1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

TABLE OF CONTENTS (continued)

	<u>PAGE</u>
4.5 Summary	91
CHAPTER 5 DESIGN VERIFICATION AND CHARACTERIZATION	
5.1 Introduction	96
5.2 Standard Cell Test Vehicle	96
5.3 Integrated Controller Unit	99
5.4 Design Characterization	104
5.5 Summary	108
CHAPTER 6 EVALUATION OF DESIGN ENVIRONMENT	
6.1 Introduction	109
6.2 Design Verification Results	109
6.3 Future Work	111
6.4 Summary	115
REFERENCES	116
APPENDIX I Notes on SPICE and Mercury (H ₃)	118
APPENDIX II Design Schematics and Simulations for the Integrated Controller Unit	130
APPENDIX III Physical Description of Fabricated and Tested I.C.s	172

DEVELOPMENT OF A CMOS VLSI DESIGN ENVIRONMENT AND THE
IMPLEMENTATION OF A MULTI-MODE DATA ACQUISITION CONTROL UNIT

by

Richard E. Vallee, B.Sc.

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Engineering of Carleton University.

Department of Electronics

Faculty of Engineering

Carleton University

Ottawa, Canada

August, 1985

The undersigned recommend to the faculty of Graduate Studies and Research the acceptance of the thesis:

"Development of a CMOS VLSI design environment and the implementation of a multi-mode data acquisition unit"

submitted by Richard E. Vallee in partial fulfillment of the requirements for the degree of Master of Engineering.

Thesis Supervisor

Chairman, Dept. of Electronics

Carleton University

August 1985

ABSTRACT

In order for universities to do research in VLSI design, several resources must be made available. Universities will require a design strategy, design methodology, computer aided design (CAD) tools with I.C. test capabilities, and access to a fabrication center. The required design aids and fabrication facilities have been made available through Phase I of the Canadian Microelectronics Corporation (CMC). It is the goal of this thesis to present a design strategy and methodology which will suit the resources available at Carleton University. The newly developed design environment is then tested through the implementation of a CMOS multi-mode data acquisition control unit. The results of this thesis clearly indicate the validity of the design environment, and its ability to generate research oriented projects on silicon with a high probability of first time correct designs. Given such a design environment it will now be possible for researchers at Carleton University to integrate large systems and play an active role in VLSI design.

ACKNOWLEDGEMENTS

The author wishes to sincerely thank his program supervisor, Professor C. H. Chan for his professional guidance, as well as his financial and moral support during the course of this thesis.

The author would also like to thank Ralph D. Mason for his invaluable assistance in the development of the standard cell library and manual, as well as the multi-mode data acquisition control unit. Sincere thanks are also extended to Carlos R. Pon for his assistance and contribution to the multi-mode data acquisition unit. Special thanks also go to my fellow colleagues, and the entire staff of the Department of Electronics at Carleton University. I would also like to thank Mike Gagnon for his assistance in testing the I.Cs and plotting of data.

Thanks are also extended to both the Canadian Microelectronics Corporation (CMC) and Northern Telecom Ltd. without which this work could not have been done, as well as to NSERC for there funding through the VLSI Mobile Radio Strategic Grant.

Last but not least, I would like to thank my wife Anne, for her support and assistance during the course of this study.

TABLE OF CONTENTS

	<u>PAGE</u>
CHAPTER 1 INTRODUCTION	
1.1 Thesis Motivation and Objectives	1
1.2 Thesis Format	2
CHAPTER 2 DESIGN ENVIRONMENT	
2.1 Introduction	4
2.2 Design Strategy	4
2.3 Design Methodology	7
2.4 CAD and Test Tools	11
2.5 Standard Cell Library	18
2.6 Summary	29
CHAPTER 3 PILOT PROJECT	
3.1 Introduction	37
3.2 Pilot Project Overview	37
3.3 Device Requirements	45
3.4 System Overview	49
3.5 Summary	52
CHAPTER 4 INTEGRATED CONTROLLER UNIT	
4.1 Introduction	53
4.2 Functional Block Diagram	53
4.3 Functional Block Design	63
4.4 Chip Floor Plan and Layout	37

TABLE OF CONTENTS (continued)

	<u>PAGE</u>
4.5 Summary	91
CHAPTER 5 DESIGN VERIFICATION AND CHARACTERIZATION	
5.1 Introduction	96
5.2 Standard Cell Test Vehicle	96
5.3 Integrated Controller Unit	99
5.4 Design Characterization	104
5.5 Summary	108
CHAPTER 6 EVALUATION OF DESIGN ENVIRONMENT	
6.1 Introduction	109
6.2 Design Verification Results	109
6.3 Future Work	111
6.4 Summary	115
REFERENCES	116
APPENDIX I Notes on SPICE and Mercury (Hg)	118
APPENDIX II Design Schematics and Simulations for the Integrated Controller Unit	130
APPENDIX III Physical Description of Fabricated and Tested I.C.s	172

LIST OF FIGURES

<u>FIGURE NO.</u>		<u>PAGE</u>
2.1	Carleton University design methodology	8
2.2	Metheus lambda 750 design environment	14
2.3	General standard cell structure	20
2.4	Standard cell inverta (from standard cell manual)	22
2.5	Standard cell inverta (HILO and transistor models)	24
2.6	Standard cell inverta SPICE simulation	25
2.7	Standard cell inverta Mercury (Hg) simulation	26
2.8	Standard cell inverta layout	27
2.9	Standard cell XOR (from standard cell manual)	28
2.10	Logic representation and truth table for XOR	30
2.11(a)	Standard cell XOR HILO model	31
2.11(b)	Standard cell XOR transistor schematic	32
2.12	Standard cell XOR SPICE simulations	33
2.13	Standard cell XOR Mercury (Hg) simulation	34
2.14	Standard cell XOR layout	35
3.1	General pilot project overview	38
3.2	General system architecture of waveform recorder	40
3.3(a)	Original input signal	42
3.3(b)	Sampling pulses	42
3.3(c)	Sampled signal	42
3.4	Sequential sampling of a waveform	44
3.5	Random sampling of a waveform	46

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
3.6	General overview of pilot project device requirements	48
3.7	General architecture of the multi-mode data acquisition control unit	50
4.1	Initial functional block diagram of integrated controller unit	54
4.2	Functional block diagram of the dedicated processor chip	56
4.3	Functional block diagram of the address generator chip	58
4.4	Real time sampling instruction flow chart	60
4.5	Timing waveform for real time sampling	62
4.6	Sequential sampling instruction flow chart	63
4.7	Timing waveform for sequential sampling	65
4.8	Random sampling instruction flow chart	66-67
4.9	Timing waveform for random sampling	69
4.10	System mode logic schematic for system interface	70
4.11	H ₃ simulation of system mode logic	72
4.12	Initial loading clocks schematic for system interface	73
4.13	H ₃ simulation of initial loading clocks	75
4.14	Programmable timers control logic schematic	76
4.15	H ₃ simulation of programmable timers control logic	79
4.16	Address counter bit slice schematic	80
4.17	Increment logic in address counter bit slice	81
4.18	H ₃ simulation of AD0-AD11 counter generating a sequential address	83
4.19	MSB address counter clock control logic	84

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
4.20	Hg simulation of MSB address counter clock control logic	85
4.21	Clock generation logic for the ADO-AD11 counter and pseudo random number generator	86
4.22	Hg simulation of clock generation logic for ADO-AD11 and random number generator	88
4.23	Floor plan for dedicated processor chip	90
4.24	Dedicated processor chip layout	92
4.25	Floor plan for address generator chip	93
4.26	Address generator chip layout	94
5.1	Sampled data results of masterJK divide by 2 at 40MHz	101
5.2	Oscilloscope traces of masterJK circuit at 20MHz	102
5.3	Sampled data results of the system mode logic	103
5.4	Sampled data results of the ADO-AD11 counter	105
AI.1	Characteristic voltage divider and the Thevenin circuit equivalent for a node, used by H ₃	128
AII.1	System mode logic schematic	132
AII.2	H ₃ simulation of system mode logic	133
AII.3	Variable logic clocks (Initial counter clocks) schematic	134
AII.4	H ₃ simulation of variable logic clocks	135
AII.5	Delay (N) counter control logic schematic	136
AII.6	H ₃ simulation of N counter control logic	137
AII.7	SAM counter control logic schematic	138
AII.8	H ₃ simulation of SAM counter control logic	139
AII.9	Clock generator circuit (masterJK divide by 2) schematic	140

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
AII.10	H ₃ simulation of clock generator circuit	141
AII.11	Delay (N) counter schematic	142
AII.12	H ₃ simulation of N counter (high order bits)	143
AII.13	H ₃ simulation of N counter (low order bits)	144
AII.14	RAM counter schematic	145
AII.15	H ₃ simulation of RAM counter (high order bits)	146
AII.16	H ₃ simulation of RAM counter (low order bits)	147
AII.17	S/H (SAM) counter schematic	148
AII.18	H ₃ simulation of SAM counter	149
AII.19	CHAN counter schematic	150
AII.20	H ₃ simulation of CHAN counter	151
AII.21	HMST counter schematic	152
AII.22	H ₃ simulation of HMST counter	153
AII.23	ADC counter schematic	154
AII.24	H ₃ simulation of ADC counter	155
AII.25	Address counter control logic schematic	157
AII.26	H ₃ simulation of address counter control logic	158
AII.27	MSB address and CHAN counter clock logic (address roll-over) schematic	159
AII.28	H ₃ simulation of address roll-over clock logic	160
AII.29	Input/Output (I/O) logic schematic (6 bits)	161
AII.30	H ₃ simulation of the I/O logic (6 bits)	162
AII.31	Address/Data (A/D) multiplexer schematic	163
AII.32	H ₃ simulation of A/D multiplexer	164

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
AII.33	ADO-AD11 (address/data) counter schematic	165
AII.34	H ₃ simulation of ADO-AD11 counter in sequential sequence mode	166
AII.35	H ₃ simulation of ADO-AD11 counter in random sequence mode	167
AII.36	Pseudo random sequence generator schematic	168
AII.37	H ₃ simulation of pseudo random sequence generator	169
AII.38	MSB address (A12-A15) counter schematic	170
AII.39	H ₃ simulation of MSB address counter	171
AIII.1	CMC multiproject bonding diagram for standard cell test vehicle	173
AIII.2	CMC multiproject bonding diagram for the dedicated processor chip	180
AIII.3	CMC multiproject bonding diagram for the address generator chip	183

LIST OF TABLES

<u>TABLE NO.</u>		<u>PAGE</u>
Table 5.1	Design characterization data from masterJK divide by 2 circuit	106

LIST OF SYMBOLS

A	Amperes
ac	Alternate Current
AD	Area of Drain
ADC	Analog to Digital Converter
ADDCLK	Address Counter Clock
AS	Area of Source
ATG	Automatic Test Generation
CAD	Computer Aided Design
Cin	Carry In
Cout	Carry Out
CIF	Caltech Intermediate Form
CK	Clock
CMC	Canadain Microelectronics Corporation
CMOS	Complementary Metal Oxide Semiconductor
CO	Counter Data Bit Zero
COMPARE	Metheus Software Package
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CSDA	Canadain Semiconductor Design Association
DAC	Digital to Analog Converter
dc	Direct Current
DFF	D flip-flop
DRC	Design Rule Checker
END	End of sampling signal
EOC	End of Conversion

LIST OF SYMBOLS (continued)

ERC	Electric Rule Checker
fc	Nyquist Frequency
GND	Ground
Hg	Mercury, Simulation Package for Metheus
HMST	Hold Mode Settling Time
Hz	Hertz
I.C.	Integrated Circuit
I/O	Input/Output
L	Transistor Channel Length
LCLK	Latch Clock
LEO	Metheus software package for Design Rule Checking
LVE	Load Variable Enable
MCLK	Master Clock
MSB	Most Significant Bit
MUX	Multiplexer
n	n-channel Transistor
NAND	Logic Function
NCC	N-Counter clock
NMOS	N-Channel Metal Oxide Semiconductor
NMOST	N-Channel Metal Oxide Semiconductor Transistor
NOR	Logic Function
NRZ	Non-Return-to-Zero
ns	nanosecond
nstart	Not Start Pulse
OCM	Ontario Center for Microelectronics

LIST OF SYMBOLS (continued)

OR	Logic Function
p	p-Channel Transistor
pf	picofarad
PG	Pattern Generator
PHLED	Physical Layout Editor (Metheus software package)
PHLEX	Physical Layout Extractor (Metheus software package)
PMOST	P-Channel Metal Oxide Semiconductor Transistor
ps	picoseconds
RAM	Random Access Memory
RAMQCBAR	RAM Pulse Bar Signal
RANCLK	Random Generator Clock
RNL	Switch Level Simulator Package developed at MIT
ROM	Read Only Memory
RSE	Random Sampling Enable
rstbar	Reset Bar
RTE	Real Time Sampling Enable
SCHEDIT	Schematic Editor (Metheus software package)
SCH2IMF	Schematic to Intermediate Form (Metheus software package)
SCH2HILO	Schematic to HILO (Metheus software package)
SCH2SIM	Schematic to Simulation (Metheus software package)
S/H	Sample and Hold
SPICE	Simulation Program, Integrated Circuit Emphasis
SPLICE	Simulation Program, Logic, Integrated Circuit Emphasis
SSE	Sequential Sampling Enable
SYMED	Symbolic Editor (Metheus software package)

LIST OF SYMBOLS (continued)

T_a	Aperture Time
T_d	Time Delay
nT_d	An Integer (n) of Time Delays
TAR	UNIX Command for Magnetic Tape Generation
μp	Microprocessor
v	volts
V _{dd}	positive supply voltage
VLSI	Very Large Scale Integration
V _{ss}	Ground or negative supply voltage
W	Transistor Gate Width
XOR	Logic Function

Chapter 1

Introduction

1.1 Thesis Motivation and Objectives

Until recently the role played by universities in VLSI design was limited. Due to the high cost of fabrication and CAD tools this would have remained so if it was not for the contribution made by the Canadian Microelectronics Corporation (CMC). CMC provides eligible universities with Northern Telecom Ltd.'s CMOS1B 5um double poly, single metal process, as well as advanced CAD and Test facilities such as the Metheus lambda 750 workstation and HP 8000 series test station.

Integrated circuits are now fully designed within the respective universities, which then send the Caltech Intermediate Form (CIF) files to the VLSI implementation center at Queen's University, Kingston Ontario. There the designs are combined to form multiproject chips following the Mead and Conway [1] approach. The designs are then sent to Northern Telecom Ltd. for mask making and fabrication. Upon completion the chips are bonded and a number of prototypes are then sent to the respective designers for testing. The complete turnaround time has now been reduced to between three and four months.

The purpose of this thesis is to develop a VLSI design environment at Carleton University using the resources made available through CMC. An integrated circuit design methodology will be

established and then tested by implementing a reasonably complex system on a chip. Once the design environment has been proven, it will be possible to do VLSI designs with a reasonable first attempt success rate. This will demonstrate that universities can play a major role in VLSI design. Due to the success of this work, a brief summary [2] has been prepared and presented at Carleton during the Ottawa-Carleton Research Institute (O.C.R.I.) day, the CMC workshop at Queen's University, and the University/Government/Industry Microelectronics (UGIM) Symposium at Auburn University, Alabama, U.S.A.

1.2 Thesis Format

Chapter 2 will fully discuss the development of a design strategy and methodology at Carleton University. This will include a full description of the resources available as well as their role in the development of a design data base. Chapter 3 will introduce the pilot project used to test the design environment. The full details of the designed project will be given in Chapter 4. Chapter 5 will focus on the testing and characterization of the designs once they have been fabricated. Chapter 6 will draw conclusions as to the success of the design strategy as well as suggest areas of future work in order to improve the available resources.

A Standard Cell Manual has also been prepared and submitted with this thesis since the entire design environment and pilot project revolve around it. The manual will also permit future

students to readily make use of the design environment established through the work of this thesis.

a semi-custom standard cell approach as outlined by A.J. Kessler and A. Ganesan [8] would be undertaken. This option allows for a better utilization of silicon and also for analog or analog and digital designs on a chip. Also because Northern Telecom does not support a gate-array structure, there would be no reduction in cost during mask making.

2.3 Design Methodology

Having decided upon a design strategy, a design methodology which will incorporate both the strategy and CAD tools available will be developed. With these factors in consideration, a design methodology as shown in figure 2.1 is proposed. The steps include;

- Device Requirements
- System Overview and Functional Block Diagram
- Functional Block Design and Simulation, Chip Floor Plan
- Layout and Resimulation
- Fabrication
- Testing

The device requirements are first analyzed in a "black box" type approach. The available inputs and desired outputs are determined as well as any specifications that must be met by these signals. This allows for an initial pin count to determine if any problems arise due to a restriction of 40 pins maximum per chip.

arrays allow short implementation times but do not tend to be area efficient.

Semi-custom design combines the feature of custom design with the possibility of automatic or semi-automatic routing. This requires the creation of custom cells or primitives within a regular grid array. Individual cells can occupy several of the spaces defined by the grid structure. Primitive cells can be digital elements such as inverters, and multiple fan-in NANDs and NORs, as well as analog elements such as opamps. Microcells are slightly more developed cells such as flip-flops and latches that incorporate 2 or more primitive cells. They can be directly composed of primitive cells or designed with fully custom primitives for each microcell function. These primitive cells and microcells can be joined together to form more complex macro cells or functional logic blocks which are then interconnected by allowing sufficient grid spacings between them. The advantages of this are that given a well established and area efficient cell library, the user only has to worry about interconnect, and that large functional blocks or macro cells may be shared between users.

Full custom design involves the tailoring of device and cell dimensions to fit a particular need. In general, a custom design combines some elements of standardization at the macrocell and microcell levels, but very little at the primitive cell level. This means that within specific macro or micro cells, the layout is governed by space and performance goals. The primary goal of

full custom design is area efficiency, not short design times.

Taking into consideration the constraints of available resources at Carleton University, one of these three design options must be selected. First the restrictions imposed by the Metheus lambda 750 Workstation will be examined. It is known that the Metheus can support gate-array libraries [4] from various vendors. The hierarchical properties and ability to develop custom cell libraries that are accessible to all users clearly demonstrates the ability to do semi-custom I.C. design. Full custom can also be supported in this context but because of a lesser degree of hierarchy the I.C.s designed must be smaller. From this it is clear that the CAD tool available does not force any real constraints on the design option.

The restrictions imposed by a university environment will now be examined. Universities are limited in two ways, time and human resources. Time, in the sense that a university semester lasts four months and that a graduate degree requires between two to four years to complete. Human resources are limited due to the fact that most graduates leave the institution upon completion of their degree and thereby leave a lack of highly qualified users from which to draw information. For these reasons a fast and simple to use design option seems desirable. This eliminates the possibility for full custom designs. A gate-array or semi-custom design approach must be used.

After studying several articles on the subject of fast design times [5], and I.C. design strategies [6,7], it was decided that

a semi-custom standard cell approach as outlined by A.J. Kessler and A. Ganagan [8] would be undertaken. This option allows for a better utilization of silicon and also for analog or analog and digital designs on a chip. Also because Northern Telecom does not support a gate-array structure, there would be no reduction in cost during mask making.

2.3 Design Methodology

Having decided upon a design strategy, a design methodology which will incorporate both the strategy and CAD tools available will be developed. With these factors in consideration, a design methodology as shown in figure 2.1 is proposed. The steps include;

- Device Requirements
- System Overview and Functional Block Diagram
- Functional Block Design and Simulation, Chip Floor Plan
- Layout and Resimulation
- Fabrication
- Testing

The device requirements are first analyzed in a "black box" type approach. The available inputs and desired outputs are determined as well as any specifications that must be met by these signals. This allows for an initial pin count to determine if any problems arise due to a restriction of 40 pins maximum per chip.

CHIP DESIGN METHODOLOGY

Figure 2.1 Carleton University design methodology.

pixels and supports 8 million shades of color. The operating system run by the Metheus is Berkley UNIX 4.1. Peripherals include an optical mouse for ease of operation, a Cipher F800 magnetic tape drive, 1 Mbyte floppy disk drive, 300 baud modem, an HP 7580 pen plotter and a lineprinter. The Metheus also has six RS-232C ports to allow links with other terminals and the university main frame (CP6).

The workstation supports all the required software to implement the design methodology proposed, as shown in figure 2.2. The blocks in figure 2.2 represent the software packages on the Metheus, and the circles are the files used or generated by the programs. The three library files are the ones which were generated in order to implement the design methodology, and is the information that is contained in the Standard Cell Manual.

In order to implement a complete standard cell library on the system, a symbol for each primitive cell must be generated using SYMED (Symbolic Editor). These symbols are stored in a library directory with read only access to each user. The user may also generate his own symbols for documentation purposes and save these in his own account. SCHEDIT (Schematic Editor) is then used to call up the desired primitives and wire them together to form a circuit schematic. The individual circuit schematics are stored in the users account. SCHEDIT is also used to generate transistor level schematics for each of the cell symbols in the library. These transistor level schematics are saved in another read only directory which can be accessed by all the users. The

functional block. Once the design has been verified, the area occupied by the circuit may now be estimated by knowing the area taken by each primitive and micro cell layout, and allowing area for interconnection of the various cells. Once this has been completed for every functional block, an overall chip floor plan is assembled using the estimated areas of each block. This will also determine which pad frame may be used. CMOS offers a choice of four pad frames, the largest being A size with 40 pins possible to D size with 18 pins. The design may be restricted either by area or pin count to a specific pad frame. If the area available is not sufficient the designer may have to restructure the device into two separate chips and modify the system overview and some functional blocks accordingly.

Having completed the required design iterations the designer is now ready to proceed to the layout phase of the methodology. The layout is done much like the functional block schematic designs. The primitive cell layouts are called into a layout editor and interconnected using the desired mask layers. To determine if the circuit layout is identical to the circuit schematic a Netlist compare may be performed between the two designs. A Netlist is a circuit description containing active elements (transistors, resistors, capacitors) and their connectivity. If the layout is correct a simulation deck is then extracted from the layout to determine the effects of interconnect capacitance. The timing constraints are then verified, and if the specifications are not met the designer must redesign the functional block at a schematic level to ease the critical.

timing paths. Having obtained a functional layout, design rule checking (DRC) is then performed on the circuit block to ensure that no layout rule violations exist due to interconnect. Once all the circuit block layouts have been completed, they are then interconnected to form the entire chip layout. The I/O pads are called into their appropriate location and the power rails are laid out to supply each block with the required voltage levels. DRC is then performed on the overall chip interconnect channels and signal buses.

Once the design stage is completed, the layout is sent to Queen's University as mentioned earlier. From there it is sent to Northern Telecom Ltd. where it is fabricated.

During the implementation of the system the designer should include test capabilities [9,10,11] which will simplify the verification of the design. Once the chip is received the design is tested and characterized using the HP 8000 series test station supplied by CMC. The design is then fully documented so that it may be referenced by future students. These students may then obtain ideas from past proven designs that will enable them to implement larger and more complex integrated circuits.

2.4 CAD and Test Tools

Now that the design methodology has been presented the attention will be turned to the CAD and Test tools available at Carleton University.

The first design aid examined is the Corvus Concept computer. The Corvus Concept is a M68000 microprocessor based system with 512 Kbytes of internal RAM and a high resolution monochromatic screen. Carleton has roughly 19 such computers networked together and served by a 40 Mbyte hard disk which supports all the required software. As a design tool the Corvus supports both SPICE and SPLICE circuit simulation packages as well as a layout editor developed at Carleton University [12]. The Corvus also supports a link to the University CP6 Honeywell mainframe computer on which large circuit simulations may be carried out. These machines serve a useful purpose in that they can be used to design primitive cells for cell libraries. Once a cell has been developed it can be transferred to the more advanced Metheus workstation where the parent library resides. As an example, the standard cells in the Standard Cell Manual that are supported by the Metheus workstation were originally developed using the Corvus Concept network.

The more advanced I.C. design work is now done using the Metheus lambda 750 workstation. The Metheus uses two M68000 microprocessors as its main CPU. One is used for system control and the other executes the operating system and applications software. The memory consists of 2 Mbyte main memory and two 33 Mbyte winchester disk drives to give a total of about 68 Megabytes. One disk supports the application software and the other serves the users. The graphics workstation uses two processors as well, one to run the display and another the bit-slice graphics generator. The 19 inch monitor has 1024 x 768

pixels and supports 8 million shades of color. The operating system run by the Metheus is Berkley UNIX 4.1. Peripherals include an optical mouse for ease of operation, a Cipher F800 magnetic tape drive, 1 Mbyte floppy disk drive, 300 baud modem, an HP 7580 pen plotter and a lineprinter. The Metheus also has six RS-232C ports to allow links with other terminals and the university main frame (CP6).

The workstation supports all the required software to implement the design methodology proposed, as shown in figure 2.2. The blocks in figure 2.2 represent the software packages on the Metheus, and the circles are the files used or generated by the programs. The three library files are the ones which were generated in order to implement the design methodology, and is the information that is contained in the Standard Cell Manual.

In order to implement a complete standard cell library on the system, a symbol for each primitive cell must be generated using SYMED (Symbolic Editor). These symbols are stored in a library directory with read only access to each user. The user may also generate his own symbols for documentation purposes and save these in his own account. SCHEDIT (Schematic Editor) is then used to call up the desired primitives and wire them together to form a circuit schematic. The individual circuit schematics are stored in the users account. SCHEDIT is also used to generate transistor level schematics for each of the cell symbols in the library. These transistor level schematics are saved in another read only directory which can be accessed by all the users. The

METHEUS lambda 750 DESIGN ENVIRONMENT

Figure 2.2 Metheus lambda 750 design environment.

transistor schematics contain all the necessary information, such as transistor aspect ratios (W and L), and areas of drain and source (A_D and A_S), to allow simulation decks to be constructed from the circuit schematic through SCH2SIM (Schematic to Simulation). Note that SCH2SIM is being used as a generic name to represent the actual programs used, such as SCH2IMF, SCH2SIM and SCH2HILO. When using SCH2SIM the user has a choice of several output formats. These formats include a simulation deck for SPICE, Mercury (Hg), or HILO. SPICE is a general purpose circuit simulator performing nonlinear dc, nonlinear transient and linear ac analyses of the design. More information on SPICE may be found in Appendix I. Mercury (Hg) is a switch level simulator for verifying the operation of combinational logic and sequential circuits. Hg is based on an RNL simulation package developed at MIT. More details on Hg can be found in Appendix I. HILO is a gate level logic simulator that provides fault simulation and can be used for test vector generation or Automatic Test Generation (ATG). Besides a choice of simulation decks the output for SCH2SIM can be directed to a Netlist file where it can be checked using ERC (Electrical Rule Checker). The ERC will check for correct transistor types, incomplete signal paths, too many threshold drops and pullup/pulldown ratio errors. The Netlist file can also be used to do comparison checks between Netlists from schematics and layout. The use of these software packages will ensure the development of the first three stages in the design methodology.

Having completed the function block schematics and simulations the designer can now develop these circuit schematics into layouts. This is done using PHLED (Physical Layout Editor), which calls upon a standard cell layout library. The user calls in the required cells and performs the required interconnect. The functional block layout files created are then saved in the users own workspace. From these layouts the required information needed to build simulation decks for SPICE and Hg can be extracted by using PHLEX (Physical Layout Extractor). By running SPICE and Hg simulations on the layout files, the designer can verify correct functionality as well as determine the effects due to interconnect capacitance. For larger files the designer can run SPICE simulations on CP6. PHLEX may also be used to extract a Netlist file from the layout and this may be compared to the circuit schematic Netlist using COMPARE. ERC may also be performed on the layout Netlist file. Once the functional circuit blocks are operational and meet the required specifications the designer then can use LEO (Design Rule Checker) to determine if the mask layers conform to the design rules. Note that LEO is used while in PHLED.

The designer continues this procedure until all functional blocks have been designed, simulated and checked. Once the designs are finished the user then calls the required pad frame and functional blocks together to complete the layout. The designer then performs DRC on the required interconnect channels. Once the design is error free it is easily converted to CIF and sent on magnetic tape to the VLSI implementation centre at

Queen's University. There the design is merged together with other designs to form a multi-project chip. The projects are converted into a pattern generator (PG) format and sent to Northern Telecom Ltd., as mentioned earlier.

After three to four months the designer will receive several prototypes of the design which will be used for testing. The verification of the designs is done using the HP 8000 series test station made available by CMC. The test station consists of an HP 4145A semiconductor parameter analyzer for process parameter measurements, a Wentworth probing station, an HP 8180A data generator for developing test vectors, and the HP 8182 data analyzer for response measurements. For the purpose of this thesis the HP 8180A and HP 8182A testing facilities will be examined.

The HP 8180A [13] is a modular, high speed (50MHz) programmable data generator for stimulating multi-channel digital circuits. The operating concept is based around a CRT and softkeys. Eight non-return-to-zero (NRZ) channels with data rates from 1Hz to 50MHz and 1 Kbit memory, are used to stimulate the desired inputs of a chip. The results of these test vector inputs are then observed using the HP 8182A.

The HP 8182A [14] is a modular, high speed data analyzer for the analysis of multi-channel digital circuits. Its operation is also based around a CRT and softkeys. It can monitor eight channels with 1024 bits per channel memory and asynchronous or

synchronous rates from 1Hz to over 50MHz. It also allows the sampling point to be delayed with 100ps resolution with respect to the active clock slope. Other features include a real time window compare with 100ps resolution for window width and placement, and real glitch detection and storage. The active probe inputs have an input capacitance of about 7pf in order to minimize loading on the chip outputs. Using these test facilities the designer may now input several test vectors and monitor the chips behaviour in order to determine correct functionality. Should the chip behaviour seem incorrect or faulty, the prober may then be used to probe the desired I/O pads or signal paths with glass openings.

The designer should now write a formal report on the project outlining any difficulties experienced and results of his/her testing in order that future students may use their work as a starting point to building more complex integrated circuit designs.

2.5 Standard Cell Library

This section will briefly describe the development of the standard cell library as well as the detailed design of two primitive cells. As mentioned earlier, the Standard Cell Manual contains a complete list and description of all the standard cells implemented for the design environment.

The standard cells are based on a 16u grid system. 16u was chosen as the grid size since this is the required pitch between

poly to metal contacts as can be seen from the 13 grid lines found in Appendix I of the Standard Cell Manual. For most small cells such as INVERTERS and NORs, the cell width will be determined by the number of I/O contacts to the cell. In general the cell width is found to be the number of I/O contacts plus 1, grid spacings. Hence a 2 input NAND is 4 grid spacings in width and an inverter is 3. The I/O contacts to the cells are found on both the top and bottom to allow easy access for both inputs and outputs. The general cell structure is shown in figure 2.3. The contact functions are symmetric about the cell, hence if the output is in grid location 4 on the top, it is also found in location 4 on the bottom. In this way the cells may be X-axis mirrored even after the interconnect has been laid out. This will also allow easier development of auto-routing software. As can be seen in figure 2.3 both Vdd and Vss (GND!) run horizontally through the cell. The Vdd rail is 9u in width, to allow placement of a contact within it, and is found 48u from the top. The Vss rail is similar but lies 48u from the bottom. 48u was chosen to allow sufficient area to run two metal lines between the rails and the I/O contacts. Beyond the Vss rail an NMOST device well can extend 23u. Beyond the Vdd rail a PMOST device well can extend 48u. This allows the PMOST to have roughly double the aspect ratio of the NMOST, which will give more symmetric rise and fall times. There is then 23u left between the two device wells which is sufficient spacing to allow the placement of a poly to metal contact. This gives a total cell height of 13 grids or 208u which is fixed for almost every cell. The

Figure 2.3 General standard cell structure.

only cells that do not follow these general rules are the input protection and output driver devices. There are more rules governing the placement and overlap of the remaining layers which can be found in the Standard Cell Manual.

The library consists of 29 primitive cells. These include several inverters with varied aspect ratios, multiple fan-in NANDs and NORs, latches and flip-flops with set and/or reset capabilities, two types of pass gates, an XOR, a 2 input mux, and I/O pad devices.

In this section an inverter named Inverta will be examined first. This inverter has a large aspect ratio and can be used to drive signals that exhibit large loading conditions. The block symbol and schematic symbol can be found in figure 2.4 (a) and (b) respectively. The block symbol shows the cell width of 3 grids as well as the I/O locations. The input lies in grid 1 and the output in grid 3. The block symbol area (3×13 grids²) can be used to estimate the area occupied by a circuit block before it is actually laid out. Also the block symbol can assist in the writing of auto-placement and auto-routing software. The schematic symbol is used by the schematic editor as mentioned earlier. The load versus delay chart for the primitive is shown in figure 2.4(c). This chart is obtained from SPICE simulations by varying the load capacitance and calculating the propagation delay. The simulation deck is obtained from the layout with PHLEX. With this chart rough timing estimates can be made during circuit design. The input capacitance is obtained from;

CELL NAME
CELL FILE

INVERTER (A size)
inverta

a) block symbol

b) logic symbol

c) load vs delay

Figure 2.4 Standard cell inverta (from standard cell manual).

$$C_{in} = C_{p-sub}[A_{p-sub}] + C_{ox}[(W \times L)_n + (W \times L)_p]$$

C_{ox} (the gate capacitance/ μ^2) and C_{p-sub} (poly over substrate/ μ^2) are given process parameters and A_{p-sub} (area of poly over substrate), W and L are obtained from the layout. From this,

$$C_{in} = 0.3 \text{ pf.}$$

The HILO circuit model is shown in figure 2.5(a). In this model "ter" is the input terminal capacitance in femtofarads, "mrise" is the slope of the rise time in seconds per farad, and "mfall" is the slope of the fall time in seconds per farad. When HILO is run on a large circuit schematic the program will use this model to emulate the inverter. The transistor schematic is shown in figure 2.5(b). This is the model that the SCH2SIM program draws upon in order to prepare a SPICE or Hg simulation deck for a circuit schematic. The results of both a SPICE and Hg simulation performed on the layout are shown in figures 2.6 and 2.7 respectively. From these results both functionality and timing characteristics can be obtained. The layout of inverter is shown in figure 2.8. This is the file that is stored in the PHLED cell library.

The second cell to be examined is the exclusive OR (XOR). The block and schematic symbols are shown in figure 2.9(a) and (b) respectively. This cell is seven grids in width, with the output in grid 1 and input in grid locations 4 and 6. The load versus delay chart is shown in figure 2.9(c). The data for this chart is obtained from various SPICE simulations done on the layout of the cell. The input capacitance for the various inputs

tr: 300
 trise: 12000
 mfall: 10000

a) HiLo Model

b) Transistor Schematic

Figure 2.5 Standard cell inverta (HILO and transistor models).

Figure 2.6 Standard cell inverter SPICE simulation.

Figure 2:7 Standard cell inverta Mercury (H₂) simulation.

Figure 2.3 Standard cell layout.

CELL NAME:
CELL FILE:

EXCLUSIVE OR
xor

a) block symbol

b) logic symbol

Figure 2.9 Standard cell XOR (from standard cell manual)

is found to be;

$$C_{in}(A) = 0.3\text{pf}$$

$$C_{in}(B) = 0.5\text{pf}$$

The B input has a higher capacitance due to the internal workings of the cell. As can be seen in figure 2.10(a), the XOR function is obtained using an 8 transistor configuration consisting of two inverters, a passgate, and a sensing circuit using a p and n transistor.

An alternate circuit that performs the XOR function is shown in figure 2.10(b). The disadvantage of this circuit is a higher transistor count of 16. By reducing the transistor count to 8, the area required can greatly be reduced. The HILO model is found in figure 2.11(a), and the transistor schematic used for SPICE and Hg simulation decks is shown in figure 2.11(b). The SPICE and Hg simulations results are shown in figures 2.12 and 2.13 respectively. The layout for this primitive is shown in figure 2.14.

2.6 Summary

In this chapter a brief summary of possible design strategies was given. Upon consideration of the constraints that exist within a university based design center, a semi-custom standard cell design strategy was chosen. A design methodology was then engineered in order to simplify the task of integrating complex circuitry onto silicon, using the proposed design strategy. The design methodology was structured so that its needs could be

a) 8 transistor XOR logic function

b) NAND-NOR logic of XOR function

TRUTH TABLE

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

Figure 2.10 Logic representative and truth table for XOR.

ter: 400

mrise: 18000

mfatl: 14000

a) Hilo Model

Figure 2.11(a) Standard cell XOR HILO model.

b) Transistor Schematic

Figure 2.11(b) Standard cell XOR transistor schematic.

Figure 2.12 Standard cell XOR SPICE simulation.

Figure 2.13 Standard cell XOR Mercury (H₃) simulation.

Figure 2.14. Standard cell XOR layout.

fully met by the available CAD and Test tools. The details of these tools as well as the software available to implement the design methodology were then presented. A detailed description of two standard cells was then given. A standard cell test vehical containing over 30 cells was fabricated in order to test and characterize the primitive and micro cells designed. The results of this chip will be presented in chapter 5.

With the available resources (i.e. design strategy, design methodology, and CAD and Test tools) it would appear that complex integrated circuits could now be designed within the required timing constraints. It is now the purpose of both chapters 3 and 4 to demonstrate the viability of the design environment established, through the implementation of a multi-mode data acquisition control unit.

Chapter 3

Pilot Project

3.1 Introduction

The previous chapter has fully described the design environment that has been implemented at Carleton University. In order to determine its effectiveness a Pilot Project was integrated using the design methodology proposed. The first section gives an overview of the project that has been chosen. From this overview, the device requirements will be determined. Following the device requirements the general integrated systems overview of the project will be discussed. The functional block diagram, as well as the other stages in the design methodology will be discussed in chapter 4.

3.2 Pilot Project Overview

The pilot project proposed is the integration of a microprocessor-controlled digital waveform recorder, implemented at Carleton University by Carlos R. Pon. [15] The project was fully designed and characterized as requirements for a Master's degree. The results of this project were very positive and made it a good candidate for integration.

The design was realized using off the shelf I.C.s and evolved around an M6800 microprocessor, as shown in figure 3.1. The M6800up had two major tasks. One was a system controller for the acquisition unit, and the second was to act as an interface to a

Figure 3.1 General pilot project overview [15].

more powerful post-processor. In order to carry out these tasks, software was developed in M6800 assembler code to allow the following operations;

1) Different Acquisition Modes

- Real Time
- Sequential (non-real time)
- Random (non-real time)

2) Select the real time sampling rate (50Hz-500KHz)

3) Select Pre-trigger or Post-trigger triggering mode for real time sampling.

4) Choice of averaging for repetitive signals.

5) Display the stored waveform on an analog scope.

6) Transfer stored data through an RS-232C serial port to a host computer.

The acquisition unit consisted of peripheral hardware for the M6800 CPU as well as a fast sample and hold (S/H), 12 bit resolution analog to digital converter (ADC) and a digital to analog converter (DAC). The general system architecture of the waveform recorder is shown in figure 3.2. The digital data from the sampled waveform was stored in a buffer memory of 2K x 12 bits of RAM. Once the buffer memory was filled, sampling would stop and the stored data would then be transferred through a parallel port to a larger bank of microprocessor controlled memory. If desired, the data could also be clocked into a DAC for recon-

Figure 3.2 General system architecture of waveform recorder [15]

struction of the analog waveform for display purposes.

The front end amplifier and trigger generator, seen in figure 3.2, were not included in the design. The reason for this is that the performance of digital acquisition units are limited by the digitizer (ADC) and memory, not the front end amplifier or triggering circuitry. In order to clearly define the specifications on the device requirements for this project, more detailed information on the acquisition modes is needed.

There are three schemes generally used by digital sampling oscilloscopes. They are;

- 1) Real Time Sampling
- 2) Sequential Sampling (non-real time)
- 3) Random Sampling (non-real time)

Real time sampling is the first scheme examined. During this most common technique, an analog signal is sampled on a periodic basis as shown in figure 3.3(a,b). The analog signal is first sampled for a short period of time, after which it is held until a new sample is taken. The result is a modulated waveform as shown in figure 3.3(c). If a signal is bandwidth limited and contains no frequency components higher than f_c , then the original signal can be fully recovered after sampling, if sampled at a rate of at least $2f_c$. Frequency f_c is typically called the Nyquist frequency. Any sampling rate less than $2f_c$ will result in aliasing. A typical digital storage oscilloscope using real

Figure 3.3(a) Original input signal [15].

Figure 3.3(b) Sampling pulses [15].

Figure 3.3(c) Sampled signal [15].

time sampling will digitize a purely analog signal as quickly as possible, and store the data in memory. Once the memory is filled, or the required data taken, the data are either processed for averaging or statistical information, or transferred to a display unit. The performance of such a system is limited by the digitization. If the sampling rate is increased, this will cause a loss in the resolution of bits, and hence introduce error in the sampled waveform. The trend in these systems is for faster and more precise ADC and/or digitizing techniques. During real time sampling the operator has a choice of post- or pre-triggering. Post-triggering is when sampling is initiated by an external trigger pulse and terminated when the desired data has been taken. Pre-triggering is when sampling is immediately initiated and terminated by an external trigger pulse.

Sequential sampling is a non-real time sampling technique. It can be used to digitize waveforms well above the usual Nyquist frequency of the system, provided such waveforms are repetitive. It consists of taking samples of the repetitive waveform in small incremental steps from a given reference point, as shown in figure 3.4. The advantage of this technique is that the S/H is now the bandwidth limiting component and not the ADC as in real time sampling. More precisely, the bandwidth is dependent on the aperture time, T_A of the S/H and in the delay time T_d . In implementing this technique, a trigger signal as well as the waveform are required as inputs. The trigger signal is used to reference the start of the periodic waveform. At some instant after the

Figure 3.4 Sequential sampling of a waveform [15].

trigger pulse, say nT_d , the waveform is sampled and then digitized. The data is then stored in a memory location proportional to the sampled delay time, nT_d , where n is an integer. The data is then stored and the delay (nT_d) is incremented $[(n + 1)T_d]$, as well as the memory address, and the cycle starts over. Once the memory has been sequentially filled, the sampling is halted and the data is transferred to a host CPU for processing or to a DAC for display purposes.

A relatively new technique known as random sampling may also be used for data acquisition of periodic waveforms. This method of sampling is very similar to that of sequential sampling. The difference lies in the sampling delay time nT_d . During sequential sampling nT_d is increased sequentially but for random sampling, nT_d is an arbitrary number as shown in figure 3.5. The digitized data is again stored in a memory location proportional to nT_d . This allows the data to be stored sequentially in terms of sampling delay, but taken in a random order. The random number used is generated from a pseudo random sequence which can be implemented in software or hardware. The M6800up system uses a table of pseudo random numbers generated from software. Once all the memory locations have been filled, the data can then be transferred for processing or display and appear in sequential order. The advantage of this technique is improved resolution over higher frequencies of periodic waveforms.

3.3 Device Requirements

The first objectives of the design strategy are to clearly

Figure 3.5 Random sampling of a waveform [15].

define the device requirements of the integrated system. A general overview of the device requirements is shown in figure 3.6. The project is divided into two main parts, the integrated system controller and the memory controller. The implementation of the integrated systems controller was performed by the author and will be presented here. The task of implementing the memory controller was given to Ralph D. Mason [16], as part of his Masters thesis.

The new data acquisition unit has a larger buffer memory of $64K \times 12$ bits. This has been divided up into 16 channels of $4K \times 12$ bits each. The user will now have the option of choosing which channel to store data in as well as the number of channels to be filled. The memory controller will allow display of any channel as well as data transfer to a host CPU. The user interface can be anything from a control panel to a host CPU. A host CPU could in essence orchestrate several of these data acquisition units in parallel, allowing the freedom to implement a very powerful data acquisition system.

Having clearly defined the system to be implemented, the attention is now focused on the requirements of the integrated system controller. The functions required of the controller unit are as follows:

i) choice of three sampling modes

- real time
- sequential

General Overview of the Device Requirements

Figure 3.5 General overview of pilot project device requirements.

- random

- ii) choice of post-triggering of pre-triggering during real time sampling.
- iii) compatability to most S/Hs and ADCs.
- iv) choice of the number of data channels to be taken, from 1 to 16.
- v) choice of starting data channel.

The inputs available to the system controller are a 50MHz system clock, a referenced trigger pulse, pre-trigger and post-trigger select signals and an end of conversion (EOC) signal from the ADC. Other required inputs will be control signals from a user interface as well as data lines with which to program the controller. The required outputs will be a S/H pulse, an ADC start pulse and a write enable or RAM pulse. These pulses will require programmable lengths in order to make the controller compatible with most S/Hs and ADCs. Other outputs needed are a 16 bit ($2^{16}=64K$) address bus and a system's status signal to allow interface capabilities. Since the controller unit will be a purely digital system, the required voltage inputs are Vdd, typically 5 volts and Vss or GND.

3.4 System Overview

The next phase in the design strategy is to propose the signal processing required in order to obtain the desired outputs from the available inputs. Figure 3.7 shows a bus oriented architecture that will accomplish the desired results. A bank of programmable timers can be used to generate the required control

Figure 3.7 General architecture of the multi-mode data acquisition control unit.

pulses for the S/H, ADC, and memory functions. The END signal can be generated by a programmable counter which counts down the number of data channels taken. The timers will use a 25MHz master clock which is obtained by dividing a 50MHz oscillator signal by two. This will give the system a T_d of 40ns. The timers will be programmed through a 12 bit bidirectional address/data bus. The bus will be driven by a special 12 bit address counter which can be made to produce a random number from a pseudo random sequence generator, or a sequential number, depending on the control signals received. A smaller 4 bit counter is used to generate the higher order or most significant address bits (MSB). This counter must also be programmable in order to allow selection of the start data channel. The control logic will use available inputs, such as trigger mode, trigger pulse and EOC from the ADC, as well as control signals from the system interface to generate the required control signals for the various counters and timers. The system interface allows the user to select the desired sampling mode. It also enables the system to be placed in load or test mode. During load mode the bidirectional bus becomes a data bus which enables the timers and counters to be programmed with data generated by the user. The activation of test mode reverses the direction of the bus allowing internal signals to be propagated off chip for test purposes. During any sampling mode, the bus carries an address used by both the memory and a delay timer. The delay timer can use the address as data during non-real time sampling, thus generating the required delay for both sequential and random sampling to

take place.

3.5 Summary

This chapter has briefly presented the system that was developed by Carlos C. Pon [15] and which will be used as a pilot project. The reason for selecting this project is that it satisfies the complexity requirements for system integration. Having introduced the system to be integrated, the actual system configuration was finalized and the device requirements for the multi-mode data acquisition unit were determined. From these high level device requirements a general system overview was established. It is now the purpose of chapter 4 to fully develop this system overview into a more detailed functional block diagram of the system. The following chapter will also present the actual implementation of these functional blocks down to the transistor level and full layout of the integrated controller unit.

Chapter 4

Integrated Controller Unit4.1 Introduction

The first two stages of the design methodology have now been completed. The next phase is the development of a functional block diagram. From the functional block diagram, circuit specifications may be obtained that will enable the design of the required functional blocks. Once these have been designed and simulated the chip floor plan and layout will be presented. This will complete the design of the integrated controller and demonstrate the implementation of the design environment.

4.2 Functional Block Diagram

From the system overview of the integrated controller presented in chapter 3 (figure 3.7), a more detailed functional block diagram of the controller must be developed. Each functional block will represent a collection of circuitry that will perform one or more tasks, and will interface with other functional blocks to implement the desired system.

An initial functional block diagram is shown in figure 4.1. This diagram contains more detailed information on the various blocks than did the system overview of figure 3.7. Specifications such as counter and timer bit sizes and multiplexing circuits are now included. Also the number of programmable timers is clearly defined. The delay timer will generate the

Figure 4.1 Initial functional block diagram of integrated controller unit.

required sequential or random delay for the respective sampling modes. The RAM pulse, S/H pulse and ADC start pulse timers will generate the required control signals for data acquisition to take place. The channel counter will be used to count down the desired data channels. The Hold Time timer has been included to allow a programmable delay between the end of the S/H pulse and the ADC start pulse. This delay is to allow for the hold-mode settling time [17] (HMST) which varies between sample and hold devices. The settling time is the amount of time required between the hold signal and a stable output voltage from the S/H.

At this stage of the design, rough estimates were done on the area that would be required to implement the various counters and timers. It quickly became obvious that using positive edge triggered D flip-flops, whose area is $208u \times 384u$ (13×24 grids²) would not enable the placement of all the required counters in a single A size pad frame. CMC's A size pad frame gives a usable area of 4040×4040 microns, not including I/O pad circuitry. Due to these limitations it was required to restructure the functional block diagram. Two functional block diagrams were developed in order to implement the integrated controller unit on two A size pad frames. The two chip designs will hence forth be known as the dedicated processor chip and the address generator chip due to the functions that each will perform.

The functional block diagram for the dedicated processor chip is shown in figure 4.2. It contains all the programmable timers and counters needed to generate the control signals for the S/H,

Dedicated Processor Chip

Figure 4.2 Functional block diagram of the dedicated processor chip.

ADC and memory functions. The system interface as well as the required control logic for the timers is also included on this chip.

The functional block diagram for the address generator chip is shown in figure 4.3. This chip generates the required 16 bit address for the data acquisition system. The A12-A15 (MSB) address bus is separate from the main AD0-AD11 address/data bus. The address bus will hence-forth refer to the AD0-AD11 address/data bus unless other wise stated. As shown in figure 4.3 the address bus is internally multiplexed to a separate data bus. The data bus can be driven by the control panel or a micro-processor and used to load the desired data for the programmable timers and start channel address. The dedicated processor chip is interfaced to the address generator chip through several control lines and the address bus. During load mode the dedicated processor uses the address bus as a data bus and the address generator feeds its separate data bus to the address bus. When the system is put into a sampling mode, the required address is put on the address bus which the delay timer can use as data. The address generator data bus is tri-stated during a sampling mode, but when the system is placed in Test mode the data bus becomes a test bus for various internal signals on the address generator chip. There are two ramifications to having a two chip system. The first is that the first byte of data taken during sequential sampling is invalid and the second is that two consecutive writes to the same address location are needed during random sampling. These will be fully discussed in the following

Figure 4.3 Functional block diagram of the address generator chip.

section.

Now that the functional block diagrams have been established, more detailed information of the signal processing and I/O of each block is required. In order to accomplish this, flow charts showing the tasks carried out by the integrated system controller have been developed. From these tasks or macro instructions the required information for the design of each functional block will be obtained, and thus permit the instructions to be encoded into hardware. The instructions are encoded so that given certain inputs to the system, the logic will cycle through the required steps to produce the desired output. The instructions use common hardware, where the hardware is similar to micro code. The integrated system controller can be implemented with three such macro instructions, one for each of the sampling techniques.

The first instruction enables real time sampling and is shown in figure 4.4. The counter/timer initialization (load mode) phase is comprised of the following steps;

- load S/H timer with data bus
- load RAM timer with data bus
- load CHAN counter with data bus
- load HOLD TIME timer with data bus
- load ADC timer with data bus
- load MSB address counter with data bus
- reset ADO-AD11 address counter to 000Hex

Each timer will latch the data it receives enabling it to reload.

REAL TIME SAMPLING MODE

Figure 4.4 Real time sampling instruction flow chart.

the original data once the count has completed. During the select sampling mode phase, real time sampling is activated followed by the choice of either post-triggering or pre-triggering. Having made these specifications the hardware then follows the course taken by the flow chart described in figure 4.4. The timers and counters will be down counters with zero count detect capabilities. Once each individual counter attains a zero count, it is reloaded with the original data and enabled to receive a new start pulse. Once the channel counter has counted down the required data channels, an END signal is generated which can then be used to reset the system to load mode. The timing waveform that is generated by this macro instruction is shown in figure 4.5.

The second macro instruction presented allows the system to implement sequential sampling and is shown in figure 4.6. As during the real time sampling initialization phase, the counters and timers are loaded with the appropriate data. In addition the user must also load the delay timer (N count) with 000_{hex}. During the select sampling mode phase, sequential sampling is activated. This enables N count to be initiated during the rising edge of the trigger pulse. As mentioned earlier the trigger pulse is referenced with some point on the periodic waveform. Once N count has reached zero it then activates the S/H timer and ~~events unfold as shown~~ in figure 4.6. Note that N count is only reloaded once the address has been incremented and the digitized data written. For this reason the first address, 000_{hex}, does

Real Time Sampling Mode Waveform

Figure 4.5 Timing waveform for real time sampling.

Figure 4.6 Sequential sampling instruction flow chart.

not contain valid data. As in real time sampling, once the specified number of data channels has been taken the END signal is generated and the system returns to load mode. The timing waveform generated during sequential sampling is shown in figure 4.7.

The last macro instruction is that for random sampling, and is shown in figure 4.3. All the timers and counters are initialized as they were for sequential sampling during the initialization phase. An added function is the enabling of the pseudo random sequence [18] generator which is given the seed number of FFF_{Hex} . From this seed number the pseudo random number generator will generate a maximum length sequence. It does this using a 12 bit shift register with four taps taken at bit 0, 3, 5, 11. The N count timer is activated as it is for sequential sampling and the sequence of events unfolds as shown in figure 4.8. Note that now the address counter is clocked on even samples only and that the random number generator is clocked on odd samples after the first digitization. The reason for this is that the address counter must clock in the initial random number (FFF_{Hex}) which can then be loaded into the delay timer. Once this delay has elapsed and the waveform digitized, this data must be placed in address location $\#FFF_{Hex}$, where # is an arbitrary channel number. For this reason the address counter is not clocked to produce a new address. The delay N is hence not changed and the waveform is again digitized after a delay of $FFF_{Hex} \times 40ns$. Now the previously taken data is placed into the correct address $\#FFF_{Hex}$ and the address and delay N are given the value of the next

Sequential Sampling Mode Waveform

Figure 4.7 Timing waveform for sequential sampling.

Figure 4.3 Random sampling instruction flow chart.

RANDOM SAMPLING MODE (continued)

Figure 4.8 Random sampling instruction flow chart.

pseudo random number. Due to this sequence of events, two writes are needed to place the correct sampled data into the appropriate address. As before, once the required data channels have been taken the END signal is generated and the system returned to load mode. An example of the timing waveform generated for random sampling is shown in figure 4.9.

4.3 Functional Block Design

This section will demonstrate how the functional blocks may now be structured and designed in order to implement the macro instructions presented. Functional block schematics will be presented along with simulation results in order to demonstrate functionality of the circuit blocks. Only a few of the schematics will be presented in this section but all the functional block schematics with their related Mercury (Hg) simulations can be found in Appendix II. The appendix has been divided into two sections. The first presents the designs and simulations required to implement the dedicated processor chip. The second section presents those designs needed to implement the address generator chip. In this chapter three designs from each chip will be examined.

The three selected dedicated processor schematics will be from the system interface functional block and the programmable timers control logic block. The first schematic is from the system interface block and is shown in figure 4.10. This is a simple 2 to 4 decoder that will allow the user to select one of

Random Sampling Mode Waveform

Figure 4.9 Timing waveform for random sampling.

Figure 4.10 System mode logic schematic for system interface.

the four system modes (real time, sequential, random, or load). The two inputs are controlled off chip by the operator and the four outputs are control lines that will be used on chip by various blocks to put the system in the desired state. From the schematic, the truth table is simply,

A pad	B pad	System Mode
0	0	LVE load variable enable
1	0	SSE sequential sampling enable
0	1	RSE random sampling enable
1	1	RTE real time enable

An Hg simulation performed on the layout of this schematic is shown in figure 4.11. It clearly shows the correct functionality of this circuit block.

The next schematic is also from the system interface functional block and is shown in figure 4.12. It is a 3 to 8 decoder with clockable outputs that will enable the individual timers and counters to be clocked separately. Initially the operator must put the system in load mode. He/she then places the desired counter data on the data bus and using the circuit shown in figure 4.12, selects the desired counter with inputs A, B, C and then uses the CLK input to latch the data into the counter. The truth table for this circuit is as follows,

i) when CLK input is low (0) - all outputs are tri-stated.

ii) when CLK input is high (1) - the system clocks activated are as follows;

Figure 4.11 is simulation of system mode logic.

Figure 4.12 Initial loading clocks schematic for system interface.

A input	B input	C input	Initial Loading Clock
1	1	1	NCC - N counter clock
1	0	0	RPC - RAM pulse clock
0	1	0	SHC - Sample/Hold clock
1	1	0	HTC - Hold Time clock
0	0	1	ADC - Analog to Digital clock
1	0	1	IAC - Initial MSB address clock
0	1	1	CNC - Channel Number clock
0	0	0	TM - Test Mode

As can be seen, the system can be put in Test mode by selecting A, B, and C inputs as low, and leaving the CLK input high. The H₂ simulation of this design was obtained from a layout and is shown in figure 4.13.

The last circuit from the dedicated processor chip is from the programmable timer control logic. This control logic enables each counter to be programmed, started, detect zero count and reload ready for another start pulse. The schematic is shown in figure 4.14. Initially when the system is first powered up, the operator selects load mode using the system interface as mentioned earlier. Once this is done a 1 appears on the D of D flip-flop 1 (DFF1) and the rstbar and setbar also have a 1 on their inputs. Startbar (nSTART) must have a value of 1 at this point. This will be a requirement that must be met by other counter control logic. The initialization clocks may now be generated using the system interface. The initialization clock, in this schematic NCC, will clock a 1 into DFF1. This will put

Figure 4.13 43 simulation of initial loading blocks.

Figure 4.14 Programmable timers control logic schematic.

the counter in a load state by placing a 1 on the LOAD line. It will also enable the counter to be started by the nSTART pulse, and disable the zero bar (nZERO) input which is used to stop the counter. DFF1 also loads a 1 into DFF2 which has a 25MHz clock rate from the master clock (MCLK) input. The output of DFF2 disables the counter clock CK. When the initialization clock, NCC, is generated, it will also set a 1 on the latch clock (LCLK) input. This will be another requirement for other control logic. In most cases, LCLK and the counter initialization clock, (in this case NCC) are the same signal. When LCLK is high the data on the bus designated for this particular counter is loaded into the latches. In this case CO₀ is the zero'th bit counter data from the bus. Since the LOAD line is high, the latch can place the data into DFF3 using the nset and nrst inputs. DFF3 and the two NANDs and latch above it form a bit slice for the counters. For a four bit counter, the bit slice is used four times. The NAND below DFF3 is also part of the bit slice but must be alternated with a NOR. The purpose of these NANDs and NORs is to give an effective NAND function for all the counter bits. The output of this NAND function will be the zero bar (nZERO) signal which is used to stop the counter. The counter is started when nSTART goes low and resets DFF1. This disables the LOAD line, enables the counter clock CK and the nZERO signal. When the nZERO signal is received, DFF1 is set causing the counter clock CK to be disabled, and enabling the LOAD line and nSTART input. The LOAD line allows the data contained in the latches to be placed in the counter flip-flops (i.e. DFF3).

An Hg simulation showing this circuit in action is shown in figure 4.15. The nZERO signal is enabled by the NOR1 input which is destined to come from the other bits of a given counter.

The three selected address generator schematics are from the AD0-AD11 counter, A12-A15 (MSB) counter control logic, and AD0-AD11 counter control logic. The first schematic is from the AD0-AD11 counter. This address counter is constructed of 12 bit slices, where a bit slice contains a D flip-flop, an incrementer and a multiplexer, as shown in figure 4.16. R_n and A_n are the n th bit of the pseudo random number generator and the address counter output respectively. By sequentially placing the bit slices together a synchronous address counter of any number of bits can be generated. Within a bit slice, a multiplexer is used to interface the pseudo random number generator with the incrementer. The D flip-flop output (Q) not only goes to the incrementer but also to the address multiplexer, where the address bus is multiplexed to the data bus. The schematic examined is the increment logic in the bit slice as shown in figure 4.17, and suggested by Dave Doucet, a Carleton University masters student. The Q output of the D flip-flop is the A input to the increment logic. The carry in (C_{in}) for each bit is the carry out (C_{out}) of the previous bit, except for the first bit where C_{in} is Vdd. The incremented output A^+ , is then fed to the muxer where, if a sequential address is required, it will be placed on the D input of the flip-flop and clocked by the address clock (ADDCLK). The A output (A_n) which is simply the Q output of the flip-flop

Figure 4. Timing diagram of programmable timers control logic.

Figure 4.13 Address counter bit slice schematic.

2

Figure 4.17 Increment logic in address counter bit slice.

is fed to the address bus. A simulation showing the ADO-AD11 counter generating a sequential address is shown in figure 4.18. Note that (an) are the nth order address bits and that (cn) and (mn) are the nth bit carry out and mux output respectively. The undefined logic states are caused by anomalies in the Mercury simulator.

The next schematic shows the control logic used to clock the MSB address counter and the channel counter. The circuit performs the equivalent of a 12 input NAND function, and is shown in figure 4.19. Due to input capacitance limitations and required pullup and pulldown aspect ratios, the highest fan-in implemented in the standard cell library was 4. By using three 4 input NANDs and one 3 input NOR, the 12 input NAND function can be implemented. The H₂ simulation showing this function is shown in figure 4.20. The 12 inputs are tapped from the address bus, allowing the circuit to detect an address of FFF_{Hex} and clock the MSB address and channel counters.

The last circuit is the logic required to clock the ADO-AD11 counter and pseudo random number generator. Recall that during random sampling (RSE), the address counter and pseudo random number generator must be clocked alternately by the RAM pulse bar (RAMBAR) signal. The control logic is shown in figure 4.21. After power up, the LVE signal is again used to put the logic in a known state. When random sampling is selected, the RSE input will be high and cause the address counter clock (ADDCLK) and pseudo random generator clock (RANCLK) to alternate. This is

Figure 4.18 H₃ simulation of AD0-AD11 counter generating a sequential address.

Figure 4.19 MSB address counter clock control logic.

Figure 4.20 Hg simulation of MSB address counter clock control logic.

Figure 4.21 Clock generation logic for the ADO-AD11 counter and pseudo random number generator.

generally done by having DFF1 divide the RAMQCBAR input frequency by two and then using that output for ADDCLK and the D input to DFF2 whose output is RANCLK. No race condition exists because the D flip-flop requires a valid D input 5ns before the clocked input. Since DFF1 and DFF2 are clocked simultaneously, the delay of DFF1 and three 2 input NANDs ensures that the D input to DFF2 will not be changed 5ns before the clock. When random sampling is not selected, RSE is low, the address counter is clocked on every rising edge of RAM bar pulse (RAMQCBAR), and RANCLK is not generated. An Hg simulation clearly showing the functionality of this control circuitry is shown in figure 4.22.

This completes the discussion on selected schematics from the dedicated controller chip and address generator chip. As mentioned earlier, all the schematics used in both chips can be found in Appendix II.

4.4 Chip Floor Plan and Layout

Once all the functional logic block schematics have been designed and simulated, rough area estimates can be made for each block and a chip floor plan may be attempted. During the implementation of this project time was a limiting factor, so each functional block was designed and laid out using the standard cells. The circuit simulations were then done on the layouts. Once the functional blocks were simulated, a more accurate floor plan could be done using the layout of each functional circuit block.

Figure 4.22 Rg simulation of clock generation logic for the AD0-AD11 counter and pseudo random number generator.

The chip floor plan for the dedicated processor chip is shown in figure 4.23. All the functional logic blocks are shown, including buffers which will be used to assist the propagation of certain signals on the chip. Once the floor plan is complete the interconnect between the various blocks may be laid out. The Metheus allows the designer to use labels during layout which are useful in keeping track of the various signals. It was found that doing the interconnect by functional block was easier than trying to run signals from block to block. Once the interconnect is completed, the designer can then select various sections and perform DRC to ensure no violations exist. The power rails may then be run to supply the various blocks with the appropriate voltage levels. It was found easier to run one rail, i.e. Vdd, to each block, then to run both Vdd and Vss concurrently to each block. When running the power rails, poly underpasses must be avoided at all costs, so in many instances, several signal lines must be modified in order not to disrupt the metal power rails. Once the layout is complete, the designer can now perform a Netlist extraction from both the layout and a synthesized schematic of the chip. These can then be compared for inconsistencies. By synthesizing a chip schematic, the interconnect is essentially performed twice, thus reducing the possibility for error in the layout. The functional block interconnect is the most error prone operation since overall chip simulations are very difficult to perform, due to time and CPU limitations. Once the Netlist compare is completed and error free, the layout is now ready for fabrication. The completed layout of the dedicated processor

Figure 4.23 Floor plan for dedicated processor chip.

chip is shown in figure 4.24.

The chip floor plan for the address generator chip is shown in figure 4.25. The functional blocks and buffers are all placed in order to facilitate the task of interconnection as much as possible. As mentioned earlier, the interconnect between various blocks is performed and the power rails are laid out. DRC is again performed on the interconnect channels to ensure design rule conformity. A Netlist compare can also be performed to reduce the possibility of incorrect signal propagation on the chip. A complete layout of the address generator chip is shown in figure 4.26.

The layouts are now converted to CIF using a command in PHLED. The file takes several minutes to generate and uses a cell hierarchy. These files are then copied to magnetic tape using a TAR command. The tape and a top level chip layout with a brief chip description are sent to the VLSI implementation center at Queen's University. The designer now has three to four months to prepare test jigs which will facilitate in the testing of the designs.

4.5 Summary

This chapter has presented the final steps of the design methodology, from functional block diagram to chip layout. From the functional block diagram and macro instructions, design specifications were extracted. From these specifications two chips, the dedicated processor and the address generator, were

Figure 4.14. Dedicated processor chip layout.

Figure 4.25 Floor plan for address generator chip.

Figure 4.25 Address generator chip layout.

designed and implemented. By following the design methodology, the designer can structure the implementation of a complex system in a hierarchical fashion as shown here. This reduces the possibility for error and simplifies the integration task. The chapter clearly depicts the usefulness of a sophisticated design environment and the advantage of a well developed design methodology supported by the appropriate CAD tools.

Chapter 5

Design Verification and Characterization

5.1 Introduction

This chapter will discuss the test results obtained from the chips generated during this thesis work. The first section is on the standard cell test vehicle which was used to determine functionality of the primitives designed for the design environment. The second section will discuss results obtained from both the dedicated processor chip and the address generator chip. The final section will give more technical results obtained from the dedicated processor chip which will help in the characterization of Northern Telecom's CMOS1B process.

5.2 Standard Cell Test Vehicle

The standard cell test chip was completely designed, using the Corvus Concept workstations, by Richard E. Vallee and Ralph D. Mason. The purpose of this chip was to determine whether the cells developed were functionally correct. Techniques for designing chips that will allow characterization of the standard cell library are given in Chapter 6. A complete list of the primitives contained on the test chip, as well as the pin assignments are given in Appendix III. The results obtained were not detailed measurements of the standard cells. The reason is that the chips were damaged while being tested for functionality. For some unknown reason, when certain inputs are given a logic high

state, the chip shorts. The problem remains unresolved even after close examination of the chip with a microscope. It is difficult to examine the chip using the Metheus, since it was generated as one large CIF file by the Corvus. The following sections will discuss some of the functional results obtained from the test chip.

The first results discussed were obtained from the input protection device and the output buffer. This circuit consisted of an input pad with two protection diodes. The grounding diode would be forward biased if inputs below V_{ss} were placed on the pad, and the voltage limiting diode would be forward biased if the input voltage were above V_{dd} . The input signal is brought to an output pad which is buffered by two cascaded inverters. Using simple test equipment capable of delivering a 2MHz input signal and observing waveforms with 50ns resolution, no propagation delay could be measured between input and output signals. The results demonstrated functionality of the I/O primitives, but did not allow determination of an upper limit on the input or output frequency.

Using an input and output pad, inverta (a size INVERTER) cascaded to invertb (b size INVERTER) were tested. The inverters proved to be functional and a propagation delay of 0.1us was measured. This would tend to indicate a fairly high delay for the two inverters, since without the inverters, no delay was measured.

A second inverter string of three inverters was tested next. The inverters were cascaded from smallest (minimum geometry) to largest (4 times minimum geometry) and then fed to the output buffers. The propagation delay measured from this circuit was 0.15us. The fact that it is larger than the previous test is reasonable since the b size inverter was much larger than the last inverter in this string.

A divide by 2 circuit made up of a masterJK flip-flop with Q and Qbar fed back as inputs, and clock and clockbar inputs driven from cascaded inverta and invertb primitives was tested. The clock signal is fed through the two inverters to produce clock and clockbar. The Q output is then buffered out through 2 inverters to an output pad. The output signal divided the input signal by 2, in the frequency domain and the circuit tracked perfectly to 2MHz. The upper operating frequency of the circuit was not determined. It is hoped that this circuit will divide the 50MHz input frequency to the dedicated processor chip.

The last circuit presented is the positive edge triggered D flip-flop. The D flip-flop used Qbar as the D input and could be made to toggle with a clock input. The output divided the input in half to the highest input frequency. The setbar and rstbar functions were tested and found functional. With the test equipment at hand, the switching speeds for setbar and rstbar could not be determined.

With the results obtained from the test chip, it would seem that the cells developed are functional and display fairly good

driving capabilities. These results would indicate that chips made from these cells have a good chance of showing functionality. It is difficult to compare results from this chip to SPICE simulations because large probing pads were placed on certain signals to allow testability of circuits in the event of defective I/O pads.

5.3 Integrated Controller Unit

This section will discuss the results obtained from both the dedicated processor chip and the address generator chip. The chips were tested at the Ontario Centre for Microelectronics (O.C.M.) on an H.P. 8000 series test station, identical to that made available to Carleton University through C.M.C. Carleton has not received all the parts required to construct a usable 40 pin test jig. Test results were obtained from a jig available at O.C.M. Due to design errors which have since been corrected, both chips could not be entirely tested. The design errors were caused by the use of analog (3 transistor) passgates. Since the P-wells of these cells were not electrically isolated, a short circuit could occur under certain conditions. The error was corrected by simply using a regular 2 transistor passgate. The dedicated processor chip results will be discussed first.

The first circuit tested is the clock generator. This is the masterJK divide by 2 circuit discussed earlier and for which more detailed test results were obtained. This circuit will also be used to characterize the CMOS1B process as discussed in section 5.4.

The following results show a maximum operating frequency of 40 MHz:

<u>Input Frequency</u>	<u>Output Frequency</u>
50MHz	12.5MHz
45MHz	15.0MHz
40MHz	20.0MHz
25MHz	12.5MHz
20MHz	10.0MHz

Above 40MHz the flip-flop cannot synchronize with the input frequency and begins to divide by 3 and 4. A plot of sampled data taken from the H.P. 8182A data analyzer at 40MHz is shown in figure 5.1. From this figure the divide by 2 function is clearly shown. The current requirements for this circuit at 40MHz were measured to be roughly 0.3 uamps. Figure 5.2 is a photograph taken from an oscilloscope showing the actual input and output traces from this circuit at 20MHz.

The second circuit tested is the system mode logic as presented in chapter 4. This is a simple 2 to 4 decoder used to set the sampling mode of the controller. Three outputs are available and the test results are shown in figure 5.3. The upper two signals are the inputs, generating 3 of the 4 possible states, and the lower 3 signals demonstrate the decoding. The results clearly show the functionality of this circuit, located in the user interface block. Due to a design error in the initial

CLOCK GENERATOR 40 MHz INPUT

Figure 5.1 Sampled data results of masterJK divide by 2 at 40MHz

Figure 4.1 Oscilloscope traces of laser pulse current at 10 Hz.

SYSTEM MODE LOGIC

Figure 5.3 Sampled data results of the system mode logic.

counter clocks of the system interface, no other circuits could be tested on the dedicated processor chip.

Results obtained from the address generator chip are now presented. Only minor design errors were found on the address generator chip when extensive simulations of the functional blocks were performed. For this reason very good test results have been obtained from this chip. The sequential address generator was initially tested and found to be fully functional. Eight of the ADO-AD11 bus lines were monitored as shown in figure 5.4. The MSB address counter was also being clocked but could not generate the full 16 addresses. The last bit has since been corrected as the simulation in Appendix II shows. The address generator was unable to generate a random sequence. This is due to a design error in the clocking of the pseudo random number generator. This circuit has again been corrected for the new submission of the design. The data path from the data bus inputs to the address bus used for programming the dedicated processor chip in load mode was tested next. This was found to be 50% operational. The data could be fed through the chip during load mode, but the chip could not be put into test mode, again due to a slight design error.

5.4 Design Characterization

Using the clock generator on the dedicated processor chip, more detailed results will be presented in order to allow some characterization of the CMOS1B process and the standard cells designed. The data is summarized in table 5.1 and contains data

AD0-AD11 ADDRESS COUNTER (AD0-AD7)

Figure 5.4 Sampled data results of the AD0-AD11 counter.

on the output rise and fall times, as well as propagation delays for various Vdd voltages and 1MHz frequency.

TABLE 5.1

<u>Vdd</u>	<u>Vin(freq)</u>	<u>Vout(freq)</u>	<u>t_r</u>	<u>t_f</u>	<u>T_r</u>	<u>T_f</u>	<u>I(Vdd)</u>
1.0v	5kHz	2.5kHz	8.8us	3.2us	1.7us	9.2us	0.7uA
3.0v	1MHz	0.5MHz	30ns	24ns	74ns	66ns	60.5uA
5.0v	1MHz	0.5MHz	18ns	12ns	40ns	36ns	636uA
10.0v	1MHz	0.5MHz	16ns	10ns	24ns	24ns	10.8mA
12.0v	1MHz	0.5MHz	14ns	8ns	26ns	22ns	16.5mA

The loading conditions under which this data was collected is estimated to be between 20pf and 30pf.

The data in Table 5.1 indicate that as the Vdd voltage is increased, both the propagation delays and rise/fall times are shortened. This is in agreement with the equation for propagation delays which states;

$$T_{pavg} = C_T (V_{dd}/2) / I_{avg} \quad (\text{equ. 1})$$

where C_T is the total capacitance load on the output, and I_{avg} is the average current through the p and n channel transistors. The average current (I_{avg}) can be approximated by the drain source current of a transistor in saturation, which is;

$$I_{avg} = I_{DS} = \frac{\mu}{2} (V_{GS} - V_T)^2$$

$$\text{for } V_{DS} > (V_{GS} - V_T) \text{ and } V_{GS} > V_T$$

where K is the device transconductance parameter and is defined as;

$$K = U_n C_{ox} (W/L)$$

where U_n is the electron mobility in the channel and C_{ox} is the gate capacitance per cm^2 . V_T is the threshold voltage with V_{DS} and V_{GS} being the drain and gate to source voltages respectively. Since K is constant and only slight variations occur in V_T , the voltage V_{GS} will increase as V_{dd} is increased causing the current I_{DS} to increase. With I_{DS} increasing with the square of the increased V_{dd} , the propagation delay (equ. 1) will decrease. The rise and fall times will increase due to the large currents available to charge and discharge the load capacitance. The results in Table 5.1 follow this same behavior. As the frequency is increased, the number of transistions in states increases, causing the average current for a given interval of time to increase. The average power dissipation for the chip is given by;

$$P_{avg} = (C_T V_{dd}^2) f$$

where f is the frequency

From the data that have been collected in Table 5.1 it will now be possible for more detailed analysis to be carried out on the output buffer of the cell library. The output buffer is the last driving stage before data is collected and thus is the only real cell for which data can be collected. From this data it should be possible to obtain better parameters with which to

perform more accurate simulations.

5.5 Summary

In this chapter results obtained from chips fabricated at Northern Telecom Ltd. were presented. The standard cell test vehicle was tested and the standard cells found to be functional. Detailed characterization of the cells was not possible due to the presence of a short circuit and the on chip signal probe pads. These signal probe pads were included so that if the I/O pads proved defective some test results could still be obtained. By placing these probing pads on internal signals, the load for those signals is difficult to estimate. Knowing now that the cells and I/O pads are functional, a new standard cell test vehicle with standardized load conditions should be submitted to allow full characterization of the standard cell library.

The results of the integrated controller unit were also presented. The threat of a Vdd to Vss short in the p-well, caused by the use of analog passgates (pass3gate) instead of the normal 2 transistor passgates, limited the amount of obtainable results from the chips. These as well as other slight design errors have been corrected, as shown by the simulation results in Appendix II. The data collected for design characterization should allow future students to adjust some of the simulator parameters to more closely reflect the actual circuit results.

The next chapter will evaluate the results obtained from the system implemented by the design environment.

Chapter 6

Evaluation of Design Environment

6.1 Introduction

This chapter will draw conclusions on the material that has been presented. The first section will discuss the design verification results in proper context and demonstrate that the design environment developed at Carleton University is a viable one. The second section will discuss future work that has already commenced and that should be done in order to strengthen the design environment.

6.2 Design Verification Results

In order to justify the results presented in chapter five, it is important to understand the time frame in which this work has been prepared. The pilot project was selected in May, 1984. At this time the only CAD tools available were the HP 8000 series test station, (arrival date April 29, 1984), and the Corvus Concept Network. After spending a few weeks determining the initial system overview that would allow integration of the controller unit, it became obvious that a design strategy would be necessary. Knowing that a Metheus Workstation was on its way, it was decided that time could best be spent developing the ground work for a design strategy. During the few months that followed a standard cell library was developed by Richard E. Vallée and Ralph D. Mason. The standard cell test vehicle was

completed on the Corvus and submitted to CMC at the end of August. The Metheus was delivered on September 21, 1984.

Once the Metheus had arrived the task was now to learn UNIX and available software on the system. The first major obstacle was the technology data base on the Metheus. The Metheus was fully implemented for NMOS design, with little resources for CMOS. The required CMOS technology files were then developed to allow the design of CMOS which would conform to the CIF available at Carleton and required by CMC. Having achieved this, the next task was to transfer the standard cell library developed over the summer to the Metheus. Once the primitives were in the machine, work could then be done in order to implement the required data base for the design environment. With the knowledge acquired by Ralph D. Mason on a one week training program in Portland Oregon, Hg simulations could then be run on circuits designed. This soon became essential because of the complexity of the circuitry. A simple 4 bit counter contains approximately 300 transistors. In order to simulate this using SPICE, several hundred CPU hours would be required. With Hg, this circuit can be tested in only a few hours.

After laying out several of the functional blocks described earlier it became obvious that the design would not fit in the largest available padframe. The system overview had to be restructured in order to allow the system to be implemented on two chips.

In order to complete the designs by the required submission

date of December 17, 1984, several of the design steps were omitted. Only limited Hg simulations were performed on the functional blocks and no DRCs were performed.

Taking into consideration the time frame in which both designs were completed, and the fact that several key design steps were omitted, the results in chapter 5 are surprising. The fact that several functional blocks demonstrated complete functionality, and that others performed without ever being simulated or DRCed is amazing. Due to the results obtained and in light of the design circumstances, it is strongly felt that the design environment is a viable one. Since the transistor count on the dedicated processor is 2822 transistors and 1990 for the address generator chip, it is felt that over the course of a normal 4 month semester, 3000 transistor designs could be generated while completing all facets of the design methodology. By performing the required DRC and functional block simulations, the probability for a first time correct design are very high.

Simulations on both the dedicated processor and the address generator chips have since been carried out on all the functional blocks. It is confident that the new designs resubmitted will have a much higher probability of success.

6.3 Future Work

Now that a viable design environment has been established, ways of improving and expanding the capabilities of the environment will be discussed.

Initially the future of the available cell library will be discussed. From the results of the standard cell test vehicle and the pilot project it is safe to assume that the cells are functional. A new standard cell vehicle should be submitted in order to characterize the cells. This could easily be laid out on the Metheus and a common load, such as an inverter, could be placed on each cell. To increase the load, two or three inverters could be used. In this way all the results could be normalized to an inverter loading. Once the cells have been characterized under various load conditions, the SPICE parameters may be tuned in order that the SPICE simulations approximate actual results. This would then allow better tuning of the H_z parameters. Once the designer has simulations that closely approximate the behavior of the cells, the probability for first time correct designs increases. Work can also be done in implementing more primitive as well as macro cells. Once large cells such as counters and shift registers have been designed and characterized, the designer can do layout work at the systems level instead of at the primitive cell level. The development of analog circuit elements would also be of great value. Macro cells such as filters, integrators, and comparators, once designed and characterized would greatly increase the number of analog, and mixed digital analog circuits that could be designed. The ability to do top down design with good functional blocks already available will not only decrease design time, but also greatly increase the probability of first time correct designs.

Future work can also be carried out in software developments. Languages supported by the Metheus include Fortran, C, and Pascal. By developing auto-placement and auto-routing software, the layout stage of the design strategy could be fully automated. By using the Netlist generated from a schematic, the program could call the required primitive or even micro cells together and perform the interconnect so that a Netlist generated from the layout exactly matches the one specified by the schematic. Also by developing efficient algorithms for both placement and routing, the packing density achieved could rival that of a well seasoned I.C. designer.

In addition to auto-placement and auto-routing algorithms, work could also be carried out in hardware description languages. This would allow future designers to implement algorithms or macro instructions, as shown in this thesis, into integrated circuit elements. Ultimately the cells could be used at the lowest level of a mixed analog/digital silicon compiler. The silicon compiler would allow the designer to specify the circuit requirements in some user language, and then construct a circuit schematic and layout conforming to the desired constraints.

Recently CMC has announced the availability of Northern Telecom Ltd.'s 3um CMOS process. Richard E. Vallee and Ralph D. Mason have already been involved in the development of a new 3um CMOS standard cell library with Siltronic Ltd. in Kanata, Ontario. The library is already comprised of 140 primitive and micro cells, and once complete, the 210 primitive and micro cells

will be made available to Carleton University. Using this new cell library, the complexity of designs can be increased 6 fold due to the large savings in area. For example, the 5um process inverter is 208u x 48u, but the 3um process inverter is 68u x 48u. This is a reduction of 5 in the area occupied by a cell. The savings in area are even more substantial for larger cells such as D flip-flops. The interconnect channels will also be reduced due to a reduced metal and poly line width. The cells are based on a 4 micron grid system with center justified I/O to each cell. This will allow cells to be used with auto-placement and auto-routing software. Work in this area is already taking place with the Ontario Centre for Microelectronics (OCM). The cells are presently compatible with Mitel's 3um ISO CMOS process and a proposed 3um process for the Canadian Semiconductor Design Association (CSDA). With minor modifications the cells can be made to conform to Northern Telecom Ltd.'s 3um process while maintaining Mitel's and CSDA's design rules. By having designs that conform to all these processes some area is wasted, but the flexibility of being able to submit designs to 3 fabrication centres far outweighs the loss in area.

With the advent of this new 3um cell library and a 6 fold increase in design complexity and based on the work presented, Carleton can design chips with upwards of 18,000 transistors. By fully managing and cultivating an interest in integrated circuit design, Carleton can play a significant role in VLSI Design in Canada.

6.4 Summary

This chapter has presented the results of both the developed design environment and the integrated circuits that it generated. The results of the pilot project presented in chapter 5 can now be more realistically evaluated, due to the circumstances with which it was implemented. Taking into consideration the design time and the fact that key design steps were omitted, and given the results presented in chapter 5, it is obvious that the design environment developed at Carleton University meets the initial goals set out by this thesis. It is clear that if the design methodology is followed, complex systems can be integrated within reasonable time frames for universities. The ways in which the design environment can be improved were presented in section 5.3. By implementing one or more of the suggestions presented, the implementation time and correctness of the integrated systems will be enhanced, thus enabling Carleton University to play a key role in VLSI design in Canada.

REFERENCES

1. Mead C., Conway L., " Introduction to VLSI Systems ", Addison-Wesley, 1980.
2. Chan C.H., Mason R., Vallee R.E., " A CMOS Implementation of a Multi-Mode Data Acquisition System ", Sixth Biennial University/Government/Industry Microelectronics Symposium, Auburn University, Alabama, pp 165-170, June 1985.
3. Lewis E.T., " Optimization of Device Area and Overall Delay for CMOS VLSI Designs ", Proceedings of the IEEE, Vol.72, No.6, June 1984.
4. Barros G., " An Overview of Array Design on Engineering Workstations ", VLSI Design, pp 42-50, February 1985.
5. Guterl F., Wallich P., Fischetti M., " In Pursuit of the One Month Chip ", IEEE Spectrum, Vol.21, pp 28-49, September 1984.
6. Anceau F., Reis R., "Complex Integrated Circuit Design Strategy ", IEEE Journal of Solid-State Circuits, Vol.SC-17, No.3, pp 459-464, June 1982.
7. Breuer M., Kumar A., " A Methodology for Custom VLSI Layout", IEEE Transactions on Circuit and System, Vol.Cas-30, No.6, pp 358-364, June 1983.
8. Kessler A.J., Ganesan A., " Standard Cell VLSI Design: A Tutorial ", IEEE Circuits and Devices Magazine, Vol.1, No.1, pp 17-33, January 1985.
9. Seth S.C., Agrawal V.D., " Cutting Chip-Testing Costs ", IEEE Spectrum, Vol.22, No.2, pp 38-45, April 1985.
10. Borriello G., Katz R., Bell A., Conway L., " VLSI System Design By The Numbers ", IEEE Spectrum, Vol.22, No.2, pp 44-50, February 1985.
11. McClusky E.J., " A Survey of Design for Testability Scan Techniques ", VLSI Design, pp 38-61, December 1984.
12. Glendinning D., Copeland M., " A Personal Workstation Environment for Electrical Engineering Education ", Masters Thesis, Department of Electronics, Carleton University, Ottawa, Canada.
13. "8180A Data Generator Operating and Programming Manual ", Hewlett Packard.

REFERENCES (Cont.)

14. "8182A Data Analyzer Operating and Programming Manual ", Hewlett Packard.
15. Pon C.R., " Design in Characterization of a Microprocessor-Controlled Digital Waveform Recorder ", Masters Thesis, Department of Electronics, Carleton University, Ottawa, Canada.
16. Mason R.D., " An Advanced Workstation Environment For CMOS VLSI Design and The Implementation of a Multi-Mode Data Acquisition Memory Control Unit ", Masters Thesis, Department of Electronics, Carleton University, Ottawa Canada.
17. Zuch E., " Data Acquisition and Conversion Handbook ", Datel Intersil, pp 153-185, June 1980.
18. Motorola Semiconductor Products Inc., " McMos Motorola Handbook ", Motorola Inc., 1974.
19. Nagel L.W., " SPICE2 : A Computer Program to Simulate Semiconductor Circuits ", ERL Memo No. ERL-M520, Electronics Research Laboratory, University of California, Berkeley, May 1975.
20. Krause M.A., Postgraduate thesis work in preparation, Department of Electrical Engineering, Queen's University, Kingston Ontario, 1984
21. Metheus Manual, " HG Theory of Operation ", Logic Simulator Reference, pp 11.18 - 11.30, January 15, 1984.

<u>Parameter</u>	<u>PMOS</u>	<u>NMOS</u>	<u>Units</u>
MJSW	0.50	0.50	-
JS	4.19E-10	1.37E-05	A/m**2
PB	0.70	0.70	v
XJ	9.00E-7	1.00E-6	m
LD	6.00E-7	7.00E-7	m

Having obtained the required model parameters, the device parameters will now be investigated. Typically the device parameters are specified for each device in the device line of the simulation deck using the following labels;

• W - the channel width (m)

L - the channel length (m)

AD - area of the drain (m**2)

AS - area of the source (m**2)

PD - perimeter of the drain (m)

→ PS - perimeter of the source (m)

NRD - number of squares of the parasitic series resistance in the drain.

NRS - number of squares of the parasitic series resistance in the source.

Level = 1 (default) Shichman-Hodges model

Level = 2 MOS2 model

Level = 3 semi-empirical model

VTO - is the extrapolated zero-biased threshold voltage for a long and wide channel device. This voltage is derived at the onset of strong inversion and marks the point where the device starts conducting if the weak inversion current is neglected. If VTO is not specified it will be calculated using NSUB and TOX.

KP - is the intrinsic transconductance parameter. If not specified, it is computed from TOX and UO (surface mobility).

GAMMA - Is the bulk threshold parameter and represents the proportionality factor relating the change in threshold voltage and backgate bias. If it is not specified, it is calculated from NSUB and TOX.

PHI - is the surface potential at strong inversion. When not specified it is also computed from NSUB.

LAMBDA - is the channel length modulation parameter and is equivalent to the inverse of the Early voltage for the bipolar transistor. It is also a measure of the output conductance in saturation. If not specified in a Level = 1 simulation, zero is used. For a Level = 2 simulation, it will be calculated.

TOX - is the oxide thickness under the gate and comes into the conduction factor, backgate bias effect and gate-channel voltage-dependent capacitance calculations.

CGSO - is the gate-source overlap capacitance per meter channel width and represents the capacitance due to lateral diffusion of the source.

CGDO - is the gate-drain overlap capacitance per meter channel width.

CGBO - is the gate-bulk overlap capacitance per meter channel length and is a field oxide capacitance. It is due to the gate extension beyond the channel over field oxide.

NSUB - is the substrate doping and is used in the derivation of most electrical parameters.

TPG - is the type of gate parameter and specifies the polarity type relative to the substrate.

UO - is the surface mobility at low gate voltages.

UCRIT - is the critical field for mobility degradation and is the limit at which the surface mobility UO starts decreasing.

UEXP - is the critical field exponent for formula which characterizes the degradation of the surface mobility.

RD and RS - are the drain and source ohmic resistance that

apply to all devices invoking the model in which they are specified. They are obtained from the number of squares in the drain and source regions respectively, and RSH.

RSH - is the sheet resistance at the drain and source diffusions.

CBD and CBS - are the zero-bias bulk-drain and bulk-source junction capacitances and are total values for all devices which reference the model in which they are presented.

CJ - is the zero-bias bulk-drain and bulk-source junction capacitance per m^2 of junction area.

MJ - is the grading coefficient of the bulk-drain and bulk-source junction and applies in all cases.

CJSW - is the zero-bias junction sidewall capacitance per meter of drain and source perimeter.

MJSW - is the grading coefficient of the junction sidewall.

IS - is the reverse current of the drain or source junctions and is a total value for all devices which reference the model containing IS.

JS - is the reverse current density of the drain or source junction per m^2 of junction area.

PB - is the bulk junction potential used in the junction capacitance calculation.

XJ - is the metallurgical junction depth.

LD - is the lateral diffusion coefficient in meters.

A table of the general SPICE model parameters needed for either a level 1 or 2 simulation are given below.

<u>Parameter</u>	<u>PMOS</u>	<u>NMOS</u>	<u>Units</u>
LEVEL	1/2	1/2	-
VTO	-0.90	0.90	v
KP	9.75E-06	3.05E-05	A/v**2
GAMMA	0.634	1.592	v**0.5
PHI	0.612	0.695	v
LAMBDA	3.00E-02	1.00E-02	1/v
TOX	8.50E-08	8.50E-08	m
CGSO	2.44E-10	2.84E-10	F/m
CGDO	2.44E-10	2.84E-10	F/m
CGBO	2.00E-12	2.00E-12	F/m
NSUB	1.98E+15	9.92E+15	1/cm**3
TPG	-1.0	1.0	-
UO	240.0	750.0	cm**2/v-s
UCRIT	6.44E+04	1.23E+05	v/cm
UEXP	0.139	0.022	-
RSH	75.0	15.0	nm/sq
CJ	1.54E-04	3.44E-04	F/m**2
XJ	0.50	0.50	-
CJSW	4.37E-10	1.09E-09	F/m

<u>Parameter</u>	<u>PMOS</u>	<u>NMOS</u>	<u>Units</u>
MJSW	0.50	0.50	-
JS	4.19E-10	1.37E-05	A/m**2
PB	0.70	0.70	v
XJ	9.00E-7	1.00E-6	m
LD	6.00E-7	7.00E-7	m

Having obtained the required model parameters, the device parameters will now be investigated. Typically the device parameters are specified for each device in the device line of the simulation deck using the following labels;

W - the channel width (m)

L - the channel length (m)

AD - area of the drain (m**2)

AS - area of the source (m**2)

PD - perimeter of the drain (m)

PS - perimeter of the source (m)

NRD - number of squares of the parasitic series resistance in the drain.

NRS - number of squares of the parasitic series resistance in the source.

If the above information is specified for each device then the model parameter list is complete. When doing SPICE simulations on the Metheus, the device parameters are typically extracted using PHLEX. PHLEX includes only the transistor W and L as device parameters. It multiplies the source and drain area with CJ, and adds interconnect capacitance of each source and drain to give a total capacitance for the specific drain and/or source node. For this reason the user should estimate an average value for AD, AS, PD, PS, NRD and NRS and evaluate the following;

$$CBD = (AD * CJ) + (PD * CJSW)$$

$$CBS = (AS * CJ) + (PS * CJSW)$$

$$IS = [(AD + AS)/2] * JS$$

$$RD = NRD * RSH$$

$$RS = NRS * RSH$$

Their values are then included with the model parameters. CBD and CBS should be included to allow some voltage dependent capacitance in the simulation. The fact that PHLEX extracts the drain and source area capacitance and adds it to the node capacitance makes for a worst case simulation. That is each simulation will have nodes with more capacitance than usual. By following the steps, reasonably good simulations for digital designs can be obtained using the Metheus. For analog simulations, more care is needed since PHLEX cannot distinguish a transistor's source or

drain. Analog SPICE decks should be generated by hand at this time. Level 3 simulations are not yet possible since the required parameter values have not yet been determined.

Af.2 Notes on H₂

H₂ is a Metneus [21] modification of RNL. RNL is a switch level simulator designed to handle ratioed logic, bidirectionality and charge sharing/storage, and was developed at MIT. It can be used to determine the functionality and approximate timing behavior of circuits at either a schematic or layout level.

Basic operation of the H₂ is as follows. A node and transistor representation of the circuit is first established. The effective resistance of the on transistor is determined by its type (n-channel or p-channel), width and length. There are three resistances associated with a transistor. They are, static, dynamic-low, and dynamic-high, which will be explained later.

Once the program has established the capacitance of each node and the resistance of each transistor, the user defined input voltages are applied and the effects determined by the following sequence until no further nodes change value.

- 1.) Activated nodes are added to the network and the charge sharing implications on the nodes capacitance and logic state on its electrical neighbors is completed.
- 2.) For each affected node the Thevenin equivalent circuit parameters V_{tnev} and R_{tnev} are calculated. The V_{tnev}

will determine the new logic state.

- 3.) If the node changes state, calculate the transition time using the node's capacitance.
- 4.) Propagate the changes to the neighboring nodes.

The simulator uses the notion of an event, where an event specifies a node in the network, a new logic state and the time at which the node's value is changed to a new logic state. The program adds an event to the list when the user changes an input, and maintains the list of events, sorted by time until all nodes have settled and the event list is empty. An event is processed by;

- a) removing the event from the event list
- b) changing the node's state to its new value
- c) calculating any consequences such as new events resulting from the node's new value

This algorithm is called an event-driven or selective trace since nodes are only added to the event list when their values change. Portions of the circuit unaffected by the current set of changes are not re-evaluated.

The transistor models can be quite simple since it is only used to predict the final logic state of the node and how long each state transition takes. The transistors are modelled as average channel resistances whose values are determined by the

logic state of the transistors' terminal nodes and transistor type. Each node is treated in a characteristic voltage divider configuration in order to calculate V_{chev} and R_{chev} , as shown in figure AI.1(a). R_H and R_L are the resistive values for the transistors that connect the node to a high and low potential respectively. The final state of a node can be characterized by a voltage source with a series resistor, or Thevenin circuit equivalent for all segments of the network that influence the given node. The Thevenin circuit is determined as shown in figure AI.1(b). V_{chev} is a voltage interval (V_- , V_+) determined generally by;

$$V_- = R_{L-} / (R_{L-} + R_{H+})$$

$$V_+ = R_{L+} / (R_{L+} + R_{H-})$$

and R_{chev} is a worst case Thevenian resistance determined by;

$$R_{chev} = R_{L+} // R_{H+}$$

Different values for R_{chev} can allow different conclusions about the node as to whether it is an input, driven, or charged node. Once the final logic state of a node has been determined using the Thevenin equivalents, the transition times are then calculated using R_{GND} and R_{VDD} which are the effective resistances of all direct paths to GND and Vdd respectively.

The node transition time is calculated from $R_{eff} * C_{eff}$ where R_{eff} is R_{VDD} for a logic 1 final state and R_{GND} for a logic 0 final state. C_{eff} is the effective capacitance of the node in

a) characteristic voltage divider for a node

b) Thevenin circuit equivalent for a node

Figure A1.1 Characteristic voltage divider and Thevenin circuit equivalent for a node, used by Hg.

question.

The three characteristic resistances specified by the user are used as follows;

static resistance - used to calculate RH and RL

dynamic-low resistance - used to calculate the resistance paths to GND

dynamic-high resistance - used to calculate the resistance paths to Vdd

In order to properly calibrate the simulator, the static resistance can be estimated from actual or SPICE measurements of the low threshold for standard logic gates. The dynamic resistances can be estimated by measuring high and low-going transition times from SPICE simulations and choosing the characteristic resistances to give an R-C time constant equal to the time the actual waveform takes to cross the desired threshold. These values are then placed in a file called rnl.co which resides in the users lib directory on the Metheus.

APPENDIX II

Design Schematics and Simulations (Hg) for the
Integrated Control Unit

AII.1 Introduction

This Appendix includes two parts. Section I contains all the schematics and Hg simulations for the dedicated processor chip. The simulations are derived from the layouts of the attached schematics. In this way it is certain that the schematic as well as the layout behave in the manner desired. The second section contains all the schematics and Hg simulations required for the address generator chip.

AII.2 Dedicated Processor Chip

The schematics and simulations have been arranged in the following order;

1) Systems Interface and Control Logic

Systems Mode Logic - figures AII.1, AII.2

Initial Counter Clocks - figures AII.3, AII.4

N Counter Control Circuitry - figures AII.5, AII.6

S/H Counter Control Circuitry - Figures AII.7, AII.8

ii) Clock Generator

Clock Generator Circuitry - Figures AII.9, AII.10

iii) Programmable Counters

N Counter - figures AII.11, AII.12, AII.13

RAM Counter - figures AII.14, AII.15, AII.16

S/H Counter - figures AII.17, AII.18

CHAN Counter - figures AII.19, AII.20

HMST Counter - figures AII.21, AII.22

ADC Counter - figures AII.23, AII.24

Figure AII.1 System node logic schematic.

Figure AII.2 is simulation of system node logic.

Figure AII.3: Variable logic clocks (Initial counter clocks) schematic.

Figure AII.4 H₃ simulation of variable logic clocks.

Figure A11.5 Delay (N) counter control logic schematic.

Figure A11.6 H₃ simulation of the I counter control logic.

Figure AII.3 13 simulation of SA4 counter control logic.

Figure AII.10 H_g simulation of clock generator circuit.

Figure A12.11 Delay 10 counter schematic.

Figure AII.12 d; simulation of 4 counter (high order bits).

Figure A11.13 4-bit simulation of a counter (low order bits).

Figure AII.14 RAM counter schematic.

Figure AII.15 is simulation of RAM counter (high order bits).

Figure AII.16 H₃ simulation of RAM counter (low order bits).

Figure AII.17 3/4 (SAM) counter schematic.

Figure AII.13 H₃ simulation of SAM counter.

Figure AII.24 d; simulation of ADC counter.

Figure AII.20 4: simulation of CHAN counter.

Figure A11.21 4MST counter schematic.

Figure AII.22 - d; simulation of HMSF counter.

Figure A11.24 H; simulation of ADC counter.

AII.3 Address Generator Chip

The schematics with respective simulations have been arranged in the following order;

i) Control Logic

Address Counter Control Circuitry - figure AII.25, AII.26

Address Roll-Over Clock - figures AII.27, AII.28

ii) I/O Logic - figures AII.29, AII.30

iii) Multiplexer - figures AII.31, AII.32

iv) Address Counters

AD0-AD11 Counter - figures AII.33, AII.34, AII.35

Pseudo Random Sequence Generator - figures AII.36, AII.37

A12-A15 Counter - figures AII.38, AII.39

Figure AII.25 Address counter control logic schematic.

Figure VII.25 Ig simulation of address counter control logic.

Figure A11.27 MSB address and CHAN counter clock logic (address roll-over) schematic.

Figure VII.28 H₃ simulation of address roll-over clock logic.

Figure A11.29 Input/Output (I/O) logic schematic to driver.

Figure A11.30 d; simulation of I/O logic (5 bits).

Figure A11.31 Address/Data (A/D) multiplexer schematic.

Figure AII.32 d_3 simulation of A/D multiplexer.

Figure M1.33 AD0-AD11 (address/data) counter schematic.

Figure ALL-34. Hg simulation of AD0-AD11 counter in sequential sequence mode.

Figure AII.35 Hg simulation of AD0-AD11 counter in random sequence mode.

Figure AII.36 Pseudo random sequence generator schematic.

Figure A11.37 H₃ simulation of pseudo-random sequence generator.

Figure A11.33 MSB address (AD12-AD15) counter schematic.

Figure AII.39 d₃ simulation of MSB address counter.

APPENDIX III

Physical Description of Fabricated and Tested I.C.s

AIII.1 Introduction

This chapter is divided into three sections, each discussing one of the chips presented in this thesis. The first section will describe the contents of the standard cell test vehicle as well as the pin out used to test the chip. The next two sections will focus on the pin outs for the dedicated processor chip and the address generator chip. Some of the techniques used to test the various designs will also be presented.

AIII.2 Standard Cell Test Vehicle

The CMC multiproject bonding diagram received from Northern Telecom is shown in figure AIII.1. The general contents of the cell as well as the pinout for the I.C. can be seen. The general pin assignment for the chip is as follows;

Vdd - pin 15

Vss - pin 35

Inputs (16) - pins 1, 2, 8, 12, 14, 17, 19, 21, 23, 25, 27,
28, 30, 33, 39, 40

Outputs (21) - pins 3, 4, 5, 6, 7, 9, 10, 11, 13, 16, 18, 20,
22, 24, 26, 29, 32, 34, 36, 37; 38

Not used (1) - pin 31

The inputs are shared between several cells in order to maximize the number of outputs. A brief description of the

REF ID: A4

OTHER IDENTIFICATION FEATURES

WAFER NUMBERS

DESIGN FILE REFERENCE

PACKAGE	IDK 4 OFI-192G		LID C-193-175-35M					
WIRE ALLOY	99% Al/1% Si	DIA.	.001"	ELONG.	1.5 - 4%	T.S.	14-16 gms	
	99% Al/1% Si		.00125"		1.5 - 4%		18-22 gms	
D/A PREFORM	ALLOY 98% Au/2% Si		RECOMMENDED SIZE		W/B METHOD			U.S.

BONDING DIAGRAM

- NOTES: 1. DIE ATTACH PAD SIZE: .000 X .000
2. ZERO GROUND

CANADIAN MICROELECTRONICS CORPORATION
SINCE 1968

CMC - VLSIIC 84-08

Figure AIII.1 CMC multiproject bonding diagram for standard cell test vehicle.

individual cells with their respective pin assignment is now given:

- masterJK flip-flop with direct clock input. Q and Qbar are fed back to J and K inputs and Q is also buffered to output pad.

clock input - pin 17

Q output - pin 18

- masterJK flip-flop with buffered clock input. Q and Qbar are fed back to J and K inputs and Q is buffered to output pad.

clock input - pin 19

Q output - pin 20

- D flip-flop with positive edge trigger. Uses a direct clock input and buffers the Q output to a pad. The D, setbar, and resetbar (rstbar) inputs are all accessible.

clock input - pin 21

D input - pin 17

setbar input - pin 19

rstbar input - pin 23

Q output - pin 22

- D flip-flop with positive edge trigger. Uses a buffered clock input and Q is fed back to the D input. The setbar and rstbar inputs are accessible and the Q output buffered out.

clock input - pin 23

setbar input - pin 21

rstbar input - pin 25

Q output - pin 24

- Level sensitive D flip-flop with setbar and rstbar. All inputs are accessible and Q output is buffered to an output pad.

clock input - pin 23

D input - pin 25

setbar input - pin 28

rstbar input - pin 27

Q output - pin 26

- D flip-flop latch using transmission gates. This is a simple static D flip-flop latch made with two inverters and two transmission gates. The clock, D and Q outputs are all accessible.

clock input - pin 27

D input - pin 28

Q output - pin 29

- The 4 bit counter is the tall structure near the center of the I.C. It is a simple ripple counter using the level sensitive D flip-flop with set and reset capabilities. Only one output is available from the circuit and it is the Q from the last bit.

clock input - pin 30

setbar input - pin 28

4th bit Q output - pin 7

- Inverter string with inverta (a size inverter) and invertb (o size inverter). Inverta is a large 3 grid size inverter, and invertb is a large 4 grid size inverter.

input - pin 14

output - pin 13

- Inverter string with inverter1 $[(W/L)_n=1]$, inverter2 $[(W/L)_n=2]$, inverter3 $[(W/L)_n=4]$

input - pin 14

output - pin 11

- Passgate or transmission gate with simple feed through from input pad to output pad. Both enable and enablebar are also accessible.

input - pin 12

enable - pin 14

enablebar - pin 8

output - pin 10

- Pass3gate is a 3 transistor analog transmission gate. It is configured in the same way as the previous passgate.

input - pin 12

enable - pin 14

enablebar - pin 8

output - pin 9

- In order to test the input and output pad devices, an input pad was directly fed to an output pad.

input - pin 33

output - pin 32

- A medium size inverter with $(W/L)_n=8$ and $(W/L)_p=13$

input - pin 2

output - pin 34

- 2 input NAND gate

A input - pin 39

B input - pin 40

output - pin 38

- 3 input NAND gate

A input - pin 1

B input - pin 39

C input - pin 40

output - pin 37

- 4 input NAND gate

A input - pin 2

B input - pin 1

C input - pin 40

D input - pin 39

output - pin 36

- 2 input NOR

A input - pin 1

B input - pin 2

output - pin 3

3 3
OF / DE

- 3 input NOR

A input - pin 1

B input - pin 2

C input - pin 40

output - pin 4

- 4 input NOR

A input - pin 1

B input - pin 2

C input - pin 39

D input - pin 40

output - pin 5

- 2 input multiplexer implemented using NAND gates.

A input - pin 40

B input - pin 1

enable input - pin 39

output - pin 6

Notice that several circuits use the same input pins in order to reduce the pin count. The chips could be tested by tying all irrelevant inputs to V_{ss} and strobing the inputs of interest with a wavetec generator. The outputs are then monitored on an oscilloscope. By increasing the input frequency and monitoring it along with the output signal, propagation delays can be measured. The SPICE and Hg simulations of most of the above primitives can be found in the standard cell manual.

AIII.3 Dedicated Processor Chip

The CMC multiproject bonding diagram received from Northern Telecom is shown in figure AIII.2. The general pin assignment for this chip is;

Vdd - pin 20

Vss - pin 40

Inputs (24) - pins 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 16, 17,
18, 19, 21, 22, 23, 24, 25, 36, 37, 38, 39.

Outputs (14) - pins 11, 12, 13, 14, 26, 27, 28, 29, 30, 31, 32,
33, 34, 35.

The pin names are as follows;

Inputs: - data bus (D0-D11) - pins 1-8, 36-39

- 50MHz clock - pin 9

- system mode logic A - pin 10

B - pin 15

- trigger pulse - pin 18

- trigger select, Pre-trig. - pin 16

post-trig. - pin 17

- EOC from ADC - pin 19

- initial counter clocks, A - pin 21

B - pin 22

C - pin 23

CLK - pin 24

- channel counter clock - pin 25

Outputs: - RSE - pin 11

RECEIVE CODE 10044 7.
 OTHER IDENTIFICATION FEATURES
 WAFER NUMBERS
 DESIGN FILE REFERENCE

PACKAGE	<u>IDK 10FI-192G</u>		LID <u>C-193-175-35M</u>				
WIRE ALLOY	<u>99% Al/1% Si</u>	DIA.	<u>.001"</u>	ELONG.	<u>1.5 - 4%</u>	T.S.	<u>14-16 gms</u>
	<u>99% Al/1% Si</u>		<u>.00125"</u>		<u>1.5 - 4%</u>		<u>18-22 gms</u>
D/A PREFORM	ALLOY <u>98% Au/2% Si</u>		RECOMMENDED SIZE		W/B METHOD		<u>U.S.</u>

BONDING DIAGRAM

NOTES: 1. DIE ATTACH PAD SIZE: .400 X .400
 2. ZERO GROUND

Figure AIII.2 CMC multiprocessor bonding diagram for the dedicated processor chip.

- Test Mode (TM) - pin 12
- 25MHz master clock - pin 13
- LVE - pin 14
- SSE - pin 26
- IAC - pin 27
- CNC - pin 28
- Test pin from initial counter clocks - pin 29
- End - pin 30
- RAM pulse - pin 31
- ADC pulse - pin 32
- S/H pulse - pin 33
- RAM pulse - pin 34
- HT pulse - pin 35

As mentioned earlier, this chip was tested at O.C.M. using an HP 8000 series test station. Probes from the station could be placed on all the inputs from the 8180A data generator and all the outputs to the 8182A data analyzer. All the inputs are made low except for those to the circuit being tested. Because of the design error in the initial counter clocks logic, all inputs to it had to remain low. A 100 ohm resistor was placed on the Vdd input to limit the current into the chip to 50 mamps. This would prevent total destruction of the chip in the event of a short. All the internal counters could not be tested due to the inability to initialize them.

AIII.4 Address Generator Chip

The CMC multiproject bonding diagram for this chip is shown in figure AIII.3. The general pin assignment is;

Vdd - pin 20

Vss - pin 40

Inputs (17) - pins 1, 2, 3, 4, 5, 6, 7, 15, 31, 32, 33, 34,
35, 36, 37.

Outputs (21) - pins 8, 9, 10, 11, 12, 13, 14, 16, 17, 18,
19, 21, 22, 23, 24, 25, 26, 27, 28, 29;
30.

The pin names are;

Inputs: - data/test bus - pins 1-7, 35-39

- IAC - pin 15

- LVS - pin 31

- RSE - pin 32

- RAM pulse - pin 33

- Test Mode (TM) - pin 34

Outputs: - address/data bus - pins 8-14, 16-19, 21

- MSB address bus - pins 22- 25

- CHAN counter clock - pin 26

- test pin for IAC - pin 27

- random generator clock - pin 28

- random generator bit 0 - pin 29

- address counter clock - pin 30

This chip was also tested at O.C.M. using the HP test station. Pin 34, TM must remain low during testing due to an

CMC MULTIPROJECT BONDING DIAGRAM

BRAND ID LABEL

ARTICLE CODE ROE A6CCPV4

OTHER IDENTIFICATION FEATURES _____

WAFER NUMBERS _____

DESIGN FILE REFERENCE _____

PACKAGE	<u>IDK40F1-192G</u>		LID	<u>C-193-175-35M</u>	
WIRE ALLOY	<u>99% Al/1% Si</u>	DIA.	<u>.001"</u>	ELONG.	<u>1.5 - 4%</u>
	<u>99% Al/1% Si</u>		<u>.00125"</u>		<u>1.5 - 4%</u>
T.S.					<u>14-16 gms</u>
					<u>18-22 gms</u>
D/A PREFORM	ALLOY <u>98% Au/2% Si</u>		RECOMMENDED SIZE	W/B METHOD <u>U.S.</u>	

BONDING DIAGRAM

- NOTES: 1. DIE ATTACH PAD SIZE: .400 x .400
2. ZERO GROUND

Figure AIII.3 CMC multiproject bonding diagram for the address generator chip.

error in the I/O logic. The data generator and analyser probes were again used to stimulate and monitor the chips performance.

END

1 7 0 4 8 6

FIN