THE CIRCUIT DESIGNS OF AN SRAM BASED LOOK-UP TABLE FOR HIGH PERMORMANE FPGA ARCHITECTURE

Prosenjit Mal, Jason. F. Cantin and Fred. R. Beyette.[1]
Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati
PO Box 210030, Cincinnati, OH, 45221-0030, USA

ABSTRACT

Look-up table (LUT) circuits are the core component of all Field Programmable Gate Arrays (FPGA's) architectures. Although considerable research has been done regarding the high-level architecture of different LUT's, very little has been done on the circuit-level description of the LUT. Though traditional LUT designs use NMOS transistors to implement pass-gates that save area and increase speed, large LUT designs require several pass gates in series. Unfortunately, multiple pass transistors in series will degrade the logic high level and thus jeopardize signal integrity. This paper explores different circuit-level implementations of the LUT circuitry with consideration towards the relative design trade-offs.

1. INTRODUCTION

Over the years, the design and implementation of digital logic circuits using Field Programmable Gate Arrays (FPGA) has seen explosive growth. Many different architecture and programming technologies have evolved to provide better designs that make FPGA technology economically viable and an attractive alternative to Application Specific Integrated Circuits (ASIC's) [1,2,3]. With the advent of new IC technologies, multi-million gate FPGAs have become a standard component in the implementation of many digital electronic systems. In the majority of FPGA's, a look-up table (LUT) circuit is the fundamental component of the programmable logic. Any improvement in LUT design has a direct impact on the FPGA architecture as a whole. Although many reports in the literature describe high-level architectures that use the LUT in different ways,[4,5,6,7,8] very little information is available on detailed circuit design for optimization of the LUT performance. For example, as device sizes are decreased, interconnect and routing delays become a bottleneck on FPGA performance. Several research groups have proposed solutions such as 3D architectures based on either standard CMOS techniques or optical interconnects [9,10,11]. While these efforts to improve chip performance focus on faster routing/interconnection and increased logic density, they neglect the speed-up advantage achievable through optimization of the LUT design. In fact, improved LUT design will have a positive impact on all FPGA architectures including the 3D architectures recently proposed. In this paper, we have introduced new circuit designs for the LUT and investigated different circuit design issues which impact FPGA performance. To allow for comparison between the various circuit approaches, we have done circuit layout for each LUT design assuming fabrication with a 1.5μm CMOS process (λ=0.8μm). Each layout was extracted and simulated using TSPICE.

2. Look-up table circuit design

Our study investigated the performance of un-clocked, 4 input look-up tables (4-LUT). Previous studies have shown that a 4-LUT is a reasonable choice for implementation of programmable logic blocks. Here we compare 4 distinct implementations strategies for a 4-LUT.

The conventional approach is to use SRAM for holding programming bits and use pass transistors/transmission gates for decoding address. NMOS Pass transistors have been preferred[7] over transmission gates, because they 1) require fewer transistors, 2) well area is conserved, 3) extra inverters are avoided and 4) junction capacitance is reduced. These factors have a direct impact on chip area and switching speed of the FPGA. The biggest disadvantage of using NMOS pass transistors is the degradation of signal integrity caused by

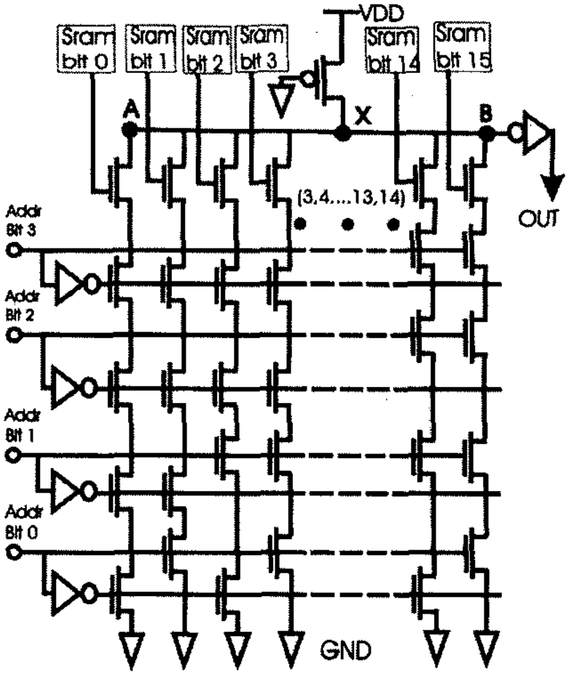

Figure 1 Circuit diagram of a 4-LUT using stacks of NMOS's with $(W/L)_P$=0.4, $(W/L)_N$=5.

[1]Correspondence: Email: beyette@ececs.uc.edu; Telephone: 15135564588;Fax: 15135567326;http://www.ececs.uc.edu/~beyette

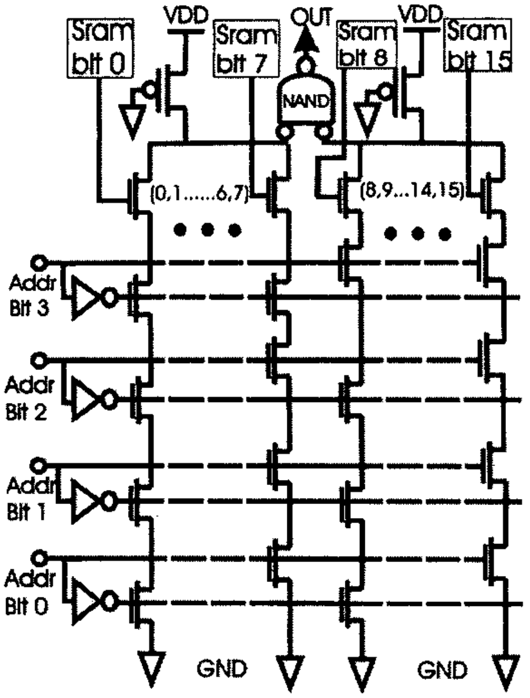

Figure 2 Circuit diagram of a 4-LUT using divided stack technique with $(W/L)_P$=0.4, $(W/L)_N$=5.

threshold voltage drop across the transistor when passing a logic '1' level. In the traditional approach [5,7,12] program bits from an SRAM are passed through several pass transistors in the decoder part of the LUT. As the size of the LUT increases beyond a couple of inputs, the integrity of the programmable bits is compromised. With decoders implemented using multiple pass transistors the situation is further exacerbated by the 'body-effect' which tends to increase the threshold voltage slightly.

To overcome this problem and at the same time to reap the benefits of NMOS transistor based designs, we have introduced an innovative LUT circuit. This circuit does not use NMOS pass-transistors and so does not suffer from the signal integrity problem described above. As shown in Figure 1 the circuit includes 16 columns with each column containing five NMOS transistors and one SRAM cell to hold the program bit. With four address bits, 1 out of 16 columns can be selected and depending on the status of the SRAM bit the match line (AB) will be either pulled-up or down. The simulation shows that at the node 'X' pull-up time (t_{up}=41.37ns) is considerably larger than the pull-down time (t_{dn}=3.66ns). This is because the node 'X' has large capacitance as 16 transistors are connected in parallel and the node is pulled up using a "static-load" NMOS structure.

Ideally, the pull-up and pull-down times should be approximately equal. One possible solution is to make the pull-up faster by using a stronger PMOS load. But any increase in the W/L ratio of the PMOS transistor will decrease the rise time t_r of the node 'X' at the cost of increasing the static power consumption of the circuit. Thus, there is a trade-off between static power consumption and rise time (t_r). More over, the circuit should have reasonable β (the ratio of W/L of the PMOS and effective W/L of the NMOS's in series) to achieve a good compromise between speed and sensitivity. Simulation has shown that for $\beta \geq 0.4$ optimum circuit performance is achieved. Further, it has been shown that for $\beta \geq 0.4$ these circuits have reasonable noise margins of NM_L=0.9V and NM_H=1.7V or better. Note that values of β chosen for some performance gain vary from one process to the other. Here, β=0.4 pertains to AMI 1.5μm process.

One can easily see that the circuit shown in Figure 1 becomes sluggish if one tries to implement a larger LUT using the same approach. Each additional input to the LUT doubles the number of columns and hence the capacitance of the node X which makes the rise time (t_r) of the node longer. One way to fight this problem and to make the circuit faster is depicted in Figure 2. The match line (AB) can be broken into multiple pieces. Each piece (i.e. 0 to 7 & 8 to 15 in Figure 2) is similar to the approach shown in Figure 1 . However, instead of using a simple inverter to drive the output load, a NAND gate is used to connect the pieces. Each line can be discharged and pulled up quickly, and output of the NAND gate goes high if either line is pulled down. As the match-line is divided into pieces, capacitance of the common node 'X' decreases, which in turn reduces the pull-up time of the node 'X' (t_{up}=19.82ns & t_{dn}=2.47ns). The progressive sizing of the pull-down NMOS's in all 16 columns of the circuit also reduced the pull down time. Progressive sizing implies that as one moves down the stack the NMOS transistors will be progressively larger.

Another technique that can be used to make the pull-up circuit faster is shown in Figure 3. Here a clocked PMOS hybrid load is used to help pull up the node X. The size of the PMOS transistor

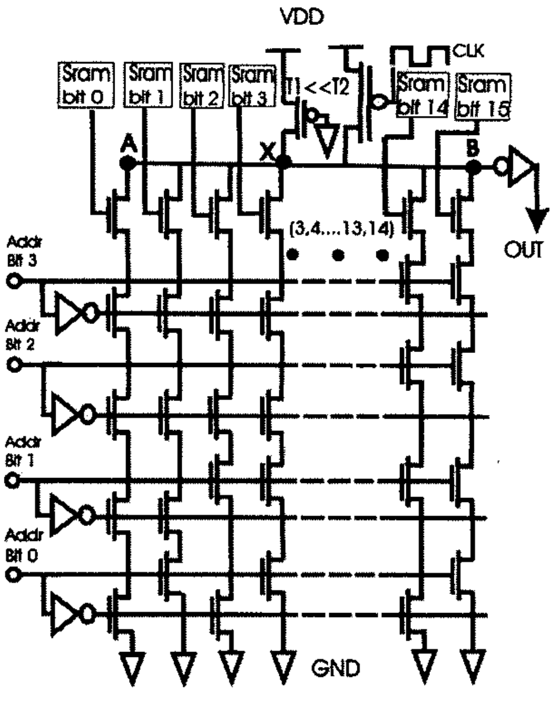

Figure 3 Circuit diagram of a clocked 4-LUT

T2 is much larger than that of PMOS transistor T1. The small PMOS-T1 (bleeder) will always be active to keep the line static. This fights noise and charge sharing. However, the large PMOS-T2 helps make pull-up fast during the low clock phase. This solution works nicely with a clocked LUT. Unfortunately, clocked LUT's are not always suitable for static logic systems.

To improve LUT performance with out the use of a clocked PMOS pull-up we must reduce the number of NMOS transistors used in the decoder circuit. Figure 4 shows one approach. In this circuit, all the NMOS transistors previously driven by address bit Addr3 are removed from the pull down stacks. The 2-input NAND gate that joins two-match lines in Figure 2 is replaced with an **OAI (OR-AND-INVERT)** gate. This is a compound gate with four inputs. The result will be the OR of 'match-line-X' and 'Addr bit3' NAND-ed with the OR of 'match-line-Y' and inverted 'Addr bit3'. We make sure that the compound gate PMOS transistor closest to the gate output is driven by the match-line. This insures the maximum decoder switching speed. Now, it is easy to make pull-up PMOS transistors larger leading to faster pull-up speeds than the previous approaches. Further, we have four NMOS transistors instead of five in each column. In contrast to the implementation of Figure 2, this approach requires total 16 fewer NMOS transistors in the NMOS stacks (columns 0 to 15). Thus, we expect an increase in LUT performance due to the removal of NMOS transistors which have finite series resistance and output capacitance. The OAI allows us to reduce the width of the pull-down NMOS's and that allows a proportional increase of pull-up PMOS for the same β. Overall, this implies a faster circuit. Also, note that the fourth input is no longer on the critical path.

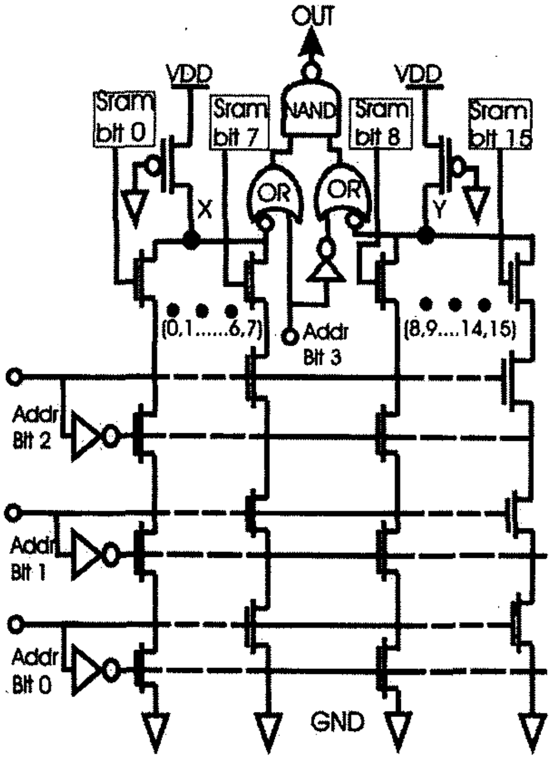

Figure 4 Circuit diagram of a 4-LUT using compound gate (OAI) with $(W/L)_P$=0.3, $(W/L)_N$=4.

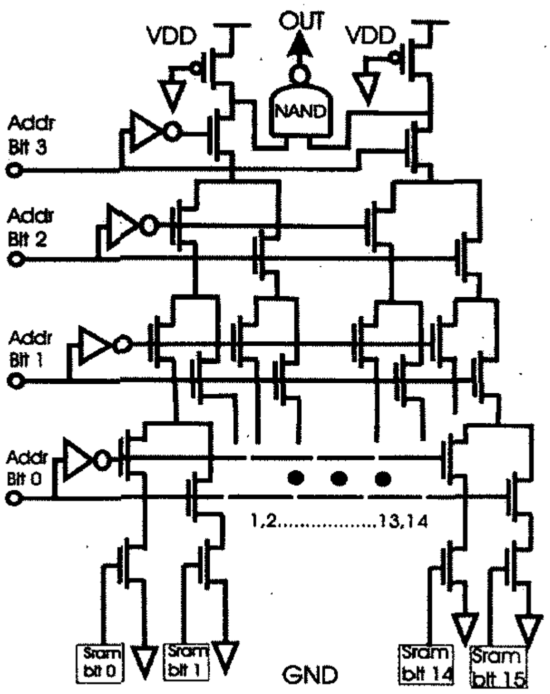

Figure 5 Circuit diagram of a 4-LUT using tree structure $(W/L)_P$=0.4, $(W/L)_N$=5.

Figure 5 shows another approach for implementing a fast LUT. Like the conventional tree-LUT [7,12] the circuit shown in Figure 5 does not use NMOS as pass gates and thus does not suffer signal degradation when passing a logic '1'. In contrast to the previous circuits, this approach merges the stacks in a tree like structure that allows lower level nodes to share NMOS transistors higher on the tree. Instead of 16 columns, we have two main trees with many sub-trees of NMOS, all passing zeros. The circuit has one NAND gate, two pull-up PMOS transistors, two NMOS transistors on the top of the tree and each NMOS is followed by two others in parallel, and so on. As number of NMOS transistors tied to match-line is only one, it helps pull-up considerably faster than other circuits. The total number of NMOS transistors required to implement this circuit is drastically reduced from 64 required to implement the circuit in Figure 1 to 30 required to implement the circuit in Figure 5. Both approaches used 16 additional NMOS transistors for SRAM inputs and two for NAND gate.

3. Simulation Results

As stated above, we have produces a layout for each circuit and then extracted and simulated with TSPICE (using BSIM parameters available on MOSIS web page [15] for AMI 1.5μm CMOS process). We have compared the results in Table 1. Since clocked LUT's are beyond the scope of this paper, we have not included the circuit shown in Figure 3. The table shows a comparative study of different figure of merits (i.e. propagation delays (t_{pHL} & t_{pLH}), average static power (P_{STA}) & average dynamic (P_{DYN}) power consumption, fan-out factor and area) for each circuit. Average dynamic power is measured when the circuit is switching at some target frequency of 3.0MHz. For calculation, we have chosen an average SRAM configuration and randomly changed the MUX inputs at the target frequency. Average static power is measured during an interval when the circuit is not switching. Fan-out factor measures a ratio of drive of the output gate to the load of the LUT inputs. The LUT approach shown in Figure 4 has better fan-out factor than other approaches. Having this fan-out factor one LUT can easily drive another LUT, and up to 4 other LUT's with a buffer inserted (assuming 10X rule). Areas of the different LUT's are comparable and we are able to improve speed without significantly sacrificing area. To maintain the required 'β' and reasonable noise margins, each circuit was tweaked to different transistor sizes for required propagation delays and power consumption. In actual layout, other tricks like splitting the pull-up PMOS into two small ones place one on each end of the match-line and ratio them, was used.

TYPE OF LUT	t_{pHL}/t_{pLH} (ns)	AV. STA. POW. (P_{STA}) (mW)	AV. DYN. POW. (P_{DYN}) (mW)	FAN-OUT FACTOR (LOAD/ DRIVE)	AREA (μm^2) X 10^3
FIG.1	47.84/5.33	0.17	0.34	1.3	74.2
FIG.2	24.10/7.22	0.39	0.41	1.3	77.9
FIG.4	18.36/2.11	2.12	2.3	1.6	74.0
FIG.5	12.69/7.15	3.9	4.3	1.3	75.9

Table 1 Results from simulation of four LUT circuits

4. SUMMARY

In summary, we have described new kind of transistor-level circuit design for the look-up table circuit in a FPGA. In contrast to conventional LUT implementations which rely on pass gates, our LUT designs do not suffer from logic level '1' degradation associated with NMOS pass transistors. We have completely avoided the use of transmission gates to achieve faster circuits at minimum area cost. We have pointed out important issues in circuit design pertaining to the performance of the circuit. A test chip has been designed where several 4-LUT's are used for the purpose of evaluation and characterization of performance and sent for fabrication to MOSIS [13] using AMI 1.5μm CMOS process. Our near future plans include further experimentation on fast Look-up table circuits using more advanced CMOS processes. Our preliminary work indicates that the approaches presented here will have a significant impact on the performance of look-up table based FPGA chips.

5. REFERENCES

[1] H. Hsieh, W. Carter, J.Y. Ja, E Cheung,S. Schreifels, C. Erickson, P.Fredin, and L. Tinkey, "Third-generation architecture boosts speed and density of field-programmable gate arrays," *in Proc. Custom Integrated Circuits Conf.*, 1990,pp.31.2.1-31.2.7.

[2] J. Rose and D. Hill, "*Architectural and Physical Design Challenges for One-Million Gate FPGAs and Beyond,*" Proceedings of the 1997 ACM Fifth International Symposium on Field-Programmable Gate Arrays, pp. 129-132. February 1997.

[3] A. Singh, et al. "*A Novel High Throughput Reconfigurable FPGA Architecture,*" Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 22-29. February 2000.

[4] S.Singh, J.Rose, P.chow and D.Lewis, "*The Effect of Logic Block Architecture on FPGA Performance,*" IEEE Journal OF SOLID-STATE CIRCUITS, Vol27,No.3,march 1992.

[5] S.Brown, R. Francis, J.Rose, and Z. Vranesic, "*Field-programmable gate Arrays*". Norwell, MA: Kluwer, 1992

[6] D, Hill and N, Woo, "*The Benefits of Flexibility in Lookup Table-based FPGA's*", IEEE transactions on Computer-Aided Design of integrated Circuits and Systems, vol. 12. No.2.February 1993.

[7] P. Chow, S. O. Seo, J. Rose, K. Chung, G. P. Monzon, and I. Rahardja, "*The Design of a SRAM-Based Field-Programmable Gate Array--Part-I* &II", IEEE transactions on VLSI systems, Vol. 7, No. 2, pp. 191-197, June 1999 and Vol. 7, No. 3, pp. 321-330, sep. 1999

[8] *The Programmable Logic Data Book*, Xilinx Inc., 1999.

[9] S. Chiricescu, M.Leeser and M.Michael Vai,"*Design and Analysis of a Dynamically Reconfigurable three-Dimensional FPGA*", IEEE transactions on VLSI systems, Vol. 9, No.1, pp. 186-196, Feb, 2001.

[10] M. Leeser,W.Meleis,M.vai, S. Chiricescu,W.Xu, and P.zavracky, "Rothko:A three dimentional FPGA, "IEEE Design Test Computers, pp.16-23,Jan-Mar. 1998.

[11] J. Van Campenhout, H. Van Marck, J. Depreitere, and J. Dambre, "*Optoelectronic FPGA's*", IEEE J. Select. Top. in Quant. Elect., Vol. 5, No. 2, pp. 306-315, March/April 1999.

[12] V.Betz, J, Rose, A. Marquardt, "Architecture and CAD for Deep-Submicron FPGAs" Norwell, MA: Kluwer, 1999

[13] MOSIS Web page at http://www.mosis.org.