Braille Number Encoder Circuit

PC/CP220 Project Phase II

Mayar Tharowat

Truth Table:

By making truth table for our Inputs (I_0 to I_3) and Outputs (P_0 to P_5), we can write out equations easily.

Inputs				Outputs						Num
I_0	I_1	I_2	I_3	P_0	P_1	P_2	P_3	P_4	P_5	
0	0	0	0	0	1	0	0	1	1	**0**
0	0	0	1	1	0	0	0	0	0	**1**
0	0	1	0	1	1	0	0	0	0	**2**
0	0	1	1	1	0	0	0	0	1	**3**
0	1	0	0	1	0	0	0	1	1	**4**
0	1	0	1	1	0	0	0	1	0	**5**
0	1	1	0	1	1	0	0	0	1	**6**
0	1	1	1	1	1	0	0	1	1	**7**
1	0	0	0	1	1	0	0	1	0	**8**
1	0	0	1	0	1	0	0	0	1	**9**
1	0	1	0	x	x	x	x	x	x	10
1	0	1	1	x	x	x	x	x	x	11
1	1	0	0	x	x	x	x	x	x	12
1	1	0	1	x	x	x	x	x	x	13
1	1	1	0	x	x	x	x	x	x	14
1	1	1	1	x	x	x	x	x	x	15

Table 1: Truth Table

According to the truth table above, we can get sum-of-products(SOP) logic equations. And it is possible to simplify sum-of-products(SOP) logic equations by using a Karnaugh map.

For the sum-of-products(SOP) form, it will be

P_0: $(\overline{i_0}\,\overline{i_1}\,\overline{i_2}\,i_3) + (\overline{i_0}\,\overline{i_1}\,i_2\,\overline{i_2}) + (\overline{i_0}\,\overline{i_1}\,i_2\,i_3) + (\overline{i_0}\,i_1\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,i_1\,\overline{i_2}\,i_3) + (\overline{i_0}\,i_1\,i_2\,\overline{i_3}) + (\overline{i_0}\,i_1\,i_2\,i_3) + (i_0\,\overline{i_1}\,\overline{i_2}\,\overline{i_3})$

P_1: $(\overline{i_0}\,\overline{i_1}\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,\overline{i_1}\,i_2\,\overline{i_3}) + (\overline{i_0}\,i_1\,i_2\,\overline{i_3}) + (\overline{i_0}\,i_1\,i_2\,i_3) + (i_0\,\overline{i_1}\,\overline{i_2}\,\overline{i_3}) + (i_0\,\overline{i_1}\,\overline{i_2}\,i_3)$

P_2: None

P_3: None

P_4: $(\overline{i_0}\,\overline{i_1}\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,i_1\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,i_1\,\overline{i_2}\,i_3) + (\overline{i_0}\,i_1\,i_2\,i_3) + (i_0\,\overline{i_1}\,\overline{i_2}\,\overline{i_3})$

P_5: $(\overline{i_0}\,\overline{i_1}\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,\overline{i_1}\,i_2\,i_3) + (\overline{i_0}\,i_1\,\overline{i_2}\,\overline{i_3}) + (\overline{i_0}\,i_1\,i_2\,\overline{i_3}) + (\overline{i_0}\,i_1\,i_2\,i_3) + (i_0\,\overline{i_1}\,\overline{i_2}\,i_3)$

Then, we can make the Karnaugh map based on the SOP form of outputs (from P_0 to P_5).

P_0:

		$i_2\,i_3$			
		00	01	11	10
$i_0\,i_1$	00	0	1	1	1
	01	1	1	1	1
	11	x	x	x	x
	10	1	0	x	x

Table 2: Karnaugh Map Table for P_0

P_1:

		$i_2\,i_3$			
		00	01	11	10
$i_0\,i_1$	00	1	0	0	1
	01	0	0	1	1
	11	x	x	x	x
	10	1	1	x	x

Table 3: Karnaugh Map Table for P_1

P_2: None

P_3: None

P_4:

		$i_2\ i_3$			
		00	01	11	10
$i_0\ i_1$	00	1	0	0	0
	01	1	1	1	0
	11	x	x	x	x
	10	1	0	x	x

Table 4: Karnaugh Map Table for P_4

P_5:

		$i_2\ i_3$			
		00	01	11	10
$i_0\ i_1$	00	1	0	1	0
	01	1	0	1	1
	11	x	x	x	x
	10	0	1	x	x

Table 5: Karnaugh Map Table for P_5

If we simplify the SOP forms of outputs (from P_0 to P_5), we will get reduced logic equations as followed below:

P_0: $i_2 + i_1 + \overline{i_0}\ i_3 + \overline{i_0}\ \overline{i_3}$

P_1: $\overline{i_1}\ \overline{i_3} + i_1\ i_2$

P_2: None

P_3: None

P_4: $\overline{i_2}\ \overline{i_3} + i_1\ i_3$

P_5: $i_2\, i_3 + i_1\ \overline{i_3} + i_0\ i_3 + \overline{i_0}\ \overline{i_2}\ \overline{i_3}$

Testing Logic:
Since Maxima is useful for testing equations, we used Maxima and entered our logic equations of outputs.

```
Maxima 5.38.1_5_gdf93b7b_dirty http://maxima.sourceforge.net
using Lisp CLISP 2.49 (2010-07-07)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
```

P_0

```
(%i1) p0: ((not i0) and (not i1) and (not i2) and ( i3)) or ((not i0) and (not i1) and
(i2) and (not i3)) or ((not i0) and (not i1) and ( i2) and (i3)) or ((not i0) and (i1)
and (not i2) and (not i3)) or ((not i0) and (i1) and (not i2) and (i3)) or ((not i0) a
nd (i1) and (i2) and (not i3)) or ((not i0) and ( i1) and ( i2) and ( i3)) or ((i0) an
d (not i1) and (not i2) and (not i3));

(%o1) ((not i0) and (not i1) and (not i2) and i3)
 or ((not i0) and (not i1) and i2 and (not i3))
 or ((not i0) and (not i1) and i2 and i3)
 or ((not i0) and i1 and (not i2) and (not i3))
 or ((not i0) and i1 and (not i2) and i3)
 or ((not i0) and i1 and i2 and (not i3)) or ((not i0) and i1 and i2 and i3)
 or (i0 and (not i1) and (not i2) and (not i3))
```

P_1

```
(%i2) p1: ((not i0) and (not i1) and (not i2) and (not i3)) or ((not i0) and (not i1)
and (i2) and (not i3)) or ((not i0) and (i1) and ( i2) and (not i3)) or ((not i0) and
( i1) and ( i2) and ( i3)) or ((i0) and (not i1) and (not i2) and (not i3))or ((i0) an
d (not i1) and (not i2) and ( i3));

(%o2) ((not i0) and (not i1) and (not i2) and (not i3))
 or ((not i0) and (not i1) and i2 and (not i3))
 or ((not i0) and i1 and i2 and (not i3)) or ((not i0) and i1 and i2 and i3)
 or (i0 and (not i1) and (not i2) and (not i3))
 or (i0 and (not i1) and (not i2) and i3)
```

P_4

```
(%i3) p4: ((not i0) and (not i1) and (not i2) and (not i3)) or ((not i0) and (i1) and
(not i2) and (not i3)) or ((not i0) and (i1) and (not i2) and (i3)) or ((not i0) and (
i1) and ( i2) and ( i3)) or ((i0) and (not i1) and (not i2) and (not i3));

(%o3) ((not i0) and (not i1) and (not i2) and (not i3))
 or ((not i0) and i1 and (not i2) and (not i3))
 or ((not i0) and i1 and (not i2) and i3) or ((not i0) and i1 and i2 and i3)
 or (i0 and (not i1) and (not i2) and (not i3))
```

P_5

```
(%i4) p5: ((not i0) and (not i1) and (not i2) and (not i3)) or ((not i0) and (not i1)
and ( i2) and (i3)) or ((not i0) and (i1) and (not i2) and (not i3)) or ((not i0) and
(i1) and (i2) and (not i3)) or ((not i0) and ( i1) and ( i2) and ( i3)) or (( i0) and
(not i1) and (not i2) and (not i3));
```

So we got the following results:
In case of p_0, it was

```
(%i48) p0, i0=false, i1=false,i2=false, i3=false;

(%o48)                          false
(%i49) p0, i0=false, i1=false,i2=false, i3=true;

(%o49)                          true
(%i50) p0, i0=false, i1=false,i2=true, i3=false;

(%o50)                          true
(%i51) p0, i0=false, i1=false,i2=true, i3=true;

(%o51)                          true
(%i52) p0, i0=false, i1=true,i2=false, i3=false;

(%o52)                          true
(%i53) p0, i0=false, i1=true,i2=false, i3=true;

(%o53)                          true
(%i54) p0, i0=false, i1=true,i2=true, i3=false;

(%o54)                          true
(%i55) p0, i0=false, i1=true,i2=true, i3=false;

(%o55)                          true
(%i56) p0, i0=true, i1=false,i2=false, i3=false;

(%o56)                          true
(%i57) p0, i0=true, i1=false,i2=false, i3=true;

(%o57)                          false
(%i58)
```

In case of p_1, it was

```
(%i58) p1, i0=false, i1=false,i2=false, i3=false;

(%o58)                               true
(%i59) p1, i0=false, i1=false,i2=false, i3=true;

(%o59)                               false
(%i60) p1, i0=false, i1=false,i2=true, i3=false;

(%o60)                               true
(%i61) p1, i0=false, i1=false,i2=true, i3=true;

(%o61)                               false
(%i62) p1, i0=false, i1=true,i2=false, i3=false;

(%o62)                               false
(%i63) p1, i0=false, i1=true,i2=false, i3=true;

(%o63)                               false
(%i64) p1, i0=false, i1=true,i2=true, i3=false;

(%o64)                               true
(%i65) p1, i0=false, i1=true,i2=true, i3=false;

(%o65)                               true
(%i66) p1, i0=true, i1=false,i2=false, i3=false;

(%o66)                               true
(%i67) p1, i0=true, i1=false,i2=false, i3=true;

(%o67)                               true
(%i68)
```

In case of p_4, it was

```
(%i28) p4, i0=false, i1=false,i2=false, i3=false;

(%o28)                               true
(%i29) p4, i0=false, i1=false,i2=false, i3=true;

(%o29)                               false
(%i30) p4, i0=false, i1=false,i2=true, i3=false;

(%o30)                               false
(%i31) p4, i0=false, i1=false,i2=true, i3=true;

(%o31)                               false
(%i32) p4, i0=false, i1=true,i2=false, i3=false;

(%o32)                               true
(%i33) p4, i0=false, i1=true,i2=false, i3=true;

(%o33)                               true
(%i34) p4, i0=false, i1=true,i2=true, i3=false;

(%o34)                               false
(%i35) p4, i0=false, i1=true,i2=true, i3=true;

(%o35)                               true
(%i36) p4, i0=true, i1=false,i2=false, i3=false;

(%o36)                               true
(%i37) p4, i0=true, i1=false,i2=false, i3=true;

(%o37)                               false
(%i38)
```

In case of p_5, it was

```
(%i38) p5, i0=false, i1=false,i2=false, i3=false;

(%o38)                              true
(%i39) p5, i0=false, i1=false,i2=false, i3=true;

(%o39)                              false
(%i40) p5, i0=false, i1=false,i2=true, i3=false;

(%o40)                              false
(%i41) p5, i0=false, i1=false,i2=true, i3=true;

(%o41)                              true
(%i42) p5, i0=false, i1=true,i2=false, i3=false;

(%o42)                              true
(%i43) p5, i0=false, i1=true,i2=false, i3=true;

(%o43)                              false
(%i44) p5, i0=false, i1=true,i2=true, i3=false;

(%o44)                              true
(%i45) p5, i0=false, i1=true,i2=true, i3=true;

(%o45)                              true
(%i46) p5, i0=true, i1=false,i2=false, i3=false;

(%o46)                              true
(%i47) p5, i0=true, i1=false,i2=false, i3=true;

(%o47)                              false
```

Depending on the results above, we got "true" for all of true cases, otherwise it was "false" for all of false cases. Therefore, it appears that the equations were correct.