

CSE 140 Lecture 17

System Design II

CK Cheng
CSE Dept.
UC San Diego

1

High Level Synthesis

- Hierarchical Description
 - 4-Phase Handshaking
- Graph Representation
 - Control/Data Flow Graphs
- Data Subsystem
 - Allocation *Components (cover all operations)*
 - Binding *spatial vs time sharing.*
- Scheduling
 - Finite State Machine

Reference:

Bluespec System

FPGAs for Software Programmers, D. Koch, F. Hannig, D. Ziener, Springer 2016

2

$$\begin{array}{r}
 \begin{array}{l}
 i=0 \quad M=0. \\
 y_2=1 \\
 M = M + X = 110
 \end{array}
 \quad
 \begin{array}{r}
 \begin{array}{r}
 110 \times 6 \\
 \times 101 \quad Y \quad 5 \\
 \hline
 110 \\
 000 \\
 110 \\
 \hline
 11110 \\
 168421 \Rightarrow 30
 \end{array}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{l}
 i=1 \\
 y_2=0 \\
 M = M_{\text{shift left}} = 11000
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{l}
 i=2. \\
 y_2=1 \\
 M = M + X = 11000 + 110 \\
 = \cancel{11110} \\
 168421
 \end{array}
 \quad
 \begin{array}{r}
 1100 \\
 110 \quad 5
 \end{array}
 \end{array}$$

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;

}

High Level Synthesis

- 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

Allocation.

Binding.

3

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
- $Z = M$

$$\begin{array}{r}
 110 \times \\
 101 \times \\
 \hline
 y_2 = 1, M = M + X \\
 = 110 \\
 1100 \\
 \hline
 y_2 = 0, 11000 \\
 \hline
 y_2 = 1, 11000 \\
 1100 \\
 \hline
 11100 \\
 \hline
 \rightarrow 30
 \end{array}$$

4

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;
}

```

7

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

```

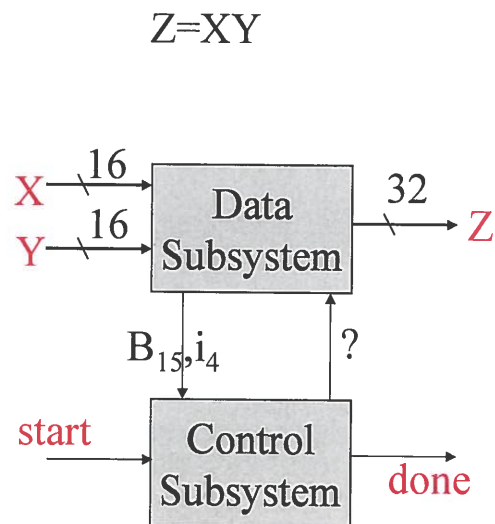
Multiply(X, Y, Z, start, done)
{ Input: X[15:0], Y[15:0] type bit-
vector, start type boolean;
  Local-Object : A[15:0], B[15:0],
M[31:0], i[4:0] type bit-vector;
  Output Z[31:0] type bit-vector,
done type boolean;

```

```

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;
}

```



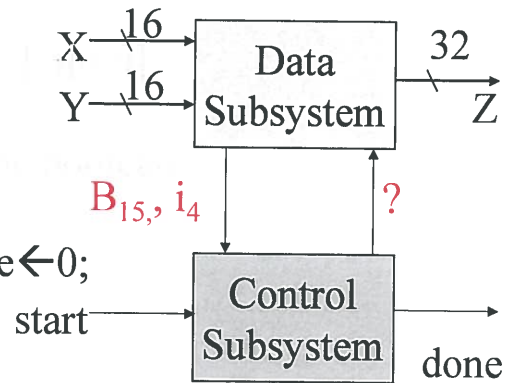
8

Step 2: Identify Condition Bits to Control Subsystem

Multiply(X, Y, Z, start, done)

```
{ Input: X[16], Y[16], Z[32], start, done
  Local: A[16], B[16], i[4], M[16]
  type bit-vector
  Output: Z[32], 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  $B_{15} = 0$  goto S4 || i  $\leftarrow$  i+1;
S3: M  $\leftarrow$  M+A;
S4: if i  $\geq$  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;
}
```



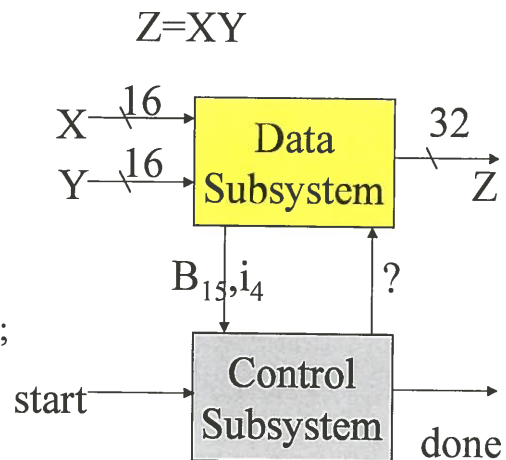
9

Step 3: Identify Data Subsystem Operations

Multiply(X, Y, Z, start, done)

```
{ Input: X[16], Y[16], Z[32], start, done
  Local: A[16], B[16], i[4], M[16]
  type bit-vector
  Output: Z[32], 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  $B_{15} = 0$  goto S4 || i  $\leftarrow$  i+1;
S3: M  $\leftarrow$  M+A;
S4: if i  $\geq$  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;
}
```



10

Step 4: Map Data Operations to Implementable functions

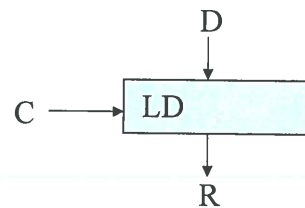
Multiply(X, Y, Z, start, done)

{		operation
	$A \leftarrow X$	$A \leftarrow \text{Load}(X)$
	$B \leftarrow Y$	$B \leftarrow \text{Load}(Y)$
	$M \leftarrow 0$	$M \leftarrow \text{Clear}(M)$
	$i \leftarrow 0$	$i \leftarrow \text{Clear}(i)$
	$i \leftarrow i + 1$	$i \leftarrow \text{INC}(i)$
S0: If start' goto S0 done $\leftarrow 1$;	$M \leftarrow M + A$	$M \leftarrow \text{Add}(M, A)$
S1: $A \leftarrow X$ $B \leftarrow Y$ $i \leftarrow 0$ $M \leftarrow 0$ done $\leftarrow 0$;	$M \leftarrow \text{Shift}(M, L, 1)$	$M \leftarrow \text{SHL}(M)$
S2: If $B_{15} = 0$ goto S4 $i \leftarrow i + 1$;	$B \leftarrow \text{Shift}(B, L, 1)$	$B \leftarrow \text{SHL}(B)$
S3: $M \leftarrow M + A$;	$Z \leftarrow M$	Wires
S4: if $i \geq 16$, goto S6		
S5: $M \leftarrow \text{Shift}(M, L, 1)$ $B \leftarrow \text{Shift}(B, L, 1)$ goto S2;		
S6: $Z \leftarrow M$ done $\leftarrow 1$ goto S0;		
}		

11

Step 5: Implement the Data Subsystem from Standard Modules

Registers: If C then $R \leftarrow D$



operation

$A \leftarrow \text{Load}(X)$

$B \leftarrow \text{Load}(Y)$

$M \leftarrow \text{Clear}(M)$

$i \leftarrow \text{Clear}(i)$

$i \leftarrow \text{INC}(i)$

$M \leftarrow \text{Add}(M, A)$

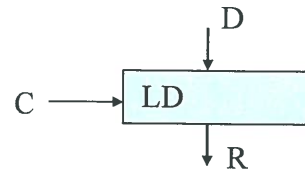
$M \leftarrow \text{SHL}(M)$

$B \leftarrow \text{SHL}(B)$

12

Storage Component: Registers with control signals

Registers: If C then $R \leftarrow D$



operation

$A \leftarrow \text{Load}(X)$

$B \leftarrow \text{Load}(Y)$

$M \leftarrow \text{Clear}(M)$

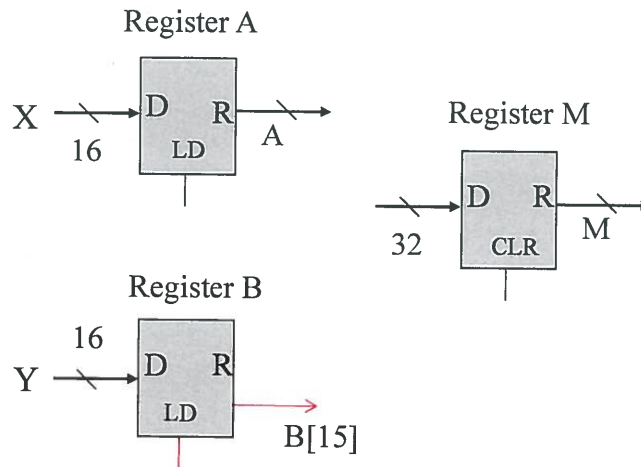
$i \leftarrow \text{Clear}(i)$

$i \leftarrow \text{INC}(i)$

$M \leftarrow \text{Add}(M, A)$

$M \leftarrow \text{SHL}(M)$

$B \leftarrow \text{SHL}(B)$



13

Data Subsystem

operation

$A \leftarrow \text{Load}(X)$

$B \leftarrow \text{Load}(Y)$

$M \leftarrow \text{Clear}(M)$

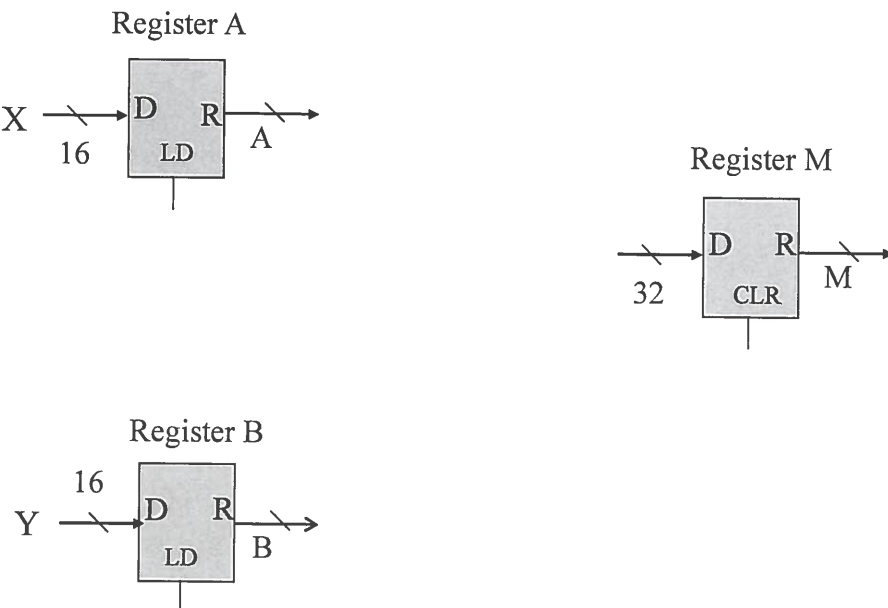
$i \leftarrow \text{Clear}(i)$

$i \leftarrow \text{INC}(i)$

$M \leftarrow \text{Add}(M, A)$

$M \leftarrow \text{SHL}(M)$

$B \leftarrow \text{SHL}(B)$



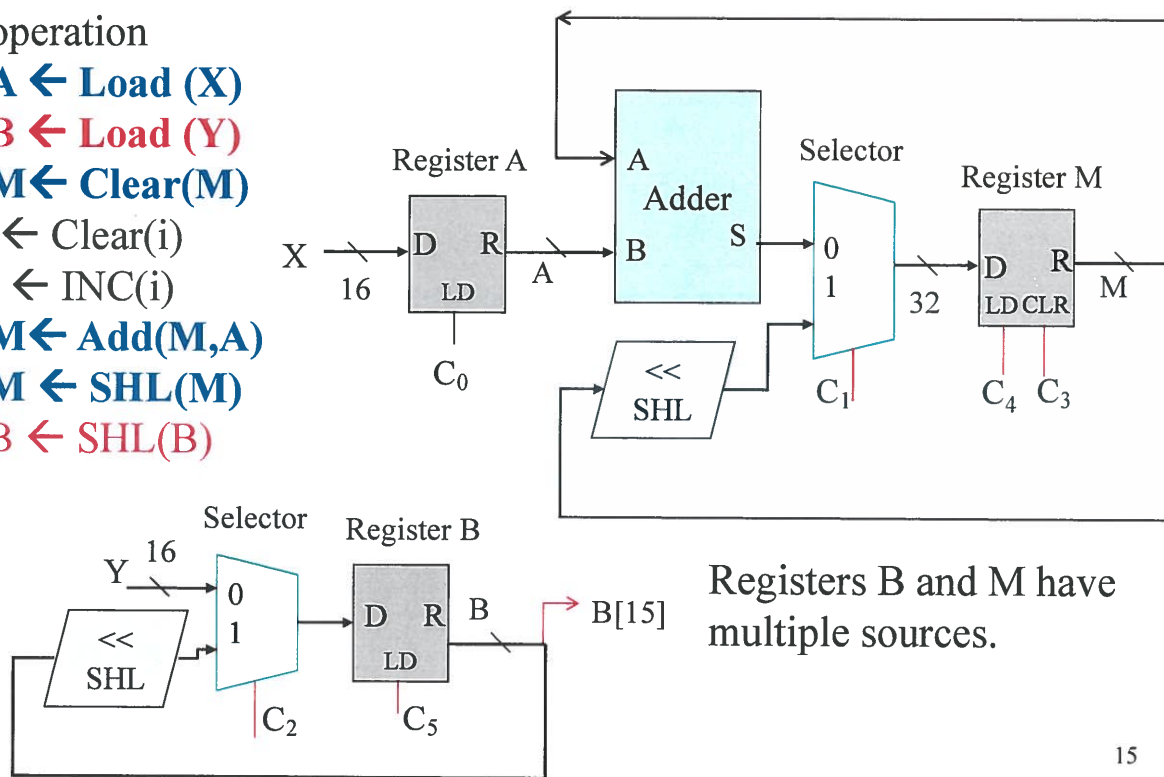
14

Function Modules: Adder, Shifter

operation

A ← Load (X)

B ← Load (Y)

$$M \leftarrow \text{Clear}(M)$$
$$i \leftarrow \text{Clear}(i)$$
$$i \leftarrow \text{INC}(i)$$
$$M \leftarrow \text{Add}(M, A)$$
$$M \leftarrow \text{SHL}(M)$$
$$B \leftarrow \text{SHL}(B)$$


Registers B and M have multiple sources.

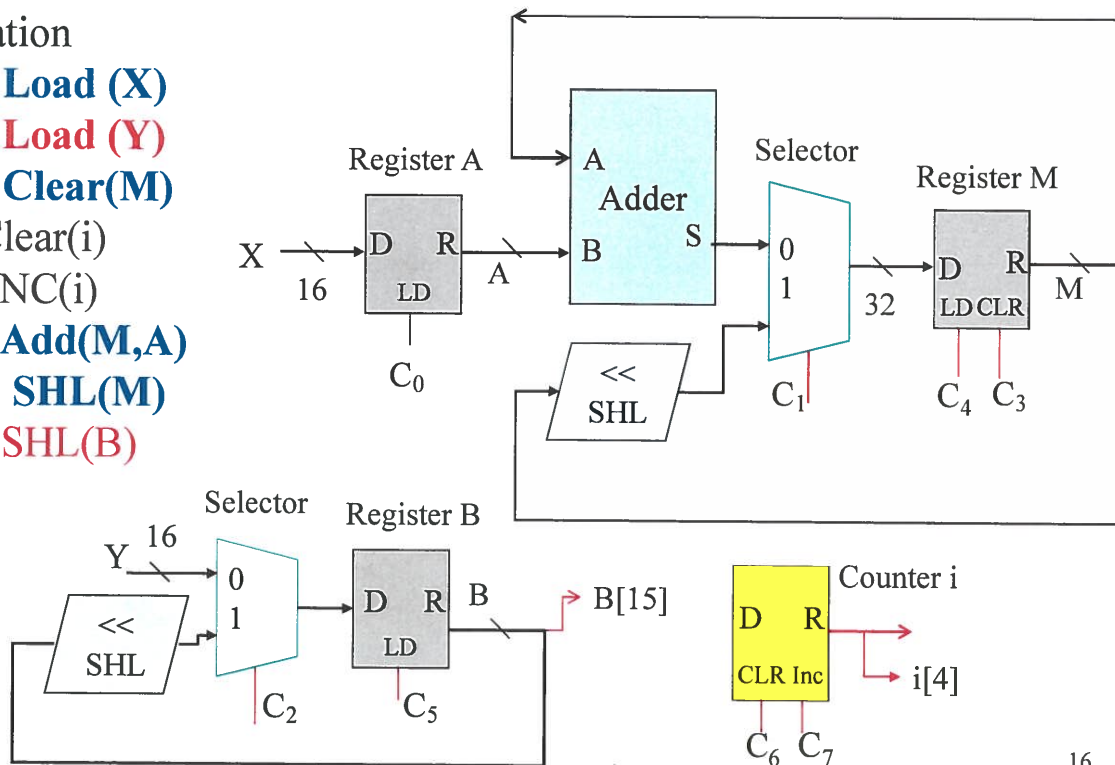
15

Function Modules: Adder, Shifter, Counter

operation

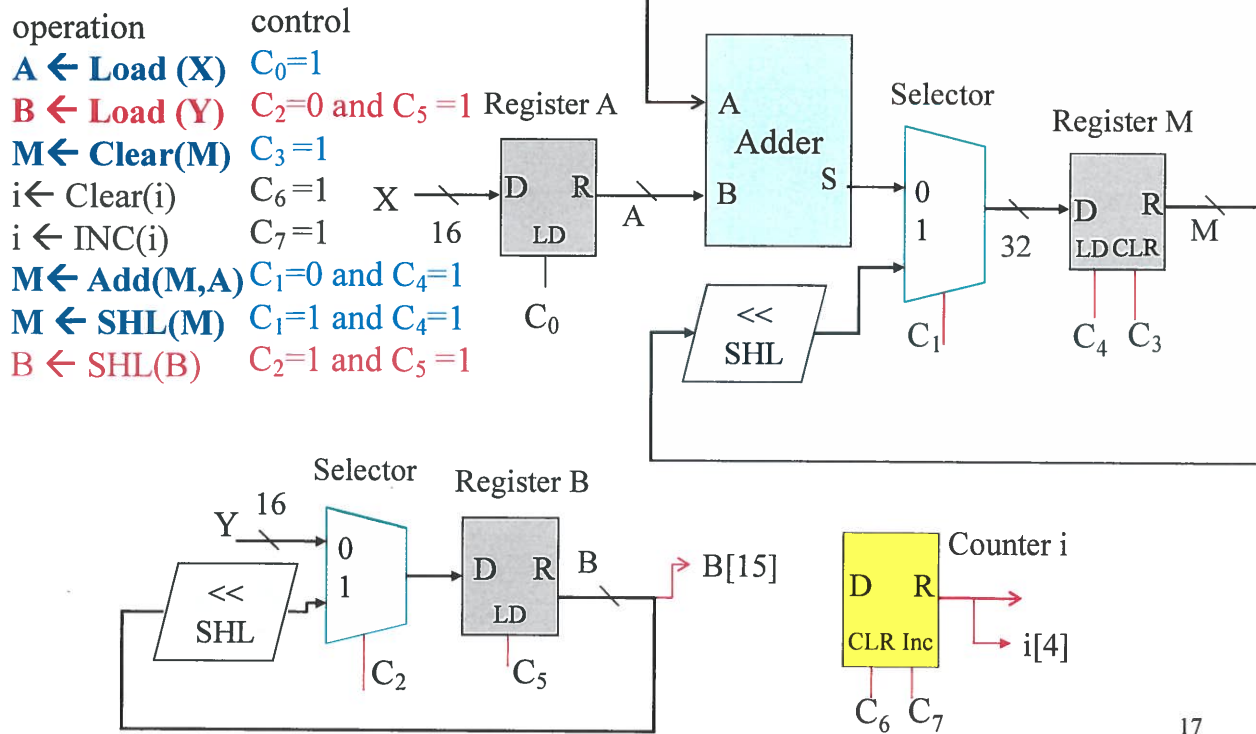
A ← Load (X)

B ← Load (Y)

$$M \leftarrow \text{Clear}(M)$$
$$i \leftarrow \text{Clear}(i)$$
$$i \leftarrow \text{INC}(i)$$
$$M \leftarrow \text{Add}(M, A)$$
$$M \leftarrow \text{SHL}(M)$$
$$B \leftarrow \text{SHL}(B)$$


16

Step 6: Map Control Signals to Operations



17

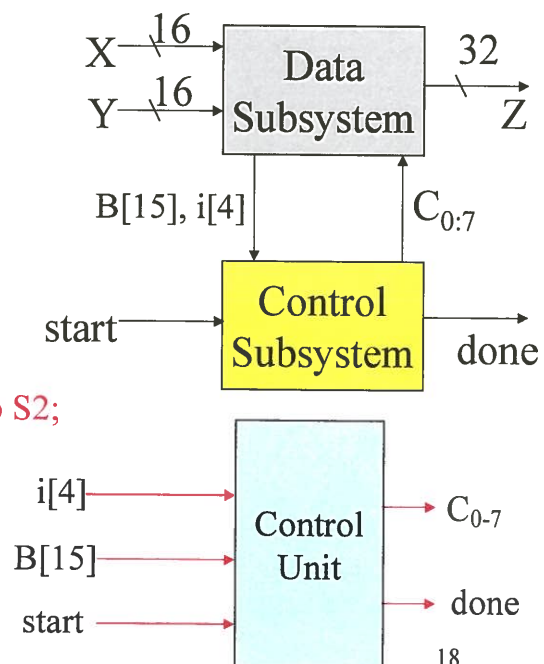
Step 7: Identify Control Path Components

Multiply(X, Y, Z, start, done)

```
{
  type: void;
  Local Object: A[15], B[15], i[4];
  Output Z[32] type: int;
  done type: bool;
}
```

```
S0: If start' goto S0 || done ← 1;
S1: A ← X || B ← Y || i ← 0 || M ← 0 || done ← 0;
S2: If B15 = 0 goto S4 || i ← i + 1;
S3: M ← M + A;
S4: if i ≥ 16, goto S6
S5: M ← Shift(M, L, 1) || B ← Shift(B, L, 1) || goto S2;
S6: Z ← M || done ← 1 || goto S0;
}
```

$Z = XY$



18