

Datapath Design Closure Challenges: An Emphasis on Speed, Power and Area

Dr. S. Gayathri¹, G.R. Sumithra²

Dept. of E&C Dept. of Electronics

¹ Sri Jayachamarajendra College of Engineering Mysuru

sgmurthy_65@sjce.ac.in

² MMK & SDMMM, Mysuru

arish.sumithra@gmail.com

Abstract—There is a lot of change in the recent design closure flows, a transition from standalone tools for synthesis, static timing analysis, placement and routing to an integrated design flows. This gradual change is due to scaling of technology, increase in design complexity and increasing demand for time to market. Several powerful features in the integrated backend design tool help in dealing with modern design requirements and to meet aggressive targets. The advanced analysis flows are used in the design of complex processor blocks. The major challenges faced in the design closure of latest chip blocks are leakage power, delay, area and reliability. This paper presents advanced techniques to minimize power dissipation, meet desired frequency without sacrificing the circuit quality and to arrive at an optimal design for a datapath logic block. But in a race to achieve these mentioned design constraints it is very much necessary to ensure the functional correctness of the design. Thus a formal verification method for maintaining equivalence between RTL and gate level netlist is carried out concurrently along the designflow.

Index Terms— **Timing Closure, Max path, Min path, Activity factor, dynamic power.**

INTRODUCTION

The high performance core is composed of several functional building blocks. Each of these blocks have logical dependence on the neighboring blocks. Design closure of a synchronous core block includes various challenges like meeting setup and hold requirements of the sequential elements, achieving design power targets and maintaining functional correctness of the Functional Unit Block. The initial design stage consists of defining circuit specifications and design constraints. The block goes through numerous design manipulations in order to meet specific constraints like area, frequency, power etc [5]. It is necessary to adapt new techniques and find new ways to meet timing of the critical paths, obtain considerable power gain and area optimization especially for complex Core blocks constructed using cutting edge technology.

This paper mainly focuses on performance verification of a datapath core block. Netlist and layout for datapath core block is implemented manually [10]. Datapath blocks provide greater opportunities for logic optimization by manual schematic changes for timing, area and power benefit. At lower technology nodes the cell delay is dominated by interconnect delay specifically for critical paths [11]. Impact of Interconnect on max timing is reasonable due to their

contribution to wire delay, repeater delay and cell delay/slope degradations. It is inappropriate to pay interconnects in terms of power and timing cost since they do not implement any logic. Thus power consumed by interconnects can be minimized by optimal routing of nets having high RC delay and high switching activity in higher metal layers. Although the proposed work focuses on the backend design closure flow some of the power features and timing fixes are to be implemented in the front end RTL design [9]. Thus both the front end and backend design teams must work in close correlation in order to deliver a chip that can perform at its best. The efficient integrated design environment provides accurate parasitic estimates in the early design phase, providing high confidence design implementation before the real layout [6]. Thus in the early design phase parasitic estimation is used until the actual parasitic extraction data from the layout is available.

The paper is organized as follows: The entire datapath design flow methodology is described in Section II. Methodologies followed for timing closure are discussed in Section III. Power analysis and low power techniques are illustrated in Section IV. The performance of industrial circuits are validated w.r.t static timing analysis and power in Section V. Finally in Section VI paper is concluded and future enhancements of the proposed work is discussed.

I. DATAPATH DESIGN FLOW

The processor is composed of number of control logic and datapath blocks. Datapath design is done manually using significant amount of human resources. Only limited tools are available for the automation of the datapath structure. It is implemented using standard cells and bit-slice structure. Automation tools already available for synthesis, place and route often are not capable of identifying datapath structures for design optimization. Figure 1 depicts the detailed design flow of a datapath block. The datapath specific design (RTL- to-Layout design) approach is presented in this section.

A. High Level Datapath Design and RTL Description

For a low cycle time design process and fulfilling the design constraints an efficient datapath design is necessary. The functional design specification will be implemented by micro-architects and logic designers. As a start point in the design flow, designers generate RTL datapath architecture defining set of operations that the circuit performs. The assignment of each operation to different piped stages ensure parallelism of circuit operations. It is essential to perform optimization at this level since circuit performance and area depends heavily on this level.

Figure 1: Datapath Design Flow

B. RTL to gate-level translation, floorplanning and circuit Design

Translating an RTL model to a detailed gate-level circuit model involves identifying the hardware resources required to implement the set of operations and binding assigns operation to specific component instances. The circuit designer initially decides floorplan and implementation style depending on the design constraints. Datapath is constructed using a bit-slice



structure where layout and design of all the bits remain same. For further optimization of large, complex designs standard- cell based methods are well-suited. Layout floorplan decides the size of the block and IO pad placement. Hierarchical design representation is quite better than the flat design. The hierarchical structure of the datapath allows clear understanding of the logic, sub-structure reuse and better layout floorplan.

C. Formal Equivalence Verification

Formal verification is the mathematical approach to verify any two circuit models considered are functionally equivalent. The aim is to prove that the two circuit representations exhibit same logic behavior. Checking the equivalence between the RTL model and the implemented schematic model is the major design concern. Both these designs are read first, then the key mapping points are identified and compared by the equivalence checker tool as shown in Figure 2. The differences in reference model and the implemented model can be viewed side by side in the schematic debugger which makes the debugging simpler.

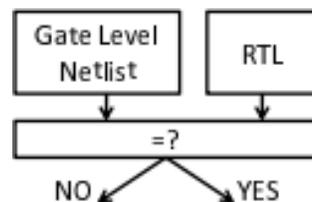


Figure 2: RTL vs netlist Equivalence Checking

D. Layout Estimation and RC Estimation

In digital integrated circuits the major contributions in the delay estimation comes from interconnect

delay due to scaling phenomenon. The critical factors associated with interconnect which determines performance in the early design phase are wire length and interconnect parasitic. Pre-layout estimations are performed from gate-level netlist. In standard-cell designs, the interconnect capacitance has become a significant part of driver loading. Wire load model is used for estimation of average load on the nets. Precision in wire delay estimations contribute for high quality of final layout implementation.

E. Timing and Quality Analysis

Timing verification of high performance integrated circuits can be achieved by Static Timing Analysis (STA) tools. Timing violations can be reduced by making trade-off between setup and hold times. The STA results should neither be highly optimistic nor pessimistic. The over-optimism lead to silicon failures, whereas over-pessimism creates unnecessary timing violations. Timing closure of processor core blocks is critical due to low cycle times and increasing chip complexity.

Quality checker tool provides rule-based quality analysis of the design. For complex designs a design rule checker is developed that performs design rule check on processor block cells by accepting netlist. The tool performs modeling, layout, electrical, timing and power checks. The violations in these categories must be detected in the early phase of design cycle

otherwise it might result in non-functional silicon in the later stages.

F. Power and Optimization

Nowadays battery life has become a bigger issue in most of the portable devices. Thus power dissipation has become the topmost design concerns. Various techniques have been employed to address power problem at various design levels. Making a proper choice of power efficient architecture is important. Power reduction is achieved at the cost of high area. Correct selection of library cells can reduce leakage power as well as active power dissipation. The power consumed by any digital circuits can be minimized by reducing capacitance, voltage or frequency. Significant power savings can be achieved by logic restructuring, clock gating, pin swapping etc.

Design optimization is performed in areas like timing, quality, area, active and leakage power. Iterative cell template replacement provides global optimization of the design. This includes insertion of low leakage cells for reducing static power, addition of min delay buffer cells for hold improvement, clock cells sizing etc. In present work Iteration with best timing/power status is implemented.

G. Layout Implementation

Physical design is the final target of the high performance datapath design. Final layout of the design is very time consuming since it is mostly handcrafted. Manually constructed datapath circuits is preferred since it has a unique bit-sliced structure. This stage involves detailed placement and routing.

Placement is carefully done in order to minimize routing congestion and to reduce total wire length. As a preliminary step metal layers usage for cell pins and ports, metal width and spacing should be defined. RC results will change by 5% after detailed place and route is completed. Post layout analysis is done with accurate RC information, once layout is DRC clean.

H. Noise Analysis

Noise injection into a circuit node causes signal logic level deviation. There is a functional failure in digital circuits due to signal level deviation. Different sources of noise are crosstalk noise, power rail noise, propagated input noise etc.

Total Noise= Propagated_Noise + Cross_Capacitance + Power Noise

Due to scaling phenomenon, the spacing between interconnects decreases and thereby increasing the coupling capacitance. Buffer insertion is applied in the post layout optimization to reduce coupling effects and improve timing. The layout break will be the last step carried out for buffer insertion in the nets. Therefore it is necessary to carefully identify and locate feasible regions for inserting buffers. Otherwise mistakes will pay high effort and complexity leading to delay in design closure. Equation for feasible buffer insertion region for meeting timing and noise constraints is derived in [2].

Next sections discusses some of the techniques followed for the timing closure and power saving in the datapath design.

II. TIMING VERIFICATION

Datapath timing verification is done using static timing analysis (STA) tools. STA tool mainly rely on the characterization data of the standard cells described in the cell libraries. Timing of synchronous circuits is verified based on the setup and hold timing constraints defined for sequential cells. In STA a path start from a CLK pin of sequential element and end at the data input of the next sequential element as shown in Figure 3.

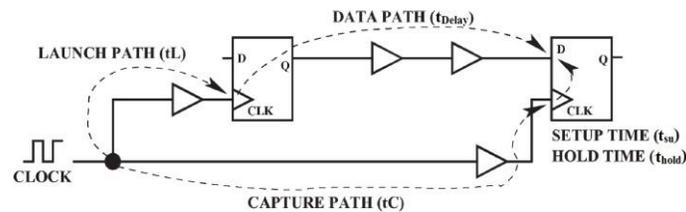


Figure 3: Example of synchronous datapath

In hierarchical design sampling point will be sequential elements inside the datapath block like latches and flip flops. STA tool takes circuit netlist, library characterization data, RC extraction data and clock period as inputs. The tool performs worst case analysis while checking path against timing constraint. The setup and hold margins are computed as follows:

$$\text{Setup_margin} = \min (tC + Tclk) - \max (tL + tDelay + tsu) \quad \text{Hold_margin} = \min (tL + tDelay) - \max (tC + thold)$$

Where T_{clk} , t_{su} , t_{hold} , t_{Delay} , t_L , t_C are clock period, setup time, hold time, combinational path delay, launch clock delay and capture clock delay respectively.

If the margin is positive then constraint is said to be satisfied or violated if it is negative. If setup time is violated then circuit still operates correctly by increasing clock period. If hold is violated then circuit will not function correctly.

Some of the techniques used for fixing timing violations are listed below:

A. Logic Restructuring

Design timing can be improved by making modifications to the circuit topology. Some of them are as follows:

1) Signal Reordering

Critical signal along the logic path should be promoted in the logic to the latest position possible so

that data will pass through less logic stages. Signal reordering by promoting critical signal is promoted ahead by one NAND gate stage is illustrated in Figure 4.

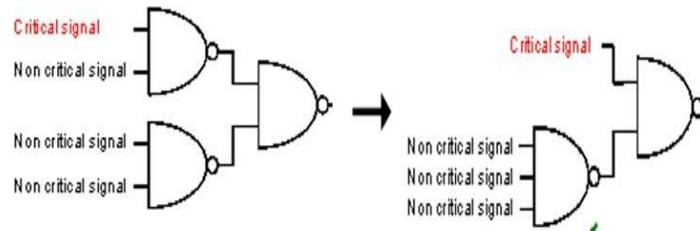


Figure 4: Signal reordering

Therefore all the non-critical computations should be completed before introducing critical signals in the design logic.

2) *Reduce load on critical path*

Non critical logic should be buffered to reduce the load on the critical signal path.

If there is a large load on the critical path, then it results in slow rise and fall times. Therefore load on the critical path is reduced by adding a small inverter for driving the non-critical path as clearly illustrated in Figure 5.

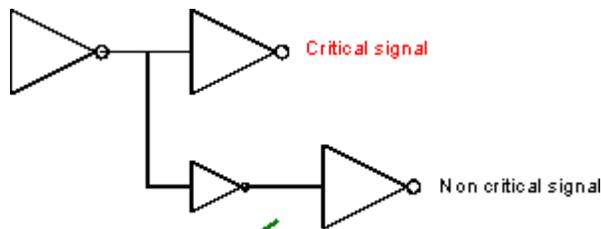
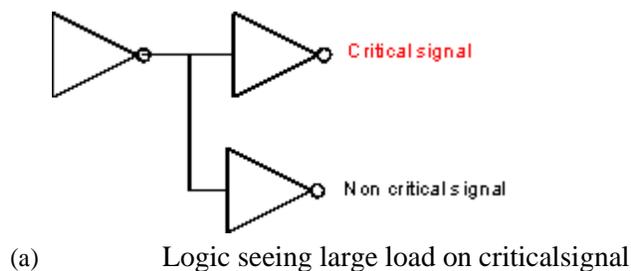


Figure 5: Load minimizing on critical path.

3) *Pin Swapping*

Critical signal should go into fastest input in stack. The fastest is the input to the device closest to the output node, either in P-stack or in N-stack.

Consider a NAND gate shown in Figure 6 where Critical signal at B nMos in N-stack is swapped with non-critical signal at AnMos. This is because transition through B nMos is slower than through A nMos. This is because of the body effect on AnMos when B nMos starts to conduct and also due to the charging of the internal capacitance.

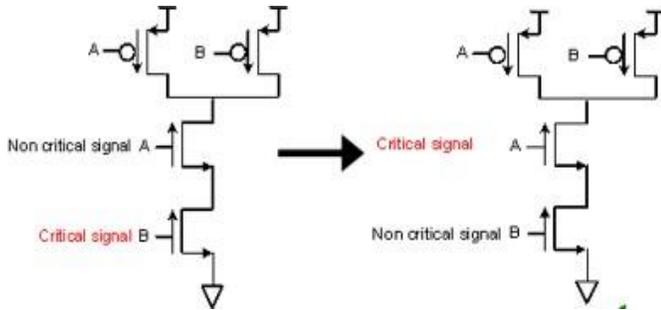
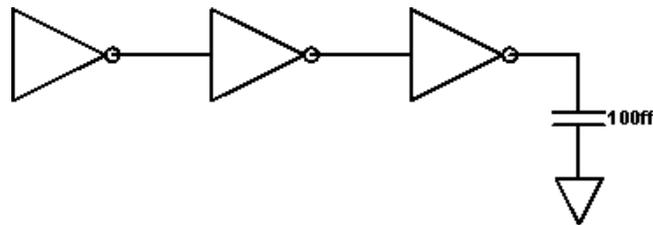


Figure 6: Critical signal swapping

4) Sizing InverterChain

It is better to drive a load capacitance through a chain of inverters. The delay characteristics is greatly influenced by the number of inverters in this chain.



In Figure 7 a chain of 3 inverters is driving a load of 100fF. Sizing is done starting from the last inverter backward. The rule of thumb is to use fanout of 4. Fanout is the ratio of gate output capacitance to the gate input capacitance. Also interconnect capacitance should not be ignored as it constitutes 20%-30% of total node capacitance

Figure 7: 3 stage Inverterchain

B. ClockTuning

Clock tuning is preferred when there are number of paths violating timing constraint in the design and sharing a common clock signal. Only clock library cells are used for clockpaths.

Clock Chopper

Clock choppers is used to delay one of the edges of clock signal. This is done with the aid of AND, OR or some other simple gates. Major problems are ill-defined pulse width and vulnerability to delay variations.

Figure 8 illustrates an OR clock chopper used for delaying falling edge of L1 clock. Since L1 is a low phase latch, the OR chopper delays the generating edge of the L1 clock. This improves the hold margin and degrades the setup margin by the amount in which the clock edge is delayed. Therefore there is a necessity to ensure that the path has enough positive setup slack to withstand the degradation. The L1 sampling clock edge remains unaffected.

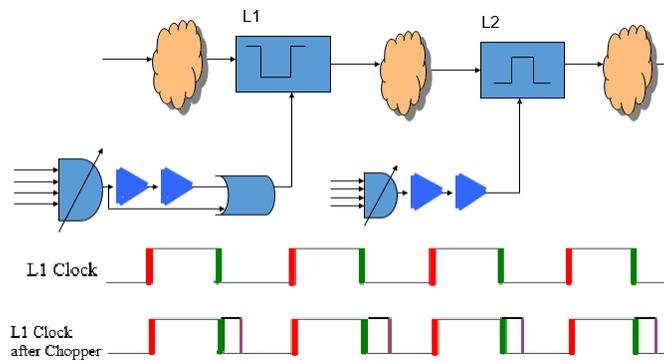


Figure 8: OR clock chopper with its corresponding waveforms

AND chopper can be used for delaying rising edge of the clock. Suppose if the OR chopper is replaced with an AND chopper in the L1 clock shown in Figure 8, the sampling clock edge will be pushed. This would improve setup slack and

degrade hold slack by the amount the sampling edge is delayed. Therefore a good positive hold slack number is required.

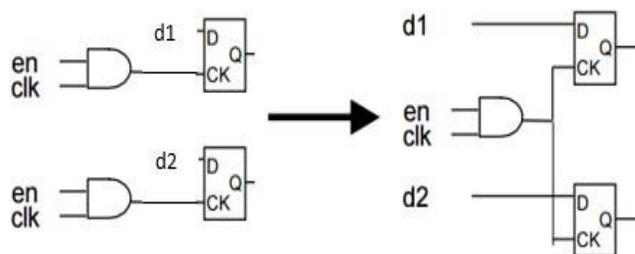
This illustrates that Setup and hold violations can be fixed by proper clock tuning. Similar to datapath clock tree is optimized by proper sizing of clock cells and trying out some topology changes. But choppers cannot be used for flip flop based designs since generating and sampling edges for flip flop are same.

Clock tuning is the last option followed during the timing optimization after trying all other implementations in datapath.

III. LOW POWER TECHNIQUES

This section describes techniques used to cut power consumption in datapath circuits. Leakage power is small and can be controlled by process parameters. Dynamic power is the major source of power dissipation caused due to charging and discharging of node capacitance for every voltage transition. All state nodes must be mapped and initialized with values. Dynamic power dissipation is calculated using the formula:

Figure 10: Clock cell merging



driver cells are placed close to each other to avoid negative impact on timing.

3) Clock Routing and Placement Optimization

The routing length of clock net should be minimized as much as possible since most of the design power is wasted in the clock network.

By means of avoiding tail routes in clocks as represented in Figure 11, there is significant reduction in power number. These cases are identified manually and routed.

$$C_{dyn} = AF * C * V_{dd}^2 * f_{clk}$$

Where AF is the number of toggles from 0 to 1 in a cycle, f_{clk} is the clock frequency, C is the total lumped capacitance and V_{dd} is the supply voltage. Since dynamic power contributes around 70% to the total power of the logic block. It is important to invest efforts first in dynamic power reduction.

Some of the low power transformations are discussed in this paper. They are:

1) Clock Gating

Clock gating is a technique of disabling clock toggles to the inactive logic with the help of an enable signal. Clock is the active power consumer and out of the total dynamic power 60% is related to the clock network.

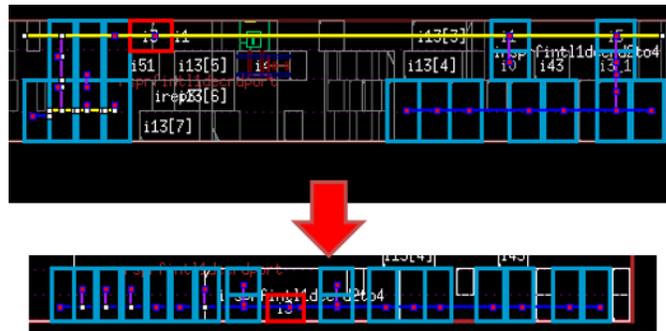


Figure11: Routing and placement optimization of clock network

4) Big Sequential

Reducing sequential will allow downsize of the driving clock tree. Designers can locate the oversized sequential cells having better setup slack from oversized sequential report.

Figure 12 illustrates downsizing a large sized latch and inserting a buffer at its output. A properly sized buffer cell allows driving the load capacitance



Figure 12: Downsizing big sequential

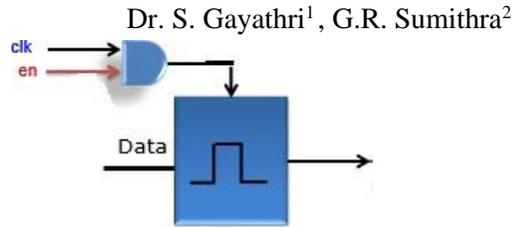


Figure 9: Clock Gating

Clock gating using enabled clock driver is shown in Figure

9. Activity Factor (AF) of clock nodes is twice that of data nodes as clocks toggle twice per cycle. This scheme is implemented to reduce AF of internal nodes but poses circuit timing challenges due to high frequency of operation.

2) Clock Merge

Clock drivers which has same input connectivity will be merged into single clock driver as shown in Figure 10. This is to increase the sequentials connected to the same driver. Merging Clock driver helps in reducing the load capacitance on the high activity factor clock net. It is necessary that the

5) Dual Latch Conversion

For latches with same clock share clock drivers and routing to reduce clock capacitance.

A dual latch has 2 data inputs, one clock input and 2 data outputs as shown in Figure 13. The two latches used for conversion are required to be present at close proximity.

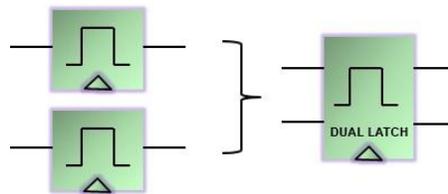


Figure 13: Dual Latch conversion

6) Local Clock Buffer Split

Clock is enabled as early as possible for optimal active power reduction. Clock gating reduces clock switching factor and therefore it should be moved early in the clock stage.

In Figure 14 AND clock gate is split into a NAND clock gate and an inverter. NAND clock gate is moved closer to its clock driver to reduce length of high AF clock net. An inverter is inserted to the low AF clock net at a distance based on the driving need

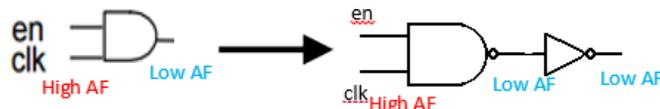


Figure 14: Clock driver split

IV. EXPERIMENTAL RESULTS

All the techniques described in the paper for removing timing violations and meeting power targets are validated using industrial datapath circuit and industrial sign-off tools. Better delays is obtained by enlarging elements. Availability of what-if analysis in the timing tool helps in quickly knowing the

change in delay and slope for each element when we enlarge or shrink them in the circuit.

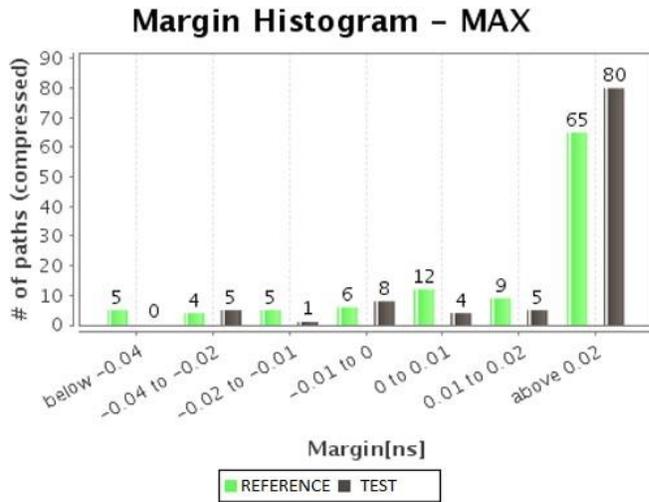


Figure 15: Setup Margin Histogram Methodologies discussed in this paper for achieving timing constraints is implemented in one of the datapath block of microprocessor for meeting setup and hold requirements of the block. Max histogram with paths compressed and paths uncompressed for HighV corner is shown in Figure 15 and Figure 17 respectively. It clearly depicts the topmost worst negative paths being reduced in the test run compared with the initial reference. Similarly min histogram with paths compressed and paths uncompressed for nominal corner in Figure 16 and Figure 18 respectively. But the topmost worst negative min paths are degraded even further. This is due to changes made in the design for fixing setup margin. The overall block timing summary data is reported in Table 1. It provides details of Worst Negative Slack (WNS), Total Negative Slack (TNS), and number of negative paths for both internal and external paths for min-max analysis.

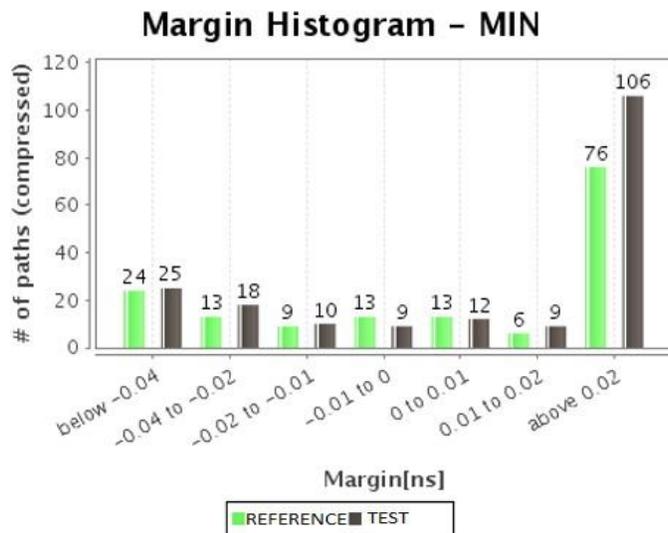


Figure 16: Hold Margin Histogram Table 1: Datapath block timingsummary

Summary & Compare					
			REFERENCE	TEST	
MAX / Setup	TNS (ps)	External	-0.428	-0.199	●
		Internal	0.000	0.000	●
	WNS (ps)	External	-0.081	-0.034	●
		Internal	NA	NA	●
	#Negative Paths	External	23	14	●
		Internal	0	0	●
MIN / HOLD	TNS (ps)	External	-2.200	-2.222	●
		Internal	-0.122	-0.116	●
	WNS (ps)	External	-0.152	-0.096	●
		Internal	-0.026	-0.029	●
	#Negative Paths	External	46	51	●
		Internal	13	11	●

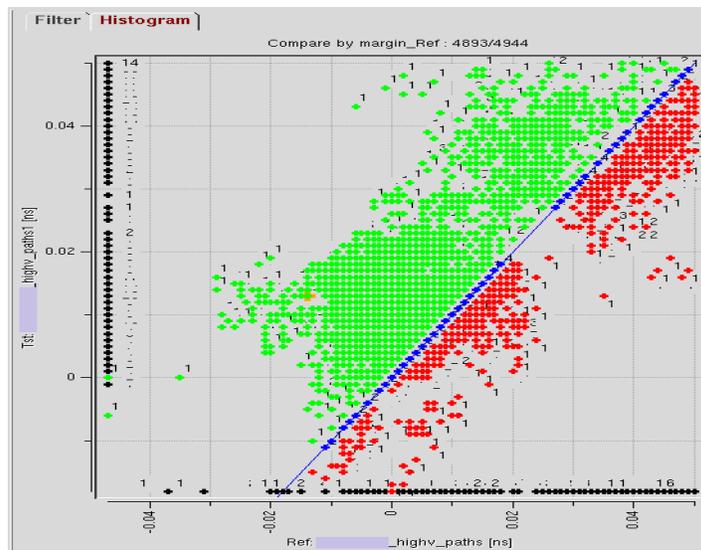


Figure 17: Setup Margin Histogram in

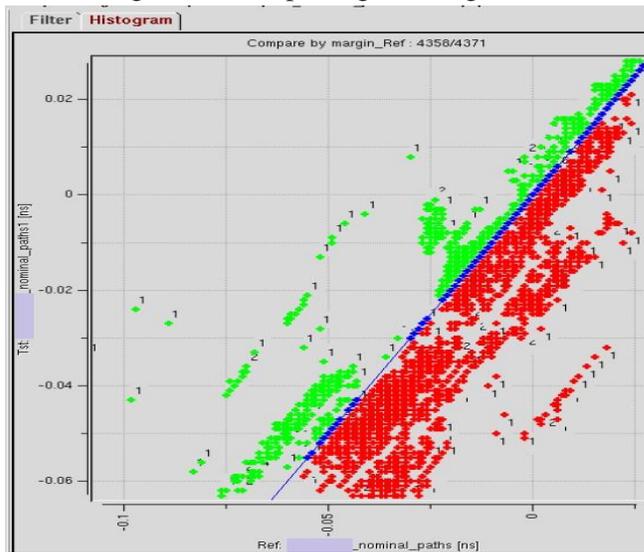


Figure 18: Hold Margin Histogram in Nominal Corner Power tools estimate power consumption by considering activity factor and node capacitance into account. Power consumed by the design is calculated for different real application scenarios. The AF and Static Probability (SP) for nets will be available in the reports along with their load capacitance values. These values differ from one power test to other. Table 2 contains block dynamic power (Cdyn) values, power budget value and their difference percentage for different workload conditions. QM is the quality metric of the power tool whose value depends on the percentage difference of activity factor between RTL and netlist, number of nodes initialized with 0 and 1, capacitance extraction quality etc.

Table 2: Comparison of Power Results

POWER TEST	BUDGET (pf)	REFERENCE RUN		TEST RUN		
		BLOCK CDYN (pf)	DIFF %	BLOCK CDYN (pf)	DIFF %	QM
Workload 1	0.9257	1.1153	20.48	1.0109	9.2	97.5
Workload 2	0.7892	0.9863	24.97	0.8815	11.69	89.32
Workload 3	3.4968	4.1222	17.88	3.8046	8.8	99.25

The power features are selectively implemented based on the net AF and capacitance values and comparison is given in Table 2. QM must remain unchanged.

V. CONCLUSION AND FUTURE WORK Designing a low power and high performance datapath

logic block poses significant challenges. Since datapath design is the most critical part in the VLSI design, modern techniques need to be followed for the datapath performance requirements. In this paper a brief discussion on the datapath design closure challenges is provided. The various techniques used for timing closure and for major power savings is

explained and their implementation results is included. Tradeoffs for low power and performance are carefully handled. Most of the techniques discussed comes under logic minimization which is good for timing as well as power.

As a future work reducing the noise introduced in the circuit is important. Therefore several circuit implementations and logic reimplementation like avoiding the usage of pass gate and high fan in gate on the block boundary, provide spacing or add buffer for reducing Cross Capacitance (Xcap) can be tried to minimize noise in the design. Further max constraints can be achieved without compromising minconstraints.

REFERENCES

[1] T.T. Ye, S. Chaudhuri, F. Huang, H. Savojand G. De Micheli, "Physical synthesis for ASIC datapath circuits," presented in IEEE International Symposium on Circuits and Systems, Phoenix-Scottsdale, AZ, USA, 26- 29 May2002

[2] Shu-Min Li, Yih-HuaiCherng and Yao-Wen Chang, "Noise-aware buffer planning for interconnect-driven floorplanning," presented in Proceedings of the ASP-DAC Asia and South Pacific Design Automation Conference, Kitakyushu, Japan, 24-24 Jan2003

- [3] J. Bhasker and Rakesh Chadha, "Static Timing Analysis for Nanometer Designs," 1 st ed., Springer US 2009, pp.15-39.
- [4] Emre Salman, Ali Dasdan, Feroze Taraporevala, Kayhan Kucukcakar and Eby G. Friedman, "Exploiting Setup-Hold-Time Interdependence in Static Timing Analysis," published in IEEE Transactions On Computer- Aided Design Of Integrated Circuits And Systems, Vol. 26, No. 6, June 2007, pp. 1114-1125.
- [5] M. Mahmood, M. Chandrasekhar, B. Sharma and A. Ginetti, "A method for timing driven datapath synthesis," presented in Proceedings of Eighth International Application Specific Integrated Circuits Conference, Austin, TX, USA, 18- 22 Sept 1995.
- [6] Liu Yang, Sheqin Dong, Yuchun Ma and Xianlong Hong, "Interconnect Power Optimization Based on Timing Analysis," presented in IEEE Computer Society Annual Symposium on VLSI, Porto Alegre, Brazil, 9- 11 March 2007.
- [7] S. Shah, P. Gupta and A. Kahng, "Standard cell library optimization for leakage reduction," presented in 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 24-28 July 2006.
- [8] S. Posluszny, N. Aoki, D. Boerstler, P. Coulman, S. Dhong, B. Flachs, P. Hofstee, N. Kojima, O. Kwon, K. Lee, D. Meltzer, K. Nowka, J. Park, J. Peter, J. Silberman, O. Takahashi and P. Villarrubial, "Timing closure by design," a high frequency microprocessor design methodology, Proceedings 37th Design Automation Conference, Los Angeles, CA, USA, 5-9 June 2000.
- [9] Padmini G. Kaushik, Sanjay M. Gulhane and Athar Ravish Khan, "Dynamic Power Reduction of Digital Circuits by Clock Gating," International Journal of Advancements in Technology, Vol. 4 No.1, March 2013, pp.79-88
- [10] S. Askar and M. Ciesielski, "Analytical approach to custom datapath design," presented in IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 7-11 Nov. 1999.
- [11] Shweta Shah, N. Mansouri and A. Nunez-Aldana, "Pre-Layout Estimation of Interconnect Lengths for Digital Integrated Circuits," 16th International Conference on Electronics, Communications and Computers, 1 March-27 Feb. 2006, Puebla, Mexico.