International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.4, July-Aug. 2012 pp-2762-2765 ISSN: 2249-6645

Reconfigurable Communication Architecture of On-Chip Segmented Bus

Sreeja. M. S.[1], Manu. T. S.[2]
[1]PG Scholar, Dept .of Electronics and Communication Engg, TKM Institute of Technology, Kollam, Kerala, India
[2]Asst.Professor, Dept. of Electronics and Communication Engg, TKM Institute of Technology, Kollam, Kerala, India

Abstract: *Modern VLSI technology makes it both feasible and economical to integrate a complex system on a single chip. These modules often require different data transfer speeds and parallel transmission capability. A conventional bus structure might not be adequate for these demands because, typically, only one attached module can transmit at a time, and a large capacitive load caused by attached system modules and long bus wires can make a bus very slow. A segmented bus architecture shows potential for improving both speed and power related features of a bus-based system. Due to segmentation of the bus, parallel transactions can take place, thus increasing the performance of the bus. In order to reduce arbitration and communication delay in the existing segmented bus, new reconfigurable architectures which will completely avoid the complicated higher level arbitration over-head with a small modification in local arbiter is proposed in this paper. The bus architectures are modeled using VHDL.*

Key words: *Reconfigurable, Segmented Bus*

I. INTRODUCTION

Modern deep-submicron silicon technologies permit increasingly complex System on Chip (SoC) designs. The growing diversity of devices results in many possible interfaces. Often, the interconnection complexity of SoC modules limits both the system design process and system performance. Furthermore, these modules often require different data transfer speeds and parallel transmission capability[1]. A conventional bus structure might not be adequate for these demands because, typically, only one attached module can transmit at a time, and a large capacitive load caused by attached system modules and long bus wires can make a bus very slow. Moreover, the increase in both functional complexity and size of modern SoC devices tends to lengthen interconnect wires between system modules. As a result, synchronous system timing based on global clocks will become more difficult, if not impossible .A viable solution to these problems is a segmented bus architecture based on asynchronous communication. Such a structure provides a flexible platform for asynchronous self-timed SoC design, including globally asynchronous, locally synchronous (GALS) designs in which each distinct system module has a self-timed interface but is internally synchronized to a local optimized clock. A self-timed interface method significantly improves system scalability, automatically removing the difficult problems related to global clock distribution. Moreover, partitioning the bus into several concurrently operating segments overcomes the performance bottleneck of a conventional bus, letting modules in a particular segment exchange data independently of modules in other segments. Simple bridges composed of tri-state buffers isolate adjacent bus segments from each other. Whenever an intersegment transfer occurs, these bridges dynamically link several successive segments to establish a connection between modules in different segments. Researchers proposed the concept of segmenting buses primarily for multicomputer architectures. More recent approaches address on-chip implementation of segmented buses. This paper targeted dynamically reconfigurable segmented-bus architecture for high-performance SoC applications. The structure not only enables faster operation than a conventional bus system but also offers lower power consumption per transferred data item. This is possible because segmentation is realized in such a way that the majority of data transfers in the system are intra-segment transactions on relatively short wires with low or moderate capacitive loads.

II. SYSTEM ARCHITECTURE

A segmented bus is a bus which is partitioned into two or more segments. Each segment acts as a normal bus between modules that are connected to it and operates in parallel with other segments. Segments can be linked dynamically to each other in order to establish connection between modules located in different segment [8]. Due to segmentation of the bus, parallel transactions can take place, thus increasing the performance of the bus. A high level block diagram of the segmented bus system is illustrated in Figure 1.

A bus-based system consists of three kinds of components (sub-systems): masters, slaves and arbiters. Generally, each of the devices connected to the bus lines may behave, depending on the situation, as either a master, or a slave. However, for simulation and assessment issues, here consider fixed functionality of these devices, that is, they will be only masters, or slaves. Only one master at a time may transfer data on the bus, thus there is need for arbitration. In a conventional single bus approach, the current master-slave connection occupies the whole length of the bus, even though the communicating devices were physically close to each other. The Segmented Bus approach would allow this kind of connection to occupy only a small portion of the bus - the segment. Arbitration at this level is provided by the Local Arbiter (LA), one for each segment. They decide which master within the segment will get access to the bus. Whenever inter-segment transfers are required, a Central Arbiter (CA) decides which of the requests can be serviced. [2]- [4].Only one inter-segment grant is given by the central arbiter.

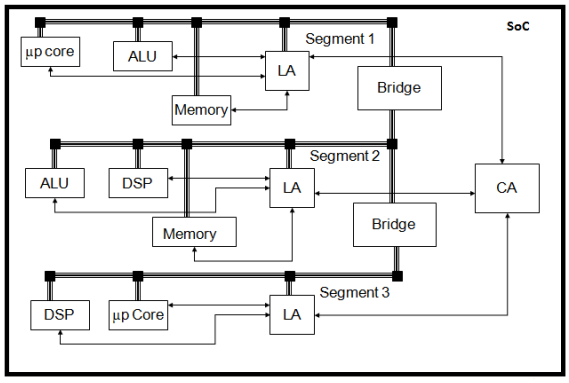

Fig 1: Segmented Bus Structure in SoC.

2.1. Existing Segmented Bus Architecture

The existing segmented bus architecture is illustrated in Fig.2. A segmented bus is partitioned into few segments. Every segment comprises of a group of masters, a group of slaves, a local arbiter, an intersegment bridge module and the physical wires for the bus address, data and control signals [8]. Modules (masters and slaves) are grouped according to their interconnection characteristics and placed on segments. Each segment operates as normal bus for modules, which are placed on it, and operates in parallel with other segments. Segments can be connected to each other in order to establish a connection between modules in different segments [8].

The segments with their components act as stand-alone buses operating in parallel, masters mostly asking services from the group of slaves placed within the same segment limits. Fig. 3 shows the structure of the existing segmented bus [8].

When a master device wants to communicate with a slave device within the same segment, the master device sends a request to its local arbiter. This requires dedicated request and grant wires between each device and the arbiter. The local arbiter grants the bus to the master and then the transaction. When multiple master devices request the bus, the arbiter grants the bus to only one device at a time. Similarly when one segment wants to communicate with other segment central arbiter gives a grant to that segment. But here only one inter-segment communication is possible at a time.

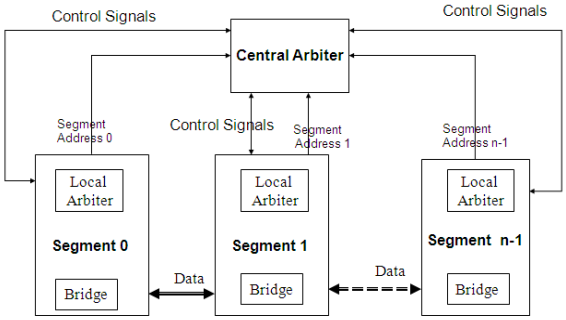

Fig 2: Block diagram of Existing Segmented Bus

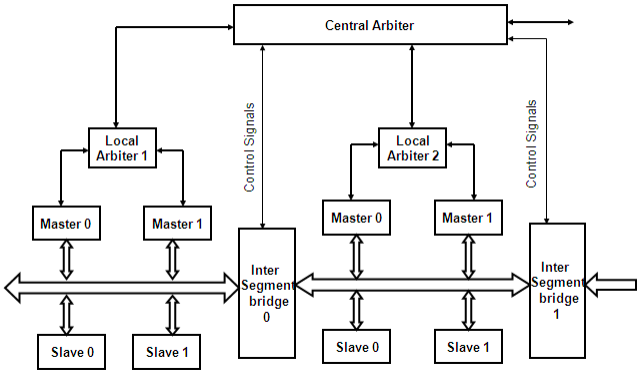

Fig 3: Structure of the Existing Segmented Bus

2.2. Proposed Segmented Bus Architecture

The existing segmented bus architecture has a central arbiter within it. An intersegment data transfer can be made only with the involvement of the central arbiter. Therefore the existing segmented bus architecture has two main disadvantages. Due to the involvement of the central arbiter, the complexity is high and so the intersegment data transfer delay is also high. This is the first disadvantage. The second disadvantage is, when the number of masters/slaves connected in the bus increases, the size and complexity of the central arbiter increases. Due to this increase, the arbitration delay increases very much. In order to overcome these two disadvantages, new reconfigurable segmented bus architecture is proposed in this project. In the proposed reconfigurable segmented bus architecture, re-configurability feature is incorporated in order to eliminate the disadvantages of the existing segmented bus architecture. Re-configurability feature means, adapting the topology of the bus based on the requests from the masters inside the segments. The Block diagram of Proposed Segmented Bus is shown in Figure 4.

The proposed architecture uses dynamic bridge-by-pass technique in order to incorporate dynamic re-configurability in the bus. So this architecture is named as Bridge-by-pass architecture. In this architecture, CA is eliminated. Therefore, the system complexity is reduced and arbitration delay is also reduced. The structure of Bridge-by-pass architecture is shown in Fig.5.The proposed architecture consists of a bus with two segments (Segment1, Segment2).

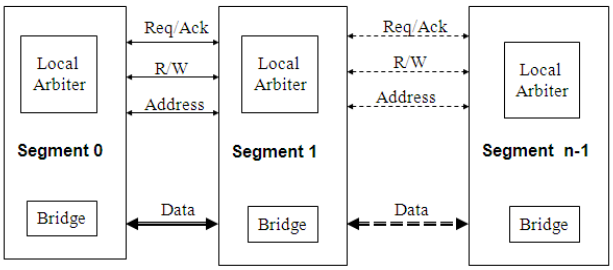

Fig 4: Block diagram of Proposed Segmented Bus

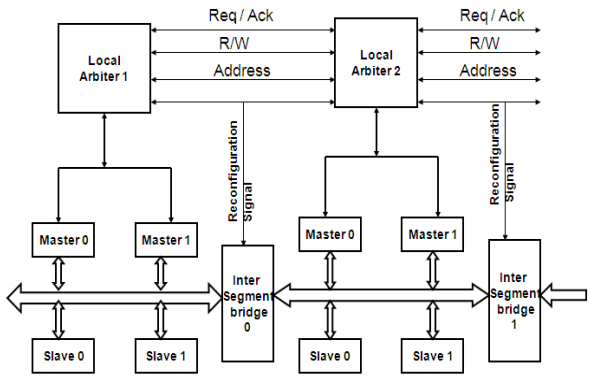

Fig 5: Proposed Segmented Bus Architecture

There are two types of data transfers that can take place in the proposed bus architecture. The first type of data transfer is intra-segment data transfer. The second type of data transfer is intersegment data transfer. In intra-segment data transfer, the data transfer occurs within the segment itself.
For example, if in Segment1, Master1 communicates either with Slave0 or Slave1 it is called intra-segment transfer. In intra-segment communication, the data transfer between a master and a slave takes place through its local arbiter. In intersegment data transfer, data transfer occurs between master of one segment and a slave of another segment. For intersegment data transfer, the existing architecture uses central arbiter for data transfer [8]. The presence of central arbiter increases the delay and system complexity. So in this Bridge-by-pass architecture, central arbiter is removed and instead local arbiter controls the data transfer. In the existing architecture, for intersegment data transfer, the master (source) sends request to its local arbiter. The local arbiter gives request, R/W and address signals to central arbiter. The central arbiter checks for the destination segment, if the destination segment is free then its corresponding local arbiter gives grant signal to the central arbiter. The central arbiter in turn gives grant signal to local arbiter of initiating segment. Then, central arbiter gives control signal to bridge so that the bridge opens and data transfer occurs between the master of source segment and slave of the destination segment.

In Bridge-by-pass architecture, for e.g. if there is an intersegment data transfer between master of Segment1 and slave of Segment2, the master sends request to its Local Arbiter (LA1 in Fig. 5). The local arbiter LA1 sends the request, R/W and address signals to the next local arbiter (LA2). LA2 checks if its bus is free. If its bus is free, LA2 sends grant signal to LA1. Then Reconfiguration unit gets ready signals from LA1 and LA2, processed it and gives reconfiguration signal to Bridge0. The Bridge0 open and data transfer takes place between Segment1 and Segment2. This is the operation of Bridge-by-pass architecture.

Consider that the existing bus architecture consists of 100 segments. Suppose 50 segments requests the central arbiter for inter segment data transfer simultaneously, the central arbiter analyzes the 50 requests and gives grant to the segment which has the highest priority. The inter segment data transfer of that highest priority segment takes place, while remaining segments wait. After the completion of data transfer, the central arbiter gives grant to the next segment which has next higher priority among all other segments. This proceeds till all the requests are processed.

In the Bridge-by-pass architecture, Segment1 has the highest priority for intersegment data transfer and Segment2 has the least priority. Within a segment, Master0 has higher priority than Master1. Suppose 50 segments requests the central arbiter for inter segment data transfer simultaneously, the local arbiter in between two segments controls the inter segment data transfer. Since the central arbiter is removed, the complexity and inter segment delay reduces. Hence when compared to existing architecture, Bridge-by-pass architecture is better in performance in terms of communication delay.

III. SIMULATION RESULTS

The bus architectures are modeled using VHDL. Simulation was done using Modelsim XE III 6.4 Simulator and the output waveforms is obtained as shown in fig.6 and 7. This model is synthesized using Xilinx ISE 9.2i. To evaluate the performance and power consumption of the bus model Xilinx Xpower analyzer is used and the results are obtained. The analysis of the results obtained is discussed in this section.

In Figure 6, Segment 1 considering which consist of two masters (Master1 & Master 2), two slaves (Slave 1 & Slave2 with address 01 and 02 respectively) and a Local Arbiter with fixed priority arbitration. Initially Master 1 and Master 2 are requesting the segment for write operation to Slave 1 and Slave 2 respectively. But Local Arbiter gives only Grant to the Master 1 and it writes the data in Slave 1.After releasing the Request of Master 1, Master 2 gets the Grant and it writes the data to Slave2

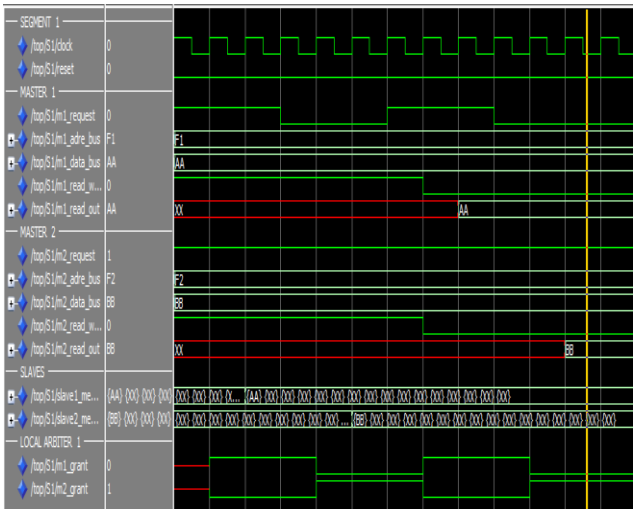

Fig 6: Intra-segment communication

In figure 7 Segment 1 considering which consist of two masters (Master1 & Master 2), two slaves (Slave 1 & Slave2 with address 01 and 02 respectively) and a Local Arbiter with fixed priority arbitration. Also Segment2 is considering with Slave 3 and slave 4 (with address 03 and 04 respectively).Initially Master 1 and Master 2 are requesting

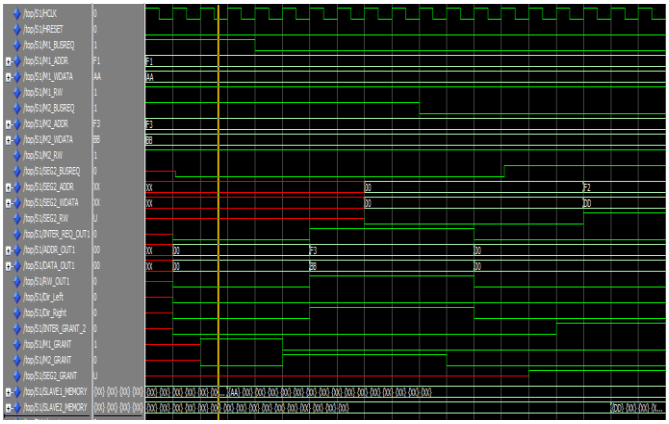

Fig 7.a.: Inter-segment Communication (segment 1)

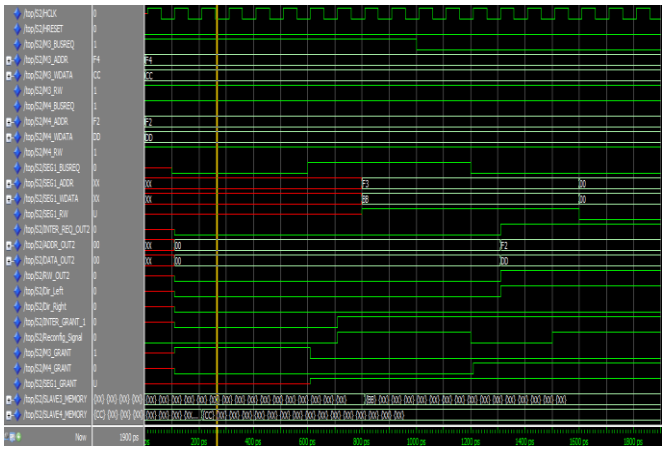

Fig 7.b.: Inter-segment Communication (segment 2)

the segment for write operation from Slave 1 and Slave 4. But Local Arbiter gives only Grant to the Master 1 it is an intra segment operation .After releasing the Request of Master 1, Master 2 gets the Grant for write operation in Slave 4. Here an inter-segment communication is needed because Master is in Segment 1 and Slave is in Segment 2.So INTER SEGMENT REQUEST becomes high and the data of the granted masters appears at the bridge. The bridge is a bi-directional buffer which will be controlled by the reconfiguration signal generated by the reconfiguration unit. After the enabling of the bridge the data is tranfered from segment 1 to segment 2 and which will write on the segment 2 as shown in figure 7.a and figure 7.b.

IV. CONCLUSION

A new dynamically reconfigurable segmented bus architecture is proposed in this paper. In the proposed architecture, speed is increased and communication delay is reduced. There is no power and area overhead. Area has reduced in the proposed architectures. The bandwidth is highly improved. The bus is highly scalable. Due to these reasons, the proposed architecture seems to be highly efficient in terms of design and performance. Hence the proposed architecture can be used in high speed SoC devices.

ACKNOWLEDGEMENT

We would like to thank the Principal, HOD and all the teaching and non-teaching staffs of TKM Institute of Technology for helping to complete the work as mentioned in the paper.

REFERENCES

[1] S.Hema Chitra " A High Performance On-Chip Segmented Bus Architecture Using Dynamic Bridge-By-Pass technique " 5th International Conference On Industrial An Information Systems, ICIIS 2010 Jul29-Aug 01,2010,India, pp 249-254

[2] S.Hema Chitra, P.T.Vanathi "Design and Analysis of Dynamically Configurable Bus Arbiters for SoCs" ICGST- PDCS, Volume 8, Issue 1, December 2008, pp. 45– 52

[3] Krishna Sekar,Kanishka Lahiri, , Anand Raghunathan, and Sujit Dey, Senior Member, IEEE "Dynamically Configurable Bus Topologies for High-Performance On-Chip Communication" IEEE transactions on very large scale integration (VLSI) systems, vol. 16, no. 10, October 2008. pp. 1413-1426

[4] Krishna Sekar,Kanishka Lahiri, , Anand Raghunathan, and Sujit Dey, Senior Member, IEEE "Dynamically Configurable Bus Topologies for High-Performance On-Chip Communication" IEEE transactions on very large scale integration (VLSI)systems, vol. 16, no. 10, October 2008. pp. 1413-1426

[5] Milica Mitic and Mile Stojeev "An Overview of On – Chip Buses" Facta Universitatis (NIS) SER.:ELEC.ENERG. vol.19,no.3, December2006, pp405-428.

ABOUT AUTHORS

Sreeja. M. S.[1], has M.Tech in VLSI and Embedded systems and B.Tech in Electronics and Communication. Currently working as Assistant professor of Electronics and Communication Engineering Department in Younus College of Engineering and Technology, affiliated engineering college under University of Kerala.

Mr.Manu.T.S[2] has M.Tech in Embedded systems and B.Tech in Electronics and Communication and he is working as an Assistant professor of Electronics and Communication Deparment in TKM Institute of Technology,Kollam, an affiliated engineering college under Cochin University of Science and Technology for last two years, Asst.Prof. Manu T.S has guided over 25 students for their accademic project ,seminars and mini projects throughout his career and he undergone many training programs in VLSI and Embedded systems .He presented and Published Papers related to FPGA designs in National and International Conferences and Journals .