J. Basic. Appl. Sci. Res., 2(3)2369-2373, 2012
© 2012, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

PARALLEL PROCESSING ON SPARTAN-3

Moch. Rif'an*, Arief Abd. Manan, Ponco S.

Electrical Department, Faculty of Engineering. University of Brawijaya, Malang, Indonesia

ABSTRACT

This paper concerns on design multiprocessor embedded on Field Programmable Gate Array (FPGA) for calculator implementation. Processor will be design with two thread and one control unit. This multi-processor works with Front and Back Processor, first processor as front and the other as back processor. Multiprocessor system designed in SMP system and Barrel system. Result of experiment are that processor need 8,9 second for single processor, but multiprocessor need 6,3 second for SMP system and 5,3 second for Barrel system.
KEYWORDS: FPGA, Embedded system, multi processor.

INTRODUCTION

Multi processor technology is implemented on personal computer at the first time in middle of ninety. A technology in multiprocessor called multi threading is a famous technology recent year. To achieve modern technology, it need along time research. Intel has claims that by hyper threading, system performance can rise about 15-30%. This technology is used in atom and i7 processor.

Many researchers are helping by FPGA. Design can embed in there to do some testing. Then, final design can be fabricated. This research tested a parallel processing Spartan-3 that aimed to make multiprocessor faster than single processor.

MATERIALS AND METHODS

Goal of this research is giving illustration about computation performance comparing between single processor and multiprocessor.

Design Specification

Multiprocessor system designed with specification:

1) Have two thread processor
2) Have one control unit and 20 bytes sharing memory.
3) Processor designed to calculate positive imaginer biner number on adder, subtraction and multiplication operation.
4) Design is embedded in FPGA XILINX SPARTAN XC3S500E type.

System design

Figure 1 shows the block diagram design of parallel processing.

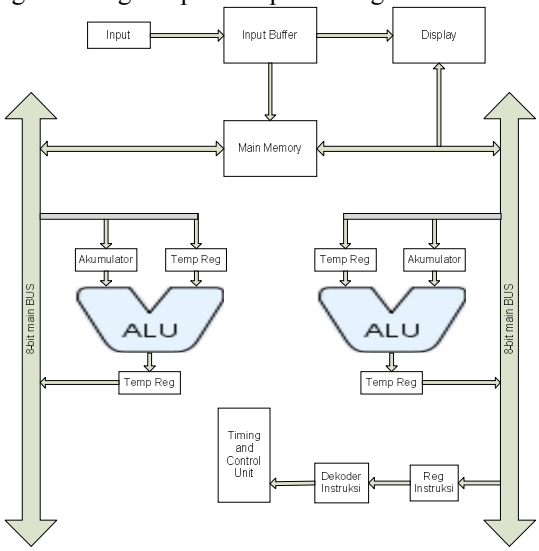


Figure1. Block Diagram Design

***Corresponding Author:** Moch. Rif'an, Electrical Department, Faculty of Engineering. University of Brawijaya, Malang, Indonesia.
Email: rifan@ub.ac.id

Input can be data, adding or subtracting command. Input buffer store all number and command, they will be transferred to main memory. Processor will process command and data.

Multiprocessor Process

When two or more people work together, they need coordination. Coordination needed so they can do their job optimally.

In this processor it is used front and back processor system. Front and back processor system is coordination system with task share for multiprocessor. A front processor control all back processor (back processor can be more than one processor). Front processor control where are the task that will be done by each back processor. Besides, front processor can handle simple task. Back processor does only the tasks given by front processor. Sometime result from back processor are transferred to the front processor, some time not. Figure 2 show front and back processor system design.

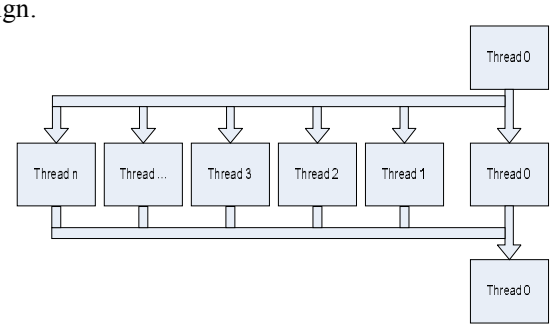


Figure 2 Simple Front and Back Processor

Figure 2 shows that thread 0 (first processor) as front processor will distribute tasks to another processors. This multiprocessor system adopted by Barrel type processor system. Result from back processor can be accepted by another back processor and be forwarded. It is done to make same load level for each processor. Switching the task between back processor are controlled by front processor, it must be not easy programming. This kind of coordination is adopted from SMP system (Symmetric multiprocessing) shown in figure 3.

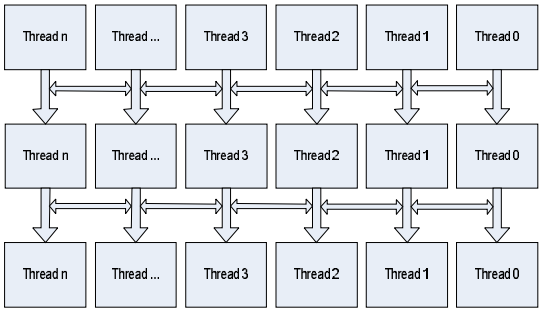


Figure 3 Front and Back Processor

Microprocessor model design

Microprocessor is designed have two thread and one control unit. First, design must be assume that all of thread can work as one processor (assumed there is only one processor), because every multiprocessor should work as a single processor system. Both First and second processor have to do processes individually, arithmetic or other process. Processing system for this single processor shown in fig. 4.

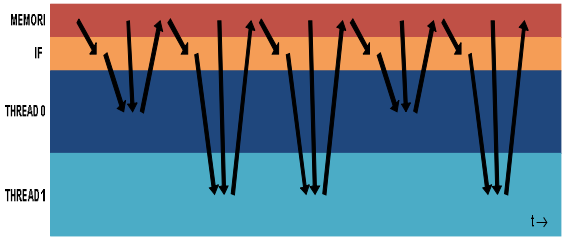


Figure 4 Instruction processing for single processor

Thus processor designed has basic instruction for single processor. Those instructions (decided by author) are:

- NOP : (instruction to do nothing)

- MVI : input data directly to the register
- INC : (add certain register by one)
- ADI : (adding immediately operation)
- SUI : (Subtraction immediatly operation)
- MULI : (multiplication immediatly operation)

Step by step instructions above are as follows:

- NOP Instruction

NOP is instruction to command processors (first and second processor) to do nothing. Flowchart for NOP instruction is shown as figure 5.

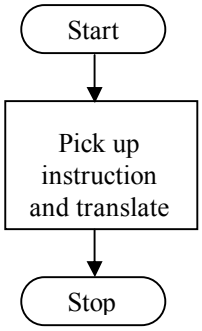


Figure 5 Flowchart for NOP Instruction

Figure 5 can be said as follows:
1) Pick up instruction from memory (Fetching instruction)
2) Translate instruction (Instruction Dispatching)
3) Instruction ended.

- INC Instruction

INC instruction is an instruction that command processor to add specific register by one. There are two registers in this processor, Accumulator1 and accumulator2 register. Flowchart of this INC instruction is shown in figure 6.

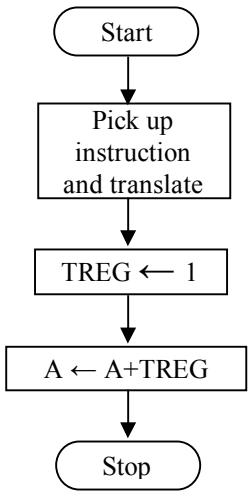


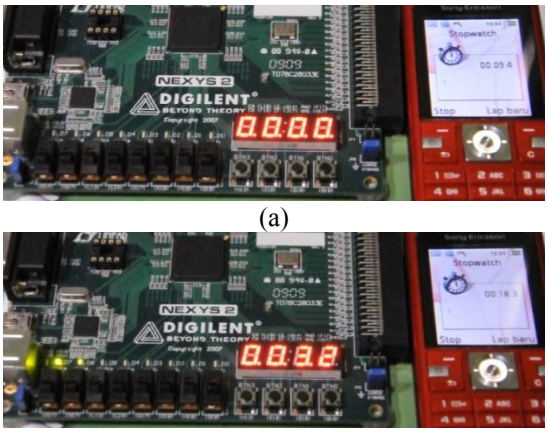
Figure 6 Flowchart of INC instructions

Step by step in Figure 6 can be explained as follows:
1) Pick up instruction from memory (Instruction Fetching)
2) Translate instruction (Instruction Dispatching)
3) Give TREG with number one, set ALU for adding process
4) Move data result from adding process to accumulator.
5) Instruction ended.

There are only two registers in this processor, instruction can be write as INC A1 or INC A2.

TEST, RESULTS AND DISCUSSION

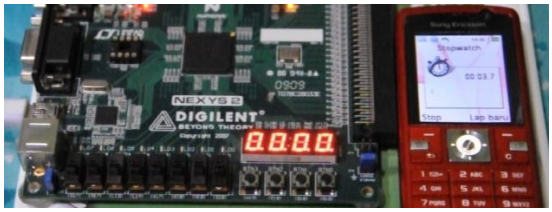
There are three different testing. Testing is carried out by giving group of command and result is shown at seven segment. First testing is done to find performance (time execution) for single processor, second testing for SMP dual processor and third testing for Barrel dual processor. Result of our testing are shown in figure 7,8 and 9.



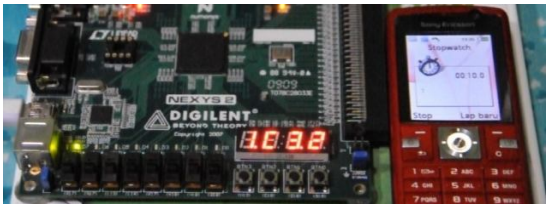
(a)

(b)

Figure 7 (a) start program for single processor (b) end program for single processor



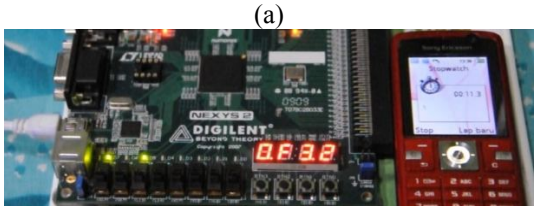
(a)



(b)

Figure 8 (a) start program SMP dual processor (b) End Program SMP dual processor

(a)



(b)

Figure 9 (a) start Program for Barrel dual processor (b) end program for Barrel dual Processor

Result of time execution test are as follows:

a) According of single processor testing, we get start time at 9,4 second, and end time at 18,3 second, so we get execution time 8,9 second (18,3 – 9,4)

b) According of SMP dual processor test, we have start program at 3,7 second and end time at 10,0 second, so we get execution time of 6,3 second (10,0 – 3,7)
c) According of SMP dual processor test, we have start program at 6,0 second and end time at 11,3 second, so we get execution time of 5,3 second (11,3 – 6,0)

CONCLUSION

In conclusion, system with multiprocessor can process several tasks faster than single processor. Although has the same number of processor, Barrel type of multiprocessor process tasks is faster than SMP type.

BIBLIOGRAPHY

[1] Altera Corporation. 2009. *Altera Stratix IV EP4SGX230 FPGA on a PCB*. California: Altera Corporation.

[2] AMD.com. *Thuban Architecture* http://www.AMD.com/thuban_architecture.html, tanggal akses 1 Mei 2010.

[3] Amdahl, G. (April 1967). *The validity of the single processor approach to achieving large-scale computing capabilities*. In Proceedings of AFIPS Spring Joint Computer Conference, Atlantic City, N.J.: AFIPS Press

[4] Chandra, Rohid dkk. 2001. *Parallel Programing in OpenMP*. London : Morgan Kaufmann Publishers.

[5] Computer_Histori. 6600 reference manualhttp://ed-thelen.org/comp-hist/CDC-6600-R-M.html#P3-5, tanggal akses 7 Mei 2010.

[6] CPU-World. *DIE_Picture*http://www.cpu-world.com/info/die_pictures.html, tanggal akses 1 Mei 2010.

[7] Intel.com. 2010. *Hyper-Threading*. http://www.intel.com/technology/platform-technology/hyper-threading/index.htm, tanggal akses 1 Mei 2010.

[8] MIPS Technologies, Inc. 2005. MIPS32 Archithecture for Programmers Volume II: The MIPS32 Instruction Set. California: MIPS Technologies.

[9] Nurmi, Jari. 2007. Processor Design "System-On-Chip Computing for ASICs and FPGAs". Finland : Springer.

[10] Smotherman, Mark. 2005. *Histori of Multithreading*. Washington: Univ. of Washington.