Maurice Karnaugh

The Karnaugh map makes it easy to minimize Boolean expressions!
A function of four variables a b c d.

The truthtable consists of 11 "1" and 5 "0". According to earlier, we know that the function can be expressed in the SoP form with 11 minterms or PoS form with 5 maxterms.

Anyone who used the Boolean algebra know that it then follows hard work to produce simpler expressions. Minterms could be combined in many different ways, which all result in different simplified expression. How do we know that we have found the minimal expression?
A map with frames at unit distance

The Karnaugh map is the Truth Table but with the minterms in a different order.

Note the numbering!

The frames are ordered in such way that only one bit changes between two vertical frames or horizontal frames. This order is called Gray code.
Two "neighbors"

The frames "5" and "13" are "neighbors" in the Karnaugh map (but they are distant from each other in the truthtable).

They correspond to two minterm with four variables, and the figure shows how, with Boolean algebra, they can be reduced to one term with three variables.

What the two frames have in common is that $b = 1$, $c = 0$ and $d = 1$; and the reduced term expresses just that.

Everywhere in the Karnaugh map where one can find two ones that are "neighbors" (vertically or horizontally) the minterms could be reduced to "what they have in common". This is called a grouping.
Four "neighbors"

Frames "1" "3" "5" "7" is a group of four frames with "1" that are "neighbors" to each other. Here too, the four minterms could be reduced to a term that expresses what is common for the frames, namely that $a = 0$ and $d = 1$.

Everywhere in Karnaugh map where one can find such groups of four ones such simplifications can be done, grouping of four.
Eight "neighbors"

All groups of 2, 4, 8, (... $2^N$) frames containing ones can be reduced to a term, with what they have in common, grouping of $n$. 

William Sandqvist william@kth.se
The Karnaugh map should be drawn on a torus (a donut). When we reach an edge, the graph continues from the opposite side! Frame 0 is the "neighbor" with frame 2, but also the "neighbor" with frame 8 which is "neighbor" to frame 10.
The optimal groupings?

One is looking for the biggest grouping as possible. In the example, there is a grouping with eight ones (frames 0,1,3,2,4,5,7,6). Corners (0,2,8,10) is a group of four ones.

Two of the frames (0, 2) has already been included in the first group, but it does not matter if a minterm is included multiple times. All ones in the logic function must either be in a grouping, or be included as a minterm. The "1" in frame 13 may form a group with "1" in frame 5, unfortunately there are no bigger grouping for this "1".

- The resulting function is apparently a major simplification compared to the original function with the 11 minterms.

William Sandqvist william@kth.se
**Incorrect groupings?**

Is there any incorrect groupings in this Karnaugh diagrams?

<table>
<thead>
<tr>
<th></th>
<th>a0</th>
<th>a1</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

William Sandqvist william@kth.se
Incorrect groupings?

Is there any incorrect groupings in this Karnaugh diagrams?

Groupings should be 2, 4, 8 (= power of two) "neighbors" vertikal or horizontal, not diagonal.

William Sandqvist william@kth.se
Ex. 6.1  Karnaugh map

<table>
<thead>
<tr>
<th>cd</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b 0 0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Ex. 6.1 Karnaugh map

\[ \text{bd} \]
Ex. 6.1 Karnaugh map

\[ \overline{bd} \]

\[ abd \]
Ex. 6.1 Karnaugh map

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

William Sandqvist william@kth.se
Ex. 6.1 Karnaugh map

\[ f = \overline{bd} + \overline{acd} + abd \]
(Ex. 6.2 Karnaugh map)
(Ex. 6.2 Karnaugh map)

\[ b^d \]

\[
\begin{array}{cccc}
  & cd & 00 & 01 & 11 & 10 \\
  a \\
  b & 00 & 1 & 0 & 0 & 1 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 1 & 1 & 1 \\
  1 & 1 & 0 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 & 0 & 1 \\
\end{array}
\]
(Ex. 6.2 Karnaugh map)

\[ \begin{array}{cccc}
\text{cd} & 00 & 01 & 11 & 10 \\
\hline
a & \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
\end{array} \]

\[ \overline{bd} \]

\[ abd \]
(Ex. 6.2 Karnaugh map)

\[\begin{array}{c|c|c|c|c}
\text{cd} & 00 & 01 & 11 & 10 \\
\hline
a & 00 & 1 & 0 & 0 \\
\hline
b & 00 & 0 & 0 & 0 \\
\hline
0 & 00 & 0 & 0 & 0 \\
1 & 01 & 1 & 1 & 1 \\
\hline
1 & 00 & 1 & 0 & 1 \\
0 & 00 & 1 & 0 & 1 \\
\end{array}\]
(Ex. 6.2 Karnaugh map)

\[ f = \overline{bd} + abd + abc \]
Grouping of "0"

The Karnaugh map is also useful for groupings of 0's. The groupings may include the same number of frames as the case of groupings of 1's. In this example, 0: s are grouped together in pairs with their "neighbors". Maxterms are simplified to what is in common for the frames.

The simplified expression is the product of three sums it represents a very substantial simplification of the original function's five maxterms.

William Sandqvist william@kth.se
De Morgan

If you use "0" as if they were "1" you will get the function inverted! (totally wrong)

With De Morgan's theorem you can invert the inverted function to get the result. (now correct)

William Sandqvist william@kth.se
Maps for other number of variables

Karnaugh maps with three and two variables are also useful.

The Karnaugh map can conveniently be used for functions of up to four variables, and with a little practice up to six variables.
Ex 6.4 change NOR to NAND

William Sandqvist  william@kth.se
Ex. 6.4

NOR-NOR to OR-AND change "straight on!"

\[(a + \overline{c})(b + \overline{c}) = (a + c)(\overline{b} + \overline{c}) = (a + c) + (b + c)\]

OR-AND \[\leftrightarrow\] NOR-NOR

William Sandqvist  william@kth.se
Ex. 6.4

\[
f(a, b, c) = (a + \overline{c}) \cdot (b + \overline{c})
\]
Ex. 6.4

\[ f(a, b, c) = (a + c) \cdot (b + c) \]
Ex. 6.4

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Ex. 6.4

\[
f(a, b, c) = \overline{c} + a \cdot b
\]
Ex. 6.4

\[ f(a, b, c) = \overline{c} + a \cdot b \]
Ex. 6.4

\[ f(a, b, c) = \overline{c} + a \cdot b \]

AND-OR NAND-NAND change gates "straight on"
Ex. 6.4

\[ f(a, b, c) = \overline{c} + a \cdot b \]

No gate on this level

\[ ab + c = \overline{ab + c} = \overline{a \cdot b \cdot c} \]

AND-OR ↔ NAND-NAND
Ex. 6.4

$$f(a, b, c) = \overline{c} + a \cdot b$$

Or algebraic:

Double invert = standard trick

\[
\overline{\overline{c + a \cdot b}} = \overline{c + a \cdot b} = c \cdot a \cdot b
\]

William Sandqvist william@kth.se
PLD-chip has output inverters

PLD circuits often have an XOR gate at the output so that they shall be able to invert the function. One can then choose to bring together 0s or 1s after what is most advantageous.

When the control signal is a "1" the gates output is b's inverse, when a is "0", the output is equal to b

William Sandqvist  william@kth.se
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \bar{f} = ? \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \bar{f} = ? \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x_1</th>
<th>x_0</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>15</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

William Sandqvist  william@kth.se
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \bar{f} = ? \]

Grouping of "1"

Grouping of "0"

William Sandqvist  william@kth.se
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]

Grouping of ”1”

\[ f = x_1 x_0 + x_2 x_0 \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]

Grouping of "1"

\[ f = \overline{x_1} \overline{x_0} + \overline{x_2} x_0 \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]

Grouping of "1"

\[ f = \overline{x_1} \overline{x_0} + \overline{x_2} \overline{x_0} \]

Grouping of "0"

\[ \overline{f} = \{ "0" \text{ as } "1" \} = x_0 + x_2 x_1 \]
Ex. 6.5 Minimize with the K-map

\[ f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ? \]

Grouping of "1"

\[
\begin{array}{cccc}
00 & 01 & 11 & 10 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 \\
\end{array}
\]

\[ f = \overline{x_1} \overline{x_0} + \overline{x_2} \overline{x_0} \]

Grouping of "0"

\[
\begin{array}{cccc}
00 & 01 & 11 & 10 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
2 & 2 & 0 & 0 \\
3 & 3 & 0 & 0 \\
\end{array}
\]

\[ \overline{f} = \{ "0" \text{ as } "1" \} = x_0 + x_2x_1 \]

This time it was advantageous to group 0s and invert the output!

William Sandqvist  william@kth.se
Ex. 6.8  Don’t Care

Sometimes, problem is such that certain input combinations is "impossible" and therefore can not occur. Such mintermer (or maxtermer) one denotes man d ("do not care") and use them as ones or zeros depending on what works best to get as big groupings as possible.

\[ f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \overline{f} = ? \]

(A risk may be that what is thought to be "impossible" still occurs!? Therefore, it may often be better to take care of all combinations.)
Ex. 6.8

\[ f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \bar{f} = ? \]

| \( x_3 \) | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| \( x_2 \) | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \( x_1 \) | 4 | 5 | 7 | 6 | 12 | 13 | 15 | 14 | 8 | 9 | 11 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |
| \( x_0 \) | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Grouping of "1"

Grouping of "0"
Ex. 6.8

\[ f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \overline{f} = ? \]

Grouping of "1"

<table>
<thead>
<tr>
<th></th>
<th>x_3</th>
<th>x_2</th>
<th>x_1</th>
<th>x_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Grouping of "0"

\[ f = x_1 x_0 + \overline{x_3} x_2 x_0 \]
Ex. 6.8

\[ f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \overline{f} = ? \]

**Grouping of "1"**

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>13</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

\[ f = x_1 x_0 + \overline{x_3} x_2 x_0 \]

**Grouping of "0"**

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>13</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>
Ex. 6.8

\[ f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \overline{f} = ? \]

Grouping of "1"

\[
\begin{array}{cccc}
    x_1 x_0 & \overline{x}_3 x_2 x_0 \\
\hline
    x_3 & 00 & 01 & 11 & 10 \\
    0 & 0 & 1 & 1 & 0 \\
    0 & 4 & 5 & 7 & 6 \\
    1 & 0 & 1 & 1 & - \\
    1 & 12 & 13 & 15 & 14 \\
    1 & 0 & 0 & 1 & 0 \\
    0 & 8 & 9 & 11 & 10 \\
\end{array}
\]

\[ f = x_1 x_0 + \overline{x}_3 x_2 x_0 \]

Grouping of "0"

\[
\begin{array}{cccc}
    x_1 x_0 & \overline{x}_2 \overline{x}_1 \\
\hline
    x_3 & 00 & 01 & 11 & 10 \\
    0 & 0 & 1 & 1 & 0 \\
    0 & 4 & 5 & 7 & 6 \\
    1 & 0 & 1 & 1 & - \\
    1 & 12 & 13 & 15 & 14 \\
    1 & 0 & 0 & 1 & 0 \\
    0 & 8 & 9 & 11 & 10 \\
\end{array}
\]

\[ \overline{f} = \overline{x}_0 + \overline{x}_2 x_1 + x_3 \overline{x}_1 \]
Alarm for water tank

USE 3 PAIR TELEPHONE WIRE TO CONNECT THIS UNIT

William Sandqvist  william@kth.se
Ex. 8.2

X
0111 (7) → 011 (3)

Tank Level Logic

U

0, 1, 2, 3, 4

BCD-display

William Sandqvist  william@kth.se
Ex. 8.2

<table>
<thead>
<tr>
<th>X</th>
<th>x_3</th>
<th>x_2</th>
<th>x_1</th>
<th>x_0</th>
<th>U</th>
<th>u_2</th>
<th>u_1</th>
<th>u_0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>0111 (7) → 011 (3)</th>
</tr>
</thead>
</table>

Tank
Level
Logic

\[
\{ u_2, u_1, u_0 \} \rightarrow U
\]
Only the in-combinations $X 0, 1, 3, 7, 15$ can occur. All other in-combinations can be used as "don’t care".

We can directly see from the table that $u_2$ and $x_3$ are same, $u_2$ can be directly connected to $x_3$. $u_2 = x_3$.

The other expressions are obtained by using their Karnaugh maps.
Ex. 8.2

<table>
<thead>
<tr>
<th>$X$</th>
<th>$x_3$</th>
<th>$x_2$</th>
<th>$x_1$</th>
<th>$x_0$</th>
<th>$U$</th>
<th>$u_2$</th>
<th>$u_1$</th>
<th>$u_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Only the in-combinations $X \ 0, 1, 3, 7, 15$ can occur. All other in-combinations can be used as "don’t care".

We can directly see from the table that $u_2$ and $x_3$ are same, $u_2$ can be directly connected to $x_3$. $u_2 = x_3$.

The other expressions are obtained by using their Karnaugh maps.

William Sandqvist william@kth.se
Ex. 8.2

\[
\begin{array}{c|cccc|cccc}
X & x_3 & x_2 & x_1 & x_0 & U & u_2 & u_1 & u_0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 1 & 1 & 2 & 0 & 1 & 0 \\
7 & 0 & 1 & 1 & 1 & 3 & 0 & 1 & 1 \\
15 & 1 & 1 & 1 & 1 & 4 & 1 & 0 & 0 \\
\end{array}
\]

\[u_1 = \overline{x_3} \cdot x_1\]

\[u_0 = \overline{x_3} \cdot x_2 + \overline{x_1} \cdot x_0\]
Ex. 8.2

\[ u_2 = x_3 \]

\[ u_1 = \overline{x_3} \cdot \overline{x_1} \]

\[ u_0 = \overline{x_3} \cdot x_2 + \overline{x_1} \cdot x_0 \]

Diagram:

- **Input Signals:**
  - \( x_3 \)
  - \( x_1 \)
  - \( x_2 \)
  - \( x_0 \)

- **Logical Operations:**
  - **AND Gates:**
    - \( \overline{x_3} \cdot \overline{x_1} \)
    - \( \overline{x_3} \cdot x_2 \)
    - \( \overline{x_1} \cdot x_0 \)

- **Output Signals:**
  - \( u_2 \)
  - \( u_1 \)
  - \( u_0 \)
P and N MOS-transistors

William Sandqvist  william@kth.se
Ex. 7.3 CMOS-gate ?
Ex. 7.3  A=0  B=0
Ex. 7.3  A=0  B=0
Ex. 7.3  A=0  B=0
Ex. 7.3    A=0  B=0
Ex. 7.3   A=0  B=0

Y = 0

William Sandqvist  william@kth.se
Ex. 7.3  A=0  B=1

Y = 1

William Sandqvist  william@kth.se
Ex. 7.3  A=1  B=1

\[ Y = 1 \]
7.3

### OR-gate

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

William Sandqvist william@kth.se
7.3 other solution

\[ X = \overline{Y} \]

Inverter!

William Sandqvist  william@kth.se
\[ X = \overline{A} \cdot \overline{B} \quad \overline{X} = \overline{\overline{A} \cdot \overline{B}} = A + B \]

\[ X = \overline{Y} \quad \Rightarrow \quad Y = \overline{X} \]

\[ \overline{X} = A + B \]

William Sandqvist  william@kth.se
7.3

\[ Y = A + B \]
Ex. 7.4 CMOS-gate
7.4 EN = 1

When EN = 1 we have an inverter.

\[ Y = \overline{A} \]
7.4 EN = 0

When EN = 0 the output is totally disconnected from the supply voltage and ground. A can no longer influence the output value.

This is a third output state, "Three State".

Many outputs could be connected to the same line ("bus"). One of the outputs at a time can be active. The other are in their Three-state condition.
Outputs from three different devices are connected to a common wire (BUS). The computer selects (Enables, EN) one at a time to be connected to the bus. The other two remain disconnected, (Three state).
Ex. 7.5 CMOS-gate

Construct the "Pull-up" network with P-transistors for the CMOS gate.
Pull-down circuit is the inverted function. The Pull-up circuit is the function noninverted:

$$\bar{Y} = A \cdot C + B \quad \Rightarrow \quad Y = \frac{\bar{A} \cdot \bar{C} + B}{\bar{A} \cdot \bar{C} \cdot \bar{B}} = \frac{(\bar{A} + \bar{C}) \cdot \bar{B}}{\bar{A} \cdot \bar{C} \cdot \bar{B}}$$
The Pull-up net must therefore consist of A and C in parallel (+) then connected in series (⋅) with B. The use of PMOS transistors inverts the variables A, B and C.

\[
\overline{Y} = A \cdot C + B \quad \Rightarrow \quad Y = \overline{A \cdot C + B} = \overline{A \cdot C} \cdot \overline{B} = (\overline{A} + \overline{C}) \cdot B
\]
A MOS-transistor "on chip"

MOS-transistor step by step:

http://micro.magnet.fsu.edu/electromag/java/transistor/
Moore than 2,000,000,000 MOS-transistors/chip!

Pentium 4 has 50,000,000 MOS-transistors