IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 4, June 2014.
www.ijiset.com
ISSN 2348 – 7968

DESIGN & DEVELOPMENT OF MMU FOR MIL-STD-1750 PROCESSOR

Manikandan.S[1], Dr. G. Indumathi [2]
[1,2]CMR Institute of Technology
Bangalore, Karnataka, India

Abstract

The aim of the project is to Design & Develop a MMU for Military Standard 1750A Microprocessor so as to function as a Processor Interface device. Design and Development of FPGA internal MMU (Memory Management Unit) by considering the black box view of a dedicated MMU device MA31750, which is going obsolete. The Memory Management Unit is a logical block designed to increase the memory addressing capability of Military Standard 1750A Processors. The objective of this project is to design and develop the MMU interface logic to extend the memory addressing capability and being operated for higher frequency (24 MHz) processor in VHDL language which is synthesizable with a simulated testbench.

Keywords: FPGA, MMU, MIL-STD-1750

I. INTRODUCTION

A memory management unit (MMU), sometimes called paged memory management unit (PMMU), is a computer hardware unit having all memory references passed through it. Primarily performing the translation of virtual memory addresses to physical addresses, it is implemented as part of the central processing unit (CPU), but it can also be in the form of a separate integrated circuit (IC). An MMU is effectively performing the virtual memory management, handling at the same time memory protection, cache control, bus arbitration and bank switching.

II. FPGA versus ASIC

FPGAs are excellent for designing and prototyping digital logic into medium-volume, medium-density applications, their high unit cost makes things difficult. On the other hand, the low unit cost of ASICs is one of the main reasons why these are considered for high-volume manufacturing. Designing a new product around FPGA allows design modifications to be quickly made throughout the development process. Once this design is complete and approved for production, the FPGA design can be migrated to an ASIC design and then produced, cutting the production unit cost greatly.

In addition, developing the FPGA and ASIC in a parallel design flow will help to speed up the process. Finally, planning for portability to an ASIC from the beginning of the project will help to speed up time-to-market and decrease costs. Good design practices such as the use of synchronous design techniques will enable the design to be ported across many different technology platforms. Finally, one of the most important things a design team can do is have good documentation of the design.

III. MEMORY MANAGEMENT UNIT

Modern MMUs typically divide the virtual address space (the range of addresses used by the Processor) into pages, each having a size which is a power of 2, usually a few kilobytes, but they may be much larger. The bottom bits of the address (the

offset within a page) are left unchanged. The upper address bits are the virtual page numbers.

With paging, the main memory is partitioned into small equal fixed sized blocks called page frames. A process to be loaded is also divided into blocks of the same sizes, called pages. When a process is to be loaded its pages could be assigned to any free page frames. This eliminates the external fragmentation problem that could occur with dynamic partitioning. It also minimizes internal fragmentation because only the last loaded page frame will suffer from internal fragmentation. In general, this fragmentation will be half the page size. For every memory reference the system must localize the correct page frame location for the page to which the address belongs. This is performed in hardware, otherwise the time consumption when doing all the address calculations and lookups would be too great. Every address generated by the CPU, a logical address, is divided into two parts, a page number and a page offset. The page number is used as an index into a page table.

The alternate Processor can address a maximum of 64kB of memory through its address lines which falls short of 1MB of memory that is available onboard. Thus, the MMU caters to this need by providing extended addressing thereby making it possible for the processor address 1MB of memory.

IV. DESIGN OF MMU

The architecture is similar to that of the existing MA31751 MMU. The principle function of MMU is to provide extended addressing to the processor by means of address translation. The Block Protection Unit (BPU) of MA31750 is not implemented in the current MMU design. The MMU module is designed only to increase the memory addressing capability of the alternate version CPU. The processor inputs the 16 bit address (ADDR_VALID_IS [15:0]) and the Address state (AS_IS [3:0]). The MMU performs address translation to output the physical page address (PPA) which is referred to by Extended Address (EA_OS [7:0]). Figure 1 shows the I/O's of the MMU module. The main memory (SRAM) is divided into 256 pages of size 4kB each. Therefore, the MMU houses an array of registers which contain the physical page addresses of all the pages available within the memory. The MMU also decodes the incoming processor address to clarify the command as either an operation on operand register or instruction registers. The page registers in both banks are initialized to provide a linear, one to one mapping of the PPA.

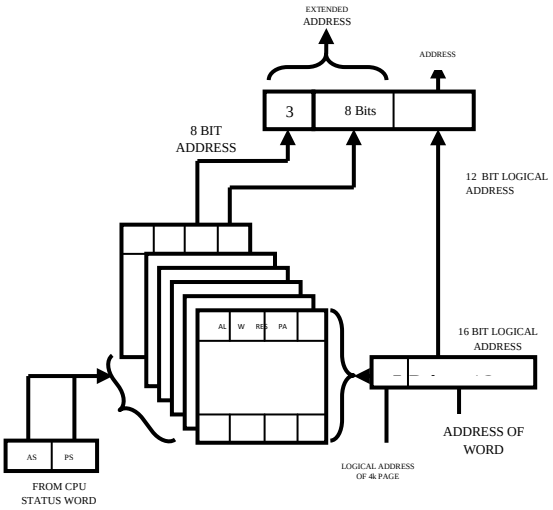


Figure 1 Memory mapping

The MMU comprises of the following components: Address Decoder, Page Registers 2 to 1 Multiplexer. The processor data is written into the specified 8 bit register address (REG_ADR_S). Write operation into the file registers occurs only during the rising edge of the write clock (W_CLK_S) and when the write enable (WE_S) is high. Two 256 to 1 multiplexers are placed one outside each set of page registers (Operand and Instruction page

ISSN 2348 – 7968

registers). Depending on the selection lines (REG_ADR_IS), the multiplexer selects the contents of required page register and relay it to the 2 to 1 mux. The 2 to 1 mux selects between the signals MMU_DATA_OUT_INST_S & MMU_DATA_OUT_OPR_S depending upon the selection line INSTRUCTION_EN_S to output MMU_DATA_OUT_OS. The data for processor read is always available on MMU_DATA_OUT_OS. The enable for processor data read (MMU_DATA_SEL_OS) is taken outside along with MMU_DATA_OUT_OS and later given as a section line in processor data bus routing.

V TRANSLATION

Two 256 to 1 multiplexers are placed one outside each set of page registers (operand and instruction page registers). Depending on the selection lines, the multiplexer selects the content of required page register and relays it to the 2 to 1 mux. TRANS_ADR_S to the multiplexer is formed by the concatenation of AS_IS [3:0] and ADDR_VALID_IS [3:0]. Depending upon the value of the selection line DI_IS, either the EA_INSTRUCTION_S or the EA_OPERAND_S is selected and relayed as the required EA_OS [7:0] as an output of the MMU module. Similar to read, the translation operation is asynchronous. EA_OS [7:0] is always available outside the MMU module irrespective of any translation enable. To make use of EA_OS [7:0], the processor asserts the IOM_IS signal to logic low.

VI. EXTENDED ADDRESSING

The extended 20 bit address is formed by concatenation of EA_OS [7:0] and the ADDR_VALID_IS [11:0]

LOGICAL ADDRESS	
AS_IS [3:0]	ADDR_VALID_IS [15:0]

PHYSICAL ADDRESS	
EA_OS [7:0]	ADDR_VALID_IS [11:0]

Table 1 Extended Addresses

VII. DESIGN SIMULATION

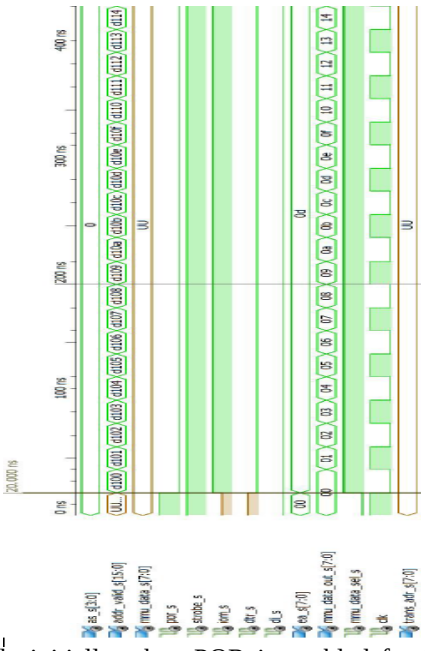
A. TEST BENCH

The VHDL code for MMU module was simulated for all combinations of addresses, Data and control signals. The MMU module was simulated for all 2^{20} combinations of processor addresses and all combinations of data.

B. SIMULATION RESULTS

Figure 2 Simulated result

Figure 2 shows the simulated result of the MMU



module initially when POR is enabled for 20 ns, the MMU Address, Data, Input Output, and DTR are disabled. Once POR is disabled, Initialization of the register array takes place, 256 x 8 for Instruction page register & 256 x 8 for Operand page register

ISSN 2348 – 7968

VIII. CONCLUSION

Memory Management Unit for extending the 16 bit processor address to reference 1M words (20 bit address) with the use of 256 Operand and Instruction page register each has been Designed, Tested for implementation. This design work will optimize the board by limiting the need for a dedicated MMU device and also helps to tide over the obsolescence in the MIL-STD-1750 compatible MMU. Future work can integrate the BPU (Block Protection Unit) for memory protection into the design as with the case of the dedicated MMU.

REFERENCES

[1] John L. Hennsley and David A. Patterson, "*Computer Architecture – A quantitative approach*", 2 edition Morgan Kaufmanns publishers inc. San Francisco, California, 1996.

[2] *Memory management overview*, Karl Ingström, Anders Daleby. Department of Computer Engineering, University of Mälardalen, Sweden, 2001.

[3] K.C. Chang, "*Digital Design & Modeling with VHDL & Synthesis*" IEEE Computer society press.

[4] Enoch O. Hwang, "*Digital Logic and Microprocessor Design with VHDL*" CENAGE Learning.

[5] Stefan Sjoholm, "*VHDL for Designers*" Pearson Prentice Hall.

[6] Pongp.Chu "*RTL Hardware Design using VHDL*".

[7] Floyd & Jain "*Digital Fundamentals*" Pearson Prentice Hall.