**Minimize test power for c6288 in 45 nm CMOS by optimal ordering of vectors**

By Dang Li

903509309

ISCAS-85 benchmark circuit c6288 16 x 16 multiplier multiplies 2 16-bit bus bit patterns and results in a 32-bit output. The figure below shows how the 2406 gates form 240 full and half adder cells arranged in a 15 x 16 matrix.



Figure 1 ISCAS-85 C6288 16x16 Multiplier

Using Integer Linear Programming (ILP) we found a set of 10 test vectors. These vectors were found by Dr. Vishwani Agrawal and Kalyana R. Kantipudi (Ref 8). These 10 vectors are currently the best possible solution set for benchmark circuit c6288. These vectors are shown below in figure 2.

|  |  |
| --- | --- |
| **Vector No.** | **16-bit multiplier** |
| **V1** | 1101101101101101 1101111111111111 |
| **V2** | 0110110110110110 1111111111111111 |
| **V3** | 0000000000000000 0010111111111111 |
| **V4** | 1011011011011011 1011111111111111 |
| **V5** | 11111111111111111 101010101010101 |
| **V6** | 11111111111111110 110101010101010 |
| **V7** | 00111111111111011 101010101010101 |
| **V8** | 00111111111111011 010101010101011 |
| **V9** | 11101101101101100 010111111111111 |
| **V10** | 11011011011011001 010101010101010 |

Figure 2.

In demanding of reducing test power in c6288 circuit, we made the assumption that the least number of input transitions will equal to the least number of circuit power consumption. Throughout previous steps we have already found the least number of test vectors. Next step we will try to optimize the ordering of the test vectors so that it will provide the least number of input transitions.

In this step we use Matlab and the source code is shown below.

Figure 3. Codes and a part of output result.

We put the result from Matlab into figure 4 below to show the total number of Hamming transitions between test vectors.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** | **V10** |
| **V1** | / | 12 | 15 | 10 | 11 | 14 | 12 | 14 | 14 | 10 |
| **V2** | 12 | / | 13 | 12 | 13 | 14 | 14 | 14 | 4 | 18 |
| **V3** | 15 | 13 | / | 15 | 26 | 23 | 23 | 19 | 11 | 17 |
| **V4** | 10 | 12 | 15 | / | 11 | 14 | 12 | 14 | 14 | 20 |
| **V5** | 11 | 13 | 26 | 11 | / | 15 | 3 | 17 | 15 | 21 |
| **V6** | 14 | 14 | 14 | 23 | 14 | / | 18 | 6 | 12 | 8 |
| **V7** | 12 | 14 | 23 | 12 | 3 | 18 | / | 14 | 18 | 22 |
| **V8** | 14 | 14 | 19 | 14 | 17 | 6 | 14 | / | 14 | 8 |
| **V9** | 14 | 4 | 11 | 14 | 15 | 12 | 18 | 14 | / | 16 |
| **V10** | 10 | 18 | 17 | 20 | 21 | 8 | 22 | 8 | 16 | / |

Figure 4.

Now the transition distance has been established, we need to find the best ordering to produce the least number of input transitions. Here we’ll be trying two possible approaches as Greedy Heuristic and TSP approach.

First of all for each vector we find the nearest distance. We will assume that by eliminating the longest link we would get the shortest open tour. From the figure shown above we see for V1 it is 10, V2 it is 4, V3 is 11, etc. Through observation we find out that V3 has the longest distance from the nearest vectors. By putting V3 at the starting point we expect to remove the longest distance from the closed tour. Thus we have our trials start from it.

1. V3-V9-V2-V1-V4-V5-V7-V8-V6-V10
2. V3-V9-V2-V4-V1-V19-V6-V8-V7-V5

Listed above are 2 results we obtained from our trials, the reason for them to exist is because after V2 there are two vectors with equal distance from V2. Thus we try both V1 and V4 route after it reached V2. Result 1 gives us 79 Hamming Transitions and result 2 gives us 78 transitions.

Then we use Matlab to examine if the results listed above are the best solutions we may obtain. In this step we will be using TSP solution to try to obtain the expected results. To achieve that here we insert a V11, which has 0 distance from all 10 vectors and we run the program from V11 through all 10 vectors then end up with V11 again. Be aware that only the vector V11 can be visited twice. The result shown below is what we found here.

1. (V11)-V3-V9-V2-V5-V7-V4-V1-V10-V8-V6-(V11)

This result gives us 77 Hamming transitions.

Simulation results

Next we are going to run our test vectors in 4 possible orderings:

1. Original ordering
2. V3-V9-V2-V1-V4-V5-V7-V8-V6-V10 (79)
3. V3-V9-V2-V4-V1-V19-V6-V8-V7-V5 (78)
4. V3-V9-V2-V5-V7-V4-V1-V10-V8-V6 (77)

Results from Hspice









|  |  |  |
| --- | --- | --- |
| Vector | Total test power consumption (uW) | Test power reduction |
| Original 128 transitions | 37.040 | / |
| 79 transitions | 23.992 | 35.4% |
| 78 transitions | 24.446 | 34.0% |
| 77 transitions | 25.190 | 31.9% |

Figure 5.

The technology is 45nm

Supply voltage is 1.0V

Vector period is 100ns

Conclusion

As we can see from the results of Hspice simulation process, our original assumption of reducing the number of input transitions will result in a reduction of logic transitions to be true. However, as we further reducing the transitions with all types of method, the expected lower test power consumption did not appear in fewer transitions situation. This result shows that least amount of input transitions will reduce the total amount of node transitions, but least total test power consumption might be obtained by other approaches. In this case we reached least amount of total test power consumption in 79 transitions situation.

References

1. Power tools slides by Murali Dharan
2. http://ptm.asu.edu/
3. <http://web.eecs.umich.edu/~jhayes/iscas.restore/c6288.html>
4. <http://www.eng.auburn.edu/~vagrawal/COURSE/E6270_Spr11/course.html>
5. Lin S, Kernighan B W. An effective heuristic algorithm for the traveling-salesman problem[J]. Operations research, 1973, 21(2): 498-516.
6. Miller C E, Tucker A W, Zemlin R A. Integer programming formulation of traveling salesman problems[J]. Journal of the ACM (JACM), 1960, 7(4): 326-329.
7. Zhang Y. Solving large-scale linear programs by interior-point methods under the Matlab∗ Environment†[J]. Optimization Methods and Software, 1998, 10(1): 1-31.
8. K. R. Kantipudi and V. D. Agrawal (2007), “A Reduced Complexity Algorithm for Minimizing N-Detect Tests," *Proc. 20th International Conf. VLSI Design*, Jan. , pp. 492-497.

*Special Thanks*

To my roommate Chi Xu for helping me to accomplish the node injection in Matlab and my classmate Huiting Zhang for the assistance with Hspice simulation process.