International Journal of Engineering & Technology, 7 (4) (2018) 5640-5644

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.24688
Research paper

Designing a dual core processor system to act as multi rate filter using embedded design techniques

Mazin Rejab Khalil *

Technical Engineering College of Mosul , Northern Technical University, Mosul, Iraq
**Corresponding author E-mail: mazinrk56@gmail.com*

Abstract

Parallel computing represents an effective Technique for reducing program execution time especially with applications that necessitate repeated computations as multistage filtering systems. The paper aims to design a dual core processor system with message passing interface communication mode to be configured on Spartan 6E FPGAs slice. C language is used to program the configured system in a single instruction single data programming mode to act as multi stages filters. Embedded design techniques were used to construct the system hardware part, while the software part of the system is developed using the software development kit associated with Xilinx integrated software environment ISE13.2. A speed up in execution time is achieved by the designed system over single processor system. It is observed that the speed up ratio improves as the number of filtering stages increases.

Keywords: *Embedded Design Techniques; Message Passing Interface; Multi Processor Systems; Multistage Filtering; Single Instruction Single Data Programming Mode.*

1. Introduction

Dual processor systems can be used to make problem solution much faster than a single processor systems, for example, multi stage filtering operations usually require long series of computations per stage, more number of filtering stages necessitates longer computation time [1]. Dual processor systems usually use parallel programming techniques that are classified according to Flynn's taxonomy into four categories known as [2]:

- Single instruction /single data streams (SISD).
- Single instruction/ multiple data streams (SIMD).
- Multiple instruction / multiple threads (MIMT).
- Multiple instruction/ single data streams (MISD).

The chosen technique must be suitable to the type of application. Dual processor systems require a proficient communication media between processors in order to achieve speed improvement target. Two outstanding inter processor communication models are commonly used; shared memory model and message passing Interface (MPI) model. Shared memory mode requires sharing a specified part of the memory space between processors, memory coherency and atomic data treatment. In MPI mode, communication between processors is accomplished by sending and receiving messages using mail box. The coherency problem is treated carefully in both modes [3], [4].

The research includes constructing the hardware part of a dual processor system with mail box to act in MPI inter processor communication mode. The hardware part is developed using embedded design techniques (EDTs) and configured on Spartan 6E FPGAs slice. The configured system is programmed using C language in a single instruction single data (SISD) parallel programming fashion to perform multi-stages filtering operations. The performance of the configured system is tested by applying a random signal to it, displaying the output waveforms on chip scope window, computing the execution time and comparing it with time taken by single processor system to perform the same multi filtering task.

2. System design

The dual processor system design includes two steps. The first step implies constructing the hardware part of the dual processor system with its peripherals, while the second step is specified for the software development.

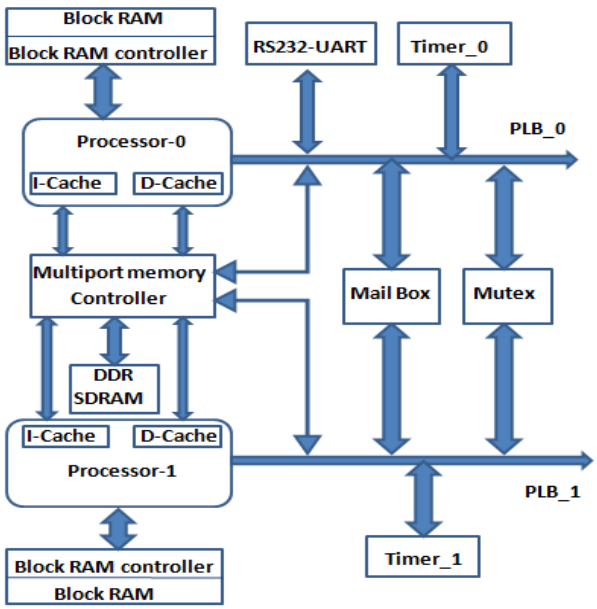


Fig. 1: Designed Dual Processor System Block Diagram.

2.1. Hardware development

The hard ware part block diagram of the designed dual processor system with MPI mode is shown in Fig.1. Two Microblaze soft core processors of 8.1 version are used, the floating point unit of each processor is enabled to operate with 32 bits floating point numbers [5], [6]. The DDR-SDRAM memory with 128MB capacity is shared by the two processors via multiport memory controller MPMC which provides access to the caches, system buses, as well as the DDR-SDRAM for one to eight ports where each port can be selected using ROUND-ROBIN arbitration algorithm [7], [8]. Processor local bus (PLB) of type 4.6 is accompanied with each processor to enable the processor to access its peripherals, it consists of control unit, address unit and READ/WRITE data unit [9]. The Universal Asynchronous Receiver Transmitter UART unit is used to receive the input signal to be processed in the form of asynchronous serial data transfer with configurable baud rate [10]. The timer unit is a timer/counter organized as two identical timer modules namely timer_0 and timer_1, each timer module has an associated load register that is used to hold the initial value for the 32 bits up/down counter, the counter when enabled start counting up or down according to the received command [11], the counter of the timer unit is used to count the number of clocks used during program execution time. Table 1 shows the address map of the designed system.

Table 1: The Address Map of the Designed Dual Processor System

Instance	Base Address	High Address
BRAM Controller	0x00000000	0x00001FFF
DDR-SDRAM	0x48000000	0x4FFFFFFF
UART	0x84000000	0x8400FFFF
mutex	0xC2400000	0xC240FFFF
Mail Box	0xCDE00000	0xCDE0FFFF
Timers	0x83C00000	0x83C0FFFF

2.2. Mail box operation

Inter processor communication is performed by the mail box whose operation in polling fashion depends on the registers listed in table 2 [4], [12]. The physical address of each register= Base address (as shown in table 1) + offset address(as shown in table 2). Writing to WRDATA register will result in transferring the data to RDDATA register in the other interface. Reading from RDDATA register will result in

Table 2: Mail Box Registers

Register Name	Offset Address	Access type	Function
WRDATA	0x0	Write	Write data address
RDDATA	0x8	Read	Read data address
STATUS	0x10	Read	Status flag for mail box
ERROR	0x14	Read	Error flags

Popping one value from the mail first-in first-out (FIFO) buffer.

2.3. Mutex operation

The mutex unit is used to provide exclusive access to a certain processor. Table 3 displays the bit definition and description of each bit of the write data register available in the mutex unit [13].

Table 3: Mutex Data Register Bit Definition

Bits	Name	Access	Function
0-22	Reserved	-----	reserved
23-30	CPUID	R/W	Unique processor ID number
31	Lock status	R/W	0=free,1=locked by CPUID

3. Software development

The software part is developed in the associated software development kit (SDK) where the application program is prepared, built , transformed into bit file and down loaded into the system memory for execution and debugging. The configured hardware of the designed dual processors system is programmed using C language to behave as multi levels low pass and high pass filters that are useful in wavelet transform calculations as in the form shown in Fig.2.The transfer functions of the selected low pass filter h_l and high pass filter h_h, Coiflets wavelet based Finite Impulse Response (FIR) filters, are [1], [14]:

h_l = [-0.0157 -0.0727 0.3849 0.8526 0.3379 -0.0727]

h_h=[0.0727 0.3379 -0.8526 0.3849 0.0727 -0.0157]

The low pass and high pass filter equations are respectively [1], [14]:

$$z_l(j) = \sum_{i=0}^{n} x[j-i].h_l(i) \quad (1)$$

$$z_h(j) = \sum_{i=0}^{n} x[j-i].h_h(i) \quad (2)$$

j= 0, 1, 2, 3….m
x = input signal.
m = No. of input signal samples.
n = No. of filter transfer function (h_l, h_h) coefficients.
z_l, z_h are the low and high frequency components of the input signal x respectively.

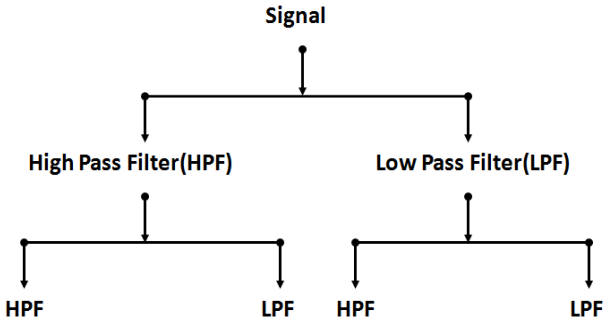


Fig. 2: Consecutive Multistage Filtering Diagram.

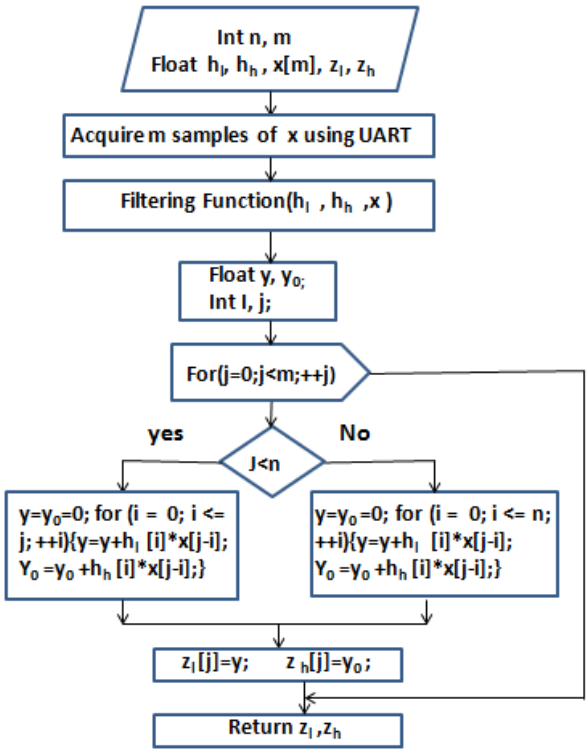


Fig. 3: The Flow Chart of the Prepared Filtering Function.

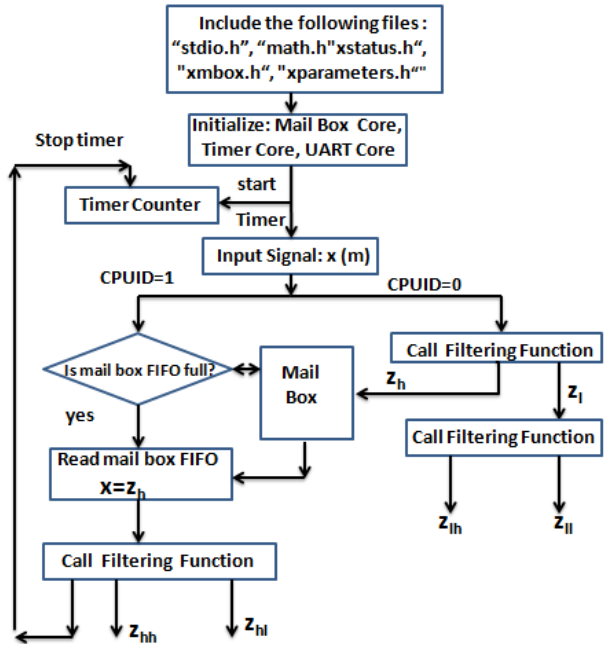


Fig. 4: The Flow Chart of the Multistage Filtering Program to be Executed on Dual Processor System.

Using (1) and (2), a filtering function is constructed to perform low pass and high pass filtering respectively. Fig.3 shows the flow chart of the constructed filtering function. The returned values of the filtering function are the low pass and high pass components of the signal namely the arrays z_l and z_h respectively.

The configured hardware is programmed by a developed software program to make the dual processor configuration act as multi-stages filtering system. The flowchart of the developed program is shown in Fig.4.

The designed system acquires the samples of the input signal to be analyzed and store it in the array x. processor-0 (CPUID=0) starts analyzing the signal x by calling the filtering function to perform first stage analysis while processor_1 (CPUID=1) is kept in wait state until receiving a message from processor_0. Once the arrays z_l and z_h are returned from the first call of the filtering function, a message is transmitted to processor_1 via the mail box to receive the array z_h. To perform second stage analysis, processor_0 calls the filtering function for the second time to analyze z_l obtaining second stage analysis of the signal x, namely the low-low pass component z_{ll}, and the low- high pass component z_{lh}. At the same time, processor_1 calls the filtering function for the first time to analyze z_h obtaining second stage analysis of the signal x, namely high-low pass component z_{hl} and high-high pass component z_{hh}. The timer counter is initialized by loading the load register with number 0x00000000, enabled with the start of sample acquisition and stopped at the end of the last filtering function call. The number of the counted clock pulses is proportional with the execution time of the program. A hyper terminal window and chip scope analyzer were used to display the computation results and the resulting waveforms.

4. Results

Fig.5 displays the first 250 samples of a random signal x that is generated in mat lab media, transformed to the configured dual processor system and displayed on chip scope window. The chip scope is accommodated to capture the data during the write phase operation as the PLB_wrDBus signal and display it using integrated bus analyzer. Chip scope window usually deals with integers while mat lab window can deal with floating point numbers therefore; each sample of the signal with floating point numbers is multiplied by 10000 in order to be displayed on chip scope window.

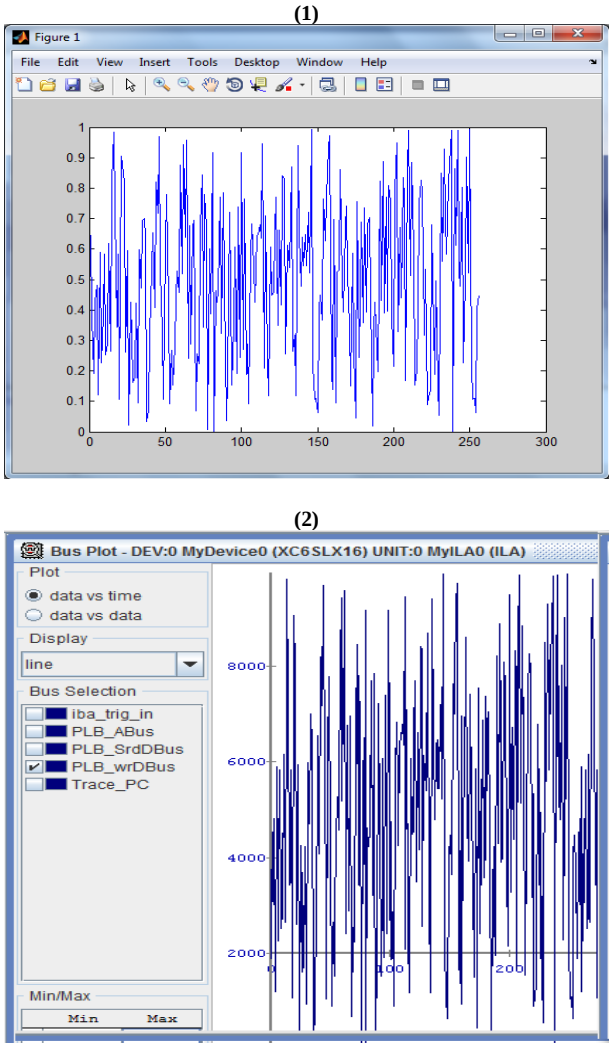
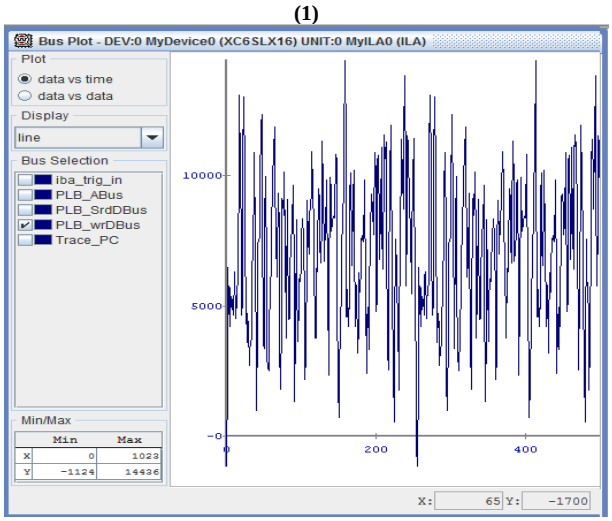


Fig. 5: Random Signal Under Test x: 1- Mat Lab Window, 2- Chip Scope Window.

Fig. 6 shows the output waveforms of the first stage decomposition; performed by processor_0, the low pass component z_l and the high pass component z_h

Results of second level filtering are z_{ll} , z_{lh}, z_{hl} and z_{hh}. The results z_{ll} and z_{lh} are computed by processor_0 and the result z_{hh} and z_{hl} are computed by processor_1. Fig. 7 displays the output wave forms of the second stage analysis signals namely, z_{ll} and z_{hh}. Fig. 8 shows the wave form of z_{lh} and z_{hl} signals.



(2)

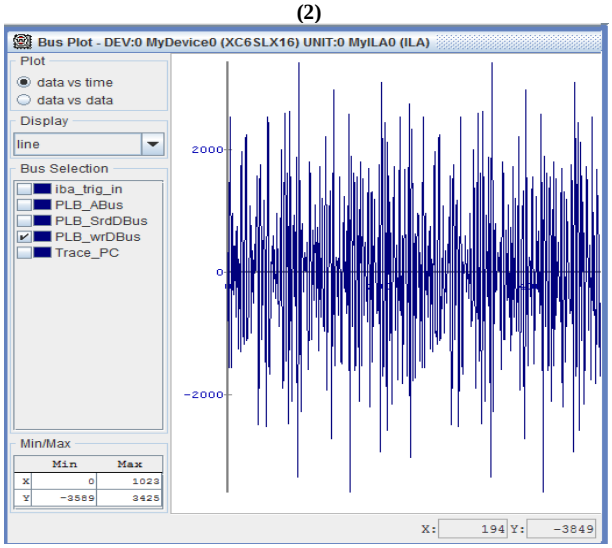
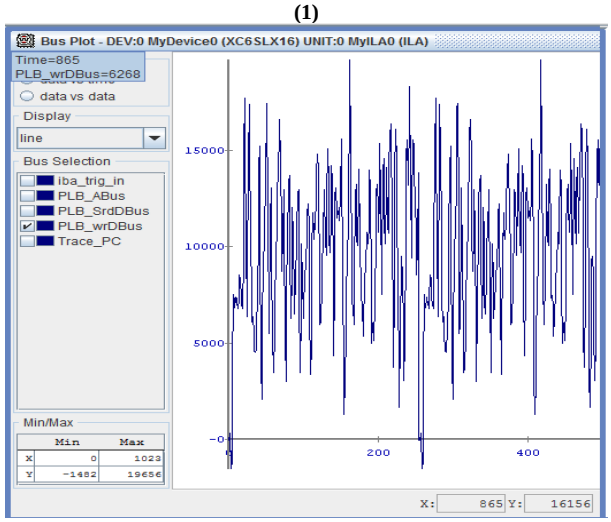


Fig. 6: First Level Decomposition: 1- Low Pass Component Z_l, 2- High Pass Components Z_h.

Fig. 9 shows the result of computing the first 32 samples of z_{hl} by the designed dual processor system displayed on hyper terminal window and the result of computing the same samples by mat lab displayed on mat lab window. The difference between the values of output samples computed by the dual processor system and the output samples computed by the mat lab is (0.0001-0.0002).
The execution time is greatly affected by the time needed for executing the filtering function, the number of filtering function calls

(1)



(2)

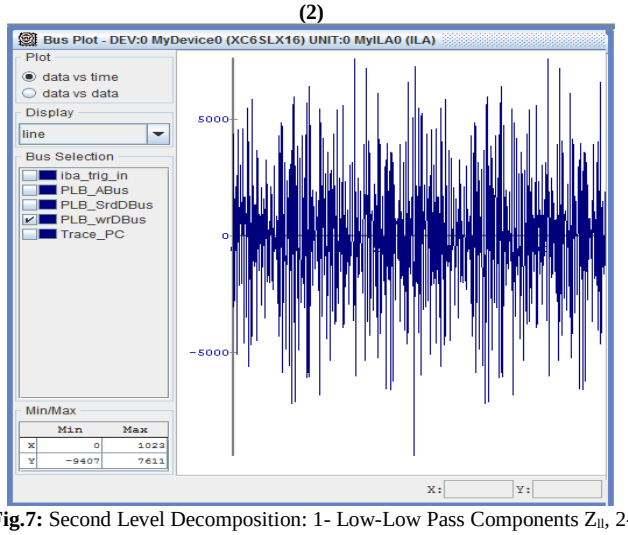
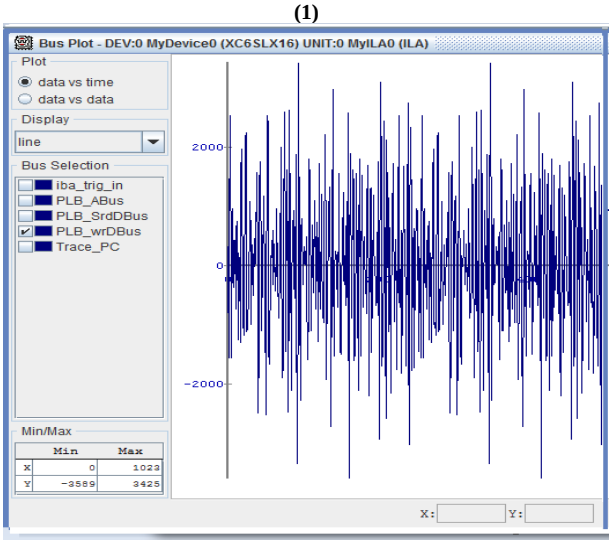


Fig.7: Second Level Decomposition: 1- Low-Low Pass Components Z_{ll}, 2- High-High Pass Components Z_{hh}.

(1)



(2)

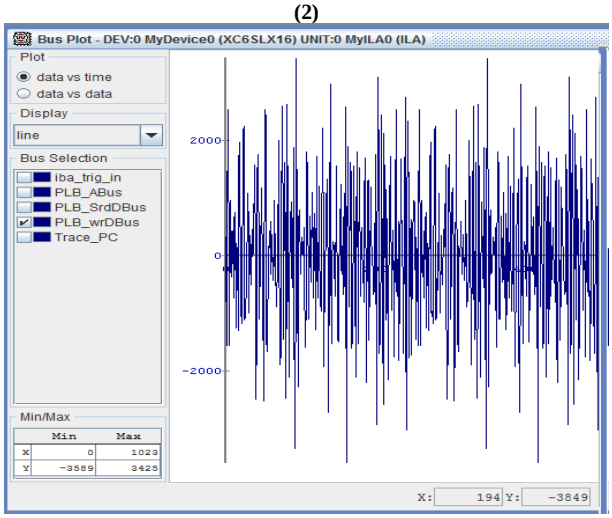
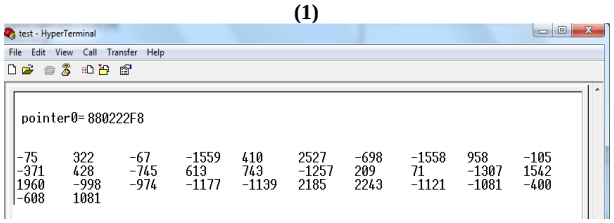


Fig. 8: Second Level Decomposition: 1- High-Low Pass Components Z_{hl}, 2- Low-High Pass Components Z_{lh}.

and the time required to exchange data between processors via mail box. Table 4 compares the number of filtering function calls per processor needed for each analysis stage in single processor and dual processor systems for three analysis levels. The execution time in dual processor system is proportional with the number of filtering function calls performed by each processor. Since processor_1 stay in wait case during the first call done by processor_0, this wait time will be considered in table 4 as a function call time. In second level analysis, for example, the number of calls in single processor system are three consecutive calls while the number of calls in dual processor system are four (three calls and one wait time), two consecutive calls are performed by each processor in parallel mode; as a result the time required for two level analysis in dual processor mode is less than that required in single processor mode.
The program execution time is calculated by (3).
Execution time = (count of timer counter-load register value)* PLB clock period (3)
The program execution speed improvement is measured when the program is applied on single and dual processor configurations. Let us define the speed up ratio (SUR) of dual processor system over single processor system as shown in (4).
SUR=dual processors system execution speed / single processor system execution speed (4)
Fig.10 shows the execution time improvement for one stage, two stages and three stages analysis. The speed up ratio for two stage analysis is (SUR= 1.48), the speed up ratio for three stages analysis is (SUR =1.66).

(1)



(2)

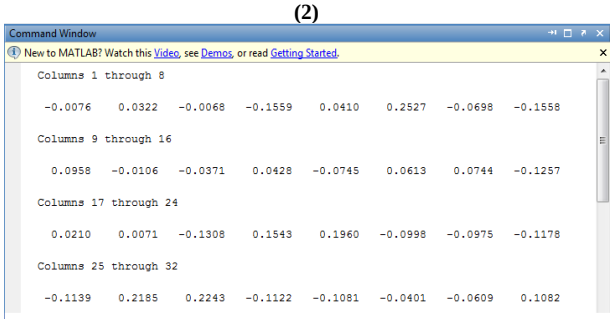


Fig. 9: Values of Selected Samples of Z_{hl}: 1- Hyper Terminal Window, 2- Mat Lab Window.

The speed up ratio improves as the number of analysis stages increases, this is due to the fact that the number of function calls in the dual processor system is decreased by (1) from the single processor system in the two stages analysis while it decreases by (3) in the three stages analysis and so on.

Table 4: Number of Filtering Function Calls In Single and Dual Processor Systems

No. of Analysis levels	No. of Function Calls in Single Processor System	No. of Function Calls in dual Processor System	Difference Column2-column3
1	1	1	0
2	3	2 by each processor	1
3	7	4 by each processor	3

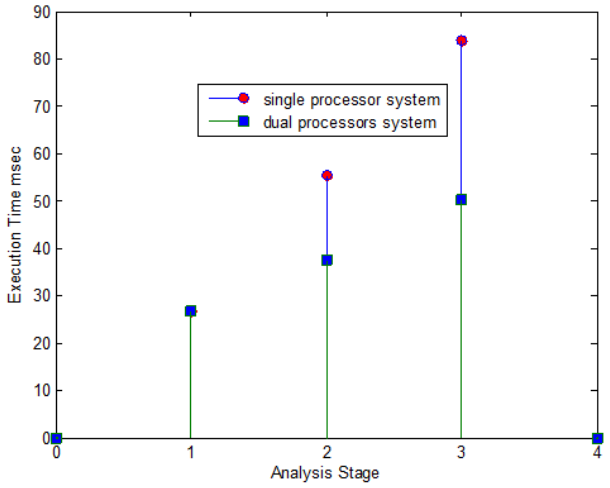


Fig. 10: Execution Time Improvement of Dual Processor System over Single Processor System.

5. Conclusions

A dual processor system is designed, implemented and downloaded on Spartan 6E FPGAs slice using embedded design techniques and programmed in SISD parallel programming mode to act as multi-stages filtering system. According to the obtained results the following conclusions are registered:

- The designed configuration when programmed presents a proficient methodology to perform multi- stages low pass and high pass filtering operations.
- The designed system constitutes the basic steps of discrete wavelet transform.
- The program execution speed in dual processor system is improved over single processor system as the analysis level increases; this is due to reducing the number of filtering function calls in dual processor system by splitting the number of calls between the two processors to be executed in parallel.
- The percentage error in the results obtained from the designed embedded configuration is ranging between (0.01-0.02) % when compared with the results obtained by using mat lab indicating high computation accuracy.

References

[1] C.L. Philips, J.M. Parr, E.A. Riskin, Signal, Systems, and Transforms, Pearson Prentice Hall, New jersey, USA, 2008.

[2] Roosta, H. Seyed, Parallel processing and parallel algorithms: Theory and computation, Springer, 2000. https://doi.org/10.1007/978-1-4612-1220-1.

[3] O. Green, Y. Birk, Scheduling directives: Accelerating shared-memory many-core processor execution, Parallel Computing, .40(2), (2014) 90-106. https://doi.org/10.1016/j.parco.2013.12.001.

[4] S. Ramesh, A. Maheo, S. Allen, MPI performance engineering with the MPI tool interface: The integration of MVAPICH and TAU. Parallel Computing, 77(2) (2018) 19-37. https://doi.org/10.1016/j.parco.2018.05.003.

[5] S.S. Omran, A.K Abdul- Abbas, Multi core processor for QR decomposition based on FPGA, International Journal of Engineering & Technology, 7(4) (2018) 2100-2105. https://doi.org/10.14419/ijet.v7i4.14517.

[6] Xilinx. Microblaze processor, reference guide, available on line: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/mb_ref_guide.pdf ; 2013, accessed Jan.2018.

[7] Xilinx. Multi-port memory controller, available on line: https://www.xilinx.com/support/documentation/ip_documentation/mpmc/v6_06_a/mpmc.pdf; 2013, accessed Jan 2018.

[8] H.Servat, J Labarta, H.C Hoppe, J Giménez, A. Pena, Understanding memory access patterns using the BSC performance tool, Parallel Computing, 78, (2018), 1-14. https://doi.org/10.1016/j.parco.2018.06.007.

[9] Xilinx. Processor local bus, available on line: https://www.xilinx.com/support/documentation/ip_documentation/plb_v46.pdf; 2010, accessed Jan 2018.

[10] Xilinx. Xps uart lite, available on line: https://www.xilinx.com/support/documentation/ip_documentation/xps_uartlite/v1_02_a/xps_uartlite.pdf; 2011, accessed Feb.2018.

[11] Xilinx, Inc. (2010, Apr), XPS, timer/counter, DS573, available on line:http://www.xilinx.com/support/documentation/ip_documentation/ds573.pdf, accessed Feb.2018.

[12] Xilinx. Logicore IP mailbox, available on line: https://www.xilinx.com/support/documentation/ip_documentation/mailbox/v2_1/pg114-mailbox.pdf; accessed may2018.

[13] Xilinx. Logicore IP mutex, available on line: https://www.xilinx.com/support/documentation/ip_documentation/mutex/v2_1/pg117-mutex.pdf; 2015, accessed Feb.2018.

[14] R. Ramos, B. Valdez-Salas, R. Zlatev et al, The Discrete Wavelet Transform and Its Application for Noise Removal in Localized Corrosion Measurements, International Journal of Corrosion, (2017), 7.