

Power Aware Multiprocessor System on Chip (MPSOC) Based on Dynamic Frequency Scaling of Soft Core Processors

¹Mitali Patel, ²Jitendra B Zalke, ³Sandeep Kumar Pandey, ⁴Anuradha D Verma, ⁵Ruchir V. Nandanwar

¹Government College of Engineering, Raipur, India

^{2,5}Department of Electronics Design Technology, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

^{3,4}Department of Electronics, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
Email: mitalip886@gmail.com, zalkej@rknc.edu, pandeys@rknc.edu, vermaad1@rknc.edu, nandanwarrv1@rknc.edu

Received: 20th September 2018, Accepted: 11th October 2018, Published: 31st October 2018

Abstract

Today's portable Embedded systems need less power consumption to satisfy today's applications needs, like smart phones, laptops, tablets etc. An option for implementing the power aware multi processor system is to introduce the concept of on demand clock scaling of multi processor system. The proposed work presents an implementation of on demand clock scaling of a multiprocessor system on a chip. The proposed architecture uses the Digital Clock Manager (DCM) of the FPGA to change the clock of the soft-core processor. Each soft-core processor in the architecture is allowed to scale the frequency through clock control unit. The clock control unit is responsible for tuning or scaling the clock frequency of processors on demand or according to the application requirements. The entire system is implemented using MicroBlaze soft core from Xilinx and tested on Vertex-5(XUPV5-LX110T) FPGA development board with Xilinx EDK.

Keywords

DCM, MicroBlaze, PE, DFS.

Introduction

Today's embedded systems need more computational power to satisfy modern applications needs, like image processing, audio/video encoding and decoding, etc. and parallel processing using multiprocessor systems on a chip (MPSoC) are an option to deal with this increasing computational needs [1][2][3].

MPSoC environment are created by connecting two or more processors together. The MPSoC environment may be based on multi-core or multi-processor. If different CPUs are sharing the same cache memory module then they are referred as multi-core, and different cache for different CPUs are termed as 'multi-processor' (MP). An MPSoC is a multiprocessor on a single silicon chip. It contains more than one processor memory modules (PMMs) and may contain ten or even hundreds of processor-memory modules (PMMs). The term many core architecture is used for hundreds of cores and more on the same chip that are connected together by appropriate means of communicating bus. Fully custom-designed chips have traditionally implemented MPSoCs, multi- and many-core architectures. During the last few years the capabilities of Field Programmable Gate Arrays (FPGAs) have increased significantly and thus FPGAs allow the implementation of these architectures as an alternative, although with less powerful PMMs today [4].

The operating clock frequency of each processing element play vital role here. In MPSoC system each processing element is operating at specified clock frequency. The power consumed by a MPSoC can be reduced by either Dynamic clock frequency scaling or voltage scaling. Voltage or frequency scaling are important here since, the power consumed by a component implemented in CMOS varies linearly with frequency and quadratically with voltage. So either by scaling the frequency or voltage at which a processing element (PE) operates, a power consumption for a operation will be reduced.

Many systems allow the processor clock to be varied [12]. More recently, there are a number of processors that allow the processor voltage to be changed. For example, the Strong ARM SA-2 processor, currently being designed by Intel, is estimated to dissipate 500mW at 600MHz, but only 40mW when running at 150MHz – a 12-fold energy reduction for a 4-fold performance reduction [5].

The work presented in this paper relies on implementation of power aware multi-processor system on chip (MPSoC), which aims at designing a dynamic clock frequency scaling strategy. In the proposed systems, individual processor core is operated at various clock frequencies. The clock frequency is decided by the main processor and controlled by clock control unit. The clock control unit is responsible for controlling the DCM configuration. For the proposed system, first the major task can be divided into small modules for each processing element in the MPSoC. These tasks can be analyzed for its computational complexity prior to allocate them to each processing element. Based on the complexity of individual modules, the frequency of the processing element is scaled i.e. if the task is

computationally intensive then the frequency of operation is increased and if the task is not computationally intensive then the frequency of operation is reduced. This eventually reduces the power dissipation of the complete system and also reduces the computation overhead on single processor [11].

The paper is organized as follows. Next section presents the background and related work. Section III describes the proposed architecture. Section IV discusses the methodology to implement the proposed system model. Experimental results are presented in Section V while conclusions and future work are drawn in Section VI.

Background and Related Work

Recently, researchers have put focus on adaptation techniques in order to cope up with dynamic and unpredictable behaviors that can appear in today's embedded systems. This section presents work that has been conducted in this direction using techniques such as reconfiguration of DCM and clock frequency scaling.

Power Consumption:

Mathematically, the total power consumption P_{of} a chip is given as:

$$P = P_{\text{Static}} + P_{\text{Dynamic}} \text{-----}(1)$$

Static power (P_{Static}) consumption in a integrated circuits is because of Leakage currents inside the semiconductor material. This leakage current is because of the sub-threshold leakage, which is an inversion current across the device, and the gate-oxide leakage which is nothing but a tunnelling current through oxide insulation of a transistor gate. The leakage is technology-dependent and cannot be altered by the chip user. The dynamic power (P_{Dynamic}) in integrated circuit are arises from switching activities of transistors. The dynamic power consumption is a function of supply voltage V , clock frequency f and chip capacitance C .

One of the benefits of using more than one processing unit versus a single one is that this allows decreasing the frequency correspondingly as per the computational intensity; which allows us to decreases the dynamic power consumption significantly. The dynamic power consumption is:

$$P_{\text{Dynamic}} = \alpha \cdot C_{\text{LOAD}} \cdot VDD^2 \cdot f_{\text{CLK}} \text{-----} (2)$$

P_{Dynamic} is the dynamic power consumption, α is the activity factor, it indicates the average number of switching events undergone by the transistors in the chip, C_{LOAD} the circuit equivalent capacitance, VDD the supply voltage, and f_{CLK} the clock frequency.

As shown above, the dynamic power consumption is dependent on supply voltage. Thus, reducing the supply voltage can dramatically reduce the power consumption for the processor. As a result, many processors are designed to have the Dynamic Voltage and Frequency Scaling (DVFS) capability, i.e. being able to dynamically vary the supply voltage and/or frequency of a microprocessor. DVFS has proven to be one of the most effective ways to reduce the power consumption and also to manage the thermal condition of the processor while meeting the required performance [6].

Power Minimization Technique in Multiprocessor System:

Power consumption and battery life are the important factors in the development of portable electronic appliances; a multimedia processor is often the most power-hungry device in such systems. Two common ways to reduce CPU power requirements is to lower the clock rate at which an entire device operates or to lower the voltage, but this can affect system performance. Meanwhile, creative chip designers have come up with a variety of innovative on-chip methods to reduce power without adversely impacting system performance [7].

The recent multiprocessor system optimizes system power consumption by employing two main techniques:

- 1) Managing active system power consumption.
- 2) Managing standby system power consumption.

Active power consumption management is used to optimize the power a system draws while processing data to achieve useful results. Here, three basic methods exist:

- Dynamic voltage and frequency scaling (DVFS)
- Adaptive voltage scaling (AVS)
- Dynamic power switching (DPS)

Static or standby power consumption management keeps an idle system in a power efficient state until further processing is required. Managed by a technique known as static leakage management (SLM), static power consumption can result in several low-power modes, from standby to a deep sleep mode, which mimics the power-off state but has faster wake-up latency.

In the proposed work emphasis is given on active power consumption management in which the power consumption is managed using dynamic frequency scaling (DFS) technique.

Dynamic Frequency Scaling (DFS):

In Dynamic frequency scaling the frequency of a processor can be adjusted either to save the power to get more battery backup or to reduce the amount of heat generated by the chip. This technique is generally used in battery operated devices like laptops and other portable devices such as mobile, tab etc. In all these portable devices the energy comes from a battery and thus is limited.

The total power dissipated in the chip is due to static and dynamic power. Few years back dynamic power accounted for approximately two-thirds of the total chip power however in contemporary CPUs and SoCs power loss due to leakage current is significant. High leakage current in deep-submicron designs have become a significant contributor to total power dissipation, as short-channel transistors require lower power supply levels to reduce power consumption. This forces a reduction in the threshold voltage that causes a substantial increase of weak inversion current. Hence, it has become extremely important to develop design techniques to reduce static power dissipation during periods of inactivity.

In recent years, however, device densities and clock frequencies have increased dramatically in CMOS devices, thereby increasing the power consumption dramatically. At the same time, supply voltages and transistor threshold voltages have been lowered, causing leakage current to become a significant problem. High-k metal-gates or power gating are used to reduce the leakage power [8]. The supply requirement depends upon the clocked frequency and can be reduced if the frequency of operation is also reduced. Optimal CPU frequency can be computed at which energy consumption for a piece of software is very low.

Performance Impact

Dynamic frequency scaling is a commonly-used method to reduce dynamic (switching) power. However it hampers the performance and hence can be used particularly for memory-bound workloads (not CPU bound)[9][10]. With reduction in the frequency, corresponding supply voltage can also be reduced[13]. A reverse but related technique is over clocking where processors (dynamic) frequency is ramped beyond specification. Specific difference between the two methods is over clocking is static and frequency scaling is dynamic. Frequency scaling algorithm software can include over clocked frequencies, but it may have risk of chip degradation.

Proposed Architecture

Figure (1) shows the overall hardware architecture of proposed system. The complete architecture is called symmetric DFS-MPSoC. It consist one additional processor called master processor that has similar architecture as main processors but that is suitable for low power process & work with the lower frequency as compared to the main (dual) processors. Table (1) shows the basic difference between the master processor and the main processors on the basis of architecture, frequency range and process technology.

The multiprocessor architecture arranged in hierarchical way, one processor as master processor and other two as a slave processor. Master processor sends processing input data to both the slave processor and again two slave processors send back the results to master processor. The two slave processor performs image processing application. The master processor communicate to the 1st slave processor via FSL_0 (sending input data), FSL_1 (receiving output data). The master processor communicates to the 2nd slave processor via FSL_2 (sending input data), FSL_3 (receiving output data). FSL allows asynchronous communication mechanism and is therefore selected to connect the multiple processor cores.

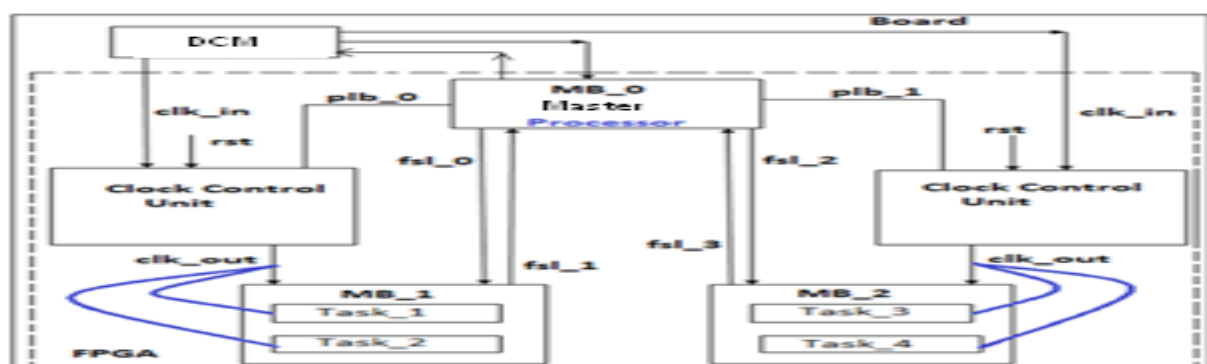


Fig 1: Proposed Hardware Architecture

As shown in fig: 1 the proposed work based on dynamic reconfiguration of DCM. To support different clock speeds on each of the slave processing element units during runtime, a configurable clock control unit is developed that can

be accessed through software. Note that since each processing core can potentially work at two different working frequencies.

	Master Processor (Power Optimization)	Main Processor (Performance Optimization)
1) Architecture	Xilinx MicroBlaze	Xilinx MicroBlaze
2) Frequency Range	Get clock from clock generator (25-125MHz)	Get clock from DCM (DFS) DFS Mode : 32-180MHz
3) Process Technology	Low Power Process	Fast/General Purpose

Table 1: Difference Between the Master Processor and The Main Processors

MicroBlaze soft-processor core:

The MicroBlaze embedded soft core is a reduced instruction set computer (RISC) processor. Figure 2 shows a block diagram of the MicroBlaze core [9]. The MicroBlaze embedded soft-core includes the following features:

- The Microblaze core implements Harvard architecture.
- The processor has Thirty-two 32-bit general purpose registers
- 32-bit instruction word with three operands and two addressing modes
- Separate 32-bit instruction and data buses that conform to IBM's OPB (On-chip Peripheral Bus)
- Separate 32-bit instruction and data buses with direct connection to on-chip block RAM through a LMB (Local Memory Bus) The LMB provides single-cycle access to on-chip dual-port block RAM.
- 32-bit address bus
- Single issue pipeline
- Instruction and data cache
- Hardware debug logic
- FSL (Fast Simplex Link) support
- Hardware multiplier (in Virtex-II and subsequent devices)
-

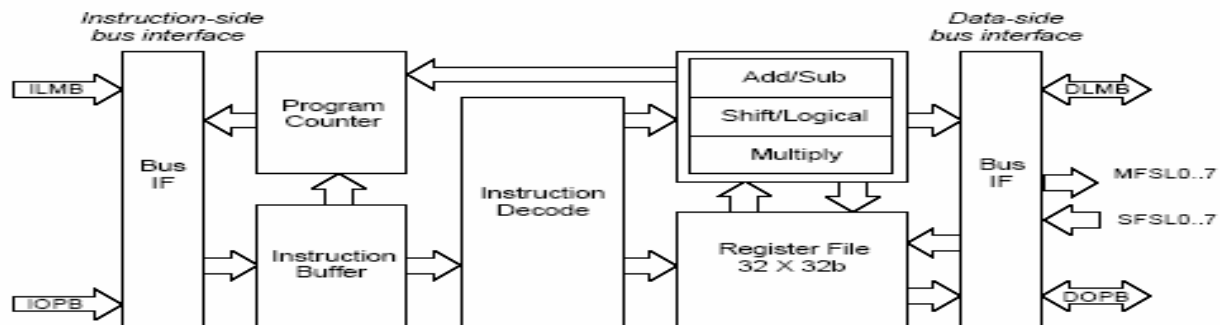


Fig 2: MicroBlaze Block Diagram[9]

Digital Clock Manager (DCM) in FPGA:

In the development of dual processor system, the Digital Clock Manager is used for the generation of different clock frequencies at which each processor is operated. The DCM primitive in Xilinx FPGA parts is used to implement delay locked loop, digital frequency synthesizer, digital phase shifter, or a digital spread spectrum. The digital clock manager module is a wrapper around the DCM primitive which allows it to be used in the EDK tool suite.

The Viterx-5 FPGA (XC5VLX110T) consists of total 12 DCMs block, six at the bottom half and six at the top half of the FPGA. The block diagram of DCM is as shown in figure 3.

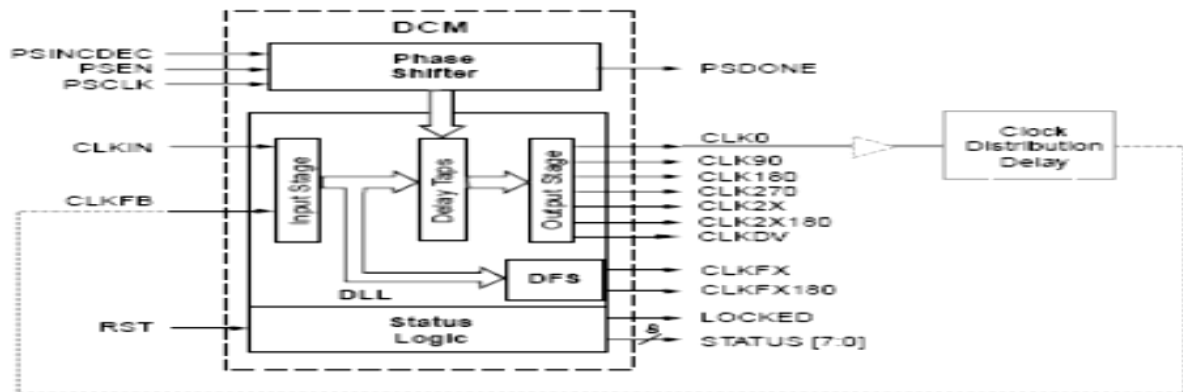


Fig 3: Block Diagram of DCM in FPGA

The Digital Clock Managers (DCMs) in Virtex-5 FPGAs provide a wide range of powerful clock management features like Clock deskew, Frequency Synthesis, Phase Shifting and Dynamic Reconfiguration.

DCM Primitives

The DCM primitives DCM_BASE and DCM_ADV are shown in Figure 4[12].

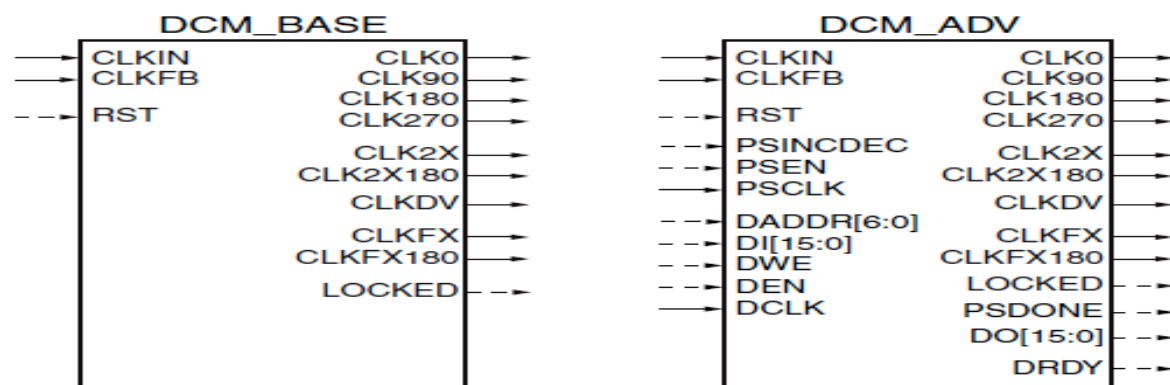


Fig 4: DCM Primitives

DCM_BASE Primitive:

The DCM_BASE primitive is used to access the basic and frequently used DCM features like clocks deskew, frequency synthesis, and fixed-phase shifting etc. Table (2) lists the available ports in the DCM_BASE primitive [3].

Available Ports	Port Names
Clock Input	CLKIN, CLKFB
Control and Data Input	RST
Clock Output	CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180, CLKDV, CLKFX, CLKFX180
Status and Data Output	LOCKED

Table 2: DCM_BASE Primitive

DCM_ADV Primitive:

The DCM_ADV primitive has access to additional ports for the dynamic reconfiguration plus all DCM features and ports available in DCM_BASE primitive. Table (3) lists the available ports in the DCM_ADV primitive.

Available Ports	Port Names
Clock Input	CLKIN, CLKFB, PSCLK, DCLK
Control and Data Input	RST, PSINCDEC, PSEN, DADDR[6:0], DI[15:0], DWE, DEN
Clock Output	CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180, CLKDV, CLKFX, CLKFX180
Status and Data Output	LOCKED, PSDONE, DO[15:0], DRDY

Table 3: DCM_ADV Primitive

Experimental Methodology:

The hardware architecture that consist three MicroBlaze soft-core processors is implemented using Xilinx EDK (Embedded Development Kit) platform [4]. For implementation of three processor system, first the system is generated using XPS platform as shown in figure (5). Figure (6) shows view of generated XPS three processor system.

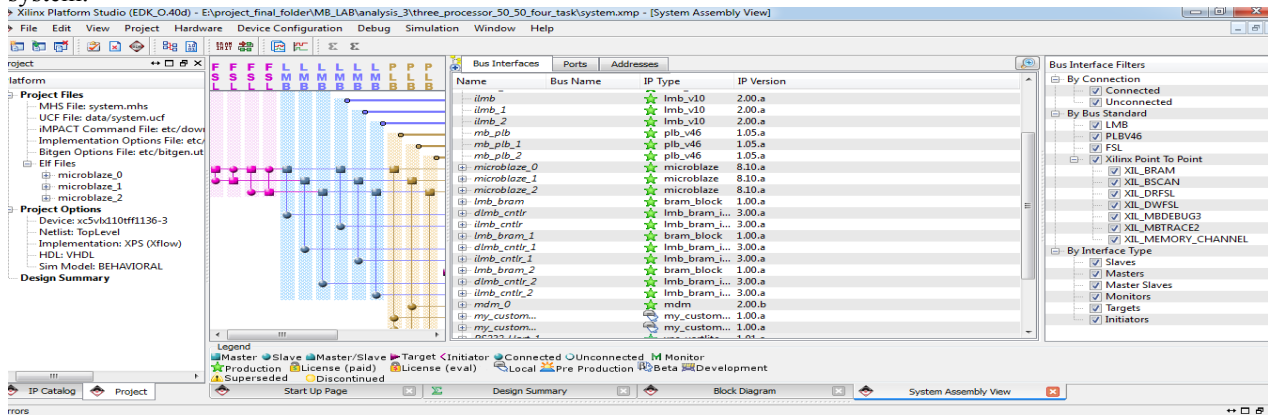


Fig. 5: View of Xilinx Platform Studio (XPS) for The Variable Symmetric Multiprocessor Architecture

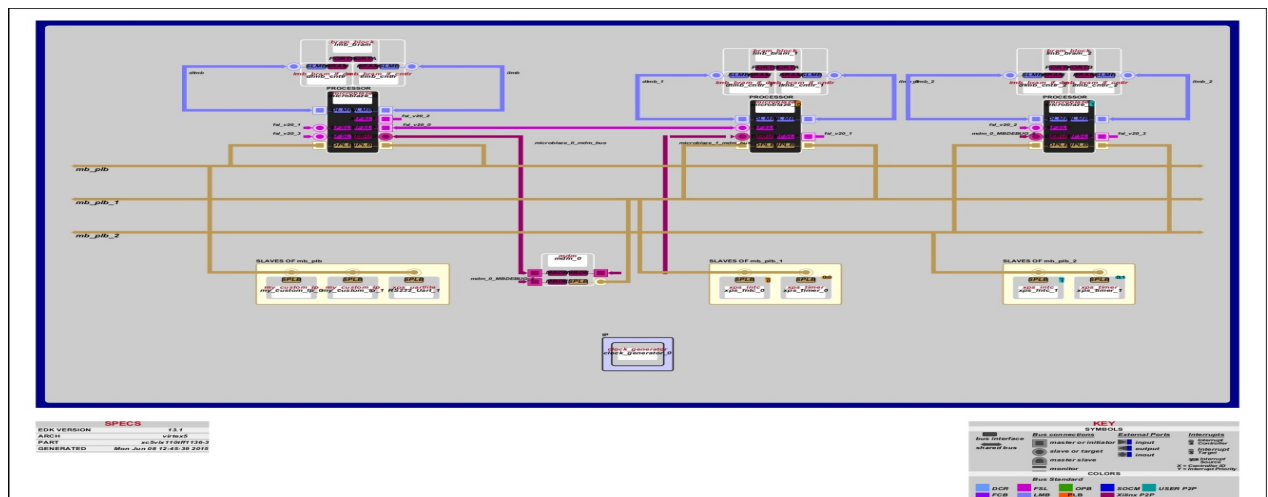


Fig. 6: MicroBlaze Based the Variable Symmetric Multiprocessor Architecture Generated by XPS

Software Application Program

The multiprocessor architecture arranged in hierarchical way & one as master processor and two as a slave processor. Master processor sends processing input data to both the slave processor and again two slave processors send back the results to master processor. The two slave processor performs image processing application. Each slave processor consist two image processing task one is highly computational intensive task another is less computational intensive task. The task computational complexity find prior to the execution of the program with the help of number of the clock ticks required for that particular program. The first slave processor consist two image processing task:

- 1) Histogram equalization of an image.
- 2) Negative of an image.

Out of these two task Histogram equalization of an image required more number of clock ticks as compared to the negative of an image. Similarly the second slave processor consist two image processing task:

- 3) Edge detection of an image
- 4) Image rotation in clockwise direction.

Out of these two task, Edge detection of an image required more number of clock ticks as compared to the Image rotation. Software application flow of the complete system is shown in figure 6.2 & flow chart representation for individual processors are shown in figure 6.3, 6.4, & 6.5.

Result & Conclusion

Figure 7.1 shows the results of image processing application performed by three processor systems. Figure 7.1 (a) shows the original input image, Figure 7.1 (b) shows the histogram equalized image, Figure 7.1 (c) shows the

negative of an image, Figure 7.1 (d) shows the edge detection of an image, & Figure 7.1 (e) shows the image rotation in clock wise direction.

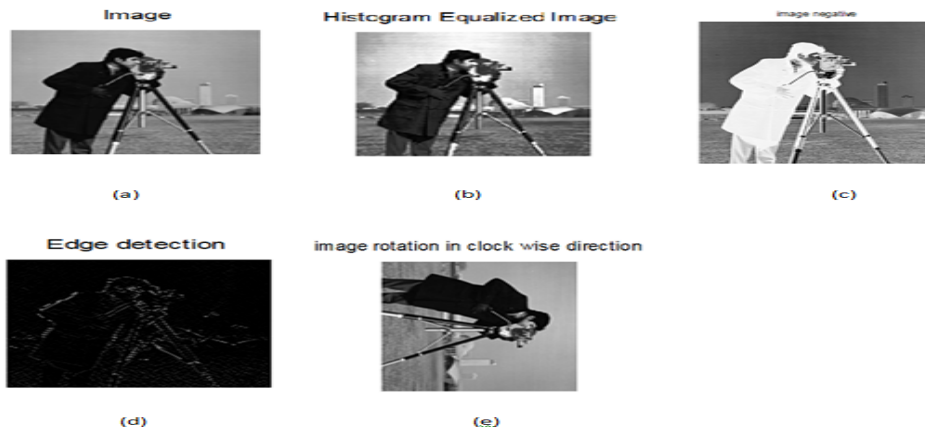


Fig. 7: Results of Image Processing Application Performed by Three Processor Systems

Timing Analysis

The timing analysis is done for the systems (no. of clock ticks as well as execution time calculated for individual task). Table (4) shows the timing analysis results in terms of no. of clock ticks required for each task by selecting the three cases of frequency variation.

Timing analysis is done with the help of XPS timer using few commands which will give the no. of clock ticks required for the particular task

Frequency (MHz)	Execution time(ms) Processor -2		Execution time(ms) Processor -3	
	T1= histogram Equalization	T2= Negative of an image	T3= Edge detection	T4=image rotation
P1=25 P2(T1=50,T2=50) P3(T3=50,T4=50)	35.73	10.86	40.46	10.86
P1=25 P2(T1=150,T2=50) P3(T3=150,T4=50)	11.88	10.86	13.47	10.86
P1=25 P2(T1=150,T2=150) P3(T3=150,T4=150)	11.88	3.60	13.47	3.6

Table 4: Execution Time Ticks Required for Each Task

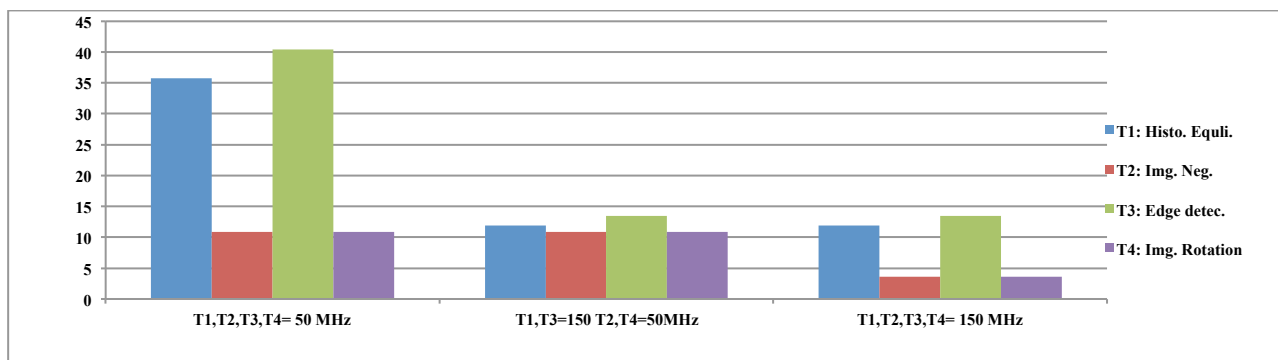


Fig. 8: Graphical Plots for The Timing Analysis

Power Analysis

Table (5) shows the different power consumption in watts for the developed systems. As seen from the table (5), the total power, dynamic power, quiescent power and total on-chip power consumed by the three processor system working when:

- 1) All the task at low frequency.

- 2) Computationally hungry task running at high & less computationally hungry task running at low frequency.
- 3) All the task at high frequency. In all the three cases master processor running at low frequencies.

Frequency (MHz)	Power(Watt)		
	Total on-chip	Dynamic	Quiescent
P1=25 P2(T1=50,T2=50) P3(T3=50,T4=50)	1.677	0.376	1.302
P1=25 P2(T1=150,T2=50) P3(T3=150,T4=50)	1.679	0.378	1.302
P1=25 P2(T1=150,T2=150) P3(T3=150,T4=150)	1.714	0.412	1.302

Table 5: Power Analysis of Developed Systems

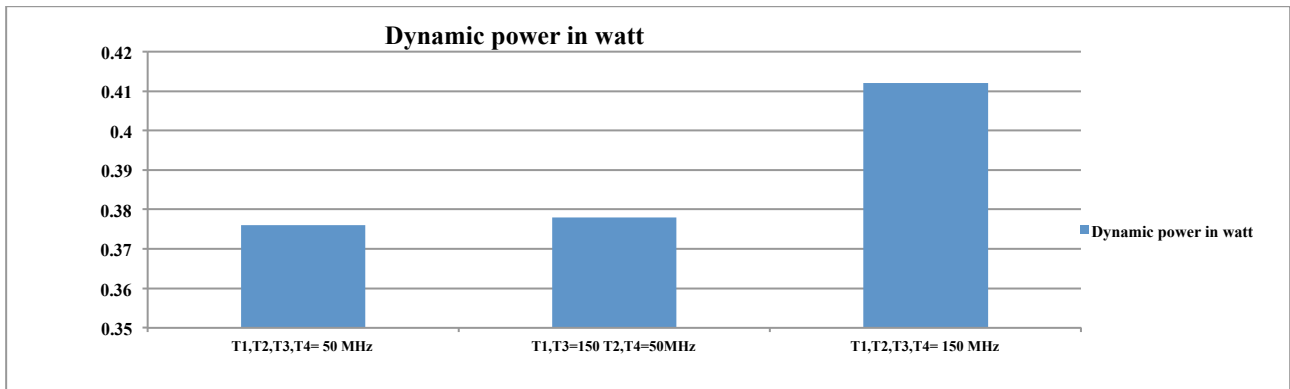


Fig.9: Dynamic Power Consumed in Systems

From the above results it can be concluded that:

- 1) When all the tasks are operating at low frequency, the maximum time at which all task available is 40.86 ms & average. execution time for each task is 24.4775 ms.
- 2) When Computational high intensive tasks are operating at high frequency & less Computational intensive tasks are operating at low frequency then the maximum time at which all task available is 13.47 ms & average execution time for each task is 11.76 ms.
- 3) When all the tasks are operating at high frequency, the maximum time at which all task available is 13.46 ms & average. execution time for each task is 8.1375 ms.

The second case is the dynamic frequency scaling of the processor at which we can change the frequency of the processor at run time. When we compare case 2) from case 1); we get there is reduction in average execution time 34.19 % & when we compare case 3) from case 2); we get there is reduction in average execution time 23.54%.

Regarding the dynamic power consumption;

- 1) When all the tasks are operating at low frequency, the dynamic power consumption is 0.376 W.
- 2) When Computational high intensive tasks are operating at high frequency & less Computational intensive tasks are operating at low frequency then , the dynamic power consumption is 0.378 W.
- 3) When all the tasks are operating at high frequency, the dynamic power consumption is 0.412 W.

When we compare case 2) from case 1); there is increase in dynamic power consumption 0.529 % & when we compare case 2) from case 3); there is reduction in dynamic power consumption 7.62%.

Future Scope

- 1) Due to inaccuracy in the simulator and not being able to dynamically vary the voltage in the present FPGAs, we are not able to develop more complicated power models with scheduling policies.

- 2) In our design, only a 3-MicroBlaze point-to-point network topology is integrated and no floating-point unit is added in the system. It would be interesting to integrate more processors and floating-point units.
- 3) Besides, some connections other than FSL, like network on chip, are also considerations for future work.
- 4) Configuring the Real time operating system for handling the tasks efficiently and performing the various applications like, audio-video processing, networking etc.

References

- [1] P.Huerta, J.Castillo, J.I.Martínez, V.López, "A MicroBlaze based Multiprocessor SoC" HW/SW Codesign Group, Universidad Rey Juan Carlos, 28933 Móstoles, Madrid Spain.
- [2] Wolf, W: The Future of Multiprocessor Systems-on-chip Proceedings of the Design Automation Conference (DAC'04) 2004. pp. 681-685.
- [3] Wolf, W: "Multimedia Applications of Multiprocessor Systems-on-Chip." Proceedings of the Design, Automation and Test in Europe Conference (DATE'05). 2005. pp. 86-89.
- [4] Stefan Aust, Harald Richter, "Energy aware MPSoC with Space-sharing for real time application" in ADVCOMP 2011, Lisbon, Portugal.
- [5] Roberta Piscitelli and Andy D. Pimentel "A High-Level Power Model for MPSoC on FPGA" IEEE International Parallel & Distributed Processing Symposium, 2011.
- [6] G. Magklis, G. Semeraro, D.H. Albonesi, S.G. Dropsho, S. Dwarkadas, and M.L. Scott, "Dynamic frequency and voltage scaling for a multiple clock-domain microprocessor". IEEE Micro, vol. 23, no. 6, pp. 62–68 , Nov/Dec 2003.
- [7] Yung-Hsiang Lu, Luca Benini, and Giovanni De Micheli, "Dynamic frequency scaling with buffer insertion for mixed working loads". IEEE transactions on computer aided design of integrated circuits and systems, vol.21, No.11, November 2002.
- [8] K. Moiseev, A. Kolodny and S. Wimer. "Timing-aware power-optimal ordering of signals". ACM Transactions on Design Automation of Electronic Systems, Volume 13, September 2008.
- [9] N. Zompakis, V. Tsoutsouras, A. Bartzas, D. Soudris and G. Pavlos, "Dynamic Frequency Scaling for MPSoCs based on Chaotic Workload Analysis," 23th International Conference on Architecture of Computing Systems 2010, Hannover, Germany, 2010, pp. 1-8.
- [10] V. Sundriyal, M. Sosonkina, F. Liu and M. W. Schmidt, "Dynamic Frequency Scaling and Energy Saving in Quantum Chemistry Applications," 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, Shanghai, 2011, pp. 837-845.
- [11] T. R. da Rosa, V. Larréa, N. Calazans and F. G. Moraes, "Power consumption reduction in MPSoCs through DFS," 2012 25th Symposium on Integrated Circuits and Systems Design (SBCCI), Brasilia, 2012, pp. 1-6.
- [12] J. C. Charr, R. Couturier, A. Fanfakh and A. Giersch, "Dynamic Frequency Scaling for Energy Consumption Reduction in Synchronous Distributed Applications," 2014 IEEE International Symposium on Parallel and Distributed Processing with Applications, Milan, 2014, pp. 225-230.
- [13] S. Höppner et al., "Dynamic voltage and frequency scaling for neuromorphic many-core systems," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.
- [14] "MicroBlaze Processor Reference Guide" Embedded Development Kit EDK 13.1
- [15] <http://www.xilinx.com/univ/xupv5-lx110t.htm>
- [16] Virtex-5 FPGA User Guide UG190 (v5.4), March 16, 2012, www.xilinx.com.
- [17] Digital Clock Manager (DCM) Module (v1.00.e) - Product Specification, DS485 March 22, 2010, www.xilinx.com.