IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 11, Issue 3, Ser. I (May. - June. 2021), PP 18-25
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Timing Optimization of Functional Unit Block in High Speed Core

V.Anandi
Associate Professor,
Department of Electronics and Communication
M.S. Ramaiah Institute of Technology
Banglore, India

M.Ramesh
Professor, (Emeritus)
Alliance University
Bangalore, Karnataka

Abstract— *In today's high performance advanced VLSI circuit design including most semicustom design gives efficient utilization of circuit resources with better productivity offering a good layout density and high performances with optimum resources in submicron designs. The main idea is to divide the processor into small partitions called Functional Unit Block (FUB) based on the different functionalities of the processor like Execution, Memory, Fetch, Decode and Out of order. The major focus of this work is to optimize the FUB in terms of timing, power and area. With the enhancement of the existing design technology and optimization methods, new design technology called datapath design is used to deliver a quality design and this work also proposes various logical optimization, RTL changes, placement, routing optimization techniques to meet the specifications in terms of power, timing. This paper discusses the design and implementation of timing convergence techniques to reduce the paths with negative margin to fix set up and hold time violations in semi-custom datapath design of FUB.*

Keywords—*datapath;synthesis;optimization;timing;clock tuning;*

Date of Submission: 28-06-2021 Date of Acceptance: 12-07-2021

I. Introduction

The continuous growth in the VLSI industry is ending up in delivering high-performance products and at the same time, it involves more challenges in the design. In the design of advanced microprocessors, different IP blocks are combined with the core and are fabricated on a single chip. The microprocessor involves both the hardware as well software components, hardware components are the memory, interface buses, IP blocks, and software components involves operating systems, library functions for the standard cells, configuration files.
The core design approach is based on the concept of divide and conquer. The entire core is divided into different clusters and the clusters are divided into sections and the sections are further divided into different Functional Unit Blocks. Sections are branched out into different units and finally the leaf cell in the hierarchy is called Functional Unit Blocks (FUB). FUBs in a full chip can include multiplier, adder, divider, register files, Latch repeaters, and receivers. The circuit design and implementation are done at the FUB level followed by integration at the section and full Chip level.

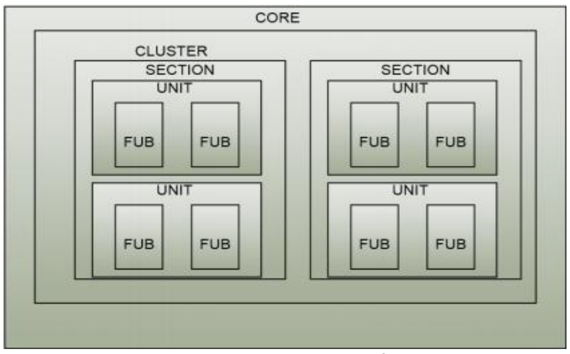

Figure 1: Hierarchy of core

In the advanced process node, it is difficult to achieve high performances with additional complexities. This paper proposes a semi-custom data path design synthesis implementation of the FUB and performs timing convergence techniques to reduce the number of paths with set up and hold time violations. As in [1] each technology process node comes up with a set of standard cells of different combination and sizes from where the digital designer can choose standard cells based on the specification. Suitable choice of cell size will help in timing optimization. Upsizing and downsizing can be helpful in timing convergence.

The paper is organised into following sections: Section II discusses the existing design approaches, III about synthesis design methods, IV introduces datapath design synthesis concepts

II. Design Approach

The circuit implementation of FUB can be based on either h full custom or semi – custom design approach. Based on the functionality of the FUB, the design approach is decided. Various circuit design approaches are listed in Fig 2.

A. Full Custom Design:

In the Full Custom design approach all the logic cells, layout specifications are designed for a particular application. If the existing library cells are not suitable for the design in terms of speed, area, power then this design approach can be used to build the library cells from scratch. The disadvantage of the full custom design approach is that increase in design time, complexity, non-recurring engineering cost also increases.

B. Semi-Custom Design:

The semi-custom design approach uses a customized mask layer, predesigned logic cells which are also called standard cells. In a semi-custom design, the area of standard cells is built as row cells. The major advantage of the semi-custom design is reduced risk in designing since the predesigned cells might be used in various other previous implementations and also it is cost-effective due to design density.

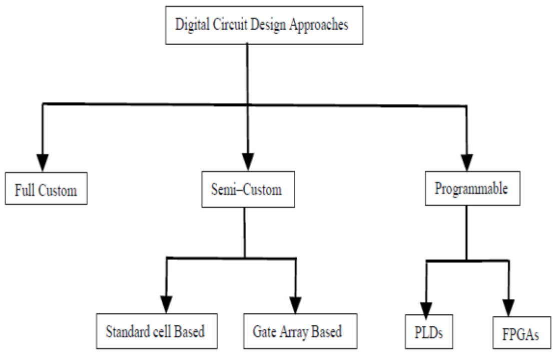

Figure 2: Design Approaches

This work is implemented as Datapath design, and the Functional Unit Block (FUB) is designed based on the RTL to design constraints of 14nm technology using characterized standard cells and simulation tools. As FUBs are the basic building blocks of the core those must be efficiently optimized in timing, power, and quality. The important factor that is affecting the performance of VLSI design is the interconnect delay. Techniques, which include buffer insertion and wire-sizing have been proven to be very effective in reducing interconnect delay. [2]

III. Synthesis Design Methodologies

Basically any design can be synthesized using two different methodologies, whose features are listed below:

A. RTL to Layout Synthesis (RLS)

This is a tool based design synthesis methodology. It can be done using Synopsys Design Compiler (DC) and physical designing is carried out using IC Compiler. It is most preferably used for controller FUBs. Any frequent changes in the control signals can be observed in the design easily and quickly. Control over design is quite less.

A. Datapath

The design is synthesized manually by the designer and **c**omplex designs can be synthesized using this design methodology to have more control over the design. The physical designing step of Place and Route is also done manually. It provides the flexibility for hierarchical design methodology.

IV. Datapath Design Synthesis

In a full chip floorplan, the datapath designs are divided into dedicated FUBs and the designer has to implement the design manually and place the standard cells in a bit slice manner. But automation tools synthesize the design without the knowledge of the placement of the cells which causes incorrect estimation of interconnect timing. Hence the datapath design synthesis is considered to be the most sub-optimal. Datapath designing and synthesis includes designing of FUBs according to the given RTL using available standard cells. This step is followed by Formal Verification of the implemented gate-level design with the given RTL code. Also checking of set up and hold margins on critical paths leading to timing optimization of the design are carried out. The paper [3] came up with a methodology that modifies the traditional ASIC design flow to integrate both logic synthesis and physical design as a single flow and the main objective by doing so is to reduce cost function that depends on timing constraints, routing constraints, area constraints of the design.

V. Timing Analysis

Timing analysis is a methodology wherein the digital designs are analysed to verify whether it is meeting timing constraints imposed by interfaces or not. Since the design is latch-based and each latch in the design has timing parameters like setup and hold time and these timing constraints must be met to ensure the proper functioning of the design, which is the main aim of the timing analysis.

A. Static timing analysis (STA):

STA analyses each path in the design based on cell delay, net delay, net RC values, metal usage, cell placement which are compared against maximum and minimum timing constraint values. Cell delays are modelled by the technology used in the library design and also depend upon the transition, load that it is driving it. Net delays are modelled based on RC extracted values, net length, and also metal layer used. Each timing path in the design constitutes the combination of cells, latches, mux, and flip-flops and the interconnection between them. To meet the desired frequency of operation, data in each timing path must be processed within the specified clock period.

Behavioral model provides the timing (valid time from the output) information within a FUB. Environmental model describes the timing specification external to the FUB and defines the required time for each output pin in the FUB as shown in Fig 3.

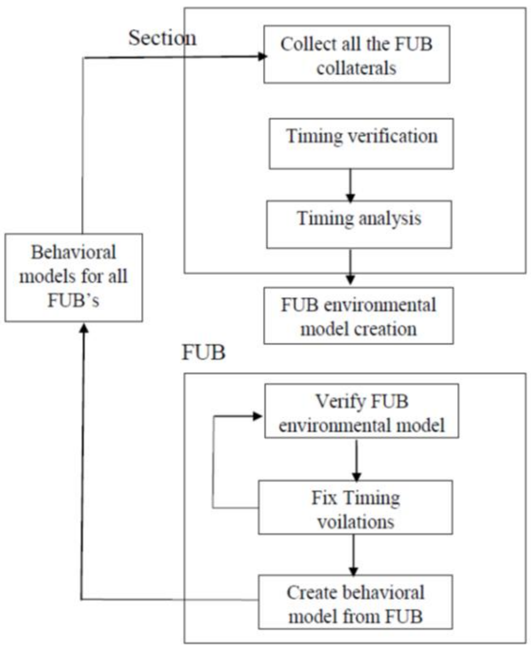
Section
Collect all the FUB collaterals
Timing verification
Timing analysis
FUB environmental model creation
Behavioral models for all FUB's
FUB
Verify FUB environmental model
Fix Timing voilations
Create behavioral model from FUB

Figure 3. STA Flow diagram

VI. Static Timing Analysis

Inputs to STA requires parameters including resistive and capacitive load, signal slopes, metal layer type, metal width, driver strength, actual required time of the pins, expected arrival time of the pins, and clock cycle time to analyze the timing in the path. Set of outputs obtained from STA analysis includes a list of all the paths with setup and hold violations along with path margins as well as path length, list of each cell slope, delay and size of each cell in the path, net information with RC extraction and also the net quality. The capacitive load that each cell is driving will also be reported with the number of fan-outs driven and behavioural timing model will be provided with valid timing requirement for input and output pins.

A. Timing paths

STA categorizes the timing paths as internal (within FUB) and external (outside of the FUB) paths as shown in figure 4. If Entire path is within the FUB and all the elements in the path are present in the FUB it is categorised as internal path. Delay paths can be classified based on the elements present as
Latch – Delay – Latch (L – D - L Path), Flip flop – Delay – Flip flop (FF – D – F Path) and Flip flop – Delay – Latch (FF – D – L Path). External paths contain a set of elements within the FUB and certain set of elements external to the FUB. There are different types based on the elements present similar to Latch – Delay – External (L – D – E Paths) , Flip flop – Delay – External (FF – D – E Paths), External – Delay – External (E – D – E Paths) . They can also be classified based on the transparent latches present as External – Transparency – External (E – T – E Paths) and Latch – Transparency – External (L –T – E Paths). Fig, 4 illustrates such of those timing paths which may be encountered in a high speed core.

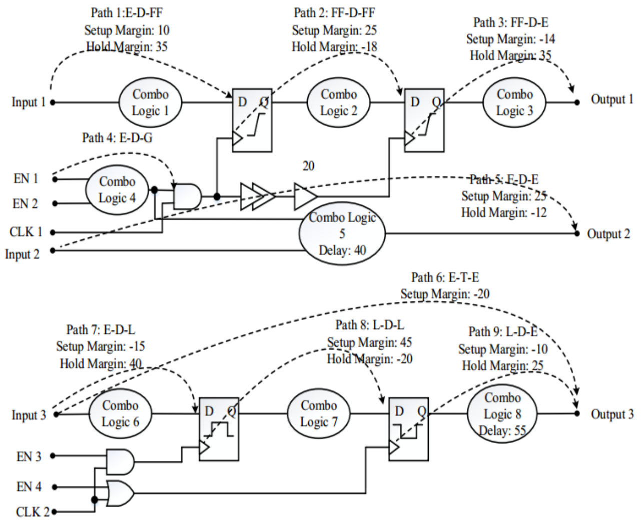

Figure 4. Example showing all timing paths

B. *Timing violations*

Violations based on timing are classified as setup time violation where an inaccurate data is captured since the data is not stable before the active edge of the clock and hold time violation is a situation where the data is not stable for a minimum amount of time after the active edge of clock. The required timing window is illustrated in the Fig. 5

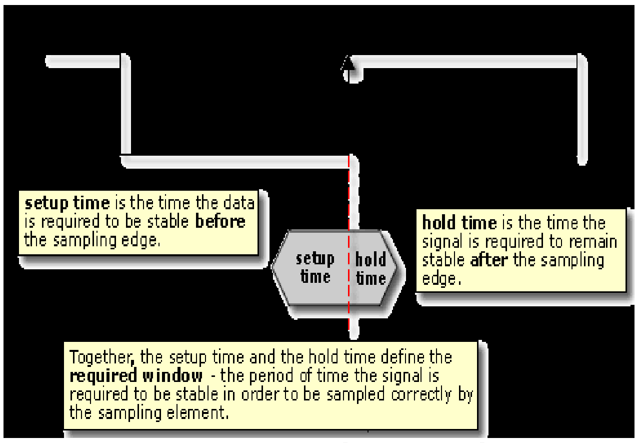
setup time is the time the data is required to be stable **before** the sampling edge.
setup time
hold time
hold time is the time the signal is required to remain stable **after** the sampling edge.
Together, the setup time and the hold time define the **required window** - the period of time the signal is required to be stable in order to be sampled correctly by the sampling element.

Figure 5. Required timing window

C. *Violation fixes*

Cell upsizing based on its load, optimised cell placement and logic optimisation are a few of the most commonly used setup time violation fixes. Timing violations can also be fixed by metal layer upgradation based on resistance and capacitance values and load splitting. Clock tuning is considered a major timing fixation. Clock tuning is a method to alter the clock arrival time at the generating or sampling edge to meet the timing constraints. Most commonly used hold time violation fixes are downsizing the cells, introduction of delay elements like buffer or inverter in the path and metal layer downgrading will have impact on the resistance and capacitance to improve hold time.

VII. Implementation

A. *Cell upsize*

If there are any setup time violations from generating edge to sampling edge as in the Fig. 6, upsizing the standard cell in the path can be useful, which increases the drive strength of the cell and reduces the path delay. But an important factor that needs to be considered is that the hold time on that particular cell is sufficiently positive enough or else it will cause hold tome violation in the path.

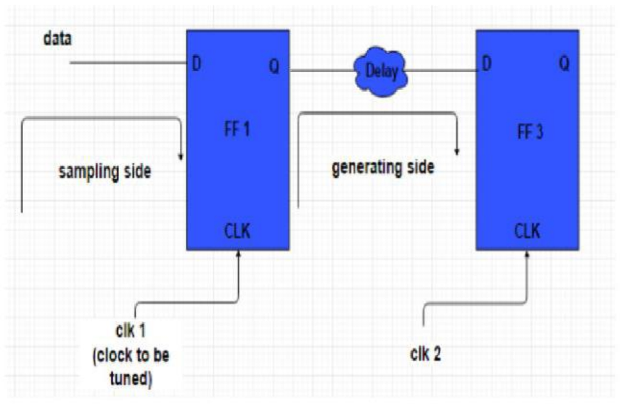

Fig 6. Setup time violations and fixes

If the design is hierarchical in nature the hold time must be verified across all the hierarchies before upsizing of the cell. Also the constraint on power consumption must be looked into, as upsizing a cell increases the drive strength in turn increasing drive current impacting the power consumption. One more factor to consider is the area and if the area is congested care should be taken in designing the layout else it will create overlaps in the layout and lead to many quality issues in the backend design of FUB.

B. Clock tuning

To fix setup time violation either the generating clock can be made faster or sampling clock can be delayed but the factor that needs to be considered is that the same clock might be driving many other sequential elements in the design and hence its cumulative effect to be taken into consideration. Hence designer must be extremely careful. Clock tuning can be accomplished by upsizing or downsizing the cells in the clock path. Also by adding or removing the clock buffers in the path, termed as clock push/pull has an effect on fixing violations.

As shown in Fig 6 clk1 is pushed, data generation is slower from FF1 and the setup margin on generating edge side of FF1 degrades, at the same time the sampling edge side of FF1 setup margin will be improved as data arrival time remains same but the clock arrival will be delayed thus, setup margin improves. Table 1 gives the brief details.

TABLE 1

Method	**Clock edge**	**Set up time**	**Hold time**
Clock push	Generating / Sampling	Degrades	Improves
Clock pull	Sampling / Generating	Improves	Degrades

C. Choice of standard cells

Each cell is characterized to have varied functionality to achieve better accuracy in post silicon with specified voltage, frequency, load and slope with worst case rise and fall delay and is very useful for the timing analysis as well. Standard cells are classified as low leakage cells as slow VT cells, skewed cells and tapered cells and have different characterisation. Tapered cells are the special form of cells where in the width of the stacked transistors are different to decrease the delay from input pin to output pin. Consider in a NAND gate, in Fig 7 the stacked NMOS transistors width will be varied, where in latest arrival signal is hooked up with wider width stacked transistor to improve the path margin. These cells are mainly used in the case where one of the input signal is critical and other is certainly meeting the timing constraint.

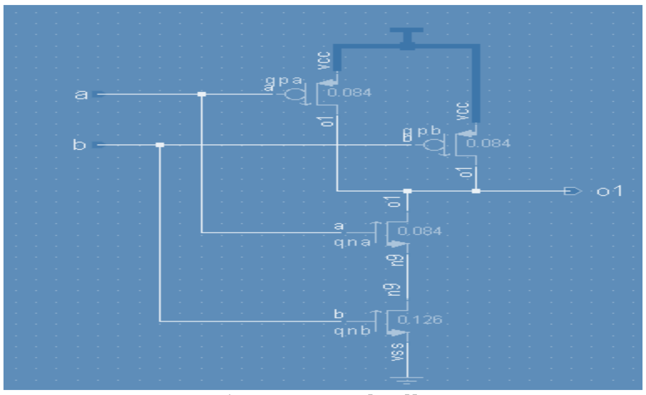

Fig 7. Tapered cells

D. Routing optimization

Improper routing methodologies including scenic routing (routing with lot of jogs) will result in unnecessary delay in the path. Designer can optimize the routing by choosing appropriate metal layers, which will result in reduction in resistance and capacitance numbers of the net and improves overall path delay. To establish connection between device 1 and device 2, the tool has routed using two metal tracks of each of the metals 3 and 4 which is unnecessarily increasing the path length and delay. This is illustrated in Fig 8. It is re-routed using one metal track of each of the metal layers 3 and 4, which reduces the number of vias there by the reducing resistance and capacitance. Hence overall net delay and path delay also comes down.

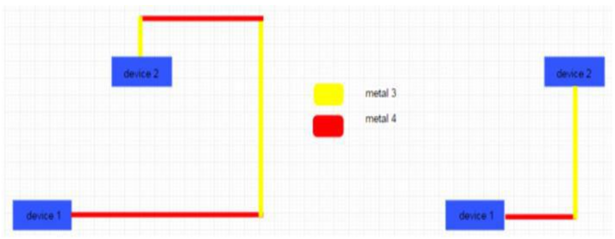

Fig 8. Routing optimisation

VIII. Timing Results

Based on the timing constraints, static timing analysis is done which gives the reports for Worst Negative Slack (WNS) and Total Negative Slack (TNS) of both setup and hold times. The results are represented below. Timing results histogram before applying the convergence methodologies is illustrated in Fig 9, which is having setup time violation with WNS of -46ns and Y-axis gives the number path count for each margin on the X-axis.

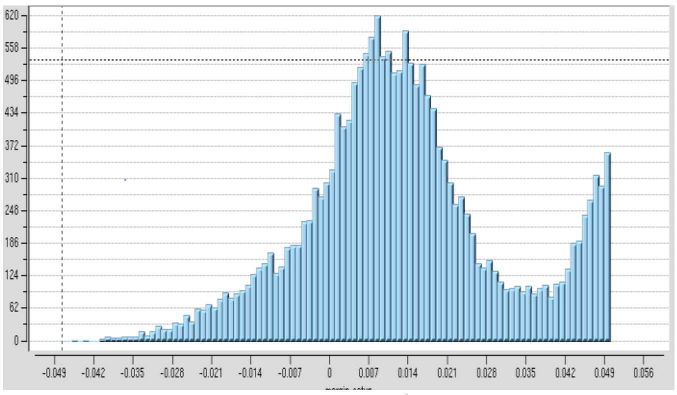

Fig 9. Set up time results before convergence

Timing histogram after applying the convergence methodologies shows setup time violation with WNS of -16ns, the reduction in WNS after applying timing convergence methodologies in Fig 10.

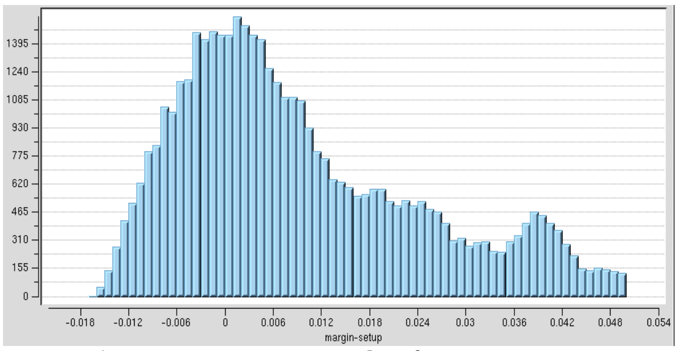

Fig 10. Setup time results after convergence

Timing histogram results before applying the convergence methodologies withhold time violation with WNS of -36ns and is illustrated in Fig 11.

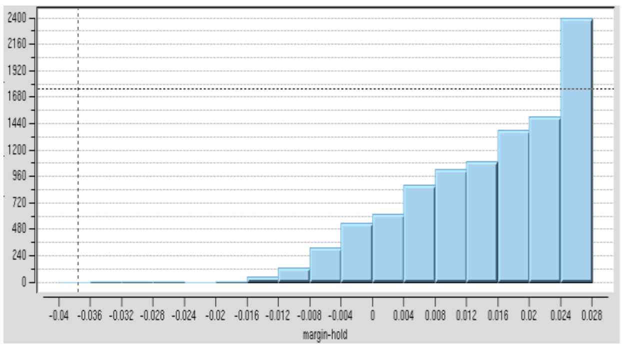

Fig 11. Hold time results before convergence

Timing histogram results after applying the convergence with hold time violation WNS of 0ns. Fig 12 shows the reduction in WNS after applying timing convergence methodologies.

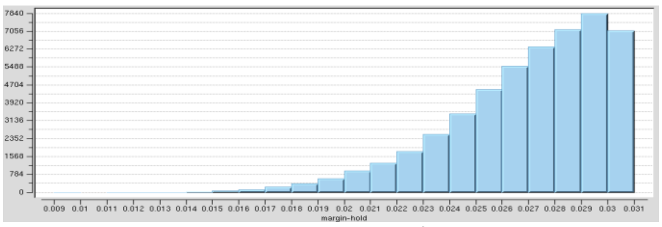

Fig 12. Hold time results after convergence

IX. Conclusion

In this paper implementation of digital circuits in accordance with back end design flow is discussed. All the timing optimization techniques were discussed in detail and results were tabulated. If timing optimization is done inefficiently and some paths have less setup/hold margins, then power optimization can create negative margins on those paths. Thus, the paths that have to be optimized with power must have good margins. In future, more opportunities for power optimization can be found from same designs by converging the remaining negative margin paths. Also, the multi-bit Flip flop technique to merge two flip flops can be even used to merge multiple latches and going further dual latches or dual flip flops can be further converted to quad latches/FFs to reduce the overall load on clock. Also one can conclude that the Datapath designs are the most structured ones and provides more flexibility and control to the designers and the resource utilization is better in these kind of designs and is feasible.

References

[1]. Linumon Thomas and Kiran V, "Timing Convergence Techniques in Digital VLSI Designs"

[2]. Mohamed Khalil Hani and Nasir Shaikh-Husin, "Simultaneous Routing and Buffer Insertion Algorithm for Interconnect Delay Optimization in VLSI Layout Design"

[3]. Narendra Shenoy, Mahesh Iyer, Robert Damiano, Kevin Harer and Hi-Keung Ma ,"A Robust Solution to the Timing Convergence Problem in High-Performance Design," ACM Transactions on Design Automation of Electronic Systems, vol. 11, no. 2, pp. 306-324, April 2006

[4]. http://www.intelpedia.intel.com