ITERATIVE COMPUTIONS

- COMBINATIONAL DESIGN

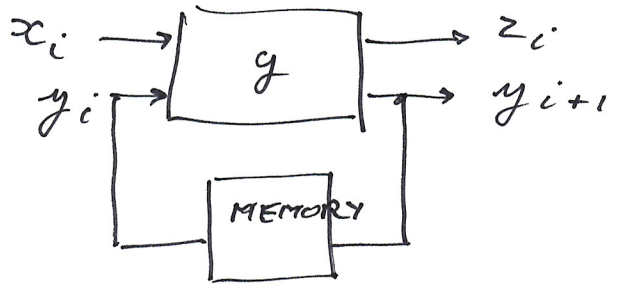
$$z = f(x)$$

- SEQUENTIAL DESIGN

$$x = (x_0, x_1, x_2 \ldots)$$

$$\begin{aligned} \text{IT1} \quad & (z_0, y_1) = g(x_0, y_0) \\ \text{IT2} \quad & (z_1, y_2) = g(x_1, y_1) \\ \text{IT3} \quad & (z_2, y_3) = g(x_2, y_2) \end{aligned}$$

$$z = (z_0, z_1, z_2, \ldots)$$



MEALY

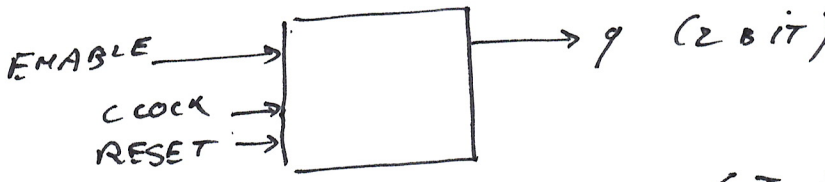
MOORE

MEALY AUTOMATON.

$$M = \langle I, Q, S, G, H \rangle$$

- I: INPUT
- Q: OUTPUT
- S: STATES
- G: STATE TRANSITION FUNCTION
- H: OUTPUT FUNCTION

TWO-BIT ENABLED COUNTER



$\langle I, Q, S, G, H \rangle$

I: enable $\{0, 1\}$

Q: q $\{00, 01, 10, 11\}$

S: S: $\{S0, S1, S2, S3\}$

G: $S \times I \to S$

$G0: \{(S0,S0), (S1,S1), (S2,S2), (S3,S3)\}$

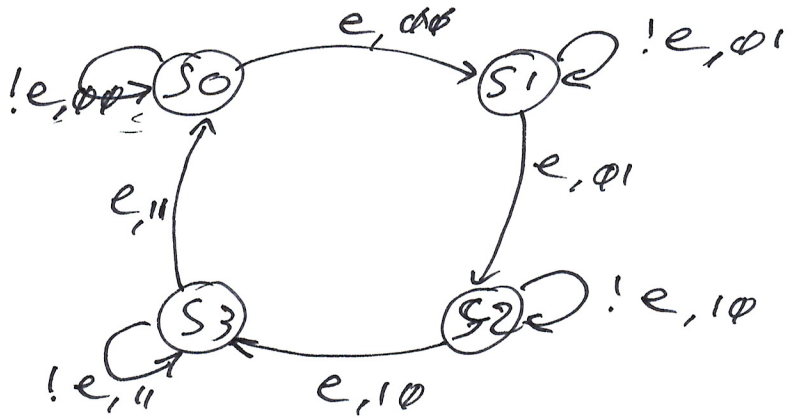
$G1: \{(S0,S1), (S1,S2), (S2,S3), (S3,S0)\}$

H: $S \times I \to Q$

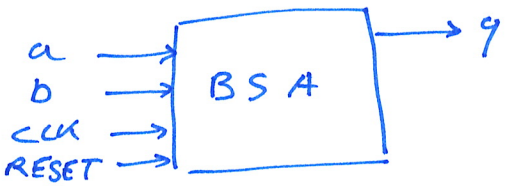
$H0: \{(S0,00), (S1,01), (S2,10), (S3,11)\}$

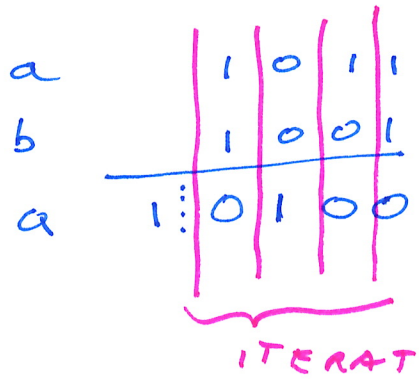
$H1: \{(S0,00), (S1,01), (S2,10), (S3,11)\}$

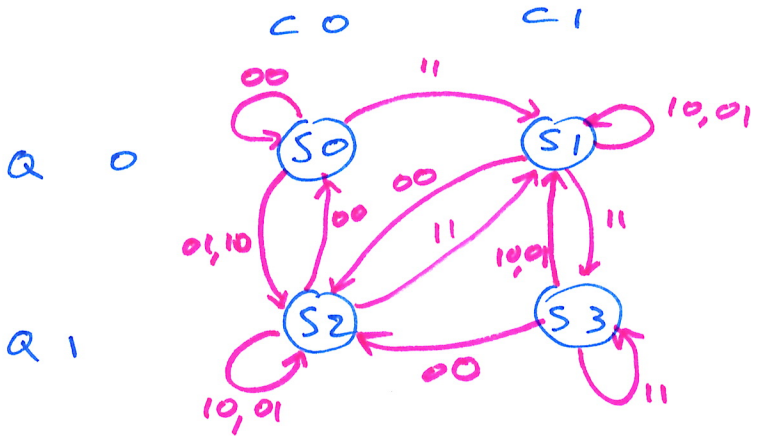
STATE TRANSITION GRAPH

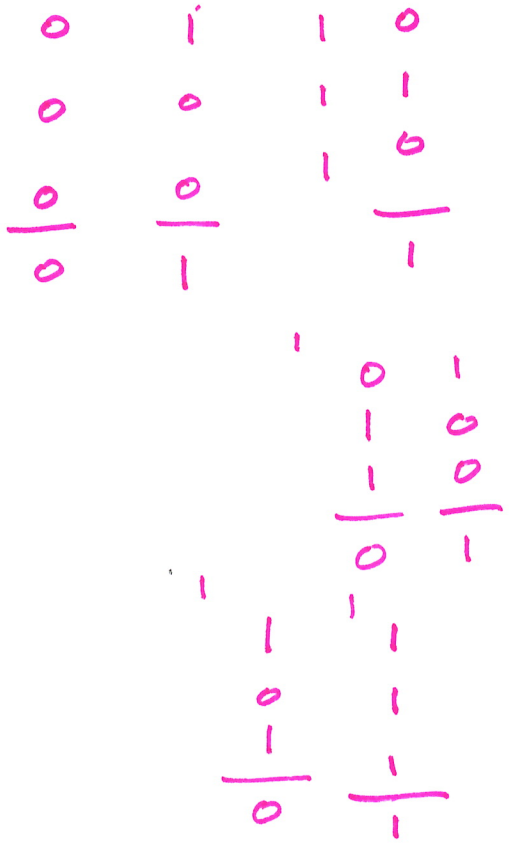


BIT-SERIAL ADDER



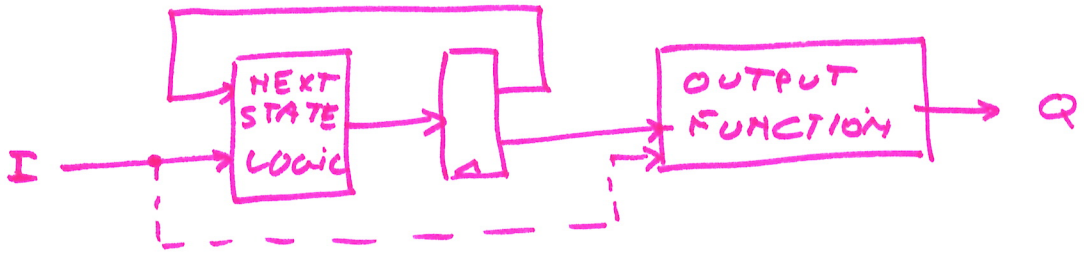






OUTPUT LABELING:

VERILOG IMPLEMENTATION



STEPS

1. CREATE STATE TRANSITION GRAPH.
2. CHOOSE STATE ENCODING
3. EXTRACT NEXT-STATE LOGIC
4. EXTRACT OUTPUT LOGIC

VERILOG → QUARTUS (2), (3), (4)