VLSI Design and Implementation of a High-Speed Multicast Switch Fabric

C. Li[†‡], *Member, IEEE,* R. Venkatesan[†‡], *Senior Member, IEEE,* and H. M. Heys[†‡], *Member, IEEE*
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, NL, A1B 3X5, Canada

***Abstract*— This paper presents the VLSI design and implementation of a new cell-based high-speed multicast switch fabric using the $0.18\mu m$ CMOS technology. Using distributed control, multistage interconnection network structure, and modular design, the multicast Balanced Gamma (BG) switch features a scalable, high performance architecture for unicast, multicast and combined traffic under both uniform and non-uniform traffic conditions. The BG switch follows predominantly an output-buffered architecture and utilizes a self-replication mechanism for multicast traffic switching. In the paper, we discuss in detail the front-end design issues of the switch using the ASIC design flow recommended by the Canadian Microelectronics Corporation (CMC) [1]. Synthesized results are provided for measures of circuit complexity and timing.**

***Index Terms*— Multicast, Balanced Gamma (BG) switch, VLSI design, CMOS technology, multistage interconnection network (MIN), scalability.**

I. Introduction

Communication network capacity and applications have been changing at an enormous rate for the past two decades. Besides very high speed, many real-time applications, such as videoconferencing, music on demand, and video on demand require messages to be sent to more than one destination. As a result, supporting multicast has become a necessary requirement for any switch designed for future broadband communication networks.

ATM-like fixed-sized packet switching attracts much interest because of its application in advanced Internet routers and switches. A variable-sized incoming IP packet is internally segmented into fixed-size ATM-like cells which are switched to the output ports, where they are reassembled into the IP datagram. Borrowing from ATM terminology, we use the term cell to identify the fixed-sized packet used in the switch, which can be ATM cells, or any other convenient data format [2].

Multistage interconnection network (MIN) design has become an attractive solution for broadband switch architecture due to the many desirable features such as self-routing, distributed control, modularity, constant delay for all input-output pairs and suitability for VLSI implementation. A multicast switch fabric using the implicit cell replication is preferred because it combines the routing and replication functions into a single unified network. This kind of design will inherit most of the attractive features of the MIN design.

In this paper, we present the VLSI design and implementation of a new scalable cell-based multicast switch called the Balanced Gamma (BG) switch. The input-output buffer design and high throughput ensure that the switch can achieve high performance while keeping the hardware complexity reasonable. The multicast BG switch utilizes a multi-path MIN design. The multicast cell replication function is integrated into the functionality of each switch element (SE) and is performed in a distributed fashion along with the routing. The use of unbuffered SEs ensures proper cell sequencing. Scalability is easy to achieve by using the SE as the basic building block.

II. Switch Architecture

A. Architecture and Switching Operation

Unlike the banyan network, which utilizes a 2×2 SE, the multicast BG network utilizes a 4×4 SE as the basic building block. Figure 1 shows the architecture of an 8×8 multicast BG switch. The basic architecture of an $N \times N$ BG multicast switch consists of N input port controllers (IPCs), an $N \times N$ multistage interconnected switch fabric that supports self-routing, self-replication and delivery acknowledgement, and N output port controllers (OPCs). No dedicated copy network to support the replication functionality is required. The IPC terminates the input signals from the network, strips the information contained in the cell header, and uses a lookup table to determine the destinations. The switch fabric is the core of the multicast BG switch. An $N \times N$ BG switch fabric consists of $n + 1$ stages, where $n = log_2 N$, with each stage consisting of N SEs numbered from 0 to $N - 1$. In stage 0, 1×2 SEs are used and in stage 1, 2×4 SEs are used. Each of the following $n-2$ stages is comprised of 4×4 SEs. The last stage is the output buffer stage, which can accept up to 4 cells per output port in one switching cycle. Network bandwidth is expanded through the first two stages and then remains the same for all subsequent stages. Through internal bandwidth expansion, the multicast BG switch can achieve better performance while keeping the hardware complexity reasonable. The OPC includes a regulator and scheduler. It updates each arrived cell with a new cell header and sends onto the output link. Details on the architecture design and the design choices justification can be found in [3], [4].

Inside the multicast BG switch, a three-phase switching operation is performed [4], [5]. First is the reservation phase

† This work is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

‡ This work is supported in part by the VLSI design facilities sponsored by Canadian Microelectronics Corporation (CMC).

§ Part of this work is reviewed by the *Intl. J. of Communication Systems*.

during which the tag of the HOL cell is routed through the SF. Tag information is used by the SF for internal path setup and is generated and attached to each cell by the IPC. Multicast cell replication is performed implicitly by the SEs along with the routing operation only when necessary. Next comes the acknowledgement phase, during which cell delivery information is reported to the IPC by use of backpressure mechanism. Based on that information, the IPC decides whether the HOL cell should be transmitted or kept in the input buffer for the next cycle. In the third phase, the payload is transmitted via the established path just as in a circuit switch. Because of the memoryless design of the SF, a cell will either reach the desired output port(s) or be kept in the input buffer. Cell sequence is thus maintained. Cell loss occurs only when the input buffer is full and a new cell arrives.

B. Self-Routing and Self-Replication Algorithm

In the multicast BG switch, there are three types of SEs: the 1×2 and 2×4 SEs are used for the first two expanding stages while the 4×4 SEs are used for all subsequent stages. Functionally, the first two types of SEs can be treated as a special (simpler) case of the 4×4 SE. Therefore, in the following discussion, the more general 4×4 SE is used, which is shown in Figure 2. The four output links are numbered 0 to 3 from top to bottom. Among the four links, link 0 and link 2 are called upper and lower regular links, respectively, while link 1 and link 3 are called upper and lower alternate links, respectively. Both the regular link and its alternate link have the same capability of reaching the same destination. Upon switching, the regular links are always used first. The alternate link is used only when the regular link has already been assigned to a connection.

In the multicast environment, tag design becomes more challenging because not only the routing information should be carried but also the cell replication information, and the tag length should be minimized to minimize the delay in the reservation phase. In the BG switch, for each SE to make the right routing and replication decision, a 2-bit tag is used by each SE for each input link. Four different actions can be taken by the SE and these are summarized in Table I.

Priority switching is a feature considered in the multicast BG switch, with up to 8 priority levels currently supported. The SE will make its decision in two steps. Firstly, the SE decides cell processing order based on the priority level associated with each cell. Secondly, incoming cells are switched following the order determined in the first step. Cells with higher priority are always processed first until all incoming cells are processed or all the sources are used up. In the latter case, the remaining low priority cells will be blocked. An example is provided in Figure 2 in which cells are coming in from the top three input links. By sorting on the priority tag, the process order is $c \rightarrow b \rightarrow a$. Following the routing and replication table, cell c is a unicast cell which requests an upper output link, it is switched to output 0 and similarly cell b is switched to output 2. For cell a, the tag bit pair '11' indicates that replication is required. The available outputs are checked and cell a is replicated and sent to both upper and lower alternative output links, links 1 and 3, respectively.

Bit 1	Bit 0	Routing Action	Replication Action
0	0	Idle (no action)	Idle (no action)
0	1	Lower link	No replication
1	0	Upper link	No replication
1	1	Both links	Replication

TABLE I

ROUTING AND REPLICATION ACTIONS BASED ON TAG PAIR INFORMATION.

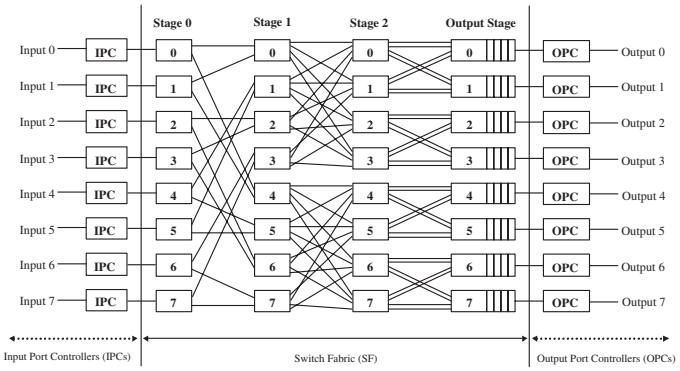

Fig. 1. The architecture of an 8×8 multicast BG switch

III. VLSI DESIGN AND IMPLEMENTATION

The analysis of the multicast BG switch has demonstrated its high performance under various unicast, multicast, and mixed traffic conditions [4]. At the same time, it is scalable in terms of architecture and performance. Therefore, the VLSI design is conducted to investigate the feasibility of building a practical switch using this architecture. Hardware description languages VHDL and Verilog and the $0.18\mu m$ CMOS technology are used. Modularity and implementation scalability are emphasized during this stage.

A. Digital System Design Flow and Methodology

In general, the design of the BG switch follows the top-down design and bottom-up implementation. VLSI design is an iterative process. An initial design idea goes through several transformations before the final hardware implementation is obtained. At each step of transformation, the designer checks the result of the last transformation, adds more information, and passes it through to the next step of the transformation. When all the design and verification steps are completed, a stream file to describe the mask layer information for the circuit will be created. This is the file which is used for chip fabrication after Design Rule Checking is completed.

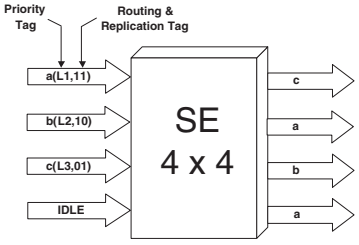

Fig. 2. Self-routing and cell replication in the 4×4 SE.

The design flow that consists of nice steps can be divided into two stages. The first four steps (RTL simulation, Synthesis, Scan Insertion, Gate-level Simulation) belong to the front-end design stage, in which VHDL and Synopsys tools are used. The remaining five steps (Floorplanning, Placement, Clock Tree Generation, Routing and Timing Verification, Stream File) comprise the back-end design stage, in which Verilog and Cadence tools are used. Simulation, synthesis and timing are the major concerns for the front-end design. A design idea is converted to a gate-level netlist (VITAL list) through this process. In the back-end design, the major interest is on the floorplanning, placing and routing of the imported gate-level netlist onto the silicon wafer. In this paper, we focus on the front-end design issues of the switch fabric.

B. Design and Implementation

The switch adopts a three-phase switching method and follows a single-plane, bufferless, and non-pipelined SF architecture. Using the divide-and-conquer strategy, the SF is first partitioned into n stage components and then each stage component is partitioned into SEs, stage controller and sequencer, as shown in Figure 3. The sequencer provides timing information for the stage controller. The stage controller, basically a state machine, provides the control signals for all SEs within the stage. The sequencers and stage controllers are now terminal nodes, while the SEs require further partitioning.

Figure 4 depicts the architecture of a 4×4 SE. To support the three-phase switching operation, the internal architecture of each SE is designed to provide two channels, forward and backward. Three major functional blocks are included: the forward-path control unit (FCU), the admission control unit (ACU) and the backward-path control unit (BCU).

The FCU is used for tag and payload transmission, which is comprised of the tag receiving buffer bank, tag pushout buffer bank, and source/path select multiplexer group. The tag receiving buffer bank is used to temporarily hold the tag bits, which are used by the FCU to make the routing and replication decision. It is a serial-in-parallel-out device. The tag pushout buffer bank stores the tags ready to be pushed out to the next stage. It is a parallel-in-serial-out device and its parallel outputs are simultaneously loaded into the tag pushout buffer bank for transmission. The multiplexer group provides the connection between the input and output ports. During the reservation phase, outputs from the pushout buffer bank are used as the inputs, while during the data transmission phase, the data input line is directly fed to the select multiplexer group and the two buffer banks are then bypassed.

The BCU is used for acknowledgement transmission. Unlike the FCU, as information is traversing from a downstream stage to an upstream stage, the length of acknowledgement is getting longer like a converging tree. In addition to the two types of buffer banks, the BCU includes the blocking information generation circuitry (BIG) and the acknowledgement output generation circuitry (AOG). BIG circuitry is used to generate a negative acknowledgement for a cell that loses its output contention. AOG circuitry is used to generate the final acknowledgement outputs that are passed back to the previous

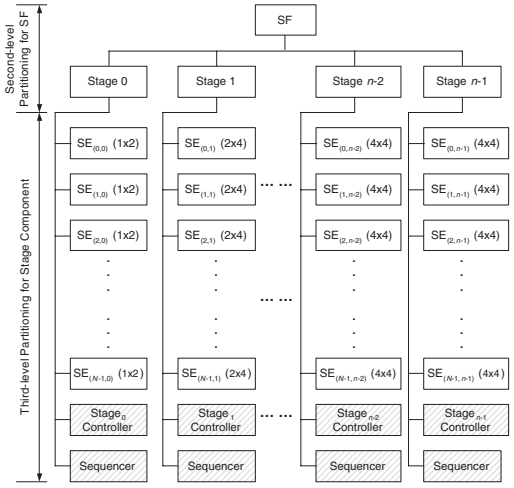

Fig. 3. Resulting structure for SF after second and third level partitioning

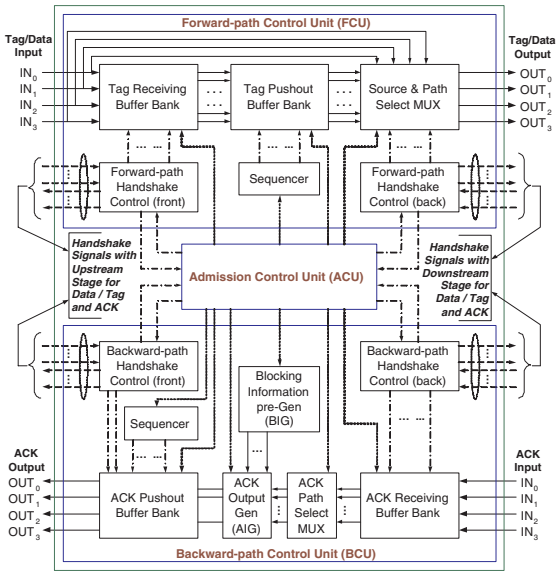

Fig. 4. The internal architecture of an 4×4 SE

stage. Two scenarios should be considered: a) when a cell is successfully transferred, and b) when blocking occurs during switching. In the first case, the work for the AOG circuitry is to concatenate the received acknowledgement with 0s (when it is a unicast cell) or concatenate the acknowledgement received from the upper link to that from the lower link from downstream stages (when it is a multicast cell), and form its final ACK output. In the second case, the acknowledgement for the blocked part is generated by the BIG circuitry immediately while the acknowledgement for the successfully transferred part comes from downstream stages. They are concatenated by using the AOG circuitry and then loaded into the pushout buffer bank for transmission.

The ACU is the heart of the SE. The most important work the ACU does is to make routing and replication decisions based on the received tag information. Priority routing is a feature considered in the multicast BG switch to resolve link contentions. The cell with the highest priority is assigned

to the output first and then the cell(s) of the next priority level and so on until either all the outputs are assigned or all active incoming cells are processed. When all the outputs are assigned, the remaining requests are blocked.

C. Functional Testing

For the complete multicast BG switch, it is too complex and difficult to conclude correct switching operation through waveform observation. Alternative and automatic verification method has been considered. A combined high-level language and hardware description language test method bas been used. A C/C++ program has been developed to generate cells to be switched by the fabric. The generated data is stored in a data file which emulates the input queue, one for each input port. During each switching cycle, the testbench program retrieves the HOL data from the file, sends to the SF, and checks the acknowledgement output from the SF. If it is a negative result, the HOL cell will be switched again during the next cycle. Otherwise, the next cell will be switched until either all data in the file is read out or the testing time is reached.

To verify the correctness of switching by the hardware, data at different places are recorded into different output files by the testbench program. Data recording is performed on a switching cycle basis. When simulation is completed, the data files are sent to a C/C++ program for analysis and verification. Payload is used to verify the source of each delivered cell for each switching cycle. The delivery status will be updated accordingly and compared with the received acknowledgement data. This process continues until all the output lines are checked. Whenever a mismatch is detected, an error message will be asserted by the C/C++ program. This process repeats for every switching cycle until all the data is processed.

IV. Hardware Complexity and Timing

Hardware complexity and timing are the two most important measures to evaluate a hardware design. The results are collected from the Synopsys synthesis tool $DesignCompiler$, using the library targeting at 0.18 μm CMOS technology. The synthesized circuit, which is in square microns (μm^2), is converted to gate count with reference to the area of a two-input NAND gate. The total gate count is comprised of combinational logic part and non-combinational logic part.

Table II presents the hardware complexity of the various SE type and its subcomponents for the 16×16 multicast BG switch. Hardware complexity of the whole SF of various switch sizes are summarized in Figure 5.

To obtain the timing, the whole 16×16 SF is constrained with the $5ns$ system clock and other constraints and synthesized using the $DesignCompiler$. With this clock rate, no positive slack is generated in the report file. Therefore, the SF can comfortably run at 200 Mbps link speed, which is enough for the OC-3 links and yields a switching capacity of greater than 3 Gbps for the overall switch. The resulting gate level circuit is simulated again using the testing method mentioned above to verify its functionality. With more advanced technologies, such as the $0.13\mu m$ and $0.09\mu m$ CMOS technology, the speed that the design can support will be even higher. Similar timing results have been achieved for larger switch sizes in which the SE and stage controller at different stages are synthesized using the same $5ns$ system clock.

Stage and SE Type	FCU		BCU		ACU		Overall	
	Comb.	Seq.	Comb.	Seq.	Comb.	Seq.	Comb.	Seq.
$SE\ (Stage_0)$	70	303	135	235	633	0	838	538
$SE\ (Stage_1)$	98	371	214	235	857	0	1169	606
$SE\ (Stage_2)$	172	500	151	176	964	0	1287	671
$SE\ (Stage_3)$	195	378	83	88	935	0	1213	466

TABLE II
HARDWARE COMPLEXITY OF THE VARIOUS SE TYPE AND ITS SUBCOMPONENTS FOR 16×16 MULTICAST BG SWITCH

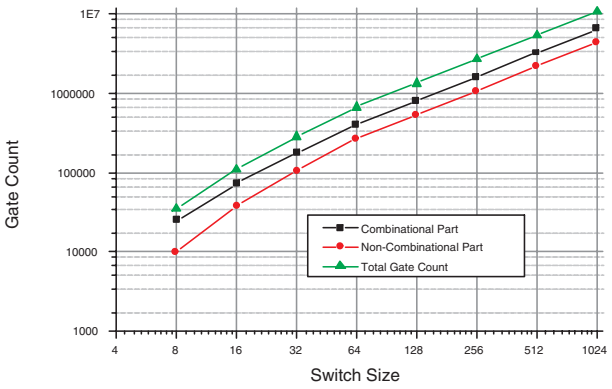

Fig. 5. Hardware complexity for switch fabric of various sizes.

V. Conclusion

In this paper, we have explained the design process of the multicast BG switch using the 0.18 μm CMOS technology. A distributed control is used in the design to fulfill the high-speed requirement. The similarity of functionality and structure of all the SEs facilitates a modular design and ease of VLSI implementation. Testing and verification methods have been discussed for large designs such as the complete switch fabric. Synthesis results are provided for switches of different sizes as well as for the each subcomponent inside the switch element. The implementation results indicate that the core of a 16×16 switch fabric can be easily fabricated into a single IC chip and can comfortably run at OC-3 link speed, which yields a switching capacity of close to 3 Gbps for the overall switch.

References

[1] Canadian Microelectronics Corporation (CMC) website: *www.cmc.ca*.
[2] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi and F. Neri, "Packet Scheduling in Input-Queued Cell-Based Switches," in *Proceedings of IEEE INFOCOM'2001*, vol. 2, pp. 1085-1094, Alaska, USA, April, 2001.
[3] C. Li, H. M. Heys and R. Venkatesan, "Design and Scalability of the Multicast Balanced Gamma (BG) Switch," *Proceedings of the Eleventh International Conference on Computer Communications and Networks (IEEE ICCCN'2002)*, p.p. 518-521, Miami, Florida, USA, October 2002.
[4] C. Li, *Design, Modelling, and Analysis of the Balanced Gamma Multicast Switch for Broadband Communications*. PhD Dissertation, Memorial University of Newfoundland, October, 2004.
[5] A. Pattavina, *Switching Theory: Architecture and Performance in Broadband ATM Networks*. New York: Wiley, 1998.