H.264 HDTV DECODER USING APPLICATION-SPECIFIC NETWORKS-ON-CHIP

Jiang Xu, Wayne Wolf
EE, Princeton University
{jiangxu, wolf}@princeton.edu

Joerg Henkel
CS, University of Karlsruhe
henkel@informatik.uni-karlsruhe.de

Srimat Chakradhar
NEC Laboratories America, Inc.
chak@nec-labs.com

ABSTRACT

This paper studied an H.264 HDTV decoder on two multiprocessor system-on-chip architectures. Two types of networks-on-chip, the RAW network and the application-specific networks-on-chip, were used. Regular-topology networks-on-chip (mesh, torus, and fat tree) have been proposed. However, we showed in this paper that the application-specific networks-on-chip provided substantial improvements in power, performance, and cost compared to regular-topology networks-on-chip. We measured the power, performance, area, total switch and link capacity, and switch and link utilization based on floorplans and circuit designs. Measurement results showed that the application-specific networks-on-chip was both faster in absolute terms and more efficient. The application-specific networks-on-chip used 39% less power, 59% less silicon area, 74% less metal area, 63% less switch capacity, and 69% less link capacity to achieve 2X performance compared to the RAW network.

1. INTRODUCTION

ITU-T recommendation H.264 has better coding efficiency and is network-friendly. An H.264 HDTV decoder chip requires low cost, low power, and high performance. System-on-chip design for H.264 systems at any level of resolution is still an open problem due to the advanced features that an H.264 decoder must support.

Several previous works discussed some aspects of the H.264 coder and decoder architecture design. Tol and others proposed to partition data over processors for an H.264 decoder [4]. Kang and others proposed a scalable H.264 decoder architecture [5]. Huang and others implemented an H.264 coder [6]. Qiang and Jin implement an H.264 coder and decoder on a single RISC processor [7]. On-chip communication network is a critical part in the H.264 HDTV decoder SoC design. However, previous works did not cover it. In this paper, we concentrated on the impacts of using different types of on-chip communication networks for the same computation nodes in the H.264 HDTV decoder SoC.

On-chip communication network gradually grows from buses and ad-hoc interconnections into sophisticated networks-on-chip (NoC) [14] [15] [16]. Dally and Towles proposed a 2-dimensional folded torus NoC [10]. Kumar and Jantsch also introduced a 2-dimensional mesh NoC, called CLICHÉ [11]. Agarwal and others presented RAW, which based on a 2-dimensional mesh NoC [12]. Forsell proposed Eclipse architecture, which is also based on a 2-dimensional mesh NoC [13]. Adriahantenaina and others presented SPIN, which is a fat-tree NoC [17]. All those NoCs use regular-topology networks.

In this paper, we propose application-specific networks-on-chip. Application-specific NoC tailors the topology and protocol (including packet format, routing algorithm, deadlock avoidance method, and signaling scheme) for each SoC. We measured the power, performance, area, total switch and link capacity, and switch and link utilization based on floorplans and circuit designs. Compared to the regular-topology NoC, measurement results showed that the application-specific NoC was both faster in absolute terms and more efficient.

In the following section, we describe the H.264 HDTV decoder and its mapping. We first present the SoC architecture based on RAW in Section 3. Section 4 details the design of application-specific NoC. Comparison results and analysis are given in Section 5. Section 6 concludes our study.

2. H.264 HDTV DECODER

ITU-T specifies an H.264 decoder (Figure 1) in the recommendation [1] [2] [3]. In the entropy decoding stage, the input video stream is interpreted, and various syntax elements are demultiplexed. Syntax elements of the video stream related with residual macroblocks are processed by the inverse transform stage. Syntax elements related with intra prediction macroblocks and motion compensated prediction macroblocks are processed by the intra-frame prediction and motion compensation stage with reference to previous decoded frames or fields. The deblocking filter stage reduces artifacts introduced by the coding process at block boundaries.

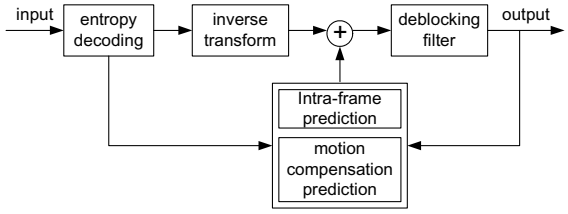

Figure 1. Block diagram of an H.264 decoder

Our study was based on an H.264 reference model JM [23]. We used the main profile and a progressive HDTV sequence

0-7803-9332-5/05/$20.00 ©2005 IEEE

with a resolution of 1920X1088 and 523 frames. The reference decoder model was partitioned and mapped to a computation architecture (Figure 2) based on the decoder diagram and program profiling. Input and output agents help to organize the input video stream and decoded video frames. The entropy decoding stage is implemented by two processors, P0 and P1. The inverse transform stage and the intra-frame prediction and motion compensation prediction stage are implemented by the processor P2. The deblocking filter stage is implemented by two processors, P3 and P4. We target a 130nm aluminum technology to implement the H.264 HDTV decoder SoC. We use Plasma core [21] for each processor.

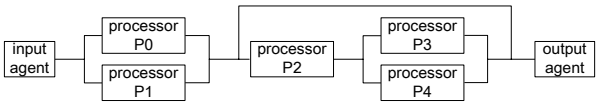

Figure 2. Computation architecture for the H.264 decoder

3. RAW ARCHITECTURE

RAW from MIT is a popular representative of multiprocessor SoC using regular-topology NoC. The RAW network has a 2-dimentional mesh topology [12]. The memories, input, and output are located at the outskirt (Figure 3). Since there are only 5 processor nodes, four tiles are left blank in a 3X3 floorplan. Both RAW and the application-specific NoC have two memories. Memory M0 serves the input agent and the processor P0 and P1. Memory M1 serves the output agent and the processor P2, P3, and P4.

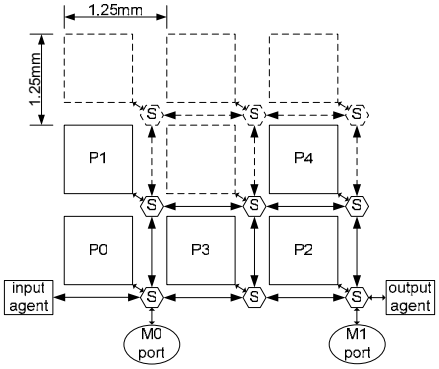

Figure 3. RAW architecture

RAW uses dimension-ordered routing. A packet has a 32-bit header, followed by data. Abs X&Y field holds the destination address; orig X&Y field holds the source address; usr field holds control information; length field holds the size of following data; F field holds the final route when a packet arrives its destination. Buffer metering is used as deadlock avoidance method. We assume the fastest timing scheme, that is, a packet header can be processed in one clock cycle, and the packet can be send out a switch port in the next cycle, if the port is available. If multiple packets compete for the same port, round-robin scheduling will be used.

Positions of the processors, the input and output agents, and the memories affect the system performance. We found the optimized positions as showed in the figure. In RAW, a 1.25mmX1.25mm tile holds a processor and a switch. Switches connect each others by 1.25mm 32-bit links. A processor is connected to a nearby switch by a 0.12mm 32-bit link. The circuits of the switches and links are designed in the same way as the application-specific NoC, which we describe in the following section.

4. APPLICATION-SPECIFIC NETWORKS-ON-CHIP

Application-specific NoC includes two types of networks: local network and global network. Global network connects all the local networks. We design the application-specific NoC based on following guidelines:

a) Grouping intellectual property (IP) nodes into local network in such a way that it increases communication locality and reduces temporal conflicts in communications
b) Application data should be in the same local network with the IP nodes who consume them
c) All the IP nodes in a local network are synchronized, and global network could be asynchronous

The application-specific NoC is designed based on communications among IP nodes [8]. A network design and simulation environment, OPNET, is used to model the topologies and protocols and analyze performance [22]. To compare the powers and costs of NoCs, floorplans are estimated, and NoC circuits are designed using SPICE [19] and Design Compiler [20]. In following, we describe the application-specific NoC in details.

4.1. SoC Architecture using application-specific NoC

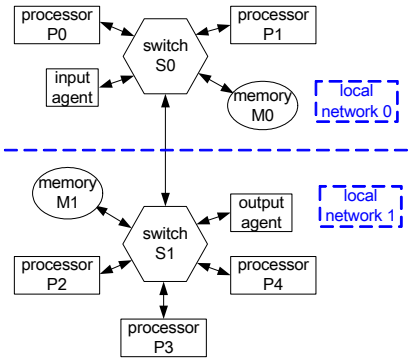

Figure 4. SoC architecture using application-specific NoC

The H.264 HDTV decoder SoC uses the application-specific NoC with two local networks (Figure 4). Local network 0 has a 5-port switch. Local network 1 has a 6-port switch. The architecture uses the same input and output agents, processors and memories as the RAW architecture. The mapping is the same as RAW. Memory M0 and M1 also serve the same groups as RAW.

4.2. Application-specific NoC protocol

In each packet, there is a 28-bit packet header, and if required, 32-bit data will follow the header. The 3-bit source field and the 3-bit dest field are used to address the 7 nodes (excluding memories and switches). The 19-bit address field

and the 3-bit dest field show the exact word address in the memory. 3-bit control field shows the operation a packet carries on. The customized packet is smaller than that of RAW, and it reduces the number of bits transmitted in each network access.

In the application-specific NoC, static routing is used. Based on the source and destination, a packet is routed against a predefined routing table. We use the static routing and fixed priorities to avoid deadlock. If multiple packets compete for the same port, packets are sent based on the priority of the source. The priority from high to low is memory M0, memory M1, input agent, processor P0, P1, P2, P3, P4, and output agent.

4.3. SoC floorplan and NoC circuits

To accurately calculate the power and area, we designed SoC floorplan (Figure 5) and NoC circuits. As in RAW, we assume the each processor holds a 1.25mm by 1.25mm area. In the SoC using the application-specific NoC, the processor P0 and P1 need 0.12mm links to connect with switch S0; the output agent needs a 1.25mm link to connect with switch S1; and switch S0 needs a 2.5mm link to connect with switch S1.

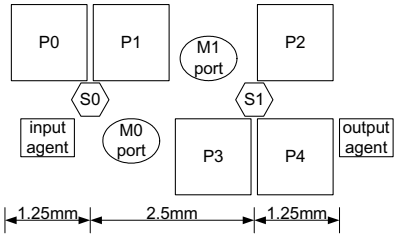

Figure 5. Floorplan of SoC using application-specific NoC

The links and the switch (except the control unit) are designed and simulated using SPICE [19], and the switch control unit is described in Verilog and synthesized in Design Compiler [20]. We model the link interconnection as a fine-grained lumped RLC network. Coupling capacitance and mutual inductances up to the 3rd neighboring wires are considered. We use the typical wire dimensions from the Berkeley Predictive Technology Model [18]. 0.12mm link interconnections use intermediate metal layer, and other link interconnections use global metal layer. The crossbar uses the intermediate metal layer. The input driver of a link interconnection is a chain of sized inverters. The circuits use 130nm aluminum technology and1.5V power supply.

5. RESULTS AND ANALYSIS

We measure the performance, power, silicon area, metal area, total switch capacity, total link capacity, switch utilization, and link utilization for each NoC. The measurements are based on cycle-accurate simulations on OPNET using statistical communication traces and circuit designs.

5.1. Definitions and formula

The performance is measured by the average number of clock cycles to process one frame. Silicon area and metal area of the two networks are gotten from circuit design. The power is measured by the average energy consumed to process one frame, and it is calculated using formula $P = \sum A_i \times E_i$. A_i is the average number of a type of network access. E_i is the energy consumed by the type of network access. The average power P is a sum of energies consumed by all types of network accesses excluding the memories, the processors, and the input and output agents.

The switch utilization is defined by formula $U_s = \frac{\sum B_i}{\sum C_i}$, where B_i is the number of bits switched by switch i in one second, C_i is the capacity of switch i in one second. The switch utilization U_s is the ratio of total number of bits switched by all the switches to total capacity of all the switches. Similarly, the link utilization is defined by formula $U_l = \frac{\sum L_i}{\sum S_i}$, where L_i is the number of bits transferred by link i in one second, S_i is the throughput of link i. The link utilization U_l is the ratio of total number of bits transferred by all the links to total throughput of all the links.

5.2. Results and analysis

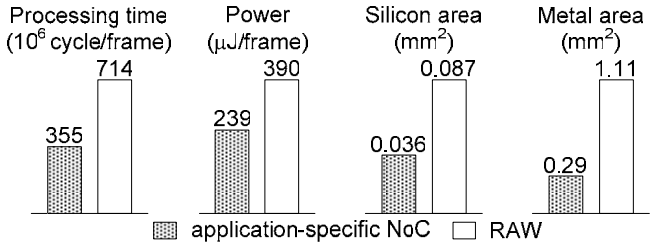

Figure 6. Performance, power, and area

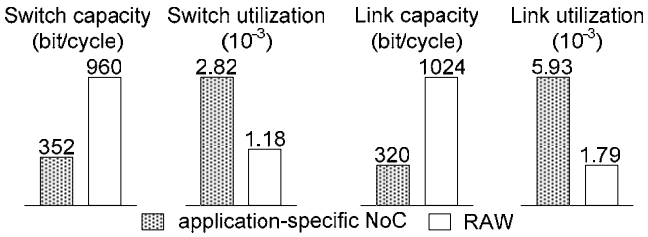

Figure 7. Capacity and utilization of switch and link

Figure 6 shows the performance (in term of the processing time), the power, and the silicon area and metal area. Compared to the regular-topology NoC – the RAW network, the application-specific NoC uses shorter processing time, lower power, and smaller silicon and metal areas. The H.264 HDTV decoder SoC using the application-specific NoC is twice faster than that using the RAW network. The customized network contributes most of the performance improvement. The application-specific NoC reduces the number of hops that a network access needs, and it has faster network access time and reduces the processing time. For example, in RAW, the processor P4 has to pass two switches and needs 3 hops to communicate with memory M1; in the

application-specific NoC, P4 only needs to pass one switch and 2 hops to communicate with M1.

The application-specific NoC uses 39% less power than the RAW network. The customized network reduces the number of switches and hops, which a network access needs, as well as the power consumed by the network access. Smaller packet size also helps reduce the power in the application-specific NoC. RAW uses a 32-bit packet header, while the application-specific NoC uses a 28-bit packet header. The application-specific NoC uses 59% less silicon area and 74% less metal area than the RAW network, because there are much less switches and links in the application-specific NoC than the RAW network. The RAW network has 9 switches and 21 links, while the application-specific NoC has only 2 switches and 10 links.

Compared to the RAW network, the application-specific NoC has the low switch capacity and link capacity, because it has fewer switches and links. However, it has the high switch utilization and link utilization (Figure 7). The application-specific NoC uses the network resources more efficient than the RAW network. The application-specific NoC has 63% lower switch capacity and 69% lower link capacity than the RAW network, but it has 139% higher switch utilization and 231% higher link utilization than its counterpart.

5.3. Summary

Because in the two H.264 HDTV decoder SoC architectures the processors, memories, and input and output agents are the same, the mapping is also the same, and the NoC circuits are designed based on the same standards, the results show the impacts of using different NoCs. The application-specific NoC is the winner in terms of high performance, low power, small silicon and metal area, and the high switch and link utilization. The JM reference model is not optimized, so the absolute performance is low. The reference model can be optimized and show much higher performance. As the analysis showed, our conclusions will not affect by the optimization of the reference model. High utilization is a sign of good network designs, because not only it indicates efficient usage of resources but also it is good for the deep-submicron (DSM) technologies. In DSM technologies, low utilization wastes not only chip areas but also leakage power.

6. CONCLUSIONS

Our study shows: first, the on-chip communication network greatly affects the performance of the H.264 HDTV decoder SoC; second, the application-specific NoC is better than the regular-topology NoC, the RAW network. The application-specific NoC used 39% less power, 59% less silicon area, 74% less metal area, 63% less switch capacity, and 69% less link capacity to achieve 2X performance compared to the RAW network. These results are consistent with another NoC study for a high-performance embedded vision system [9].

7. REFERENCES

[1] ITU-T Recommendation H.264, “Advanced video coding for generic audiovisual services”, May 2003

[2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra,"Overview of the H.264/AVC video coding standard", IEEE Transactions on Circuits and Systems for Video Technology, July, 2003, p 560-576

[3] Jorn Ostermann, Jan Bormans, Peter List, Detlev Marpe, Matthias Narroschke, Fernando Pereira, Thomas Stockhammer, Thomas Wedi, “Video coding with H.264/AVC: Tools, performance, and complexity”, IEEE Circuits and Systems Magazine, 2004, p 7-28

[4] Erik B. Van Der Tol, Egbert G.T. Jaspers, Rob H. Gelderblom, "Mapping of H.264 decoding on a multiprocessor architecture", The International Society for Optical Engineering, v 5022 II, 2003, p 707-718

[5] Hae-Yong Kang, Kyung-Ah Jeong, Jung-Yang Bae, Young-Su Lee, Seung-Ho Lee, "MPEG4 AVC/H.264 decoder with scalable bus architecture and dual memory controller", IEEE International Symposium on Circuits and Systems, 2004, p II145-II148

[6] Yu-Wen Huang, Bing-Yu Hsieh, Tung-Chien Chen, Liang-Gee Chen,"Hardware architecture design for H.264/AVC intra frame coder", International Symposium on Circuits and Systems,May 2004

[7] Qiang Peng, Jin Jing, "H.264 codec system-on-chip design and verification", 5th International Conference on ASIC, Oct. 2003

[8] Jiang Xu, Wayne Wolf, Joerg Henkel, Srimat Chakradhar, “A Methodology for Design, Modeling, and Analysis of Networks-on-Chip”, ISCAS 2005

[9] Tiehan Lv, Jiang Xu, I. Burak Ozer, Wayne Wolf, Joerg Henkel, Srimat Chakradhar, “A Methodology for Architectural Design of Multimedia Multiprocessor Systems-on-Chips”, IEEE Design and Test of Computers, Dec. 2004

[10] W. Dally, B. Towles, “Route packets, not wires: on-chip interconnection networks”, Proceedings of the 38th Design Automation Conference, 2001

[11] S. Kumar, A. Jantsch, J–P Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani, ”A network on chip architecture and design methodology”, IEEE Computer Society Annual Symposium on VLSI, 25-26 April 2002

[12] M. Taylor, “The Raw prototype design documentation” v5.00, http://cag-www.lcs.mit.edu/raw/documents/

[13] M. Forsell, “A scalable high-performance computing solution for networks on chips”, IEEE Micro, Volume: 22, Issue: 5, 2002

[14] R. Hofmann, B. Drerup, “Next generation CoreConnect processor local bus architecture”, Annual IEEE International ASIC/SOC Conference, 25-28 Sept. 2002

[15] D. Flynn, “AMBA: enabling reusable on-chip designs”, IEEE Micro, Volume: 17 Issue: 4, July-Aug. 1997

[16] D. Wingard, “MicroNetwork-based integration for SOCs”, Design Automation Conference, 18-22 June 2001

[17] A. Adriahantenaina, H. Charlery, A. Greiner, L. Mortiez, C.A. Zeferino, “SPIN: a scalable, packet switched, on-chip micro-network”, DATE 2003

[18] www-device.eecs.berkeley.edu/~ptm/interconnect.html

[19] www.cadence.com

[20] www.synopsys.com

[21] www.opencores.org

[22] www.opnet.com

[23] http://iphome.hhi.de/suehring/tml/