INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

# Instant Access Memory Design based on an FPGA

L.ANUSHA[1], S.KONDAIAH[2]
[1]PG Student, Dept of ECE, SITS, Kadapa, AP, India.
[2]Assistant Professor, Dept of ECE, SITS, Kadapa, AP, India.

***Abstract***— Since the days when they were only a small bunch of logic gates that could design any Boolean Function, Field Programmable Gated Array (FPGA) have come a long way and evolved into many complex chips. The FPGAs have now become a very essential part of the semiconductor and Very Large Scale Integration industry. This paper presents a memory design, based on FPGA which can locate the address required by CPU almost instantly. This memory design works efficiently in Memory intensive jobs also. In this paper, the same hardware architecture has been designed and synthesized with different memory sizes and then they are compared by analyzing the number of lookup tables (LUTs) they use, their on-chip power and the delays associated with them.

## 1. INTRODUCTION

With the ever-increasing speed at which networks are running and the need for instantaneous access to data, it seems that there can never be enough memory. In most FPGA devices, the amount of high-speed SRAM like memory is now in the 300 to 400Mb range. This is rarely adequate for buffering, tables, statistics, etc. The predominant approach by the FPGA vendors has been to attach significant resources of High Bandwidth (HBM). Although this addresses some memory issues very well, it struggles to support high-speed, random access requirements. To date, the approach has been to use QDR SRAM which provides high-speed memory access to an FPGA through a bus that is 100+ pins wide. Many customers are hitting limitations with the use of QDR. They need more high-speed, random access memory and the QDR is too shallow, uses too many pins, and is difficult to expand.

State-of-the-art commercial FPGAs offer several hundred thousand logic cells along with specialised function units connected via a configurable network. In order to configure a circuit the user needs to load configuration data into the SRAM of the device. This data is generated by CAD tools and is most often externally loaded onto the device via a configuration port. The FPGA's reconfiguration involves updating the entire, or a part of, the configuration SRAM. As reconfiguration time is roughly proportional to the amount of configuration data to be loaded onto an FPGA, reconfiguration delay can become critical for applications that demand circuit modification at run-time. Rapid partial reconfiguration is therefore desirable as devices continue to scale up in size. The amount of configuration data of an FPGA grows in proportion to the device size. For example, the configuration bit-stream size for a Virtex XCV1000 device is approximately 738KB. The latest Virtex-4 XCV4FX140, which is almost 6 times larger than an XCV1000 in terms of logic resources, has a configuration bit-stream size of 5.7MB. The configuration port for this device is 8-bits wide and can be clocked at 100MHz. Assuming that the data can be delivered at this rate, it will take 57 ms to load the entire configuration bit-stream. In embedded systems the FPGA is usually programmed from a relatively slow flash memory. The System Ace device from Xilinx offers data rates of up to 152MBits/sec. In this case, the time needed to completely reconfigure an XCV4FX140 will be about 3 seconds. When a circuit switches between several configurations then this time can significantly degrade the system performance. Moreover, large configuration sizes are also not desirable from an on-board storage perspective. A solution to this problem is partial reconfiguration, whereby the user loads a subset of configuration data. The configuration memory of a partially reconfigurable FPGA is internally organised into words just like a conventional RAM and all read/write transactions occur in multiples of these units. In Virtex devices the smallest unit of configuration is called a frame which spans the entire height of the device and configures a portion of a column of user resources. The (re)configuration time is proportional to the number of frames to be loaded.

## 2. METHODOLOGY

In this paper, a 2-Dimensional Memory Design using Row and Column Decoders is presented. This design does not make use of a clock which is why it executes the operation almost instantly. Fig. 4 shows the flowchart of the procedure to make such a 2-Dimensional Memory Design. The detailed procedure is explained below Create Respective Modules. The Modules of Memory Cell, Decoder and Tri-State Buffer are

ISSN: 2057-5688

made in Verilog. Instantiate Memory Cell Memory Cell is instantiated as many times as is required using a for-loop inside a generate statement. Verilog Generate statements are used to write synthesizable RTL. They can be used to create multiple instantiations or conditional instantiations of a module. Here, a generate statement is used to create multiple instantiations of Memory Cell and Tri-State Buffer.
Instantiate Tri-State Buffer Tri-State Buffers are used to drive the final output of the design. The outputs of all the individual Memory Cells drive the output of the design through Tri-State Buffers. A multi-input OR gate could also have been used in place of Tri-State Buffers. This OR gate will take all the outputs of individual Memory Cells and OR them, but that would increase the hardware requirements of the design. An OR gate would also produce erroneous outputs if the outputs of the inactive Memory Cells float.

**Field Programmable Gate Array (FPGA):**

- Gate array: a VLSI circuit with some pre-fabricated gates repeated thousands of times
- Designers have to provide the desired interconnection patterns to the manufacturer (factory)
- A field programmable gate array (FPGA) is a VLSI circuit that can be programmed in the user's location
- Easier to use and modify
- Getting popular for fast and reusable prototyping
- There are various implementations for FPGA

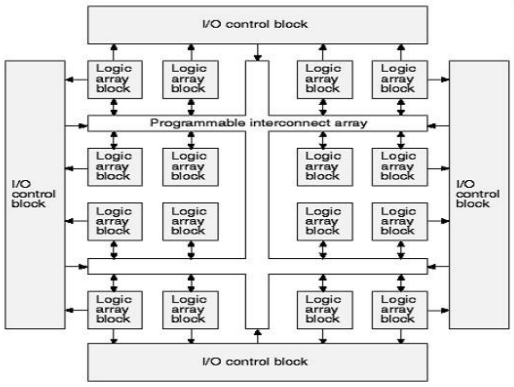
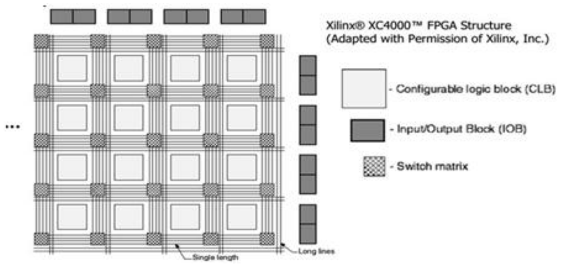


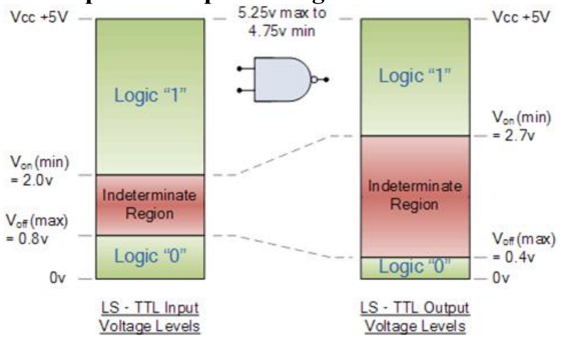
**Figure:** Field Programmable Gate Array (FPGA)

**FPGA structure (Altera)**


Xilinx® XC4000™ FPGA Structure
(Adapted with Permission of Xilinx, Inc.)
- Configurable logic block (CLB)
- Input/Output Block (IOB)
- Switch matrix
Single length
Long lines

**Digital Logic Circuits:-**
**TTL Circuit:-**
TTL logic IC's use NPN and PNP type Bipolar Junction Transistors while CMOS logic IC's use complementary MOSFET or JFET type Field Effect Transistors for both their input and output circuitry. As well as TTL and CMOS technology, simple Digital Logic Gates can also be made by connecting together diodes, transistors and resistors to produce RTL, Resistor-Transistor logic gates, DTL, Diode-Transistor logic gates or ECL, Emitter-Coupled logic gates but these are less common now compared to the popular CMOS family.

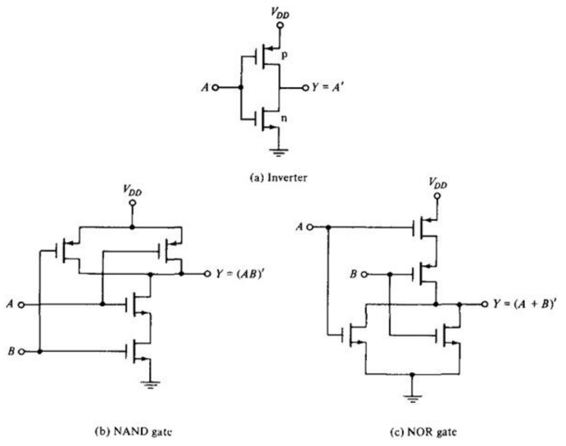
**TTL Input & Output Voltage Levels:-**



When using a standard +5 volt supply any TTL voltage input between 2.0 V and 5 V is considered to be a logic "1" or "HIGH" while any voltage input below 0.8 V is recognized as a logic "0" or "LOW". The voltage region in between these two voltage levels either as an input or as an output is called the Indeterminate Region and operating within this region may cause the logic gate to produce a false output. The CMOS 4000 logic family uses different levels of voltages compared to the TTL types as they are designed using field effect transistors, or FET's. In CMOS technology a logic "1" level operates between 3.0 volts and 18 volts and a logic "0" level is below 1.5 volts.

**COMPLEMENTARY MOS (CMOS):**
Complementary MOS circuits take advantage of the fact that both n-channel and p-channel devices can be fabricated on the same substrate. CMOS circuits consist of both types of MOS devices

interconnected to form logic functions. The basic circuit is the inverter, which consists of one p-channel transistor and one n -channel transistor.



**CMOS logic circuits**

Now consider the operation of the inverter. When the input is low, both gates arc at zero potential. The input is at -VDD relative to the source of the p-channel device and at 0 V relative to the source of the n-channel device. The result is that the p-channel device is turned on and the n-channel device is turned off. Under these conditions, there is a low-impedance path from VDD to the output and a very high-impedance path from output to ground. Therefore, the output voltage approaches the high level VDD under normal loading conditions. When theinput is high, both gates are at VDD and the situation is reversed: The p-channel device is off and the n-channel device is on. The result is that the output approaches the lowlevel of 0 V. CMOS

**TRANSMISSION GATE CIRCUITS:**

The transmission gate is essentially an electronic switch that is controlled by an input logic level. It is used for simplifying the construction of various digital components when fabricated with CMOS technology. It consists of one n-channel and one p-channel MOS transistor connected in parallel. The n- channel substrate is connected to ground and the p-channel substrate is connected to VDD. When the N gate is at VDD and the P gate is at ground, both transistors conduct and there is a closed path between input X and output Y. When the N gate is at ground and the P gate at VDD both transistors are off and there is an open circuit between X and Y.

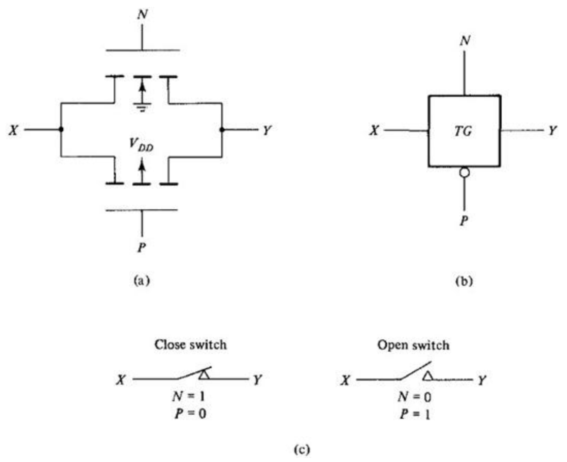
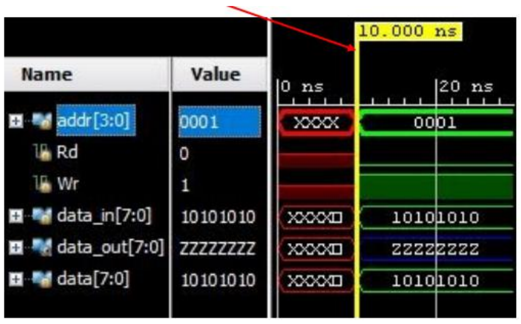


**Figure:** Transmission Gate (TG)

**3. SIMULATION RESULTS**

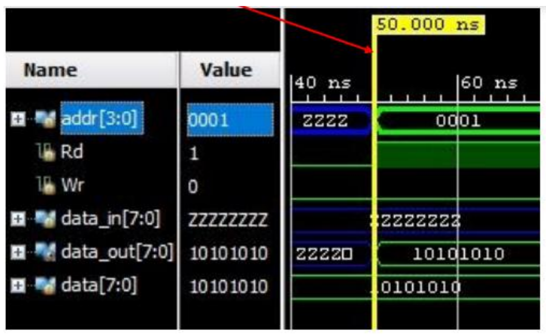
This section shows the simulation results for Read and Write operations. It also shows that both the Read and Write operations were performed instantly. The figure shows the Write Operation. At 10ns, the Wr signal went high and the adder signal was assigned the address of the Memory Cell that was to be written. At the same instant, the data register of this particular Memory Cell was updated with the value that was sent to the Memory via data in signal and the Write Operation was successfully executed



| Name | Value | 0 ns | 10.000 ns | 20 ns |
|---|---|---|---|---|
| addr[3:0] | 0001 | XXXX | 0001 | |
| Rd | 0 | | | |
| Wr | 1 | | | |
| data_in[7:0] | 10101010 | XXXX | 10101010 | |
| data_out[7:0] | ZZZZZZZZ | XXXX | ZZZZZZZZ | |
| data[7:0] | 10101010 | XXXX | 10101010 | |

Figure: Write operation

Fig. shows the Read Operation. At 50ns, the Rd signal went high and the adder signal was already assigned the address of the Memory Cell that was to be read. At the same instant, the data_out of the Memory was assigned the value stored in the register of this particular Memory Cell and the Read Operation was successfully executed.



| Name | Value |
|---|---|
| addr[3:0] | 0001 |
| Rd | 1 |
| Wr | 0 |
| data_in[7:0] | ZZZZZZZZ |
| data_out[7:0] | 10101010 |
| data[7:0] | 10101010 |

Figure: Read Operation

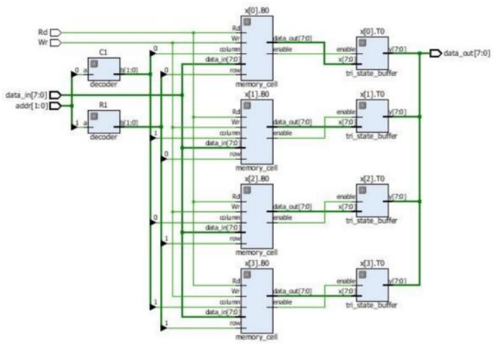


Figure: shows the schematic of a 4-byte memory

## CONCLUSION

Memory design based on FPGA which can locate the address required by CPU almost instantly is presented in this paper. This proposal of memory design works efficiently in memory-intensive jobs also. In this proposal, a hardware architecture has been designed and the same is synthesized with different memory sizes which are then compared by analysing the number of lookup tables (LUTs) used, on-chip power and the delays associated. This comparison is then reported for more clarity.

## REFERENCES

[1] S. Brown and J. Rose, “FPGA architectural research- Survey” in IEEE Design & Test of Computers, vol. 13, issue 4, 1996, p.42-57

[2] E. Monmasson and M. N. Cirstea, “FPGA design methodology for Industrial Control Systems” in IEEE transactionson Industrial applications, vol. 54, issue 4, August 2007, p.1824-1842.

[3] Atitallah, R. B., Viswanathan, V., Belanger, N., & Dekeyser, J.-L. (2018). FPGA-Centric Design Process for Avionic Simulation and Test. IEEE Transactions on Aerospace and Electronic Systems, 54(3), 1047–1065.

[4] Y. Chen , H. Helen Li, I. Bayram andE. Eken, “Recent Technology advances of Emerging memories” in IEEE Design & Test of Computers, vol.34, issue 3, June 2017, p.8-22.

[5] Woo Young Choi, “Three-Dimensional Stackable Electromechanical Nonvolatile memory cell(H cell) for four bit operation” in IEEE Electron Device Letters, vol. 31, issue 1, Jan 2010, p29-31.

[6] S. H. Teen, J. H. Lim and L. L. Lim, “IC Layout Design of Decoder Using Electric VLSI Design System” in International Journal of Electronics and Electrical Engineering vol. 3, No. 1, February, 2015, p.54-60

[7] Ming Zhao ,Xiaolin Zhang ,Ling Zhao and Chen Lee, “Design of a High Throughput QC-LDPC Decoder with TDMP Scheduling” in IEEE Transactions on Circuits and Systems II: Express Briefs vol. 62 , issue 1 , Jan. 2015 p.56-60.

[8] Malviya, Deepti & Majumdar, Shubhankar & Mishra, Mayank & Bansod, Prashant. (2012), “Comparative Study on Implementation of Various Decoder Architecture” at ICICES Chennai, February 2012, DOI:13140/2.1.4511.5524.