

System Designs

CSE 140 Lecture 16 System Designs

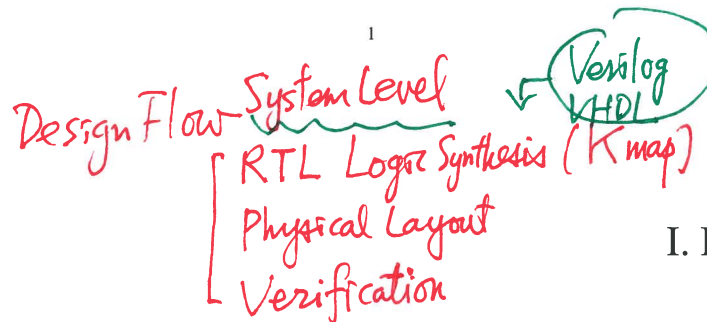
Methodology

CK Cheng
CSE Dept.
UC San Diego

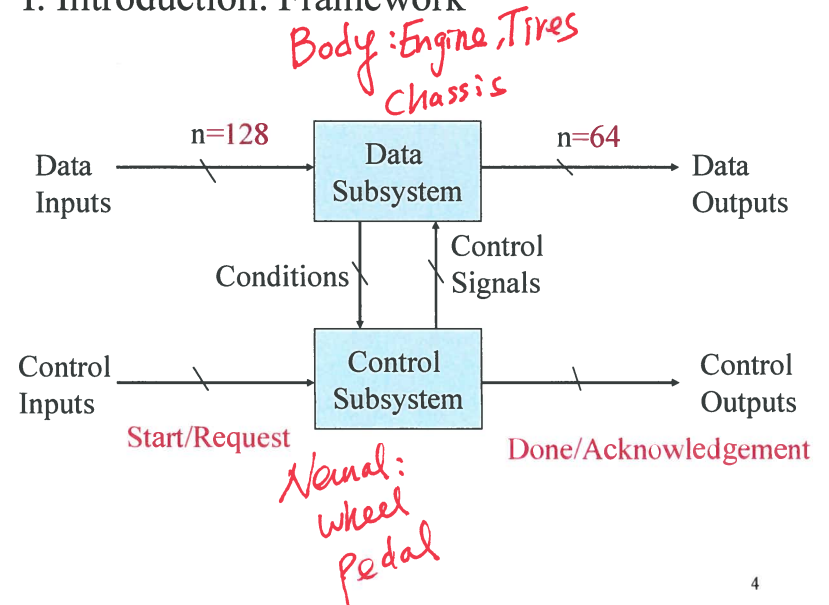
- Introduction
 - Methodology and Framework
- Components
- Specification
- Implementation

Introduction

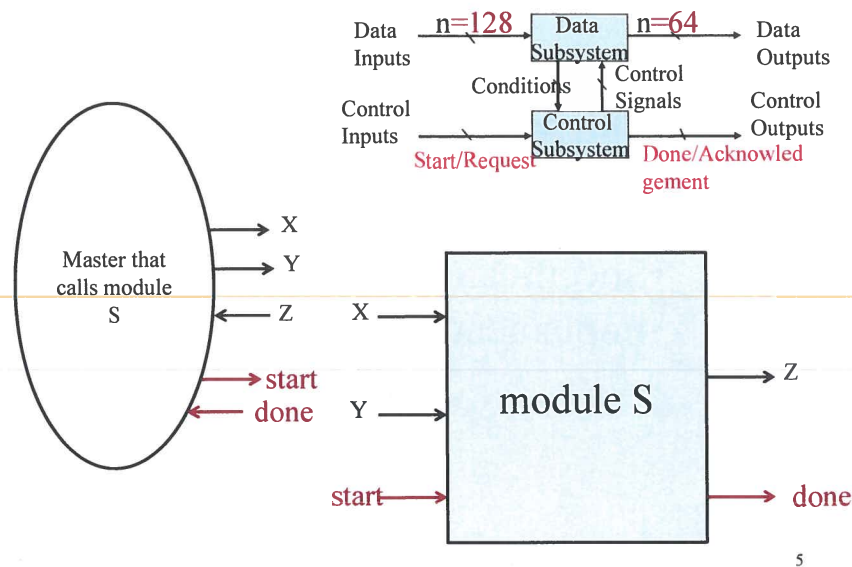
- Methodology
 - Approach with success stories.
 - Hierarchical designs with interface between the modules (BSV). *10 billions trans.*
- Data Subsystem and Control Subsystem
 - For n-bit data, each operation takes n times or more in hardware complexity.
 - Data subsystem carries out the data operations and transports.
 - Control system sequences the data subsystem and itself.



I. Introduction: Framework

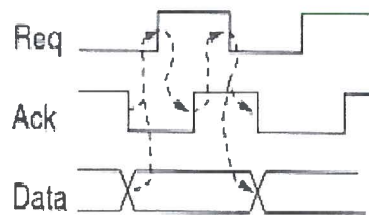


I. Introduction: Handshaking

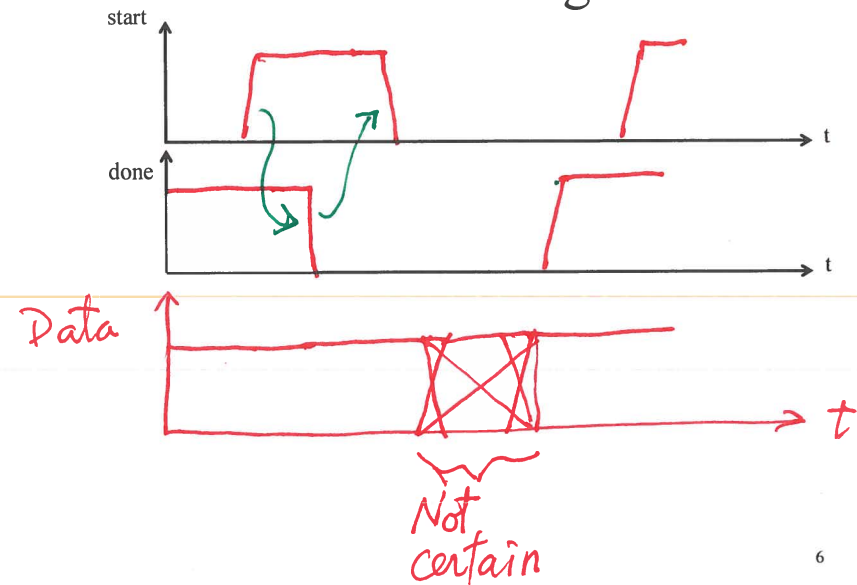


ARM AMBA: Advanced Microcontroller
Inc. Bus Architecture.

Handshaking



Handshaking



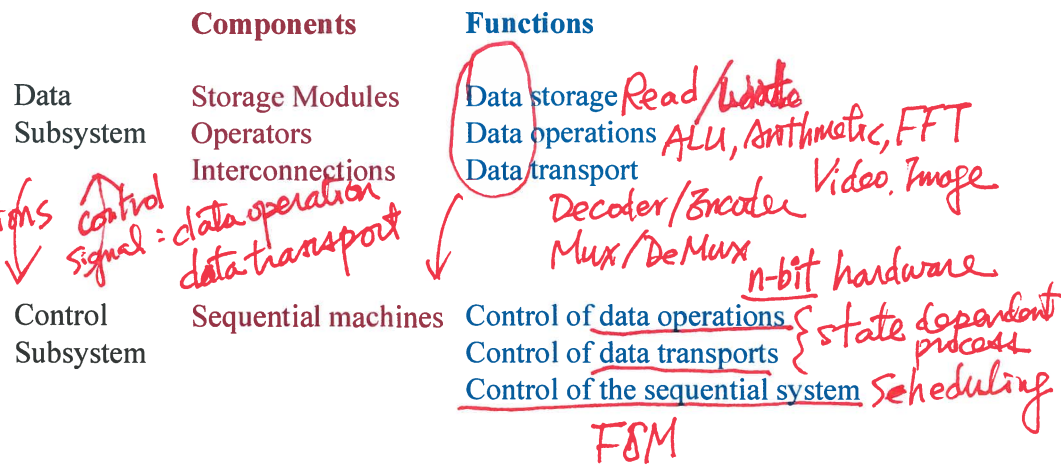
9WB

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

II. Components



9

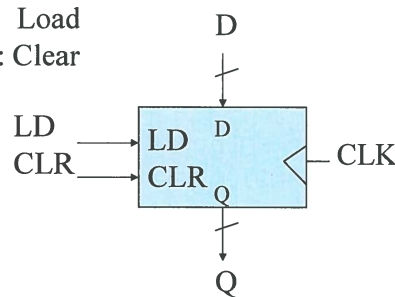
II. Data Subsystem Components

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

10

II. Components: Storage Modules, Register

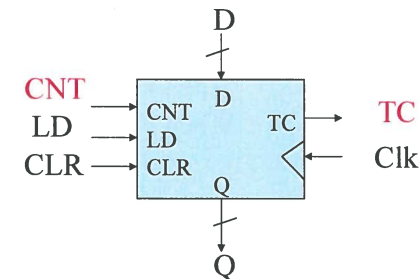
LD: Load
CLR: Clear



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

11

Modulo-n Counter



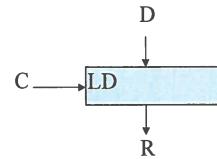
$$\begin{aligned}
 Q(t+1) &= (0, 0, \dots, 0) && \text{if CLR} = 1 \\
 &= D && \text{if LD} = 1 \text{ and CLR} = 0 \\
 &= (Q(t)+1) \bmod n && \text{if LD} = 0, \text{CNT} = 1 \text{ and CLR} = 0 \\
 &= Q(t) && \text{if LD} = 0, \text{CNT} = 0 \text{ and CLR} = 0
 \end{aligned}$$

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

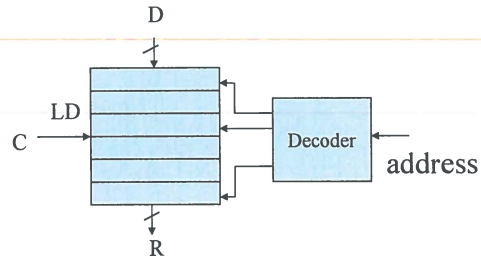
12

Storage Component: Registers, Array of Registers

Registers: If C then $R \leftarrow D$

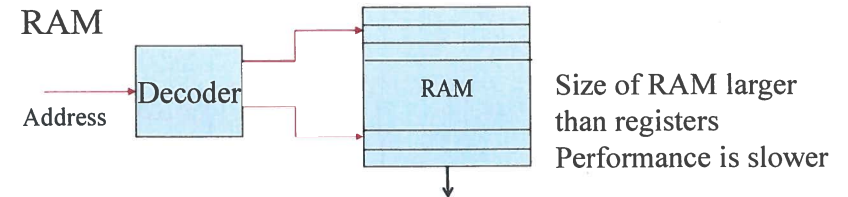


Register Array: If C then $R_{\text{address}} \leftarrow D$
Sharing connections and controls

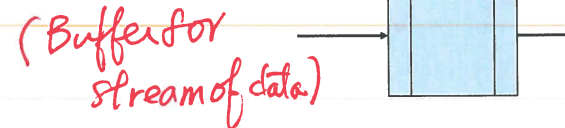


13

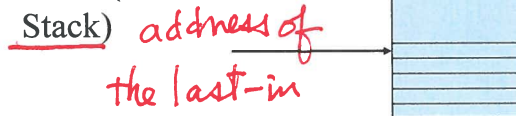
Storage Components: RAM, FIFO, LIFO



FIFO (First in first out)

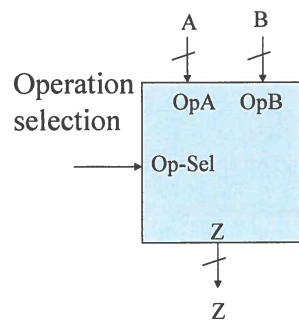


LIFO (Last in first out:



14

Functional Modules



CASE Op-Sel Is
When F1, $Z \leftarrow A \text{ op1 } B$
When F2, $Z \leftarrow A \text{ op2 } B$
.
.
End CASE

Example:
CASE Op-Set Is
 $Z \leftarrow (A + B) \bmod 2^n$ if Op-Sel=addition,
 $Z \leftarrow (A - B) \bmod 2^n$ if Op-Sel=subtraction
End CASE

15

Interconnect Modules (Wires and Switches)

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

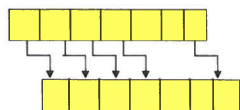
1. Single line (shifting, time sharing)



16

metal wire 100nm width - 10nm (many layers)
transistor 50nm x 50nm

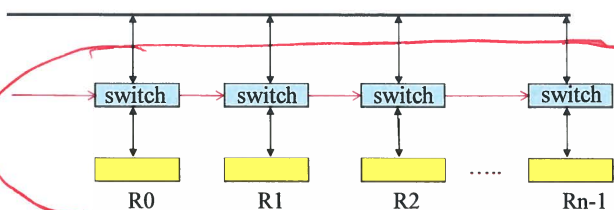
2. Band of Wires (BUS)



possible communication

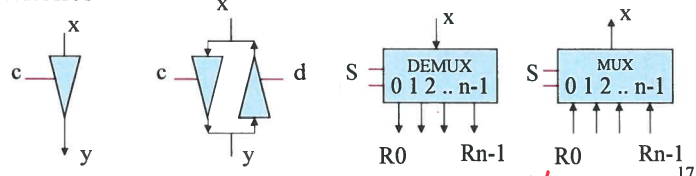
$$C(n, 2)$$

3. Shared Bus



Mux
DeMux

Switches

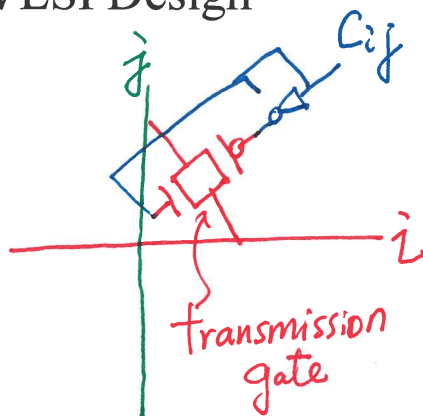
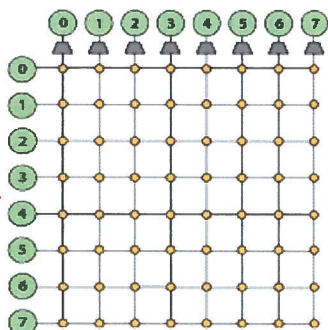


n devices share the same bus.
(time sharing)

(channel)

Crossbar Switches: VLSI Design

n buses



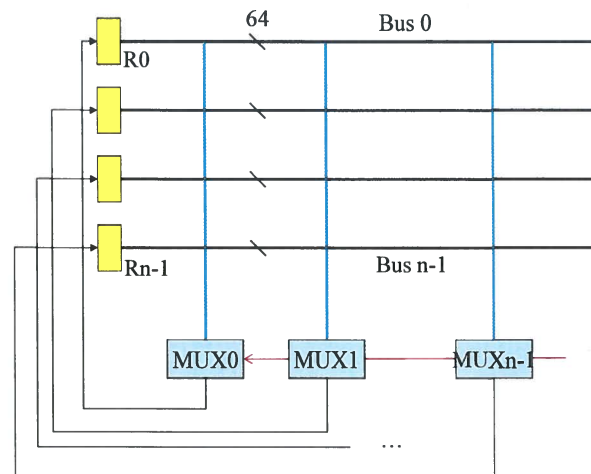
transmission gate

$$C(m, 2) = \frac{m(m-1)}{2} \text{ possible communication}$$

$$m \times n \text{ switches } (N^2 \text{ switches})$$

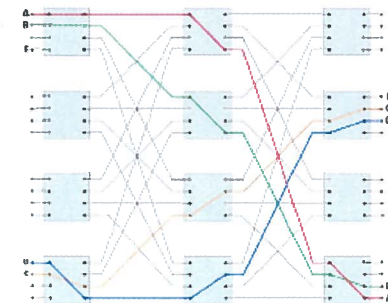
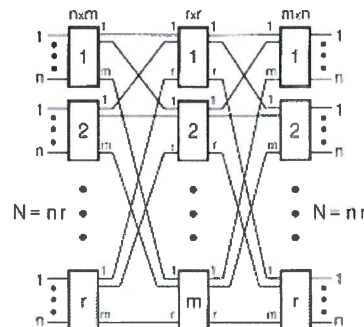
4. Crossbar (Multiple buses running horizontally)

m simultaneous transfers are possible, but more expensive.



m devices share n buses
(n)

Multistage Network: Clos Network



$$N = n \times r \quad m \times r \times r \quad N = n \times r$$

$$r \times n \times m$$

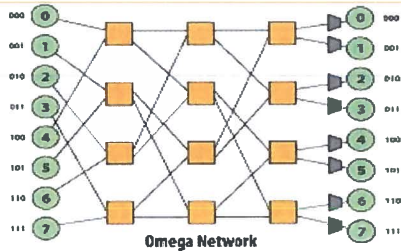
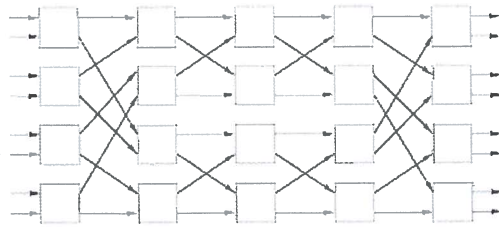
$$r \times n \times m$$

#switches

$$= 2 \times \frac{n \times r \times m}{2} + m r^2$$

$$= 2 \times N \times m + m r^2$$

Multistage Networks: Benes and Omega Networks



Blocking: Previous assignment can block the next route.

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

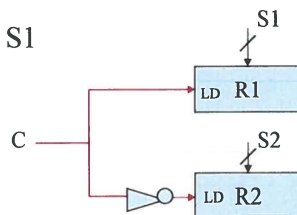
$R1 \leftarrow R2$: Registers and Interconnect

Begin, End: Control

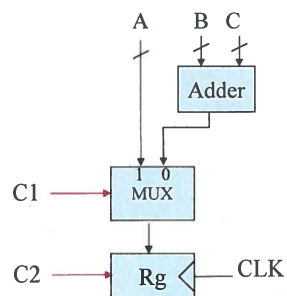
if () then (), ENDIF: Control

22

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

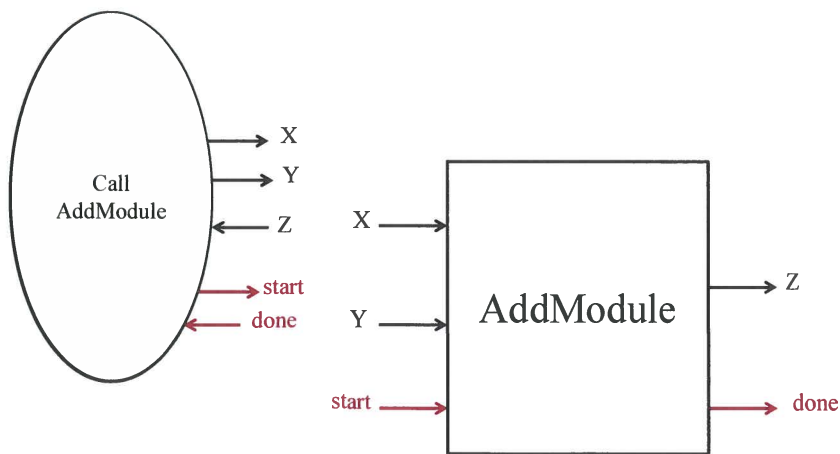
VI. Implementation: Example

```
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

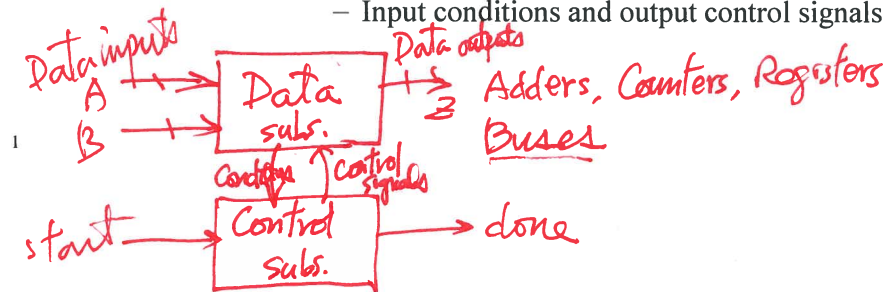
Design Process

CSE 140 Lecture 17 System Design II

CK Cheng
CSE Dept.
UC San Diego

*Partition
into
Data/Control
subsystems*

- Describe system in programs
- Data subsystem
 - List data operations
 - Map operations to functional blocks
 - Add interconnect for data transport
 - Input control signals and output conditions
- Control Subsystem
 - Derive the sequence according to the hardware program
 - Create the sequential machine
 - Input conditions and output control signals



2
IOWA

Example: Multiplication

Arithmetic

$$Z = X \times Y$$

- $M = 0$

- For $i = n-1$ to 0

- If $Y_i = 1$, $M = M + X \times 2^i$

- $Z = M$

Input X, Y

Output Z

Variable M, i

- $M = 0$
- For $i = n-1$ to 0
 - If $Y_{n-1} = 1$, $M = M + X$
 - Shift Y left by one bit
 - If $i \neq 0$, shift M left by one bit
- $Z = M$

Implementation: Example

```

Multiply(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], M[31:0], i[4:0] type bit-
vector;
  Output Z[31:0] type bit-vector,    done type Boolean;
  S0: If start' goto S0 || done  $\leftarrow$  1;
  S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || i  $\leftarrow$  0 || M  $\leftarrow$  0 || done  $\leftarrow$  0;
  S2: If B15 = 0 goto S4 || i  $\leftarrow$  i+1;
  S3: M  $\leftarrow$  M+A;
  S4: if i >= 16, goto S6
  S5: M  $\leftarrow$  Shift(M,L,1) || B  $\leftarrow$  Shift(B,L,1) || goto S2;
  S6: Z:  $\leftarrow$  M || done  $\leftarrow$  1 || goto S0;
}

```

5

Step 0: Syntax

```

Multiply(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], M[31:0], i[4:0] type bit-
vector;
  Output Z[31:0] type bit-vector, done type boolean;
  S0: If start' goto S0 || done  $\leftarrow$  1;
  S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || i  $\leftarrow$  0 || M  $\leftarrow$  0 || done  $\leftarrow$  0;
  S2: If B15 = 0 goto S4 || i  $\leftarrow$  i+1;
  S3: M  $\leftarrow$  M+A;
  S4: if i >= 16, goto S6
  S5: M  $\leftarrow$  Shift(M,L,1) || B  $\leftarrow$  Shift(B,L,1) || goto S2;
  S6: Z:  $\leftarrow$  M || done  $\leftarrow$  1 || goto S0;
}

```

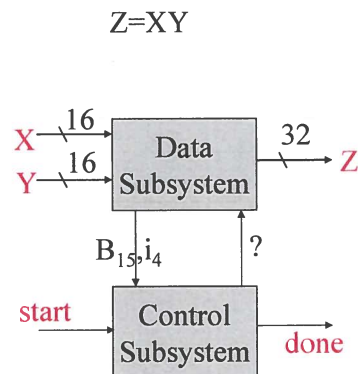
6

Step 1: Identify Input and Output of data and control subsystems

```

Multiply(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],
M[31:0], i[4:0] type bit-vector;
  Output Z[31:0] type bit-vector,
done type boolean;
  S0: If start' goto S0 || done  $\leftarrow$  1;
  S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || i  $\leftarrow$  0 || M  $\leftarrow$  0 ||
done  $\leftarrow$  0;
  S2: If B15 = 0 goto S4 || i  $\leftarrow$  i+1;
  S3: M  $\leftarrow$  M+A;
  S4: if i >= 16, goto S6
  S5: M  $\leftarrow$  Shift(M,L,1) ||
B  $\leftarrow$  Shift(B,L,1) || goto S2;
  S6: Z:  $\leftarrow$  M || done  $\leftarrow$  1 || goto S0;
}

```



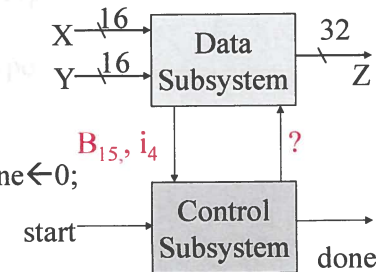
7

Step 2: Identify Condition Bits to Control Subsystem

```

Multiply(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], M[31:0],
i[4:0] type bit-vector;
  Output Z[31:0] type bit-vector, done type
boolean;
  S0: If start' goto S0 || done  $\leftarrow$  1;
  S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || i  $\leftarrow$  0 || M  $\leftarrow$  0 ||
done  $\leftarrow$  0;
  S2: If B15 = 0 goto S4 || i  $\leftarrow$  i+1;
  S3: M  $\leftarrow$  M+A;
  S4: if i >= 16, goto S6
  S5: M  $\leftarrow$  Shift(M,L,1) || B  $\leftarrow$  Shift(B,L,1) ||
goto S2;
  S6: Z:  $\leftarrow$  M || done  $\leftarrow$  1 || goto S0;
}

```



8