A UNIFIED ARCHITECTURE OF A SYSTOLIC, INTEGRATED SENSOR ORIENTED, PROCESS MONITORING NETWORK

by

Isaac Shpancer

A Thesis

presented to the University of Manitoba

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the

Department of Electrical Engineering

Winnipeg, Manitoba, 1983

A UNIFIED ARCHITECTURE OF A SYSTOLIC, INTEGRATED SENSOR ORIENTED, PROCESS MONITORING NETWORK

BY

ISAAC SHPANCER

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

Monitoring of data skew sensitive or vector oriented processes is limited with the present data acquisition system architectures. A unified vector oriented architecture that eliminates the traditional barriers (both technological and conceptual) between the phenomena sensing, acquisition, processing and networking layers, utilizing silicon as the implementation material, has been postulated. A new converter architecture has been developed based on the non fatigue properties of silicon to form highly accurate integrated sensors, thus extending the concept of systolic computation to a new integrated sensing, correction, conversion and computation structure - the systolic converter. The problem of a vector-oriented process occurring in a globally monitored area, partitioned into locally monitored areas, has also been identified. A new systolic network architecture has been developed to facilitate distributed systolic converter synchronization and systolic processing throughout the network. The thesis also shows that for a subspace of parallel algorithms, a systolic network can be designed with the help of a unified set of design rules that apply for both the microarchitecture (VLSI level) and macroarchitecture (network level). The systolic network extends and unifies the basic concepts of systolic computations and local network architecture, by introducing a Unified Network Element as the network counterpart of a systolic computational cell. It has also been shown that subspaces of systolic algorithms are directly applicable to global systolic network processing. The vitality of the systolic network has been proven by its methodological design implementation and testing in various applications never attempted before.

ACKNOWLEDGEMENTS

I would like to thank Dr. Witold Kinsner, my advisor, for his generous help and gentle illumination and guidance throughout the research work.

I would also like to thank the members of the advisory committee: Drs. E. Schwedyk, H.C. Card and M. Doyle for their contribution to the work structure and its essence. Thanks are due to Dr. A. Wexler for the encouragement in this work and the opportunity to acquire a better appreciation of fields.

The work of my colleague D. Hildebrand on the implementation of the unified network element and network protocol is appreciated. Thanks also goes to my colleague M. Jullian for reading of the thesis and to other colleagues for contributing discussions. The devoted work of V. Smith on the word processor is acknowledged.

The University of Manitoba is acknowledged for providing the atmosphere of a house of learning. The financial support of the Industrial Applications of Microelectronics Centre and the Natural Sciences and Engineering Council (NSERC) are gratefully acknowledged. Thanks also belongs to Linear Technology Inc. and the Atomic Energy of Canada Ltd., Chalk River Labs, for useful discussions. Special thanks go to the founders of the Microelectronics Centre who made it possible that research, industrial training and product development can all support one another.

And finally, this work also belongs to my wife Tova, my daughters Tal (morning dew), Adva (ripple) and my son Gal (wave).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

TABLE OF ABBREVIATIONS

A/D	Analog to Digital
BER	Bit Error Rate
CA	Channel Attention
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAT	Computer Aided Tomography
CC	Conversion/Correction
CCC	Conversion/Correction Cell
CCH	Conversion/Correction Hybrid
CCU	Conversion/Correction Unit
CF	Cost Function
CSCE	Composite Systolic Cell
DAS	Data Acquisition System
DMA	Direct Memory Access
DP	Destination Pointer
DPLL	Digital Phase Locked Loop
EMI	Electromagnetic Interference
FIFO	Fist-In First-Out
FRAM	RAM Based FIFO
FRC	FIFO RAM Controller
FTF	Fall Through FIFO
GMA	Globally Monitored Area
HS	Hybrid Sensor
ICS	Integrated Coherent Sensor
IDAS	Isolated Data Acquisition System
IOP	Input/Output Processor
IPSP	Inner Product Step Processor
IS	Integrated Sensor
MA	Monitored Area
MIDAS	Multiplexed Isolated Data Acquisition System
MUX	Multiplexer
NIDAS	Non-Multiplexed Isolated Data Acquisition System
NMRR	Normal Mode Rejection Ratio
NSA	Network Signature Analyzer
PLC	Programmable Logic Controller
PLL	Phased Locked Loop
RA	Radiation Area
RAM	Random Access Memory
RP	Read Pointer
S & H	Sample & Hold
SA	Successive Approximation
SAU	Systolic Array Unit
SC	Systolic Converter
SCN	Status Counter
SCT	Systolic Converter Throughput
SICO	Simultaneous Converter
SIMD	Single Instruction Multiple Data
SP	Source Pointer
SSC	Strobed Systolic Converter
ST	System Throughput
UNE	Unified Network Element
UNP	Unified Network Protocol
USCM	Universal Simultaneous Counting Module
V→f	Voltage to Frequency
VLSI	Very Large Scale Integration
WP	Write Pointer

CHAPTER I

INTRODUCTION

1.1 Problem Definition

A typical data acquisition system (DAS) is shown schematically in Fig. 2.1(a). Such a system includes sensors, signal conditioning units, a multiplexer (MUX), one analog-to-digital (A/D) converter and a computer. Although the basic building blocks of such a system may take different forms, the blocks are usually considered as lumped devices or subsystems, performing their functions and communicating data under the control of the computer. Such lumped general purpose DAS architectures traditionally utilize discrete, hybrid, or large scale integration (LSI) components for the sensors and conversion blocks, while very large scale integration (VLSI) is usually confined to the computing and control and communication blocks.

Such a DAS usually collects data from a remotely monitored area (MA) through a set of sensors. To assure that the data acquired from sensors located in noisy environments are not contaminated with unwanted noise, multiplexed isolated data acquisition systems (MIDAS) are utilized. The MIDAS exhibits the following features: (i) Ohmic isolation between the computer and an MA; (ii) High common-mode noise rejection; and (iii) High normal-mode noise rejection. Implementation of the MIDAS is based on the following three architectures: (i) Guarded DAS; (ii) Flying capacitor DAS; and (iii) DAS with isolation amplifiers, as shown in Fig. 2.2 and discussed further in Chapter 2.

All three MIDAS architectures have many drawbacks which include: (i) The data acquisition chain incorporates unnecessary translations of signals from discrete to analog form, and vice versa, thus reducing the overall DAS accuracy and reliability; (ii) The isolation schemes require either electromechanical devices or expensive hybrid isolation amplifiers; (iii) All channels, each associated with a sensor and local or remote conditioning unit, are scanned sequentially, thus reducing the DAS throughput; (iv) The sequential channel scanning results in an inherent skew of data, i.e., each successive data point acquired is delayed with respect to its predecessor; (v) No unified generic sensor family exists for phenomena measurements; and (vi) The overall system for a large number of channels is bulky and does not lend itself to miniaturization.

Non-multiplexed isolated data acquisition systems (NIDAS) schemes based on N conversion units, each attached to a single channel, are shown in Fig. 2.1(b) and Fig. 2.3, respectively. Although the NIDAS schemes enhance the system data throughput and may reduce the data skew (if all conversion units are activated simultaneously), other drawbacks remain unchanged. The two enhancements may be offset by a new computational bottleneck due to the increased data throughput.

As a result of the previous discussion, we could classify the conversion methods into three groups: (i) Scalar; (ii) Vector; and (iii) Array. The scalar conversion method relies on a single data conversion unit which is time-multiplexed between N channels. Except for the single channel configuration (N=1), this method inherently

leads to data skew between channels. The vector conversion method is based on N conversion units, each attached to a single channel, that are activated simultaneously. If the conversion time of an A/D converter is not a function of a corresponding channel input signal, data skew is eliminated completely. The array conversion method is based on two or three dimensional assemblies of simultaneously activated conversion units. The vector conversion method results in a data set, with N elements. The set shall be called either a state vector of an MA or a state vector. Process monitoring applications that are data skew sensitive, will be referred to as vector oriented processes.

Converters based on either the scalar or non-scalar conversion methods will supply streams of data in the form of scalars or vectors or arrays. The streams can be processed by different computing structures, such as microprocessors, multiple processors and systolic arrays. A mismatch between the form of input data (scalar or vector) and the selected computer will naturally lead to a computational bottleneck. For example, consider a vector oriented process to be monitored at a single MA. To eliminate the data skew, one would use N simultaneously activated converters (NIDAS architecture) to obtain a stream of state vectors. If the state vectors would have to be processed by a microprocessor, according to a given algorithm, the mismatch would occur, and the computational bottleneck could impose a limit on the system data throughput. The system throughput is reduced even further, if the microprocessor must be involved in the compensation and linearization of sensors.

The single MA data acquisition and processing just described may not be adequate for processes exhibiting variations in their characteristics from one area to another. In such a case, a vector oriented process occurring within a globally monitored area (GMA), should be partitioned into vector oriented subprocesses, each related to a distinct MA. Consequently, acquisition and processing of the global state vector, through a distributed vector oriented NIDAS, poses the following architectural problems: (i) How to obtain a global state vector from the distributed NIDAS, each capable of monitoring a vector oriented subprocess in a single MA; and (ii) How to process the global state vector. The general purpose nature of a conventional NIDAS provides only a partial solution or no solution at all to the questions. Clearly, a microprocessor will not be adequate to assure a high system data throughput.

The advent in VLSI circuits and systems, particularly in highly concurrent computational structures and integrated sensors have recently provided excellent opportunity for high performance special purpose, distributed vector oriented NIDAS. However, most of the developments reported in the available literature concentrate on improvements of various DAS building blocks through research conducted in disjoint disciplines (e.g. integrated sensors, VLSI computing architectures and networks), with no major impact on DAS architectures. The research presented in this thesis is intended to fill the gap by providing a unified network architecture for vector oriented processes. The architecture utilizes VLSI methodology (with required extension)

throughout the phenomena sensing, data acquisition, data processing and data communications layers.

1.2 Research Objectives

The following three major research objectives were set in this thesis:

a) To develop a unified network architecture for vector oriented processes;

b) To formulate the building blocks of the architecture which include: (i) Integrated Sensors; (ii) Systolic Converter; (iii) A Unified Network Element; and (iv) A Systolic Network. The name systolic has been selected to emphasize the concurrent nature of the data conversion method, data processing, as well as global synchronization of the network related to vector oriented processes; and

c) To develop the kernel of a design methodology for the unified architecture.

The three research objectives were intended to advance the knowledge, technology, and design methodology, respectively, reflecting an integrated research, engineering and technology transfer environment where the thesis work was conducted. The results presented in this work fulfil the research objectives.

1.3 Literature Survey of Related Work

A literature survey on data skew sensitive distributed DAS was

conducted with emphasis on the following application areas: (i) Nuclear site and nuclear dump radiation monitoring systems; (ii) Nuclear physics experimentation; (iii) Meteorological DAS; (iv) Wind tunnel tests; (v) Offshore oil rigs; and (vi) Lasers. Although a vast literature and many books [1-4] refer to DAS, the survey has revealed no direct references related to the scope of the thesis, excluding a technical report, presentation and two publications related to this work that are described below.

A nuclear radiation monitoring system, utilizing a hierarchical multiloop network architecture and simultaneous counting modules, was developed and implemented at the Nuclear Research Centre Negev (NRCN) in Israel in 1977-1980 [5].

The above system was presented to the Electrical, Instrumentation and Control Division of the Chalk River Nuclear Labs [6]. Valuable data on the intelligent safety system for the new generation of CANDU reactor was obtained in subsequent discussions, leading to a systolic reactor shut-down system, also described in [7] and [8]. A systolic network architecture for radiation monitoring of a nuclear site is described in [7]. A data-skew sensitive process monitoring system with emphasis on integrated sensors and systolic conversion, followed by a synthesis of a systolic shut-down system of a CANDU reactor, is described in [8]. Other work related to more limited scope of the thesis is given below.

A DAS based on simultaneous activation of multiple sample and hold

units each connected to a remote sensor, followed by a tree of multiplexers associated with A/D units, is described in [9]. Neither the computational bottleneck nor the isolation problem is addressed in that work. A simultaneous counting system for neutron activation measurement of the Shiva-Laser facility, is described in [10]. The system detects α and β radiation from four detectors similar to the method described in [5]. A layered network architecture with limited synchronization features was developed at the Chalk River Labs and is described in [11], [12]. Their system does not include a vector oriented process control implementation, although the network exhibits features that with few modifications could enable such an implementation.

Integrated sensor fabrication technology requiring costly laser trimming to achieve accuracy is described in [13], followed by a suggestion of a microprocessor based sensor calibration. No exact procedures are mentioned and sensor analog output is assumed. We can conclude from the survey that the VLSI technology has had almost no overall impact on the distributed NIDAS and vice versa.

1.4 Work Organization

The work is divided into ten chapters, following a bottom to top approach that reflects the evolution of the ideas through an analysis and synthesis spiral.

Chapter II analyzes simultaneous conversion methods and pulse

stream sensors. A uniprocessor based architecture is synthesized and its performance is assessed. A description of a nuclear site radiation monitoring system is provided to emphasize the scope of the problem. Key limitations of the architecture are analyzed to enhance the understanding of the problem defined in the introduction.

Chapters III to VI provide the kernel of the work, following the natural data flow and problem definition. Chapter III analyzes the mechanical properties of silicon suitable for phenomena sensing. An integrated sensor architecture is synthesized to enable concurrent sensor linearization and sensor data conversion in an integrated systolic cell.

Further integration occurs in Chapter IV where the composite problem of state vectors, sensing conversion and processing is analyzed, resulting in a new systolic conversion method and its VLSI embodiment (a systolic converter) that are described. A detailed architecture of a systolic converter for a CANDU reactor shut-down is synthesized, emphasizing the key architectural features and the level of obtained integration.

The processing of global state vectors from a partitioned GMA is analyzed pointing to the need for a new network architecture called a systolic network. Chapter V describes the network extensions of a systolic converter to form a unified network element (UNE) capable of sensing, processing and communicating. A network architecture, supporting distributed UNEs for global state vector processing is

synthesized using the same principles as for the systolic converter. The analogy between a systolic cell and a UNE is further emphasized when the network is operated as a computational structure with state vectors as the basic data elements. A new concept, resulting from the systolic nature of the network architecture, called network signature, is described which enables network diagnosis, supervision and performance evolution.

Chapter VI provides an analysis of the network performance when operated in two basic modes: (i) Global state vectors collection; and (ii) Global state vectors processing. Systolic algorithms for matrix vector multiplication are expanded to work on a systolic network and the first order throughput and congestion measures are derived. The network architecture for this extension is described utilizing the systolic cell and UNE symmetry.

An object oriented design methodology, composed of the basic architectural building blocks (integrated sensors, systolic converter, UNE, systolic network) is provided in Chapter VII. Each building block has its associated set of attributes facilitating the design and synthesis of a variety of configurations. Each configuration is fully described by a set of objects and their associated sets of attributes embedded in the unified architecture.

Chapter VIII provides a case analysis of a reactor shut-down system, described in Chapter IV, by applying the design methodology of Chapter VII. A description of the application of the systolic network

architecture and methodology is provided in Chapter IX. Various data skew sensitive processes and their architectural solutions are presented. Of special interest are the applications related to coherent systolic array conversion and laser strobed applications that also point to a new research avenue as a direct extension of this work.

Chapter X provides both the conclusions from this work and recommendations for further research in the area of systolic array conversion and the required systolic network extensions. The impact of integrated optoelectronics and fibre-optic sensor technology on the architecture has to be further investigated.

The results of this work should be of interest to both researchers and designers in the related fields.

CHAPTER II

SIMULTANEOUS CONVERSION METHOD

This chapter will describe a lumped non-multiplexed isolated data acquisition system (NIDAS) architecture suitable for vector oriented process monitoring. Pulse stream hybrid sensors will be developed to eliminate the distinction between monitoring of discrete and continuous phenomena. The vector conversion method, in combination with a generic family of hybrid sensors, constitutes an all digital simultaneous conversion method. Uniprocessor embodiments of this method will be described, followed by their performance analysis. Although the lumped NIDAS architecture exhibits superior performance to any MIDAS, its limitations will be analyzed to serve as a kernel for an integrated vector oriented architecture described in Chapter III. The chapter is of importance for both researchers and designers since many of the possible embodiments of integrated architectures could be tested with lumped implementations.

2.1 Multiplexed and Nonmultiplexed DAS

A lumped DAS could take either of the following two forms: (i) A multi-plexed DAS when a single conversion unit is time-multiplexed between N channels (each channel represents a sensor and the associated signal conditioning), as shown in Fig. 2.1(a); or (ii) A nonmultiplexed DAS with a conversion unit attached to each channel, as shown in Fig. 2.1 (b).

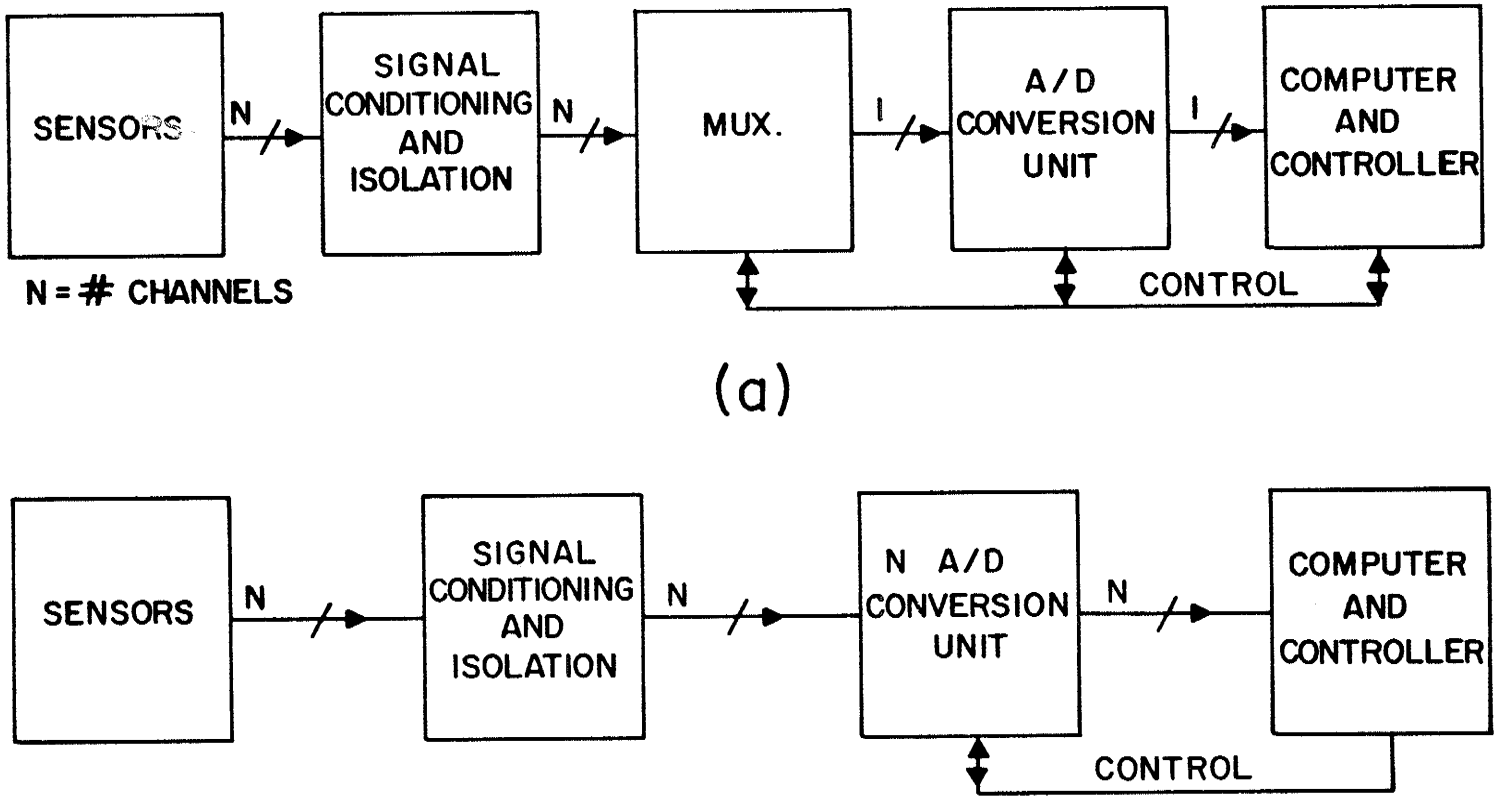


Fig. 2.1. Data acquisition systems. (a) Multiplexed. (b) Nonmultiplexed.

2.1.1 Multiplexed isolated DAS

Reliable sensor data acquisition of a remotely monitored area (MA) in a noisy environment will require an isolated DAS (IDAS) with the following features: (i) Ohmic isolation between the computer and MA; (ii) High common mode noise rejection; and (iii) High normal-mode noise rejection. There are three basic multiplexed isolated DAS (MIDAS) architectures: (i) Guarded MIDAS; (ii) Flying capacitor MIDAS, and; (iii) MIDAS with isolation amplifiers.

The guarded MIDAS, shown in Fig. 2.2(a), relies on a three wire relay analog multiplexer (MUX) and a guarded isolated dual-slope conversion unit that provides the isolation barrier, as well as common and normal mode noise rejection. The normal mode rejection is an embedded feature of the converter due to the integration over a constant period of time of the sensor output, resulting in a comb filter characteristics.

The flying capacitor MIDAS (Fig. 2.2(b)) relies on a two-wire relay MUX that switches a capacitor, charged to the channel voltage, to the input of the successive approximation (SA) A/D converter. The capacitor switched between the input and output acts as an isolated sample and hold (S & H) device.

The third method (Fig. 2.2(c)) relies on an isolation amplifier attached to each channel to provide the isolation barrier, followed by a single ended solid state MUX.

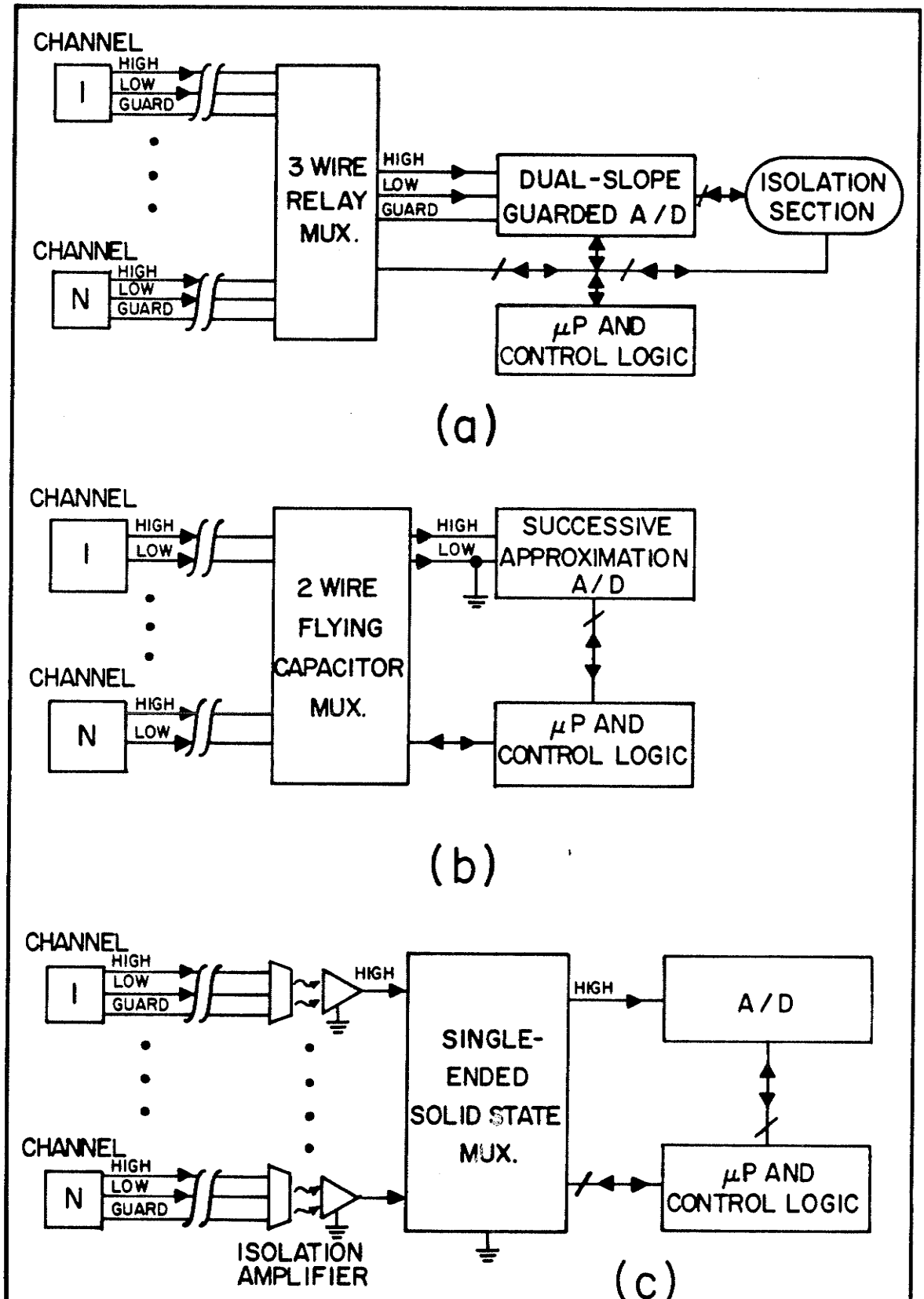


Fig. 2.2. Multiplexed isolated data acquisition system (MIDAS)
(a) Guarded MIDAS. (b) Flying capacitor MIDAS.
(c) MIDAS with isolation amplifiers.

As highlighted in Chapter I, all three MIDAS architectures have the following characteristics: (i) The data acquisition chain incorporates unnecessary translations from discrete to analog form and vice versa, thus reducing the overall system reliability and accuracy. Furthermore, sensors with pulse stream outputs resulting from the measured phenomenon (e.g. radiation, shaft speed) must be conditioned to an analog form; (ii) The isolation scheme requires either electromechanical devices or hybrid isolation amplifiers; (iii) All channels, each associated with an analog sensor or a conditioned discrete sensor, are scanned sequentially, thus reducing the MIDAS throughput (the number of converted channels/second); (iv) The sequential channel scanning and conversion result in an inherent data skew of related sensor signals; (v) No unified generic sensor family exists for phenomena measurements; and (vi) The overall system for a large number of channels is bulky and does not lend itself to miniaturization. A nonmultiplexed isolated DAS (NIDAS) architecture capable of removing some of the problems of MIDAS is described in the following section.

2.1.2 Nonmultiplexed Isolated DAS (NIDAS)

Three basic architectures for nonmultiplexed isolated DAS (NIDAS) are shown schematically in Fig. 2.3 (a)(b)(c). They are intended to increase system data throughput and reduce data skew, thus addressing two of the drawbacks associated with a MIDAS. If certain conditions are satisfied, data skew can be eliminated completely forming a lumped embodiment of the vector conversion method.

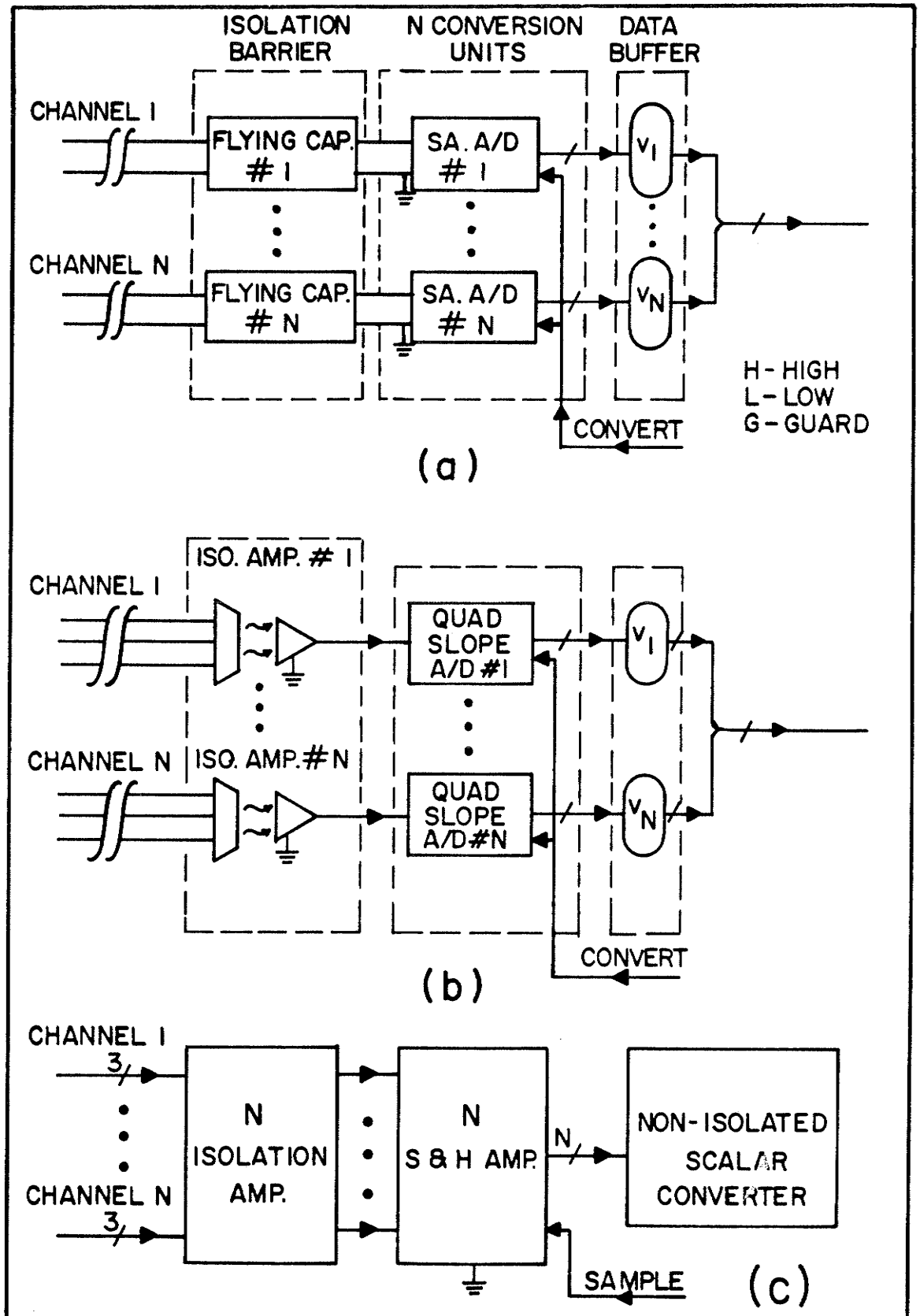


Fig. 2.3. Nonmultiplexed isolated data acquisition system (NIDAS) (a) Flying capacitor NIDAS. (b) Isolation amplifier based NIDAS. (c) Simultaneous sample and hold NIDAS.

The architecture presented in Fig. 2.3(a) relies on N flying capacitor relays for isolation and sampling, followed by N successive approximation (SA) A/D coverters. The output of each converter v_j (for $1 \leqslant j \leqslant N$) is latched into a corresponding data buffer. The data from each buffer is further read and processed by a uniprocessor.

The second architecture shown in Fig. 2.3(b) incorporates N isolation amplifiers, followed by N quad-slope (or dual slope) monolithic A/D converters. The output data of each converter v_j is latched into a corresponding data buffer for further processing by a uniprocessor.

A modification to the above architectures is obtained, as shown in Fig. 2.3(c). Here, N isolation amplifiers are followed by N S&H units that are activated simultaneously. The state vector, captured in its analog form by the S & H array is converted sequentially by a fast single ended solid state MUX and SA A/D converter chain. The method described above reduces the overall cost of the NIDAS.

If all the sampling and conversion units in the above three configurations are activated simultaneously, and the sampling/ conversion time for each channel is not a function of the sensor signal, the architectures represent embodiments of the vector conversion method. On the other hand, if the sampling/conversion time depends on the amplitude of the sensor signal, data skew between channels may not be eliminated, even though all the units are activated simultaneously. Consequently, the architectures could not be used for the vector conversion method.

The elimination of data skew and the increase in the system throughput solve only two of the problems common to MIDAS. Even if monolithic conversion units or simultaneous S & H modules are used, the system is very bulky due to the additional hardware required to alleviate the remaining drawbacks. The increase of throughput, approximately by a factor of N, creates a new computational bottleneck at the uniprocessor, thus potentially limiting the number of channels.

We have analyzed above the multiplexed and nonmultiplexed isolated DAS architectures to identify their limitations. The discussion also singled out simultaneous activation and the independence of the sampling/conversion time of the amplitude of input signals as the necessary and sufficient conditions for a NIDAS with analog sensors to be vector oriented. The next section will assess the time and data dependence for three fundamental conversion methods, partially reflected in the previous discussion.

2.2 Scalar Vector and Array Conversion Method

A conventional DAS could be subdivided into a conversion subsystem followed by a processing subsystem. Let us observe the data flow and time dependence of the scalar, vector, and array conversion methods, all suitable for lumped implementations. The data flow of the three basic methods is schematically illustrated in Fig. 2.4(a)(b)(c), respectively.

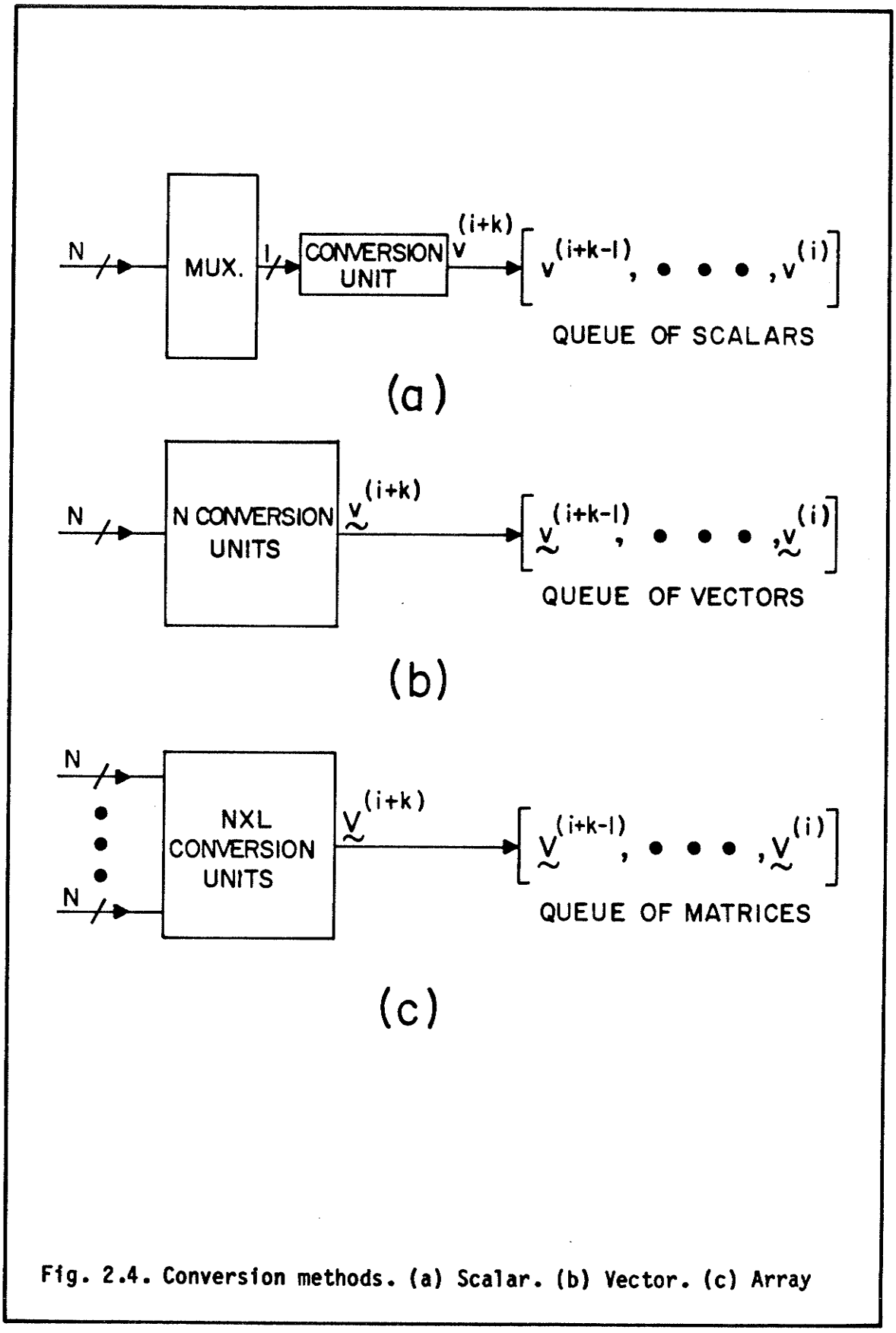


Fig. 2.4. Conversion methods. (a) Scalar. (b) Vector. (c) Array

The scalar conversion method, shown in Fig. 2.4(a), relies on a single conversion unit which is time multiplexed between N channels. The data element $v^{(i+k)}$ at the output of the conversion unit represents the measured sensor signal (and the related phenomenon) at a time t_{i+k}. Since each of the data elements has a different time tag, the resulting queue will be referred to as a queue of scalars. The k scalars contained in the queue are further processed following a given algorithm. It should be noted that the scalars do not have to represent equally time-spaced measurements or a sequential scanning of the channels.

The vector conversion method, shown in Fig. 2.4(b), utilizes N simultaneously activated conversion units, connected to N sensors spatially distributed in an MA. The resulting data set has no time skew between its N data elements. The data set is presented as an N tuple $\underset{\sim}{v}^{(i+k)}$, where the superscript denotes the common time tag to all vector elements. The vectors are placed on a queue of length k for further processing. Since a vector represents the state of MA at a given time, it could be viewed as a point in an N dimensional vector space. The name state vector reflects the meaning of $\underset{\sim}{v}^{(i+k)}$. The queue of k state vectors represents a trajectory in the vector space from $\underset{\sim}{v}^{(i)}$ to $\underset{\sim}{v}^{(i+k-1)}$ following the vector oriented process in an MA.

The array conversion method, shown in Fig. 2.4(c), utilizes an NxL array of conversion units attached to a spatially distributed NxL set of sensors. The resulting matrix $\underset{\sim}{V}^{(i+k)}$ represents the state of MA at

time t_{i+k}. The matrices are placed on a queue of length k for further processing.

The scalar, vector and array conversion methods just described imply separate conversion and processing building blocks of the DAS. The lumped architectures for NIDAS related to vector conversion could be further improved by a new approach to sensor architecture, resulting in an all digital NIDAS. We will describe a generic family of pulse stream hybrid sensors for monitoring radiation and other phenomena.

2.3 Pulse Stream Sensors for Radiation Sensing

In this section, we will refer to sensing of α, β, γ, X-ray and neutron radiation. Radiation detectors can generate charge pulses that could be discriminated and analyzed in two basic modes: (i) Counting mode; (ii) Continuous mode. The counting mode can be further divided into two modes: (i) Single channel, when all counts represent radiation in a given energy band; and (ii) Multichannel, when the charge pulses are discriminated with regard to the energy they represent to obtain the radiation spectra. Based on this classification, it is obvious that a multichannel radiation analyzer could be viewed as a composite of many single channels. This is the basis for looking for a unified sensor architecture for radiation measurement in the single channel pulse counting mode. Once this goal is obtained, a radiation area could be monitored by fixed amplitude, pulse stream sensors. Reference [14] will serve as the basic textbook for radiation detection and measurements in the following analysis.

A hybrid fixed amplitude, pulse stream sensor for radiation detection is shown schematically in Fig. 2.5(a). The sensor consists of a radiation detector which can be remotely located from the radiation signal processing section. The detector output is fed to a charge sensitive preamplifier, followed by a discriminator and a differential line driver. The output of the sensor is a pulse stream with an instantaneous rate f. A radiation sensor could be constructed to operate also in the continuous mode, as shown in Fig. 2.5(b). The radiation detector will usually generate a current that is transformed to a pulse stream through a current to frequency converter. The hybrid sensor for radiation measurement, either in the discrete mode, or continuous mode is unified in the symbol shown in Fig. 2.5(c). The symbol identifies a hybrid sensor by HS, the measured phenomenon is denoted by P (e.g. α, β, γ) and the compensating phenomenon (if needed) by T.

The detector types, most suitable for various radiation sensing in pulse counting mode, are organized in a matrix form in Table 1. The matrix is based on the textbook [14], and provides the detector type for construction of α, β, γ and X-ray pulse stream sensors. The key sections of [14] describing each detector for a specific radiation, are given in Table 1 for easy access. Organic scintillators for γ detection are covered in [15-16]. Semiconductor detection for X-ray and γ are discussed in [17].

Fast and slow neutron detection methods utilize the following

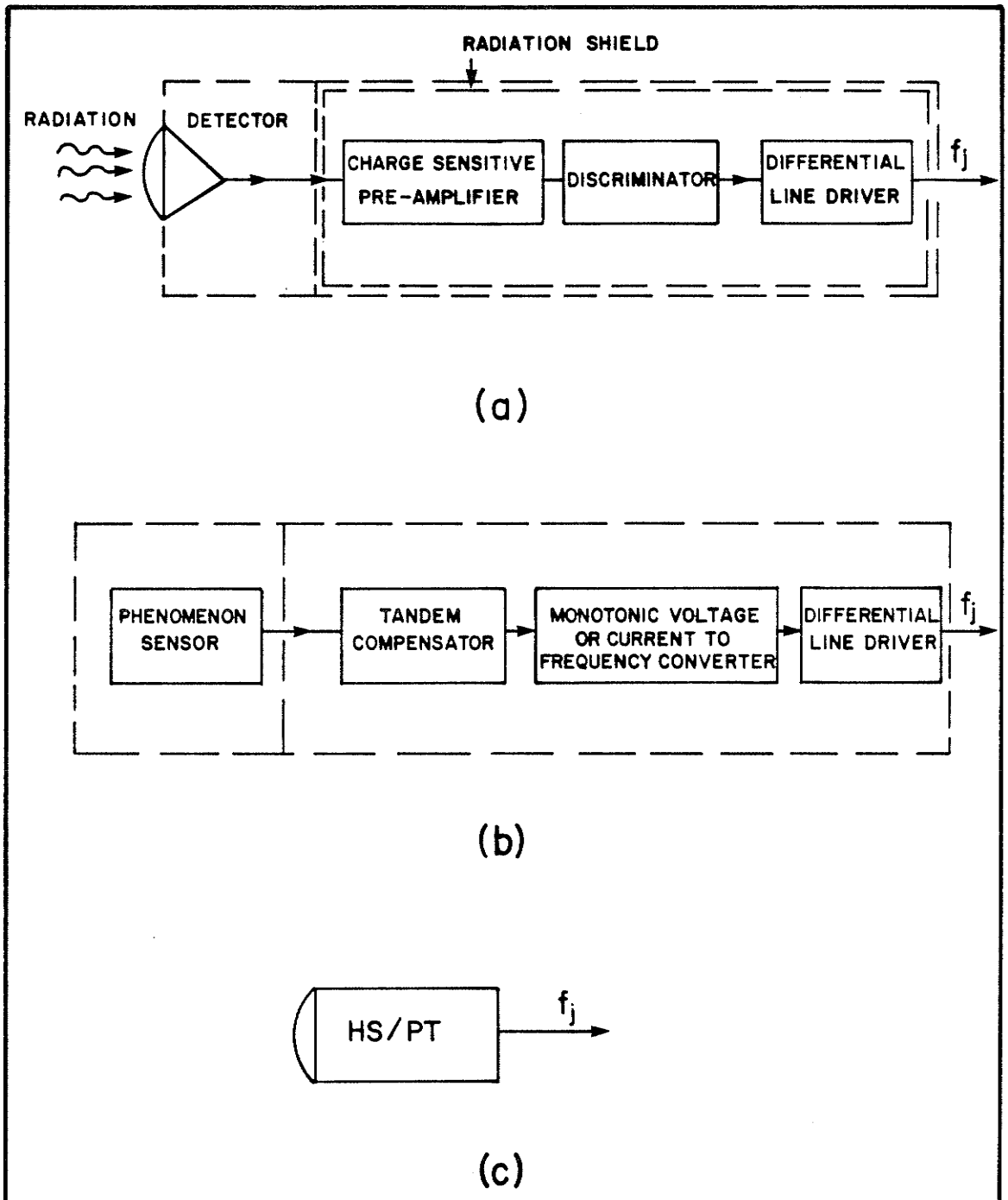


Fig. 2.5. Hybrid fixed amplitude pulse stream sensor for radiation detection. (a) In discrete mode. (b) In continuous mode. (c) Symbol.

Table 1. Pulse stream radiation detectors matrix.

RADIATION / DETECTOR	ALPHA	BETA	X–RAY & GAMMA
G–M COUNTER	NA	NA	GAMMA RAYS
PROPORTIONAL COUNTERS	WINDOWLESS FLOW CNTR. ALPHA COUNTING	ORGANIC SCINTILLATORS WINDOWLESS FLOW CNTR. GAS-PROP. SCIN. CNTR.	X–RAY & GAMMA RAY SOURCES
ORGANIC SCINTILLATORS	RESPONSE OF ORGANIC SCINTILLATORS	RESPONSE OF ORGANIC SCINTILLATORS	RESPONSE OF ORGANIC SCINTILLATORS
INORGANIC SCINTILLATORS	INORGANIC SCINTILLATORS	INORGANIC SCINTILLATORS	INORGANIC SCINTILLATORS
LIQUID SCINTILLATOR	LIQUID SCINTILLATOR COUNTERS	LIQUID SCINTILLATOR COUNTERS	LIQUID SCINTILLATOR COUNTERS
SEMICONDUCTORS	ALPHA PARTICLE SPECTROSCOPY	ELECTRON SPECTROSCOPY	GAMMA RAY SPECTROSCOPY WITH Ge(LI) DETECTORS

pulse stream detectors [18-20]: (i) BF_3 counters; (ii) ^3He counters; (iii) Boron-lined proportional counters; (iv) Fission counters; (v) Boron loaded scintillators; (vi) Lithium loaded scintillators; (vii) Recoil proportional counters; and (viii) Recoil scintillators.

The key result of the discussion above is that hybrid pulse stream radiation sensors could be constructed with the proper selection of detectors and signal conditioning circuitry. A complete signal conditioning circuitry is available in a hybrid form as a result of military and space products utilizing pulse counting radiation detection [24].

The next section will describe pulse stream sensors for continuous phenomena (e.g. temperature T, pressure P) monitoring.

2.4 Pulse Stream Sensors for Continuous Phenomena

A pulse stream output sensor for continuous phenomenon measurements is depicted in Fig. 2.5(b). The phenomenon sensor related to channel j outputs a voltage or current through a tandem compensator (usually compensating for temperature) that is transferred to a pulse stream f_j via a monotonic voltage or current to frequency converter. Some commercially available pulse stream hybrid sensors for pressure, temperature and flow, have recently been reported [25]. A new kind of an all digital NIDAS architecture will be described in the next section as a direct result of the pulse stream sensor architecture. Although

the description will relate to the implementation of a radiation monitoring system for a nuclear site [5], it could be extended to other vector oriented processes.

2.5 Uniprocessor Implementation of a Simultaneous Converter

An embodiment of a vector conversion method utilizing pulse stream hybrid sensors is described in the following sections and in greater detail in [5]. Although initially designed for vector oriented radiation monitoring, it is directly applicable to other fields.

2.5.1 Simultaneous Converter Architecture

Analog sensor NIDAS architectures were described in Sec. 2.1.2 as an embodiment of the vector conversion method. If pulse stream sensors are utilized and the conversion unit is a gated counter, a new architecture is formed that will be called a simultaneous converter (SICO).

The architecture of a SICO is shown in Fig. 2.6. A radiation area MA is monitored by N fixed amplitude pulse stream hybrid sensors. Each sensor related to channel j is generating a pulse stream of instantaneous rate f_j as a function of the radiation rate. The pulse streams are isolated via N opto-gates to provide the required barrier. A cluster of programmable counters with a common conversion gate forms N simultaneous conversion units, satisfying the basic conditions of a vector conversion method. The state vector that represents the average radiation rate is time tagged and further processed via a micro-

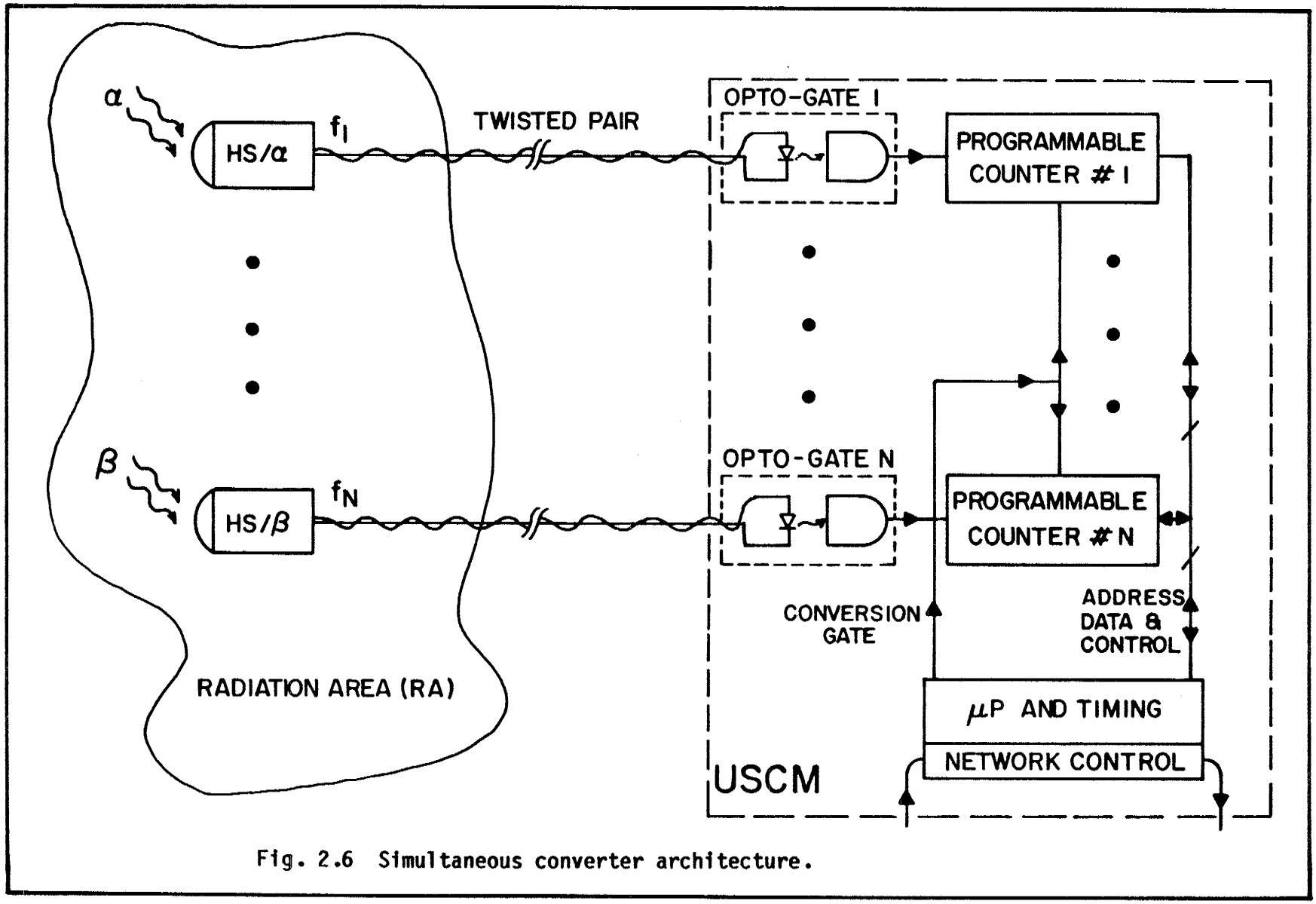


Fig. 2.6 Simultaneous converter architecture.

processor. A microprocessor (μP) is usually sufficient to perform the local processing.

Such a μP based unit that performs the described function is called a Universal Simultaneous Counting Module (USCM). The immediate advantages of the USCM embodiment are: (i) A unified sensor family; (ii) Improved noise immunity due to the transmission method and the averaging process in the counter (similar to a comb filter for a quad slope A/D); (iii) Digitally implemented optical isolation; (iv) No analog conversion units; (vi) Utilization of VLSI components in the conversion and processing sections.

2.5.2 USCM Data Flow and Pipelining

The data flow of a USCM is illustrated in Fig. 2.7(a). The instantaneous state vector $\underset{\sim}{f}$ is converted and time-tagged by N commonly gated counters resulting in a state vector $\underset{\sim}{\bar{f}}^{(i)}$ that is buffered in a queue of length one. The state vector data set is viewed as contigiuous locations in the μP address space. A direct memory access (DMA) controller, operating in the memory to memory transfer mode, enables further queueing of state vectors in the system RAM, as well as initialization sequences without μP intervention. The processing of the queue is performed by the μP, and is pipelined with the state vector conversion.

The pipelined conversion, DMA and processing scheme is shown in the diagram of Fig. 2.7(b). The conversion, processing and

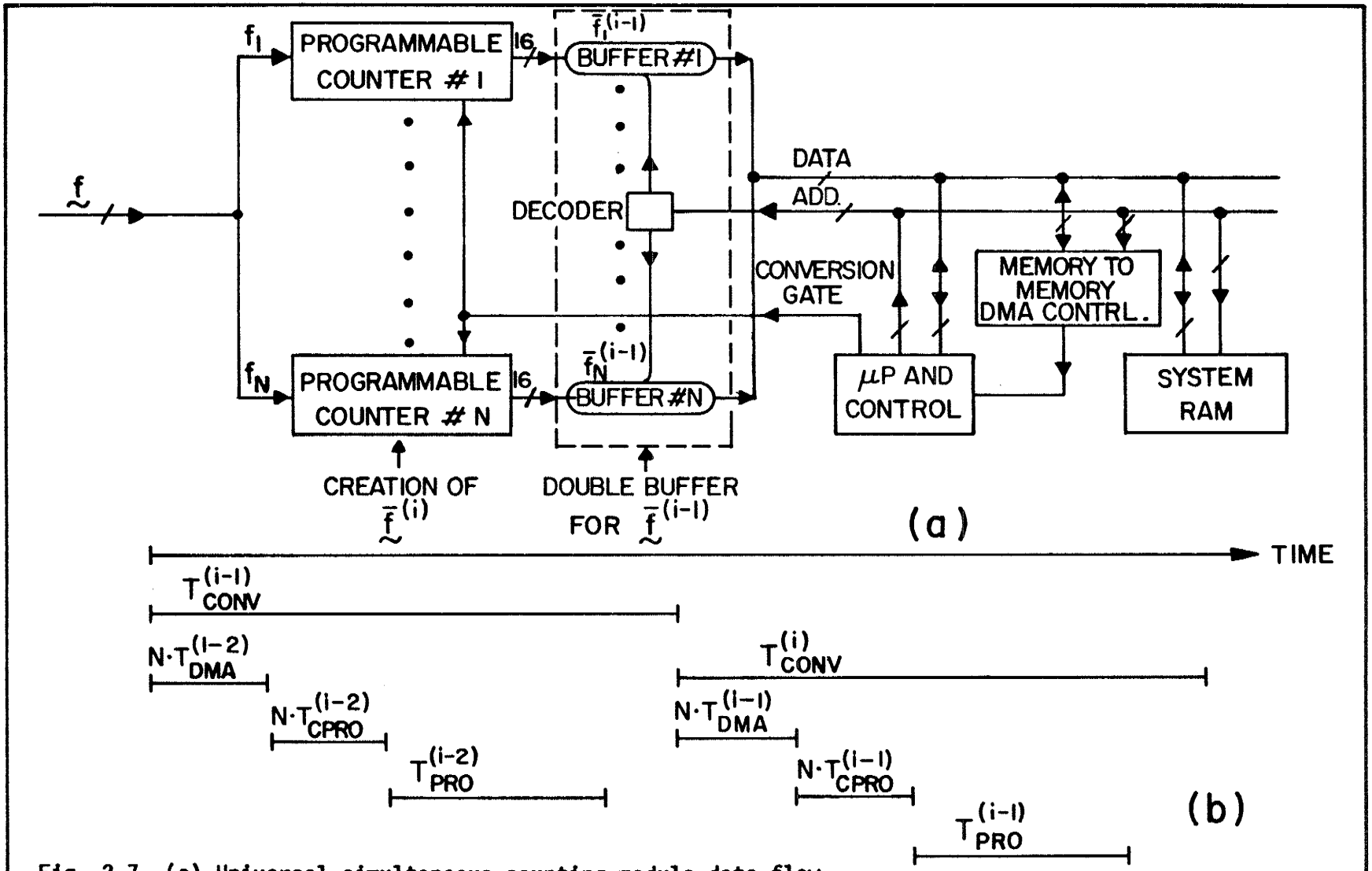


Fig. 2.7. (a) Universal simultaneous counting module data flow.
(b) Pipelined conversion and processing.

queueing sequence involves the following times: Conversion time (T_{CONV}), DMA cycle time per channel (T_{DMA}), processing time per channel (T_{CPRO}), and processing time per state vector (T_{PRO}). If the following condition holds

$$T_{CONV} > N \cdot T_{DMA} + N \cdot T_{CPRO} + T_{PRO} \qquad (2.1)$$

then the DMA and processing operations on a state vector $\underset{\sim}{f}^{(i-1)}$ can be pipelined with the conversion of the instantaneous state vector $\underset{\sim}{f}^{(i)}$.

2.5.3 USCM Throughput

The system throughput (ST) of a scalar, vector or array converter shall be defined as the number of converted and processed data sets. ST_S, ST_V, ST_A will denote the system throughput for a composite conversion and processing structure for the scalar, vector and array methods, respectively. For a pipelined USCM architecture, the system throughput ST_V can be expressed by

$$ST_V = \frac{1}{T_{CONV}} \qquad (2.2)$$

and is measured by the number of vectors per second (vector/sec). In order to compare the pipelined USCM ST_V to a MIDAS ST_S we have to use the following relation to obtain the equivalent throughput

$$ST_S = N \cdot ST_V \qquad (2.3)$$

For N = 252 and T_{CONV} = 0.02 sec. we obtain ST_V = 50 vector/sec and

ST_S = 12,600 channels/sec! Most of the MIDAS run at rates not exceeding 200 channels/sec.

2.5.4 Network Architecture

A four level hierarchical multiloop network architecture is shown in Fig. 2.8. The network has been implemented and is operational [5]. Each USCM monitors a radiation area (RA) and communicates with a minor loop controller via a centralized token passing protocol. Minor loops are associated with a section that is partitioned into RAs. A major loop controller collects the processed and reduced data through minor loop controllers for further processing in the supervisory computer. Although the system exhibits superior performance to any other existing MIDAS [5], it has the following drawbacks: (i) A computation bottleneck is created at the USCM for certain applications requiring large number of channels and extensive preprocessing; (ii) The USCM is not synchronized forming a data skew between each RA state vector; (iii) The physically large hybrid sensors limits very dense monitoring of RA; (iv) Three separate network entities (USCM, minor loop controller, and major loop controller) are required; (v) All vector oriented processing is performed via a uniprocessor and the network serves the communication role only. All these drawbacks can be removed or significantly reduced by a systolic network architecture presented in the following chapters.

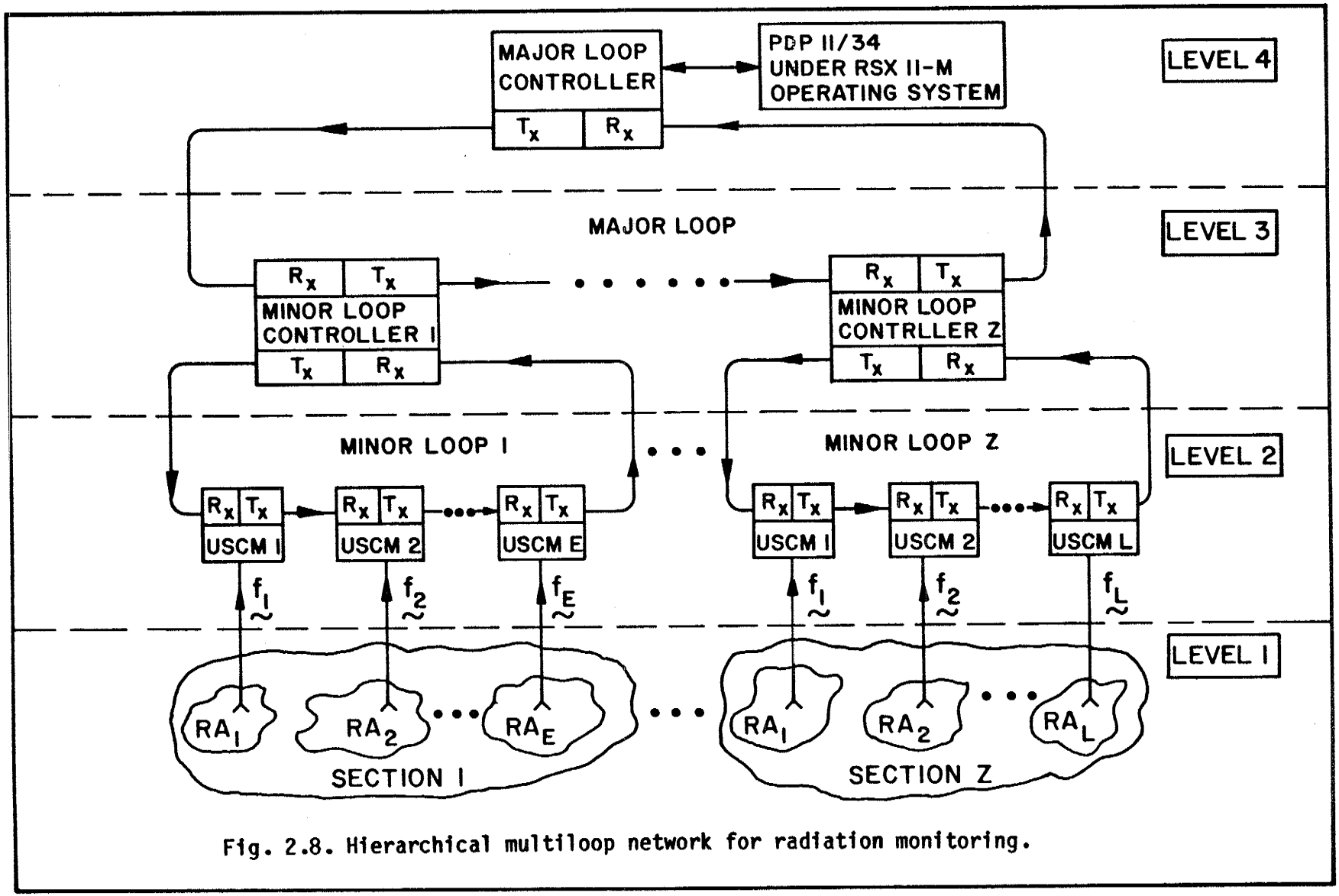


Fig. 2.8. Hierarchical multiloop network for radiation monitoring.

CHAPTER III

INTEGRATED SENSOR ARCHITECTURE

The impact of pulse stream fixed amplitude hybrid sensors (HS) on vector oriented DAS was discussed in the previous chapter. Three problems were identified as a result of the analysis of a USCM: (i) For a large number of channels, a computational bottleneck is created as a result of the linearization algorithm for each sensor; (ii) High accuracy HS are expensive; (iii) Miniature HS are not available, limiting dense sensor array applications in small MA; and (iv) Short and long term high repeatability (implying non-fatigue properties of the sensor) is extremely difficult to achieve, resulting in periodic calibrations that are costly and reduce the system availability.

Integrated sensor (IS) technology utilizes the mechanical and electrical properties of silicon for phenomena sensing and signal processing on a single monolitic chip. The mechanical properties of silicon suitable for phenomena sensing are covered extensively in [26]. Various technologies for design and fabrication of IS are described in [27], while the non-fatigue properties of silicon are addressed in [26]. A Foxboro IS for pressure measurement, tested from 0 to 10^4 psi at 40 Hz for over 5×10^9 cycles (4 years) without any noticable degradation, is reported in [26].

A valuable source of research in IS is provided in two special issues of IEEE Transactions on Electron Devices [28], [29]. Two publications [30], [31], describe a fixed amplitude pulse stream sensor for pressure measurement. Specific integrated sensors for radiation

detection are described in [32], [33].

Fabrication of such an IS for applications demanding high accuracy is a difficult task requiring costly laser trimming and elaborate automated testing. In this chapter we will develop a conversion/correction method that requires only highly repeatable IS, relying on the non-fatigue properties of silicon, thus further reducing the cost of IS. The usual nonlinear characteristics of the IS will be corrected as an inherent part of the conversion process utilizing a new vector oriented DAS architecture, the systolic converter. The systolic converter will eliminate the traditional barrier between sensor data conversion/correction and processing.

3.1 Optimal Integrated Sensor for Systolic Conversion

The proposed functional layout for an IS is shown in Fig. 3.1(a). It consists of a phenomenon sensor and an optional compensating phenomenon sensor (usually temperature), followed by an on chip tandem compensator, as described in [27], [31]. The signal from the tandem compensator is then converted to frequency by using a monotonic voltage to frequency converter (V→f) such as the charge balancing V→f converter described in [34]. Figure 3.1(b) shows the preferred symbol for such an IS. The last two characters denote the measured phenomenon (e.g. P for pressure) and the optional compensating phenomenon (e.g. T for temperature).

In general, the measured transfer function of the proposed IS

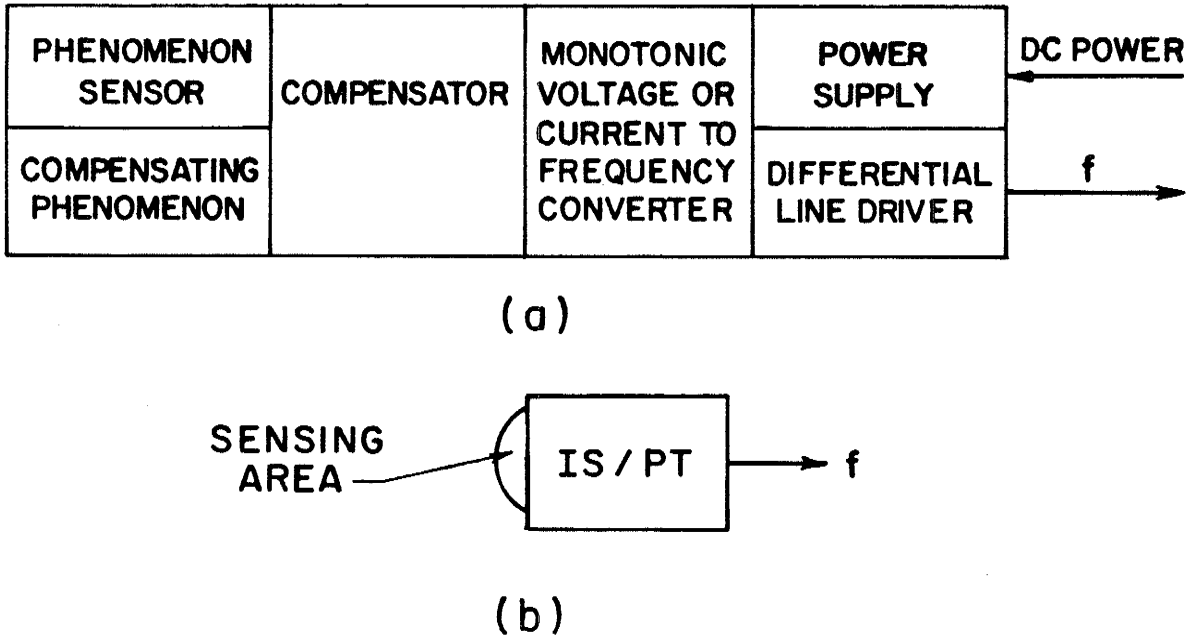


Fig. 3.1. Integrated sensor for a systolic converter. (a) Functional layout. (b) Symbol.

(phenomenon as function of frequency) may not be linear. To correct for the nonlinearity, a set of offset and gain parameters can be derived using optimal piecewise linearization parameters. In order to minimize the number of segments of the piecewise linearization, the V→f should be monotonic to reflect only the phenomenon sensing nonlinearities. However, optimization of the V→f monotonicity and the number of segments for a given accuracy are merely system design parameters.

The key properties of the proposed IS architecture are: (i) A highly accurate phenomenon measurement with quasi-linear, repeatable integrated sensors; (ii) Arbitrarily high resolution due to the monotonic characteristics of the V→f; (iii) Operation in the frequency mode, thus resulting in high noise immunity; (iv) Simple signal isolation through optical coupling; (v) No distinction between discrete and continuous measured phenomena; and (vi) Standardization of the output signal.

3.2 Integrated Sensor Linearization

Let us assume that the instantaneous state vector $\underset{\sim}{f}$ completely represents the monitored process. Each of the N elements of $\underset{\sim}{f}$ is generated by an IS and should correspond to the measured phenomenon P at a specific location. Since each IS may have a different transfer function P(f), the accurately converted and time-tagged vector $\underset{\sim}{\bar{f}}^{(i)}$ may not represent the monitored process. The following sections describe the method of obtaining the correction parameters to be used by the

systolic correction/conversion array to derive the true state vector representation. The proposed correction conversion method could be utilized in both microprocessor and systolic converter implementations.

3.2.1 Corrections for Uniprocessor Implementations

For the initial discussion, let us assume a linear IS transfer function (or one segment) with a gain and offset errors, as shown in Fig. 3.2(a). An ideal IS would transform the upper (P_{MAX}) and lower (P_{MIN}) measured phenomenon into the corresponding upper (f_{MAX}) and lower (f_{MIN}) frequencies. Instead, the non-ideal IS produces f_{MAX} and f_{MIN} corresponding to another matched pair of P_H, P_L on the non-ideal segment. It is sufficient to measure the values of P_H and P_L at the corresponding frequencies f_{MAX} and f_{MIN} for a sensor and use the values to compute the necessary correction. It is convenient to arrange all the measured (non-ideal) and desired (ideal) values in an array, called here a personality matrix, given by

$$\underset{\sim}{A} = \begin{bmatrix} P_L & f_{MIN} & P_{MIN} \\ P_H & f_{MAX} & P_{MAX} \end{bmatrix} \qquad (3.1)$$

Let us now derive the correction function that transforms a non-ideal IS to the desired ideal IS. Any point P on the non-ideal line segment (Fig. 3.2(a)) can be computed using a simple sum of the following two line segments

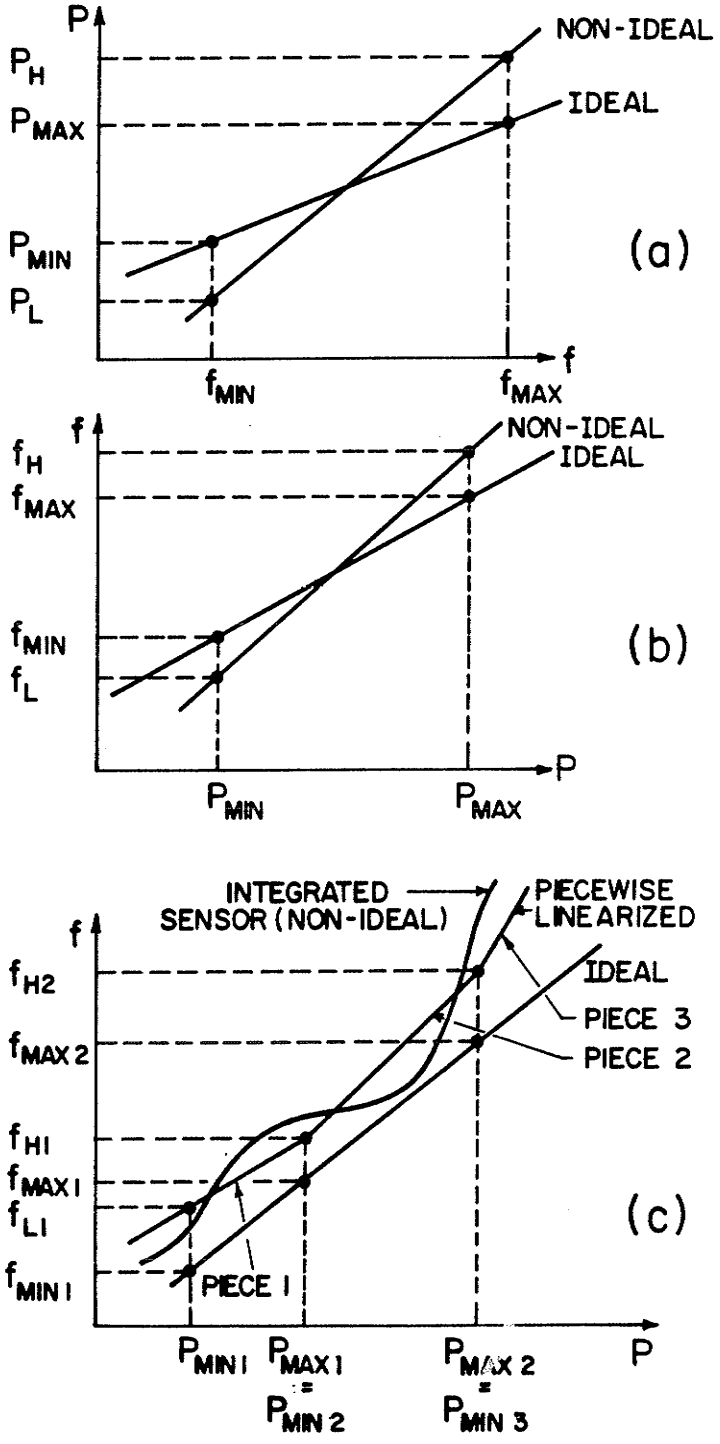


Fig. 3.2. Ideal and non-ideal characteristics of an integrated sensor. (a) For derivation of uniprocessor correction. (b) For systolic converter correction. (c) For generalized sensor correction.

$$P = \frac{f-f_{MIN}}{f_{MAX}-f_{MIN}} \cdot P_H - \frac{f-f_{MAX}}{f_{MAX}-f_{MIN}} \cdot P_L \qquad (3.2)$$

The point P can be corrected to the desired (ideal) value by a simple transformation (equivalent to a shift and rotation) given by

$$P^* = \frac{P - P_L}{P_H - P_L} \cdot P_{MAX} - \frac{P - P_H}{P_H - P_L} \cdot P_{MIN} \qquad (3.3)$$

Rearranging Eq. (3.3) yields

$$P^* = \frac{P_{MAX} - P_{MIN}}{P_H - P_L} \cdot P + \frac{P_H P_{MIN} - P_L P_{MAX}}{P_H - P_L} \qquad (3.4)$$

This form of transformation emphasizes the gain (slope) and offset calculated from the personality matrix elements for each sensor. These parameters must be computed and stored for the required correction of all state vectors.

In a uniprocessor implementation, such as the USCM, the correction process is performed by first measuring the personality matrix of each sensor (off line) and storing it in the local memory. For each measurement, the value of P is calculated from the available frequency f using Eq. (3.2) and corrected using Eq. (3.4). Since the correction is performed by one processor, there is an inherent computational bottleneck. The following subsection will expand the correction method for a systolic converter in order to remove the processing bottleneck.

3.2.2 Corrections for Systolic Converter Implementations

Figure 3.2(b) shows an ideal and non-ideal IS with a transfer function f(P). The IS is supposed to transform a set of upper (P_{MAX}) and lower (P_{MIN}) phenomenon into a set of upper (f_{MAX}) and lower (f_{MIN}) frequencies. Instead, for P_{MAX}, P_{MIN} we obtain a matched set f_H, f_L. It is again convenient to arrange the measured and desired values in a personality matrix $\underset{\sim}{B}$ as follows

$$\underset{\sim}{B} = \begin{bmatrix} f_L & P_{MIN} & f_{MIN} \\ f_H & P_{MAX} & f_{MAX} \end{bmatrix} \tag{3.5}$$

The duality between matrices $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is obvious and is obtained by exchanging f and P. The corrected frequency f^* can be computed from the measured frequency f, using a transformation similar to Eq. (3.3)

$$f^* = \frac{f - f_L}{f_H - f_L} \cdot f_{MAX} - \frac{f - f_H}{f_H - f_L} \cdot f_{MIN} \tag{3.6}$$

or after rearrangement

$$f^* = \frac{f_{MAX} - f_{MIN}}{f_H - f_L} \cdot f + \frac{f_H f_{MIN} - f_L f_{MAX}}{f_H - f_L} \tag{3.7}$$

It will be shown in Sec. 3.3 that the gain multiplication and

offset addition could be achieved as an integral part of the conversion scheme in a systolic converter.

For a more complex IS transfer function, such as that shown in Fig. 3.2(c), the personality matrix $\underset{\sim}{B}$ could be further expanded to achieve IS piecewise linearization. For a given IS, an automated measurement is performed to obtain the transfer function f(P) with sufficient accuracy, as shown in Fig. 3.3. This function is approximated by an optimal piecewise linearization program. The personality parameters for each segment are obtained, using Eq. (3.5) and the expanded matrix $\underset{\sim}{B}$ for L segments is given by

$$\underset{\sim}{B} = \begin{bmatrix} f_{L1} & P_{MIN1} & f_{MIN1} & f_{H1} & P_{MAX1} & f_{MAX1} \\ f_{L2} & P_{MIN2} & f_{MIN2} & f_{H2} & P_{MAX2} & f_{MAX2} \\ & & \vdots & & & \\ f_{LL} & P_{MINL} & f_{MINL} & f_{HL} & P_{MAXL} & f_{MAXL} \end{bmatrix} \qquad (3.8)$$

Note that $P_{MAXm} = P_{MINm+1}$ for $1 \leq m < L$. A matched pair of gain $\underset{\sim}{G}_j$ and offset $\underset{\sim}{S}_j$ vectors can now be calculated (see Eq. (3.7)) with the number of elements corresponding to the number of segments. The following section describes the use of these vectors in the conversion/correction array of a systolic converter.

The process described above could be divided into two phases: (i) Offline automated measurements to obtain f(P), $\underset{\sim}{B}$ and ($\underset{\sim}{S}_j$, $\underset{\sim}{G}_j$) for a given IS; and (ii) Real-time conversion correction when the offset $\underset{\sim}{S}_j$ and gain G_j vectors are used within the systolic conversion/correction cell as described in the following section.

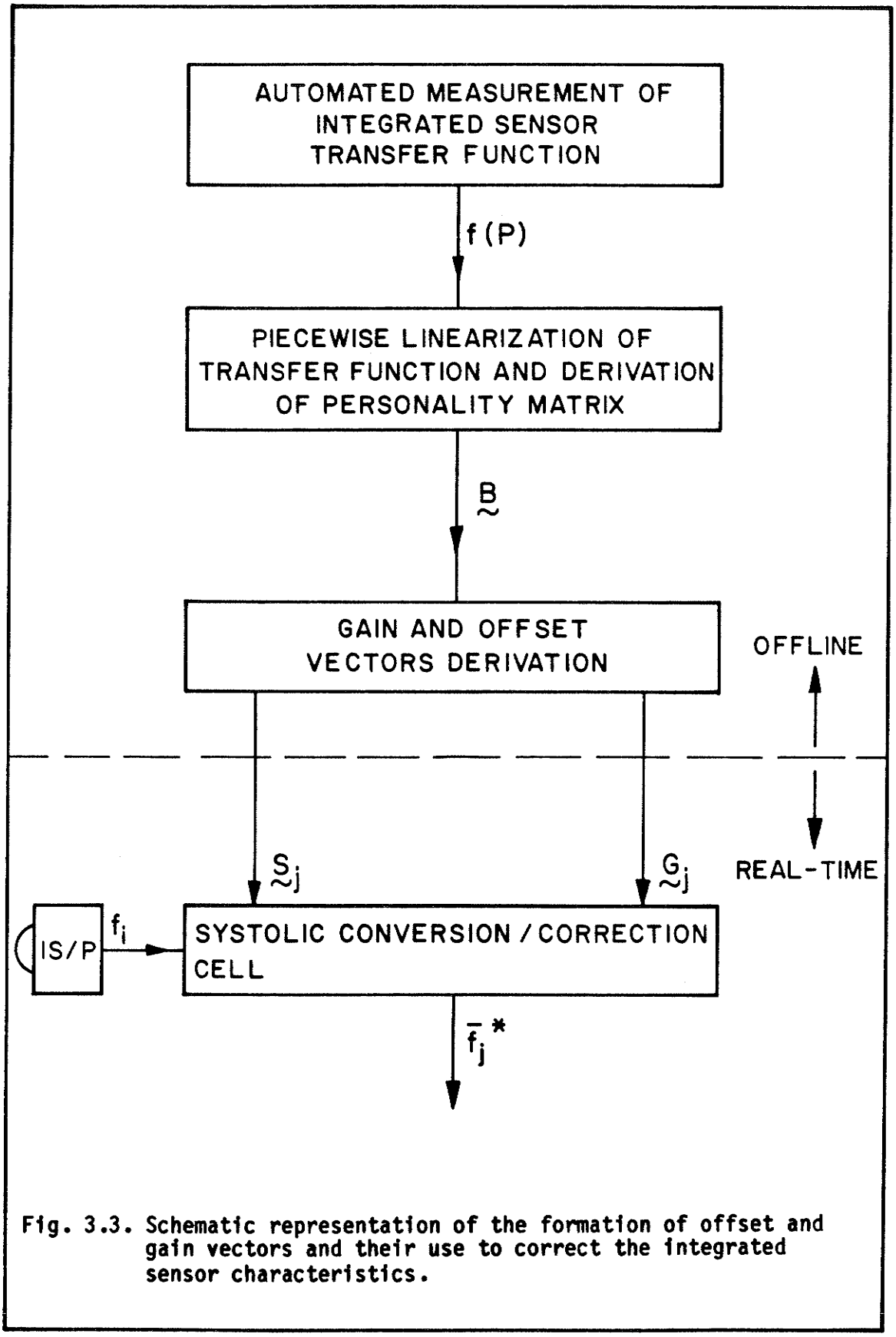


Fig. 3.3. Schematic representation of the formation of offset and gain vectors and their use to correct the integrated sensor characteristics.

3.3 Systolic Conversion/Correction Array

The conversion process using an averaging counter was described in Sec. 2.5.1 and served as the basis for the USCM. An embodiment of a composite real-time conversion correction cell that utilizes the derived $\underset{\sim}{S}_j$ and $\underset{\sim}{G}_j$ vectors for sensor linearization will be described in this section.

The IS on channel j, shown in Fig. 3.4, transmits an instantaneous frequency f_j which is counted within the conversion counter gated by the output of the conversion gate counter. The value in the conversion counter accummulated within a constant conversion time Δt_i represents the average pulse rate $\bar{f}_j$. Changing the conversion time-gate is equivalent to multiplication of the conversion counter result (gain compensation), while the initial value for the conversion counter will correspond to an addition/subtraction operation (offset compensation). The composite conversion/correction method is implemented using two first-in first-out (FIFO) buffers for storing the $\underset{\sim}{G}_j$ and $\underset{\sim}{S}_j$ vectors, each consisting of L elements derived from the L segment piecewise linearization process. The $\underset{\sim}{S}_j$ and $\underset{\sim}{G}_j$ vectors are loaded into the FIFOs at the initialization time and maintained there through a data recirculation path. The conversion/correction process is started by loading the conversion gate counter and the conversion counter with the corresponding gain and offset elements for the first segment. The conversion gate counter is also capable of generating an overflow

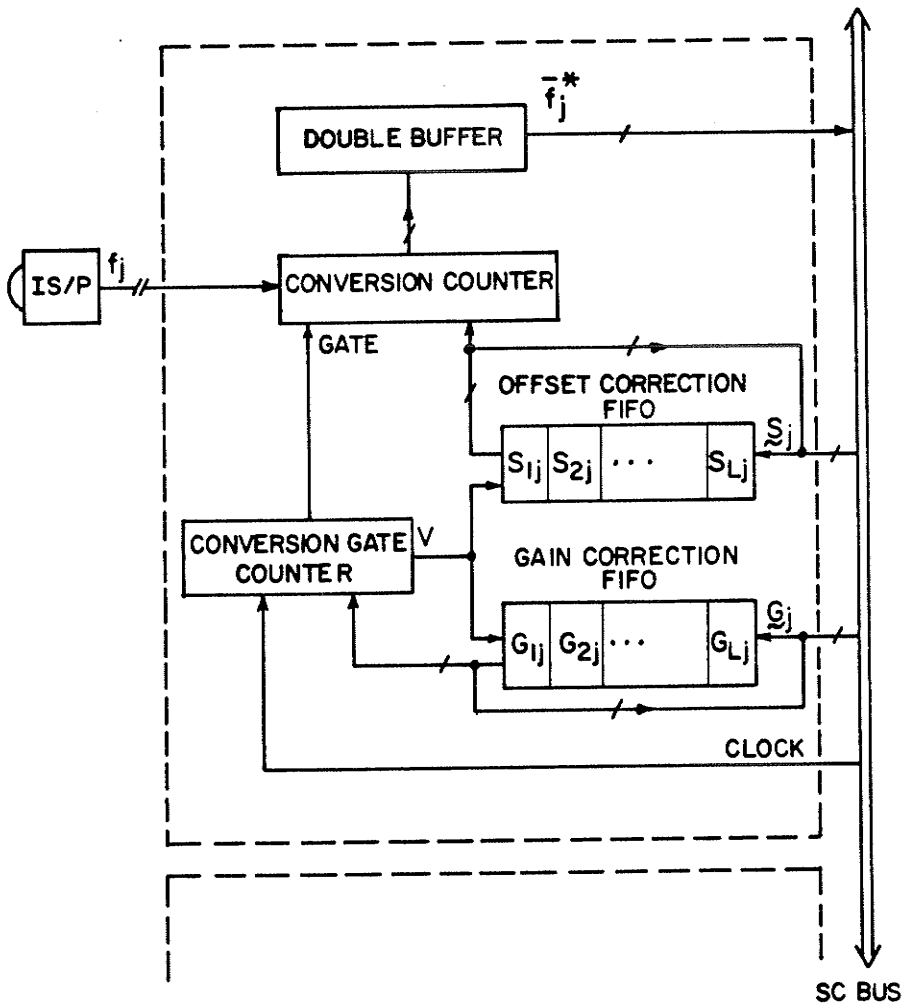


Fig. 3.4. Systolic conversion/correction cell.

signal (the segment time is proportional to the clock frequency and G_{mj} for $1 \leq m \leq L$) in order to extract the next gain and offset correction elements. The process is continued for L segments resulting in a corrected measurement $\bar{f}_j^*$ in the double buffer. The FIFOs are maintained at the end of the conversion/correction process at their initial state, signified by the data feedback path around both FIFOs.

The conversion/correction process results in a slight variation in the total conversion period which depends on the non-linearity of the specific IS. Since all conversion/correction cells are activated simultaneously via a conversion systole (T_{cs}), the data elements representing $\underset{\sim}{\bar{f}}^*$ will appear slightly desynchronized. The double buffer will serve as a pipeline register to assure the data integrity of $\bar{f}^*$. The conversion systole time must be defined as the inverse of the rate of converted state vectors $\underset{\sim}{\bar{f}}^*$. This definition assures that the vector conversion time for a specific set of IS with the associated $\underset{\sim}{G}_j$ and $\underset{\sim}{S}_j$ vectors is less than the conversion systole. The conversion and processing of the state vectors are pipelined to maximize the systolic converter throughput, as in the lumped USCM implementation.

The conversion/correction method could also be viewed as a continuous filtering process of $\underset{\sim}{\bar{f}}$ into $\underset{\sim}{\bar{f}}^*$, performed concurrently with the filtering that results from the averaging process. It should be noted that the two processes affect each other. Usually the conversion gate will be set to one mains cycle ($\frac{1}{60}$ sec) in order to obtain an infinite normal mode rejection ratio (NMRR) of 60Hz induced noise due

to the comb filter averaging effect. The normal mode rejection ratio (NMRR) will be slightly degraded as a result of the non optimal conversion time.

We can summarize the key advantages of the proposed IS and conversion/correction architecture as follows: (i) Highly accurate phenomena measurement with only quasi-linear repeatable IS; (ii) Arbitrary high resolution obtained by the IS architecture and the control of the conversion gate; and (iii) A generic family of IS with the properties described in Sec. 3.1. The sensing conversion/correction architecture also removes the computational bottleneck associated with linearization and filtering of sensor signals. The problem of state vector acquisition and processing is addressed in the following chapter.

CHAPTER IV

SYSTOLIC CONVERSION AND PROCESSING METHOD

In order to facilitate the description of the systolic conversion and processing method, we shall first review the basics of systolic arrays and their generalizations. This review should also emphasize the difference between our work and the work reported elsewhere.

In his fundamental work [35], Kung identified three orthogonal dimensions of the space of parallel algorithms: (i) Concurrency control; (ii) Module granularity; and (iii) Communication geometry. Important positions that parallel algorithms can assume along each dimension, are described in three hierarchical tree structures. Let {concurrency controls}, {modules granularities}, and {communication geometries} be the sets of leaves of the respective trees. The cross product {concurrency control}x{module granularities}x{communication geometries} represents the space of parallel algorithms. A systolic algorithm subspace refers to the algorithm subspace, where concurrency control is distributed, synchronous and simple at the module level, as well as module granularity is of small constants. The architectures for the embodiment of systolic algorithms are called systolic architectures. The word "systole" was originally borrowed by Kung from physiologists who use it to refer to the rhythmically recurrent contractions of the heart and arteries that pulse blood through the body. A systolic architecture will consist of small processors each "pumping" data in and out, and performing some type of computation to maintain the data flow in the network.

As Kung pointed out [35], the subspace formed by {systolic} x {communication geometries} is of great importance. If the communication geometry is a one dimensional array, a variety of algorithms results, such as matrix-vector multiplications, convolution, finite impulse response (FIR), and adaptive filtering, as well as other real-time signal processing algorithms. Similarly, if the communication geometry is a two dimensional square or hexagonal array, a variety of algorithms results such as matrix-matrix multiplication, image processing, discrete Fourier transform (DFT), and numerical relaxation, as described in [36], [37], [38].

Although systolic architectures are naturally identified with VLSI implementation [37], they are not confined to it [38]. In this work, we shall mostly be confined to the {systolic}x{linear array} subspace.

Simultaneous conversion and systolic array processing provide a good match for vector oriented process monitoring. However, they should not be used indiscriminately because of the following drawbacks: (i) Corrections and filtering of sensor data have to be performed as part of the processing algorithms within the systolic array, thus reducing system throughput; (ii) The systolic design rules apply only to the local monitoring problem (over an MA) and not to the global monitoring problem (over the GMA). This chapter will describe the aggregate conversion and processing method resulting in a new systolic converter architecture. The architecture will be further expanded to a systolic network, following the same set of unified systolic architecture design rules, as described in Chapter V.

4.1 Systolic Converter Overview

Figure 4.1 illustrates the fundamental functional partitioning of a systolic converter [8]. The blocks should not be viewed simply as lumped subsystems, clearly separated in space and time, because for some applications that separation may vanish and the system becomes fully integrated. Other applications, such as the shut-down system described in Sec. 4.4, may be very similar to the general partitioning of Fig. 4.1.

The conversion/correction (CC) array consists of N conversion/correction (CC) cells, each attached to a single input channel carrying a fixed-amplitude discrete pulse stream. The CC array is controlled by a conversion systole (T_{cs}), and produces a state vector. The state vector is then transferred to a systolic processor array. The array requires various parameters for processing. The parameters are supplied from the parameter arrays, preloaded during system initialization. The processing is controlled by a processing systole (T_{ps}) and produces the desired output channels. Data transfers over the network are controlled by the network systole (T_{ns}). The latter may not be required for some applications. Nevertheless, the system must be synchronized globally and locally.

4.2 Systolic Converter Signal and Data Flow

4.2.1 Definitions

We shall now define a state vector (data vector) for a monitored

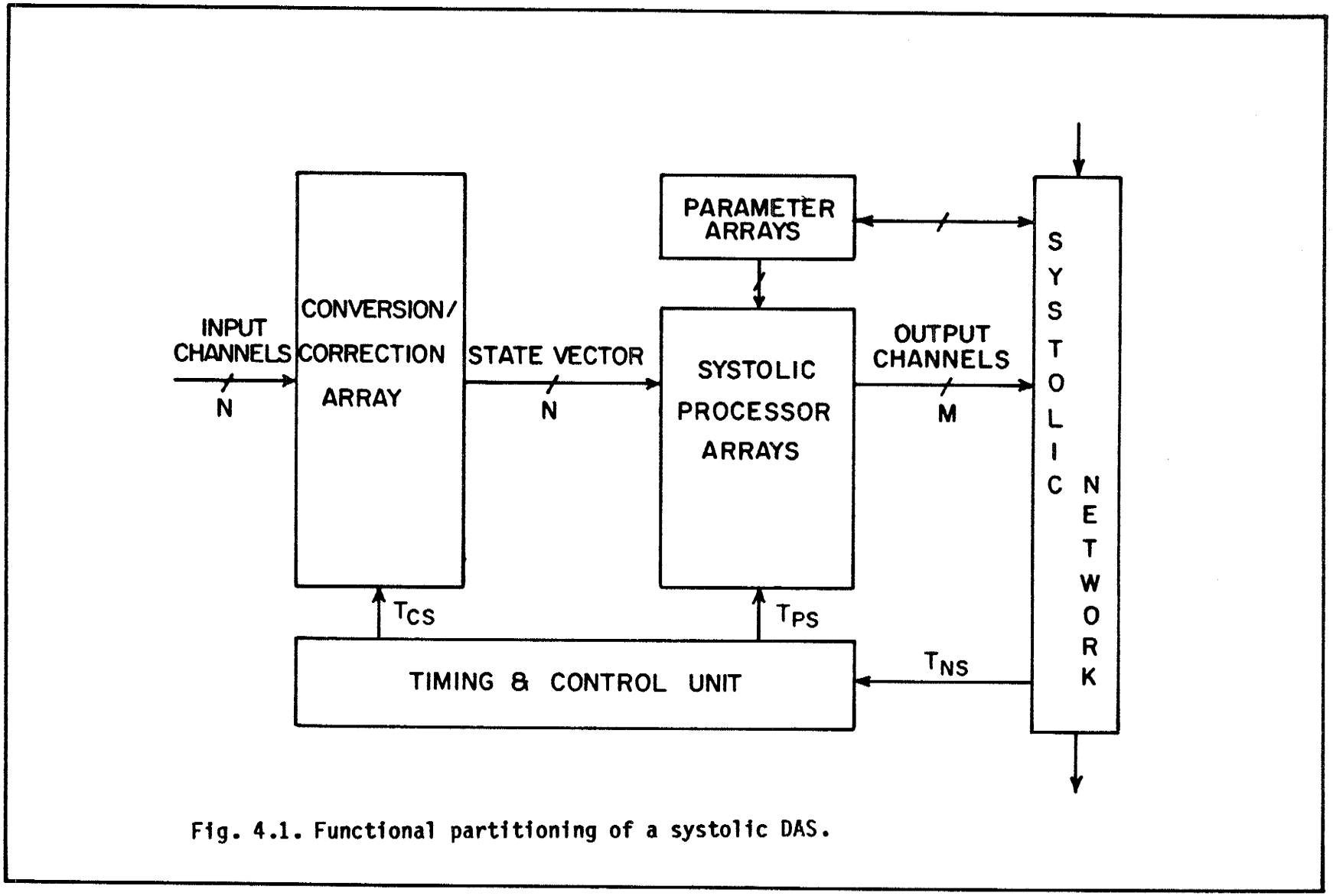


Fig. 4.1. Functional partitioning of a systolic DAS.

area (MA) with N channels in such a way that the same formulation could apply to any pulse-stream measured phenomenon. Let us assume that each channel contains only one phenomenon pulse source (one pulse stream integrated sensor). Each integrated sensor (IS) on channel j provides a fixed amplitude pulse stream, f_j, whose instantaneous rate depends on the measured phenomenon. The set of all f_j signals at any time t constitutes an instantaneous state vector, $\underset{\sim}{f}$, given by

$$\underset{\sim}{f} = [f_1, f_2, \ldots, f_N]^T \qquad (4.1)$$

where T denotes transposition of the column vector. The average pulse rate $\bar{f}_j^{(i)}$ generated on channel j within a time interval Δt_i is given by

$$\bar{f}_j^{(i)} = \frac{1}{\Delta t_i} \int_{t_i - \Delta t_i}^{t_i} f_j dt \quad ; \quad \forall j \in \{1,2,3,\ldots N\} \qquad (4.2)$$

where t_i is the reference time and Δt_i is given by

$$\Delta t_i = t_i - t_{i-1} \qquad (4.3)$$

This column state vector for all N channels at time t_i may also be corrected to compensate for gain, drift and nonlinearity errors of each IS, as described in Sections 3.2.1 and 3.2.2. The corrected state vector is defined as

$$\underset{\sim}{\bar{f}}^{*(i)} = [\, \bar{f}_1^{*(i)}, \bar{f}_2^{*(i)}, \ldots, \bar{f}_N^{*(i)} \,]^T \qquad (4.4)$$

where * denotes the required correction operation, as described in Sec. 3.2 and 3.3.

Finally, the state history of the MA over a period $\Delta T = t_{i+k} - t_{i-1}$ can be fully represented by a queued state matrix $\underset{\sim}{F}^{(k,i)}$ defined as

$$\underset{\sim}{F}^{(k,i)} = [\underset{\sim}{f}^{*(i+k)}, \underset{\sim}{f}^{*(i+k-1)}, \ldots, \underset{\sim}{f}^{*(i)}] \qquad (4.5)$$

4.2.2 Signal and Data Flow

Figure 4.2 is another form of the functional partitioning of Fig. 4.1 and is intended to graphically summarize the operation of such a systolic converter. Data are acquired from a remote cluster of integrated sensors, each transmitting a stream of pulses with the instantaneous rate of f_j. The integrated sensors are identified by a symbol with a tag IS/P where P specifies the measured phenomenon. The instantaneous vector $\underset{\sim}{f}$ is converted to a non-skewed, averaged, corrected and time-tagged vector $\underset{\sim}{f}^{*(i)}$ which, in turn, is queued, processed, and transferred to other subsystems using systolic circuits [39].

We can clearly see that the systolic conversion and processing concept provides a highly pipelined vector type conversion and processing architecture that removes the traditional barrier between data conversion and processing.

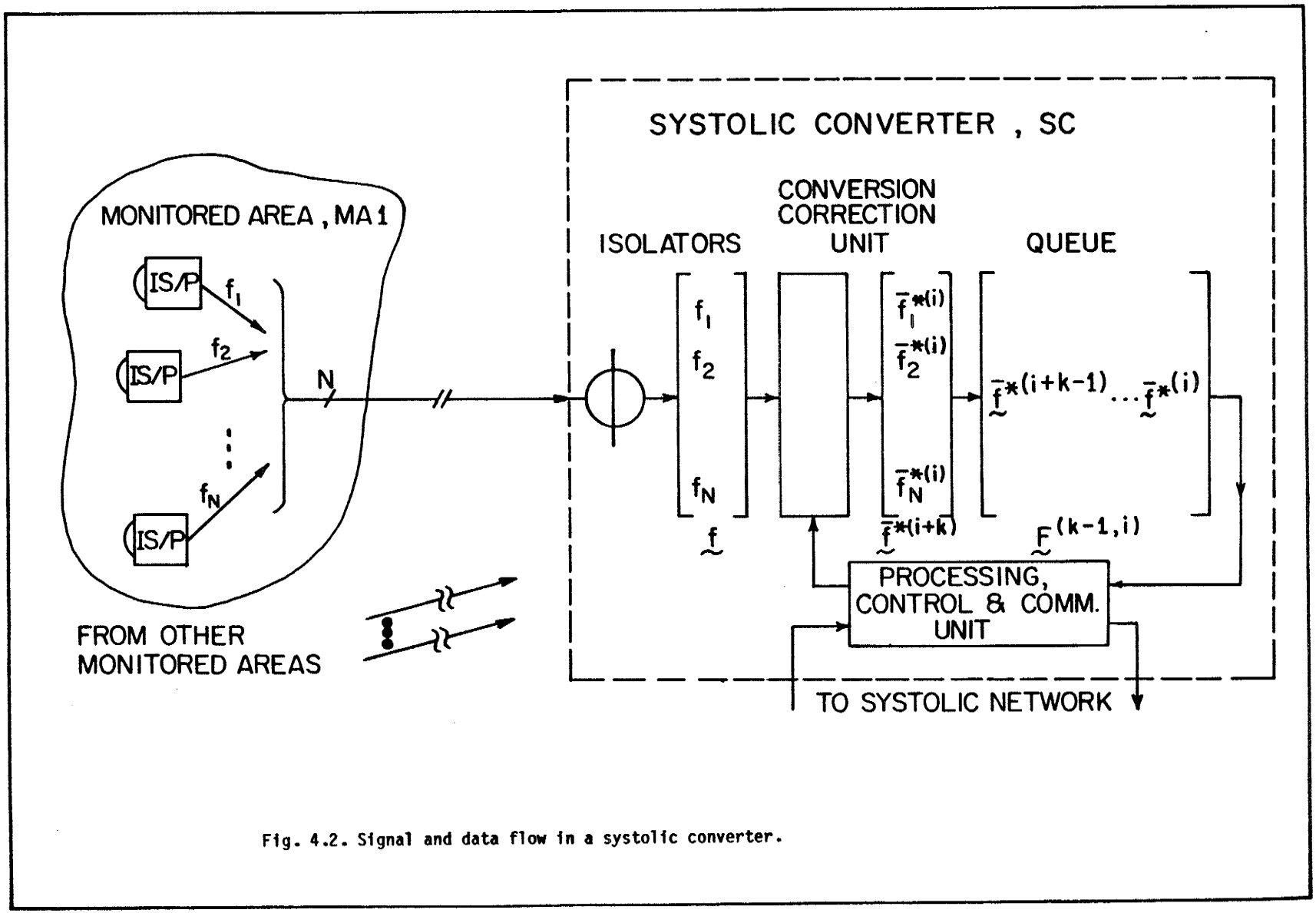


Fig. 4.2. Signal and data flow in a systolic converter.

4.3 Systolic Converter Throughput and Pipelining

The systolic converter (SC) throughput is defined as the number of state vectors that are converted and processed per second. The inherent pipelining of conversion and processing resulting from the vector queue within the SC dictates that the total pipelined processing time of a state vector $\underset{\sim}{f}$ cannot exceed the conversion period Δt.

This statement is also true for a microprocessor-based USCM which can be viewed as a uniprocessor systolic converter as described in Sec. 2.5.3. Since the processing is performed on the state vector elements according to the specific systolic rules of a selected problem, it affects the throughput of the SC. We will derive the measure of the SC throughput as a function of the systolic array processing architecture, number of vector elements and conversion time.

Let us define a conversion systole, T_{cs}, to be the inverse of the rate of converted state vectors $\underset{\sim}{\bar{f}}^*$, and the synchronous systolic array clock as the processing systole, T_{ps}. For a specific problem or classes of problems, the processing time, T_p, can be calculated from

$$T_p = Q \cdot T_{ps} \tag{4.6}$$

where Q is an integer which is a function of the number of vector elements, N, as well as the number of processors, n, within the array,

the level of pipelining, and the systolic array geometry, as described in [35]. This functional dependence can be denoted by

$$Q = f(N, \bullet) \tag{4.7}$$

Consequently, the upper bound for the systolic converter vector throughput (SCT_v), is given by

$$SCT_v = \frac{1}{T_{cs}} \text{ [vectors/sec]} \tag{4.8}$$

Since the converter systole is chosen as a function of the required averaging period, it limits the number of channels that could be handled by the systolic converter according to the following relation

$$T_{cs} > T_p = Q \bullet T_{ps} \tag{4.9}$$

or

$$SCT_v < \frac{1}{Q \bullet T_{ps}} \tag{4.10}$$

From Eq. (4.10) we can derive the number of elements in the state vector, given the proper design parameters, as illustrated by the example below.

Figure 4.3 illustrates the processing and conversion pipelining and the conversion (T_{cs}) and processing (T_{ps}) systoles. Let us

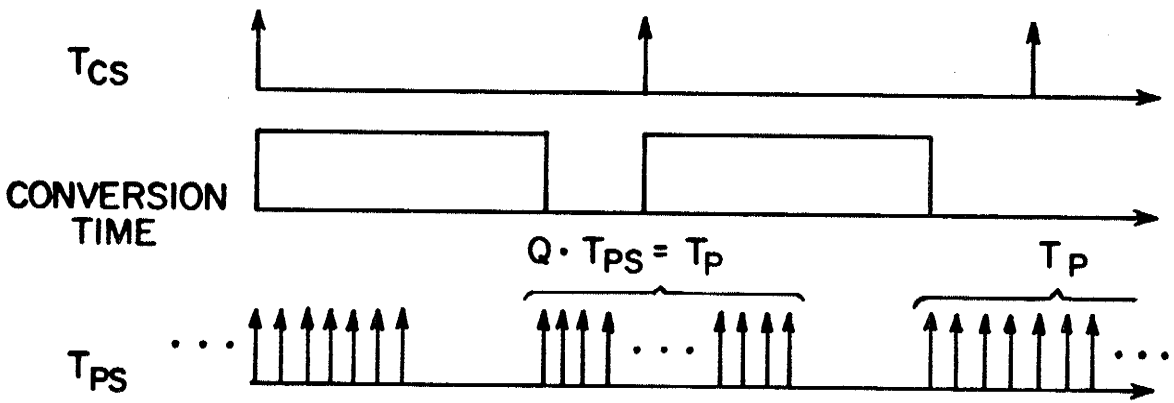


Fig. 4.3. Timing of conversion and processing systoles.

consider a systolic converter with a conversion systole of $T_{cs} = 0.1$ sec and a processing systole of $T_{ps} = 10^{-5}$ sec. The state vectors with N elements are multiplied by an NxN band limited matrix of band width w. As shown in [35], $Q = 2N + w$ for a linear systolic array of w processors. In this case, Eq. (4.10) becomes

$$SCT_v < \frac{1}{(2N + w) \cdot T_{ps}} \tag{4.11}$$

For example, for w = 4, the systolic converter vector throughput is

$$10 < \frac{1}{(2N + 4) \cdot 10^{-5}}$$

or

$$N < 5 \cdot 10^3$$

It is seen that the equivalent rate of a conventional scalar DAS will be given by $N \cdot SCT_v$ or approximately 50,000 channels/sec! This throughput can be achieved because there is no distinction between the conversion and processing in this unified systolic structure.

The following section will describe how these properties could be utilized in a systolic converter architecture suitable for shut-down of a CANDU reactor.

4.4 Analysis of a Systolic Shut-Down System

This section analyzes a systolic, integrated sensor oriented field mapping architecture for shut-down of a critical process (e.g., a nuclear reactor). The architecture is further described in the case analysis of a CANDU reactor presented in Chapter VIII and IX.

4.4.1 Data Flow of a Systolic Shut-Down System

A phenomenon field (e.g. neutron flux in a reactor core) is mapped by a cluster of integrated sensors with a spatial distribution optimized for the specific application geometry. The order of the elements in the vector $\underset{\sim}{f}$ corresponds to the sensor location. That time-space information is essential for the proposed architecture. Figure 4.4 illustrates the data flow of the shut-down system. An integrated sensor mapped area provides a vector $\underset{\sim}{f}$ which is converted, corrected and time-tagged within the conversion/correction array. The array utilizes the systolic conversion/correction cells for all N elements of the vector $\underset{\sim}{f}$. We observe that for each IS on channel j we have two associated vectors $\underset{\sim}{G}_j$, $\underset{\sim}{S}_j$ (each of L elements), resulting in two corresponding offset $\underset{\sim}{S}$ and gain $\underset{\sim}{G}$ matrices consisting of NxL elements.

The conversion/correction process performed systolically on all N vector elements results in a vector $\underset{\sim}{f}^{*(i)}$ which is multiplied in the weighting unit by a weighting matrix $\underset{\sim}{W}$. The elements in $\underset{\sim}{W}$ reflect the spatial weight of a specific sensor and the coupling with other

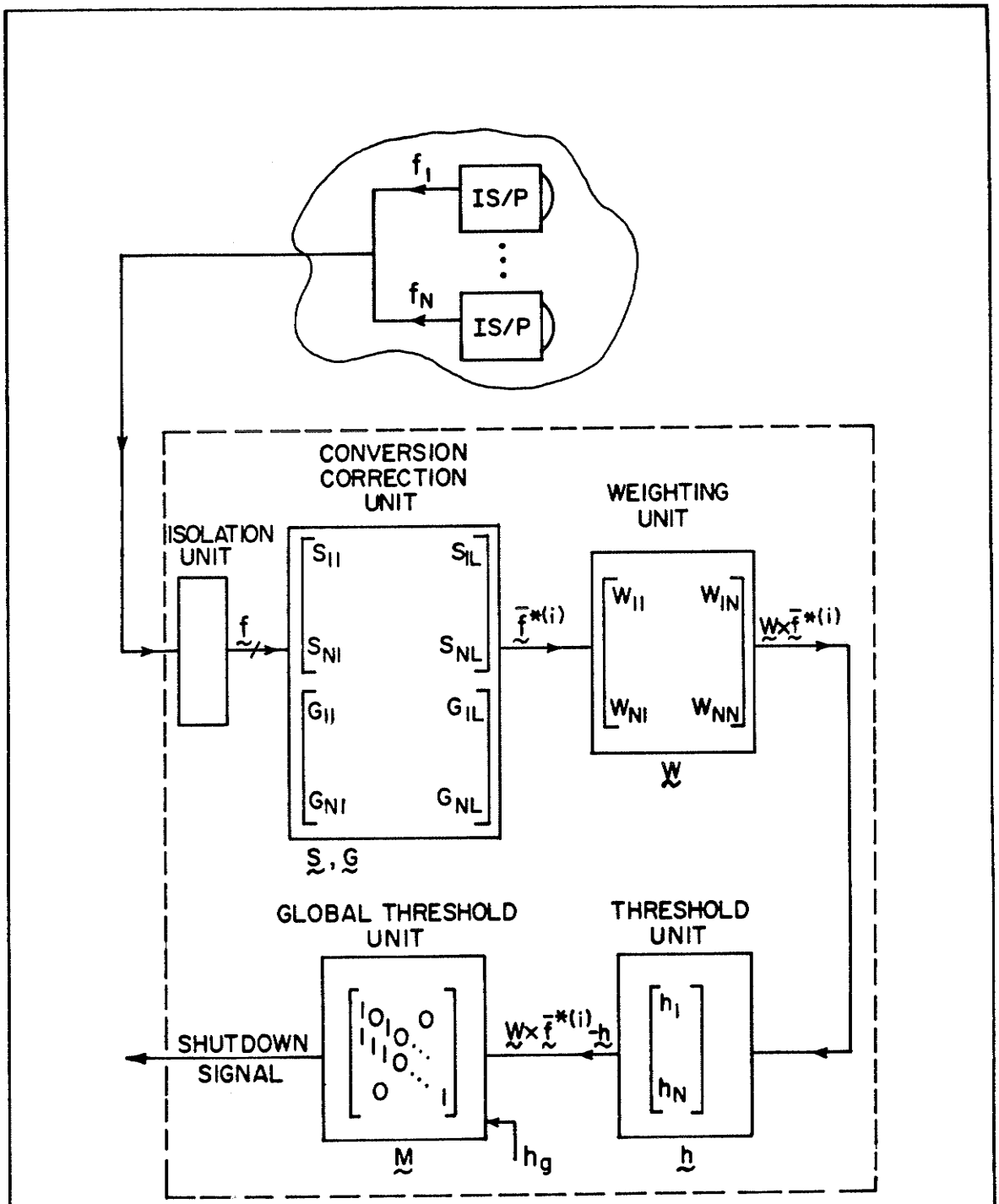


Fig. 4.4. Data flow of a generalized systolic shut-down system.

sensors.

The threshold unit subtracts a threshold vector $\underset{\sim}{h}$ from the weighted vector $\underset{\sim}{W}x\bar{\underset{\sim}{f}}^{*(i)}$. The global threshold unit multiplies the output vector of the threshold unit by a mask matrix $\underset{\sim}{M}$. The resulting vector elements are compared with a global threshold set point h_g, and the global shut-down signal is generated.

The shut-down system described above enables pipelined vector oriented field mapping and processing in a unified systolic structure. The approach allows the designer of a systolic converter to integrate the data acquisition problem with the data processing problem under a unified systolic structure. This systolic architecture for a shut-down system is described in the following section.

4.4.2 Systolic Shut-Down System Architecture

The architecture of a shut-down system is shown schematically in Fig. 4.5 and is similar to the general structure of Fig. 4.1. The conversion/correction unit is triggered by a conversion systole (T_{cs}) generated from the timing and control unit. The resulting vector $\bar{\underset{\sim}{f}}^{*}$ is held in a swinging FIFO structure to obtain the pipelining of conversion and processing, as analyzed in Section 4.3. The FIFO structure is implemented utilizing a dual-pointer RAM-based FIFO controller [40]. This structure views the double buffer within the conversion/correction unit as contiguous locations in memory using

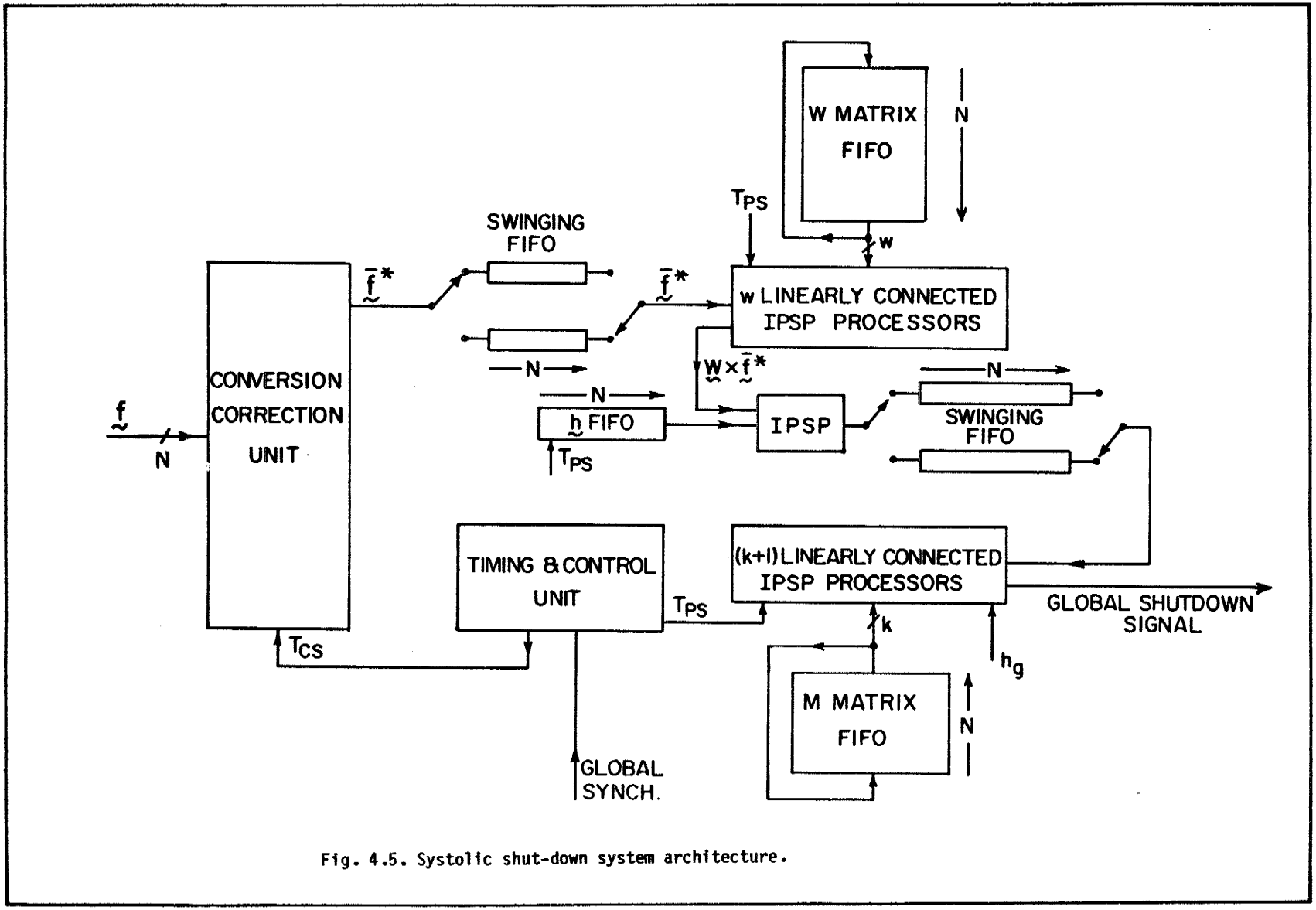


Fig. 4.5. Systolic shut-down system architecture.

pointer manipulations instead of shift registers.

The matrix vector multiplication is performed systolically by a linearly connected array of inner product step processors (IPSP), following the method described in [37]. For a weighting matrix $\underset{\sim}{W}$ with a band width of w, the computation can be completed by the IPSP array in 2N+w processing systoles (T_{ps}). The resulting vector is subtracted from the associated threshold vector, pipelined with the $\underset{\sim}{W}\underset{\sim}{x}\underset{\sim}{f}$ process, and is available at the swinging FIFO for multiplication with the mask matrix $\underset{\sim}{M}$. The matrix elements are held in the proper order in the $\underset{\sim}{W}$ matrix FIFO which contains NxW elements. The FIFO output is fed back to the input to assure data integrity for the next vector multiplication.

The output of the threshold unit implemented by one IPSP is placed in the swinging FIFO and multiplied by a matrix $\underset{\sim}{M}$ using the method described earlier. The matrix $\underset{\sim}{M}$ of band width k is fed in the proper order by the $\underset{\sim}{M}$ matrix FIFO into the k linearly connected IPSPs, and synchronized by T_{ps}. The net result is compared to h_g by the k+1 IPSP to obtain the shut-down signal.

For a typical neutron flux measurement in a reactor core we have N=50, a matrix band width w=20, and a conversion systole T_{cs} to be less than 20 msec. From these values we derive Q=2•50+20=120. The processing time (T_p) must be less than 20 msec. Using Eq. (4.9) we derive that $T_{ps} < \frac{T_{cs}}{Q} \cong 166$ μsec. The result shows that a slow IPSP (which improves the overall system noise immunity) is adequate. An ultra

reliable system is obtained by triplicating the whole structure (including sensors) and adding majority logic to the shut-down signals as described in Chapter VIII.

The key architectural features of the systolic shut-down system could be summarized as follows: (i) Pipelined conversion correction and processing; (ii) Pipelined systolic processing; (iii) Serial data path between conversion/correction units and computational arrays utilizing a RAM based FIFO structure; (iv) The IPSPs are relatively slow, enhancing the overall system noise immunity; and (v) The global synchronization and the extreme data reduction rate enable high reliability redundant structures.

The systolic shut-down system is further expanded to monitor a critical process over the GMA through a systolic network, as described in Chapter VIII.

4.5 Strobed Systolic Converter

In many repeatedly triggered processes such as pulsed lasers, fusion, plasma and accelerator experiments, an intense electromagnetic interference (EMI) pulse is generated. Although the duration of the pulse and its delay from the process trigger are known, a MIDAS system monitoring relatively slow process parameters such as temperature and pressure will confront the following problems: (i) The scanning process is asynchronous with the EMI pulse, thus noise is injected to various channels randomly; (ii) Sensors may be saturated, requiring nonlinear

digital filtering on all scanned channels; and (iii) The isolation is centralized exposing the analog signal wires to severe EMI. The strobed systolic converter architecture described below, eliminates this drawback while pointing to a new property of the systolic conversion method.

4.5.1 Strobed Systolic Converter Timing

Figure 4.6(a) represents the timing diagram of a triggered process monitored via a strobed systolic converter. The EMI pulse is confined to a period indicated by a logic signal, the EMI strobe. During the EMI strobe, counting operations within all systolic conversion/correction cells are inhibited, eliminating any noise induced counts. Counting is resumed at the end of the EMI strobe. The total effective conversion/correction time remains unchanged. The method described above eliminates sensor-induced noise completely without any post processing requirements as with a MIDAS. The method points to the vector filtering capabilities through a common conversion gate, as opposed to scalar filtering on a MIDAS.

4.5.2 Hybrid Sensor with Fiber-Optic Output

The hybrid sensor architecture depicted in Fig. 4.6(b) utilizes an IS for phenomena measurement, followed by a digital fibre-optic link. The IS has to possess the following properties: (i) The input stage should be protected to sustain the induced EMI noise without damage; and (ii) The IS has to recover (if in saturation) at a relatively short

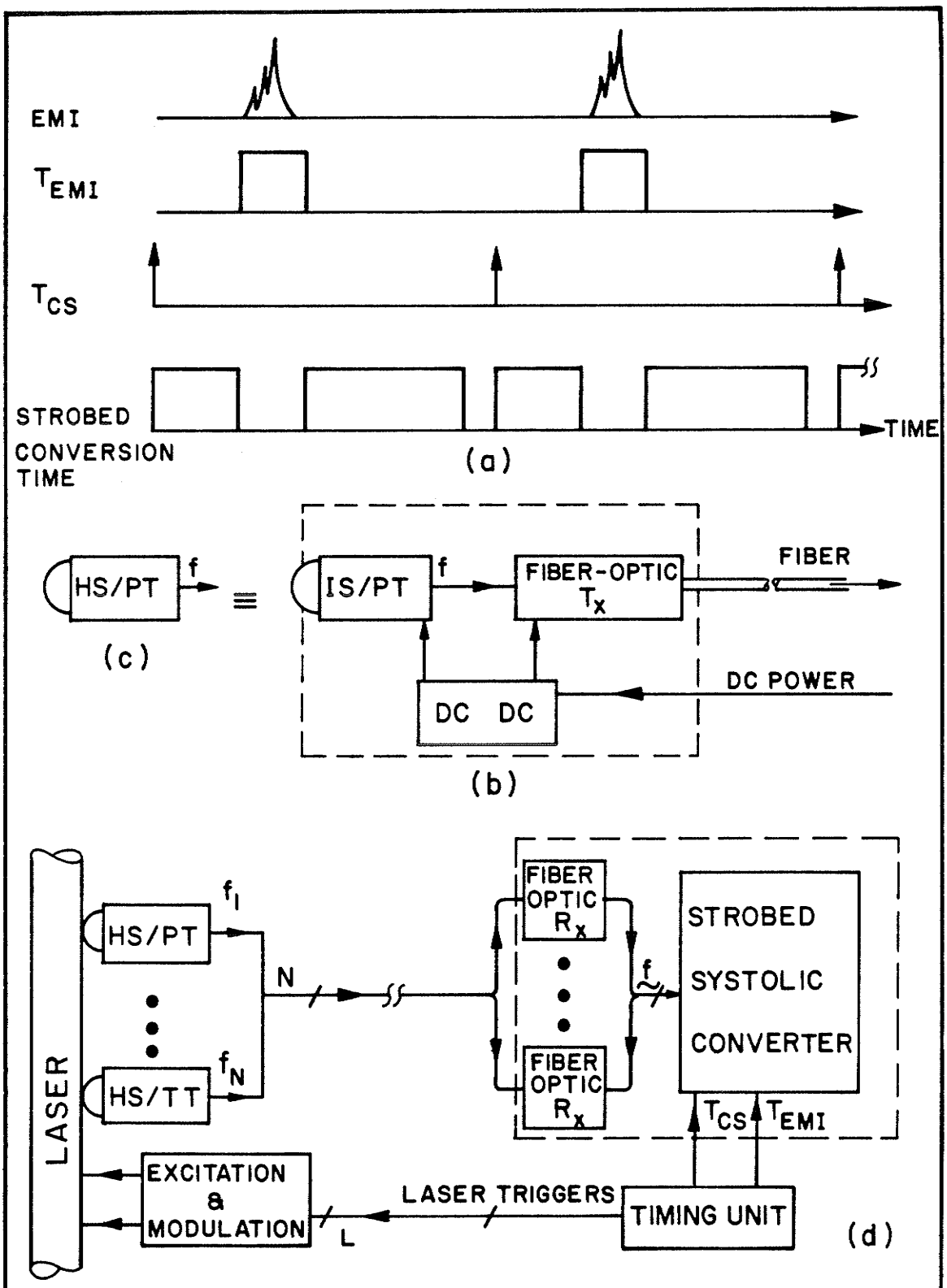


Fig. 4.6. (a) Strobed systolic converter timing. (b) Hybrid sensor with fiber-optic output. (c) Hybrid sensor symbol. (d) Strobed systolic converter for a pulsed laser control system.

period. The EMI strobe will be adjusted to inhibit all systolic conversion/correction cells during the EMI pulse and HS recovery time. The advent of integrated optoelectronics [41] can provide the technology for an IS with on chip sensing and optical transmission blocks.

4.5.3 Strobed Systolic Converter for Pulsed Laser Applications

A pulsed laser monitored via a strobed systolic converter is shown in Fig. 4.6(d). An array of HS for pressure and temperature measurements is transmitting light pulse streams into a fibre bundle. N fibre optic receivers convert the light pulse streams into electrical pulses. The timing unit generates the conversion systole (T_{CS}) and the EMI strobe (T_{EMI}) for the strobed systolic converter (SSC), as well as the trigger signals for the excitation and modulation of the pulsed laser. The noise-free state vector is further processed following a specific control algorithm.

The strobed systolic architecture exhibits the following properties: (i) Simultaneous noise gating on all channels that eliminates further state vector filtering; (ii) Enhanced system throughput as a direct result of (i); and (iii) Very high EMI immunity as a result of the hybrid sensor and noise gating.

It should be noted that if the EMI pulse is common to the GMA, such as in large fusion or accelerator experiments, a strobed systolic network could be devised by transferring the strobe timing via an

extension field of the network systole token.

The strobed systolic conversion method is based on the coherency between the triggered process and the conversion systole in the time domain. This concept could be further expanded to the frequency domain, as shown in the next section.

4.6 Coherent Systolic Conversion Method

The basic method of coherent detection is described in [4], pp. 506-511, and was used in many application areas. This section will describe a new coherent systolic conversion method that relies on the time, phase and frequency coherency between the actively measured process and the conversion systole.

4.6.1 Integrated Coherent Sensor

Figure 4.7(a) illustrates the basic arrangement of a coherent measurement. A source is emitting a signal with a controllable frequency ($\omega_{c\ell}$) and an amplitude s_ℓ, where $1 \leq \ell \leq L$. A comb frequency synthesizer enables the emission of narrow band signals from the source in distinct frequencies. The emitted signal is coupled to the modulating medium and detected by an integrated coherent sensor (ICS). Usually, N ICSs will be sensing the phenomenon, coherently related to the source $\omega_{c\ell}$. The sensed phenomenon on ICS_j ($1 \leq j \leq N$) is detected utilizing a coherent detector locked on $\omega_{c\ell}$ and the output is fed (if needed) to a V→f converter to generate a pulse stream $f_{\ell j}$. No

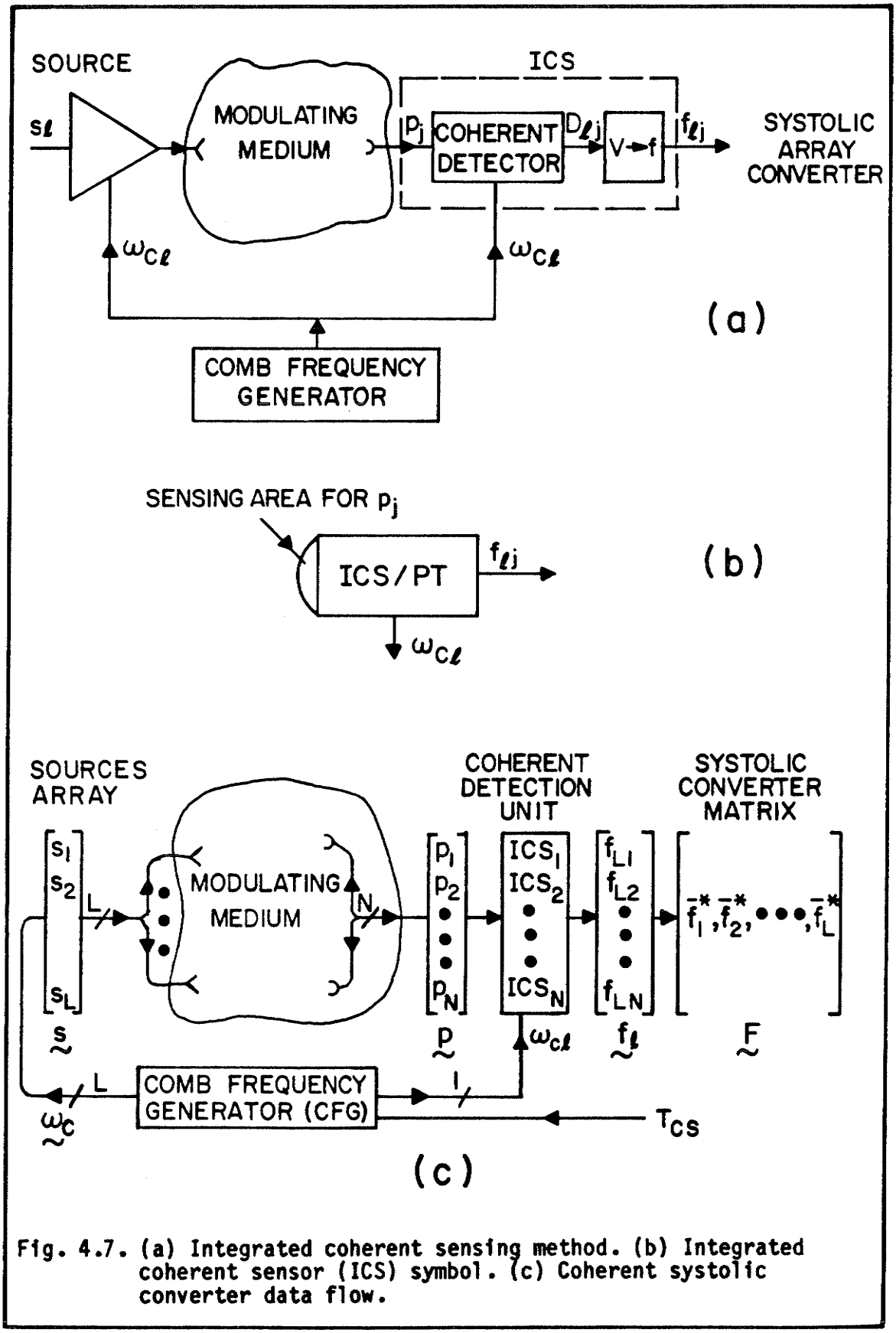


Fig. 4.7. (a) Integrated coherent sensing method. (b) Integrated coherent sensor (ICS) symbol. (c) Coherent systolic converter data flow.

limitations are posed on either the wavelength (light, sound, X-ray, γ radiation, etc.), or the coupling method with the modulating medium (either contact or noncontact). There is also no requirement that the modulating medium will be stationary.

Figure 4.7(b) provides the symbol of an ICS on channel j locked on source ℓ with lock frequency $\omega_{c\ell}$ and an output $f_{\ell j}$. All sources can also be commonly strobed to generate a frequency burst of a known period, as described in Sec. 9.5.

4.6.2 Coherent Systolic Converter (Vector Mode)

Let us consider an arrangement in which an array of L sources emits simultaneously signals with different frequencies into a modulating medium, as shown in Fig. 4.7(c). The simultaneous modulation of all sources in the array is emphasized by using a vector notation $\underset{\sim}{\omega_c}$ and $\underset{\sim}{s}$ for the modulated variables. A single array of N ICSs is utilized to sense the phenomenon resulting from the modulating medium. Let us define a sensed phenomenon $p_{j\ell}$ as a result of emitting only a single signal at a frequency $\omega_{c\ell}$. The sensed phenomenon p_j will be given by

$$p_j = \sum_{\ell=1}^{L} p_{j\ell} \tag{4.12}$$

if a properly designed linear sensor is used.

Let us now return to the signal and data flow of the coherent

systolic converter, as shown in Fig. 4.7(c). An array of L sources emits signals with corresponding discrete spectra $\underset{\sim}{\omega}_c$ and amplitudes $\underset{\sim}{s}$. For a given conversion time (in synchronization with T_{cs}) a single coherent frequency $\omega_{c\ell}$ is applied to all N ICSs in the coherent detection units, resulting in a vector $\underset{\sim}{f}_\ell$. The vectors, each associated with a different frequency, are converted and organized in a matrix $\underset{\sim}{F}$ for further processing. (The matrix $\underset{\sim}{F}$ should not be confused with a time-tagged state vector queue.) Although the data elements of each vector $\bar{f}^*_\ell$ are non-skewed, the vectors $\bar{f}^*_\ell$ of the matrix $\underset{\sim}{F}$ are skewed.

The method described above is specially suitable for a stationary modulating medium and a discrete spectra source array that must have common activation.

The method could be utilized for the construction of a computer aided tomography (CAT) machine that eliminates the traditional mechanical gantry. A set of radiation sources emitting different energies is spatially distributed above the modulating medium, while a coherent detector array is placed below the medium. The detector array can be tuned to respond to a specific energy by varying the discriminators threshold of the hybrid radiation sensor, as shown in Sec. 2.3. The matrix $\underset{\sim}{F}$ is obtained by L successive coherent conversions that correspond to <u>different angles</u> with <u>no moving</u> mechanical gantry.

Another application of the method, further described in Chapter IX, refers to a non-stationary medium and utilizes laser light to obtain the velocity profile of a moving medium by coherent Doppler shift detection. Such a method could also be viewed as a composite strobed and coherent systolic conversion, capable of eliminating the mechanical noise of the modulating medium.

To allow for real-time measurement of a nonstationary process occurring in the modulating medium, a strobed coherent systolic array conversion method has been developed and is presented in the following section.

4.6.3 Coherent Systolic Array Converter

Figure 4.8(a) illustrates the signal and data flow in a coherent systolic array converter. An array of L sources is emitting signals with corresponding amplitudes and frequencies denoted by $\underset{\sim}{s}$ and $\underset{\sim}{\omega_c}$ into the modulating medium. An array of NxL ICSs is organized as linear arrays, each locked on a corresponding source with $\omega_{c\ell}$. The coherent detection array, locked on $\underset{\sim}{\omega_c}$ and triggered by T_{cs}, results in a time-tagged state array represented by $\underset{\sim}{F}^{(i+k)}$. The state array is queued for further processing.

The architecture described above could be used for real-time CAT machines. For example, a machine may consist of L radiation sources and L arrays of N detectors in different angles, locked on $\underset{\sim}{\omega_c}$. At the

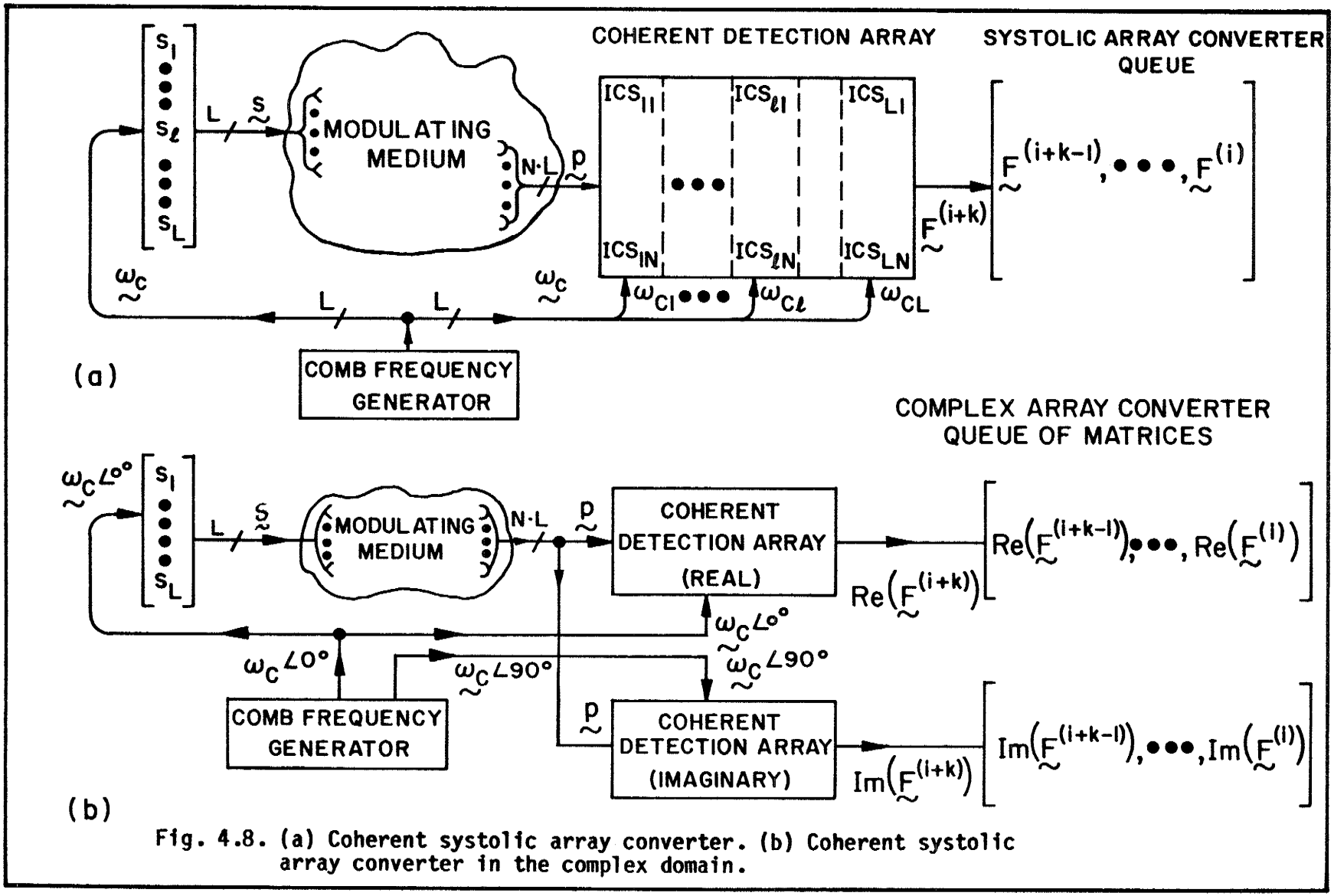


Fig. 4.8. (a) Coherent systolic array converter. (b) Coherent systolic array converter in the complex domain.

end of the conversion systole, a matrix $\underset{\sim}{F}^{(i)}$ is generated, representing L different viewing angles captured simultaneously. This method enables the construction of a CAT scanner for real time two dimensional imaging with no moving gantry.

4.6.4 Coherent Systolic Array Converter in the Complex Domain

In the above discussion, we assumed that the modulating medium can interact with the emitted signals causing a change in amplitude, frequency, or phase at the sensors. Again, no restrictions are posed on the signal wavelength (absolute or relative to the modulating medium), and its distribution and coupling method. In the case where the phase relation between the source array and sensor array is also of importance (e.g., direct observation of moving magnetic bubbles where the medium not only attenuates laser beams but also changes their polarization angle), a new method called a coherent systolic array conversion in the complex domain is needed.

Such an architecture is presented in Fig. 4.8(b). Two coherent detection arrays are utilized. One array is locked on the frequency set $\underset{\sim}{\omega}_c \angle 0^0$, while the other array is locked on the same set of frequencies but with a 90^0 phase shift, as denoted by $\underset{\sim}{\omega}_c \angle 90^0$. This coherent detection scheme will result in a complex state array represented in the real (in phase) and imaginary (90^0 out of phase) domain by $\mathrm{Re}(\underset{\sim}{F}^{(i+k)})$ and $\mathrm{Im}(\underset{\sim}{F}^{(i+k)})$, respectively. Both real and imaginary state arrays are placed on a queue for further systolic processing.

The coherent systolic array conversion method could be utilized in a variety of application areas such as computed tomography, field mapping and laser radar, each in a special purpose parallel algorithm. Most of the algorithms will reside in the {systolic}x{two dimensional square array}, or {systolic}x{two dimensional hexagonal array} subspace. The coherent systolic array converter must be viewed as a composite architecture, unifying the acquisition and processing problems for a given application area. The computational requirements of some future real-time biomedical imaging algorithms, also applicable to coherent systolic array conversion architectures, are partially identified in [42], [43], [44].

In this chapter, we have presented the systolic conversion method and the space of related systolic converter architectures. Of special interest are systolic converter architectures that are coherent with the vector oriented process in time (strobed SC), frequency (coherent array converter), and phase (complex coherent array converter). The description of systolic converter types is not complete and further research in this area is required. Furthermore, the network could also be designed to be coherent with processes in various MAs in the entire GMA, by extensions to the network systole. This concept requires again further research. The next chapter will describe a systolic network architecture for distributed systolic converters over a GMA.

CHAPTER V

SYSTOLIC NETWORK

The systolic converter architecture described in the previous chapter enables composite acquisition and processing of state vectors from a vector oriented process occuring in a single MA. In many cases, a vector oriented process occurs in an area that is too large to be monitored by a single systolic converter. The term "large" may refer not only to a physically large area but also to a physically small area with a process requiring a large number of sensors. Such a large area is referred to as a globally monitored area (GMA) and is partitioned primarily into independent MAs, each connected to a systolic converter. The global state vector from the GMA has, as its elements, state vectors from the distributed MAs. The necessary condition to obtain the global state vector is no data skew between its vector elements and scalar elements. Since each state vector has no internal data skew, it is sufficient to require no skew between state vectors. To satisfy the above requirements, all distributed systolic converters must be activated simultaneously.

The simultaneous activation of a systolic converter, each processing locally the resulting state vectors, has no meaning if the data cannot be processed globally to monitor the vector oriented process over the entire GMA. Let us now draw the analogy between the composite acquisition and processing problems regarding vector oriented process in a MA, and the GMA. The systolic conversion/correction cell in the MA is analogous to a systolic converter in the GMA. The simultaneously

activated array of conversion/correction cells in the MA is equivalent to a synchronized array of systolic converters in the GMA. If we further assume that the processing algorithms in the MA belong to the subspace {systolic}x{linear array}, then what are the equivalent synchronization method, module granularity and communication structures, in their broader sense, for a systolic network over the GMA? The answer will be provided in this chapter.

5.1 Communication Geometry and Access Methods for Systolic Networks.

The systolic converter architecture, suitable for a single vector oriented process has to be expanded to facilitate a vector oriented process in GMA. One of the required extensions is the synchronization of all distributed systolic converters. However, the communication geometry is not a simple scaled-up version of a VLSI linear systolic array geometry for the following reasons: (i) Parallel data exchange with lock-step mechanisms [35], commonly used in synchronous architectures such as systolic or SIMD (Single Instruction Multiple Data) [45], [46], is not practical due to the spatial distribution of MA; (ii) The influence of noise on the network (macroarchitecture) cannot be neglected, as in microarchitectures; and (iii) The global synchronization scheme for both systolic converter synchronization and distributed computing must be independent of the spatial distribution of the MAs.

Recent evolution in the area of local network architectures (LNA) and the experience from the multiloop network [5] served as a basis for a new systolic network architecture. The proposed architecture is

derived from a well known LNA with the proper extensions that are required for global vector oriented processes. Based on the work of the IEEE Local Network Standards Committee [47-49], two basic communication medium access methods have been identified: controlled and random. The random access method has three major limitations that eliminate its use for a systolic network and could be summarized as follows: (i) Data transfer time is non deterministic; (ii) The medium access control is distributed and the network synchronization is a function of its topology; and (iii) No provision exists for simultaneous data transfers between adjacent stations due to the common bus topology.

The controlled access method is based on a token-passing access scheme. The method is completely general and is used for mutual exclusion of processes requiring shared resources [50]. It relies on a specific data structure that is a key (or token) to the shared resource and is modified to reflect its usage while preventing further collisions. Once reconstructed, it is passed to another user. Token-passing could be utilized in two network topologies: (i) A loop where the token can be passed sequentially; (ii) A logical loop, or token-bus, where the token is passed to various stations sharing a common bus in an order which forms a logical loop [48]. Token passing loop architectures (both physical and logical) have the following properties: (i) The token, controlling the medium access, is just a special data entity within the regular data stream; (ii) Medium access and data delays are fully deterministic; and (iii) As a result of (i), there is no need for a collision detection mechanism which would depend on network topology

and placement as for the non deterministic method [51].

Since data can also be encoded to be self clocked (e.g. Non-Return to Zero Inverse-NRZI, Manchester), a token passing loop architecture can utilize a unidirectional transmission medium for clock extraction, medium access, and global network synchronization irrespective of its physical layout. This property is extremely useful if single-mode fibre-optics is used as the communication medium. The advanced VLSI architectures that partially implement token passing network protocols [51], made the required extensions for a systolic network technically plausible with current technology. The following section will describe network extension for a systolic converter utilizing an extended token passing network.

5.2 The Unified Network Element (UNE)

The architectural evolution of the systolic converter was governed by the following design principles: (i) At every systolic processor, one uses very simple control for communication and processing and makes local memory storage relatively small, constant and independent of the size of the network; (ii) One uses as many processors as can be kept simultaneously productive and locates them as close as possible to the data they require; and (iii) One utilizes pipelining and multiprocessing for parallel algorithms that have simple and regular data flow thus resulting in simple and regular geometries for the data path. The same design principles were utilized to derive the Unified Network Element (UNE) serving as the basic element in the construction of a systolic

network.

The UNE architecture is shown schematically in Fig. 5.1 It consists of a multiple processor structure with four basic building blocks:

a) A systolic converter with a conversion systole synchronized on a network systole;
b) A loop slave procesor for communication with higher levels of the sytolic network;
c) A loop master processor for communication with lower levels of the systolic network; and
d) A graphics processor for operator interface and network diagnostics.

Careful analysis of the communication topology of the UNE reveals its symmetry with the linearly connected systolic array processor. Data can be transferred to and from adjacent UNE in various configurations.

One of the most useful topologies in process control application is the hierarchical multilevel loop architecture, as described in [52], with its reliability aspects analyzed in [53], [54]. Such a network topology was first implemented using the USCM and different loop controllers, as described in Sec. 2.5.4. The multiple processor architecture of the UNE is dynamically reconfigurable to form six systolic network entities. All the configurations will be required to construct a systolic network without the need for different network entities.

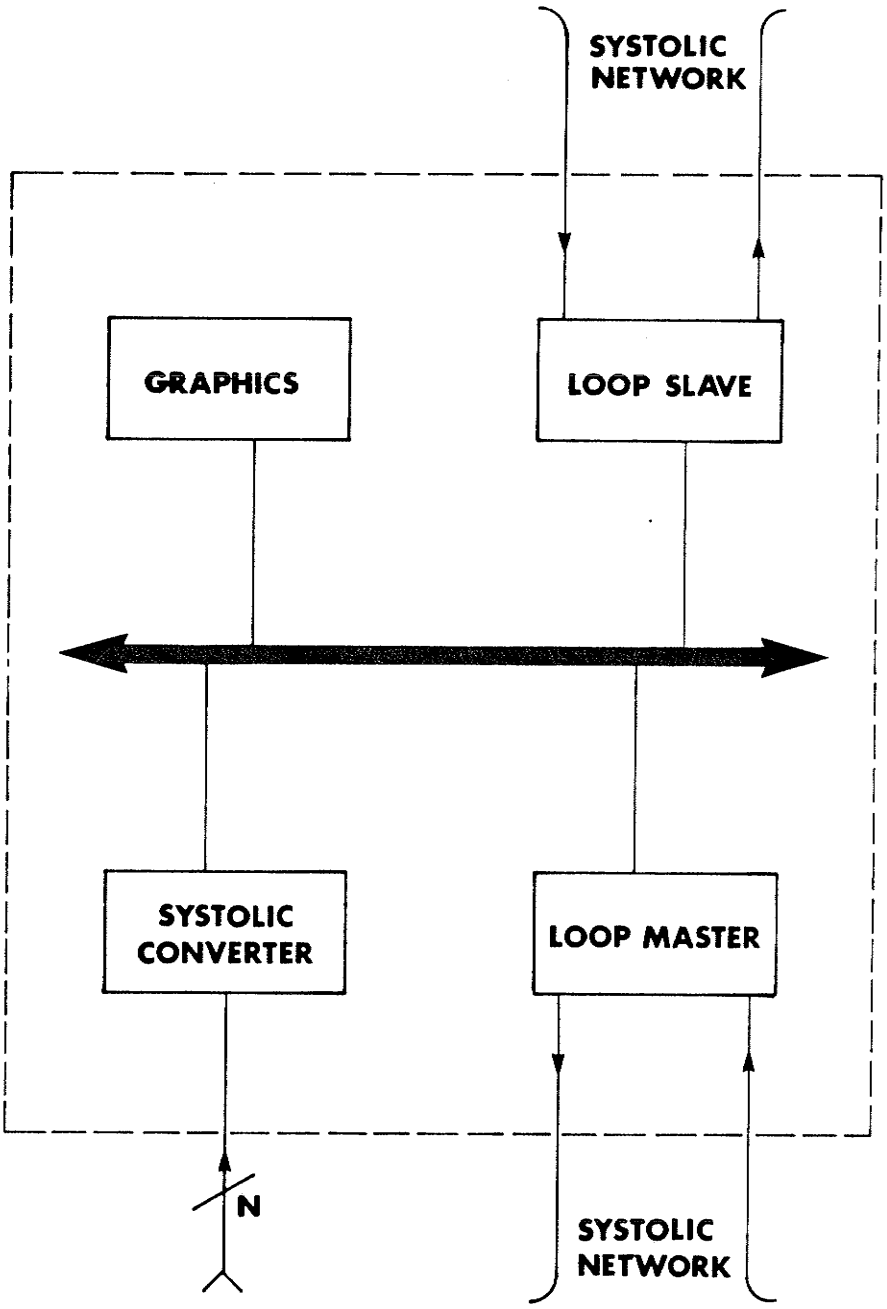


Fig. 5.1. Unified Network Element (UNE) reconfigurable architecture.

Let us describe the six network entities and their placement within the systolic network topology, as shown in Fig. 5.2 and Table 2:

a) A STATION consists of a synchronized systolic converter and a loop slave communication processor. Its primary use is for monitoring a vector oriented process in a single MA as part of the GMA;
b) A LOOP CONTROLLER consists of a loop master executing the following functions: (i) Generation of tokens for global network synchronization; (ii) Collection of the global state vector; and (iii) Distribution of parameter sets to various network entities;
c) A GATEWAY consists of loop-slave and loop-master communication processors and performs the following functions: (i) Synchronization of loops in adjacent levels; (ii) Concentration of state vectors at various levels; and (iii) Distribution of parameter sets to various network entities;
d) A NETWORK SIGNATURE ANALYZER (NSA) that serves as a reconfigurable analysis tool as well as a network optimization tool. Due to the systolic nature of the network, a network signature can be captured at various nodes, as described in Chapter VI. The NSA could be configured as: (i) A Station with simulated SC state vectors; (ii) A Loop Controller; and (iii) A Gateway, and placed at various network nodes, as denoted by an asterisk in Fig. 5.2.

The three basic network entities enable the construction of a hierarchical multiloop architecture based on a single reconfigurable UNE. The next section will describe the synchronization principle of a

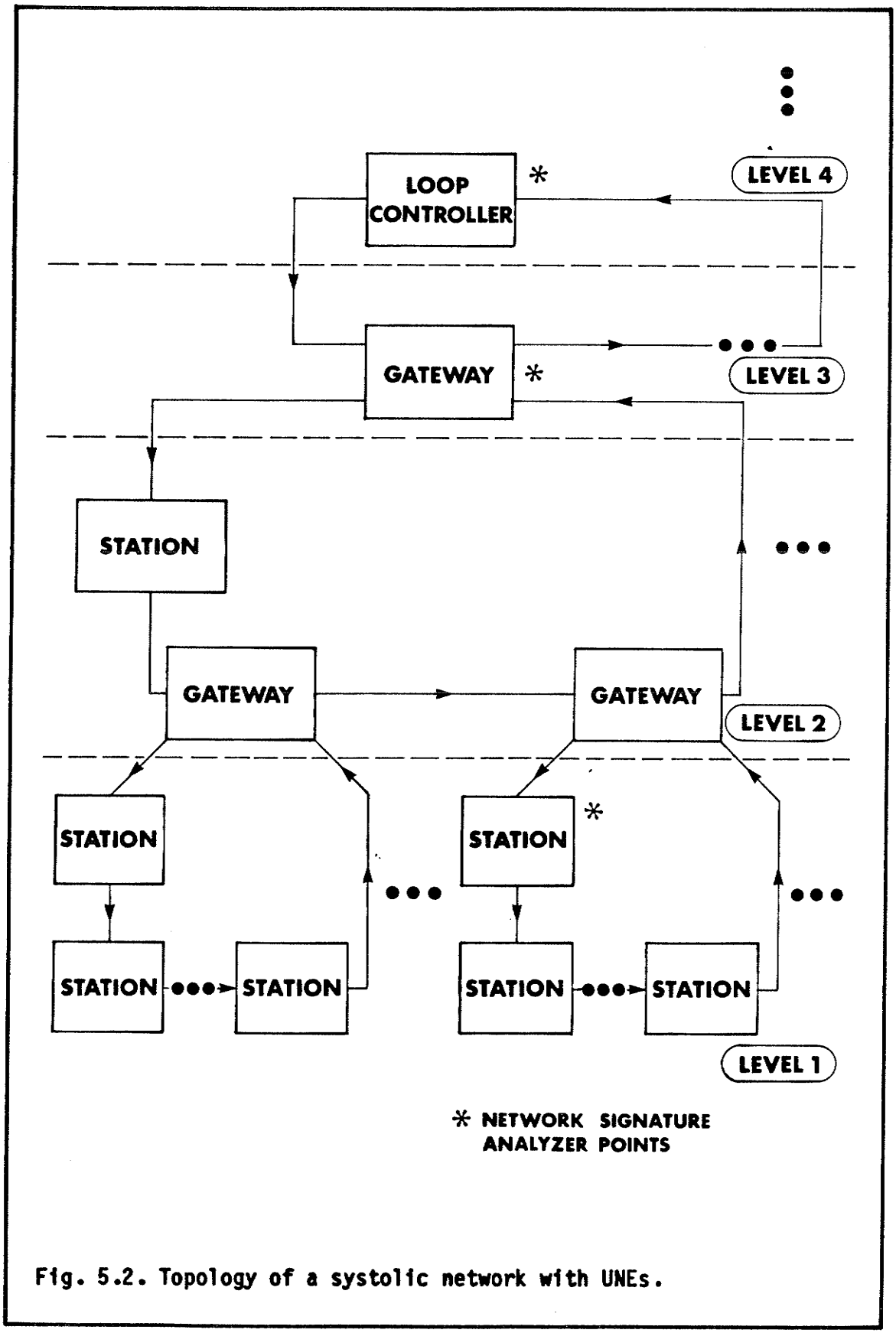


Fig. 5.2. Topology of a systolic network with UNEs.

Table 2. Possible configurations of UNEs in a systolic network (1-active and 0-nonactive).

	LOOP SLAVE	LOOP MASTER	GRAPHICS	SYSTOLIC CONVERTER
STATION	1	0	0	1
LOOP CONTROLLER	0	1	0	0
GATEWAY	1	1	0	0
NSA (STATION)	1	0	1	SIMULATED DATA
NSA (LOOP CONTROLLER)	0	1	1	0
NSA (GATEWAY)	1	1	1	0

systolic network.

5.3 Network Synchronization

The required global network synchronization can be obtained via the data stream of an extended token passing protocol in two steps: (i) Clock extraction and bit level synchronization; and (ii) Global network synchronization.

5.3.1 Clock Extraction and Bit Level Synchronization

Various self clocking encoding schemes such as NRZI and Manchester [55], enable clock extraction from the data stream utilizing the digital phase-lock technique. Two basic techniques exist for digital phase locked loop (PLL) clock extraction: (i) Charge-pump or sequential logic phase/frequency detector, as described in [56]; and (ii) Quadrant all digital PLL [57]. The second method relies on digital phase adjustments and a flywheel local oscillator (usually 32 times the data rate) to extract the data sampling clock.

Reconfiguration of the UNE into one of the three basic network entities requires rearrangement of the digital phased locked loop (DPLL) clock extraction scheme, as shown in Fig. 5.3. A loop slave in a station has its transmitter (T_x) locked on the DPLL clock obtained from the receiver (R_x) data stream, as shown in Fig. 5.3(c) and indicated by a closed S_1 switch. A loop master in a Loop Controller, shown in Fig. 5.3(b), maintains all loop slaves in a lock condition including R_x. Hence, its transmitter should not be phase locked on its

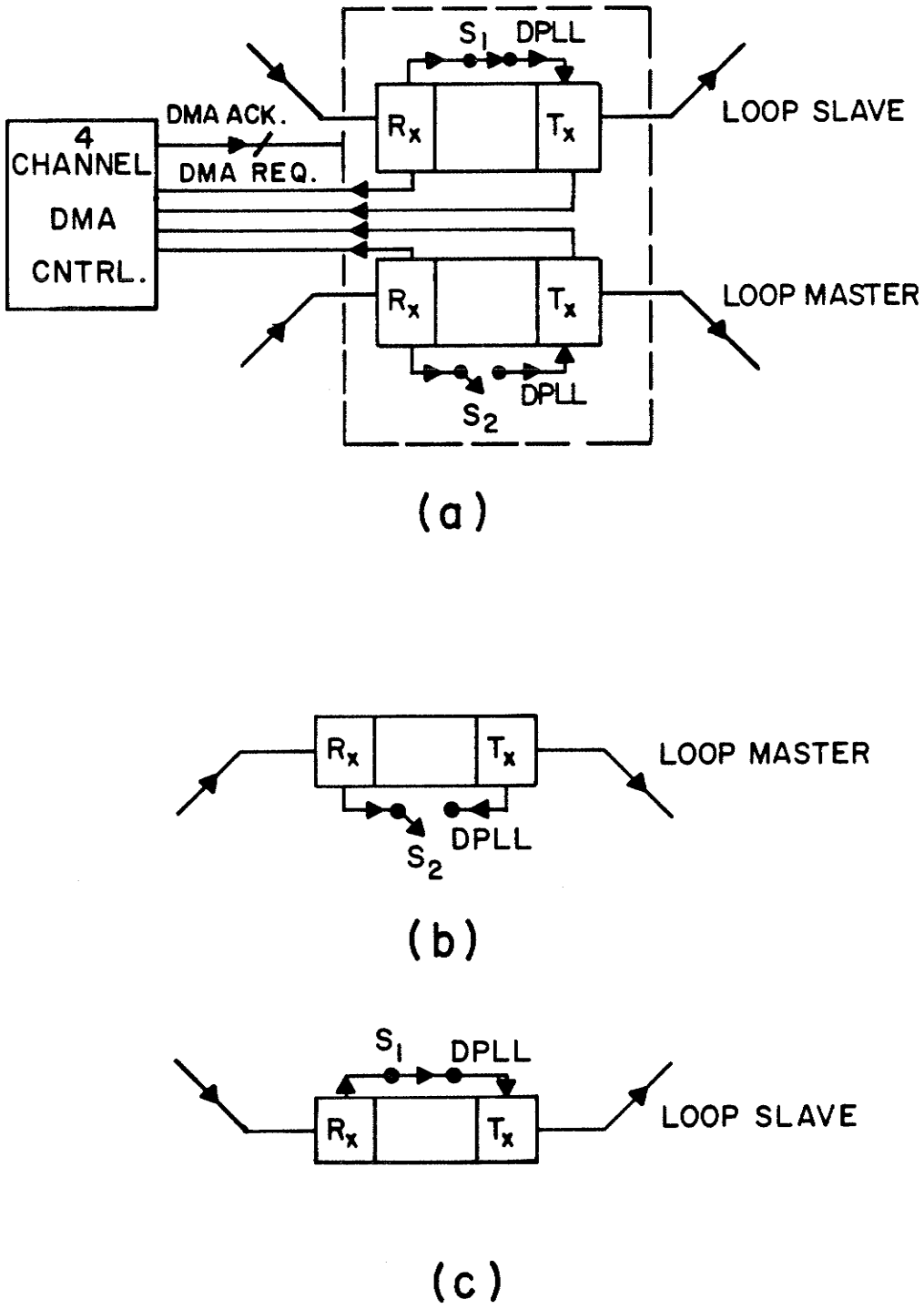


Fig. 5.3. Digital phase locked loop synchronization. (a) For gateway. (b) For loop controller. (c) For station.

receiver as indicated by an open S_2 switch. The gateway shown in Fig. 5.3(a) could be simply viewed as a superposition of both structures. A four channel DMA controller is then sufficient for data transfer in the three configurations.

5.3.2 Systolic Network Synchronization

The method for systolic network synchronization relies on the following features of token passing in a physical or logical loop topology: (i) Token extraction is fully deterministic; (ii) Given a stationary network, all token delays are predefined as a function of the current network topology; (iii) A token can also carry an associated absolute time tag, thus enabling digital phase lock [58]. Figure 5.4 depicts the network synchronization process.

A UNE configured as a Station extracts a token and its related time tag via its loop slave. The synchronization error unit receives the following four inputs: (i) R_x data clock; (ii) Token received flag followed by absolute time tag; (iii) Conversion systole T_{cs}; and (iv) The path delay of the token from its origin (loop controller or gateway) to the R_x. The calculation is based on the network topology status which is commonly distributed within the loop and contains information on network entities that are on or off the loop as well as their interconnecting medium length.

The phase difference result at the output of the synchronization unit, is fed to a digitally programmable delay unit that triggers the

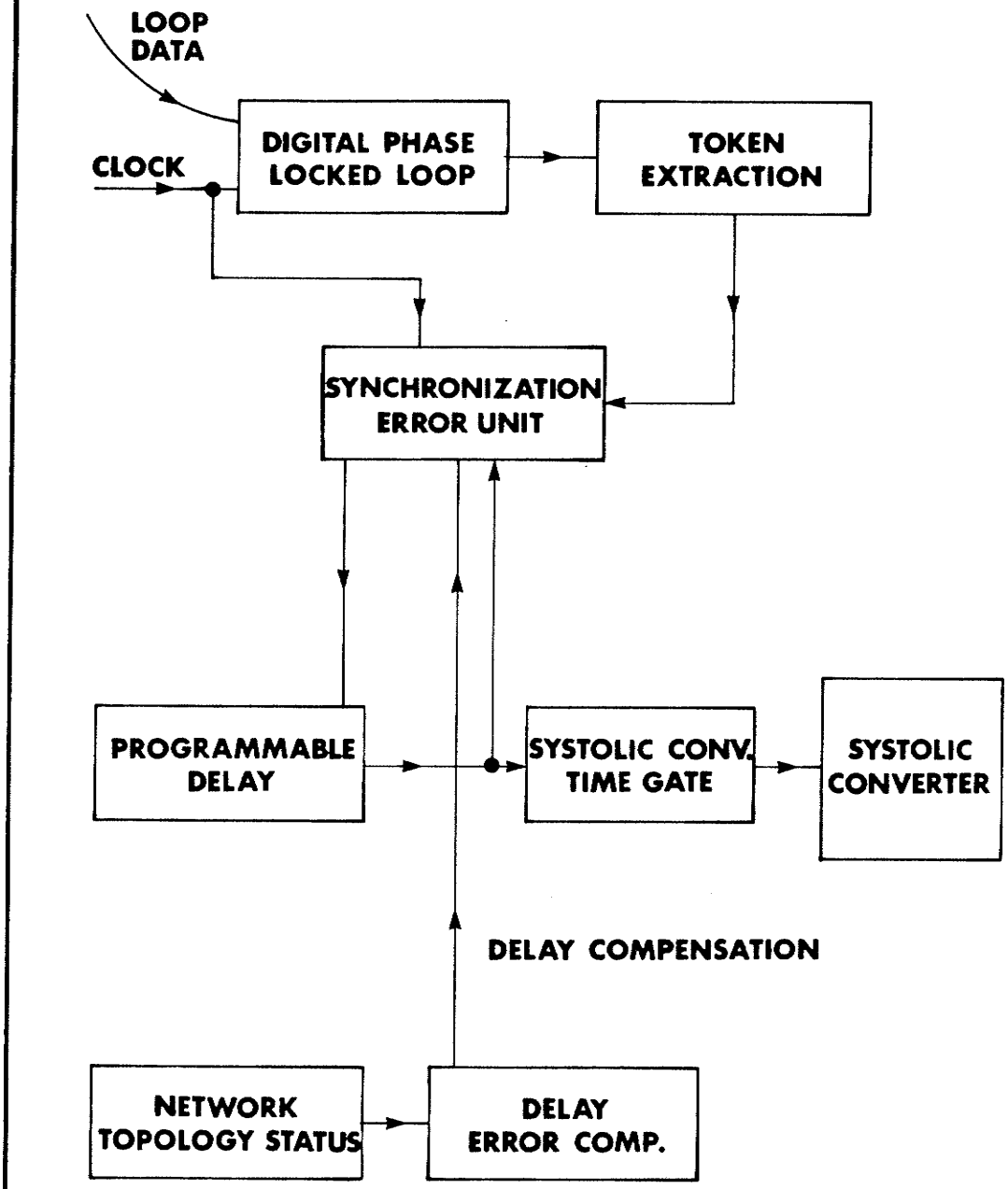


Fig. 5.4. A scheme for systolic network synchronization.

systolic conversion time gate. The closed-loop synchronization scheme resembles the DPLL operation at the bit level. The net result of the synchronization scheme is to lock all conversion systoles on a common time-tagged token, thus facilitating global state vector conversion and processing.

Let us now define the period between network systoles (T_{ns}), each embodied as a time-tagged token, as the inverse of the rate of the global state vectors obtained from the systolic conversion, processing the data reduction through the systolic network. In a hierarchical systolic network based on UNEs, (see Fig. 5.2), the network is synchronized from top to bottom. A network systole is sent from a loop controller to all gateways which propagate the systole to the lower level using the mechanism described above. The network topology and its status are commonly distributed and updated at the proper levels in order to maintain global synchronization.

We can observe that the systolic network is governed by the following three systoles: (i) network (T_{ns}), (ii) conversion (T_{cs}), and (iii) processing (T_{ps}). The same rules, that call for global synchronization while allowing same local desynchronization for a VLSI systolic array, are now expanded for the systolic network. The following section will describe a unified network protocol (UNP) for the systolic network to facilitate the hardware oriented description of a UNE.

5.4 Layered Systolic Network

Token passing protocols for loop architectures with centralized control are referred to as Newhall loop protocols [59], [60]. The reasons for using the Newhall loop protocol as a base for the systolic network were mostly given in Sec. 5.1. The utilization of the token for synchronization, as described in Sec. 5.2, implies centralized control resulting in a Newhall loop. The Newhall loop protocol is further modified for systolic network applications and will be described in more detail in Sec. 5.5.

The multiprocessor architecture of the UNE allows for a unified reconfigurable hardware structure to serve the three basic network entities: Station, Gateway and Loop Controller for the network construction. The same approach can be used in the design of the unified network protocol (UNP) of the systolic network. A reconfigurable firmware structure, sufficient to support a systolic network, has been designed based on four UNP entities (see [71]).

The four UNP entities and their placement within the three network entities are shown in Fig. 5.5. Each of the UNP entities constists of a pertinent communication processor and a matched set of software tasks. The hierarchical nature of the systolic network is shown, as well as the command (including tokens for T_{ns}) and data flow. The four UNP entities are:

a) An EMITTER to send commands and data to lower levels;

b) A COLLECTOR to receive commands and data from a higher level;

c) A SERVER to respond with data to a higher level; and

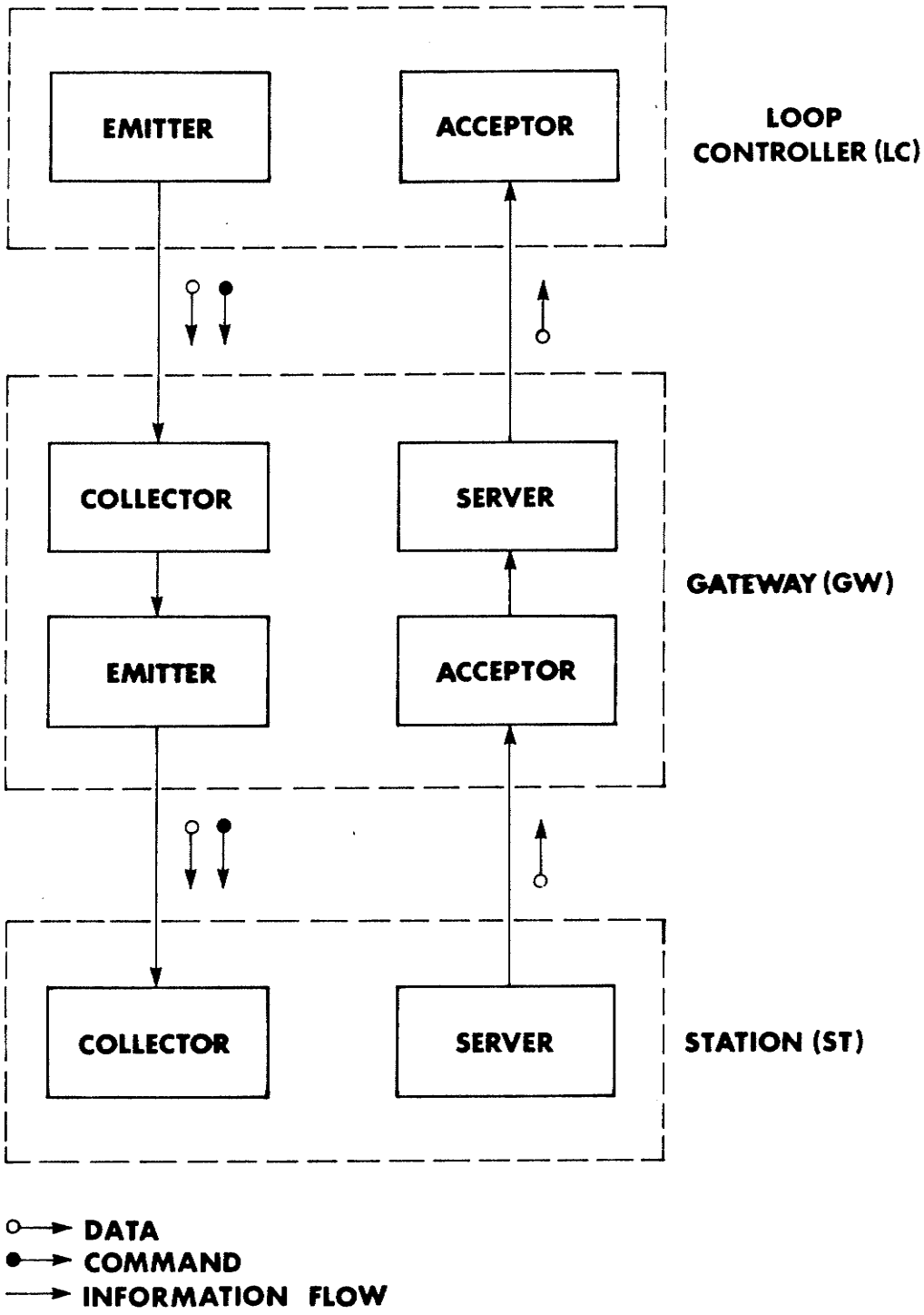


Fig. 5.5. Command and data flow in a Unified Network Protocol (UNP).

d) An ACCEPTOR to accept the data from a lower level.

The UNP architecture exhibits the following properties: (i) Only four entities are needed for the implementation of a systolic network with an arbitrary number of loops and levels; (ii) Commands (and T_{ns} tokens) are flowing only from upper levels to lower levels; (iii) Data flowing from bottom layers to top layers are always a response to a command (with the exception for network failures). The UNP architecture for a systolic network implies reduction in the data flowing upwards and expansion in commands and data flowing downwards.

The layered network architecture, known as the International Standards Organization Open System Interconnect (ISO OSI) seven layer architecture [61], [62] and also adopted and modified by the IEEE 802 local network standards committee [47], has provided the basis for coping with the design complexity of a systolic network. The two lower layers of the ISO OSI model (physical and data link layers) are mapped into three corresponding layers of the local area network model (physical signaling, media access control, and logical link control), [48], [49]. In our discussion, we shall follow the IEEE 802 model for token passing layered loop architecture.

Figure 5.6 provides the layered network architecture for the four basic UNP entities. The physical signalling and medium access control (centralized token passing) were omitted for clarity, while the link and network layers are shown at various UNP entities. It should be pointed out that only one software package for the link layer and net-

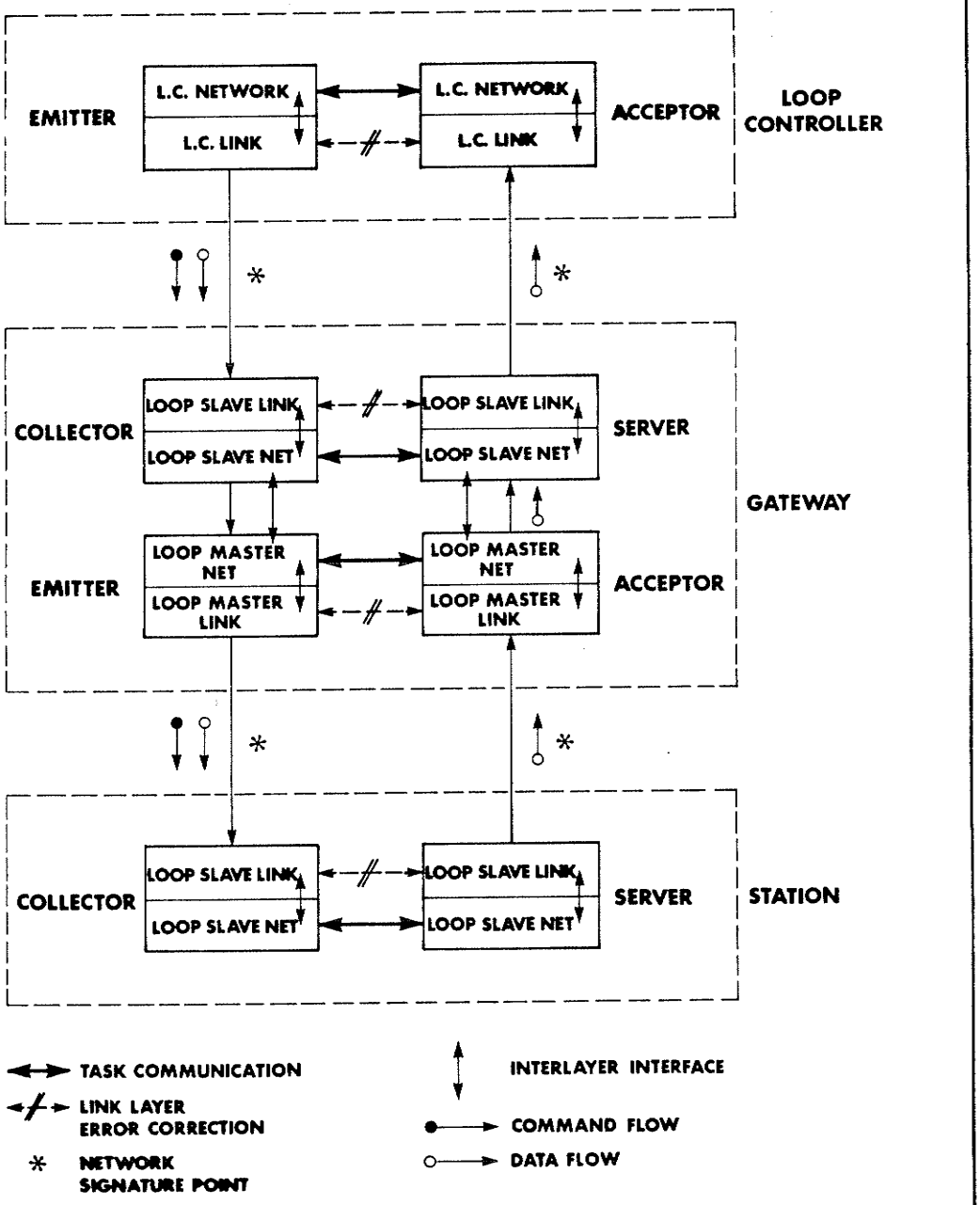


Fig. 5.6. Layered systolic network architecture.

work layer was designed to be reconfigurable at various network entities (see [71]).

The UNE concept supported by the UNP architecture form a unified systolic network that possesses the following features: (i) One UNE forms all three basic network entities; (ii) The UNP forms four basic communication entities; (iii) Two layers are used at the various network levels; (iv) Command and data flow are hierarchical and step locked locally and globally; (v) No queue exists for command extensions; (vi) Global and local network synchronization for systolic converters within loop stations.

5.5 Newhall Loop Architecture Modifications

The Newhall loop architecture first described in [63] has the following key features: (i) Token passing medium access; (ii) Centralized control for token distribution, data transfer and token claim; and (iii) Data transfers are only between loop slaves to a loop master. Let us highlight the modification to the Newhall loop for a systolic network.

5.5.1 Systolic Converter Synchronization and Pipelining

The basic mechanism for generation of the network systole T_{ns} was described in Sec. 5.3.2. Processing and conversion were pipelined in the USCM and SC to enhance system throughput. Since distributed processing of state vectors implies their exchange between various net-

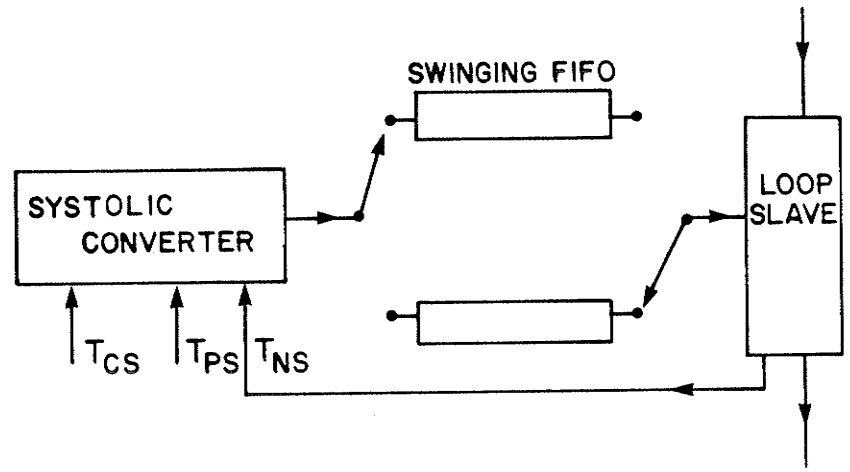
work entities a proper scheme must be provided.

The pipelining and synchronization scheme is shown in Fig. 5.7(a). The T_{ns} is generated to synchronize the SC. A swinging FIFO scheme is utlized to facilitate concurrent systolic conversion and state vector processing via the systolic network.

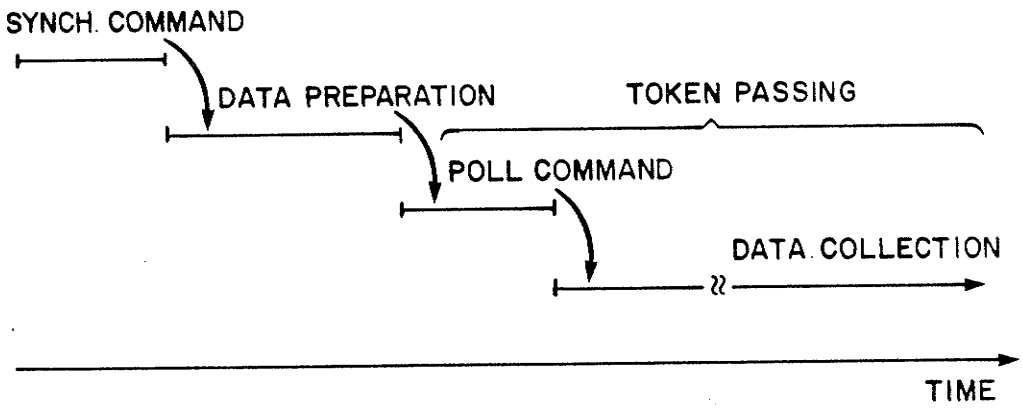
The conversion and processing of a modified Newhall loop is shown in Fig. 5.7(b). A synchronization command, implementing a time-tagged token, is generated to synchronize all SCs. State vectors are simultaneously acquired and concurrently processed at the SCs and the results placed with the swinging FIFO. A poll command, implemented by a token followed by a specific tag initiates consecutive transfers of locally processed state vectors from the distributed UNEs through their loop slave to the loop master. Although this scheme resembles an SC with one processor, it is useful in many applications where extremely high data reduction can be obtained at the MA level (e.g., in a global shut down system where each SC connected to an MA generates a single bit only). For such applications, the global state vector consists of bits obtained from the individual UNEs. Chapter VI will describe further modification to the Newhall loop to facilitate systolic processing throughout the network.

5.5.2 Loop Control Exchange and Loop Fragmentation

One of the key problems in the Newhall loop is the identification of a line cut in the loop and the proper recovery procedure from such a



(a)



(b)

Fig. 5.7. (a) Systolic converter synchronization and pipelining.
(b) Distributed systolic conversion and processing.

failure. Figure 5.8 describes the three steps of line cut (the term cut should be treated as generic to a complete loss of signal path between adjacent stations for any reason): (i) Identification; (ii) Control exchange; and (iii) Recovery. To illustrate the phase lock of a T_X and R_X in a loop master and a loop slave, we introduce the following symbology: (i) A loop master is indicated by two separated half circles as in Fig. 5.8(c), indicating unlocked R_X and T_X; and (ii) A loop slave is indicated by joined half circles as in Fig. 5.8(a), indicating a locked T_X and R_X. In both figures abbreviations are used for receiver data (R_D), transmitter data (T_D), and receiver and transmitter clocks (R_C, T_C). The digital phase-locked loop clock is a multiplicity of the baud rate and denoted by Z•clock, while the extracted data clock is denoted by DPLL and is intended to feed either the R_C or both the R_C and T_C.

The line cut is identified (Fig. 5.8(a)) as a result of signal loss at the R_X input followed by a flywheeling PLL. As a result, the loop can now be viewed as two separate linearly connected fragments of the loop, as shown in Fig. 5.8(b). The right fragment is locked on T_X of the station closest to the cut, while the left fragment is locked on the loop master T_X. A special token indicating the position of the cut is passed to the loop master R_X. The two loop fragments are now fully controlled by two masters. When the line cut is repaired at the cut point, indicated by the token passed to the loop master, network control is restored. It should be noted that this procedure could be utilized in high reliability loop architectures for implementation of self-healing features. The techniques for self-healing loop

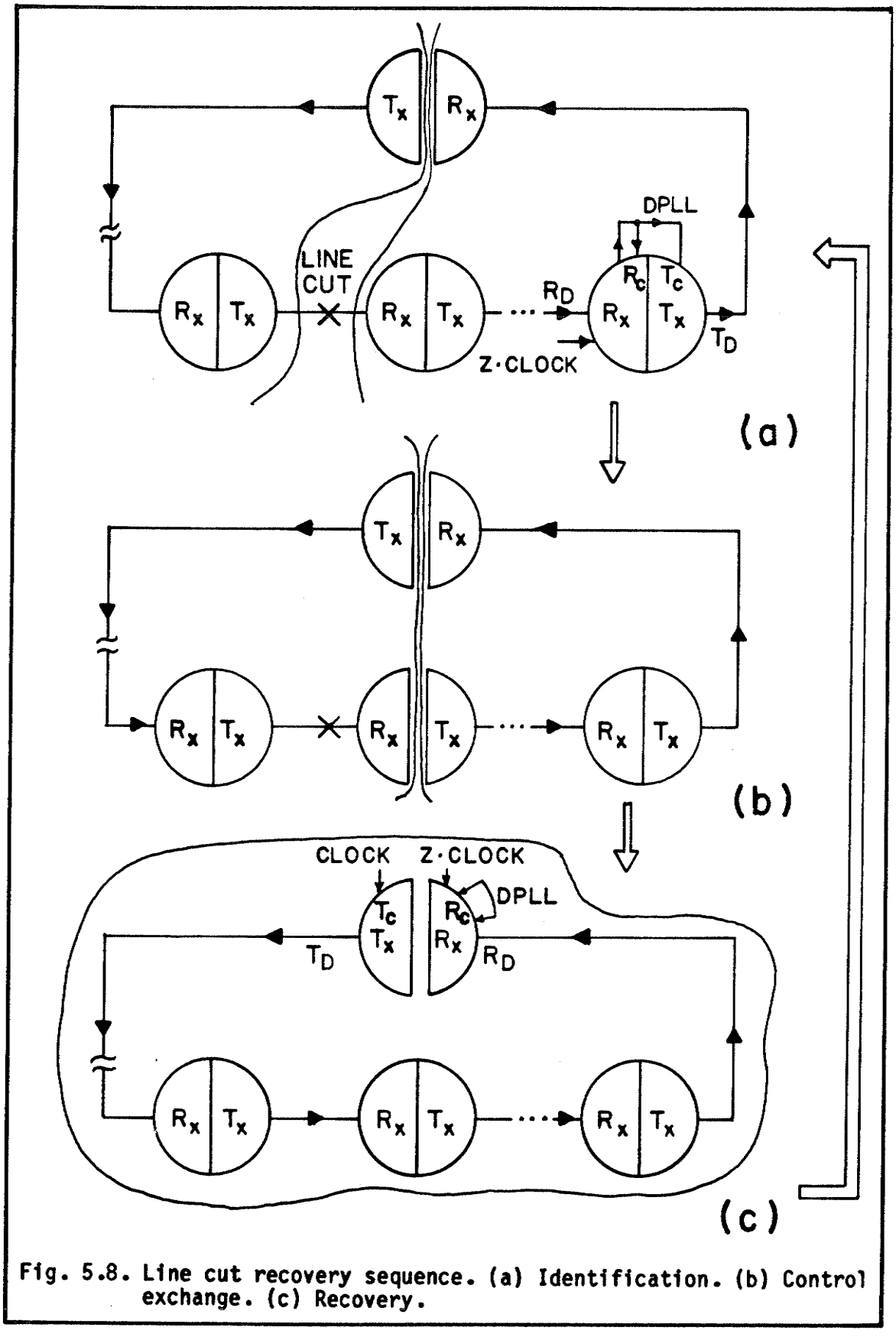


Fig. 5.8. Line cut recovery sequence. (a) Identification. (b) Control exchange. (c) Recovery.

architectures are described in [59], pp. 53-58.

The most important result of the above discussion is that a modified Newhall loop could be broken into fragments that can transmit data upstream. If two loops are used in opposite directions, data exchanges in both directions can occur concurrently between adjacent stations, thus forming a linearly connected systolic network, as described in Chapter VI.

CHAPTER VI

VECTOR ORIENTED PROCESSING ON A SYSTOLIC NETWORK

This chapter addresses one of the fundamental questions of how to acquire, transfer, and process individual state vectors from each MA in order to achieve the desired global results, corresponding to the entire global monitored area (GMA). The concepts presented in this chapter are essential to the understanding of the systolic network in that they provide a unified solution to the problem of the total network synchronization which includes not only data acquisition and transfer, but also data processing. The synchronization of data acquisition and transfer from a single MA or a cluster of MAs was discussed in the previous chapters. The inclusion of data processing poses new and difficult synchronization problems, the solution of which is presented in this chapter.

In the search for the solution, the following two objectives played an important role: (i) The network must be deterministic in order to assure no data skew between the state vectors from the separate MAs; and (ii) The synchronization must be achieved through constructs which are naturally related to those already introduced at a lower level in the previous chapters. The latter objective was necessary to achieve the unified design methodology, presented in the next chapter, and to relate the constructs to those found in the systolic arrays, as introduced by Kung [35].

The deterministic feature of the network is assured by an ordered sequence of events in the entire network and its entities (such as the

Station, Loop Controller, Gateway, Loop Analyzer, and the UNP). Furthermore, the entry of data into the network is not random, even though the measured quantities may occur randomly at the sensor level, as in radiation monitoring.

We shall first address the synchronization problem in systolic arrays and its relation to the systolic network.

6.1 System Timing in VLSI and Systolic Networks

In the design of synchronous systems, one must consider synchronization failure as a fundamental problem. The failure refers to a metastable state of a synchronous system when asynchronous data arrive at its inputs. The probability of a synchronization failure cannot be reduced to zero. It can, however, be minimized and calculated using the Mead criterion, as described in [37], pp. 236-242.

The problem also appears in the synchronization between distinct synchronous subsystems, each with its own clock. Again, the probability of a synchronization failure between the subsystems is non zero. The solution to the problem is self-timing. A self-timed element can be designed as a synchronous system with an internal clock that can be stopped synchronously and restarted asynchronously. Following Mead's definition, a self-timed system is either (i) A self-timed element, or (ii) A legal interconnection of self-timed elements.

Let us consider the local and global synchronization problems of the systolic network in order to demonstrate that the previous constructs fully satisfy the synchronization requirements. The systolic converter has been designed as a self-timed system with two basic elements that are internally synchronous. The first element is the systolic conversion/correction array. Although the conversion gate for each cell within the element is dependent on $\underset{\sim}{S}_j$ and $\underset{\sim}{G}_j$, the whole array can be synchronously stopped by a strobed conversion gate signal, then asynchronously restarted. The second element is the systolic array governed by the processing systole, T_{ps}.

In addition, the synchronization of the systolic converter, SC, located at different UNEs is also based on the self-timing principle because the DPLL is a synchronous finite state machine that can be viewed as an element in relation with the received data! The SC, governed by its conversion systole, is an element as viewed by the token driven synchronization unit. The name Unified Network Element has now a new meaning in the context of the systolic network! Let us clarify that in the following section.

6.2 Vector Oriented Processing on a Modified Newhall Loop

Let us analyze a simple case of a vector oriented process monitored by three UNEs, each related to an MA, in a single level systolic network. Let us further assume that the global unprocessed state vector $\underset{\sim}{f}^{*(i)}$ has to be multiplied by a global matrix $\underset{\sim}{A}$ (see Eq. 6.1).

$$\left[\begin{array}{c|c|c} A_{11} & A_{12} & A_{13} \\ \hline A_{21} & A_{22} & A_{23} \\ \hline A_{31} & A_{32} & A_{33} \end{array}\right] \left[\begin{array}{c} \underset{\sim}{f}_1^{*(i)} \\ \hline \underset{\sim}{f}_2^{*(i)} \\ \hline \underset{\sim}{f}_3^{*(i)} \end{array}\right] \qquad (6.1)$$

$$\underset{\sim}{f}_G^{*(i)}$$

If each state vector consists of N elements, then each of the submatrices in $\underset{\sim}{A}$ may be an NxN square matrix. For such a vector oriented process monitoring, the following two processing structures are plausible: (i) Centralized; (ii) Distributed. The structures are related to the Newhall loop.

The centralized structure relies on gathering the state vectors from the stations to the loop controller in order to perform the multiplication of $\underset{\sim}{A}$ by $\underset{\sim G}{f}^{*(i)}$ on a systolic array, as will be further analyzed in the following sections.

The distributed structure is based on the following four stages: (i) The global state vector $\underset{\sim}{f}_G^{*(i)}$ is collected by the loop controller; (ii) A copy of the global state vector is distributed to the UNEs; (iii) Each UNE performs a multiplication of a single row of the NxN submatrices by the global state vector; and (iv) The results are transferred to the loop controller. It is seen that since in both methods there are no direct data transfers between UNEs, the algorithm space is limited. In order to achieve concurrent data transfers between the UNEs, a new network topology must be introduced. The new structure is described in the following section.

6.3 Processing on a Systolic Network

Figure 6.1(a) depicts a linearly connected array of inner product step processors (IPSP) performing a systolic multiplication of a vector $\underset{\sim}{x}$ and a matrix $\underset{\sim}{A}$ which results in a vector $\underset{\sim}{y}$ [37]. The array has the following four features: (i) Data exchanges occur concurrently between the adjacent IPSPs in opposite directions over the array (step locked); (ii) Flow of input and output is in opposite directions; (iii) Loading of $\underset{\sim}{A}$ elements is pipelined with the computation flow; and (iv) The flow of the pertinent elements of $\underset{\sim}{A}$ into the array is predefined. The above properties could be used in the synthesis of the desired systolic network to operate on state vectors which can be treated as the basic

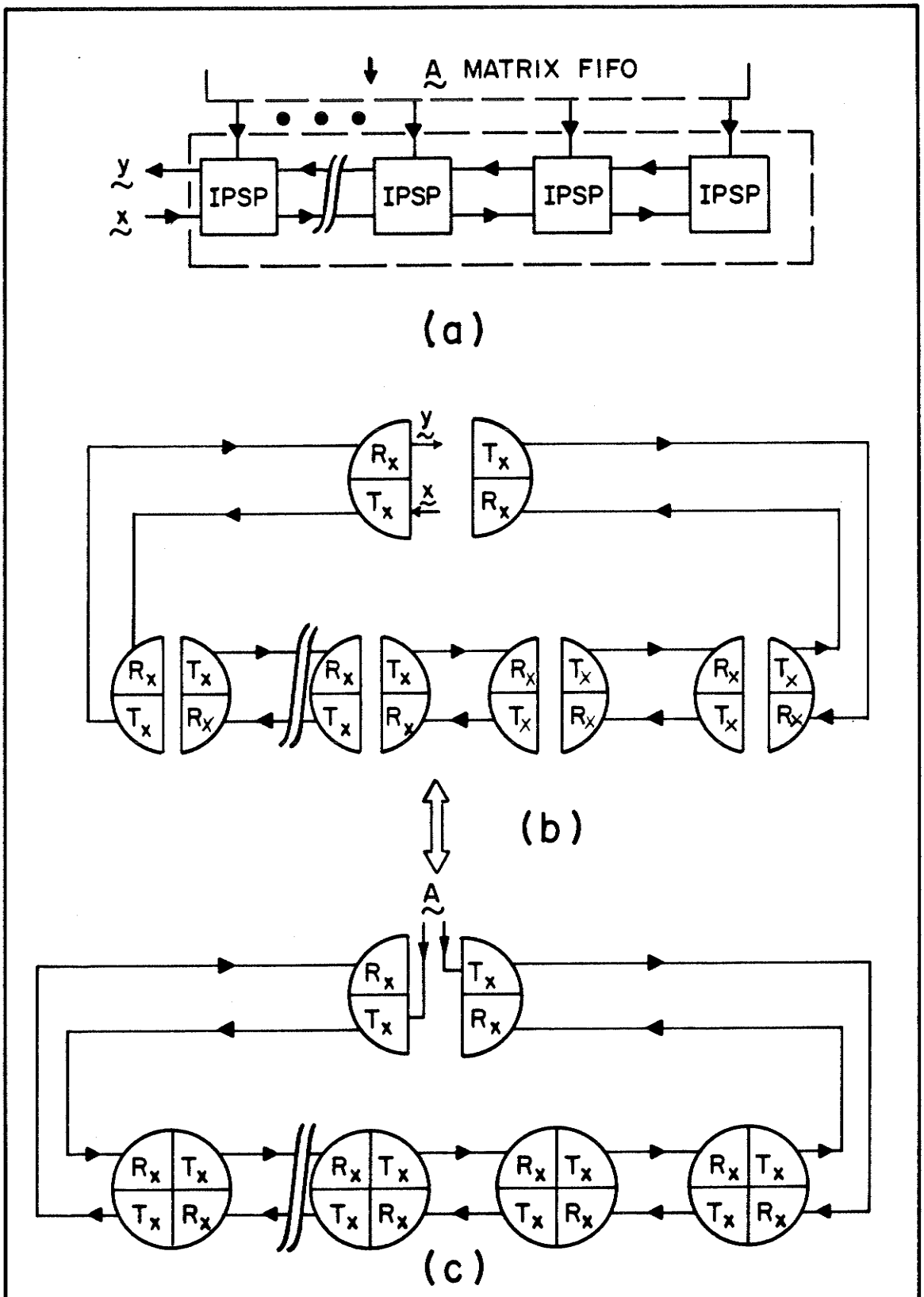


Fig. 6.1. (a) Linearly connected systolic array. (b) Systolic network in exchange mode. (c) Systolic network in broadcast mode.

data elements, and to perform the required matrix multiplication on the global state vector.

Figure 6.1(b) illustrates graphically the analogy between the systolic array and systolic network. Two loops are utilized with data flow in opposite directons. A special token is used to signal all stations to enter the exchange mode. In this mode, each UNE unlocks its transmitter from the receiver. Unlike in the unidirectional Newhall loop, this structure permits concurrent data exchanges in opposite directions. Following the data exchange and computation, the elements of matrix $\underset{\sim}{A}$ may be transferred to the UNEs. The transfer occurs concurrently to all network entities. Such a transfer shall be called the broadcast mode and is shown in Fig. 5(c). The distribution of $\underset{\sim}{A}$ elements to the UNEs could also be performed at the network initialization time. The basic data element that is utilized here is a state vector, while the basic element of $\underset{\sim}{A}$ is a submatrix.

The immediate advantage of the systolic network is the simplicity of applying known parallel algorithms, developed for VLSI implementations to vector oriented process monitoring. The self-timing property of the network via a token passing mechanism and the organization of the basic network entity as an element can now be seen in a new light.

The following two sections will present some first order measures to the systolic network performance in the centralized and systolic modes.

6.4 Systolic Network in Centralized Mode

This mode will be utilized primarily when the local processing of state vectors at the distributed UNEs results in a relatively high data reduction. The output vector of the systolic converter will be called a reduced state vector $\underset{\sim}{f}_R^{(i)}$. The total time, T_{GP}, to poll the reduced state vectors and to process them is given by

$$T_{GP} = N \cdot T_F + N \cdot T_{BIT} + Q \cdot T_{ps} \qquad (6.2)$$

where N is the number of UNEs (and also state vectors), T_F is the transmission time of a frame containing a reduced state vector from the UNE to the loop controller, T_{BIT} is the inserted one bit delay for the token, Q is an integer that represents the number of required computation steps for the matrix and vector multiplication at the loop controller, and T_{ps} is the processing systole. To pipeline the conversion, as well as the state vector processing and reduction at all UNEs, with centralized processing at the loop controller, the network systole T_{ns}, must satisfy the following condition

$$T_{ns} > T_{GP} \qquad (6.3)$$

Equation (6.3) is similar to Eq. (2.1) for a USCM except for the data elements which are now the reduced state vectors. The total time T_{GP} should be shorter than the network systole T_{ns} to allow a margin for self timing.

In applications where extreme data reduction can be achieved, such as in the global shut-down system, a microprocessor based UNE, configured as a loop controller, is adequate to process the data.

6.5 Systolic Network Throughput and Pipelining

For a systolic network, described in Sec. 6.3, the total time to process the reduced state vectors is given by

$$T_{GP} = (T_{EX} + Q \cdot T_{ps}) \cdot Q_G \qquad (6.4)$$

where T_{EX} is the time for vector data exchange between UNEs, Q is the required number of processing systoles within each UNE, and Q_G is the number of exchanges that are required for computational network systoles. The same condition given by Eq. (6.3) applies again to assure pipelining of conversion and processing. It is assumed that the loading of the proper submatrices to the UNEs is performed at initialization time.

As a result of the above discussion, we can clearly see that the logical ring, also known as the token bus, is a topology not adequate for systolic network implementations since it cannot provide for the necessary concurrent exchange of state vectors. Instead, the systolic network requires the proposed dual loop construct.

6.6 Network Signature

As indicated in the introduction to this chapter, one of the fundamental requirements imposed on a network that could be called systolic is the deterministic flow of events, governing the conversion, transfer and processing of all the state vectors in the network. With its global and local systoles, it is possible to conjecture that any sytolic network will exhibit characteristic state transition patterns at various network entities and that the patterns may be used to assess the performance of the network and the correctness of its operation. This method of verifying the operation of a systolic network is an extension of the Circuit Signature Analysis, as introduced by Hewlett-Packard [63a], and may provide a viable tool for network analysis (diagnostics), not suggested in the related work [47].

A formal representation of network protocols is usually done with the help of either of the following tools: (i) Finite state model [64]; (ii) Petri nets [65]; and (iii) Abstract process network [65]. The IEEE 802 task force [47] preferred the adoption of the finite state model representation for token-passing protocols due to the limited design and synthesis capabilities of Petri nets. Since a UNE in a systolic network can be represented internally as a synchronous finite state machine including the modified token-passing protocol, its state transitions could be monitored and time-tagged relative to the state vector processing. A Network Signature Analyzer could use the deterministic features of the systolic network as well as finite state presentation of the UNP (unified network protocol) to obtain the following: (i) Network performance in terms of commands, tokens,

conversion and processing of both the state vectors, and global state vector; (ii) Monitoring of the network topology and status (active level, loops, network entities); (iii) Presentation of a network signature at various network nodes for diagnostics and maintenance.

Since all the data and systoles flow through the loop topology, the systolic network can be analyzed (diagnosed) from any node. Other network topologies, such as the star configuration, will not allow such an analysis from its branches. Since the UNP is implemented with a layered architecture, we can include a diagnostic layer at each of the UNP entities. The network analyzer interfaces with the diagnostic layer to obtain pertinent data on the event flow (state transition, tokens, state vector transfers, line cuts) at various network entities, as well as to provide stimuli (tokens) to control the signature gathering process. In this capacity, the network analyzer serves as a remote diagnostics probe. The function of the remote diagnostic probe can also be found in the testing and verification of VLSI devices and systems if and only if a diagnostic layer is provided throughout the system elements. This further emphasizes the parallel between the macroarchitectures (systolic network) and microarchitectures (systolic converter).

In conclusion, the network analyzer is essential to the systolic network in that it provides information about its key performance parameter - the bit error rate (BER). For the systolic network to operate correctly, the BER must be very low. This requirement has two basic reasons: (i) If the BER is too high, the probability of

appearance of more than one token in the data stream increases and may lead to loss of synchronization, and (ii) Time consuming retransmissions may also lead to pipeline overflow (loss of synchronization). For example, with fibre-optic links that exhibit 10^{-14} (BER) [66] or a high noise immunity triaxial cable with serial error correction scheme, the error rate can be accommodated by the self-timing design of the UNE. The net result of noise coupled to the network should appear as equivalent to self-timed data transfers. Since at each UNE the token-passing controller has a tally of various communication errors, they can be polled through the diagnostics layer to obtain the noise signature of the network.

An extensive survey of research work (both experimental and theoretical) on Newhall loop performance when data can be entered to the network randomly is provided in [59], pp. 13-31 and 53-72. The performance models are based on queueing theory and are compared to simulation of such a loop network. Unfortunately no literature is available to compare those models with actual Newhall loop implementations. In the systolic network, random entry of state vectors to the network never occurs because of the integrated sensors oriented systolic conversion method. The performance of a systolic network is determined during the design process and is fixed for the pertinent application thereafter.

A design methodology for a systolic network will be presented in the next chapter.

CHAPTER VII

SYSTOLIC DAS DESIGN METHODOLOGY

The systolic DAS design methodology presented in this chapter consists of two key components: (i) An object oriented architecture that defines all system building blocks (objects), their associated set of attributes and the run time primitives for manipulation of objects; and (ii) A design framework for top down analysis/synthesis of a systolic DAS.

Object oriented architectures [67],[68] are commonly used in operating systems. The key advantages of such architectures are: (i) The designer picks up only the pertinent objects for the embodiment of the solution to his problem, and does not have to be intimately familiar with all of them; (ii) The designer uses the appropriate set of primitives for his application; (iii) Design modifications and enhancements affect only the pertinent objects and do not propagate elsewhere; and (iv) System decomposition is natural and follows the objects hierarchy. Let us describe the object oriented architecture for the systolic DAS.

7.1 Object Oriented Architecture for Systolic DAS

We will follow a bottom to top description of the objects that constitute a systolic DAS, their set of attributes, and some of the primitives to manipulate them.

7.1.1. Integrated Sensor

An integrated sensor (IS) can be characterized by the following set of attributes: (i) Measured phenomenon; (ii) Transfer function P(f) of the IS; (iii) Compensating phenomenon; and (iv) Gain and offset vectors.

7.1.2 Integrated Coherent Sensor

An integrated coherent sensor (ICS) can be characterized by both the set of attributes, as specified in Sec. 7.1.1, and the set of coherent frequencies.

7.1.3. Systolic Converter

The set of attributes of a systolic converter (SC) includes: (i) Vector; (ii) Array; (iii) Coherent; (iv) Strobed; (v) Real; (vi) Imaginary; (vii) Number of sensors; (viii) Sensor types; (ix) T_{cs}, T_{ps}, T_{ns} sets; and (x) Gain and offset correction matrices. It should be noticed that the set of attributes depends heavily on the specific application of the SC. No attempt has been made here to limit the number of attributes nor their common appearance in the description. This is left to the SC designer. Furthermore, we believe that more classes of SC will be discovered as a result of future design work.

The processing algorithm of the state vector and the systolic converter architecture cannot be described with a limited set of simple attributes and must follow VLSI design methodology such as that

described in [37].

Primitives exist to manipulate the following SC characteristics: (i) Set T_{ps}, T_{ns}, T_{cs}; (ii) Load $\underset{\sim}{S}$ and $\underset{\sim}{G}$ matrices; (iii) Initialize SC; (iv) Obtain diagnostics data; and (v) Set configuration. Again, the set of primitives is not complete and will be expanded as a result of future SC design work.

7.1.4. Unified Network Element

A UNE can have one of the following attributes: (i) Station; (ii) Gateway; (iii) Loop Controller; and (iv) Network analyzer. As Network Analyzer, it may be configured as one on the three network entities. The selected configuration together with the UNP completely define the UNE. If an SC is included in the UNE configuration, a pointer to its set of attributes must be provided to complete the description.

The following is a set of primitives: (i) Configure UNE; and (ii) Provide a network signature command set if the UNE is configured as Network Analyzer.

7.1.5 Systolic Network

The systolic network requires the following set of attributes: (i) A description of network topology in terms of levels, loops, network entities placement and Network Analyzer hook-up points; (ii) Comput-

ational structure of the network identifying centralized, distributed or systolic modes for various loops and network entities; (iii) Network systole T_{ns}; (iv) Bit rate for each loop; and (v) Compensation parameters for network entities (as a function of their placement) to be used for network systole compensation algorithms.

The following set of primitives will be used to manipulate the network: (i) Isolate a loop; (ii) Isolate a specific network entity; (iii) Change T_{ns}; (iv) Update network topology; and (vi) Obtain network signature from a node, loop, level or network entity.

The object oriented architecture described above is used with a top-down framework as a basis for construction of a systolic DAS.

7.2 Top-Down Design Main Guidelines

The following steps should be generally followed to obtain the detailed architecture of a systolic DAS:

a) Verify that the process is vector oriented and cannot be monitored properly with any scalar DAS;
b) Find the scope of the monitored area. Identify a GMA and partition it into MA;
c) Specify the state vectors and their required processing at each MA;
d) Identify the network topology as a result from the GMA partitioning, and their required processing and hierarchy;

e) Specify each UNE in the network;

f) Specify the set of integrated sensors for each SC;

g) Derive the SC architecture for each station;

h) Specify global state vector processing and obtain T_{ns}. Calculate T_{GP} and check if the pipelining of processing and conversion can be satisfied.

Neither the object oriented architecture nor the design framework should limit the freedom of the designer of a vector oriented process monitoring system. They should be treated only as a kernel developed to ease the design complexity. The following chapter will analyze a test case to further emphasize the design methodology.

CHAPTER VIII

CASE ANALYSIS - CANDU REACTOR SHUT-DOWN SYSTEM

The systolic shut-down system architecture described in this chapter has been developed according to specific functional requirements for an intelligent safety system for the future generation of CANDU reactors [6].

8.1 Functional Requirements

The timing requirements for the shut-down system are depicted in Fig. 8.1(a). A rise in reactor criticality will be reflected in the spatial neutron flux [33]. The optimal number of self powered neutron flux sensors [33] is 50. The number of neutron flux sensors determines the maximum allowable power of the reactor as well as the required response time for the shut-down system. Since the insertion of the sensors into the core reduces the full power, an optimum operating region is derived through a simulation which determines the number of sensors, shut-down response time and maximum allowable power.

The advantage of an intelligent safety system is that the reactor can be operated at a higher power maintaining the overall safety requirements. As shown in Fig. 8.1(a), the total response time of the shut-down system should be less then 50 msec, including conversion of sensors pulse streams and processing of the obtained data.

For ultra high reliability of the intelligent safety system, triple redundancy is required. This leads to three sets of sensors (a

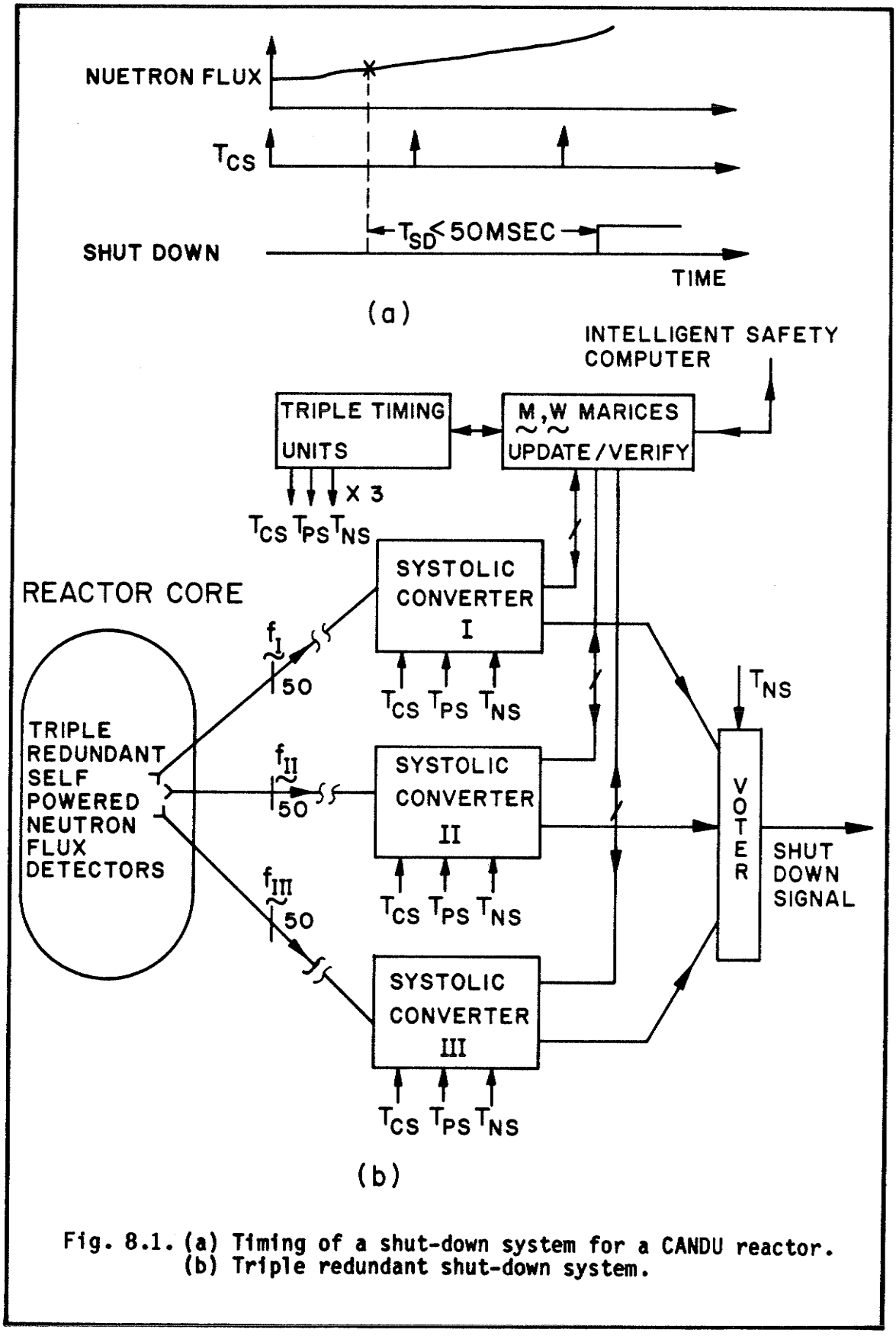


Fig. 8.1. (a) Timing of a shut-down system for a CANDU reactor.
(b) Triple redundant shut-down system.

total of 150 in the reactor core) and three individual shut-down systems. A 50x50 weighting matrix $\underset{\sim}{W}$ with a band width w=20 reflects the spatial distribution and interaction of the sensors. A mask matrix $\underset{\sim}{M}$ (with elements that are either zero or one) is also provided to conduct various logic functions on the data and to facilitate sensor masking or logical grouping. The updating of the matrices occurs a few times each day and should not disturb the shut-down system.

8.2 Top Down Design

We will follow the design procedure described in Sec. 7.2.

a) The process is vector oriented as a result of the requirement of no data skew from the neutron flux sensors in order to represent the flux spatial distribution in the reactor;

b) The monitored area is the reactor core. No further partitioning is required;

c) The state vector consists of 50 elements, each representing flux at given measurement point in the reactor core. Following the requirement for pipelining conversion and processing, the conversion time is set to 20 msec. The total processing time should be less than 20 msec. The processing requirements are identified in Sec. 4.4.1;

d) A loop architecture (dual redundant) will connect all three UNEs;

e) A UNE configured as a station is specified to accommodate the SC;

f) A modification to the self-powered neutron flux sensor is required to obtain fixed amplitude pulse stream. For a T_{cs}=20msec, maximum

output frequency should be set to 5.10^5 Hz to obtain a 4 decade dynamic range in the conversion counter. The modification will incorporate a V→f converter following the low current amplifier [33];

g) The specific SC architecture suitable for this application was described in Sec. 4.4 and its response time was obtained. Since the functional requirements call for triple redundancy, we shall now expand the individual shut-down system into a triple redundant system, as depicted in Fig. 8.1(b).

Each of the three systolic converters is connected to a set of modified neutron flux sensors. A triple redundant timing unit distributes T_{cs}, T_{ps}, T_{ns}. The shut-down signals from three systems are fed into a majority voter clocked by T_{ns} generating the shut-down signal.

Since each SC is a part of a station, the matrices $\underset{\sim}{M}$ and $\underset{\sim}{W}$ could be loaded to the loop slave. The loading procedure to the SC will follow four consecutive steps: (i) The loop slave receives and verifies matrices $\underset{\sim}{M}$ and $\underset{\sim}{W}$ (ii) The matrices are loaded into the non active matrix FIFOs (a swinging matrix FIFO scheme is utilized); (iii) At the end of the current processing, the active and non active FIFOs are exchanged at all three systolic converters via a specific token.

As a result of the top down design, let us identify the shut-down system objects and their sets of attributes:

a) Hybrid radiation sensors for normalized neutron flux sensing based on modification to the self powered neutron flux detectors [33]. The output frequency of the hybrid sensor will span from 1.10^5 Hz to 5.10^5 Hz for a zero to full power normalized neutron flux;

b) The systolic converter will be connected to 50 sensors of the type described in a T_{cs} = 20 msec and $T_{ps} \cong 166\mu$sec. The SC will have the architecture described in Sec. 4.4 with the extensions for on line loading of the $\underset{\sim}{M}$ and $\underset{\sim}{W}$ matrices.

c) Three UNEs are configured as Stations and one UNE as a loop controller. All UNEs will have a dual redundant loop architecture; and

d) The systolic network will consist of a single loop. The bit rate is set to 1 Mbit/sec. The network systole is set to 25 msec to obtain 40 synchronized majority votes per second, allowing a 5 msec margin for the conversion/correction arrays and the loading of $\underset{\sim}{M}$ and $\underset{\sim}{W}$ matrices.

The next section will provide a detailed system architecture for the systolic shut-down system.

8.3 Detailed System Architecture

This section presents a synthesis of the detailed architecture of two objects, the UNE and the systolic converter, developed for the special case of the reactor shut-down system. A five level architectural hierarchy will be utilized in the design process of the objects. The levels correspond to the level of abstraction and are defined as follows: (i) UNE or system level; (ii) Subsystem level; (iii) Unit level; (iv) Chip level; and (v) Cell or chip building block level. The UNE contains the SC as a custom subsystem. We will, therefore, use the three bottom levels (cell, chip, unit) to describe the SC architecture and the top three (system, subsystem, unit) to identify the UNE structure. Although any design of a SC is a special case, common properties and building blocks of the SC will be highlighted.

Figure 8.2 shows the UNE architecture which is based on currently available multiple processors. It is seen that there are two subsystems, the loop slave and the systolic converter. Each of the two subsystems consists of a set of units controlled by a dedicated microprocessor via a local bus. The two or more subsystems also have access to the system bus and system memory through a bus arbitration and controlled access scheme for a multiple processor bus environment, as defined by the IEEE P976 proposed bus standard developed by Intel Corp. [69]. Communication between the two subsystems is established via shared regions in the system memory. As shown by the dotted lines in Fig. 8.2, the two subsystems can be packaged on separate boards

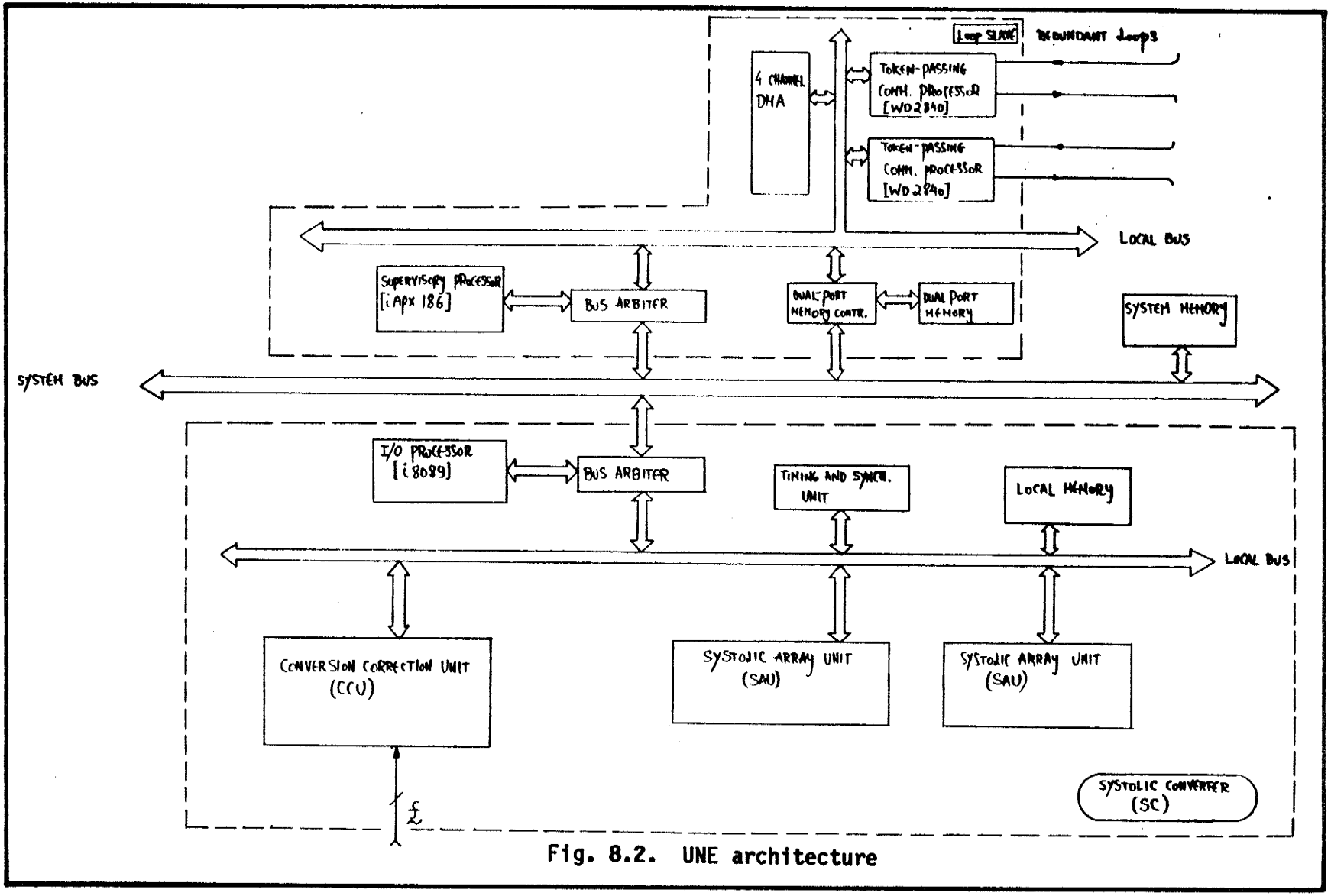


Fig. 8.2. UNE architecture

compatible with the P976 bus standard with control microprocessors facilitating local and system bus arbitration within the multiple processor environment. Let us now describe the two subsystems at the unit and partially chip levels.

The loop slave subsystem is controlled by an iAPX 186 supervisory microprocessor over a local bus. Two WD 2840 token passing communication processors [51] are used to implement the link layer of a centralized systolic network at 1 MBit/sec. A four channel DMA controller is utilized for memory-based transfers to the dual-port memory, which is also accessible through the system bus by the SC control processor.

The SC subsystem is controlled by a special 8089 input/output microprocessor (IOP) capable of performing intelligent DMA transfers as well as control of high speed peripheral units and chips. The IOP executes the specific control and data transfer functions for the following SC units: (i) The conversion/correction unit (CCU); and (ii) The systolic array units (SAUs). It is important to realize that the IOP does not participate in any of the processing functions that reside within the units. Furthermore, the IOP executes the control and data transfer tasks under the supervision of the iAPX 186 within the loop slave subsystem.

We will now follow a bottom to top approach to derive the detailed architecture of the SC by using the cell, chip and unit levels as well as by identifying a unified storage and synchronization scheme .

8.3.1 Dual Pointer RAM Based FIFO

A FIFO memory was already proposed for various functions within the following systolic shut-down building blocks:

a) A systolic conversion/correction cell with local storage of the offset and gain vectors;
b) Properly ordered storage of the $\underset{\sim}{W}$ and $\underset{\sim}{M}$ matrices for the linearly connected systolic array; and
c) An interface between two self-timed elements designed as synchronous systems such as the conversion/correction unit and the systolic array, or the two systolic arrays.

Let us consider two common architectures for the FIFO memory in order to select the one most suitable for the systolic shut-down architecture. Figure 8.3(a) depicts the control and data flow of the first architecture, the fall-through FIFO (FTF). Data are stored in a series of internal registers with control flip-flops to keep track of used stack locations. Data in the registers are propagated to the bottom of the FTF whenever the next register is empty. The FTF architecture enables two distinct synchronous subsystems to maintain a unidirectional data flow at different instantaneous data rates. Such a scheme is shown in Fig. 8.3(b) which also depicts the control signals and data flow associated with an FTF and two communicating subsystems. Figure 8.3(c) shows a data flow oriented FTF circuit symbol with input and output control and data ports.

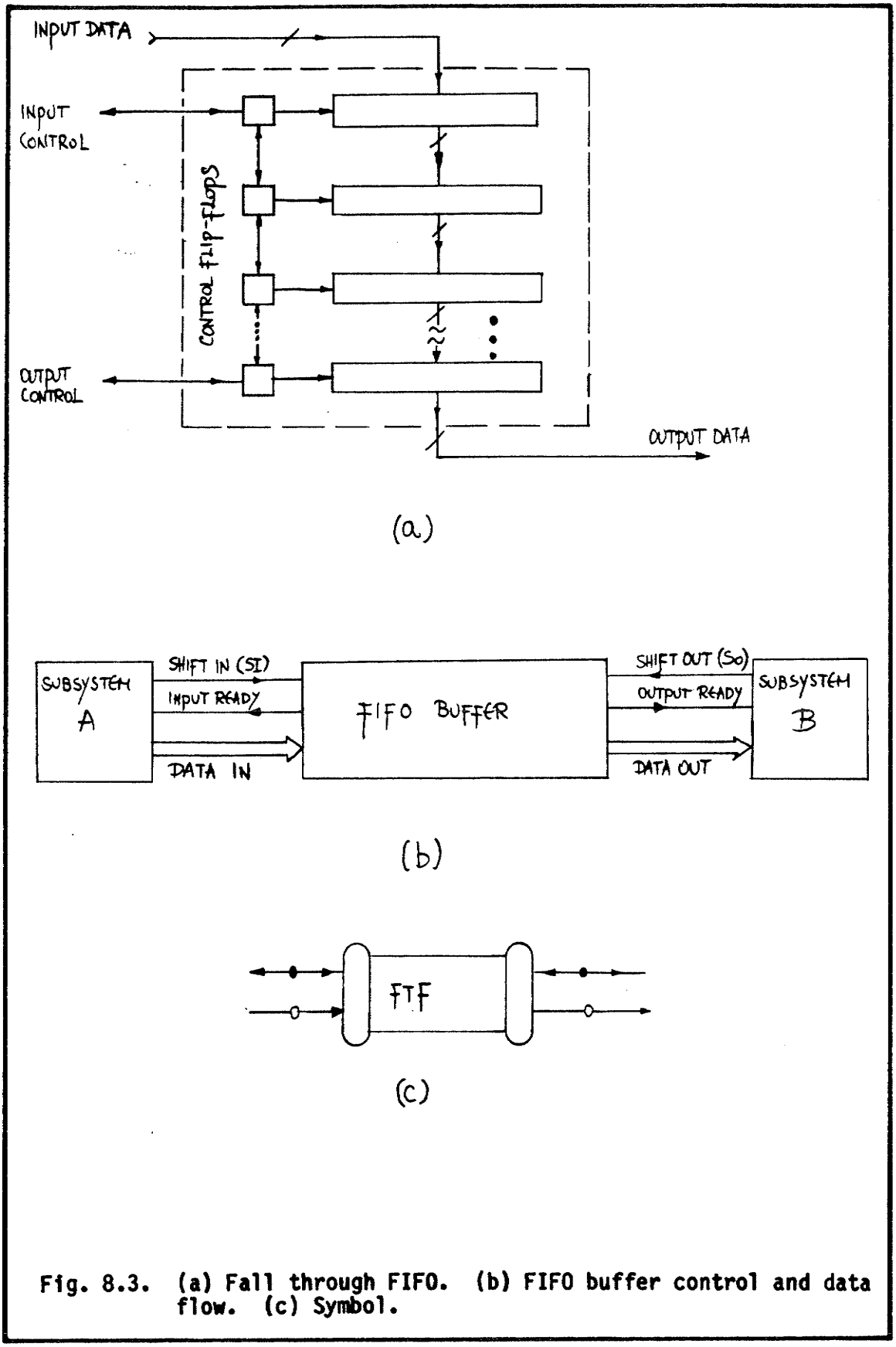


Fig. 8.3. (a) Fall through FIFO. (b) FIFO buffer control and data flow. (c) Symbol.

The FTF architecture described above has the following drawbacks: (i) The fall through delay is a function of the number of empty register locations between the entry port and last entry on the queue; (ii) For a large FTF the fall through delay may become significant thus limiting system performance; and (iii) If data has to be maintained in the FIFO, a recirculation path and external circuitry is required.

A different structure of the FIFO buffer capable of removing the above drawbacks is shown in Fig. 8.4(a). The RAM used in such a structure can be viewed as a circular buffer in which its top address is followed by its bottom address. A read pointer (RP) and a write pointer (WP) are utilized for address generation, while a status counter (SCN) monitors their relative clockwise displacement. Initialization of the buffer resets all pointers to zero. The two pointers and counter are incremented or decremented in a cyclic manner modulo buffer length. A write operation consists of the following sequence of events:

a) Check if SCN is smaller than the buffer length to avoid overwriting; and
b) Write into buffer at address WP, then increment WP and SCN.

A read operation consists of the following sequence:

a) Check if SCN > 0 to avoid reading of invalid data; and
b) Read from memory at address RP, then increment RP and decrement SCN.

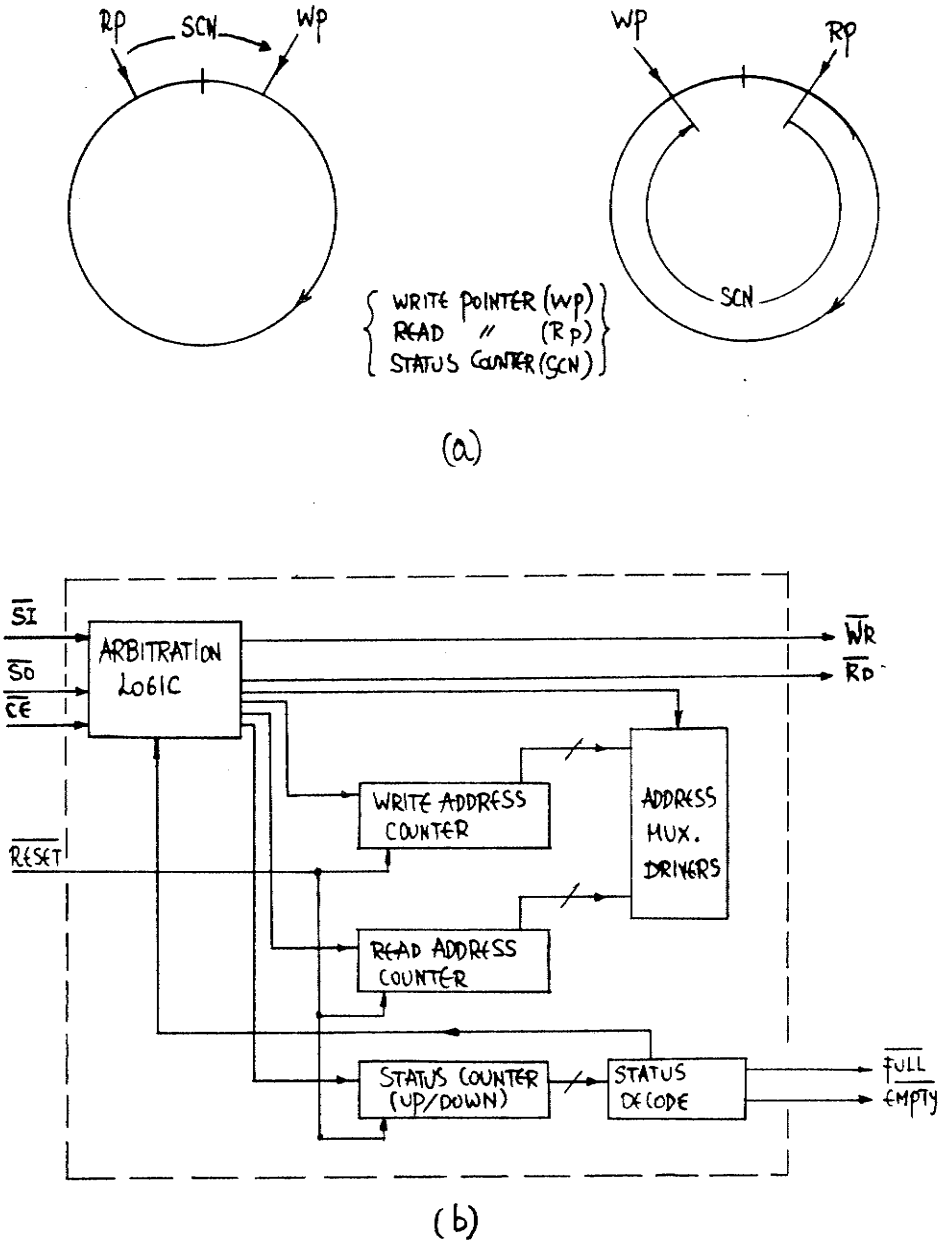
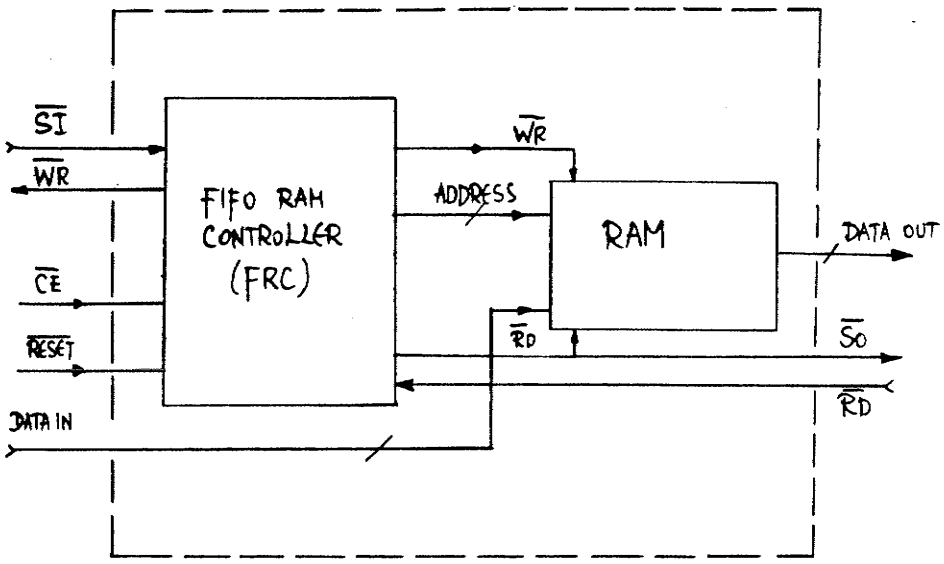


Fig. 8.4. (a) Dual pointer RAM based FIFO. (b) FIFO RAM controller architecture.

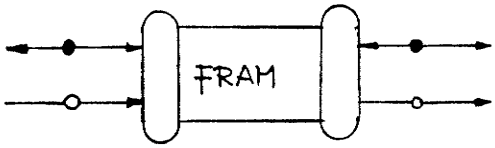
In order to effect proper operation of such a dual-pointer RAM-based FIFO, a controller shown schematically in Fig. 8.4(b) can be used. Three counters are used for the WP, RP, and SCN. The counters can be set to zero by the RESET signal. The arbitration logic resolves collisions between shift-in (SI) and shift-out (SO) requests on a first-in first-served basis, while generating the WR and RD pulses to acknowledge the subsystems involved in the data transfers. The controller also generates FULL and EMPTY status signals.

The FIFO RAM controller (FRC), together with the RAM buffer, form a unidirectional control and data flow construct. The composite construct will be referred to as a FIFO RAM (FRAM)(see [40]), and will be used extensively in the synthesis of various architectural building blocks. Detailed control and data flow of such a FRAM is depicted in Fig. 8.5(a). We observe that the usual FIFO type control signals and data flow are fully preserved, except for the optional FULL and EMPTY status lines. The similarity to the FTF is further emphasized by the FRAM circuit symbol (see Fig. 8.5(b)) which high- lights the two port access and unidirectional data flow.

The FRAM has the following key properties: (i) No fall through delay; (ii) Data is maintained in the FIFO without the need for a recirculation path; (iii) For a large FIFO, the FRAM is more cost effective than the FTF since no control is required for each register; (iv) No delays on reset; and (v) The counter-based FRC is well matched to the counter-based conversion/correction cell. For these reasons, the FRAM will be utilized throughout the synthesis process of the detailed architecture.



(a)



(b)

Fig. 8.5. (a) Detailed control and data flow of a RAM based FIFO buffer. (b) Symbol.

8.3.2 Conversion/Correction Unit (CCU)

In this section, we shall describe the conversion/correction cell with emphasis on the loading and utilization of the offset and gain vectors. A conversion/correction chip will be identified. We shall also show the conversion/correction unit and its interface to the systolic array unit via a FRAM which will facilitate buffering, pipelining and synchronization between the self timing elements.

8.3.2.1 Conversion/Correction Cell (CCC)

The CCU is synthesized following a bottom to top approach through the cell, chip and unit levels. In order to design a conversion/correction cell (CCC), we have to provide a mechanism for loading, verification and local storage of the offset ($\underset{\sim}{S}_j$) and gain ($\underset{\sim}{G}_j$) vectors associated with each CCC. We observe that since both vectors have the same number of elements (for L segments), the elements of the two vectors can be interlaced in order to control the composite local storage by utilizing only one FRC. The CCC is shown schematically in Fig. 8.6(a) and emphasizes this design decision. The vectors $\underset{\sim}{S}_j$ and $\underset{\sim}{G}_j$ are interlaced in the RAM, with their elements stored in odd and even locations, respectively. The FRAM read port is used to load the counter section with L matched pairs of offset and gain vector elements, and the write port is used to load the FRAM, with the two interlaced vectors. After loading the vectors into the FRAM, its contents can be verified by reading them through the write port. This

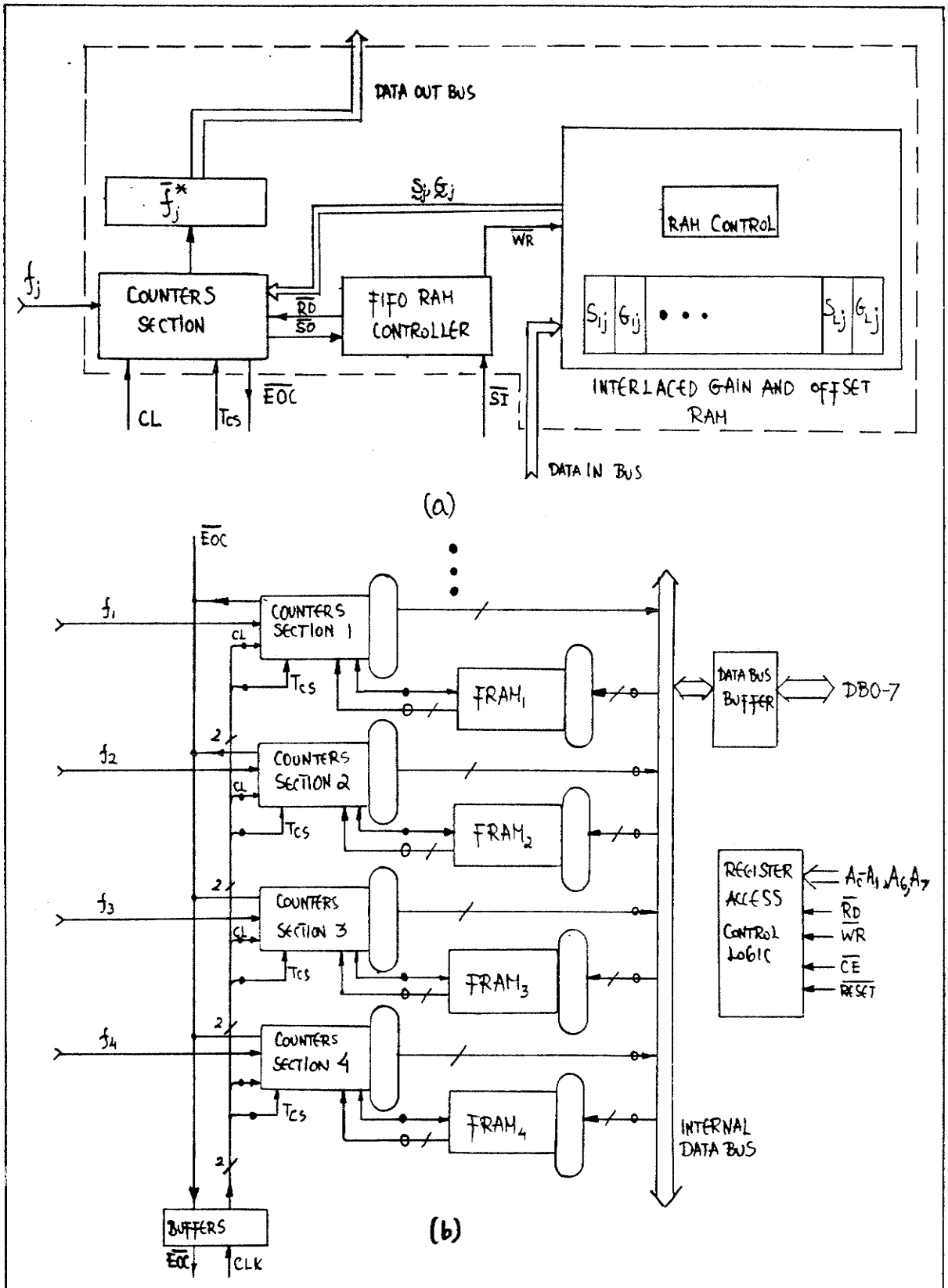


Fig. 8.6. (a) Conversion/correction cell. (b) Conversion/correction chip.

is achieved in a special FRAM mode by inhibiting the other port. Once verified, the FRAM is reset at the beginning of a conversion/ correction cycle, triggered by T_{cs}, and data is accessed by the RP. There is no need for SCN > 0 check. The end of the conversion cycle is indicated by an end-of-conversion (EOC) tri-state line that is pulled low.

8.3.2.2 Conversion/Correction Chip

The conversion/correction cell, comprising of a counter section and a local storage section, serves as the basic building block to form a conversion/ correction chip. The data needed for the conversion/correction process is stored in the FRAM via the write port, while the result of the conversion is stored in the result counter. In order to facilitate DMA to the chip and unit level, we have to design a unified access scheme for extraction of the state vector and for loading and local storage of $\underset{\sim}{S}$ and $\underset{\sim}{G}$ matrices. We shall first analyze the chip level architecture and then expand the discussion to the unit level. The chip will consist of four CCCs with the associated circuitry for controlled access to its ports.

The functional diagram of the conversion/correction chip is shown in Fig. 8.6(b). All counter section result ports containing $\bar{\underset{\sim}{f}}^*$ appear as contiguous locations in memory. The port type is selected by address line A_6, while the specific port is selected by A_0 - A_1. Control and status registers (not shown) are selected by A_7. The reset

input (RESET) clears the counters section and FRAM pointers to zero. EOC line facilitates an extendable wired OR bus for the whole CCU. Figure 8.7(a) shows such a chip, containing four CCCs with additional active line termination and optical isolation for each pulse stream input which could be mounted together on a ceramic substrate to form a conversion/correction hybrid (CCH). With the advent of integrated optoelectronics, it should be possible to integrate the isolation and conversion section on the same chip. Direct coupling of fibre-optic bundles to the chip should solve the severe input pin limitation and signal conversion, addressed later.

8.3.2.3 CCU Decoding Scheme and Data Access Method

A CCU based on 16 conversion/correction hybrids will be described. Although decoding and data access are shown for only 64 channel CCU, the scheme is general and could be expanded for much larger capacity. The utilization of an input-output processor (IOP) in the CCU should not be viewed as an implementation detail since it represents a general data access and synchronization scheme that could be utilized if further integration is attempted.

The CCU consists of up to 16 CCHs, as shown in Fig. 8.7(b). All CCHs are connected to the local bus and are controlled by the IOP. The internal ports of each CCH are accessed via A_0 - A_1, A_6, A_7, while A_3 - A_5 are decoded to generate 16 CE lines for the CCHs. The decoding scheme, once A_6 is selected, assures that the ports (of the two types) appear as contiguous locations in memory and, therefore, can be

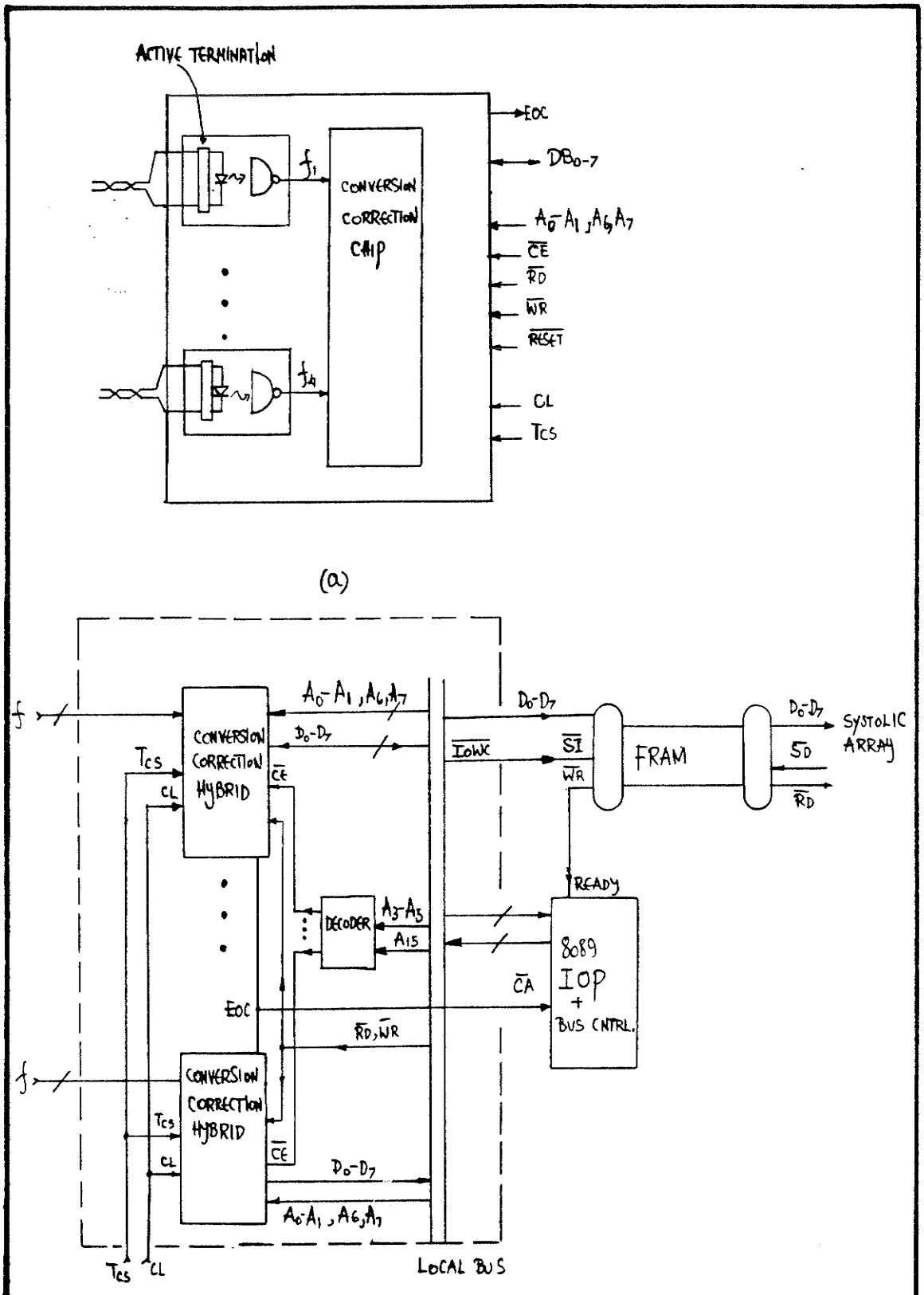


Fig. 8.7. (a) Conversion/correction hybrid. (b) Conversion/correction unit.

accessed using DMA techniques. It should be noted that the FRAM ports are viewed as contiguous locations in memory although they serve as a window for data entry, a concept that will be emphasized in the following section.

At the end of the conversion/correction process, an EOC signal is generated within the CCU. The 8089 IOP recognizes the EOC as a DMA channel attention (CA) signal to initiate consecutive reads from the CCU to the input port of the FRAM. Fig. 8.8(a) illustrates the general window process for fast data transfers between two memories, not necessarily on the same bus, implemented through a channel. For many practical purposes, a channel could be viewed as two back-to-back DMA controllers with a pipelining register. Two types of window processes are depicted in Fig. 8.8(b) and 8.8(c). The horizontal arrow indicates data flow from source (memory A) to destination (memory B). Let us define two pointers, a destination pointer (DP) and a source pointer (SP). If both SP and DP are incremented through the window process, contiguous memory data is transferred from source to destination residing in separate address spaces, as shown in Fig. 8.8(b). If the DP remains unchanged throughout the process while SP is incremented, as depicted in Fig. 8.8(c), data is windowed through a single port.

Since we designed the CCU so that the state vector $\underset{\sim}{\hat{f}}^*$ resides in contiguous locations of the local memory within the address space of the IOP, it could be accessed via a window process.

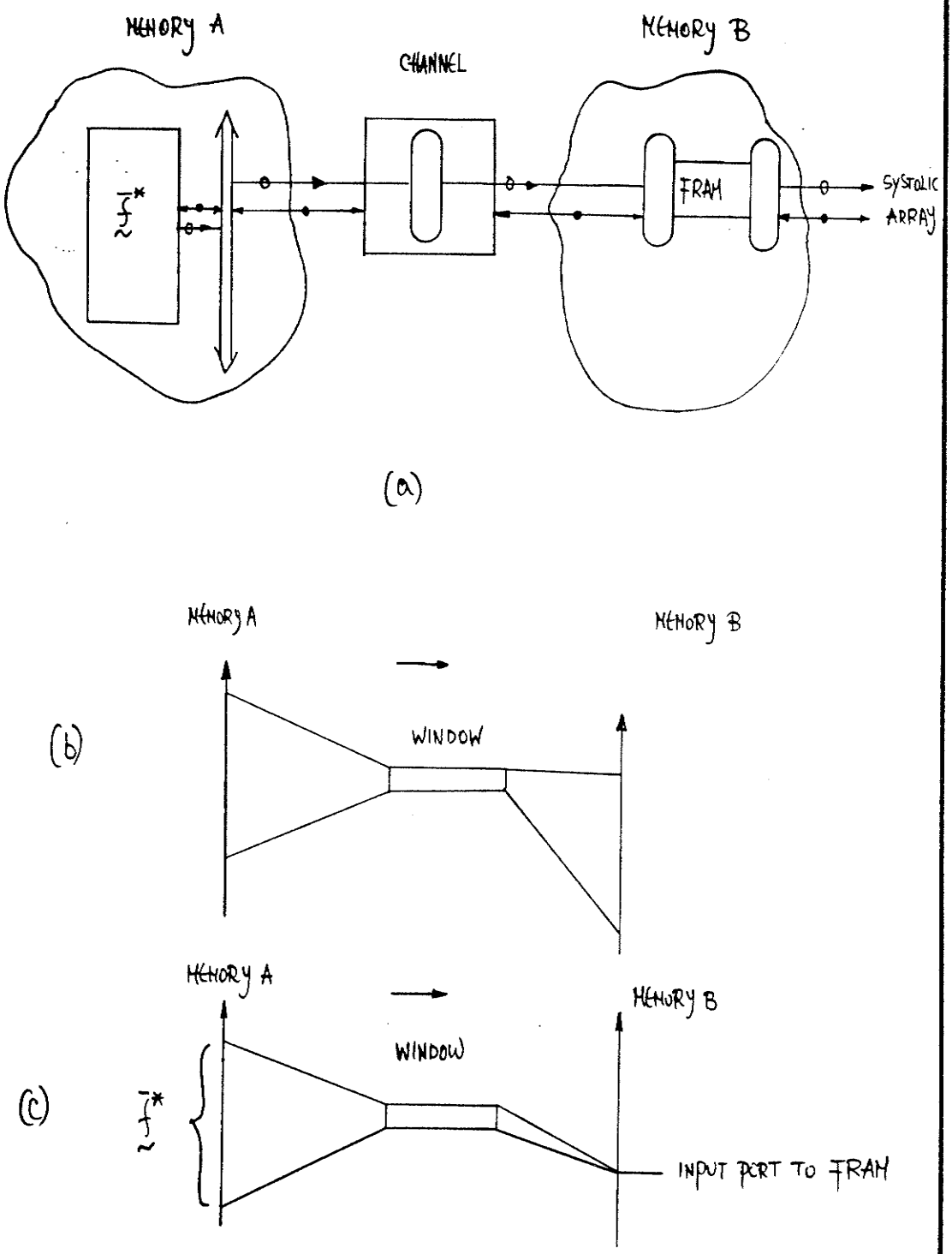


Fig. 8.8. (a) Channeling state vector to FRAM. (b),(c) Window topologies.

In the design of the SC we pipelined conversion/correction and processing. To allow for data flow between the CCU and the systolic array unit (SAU), each designed as a self-timing synchronous element, a FRAM is used as shown at the top right corner of Fig. 8.7(b). Since $T_p < T_{cs}$, elements of the state vector could be windowed to the input port of the FRAM while being extracted by the SAU at a pace set by a computation systole. The conversion/correction could, therefore, proceed in parallel with the computation process.

The interface scheme between the CCU and SAU utilizing a FRAM and a window is implemented via the 8089 IOP. Destination synchronization to the FIFO write port is utilized. The pipelining of processing and conversion is achieved through the window process, initiated by EOC through the CA signal to the FRAM.

8.3.3 Systolic Array Unit (SAU)

In this section we shall develop, bottom-to-top, the systolic processor cell, chip and unit. Again, we will identify the interface of the SAU to the CCU and the access method to its local memory without interruption of the systolic computation flow.

Let us review the required data flow of matrix elements, with a band width w into an array of w linearly connected inner product step processors (IPSPs), as depicted in Fig. 8.9(a). The following two properties should be observed: (i) Odd IPSPs are activated for odd T_{ps}, and even IPSPs are activated for even T_{ps} throughout the computation sequence; and (ii) N matrix elements are needed for each IPSP for 2N

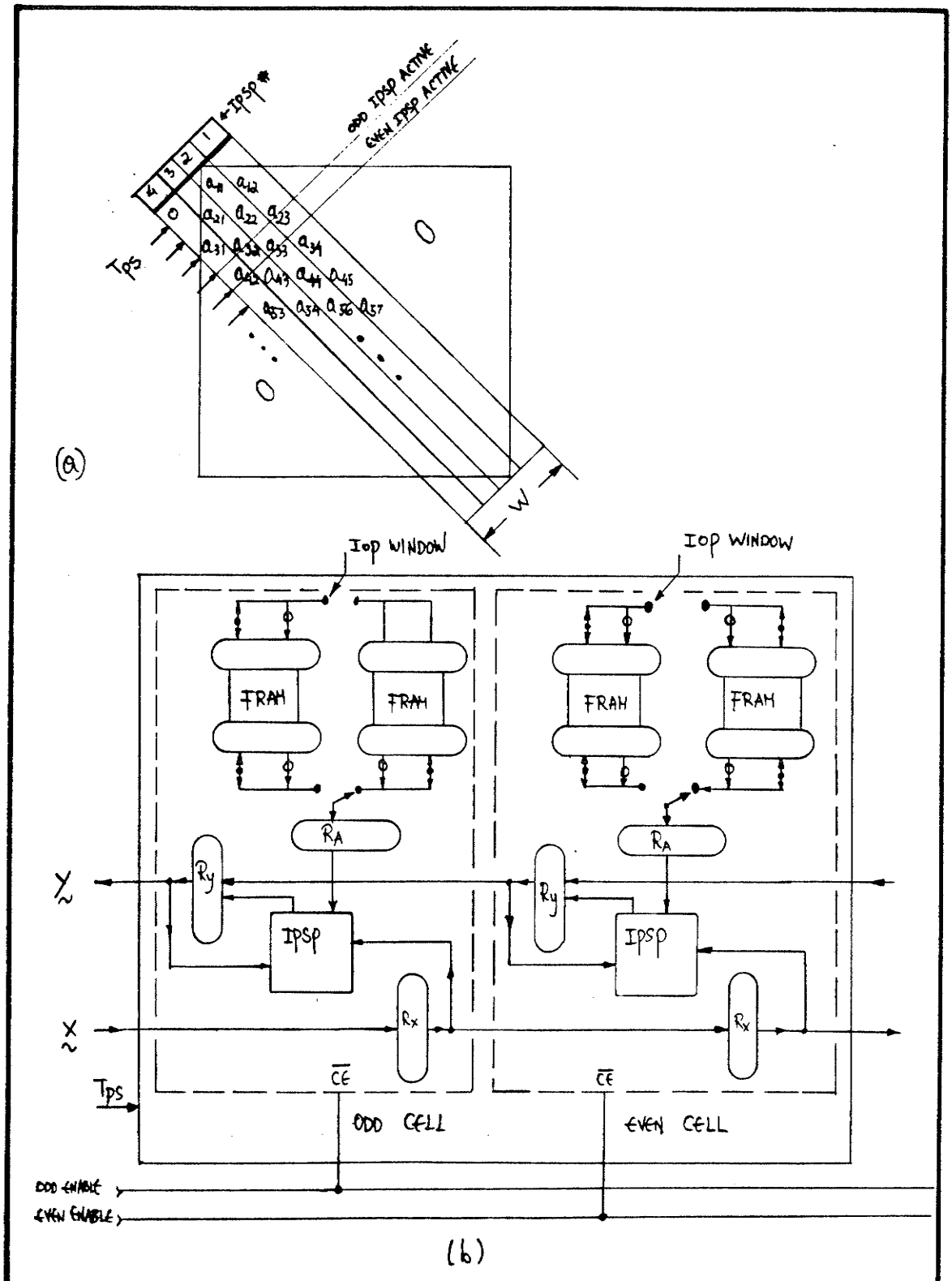


Fig. 8.9. (a) IPSP and memory organization and activation. (b) Functional organization of a composite systolic cell (CSCE).

processing systoles, since a non-activated IPSP implies no data shift of the associated matrix element.

As a direct result of these properties, we could organize an IPSP and its associated FRAM (of length N) in a systolic processor cell, as depicted in Fig. 8.9(b). In order to facilitate loading and verification of a new set of matrix elements without computation disruption, a swinging FRAM architecture is proposed. The IPSP utilizes one FRAM while the other could be accessed through the write port by an IOP window process (including verification as a special mode).

Since only odd or even processors are activated for each T_{ps}, we propose a composite systolic cell (CSCE) comprising of odd and even IPSPs and their associated swinging FRAMs, as shown in Fig. 8.9(b). The activation of the cell is governed by odd enable and even enable separate control lines. The necessary register set for each cell, as well as the data flow and activation scheme are also depicted in Fig. 8.9(b). The entire CSCE could be packaged as a chip for the implementation of the SAU.

Let us review the loading method of a matrix of band width w into a SAU comprised of $\frac{w}{2}$ CSCEs, as shown in Fig. 8.10(a). Each of the N elements associated with a CSCE (either odd or even) is windowed from local memory to the respective FRAM port. Once all w FRAMs are loaded

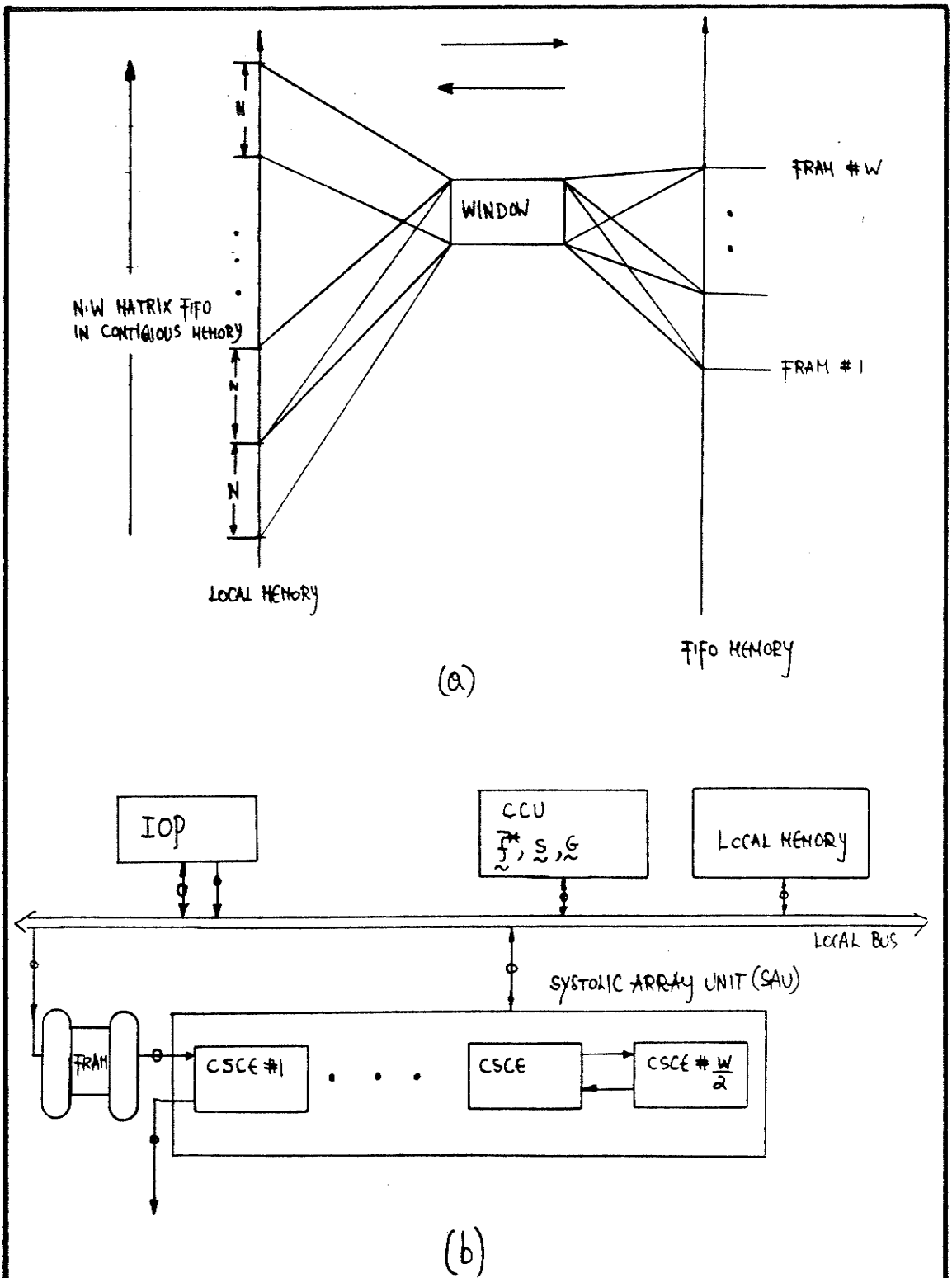


Fig. 8.10. (a) Window scheme for loading systolic array unit (SAU) memory. (b) IOP window control scheme.

and verified, a matrix context switch is initiated (at the end of T_p), assuring non-interrupted computation flow.

Figure 8.10(b) provides a block diagram of the SAU and its interface with the rest of the system. The SAU is composed of $\frac{w}{2}$ CSCEs, and requests state vector data elements through a FRAM. The data is windowed via the IOP from the CCU to the FRAM. All CSCEs FRAM ports are viewed as memory locations in the IOP local memory space.

The first data element of the $\underset{\sim}{W} \times \bar{\underset{\sim}{f}}^*$ operation is available at the SAU output after w steps and is updated every two computation systoles. A FRAM is used to provide the buffering and synchronization between the two SAUs. As a result of the pipelining described above, we observe that the whole computation process is completed in $2N + w + k$ steps where w and k are the band width of the $\underset{\sim}{W}$ and $\underset{\sim}{M}$ matrices.

8.3.4 Cost Function for a Systolic Converter

The systolic converter is composed of three entities: (i) The CCU, (ii) The SAU; and (iii) IOP and control. Let us define a cost function, C, for the SC with w and N as variables. Following the approach of Mead [37, p. 316], C will be expressed in terms of area and time. Based on the chip set that was identified for the CCU and SAU, we can obtain the required area. The area can then be multiplied by the inverse of the computation systole which represents the required computation and memory bandwidth of the IPSP, and normalized by the

number of channels, as follows:

$$C = \left(\frac{N}{4} k_1 + \frac{w}{2} k_2 + k_3 \right) \left(\frac{T_{cs}}{2N+w} \right)^{-1} N^{-1} \qquad (8.1)$$

where $\frac{N}{4}$ is the number of conversion/correction chips, $\frac{w}{2}$ is the number of CSCE and k_1, k_2 and k_3 represent the relative area for the CCU, SAU and IOP chips. After rearrangement, we obtain

$$C = \left(\frac{k_1}{4} + \frac{w}{2} \frac{k_2}{N} + \frac{k_3}{N} \right) (2N + w) \frac{1}{T_{cs}} \qquad (8.2)$$

In order to emphasize the direct dependence on w and, more specifically on the number of CSCE $(\frac{w}{2})$, it is convenient to redefine the cost function as

$$CF(w) \equiv C \frac{T_{cs}}{2} = \left(\frac{k_1}{4} + \frac{\frac{w}{2} k_2 + k_3}{N} \right) \left(N + \frac{w}{2} \right) \qquad (8.3)$$

or

$$CF(w) = \frac{k_2}{N} \left(\frac{w}{2} \right)^2 + \left(\frac{k_1}{4} + k_2 + \frac{k_3}{N} \right) \frac{w}{2} + \frac{k_1 N}{4} + k_3 \qquad (8.4)$$

It is seen that the cost function CF(w) is a parabola. To demonstrate its quadratic increase, the function is plotted in Fig. 8.11(a), for N=50, k_1= 1, k_{21}= 10, k_{22}= 0.1, k_3= 50.

Let us now derive the cost function CF(N) and find an optimal N (N_{opt}) for a given w. The area of CSCE is also a function of N since the memory in the FRAM increases to accommodate more matrix elements. Let us assume that

$$k_2 = k_{21} + k_{22}N \tag{8.5}$$

where k_{21} represents the control area of the FRAM, while k_{22} represents the relative area for a matrix storage element. After substitution in Eq. 8.4, we obtain

$$CF(N) = \frac{(k_{21} + k_{22}\,N)(\frac{w}{2})^2}{N} + (\frac{k_1}{4} + k_{21} + k_{22}N + \frac{k_3}{N})\,\frac{w}{2} + \frac{k_1 N}{4} + k_3 \tag{8.6}$$

or

$$CF(N) = (k_{21}(\frac{w}{2})^2 + k_3\,\frac{w}{2})\,\frac{1}{N} + (\frac{k_1}{4} + k_{22}\,\frac{w}{2})N + k_{22}(\frac{w}{2})^2 + \frac{k_1}{4}\,\frac{w}{2} + k_3 \tag{8.7}$$

For simplicity, we shall write Eq. 8.7 in a parametric form as

$$CF(N) = a\,\frac{1}{N} + bN + c \tag{8.8}$$

The minimum value of N can be found by setting the partial derivative of the cost function, for a given w, to zero.

$$\frac{\partial CF(N)}{\partial N} = -\frac{a}{N^2} + b = 0 \tag{8.9}$$

and, finally,

$$N_{opt} = \left[\frac{k_{21}(\frac{w}{2})^2 + k_3(\frac{w}{2})}{\frac{k_1}{4} + k_{22}(\frac{w}{2})}\right]^{\frac{1}{2}} \tag{8.10}$$

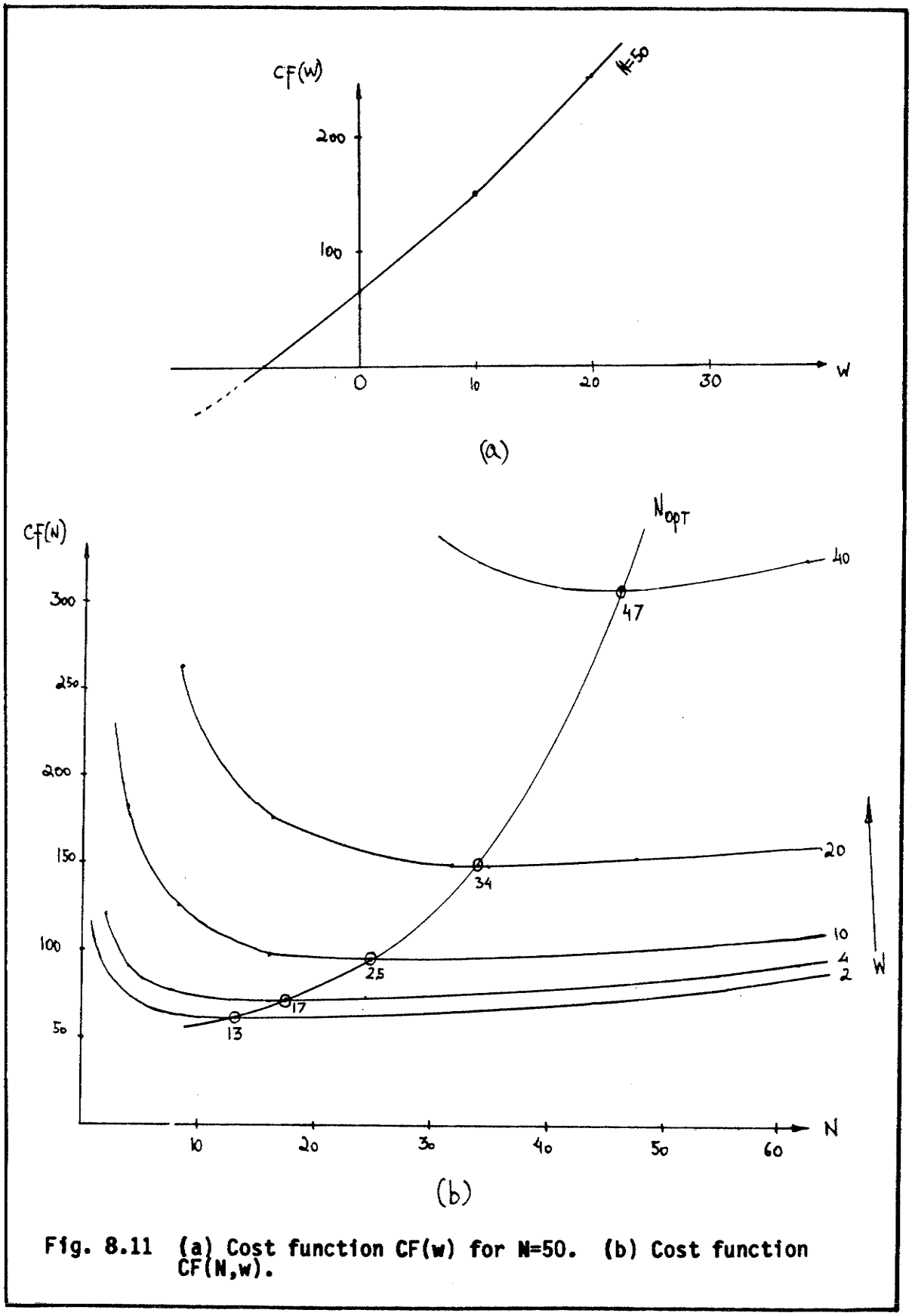


Fig. 8.11 (a) Cost function CF(w) for N=50. (b) Cost function CF(N,w).

The cost function CF(N) with w as a parameter is plotted in Fig. 8.10(b) for k_1= 1, k_{22}= 0.1, k_3 = 50. The optimum number of channels for a given w is indicated. The discussion of the curves is provided in the next section, together with other features of the system just described.

8.3.5 Discussion

A detailed architecture at the cell, chip and unit levels was described in the previous sections. Although the level of integration within the SC is based on the current technology, the following architectural features will be retained with further integration:

a) Utilization of the FRAM as a local storage and synchronization constuct;
b) The design of the conversion/correction and systolic array units as elements communicating via a FRAM;
c) Window process for data transfers between SC units;
d) Swinging FRAM for noninterrupted processing scheme as well as pipelining; and
e) Even/odd composite systolic cell for matrix vector multiplication.

The cost function CF, based on area•time product, was developed for an architecture with limited level of integration. However, the following properties of a CF were observed:

a) For a given number of channels, the cost function is quadratically increasing with w, reflecting the increase in the computation task for denser matrices;
b) For a given w and T_{cs}, a faster systolic array unit is required as N increases;
c) For a given w, we can derive an optimum number of channels for the architecture. This optimum is obtained as a composite of two opposite trends: (i) Distributing the fixed control cost among a larger number of channels; and (ii) increasing computation cost per channel due to the decrease in computation systole;
d) The optimum number of channels increases monotonically with w.

Various problems have to be taken into account when further integration is attempted for various SC chips:

a) The pin count on the conversion/correction chip increases as O(N).
b) The processing and local memory bandwidth of the composite systolic cell increases as O(N) for a given w.
c) Power dissipation within the integrated sensors and conversion/correction cell increases linearly with frequency, thus limiting T_{cs}.

In order to alleviate the above problems, advancements in the fibre-optic sensing and integrated optoelectronics technology are required to enable further integration. The results of such advancements could be viewed as reducing k_1 and k_{22} in comparison with

k_{21} to obtain a larger N_{opt} for a given w.

In the following chapter, we will describe application areas where the design methodology was applied.

CHAPTER IX

APPLICATION AREAS

The systolic DAS design methodology provided in Chapter VII has been applied to various vector oriented process monitoring practical problems. Each of the following sections will be related to a generic application area and will provide a description of the problem and the architectural solution. The objects and key concepts that were utilized, will also be described. This organization will enable us to have an overall view of the vitality of the architecture, while pointing to further extensions that require more research.

9.1 Agriculture

Very large experiments in agriculture research require control and monitoring of many environmental chambers during a long period. Data acquired from various chambers have to be cross-correlated and processed. Control of the chamber's environment must be accurate and very reliable.

A systolic network architecture was conceived, designed, and implemented based on the current microprocessor technology. The following objects were implemented:

a) A μP based UNE reconfigurable into a station, loop controller or gateway. A network analyzer was built supported by a diagnostic

layer for state transition data gathering of all network stations;

b) A UNP for a modified Newhall loop was implemented to support the four basic UNP entities with an additional network signature layer.

The physical layer of the network was implemented utilizing a Z8530 dual communication processor operating at 128 Kbaud with digital phase locked loop. The following features of the systolic network were tested:

a) Line cut detection and phase locked synchronization under fragment mode;
b) Centralized systolic network mode;
c) Token based synchronization of the network;
d) UNE firmware architecture;
e) UNP software architecture;
f) Network analyzer supported by diagnostics layer for remote analysis and testing of network stations; and
g) State transition capturing and logging of the UNP to facilitate network signature.

The network is now a commercial product distributed by an independent company. Some of the design details were taught in a peripheral processors course [70] and are covered in detail in the corresponding project documentation set [71].

9.2 Lasers

A control system for two experimental pulsed HgBr lasers with very high reliability and EMI immunity was required. The architecture for the control system was derived, utilizing the following objects:

a) A 256-channel strobed systolic converter, as described in Sec. 4.5, based on programmable counters and controlled by an LSI/11 processor;
b) A hybrid sensor, as described in Sec. 4.5.2, was built and tested in the presence of three large pulsed lasers;
c) An architecture for enhancing the capacity of a programmable logic controller (PLC) through a strobed SC was provided; and
d) A dual redundant systolic network with centralized processing was designed with extensions for the PLC.

The systolic data acquisition and control system is being implemented at the company site for a continuous 60-day laser test. The architecture is described in [72].

9.3 CAD/CAM

A computer aided design (CAD) and computer aided manufacturing (CAM) architecture for the garment industry was required to fulfill the following objectives: (i) Optimal plant loading at both plant and corporate levels; (ii) Work flow control at corporate and plant levels; (iii) Payroll data collection; (iv) Style dependent production line reconfigurability; (v) Integration of isolated CAD and CAM; (vi) Local and global expansion capabilities; and (vii) Minimum system down-time.

A general system architecture for such a labour intensive application was derived [73], utilizing the following objects:

a) A UNE with modifications to the converter to accommodate user entered data;
b) A multilevel loop architecture with distributed processing of payroll data, and plant loading; and
c) A UNP for both CAD and CAM, allowing for network interfacing of various existing and future computerized tools.

A new concept of plant reconfiguration based on Petri-net theory was introduced, in order to unify the analysis, reconfiguration and presentation of plant loading.

9.4 Nuclear Industry

A radiation monitoring system for a nuclear site has been designed and implemented, as described in Sec. 2.4 and in [5]. The major drawback of the implemented architecture can now be removed by the use of the systolic network, with significant improvement to the system performance. Such a system is described in the next section.

9.4.1 Composite Radiation and Micrometeorology Monitoring System

A vector oriented radiation and micrometeorology architecture for a nuclear site is shown in Fig. 9.1. The systolic network has five levels. Level one consists of UNEs configured as stations, capable of

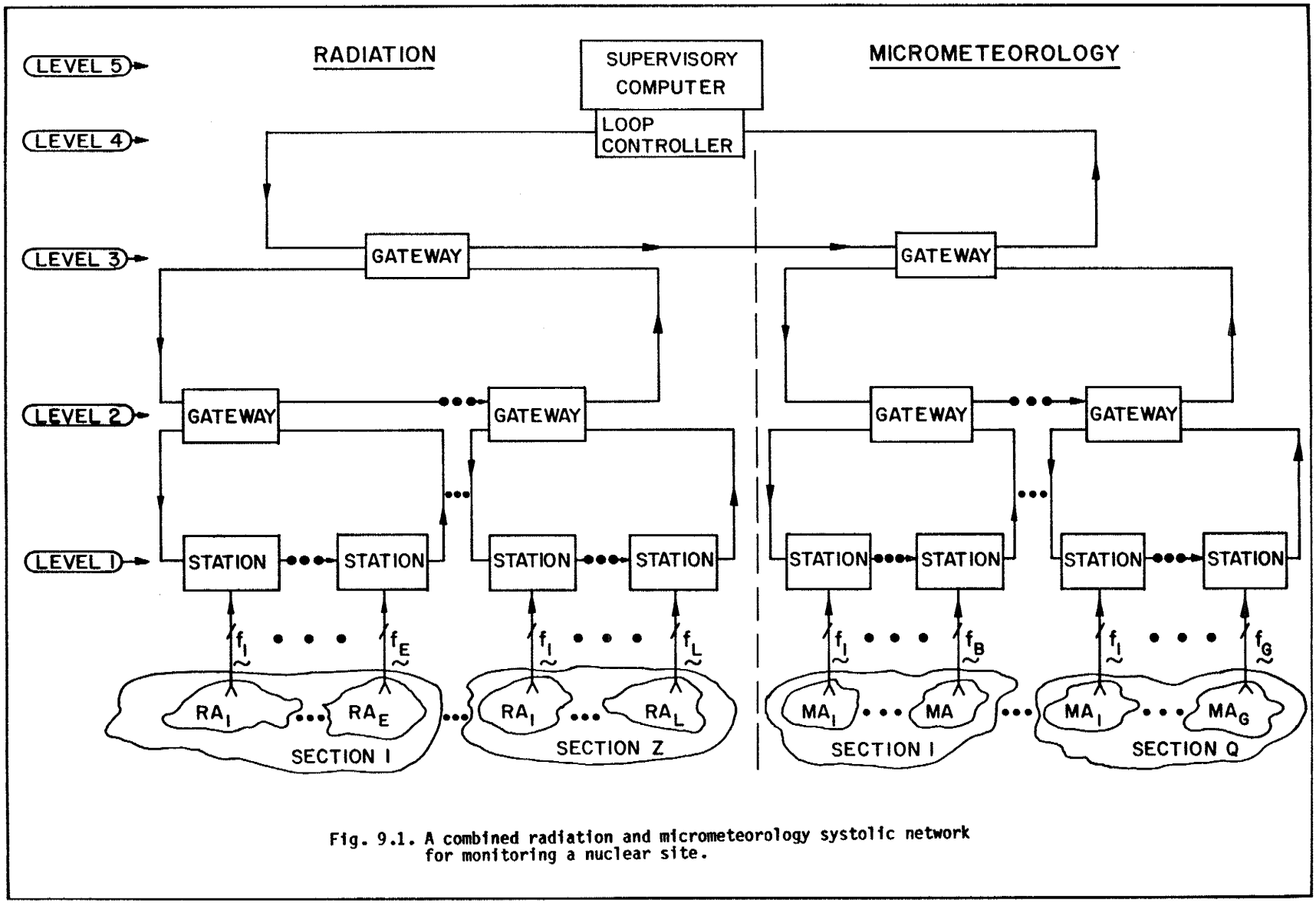


Fig. 9.1. A combined radiation and micrometeorology systolic network for monitoring a nuclear site.

monitoring radiation or micrometeorological phenomena such as wind speed, pressures, temperature, in close proximity to the reactor site. Levels two and three consist of gateways to facilitate partitioning of the monitored site as well as distributed and systolic computation. Level four comprises of the loop controller, while level five consists of a mainframe computer for correlation of radiation and micrometeorology state vectors. Conventional monitoring systems [74], [75], enable the prediction of an evacuation circle around a reactor in case of an accident. The composite system enables the assessment of prioritized evacuation of sectors. Extensions for monitoring nuclear waste storage facilities with better performance than the one currently being designed [76] are mandatory and could use the above architecture.

9.4.2 Reactor Shut-Down System

A systolic shut-down system architecture for a CANDU reactor was derived, as discussed in Sec. 4.4 and in Chapter VIII. The shut-down system could be expanded for a GMA partitioned into MAs, each governed by a systolic shut-down system to form a distributed global shut-down system. The loading of the matrices to various SCs could follow the swinging matrix FIFO procedure that was described in Sec. 8.2. The exchange of all matrices could be controlled by the network systole. Centralized processing will be utilized for the loop due to the data reduction at each UNE. Nuclear waste facilities [76] are also prime candidates for a global shut-down system.

9.5 Steel Industry

An interesting problem in the steel industry is the control of tension within a continuously processed metal bar in a rolling mill plant. Figure 9.2 illustrates the process. A metal bar is formed in between successive mills, located along the metal bar path. Each of the L mills is driven by a DC motor and has an instantaneous shaft speed ω_ℓ where $1 \leq \ell \leq L$. Tension in the metal bar should be minimized in order to maintain the quality of the produced steel.

The architecture for the control system is based on the simultaneous measurement of the velocity profile of the bar along its path as well as the shaft speeds. Obtaining the two state vectors that represent the velocity profile $\underset{\sim}{v}$ and the shaft speed $\underset{\sim}{\omega}$ to form a global state vector of the system, is essential for the optimal control.

The systolic converter producing the velocity state vector could be classified as a strobed and coherent converter. A HeNe laser is strobed to generate a beam that is split into a reference beam and an illuminating beam. The illuminating beam is fed into a bundle of N single mode fibres via a beam splitter. The fibres illuminate the rolling bar at selected points and part of the reflected beam is captured at the matched fibre sensor. The frequency of the reflected light is shifted as a function of the velocity of the bar at the illuminated area due to the Doppler effect [77]. The reference beam and the reflected beams are mixed to produce an N channel pulse stream with an instantaneous frequency related to the corresponding velocity. This operation represents a coherent detection in the frequency domain

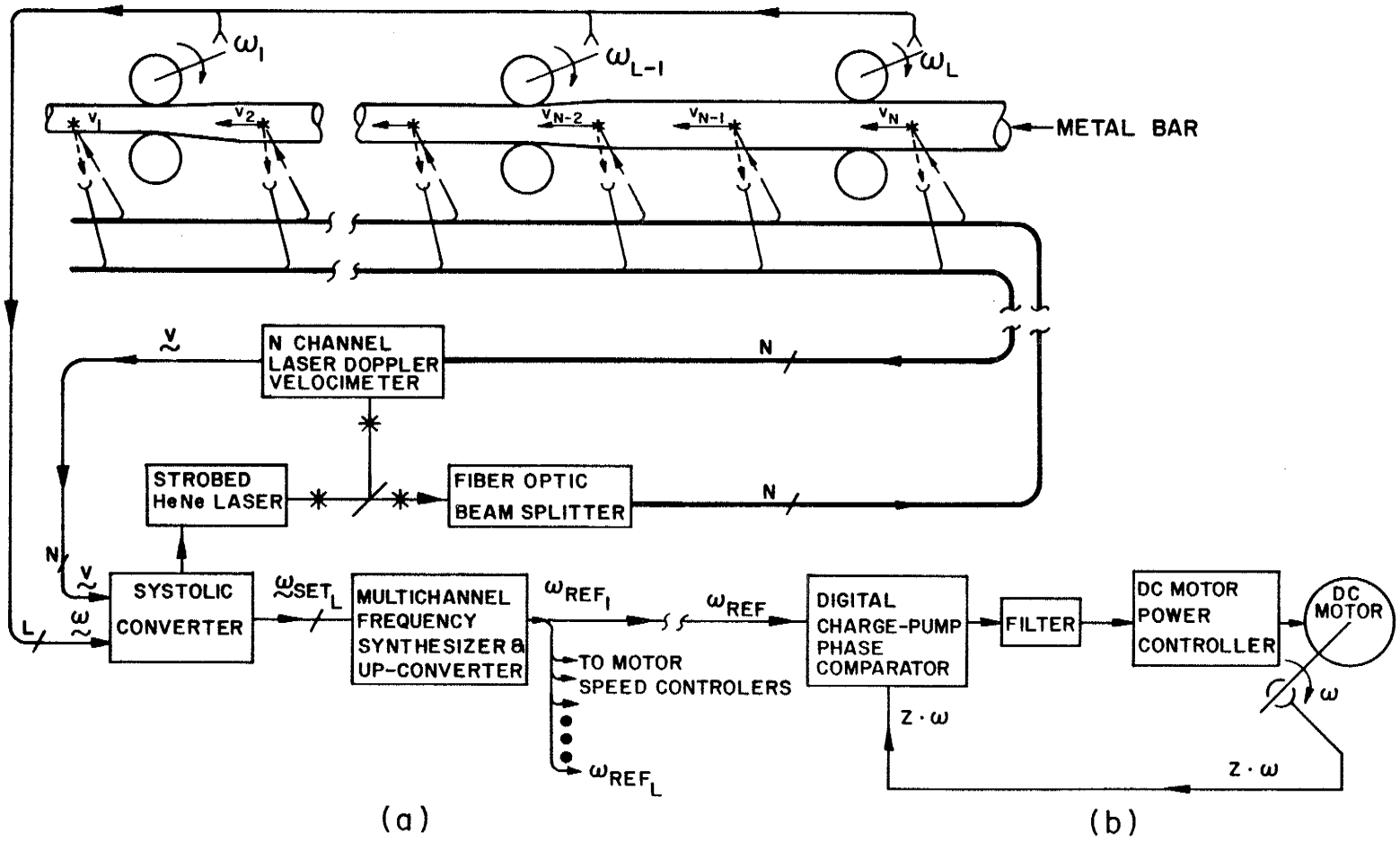


Fig. 9.2. (a) A laser strobed tension control system for a rolling mills plant. (b) Phased locked motor speed control system.

(the single mode fibres that preserve a constant phase shift of the beams are mandatory and should not be replaced by a multimode fibre), resulting in an instantaneous state vector $\underset{\sim}{v}$. The shaft speeds of L motors are picked up (multiplied by Z as a result of a slotted disk) and fed to the systolic converter representing an instantaneous state vector $\underset{\sim}{\omega}$. The systolic converter performs the following functions: (i) Conversion of the global state vector; (ii) Processing the stream of state vectors to generate a control vector $\underset{\sim}{\omega}_{SET}$.

The control vector is fed into an L channel frequency synthesizer and up converter that provides a reference frequency $\omega_{REF\,\ell}$ to each of the L phase locked DC motor speed control systems, as shown in 9.2(b).

The strobing time of the laser, if chosen properly, eliminates any influence of mechanical vibration serving as a noise gate. The strobed and coherent systolic conversion method, described above, expands the description given in Sections 4.5 and 4.6, by showing that they can be combined.

The variety of application areas for the systolic DAS architecture was described in order to prove the vitality of the architecture. Other specific applications have also been attempted with success. We strongly believe that future implementations and research will expand even further the application space and the understanding of systolic data acquisition systems.

CHAPTER X

CONCLUSIONS AND RECOMMENDATIONS

Conventional scalar type data acquisition systems (DAS) have drawbacks which make them unsuitable for data skew sensitive, vector oriented process monitoring applications. It was shown that the solution to the drawbacks may be provided by a new architectural approach and not on mere technological improvement to various DAS building blocks via VLSI. A new architecture was proposed for highly repeatable and accurate integrated sensors based on the non fatigue properties of silicon and a new conversion/correction method, both yielding manufacturing cost reduction and reliability improvement. It was shown that removing the traditional barriers between sensing, conversion and processing leads to a new integrated structure, the systolic converter, that enables high throughput vector oriented process monitoring. The proposed generic systolic converter architectures have been applied in application areas that were either never attempted at all or solved only partially by other DAS architectures. This proves the vitality of the new architecture and its superior performance. Vector oriented data acquisition systems for globally monitored processes were not attempted before. A new network architecture was proposed that allows global synchronization of distributed systolic converters and systolic processing throughout the network. The new systolic network enabled certain systolic algorithms, originally designed for VLSI implementations, to operate on state vectors as data elements. It was shown that the systolic network could

be implemented with modified token passing network architectures.

An effort was made to minimize the number of basic building blocks of the architecture (objects). A general systolic multilevel loop architecture was described with only three basic network entities. A new layer, called a network diagnostic layer, was proposed for the systolic network to facilitate further performance measurements. Based on implementation experience, the diagnostic layer proved itself as a vital tool in system maintenance. A kernel for a design methodology of the systolic network was utilized in various areas and proven as a viable tool in system design, implementation and testing.

The key contributions of this work, in our view, are:

a) Proposed scheme for the removal of the traditional barrier between sensing, conversion, local processing, networking and global processing for vector oriented process monitoring systems to form a unified systolic network architecture;
b) Demonstration that a set of systolic architecture design rules can be extended beyond the VLSI chip in the case of vector oriented systolic networks;
c) Classification of conversion methods and derivation of a generic family of systolic converters;
d) Utilization of silicon for sensing, transmission, conversion processing and networking;
e) An attempt to provide several missing links between seemingly disjoint research disciplines in the context of systolic DAS;

f) Provision of a kernel for a design methodology for vector oriented DAS applicable with current and future technologies; and

g) The use of a deterministic approach to the design and development of systolic networks.

Further research is required to either improve or extend this work. The topics for further research are:

a) The impact of integrated optoelectronics and fibre-optic sensors on the overall architecture, especially on the integrated sensor and systolic converter objects;

b) Extension of the systolic array converter and coherent systolic converter concept to include the newly developed algorithms for the {systolic}x{square or hexagonal} subspace. Special attention should be given to field mapping and imaging applications which demand composite sensing, conversion and reconstruction. We conjecture that the coherent array converter architecture is the kernel for three dimensional real-time computed tomography;

c) Research to obtain an optimal systolic network topology and protocol for global monitoring of an array oriented process, preferably loop based;

d) Extension of the architecture to include deterministic queues of state vectors and matrix - matrix multiplication algorithms;

e) Theoretical development of a systolic network signature analysis;

f) The effects of wide dynamic range sensing on the integrated sensor, systolic converter and systolic network architecture; and

g) Extension of the design methodology for the current and future

architectures.

I humbly believe that this work is only the tip of an iceberg, an expression I can hardly avoid in the cold and sunny Manitoba. We shall vigorously pursue the topics so that the tip outlined in this work will not melt away during the sunny seasons here or elsewhere.

REFERENCES

1. E. R. Hnatek, *A User's Handbook of D/A and A/D Converters,* New York (NY): John-Wiley, 1976.

2. *Integrated Circuits, Converters, Data Acqusition Systems and Analog Signal Conditioning Components,*, Norwood, Mass.: Analog Devices Corp., 1979.

3. *New Developments in Data Acquisition Systems 1980/81,* Norwood, Mass: Analog Devices, 1980.

4. R.J. Higgins, *Electronics with Digital and Analog Integrated Circuits,* Englewood Cliffs (NJ): Prentice Hall, 1983.

5. I. Shpancer and W. Kinsner, "A hierarchical design of a radiation monitoring system," *Technical Report MC82-2,* Industrial Applications of Microelectronics Centre, Winnipeg, Canada, R3T2N2, February 1982, 25 pp.

6. I. Shpancer and W. Kinsner, "A hierarchical radiation monitoring system," presentation to the Electronics, Instrumentation and Control Division, AECL-Chalk River Nuclear Laboratories, Ontario, Canada, April 2, 1982.

7. I. Shpancer and W. Kinsner, "A systolic radiation monitoring system," *IEEE Trans. Nuclear Science,* vol. NS-29, no. 6, pp. 1512-1518, Dec, 1982.

8. I. Shpancer and W. Kinsner, "A systolic, integrated-sensor-oriented, process monitoring system," in *Proc. Sixteenth Asilomar Conference on Circuits, Systems and Computers,* pp. 140-148, (Pacific Grove, CA, Nov. 8-10, 1982).

9. J.A. Zumbado, "Data acquisition system architecture with dedicated sample and holds," *IBM Technical Disclosure Bulletin,* vol.23, no.9, pp. 4210-4213, Feb., 1981.

10. B.A. Jones, "Microprocessor controlled data acquisition instrument for neutron activation measurements," *IEEE Trans. Nuclear Science,* vol. NS-28, no.5, pp. 3819-3821, Oct. 1981.

11. A.C. Capel, G.E. Gilks, R.A. Baso and G. Yan, "A distributed local area network using non-contention protocols," in *Proc. Computer Networking Symposium,* pp. 12-17, (Aneheim, CA, Dec. 15-17, 1981).

12. R.R. Shaw, A.C. Capel and C.F. Penson, "Distributed terminal support in data acquisition systems for nuclear research reactors," *Distributed Systems for Nuclear Power Plants Working Group Meeting,* Chalk River, Ontario, May 14-16, 1980, 12 pp.

13. P.W. Barth, "Silicon sensors meet integrated circuits," *IEEE Spectrum,* pp. 33-39, Sept. 1981.

14. G.F. Knoll, *Radiation Detection and Measurement,* New York (NY): John-Willey, 1979.

15. S.M. Shafroth, *Scintilation spectroscopy of Gamma radiation,* London: Gordon and Breach, 1974.

16. F. Adams and R. Dams, *Applied γ-ray spectroscopy,* 2nd Edition, Oxford: Pergamon Press, 1970.

17. A. Coche and P. Siffert, "Lithium-drifted silicon and germanium detectors," *Semiconductors Detectors,* Amsterdam, Elsevier: North Holland, 1978.

18. A. Lorenz, *A Review of Neutron Detection Methods and Instruments.* UCID-16325, 1973.

19. J.f. Boland, *Nuclear Reactor Instrumentation (In Core).* New York: Gordon and Breach, 1970.

20. J.M. Harrer and J.K. Beckerley, *Nuclear Power Reactor Instrumentation System Handbook.* vol. 1, chs. 2 & 3, TID-25952-PI, 1973.

21. F.E. LeVert and S.A. Cox, "A fully encapsulated directonal self-powered gamma ray detector in in-core nuclear reactor measurements," *IEEE Trans. on Nuclear Science,* vol. NS-27, no. 1, pp. 795-797, Feb. 1980.

22. H. Bock and E. Balcar, *Nuclear Instrumentation and Methods,* pp. 124-563, 1975.

23. V.V. Verbinski and R. Giovannini, *Nuclear Instrumentation and Methods,* pp. 114-205, 1974.

24. A-203-charge sensitive preamplifier discriminator/shaping amplifier, AMPTEK Inc., 6 De Angelo Drive, Bedford, Mass. 01730.

25. J. MacDermott, "Hybrid frequency output transducers provide digital system advantages," *EDN,* pp. 61-69, Nov. 10, 1982.

26. K.E. Peterson, "Silicon as a mechanical material," *Proc. of IEEE,* vol. 70, no. 5, pp. 420-547, May 1982.

27. *Concept and Technologies for Integrated Sensors.* Lecture notes of a one-day seminar, University of California, Berkely, San Francisco, Feb. 8, 1982.

28. *IEEE Trans. on Electron Devices,* vol. ED-26, no. 12, pp. 1880-1836, Dec. 1979.

29. *IEEE Trans. on Electron Devices,* vol. ED-29, no. 1, pp.34-136, Jan. 1982.

30. C.S. Sander, J.W. Knutti and J.D. Meindl, "A monolithic capacitive pressure sensor with pulse period output," *IEEE Trans. Electron Devices,* vol. ED-27, no. 5, pp. 927-931, May 1980.

31. J.M. Borky and K.D. Wise, "Integrated signal conditioning for silicon pressure sensors," *IEEE Trans. on Electron Devices,* vol. ED-26, no. 12, pp. 1906, Dec. 1979.

32. W.R. Daws, "An IC compatible ionizing radiation detector," *IEEE Trans. on Nuclear Science,* vol. NS-28, pp. 4152-4279, Dec. 1981.

33. C.J. Allan, A.M. McDonald and J.M. Cuttler, "The development and use of a travelling flux detector system for CANDU reactor," *IEEE Trans. on Nuclear Science,* vol. NS-28, pp. 720-727, Feb. 1981.

34. L. DeVito, "V→f converters demand accurate linearity testing," *Electronics,* pp. 105-110, March 8, 1982.

35. H.T. Kung, "The structure of parallel algorithms," *Advances in Computers,* vol. 19, pp. 65-108, 1980.

36. J.M. Speiser, H.J. Whitehouse and K. Bromely, "Signal processing applications for systolic arrays," in *Proc. Asilomar Conference on Circuits, Systems and Computers,* pp. 100-104, (Pacific Grove, CA., Nov. 17-19, 1980).

37. C. Mead and L. Conway, *Introduction to VLSI Systems,* Reading, Mass: Addison-Wesley, 1980, pp. 271-292.

38. K. Bromely, J.J. Symanski, J.M. Speiser and H.J. Whitehouse, "Systolic array processor developments," in *CMU Conference on VLSI Systems and Computations,* pp. 273-284, (Carnegie-Mellon Univ., Pittsburgh, PA, Oct. 18-21, 1982). Available from Computer Science Press, Rockville (MA): 1981.

39. L.J. Guibas and F.M. Liang, "Systolic stacks, queues and counters," *Conference on Advanced Research in VLSI,* Cambridge, M.I.T., pp. 155-164, Jan. 1982.

40. J. Seltzer, "Organize RAMs as FIFO buffers with LSI controller chip," *Electronics,* pp. 209-216, Sept. 30, 1981.

41. N. Bar-Haim, I. Uri and A. Yariv, "Integrated optoelectronics," *IEEE Spectrum,* pp. 38-45, May 1982.

42. B.K. Gilbert and L.D. Harris, "Advances in processor architecture, display and device technology for biomedical image processing," *IEEE Trans. on Nuclear Science,* vol. NS-27, no. 3, pp. 1197-1206, June 1980.

43. J. Kline, "Impact of VLSI on medicine," *VLSI Electronics,* vol. 2, in N.G. Einspruch (ed.). New York: Academic Press, 1981, pp. 171-188.

44. E.E. Swartzlender and B.K. Gilbert, "Supersystems: Technology and Architecture," *IEEE Trans. Computers,* vol. C-31, pp. 399-409, May 1982.

45. M. Warpenburg and L.J. Siegel, "SIMD image resampling," *IEEE Trans. on Computers,* vol. C-31, no. 10, pp. 934-942, Oct. 1982.

46. S.Y. Kung, "Wavefront array processor: Language, Architecture and Applications," *IEEE Trans. on Computers,* vol. C-31, no. 11, pp. 1054-1068, Nov. 1982.

47. *IEEE Project 802 Local Networks Standards Committee Draft B,* available from IEEE Computer Society, P.O.B. 639, Silver Spring, MD 20901.

48. W. Myers, "Towards a local network standard," *IEEE Micro,* vol. 2, pp. 28-42, Aug. 1982.

49. W.B. Riley, "Local area networks move beyond the planning stage," *Electronic Design,* pp. 155-164, Nov. 11, 1982.

50. J.L. Peterson, *Petri Net Theory and the Modeling of Systems* Englewood Cliffs (NJ): Prentice-Hall, 1981, pp. 62-69.

51. M. Stieglitz, "Token passing cashes in with controller chip," *Electronic Design,* pp. 181-185, Oct. 14, 1982.

52. C. Weitzman, *Distributed Micro/Minicomputer Systems,* Englewood Cliffs (NJ): Prentice-Hall, 1980, pp. 72-79.

53. P. Zafiropulo, "Reliability optimization in multiloop communication networks," *IEEE Trans. on Comm.,* vol. COM-21, no. 8, pp. 898-907, Aug. 1973.

54. J.R. Pierce, "How far can data loops go?", *IEEE Trans. on Comm.* vol. COM-20, no. 3, pp. 527-530, June 1972.

55. L. Sanders, "Using Manchester code in data communication," *Electronics Design Week Conference Proceedings,* vol. L3, Boston (MA): Hayden, 1981, pp. 77-100.

56. F.M. Gardner, "Charge-pump phase lock loops," *IEEE Trans. on Comm.,* vol. COM-28, no. 11, pp. 1849-1858, Nov. 1980.

57. "Using the 8273 SDLC/HDLC protocol controller," *Peripheral Design Handbook,* Intel Corp., Aug. 1981, pp. 2-272.

58. W.C. Lindsey and C.M. Chie, "A survey of digital phase-locked loops," *Proc. IEEE,* vol. 69, no. 4, pp. 410-431, Apr. 1981.

59. C. Tropper, *Local Computer Network Technologies,* New York (NY): Academic Press, 1981, pp. 11-13.

60. M.T. Liu, "Distributed loop computer networks," *Advances in Computers,* vol. 17, 1978, pp. 171-173.

61. H. Zimmermann, "OSI reference model - the ISO model of architecture for open systems interconnection," *IEEE Trans. on Comm.,* vol. COM-28, no. 4, pp. 425-432, Apr. 1980.

62. A.S. Tenenbaum, *Computer Networks,* Englewood Cliffs (NJ): Prentice-Hall, 1981, pp. 15-21.

63. W.D. Farmer and E.E. Newhall, "An experimental distributed switching system to handle bursty computer traffic," in *Proc. ACM Symp. Problems Optimization in Data Comm. Systems,* pp. 1-34, (Pine Mountain, Georgia, Oct. 1969).

63a. Application Articles on Signature Analysis, Application Note 222-2, Hewlett-Packard Publication 02-5952-7593.

64. A.A.S. Danthine, "Protocol representation with finite state models," *IEEE Trans. on Comm.,* vol. COM-28, no. 4, pp. 632-643, Apr. 1980.

65. L.J. Mekly and S.S. Yan, "Software design representation using abstract process networks," *IEEE Trans. on Software Engineering,* vol. SE-6, no. 5, pp. 420-434, Sept. 1980.

66. Codenet-The Fiber Optic Ethernet Local Area Network Physical Layer Specifications Version 1.0, Codenoll Technology Corp., 1086 North Broadway, Yonker, NY, 10701.

67. Introduction to the iRMX 86 Operating System, Intel Corp. 1982. Order No: 9803124-04.

68. J. Hemenway, "Object oriented design manages software complexity," *EDN,* pp. 141-146, Aug. 19, 1981.

69. IEEE Proposed Microcomputer System Bus Standard-P976 Bus, Intel Corp., Oct. 1980.

70. Peripheral Processors, Lecture Notes, Industrial Applications of Microelectronics Centre (IAMC) Lectures 7-10, IAMC, 1982.

71. 0016-001 Project Design File. IAMC, 1981-1983.

72. I. Shpancer, *Functional Analysis of the Control System for Two HgBr Laser Systems,* IAMC, Feb. 14, 1983, 70 pp.

73. I. Shpancer and W. Kinsner, *Garment Industry CAD/CAM System Architecture,* IAMC, March 3, 1982, 57 pp.

74. L. Battist, "Environmental measurement requirements resulting from the TMI-2 accident," *IEEE Trans. on Nuclear Science,* vol. NS-28, no. 1, pp. 231-244, Feb. 1981.

75. D.P. Serpa, "Emergency monitoring, assessment and response for Diabolo Canyon nuclear plant," *IEEE Trans. on Nuclear Science,* vol. NS-28, no. 1, pp. 236-241, Feb. 1981.

76. R.L. Brodzinski, "In site radiation measurements and instrumentation for monitoring nuclear waste storage facilities," *IEEE Trans. on Nuclear Science,* vol. NS-27, no. 4, pp. 1277-1279, Aug. 1980.

77. B.M. Watrasiewicz and M.J. Rudd, *Laser Doppler Measurements,* Toronto, Ontario: Butterworth & Co., 1976.