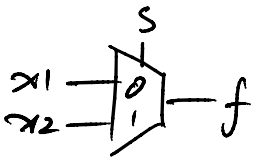
Verilog: procedural statement "always" Block

Recall 2-to-1 mux




```
module mux(input x1, x2, S, output f);
  assign f = (~S & x1) | (S & x2);
end module
```

→ alternative code (using if-else):

```
module mux(input x1, x2, S, output f);
  reg f;                    (*1)  (*3)
  always @(x1, x2, S)
    if (!S)                 (*2)
      f = x1;
    else
      f = x2;
end module
```

alternative code:

```
f = x1;
if (S==1)
  f = x2;
```

(*1) any signal assigned a value inside a "always" block must be declared as reg

(*2) if-else statement must be used inside a "always" block

(*3) sensitivity list = variables that will affect the assigned signal. (all inputs) alternatively: always @(*)

(*4) if more statements inside the "always" block is needed, use

```
begin
  .
  .
  .
end
```

if-else is considered as one statement

Case statement (used inside "always" block)

```
case (S)
  1'b0 : f = x1;
  1'b1 : f = x2;
  default:
end case
```

1 bit, binary value

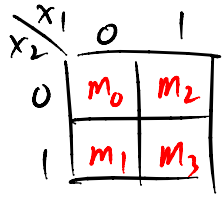
default: cover remaining (unused) cases.

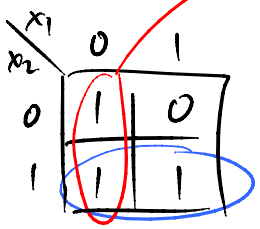
Karnaugh Map (K-map)

→ minimize logic expressions

eg.

x_1	x_2	f	
0	0	1	← $\bar{x}_1\bar{x}_2$ (m_0)
0	1	1	← $\bar{x}_1 x_2$ (m_1)
1	0	0	
1	1	1	← $x_1 x_2$ (m_3)





$m_0 + m_1$

$\bar{x}_1\bar{x}_2 + \bar{x}_1 x_2$

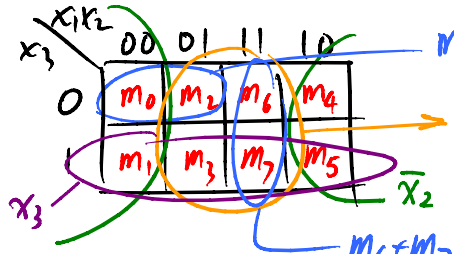
$\bar{x}_1$

$m_1 + m_3$

$\bar{x}_1 x_2 + x_1 x_2$

x_2

3-variable K-map (x_1, x_2, x_3) 010



$m_0 + m_2 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1 x_2\bar{x}_3 = \bar{x}_1\bar{x}_3$

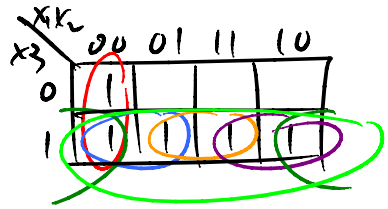
$m_2 + m_3 + m_6 + m_7 = \bar{x}_1 x_2 + x_1 x_2 = x_2$

$m_6 + m_7 = x_1 x_2\bar{x}_3 + x_1 x_2 x_3 = x_1 x_2$

Implicant: any product term for which the function is equal to 1.

m_0, m_1, m_3, m_5, m_7

$\bar{x}_1\bar{x}_2,\ \bar{x}_1 x_3,\ x_2 x_3,\ x_1 x_3,\ \bar{x}_2 x_3,\ x_3$



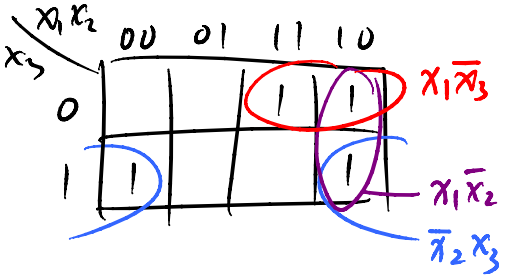
Literal: a variable, or its complement $(x_1, \bar{x}_1)$

Prime Implicant (PI): an implicant for which it is not possible to remove any literal & still have a valid implicant (draw the biggest circling for each "1" entry) $\bar{x}_1\bar{x}_2,\ x_3$

Cover: any set of implicants that cover all the 1's of a function.

example $f(x_1, x_2, x_3) = \sum m(1, 4, 5, 6)$

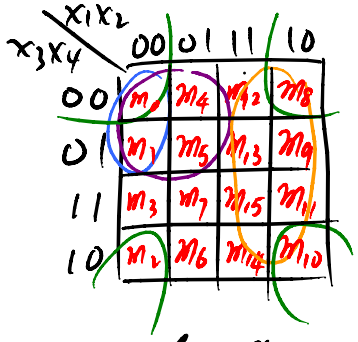
cover: $x_1\bar{x}_3 + x_1\bar{x}_2 + \bar{x}_2 x_3$



Essential PI. a P.I. that must be included in any cover of the function.

minimal cover: $x_1\bar{x}_3 + \bar{x}_2 x_3$

4-variable K-map



$m_0 + m_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3$

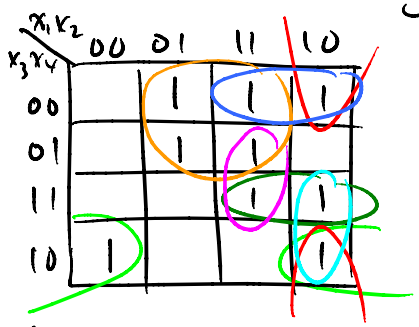
$m_0 + m_1 + m_4 + m_5 = \bar{x}_1 \bar{x}_3$

$m_8 + m_9 + m_{10} + m_{11} + m_{12} + m_{13} + m_{14} + m_{15} = x_1$

$m_0 + m_2 + m_8 + m_{10} = \bar{x}_2 \bar{x}_4$

$f = \sum m(2, 4, 5, 8, 10, 11, 12, 13, 15)$

example



P.I.s: $x_2\bar{x}_3$, $x_1x_3x_4$, $x_1\bar{x}_2\bar{x}_4$, $x_1\bar{x}_3\bar{x}_4$
$\bar{x}_2x_3\bar{x}_4$, $x_1x_2x_4$, $x_1\bar{x}_2x_3$

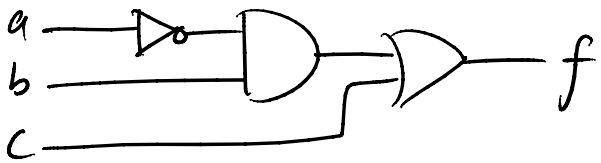
Essential P.I.s: $x_2\bar{x}_3$ (due to m_4, m_5)
$\bar{x}_2x_3\bar{x}_4$ (due to m_2)

minimal cover $f = x_2\bar{x}_3 + \bar{x}_2x_3\bar{x}_4 + x_1x_3x_4 + \begin{cases} x_1\bar{x}_2\bar{x}_4 \\ x_1\bar{x}_3\bar{x}_4 \end{cases}$ or

Summary:
1. Find all P.I.s (largest groups of 1's)
2. Identify Essential P.I.s. If essential P.I.s cover all the 1's of the function, then we are done!
3. Choose non-essential P.I.s to obtain a min-cost* cover

* cost is measured as total # of gates + total # of inputs

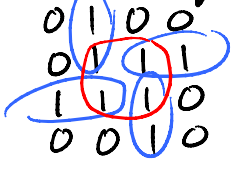
i.e.



Cost = 3 + 5 = 8
(gates) (inputs)

If ▷o— are not counted, then Cost = 2 + 4 = 6

Back to K-map example (4-variable)



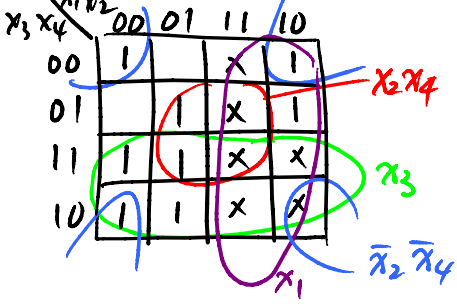
Note: in this case, we don't always need the largest groups of 1's. because all ⬭ would make a cover

Don't-care minterms: sometimes we know that certain combinations of inputs to a function either **can't occur** or we **don't care** about them. For example, we wish to display decimal digit (only) 0–9 on a 7-seg. display

Display	x_1	x_2	x_3	x_4	h_0	h_1	h_2	h_3	h_4	h_5	h_6
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
	0	0	1	0	1						
	0	0	1	1	1						
	0	1	0	0	0						
	0	1	0	1	1						
	0	1	1	0	1						
	0	1	1	1	1						
	1	0	0	0	1						
9	1	0	0	1	1						
	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x						x
	1	1	0	0	x						x
	1	1	0	1	x						x
	1	1	1	0	x						x
	1	1	1	1	x	x	x	x	x	x	x

h_0 | h_5 | h_6 | h_1 | h_4 | h_2 | h_3

K-map for h_0



$h_0 = x_3 + x_1 + x_2x_4 + \bar{x}_2\bar{x}_4$

Note: in this K-map, we choose all "x" to be "1". On other K-maps, we may pick some "x" to be "0" to help us get minimal-cost cover

5-variable K-map $(x_1x_2x_3x_4x_5)$