

Dynamic Power Reduction of Microprocessors for IoT Applications

M. A. El-Razek ^{1,*}, M. B. Abdelhalim ^{2,a} and H. H. Issa ^{3,b}

¹Department of Electrical and computer engineering, Higher technological institute
10th of Ramadan city, Cairo.

²College of Computing and Information Technology (CCIT), Arab Academy for Sciences
Technology and Maritime Transport (AASTMT), Heliopolis, Cairo, Egypt.

³Department of Electronics and Communications Engineering Arab Academy for Sciences
Technology and Maritime Transport (AASTMT), Heliopolis, Cairo, Egypt.

*engmohamedabdalrazek@gmail.com, ^ambakr@ieee.org, ^bhanady.issa@aast.edu.eg

Abstract – Many companies understand the value of the Internet of Things (IoT) and invest considerable time, money, and effort to realize its potential. However, every new technology faces a million challenges in its initial phases. IoT also poses some grave issues that need to be tackled well in order to utilize its fullest potential. One of these problems is the microprocessor power consumption. It is a major factor that limits the performance of the processor and leads to decrease the microprocessor reliability, life expectancy. There are a lot of techniques that reduce the total power consumed by a microprocessor system. In this paper, we use a clock-gating and Architectural alternatives-based power optimization techniques to reduce the power consumption of storm core processor. It is a 32-bit RISC processor which is compatible to ARM's 32-bit instruction set. All experiments are done on Xilinx Spartan 3E. The total power, dynamic and static powers are calculated by Xpower analyzer. The proposed design reduces the power consumption approximately up to 20%. **Copyright © 2016 Penerbit Akademia Baru - All rights reserved.**

Keywords: Dynamic power consumption, Internet of things; Microprocessor dynamic power; Clock gating; Architectural alternatives-based power optimization

1.0 INTRODUCTION

The Internet of Things (IoT) is a system of interrelated computing devices, its links smart objects to the Internet. Everything in this system has the ability to transfer data over a network without requiring human interaction [1]. Researchers estimate that IoT will consist of 50 billion devices connected to the Internet by 2020 [2]. IoT can help us in transportation, inventory management, promotions, vending technology and many other fields [3,4]. IoT components include sensors, power management devices, amplifiers, signal acquisition devices, microcontrollers, processors, battery monitoring ICs, low power RF ICs and more. Sensors are used for temperature, motion, humidity, acceleration, tilt, pressure, shock, gas, pH, sound and

infrared applications [4,5]. Companies such as Intel, Microchip, Atmel, and ARM are working on such process technologies so that the processors are tailored to meet IoT specifications [6-9].

Nowadays microprocessors are fabricated using CMOS technology [10]. The power consumption is increased with new VLSI technology as the number of transistors are dramatically increased. Table 1 lists some examples of Intel microprocessors used in IoT applications with their different technologies and the corresponding power consumption. Hence, microprocessors consume a lot of power which is converted to heat energy that affect the microprocessor stability and performance. The microprocessor designers tried to solve this problem by using cooling systems, which have a very high cost that is sometimes nearly equal to the computers [11]. Therefore, microprocessor designers try always to reach a microprocessor with a low power consumption, high speed and small fabrication area [12].

Table 1: Performance and power of Intel Microprocessor [13-16]

Technology	Product	Frequency	Power (watts)
0.8 μm	I486	66 MHz	4.9
0.8 μm	Pentium	66 MHz	13
0.6 μm	Pentium	100 MHz	10.1
0.6 μm	Pentium Pro	150 MHz	29.2
180 nm	Pentium III	1.0 GHz	29
180 nm	Pentium 4	2.0 GHz	75.3
65 nm	Core 2 Quad Q6700	2.66 GHz	95
32 nm	core i3-530	3.06 GHz	73
32 nm	Core i5-661	3.33 GHz	87
32 nm	Core i7-970	3.2 GHz	130
45 nm	Xeon E7460 (Used for IoT applications)	2.66 GHz	130
45 nm	Atom 230 (Ultra low voltage processor Used for IoT applications)	1.6 GHz	4
32 nm	Atom D2500 (Ultra low voltage processor Used for IoT applications)	1.87 GHz	10

The total power consumption of the microprocessor is divided into dynamic power, leakage power and short circuit power. The dynamic power is the major part of the total power dissipation of the microprocessor [12]. Dynamic power dissipation in one node is given as:

$$p = \alpha c v^2 f \quad (1)$$

Where α is the switching activity, f is the clock pulse frequency, C is the load capacitance and V is the supply voltage [19].

In the short circuit power, the short current I_{short} momentarily flows between the supply voltage and the ground during switching of CMOS gates. The power is based on estimating the duration of the short current flow, and the amount of the short current I_{short} [20]. The leakage power is the power consumed when the devices are active but the signals do not change their values. In CMOS technology, static power consumption is due to leakage [13,21]. Table 2 tabulates a

comparison between 45 nm and 15 nm floating point unit (FPU). It shows that the static power depends on the technology generation [22].

Table 2: Comparison between 45 nm and 15 nm FPU [22]

	45 nm	15 nm
	T=5ns	T = 5ns
Total dynamic power(mw)	1.8230	0.5206
Cell leakage power (mw)	0.2250	0.1134
Total power (mw)	2.0480	0.6340

The dynamic power donates by 60% - 90 % of the total power dissipation [23, 24]. Clock gating technique is the one of the prime techniques used to reduce the dynamic power. This technique is based on switching off the clock by adding more gates in the clock tree [23, 24]. Accordingly, the switching capacitances of the clock network is reduced and hence the power consumption [25].

A lot of research papers are focusing on dynamic power consumption. In [12] Vasanth and Michael presented survey on the power reduction techniques. These techniques are applied at various levels ranging from circuits to architectures and architectures to system software. There are many techniques in architectural level such as gate level optimization techniques and clock gating techniques. In [26] Himanshu, et al discussed advantages and disadvantages of four basic clock gating techniques using AND gate, Latch, flip flop and MUX. Soni, et al in [27] used the four clock gating techniques on Arithmetic logical unit. They implemented their design on Spartan 3 (90nm) FPGA platform and the power was reduced by 30% to 43%. In [25] Mahendra, et al used the various clock gating techniques on D-flip flop and on 16 bits register. They implemented their design on Virtex-6 (40nm) FPGA platform and he reduced the power consumption by 14% to 15%. In [28] Roopa and S.Y.Kulkarni implemented two clock gating techniques (MUX and AND gate) to 16 bits Arithmetic logical unit on Spartan 6 (45nm) FPGA platform and he reduced the power by 5% to 10%. In [23] Dr. Neelam and Akash presented a simple method to reduce dynamic power consumption of 4 bits synchronous counter by 11% using clock gating technique and implement their design on Virtex-7 (28nm) FPGA platform.

This paper is organized as follows: the selected core processor and the methodology of this work are described in section 2.0, the experimental results are presented in section 3.0. And the final conclusions and future works are presented in section 4.0.

2.0 PROPOSED METHODOLOGY

The purpose of this paper is to reduce the microprocessor dynamic power consumption for IoT applications by using two different techniques, architectural alternatives-based power optimization and clock gating technique. In the first technique, we aim to optimize the hardware of the storm core processor to reduce the dynamic power consumption. And the second technique, the Clock gating technique which, prevents the clock input to the functional modules which are idle. This implies turn off the clock if not needed. We use FPGA technology because it offers flexibility and rapid prototyping capabilities. Accordingly, FPGA technology

provides the ability to prove idea or concept and verify it in hardware without going through the long fabrication process of custom ASIC design. This paper focuses on storm microprocessor which is designed by Stephan Nolting [29]. It is a 32-bit RISC open source soft-core processor which is compatible to ARM's 32-bit instruction set family (ARMv2). The architecture of the storm microprocessor is shown in Fig.

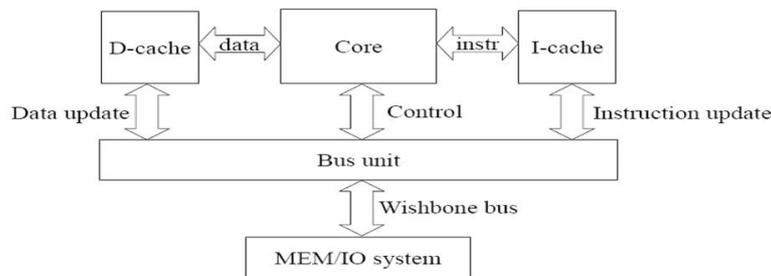


Figure 1: Storm microprocessor Architecture [29]

The detailed architecture of “core” module in storm processor is shown in Fig. 2.

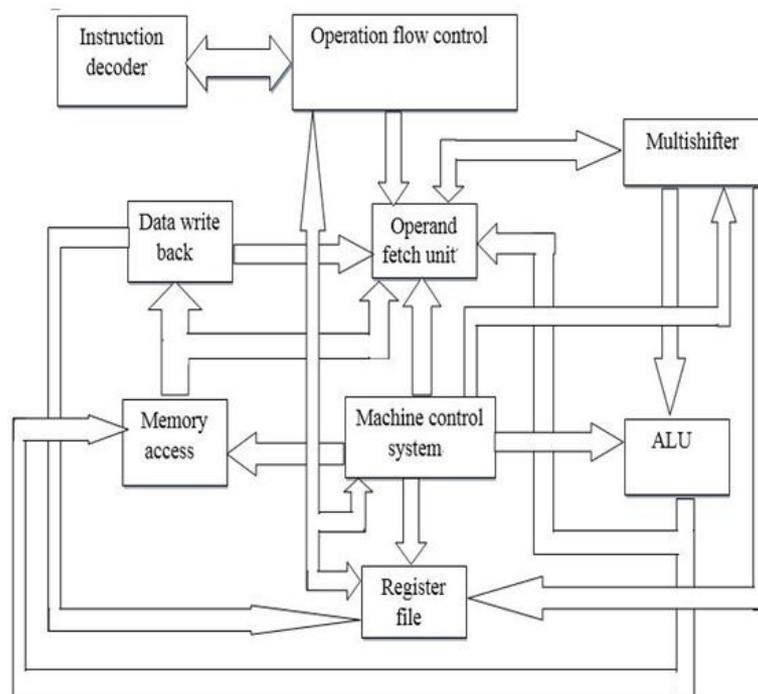


Figure 2: The internal structure of the “core” module

Storm microprocessor has 8 stages pipelined instruction execution. The first stage is an Instruction access (IA). In this stage a new instruction cycle starts with the output of the new value for the program counter which comes from a machine control system module. The new value is the old value + 4, since all instructions are 32 bits wide and have to be aligned. The second stage is an Instruction fetch (IF). In this stage the instruction cache (I-cache) accepts the instruction request and delivers the requested data. The third stage is Instruction decoder

(ID) which decodes the applied opcode into internal control signals. The next stage is Operand fetch (OF) which loads the decoded control information into the needed registers from the register file module. Also operand fetch unit fetch operand in this cycle. In the next stage Multiplication / Shift (MS) a multiplication or a shift of the operands can be applied. The following stage is the Execution stage (EX) where the arithmetical and logical operations take place in the ALU module. The next stage is a Memory access (MA) which can update the machine status registers and the program counter. The last stage is Data write back (WB) stage [29].

The storm microprocessor is equipped with two cache units: a data cache and an instruction cache. Both caches are full associative and can store data from/to any MEM/IO location. The number of cache pages as well as the page size and its coherency strategies can be configured for each cache independently. Together with a bus unit, which connects the cache memories via a 32-bit pipelined Wishbone interface to the rest of the system, these four blocks (core, i-cache, d-cache, bus unit) form the STORM_TOP unit [29].

There are some differences between stormcore processor and ARM processor such as No multiply-long and multiply-accumulate-long instructions are implemented in storm core and there are no restrictions for the use of any register as operand/destination for all instructions [29].

The ARM is very prevalent in low power embedded applications such as Ipods, Palm Pilots, Moreover ARM technology meet the needs to secure interconnectivity of IoT, it provides the quickest and most efficient path to deploying connected platforms and associated services [8]. It is designed for low power consumption, So the ARM technology used in IoT applications such as smart parking, congestion control, smart roads and smart medication [8, 30].

The power consumption of the core module is simulated using XPower analyzer from Xilinx. The simulation results show that the dynamic power consumes around 64% of the total power. The ALU module consumes 43% of the total dynamic power, Multishifter consumes 17% and machine control system consumes 37% of the total dynamic power of the core when a simple application run on the stormcore microprocessor.

3.0 EXPERIMENTAL RESULTS

The dynamic power of the core unit of the storm processor is reduced by two different techniques. These techniques are architectural alternatives – based power optimization technique and clock gating technique. Both techniques are verified by applying VCD file on the XPower analyzer on 90-nm Spartan 3E from Xilinx. All experiments are done with 50 MHz clock frequency. The core module is subjected to different benchmarks which are used to test dynamic power consumption. All these benchmarks are C program. First benchmark adds two 32-bit random numbers, while the second one contains different switch case statements. The third benchmark is a Finite Impulse Response (FIR) filter applied on 32-bit random data. The fourth benchmark, which applied on four 32-bit random data patterns, adds, subtracts, divides and multiplies different 32-bit numbers. The fifth benchmark is Fast Fourier Transform (FFT) which commonly used in IoT applications [31]. The last benchmark used to calculate Cyclic Redundancy Check (CRC) which is a part of the digital ZigBee transmitter and has a potential in IoT [32].

3.1 Architectural Alternatives – Based Power Optimization

This technique focuses on the addition process on the ALU module to examine this technique. The adder module is based on a behavioral method. There are different techniques used for implementing adder circuit such as ripple adder, carry look ahead adder (CLA), Carry Bypass Adder (CBA), Carry Skip Adder (CSKA) and Carry Save Adder (CSA). The power consumption for all these techniques are varied from 0.2 to 1.1 mW as shown in Fig. 3 [33].

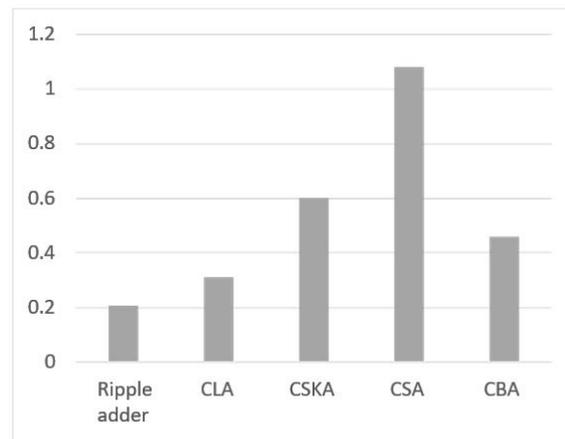


Figure 3: Power consumption by different adder circuits (in mW) [33]

We choose ripple adder and carry look ahead adder because they consume the least power [33]. Table 3 shows the power consumption of the ALU with default adder, ripple adder and CLA module for all benchmarks above.

Table 3: Comparison among the Dynamic power consumption of default adder technique, ripple and, CLA adders

Bench mark	Default ALU	ALU with ripple adder	ALU with CLA
Add two input number	144 mw	140mw	151 mw
(CRC)	144 mw	140 mw	151 mw
Switch case	158mw	154 mw	159 mw
Add , subtract , division , Multiplication	157mw	151 mw	160 mw
FIR	149 mw	144 mw	154 mw
FFT	148 mw	144 mw	153 mw

As we expected the optimization in hardware not affect significantly on dynamic power, so when we use ripple adder we achieve 3.8% dynamic power reduction and the dynamic power increase by 3.9% when we use the carry look ahead adder.

3.2 Clock Gating Technique

To use clock gating technique, we have to disable the clock applied to a certain module in the storm core when this module is not active. The storm core has 9 modules as mentioned before. The ALU and multi-shifter modules are active most of the time and they involved in most of the operations in the processor because of the ALU holds the primary data operation unit. Furthermore, it handles the data access to/from the machine control registers and to/from the system coprocessor. The multi-shifter module contains two parts, multiply unit and barrel shifting unit. The multiply unit

calculates a 32x32 bits operation and outputs the lower 32 bits of the result to the ALU. The barrel shifting unit performs the barrel-shifting of the data in ALU data path. The shift value can either be an immediate value directly from the opcode or a register value. Machine control system module holds the machine control circuits, which include the program counter, the current and saved machine status register as well as the interrupt handler, the branch system and the context change system. Also the internal system control coprocessor is located here. The storm core processor has six different operation modes as tabulated in Table 4. After reset, the processor starts operation always in System mode. To change the mode, the corresponding mode code has to be written to the lowest 5 bits of the Current Machine Status Register CMSR (CPSR in ARM).

Table 4: Operations modes of the storm core microprocessor [29]

Mode	Mode code
User, USR	“10000”
System, SYS	“11111”
Undefined Instruction, UND	“11011”
Supervisor, SVP	“10011”
(Instruction) Abort, ABT (IAB)	“10111”
(DATA) Abort, ABT(DAB)	“10111”
Reserved	--
Interrupt Request, IRQ	“10010”
Fast Interrupt Request, FIQ	“10001”

Machine Control System module contains subsection process for each mode. We applied the clock gating technique on the machine control module. By using the control signal coming from the instruction decoder combined with the system clock through AND gate and feeding the output signal to the interrupt mode subsection as shown in Figure 4, the combined signal is called the gated clock signal.

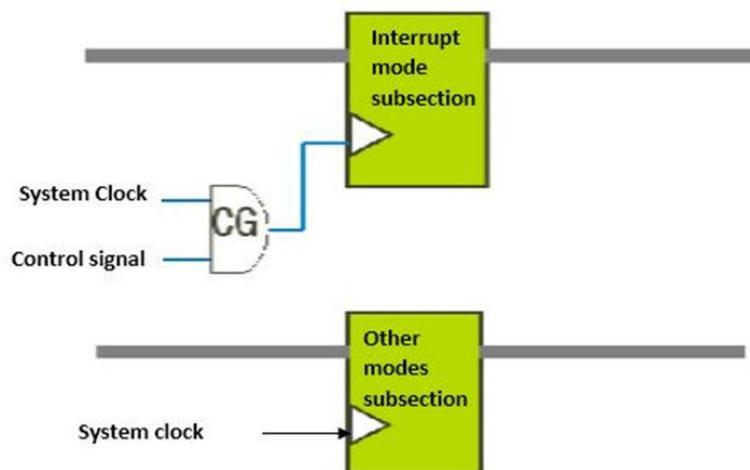


Figure 4: Gated clock and system clock

We disabled the clock which feed the interrupt mode subsection by control signal which comes from instruction decoder and indicates that, when this signal is zero the interrupt mode is not active (most of the time) and when the control signal is one the interrupt mode is active. So, when the control signals equal zero the gated clock equal zero and when the control signals

equal one the clock feeds the interrupt mode subsection. As a result, the dynamic power consumption is reduced by up to 27% as tabulated in Table 5.

The design summary displays the maximum frequency, Net skew, maximum delay and fan out of the design before modification and after modification as tabulated in Table 6.

4.0 CONCLUSION AND FUTURE WORK

This paper proposed two methods Architectural alternatives-based power optimization technique and clock-gating technique to reduce dynamic power of microprocessor for IoT applications which needs low power consumption. The synthesis and (place & route) processes are performed by Xilinx ISE 14.5 and performed on 90 nm Spartan 3E FPGA (XC3S500E) device.

Table 5: Comparison between storm core Dynamic power consumption before modification and with clock gating technique.

Benchmark	Default design		With gated clock		Dynamic power reduction percentage
	Dynamic Power	Utilization	Dynamic Power	Utilization	
Add two input number	144 mw	23%	114 mw	22%	21%
CRC	144 mw	24%	114 mw	23%	21%
Switch case	158 mw	23%	118 mw	22%	25%
Add, subtract, division, Multiplication	157 mw	23%	116 mw	22%	26%
FIR	149 mw	24%	109 mw	23%	27%
FFT	148 mw	24%	108 mw	23%	27%

The Power calculation is performed by Xpower analyzer and simulation is performed by ModelSim SE 6.5. The results show that the Architectural alternatives-based power optimization reduces the dynamic power up to 3.8 % and clock gating technique significantly reduces the dynamic power consumption up to 27 %. In addition, it decreases both the fan out of the design by 11% and the maximum frequency by 14%. The future scope is to use ASIC technology to obtain real power consumption values.

Table 6: Design summary

	Default core	After modification
Net skew	85ps	System clock: 85 ps
		Gated clock: 51 ps
Fan out	1726	1537
Max. delay	201 ps	System clock: 201 ps
		Gated clock: 171 ps
Max. frequency	62.884MHz	53.490 MHz

REFERENCES

- [1] Stankovic, John A. "Research directions for the internet of things." *IEEE Internet of Things Journal* 1, no. 1 (2014): 3-9.
- [2] Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of things: A survey on enabling technologies, protocols, and applications." *IEEE Communications Surveys & Tutorials* 17, no. 4 (2015): 2347-2376.
- [3] Menon, Abhilash, and R. Sinha. "Implementation of Internet of Things in Bus Transport System of Singapore." *Asian Journal of Engineering Research*, Forthcoming (2013).
- [4] Dahir, Hazim, and Bill Dry. *People, Processes, Services, and Things: Using Services Innovation to Enable the Internet of Everything*. Business Expert Press, 2015.
- [5] Swan, Melanie. "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0." *Journal of Sensor and Actuator Networks* 1, no. 3 (2012): 217-253.
- [6] Retrieved on July 28, 2016 from: intel.com/content/www/us/en/internet-of-things/overview.html.
- [7] Retrieved on July 28, 2016 from: microchip.com/design-centers/internet-of-things
- [8] Retrieved on July 28, 2016 from: arm.com/markets/internet-of-things-iot.php
- [9] Retrieved on July 28, 2016 from: atmel.com/applications/IOT/default.asp
- [10] Šojat, Zorislav, Karolj Skala, Branka Medved Rogina, Peter Škoda, and Ivan Sović. "Implementation of advanced historical computer architectures." In *Embedded Engineering Education*, pp. 61-79. Springer International Publishing, 2016.
- [11] Hamann, Hendrik F., Alan Weger, James A. Lacey, Zhigang Hu, Pradip Bose, Erwin Cohen, and Jamil Wakil. "Hotspot-limited microprocessors: Direct temperature and

- power distribution measurements." *IEEE Journal of Solid-State Circuits* 42, no. 1 (2007): 56-65.
- [12] Venkatachalam, Vasanth, and Michael Franz. "Power reduction techniques for microprocessor systems." *ACM Computing Surveys (CSUR)* 37, no. 3 (2005): 195-237.
- [13] Grochowski, Ed, and Murali Annavaram. "Energy per instruction trends in Intel microprocessors." *Technology@ Intel Magazine* 4, no. 3 (2006): 1-8.
- [14] Retrieved on July 28, 2016, from: ark.intel.com/
- [15] Retrieved on July 28, 2016, from: intel.com/content/www/us/en/internet-of-things/products-and-solutions.html
- [16] Retrieved on July 28, 2016, from: ark.intel.com/products/series/36934/Intel-Xeon-Processor-7400-Series#@Server
- [17] Mazhar, Hammad, and Dan Negrut. Comparison of openc1 performance on different platforms using vexcl and blaze. TR-2015-01, 2015.
- [18] Kumar, Ravi, KV Rachana Rao, A. Dakshinamurthy, and D. Chatterjee. "Evolution of Processors and its Implication in Data Computation Capability." 10th Biennial International Conference & Exposition, India 2013.
- [19] Nafkha, Amor, Jacques Palicot, Pierre Leray, and Yves Louët. "Leakage power consumption in FPGAs: thermal analysis." In 2012 International Symposium on Wireless Communication Systems (ISWCS), pp. 606-610. IEEE, 2012.
- [20] Li, Sheng, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi. "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures." In Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 469-480. ACM, 2009.
- [21] Kim, Nam Sung, Todd Austin, David Baauw, Trevor Mudge, Krisztián Flautner, Jie S. Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. "Leakage current: Moore's law meets static power." *computer36*, no. 12 (2003): 68-75.
- [22] Salehi, Soheil, and Ronald F. DeMara. "Energy and Area Analysis of a Floating-Point Unit in 15nm CMOS Process Technology." In SoutheastCon 2015, pp. 1-5. IEEE, 2015.
- [23] Prakash, Dr Neelam R. "Akash, Clock Gating for Dynamic Power Reduction in Synchronous Circuits." *International Journal of Engineering Trends and Technology (IJETT)* 4, no. 5 (2013).
- [24] Srinivasan, Nandita, Navamitha S. Prakash, D. Shalakra, D. Sivaranjani, and B. Bala Tripura Sundari. "Power Reduction by Clock Gating Technique." *Procedia Technology* 21 (2015): 631-635.

- [25] Dev, Mahendra Pratap, Deepak Baghel, Bishwajeet Pandey, Manisha Pattanaik, and Anupam Shukla. "Clock gated low power sequential circuit design." In Information & Communication Technologies (ICT), 2013 IEEE Conference on, pp. 440-444. IEEE, 2013.
- [26] Chaudhary, H., Goyal, N., and Sah, N. "Dynamic Power Reduction Using Clock Gating: A Review" International Journal of Electronics & Communication Technology 6, no.1 (2015).
- [27] Soni, Dushyant Kumar, and Ashish Hiradhar. "Dynamic Power reduction of synchronous digital design by using of efficient clock gating technique."
- [28] Kulkarni, Roopa, and S. Y. Kulkarni. "Power analysis and comparison of clock gated techniques implemented on a 16-bit ALU." In Circuits, Communication, Control and Computing (I4C), 2014 International Conference on, pp. 416-420. IEEE, 2014.
- [29] Retrieved on Jylu 28, 2016, from: opencores.org/
- [30] Adusumilli, Swaroop. "System and method to reduce power consumption in advanced RISC machine (ARM) based systems." U.S. Patent 6,438,700, issued August 20, 2002.
- [31] Tran, Thi Hong, Soichiro Kanagawa, Duc Phuc Nguyen, and Yasuhiko Nakashima. "ASIC design of MUL-RED Radix-2 Pipeline FFT circuit for 802.11 ah system." In Low-Power and High-Speed Chips (COOL CHIPS XIX), 2016 IEEE Symposium in, pp. 1-3. IEEE, 2016.
- [32] Elarabi, Tarek, Vishal Deep, and Chashamdeep Kaur Rai. "Design and simulation of state-of-art ZigBee transmitter for IoT wireless devices." In 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 297-300. IEEE, 2015.
- [33] Kaur, Jasbir, and Lalit Sood. "Comparison Between Various Types of Adder Topologies." IJCST vnuno-ol 6 (2015).