Circuits and Systems, 2016, 7, 1361-1368
Published Online June 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.78119

A New Clock Gated Flip Flop for Pipelining Architecture

Krishnamoorthy Raja[1], Siddhan Saravanan[2]

[1]Excel Engineering College, Komarapalayam, India
[2]Muthayammal Engineering College, Rasipuram, India
Email: krajameae@gmail.com, saravanan.nivi@gmail.com

Received 29 March 2016; accepted 25 April 2016; published 9 June 2016

Abstract

The objective of the work is to design a new clock gated based flip flop for pipelining architecture. In computing and consumer products, the major dynamic power is consumed in the system's clock signal, typically about 30% to 70% of the total dynamic (switching) power consumption. Several techniques to reduce the dynamic power have been developed, of which clock gating is predominant. In this work, a new methodology is applied for gating the Flip flop by which the power will be reduced. The clock gating is employed to the pipelining stage flip flop which is active only during valid data are arrived. The methodology used in project named Selective Look-Ahead Clock Gating computes the clock enabling signals of each FF one cycle ahead of time, based on the present cycle data of those FFs on which it depends. Similarly to data-driven gating, it is capable of stopping the majority of redundant clock pulses. In this work, the circuit implementation of the various blocks of data driven clock gating is done and the results are observed. The proposed work is used for pipelining stage in microprocessor and DSP architectures. The proposed method is simulated using the quartus for cyclone 3 kit.

Keywords

Selective Look Ahead Clock Gating, Clock Gating, Clock Networks, Dynamic Power Reduction

1. Introduction

In digital systems, power consumption becomes a most important parameter to be reduced. Clock gating is one of the design methodologies for reducing the power consumed by digital systems. Several gating methods like synthesis-based, data-driven method and auto-gated FFs (AGFF) are simple but yield relatively small power savings. In literature, several methods are been presented. In this paper [1] [2] Luca Benini *et al.* stated on Dy-

How to cite this paper: Raja, K. and Saravanan, S. (2016) A New Clock Gated Flip Flop for Pipelining Architecture. *Circuits and Systems*, **7**, 1361-1368. http://dx.doi.org/10.4236/cs.2016.78119

namic power management (DPM) execution levels with a base number of dynamic segments or a base burden on such segments are given demand administrations to powerfully reconfiguring outline strategy DPM incorporates an arrangement of strategies that accomplishes vitality proficient calculation by specifically killing framework segments when they are unmoving. In this paper, study depends on a few ways to deal with framework level element power administration. M. Samy Hosny *et al.* [3] presented about late silicon process innovation headways have given chip planners joining capacities that were never conceivable, and have prompted another rush of complex ASICs (Applied Specific Integrated Circuits). These propelled forms accompany new difficulties. This paper shows a percentage of the difficulties in profound submicron advancements, which require new outline hones. We exhibit a few issues identified with timing techniques and timing conclusion that are experienced amid the outline of a FEC40 ASIC, and a system is proposed to moderate some of these issues. In [4] Chunhong Chen *et al.* display an action delicate clock tree development procedure for low power outline of VLSI clock systems. Chunhong Chen presents the term of hub distinction taking into account module action data, and demonstrates its association with the force utilization. A twofold clock tree is manufactured utilizing the hub distinction between various modules to upgrade the force utilization because of the interconnections and furthermore build up a technique to decide gating signals with least number of moves. After the clock tree is developed, the gating signs are upgraded for further power investment funds. Amir H. Farrahi *et al.* [5] examine lessening the force utilization of a synchronous computerized framework by minimizing the aggregate force devoured by the clock signals. Amir proposes three novel action driven issues: a clock tree development issue, a clock door insertion issue, and a zero-skew clock entryway insertion issue. The target of these issues is to minimize framework's energy utilization by building an action driven clock tree. This paper proposes an estimation calculation in view of recursive coordinating to take care of the clock tree development issue and furthermore propose an accurate calculation utilizing the dynamic programming. Shmuel Wimer *et al.* [6] [7] presented on Clock gating is extremely helpful for decreasing the force devoured by advanced frameworks. Three gating strategies are known. The most well known is union based, determining clock empowering signals in light of the rationale of the basic framework. An information driven system stops the vast majority of those and yields higher force investment funds, however, its usage is perplexing and application subordinate. Method three called auto-gated FFs (AGFF) is simple but yields nearly small power savings. This paper presents a novel method called *Look-Ahead Clock Gating* (LACG), which combines all the three. LACG processes the clock empowering signs of each FF one cycle early, taking into account the present cycle information of those FFs on which it depends. Shmuel Wimer *et al.* [8] proposed on VLSI chips have Gclock signal in these days a mainstream design methodology for less switching power consumption. This paper builds up a probabilistic model of the clock gating system that permits us to evaluate the normal force investment funds and the inferred overhead. Expressions for the force investment funds in a gated clock tree are displayed and the ideal gater fan-out is inferred, in light of flip-failures flipping probabilities and process innovation parameters. The timing ramifications of the proposed gating plan are examined. The gathering of FFs for a joint timed gating is likewise examined.

2. Literature Survey

In Literature several algorithm is proposed for the designing the structure for high-speed multiplier circuit. Recently hybrid methods were also proposed like wave pipeline structures when array multipliers were still dominating the world of computing device. In their paper [9] Alexandru Amaricai *et al.* have proposed a dedicated Divide add fused architecture which performs the combined operation of floating-point (FP) division and addition/subtraction. The fused design unit increases the accuracy and performance of applications where this combined operation is frequent, such as the interval Newton's method or the polynomial approximation. The proposed DAF unit even though looks like FP multiply-accumulate units the divider is designed based on digit-recurrence algorithms. The design tradeoff is lesser latency for best cost. In [10] a fused floating point based FFT implementation is proposed based on two fused floating-point operations. The proposed operation is based on fusing two-term dot product and an add-subtract unit. The work is further extended in the radix-2 and radix-4 butterflies implementation efficiently with the two fused floating-point operations. The paper proves that the fused FFT butterflies are about 15 percent faster and 30 percent smaller than a conventional implementation. Also the findings demonstrate the numerical results to be slightly more accurate through the usage of fewer rounding operations. Nikolaidis *et al.* [11] proposed a novel method for the accurate calculation of the transition activity at the nodes of a multiplier-accumulator (MAC) architecture for finite impulse response filters. The transition activity per bit of a signal word is modeled according to the dual-bit-type (DBT) model. An efficient

analytical method based on multiplexing in time of signal sequences with known statistics has been proposed for the determination of the signal statistics at each node of the MAC architecture. The paper presents the experiments carried out both with synthetic and real data and proves its efficiency. Nowadays compressors are widely been used for multiplier implementation. A 16-Bit by 16-Bit MAC is implemented using Fast 5:3 Compressor cells. The cell is designed by applying two rows of fast 2-bit adder cells to five rows in a partial product matrix. The paper reports a 14.3% speed improvement in terms of XOR gate delay on the usage of compressor cell. For a dynamic CMOS circuit implementation using 0.225 μm bulk CMOS technology the reported speed improvement is 11.7% with 8.1% less power consumption. Young-Ho Seo *et al.* [12] proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. The overall performance was elevated b proposing a CSA tree using 1's-complement-based radix-2 modified Booth's algorithm (MBA) and a modified array for the sign extension. The proposed tree architecture propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to decrease the number of the input bits of the final adder. The intermediate results are accumulated in the type of sum and carry bits through pipeling through which performance is improved. The paper reports the experimental results of the proposed architecture in 250 nm, 180 nm, 130 nm, and 90 nm standard CMOS library. Wen-Chang Yeh *et al.* [13] in their paper presented a novel split-radix fast Fourier transform (SRFFT) pipeline architecture design using mapping methodology. The latency between complex multiplication and butterfly operation is balanced. The reported power consumption is reduced by an amount of 15%. A redundant arithmetic based FFT butterfly implementation based on utilization of carry-save adders and a signed-digit representation of the multipliers in the multiplications is proposed in [14]. Other works based on sum-of-product (SOP) blocks [15], fast multiplier [16] is done. Marc Daumas *et al.* [17] proposed a booth multiplier accepting both a redundant and redundant input with no additional delay. Other multiplier design includes Left-to-Right Array Multiplier Design proposed by Zhijun Huang *et al.* [18] which is based on signal flow optimization, left-to-right leapfrog (LRLF) signal flow, and splitting of the reduction array. In [19] Monica Donno *et al.* present a procedure in which low-power clock trees are acquired through forceful abuse of the clock-gating innovation. Recognizing elements of the technique are: 1) the capacity of figuring effective clock-gating conditions that go past the basic topological hunt of the RTL source code; 2) the ability of deciding the clock tree sensible structure beginning from a RTL depiction; 3) tThe capacity of incorporating into the expense work that drives the era of the clock tree structure both useful (*i.e.*, clock enactment conditions) and physical (*i.e.*, floor arranging) data; 4) the ability of creating a clock tree structure that can be integrated and directed utilizing standard, industrially accessible back-end instruments.

3. Background Methodology

Autogated circuits are now used in most of the computing devices where the timing errors are more. The circuit has a latch, a XOR gate. A master latch becomes transparent on the falling edge of the clock and the XOR gate indicates whether or not the slave latch should change its state. The output of the master latch should stabilize within the setup time of the next clock. The method suffers from two major drawbacks. Firstly, only the slave latches are gated, leaving half of the clock load not gated. Secondly, serious timing constraints are imposed on those FFs residing on critical paths, which avoid their gating. These drawbacks are rectified in look-ahead clock gating (LACG) which works on gating the master latch also, making it applicable for large and general designs and avoiding the tight timing constraints. LACG is based on using gates to generate clock enabling signals of preceding FFs. But the gate signal generator circuit has a narrow window around the clock rising edge. **Figure 1** shows an example circuit and the power consumed by the extra latch can be reduced by gating its clock input clk_g. It is subsequently shown that clk_g probability is very low and it is therefore not further being gated. An overhead in this design is the consumption of power in the additional FF used for gating.

The design also requires proper signal sequencing. Even though clock gating is one of the important technique to reduce the dynamic power the application of Clock gating at all levels is a limiting factor due to designing the clock enabling signals is complex. Apart from the above diagram discussed the clock gating signals are derived by Pulse generators and clock generation circuits. The clock capacitive load occupies nearly 70% of their total load. The blocks are increasingly ordered by their data-to-clock activity ratio. In addition the switching of system's clock load is redundant, but consumes most of its power. The above issues can be solved by adding the data-driven clock gating methodology to the existing design. The data-driven gating method is shown in **Figure 2**.

Data-driven gating is illustrated in **Figure 2**. A flip flop finds out that its clock can be disabled in the next

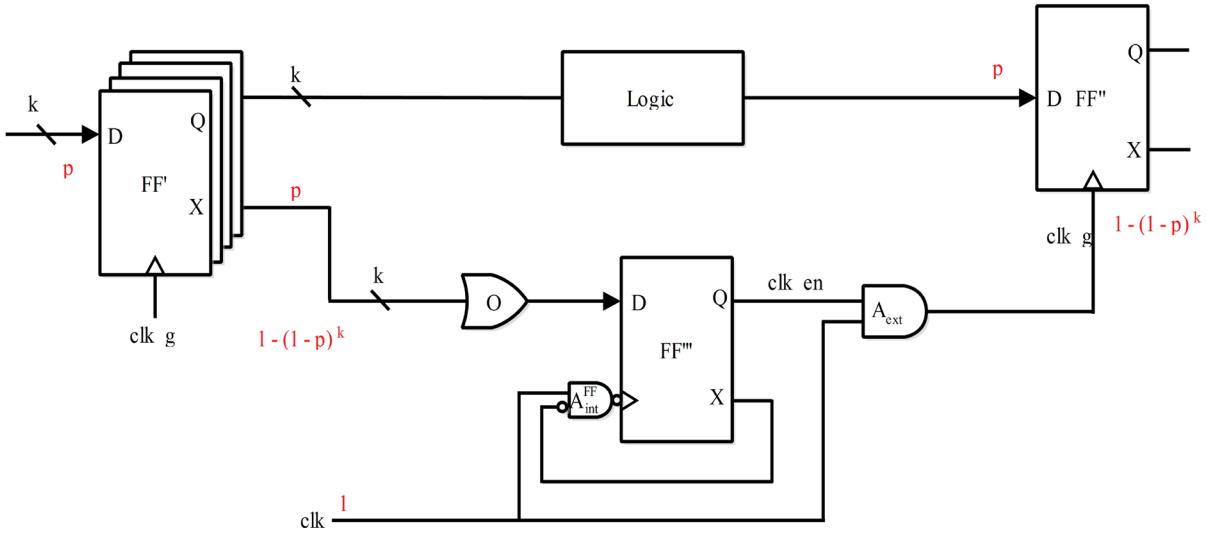

Figure 1. LACG of general logic.

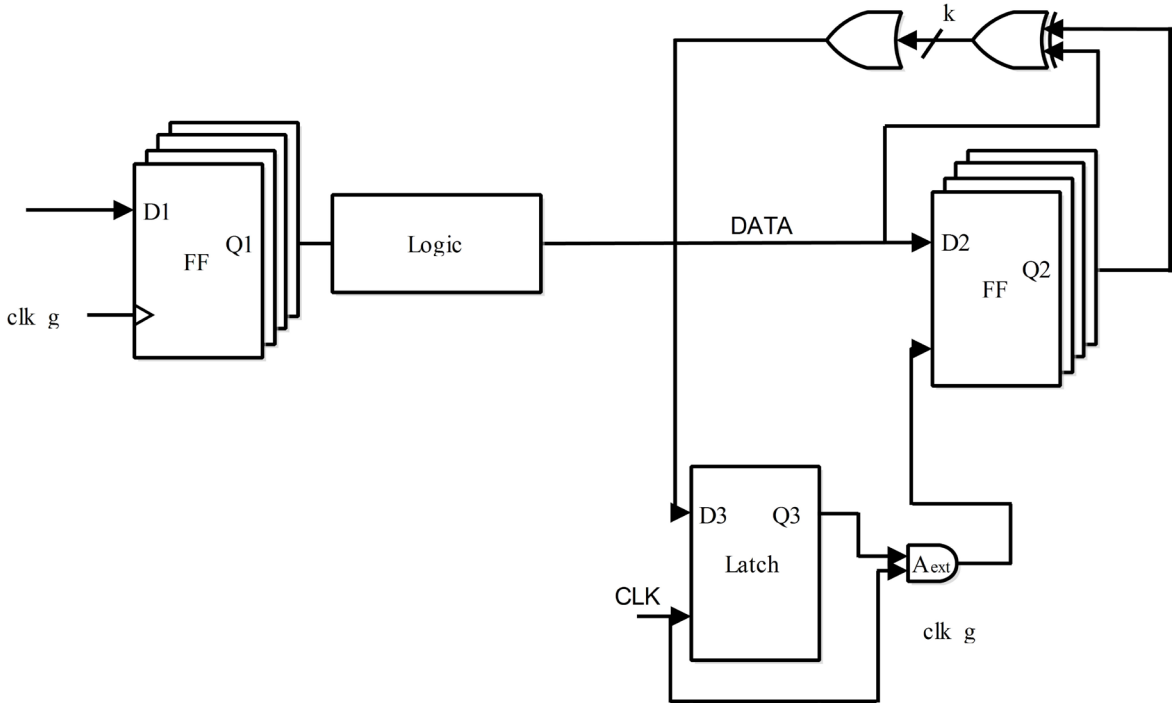

Figure 2. Data driven clock gating flip flop.

cycle by XORing its output with the present input data that will appear at its output in the next cycle. The outputs of XOR gates are ORed to generate a joint gating signal for FFs, which is then latched to avoid glitches. The combination of a latch with AND gate is used by commercial tools and is called Integrated Clock Gate (ICG) [11]. It is beneficial to group FFs whose switching activities are highly correlated. The work in [10] addressed the questions of which FFs should be placed in a group to maximize the power reduction, and how to find those groups. Data-driven gating suffers from a very short time-window where the gating circuitry can properly work. This is illustrated in **Figure 2**. The cumulative delay of the XOR, OR, latch and the AND gater must not exceed the setup time of the FF. Such constraints may exclude 5% - 10% of the FFs from being gated due to their presence on timing critical paths [10]. The exclusion percentage increases with the increase of critical paths, a situation occurring by downsizing or turning transistors of non-critical path to high threshold voltage (HVT) for further power savings.

The pipeline architecture block of an existing methodology is given in **Figure 3**. The 3 stage pipeline architecture consists of 3 M-Bit flip-flops and gates as combinational block. Stage 1 and stage 3 in the diagram is clock gated and the amount of delay taken by stage is reduced by gating with a nominal power reduction.

The proposed architecture is shown in **Figure 4** where the additional flipflop is replaced with a multiplexer.

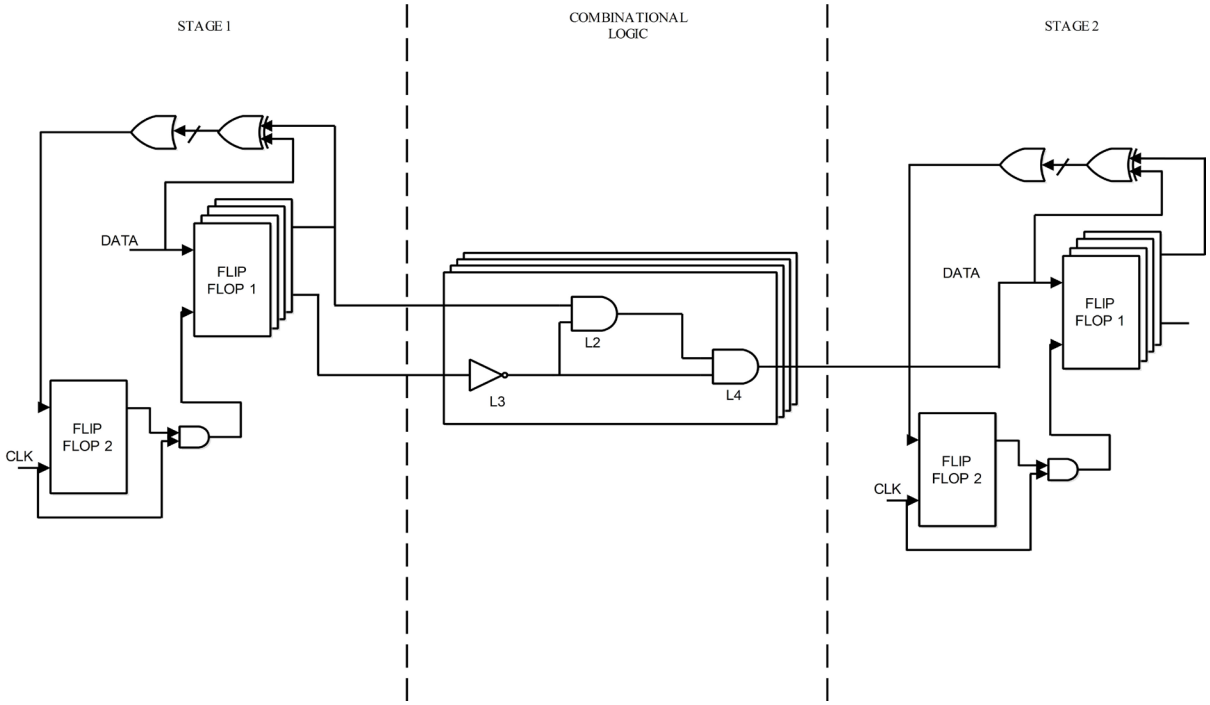

Figure 3. The clock gated pipeline architecture structure.

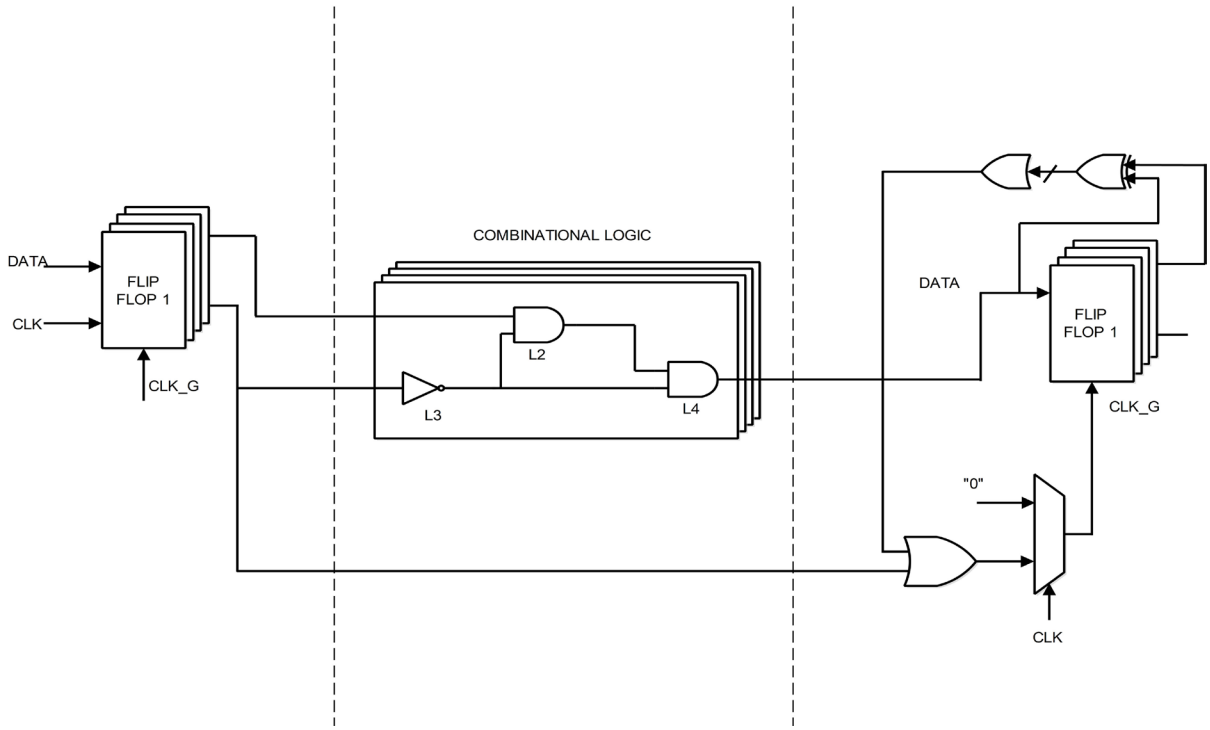

Figure 4. Proposed gated pipeline architecture structure.

The structure combines the selectivity and look-ahead architecture. The method is hybrid by combining the advantage of clock gating and data driven architecture, the analysis and results shows that the proposed method is advantage when compared to existing methods. Circuit implementation of data-driven clock gating is illustrated in **Figure 2**. A FF finds out that its clock can be disabled in the next cycle by XORing its output with the present input data that will appear at its output in the next cycle. The outputs of XOR gates are ORed to generate a joint gating signal for FFs, which is then latched to avoid glitches. The combination of a latch with AND gate is used by commercial tools. But the difficulty of data-driven gating is its design methodology. To maximize the power savings, the FFs should be grouped such that their toggling is highly correlated.

4. Proposed Methodology

Pipelining is a design technique to speed up the execution time of the computation circuits by which the performance and throughput will be improved. The methods under test for the proposed methodology use Flip-flops as the basic elements of pipelining and synchronize the data flow during computation. The elements chosen are with low latency and power consumption.

The pipeline architecture block of a existing methodology is given in **Figure 3**. The 3 stage pipeline architecture consists of 3 M-Bit flip-flops and gates as combinational block. Stage 1 and stage 3 in the diagram is clock gated and the amount of delay taken by stage is reduced by gating with a nominal power reduction. The proposed architecture is shown in **Figure 4** where the additional flipflop is replaced with a multiplexer. The structure combines the selectivity and look-ahead architecture. The method is hybrid by combining the advantage of clock gating and data driven architecture. The analysis and results shows that the proposed method is advantage when compared to existing methods.

5. Result and Discussion

A 3 stage pipelining is designed using proposed FLIP FLOP. Timing analysis and power analysis is been done. The work is done with a 3 stage consisting of 3 combinational blocks and 3 sequential blocks. The block is designed for multibit upgrade. The design will be further improvised and tested for 5 stages with the proposed FLIP FLOP for a particular application. The combinational block is updated with array multiplier and RCA block to mock the actual pipeline stage in a processor core. The block is designed for multibit upgrade. The design will be further improvised and tested for 5 stages with the proposed FLIP FLOP for a SOC application. In the pipeling 3 stages are present. STAGE 1 & 3 corresponds to DDCG FLIP FLOP and stage 2 corresponds to combinational logic function. **Table 1** gives the power analysis of the three blocks. While **Table 2**, **Table 3** presents the Timing Analysis for Stage 1 & 3 and stage 2 respectively. While **Table 4** provides the timing analysis of the proposed pipelining architecture using gated flipflop.

Table 1. Power analysis for block 1, 2 & 3.

Parameter	Block 1 In (μw)	Block 2 In (μw)	Block 3 In (μw)
Total Thermal Power Dissipation	350.17	328.95	331.63
Core Dynamic Thermal Power Dissipation	4.52	1.06	1.14
Core Static Thermal Power Dissipation	303.24	303.03	303.05
I/O Thermal Power Dissipation	42.40	24.87	27.44

Table 2. Timing analyzing of the stage 1 & 3.

Type	From	To	Actual Time In (ns)
Worst-Case Tsu	Clk	Block3:L3|Dflipf:L1|Nandgate2:L3|D~1	5.945
Worst-Case Tco	Block3:L3|Dflipf:L1|Nandgate2:L3|D~1	N	9.998
Worst-Case Tpd	Clk	N	12.754
Worst-Case Th	D[0]	Block3:L3|Dflipf:L1|Nandgate2:L3|D~1	2.550
Clock Setup: 'Clk'	Block3:L3|Dflipf:L1|Nandgate2:L3|D~1	Block3:L3|Dflipf:L1|Nandgate2:L3|D~1	299.31 MHz (Period = 3.341 Ns)

Table 3. Timing analyzing of the stage 2.

Type	From	To	Actual Time In (ns)
Worst-Case Tpd	Q	D	9.112

Table 4. Power analysis for clock gating based on proposed flip flop.

Parameter	Stage 1 In (μw)	Stage 2 In (μw)	Stage 3 In (μw)
Total Thermal Power Dissipation	327.57	327.53	327.57
Core Dynamic Thermal Power Dissipation	1.53	0.26	1.53
Core Static Thermal Power Dissipation	303.01	303.01	303.01
I/O Thermal Power Dissipation	23.02	24.26	23.02

6. Conclusion

The design objective of the work for a new clock gated based flip flop for pipelining architecture is met. The method will be suitable for the pipelining used in DSP and Microcontroller devices. The clock gating will increase the performance and reduce the power consumption. The Selective Look-Ahead Clock Gating reduces the power by reducing the ON period of the device based on computing the clock enabling signals of each FF one cycle ahead of time, based on the present cycle data of those FFs on which it depends. In this work, the design is done to stop the majority of redundant clock pulses. The power analysis and timing analysis for the different blocks involved in the look-ahead clock gating is observed. From the analysis, it's found that the proposed method consumes less power when compared to the conventional method. The power consumption of the buffer stages is reduced in the proposed method. In near future, optimistic methodology will be adopted to reduce the delay further in the processing element block.

Acknowledgements

The authors are thankful for the support from the Nanoelectronics and Integration Division (NAID) of IRRD Automatons (Institute for Robotics: Research and Development), Karur, India.

References

[1] Benini, L., Bogliolo, A. and De Micheli, G. (2000) A Survey on Design Techniques for System-Level Dynamic Power Management. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, **8**, 299-316. http://dx.doi.org/10.1109/92.845896

[2] Hosny, M.S. and Yuejian, W. (2008) Low Power Clocking Strategies in Deep Submicron Technologies. *Proceedings of IEEE International Conference on IC Design & Technology*, *ICICDT*, Austin, 2-4 June 2008, 143-146.

[3] Chunhong, C., Changjun, K. and Majid, S. (2002) Activity-Sensitive Clock Tree Construction for Low Power. *Proceedings of the* 2002 *International Symposium on Low Power Electronics and Design*, Article No. 7502608, 279-282. http://dx.doi.org/10.1109/lpe.2002.146755

[4] Farrahi, A., Chen, C., Srivastava, A., Tellez, G. and Sarrafzadeh, M. (2001) Activity-Driven Clock Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **20**, 705-714. http://dx.doi.org/10.1109/43.924824

[5] Wimer, S. and Koren, I. (2012) The Optimal Fan-Out of Clock Network for Power Minimization by Adaptive Gating. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, **20**, 1772-1780. http://dx.doi.org/10.1109/TVLSI.2011.2162861

[6] Wimer, S. and Koren, I. Design Flow for Flip-Flop Grouping in Data Drive Clock Gating. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*. (To Be Published)

[7] Donno, M., Macii, E. and Mazzoni, L. (2004) Power-Aware Clock Tree Planning. *Proceedings of the* 2004 *International Symposium on Physical Design*, Arizona, 18-21 April 2004, 138-147. http://dx.doi.org/10.1145/981066.981097

[8] Muller, M., Simon, S., Gryska, H., Wortmann, A. and Buch, S. (2006) Low Power Synthesizable Register Files for Processor and IP Cores. *Integration*, *the VLSI Journal*, **39**, 131-155. http://dx.doi.org/10.1016/j.vlsi.2004.08.001

[9] Amaricai, A., Vladutiu, M. and Boncalo, O. (2010) Design Issues and Implementations for Floating-Point Divide-Add Fused. *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, **57**, 295-299.

[10] Swartzlander, E.E. and Saleh, H.H.M. (2012) FFT Implementation with Fused Floating-Point Operations. *IEEE Transactions on Computers*, **61**, 284-288. http://dx.doi.org/10.1109/TC.2010.271

[11] Nikolaidis, S., Karaolis, E. and Kyriakis-Bitzaros, E.D. (2000) Estimation of Signal Transition Activity in FIR Filters Implemented by a MAC Architecture. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **19**, 164-169. http://dx.doi.org/10.1109/43.822629

[12] Seo, Y.-H. and Kim, D.-W. (2010) A New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, **18**, 201-208.

[13] Yeh, W.-C. and Jen, C.-W. (2003) High-Speed and Low-Power Split-Radix FFT. *IEEE Transactions on Signal Processing*, **51**, 864-874. http://dx.doi.org/10.1109/TSP.2002.806904

[14] Bruguera, J.D. and Lang, T. (1996) Implementation of the FFT Butterfly with Redundant Arithmetic. *IEEE Transactions on Circuits and Systems II*: *Analog and Digital Signal Processing*, **43**, 717-723.

[15] Zimmermann, R. and Tran, D.Q. (2003) Optimized Synthesis of Sum-of-Products. *Conference Record of the* 37*th Asilomar Conference on Signals*, *Systems and Computers*, Washington DC, 9-12 November 2003, 867-872. http://dx.doi.org/10.1109/acssc.2003.1292036

[16] Wallace, C.S. (1964) A Suggestion for a Fast Multiplier. *IEEE Transactions on Electronic Computers*, **EC-13**, 14-17. http://dx.doi.org/10.1109/PGEC.1964.263830

[17] Daumas, M. and Matula, D.W. (2000) A Booth Multiplier Accepting Both a Redundant or a Non Redundant Input with No Additional Delay. *IEEE International Conference on Application-Specific Systems*, *Architectures*, *and Processors*, Boston, 10-12 July 2000, 205-214.

[18] Huang, Z. and Ercegovac, M.D. (2005) High-Performance Low-Power Left-to-Right Array Multiplier Design. *IEEE Transactions on Computers*, **54**, 272-283. http://dx.doi.org/10.1109/TC.2005.51

[19] Tsoumanis, K., Xydis, S., Efstathiou, C., Moschopoulos, N. and Pekmestzi, K. (2014) An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator. *IEEE Transactions on Circuits and Systems—I*: *Regular Papers*, **61**, 1133-1143. http://dx.doi.org/10.1109/TCSI.2013.2283695