Vectored Interrupt Controller Implementation of Advanced Bus Architecture on FPGA

Vasanth H [1], Dr.A.R.Aswath [2]

[1] VLSI Design & Embedded systems, M.Tech,
DSCE, Bangalore-78,
hvasanth87@gmail.com
[2] Dept. of E&C, Professor, E&C
DSCE, Bangalore-78.

Abstract- Interrupt controller is designed with the concept of priority based selection of peripherals which requires immediate attention or service. Here AHB is optimized to interface with VIC to initiate data transfer on the AHB. Both read and write cycles are designed with AHB bus.

*Keywords***:** AMBA, AHB, Interrupt Controller, VIC.

1.   INTRODUCTION

As no of components or peripherals are increasing on a single chip design of system on chip is getting complicated. Design of System-on-Chip (SoC) receives a great deal of attention in recent days. Interrupt controller are important due to the fact that Processors and peripherals usually communicate with each other with interrupt [1]. With the development of SoC technique, the communication between processors and peripherals becomes a problem as processors have limited interrupt ports which is far less than the total interrupt signals of peripherals and other processors.
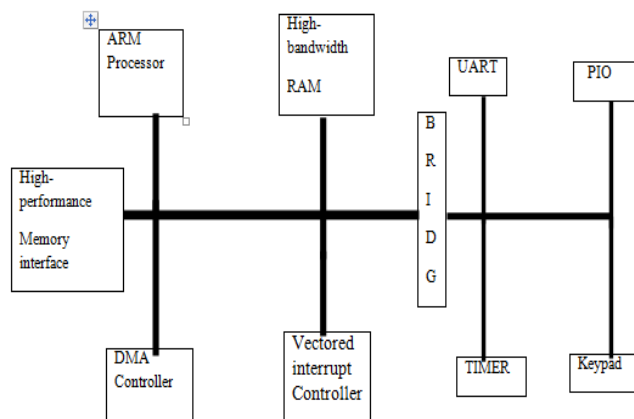

Fig1: An Interrupt controller in AMBA based system

An AMBA based microcontroller typically consists of a high performance system backbone bus, able to sustain the external memory bandwidth, on which the CPU on-chip memory and other vectored interrupt controller devices reside [2]. This bus provides a high bandwidth interface between the elements that are involved in the majority of transfers.

The AHB slave main function is an interface unit that allows AHB logic to initiate a data transfer on the AHB. The AHB specifies the type transaction to be executed on the slave through a user friendly interface.

The VIC is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) peripheral that is developed, tested, and licensed by ARM [3].

II. METHODOLOGY:

In this project it requires the usage of Verilog HDL for designing the RTL code. Simulation is done using Model Sim and corresponding synthesis is done with Xilinix 12.2v and finally implementing on SPARTAN 3XC3S 400 FPGA board.

III. AMBA Specifications

A.   Overview of AMBA Specifications:

The Advanced Microcontroller Bus Architecture (AMBA) specification defines an on-chip communications standard for designing high-performance embedded microcontrollers [3]. Three distinct buses are defined within the AMBA specification:
- Advanced High-performance Bus (AHB)
- Advanced System Bus (ASB)
- Advanced Peripheral Bus (APB)

B.   AMBA AHB:
AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs [3]. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation. AMBA AHB implements the features required for high-performance, high clock frequency systems including:
- burst transfers
- split transactions
- single-clock edge operation
- non-tristate implementation

A typical design contains the following components:

AHB master
AHB slave
AHB arbiter
AHB decoder

IV. INTERRUPT CONTROLLER

A.   Interrupt Controller:

It is a device used to combine several sources of interrupt onto one or more CPU lines while allowing priority levels to be assigned to its interrupt outputs. Interrupt with the highest priority level is asserted to processor for interrupt processing [4]. Many structures of interrupt controller have been proposed by former papers, as in [5], but they are neither priority configurable nor interrupt-combinable.

The VIC is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) peripheral that is developed, tested, and licensed by ARM. The VIC provides a software interface to the interrupt system. In a system with an interrupt controller, software must determine the source that is requesting service and

where its service routine is loaded. A VIC does both of these in hardware [6]. It supplies the starting address, or vector address, of the service routine corresponding to the highest priority requesting interrupt source.

B. Interrupts in ARM:

In an ARM system, two levels of interrupt are available:
- ◦ Fast Interrupt Request (FIQ) for fast, low latency interrupt handling

- ◦ Interrupt Request (IRQ) for more general interrupts.

C. Features of Interrupt Controller:

The features supported are:
- Uses the AMBA AHB protocol.
- Up to 32 interrupt source.
- High level sensitive, interrupt source type.
- Support for 32 vectored interrupts.
- Fixed interrupt priority level.
- Fixed IRQ and FIQ generation.
- Software interrupts generation.
- Interrupt enable.
- Raw interrupt status.
- Interrupt source get acknowledgment.

V. Design OF AMBA AHB INTERRUPT CONTROLLER

The vectored interrupt controller is mainly divided in to three blocks namely:
- Peripheral interface
- CPU interface
- AHB slave interface

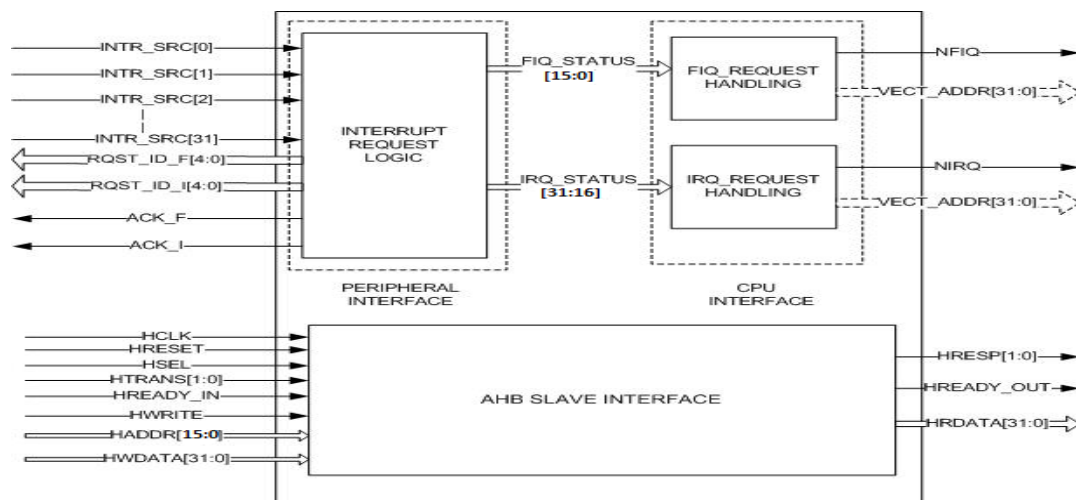The Block diagram of vectored interrupt controller is shown in figure 2

Fig 2: Block diagram of vectored interrupt controller

A. Peripheral Interface:

- ### Interrupt Request Logic:

Interrupt request logic receives 32 Intr_src lines from CPU peripherals and combines with the software interrupt which are written by CPU on Soft_Int register and enable the user selected interrupts by gated enabling and separate the 32 request lines into 16 fast interrupt request and 16 general interrupt request and also encode the filtered output generates two separate request id for FIQ's and IRQ's.
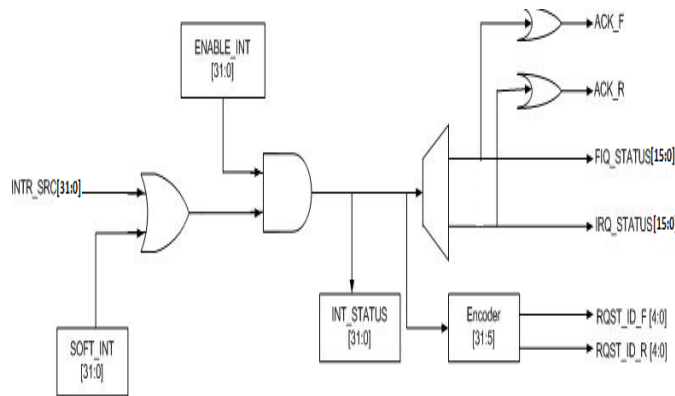

Figure 3: Interrupt request logic

B. CPU Interface:

- ### FIQ request handling:

Here we use fixed priority logic for lower 16 bits of Intr_src and generates the nfiq signals which is active low and selects the vector address of the respective peripheral from vectored table to the CPU.The FIQ request handling shown in figure 6 asserts the nfiq signal. i.e. if FIQ_status is nonzero, set the nfiq as low. It selects the vectored address of the corresponding fast interrupt request. Send it to CPU through AHB slave interface
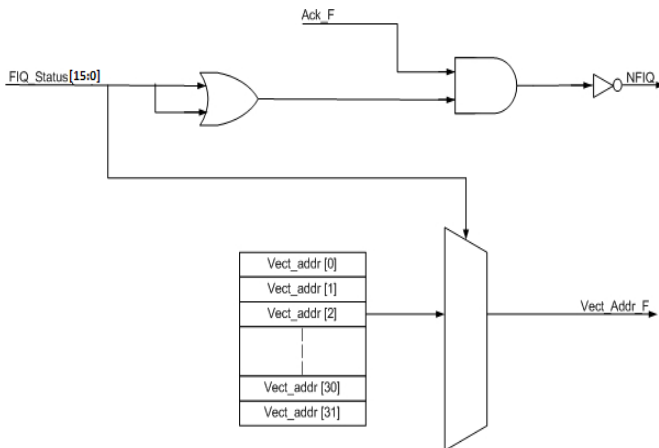
Figure 4:FIQ request handling

. It will select the vectored address from the vectored address table, vectored address table is the memory configuration space, which contain the subroutine of the each interrupt request. The vectored addresses in the vectored table are programmable. nfiq is active low signal for CPU.

- **IRQ Request Handling:**

Here we use fixed priority logic for lower 16 bits of Intr_src and generates the nfiq signals which is active low and selects the vector address of the respective peripheral from vectored table to the CPU.The IRQ request handling shown in figure 7 Asserts the nirq signal. i.e., if irq_status is nonzero, set the nfiq as low and selects the vectored address of the corresponding interrupt request. Send it to CPU through AHB slave interface. It will select the vectored address from the vectored address table, vectored address table is the memory configuration space, which contain the subroutine of the each interrupt request.

IRQ_status acts as a select line for vectored address selection. nirq is active low signal for CPU.
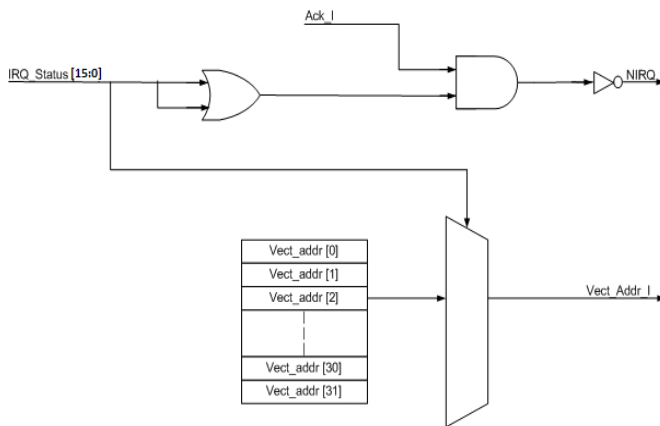

Figure 5:IRQ request handling

D. AHB Slave Interface:

An AHB bus slave responds to transfers initiated by bus masters within the system.

The AHB slave shown in figure 6 maps the memory configuration space with the interrupt controller and perform the data transaction as AHB asserts its signal. In this block asserts Hready_out as high and Hresp as OKAY, because we designed Interrupt controller as a single slave.

Figure 6: AHB slave interface

VI. SIMULATION RESULTS:

A. AHB Write Cycle:

Figure 7 shows how AHB signals got asserted to perform Data write through the AHB to Vectored Interrupt Controller. In order to write the Data Hwrite must be high. Data write to the address which is specified on the Haddr bus through HWdata bus

Figure 7: AHB Write Cycle

Red line on Hread signal is due to write operation being performed.

B. AHB Read Cycle

Figure 8 shows how AHB signals got asserted to perform Data read through the AHB from Vectored Interrupt Controller. In order to read the Data Hwrite must be

low. Data read from the address which is specified on the Haddr bus through HRdata bus.
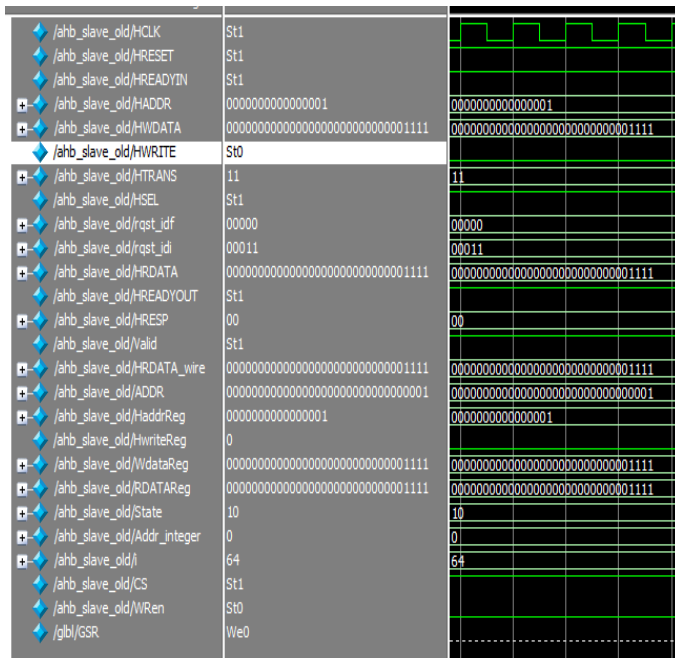

Figure 8: AHB Read Cycle

C.   Integrated Interrupt controller simulation:

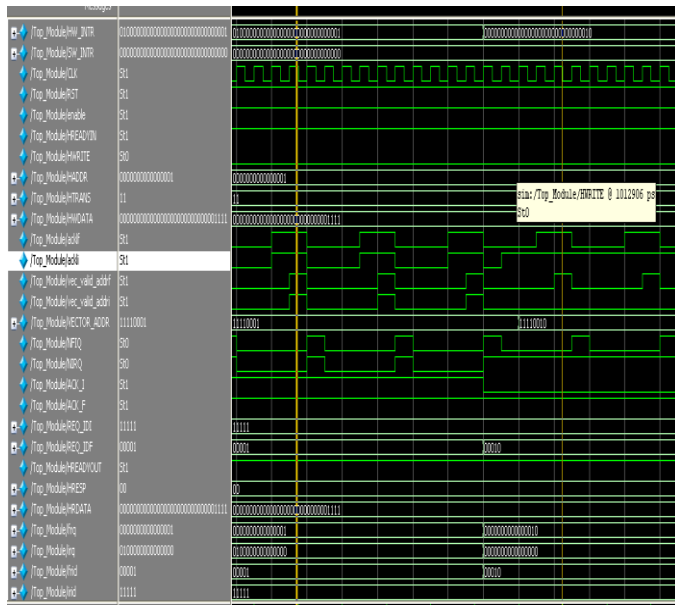Figure 9 shows how interrupt controller got asserted to perform generation of fiq, irq to ack signals from CPU.

Figure 9: Interrupt Controller

VII IMPLEMENTATION RESULT

Synthesis of the Integrated Interrupt Controller implementation is shown below with respect to small functions like gates, inverters large units like multiplexers, encoders [8].

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	88	7,168	1%	
Number of 4 input LUTs	137	7,168	1%	
Number of occupied Slices	107	3,584	2%	
Number of Slices containing only related logic	107	107	100%	
Number of Slices containing unrelated logic	0	107	0%	
Total Number of 4 input LUTs	199	7,168	2%	
Number used as logic	121			
Number used as a route-thru	62			
Number used for Dual Port RAMs	16			
Number of bonded IOBs	128	141	90%	
Number of BUFGMUXs	1	8	12%	
Average Fanout of Non-Clock Nets	2.22			

Figure 10: Implementation results

VIII CONCLUSION

Vectored interrupt can serve two interrupt request at a time, one FIQ request and one IRQ request and it will give two separate acknowledgements for both FIQ and IRQ requests. Due to the parallel processing of FIQ and IRQ requests the VIC has a low latency has a faster execution speed.
Finally interrupt controller is implemented and its results are discussed

VII. References

1. Wei Chipin,"Design of a Configurable Multichannel Interrupt Controller " 2010 Second Pacific-Asia Conference on Circuits, Communications and System (PACCS)
2. ARM Corporation, Prime cell Vectored interrupt controller PL192, reference manual, 2002.
3. ARM Corporation, AMBA specification 3.0, reference manual, 2006
4. Horelick, Dale; "Simple, Versatile CAMAC Crate Controller and Interrupt Priority ENC Oding Module "Nuclear Science, IEEE Transactions on Volume: 22 ,Issue: 1 Digital Object Identifier: 10.1109/TNS.1975.4327692 Publication Year: 1975,
5. De Gloria, A. Faraboschi, P.Olivieri, M.;"A self timed interrupt controller: a case study in asynchronous micro-architecture design", ASIC Conference and Exhibit, 1994. Proceedings, Seventh Annual IEEE International Digital Object Identifier: 10.1109/ASIC.1994.404555 Publication Year: 1994, Page(s): 296 – 299
6. Vectored Interrupt Controller Usage and Applications-www.altera.com/literature/an/AN595.pdf
7. Digital Design principles and practices, John F. Wakerly, third edition, Prentice Hall Publication, 2000.
8. J. Basker, a Verilog HDL Synthesis 1st Edition, 1998.