Design and demonstration of a multi-technology FPGA for photonic information processing

Prosenjit Mal, Chris Hawk, Kavita Toshniwal and Fred R. Beyette Jr.¹
Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati PO Box 210030 Cincinnati, OH, 45221-0030, USA

ABSTRACT

We present here a novel architecture for of a multi-technology field programmable gate array (MT-FPGA). Implemented with a conventional CMOS VLSI technology the architecture is suitable for prototyping photonic information processing systems. We report here that this new FPGA architecture will enable the design of reconfigurable systems that incorporate technologies outside the traditional electronic domain.

Keywords: Optoelectronics, Optical data processing

1. INTRODUCTION

Electronic Field Programmable Gate Array (FPGA) consists of prefabricated arrays of generically programmable logic cells and interconnection wires that can be configured by the user to implement digital logic circuits. FPGA’s can be programmed/configured to realize arbitrary digital logic level designs. Since their introduction in 1985-86 [1], Field Programmable Gate Array’s (FPGA) have been developed into a major device technology that is suitable for applications requiring programmable/reconfigurable digital logic. Many different architecture and programming technologies have evolved to provide better designs that make FPGA technologies economically viable and an attractive alternative to Application Specific Integrated Circuit (ASICs) [2,3,4]. State-of-the-Art FPGA’s have superior logic density, chip cost and performance specifications when compared to low-end microprocessor. With more than a million programmable gates per chip, current FPGA’s can be used to implement digital systems operating at frequencies greater than 200 MHz. In many cases, it is possible to implement an entire system using a single FPGA. This is very economical for specialized applications that do not require the performance of custom hardware. As FPGA is programmable at the hardware level, it exploits the advantages of both general-purpose processors and specialized circuits i.e. ASIC. Many FPGA’s can be reprogrammed in-circuit allowing applications that exploit re-configurability and evolvable hardware. While a wide variety of multi-technology devices ranging from photonic information processing devices [5, 6, and 7], MEMS devices, telecommunication and digital data processing systems to biological sensors have been implemented, research in this area has for the most part been limited to systems built with application-specific devices. While these designs are well optimized for a specific application, they do not provide the flexibility associated with generically reconfigurable/programmable hardware. As with any Application Specific Integrated Circuit (ASIC), this approach is very costly, and the turn-around time between design iterations may be several months. This approach is not attractive for any multi-technology test-bed systems where the system designer depends on a rapid prototyping/experimentation environment that allows for optimization of processing algorithms and system architecture. The difficulty of using custom designed multi-technology VLSI components is overcome in the proposed research with the introduction of Multi-technology Field Programmable Gate Array (MT-FPGA) with innovative system architecture. Building on the programmability of FPGA technology, the proposed research will extend the FPGA capabilities into multi-technology realm by incorporating different multi-technology blocks in new MT-FPGA architecture. Over the past decade there has been a tremendous need for incorporation of non-traditional multi-technologies (ex. photonics devices [8, 9], analog devices, MEMS devices and microwave components) into traditional CMOS VLSI systems. The integration of these technologies requires a new kind of FPGA that merges conventional FPGA technology with mixed signal and other multi-technology devices. To demonstrate the feasibility of Multi-technology Field Programmable Gate Array (MT-FPGA), a new FPGA architecture is proposed here. This MT-FPGA can exploit the capabilities of both traditional FPGA’s and the ability to work outside the electronic domains with multi-technology device capabilities. This

¹Correspondence: Email: beyette@ececs.uc.edu; Telephone: 1 513 556 4588; Fax: 1 513 556 7326; http://www.ececs.uc.edu/~beyette.
new class of field programmable device will extend the flexibility, rapid prototyping and reusability benefits associated with conventional electronic into the multi-technology domain enabling the development of a whole new class of integrated and embedded systems. Based on very-large-scale integration (VLSI) device technology, the proposed MT-FPGA will monolithically integrate all necessary components in a single complementary metal-oxide semiconductor (CMOS) chip. The FPGA architecture presented here allows for the incorporation of a variety of multi-technology blocks like optical blocks, Analog-Mixed-Signal blocks, RF blocks, SRAM blocks, MEMS blocks, chemical/biological sensor blocks, etc.. A multi-technology system designer can readily implement any prototype system with (1) logic parts in programmable section of MT-FPGA (2) Multi-technology parts by incorporating different multi-technology blocks in the dedicated location within the architecture from standard library. Since the implementation resembles a conventional FPGA; the idea is robust and scalable. Further, existing high-level CAD tools can be readily modified to provide a CAD environment that is comparable to existing technology. In most of the cases, optimization of the prototype design requires change in logic circuitry and shuffling the distribution of multi-technology blocks. The proposed MT-FPGA architecture strongly supports above requirements and facilitate a generic architecture for rapid prototyping/experimentation. Note that features like programmability and fast time-to-market market made conventional FPGA to grow so rapidly in digital circuit world while the proposed MT-FPGA manifests the same advantages to become potential leader in “Programmable multi-technology design space (PMTDS)”. The new MT-FPGA can exploit the capabilities of both traditional FPGA’s and the ability to work outside the electronic domains with multi-technology device capabilities. This new class of field programmable device will extend the flexibility, rapid prototyping and reusability benefits associated with conventional electronic into the multi-technology domain enabling the development of a whole new class of integrated and embedded systems. Based on very-large-scale integration (VLSI) device technology, the proposed MT-FPGA will monolithically integrate all necessary components in a single complementary metal-oxide semiconductor (CMOS) chip. To highlight the connectivity and communication between the Multi-technology environments and programmable logic component of the proposed MT-FPGA, a user threshold programmable optical sensor block with photonic information processing capabilities will be developed.

2. FEATURES OF THE NEW MT-FPGA

The design and implementation of a new multi-technology FPGA architecture is proposed here. In this novel architecture, a modular approach is proposed where each multi-technology block will be surrounded by electronic programmable logic blocks (PLBs) with an internal bus for communication internally within the cluster and externally with the routing channels. Some preliminary but encouraging results towards implementation of the proposed MTB are also presented.
Figure 1 illustrates the high level architectural definition of the proposed multi-technology field programmable gate array (MT-FPGA). The array of blocks resembles the symmetrical array used in the layout of electronic FPGA’s where each logic block is replaced by a cluster of blocks consisting of a multi-technology block (MTB) and four Programmable logic blocks (PLBs). This cluster is called a multi-technology logic cluster (MTLC). The regular array of MTLC is interspersed with horizontal and vertical routing channels. The routing channels are segmented and typically form a hierarchy of different length classes. Different length segments are connected in programmable fashion through the Switch blocks (SB). The inputs and outputs of different MTLCs can be connected to the routing channels in configurable ways using the connection blocks (CB). The floor plan of a MTLC (Figure 2) shows that each MTLC is made up of four PLBs surrounding a single Multi-technology block with an internal bus for communication internally within the cluster and externally with the routing channels. The MTB may contain one to several multi-technology devices (ex. Photonics, RF components, SRAM or DRAM based storage devices, MEMS, Chemical/biological sensors etc.).

Figure 3 shows the logic diagram for one of the programmable logic blocks (PLB’s) used in the MTLC. Each PLB contains a 4-input look-up table (4-LUT) and a user flip-flop. The user can program the look-up table by placing the truth table of any four input function into the 16-addressable locations of the 4-LUT. The user flop along with the look-up table can be used for implementation of sequential circuit. The user flop is reset asynchronously with user reset and driven by a two-phase system clock (CLKA & CLKB). The output MUX can be programmed to select either the output from the flop or the output directly from the 4-LUT. Depending on the output selection, the PLB output will be written on the designated internal bus line through a buffer. The scan mux in PLB allows the user flops to form a scan chain that will be used for testing the functionality and performance of the MT-FPGA hardware. Like conventional FPGAs, the trade-off between area and density is an important issue in this design. Chow et al [10, 11] have addressed this problem in conventional FPGA’s and shown that in conventional FPGA’s, the use of hard-wired logic blocks (HLB) using an L3-4.2 tree topology significantly improves speed and area utilization of a conventional FPGA architecture. Thus, the prudent replacement of slow programmable links with hard-wired connections in the critical path between PLBs can significantly reduce routing delay at the cost of some connection flexibility. The architecture presented here builds on this idea by incorporating the MTB into the L3-4.2 topology identified by Chow et al as the optimal configuration. Figure 4 illustrates how this topology is applied to the MTLC. As indicated in Figure 4 the four PLB’s communicate with the multi-technology block (and each other) using a 16-bit internal bus. MTB’s are hybrid technology blocks that may be required to exchange raw and/or processed data with PLB’s using a set of programmable connections.

As shown in Figure 5, the 16-bit internal bus is subdivided as follows:
- 4-bits are used for output signal distribution from PLB’s.
- 4-bits are designated for two-way communications with MTB.
- 4-bits are dedicated for signals coming from other MTLC via the routing channels and controlled by programmable switch blocks.
4-bits are used for clock distribution, reset signal & configuration enable.

Figure 5 shows the connectivity within a multi-technology logic cluster. As stated above, bit lines (4..1) receive inputs from routing channels (RC). These inputs are generated from other MTLCs and routed through switch blocks and connection blocks before reaching the location with this MTLC. Bit lines (5..8) are driven by outputs from four PLBs. Four outputs from multi-technology block drive bit lines 9..12. The outputs from the multi-technology block are not limited to digital signals. A two-bit analog internal bus (not shown) is also included to accommodate analog outputs from the MTB. Bit lines from (13..16) are used for distribution of global signals (i.e. CLKA & CLKB, reset and configuration Enable). These signals are not shown in the diagram. There are four PLBs in a cluster. Each PLB has a four input look-up table (4-LUT) with each input of the 4-LUT connected to an 8:1 MUX (if it is not hardwired connected). PLB1 and PLB2 do not have any hardwired-connected inputs. A proper distribution of connectivity between the LUT inputs and the internal bus can significantly reduce the size of the MUX while imposing a negligible cost in terms of interconnections flexibility. A smaller interconnection network is always preferable from the standpoint of area and speed.

Table 1: Interconnections assigned to each of the PLB inputs

<table>
<thead>
<tr>
<th>INPUT1</th>
<th>INPUT2</th>
<th>INPUT3</th>
<th>INPUT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLB1 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB1 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB1 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB1 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
</tr>
<tr>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
</tr>
<tr>
<td>2—FROM RC (1,2)</td>
<td>2—FROM RC (2,3)</td>
<td>2—FROM RC (3,4)</td>
<td>2—FROM RC (4,1)</td>
</tr>
<tr>
<td>2—FROM MTB (9,10)</td>
<td>2—FROM MTB (10,11)</td>
<td>2—FROM MTB (11,12)</td>
<td>2—FROM MTB (12,9)</td>
</tr>
<tr>
<td>PLB2 SAME AS ABOVE</td>
<td>PLB2 SAME AS ABOVE</td>
<td>PLB2 SAME AS ABOVE</td>
<td>PLB2 SAME AS ABOVE</td>
</tr>
<tr>
<td>PLB3 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB3 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB3 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB3 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
</tr>
<tr>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
</tr>
<tr>
<td>2—FROM RC (1,2)</td>
<td>2—FROM RC (2,3)</td>
<td>2—FROM RC (3,4)</td>
<td>2—FROM RC (4,1)</td>
</tr>
<tr>
<td>2—FROM MTB (9,10)</td>
<td>2—FROM MTB (10,11)</td>
<td>2—FROM MTB (11,12)</td>
<td>2—FROM MTB (12,9)</td>
</tr>
<tr>
<td>PLB4 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB4 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB4 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
<td>PLB4 CONNECTED TO 8:1 MUX OUT OF 8 (REFER FIG.5)</td>
</tr>
<tr>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
<td>4—OUTPUTS OF 4 PLBS</td>
</tr>
<tr>
<td>2—FROM RC (1,2)</td>
<td>2—FROM RC (2,3)</td>
<td>2—FROM RC (3,4)</td>
<td>2—FROM RC (4,1)</td>
</tr>
<tr>
<td>2—FROM MTB (9,10)</td>
<td>2—FROM MTB (10,11)</td>
<td>2—FROM MTB (11,12)</td>
<td>2—FROM MTB (12,9)</td>
</tr>
<tr>
<td>HARD-WIRED TO OUTPUT OF PLB1</td>
<td>HARD-WIRED TO OUTPUT OF PLB2</td>
<td>HARD-WIRED TO OUTPUT OF PLB3</td>
<td>HARD-WIRED TO OUTPUT OF PLB4</td>
</tr>
</tbody>
</table>

In contrast to PLBs 1 and 2, the inputs to PLBs 3 and PLB4 are not completely programmable connections. The output from PLB1 is hardwired to one of the inputs to PLB3 through the internal bus. Similarly, the outputs from PLBs 2 and 3 are hardwired to two of PLB4’s inputs. This hardwire configuration allows the PLBs to conform to a L3-4.2 topology. This type of architecture reduces the size/complexity of the interconnection matrix beyond what is required for complete logic utilization. This reduction in interconnection matrix is directly translated in to reduction of area and an increase in FPGA speed. Table 1 describes the detailed interconnections among inputs of the 4-LUT’s and the internal bus. PLB1
and PLB2 have identical interconnection matrix where as PLB3 and PLB4 are different due to their hardwired connections.

In the MT-FPGA chip design, two different types of programmable switch blocks are required. The LUTs and output drivers of the CLBs produce digital signals that need to be directed around the chip to some desired destination. Additionally, the MTB blocks will also require analog input and output connections that also must be programmable. Transmission Gate switches provide a minimal area solution that is applicable to both domains. In total, the routing channels of the MT-FPGA will each contain six digital and two analog routing tracks. For analog switches, it is most important that the switches do a good job of preserving the signal integrity of the analog data. Ideally, we would like the switch to behave like a wire while the switch is in the on state, and like an open circuit while in the off state. Additionally, to allow a reasonable number of MTLC units to be placed on the FPGA, it is critical for our switches to consume as little area as possible. Digital switches do not have the same signal fidelity requirements due to the built in noise margins that are inherent to all digital circuits. Instead, for the digital circuits we will focus on propagation delay and on correct operation even on high fan out nets.

For digital systems there are a number of well known robust solutions that provide good performance and are scalable over large designs with high fan-out. Unfortunately, these solutions often require additional area overhead due to the addition of buffers and the requirement to have two control bits per switch. Because of the limited scale (i.e. only a 3 x 3 grid of MTLCs), and because of the possibility of reusing the same switch structure for both the analog and digital switches, we have decided to choose a transmission gate design for the basic switch component in both the digital and the analog switch block. The transmission gate, since it is bidirectional, only requires one bit of control. Buffered switches by contrast usually are not bidirectional and require two control bits. The control bits in this case are stored in a SRAM cell whose size dominates the area of the switch structure.

The digital switch does not necessarily require the TG for proper operation. A pass transistor could suffice. However, the addition of the PMOS transistor helps decrease the overall switch resistance and allows full VDD to ground operation of the digital signals. Both of these characteristics are helpful when the individual switches are not being driven by their own buffers. The switch design and routing channel configuration are shown in Figure 6 a & b. To further assist the performance of the digital routing tracks, level restoring circuits have been added for the purpose of compensating for the voltage drop across switches and helping to improve the overall delay of transmission along the digital routing tracks. The level restoring circuits consist of a sensing transistor that assists in pulling the routing channels up or down when the intermediate nodes reach a certain threshold. Simulations on cadence have shown this approach to be effective to some degree at reducing the overall delay of certain test cases with high fanout and/or high number of switches in series. Another critical aspect of the switch block structure is the flexibility available for each wire entering the switch block. At one extreme, one could design a switch block that would allow each routing channel at an intersection to route to every other routing channel at the same intersection. Studies have shown, however, that such high degree of connectivity is not needed to get good routability out of a FPGA design. As long as there is sufficient connection block flexibility, the switch block flexibility can be relatively low. For our chip, we have chosen to have a flexibility of three for each routing channel. This means that each routing track can choose among three different destination routing channels. Furthermore, the switch blocks in our design are disjoint. This means that each routing track within a routing channel can only connect to other routing tracks that are in the same position within the channel as the original. Input/Output in the MT_FPGA are handled via the I/O blocks (Figure 7). Due to pin limitation, all the tracks from the routing channel cannot be connected to the I/O pads. Hence to meet the pad requirement, I/O blocks are designed such that the routing channels terminate at the I/O block. The I/O Block is designed to handle two digital lines and one analog track at any given time providing an option of input, output or bi-directional

![Figure 6 a: The circuit for a single switch. 6 b: Cluster of switches at an intersection.](image-url)
Increasing the number of digital lines that can be handled at a given time can improve the flexibility, but this design limits the number to two due to the area constraint.

In the circuit the Analog lines being sensitive require the need to be shielded (Figure 8). For this a metal layer is placed over the tracks and this metal layer is connected to the ground. Also the tracks are surrounded on either side by n-substrate contact which is connected to the highest potential. Multiplexers are used for selection of the lines for the output mode. The same multiplexer acts as a de-multiplexer in input mode. The select line used for defining the direction of the pad is used to determine whether a multiplexer (mux) or a de-multiplexer (demux) is in effect. Programming of the mux/demux is done using SRAM programming bits. The Simulation results shown in Figure 9 are for a case where the signal is given to one of the tracks between D0-D3 and the output is obtained at D0-3. This is referred to as output mode while the input given to the signal D2-D5 is obtained at one of the signals among D2-D5 and this is referred to as the input mode. The selection of the bit between D0-D3 and D2-D5 is done using the SRAM bits referred in the figure as sel signals. And the enable bit high or low decides whether it is output mode or input mode. So for this case the sel_1 is high and sel_2 is low. For analog signals as buffers are not used no SRAM bit is required for the direction. The only control bit required is for the selection of the analog line and is referred and sel_3 in the Figure 9. To highlight the connectivity and communication between the MTB and PLB cluster in MTLC, a user threshold programmable photoreceiver multi-technology block similar to one described in [12] is used to receive optical input data. One can use this kind of architecture for smart photonic information processing.

3. Preliminary Results from the Proposed MT-FPGA Hardware Test Chip:

The implementation of the entire MT-FPGA architecture has been planned in two phases. In the first phase [13], a test chip has been designed for the purpose of evaluating the MTLC functionality. The test chip has been fabricated using a 1.5 μm AMI CMOS foundry service. To characterize the MTLC chip, a simple but stepwise testing strategy was followed as described below. (All the time response traces were collected using Agilent 54616B oscilloscope and other logic traces are from HP1662A logic analyzer. To generate test inputs and sequence of program bits, a HP pattern generator was used). Test results shows that chip is completely functional and produce results as expected. Figure 10 is a detail ‘PIN-OUT’ and ‘PAD_
SIGNAL_NAME’ diagram. It also shows how PLB’s are oriented within the chip and their associated soft-links and firm links.

1) **Testing of the Configuration chain**: There are two configuration chains in the chip. One configuration chain is used to hold program bits when the other keeps the row enable bits. Program bits are stored in SRAM cells that are arranged in 8 rows and 4 columns. Reset the chip for a full cycle. This will make all general-purpose flops to zero. Assert the Enable configuration (‘ENCFIN_H’) bit to logic high. Load configuration chain by electronically loading a sequence of alternating 1’s and 0’s. The sequence can be observed at the output of some shift-register. This confirms that configuration chain is functioning as expected. During programming of the chip, one row is selected in each turn and receives program bits. Then it goes to next row and so on until the programming of all SRAM cells is over. (Due to lack of enough routing space in the present chip, PLB 3 & 4 are programmed separately from PLB 1 & 2. PLB 1 and PLB 2 are programmed directly from chip pins using program bits and select lines. This conserves some silicon real estate as amount of routing is reduced.)

2) **Testing of the Scan chain**: Initialize the chip. Enable the ‘test mode’ of the chip. Reset the chip for a full cycle. This will make all general-purpose flops to zero. Load a bit pattern electronically into the chip through input pin (SCAN_IN_H). The pattern is observable at the output of the scan chain (SCAN_OUT_H). Refer Figure 11. Steps 1 & 2 confirm that the device is programmable and controllable. It is ready for characterization of MTLC.

3) **Configuration of a PLB as an inverter**: To show the functionality of the PLB, the example of a simple digital gate (i.e. an inverter) is demonstrated here. Initialize the chip. Enable the ‘configuration mode’ of the chip. Reset the chip for a full cycle. Choose any Programmable logic block ‘PLB 2’. Configure it to an independent inverter cell and program it in such a way that ‘PLB2’ be connected directly to input pin (SOUTH_IN_H) and output pin (PLB2_OUT_H).

---

Figure 10: Pin-out diagram of a MTLC chip

Figure 11: Logic Analyzer traces showing one scan chain testing of MTLC
4) **Characterization of a PLB when configured as an inverter:** After configuring the ‘PLB2’ as an inverter, it can be tested for its functionality next. Feed a square pulse train (0-4V) at the input (SOUTH_IN_H) of the cell and measure the time response of the cell at the output (PLB2_OUT_H). Figure 12 shows propagation delay \( t_P \) between input and output waveforms during low-to-high (\( t_{pLH} \)) and high-to-low (\( t_{pHL} \)) at 50% transition points of the programmable logic block ‘PLB2’. The overall propagation delay (\( t_P \)) is given by the expression:

\[
t_P = \frac{(t_{pLH} + t_{pHL})}{2} = \frac{(26+25)}{2} \text{ns} = 25.5 \text{ns}.
\]

The same timing traces can be used to find the rise and fall time of the inverter. Rise and fall times are generally defined as time between 10% - 90% change of the output. Figure 13 exemplifies the input and output characteristics of the test inverter. Top trace is a square pulse input fed to the inverter and output is a reverse square pulse as expected.

5) **Characterization of the MTLC:** As mentioned earlier, hardwire connected PLB’s are used in MTLC to make a faster cluster of logic blocks and multi-technology block within a reasonable area cost. As shown in Figure 14, PLB1, PLB2 and PLB4 are used as a cluster. Each PLB is programmed as an inverter as described above. Solid lines indicate hard-wire connection between output of a PLB and one of the four inputs of other PLB’s. The dash lines show soft-links from the output of a PLB to the input of the other PLB’s through 8:1 input MUX. For example, each input of PLB1 & PLB2 has programmable connections (i.e. soft-links) via 8:1 MUXes which choose 1 out of 16 input connections. On the other hand, PLB4 has two programmable connections or soft-links via 8:1 MUXes and two hard-wire connected inputs from the outputs of PLB2 & PLB3. In the Figure 14, one input of PLB2 is hard-wired from the output of PLB1. The hard-wire connection or firm link used in place of soft-links to remove the bulky 8:1 MUX’s in the critical path of the MTLC and thus significantly improve the speed of the cluster with a considerable amount of chip area conservation. To understand the dynamic behavior of the MTLC and the speed advantage of firm-link over soft-link, two test circuits were implemented Using a) Firm links and b) Combination of firm links and soft links. Firm link offers a direct connection...
between PLB’s at the cost of flexibility, a benefit associated with programmable link or soft link.

Figure 14: Implementation of a ring oscillator with firm links

6) **Using FIRM LINKS**: Ring oscillator can be used as a de-facto standard circuit for delay measurement [14] in the present work. The period ‘T’ of the oscillation depends on the propagation delay of signal passing through the complete ring. A ring oscillator that consists of three inverters and connected in a circular chain was programmed on three PLB’s in the MTLC with two firm links and one soft-link. PLB1, PLB2 & PLB4 were programmed for three inverters and all but one hard-wired connection among them. The only soft-link (only option) was chosen between PLB1 and PLB4 to complete the circular ring-oscillator. See Figure 14. Output is available at either of these pins (PLB1_OUT_H, PLB2_OUT_H, and PLB4_OUT_H). Figure 15 shows the frequency of oscillation of the ring oscillator is 12.12 MHz. This is about 3 times faster than our earlier similar implementation. This speed-up of the circuit attributes to ‘few firm links in place of all soft-link approach’ and the use of ‘innovative fast Look-up table circuit’ replacing slow ones [15].

7) **Using Combination of SOFT& FIRM LINKS**: The Second ring oscillator circuit was programmed on the same set of logic blocks PLB1, PLB2 & PLB4. This time the firm-link between PLB2 and PLB4 were replaced with a soft-link keeping other links same as before (Figure 16). This gives rise to a ring oscillator that consists of three inverters, connected in a circular chain and programmed on three PLB’s in the MTLC with two soft-links and one firm-link. Output is available at either of these pins (PLB1_OUT_H, PLB2_OUT_H, and PLB4_OUT_H). Figure 17 shows the frequency of oscillation of the ring oscillator is 10.75 MHz. This implies that the time penalty for replacing a firm-link with a soft-link is about (73ns-62ns=) 11ns using the AM1 1.5µm CMOS process. This leads us to the conclusion that uses of chosen firm-links in the critical path in the MTLC can speed up the whole cluster with negligible loss in flexibility of connections.

Figure 15: Output from a ring oscillator with firm links

Figure 16: Implementation of a ring oscillator with combination firm link and soft links.
Implementation of a test circuit (a finite state machine): So far, we have tested only combinational logic implementations with MTLC. As one can see from figure that at the output of PLB’s a user flip-flop is used to latch the output from the Look-up Table synchronized with a two phase (non-overlapping) external clock “CLKA & CLKB”. User can program the output MUX to select either normal or latched output. Since the clocks are also included as inputs to the PLB’s the user may alternatively use the MTLC to create signals that are derived from the clocks. User can also implement a sequential circuits with MTLC where unlike combinational circuit, outputs depends not only on its current inputs, but also on the past sequence of inputs. To demonstrate a ‘Clocked Synchronous Finite State machine (CS-FSM)’ following example is chosen.

Implementation of a 1-bit slice pipelined full-adder: Two n-bit binary numbers, synchronized with clocks are given as inputs to the circuit. The circuit can add two binary bits with carry from previous addition with each clock cycle. So, total ‘n’ clock cycles are required to add two n-bit numbers. For the pipeline operations, the ‘SUM’ & ‘CARRY’ outputs are buffered with clocked flip-flops. PLB2 was used to generate carry bit and PLB3 was used to make sum bit.

**STEP1:** - Initialize the chip. Enable the ‘configuration mode’ of the chip. Reset the chip for a full cycle. Choose the Programmable logic block ‘PLB 2’ & ‘PLB3’. Configure PLB2 to a ‘SUM maker’ and PLB3 as ‘carry generator’ by loading appropriate truth tables in respective cells. Program them to connect directly to I/O pins. Connection between the output of PLB2 and one of inputs of PLB3 was chosen as ‘SOFT-LINK’. Input pins are SOUTH_IN_H (28) & EAST_IN_H (17) and output pins are PLB2_OUT_H (37) & PLB3_OUT_H (38).
STEP 2: Next step after programming the chip is to test it with test patterns. Any two n-bit binary numbers ‘S_DAT1’ and ‘E_DAT2’ (01100110 & 01010101 respectively) are clocked to the inputs (SOUTH_IN_H & EAST_IN_H) of the PLB’S. See Figure 18. Note that in the figure leftmost bits are LSB’s as they came before the MSB’s in sequence. Result sum bits and carry bits are generated after one clock cycle of the input bits. Sum and carry generators produce present sum and carry bits based on present inputs as well as one cycle past carry bit. Result sum and carry bits as shown in the figure are ‘10111011’ and ‘0000100’ respectively. This completes the evaluation of the ‘Clocked Synchronous Finite State machine (CS-FSM)’. The output is stored in buffers to facilitate pipeline operation.

4. CURRENT STATUS

Phase 1: Implementation of the single chip MTLC

Phase 1 work, involving designing and testing of chip (fabricated through MOSIS [16] CMOS foundry services), has been completed. The chip is fully functional and produces expected results. Simulations results compare well with the test results. The maximum power consumed by the chip is 5mA X 5V=25mW.

Phase 2: Implementation of the single chip MT-FPGA

Based on the knowledge gained from previous research work on MTLC design, phase 2 work has focused on the circuit and layout design of the prototype MT_FPGA chip. Since the implementation of the prototype MT_FPGA chip has used TSMC 0.35 µm process while our previous experiences are based on AMI 1.5 µm process, there are challenges introduced by the sub-micron CMOS VLSI design. The challenges faced in this phase are:

... Design of PLB’s, switch blocks, connection blocks, routing structures and multi-technology blocks (i.e. photodetectors /receivers with sufficient responsivities, frequency response and uniformity) which work reliably with low noise margins and minimum voltage drop across the routing channels and switch/connection blocks. As number MTLC are 9 in second phase of design, low power design is also major challenge in this phase.

... Minimization of the cross talk among analog parts in MTB and digital parts of the circuit is required. Increasing number of MTLC and decreasing the feature size will decrease the MTLC-to-MTLC spacing. Therefore, it is expected that the cross talk between MTLC’s will increase.

... Minimizing the substrate cross talk in the sub-micron mixed-signal CMOS VLSI design in MTB’s.

... Low power design methodologies are employed to reduce the power consumption of the MT_FPGA chip.

The layout design of the MT_FPGA chip is fabricated in this phase using TSMC 0.35 µm process through MOSIS. Currently, the fabricated MT-FPGA chip is being tested comprehensively. The testing results will be used to address the issues mentioned earlier. Tasks in this phase will include:

... Characterization and performance evaluation of the PLB’s, switch/connection blocks, routing structure and MTB’s (i.e.photodetectors/receivers, including optical power requirements, frequency response of the receivers).

... Measurement of electrical specifications of the MT-FPGA.

... Working demonstration of the functionality of the entire MT-FPGA chip.

... Comparison and analysis of the performances of MT-FPGA chip using different technologies to address different issues.
REFERENCES