I, Vishal Chadha, hereby submit this work as part of the requirements for the degree of:

Master Of Science

in:

Computer Engineering

It is entitled:

Design And Implementation Of A Second Generation Logic Cluster For Multi-Technology Field Programmable Gate Arrays

This work and its defense approved by:

Chair: Dr. Fred R. Beyette Jr.
Dr. Karen Tomko
Dr. Wen-Ben Jone
Design And Implementation Of A Second Generation Logic Cluster For Multi-Technology Field Programmable Gate Arrays

A thesis submitted to the
Division of Graduate Studies and Research of the University of Cincinnati
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in the Department of

Electrical & Computer Engineering and Computer Science

of the College of Engineering

September 2005 By
Vishal Chadha
B.S., (E&CE), Indian Institute of Technology,
Roorkee, India, 2000

Thesis Advisor and Committee Chair: Dr. Fred R. Beyette Jr.
Design And Implementation Of A Second Generation Logic Cluster For Multi-Technology Field Programmable Gate Arrays

Vishal Chadha

ABSTRACT

In recent years, Multi-Technology applications are gaining increased acceptance due to the feasibility of large complex designs to integrate different technology components like analog devices, Memory, MEMS devices, microwave components etc on to a single silicon fabric. But the turn around time of these designs is still quite high due to the lack of prototyping devices. Though Field Programmable Gate Arrays (FPGAs) have revolutionized programmable/ re-configurable digital logic technology, one limitation of current FPGAs is that the user is limited to strictly digital electronic designs. Thus, they are not suitable for applications that are not purely electronic, such as optical communications, photonic information processing systems and other multi-technology applications mentioned above. While a wide variety of multi-technology devices have been implemented, research in this area has for the most part been limited to systems built with application-specific devices. Though they are well optimized for a specific application, they lack the flexibility associated with generically re-configurable/programmable hardware and are costly with high turn-around time. So in 2002, the novel idea of a Multi-technology Field Programmable Gate Array (MT-FPGA) was proposed [1] to extend the flexibility, rapid prototyping and reusability benefits associated with conventional FPGA technology into photonic and other multi-technology domain and give rise to the development of a wider class of programmable integrated systems. While that work demonstrated that by careful design and integration of
mixed-technology components with digital logic, the benefits associated with conventional FPGAs could be realized in mixed technology systems, the chip design done for that work was not well suited for implementations in modern systems. In particular, the design was not compatible with modern CAD resources that allow for efficient system design device programming. Further, the digital logic clusters did not include some of the specialized logic, like, a dedicated carry-chain for arithmetic operations that has become common in FPGA devices.

In this thesis, research has been done to make the Multi-Technology Logic Cluster (MTLC) design much faster, smaller and versatile by using improved process technology, optimized architecture/floorplanning and adding data processing capabilities (addition, subtraction and multiplication) so that these components match the performance expected from current applications. In other words, this thesis is the next step in the evolution of the MT-FPGA to provide high-performance solutions for complex applications. The first part of the thesis deals with the motivation for research, general design optimization alternatives available and a general description of an Island style MT-FPGA architecture as previously proposed [1]. To provide an overview, various components of the Multi-Technology Logic Cluster (MTLC) like the 4-bit LUT, 8X1 Multiplexers (soft-connects), L3-4.2 Programmable Logic Block (PLB) tree structure, 16-bit bus and the Connection Blocks (CB) are briefly discussed. Second part of the thesis provides an analysis of the speed, area, power, floorplanning, functionality, programming and circuit design aspects of the existing design and provides in-depth suggestions for design optimization. These suggestions based on the general guidelines for performance improvement of the first part form the foundation of the new design structure proposed in this thesis. The third part involves the design verification (functional
and timing) and analysis of the second generation MTLC at the schematic, layout and post-silicon stages of the design. Finally, individual components are characterized and comparative results with respect to the previous design are shown, then the conclusions are drawn from the results and future work is suggested.
I wish to express my sincere gratitude to my thesis advisor Dr. Fred R. Beyette Jr. for his guidance and support on this project. His insight into the various aspects of the design and implementation was most valuable to the successful completion of this research initiative. I would like to thank Dr. Karen Tomko for her advice on the CAD issues in the project. I would take this opportunity to convey my heartfelt love and gratitude to my parents whose unconditional love, hard work and able guidance has made me capable of pursuing things that I cherish. This journey wouldn’t have been possible if not for them and my wonderful sister and friend Bhanu Kaura.

I am thankful to my colleagues, especially Prashant Bhadri, for their support in and out of the laboratory, which helped this project in a big way. Working with them at Photonics Systems Development Laboratory (PSDL) has been a great experience for me.

In particular, I would like to thank Guorong Ma and the CDK team at the North Carolina State University for providing and supporting the Cadence Design Kit used throughout this project. Without your support this project would not have been possible. I would like to convey my special thanks to MOSIS foundry service for fabricating the project. I addition, would like to thank Anindo Mukherjee, Kumar Anand, Tarun Joshi and Amit Sharma for their encouragement and support. In the end, would also like to acknowledge Mr. Chris Isbell and Mr. Rob Montjoy for helping me during various phases of my research.
# TABLE OF CONTENTS

LIST OF FIGURES........................................................................................................ix

LIST OF TABLES...........................................................................................................xi

GLOSSARY OF TECHNICAL TERMS .................................................................xii

1. INTRODUCTION .................................................................................................1

1.1 MOTIVATION AND RESEARCH ...............................................................1

1.2 GENERAL DESIGN OPTIMIZATION TECHNIQUES EXPLORED..............1

1.2.1 Technology Scaling .................................................................................2

1.2.2 Use of Logical Effort ............................................................................3

1.2.3 Improved Floorplanning .......................................................................5

1.2.4 Functionality Enhancements ...............................................................6

1.3 GENERAL RESEARCH OBJECTIVES .......................................................6

1.4 EXPECTED CONTRIBUTION TO THE RESEARCH ...............................8

1.5 OVERVIEW OF THE THESIS .................................................................8

2. BACKGROUND AND RELATED RESEARCH ...........................................9

2.1 OVERVIEW OF AN ISLAND STYLE MT-FPGA ......................................9

2.2 DESCRIPTION OF MULTI-TECHNOLOGY LOGIC CLUSTER .............10

2.2.1 Programmable SRAM Cell .................................................................10

2.2.2 4-Bit Look-Up Table ........................................................................11

2.2.3 8X1 Multiplexer ................................................................................12

2.2.4 User Flip-flop .....................................................................................12

2.2.5 Programmable Logic Block (PLB) .....................................................13
2.2.6 Multi-Technology Block (MTB) ................................................................. 14
2.2.7 Connection Block (CB) ................................................................. 14
2.2.8 MTLC Floorplanning ................................................................. 15
2.3 RELATED FPGA RESEARCH ................................................................. 16
  2.3.1 Optimum Design Parameters for Cluster-Based Logic Blocks in FPGAs ...... 17
  2.3.2 Speed and Density Improvement Using Hard-Wired Logic Blocks in FPGAs…… 17
  2.3.3 Optimum Connection Block Flexibility ........................................ 18
2.4 SUMMARY ........................................................................................ 19
3. ANALYSIS OF FIRST GENERATION MTLC ........................................ 21
  3.1 FUNCTIONALITY AND FLEXIBILITY .............................................. 21
      3.1.1 Non-Optimum Functionality in Arithmetic Operations .................... 21
      3.1.2 Integration of a Dedicated Carry-Chain in the PLB for Arithmetic Operations .... 22
      3.1.3 Limited Flexibility Within the MTLC for Arithmetic Operations .......... 22
      3.1.4 Incrementing Multiplexer Sizes in the MTLC for More Flexibility ......... 23
  3.2 CONNECTION BLOCK DESIGN ....................................................... 24
      3.2.1 Circuit Design Issues .......................................................... 25
      3.2.2 Proposed Circuit Design ...................................................... 25
  3.3 SPEED AND POWER ................................................................. 26
      3.3.1 Speed and Power Dissipation Issues in Arithmetic Operations ............ 26
      3.3.2 Measures to Improve Speed and Reduce Power Dissipation ............... 27
  3.4 AREA AND FLOORPLANNING .................................................... 27
      3.4.1 Area and Floorplanning Issues .............................................. 27
      3.4.2 Stacked PLB Based Floorplanning Solution .................................. 28
  3.5 CAD IMPLEMENTATION ........................................................... 29
3.5.1 Programming Challenges ................................................................. 29
3.5.2 Solution Through New Floorplanning ............................................. 31

4. SECOND GENERATION MTLIC DESIGN EVALUATION ........... 33

4.1 FUNCTIONAL VERIFICATION ......................................................... 36
4.1.1 MTLIC Programming ............................................................... 37
4.1.2 Scan-Chain Verification ............................................................ 38

4.2 TIMING VERIFICATION ............................................................... 40
4.2.1 Typical and Worst Case Logical Operation Delay ....................... 40
4.2.2 Worst Case Addition Operations .............................................. 41
4.2.3 Worst Case Multiplication Operations ..................................... 42
4.2.4 Firm-Link Vs Soft-Link Timing Delays ..................................... 43

4.3 POWER DISSIPATION ESTIMATION .......................................... 46

4.4 AREA COMPUTATION ................................................................. 46

4.5 PROGRAMMING EFFICIENCY COMPUTATION .......................... 47

5. CONCLUSIONS ................................................................. 48

5.1 COMPARATIVE RESULTS OF THE TWO MTLIC ....................... 48

5.2 SUMMARY OF THE THESIS ....................................................... 49

5.3 SUGGESTIONS FOR FUTURE WORK ........................................ 50

BIBLIOGRAPHY ................................................................. 52

APPENDIX A .................................................................. 53

APPENDIX B .................................................................. 57

APPENDIX C .................................................................. 70
LIST OF FIGURES

1. INTRODUCTION

2. BACKGROUND AND RELATED RESEARCH

2.1: Island Style MT-FPGA Generic Block ................................................................. 9
2.2: SRAM Cell ........................................................................................................ 10
2.3: 4-LUT using tree structure .............................................................................. 11
2.4: Tri-state Buffer Based 8X1 Multiplexer .......................................................... 12
2.5: User Flip-Flop .................................................................................................. 13
2.6: Programmable Logic Block ............................................................................ 14
2.7: Connection Block Description ....................................................................... 15
2.8: MTLC Floorplanning ...................................................................................... 16
2.9: Structure of a BLE and Logic Cluster .............................................................. 17
2.10: a) Network of basic blocks ........................................................................... 18
2.10: b) Hard-wired Connections and Logic Blocks .............................................. 18
2.10: c) Faster HLBS .............................................................................................. 18
2.11: L3-4.2 Tree of PLBs .................................................................................... 19
2.12: Definition of flexibility of Connection Block .............................................. 19

3. ANALYSIS OF FIRST GENERATION MTLC

3.1: New PLB Schematic with Carry-Chain .......................................................... 22
3.2: 10-to-1 Multiplexer ....................................................................................... 23
3.3: 1-to-4 De-multiplexer with Tri-State .............................................................. 25
3.4: New Connection Block Details ...................................................................... 26
3.5: Detailed Carry-chain Logic Circuitry .............................................................. 27
3.6: Detailed Floorplanning Solution ..................................................................... 28
3.7: Showing Empty SRAM Bits in Old Floorplanning ....................................... 30
3.8: Less Wasted Bits in New Floorplanning ........................................................ 31

4. SECOND GENERATION MTLC DESIGN EVALUATION

4.1: Top-Level 2G-MTLC Schematic ................................................................. 33
4.2: Full-Custom 2G-MTLC Layout .................................................................... 34
4.3: Photograph of Test Setup for 2G-MTLC Chip ............................................ 35
4.4: 2G-MTLC Programming Scheme ............................................................... 36
4.5: MTLC Programming Sequence & Shift-Register Functional Verification .... 37
4.6: Post-Silicon Test Result of 7-bit Shift-Register ........................................... 38
4.7: 4-Bit Internal MTLC Scan-Chain ............................................................... 39
4.8: Post-Silicon Test Result For 4-Bit Internal MTLC Scan-Chain .................... 39
4.9: Post-Silicon Timing For a PLB as an Inverter ............................................. 40
4.10: Worst Case Addition Operation ................................................................. 41
4.11: Multiplication in 2G-MTLC .................................................................42
4.12: Worst Case Multiplication Operation .....................................................42
4.13: Ring Oscillator With Two Firm-Links and One Soft-Link .......................43
4.14: Post-Silicon Ring Oscillator Output With 2 Firm-Links & 1 Soft-Link .........43
4.15: Ring Oscillator With One Firm-Link and Two Soft-Links ..........................44
4.16: Post-Silicon Ring Oscillator Output With 1 Firm-Link & 2 Soft-Links ..........44
4.17: Ring Oscillator With All Soft-Links ......................................................45
4.18: Post-Silicon Ring Oscillator Output With All Soft-Links .........................45
4.19: Area Reduction Through Better Floorplanning Only ...............................46

5. CONCLUSIONS

5.1: 2G-MTFPGA Layout ..............................................................................50
LIST OF TABLES

1. INTRODUCTION

2. BACKGROUND AND RELATED RESEARCH

3. ANALYSIS OF FIRST GENERATION MTLC
   3.1: New PLB Inputs .................................................................24

4. SECOND GENERATION MTLC DESIGN EVALUATION
   4.1: Summary of 2G-MTLC Timing, Area And Power Data .........................47

5. CONCLUSIONS
   5.1: Comparative Results Between 1G-MTLC And 2G-MTLC ........................47
GLOSSARY OF TECHNICAL TERMS

ASIC
Application specific integrated circuit

CMOS
Complementary Metal Oxide Semiconductor. Technology used to manufacture silicon integrated circuits

Delay flip-flop or D-flop
The input is copied to the output delayed by one clock cycle

FPGA
Field programmable gate array

Flip-flop
This element has two stable states, which are toggled on different events, depending on the type

IC
Generic term. Integrated Circuit. A set of gates etched on a silicon wafer. An IC in a package is commonly referred to as a Chip. Chips are also called ICs

MOSFET
Metal Oxide on Silicon Field Effect Transistor. A chip technology where the transistors are implemented as Field Effect Transistors

MOS transistor
Metal oxide semiconductor transistor

NMOS
N-channel MOS

PMOS
P-Channel MOS

SRAM
Static random access memory. As opposed to DRAM, this type of memory doesn't need a continuous refresh, as the information in it is stored by flip-flops

VLSI circuit
Very large scale integrated circuit

MT-FPGA
Multi-Technology Field Programmable Gate Array

MTLC
Multi-Technology Logic Cluster

PLB
Programmable Logic Block

MTB
Multi-Technology Block

CB
Connection Block

SB
Switch Block
CHAPTER 1

1. INTRODUCTION

1.1 MOTIVATION AND RESEARCH

Programmable logic devices have been around in the industry for two decades in various flavors including PLA, CPLD and FPGA. While these devices have seen tremendous technical and commercial success, they all cater to purely digital applications. Unfortunately, there were no re-programmable device alternatives for mixed-signal and multi-technology applications that are in greater use today. That led to some prior work in the novel field of Multi-Technology FPGA design [1] which was a wonderful proof-of-concept initiative and was flexible enough to port many applications. Now, since these applications have become demanding in terms of area, speed, power and flexibility, MT-FPGA designs should also be upgraded to provide improved performance. The motivation behind this research is to find and implement those techniques that will make the existing MT-FPGAs more capable of delivering results in terms of input-to-output delays (speed metric), number of multi-technology logic blocks used (area metric) and average power dissipated (power metric) in order to realize a wide variety of circuits (flexibility metric). The next section describes the general design optimization tools a designer has at his disposal.

1.2 GENERAL DESIGN OPTIMIZATION TECHNIQUES EXPLORED

The circuit topologies of various components of the existing MTLC design [1] were found to be appropriate for the current research and were retained. In other words, the research
initiative of this thesis builds upon the groundbreaking work previously done at the University of Cincinnati. But in order to optimize the design further to reach the thesis objectives, various other alternatives were explored. These design paradigms pertain to optimizations at various levels of design and implementation, be it transistor and gate level in the design side, to floorplanning, placement and routing in the implementation side. These are as follows.

1.2.1 TECHNOLOGY SCALING

Migrating to a better CMOS technology of a lower feature size and supply voltage provides lots of advantages in terms of area, speed and power dissipation with respect to the older CMOS technologies. Mostly, a General Scaling Model is followed where the supply voltage scaling is not proportional to the device scaling as there are intrinsic device voltages such as the silicon band-gap and built-in junction voltages which cannot be scaled [2]. Moreover, the threshold voltage has a limited scaling capacity, so very low supply voltages might lead to devices which are not completely off leading to the dissipation of short-circuit power. In general, with scaling, area per device is reduced due to the decreased dimensions, which generally leads to lower overall circuit size. Typically, the speed of circuits is increased as the intrinsic transistor speed increases due to lower internal capacitances. In addition, due to lower area the local interconnect capacitances and resistances are decreased leading to further reduction in delays. Finally, power dissipation per device is also reduced mainly due to the decrease in the supply voltage and internal/external capacitances (dynamic power dissipation).

However, designing in a smaller technology comes with its own challenges like sub-threshold current leakage which increases the static component of the overall power dissipation.
Moreover, as the die area increases the global interconnect delays start to become the dominant delay component as the transistors become faster. Also, the power density (power dissipated per unit area) increases putting a strain on the die material and actually leading to higher power dissipation per chip as the chip area is also increasing.

Even if the floorplanning of the macro components is kept the same, a good amount of area savings can be made by just lowering the feature size. So, technology migration is one of the better ways of packing more Logic Blocks, Connection Blocks and Switching Blocks in the same die area, thereby increasing the MT-FPGA flexibility to port more diverse applications.

### 1.2.2 USE OF LOGICAL EFFORT

In complex logic gate networks, transistor sizes determine the speed, energy consumption, circuit area and satisfaction of delay constraints. So, design methodologies in choosing the optimum gate sizes in the presence of trade-offs between speed and energy are critical to the success of the design. Logical Effort [3] is one such methodology that builds upon the sizing and delay analysis of an inverter chain and generalizes it for any type of logic gate path in a network. Delay of a gate is thus given by -

\[
    t_p = t_{p0} (p + g*f)
\]

where,

- \(t_{p0}\) = intrinsic delay of a minimum sized inverter,
- \(p\) (parasitic delay) = ratio of intrinsic delays of the gate to a minimum sized inverter,
- \(g\) (logical effort) = ratio of input capacitance of the gate to the input capacitance of a minimum sized inverter delivering the same output current,
- \(f\) (electrical effort) = ratio of external load to input capacitance of the gate.
Logical effort represents the fact that for a given load, complex gates have to work ‘harder’ than inverter to produce a similar response. It is a function of topology and independent of the sizing, that is, Logical Effort increase with gate complexity. Values for $p$ and $g$ for some common gates are:

- $p = g = 1$ for an INV,
- $p = 2$ and $g = 4/3$ for 2 input NAND,
- $p = 2$ and $g = 5/3$ for 2 input NOR

So, the total delay of a path through a combinational logic block can be expressed as:

$$
t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0}^* \sum_{j=1}^{N} (p_j + f_j * g_j)
$$

where, $f_j g_j$ is also known as the Gate Effort of a complex gate.

By finding $N$-1 partial derivatives and setting them to zero, it is found that ‘each stage should bear the same gate effort’:

$$
f_1 g_1 = f_2 g_2 = \ldots = f_N g_N
$$

The logical effort along a path compounds by multiplying the logical efforts of all the gates in the path, yielding the Path Logical Effort $G$:

$$
G = \prod_{j=1}^{N} g_j
$$

Path Electrical Effort $F$, which relates the load capacitance of the last stage to the input capacitance of the first stage, is given by:

$$
F = C_L / C_{g1} = \prod_{i=1}^{N} f_i / b_i = (\prod_{i=1}^{N} f_i) / B
$$

The Path Branching Effort, which measures the total load overhead necessary to drive all the given nets in the path is given by:
\[ B = \prod_{i=1}^{N} b_i \]

Where \( b = (C_{\text{on-path}} + C_{\text{off-path}})/C_{\text{on-path}} \)

Finally, the Total Path Effort is given by:

\[ H = \prod g_{f_i} = GFB \]

The gate effort that minimizes the path delay is

\[ h = \sqrt[3]{H} \]

and the minimum delay through the path is

\[ D = t_{p0} \left( \sum_{j=1}^{N} p_j + N \sqrt[3]{H} \right) \]

A detailed treatment to this simple but elegant approach can be found in [3]. Thus the Logical Effort metrics offer a straightforward means of determining the fastest and optimally CMOS implementation of a logic function. Careful logic gate sizing can help in reducing the overall MTLC implementation area and avoid excessive loading of one logic stage by the following logic stage. This will lead to lesser layout area, better speed and lower power.

**1.2.3 IMPROVED FLOORPLANNING**

The way in which designed circuits are implemented on the silicon has a great impact on the performance of the final fabricated design [4]. As discussed above, when designing in lower technologies, the interconnect delays are the major reason for slowdown. Thus, measures should be taken to carefully floorplan the macro-blocks so that the interconnection routing is minimal. In addition, good floorplanning helps to tightly pack the logic together leading to smaller area. Smaller area is the key to noise immunity and usually leads to smaller capacitances and resistances and thereby reduces power dissipation. Moreover, presence of
more number of metal layers in a CMOS process helps in reducing the effective area taken for routing by overlapping their layout.

Usually, the Basic Logic Blocks in FPGAs are made using the ‘full-custom’ transistor level design methodology to fully utilize the drawn space. Optimizing the MTLC floorplanning can be critical to the performance of an MT-FPGA and thus calls for new arrangement of PLBs, the MTB and the Connection Blocks.

1.2.4 FUNCTIONALITY ENHANCEMENTS

In addition to speed, area and power optimizations, functionality enhancements in terms of efficiently performing different operations other than the basic logic calculations are also central to the overall design upgrade. Though groups of Basic Logic Blocks (BLE) are collectively capable of performing many complex operations like addition, multiplication, division etc on vector inputs, these operations would be very expensive in terms of number of logic blocks used, operation speed achieved and power dissipated. So it makes good design sense to allocate some area for dedicated circuitry that is common to these operations like carry-chain logic for arithmetic operations. Many commercial FPGAs like the ones form Xilinx [5], Altera etc incorporate these arithmetic carry-chains to reduce the overall logic count and realizing wide-input logic among other benefits. Other enhancements that have been made in cluster-based commercial FPGAs are the addition of dedicated multipliers and on-chip memory for DSP applications [5].

1.3 GENERAL RESEARCH OBJECTIVE

Though a previous MTLC design exists, this is the first time an approach has been considered to optimize the MTLC design on all three design metrics- Speed, Area and
Power. In addition, dedicated data processing capability has been provided in the form of carry-chain logic circuitry to efficiently perform arithmetic operations on external (from Routing Channels) as well as internal (from the MTB and PLBs) data. So all of the design alternatives suggested in the previous section in addition to other critical circuit design changes are incorporated for implementing the second generation MTLC.

In this thesis we investigate the design constraints enforced by latest high-speed, low-power multi-technology applications requiring large number of MTLCs and higher Routing Channel (RC) capacity for their execution. In addition, the CAD tools designed to generate the bit-stream to populate the SRAMs impose extra floorplanning and placement constraints. The technical objectives for this research are summarized below:

- To design a multi-technology logic cluster that is smaller in area as compared to the previous design and works faster with equal or better Routing Channel load driving capability.
- The average power consumption of the MTLC at its highest operating speed should be smaller than the previous design.
- In addition to logic operations, the MTLC should be able to perform a variety of arithmetic operations with a better utilization of its resources.
- The MTLC flexibility should be more in terms of external and internal connectivity so that the final MT-FPGA is more flexible.
- The Connection Blocks in the MTLC should have the proper tri-state behavior so that there is no input/output from/to the Routing Channels when not required.
- The physical layout and placement of the SRAMs in the MTLC should facilitate the implementation of indigenously developed CAD for programming bit-stream
• Undertake a quantitative performance comparison between the previous and the new
geneneration MTLC to support the new design.
• Suggest future improvements to MTLC design.

1.4 EXPECTED CONTRIBUTIONS TO THE RESEARCH

The research is expected to make a significant contribution in the development of MT-FPGA devices for high-performance applications. By integrating these second generation MTLCs with other improved components like the Switch Blocks and the I/O Blocks, high capacity MT-FPGAs operating at higher clock speeds with easier programmability can be built that can work on a wide range of applications related to mixed-signal, opto-electronics, MEMS etc. The pre- and post-silicon timing data provided by the thesis will help in closely predicting the performance of an MT-FPGA of any size for the current CMOS processing technology.

1.5 OVERVIEW OF THE THESIS

The thesis has been organized into five chapters. The following chapter provides the overview of the previous research in the field of MT-FPGAs and other related topics that influence the current research. In Chapter 3, an attempt has been made to analyze the first generation MTLC in detail. Various aspects of circuit design, be it circuit style, floorplanning, sizing, functionality, flexibility etc are studied to find pertinent issues and suggest feasible solutions. Chapter 4 pertains to the evaluation of the proposed MTLC design for functionality, flexibility, timing, size and power at the schematic, layout and post-silicon level. Finally, Chapter 5 gives the comparative results of the two generations of MTLC design and suggests future work.
CHAPTER 2

2. BACKGROUND AND RELATED RESEARCH

2.1 OVERVIEW OF AN ISLAND STYLE MT-FPGA

Figure-2.1 shows a generic block of the island style MT-FPGA topology as proposed in the previous research [1]. In this style of FPGA, multi-technology logic clusters (MTB and PLB) are arranged in a regular array and each MTLC is surrounded by a group of horizontal and vertical connection wires collectively known as Routing Channels (RC). These routing channels can be segmented and thus can form a hierarchy of different length classes. These segments are connected in a programmable fashion through the use of Switch Blocks (SB). The routing channels have a) digital signal lines to connect the PLB outputs and digitized MTB outputs of an MTLC to PLB inputs of other MTLCs and b) electrically isolated signal lines to connect raw MTB outputs from one MTLC to MTB inputs of other MTLCs. The input and output pins of an MTLC can be connected to the routing channels in a configurable way using Connection Blocks (CB). In this way, the islands of multi-technology logic clusters can communicate with each other in a manner that can be determined by the programming of the SBs and CBs. Moreover, the signals from/to the chip input/output pads can be connected to the routing channels through programmable I/O...
Blocks. Effectiveness of the MT-FPGA architecture depends on many parameters like good MTLC design, the interconnection flexibility provided by the Switch Blocks and Connection Blocks, I/O Block flexibility for efficient external interface of the design, routing channel width and segmentation scheme used, among others. A detailed analysis of these parameters and their effect on the architecture efficiency is provided in [6]. Since good MTLC design is the objective of this research, following section provides a description of the first MTLC design proposed as a significant part of the novel MT-FPGA architecture [1].

2.2 DESCRIPTION OF MULTI TECHNOLOGY LOGIC CLUSTER

At the heart of the MT-FPGA design is the multi-technology logic cluster which provides digital signal logic operation capabilities along with specialized operations on a variety of signal types through the integration of multi-technology modules into the design. The following subsections describe each of the components that constitute an MTLC.

2.2.1 PROGRAMMABLE SRAM CELL

SRAM cells are the basic storage units used to store the configuration bits for any FPGA device including our MT-FPGA. These are volatile memory circuits which can be re-programmed any number of times depending on the changes in functionality and interconnection requirements of the target applications. Figure-2.2 shows the gate level description of the circuit. When the Sel signal is ‘High’, any data can be put into the cell through the In signal. But when the
$Sel$ is ‘Low’, the cell latches the data and a change in the input cannot change the output. Back to back inverters 1 and 2 form the latch in which the inverter 2 is called a ‘trickle inverter’ and is made much weaker than inverter 1 to allow new input data to override any previous data of the opposite logic level when $Sel$ is ‘High’.

### 2.2.2 4-BIT LOOK-UP TABLE

A look-up table (LUT) is the fundamental component of any programmable logic including an MT-FPGA. A four input LUT can compute one of $2^6 = 65536$ logic functions depending on the programming bits. Circuit design and implementation of an LUT is critical to the success of an MT-FPGA in terms of speed, power and area metrics. Figure-2.3 shows the circuit design chosen for the first generation MT-FPGA [1].

One of the main reasons for selecting this topology is its smaller area and higher speed as compared to many other options like clocked LUT, NMOS tree LUT, NMOS stack LUT etc as described in [1]. This design merges NMOS stacks in a tree like fashion that allows lower NMOS nodes to share the higher nodes leading to a smaller area. Also, the SRAM bits are used as gate inputs rather than the source inputs avoiding any signal degradation, which is the case in conventional NMOS tree LUT. This design in some ways is a part of the Ratioed
Logic CMOS gate family [2] and thus has to conform to its design paradigms. The PMOS load has to be carefully sized in such a way that the $t_{PHL}$ is low, which requires it to have a strong drive, while to have a low $t_{PLH}$ and proper functioning it should be weak with respect to the equivalent NMOS pull-down size. One shortcoming of this design is the short-circuit static power dissipation which results when both PMOS load and NMOS pull-down are ‘ON’. But this trade-off had to be made as the layout area was limited.

2.2.3 8X1 MULTIPLEXER

As a part of the routing architecture, the multiplexers are used to provide connection flexibility within the MTLC. Figure-2.4 shows a tri-state buffer based 8-to-1 multiplexer whose output can be connected to one of the inputs of the 4bit-LUT. In the MTLC design, the input to these multiplexers can come from the routing channels, the outputs of the PLBs and digitized outputs from the MTBs. So by configuring these multiplexers, logical operations can be performed on different combinations of these signals. Though tri-state buffer based multiplexers take more area, they provide signal regeneration through a low impedance path with either power supply or ground. This is good for the signals coming from the routing channels after significant delay and loss of drive.

2.2.4 USER FLIP-FLOP
Figure-2.5 illustrates the composition of a user flip-flop. The user flip-flop provides the storage element for the sequential logic implementation in a PLB. The output of the previous operation can be stored in this flip-flop and fed back through the multiplexers to the four input look-up table. Moreover, these can form a part of a scan-chain which can be used to check the correct functionality of the LUT during the test mode. A reset signal is also provided to clear the contents of the flip-flop.

2.2.5 PROGRAMMABLE LOGIC BLOCK

A Programmable Logic Block (PLB) is composed of the above mentioned components in a manner as shown in figure-2.6. Diversity of logic functions that can be implemented is provided by the LUT while the flexibility of connection is given by the various multiplexers. The truth table of the function to be implemented is stored in the SRAMs of the LUT and is accessed using the four bit input combination coming from the
SRAM programmable multiplexers. The corresponding output bit can be accessed either directly or through a user flip-flop using a SRAM configurable output-mux depending on the combinational or sequential logic requirement. Test signal is used to implement a scan-chain using the user flip-flops and also adds the controllability and observability required by using the scan-mux. CLKA and CLKB are the non-overlapping clock sources used in the user flip-flop. Implementation style of the PLB is the key to smaller area of the MTLC and in turn leads to a smaller overall layout. It also specifies the physical arrangement and interconnection of the SRAMs which is critical in determining the ease of programming and designing CAD tools and the time taken to program all the SRAMs before the MT-FPGA can be used.

2.2.6 MULTI-TECHNOLOGY BLOCK

It is the seamless integration of a multi-technology block into a cluster of programmable logic blocks that differentiates an MT-FPGA from a conventional FPGA. These multi-technology blocks can be anything from analog blocks like sensors, opto-electronic blocks like photo receivers/detectors, mixed-signal blocks like analog-to-digital converters to MEMS and Radio Frequency blocks, just to mention a few. Only digitized MTB signals can be used by the PLBs for logic operations. Careful design and layout techniques are required to put digital circuits and these mixed technology blocks in close proximity. These circuits and their respective design issues are outside the scope of this research and are covered in-depth in [1, 9].

2.2.7 CONNECTION BLOCK
Apart from the Switch Blocks and PLB multiplexers, Connection Blocks are the final component of the routing architecture. They are responsible for the interconnection between the MTLC inputs/outputs and the routing channel wires. Connection block description is given in figure-2.7. It shows a group of four transmission gate based multiplexers and de-multiplexers for each direction. Transmission gates are used to save area at the cost of speed and simplicity. Connection block flexibility, \( F_C \), is defined as the number of routing channel wires to which an MTLC signal can be connected and was chosen to be 4 for digital and 2 for analog signals.

**2.2.8 MTLC FLOORPLANNING**

Figure-2.8 illustrates detailed floorplanning with combined signal interconnection to give a consolidated view of the MTLC physical structure. Each MTLC is made up of four PLBs surrounding an MTB. A 16-bit internal bus runs between them in a ring fashion and is used as a communication link among the PLBs, between PLBs and MTB and between the MTLC core and the Connection Block. So all the data to and from the external and internal sources is put on the bus before being distributed. Since both raw and/or processed data can be provided by the MTB, appropriate signal lines have to be provided in the signal bus. The 16-
bit internal bus is subdivided as follows-

1) 4-bits for output signal distribution from the PLBs (1,2,3,4)

2) 4-bits for bi-directional communication with the MTB (5,6,7,8)

3) 4-bits for signals coming from other MTLCs through the routing channels (9, 10, 11, 12)

4) 4-bits for clock distribution, reset and configuration enable (13,14,15,16)

As shown in Figure-2.8, interconnections among the PLBs are a combination of soft and firm links to achieve area and speed advantages [7, 8]. The links shown between the MTB and the PLBs are ‘optional firm links’ and could be created by the user if the MTB requires any direct output from the PLBs.

2.3 RELATED FPGA RESEARCH

Since the MTLC consists of a group of four programmable logic blocks with a single multi-technology block, it closely resembles cluster-based logic block designs in conventional FPGAs. Also, some kind of hierarchy can exist internally among the logic blocks within the MTLC. So to a certain extent, the experimental and analytical results applicable to them can be applied to MTLCs as well. Other related research topics that influenced the present
research are also presented.

2.3.1 OPTIMUM DESIGN PARAMETERS FOR CLUSTER-BASED LOGIC BLOCKS IN FPGAs

Two key architectural parameters for a cluster-based FPGAs are the number of BLEs (Basic Logic Element) in each cluster (N) and the number of distinct inputs required by the cluster of size N. Figure-2.9 shows the structure of an FPGA cluster consisting of N BLEs with I distinct inputs (I<4N) from the routing channels. The N outputs of the BLEs are also shared among themselves. Through technology-mapping, placement and routing of 20 MCNC benchmark circuits on various architectures, values of N and I giving the most area-efficient architecture were found [10]. It was found that a cluster having 4 BLEs leads to most area-efficient architecture compared to a single BLE based one. It was also shown that a cluster of N BLEs needs only I = 2N+2 distinct inputs (vs. the 4N maximum) for complete logic utilization leading to lower interconnect area.

2.3.2 SPEED AND DENSITY IMPROVEMENT USING HARDWIRED LOGIC BLOCKS IN FPGAs

In fine-grained FPGAs, all connections between the basic logic blocks are programmable and hence are inherently slower and larger than hardwired connections. One way to improve
speed and density is through the construction of coarse-grained logic blocks (called Hard-wired Logic Blocks) from several basic logic blocks and containing some hardwired interconnections in addition to soft-connects [7]. Figure-2.10(c) shows an example of the

![Figure-2.10: a) Network of basic blocks b) Hard-wired Connections and Logic Blocks c) Faster HLBs](image)

speed and area gain achieved in the critical path (maximum logic levels) from block 1-5 in Figure-2.10(a) through the use of HLB shown in Figure-2.10(b). Due to the HLB, programmable connections have reduced from 9 to 4 with only two in the critical path. Experimental results on MCNC benchmarks showed that an HLB of Figure-2.11 containing L3-4.2 tree of basic logic blocks (PLBs) leads to optimum speed-area benefits.

2.3.3 OPTIMUM CONNECTION BLOCK FLEXIBILITY

As shown in Figure-2.12, Connection Block Flexibility is measured in the number of programmable connections an input or output of a Logic Block can make with the Routing Channel lines. Implementation cost in terms of area and delay must be kept in mind while

![Figure 2.11: L3-4.2 Tree of PLBs [1]](image)
deciding on the interconnection flexibility alternatives in an FPGA. Though mostly technology dependent, these estimates can be made by knowing the number of switches per tile (repetitive blocks containing the logic block and switching structures on the two sides of the block) in an FPGA. The number of switches per tile is the sum of Switch Block and Connection Block switches and are given by -

\[
\text{Number of Switches in Switch Block} = 2F_S W
\]
\[
\text{Number of Switches in Connection Block} = T^P F_C
\]

Where \( F_S \) is the Switch Block Flexibility, \( W \) is the Routing Channel Width, \( F_C \) is the Connection Block Flexibility, \( T \) is the number of sides where the logic pins are accessible and \( P \) is the total number of logic pins in the logic block.

Experimental results in [11] on benchmark circuits showed that the minimum number of switches occurred when \( 3 \leq F_S \leq 4 \) and \( 0.7 \leq F_C/W \leq 0.9 \). Optimum value of \( F_C \) is significant in determining the area and delay for the MTLC design.

2.4 SUMMARY

This chapter gives the information regarding the amount of research work that has already been put into the development of MT-FPGAs. Various circuit options for each component of the MTLC were evaluated on different design parameters before selecting the best one that helped in realizing the design goals. Accordingly, floorplanning followed by placement
and routing were performed to construct the whole MT-FPGA chip as a proof-of-concept.

In order to have a still deeper understanding of each component of an MT-FPGA, the reader must go through the literature presented in [1]. In addition, other research initiatives that have a direct impact on the present research have been outlined with their references. Detailed study of the first generation MT-FPGA and other related research material gives ample preparation to analyze the MTLC design, find pertinent design and implementation issues and suggest changes that form the basis of developing a second generation MTLC structure.
CHAPTER 3

3. ANALYSIS OF FIRST GENERATION MTLC

This chapter is about an in-depth analysis of the First Generation Multi-Technology Logic Cluster (1G-MTLC). Various aspects of its design are covered seeking various ways to optimize the circuits and layout in order to have better speed, area and power.

3.1 FUNCTIONALITY AND FLEXIBILITY

Functionality of an MTLC pertains to the efficiency with which the MTLC can be used for different types of operations like logical, arithmetic etc. Flexibility is the measure of ease with which an MTLC can be connected to maximum number of different external RC lines through programmable switches while satisfying its area budget. This section analyzes the 1G-MTLC and suggestions are made to resolve any issues encountered through this analysis.

3.1.1 NON-OPTIMUM FUNCTIONALITY IN ARITHMETIC OPERATIONS

In the First-Generation Multi-Technology Logic Cluster (1G-MTLC), the four PLBs are good enough to realize complex logic functions needed in most applications. But for arithmetic operations, each bit calculation will require two PLBs – one for sum and one for carry generation. This is a big price to pay as the PLBs are not fully utilized in terms of inputs used (two out of four are used) and the logic function realized inside them (XOR/AND). This problem becomes severe if lots of wide-input data calculations are being performed as they can consume most of the PLBs, leaving inadequate number of PLBs for logic calculations.
3.1.2 INTEGRATION OF A DEDICATED CARRY-CHAIN IN THE PLB FOR ARITHMETIC OPERATIONS

The above problem of spending more than necessary PLBs in the arithmetic operations can be alleviated using dedicated carry-chains which pass the carry-out signal based on the generate, propagate and delete carry information generated using the two 1-bit inputs for both addition and subtraction. Figure-3.1 shows the suggested PLB schematic with dedicated carry-chain logic which is internally very similar to the one used in Vertex™ family of FPGAs by Xilinx© Corporation [5]. Multiplication also uses the same logic, besides generating a partial product term (using logical AND), as it is an addition and shift operation in essence. Section 3.3.2 shows the detailed description of the proposed carry-chain logic. The additional layout space dedicated to it increases the area per PLB but greatly reduces the area required to implement arithmetic operations in terms of number of PLBs, which is a good area trade-off.

3.1.3 LIMITED FLEXIBILITY WITHIN THE MTLC FOR ARITHMETIC OPERATIONS

After the carry-chain has been added, a cluster of four PLBs can perform arithmetic operations like addition and subtraction on two 3-bit numbers resulting in a 4-bit output, while multiplication can be performed on two 2-bit numbers leading to a 4-bit output. But
the circuit resources in the 1G-MTLC are inadequate (only four) to support that number of independent input bits in the case when all inputs to the MTLC come from the routing channel, which is a very common case. So there is a limited flexibility in the number of routing channel input that can be fed to the MTLC. To circumvent this problem, eight instead of four inputs should be able to connect with the PLBs inside the MTLC through multiplexers feeding the four-input Look-Up Table.

**3.1.4 INCREMENTING MULTIPLEXER SIZES IN THE MTLC FOR MORE FLEXIBILITY**

![Figure-3.2: 10-to-1 Multiplexer](image)

To increase the flexibility of the MTLC in taking the inputs from the routing channels, the multiplexers providing soft-connects to the 4-LUT have to be increased in size. The 1G-MTLC has an 8x1 multiplexer size and only one RC signal from each side can be connected to the LUT input leading to just four independent RC inputs. To take this number to eight the multiplexer size has to be increased to 10x1 with four select signals and discarding select signals which correspond to the input signals beyond the count of ten, as shown in figure-3.2. This will lead to a minor increase in PLB area but also increase the much needed flexibility. Due to this size change, the inputs that can be connected to the 4-bit LUT of the PLBs in the MTLC have been revised and presented in table-3.1.
Table 3.1: New PLB Inputs

As the table 3.1 shows, PLB1 and PLB2 have identical interconnection matrix, while PLB3 and PLB4 are different because of their hard-wired connections conforming to the L3-4.2 tree structure.

### 3.2 CONNECTION BLOCK DESIGN

The Connection Block consists of multiplexers and de-multiplexers for digital and analog signals with inputs or outputs connected to the Routing Channels depending on the direction of signal flow.
3.2.1 CIRCUIT DESIGN ISSUES

In the 1G-MTLC, since a RC is shared by two opposite MTLCs in each direction, the multiplexers/de-multiplexers may take the inputs or feed the outputs to the same RC lines. This leads to two functional problems. First, two different logic outputs to the same RC line can occur leading to an indeterminate logic state of the output. Second, since the two digital MTB signals can be configured as either inputs or outputs, they might connect to the same RC line and again lead to an indeterminate logic state. These situations are likely to result in incorrect functioning of the MTLC if not taken care of through circuit design change.

3.2.2 PROPOSED CIRCUIT DESIGN

The signal integrity issue discussed above can be taken care of by adding tri-state buffers at the input and outputs of the Connection Block multiplexers/de-multiplexers on all sides of the MTLC. That way, the MTLC signals connecting to the RC lines can be tri-stated using appropriate programming bits as and when required. This will avoid the indeterminate logic state as only one signal at a time will be feeding the line with full strength ‘1’ or ‘0’ logic through Vdd and Gnd power supplies respectively, while other signals to the same line will be in high impedance state. The new circuit diagram for a 1x4 de-multiplexer is shown in figure-3.3.

Figure-3.3: 1-to-4 De-multiplexer with Tri-State
In order to further boost the routability of applications to be ported onto the MT-FPGA, it was decided to increase the routing channel width from six in the earlier MT-FPGA to ten in the new MT-FPGA to be designed by using the 2G-MTLC. In addition, to tap two inputs from the Routing Channels, one extra 4x1 multiplexer was added in each direction. Based on these reasons, the Connection Block design has been expanded from its original configuration [1] to the new connectivity schematic of figure-3.4.

3.3 SPEED AND POWER

Speed and power are an integral part of design performance. This section delves into the analysis of these two metrics for the 1G-MTLC and suggests measures to improve them.

3.3.1 SPEED AND POWER DISSIPATION ISSUES IN ARITHMETIC OPERATIONS

The speed of the arithmetic operations like addition, subtraction and multiplication will be particularly low in the 1G-MTLC as the ripple carry-out is generated by a PLB whose delay is quite large due to the presence of more than the minimum required logic (input multiplexers and 4-LUT logic) between the carry-in and carry-out. This delay becomes worse for wide input data as the carry ripple delay is proportional to the number of input bits.
power consumption is also increased as more than the required logic is switching during the arithmetic operations.

### 3.3.2 MEASURES TO IMPROVE SPEED AND REDUCE POWER DISSIPATION

Both the speed and power dissipation can be improved by using dedicated carry-chain logic for arithmetic operations [5] and by optimum logic gate sizing in the critical data path. First, the detailed schematic of carry-chain logic is shown in Figure-3.5. This avoids the use of a PLB for ripple carry-out calculation and thereby decreasing delay since the dedicated carry logic has far fewer logic gates between the carry-in and carry-out signals. Also, since a small number of logic gates are undergoing transition, power is saved. Optimum gate sizing for the critical data path helps in minimizing the overall effective capacitance leading to lower delays (delay $\sim R \times C_{\text{effective}}$) and dynamic power ($P_{\text{dynamic}} = C_{\text{effective}} \times V_{\text{dd}}^2 \times f_{\text{clk}}$).

### 3.4 AREA AND FLOORPLANNING

This section analyzes the area requirement of the first generation MTLC and gives suggestions to improve it through floorplanning techniques.

#### 3.4.1 AREA AND FLOORPLANNING ISSUES

The detailed 1G-MTLC floorplanning diagram of Figure-2.8 shows that the Multi-
Technology Block (MTB) is surrounded by a 16-bit signal bus with one PLB and one CB on each side. The inputs and outputs for the MTB, PLB and CB are tapped from the signal bus. This floorplanning scheme, though very logical, leads to more area than actually required. One reason is that the ring style of the signal bus adds twice the bus width to the length and width of the overall layout. Secondly, the placement of a PLB-CB combination on each side of the MTB prevents the layout from taking the smaller area advantage of the L3-4.2 tree topology [7] which requires close proximity of PLBs to pack more logic in less silicon real estate. The area constraint due to fixed die size mandates the minimization of MTLC area through better floorplanning.

**3.4.2 STACKED PLB BASED FLOORPLANNING SOLUTION**

In order to utilize the area benefits of the L3-4.2 tree structure, the PLBs can be stacked with the MTB to the side of the stack. Figure-3.6 shows the detailed floorplanning of the proposed solution. The close proximity of the PLBs facilitates the utilization of any white space through placement of extra logic circuitry like the dedicated carry-chain, user flip-flops etc. As shown in the figure-3.6, even within the PLB, the layout of the 10x1 multiplexers around the 4-LUT has been done in such a way that it allows the

*Figure-3.6: Detailed Floorplanning Solution*
incorporation of the carry-chain logic on one side and user flip-flop on the other. In addition, the Connection Blocks are also placed in a way that helps in packing them with the PLBs and the MTB with minimum area overhead. Moreover, the signal bus width now contributes only once to the overall area instead of twice as in the 1G-MTLC. The full-custom design methodology greatly helps in achieving minimum area for the MTLC.

3.5 CAD IMPLEMENTATION

As suggested earlier, SRAMs are the storage elements that are used to pre-program the MTLC according to a particular logical or arithmetic computation for which it is going to be used later. An SRAM can be used for either storing the output of a function (in the Look-Up Table) or selecting a particular input as the output (in multiplexers/de-multiplexers). So, before using the MTLC, a programming sequence is undertaken to store strings of logical ‘0’ or ‘1’ (called the bit-stream) in the SRAMs according to their interpretation by the architecture. The Computer Aided Design (CAD) software required to generically program the MTLC (and thereby the MT-FPGA) for any particular application was also developed in parallel by a separate team of students at the University of Cincinnati [12, 13]. During its development some challenging issues came up which reduced the ease with which a bit-stream could be generated to program the 1G-MTLC and the whole MT-FPGA chip. This section underlines those issues and suggests remedial approach.

3.5.1 PROGRAMMING CHALLENGE

The Xbits object oriented framework approach [12, 13], developed at the University of Cincinnati takes a placed and routed circuit description for a particular application and generates a bit-stream according to the MT-FPGA architecture as described in a custom Architecture Description File Format (ADFF). An ADFF essentially contains the x and y co-
ordinates of the SRAMs for each programmable sub-component of an MTLC. This format allows the CAD designer to specify various MT-FPGA architectures which can be integrated into the rest of the design framework. One such ADFF was generated for the 1G-MTLC based on the physical locations of its SRAMs. These co-ordinates have to be provided by the layout designer to the CAD designer. As shown in Figure-3.7, the floorplanning of the 1G-MTLC leads to two main issues, which are as follows:

- The circular arrangement of the SRAMs leads to a lot of empty and wasted programming bits in the center and corners of the MTLC as there is no SRAM at the cross-point of Enable and Data lines in those areas. This reduces the programming efficiency of the generated bit-stream and therefore it takes more time than needed to program the MTLC.

- The circular arrangement of the SRAMs also generates a larger SRAM position database in the ADFF as there is not much symmetry in the positioning of the SRAMs. Therefore, the x and y co-ordinates of all the SRAMs have to be stored in the MTLC description.

The above two issues become all the more important in large size MT-FPGAs. In those cases the large amount of wasted bits and programming database limit programming efficiency.

![Figure-3.7: Showing Empty SRAM Bits](image-url)
3.5.2 SOLUTION THROUGH NEW FLOORPLANNING

The stacked PLB based floorplanning scheme as described before helps in meeting many programming challenges encountered in the 1G-MTLC. The most important reason for this is the regular stack arrangement of the programmable SRAMs in the MTLC. Due to this fact, from the CAD tool point of view, the whole MTLC can be thought of as a grid of programmable switches. As shown in figure-3.8, there are fewer wasted bits in the new floorplan as there is an SRAM at almost every cross-point of Enable and Data lines. In addition, the regularity of SRAM placement allows the CAD tool to keep only a minimum number of SRAM co-ordinates in the ADFF as reference while calculating all other co-ordinates by adding fixed offsets. For example, the co-ordinates of the SRAM bits in LUTs and 10x1 multiplexers for PLB 3, 4 and 2 can be calculated by adding different offsets to the SRAM co-ordinates of LUT and 10x1 multiplexer for PLB1.

This feature can be extended to a bigger scale when making a two-dimensional array of MTLCs and Switch Blocks (SB) forming an MT-FPGA and thus minimizing the SRAM co-ordinate database required for a complete MT-FPGA representation for the CAD software.

Figure-3.8: Less Wasted Bits in New Floorplanning
The signal bus in the new floorplanning is a 20-bit bus and has the following signals:

1. 4-bits for output signal distribution from the PLBs (1, 2, 3, 4)
2. 4-bits for bi-directional communication with the MTB (5, 6, 7, 8)
3. 8-bits for signals coming from other MTLCs through the Routing Channels (9, 10, 11, 12, 13, 14, 15, 16)
4. 4-bits for clock distribution, Reset and Configuration Enable (17, 18, 19, 20)
CHAPTER 4

4. SECOND GENERATION MTLC DESIGN EVALUATION

In the previous chapter, a detailed account of the research that went into finding various techniques to improve the first generation MTLC design was presented. All the measures suggested in the previous chapter culminated in the design and implementation of Second Generation MTLC (2G-MTLC). This chapter presents the part of the research that deals with the evaluation of the new design and its implementation in silicon.

Figure-4.1 shows the top-level schematic of the 2G-MTLC containing the four PLBs, one MTB, internal signal bus and Connection Blocks on all four sides.

![Figure-4.1: Top-Level 2G-MTLC Schematic](image-url)
Figure-4.2 is the corresponding full-custom 2G-MTLC layout with the MTB, conforming to the five metals and one polysilicon (1p5m) 0.25µm TSMC layout design rules.

Figure-4.2: Full-Custom 2G-MTLC Layout

The functional, timing and power estimation simulations were performed at both pre- and post-silicon stages of the design cycle. In the pre-silicon stage, simulations were performed on the extracted layout (netlist extracted from figure-4.2) - which contains distributed resistances and capacitances in addition to transistors, and transistor-level schematic of the proposed MTLC design implementation (netlist extracted from figure-4.1). While in the post-silicon stage, the 40-pin chip (code named T53U-AB) fabricated by the MOSIS foundry service was tested using a HP-16500A Logic Analyzer/Pattern Generator kit for data generation and acquisition. All timing response traces were collected using Agilent 54616B oscilloscope. The whole setup used for T53U-AB testing is shown in figure-4.3. Hspice was
Figure-4.3: Photograph of Test Setup for 2G-MTLC Chip

used for simulation because of it’s high accuracy. The full-custom transistor level schematic and layout of the design for 2.5V, 0.25µm TSMC CMOS process was made using Cadence Design System’s Virtuoso® Schematic and Layout Editors provided by Cadence Design Kit (CDK) from the North Carolina State University.

Before formal design evaluation begins, the concept of ‘manual’ MTLC programming, used in every form of verification of the new design discussed ahead, must be understood. Since CAD software tool discussed before are generally used only for programming the whole MT-FPGA, for just one MTLC, a bit-stream is manually generated and incorporated into the Hspice simulation as a bit-vector file. This manual generation requires the full understanding of each SRAM’s purpose in the design.

Since the delay timings can be measured using the same simulation result that checks for the correct design functionality, functional and timing verification for the logical and arithmetic operations are merged and presented in the sub-section 4.2. Functional verification of other
components is presented next. For the same reason, only the simulation result waveforms for extracted layout are shown as they are the closest to the post-silicon results. And wherever appropriate, results obtained from the post-silicon chip testing are also provided along with the per-silicon simulation results. Table-4.1, presented at the end of the chapter, summarizes all the timing, area and power data collected for the 2G-MTLC. The observations and results of the evaluation are as follows.

4.1 FUNCTIONAL VERIFICATION

Simulations were performed on the design to check the correctness of functionality of each sub-component of the 2G-MTLC. That is, test inputs were given and the outputs were compared against expected results to establish correctness.

4.1.1 MTLC PROGRAMMING

MTLC programming is performed using two shift-registers – one for enabling the correct column of SRAMs to be programmed and the other to feed data to those enabled SRAMs. The shift-registers operate on two non-overlapping clocks CLKA and CLKB, and have asynchronous Reset signal to clear the data in the registers. Figure-4.4 illustrates the scheme. The SRAMs are organized as a 32-by-7 matrix with a 7-bit shift-register providing the Enable signals to the seven columns and a 32-bit shift-register providing Data to the thirty two rows. The simulation waveforms
programming the 2G-MTLC as a 3-bit adder are shown in figure-4.5. The serial output of the 7-bit shift-register is observed after 7x32 clock cycles (each bit of the shift-register has to be 'high' for 32 cycles for the 32Bit shift-register to be loaded) and is the same as the input bit pattern, thus verifying the correctness of the column configuration chain. Similarly the 32-bit row configuration chain is verified. Figure-4.6 shows the corresponding result for the 7-bit shift-register obtained from the post-silicon chip testing. It indicates the correct functioning of the 7-bit shift-register required to program the 2G-MTLC. Similarly, the 32-bit shift-register was also found to function properly.
4.1.2 SCAN-CHAIN VERIFICATION

A 4-bit scan-chain is formed by the serial connection of user flip-flop in each PLB of the 2G-MTLC. In order to activate this scan-chain, first the 2G-MTLC has to be programmed to activate the sequential mode (that is, to activate the User Flip-Flops) using the appropriate bit-stream. Then the Test signal is asserted ‘High’, after which serial input is provided by the ScanIn signal and is shifted at every clock cycle using CLKA and CLKB. Figure-4.7 and 4.8 show the simulation and test results respectively. It establishes the correctness of the scan-chain as the same input bits can be observed at the output (ScanOut) after four clock cycles. So during chip testing, any output of the PLB can be observed at the ScanOut in order to deduce the correct functionality of the PLB as a whole logical entity.
Figure-4.7: 4-Bit Internal MTLC Scan-Chain

Figure-4.8: Post-Silicon Test Result For 4-Bit Internal MTLC Scan-Chain
4.2 TIMING VERIFICATION

In order to gather the timing data for logical and arithmetic mode of operations, first the 2G-MTLC is programmed ‘manually’ and then time delay between the observed outputs and input vectors is measured.

4.2.1 TYPICAL AND WORST CASE LOGICAL OPERATION DELAY

Typical logic operations use only one PLB to produce their output and therefore give the best case timing delay. On the other hand, worst case occurs when the output of PLB1 is used by PLB3 which in turn is used by PLB4 through soft-connects. This sequence is due to the L3-4.2 tree structure of the PLBs in the 2G-MTLC. Figure-4.9 shows the post-silicon typical timing delays, $t_{\text{plH}}$ and $t_{\text{plL}}$, for a PLB programmed as an inverter.

So, the typical propagation delay $t_p$ is given by-

$$t_p = \left( \frac{t_{\text{plH}} + t_{\text{plL}}}{2} \right)$$

$$t_p = \left( \frac{4.8 + 3.4}{2} \right)/\text{ns}$$

$$t_p = 4.1 \text{ ns}$$

Worst case logical operation delay is 7.9ns and corresponds to one of many scenarios where a three inverter ring oscillator is formed by PLB1, PLB3 and PLB4 using all soft-connects.
4.2.2 WORST CASE ADDITION OPERATIONS

Every time an addition operation is to be performed, a 3-bit Ripple Carry Adder (RCA) is formed between the PLB4 (LSB) – PLB2 – PLB1 – PLB3 (MSB) using the dedicated carry-chain. Again this sequence is due to the L3-4.2 tree structure in the 2G-MTLC. RCA is implemented as it is the simplest and takes the least area at the cost of speed, but only for wide-bit additions. So, the worst case occurs when the carry ripples through the adder from the LSB to the MSB, as in the case of $111 + 001 = 1000$ after 7.2ns, as shown in figure-4.10.

Figure-4.10: Worst Case Addition Operation
4.2.3 WORST CASE MULTIPLICATION OPERATIONS

For 2-bit multiplication operations, the PLBs generate partial products (logical AND) and perform ripple-carry addition on them as shown in figure-4.11. The worst case will involve rippling of carry, as in the case of $11 \times 11 = 1001$ after 6.3ns, shown in figure-4.12.

Figure-4.11: Multiplication in 2G-MTLC

Figure-4.12: Worst Case Multiplication Operation
4.2.4 FIRM-LINK VS SOFT-LINK TIMING DELAYS

To calculate the speed penalty of replacing a firm-link by a soft-link, three inverter ring oscillators are made by programming the PLBs as inverters and various combinations of hard-links or soft-links are used to connect them. This leads to three possible cases –

1. Two Firm-Links and one Soft-Link –

![Figure-4.13: Ring Oscillator With Two Firm-Links and One Soft-Link](image)

As shown in figure-4.13, ring oscillator is formed between PLB1, PLB3 and PLB4 with firm-links between PLB1 - PLB3 and PLB3 – PLB4, and soft-link between PLB4 – PLB1. Post-silicon test result shows the frequency of the ring oscillator as 82.48MHz with a time period of 12.12 ns, as depicted in figure-4.14.

![Figure-4.14: Post-Silicon Ring Oscillator Output With 2 Firm-Links & 1 Soft-Link](image)
2. One Firm-Link and Two Soft-Links –

As shown in figure-4.15, ring oscillator is formed between PLB1, PLB2 and PLB4 with firm-link between PLB2 – PLB4, and soft-links between PLB4 – PLB1 and PLB1 - PLB2. Post-silicon test result shows the frequency of the ring oscillator as 69.34MHz with a time period of 14.42 ns, as depicted in figure-4.16.

These results imply that the penalty for replacing a firm-link with a soft-link is (14.42ns - 12.12ns =) 2.3ns for TSMC 0.25µm process. So, a firm-link can be replaced by a soft-link wherever a timing penalty of 2.3ns can be tolerated.
3. All Soft-Links –

As shown in figure-4.17, ring oscillator is formed between PLB1, PLB2 and PLB3 with soft-links between all of them. Post-silicon test result shows the frequency of the ring oscillator as 63.29MHz with a time period of 15.8 ns, as depicted in figure-4.18.

These results imply that the penalty for replacing two firm-links with two soft-links is (15.80ns -12.12ns =) 3.68ns for TSMC 0.25µm process. So, two firm-links can be replaced by two soft-links wherever a timing penalty of 3.68ns can be tolerated.
4.3 POWER DISSIPATION ESTIMATION

Power dissipated by each sub-component of the 2G-MTLC is estimated in the pre-silicon stages through HSPICE simulations using specific power measurement statements for each sub-circuit. These simple results can be used to estimate power expenditure for cases where a larger number of similar sub-components are active within the 2G-MTLC.

Worst case power consumption occurs when all the four PLBs are active with highest rate of input switching activity. A subset of this scenario is a three inverter ring oscillator operating at its highest frequency, which is 82.48MHz where two firm-links and one soft-link are used. The power dissipated in the MTLC at this speed is close to 10mW.

4.4 AREA COMPUTATION

As already shown by figure-4.2, the area of the 2G-MTLC is 295µm x 215µm = 63425µm² as compared to 560µm x 560µm = 313600µm² of 1G-MTLC. This reduction of about 80% in area is the result of new floorplanning technique and technology scaling. For the same
CMOS technology (TSMC 0.35µm), area improvement of 27% is achieved through better floorplanning only, as shown in figure-4.19. Additional improvement in area utilization is achieved through technology scaling.

4.5 PROGRAMMING EFFICIENCY COMPUTATION

Programming Efficiency is given by the expression -

\[
PE = \left( \frac{\text{Useful Bits in the Bit-Stream}}{\text{Total Bits used in the Bit-Stream}} \right) \times 100\%
\]

Due to stack-based floorplanning in 2G-MTLC, there is less wastage of programming bits leading to \( \left( \frac{175}{224} \right) \times 100\% = 78.17\% \) programming efficiency. This means that almost every bit generated by the bit-stream finds a corresponding SRAM in the layout. In comparison, the programming efficiency of 1G-MTLC is 35.56%.

<table>
<thead>
<tr>
<th>TYPE OF COMPONENT</th>
<th>TYPE OF PROCESS</th>
<th>Delay Improvement (%)</th>
<th>Power Improvement (%)</th>
<th>Area Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TSMC 0.35µm</td>
<td>TSMC 0.25µm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technology</td>
<td>Technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Propagation Delay</td>
<td>Propagation Delay</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( t_{\text{PLH}} ) (ns)</td>
<td>( t_{\text{PLH}} ) (ns)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( t_{\text{PHL}} ) (ns)</td>
<td>( t_{\text{PHL}} ) (ns)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power mW at 33MHz</td>
<td>Power mW at 33MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( I=0.2\mu\text{A} )</td>
<td>( I=0.2\mu\text{A} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Input Look-Up Table</td>
<td>4-LUT</td>
<td>4 Input Multiplexer to 4-LUT</td>
<td>User Multiplexer</td>
<td>User Flip Flop</td>
</tr>
<tr>
<td>Time Delay, Power and Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>1.11</td>
<td>0.493</td>
<td>591x496</td>
<td>11725.44</td>
</tr>
<tr>
<td>0.95</td>
<td>0.8</td>
<td>0.045</td>
<td>188x252</td>
<td>1835.04</td>
</tr>
<tr>
<td>0.5</td>
<td>0.49</td>
<td>0.0699</td>
<td>79x151</td>
<td>477.16</td>
</tr>
<tr>
<td>0.72</td>
<td>0.7</td>
<td>0.0205</td>
<td>66x222</td>
<td>573.37</td>
</tr>
<tr>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Table-4.1: Summary of 2G-MTLC Timing, Area and Power Data

Table-4.1 gives the delay, area and power measurements of the various sub-components of the new 2G-MTLC and compares them with 1G-MTLC data provided in by the previous research [1]. Considerable improvement is made on all fronts except for increase in delay in the multiplexer to the 4bit-LUT because of the new structure. Carry-chain logic is a new feature of 2G-MTLC, so its performance cannot be compared to 1G-MTLC.
CHAPTER 5

5. CONCLUSIONS

After describing the design, simulation and test of the Second Generation Multi-Technology Field Programmable Gate Array Logic Cluster (2G-MTLC) in the previous chapters, this chapter compares the two generations of MTLC on area, speed, power and programming efficiency metrics, in addition to summarizing the work and making suggestions to future researchers interested in MT-FPGAs.

5.1 COMPARATIVE RESULTS OF THE TWO MTLC

<table>
<thead>
<tr>
<th>Figure of merits of an MTLC</th>
<th>TSMC 0.35μm Technology</th>
<th>TSMC 0.25μm Technology</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter (Propagation Delay)</td>
<td>tpLH: 9.6ns , tpHL: 6ns</td>
<td>tpLH: 4.8ns , tpHL: 3.4ns</td>
<td>47.4</td>
</tr>
<tr>
<td>Average Propagation Delay</td>
<td>9.6+6ns = 15.2ns</td>
<td>4.8+3.4ns = 8.2ns</td>
<td></td>
</tr>
<tr>
<td>Ring Oscillator (3 Inverters) Frequency</td>
<td>Frequency =</td>
<td>Frequency =</td>
<td></td>
</tr>
<tr>
<td>2 Firm-Links and 1 Soft-Link</td>
<td>64.5MHz</td>
<td>82.8MHz</td>
<td>27.1</td>
</tr>
<tr>
<td>1 Firm-Link and 2 Soft-Links</td>
<td>60.9MHz</td>
<td>58.3MHz</td>
<td>13.68</td>
</tr>
<tr>
<td>Time Penalty for replacing one Firm-Link by Soft-Link</td>
<td>1ns</td>
<td>2.3ns</td>
<td>-120</td>
</tr>
<tr>
<td>Power consumption per MTLC without Multi-Tech Block</td>
<td>&lt;17mW</td>
<td>&lt;10mW</td>
<td>41.2</td>
</tr>
<tr>
<td>MTLC area</td>
<td>31300μm²*2</td>
<td>6345μm²*2</td>
<td>79.78</td>
</tr>
<tr>
<td>Programming Efficiency</td>
<td>36.66%</td>
<td>78.17%</td>
<td>119.8</td>
</tr>
<tr>
<td>Testability of various Scan-Chains</td>
<td>All worked as expected</td>
<td>All worked as expected</td>
<td>NA</td>
</tr>
<tr>
<td>Other test cases</td>
<td>1) Demonstration of a clocked synchronous finite state machine (CS-FSM) 2) Combinational logic</td>
<td>1) Demonstration of a clocked synchronous finite state machine (CS-FSM) 2) Combinational logic</td>
<td>NA</td>
</tr>
<tr>
<td>Example circuits implemented</td>
<td>1) 1-bit slice pipelined full adder capable of adding two continuous serial bit streams 2) Different Combinational functions involving NAND/NOR/complement</td>
<td>1) 3-bit parallel Full Adder 2) 2-bit Multiplier 3) Different Combinational functions involving NAND/NOR/complement</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table-5.1: Comparative Results Between 1G-MTLC And 2G-MTLC
The results summarized in the table-5.1 above show that success has been achieved in designing and implementing a 2G-MTLC, which has better performance owing to new floorplanning, lower feature size CMOS fabrication technology and feature enhancements through circuit design. Major improvement has been made in terms of an 80% reduction in MTLC area due to the new floorplanning, optimum transistor sizing and lower feature size technology. New floorplanning also leads to a 120% improvement in programming efficiency leading to less time taken to program the MTLC. Power dissipation has been reduced by about 41%, though operating at a higher frequency, due to the reduction in supply voltage from 3.3V to 2.5V and the overall effective capacitance of the circuit resulting from a smaller layout. Speed of the MTLC has been improved by about 27% due to lower technology migration and tighter layout, again through new floorplanning. In addition, parallel 3-bit addition and 2-bit multiplication operations as performed by the 2G-MTLC give better performance than the just serial addition as performed by the 1G-MTLC. However, the penalty taken for replacing a firm-link by a soft-link has worsened from 1ns to 2.3ns because of the shift from 8X1 multiplexer to 10X1 multiplexer which had more transistors in the signal path.

5.2 SUMMARY OF THE THESIS

In this thesis, an effort has been made to remove most of the performance bottlenecks of the 1G-MTLC through detailed analysis of the problems and then suggesting remedies based on many design and implementation techniques available to a digital circuit designer, specifically, lower feature size, better floorplanning, optimum transistor sizing and functionality enhancement. This enables the 2G-MTLC to meet the performance requirements of the more sophisticated current multi-technology applications. Finally, the
three stages of design cycle—schematic development, layout design and chip fabrication, are covered with HSPICE simulation as the pre-silicon verification tool and HP Logic Analyzer as the validation tool. The MTLC designed in this research project is one of the important components in the development of a new 96-pin 2G-MTFPGA chip containing 5-row by 3-column 2G-MTLC array, using TSMC 0.25µm CMOS process technology through MOSIS foundry service. This chip development has been a collective effort of a three member team (including the author) in the PSDL group at the University of Cincinnati. The fullchip layout is shown in figure- 5.1.

Figure-5.1: 2G-MTFPGA Layout

5.3 SUGGESTIONS FOR FUTURE WORK
The research presented in this thesis has been an appreciable step forward in the development of more sophisticated MT-FPGAs. The future research can be pursued in several lines that can further increase the functionality and performance of the MT-FPGA.

1. **Implementing wider bit arithmetic operations**

   In the present research, only 3-bit wide addition and 2-bit wide multiplication operation can be performed due to layout constraints in the 2G-MTLC. So, by going for lower feature size technology than the current one and even better floorplanning if possible, provisions can be made to process wider bit input data which are more common in today’s applications. This can be done by running the dedicated carry-chain between adjacent MTLCs so that the carry-out bit from the PLB generating the most significant bit can be fed as carry-in to the PLB generating the least significant bit of the adjacent MTLC.

2. **Implementing on-chip memory**

   A further extension to logical and arithmetic capabilities provided by the current MT-FPGA will be the incorporation of on-chip memory. The SRAM bits used to store the function outputs in the 4-bit LUT can serve a dual purpose of storing intermediate results for applications that require lot of data processing. For achieving this goal, peripheral circuitry will have to be designed around the memory elements for generating the required control/data signals. In addition, separate consolidated memory arrays can be made which can be placed instead of the MTB as required by the range of target applications. Conventional FPGAs already have the feature of distributed or centralized on-chip memories [5]. The presence of these memory elements increases the operating frequency of the FPGA as off-chip memory communication is eliminated.
BIBLIOGRAPHY


**APPENDIX-A**

**MOSIS PARAMETRIC TEST RESULTS FOR TSMC 0.25µm TECHNOLOGY**

**RUN:** N94S  
**TECHNOLOGY:** SCN025  
**FEATURE SIZE:** 0.25 microns  
**VENDOR:** TSMC

**INTRODUCTION:** This report contains the lot average results obtained by MOSIS from measurements of MOSIS test structures on each wafer of this fabrication lot. SPICE parameters obtained from similar measurements on a selected wafer are also attached.

**COMMENTS:** TSMC 025SPPM.

<table>
<thead>
<tr>
<th>TRANSISTOR PARAMETERS</th>
<th>W/L</th>
<th>N-CHANNEL</th>
<th>P-CHANNEL</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINIMUM</td>
<td>0.36/0.24</td>
<td>0.56</td>
<td>-0.51</td>
<td>Volts</td>
</tr>
<tr>
<td>Vth</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHORT</td>
<td>20/0.24</td>
<td>547</td>
<td>-262</td>
<td>uA/um</td>
</tr>
<tr>
<td>Idss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vth</td>
<td>0.59</td>
<td>-0.54</td>
<td></td>
<td>Volts</td>
</tr>
<tr>
<td>Vpt</td>
<td>7.6</td>
<td>-5.7</td>
<td></td>
<td>Volts</td>
</tr>
<tr>
<td>WIDE</td>
<td>20/0.24</td>
<td></td>
<td>-7.2</td>
<td>Volts</td>
</tr>
<tr>
<td>Ids0</td>
<td>6.2</td>
<td>-3.7</td>
<td></td>
<td>pA/um</td>
</tr>
<tr>
<td>LARGE</td>
<td>20/20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vth</td>
<td>0.51</td>
<td>-0.57</td>
<td></td>
<td>Volts</td>
</tr>
<tr>
<td>Vjbkd</td>
<td>6.1</td>
<td>-7.0</td>
<td></td>
<td>Volts</td>
</tr>
<tr>
<td>Ijlk</td>
<td>-29.3</td>
<td>-6.5</td>
<td></td>
<td>pA</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.39</td>
<td>0.52</td>
<td></td>
<td>V^0.5</td>
</tr>
<tr>
<td>K' (Uo*Cox/2)</td>
<td>108.6</td>
<td>-26.5</td>
<td></td>
<td>uA/V^2</td>
</tr>
</tbody>
</table>

**COMMENTS:** Poly bias varies with design technology. To account for mask and etch bias use the appropriate value for the parameter XL in your SPICE model card.

<table>
<thead>
<tr>
<th>Design Technology</th>
<th>XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCN5M_DEEP (lambda=0.12)</td>
<td>0.03</td>
</tr>
<tr>
<td>thick oxide, NMOS</td>
<td>0.02</td>
</tr>
<tr>
<td>thick oxide, PMOS</td>
<td>-0.03</td>
</tr>
<tr>
<td>TSMC25</td>
<td>0.03</td>
</tr>
<tr>
<td>thick oxide, NMOS</td>
<td>0.03</td>
</tr>
<tr>
<td>thick oxide, PMOS</td>
<td>0.03</td>
</tr>
<tr>
<td>SCN3M_SUBM (lambda=0.15)</td>
<td>-0.03</td>
</tr>
<tr>
<td>thick oxide, NMOS</td>
<td>0.02</td>
</tr>
<tr>
<td>thick oxide, PMOS</td>
<td>-0.03</td>
</tr>
</tbody>
</table>

1 http://www.mosis.org/cgi-bin/cgiwrap/umosis/swp/params/tsmc-025/n94s-params.txt
<table>
<thead>
<tr>
<th>FOX TRANSISTORS</th>
<th>GATE</th>
<th>N+ACTIVE</th>
<th>P+ACTIVE</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth</td>
<td>Poly</td>
<td>&gt;15.0</td>
<td>&lt;-15.0</td>
<td>Volts</td>
</tr>
</tbody>
</table>

**PROCESS PARAMETERS**

<table>
<thead>
<tr>
<th>UNITS</th>
<th>N+ACTV</th>
<th>P+ACTV</th>
<th>POLY</th>
<th>MTL1</th>
<th>MTL2</th>
<th>MTL3</th>
<th>MTL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet Resistance</td>
<td>4.7</td>
<td>3.5</td>
<td>4.2</td>
<td>0.06</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>Width Variation</td>
<td>0.05</td>
<td>0.13</td>
<td>0.10</td>
<td>-0.19</td>
<td>-0.00</td>
<td>-0.04</td>
<td>-0.07</td>
</tr>
<tr>
<td>(measured - drawn)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Resistance</td>
<td>6.7</td>
<td>5.7</td>
<td>5.7</td>
<td>2.02</td>
<td>4.07</td>
<td>5.79</td>
<td></td>
</tr>
<tr>
<td>Gate Oxide Thickness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58</td>
</tr>
</tbody>
</table>

**PROCESS PARAMETERS**

<table>
<thead>
<tr>
<th>MTL5</th>
<th>N_WELL</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet Resistance</td>
<td>0.03</td>
<td>1191</td>
</tr>
<tr>
<td>Width Variation</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>(measured - drawn)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Resistance</td>
<td>8.13</td>
<td></td>
</tr>
</tbody>
</table>

**CAPACITANCE PARAMETERS**

<table>
<thead>
<tr>
<th>N+ACTV</th>
<th>P+ACTV</th>
<th>POLY</th>
<th>MTL1</th>
<th>MTL2</th>
<th>MTL3</th>
<th>MTL4</th>
<th>MTL5</th>
<th>N_WELL</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (substrate)</td>
<td>1872</td>
<td>1877</td>
<td>97</td>
<td>38</td>
<td>19</td>
<td>13</td>
<td>8</td>
<td>8</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (N+active)</td>
<td>5912</td>
<td>50</td>
<td>20</td>
<td>14</td>
<td>11</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (P+active)</td>
<td>5691</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (poly)</td>
<td>63</td>
<td>17</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (metal1)</td>
<td>37</td>
<td>15</td>
<td>9</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (metal2)</td>
<td>38</td>
<td>15</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (metal3)</td>
<td>38</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area (metal4)</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um^2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (substrate)</td>
<td>440</td>
<td>352</td>
<td>23</td>
<td>60</td>
<td>56</td>
<td>42</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (poly)</td>
<td>70</td>
<td>42</td>
<td>30</td>
<td>24</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (metal1)</td>
<td>52</td>
<td>36</td>
<td>29</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (metal2)</td>
<td>49</td>
<td>36</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (metal3)</td>
<td>52</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fringe (metal4)</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap (N+active)</td>
<td>627</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap (P+active)</td>
<td>559</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aF/um</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CIRCUIT PARAMETERS**

<table>
<thead>
<tr>
<th>UNITS</th>
</tr>
</thead>
</table>
### Inverters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>K</th>
<th>K</th>
<th>Volts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vinv</td>
<td>1.0</td>
<td>1.03</td>
<td></td>
</tr>
<tr>
<td>Vinv (100 uA)</td>
<td>1.08</td>
<td>1.53</td>
<td>1.12</td>
</tr>
<tr>
<td>Voh (100 uA)</td>
<td>1.44</td>
<td>2.06</td>
<td>0.30</td>
</tr>
<tr>
<td>Vinv</td>
<td>1.44</td>
<td>2.06</td>
<td>1.18</td>
</tr>
<tr>
<td>Gain</td>
<td>1.44</td>
<td>2.06</td>
<td>-18.28</td>
</tr>
</tbody>
</table>

**COMMENTS:** DEEP_SUBMICRON

---

### N94S SPICE BSIM3 VERSION 3.1 PARAMETERS

**SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8**

* DATE: Feb 9/01
* LOT: N94S                WAF: 08
* Temperature_parameters=Default

```plaintext
.MODEL CMOSN NMOS (LEVEL = 49)
+VERSION = 3.1
+TNOM = 27
+TOX = 5.8E-9
+XJ = 1E-7
+NCH = 2.3549E17
+VTH0 = 0.408619
+K1 = 0.4144712
+K2 = 0.0122732
+K3 = 1.000752E-3
+K3B = 0.8229319
+W0 = 1.000961E-7
+NLX = 2.808104E-7
+DVT0W = 0
+DVT1W = 0
+DVT2W = 0
+DVT0 = 0.211976
+DVT1 = 0.0981279
+DVT2 = -0.073355
+U0 = 345.4738707
+UA = -6.65353E-10
+UB = 2.037713E-18
+UC = 3.117558E-11
+VSAT = 8.99117E4
+A0 = 0.631435
+AGS = 0.1288136
+B0 = 3.658209E-7
+B1 = 1.05362E-6
+KETA = 6.322964E-3
+A1 = 9.606552E-4
+A2 = 1
+RDSW = 120
+PRWG = 0.5
+PRWB = -0.2
+WR = 1
+WINT = 1.765394E-9
+LINT = -2.126447E-8
+XL = 3E-8
+XW = 0
+DWG = -2.42828E-9
+DWB = 9.185176E-9
+VOPF = -0.0976674
+NFACTOR = 0
+UCIT = 0
+CDSC = 2.4E-4
+CDSCD = 0
+CDSCB = 0
+ETA0 = 0.0491777
+ETAB = 3.125491E-3
+DSUB = 0.2870242
+PCLM = 1.8534523
+PDIBC1 = 0.6857768
+PDIBLC2 = 0.01
+PDIBC = 0.050458
+DROUT = 1
+PSCBE1 = 8E10
+PSCBE2 = 2.278455E-8
+PVAG = 0.2666656
+DELTA = 0.01
+RSH = 4.7
+MOBMOD = 1
+PRT = 0
+UTE = -1.5
+KT1 = -0.11
+KT1L = 0
+KT2 = 0.022
+UA1 = 4.31E-9
+UB1 = -7.61E-18
+UC1 = -5.6E-11
+AT = 3.3E4
+WL = 0
+WLN = 1
+WW = -1.22182E-16
+WLN = 1.2127
+WNL = 0
+LL = 0
+LLN = 1
+LWN = 1
+LWL = 0
+CAPMOD = 2
+XPART = 0.4
+CGDO = 6.27E-10
+CGSO = 6.27E-10
+CGBO = 1E-12
+CGJ = 1.916804E-3
+PB = 0.99
+MJ = 0.4744225
```
CJSW = 4.549787E-10  PBSW = 0.99  MJSW = 0.3235275
CF = 0  PVTH0 = -8.631195E-3  PRDSW = 0
PK2 = 3.5315E-3  WK ETA = -4.378878E-3  LK ETA = -0.0351436

.MODEL CMOSP PMOS ( LEVEL = 49
.VERSION = 3.1
+NXT  = 1E-7  NCH = 4.1589E17  VTH0 = -0.556523
+K1  = 0.5416631  K2 = 0.0245545  K3 = 0
+K3B = 8.6601778  W0 = 6.279195E-6  NLX = 1E-9
+DVTOW = 0  DVT1W = 0
+DVT0 = 4.3499141  DVT1 = 0.7927238  DVT2 = -0.072391
+U0 = 134.7042677  UA = 1.795355E-9  UB = 1E-21
+UC = -1E-10  VSAT = 134.7042677  U0 = 1.34.7042677
+AGS = 0.1869313  B0 = 1.602029E-6  B1 = 5E-6
+KETA = 0.0127477  A1 = 1.818942E-3  A2 = 0.576764
+RDSW = 1.418086E3  PRWG = 0.0500478  PRWB = -0.2753995
+WR = 1  WINT = -1.899142E-8  LINT = 2.725798E-8
+XR = 3E-8  WQ = 0
+DWB = 3.679257E-9  WQF = 0.0925825  NFACTOR = 0
+CID = 0  CDSC = 2.4E-4  CDSCD = 0
+CDSCB = 0  ETA0 = 0.2  ETA = -
+DSUB = 1.1514586  PCLM = 1.8859363  PDIBLC1 = 0
+PDIBLC2 = 0.0969611  PDIBLCB = -1E-3  DROUT = 0.3487536
+PSCBE1 = 2.171928E10  PSCBE2 = 6.262781E-9  PVAG = 15
+DELT = 0.01  RSH = 3.5  MOBMOD = 1
+PRT = 0  UTE = -1.5  KT1 = -0.11
+KT1L = 0  KT2 = 0.022  UA1 = 4.31E-9
+UB1 = -7.61E-18  UC1 = -5.6E-11  AT = 3.3E4
+WL = 0  WLN = 1  WW = 0
+WLN = 1  WWL = 0  LL = 0
+LLN = 1  LW = 0  LWN = 1
+WNL = 0  CAPMOD = 2  XPART = 0.4
+CGDO = 5.59E-10  CGSO = 5.59E-10  CGBO = 1E-12
+CJ = 1.883133E-3  PB = 0.9807855  MJ = 0.4655692
+CJSW = 3.733241E-10  PBSW = 0.7634647  MJSW = 0.3208898
+CF = 0  PVTH0 = 6.146287E-3  PRDSW = -9.6506777
+PK2 = 3.091063E-3  WK ETA = 6.365227E-3  LK ETA = -0.0103601
)
APPENDIX-B

DETAILS OF SRAM DISTRIBUTION IN THE MTLC

SRAM Distribution In The MTLC
DESCRIPTION OF CONFIGURATION BITS IN THE MTLC

\[ S_{X,Y} = \text{SRAM at the cross-point of Data Line \# X and Enable Line \# Y} \]

Description Of MTLC Connection With The Routing Channels

1. For PLB1

A) 4-Bit LUT Inputs

<table>
<thead>
<tr>
<th>Input ( a_0 )</th>
<th>Input ( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_4,0 S_5,0 S_6,0 S_7,0 ) = Selection Bits</td>
<td>( S_3,0 S_2,0 S_1,0 S_0,0 ) = Selection Bits</td>
</tr>
<tr>
<td>0000 → PLB1 Output</td>
<td>0000 → PLB1 Output</td>
</tr>
<tr>
<td>0001 → PLB2 Output</td>
<td>0001 → PLB2 Output</td>
</tr>
<tr>
<td>0010 → PLB3 Output</td>
<td>0010 → PLB3 Output</td>
</tr>
<tr>
<td>0011 → PLB4 Output</td>
<td>0011 → PLB4 Output</td>
</tr>
<tr>
<td>0100 → RC1</td>
<td>0100 → RC2</td>
</tr>
<tr>
<td>0101 → RC2</td>
<td>0101 → RC3</td>
</tr>
<tr>
<td>0110 → RC7</td>
<td>0110 → RC8</td>
</tr>
<tr>
<td>0111 → RC8</td>
<td>0111 → RC5</td>
</tr>
<tr>
<td>1000 → mtb1</td>
<td>1000 → mtb2</td>
</tr>
<tr>
<td>1001 → mtb2</td>
<td>1001 → mtb3</td>
</tr>
<tr>
<td>other → Not Allowed</td>
<td>other → Not Allowed</td>
</tr>
</tbody>
</table>
Input $a_2$

$S_{4,5}S_{5,5}S_{6,5}S_{7,5} = \text{Selection Bits}$

0000 $\rightarrow$ PLB1 Output
0001 $\rightarrow$ PLB2 Output
0010 $\rightarrow$ PLB3 Output
0011 $\rightarrow$ PLB4 Output
0100 $\rightarrow$ RC3
0101 $\rightarrow$ RC4
0110 $\rightarrow$ RC5
0111 $\rightarrow$ RC6
1000 $\rightarrow$ mtb3
1001 $\rightarrow$ mtb4
other $\rightarrow$ Not Allowed

Input $a_3$

$S_{3,5}S_{2,5}S_{1,5}S_{0,5} = \text{Selection Bits}$

0000 $\rightarrow$ PLB1 Output
0001 $\rightarrow$ PLB2 Output
0010 $\rightarrow$ PLB3 Output
0011 $\rightarrow$ PLB4 Output
0100 $\rightarrow$ RC4
0101 $\rightarrow$ RC1
0110 $\rightarrow$ RC6
0111 $\rightarrow$ RC7
1000 $\rightarrow$ mtb4
1001 $\rightarrow$ mtb1
other $\rightarrow$ Not Allowed

B) 4-Bit LUT Contents

4-Bit LUT Input $a_3 \ a_2 \ a_1 \ a_0 \rightarrow \overline{S_{X,Y}}$ as Output

\[
\begin{align*}
0000 & \rightarrow \overline{S_{0,2}} \\
0001 & \rightarrow \overline{S_{1,2}} \\
0010 & \rightarrow \overline{S_{2,2}} \\
0011 & \rightarrow \overline{S_{3,2}} \\
0100 & \rightarrow \overline{S_{4,2}} \\
0101 & \rightarrow \overline{S_{5,2}} \\
0110 & \rightarrow \overline{S_{6,2}} \\
0111 & \rightarrow \overline{S_{7,2}} \\
1000 & \rightarrow \overline{S_{0,3}} \\
1001 & \rightarrow \overline{S_{1,3}} \\
1010 & \rightarrow \overline{S_{2,3}} \\
1011 & \rightarrow \overline{S_{3,3}} \\
1100 & \rightarrow \overline{S_{4,3}} \\
1101 & \rightarrow \overline{S_{5,3}} \\
1110 & \rightarrow \overline{S_{6,3}} \\
1111 & \rightarrow \overline{S_{7,3}}
\end{align*}
\]

C) Carry-Logic

- $S_{2,4}$ Selection bit is used to select between Addition and Multiplication operation
  
  $0 \rightarrow$ Addition
  $1 \rightarrow$ Multiplication

- $S_{5,4}$ Selection Bit is used to select between Logical (Only LUT output) and Arithmetic (Addition/Multiplication) operation
  
  $0 \rightarrow$ Logical operation
D) User Multiplexer

- $S_{2,1}$ Selection bit is used to select between Carry-logic output and User Flip-flop output (to realize a sequential circuit)

0 $\rightarrow$ Carry-logic output
1 $\rightarrow$ User Flip-flop output

2. For PLB2

A) 4-Bit LUT Inputs

<table>
<thead>
<tr>
<th>Input $a_0$</th>
<th>Input $a_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{28,0}S_{29,0}S_{30,0}S_{31,0}$ = Selection Bits</td>
<td>$S_{27,0}S_{26,0}S_{25,0}S_{24,0}$ = Selection Bits</td>
</tr>
<tr>
<td>0000 $\rightarrow$ PLB1 Output</td>
<td>0000 $\rightarrow$ PLB1 Output</td>
</tr>
<tr>
<td>0001 $\rightarrow$ PLB2 Output</td>
<td>0001 $\rightarrow$ PLB2 Output</td>
</tr>
<tr>
<td>0010 $\rightarrow$ PLB3 Output</td>
<td>0011 $\rightarrow$ PLB3 Output</td>
</tr>
<tr>
<td>0011 $\rightarrow$ PLB4 Output</td>
<td>0011 $\rightarrow$ PLB4 Output</td>
</tr>
<tr>
<td>0100 $\rightarrow$ RC3</td>
<td>0100 $\rightarrow$ RC4</td>
</tr>
<tr>
<td>0101 $\rightarrow$ RC4</td>
<td>0101 $\rightarrow$ RC1</td>
</tr>
<tr>
<td>0110 $\rightarrow$ RC5</td>
<td>0110 $\rightarrow$ RC6</td>
</tr>
<tr>
<td>0111 $\rightarrow$ RC6</td>
<td>0111 $\rightarrow$ RC7</td>
</tr>
<tr>
<td>1000 $\rightarrow$ mtb1</td>
<td>1000 $\rightarrow$ mtb2</td>
</tr>
<tr>
<td>1001 $\rightarrow$ mtb2</td>
<td>1001 $\rightarrow$ mtb3</td>
</tr>
<tr>
<td>other $\rightarrow$ Not Allowed</td>
<td>other $\rightarrow$ Not Allowed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input $a_2$</th>
<th>Input $a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{28,5}S_{29,5}S_{30,5}S_{31,5}$ = Selection Bits</td>
<td>$S_{27,5}S_{26,5}S_{25,5}S_{24,5}$ = Selection Bits</td>
</tr>
<tr>
<td>0000 $\rightarrow$ PLB1 Output</td>
<td>0000 $\rightarrow$ PLB1 Output</td>
</tr>
<tr>
<td>0001 $\rightarrow$ PLB2 Output</td>
<td>0001 $\rightarrow$ PLB2 Output</td>
</tr>
<tr>
<td>0010 $\rightarrow$ PLB3 Output</td>
<td>0010 $\rightarrow$ PLB3 Output</td>
</tr>
<tr>
<td>0011 $\rightarrow$ PLB4 Output</td>
<td>0011 $\rightarrow$ PLB4 Output</td>
</tr>
<tr>
<td>0100 $\rightarrow$ RC1</td>
<td>0100 $\rightarrow$ RC2</td>
</tr>
<tr>
<td>0101 $\rightarrow$ RC2</td>
<td>0101 $\rightarrow$ RC3</td>
</tr>
<tr>
<td>0110 $\rightarrow$ RC7</td>
<td>0110 $\rightarrow$ RC8</td>
</tr>
<tr>
<td>0111 $\rightarrow$ RC8</td>
<td>0111 $\rightarrow$ RC5</td>
</tr>
<tr>
<td>1000 $\rightarrow$ mtb3</td>
<td>1000 $\rightarrow$ mtb4</td>
</tr>
<tr>
<td>1001 $\rightarrow$ mtb4</td>
<td>1001 $\rightarrow$ mtb1</td>
</tr>
<tr>
<td>1010 $\rightarrow$ Not Allowed</td>
<td>1010 $\rightarrow$ Not Allowed</td>
</tr>
</tbody>
</table>

B) 4-Bit LUT Contents
4-Bit LUT Input $a_3 \ a_2 \ a_1 \ a_0 \xrightarrow{SRAM\ Location} S_{x,y}$ as Output

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>$S_{24,2}$</td>
</tr>
<tr>
<td>0001</td>
<td>$S_{25,2}$</td>
</tr>
<tr>
<td>0010</td>
<td>$S_{26,2}$</td>
</tr>
<tr>
<td>0011</td>
<td>$S_{27,2}$</td>
</tr>
<tr>
<td>0100</td>
<td>$S_{28,2}$</td>
</tr>
<tr>
<td>0101</td>
<td>$S_{29,2}$</td>
</tr>
<tr>
<td>0110</td>
<td>$S_{30,2}$</td>
</tr>
<tr>
<td>0111</td>
<td>$S_{31,2}$</td>
</tr>
<tr>
<td>1000</td>
<td>$S_{24,3}$</td>
</tr>
<tr>
<td>1001</td>
<td>$S_{25,3}$</td>
</tr>
<tr>
<td>1010</td>
<td>$S_{26,3}$</td>
</tr>
<tr>
<td>1011</td>
<td>$S_{27,3}$</td>
</tr>
<tr>
<td>1100</td>
<td>$S_{28,3}$</td>
</tr>
<tr>
<td>1101</td>
<td>$S_{29,3}$</td>
</tr>
<tr>
<td>1110</td>
<td>$S_{30,3}$</td>
</tr>
<tr>
<td>1111</td>
<td>$S_{31,3}$</td>
</tr>
</tbody>
</table>

C) Carry-Logic

- $S_{26,4}$ Selection bit is used to select between Addition and Multiplication operation
  
  0 $\rightarrow$ Addition
  1 $\rightarrow$ Multiplication

- $S_{29,4}$ Selection Bit is used to select between Logical (Only LUT output) and Arithmetic (Addition/Multiplication) operation
  
  0 $\rightarrow$ Logical operation
  1 $\rightarrow$ Arithmetic operation

D) User Multiplexer

- $S_{26,1}$ Selection bit is used to select between Carry-logic output and User Flip-flop output (to realize a sequential circuit)
  
  0 $\rightarrow$ Carry-logic output
  1 $\rightarrow$ User Flip-flop output

3. For PLB3

A) 4-Bit LUT Inputs

<table>
<thead>
<tr>
<th>Input $a_0$</th>
<th>Input $a_1$</th>
</tr>
</thead>
</table>

Hard-wired PLB1 Output $S_{11,0}S_{10,0}S_{9,0}S_{8,0}$ = Selection Bits

0000 $\rightarrow$ VDD
0000 → GND
0001 → PLB2 Output
0010 → PLB3 Output
0011 → PLB4 Output
0100 → RC2
0101 → RC3
0110 → RC8
0111 → RC5
1000 → mtb1
1001 → mtb2
other → Not Allowed

B) 4-Bit LUT Contents

\[ S_{12,5}S_{13,5}S_{14,5}S_{15,5} = \text{Selection Bits} \]
\[ S_{11,5}S_{10,5}S_{9,5}S_{8,5} = \text{Selection Bits} \]

\[ \begin{align*}
0000 & \rightarrow \overline{S_{8,2}} \\
0001 & \rightarrow \overline{S_{9,2}} \\
0010 & \rightarrow \overline{S_{10,2}} \\
0011 & \rightarrow \overline{S_{11,2}} \\
0100 & \rightarrow \overline{S_{12,2}} \\
0101 & \rightarrow \overline{S_{13,2}} \\
0110 & \rightarrow \overline{S_{14,2}} \\
0111 & \rightarrow \overline{S_{15,2}} \\
1000 & \rightarrow \overline{S_{8,3}} \\
1001 & \rightarrow \overline{S_{9,3}} \\
1010 & \rightarrow \overline{S_{10,3}} \\
1011 & \rightarrow \overline{S_{11,3}} \\
1100 & \rightarrow \overline{S_{12,3}} \\
1101 & \rightarrow \overline{S_{13,3}} \\
1110 & \rightarrow \overline{S_{14,3}} \\
1111 & \rightarrow \overline{S_{15,3}}
\end{align*} \]

C) Carry-Logic
• $S_{11,4}$ Selection bit is used to select between Carry-In signal ‘0’ and ‘1’

0 → Carry-In is ‘1’
1 → Carry-In is ‘0’

• $S_{13,4}$ Selection Bit is used to select between Logical (Only LUT output) and Arithmetic (Addition/Multiplication) operation

0 → Logical operation
1 → Arithmetic operation

D) User Multiplexer

• $S_{10,1}$ Selection bit is used to select between Carry-logic output and User Flip-flop output (to realize a sequential circuit)

0 → Carry-logic output
1 → User Flip-flop output

4. For PLB4

A) 4-Bit LUT Inputs

Input $a_0$  Input $a_1$

Hard-wired PLB3 Output  Hard-wired PLB3 Output

Input $a_2$  Input $a_3$

$S_{20,5}S_{21,5}S_{22,5}S_{23,5}$ = Selection Bits  $S_{19,5}S_{18,5}S_{17,5}S_{16,5}$ = Selection Bits

0000 → PLB1 Output  0000 → PLB1 Output
0001 → VDD  0001 → VDD
0010 → GND  0010 → GND
0011 → PLB4 Output  0011 → PLB4 Output
0100 → RC1  0100 → RC3
0101 → RC2  0101 → RC4
0110 → RC7  0110 → RC5
0111 → RC8  0111 → RC6
1000 → mtb1  1000 → mtb3
1001 → mtb2  1001 → mtb4
other → Not Allowed  other → Not Allowed

B) 4-Bit LUT Contents

4-Bit LUT Input $a_3 \ a_2 \ a_1 \ a_0$ Selects SRAM Location
$S_{x,r}$ as Output
<table>
<thead>
<tr>
<th>Binary</th>
<th>Selection</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>S_{16,2}</td>
<td>1000 → S_{16,3}</td>
</tr>
<tr>
<td>0001</td>
<td>S_{17,2}</td>
<td>1001 → S_{17,3}</td>
</tr>
<tr>
<td>0010</td>
<td>S_{18,2}</td>
<td>1010 → S_{18,3}</td>
</tr>
<tr>
<td>0011</td>
<td>S_{19,2}</td>
<td>1011 → S_{19,3}</td>
</tr>
<tr>
<td>0100</td>
<td>S_{20,2}</td>
<td>1100 → S_{20,3}</td>
</tr>
<tr>
<td>0101</td>
<td>S_{21,2}</td>
<td>1101 → S_{21,3}</td>
</tr>
<tr>
<td>0110</td>
<td>S_{22,2}</td>
<td>1110 → S_{22,3}</td>
</tr>
<tr>
<td>0111</td>
<td>S_{23,2}</td>
<td>1111 → S_{23,3}</td>
</tr>
</tbody>
</table>

C) **Carry-Logic**

- **S\_{18,4}**: Selection bit is used to select between Addition and Multiplication operation
  
  0 → Addition  
  1 → Multiplication

- **S\_{21,4}**: Selection Bit is used to select between Logical (Only LUT output) and Arithmetic (Addition/Multiplication) operation
  
  0 → Logical operation  
  1 → Arithmetic operation

D) **User Multiplexer**

- **S\_{20,1}**: Selection bit is used to select between Carry-logic output and User Flip-flop output (to realize a sequential circuit)
  
  0 → Carry-logic output  
  1 → User Flip-flop output

5. **For Connection Block NORTH**

A) **De-multiplexer**

- PLB1 Output → One of Routing Channel Lines E4, E5, E6, E7 using selection bits S_{1,6}, S_{2,6} and Tri-stated through S_{0,6}

\[
\begin{array}{c c c c}
S_{0,6} & S_{1,6} & S_{2,6} & Selects Output \rightarrow \text{RC#} \\
0 & X & X & Z \\
1 & 0 & 0 & \rightarrow \text{E4}
\end{array}
\]
B) Two Digital Multiplexers

- One of Routing Channel Lines E1, E5, E7, E9 → RC5 Line in the signal bus through selection bits $S_{4,6}, S_{3,6}$

\[
\begin{array}{cccc}
0 & 0 & \rightarrow & E1 \\
0 & 1 & \rightarrow & E5 \\
1 & 0 & \rightarrow & E7 \\
1 & 1 & \rightarrow & E9
\end{array}
\]

- One of Routing Channel Lines E0, E2, E4, E8 → RC1 Line in the signal bus through selection bits $S_{6,6}, S_{5,6}$

\[
\begin{array}{cccc}
0 & 0 & \rightarrow & E0 \\
0 & 1 & \rightarrow & E2 \\
1 & 0 & \rightarrow & E4 \\
1 & 1 & \rightarrow & E8
\end{array}
\]

C) Two Analog Multiplexers

- One of Analog Routing Channel Lines AR0, AR1 → mtb1 Line in the signal bus through selection bit $S_{7,6}$ and Tri-state through $S_{8,6}$

\[
\begin{array}{cccc}
0 & X & \rightarrow & Z \\
1 & 0 & \rightarrow & AR0 \\
1 & 1 & \rightarrow & AR1
\end{array}
\]

- One of signal bus lines mtb1, mtb2 → Analog Routing Channel Lines AR0 through selection bit $S_{10,6}$ and Tri-state through $S_{9,6}$

\[
\begin{array}{cccc}
0 & X & \rightarrow & Z \\
1 & 0 & \rightarrow & mtb1 \\
1 & 1 & \rightarrow & mtb2
\end{array}
\]

6. For Connection Block EAST
A) PLB Output De-multiplexer

- PLB4 Output → One of Routing Channel Lines S0, S1, S2, S3 using selection bits $S_{14,6}$, $S_{13,6}$ and Tri-stated through $S_{15,6}$

$S_{15,6}$ $S_{14,6}$ $S_{13,6}$ \( \xrightarrow{\text{Selects Output}} \) RC#

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>RC#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>S0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>S1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>S3</td>
</tr>
</tbody>
</table>

B) Two Digital Multiplexers

- One of Routing Channel Lines S1, S3, S5, S7 → RC4 Line in the signal bus through selection bits $S_{19,1}$, $S_{20,6}$

$S_{19,1}$ $S_{20,6}$ \( \xrightarrow{\text{Selects Input}} \) RC#

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>RC#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>S1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>S3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>S5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>S7</td>
</tr>
</tbody>
</table>

One of Routing Channel Lines S2, S4, S6, S8 → RC8 Line in the signal bus through selection bits $S_{19,6}$, $S_{18,6}$

$S_{19,6}$ $S_{18,6}$ \( \xrightarrow{\text{Selects Input}} \) RC#

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>RC#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>S2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>S4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>S6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>S8</td>
</tr>
</tbody>
</table>

C) Digital Multiplexer and De-multiplexers

- One of Routing Channel Lines S5, S6 → mtb3 Line in the signal bus through selection bit $S_{11,6}$ and Tri-state through $S_{12,6}$

$S_{12,6}$ $S_{11,6}$ \( \xrightarrow{\text{Selects Input}} \) RC#

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>RC#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>S5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>S6</td>
</tr>
</tbody>
</table>

- Signal bus line mtb4 → S3, S4 through selection bit $S_{17,6}$ and Tri-state
through $S_{16,6}$

\[
S_{16,6} S_{17,6} \xrightarrow{Selects} \xrightarrow{Output} RC#
\]

\[
\begin{align*}
0 & \quad X \quad \rightarrow \quad Z \\
1 & \quad 0 \quad \rightarrow \quad S3 \\
1 & \quad 1 \quad \rightarrow \quad S4
\end{align*}
\]

7. For Connection Block SOUTH

A) De-multiplexer

- PLB2 Output $\rightarrow$ One of Routing Channel Lines E2, E3, E4, E5 using selection bits $S_{30,6} S_{29,6}$ and Tri-stated through $S_{31,6}$

\[
S_{31,6} S_{30,6} S_{29,6} \xrightarrow{Selects} \xrightarrow{Output} RC#
\]

\[
\begin{align*}
0 & \quad X \quad X \quad \rightarrow \quad Z \\
1 & \quad 0 \quad 0 \quad \rightarrow \quad E2 \\
1 & \quad 0 \quad 1 \quad \rightarrow \quad E3 \\
1 & \quad 1 \quad 0 \quad \rightarrow \quad E4 \\
1 & \quad 1 \quad 1 \quad \rightarrow \quad E5
\end{align*}
\]

B) Two Digital Multiplexers

- One of Routing Channel Lines E1, E5, E7, E9 $\rightarrow$ RC6 Line in the signal bus through selection bits $S_{4,6} S_{3,6}$

\[
S_{27,6} S_{28,6} \xrightarrow{Selects} \xrightarrow{Input} RC#
\]

\[
\begin{align*}
0 & \quad 0 \quad \rightarrow \quad E1 \\
0 & \quad 1 \quad \rightarrow \quad E5 \\
1 & \quad 0 \quad \rightarrow \quad E7 \\
1 & \quad 1 \quad \rightarrow \quad E9
\end{align*}
\]

- One of Routing Channel Lines E0, E2, E4, E8 $\rightarrow$ RC2 Line in the signal bus through selection bits $S_{25,6} S_{26,6}$

\[
S_{25,6} S_{26,6} \xrightarrow{Selects} \xrightarrow{Input} RC#
\]

\[
\begin{align*}
0 & \quad 0 \quad \rightarrow \quad E0 \\
0 & \quad 1 \quad \rightarrow \quad E2 \\
1 & \quad 0 \quad \rightarrow \quad E4 \\
1 & \quad 1 \quad \rightarrow \quad E8
\end{align*}
\]

C) Two Analog Multiplexers
• One of Analog Routing Channel Lines AR0, AR1 → mtb2 Line in the signal bus through selection bit $S_{24,6}$ and Tri-state through $S_{23,6}$

\[
\begin{align*}
&S_{23,6} S_{24,6} \quad \text{Selects} \\
&\quad \text{Input} \quad \rightarrow \quad \text{RC#} \\
&0 \quad X \quad \rightarrow \quad Z \\
&1 \quad 0 \quad \rightarrow \quad \text{AR0} \\
&1 \quad 1 \quad \rightarrow \quad \text{AR1}
\end{align*}
\]

• One of signal bus lines mtb1, mtb2 → Analog Routing Channel Lines AR1 through selection bit $S_{21,6}$ and Tri-state through $S_{22,6}$

\[
\begin{align*}
&S_{22,6} S_{21,6} \quad \text{Selects} \\
&\quad \text{Input} \quad \rightarrow \quad \text{mtb#} \\
&0 \quad X \quad \rightarrow \quad Z \\
&1 \quad 0 \quad \rightarrow \quad \text{mtb1} \\
&1 \quad 1 \quad \rightarrow \quad \text{mtb2}
\end{align*}
\]

8. For Connection Block WEST

A) PLB Output De-multiplexer

• PLB3 Output → One of Routing Channel Lines S6, S7, S8, S9 using selection bits $S_{17,0} S_{18,0}$ and Tri-stated through $S_{16,0}$

\[
\begin{align*}
&S_{16,0} S_{17,0} S_{18,0} \quad \text{Selects} \\
&\quad \text{Output} \quad \rightarrow \quad \text{RC#} \\
&0 \quad X \quad X \quad \rightarrow \quad Z \\
&1 \quad 0 \quad 0 \quad \rightarrow \quad \text{S6} \\
&1 \quad 0 \quad 1 \quad \rightarrow \quad \text{S7} \\
&1 \quad 1 \quad 0 \quad \rightarrow \quad \text{S8} \\
&1 \quad 1 \quad 1 \quad \rightarrow \quad \text{S9}
\end{align*}
\]

B) Two Digital Multiplexers

• One of Routing Channel Lines S1, S3, S5, S7 → RC3 Line in the signal bus through selection bits $S_{22,0} S_{21,0}$

\[
\begin{align*}
&S_{22,0} S_{21,0} \quad \text{Selects} \\
&\quad \text{Input} \quad \rightarrow \quad \text{RC#} \\
&0 \quad 0 \quad \rightarrow \quad \text{S1} \\
&0 \quad 1 \quad \rightarrow \quad \text{S3} \\
&1 \quad 0 \quad \rightarrow \quad \text{S5} \\
&1 \quad 1 \quad \rightarrow \quad \text{S7}
\end{align*}
\]

• One of Routing Channel Lines S2, S4, S6, S8 → RC7 Line in the signal bus through selection bits $S_{15,0} S_{14,0}$
C) Digital Multiplexer and De-multiplexers

- One of Routing Channel Lines S3, S4 → mtb4 Line in the signal bus through selection bit S_{20,0} and Tri-state through S_{19,0}

\[
\begin{array}{c|c|c}
S_{15,0} & S_{14,0} & \text{RC#} \\
\hline
0 & 0 & \rightarrow \text{S2} \\
0 & 1 & \rightarrow \text{S4} \\
1 & 0 & \rightarrow \text{S6} \\
1 & 1 & \rightarrow \text{S8}
\end{array}
\]

- Signal bus line mtb3 → S5, S6 through selection bit S_{13,0} and Tri-state through S_{12,0}

\[
\begin{array}{c|c|c}
S_{12,6} & S_{11,6} & \text{RC#} \\
\hline
0 & X & \rightarrow \text{Z} \\
1 & 0 & \rightarrow \text{S3} \\
1 & 1 & \rightarrow \text{S4}
\end{array}
\]

\[
\begin{array}{c|c|c}
S_{12,0} & S_{13,0} & \text{RC#} \\
\hline
0 & X & \rightarrow \text{Z} \\
1 & 0 & \rightarrow \text{S5} \\
1 & 1 & \rightarrow \text{S6}
\end{array}
\]
APPENDIX-C

MTLC PROGRAMMING CYCLE

The MTLC is programmed through Row and Column shift-registers using the serial-parallel methodology. The Column shift-register is connected to the enable lines which activate one column of SRAMs at a time to be loaded with data provided by the Row shift-registers. The following waveform describes the programming sequence.

MTLC Programming Cycle

This figure shows that one clock cycle is used to shift in the enable bit (Col_Data) in the Column shift-register and then shifted subsequently after each 32 clock cycles 6 more times within which 32 data bits (Row_Data) are fed through the Row shift-register to the SRAMs. The Reset signal clears the Row and Column shift-registers to avoid corruption of SRAM data when no programming is happening. The clock speed used to program the 7 X 32 array of SRAMs in the MTLC is 33.33MHz with 33.33% duty cycle.