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.

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.

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

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

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses
on Microfiche Service

Services des thèses canadiennes
sur microfiches

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

Ralph Dickson Mason

Date of Birth — Date de naissance

Nov. 11, 1961

Country of Birth — Lieu de naissance

Canada

Permanent Address — Résidence fixe

491 Stellarton Road
New Glasgow N.S.
B2H 1M6

Title of Thesis — Titre de la thèse

AN ADVANCED WORKSTATION ENVIRONMENT FOR CMOS VLSI
DESIGN AND THE IMPLEMENTATION OF A MULTI-MODE
DATA ACQUISITION MEMORY CONTROL UNIT

University — Université

Carleton University

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

Master of Engineering

Year this degree conferred — Année d'obtention de ce grade

August, 1985

Name of Supervisor — Nom du directeur de thèse

C. H. Chan

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies of
the film.

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission

L'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et de
prêter ou de vendre des exemplaires du film

L'auteur se réserve les autres droits de publication, ni la thèse
ni de longs extraits de celle-ci ne doivent être imprimés ou
autrement reproduits sans l'autorisation écrite de l'auteur

Date

August 16, 1985

Signature

Ralph Dickson Mason

AN ADVANCED WORKSTATION ENVIRONMENT FOR CMOS VLSI DESIGN

AND THE IMPLEMENTATION OF A MULTI-MODE DATA ACQUISITION

↑
MEMORY CONTROL UNIT

by

Ralph D. Mason, B. Sc.

A thesis submitted to the Faculty of Graduate Studies and
Research in partial fulfillment of the requirements for a
degree of

Master of Engineering

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:

"An Advanced Workstation Environment for CMOS VLSI Design
and the Implementation of a Multi-Mode Data Acquisition
Memory Control Unit"

submitted by Ralph D. Mason in partial fulfillment of the
requirements for the degree of Master of Engineering.

Thesis Supervisor

Chairman, Dept. of Electronics

Carleton University

August 1985

ABSTRACT

The purpose of this thesis is to develop within the university an environment for the design of complex integrated circuits. As a test vehicle for this environment the design of a memory controller unit for a multi-mode data acquisition system was carried out. The design environment is centered around an advanced IC design workstation and uses a standard-cell approach. The memory controller unit is a 2900 transistor chip integrated in 5u CMOS on a $(5000 \mu)^2$ die and is part of a three chip set data acquisition system.

ACKNOWLEDGEMENTS

The author wishes to thank with deep appreciation his program supervisor Dr. C. H. Chan for his guidance and support during the work on this thesis.

The author is also indebted to Richard Vallee with whom the work on the standard cells and data acquisition system was carried out, to Carlos Pon for his work on the original breadboarded system, to Michael Gagnon for his aid in testing the memory controller, and to all the technical and support staff in the electronics department.

The author would also like to acknowledge the contributions of the Canadian Microelectronics Corporation and Northern Telecom Limited in terms of advanced design tools and fabrication facilities, and the Natural Sciences and Engineering Research Council of Canada for its funding through the mobile radio strategic grant.

Finally I would like to thank my wife Anne Marie for her support and help during my studies.

TABLE OF CONTENTS

	<u>PAGE</u>
CHAPTER 1 INTRODUCTION	
1.1 Motivation for Thesis	1
1.2 Background Information	1
1.3 Thesis Objective	3
1.4 Thesis Organization	4
CHAPTER 2 DESIGN METHODOLOGY	
2.1 Introduction	6
2.2 General Overview	7
2.3 VLSI Design	
2.3.1 Overview	12
2.3.2 Standard Cell Approach	15
2.3.3 Cell Format	19
2.3.4 Example Cell	22
2.4 Summary	24
CHAPTER 3 DESIGN TOOLS	
3.1 Introduction	30
3.2 Corvus Concept System	32
3.3 Honeywell CP-6	35
3.4 Metheus System	
3.4.1 Overview	37
3.4.2 Logic Design Tools	40
3.4.2 Physical Design Tools	42
3.4.3 Netlist Verification Tools	43
3.4.4 Design Case Study	45
3.5 Test Tools	54
3.6 Summary	55
CHAPTER 4 SYSTEM DESIGN	
4.1 Introduction	58
4.2 Device Requirements	
4.2.1 Multi-Mode Data Acquisition System	62
4.2.2 Memory Controller Unit	65
4.3 System Overview-Functional Block Diagram	65
4.4 Synchronizer and Scan Circuits	74
4.5 Functional Block Design	87
4.6 Fabrication and Testing	98
4.7 Summary	98

TABLE OF CONTENTS (continued)

	<u>PAGE</u>
CHAPTER 5 TEST RESULTS	
5.1 Introduction	101
5.2 Standard Cell Test Vehicle	102
5.3 Memory Controller Unit	105
5.4 Summary	129
CHAPTER 6 CONCLUSION	
6.1 Summary	133
6.2 Future Work	135
REFERENCES	138
APPENDIX I Software References	140
APPENDIX II Functional Block Designs	147
APPENDIX III CMOS Flip-Flop Analysis	159

LIST OF FIGURES

<u>FIGURE NO.</u>		<u>PAGE</u>
2.1	Chip Design Methodology	9
2.2	Standard Cell Hierarchy	17
2.3	Standard Cell Layout Rules	21
2.4	3 - Input NOR Layout	23
2.5	3 - Input NOR SPICE Simulation	25
2.6	3 - Input NOR Hg Simulation	26
2.7	3 - Input NOR Schematic	27
2.8	3 - Input NOR Load vs. Delay, Logic, and Block Symbol	28
3.1	Design Tools Network	31
3.2	Metheus Design Environment	39
3.3(a)	Nand2 Logic Diagram	41
3.3(b)	Nand2 Schematic Diagram	41
3.4	Functional Block Design Flowchart	46
3.5	2-to-4 Decoder Schematic	47
3.6	2-to-4 Decoder Schematic Netlist	48
3.7	2-to-4 Decoder Hg Simulation	50
3.8	2-to-4 Decoder Layout	51
3.9	2-to-4 Decoder Layout Netlist	52
3.10"	2-to-4 Decoder Netlist Comparison	53
4.1	Digital Waveform Recorder System Configuration	59
4.2	Digital Waveform Recorder System Architecture	60
4.3	General Overview of Device Requirements	64
4.4	Memory Controller Functional Block Diagram	66

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
4.5	Upcounter Schematic	68
4.6	Downcounter Schematic	70
4.7	System Timing Diagram	72
4.8	Sync1 Schematic	75
4.9	Sync1test Schematic	78
4.10	Sync1test SPICE Simulation 1	79
4.11	Sync1test SPICE Simulation 2	80
4.12a	DRAM Timing	81
4.12b	Delay SPICE Simulation	83
4.13	Scan Circuit Schematic	86
4.14	Johnson Counter Schematic	88
4.15	Johnson Counter Hg Simulation	89
4.16	Latct4 Schematic	91
4.17	Latct4 Hg Simulation	92
4.18	Sync1 Hg Simulation	94
4.19	Scan Circuit Hg Simulation	96
4.20	Memory Controller Chip Floor Plan	99
5.1	Standard Cell Test Vehicle Chip	103
5.2	Memory Controller Chip	106
5.3	Dat 1 Test Results	108
5.4	Dat 1 Hg Simulation	109
5.5	Dat 2 Test Results	110
5.6	Dat 2a Hg Simulation	112
5.7	Dat 2b Hg Simulation	113
5.8	Dat 3 Input Test Signals	115

LIST OF FIGURES (continued)

<u>FIGURE NO.</u>		<u>PAGE</u>
5.9	Dat 3 Output Test Signals	116
5.10	Dat 3 Hg Simulation	117
5.11	Dat 4 Input Test Signals	118
5.12	Dat 4 Output Test Signals	120
5.13	Dat 4 Hg Simulation	121
5.14	Dat 5 Test Results	122
5.15	Dat 5 Timing Delays	123
5.16	Dat 5 Hg Simulation	125
5.17	Dat 6 Input Test Signals	126
5.18	Dat 6 Output Test Signals	127
5.19	Dat 6 Hg Simulation	128
5.20	Dat 7 Test Results	130
5.21	Dat 7 Hg Simulation	131
AII.1	Latlat Schematic	148
AII.2	Latlat Hg Simulation	149
AII.3	Muxmux Schematic	151
AII.4	Muxmux Hg Simulation	152
AII.5	Dectest Schematic	154
AII.6	Dectest Hg Simulation	155
AII.7	Dec2p Schematic	157
AII.8	Dec2p Hg Simulation	158
AIII.1	Synctest Schematic	160
AIII.2	CMOS Flip-Flop Schematic	161
AIII.3	Flip-Flop dc Bias.	164

LIST OF SYMBOLSSECTION

A	synchronizer aperture time	4.4
A(0-15)	Address inputs	4.5, 5.3
ac	Alternate Current	3.4, APP1
A/D	Analog to Digital Converter	1.2, 4.1, 4.2, 4.3, 4.5
ATG	Automatic Test Generation	3.4, APP1
b	complement of signal	4.5, 5.2, 5.3
B(0-15)	internal counter outputs	5.3
BJT	Bipolar Junction Transistor	APP1
CAD	Computer Aided Design	1.2, 3.3
CAS	Column Address Strobe	4.5, 5.3, 4.4
CC.DCOMM	Corvus Communications Program	3.2, APP1
CDEC	block size latch clock	4.5, APP2
CDIS	synchronous port latch clock	4.5, APP2
CIF	Caltech Intermediate Form	1.2, 2.2, 3.2 APP1
CIFDRAW	Corvus Layout Program	3.2, 3.4, APP1
CIF2PHL	Translation From CIF to Metheus Layout	3.4, APP1
CK	Clock	4.5, 5.3, APP2
CMC	Canadian Microelectronics Corporation	1.2, 1.3, 4.6, 2.3, 5.1, 6.1, 6.2
CMOS	Complementary Metal Oxide Semiconductor	1.2, 3.2, APP1 APP3
COMPARE	Metheus Netlist Comparison Routine	3.4, APP1
CPU	Central Processing Unit	3.2, 3.4, 4.1
CRW	asynchronous port latch clock	4.5
CT(0-12)	internal counter outputs	4.5, APP2
C(0,2,6)	johnson counter outputs	5.3

LIST OF SYMBOLS (continued)

SECTION

D	synchronizer delay time	4.4
D/C	Digital to Analog Converter	4.1, 4.2
dc	Direct Current	3.4, 5.3, APP1
DIS	synchronous port latch output	APP2
DEC	block size latch output	APP2
DRAM	Dynamic Random Access Memory	4.2, 4.3, 4.5, 4.4, 5.3, APP2
DRC	Design Rule Checker	2.3, 3.2, 3.4
DSP	Digital Signal Processing	2.3
dt	small change in time	4.4, APP3
DVM	Digital Volt Meter	1.2, 1.3
e(exp)	exponent (2.71)	4.4
ENA8	Enable lower order bits of 8 bit mux	4.5, 5.3, APP2
ENA16	Enable synchronous bits of 16 bit mux	4.5, 5.3, APP2
EOC	End of Conversion	4.3, 4.5, 4.4, 5.3
ERC	Electric Rule Checker	3.4, APP1
FPS	Floating Point Processor	3.1, 3.3, 3.4
GND	Ground	2.3
GPIB	General Purpose Interface Bus	3.4
Hg	Switch Level Simulator	2.3, 3.4, 3.6, 4.7, 5.1, 5.3, APP1
HILO	gate level simulator	2.2, 3.1, 3.4, APP1
HP	Hewlett Packard	1.2, 2.2, 3.1, 3.2, 3.4, 3.5, 5.1, 5.3, APP1

LIST OF SYMBOLS (continued)SECTION

IC	Intergrated Circuit	1.1, 1.2, 1.3, 2.1, 2.3, 3.1, 3.4, 3.5, 3.6, 4.4, 6.1, 6.2, APP1
I_g	Gate current	APP3
IN(0-5)	test circuit parallel inputs	4.5
I/O	Input/Output	2.3, 3.4, 4.1, 6.2, 4.4
JFET	Junction Field Effect Transistor	APP1
K	Kilo (Thousand)	1.2, 3.2, 3.5 4.1, 4.2, 4.3 4.5, 4.4, APP2 APP3
KHz	thousand cycles per second	4.1, 4.3, 4.5, 5.3
L	transistor Length	APP3
LAN	Local Area Network	3.1
LCD	Liquid Crystal Display	4.6, 6.1
LD(1d)	Load	4.5, APP2
LE045	Metheus Design Rule Checker	APP1
LSI	Large Scale Integration	1.3
M	Million	3.2, 3.4, 3.5 3.6
M3	transistor M3	APP3
M4	transistor M4	APP3
MFLOPS	Million Floating Point Operations	3.1
MHz	Million cycles per second	4.1, 4.3, 4.5, 5.2, 5.3
MOS -	Metal Oxide Semiconductor	3.4, APP1
MOSFET	Metal Oxide Semiconductor Field Effect Transistor	APP1

LIST OF SYMBOLS (continued)SECTION

ms	millisecond	4.3, 4.5, 4.4
MSI	Medium Scale Integration	1.3, 3.2
n	n-channel Transistor	2.3, 6.2
NAND	Logic Function	3.4, 3.6, APP2
NMOS	N-Channel Metal Oxide Semiconductor	4.1, APP1
NOR	Logic Function	2.3, 3.4, APP2
ns	nanosecond	4.3, 4.5, 5.2, 5.3, 4.4
op-amp	Operational Amplifier	3.2
ORR	Output for Read and Refresh	4.5
ORW	Output for Read and Write	4.5
OSWRR	Output of Switch for Read and Refresh	4.5
OSWRW	Output of Switch for Read and Write	4.5
OUT(0-7)	Output addresses	5.3
p	p-Channel Transistor	2.3, 6.1
PHLED	Physical Layout Editor	3.4, APP1
PHLEX	Physical Layout Extractor	3.4, 3.6, 5.1, APP1
PHLPLOT	Physical Layout Plotter	3.4, 3.6, APP1
PHL2PHL	Physical Layout Scaling Routine	3.4, APP1
PSRR	Power Supply Rejection Ratio	2.2
RAM	Random Access Memory	2.3, 3.2, 4.1 6.2, 4.4
RAS	Row Address Strobe	4.5, 5.3, 4.4
ROM	Read Only Memory	1.2, 2.3, 4.1 6.2
R/REF	synchronous memory access signal	4.5, 4.4, APP3
RSTN	johnson counter reset	4.5

LIST OF SYMBOLS (continued)SECTION

RS-232	serial transmission port	3.2, 3.4
RW(0-15)	asynchronous port latch outputs	APP2
RWN	Read/Write Not	APP3
R/WIN	Read/Write Input Select	4.3, 4.5, 5.3
R/WOUT	Read/Write Output	5.3, 4.4
RW(0-15)	asynchronous port latch outputs	4.5
s	seconds	4.4, APP3
SCHEDIT	Metheus Schematic Editor	3.4, APP1
SCHPLOT	Metheus Schematic Plotter	3.4, APP1
SCH2IMF	Schematic to Intermediate Form	3.4, APP1
SCH2HILO	Schematic to HILO	3.4, APP1
SCH2SIM	Schematic to Simulation	3.4, APP1
S/H	Sample and Hold	4.1, 4.2
SIMGR	Metheus Simulation Grapher	3.4, APP1
SPICE	Transistor Level Simulator	2.3, 3.2, 3.2, 3.4, 3.6, 5.3, APP1, APP3
SPLICE	Mixed Mode Simulation Program	3.2, 3.3, APP1
SVS	Silicon Valley Software	3.2
SYMED	Metheus Symbolic Editor	3.4, APP1
SYMPLOT	Metheus Symbolic Plotter	APP1
t	time	4.4, APP3
T	Toggle	4.3, 4.5
TEST	Test mode enable	4.5, 5.3, 4.4
TESTCK	Test circuit Clock	4.5, 5.3
TESTIN	Test circuit Input	4.5, 5.3, 4.4

<u>LIST OF SYMBOLS</u> (continued)		<u>SECTION</u>
TESTOUT	Test circuit Output	4.5, 5.3, 4.4
t_{hold}	hold time	4.4
T_s	Saturation Transit Time	4.4, APP3
TTL	Transistor Transistor Logic	2.3
μ	micro	1.2, 2.3, 4.6 6.2, APP2
μ	mobility	APP3
μp	microprocessor	4.1
μs	microsecond	4.3, 4.5, 4.4 5.2
μw	microwatt	5.3
V	Voltage	APP3
V_{dd}	positive supply voltage	2.3, APP3
V_{ds}	Drains-Source Voltage	APP3
VLSI	Very Large Scale Integration	2.1, 2.3, 2.3, 2.4, 3.2. APP1
V_t	Threshold Voltage	APP3
V_{tn}	N channel transistor Threshold Voltage	APP3
V_{tp}	P channel transistor Threshold Voltage	APP3
ZERO	zero detect circuit	APP2
3M16	16 bit multiplexer output	APP2
5m8	8 bit multiplexer output	APP2
4m8	8 bit multiplexer output	APP2

CHAPTER 1

INTRODUCTION

1.1 Motivation for Thesis

The motivation for this thesis was to gain experience in Integrated Circuit (IC) design at both the circuit and system level and to take this gained knowledge and incorporate it into a design environment. The design environment should enable future faculty and students with little experience in IC design, to begin designing complex chips without spending a great deal of time learning all of the details involved. It should also aid more experienced designers by giving them a structured method for their designs. In addition, this work could offer a tool for graduate course work.

1.2 Background Information

Until recently, the ability for students to gain experience in integrated circuit design within the university was rather limited. Due to the high cost of fabrication facilities and Computer Aided Design (CAD) tools this would have remained so if it was not for the contribution of non-university resources.

The formation of the Canadian Microelectronics Corporation (CMC) by the federal government in 1982 has since permitted

students to do very advanced integrated circuit design. CMC provides eligible universities with a choice of advanced CAD tools, such as the Metheus lamoda 750 workstations and HP 8000 series test stations. Through Northern Telecom Ltd., CMC makes available a 5u double poly, single metal CMOS process and will soon offer Northern Telecom's 3u CMOS process.

IC designs are fully implemented within the respective universities, which then send the Caltech Intermediate Form (CIF) files to the implementation center at Queen's University, Kingston, Ontario, where the designs are combined to form multiproject chips, following the Mead and Conway [1] approach. The designs are then sent to Northern Telecom for mask making and fabrication. Upon completion of fabrication a number of prototypes are then sent to the designer for testing. The complete turnaround time from submission to receipt of prototypes is four months.

Since the formation of CMC, a number of designs have been fabricated for Carleton University. In the first submission two chips implementing different Analog to Digital (A/D) conversion techniques were designed. As part of a graduate course offered by the department of electronics three designs were sent to CMC for the second submission. These included an adaptive echo cancellation unit and high speed phase lock loop building blocks, as well as a single chip Digital Volt Meter (DVM). The third submission included two test vehicles. The first was composed of the standard cells developed at Carleton, while the other was

used to evaluate a number of Read Only Memory (ROM) architectures for a residue number generator chip. In the fourth submission four chips were submitted. These included a three chip set for the multi-mode data acquisition system and a complete 2K ROM for the residue number generator. The last submission included three chips. Two of the multi-mode data acquisition chips were resubmitted to correct some minor errors and a chip was submitted by students in the graduate IC design course which included a lowpass and bandpass filter.

For the two initial submissions the turnaround time was between eight to twelve months, with the result that the student(s) had completed their studies before receiving the prototypes and therefore none of the silicon chips was tested. However the people at CMC have been working hard to improve their service and the last three submissions have consistently had a four month turnaround time.

1.3 Thesis Objective

After the submissions in April 1984 it became obvious that some design strategy was necessary to design chips of the complexity of a DVM or echo cancellation unit and expect any success. The sheer size of the project requires that all people involved use the same design procedures as well as having close communications throughout the design period.

The objective of this thesis was to develop an environment to meet these needs, as well as those of the inexperienced designer, and to demonstrate its validity by designing a complex system within this environment.

The provision of advanced design tools by CMC was essential in the development of a workable design environment. However, they are by no means the only requirement. In order to have an environment where professors, students, and people from industry with varying degrees of experience can design IC's, a standard cell approach was chosen. In addition to choosing an appropriate design methodology the tools to be used had to be configured for the environment

The memory controller chip is based partly on some work done by a previous M. Eng student at Carleton University [2]. As fulfillment of his thesis the design and characterization of a digital waveform recorder was carried out. The overall system was composed of a large number of LSI and MSI chips occupying an 11 X 8 inch printed circuit board. The integration of a large part of this system was to be carried out by Richard E. Vallee and the author.

1.4 Thesis Organization

The thesis is divided up into two main sections. The first section is on the design environment with chapter 2 covering the design methodology and chapter 3 covering the design tools. The

second section which covers the actual design of the memory controller unit includes chapter 4 on the system design and chapter 5 on the test results obtained. Finally in chapter 6 conclusions are summarized and recommendations for future work are given.

CHAPTER 2

DESIGN METHODOLOGY

2.1 Introduction

The design of a VLSI chip can be summed up in one word as "complex". The design covers a large number of stages all the way from initial conceptualization to a fully functional chip. In order for any individual or groups of individuals to succeed they must have (a) a clear and concise picture of how to get from the start to the finish and (b) a large number of both human and machine resources. In this chapter the author will deal with the first of these requirements and hopefully give the reader some insight into what is involved and the methods used at Carleton to realize this goal.

In the past, the design of IC's was limited to a relatively small group of individuals. Each of these developed their own ideas and methods on how to achieve the end product of a working chip. However as the technology matures it becomes increasingly evident that this approach is unacceptable. There are two main factors responsible for this. As the size of chips are increased and feature sizes decreased the complexity of the designs is therefore increasing. This is abundantly evident if one considers the evolution of the microprocessor. The first micro's were four bit versions of several hundred transistors

performing basic control functions. The present day versions are 32 bits with advanced features such as on-chip cache memory and memory management capabilities, and contain anywhere up to half a million transistors. As the complexity of the chips increases the number of people and time involved in the design also increases. Added to this is the desire to make the technology available to larger groups of people who have little or no experience in IC design.

At Carleton we have overcome both of these problems. The choice of a standard cell approach enables the designer to isolate him or herself from transistor level considerations while not sacrificing too much of the performance available in a full custom approach, and is more amenable to a fixed design method where all designers access the same data base (standard cells). The standard cells are incorporated into a methodology where each stage of the design is performed in a fixed order to bring the designer from start to finish while minimizing both design time and possible design errors. The last two considerations are particularly relevant in a university where students have a limited time to finish their projects and generally cannot afford the luxury of even one iteration.

2.2 General Overview

As stated in the previous section, the design methodology should bring the designer from the initial conceptualization all

the way to the final working product. Our design method can be broken down into the following stages [3],[4].

- (1) Device Requirements
- (2) System Overview and Functional Block Diagram
- (3) Functional Block Design and Simulation -
Chip Floor Plan
- (4) Layout and Resimulation
- (5) Fabrication
- (6) Testing

The first stage is the specification of device (where device refers to the overall system) requirements (see figure 2.1). It is important that a complete understanding of what is required be reached to avoid problems at a later stage.

The device requirements should spell out the pin level interactions, the functionality needed from the device, and some idea of speed of operation. Other factors such as pull-ups on inputs and specific signals on a selected pin should also be stated. For analog circuitry things such as Power Supply Rejection Ratio (PSRR), noise levels, and reliability should be included.

Another important consideration is an overall testing strategy. Precautions such as tristating outputs, multiplexing input and output pads for normal and test modes, and adding reset capabilities to registers and counters will be required. If some sort of scan chain testing is to be used in the system, this

CHIP DESIGN METHODOLOGY

figure 2.1 Chip Design Methodology

needs to be stated. Auxiliary test points at the internal nodes of a chip, particularly, in the analog sections could ease the prove-in of a prototype.

Once the device requirements have been spelled out the designer is ready to proceed to the next stage where a system overview and functional block diagram will be developed. The VLSI designer must decide what will be inside the device. All functions should be specified and blocked out. All timing considerations and critical sections should be stated. The functionality of the device should be partitioned into subfunctions and those subfunctions into sub-subfunctions until the designer is satisfied that the design can be reasonably handled. This partitioning into functions with predefined interactions between them will simplify the design. It also allows the design process to be split up over a few designers.

The next stage after the system has been divided down to the functional block level is to design and simulate the individual blocks. At the same time as each block is being designed, a chip floor plan must be developed. The blocks are configured to provide for minimum interconnect between blocks as well as minimum area. If the number of blocks cannot be configured to the given die size as seen by the chip floor plan then it is necessary to rethink the system overview and/or device requirements. As an example, if the amount of circuitry cannot be fitted on a single chip, it may be necessary to reduce the total circuitry or use an additional chip. This would require a

repartitioning of the system with the new device requirements taken into account.

After the functional blocks are designed and chip floor plan completed the next stage is the actual layout of the functional blocks. Using the standard cell approach the necessary cells are butted together and connected in the intervening channels. The layout is completed hierarchically starting from smaller blocks, proceeding to larger blocks and finally completing the chip. At present this is done manually but automation of this process in the near future is imperative. While the blocks are being laid-out the electrical parameters from the layout can be extracted and a resimulation can be completed with the added interconnect capacitance included. If the functional blocks with added interconnect do not meet specifications it will be necessary to go back to the functional block design stage. The method used can be considered a top down design with a bottom up layout. Once the chip is laid out and resimulated it is ready for mask making and chip fabrication.

The layout is converted into CIF format and put on magnetic tape to be sent to the CMC VLSI Implementation Centre in Queens University. At Queens the submissions (which must use one of the four standard pad frames) from a number of universities are configured into a multiproject chip to insure efficient use of silicon. The chips are then sent to Northern Telecom for fabrication. At present the turnaround time is four months from submission of project to receipt of prototype chips.

Once the chips are received they are ready for testing and verification. At present, the test vector generation capabilities of the Metheus are not exploited as an interface between the HILO simulator and the HP test equipment is not available. However, the test vectors can be generated manually on the HP test equipment with a little patience.

In general, once the design is completed and tested it would require further modifications to correct any errors or to meet any tight specs that were not met on the first pass. At Carleton the full-time M. Eng. program operates on a two year basis. The first eight months are usually devoted to coursework. Of the remaining sixteen months it requires four to eight months to complete the initial formulation, design, and layout of the project, and four months for fabrication. This leaves very little time to complete the testing and documentation for the thesis. Due to these time constraints the aim is to generate working chips on the first try and to emphasize functionality more than system performance. For those students in a Ph. D. program or faculty members this constraint is relaxed and a more comprehensive design is possible.

2.3 VLSI Design

2.3.1 Overview

VLSI design fits into three broad categories which includes gate arrays, standard cells, and full custom [4]. The word gate

array is used to define designs in which the user reconfigures preprocessed wafers using only a few mask levels. Standard cell is used to imply the interconnection of predefined and pre-laid out function blocks. The full custom approach involves the interconnection of devices at the transistor level in a random manner to achieve maximum packing density.

Gate arrays offer one or two layers of interconnection and can be turned around in a few days to one week. Design with gate arrays is also relatively simple; many of the advanced engineering work stations have software to convert logic diagrams and netlists (network listings) into automatically routed gate-array designs. However, gate arrays also have a number of disadvantages: they are limited to about 5000 gates, although some 10000-gate products have been announced. Moreover, gate arrays are relatively inefficient in chip size, circuit speed, and power consumptions. The inefficiency in chip size, as much as a factor of 10 compared with handcrafted designs stems from two factors: first, gate arrays come only in a few sizes, so a designer who specifies, say, a 1005-gate circuit may have wasted 50 percent of the available space on a 2000 gate chip. Second, the gate array architecture can use a great deal of space to implement certain functions, such as memory cells. But if the quantity of chips required is small enough that fabrication is a relatively small part of the total cost and if the turnaround time is critical, gate arrays can do an excellent job.

Standard cells are another design technique aimed at rapid

implementation. The designer chooses parts from a list of possible components (these are much like standard TTL catalogues) and decides how they will be wired together. The completed netlist is then laid out. If a place and route program is available the netlist is given to the computer, which places the components in rows and runs wires between them. Standard-cell parts must be fabricated from scratch, which makes production turnaround slower than with gate arrays, but they are noticeably smaller and faster because the individual standard cells can be optimized for area and speed. Because standard cells use the same row layout as gate arrays, they have similar although less significant size and speed constraints. And when large blocks spanning several rows are introduced into a design, relatively long wires may be needed to make connections.

A major complaint about gate-array and standard-cell techniques is that they waste chip area by laying out functional components in ~~long~~ rows and then wiring them together through the adjacent channels. In addition, the wiring channels slow circuits by forcing long connection paths for some nets, and increased power is needed to drive the extended connections. Furthermore, because the components used in gate arrays and standard cells are generalized, they are less than optimal for any particular circuit. Hand layout of chips can alleviate these problems by closely clustering parts of the same functional block, making connections by abutting circuit elements, and reshaping functional block so they can be fitted into a minimum

area. Unfortunately, hand layout of entire circuits proceed very slowly. One engineer can usually handle fewer than 100 transistors a day, and errors are almost certain to occur.

2.3.2 Standard Cell Approach

Of the three possible choices the standard cell approach seems ideally suited to our needs. We do not require the fast turnaround time of a gate array approach, and even though it would decrease the actual design time it would severely limit the scope of the designs that we could do research on. A full custom approach is also not practical for large designs due to the limited amount of time, design experience, and number of people involved in IC design at the University. However full custom is still applicable for small projects where performance dictates its use. The end result is that we are left with a standard cell approach as the only way to meet the majority of our needs under the constraints imposed by the university environment.

The definition of standard cells varies from person to person. In the previous section we defined standard cells as pre-defined and pre-laid out functional blocks. This statement although accurate does not give the complete picture. Standard cells incorporate a bit of both of the other design approaches. Using standard cells forces us to use a grid system where Input and Output (I/O) connections are pre-defined and connections between cells must run between rows of cells in interconnect channels. This is the same situation present in gate arrays. We

can see that this results in a number of similarities between the two approaches. For instance, the automatic place and route routines used for gate arrays can usually be made compatible with standard cells after a few modifications such as taking into account of variable height cells. Standard cells also have similarities to full custom design. The design of the cells is an actual full custom design with all mask layers being taken into account. The placement of large macros such as memory arrays and complete microprocessors further narrows the gap between the two approaches.

Standard cells can be classified into a number of categories [5] (see figure 2.2). At the bottom level are primary cells. These include all the basic building blocks such as inverters, multiple input combinational logic, latches, flip-flops, and muxers. They come in a number of varieties and include such variations as static, dynamic, tri-state, and high drive types. In addition to these there are other primary cells for I/O, and analog functions. These primary cells which are the foundation of the of the cell library are usually of one fixed height although in some cases two may be used. The I/O and analog cells usually do not have the same constraints as the digital cells and are almost always placed and routed manually with more detail being paid to such considerations as interconnect capacitance, noise, and power supply isolation. At the next level up are macro cells. These macro cells can be divided up into two categories called custom and standard macros. The first type of

STANDARD CELL HIERARCHY

figure 2.2 Standard Cell Hierarchy

macro cells are the full custom type. These are large functional blocks in which all the transistors are hand drawn to achieve the best possible area, speed, and power performance. These cells usually have an arbitrary size although some flexibility may be sacrificed by making them compatible with a grid system. The other type of macros are called standard macros and are made up of primary cells which are connected together and optimized interconnect channels chosen to give the best possible performance. These macros are almost always compatible with any of the primary cells and can be used by the auto place and route software with some minor modifications. In some systems these macros can be generated automatically to give the designer added flexibility. For instance he/she may wish to have a 3 to 7 decoder instead of a 3 to 8 decoder which could be easily completed. At the highest level are mega macros. These are large full custom blocks which can be incorporated onto the chip. Examples of this would be an 8 bit microprocessor, a large block of Random Access Memory (RAM), or a Digital Signal Processing (DSP) block.

Given the wide range of choices the standard cell designer has a great deal of flexibility. He/she can choose from both digital and analog cells as well as complete systems in the form of super macros. As an example, a designer could design a microcomputer system by placing the core of a microprocessor on chip and adding a few kilobytes each of RAM and ROM and then use other standard cells to add I/O and other specialized functions.

2.3.3 Cell Format

The development of the cell library was carried out from June to August 1984 by Richard Vallee and the author. Considering the short period of time available as well as our inexperience in standard cell design at that time, we still managed to implement a base of 29 primary cells which enabled us to design a wide range of functions. The primary cells were limited to digital and I/O functions, however, analog cells are presently under development by another group of graduate students. In addition, a library of approximately 200 3u cells was developed by Richard E. Vallee and the author for Siltronics Ltd. in Kanata, Ont. during the period from January to April 1985. These cells can be easily made compatible with Carleton's system and will provide additional capabilities for future designers.

A listing of the primary standard cells available at present is as follows:

inverter1	latch
inverter2	latchsr
inverter4	dytflip
inverter	dytflipsr
inverta	dflip
invertb	dflipsr
nand2	dffpet
nand3	dffpets
nand4	dffpetr

nor2	outputpad
nor3	iopad
nor4	inputpad
xor	inputcpad
mux	passgate
pass3gate	

These cells are based on a grid system where one grid equals 16 x 16 micron². Each cell (except I/O cells) has a fixed height of 13 grids (208 microns), and a variable width of an integer number of grid spacings. Cell I/O locations are available at both the top and bottom of the cell (as seen in figure 2.3). Connections to the cells can be made using polysilicon or metal. The Vdd rail is located three grids away from the top and is nine microns in width to allow placement of a five micron contact. Similarly the GND rail is located three grids away from the bottom of the cell. The p transistor device well can extend 48 microns below the Vdd rail whereas the n transistor device well extends 23 microns above the GND rail, this provides a larger area for p transistors and thus more symmetric rise and fall times. In general, polysilicon and metal are run perpendicular to each other where possible.

The cells are designed such that when they are placed on the grid with the rails butting together no design rule errors will be created. To achieve this, and at the same time have a reasonable packing density, a rather complex set of cell layout rules are required. In implementing a design the cells are

design is completed a Design Rule Check (DRC) is carried out and all the important electrical parameters for simulation must be extracted. Initially these two steps were done manually, however, once the Metheus was received the extraction and DRC were automated.

After the parameters are extracted the cell is simulated using SPICE and Hg. This includes simulations under various loading conditions, and simulations to determine all the important timing values, as well as correct functionality (see figures 2.5 and 2.6).

In the next stage of the design a schematic of the cell is captured and all the important electrical and logical parameters are incorporated in a file with the schematic. This allows the schematic representation of the cell with its associated properties to be used for further simulation purposes (see figure 2.7).

Upon completion of the cells they were put on a test vehicle for characterization. This was submitted to CMC in August of 1984 and received in late December of the same year. A summary of the results can be found in chapter 5. Once the cells were designed, simulated, and characterized as well as generation of schematics with associated properties and cover page with relevant data (see figure 2.8), they were documented and put in a standard cell manual for future use.

placed in rows allowing interconnect to be performed above and below the cells. The overall structure generated is that of alternating interconnect channels and rows of cells. For more details on the standard cells refer to the Standard Cell Manual [6].

2.3.4 Example Cell

In this section the design of a typical standard cell (3-input NOR) will be covered.

The first consideration in the design of a NOR gate is that since our cells are designed with symmetric rise and fall times the p channel transistors will necessarily be significantly larger than the n channel transistors to compensate for the differences in electron and hole mobility. Symmetric rise and fall times were chosen to simplify the design process, and give better switching and high speed performance. If one considers a ratio of 2.5 : 1 in mobilities than for a three input NOR we will require the p transistors (which are in series) to be 7.5 times as wide as the n transistors. The first step is then to maximize the p transistor within the specified area and complete the design by laying out the n transistors which will have abundant area left over (see figure 2.4). We can also see that although the transistor ratios do not come out to exactly 7.5 : 1 that they approximate this. Simulation with added parasitic capacitances are performed with the transistor sized adjusted to give approximately equal rise and fall times. Once the cell

figure 2.4 3 - Input NOR Layout

design is completed a Design Rule Check (DRC) is carried out and all the important electrical parameters for simulation must be extracted. Initially these two steps were done manually, however, once the Metheus was received the extraction and DRC were automated.

After the parameters are extracted the cell is simulated using SPICE and Hg. This includes simulations under various loading conditions, and simulations to determine all the important timing values, as well as correct functionality (see figures 2.5 and 2.6).

In the next stage of the design a schematic of the cell is captured and all the important electrical and logical parameters are incorporated in a file with the schematic. This allows the schematic representation of the cell with its associated properties to be used for further simulation purposes (see figure 2.7).

Upon completion of the cells they were put on a test vehicle for characterization. This was submitted to CMC in August of 1984 and received in late December of the same year. A summary of the results can be found in chapter 5. Once the cells were designed, simulated, and characterized as well as generation of schematics with associated properties and cover page with relevant data (see figure 2.8), they were documented and put in a standard cell manual for future use.

figure 2.5 3 - Input NOR SPICE Simulation

figure 2.6 3 - Input NOR Hg Simulation

figure 2.7 3 - Input NOR Schematic

CELL NAME 3 INPUT NOR
CELL FILE nor3

a) block symbol

b) logic symbol

figure 2.8 3 - Input NOR Load vs. Delay, Logic, and Block Symbol

2.4 Summary

In this chapter a methodology for designing integrated circuits was presented. This methodology is based on a standard cell approach to VLSI design and offers an experienced or inexperienced designer a quick and efficient means of implementing a complex design which will have a high probability of being successful on the first try. This design methodology was specifically catered to the environment at Carleton with its particular constraints on time, and resources. In section one a brief introduction was given, while in section two a more detailed overview of the design method was presented. In section three the standard cell approach to VLSI design was covered and additional information on the standard cell library available at Carleton was given.

△ If we consider the design of a chip as a journey from point A (idea) to point B (end product) then we could consider standard cells as the road we take, and the design method as the signs along the road pointing us to our destination. The next question, which we will cover in the following chapter is how do we get there? >

CHAPTER 3

DESIGN TOOLS

3.1 Introduction

With the increasing complexity of integrated circuit designs it becomes essential to have the proper tools to design, simulate, and test circuits. With advanced features such as design rule checkers, netlist comparators, and circuit parameter extractors it is possible to greatly increase the probability of success on the first try. With large circuits such as on an IC the amount of data produced becomes so large that an individual or even a group of individuals cannot completely analyze it. It is here where the computer with its great capacity for data analysis takes over. By incorporating the available tools into our design methodology we are able to effectively use each for that particular task that is best suited.

The tools available at Carleton have been configured into a rather elaborate network (see figure 3.1). At the heart of the network is an advanced IC design station, the Metheus lambda 750. Connected to this are three terminals. A high resolution graphics station on which the full power of the system and its software can be used. A VC404 text terminal on which housekeeping chores, such as file management and text entry can be carried out, and a Corvus Concept microcomputer which is

DESIGN TOOLS NETWORK

figure 3.1 Design Tools Network

connected to 18 additional Concepts on an Omninet Local Area Network (LAN). The Metheus is also connected to two high speed numerical processors, one of which is the university mainframe (Honeywell CP6), and the other is a floating point processor (FPS) which is capable of speeds of up to 12 MFLOPS (million floating point operations), or approximately 10 times that of the CP6. In addition, a 9600 baud Decwriter printer and an HP 7580 plotter are connected to the network for generating hard copy and color plots. By connecting all these systems into a network we are able to provide a high powered research tool for IC design as well as a teaching environment for graduate and undergraduate students.

In addition to this network there is a Wentworth IC probe station and HP8000 series test equipment available for testing the completed IC. In the future we hope to interface this test equipment to the Metheus where we will be able to automatically generate test vectors using the HILO simulator.

3.2 Corvus Concept

There are 18 Concepts in the department, all of which are connected to the Omninet network. They use a number of shared resources which include 60 Mbytes of hard disk memory, a number of floppy disk drives, a Dataproducts line printer, and HP7211 color plotter, and a Nec 3410 letter quality printer. A single Concept is connected to the Metheus over a 9600 baud RS-232C

port. This unit also has a separate 20 MByte hard disk drive and floppy disk drive. From this unit we can run a program to emulate a VT100 and communicate with the Metheus as either a dumb terminal, or for transferring data between the two systems we can run a communications program called CC.DCOMM. CC.DCOMM can also be used for direct communications between the Concepts and either of the numerical processors.

The Corvus Concept has implemented on it a VLSI package [7]. Central to this package is the integrated circuit layout program called CIFDRAW, which was developed at Carleton and generates CIF code for silicon foundry fabrication. This program is written in SVS Pascal and uses the screen graphics capabilities of the Corvus. Supporting the IC layout are Concept-specific implementations of SPICE and SPLICE for analog simulation and mixed mode analog and logic simulation respectively. These packages are realized in SVS FORTRAN. The three applications programs all require a Concept configured with 512 Kbytes of RAM with the Motorola 68000 CPU. The 68000 has approximately 1/30 the floating point speed of a VAX780 and small simulations using SPICE such as a single CMOS op-amp, require 5 to 10 min.

CIFDRAW permits an IC designer to create Caltech Intermediate Form (CIF) symbols as a means of describing an integrated circuit. Based on the book by Mead and Conway [1], the designer uses the Concept as a graphical editor creating rectilinear rectangles using a pair of graphic cursors, (one is static, and the other is dynamic), which are controlled by the keyboard

function keys. The program supports eleven layers making it suitable for both NMOS and CMOS processes. The program is completely function key driven, with appropriate function key annotation. The key labels provide easy prompting with little instruction for new users.

Although it is very easy to learn and use, this layout editor suffers from a number of drawbacks. The Concepts monochrome screen has quite reasonable resolution (720*560), however, with up to 11 mask layers available in the CMOS process it can become quite difficult to distinguish the actual layers. With continued use the designer can adjust himself to this problem. Probably the biggest drawback with CIFDRAW is that it does not use a hierarchical structure, and therefore whenever a cell is called into a larger cell it is flattened. The end result being that for large designs, the flattened CIF file becomes excessively large. For a typical MSI circuit it would require much more than the 1500 blocks of hard disk memory allocated to each user and would take up to 20 minutes to read in the file. Because of the flat file nature of CIFDRAW whenever a new cell is called in, if it is not placed at the correct spot the whole file must be erased, and the original file called in with another attempt being made at placing the new cell in the correct location. As the circuit layout expands this task becomes exceedingly difficult as was seen in the design of the standard cell test vehicle which required several hours to make minor changes when it was near completion.

Considering the limited computing power of a single micro and the inherent limitations of the CIFDRAW program the Corvus network is best suited for the design of individual standard cells or small functional blocks. The cells can be simulated and laid out on the Corvus and if the cells are fairly large SPICE simulations can be carried out on one of the numerical processors. With the HP7211 plotter, checkplots of the cells can be generated and using CC.DCOMM we can transfer the cell to the Metheus where the can be incorporated into the library. The Concept network is also ideally suited for teaching purposes. It provides a large number of students with both layout and simulation capabilities and would be appropriate for almost all the projects covered throughout a one term course. Once a project is completed it can be sent to the Metheus to perform high level functions such as DRC's.

3.3 Numerical Processors

Circuit simulators such as SPICE or SPLICE are computationally intensive in that they require large numbers of floating point matrix operations. Coupled with this is the fact that, as the size of the circuits increase, the number of operations as well as memory requirements involved increases exponentially. Therefore highly accurate simulations of large standard cells or functional blocks presents a problem. For simulations of some of the larger cells on the Corvus the simulation time becomes quite large due to the limited computing power. The complete

simulation of a D flip-flop (35 transistors) using SPICE level 2 would require up to 30 minutes. For functional blocks, such as the delay circuitry for the memory controller (87 transistors), the memory limitations make simulation impossible. The simulation time problem also occurs on the Metheus where even with a numerical accelerator board the simulation time can run into hours. In doing large designs this time lapse is unacceptable and therefore another route is necessary.

Connected to our design network we have two numerical processors. The honeywell CP6 is the university mainframe. It offers both SPICE and SPLICE simulation packages with a 10 fold improvement in simulation time when it is not under heavy use. However, for smaller cells the overhead for the time sharing on the mainframe makes simulation times comparable to that attainable on the Concepts. In addition an FPS processor has recently been obtained at the university. This parallel processor uses state-of-the-art technology, and has been specifically designed for high speed floating point operations. The FPS is connected to a network of Apollo workstations where its main use is in CAD development. However, because of its limited use at the present time it is made available to us for use in large SPICE simulations. At present, it has not been used extensively but it promises to be an additional important resource.

3.4 Metheus Lambda 750

3.4.1 Overview

The Metheus is at the center of the network. This IC design workstation has a majority of the most advanced features available at present, and coupled with all the support tools forms a complete design system [8]. The Metheus itself is centered around a set of dual processor Motorola 68010 CPU's. One of these 16 bit processors acts as the main CPU while the other takes care of all the I/O functions. The main memory for the system is 2 MBytes which is expandable up to 4 MBytes in 1 MByte increments. In addition, the system has two 33 MByte Winchester hard disks. All the graphics functions for the system are controlled by a specialized bit slice graphics processor which accommodates 16 bit-planes of (1024 X 1024) bits and interfaces to a 19 inch Hitachi color monitor. The I/O controller interfaces to a number of peripherals which are included in the following list:

- Keyboard
- Optical Mouse
- 1 MByte Floppy Disk Drive
- Cipher F800 Tape Drive
- HP7580 Color Plotter
- Decwriter Printer
- 300 Baud Modem
- Hardware Accelerator Board

Graphics Monitor

Winchester Hard Disks

In addition, six RS-232C ports and two GPIB interface ports are available to support a multiuser environment. Of the extra RS-232C ports one each is used for the VC404, Corvus Concept, CP6, FPS (through an Apollo), and HP plotter. Of the two GPIB ports one will be used to interface to the HP test equipment.

The Metheus runs on Berkeley Unix version 4.1 operating system. The system supports multiple display channels available on the graphic terminal as well as additional users connected through an RS-232C port. In addition, there are a complete set of on-line manual pages for quick reference. Also available are a number of utilities such as screen and text editors, text formatters, electronic mail, data base management and C, fortran, and pascal programming languages.

The Metheus applications tools can be divided into three main groups (see figure 3.2) which will be covered in the following subsections. The logic design tools include all the programs associated with schematic capture and those that interface to the netlist verification tools. In addition, a graphics editor for general purpose applications such as documentation is available (top half of figure 3.2). The physical layout tools include all those associated with the mask level representation (layout) of the circuits as well as format conversion routines

METHEUS LAMBDA 750 DESIGN ENVIRONMENT

figure 3.2 Metheus Design Environment

and interface programs to the netlist verification tools (lower left of figure 3.2). The netlist verification tools include all the simulators as well as programs that verify direct schematic and layout correspondence, and electrical rules violations. A special graphical interface from the simulators to the graphics terminal or plotter is also available (lower right of figure 3.2).

3.4.2 Logic Design Tools

The logic design tools include a number of programs (see APPENDIX I for more details) to aid the designer. SYMED is a general purpose graphics editor whose primary function is to generate symbols for circuits and documentation. Associated with SYMED is SCHEDIT which is a schematic editor for connecting the circuit symbols together. Each of these editors interfaces to a library of symbols and schematics for the standard cells, which they have read access only.

In designing a circuit a user first calls in a symbol from the library which will have a .sym extension (e.g. nand2.sym) this symbol would be the classical 2-input NAND symbol (see figure 3.3a). Directly associated with this symbol, as seen in figure 3.3b, is the CMOS schematic of the gate (nand2.sch). In addition to the schematic there are a number of properties associated with the schematic. These include such things as transistor sizes, area of interconnect, and delay vs load information. These properties are important later when they are

figure 3.3(a) Nand2 Logic Diagram

Figure 3.3(b) Nand2 Schematic Diagram

used as input data to the simulators. As the designer builds up the circuit he/she will call in additional symbols and connect them with wires. If he/she requires circuits not available in the library new symbol and schematic (with associated properties) will have to be generated.

Once the circuit has been completed it can be plotted on the HP plotter using SCHPLOT, or if a plot of a symbol or diagram is required SYMPLOT can be used. The next stage will then be to use one of the three translation programs SCH2SIM, SCH2IMF, or SCH2HILO to convert the schematic into a netlist format that can be used by one of the simulators.

3.4.3 Physical Design Tools

The physical design tools are used to interface to the actual circuit layouts. The central program is PHLED, which is the physical layout editor. This program which uses the full capabilities of the graphics terminal can be used to manipulate the geometries of the mask layers. With high level functions such as yank, wire, and fill the layout of circuits is much easier than using a program such as CIFDRAW. Also the layout format is hierarchical so that complete chip layouts can easily be accommodated.

In laying out a circuit the designer would first call in a cell from the cell layout library. When calling in complete cells there is no need to have all mask layers visible.

Therefore the cell is left unexpanded with only I/O ports visible. This reduces the load on the graphics processor and helps reduce any confusion in visualizing the remainder of the circuit. Also the designer is not concerned with what is inside the cell (especially if he/she has no experience in IC design). After the cells are called in they are connected with polysilicon and metal to form the desired circuit. While connections are being done the designer can interactively DRC the interconnect, or once finished, can run a DRC on the whole block of circuitry. When design rule errors are present the location of the error is written into a file which can be input to PHLED to give the location of the error, and thus make corrections rather effortless. When the layout is completed the block can be plotted on the HP plotter using the PHLPLOT program, for additional checking.

After the plot has been completed the physical layout extractor program (PHLEX) can be used to generate netlist files for either the SPICE simulator (.imf format) or the Hg simulator (.sim format). Also available are the programs PHL2PHL, PHL2CIF, and CIF2PHL for scaling the layout or converting it to the appropriate format.

3.4.4 Netlist Verification Tools

The netlist verification tools are based on the three simulators; SPICE, Hg, and HILO, each of which has a particular

application. SPICE is a circuit simulator which provides detailed timing analysis as well as dc and ac analysis. Hg is a switch level simulator which provides less accurate timing analysis on sequential and combinational MOS digital circuits. Although the analysis is less accurate, the amount of circuitry that can be simulated is orders of magnitude larger. As an example, it was possible to simulate a complete chip (2900 transistors) all the way down to the transistor level using Hg. HILO is a gate level logic simulator in which each cell can be modeled by delay vs. load conditions. In addition, fault simulation and automatic test vector generation are available on HILO.

Once the designer has completed the initial schematic it should then be converted to a netlist and simulated until he/she is satisfied with the results. The output of the simulators can be fed into a special graphics routine called SIMGR which will interface to the graphics terminal or the HP plotter, and aid in the analyzing the simulator results. He/She then completes a layout of the circuit and again generates a netlist. This second netlist should correspond to the first with only the added interconnect area being different. To ensure that the layout is correct a program called COMPARE can be run which will compare the two netlists and check for the same connectivity, node names, and properties. If there are any errors they will be fed back graphically to PHLED where they can be easily spotted. Once this is completed the designer can resimulate the circuit with added node capacitances to ensure that the actual circuit will conform

to the required specifications. This completes the design of a functional block. The designer would then repeat the process on a new functional block until all the blocks are completed.

3.4.5 Design Case Study

In this section we will consider the design of a complete functional block in the form of a 2 to 4 decoder. The design process covers a number of steps (as seen in figure 3.4) from start to end. The first stage in the design of the decoder (once the standard cells have been designed) is to put together the schematic using SCHEDIT (see figure 3.5). This will involve the use of the standard cell schematic and symbol files in the cell library as well as the users own library. Once the schematic has been generated the user then generates a netlist for the particular simulation desired (see figure 3.6). Generally the netlist file would then be checked for electrical errors by using the ERC program, however, this program has a large number of problems with it and has not been fully integrated into the environment as of yet. If the functional block is not too large it is possible to do a SPICE simulation. Although a 2 to 4 decoder is probably not too large for this type of simulation a simulation using H₂ was done for demonstration purposes. If the timing is not too critical an H₂ or HILO simulation can be done with relative ease and with the results of previous SPICE simulations of the standard cells to input into the H₂ and HILO simulators, the results will be fairly accurate. To do the

FUNCTIONAL BLOCK DESIGN

figure 3.4 Functional Block Design Flowchart

figure 3.5 2-to-4 Decoder Schematic

simulation a command file is generated which directs the operation of the simulator. For example, which nodes to observe, which nodes to put input signals on etc. The output of the simulation is then directed to the grapher program called SIMGR for observation and can be output as hardcopy on the plotter or printer (see figure 3.7). Once the simulation is completed it may be necessary to change the properties (ex. transistor sizes) of the individual cell to meet a strict timing requirement. This can be done easily by editing the properties in SCHEDIT, generating a new netlist and resimulating using the same command file.

Once the simulations have been completed to satisfaction the next stage is to layout the decoder. For this case we would put together four 2-input NOR gates and two inverters connecting the appropriate nodes. During or after the decoder is laid out we can do DRC's and finally generate a checkplot using the PHLPLOT routine (see figure 3.8). We must next run PHLEX to extract the electrical parameters in the form of a netlist (see figure 3.9). The generated netlist can be verified by comparing it with the schematic netlist (see figure 3.10). In this example we see that it has picked up two errors. There are actually two files in figure 3.10, one for graphical feedback to PHLED and the other for feedback to SCHEDIT. The errors involved were caused because two ports on the schematic were labelled with lower case letters. Although these errors would not affect the overall functional block design they demonstrate the ability of the program to pick

figure 3.7 2-to-4 Decoder Hg Simulation

figure 3.8 2-to-4 Decoder Layout

I/O LOCATIONS

TRANSISTORS

NODES

figure 3.9. 2-to-4 Decoder Layout Netlist

ERRORS TO INPUT TO SCHEMATIC

:echo:svsmod1 checked against svsmod12

:echo: Bad names

:count 279 206 2 2: :paint e: :label e 11_81: :halt

:count 279 206 2 2: :paint e: :label e 11_81: :halt

:echo End of errors

ERRORS TO INPUT
TO LAYOUT

Figure 4.10 2-to-4 Decoder Netlist Comparison

up such details. Once the correct layout is achieved the laid out circuit is simulated again using the same command file but with the additional interconnect capacitances. In this case the capacitance is so small that there is an undetectable change in the circuit delays. If the specs are still met then the functional block design is complete, otherwise we have to go back and redo the layout to minimize capacitance, or all the way back to the start and use a different schematic to realize the function we require.

3.5 Test Tools

The functional testing of an IC can in general be carried with a limited amount of resources given time and patience. However, the provision of HP8000 series IC test equipment greatly facilitates this process and for implementing highly complex timing waveforms is almost essential. A probe station to examine internal signals is also of great value. The test equipment available at Carleton includes the following:

HP9816	Microcomputer
HP7475	Color Plotter
HP8180	Data Generator
HP8182	Data analyzer

Wentworth Probe Station

In addition to the above list there are a number of tools available for fabrication process analysis and characterization,

however, because they are in general not used a great deal by the IC designer the author will not elaborate on their capabilities.

The 8180 data generator is a high speed (50 MHz) programmable data generator which (present configuration) can generate up to 8 channels of data to stimulate the digital circuit under testing. The 8182 is a data analyzer which can collect (present configuration) up to 12 channels (each with 1 kbit of memory) of simultaneous data. By combining these two units a complete system for stimulus and response of digital circuits is obtained [9], [10]. The operation of these two units can be stand alone (softkey programming) or under the control of a host microcomputer as in our case. The 9816 microcomputer is used to control both the data analyzer and generator. This computer which is programmable in basic is also connected to the 7475 plotter to generate timing diagrams of the collected data. In addition, if access to internal nodes on the chip is required the Wentworth probe station can be used.

3.6 Summary

In this chapter the design tools used at Carleton were covered. In the first section an overview of the computer network was presented with the succeeding sections concerned with the details of each part of the network. Finally, a section was presented on tools used for testing the completed IC.

This chapter shows that by connecting our resources into a

network each tool can interact with the others and perform the job it is best suited for. This means we use resources such as the Corvus Concept Network for doing low level design or teaching, where we require a large number of design units with basic capabilities. It also means we use our big numerical processors for computational intensive tasks, such as simulations, and we use our advanced tools such as the Metheus for high level designs that are present in research projects.

Although the environment we work in is very amenable to IC design it is by no means perfect. As mentioned previously there are a number of problems encountered when using the Concepts and this is also the case with the Metheus. Because this tool is newly developed it therefore inherently will have bugs. Some of the major problems include: The plotting of symbols and schematics can be at times unpredictable. The sizing of the file to fill the paper can at times result in plots which are totally out of proportion. The PHLPLOT routine, because it flattens out the file can generate excessive amounts of data for large plots. In generating a 2000 transistor plot it requires 15 Mbytes of hard disk and 12 hours to plot. The layout extractor PHLEX when extracting a netlist for SPICE randomly determines the source and drain of the transistor. If the choice is made incorrectly the thresholds will be incorrect (due to body effect) and can drastically affect the simulated performance of the circuit. For digital simulations it is possible to accept some error, however, for analog circuits it is unacceptable. However, it still extracts parasitics which save a great deal of measurements and

calculations. The switchlevel simulator Hg, while perfect for general combinational circuits, has problems dealing with transmission gates. For the chip that was designed this resulted in the use of NAND gate flip-flops instead of transmission gate flip-flops to be able to achieve complete circuit simulation. This meant an increase in overall area associated with the NAND flip-flops. The gate level simulator HILO operates only on schematics and therefore is limited in use because we cannot compare a layout with the schematic to verify correct functionality. The test tools available make the testing of the IC a relatively easy process however, they have a limited number of channels (8 input and 12 output) which means we have to do a bit of juggling for a 40 pin chip.

In conclusion, although there are a number of problems, the use of the available tools makes the design of a complex system with high probability of success still possible. We will demonstrate this in the next chapter where we will consider the design of a complete system within the environment we have developed using the aforementioned tools, to show that the system we use is appropriate and that it will give successful results.

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

As part of a pilot project to test whether or not it was possible to successfully design and integrate a chip within our new design environment a three chip set multi-mode data acquisition unit was proposed [11].

As part of his M. Eng. work at Carleton, Carlos Pon designed and characterized a microprocessor-controlled digital waveform recorder [2]. The digital waveform recorder consisted of (see figure 4.1) a 6800 system controller and an acquisition unit which interfaced to a Corvus Concept computer for post-processing, a digital plotter for hard copy of the waveforms, and an analog scope for continuous waveform display. The new multi-mode data acquisition system was to replace the 6800 microprocessor with its several Kbytes of code with a dedicated controller and part of the acquisition unit with a memory controller chip.

A more detailed diagram of the waveform recorder can be seen in figure 4.2. The M6800up interfaces with a fast sample and hold (S/H) HTCD300, a 12 bit analog to digital (A/D) converter ADC-EH1B3, a 12 bit digital to analog (D/A) converter DAC-69-12B,

figure 4.1 Digital Waveform Recorder System Configuration

figure 4.2 Digital Waveform Recorder System Architecture

2 Kbytes of buffer memory, and a large amount of glue circuitry. On the CPU data bus, in addition to the 6800 there is a serial port to the Corvus, RAM and ROM for programming the 6800, a parallel I/O port to interface to the main part of the acquisition unit, and a floppy disk controller for memory backup. Several Kbytes of code were required to support the real time random, and sequential sampling modes.

The real time sampling technique is the only method that can acquire single event signals. To accomplish this it takes samples of the input signal continuously for a finite time. A train of sampling pulses is generated which acts as a fast switch connecting to the analog input signal for a very short period of time and then disconnecting for the remainder of the sampling period. In this method the A/D and memory must be able to operate at the sampling frequency. Random and sequential sampling techniques only work for repetitive events. Both techniques work well for high frequency signals. These implementations consists of taking one data point per sweep. For sequential sampling the input signal is sampled at a slightly advanced point on each sweep. In random sampling the sampling pulses are not held in a synchronized relation with the signal, but instead are sampled with reference to a delay generated by a pseudo random sequence generator.

During sequential sampling, the waveform is digitized in successive increments of time with respect to the signals referenced trigger pulse. This data is sequentially stored in

the buffer memory until all 2K bytes of memory are filled.

Random sampling digitizes the waveform at random intervals with respect to the signals referenced trigger pulse. This data is then stored at the address given by the random number.

The buffer memory consists of 2K x 12 bits of NMOS static RAM. The address of the memory is controlled by a 12 bit synchronous counter with 11 bits being used to access the 2048 RAM location. The 12th bit is used as a flag for memory full. The memory and address counter are synchronized with the same signal used to clock the A/D converter. Once a waveform has been fully digitized, the data can be continuously cycled and latched into the D/A for display on the scope or transferred via the CPU bus to the Concept for additional processing.

The new system was to incorporate the majority of the features found in the previous system and to add some additional ones. This included upgrading the maximum real time sampling rate to 1 MHz from 500KHz, increasing the buffer memory from 2 Kwords to 64 Kwords, and offering a variable size and number of data channels.

4.2 Device Requirements

4.2.1 Multi-Mode Data Acquisition System

In designing the multi-mode data acquisition system the first step was to determine the overall device requirements. The

analog components of the system were to be left unchanged. Therefore, we kept the same configuration for the vertical preamp, S/H, A/D, D/A, and trigger generator (See figure 4.3). The system was now required to interface to all these components and in addition, the system memory was to be upgraded to 64 Kwords to provide up to 16, 4 Kword channels of data with complete programmability in channel size and channel memory location.

To achieve this increased memory the use of 64K Dynamic Random Access Memories (DRAM's) were chosen to minimize overall system size and still meet the 1 MHz maximum operation frequency. In early stages of the project it was decided to break the system up into two parts, each of which could be designed independently of the other. Richard Vallee was to integrate the system controller [12] on what eventually required two chips, and the author was to integrate the memory controller. Both the system and memory controller are under control of the user interface which would control such things as sampling mode, size of data channel, and starting address of data channel. The system controller, in addition to controlling the S/H, and A/D would generate the addresses for sampled data and feed it to the memory controller unit which would in turn interface to the DRAMS as well as any external processors.

General Overview of the Device Requirements

figure 4.3 General Overview of Device Requirements

4.2.2 Memory Controller Unit

The memory controller unit is to provide an interface between the system controller and external processors to the 64 Kword memory. The system is designed for dynamic memory, however, static memory can be used with external demultiplexing of the lower and higher order addresses. The controller is to divide the memory into variable sized blocks which can range between 512 and 4K words (in 512 word increments). In addition, the memory can be accessed from any starting address to enable the user to zoom in on any one section of the waveform for a more detailed picture. The memory should be accessed either synchronously (refresh and block transfers to external processor) or asynchronously (writing data into memory after it is sampled) with the controller taking care of the arbitration. In addition, the controller has to generate the complex timing, multiplexed addresses, and continuously refresh for the DRAM'S.

4.3 System Overview - Functional Block Diagram

The memory controller chip (as seen in figure 4.4) can be divided into a number of sections. Proceeding from left to right we see that the initial address is fed into one of two ports. The top port is for asynchronous accesses whereas the bottom port is for synchronous accesses. In addition, the block size can be loaded into a small 4 bit latch from the higher order addresses. As we continue to the right we see that these two 16 bit ports

MEMORY CONTROLLER FUNCTIONAL BLOCK DIAGRAM

figure 4.4 Memory Controller Functional Block Diagram

are multiplexed and then the higher and lower order bits are multiplexed to give the final 8 bit address which is passed on to the DRAMs. In the upper right corner we have the synchronizer which arbitrates between the synchronous and asynchronous access as well as generating the control signals for the DRAMs. In the lower right corner, we have a divide by 2 and johnson counter which generate the overall timing signals for the system, as well as a scan circuit which is incorporated on the chip to give access to internal nodes for reading or programming the state of the system.

Operation of the synchronous port consists of an initial 16 bit start address being loaded into a latch under the control of the user interface. This requires a pulse on CDIS to load the address which is fed to the memory controller through the user interface and system controller. For the selection of the memory block size the same procedure is followed, only now CDEC is pulsed, and only the four higher order bits of the address bus are latched to give channel size selection of from 512 to 4K words. By having complete programmability of the start address and memory block size we can select any section of memory for display or data transfer purposes. As an example, if a 4K point trace of a waveform on the analog scope is too weak we can break it up into 512 word sections where we will be able to observe the finer details. The start address, once loaded, is fed into a 12 bit programmable recycling up counter (See figure 4.5), and a 4 bit latch for the higher order bits. This ripple counter is clocked by the johnson counter and is loaded under the control of

the block size counter. When doing a block transfer, or refresh, the counter is incremented after each access. The counter is continually clocked so that the addresses will be changed and the DRAM will be continually refreshed every 256 or 512 counts (depending on type of DRAM). The outputs of the counter and its associated latch are fed into a 32 bit 2-to-1 multiplexer where they are multiplexed with the asynchronous port. The output of this multiplexer is then fed into a 16 bit 2-to-1 multiplexer to give the multiplexed 16 bit address to the DRAMs. The block size latch outputs are fed into a programmable reloading down counter. This 12 bit counter (see figure 4.6) has the 4 higher order bits programmable with the 8 lower order bits settable so that the minimum size 512 word block is achieved. The counter is clocked by the same clock as the upcounter and receives a load enable pulse from the johnson counter so that when a zero count is reached, it reloads itself as well as sending out a signal to reload the down counter and associated latch.

The asynchronous port consists of a 16 bit latch which feeds directly into the 32 bit multiplexer. The loading of this latch is under the control of the EOC signal which is sent out by the A/D converter once it has completed conversion and the correct data is present on the data bus. The EOC signal is also sent to the synchronizer along with a R/WIN signal. The synchronizer will use the EOC signal for starting an asynchronous access and the R/WIN signal to choose between an asynchronous read or write.

The synchronizer circuitry is based on the approach used in

figure 4.6 Downcounter Schematic

the Intel IRAM controller [13]. Whenever synchronous and asynchronous accesses take place there is always the possibility of the two occurring at almost the exact same time. If this happens, there is the possibility of the system entering a metastable state from which it will take an arbitrarily long time to recover from. The approach used here is not to eliminate this possibility which is believed by most people to be impossible, but instead to reduce it to an acceptable level. The result is the use of a flip-flop with a fast switching time and therefore reduces the probability of failure. Also incorporated in the synchronizer is additional circuitry for generating delay signal, multiplexing asynchronous and synchronous signals, and flip-flops for latching signals.

The system timing control circuitry consists of a divide by two T flip-flop for squaring off the external clock which is then fed into an octal johnson counter which will provide three non-overlapping clocks to the internal circuitry. The clock sequence consists of a clock which is generated by an external 4 MHz crystal. This is fed into the T flip-flop to give clock C (see figure 4.7). This 2 MHz clock is then fed into the johnson counter to give us three 500ns pulses (A0, A2, A6) to control the internal circuitry. The overall clock period is divided down to 250 KHz. This provides us with a synchronous access every 4us. This is important for two reasons. First, because synchronous accesses are doing the refresh for the DRAM we require a minimum of 2ms between refresh for each bit to maintain data integrity.

SYSTEM TIMING

figure 4.7 System Timing Diagram

For a 64K DRAM this means we must access one of the rows a minimum of every 16 μ s. Secondly, because the data is being sent to an analog scope we have to worry about display rates. If we are using up to 4K points in each channel, and we require a minimum of 20 ms per raster to provide flicker free display. We will require a display rate of 4.88 μ s. The choice of 4 μ s will meet both of these specs. The output of the johnson counter consists of these pulses. During the first pulse (A0), the synchronous port counters are clocked to give a new synchronous address and decrement the block size count. One microsecond later the pulse (A2) is sent to the synchronizer which will start a synchronous access to memory. Finally, a pulse is sent to the down counter (A6) which will enable it to reload the counters if the zero count state has been reached.

The scan circuitry was included to give access to a number of internal nodes which are otherwise unobservable. By using this scan circuitry we can read or program nodes. This means by choosing the correct internal nodes we are able to put the controller into a number of predetermined states and therefore further enhance testing of the chip. Additional information on the scan and synchronizer circuits will be presented in the following section.

4.4 Synchronizer and Scan Circuits

4.4.1 Synchronizer

Conventional logic circuits can monitor requests for refreshing (internal) and memory access (external). When a request for either is received while the other is being serviced, the request is simply queued and satisfied in its turn. The problem with such a circuit is that the two requests may occur virtually simultaneously, and some mechanism must guarantee a quick decision as to grant service to which request. It matters little which one is acted on, so long as one is acted on promptly. The circuit that arbitrates such a decision, called an arbiter or synchronizer (see figure 4.8), has stirred some interest among circuit theorists and system designers during the last decade. The central issue is the response time of a bistable element subjected to simultaneous conflicting inputs. A conventional cross-coupled inverter latch runs the risk of entering a metastable, or balanced, intermediate state when its inputs are nearly equal.

It is fairly easy to estimate the probability of synchronization failure with a fairly simple model [1]. If we consider that a device is in a metastable state at some time T , the probability that it will have left this state at some time $T + dt$ as dt approaches zero should be proportional to dt . This assumption produces a simple model in which an exit from a metastable state is based on a Poisson process of rate P and the

figure 4.8 Sync1 Schematic

probability of exit from a metastable state in a period of time D is given by e^{-PD} . This simple model has been analyzed experimentally and is consistent with analysis based on circuit models [14].

In order to estimate the probability of a fault in a synchronous system due to the nonvanishing probability that a synchronization will take longer than some bounded time, one must also calculate the probability that a synchronization event will put the synchronizer into a metastable condition. For most synchronization, the asynchronous level to be synchronized will transit sufficiently far away from the synchronization time that the flip-flop will be overdriven in the usual way. Only over a rather narrow time aperture, denoted here as A , does the occurrence of a transition result in the synchronizer taking longer than the usual delay time. The boundaries of this aperture are not sharp but may be treated as such, so that for a frequency F of transitions of the asynchronous signal, the probability of a metastable condition resulting is FA . The overall probability of system failure at each synchronization event, FAe^{-PD} , depends on P and A , which are parameters of the flip-flop, on D , which is a parameter of the synchronous system, and on F which is a parameter of the asynchronous signal. For our system the maximum frequency (F) is set at 1 MHz. The time allowed for a memory access is 500ns, with up to 400ns required to access the DRAM this leaves 100ns delay time (D) in which the synchronizer must perform its job. The parameter P is dependent on the flip-flop used and is equivalent to $4/T_s$ where T_s is the

transistor saturation transit time. For our flip-flop P has been calculated to be $3.36 \times 10^{-10} \text{ s}^{-1}$ (see APPENDIX III). Experimental determination of A for a number of circuit families [1], [15], [16] indicate that A is a small fraction of T_S , say about $T_S/10$ or $1.19 \times 10^{-11} \text{ s}$. However, by decreasing the switching time the value of A is reduced [16], [13]. The overall probability of failure (PF) is given by:

$$PF = (10^6 \text{ s}^{-1})(1.19 \times 10^{-11} \text{ s}) \exp [(3.36 \times 10^{-10} \text{ s}^{-1})(1 \times 10^{-7} \text{ s})]$$

which means a failure rate less than one in 10^{99} . This extremely small probability is mainly due to the rather large (100ns) amount of time available for recovering from the metastable state. The part of the synchronizer concerned with arbitration is seen in figure 4.9. The action of this circuit can be seen in figures 4.10 and 4.11 where the first figure shows normal synchronous and asynchronous accesses, and the second shows what happens if both occur very close to one another. The exponential exit rate in the last figure is clearly evident.

The operation of a dynamic RAM is complex. Not only do its multiplexed address inputs require delayed timing signals, but it must be refreshed continually [17].

During an access to a RAM (see figure 4.12a), the first step requires that a row address be presented to the multiplexed address inputs. As the row address strobe (RAS) goes low, the address is latched into the row latch, and decoded to the memory array. There, the outputs from the selected row are presented to

SYNCTEST

figure 4.9 Synctest Schematic

figure 4.10 Synctest SPICE Simulation 1

figure 4.11 Syncctest SPICE Simulation 7

2 2
OF / DE

DRAM TIMING

figure 4.12a DRAM Timing

the sense amplifiers. Row addresses must be held on the address inputs for a predetermined time (t_{hold}) after RAS switches low. At this time, the row address can be replaced by a column address. When a column address is valid, the column address strobe (CAS) goes low to latch the address into the column latch. Column addresses are decoded to allow a selected sense amplifier to send data to each output data-latch (during a read cycle). In a write cycle, with the Write Enable signal (WE) already low, a low-going CAS signal causes the selected cell to be set to the value at the data input.

Besides a complex sequencing arrangement, dynamic RAMs must be refreshed to prevent the capacitor in each cell from losing its charge, which represents information. If any row is not accessed for too long a period of time (the refresh period) capacitors will discharge, causing the voltage to drop below the sense amplifier threshold. Then, when the row is finally accessed, its outputs will appear as all zeros or all ones, depending on which side of the sense amplifier is accessed. Most RAMs have 2- μ s minimum refresh times, but 64-K dynamic RAMs are typically much higher. For some RAMs, 128 rows must be refreshed in 2 ms, while others require refreshing 256 rows in 4 ms. With distributed refreshing, one row must be refreshed every 16 μ s for proper operation.

In the synchronizer all DRAM access timing is generated from one of the strobes, R/REF in the case of synchronous access and EOC in asynchronous accesses (see figure 4.12b). An on-chip delay

figure 4.12b Delay SPICE Simulation

automatically generates a guaranteed selectable row address hold time (delay between $V(out1)$ and $V(out2)$). At this point the memory controller switches the address outputs from the rows to columns $V(out2)$. Then another on-chip delay generates a guaranteed column address set-up time before CAS (delay between $V(out3)$ and $V(out2)$, so that CAS i.e. $V(out3)$ automatically strobes the column address into the DRAMS. Read or write cycles are controlled by R/WOUT signal which is held high for synchronous access and under external control for asynchronous accesses. The automatic access mode makes the dynamic RAM appear static with regards to access timing. Only one signal EOC is necessary after valid parallel addresses are presented.

SCAN CIRCUIT

Designing and fabricating a VLSI chip is pointless if the chip does not work. Yet methods of testing a chip quickly and cheaply have received scant attention until recently. The area has suffered from both a lack of understanding by engineering managers of the importance of testing, and a lack of computer tools to help the test engineer design test programs. The technologies for rapid design and fabrication of custom chips have far outpaced the technology of testing.

The design methods known loosely as "design for testability" involve adding extra circuitry to ICs to allow engineers to write test programs more quickly and to test the ICs with inexpensive

equipment [18]. These methods, however, increase the size of ICs, thus making them more expensive to manufacture and degrading their performance somewhat. The types of things that can be done include adding test lines to combinational logic, using reset signals to put register and counters in known states, breaking up counters to reduce test states, and multiplexing I/O pins for test mode capabilities. Another technique which has gained recent popularity is that of scan design.

In scan design the pin requirements for testing the chip are reduced by incorporating a number of flip-flops and multiplexers in such a way that the data can be loaded into the flip-flop in a parallel manner and clocked out serially. Therefore the larger the number of flip-flops (i.e. signals) the greater the number of pins saved. In the design (see figure 4.13) each flip-flop receives its data either from its normal path (internal signals), or from the output of the previous flip-flop, depending on the TEST pins state. The first flip-flop is an exception in that its input comes from a test input pin (TESTIN). The last flip-flop has its output go to a test output pin (TESTOUT). The circuit can be placed in test mode by setting TEST high. In this mode all the flip-flops are configured as shift register thus allowing a serial data stream to be pre-loaded or read-out off each storage element.

In the memory controller with the addition of test circuitry we gain access to almost all the important internal nodes which enables a more thorough testing of the chip.

SCAN CIRCUIT

Figure 4.13 Scan Circuit Schematic

4.5 Functional Block Design

In this section we will cover the actual functional block design of some example sections of the memory controller unit. This will consist of a brief explanation of each block's function, a detailed schematic, and timing waveform. For the details on the remainder of the functional blocks refer to APPENDIX II.

4.5.1 Johnson

Johnson is a complete octal Johnson counter in which the A0, A2, A4 and A6 outputs are provided. These outputs provide four non-overlapping clocks which are used for the overall system timing. The Johnson counter consists of four D flip-flops with a 2-input NOR gate, five 2-input NAND gates, and six inverters to provide decoding of states and buffering of the output signals (See figure 4.14). In addition, the flip-flops have a reset available to ensure start-up in an allowed state.

The operation of Johnson can be seen in figure 4.15 where RSN is used to initialize the counter into the zero state. The clock C is then started with resulting non-overlapping clocks being generated at the outputs A0, A2, A4 and A6 in succession.

4.5.2 Latct4

In order to achieve a better packing density the two sets of 16 bit latches for the synchronous and asynchronous ports, and the

JOHNSON COUNTER

figure 4.14 Johnson Counter Schematic

figure 4.15 Johnson Counter Hg Simulation

programmable up counter are, combined in a bit slice arrangement called latch. As a demonstration, we have four of these combined (as seen in figure 4.16). The second set of latches which are used for latching the asynchronous address are not shown in order to facilitate visualizing the schematic for this functional block. The latches for the asynchronous port would have the same input (A0 for example) and are clocked by EOC and EOCb.

In latct4 we have the four address inputs (A0 - A3), four counter outputs (CT0 - CT3), and four asynchronous port outputs (RW0, RW3). CDIS is used to load the synchronous port latches, CRW loads the asynchronous port latches, LD is used to load the programmable up counter, and CRIP clocks this ripple counter. In the first 6 time step of the simulation (see figure 4.17) we have the clocking in of the asynchronous port addresses of both logic levels to demonstrate that all stages are totally loadable. We then have CDIS and LD going high successively to cause a loading of the address into the synchronous port latches and counter, which can be seen just after 70u. This sequence of step is repeated again to load in the opposite logic levels. At the 1.5 us mark the counter clock CRIP is started and we can observe the clock counting up in the following stages of the simulation. In the memory controller chip the bit slice latct is formed into a 12 bit counter which also encompasses the 12 lower order bits of the synchronous and asynchronous address latches.

LATCT4

figure 4.16 Latct4 Schematic

figure 4.17 Latct4 Hg Simulation

4.5.3 Syncl

Syncl is the synchronizer (see figure 4.8) and consists of two switch circuits connected in an arrangement to form a cross coupled inverter pair which is essentially a fast flip-flop. This is preceded by two D flip-flops which latch the read/write, (R/WIN) and address available (EOC) status of the asynchronous port. After arbitration between the two ports has taken place one of the two flip-flops following the switches is clocked which provides a pulse at the output to be fed into the delay circuitry. From the delay circuitry the required signals for the DRAM'S as well as for multiplexing the high and low order address bits are provided. These signals are then fed into a multiplexer (which multiplexes between ports), and from there to external and internal circuitry as required. In addition, the delay circuitry also provides signals for resetting the flip-flop which initiates them for a new memory access.

The synchronizer operation can be seen in figure 4.18. Initially we have an asynchronous port access with EOC going high (R/REF goes low just after the EOC pulse but it will be queued until the next access). When EOC goes high we see the output of its switch (OSWRW) going high and the output of the associated D flip-flop (ORW) also going high. With ORW going high the multiplexers are set to the asynchronous port (ENAL6 goes low) and the control signal RAS, ENA8, and CAS are generated successively. Once the asynchronous access has been completed we see that the synchronous access has been queued. The result being that OSWRR,

figure 4.18 Sync1 Hg Simulation

ORR, and ENA16 go high, and the control signals RAS, ENA8, and CAS are pulled low again. In the second half of the simulation the same thing is repeated only this time synchronous access takes place first and the asynchronous access is queued.

4.5.4 Scan Circuit

The scan circuit consists of six D flip-flops and nine 2-to-1 multiplexer as seen in figure 4.13. This can be divided into two parts. The first part has three signals that can be read and fed through to the internal circuitry or programmed. The second part has three signals that can be read only. The three programmable nodes are used to control the internal multiplexing and counter loading whereas the three readable nodes are used to gain access to the Johnson counter outputs and determine the overall state of the memory control unit.

The operation of the scan circuit is seen in fig. 4.19. TESTCK controls the clocking of the flip-flops while TEST determines whether data is fed from flip-flop to flip-flop (serial) or from the internal nodes into the flip-flops (parallel). TESTIN goes to the D input of the first flip-flop and therefore determines the data that is loaded. RESET controls the reading/programming of the first three nodes, IN (1 - 6) are the internal node values, TESTOUT is the output of the last flip-flop and is used to read the internal node values out serially. OUT (1-3) are the values that internal nodes are programmed to. In the first stage TEST is held high so that data is fed serially between flip-flops, RESET is high so that

Figure 4.19 Scan Circuit Hg Simulation

internal nodes OUT (1-3) will be programmed. We then change TESTIN between high and low and observe the changing output after each TESTCK rising edge. The signals IN (1 - 6) are changed at 7.2u to show that they have no effect on the outputs in this mode. After three more clock cycles the first "0" input has propagated through the first three stages. At this point RESET is brought low and we observe that OUT (1-3) change to the values of the D input as is expected. The clock is then continued for a number of clock cycles and we can see the data coming out of TESTOUT. At 9.8u TEST is brought low and we can observe that OUT (1 - 3) now follow IN (1 - 3) as is expected.

While designing the memory controller unit the work on the design environment was also being completed. Due to the short period of time from receipt of the Metheus (October) to the submission deadline (December) it was necessary to implement the designs while skipping the schematic capture and simulation stage. The result being that the functional block design was carried out originally on rough drawn schematics (as one would generally do), and then directly laid out with no schematic capture or schematic simulation. Therefore the simulations that were carried out were from the actual circuit layout with interconnect capacitance included.

The chip floor plan for the memory controller was carried out in parallel with the functional block design. Because of the relatively large size of the cells (especially flip-flops) it was necessary to pack the cells in bit slice arrangement in a number

of places, to reduce the number of signal lines. The result (see figure 4.20) was that we were able to incorporate the complete controller on a single chip with additional silicon available for test circuitry.

4.6 Fabrication and Testing

The chip design and simulation was completed in the period from October to December and was then submitted to CMC. The chip was then processed by the people at CMC and sent to Northern Telecom in a multiproject chip form. Four prototypes of the memory controller were received in late April. During the period from December to April work on a special LCD controller and 3u cell library was carried out at Siltronics. From April until August, the testing (covered in chapter 5), documentation of Standard Cell Library, and completion of thesis was carried out.

4.7 Summary

In this chapter we have covered the design of the memory controller for the multi-mode Data Acquisition Unit. In the introduction a short description was given on the Data Acquisition Unit's predecessor, the Digital Waveform Recorder along with some of the aims of this project. Following the format of our design methodology the device requirements of the overall system and memory controller itself were covered in section two. Section three gave an overview of the system

figure 4.20 Memory Controller Chip Floor Plan

architecture and functional block specifications with section four giving a more detailed look at the synchronizer and scan circuits. Information on additional functional blocks was presented in section 5, and a brief description of the timetable for fabrication and testing was given in sections 6.

Once each section of the memory controller had been designed and simulated they were connected together to form the completed chip. This chip could then be simulated right down to the transistor level using Hg. In the next chapter we will compare some of the complete-chip simulations with the actual measured data. Also some information on the standard cell test vehicle results will be presented.

CHAPTER 5

TEST RESULTS

5.1 Introduction

In this chapter the experimental results for the standard cell test vehicle and the memory controller unit will be presented. The first section will deal with the test vehicle, which was submitted in August, 1984. This chip contains the standard cells which had been designed up to that date. It was received back from CMC in late December with testing being carried out between January and April, 1985. In the following section, test results for the memory controller unit will be presented. Once the complete memory controller layout had been finished, it was run through the program PHLEX on the Metheus to extract electrical parameters. With the electrical parameters extracted the complete chip could be simulated using Hg. These simulations are, however, quite slow with a single timestep taking several minutes. By simulating the complete chip we were able to determine a number of overall errors and make corrections. In this section we will compare results of the simulations with actual experimental data from the HP test equipment. Finally, a short summary will be given.

5.2 Standard Cell Test Vehicle

The standard cell test vehicle was designed with the intention of verifying each cells functionality only. To do a complete characterization it would require a number of the cells with varying loads with measurements on a large number of chips. Our testing was limited to four 40 pin chips. At the time of submission, the following cells were incorporated on chip:

inverter 1	nor3
inverter 2	nor4
inverter 4	passgate
inverter	pass3gate
inverta	mux
invertb	latch
nand2	dflip
nand3	dffpet
nand4	inputpat
nor2	outputpad

In addition, two special cells were also incorporated. These included a master slave JK flip-flop which was used as a high speed divide-by-two stage for the system controller of the data acquisition system, and a 4 bit ripple counter.

A chip photograph with associated pad connections can be seen in figure 5.1. The testing that was carried out before the test equipment was set-up, and was done on a bread board with a waveform generator having 2 MHz maximum frequency and a 20 MHz

WAFER NUMBERS _____

DESIGN FILE REFERENCE _____

PACKAGE	<i>IDK 4 OF 1-192G</i>		LID	<i>C-193-175-35M</i>	
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 5.1 Standard Cell Test Vehicle Chip

oscilloscope with 50ns resolution. When testing this chip, if some of the input pins were toggled, the power rails were shorted and the IC destroyed. An extensive search for shorts using the probe station proved unsuccessful, and as of present the cause is unknown. However, it was still possible to test a number of the cells as seen below.

The first circuit tested was that of an input pad (with associated protection devices) and an output pad with two cascaded inverter stages. The input was driven at the maximum 2 MHz frequency and no propagation delay was observable with 50ns resolution. This demonstrated the functionality of the input and output devices. However, the maximum operating frequency was not determined.

In the next test, inverta and invertb, which were cascaded together were tested and a propagation delay of 0.1us was observed. This verifies their functionality because without these inverters (as seen in the first test), any delay was undetectable. An inverter string of inverter1, inverter2 and inverter4 was then tested. Inverter1 was a minimum geometry inverter with inverter2 and inverter4 being two and four times minimum geometry. The result was a 0.15us delay which is reasonable considering the additional inverter stage and the reduced drive of the final inverter4 stage as compared to the inverta stage of the previous test.

A divide-by-two circuit made up of a master JK flip-flop with

Q and Qb fed back as inputs and CK and Ckb driven from cascaded inverters was then tested. The output signal was the input signal divided by two and had no problems operating at the 2 MHz maximum operating frequency. This circuit had been designed for 50 MHz operation and when tested in the system controller was able to achieve 40 MHz.

The last circuit tested successfully was a positive edge triggered D flip-flop with setbar and resetbar. This flip-flop had Qb fed back to the D input to give a toggle flip-flop. This circuit also functioned correctly and divided the input frequency in half, as was expected. This was operated at 2 MHz and testing of the reset and set functions showed their correct operations. Unfortunately with the test equipment available, the switching speed of the set and reset could not be determined.

With the test results obtained we know that at least some of the cells are functional. In further testing on the system and memory controller the operation of a larger number of the cells was validated.

5.3 Memory Controller Unit

The memory controller (see figure 5.2) circuit was tested using the complete test resources available at the university with the HP test equipment proving to be an extremely valuable tool. In this section, some of the results of this testing will be presented and compared with the simulation results. The aim of

WAFER NUMBERS _____

DESIGN FILE REFERENCE _____

PACKAGE	<u>IDK 40FI-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 5.2 Memory Controller Chip

this testing was to demonstrate functionality of the chip, however, some detailed timing will also be presented.

5.3.1 DAT 1

The first test carried out was to verify the functionality of the scan test circuit. All the inputs except TEST, TESTOUT, TESTIN, and TESTCK were held low. TEST was held high to put the test circuit in the serial shift register mode. The circuit was then clocked by TESTCK with the input data being changed from "0" to "1", as seen in figure 5.3. The output data on TESTOUT can be seen coming out after 6 clock cycles which is expected for a shift register with 6 stages. No delays are seen on this timing diagram due to the sampling nature of the test equipment, however, this circuit was clocked up to 15.3 MHz and was still functional.

A simulation using the same inputs on the Metheus proved identical (see figure 5.4) with the ~~outputs~~ again coming out after 6 clock cycles.

5.3.2 DAT 2

The next test was to verify the operation of the overall system timing, and the generation by the johnson counter of the C0, C2, and C6 clock pulses. In this rather complicated simulation (see figure 5.5) the system is first reset (RESET goes low) to put the johnson counter in the proper 0 state. A single clock on TESTCK, with TEST low, loads the state of the johnson counter into the scan circuit. TEST is then pulled high to put

DAT1

figure 5.3 Dat 1 Test Results

Figure 5.4 Dat 1 Hg Simulation

DAT2

Figure 5.5 Dat 2 Test Results

the scan circuit in shift register mode, and succeeding pulses on TESTCK will clock the data out on TESTOUT. The system clock CK is then run for a number of cycles to put the system into a new state and the test sequence is repeated again. This is done three times to show that the system is in fact generating the three clock pulses C0, C2 and C6. For the first sequence we see that TESTOUT goes high for the first TESTCK pulse. This is correct as C0 was fed into the last stage of the scan circuit. In the next sequence TESTOUT goes high on the third TESTCK pulse indicating that C2 is high (C2 is fed into the third last stage). Finally, in the last sequence, TESTOUT goes high after the second TESTCK pulse which indicated C6 (which is fed into the second last scan circuit stage) is high.

A simulation with the same inputs using the H₂ simulator proved to be different. The simulation on H₂ (see figure 5.6) seemed to be clocking one clock later on TESTCK. Examination of the C0, C2, and C6 signals (see figure 5.7) showed that they were operating correctly. After a number of further simulations it was determined that the input multiplexers in the circuit, although operating on the actual chip, were not being simulated correctly. These multiplexers consist of oppositely clocked transmission gates with their outputs connected together. When the TEST input went high the multiplexers were changing from parallel to serial mode, however, the outputs were not driven to the correct state until the following clock cycle. This further emphasizes the problems encountered with transmission gates using

figure 5.6 Dat 2a Hg Simulation

Figure 5.7 Dat 2b Hg Simulation

Hg.

5.3.3 DAT 3

In dat 3 a test of the asynchronous port and multiplexers was carried out (see figure 5.8). In this test TESTIN is held low and successive clock pulse on TESTCK, clock zeroes into the first three stages of the scan circuit. The signals ENA16, ENA8 and LD are thus programmed low so that the multiplexers will be set to the high order bits of the asynchronous port. High order bits, A12, A13, A6, A7, and A8 (these numbers come about because of the way the ports were put on the layout) are then toggled with clock pulses on EOC latching in the data. The outputs, OUT (3-7) changed as was expected (see figure 5.9). Three of the higher order bits were not functional. Simulation previous to testing had shown this and it was determined that three of the higher order bits were not correctly connected to the circuitry. This problem would not have occurred except for the fact that the simulations of the overall chip were not completed before submission time.

The simulation results for the same inputs (see figure 5.10) gave identical results to those in figure 5.9, further verifying the operation of the asynchronous port.

5.3.4 DAT 4

In dat 4 a test of the synchronous port was carried out. In this test (see figure 5.11) a "0", "1", "0" was loaded into the scan circuit by signals on TESTCK and TESTIN. This programs the

DAT3IN

figure 5.8 Dat 3 Input Test Signals

DAT3OUT

figure 5.9 Dat 3 Output Test Signals

figure 5.10 Dat 3 Hg Simulation

DAT4IN

figure 5.11 Dat 4 Input Test Signals

multiplexers for the higher order bits of the synchronous address and enables the loading of the synchronous up counter, so that the addresses will be fed right through to the multiplexers. The addresses A12, A13, A6, A7, and A8 were then toggled. The resulting output addresses OUT[3-7] were then toggled at the 9, 10 and 11 clocks as seen in figure 5.12. The change in states on the output during earlier clocks is due to the fact that as the control data is being loaded in (1, 0, 1,) the multiplexer states are being changed and therefore the changes on the outputs.

The complementary simulation (see figure 5.13) agreed very closely even with the same type of output switching when the control data is loaded.

5.3.5 DAT 5

In dat 5 a test was done on the synchronizer to determine its correct operation and generation of DRAM control signals. The test consisted of starting an asynchronous access by clocking EOC with alternating write and read accesses (see figure 5.14). The result was the generation of the correct RAS, CAS, and R/WOUT pulses. The falling edge delay for RAS to CAS cannot be seen due to the data analyzer sampling frequency, but an oscilloscope photograph of the signal (see figure 5.15) showed the correct operation with CAS delayed by 90ns from RAS. The 30ns difference from the SPICE simulations can be explained by the loading on the output pad which was not simulated for. The maximum access frequency for the asynchronous port was determined to be 855 KHz

DAT4OUT

Figure 5.12' Dat 4 Output Test Signals

Figure 5.13 Dat 4 Hg Simulation

DATS

figure 5.14 Dat 5 Test Results

figure 5.15 Dat 5 Timing Delays

compared to the simulated frequency of 1MHz. This is due to the fact that the inverter chain which causes the reset of the port to enable a new access is slower than simulated. At present, the only explanation is that of simulation inaccuracy. In fact it has been learned that the circuit was actually fabricated in 4um technology rather than 5um which, due to the non-linear scaling of the transistor effective channel width, would actually decrease the inverter delay.

The simulated results as seen in figure 5.16 show the same waveforms with successive write and read accesses taking place.

5.3.6 DAT 6

In dat 6 a test of the synchronous up counter was made. As seen in figure 5.17 the multiplexes were set to the low order bits of the synchronous address and the counter load was disabled so the counter would be free running. The system clock (CK) was then started and outputs were observed (see figure 5.18). As seen, the counter is counting up as expected with only five lower order bits being shown here.

The simulation for these conditions is shown in figure 5.19. We have used C0 as our clock instead of CK, which reduces simulation time by sixteen times. The outputs B0 - B4 are internal outputs of the counter. The external output could not be observed correctly due to the simulator giving unreliable results when the ENA8 signal is programmed high. When ENA8 is programmed high the simulator actually causes it to oscillate at

figure 5.16 Dat 5 Hg Simulation

DATA IN

Figure 5.17 Dat 6 Input Test Signals

DATA6OUT

figure 5.18 Dat 6 Output Test Signals

figure 5.19 Dat 6. Hg Simulation

a high frequency. The only explanation for this result seems to be a feedback in an inverter stage thereby giving an oscillator. Additional simulation narrowed the problem down to only one possible inverter, however, an extensive search for the feedback path was unsuccessful.

5.3.7 DAT 7

In dat 7 the synchronizer operation during a synchronous access is observed (see figure 5.20). The system clock (CK) is run and a synchronous access is generated every 4us. The DRAM control signals RAS, and CAS can be observed going low with the R/WOUT signal being held high.

The simulation (see figure 5.21) gives the same results thereby validating the overall operation of the synchronizer.

In addition to the above results, measurements on the power dissipation showed it to vary from 10.5uw dc to 23.5 uw with the external clock running at the nominal 4 MHz frequency. Observed maximum frequencies of 15 MHz for the shift register agrees with expected maximums suggested by Northern Telecom.

5.4 Summary

In this chapter the results of the testing of the Standard Cell test vehicle and testing and simulations of the Memory Controller chip were presented. The Standard Cell test vehicle demonstrated the functionality of a number of the Standard Cells. However, due to shorting of the chip, not all the cells could be

DATA?

figure 5.21 Dat 7 Hg Simulation

tested. In the Memory Controller chip extensive testing showed that the majority of the chip was functional with some minor errors which could have easily been corrected, given a longer simulation time before fabrication.

The positive results obtained with the memory controller show that the design environment is correct by the fact that we have achieved our overall goal of a working chip. In the next chapter we will present conclusions of the thesis as well as future considerations.

CHAPTER 6

CONCLUSIONS

6.1 Summary

The aim of this Thesis was to develop a design environment in which the design of complex IC's could be carried out with a high probability of success on the first try. The design environment which was developed is based on a design methodology which will enable the designer to proceed in a step by step manner to accomplish the end product of a working IC. This design method is particularly suited to the design tools that we have available at Carleton. The design environment is also based on a standard cell approach where the designer with little or no experience, as well as the experienced designer, can realize a working chip.

The design environment incorporates a network of design tools which enable the users to do low level design tasks on the numerous smaller machines while the higher level functions can be accomplished on the advanced workstation. In addition to research projects, the environment will also accommodate teaching requirements with the students doing the majority of the work on the low end terminals and an instructor or faculty member completing some of the tasks better suited to the advanced workstation.

The design tools available offer the designer a wide variety of software packages which enable designers to carry out most of the tasks that they will need. This includes functions, such as design rule checking, electrical rule checking and electrical parameter extraction. By using these tools as specified by the design methodology the designer can proceed in a hierarchical manner, thereby completing the design while reducing any possibilities of errors. The Design tools are centered around a Metheus lambda 750 workstation which is supplemented by a large number of Corvus Concept workstations and two high speed numerical processors for computational intensive tasks.

The design environment was verified by the design of a memory controller unit which was designed, laid out, and simulated within a three month period, from October to December. The design of this 2900 transistor chip further emphasizes the feasibility of this environment which enables a single designer to accomplish this scale of complexity chip in such a short period of time.

The fabricated memory controller chip, upon receipt from CMC, was tested and found to be almost completely functional. A few minor errors were incurred, however, these can be explained by the fact that simulations on the overall chip were not quite completed before the simulation deadline and therefore the errors were not detected. The very good results obtained further demonstrates the ability to design a complete working chip on the first iteration which is becoming almost essential in university

as well as industrial environments.

6.2 Future Work

It has been recently announced by CMC that as of September, 1985, Northern Telecom's 3u CMOS process will be made available to the universities. This is opportune in that over the four month period from January to April, 1985, work was carried out by Richard Vallee and the author in the development of a 3u standard cell library. This new library is based on cells which have symmetric n and p channel transistor aspect ratios rather than symmetric rise and fall times, however, the fan-in of the cells is limited to four to prevent drastic differences in rise and fall times. Because the cells have equal size transistors, the cell size is reduced as well as giving complementary functions by simply inverting the p and n type diffusions.

The cells come in a number of varieties. The basic cells are all single height cells of 68u with I/O locations on an integer number of the 4u grid. The contacts which are present on I/O's of the present library have been removed, and are now placed in the interconnect channel to further enhance size reduction. In addition to single height cells are double height, variable height custom macro cells, and megamacro ROM and RAM arrays. Special cells such as low power crystal oscillator and LCD phase drivers are also being developed. The total number of cells at present is approximately 200, which should greatly enhance the design capabilities at the university. The new cells

with their improvements in design and smaller geometries will give at least a six-fold increase in packing density, thereby making designs in the 20,000 transistor range easily obtainable.

Recently, CMC also announced that a new round of workstation grants was beginning. This means that should Carleton receive a new workstation, the ability for implementing designs will be greatly enhanced. At present, with four or five graduate students doing designs, the system becomes overloaded and making submission schedules is difficult. With a new workstation this problem should be alleviated a great deal.

The design tools present at Carleton go a long way in doing IC designs, however, we are still limited in a number of areas. If chips of increasing complexity are to be designed, it is essential that some autorouting and autoplacement routines be incorporated within the environment. By having these tools, manual errors in placing and routing are reduced, if not eliminated. In addition, the time required in doing the actual place and route can be reduced from weeks to days, if not hours. Also of importance is the interfacing of the design tools to the test tools. By doing this, the designer will not have to repeat the task of generating test vectors for both simulations and actual chip testing. Finally, work on the design of additional analog functional blocks is also essential in providing even more flexible design capabilities.

This thesis was a pilot project to develop a working

environment enabling IC design at Carleton. At present the designs have been limited mainly to projects for learning purposes. Carleton and Ottawa Universities recently obtained a strategic grant to do work on the development of a mobile radio system. A portion of this project which will be carried out by the Carleton electronics department will be concerned with integration of parts of the mobile radio onto silicon. With the development of the new 3u standard cell library and the design environment as presented in this thesis, the first steps have already been made. Future faculty and students working on this or similar projects will now have both a method and a means to aid them in their endeavors.

LEO45 performs a batch hierarchical and incremental layout rule check of a phled design. LEO45 opens each .phl file in the design to determine when LEO45 was last run. A cell is re-checked only if it, or its descendants in the hierarchy, have been modified since the last check. In the case where a cell has not itself been modified since the last check, but some of its descendants have, only the parts of the cell in the vicinity of the modified descendants are rechecked. When a cell is checked, design rule violations are added to the .phl file in the form of a bright blotch of paint and a label, both on the violation layer. In addition LEO45 can also be invoked from the phled layout editor to interactively check a portion of the design currently being edited. Interactive checks are non-hierarchical and non-incremental.

PHLED is an interactive, color graphics editor that allows you to create mask files for use in fabrication of VLSI chips. It provides multiple bit planes and color graphics to aid in designing the various layers of the chip. A versatile set of commands and a "mouse" controlled cursor allows you to interactively create physical layout cells, which can then be combined hierarchically to form the final design for the chip. In addition, PHLED provides a utility that allows you to convert physical layout (.phl) files into the industry standard Caltech Intermediate Form (CIF) format.

PHLEX converts a physical layout in a .phl format into a netlist file in a .sim format. The simulators, electrical rules

REFERENCES (Cont.)

13. R. Beresford, " Smart Memories Seek Honors in Proliferating Small Systems ", Electronics, pp. 89-98, July 1982.
14. M. Hurtado, " Dynamic Structure and Performance of Asymptotically Bistable Systems ", Washington University D. Sc. dissertation, 1975.
15. J. Thomas, C. Chaney, F. U. Rosenberger, " Characterization and Scaling of MOS Flip-Flop Performance in Synchronizer Applications ", Proceedings of the Caltech Conference on VLSI, January 1979.
16. I. Catt, " Time Loss Through Gating of Asynchronous Logic Signal Pulses ", IEEE Transactions on Electronic Computers, pp. 108-111, February 1965.
17. M. Evans, C. Carinalli, " DP8400 Family of Memory Interface Circuits ", National Semiconductor Application Note 302, March 1983.
18. Eichelberger and Williams, " A Logic Design Structure for Testability ", Proceedings of the 14th Design Automation Conference, 1977

APPENDIX I

SOFTWARE REFERENCES

CC.DCOMM is a Corvus Concept program to allow file transfers between itself and other computers

CIF2PHL is a translation program which will convert a layout in Caltech Intermediate Form (.CIF extension) into the Physical Layout Editor form (.phl extension).

COMPARE uses a graph isomorphism algorithm (speeded by comparing labels on nodes) to compare two netlist files, determine whether they represent the same circuit, and if not, determine what the differences are. The netlists must be in the .sim format and typically comes from either a schematic (via SCH2SIM) or from a physical layout (via PHLEX).

ERC checks a simulation netlist for various kinds of electrical errors. These include basic continuity and signal propagation checks, as well as several MOS-specific rules such as threshold drops and rises. Pullup/pulldown ratios are computed only if depletion pullups are present in the circuit. The netlist file must be in the .sim format. ERC prints various statistics and sends a summary of the errors found to the standard output. Graphical feedback information is written into a file with a .erc extension.

Hg is a logic simulator which performs switch-level simulations on transistor-level circuits. The switch-level simulation takes into account channel resistances, transistor threshold levels, and parasitic capacitances, in order to provide timing information during the simulation process.

HILO is a gate-level logic simulator. It can perform nominal delay simulation, dual delay simulation, fault simulation and fault list interrogation, and automatic test generation (ATG). HILO simulates the extracted schematics in its data base. HILO command and waveform files (created by the user) specify the kind of simulation and other information.

CIFDRAW permits an IC designer to create Caltech Intermediate Form (CIF) symbols as a means of describing an integrated circuit. Based on the book by Mead and Conway, the designer uses the CONCEPT as a graphical editor creating rectilinear rectangles using a pair of graphic cursors (one is static, and the other is dynamic), which are controlled by the keyboard function keys. The program supports eleven layers making it suitable for both NMOS and CMOS processes. The program is completely function key driven, with appropriate function key annotation. The key labels provide easy prompting with little instruction for new users. A locally attached Hewlett-Packard (HP) 7221 plotter is supported via a two dimensional graphics package, which supports three graphics devices: a HP7211 plotter, Concept graphics screen, and a DEC LA120 equipped with a Selanar graphics board.

LEO45 performs a batch hierarchical and incremental layout rule check of a phled design. LEO45 opens each .phl file in the design to determine when LEO45 was last run. A cell is re-checked only if it, or its descendants in the hierarchy, have been modified since the last check. In the case where a cell has not itself been modified since the last check, but some of its descendants have, only the parts of the cell in the vicinity of the modified descendants are rechecked. When a cell is checked, design rule violations are added to the .phl file in the form of a bright blotch of paint and a label, both on the violation layer. In addition LEO45 can also be invoked from the phled layout editor to interactively check a portion of the design currently being edited. Interactive checks are non-hierarchical and non-incremental.

PHLED is an interactive, color graphics editor that allows you to create mask files for use in fabrication of VLSI chips. It provides multiple bit planes and color graphics to aid in designing the various layers of the chip. A versatile set of commands and a "mouse" controlled cursor allows you to interactively create physical layout cells, which can then be combined hierarchically to form the final design for the chip. In addition, PHLED provides a utility that allows you to convert physical layout (.phl) files into the industry standard Caltech Intermediate Form (CIF) format.

PHLEX converts a physical layout in a .phl format into a netlist file in a .sim format. The simulators, electrical rules

checker (ERC), and netlist comparator (COMPARE) use this file to verify the design. PHLEX operates on MOS type layouts with either 90 or 45 degree geometries. PHLEX converts the primitive devices in the layout directly into devices and nodes in the netlist file. The resulting netlist file is thus automatically flattened to the primitive device level. PHLEX also provides a switch that allows the netlist file to be produced in the .imf format for use with SPICE.

PHLPLOT reads a layout description, in the physical-layout (.phl) format, and produces a plot on one of several output devices, including the Versatec 8000 series matrix printer/plotters and the HP 7580, and 7585 8-pen plotters.

PHL2PHL is a translator which will scale a Physical Layout Editor file (.phl extension) by some integer value.

SCHEDIT forms the heart of the logic design system. It is an interactive, color graphics editor that allows you to create schematics of logic circuits using standard logic symbols and wiring commands. You can create and modify your logic designs at the block, gate, or transistor level, or using a combination of the three. The ability to assign physical and electrical properties to the symbols, either directly or through libraries of standard symbols, allows you to generate netlist models of your logic design, which you can then simulate with the logic and circuit simulators.

SCHPLOT reads one or more schematic editor (.sch) files and

plots them on either a Versatec 8000 series matrix printer/plotter, or an HP 7580/7585 plotter.

SCH2HILO translates the schematic file (*.sch) into an intermediate netlist file (*.hilo). When HILO is run it extracts the hilo properties from the schematic and symbol files, then executes HILO to compile the resulting circuit description into the HILO data base.

SCH2IMF translates the schematic file (*.sch) into an intermediate netlist file (*.imf). When spice is run on a file (*.imf) it will create an input file (*.spi) for spice. The general spice output file (*.spo) will finally be converted into a graph file (*.gr) for the simulation grapher.

SCH2SIM reads schematic files (.sch) and produces netlist files (.sim). The output files have a .sim extension. Many intermediate hierarchical netlist files are also optionally produced.

SIMGR is an interactive graphing program that plots data from an input file onto the display screen or on a plotter. The grapher automatically scales the data, generates labeled axes, and uses color to distinguish the traces from one another. The grapher accepts the output files produced by the simulators Hg, SPICE, and HILO. It may also be used to graph data from other simulators, provided that the data is supplied to it in the required format.

SPICE is a general-purpose circuit simulation program for

non-linear dc, non-linear transient, and linear ac analysis. Circuits may contain resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, transmission lines, and the four most common semiconductor devices: BJT's, JFET's, MOSFET's, and diodes.

SPLICE is a mixed-mode simulation program for large scale integrated circuits. It performs concurrent electrical and logical simulation using event driven selective-trace techniques. The electrical analysis uses an Iterated Timing Analysis, which performs accurate electrical waveform analysis much faster than SPICE. The logic features a MOS-oriented state model and a fan-out dependent delay model, which handles bidirectional transfer gates in a consistent manner.

SYMED is a general purpose graphics editor that you use to create new logic symbols. These symbols can then be associated, hierarchically, with lower level schematics made up of transistors, logic gates, or a combination of the two. In this manner, you can create high-level symbols for specific functions in your design. You can also use SYMED to create block diagrams and other graphics, which can be included in your design documentation.

SYMPLOT reads one or more symbol editor (.sym) files and plots them on either a Versatec 8000 series matrix printer/plotter, or an HP 7580/7585 plotter. SYMPLOT functions through a shell script that reads in .sym files, creates a

temporary .sch file, and calls the SCHPLOT command to plot the symbol from the temporary .sch file.

APPENDIX II

FUNCTIONAL BLOCK DESIGNS

AII.1 Latlat

Latlat includes four sets of latches (see figure AII.1) which are the higher order bits for the asynchronous and synchronous ports, the bits for block size selection, and additional bits for clocking through the higher order bits of the synchronous address. Latlat is also a bitslice approach and has the identical pitch and signal feedthroughs as Latct. By doing this we are able to group Latct and Latlat blocks together to form all the latches on the input address bus as well as the programmable up counter used for synchronous port operations.

The operation of latlat can be seen in figure AII.2. CRW, CDIS, and CDEC are clocks for the latches to the asynchronous port, synchronous port, and block size selection respectively. LD is used to load the up counter, A is the address input, CT is the counter output, RW is the asynchronous port output, DIS is the output after the first stage of synchronous port latches, and DEC is the block size latches output. At 0.1u and 0.4u we can see the loading of a "0" and a "1" into the asynchronous port determined by the a input and CRW. At 0.9u and 1.3u we see the same operation performed on the synchronous port. This is under the control of A, CDIS, and LD, with LD occurring after CDIS to

LATLAT

figure AII.1 Latlat Schematic

figure AII.2 Latlat Hg Simulation

will therefore not be coverable in detail here.

AII.3 Dectest

Dectest (see figure AII.5) consist of the lower four bits of the down counter as well as the zero detect circuitry. The circuit consists of four T flip-flops connected serially, in addition each stage has alternating 2-input NAND'S and 2-input NOR's to provide a "0" out of the last stage when zero count has been reached. This zero count is then "nored" with a LD_b signal (generated from the johnson counter) which finally clocks a D flip-flop who's outputs are used to reload the counter. Once the LD signal goes low the D flipflop is reset and the circuit can continue counting down until a zero count is reached again.

The operation of this circuit can be seen in figure AII.6 where the clock (CK), and load (LD) signals are generated from the johnson counter, DEC (0-3) are the outputs of the counter stages, ZERO is the zero detect output, and set is the signal generated to reload the counter. The counter is counting along until 5.7u when, a zero count is reached. This generates a zero count signal (ZERO) and when LD goes high the SET signal goes high which causes the reloading of the counter. Once the LD signal goes low the D flip-flop is reset.

AII.4 Dec2p

Dectest provides the circuitry for the four lower order bits of the 12 bit counter. As seen in the previous section on a zero

figure AII.3 Muxmux Schematic

Figure A11.4 Muxmux Bg Simulation

will therefore not be coverable in detail here.

AII.3 Dectest

Dectest (see figure AII.5) consist of the lower four bits of the down counter as well as the zero detect circuitry. The circuit consists of four T flip-flops connected serially, in addition each stage has alternating 2-input NAND'S and 2-input NOR's to provide a "0" out of the last stage when zero count has been reached. This zero count is then "nored" with a LDb signal (generated from the johnson counter) which finally clocks a D flip-flop who's outputs are used to reload the counter. Once the LD signal goes low the D flipflop is reset and the circuit can continue counting down until a zero count is reached again.

The operation of this circuit can be seen in figure AII.6 where the clock (CK), and load (LD) signals are generated from the johnson counter, DEC (0-3) are the outputs of the counter stages, ZERO is the zero detect output, and set is the signal generated to reload the counter. The counter is counting along until 5.7u when, a zero count is reached. This generates a zero count signal (ZERO) and when LD goes high the SET signal goes high which causes the reloading of the counter. Once the LD signal goes low the D flip-flop is reset.

AII.4 Dec2p

Dectest provides the circuitry for the four lower order bits of the 12 bit counter. As seen in the previous section on a zero

DECTEST

Figure AII.5 Dectest Schematic

figure AII.6 DecTest Hg Simulation

count these bits are all set high. However, to have a channel size selection we require some programmability. This is provided by the 4 higher order bits which allow a selection from 512 to 4K bytes. Dec2p (see figure AII.7) contains two of the higher order bits. Each bit consists of a T flip-flop, alternating NAND's and NOR's for zero detect and NAND's feeding into SETb and RSTb of the T flip-flop to enable programming.

The operation can be seen in figure AII.8 where CK comes from the Johnson counter LD would be generated from the zero detect circuitry (equivalent to SETb of the previous simulation), A (0 - 1) and dec (0 - 1) are the input and output signals respectively. As can be seen, counting is taking place normally until the ld signal goes high which loads in the data on A0 and A1 showing the proper action of the circuit.

figure AII.7 Dec2p Schematic

figure AII.8 Dec2p Hg Simulation

APPENDIX III

CMOS FLIP-FLOP ANALYSIS

In this appendix an analysis of a CMOS cross-coupled inverter pair is carried out to determine parameters for synchronization failure.

The cross-coupled pair used in the synchronizer (see figure AIII.1) can be simplified to that of figure AIII.2, if we consider that in the metastable state the inputs RWN and R/RRN will have had sufficient time to be pulled low, and the outputs of their associated inverters will be high.

If we consider the analysis for the simplified case where the transistor aspect ratios are chosen such that the transistor betas (k) are equal, then in the metastable state the cross-coupled pair will sit at the inverter switching thresholds of $V_{dd}/2$. At the inverter switching threshold the p and n channel transistor will both be in saturation, and for the left side inverter we can write

$$I_{pu} = k (V_2 - V_{tp})^2$$

$$I_{pd} = k (V_2 - V_{tn})^2$$

where $V_{tp} = V_{tn} = V_t$ is the threshold voltage. If V_2 is now increased by a small voltage dV_2 we will observe a current I_g given by

$$I_g = I_{pu} - I_{pd} = k[(V_2 - dV_2 - V_t)^2 - (V_2 + dV_2 - V_t)^2]$$

SYNCTEST

figure AIII.1 Synctest Schematic

CMOS FLIP-FLOP

figure AIII.2 CMOS Flip-Flop Schematic

$$I_g = -4\kappa (V_2 - V_t) dV_2$$

as dV_2 approaches zero, we can write

$$dI_g/dV_2 = -4\kappa (V_2 - V_t)$$

although I_g is a function of both V_2 and V_1 the dependence on V_1 is proportional to the channel-length modulation parameter and for a first order analysis can be neglected. The current I_g is also a function of the total gate capacitance (C_g) of M3 and M4, and is given by:

$$I_g = C_g dV_1/dt$$

We can now equate the last two equations and write

$$dI_g/dt = -4\kappa (V_2 - V_t) dV_2/dt = C_g d^2V_1/dt^2$$

where the coefficients on each side can be written as

$$I = \text{charge in transit/saturation transit time} = C_g(V - V_t)/T_s \\ = \kappa (V - V_t)^2$$

giving the resulting differential equation.

$$(-4/T_s) dV_2/dt = d^2V_1/dt^2$$

At time $t=0$ the same analysis can be made for the right side of the cross-coupled pair, such that dV_1/dt approximately equals dV_2/dt and we can then write.

$$-4/T_s dV_2/dt = d^2V_2/dt^2$$

The solution to this equation is an exponential diverging from equilibrium and is given by

$$V = V(0) e^{-4t/T_s}$$

where $V(0)$ is the initial equilibrium voltage. If we compare this last equation to our equation in section 4.6 we see that P equals $-4/T_s$. Where T_s is given by

$$T_s = L^2/(uV_{ds})$$

and L is the transistor channel length (4 microns), μ is the carrier mobility (which we will take the worst value for holes of $750 \text{ cm}^2/\text{Vs}$), and V_{ds} is the drain source voltage. A SPICE level 2 simulation for V_{ds} determined it to be 1.82 V (at equilibrium) for the flipflop used in the synchronizer (see figure AIII.3), with the resulting values of $1.19\text{E-}10 \text{ s}$ and $3.36\text{E}10 \text{ s}^{-1}$ for T_s and P respectively.

Figure AIII.3 Flip-Flop dc Bias

END

1 5 0 4 8 6

FIN