Star-Mesh NoC based Multi-Channel H.264 Decoder Design

June-Young Chang
SoC Research & Development Group
Electronics and Telecommunications Research Institute,
Daejeon, Korea
jychang@etri.re.kr

Won-Jong Kim, Young-Hwan Bae, Mi-Young Lee,
Ju-Yeob Kim, Han-Jin Cho
SoC Research & Development Group
Electronics and Telecommunications Research Institute
Daejeon, Korea
wjkim, yhbea, sharav, juyeob, hjcho@etri.re.kr

Abstract—**In this paper we described the architectural exploration of Star-Mesh NoC based multi-channel H.264 decoder. The Star-Mesh NoC is comprised of local star switch and global mesh switch. By analyzing data transfers among the processors, IPs, and memories, we partitioned IPs into clusters to map then to Star-Mesh NoC architecture. In order to enhance data parallelism and NoC utilization, H.264 decoder IPs with much data traffic are mapped to star switch and shared memory is connected to mesh switch where star switch connected to mesh switch with 1-hop. We explored several mapping architecture to achieve improvement of the system throughput.**

Keywords- NoC; Multi-Channel H.264 decoder; Star-Mesh NoC; Architecture exploration

I. INTRODUCTION

As the System-On-a-Chip (SoC) grows in design complexity, data traffic of IP cores becomes increasingly important. Especially in multimedia SoCs design, such as video phone, teleconference system, 3G-324M, MPEG-4, H.264, and HDTV codec, a considerable amount of data traffic is required. To accommodate all modules with sufficient data traffic bandwidth, SoC designer must pay attention to on-chip interconnect design [1]. In NoC(Network-on-Chip)-based design, forecasting data traffic in SoC's and designing suitable data communication architecture are important. The NoC architecture provides parallel communication among existing IP cores to improve data traffic bandwidth. Also NoC's direct connection feature between IP cores eliminates the need for different interface implementations for different bus widths thus improving the scalability of communication architecture. As the number of IP cores on SoC increases, bus efficiency will degrade due to bus collisions from multiple masters. NoC is the a communication architecture for SoC design that overcomes the limits of OCB(On-Chip Bus) architecture by providing higher data traffic bandwidth and better scalability [2][3]. In NoC based SoC design, the system performance depends on 1) NoC architecture, 2) the clustering of application IPs by the data traffic, 3) Shared memory mapping are influenced on the system performance.

In this paper we describe the architectural exploration of Star-Mesh NoC based multi-channel H.264 decoder. The Star-Mesh NoC architecture, which analyzes the data traffic and utilization of the shared memory, improves the system throughput. The result is verified through a simulation. This paper is organized as follows: In section II we give an overview of the Star-Mesh NoC architecture. The architectural exploration of Star-Mesh NoC based multi-channel H.264 decoder is described in section III. In section IV explain various architecture exploration results to implement multi-channel H.264 decoder on Star-Mesh NoC. Finally section V describes conclusions.

.

II. STAR-MESH NETWORK-ON-CHIP

NoC architecture has structures like Star, Mesh, Tree, Ring, which can use appropriate NoC structure in application fields. In this paper, we suggest a Star-Mesh NoC where local communication is done by star switch and global

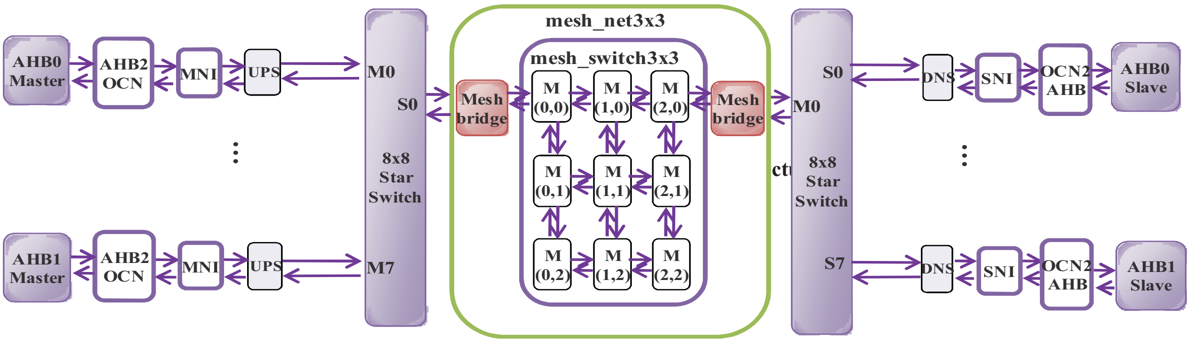

Figure 1. Star-Mesh NoC architecture

978-1-4244-2599-0/08/$25.00 ©2008 IEEE

communication is done by mesh. Star-Mesh mixed structure has communication architecture which can make parallelism into its highest, and offers easy access to memory. It can maximize locally parallel communication and provide easy access to global resources. In Star-Mesh NoC, Star NoC can be a 8x8 star switch which can connects 8 masters or slave IPs via MNI(Master Network Interface) or SNI(Slave Network Interface). AHB2OCN and OCN2AHB are provided IPs with AHB interface. Mesh NoC is organized in 3 components. Mesh switch uses X-Y routing algorithm. Star-Mesh bridge is used to connect star switch and mesh switch, it converts star-type data format to mesh-type data format. Mesh network interface is the network interface that connects mesh switch to IP directly. Star switch can basically connect 8 IPs with 8x8 switch. Mesh NoC is built with 3x3 switch, 4x4 switch and 5x5 switch. An example Star-Mesh NoC architecture with 3x3 mesh NoC is shown in Figure 1.

III. Atrchitecture Exploration of Multi-Channel H.264 Decoder

In this section, we explain the architecture exploration to implement multi-channel H.264 decoder with various mapping on Star-Mesh NoC. We also explain the way to achieve the highest parallelism by clustering and how to improve utilization in memory by IP mapping.

A. Communication based clustering of H.264 decoder

Figure 2 shows an implementation of 1-channel H.264 decoder SoC based on single bus architecture [4].

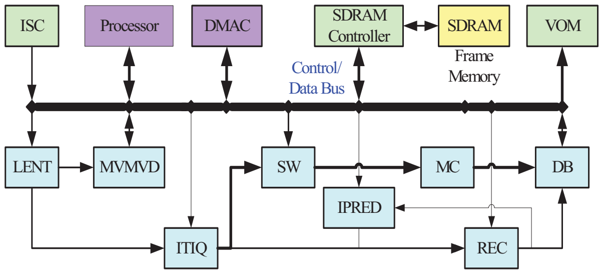

Figure 2. Single-bus architecture of H.264 decoder

It is composed of 32-bit RISC processor and dedicated hardware engines (IPs). The hardware engines are implemented for fixed functions in H.264, such as input stream control (ISC), low-level entropy decoding (LENT), inverse transform and inverse quantization (ITIQ), intra-frame prediction (IPRED), inter-frame prediction and motion compensation (MC), reconstruct frame (REC), and de-blocking filter (DB). Each IP has a local buffer for pipelined or parallel operation. The processor controls IPs and decodes stream header information. Reference frames and reconstructed frames are stored in SDRAM.

We analyzed data traffic of H.264 decoder to maximize parallelism and map it on Star-Mesh NoC. Average data traffic amounts among H.264 decoder IP are shown in TABLE I. There are lots of data traffic between processor and ITIQ, MVMVD, IPRED and between DMA and LENT, MC, DB, VOM. Therefore we can group then into two clusters one cluster contains processor, ITIQ, MVMVD, IPRED and REC and the other DMA, LENT, MC, REC, DB and VOM. To do map 1-channel H.264 decoder we need at least two 8x8 star switch and a mesh switch for connection between star switches.

TABLE I. Data traffic of H.264 decoder

From/To	Proc.	MEM(DMA	DB	MC	ITIQ	LENT	VOM	REC	MVMVD	IPRED
Proc.	-	-	11,092	10,972	6,348	7,920	1	236	33,804	2,022
MEM(DM	-	-	22,568	362,837	-	2,139	163,584	1,351	-	-
DB	-	242,336	-	-	-	-	-	-	-	-
MC	-	11,760	38,107	-	-	-	-	-	-	-
ITIQ	-	-	-	202,645	-	-	-	21,883	-	-
LENT	13,881	-	-	-	224,528	-	-	-	16,887	-
VOM	-	-	-	-	-	-	-	-	-	-
REC	-	912	10,942	459	-	-	-	-	-	4,949
MVMVD	28,208	-	-	-	-	-	-	-	-	-
IPRED	-	-	-	-	-	-	-	10,942	-	-

B. Mapping into Star-Mesh NoC

H.264 decoders IPs are separated into two clusters and they are mapped on Star-Mesh NoC. To improve data parallelism, IPs having a lot of data traffic are connected to same star switch. Star SW0 connects DMAC, LENT, MC, REC, DB and VOM, and SW1 connects processor, ITIQ, MVMVD and IPRED. SDRAM used for storing video data are mapped in the same way as SW0. 1-channel H.264 decoder is organized by star SW0 and SW1, between these SW there is a mesh switch and provides several channels for data transfers.

1) Single SDRAM

The structure of 2-channel H.264 decoder mapped to Star-Mesh NoC is shown in Figure 3. 1-channel H.264 decoder uses two star switches. There is a mesh switch between them provides several channels for data transfers. Decoder_0 communicates data through star SW0 and SW1, Decoder_1 process data through SW2 and SW3. DMA controller and SDRAM are mapped to star switches SW0 and SW2.

In this structure, each DMA controller transfers data in parallel via each channel with separate SDRAMs. Therefore it requires two discrete SDRAMs. However in industrial applications using two discrete SDRAMs is quite a big burden. TABLE II shows the result of simulation with system clock 30MHz for foreman CIF (352x288) video sequence by 2-channel H.264 decoder. Processing 7 frames takes an average of 326.9ms in each decoder.

TABLE II. Simulation result of Single SDRAM

2-channel H.264 decoder(Single SDRAM)								
Frame(ms)	I	P1	P2	P3	P4	P5	P6	AVE
Decoder_0	147	362	340	355	362	361	363	**326.9**
Decoder_1	147	362	340	355	362	361	363	**326.9**

2) Common SDRAM

To reduce resource overhead we can share SDRAM and SDRAM controller for all the decoders. To share resources fairly we connected them to the center mesh switch MS11 as

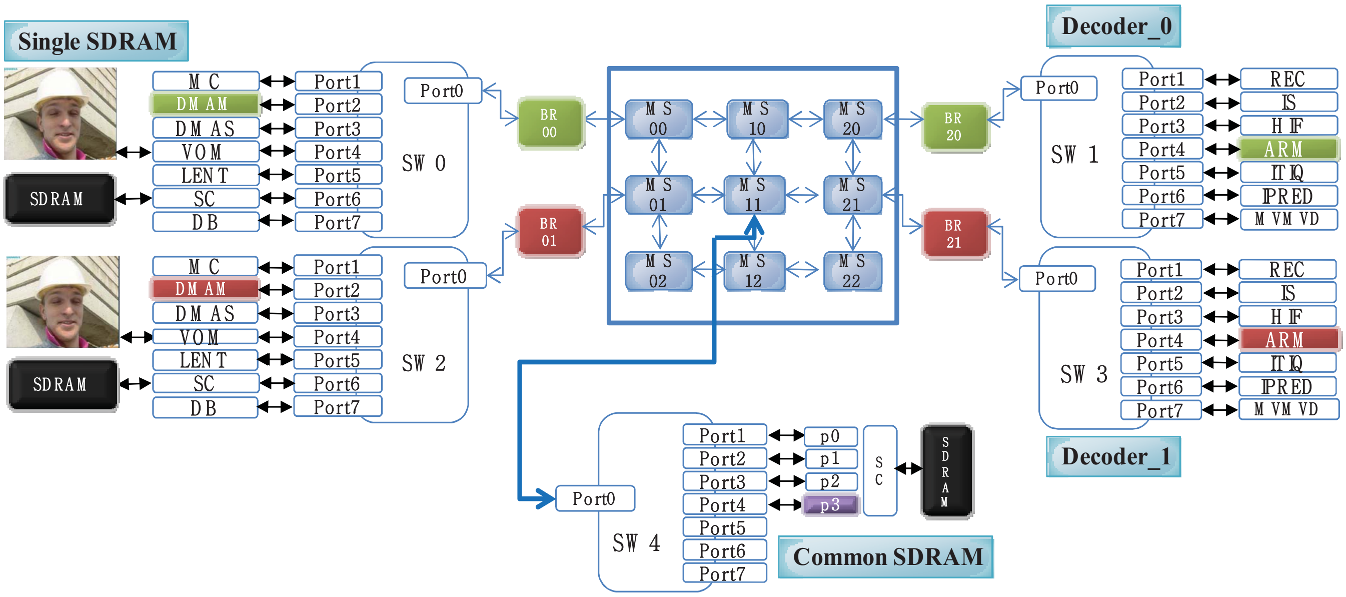

Figure 3. Star-Mesh NoC based 2-channel H.264 decoder

shown in Figure 3. In this architecture, sharing resources can improve resource utilization but the performance is degraded because of increased data path length and data transfer conflictions. In Figure 3, SW2 IPs transfers data by MS01->MS11 to access SDRAM which is connected to MS11 while SW0 IPs SDRAM by 2-hop. In the Star-Mesh NoC architecture with common SDRAM, we can improve performance by using higher frequency clock for the SDRAM.

For low-power design we can use low frequency clock for the SDRAM. We can trade-off for low power and high performance between both architectures.

TABLE III. Simulation result of Common SDRAM

2-channel H.264 decoder(Common SDRAM)								
Frame(ms)	I	P1	P2	P3	P4	P5	P6	AVE
Decoder_0	238	967	893	945	968	966	967	**849.1**
Decoder_1	238	967	895	946	968	948	987	**849.9**

IV. Star-Mesh NoC based Multi-Channel H.264 Decoder

We now extended 1-channel H.264 decoder to 3-channel H.264 decoder as shown in Figure 4. Separate 3-channel H.264 decoder by data traffic and allocate it into star switch. By connecting each DMA to star switch with 1-hop distance from the star switch with SDRAM, it uses only 1-hop latency for data transfers.

TABLE IV. Simulation result of 3-channel H.264 decoder

3-channel H.264 decoder								
Frame(ms)	I	P1	P2	P3	P4	P5	P6	AVE
Decoder_0	240	983	908	959	983	982	983	**862.6**
Decoder_1	244	1,032	1,028	1,033	1,029	1,031	1,029	**918.0**
Decoder_2	246	1,032	1,028	1,024	1,029	1,031	1,035	**917.9**

The SDRAM controller that controls SDRAM will be organized with 4-channels: channel-0 is allocated to Decoder_0, channel-1 is allocated to Decoder_1, channel-2 will be allocated to Decoder_2, channel-3 is used to by ARM processor control SDRAM controller. When all decoders access SDRAM at the same time, the priority controller in MS11 switch will access them one after another. Table IV describes the simulation result with Seamless CVE[5].

V. Conclusions

In this paper explained various kinds of architectural exploration results to implement multi-channel H.264 decoder on Star-Mesh NoC. We analyzed H.264 decoder data traffic and star switch mapping of the IPs where there are lots of traffic, and building a Star-Mesh NoC structure by connecting it altogether by mesh switch. By using common SDRAM in Star-Mesh NoC architecture, we could increase resource utilization. We can trade-off for low power and high performance between both architectures.

Acknowledgment

This work was partly supported by the IT R&D program of MIC/IITA [2008-S-001-01, A Smart Eye based on Convergence SoC]

References

[1] Kyeong Keol Ryu, Eung Shin, Mooney, V.J., “A comparison of five different multiprocessor SoC bus architectures”, Digital Systems, Design Proceedings. Euromicro Symposium on , Sept. 2001. 4-6, pp.202–209,

[2] June-Young Chang, Won-Jong Kim, Young-Hwan Bae, Jin-Ho Han, Han-Jin Cho and Hee-Bum Jung, Performance Analysis for MPEG-4 Video Codec based on On-Chip Network”, ETRI Journal, vol.27, no.5, Oct. 2005, pp.497-502.

[3] June-Young Chang, Young-Hwan Bae, Mi-Young Lee, Han-Jin Cho, “On-Chip Network based Virtual Platform for Multimedia Applications”

International Conference on Circuits/Systems, Computer and Communications, vol2, July.2007, pp.649-650

[4] Wonjong Kim, Seunchul Kim, Hanjin Cho, " Architecture Exploration of H.264 Decoder SoC," The 21rd International Technical Conference on Circuits/Systems, Computers and Communications, 2006, pp.I469-I471

[5] Seamless CVE manual :

[6] http://www.mentor.com/products/fv/hwsw_coverification/seamless/

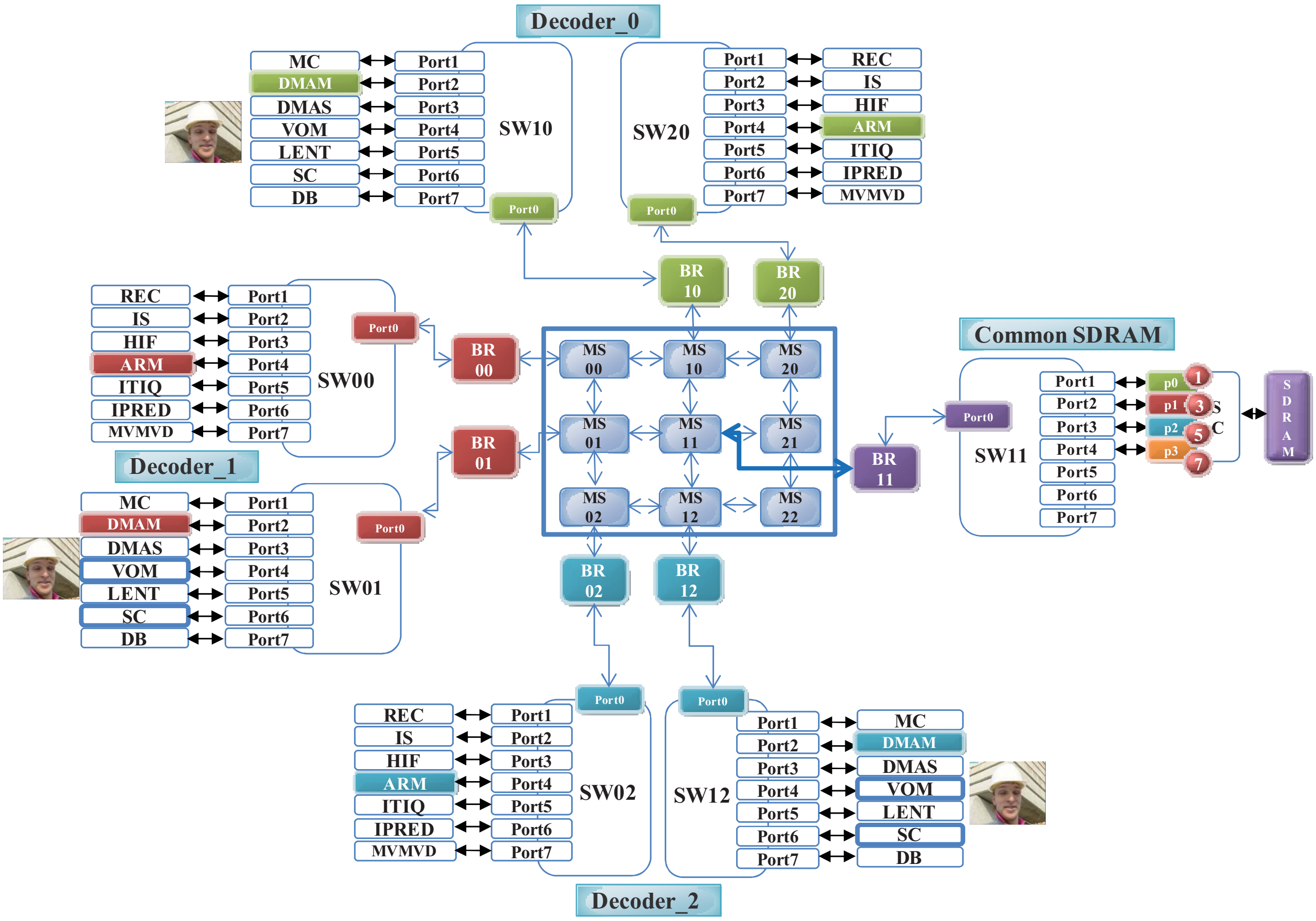

Figure 4. Star-Mesh NoC based 3-channel H.264 decoder