ADVANCED RECONFIGURABLE FPGA ARCHITECTURE: MAXIMIZING DENSITY WITH OVERLAY VIA-SWITCH CROSSBARS

Dr.G. Anantha rao
Electronics and communication Engineering department
GMR Institute of Technology, Rajam
Andhra Pradesh, India

Sai Jaswanth
Electronics and communication Engineering department
GMR institute of Technology, Rajam
Andhra Pradesh, India
Jaswanths768@gmail.com

Abstract

This paper proposes a highly dense reconfigurable architecture that introduces via-switch device, which is a nonvolatile resistive-change switch and is used in crossbar switches.

Via-switch is implemented in back-end-of-line layers only, and hence the front-end-of-line (FEOL) layers under the crossbar can be fully exploited for highly dense logic blocks. The proposed architecture uses the FEOL layers for fine-grained lookup tables and coarse-grained arithmetic/memory units for improving performance and compatibility with various applications.

A case study of application mapping shows the proposed architecture can reduce the array area by 21.7%, thanks to the bidirectional interconnection. Thanks to 18F2 footprint and one order of magnitude lower resistivity of via-switch compared to MOS switch, the crossbar density is improved by up to 26× and the delay and energy in the interconnection are reduced by 90% and 94% at 0.5-V operation.

Key points:1)Via switch device.
2)FEOL
3)BEOL

Introduction

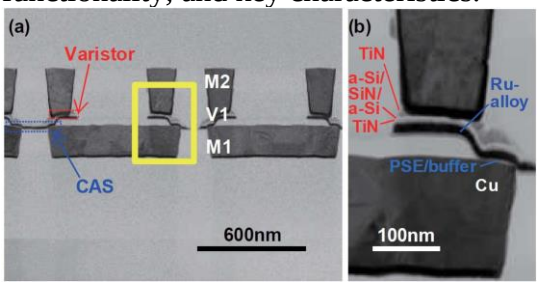
This paper proposes a highly dense reconfigurable architecture that introduces via-switch device, which is a nonvolatile resistive-change switch and is used in crossbar switches.

Via-switch is implemented in back-end-of-line layers only, and hence the front-end-of-line (FEOL) layers under the crossbar can be fully exploited for highly dense logic blocks. The proposed architecture uses the FEOL layers for fine-grained lookup tables and coarse-grained arithmetic/memory units for improving performance and compatibility with various applications.

A case study of application mapping shows the proposed architecture can reduce the array area by 21.7%, thanks to the bidirectional interconnection. Thanks to 18F2 footprint and one order of magnitude lower resistivity of via-switch compared to MOS switch, the crossbar density is improved by up to 26× and the delay and energy in the interconnection are reduced by 90% and 94% at 0.5-V operation.

I. VIA SWITCH

The 2V-1CAS, named Via-switch, is a compact and nonvolatile switch incorporating a Crossbar Array Structure (CAS) and two varistors. Its design caters to accommodating multiple fan-outs within a crossbar switch . Precise individual programming of each cross-point is achieved via two control lines linked to the varistors, eliminating the need for access transistors. The Via-switch boasts integration within a minimal footprint of $18F^2$. Section II-B will delve into the programming specifics of the crossbar via-switch, detailing its device structure, functionality, and key characteristics.



II. SRAM FPGA ARCHITECTURE

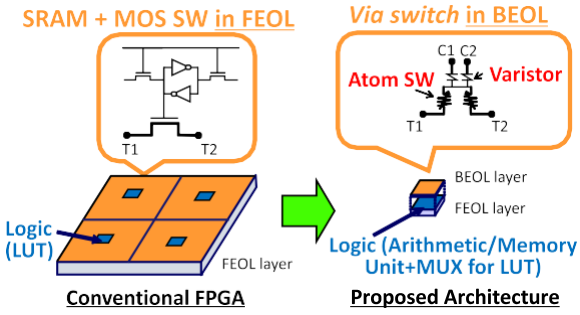
FPGA, a widely recognized type of reconfigurable integrated circuit, typically comprises programmable Logic Blocks (LBs), configurable I/O blocks, and adaptable routing resources. LBs are organized in an array, and routing channels form connections between these blocks in an island-like architecture. Each LB commonly contains Look-Up Tables (LUTs) and optionally includes D-type flip-flops (D-FFs). The k-input LUT (k-LUT) integrates 2k SRAM elements alongside a 2k-to-1 multiplexer. Additionally, coarser-grained LBs like the multiplier-accumulator (MAC) unit and block memory are embedded within the array. At every intersection of routing channels, a switch block (SB) facilitates connections between wire segments. Adjacent to each LB's edge, a connection block enables wire segment connections and port access for the blocks. Configuration of routing resources is maintained by on-chip SRAM elements.

I. INTERCONNECT STRUCTURE

A. cross bar programming

When utilizing a varistor for device selection during programming, it's crucial to address and mitigate the sneak path problem. This issue is exemplified in a prior study that implemented the via-switch with a CAS and a single varistor (1V-1CAS) , and a similar 1D2R device incorporating one diode and two variable resistors was recently documented in . the crossbar structure displays horizontally and vertically aligned signal lines, while control lines are routed diagonally. This specific crossbar configuration allows programming wherein only one intersection can be activated at most for each horizontal and vertical signal line, thereby disallowing multiple fan-outs.

Let's delve into the repercussions of attempting to activate multiple 1V-1CAS via-switches along a vertical signal line. Figure 8 elucidates the programming steps, each involving the activation of an atom switch. A positive voltage ("1") is applied to one of the signal lines, while a ground voltage ("0") is directed to one of the control lines, leaving the other lines floating.



B.Proof of sneak path avoidance

- We've established that the 2V-1CAS via-switch crossbar structure doesn't encounter the sneak path problem under a specific constraint allowing multiple ON via-switches in only

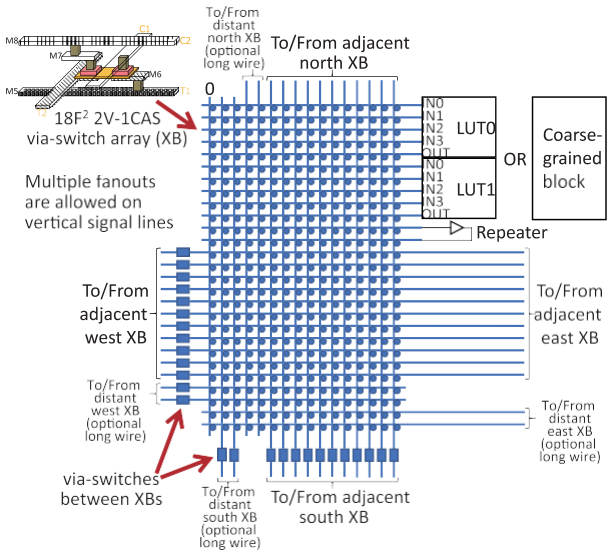
one direction. Demonstrating this, we'll employ an induction-based proof procedure.

- Proving the absence of the sneak path problem in the programming of a 1x1 crossbar (a single via-switch) is self-evident.
- Assuming the absence of the sneak path problem in each programming step for any configuration pattern in an MxN crossbar:
- a) Show that the sneak path problem doesn't arise at each programming step for any configuration pattern in an (M+1)xN crossbar.
- b) Show that the sneak path problem doesn't arise at each programming step for any configuration pattern in an Mx(N+1) crossbar.
- Once steps 1 and 2 are proven, it follows that there's no sneak path problem in the programming of crossbars of any size.

This induction-based approach allows us to establish the absence of the sneak path problem systematically across different crossbar sizes by building upon smaller proven cases.

C.Propose interconnect structure

Thanks to the 18F2 (6F, 3F) via-switch, specifically designed for implementation solely on Back-End of Line (BEoL) layers, compact crossbars can be achieved. For instance, a 100x100 crossbar can be realized within dimensions of 60µm x 30µm in a 65-nanometer node (F100nm). Consequently, the interconnect resistance per crossbar remains low, eliminating the necessity for a repeater for each crossbar. Furthermore, the inherent bidirectional signal transmission capability of the via-switch enhances routing efficiency per signal line. This improvement leads to a reduction in the required number of signal lines.

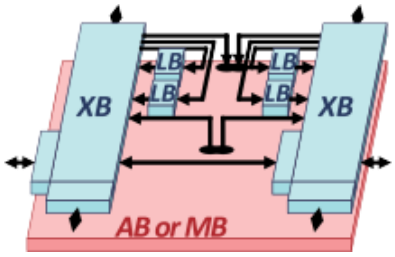


V.LOGIC STRUCTURE

- UNIT TITLE
- The programmable logic resource within the proposed architecture comprises both fine-grained blocks (e.g., LUT) and coarse-grained blocks (e.g., multiplier and memory), mirroring contemporary commercial FPGAs. The architecture is structured as a 2-D array of "unit tiles". Each unit tile includes two crossbar blocks (XBs), four fine-grained LBs, and either a coarse-grained arithmetic block (AB) or memory block (MB). The larger area and pin counts of coarse-grained blocks (e.g., AB or MB) necessitate connections to multiple XBs.

- To optimize area efficiency, the transistor area under the crossbar should be fully dedicated to LBs. To identify suitable LBs, let's delve into the logic area and switch area occupied by LBs, XBs, and ABs/MBs. To facilitate this,

we define certain parameters. Ntr represents the number of tracks between two adjacent XBs, assuming an equal number of vertical and horizontal tracks for simplicity. Nlocal_IO denotes the total count of local interconnects between an XB and the I/O pins of the LBs and the AB/MB.

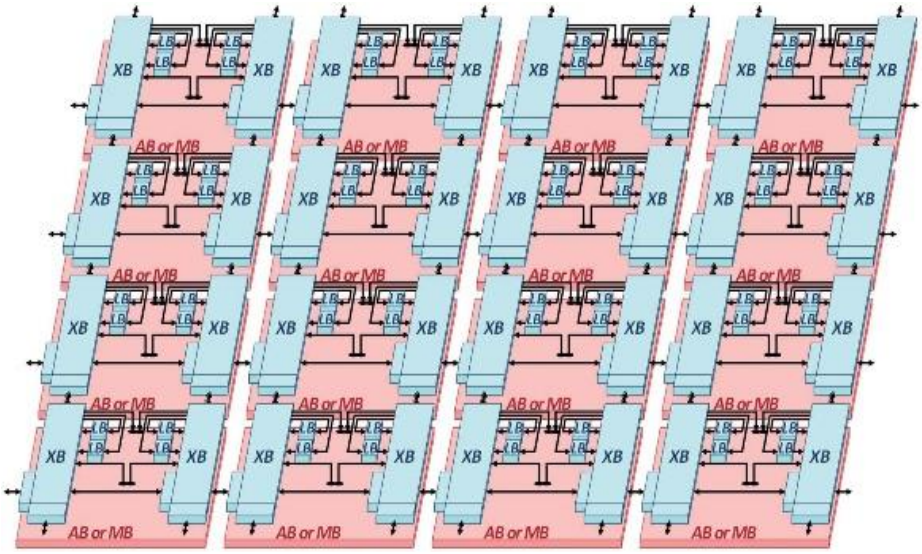


B. XB, LB, and AB/MB

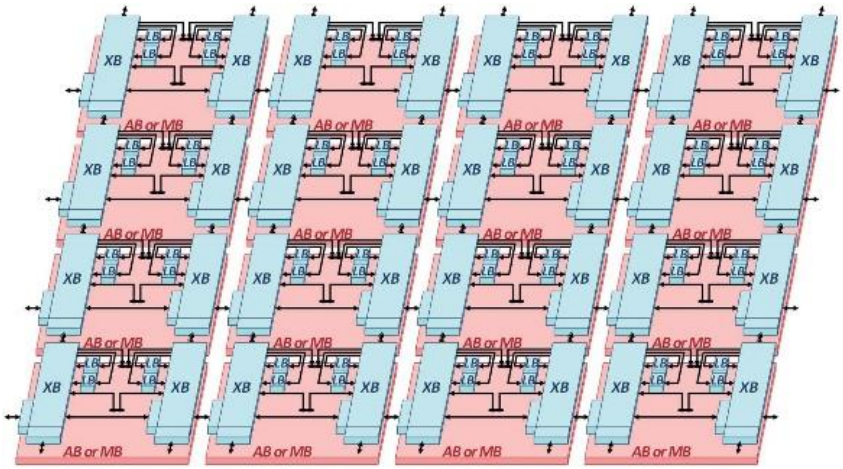
- The count of switches within an XB is determined by (Nlocal_IO * Ntr) * Ntr, with the XB area calculated as (Nlocal_IO * Ntr) * Ntr * 18F2.
-

Moving on to the area computation of an LUT, the fundamental component of the LB. A traditional SRAM-based 4-input LUT comprises 16 SRAM cells and a 16-to-1 multiplexer (16-MUX). In [reference], a 4-LUT is constructed using 32 CASs and a 16-MUX (referred to as a 0/1-type LUT), while [26] introduces an enhanced 4-LUT architecture employing 32 CASs and an 8-MUX.

• **Parameter/** • **Feature**	• **Description/Value**
• Logic Density	• 3 million logic elements
• Reconfiguration Speed	• 100 microseconds
• Via-Switch Crossbars	• 8 layers with 256x256 switch matrix
• Power Efficiency	• 0.5 Watts per million gates
• Maximum Operating Frequency	• 1 GHz
• On-chip Memory	• 10 MB embedded memory
• Configuration Interfaces	• PCIE, JTAG, Ethernet
• Supported Standards	• OpenCL, VHDL, Verilog
• I/O Interfaces	• 500 high-speed transceivers
• Area Efficiency	• 70% utilization of available FPGA area



(a)



(a)

CONCLUSION:

The authors of this term paper implemented an advanced reconfigurable FPGA configuration, resulting in a 21.7% reduction in the chip's area. This reduction significantly boosted circuit performance, by decreasing delay and energy usage by 90% and 94%, respectively, especially at during 0.5-V operation. Additionally, the power per unit area decreased, leading to a remarkable increase in circuit efficiency. Moreover, the efficient utilization of approximately 70% of the circuit's area further contributed to the overall enhanced efficiency.

REFERENCES:

- I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," IEEE Trans. Comput.-Aided Design Integral . Circuits Syst., vol. 26, no. 2, pp. 203–215, Feb. 2007.
- M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3-D FPGA," IEEE Trans. Computer.-Aided Design Integral. Circuits Syst., vol. 26, no. 2, pp. 216–229, Feb. 2007.
- R. Rajaei, "Radiation-hardened design of non volatile MRAM based FPGA," IEEE Trans. , vol. 52, no. 10, Oct. 2016, Art. no. 3402010.
- W. Zhao, E.Belhaire , V.Javerliac , C Chappert , and B.Dieny, "Evaluation of a non-volatile FPGA based on MRAM technology," in Proc. ICICDT, 2006, pp. 1–