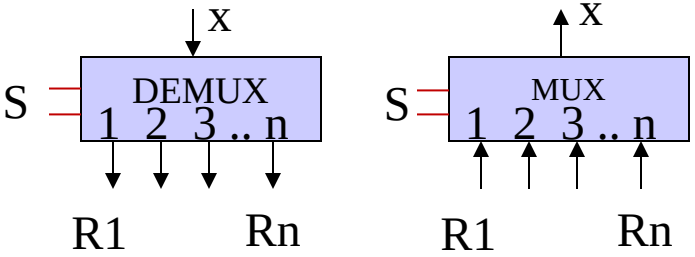


3. Shared Bus



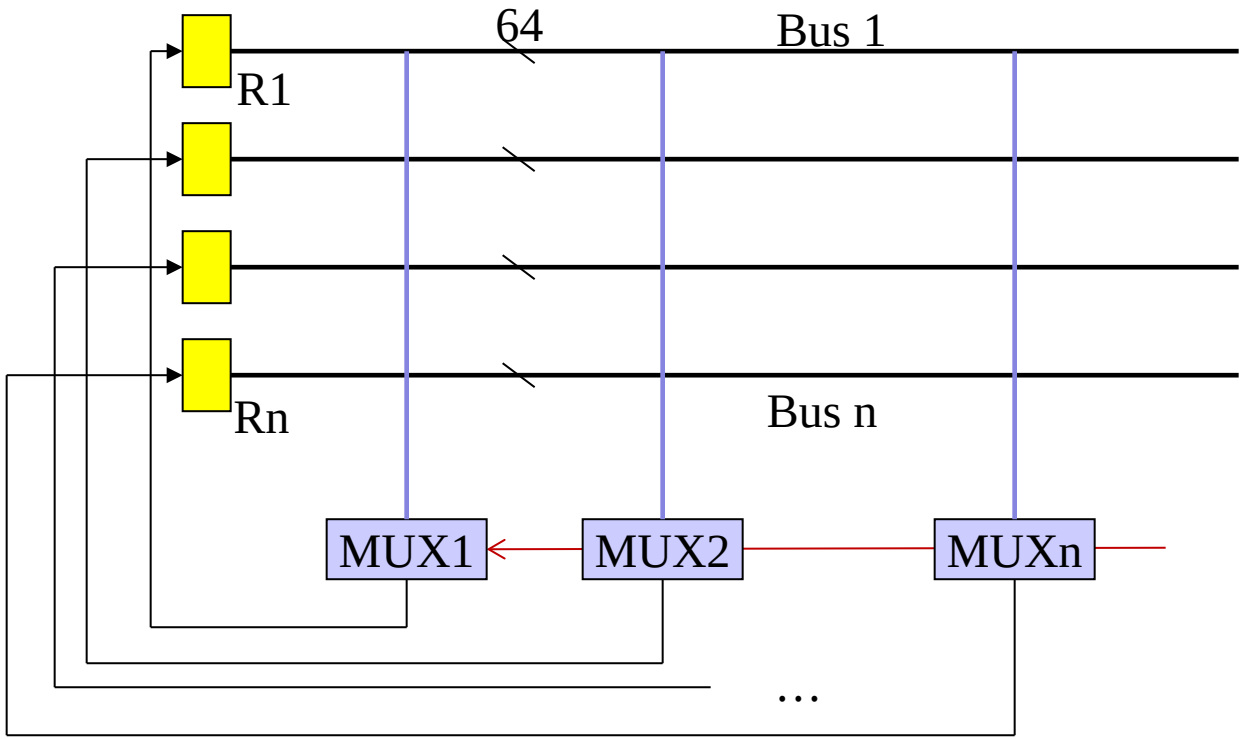
Switches





4. Crossbar (Multiple buses running horizontally)

m simultaneous transfers are possible, but more expensive.



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

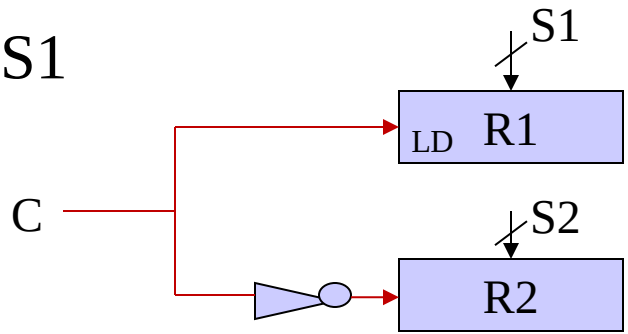
Z ← A + B: Registers, Adder, Interconnect

R1 ← R2: Registers and Interconnect

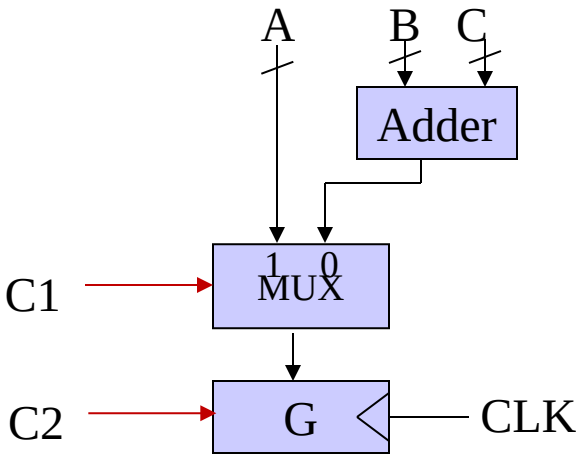
Begin, End: Control

if ( ) then ( ), ENDIF: Control

Ex. If C then R1 ← S1
Else R2 ← S2
Endif;



If C1 then X ← A
Else X ← B + C
Endif
If C2 then G ← X
Endif



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;

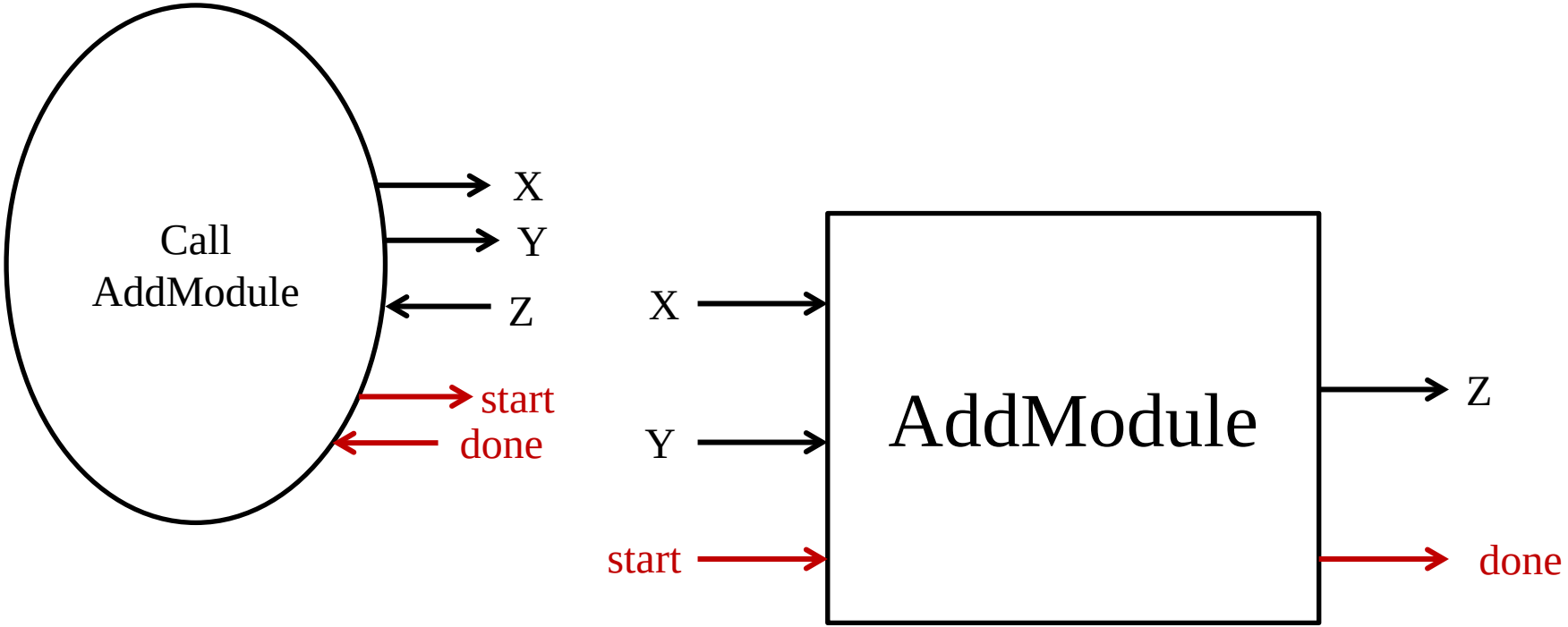
S1: A ← X || B ← Y || done ←0;

S2: Z ← Add(A, B) || goto S0;

}

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

AddModule(X,Y,start,done)



Hand Shaking