Power Verification of RTL Designs

Abhishek Jain, Mugdha Thorat
Adviser: Prof. Morris Jones
Department of Electrical Engineering, San Jose State University, San Jose, California 95192.

Introduction

Power consumption in digital circuits has always been an important aspect of product development. With the advent of new portable and wearable technologies, power estimation and measurement is becoming a prime factor. Estimating power accurately at early stages of design cycle is extremely critical to achieve the most optimization.

This project aimed at measuring and verifying the power values of a design at RTL level. The Device Under Test (DUT) is "ARM Cortex-M0 IP". With the use of ARM compiler toolchain, various standard test cases in the form of C were generated.. These test cases were then used to perform Verilog simulations to obtain switching activity. RTL simulation, synthesis, gate level simulations & PERL scripting were some of the tasks being used to calculate the power values.

Methodology

Dynamic power estimation is usually done taking into consideration the average power dissipated by the circuit. In a RTL design power can be computed using the below formula:

$$P_{dyn} = \sum_i 0.5 * C_{Li} * V_{DD}^2 * r_i * f_{clk} \qquad \text{(equation 1)}$$

r_i= number of times node i changes states per clock cycle

C_{Li} = CMOS load capacitance of gate i.

Value Change Dump file (VCD file) :

- Generated after simulation of RTL or gate level Netlist
- It contains the switching activity information of all the signals in the design, so it can be used to translate the switching into dynamic power values
- It is divided into three sections shown below:

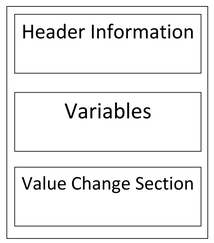


Fig. 1 VCD File structure

Header: Timestamp, simulator version, timescale
Variables : Scope Information, List of signals instantiated within the scope.
Value Change : Time ordered value changes of signals.

Methodology

The variable section gives information about the signals in the design and value change section stores their switching activity.

eg., variable declaration: $var wire 1 * en_q $end

```
Value change:   #0          // timestamp
                1^          //value change
                b1010 &     // a vector signal
                #4
                0^
```

Flow of the Project:

1. Compile test C Code and generate binary file.
2. Perform Verilog simulation using the binary file to generate a VCD file.
3. Parse the VCD file and extract switching activity.
4. Calculate the dynamic power using equation 1.

The top script run.pl runs the complete flow of the project. Parsing of VCD file is done by another script switching.pl. Its flowchart is depicted below. The third script power.pl performs power calculations.

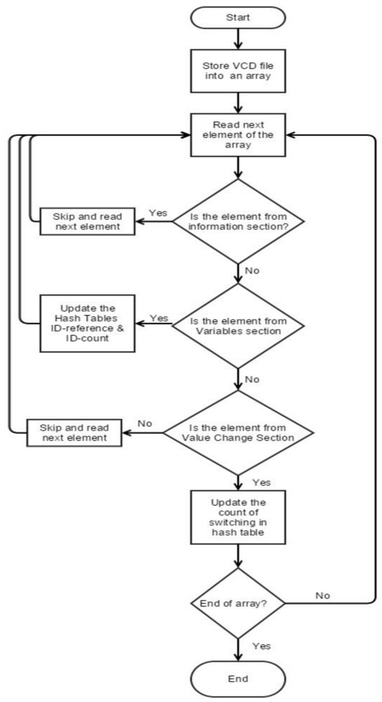


Fig. 2 Flowchart for switching.pl script

Results

The minimum dynamic power is first calculated for a simple while(1) loop. Minimum power is assumed to be the power dissipated by the core when it is in active mode. The interfaces however are not used. Power value is validated against datasheets for Cortex M0 based MCUs by vendors like STMicroelectronics, NXP, Freescale and STM32 series hardware kit.

- Console output of the script

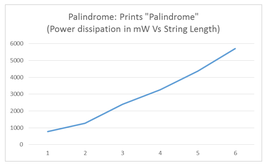
```
******** 4. calculate power
starting power
Multiplication factor is: 4.468608e-07

***********
power = 66.857078592 mW

***********
done
******** done
```

The average power values calculated from the datasheets were around 50mW for minimum configuration.

Test C codes used

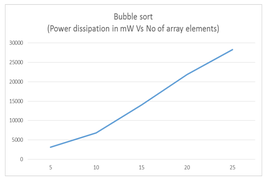
- Series of integers
- Palindrome
- Bubble Sort
- Fibonacci Series

- Power values for Palindrome



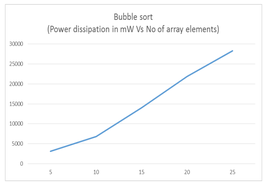
Palindrome: Prints "Palindrome"	
String length	Power dissipation in mW
1	776.06
2	1279.65
3	2406.81
4	3273.68
5	4369.69
6	5699.72

- Power values for Bubble Sort : Print entire array



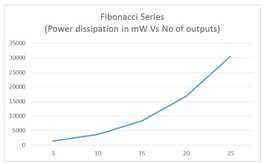
Bubble sort	
array elements	Power dissipation in mW
5	3098.01
10	6792.34
15	14006.85
20	21861.11
25	28305.90

- Power values for Bubble Sort : Print Last element



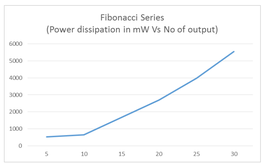
Bubble sort	
No of array elements	Power dissipation in mW
5	1194.46
10	2591.02
15	6605.62
20	12036.87
25	15710.35

- Power values for Fibonacci Series: Print entire series



Fibonacci Series	
No of outputs	Power dissipation in mW
5	1422.49
10	3668.38
15	8384.14
20	16821.68
25	30565.49

- Power values for Fibonacci Series: Print last element



Fibonacci Series	
No of outputs	Power dissipation in mW
5	529.53
10	645.16
15	1674.68
20	2698.98
25	3975.68
30	5563.36

Summary

It can be seen from the graphs above that the power consumption increases exponentially as the complexity of the code increases. The power dissipated is more when the peripherals related to printing the data to the console come into picture. It can also be established that the power dissipated only for computation, without the use of interfaces, also grows exponentially with the complexity of the code. This is in accordance with the logic that as the complexity of the code increases, the VCD file generated is bigger thus resulting into higher power in terms of the switching activity.

Key References

1. Farid N. Najm "A Survey of Power Estimation Techniques in VLSI Circuits" Coordinated Science Laboratory University of Illinois at Urbana-Champaign, Urbana, IL 61801.
2. A Shen, A. Ghosh, S. Devdas and K Kuetzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks", IEEE/ACM International Conference on Computer Aided Design, pp 402-407, Santa Clara,CA, Nov 8-12 1992
3. S Devdas , K Kuetzer, and J White "Estimation of power dissipation in CMOS combinational circuits using Boolean function manipulation" IEEE transactions on Computer-Aided Design vol 11, no 3 pp 373-383, March 1992.

Acknowledgements

We would like to thank our project adviser, Prof. Morris Jones for his motivational advising and constant supervision on our progress. We would also like to thank our project co-adviser and graduate adviser Dr. Thuy Le for giving us an opportunity to work on our project and facilitating our project work.