ANT-on-YARDS: FPGA/MPU Hybrid Architecture for Telecommunication Data Processing

Akihiro Tsutsui, *Member, IEEE,* and Toshiaki Miyazaki, *Member, IEEE*

Abstract— **This paper presents a novel system architecture that combines tightly coupled field programmable gate arrays (FPGA's) and a microprocessing unit (MPU) that we have developed. This system architecture comprises three main programmable devices which yield high flexibility. These devices are a reduced instruction set computer (RISC)-type MPU with memories, programmable interconnection devices, and FPGA's. This system supports various styles of coupling between the FPGA's and the MPU which makes several data processing operations more effective. Furthermore, we indicate the most suitable applications for the system. They are telecommunication data processes involving complex protocol operations and network control algorithms. In this paper, two applications of the system are given. One is for operation, administration, and management (OAM) cell processing on an asynchronous transfer mode (ATM) network. The other is a dynamic remote reconfiguration protocol that enables the functions of the transport data processing system to be updated or changed on-line.**

Index Terms— **Codesign, field programmable gate array (FPGA), microprocessor, telecommunication, yet another redefinable system (YARDS).**

I. Introduction

CONVENTIONAL implementations of telecommunication systems use fixed hardware since importance was placed on high-speed transport data processing rather than flexibility. However, today's enthusiasm for the inter-networking trend is forcing network systems to support a wide variety of communication protocols, even those are not yet fully developed. Therefore, future network systems must offer not only high performance but also high flexibility [1].

The asynchronous transfer mode (ATM) technique is one of the solutions to this problem [2]. It is suitable for multimedia data communication because it is based on the concept of the "virtual path (VP)" which can handle several data bandwidths equally and flexibly. In addition, because the minimum transfer unit (the cell) is quite small, hardware implementation is so simple that it is easy to construct high-throughput network systems [3]. With the progress of optical communication technologies, the ATM network has the potential to realize high performance and flexible telecommunication services.

When providing high quality multimedia services, it is indispensable to achieve a high performance and flexible network control and management [1]. Up to now, there has been no other choice but to realize these as software components because they are complex and require frequent updating. In the near future, however, some of the operations might be implemented as hardware to achieve adequate data manipulation rates. Thus, lower layer telecommunication protocols will require more flexible implementation and higher ones will be forced to achieve higher performance. Therefore, an efficient technique for fusing hardware and software is becoming indispensable for building the next generation network systems. Unfortunately, no existing design technology in the telecommunication field can achieve this fusion.

Given this situation, we considered the field programmable gate array (FPGA) to be the key to future telecommunication systems and have developed an original FPGA especially designed for high-speed telecommunication data processing [4], [5]. Using this device, we also constructed a reconfigurable signal-transport system dedicated to real-time emulation of transport processing circuits [6]. This system is useful in implementing lower layer transport operations. However, higher layer protocols such as network management applications mentioned above include excessively complex logic operations which are not suitable for FPGA's. Microprocessing units (MPU's) and program logic implemented as software appear indispensable to meet these goals.

There are some reports on hybrid systems which consist of tightly coupled FPGA's and MPU's [7], [8]. In these systems, both FPGA's and MPU's cooperate with each other to execute operations. In general, however, it is difficult to find an actual target application that maximizes the performance of the system, because it is difficult to divide a given problem into two implementation styles: hardware and software. Based on our experience, we discovered an effective application for the tightly coupled system in the field of telecommunications. The lower layer telecommunication protocols are suitable for hardware implementation while the higher layer protocols should be realized as software. In the field, both high throughput to support real-time data transmission and flexibility to support various protocols are required.

Thus, we developed a novel architecture for a system comprising tightly coupled FPGA's and MPU's. This architecture is suitable for implementing flexible and real-time transport data processing operations involving complex protocol operations.

This paper presents the new architecture of the hybrid system and its advantages. General ideas for achieving efficient coupling of software (MPU) and hardware (FPGA) are also mentioned. Moreover, application models of the hybrid system in the telecommunications field and experimental implementations for a few instances are shown.

Manuscript received April 4, 1997; revised November 1, 1997.

The authors are with NTT Optical Network Systems Laboratories, Kanagawa 239-0847 Japan.

Publisher Item Identifier S 1063-8210(98)02955-2.

1063–8210/98$10.00 © 1998 IEEE

IPR2018-01694
EXHIBIT
2060

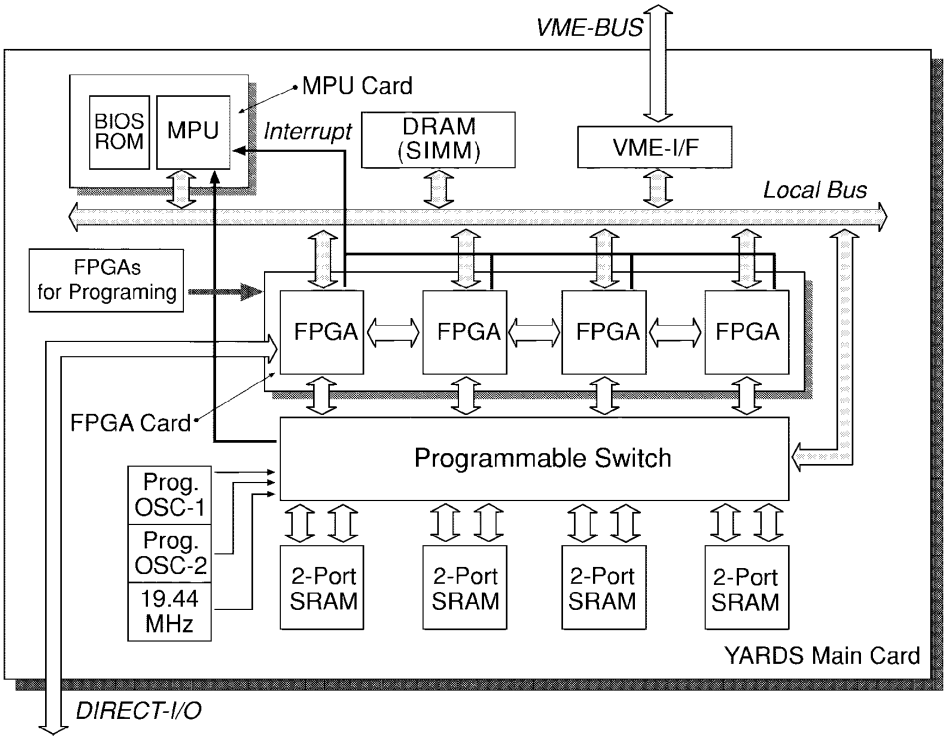

Fig. 1. Basic architecture of YARDS.

II. Related Work

Some systems consisting of MPU's and FPGA's have been proposed [7], [8]. In those hybrid systems, some particular operations suitable to hardware implementation are performed by specially designed logic circuits on the FPGA's while the MPU works in conjunction with them.

They communicate with each other using the MPU's local bus. Thus, the FPGA's are treated as coprocessors or special peripheral devices of the MPU. Moreover, specially designed FPGA's dedicated to coworking with the MPU were developed [9].

Most of the target applications for these systems involve numerical calculation or digital signal processing such as video-CODEC. However, it seems difficult to find an effective application which derives the maximum performance from the system. One reason for this problem is that it is difficult to divide the target application into hardware and software components. Moreover, data transmission between the MPU and FPGA becomes a bottleneck which prevents the system from attaining a high throughput.

III. YARDS and ANT

A. Overview of YARDS

We developed a new system architecture comprising tightly coupled FPGA's and an MPU named "yet another redefinable system" (YARDS). Fig. 1 shows the basic architecture of YARDS. The main parts are an MPU, an FPGA array, programmable switching devices, and two-port SRAM's. Fig. 2 shows the system overview of YARDS. It consists of three cards: the main card, the MPU card, and the FPGA card. The main card contains devices for interconnecting the FPGA part and the MPU part of the system. The MPU card consists of a reduced instruction set computer (RISC) MPU with a BIOS-ROM. The FPGA card features a multiple FPGA array.

YARDS also has two external interfaces: a VME-Bus I/F and a direct I/O channel derived from the FPGA card. Using the VME-Bus, this system can communicate with other YARDS or host computer systems. We utilize it for controlling and monitoring the system. The direct I/O channel provides a direct data communication between other devices and the FPGA card. The front view of YARDS is shown in Fig. 3 and the back is shown in Fig. 4.

B. YARDS Main Card

The main card contains interconnection elements for the FPGA's and the MPU. They comprise field programmable switching devices (I-Cube) that support connections among its pins such as unidirectional or a bus. The local-bus signals and a few interrupt pins of the MPU and most I/O pins of the FPGA's are connected directly to these switching devices.

The two-port SRAM's on the main card have various uses. These are connected directly to the switching devices. By suitably configuring the connection pattern of the switches, those SRAM's can be used as shared memories or buffers by

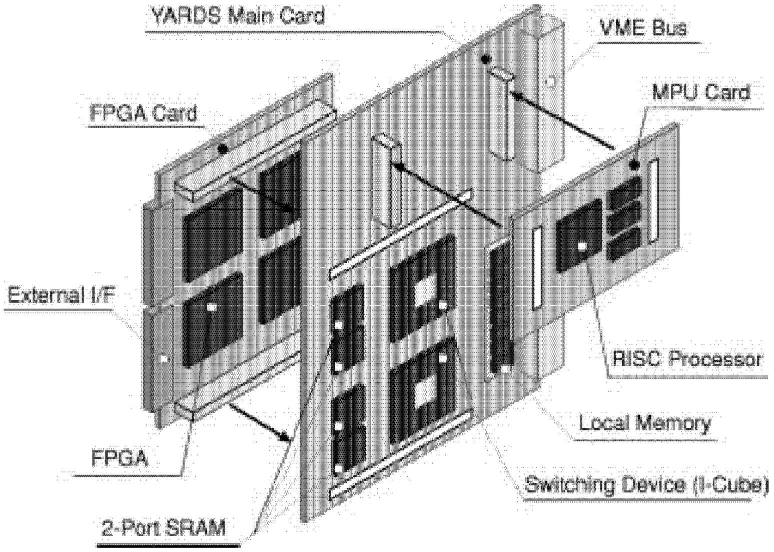

Fig. 2. System overview of YARDS.

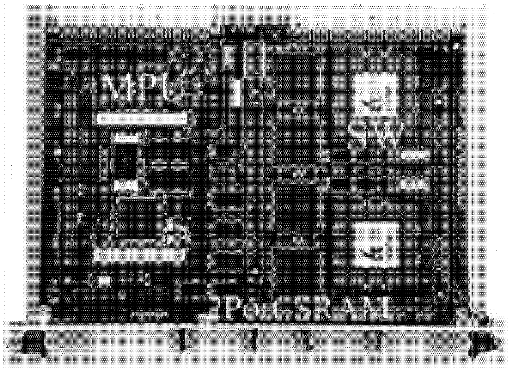

Fig. 3. Picture of YARDS (front).

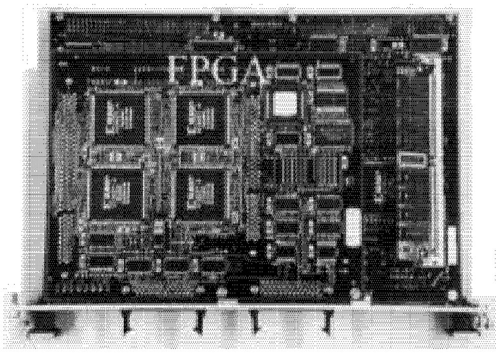

Fig. 4. Picture of YARDS (back).

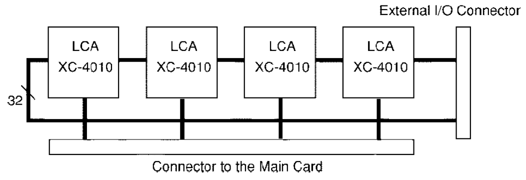

Fig. 5. FPGA card (using LCA).

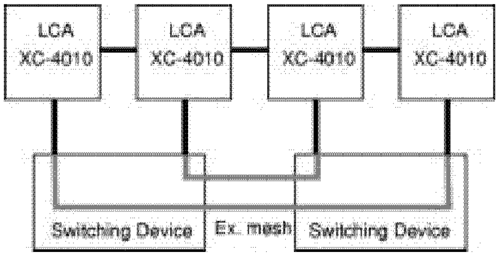

Fig. 6. Example of a link topology (using LCA).

the FPGA's and the MPU. Thus, using these programmable switching devices and two-port SRAM's, various connection patterns can be established between the FPGA's and the MPU. The programming and control logic of the system are implemented using an FPGA on the main card. Thus the system can support several kinds of FPGA's and MPU's by reprogramming the FPGA.

YARDS has three clock generators in addition to the MPU's base clock (20 MHz). One of these provides a fixed clock speed of 19.44 MHz. This is the base clock for a typical telecommunication circuit which handles a 155 Mb/s synchronous digital hierarchy (SDH) interface. Others are programmable clock generators.

C. FPGA Card and MPU Card

We wanted to try various types of FPGA or MPU devices because YARDS is an experimental system. Thus the FPGA

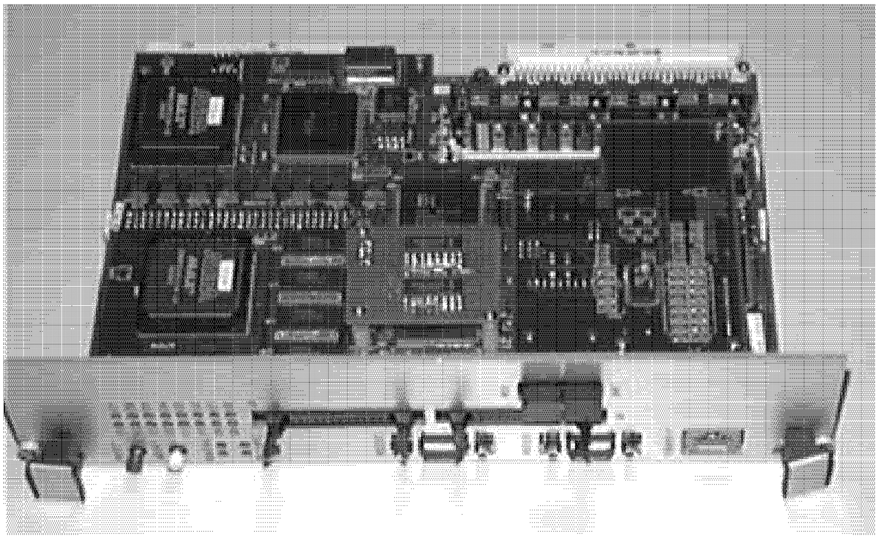

Fig. 7. Picture of ANT.

part and the MPU part of the system were designed as separate daughter cards. They can be replaced with cards containing other devices. As for the MPU part, we have little choice owing to the constraint of local bus compatibility. However, as for the FPGA part, there are many candidates. Actually, we have developed two types of FPGA cards which consist of different FPGA devices. We adopted Xilinx LCA (XC4010) and ALTERA MAX (MAX-9000). Fig. 5 shows a block diagram of the LCA array card. It consists of four devices (LCA) that are connected directly to each other.

Their link topology on the card is a cascade. However, most of the I/O pins of the FPGA's can be connected to the programmable switching devices on the main card and some of them are connected to external I/O connectors. Using the switches, the topology of the links among the FPGA's can be varied as shown in Fig. 6.

Generally speaking, different kinds of FPGA's require different configuration methods. Therefore, the configuration and logic control of an FPGA must be changed for each device. This difference in the logic configuration can be absorbed by the programming control FPGA's (LCA XC4010 and XC3030) mounted on the main card. All of the pins for configuration and control of the FPGA's on the FPGA card are connected to this programming control FPGA via the switches. By reprogramming to achieve a suitable configuration logic circuit on the programming control FPGA, we can handle different kinds of FPGA's.

The MPU card comprises the MPU and a BIOS-ROM. We adopted a 32-bit RISC microprocessor (Hyperstone E-1) which has a simple architecture and is easy to use. All local bus signal lines are connected to the main card bus via the connectors. This RISC microprocessor has five different interrupt signals. A few interrupt signal pins are connected to the FPGA card directly, the rest pins are connected to switching devices on the main card via the connectors.

D. ANT Architecture

Our main target application for YARDS is an intelligent network node system for the ATM network. For this, we developed an ATM Network Termination (ANT) card which acts as the network interface for YARDS. It is designed as a single board computer system with a 155 Mb/s ATM network interface daughter card. Fig. 7 is a photograph of an ANT card. This card has a direct I/O channel which can be connected to the FPGA's on YARDS. Coupling the card and YARDS yields an intelligent network interface for the single-board computer as shown in Fig. 8.

Fig. 9 shows the block diagram of ANT. The single board computer part of ANT is designed as a standard VME-Bus card. It consists of a RISC-type microprocessor (MIPS R3000), memories, and some peripheral devices. A serial communication interface and an Ethernet interface are also implemented on the card. This computer system is managed by the real-time operating system called VxWorks [10] and can stand alone.

The ATM interface daughter card comprises a 155 Mb/s optical interface, ATM physical layer processor (PHY Device), ATM adaptation layer processor (SAR Device), and SRAM memory devices. Compared to ordinary interface cards, our daughter card has a special data channel which can be linked to the direct I/O port of YARDS. This channel is connected to the ATM physical layer processor directly. Thus, raw ATM cells can be sent and received by YARDS.

The logging memory placed between the ANT main card and the ATM interface daughter card provides the communication channel between the PHY device of ANT and the RISC microprocessor. Both can send or receive raw ATM cells via this logging memory. This feature is useful for tapping and monitoring the ATM link connected to ANT and processing status in YARDS.

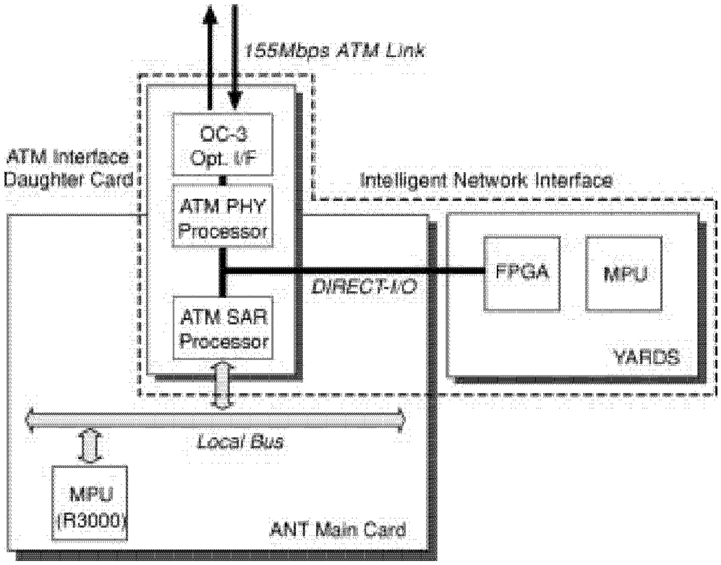

Fig. 8. Model of intelligent network interface.

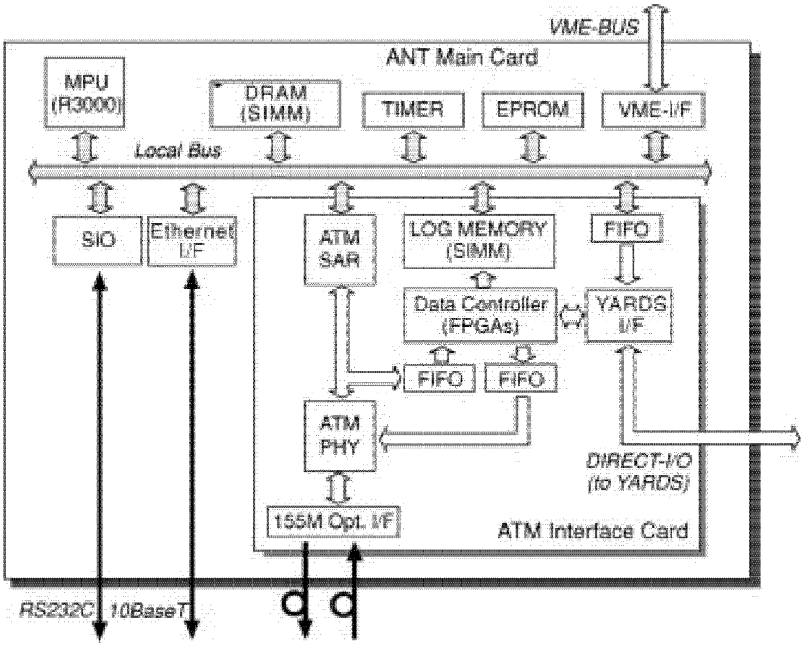

Fig. 9. Block diagram of ANT.

IV. Advantages

A. Flexible Interconnection Between MPU and FPGA

Most conventional hybrid systems employ a bus architecture [8]. They treat the FPGA's as a coprocessors or I/O devices as shown in Fig. 10. This interconnection style couples these devices tightly. However, there are some problems that prevent harmonious cooperation.

For example, immediate communication from an FPGA to the MPU is difficult to implement using this interconnection style. In addition, local bus congestion caused by the communication between an MPU and FPGA's must be considered. Therefore, using only the bus architecture restricts the applications of the system.

YARDS supports three different styles of connection between FPGA's and MPU: a bus, a direct interrupt, and a

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**