Outline

• Forming Boolean functions
  – Minterms, maxterms, SOP, POS etc.

• Simplifying Boolean functions
  – Implicate, implicate, etc.
  – Karnaugh map

• Implementing Boolean functions
  – Decoders
  – Multiplexers
Definitions for Forming Boolean functions

- **Implicant**: product of literals
  \[ ABC, AC, BC \]
- **Implicate**: sum of literals
  \[ (A+B+C), (A+C), (B+C) \]
- **Minterm**: AND that includes all input variables
  \[ ABC, A′BC, AB′C \]
- **Maxterm**: OR that includes all input variables
  \[ (A+B+C), (A′+B+C), (A′+B′+C) \]

- **Canonical SOP**: the sum of all minterms
- **Canonical POS**: the product of all maxterms
Derive the SOP canonical form, and the POS canonical form.

\[ f(a, b, c) = ? \]
Simplification using K-map

1. Find all prime implicants.

2. Find all essential prime implicants.

3. Collect all essential prime implicants + a minimal number of other prime implicants so that ON-set is totally covered.
a) List all prime implicants.
   a’c’, a’bd, bcd, acd, a’b’d’

b) List all essential prime implicants
   a’c’, acd, a’b’d’

c) List all minimal SOP forms.
   \[ F = a’c’ + acd + a’b’d’ + a’bd \]
   \[ F = a’c’ + acd + a’b’d’ + bcd \]
Practice Problem 3

a) List all prime implicates.
   \[ a' + c, \ a' + d, \ a + c' + d', \ a + b' + c', \ b' + c' + d \]

b) List all essential prime implicates
   \[ a' + c, \ a' + d, \ a + c' + d' \]

c) List all minimal POS forms.
   \[ F = (a' + c)(a' + d)(a + c' + d')(a + b' + c') \]
   \[ F = (a' + c)(a' + d)(a + c' + d')(b' + c' + d) \]
A logic network has three inputs (A, B, C) and one output (Z). In the following situations we set the output Z to either a logic zero or one:

1) The output Z is logic 1 when the binary value of ABC is greater than 3 and odd.
2) When the binary value of ABC is greater than 3 and even, Z is a logic 0.
3) When the binary value of ABC is less than 3, the output Z follows the result of the expression $B \oplus C$
4) Don’t care otherwise

Give K-map, minimal SOP and minimal POS.

$$Z = C + A'B = (A'+C)(B+C)$$
Multiplexerer

\[ f = ap'q' + bp'q + cpq' + dpq \]
Practice Problem 5

- Implement the function $f(a,b,c) = a'b + abc' + b'c$ using a minimal network of 2:1 multiplexers and minimum number of inverters. Do not use any other logic gate.
• If enable = 0, all outputs = 0
• If enable = 1, output $y_i = 1$ when $(ABC)_2 = i$

a) If EN = 1, C = 0, B = 1 and A = 0, which of the outputs is one? $y_2$

b) If EN = 1, C = 0, B = 1 and A = 1, which of the outputs is one? $y_5$

c) If EN = 0, C = 1, B = 0 and A = 1, which of the outputs is one? No one
Practice Problem 6

out(a, b, c, d) = b + c' + d'
Tree of Decoders

Implementing a function with n inputs using n to $2^n$ decoder is straightforward, but can we use smaller decoders?

Build a 4:16 decoder from 3:8 ones.
Practice Problem 7

Use 2:4 decoders and a minimal number of other gates to output a 1 for any prime number less than 16.