Efficient Methods for Display Power Estimation and Visualization

Srikanth Reddy Rolla
Emulation Engineer, Intel Corporation
1900 Prairie City Road, Folsom, CA - 95630
srikanth.reddy.rolla@intel.com

Aakash Modi
Graphics Hardware Engineer, Intel Corporation
1900 Prairie City Road, Folsom, CA - 95630
aakash.k.modi@intel.com

Abstract - As power is becoming more prominent in graphics domain, it is critical to estimate power to meet average and peak power budgets in pre-silicon environment. Typically, power in Intel Display IP is calculated using RTL simulations based on toggling of all logic in the design. This approach is useful for single frame tests, however for large frame and high-resolution tests like UHD and 8K, it is impractical to scale this approach and would take 1-2 weeks slowing down the entire process. This paper discusses methodology to solve the problem using emulation instead to get a speedup of 7-14x in generating waveforms. The speedup enabled the power optimization team to estimate and analyze display power faster, thereby providing architects and designers prompt feedbacks to improve power for the design.

I. INTRODUCTION

To quantify the power consumption of the Display IP, following methodology is used:

1. Identification of workloads based on the performance requirement of the product. Factors that determine performance are the resolution of the frame and the speed at which the frames need to be processed (FPS).
2. Running the identified workloads on the RTL model and generating waveforms (FSDB/SAIFs)
3. Feeding these waveforms and synthesized netlists to the power tool (PrimeTime PX) to generate the power reports for the workloads

Step (2) of the above methodology can be accomplished by using either simulation (using servers) or emulation (using FPGAs). Over the years, power estimation was done using simulation. However, as the resolution of the frames and/or FPS increased (due to higher performance requirements), the time taken to generate waveforms in step (2) increased exponentially through simulation. For instance, time taken to generate waveforms for a single frame workload running at a resolution of 320p versus a workload running at resolution of 8K increased from 6 hours to over 2 weeks. Such long times to generate waveforms and thereby, the power reports were the bottleneck in estimating and optimizing the power of the IP. Simulation methodology was clearly not feasible to ensure that the power activities were carried out efficiently. In display, each frame is of length 16ms for 60Hz refresh rate and RTL simulations for the full frame usually takes 3-4 days. So, an active region of the frame is usually selected to estimate the power, but this approach is inconclusive. Also, for higher resolution tests like UHD and 8K, estimating the power for the entire frame in pre-silicon environment is challenging. To overcome this bottleneck, emulation infrastructure was set-up which reduced the long-run times in generating waveforms thereby helped in running multiple workloads. Apart from the regular speedup, emulation framework also provided optimizations to reduce the runtimes even more and one such optimization technique is shown in Fig. 3.

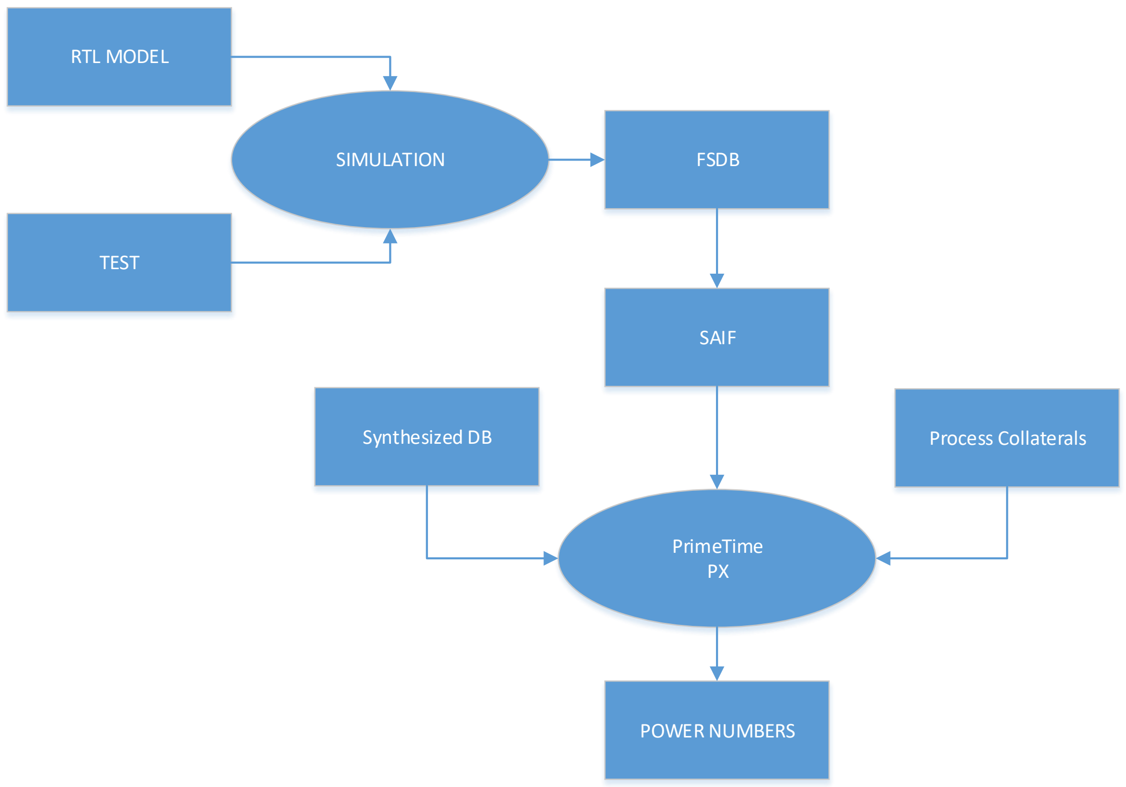

Figure. 1: Power Estimation Flow in Simulation

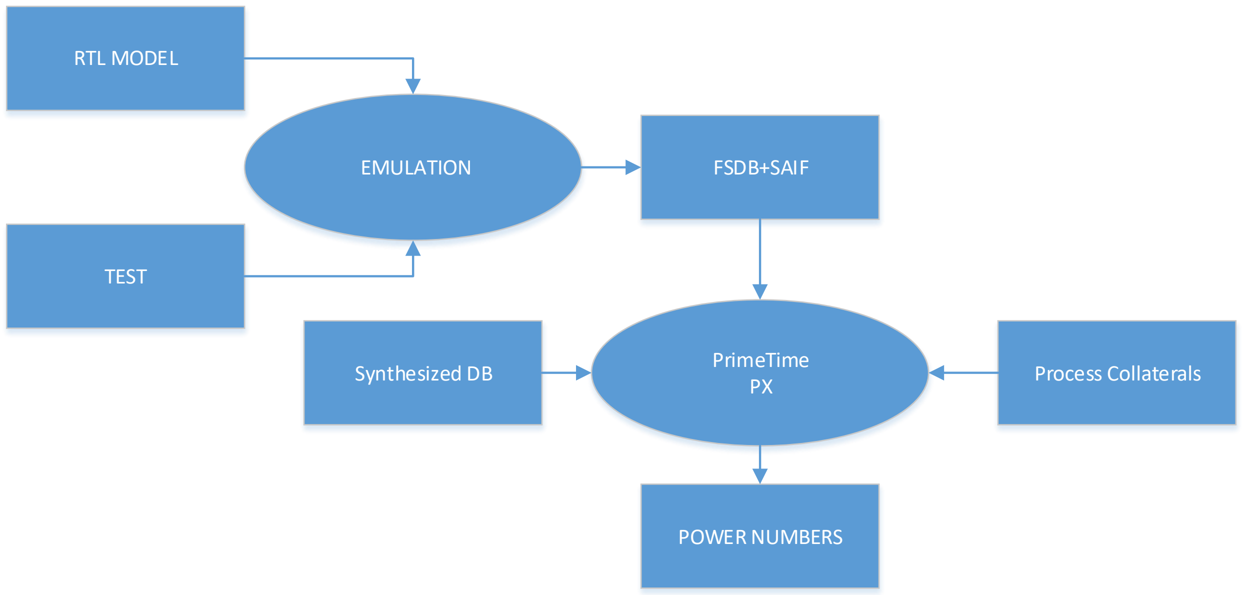

Figure. 2: Power Estimation Flow in Emulation

II. SPLIT METHOD

In regular SAIF run, the test is fully run and switching activity is captured at every edge. In split method, the full waveform run is divided into multiple individual smaller waveform runs (customizable). These can be run in parallel to generate switching activity interface file (SAIF) and combined to create the full SAIF file. In split method, all simulations will start from time 0 but, the SAIF is dumped only for the desired period. For example, SAIF run from 12-16ms will run till 12ms but will start generating SAIF from 12 to16ms as shown in Fig. 3. This entire flow is automated without any intervention from the end user. The number of split runs is customizable, and the technique improved the speedup and reduced the runtime by 4x for 4-split runs and 8x for 8-split runs. Apart from generating power estimation data, the emulation method also provides a graphical plot of peak and average power for the entire simulation, and a graphical representation of switching activity across time for all partitions in the design which assists in finding and debugging power hotspots in the design.

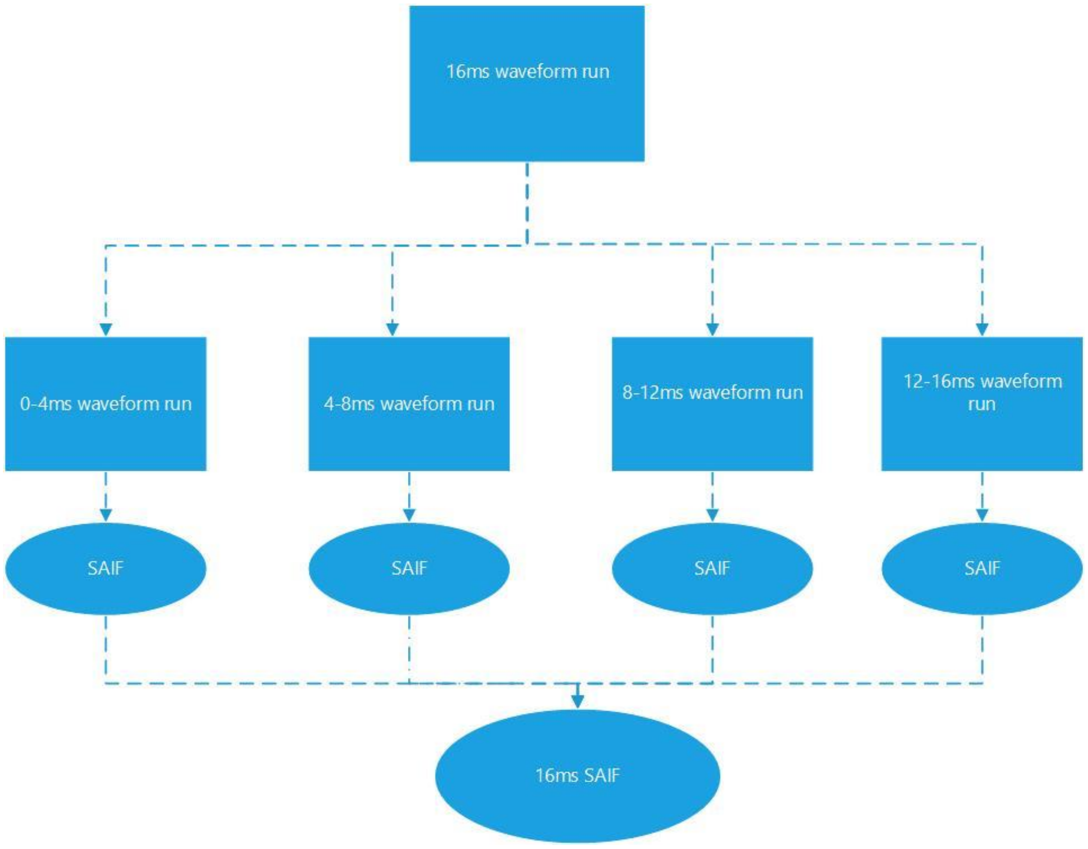

Figure. 3: Split Run method

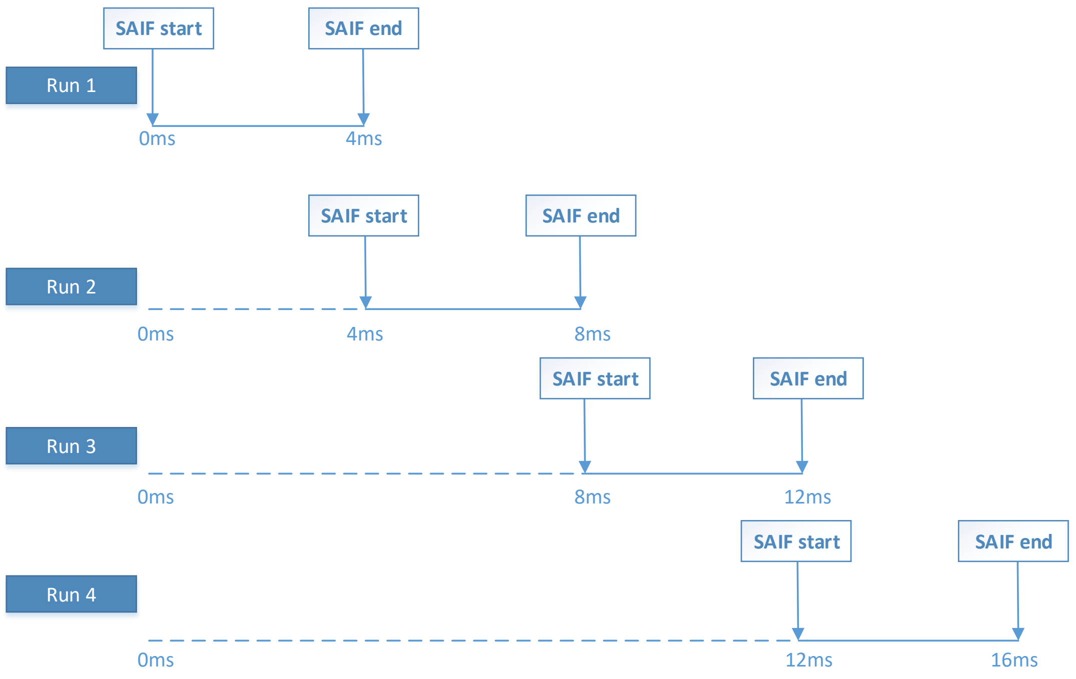

Figure. 4: Split Run Simulations

III. IMPACT

Due to long runtimes and huge waveforms dumps, the SAIF in simulation is only generated for 3ms during active portion of the frame and is scaled to get the activity for the entire frame (~16ms). The scaling is done by running the simulation occasionally for entire frame and finding the power scalar between the 3ms window and full frame (16ms). This can be used only for synthetic tests which use same image file and cannot be used for randomized tests. The simulation will only provide approximate power consumption by the design but doesn't provide overall accurate power consumption and might miss any power hotspots in the design. Emulation provides a framework to generate SAIF for full frame by running various workloads.

IV. RESULTS

1. Run without split method:

	Time Taken for 16ms 4K frame
Simulation Run	72 hours
Emulation Run	26 hours

2. Emulation run with split method:

a. 4-split runs: Total runtime is ***7 hours 4 mins*** because all four runs are parallel.

0 – 4ms	7 hours
4ms – 8ms	7 hours 1 mins
8ms – 12ms	7 hours 3 mins
12ms – 16ms	7 hours 4 mins

b. 8-split runs: Total runtime is ***3 hours 11 mins*** because all eight runs are parallel.

0 – 2ms	3 hours 6 mins
2ms – 4ms	3 hours 7 mins
4ms – 6ms	3 hours 7 mins
6ms – 8ms	3 hours 8 mins
8ms – 10ms	3 hours 9 mins
10ms – 12ms	3 hours 10 mins
12ms – 14ms	3 hours 10 mins
14ms – 16ms	3 hours 11 mins

The results shown has few limitations like waveform uploads provided by the emulation platform. Also, the 10x speedup is due to having multiple clocks in the design but other designs with fewer clocks can see more than 10x speedup. The speedup can be increased even more by using other platforms having no limitations on the waveform uploads.

V. CONCLUSION

Designs not meeting power requirements are bad as not meeting functional requirements and when there is tighter time to market schedules, it is very important to run various workloads covering multiple scenarios to identify power hotspots in the design. Relying on simulation methodology for power estimation can be efficient for smaller test cases but for testcases with large frame rates and multiple frames, it becomes inefficient and emulation improves the speed-up to run various workloads.