Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July, 2021:1333 – 1339

CACC-ComputerArchitectureforCombinatorics Computation

Francis John Magallanes [a], Ralph Joshua Vasquez [b], John Matthew Vong[c] Dino Ligutan[d], Cesar Llorente[e]

[a] Department of Electronics and Computer Engineering, De La Salle University, Manila, Philippines
[a]francis_magallanes@dlsu.edu.ph, [b]ralph_vasquez@dlsu.edu.ph, [c]john_matthew_vong@dlsu.edu.ph, [d]dino.ligutan@dlsu.edu.ph, [e]cesar.llorente@dlsu.edu.ph

Abstract

Combinatoricshasbeenaroundformanyyears now and it has been utilized for multiple areas inmathematicssuchasprobabilityandstatistics.Thispaper aims to create a system that can computecertaincombinatoricsproblemsmainlyfocusingoncombination, permutation, and factorial problems. Thesystemimplementedisalsoequippedwithmultiplecomponentsthat would aid in the computation of theoutput which include the main components of the MIPSarchitectureaswellascomponentsthatareimplemented for the computation of the combinatoricsalgorithms such as multiplication and division unit. TheimplementedMIPSarchitecturealongwiththeDatapath and control units all work together to create asystem that may manipulate data and run an algorithmtoprovide a desirable output.

.

Keywords: -Combinatorics,RISCArchitecture,ComputerArchitecture, MIPS architecture modification

1. Introduction

Factorial is a method in mathematics whichis the product of all the positive integers that is lessthanorequaltoagivenpositiveintegerandisrepresentedbythatnumbertogetherwithanexclamation point. A perfect example is when a givenis4!,itneedstomultiplystartingfrom1upto4,1x2 x 3 x 4, and the product of these numbers is theresultofthatfactorial[1].

Permutations are the number of ways a setcanbe arranged into some sort of sequence whilecombinations,ontheotherhand,aresimplyanunorderedselectionofmemberswithinaset[2].Togetherthesetwoac tsofarrangingorselectingmembers of a set would be a form of a greater entityknownascombinatoricswhichisessentiallythestudyoffinitediscretestructures.

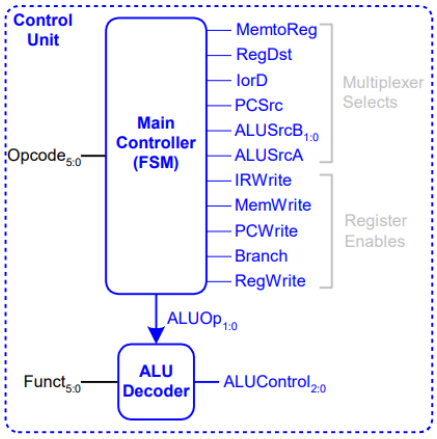

Figure1.ControlUnitofaBasicMulti-CycleMIPS

The uses of combinatorics would be able to open calculations whose applications would be able to be used to solve problems which would utilize combination or permutation instructions to provide solutions to a problem.

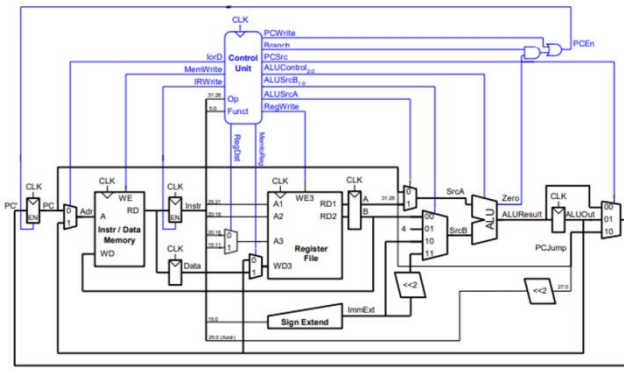

Figure2.CompleteDiagramofaBasicMulti-CycleMIPS

Another application which would be helpful is utilizing factorials and combinatorial reasoning to determine the number of iterations for certain computations or programs running as well as the number of possibilities that may exist from certain other systems such as a lock combination. It can answer questions such as "How many possible lock combinations there may be given the constraints should there be any? "

In this study, a computer architecture based on MIPS [3] is modified to perform combinatorics computation. The modification of the Instruction Set Architecture is due to the additional instructions that will handle the combinatorics computation. The block diagram of the control unit with the additional control signals in the control word is shown in Figure 1. Changes to the MIPS Datapath is depicted in Figure 2.

The concept of the project aims to provide a computer architecture that is based on the mathematical concept called Combinatorics, also known as combinatorial mathematics. In this field, it allows the program to count and handle large amounts of numbers that can be treated in solving probabilities and combinations depending on a given set of data.

The general objective of this project is to develop a specialized computer that will be able to perform operations in the field of combinatorics. The following are the specific objectives of project:

1. To develop an instruction set architecture that can perform factorial, permutation, and combination operation..

2. To develop the Data path and control unit that can execute combinatorics operations such as permutation and combination operations.

3. Todevelopahardwarethatwilldothefactorialoperationefficiently.

4. To develop a hardware that can do themultiplicationanddivisionoperations.

5. Toverifythecorrectnessoftheimplementedspecializedcomputer.

2. Statement of Contribution/Methods

Application Area

Factorial instruction integrated in the ISA of MIPS that do not include loops, thus speeding up the computation process. To provide support, a Multiplier unit is modeled and added to the ALU.

Xilinx Vivado 2019 will be then used to simulate and synthesize the design. The simulations will be executed to prove that the system is working as intended and there is no conflict on the design.Tomeetthedesiredfunctionalityoftheapplication, three key components must be taken intoconsideration which are the multiplication, division,and factorial units. These three processes will be thekeyinstructionsfortheapplicationtobeimplemented and the algorithm to which they wouldbeimplementedwouldalsobetakenintoconsideration.Forthemultiplicationprocess,theprocessisstraightforwardwithtwoinputsbeingmultipliedtoeachotherwiththesimpleprocessshownas:

output =inputA*inputB

Forthedivisionprocess,thealgorithm is shown in Figure 3, which is essentially a repeated subtractionprocesswithinaloopandtheprocessisshownas:

Loop:

InputA = InputA - InputBCounter++;

Exit when InputB> Input AReturnCounter.

Fig.3.Pseudocodefortherepeatedsubtractionversionofdivi
sion

Lastlyforthefactorialprocess,twoalgorithmswere proposed which are implementingthe factorial process as an iterative method and as aninstruction.

Instruction Set Architecture Modifications

Table 1 shows the instructions that can be executed by the designed system together with its opcode and its instruction format (based on the MIPS instruction format). The instructions multiply unsigned(multu), divided unsigned (divu), and factorial (fac) are included in the ISA.

PermutationTesting

Thefirsttestthat was done to verify thecreatedalgorithmisbyperformingapermutationtestingproblemthat asks to choose a group of 2 outof 12 possible individuals which would have the formof12P2or12 permutation 2. The testing processfollowed the permutation formula which is denotedby:

$$nPr = \frac{n!}{(n-r)!}$$

Test codes in assembly language for MIPS will verify the functionality of the Permutation and Combination functions.

Speedup and Statistical Analysis

Two computer programs will be used to determine the speedup of the execution. The average Cycles Per Instruction (CPI) is determined based on the following equation,

$$speedup = \frac{Loop\ Execution\ Time - based\ factorial}{Instruction\ Execution\ time - based\ factorial}$$

As for the statistical analysis, skewness will be used to describe the distribution of the CPI in the instructions executed. Skewness will describe whether most of the instructions executed have lower CPI or higher CPI.

3. Results, Discussion and Conclusion

Figure 7 shows the timing diagram of the permutation computation which executes the constructed instruction as seen in the Computer Performance Testing part. The permutation problem has the variables 12 and 2 which were stored at registers 4 and 5, respectively.

The difference of these two inputs were solved and stored at register 6 which computes the factorials of the variables inside registers 4 and 6 wherein they were stored at the same location to be overwritten by its factorial values.

Figure 8 shows the timing diagram of the output of the combination operation. The inputs 10 and 3 are stored in registers 16 and 17 respectively and their difference in register 11 then their factorials are computed for and stored in the same registers before applying the multiplication and division steps denoted by the combination operation formula and has produced an output of 120 in decimal or 00000078 in hexadecimal which matches with the expected output of the program without having any errors in any of the steps from input to processing to output. The algorithm was run in 54 program counts.

Figure 9 and figure 10 shows the timing diagram of the respective assembly program and its output. The output is stored in memory at the memory location 512,513,514, and 515. The output is stored in a big-endian format. Thus, the output of both assembly programs is 0x00375F00 or 3628800 in decimal. This output follows the expected result of "10!".

It is also shown in the figures that the loop-based factorial program finished its store instruction at 1,965 ns while the instruction-based factorial program finished its storeinstruction at 175 ns.

Given that the program will start at 10 ns because of the reset, the execution time of loop-based factorial

program is 1,955 ns while the execution time of the instruction-based factorial program is 165 ns. With this execution time of both assembly programs, the calculated speedup is 11.85 based on(4). Other information such as the number of instructions executed, and the average CPI is also determined, and it can be seen in table 4.

Table 4 shows that the instruction-based program has a higher average CPI of 4.125 than the loop-based program which has an average CPI of3.62. The loop-based factorial program has a mean of 3.62, a mode of 4, and a sample standard deviation of0.52.This gives the value of skewness of the distribution as -0.73. The computed skewed value indicates that the distribution of CPI is moderately negatively skewed. This means that the tail of the distribution is at the left of the distribution and the mean will be less than the mode which is true based on the computed values [9]. As for the instruction-based factorial program, it has a mean of 4.125, a mode of 4, and a sample standard deviation of 0.5. The skew of the distribution is 0.25. The computed skewed value indicates that the distribution of the CPI is symmetrical even though it is notobvious in the histogram. This means that the mean of the distribution is close to the mode of the distribution which is true based on the computed values [9]. The computed skew values for both distributions explain why the average CPI of the loop-based factorial assembly program is lower than the instruction-based factorial assembly program.

The distribution of CPI for the loop-based factorial program is negatively skewed. This means that the mean is lower than the mode which in this case, the mode is four.

As for the instruction-based factorial program, it has a symmetric distribution. This means that the mean is close to the mode which in this case, the mode is four. With these two things being said, the shape of the distributions affected the mean for both test cases. Since the shape of the distributions is not the same, it can be said that comparing the average CPI of the test cases is not a good measure to determine whether there is an increase in performance.

In conclusion, the proponents developed a modified MIPS architecture to perform operations in the field of combinatorics. The developed computer follows the multicycle MIPS implementation. Additional components were also added to the multicycle MIPS architecture so that the specialized computer can do the multiplication, division, and factorial. The developed ISA focuses mostly on the arithmetic operations so that the combinatorics operations such as combination and permutation can be performed. The developed ISA also follows the MIPS instruction format and most ofits opcode is based on the MIPS instruction set. Results of the analysis shows that the developed specialized computer can do factorial, permutation, and combination operations with no error from the expected output. The analysis shows that using the factorial instruction developed by the designers, there will be an almost 12 times faster execution time of the factorial. It is also said that the instruction for the factorial operation is not dependent on the operand and thus, the execution time will remain constant even if the operand changes.

In speedup and statistical analysis, there is almost 12 times speedup when using the instruction-based factorialassembly program in performing factorial operation through the speedup metric.

This can be clearly seen in the significant difference of the execution time between the two programs. This speedup can also increase as the operand for factorial operation increases since the loop-based factorial operation is dependent on the operand while the instruction-based factorial operation is not dependent on the operand. Also, the comparison between the average CPI of the instruction-based factorial program and the average CPI of the loop-based factorial program is not used since the distributions of the two programs are not the same and it can be misled to say that the loop-based program is faster than the instruction-based program using the average CPI as a metric.

References

[1] Britannica, T. Editors of Encyclopaedia (2013, October 18). Factorial. Encyclopedia Britannica. https://www.britannica.com/science/factorial

[2] BYJU's (n.d.). Combinatorics. https://byjus.com/maths/combinatorics/#:~:text=Applications%20of%20combinatorics&text=Communication%20networks%2C%20cryptography%20and%20n etwork,Scientific%20discovery

[3] J.D. Dumas, Computer Architecture: Fundamentals and Principles of Computer Design, CRC Press, 2006.

[4] TutorialsPoint (n.d.). VLSI Design - VHDL Introduction. https://www.tutorialspoint.com/vlsi_design/vlsi_design_vhdl_introduction.htm

[5] Childers, B. (2017, April 11).ch4-slides-multicycle [PowerPoint slides]. School of Computing and Information, University of Pittsburgh. http://people.cs.pitt.edu/~childers/CS0447/lectures/le ct-multicycle.pdf

[6] Capkun, S. and Gürkaynak, F.K. (2014). Multi-cycle MIPS Processor [PowerPoint slides]. Design of Digital Circuits 2014.

[7] https://syssec.ethz.ch/content/dam/ethz/special-intere st/infk/inst-infsec/system-security-group-dam/educati on/Digitaltechnik_14/21_Architecture_MultiCycle.pdf

[8] Stephanie, G. (2020, September 4). Pearson's Coefficient of Skewness. Statistics How To. https://www.statisticshowto.com/pearsons-coefficient-of-skewness/.

[9] Dugar, D. (2020, July 18). Skew and Kurtosis: 2 Important Statistics terms you need to know in Data Science. Medium. https://codeburst.io/2-important-statistics-terms-you- need-to-know-in-data-science-skewness-and-kurtosis

Table 1. Additional Instruction to MIPS ISA

Instruction	Mnemonic	Opcode	Funct	Comments
Multiplyunsigned $rsand$rt	mltfu$rs,$rt	000000	011001	The$rdfieldsettozero
Divide unsigned$rs and $rt	Divu$rs,$rt	000000	011011	The$rdfieldsettozero
Factorialthe$rs	Fac$rs	000000	011100	The$rd, $rtfieldsettozero
Move from HI reg to $rd	Mfhi$rd	000000	010000	The$rs, $rtfieldsettozero
Move from LO reg to $rd	Mflo$rd	000000	010000	Thersrt fieldsettozero

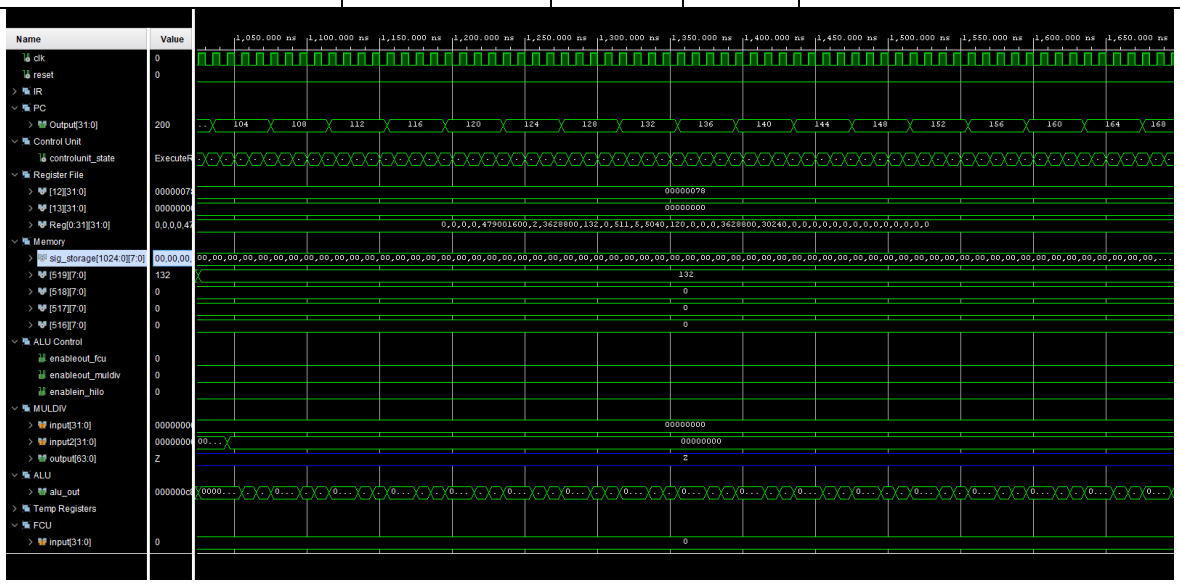

Figure 7. Timing diagram of the permutation operation

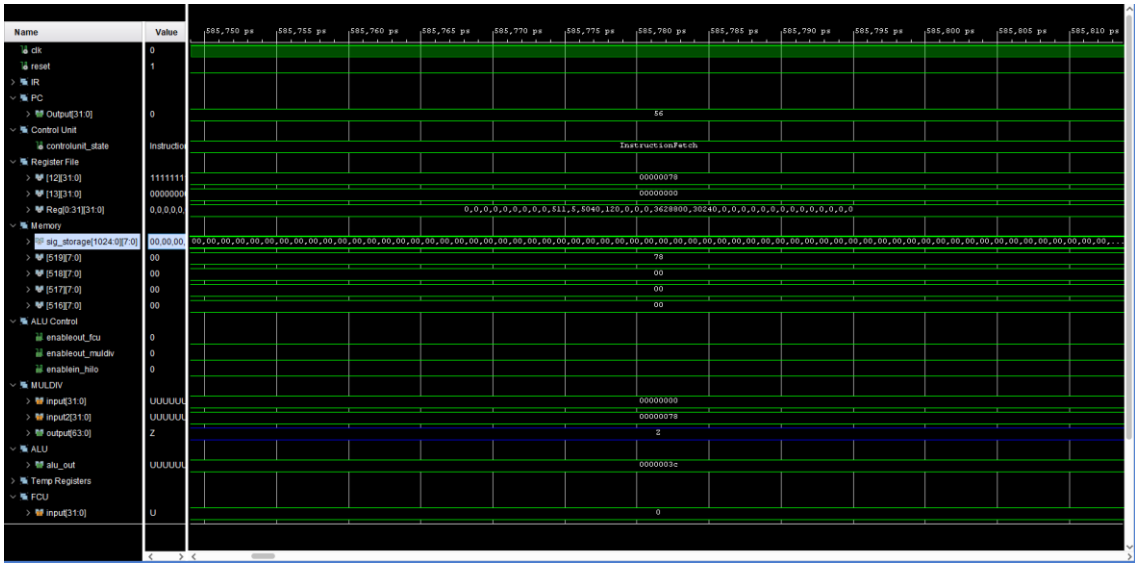

Figure 8. Timing diagram of the combination operation

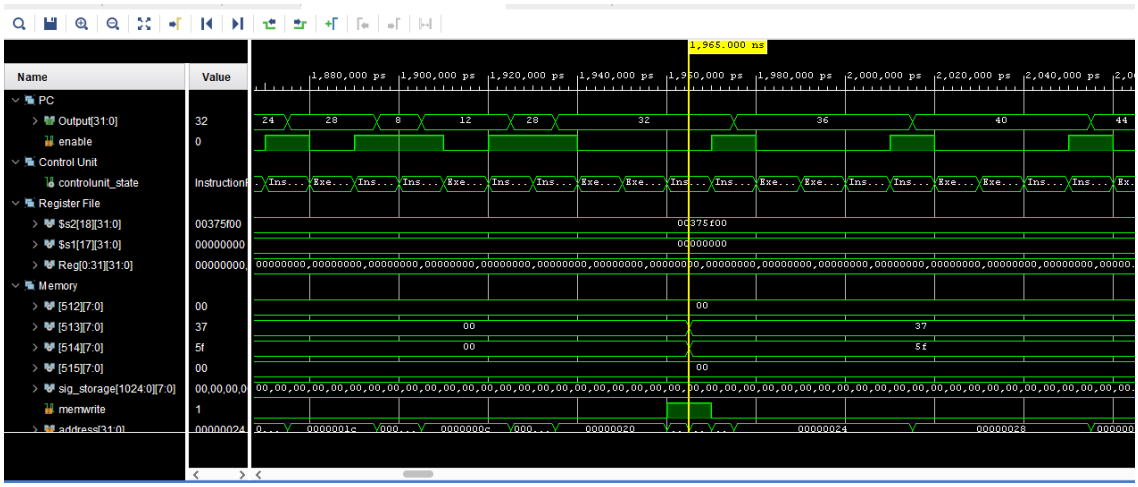

Figure 9. Timing diagram of the loop-based program which shows the output at memory location 512, 513,514, and 515

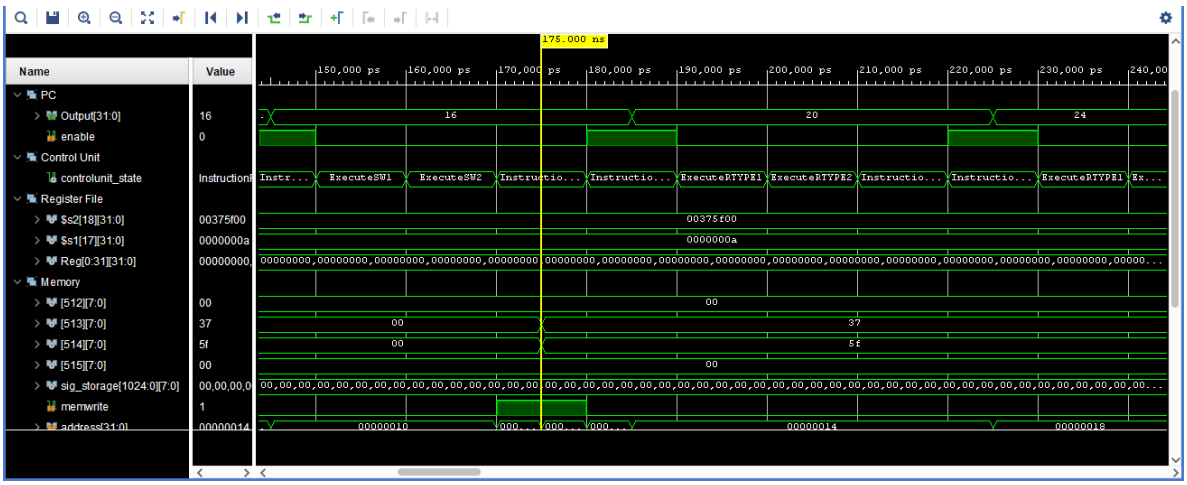

Figure 10. Timing diagram of the loop-based program which shows the output at memory location 512,

513,514, and 515

Table 4. Comparison of execution times, number of executed instructions and average CPI of the loop-based program and the special instructions for combinatorics computation

Implementation	ExecutionTime	NumberofExecutedInstructions	AverageCPI
Loop-basedprogram	1,955ns	54	3.62
Instruction-basedprogram	165ns	4	4.125