VME 시스템 제어기의 FPGA 구현

배 상 현† · 이 강 현††

요 약

산업분야의 공장자동화와 자동 측정장비의 다중 프로세서 환경의 시스템 성능을 향상시키는 표준버스가 필요하다. VME 버스는 이러한 명세에 적합하지만, 소규모 패키지와 보드의 낮은 집적성 사양을 가지고 있다. 더욱이 보드와 반도체 집적성은 개발시간, 연구비용, 현장진단에 영향을 주는 중요한 문제로 대두되어 있다.

이러한 추세에 맞추어, 본 논문에서는 VME 버스와 제어기 모듈 사이의 주기능인 중재, 인터럽트, 인터페이스를 Revision C.1(IEEE Std. P1014-1987)의 통합환경으로 구성하고, 설계된 VME 시스템 제어기를 Slot 1에 장착할 수 있도록 FPGA 상에 구현한다. 제어 및 기능 모듈의 동작은 VHDL의 mid-fixed 방식으로 코딩을 하고 검증하였다. 실험을 통하여 VME 시스템 제어기의 가장 중요한 동작인 버스 타이머의 버스 에러 신호가 56μs 이내에 발생된 것과, 제어모듈과 기능모듈의 정확한 상호 동작도 확인하였다.

그러므로 구축된 VHDL 라이브러리는 VME 버스 기반 시스템과 ASIC 설계에 응용할 수가 있다.

FPGA Implementation of VME System Controller

Sang Hyun Bae† · Kang Hyeon Rhee††

ABSTRACT

For FA(factory automation) and ATE(automatic test equipment) in the industrial area, the standard bus needs to increase the system performance of multiprocessor environment. VME(versa module european package format) bus is appropriated to the standard bus but has features of small package and low board density. Beside, the density of board and semiconductor have grown to become significant issues that affect development time, project cost and field diagnostics.

To fit this trend, in this paper, we composed Revision C.1(IEEE Std. P1014-1987) of the integrated environment for the main function such as arbitration, interrupt and interface between VMEbus and several control modules. Also the designed VME system controller is implemented on FPGA that can be located even into Slot 1.

The control and function modules are coded with VHDL mid-fixed description method and then those operations are verified by simulation. As a result of experiment, we confirmed the most important that is the operation of Bus timer about Bus error signal should occur within 56μs, and both control and function modules have the reciprocal operation correctly.

Thus, the constructed VHDL library will be able to apply the system based VMEbus and ASIC design.

※This study was supported by Factory Automation Center for Parts of Vehicles(FACPOV) in Chosun university, Kwangju, Korea. FACPOV is designated as a regional research center of Korea Science and Engineering Foundation(KOSEF) and operated by Chosun university.

† 정 회 원:조선대학교 전산통계학과
†† 정 회 원:조선대학교 전자정보통신공학부

논문접수:1997년 7월 5일, 심사완료:1997년 10월 22일

1. Introduction

Improved productivity and management systematization on the factory automation are the essential factors in industry today. For the automation, almost multiprocessing system is based on the bus architecture because that is implemented easily and low cost. Standard bus such as GPIB, Multibus, VMEbus and VXIbus should be support to the implementation of automation, whereas the choice method of bus is still an open problem for the improvement of system performance.

The basis of this goal, VMEbus has several advantages such as an extension of function and performance, a definition of accurate specification and a flexible interface to 32 bit microprocessor, so VMEbus can be used an overall industry area that necessitate to automatize and systematize the measure equipment. Also VMEbus has high speed of data transmission and an excellent compatibility to each other system, so that is widely used and stretched over from the factory automation to military area. In a domestic industry area, the module development of measure and control is progressed actively. On the cost and requirement technology, the development of function module is easily implemented and has high level. But the development of control module of the system operation is relatively low level too.

Main function of system controller is an use arbitration of bus and the management of interrupt, and VME system is able to control intellectually the sequence of priority about the request of bus interrupt for more efficiently use of bus and the interrupt management. Moreover, in the multiprocessing environment, each processor requires the interface module that will able to operate the adaptable protocol of VMEbus[1].

In this condition, VME system controller requires the equipped several modules such as Arbiter that arbitrates use of bus, Interruptor, Interrupt handler, Interrupt daisy-chain driver, Bus timer that generates error signal at use of bus and Requester[2], [3]. And the most of control modules are distributed through several slots of VME system.

In this paper, according to these several particulars requested, we designed the integrated VME system controller. For the reciprocal operation correctly between both control and function modules, the operation of each module of VME system is coded with VHDL mid-fixed description method and then synthesized and implemented on FPGA. We also considered the integrated environment of VME system controller that can be located even into Slot 1.

The remainder of this paper is composed as follows. In section 2, it is discussed the background summary of VME systems. In Section 3, for the coding with VHDL behavioral description, the operation of each module of VME system is specified to fit the system clock. In Section 4, VME control system coded with VHDL description is synthesized and implemented on FPGA, so investigated and analyzed the experimental results. Conclusions are drawn and point to future research directions in Section 5.

2. Background of VME System

In this section, we briefly discuss the concepts of VME system architecture[1], [3], operation and specification. With advanced semiconductor technology, development of microprocessor should be the kernel of electronic industry. Because the performance of bus was not as improvable as the performance ability of microprocessor. The performance evaluation of system is influenced to the transmission ability than the data processing ability.

In the later half of 70 years, the activity is began to develop a new industrial standard bus that is to fit the transmission ability with the advent of 32 bit microprocessor. So IEEE P896 committee was organized then Versa bus is announced in 1979. VMEbus is applied Eurocard Package Format with Versa bus so announced Revision A in 1981. As first announced,

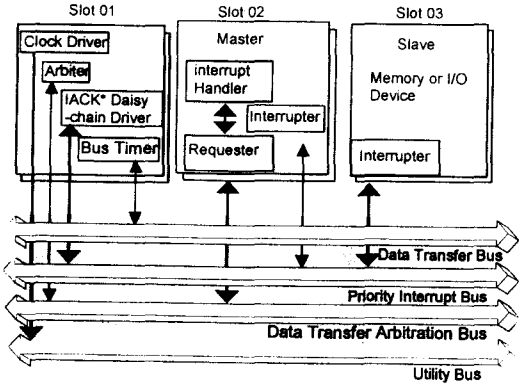
VMEbus was 32 bit bus that is the target to 32 bit microprocessor and now, the present state of application is Revision C.1(IEEE Std. P1014-1987).

The specification of VME system is described as follows.

- ▶ 32 bit address/data
- ▶ Non-multiplexed address/data
- ▶ Asynchronous transmission method
- ▶ Max. transmission speed : 40 MB/sec
- ▶ data transmission unit : 1 − 256 byte
- ▶ Interrupt : 7 stage priority order method
- ▶ Bus arbitration : 4 stage priority order/round robin

Like as above specification of VME system that is composed of mainly two modules. One of them is control module that controls and distributes bus, and transmits data between each other module, the other one is measuring and processing the installed modules on system[4], [5].

The whole composition of VME system is shown in (Fig. 1). System controller located on Slot 1 takes charge of the several operation such as control, bus distribution and data transmission of each module. Controller has many core modules those are Arbiter that arbitrates use of bus, Interrupter, Interrupt daisy-chain driver, Bus timer that occurs error signal and Requester that requests use of bus.



(Fig. 1) Block diagram of VME system

3. VME System Controller Design

Most recently, the structure trend of VME system controller includes several modules such as over one microprocessor, memory, basic I/O module and the function of data processing by service on real-time operation system[6], [7].

As shown in (Fig. 1), VME interface modules are located on Slot 2 and 3 those are added to system controller located on Slot 1 through VMEbus. To fit this trend, in this paper, our goal of design is dealt as follows. Arbiter, Interrupt daisy chain driver and Bus timer on Slot 1, and Interrupt handler, Interrupter and Requester on Slot 2 and 3, those are composited to the integrated environment and will be set to the standard Revision C.1(IEEE Std. P1014-1987).

For the implementation of the integrated environment on FPGA located even into Slot 1. we coded VME system by VHDL description of mid-fixed method[8-10]. Firstly, design is progressed by coding of VHDL behavioral description about each control and function modules. Secondly, those modules are jointed into the block level by the structural composition. And then through the synthesis and verification[11], the design is implemented on FPGA.

3.1 Arbiter

Arbiter that arbitrates use of bus of VME system is composed of Single level arbiter, Priority arbiter and Round-robin arbiter. It is described in 〈Table 1〉 where, BR means that "bus request". In this paper, we consider only Priority arbiter because that is excellent at the efficiency of bus arbitration.

According to 4 step priority order, Priority arbiter arbitrates use of bus. In here, the condition of BCRL signal generation should be also considered, where BCRL means that "bus clear". To satisfy this condition, according to the present master level that waits

for the requestion order, priority order algorithm is defined in 〈Table 2〉.

And the composition of Arbiter that arbitrates the right of use of bus is shown in (Fig. 2).

〈Table 1〉 Arbitration description

Arbitration type	content	Priority order
(SGL)Single level arbitration	Using only one bus request level	BR3, BG3IN, BG3OUT
(PRA)Priority arbitration	Allow the priority order, when receive BR signal of other level, then process the signal of higher priority order	BR3>BR2>BR1>BR0
(RRS) Round-robin arbitration	Allow all request level to the same priority order	....

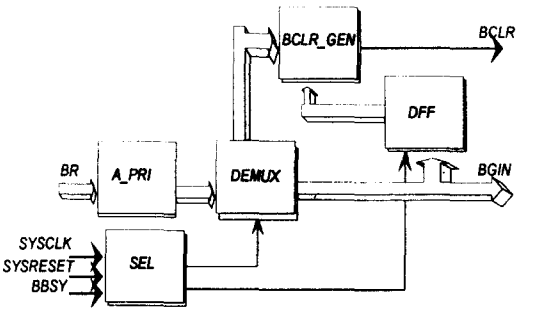
〈Table 2〉 The condition of BCRL signal generation

BCLR*STATE		Pending bus request level			
		3	2	1	0
Level of Current Master	3	1	1	1	1
	2	0	1	1	1
	1	0	0	1	1
	0	0	0	0	1

0 = assert, 1 = negate

In here, Arbiter permits use of bus(BGIN) after received the requested bus(BR) from each master. The blocks of BCLR_GEN generates BCLR signal that stops use of bus when higher order is received to compare the priority order between requested order presently and came newly. Because the operating is duplicate decision logic of priority order is using reduce the circuit size.

The logic of BCLR_GEN compares ed order presently and came newly for the generation of BCLR signal. The remainder part is the control logic for enabling DFF block. The pin function of (Fig. 2) is described in 〈Table 3〉.



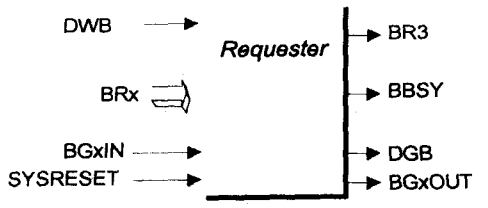
(Fig. 2) Block diagram of arbiter proposed

〈Table 3〉 Pin definition and there operation specification of (Fig. 2)

BR(bus request)	The bus request lines are used by each requester to request use of the DTB. They allow the arbiter to award use of bus by driving a bus grant daisy-chain line low. This low level propagates to down the daisy-chain, typically passing through several boards in the process. If a board never uses a particular request/grant level, the signal is passed through that board.
BBSY (bus busy)	Once a requester has been granted control of the DTB by the way of bus grant daisy-chain, it drives BBSY low. Then it has control of the DTB until it released BBSY, the release of BBSY allows the arbiter to grant the DTB to the some other requester.
BCLR (bus clear)	The current master is not required to relinquish the bus within any prescribed time limit. It can continue transferring data until it reaches an appropriate stopping point, and then allow its on-board requester to release BBSY.
BGIN(bus grant)	Where the board uses a request/grant level x, the corresponding signal BGxIN is gated on board. If its on-board requester is currently requesting the DTB on that level, it does not pass the low level to its BGxOUT. Otherwise, it is passed on the low level.

3.2 Requester

Requester receives the internal bus request signal (DWD) form Interrupt handler or Master and then sends bus request signal(BR3) to Arbiter for hand shaking. In Revision C.1(IEEE Std. P1014-1987), it supports 4 request level(BRx; x = 1~3). In this paper, we consider only BR3 and the composed Requester is shown in (Fig. 3).

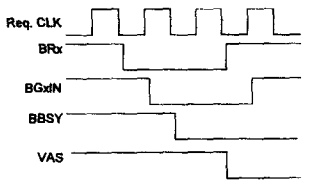


(Fig. 3) Block diagram of requester proposed

To observe the defined operation of each signal in (Fig. 3), DGB is permitting the signal of use of bus from Requester to Master and Interrupt handler. Also DGB is defined by BGxIN(x=1~4) that is transmitted about daisy-chain type from Arbiter. BBSY is an acquainting signal and Requester is in use of bus.

According to negate the point of BBSY signal, it is classified 3 algorithms such as RWD(Release-When-Done), ROR(Release-On-Request) and FAIR Requester. By the way, we chose ROR method to negate the BBSY signal when other Requester is drove after the Master was negating DWB signal. And, for the flexibility of this operation, we designed additionally the logic circuit block that generates BBSY signal from BRx signal. And VAS confirms that BERR may be high.

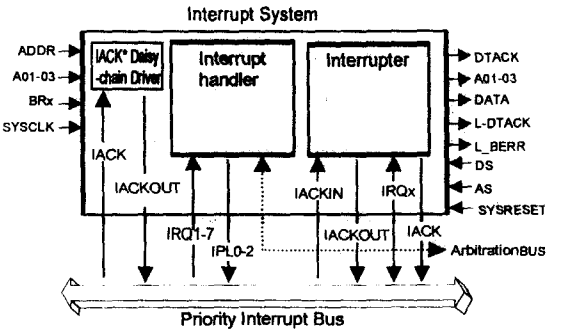
So it is easily operated about the abandoned use of bus that linked the relation of all structural methods. The timing of this operation is shown (Fig. 4).



(Fig. 4) Timing chart of requester

3.3 Interrupt system

In (Fig. 5), Interrupt system of VMEbus designed is composed of the Daisy-chain driver, Handler and Interrupter. Handler received IRQx from Interrupter that located in function module, and then send IPL0-2 to processor. Also Handler sends Interrupter a related signal of interrupt such as STATUS ID and DTACK those are generated by processor. Then STATUS ID and DTACK arbitrate use of interrupt to fit the priority order. After Interrupter received IACKIN signal that is transmitted from Daisy-chain driver and then compares the interrupt priority order that is sent by self.



(Fig. 5) Block diagram of interrupt system proposed

If IACKIN signal and the interrupt priority order are same then Interrupter operates the interrupt routine else, sends IACKOUT signal to next module. Daisy-chain driver is operated by DS0 and IACK signal, and then IACKOUT signal is also negated when AS signal is negated. IPL signal is wholly generated by IRQ1~7 through Interrupt handler. For the generation of IPL signal, IRQ boolean equation is presented in (Fig. 6). And STATUS ID is considered only DS0 signal using 8 bit.

$$IPL2 = \overline{}IRQ7 + IRQ7 \cdot \overline{}IRQ6 + IRQ7 \cdot IRQ6 \cdot \overline{}IRQ5 + IRQ7 \cdot IRQ6 \cdot IRQ5 \cdot \overline{}IRQ4$$

$$IPL1 = \overline{}IRQ7 + IRQ7 \cdot \overline{}IRQ6 + IRQ7 \cdot IRQ6 \cdot IRQ5 \cdot IRQ4 \cdot \overline{}IRQ3 + IRQ7 \cdot IRQ6 \cdot IRQ5 \cdot IRQ4 \cdot IRQ3 \cdot \overline{}IRQ2$$

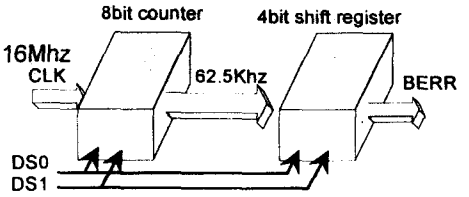
$$IPL0 = \overline{}IRQ7 + IRQ7 \cdot IRQ6 \cdot \overline{}IRQ5 + IRQ7 \cdot IRQ6 \cdot IRQ5 \cdot IRQ4 \cdot \overline{}IRQ3 + IRQ7 \cdot IRQ6 \cdot IRQ5 \cdot IRQ4 \cdot IRQ3 \cdot IRQ2 \cdot \overline{}IRQ1$$

(Fig. 6) IPL signal generation

3.4 Bus timer

Bus timer generates BERR signal automatically when the decided bus timing is over and then sends BERR signal to the system module that uses bus for prohibiting the occurring deadlock. That is, master

board starts VMEbus cycle, at that time, Bus timer is drove and then it must finish VMEbus cycle. In this paper, we designed Bus timer that is operated on 16MHz that is divided 32MHz system clock by 2. This is shown in (Fig. 7).



(Fig. 7) Block diagram of bus timer

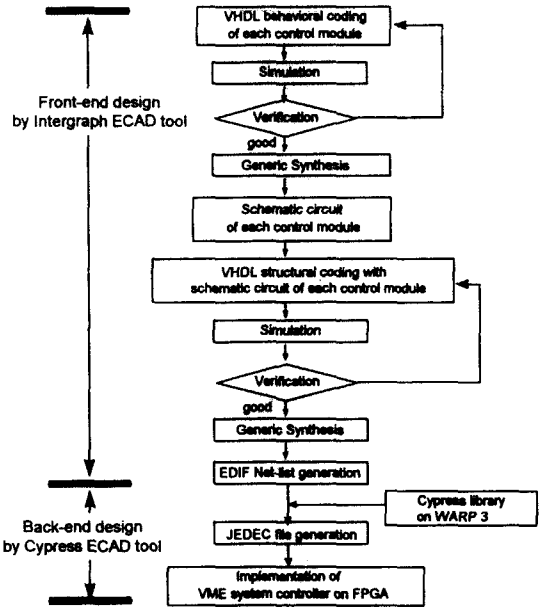
Firstly, for BERR signal generation, we used 8 bit counter to divide the system clock by 256 and so 62.5KHz is generated. Again using 4 bit shift registers, if AS signal is low continually during 56μs then Bus timer is driving BERR signal and master board must stop VMEbus cycle. And DS0 signal is used be control signal to monitor the state of use of bus. This operation of condition characteristics is described in (Fig. 8).

1. Start of Timer is that VAS is drove on time.
2. If Time out signal is occurred, BERR is driving.
3. BERR is driving continually until VAS will be high.

(Fig. 8) Timer operation of condition characteristics

4. Design Implementation and Experimental Results

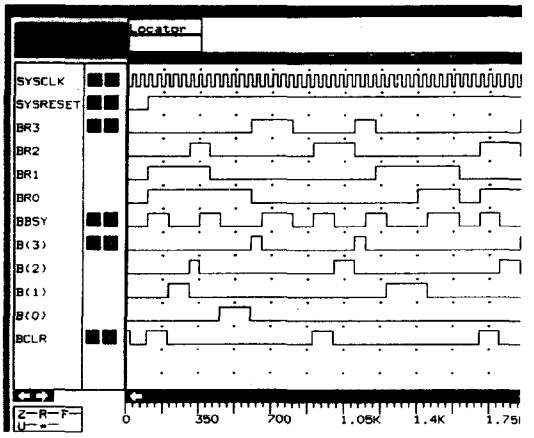
In this section, we designed VME system controller for the integrated environment on Slot 1 to fit Revision C.1(IEEE Std. P1014-1987). VME system controller designed is simulated using Intergraph ECAD tools after VHDL coding with mid-fixed description method, and through synthesize and verification then implemented on FPGA.



(Fig. 9) Design flow chart for VME system controller proposed

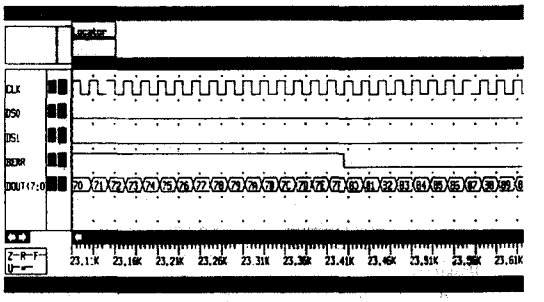
First of all, as front-end, we partitioned each control modules into the functional blocks then simulated after described with VHDL behavioral coding. Then each control modules are synthesized after behavioral verification. As next stage, all of the synthesized modules of the schematic circuits are coded VHDL to the relationship of structural connection again. The structural description coding is synthesized after structural verification, and then the schematic circuit of the designed VME system controller is generated with EDIF net-list. Lastly, as back-end, we implemented this net-list on FPGA using Cypress library of WARP 3 tool. This design flow chart is shown in (Fig. 9).

Simulation is proceeding to 2 times functionally, each other control module that is the functions of bus arbitration and interruption. In (Fig. 10), simulation of bus arbitration of Arbiter is represented. As a result, the correct priority order is defined as Requester signal(BR) of the other level and BCRL signal is drove to fit the priority order algorithm.

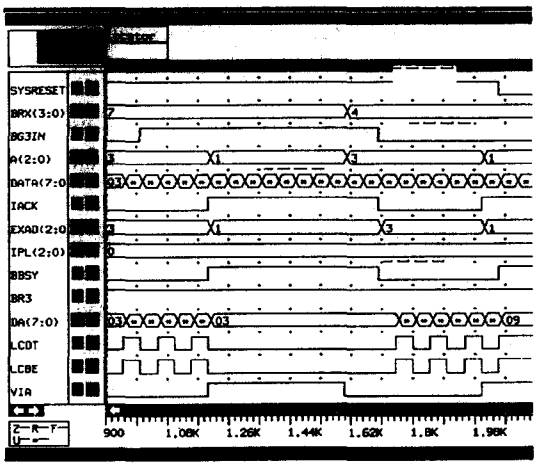


(Fig. 10) Timing chart of bus arbitration

As another result, BERR signal is drove when DS signal is not negated for 56μs through the states(DS0, DS1) of Bus timer in (Fig. 11). And Arbiter circuit synthesized is shown in (Fig. 12). In the operation of interrupt duration, timing chart of Interrupt handler is shown in (Fig. 13). And the interrupt duration is started when Interrupter handler is receiving the request signal(IRQ) from Interrupter. Within the interrupt duration, we confirmed that if the permission signal of interruption(DWB) is going to Requester then processor is sending STATUS ID to Interrupter after receiving use of bus permission(BR, BGIN). And part of Interrupter circuit synthesized is shown in (Fig. 14).

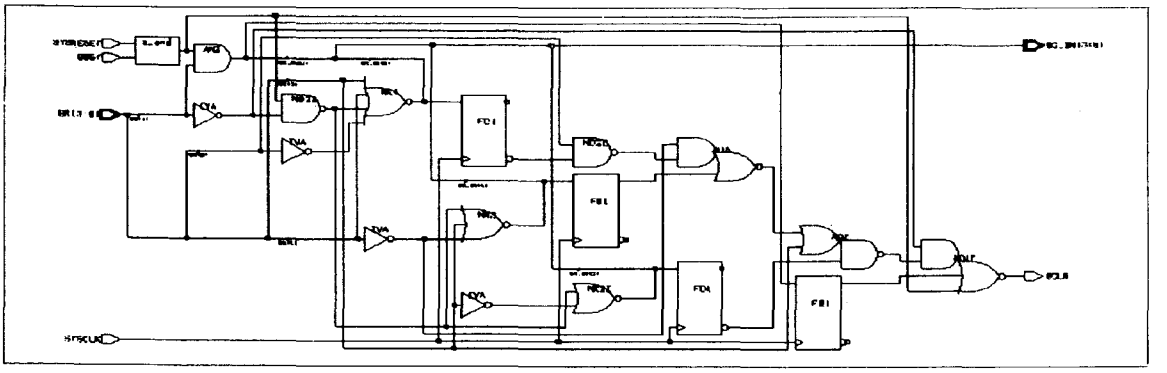


(Fig. 12) Schematic circuit of arbiter

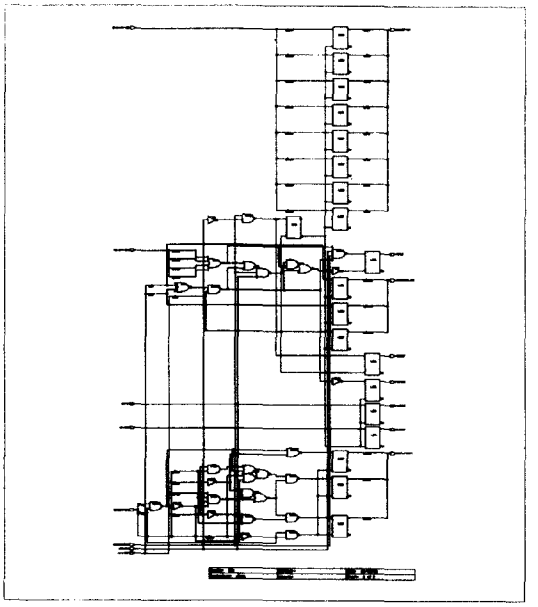


(Fig. 13) Timing chart of interrupter

As a result of implemented on FPGA, it necessitated the number of 412 cells and 36 FFs on Cypress



(Fig. 11) Timing chart of bus timer



(Fig. 14) Schematic circuit of interrupter

7C382A. The compared characteristics of VME system controller between the typicalness and implementation is shown in 〈Table 4〉.

In 〈Table 4〉, The implementation of VME system controller is satisfying thus the result of this paper will be customize to ASIC afterward.

5. Conclusion

In this paper, we integrated the modules of VME system controller which are laid on the Slot 1 or other scattered Slots. And most modules of VME system controller are designed and implemented on FPGA one-chip even into Slot 1. VME system controller is behaviorally and structurally coded with VHDL mid-fixed description method. And through the verification that is considered with the reality delay time of synthesized circuits, we confirmed that the function of each module has correct operation. Especially, bus error signal is correctly occurred within 56μs. The designed system is satisfied with Revision C.1(IEEE Std P1014-1987) for VME system controller. And this design can be the construction of general library. Thus, on ASIC design using a result of this paper, it can reduce the error, cost and time of design so widely apply to the

〈Table 4〉 Comparison between typical and the implementation of this work

Item	Typical		This work		Remark
Control and function modules	Arbiter Daisy chain driver Bus timer	Slot 1	Arbiter Daisy chain driver Bus timer Interrupt handler Interrupter(Slot 2) Requester Interrupter(Slot 3)	Slot 1	Satisfaction of the Integrated Environment
	Interrupt handler Interrupter Requester	Slot 2			
	Interrupter	Slot 3			
System clock	Above 20MHz		32MHz		High speed
Arbitration	Single level Arbitration Priority Arbitration Round robin Arbitration		Priority Arbitration		Excellent
Error detection	BERR signal		Use of BERR signal		Possible
Occurred time of BERR signal	56 μs		56 μs		While DS0 and DS1 are not negate, BERR signal is drove correctly.
Specification	Revision A		Revision C.1(IEEE Std. P1014-1987)		Newly revision

system based VMEbus.

Acknowledgement

The author wish to thank the hidden reviewers for a careful reading of the manuscript, Research Cheol Seo, Hitron system Co. for the experimentation and Prof. Yong Deak Kim, Ajou Univ. for valuable advice and help.

References

[1] Wade. D. Peterson, 'The VMEbus Handbook,' A. VITA Publication May 1992.

[2] Werner Hilf and Anton Nausch, 'The M68000 family,' Vol. 2, Prentice Hall, 1991.

[3] VFEA International Trade Association, 'The VMEbus Specification,' A. VITA Publication.

[4] M. Bullock, "Designing SMT Boards for In-Circuit Testability," IEEE Proc. Int'l. Test Conf., pp. 606-613, Sept. 1s987.

[5] K. P. Parker, "Integrating Design and Test: Using CAE Tools for ATE Programming," IEEE Computer Society Press, Los Alamitos CA 1987.

[6] 김용득, "VMEbus용 시스템 제어기 설계 연구", 대한전자공학회 추계학술대회 논문집, 17권 2호, 1994.

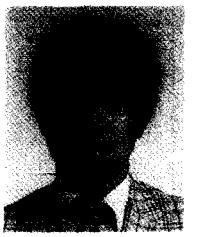
[7] 김용득, "VMEbus용 시스템 제어기를 이용한 실시간 시스템 구현에 관한 연구", 대한전자공학회 추계학술대회 논문집, 18권2호, 1995.

[8] Armstrong and Gray, 'Structured Logic Design with VHDL,' PTR Prentice Hall, 1993.

[9] Douglas E. Ott and Thomas J. Wilderotter, 'A Designers Guide To VHDL Synthesis,' ITTT AVIONICS, 1995.

[10] Douglas J. Smith, 'HDL Chip Design,' Foreword by Alex Zamfirescu, Doone Publication, 1996.

[11] M. A. Breuer and J. C. Lien, "A test and maintenance controller for a module containing testable chips," Proc. Int'l. Test Conf., pp. 502-513, 1988.



이 강 현

1977년 조선대학교 전자공학과 학사

1981년 조선대학교 대학원 전자공학과 석사

1991년 아주대학교 대학원 전자공학과 박사

1977년~현재 조선대학교 전자, 정보통신공학부 교수

1991년 미 스탠포드 대학교 CRC 협동연구원

1996년 12월~1997년 2월 호주 Wollongong 대학교, Sydney 대학교 SEDAL 방문교수

1995년~1996년 조선대학교 RRC 산학협력실장

1996년~현재 조선대학교 전자, 정보통신공학부 학과장

관심분야: 멀티미디어 VLSI 시스템 설계

배 상 현

1982년 조선대학교 전기공학과 졸업(공학사)

1984년 조선대학교 대학원졸업 (공학석사)

1988년 일본 동경도립대학 정보통신공학부(공학박사)

1984년~1985년 일본 동경공대 객원 연구원

1995년~1996년 일본 과학기술원 객원교수

1997년~현재 조선대학교 전산통계학과 부교수

관심분야: 퍼지시스템, 유전자 알고리즘, 전문가시스템, 인공 신경망, 로보틱스, 지식처리