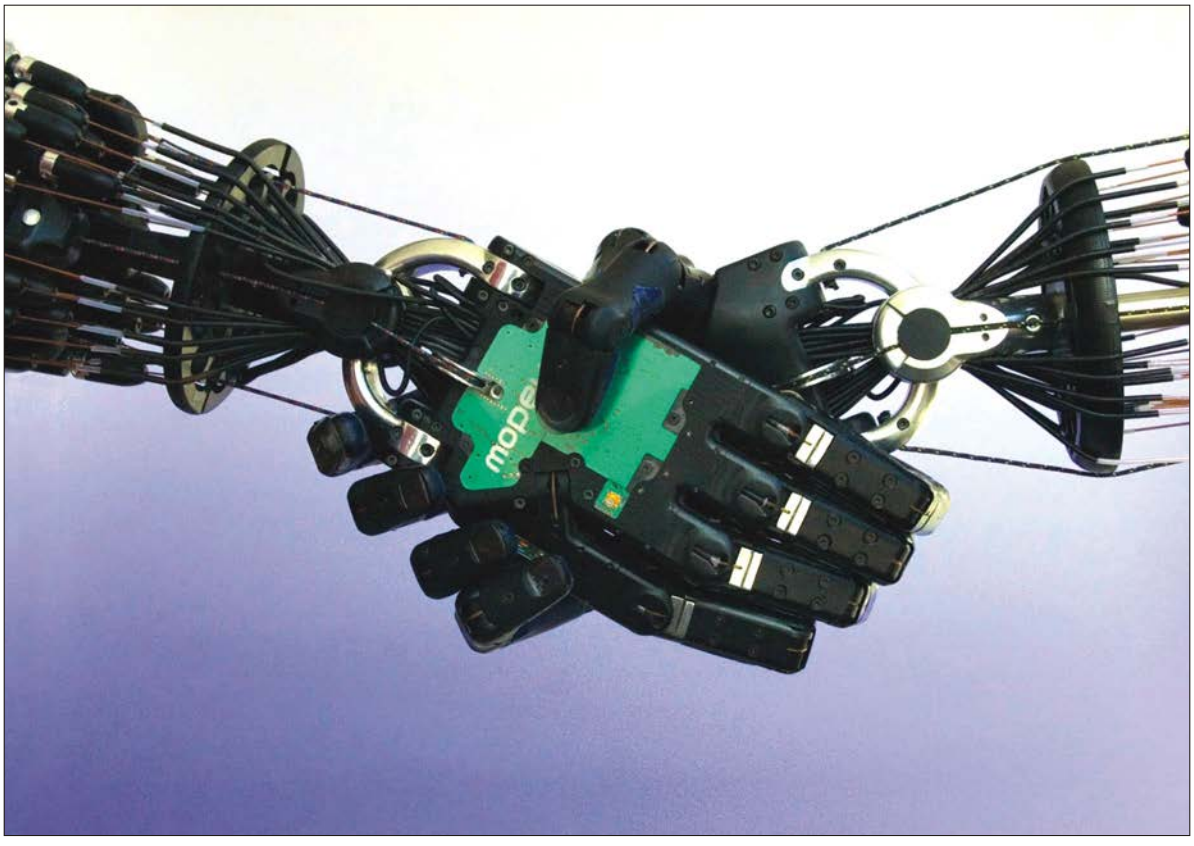

Journal of Artificial Intelligence

ISSN 1994-5450

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Artificial Intelligence 6 (2): 168-174, 2013
ISSN 1994-5450 / DOI: 10.3923/jai.2013.168.174

Design of AXI-PCIE Interface for Industrial Ethernet Applications

R. Arun and R. Muthaiah
School of Computing, SASTRA University, Thanjavur-613402, Tamil Nadu, India

Corresponding Author: R. Arun, School of Computing, SASTRA University, Thanjavur-613402, Tamil Nadu, India
Tel: +91 9894312265

ABSTRACT

The tremendous growth of Ethernet in the field of communication networks has enabled various applications in industries with greater performance and accuracy. Ethernet's consistency and integrated communication capabilities are used in various industrial applications where timing is of a major concern. In spite of these ubiquitous trends of Ethernet, Carrier Sense Multiple Access/Collision Detection (CSMA/CD) and combinational mechanisms are some of the contradictory mechanisms in real time industrial applications. Although it plays a vital role in avoiding collisions between different environments, it uses more of the critical time in processing, which is non-affordable in real time scenarios. Hence, the need for an Industrial Ethernet that employs the traditional features of Ethernet with real time capabilities is very much in demand. In this study, an Industrial Ethernet module-soft Intellectual Property (IP) core design is proposed, it meets the various specifications of Ethernet with the time specific features for real time applications.

Key words: Ethernet, advanced extensible interface, peripheral component interconnect express, carrier sense multiple access

INTRODUCTION

Industrial Ethernet operates on top of legacy Ethernet by complying with IEEE 802.3 standardization without any adoptions. An Industrial Ethernet node can be created easily with a personal computer equipped with a Network Interface Card (NIC). However, the upper layer of the Industrial Ethernet communication stack has to enable static determinism. This can be achieved by avoiding collisions with a slot communication network management executed within the Industrial Ethernet data link layer and still applying CSMA/CD (carrier sense multiple access/collision detection) is also known as IEEE 802.3.

Industrial Ethernet defines two operation modes called Basic Ethernet mode, which refers to an ordinary Ethernet device with CSMA/CD medium access and Industrial Ethernet mode, in which the industrial ethernet Managing Node (MN) controls the network access of the Controlled Nodes (CN) and thus achieves a determinism communication system. On top of the Industrial Ethernet data link layer there is an optional UDP and/or TCP/IP stack, which is often used to integrate a web-or file-server (HTTP or FTP) into the Industrial Ethernet node. However, these types of messages (e.g. TCP/IP) are handled like asynchronous Industrial Ethernet frames, thus, must not affect determinism of the network.

Industrial Ethernet uses cycles as a communication framework that consists of two phases, the isochronous phase is initiated by the Start of Cyclic frame (SoC), which is the only frame that may

be used for the timing of a node. The MN transmits the SoC as a multicast frame received by all nodes and determines the network jitter. The SoC can be used by the nodes to drive their outputs or read their inputs synchronously to the network system. Followed by the SoC, real-time data is exchanged by polling all available and configured nodes with Poll-Request frames (PReq). The PReq is only received as a unicast by the addressed node, transfers data to the CN and forces the CN to response by a Poll-Response (PRes). The PRes frame from the polled node is distributed as a multicast to all other nodes, which actually enables cross traffic between all controlled nodes in the network. The happening of PReq and PRes refers to an exclusive slot communication model for every node. After the last slot the asynchronous phase Start of Asynchronous (SoA) packet is started. This frame allows the MN to allocate the asynchronous phase to a network node that wants to send asynchronous data or TCP/IP messages.

SIGNIFICANCE OF MANAGING NODE

Poll response chaining: The communication method presented before is very efficient if the nodes have huge process data for exchange. However, if we assume an actuator that is controlled by a few bytes, the PReq-or in general the Ethernet frame has a huge overhead 5. Thus, the inefficient frame would allocate valuably time of the Industrial Ethernet cycle. Industrial Ethernet addresses this problem by Poll Response Chaining, which enhances the performance of networks that uses many nodes with small process data amount. The MN commonly initiates the isochronous phase with the SoC packet; however, afterwards the PRes MN is distributed to the network as a multicast. The Controlled Nodes which are configured to support Poll Response Chaining transmit their PRes frames after a certain time, which is defined at start up of the network. After all PRes have been received by the MN it starts to transmit PReq frames for nodes that do not support Poll Response Chaining or are not configured to participate that way. The asynchronous phase is executed as already presented before.

Isochronous phase: At the beginning of an Industrial Ethernet cycle, the MN shall send a SoC frame to all nodes via Ethernet multicast. The send and receive time of this frame shall be the basis for the common timing of all the nodes. Only the SoC frame shall be generated on a periodic basis. The generation of all other frames shall be event controlled (with additional time monitoring per node).The MN shall start the isochronous data exchange after the SoC frame has been sent. A PReq frame shall be sent to all the configured as well as active nodes. The occupied node shall respond by a PRes frame.

Asynchronous phase: During the asynchronous phase of the cycle, access to the Industrial Ethernet network is to be granted to one CN or to the MN for the transfer of a single asynchronous message only. There shall be two types of asynchronous frames available: The Industrial Ethernet ASnd frame shall use the Industrial Ethernet addressing scheme and shall be sent via unicast or broadcast to any other node. A Legacy Ethernet message is to be sent. If no asynchronous message transmission request is pending at the MN scheduling queues and MN shall issue a SoA without assignment of the right to send to any node. No ASnd frame will follow to the SoA frame in this case. The MN shall start the asynchronous phase with the SoA. The SoA shall be used to identify CNs, request status information of a CN, to poll asynchronous-only CNs and to grant the asynchronous transmit right to one CN. The first frame in the asynchronous phase is the SoA frame and it signals all CNs that all isochronous data has been exchanged during the isochronous phase.

Multiplexing: Every Industrial Ethernet cycle, is very often not necessary for every Controlled Node in the network. Thus the Industrial Ethernet standard introduces a multiplexing scheme, which allows optimizing the bandwidth of the isochronous phase. This can be achieved since several CNs within a network share less slots, compared to being polled every cycle.

Cycle time: The Industrial Ethernet cycle time of a network is primarily defined by the application's needs-for instance motion control requires cycles far below 1 msec. The cycle time furthermore implies the processing capability of every individual node within the network. An Industrial Ethernet node (MN and CN) is a hard real-time system with the deadline defined by the Industrial Ethernet cycle. Secondarily the cycle time is defined by the nodes participating in the network and the I/O data size respectively the frame size. Industrial Ethernet cycle is split into the isochronous and asynchronous phase. The first phase time duration is defined by the number and size of PReq and PRes frames. The asynchronous phase length is given by the SoA size (minimum Ethernet frame) and the maximum allowed frame size of asynchronous data transfers.

SOPC DESIGN FLOW

The SOPC design is a powerful tool in Quartus II tool and is used for higher level abstraction of designs to the Quartus II tool. Figure 1 shows the system design using SOPC builder.

DESIGNING OF INDIVIDUAL UNITS IN XILINX

The various individual blocks are designed and integrated. They include the Serial Peripheral Interface (SPI) unit described in the result section, Fig. 2 and 3 show the Media Access Control (MAC) unit and filter unit, respectively.

AXI-PCI EXPRESS (PCIE) IMPLEMENTATION

PCI express introduction: PCI Express is one of the greater growths in the technology aspect over the PCI bus, providing greater bandwidth so that architecture meets the industry requirements of next generation devices. It uses a typical two way lane type communication standard. Each lane consists of a single transmit and a receive pair transferring data up to 2.5 Gbps (Wilen *et al.*, 2003).

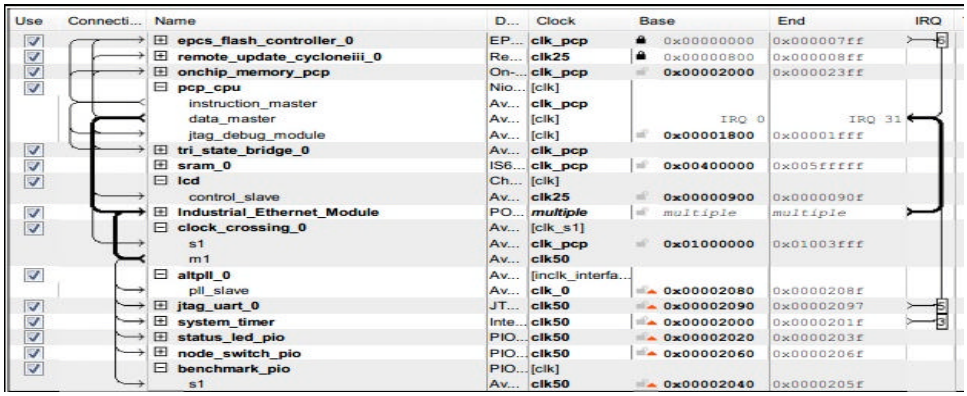

Fig. 1: SOPC system design view

Fig. 2: Synthesis block of MAC unit

Fig. 3: Synthesis block of filter

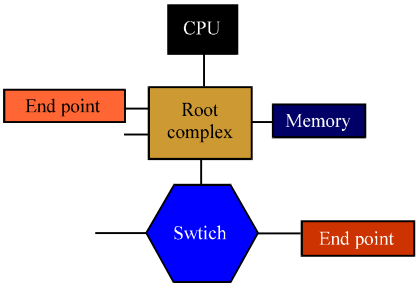

Fig. 4: Typical PCIe system with root complex and endpoint

Root Complex (RC): The main part of the PCI Express interface architecture is the Root Complex that has one or more Host Bridges. Figure 4 shows a sample PCIe system with a Root Complex. Every host bridge is connected to Endpoints or Switches, to the Root Complex (Maestro and Reviriego, 2010)

System CPU(s) and system memory are also connected to the Root Complex, usually through a system bus. The PCIe configuration space in Fig. 5 provides the entire PCIe configuration accessibility.

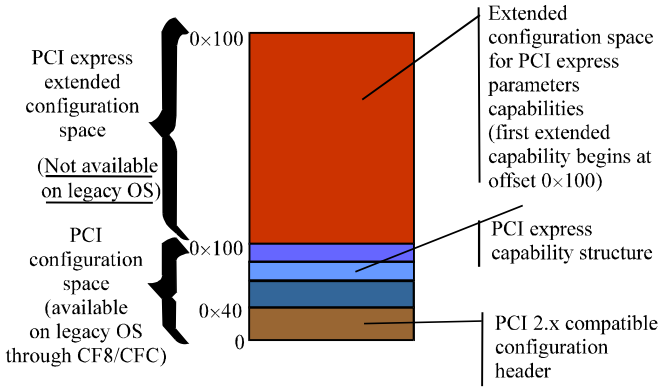

Fig. 5: PCIe configuration space

PCI express slots: These slots are the physical connectors in PCI Express adapters that are used to connect to the PCI Express fabric to the PCI Express fabric.

PCI express-to-PCI bridge: A PCI Express device used to connect PCI devices to the PCI Express fabric.

PCI express endpoint: Any device that has a downstream devices attached to it. Some of the PCI Express Endpoints include the graphics processing unit, network and storage controllers. Thereby every device is initialized and recognized by the CPU through the device ID/vendor ID.

AXI INTERFACE

The Advanced eXtensible Interface (AXI) Endpoint (EP) bus is an interface between the AXI4 bus and PCI Express. The AXI Bridge for PCI Express provides transaction level translation of memory-mapped AXI4 bus commands to PCIe TLP packets and PCIe Memory read and write request TLP packets to memory mapped AXI4 bus commands. The AXI Bridge for PCI Express in Fig. 6 contains two sections: the memory mapped AXI4 to AXI4-Stream Bridge and the AXI4-Stream Enhanced Interface for PCIe. The memory-mapped AXI4 to AXI4-Stream Bridge contains a register block and two functional half bridges, referred to here as the Slave Bridge and Master Bridge.

The Register Block contains registers used in the AXI Bridge for PCI Express for dynamically mapping the MM address range provided using AXIBAR parameters to an address for PCIe range. The Slave Bridge connects to the memory mapped AXI4 bus as an AXI Slave device and the Master Bridge connects to the memory mapped AXI4 Bus as an AXI Master device. The Slave Bridge provides termination of memory mapped AXI4 transactions from an AXI master device (such as a Processor). The Slave Bridge provides a way to translate addresses that are mapped within the AXI Memory Mapped address domain to the domain addresses for PCIe. When a remote AXI master initiates a write transaction to the Slave Bridge, the write address and qualifiers are captured and write data is queued in a First in First out (FIFO). These are then converted into one or more MemWr TLPs which are passed to the integrated block for PCI Express. A second remote AXI master initiated write request write address and qualifiers may then be captured and the associated write data queued pending the completion of the previous write TLP transfer to the integrated block for PCI Express (Gao *et al*., 2011).

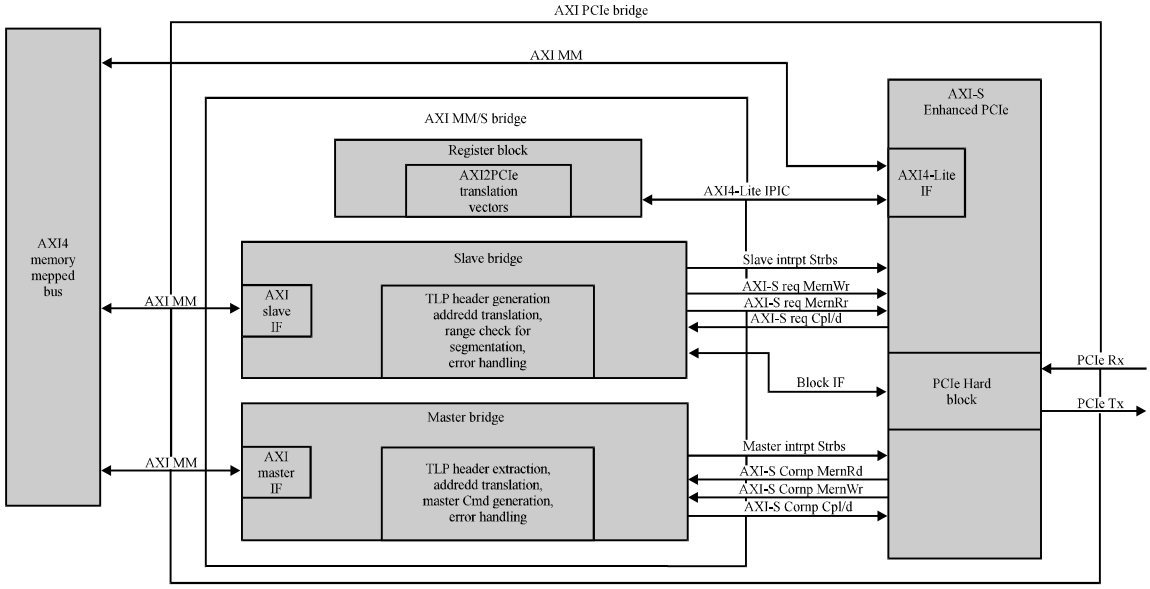

Fig. 6: AXI PCIe bridge

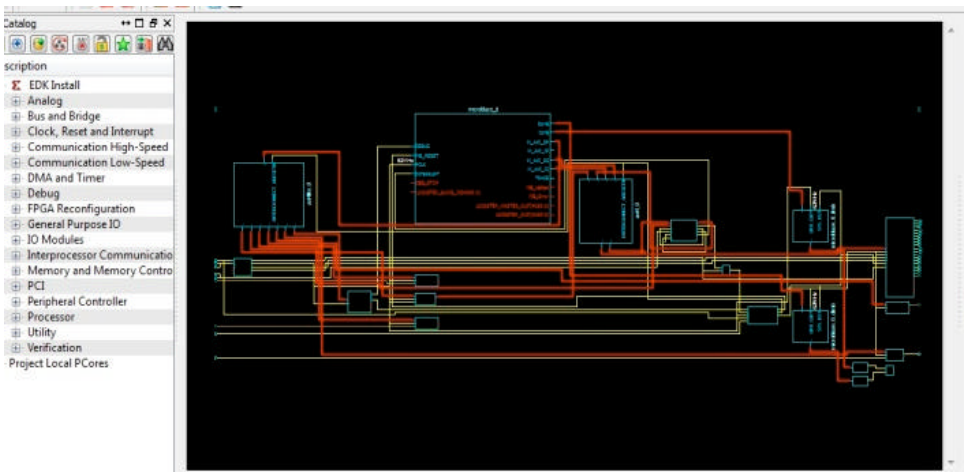

Fig. 7: System design of AXI interface with processor

SYSTEM DESIGN

The overall system design is done using Xilinx Platform Studio (XPS), showed in the Fig. 7 and then integrated with the Industrial Ethernet model in Xilinx Software Development Kit (SDK). The output is analyzed using the logic analyzer and the cycle time is verified. The PCIe is analyzed through the PCI Tree tool, this tool provides us the information regarding the various peripheral cards connected to the system.

The SP605-Xilinx Spartan6 board has an in-built PCIe interface with Microblaze processor. The BAR configuration is also viewed through the PCI tree.

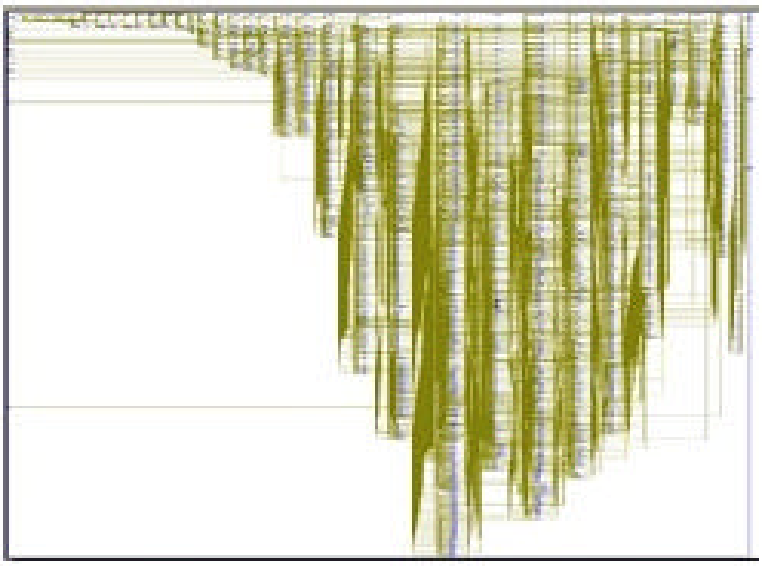

Fig. 8: PCIe tree configuration view

RESULTS

The performance and timing analysis of the designed soft IP core for Industrial Ethernet module are observed and its timing reports show the total cycle time. This provides a useful method for various real time issues faced in industries using Ethernet and proves to give more dedicated performance.

DISCUSSION

The final configured view of the PCI Express is showed in PCI Tree in Fig. 8. The implementation of this soft IP core Industrial Ethernet module is implemented through the FPGA with PCI Express for higher performance and better timing requirements. The scope of the study deals with the development of a basic module in place of the ubiquitous Ethernet (IEEE 802.3), to be more efficient for real time applications in industries and in time to come Industrial Ethernets will provide more deterministic and ubiquitous solutions for real time applications at industries.

REFERENCES

Gao, Z.N., Q.W. Chen and W.L. Hu, 2011. A new experimental platform for networked control systems based on CAN and switched-ethernet. Inform. Technol. J., 10: 219-230.

Maestro, J.A. and P. Reviriego, 2010. Energy efficiency in industrial Ethernet: The Case of Powerlink. IEEE Indus. Electron. Trans., 57: 2896-2903.

Wilen, A., J.P. Schade and R. Thormburg, 2003. Introduction to PCI Express: A Hardware and Software Developer's Guide. Intel Press, USA.