CS2100: Computer Organisation

Tutorial #6: Boolean Algebra, Logic Gates and Simplification

(Week 8: 9 – 13 March 2026)

Answers to Selected Questions

By default, we assume that complemented literals are NOT available, unless otherwise stated.

Logic constants (0 and 1) are always available, and they are considered (degenerate form of) SOP and POS expressions.

The above are to be assumed from now onwards and may not be repeated in future tutorials, assignments and final exam.

In general, students are weak at identifying the EPIs (essential prime implicants) correctly. Make sure you do the self-exploratory questions. Finding out how Quine-McCluskey's method works (though not in the syllabus) may help you understand better.

1. Using Boolean algebra, simplify each of the following expressions into simplified **sum-of-products (SOP) expression**. Indicate the law/theorem used at every step.

   (a) $F(x,y,z) = (x + y \cdot z') \cdot (y' + y) + x' \cdot (y \cdot z' + y)$

   (b) $G(p,q,r,s) = \Pi M(5, 9, 13)$

   [**Tip:** For (b), it is easier to start with the given expression and get done in about 6 steps, rather than expanding it into sum-of-products/sum-of-minterms expression first.]

   Answers: Note that there could be more than one way of derivation. Only part (b) answer is shown.

   (b) $G(p,q,r,s) = \Pi M(5, 9, 13)$

   $= (\mathbf{p + q' + r + s'}) \cdot (p' + q + r + s') \cdot (\mathbf{p' + q' + r + s'})$

   $= ((\mathbf{p \cdot p'}) + (q' + r + s')) \cdot (p' + q + r + s')$ [distributive]

   $= (\mathbf{0} + (\mathbf{q' + r + s'})) \cdot (p' + q + r + s')$ [complement]

   $= (q' + \mathbf{r + s'}) \cdot (p' + q + \mathbf{r + s'})$ [identity]

   $= (\mathbf{q' \cdot (p' + q)}) + (r + s')$ [distributive]

   $= \mathbf{p' \cdot q' + r + s'}$ [absorption 2]

   Reminder: Write · for AND, and not to leave it out. For example, for "*x* AND *y*", write *x*·*y* and not *xy*. Writing *xy* when it should be *x*·*y* will receive zero mark.

2. We assume you have done the self-exploratory questions above.

   Present your answers for the expressions in (a), (b), (c) in SE3. You do not need to draw the K-maps.

   Now, draw the K-map for (d) below, determine the number of PIs and EPIs in the K-map, and find the simplified SOP and POS expressions.

   (d) $F4(A,B,C,D) = \Pi M(4, 8, 9, 11, 12) \cdot D(2, 3, 6, 7, 10, 14)$

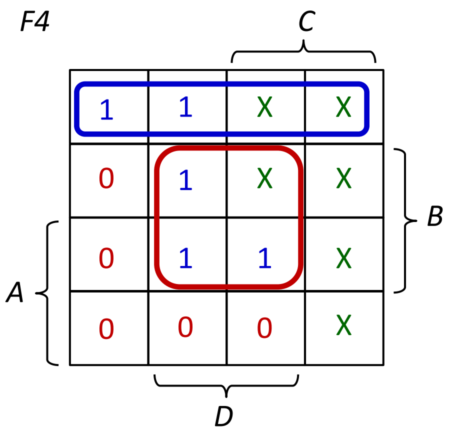
   Answers:

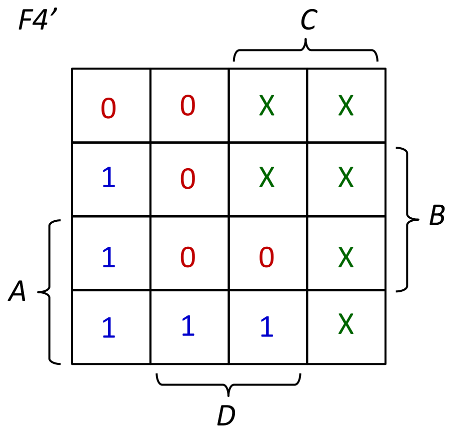
   (a) $F1(A,B,C,D) = \Sigma m(5, 8, 10, 12, 13, 14) = \mathbf{A \cdot D' + B \cdot C' \cdot D}$ (3 PIs, 2 EPIs)

   (b) $F2(W,X,Y,Z) = \Pi M(0, 1, 2, 8, 9, 10) = \mathbf{X + Y \cdot Z}$ (2 PIs, 2 EPIs)

   (c) $F3(K,L,M,N) = \Sigma m(1, 7, 10, 13, 14) + d(0, 5, 8, 15)$

   $= \mathbf{L \cdot N + K \cdot M \cdot N' + K' \cdot L' \cdot M'}$ or $\mathbf{L \cdot N + K \cdot M \cdot N' + K' \cdot M' \cdot N}$ (6 PIs, 1 EPI: $L \cdot N$)





Number of PIs: 4 ($A' \cdot B'$, $A' \cdot D$, $B \cdot D$, $B \cdot C$)
Number of EPIs: 2 ($A' \cdot B'$, $B \cdot D$)
Note that $A' \cdot C$ is <u>not</u> a PI as it consists of all don't cares.

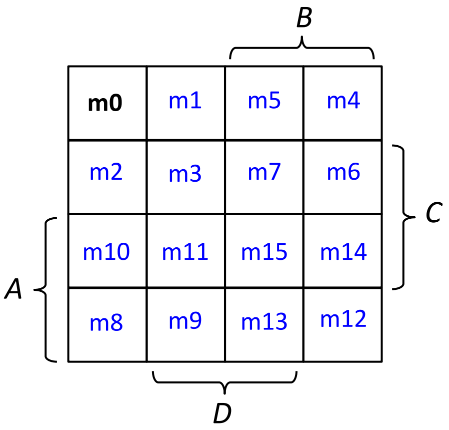
SOP expression: $F4 = A' \cdot B' + B \cdot D$

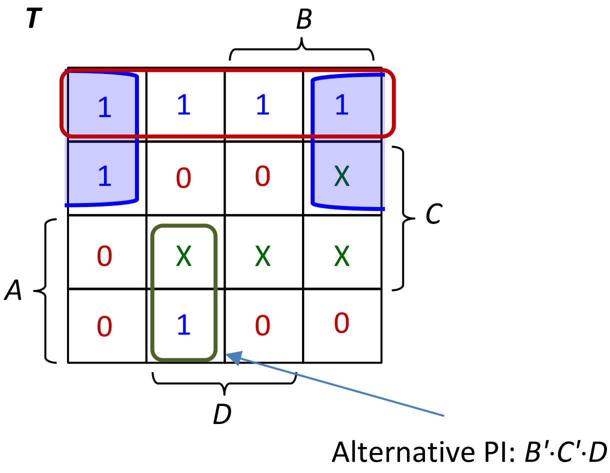
$F4' = A \cdot B' + B \cdot D'$

$F4 = (A \cdot B' + B \cdot D')' = (A'+B) \cdot (B'+D)$

POS expression: $F4 = (A'+B) \cdot (B'+D)$

3. (a) The following K-map layout is used for a 4-variable Boolean function *T*(*A*,*B*,*C*,*D*). Fill in the minterm positions m1 to m15 into the respective cells. m0 has been filled for you.





(b) Given the following 4-variable Boolean function:

$$T(A,B,C,D) = \Pi M(3,7,8,10,12,13) \cdot X(6,11,14,15)$$

where X's are the don't-cares, write out the Σm notation for *T*(*A*,*B*,*C*,*D*).

(c) Draw the K-map for *T* using the layout above.

(d) How many PIs (prime implicants) are there in the K-map? List out all the PIs.

(e) How many EPIs (essential prime implicants) are there? List out all the EPIs.

(f) What is the simplified SOP expression for *T*? List out all alternative solutions.

(g) What is the simplified POS expression for *T*? List out all alternative solutions.

(h) Implement the simplified SOP expression for *T* using a 2-level AND-OR circuit and a 2-level NAND only circuit.

Answers:

(a) See above.

(b) $T(A,B,C,D) = \Sigma m(0,1,2,4,5,9) + X(6,11,14,15)$.

(c) See K-map above.

(d) 4 PIs: $A' \cdot D'$, $A' \cdot C'$, $A \cdot B' \cdot D$ and $B' \cdot C' \cdot D$.

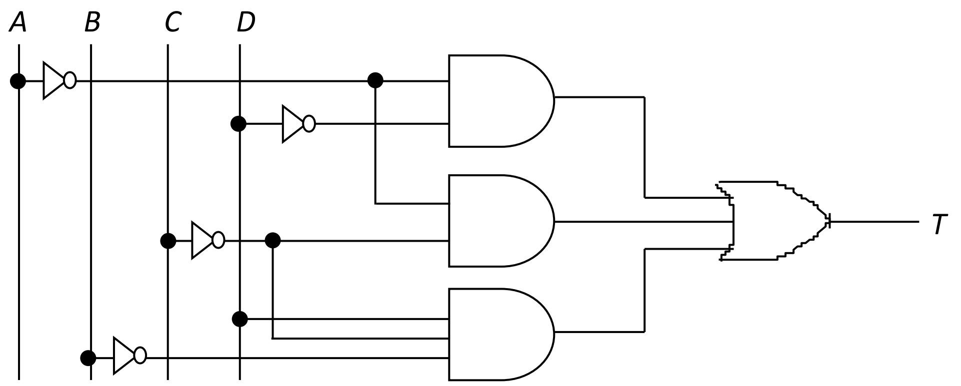
(e) 2 EPIs: $A' \cdot D'$ and $A' \cdot C'$.

(f) SOP expression: $T(A,B,C,D) = A' \cdot D' + A' \cdot C' + B' \cdot C' \cdot D$ or $A' \cdot D' + A' \cdot C' + A \cdot B' \cdot D$.

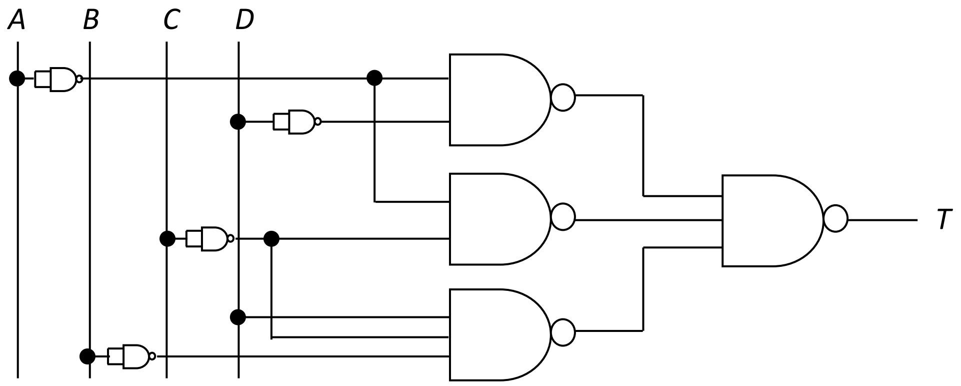
(g) POS expression: $T(A,B,C,D) = (A' + D) \cdot (C' + D') \cdot (A' + B')$.
[Working: $T'(A,B,C,D) = A \cdot D' + C \cdot D + A \cdot B$.]

(h) Take the expression $A'\cdot D' + A'\cdot C' + B'\cdot C'\cdot D$

2-level AND-OR circuit:



2-level NAND circuit:



Students: Draw logic diagrams neatly with <u>straight lines</u>. Draw thick dots to represent forks.

Using Quine McCluskey to find the simplified SOP expression for *T*.

(Just for illustration. Quine McCluskey is not in the scope of CS2100, but knowing it will strengthen your understanding of K-map, and appreciate why K-map is faster and easier.)

$T(A,B,C,D) = \Sigma m(0,1,2,4,5,9) + X(6,11,14,15)$.

PI chart:

✓0: 0000	✓0,1: 000-	0,1,4,5: 0-0-
	✓0,2: 00-0	0,2,4,6: 0--0
	✓0,4: 0-00	
✓1: 0001	✓1,5: 0-01	
✓2: 0010	1,9: -001	
✓4: 0100	✓2,6: 0-10	
✓5: 0101	✓4,5: 010-	
✓6: 0110	✓4,6: 01-0	
✓9: 1001	6,14: -110	
✓11: 1011	9,11: 10-1	
✓14: 1110	11,15: 1-11	
✓15: 1111	14,15: 111-	

These are PIs. (0,1,4,5: 0-0-; 0,2,4,6: 0--0; 1,9: -001; 9,11: 10-1)

These are not PIs because they contain solely don't-cares. (6,14: -110; 11,15: 1-11; 14,15: 111-)

Reduced PI Chart:

Collecting the 4 PIs, we draw this reduced PI chart:

PI	Minterms						Don't-cares	
	0	1	2	4	5	9	6	11
0,1,4,5: 0 - 0 - ($A' \cdot C'$)	▪	▪		▪	▪			
0,2,4,6: 0 - - 0 ($A' \cdot D'$)	▪		▪	▪			▪	
1,9: -0 0 1 ($B' \cdot C' \cdot D$)		▪				▪		
9,11: 1 0 - 1 ($A \cdot B' \cdot D$)						▪		▪

Look under the minterms columns to find any column containing just one dot.
Since minterm m2 is covered only by $A' \cdot D'$, so $A' \cdot D'$ must be an EPI.
Likewise, minterm m5 is covered only by $A' \cdot C'$, so $A' \cdot C'$ must be an EPI.
Minterms m0, m1, m2, m4, m5 are covered by these 2 EPIs, leaving only minterm m9, which can be covered either by $B' \cdot C' \cdot D$ or $A \cdot B' \cdot D$.

4. A circuit takes in four inputs *K,L,M,N* and generates 3 outputs *X,Y,Z* as follow:

$X(K,L,M,N) = 1$ if $KL = MN$, or 0 otherwise,
where *KL* and *MN* are 2-bit unsigned integers.

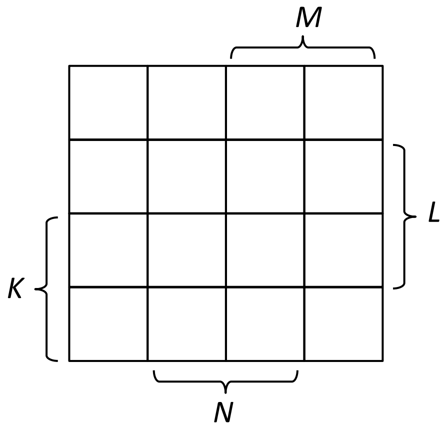
$Y(K,L,M,N) = 1$ if $KL \leq MN$, or 0 otherwise,
where *KL* and *MN* are 2-bit unsigned integers.

$Z(K,L,M,N) = 1$ if $KLM < LMN$, or 0 otherwise,
where *KLM* and *LMN* are 3-bit unsigned integers.

For parts (a) – (c) below, you may assume that the input 0000 will not occur.

(a) Fill in the truth table for the circuit. Write 'd' for don't cares.

(b) Fill in the K-maps of *X*, *Y* and *Z* using the layout given below.



(c) Write out the simplified SOP expressions of *X*, *Y* and Z.

(d) After designing the circuit according to the simplified SOP expressions in (c), if you feed the input 0000 into it, what will be the outputs?

(This is an important question that reminds students on the use of don't-cares.)

Only answers for (c) and (d) are shown:

(c) $X = K' \cdot L \cdot M' \cdot N + K \cdot L' \cdot M \cdot N' + K \cdot L \cdot M \cdot N$

$Y = M \cdot N + K' \cdot N + K' \cdot M + L' \cdot M$

$Z = K'$

(d) Input *KLMN* = 0000; output *XYZ* = 001.