DESIGN AND SIMULATION OF THREE ATM ASICS

Chan Kim, Jong Arm Jun, Sang Ho Lee, Jae Geun Kim

Switching and Transmission Labs / ETRI
161 Kajeong-dong Youseong-goo Taejon, 305-350 Korea
e-mail : ckim@etri.re.kr

Abstract

In ASIC developments, effective simulation leads into functionally more reliable chip as well as faster development time. This paper describes the design and the simulation techniques used in 3 ATM ASIC developments in ETRI. The three ASICs described is : ASAH-NIC - a 155Mbps ATM SAR chip with internal PCI interface and SDH framer, ASAH-L4 - a 622Mbps bidirectional ATM Layer Processing chip with UPC, OAM, QOS buffering capabilities, and ASAH-P4 - a 622Mbps ATM Physical layer chip.

1. Introduction

In ASIC developments, simulation is the most important procedure of the development cycles and most time consuming work. Therefore, effective simulation leads into functionally more reliable chip as well as faster development time. This paper describes the simulation techniques used in 3 ATM ASIC developments in ETRI. These techniques can be generally applied to other ASIC designs.

The three ASICs described in this paper is : ASAH-NIC - a 155Mbps ATM SAR chip with internal PCI interface and SDH framer, ASAH-L4 - a 622Mbps bidirectional ATM Layer Processing chip with UPC, OAM, QOS buffering capabilities, and ASAH-P4 - a 622Mbps ATM Physical layer chip. This paper shows some example methods for making testbench as close to the real environment as possible and some special techniques to avoid common problems in the gate simulation process.

2. ASAH-NIC : ATM SAR chip with PCI core and SDH framer

2.1 Function and Architecture

ASAH-NIC is a 155Mbps SAR(Segmentation and Reassembly) processing ASIC with internal PCI interface controller and 155Mbps SDH framer. It has the architecture shown in Fig. 1.

The PCI core, which was bought from Phoenix Technologies Inc. has PCI master and slave functions with four FIFOs each for master, slave and read, write operation. It provides convenient handshaking interface with the application side for ASIC designers. The DMA master enables ASAH-NIC to read and write as a PCI master and the DMA slave enables ASAH-NIC to be read or written as a PCI slave.

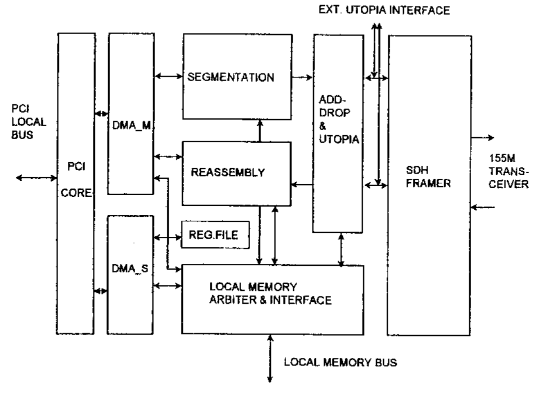


Fig. 1. Architecture of ASAH-NIC

The segmentation block segments the packets in the PCI memory into ATM cells in the order specified in the schedule table in the external SRAM. It reads the VC table pointed to by the schedule table and the buffer descriptor pointed to by that VC table and decodes the DMA parameters according to the AAL type and buffer reading location. It then commands PCI memory read to the DMA master and updates the VC table contents like buffer read count and temporary CRC32 value. The actual cell assembly is performed at the DMA master's read part. Since the arrival of the buffer descriptor and the actual segmentation of the buffer is not in the same order, the buffer descriptor spaces are managed as linked list and there is the buffer descriptor queue in the external SRAM for communication with external CPU. It manages the buffer read pointer in the buffer for many connections.

The reassembly reads ATM cells from the cell buffer in the external SRAM and reads the VC table address that was attached to the cell by the UTOPIA receiver when the cell was written into the cell buffer. It then decodes the DMA parameter for PCI memory write according to the AAL type and the last writing location in the buffer etc. The partial CRC32 and buffer write pointer is kept in the VC table. Since the input cells are reassembled sequentially into the PCI memory, the cell buffer space is managed as FIFO.

The local memory arbiter and interface logic provides for a common burst read and burst write interface to its client blocks independently of the external SRAM speed. It has internal FIFO for the speed adaptation but the latency caused by the arbitration was minimized. It also supports two level priority control and fairness among lower priority blocks.

UTOPIA and add-drop interfaces with physical layer and when a cell is received, it looks up the connection table using hash chain method and determines whether the cell should be dropped to the cell buffer or bypassed back to the transmitter. UTOPIA is served at the top priority in the local memory arbiter. STM framer performs 155Mbps ATM physical layer processing including cell rate adaptation, HEC insertion/validation, cell delineation, cell payload scrambling, frame overhead insertion/extraction, frame scrambling etc. The framer logic was a formally developed circuit using schematic capture.

2.2 Simulation

The difficulty of simulation involving PCI bus resides in the fact that PCI is very complex to simulate. The protocol is complex and control is distributed among several masters and slaves. The testbench of ASAH-NIC looks like Fig.2. It is based on the testbench supplied by Phoenix Technologies Inc., the supplier of the PCI core used in ASAH-NIC. This testbench was modified so that ASAH-NIC's PCI master and slave function can be verified with other main ATM functions. With ASAH-NIC attached to a PCI bus, there are other virtual PCI maters and PCI slaves each having memory pools for data holding. There are also a PCI arbiter and a protocol monitor. By controlling the virtual masters and slaves to setup and run ASAH-NIC, the function of ASAH-NIC can be verified. The virtual PCI masters' actions can be initiated by calling predefined procedures supplying parameters such as PCI command, PCI address and burst length, and the master pool's address to be read or written for the master action. The slave memory pool can be read or written with procedures.

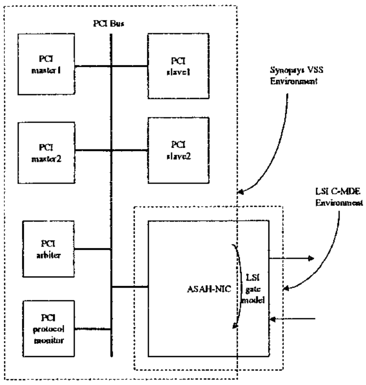


Fig. 2. Testbench of ASAH-NIC

Since the SDH framer was a schematic design, it was omitted in the VHDL simulation but a pseudo VHDL model was substituted for loopback simulation. Later, after the circuit was synthesized, only ASAH-NIC was simulated in LSI's C-MDE environment supplying and verifying vectors and the primary I/O of the chip. Connection table and other table setups are done through a virtual PCI master's action. The segmentation operation can be verified by setting up the tables and triggering the segmentation and re-assembly operation in loopback mode. By observing the PCI master memory read, write done by ASAH-NIC together with the PCI slave write and read done by outside masters, segmentation and reassembly actions can be verified.

3. ASAH-L4 : 622Mbps ATM Layer Processing ASIC

3.1 Functions and Architecture

Fig.3 shows the architecture of ASAH-L4 ASIC. ASAH-L4 is a 622Mbps ATM layer processing ASIC which performs header translation, routing, real time OAM, QOS buffering and ABR processing for both ingress and egress traffics. It has UTOPIA 2 interface and receive ready type switch interface. All the connection information and cell data are stored in three local SRAMs physically separated for RX VC table, TX VC table and the cell buffer memory through three independent SRAM interfaces. Each processing block of the ASIC accesses the local SRAM through memory arbiter and interfaces.

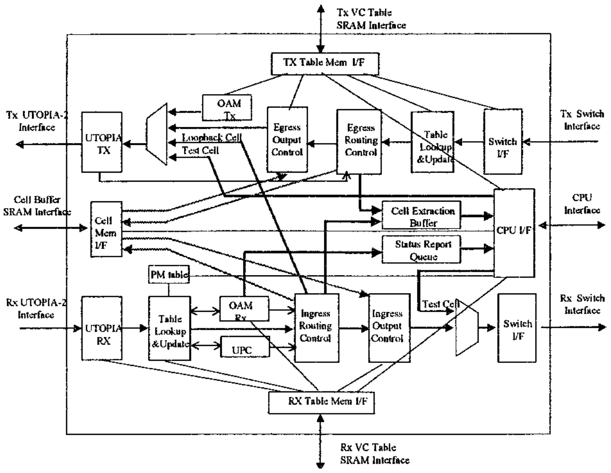


Fig. 3. Architecture of ASAH-L4

In the receive direction, UTOPIA_RX reads cells from a port and looks up the connection table's address while receiving the cell. The VTSB_RX reads the contents of the VC table necessary to perform the enabled function for the connection. Then as soon as the connection data is ready, with the cell still being written or already written out into the UTOPIA_RX's internal FIFO, the UPC and OAM_RX begins action for the cell. After the UPC and OAM action is finished, the QBWB_RX, based on information from all the blocks, reads and routes the cell to the appropriate block or discards it. The QBWB_RX performs cell routing such as looping, CPU extraction, passing to the QBRB_RX(in non QOS mode), or writing to the QOS buffer(in QOS mode) which is implemented using linked list method on the external SRAM. The QBWB_RX also does all the chores for the queue management and scheduling. In QOS mode, when a queue is selected for service, the QBWB_RX commands its service and the QBRB_RX reads the cell from the QOS buffer and sends it to the switch interface through the RX_MUX. In the mean time, header is translated in the QBWB_RX and routing tag is attached in the QBRB_RX. All the processing are pipelined to increase the throughput so that cell moving transaction is performed between local block interfaces.

In the transmit direction, the actions are similar to the receiver. The SWIB_RX receives cells and searches for the VC table address. The VTSB_RX loads and updates the connection data. When the connection data is ready, the cell is routed to appropriate block by the QBWB_TX as in the receiver. In QOS mode, the cells are stored in the external SRAM and read and sent by the QBRB_TX after scheduling. The UTOPIA_TX sends the cell to the destination port with multicast capability. OAM actions are done in OAMB_RX, OAMB_TX with the help of QBWB_RX, QBRB_TX. The details of OAM and UPC will not be explained.

3.2 Simulation

The test bench of ASAH-L4 is composed of the chip itself and several processing entities or behavioral processes in VHDL as shown in Fig. 4. These test bench was duplicated and modified into many test cases to test special feature of ASAH-L4 such as various routing, fault management, performance management, QOS buffering, etc. The SRAM model was written in VHDL for synchronous burst SRAM and later in the simulation process, MICRON's SRAM model was used for simulation.

CPU process initializes the chip and sets up connection data and handles interrupt by periodically monitoring interrupt pin of the chip. The cell generation is done by a separate process which is enabled by the CPU process. In the cell generation process, the scenario of what kind of and how many cells for what connection will be sent is programmed using several flags and counters.

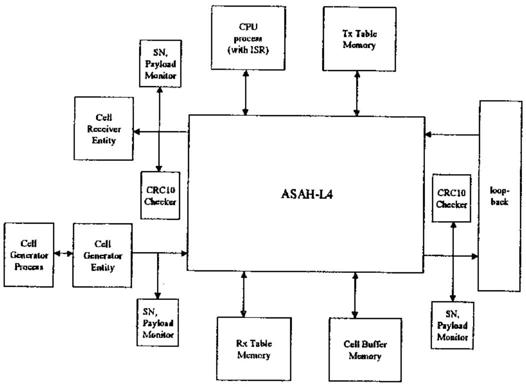


Fig. 4. Test bench of ASAH-L4

The cell generator process uses a procedure for parameter passing and activating the cell_gen entity. This procedure is overloaded (a VHDL term) and can be used to generate simple user cell, various OAM cells(FM and PM) and RM cells with appropriate parameter passing to cell_gen entity. The procedure uses go-done handshake with the cell gen entity which actually generates the cell and passes it to ASAH-L4. The sequence number(SN) is independently generated for separate connections in the cell_gen.

The Switch Loopback entity receives cells from ASAH-L4 and loops them back to ASAH-L4 with internal FIFO and back pressure handling. The Cell Receiver Entity simulates the physical layer chips. It has level-2 mode and direct status sampling mode. To be assured of the proper operation of the chip, it is not enough to see the wave form. It is too time-consuming to look for a very rare but possible error in the cell stream. By putting a sequence number and payload checker at the primary interfaces in the testbench, many inter-block synchronization errors and other problems like linked list handling errors were found out and fixed. If an error is detected an error flag is set and optionally displayed at the terminal.

There is also CRC10 Checker and cell monitor for OAM cells. This enabled the designers to easily detect a CRC10 error(like missing CRC 10 calculation for OAM cell and calculating CRC 10 for the following cell during the loop-back.).

There were many versions of this testbench each for test special features of the function. In addition to some basic testbenches for routing and translation, QOS buffering, others were written and used for OAM(FM and PM), RM, and UTOPIA, random cell sequence, etc.

Since it takes too much time to simulation one second, the internal timing counter is modified to have a scaling function. At the test mode, the timer operation is shrink so that one second pulse is generated 100000 times frequently. By testing all the cases for FM, for VCC, VPC and for segment end point, and connection end points, the FM operations is verified.

4. ASAH-P4 : 622Mbps ATM Physical Layer ASIC

4.1 Functions and Architecture

The architecture of the ASIC is shown in Fig. 5. It has a generic 8 bit CPU interface, 8 bit interface to the lower transceiver and 16 bit UTOPIA interface to the upper ATM layer. The cell processing and the VC processing is done at 77.76MHz speed and most of the STM processing is done at 19.44MHz for de-multiplexed streams except for framing, B1 processing and scrambling or descrambling.

In receiver direction, ASAH-P4 receives STM4-4c frame data from the transceiver and synchronizes to the frame and divides it into four 19M STM stream after descrambling. Then it extracts SOH data and processes them. The pointer value is interpreted and from the VC4-4c data, POH bytes are extracted and processed. The C4-4c data is carried to the cell processor where cells are delineated and written into the FIFO after HEC verification and descrambling and idle cell filtering. The ATM cells are read and sent to the ATM layer through UTOPIA interface.

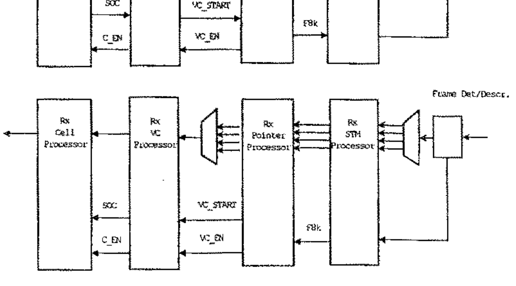
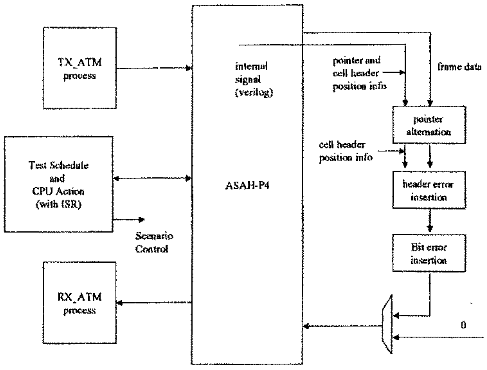


Fig 5. Architecture of ASAH-P4

In the transmit direction, ATM cells are written into the FIFO by the ATM layer through UTOPIA interface and by the ATM layer. Cells are read from the FIFO and after HEC insertion and payload scrambling, and idle cell insertion for rate coupling, C4-4c data stream is formed. Then POH are inserted and passed to the pointer generator where pointer is inserted and the VC4-4c is mapped to the transmit frame. The SOHs are inserted to the AU data and the whole frame is scrambled and transmitted.

4.2 Simulation



TX_ATM process
Test Schedule and CPU Action (with ISR)
Scenario Control
RX_ATM process
internal signal (verilog)
ASAH-P4
pointer and cell header position info
frame data
pointer alternation
cell header position info
header error insertion
Bit error insertion
0

Fig. 6 Test Bench of ASAH-P4 ASIC

The test bench is composed of the ASIC chip, ATM layer processes, CPU processes, pointer alteration module and error injection. The testbench was written in verilog.
TX_ATM and RX_ATM simulate the upper ATM processing sending and receiving ATM cells through UTOPIA interface. The CPU process initializes the chip and manages several flags and counters to govern the simulation scenario. For fault test, the chip is set into the 9x30 test mode in which the entire chip runs at frame format of 9x30 not in normal 9x270 mode. This reduced frame mode is very useful in reducing the function verification time as well as reducing the number of test vectors for fault test. For verifying alarm processing, the loopback data is suppressed and then recovered to see the OAM actions and interrupt processes during the period. The scenario is controlled in such a way that scaling frame format makes the interval scaled accordingly.
For verification of error monitoring for frame data error, corresponding error pattern is injected in the data and to inject cell header error, the C4-4c payload region and cell header location information is derived from transmitter's internal signals. This is possible by the verilog's feature that enables to access signals across hierarchies.

Another important function to verify is the receiver's pointer interpretation. To see if it works fine, the pointer values should be changed in the loopback path and the VC data phase should also be changed in the received frame. In the simulation of pointer interpreter's action, the pointer value which is initially 1 is incremented and decremented and set to 3 and abruptly changes back to 1. The VC data phase is changed accordingly as specification. This was done by the pointer manipulation circuit (will not be explained here).

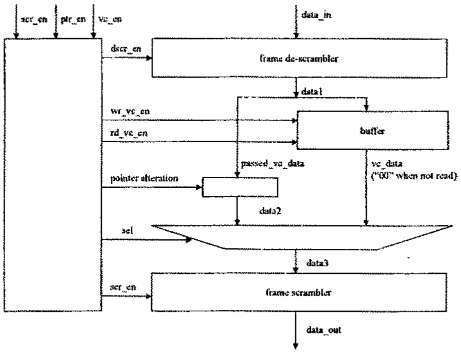


Fig. 7. Pointer Manipulator Circuit

Since there are many clocks in this circuit, in gate simulation, the outer clock phase and frequency was adjusted to avoid timing violation between clocks. Furthermore, all clocks begin some time after the reset to avoid the recovery violation and several clocks were given during the reset to let any un-initialized flip-flop to get out of unknown values.

5. Conclusion

The simulation techniques for three ASIC developments were explained. All these ASICs were successfully developed and have been or are being used in our system development. These techniques can be applied to other types of ASIC developments.

6. References

1. Chan Kim et al, "An Implementation of a 622Mbps ATM Physical Layer ASIC," ICT97, Melbourne, Apr. 1997.

2. Chan Kim et al, "Design and Implementation of a High-Speed ATM Host Interface Controller," ICOIN-12, Tokyo, Jan.98

3. Chan Kim et al, "Design and Implementation of an ATM Segmentation Engine with PCI Interface," ISCAS'98, Monterey, May. 1998.

4. Chan Kim et al, "Implementation of a QOS buffering in ASAH-L4 ASIC with Weighted Round-Robin and Maximum Delay Threshold," ICACT'99, Muju, Feb. 1999.

5. Chan Kim et. al, "Implementation of OAM functions in a 622Mbps ATM layer processing ASIC," BSS'99. Kingston, Canada, Jun 1-3, 1999