NAPIER.
University
School of Engineering

Electronic Systems
Module : SE32102

Digital Assignment
Design of a Gray Code Generator and Decoder

By
Klaus Jørgensen.
Napier No. 04007824

Teacher.
Jay Hoy
Abstract:

In this paper is there a little background about the Gray code. There is also showed one way to build a Gray code generator from 0 to 15, with State Transition Diagram, State Table, Karnaugh maps, reduction of boolean algebra, circuit and simulations, the same is done for a Gray code to binary decoder, and the two circuits is also tested together.
Table of contents:

Introduction:........................................................................................................................................... 4

Theoretical Design of a Gray Code Generator. ............................................................................. 5

State Transition Diagram. .................................................................................................................. 5

State Table of the Gray code generator......................................................................................... 6

Karnaugh maps for the Gray code generator............................................................................... 6

Gray code generator Circuit schematic and Simulation. ........................................................... 9

Gray Code to Binary Decoder........................................................................................................ 10

State Transition Diagram. ............................................................................................................... 10

State Table of the Gray code generator........................................................................................ 10

Karnaugh maps for the Gray code to binary decoder................................................................. 11

Gray code to binary decoder Circuit schematic and Simulation................................................. 12

Total circuit and simulation of Gray code generator and binary decoder................................. 13

Conclusion. ....................................................................................................................................... 14

Reference. .......................................................................................................................................... 15
Introduction:
Gray code was invented for practical reasons. Because it only changes one bit at the time, it can be used to determine the position of a cogwheel on a big machine for instance. The gray code or binary code is printed on the cogwheel, and some sensors are reading the signals, and transmit the signal to the software.
If the binary code is printed on the cogwheel, there is a bigger possibility for an error in the reading of the code, because the binary code at some times changes more than one bit at the time, for example the change from 7 (0111) to 8 (1000) figure 2, where all 4 bits has to change, there is a big possibility for an error, if there is a little delay in the circuit.
If the gray code is used instead of the binary code there is a much smaller possibility for an error because the gray code only changes one bit at the time [3, 4].
Figure 1 shows an example of how the binary code or gray code can look like on a cogwheel.

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

Figure 1

Figure 2
Theoretical Design of a Gray Code Generator.

The 16 bit Gray code generator is made by 4 JK flip-flops and 10 gates, one JK flip-flop for each output, QA is the MSB (Most Significant Bit) and QD is the LSB (Least Significant Bit).

Figure 3 shows a chart of how the JK flip-flop works.
Q is the Present state, Q+ is the Next state, J and K is the inputs of the flip-flop, to get from Q to Q+, J and K have to get the value as shown in figure 3, the x is a don’t care state is can be 1 or 0 it doesn’t matter [1].

<table>
<thead>
<tr>
<th>Q</th>
<th>Q+</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3

State Transition Diagram.

To get a view of how the Gray code generator must work a State Transition diagram is made, it showed in figure 4.
State Table of the Gray code generator.

Figure 5 shows the State table for the Gray code generator, it shows the Present, Next and the Change state of the Gray code generator.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Change State</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA QB QC QD</td>
<td>QA+ QB+ QC+ QD+ JA KA JB KB JC KC JD KD</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 1</td>
<td>0 x 0 x 0 x 1 x</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>0 0 1 1</td>
<td>0 x 0 x 1 x x 0</td>
</tr>
<tr>
<td>2 0 1 1</td>
<td>0 1 1 0</td>
<td>0 x 1 x x 0 0 x</td>
</tr>
<tr>
<td>3 0 1 0</td>
<td>0 1 1 1</td>
<td>0 x x 0 x 0 1 x</td>
</tr>
<tr>
<td>4 0 1 1</td>
<td>0 1 0 1</td>
<td>0 x x 0 x 1 x 0</td>
</tr>
<tr>
<td>5 1 1 1</td>
<td>0 1 0 0</td>
<td>0 x x 0 0 x x 1</td>
</tr>
<tr>
<td>6 1 0 1</td>
<td>1 1 0 0</td>
<td>1 x 1 x x 0 0 x</td>
</tr>
<tr>
<td>7 1 0 0</td>
<td>1 1 0 1</td>
<td>1 x 0 x 0 0 x 1 x</td>
</tr>
<tr>
<td>8 1 1 1</td>
<td>1 1 1 0</td>
<td>x 0 x 0 x 0 x 1 x</td>
</tr>
<tr>
<td>9 1 1 0</td>
<td>1 1 1 1</td>
<td>x 0 x 0 1 x x 0 x</td>
</tr>
<tr>
<td>10 1 0 0</td>
<td>1 0 1 0</td>
<td>x 0 0 x x x x 0 x</td>
</tr>
<tr>
<td>11 1 0 1</td>
<td>1 0 0 1</td>
<td>x 0 0 x 0 x 1 x 0 x</td>
</tr>
<tr>
<td>12 1 0 0</td>
<td>1 0 0 0</td>
<td>x 0 0 x 0 x x 1 x 0 x</td>
</tr>
<tr>
<td>13 0 0 1</td>
<td>1 0 1 1</td>
<td>x 0 0 x x x x 1 x 0 x</td>
</tr>
<tr>
<td>14 0 0 0</td>
<td>1 0 0 0</td>
<td>x 0 0 x 0 x 0 x 1 x 0 x</td>
</tr>
<tr>
<td>15 1 0 0</td>
<td>0 0 0 0</td>
<td>x 1 0 x 0 x 0 x 0 x</td>
</tr>
</tbody>
</table>

Karnaugh maps for the Gray code generator.

The karnaugh maps and the boolean algebra expression is shown below, with the circuit for the boolean algebra.

**JA**

\[
\begin{array}{c|cccc}
QCQC & 00 & 01 & 11 & 10 \\
\hline
00 & 0 & \textcolor{red}{1} & x & x \\
01 & 0 & 0 & x & x \\
11 & 0 & 0 & x & x \\
10 & 0 & 0 & x & x \\
\end{array}
\]

\[JA = QBQCQD\]

**KA**

\[
\begin{array}{c|cccc}
QCQC & 00 & 01 & 11 & 10 \\
\hline
00 & x & x & 0 & 1 \\
01 & x & x & 0 & 0 \\
11 & x & x & 0 & 0 \\
10 & x & x & 0 & 0 \\
\end{array}
\]

\[KA = QBQCQD\]
JB

\[
\begin{array}{c|ccccc}
QDQB & 00 & 01 & 11 & 10 \\
\hline
00 & 1 & x & x & 0 \\
01 & x & x & 1 & 0 \\
11 & 0 & x & x & 0 \\
10 & 0 & x & x & 0 \\
\end{array}
\]

\[JB = QAQCQB\]

KB

\[
\begin{array}{c|ccccc}
QDQB & 00 & 01 & 11 & 10 \\
\hline
00 & x & 0 & 0 & x \\
01 & x & 0 & 1 & x \\
11 & x & 0 & 0 & x \\
10 & x & 0 & 0 & x \\
\end{array}
\]

\[KB = QAQCQB\]

JC

\[
\begin{array}{c|ccccc}
QDQB & 00 & 01 & 11 & 10 \\
\hline
00 & 0 & 0 & 0 & 0 \\
01 & x & x & x & x \\
11 & x & x & x & x \\
10 & 1 & 0 & 0 & 0 \\
\end{array}
\]

\[JC = QAQBQB + QAQBQB\]

\[= (QA \oplus QB)*QD\]

KC

\[
\begin{array}{c|ccccc}
QDQB & 00 & 01 & 11 & 10 \\
\hline
00 & x & x & x & x \\
01 & 0 & 0 & 0 & 0 \\
11 & 0 & 1 & 0 & 1 \\
10 & x & x & x & x \\
\end{array}
\]

\[KC = QAQBQB + QAQBQB\]

\[= (QA \oplus QB)*QD\]

\[=1\]
Design of a Gray Code Generator and Decoder

### JD

<table>
<thead>
<tr>
<th>QDQC</th>
<th>QM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>x  x</td>
</tr>
<tr>
<td>1 0</td>
<td>x  x</td>
</tr>
</tbody>
</table>

JD = \( \overline{QAQBQC} + \overline{QAQBQC} + QAQBQC + QAQBQC \Rightarrow \)

\( (QAQB)^*QC + (QAQB)^*QC = (QA \oplus QB)^*QC \Rightarrow \)

\( (QAQB)^*QC + (QAQB)^*QC = (QA \oplus QB)^*QC \Rightarrow \)

\( QA \oplus QB \oplus QC \Rightarrow XOR + XOR \)

**Figure 12**

### KD

<table>
<thead>
<tr>
<th>QDQC</th>
<th>QM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>x  x</td>
</tr>
<tr>
<td>0 1</td>
<td>x  x</td>
</tr>
<tr>
<td>1 1</td>
<td>1 0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 1</td>
</tr>
</tbody>
</table>

KD = \( QAQBQC + QAQBQC + QAQBQC + QAQBQC \Rightarrow \)

\( (QAQB)^*QC + (QAQB)^*QC = (QA \oplus QB)^*QC \Rightarrow \)

\( QA \oplus QB \oplus QC \Rightarrow XOR + XOR \)

**Figure 13**
Gray code generator Circuit schematic and Simulation.

To make the Gray code generator on figure 14, there are used 4 Flip-Flops, 10 And-gates, 3 Or-gates and 1 inverter.

The simulation of the Gray code generator in figure 14 is shown in figure 15. The clock is on 1MHz, the length of one clock pulse is 1µs.

- $T_p = \frac{1}{f} \Rightarrow \frac{1}{1M} = 1\mu s$

The simulation on figure 15 shows the same pattern as the State table for the Gray code generator in figure 5 on page 6.
Gray Code to Binary Decoder.

State Transition Diagram.

To get a view of how the Gray code to binary decoder must work a State Transition diagram is made, to see what the output of the Gray code to binary decoder shall show, it showed on figure 16.

State Table of the Gray code generator.

Figure 17 shows the State table for the Gray code to binary decoder, it shows the input from the Gray code generator and the output from the Gray code to binary decoder.

<table>
<thead>
<tr>
<th>Gray Code Input</th>
<th>Binary Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA QB QC QD</td>
<td>A B C D</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>2 0 1 1</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>3 0 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>4 0 0 1</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>5 0 1 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>6 0 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>7 0 0 1</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>8 1 1 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>9 1 1 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>10 1 1 1</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>11 1 1 1</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>12 1 0 1</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>13 1 0 1</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>14 1 0 0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>15 1 0 0</td>
<td>0 0 1 1</td>
</tr>
</tbody>
</table>
Karnaugh maps for the Gray code to binary decoder.

The karnaugh maps and the boolean algebra expression is shown below, with the circuit for the boolean algebra.

\[ QA \]

\[ \begin{array}{c|cccc}
D & AB \\
\hline
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
\end{array} \]

\[ QA = A \]

\[ QB \]

\[ \begin{array}{c|cccc}
D & AB \\
\hline
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
\end{array} \]

\[ QB = \overline{AB} + \overline{A}B \]

\[ QC \]

\[ \begin{array}{c|cccc}
D & AB \\
\hline
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
\end{array} \]

\[ QC = \overline{C} * (\overline{A}B + AB) + \overline{C} * (\overline{AB} + AB) \Rightarrow \]

\[ \overline{C} * A + B + \overline{C} * A + B \Rightarrow \]

\[ \overline{C} \oplus (A + B + A + B) \Rightarrow \]

\[ A + B + C \Rightarrow 3 \text{ input XOR} \]

\[ QD \]

\[ \begin{array}{c|cccc}
D & AB \\
\hline
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
\end{array} \]

\[ QD = AB * (\overline{CD} + \overline{CD}) + \overline{AB} * (CD + CD) + \overline{AB} * (\overline{CD} + \overline{CD}) \Rightarrow \]

\[ \overline{AB} * C + D + \overline{AB} * \overline{C} + D + \overline{AB} * C + D + \overline{AB} * \overline{C} + D \Rightarrow \]

\[ A + B * C + D + A + B * C + D \Rightarrow \]

\[ A + B + C + D \Rightarrow 4 \text{ input XOR} \]
Gray code to binary decoder Circuit schematic and Simulation.

There is only used 3 Exclusive OR gates to make the Gray code to binary decoder, and it is showed in figure 22 and the simulations result is showed in figure 23. To make the simulation a 4-bit data generator was used to make the Gray code from 0 to 15, as it is showed in figure 17 on page 10.
Total circuit and simulation of Gray code generator and binary decoder.

The output FA, FB, FC and FD is the output from the JK flip-flops and is the put from the Gray code generator, the output BA, BB, BC and BD is the output from the Gray code to binary code decoder.
Conclusion.

The gray code generator is made of 4 JK flip-flops and 10 gates, U13 (Exclusive OR) and U14 (Inverter) makes an Exclusive NOR gate this it don because there is no Exclusive NOR gate in Tina, U17 and U18 was meant to be 3 inputs Exclusive OR gates but in Tina there only is 2 inputs Exclusive OR gates, if all this was change the numbers of gate still would be 10.

If the gray code generator should be build in practical, the 4 And gates with 3 inputs (U5, U6, U7, U9) would be placed in 2 IC’s with each 3 And gates, and the 2 And gates with 2 inputs would be placed in one IC with 4 And gates, to save one IC, the 2 And gates with 2 inputs (U15, U16) can be switch with the 2 And gates with 3 inputs, and one input of each And gate, can be connected to the 5V power a logic 1.

The gray code to binary code is made of 3 Exclusive OR gates with 2 inputs each, the decoder could have been made with 1 Exclusive OR gates with 2 inputs, 1 Exclusive OR gates with 3 inputs, Exclusive OR gates with 4 inputs, but this would not be are good way to make the circuit in practical, because the circuit would be made of 3 IC’s, if the circuit is made with 3 Exclusive OR gates with 2 inputs each there only have to be used one IC.

The complete circuit on page 13 figure 24 works fine, it makes a gray code from 0 to 15 and the decoder from gray code to binary code also works, as the diagram of the outputs shows on page 13, figure 24.

Klaus Jørgensen 23 November 2004
Reference.

   ISBN : 87-600-0126-7

2. Introduction to Electrical Engineering, by Mulukutla S. Sarma,

3. Digital System Design second edition, by Barry Wilkinson
   ISBN : 0-13-220286-7

4. Digital Electronics an introduction to theory and practice second edition
   by William H. Gothmann
   ISBN : 0-13-212084-4