Discussion

Karnaugh Maps, Minimization, & FSMs

Karnaugh Maps

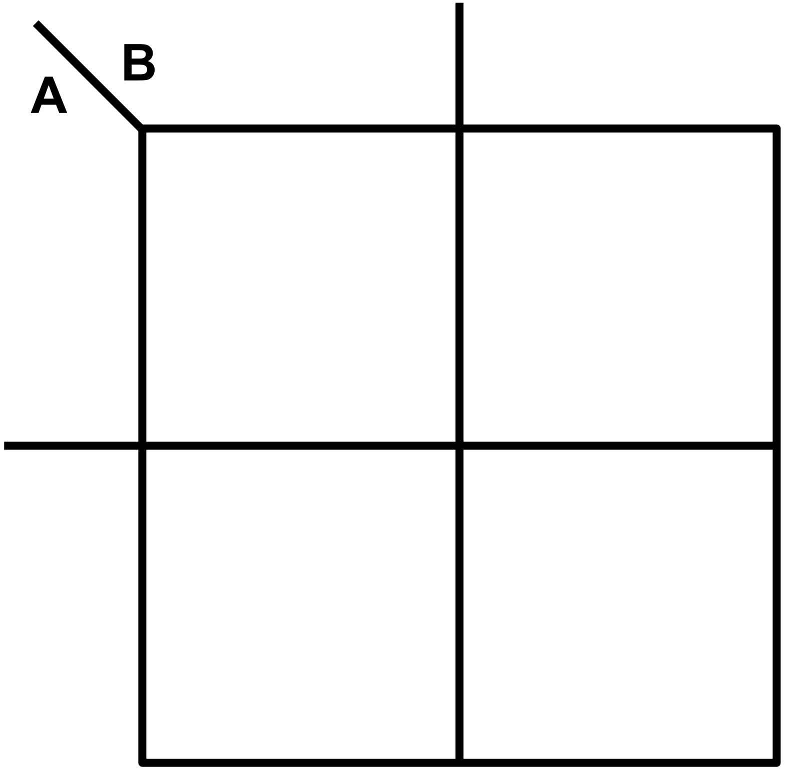
- Visual method for boolean simplification
- Makes use of 2 important boolean identities:

$$\mathbf{(1 + A) = 1} \qquad\qquad \mathbf{(A + \overline{A}) = 1}$$

- Leverages gray coding to organize neighboring minterms
  - Adjacent minterms only differ by a single bit!

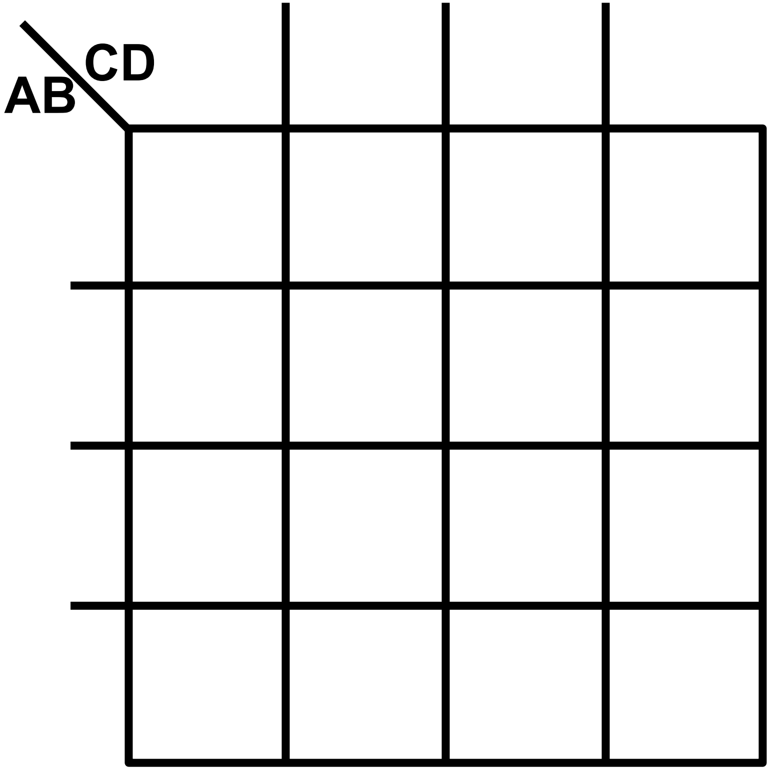
2 variable Karnaugh map

$$F(A,B) = \overline{A}B + \overline{A}$$



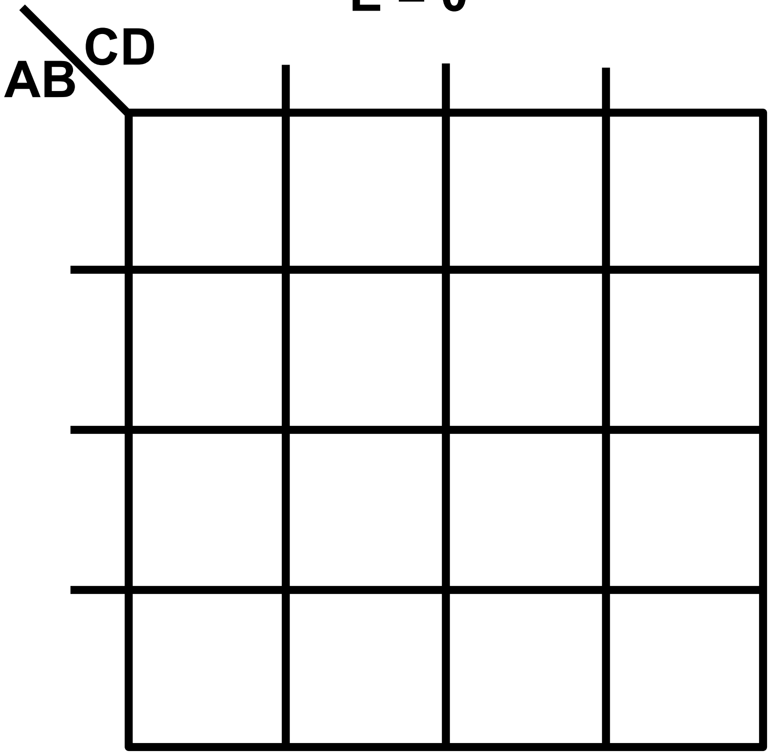
4 variable Karnaugh map

$$
\begin{aligned}
\mathbf{F(A,B,C,D)} = {} & \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \\
& \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + \\
& AB\overline{C}\,\overline{D} + ABC\overline{D} + \\
& A\overline{B}CD + A\overline{B}\,\overline{C}\,\overline{D} + \\
& ABCD
\end{aligned}
$$

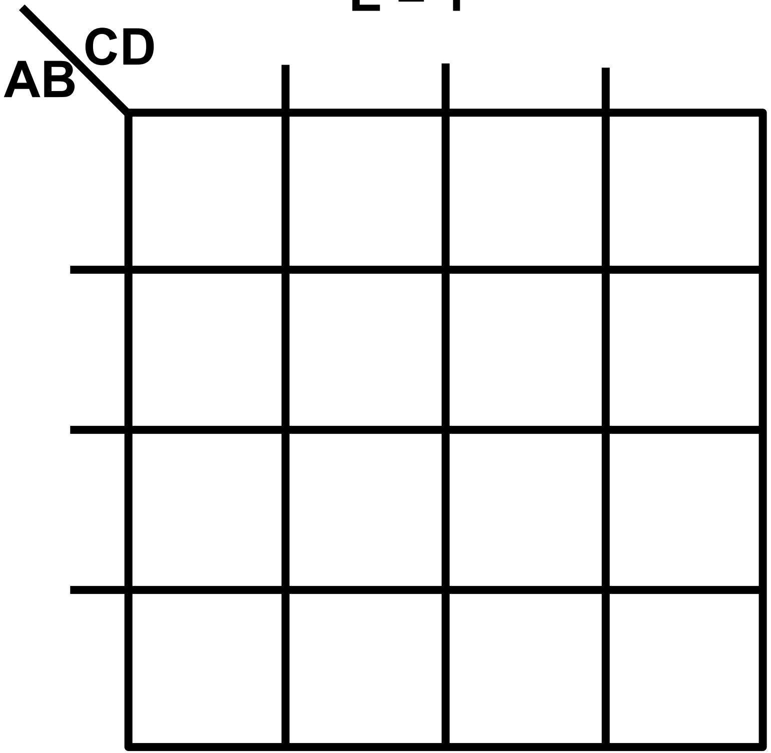


5 Variable Karnaugh Map!! (The Bad Way)

E = 0



E = 1

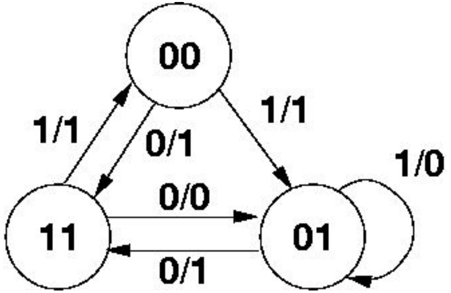


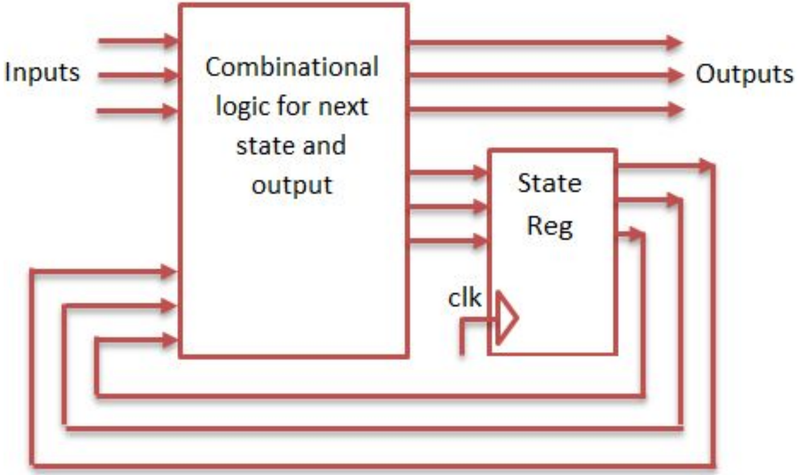
5 Variable Karnaugh Map!! (The Good Way)

CDE AB								

Finite State Machines

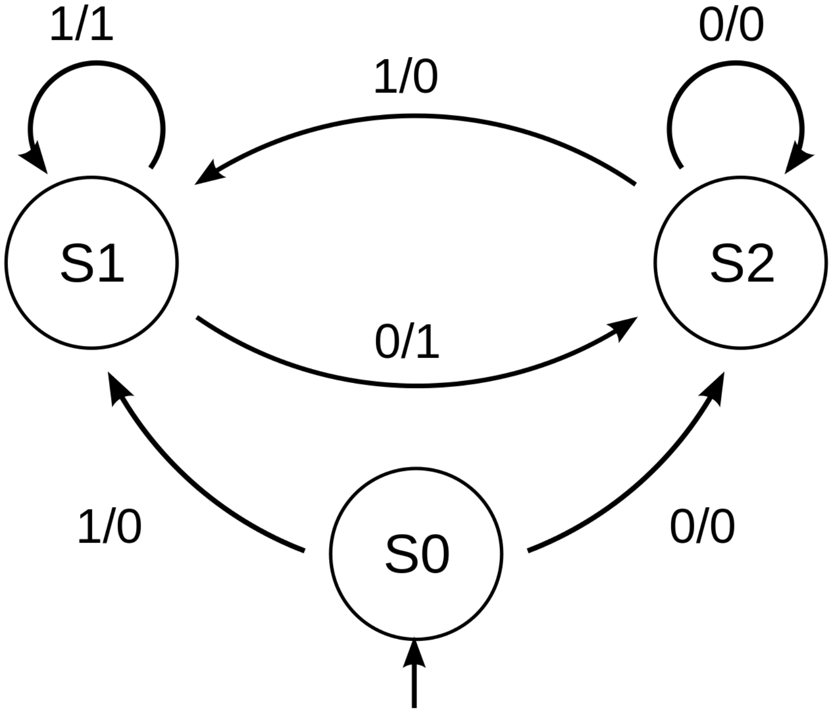
- What are they used for?
- Mealy vs. Moore?
- States are the memory!
- Can be represented in many ways
  - Bubble diagrams
  - Tables
  - Case statements in verilog





Building transition table

Input	State	Next State	Output



Building the circuit

Input	State	Next State	Output
0	S0	S2	0
1	S0	S1	0
0	S1	S2	1
1	S1	S1	1
0	S2	S2	0
1	S2	S1	0

Equations:

S1 =

S2 =

Output =

Building a Bubble Diagram

Input	State	Next State	Output
Timer	Green	Green	N/A
Button	Green	Yellow	N/A
Timer	Yellow	Red	N/A
Timer	Red	Green	N/A