Dual Core Capability of a 32-bit DLX Microprocessor

DEAN MICHAEL B. ANCAJAS, ANASTACIA P. BALLESIL, JOHN RICHARD E. HIZON,
EUGENE A. OPELINIA, JOY ALINDA P. REYES, ALLAN GORDON L. SEPILLO,
WINSTON A. SUMALIA, WILSON M. TAN
Intel Microprocessors Laboratory,
Department of Electrical and Electronics Engineering
University of the Philippines, Diliman

***Abstract*— We report an implementation of a 32-bit DLX Microprocessor capable of operating in a dual core environment. The processor was modified for it to be capable of operating atomic instructions, a requirement in a dual core environment. The dual core environment was simulated using a similar core acting as a pseudo slave core. The resulting processor can then be interfaced with another instance of the same processor to function as a dual core processor. It can also be interfaced with a DSP co-processor that is compatible with the handshaking protocols of the processor.**

The resulting implementation yielded a power reduction of 17.9%(due to a more efficient register file) and an area overhead of 23%(due to additional blocks needed for dual core capability) compared to previous DLX implementations of the laboratory.

***Index Terms*— dual core, multithreading, multicore, DLX processor**

I. INTRODUCTION

THE goal in processor design has always been higher performance for smaller area and lesser power consumption. Increasing the clock rate has been the traditional and popular approach used in achieving acceptable compromises between the three. While attainable clock rates may still increase in the near future by advances in processor design and materials engineering, performance gains from such advances alone will hardly be able to keep up with Moore's Law and the industry's need for ever increasing processing capability. Even today, Intel's fastest single core processor never exceeded the 4 GHz barrier. One of the approaches being explored as a workaround to this problem is multithreading, specifically, multicore processor implementations. They offer higher performance per watt ratio than traditional single core processors. Multicore processors are able to execute multiple independent programs simultaneously by using multithreading.

A Dual Core Capable DLX(DCC DLX) processor was implemented using *Verilog* Hardware Description Language at the Register Transfer Level using standard cells from a 0.25um CMOS process. The design followed the SMP architecture and was implemented in a shared memory environment.

A. DLX Microprocessor

DLX (pronounced as "Deluxe") is a model architecture developed by John L. Hennesy and David A. Patterson and is intended to be used as an instructional tool in the field of Computer Architecture. Believed to be the world's second polyunsaturated computer, its design philosophy is very similar to a group of *Reduced Instruction Set Computer* (RISC) designs.

DLX is a clean and simplified version of *Microprocessor without Interlocked Pipeline Stages* (MIPS) architecture. Its load/store architecture is characterized by its pipeline efficiency, the use of a simple load/store instruction set, and its efficiency as a compiler target.

B. Taxonomy of Parallel Architectures

The idea of using multiple procesors both to increase performance and to improve availability dates back to the earliest electronic computers. About 40 years ago, Flynn[5] proposed a simple model of categorizing all computers that is still useful today.[10] In this classifications, the DCC DLX is classified under *Multiple instruction streams, multiple data streams processor(MIMD)*. In this setup each processor fetches its own instructions and operates on its own data. MIMD computers exploit thread-level parallelism, since multiple threads operate in parallel. Famous examples of MIMD computers are Intel's Core Duo and AMD's Athlon 64 X2.

C. Atomic Instructions

Atomic instructions atomically access and update one or more memory locations. Being atomic means that the set of actions done is considered a single action. Atomic Instructions are required in multicore processors in order to facilitate variable locking mechanisms and to insure that at any one time only one processor is accessing a shared variable[7]. The following atomic instructions are added to the DLX Instruction set.

```
CompareAndSwap:CAS(address,old,new)
```

1-4244-1272-2/07/$25.00

```
    if old == (address)
       (address) <= new
FetchAndAdd:FAA (k,loc)
    (loc) <= (loc) + k
TestAndSet:TAS (reg)
    if (reg) == #unlocked
       (reg) <= #locked
Loadlink:LL (R1,R2,R3)
    R1 <= R2; R2<=(R3)
StoreCond:SC (R1,R2,R3)
    if R1 == R2
       (R2) <= R3
```

II. Methodology

To be able to implement a dual core capable DLX core, several assumptions were made. Simultaneous Multiprocessing (SMP) was the multicore architecture used wherein two processors share a single memory and address space. SMP was chosen because it is the best architecture suited for multicore implementations with low number of cores. Also it was assumed that the DCC DLX core will be given the first memory access after boot up. This case was assumed because a deadlock would occur if both cores would try to access at boot up.

A. Implementation

HDL was used to model systems at various levels of abstraction. The project design and implementation started at the behavioral level. After verifying the functionality in the behavioral level the design was then coded in rtl. The blocks were synthesized to generate the netlist from the standard cells used. The synthesized gates were simulated again, but this time, with the necessary timing delays taken into account. The place and route tool was used to produce and optimize the layout of the circuit. The final layouts were constructed to verify communication between the computational cores. Results of the Base DLX implementation were then compared with the final multicore capable DLX implementation. Illustrated in Figure 1 is the design methodology.

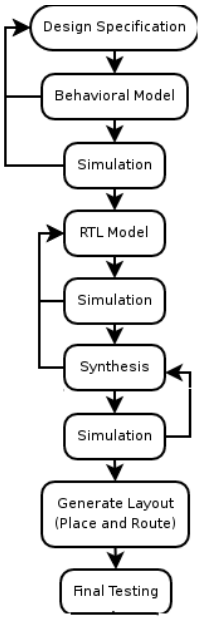

Fig. 1. Design Methodology

B. Testing/Verification

1) Behavioral Level Testing: To verify the functionality of the Behavioral model of the Dual Core capable DLX Microprocessor the group used several test cases and stages. First, the individual blocks were tested for proper functionality before integration. After the individual blocks were tested and verified, blocks were integrated on a per-stage basis. This yielded 5 major blocks, which correspond to the 5 stages of an execution of instruction. After the major blocks have been tested and verified, all these blocks were combined to make the dual core capable 32-bit DLX Microprocessor. The DCC DLX was then tested under normal pipeline execution, without the occurrence of any hazards or interrupts. Then the behavioral model was tested for proper execution of the 51 instructions in the DLX instruction set. The DCC DLX Microprocessor was also tested for all the possible forwarding paths in the pipeline and all types of stalls due to data hazards. After exhausting all cases of data hazards, the behavioral model was verified under all possible interrupts, with proper prioritization of the interrupt signals. It was tested using sample programs such as factorial and greatest common factor (GCF). These sample programs combined all the important aspects in the testing of the behavioral model. At this stage only one core was tested, since the Dual Core Capable DLX microprocessor can also function as a single processor. After the testing of the dual core capable DLX as a single processor, the dual core capability of the processor was verified by testing the atomic instructions. From the point of view of the DCC DLX, there is another processor executing a different set of programs which needs to access the memory.

2) RTL Level Testing: The RTL model was also tested for its functionality. All the test benches that were used in the behavioral model were also used in this level of testing. The same sequence of testing followed, first one DCC DLX microprocessor (functioning as a single processor) was tested in the RTL level, and then two dualcore capable DLX microprocessors were tested also in the RTL level.

3) Gate Level Testing/Netlist Simulation: The Standard Delay File (SDF) was extracted during synthesis and was used in the simulation of the synthesized codes. The synthesized codes were tested using the GCF and Factorial test programs. At this stage, we expected different simulation results from the previous levels since gate delays were

introduced.

4) Layout Testing: The generated layout was tested using the GCF, Factorial test programs and the atomic instructions. Results from this simulation coincided with the results in the gate level testing. This is also the level where the maximum frequency of the Dual core capable DLX microprocessor is experimented and obtained.

5) Verification: In this stage the synthesized netlist is compared with the netlist of the layout in able to confirm the functionality of the layout. The chip layout was checked for design rule errors from the CMOS process used.

III. Results And Analysis

In this section, results gathered from the testing phase of the project were compared to the results of previous DLX implementations of the host laboratory, specifically [3].

A. Area Reports

The Dual Core Capable DLX is larger in area than the Base DLX base core by 23%. Since the DCC DLX is capable of operating at a higher frequency–maximum of 83.33Mhz while Base DLX's maximum is at 25Mhz–the synthesis tool have sacrificed area for speed[2]. The difference in area can also be attributed to the difference in coding styles and the languages used (Verilog vs VHDL). Considering that the group used *Synopsys* tools(instead of Cadence) synthesis tools were also considered to play a part in the reported disparity.

The total area for the DCC DLX is 1.78 mm^2 while that of Base DLX is 1.438 mm^2.

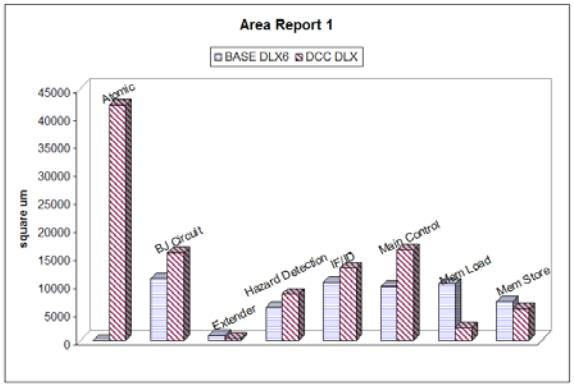

Fig. 2. Area Comparison on Selected Blocks of Base DLX and DCC DLX

B. Timing Reports

Three different programs were used to check the functionality of the DLX processor; namely, the GCF, Multiply, and Factorial programs. The performance of the DLX7 core was then evaluated and compared to the previous Base DLX implementation as seen on Table I. These programs were ran assuming that the second core was not enabled, meaning bus control goes solely to the DLX core. A clock frequency of 25 MHz and memory access time of one clock cycle or 40 ns were assumed which are also the same assumptions used in the base Base DLX.

From the results obtained, we can see that there was a slight increase in the execution time of the DLX processor. This increase is attributed to the addition of the Bus Arbitration Unit, which was responsible for obtaining bus access to the memory. Additional clock cycles were needed to evaluate the bus condition causing an increase in the memory access time of the processor and therefore delaying the program execution. However, since the other core is idle, this additional delay can be seen only at start-up. Upon establishing bus control, normal delays could be

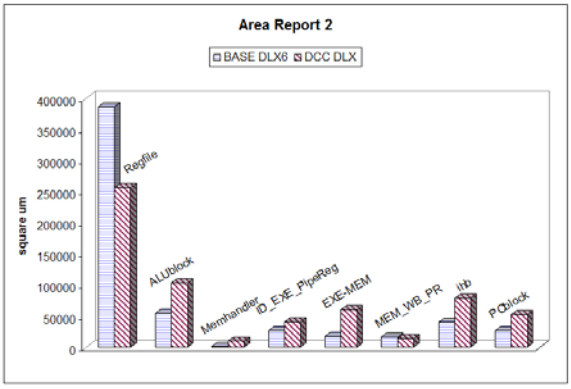

Fig. 3. Area Comparison on Selected Blocks of Base DLX and DCC DLX

MULTIPLY	***hex***	***cc***	***cc***
Inputs (hex)	***Results***	***Base DLX***	***DCCDLX***
8,181C33	C0E198	99	105
F,181C33	169A6FD	169	175
5F,5F	2341	967	973
59,FFFFFFFE	FFFFFF4E	907	915
2,75BCD15	EB79A2A	39	45
GCF	***hex***	***cc***	***cc***
Inputs(dec)	***Results***	***Base DLX***	***DCCDLX***
[200,218]	2	349	351
[193,180]	1	343	345
[17,200]	1	295	309
[156,169]	D	205	219
[47,85]	1	163	171
FACTORIAL	***hex***	***cc***	***cc***
Inputs (dec)	***Results***	***Base DLX***	***DCCDLX***
5	78	155	157
6	2D0	223	225
8	9D80	389	391
10	375F00	595	597
12	1C8CFC00	841	843

TABLE I
Timing Results for GCF, Multiply, and Factorial Programs

seen on memory accesses done by the DLX core. In the case that the other core becomes active, additional delays would be seen when having memory accesses. The overall timing overhead of the Dual Core Capable DLX core is only 216.27 ns on the average, which is equivalent to 5 1/2 clock cycles.

C. Power Statistics

An increase in power consumption was expected since we added additional blocks to implement the dual core capability. We can see that additional power was consumed by the Atomic block and the Bus Arbiter. The power results of the three tests (GCF, Multiply, Factorial) done on the Dual Core Capable DLX core can be seen on Appendix C. The results were then compared to the Base DLX core power results. The reduction in total average power for the current DLX model is attributed to a significant power reduction in our Register File. From (Base DLX) 25.6mW, we were able to reduce it by half to (DCC DLX) 12.98mW. Since the area of the register file is smaller (see Table 3) then naturally it has lesser components which can lead to lesser power consumption. Other than that, all other blocks registered slightly higher power consumption due to additional functionalities incorporated on the blocks.

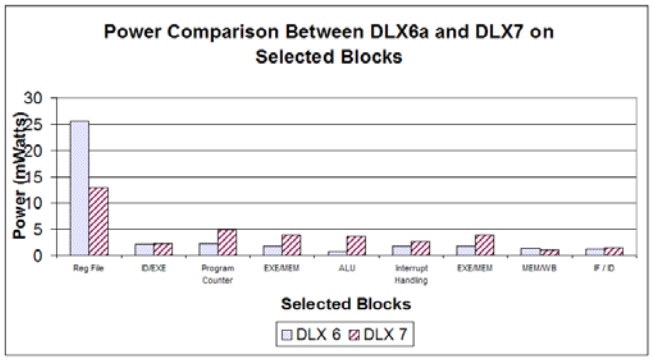

Fig. 4. Power Comparison of Major Blocks

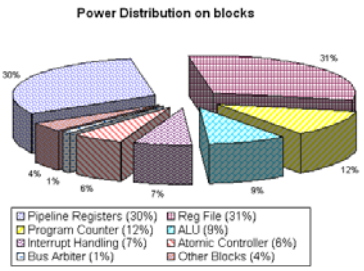

Fig. 5. Power Distribution among blocks

A summary of the power consumption of major blocks can be seen on Figure 4. Even though the power consumption of the Register File was reduced, it still consumed the most power among the major blocks. Power consumption of modified blocks also increased when compared to Base DLX since generally, additional logic were implemented in almost all of the blocks.

IV. Conclusion

This project demonstrated how the DLX processor could be modified to include dual core capability. The dual core capability of the DLX core was implemented using Verilog HDL and yielded an area overhead of 23% and power reduction of 17.9%. Addition of the dual core capability does not interfere with the processor's performance as a single core. However, the performance (specifically, throughput) advantage of a dual core system was not fully demonstrated as the cores lacked instruction caches critical to full multicore system operation.

The development of dual core capability for the DLX microprocessor is an important and crucial step in the laboratory's processor development program. For the DLX microprocessor to maintain its relevance in modern times, it has to cope and catch up with technologies that are being implemented, tried, and studied both in the industry and the academe; hence, the importance of this endeavor and its related successors.

References

[1] Bautista J. A. et. al. ***"High Level Design Implementation and Characterization of a 32-Bit 5-Stage Pipelined DLX Microprocessor with Single Level Cache"*** Undergraduate Student Project, Department of Electrical and Electronics Engineering, University of the Philippines, Diliman, November 2005.

[2] Bhatnagar H. ***"Advanced ASIC Chip Synthesis Using Synopsys Design Compiler, Physical Compiler, Prime Time."*** 2nd ed. Kluwer Academic Publishing. 2002

[3] Cantavieja, D. F. et. al. ***"Design and Implementation of a Power-Optimized DLX Microprocessor"*** Undergraduate Student Project, Department of Electrical and Electronics Engineering, University of the Philippines, Diliman, November 2005.

[4] Di Giacomo, J. ***"VLSI Handbook. Silicon, Gallium Arsenide and Superconductor Circuits."*** Mc Graw Hill. 1989

[5] Flynn, M.J. "***Very High-Speed Computing Systems***" Proceedings in IEEE 54:12 December, 1901-1909

[6] Hamacher, C. et.al. ***"Computer Organization."*** 5th edition. McGraw-Hill Higher Education

[7] Hovemeyer D., William, P., Spacco J. ***"Atomic Instructions in Java."*** Department of Computer Science, University of Maryland. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP 2002)*, Málaga, Spain, June 10-14, 2002.

[8] Mano M.M. ***"Digital Design"*** 3rd ed. Prentice Hall. 2001

[9] Patterson D.A., Ditzel D. ***"The Case for the Reduced Instruction Set Computer."*** In *ACM SIGARCH Computer Architecture News*, v.8 n.6, p.25-33, October 1980

[10] Patterson D.A., Hennesy JL. ***"Computer Architecture: A Quantitative Approach."*** 4th beta edition. Morgan-Kauffman, 2006.