CSE 140 Lecture 13
System Designs

CK Cheng

CSE Dept.

UC San Diego

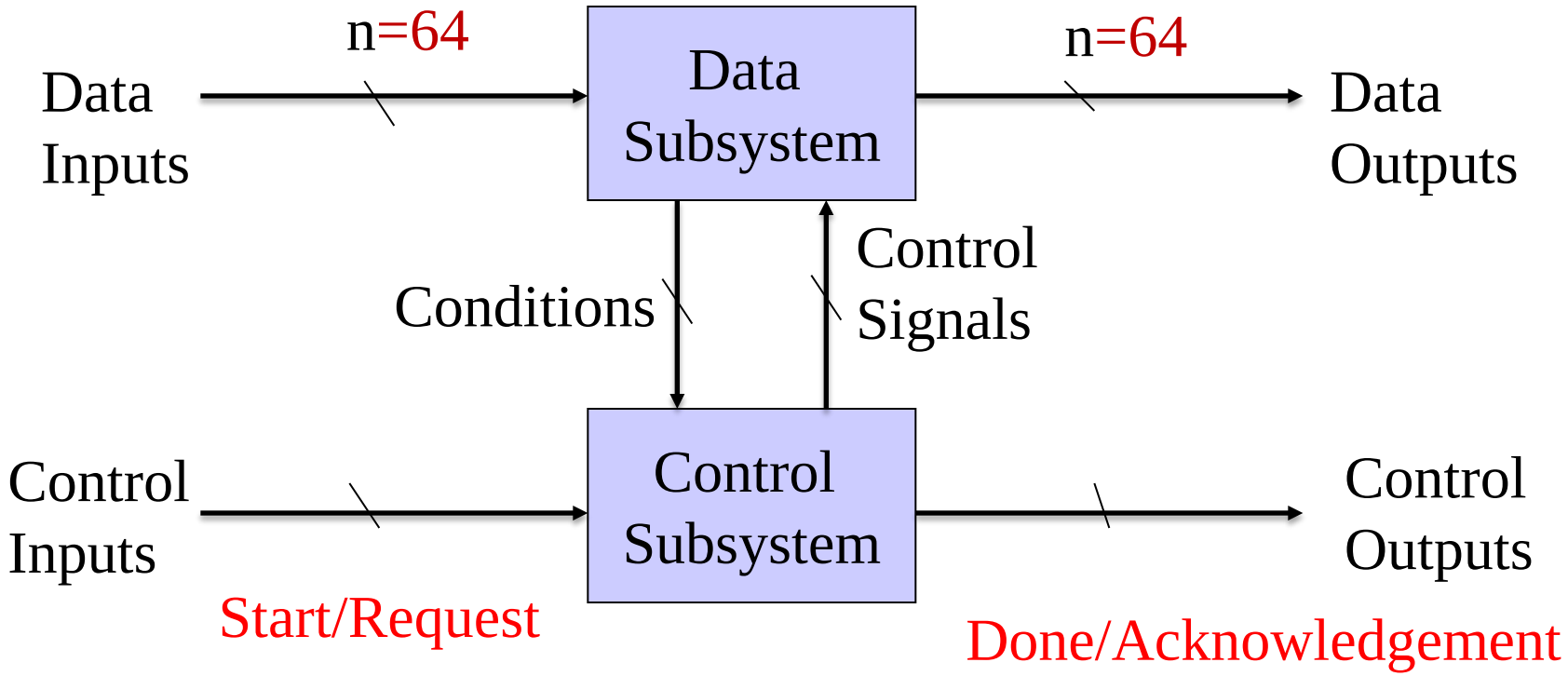
System Designs

- 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



II. Components

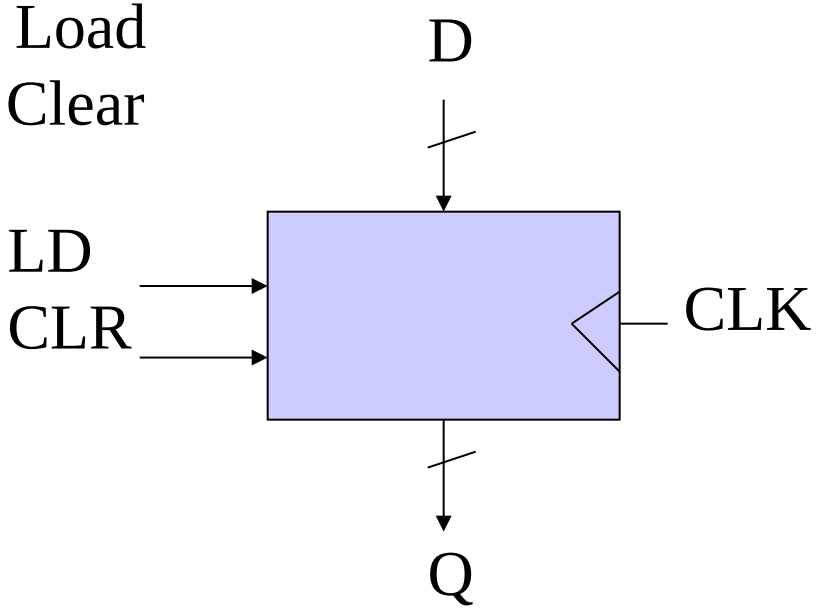
	Components	**Functions**
Data Subsystem	Storage Modules	Data storage
	Operators	Data operations
	Interconnections	Data transport
Control Subsystem	Sequential machines	Control of data operations
		Control of data transports
		Control of the sequential system

II. Data Subsystem Components

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

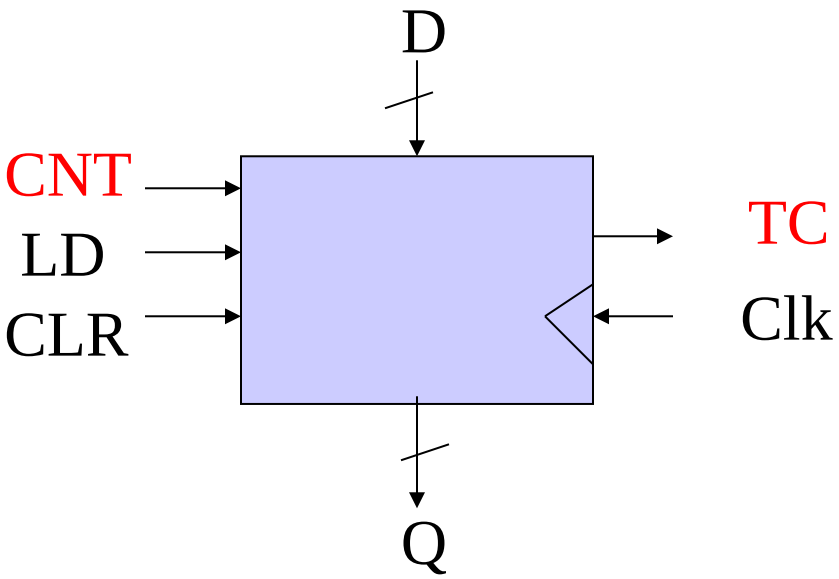
II. Components: Storage Modules, Register

LD: Load
CLR: Clear



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

Modulo-n Counter



Q (t+1) = (0, 0, .. , 0)		if CLR = 1
	= D	if LD = 1 and CLR = 0
	= (Q(t)+1)mod 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

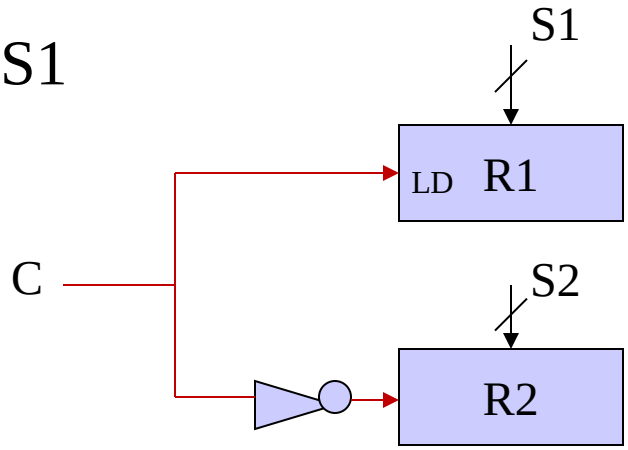
III. Specification: Program

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

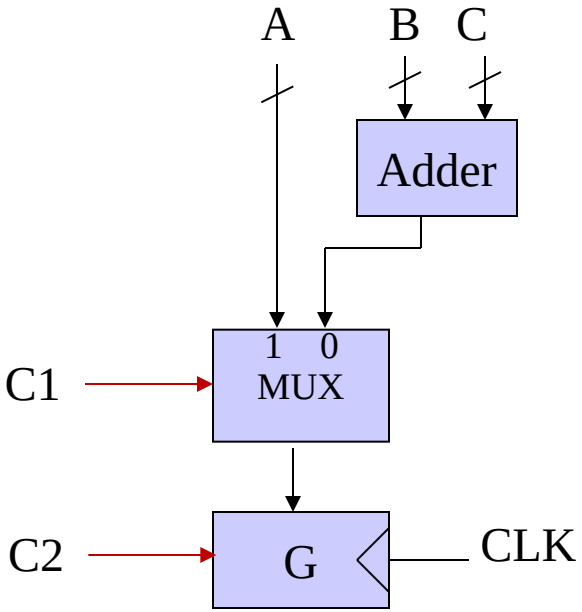
Example:

Signal S1, S2, R[15:0]: FFs, Registers, wires
Z ← 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



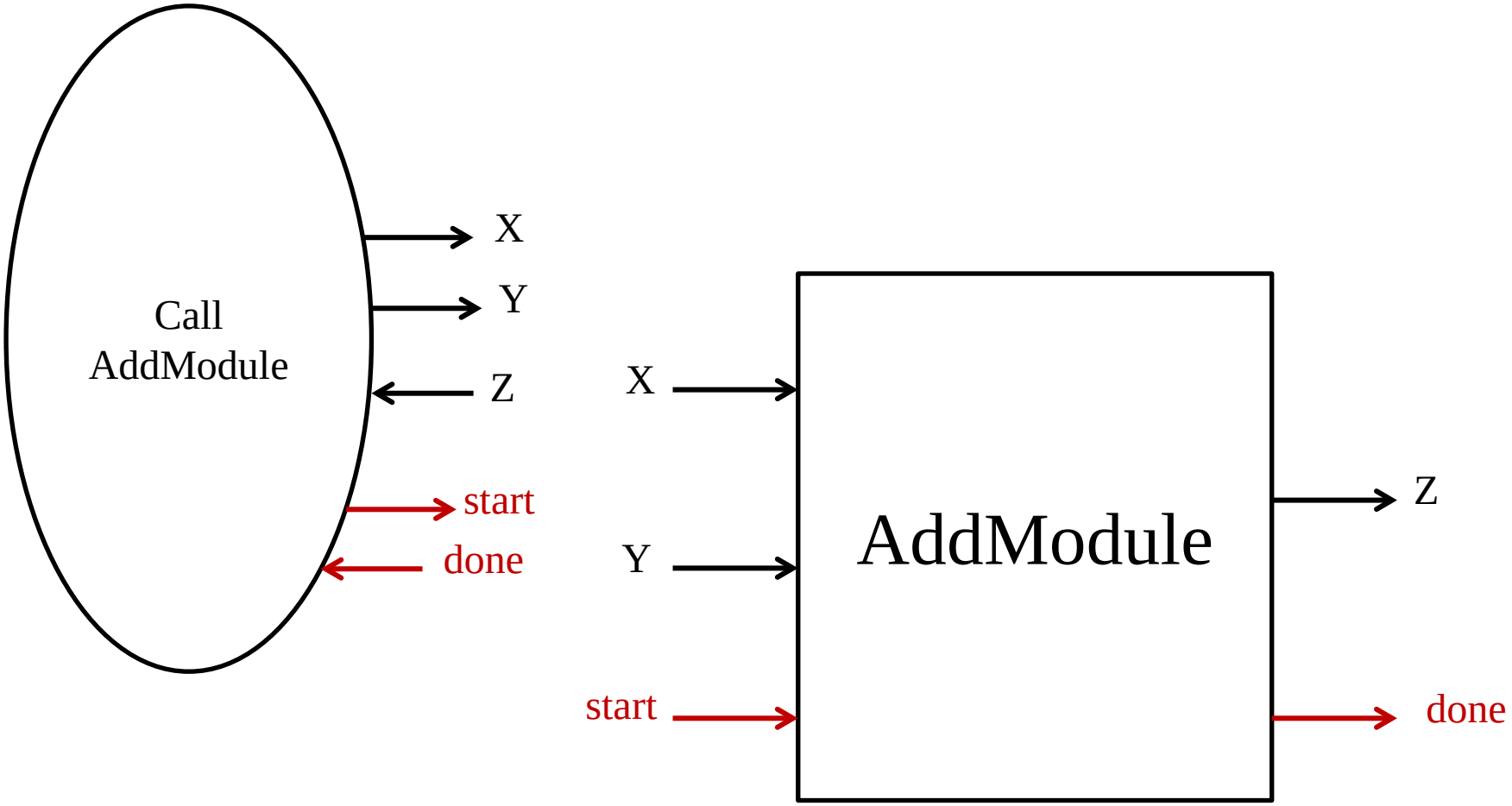
VI. Implementation

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

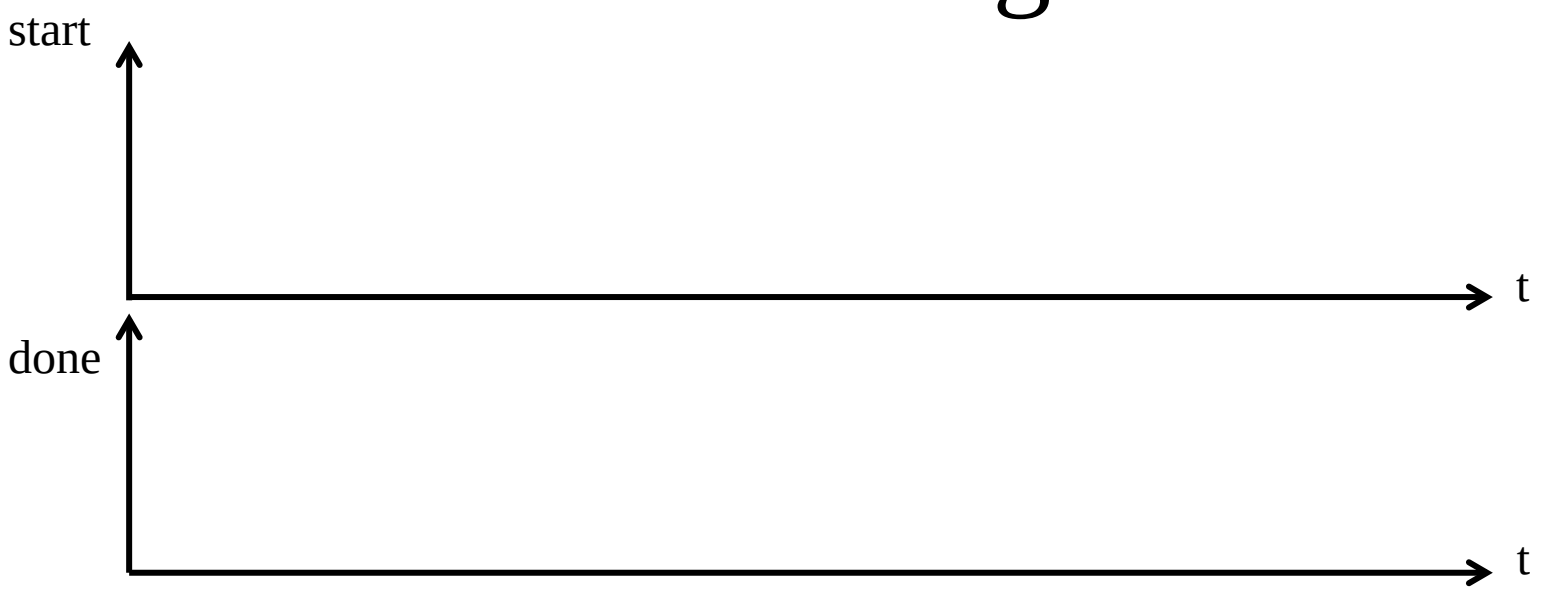
VI. Implementation: Example

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 ← 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



iClicker

How many clock cycles does the AddModule take to complete one handshaking iteration?

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

Logistics: Grading

Grade on style, completeness and correctness

- zyBook exercises: 20%
- iClicker: 9% (by participation up to three quarters of classes)
- Homework: 15% (grade based on a subset of problems. If more than 85% of class fill out CAPE evaluations, the lowest homework score will be dropped)
- Midterm 1: 27% (T 5/2/17)
- Midterm 2: 28% (Th 6/8/17)
- Final: 1% (take home exam, due 10PM, Th 6/15/17)
- Grading: The best of the following
  - The absolute: A- >90% ; B- >80% of total 100% score
  - The curve: (A+,A,A-) top 33+ε% of class; (B+,B,B-) second 33+ε%
  - The bottom: C- above 45% of absolute score.

Logistics

Grade on style, completeness and correctness

- zyBook exercises: 20% (Half of the points will be added for exercises that missed the due date but are completed by 6/11)
- iClicker: 9% (the points are prorated and saturate up to three quarters of classes)
- HW5 solution will posted asap. Thus, there will be no time for late submission.
- New office hrs for W10 will be posted on the web.
- No discussion session for W10.
- Review session (note for Sunday 6/4, session starts after 2PM)
  - A. Saturday 6/3
  - B. Sunday 6/4