

Power Estimation Techniques – what to expect, what not to expect

Prakash Parikh,
Aquantia Inc.,
700 Tasman Drive,
Milpitas, CA
pparikh@aquantia.com

Abstract:

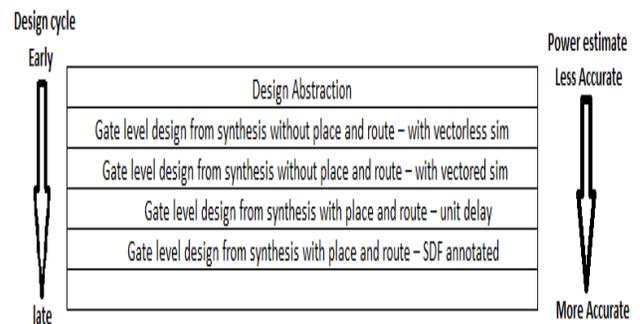
Accurate and efficient power estimation is a key element for low power design. Various power estimation techniques are available using various Electronic Design Automation (EDA) vendor tools. Using EDA tools from different vendors and using various flows for power estimation that include different abstraction levels such as Behavioral, RTL and Gate level, a comparative study can be done to analyze various tradeoffs. It is important to apply different power estimation techniques on the same RTL or Gate level netlist and perform a comparative analysis of the generated results. Analyzing these results can provide useful data for design decisions such as selection of optimum architecture, hardware versus software implementation, voltage selection, process node selection and power gating versus clock gating for particular sub block or the entire design. This paper presents the results for power estimation using various techniques and discusses the architectural decisions made based on those results.

I. INTRODUCTION

Power estimation using EDA tools is not a new subject and there are various power estimation techniques available using various EDA vendor tools. This paper provides the results from using different methodologies and using different EDA tools and then shows the comparison between different flows and the various tradeoffs involved.

Power estimates can be made available during different phases of the chip tapeout cycle using different abstraction levels. The estimates become more accurate as the abstraction levels get closer to the final tapeout netlist for power simulation. The figure below (Figure 1) shows the tradeoffs in power simulations at different abstraction levels. The accuracy of the power simulation results improve as the abstraction levels become more detailed. The more accurate results are available much later in the design cycle. The vectorless simulation method provides early power estimates when infrastructure is not yet developed for providing stimulus to the design. An activity file is not available for such scenario. In absence of activity file from actual simulation, vectorless approach assumes toggle activity of each input node and after that it estimates power for Gate level netlist.

Figure 1: Power estimates and accuracy tradeoffs



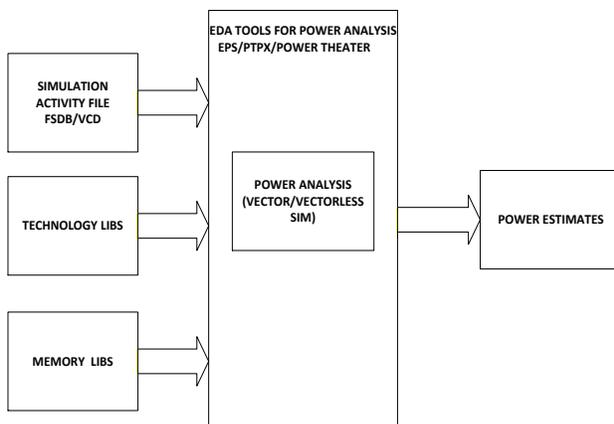
To get the most accurate power estimate one needs to wait till the final netlist. However, that might be too late to get early feedback on design decisions and make changes based on the power estimate data. There is a tradeoff between the accuracy of the power estimate data and how early the power estimate data can be made available. Following are the three different phases of the ASIC tapeout cycle where power estimates were tried and results generated:

- 1) Power estimates using RTL synthesized netlist (without place route).
- 2) Power estimates using place and route netlist with vectorless simulation.
- 3) Power estimates using place and route netlist using vectored simulation with unit delay libs.
- 4) Power estimates using place and route netlist using vectored simulation with SDF annotations.

II. POWER ESTIMATION TESTBENCH:

Various RTL and corresponding gate level netlist blocks were used as “Design under Test” (DUT) for the power estimation. Vectorless and vectored simulations were performed. Three different EDA tools (PTPX from Synopsys, EPS from Cadence and PowerTheater from Apache) were used for generating results. Different voltage selections were studied for 40nm and 28nm process nodes.

Figure 2: Testbench for Power Simulation



Following are the steps for generating power estimates:

- a) Vectorless simulation
 - 1) Select toggle activity for each node.
 - 2) Select the appropriate technology libs.
 - 3) Compile the netlist and generate the power estimates.
- b) Vectored simulation
 - 1) Run the gate level simulation with SDF and generate the activity file (vcd).
 - 2) Provide the activity file and technology libs to the power estimator tool.
 - 3) Select the activity period of interest.
 - 4) Compile the netlist and generate the power estimates.

Simulation Activity File required for vector simulation in Figure 2 above can be generated using the following flows.

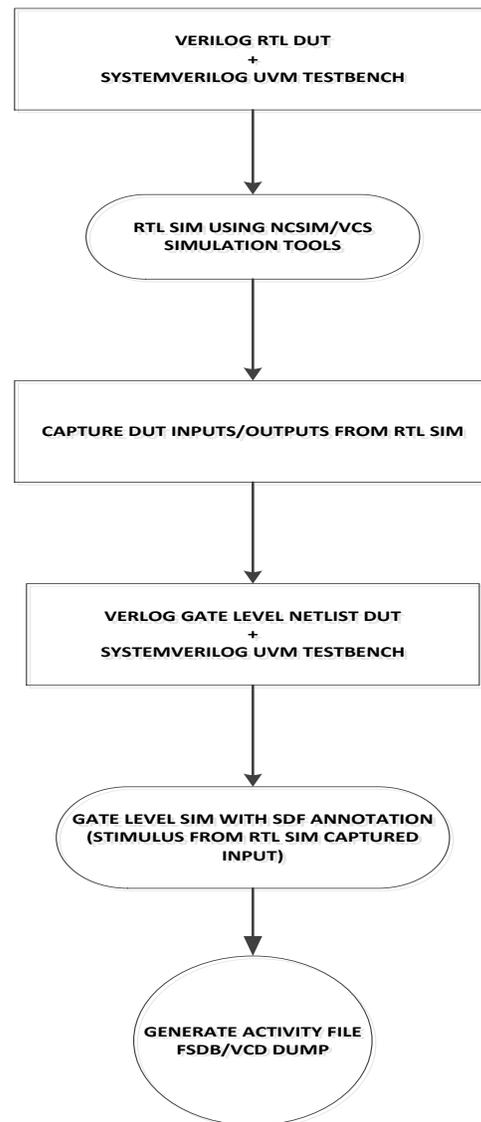
- a) Reusing RTL simulation testbench: In this approach, the testbench for RTL simulation is reused for gate level simulation. First RTL simulation is done using the SystemVerilog/UVM testbench with Verilog RTL DUT. The golden results are captured from RTL simulations and then RTL DUT is replaced with Gate level netlist reusing the same testbench used for the RTL simulations. Once the Gate level simulation results are verified to match the RTL simulations results, the activity file is generated for the appropriate time window.
- b) Capture and playback method: In this approach, the testbench for RTL simulation is used for capturing the inputs and outputs of the DUT. Once the inputs/outputs are captured from the RTL simulations, the same inputs get played on the Gate level netlist. To validate the Gate level simulations results, the outputs of the Gate level simulations are compared against the RTL simulations captured outputs. The activity file is generated for the appropriate

time window once the gate simulations results are confirmed to be valid results.

When the RTL DUT has a standalone testbench, method (a) (above) for generating the activity file saves time. For the scenario where RTL DUT is part of the top level testbench, method (b) (above) provides a faster way of simulating and makes debugging efforts easier, since all other top level modules are not needed when using the playback and capture method. So depending upon the existing RTL simulations testbenches available, one can decide to use the method (a) or method (b) (above) to generate an activity file for power simulations.

Figure (3) below provides the activity file generation flow for power simulation. The first step we followed in the power simulation using Gate level netlist was to construct RTL DUT testbench using SystemVerilog. We captured the data inputs and data outputs of the DUT from the RTL simulation in text files. We constructed a new testbench with Gate level netlist as DUT. The inputs to the Gate level netlist were provided from the text file generated in the step above. We compared the output of the Gate level netlist simulation with the output captured in the text file from the RTL DUT simulation. If the Gate level netlist output did not match the RTL simulation output, both simulation results were debugged and all the issues with design were fixed. Once we matched the RTL simulation output with the Gate level netlist, simulation waveform dump in the waveform viewer was analyzed for the appropriate activity window selection. The activity window was selected representing the peak power activity or typical power activity based on the requirements. Once the activity window was chosen, we performed the final Gate level simulation to generate the activity file (.vcd).

Figure 3: Activity file generation flow



- 1) Vectorless data with fixed toggle activity for core voltage – 0.8V
- 2) Vectorless data with fixed toggle activity for core voltage – 0.85V
- 3) Vectored data for voltage – 0.8V
- 4) Vectored data for voltage – 0.85V

Different EDA vendors tools were used for all four of the above simulation scenarios.

III. POWER ESTIMATION RESULTS AND DECISIONS MADE

Table 1: Power estimates using different EDA tools

TRUE LEAKAGE CORNER (power in mW)			
DSP Block 40nm	Static	Dynamic	Total
PTPX	152.33	545.4	697.73
EPS	154.7	572.2	726.9
PowerTheater	151.29	535.3	686.75

Table 1, above, shows power numbers generated using three different EDA vendor tools. The static power matches very closely for the three tools where dynamic power is within 5-10 % range. Thus all the tools used provide similar estimates.

Table 2: Technology Node selection using power estimates

28nm Netlists	Power in mW		
True Lkg	Static	Dynamic	Total
0.75V	84.3	250	335
0.85V	76.9	294	371
Diff (%)	-9.62	14.97	9.7

Table 2, above, shows power numbers generated using nominal voltages of 0.75V and 0.85V for true leakage corner using the same netlist. As seen in this table, the overall power improvement by lowering the voltage is less than 10%. This helped in selecting the 0.85V as voltage since the benefits for physical design to close the timings easily outweighs the power saving of less than 10%.

Table 3 : Vectorless and Vector power simulation results comparison

VECTORLESS SIM	TRUE LEAKAGE CORNER		
DSP Block 40nm - 0.88V, 125C	Static	Dynamic	Total
internal reg power	44.25	201.5	246
internal latch power	0	0.02	0.02
memory power	5.3	32.55	37.85
other internal power	152.5	635	785
clock power	7.85	16.25	48.22
Total power	209.9	885.32	1092.97
DSP Block 28 nm - 0.935V, 125C	Static	Dynamic	Total
internal reg power	7.65	199.5	207
internal latch power	0.05	0.95	1
memory power	1.23	28.64	29.88
other internal power	18.6	453	471.5
clock power	1.33	12.75	14.1
Total Power	28.86	694.84	723.48

VECTORED SIM	TRUE LEAKAGE CORNER		
DSP Block 40nm - 0.88V, 125C	Static	Dynamic	Total
internal reg power	33.95	183.5	217.5
memory power	39.4	55	94.5
other internal power	77.5	288.5	366
clock power	0.44	8.3	8.75
Total power	151.29	535.3	686.75
DSP Block 28 nm 0.825V, 125C	Static	Dynamic	Total
internal reg power	4.99	111.5	116.5
memory power	5.91	33	38.91
other internal power	13.9	227	241
clock power	0.12	3.8	3.92
Total Power	24.92	375.3	400.33

Table 3, above, shows the comparison of vectorless and vectored simulation for various scenarios. The vectorless simulation data was available much earlier

in the ASIC cycle compared to vectored simulation data since the vectored simulation had to wait for Verification to develop a reasonable test representing real life scenarios. The vectorless simulation provided a good estimate of static power but the estimate of dynamic power was way off. The vectored simulation provided a more accurate dynamic power estimate compared to the actual power measured in the lab. Also, the power between 28nm and 40nm technology nodes was compared and that comparison provided a good estimate of power saving in 28nm technology.

Table 4: Power gating decisions using power estimates

VECTORLESS SIM	TRUE LEAKAGE CORNER		
	Static	Dynamic	Total
DSP Block 40nm			
Sub block A	23.7	95.32	119.02
Sub block B	18.42	70.4	88.82
Sub block C	89	365.3	454.3

From the results obtained in Table 4, we decided to power gate Sub block C and save leakage power for the EEE (Energy Efficient Ethernet) mode of 10GBase-T. The EEE mode needs to be run only when supporting EEE mode of operation. We decided to power gate Sub block C based on vectorless simulation data since leakage power numbers were important. By applying power gating to Sub block C, we save 454mW of leakage power. We were able to keep the design within power budgets based on these power gating decisions. There was no need to do vector simulation in this particular case since the static power estimate was available from the vectorless simulation and the dynamic power number was not that important for the power gating decision.

Another example of power saving from power simulation is the 1G mode for the 10GBase-T chip. The 10GBase-T chip supports 10G mode but for backward compatibility it also supports 1G and other lower rates. By power gating 1G block, it became easy to save the entire leakage power of the 1G block, during the 10G mode of operation.

One another advantage of power estimation is to predict maximum transient current. From the same power estimation simulation, we could plot the current profile. We could determine the maximum current surge from this current profile. We used the maximum current surge information to calculate the capacitance required for the package design.

Table 5: Clock gating decisions using power estimates

VECTOR SIM	TRUE LEAKAGE CORNER		
	Static	Dynamic	Total
DSP Block 40nm			
FIR (Filter)	22.32	89.4	111.72
FFT	60	180	240
IIR (Filter)	30.4	100.2	130.6

Table 5 shows the results that were obtained for the Radio Frequency Interference (RFI) Cancellation block. The RFI block contains sub blocks such as FIR filter, FFT (Fast Fourier Transform) block and IIR filter. Since the FFT operation is not required to estimate frequency at all times and the FFT block needs to be triggered only when a RFI event is detected, we decided to clock gate the FFT block and potentially save the 180mW dynamic power, based on the power analysis report results. We had to run the simulations with vector simulation since the vectorless simulations would not have provided accurate dynamic power estimates.

Table 6: Architecture selection using power estimates

TRUE Leakage Corner	TRUE LEAKAGE CORNER		
	Static	Dynamic	Total
DSP Block 40nm			
FIR (Filter) - 2's complement format	22.32	89.4	111.72
FIR (Filter) - signed magnitude format	20.4	62.4	82.8

TRUE Leakage Corner			
DSP Block 40nm	Static	Dynamic	Total
FIR (Filter) - data - 14 bits, coef - 10 bits	18.11	42.12	60.23
FIR (Filter) - data - 11 bits, coef - 9 bits	12.08	30.23	42.31

The first part of the Table (6) shows the estimated power for the two different implementations of the FIR filter. In order to take advantage of the Gaussian nature of the data, we implemented a signed magnitude FIR filter. We compared signed magnitude against the 2's complement data format implementation of the FIR filter and signed magnitude implementation provided significant power savings.

As shown in the second part of the Table (6), for another case of architecture selection, we studied FIR implementation with 14 bits of data and 10 bits of coefficients and another FIR implementation with 11 bits of data and 9 bits of coefficients. From the power estimates, we decided to implement 11 bits data and 9 bits coefficients FIR design for the 30% power saving since there were 20 instances of such FIR in the design. In order to compensate for these reduced data and coefficients bits, we increased filter in the data path for another filter. This resulted in less power penalty compared to the power saved with 11 bits wide FIR architecture selection.

The following results were derived from power estimations for different scenarios mentioned in Table 1 to Table 6 above.

- 1) Static power estimation using vectorless simulation provided an accurate leakage power prediction.
- 2) Vectorless power estimation provided relative comparison of total power between different voltages for the design. The absolute dynamic power values that we predicted using the vectorless simulation technique were off by 30 % when we measured the actual power in the lab.

3) Dynamic power estimation technique results using three EDA vendor tools, PTPX, EPS and PowerTheater, were within a 10% range. The static power estimated using the same three EDA vendor tools were within a 2-3% range.

4) When we measured the actual chip power in lab, it correlated very well with the power estimated using dynamic power estimation techniques.

5) The power estimated using a maximum leakage corner (125C) provided the worst case power. The power estimated using a typical corner (25C) provided more optimistic power numbers when compared to actual power measured in the lab. The actual power measured in the lab for 100m link operation of the DSP system matched power simulation results obtained with 115C temperature and with nominal voltage.

6) Overall the power saving at a core voltage of 0.75V is much less compared to 0.85V core voltage for the 28nm node. Considering the tradeoffs between power saving versus challenges for timing closure for physical design, we made a decision to go for the higher voltage of 0.85V thereby sacrificing some of the power savings. (Table 2).

7) Overall it helped us to lower power in the design with various design decisions such as applying appropriate clock gating, power gating various parts of the design and replacing power hungry logic with custom cells implementation.

IV. LIMITATIONS:

For the initial revisions of the chip, when we measured the actual power in the lab it did not compare well against the power estimation results. It took quite a few iterations for us to select the temperature for technology libraries characterization. The room temperature 25C results were too optimistic whereas 125C results were too pessimistic for the power numbers in the real working conditions. After many iterations, we observed that the technology libraries characterized for 115C temperature provided the best estimate real working conditions power for the 10G-BaseT chip.

For the power gating purposes, we calculated power saving to be leakage power for the particular block that is power gated. In the actual implementation, it introduced quite a few gates to implement power gating functionality. We could not estimate the negative impact on static and dynamic power upfront because of the usage of all these nontrivial amount of gates to implement power gating functionality.

[3] Making ASIC power estimates before the design
By Bob Eisenstadt, EDN Network

V. CONCLUSIONS:

1) Power estimation using vectorless power simulation/estimation techniques were efficient in simulation runtime compared to vectored power simulation/estimation techniques.

2) The dynamic power estimated when using vectorless simulation was quite off from the actual power. The dynamic power estimation accuracy was much higher using vectored power simulation/estimation techniques.

3) Power estimated using vectored simulation with SDF provided useful information about power hungry blocks/logic of the design, power consumed by memory, tradeoffs when selecting different core voltages for the design, the effectiveness of clock gating and power gating various logic of the design.

4) Power estimated earlier in the design cycle was less accurate as an absolute number but it provided useful relative comparison numbers when exploring different architectures.

5) Power estimated for the maximum leakage corner provided useful data for the worst case power and was used for the package design.

VI. REFERENCES:

[1] Power estimation using Synopsys PrimeTime Tutorial

[2] Cadence Encounter Power System, Unified power analysis for faster design optimization and signoff
http://www.cadence.com/rl/Resources/datasheets/encounter_power_system_ds.pdf

