

CSE140: System Design

Xinyuan Wang

05/31/2019

Example: SP18 Final

(System Designs) Implement the following algorithm:

Alg(X, Y, Z, start, W, done);

Input X[7:0], Y[7:0], Z[7:0], start;

Output W[7:0], done;

Local-object A[7:0], B[7:0], C[7:0];

S0: If start' goto S0 || done $\Leftarrow$ 1;

S1: A $\Leftarrow$ X || B $\Leftarrow$ Y || C $\Leftarrow$ Z || done $\Leftarrow$ 0;

S2: if (A[7]) goto S5 || B $\Leftarrow$ B+C;

S3: A $\Leftarrow$ A+B;

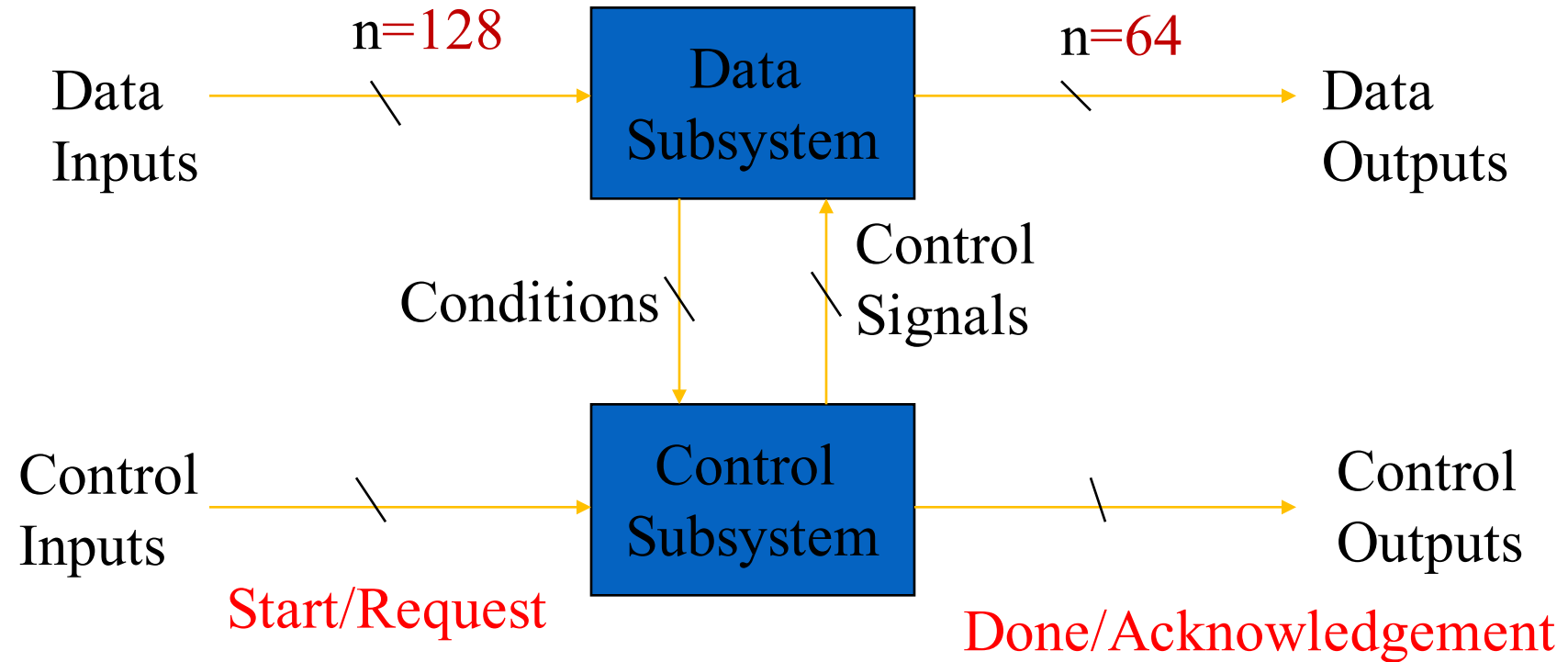
S4: if C'[7] goto S2 || C $\Leftarrow$ Inc(C);

S5: A $\Leftarrow$ Inc(A) || if C[7] goto S2;

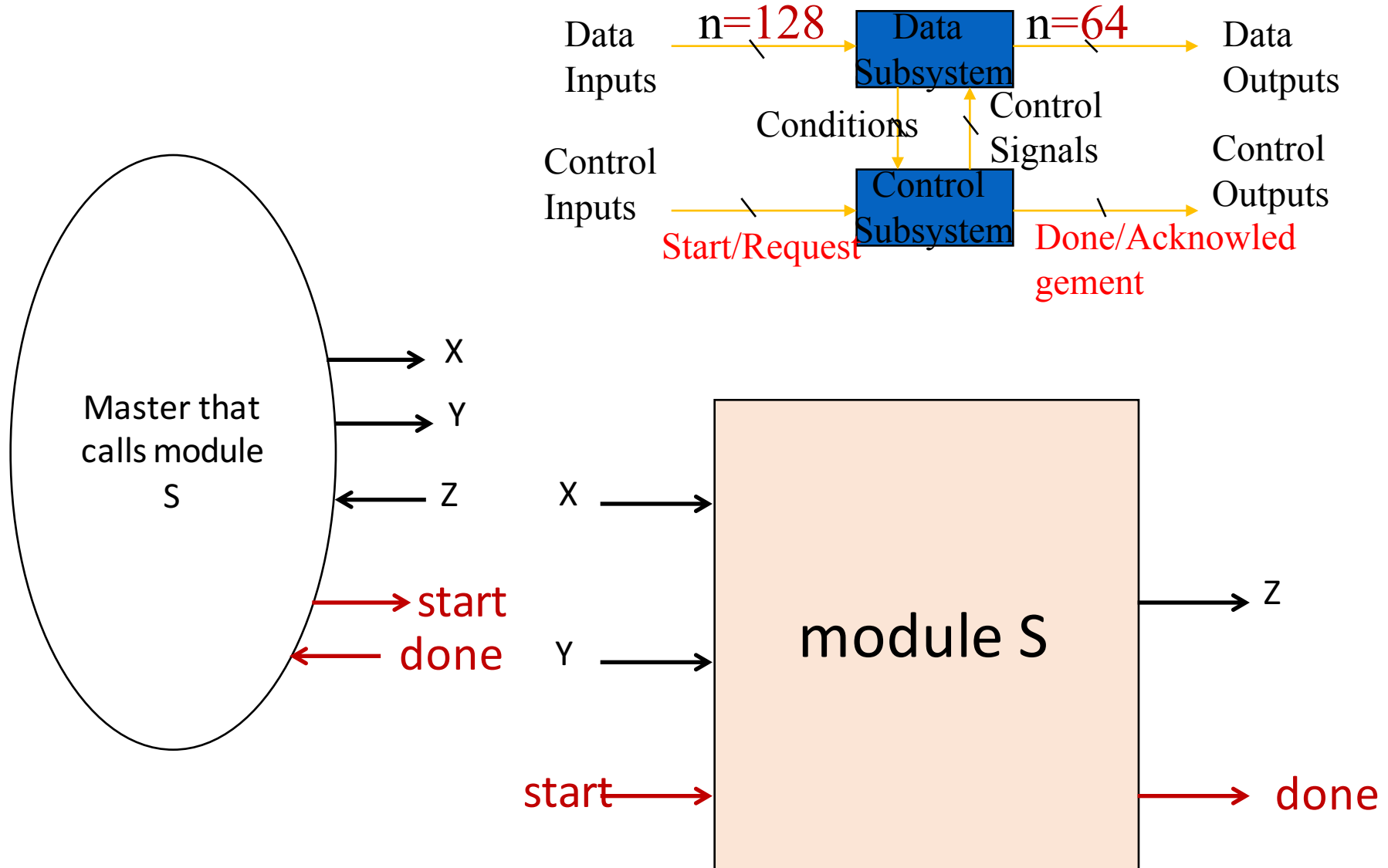
S6: W $\Leftarrow$ C || goto S0;

End Alg

Framework



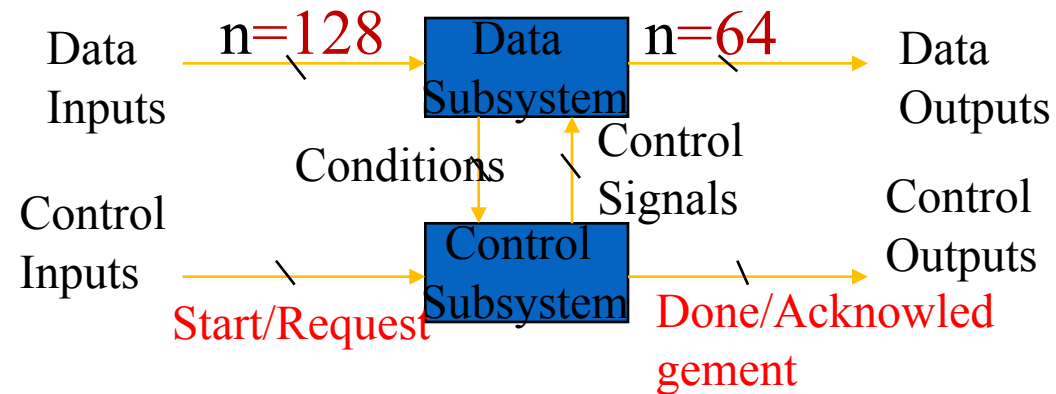
Handshaking



Before implementing the data and control subsystems

With the given description of system

- Identify Input and Output of data and control subsystems
- Identify Condition Bits to Control Subsystem
- Identify Data Subsystem Operations
- Map Data Operations to Implementable functions

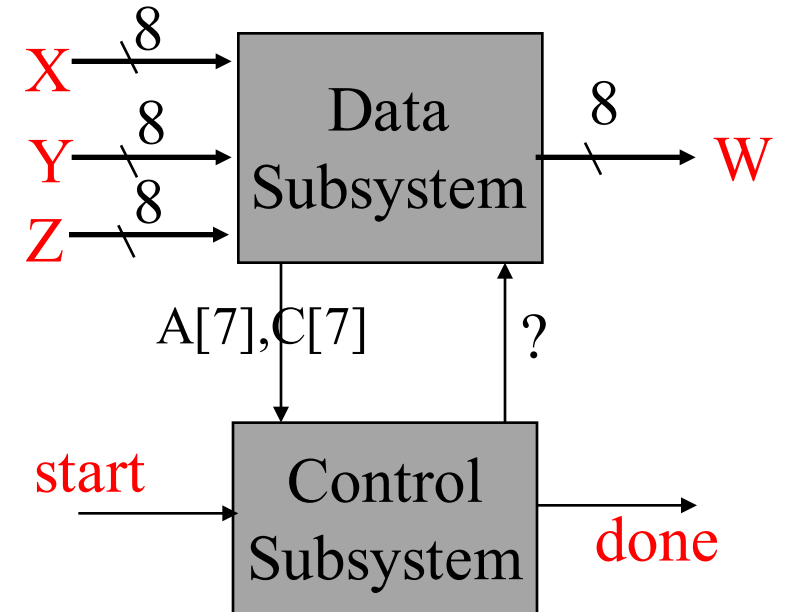


Before implementing the data and control subsystems

With the given description of system

- **Identify Input and Output of data and control subsystems**

```
Alg(X, Y, Z, start, W, done);  
Input X[7:0], Y[7:0], Z[7:0], start;  
Output W[7:0], done;  
Local-object A[7:0], B[7:0], C[7:0];  
S0: If start' goto S0 || done  $\Leftarrow$  1;  
S1: A  $\Leftarrow$  X || B  $\Leftarrow$  Y || C  $\Leftarrow$  Z || done  $\Leftarrow$  0;  
S2: if (A[7]) goto S5 || B  $\Leftarrow$  B+C;  
S3: A  $\Leftarrow$  A+B;  
S4: if C'[7] goto S2 || C  $\Leftarrow$  Inc(C);  
S5: A  $\Leftarrow$  Inc(A) || if C[7] goto S2;  
S6: W  $\Leftarrow$  C || goto S0;  
End Alg
```

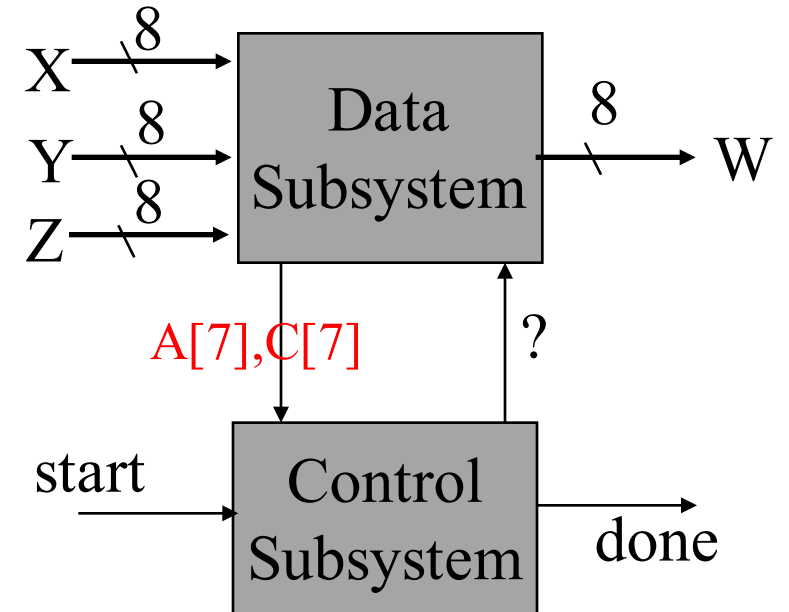


Before implementing the data and control subsystems

With the given description of system

- Identify Input and Output of data and control subsystems
- **Identify Condition Bits to Control Subsystem**

```
Alg(X, Y, Z, start, W, done);  
Input X[7:0], Y[7:0], Z[7:0], start;  
Output W[7:0], done;  
Local-object A[7:0], B[7:0], C[7:0];  
S0: If start' goto S0 || done  $\leftarrow$  1;  
S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || C  $\leftarrow$  Z || done  $\leftarrow$  0;  
S2: if (A[7]) goto S5 || B  $\leftarrow$  B+C;  
S3: A  $\leftarrow$  A+B;  
S4: if C'[7] goto S2 || C  $\leftarrow$  Inc(C);  
S5: A  $\leftarrow$  Inc(A) || if C[7] goto S2;  
S6: W  $\leftarrow$  C || goto S0;  
End Alg
```



Before implementing the data and control subsystems

With the given description of system

- Identify Input and Output of data and control subsystems
- Identify Condition Bits to Control Subsystem
- **Identify Data Subsystem Operations**

Alg(X, Y, Z, start, W, done);

Input X[7:0], Y[7:0], Z[7:0], start;

Output W[7:0], done;

Local-object A[7:0], B[7:0], C[7:0];

S0: If start' goto S0 || done $\leftarrow$ 1;

S1: $A \leftarrow X \parallel B \leftarrow Y \parallel C \leftarrow Z \parallel$ done $\leftarrow$ 0;

S2: if (A[7]) goto S5 || $B \leftarrow B+C$;

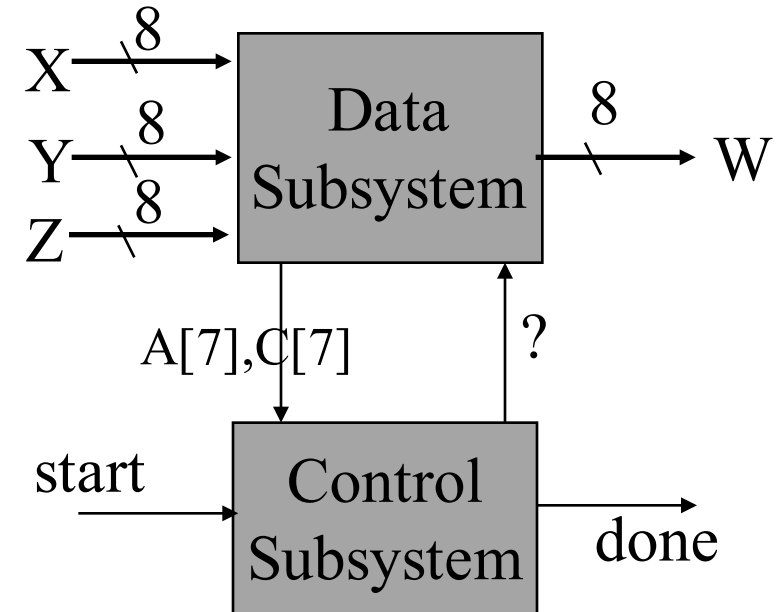
S3: $A \leftarrow A+B$;

S4: if C'[7] goto S2 || $C \leftarrow \text{Inc}(C)$;

S5: $A \leftarrow \text{Inc}(A)$ || if C[7] goto S2;

S6: $W \leftarrow C$ || goto S0;

End Alg



Before implementing the data and control subsystems

With the given description of system

- Identify Input and Output of data and control subsystems
- Identify Condition Bits to Control Subsystem
- Identify Data Subsystem Operations
- **Map Data Operations to Implementable functions**

Alg(X, Y, Z, start, W, done);

Input X[7:0], Y[7:0], Z[7:0], start;

Output W[7:0], done;

Local-object A[7:0], B[7:0], C[7:0];

S0: If start' goto S0 || done $\leftarrow$ 1;

S1: A $\leftarrow$ X || B $\leftarrow$ Y || C $\leftarrow$ Z || done $\leftarrow$ 0;

S2: if (A[7]) goto S5 || B $\leftarrow$ B+C;

S3: A $\leftarrow$ A+B;

S4: if C'[7] goto S2 || C $\leftarrow$ Inc(C);

S5: A $\leftarrow$ Inc(A) || if C[7] goto S2;

S6: W $\leftarrow$ C || goto S0;

End Alg

state	Instruction	Operation
S1	A $\leftarrow$ X	A $\leftarrow$ Load(X)
S1	B $\leftarrow$ Y	B $\leftarrow$ Load(Y)
S1	C $\leftarrow$ Z	C $\leftarrow$ Load(Z)
S2	B $\leftarrow$ B+C	B $\leftarrow$ Add(B,C)
S3	A $\leftarrow$ A+B	A $\leftarrow$ Add(A,B)
S4	C $\leftarrow$ Inc(C)	C $\leftarrow$ Inc(C)
S5	A $\leftarrow$ Inc(A)	A $\leftarrow$ Inc(A)

state	Instruction	Operation
S0	done $\leftarrow$ 1	
S6	W $\leftarrow$ C	Wire

Implement the data subsystem

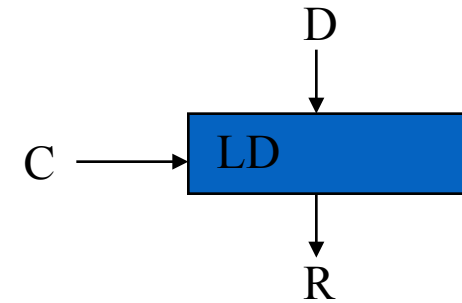
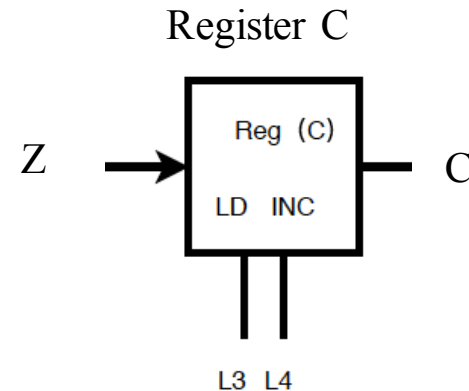
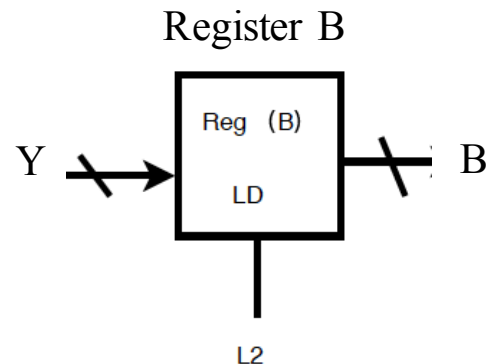
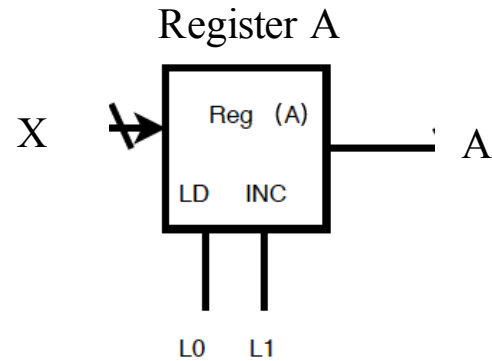
- List data operations
- Map operations to functional blocks
- Add interconnect for data transport
- Input control signals and output conditions

Implement the data subsystem

- List data operations
- Map operations to functional blocks: registers**

Registers: If C then $R \leftarrow D$

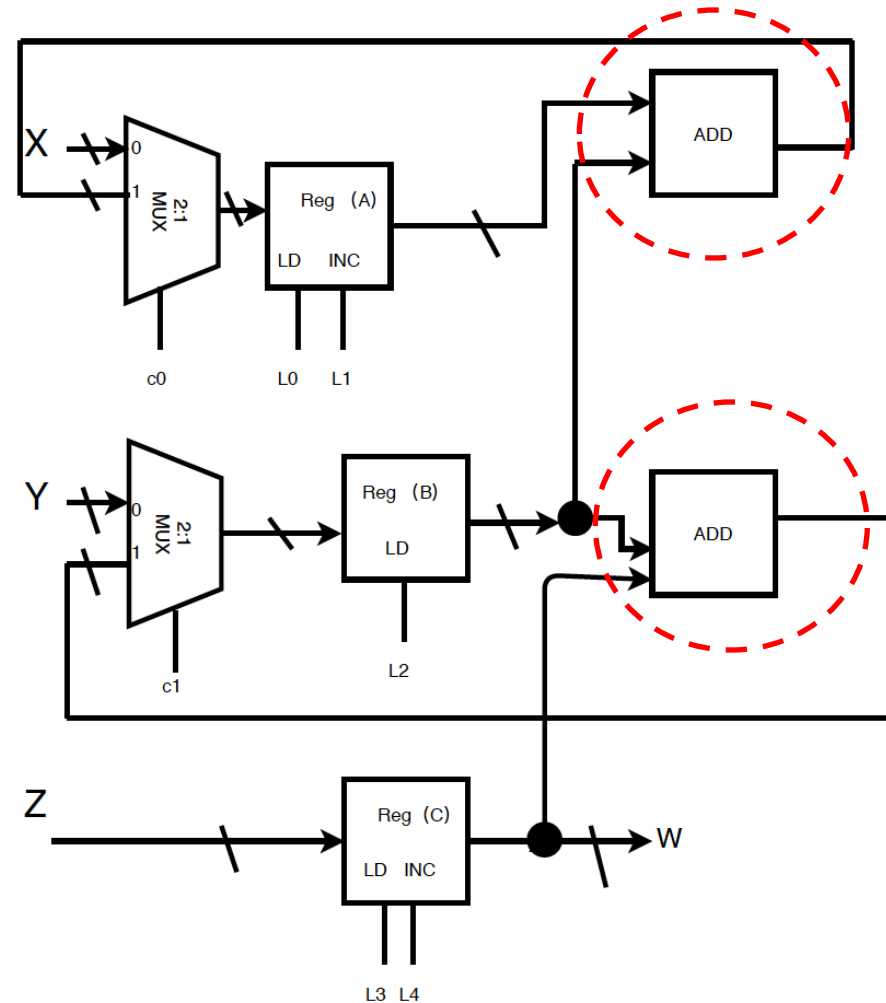
Operation
$A \leftarrow \text{Load}(X)$
$B \leftarrow \text{Load}(Y)$
$C \leftarrow \text{Load}(Z)$
$B \leftarrow \text{Add}(B, C)$
$A \leftarrow \text{Add}(A, B)$
$C \leftarrow \text{Inc}(C)$
$A \leftarrow \text{Inc}(A)$



Implement the data subsystem

- List data operations
- **Map operations to functional blocks: adders**
- Add interconnect for data transport

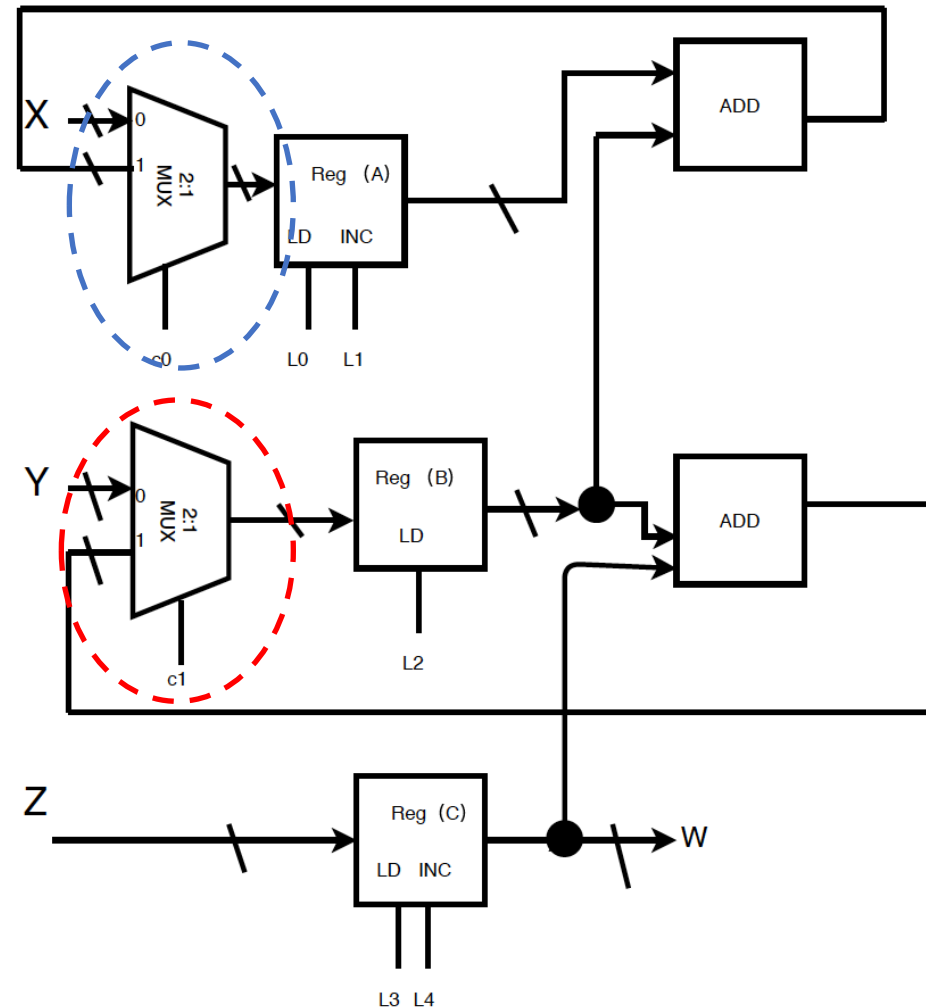
Operation
$A \leftarrow \text{Load}(X)$
$B \leftarrow \text{Load}(Y)$
$C \leftarrow \text{Load}(Z)$
$B \leftarrow \text{Add}(B, C)$
$A \leftarrow \text{Add}(A, B)$
$C \leftarrow \text{Inc}(C)$
$A \leftarrow \text{Inc}(A)$



Implement the data subsystem

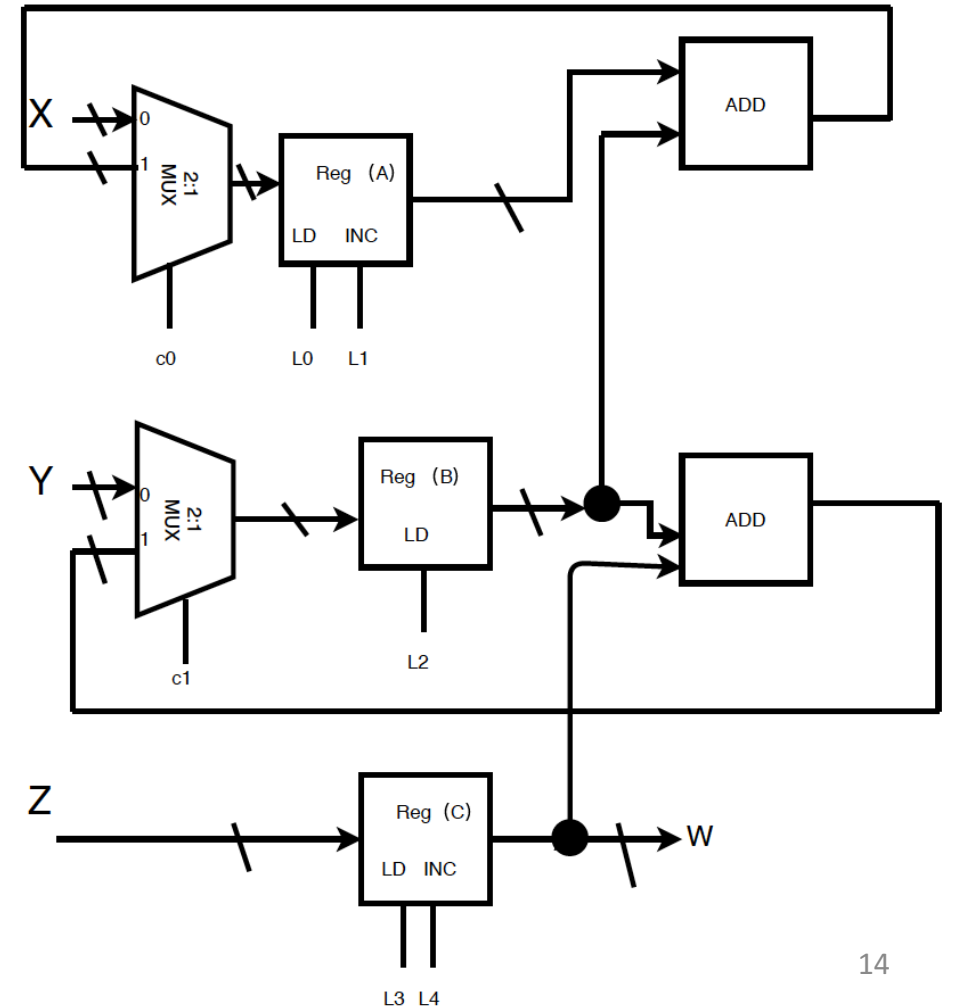
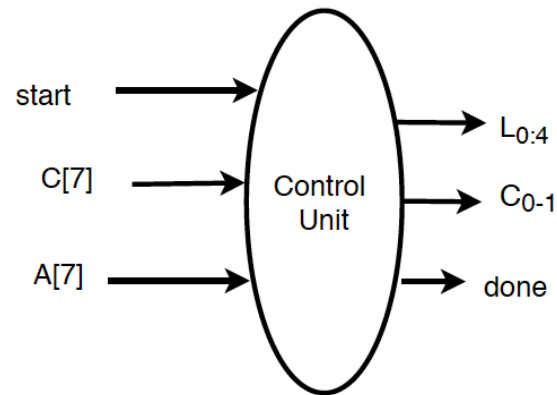
- List data operations
- **Map operations to functional blocks: mux for multiple sources**
- Add interconnect for data transport

Operation
$A \leftarrow \text{Load}(X)$
$B \leftarrow \text{Load}(Y)$
$C \leftarrow \text{Load}(Z)$
$B \leftarrow \text{Add}(B, C)$
$A \leftarrow \text{Add}(A, B)$
$C \leftarrow \text{Inc}(C)$
$A \leftarrow \text{Inc}(A)$



Implement the data subsystem

- List data operations
- Map operations to functional blocks
- Add interconnect for data transport
- **Input control signals and output conditions**



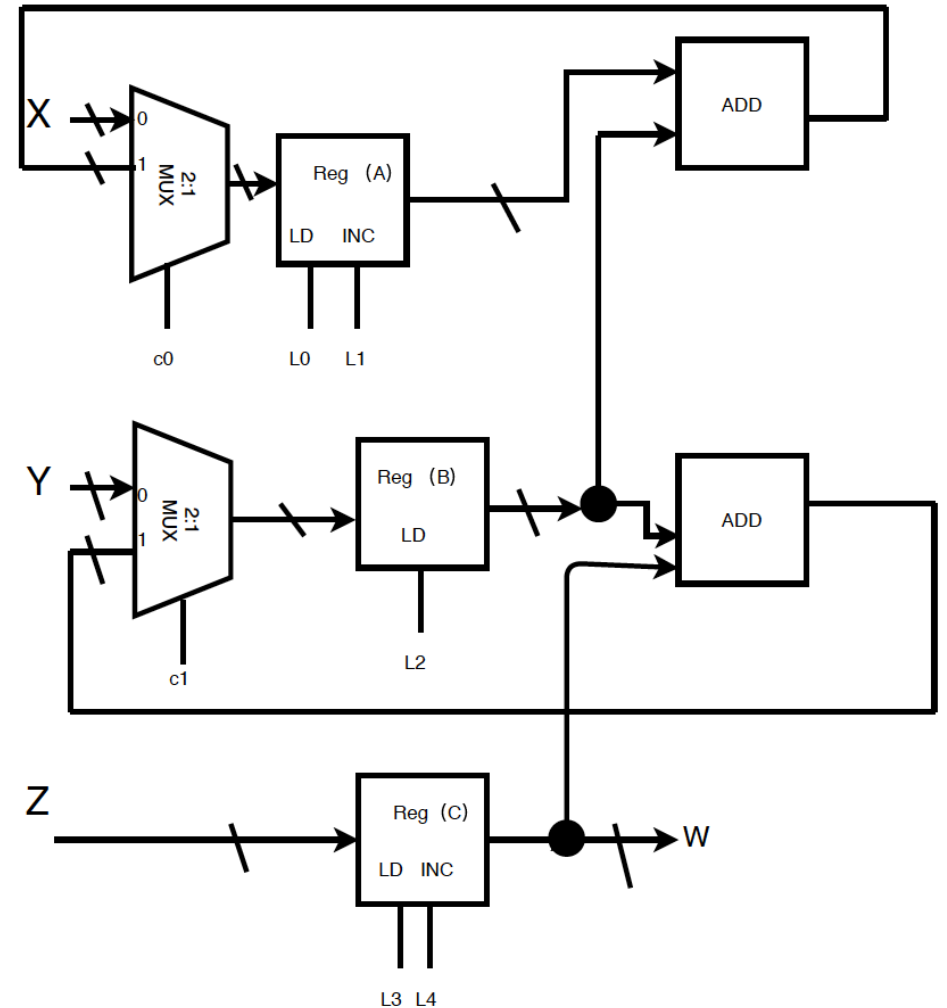
Implement the control subsystem

- Map control signals to operations
- Identify control path components
- Express the control subsystem with FSM
- Implement the control subsystem

Implement the control subsystem

- **Map control signals to operations**

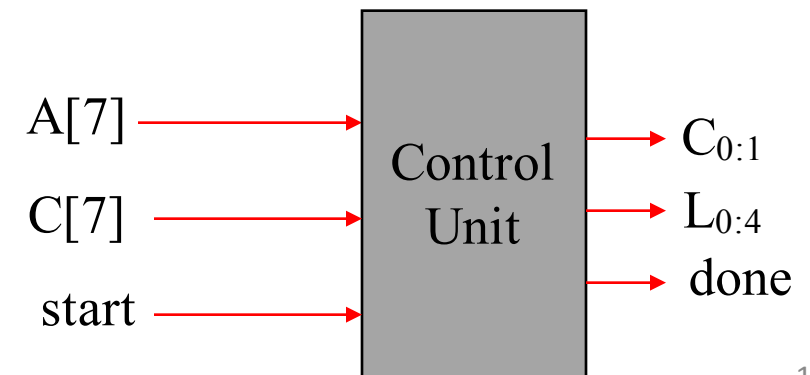
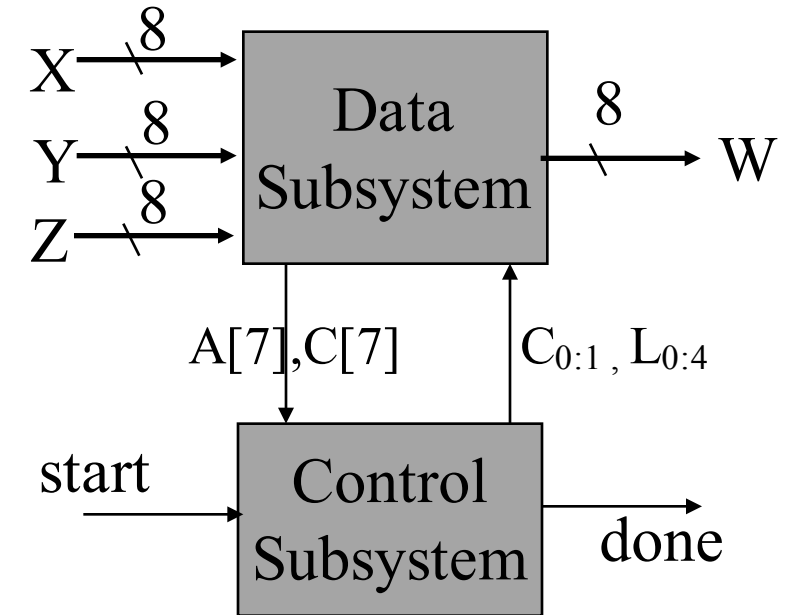
Operation	control
$A \leftarrow \text{Load}(X)$	$C_0=0, L_0=1 \text{ and } L_1=0$
$B \leftarrow \text{Load}(Y)$	$C_1=0 \text{ and } L_2=1$
$C \leftarrow \text{Load}(Z)$	$L_3=1 \text{ and } L_4=0$
$B \leftarrow \text{Add}(B,C)$	$C_1=1 \text{ and } L_2=1$
$A \leftarrow \text{Add}(A,B)$	$C_0=1, L_0=1 \text{ and } L_1=0$
$C \leftarrow \text{Inc}(C)$	$L_3=0 \text{ and } L_4=1$
$A \leftarrow \text{Inc}(A)$	$L_0=0 \text{ and } L_1=1$



Implement the control subsystem

- Map control signals to operations
- **Identify control path components**

```
Alg(X, Y, Z, start, W, done);  
Input X[7:0], Y[7:0], Z[7:0], start;  
Output W[7:0], done;  
Local-object A[7:0], B[7:0], C[7:0];  
S0: If start' goto S0 || done  $\leftarrow$  1;  
S1: A  $\leftarrow$  X || B  $\leftarrow$  Y || C  $\leftarrow$  Z || done  $\leftarrow$  0;  
S2: if (A[7]) goto S5 || B  $\leftarrow$  B+C;  
S3: A  $\leftarrow$  A+B;  
S4: if C'[7] goto S2 || C  $\leftarrow$  Inc(C);  
S5: A  $\leftarrow$  Inc(A) || if C[7] goto S2;  
S6: W  $\leftarrow$  C || goto S0;  
End Alg
```



Implement the control subsystem

state	Instruction	Operation
S1	$A \Leftarrow X$	$A \Leftarrow \text{Load}(X)$
S1	$B \Leftarrow Y$	$B \Leftarrow \text{Load}(Y)$
S1	$C \Leftarrow Z$	$C \Leftarrow \text{Load}(Z)$
S2	$B \Leftarrow B+C$	$B \Leftarrow \text{Add}(B,C)$
S3	$A \Leftarrow A+B$	$A \Leftarrow \text{Add}(A,B)$
S4	$C \Leftarrow \text{Inc}(C)$	$C \Leftarrow \text{Inc}(C)$
S5	$A \Leftarrow \text{Inc}(A)$	$A \Leftarrow \text{Inc}(A)$

control

$C_0=0, L_0=1$ and $L_1=0$

$C_1=0$ and $L_2=1$

$L_3=1$ and $L_4=0$

$C_1=1$ and $L_2=1$

$C_0=1, L_0=1$ and $L_1=0$

$L_3=0$ and $L_4=1$

$L_0=0$ and $L_1=1$

state	Instruction	Operation
S0	$\text{done} \Leftarrow 1$	
S6	$W \Leftarrow C$	Wire

<i>state</i>	<i>L0</i>	<i>L1</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>	<i>c0</i>	<i>c1</i>	done
<i>S0</i>	0	0	0	0	0	-	-	1
<i>S1</i>	1	0	1	1	0	0	0	0
<i>S2</i>	0	0	1	0	0	-	1	0
<i>S3</i>	1	0	0	0	0	1	-	0
<i>S4</i>	0	0	0	0	1	-	-	0
<i>S5</i>	0	1	0	0	0	-	-	0
<i>S6</i>	0	0	0	0	0	-	-	0

Alg(X, Y, Z, start, W, done);

Input X[7:0], Y[7:0], Z[7:0], start;

Output W[7:0], done;

Local-object A[7:0], B[7:0], C[7:0];

S0: If start' goto S0 || $\text{done} \Leftarrow 1$;

S1: $A \Leftarrow X$ || $B \Leftarrow Y$ || $C \Leftarrow Z$ || $\text{done} \Leftarrow 0$;

S2: if (A[7]) goto S5 || $B \Leftarrow B+C$;

S3: $A \Leftarrow A+B$;

S4: if C'[7] goto S2 || $C \Leftarrow \text{Inc}(C)$;

S5: $A \Leftarrow \text{Inc}(A)$ || if C[7] goto S2;

S6: $W \Leftarrow C$ || goto S0;

End Alg

Implement the control subsystem

- Map control signals to operations
- Identify control path components
- **Express the control subsystem with FSM**

Alg(X, Y, Z, start, W, done);

Input X[7:0], Y[7:0], Z[7:0], start;

Output W[7:0], done;

Local-object A[7:0], B[7:0], C[7:0];

S0: If start' goto S0 || done $\leftarrow$ 1;

S1: A $\leftarrow$ X || B $\leftarrow$ Y || C $\leftarrow$ Z || done $\leftarrow$ 0;

S2: if (A[7]) goto S5 || B $\leftarrow$ B+C;

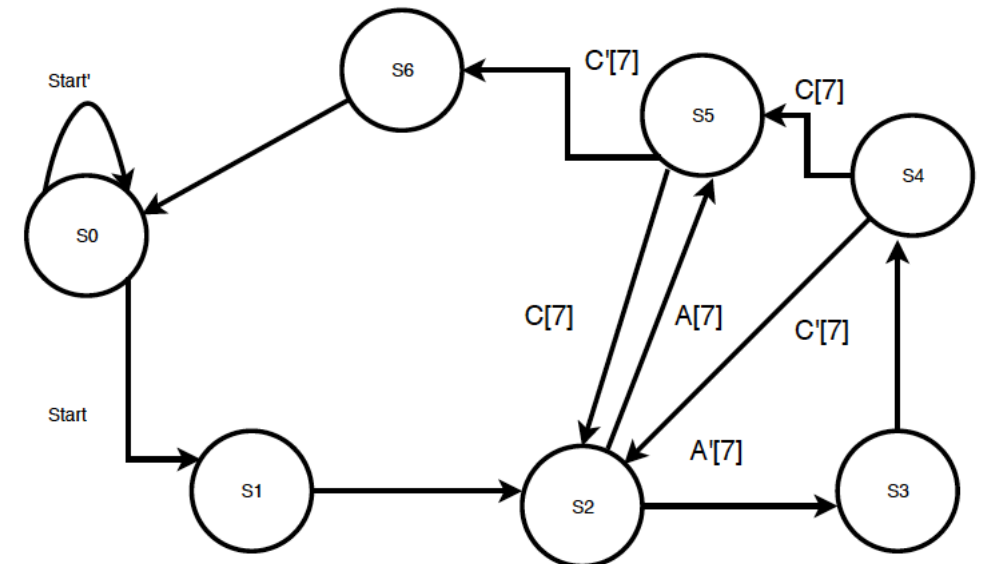
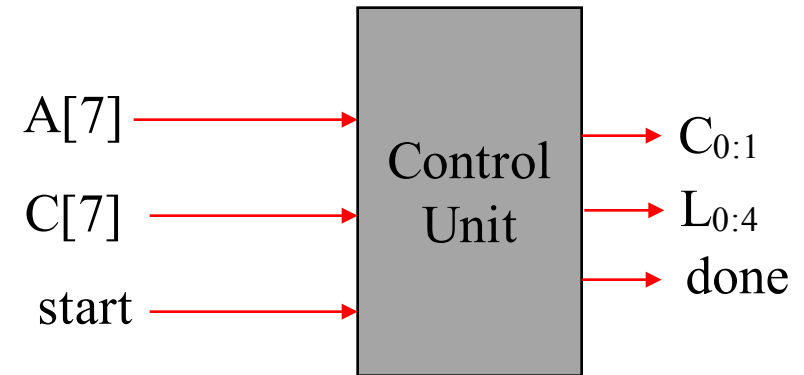
S3: A $\leftarrow$ A+B;

S4: if C'[7] goto S2 || C $\leftarrow$ Inc(C);

S5: A $\leftarrow$ Inc(A) || if C[7] goto S2;

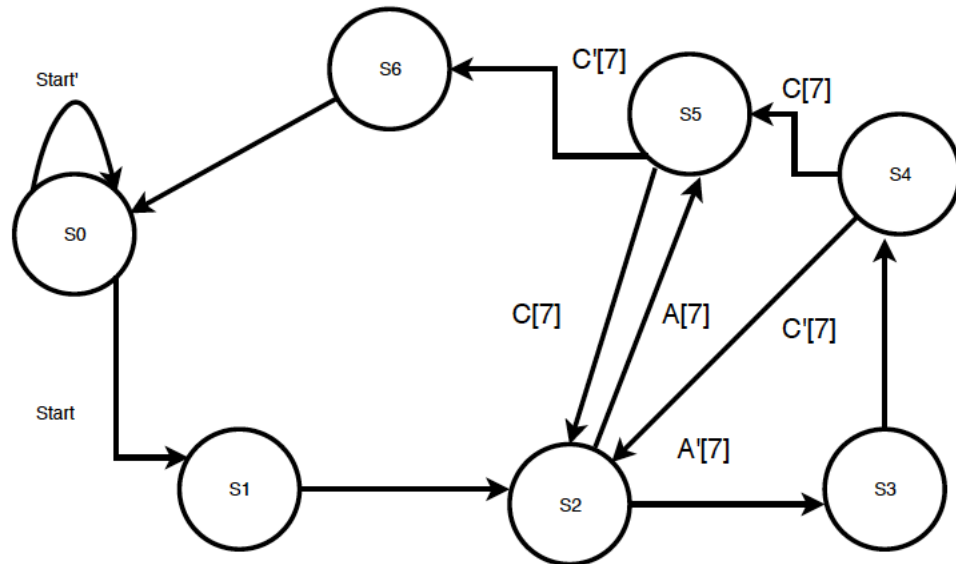
S6: W $\leftarrow$ C || goto S0;

End Alg



Implement the control subsystem

- Map control signals to operations
- Identify control path components
- Express the control subsystem with FSM
- **Implement the control subsystem**



How to implement the state assignment?

State Assignment

Binary	$b_2b_1b_0$
S0	000
S1	001
S2	010
S3	011
S4	100
S5	101
S6	110
S7	111

Gray	$b_2b_1b_0$
S0	000
S1	001
S2	011
S3	010
S4	110
S5	111
S6	101
S7	100

One Hot	$b_7b_6b_5b_4b_3b_2b_1b_0$
S0	0 0 0 0 0 0 0 1
S1	0 0 0 0 0 0 1 0
S2	0 0 0 0 0 1 0 0
S3	0 0 0 0 1 0 0 0
S4	0 0 0 1 0 0 0 0
S5	0 0 1 0 0 0 0 0
S6	0 1 0 0 0 0 0 0
S7	1 0 0 0 0 0 0 0

One Hot Encoding: n bits for n states. Bit $i=1$ for state i .

Control Subsystem: One-Hot State Machine Design

Input: State Diagram

1. Use a flip flop to replace each state.

Set the flip flop which corresponds to the initial state and reset the rest flip flops.

2. Use an OR gate to collect all inward edges.

3. Use a Demux to distribute the outward edges.

Implement the control subsystem

- Map control signals to operations
- Identify control path components
- Express the control subsystem with FSM
- **Implement the control subsystem**

