THE OHIO STATE UNIVERSITY

COLLEGE OF ENGINEERING

ECE 3561
Advanced Digital Design
Class 31: System Controller Design 1

Drew Phillips

Spring 2024

Sequential Circuit Design Process

1) **State / Output Diagram / Table**
2) **Minimization of Number of States**
3) **State Variable Assignment**
4) Transition / Output Table
5) Selection of Flip Flop Types
6) Excitation Table
7) Excitation Equations
8) Output Equations
9) Logic Diagram

System Controller Design

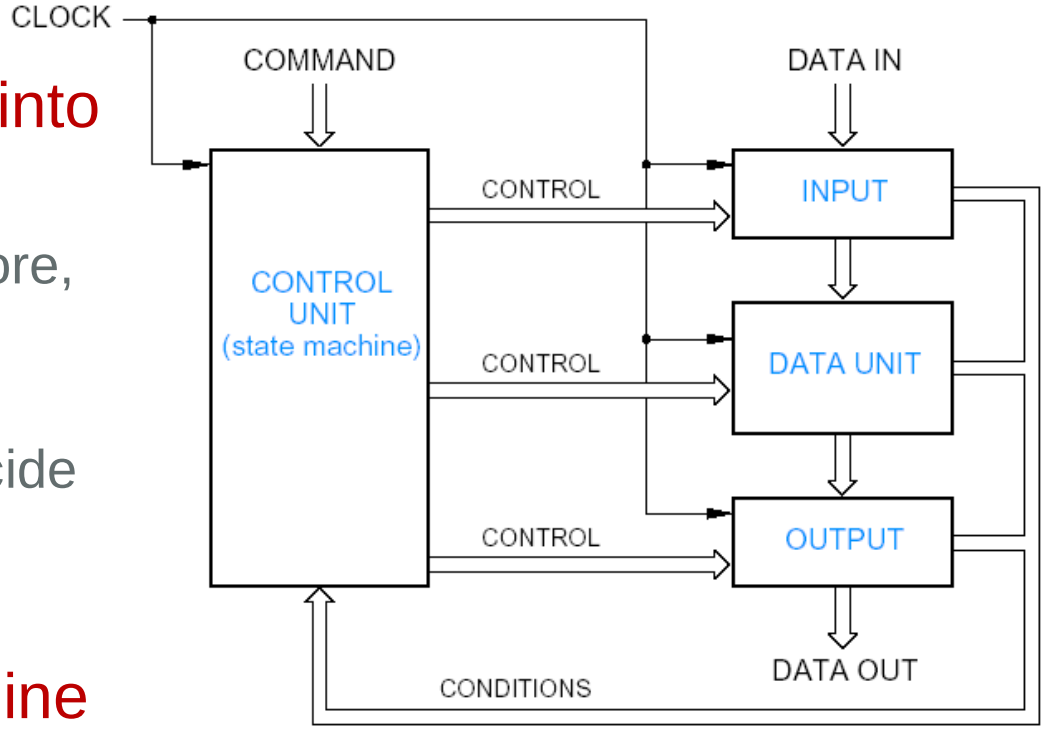
- We will use our sequential circuit design process and apply it to design system controllers for more complicated circuits
- Our system controller design process is like the *opposite* of our system controller analysis process

Recall System Controllers

- Control Logic Building Blocks (LBBs) and other combinational / sequential circuitry
- When analyzing or designing, we can isolate the system controller from the rest of the circuit

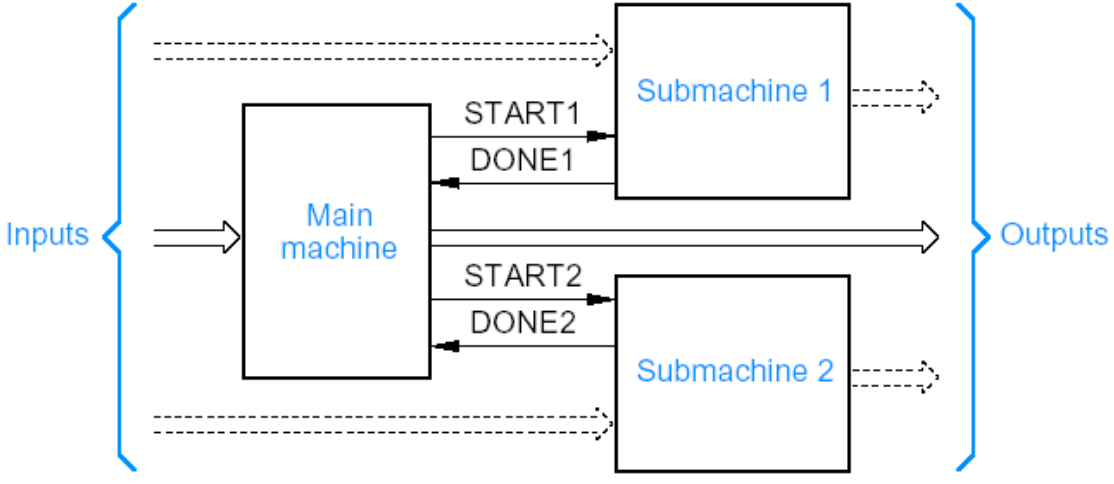
Recall System Controllers

- Generally we split synchronous systems up into two parts:
  - Data Unit: process data (store, route, combine)
  - Control Unit: start and stop actions, test conditions, decide what to do next
- Only the Control Unit is designed as a state machine



Recall System Controllers

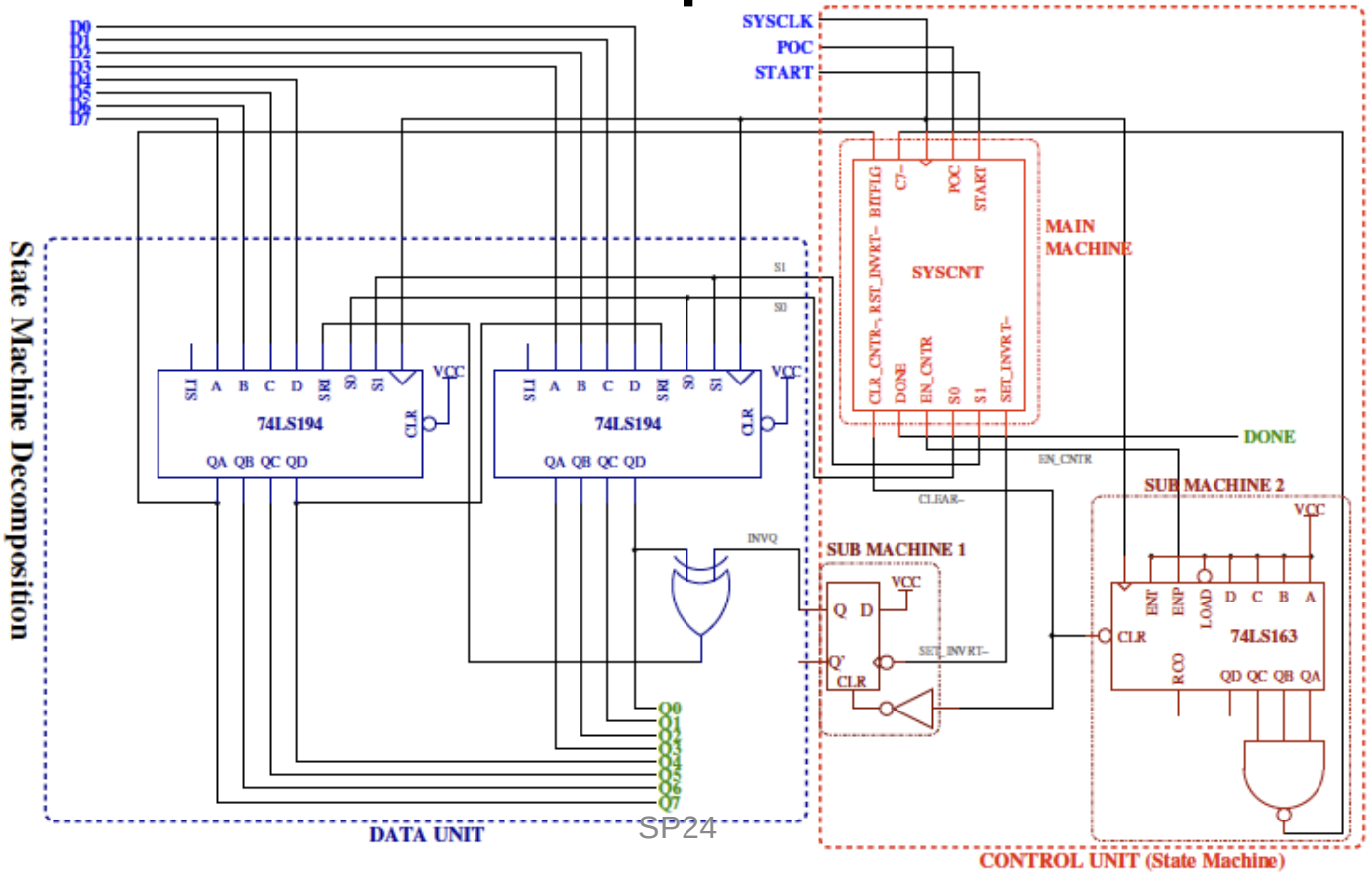
- Control Unit may be further partitioned:
  - Main machine: system controller
  - Sub machines: counter, FF, etc.



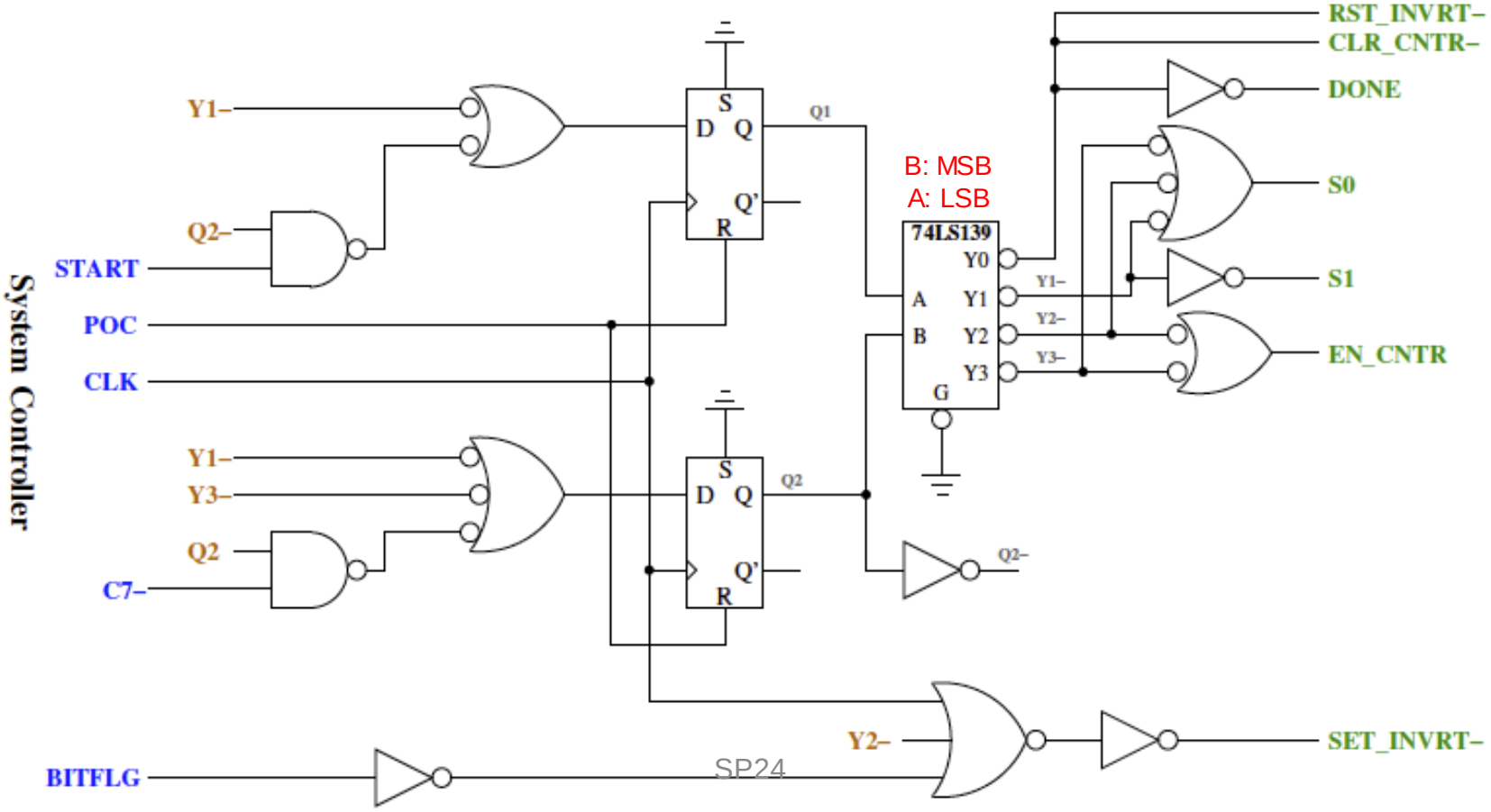
Recall System Controller Analysis

- Isolate the system controller from the rest of the circuit
- System Controller Analysis:
  1) Identify the input/output signals of the system controller
  2) Derive the excitation equations and output equations
  3) Construct a state/output table
     - The entries may have **variables**
  4) Construct a state diagram

Recall Two's Complement Machine



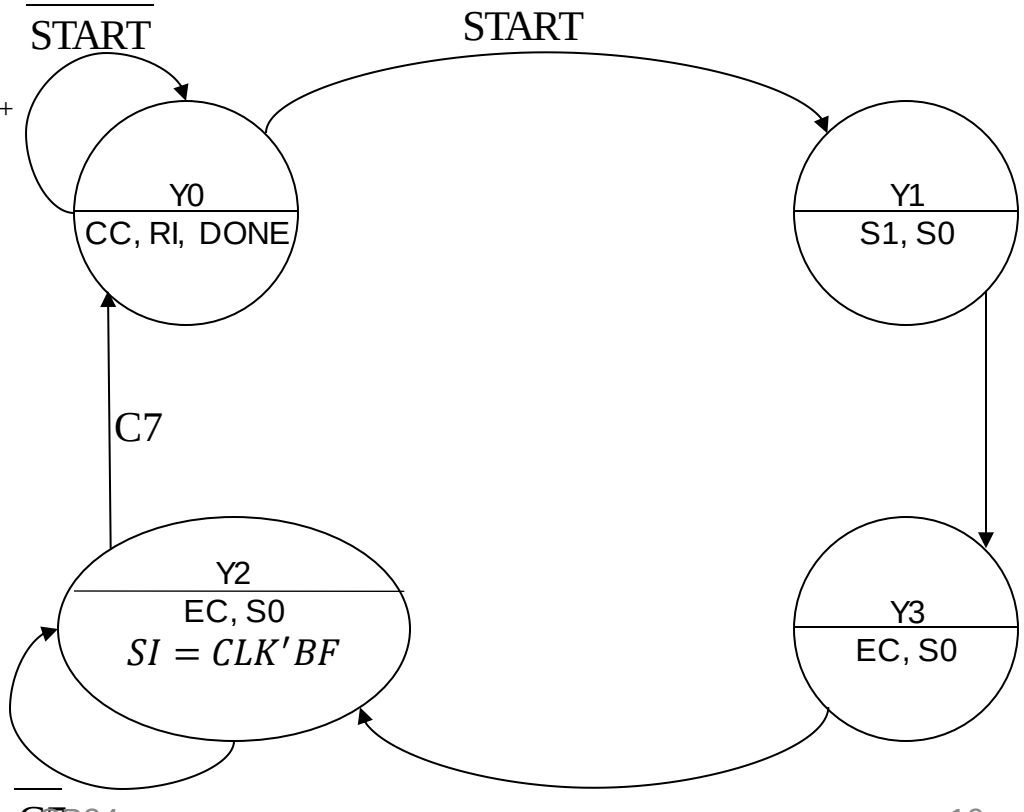
Recall Two's Complement Machine System Controller



Recall Two's Complement Machine System Controller

	P.S.		Outputs							N.S.	
	Q2	Q1	EC	CC	S1	S0	RI	SI	DONE	$Q2^+$	$Q1^+$
Y0	0	0	0	1	0	0	1	0	1	0	ST
Y1	0	1	0	0	1	1	0	0	0	1	1
Y3	1	1	1	0	0	1	0	0	0	1	0
Y2	1	0	1	0	0	1	0	↑	0	$\overline{C_7}$	0

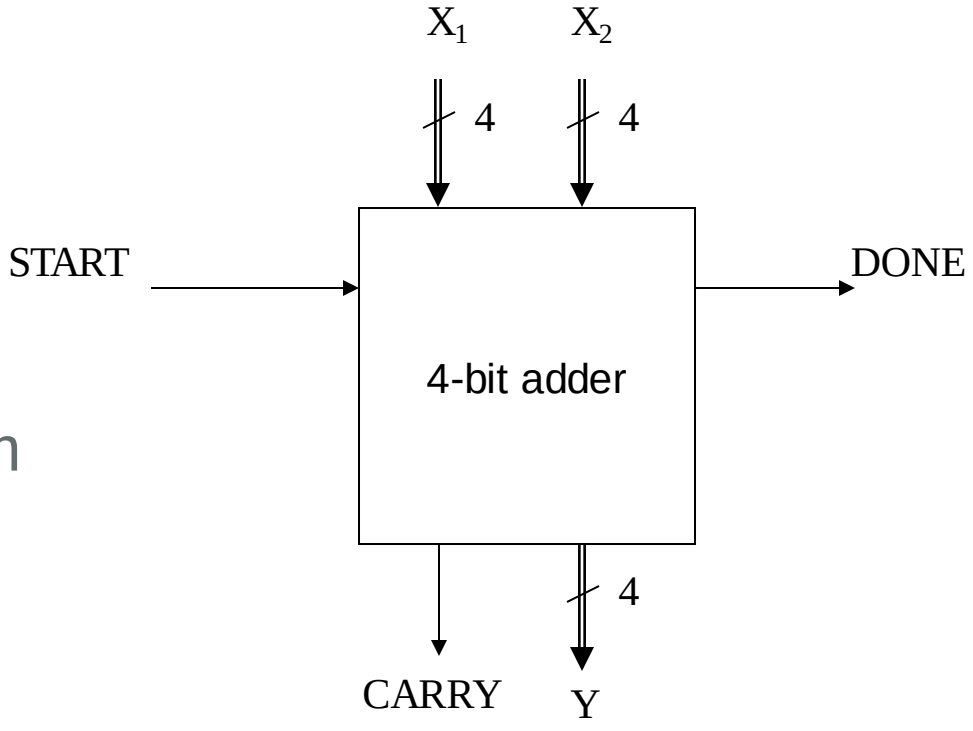
$CLK'BF$



What functionality does this state diagram represent?
Y0: wait for start signal; the previous conversion is done
Y1: load (S1=S0=1) the binary number at shift register input
Y3: right shift (S0=1) the first bit and enable counter (BF is not valid)
Y2: right shift the remaining bits, but also check BF to set inverter

System Controller Design: 4-Bit Binary Serial Adder

- Design a 4-bit binary serial adder with:
  - Parallel input feed of X1 and X2 (4-bit numbers)
  - Output feed of Y (4-bit sum of X1 and X2)
  - Final carry available
  - Start with START and assert DONE when done



X_1 plus X_2 → Y

System Controller Design: 4-Bit Binary Serial Adder

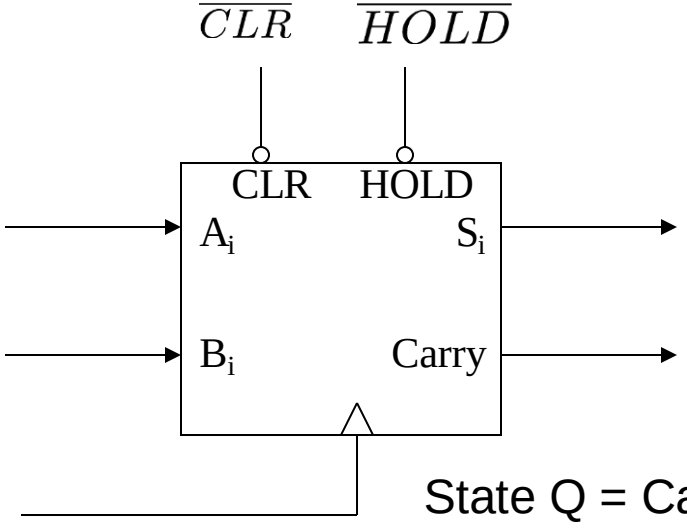
- Let's look at an example of 4-bit binary serial addition to understand the design
- What are all the functions we need LBBs or a system controller to do?

```
    0101 (5)
+   0001 (1)
_____________
       0      c=1
     1        c=0
   1          c=0
 0            c=0
_____________
    0110 (6)  c=0
```

System Controller Design: 4-Bit Binary Serial Adder

- What components/LBBs do we need to do these functions?

Recall 1-Bit Binary Serial Adder (PRE)

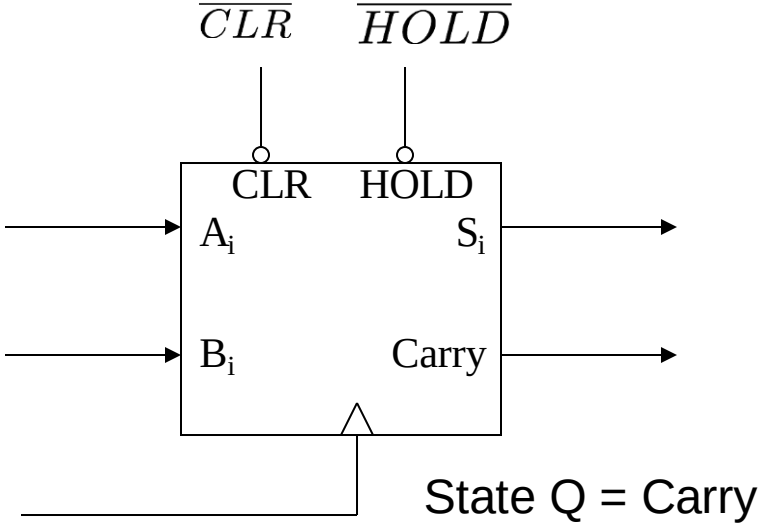


Q	HOLD	A_i	B_i	Q^+	S_i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	X	X	0	X
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	X	X	1	X

CLR – clear the previous carry synchronously

HOLD – hold the previous value of the carry

Recall 1-Bit Binary Serial Adder (PRE)



CLR – clear the previous carry synchronously

HOLD – hold the previous value of the carry

System Controller Design: 4-Bit Binary Serial Adder

conceptually

circuit in reality

0101 (5)
+ 0001 (1)

0110 (6)

Relative movement

1-bit full adder
moves to higher bits

Sate	Carry	Reg 1	Reg 2	Counter
a	-	-	-	0
b	-	-	-	0
c	0	0101	0001	1
c	1	0010	0000	2
c	0	0001	1000	3
c	0	0000	1100	4
d	0	0000	0110	5

Shift to right

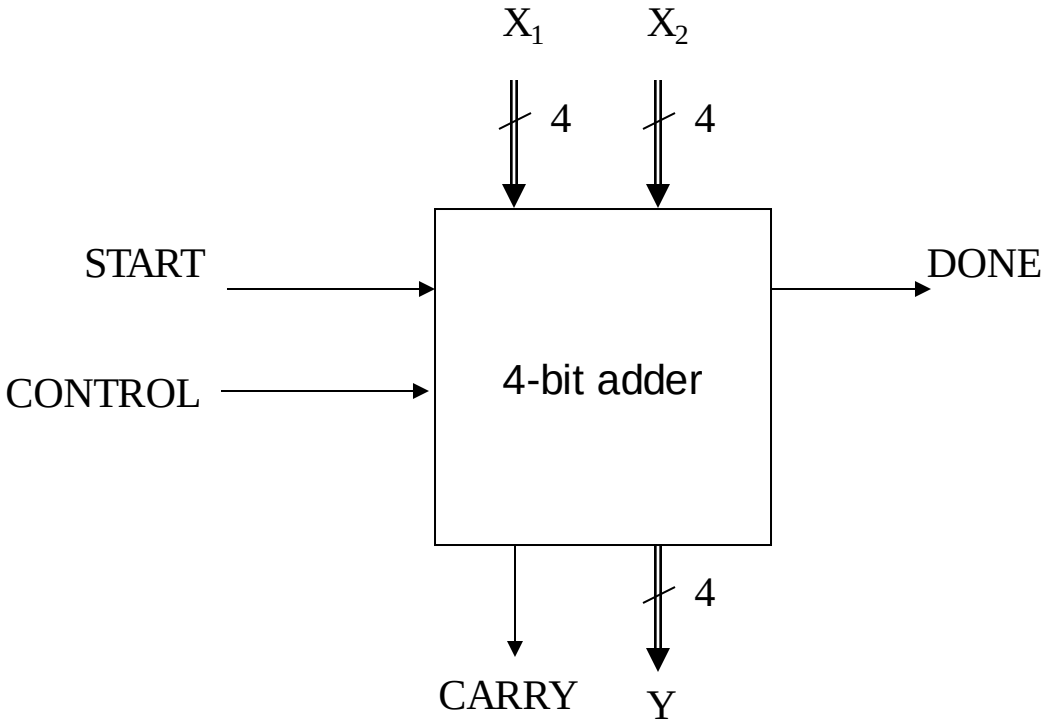
+

Fix the full adder

The sum of each pair of bits and carry gets right shifted into the X2 register until the final answer is in X2 register.

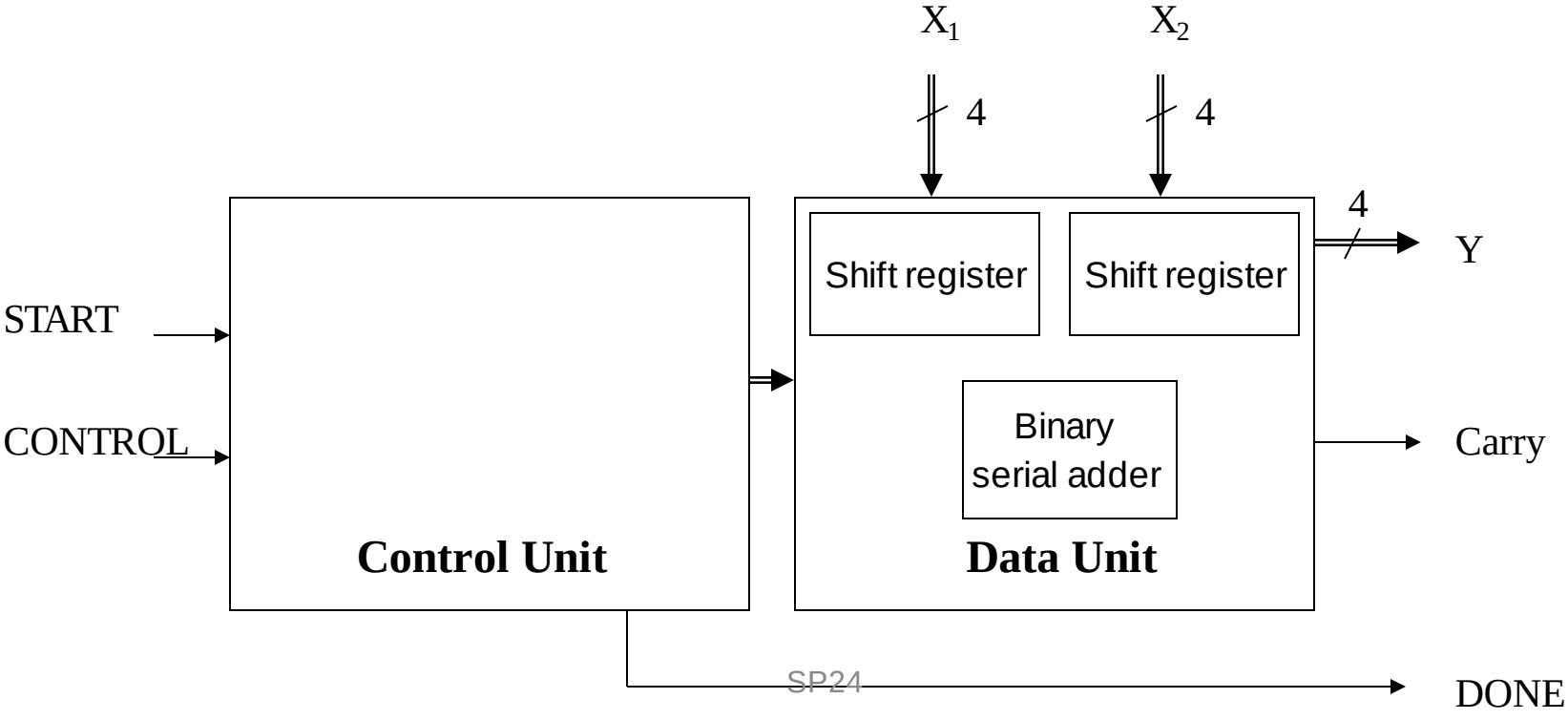
System Controller Design: 4-Bit Binary Serial Adder

- Additional considerations:
  - Use a counter to determine when Y comes out
  - START may be long: user will negate START only after detecting DONE
  - CONTROL = 1: $X_1+X_2 \rightarrow Y$
    CONTROL = 0: $X_1+Y \rightarrow Y$

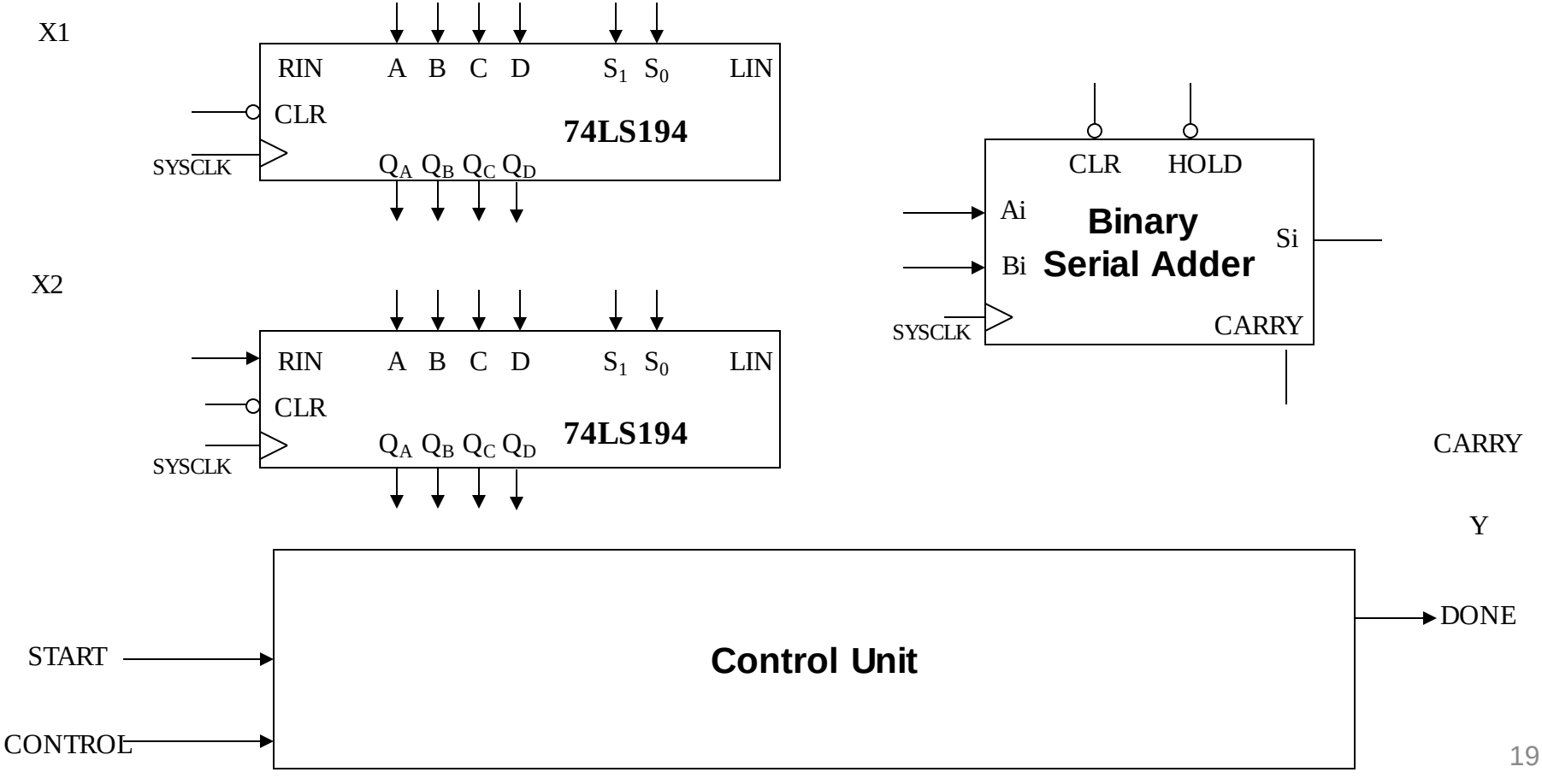


System Controller Design: 4-Bit Binary Serial Adder

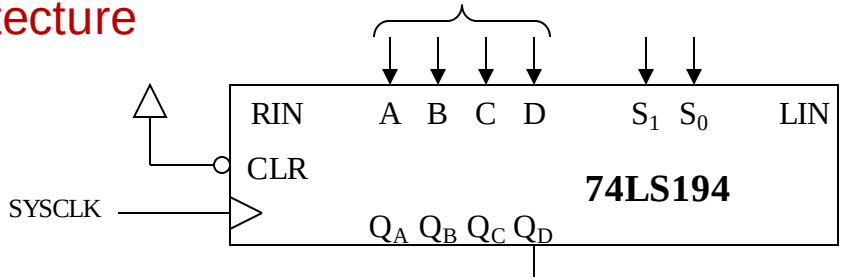
- Data Unit and Control Unit:

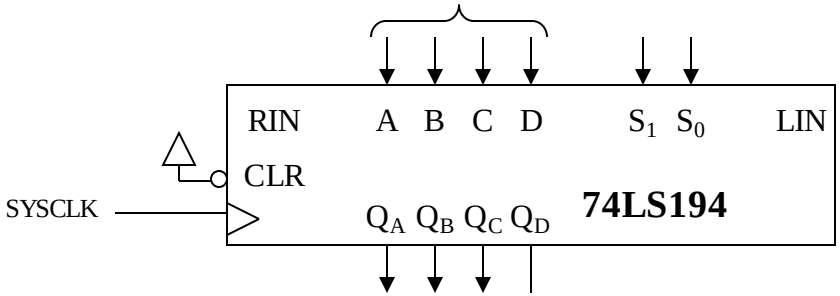


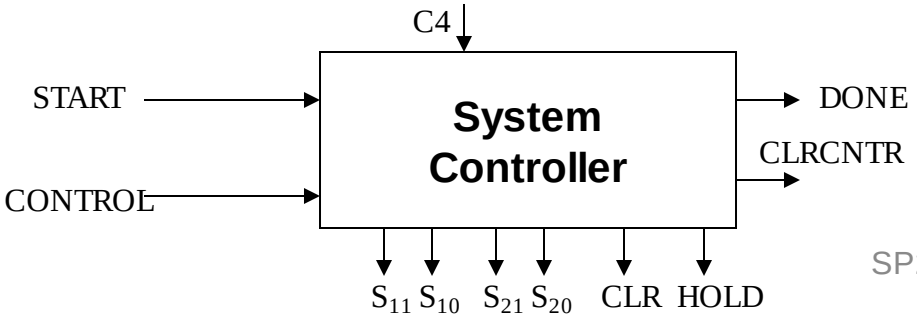
Architecture



Architecture







Can you make all the connections?