\[ \text{Q1 P7 ch11} \]

(a) \( SDC \) are straightforward.

\[ SDCy_2 = y_2 \oplus ac \text{ and so on...} \]

(b) \( ODCy_4 = (y_5 = 0 \land y_2 = 1) \)

\[ = \overline{y}_5 + y_2 \\
= ab + ac \]

\( ODCy_5 = y_4 = 0, \ y_2 = 1 \)

\[ = \overline{y}_4 + y_2 \\
= \frac{a}{a \overline{c} + ac} = \overline{a} + c = a + c \]

(c) \( F_4 = a \overline{c} \)

\[ F_4 + ODCy_4 = a \overline{c} + ab + ac = a \]

\( F_5 = \overline{ab} = \overline{a} + \overline{b} \)

\[ F_5 + ODCy_5 = \overline{a} + \overline{b} + a + c \]

\[ = \overline{b} \]
$\text{ODC } h \equiv g = 1 = ab$

$\text{ODC } g \Theta z_1 \equiv h = 1 = \overline{c}$

$\text{ODC } g \Theta z_2 \equiv f = 1 = abc$

$\text{ODC } f \Theta z_2 \equiv g = 1 = ab.$

Simplify $g \cdot h \text{ w/ } \text{ODC } h + \text{ODC } h$

\[= \overline{c} + ab = \overline{c}\]

No simplification.

Simplify $g = ab$ w/ $\overline{c}$ or $abc \rightarrow$ No simplification.

Simplify $f \text{ w/ } \text{ODC } f = ab$

\[= ab \overline{c} + ab = ab(c+1) = ab.\]

But $ab = \text{D.C}$

So $f = 0$
Note that the set of maximal compatibles is itself the solution, all of these are needed for covering.
1. Extra write-up for minimizing M1, and for the encoding problem.
2. M1 is an incompletely specified machine. The first thing that you need to do is to identify compatible pairs. You can use either the merger table, or the merger graph. If you build the merger table, you can see that the only compatible (BD) gets cancelled because its implied pair (CD) is not compatible.


Note that (AE) and (CF) are the only compatibles in the maximal set that contain C and E. Therefore, they have to be in the solution. That takes care of A, C, E, F. What about B and D? They are also in different compatible groups. Clearly, all 4 have to be in the solution. Now the question is whether their implied pairs are also in the maximal compatible sets? That they are.

Note: From the maximal compatible set, we can find a minimum number of compatibles that cover the entire machine. This guarantees covering, but not closure. On the other hand, the set containing all maximal compatibles is clearly a closed covering. Therefore, this is an upper bound on the # of states. However, in general, we may be able to get a better solution - a smaller set. And thats why we need to go into that compatibility graph and all that we studied in class.

In our current example, we don’t need to solve the problem on the compatibility graph. This is because, from the maximal compatible set we can figure out that we need all maximal compatibles just to cover the machine. And since the set of all maximal compatibles also guarantees closure, we can just use the set of all maximal compatibles as our solution.

However, for your benefit, I’m still solving the problem.
Compatible pairs:

$\begin{array}{cccc}
B & C & D & E \\
\times & X & X & X \\
\times & X & X & X \\
X & X & X & X \\
A & B & C & D \\
\end{array}$

$\begin{array}{cccc}
AB, AE, BC, BD, CD \\
AB, AE, BCD \\
\downarrow \\
\text{cover}\: \: \: \: \: \text{yes!} \\
\text{but closure?} \\
\end{array}$

Cover 1: $\rightarrow$ AE, CD, BD $\rightarrow$ 3 states.

Cover 2: $\rightarrow$ BC, AE, CD, BD

(BCD), (AE) $\rightarrow$ 2 states.
Now let us solve M2 using BCP.

Max. comp.: BCD, AB, AE

\[
P_1 \lor BCD \rightarrow AE
\]

\[
P_2 \lor AB \rightarrow \phi
\]

\[
P_3 \lor AE \rightarrow CD
\]

\[
BC \rightarrow AE \times \text{not prime}, \leq p_1
\]

\[
P_4 \lor BCD \rightarrow CD
\]

\[
CD \rightarrow BD, AE \times \text{not prime}, \leq p_2
\]

\[
A \rightarrow \phi \times \text{not prime}, \leq p_2
\]

\[
B \rightarrow \phi " "
\]

\[
P_5 \lor C \rightarrow \phi
\]

\[
P_6 \lor D \rightarrow \phi
\]

\[
P_7 \lor E \rightarrow \phi
\]

State A := \( p_2 + p_3 \)  

B := \( p_1 + p_2 + p_4 \)  

C := \( p_1 + p_5 \)  

D := \( p_1 + p_4 + p_6 \)  

E := \( p_3 + p_7 \)  

\[
P_1 \rightarrow p_3 = \overline{p_1 + p_3} \quad C_1
\]

\[
p_3 \rightarrow p_1 = \overline{p_3 + p_1} \quad C_2
\]

\[
BD \rightarrow CD
\]

\[
p_4 \rightarrow p_1
\]

\[
\overline{p_4 + p_1} \quad C_3
\]
Set-up BCP for M2.

Columns = prime compatibles \( P_1, \ldots, P_7 \)

Rows = state covering + closure

Select a minimum \( \# \) of primes (columns) s.t.

all constraints (rows) are SAT.

\[
\begin{array}{ccccccc}
  & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 \\
A & 1 & 1 & 1 & - & - & - & (P_2 + P_3) \\
B & 1 & 1 & - & - & 1 & - & (P_1 + P_4) \\
C & 1 & - & - & - & - & 1 & (P_1 + P_5) \\
D & 1 & - & - & - & - & 1 & (P_1 + P_4 + P_6) \\
E & - & - & 1 & - & - & 1 & (P_3 + P_7) \\
\end{array}
\]

No essentials, no dominance

\[ \Rightarrow \text{cyclic, } M \]
Now compute M11S.

ignore rows with zeros (c1, c2, c3)

\[ L.B = M11S = \{ \text{row A} \quad \text{row B} = 2^2 \quad \text{row C} \quad \text{row E} \} = 2 \]

maximally independent set of constraints.

Branch on \( p_1 \) (greedy, most entries)

\[ p_1 = 1 \]

\[ \begin{array}{ccc}
A & 1 & 1 \\
E & 1 & 1 \\
c1 & 1 & 1 \\
\end{array} \]

\[ p_1 + p_3 = L.B = 2 \]

Stop.

\[ p_1 \& p_3 \text{ solution} \]

(CD) (AE)

2 state M/C.
Q3 Static False paths.

Path 1: \( w \rightarrow y_1 \rightarrow y_3 \rightarrow f. \)
Path 2: \( x \rightarrow y_1 \rightarrow y_3 \rightarrow f. \)
Path 3: \( x \rightarrow y_2 \rightarrow y_3 \rightarrow F. \)
Path 4: \( y \rightarrow y_2 \rightarrow y_3 \rightarrow F. \)
Path 5: \( y \rightarrow F. \)

\( p_1 = \text{Propagate } w \text{ to } F \) require \( x=1 \) \& \( y_2=1 \) \& \( y=0. \)

\( p_1 = \text{false} \)

Statically, \( w/1 \) and \( w/0 \) not testable.

Put \( w=1 \) & simplify or \( w=0 \) and simplify.

\( y_1=0. \)

\( F = y \) (beautiful ckt!)
Path p2: needs \( y_2 = 0 \) \& \( x = 1 \)

\[ \Rightarrow x . \bar{y} @ y_3 = 0. \]
also false.

\( S_0 @ x \neq \text{testable}. \)
put \( x = 0 \) \& simplify, \( F = y_1 \).

No surprise that \( P_3 = \text{false} \) too.

\( P_4? \quad y \rightarrow y_2 \quad \text{requires} \quad x = 1 \)

\( y_2 \rightarrow y_3 \quad \text{requires} \quad y_1 = 1 = x = w = 1 \)

statically, \( P_4 \) not false.

But \( y \rightarrow y_2 \rightarrow y_3 \rightarrow D \)

\[ \text{faster.} \]

So \( y \rightarrow y_2 \rightarrow y_3 \rightarrow F \) has no effect, so it is actually false if timing were to be analyzed.

Static sensitization \[\n\Rightarrow\text{false paths.}\]
Now the other way:

\[ P_1 = \text{is it truly timing false?} \]

\[ \begin{align*}
  w &= \uparrow \\
  y_1 &= \uparrow @1 \\
  z &= \uparrow @2 \\
  y_2 &= \uparrow \\
  y &= 0 \\
\end{align*} \]

\[ P_1 \text{ controlled by } x, \text{ also re-convergent at } y_3 \]

Assume gate delays = 1

\[ w = \uparrow \text{ rising transition.} \]
\[ x = \uparrow \text{ rising transition.} \]
\[ y = 0 \]

Glitches flowing through \( P_1 \& P_2 \& P_3 \)

So they are not "really false."