

An Analysis of On-Chip-Network Communication Properties on Shared and Multiple Channel Architecture

Sanghun Lee and Chanho Lee

School of Electronic Engineering, Soongsil University, Seoul, Korea

511 Sangdo-dong, Dongjak-Gu, Seoul, Korea

E-mail: {blueye76, chlee}@ssu.ac.kr

Abstract: The AMBA AHB has been popular bus architecture so far due to its simple architecture. Complex system-on-chips (SoCs) with multiple masters usually employ bus architecture with multiple layers or multiple channels because they require large bandwidth. We construct four systems with different bus configuration to analyze and compare the performance of shared buses and on-chip-network with multiple channels. The systems include multiple masters in parallel processing. The shared bus architecture with multiple layers provides less bandwidth than expected and the on-chip-networks with independent multiple channels show better performance. The system with multiple channels shows 2-3 times better performance than that with the AHB. The results are verified by simulation and implementation using an FPGA.

1. Introduction

As fabrication technology and EDA tools are developed, it gets closer to implementing system-on-chips (SoCs) for parallel computation of large amount of data, such as multimedia data which require a lot of workload. The data communication in a system with multiple processors is usually limited by the bottleneck of communication channels, not by the performance of processors. The system performance degrades when concurrent access requests are made to a shared communication resource. The problem comes from the shared bus architecture of the most existing SoC on-chip-buses.

The AMBA AHB (Advanced High-performance Bus) which has the largest market-share is relatively simple in architecture compared with other buses. The AHB is applied easily to small scale SoCs. Therefore, the AHB has been the representative of the SoC market though the bus efficiency shows the limitation in performance with the growing SoC scale. Consequently, the AHB is not suitable for the SoCs with multi-masters where frequent access contention to the bus occurs. A development environment and flow may be complicated since a system developer must consider many conditions to improve system performance. Therefore, we need a new SoC on-chip-bus architecture which overcomes the problems of the AHB.

When function blocks or intellectual properties (IPs) are added to a system, available bandwidth for each IP may be decreased due to fixed bandwidth of most conventional bus architecture. The conventional SoC buses such as AMBA AHB [1], Wishbone [2], and CoreConnect [3] have limitations on the number of masters on the same bus to obtain a desired bandwidth and to avoid complex arbitration. In order to solve the problem, a system needs to have multiple bus segments with bridges and complex

topology. Designers have to thoroughly understand the bus protocol and the amount of traffic on the bus. A hierarchical architecture requires a propagation latency passing through bridges, and a centralized bridge structure causes similar problems as with the shared bus. Although, the Interconnect Matrix is developed to solve these problems, we can not avoid increase in area and re-design of the existing AHB-based system [4]. The conventional bus architecture shares similar problems to those of the AHB bus because of the shared bus architecture. We need a new bus system to solve the problem of the conventional buses in order to satisfy the requirement of the data communication in the growing SoCs.

Until now, a number of bus protocol and bus architecture have been released. The AMBA including AHB and AXI (Advanced eXtensible Interface) is still popular bus architecture because of easy connectivity with ARM processors, open license, and simple protocol. The AMBA AHB is widely used in the academic and industrial worlds. However, the share bus systems such as the AMBA AHB are not adequate for parallel process systems with multiple masters even with multiple layers. On-chip-network systems with independent multiple channels can solve the problem.

In this paper, we compare communication properties of four systems with different bus architecture in cycle level to analyze the properties and the performance of the shared bus architecture and the multiple channel architecture. Four systems with AHB 1-layer, AHB 2-layer, 1-XR SNA, and 2-XR SNA [5] are designed and compared. The results are verified by simulation and implementation on an FPGA.

2. Bus and On-Chip-Network Architecture

The AHB employs shared bus architecture which is simple. Nonetheless, it has two critical problems when a system has lots of IPs. One is the limitation in bandwidth for multiple masters. The other is the limited operating frequency due to the heavy loading as the number of slave increases. Communication bottleneck problem occurs mainly in bridge-used system. Bus bridges for multiple layers on the shared bus, which were introduced to increase the bandwidth, does not solve the problem because the bus system becomes a single layer again once a bridge is activated, and the location of an IP affects the performance a lot.

Figure 1 shows an on-chip-network system using the SNA [5]. The SNA consists of crossbar routers (XR) which provide multiple communication channels, a crossbar router bridge (XRB) which support connection between crossbar routers, and switch wrappers/bridges (SW/SB) which make

connection between IPs or subsystems and a crossbar router. The SNA has minimized global wires since it uses the XSNP in which most sideband signals form into a packet and the interconnection wires exist between crossbar routers and IPs in short distance.

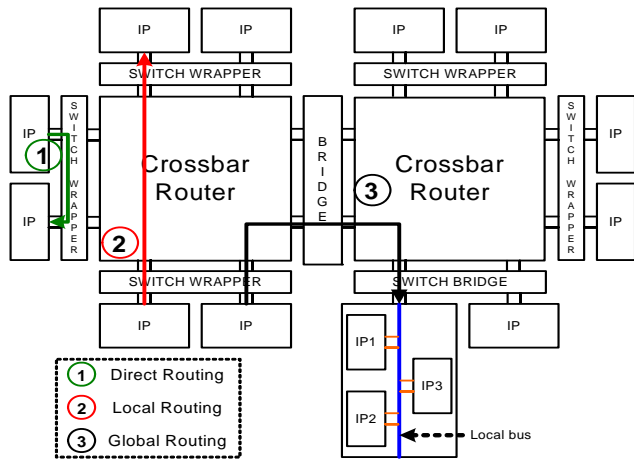
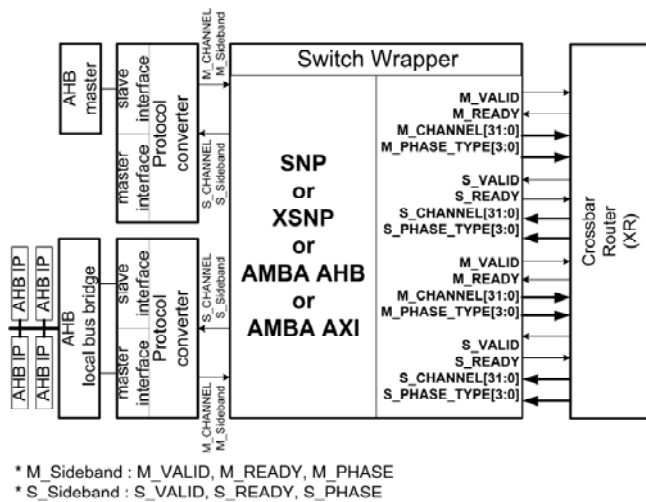


Figure 1. SNA Architecture with two crossbar routers

The SNA does not distinguish master and slave interfaces since the interface is symmetric for the initiator and the destination. The role of the initiator (master) and the destination (slave) is determined by the function of IPs. The interface of an IP with the capability of initiator, destination, or both initiator and destination is the same in the SNA system. It gives the flexibility of system design.

The SNA employs the XSNP as an interface protocol to maintain compatibility with the AMBA AHB protocol. An AHB-to-XSNP converter is added when an AHB-compliant IP is attached. Figure 2 shows the block diagram of the switch wrapper and the protocol converter. The AHB-to-XSNP converter has one master and one slave interface. It can accommodate a master, a slave, a master and a slave, or an AHB-to-AHB bridge. The crossbar router provides up to 8 master and 8 slave interfaces, and more than 16 AHB IPs can be connected using local AHB-to-AHB bridges.



* M_Sideband : M_VALID, M_READY, M_PHASE
 * S_Sideband : S_VALID, S_READY, S_PHASE

Figure 2. Block diagram of protocol converters and XSNP switch wrapper.

A crossbar router can provide 4 channels simultaneously, and a 2-XR system can hold 6-8 channels. The channels in the crossbar router are independent and the routing is guaranteed unless a source or a destination coincides in the path.

3. Simulation and Verification

We construct a system with FIR (finite impulse response) filters and a bus system to verify the function and the performance of the SNA. Figure 3 shows the structure of an FIR filter. FIR filter consists of an FIR filter core, a control register, a source address register, a destination address register, a coefficient value register, and AHB master and slave interfaces. The control register is used to control interrupt operation, read/write transfers and flow control. The source address register receives addresses of source image supplied from a CPU or a neighboring FIR filter. The destination address register receives addresses of destination memory for the filtered image. The coefficient value register receives coefficient values for the FIR filter from a CPU. The FIR filter includes both the AHB slave interface and the AHB master interface because it should issue transactions to other FIR filters and receive transactions from another FIR filter or a CPU.

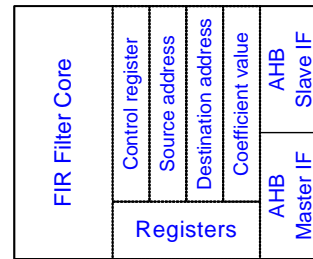


Figure 3. Structure of FIR filter

Figure 4 shows operation sequence of FIR filter systems. A temporary storage (TS) is used to save intermediate operation results. FIR filter A reads image data from TS A (1), and the FIR filter A saves the processed data to TS B (2). The FIR filter A notifies the completion of write transfer to FIR filter B through the slave interface (3). FIR filter B read 16 image data from TS B (4), and announces the completion of reading TS B to FIR filter A (5). The FIR filter B saves the image data after filtering.

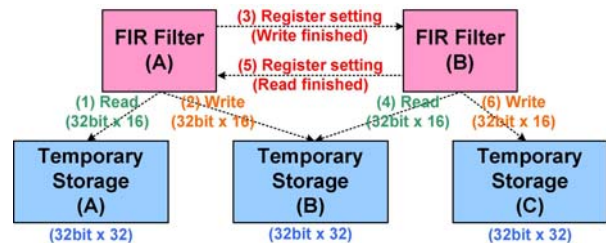
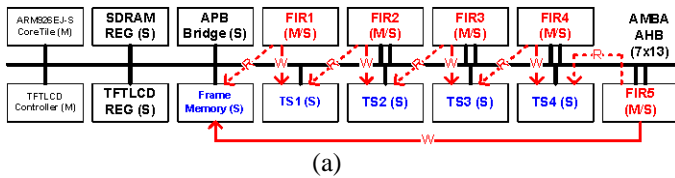


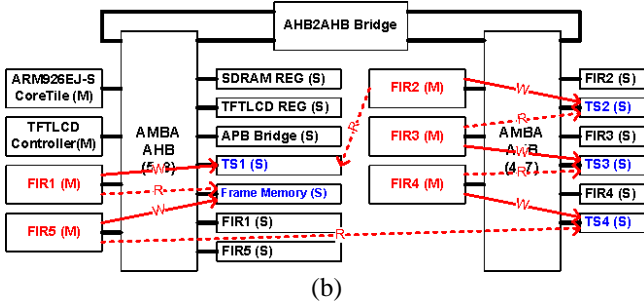
Figure 4. Operation sequence of FIR filter system

Figure 5 shows FIR filter systems which are composed of an embedded processor, five FIR filters, four temporary storages of 32 x 32 bits, a frame memory, and peripherals. Source images of QVGA size (320 x 240 pixels) in the

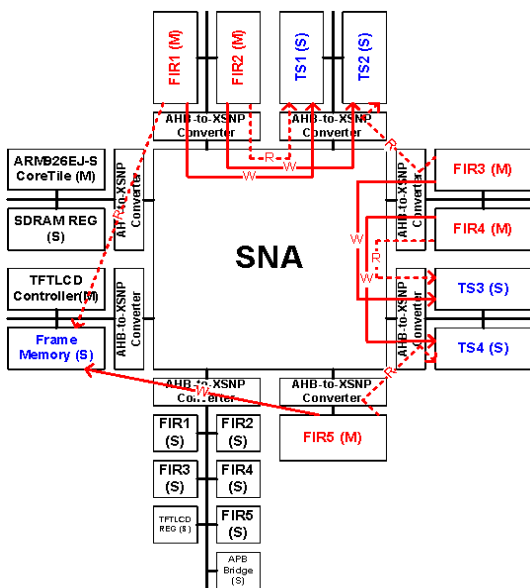
frame memory are filtered by 5 FIR filters in sequence, and the intermediate data are stored in the corresponding temporary storage. The final filtered images are stored in the frame memory. The filters can be operated in parallel and the performance of the parallel processing is determined by the bandwidth of the corresponding bus system. The performance is measured for four configurations with different bus systems, which are a single layer AHB, a double layer AHB, a single XR system, and a double XR system. The four systems have exactly the same configuration except the bus system, and perform the same operation. Therefore, the performance of a system represents that of the corresponding bus system. The SNA-based FIR filter systems include the same IPs as the AHB-based systems. The SNA replaces the AHB system as an on-chip-network. Additionally, the SNA-based system needs AHB-to-XSNP converters to convert protocols from AHB to XSNP, or XSNP to AHB. Actually, the protocol converters are not SNA component, since they are used for AHB IPs only.



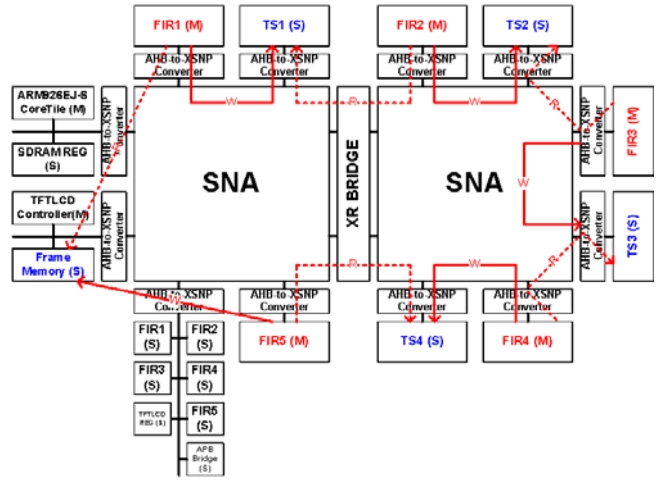
(a)



(b)



(c) FIR filter system with a single XR



(d)

Figure 5. Configurations of FIR filter systems (a) FIR filter system with a single layer AHB (b) FIR filter system with a double layer AHB (c) FIR filter system with a single XR (d) FIR filter system with double XRs

Figure 6 shows the total number of execution cycles of the FIR filter systems. The latency for read and write transactions of temporary storages can be configured from 0 to 3 to represent the operation of various slaves. Figure 6(a) shows the operation results when the latency for read transaction (R) is 3 and that for write transaction (W) is 3. In order to investigate the influence of the burst transfer types, we apply five burst transfer types, that is, single, 4-burst, 8-burst, 16-burst, and mixed burst which is the combination of 16-bursts for filters A and B, 8-bursts for filter C, and 4-bursts for filters D and E. The numbers of execution cycles for the AHB double layer do not decrease a lot because of the propagation latency and frequent SPLIT transfers to avoid dead-lock situations in the AHB-to-AHB bridge. The double XR system has twice the performance of the single XR system for the SNA systems. It implies that the double XR system provides twice the bandwidth of the single XR system and the crossbar router bridge works fine. Generally, systems based on the SNA shows much higher performance by approximately 2-4 times than the AHB systems. The number of execution cycle of the systems based on the SNA are larger for the single transfers than for the burst transfers because a single transfer according to the XSNP requires at least 3 cycles. The XSNP is not efficient for single transfers. Nevertheless, the SNA shows much higher performance even for single transfers in parallel processing systems.

Figure 6(b) and (c) shows the effects of R and W for the single transfers and 8-burst transfers. The total number of execution cycles decrease as the latency decreases as expected. The difference of the performance becomes smaller for single transfers than for the burst transfers because the SNA is not efficient for the single transfers. The SNA which represents on-chip-networks with multiple channels shows much better performance with larger R and W values. Large R and W values indicate that the required bandwidth for the data communication is high. Figure 6 shows that the double layer structure of the shared bus is not efficient for the complex system with multiple masters

which requires high data bandwidth, and the on-chip-network with multiple independent channels are necessary.

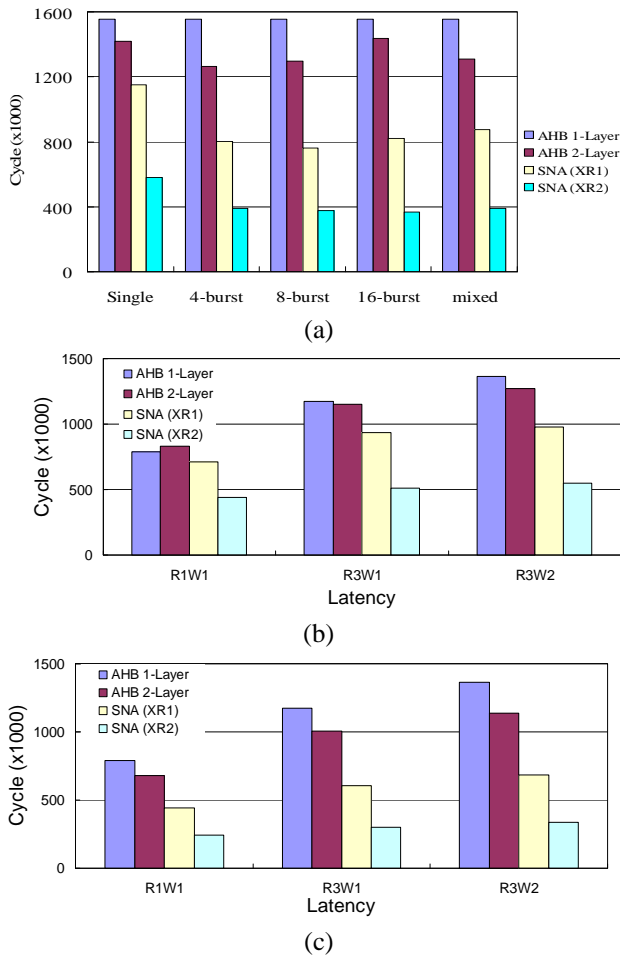


Figure 6. Comparison of execution cycles with AHB and SNA. (a) Performance according to burst types and bus architecture (R=3, W=3) (b) Performance for single transfers according to R and W (c) Performance for 8-burst transfers according to R and W

Table 1 shows synthesized areas of FIR filter systems using Synopsys Design Compiler and a 0.25um CMOS technology at 100MHz. Each area includes bus system. Although, the area occupied by logic gates is larger for the SNA, the area occupied by interconnection wires are getting larger for the AHB system after placement and routing job, and the area of a bus system is smaller compared with those of processing units for systems with multiple processors. Table 3 shows that the area occupied by the SNA system may not be concerned in real world systems.

Table 1. Synthesized area of FIR Filter systems (0.25um CMOS @ 100MHz)

Area [gate counts]	AHB (1-layer)	AHB (2-layers)	SNA (1-XR)	SNA (2-XRs)
Total	135,125	147,136	172,388	196,248
On-chip-network	2,221	7,051	12,271	28,941

We verify the operation and the performance of the FIR filter systems by prototyping on an FPGA board. Figure 7 shows the original and filtered images displayed on the TFTLCD panel of the prototyping board. The frame memory has 8 images. The FIR systems read and filter the images one by one, and display 4 images at a time. The filter systems repeat the process. The performance of a system can be measured by how fast the display is refreshed. The SNA system shows the higher performance than the AHB system as expected in the simulation.



Figure 7. Original and filtered images on a display device of a prototyping board

4. Conclusion

In this paper, we compared performance of shared architecture and multiple channel architecture in parallel application. And we compared on-chip-network architecture and communication properties. We simulated on multi-FIR filters system which has 4 parallel processable FIR Filters. We get performance efficiency of communication by up to 50% than AHB architecture. The proposed SNA architecture is suitable to system which has much parallel communication and parallel computation. Also, burst transfer efficiency according to burst type is significant than AHB system. Area overhead of the on-chip-network is not significant as compared with area of the entire system.

References

- [1] ARM(1999) AMBA Specification, Revision 2.0
- [2] W. Peterson(2002) WISHBONE SoC Architecture Specification, Revision B.3, Silicore Corporation.
- [3] IBM(1999), CoreConnect Bus Architecture.
- [4] ARM(1999), ARM DDI 0170A.
- [5] S. Lee and C. Lee, "A High Performance SoC On-chip-bus with Multiple Channels and Routing Processes," IFIP VLSI-SoC2006, pp. 86-91, Nice, France, October 2006.
- [6] Lee, S., Lee, C., Kim, E.-S., Lee, H.-J. : A High Performance AMBA AHB-compatible SoC Network Protocol for SoC Network Architecture, Vol. 2. ITC-CSCC Conference, Chiang Mai, Thailand (2006) 49-52