Design and Implementation of Power Estimation Technique for Digital Circuits

Aarthy M.
Assistant Professor
School of Electronics (SENSE)
VIT University
Vellore Tamilnadu, India

Parekh Mihir H.
Student
School of Electronics (SENSE)
VIT University
Vellore Tamilnadu, India

Nirav Trivedi
Student
Information & Comm. Tech.
DAIICT college
Gandhinagar Gujarat, India

***Abstract*— Nowadays modelling and estimation of power dissipation in the early stages of VLSI design flow has become more important since the intensive scaling of transistors results in higher leakage currents. Various highly advanced Electronic Design Automation (EDA) tools have emerged in recent years to address the issue of power consumption. With a vast number of these power measurement tools emerging, analyzing power consumed by digital circuits has not only become easier but also more effective methods are deployed to optimize digital circuits to dissipate less power. This paper describes the technique of estimating the dynamic as well as the static power consumed by the digital circuit. This technique uses various design constraint files as its input such as gate-level netlist, library file, VCD file and produces the power report. This method uses "gpdk 180nm" library file for estimating the power. This process of power estimation is implemented using combination of C++ and PERL coding languages. To calculate the power RTL coding is done in Verilog using CADENCE NC-Launch and CADENCE RTL compiler is used to generate the "Gate-Level Netlist". For various standard digital benchmark circuits the power report produced by this technique is tested and compared with the power report generated by CADENCE RC-Compiler.**

Keywords*— *Netlist; Value Change Dump (VCD); Dynamic Power; Static Power; EDA

I. INTRODUCTION

Digital circuits in today's modern advance designs are characterized and parameterized by various complex issues. Among these issues, although timing remains critical but power has become more important in achieving design success. Nowadays estimation and management of power has become a challenge and a key factor for various chip designers since it affects battery life, cooling requirements, packaging decisions, design performance and chip reliability [1]. Hence accurate power analysis is very much required for validating the designs. Today designers must consider the impact of timing and area while estimating the power dissipated by a digital circuit.

The power consumption of various electronic products has become a major design factor with the increase in speed, mobility and miniaturization of these products [2]. If we take an example of mobile devices then for these devices the power consumption determines their battery life-time. Thus if the various mobile designers want to increase the battery life of the mobile then reduction in the power dissipation of digital circuits is must.

Thus, we can say that with the increasing usage of the hand-held devices by the consumers in day-to-day life, implementing low power design methodologies has become certain. But before implementing the low power design methodologies we require to find or estimate the power consumed by the circuit. For a particular application one of the important requirements is to know that how much power a circuit should dissipate. So after designer completes his coding part by keeping in mind all the specifications given to him, a power calculation for that design needs to be done in order to confirm that the design meets the required specifications in terms of power. This is done before sending the chip for fabrication. Hence it is very important to build power calculation tools that can measure accurate power values.

Various EDA (Electronic Design Automation) tools such as CADENCE RC Compiler, Synopsis Prime-Time, etc. have been developed today. These tools are targeted specifically towards power domain. The usage of these tools is classified depending on the layer of abstraction they are used in [2]. The three main layers of abstraction are RTL (Register Transfer Level), the gate and the transistor level. This paper mainly focuses on the gate-level abstraction for calculating the power consumed by various digital circuits.

In this paper Section-II gives the background, the proposed technique is described in Section-III, experimental results are given in Section-IV, Section-V concludes the paper and Section-VI shows the acknowledgment with references mentioned in Section-VII.

II. BACKGROUND

A. Components of Power Dissipation

In any digital CMOS (Complementary Metal Oxide Semi-conductor) circuits, the power is consumed by the circuit whenever the load capacitors in the circuit are either charged from 0 to Vdd or discharged from Vdd to 0 voltage values. This is known as the switching of the capacitors. Large VLSI circuits contain different modules such as RAMs, FIFO, Functional Units, etc. So in order to reduce the power consumed by a digital circuit, any of these modules can be turned off when not in used. But before understanding the power reduction methodologies, it is important to know the various factors contributing towards the power dissipation. The power dissipation of digital CMOS circuits can be described by

$$Pavg = P\ dynamic + P\ static$$

where 'Pavg' is the average power dissipation, 'P dynamic' is the dynamic power dissipation due to switching of transistors, and 'P static' is the static power dissipation.

1) Static Power

Static Power is defined as the power dissipated by a gate or a standard cell (in ASIC terms) when it is not active or turned off. Theoretically speaking, there is no static power in CMOS circuits since in the steady state no current flows directly from Vdd to ground. But practically this is not feasible since MOS transistor is not a perfect switch. There will always be leakage currents, sub threshold currents, and substrate injection currents, which give rise to the static component of power dissipation. [2] [4].

Thus in terms of ASIC design static power is nothing but the leakage power dissipated in a standard logic cell. ASIC or Library vendors provides the value of the leakage power dissipated by a particular standard cell. For example in gpdk 180nm library file there are total 470 standard cells which comprises of combinational and sequential cells. The leakage power values for all these cells are defined in this library file.

2) Dynamic Power

Dynamic Power can be defined as the power dissipated by the circuit when it is in on state or active. A circuit is said to be active whenever there is a voltage change on the nets of the circuit due to some stimulus applied. In other words, dynamic power is caused by the charging and discharging of the parasitic capacitors associated with output wire of the gate.

Dynamic power of a circuit is composed of

a) Switching power
b) Internal power

a) Switching Power

The switching power of a driving cell is the power dissipated by the charging and discharging of the load capacitance at the output of the cell [2]. The total load capacitance at the output of a driving cell is the sum of the net and gate capacitances on the driving output. The charging and discharging are result of logic transitions. Switching power increases as logic transitions increase. Therefore, the switching power of a cell is a function of both the total load capacitance at the cell output and the rate of logic transitions. [3] Switching power comprises 70-90 percent of the power dissipation of an active CMOS circuit.

The switching power is determined by the capacitive load and the frequency of the logic transitions on a cell output [4].

$$Switching\ Power = Cload * V^2 * f$$

where the total load capacitance (Cload) is the sum of the net and gate capacitances on the driving output, and the frequency (f) is the rate of state transitions which is given as $f = N(T)/2T$ in which N(T) = the number of logic transitions observed at the output wire of a cell in a period T. This period T is nothing but the simulation time elapsed [4].

b) Internal Power

The internal power is caused by the charging of internal loads as well as by the short-circuit current between N and P transistors of a gate when both transistors are on [1].

$$Internal\ Power = (Cint * V^2 * f) + (V * Isc)$$

ASIC or Library vendors provide power models for the internal power consumption of CMOS cells, which are characterized with different driver output loads and input signal transition times.

III. PROPOSED TECHNIQUE

In order to analyze the power consumption of a design, it is important to consider all the factors which contribute to both the static and dynamic power. So, before understanding the algorithm let us go through the power analysis requirements.

A. Power Analysis Requirements

In order to analyze the power consumption, the following models are required:

1) Gate-Level Netlist

RTL Logic Synthesis tool converts the RTL coding into the Gate-Level Netlist. A Gate-level Netlist is the description of the circuit in terms of gates and connections between them. It represents architecture as asset of registers and combinational logic. Netlist is required to estimate the power in order to identify the connections and to find which

standard cells are used by the synthesis tool to realize a particular design.

2) Cell- Library File

Since we are dealing with the semi-custom design, all the characteristics of the standard cell are defined in a cell-library. For power estimation process cell library is required for extracting the values of pin-capacitances associated with the standard cells used in the design. It is also required for extracting the values of leakage power and estimating the total leakage power consumed by the design.

3) Signal Activity

The dynamic power is directly proportional to the number of state transitions taking place at the output of each standard cell.

The number of transitions taking place at the output is known as the signal activity. This signal activity can be obtained from the .VCD file.

4) Net Capacitances

Net parasitics (or capacitances) affect the dynamic power of the design. Switching power is directly proportional to the net capacitance. This net capacitance is the combination of both the input capacitance and the output capacitance of the standard cells used in the design. Fig. 2 shows the models for power analysis requirements

B. Proposed Algorithm

The following steps shows how the power can be estimated by using the gate-level netlist, gpdk 180nm library file and VCD file. The technique focuses on calculating the leakage and the switching power for a particular circuit design.

Step-1: Write a Verilog code for the desired digital circuit and synthesize it by using gpdk 180nm technology library and generate the gate-level netlist. In this paper this step is done using CADENCE RTL Compiler.

Step-2: From the netlist we want to know what is the hierarchy of the circuit i.e. how many levels of instantiations are present in the circuit and the types of standard cells used with input and output variable declaration. So we will manipulate the netlist with the help of C++ coding in such a way that it will generate the files containing the required information. The two files generated at the output of this step are (1) file_containing_inputs and (2) file_containing_outputs. Fig. 1 represents the flow-chart of this step

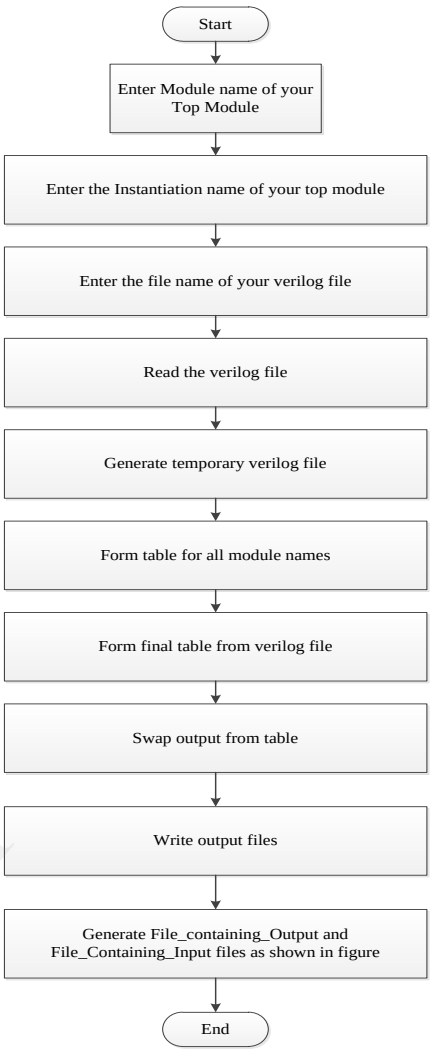

Fig. 1 Flow-chart for Step-2

Step-3: In this step we want to find the switching activity, so generate the VCD file. Thereafter read the VCD file and find the value of N(T) by counting the total number of transitions taking place at the output wire and find the value of 'T' by measuring the simulation time elapsed or by finding the critical path delay. Also from the VCD file for each and every wire used in the design find its hierarchal signal name. These signal names will be used to identify whether the wire is input or output. Fig. 2 represents the flowchart for this step.

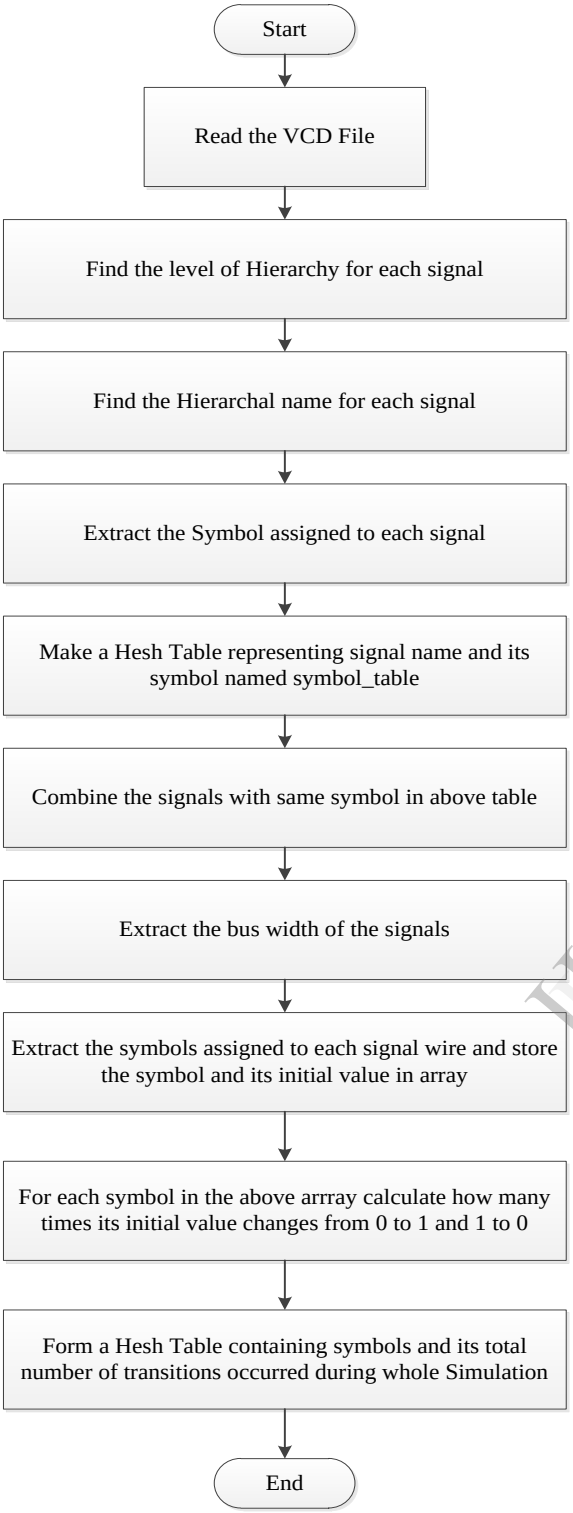

Fig. 2 Flow-Chart for Step-3

Now before executing this step, it is necessary to understand how Cadence tool calculates the number of transitions for a particular circuit because we want to match the power calculated by this technique to that of cadence.

In order to calculate power from Cadence tool only verilog code is required and VCD file is not required. So it calculates the total number of transitions taking place at each output wire by taking all the possible input combinations. For example if a digital circuit is having 'n' inputs then Cadence will count the total number of transitions taking place at the output of a cell by considering all the '2^n ' possible input combinations. Hence we require that the proposed technique also counts the transitions by considering all input combinations. Thus in order to achieve this we have used a 'Linear Feedback Shift Register (LFSR)' which randomly generates all the possible input combinations by implementing its primitive polynomials i.e. for 3-bit input the implementation of primitive polynomial LFSR will generate all the possible states from 1 to 7 randomly. Note that the LFSR will generate all states except 0, because the generation of this state will lock the LFSR permanently. Always the initial value of LFSR must be given as all ones, by doing so it will generate all the states except 0 at each clock cycle randomly.

Now this LFSR must be used in the test-bench and then the VCD file must be generated. After generating this VCD file the flowchart for step-3 must be executed in order to get the same switching activity as that of Cadence.

Step-4: Read the library file and prepare a lookup table containing the name all the standard cells. Now from the previous step we know the input and output signal names. So, now read the outputs obtained from step-1 and form the signal name along with the hierarchy. Then for each cell used in the design find which signals are going as an input and which signals are coming as the output for a particular cell and from that extract the pin name. From the cell name and its particular pin extract the value of input and output capacitance for each input-output wire. Then calculate the value of load capacitance for each wire i.e.

Cload = o/p cap. of driver cell + i/p cap. of driving cells

In the pre-layout phase, the wiring Capacitance is also included in C_i. Since wiring capacitance depends on the placement and routing of the gate-level netlist, accurate estimation cannot be obtained before the physical design phase. However, there is still a need to perform capacitance estimation before physical design because the latter are lengthy processes [4]. One way to solve this problem is to predict the wire length of a net from the number of pins incident to the net. This is called the "Wire-Load Model" in ASIC terms [4]. A wire load model provides a mapping of net's pin-count to the wiring capacitance without actually knowing the exact length of a net. Thus, the pre-layout wire-load model coupled with pin-capacitance characterization can provide a good capacitance estimate for gate-level power analysis [4].

Now for every library file a wire-load model is defined which gives the approximate value of wiring capacitance. So

after calculating the load capacitance we need to add this wiring capacitance also. Thus extract the value of wiring capacitance from the library file and find out total how many output wires are present in the circuit design. Hence now the total load capacitance can be given and calculated as

$$C_{total} = Cload + (C_{wire} * \text{total no. of output wires})$$

Fig. 3 and 4 shows the flowchart for this step.

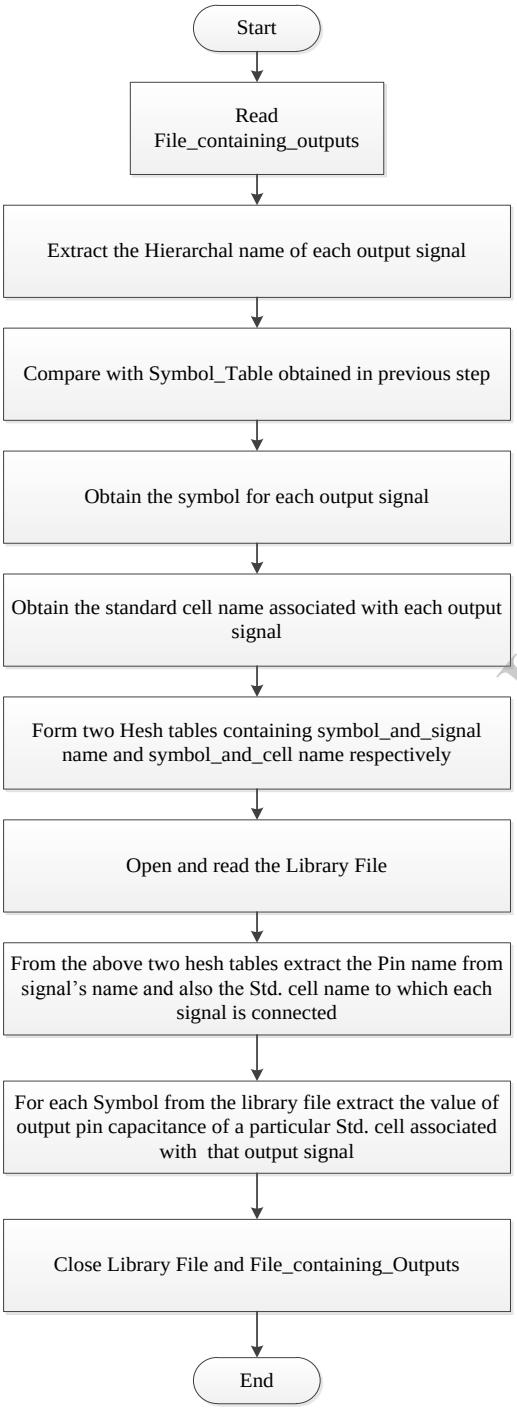

Fig. 3 Flow-Chart1 for Step-4

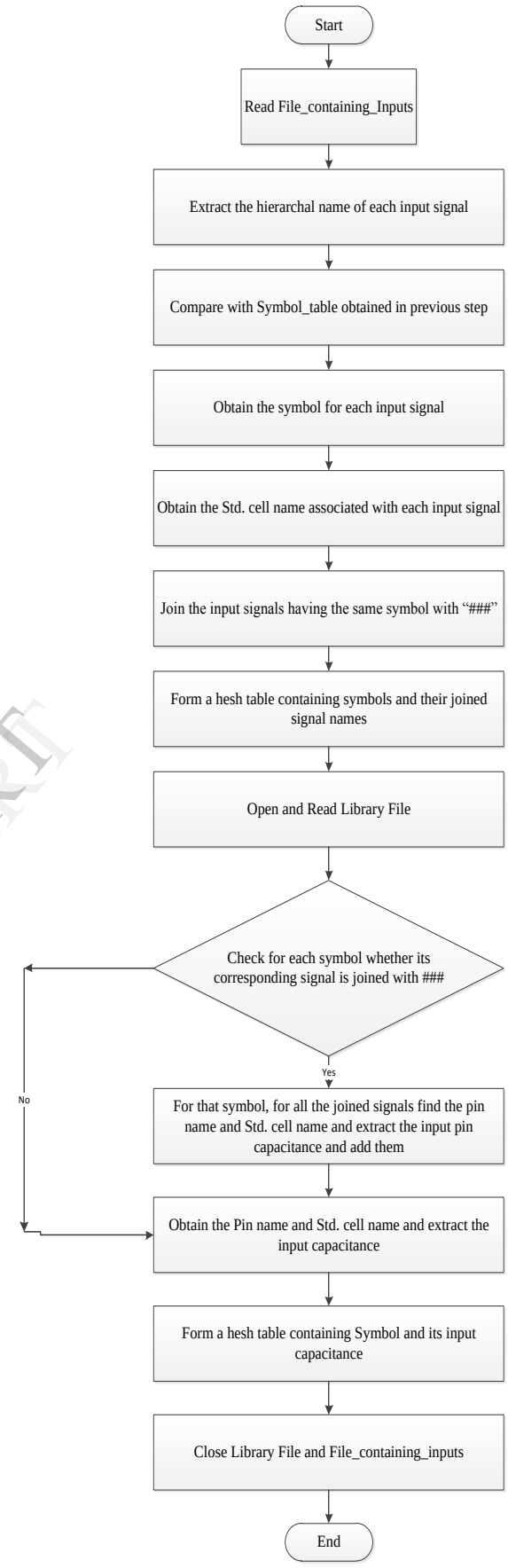

Fig. 4 Flow-Chart2 for Step-4

Step-5: The value of operating frequency can also be obtained by finding the critical path delay for the circuit and taking its inverse. This critical path delay could be found with the help of "CADENCE rc gui". Since we are using gpdk 180nm technology library file so V = 1.8v.

Step-6: Extract the values of internal power from the .lib file for the cells used in the design and the value of T i.e. simulation time elapsed can be obtained from VCD file. Now dynamic power can be calculated using following equation.

$$D.P. = 0.5 * \sum(N(T)*Cload) * V^2 * (1/T)$$

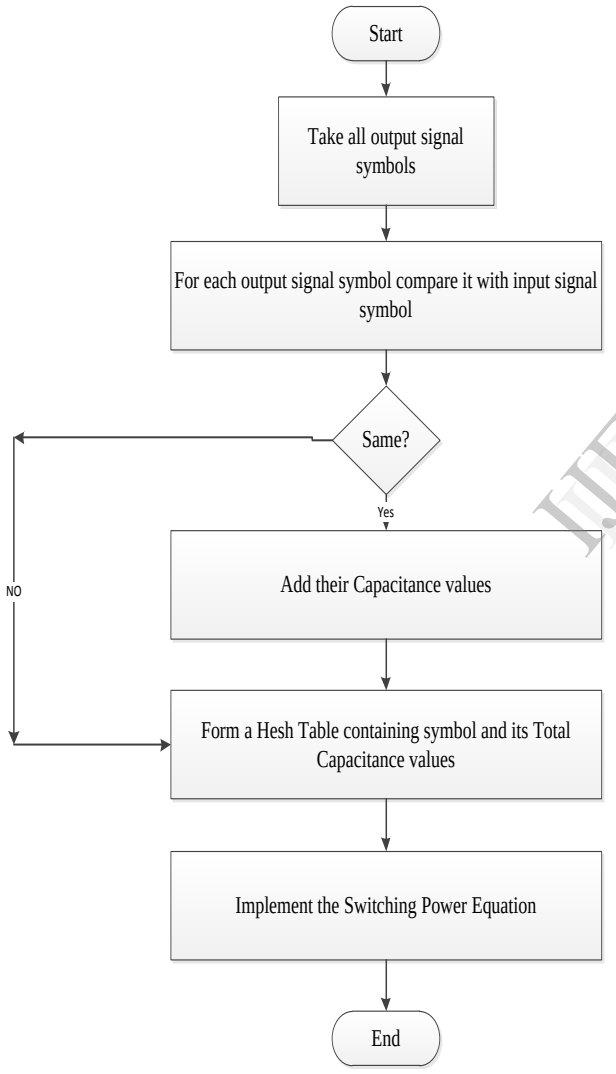

Fig. 5 Flow-Chart for Step-6

Step-7: Calculate Leakage power. In library file leakge power is given for each standard cell. So just calculate and find the total no. of standard cells used in the circuit and compare it with the library file given to obtain the leakage power for the cells used and add them all to get the 'total leakage power'. Thus,

$$\text{Static power} = \sum \text{leakage power}$$

Fig. 6 shows the flowchart for this step.

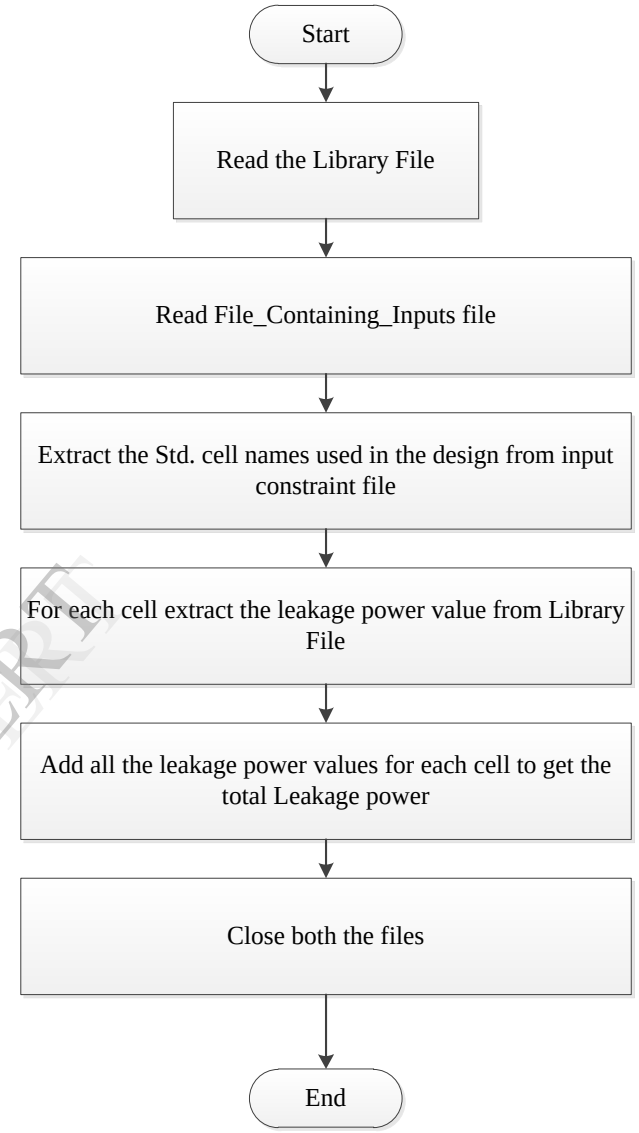

Fig. 6 Flow-Chart for Step-7

Among all the steps, step-2 and steps-3, 4, 5, 6, 7 are implemented using C++ and PERL language respectively.

IV. EXPERIMENTAL RESULTS

Here first of all we will see the outputs of each and every step obtained for a simple circuit i.e. full-adder. Fig. 7 shows the netlist obtained from the Cadence and Fig. 8 and 9 shows the output obtained at the end of step-1 i.e. file_containing_inputs and file_containing_outputs respectively.

From the Fig. 8 and 9 it is clear that how the netlist should be manipulated with the help of C++ code in order to find the level of hierarchy and the names of standard cells used in the design along with their instantiated names in the netlist file.

Fig. 10 shows the hierarchal signal name of each and every wire used in the design, its corresponding symbol defined in VCD file and the total number of transitions taking place at each wire due to various input combinations applied by LFSR.

```
// Generated by Cadence Encounter(R) RTL Compiler v09.10-s233_1

module halfadder(a, b, s, c);
  input a, b;
  output s, c;
  wire a, b;
  wire s, c;
  ADDHXL g15(.A (b), .B (a), .S (s), .CO (c));
endmodule

module halfadder_1(a, b, s, c);
  input a, b;
  output s, c;
  wire a, b;
  wire s, c;
  ADDHXL g15(.A (a), .B (b), .S (s), .CO (c));
endmodule

module fulladder(a, b, c, x, y);
  input a, b, c;
  output x, y;
  wire a, b, c;
  wire x, y;
  wire t1, t2, t3;
  halfadder ha1(a, b, t1, t2);
  halfadder_1 ha2(c, t1, x, t3);
  OR2XL g1(.A (t3), .B (t2), .Y (y));
endmodule
```

Fig. 7 Netlist obtained from Cadence tool for Full-Adder circuit

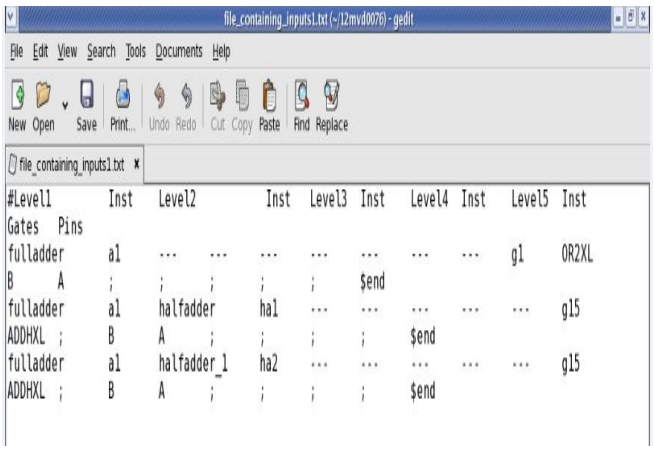

Fig. 8 File_containing_inputs

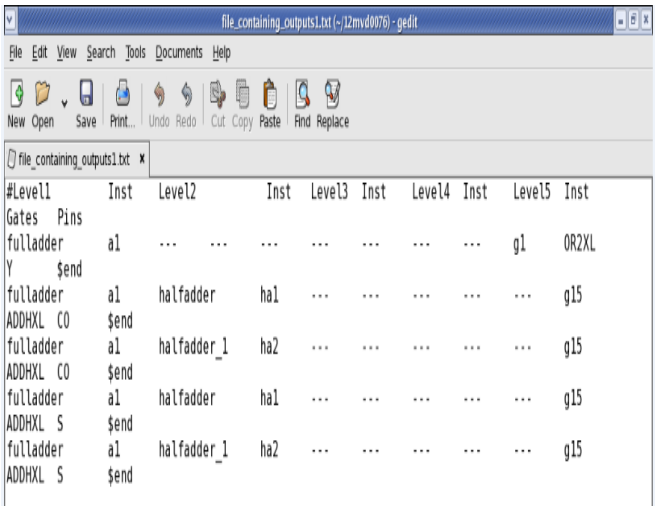

Fig. 9 File_containing_outputs

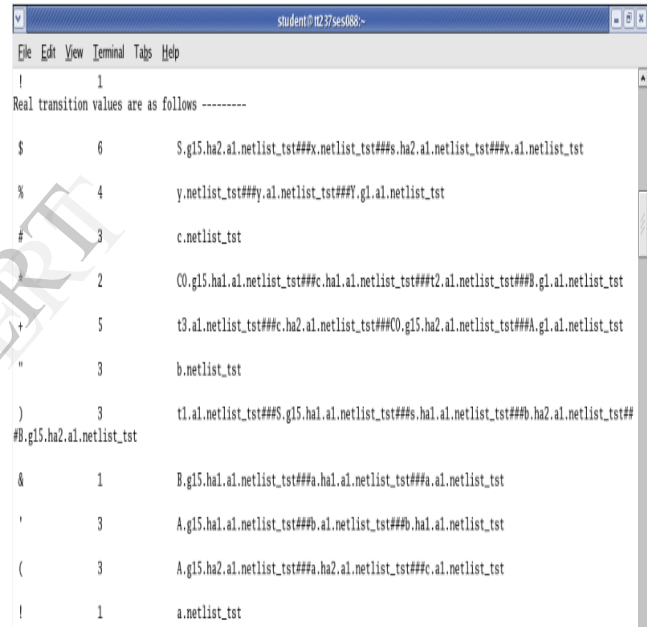

Fig. 10 Transition Values and Hierarchal signal names obtained from VCD file

Fig. 11 shows the Symbol, Std. Cell name and the hierarchal signal name obtained from file_conataining_outputs file. It shows that which output wire (represented by symbol) is connected to which cell and what is its hierarchal name. This information is used to extract the output wire capacitance values from the library file. Output wire capacitance values extracted is shown in Fig. 12.

Fig. 11 Symbol, Cell name and output signal name obtained from file_containing_outputs

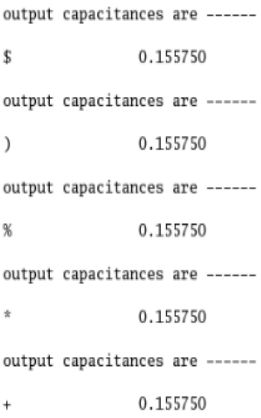

Fig. 12 Output wire Capacitances

Fig. 13 shows the input signal's hierarchal name and the symbols associated with them which are obtained file_containing_inputs. The table in the figure named "after joining signals" indicated that the input signals having the same symbol are joined together with "###". If the signals are joined with "###" then it shows that a particular cell is having more than one fan-outs. If more than one fan-out is their then the input pin capacitance of that joined signals are extracted from the library file and are added together. Afterwards this added input capacitance value will be added with the output capacitance of the same wire (representing the same symbol). Fig. 14 shows the values of input pin capacitances obtained and the values of total wire capacitance for all output wires.

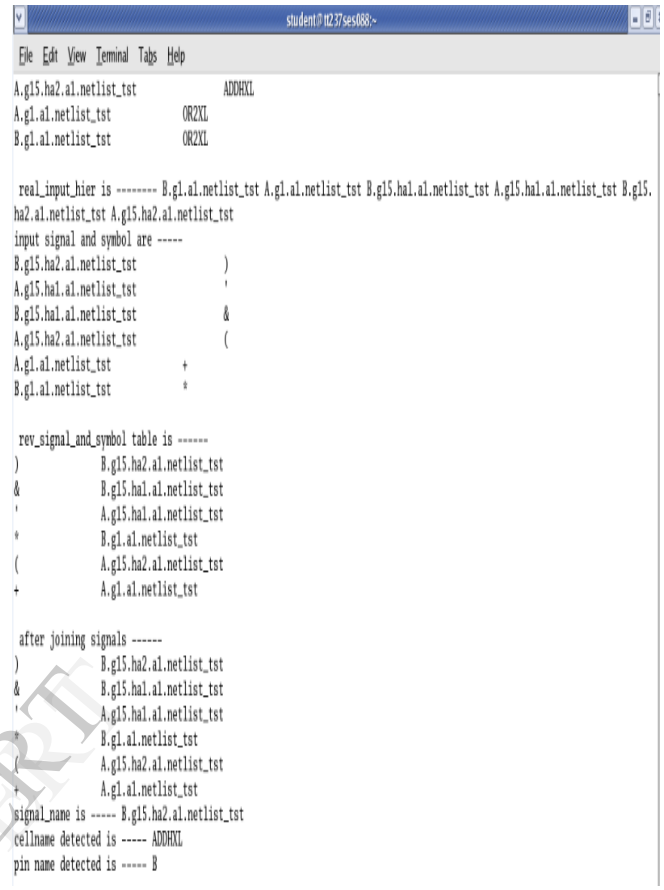

Fig. 13 Symbol and hierarchal signal input signal names obtained from file_containing-inputs file

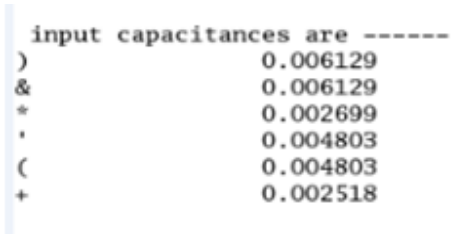

Fig. 14 Input wire capacitance values and total wire capacitance values for every output signal of the circuit

Fig. 15 shows the "Final Power Report" obtained for full-adder circuit. The report shows the names of the constraint files used in the process, library file used, units of various quantities and the value of switching and leakage power obtained with the proposed technique.

```
                                POWER REPORT

Created By: Mihir H. Parekh (12MVD0076)
Date : Fri Apr  4 11:49:53 2014
*****************************************************************
******************

1) Simulation Activity File :   fulladder_netlist.vcd
2) Input Constraint File : file_containing_inputs1.txt
3) Output Constraint File :     file_containing_outputs1.txt
4) Library File :    fast.lib
5) Wire Load Model Mode : enclosed


Library used :  TSMC18 Fast.Lib (for 180nm_technology)



Global Operating Voltage = 1.98
Power-specific unit information :
      Voltage Units = 1V
      Capacitance Units = 1.000000pf
      Time Units = 1ps
      Dynamic Power Units = 1uW
      Static Power Units = 1nW



Total Switching Power    =  7.2543924486 uW

Total Leakage power      =  3.384184716 nW
```

Fig. 15 Final Power Report obtained for Full-Adder Circuit

Similarly this proposed technique is used to find the power for various ISCAS 85 benchmark circuits and the power obtained by this technique is validated and tested against Cadence Tool for the same benchmark circuits.

Table I shows the values of switching and leakage power obtained for various benchmark circuits by proposed technique and by Cadence tool. It can be easily seen that the leakage power is exactly the same but there is a variation of ± 15% in the values of switching power. This variation is due to the fact that the proposed technique is based on Gate-Level Power analysis and the Cadence tool calculates the power at higher abstraction level. So with this much of variation the proposed technique is validated.

TABLE I
Power Values Obtained for Different Circuits by Proposed Technique and by Cadence Tool

Circuit	Proposed Technique		Cadence Tool	
	Switching Power (uW)	***Leakage Power (nW)***	***Switching Power (uW)***	***Leakage Power (nW)***
Full-adder	7.254	0.032	8.344	0.033
c17	7.656	0.872	7.404	0.870
c432	283.718	24.8096	323.649	24.81
74181	136.88	15.761	139.32	15.760
74182	16.97	2.4056	25.393	2.41
74283	65.5875	11.236	67.428	11.24
74L85	84.39	8.02	73.675	8.03

V. CONCLUSION AND FUTURE WORK

This paper presented a novel technique for estimating the power consumed or dissipated by the Digital Circuits by using various files such as Netlist file, .VCD file and Library File as an input and generating the final power report for various circuits as output. The proposed technique has been tested for various standard ISCAS 85 benchmark circuits and the power values obtained are validated by finding the power calculated by Cadence Tool for the same circuits. The output results and the power report obtained by this technique solely use the gpdk 180 nm technology library file. Thus, in future the work can be extended for various different library files provided by different vendors and the technique can be made more generalized. Hence overall this technique could be useful for various companies making the EDA tools for estimating the power dissipated by Digital Circuits.

VI. ACKNOWLEDGMENT

It is with great pleasure and pride that we are presenting this paper before you. At this moment of triumph, it would be unfair to neglect all those who helped us in the successful completion of this project work.

We express our deep sense of gratitude to Asst. professor M.S.RAVI from the department of SENSE (School of ElectroNicS Engineering), VIT University for giving us support at every stage of this project work. We are indebted to his esteemed guidance, constant encouragement and fruitful suggestions from the beginning to the end of this

project. His trust and support inspired us in the most important moments of making right decisions.

Also we are thankful to Dr. Harish Kittur, Program Manager VLSI Division, VIT University for providing a solid background for our project and research thereafter.

VII. REFERENCES

[1] Expanding the Synopses Prime-Time solution with Power Analyses, Gordon Yip, Product Marketing Manager, Synopses Inc, June 2006.
[2] A verilog based Simulation Methodology for estimating Power and Area, A thesis submitted for the partial fulfillment of the award for degree of masters in VLSI by Ramesh Guntupalli, NIT Rourkela, India.
[3] D.Soudris, C.Piguet, C.Goutis, "Designing CMOS circuits for Low Power", Kluwer Academnic Publishers, 2002.
[4] A book on "Practical Low Power Digital VLSI Design" by Gary Yeap, chapter-2, pp. 27-41.
[5] Ramesh Guntupalli and K.K.Mahapatra, An accurate Estimation of power using Verilog, International Conference on Electronics, Information and Communication Systems Engg. ICEICE-2011.
[6] C.P.Ravikumar, Mukul R. Prasad and Lavneet S. Hora, Estimation of Power from Module Level Netlist, Department of Electrical Enginnering, IIT Delhi India, IEEE trans, vol. 43., June 2012.
[7] Ronnie L. Wright and Michael A. Shanblatt, Improved Power Estimation for Behavioral and Gate-level Design, IEEE trans, 2001.
[8] Savithri S., Venkatesan R. and Bhaskar S., An assertion Based technique for Transistor level Dynamic Power Estimation, Motorola India Electronics Ltd., IEEE.
[9] A PDF on Gate Level Simulation Methodology, Cadence, India.
[10] Power Estimation using Synopses Prime-Time, ECE5950 Tutorial 5, version e2088b6, by Derek Lockhart.
[11] Sharif Digital Flow Introduction part-1: synthesis and Power analysis, by Nemat Allah Ahmadyan, Dependable System Lab, CE department, Sharif university of Tech. 2009, A PPT from VLSI shareflow website.
[12] Nourivand.A, ChunyanWang, Ahmad.M.O "A VHDL- based technique for an accurate estimation of leakage power in digital CMOS circuits", in the 3rd international IEEE-NEWCAS conf., 2005.pp.47-50.
[13] J. P. Halter and F. Najm, "A gate-level leakage power reduction method for ultra low-power CMOS circuits," in Proc. IEEE Custom Integrated Circuits Conf., 1997, pp. 475–478.
[14] Sagahyroom.A, Placer.J, Burmood.M, Massoumi.M "A VHDL-based simulation methodology for estimating switching activity in static CMOS circuits," in proc. IEEE SAIC conf. 1988, pp. 295-300.
[15] J.Placer, A.Sagahyroon and M. Massoumi, "A Framework for Estimating Maximum Power Dissipation in CMOS Combinational Circuits Using Genetic Algorithms," IEEE Southeastern Symposium on System Theory, pp. 348-352, 1997.
[16] J.Y.Lin et al., "A Cell-Based Power Estimation in CMOS Combinational Circuits," Proceedings of the International Conference on Computer- Aided-Design, pp. 304-309, 1994.
[17] Power Compiler user guide, Version Y-2006.06, June 2006.
[18] A Technical report on "Designing transformed Linear Feedback Shift Register with Minimum Hardware Cost" by Laung-Terng Wang, Nur A. Touba, Richard P. Brent, Hui Xu, and Hui Wang, UT-CERC-12-03 November 8, 2011.
[19] Website: http://www.vlsiip.com/power/