FPGA BASED DATA PROCESSOR FOR DISCHARG MEASUREMENT USING PROPELLER CURRENT METERS

A DISSERTATION

Submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

Vith Specialization in Measurement & Instrumentation)

By

SIRAT MOIN UDDIN

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

JUNE, 2005

I.D. No'. MT-295/2005-26/SMU-HKV-VK

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this dissertation entitled, **"FPGA Based Data Processor for Discharge Measurement Using Propeller Current Meters"** in the partial fulfillment of the requirement for the award of the degree of Master of Technology in Electrical Engineering with specialization in Measurement and Instrumentation submitted in the Department of Electrical Engineering, Indian Institute of Technology-Roorkee is an authentic record of my own work carried out during the period from August 2004 to June 2005, under the supervision of Prof. H.K. Verma and Prof. Vinod Kumar, Department of Electrical Engineering, IIT-Roorkee.

I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Dated: 29 June 2005

Place: IIT-Roorkee

Sirat Moin Uddin

This to certify that this project is a result of an original and sincere effort made by the candidate. All the statements made above are true to the best of my knowledge.

Dated: 29 June 2005	Prof. H.K. Verma	Prof. Vinod Kumar
Place: IIT-Roorkee	Dept. of Electrical Engg	Dept. of Electrical Engg
	IIT-Roorkee	IIT-Roorkee

ACKNOWLEDGEMENTS

"Thanks to almighty ***god*** *for helping me in each & every stage of life"*

I take this opportunity to express my sincere gratitude to my guides, **Prof. H.K. Verma** and **Prof. Vinod Kumar**, Department of Electrical Engineering, Indian Institute of Technology-Roorkee for their valuable guidance, support and suggestions at various stages of this research work. I feel privileged to be associated with them and it was a great pleasure in learning the practical aspects of Instrumentation and Measurement field under their aegis and guidance.

I have deep sense of admiration for their innate goodness and inexhaustible enthusiasm. The valuable hours of discussions and suggestions that I had, with them undoubtedly helped in supplementing my thoughts in the right direction for attaining the desired objective. Working under their guidance will always remain a cherished experience in my memory and I will adore it through out my life.

My heartfelt gratitude and indebtedness goes to all the teachers of **M&I** group who, with their encouraging and caring words, constructive criticism and suggestions, have contributed directly or indirectly in a significant way towards completion of this report.

Special, sincere and heartfelt gratitude goes to my **parents** whose sincere prayers, best wishes and encouragement has been a constant source of assurance, guidance, strength and inspiration to me.

I express my thanks to the Xilinx Inc. USA, for providing me the evaluation version of the ISE Web-Package, without which the project development would not have been possible.

Sirat Moin Uddin

ABSTRACT

The dissertation focuses on the measurement of flow using propeller current-meters in open-channels. The propeller current meter, (an adaptation of turbine flow meter) is designed to measure the flow of water in the pipes/channels (closed and open). The dissertation involves the following work:

Firstly, the proper number and location of propeller-current meters in open channel is worked out. As the flow is non-uniform along the width and depth of channel, so a matrix of measuring points is determined. It is then followed by measurement of speed of current meter. The point velocities measured by current meter are then numerical interpolation to calculate the velocity profile along the depth and width of channel. Subsequently, the velocities at these locations are integrated w.r.t to depth and width of channel to obtain the total discharge in the open-channel.

Data acquisition software has been developed for obtaining the data from the flow meters in 'G' programming in LabVIEW environment.

Lastly, the FPGA-based data processor has been programmed and simulated for necessary data acquisition from these current meters, using VHDL as the hardware descriptor language (HDL).

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS AND ACRONYMS

ABEL	Advanced Boolean Expression Language
ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
CPLD	Complex Programmable Logic Device
CSI	Cubic Spline Interpolation
DAQ	Data Acquisition
DIO	Digital Input/Output
Floor-Planning	Process of choosing the best grouping & connectivity of Logic in a design
FPGA	Field Programmable Gate Array
GPIB	General Purpose Interface Bus, IEEE-488 standard
GUI	Graphical User Interface
HDL	Hardware Descriptor Language
IEEE-1394	FireWire Interface Bus
IEEE	Institute of Electrical and Electronics Engineers
IOB	Input –Output Block
IS	Indian Standard
ISE	Integrated Software Environment
ISO	International Organization for Standardization
LCA	Logic Cell Array
LED	Light-Emitting Diode
LUT	Look Up Table
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
MATLAB	Matrix Laboratory
MAX	Measurement and Automation Explorer
NCD	Native Circuit Description
PC	Personal Computer

PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PROM	Programmable Read Only Memory
TCP/IP	Transmission Control Protocol/ Internet Protocol
TTL	Transistor-Transistor logic
USB	Universal Serial Bus
VHDL	VHSIC-hardware descriptor language
VHSIC	Very High Speed Integrated Circuit
VI	Virtual Instrument
VLSI	Very Large Scale Integration

CONTENTS

Page Number.

Chapter 1: Introduction

Chapter 2: Flow-Velocity Measurement in Open-Channel

Chapter 3: Methods for Open-Channel Discharge Computation

Chapter 4: Virtual Instrumentation

Chapter 5: Field Programmable Gate Arrays

Chapter 6: Discharge Calculation

Chapter 7: Data Acquisition and its FPGA Implementation

Chapter 8: Conclusions and Scope for Future Work

Chapter 1

INTRODUCTION

1.1 Introduction

As the hydropower comes from the conversion of energy of flowing water through the turbine into useful mechanical power, and this (power) is converted into electricity with the help of electric generators. The amount of energy captured depends upon the amount of water flowing through the turbine. Discharge being one of the main parameters in the performance testing of turbines at power plants, so accurate determination of the rate of flow or discharge is our main concern. This ***dissertation involves the measurement of discharge using the propeller current meters in an open-channel***. The methods and the techniques applied in such measurements are of great significance for more efficient utilization of capital and the manpower involved.

Discharge measurements in hydropower stations require precise evaluation of flow through turbines, since it has direct influence on turbine efficiency. Like wise, flow measurement is also required for pumping stations on irrigations projects etc. since it has economic consequences.

For the measurement of flow we require flowmeters, which generally, can be classified into several categories namely as mechanical, electromagnetic, ultrasonic, etc [17,19].

Current meters are being used since decades for flow measurement in large pipe diameters apart from rivers and open channels. A current meter is an instrument used to measure the velocity of flowing water or liquid. The principle of operation is based on the proportionality between the velocity of the water and the resulting angular velocity of the meter rotor. By placing a current meter at a point in a stream and counting the number of revolutions of the rotor during a measured interval of time, the velocity of water at that point is determined

The propeller current meter consists of a propeller mounted in a bearing and a shaft. The fluid to be measured is passed through the current meter, causing the propeller to spin with a rotational speed proportional to the velocity of the flowing fluid. A device to measure the speed of the propeller is employed to make the actual flow measurement. The sensor can be mechanically gear-driven shaft to a meter or a more commonly electronic sensor that detects the passage of each propeller generating a pulse.

1.2 Introduction to FPGA's

Field Programmable Gate Arrays, FPGA's are generic devices that contain a vast number of basic digital components. The interconnections between these components are defined by the user. By specifying these interconnections different digital circuits may be realized. FPGA's are ***re-programmable*** devices, making the change from one digital circuits to another as simple as downloading a new interconnections file, greatly facilitating the design and debugging of complex digital circuits. Traditionally, digital logic design has been limited to manual implementation using TTL and small scale integrated components. Using these standard logic devices have proved to be an inexpensive approach, but require "wire-wrapping" and other difficult means of circuit board assembly. As a result, a significant portion of one's effort is focused on complementing and debugging the physical connection between devices. Using the FPGA technology enables the user to perform much more complex digital designs in a timely manner. This is because the design is made in higher-level computer languages like VHDL, Verilog-HDL or ABEL and circuit generation is effectively performed using available software. The number of basic components in single FPGA chip has grown significantly making it possible to realize very complex digital circuits on FPGA's. This coupled with their reasonable cost and their ease of programming and debugging has recently led many firms to adopt this technology, and trend is still going.

1.3 Introduction to Virtual Instrumentation

The rapid adoption of the PC in the last 20 years catalyzed a revolution in instrumentation for test, measurement, and automation. One major development resulting from the ubiquity of the PC is the concept of **Virtual Instrumentation**, which offers several benefits to engineers and scientists who require increased productivity, accuracy, and performance. ***A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments.*** Executable programs made in LabVIEW are called virtual instruments (VIs). LabVIEW is a program used to automate testing and data gathering. It is basically a graphical programming language in which the user can set up the program to manipulate and store data.

1.4 Literature Review

Although lots of papers were reviewed, some of the base papers, scope and analysis, are presented and discussed in brief manner.

Mikhail et al. [1] gives the analysis on the different test methodologies and their applicability to the performance testing of power at St. Lawrence Power plant. An ICMS scheme was developed for the discharge measurement using Velocity-Area method.

Simpson et al. [2] described the various methods for accurate estimation of discharge in a tidal channel. A description of various errors for discharge calculation was carried out.

R.W. Herschy [3] carried out an analysis of uncertainties of current meter measurements under the new international recommendations.

A paper on the comparison of discharge measurements made by the Current Meter and Acoustic Scintillation methods at La Grande-1 plant was presented by Proulx et al. [4]. Conformity between current meter and ASFM between the two was analyzed.

A brief description of data acquisition system for the discharge measurement using OTT Can Bus System was presented by Felder et al. [5]. Basic scheme and a concept of current meter matrix were described.

Rao et al. [6] discusses the process of calibration of current meters for high velocity flow measurement. The paper emphasis the need for re-calibration of the current meters if the flow measurement is to be carried out for high degree of accuracies.

Staubli [7] described the data processing scheme using the N.I. LabVIEW software. The data acquisition was done at Emmenweid power plant at Lucerne.

Advantages of current meter measurements were presented by Kerchan et al. [8].

Nicholas et al. carried out an analysis on the profile exploration using the current meter method and the effect on the calculated discharge was presented in the same [9]. Optimum number of paths for the C.M. measured was also discussed and critically analyzed.

Rus et al. described the flow measurement with large number of current meters are possible. A built in system for the accurate discharge calculation was proposed in the literature [10].

Lemon et al. [11] described the flow measurement using the graphical method using the Romberg integration, with a cubic spline interpolation for measured points.

Gianalberto Grego made a comparative study and analysis of various flowmeters Caneva Generating Power Plant. It also describes the use of J.Coffins Cubic Curves for the calculation of velocity profile in the channel [12].

A paper on the FPGA based data acquisition was proposed and implemented on the chip by Nurdan et al. [13].

1.5 Statement of Problem

The main objectives of this thesis are:

1. To calculate the discharge/flow-rate using the propeller current meter in an open-channel that involves the tasks such as:

a. Study of current meters

b. Study of national & international standards for open-channel discharge measurement.

c. Proper number & locations of current meters are worked out in channel where discharge is to be measured.

d. Develop the GUI for data acquisition in LabVIEW.

2. Design software in MATLAB for calculation of discharge that includes:

i. Discharge calculation by Graphical methods.

a. Through Curve Fitting Method.

b. Through Cubic Spline Interpolation.

c. Power Law Extrapolation Scheme.

ii. Determination of Velocity Profile for each cross-section.

3. Analysis of computer discharge:

a. Discharge calculation by Arithmetic methods.

b. Comparison of arithmetic method with the graphical methods.

c. Uncertainty analysis for Open-Channel Discharge.

4. Design the architecture model for the necessary data acquisition from these current meters using the VHDL simulator (Modelsim) for verifying the result and implementation by Xilinx ISE Web-Package 6.3.

➢ To synthesize this behavior model and get the results by analyzing the synthesis reports generated.

➢ To carry out simulation in Xilinx-Modelsim for the functional verification.

1.6 Overview of the Report

The report is organized in two main parts, the first part extending from chapter 1 to chapter 5, describes the basic theory and relevant information necessary for proper understanding of the thesis. The second part from chapter 6 to chapter 8 describes the proposed scheme, work and analysis carried out in this thesis including the result and conclusions drawn from it.

Chapter 1: In the first chapter, introduction to Current Meters, FPGA's, and Virtual Instrumentation is given. Also, previous work carried out in the research field and the scope of report is presented in a brief manner.

Chapter 2: In this chapter, the basic knowledge of open-channel and principle of operation of current meters are presented. Also, the various methods for the computation of mean velocity in open-channel by the propeller current meters are described.

Chapter 3: This chapter describes the various methods for the computation of open-channel discharge, as recommended by ISO-748 standard and IS-1192.

Chapter 4: Gives a brief introduction to the field of Virtual Instrumentation and its advantaged when compared with the traditional techniques of instrumentation.

Chapter 5: Reviews of FPGA, its architecture and various steps for designing the FPGA is described.

Chapter 6: An improved technique for the open-channel discharge calculation has been discussed in detail. An algorithm for the same is described and a case study is presented.

Chapter 7: Data Acquisition and its FPGA Implementation of our scheme are described.

Chapter 8: Gives conclusions and possibility of future work in continuation to the work presented in this dissertation.

Appendix A: Specification of the DAQ Card, (PCI-6024E)

Appendix B: BNC-Connector Series for E Series DAQ Device

Chapter 2

FLOW-VELOCITY MEASUREMENT IN AN OPEN-CHANNEL

2.1 Basic Concept of Open-Channel Flow

Flow can be classified into open channel flow and closed conduit flow, [25]. Open channel flow conditions occurs whenever the flowing stream has a free or an unconstrained surface that is open to the atmosphere. Flows in canals or in vented pipelines which are not flowing full are typical examples. The presence of the free water surface prevents transmission of pressure from one end of the conveyance channel to another as in fully flowing pipelines. Thus, in open channels, the only force that can cause flow is the force of gravity on the fluid. As a result, with steady uniform flow under free discharge conditions, a progressive fall or decrease in the water surface elevation always occurs as the flow moves downstream.

The actual distribution of flow velocity is generally quite complex. Open channel flow is often laminar or near-laminar, with the different layers moving at different velocities. Flow velocity at the contact point with the channel boundary is low. Typically, the highest velocity flow is located in the center of the flow channel and slightly below the water surface. Fig. 2.1 shows typical velocity profile or vertical velocity distribution under open channel flow conditions.

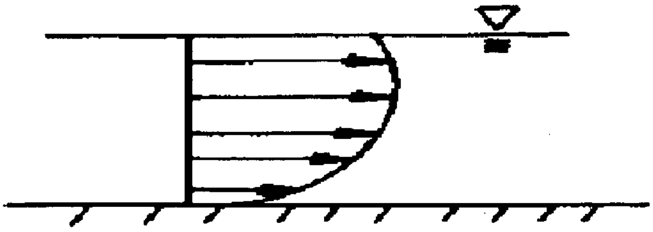


Fig.2.1 Typical Velocity Profile in Open Channel, (Source: www.usgs.com, [25])

A general knowledge of velocity distributions is extremely important in evaluating and selecting a method of flow measurement. Sites with irregular or complicated channel geometries, such as meanders or riffle areas, can cause a decrease in measurement accuracy when using methods that rely on velocity measurements to calculate discharge.

In hydraulics, a pipe is any closed conduit that carries water under pressure. The filled conduit may be square, rectangular, or any other shape, but is usually round. If flow is occurring in a conduit but does not completely fill it, the flow is not considered pipe or closed conduit flow, but is classified as open channel flow. Flow occurs in a pipeline when a pressure or head difference exists between ends. The rate or discharge that occurs depends mainly upon (1) the amount of pressure or head difference that exists from the inlet to the outlet; (2) the friction or resistance to flow caused by pipe length, pipe roughness, bends, restrictions, changes in conduit shape and size, and the nature of the fluid flowing; and (3) the cross-sectional area of the pipe.

2.2 Current Meters

Current meters are kind of ***turbine flow meters*** that measure the flow of water in pipes, open-channels, etc. They are velocity measuring devices that sample at a point. These flowmeters are typically manufactured from austenitic stainless steel but are also available in a variety of material, including plastics. The advantages of these flowmeters are listed below as:

1. Corrosion-Resistant
2. Analog or Pulse Output
3. Wide operating range
4. Wide variety of Electronics available
5. Good Repeatability and Accuracy
6. Cheap and easily available

Each point velocity measurement by the current meter is assigned to a meaningful part of the entire cross section passing flow. The velocity-area principal is used to compute discharge from current-meter data. Total discharge is determined by summation of partial discharges.

2.3 Principle of Operation

A current meter consists of a rotor or propeller mounted on a bearing and shaft. The fluid to be measured in passed through the housing, causing the propeller to spin with the rotational speed proportional to the velocity of the flowing fluid within the meter. A device to measure the speed of the rotor is employed to make the actual flow measurement. The sensor is generally an electronic type sensor that detects the passage of each rotor blade generating a pulse. The principle of the operation is based on the proportionality between the velocity of the water and the resulting angular velocity of the meter rotor.

By placing a current meter at a point in a stream and counting the number of revolutions of the rotor during the measured interval of time, the velocity of water at that point is determined. The number of revolutions of the rotor is obtained by an electrical circuit through the contact chamber. Contact points in the chamber are designed to complete an electrical circuit at selected frequency of revolutions. Contact chamber can be selected having contact points that will complete the circuit twice per revolutions, or once per revolution. The electrical impulse produces an audible click in a headphone or registers a unit on a counting device. The intervals during which meter revolutions are counted are timed with a stop watch.

The current meter is put in the flow with the propeller axis parallel to the flow direction and the propeller peak against the flow. The rotational speed N, (Hz) of the propeller is a linear function of the flow velocity V, m/s in the measuring point;

$$V = (K) * N + B; \tag{2.1}$$

Where;

N: It is the number of pulses counted for a given preset time, Hz

K: Hydraulic Pitch of Propeller, m

B: Characteristics of Current Meter, m/s

The K and B are constants of the respective current meter and have to be determined by calibration tests [16].

2.4 Methods of Determination of Mean Velocities in Channel

The mean velocity of the water in each vertical can be determined by any of the following methods, depending on the time available and having regard to the width and depth of the water, to the bed conditions. The following methods are used to determine mean velocities in a vertical section from the current meters:

- Two-Point Method
- Vertical Velocity-Curve Method
- Subsurface Method
- Depth-Integration Method
- Two-Tenths Method
- Six-Tenths Depth Method
- Three-Point Method, Five-Point method, etc.

2.5 Choice and Classification

The above mentioned methods for the determination of mean velocity in the vertical cross-section are grouped in conformity to IS-1192 and ISO-748, depending upon the time available, width and depth of the channel, bed conditions in the measuring section and the upstream reach, rate of variation of level, degree of accuracy wanted and equipment used.

So, finally these methods are as follows:

a) Velocity Distribution Method (see 2.5.1).

b) Reduced Point Methods (see 2.5.2).

c) Other Methods (see 2.5.3).

d) Integration Method (see 2.5.4).

2.5.1 Velocity-Velocity Distribution method

The measurement of velocity by this method shall consist of velocity observations made at a number of points along the vertical between the surface and the bed of the channel. The spacing of the measuring points shall be so chosen that the difference of velocities between two adjacent points shall be not more than 20 percent w. r. t higher value of the two. The mean velocity of that vertical and its position shall then be determined from the graph. The method is very accurate, depending upon the number of data points measured for profile, but is time consuming and costly. When the turbulent flow condition exists, the velocity curve can be extrapolated from the last measuring point to the bed or wall by calculating $\mathbf{v_x}$ from the equation [18, 19];

$$v_x = v_a\left(\frac{x}{a}\right)^{\frac{1}{m}} \qquad (2.2)$$

where;
$\mathbf{v_x}$ is the open point velocity in the extrapolated zone at a distance x from the bed or wall.
$\mathbf{v_a}$ is the velocity at the last measuring point at a distance a, from the bed or wall.

The mean velocity $\mathbf{v_x}$ between the bottom (or a vertical side) of the channel and the nearest point of measurement can be calculated directly from the equation [18, 19]:

$$\bar{V}x = \frac{m \; * v_a}{m+1} \qquad (2.3)$$

Generally m lies between 5 and 7 but it may vary over a wider range depending on the hydraulic resistance. The value m=5 applies to coarse beds or walls while m=10 is characteristic of smooth beds or walls.

2.5.2 Reduced Point Methods

These methods, less strict than methods exploring the entire field of velocity, are used frequently because they require less time than the velocity-distribution method (7.1.5.2). They are based, however, on assumed velocity profiles.

2.5.2.1 Two point method

Velocity observations shall be made at each vertical by exposing the current-meter at 0.2 and 0.8 of the depth below the surface. The average of the two values shall be taken as the mean velocity in the vertical.

2.5.3.2 One Point-method / Six-Tenth Method

Mean velocity is measured at each vertical by exposing the current meter at 0.6 of the depth below the surface. The value observed should be taken as the mean velocity in the vertical.

2.5.3.3 Sub-Surface Method

The subsurface method involves measuring the velocity near the water surface and then multiplying it by a coefficient ranging from 0.85 to 0.95, depending on the depth of water, the velocity, and the nature of the stream or canal bed. The difficulty of determining the exact coefficient limits the usefulness and accuracy of this method.

2.5.2 Other Methods

This method is further sub-classified into 3 categories for the calculation of mean-velocity. The section 2.5.2.1 to 2.5.2.3 describes the various schemes used for the computation of mean velocity in the vertical transect.

2.5.2.1. Six-Point Method

Velocity observations are made by exposing the current meter on each vertical at 0.2, 0.4 0.6 and 0.8 of the depth below the surface and as near as possible to the surface and the bottom.

$$V_{(mean)} = \mathbf{0.1}^* \left(V_{sur} + \mathbf{2}^*V_{0.2} + \mathbf{2}^*V_{0.4} + \mathbf{2}^*V_{0.6} + \mathbf{2}^*V_{0.8} + V_{bed} \right) \qquad (2.4)$$

2.5.2.2. Five-Point Method

In this method velocities are measured on each vertical at 0.2, 0.6 and 0.8 of the depth below the surface and as near as possible to the surface and the bottom. The mean velocity may be determined from the below eq;

$$V_{(mean)} = \mathbf{0.1} * (V_{sur} + \mathbf{3} * V_{0.2} + \mathbf{3} * V_{0.6} + \mathbf{2} * V_{0.8} + V_{bed}) \qquad (2.5)$$

2.5.2.3 Three-Point Method

Velocity observations are made by exposing the current meter at each vertical at 0.2, 0.6 and the 0.8 of the depth below the surface. The average of the three values may be taken as the mean velocity in the vertical.

$$V_{(mean)} = 0.25 * (V_{0.2} + 2 * V_{0.6} + V_{0.8}) \qquad (2.6)$$

2.5.4 Integration Method

In this method, the current-meter is lowered and raised through the entire depth on each vertical at a uniform rate. The speed at which the meter is lowered or raised should not be more than 5 % of the mean water velocity and should not in any event exceed 0.04 m/s. Two complete cycles should be made on each vertical and if the results differ by more than 10 %, the operation (two complete cycles) should be repeated until results within this limit are obtained. This method is suitable for propeller-type current-meters and cup-type current meters and for electromagnetic current-meters, provided the vertical movement is less than 5 % of the mean velocity. The integration method gives good results if the time of measurement allowed is sufficiently long (60 s to 100 s).

2.6 Conclusion

This chapter describes the fundamental knowledge of open-channel and principle of operation of current meters. Also, the various methods for the computation of mean velocities in open-channel by the propeller current meters were discussed.

Chapter 3

METHODS FOR OPEN-CHANNEL DISCHARGE COMPUTATION

3.1 Discharge or Flow rate:

Stream flow or discharge, is defined as the volume rate of flow of water over a given period of time [17]. In other words it can also be described, as the product of mean channel velocity and the cross-sectional area of the section. Discharge is usually measured in cubic meters per second. When using the propeller current meter, the observations of the width, depth, and the velocity are taken at intervals in the different cross-sections of the channel. In order to measure the mean velocity of flow in a channel, any of the scheme presented in section (2.4) can be used.

The velocity-area principle [18] is used to compute discharge/flow-rate from current-meter data. In this method, the channel is divided into finite number of sub-sections. The partial discharge in each sub-section is computed by multiplication of mean velocity of flow to the area of cross-section. A partial discharge is the product of an average point or vertical line velocity and its meaningfully associated partial area, expressed as:

$$q_n = \overline{V}_n a_n \tag{3.1}$$

Total discharge is determined by summation of partial discharges in the cross-sections and is described as:

$$Q = \sum_{1}^{n} \left(\overline{V}_n a_n \right) \tag{3.2}$$

Represents the computation, where Q is the total discharge, a_n is the individual subsection area, and the V_n is the corresponding mean velocity of the flow normal to the subsection. The cross-sections are defined by depths at verticals 1, 2..., n, where n is the number of verticals in the channel. At each vertical the velocities are sampled by current meter to obtain the mean velocity for each sub-section.

3.2 Site Selection for Open-Channel Discharge Measurement

The selection of a suitable site for making a discharge measurement will greatly affect the accuracy of that measurement. The stream should be straight above and below the measuring section with the main thread of flow parallel to the banks. As a rule, the stream should be straight for at least three channel widths above and below the selected section. The streambed should be free of large rocks, pier, weeds, or other obstructions that will cause turbulence or create a vertical component in measured velocity. Velocity observations are made at each vertical preferably at the same time as measurement of depth, especially in the case of unstable beds. They are made by any one of the standard methods using current-meters. If unit width discharge is required, it is generally computed from the individual observations. In the integration method (2.5.4), the mean velocity is obtained directly. The discharge is computed either ***arithmetically or graphically*** by summing the products of the velocity and corresponding area for a series of observations in a cross-section. The section (3-3, 3-4) discusses the various methods used for open-channel discharge calculation using the propeller-current meters, as recommended by ISO-748 and IS-1192, [18, 19].

3.3 Graphical methods

The basic idea regarding all the methods, described below is that a smooth curve is drawn for the determination of velocity profile along the depth and width of the section. Following are the various methods:

3.3.1 Depth -Velocity Integration

The velocities readings are taken at number of points on the vertical transect. The velocity readings recorded for each vertical are plotted against depth as shown in Fig. 3.1. The area contained by the velocity curve produced for each vertical gives the discharge for unit width or known as ***unit-width discharge*** of the corresponding section. Where necessary, velocity curves can be extrapolated to the surface and bed using the methods described in section (2.5.1).

The values of unit-width discharges ($\bar{v}$. d) are then plotted for each vertical cross-section and joined to form a continuous curve. The area enclosed between this curve and the line representing the water surface gives the total discharge through the section. In the case of velocity measurements by the integration or reduced point methods, the unit-width discharge at each vertical is obtained directly as the product of the mean velocity $\bar{v}$ and the corresponding depth, d. When velocity measurements are not carried out on the same verticals on which the depth measurements are made, the $\bar{v}$ curve shall be plotted across the width of the stream and the value of $\bar{v}$ corresponding to the verticals where depth measurements are made shall be taken for plotting the $\bar{v}$ d curve.

$$Q = \int q_i db \text{ or } Q = \sum \bar{v}_i d_i \Delta B \qquad (3.3)$$

where, Q is the total discharge in the open-channel, m^3/s

$\bar{v}_i$ is the average velocity in i^{th} segment of vertical, m^2/s

d_i is the depth of segment of the i^{th} cross-section in vertical, m

ΔB is the incremental width along the channel, m

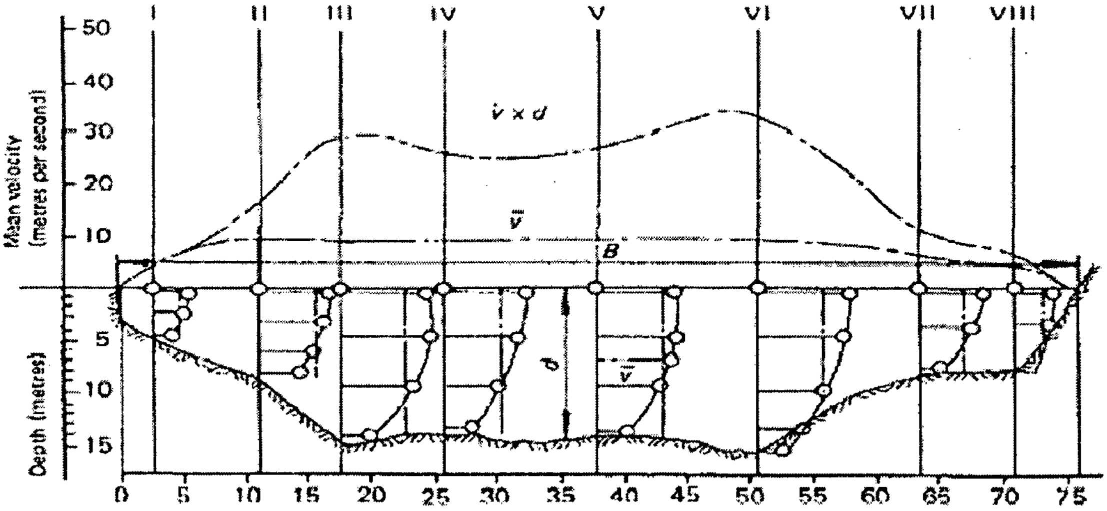


Fig. 3.1: Computation of Discharge from Current-Meter Measurements (Depth-Velocity Integration method), [Source: ISO: 748 [18], IS-1192, [19]]

Fig. 3.1 explains the above mentioned scheme, for the discharge calculation. There are various methods in which the velocity profile in the vertical transects can be plotted [11, 12]. The smooth curve for the same can either be plotted by curve fitting methodology (3.3.1.1) or by using some interpolating scheme.

3.3.1.1 Method of Curve-Fitting

In this method the point velocities at each vertical transect is curve fitted by some polynomial. The order of polynomial is one less than the number of points. The polynomial that have minimum ***norm of residue*** is selected as the best estimate for the velocity profile along the vertical cross-section. The polynomial is then integrated with limits being the water surface elevation and upper limit being channel bed. It gives the partial discharge in the cross-section or more commonly known as ***unit-discharge***. The average velocity in the section can be found out by dividing the unit-discharge by the total channel depth. This unit-discharge is again curve fitted by some polynomial along the channel width. The polynomial is then subsequently, integrated with limits being the two end of channel. The value of integral gives the area under the curve, hence the total discharge in the channel. It is to be assumed for discharge calculations that the velocities at the channel bed and at the two corners of the channel are negligible, hence are taken as zero velocities.

3.3.1.2 Cubic Spline Interpolating Scheme

In this approach, a set of cubic spline functions [11, 12] is used for velocity distribution approximation. Spline functions are partial polynomial functions that are connected in measuring points and have the same second derivative in this point. The algorithm using a set of cubic spline functions is very stable for unlimited number of measuring points and for any distance distribution of measuring points throughout the cross-section (unlike other polynomials of higher power, e.g. Newton‘s interpolation polynomial for non-equidistant spacing etc.). Here, the same procedure is carried out for average velocity measurement, described in section (3.3.1.1), but instead of higher degree polynomial function, the data is interpolated in the interpolating range and extrapolation can be done to find the surface velocity of the channel if the current meter is not placed very close to surface. This method of cubic spline interpolation is considered to be most commonly used for finding the velocity distribution in the vertical transect and horizontal velocity profile along the width of the channel [9, 11].

3.4 Arithmetic methods

The discharge computed by arithmetic methods is further sub-classified as:

1. Mean-section method (3.4.1)
2. Mid-Section Method (3.4.2)

3.4.1 Mean-Section method

In this method, the cross-section is regarded as being made up of a number of segments, each bounded by two adjacent verticals. If v_1 and v_2 are the mean velocities at the first and second verticals respectively, if d_1 and d_2 are the total depths measured at verticals 1 and 2 respectively, and if b is the horizontal width between the said verticals, the discharge of a segment is taken to be:

$$Q = \left(\frac{\overline{v_1} + \overline{v_2}}{2}\right)\left(\frac{d_1 + d_2}{2}\right)b \qquad (3.4)$$

This is repeated for each segment and the total discharge is obtained by adding the discharge from each segment. The additional discharge in the segments between the bank and first vertical, and between last vertical and the other bank, can be estimated from the above equation, on the assumption that the velocity and depth at the banks are zero. If, however, this discharge is a significant proportion of the total flow, then the equation given in section (2.5.1) can be used to obtain the mean velocity in the region of the bank, [18].

3.4.2 Mid-section Method

Assuming straight-line variation of v-d, the discharge in each segment shall be computed by multiplying v-d by the corresponding width measured along the water-surface line. This width shall be taken to be the sum of half the width from the adjacent vertical to the vertical for which v-d has been calculated plus half the width from this vertical to the corresponding adjacent vertical on the other side. The value for v-d in the two half-widths next to the banks may be taken as zero. For this reason, the first and last verticals of a measurement should be as close to the banks as possible if the mid- section method of calculation is used. The computation is carried out at each vertical and the total discharge through the section is obtained by summing these partial discharges as follows, [18]. So partial discharge computed by this method can be described as given in Eq. 3.5

$$Q = \Sigma q_i \frac{b_i + b_{i+1}}{2} \quad or \quad Q = \Sigma \bar{v}_i d_i \frac{b_i + b_{i+1}}{2} \tag{3.5}$$

3.5 Conclusion

This chapter describes the various methods for the calculation of discharge in an open-channel using the propeller current meters. The computation of discharge is, either done through the graphical means or arithmetic means. Both these methods are in conform to the international standard, ISO-748, [18] and national standard, IS-1192, [20].

Chapter 4

VIRTUAL INSTRUMENTATION

4.1 Introduction to LabVIEW

LabVIEW, (Laboratory Virtual Instrument Engineering Workbench) is a development environment based on graphical programming. LabVIEW uses terminology, icons, and ideas familiar to technicians, scientists, and engineers. LabVIEW relies on graphical symbols rather than text-based language to describe programming actions. LabVIEW is also integrated fully for communication with hardware such as GPIB, RS-232, RS-485, and plug-in data acquisition boards like PCI; PCMCIA, USB, IEEE-1394, ISA etc. LabVIEW also has built-in libraries for using software standards such as TCP/IP networking and ActiveX controls etc. The programs that are made in LabVIEW environment are known as Virtual Instruments.

4.2 Virtual Instruments versus Traditional Instruments

Stand-alone traditional instruments such as oscilloscopes and waveform generators are very powerful, expensive, and designed to perform one or more specific tasks defined by the vendor. However, the user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, and the functions available to the user, are all specific to the nature of the instrument. In addition, special technology and costly components must be developed to build these instruments, making them very expensive and slow to adapt. Virtual instruments, by virtue of being ***PC-based***, inherently take advantage of the benefits from the latest technology incorporated into off-the-shelf PCs [21]. These advances in technology and performance, which are quickly closing the gap between stand-alone instruments and PCs, include powerful processors such as the Pentium-IV and operating systems and technologies such as Microsoft Windows XP, .NET, and Apple Mac OS. In addition to incorporating powerful features, these platforms also offer easy access to powerful tools such as the Internet. Traditional instruments also frequently lack portability, whereas virtual instruments running on notebooks automatically incorporate their portable nature [21, 26].

4.3 VI's Software

Software is the most important component of virtual instruments. With the right software tool, engineers and scientists can efficiently create their own applications, by designing and integrating the routines that a particular process requires. They can also create an appropriate user interface that best suits the purpose of the application and those who will interact with it. They can define how and when the application acquires data from the device, how it processes, manipulates and stores the data, and how the results are presented to the user. With powerful software, you can build intelligence and decision-making capabilities into the instrument so that it adapts when measured signals change inadvertently or when more or less processing power is required. An important advantage that software provides is modularity. When dealing with a large project, engineers and scientists generally approach the task by breaking it down into functional solvable units. These subtasks are more manageable and easier to test, given the reduced dependencies that might cause unexpected behavior. Ultimately we can design a virtual instrument to solve each of these subtasks, and then join them into a complete system to solve the larger task. Every ***executable LabVIEW*** program or ***Virtual Instrument*** consists of three main parts [21, 22, 26]:

i. **Front Panel** – How the user interacts with the VI.
ii. **Block Diagram** – The code that controls the program.
iii. **Icon/Connector** – Means of connecting a VI to other VIs.

4.3.1 Front Panel

The Front Panel is used to interact with the user when the program is running. Users can control the program, change inputs, and see data updated in real time. Every front panel control or indicator has a corresponding terminal on the block diagram. When a VI is run, values from controls flow through the block diagram, where they are used in the functions on the diagram, and the results are passed into other functions or indicators. The front panel is the user interface of the VI. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

Fig. 4.1 show the typical block diagram created in LabVIEW environment. It shows the control and indicator palette. Control palette is shown by the two knobs while the indicator palette is shown by the waveform graph.

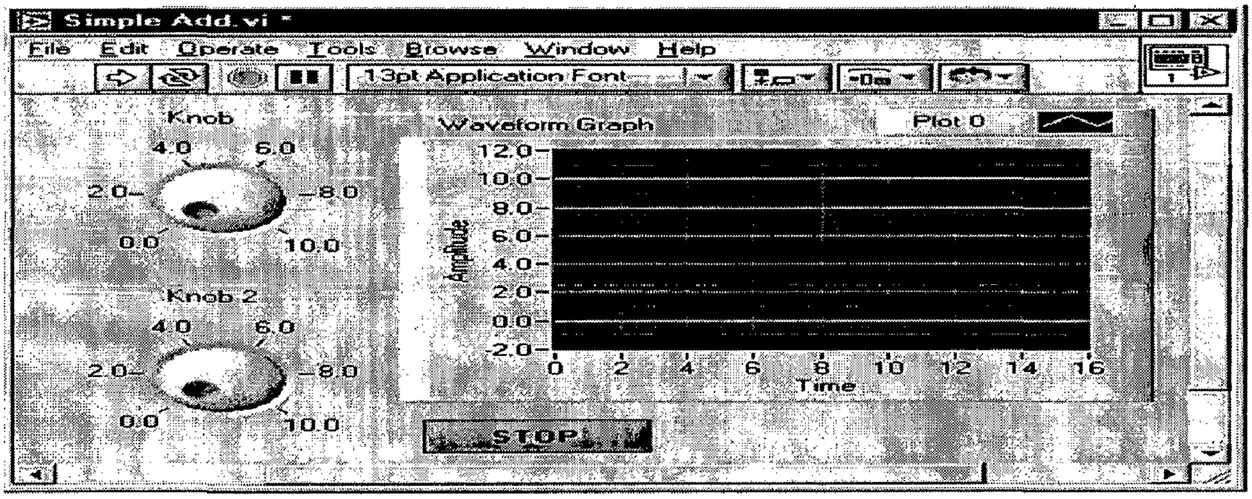


Fig 4.1 Typical VI's Front Panel (Source: NI's LabVIEW software version 6.0.3)

4.3.2 Block Diagram

The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. The VI receives instructions from a block diagram, which you construct in G.

A corresponding terminal of front panel is shown in Fig. 4.2. The knobs and waveform are shown as the two terminals on the block diagram

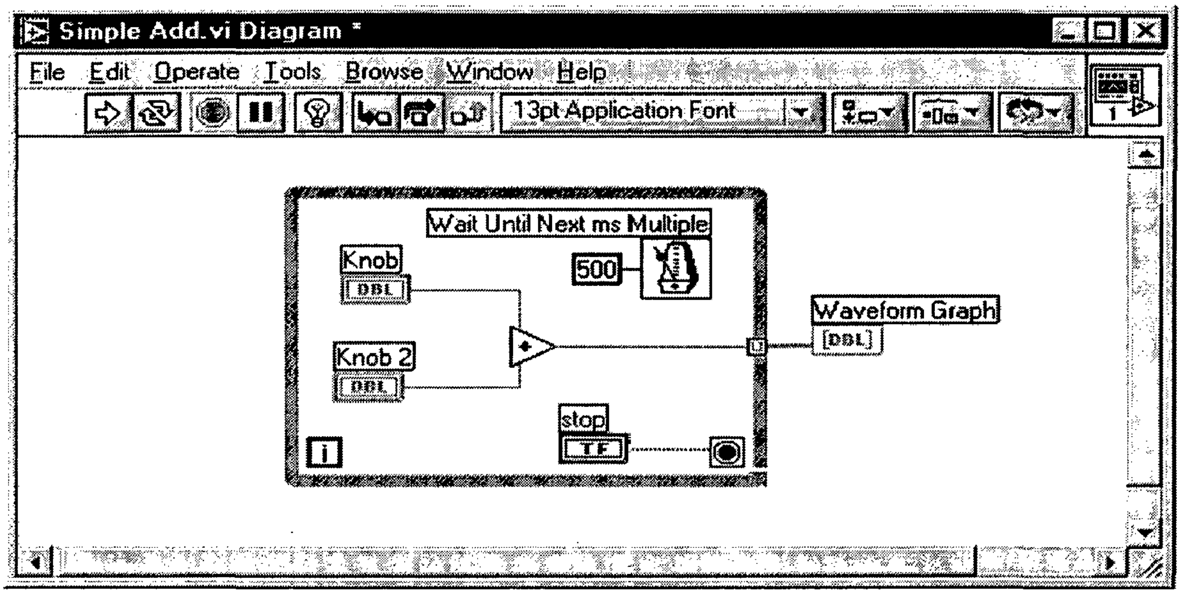


Fig. 4.2 Typical VI's Block Diagram (Source: NI's LabVIEW software version 6.0.3)

4.3.3 Wires

Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

4.3.4 Icon/ Connector

Every VI displays an icon, in the upper right corner of the front panel and block diagram windows. An icon is a graphical representation of a VI. It can contain text, images, or a combination of both. If you use a VI as a sub-VI, the icon identifies the sub-VI on the block diagram of the VI. The connector shows terminals available for transfer or data to and from the sub-VI. There are several connector patterns to choose from.

4.4 Role of LabVIEW in Virtual Instrumentation

LabVIEW is an integral part of Virtual Instrumentation because it provides an easy-to-use application development environment designed specifically with the needs of engineers and scientists in mind. LabVIEW offers powerful features that make is easy to connect to a wide variety of hardware and other software. Hence, finally some of the chief features of Virtual Instruments can be summarized as:

i. Graphical Programming
ii. Connectivity and Instrument Control
iii. Open Environment
iv. Reduces Cost and Preserves Investment
v. Multiple Platforms
vi. Distributed Development
vii. Analysis Capabilities
viii. Visualization Capabilities
ix. Flexibility and Scalability -- Key Advantages

4.5 Conclusion

Virtual instrumentation is fueled by ever-advancing computer technology and offers you the power to create and define your own system based on an open framework. This concept not only ensures that your work will be usable in the future but also provides the flexibility to adapt and extend as needs change. LabVIEW is designed with scientists and engineers in mind, providing powerful tools and a familiar development environment created specifically for the design of virtual instruments.

Chapter 5

FIELD PROGRAMMABLE GATE ARRAY

5.1 FPGA Architecture

An FPGA consist of an array of logic cells surrounded by programmable I/O blocks, and connected with programmable interconnects, as shown in Figure 5.1,[24]. The programmable logic cells are building blocks from which logic circuits are constructed. Their architecture ranges from very simple cells, such as cells that can implement any Boolean function of two one-bit inputs, to complex cells with several inputs and outputs using lookup tables to implement the logic function. FPGA's with simple cells are termed fine-grained FPGA's, while FPGA's with complex cells are called coarse-grained FPGA's. Apart from circuitry for combinational functions, cells also contain one or more flip flops as storage elements. At the periphery of the array of the logic cells, input/output cells establish the link between the logic cells and the external package pins.

FPGA's not only differ in their cell architecture, but also in the routing network, which interconnects the logic cells. The routing network usually provides direct connections between neighboring cells. The routing network is not only used to connect logic cells, but also to connect the logic cells to the input/output cells.

An FPGA is programmed by setting each of its programmable elements (i.e. logic cells and routing networks) to the desired state. Programmable elements for each cell can be programmed via electronic programmable switches. Anti-fuse technique is one of the most commonly used for programming the FPGA's. In the following section we will define the common architecture details of several most popular FPGA's from Xilinx, Altera and Actel Corporations.

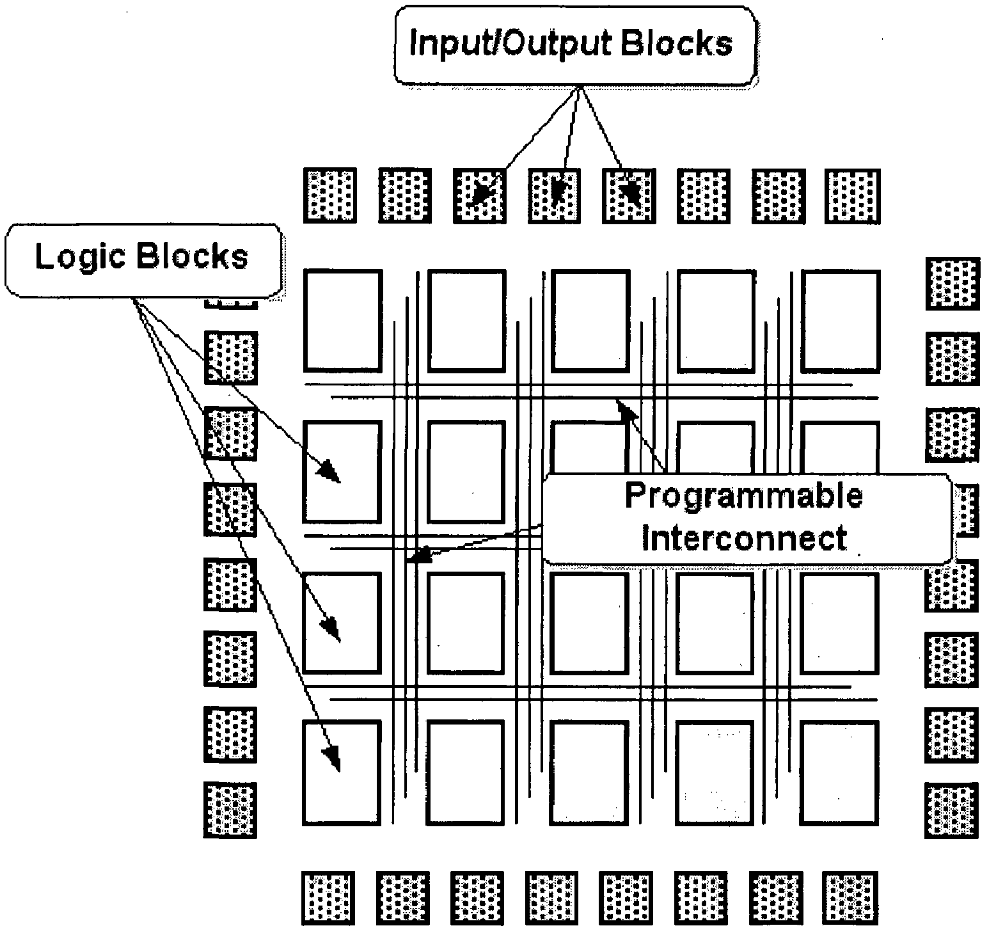


Fig. 5.1 FPGA Architecture (Source: www.xilinx.com, [24])

As shown in Fig. 5.1, three main components that make up FPGA are:

1. **Configurable Logic Block, (CLB)**: Provides the functional elements for constructing the user's logic.

2. **Input/ Output Blocks, (IOB):** It is the interface between package pin's and internal signal lines of the FPGA chip.

3. **Programmable Interconnect**: They provide the routing paths to connect inputs and outputs of CLB and IOB's onto appropriate network.

5.2 Application Areas

- ASIC (Application Specific Integrated Chip)
- Random Logic Implementation
- Proto-Typing
- Reconfigurable Hardware
- On-site Reconfiguration

5.3 Advantages of FPGA's

- Low risk
- Effective Design Verification
- Low Testing Cost
- Future Modifications
- Rapid Turnaround
- Low Tooling cost

5.4 Limitation of FPGA's

- Speed of Circuitry
- Design Methodology
- Chip size & Cost
- Complex Design Software's

5.5 FPGA Design Process

The FPGA Design methodology provides an overview of the steps involved in designing a Field Programmable Gate Arrays (FPGA).

The various steps used in design of FPGA are, discussed in proceeding section and are shown in Fig 5.2.

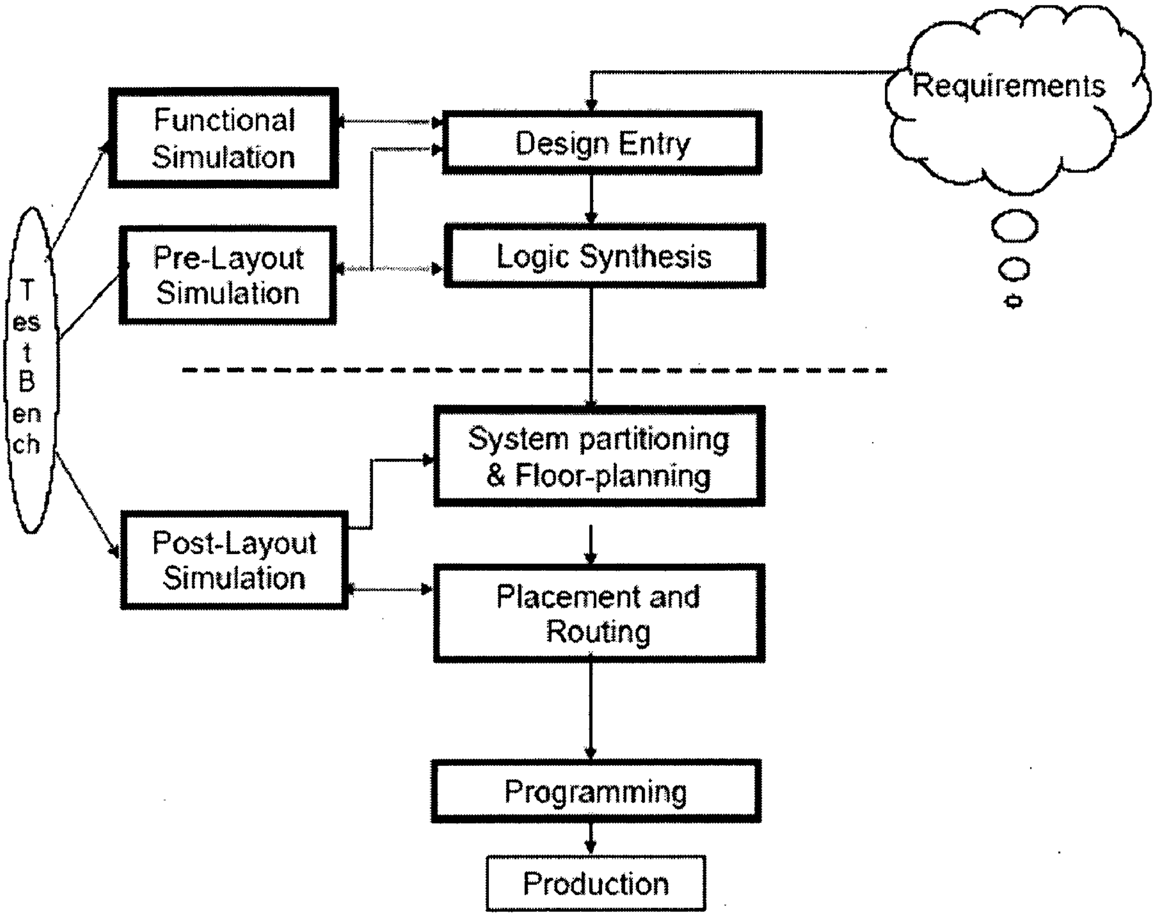


Fig. 5.2 Steps performed in FPGA design (Source: www.xilinx.com, [24])

5.5.1 Design Entry

The designer's first task is to describe the design's intended function. Typically this functionality is specified in a document, such as a functional specification, written in a natural language such as English in order to facilitate its development. Once the specification is finalized, the designer then translates the specification into a form that can be understood by software tools in order to direct the creation of silicon. The two principal design description methods are:

- Hardware Description Languages (HDLs), generally used for designs of 50 thousand gates or more

- Schematic Capture, an older method, suitable only for sub-50k gate designs and is generally less often used today

The two dominant HDLs are Verilog and VHDL. Verilog and VHDL are languages much like programming languages, such as C or C++, but they have been designed specifically for describing hardware behavior. HDLs allow designers to describe the function of their designs at a high level, often independent of the eventual implementation in silicon, much as a programmer can describe a function in the C language without knowing the specific compiler that will create the executable object code.

With schematic capture tools, graphical representations of the logic functions are placed on a computer screen and are manually connected by the designer. Schematic capture requires the designer to enter a much lower-level description of the design, implemented directly in the logic circuits available from the FPGA technology.

5.5.2Design Analysis and Simulation

After entering a design in an HDL, the designer begins the process of analyzing what was entered to determine if it correctly implements the intended function. The traditional method is through simulation, which evaluates how a design behaves. Simulation is a mature, well-understood process, and there are many simulators available that accept HDLs written in VDHL, Verilog, or increasingly, both languages, such as Verilog-XL™ and Leapfrog™ from Cadence; VSS™ from Synopses and MTI™ from Model Technology, Inc. Fig. 5.3 represent the traditional simulation process.

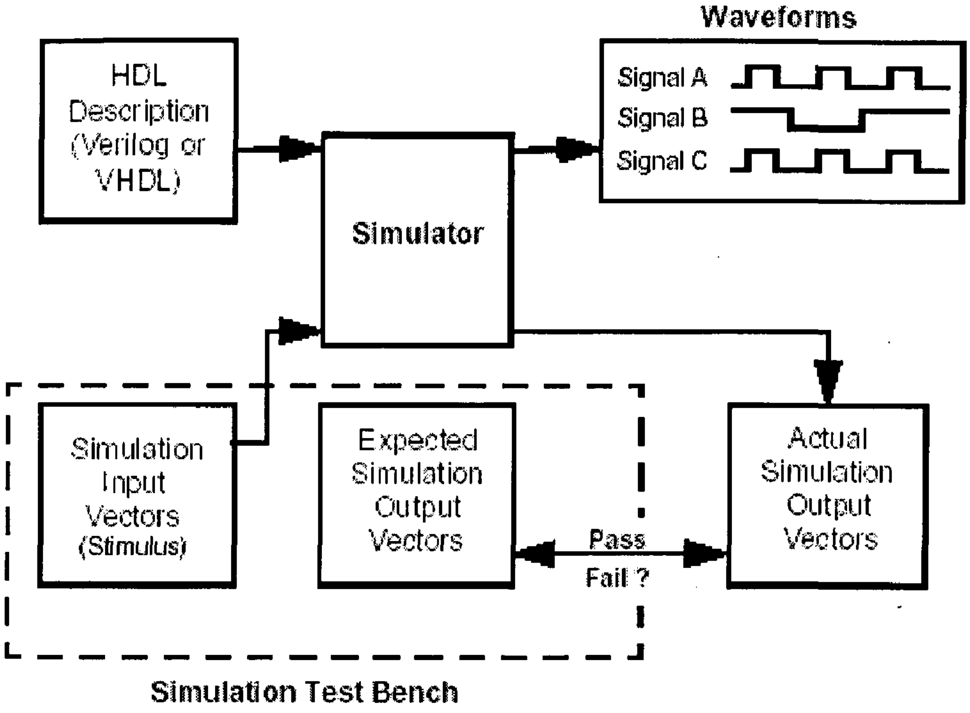

Waveforms
HDL Description (Verilog or VHDL)
Simulator
Signal A
Signal B
Signal C
Simulation Input Vectors (Stimulus)
Expected Simulation Output Vectors
Pass
Fail ?
Actual Simulation Output Vectors
Simulation Test Bench

Fig. 5.3 Traditional Simulation process (Source: www.xilinx.com, [24])

As shown in Fig.5.3, the VHDL or Verilog, which describes the design function, is read into the simulator tool along with a set of input vectors created by the designer. The simulator generates output vectors that are captured and evaluated against a set of expected values. If the output values match the expected values, then the simulation passes; if the output values differ, then the simulation is said to fail and the design needs to be corrected.

5.5.3 Logic Synthesis

Logic synthesis is the basic step that transforms the HDL representation of a design into technology-specific logic circuits. FPGA vendor provides the logic circuits in a form called a "synthesis library". As the synthesis tool breaks down high-level HDL statements into more primitive functions, it searches this library to find a match between the functions required and those provided in the library. When a match is found, the synthesis tool copies the function into the design (instantiates the circuit) and gives it a unique name (cell instance name). This process continues until all statements are broken down and mapped (synthesized) to logic circuits. There are potentially hundreds, or even thousands, of different combinations of logic circuits that can implement the same logical function. The combination chosen by a synthesis tool is determined by the synthesis constraints provided by the designer. These constraints define the design's performance, power, and area targets. A design driven primarily by performance criteria may use larger, faster circuits than one driven to minimize area or power consumption. The logic synthesis process is given in Fig. 5.4.

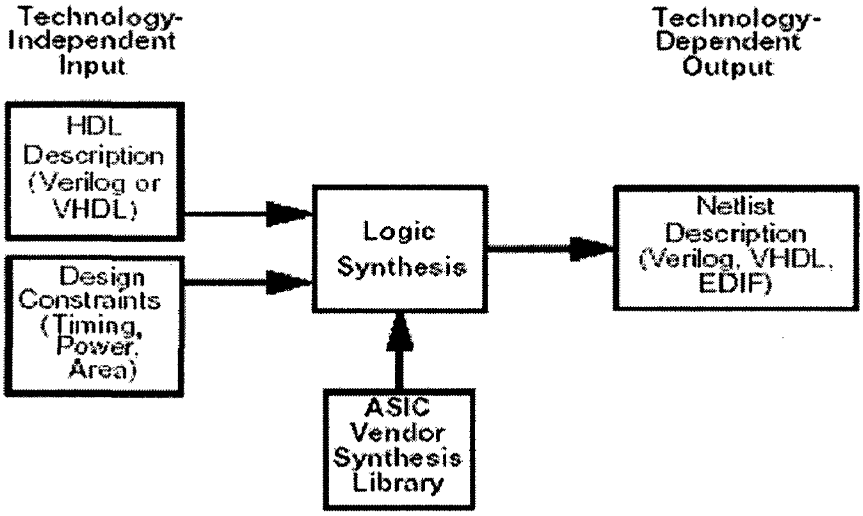

Technology-Independent Input
Technology-Dependent Output
HDL Description (Verilog or VHDL)
Design Constraints (Timing, Power, Area)
Logic Synthesis
Netlist Description (Verilog, VHDL, EDIF)
ASIC Vendor Synthesis Library

Fig.5.4. Logic Synthesis Process (Source: www.xilinx.com, [24])

5.5.4 Floor Planning

Floor planning is the process of choosing the best grouping and connectivity of logic in a design. Floor planning can be done automatically by the tool, or manually by the designer to increase density, routability, or performance. Most often the smallest area design is also the highest performance design. This flies in the face of many design methodologies, where area and speed are considered to be things that should be traded off against each other. The reason this is so, because there are limited routing resources, and the more routing resources that are used, the slower the design will operate. Optimizing for minimum area allows the design to use.

- Fewer resources
- Compact placement of design
- Shorter interconnect distances
- Less routing resources
- Faster end-to-end signal paths
- And even faster and more consistent place and route times

5.5.5 Design Verification

The design verification performed at this point in the design process ensures, through automated checking, that the design (1) is functionally correct, and (2) meets physical constraints in terms of performance, testability, power, and technology-specific electrical checks.

5.5.5.1 Functional Verification

Designs, as we have seen, are functionally verified before synthesis using simulation. Now, after synthesis, the design is re-simulated to ensure that its function has not been corrupted by the synthesis process. The traditional verification method is to re-simulate the gate-level version of the design. The process is straight-forward. The gate-level version of a design should produce the exact same functional results as the pre-synthesis version of the design, given the same set of stimulus (input vectors). Fig. 5.5 describes the functional verification process.

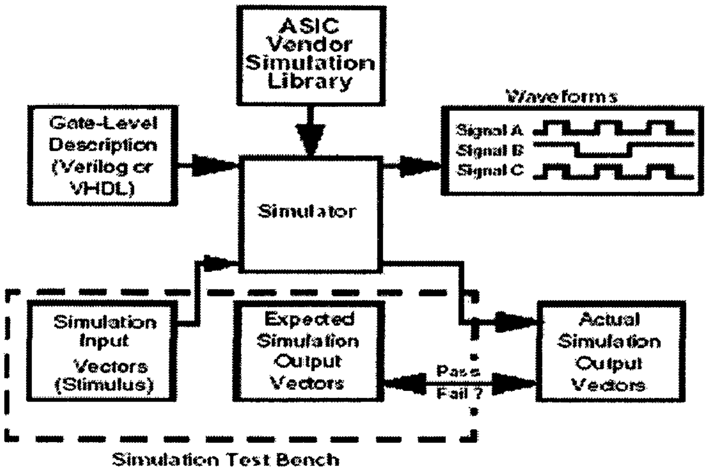

ASIC Vendor Simulation Library
Gate-Level Description (Verilog or VHDL)
Simulator
Waveforms
Signal A
Signal B
Signal C
Simulation Input Vectors (Stimulus)
Expected Simulation Output Vectors
Pass Fail ?
Actual Simulation Output Vectors
Simulation Test Bench

Fig. 5.5 Gate Level Verification, (Source: www.xilinx.com, [24])

5.5.5.2 Formal Verification

Formal verification achieves the same purpose as gate-level simulation, which is to guarantee that the function of the design was not altered or corrupted by the synthesis process. The method, however, is very different. A formal verification tool breaks a design down into a set of Boolean or logical expressions. This process is repeated on a second version of a design, and the logical expressions are compared for equivalence. The comparison of the two designs is exhaustive and not driven by evaluating different design states created by input vectors. There are no input or output vectors required. In addition, formal verification is also extremely useful for comparing two technology-dependent versions of a design for equivalence.

5.5.5.3 Timing Verification

The purpose of timing verification is to determine if a design, once mapped to a specific technology library of circuits, meets the specified performance target. Once again, the traditional method is based on gate level simulation, and run-time and design coverage issues make this method impractical for large designs

5.5.6 Implementation

In the design implementation stage, the net-list produced by the design entry program is converted into the bit stream file, which configures the FPGA. The first step maps the design onto the FPGA resources, the second step places or assigns logic blocks created in the mapping process in specific locations in the FPGA and lastly followed by routing the interconnect paths between the logic blocks. The output is a Logic Cell Array file (LCA) for a particular FPGA. This LCA file is then converted into a bit stream file for configuring the FPGA.

5.5.7 FPGA Configuration

After the design has been completely routed, the Synthesizer configures the device so that it can execute the desired function. Using the fully routed NCD file as input, it produces a configuration bit stream, a binary file with a .bit extension. The bit file contains all of the configuration information from the NCD file defining the internal logic and interconnections of the FPGA, plus device specific information from the other files associated with the target device. The binary data in the BIT file can then be downloaded into the FPGA's memory cells, or it can be used to create a PROM file. Hence, the configuration can be described as, a process in which the circuit design (bit stream file) is downloaded into the FPGA.

5.6 Summary

Architecture of FPGA's was described with the advantage and limitations of FPGA are presented in brief. It was then followed by the full FPGA's design process from design entry to implementation onto the chip for carrying out the desired task.

Chapter 6

DISCHARGE CALCULATION

6.1 Test Data

For the development of methodology of discharge calculation on computer and the comparison of the results with other (conventional) methods, data from the test run was used. The test was aimed at the measurement of discharge in open-channel using propeller current meters. The channel was rectangular in shape and dimension of channel were 19.21m total width and total depth of 2.4 m from the water surface.

Measurement of the discharge for a turbine requires that a location in the open-channel be chosen as the measurement plane, and a number of sampling sections be established across it. It was also found out that the flow in channel was non-uniform along the width and depth so, in order to collect the point velocities from the current meters, the channel was divided into horizontal and vertical planes.

Hence, a matrix of measuring points was formed, which is shown in Fig. 6.1. An arrangement of 44 points in rectangular section was proposed for the acquisition of velocities from the open channel.

The number of transects required to sample in the vertical is achieved by placing the transducer array attached to a movable frame, which was moved horizontally downwards to required elevations. It is assumed that the discharge during the sampling intervals was constant.

6.2 Matrix of Measuring Points

Eleven current meters were located at the depths of 0.48m, 0.96m, 1.44m and 1.92m depth from the water surface and laterally they were placed at distances of 1.0m, 2.7m, 4.4m, 6.1m, 7.8m, 9.5m, 11.2m, 13.11m, 14.81m, 16.51m, and 18.21m from the left side of channel. As described in section, (6.1) a matrix of 11X4 = 44 current meters locations was formed that gives the flow velocities (point velocities) at these locations. At every depth, the revolutions made by the eleven current meters, over the 120s interval were counted electronically. Five sets of such readings were taken at four minute intervals at one depth and the average flow velocity of water was worked out at the eleven current meters locations as shown in the table [1].

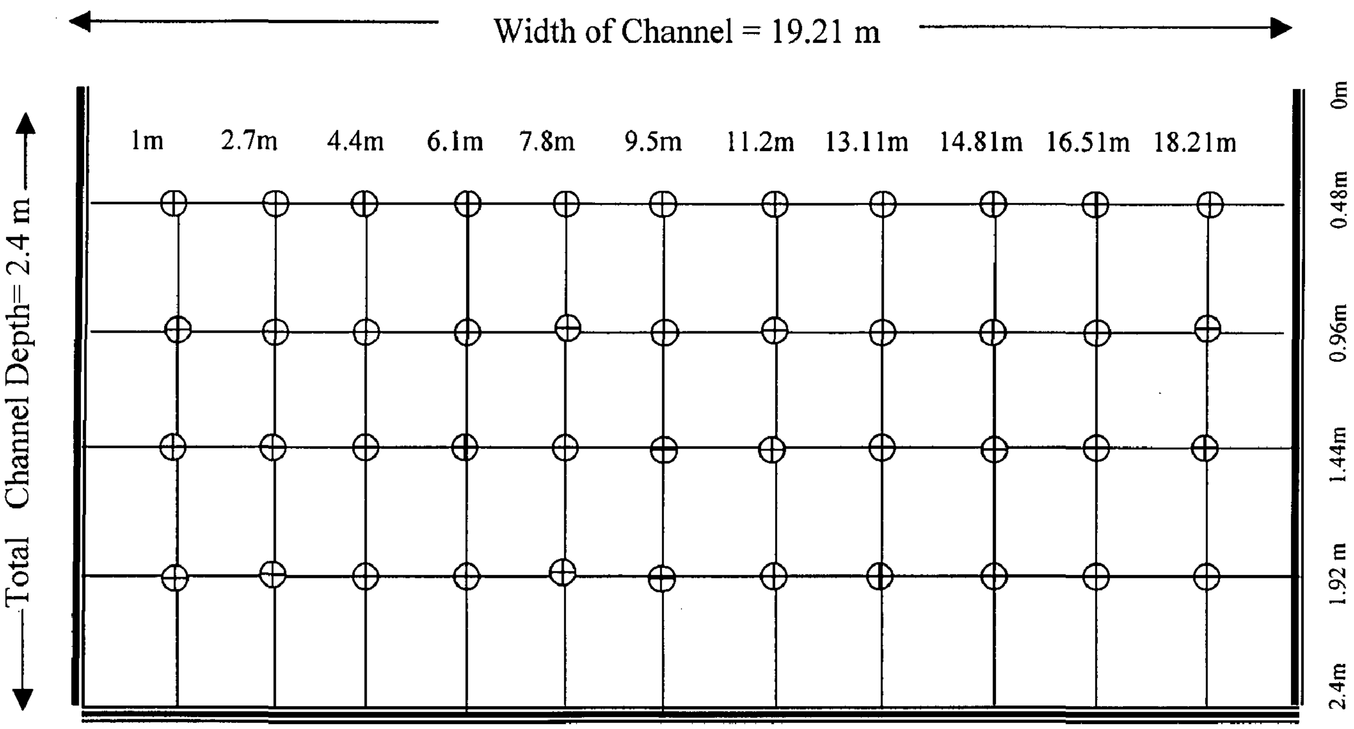


Fig. 6.1 Matrix (11 X 4) of Measuring Points formed in the Open-Channel.

First and last current meters were placed at a distance of 1.0m from the two ends of channel. Rest all current meters were places at a separation of 1.70m, except the separation was 1.910m between 8th and 9th current meters. The average point velocity measured by the current meters is tabulated in Table 6.1.

18.21 m	0.34 m/s	0.36 m/s	0.340 m/s	0.2931 m/s
16.51 m	0.640 m/s	0.7311 m/s	0.7402 m/s	0.6023 m/s
14.81 m	0.8136 m/s	0.929 m/s	0.940 m/s	0.8187 m/s
13.11 m	0.9399 m/s	1.0754 m/s	1.1636 m/s	1.1014 m/s
11.20 m	1.1282 m/s	1.2484 m/s	1.3431 m/s	1.2123 m/s
9.50 m	1.2556 m/s	1.3281 m/s	1.3867 m/s	1.2524 m/s
7.80 m	1.40 m/s	1.39 m/s	1.36 m/s	1.18 m/s
6.1 m	1.45 m/s	1.4521 m/s	1.4104 m/s	1.201 m/s
4.4 m	1.531 m/s	1.51 m/s	1.46 m/s	1.35 m/s
2.7 m	1.543 m/s	1.52 m/s	1.46 m/s	1.40 m/s
1.0 m	0.983 m/s	1.0751 m/s	1.1593 m/s	1.0771 1 m/s
Width / Depth	0.48 m	0.96 m	1.44 m	1.92 m

Table 6.1 Point Velocities Measured By Current Meters in the Open-Channel

6.3 Scheme for Computation of Open-Channel Discharge

Fig. 6.2, describe the various steps involved for the discharge computation in an open-channel using the propeller current meters

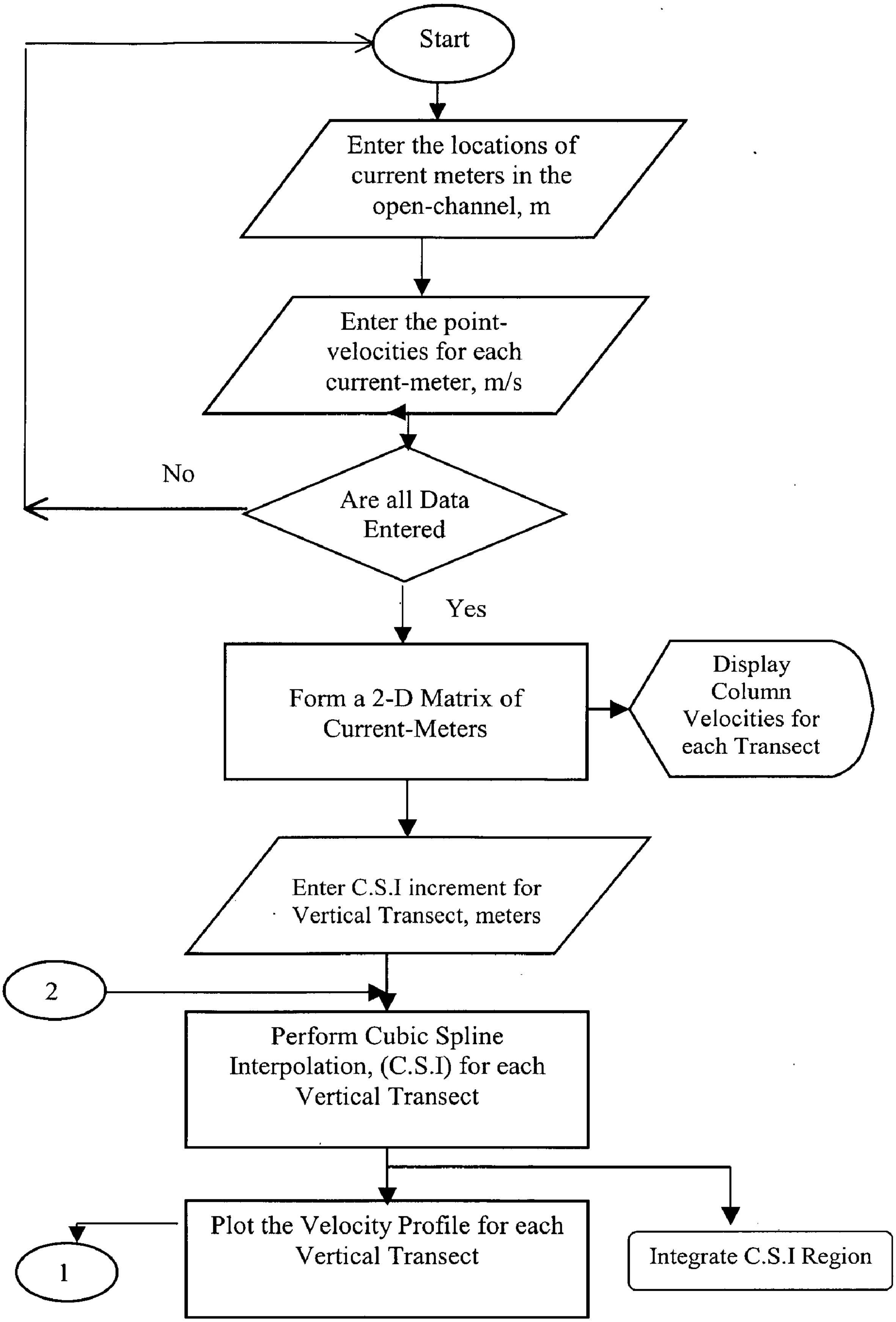

Start
Enter the locations of current meters in the open-channel, m
Enter the point-velocities for each current-meter, m/s
No
Are all Data Entered
Yes
Form a 2-D Matrix of Current-Meters
Display Column Velocities for each Transect
Enter C.S.I increment for Vertical Transect, meters
2
Perform Cubic Spline Interpolation, (C.S.I) for each Vertical Transect
Plot the Velocity Profile for each Vertical Transect
1
Integrate C.S.I Region

Fig. 6.2 (Contd.)

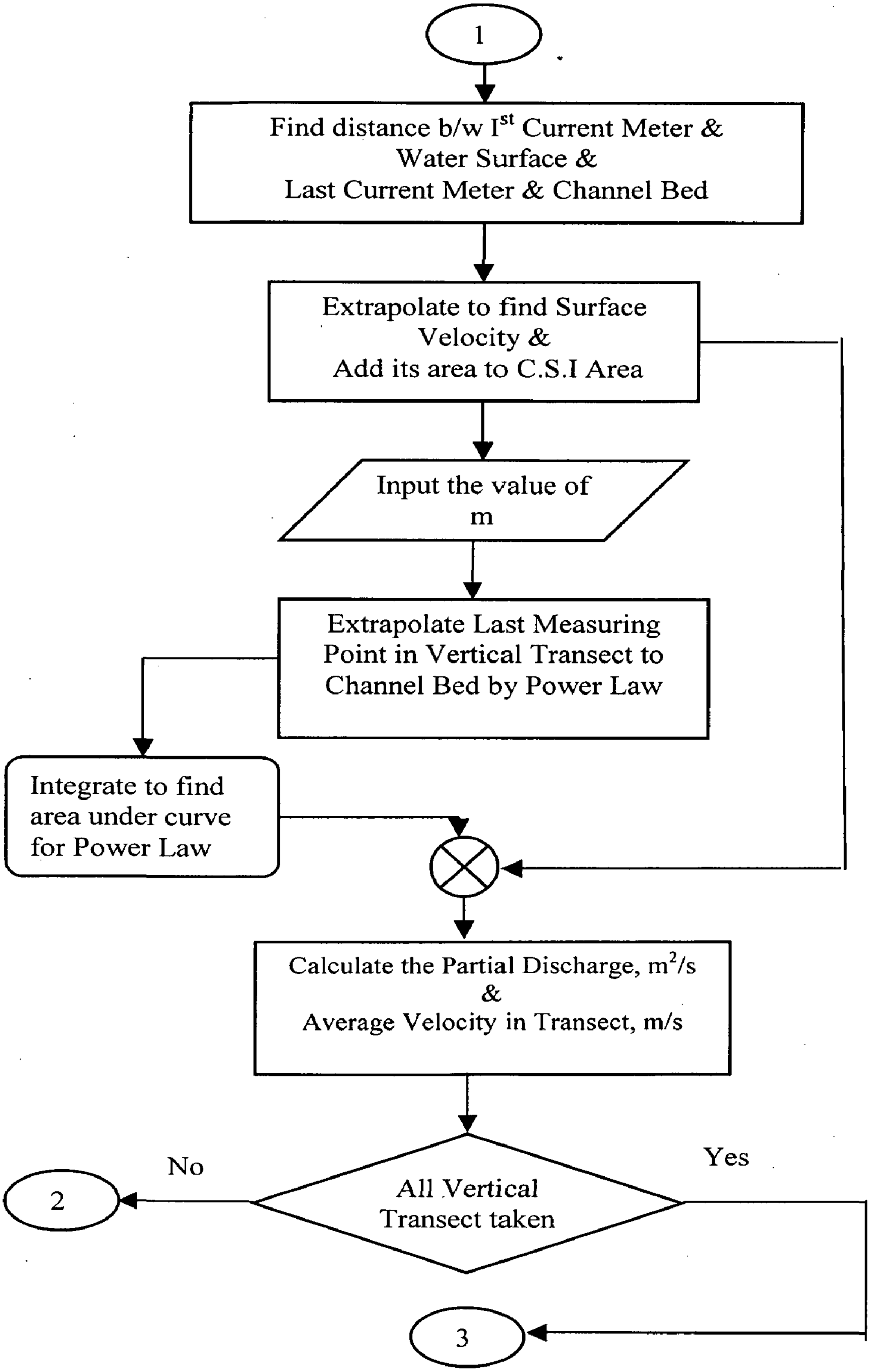

1
Find distance b/w Ist Current Meter & Water Surface & Last Current Meter & Channel Bed
Extrapolate to find Surface Velocity & Add its area to C.S.I Area
Input the value of m
Extrapolate Last Measuring Point in Vertical Transect to Channel Bed by Power Law
Integrate to find area under curve for Power Law
Calculate the Partial Discharge, m^2/s & Average Velocity in Transect, m/s
All Vertical Transect taken
No
Yes
2
3

Fig. 6.2 (Contd.)

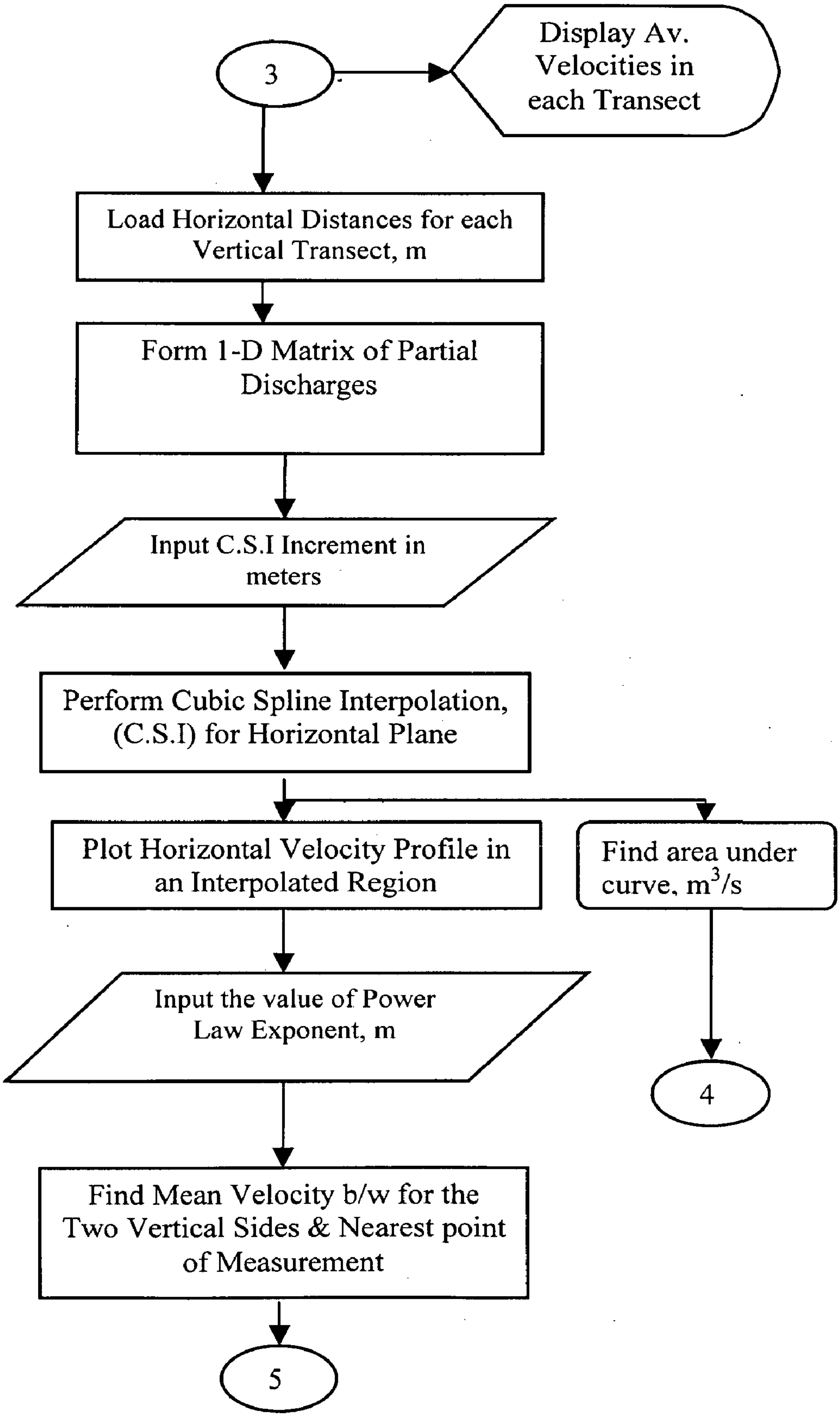

3
Display Av. Velocities in each Transect
Load Horizontal Distances for each Vertical Transect, m
Form 1-D Matrix of Partial Discharges
Input C.S.I Increment in meters
Perform Cubic Spline Interpolation, (C.S.I) for Horizontal Plane
Plot Horizontal Velocity Profile in an Interpolated Region
Find area under curve, m³/s
Input the value of Power Law Exponent, m
4
Find Mean Velocity b/w for the Two Vertical Sides & Nearest point of Measurement
5

Fig. 6.2 (Contd.)

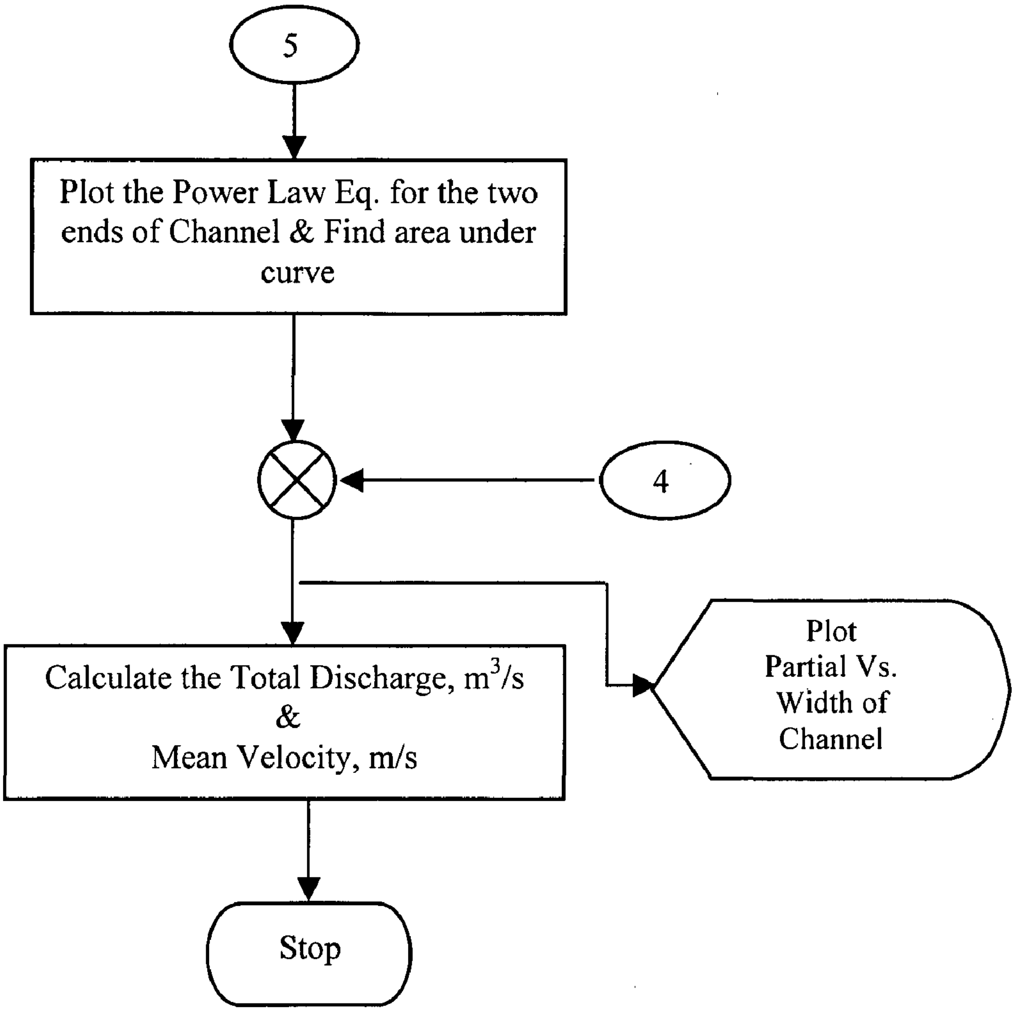

5
Plot the Power Law Eq. for the two ends of Channel & Find area under curve
4
Calculate the Total Discharge, m3/s & Mean Velocity, m/s
Plot Partial Vs. Width of Channel
Stop

Fig. 6.2 Flow Chart for Open-Channel Discharge Computation using Power Law Extrapolation Scheme

6.4 Discharge Computations

The open-channel discharge was calculated by the Graphical means, and by the Arithmetic method. Following sections describes the computation results by both the methods. First, the results from the Arithmetic methods are presented, followed by results from the Graphical means for discharge measurement.

6.4.1 Arithmetic Methods

The basic theory regarding this method is presented in section, (3.4.1). Here, the point velocity at 0d, 0.2d, 0.4d, 0.6d, 0.8d and near the channel bed was taken. Mean velocity was calculated as give in Eq. 2.4 Surface velocity was computed by extrapolating the data. The velocity at the channel bed was taken to be zero, because of the effect of bed resistance, to the flow of water. Data for the point velocities are presented in Table 6.1.

6.4.1.1 Mean-Section Method

Here, partial discharges were calculated for each vertical cross-section of channel. As described in Eq. 3.4 partial discharge was computed individually, and summation of partial discharges for 1 to 11 verticals gives the total discharge. Table 6.2 shows the computed partial discharges in the case of mean section method

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Partial Discharge cu.m/s	1.18	4.76	5.508	5.56	5.14	4.8	4.69	4.81	3.55	2.833	1.85

Total Discharge, Q = $\sum$ Qi = 44.73 m^3/s

Table 6.2 Partial Discharge Computed by Mean-Section Method

6.4.1.2 Mid-Section Method

As in case of mean-section method, partial discharges for mid-section method were also calculated for each vertical cross-section of channel. The partial discharge was computed individually by using the Eq 3.5, and summation of partial discharges for 1 to 11 verticals gives the total discharge. Table 6.3 shows the computed partial discharges in the case of mid-section method.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Partial Discharge cu.m/s	3.2	5.53	5.308	5.237	5.075	4.931	4.912	4.183	3.178	2.48	0.962

Total Discharge, $Q= \sum Qi=45.0089\ m^3/s$

Table 6.3 Partial Discharge Computed by Mid-Section Method

The total discharge as computed by mean section method was 44.73m^3/s while as computed by mid-section method was 45.0089m^3/s.

6.4.2 Discharge Measurement Using the Graphical Method

Discharge computation by graphical means was described in section, (3.3). First, the results obtained by the curve fitting method is presented (6.4.2.1), followed by the Cubic Spline Interpolating Scheme, (6.4.2.2), and lastly followed by results obtained by improved technique using power law extrapolation (6.4.2.3).

6.4.2.1 Discharge Computation by Curve Fitting Method

In the curve fitting method for the discharge computation, i.e. along each vertical transect velocity profile computation was done. It was assumed that, velocity at the channel beds were zero. The partial discharge was computed for each vertical transect along the width of channel. It was later integrated by Adaptive Lob Otto Quadrature to get the mean velocity in vertical cross-section; with limits being channel bed elevation as upper limit and channel surface being the lower limit. Later the partial discharges were curve-fitted by some polynomial, along the width of channel. It was interpolated and integrated by same techniques, to get the total discharge in the channel. Mean-velocity in the channel was found out by dividing the total discharge to the area of cross-section of channel. The total discharge computed by the curve-fitting method comes out to be 44.19087 m^3/s. The mean-velocity in the channel, as computed by this method was 0.9585 m/s. For each vertical transects the surface velocity was simply computed by extrapolating the curve, i.e. to the zero elevation from the water surface. More flow was found towards the left side of channel. It was due the presence of a small bend upstream in the channel. Figures 6.3 to 6.12 illustrate the results obtained by the above technique.

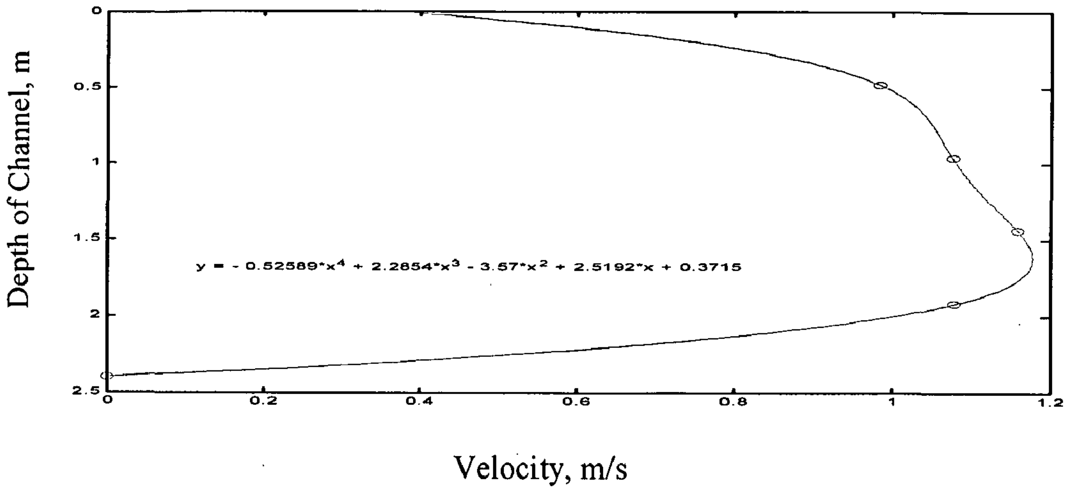


Fig.6.3 Velocity Profile along I -Vertical

The Partial Discharge = **2.2774** m^2/s

The Mean velocity at I -Vertical = **0.948166** m/s

Surface Velocity = **0.3715** m/s

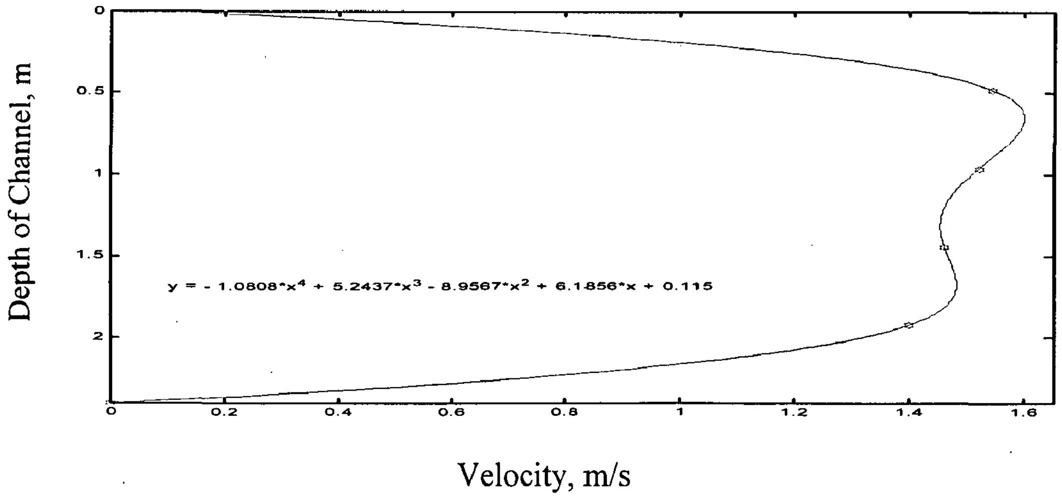


Fig. 6.4 Velocity Profile along II -Vertical

The Partial Discharge = **3.0994** m^2/s

The Mean Velocity at II -Vertical = **1.291141** m/s

Surface Velocity = **0.115** m/s

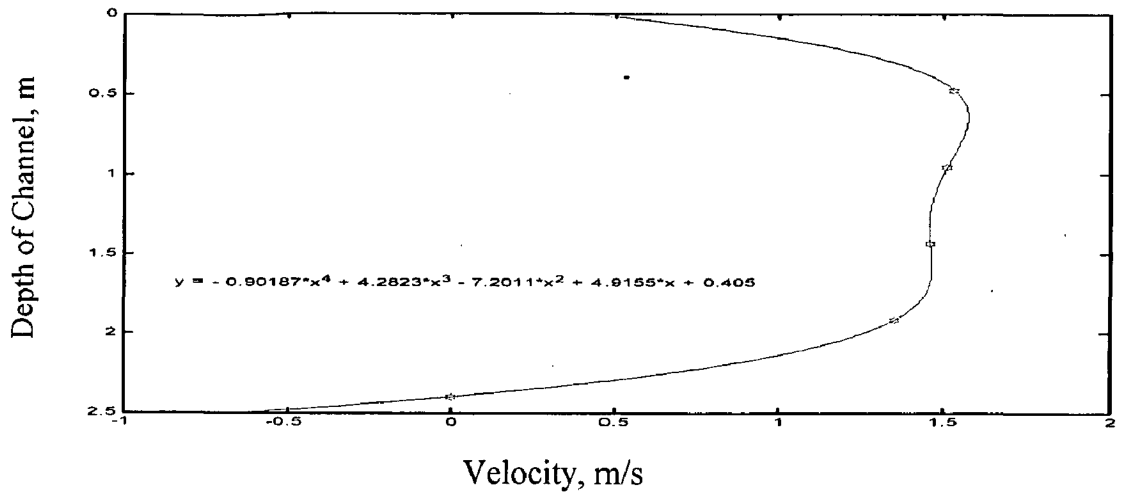


Fig.6.5 Velocity Profile along III -Vertical

The Partial Discharge = **3.10226** m^2/s

The Mean Velocity at III -Vertical = **1.2914** m/s

Surface Velocity = **0.405** m/s

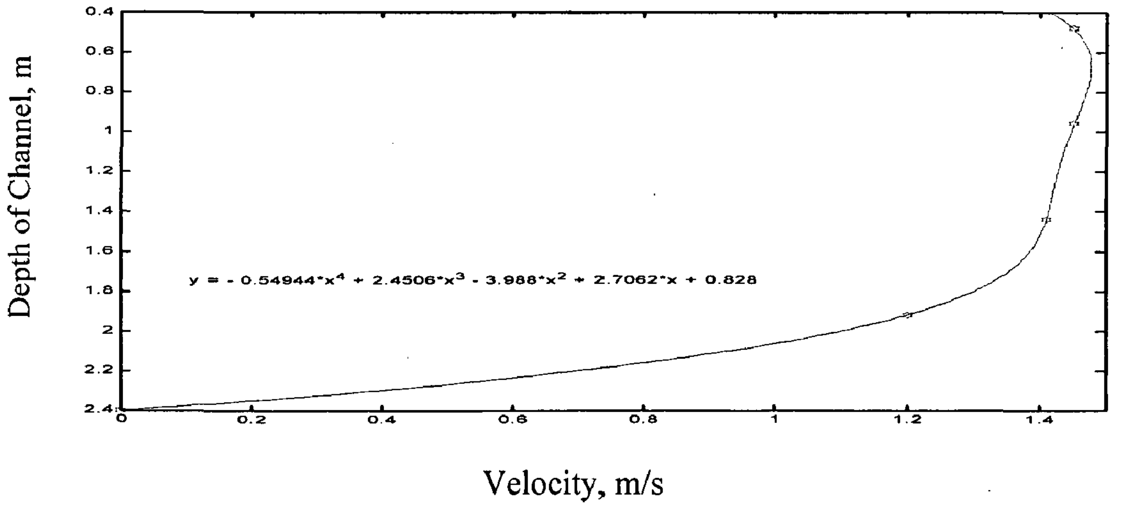


Fig.6.6 Velocity Profile along IV - Vertical

The Partial Discharge = **2.89806** m^2/s

The Mean Velocity at IV -Vertical = **1.20752** m/s

Surface Velocity = **0.828** m/s

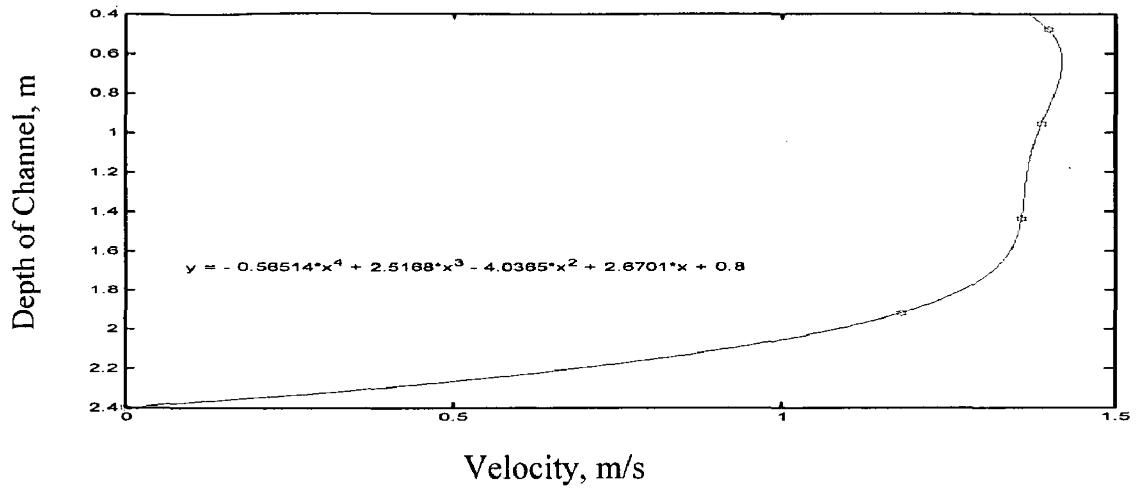


Fig.6.7 Velocity Profile along V- Vertical

The Partial Discharge = **2.8850** m^2/s

The Mean Velocity at V-Vertical = **1.2020** m/s

Surface Velocity = **0.8** m/s

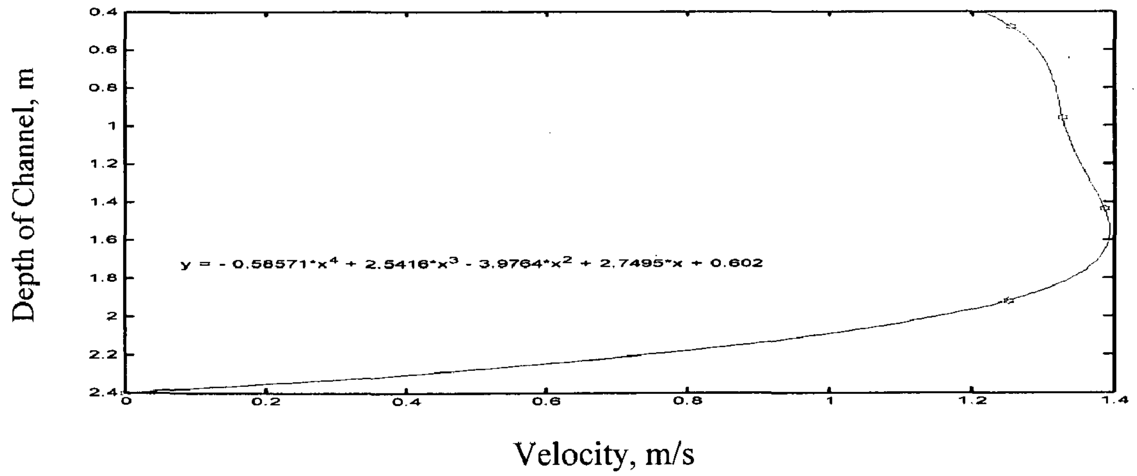


Fig.6.8 Velocity Profile along VI - Vertical

The Partial Discharge = **2.7936** m^2/s

The Mean Velocity at VI -Vertical = **1.164** m/s

Surface Velocity = **0.602** m/s

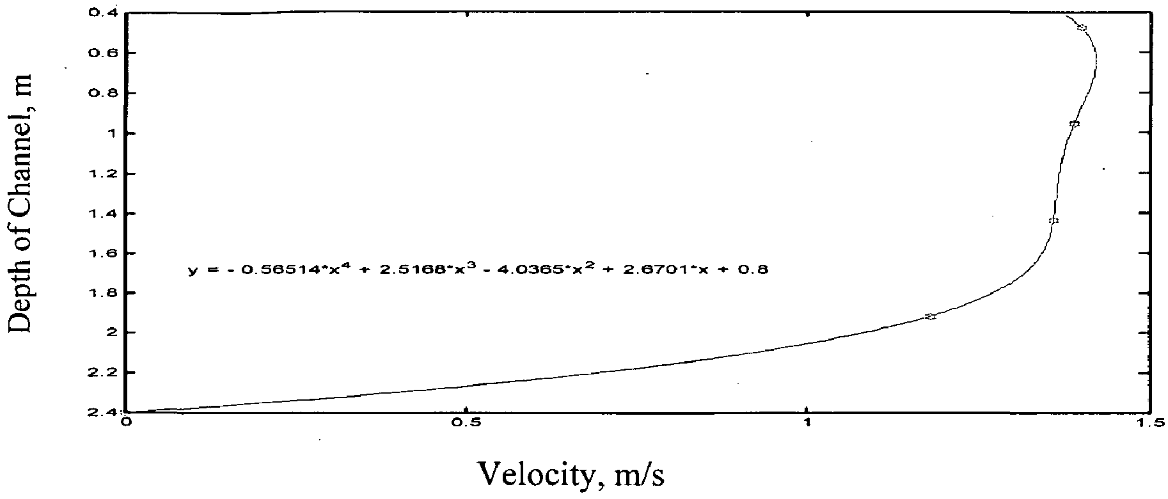


Fig.6.7 Velocity Profile along V- Vertical

The Partial Discharge = **2.8850** m^2/s

The Mean Velocity at V-Vertical = **1.2020** m/s

Surface Velocity = **0.8** m/s

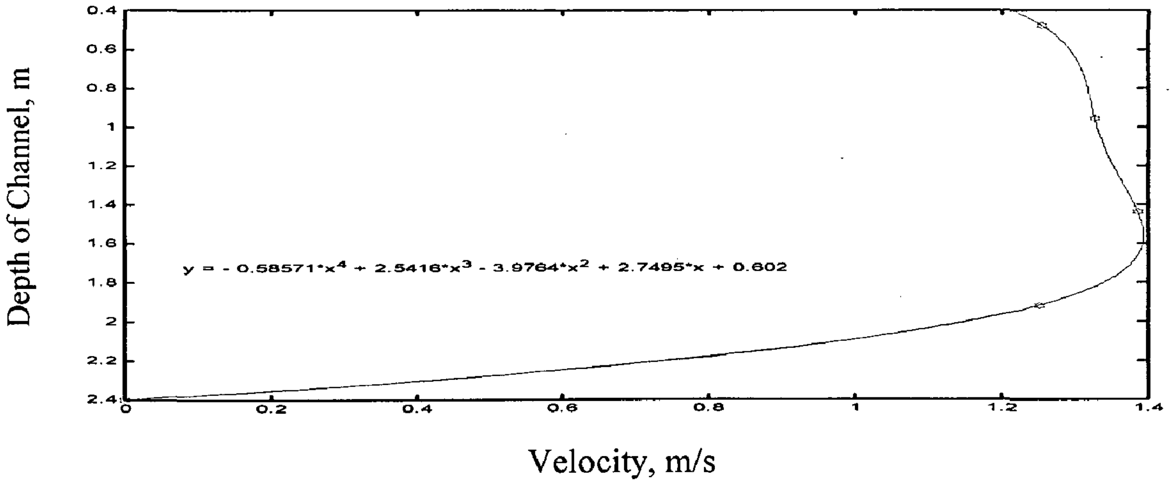


Fig.6.8 Velocity Profile along VI - Vertical

The Partial Discharge = **2.7936** m^2/s

The Mean Velocity at VI -Vertical = **1.164** m/s

Surface Velocity = **0.602** m/s

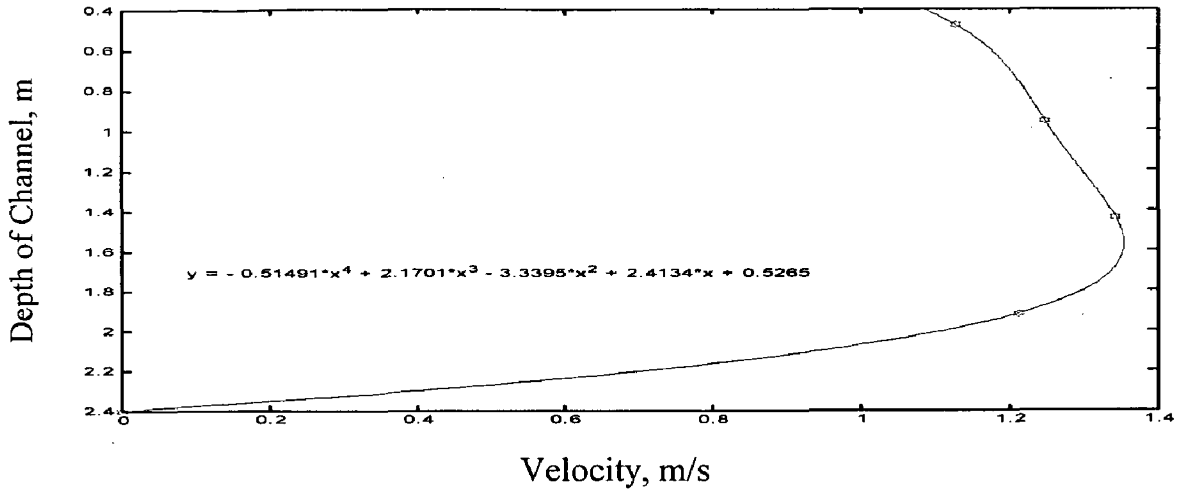


Fig.6.9 Velocity Profile along VII - Vertical

The Partial Discharge = **2.6254** m^2/s

The Mean Velocity at VII -Vertical = **1.0939** m/s

Surface Velocity = **0.5265** m/s

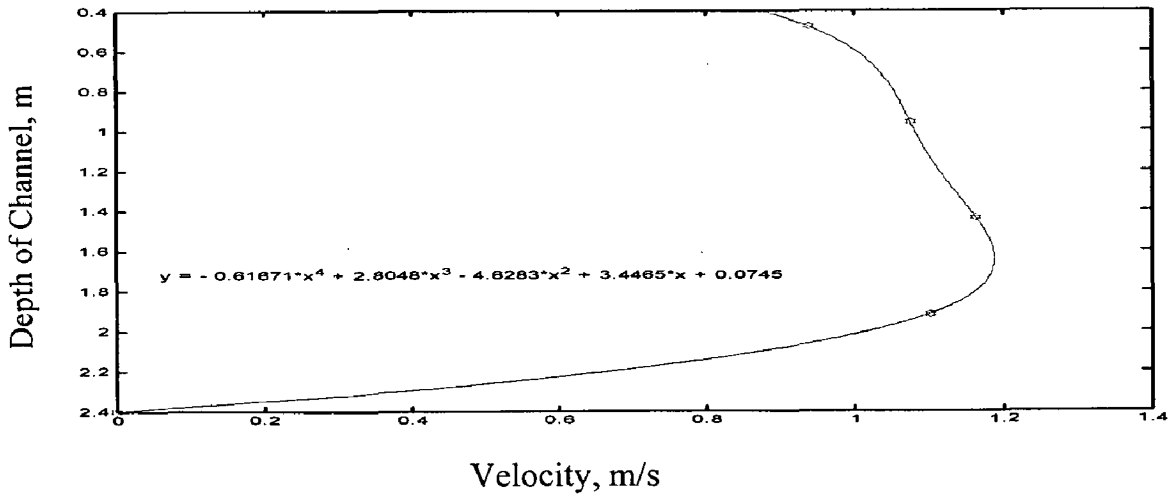


Fig.6.10 Velocity Profile along VIII - Vertical

The Partial Discharge = **2.2204** m^2/s

The Mean Velocity at VIII -Vertical = **0.925166** m/s

Surface Velocity = **0.0745** m/s

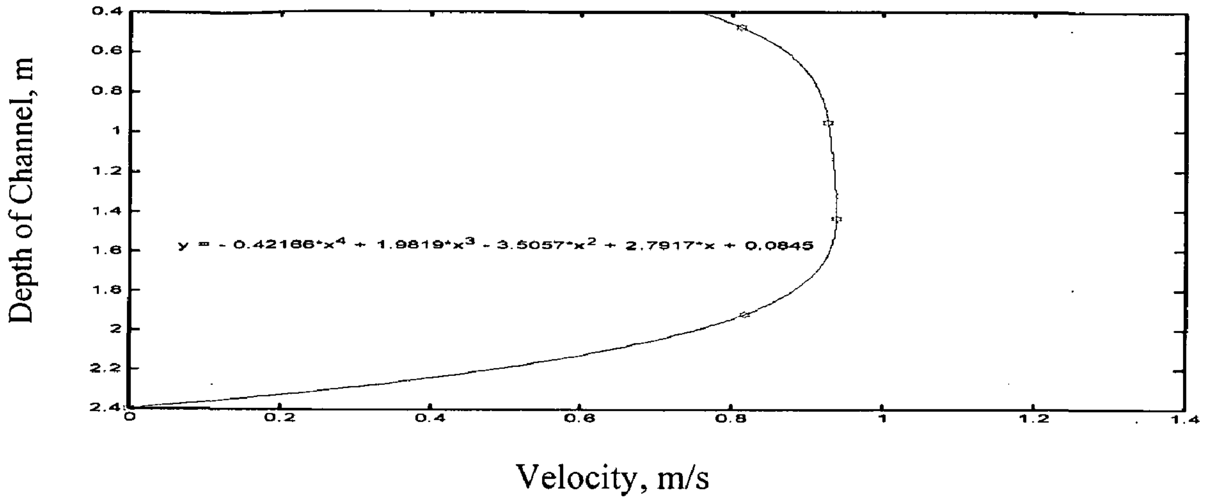


Fig.6.11 Velocity Profile along IX - Vertical

The Partial Discharge = **1.9014** m^2/s

The Mean Velocity IX -Vertical = **0.79225** m/s

Surface Velocity = **0.0845** m/s

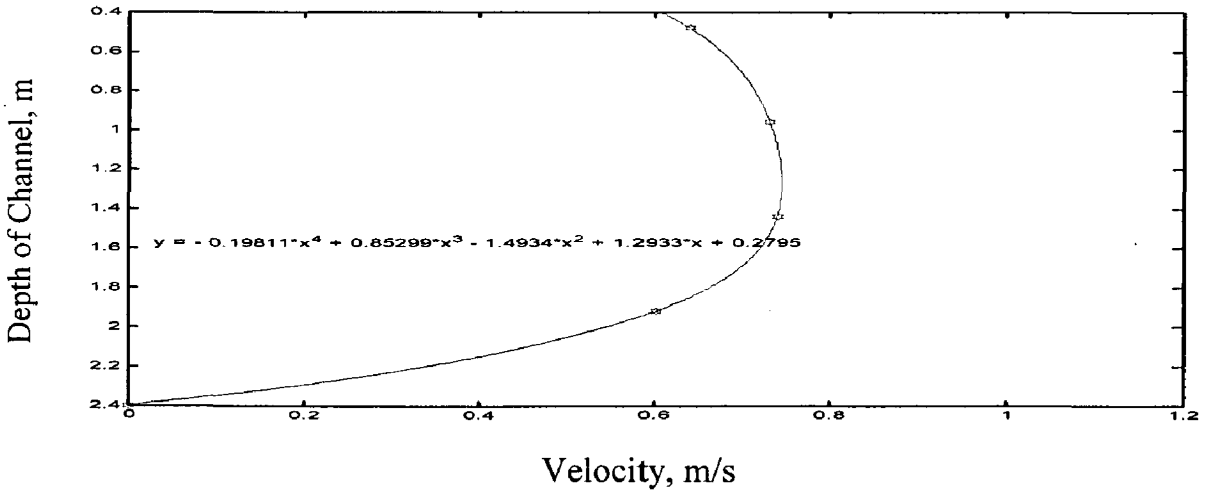


Fig.6.12 Velocity Profile along X- Vertical

The Partial Discharge = **1.434** m^2/s

The mean velocity at X-Vertical = **0.5975** m/s

Surface Velocity = **0.2795** m/s

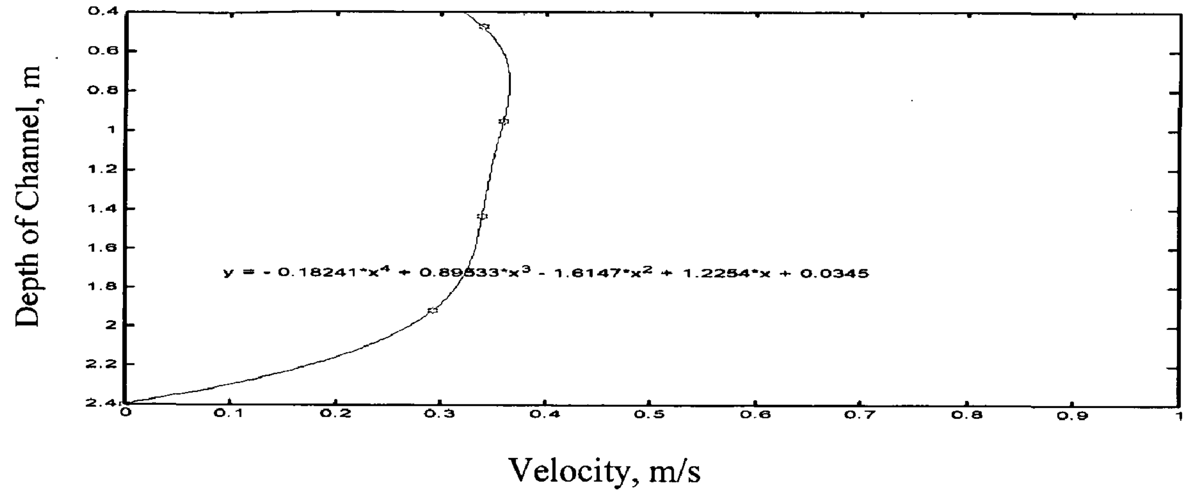


Fig.6.13 Velocity Profile along XI- Vertical

The Partial Discharge = **0.6927** m^2/s

The mean velocity at XI -Vertical = **0.28862575** m/s

Surface Velocity = **0.0345** m/s

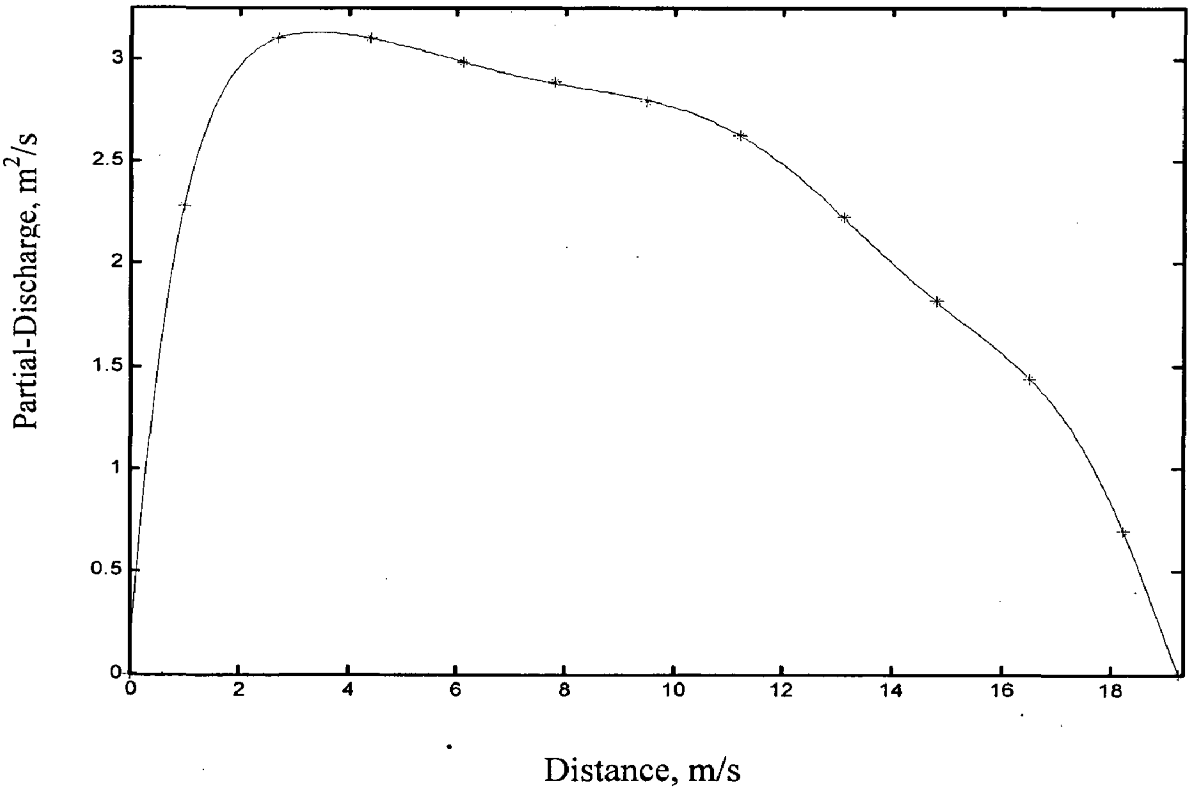


Fig. 6.14 Partial-Discharge vs. Distance

Polynomial Eq. = 0.45e-10*x. ^12+0.532e-8*x. ^11-0.2722e-066*x. ^10+0.8053e-06*x. ^9-0.008153*x. ^8+.00196*x. ^7-.017381*x. ^6+0.1067511*x. ^5-.045510*x. ^4+1.3351661*x. ^3-2.8462*x. ^2+4.13589*x-.016459e-10

The Total Discharge in an Open-Channel is **44.19087 m^3/s**
The Mean Velocity in an Open-Channel is **0.9585 m/s**

The computed discharge by the curve fitting method was found put to be 44.190 m^3/s, with average velocity of flow in the channel being 0.958 m/s. Velocities profiles for each vertical transect was calculated along with the functional polynomial, representing the function of velocity, y with respect to depth, x. The order of polynomial was one less than the number of points in the measuring section. Fig. 6.14 shows the variation of partial discharges w.r.t the distance. Area under this curve gives the total discharge in the channel.

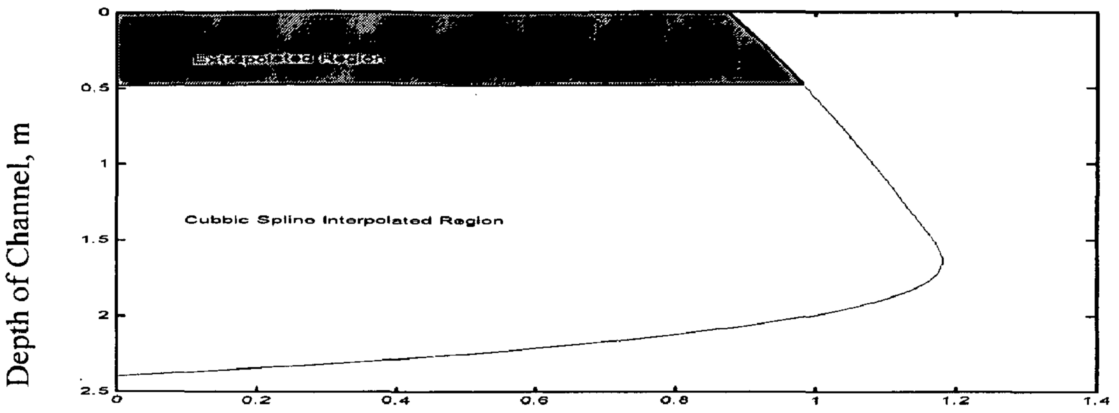
6.4.2.2 Cubic Spline Interpolation Scheme

Like the curve fitting method, velocity profiles were computed by the cubic spline interpolation at an increment of 1cm along the depth as well as the width of channel.

It was assumed that the velocities at the channel beds were zero. Partial discharge is computed for each vertical transect along the width of channel. Like the curve fitting method the mean-velocity in the channel was found out by dividing the total discharge to the area of cross-section.

Surface velocity extrapolation was done by cubic spline interpolation. Figures 6.15 to 6.26 show the results obtained by the above technique. Shaded region in the velocities profile curve shows the extrapolation region. The final value of computed discharge by this method comes out to be 45.6883 m^3/s with the mean velocity of flow being 0.990m/s.

Fig. 6.15 to 6.25 illustrate the velocity profiles in vertical transect of the channel. The Fig. 6.26 illustrate plot of partial discharge versus the distance from left side of channel.

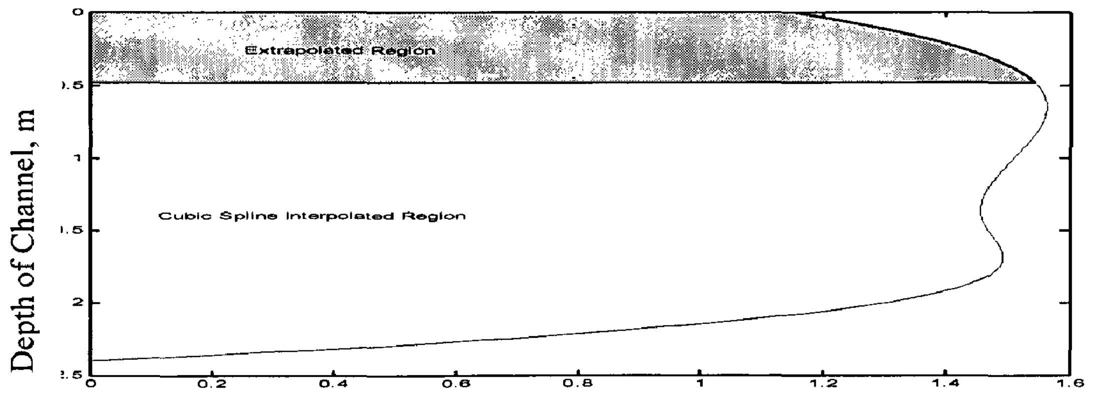


F Velocity, m/s

The Partial Discharge = **2.3522** m^2/s

Mean Velocity in at I -Vertical Transect = **0.98008** m/s

Surface Velocity = **0.874** m/s



Velocity, m/s

Fig. 6.16 Velocity Profile along II - Vertical

The Partial Discharge = **3.253** m^2/s

Mean Velocity in at II -Vertical Transect = **1.34804** m/s

Surface Velocity = **1.147** m/s

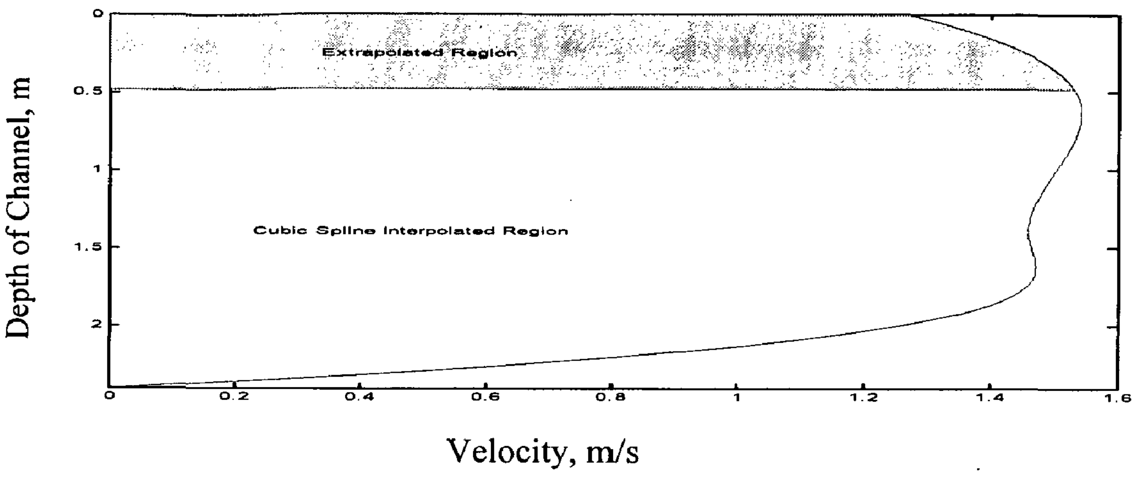


Fig. 6.17 Velocity Profile along III - Vertical

The Partial Discharge = **3.2305** m^2/s

Mean Velocity in at III -Vertical Transect = **1.34604** m/s

Surface Velocity = **1.26677** m/s

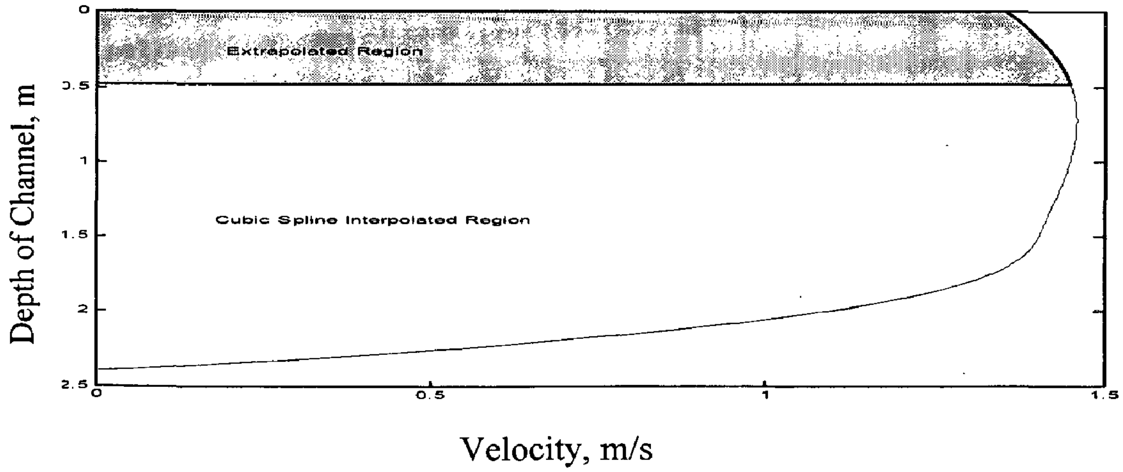


Fig. 6.18 Velocity Profile along IV - Vertical

The Partial Discharge = **3.0588** m^2/s

Mean Velocity in at IV - Vertical Transect = **1.4567** m/s

Surface Velocity = **1.353** m/s

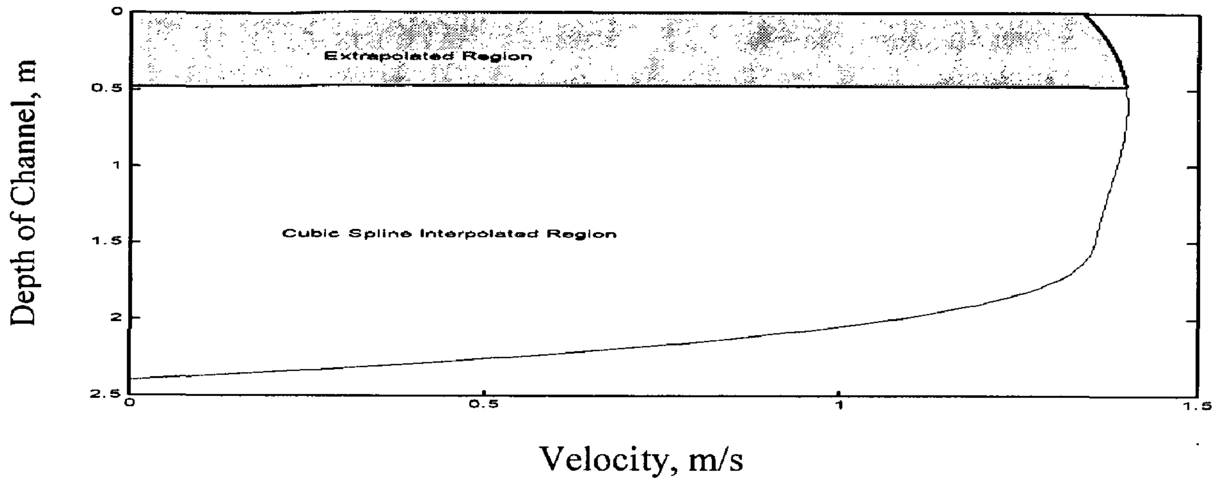


Fig. 6.19 Velocity Profile along V - Vertical

The Partial Discharge = **2.9654** m^2/s

Mean Velocity in at V - Vertical Transect = **1.23558** m/s

Surface Velocity = **1.345** m/s

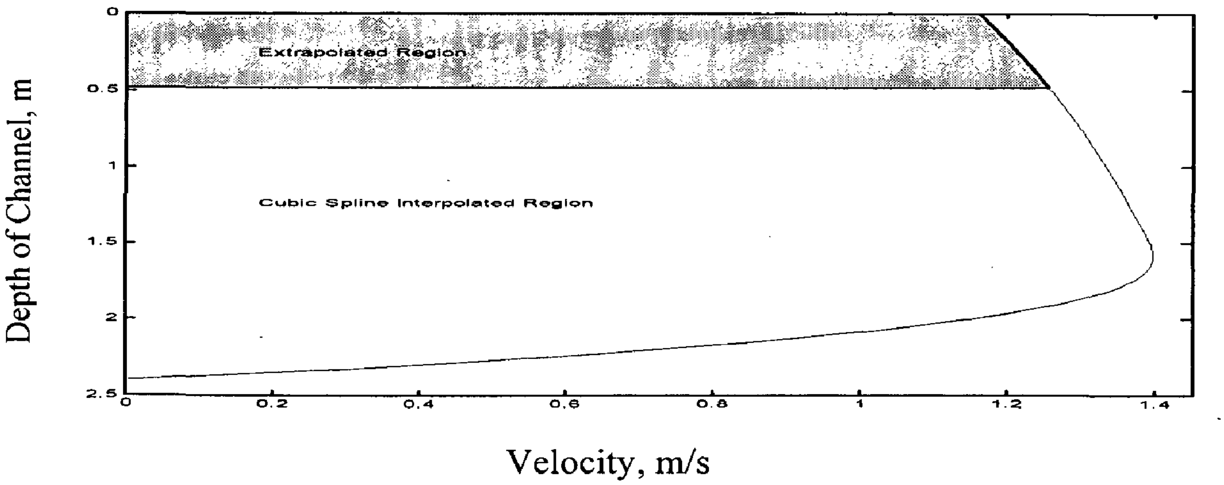


Fig. 6.20 Velocity Profile along VI - Vertical

The Partial Discharge = **2.8773** m^2/s

Mean Velocity in at VI - Vertical Transect = **1.1988** m/s

Surface Velocity = **1.16178** m/s

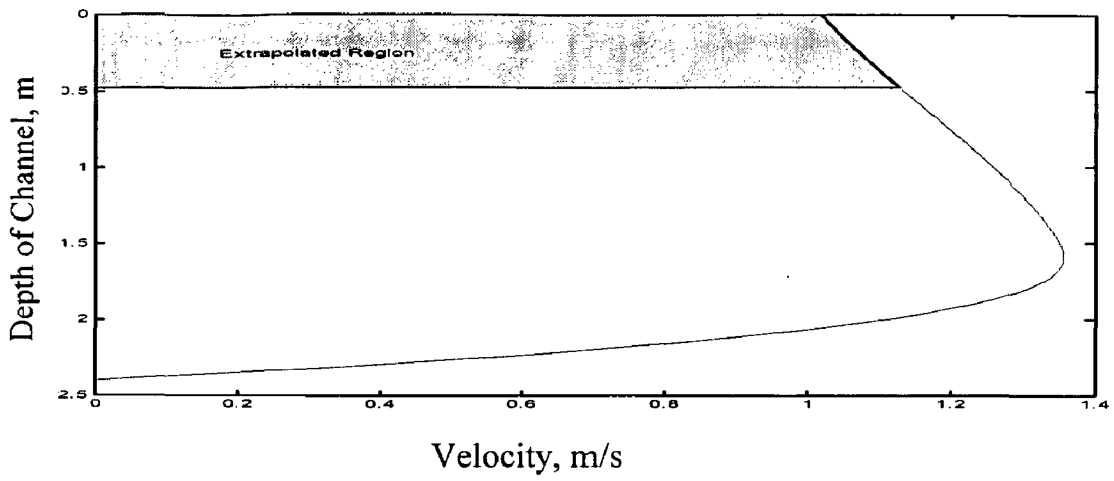


Fig. 6.21 Velocity Profile along VII - Vertical

The Partial Discharge = **2.6992** m^2/s

Mean Velocity in at VII - Vertical Transect = **1.124667** m/s

Surface Velocity = **1.0185** m/s

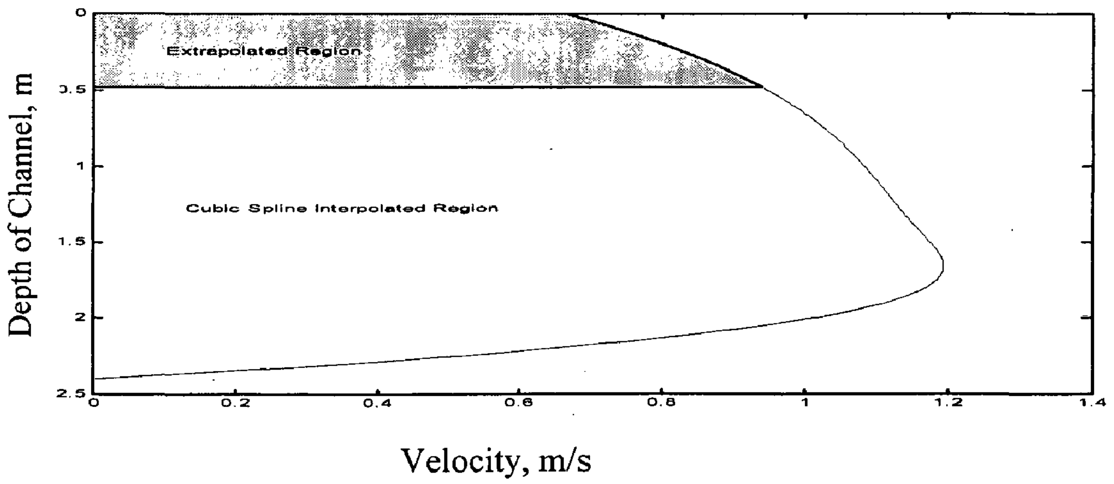


Fig. 6.22 Velocity Profile along VIII- Vertical

The Partial Discharge = **2.3082** m^2/s

Mean Velocity in at VIII-Vertical Transect = **0.96175** m/s

Surface Velocity = **0.6638** m/s

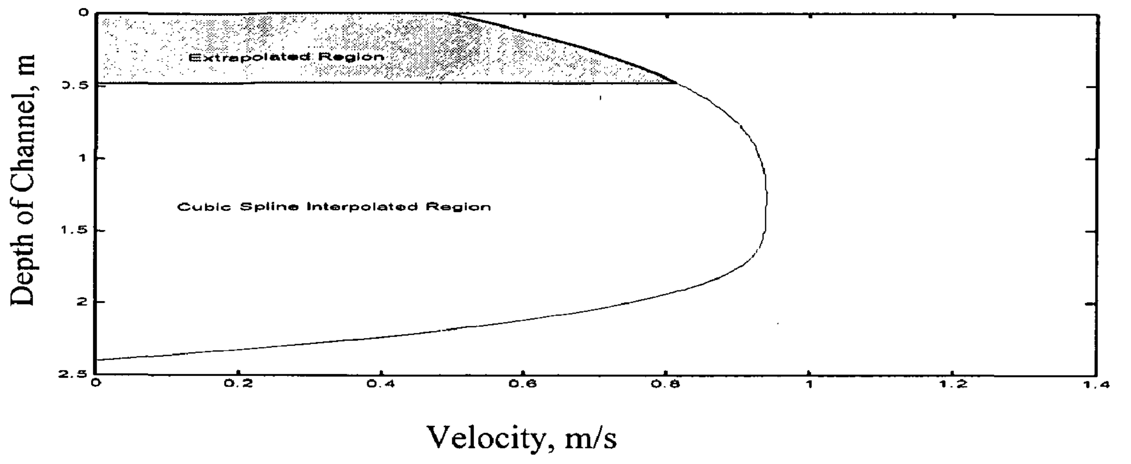


Fig. 6.23 Velocity Profile along IX - Vertical

The Partial Discharge = **1.8723** m^2/s

Mean Velocity in at IX - Vertical Transect = **0.780125** m/s

Surface Velocity = **0.4871** m/s

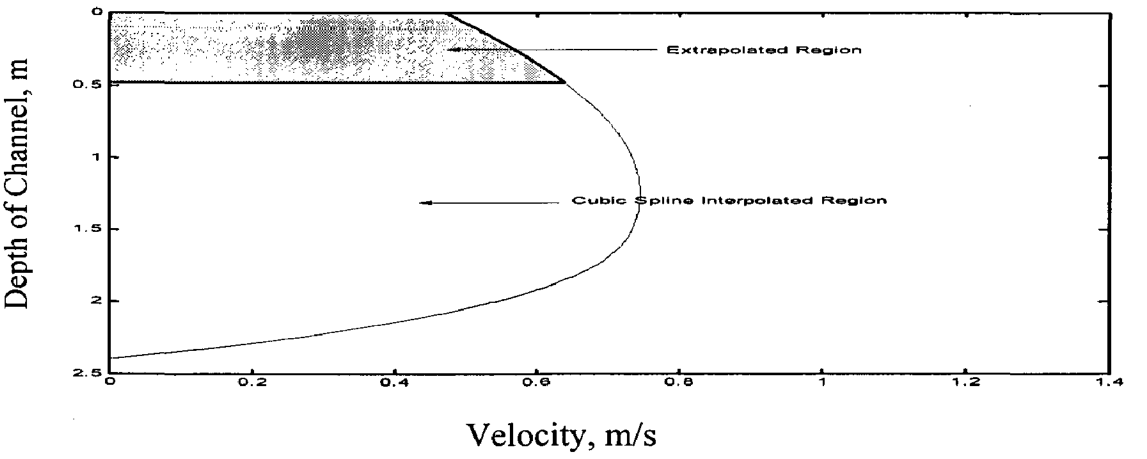


Fig. 6.24 Velocity Profile along X- Vertical

The Partial Discharge = **1.4619** m^2/s

Mean Velocity in at X-Vertical Transect = **0.6091** m/s

Surface Velocity = **0.48** m/s

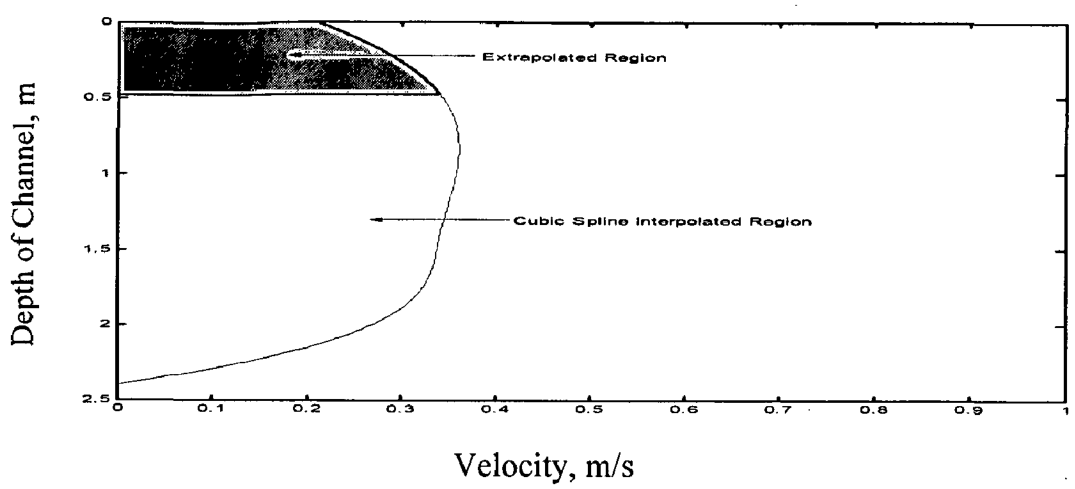


Fig. 6.25 Velocity Profile along XI- Vertical

The Partial Discharge = **0.7188** m^2/s

Mean Velocity in at XI-Vertical Transect: **0.2995** m/s

Surface Velocity = **0.20884** m/s

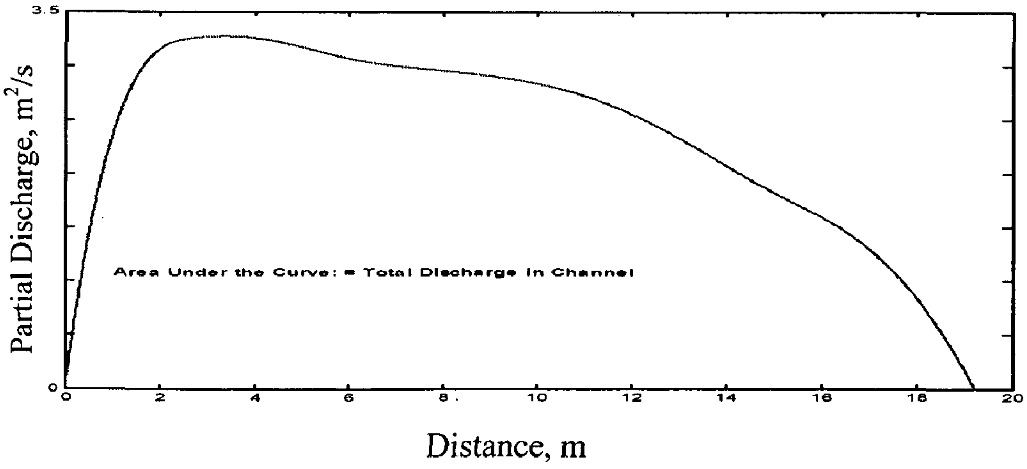


Fig. 6.26 Partial-Discharge vs. Distance

Total Discharge in an Open-Channel = **45.6883** m^3/s

Mean Velocity in the Channel = **0.99098** m/s

Fig. 6.26 illustrate the plot of partial discharge versus the distance from the left side of channel. Integrating the partial discharge w.r.t depth gives the total discharge in the channel. The value of discharge was found out to be 45.6883 m^3/s and the mean channel velocity as computed by this method comes out to be 0.99098 m/s.

6.4.2.3 Improved Scheme using Power Law Extrapolation

In order to calculate the partial discharge in the vertical cross-section, it was divided into three different elements:

In the first part, only the interpolation was done by Cubic Spline Technique. The interpolating region in each vertical depth for our case was from 0.48 m to the 1.92m from the water surface.

Then the extrapolation was carried out, by Cubic Spline to find the surface velocity and bounded area by the curve between 0 depths and 20% depth from the water surface.

Lastly, between the channel beds and last measuring point, power law extrapolation was carried to find the area, i.e. the partial discharge. The value of power law exponent was taken as 5; minimum permitted by the ISO-748 and IS-1192 standard.

Total partial discharge was calculated by adding the interpolated region's area, area of surface extrapolated region and area bounded by power law extrapolated region. All the other values for the calculation of open-channel discharge were taken to be constant.

So, the total partial discharges for each vertical transect was integrated, w.r.t the distance, to give the total discharge. At the two ends, the power law extrapolation is used for the computation of discharge. Here, also the value of m was taken as 5. Adding the two we get the total discharge in the channel, and mean velocity is determined by dividing the total discharge to area of cross-section. The total discharge as computed by this method comes out to be 48.3154 m^3/s with the mean-velocity of flow being 1.04 m/s. The discharge between the left wall and first measuring points was 2.02 m^3/s, and last measuring point and the opposite wall was 0.6278 m^3/s. Between the first and last measuring points computed discharge was 45.658 m^3/s. Fig.6.27 to 6.37 presents the velocity profiles for each vertical transect. In these figures, shaded upper region shows the surface extrapolated area, while the lower shaded region shows the extrapolated region between the channel beds and last point of measurement.

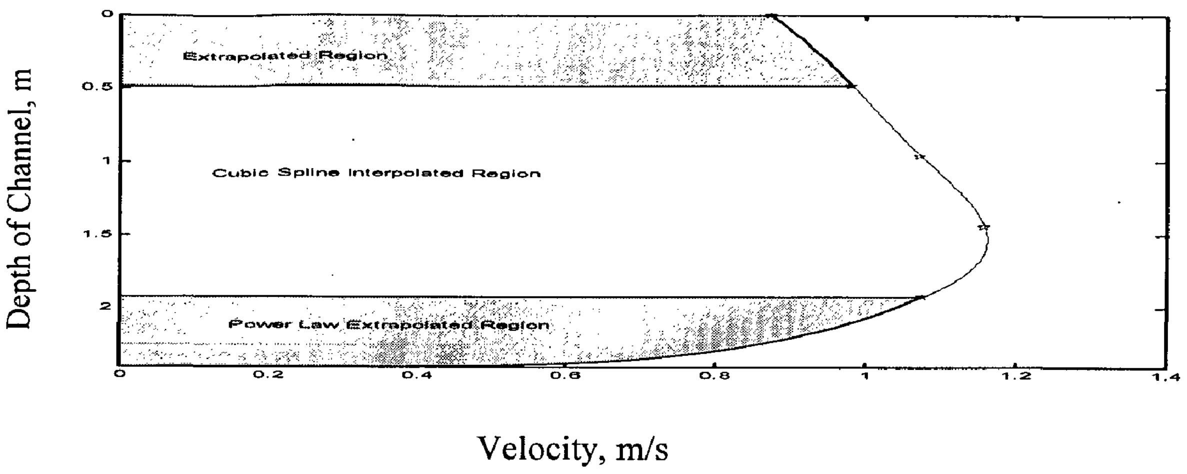


Fig. 6.27 Velocity Profile along I- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.027** m^2/s

Power Law Extrapolated Area = **0.4122997** m^2/s

Total Partial-Discharge = **2.43957** m^2/s

Mean-Velocity at I Vertical Transect = **1.01648** m/s

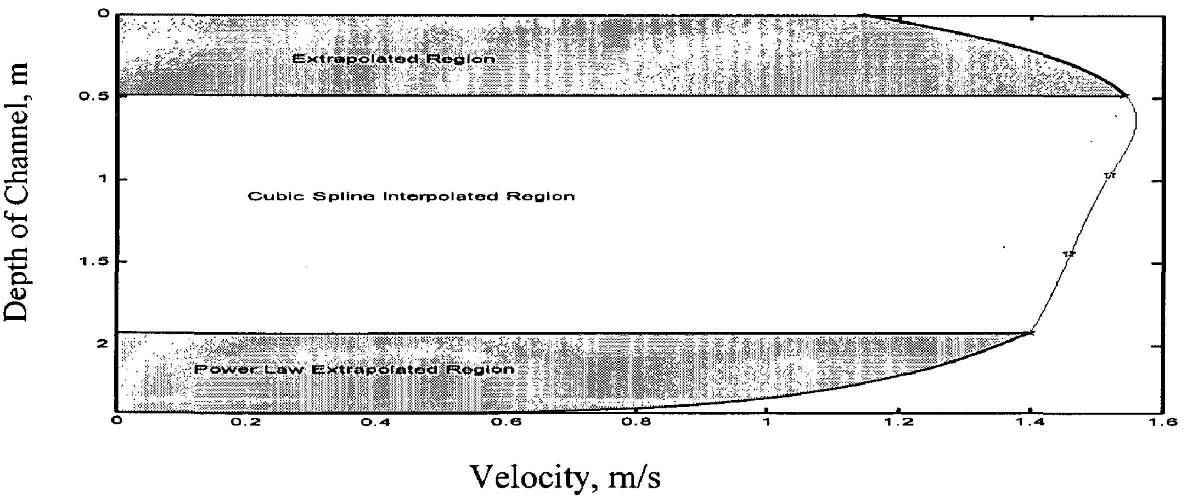


Fig. 6.28 Velocity Profile along II- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.815825** m^2/s

Power Law Extrapolated Area = **0.557747** m^2/s

Total Partial-Discharge = **3.3735** m^2/s

Mean-Velocity at II Vertical Transect = **1.4056** m/s

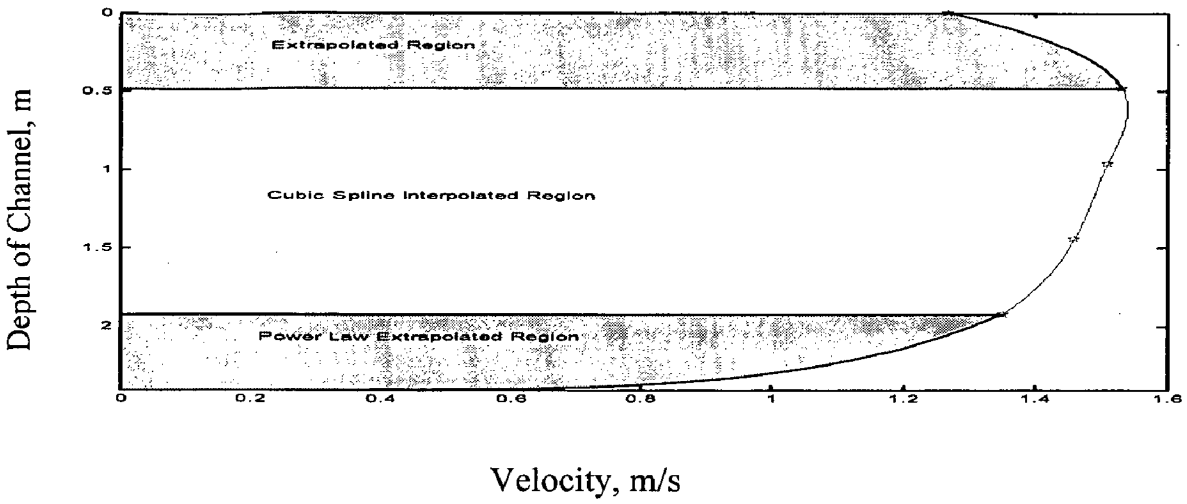


Fig. 6.29 Velocity Profile along III- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.81** m^2/s

Power Law Extrapolated Area = **0.537** m^2/s

Total Partial-Discharge = **3.354** m^2/s

Mean-Velocity at III Vertical Transect = **1.3970** m/s

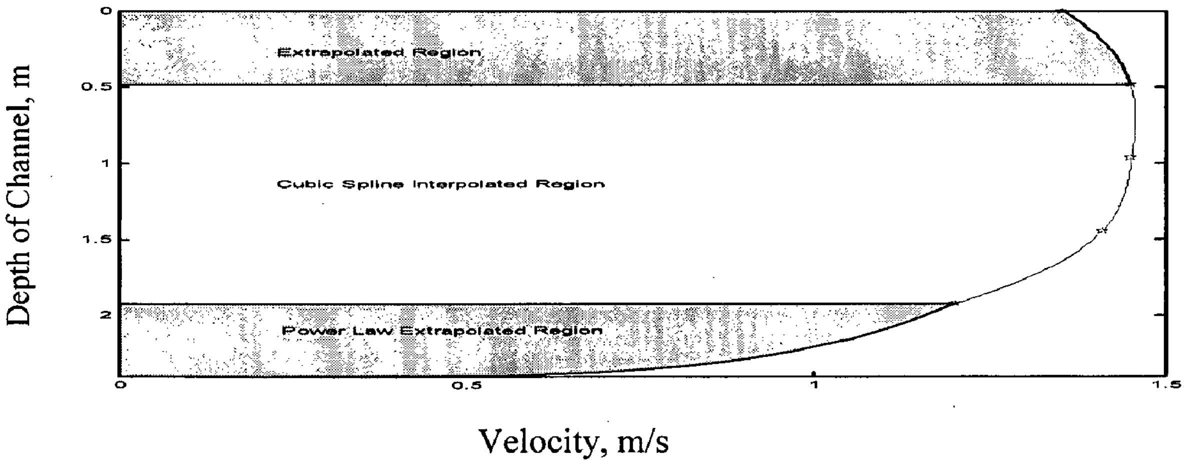


Fig. 6.30 Velocity Profile along IV- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.7028** m^2/s

Power Law Extrapolated Area = **0.478** m^2/s

Total Partial-Discharge = **3.1824698** m^2/s

Mean-Velocity at IV Vertical Transect = **1.3269091** m/s

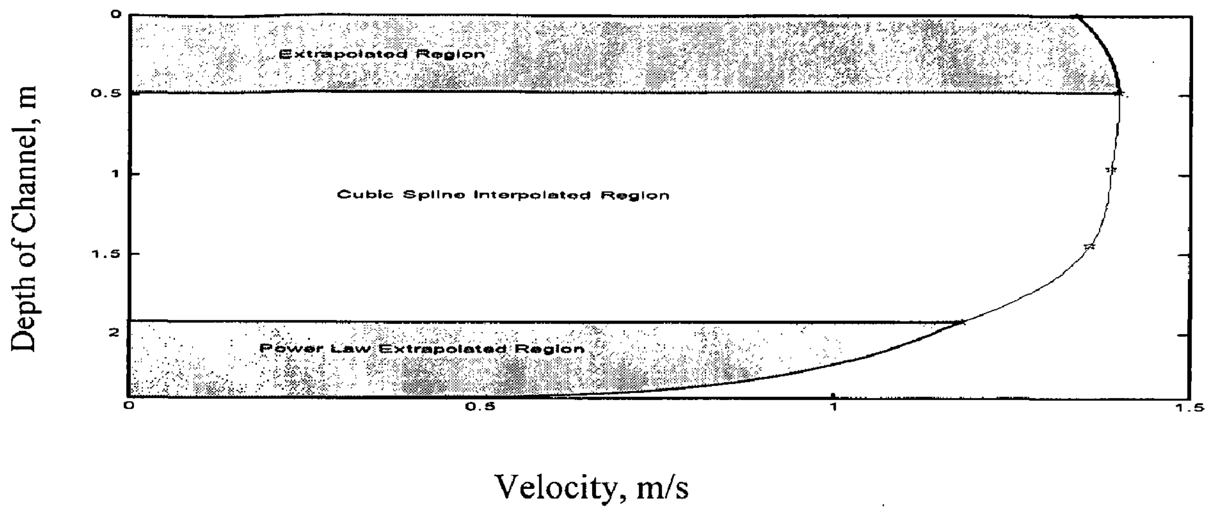


Fig. 6.31 Velocity Profile along V- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.6143** m^2/s

Power Law Extrapolated Area = **0.4701** m^2/s

Total Unit-Discharge = **3.08449** m^2/s

Mean-Velocity at V Vertical Transect = **1.2852** m/s

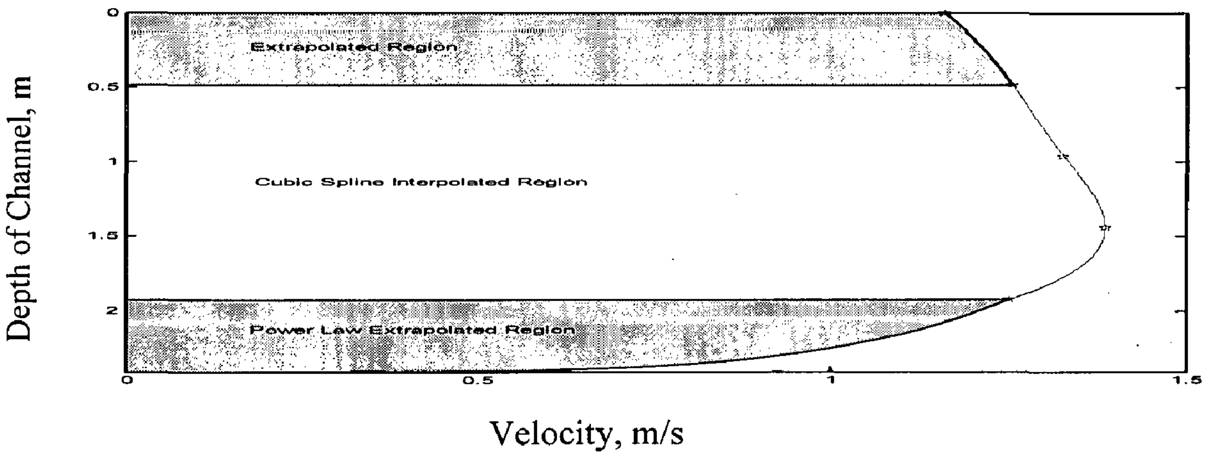


Fig. 6.32 Velocity Profile along VI- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.5023** m^2/s

Power Law Extrapolated Area = **0.49894** m^2/s

Total Unit-Discharge = **3.001** m^2/s

Mean-Velocity at VI Vertical Transect = **1.25052** m/s

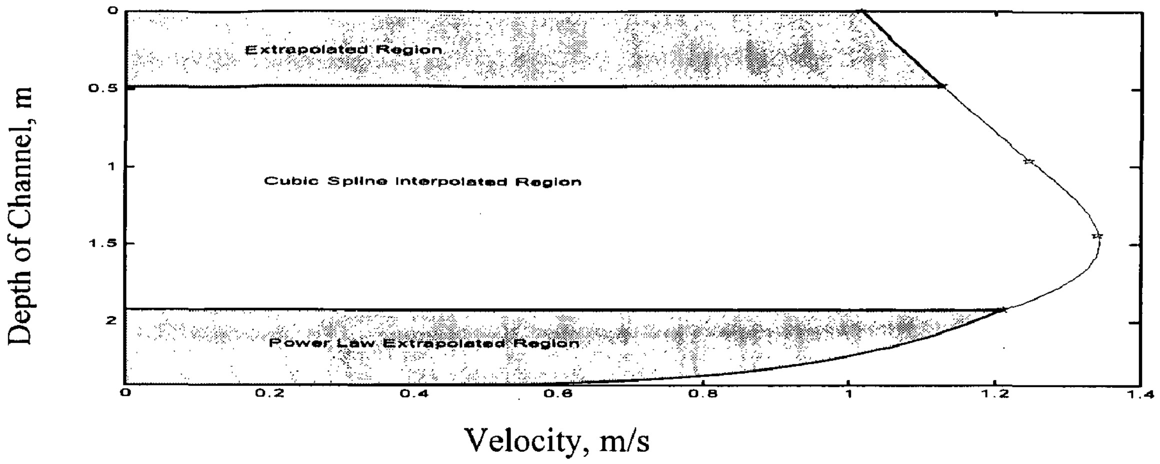


Fig. 6.33 Velocity Profile along VII- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **2.33801** m^2/s

Power Law Extrapolated Area = **0.4829** m^2/s

Total Unit-Discharge = **2.82189** m^2/s

Mean-Velocity at VII Vertical Transect = **1.17037 m/s**

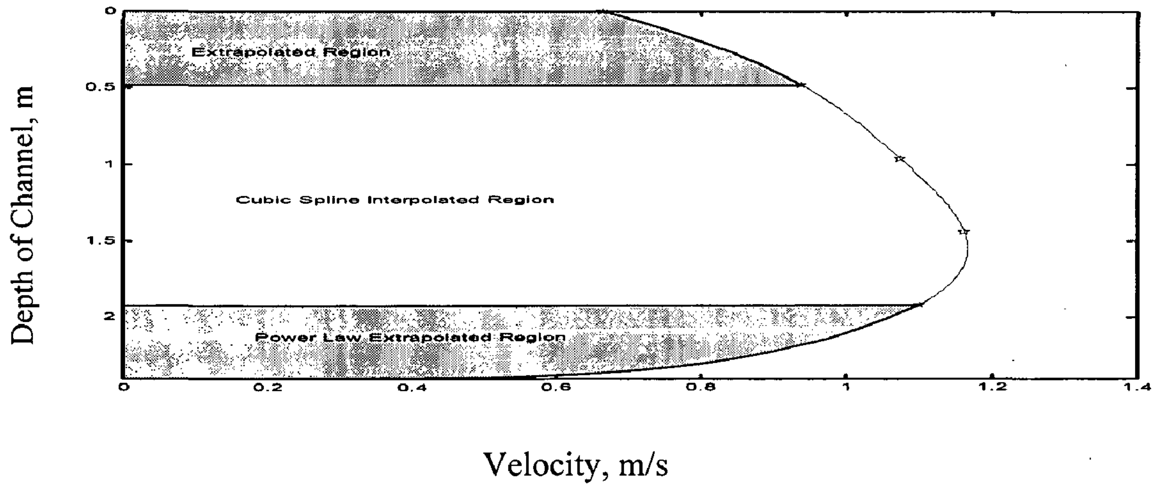


Fig. 6.34 Velocity Profile along VIII- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **1.9728** m^2/s

Power Law Extrapolated Area = **0.4387** m^2/s

Total Unit-Discharge = **2.41578** m^2/s

Mean-Velocity at VIII Vertical Transect = **1.00482**

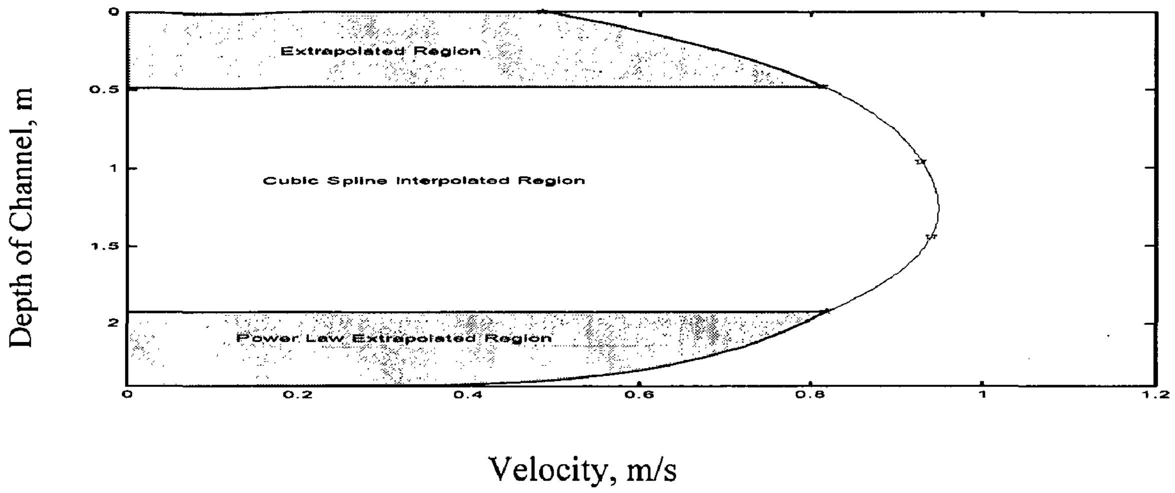


Fig. 6.35 Velocity Profile along IX- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **1.62894** m^2/s

Power Law Extrapolated Area = **0.32616** m^2/s

Total Unit-Discharge = **1.9546** m^2/s

Mean-Velocity at IX Vertical Transect = **0.81452 m/s**

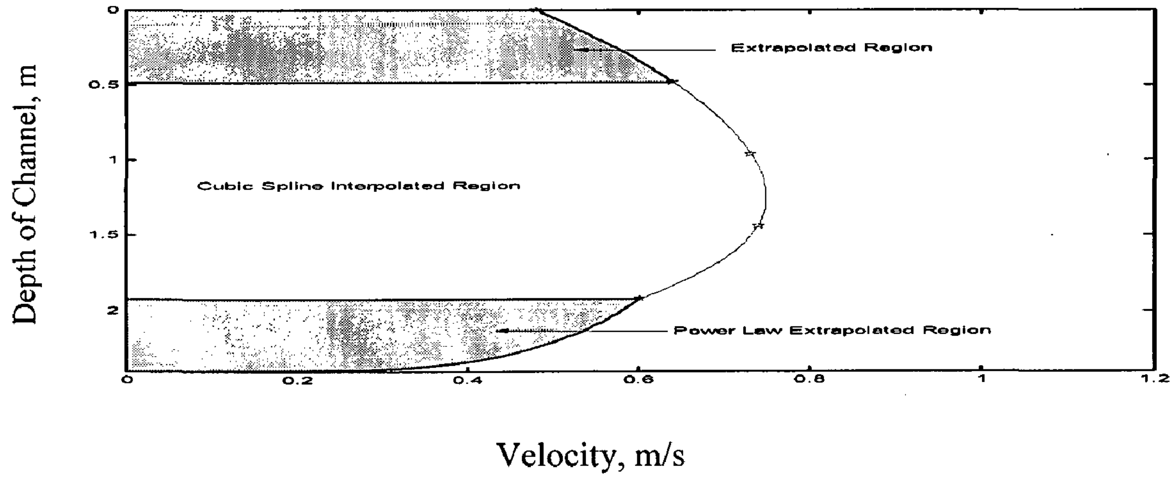


Fig. 6.36 Velocity Profile along X- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **1.2904** m^2/s

Power Law Extrapolated Area = **0.2399** m^2/s

Total Unit-Discharge = **1.530** m^2/s

Mean-Velocity at X Vertical Transect = **0.6373** m/s

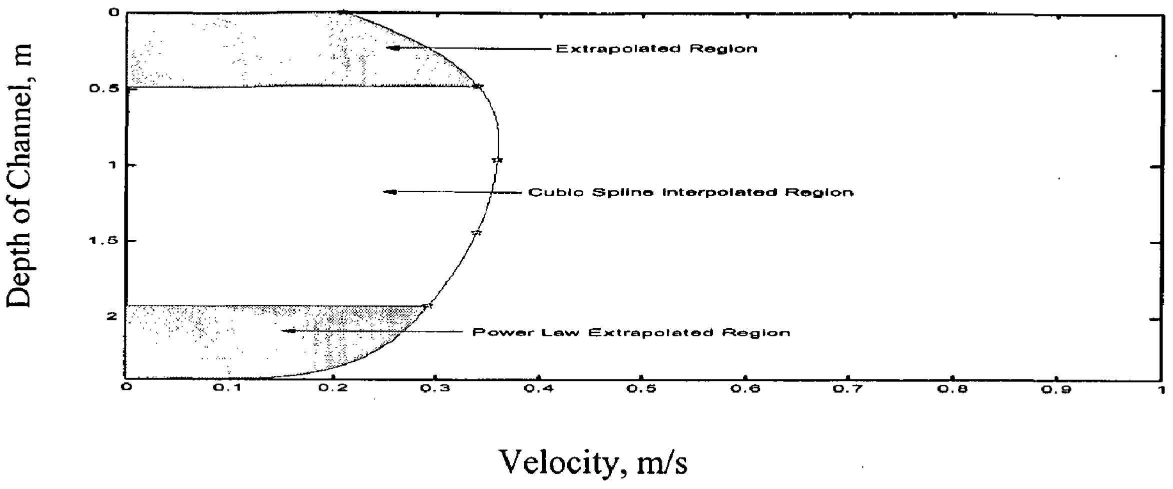


Fig. 6.37 Velocity Profile along XI- Vertical

Cubic Spline Interpolated Region' Area + Surface Extrapolated Area = **0.630697** m^2/s

Power Law Extrapolated Area = **0.11672** m^2/s

Total Unit-Discharge = **0.74746** m^2/s

Mean-Velocity at XI Vertical Transect = **0.311445** m/s

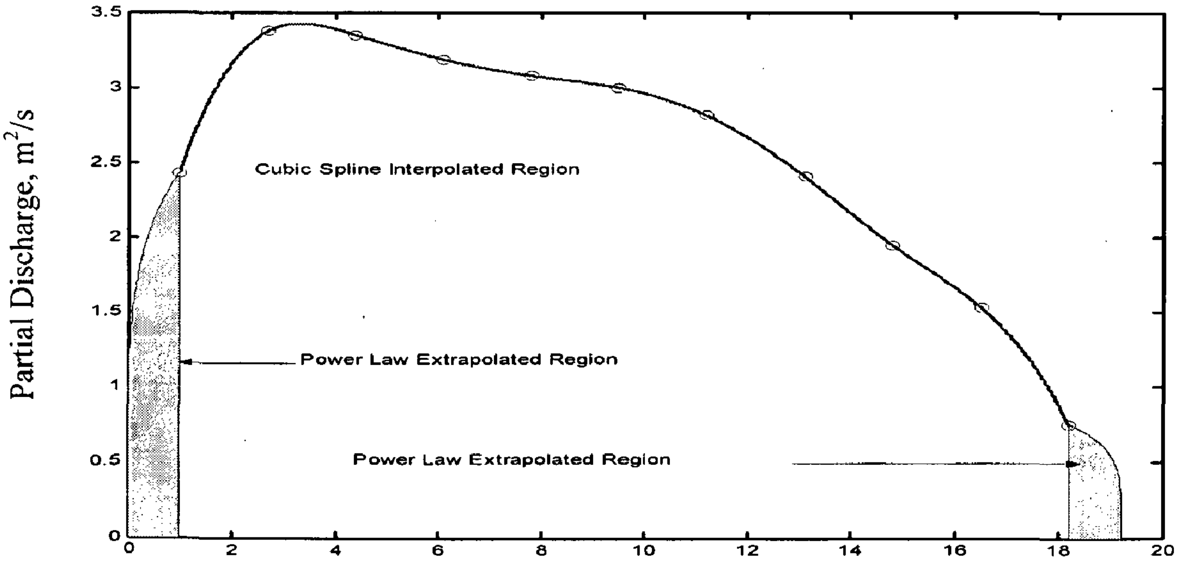


Fig. 6.38 Partial Discharge vs. Distance

Total Discharge in an Open-Channel from 1m to **18.21** m Width = **45.658** m^3/s

Left Side Discharge = **2.0296** m^3/s

Right Side Discharge = **0.6278** m^3/s

Total - Discharge = **48.3154** m^3/s

Average Velocity in the channel = **1.047965** m/s.

A plot of partial discharge versus the distance from the left side of channel is shown in Fig. 6.38. Shaded region at the two ends of graphs shows the extrapolated area as computed by the power law. The value of m, here was taken as 5, same as in the case of extrapolation between the channel beds and last measuring point. The following section presents the analysis of the computed discharge as computed by the arithmetic as well as the graphical methods.

6.5 Analysis

MATLAB programs were written for the total discharge calculation for the graphical method described previously. The program also, computes the velocity profile in each vertical cross-section and average flow velocity. The average flow velocity in vertical cross-section is computed by the dividing the area under curve or partial discharge to the depth of cross-section.

It was assumed for discharge calculations by curve fitting and cubic spline methods that the velocities at the channel bed and at the two corners of the channel were negligible, hence were taken as zero velocities.

In case of power law extrapolation method the value of power law exponent was taken to be 5, as recommended by ISO-748 and IS-1192 standard. The interpolation was carried out by Cubic Spline technique, i.e. from 0.48m to 1.92m for the each vertical transect in channel. Similarly, the cubic spline interpolation was carried for unit-discharges for each vertical transect along the width of channel. The interpolating range was from 1.0m to18.21m from the left side of channel. The increment for the C.S.I. was set at 0.01m for depth and same for the width of channel. This is considered to be a high degree of accuracy, since it computes the point velocities at every 1.0cm interval.

Surface Extrapolation was done by the same technique and its area under the curve was added to the Cubic Spline Interpolated area.

From the velocity profiles, it was found that flow was non-uniform along the width of the channel, and more flow was found in left side of the channel. The reason for this was the presence of bend upstream of channel. So, instead of uniform flow along the width more flow was towards the left side of channel.

The open-channel discharge computed by the curve fitting method was found out to be 44.19087 m^3/s and that computed by cubic spline method, was 45.6883 m^3/s. The percentage difference between them was due to the reason that the extrapolated surface velocity as computed by cubic spline method was very much in accordance with the assumed velocity profile in the open channel flows, while that computed by curve fitting method was very less, that could not be taken as the actual flow velocity.

The results were, validated by doing the interpolation other way round. The point-velocities were first, laterally interpolated along the depth and then integrated to find the partial discharges for each depth. The average flow velocity was computed by dividing the partials discharge to the total width of channel. The computed results were in good agreement with the former technique discussed above.

Discharge computed with power law extrapolation technique was found out to be 48.31 m^3/s. This was validated by same technique mentioned above, and was in agreement within 1%. The reason for the higher value of discharge when compared by other methods was due to power law used at the bottoms and the two sides of channel.

6.5.1 Uncertainty Analysis

No measurements of a physical quantity can be free from the errors which may be associated with either systematic bias caused by errors in the standardizing equipment or a random scatter caused by a lack of sensitivity of the measuring equipment. The former is unaffected repeated measurements and can be reduced only if more accurate equipment is used for the measurements.

When considering the possible uncertainty of any measurement of the discharge in an open-channel, it is not possible to predict this uncertainty exactly, but an analysis of the individual measurements which are required to obtain discharge can be made [3, 14, 18, 19].

6.5.2 Sources of Uncertainty

The source of uncertainty may be identified by considering a generalized form of the working equation for velocity-Area method:

$$Q = \sum b_i * d_i * \bar{v}_i \qquad (6.1)$$

where Q is the total discharge, m^3/s

b_i is the width of the Channel ,m

d_i is the depth of Channel, m

$\bar{v}_i$ is the average velocity of the water in the i^{th} of the m verticals or segments into which the cross-section is divided.

The overall uncertainty in the discharge is then composed of:

a). Uncertainties in depth; X'_d

b). Uncertainties in widths; X'_b

c). Uncertainties in determination of local point-velocities.

- Time of Exposure, X'_e
- Number of points in a vertical, X'_p
- Current-Meter ratings, X'_c

d). Uncertainties in the use of the velocity-area method particularly those concerned with the number of verticals and the number of points in each verticals.

a). Uncertainty in Width, (X'_b)

Range of Width = 0m to 19.21 m

Likely, Absolute Error = 5 mm

Thus, Relative Error = ± 0.02602811 %

b). Uncertainty in Depth, (X'_d)

Range of Depth = 0m to 2.4 m

Likely, Absolute Error = 5 mm

Thus, Relative Error = ± 0.208333 %

c). Uncertainty in Mean-Velocity, X'_v

Time of Exposure = 120s

Method used = Velocity Distribution

Current-Meter = Individual Calibrated

Number of Verticals = 11

So, mean-velocity uncertainty = ± 2 %

Total Uncertainty for Velocity-Area Method = $\pm (X'^2_b + X'^2_d + X'^2_v)^{1/2}$

$= \pm 2.1074$ %

This implies that all five methods of discharge calculation described earlier the uncertainty will be of ± 2.1074 % respectively.

Chapter 7

DATA ACQUISITION AND ITS FPGA IMPLEMENTATION

Data Acquisition has become an indispensable task in almost all areas of sciences and also in modern day industry. Most research projects need data in order to answer a proposed research problem. Industries need to collect data for the purpose of analysis and proposing new designs. So, it can be defined as follows;

"**Data acquisition** *is the term used to define the process of collecting information, usually in an automated fashion, on a broad variety on variables."*

A typical data acquisition system consists of individual sensors with the necessary signal conditioning, data conversion, data processing, multiplexing and associated transmission, and storage and display systems. Processing may consist of a large variety of operations, ranging from simple comparison to complicated mathematical manipulations. It can be for such purposes as collecting information (averages), converting the data into a useful form and then using this data for the purpose of controlling a process and various other purposes, [26,27].

7.1 Objectives of Data Acquisition Systems

i. It must acquire the necessary data, at correct speed and at the correct time.

ii. Use of all data efficiently to inform the operator about the state of the plant.

iii. It must monitor the complete plant operation to maintain optimum and safe operation.

iv. It must be flexible and capable of being expanded for future requirements.

v. It must be reliable.

7.2 Instrumentation setup for DAQ

As in our research problem, we need to acquire the data or point-velocities from the propeller current meters for the computation of open-channel discharge; a necessary data acquisition unit must be used. The data acquisition card can be used for the acquisition of the same. The data acquisition, (DAQ) card is a multifunction device that contain analog to digital converter, digital to analog converter, timer, counter, inbuilt function generator etc. The PCI-6024E card was used. Specification for it is given in the Appendix A.

The output signal from the propeller current meters is in the form of pulses. Counting these pulses for a given period of time and applying the calibration equation for each current-meter we get the point-velocity at that location. When mean-velocity in the channel is multiplied by the area of cross-section, we get the total discharge in the channel.

A virtual instrument is made in the LabVIEW environment for doing the necessary data acquisition from the propeller current meters. As described in section (4.3), a virtual instrument basically consists of front panel and the corresponding block diagram. A code for virtual instrument is written in block diagram window in 'G' and front panel is the actual Graphical User Interface, GUI. The GUI's made for the same is illustrated in section 7.2.1.

Fig. 7.1 shows the instrumentation setup for the computation of open channel discharge with the help of current meters. A matrix of current meter location is formed from which, the pulse outputs from these current meters are fed into Digital Input/Output Lines of the BNC-connector, 2120. The BNC-Connector acts as the interface to the personal computer and the DAQ card, PCI-6024E. After that flow-velocity at point is determined in LabVIEW software and computation of discharge is done in MATLAB software. The algorithm for the point velocity acquisition from the current meters is shown in Fig. 7.2 and algorithm for discharge computation was presented in section (6.3).

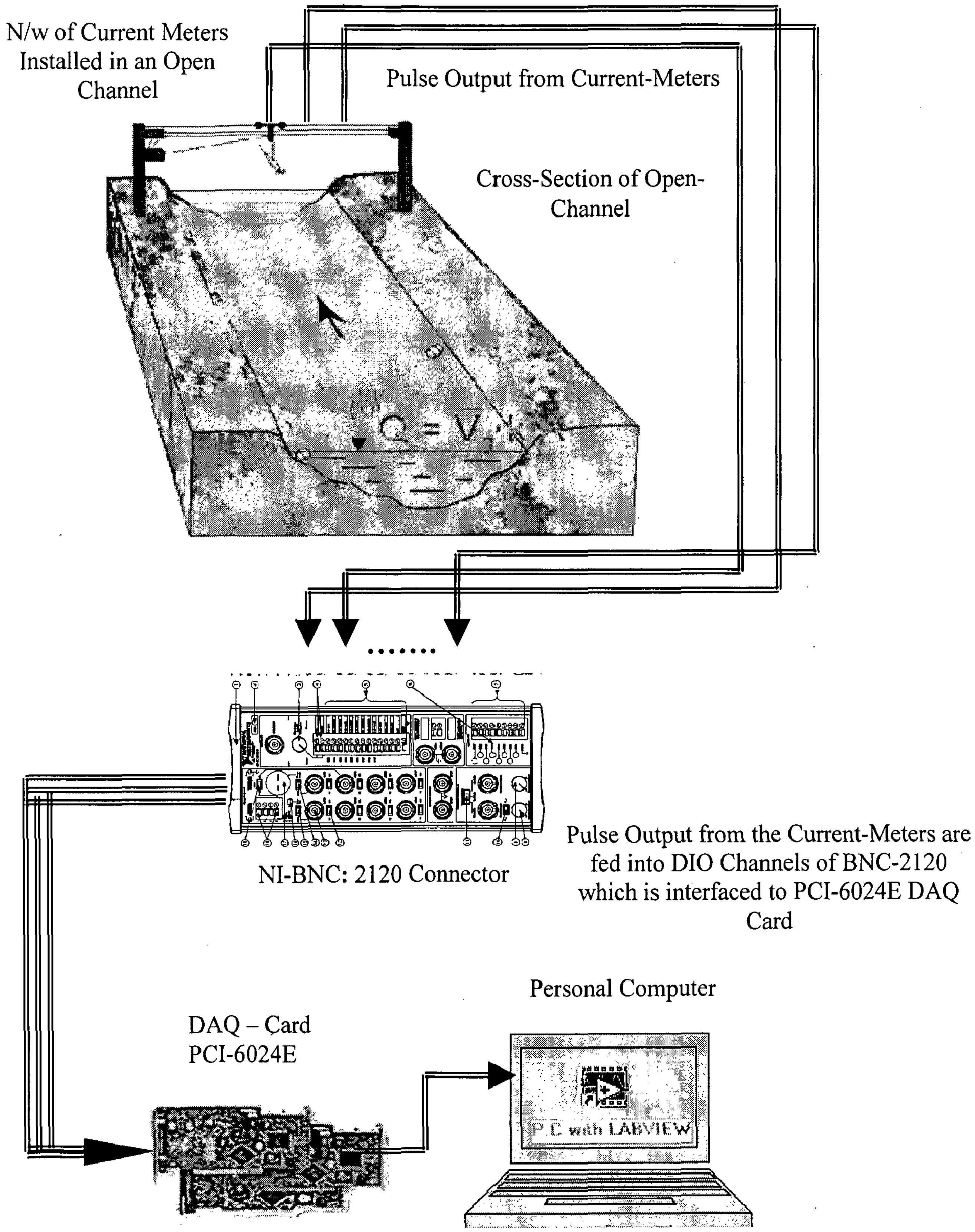

N/w of Current Meters Installed in an Open Channel
Pulse Output from Current-Meters
Cross-Section of Open-Channel
Q = V
•••••••
NI-BNC: 2120 Connector
Pulse Output from the Current-Meters are fed into DIO Channels of BNC-2120 which is interfaced to PCI-6024E DAQ Card
Personal Computer
DAQ – Card PCI-6024E
P.C with LABVIEW

Fig.7.1 Instrumentation Setup for Data Acquisition from Current-Meters

In order to acquire the point velocity from the current meters, the basic scheme for the same is shown in Fig. 7.2.

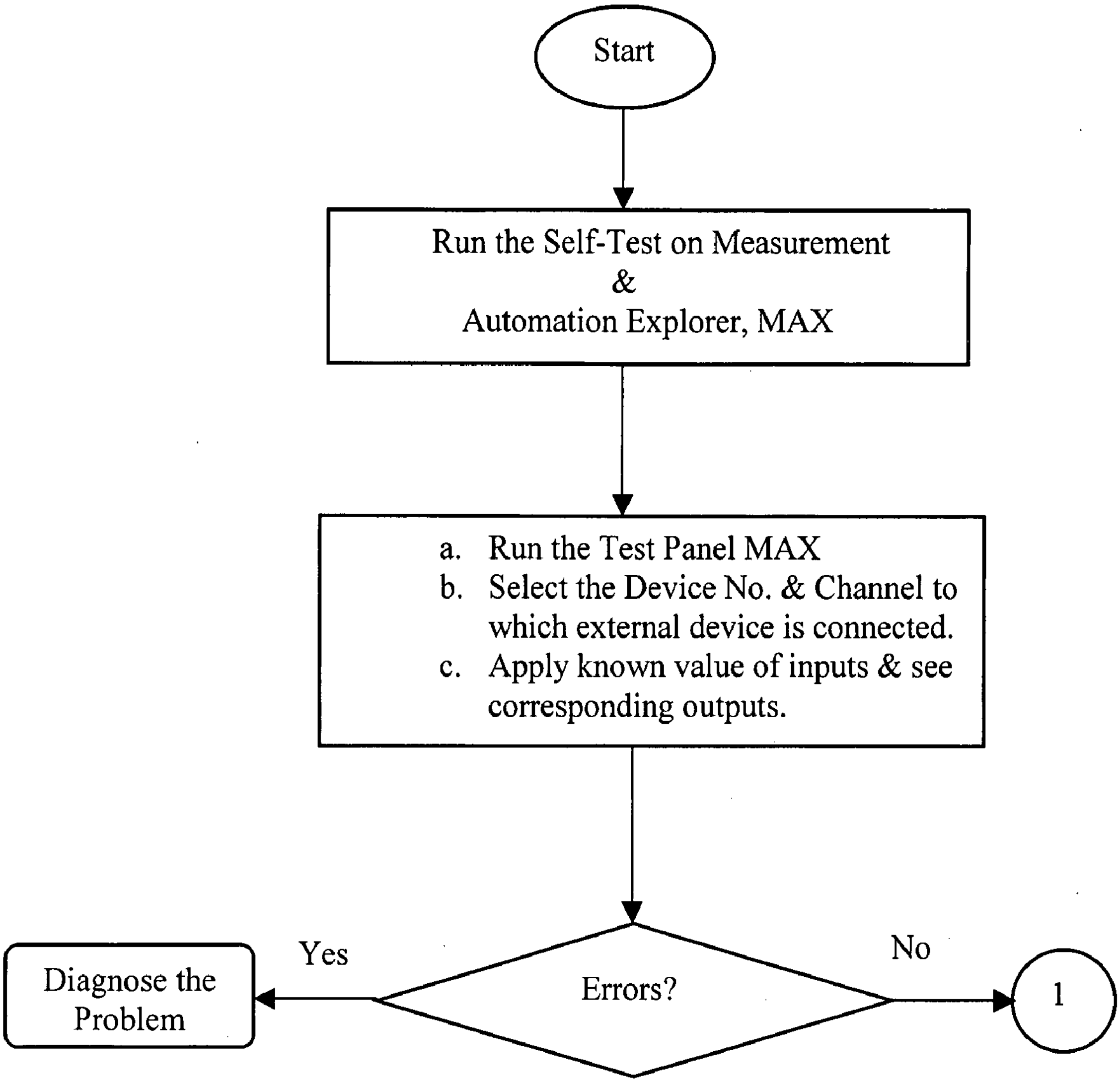

Start
Run the Self-Test on Measurement & Automation Explorer, MAX
a. Run the Test Panel MAX
b. Select the Device No. & Channel to which external device is connected.
c. Apply known value of inputs & see corresponding outputs.
Errors?
Yes
Diagnose the Problem
No
1

Fig 7.2. Flow-Chart for Point-Velocity Acquisition in LabVIEW, (Contd.)

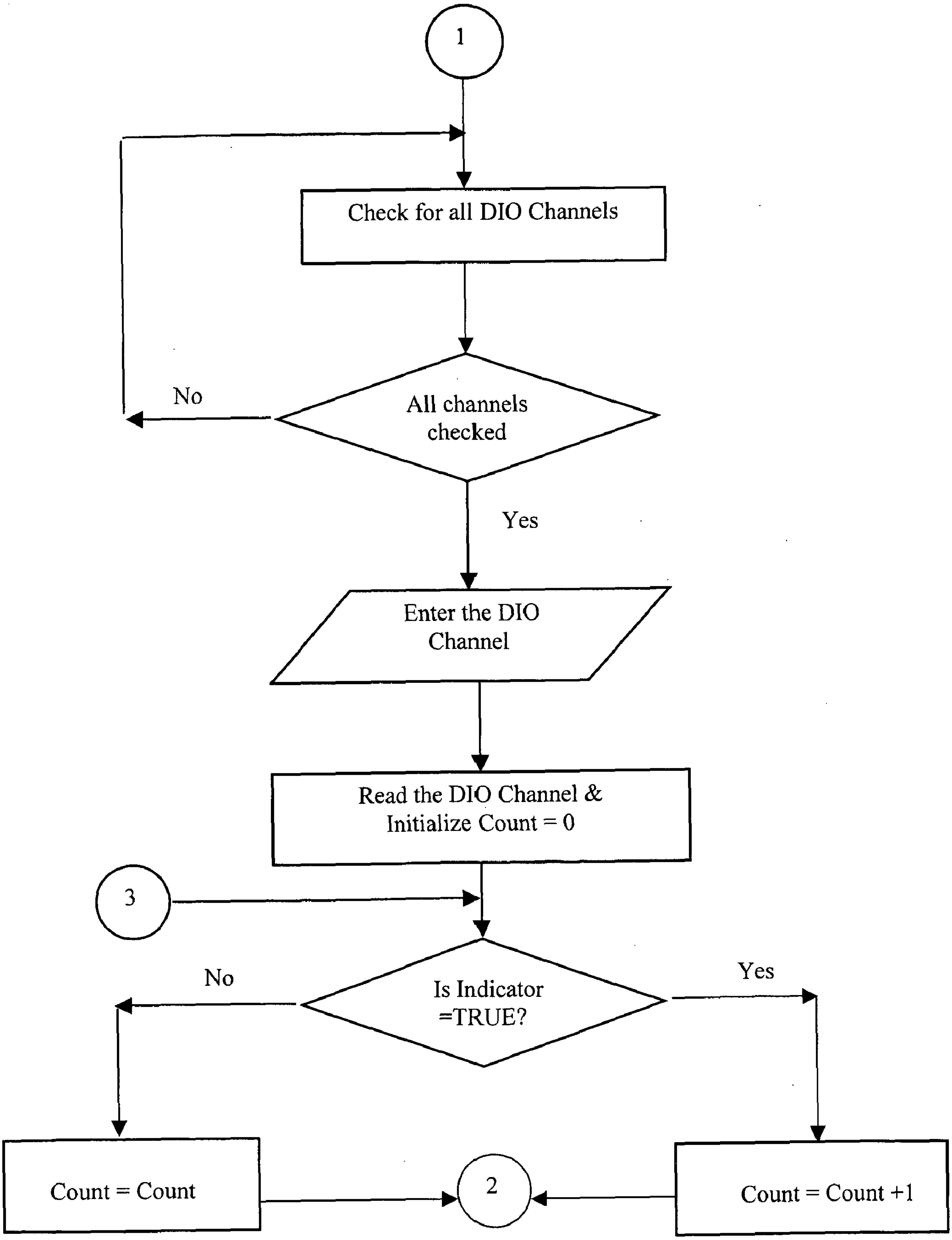


Fig 7.2. Flow-Chart for Point-Velocity Acquisition in LabVIEW (Contd.)

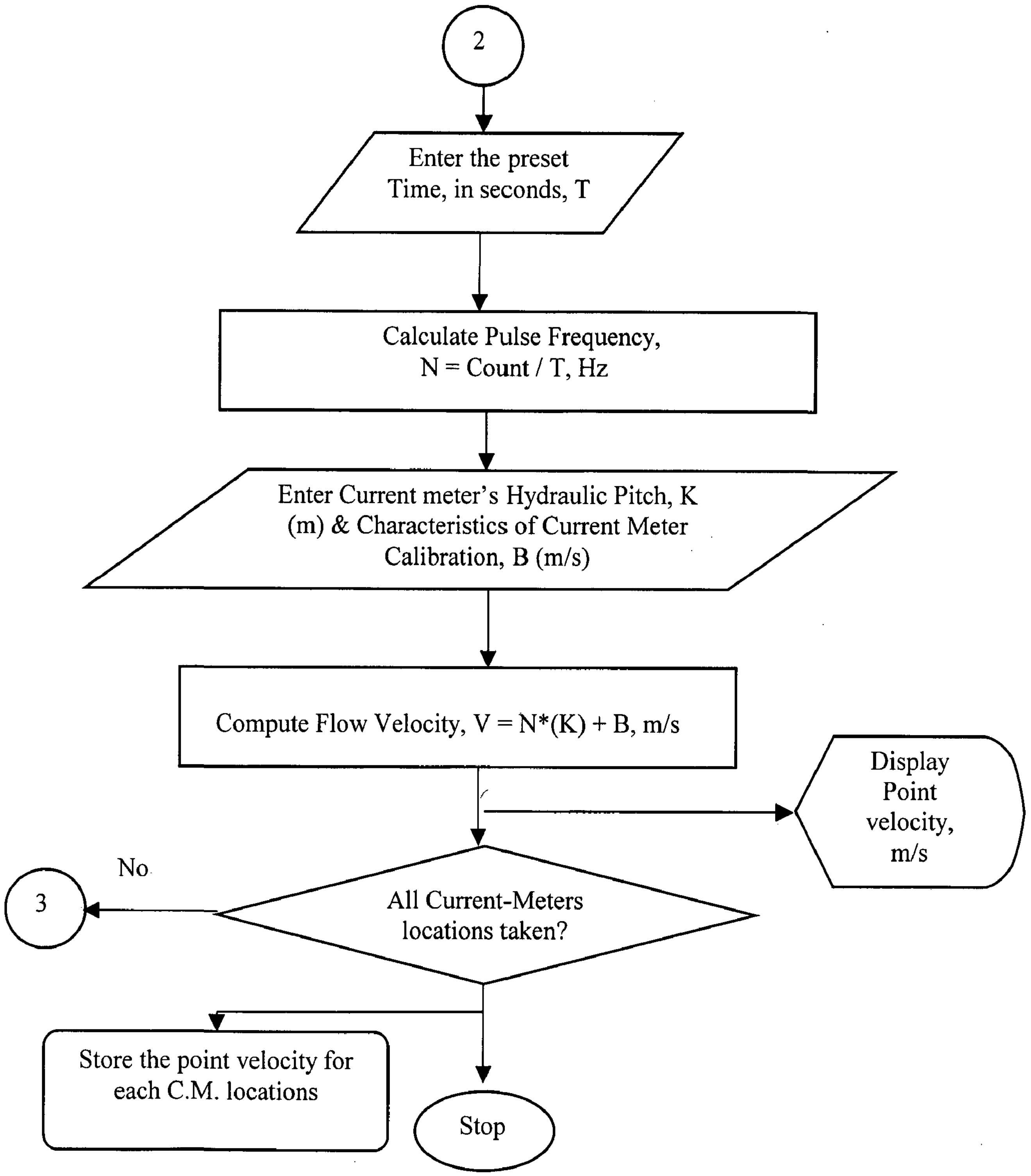
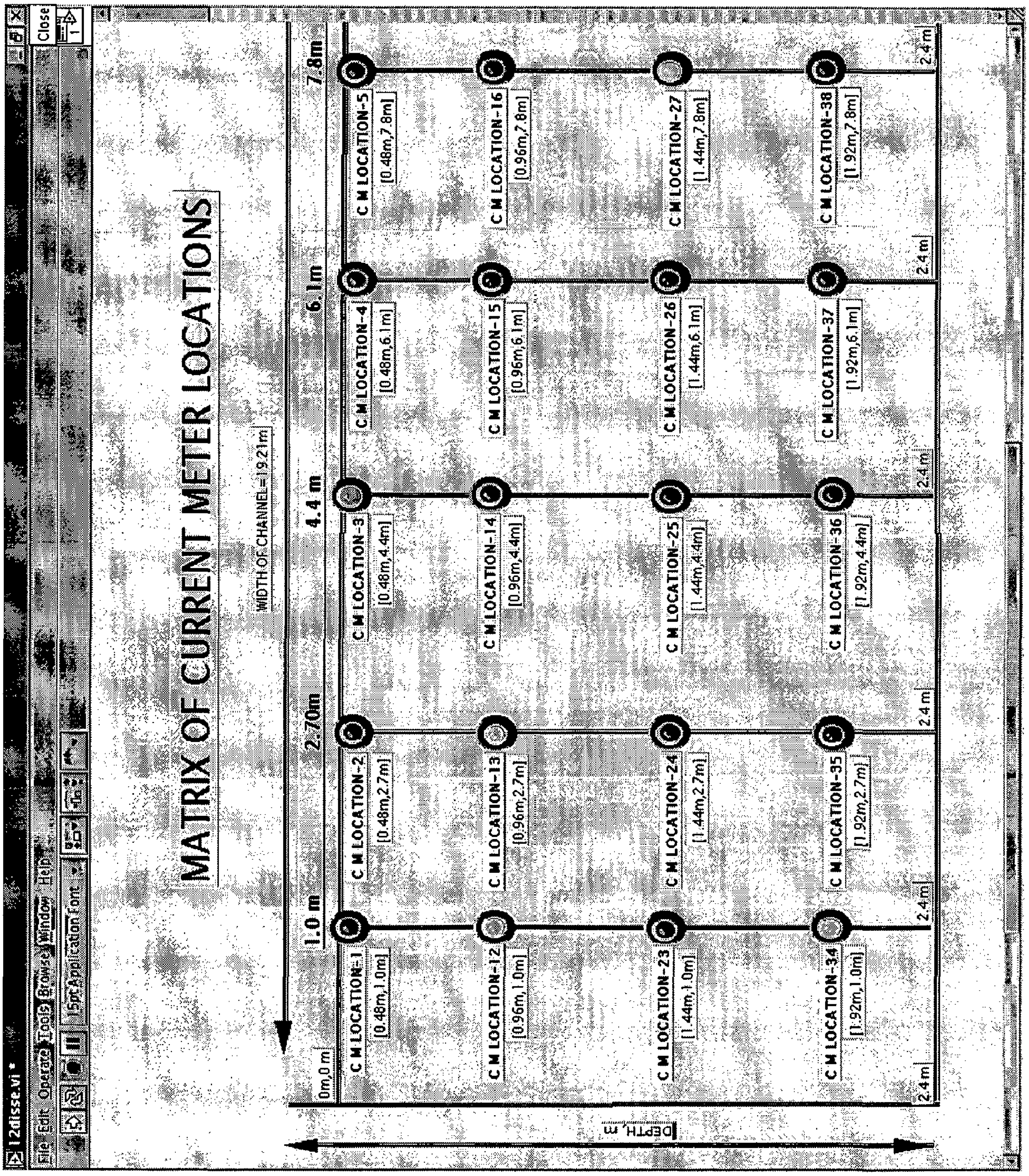

2
Enter the preset Time, in seconds, T
Calculate Pulse Frequency, N = Count / T, Hz
Enter Current meter's Hydraulic Pitch, K (m) & Characteristics of Current Meter Calibration, B (m/s)
Compute Flow Velocity, V = N*(K) + B, m/s
Display Point velocity, m/s
All Current-Meters locations taken?
No
3
Store the point velocity for each C.M. locations
Stop

Fig 7.2 Flow-Chart for Point-Velocity Acquisition in LabVIEW

7.2.1 Graphical User Interface for Data Acquisition from Current-Meters

The GUI is made in LabVIEW environment for the acquisition of point velocity from the current meters. The front panel is shown in Fig. 7.3. This is the indicator panel, in which a matrix of current meter locations is shown. An array of eleven current meters was placed at a depth of 0.48, 0.96, 1.44 and 1.92m from the water surface. The LED's in Fig.7.3 shows the location of current meters in the channel.



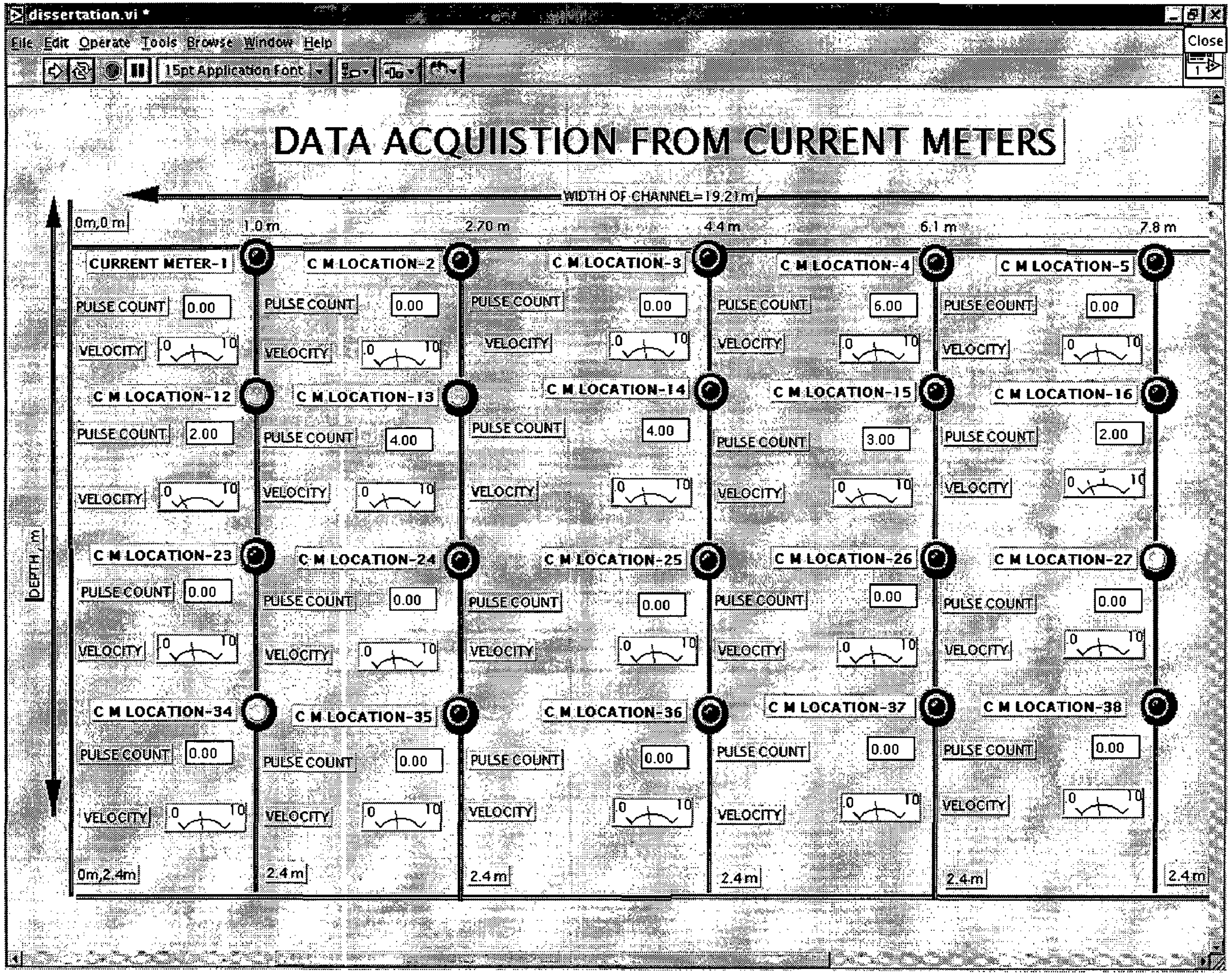
MATRIX OF CURRENT METER LOCATIONS

WIDTH OF CHANNEL=19.21m

1.0 m	2.70m	4.4 m	6.1m	7.8m
C M LOCATION-1 [0.48m,1.0m]	C M LOCATION-2 [0.48m,2.7m]	C M LOCATION-3 [0.48m,4.4m]	C M LOCATION-4 [0.48m,6.1m]	C M LOCATION-5 [0.48m,7.8m]
C M LOCATION-12 [0.96m,1.0m]	C M LOCATION-13 [0.96m,2.7m]	C M LOCATION-14 [0.96m,4.4m]	C M LOCATION-15 [0.96m,6.1m]	C M LOCATION-16 [0.96m,7.8m]
C M LOCATION-23 [1.44m,1.0m]	C M LOCATION-24 [1.44m,2.7m]	C M LOCATION-25 [1.44m,4.4m]	C M LOCATION-26 [1.44m,6.1m]	C M LOCATION-27 [1.44m,7.8m]
C M LOCATION-34 [1.92m,1.0m]	C M LOCATION-35 [1.92m,2.7m]	C M LOCATION-36 [1.92m,4.4m]	C M LOCATION-37 [1.92m,6.1m]	C M LOCATION-38 [1.92m,7.8m]

Fig. 7.3 A V.I.'s Front Panel showing Matrix of Current Meter Locations

Fig. 7.4 shows the indicator palette for the display of pulse-count and the corresponding computed point velocities from the current meters. The figure also shows the locations of current meters with respect to depths and widths of channel.



dissertation.vi *

DATA ACQUIISTION FROM CURRENT METERS

WIDTH OF CHANNEL=19.21m

0m,0 m | 1.0 m | 2.70 m | 4.4 m | 6.1 m | 7.8 m

CURRENT METER-1 — PULSE COUNT 0.00
C M LOCATION-2 — PULSE COUNT 0.00
C M LOCATION-3 — PULSE COUNT 0.00
C M LOCATION-4 — PULSE COUNT 6.00
C M LOCATION-5 — PULSE COUNT 0.00

C M LOCATION-12 — PULSE COUNT 2.00
C M LOCATION-13 — PULSE COUNT 4.00
C M LOCATION-14 — PULSE COUNT 4.00
C M LOCATION-15 — PULSE COUNT 3.00
C M LOCATION-16 — PULSE COUNT 2.00

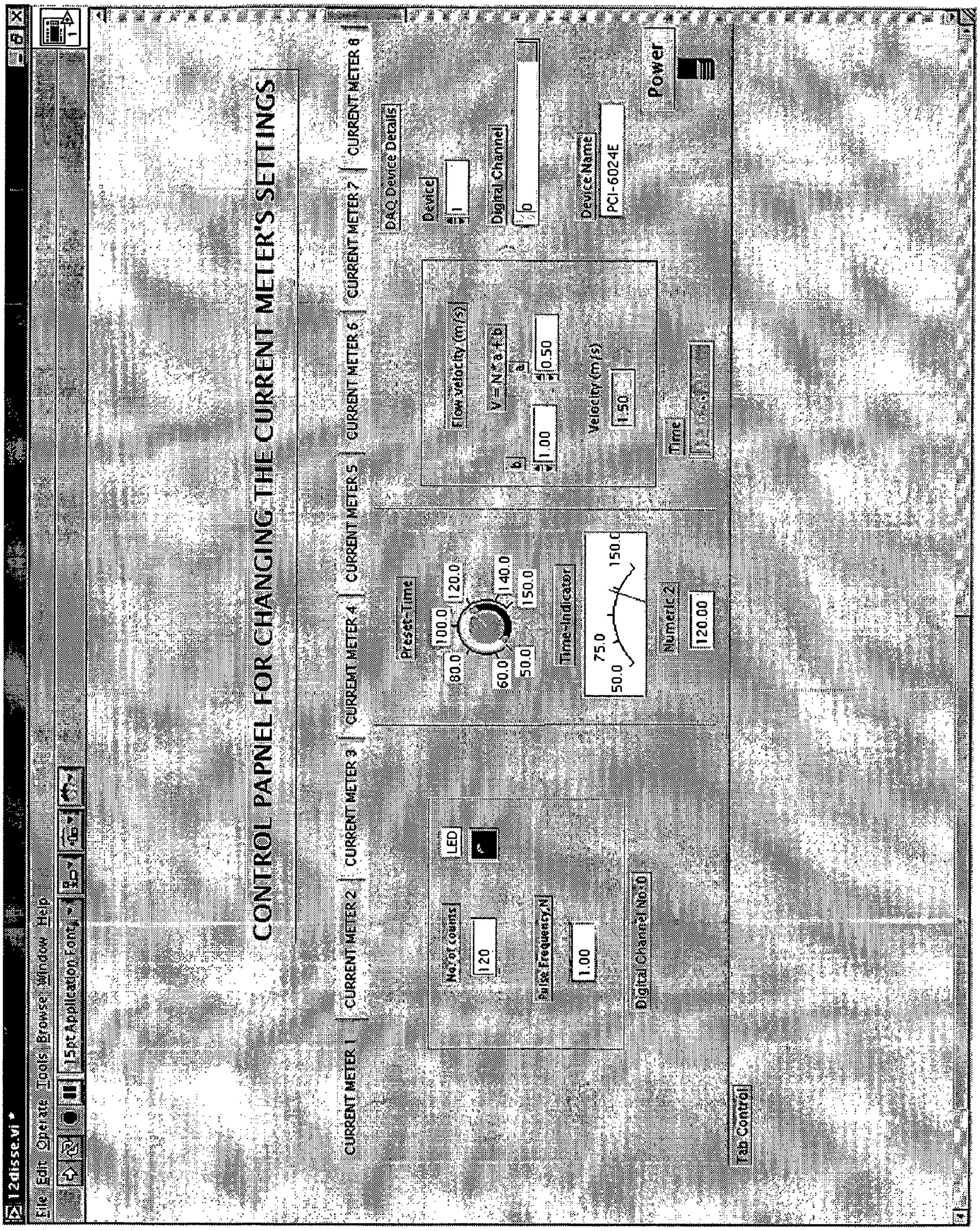
C M LOCATION-23 — PULSE COUNT 0.00
C M LOCATION-24 — PULSE COUNT 0.00
C M LOCATION-25 — PULSE COUNT 0.00
C M LOCATION-26 — PULSE COUNT 0.00
C M LOCATION-27 — PULSE COUNT 0.00

C M LOCATION-34 — PULSE COUNT 0.00
C M LOCATION-35 — PULSE COUNT 0.00
C M LOCATION-36 — PULSE COUNT 0.00
C M LOCATION-37 — PULSE COUNT 0.00
C M LOCATION-38 — PULSE COUNT 0.00

0m,2.4m | 2.4 m | 2.4 m | 2.4 m | 2.4 m | 2.4 m

Fig. 7.4 A V.I.'s Front Panel created in LabVIEW Software for the determination of point-velocity from the Current-Meters

Fig. 7.5 shows the GUI for changing the current meter's settings. Also, shown in the figure is the DAQ card specification.



No.of counts: 120

Pulse Frequency,N: 1.00

Numeric 2: 120.00

a: 0.50

b: 1.00

Velocity (m/s): 1.50

Time

Device: 1

Digital Channel: 0

Device Name: PCI-6024E

Fig. 7.5 A V.I.'s Front Panel created in LabVIEW Software for the changing Current Meter's setting

7.2.2 Block Diagram Code for Computation of Point-Velocities

Fig 7.6 shows the algorithm for the computation of point velocity in 'G' language. As every front panel has a corresponding terminal on the block diagram in the virtual instrument made in LabVIEW, corresponding terminal for Fig 7.3, 7.4 & 7.5 are shown in Fig.7.6. This is the single block diagram code for the counting the number of pulses and for correspondingly computation of point velocity. For the acquisition in other channels, code remains same with the exception of different channel number assigned to DIO channel control as shown in Fig. 7.5.

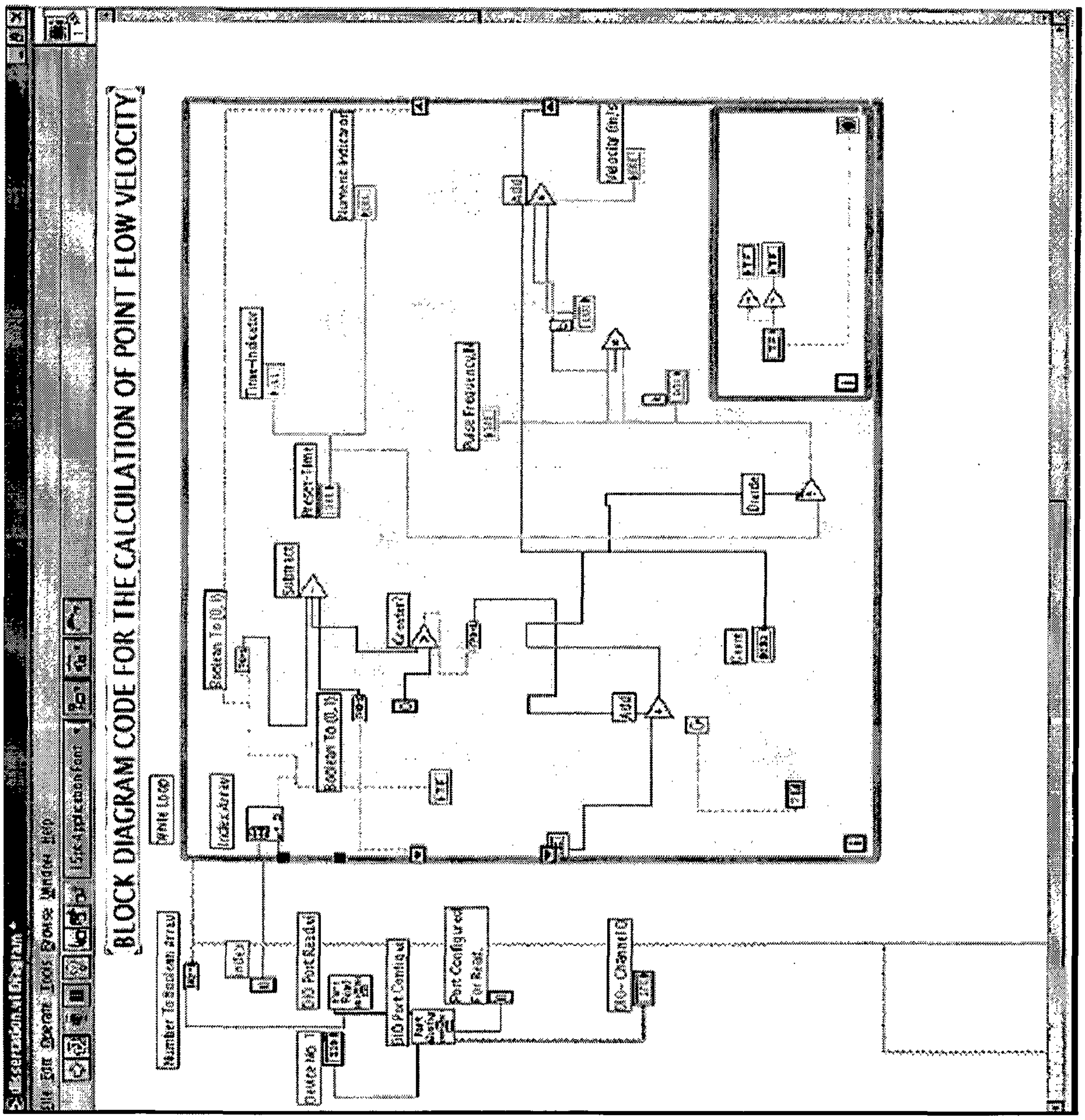


Fig. 7.6 A V.I.'s Block-Diagram code created in LabVIEW Software for Point-Velocity Determination

7.3 FPGA Implementation for Data Acquisition

As already, discussed in section (2.2), the output from the propeller current meters are in the form of pulses, which is further measured, processed and scaled by commercial data acquisition cards and with the help of software provided by the company. Using these, commercial devices and costly software, there is the problem that the user entirely depends on the specification provided by the company for the software as well as the hardware.

So, with the advent of Very Large Scale Integration, (VLSI) any of the functions performed by traditional devices and instruments can be implemented on the hardware. Hardware implementations of genetic algorithm, neural networks, etc. were made possible with the advances in the area of VLSI. Being re-programmable, ease of adjustment and fast development time, FPGA's have gained rapid progress mainly in areas in research and development. So, a technique is proposed in which the data acquisition is done from the propeller current-meters on the Field Programmable Gate Array chip.

Here, acquisition of data from the current meters requires the counting the number of pulses. A dedicated eight bit pulse-counter is designed for this purpose. The program for the same is developed in VHDL, as the hardware descriptor language. The reason for choosing this language is that, it is universally used for burning the FPGA's, easy to understand and standardized by IEEE. Writing a VHDL code requires the entity declaration and the corresponding architecture body.

An entity defines the external specifications of a circuit or a sub-circuit. Using the information provided in an entity declaration, all the information needed to connect that portion of a circuit in to a higher-level circuit is obtained. The actual information of the circuit is not included in the entity declaration.

The behavior modeling of the proposed logic is done in architecture of the corresponding entity. It can be either, behavioral, structural or concurrent. An entity can have a number of different or unique architecture and can be shown as in the figure 7.7.

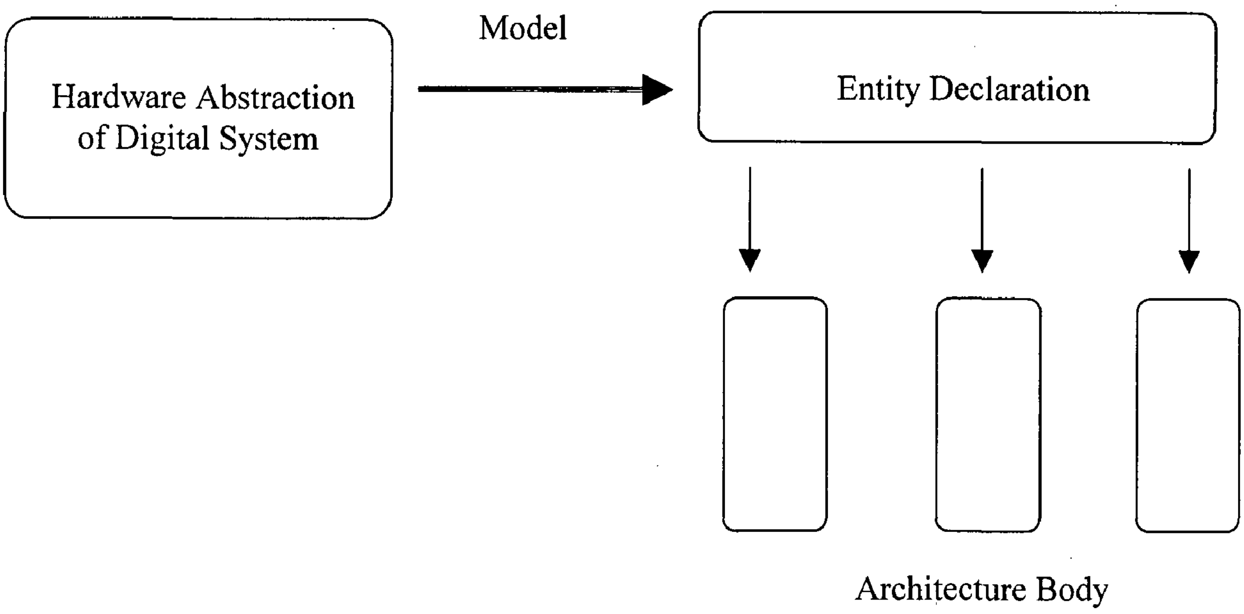


Fig.7.7 Entity and its Model Architectures

Every entity declaration must be accompanied by at least one corresponding architecture. An architecture body may be modeled as a set of interconnected components or as a set of concurrent assignment statements or as a set of sequential assignments statements or as a combination of the above three. The name of the architecture body is chosen randomly. The most commonly used approaches for the architecture description are behavioral and structural.

7.4 Scheme for Data Acquisition on FPGA

Here, we have used the behavioral methodology for the desired function. Structural type of technique can also, be used. As described above, two sets of inputs were taken, namely counter clock and reset or clear input. The input pulse is connected to clock input. The count is the variable of type integer. The minimum value of count is 0 and the maximum can be 255. It will be reset to 0 after the next clock pulse, i.e. like a conventional counter after the terminal count is reached, sets back to 0. The main logic behind the data acquisition from the propeller current meters; involves the initialization of count to zero, if clear input is one else, if clock input has a event and clock input is one, then increment the value of local variable count to one, else do not change the value. If the value of count reaches the terminal count, i.e. 255, set the value of count to 0. Lastly, substitute the values of local variable count to entity port.

The reason for choosing the data out port to integer is that, it was found that the entity declaration of ports does not specify the bit-width for the data to be loaded into the pulse-counter entity. Also, choosing the data type as integer, multiple type of counters can be instantiated at the same time. The size of pulse counter was taken as 8-bit since in practical field measurements the number of counts from the current meters lies in the 0-255 range generally, for a flow velocity around 1m/s with the averaging time, of 100-120 seconds.

In this research work, behavioral simulations were carried in XILINX Modelsim. Generating the bit file, which is downloaded to FPGA chip after successful synthesis, floor-planning, placement and routing and implementation, was done in XILINX ISE, Web-Package version 6.3. The FPGA for the data acquisition was programmed for XILINX Spartan-X2S15CS144 device type having 144 pins in all.

7.4.1 Software used for FPGA Implementation

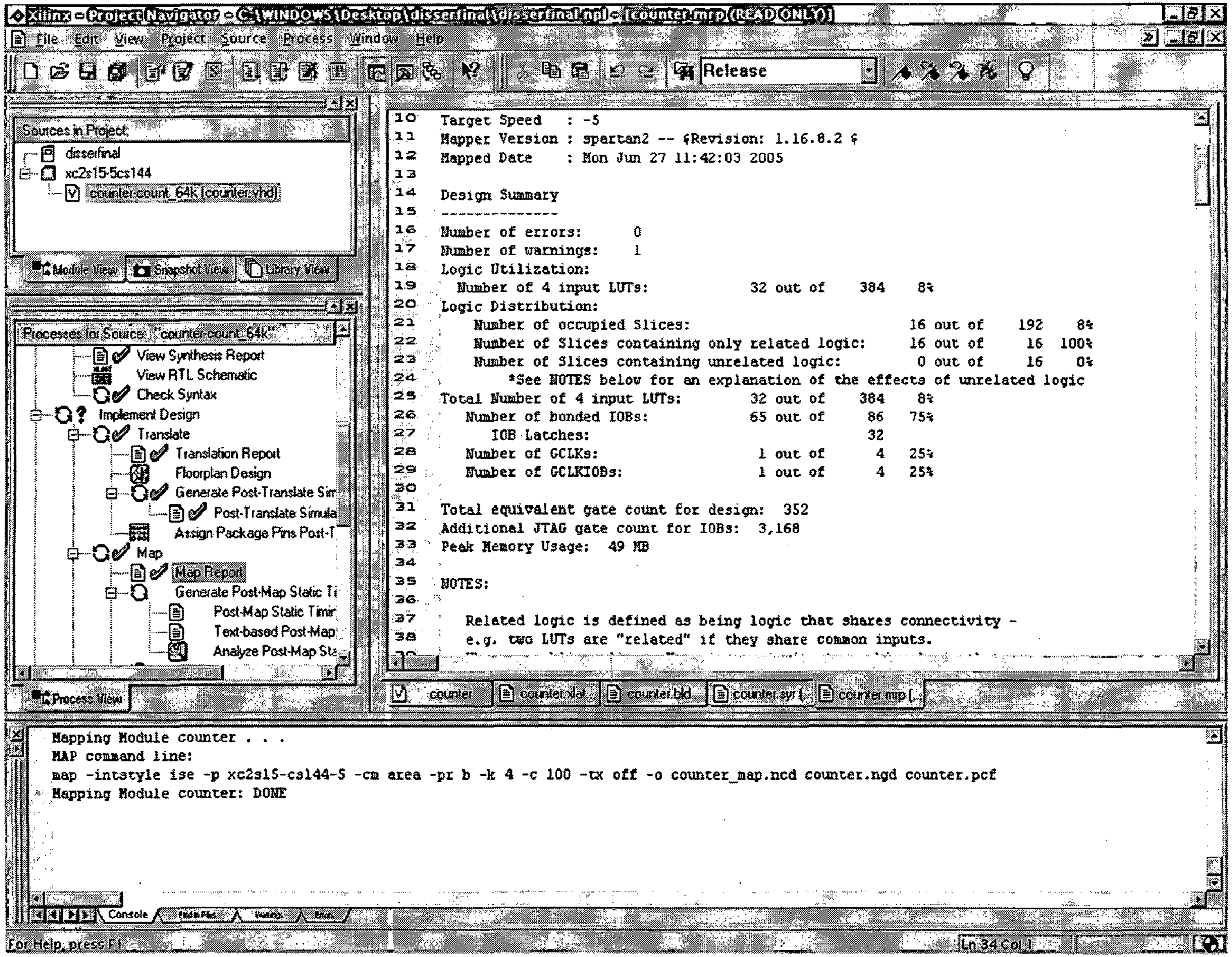
Integrated Software Environment (ISE) is the Xilinx design software suite. ISE can be used by a full spectrum of designers, from the first time CPLD designer to the experienced ASIC designer transitioning to FPGA. ISE enables the designer to start design with any of a number of different source types, including:

- HDL (VHDL, Verilog HDL, ABEL)
- Schematic design files
- State Machines
- IP Cores

The Project Navigator interface provides a visual design for the entire process. The Project Navigator's sources display presents the design's hierarchy after examining design sources, maintains project status, and allows project archiving and snapshots.

All the steps from design entry, simulation, synthesis, and implementation can be carried successfully for the target FPGA device. From the Project Navigator's processes display, we can also automate the front-to-back implementation process using a graphical display and "pull model" showing all the steps in the process.

Fig. 7.8 shows the navigator window in which the VHDL code was written and synthesized, and implemented onto the FPGA. Various successful stages as described in section (5.5), is illustrated in the figure.



```
Target Speed   : -5
Mapper Version : spartan2 -- $Revision: 1.16.8.2 $
Mapped Date    : Mon Jun 27 11:42:03 2005

Design Summary
--------------
Number of errors:      0
Number of warnings:    1
Logic Utilization:
  Number of 4 input LUTs:          32 out of    384    8%
Logic Distribution:
    Number of occupied Slices:                          16 out of    192    8%
    Number of Slices containing only related logic:     16 out of     16  100%
    Number of Slices containing unrelated logic:         0 out of     16    0%
       *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs:        32 out of    384    8%
   Number of bonded IOBs:            65 out of     86   75%
      IOB Latches:                                  32
   Number of GCLKs:                   1 out of      4   25%
   Number of GCLKIOBs:                1 out of      4   25%

Total equivalent gate count for design:  352
Additional JTAG gate count for IOBs:  3,168
Peak Memory Usage:  49 MB

NOTES:

   Related logic is defined as being logic that shares connectivity -
   e.g. two LUTs are "related" if they share common inputs.
```

```
Mapping Module counter . . .
MAP command line:
map -intstyle ise -p xc2s15-cs144-5 -cm area -pr b -k 4 -c 100 -tx off -o counter_map.ncd counter.ngd counter.pcf
Mapping Module counter: DONE
```

Fig. 7.8 Project Navigator Window showing the various successful stages of FPGA Design

Fig. 7.9 illustrates the simulation carried out in the Modelsim version 5.5. Test bench was written in the Modelsim for simulating the logic. The input was pulse of 100ns duration. The counter was found to be counting the number of pulses correctly, and was correspondingly verified.

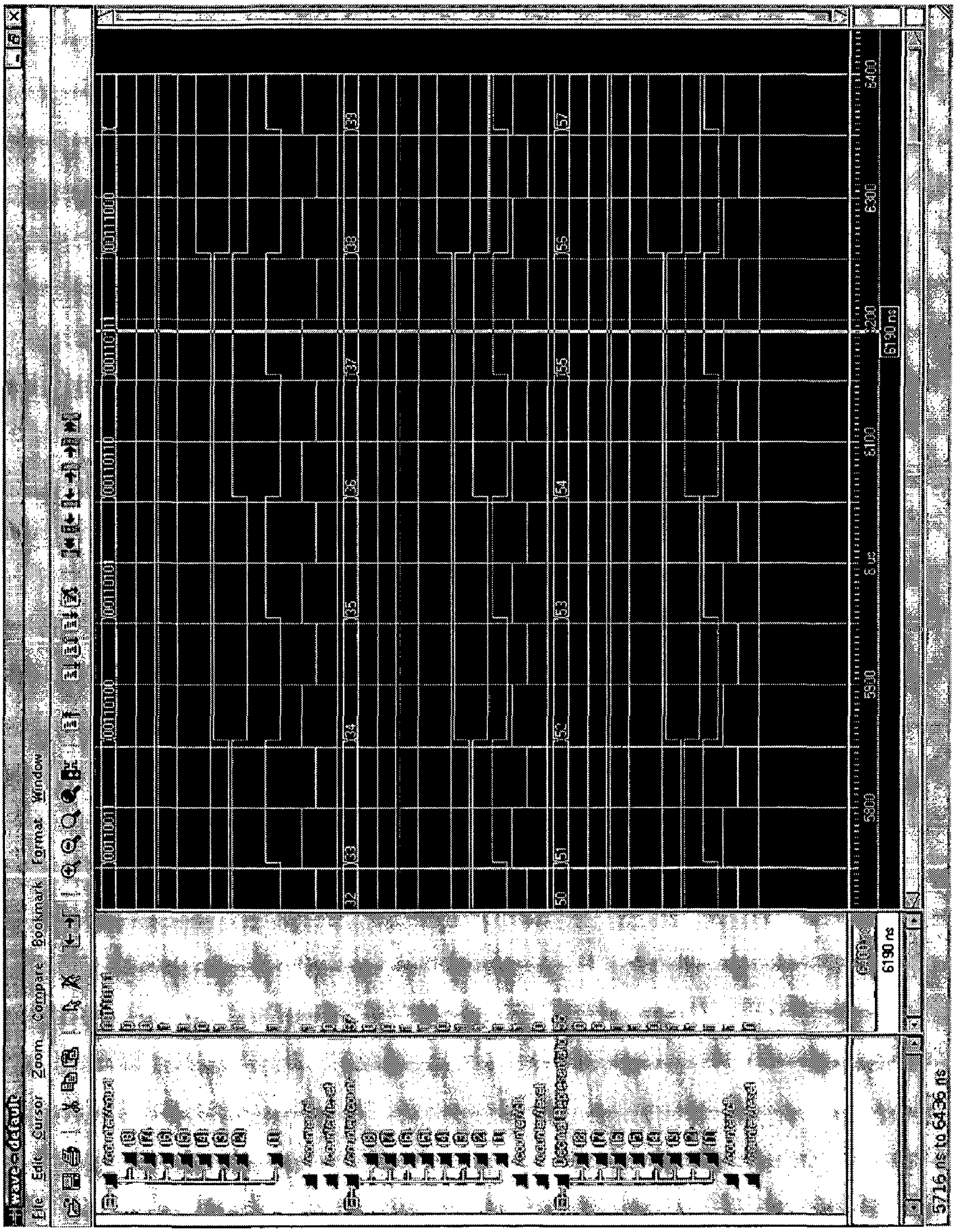


Fig. 7.9 Modelsim Simulation for the Data Acquisition on FPGA

Chapter 8

CONCLUSIONS & SCOPE FOR FUTURE WORK

8.1 Conclusions

The main contribution in this research work consists in the development of methodology, algorithm and a computer program for discharge calculation from the readings of propeller current meters. The methodology conforms to ISO-748 and IS-1192 standards, using the velocity area method. The uncertainty analysis was also carried out for the computation of discharge.

A MATLAB code has been written for discharge computation using the Graphical method. Three individual codes were written separately for each of curve fitting method, cubic spline technique and power law extrapolation scheme. Also, discharge was computed using other methods like mean-section and mid-section. A comparison was done taking power law method as a reference, with the other methods. The reason for choosing the power law is due to the fact that, in practical applications the velocities profile near the channel bed and the two sides channel follow power law. Also, in our scheme the resolution for depth and width was taken as 1.0 cm. so by this resolution naturally, our computed discharge is very accurate when compared with mean and mid-section method where the normally resolution is greater then 1.0 m.

A Graphical User Interface was made in 'G' program, in LabVIEW software. This GUI was tested for actual data acquisition using PCI-6024E, DAQ Card, for interfacing propeller current meters. The virtual instrument was found out to be working normally, and could be used in actual field conditions. A maximum of eight sensors can be connected to the DAQ card.

But, the number of digital channels in the virtual instrument can be increased to any number, by choosing DAQ card specifications suitably. As for the software this can be done by simply adding the number of channels in the block diagram source code in the virtual instrument.

Lastly, simulation of FPGA data processor was done in Modelsim, and implementation up to the bit file generation was done on the FPGA by XILINX-Web Package 6.3.

8.2 Future Scope

There is more that can be done in this field of Flow-Measurement. Since open-channel flow is highly complex, high degree accuracy in discharge calculation requires rigorous mathematical analysis. A model on the lines of closed conduit model can be worked for the discharge computation in open-channels.

Discharge obtained from propeller current meters can also be validated by other methods of flow measurement like ultrasonic flowmeters, ADCP, electromagnetic flowmeters, etc.

Improved methods of interpolation and integration now available can be used for the computation of open-channel discharge using graphical method more accurately. Instead of the use of polynomials for curve fitting, a combination of exponential, trigonometric and polynomial function can give better curve fitting.

Three different languages have been used for different tasks, MATLAB for discharge computation, LabVIEW for data acquisition from the current meters and VHDL for the FPGA programming. Instead common language could be considered for all these tasks. For example MATLAB version 7 with XILINX System Generator supports FPGA simulation and synthesis in addition to mathematical computations and data acquisition.

Also, a full System On Chip, (SOC) can be made on the FPGA that not only carries out the data acquisition from the current meters, but also the scaling, data manipulation and data analysis.

REFERENCES

1. Albert F. Mikhail, Robert J. Knowlton, "Performance Testing of The ST. Lawrence Power Project Using Current Meters", P.Eng., Ontario Power Generation and, P.E., New York Power Authority, International Group on Hydraulic Efficiency Measurements (IGHEM) July 2000.

2. Michael R. Simpson, Roger Bland, "Methods for Accurate Estimation of Net Discharge in a Tidal Channel", IEEE Journal of Oceanic Engineering, Vol. 25, No. 4, October 2000.

3. R.W. Herschy, "The Uncertainty in a Current Meter Measurement", Elsevier-Flow Measurement and Instrumentation 13 (2002) 281–284.

4. Gilles Proulx, Eric Cloutier, Latif Bouhadji, David Lemon, "Comparison of Discharge Measurement by Current Meter and Acoustic Scintillation Methods at LA-Grande-1", Hydro-Québec, Essais spéciaux de production 1986 Mills Road, Sidney, BC, Canada.

5. Dr. Anton Felder, Dr. Horst Neyer, "Discharge Measurement with OTT Can-Bus-Current-Meter at Hydro Power Station Limmritz".

6. G. J. Rao, R. K. Kondayya, " Calibrating Current Meters for High Velocity Flow Measurement in Large Pipe Sections", Indian Journal of Power & River Valley Development ,II ISSN 0019-5537,Vol. 50,2000, No.-112.

7. Thomas Staubli, HTA Lucerne, "Flow Field Upstream of A Trash Rack Measured With an Acoustic Doppler Probe", IGHEM 2000, July 10-12, 2000, Kempten, Germany.

8. V. Kercan, V. Djelic, T. Rus, V. Vujanic, "Experience with Kaplan Turbine Efficiency Measurements-Current Meters and /or Index Test Flow Measurement", IGHEM Seminar-Montreal 1996.

9. Nicolas Caron, J-M Levesque, "Current-Meter Method and Profile Exploration at Brislay Power Plant", IGHEM Montreal, 1996.

10. Tomaz Rus, Vesko Djelic, "Absolute Flow Measurement on HPP-OZBALT using 320 Current-Meters Simultaneously", IGHEM 2000.

11. David D. Lemon, Nicolas Caron, Ward W. Cartier, Gilles Proulx, "Comparison of Turbine Discharge Measured by Current Meters and Acoustic Scintillation Flow Meter At Laforge-2 Power Plant" ASL Environmental Sciences Inc., Sidney, IGHEM1998.

12.Gianberto Grego, "Comparative Flow Measurements at The Caneva Generating Plant Unit 2", ENEL S.p.a CRIS UGI Corso del Popolo,245, IGHEM-Monteal,1996.

13. K. Nurdan, T.C Onka-Nurdan, H.J. Besch, B. Freisleben, N.A. Pavelc, A.H. Walenta, "FPGA-based Data Acquisition System for a Compton Camera", Elsevier Nuclear Instruments and Methods in Physics Research A 510 (2003) 122–125.

14. "Liquid Flow Measurement in open-channels-velocity area methods-Collection and processing and processing of data for determination of errors in measurement", Second edition - 1985-01-15, ISO-1088.

15. Liquid flow measurement in open channels -Rotating element current-meters, Third edition1988-06-01, ISO 2537: 1988 (E).

16. "Liquid flow measurement in open channels – Calibration of rotating-element current-meters in straight open tanks", Ref. ISO-3354, 1976-E.

17. "Hydrometric determinations – Vocabulary and symbols", Fourth edition, 1996, IS0 772:1996(E/F).

18. "Measurement of liquid flow in open channels - Velocity-area methods", Third edition, 1997-08-01, ISO-748.

19. "Velocity Area Methods for Measurement of Flow of Water in Open-Channels, Ist Revision, IS-1192-1981, UDC 532-543:532-573.

20. "Liquid Flow Measurement In Open-Channels-Velocity-Area Methods-Collection and Processing of Data For Determining of Errors In Measurement", IS-14573:1998, ICS 17.120.20, BIS 1998.

21. Edoardo Carminati and A. Ferrero, "A Virtual Instrument for the measurement of the Characteristics of Magnetic Materials ", IEEE Transaction on Instrumentation and Measurements, Vol. 41, No. 6, December 1992.

22. Hayles, Tim, "Developing Networked Data Acquisition Systems with NI-DAQ", National Instruments, http://www. natinst.com/pdf/instrupt/appnotes/an116.pdf.

23. Alessandro Ferrero, "Software for Personal Instruments", IEEE Transactions on Instrumentation and Measurements, Vol.39, No. 6, December 1990.

24. http://www.xilinx.com

25. http://www.usgs.com

26. http://www.ni.com

27.Albert D. Helfrick, William D. Copper, "Modern Electronic Instrumentation and Measurement Techniques', PHI-Pvt. Limited, IInd Edition, 1994.

28. Data Acquisition & Control Handbook, Keithley Instruments, http://www.Keithley.com.

29. User guide, "BNC-2120 Connector Accessory for E Series Devices", National Instruments, July, 1999.

SPECIFICATION OF DAQ CARD, (PCI-6024E)

Analog Input Characteristics

Number of channels	16 single-ended or 8 differentials
(software-selectable per channel)	
Type of ADC	Successive approximation
Resolution	12 bits, 1 in 4,096
Sampling rate	200 kS/s guaranteed
Input signal ranges	Bipolar only
Input coupling	DC

Amplifier Characteristics

Normal powered on	100 G. in parallel with 100 pF
Powered off	4 k. min
Overload	4 k. min
Input bias current	±200 pA
Input offset current	±100 pA
CMRR (DC to 60 Hz)	
Gain 0.5, 1.0	85 dB
Gain 10, 100	90 dB

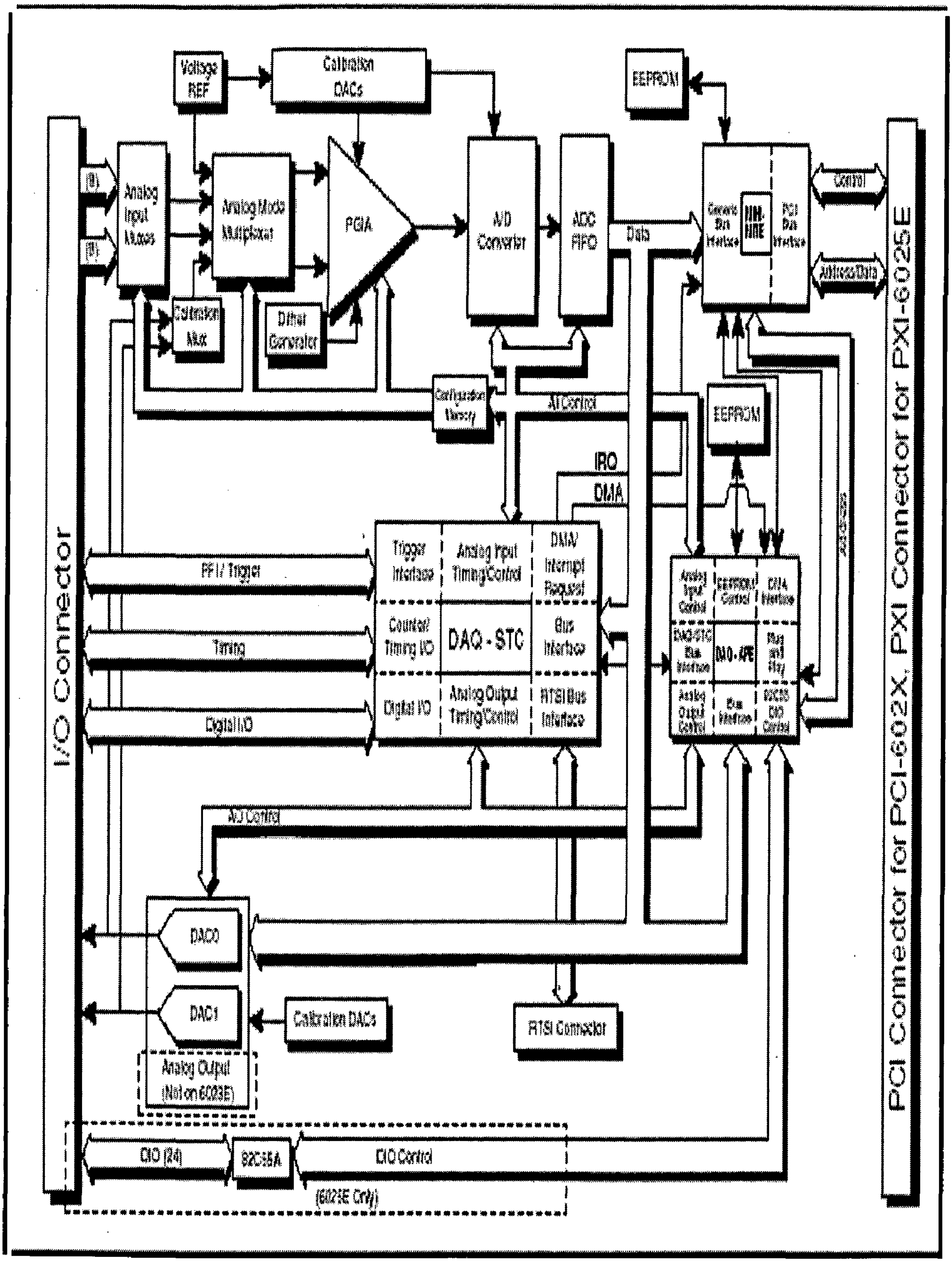
Analog Output Characteristics

Number of channels	2 voltages
Resolution	12 bits, 1 in 4,096
Max update rate	
DMA	10 kHz, system dependent
Interrupts	1 kHz, system dependent
Type of DAC	Double buffered, multiplying

Voltage Output

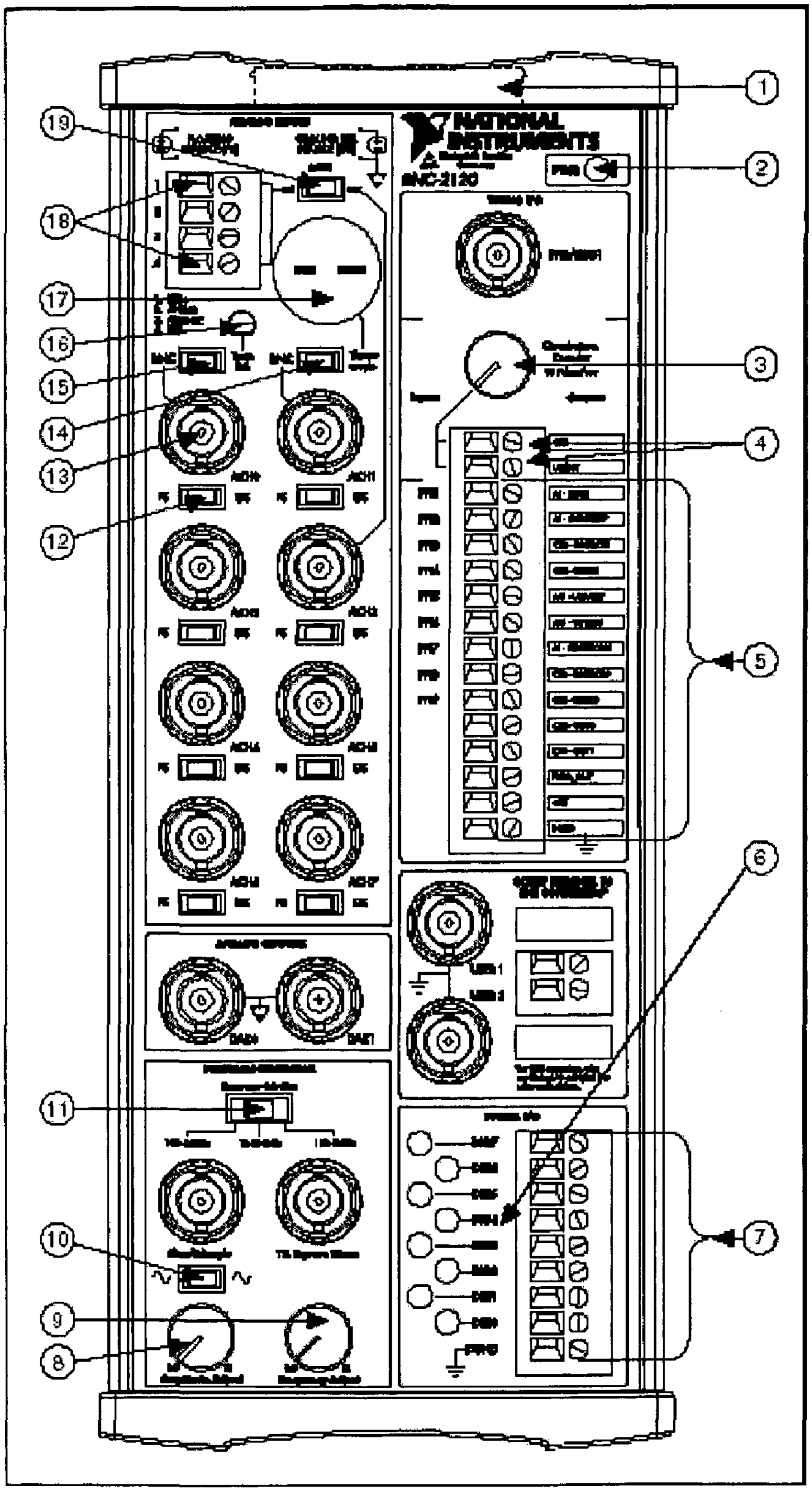
Range	± 10 V
Output coupling	DC
Output impedance	0.1 max
Current drive	±5 mA max
Protection	Short-circuit to ground

Figure shows the internal architecture of PCI-6024E DAQ Card



Appendix-B

BNC-CONNECTOR FOR E SERIES DAQ DEVICE



1 68-Position I/O Connector
2 Power LED Indicator
3 Quadrature Encoder Knob
4 Quadrature Encoder Output Screw Terminals
5 TIO Screw Terminals
6 DIO LED State Indicators
7 DIO Screw Terminals
8 Function Generator Amplitude Adjust Knob
9 Function Generator Frequency Adjust Knob
10 Sine/Triangle Waveform Selection Switch
11 Frequency Range Selection Switch
12 Floating Source (FS)/ Ground-Referenced Source (GS) Switch
13 BNC Connector
14 BNC/Thermocouple Selection Switch
15 BNC/Temperature Reference Selection Switch
16 Temperature Reference
17 Thermocouple Input Connector
18 Resistor Measurement Screw Terminals
19 Resistor/BNC Selection Switch

BNC-2120 Connector

Analog Input

Number of channels (default) ....Eight differential
Field connections (default) ....Eight BNC connectors
Protection...No additional protection provided.

Optional connections
Thermocouple ....Uncompensated miniature
connector, mates with 2-prong
miniature or sub-miniature
connector
Resistor ....   Two screw terminals
Resistor measurement range .... 100 . to 1 M.
Resistor measurement error .... =5%
Screw terminals...................................... Four positions, no larger than
24 AWG wire
Switches .... Eight for selecting floating source
or grounded source inputs
One for selecting BNC or
temperature reference IC
One for selecting BNC or
thermocouple connector
One for selecting BNC or resistor
screw terminals

Analog Output

Field connection..................................... Two BNC connectors

Digital Input/Output

Screw terminals......................................Nine positions, no larger than
24 AWG wire
LED state indicators............................... Eight, one per DIO line
Protection (DC max V)
Powered off....................................   ±5.5 V
Powered on ....   +10/–5 V
Drive
Vol ....   0.6 V, 8 mA
1.6 V, 24 mA