1. Create a Karnaugh map for output b1. Use this Karnaugh map to determine the simplified SOP

expressions for b1. Repeat this process to find the simplified POS expression for b1.

**B1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| g3,g2\g1,g0 | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 1 | 1 | 0 | 0 |

**B1 = (g3’)\*g2\*(g1’) + (g3’)\*(g2’)\*g1 + g3\*g2\*g1 + g3\*(g2’)\* (g1’)**

**CPOS B1 = (g3 + g2 + g1)\*(g3 + g2’ + g1’)\*(g3’ + g2’ + g1)\*(g3’ + g2 + g1’)**

2. Compare the pre-fit equations and post-fit equations from ispLEVER with your simplified SOP

and POS expressions obtained from your Karnaugh maps. Are they equivalent? If not, how do they differ?

The post and pre fit equation are exactly the same as each other. Compared to my karnaugh map derived equations they are the exact same for CSOP. For POS the inverted equation was the logical inversion of my equation. This is because the inverted equation showed !b1 while my equation represents b1. By applying demorgan’s law to their equation it can be seen that they are logical inverses.

Interestingly in the CPOS place and route pre and post fit equations used SOP logic. This is likely because it was easier to implement

notes:

a. In the equations in ispLEVER, the # symbol represents OR, the & symbol represents AND, and

the ! symbol represents NOT.

b. The pre-fit and post-fit equations listings give equations for both the normal and complement of

each function. This is done because it is sometimes easier (requires fewer product terms) to implement the complement of a function and then configure the PLD to complement the output of the

OR gate creating the function than it is to implement the normal form of the function.

3. From the chip report determine exactly which equations were actually implemented in the PLD

for each design.

CSOP Chip report: My simplified CSOP equation was implemented for B1 in the CSOP chip report.

**B3 = G3**

**B2 = g3\*(g2’) + (g3’) \* g2**

**B1 = (g3’)\*g2\*(g1’) + (g3’)\*(g2’)\*g1 + g3\*g2\*g1 + g3\*(g2’)\* (g1’)**

**B0 = g3’ \* g2’ \*g1’\*g0 + g3’ \* g2’ \*g1\*g0’ +g3’ \* g2 \*g1\*g0 + g3’ \* g2 \*g1’\*g0’ +g3 \* g2 \*g1’\*g0 + g3 \* g2 \*g1\*g0’ +g3 \* g2’ \*g1\*g0 + g3 \* g2’ \*g1’\*g0’**

CPOS Chip report:

**B3 = G3**

**B2 = g3\*(g2’) + (g3’) \* g2**

**B1 = (g3’)\*g2\*(g1’) + (g3’)\*(g2’)\*g1 + g3\*g2\*g1 + g3\*(g2’)\* (g1’)**

**B0 = g3’ \* g2’ \*g1’\*g0 + g3’ \* g2’ \*g1\*g0’ +g3’ \* g2 \*g1\*g0 + g3’ \* g2 \*g1’\*g0’ +g3 \* g2 \*g1’\*g0 + g3 \* g2 \*g1\*g0’ +g3 \* g2’ \*g1\*g0 + g3 \* g2’ \*g1’\*g0’**

4. Compare the “Technology Flattened to gates view” schematics for the two designs. Are the two

schematics identical? If not, are they logically the same?

The two schematics are identical and logically the same, just like the chip report equations.

5. From the HDL Analyst “Technology Flattened to gates view” schematic for your CSOP design

write the Boolean equations for outputs b3 and b2 and b1 as implemented by the synthesizer.

B3 = G3

B2 = G3 xor G2

B1 = !(g1 xor (g2 xnor g3 ))