Practice 2:

Digital Systems

Boolean Algebra

Boolean Algebra – Basic Identities

- Additive

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + \overline{A} = 1$$

- Multiplicative

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot \overline{A} = 0$$

Boolean Algebra – Basic Identities

- Some Important Identities

$$A \cdot \overline{A} = 0$$

$$A \cdot A = A = A + A$$

$$A + \overline{A} = 1 = 1 + x$$

$$A + \overline{A}B = A + B$$

$$\overline{A} + AB = \overline{A} + B$$

$$\overline{AB} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Boolean Algebra – Function Minimization

- Minimize the following function:

$$f=\left(\bar{A}+AB\bar{C}\right)+\left(A+\bar{A}\bar{B}C\right)\left(\overline{A\left(\bar{A}+\bar{B}+C\right)}\right)$$

$$\overline{A\left(\bar{A}+\bar{B}+C\right)}=\bar{A}+\overline{\bar{A}+\bar{B}+C}\quad=\bar{A}+AB\bar{C}\quad=\bar{A}+B\bar{C}$$

$$=\left(\bar{A}+B\bar{C}\right)+\left(A+\bar{B}C\right)\left(\bar{A}+B\bar{C}\right)=$$

$$=\left(\bar{A}+B\bar{C}\right)\left(1+\left(A+\bar{B}C\right)\right)$$

$$=\bar{A}+B\bar{C}$$

Boolean Algebra – Complementation

- DeMorgan's Theorem:
  - NAND: $f = \overline{A \cdot B} \quad \Rightarrow \quad f = \bar{A} + \bar{B}$
  - NOR: $f = \overline{A + B} \quad \Rightarrow \quad f = \bar{A} \cdot \bar{B}$
- Find the complementary expression for $f = A + BC + \bar{A}\bar{B}$

$$\bar{f} = \overline{A + BC + \bar{A}\bar{B}}$$

$$= \bar{A} \cdot \overline{BC} \cdot \overline{\bar{A}\bar{B}} \quad = \bar{A} \cdot \left(\bar{B} + \bar{C}\right) \cdot \left(A + B\right)$$

$\bar{A} \cdot \left(A + B\right) = \bar{A}B$

$$= \bar{A}B \cdot \left(\bar{B} + \bar{C}\right)$$

$B \cdot \left(\bar{B} + \bar{C}\right) = B\bar{C}$

$$= \bar{A}B\bar{C}$$

Boolean Algebra – Universality

- A universal set in Boolean Algebra comprises of the following functions:
  - NOT
  - AND or OR

- Several complex gates can independently comprise universal sets, such as:
  - NAND
  - NOR
  - MUX

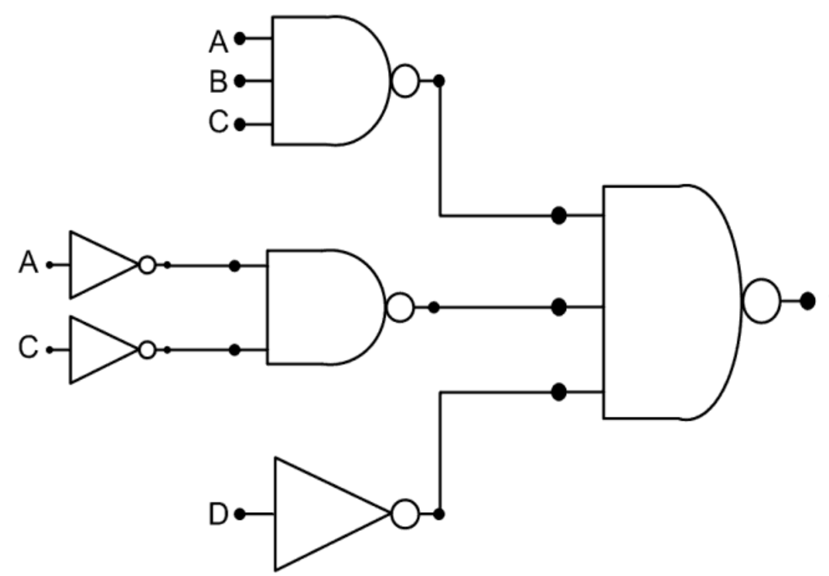
Boolean Algebra – Universality

- Implement the following function using only NAND gates.

$$f = ABC + \bar{A}\bar{C} + D$$

$$\bar{\bar{f}} = \overline{\overline{ABC + \bar{A}\bar{C} + D}}$$

$$= (DeMorgan) = \overline{\overline{ABC} \cdot \overline{\bar{A}\bar{C}} \cdot \bar{D}}$$



Mutliplexers

Multiplexers - Reminder

Multiplexers - Example

- Implement the following function using 4→1 Multiplexers:

$$f\left(A,B,C,D\right)=\sum\left(0,1,5,7,13,14\right)$$

$$f\left(\bar{A}\bar{B}\right)=\bar{C}$$

$$f\left(\bar{A}B\right)=D$$

$$f\left(AB\right)=C\oplus D$$

$$f\left(A\bar{B}\right)=0$$

CD \ AB	00	01	11	10
00	1	0	0	0
01	1	1	1	0
11	0	1	0	0
10	0	0	1	0

Multiplexers

- Implement the following function using 4→1 Multiplexers:

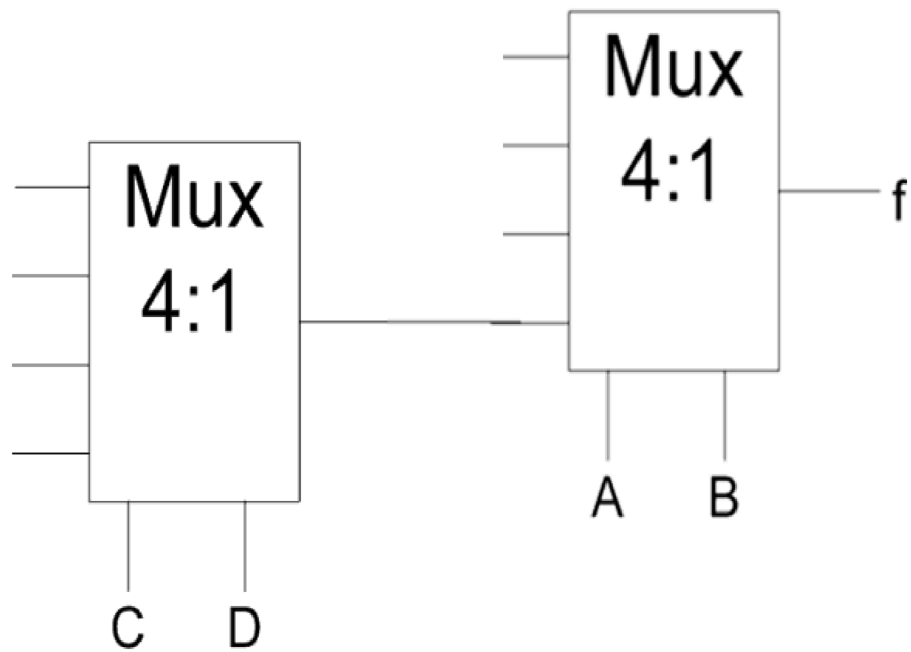
$$f(A,B,C,D)=\sum(0,1,5,7,13,14)$$

$$f(\bar{A}\bar{B})=\bar{C}$$

$$f(\bar{A}B)=D$$

$$f(AB)=C\oplus D$$

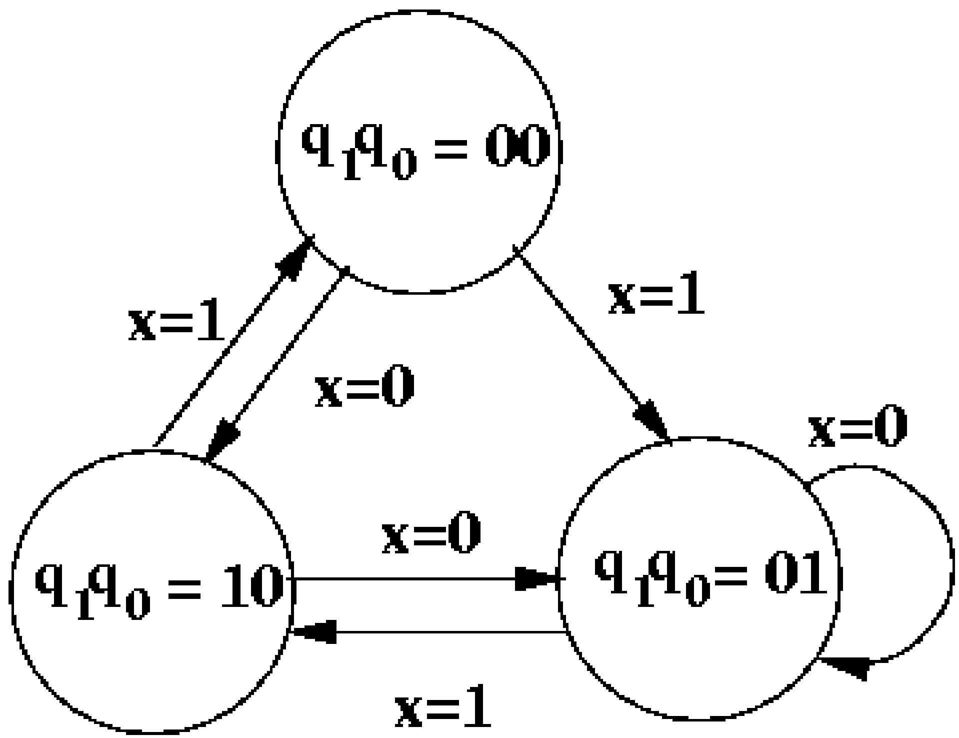
$$f(A\bar{B})=0$$



Finite State Machines

Finite State Machines

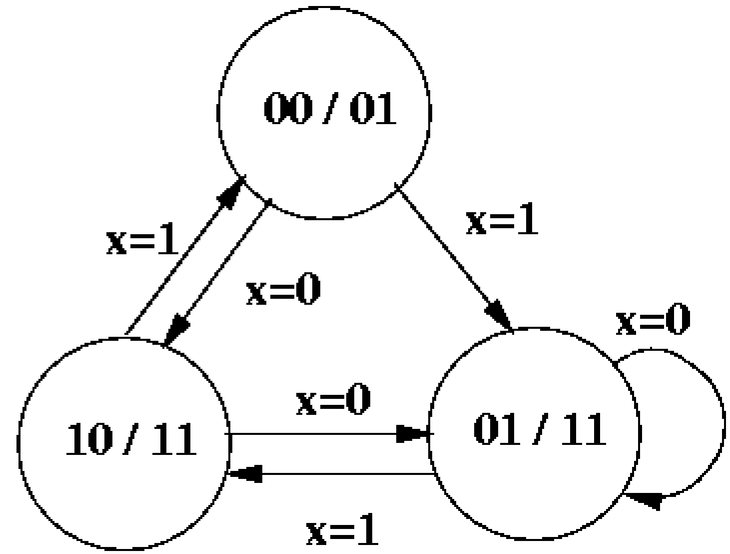
- All digital circuits are constructed using FSMs:



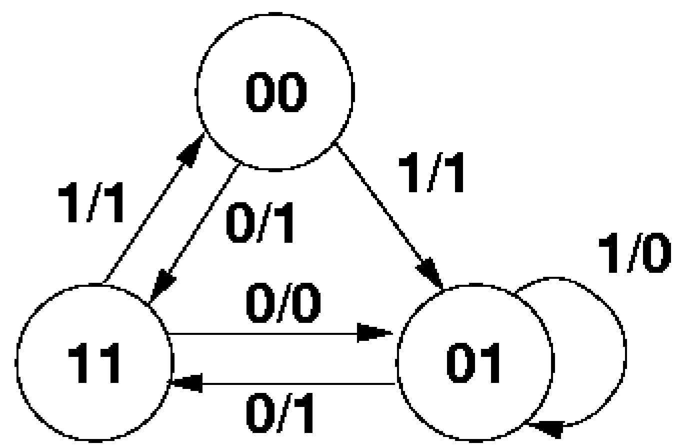
  - q is the *state vector*
  - x is the *input vector*

Finite State Machines

- Moore machines have a predefined output for each state:

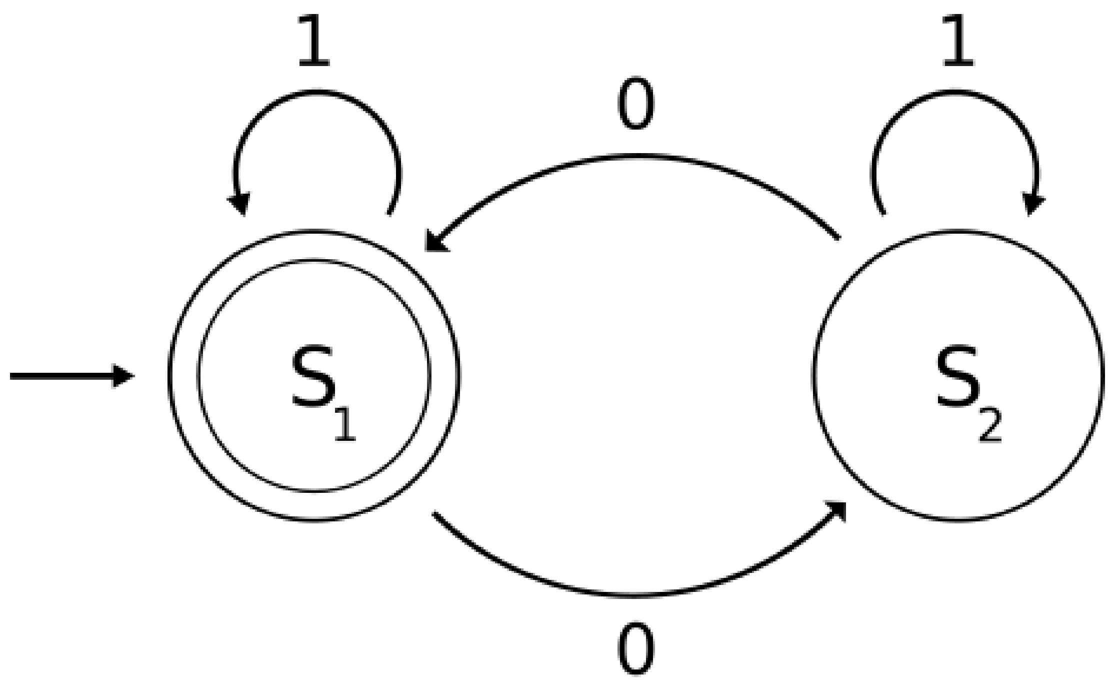


- Mealy machines have an output that is determined by the previous state and the input:



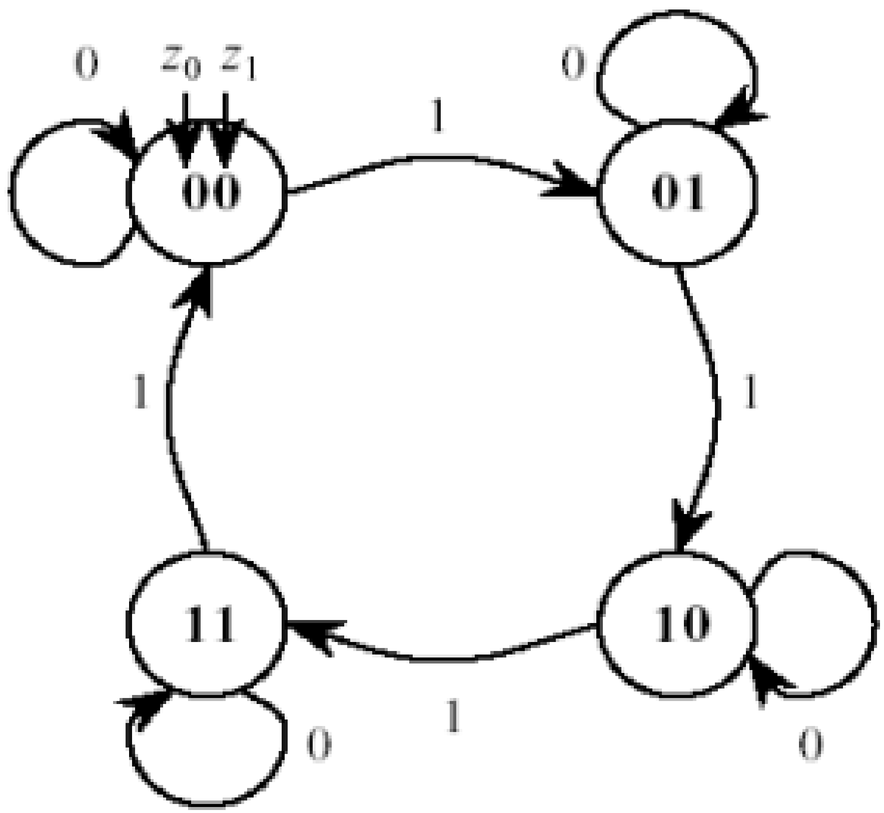
Finite State Machines - Example

- Parity Counter



Finite State Machines – Example 2

- 2-bit Counter



Bonus Qustion

If we have time…

Bonus Question – From Job Interviews

- *Reminder*: Encoder
  - A (one-hot) encoder receives 2^n bits and outputs the binary position of the only *1*.
  - This is the opposite operation of a *Decoder*.

- A "Priority Encoder" or a *Leading Ones Detector*:
  - Outputs the binary position of the first *1* in the input vector.
  - These are often used to select the highest priority *interrupt* if several are asserted at the same time.

Bonus Question – From Job Interviews

- Create a 4-bit Priority Encoder:
  - 4 bit input vector ($I_3I_2I_1I_0$)
  - 2 bit output (O_1O_0) – encoding the place of the leading '1'
  - 1 bit flag (F) – showing that no '1' was found

Bonus Question – From Job Interviews

I3	I2	I1	I0	O1	O0	F
X	X	X	1	0	0	0
X	X	1	0	0	1	0
X	1	0	0	1	0	0
1	0	0	0	1	1	0
0	0	0	0	X	X	1

$$F = \overline{I_3}\,\overline{I_2}\,\overline{I_1}\,\overline{I_0} = \overline{I_3 + I_2 + I_1 + I_0}$$

$$O_1 = \overline{I_1}\,\overline{I_0} = \overline{I_1 + I_0}$$

$$O_0 = \overline{I_0 + I_2\overline{I_1}}$$

Bonus Question – From Job Interviews

- Using 4-bit Priority Encoders, create a 16-bit Priority Encoder.