Seungtack Baek

CE6305-501

Homework3

1. First, determine logic for ,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | x |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | x |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | x |
| 1 | 1 | 0 | 0 | x |
| 1 | 1 | 0 | 1 | x |
| 1 | 1 | 1 | 0 | x |
| 1 | 1 | 1 | 1 | x |

Quine-McCluskey yields  and applying De’Morgan’s theorem, we get , which corresponds to the following diagram with delay *6I*. Yet, one correction will be that, since the fan-out of  will be two (we don’t know yet, but will soon), the delay to get  is *I+2I+3·2I = 9I*. So far, we have Yet, instead of using  straight, we can use buffer, which is only two inverters back to back, so  delay now is only *I+2I+3I+2I=* ***8I***



Now, we produce carry out, , and intermediate sum, , for the next stage, given  signal from previous stage.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | x | x | x | x |
| 0 | 0 | 1 | 1 | 1 | x | x | x | x |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x |
| 1 | 1 | 0 | 0 | 0 | x | x | x | x |
| 1 | 1 | 0 | 0 | 1 | x | x | x | x |
| 1 | 1 | 0 | 1 | 0 | x | x | x | x |
| 1 | 1 | 0 | 1 | 1 | x | x | x | x |
| 1 | 1 | 1 | 0 | 0 | x | x | x | x |
| 1 | 1 | 1 | 0 | 1 | x | x | x | x |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x |

Quine-McCluskey yields





Both will result in a delay of ***8I***. as shown below

Now, we can observe that the delay for a signal with OR’ing *N* minterms with *M* terms is given by 1(inversion delay) + *N* + *M*, if only NAND and NOT gates are use.

Applying this fact to both  and , which both have 5 minterms 5 terms, we get *11I* for both signals. Yet again, the fan-out  is 2, instead of making the delay of last NAND to 6I, we will apply buffer. So, the delay to get  is ***13I***.

Lastly, we can get the signal delay for 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | x | x |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | x | x |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | x | x |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | x | X |
| 1 | 0 | 1 | 1 | x | x |
| 1 | 1 | 0 | 0 | x | X |
| 1 | 1 | 0 | 1 | x | X |
| 1 | 1 | 1 | 0 | x | X |
| 1 | 1 | 1 | 1 | x | X |



Applying the acquired fact, we can see that this signal is obtained by OR’ing 3 minterms with 2 terms. The delay to get the  is given by *I+2I+3I* = ***6I***.

Finally, the delay for the critical path is 8I + 13I + 6I = ***27I***.

2. Since we have  with , , we can determine the range of . Calculating the minimum range,

 so the minimum value of  is 1.

 so the minimum value of  is 1.

Thus, the minimum range of  is **[-1, 1]**.

Now, finding breakpoints for the ouput and transmission signal for , we construct a table,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Option 1 | | | Option 2 | | |
|  |  |  |  |  |  |  |  |
| -16 | [-17,-15] | [-1,1] | 0 | -1 |  |  |  |
| … | … | … | … | … | … | … | … |
| -8 | [-8,-6] | [7,9] | 8 | -1 |  |  |  |
| -7 | [-8,-6] | [8,10] | 9 | -1 | [-8,-6] | -7 | 0 |
| -6 | [-7,-5] | [9,11] | 10 | -1 | [-7,-5] | -6 | 0 |
| -5 | [-6,-4] | [10,12] | 11 | -1 | [-6,-4] | -5 | 0 |
| -4 | [-5,-3] |  |  |  | [-5,-3] | -4 | 0 |
| … | … | … | … | … | … | … | … |
| 8 | [7,9] |  |  |  | [7,9] | 8 | 0 |
| 9 | [8,10] | [-8,-6] | -7 | 1 | [8,10] | 8 | 0 |
| 10 | [9,11] | [-7,-5] | -6 | 1 | [9,11] | 8 | 0 |
| 11 | [10,12] | [-6,-4] | -5 | 1 | [10,12] | 8 | 0 |
| 12 | [11,13] | [-5,-3] | -4 | 1 |  |  |  |

Now, determining breakpoint between 0 and 1 first, we can choose 9, 10 or 11. Yet, we choose 9 because we can compare  with 8, which is a power of 2 thus easier to compare. Next, we determine breakpoint between -1 and 0. We can choose either -5 or -7, for the same reason above. Yet, choosing -7 so we can compare  with -8 will be more consistent to our choice above. So,

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | [-16,-8] | [-7,8] | [9,24] |
|  | -1 | 0 | 1 |

3. There can only be **one representation of 0** in redundant number system. Each digit should not exceed the radix in either positively or negatively, no two digits cannot cancel each other to make zero. So only one representation is possible.