# Power Upgrading of Nuclear Power Plant by Improving Shutdown System No.1 with FPGA Technology

**Jingke She**
**Ph. D Candidate**
**Department of Electrical and Computer Engineering**
**The University of Western Ontario**
**London ON N6A 5B9**

## Abstract

*Power upgrading of existing nuclear power plants has been a hot topic among the world. There are now many different methods to accomplish this objective. Making the response time of SDS1 shorter is one of them. Here we propose an FPGA realization of the control logic implementation of SDS1, which may shorten the trip parameter processing time and, consequently, enhance the operation power level without disturbing the safety margin.*

## I. Introduction

The research and improvement in CANDU NPP (Nuclear Power Plants) never stops. The refurbishment of old CANDU NPPs also has been carried on since late 1990s, in order to extend their lives for many more years [1]. The renovation of new type and the refurbishment of old ones provide many new ideas and, therefore, many research topics and opportunities. Improving the response speed of Shutdown System No.1 (SDS1) is one of them. In this paper, we propose a new way to implement the SDS1 control logic, which utilizes the advantages of programmable logic devices to both improve the response speed and give explicit qualifications for SDS1.

After a brief survey and introduction of SDS1, the analysis of the control logic will be shown, based on which the digital system model will be built. This model is proposed to be realized by using FPGA, a well developed programmable logic device; thus the introduction and discussion of FPGA will be mentioned right after building the digital system model. Since the requirements are high and strict for equipments used in nuclear power plants, feasibility of using FPGA technology is then necessary. This part will be discussed in the last section.

# II. Shutdown System No.1 Control Logic

As a safety system in a nuclear power plant, SDS1 must qualify for very strict criteria, which means it should:

- be reliable enough,
- have online testing ability,
- have enough redundancy,
- and perform its function on time whenever necessary [2], [3].

To obtain these qualifications, SDS1 control was designed to be a triplicate, relay logic applied system.

There are totally three trip channels (D, E, and F), having completely independent and physically separated power supplies, trip parameter sensors, instrumentation trip logic and annunciation. The overview block diagram is shown in **Fig. 2.1**

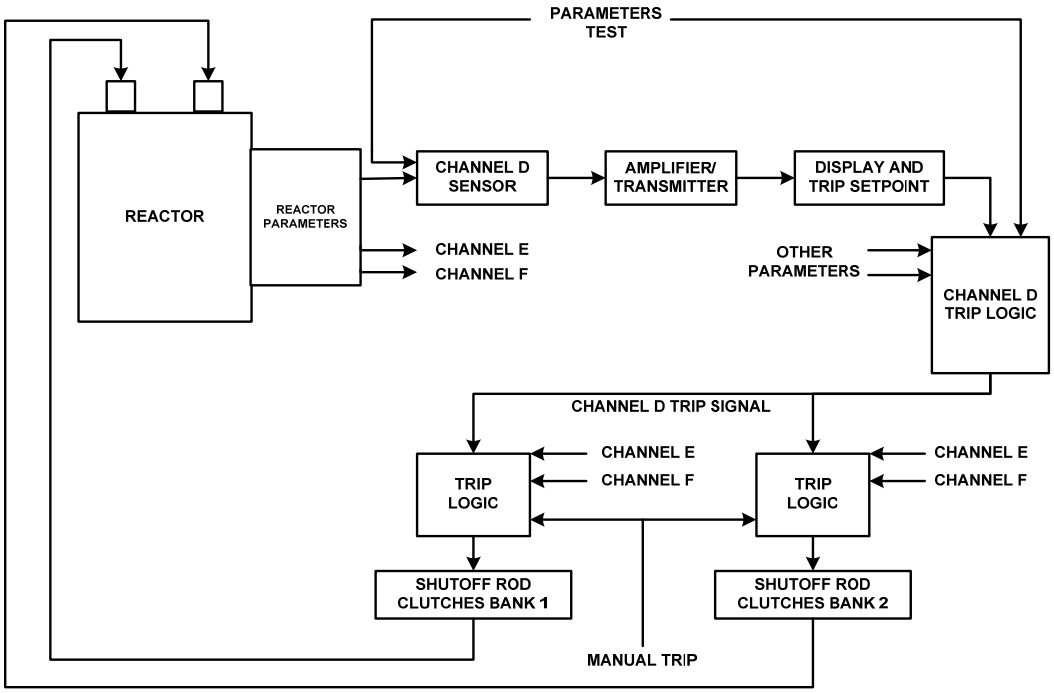


**Fig. 2.1 Block Diagram of SDS1 Control**

There are totally ten parameters selected to be the trip parameters. They are selected such that there are adequate measurements for all process failures identified. They mainly concern about the pressure/flow of the heat transport loops and the operating power. All these ten parameters are listed in reference [4] along with the setpoints, detector types, and protective coverage (The $11^{th}$ and $12^{th}$ parameters are additional, which only care about the manual trip and the start-up).

For detailed dataflow in the control logic, please refer to **Fig. 2.2**.

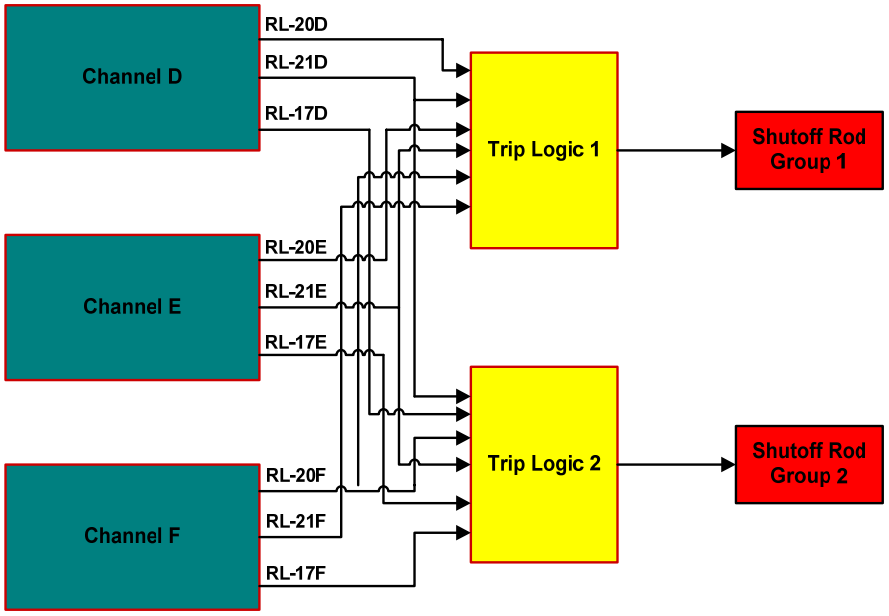


**Fig. 2.2 Dataflow in SDS1 Control**

With the development of computer technology, many control systems in NPP have been computerized, including SDS1 [5]. The signals come from sensors (through amplifiers) have been digitalized before entering the channel trip logic. Inside the channel trip logic, PDCs (programmable digital comparators) are applied to implement trip parameters that require extensive conditioning or setpoints to be functions of reactor power and/or pump configuration [6]. Each channel has two PDCs and seven of ten parameters are distributed between these two (the other three parameters are related to neutron flux and don't go through a PDC). The diagram of channel trip logic is illustrated in **Fig. 2.3**

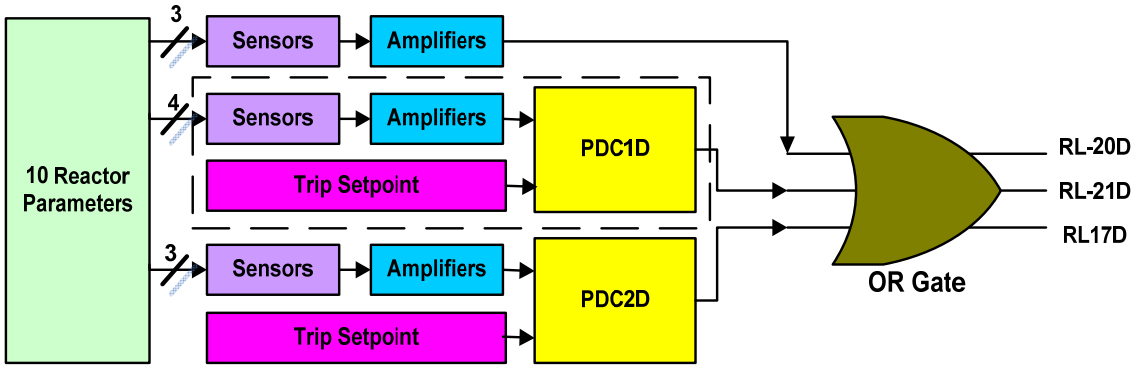


**Fig. 2.3 SDS1 Control Channel (Channel D)**

PDCs used here are actually microprocessor based control components. As we know, a microprocessor needs an instruction-set to support its operation. It needs to refer to the instruction-set frequently in order to realize the control objectives. This working scheme, which takes both time and complexity, can be improved to have shorter processing time and simpler complexity such that it will enhance the response speed of SDS1 and also make a more explicit way to apply for CNSC license.

# III. Digital System Model for SDS1 Control Logic

Considering the microprocessor based control can be improved by direct hardware implementation, we then propose that logic devices (e.g. FPGA) can be used to finish this task. This is related to digital system design and implementation work.

First the PDC logic will be divided into several logic function blocks (each for one parameter) since one PDC deals with several trip parameters. These function blocks are different with each other because every parameter works under different conditions [4]. Generally speaking, the structure of such a logic function is supposed to be like what is shown below in **Figure. 3.1:**

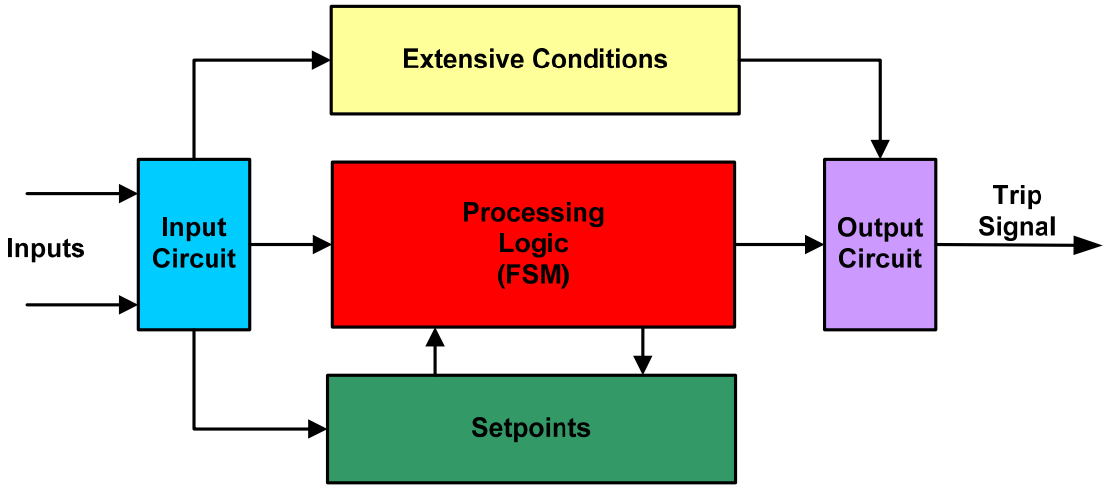


**Fig. 3.1 Structure of a Parameter Logic Function**

Based on this top design, behavior and function description will be assigned to each block, followed by the inner structure configuration.

The input circuit mainly deals with the input signal optimization and allocation, providing the three main logic blocks (Processing Logic, Setpoints, and Extensive Conditions) proper input signals.

In the setpoints block it stores all the setpoints for this specific parameter. These setpoints can be initialized into this block or can be configured by outside control. The initial setpoints and the setpoints configuration all come from the input signals. Because of the requirement of high reliability, the data stored in the setpoints block must match the data at the input as accurate as possible.

The “Extensive Conditions” block deal with the “condition out” case. When the extensive conditions are effective, no matter what the result of the processing logic is, this system won’t give out a trip signal. Therefore, it is also a determination logic in this “Extensive Conditions” block, but not as complex as that in the “Processing

Logic".

The "Processing Logic" in this system is the core. It deals with the determination of whether to give out trip signal or not based on the comparison between the input signal and the stored/modified setpoints. Obviously there should be a finite state machine (FSM) for this block. The working scheme of this FSM is as following:

- Keep eyes on the input parameter status data;
- Whenever a parameter status data changes, determine the data type (there maybe more than one status input for one parameter) and the corresponding setpoints;
- Send a request to the Setpoints block for the specific data and the Setpoints block will provide the currently configured setpoints;
- Check if the current parameter status exceeds the configured setpoints and activate the trip signal if it does.

For example, parameter "Pressurizer Low Level" is related to neutron flux, pressure sensors, and handswitch-controlled modifications, its processing block is then designed as shown in **Fig. 3.2** (for detailed relationship of all the signals, please refer to reference [4]):

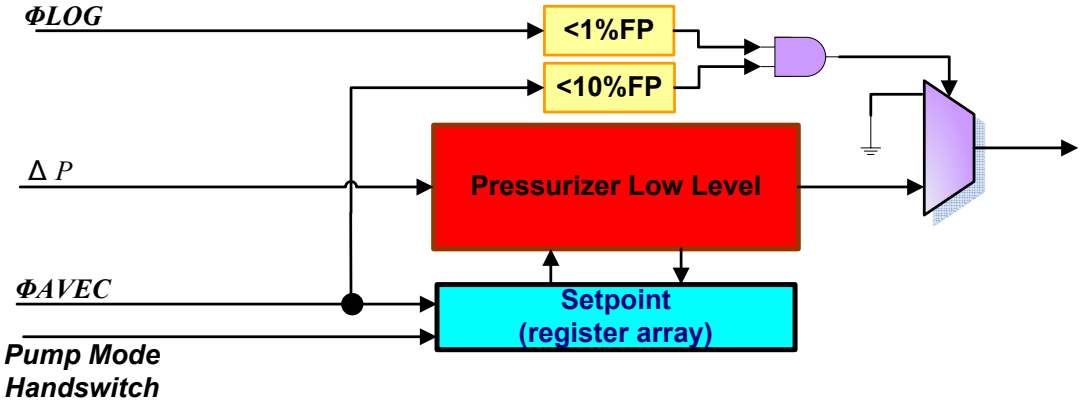


**Fig. 3.2 Processing Block for One Parameter**

The "Output Circuit" provides a way for the "Extensive Conditions" block to control the output. If the extensive conditions are satisfied, the signal from the "Extensive Conditions" block will be in charge of the "Output Circuit" and block the output of the "Processing Block" such that the "condition out" case is realized. The final output signal will also get some processing here such that it is strong and stable enough to go through the relay trip logic without much decay or oscillation.

With this accomplished "Top-down" system design, the work will go on to the hardware implementation using FPGA. The next section is introduction and discussion of FPGA.

# IV. FPGA Technology

FPGA stands for Field Programmable Gate Array. Normally, an FPGA consists of three kinds of programmable circuits and one SRAM. The three programmable circuits are:

- CLB: Configurable Logic Blocks, the main components of an FPGA that realize the logic functions;
- IOB: I/O Blocks that provide the connection between the outlet pins and the inside logic array;
- IR: Interconnection Resource that connects every CLB. IOBs and CLBs are also connected by IR.

**Fig. 4.1** shows a structure overview of an XC4000 FPGA:

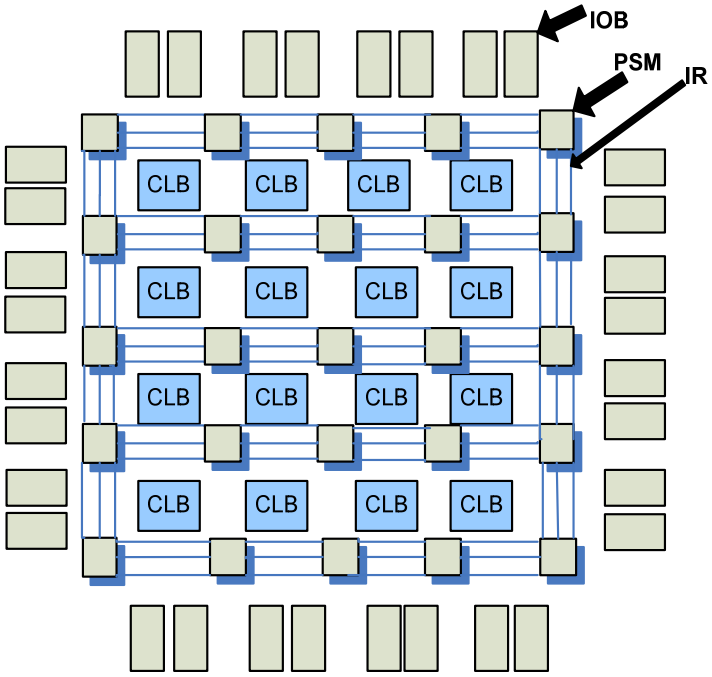


**Fig. 4.1 Structure Overview of XC4000 FPGA**

Every component here is programmable. Users can choose what function is realized in the CLB, how many CLBs are connected, and how these CLBs are connected to IOBs. This gives FPGA devices great flexibility and has made it one of the most widely used logic device in industry.

Today's technology development has also brings new characteristics to FPGA:

- High Density: 3 million—4 million gates, 1140 I/O pins, 1 Gb/s in Altera's APEX II series FPGA;
- High Performance: advanced Cu-Al CMOS technology brings fast and consistent signals;
- Short Development Period: well developed EDAs from the manufacturers help shorten the development period (A matured 100,000 gates logic development only takes less than four hours);

- Online Programming: directly programming to the device and function testing to printed circuit board.

It can be concluded from the above introduction that FPGA is a result of the development of VLSI and CAD technology. It has high integrated density, small volume, and programmability at user's end. It allows the circuit designers to achieve the design objectives by applying PC based design input, simulation, testing and verification, which shortens the developing period and reduces the cost. Furthermore, using FPGA devices can integrate products at circuit board level into products at chip level, which, as a consequence, reduces the power consumption and enhances the reliability. The online programmable ability of FPGA also gives the user great flexibility to test and modify the design [7].

Due to its great advantages in industry application, we are motivated to use this well developed and wildly used device to implement the control logic of SDS1. Another reason is that the time delay of the digital device today is shorter and shorter with the development of CMOS technology. Nanosecond level delay is not a goal that could not be reached any more, thus we are hoping we can shorten the SDS1 response time by using the advanced FPGA.

The IDE (Integrated Development Environment) tool kits used for FPGA design today are very matured and powerful, as a result of the EDA (Electronic Design Automation) development. They can do high level synthesis for the design inputs (principle diagram, waveform, and HDL code). It can also do design automation work such as floor planning, connection routing, etc. With the help of EDA tool, the FPGA design work will be easy and reliable.

Using the EDA tool, for example, Quartus II from Altera, we can then implement our system design of the SDS1 control logic to an FPGA board. The verification work, such as connecting this FPGA to a CANDU simulator to do the test will be carried on in our future research work.

# V. Feasibility and Advantages of using FPGA

One of the reasons of using FPGA is that nuclear power plant requires highly reliable I/C, which FPGA can easily achieve. From the application experience gained in industry, programmable logic control has been proved to be reliable for nuclear power plant application [6]. Moreover, the matured semiconductor manufacture technologies and hardware design methods also guarantee this. For example, FPGA technology has been utilized in satellite and other space technology for many years [5]. This shows that FPGA can work under high radiation and extreme temperature environment that is similar to that of NPP containment building.

It is also easy to do maintenance for FPGAs. The property of high integration ensures

that only a few FPGA boards are needed for a system because a high density FPGA can even offer several applications in one FPGA board. As a consequence, the maintenance cost is greatly reduced as well as the complexity.

One other reason, as has mentioned in the introduction section, is that using direct hardware implementation of SDS1 control logic provides an easier and explicit method to show CNSC the possibility coverage and qualification of SDS1, which can reduce the license application period.

The last reason, which is also the most important one, is that applying FPGA technology to improve the SDS1 control logic may enhance the operating power. This is called “Power Upgrading”.

As we know, the main task of SDS1 is to trip the reactor core by dropping the shutoff rods during an accident so that the chain reaction can be stopped and the reactor power can be pulled down. Today, when an accident occurs, the time for the shutdown system to drop the shutoff rods is hundreds of milliseconds. For this reason, reactors must operate at a safe power level that is much lower than their maximum possible power level. The slower SDS1 responds to the power rising due to accident, the wilder the gap between the maximum power level and the operating power level. Then we are greatly motivated that we may get extra power if we can shorten the response time of SDS1 such that the reactor can be operating at a higher power level without affecting the safety margin. This idea is illustrated in **Fig. 5.1**. Since the delay time of today’s advanced digital devices has reached nanosecond level, we choose FPGA to be the device used to implement the control logic of SDS1. Moreover, direct hardware implementation of the SDS1 control logic will be easier for showing the possibility coverage that is necessary for getting license from CNSC.

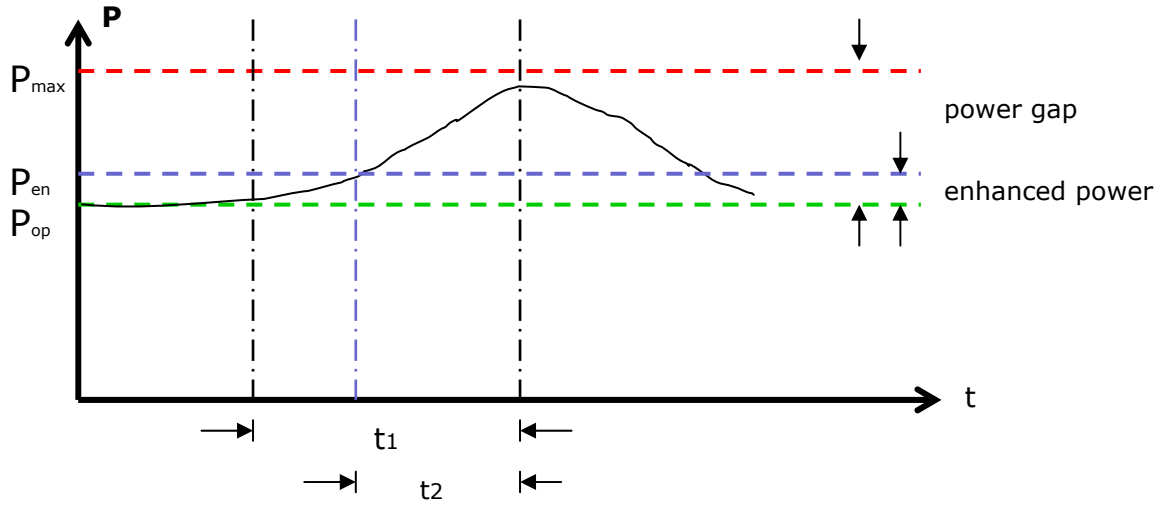


**Fig. 5.1**

## Summary

The technology renovation, which is being carried on for the existing CANDU refurbishment and new CANDU design, has provided many new topics to research in CANDU instrumentation and control. Improving the SDS1 control is one of them. By using FPGA technology, we propose that the response time of SDS1 to an accident may be shortened such that the reactor can operate at a higher power level without affecting the safety margin. Moreover, the direct hardware implementation supported by the matured EDA tools also provides an explicit way to apply for CNSC license.

# VI. References

[1] H.M.Banford & R.A.Fouracre, "Nuclear Technology and Ageing", 0883-7554/99, IEEE, 1999

[2] Atomic Control Board Canada, “Requirements for Shutdown Systems for CANDU Nuclear Power Plants", Regulatory Documentation R-8, 1991

[3] C.Senthil Kumar, A.John Arul, Om Pal Singh, and K.Suryaprakasa Rao, "Reliability Analysis of Shutdown System", Annals of Nuclear Energy 32 (2005) 63-87

[4] "Design Manual of Safety System for CANDU Nuclear Power Plant", AECL Design Manual.

[5] J. de. Grosbois and R. Nishimura, “High-Level Design Automation and Field-Programmable Devices: Technology and Applications”, AECL Report COG-97-028 (Rev.0), November 1997

[6] T.Kouthon & J.D.Decotignie, "Improving Time Performances of Distributed PLC Applications", 0-7803-3685-2/96, IEEE

[7] Zhenyong Chu and Muyun Weng, “Design and Application of FPGA”, Xidian University Press, 2002

[8] D.Pirus, Edf Septen, "Future Trends in Computerized Operation", IEEE 7th Human Factors Meeting, 2002

[9] R.G.Huget, M.Viola and P.A.Froebel, "Ontario Hydro Experience in the Identification and Mitigation of Potential Failures in Safety Critical Software Systems", IEEE Transaction on Nuclear Science, VOL. 42, No.4, August 1995

[10] R.G.Orendi, D.S.Petras, S.R. Prokopovich, R.R.Oft, "Outage Optimization versus Shutdown Risk", 92CH3233-4, IEEE, 1993