Summary

Flip flops v. latches
- •Flip flops are edge triggered
- •Latches are level sensitive

Many variants
- •S-R, J-K, D, T latch/flip-flop
- •D-type most useful in processor design

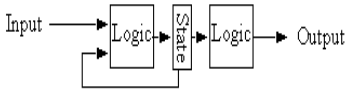
Beware of race conditions
- •Need edge triggered device in any feed-back path

State machines design

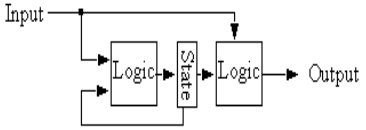
Interesting state machines have inputs and outputs.

State will change depending on input.

Two classes are Mealy and Moore:
- •Moore machine: output = f(state)



Mealy machine: output = f(state,input)



Generally: Moore machines have more states (mnemonic), but are easier to understand.

Design Process

General Procedure
- •Step 1: Create a State Diagram
- •Step 2: Write down a State Transition Table
- •Step 3: Figure out the inputs to the flip flops using the excitation table.
- •Step 4: Figure out functions for input to flip flops
- •Step 5: Implement the machine

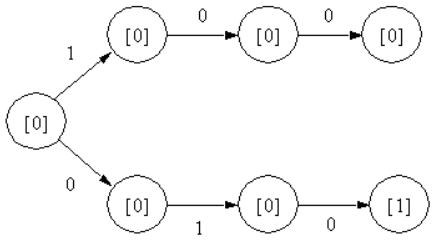
Example
- •"Write a string recognizer that recognizes the pattern 010 in its input.
- •When an input bit pattern is recognized, output a 1.
- •When the sequence 100 is seen, output zeros thereafter until a reset is asserted"
- •Examples:

X: 001010100010
Z: 000101010000

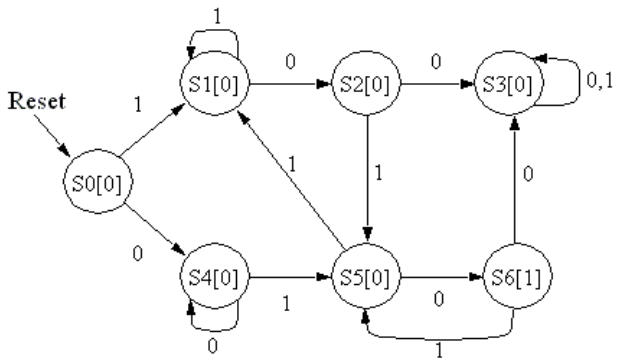
X: 0110110100100
Z: 0000000010000

1. State diagram creation

Start with easy paths, fill in the rest, reuse as much as possible!
This is a ***Moore*** machine:



2. Final state diagram



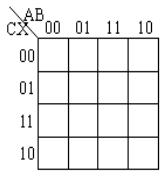
3. Symbolic State Transition Table:

State Number	Current State	Next (X=0)	Next (X=1)	Output
	S0			
	S1			
	S2			
	S3			
	S4			
	S5			
	S6			

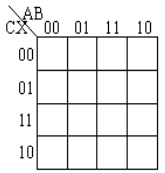
4. Figure out flip flop inputs

A	B	C	X	A'	B'	C'
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

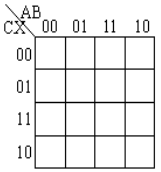
5. Reduce using K-maps



A' =



B' =

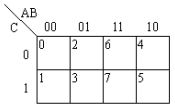


C' =

6. Output Logic

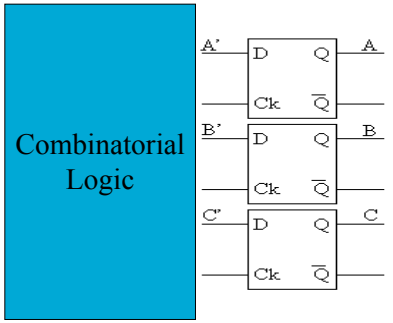
A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	X

Karnaugh map



O =

6. Final Circuit



State minimization

Goal: to minimize the number of states in a state diagram.

- Basic idea: Identify and combine states with equivalent behavior.
- 2 states are equivalent if the output is the same and, for each input combinations, the next state is the same state or an equivalent state.

Approach:

- Start with state transition table.
- Identify states with the same behavior
- If such states go to the same next state, combine them and rename each occurrence of the old state in the state table.
- Repeat with new state table until no new combinations are possible.

Next Time : Minimization

State Transition Table

Prefix	Name	X=0	X=1	Output (X=0)	Output (X=1)
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S7	S8	0	0
01	S4	S9	S10	0	0
10	S5	S11	S12	0	0
11	S6	S13	S14	0	0
000	S7	S0	S0	0	0
001	S8	S0	S0	0	0
010	S9	S0	S0	0	0
011	S10	S0	S0	1	0
100	S11	S0	S0	0	0
101	S12	S0	S0	1	0
110	S13	S0	S0	0	0
111	S14	S0	S0	0	0

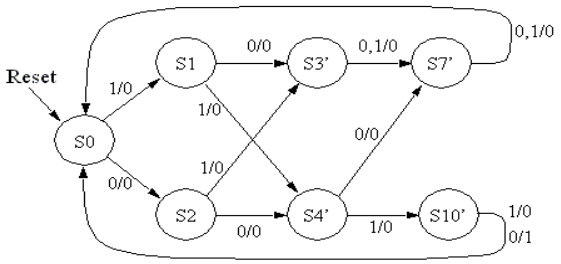
Modified State Transition Table

Prefix	Name	X=0	X=1	Output (X=0)	Output (X=1)
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S7’	S7’	0	0
01	S4	S7’	S10’	0	0
10	S5	S7’	S10’	0	0
11	S6	S7’	S7’	0	0
00x, 010 100 11x	S7’	S0	S0	0	0
011 101	S10’	S0	S0	1	0

Modified State Transition Table (2nd iteration)

Prefix	Name	X=0	X-1	Output (X=0)	Output (X=1)
Reset	S0	S1	S2	0	0
0	S1	S3’	S4’	0	0
1	S2	S4’	S3’	0	0
11	S3’	S7’	S7’	0	0
01, 10	S4’	S7’	S10’	0	0
00x, 010 100 11x	S7’	S0	S0	0	0
011 101	S10’	S0	S0	1	0

State Diagram:



Summary

Components
- •Flip flops (edge triggered)
- •Latches (usually level triggered).

Designing Sequential Circuits
- •Step 1: Create a State Diagram
- •Step 2: Write down a State Transition Table
- •Step 3: Do state minimization
- •Step 4: Do state assignment
- •Step 5: Figure out the inputs to the flip flops using the excitation table.
- •Step 6: Figure out functions for input to flip flops