A Novel Delay-Aware Packing Algorithm for FPGA Architecture Using RFET

Sheng Lu
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
sxl2408@mavs.uta.edu

Liuting Shang
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
liuting.shang@mavs.uta.edu

Sungyong Jung
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
jung@uta.edu

Zhenlin Pei
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
zhenlin.pei@mavs.uta.edu

Yichen Zhang
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
yichen.zhang@mavs.uta.edu

Chenyun Pan
Department of Electrical Engineering
The University of Texas at Arlington
Arlington, USA
chenyun.pan@uta.edu

***Abstract*—Reconfigurable devices are gaining increasing attention as a viable alternative and supplementary solution to traditional CMOS technology. In this paper, we develop a more efficient 2-input look-up table (LUT) based on the reconfigurable field-effective transistors (RFETs), leading to a smaller transistor usage and a smaller critical path delay. The cells are organized into regular matrices, known as MClusters, with a fixed interconnection pattern to replace LUTs in field-programmable gate arrays (FPGAs). To improve the efficiency of utilizing this structure, we design a SAT-based delay-aware packing algorithm to better utilize logical gates for the MCluster structure. Finally, we combine this algorithm with FPGA simulation tools to form a comprehensive benchmarking flow. A series of benchmark tests show that under the optimal design, up to 35% and 30% reduction can be achieved in delay and energy-delay product (EDP), respectively, compared to the traditional CMOS FPGAs.**

Keywords—Reconfigurable FET, field-programmable gate arrays, look-up table, partitioning algorithm, packing algorithm, technology mapping, performance benchmarking.

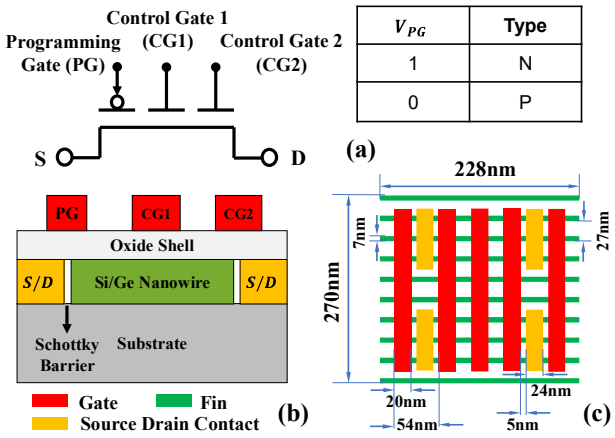
I. Introduction

FIELD programmable gate arrays (FPGAs) are advanced integrated circuits that consist of configurable logic blocks (CLBs) interconnected via programmable routing resources such as look-up tables (LUTs) and flip-flops [1, 2]. This architecture enables dynamic reconfiguration of logic functions, providing FPGAs with remarkable versatility. Unlike application-specific integrated circuits (ASICs) that are designed for specific tasks, FPGAs can be reprogrammed to perform various functions or update existing ones. This adaptability fosters rapid prototyping and design iteration, significantly reducing development time and costs.

Research on FPGAs, like other integrated circuits, aims to reduce size, delay, and power consumption. In addition to solutions for optimizing existing architectures, several studies have demonstrated that using emerging electronic devices, such as carbon nanotube transistors (CNFETs) [3], memristors [4], spintronic [5], etc., to optimize the performance of FPGAs by leveraging their unique device characteristics, which differ from those of CMOS. Among the emerging devices, the reconfigurable field-effect transistor (RFET) is a device controlled by multiple gates. Its program gate controls the polarity of the device, allowing it to switch between N- and P-type semiconductors, and its control gates control the on-and-off state of the transistor.

Due to the properties of RFETs, researchers have developed various methods for constructing FPGAs with RFETs. For example, Jamaa et al. proposed an FPGA design that utilizes double-gate CNFETs to create reconfigurable logic gates [3]. Gaillardon et al. developed a new FPGA architecture based on silicon nanowire RFETs, which organizes reconfigurable logic gates in a specific topology to form an efficient computation cluster that replaces traditional LUTs [6]. Cheng et al. designed hybrid topologies to efficiently map any function on nano-grain cell-based architectures [7]. These works typically use efficient reconfigurable logic gates, such as NAND/NOR, and XOR/XNOR, which are combined into logic units that are capable of implementing various logic functions instead of LUTs in the FPGA. Although RFET-based logic gates are compact and have low delay, the logic functionality is less versatile compared to LUTs.

To create such logic units efficiently, two crucial steps are typically involved, including (i) clustering, which involves gathering logic gates for a logic unit, and (ii) mapping, which determines whether these gathered logic gates can be mapped to a single reconfigurable logic unit. Gaillardon et al. developed a reconfigurable device-based circuit using a VPack-based packing algorithm combined with a recursive mapping algorithm [6]. Here, VPack is a seed-based greedy algorithm that combines as many cells around the seed as possible. Although the VPack-based packing algorithm has high packing efficiency [1], it does not consider the delay of the final circuit or the energy consumption caused by routing. Furthermore, the recursive mapping algorithm does not efficiently utilize the reconfigurable logic units, resulting in many inserted buffers.



V_{PG}	Type
1	N
0	P

Fig. 1. (a) Symbol of RFET device, which can be switched between N- and P-type by different gate voltages on the program gate. (b) Cross-section view of tri-gate RFET transistor. (c) Layout of tri-gate RFET transistor.

In this paper, we take advantage of the multi-gate characteristics of RFET and its reconfigurable characteristics to design a more compact 2-input LUT. Based on the work of Gaillardon et al. [6], these 2-input LUTs are arranged into regular matrices, called MClusters. To maximize the number of logic gates that can be mapped to a single Mcluster structure, we develop a delay-aware packing algorithm based on partition. This algorithm also utilizes Boolean satisfiability (SAT) solver-based reconfigurable synthesis which can determine whether certain gate connections can be mapped to a single MCluster circuit [8]. This will substantially improve the utilization of the Mcluster and lead to a smaller number of clusters for the FPGAs.

The major contributions of this paper are listed below.

- We design a compact and efficient matrix-based logic unit based on RFETs to replace the traditional 6-input LUTs in FPGAs.
- We develop a delay-aware packing algorithm based on partition and SAT solver to greatly enhance the technology mapping efficiency.
- We develop a comprehensive design framework by integrating the proposed packing algorithm with FPGA simulation tools to perform system-level optimization to showcase the advantage of the proposed design.

II. Modeling Approach

A. RFET Device-Level Characteristics

RFETs have multiple gates containing a programming gate (PG) for switching transistor polarity and multiple control gates (CGs) for controlling the on and off states of transistors. Fig. 1(a) and Fig. 1(b) show the symbol of a tri-gate RFET and its corresponding cross-section view. The RFET is turned on only when the PG and CGs are at the same voltage value. It is important to note that the on-resistance of the device is mainly determined by the resistance of the source-sided barrier. Increasing the number of gate terminals does not have a significant effect on the current through the device [9]. The single fin RFET on-state current is about 15μA at 0.8V Vdd, as shown in [11], which is three times lower than its CMOS

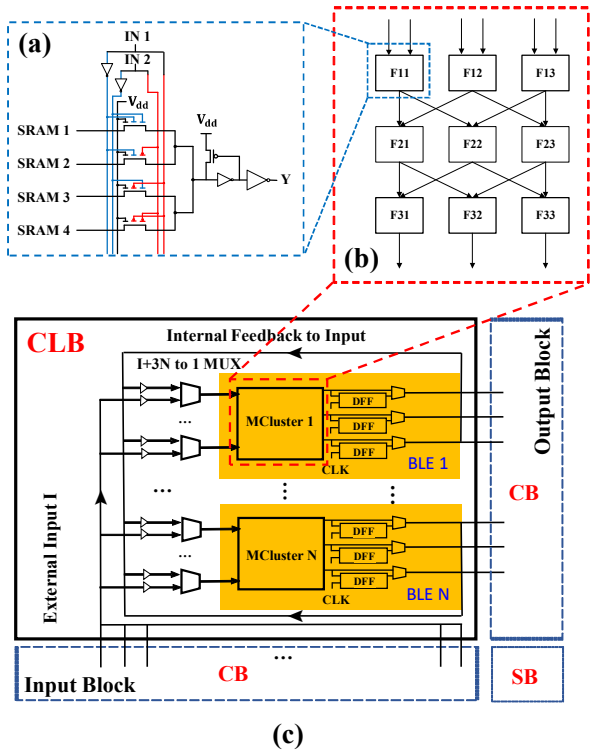


Fig. 2. (a) Schematic for 2-input LUT based on RFETs, which can represent for any 2-input logic functions by different SRAM configuration. (b) A MCluster of 9 2-input LUTs with fixed interconnect topology. (c) A generic FPGA architecture and detailed logic cluster structure.

counterpart based on ASAP 7nm PDK [10]. The gate capacitance of RFET and CMOS devices are similar, assuming they have the same number of input fins of 3.

To accurately determine the area of the RFET-based logic cell, which affects the overall FPGA area and global routing channel length, we design individual transistor layouts based on ASAP 7nm PDK design rules and RFET physical structure [10], as shown in Fig. 1(c).

B. RFET-based Logic Cell and MCluster for FPGA

RFETs can be used to realize more compact LUTs/ multiplexers (MUXes) due to their multi-gate characteristics. As shown in Fig. 2(a), the program gate of the transistor is fixed to Vdd, leading to an N-type transistor. Conventional 2-input LUTs typically require two columns of pass transistors, but the multi-gate characteristic of RFETs allows them to be compressed into a single-column structure, significantly reducing the number of transistors and critical path length.

To properly balance the delay and energy consumption of FPGA systems, we adopt the design of MCluster, as shown in Fig. 2(b), which utilizes a matrix of 2-input LUTs to form a 6-input, 3-output logic cell. Based on the simulation, the MCluster is found to have 53% fewer transistors, 43% fewer SRAMs, 43% less area, 40% less delay, and 50% less switching energy compared to the 6-input LUT using 7nm FinFET transmission gates. These advantages come at the cost of less function-level versatility compared to traditional LUTs, and such trade-offs

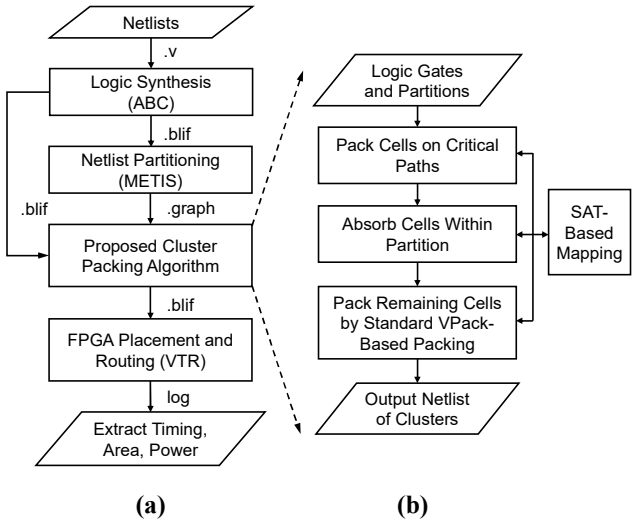


Fig. 3. Proposed system-level simulation framework. (a) Overall benchmark process for MCluster, and (b) proposed delay-aware packing algorithm flow.

will be evaluated at the system level to understand the true value of the proposed design.

Fig. 2(c) shows the internal structure and related parameters of the FPGA used in this paper. Different from the traditional LUT-based FPGA, each CLB contains multiple basic logic elements (BLEs) that have three outputs instead of one output, which will induce additional local routing costs, and the impact from such an area overhead will also be discussed in Section III.

C. System-Level Simulation Flow

The overall proposed simulation flow is shown in Fig. 3(a). First, the benchmark netlists are synthesized from Verilog format to 2-input logic gates format by ABC [12]. Then, we develop a delay-aware packing algorithm to pack logic gates into MClusters, and the packing flow is shown in Fig. 3(b), which will be discussed in detail in the next subsection. Finally, we perform FPGA placement and routing based on an open-source VPR tool [13], and the area and timing results can be extracted accordingly. The energy consumption can be obtained based on an activity factor file generated by ACE 2.0, which is integrated into the VPR flow [14]. Unless specified elsewhere, the default system configuration utilizes tri-gate RFET devices with a Vdd of 0.8V. Each CLB consists of 10 BLEs, with a 6-input MCluster in each BLE.

D. Delay-Aware Packing Algorithm based on Partition

To reduce the global delay of the circuit while optimizing dynamic energy consumption, we combine the algorithm of network partitioning with delay-aware packing. First, the circuit structure is divided into smaller block circuits of equal size using Metis partition tool [15]. The partitioning process recursively divides the circuit into two parts until the number of logic gates in each partition is fewer than the target minimum size P_{min}.

The proposed packing algorithm mainly comprises three steps, as illustrated in Algorithm 1. The first step involves

Algorithm 1: Delay-aware Packing Algorithm

```
// Constrain from cluster physical structure
DECLARE Physical_Constraint
 // Transform string text to number matrix
 netlist_graph = Blif2Graph (blif_file)
 // Recursive based network bi-partition
 partition_result = Partition_Recursive (netlist_graph, min_part_size)
 // First packing cells on critical paths
 FOR (Longest N percent critical path)
        new_cluster = first_unpacking_cell_on_path
        WHILE satisfy Physical_Constraint
                IF Percy Mapping Pass
                        new_cluster add next_cell_on_path
                ENDIF
        ENDWHILE
        clusters add new_cluster
 ENDFOR
 // Then absorbing cells from the same part
 FOR (Each clusters)
        WHILE satisfy Physical_Constraint
                Best_cell = Cell_Absorbing (clusters,partition_result)
                IF Percy Mapping Pass
                        current_cluster add Best_cell
                ENDIF
        ENDWHILE
 ENDFOR
 // Seed-based packing for remaining cells
 WHILE (unpacking_cells != NULL)
        new_cluster = Seed (unpacking_cells)
        WHILE satisfy Physical_Constraint
                Best_cell = Cell_Absorbing (new_cluster,partition_result)
                IF Percy Mapping Pass
                        new_cluster add Best_cell
                ENDIF
        ENDWHILE
        clusters add new_cluster
 ENDWHILE
```

packing the logic gates on the top $Crit_N$ percent of the critical paths to minimize the number of MClusters on the critical path. In the second step, for each packed MCluster that is not fully utilized, nearby logic gates from the same partition are added. Finally, the remaining unpacked gates are packed using the traditional VPack-based packing algorithm. Similar to the second step, the gates in the same MCluster can only come from the same partition to maintain the proximity of the logic gates. This ensures the gates that are close to each other in physical position remain close to each other after packing, minimizing the dynamic energy consumption caused by global routing. We will sweep parameters, P_{min} and $Crit_N$, to optimize the overall system-level performance metrics in Section III.

III. Simulation Results

A. Packing Algorithm Efficiency Comparison

We first quantify the performance improvement of the proposed delay-aware packing algorithm compared to the traditional VPack-based packing algorithm for the area, delay, and switching energy. As shown in Table I, the proposed packing algorithm provides a significant advantage on relatively large-scale circuits in terms of area. For small-scale circuits, Vpack has a smaller area because VPack allows for more efficient packing of smaller circuits, reducing the number of CLBs and overall circuit area.

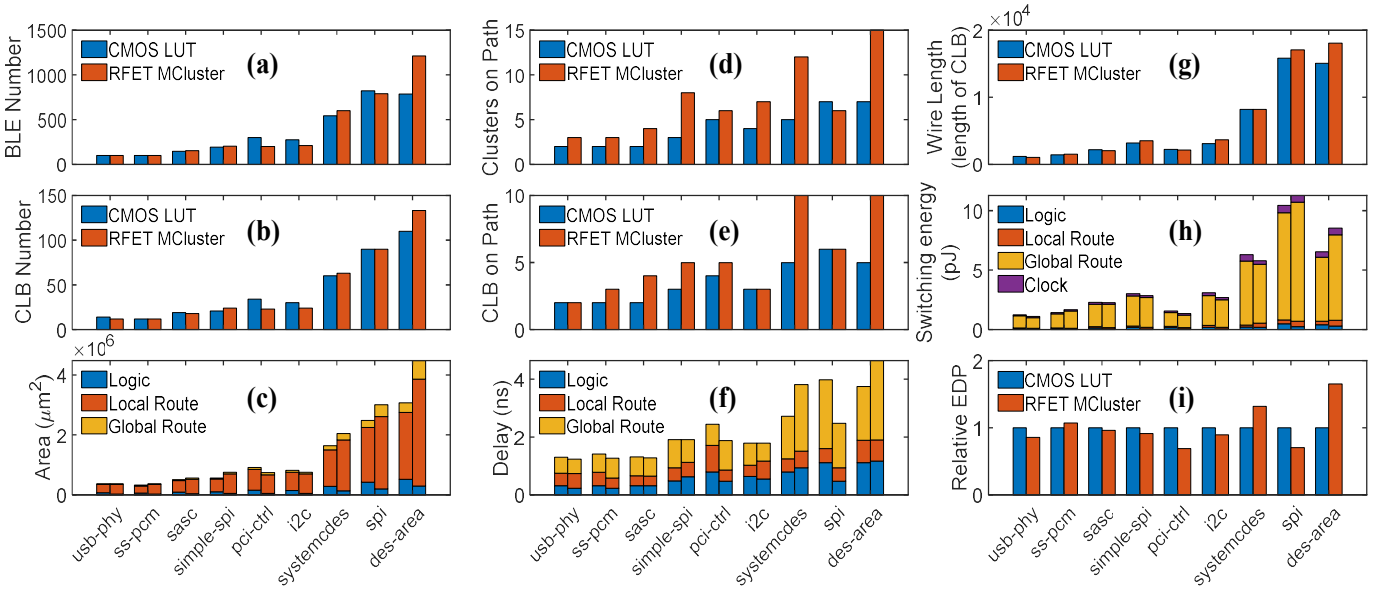


Fig. 4. Comparison of optimized (a) BLE number, (b) CLB number, (c) area, (d)-(e) number of LUTs/clusters and CLBs on critical path, (f) delay, (g) total wire length, (h) switching energy, (i) relative energy-delay-product between FPGAs using proposed RFET-based MCluster and traditional CMOS-based LUT. Here, the system operates under the optimal partition size and percentage of critical path for the first step of the packing algorithm.

In terms of delay, the proposed algorithm prioritizes the packing of logic gates on the critical path so that these gates are assigned to the same or nearby CLBs, thus reducing the delay caused by global routing. In terms of switching energy, the proposed algorithm utilizes the partition process to keep the logical gates adjacent to each other in the original netlist stay close, effectively reducing the total length of the metal wires required for global routing, and ultimately reducing the switching energy.

B. System-Level Performance Comparison

Finally, a comprehensive comparison between FPGAs using RFET-based MCluster and CMOS-based LUT is performed. As shown in Fig. (a)-(c), while RFET has a clear advantage in terms of logic area, it does not improve the overall area because each MCluster has three outputs and requires more resources for local routing. In terms of delay, the RFET MCluster generally has a smaller delay, as shown in Fig. (d)-(f). This is due to the smaller unit delay of the RFET MCluster and the critical-path-based packing algorithm. As mentioned in Section II, partitioning-based packing can effectively reduce switching energy consumption. Fig. (g)-(h) illustrate that MCluster can be just as good as, or even better than, CMOS LUTs in terms of energy efficiency due to the algorithm's optimization. Finally, Fig. (i) shows the system-level performance, indicating that most benchmarks using RFET MCluster provide a better performance, where up to 30% EDP reduction can be observed compared to the CMOS counterparts using 6-input LUT. The results show that some netlists do not experience performance improvements after implementing the RFET MCluster, which is primarily due to the high number of CLBs on their critical path, which will be further optimized in our future work.

Table I
AREA, DELAY, AND ENERGY COMPARISON BETWEEN VPACK-BASED PACKING AND PROPOSED PACKING ALGORITHMS

	Logic Gate Area ($\times 10^8$ F^2)		**Critical Path Delay (ns)**		**Switching Energy (pJ)**	
Netlist	**VPack**	**Proposed**	**VPack**	**Proposed**	**VPack**	**Proposed**
usb-phy	1.4	1.5	1.42	1.24	1.48	1.1
ss-pcm	1.3	1.5	1.39	1.27	1.87	1.69
sasc	2.2	2.3	1.91	1.28	2.85	2.73
simple-spi	2.9	3.1	2.2	1.91	3.55	2.87
pci-ctrl	3.3	3.0	2.35	1.88	1.54	1.34
i2c	3.0	3.1	2.1	1.79	2.98	2.69
systemcdes	10	8.2	3.12	3.81	7.53	5.8
spi	14	12	3.35	2.48	12.97	11.28
des-area	21	18	5.21	4.64	11.34	8.53
Average Improvement	**1.0%**		**13.0%**		**15.8%**	

IV. CONCLUSIONS

In this paper, we design a compact and efficient RFET-based 2-input LUT and organize multiple of these logic cells into regular matrices, known as MClusters, with a fixed interconnection pattern to replace LUTs in the FPGA. A delay-aware packing algorithm is designed for this structure to optimize critical path delay and switching energy. Final benchmark simulation results demonstrate up to 35% delay optimization and 30% EDP optimization at the system level based on RFET MClusters.

REFERENCES

[1] V. Betz *et al.*, *Architecture and CAD for deep-submicron FPGAs*. Springer Science & Business Media, 2012.

[2] J. Trommer *et al.*, "Functionality-enhanced logic gate design enabled by symmetrical reconfigurable silicon nanowire transistors," *IEEE Transactions on Nanotechnology,* vol. 14, no. 4, pp. 689-698, 2015.
[3] M. H. B. Jamaa *et al.*, "FPGA design with double-gate carbon nanotube transistors," *ECS Transactions,* vol. 34, no. 1, p. 1005, 2011.
[4] T. N. Kumar *et al.*, "A novel design of a memristor-based look-up table (LUT) for FPGA," in *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2014: IEEE, pp. 703-706.
[5] S. M. Williams and M. Lin, "Architecture and circuit design of an all-spintronic FPGA," in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2018, pp. 41-50.
[6] P.-E. Gaillardon *et al.*, "A novel FPGA architecture based on ultrafine grain reconfigurable logic cells," *IEEE Transactions on very large scale Integration (VLSI) Systems,* vol. 23, no. 10, pp. 2187-2197, 2014.
[7] K. Cheng *et al.*, "Hybrid topologies for reconfigurable matrices based on nano-grain cells," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, 2017: IEEE, pp. 1-8.
[8] L. Shang *et al.*, "Towards Area Efficient Logic Circuit: Exploring Potential of Reconfigurable Gate by Generic Exact Synthesis," *IEEE Open Journal of the Computer Society,* vol. 4, pp. 50-61, 2023.
[9] T. Mikolajick *et al.*, "Reconfigurable field effect transistors: A technology enablers perspective," *Solid-State Electronics,* vol. 194, p. 108381, 2022.
[10] L. T. Clark *et al.*, "ASAP7: A 7-nm finFET predictive process design kit," *Microelectronics Journal,* vol. 53, pp. 105-115, 2016.
[11] J. N. Quijada *et al.*, "Germanium Nanowire Reconfigurable Transistor Model for Predictive Technology Evaluation," *IEEE transactions on nanotechnology,* vol. 21, pp. 728-736, 2022.
[12] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool," in *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings 22*, 2010: Springer, pp. 24-40.
[13] K. E. Murray *et al.*, "Vtr 8: High-performance cad and customizable fpga architecture modelling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS),* vol. 13, no. 2, pp. 1-55, 2020.
[14] J. Lamoureux and S. J. Wilton, "Activity estimation for field-programmable gate arrays," in *2006 International Conference on Field Programmable Logic and Applications*, 2006: IEEE, pp. 1-8.
[15] G. Karypis and V. Kumar, "METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices," 1997.