

Semi-Custom Design of Functional Unit Block Using Data Path Methodology in Data Cache Unit

Aarti Patel
SVNIT, Surat, Gujarat, India

Prashant K.Shah
Associate Professor, ECED-SVNIT

Abstract

Chip design commences with the conception of an idea dictated by the market. These ideas are then translated into architectural and electrical specifications. The architectural specifications define the functionality and partitioning of the chip into several manageable blocks. A Functional Unit Block (FUB) is a small part of the micro-processor that is characterized by an RTL code. This project discusses the design of a structured data path functional unit block of a microprocessor. It explains the total flow of the backend design starting from implementing the schematic circuit from given RTL code using standard library cells to final layout. The design will be fulfilled by all the given constraints like operating frequency, timing violations, area, power, noise, reliability and circuit quality.

Keywords — RTL , Functional Unit block , Formal Equivalence Verification

I. INTRODUCTION

Cache data memory is used to store the data from main memory as shown in the figure 1. For example if we are designing 4 ways of such 256 byte cache data memory to design 4 way set associative cache memory of 1KB. If address requested by microprocessor is present in cache tag memory then data related to hit address is given to microprocessor packaging and cooling costs but also causes reliability problems, since the mean time to failure.

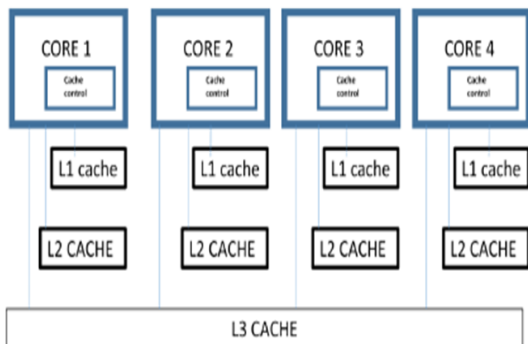


Figure 1 cache organization

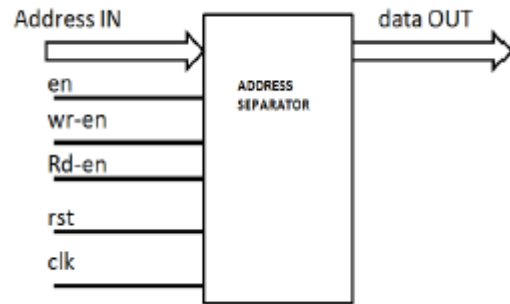


Figure 2 : DATA CACHE UNIT

II. FORMAL EQUIVALENCE VERIFICATION

Any backend implementation without its logical correlation with the RTL model is invalid. So, this step is the first stage after the schematic implementation from the RTL code in order to find logical bugs in the implemented design to avoid silicon bug at the later stage. It is always suggestable to ensure that the schematic logic is equal to RTL before starting the static timing analysis and other optimization techniques.

Formally, RTL is the architectural specifications from design architect and schematic is output from logic synthesis. Here comparison of the RTL (Reference) language with the Netlist generated from synthesis stage is done. This check is purely mathematical and is not based on any simulation. So it does not depend on the test vectors applied. Since the results are from mathematical proofs, it is 100 percent accurate and reliable.

Different stages of FEV:

1. Netlist extraction
2. Mapping
3. Verification

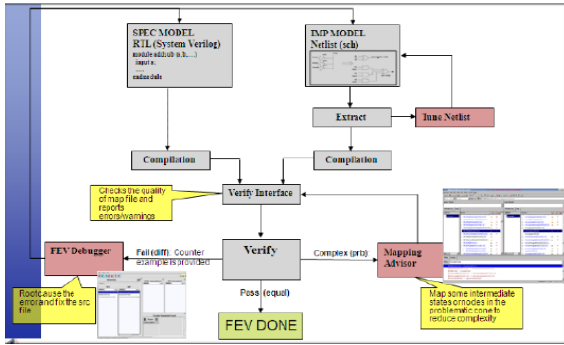


Figure 3 Netlist Extraction

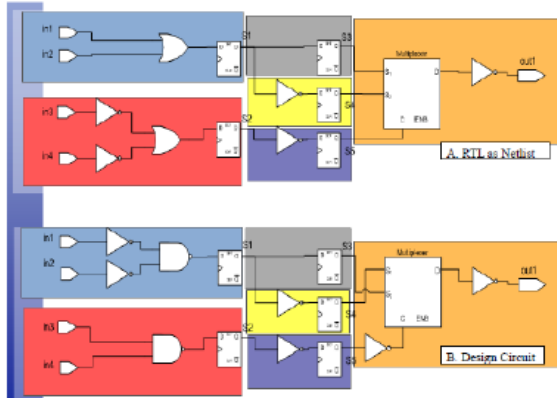


Figure 4 Mapping

VERIFICATION

FEV verification stage follows Divide and Conquer framework to perform the verification.

As shown in the figure the entire implementation is divided into small cones. The vertex of each cone is called as cut point. FEV will perform equivalence verification check for each and every cut points. FEV follows certain criteria to decide what all points in the design can be modeled as a cut point. Like, each output node of a sequential (Flop or a Latch) is modeled as a cut-point. In few cases, where the design is much complex, the FEV cannot be able to decide on the nodes which can be modeled as a cut point. In such cases, the DE has a flexibility to add cut points manually by inserting a mapping in the map file. In general each and every node present in the map file is considered as a cut point. In few cases there may be a requirement like not to use one particular node modeled by the tool as a cut point. In such cases, DE can use a switch in the same map file saying never use this node as a cut point. If there is any analog block present inside the design, then an additional file is required for FEV as an input which describes the logical relation between the input and

output nets of the analog block. Once the FEV is clean Design Engineer (DE) will proceed for the timing verification. In few cases, for meeting the timing requirements there may be a requirement of addition or removal of few cells. In case of major modifications in the FUB, it is required for a DE to rerun the FEV and ensure that the SCH & RTL are logically equal with all the edits in SCH.

III. TIMING ANALYSIS

Timing analysis is the crucial part of any digital system analysis. With the growing technology, Speed is one of the basic parameters which judge the performance of any digital system. Timing analysis provides a methodical base for the analysis of a digital circuit to determine if the timing constraints imposed by components or interfaces are met or not. Since each device input can have many sources whose timing can vary with circuit operation mode, the timing analysis can be very complicated and time consuming.

The backend design of the microprocessor core involves the usage of latch based design in order to improve the performance.

Setup violations can be defined as the difference between actual time the signal is at data pin of sampling cell and required time the signal has to be at data pin of sampling cell. Setup violation limits the circuit frequency. Hold violations is defined as the difference between actual times the signal is held stable at data pin of sampling cell and required time the signal has to be held stable at data pin of sampling element. Hold violation is independent of clock frequency.

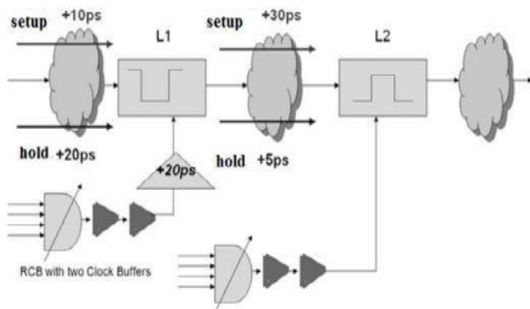
A. Methodology For Timing Fixes

1. Upsizing cells (increases area and input capacitance)
2. Load splitting
3. Reducing Fan-out (inserting buffer for non-critical paths)
4. Changing placement of cells (minimizes interconnect delay)
5. Replacing Flip-flops by two latches
6. Insertion of min delay buffers for
7. fixing of hold violation

8. Clock tuning.- In some cases with timing violation insertion of buffer or upsizing the cells will not solve the issue. In such case, fixing of timing is with clock tuning. Delaying the rising edge of the clock is going to fix the setup violation and advancing the clock will fix the Hold violation. This moving of the clock is called as clock tuning. However, delaying the clock to fix the setup time on a particular path is going to worsen the setup time on the paths, which are immediately next to the latch.

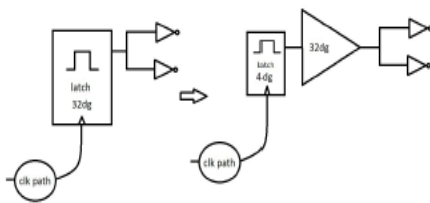
Clock tuning can be done on two ways:-

- Adding or removing buffers in the clock path:- In this method, we just add a clock buffer in the clock path or remove a clock buffer from the clock path based on the requirement of the set up or hold violation. As an example, the figure shows adding a clock buffer in the clock path of Latch 1.

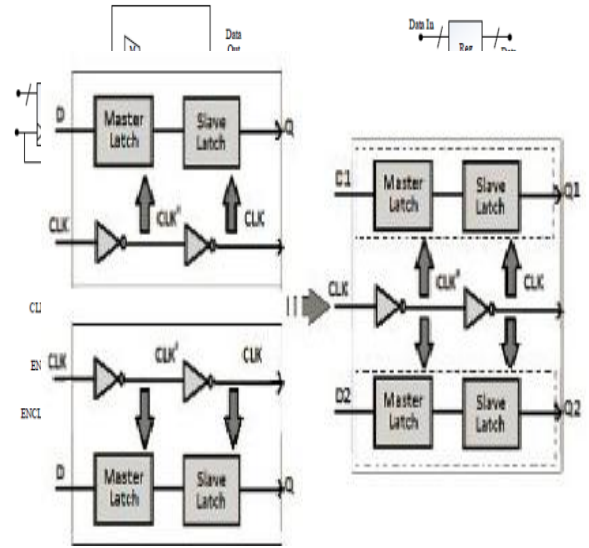


IV. POWER ANALYSIS

A. SEQUENTIAL DOWNSIZING



B. LATCH BASED CLOCK GATING

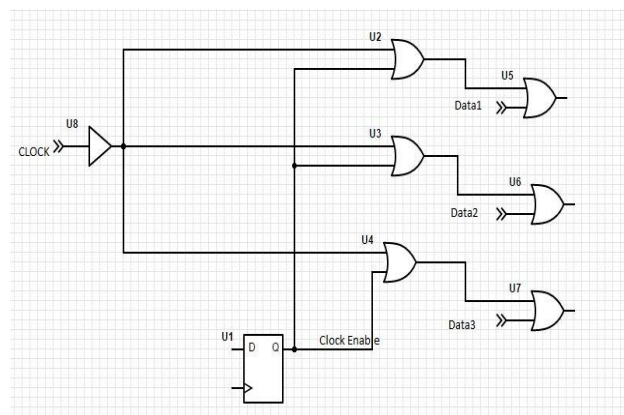


C. MULTI-BIT FLIPFLOP

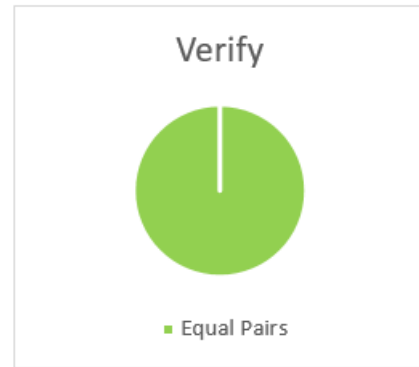
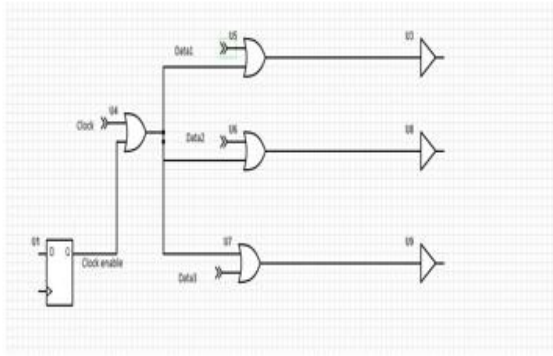
D. CLOCK OPTIMIZATION

Clock nets have high Activity factor. So reducing the load of the clock network reduces the dynamic power as $C_{effective}$ reduces. Clock optimization refers to general methodologies of proper formation of clock tree. Some of the examples are shown below.

CLOCK TREE WITH HIGHER LOAD



CLOCK TREE WITH LESS LOAD



V. RESULTS

[1] FUNCTIONAL EQUIVALENCE OF DESIGN

Parameter	Specification interface mapping	Implementation Interface Mapping
Mapped Input	549/549 (100%)	569/569(100%)
Mapped output	506/506 (100%)	533/533(100%)
Mapped States	6572/6648(98.9%)	6635/6711 (98.9%)

Verification Status

Power	Before Experiment(mW)	After Experiment (mW)	Percentage Improvement
Total Power	6.16251	5.73557	6.92%
Dynamic power	5.71625	5.31834	6.96%
Short circuit Power	0.26304	0.25736	2.16%
Leakage power	0.18322	0.15987	12.744%
Total Pairs		7271	
Equivalent Pairs		7271	

[2] POWER RESULTS FOR DATA PATH UNIT BLOCK

Capacitance	Before Experiment (pF)	After experiment (pF)	Percentage Improvement
Capacitance	48.13321	47.32214	1.6%
Gate Capacitance	10.66194	10.38576	2.59%
Diffusion Capacitance	3.86597	3.61736	6.4%
Interconnect Capacitance	9.26833	9.16173	1.15%
External Capacitance	23.31921	23.19421	0.53%
Effective Capacitance	5.7031	5.3006	7.05%

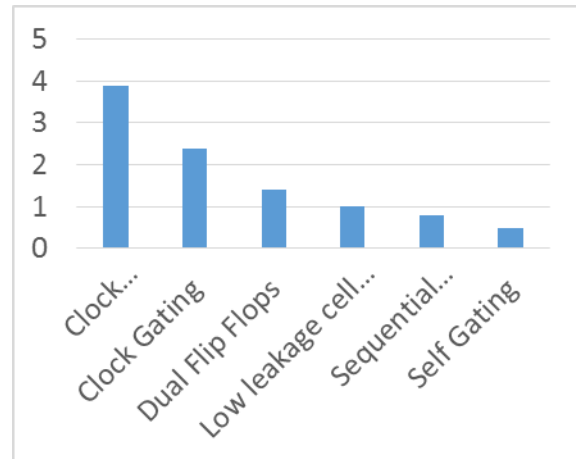


Figure 5 Power Gain Post Experiment

[3] TIMING RESULT

Parameter	Before Experiment(ns)	After Experiment(ns)
WNS	-0.036	+0.007
TNS	-3.127	-0.063
Hold WNS	-0.053ps	-0.009

VI. CONCLUSION

A complete Physical design flow for is done for cache Associativity varying block and cache prefetching block. The Schematic implemented is verified with the RTL and found that it is matching with the RTL logic. Initially the designs were exhibiting negative slacks in both setup and hold. The timing is converged to 0psec later in both in external and internal path and till -30ps for hold as per the design target. The bad extraction due to the timing convergence is recovered in layout loop. The Dissertation work has provided a deep knowledge on custom digital circuit design flow and register file design for the latest technology with advanced industry tools. Designing functional unit blocks in advance process node and optimizing in timing and power for the next generation microprocessor was really challenging. According to the new process core architecture the functional unit blocks are designed. As the number of core increases the power consumption also plays a major factor in design of microprocessor. So with the multi core processor the design target is to achieve low power at high speed. Various design and implementation challenges are analysed and properly addressed with optimized solutions. Many backend and front end

methods are implemented to achieve power, performance and area. Complete mode of work for the backend design is done to enable FEV, Timing, Extraction, Noise, Power Design quality on the data path blocks and mixed design block before completing the milestone. In future there is enough scope to implement new architecture with improved power performance targets and speed.

REFERENCES

- [1] M. law, "Intelprocessors,"<http://www.intel.com/museum/archives/historydocs/mooreslaw.htm>.
- [2] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, "Digital Integrated circuits. Prentice hall Englewood Cliffs", vol. 2, 2002
- [3] I. library, "Intel design methodology," <http://www.intelpedia.intel.com>.
- [4] Kei-yong khoo , "Formal Verifications in modern chip design", IEEE , 26th February 2007
- [5] Guangqiu Chen, Hidetoshi Onodera , Keikichi Tamaru "Timing and Power Optimization by Gate Sizing Considering False Path" , IEEE 6th August 2002.
- [6] R. W. Keyes,"The impact of moore's law",IEEE solid – state circuits society Newsletter,vol. 11,no. 5,pp,25-27,Sept 2006
- [7] H. Bhatnagar, Advanced ASIC chip synthesis:Using S ynopsys Design Compiler Physical Compiler and Prime Time 2nd .[Online]Available :<http://ebooks.kluweronline.com>
- [8] S. Pallela, N. Menezes, and L.T. Pillage,"Low Power IC Clock tree design",in Proceedings of the IEEE 1995 Custom Integrated Circuits Conference, May 1995,pp.263-266.
- [9] W.M.D.J.G.Xi,"Buffer Insertion and sizing under process variations for low power clock distribution",in 32nd Design Automation Conference,June 1995,pp.491-496
- [10] V.Tirumalashetty and H. Mahmoodi, "Clock gating and negative edge triggering for energy recovery clock", in 2007 IEEE International Symposium on Circuits and Systems,May 2007,pp. 1141-1144.