World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 3.419

DESIGN AND VERIFICATION OF PRIORITY CONFIGURABLE INTERRUPT CONTROLLER

Mukthi. S. L[1*] and Dr. A. R. Aswatha[2]

[1]Department of Electrical and Electronics Engineering, Jain University, Bangalore, Karnataka, India.

[2]Department of Electronics and Communication Engineering, DSCE, Bangalore, Karnataka, India.

Article Received on 23/08/2016 **Article Revised on 13/09/2016** **Article Accepted on 02/10/2016**

***Corresponding Author**
Mukthi. S. L.
Department of Electrical and Electronics Engineering, Jain University, Bangalore, Karnataka, India.

ABSTRACT

Interrupt helps in communicating the processors and peripherals. But number of interrupt ports on processor are far less compared to interrupt signals from other processor and peripherals in a system on chip (SOC) design. This creates an serious issue in the multi-processor SOC design also. Interrupt Controller is the key solution to above problem. The function of Interrupt Controller is to sort the interrupt signals from peripherals and processor based on priority level and put forward the interrupt which has got highest priority for processing. The priority level assignment can be configured. In this proposed design, priority level is configured by software. The proposed design is also combining interrupts and then assert the interrupts to the processor. Our design can handle up to 60 interrupt signals from various peripherals and produces the 12 output signals. The AMBA AHB act as bus interface between processor and Interrupt Controller Formal verifications of the proposed Interrupt controller has been done.

Software used

Operating system: Redhat.

Simulation: Cadence (Design and Verification), Verilog.

KEYWORDS: *AMBA, AHB, SOC, ARM.*

I. INTRODUCTION

In recent years the usage of many peripherals and multiprocessor in SOC design is becoming large and lager. But as the number of peripherals increases the communication between the processor and peripherals become difficult, since there are limited interrupt ports as the processor.

An interrupt controller is need to overcome this problem. Interrupt controller receives the interrupt signals form peripherals and assign the priority to each individual and asserts the expected interrupt to the processor.

Formal verification is used to verify whether the particular design meets the functional requirement of the design specifications. Simulation based technology can be used for functional verification of the design.

Code coverage analysis is used to describe the extent to which the source code of a program is tested by specific test suit. The analysis can be done Register Transfer Level(RTL) of the design. The priority configurable concept and interrupt combinable design is proposed in this paper. The AMBA AHB acts as bus interface between Interrupt controller and processor. Formal verification of the design, complete code coverage and false paths eliminations was done. This paper is organized as followes.Part II: Description of Interrupt Controller Architecture. Part III: The Simulation and Synthesis result. Part IV: Conclusion.

II. ARCHITECTURE

The proposed Interrupt controller has 3 main components as shown in Fig.1.[3]

- AHB controller
- Register set
- Interrupt management logic unit

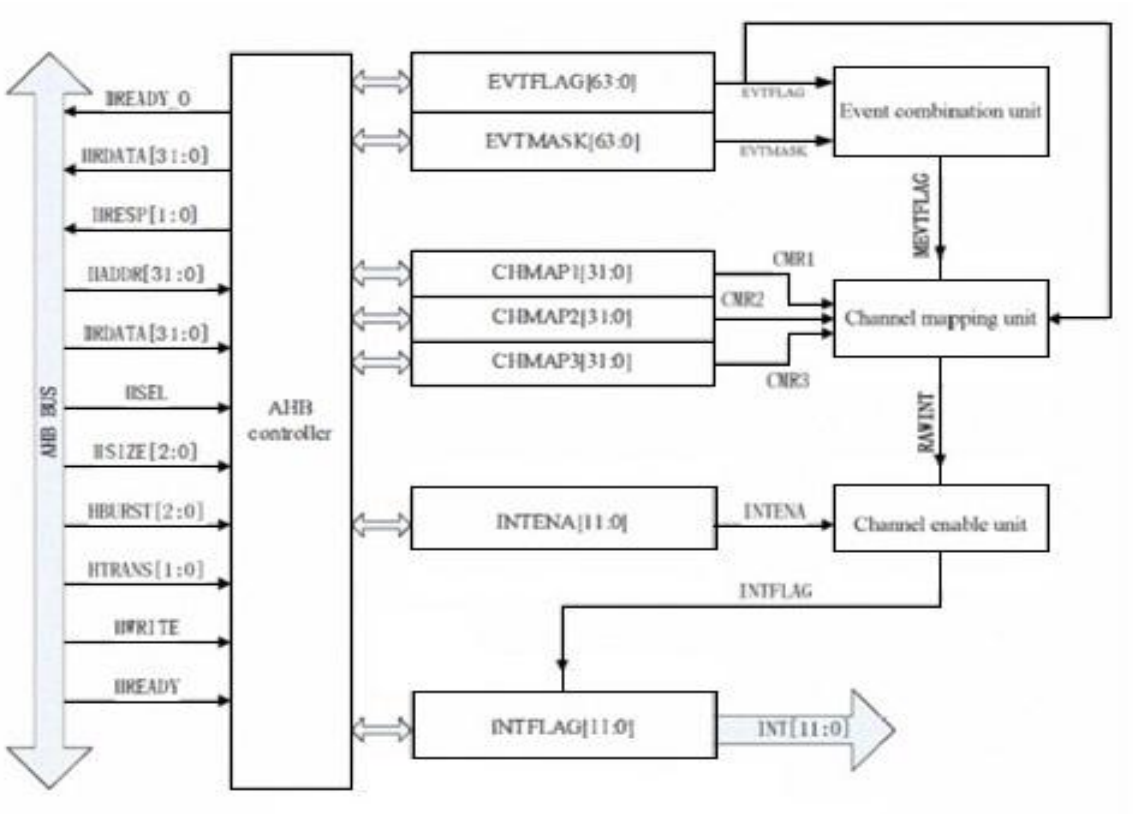

Fig 1: Interrupt Controller structure.

AHB controller processes the interrupt controller. The information about interrupt inputs, combined interrupts and mapping configuration is stored in the register set. Interrupt management logic unit is involved in combining and mapping of interrupts.

A. AHB controller

On-chip communications standard which is required for designing high-performance embedded microcontrollers is provided by AMBA specification of ARM. The AHB acts as the high-performance system backbone bus which is mainly used for high-performance, high clock frequency system modules. It provides high-bandwidth operation and supports efficient connection of processors and supports multiple bus masters.[2] Our Interrupt controller is interfaced to the processor through standard AHB bus. Accessing of registers in the interrupt controller by the processor is done through the AHB bus only.

B. Register Set

The proposed Interrupt controller has a set of registers named, Event Flag Register, Event Mask Register, Channel Mapping Register, Interrupt enable registers and Interrupt Flag Registers. The AHB controller establishes the communication to these register set by the signals which is given below.

Wdata/Rdata: These are input/output for register set. The data write/read is done by the processor.

regaddr: address signal to access the registers.

wr_valid/rd_valid are enable input signal for register set. These signals can be set and clear to enable writing read operation.

Event flag set**(EVTFLAG_SET)** and Event Flag clear registers**(EVTFLAG_CLR)** are Virtual registers They can be set and clear. If the processor needs to write to the Event Flag Register, the corresponding bits of Event flag set register are set to logic one. Interrupt combination is done by Event Mask Register**(EVTMASK)**. Interrupts can be included in combination interrupt by setting the bits of Event Mask Register to logic '0'. Masked event flag registers store the status of interrupts. By checking these registers can know which interrupts generates a combination interrupt.

Channel mapping register: It is a set of three registers used for channel mapping. Each channel has 8 bit interrupt output which represents the interrupt mapped on to the corresponding channel.

INTENA signal enables the channels to put forward interrupt request to processor. **INTFLAG** is a read only registers which stores the status of channeled interrupts. The Setting or Resetting of bits of this register tells about the asserted interrupts. Only enabled channel interrupt can set or reset the corresponding bit of INTFLAG. Configuration of INENA register is done by Software, through AHB bus interface.

C. Interrupt Combination Logic

Fig. 2 depicts the Event combination unit. EVTFLAG[3:0] is used as combination interrupt. The 60 interrupt inputs are connected to EVTFLAG[63:4]. In this design, EVTFLAG[0] is generated by EVTFLAG[15:4], EVTFLAG[1] is generated by EVTFLAG[31:16], EVTFLAG[2] is generated by EVTFLAG[47:32], EVTFLAG[3] is generated by EVTFLAG[63:48]. Event combination unit in Fig. 1 services for the interrupt combination function[3].

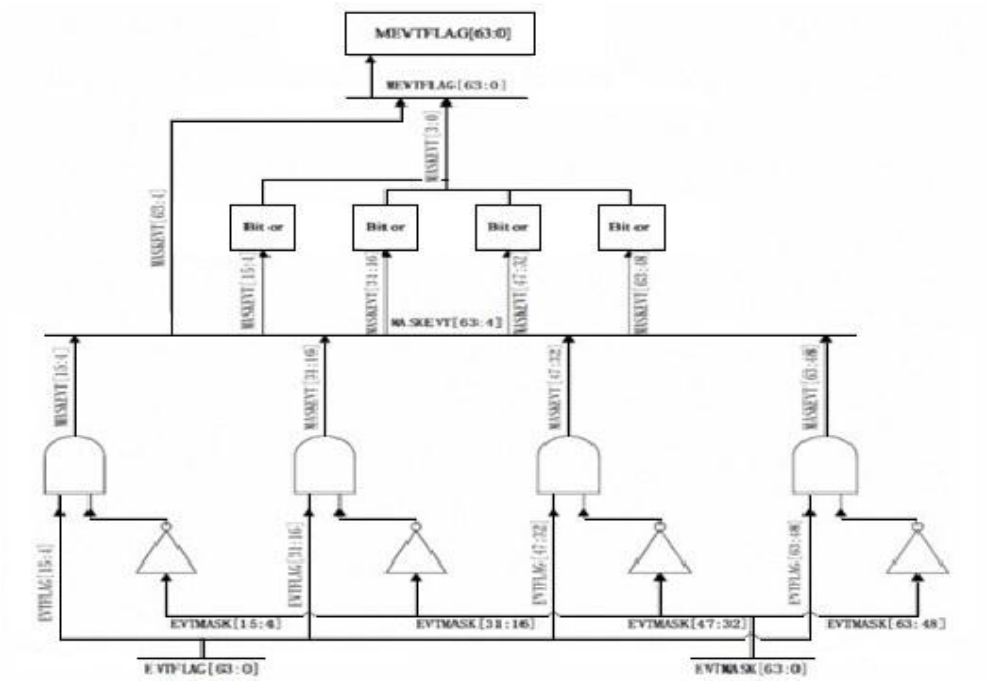

Fig.2 Interrupt Combination Unit.

The Fig. 3 illustrates the diagram of interrupt combination unit.

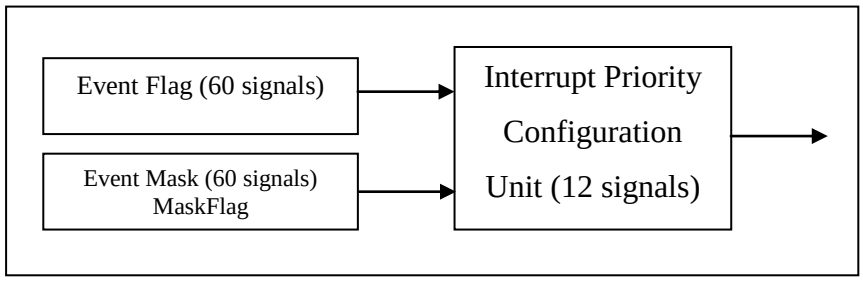

Fig 3: Block Diagram of Interrupt Combinational Unit.

The signals (EVTFLAG, EVTMASK) from the event flag and event mask are collected and combined to produce four common interrupt signals MEVTFLAG[3:0].

Event mask signal is first inverted and then it is anded with Event flag signal to produce masked interrupt signals (MASKEVT). MASKEVT is the interrupt masked by EVTMASK register. These masked interrupt signals are then combined by using Bit OR logic which finally gives the combined interrupt signal by reducing the number of interrupt signals. The design of the combination unit is depicted in the Fig. 2. The output signal MEVTFLAG[63:0] shows the status of interrupts. The first four field MEVTFLAG [3:0] shows the combination Interrupts. From MEVTFLAG [63:4] fields shows the interrupt status masked by event mask.

The signal MASKEVT [63:4] is generated as follows.[3]

MASKEVT[63:4]=EVTFLAG[63:4]&(~EVTMASK[63: 4]) The MEVTFLAG[63:4] is same as MASKEVT[63:4].MEVTFLAG[3:0] is generated as follows.[3]

MEVTFLAG[0]=| MASKEVT[15:4]

MEVTFLAG[1]=| MASKEVT[31:16]

MEVTFLAG[2]=| MASKEVT[47:32]

MEVTFLAG[3]=| MASKEVT[63:48]

D. Interrupt priority configuration

Software is used for configuration of Priority of interrupt. In our design, there are total of 12 different interrupt priority levels from RAWINT[0] to RAWINT[11] where RAWINT[0] indicates the highest and lowest priority level any signal can be configured to any of the 12 priority levels. The block diagram of Interrupt Priority Configuration Unit is given in fig 4.

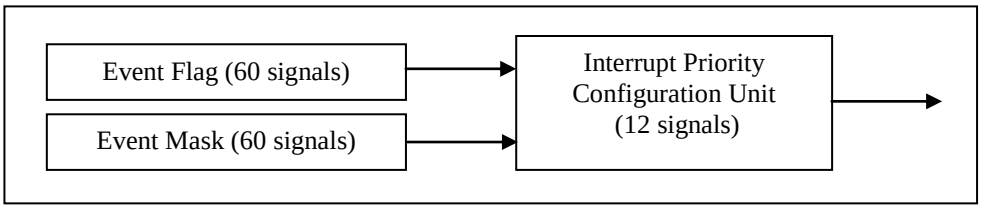

Fig. 4 Block Diagram of Interrupt Priority Configuration Unit.

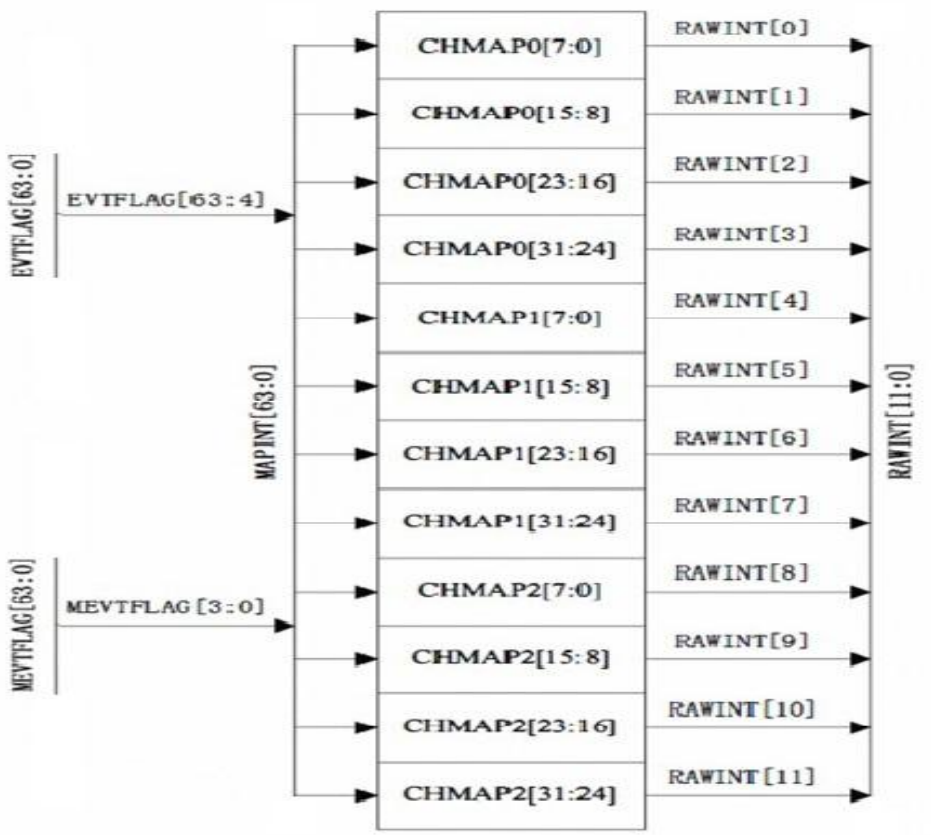

Fig 5: Structure of Channel Mapping Unit.

In this design, the signals from event mask and event flag are collected and combined to produce a 4 common interrupt signals. Then the signals are diverted to the desired interrupt level. A single level of interrupt can have a maximum of 8 interrupt signals. Here in this design multiplexer is used to assign the various interrupt signals to required levels. Then these signals are sent for the processor for their services. The channel mapping unit design is illustrated in Fig. 5. The channel mapping is realized by channel mapping registers(CHMAPX). EVTFLAG[63:4] is the interrupt inputs and combination interrupt MEVTFLAG[3:0] are the interrupt source for the channel mapping registers. Each interrupt channel of CHMAPx registers is an eight bit register. The connection between Interrupt channel and CHAMP registers is clearly shown in Fig. 5.

III. RESULTS

Fig 6: shows the circuit obtained through cadence.

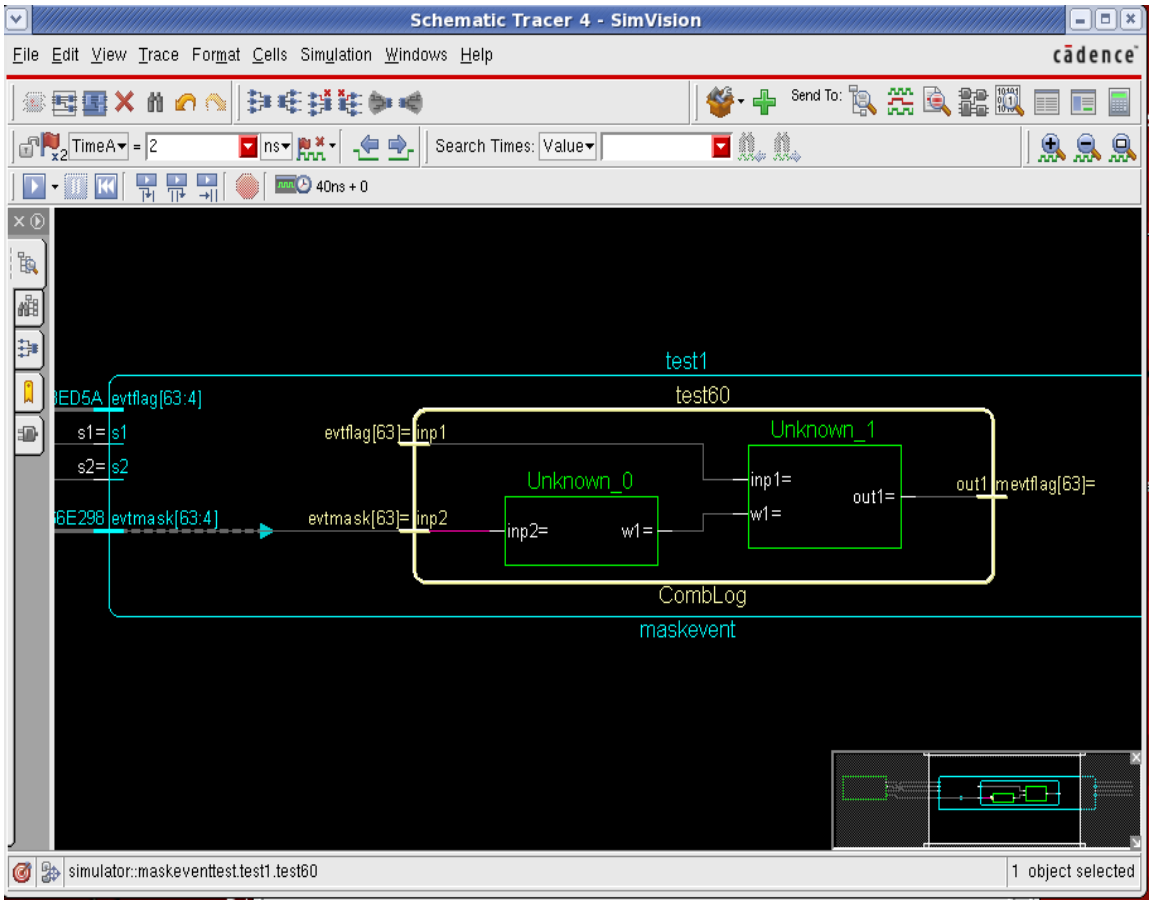

Fig 6: Screen Shot of Interrupt Combination Logic Block.

OUTPUT WAVEFORMS

OUTPUT OF INTERRUPT COMBINATIONAL LOGIC

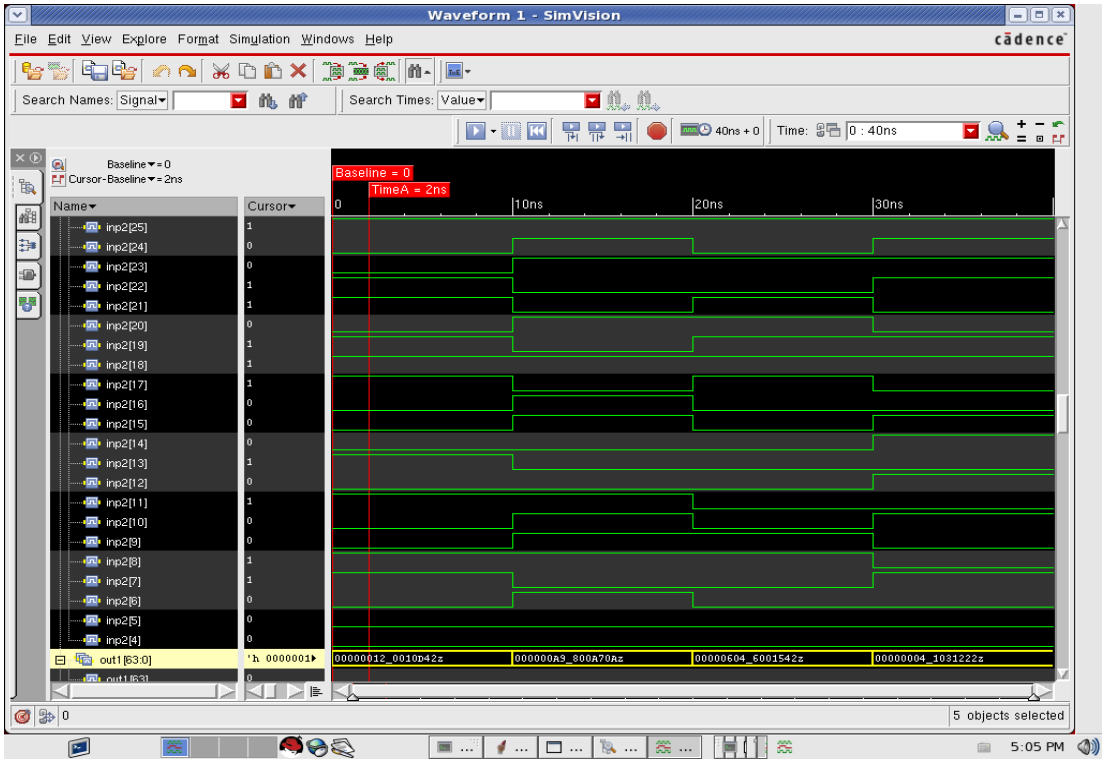

Fig 7: Screen Shot of Interrupt Combinational Logic input waveforms(event flag)

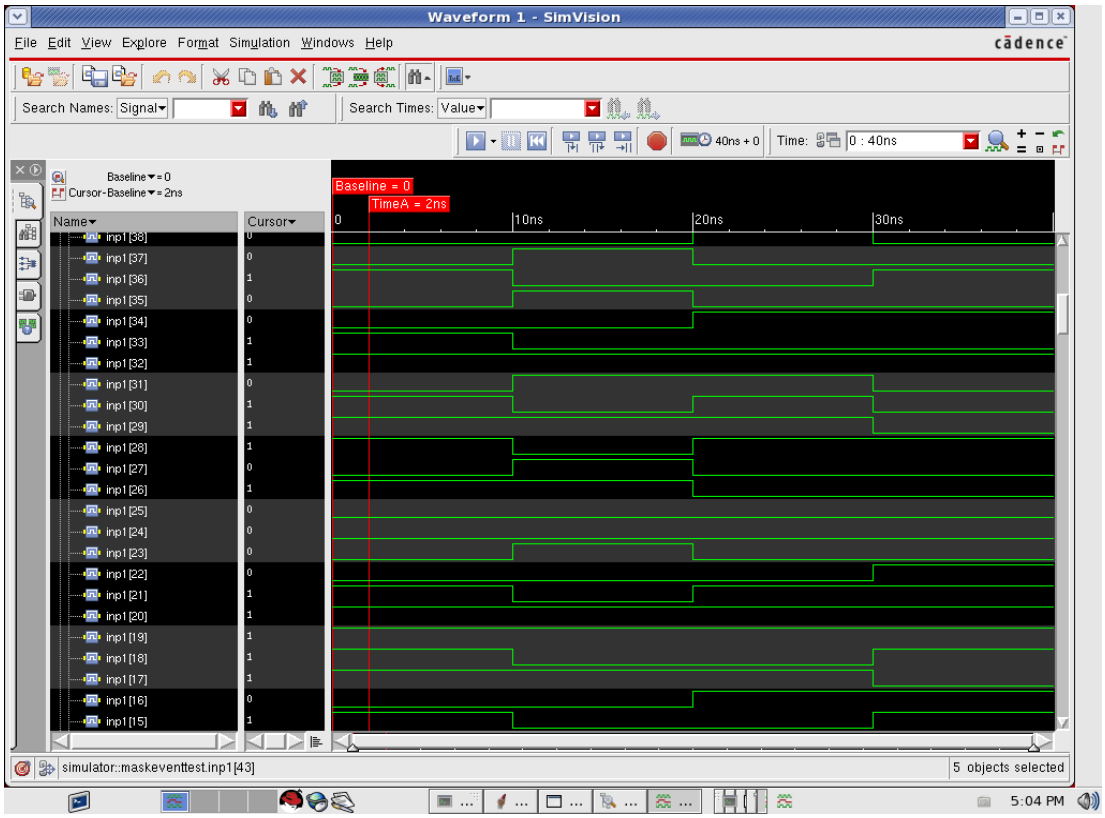

Fig 8: Screen Shot of Interrupt Combinational Logic input waveforms (event mask)

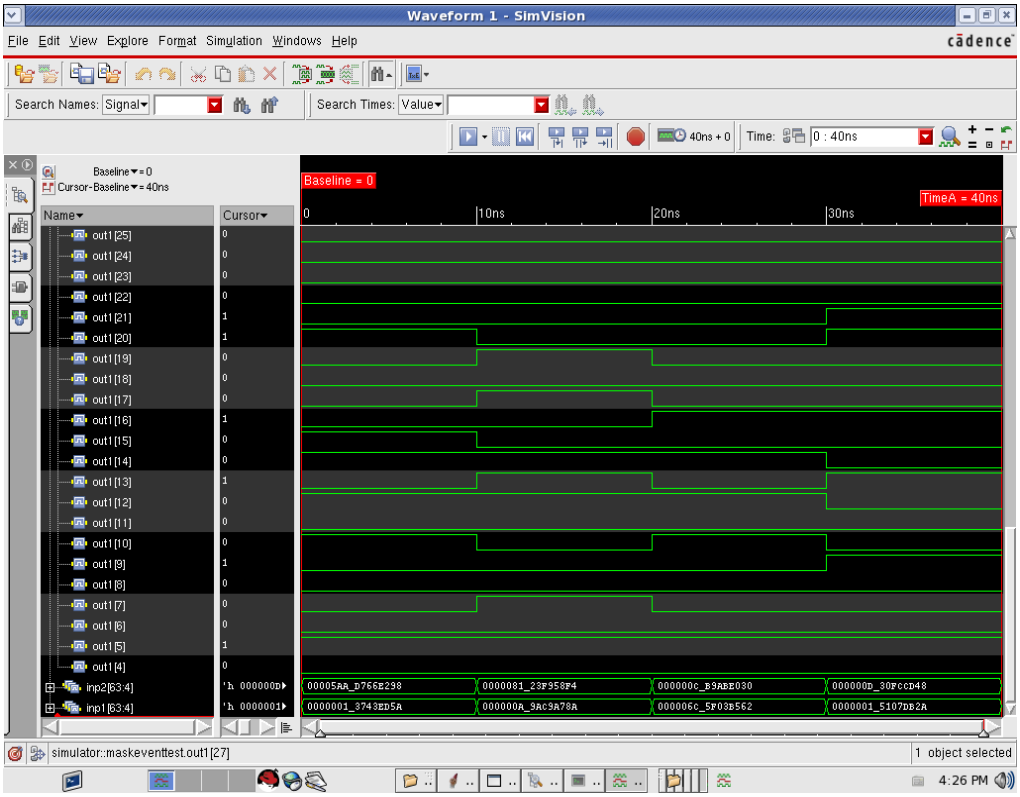

Fig. 9: Screen Shot of Interrupt Combinational Logic output waveforms.

OUTPUT OF INTERRUPT PRIORITY CONFIGURATION Output obtained from cadence tool.

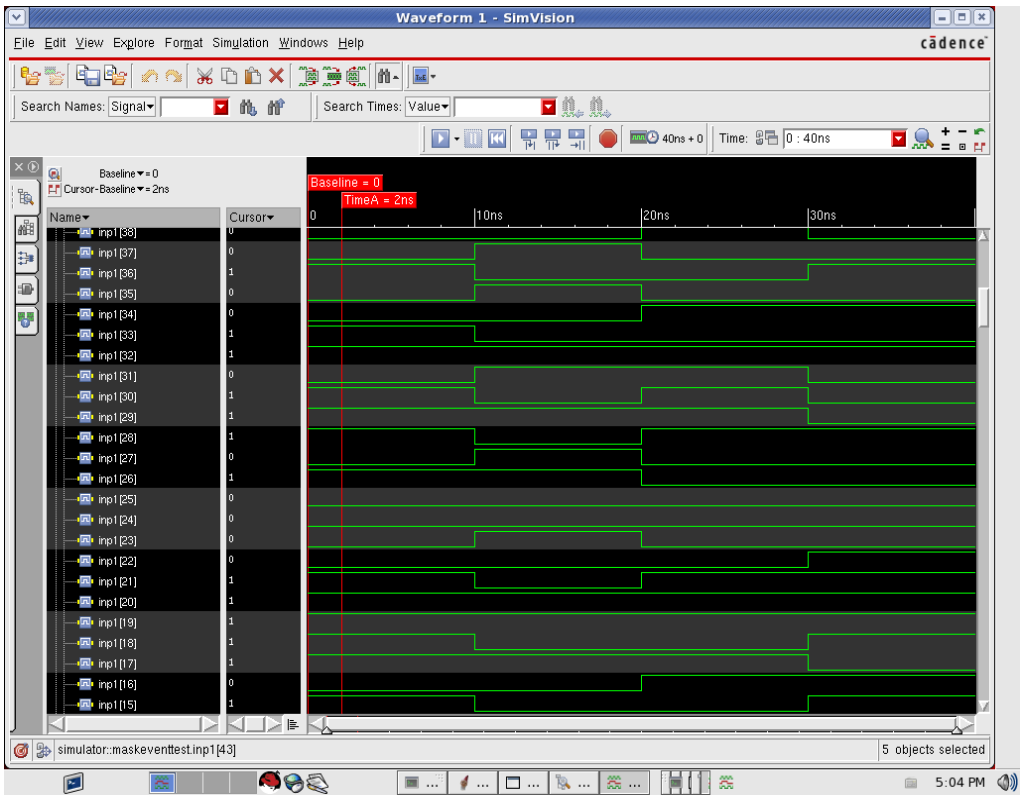

Fig. 10: Screen Shot of Interrupt Priority Configuration input waveforms (event flag)

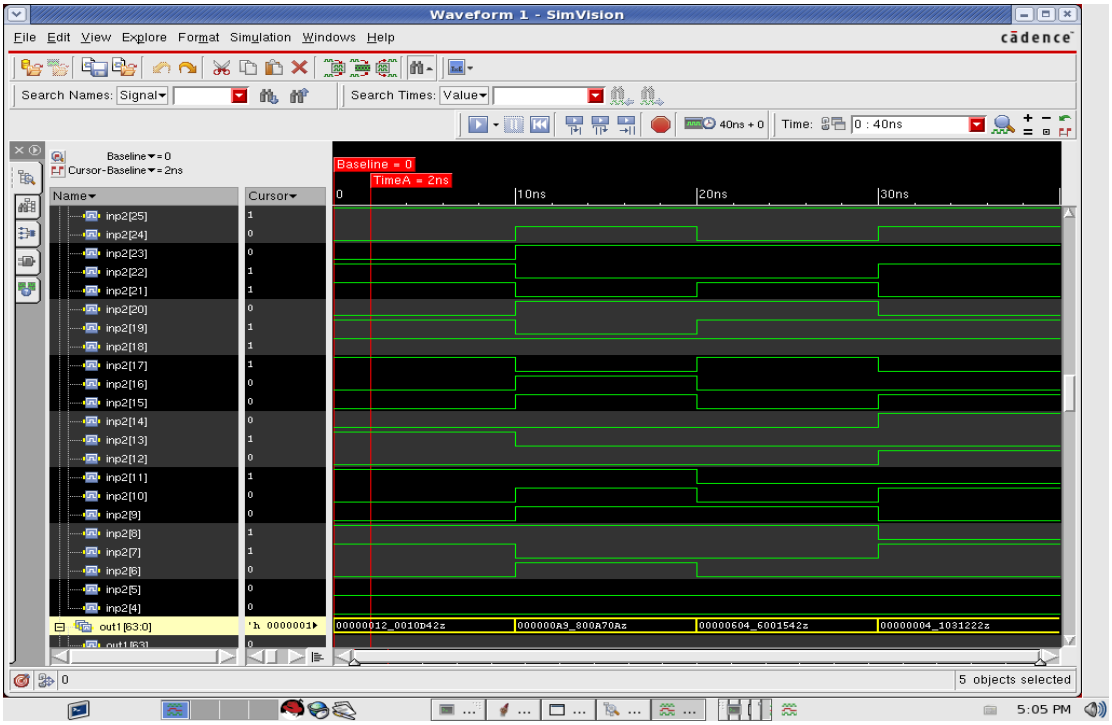

Fig. 11: Screen Shot of Interrupt Priority Configuration Logic input waveforms (event mask).

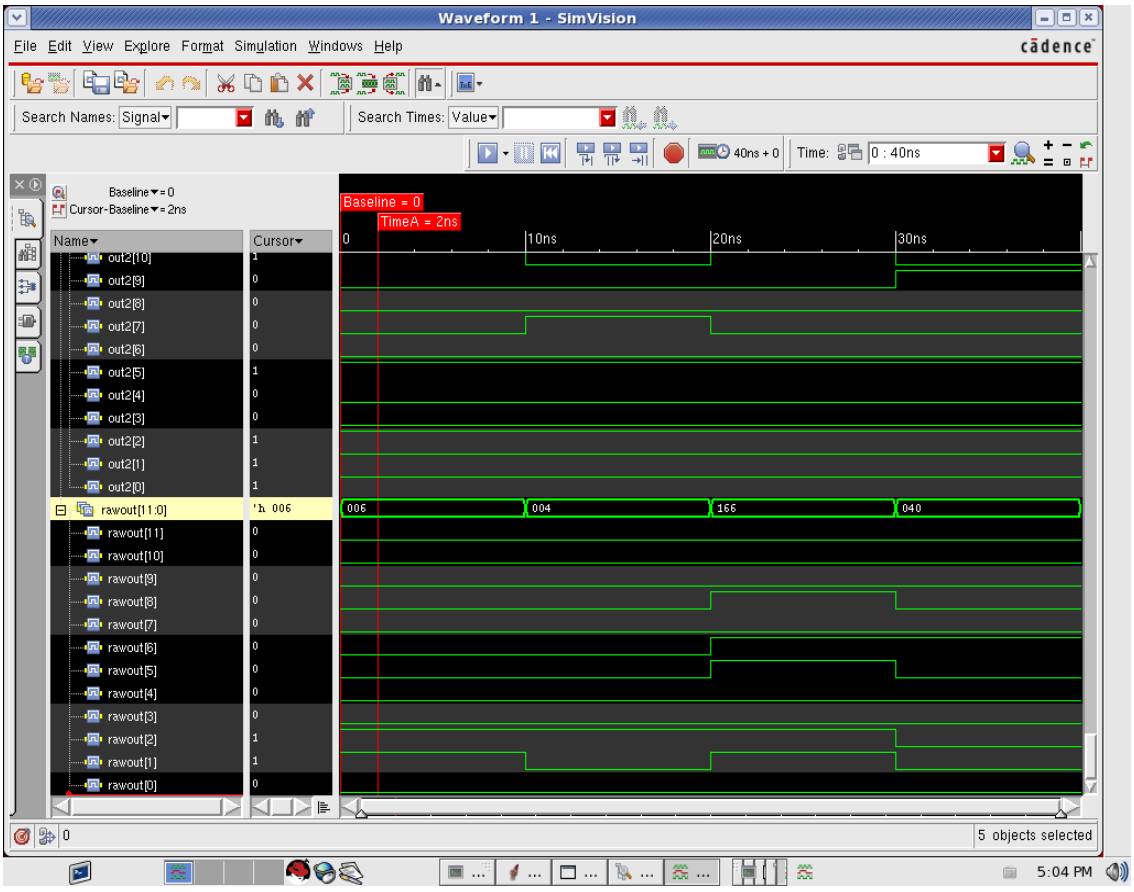

Fig. 12 Screen Shot of Interrupt Priority Configuration Logic output waveforms.

IV. CONCLUSION

In this paper, the Design of Priority configurable Interrupt controller is proposed. The proposed design can accept a maximum of 60 interrupt signals. Our interrupt controller have total of 12 different interrupt priority levels. Input interrupt signals are collected and combined to produce 4 common interrupt signals. Interrupt controller and processor communicate through AHB bus interface. Verification of the proposed design is done by writing test benches with various conditions were created. Test cases with variety of inputs were provided to get the expected results and all the lines in the code are covered. Complete code coverage and all false paths were eliminated to get a effective design. Implementation of the proposed design is done by using cadence tool.

REFERENCES

1. ARM Corporation, AMBA specification 2.0. Reference manual, 2006.
2. AMBA Open Specifications-ARM http://www.arm.com
3. Wei Chipin, Li Zheng Qingwei, Ye Jianfei.and Li Shenglong.” Design of a configurable multichannel interrupt controller”, 2010 Second Pacific-Asia Conference on Circuits Communications and System, 2010.
4. Digital Design Principles and Practises, John F.Wakerly, Third edition, Prentice Hall Publication, 2000.
5. J. Noseworthy” Efficiant communications between an Embedded processor and reconfigurable Logic on FPGA”.IEEE Transaction on Very Large Scale Integration (VLSI) Systems., August 2008; 16(8): 83-1090.
6. Nazeih M. Botros,”HDL PROGRAMMING(VHDL and VERILOG)”, Dreamtech Press, 2009 edition.