

Power Analysis and Optimization Techniques of an 8-bit FIR Filter from RTL through GDSII

C. Ashok Kumar
Prof.& Head, Dept.Of ECE,
CMR Engineering College,
Hyderabad, India

B.K.Madhavi, PhD.
Prof., ECE Dept,
KMIT, Narayanguda
Hyderabad, India

K. Lal Kishore, PhD.
Vice-Chancellor,
JNTU, Ananthapur,
Andhra Pradesh, India

ABSTRACT

Power dissipation in CMOS circuits has put forth many technical challenges for VLSI design engineers. Dynamic power and leakage power have increased with increase in frequency of operation and transistor scaling. Signal and image processing applications require FIR filter as a sub system which has multipliers and adders as sub units. Reducing power dissipation in FIR filter will reduce power dissipation in complex circuits. In this paper, we present design and analysis of FIR filter optimizing area, power and speed performances. The low power techniques and area optimization techniques as recommended by EDA tool vendors are evaluated and optimum constraints are chosen to get the best estimation of power and speed performances. An 8-bit FIR filter is designed using Matlab FDA tool, the HDL model developed is synthesized using ASIC design tools and physical design is carried out. Various optimization techniques at every stage is used to constrain the design for optimum implementation. Choice of multipliers and adders and their impact on power is estimated. The design oriented techniques adopted to achieve power savings up to 36% using the clock gating and using the data path operator isolation here, 7% of the power has been saved. Power targets achieved would be compared to match quantitatively with the power numbers of the same design at gate level netlist after synthesis and gate level netlist after Placement & Routing.

Keywords: RTL power, early estimation, Design Oriented Reduction techniques

1. INTRODUCTION

Historically, VLSI designers have used circuit speed as the "performance" metric. Large gains in terms of performance and silicon area have been made for digital processors, microprocessors, DSPs (Digital Signal Processors), ASICs (Application Specific ICs), etc. In general, "small area" and "high performance" are two conflicting constraints. The IC designers' activities have been involved in trading off these constraints. Power dissipation issue was not the design criterion but an afterthought. In fact, power considerations have been the ultimate design criteria in special portable applications such as wristwatches and pacemakers for a long time [1]. Battery-powered systems such as laptop/notebook computers, electronic organizer are the key products that are in demand and low power consumption is desired as they operate on remote power sources. The need for these systems arises from the need to extend battery life. Many portable electronics goods use the rechargeable Nickel Cadmium (NiCd) batteries [2][3]. Although the battery industry has been making efforts to develop batteries with higher energy capacity than that of NiCd, a strident increase does not seem imminent. The expected improvement of the energy density is 40% by the turn of the century. With recent NiCd batteries, the energy density is around 20 Watt W hour/pounds and the

voltage is around 1.2 V [1]. So, for example, for a notebook consuming a typical power of 10 Watts and using 1.5 pound of batteries, the time of operation between recharges is 3 hours. Even with the advanced battery technologies, such as Nickel-Metal Hydride (Ni-MH), which provide large energy density characteristics (- 30 Watt-hour/pound), the lifetime of the battery is still low [1]. Since battery technology has offered a limited improvement, low-power design techniques are essential for portable devices [4].

Low-power design is not only needed for portable applications but also to reduce the power of high-performance systems. With large integration density and improved speed of operation, systems with high clock frequencies are emerging. These systems are using high-speed products such as microprocessors. The cost associated with packaging, cooling and fans required by these systems to remove the heat are increasing significantly. Table 1 shows SIA Roadmap for Power Dissipation in Current and Future Microprocessors [1]. This table demonstrates that, at higher frequencies, the power dissipation is too excessive.

Table 1: SIA Roadmap for Power Dissipation in Microprocessors

Year	1999	2002	2005	2008	2011	2014
Feature size(nm)	180	130	100	70	50	35
Logic trans/cm ²	6.2M	18	39M	84M	180M	390M
Clock (MHz)	1250	2100	3500	6000	10000	16900
Chip size(mm ²)	340	430	520	620	750	900
Power supply(V)	1.8	1.5	1.2	0.9	0.6	0.5
Power(mW)	90	130	160	170	175	183

Chandrakasan et.al [5] while discussing the power-down strategies in synchronous designs, mentioned the importance of minimizing switching activity by powering down execution units when they are not performing "useful" operations. This is an important concern since logic modules can be switching and consuming power even when they are not being actively utilized. H. Kapadia et.al [6] undertook the concept of control-signal gating to reduce the amount of power being wasted in the data path due to switching activity that does not contribute to the functionality of the circuit. This can be used early in the design flow. Section II discusses FIR filter design, section III discusses low power techniques, section IV and section V discuss power reduction techniques, section VI discusses low power design flow, and section 7 is conclusion.

2. FIR FILTER DESIGN

A filter is used to modify an input signal in order to facilitate further processing. A digital filter works on a digital input (a sequence of numbers, resulting from sampling and quantizing an analog signal) and produces a digital output. According to Dr. U. Meyer-Baese [7], “the most common digital filter is the Linear Time-Invariant (LTI) filter”. Designing an LTI involves arriving at the filter coefficients which, in turn, represents the impulse response of the proposed filter design. These coefficients, in linear convolution with the input sequence will result in the desired output. The linear convolution process can be represented as [8]:

$$y[n] = x[n] * f[n] = \sum_k x[k]f[n-k] = \sum_k f[k]x[n-k] \quad (1)$$

$y[n]$ signifies the output of the filter and $x[n]$ is the digital input to the filter. A filter with a finite value for k is said to be a Finite Impulse Response (FIR) filter. It can be inferred that the output of an FIR filter remains dependant only on the inputs and the coefficients. Therefore, the FIR filter detailed above is an LTI filter [8]. Equation (1) can be re-written as follows, for an order of L , as follows:

$$y[n] = x[n] * f[n] = \sum_{k=0}^{L-1} f[k]x[n-k] \quad (2)$$

Figure 1 shows the schematic of an FIR filter of order L :

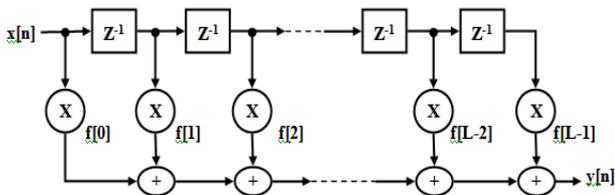


Figure 1: FIR filter of order L , with constant coefficients

Calculating the constant coefficients of such a digital filter involves considerable amount of computation and this is generally performed using software tools. The Filter Design and Analysis (FDA) tool packaged along with MATLAB is such a tool. The coefficients of an FIR filter, as mentioned earlier, denote the impulse response of the filter. It is imperative for any system implementation of such a filter to use a number format that represents the coefficients to as much precision as allowed by the resource constraints. The double length floating point notation for filter coefficients, used by the FDA tool poses immense challenges in terms of cost and resources, while implementing on an FPGA. To overcome this, the filter coefficients have to be quantized to a fixed point notation, resulting in the introduction of a certain amount of imprecision. This chapter details the process of designing filters and analyzing the effects of coefficient quantization on the overall response using the MATLAB FDA tool. Hardware description for the filter implementation is generated in Verilog HDL and simulations of the hardware description are performed using Modelsim.

The Filter Design and Analysis Tool (FDA Tool) is a graphical user interface (GUI) available in the Signal Processing Toolbox of MATLAB for designing and analyzing filters. It takes the filter specifications as inputs. Table 2 shows the specifications of an FIR low pass filter.

Table 2: FIR filter specifications

Filter performance parameter	Value
Pass band frequency	11000 Hz
Pass band attenuation	3 dB
Stop band frequency	12000 Hz
Stop band attenuation	65 dB
Sampling frequency	48000

The sampling frequency is chosen as 4 times the stop band and the filter has a steep transition band with a width of 1000 Hz. These specifications are fed as inputs to the FDA tool in MATLAB R2010b. The tool performs the filter design calculations using double precision floating point numeric representation and displays the response of a reference filter. The designed filter is of order 84. It must be noted that the FDA Tool uses double precision floating point representation for the design calculations. This allows the tool to achieve a fair degree of precision, which is reflected in the close-to-ideal response of the reference filter. Figure 2 shows the response of the reference filter, in detail. Specifications of the filter, namely, pass band, stop band, transition band, pass band ripple and stop band ripple are denoted in the screen shot.

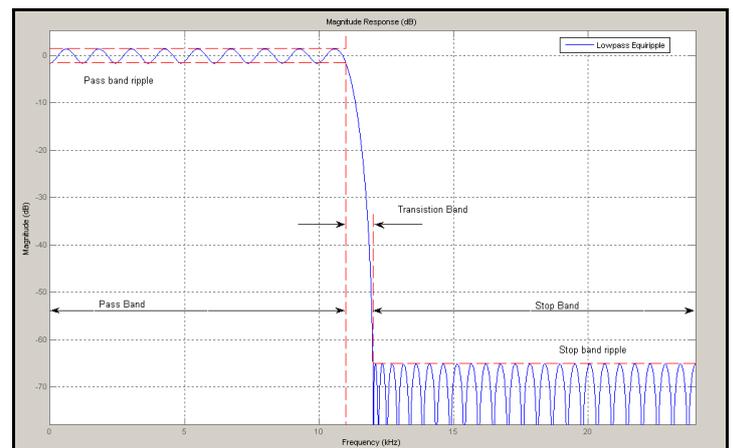


Figure 2: Response of the reference filter

The response shown is calculated from 0 Hz (DC) up to 24000 Hz, which is half of the sampling frequency specified ($F_s / 2$). The pass band response ripples with 0 dB as centre and the stop band ripples are all below 65dB in magnitude. The filter has a steep transition band starting at 11000 Hz, achieving stop band attenuation at 12000 Hz. The accuracy of a digital filter is limited by the finite word length used in its implementation. When a filter is constructed with digital hardware, the minimum word length needed for specified performance accuracy must be determined [9]. An ideal filter requires infinite word length to truly represent the filter coefficients. However, the resource constraints associated with hardware implementation pose the challenge of using fixed point arithmetic in VLSI implementations, for the sake of cost and speed. The simplest and most widely used approach to the problem is to round off the optimal infinite precision coefficients to a b -bit representation [9]. Two such formats namely, Q 16.14 and Q8.7 are considered to analyze the effects of varying the word length while quantizing the filter coefficients. As a result of the fixed

point finite word length used in a digital filter, each coefficient is replaced by its t-bit representation. That is, the coefficient a_k is replaced by $(a_k + \alpha_k)$, with α_k bounded in absolute value by 2^{-b} . Similarly, b_k is replaced by $(b_k + \beta_k)$. The new transfer function $[H(z)]_t$ is given by [9]:

$$[H(z)]_t = \frac{\sum_{k=0}^M (b_k + \beta_k) z^{-k}}{1 + \sum_{k=0}^L (a_k + \alpha_k) z^{-k}} \quad (3)$$

3. CMOS POWER CONSUMPTION

Generally, power-efficient CMOS design [3] involves the minimization of one or more of the terms in the basic power consumption equation

$$P = C_L V_{dd}^2 f + V_{dd} I_{dd} \quad (4)$$

Or, in its more detailed version

$$P = C_L V_{dd} V_{swing} f + V_{dd} Q_{sc} f + V_{dd} I_{lkg} + V_{dd} I_{through} \quad (5)$$

Where P represents the total power consumed, V_{dd} represents the supply voltage, I_{dd} represents the static current drawn from the supply, C_L represents the load capacitance, and f represents the switching frequency. The VI term represents the static, or DC power consumption, while the CV^2f term represents the dynamic power consumption. In the more detailed version, V_{swing} represents the signal voltage swing (which for CMOS is usually equal to V_{dd}), Q_{sc} represents the charge consumed due to short-circuit Momentary current (also known as crowbar current) drawn from the supply during switching events, I_{lkg} represents the parasitic leakage current, and $I_{through}$ represents the (by design) quiescent static current. The first two terms of this equation represent the dynamic power consumption, while the latter two represent the static power consumption.

Until relatively recently, the design and design automation communities viewed low-power design as being primarily focused on the CV^2f component; however, with chips inexorably becoming bigger, faster, and more power-hungry, it has become clear that the problems, as well as their solutions, are much more complicated.

4. DYNAMIC POWER CONSUMPTION COMPONENT

The dynamic power dissipation P_{dyn} , is caused by the charging and discharging of parasitic capacitances in the circuit [4]. Illustrated below are the computations of the dynamic dissipation through the example of a CMOS inverter driving a load capacitor C_L .

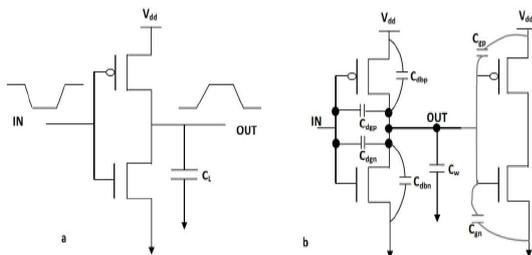


Figure 3: The function of CMOS inverter: (a) CMOS inverter, (b) capacitor C_L elements, (c) charging phase, and (d) discharging phase.

As shown in Figure 3(a) The load capacitance C_L depicted in Figure 3(b) consists of the gate capacitance of subsequent inputs attached to the inverter output C_{gp} and C_{gn} , interconnect wire capacitance C_w , and the diffusion capacitance on drains of the inverter transistors C_{gdn} , C_{gdp} , C_{dbn} , C_{dbp} . P_{dyn} comprises the sum of two power components: the first one occurs during low-to-high the output transition (i.e., charging phase), while the second one during high-to-low transition (i.e., discharging phase). More specifically, for every low-to-high output transition in a digital CMOS gate, the capacitance C_L on the output node incurs a voltage change ΔV , drawing energy of $C_L (\Delta V)^2$ (joules from the supply voltage). During this process, one-half of the energy is stored in the capacitor, whereas the second half is dissipated in the PMOS and interconnects wire. In the case of simple inverter, it holds $\Delta V = V_{dd}$, and thus, the power consumption is given by:

$$P = C_L V_{dd}^2 a_{0 \rightarrow 1} f \quad (6)$$

Where $a_{0 \rightarrow 1}$ is an activity factor that represents the average fraction of clock cycles in which a low-to-high transition occurs, and f is the clock frequency. Similarly, a high-to-low transition dissipates the energy stored on the capacitor C_L in NMOS transistor, pulling the output low. Consequently, the total dynamic power consumption is given by the golden formula:

$$P_{dyn} = C_L V_{dd}^2 a f \quad (7)$$

Here, we focus on the circuit level techniques for reducing the P_{dyn} power component. The remaining two power components are analyzed and appropriate techniques are discussed in other chapters. It should be stressed that the circuit techniques described for reducing dynamic power consumption may have impacts on the performance and silicon areas as well as on the remaining power components.

5. POWER REDUCTION APPROACHES

Equation (7) calculates the dynamic power consumption, of CMOS logic gates. It can be easily inferred that P_{dyn} is proportional to the load capacitance, C_L , the square of V_{dd} , the switching activity, a, and the clock frequency, f. Consequently, power consumption reduction can be achieved by:

- Reducing of output capacitance, C_L
- Reducing of supply voltage, V_{dd}
- Reducing of switching activity, a
- Reducing of clock frequency, f

Thus, a designer should devise new techniques aiming at the decrease of each above-mentioned parameter or any combination of them. A very popular low strategy concerns the reduction of the switched capacitance or effective capacitance, C_{eff} , which is defined as the product of output capacitance times switching activity (i.e. a. C_L). Generally, the two main low-power reduction strategies concern the reduction of supply voltage and the switched capacitance. The reason is that we consider the throughput rate of a low powered-designed circuit remains the same with an existing circuit. In particular, the reduction of power supply voltage has the major impact on the power consumption due to the quadratic dependence of V_{dd} . Although such reduction is usually very effective, the circuit delay increases and system throughput degrades. In addition, the shift of industry from a supply voltage to a smaller one is quite expensive and slow due to, for instance, the compatibility issues of input/output signals with the peripheral circuits. In contrast, the reduction of the switching activity or the capacitance for a certain technology depends mainly on the designer's creativity. Thus, someone can reuse an existing silicon technology achieving satisfactorily level of power consumption without the need for purchasing new technology libraries, which may lead to design cost reduction. In other words, a designer may proceed to a more advance silicon technology only if he or she has explored all the possibilities for realizing a circuit with an existing technology considering the design cost and time-to-market constraints. The reduction of switching activity requires among others a detailed analysis of signal transition probabilities, careful redesign of circuit nodes with high activity, balanced paths, and selection of appropriate logic style. The capacitance load can be reduced by, for instance, technology scaling, transistor resizing, and logic family selection.

6. INTRODUCTION TO LOWPOWER DESIGN FLOW

It is at the beginning, when the complexity is still small and can well be understood under different aspects, that the important decisions are made, which will lead to success or failure. Once a design has been developed to a large structure of logic and wires, it is difficult to cure problems, which, in many cases, also started small and eventually became large, hard to solve, and without major design re-spins, these problems may cost months of design time, major engineering resources, and can be responsible for missed marketing opportunities. This chapter covers the low power design aspects for the complete ASIC flow, concentrating primarily on the approaches at the RTL level and its advantages.

6.1 Design flow overview

Given an appropriate variety of tools, effective use is often dependent upon a well-structured design flow. For example, for power optimization as for other parameters, such as performance and cost, it is critical to architect the system properly at the beginning and successively refine it as the project proceeds. Such a multilevel approach increases the likelihood of meeting design goals by providing both early visibilities into critical issues as well as multiple opportunities for mitigation. Much of digital design is performed today utilizing a top-down or modified top-down design flow. Here top refers to the higher levels of design abstraction, such as the system, behavior, and register transfer (RT) levels, and time flows downward toward the lower levels of design abstraction, such as the gate and transistor levels. In this case, flow refers to the sequence of tasks; however, the flow of detailed design information is

somewhat less clear. In conventional practice, detailed design information tends to follow a feedback design flow, wherein information about particular power characteristics does not become available until the design has progressed to the lower abstraction levels. A feedback design flow features a relatively lengthy feedback loop from the analysis results obtained at the gate or transistor level back up to the design tasks at the RT level and above. Thus, information about the design's power characteristics is not obtained until quite late in the design process. Once this information is available, it is fed back to the higher abstraction levels to be used in determining how to deal with the power issues of concern. The farther the lower-level power analysis results exceed the target specification, the higher the abstraction level in which the design must be changed.

By comparison, a feed forward approach, illustrated in Figure 2, replaces these lengthy, cross abstraction feedback loops with more efficient abstraction specific loops. Thus, the design that is fed forward to the lower abstraction levels is much less likely to be fed back for reworking, and the analysis performed at the lower levels becomes essentially a verification task. The key concept is to identify, as early as possible, the design parameters and trade-offs that are required to meet the project's power specs. This helps to ensure that the design being fed forward is fundamentally capable of achieving the power targets. Later in the design flow, optimizations at the lower levels can be used to further minimize the power as desired. A high-level analysis tool, such as Power Theater, which can accurately predict power characteristics, enables the feed forward flow. These early, high-level analysis capabilities are employed to make informed trade-offs, such as which algorithms and architectures to employ, without having to resort to detailed design efforts or low-level implementations to assess performance against the target power specification. Compared with the traditional top-down methods, the key difference and advantage is added by the early prediction technology. Proceeding in parallel with, or sometimes ahead of, the architecture development is the design of the library macro functions and custom elements, such as datapath cells. These are used in the subsequent implementation phase in which the RTL design is converted into a gate-level netlist. At this point, appropriate optimizations are performed again, and power is re-estimated with more detailed information, such as floor planned wiring capacitances. The power grid is planned and laid out using this power data. Once the design has been synthesized into a technology mapped gate-level netlist, lower level power optimizations can be employed, using a tool to further reduce dynamic or leakage power consumption. Specific goals or issues, such as battery life or noise margin repair, will determine the particular optimizations employed. These optimizations can be performed either before (using estimated wiring parasitics) or after routing (using extracted wiring parasitics). In either case, after the design has been routed and optimized, a final tape-out verification and electrical verification check is performed with an electrical sign-off tool. In this step, power is calculated and used to compute and validate key design parameters, such as total power consumption in active and standby modes, junction temperatures, power supply droop, noise margins, and signal delays. Thus, power is analyzed and optimized multiple times, at each abstraction layer following the feed forward approach. Each analysis is successively refined from the previous analysis by using information fed forward from prior design decisions along with new details produced by the most recent design activities. Each optimization, at the various abstraction layers, results in more efficient logic structures to feed forward to the downstream design tasks, thereby successively squeezing out the wasted power. This approach encourages design efforts to be

spent up front, at the higher abstraction levels, where design efforts are most effective in terms of minimizing and controlling power. In addition, because power-sensitive issues are tracked from the beginning to the end, the likelihood of a late surprise issue is minimized.

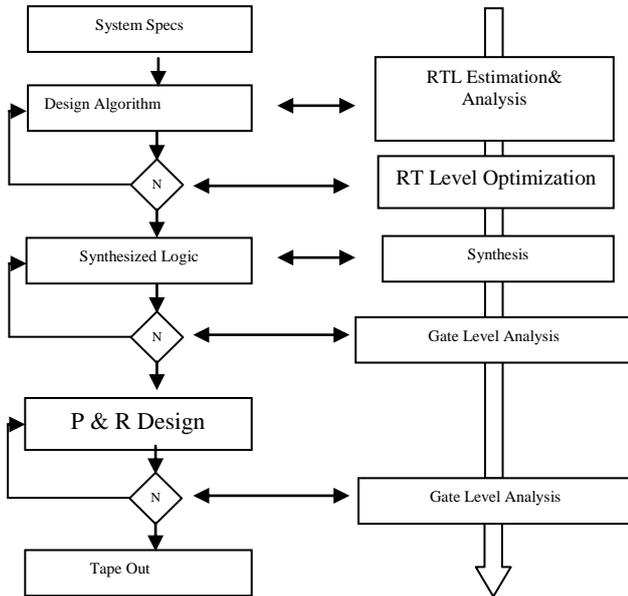


Figure 4: Feed Forward Design Flow

The techniques mentioned above help understand the underlying concepts as well as the limitations of the current low-power design flow compared to the one that is proposed in the paper, shown below in the figure 4, which shall guide in optimizing the global system architecture for low power and help them selecting and further optimizing the algorithms to be implemented at lower levels. The figure of merit in reducing the power consumption by making the right decisions during this early phase covers several orders of magnitude.

Just to illustrate the potential: For a given design, there could exist three to four known and well-understood algorithms. They all perform exactly the same task: taking a set of inputs and process them in an order according to the chosen architecture. Despite the exactly same functional behavior, however, they all perform differently with respect to the computation time, memory usage, and the power consumption. Selecting the most power efficient one can be a product-differentiating factor. In general flow the power numbers of the designs are obtained only after the synthesis stage, which means all the designs need to be taken through the ASIC flow till synthesis stage. This would consume lot of time and eventually in the time to market for the chip would increase. In the proposed flow, the power numbers could be obtained at the RTL level itself, by estimation saving the time taken for the designs to be taken till the synthesis stage.

6.2 RTL Level Estimation and Analysis

Analysis is based on an existing design at any level (i.e., the structure is given, typically in terms of a netlist of components). These modules are pre-designed and for each one a power model exists. These power models can be evaluated based on the activation of the modules. Thus, power analysis is the task of evaluating the power consumption of an existing design at any level. It is used to verify that a design meets its power and reliability constraints (e.g., no electron migration occurs, no hot spots will burn the device, and no voltage drops will cause

spurious timing violations). Power analysis finally helps to select the most cost efficient chip package. In contrast, estimation builds on incomplete information about the structure of the design or part of the design under consideration. The design does not yet exist and can only be generated based on assumptions about the later physical implementation of the design, its modules, its interconnect structure, and physical layout. In summary, estimation requires design prediction followed by analysis; for instance, if the floorplan of a design is not yet available, interconnect power estimation first requires a floorplan prediction. Power estimation is applied to assess the impact of design decisions and compare different design alternatives on incomplete design data.

6.3 Power Analysis

The power analysis of the design is done using a high-level analysis tool, which supports the proposed flow. The chart shown below in the figure 5 shows the generic analysis flow.

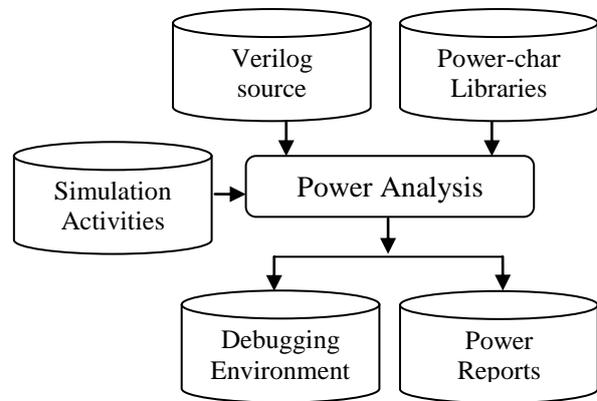


Figure 5: Power Analysis Flow

The Verilog source provides the Netlist input for the analysis and the power character library inputs provide the power information to be taken from for analysis. The simulation activity inputs are given in the form of a vcd file and input frequencies, to provide the switching activity information. The out puts would be the power reports and a debugger environment.

Power Contribution	new.res		
	Static	Dynamic	Total
Internal power	6.81uW	5.48mW	5.49mW
IP Core power	0W	0W	0W
Pad power	0W	0W	0W
Clock power	340nW	843uW	844uW
Total power	6.82uW	9.32mW	9.34mW

Figure 6: Power Summary at RTL stage

The static and the dynamic power numbers of the 8-bit FIR are shown in the figure 6 along with the clock power. In order to reduce power at the sub system level two major power reduction techniques have been adopted in this design. Figure 7 shows the techniques. In Figure 7(a) the concept of clock gating is introduced by having an AND gate thus reducing more than 50% of power. In Figure 7(b) logic level reducing is introduced that reduces power by more than 10%. Both the techniques are adopted at the subsystem level.

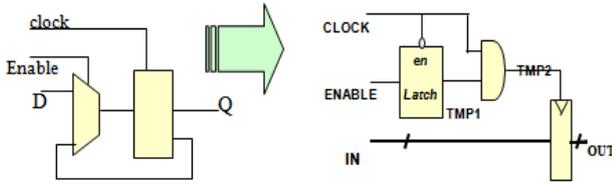


Figure 7(a)

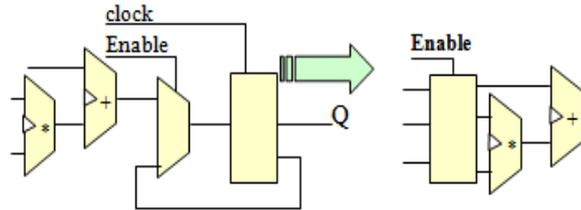


Figure 7(b)

Figure 7(a & b): Power reduction techniques

6.3.1 RTL Reduction:

The RTL reduction is performed on the design to identify the blocks where the power could be reduced. As the design used here is a simple 8-bit FIR filter, all the reduction techniques couldn't be applied, but the most important ones discussed, like converting a register file to latch file and explicitly enabling the clock, which use clock gating concept. The reduction index points to the code as shown in the figure below, to which parts of the code the reductions could be applied along with suggestions and the instance in the hierarchical tree. The percentage savings that could be gained are calculated as well. Following these suggestions the code could be reduced and the reduction in the power numbers could be attained. In the below figure, the first blue column on the top shows the local explicit clock enabling reduction methodology that could be applied to the design. It also shows the source code, where the methodology could be implemented. From results, the savings expected is around 46%. The savings achieved by implementing the suggested methodology is 32%. Data path operator isolation is the second methodology, that's adopted in the design. By adopting the above-mentioned methodology, savings up to 7% was expected results achieved with the modification of the code is 5%. Figure 8 shows the synthesis flow adopted in this work.

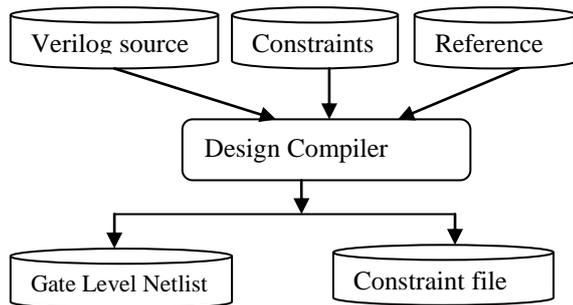


Figure 8: Synthesis Flow

The frequency of operation of the design is 370 MHz and the slack is 0. The DC part is done to get a synthesized netlist which needs to be fed to the later parts of the flow and to get power numbers, which serve as a mode of comparison between the prime power results and DC results. The power consumed by the synthesized netlist as given by DC is 6 mW with 4μW of leakage power. The leakage power numbers of the prime power (5μW) closely match with the static numbers of DC (6μW). The dynamic results of the prime power runs (10mW) do not match

quantitatively with the numbers of the DC runs. The reason being the dynamic power is primarily dependent on the switching activity of the inputs. In the case of the dynamic power numbers of Prime Power, the VCD file provides the switching activity. Comparing the results with the zeromim flow at the RTL stage, the static internal power is the same, because the internal power is taken from the power character libraries and the libraries are same, so the internal remained the same. Coming to the static clock power, here at the RTL stage, the clock power is analyzed using a clock nets file, which includes the name of the clock and the clock buffers to be used for the clock tree. Because of this clock nets file, the static power numbers and the dynamic power numbers of the clock have gone high and more accurate.

6.3.2 Gate level analysis: P&R Design

The gate level Netlist from DC is given to a physical design tool, here ASTRO to get a placed and routed design. The inputs to the ASTRO from the synthesis stage are the Netlist file and an sdc file, giving the design constraints. The reference library is taken. The physical design flow is represented below in Figure 9.

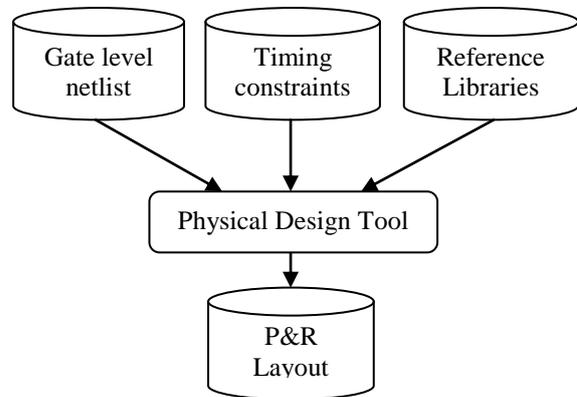


Figure 9: P&R Flow for FIR filter



Figure 10: Power planning of the 8-bit FIR

The net list obtained from ASTRO is floor planned, placed and routed. The net list is taken from ASTRO along with the spec files (Figure 10). These are used to get the power numbers for

the physical design stage. The snapshots of the power planned, a routed clock is shown in the figure 11. The power numbers achieved throughout the flow are compared below.

RTL: **8.42mW**
Gate Level: **10.71mW**
P & R: **15.13mW**

From the above results, it could be seen that the power numbers achieved became more accurate at the later levels of abstraction.

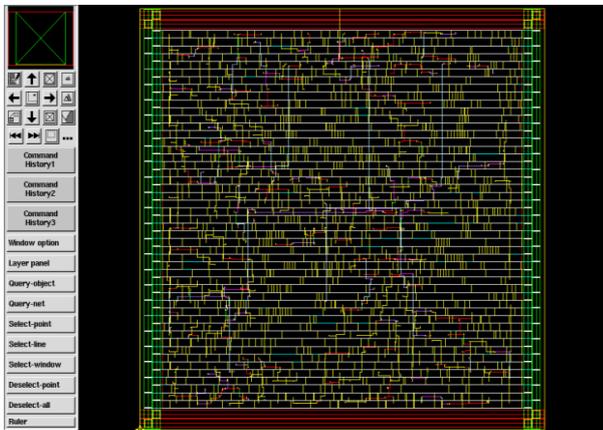


Figure 11: Routed Clocks

7 CONCLUSIONS

Power-sensitive design has become an essential focus in this age of wireless and multimedia computing, but it is no longer directed at simply reducing the amount of power consumption. Power-related issues now directly affect many facets of design and these issues are sufficiently complex as to require significant amounts of design automation. RTL design and analysis early in the design process is done and the reviewed design oriented techniques are adopted in the design to achieve considerable reductions. The adoption of design-oriented techniques has given an early visibility of power. The feed forward design flow approach makes the power analysis process easier. Using the clock gating concept in the present design, 36% of the total power has been saved. Using the data path operator isolation here, 7% of the power has been saved.

8. REFERENCES

- [1] The National Technology Roadmap for Semiconductors, Semiconductor Industry Association, 2005.
- [2] http://www.avicenne.com/gb/batteries_and_power_supply_publications.htm, Dec, 2006.
- [3] Sequence Design, Power Theater Reference Manual, Sequence Design Inc., Santa Clara, CA, 2006.
- [4] Christian Piguet “Low-power CMOS Circuits Technology, Logic Design and CAD Tools”, CRC Press, ISBN 0-8493-1941-2, 2005.
- [5] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power CMOS digital design IEEE J. of Solid-State Circuits”, Vol. 27, pp. 473–484, 1992
- [6] H. Kapadia, L. Benini, and G. De. Michele, “Reducing switching activity on data path busses with control-signal gating”, IEEE J. Solid-State Circuits, vol. 34, pp 405-414, Mar. 1999
- [7] U. Meyer-Baese – Digital Signal Processing with Field Programmable Gate Arrays Second Edition – Springer, p.109.
- [8] KODEK - Design of Optimal Finite Word length FIR Digital Filters Using Integer Programming Techniques - IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 3, JUNE 1980.
- [9] Wonyong Sung and Ki-II Kum - Simulation-Based Word-Length Optimization Method for Fixed-point Digital Signal Processing Systems - IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 43, NO. 12, DECEMBER 1995.
- [10] Albert Tung-Hoe Wong, New scalable systolic array processor architecture for discrete convolution
- [11] D.Rutland, “Why Computers Are Computers”, Wren Publishing, 1995. [8] W. Nebel and J. Mermet, eds., “Low-Power Design in Deep Submicron Electronics, NATO ASI Series”, Kluwer Academic Publishers, 1997
- [12] Benini, L. and de Micheli, G., “System-level power optimisation: techniques and tools”, ACM Trans. on Design Automation of Electronic Syst., Vol. 5, pp 115–192, 2000.
- [13] Donno, E. Macci, and L. Mazzoni, “Power-Aware Clock Tree Planning”, Proc. ACM/IEEE Int. Symp. Physical Design, pp. 138-147,2004
- [14] J. Rabaey and M. Pedram, Ed., “Low Power Design Methodologies”, Kluwer, 1995.