Robert Dick, Fall '07 Final

1) \( f = \Sigma(0, 1, 2, 5, 6, 7) \)

<table>
<thead>
<tr>
<th>( \Sigma_6 )</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\hat{f} &= \bar{a} \bar{b} + b \bar{c} + ac \\
\bar{f} &= \bar{a}b + b \bar{c} + ac \\
f &= \bar{a}b \cdot \bar{b} \cdot ac \\
f &= (a+b)(\bar{b}c)(\bar{a}c)
\end{align*}
\]

Where to start?

Consider two alternatives, after which all subsequent steps are obvious.

Each new column covers at most two rows. Therefore, at least 3 columns required for 6 rows.

\( \Sigma \overline{000}, \overline{010}, 1 \overline{13} \) and \( \Sigma \overline{001}, \overline{011}, 1 \overline{13} \)

Equally good.
2) NP-completeness is the property of problems for which solutions can be checked in polynomial time (in terms of the input size) by a deterministic, i.e., Turing-equivalent abstract machine (oftentimes called a deterministic machine).
3) a) Develop a P-time algorithm
   + fast solution
   - probably impossible
b) try vs. an exponential time algorithm
   + optimal results
   - only finishes for small problem instances
c) Develop an approximation algorithm
   + bounded deviation from optimality
   - can be very difficult
d) Identify previously unknown constraints on problem instances
   of interest to allow P-time algorithm
   - optimal and fast
   - doesn’t always work
e) Develop a fast heuristic
   + may work pretty well much of the time
   - generally hard to bound deviation from optimality
4. Consider whether each cube is relatively essential by cofactoring other cubes by it and checking for tautology.

\[
\begin{array}{c|c|c}
0\times00 & XXX0 & \rightarrow a: x\times\times \\
0\times01 & XXX1 & \rightarrow b: x\times\times \\
x1\times x & \Rightarrow & \quad \quad \quad \quad \text{Tautology} \\
x1\times x & \quad & \quad \\
x1\times x & \quad & \quad \\
x1\times x & \quad & \quad \\
1\times\times x & 000X & \quad \\
\end{array}
\]

\[
\begin{array}{c|c|c}
000x & X0xx & \rightarrow b: x\times\times \\
0\times01 & \quad & \quad \\
x1\times x & \Rightarrow x1xx & \rightarrow b: x\times\times \\
x1\times x & \quad & \quad \quad \quad \text{Tautology} \\
x1\times x & \quad & \quad \\
x1\times x & \quad & \quad \\
1\times\times x & 0x00 & \quad \\
\end{array}
\]

\[
\begin{array}{c|c|c}
000x & x0xx & \rightarrow b: x\times\times \\
0\times00 & \quad & \quad \\
x1\times x & \Rightarrow x1xx & \rightarrow b: x\times\times \\
x1\times x & \quad & \quad \quad \quad \text{Tautology} \\
x1\times x & \quad & \quad \\
x1\times x & \quad & \quad \\
1\times\times x & 0x01 & \quad \\
\end{array}
\]
<table>
<thead>
<tr>
<th>000X</th>
<th>010X</th>
<th>11XX</th>
<th>111X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \overline{b} = \not x \]

\[ \text{XXX is relatively essential} \]

**Relatively essential \{ XXX \}**

Determine which redundant cubes covered by relatively essential cubes:

- O00X | XXX = null
- 0X00 | XXX = null
- 0X01 | XXX = null
- X1XX | XXX = X1XX not tautology
- X11X | XXX = X11X not tautology

None is covered, all are partially redundant
After removing minterms covered by relatively-essential cubes, these remain dominated

minimal covering: \(1\overline{x}x\), \(x\overline{1}x\), \(000\overline{x}\)
\[ f = \overline{bd} + \overline{a} \overline{cd} + \overline{a} \overline{b} + ad + ac \]

**Case:**

\[ a = \overline{b} + \overline{d} + c \quad k_1 = 2 \]

\[ b = \overline{d} \quad k_2 = 1 \]

\[ c = a \quad k_3 = 1 \]

\[ d = \overline{b} \]

\[ \overline{a} = bd \]

\[ f = a (\overline{b} + \overline{d} + c) + \overline{b} \overline{d} + \overline{a} \overline{b} \overline{d} \]
6) High-k dielectric allows gate thickness to be increased while still forming a channel w. the same Vgs. This reduces gate (tunneling) leakage, which would have otherwise been a big problem in 45nm processes.

![Diagram]

High Vgs forms channel only when 91 does not have excess electrons. Very high Vgs can be used to tunnel electrons from 5 to 91. Very low Vgs or ultra violet light can be used to remove electrons from 92.
q) CT

Diagram with arrows indicating transitions.

P

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>B</td>
</tr>
</tbody>
</table>

A: 00
B: 11
C: 10
D: 01
10) \((0+1)^* 01 (0+1)^* + (0+1)^* 01\)

**MFA**

**DFA**

After this point, always produce 1 output. Implication chart would yield same result.

**Min DFA**
11) Two out of three majority vote logic.  
Unary of full adder.

12) Yes. It wouldn't change the timing diagram much because address was driven a cycle before data. The components on the bus might need to internally latch the address, however.