Asian Journal of Information Technology 5 (12): 1458-1463, 2006
© Medwell Journals, 2006

Analysis of Reconfigurable Architecture for Delay Sensitive Voice Streams over IP Networks

[1]P.K. Jawahar and [2]V. Vaidehi
[1]Research Scholar, Anna University, Chennai, 25, India
[2]Department of Electronics Engineering., MIT, Anna University, Chennai, 44, India

Abstract: IP networks are not suitable for carrying delay sensitive real time data like voice, video as they offer only best effort services. There is no guarantee that real time packets will be given preference over other non real packets and they will not be dropped during congestion. Thus, Quality of Service (QoS) for these real time multimedia streams is highly deteriorated in the network as it has no control mechanism to improve it. Only in the receiver side, various algorithms can be applied to improve quality of service for real time data transmission as the receiver can not request the sender to retransmit a missing or corrupted packet.. An adaptive reconfigurable architecture to cater the variations in the quality of service is discussed in this study. A controller acts as a static module and decides the dynamic module that has to be loaded on run time to improve the degraded QoS parameter instantaneously, making the architecture a dynamic one. Though FPGAs are power hungry devices, only few modules are loaded into FPGA on runtime reducing the overall power consumption as the new Virtex family devices consume less power. Using Xilinx Virtex devices, the implementation of partial reconfiguration of modules pertaining to QoS enhancement for VoIP networks is discussed in detail in this study.

Key words: Congestion, deteriorated, consumption, FPGA

INTRODUCTION

Real time data are always delay sensitive, as the transmission of real time data over IP networks with more delay lacks the originality of the data and leads to poor quality at the receiver. Since network is not under the control of users, transmission and reception of real time multimedia information like voice, video etc., always encounter packet loss, delay , jitter degrading the quality of service.

Some measures are to be taken by the receiver to compensate these QoS parameters. Packet loss can be compensated by loss concealment algorithms to replace the lost packets so that receiver cannot perceive the packet loss. Delay causes echo generation which can be cancelled using adaptive echo cancellation algorithms. Jitter can be compensated using buffering techniques[1]. Other parameters like bandwidth control require similar algorithm to improve QoS[1].

Algorithms related to QoS parameters can be implemented to the received real time packets in the IP networks. It is impractical that all the algorithms can be applied to a packet Simultaneously and only one algorithm will be active at a time. Hence, while implementing these algorithms on FPGA, all the modules need not be loaded during run time. Based on the quality of packet, the manipulation that has to be done on the packet may vary.

The drawback of implementing these algorithms as hardware solutions is they are not flexible but reliable and efficient. Dynamic reconfigurable systems are one in which part of the device can be reconfigured while the rest continues its normal operation without physically resetting the system. If some bits of the new frame do not change in comparison to the older one, it is guaranteed that there will be no glitches on this bits during the reconfiguration. Table 1 shows the reconfiguration speed of various Virtex FPGAs. Reconfiguration of modules pertaining to VoIP algorithms should take place quickly in the order of milliseconds (max.125 ms) and also this table proves that with Virtex devices, reconfiguration is possible with the minimum speed is 16 ms which is well below 125 ms.

Dynamic reconfiguration makes the hardware very flexible as the FPGA has the capability of modifying its structure on runtime. There are two ways of implementing dynamic reconfiguration namely modular based and difference based reconfiguration. In module based reconfiguration , the entire project is divided into multiple independent modules based on their functionality and

Corresponding Author: P.K. Jawahar, Research Scholar, Anna University, Chennai-25, India

Table 1: Reconfiguration speed

		Min. time to reconfigure (µs)		
Device	Bits per Frame	1 Frame	1 LUT	1CLB Column
XCV50	384	290	1730	50
XCV300	672	430	2950	8330
XCV800	1088	65	470	13425
XCV1000	1248	70	55	15380

depending on the packet status, any one of the modules will be active rather activating all at a time which are not required. The active modules will be connected to static modules using bus macros. Modular design is a flow which allows the user to construct a FPGA layout from from partial layouts[2].

In difference based reconfiguration, a small change is made in the design using FPGA EDITOR. Based on the difference between two designs, a bit stream is generated. Switching between one module to another module is done by implementing the differences in the bit streams which can be done quickly[2].

Many designers prefer module based partial reconfiguration as the implementation is very easy and the same is discussed here for enhancement of QoS in VoIP networks.

Dynamic reconfigurable architecture for voip network: To apply the various algorithms for VoIP environment and considering the existing architecture, the computation structure has been divided into two basic groups: Filtering (for echo cancellation) and packet loss concealment.

Echo cancellation is an important design consideration of VoIP systems, as echo is annoying and objectionable with the increasing round-trip delay and amplitude in particular for delays of more than 20 ms Fig. 1. The acceptable delay, for bi-directional real-time streaming applications, is usually limited by 50 ms[3]. Many versions of LMS algorithms are suggested for echo cancellation and Dichotomous Coordinate Descend Algorithm is comfortable for VLSI implementation[1].

Packet loss is another important problem with regard to the deployment of Internet real-time services. Loss concealment algorithms typically add a delay of at least that corresponding to one packet length, because the algorithm is triggered only when a missing packet has been detected.

Silence substitution is the simple method of concealing single packet loss. Linear predictive algorithms are used to conceal single packet as well as burst losses[1].

In order to improve the overall system performance, the major principles like:

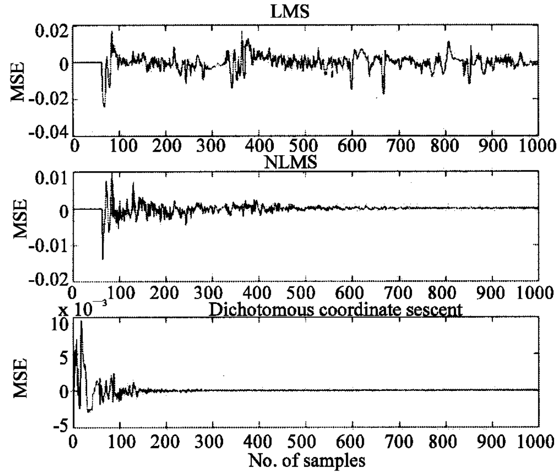


Fig. 1: LMS, NLMS, DCD implementation of echo cancellation

- Memory must be present to support reconfigurable modules to operate quickly
- Bus macros and bus widths must be wide for quick routing and
- there should be more than one to investigate the parallelism inherent in the algorithms[4]. The architecture Fig. 2 consists of mainly a) Controller
- Configuration memory
- Programmable I/O d) Switching matrix with memory
- Reconfigurable Module. The function of controller is to load the memory with configuration words and to generate the address sequence needed by the targeted algorithms.

Programmable I/O or Ethernet interface is used to interface IP network with the controller and to receive voice packets. Configuration memory are stored with configuration words meant for reconfigurable modules[4].

The functions of Controller, Programmable I/O and Switching Matrix can be integrated into a single block and can be implemented as a Micro Blaze 32 bit soft core processor in Spartan 3 devices or hardcore PowerPC processor in Virtex IV devices.

Modular based partial reconfiguration: The advantages of reconfiguration are versatility, upgradeability and area savings. In Xilinx Spartan/Virtex devices, partial reconfiguration is possible and in particular, Virtex series devices are well suited for applications which support many different algorithms. Spartan devices have limitations in allocating the area for reconfigurable module and glitches may occur during reconfiguration process[5].

In Fig. 3, the design flow for implementing partial reconfiguration for Xilinx FPGA devices is discussed.

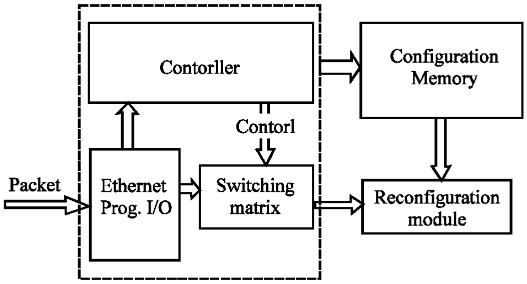


Fig. 2: Architecture

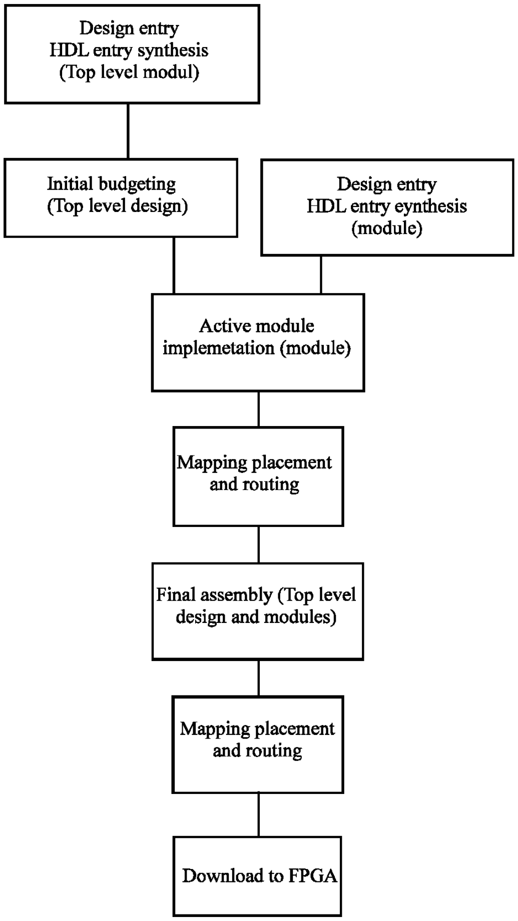

Design entry HDL entry synthesis (Top level modul)
Initial budgeting (Top level design)
Design entry HDL entry eynthesis (module)
Active module implemetation (module)
Mapping placement and routing
Final assembly (Top level design and modules)
Mapping placement and routing
Download to FPGA

Fig. 3: Design flow

Figure 4 shows the sequence to be followed to apply partial dynamic reconfiguration[6]. Using partial reconfiguration guidelines, develop and synthesize HDL code as a part of design entry. In the initial Budgeting phase, for the top level design and each modules involved, design the floor plan, User Constraints File (UCF) and timing constraints. Then run the NGDBUILD,MAP,PAR tools for active implementation of each module of reconfigurable module. It is then followed by assembling phase in which every possible combination of fixed and reconfigurable module device configurations are to be assembled and verify the design using timing analysis or functional simulation[7].

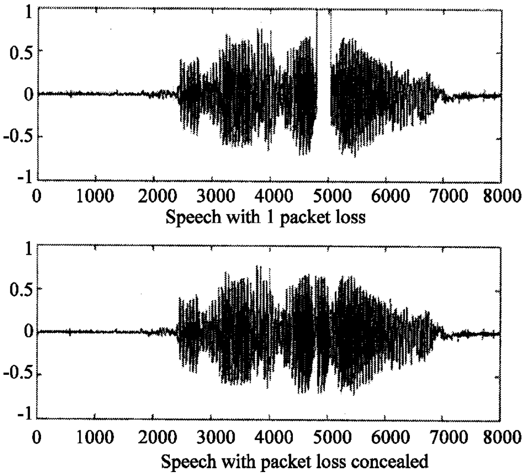


Fig. 4: Packet Loss Concealment Using Linear Predictive coding

Using FPGA_Editor, routing conflicts across module boundaries are verified. A sample lay out of a system with two reconfigurable modules and their allocation obeying the guidelines of partial reconfiguration is shown in Fig. 2. Some of the important guidelines are

- The Clocking Logic is always separate from reconfigurable module
- Reconfigurable module height is always the full height of the device
- Horizontal placement of the module must be on the four slice boundary and leftmost placement must be in the order of four
- IOBs above the top edge and bottom edge of a reconfigurable module are part of the specific reconfigurable module's resource alone.
- The boundary of a reconfigurable module cannot be changed
- Communication between fixed and dynamic modules can be through special bus macros[7] Fig. 5.

Virtex II Devices provide hard macros whereas Virtex IV and Spartan devices need specific bus macros to be developed by the designer. A sample bus macro in verilog code for Virtex IV is listed below.

```
`define SIZE 15
module bm16 (LI, LT, RI, RT, O);
  input  [`SIZE:0] LI, LT, RI, RT;
  output [`SIZE:0] O;

bm_4b_v2     bus1     (.LI(LI[3:0]),     .LT(LT[3:0]),
.O(O[3:0]),.RI(RI[3:0]), .RT(RT[3:0]) );
```

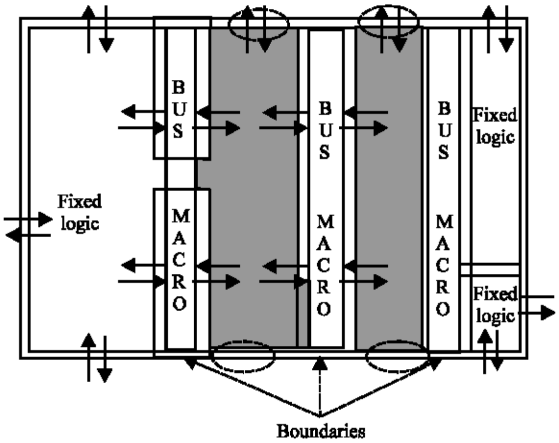


Fig. 5: Allocation of reconfigurable modules

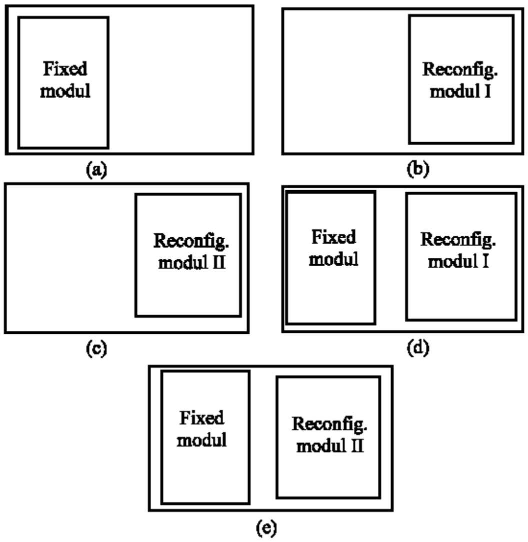


Fig. 6: Implementation of fixed and reconfigurable modules

```
bm_4b_v2 bus2 (.LI(LI[7:4]), .LT(LT[7:4]), .O(O[7:4]),.RI(RI[7:4]), .RT(RT[7:4]) );
bm_4b_v2 bus3 (.LI(LI[11:8]), .LT(LT[11:8]), .O(O[11:8]),.RI(RI[11:8]), .RT(RT[11:8]) );
bm_4b_v2 bus4 (.LI(LI[15:12]), .LT(LT[15:12]), .O(O[15:12]),.RI(RI[15:12]), .RT(RT[15:12]) );
endmodule
module bm_4b_v2 (LI, LT, RI, RT, O);//synthesis syn_black_box
input [3:0] LI, LT, RI, RT;
output [3:0] O;
endmodule
```

Implementation of dynamic reconfigurable architecture for voip networks: The design flow as explained in this study is applied for the implementation of the partial dynamic reconfiguration of algorithms. For echo cancellation, a 16 tap adaptive FIR filter was developed using verilog RTL code and tested as an individual module with XC3s500e device of Xilinx Spartan3e starter kit using Xilinx ISE 8.2i. If Virtex 4 is used, the built in DSP slices and MAC units will be comfortable for this testing. Packet loss concealment algorithm was tested for single packet loss using past packet substitution[5]. But it needs further testing with Linear Predictive methods.

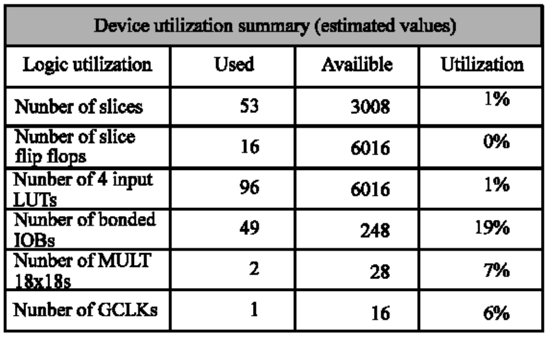
Fig. 7: Micro blaze layout

Fig. 8: Echo cancellation layout

Using PACE, Micro Blaze modules namely, controller, Ethernet MAC, GPI/O, DDRAM are oriented on one area and the user constraint file is created during design entry and initial budgeting. MicroBlaze acts as a top module and the echo cancellation module and packet loss concealment modules act as reconfigurable modules which are assigned same area using PACE. The flow and the implementation is shown in Fig. 6a-e.

It is noted that the directory structure should be maintained successful partial reconfiguration. Figure 7 and 8 show the FPGA layout of the Microblaze and Echo

Fig. 9: Block diagram of microblaze



Device utilization summary (estimated values)			
Logic utilization	Used	Availible	Utilization
Nunber of slices	53	3008	1%
Nunber of slice flip flops	16	6016	0%
Nunber of 4 input LUTs	96	6016	1%
Nunber of bonded IOBs	49	248	19%
Nunber of MULT 18x18s	2	28	7%
Nunber of GCLKs	1	16	6%

Fig. 10: Device utilization summary for adaptive filter

cancellation modules. For echo cancellation, adaptive LMS algorithm is applied to find the error coefficient. Sysgen 8.2i with Matlab was used for this simulation and synthesis. For packet loss, each packet is buffered and with the help of RTP format, the loss of packet is identified from the sequence number and the previous packet is substituted in the present packet position which adds some delay[9].

Micro Blaze was developed using Xilinx EDK 8.2i for modular design flow. The block diagram developed from EDK is shown in Fig. 9.

Figure 10 shows the device utilization summary for adaptive filter applied for echo cancellation tested in Virtex 4 protoboard. As Virtex 4 family has built in DSP slices and MAC units, implementation of FIR filters are possible[6].

CONCLUSION

Transmission of real time data over IP networks and maintaining best quality is always a difficult task. Many software solutions were developed and in this study a hardware solution is proposed for improving quality of service for real time data. Developing partial reconfigurable architecture for any application is really challenging and in particular for testing real time voice packets over IP networks requires more algorithms and more tools for testing. Xilinx provides wide range of tools for dynamic partial reconfiguration. Partial reconfiguration is tested and verified with EDK and ISE tools. This can be applied for real time video and any real time multimedia which increases the complexity of the architecture further.

REFERENCES

1. Jawahar, P.K. and V. Vaidehi, 2005. Enhancement of Quality of Service in VoIP networks using Partial Runtime Reconfigurable Architecture (PaRuRa), International Conference on Intelligence Systems, Kuala Lumpur.
2. Sedcole, P. *et al.*, 2006. Modular dynamic reconfiguration in Virtex FPGAs, IEE Proceedings on computer Digital Tech., pp: 153-3.
3. Xiu zhong Chen and Chufeng Wang *et al.*, 2003. Survey on QoS Management of VoIP, Proceedings of ICCNMC.
4. Ressell Tessier and Wayne Burleson, 2001. Reconfigurable Computing for Digital Signal Processing: A survey, in J. VLSI Signal Processing, 28: 7-27.
5. Haase, A. and C. Kretzschmar *et al.*, 2000. Design of Reed Solomon Decoder using Partial Dynamic reconfiguration of Xilinx Virtex FPGAs-a Case study.
6. Xilinx Inc, 2000. Virtex FPGA series Configuration Architecture user Guide, Application Note XAPP151.
7. Xilinx Inc, 2004. Two flows for partial reconfiguration: Module Based or Difference Based Application Note XAPP290.
8. Florent Berthelot and Fabienne Nouvel, 2006. Partial and Dynamic reconfiguration of FPGAs: A top down design methodology for and automatic implementation, Proceedings of 2006 emerging VLSI technologies and Architectures, ISVLSI.
9. Uyless, 2000. Black Voice over IP Prentice Hall.
10. www.Xilinx.com