EEPROM Memory Controller Architecture for an Out of Order Execution

Shankaran S[1], K. Hari Priya[2]

[1]M.Tech Student
CVSR College of Engineering, Secunderabad, India
sshankaran3@gmail.com

[2]Associate Professor, ECE Department
CVSR College of Engineering, Secunderabad, India
kichapriya@yahoo.co.in

Abstract: ***In modern day computer technology the time taken to access the information from the memory is larger compared to the time taken to execute the instructions. So we have to reduce the access time of the processor. An attempt has been made to reduce the time taken to execute the instructions by implementing several executing techniques so that the processor does not remain idle. In sequential execution the processor has to remain idle for a longer time in this process the processor is not used efficiently. Therefore to overcome this barrier out of order execution has been proposed by which the time taken to execute the instructions is reduced to a greater extent. In out of order execution the response time is reduced. The out of order also allows a prioritized handling of multiple operations. EEPROM is a nonvolatile memory which can be programmed and erased by using field electron emission the main characteristic of EEPROM is the way the memory programs and erases. Electrically erasable programmable read only memory controller architecture has been employed to enhance the performance of the system by executing the instructions in decoupled and out of order pattern so that the execution of instruction takes place at a faster rate and the efficiency of the processor is increased.***

Keywords: EEPROM, RAM, DRAM, SRAM, OoO.

1. Introduction

In modern day computer technology then taken to access the information from the memory is larger compared to the time taken to execute the instructions. Therefore an attempt has been made to reduce the time taken to execute by implementing several executing techniques few of them are decoupled execution and an out of order execution by which the efficiency of the system can be improved to a greater extent. EEPROM (Electrically erasable programmable read-only memory) in this type the data can be rewritten electrically, while the chip is on the circuit board, but the writing process is slow. This type is used to hold firmware, the low level microcode which runs hardware devices, such as the BIOS program in most computers, so that it can be updated. In this paper we study the characteristics of electrically erasable programmable read only memory and compare it with other devices, also we discuss about in order and out of order processors. And also about memory controllers how it accesses memory and assign priorities to the tasks to be performed. We discuss about the various kinds of execution and in particular how the out of order execution helps in improving the performance of the system by executing the instructions in an out of order fashion.

2. EEPROM

Property (1): Shift in Demand Confirms the Value of Parallel EEPROM: A change in emphasis from Parallel to Serial architecture (due to the availability of larger densities, flexible interfaces, cascading features, and a cheaper cost structure) has steered demand for fine granularity data storage in high volume, price compressed telecom applications towards the Serial EEPROM platform, in some applications. This shift in high speed, fine granularity data storage demand peaked in 1998 and has stabilized in the first half of this year. The result is a smaller set of system applications and a smaller demand, albeit a more sharply defined set of end-system performance criteria that are satisfied by the unique features of the Parallel EEPROM. These unique Parallel EEPROM features include programming flexibility, high reliability, and comparatively faster data transfer rates. These enable the Parallel EEPROM to separate itself from the pack of data storage solutions and provide benefit to network switch applications, fiber optic equipment, telecom base stations and energy meter systems. Property (2) Flexibility: The AT28C010, 1 megabit Parallel EEPROM, can be programmed byte-by-byte or in the full page format versus the page-only format required by a serial-interface 1 megabit device. Byte-level programmability, without the need to issue multiple write instruction sets required by serial device programming, allows the designer of metering and network switching equipment to specify fine-granularity updates without having to employ software overhead, code shadowing or complete block/page rewrite.

3. Execution Techniques

3.1. An Out of Order Execution (OoO)

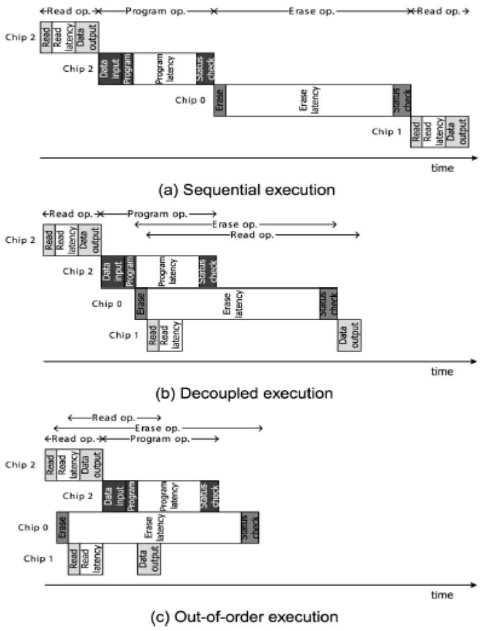
In computer engineering, OoO is a paradigm used in most high-performance microprocessors to make use of instruction cycles that would otherwise be wasted by a certain type of costly delay. In this paradigm, a processor executes instructions in an order governed by the availability of input data, rather than by their original order in a program. In doing so, the processor can avoid being idle while data is retrieved for the next instruction in a program, processing instead the next instructions which are able to run immediately. The benefit of OoO processing grows as the instruction pipeline deepens and the speed difference between main memory (and cache memory) and the processor widens. On modern machines, the processor runs many times faster than the memory, so during the time an in-order processor spends waiting for data to arrive, it could have processed a large number of instructions.

3.2. Sequential Execution

The process of executing the instructions one after the other in a serial order without jumping the order is known as serial execution .In this process the processor has to remain idle for a longer period for accessing the information from the memory therefore lot of time is wasted and the performance of the system is not utilized to its maximum. In order to keeps the processor busy executing the instructions the data has to be accessed from the memory at a greater speed for which decoupled execution is employed. In decoupled execution the time taken to execute the instructions is lesser when compared to the time taken by sequential execution.

3.2. Decoupled Execution

In decoupled execution a command is issued during the latency time of the previous command whereas in serial execution a command is issued only after the execution of the previous command. In decoupled execution though the instructions are executed at a faster rate they are executed only in the order specified. Therefore an out of order execution technique can be considered as an advancement of decoupled execution in which the instructions are executed in an out of order by reducing the time taken to execute the instructions and increasing the performance of the system. The time taken by an out of order technique to execute the instruction is lesser when compared to the time taken by decoupled execution. Decoupled architectures are generally thought of as not useful for general purpose computing as they do not handle control intensive code well.[2] Control intensive code include such things as nested branches which occur frequently in operating system kernels. Decoupled architectures play an important role in scheduling in Very long instruction word.



(a) Sequential execution

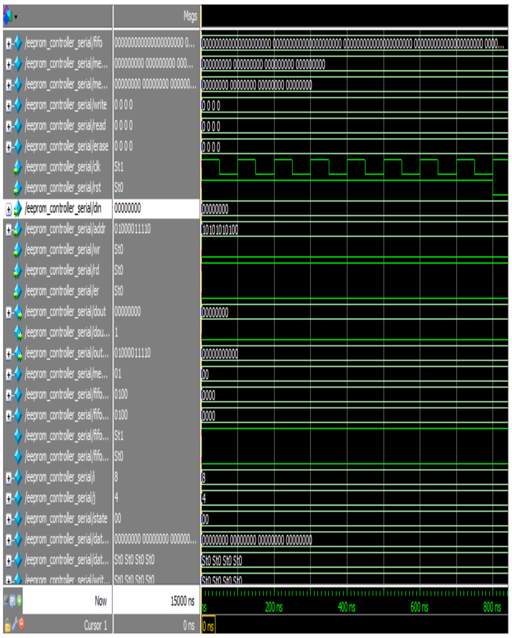
(b) Decoupled execution

(c) Out-of-order execution

Figure 1: Pictorial representation of serial, decoupled and an out of order execution

4. Implementation and Results

Simulation Results for Serial Execution



	Msgs
/eeprom_controller_serial/fifo	00000000000000000000 0...
/eeprom_controller_serial/me...	00000000 00000000 000...
/eeprom_controller_serial/me...	0000000 0000000 000000...
/eeprom_controller_serial/write	0 0 0 0
/eeprom_controller_serial/read	0 0 0 0
/eeprom_controller_serial/erase	0 0 0 0
/eeprom_controller_serial/clk	St1
/eeprom_controller_serial/rst	St0
/eeprom_controller_serial/din	00000000
/eeprom_controller_serial/addr	01000011110
/eeprom_controller_serial/wr	St0
/eeprom_controller_serial/rd	St0
/eeprom_controller_serial/er	St0
/eeprom_controller_serial/dout	00000000
/eeprom_controller_serial/dou...	1
/eeprom_controller_serial/out...	01000011110
/eeprom_controller_serial/me...	01
/eeprom_controller_serial/fifo...	0100
/eeprom_controller_serial/fifo...	0100
/eeprom_controller_serial/fifo...	St1
/eeprom_controller_serial/fifo...	St0
/eeprom_controller_serial/i	8
/eeprom_controller_serial/j	4
/eeprom_controller_serial/state	00
/eeprom_controller_serial/dat...	0000000 0000000 000000...
/eeprom_controller_serial/dat...	St0 St0 St0 St0
/eeprom_controller_serial/writ...	St0 St0 St0 St0
Now	15000 ns
Cursor 1	0 ns

Figure 2: Waveform for serial execution

Simulation Results for Decoupled Execution

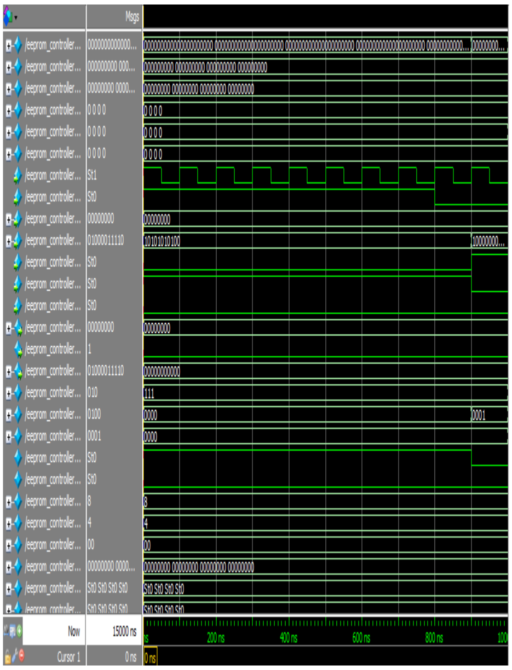


Figure 3: Wave for decoupled execution

Simulation Results for EEPROM Memory

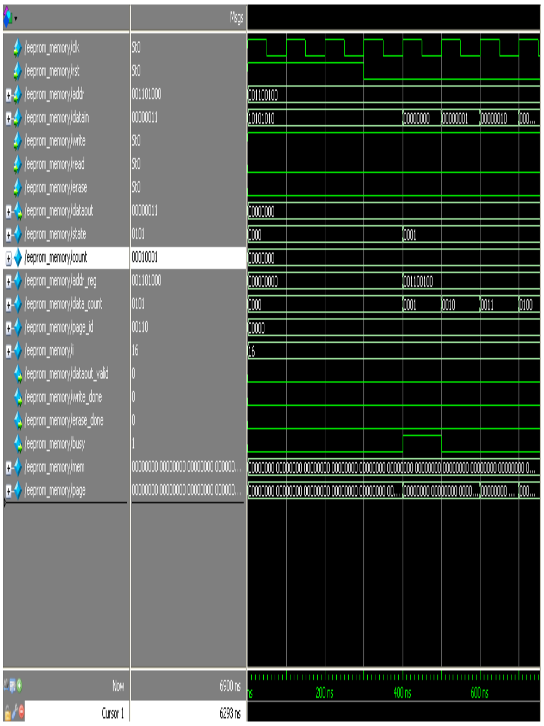


Figure 4: Waveform for EEPROM memory

Simulation results for out of order execution

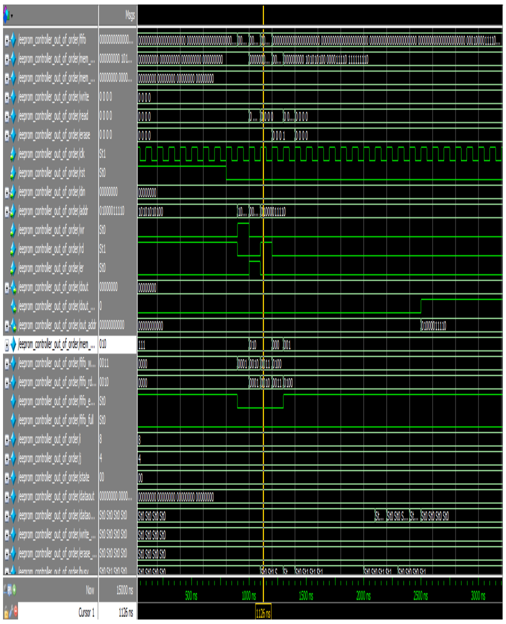


Figure 5: Waveform for out of order execution

5. Conclusion

We have presented an electrically erasable programmable read only memory controller architecture for executing multiple instructions in out of order pattern. We have shown how Out of order execution technique plays a crucial role in executing the instructions at a greater speed .Response time of the processor is reduced as much as possible by executing the instructions in out of order fashion. It provided handling of multiple operations allowing appropriate levels of service to be given at different stream of operations. EEPROM memory controllers are used in digital potentiometers and real time clocks etc where response time plays a key role. Faster data rate provide by the parallel platform also benefit fiber optics, storage area networks and switch applications. EEPROM are identified as flexible reliable and faster data storage system. We are currently exploring different approaches to parallelizing EEPROM management software to realize the full potential of out of order execution. Efforts are put to reduce the latency time. We are also trying to reduce the response time of the processor .In near future out of order execution plays a vital role in every memory controller.

References

[1] C.G. Hwang, “Nanotechnology Enables a New Memory Growth Model,” Proc. IEEE, vol. 91, no. 11, pp. 1765-1771, Nov. 2003

[2] G.E. Moore, “Cramming More Components onto Integrated Circuits,” Electronics, vol. 38, no. 8, pp. 114-117, Apr. 1965.

[3] Samsung Electronics, Datasheet: 1G x 8 Bit/2G x 8 Bit NAND flash memory,http://www.datasheetcatalog.org/datasheets2/12/1244179_1.pdf, Mar. 2005.

[4] E. Gal and S. Toledo, “Algorithms and Data Structures for Flash Memories,” ACM Computing Surveys, vol. 37, no. 2, pp. 138-163, 2005.

[5] L.M. Grupp, A.M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P.H, P.H. Siegel, and J.K. Wolf, “Characterizing Flash Memory: Anomalies, Observations, and Applications,” Proc. 42nd Ann. IEEE/ACM Int’l Symp. Microarchitecture, 2009

[6] ONFI, ONFI 2.3 Specification, http://onfi.org/specifications/, Aug. 2010

[7] R. Schuetz, O. HakJune, K. Jin-Ki, P. Hong-Beom, S.A. Przybylski, and P. Gillingham, “Hyper Link NAND Flash Architecture for Mass Storage Applications,” Proc. 22nd IEEE Non-Volatile Semiconductor Memory Workshop, pp. 3-4, 2007.