2026 IEEE 34th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM) | 979-8-3315-5815-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/FCCM68464.2026.00032

Enabling Context-Switchable Monolithic 3D FPGA Design Using Bistable Ferroelectric Inverters

Faaiq Waqar[1*], Matthew Chen[1], Zifan He[2], Zishen Wan[1,3], Minji Shon[1], Wei-Hsing Huang[1], Jason Cong[2], Shimeng Yu[1*]
[1]Georgia Institute of Technology; [2]University of California - Los Angeles; [3]Harvard University
[*]Email: faaiq.waqar@gatech.edu, shimeng.yu@ece.gatech.edu

***Abstract*—This work proposes a time-multiplexed, non-volatile FPGA design based on back-end-of-line (BEOL) compatible ferroelectric field-effect transistors (FeFETs) for low-power edge acceleration. In battery-constrained microrobotic applications targeting localization and odometry, we show that a power-gateable reconfigurable FPGA can save >50% of on-board resources and improve cumulative localization error, system power, and FPS by 75% over a baseline single-pipeline deployment, and by 25–35% over existing partial-reconfiguration techniques. Using an FeFET as the storage element in a bistable, high-gain inverting topology, the fabric targets ultra-low standby power while improving programmable-mesh density and critical latency. Through integration of bistability using a virtual rail, we demonstrate 3.74–12.5× lower standby power and non-volatility that enables power gating while maintaining sub-ns context-switching time at the cell level. Using the latest Verilog-to-Routing (VTR) flow integrated with experimentally calibrated BSIM+Preisach SPICE models, we evaluate and compare monolithic 3D (M3D) proposals at tile- and system-level integration of a 4-configuration bit-cell on Koios deep-learning benchmarks in the 7 nm technology node with 34% improvement in density over a single-configuration conventional FPGA and 40% improvement in area-delay-squared (AT^2) product.**

I. Introduction

The field-programmable gate array (FPGA) provides a reconfigurable hardware substrate that enables application-specific acceleration without the non-recurring engineering and fabrication costs of fixed-function application-specific integrated circuits (ASIC). This capability is particularly attractive for edge computation (e.g., robotic rovers and drones, software-defined radio networks, and portable multimedia systems) where battery-bound energy budgets and deterministic low latency must be met while algorithms, models, and protocols adapt during/after deployment [1]–[3]. Nevertheless, despite hard macros' integration (e.g., BRAM/URAM and DSP), general-purpose FPGA fabrics typically trail comparable ASIC implementations by roughly 3–6× in performance [4] and 10-40× worse in logic density [5], [6].

To increase the effective logic density mapped to an FPGA, one approach is to time-multiplex the configuration layer across mutually exclusive functions. Existing Xilinx FPGAs support dynamic runtime reconfiguration via the Internal Configuration Access Port (ICAP) [7], which has demonstrated strong utility in real-time applications where environment/prompt dependent algorithmic deployment proves advantageous [8]–[10]. However, partial reconfiguration incurs significant temporal (∼100's of ms) and energy overhead proportional to the reconfigured bitstream size and inversely

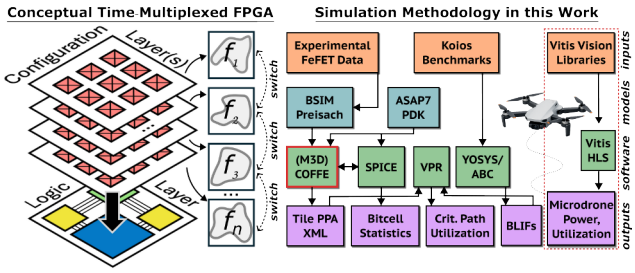

Fig. 1. (a) Conceptual depiction of an n function time-multiplexed FPGA with distinct configuration layers. (b) Experimental methods employed.

proportional to ICAP throughput, and is largely confined to SRAM-configured FPGAs where reprogramming overhead is relatively low but the configuration is volatile (i.e. erased when the power is turned off) [11]. Partial reconfiguration can reduce power when combined with clock gating because (i) bitstream contents can strongly affect standby power [12] and (ii) clock distribution can consume 14–22% of on-chip dynamic power [3]. Yet SRAM volatility limits aggressive fine-grained power gating, which becomes an increasingly critical challenge as FPGA static power reductions stagnate in advanced nodes (e.g., up to 56-58% of total power) [13], [14].

An alternative approach is to store multiple configurations on-chip and switch among n preloaded contexts, where each context implements a function f_i (Fig. 1a). Trimberger et al. first described the micro-architecture and programming model for a conceptual time-multiplexed FPGA [15], [16], but the proposed use of additional configuration storage for each context exacerbates a central FPGA challenge: SRAM-based reconfigurability is area- and energy-expensive. The crux of this challenge stems from the configuration fabric — SRAM cells create a programmability overhead used to emulate logic and routing that would be hardened in an ASIC. In modern FPGAs, configuration memory can dominate on-chip storage, exceeding URAM/BRAM capacity by roughly 4:1 [17].

In this work, we propose a time-multiplexed non-volatile FPGA design based on state-of-the-art amorphous oxide semiconductor (AOS)-channel FeFET bitcells programmed between two threshold-voltage states. By leveraging the FeFET as the storage element (rather than a routing switch [18]), our design targets low standby power while improving density, mitigating pass disturbance, and enhances performance in through a miniaturized programmable fabric. This paper makes the following contributions:

2576-2621/26/$31.00 ©2026 IEEE
DOI 10.1109/FCCM68464.2026.00032

- We study the benefits and set design targets based on a time-multiplexed, power-gatable FPGA for microrobotic autonomy workloads, showing $> 50\%$ on-board resource (e.g., LUTs, BRAMs) savings and 25–35% improvement in cumulative FPS, localization error and power versus existing partial-reconfiguration techniques.
- We propose a BEOL-compatible multi-context FPGA architecture (MCB-FPGA) that uses multi-context FeFET-based configuration inverters (FeInverters) to enable rapid context switching while targeting ultra-low standby power through non-volatility and bistability.
- We develop the multi-context configuration bit-cell requiring $n + 3$ transistors per cell, including operating/voltage-selection criteria that tolerate device variation and enable reliable operation with small FeFET memory windows; we also describe how multi-context arrays can be compactly programmed/selected.
- We provide a variability-aware, cross-layer evaluation flow for an MCB-FPGA in the 7 nm process, demonstrating improvements of density by 34% and deep-learning acceleration AT^2 by up to 59% on Koios benchmarks without utilization of context switching.

II. Background and Prior Work

A. BEOL-Compatible (Ferroelectric) Transistors

Ferroelectric films exhibit a spontaneous polarization that can be reversibly switched by an electric field. In an FeFET (Fig. 2a), the bound charge from out-of-plane polarization P_z shifts the surface potential and thus the threshold voltage: program/erase pulses (V_{PGM}, V_{ERS}) set high-V_t (HVT) and low-V_t (LVT) states, with a memory window $MW \equiv |V_{th} - V_{tl}|$ (Fig. 2b). Early FeFETs used perovskite stacks (e.g., BTO, SBT) but were limited by CMOS compatibility [19]. The discovery of ferroelectricity in orthorhombic hafnia (HfO_2)-based thin films (e.g., HZO) revitalized FeFETs for embedded non-volatile-memory (eNVM) and compute-in-memory [20], [21]. The most mature silicon-proven process is GlobalFoundries' 28 nm silicon-channel FeFET platform [22]. However, HfO_2 polycrystallinity induces grain/domain and phase nonuniformity and charged defects (e.g., electron trapping), leading to intrinsic variability that must be addressed in circuit integration (Section 5) [23], [24].

BEOL integration (i.e., the placement of active devices above CMOS among interconnect layers) requires a limited thermal budget ($< \ 400\,^{\circ}\mathrm{C}$), motivating low-temperature channel deposition of materials such as amorphous oxide semiconductors (AOS). Indium-oxide–based AOS (In_2O_3 family) offer moderate electron mobility ($\sim$10–20 $\mathrm{cm}^2/\mathrm{V}\cdot\mathrm{s}$) and wide bandgaps that suppress leakage; oxygen-affine dopants (Zn, Sn, W) are commonly used to reduce oxygen vacancies and improve V_t stability [25]. AOS-channel BEOL-compatible FeFET prototypes have been demonstrated by academia and industry (e.g., IMEC, Intel, TSMC), reporting logic-compatible voltages, large memory windows, low interfacial trap densities, and strong cycling endurance down to 50 nm channel lengths [26]–[29].

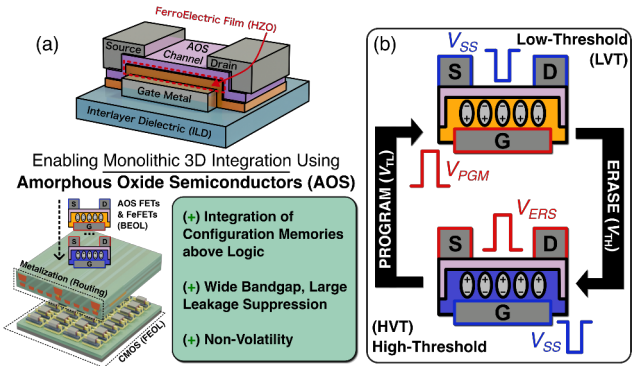

Fig. 2. (a) Structure of a back-gated oxide channel FeFET and posited benefits of M3D configuration memory integration. (b) Switching dynamics of HZO film in FeFET inducing high and low threshold states.

B. Alternative Configuration Memories

Prior works have shown that configuration memory dominates FPGA fabric footprint [30], increasing interconnect length and static power (often dominated by buffer insertion [31]). This becomes more acute at FinFET-era nodes and beyond as wire R/C parasitics worsen through pitch miniaturization and minimum SRAM footprint stagnates [32], [33]. To address this, multiple BEOL-integrable configuration devices have been investigated. Since many two-terminal eNVMs program into high-/low-resistance states (HRS/LRS), a commonly studied topology is a two-device bipolar voltage divider, where the HRS device creates a measurable intermediate node voltage. Implementations have been suggested using RRAM, PCM, and NEM relays [31], [34], [35], though scaling in mature logic platforms remains limited. [36] and [37] instead port an SRAM-like topology into the BEOL using AOS transistors, preserving programming voltages but trading off noise margin and programming speed.

Other works propose eliminating the configuration bitcell + pass-gate combination in routing altogether. Because the SRAM storage node (Q) controls switches by gate voltage, a non-volatile device that directly provides an HRS/LRS conduction path can, in principle, realize a programmable switch without dedicated configuration memory overhead (as explored with RRAM [38], NEM [31], and FeFET-based [39] routing). However, *high programming voltages* often require guard rings or thick-oxide devices, which erode density benefits due to the necessity for tight co-integration with surrounding routing logic [40]. Reducing the programming voltage introduces pass-disturb concerns [37]. Furthermore, matching the drive strength and signal suppression of a CMOS pass gate introduces a sizing–leakage–delay trade-off when substituting eNVM-based switches [38], [39].

C. Time-Multiplexed Memory Implementations

Although the term "time-multiplexed FPGA" is most closely associated with Xilinx's context-switchable FPGA concept, where multiple functions are stored in configuration memory [15], the proposition has peered elsewhere in FPGA micro-architectural research. Examples include "virtual wires"

for overcoming pin limitations in logic emulation [41] and dynamically programmable gate arrays (DPGAs) [42], [43]. The first DPGA implementation suggested reducing configuration overhead using 3T eDRAM cells that refresh a storage node each clock edge, but this incurs high dynamic power [42]. [43] proposed a ferroelectric capacitive divider to emulate a multiplexer between two DPGA configurations; however, the approach is limited to $n = 2$, and how a small capacitive window affects pass-level voltage strength is not rigorously studied. More recently, [18] used the FeFET gate to control parallel switches implementing separate configurations. However, **(1)** as is the case with single-configuration FeFET switches [39] the baseline latency is degraded by the lower current density of FeFETs relative to CMOS, and this penalty increases with the number of configurations, **(2)** the multi-configuration LUT still requires external multiplexing to select among separate LUTs, and **(3)** the study is based on individually measured, large FeFET devices (500 nm $\times$ 500 nm), and therefore does not capture the implications of device-to-device (D2D) variation in scaled FeFETs or the system-level performance impact in large circuits, which we tackle in this work.

III. Motivational Application: MicroRobotics

In the late 80's, Brooks and Flynn suggested that the future of planetary exploration would employ cheap micro-robotic swarms rather than a single expensive multi-objective robot [44]. Three decades later, this vision is increasingly plausible—but energy and mass budgets become progressive tight by way of miniaturization. As an example, in larger drones, propulsion power strongly outpaces all other subsystems, so even sizable improvements in compute efficiency translate to only modest gains in flight time. Conversely, **as platforms shrink to micro- and nano-scale drones, the computational share is first-order**: heavy autonomy workloads can account for ∼10–20% of total power in small drones, where reducing compute energy can yield up to ∼20% longer flight time. [45]. Moreover, the on-board compute modules contribute non-trivial mass/volume to the chassis, increasing thrust and battery requirements.

A major driver of this compute demand is state estimation: autonomous platforms must continuously localize and determine 6 degrees-of-freedom (DoF) pose to enable motion planning, navigation, and stabilization [9]. Critically, however, localization algorithms embody a design space including map-based registration, compute-light visual-inertial odometry (VIO), and simultaneous-localization-and-mapping (SLAM), in which the most suitable method depends on operating conditions such as whether a prior map/GPS exists, the number of obstacles present and the volume of the enclosed space (Fig. 3a), with orders of magnitude ranges in frame operating speeds (FPS) and absolute error [9]. FPGAs offer a practical kernel acceleration platform to reduce the power overhead of heavy workloads like SLAM, translating into meaningful flight-time gains for small drones. This design space and need for reduced computing power has also lead to proposals to leverage partial reconfiguration to reduce standby power during clock

TABLE I
Vitis P&R Resource Utilization and Speed on ZCU104

Kernel Design	LUT	FF	DSP	BRAM	SRL
Kalman Filter (KF)	19795	33242	63	20	1417
ORB-SLAM (SLAM) [46]	168320	90474	180	285	828
Stereo Pipeline (SP)	45649	41389	198	568	1898
Descriptor BF (DBF)	14895	17648	12	171	665
Cumulative Deployment	**248659**	**182753**	**448**	**2163**	**4808**
Time-Muxed FPGA	**168320**	**92479**	**198**	**568**	**4808**
Resources Saved	***32.3%***	***49.4%***	***55.8%***	***73.7%***	***0%***

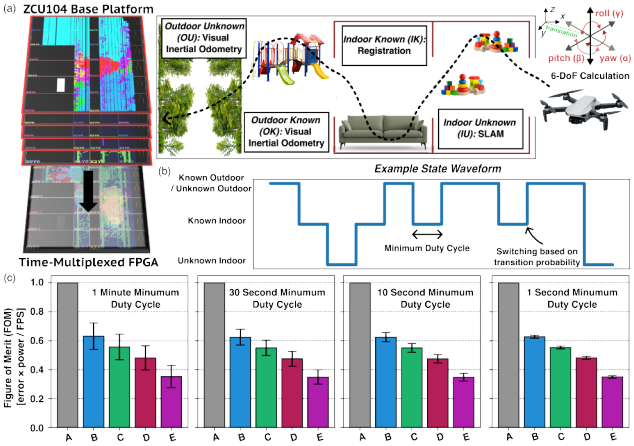

Fig. 3. (a) Mapping of multiple scenario-dependent acceleration functions (VIO, SLAM, Registration) onto test-vehicle ZCU104 in Vitis HLS. (b) example 10-minute waveform employed in case study with demarcation of duty cycle. (c) comparison of application scenarios (A) utilization of a global SLAM pipeline (B) integration of all accelerator functions on FPGA (C) the addition of clock gating (D) addition of partial reconfiguration for static power reduction [47] (E) using a time-muxed substrate with power gating.

gating [1], and the specialized deployment/acceleration of VIO/SLAM kernels [8], [9] if the tradeoffs with reprogramming time and savings are carefully considered [11].

The core family of environment-targeted perception algorithms creates an opportunity: a time-multiplexed FPGA that rapidly switches between algorithms while power-gating inactive blocks could substantially improve area, real-time performance, and power. We therefore start with a study of the benefits of a technology-agnostic time-multiplexed FPGA. Using the Vitis Vision library and an HLS implementation of ORB-SLAM components [46], we map a representative localization suite of accelerator functions (SLAM, Kalman filtering (KF), stereo block matching (SP), and descriptor-based feature matching (DBF)) onto a Xilinx UltraScale+ ZCU104 using Vitis HLS. For each module, we measure static and dynamic power during operation, bitstream size, and FPGA resource utilization; Table I summarizes post-synthesis utilization. In a black-box reconfigurable FPGA, we assume that components of configurable logic blocks (CLBs) such as LUTs and FFs can be functionally context switched while shared macros such as DSP units and BRAMs can be time-shared through the reconfigurable mesh (i.e. switch blocks (SBs) and connection blocks (CBs)), and that the embedded time-multiplexing functionality allows contexts to be switched within a few cycles. Using a

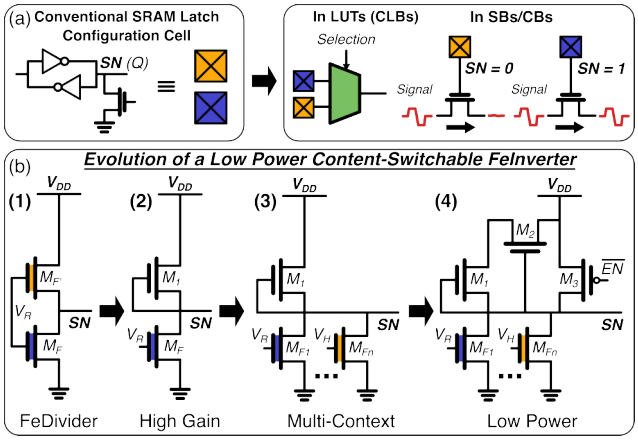

Fig. 4. (a) Application of storage-node (SN) voltage in FPGA circuits (b) Iterative design of a context-switchable FeFET configuration cell in this study, indicating each stepwise addition to enable n + 3 FET bitcell.

TABLE II
BINARY OPERATION OF CONTENT-SWITCHABLE FEINVERTER (3,4)

$V_{G,MF1}$	M_{F1} State	$V_{G,MF2}$	M_{F2} State	SN Voltage (V_{SN})	Config
V_R	HRS (V_{th})	V_H	LRS or HRS	V_{DD} (1)	1
V_R	LRS (V_{tl})	V_H	LRS or HRS	V_{SS} (0)	1
V_H	LRS or HRS	V_R	LRS (V_{tl})	V_{SS} (0)	2
V_H	LRS or HRS	V_R	HRS (V_{th})	V_{DD} (1)	2

black box time-muxed FPGA with $n = 4$ functions, we show that a multi-vision algorithm deployment can be implemented with less than 50% of the on-board resources when the ORB-SLAM pipeline is implemented as a discrete configured logical state, while KF, SP and DBF are implemented as the other.

We then use these Vitis-derived measurements with the analytical model in [3] to compare the product of localization error (m), operation power (W) and seconds-per-frame (FPS^{-1}) in five operation scenarios using a single pipeline, adding clock gating and static power reductions through partial reconfiguration, and finally observing the impact of rapid ICAP-free time-multiplexing and power-gating (Fig. 3b). We assume that the baseline FPGA can achieve effective ICAP throughput of ∼737 MB/s [48], and that the reconfiguration process on a base FPGA consumes ∼200 mW based on measurements in [3]. Then, under random state selection (outdoor, indoor with/without map) with an equal probability of each state (based on state fractions from the EuRoC micro aerial vehicle dataset [49]) under a parameterized duty cycle, we derive figure-of-merit extraction for 100 randomly sampled 10-minute waveforms (Fig. 3c). We observe that the ability to power gate and rapidly reconfigure between accelerator functions improves operation figure-of-merit (FOM) by 75% over the single-pipeline basecase, and ∼25-35% over leading static-power/clock power reduction techniques, especially at short duty cycles where even optimistic ICAP throughput limits lowest seconds-per-frame.

These prior results (e.g., benchmark *E*) establish the system-level requirements that the ideal non-volatile time-multiplexed FPGA substrate must satisfy (i.e., cycle-level reconfiguration overheads, configuration state persistence, minimal alteration to FPGA speed) to obtain the posited benefits in battery-bound edge acceleration. The following section targets engineering a bottom-up AOS-FeFET based implementation, introducing the proposed configuration bit-cell and circuit-level design principles that enable a practical multi-context FPGA.

IV. CELL DESIGN AND GUIDELINES

To design a ferroelectric bitcell suitable as a substitute for SRAM (Fig. 4a) in a time-multiplexed FPGA, we identify several desirable criteria that enable context-switchable design. **(1)** The CMOS footprint of a bitcell should be smaller than that of HD-SRAM without sacrificing operating frequency. **(2)** The device should **deterministically** deliver a pass voltage (V_{DD}) to the reconfigurable mesh under device-to-device (D2D) variation. **(3)** The device should mitigate standby power. **(4)** The cell should support multiple functions with minimal logic overhead and short switching time. In this section, we demonstrate a realized topology that can be integrated within the constraints of BEOL FeFETs reported in the literature.

Analysis is carried out using a piecewise square-law model, from which insights are verified in SPICE simulations built on a FeFET compact model (Preisach+BSIM model) [50] calibrated to GlobalFoundries' 28 nm FeFETs at 500 nm gate length for FEOL devices [18], [22] and TSMC's 50 nm gate length device for BEOL devices [26] to reproduce conditions of prior work [18]. CMOS devices are implemented in a 7 nm predictive process-design-kit (PDK) [51].

A. Depletion-Mode Configuration Inverter

We start by developing a general framework for a compact FeFET-based configuration bit. The design principles below are device-agnostic and apply to any FET technology that can be switched between two quasi-deterministic threshold-voltage states. We consider two basis topologies for a reconfigurable bit: (i) a bipolar ferroelectric divider (FeDivider), reminiscent of previously proposed solutions using eNVMs, and (ii) a static-input depletion-mode ferroelectric inverter (FeInverter), which is a novel contribution of this work.

A bipolar FeDivider uses two FeFETs with opposite polarities (Fig. 4b) driven by a common read voltage V_R. If the bottom device (M_F) is in a low-resistance state (i.e., the V_{tl} state), most of the voltage drops across the top device ($M_{F'}$, passing a '0'); the complementary state passes a '1'. The read voltage selection follows directly from the passable storage-node voltage $V_{SN} = V_R - V_{tl}$. To deliver a strong '1', we require $V_R \geq V_{tl} + V_{DD}$. To avoid activating the high-V_t (HVT) state, we impose $V_R \leq V_{th}$. Together, these constraints require the FeFET memory window (MW) to exceed the desired pass range (V_{DD}).

However, such a large MW becomes increasingly fragile under device-to-device (D2D) variation as the FeFET is miniaturized or undergoes aging (Section 5). To obtain a more feasible operating margin, we turn to a high-gain inverting topology. Fig. 4b and Table II illustrate the proposed FeInverter and its switching behavior, which is a function of the programmed polarization state of M_F and the shared read

level V_R. Here, M_1 is a depletion-mode FET with $V_{t,M1} < 0$ whose gate is tied to the intermediate storage node V_{SN}, and M_F is an FeFET programmable between V_{th} and V_{tl}. Because **(1)** M_1 is self-biased by V_{SN}, **(2)** the inverter is connected to rails $[V_{SS}, V_{DD}]$, and **(3)** $V_{t,M1} < 0$, M_1 remains on over a wide range of V_{SN}; consequently, the switching point is set primarily by the programmed threshold of M_F. This yields an important operating principle: a non-volatile bit that maps the shared read voltage to V_{SN} should use a read level **between the two programmed thresholds**, i.e., $V_R \in (V_{tl}, V_{th})$; Therefore, the robustness of stored state is resolved by the sharp inverter gain, the expressions for which are derived based on the finite transition window in Appendix A.

Fig. 5a plots the resulting requirement versus the transconductance coefficient ratio of the load (β_{M1}) and the state FeFET (β_{MF}), $\beta_R \equiv \beta_{MF}/\beta_{M1}$. This outcome is significant: in the proposed cell, even devices with a small MW can be used without introducing bit errors or degrading performance through reduced switch drive current (Fig. 5b) — an advantage that is not achievable in prior eNVM-based pass-gate replacement solutions (Section 5).

B. Supporting Multiple Configuration Functions

Having established single-function operation, we extend the cell to support n configuration functions by placing multiple FeFET state transistors in parallel. Specifically, we replace M_F with a set $\{M_{F1}, \ldots, M_{Fn}\}$ connected between the storage node V_{SN} and V_{SS}. An n:1 decoder selects one selection FeFET by driving its gate to the shared read level V_R, while all unselected devices are inhibited by a hold bias V_H (Fig. 4b). This design adds only one FeFET per additional function (plus decoder routing), enabling rapid multi-context configuration with a low device overhead. Derivations for V_R and V_H operating ranges, and the resulting impact on programming voltage bounds for the FeFET are expressed in Appendix A.

We introduce two additions that preserve high gain while suppressing the static leakage current by turning off the load branch once V_{SN} is pulled down. **(1)** We add a gating-feedback transistor M_2, **(2)** a PMOS M_3 as a *startup-assist* device using a common enable signal asserted during context switching. As shown in Fig. 5b, the modified low-power cell maintains strong gain (78%) relative to the base depletion inverter baseline.

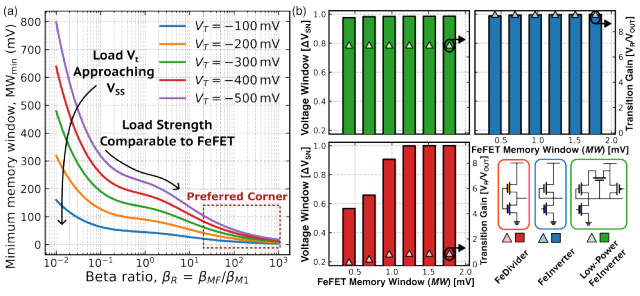

Fig. 5. (a) Minimum MW requirement versus β_R and $V_{t,M1}$ for the proposed FeInverter (analytical model in Appendix A). (b) SPICE simulated comparison of transition gain and maximum voltage window ($\Delta V_{SN} \equiv V_{SN,1} - V_{SN,0}$) between FeDivider and FeInverter topologies.

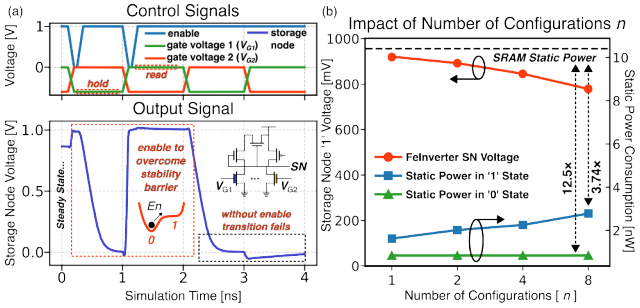

Fig. 6. (a) Switching transients of an $n = 4$ FeInverter switched between two opposite configurations with and without enable startup-assistance. (b) Comparison of FeInverter static power and storage node voltage vs. n

In Fig. 6a, we show transition waveforms for a two-configuration cell storing and switching between complementary binary values. All signal transitions use 100 ps rise/fall times. We apply a 50 ps enable pulse in the first two stages and omit the enable in the latter two. Without the startup-assist PMOS, a $0 \rightarrow 1$ transition is not achievable because the cell cannot reliably exit the localized stable operating region. We also observe that the cell can be context switched reliably within a 1 ns period, thus allowing us to design an FPGA with context-switching that is decoder limited (i.e. capable of single-digit cycle transition). Fig. 6b visualizes static power reduction and maximum '1' voltage (V_{SN}) in the FeInverter as a function of state and the number of configurations.

C. Programming the Grid and Physical Design

The following describes how to program all n-configuration cells. We organize configuration bits into an array with select lines (SLs), bit lines (BLs), and word lines (WLs). Each SL connects to the drain of all gating-feedback (M_2) FETs in a row; each BL connects to the source of all n state FeFETs in a row; and a set of n WLs connects to individual state FeFETs along a column. We define programming as a two-phase process: **(1)** *bulk erase*, where a V_{ERS} operation is applied to all state FeFETs by asserting V_{ERS} on all SLs/BLs while holding all WLs at V_{SS}. To mitigate dielectric stress on the startup-assist device during bulk erase, EN is held at V_{PGM}. **(2)** *row-wise programming*, where a selected WL is asserted to V_{PGM} while the remaining WLs are inhibited using V_{INH} (Fig. 7a-b). Columns targeted for programming place SLs/BLs at V_{SS}, while non-target columns are inhibited. The array programming scheme can be extended to the shift-register based programming scheme presented in Xilinx patents [52] using a set of pull-up/pull-down devices for global erasure in BLs/SLs, and employing thick-oxide devices for the programming decoder if V_{PGM} significantly exceed V_{DD} (Fig. 7c).

Using 7 nm design rules, we evaluate FeInverter layout compactness under three integration options: **(i)** a single FEOL tier (M_3) and a single BEOL tier without P/N active-area overlap, **(ii)** the same two-tier stack with P/N active-area overlap, and **(iii)** a three-tier stack consisting of one FEOL tier plus separate BEOL tiers for NMOS devices and FeFETs (related three-tier M3D organizations have been explored for heterogeneous-channel eDRAM [53] and BEOL SRAM [37]). We scale cell

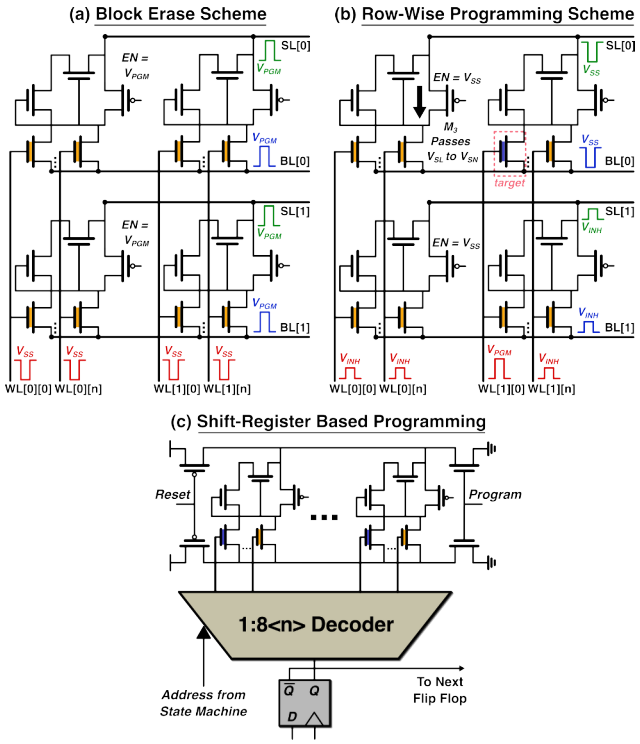

Fig. 7. (a)-(b) Block erase and programming of n function FeInverters in array layout. (c) Adaptation of FeInverter programming based on shift-register programming scheme patented by Xilinx [52].

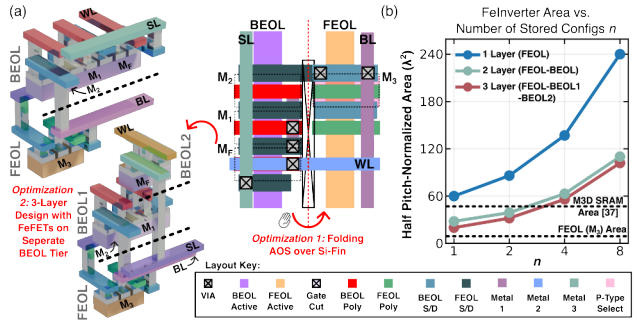

Fig. 8. (a) Layout optimizations of FeInverter in 2 & 3-tier designs (b) FeInverter area scaling as a function of n in 7 nm design rules [32], [51].

area with the number of configurations n using the minimum AOS device dimensions reported in [26] and apply λ^2 scaling based on half-pitch [30]. Under layouts (ii) and (iii), the FeInverter provides 1.68-2.35× higher density than a minimum-sized M3D SRAM baseline [37], and maintains this advantage up to $n = 4$, and additionally enables state non-volatility within 2.7-9.1× lower static power at $V_{DD} = 1$ V (Fig. 8).

V. On Scaled FeFET Circuits

A. The Trouble with (Fe)Pass-Gates

The multi-domain nature of ferroelectric films broadens the distribution of FeFET threshold states as devices are scaled, because individual domains exert a larger influence on the surface potential, particularly in the LVT state (Fig. 9a). As a result, circuit studies should not omit FeFET device-to-device (D2D) variation, especially when conclusions are drawn from relatively large experimental devices [54], [55]. In Section 4, we introduced an integration approach that mitigates the impact of D2D variation by selecting operating voltages and designing a cell with high input–output voltage gain. Here, we evaluate this approach under total reported V_t range and Monte Carlo (MC) simulation using experimental distributions from the literature and compare against FeFET pass-gate routing strategies [18], [29], [39]. We fit a BSIM+Preisach model calibrated to GF's silicon FeFETs with $L_G = 500$ nm for FEOL devices in order to reproduce and compare to the time-multiplexed pass gate strategy demonstrated in [18]. Additionally, we fit a model to TSMC's BEOL-compatible AOS FeFET with $L_G = 50$ nm [26] to demonstrate the scalability of our design using a miniaturized process corner for M3D integration. Sampling is performed using distributions constructed from the reported inter-quartile range (IQR) in each work; the resulting parameter set is summarized in Table III.

Using the reported LVT/HVT states of the $L_G = 50$ nm, $W = 60$ nm AOS FeFET [26], we sweep the memory states to determine a read voltage based on the transition window of the worst-case memory window, and report the storage-node window $\Delta V_{SN} \equiv V_{SN,1} - V_{SN,0}$ for worst-, median-, and best-case devices across $n = 1, 2, 4$ configurations (Fig. 9b). Even under the worst-case MW (∼200 mV), the FeInverter ΔV_{SN} decreases by only ∼120 mV. For comparison, we implement baseline multi-configuration pass-gate multiplexers (2:1 and 4:1) composed of a series selection NMOS and a pass FeFET in a parallel structure, simulated in SPICE using the GF-based Preisach+BSIM model and minimum-size low-power CMOS devices (Fig. 9c). Operating voltages are matched to [18]. For the FeInverter, we first run MC to obtain the distribution of V_{SN} and then apply sampled V_{SN} values in a second-stage MC of the CMOS multiplexer; for FeFET pass gates (Fig. 9d), sampled V_t values are applied directly to the routing devices. All FeFETs are sampled independently, and we consider a worst-case FeInverter condition where the held-state FeFETs store the opposite configuration. Because pass-gate performance is constrained by memory window, we evaluate routing performance using the larger-MW device model, while sampling both scaled and unscaled FeFET distributions.

As shown in Fig. 9e, although the best-case single configuration FeFET pass-gate delay can approach CMOS performance using FeInverters, pass-gate designs exhibit substantially worse distributional behavior, especially under scaled-device variation. Increasing the number of configurations fur-

TABLE III
Literature-Based Monte-Carlo Parameterization

Source	Parameter	Mean (μ)	STD (σ)	Notation
(FEOL) GF; $W = 500$ nm [18]	V_{th}	1V	33mV	Large
(FEOL) GF; $W = 500$ nm [18]	V_{tl}	-50mV	92mV	Large
(BEOL) TSMC; $W = 60$ nm [26]	V_{th}	110mV	62mV	Scaled
(BEOL) TSMC; $W = 60$ nm [26]	V_{tl}	-280mV	147mV	Scaled

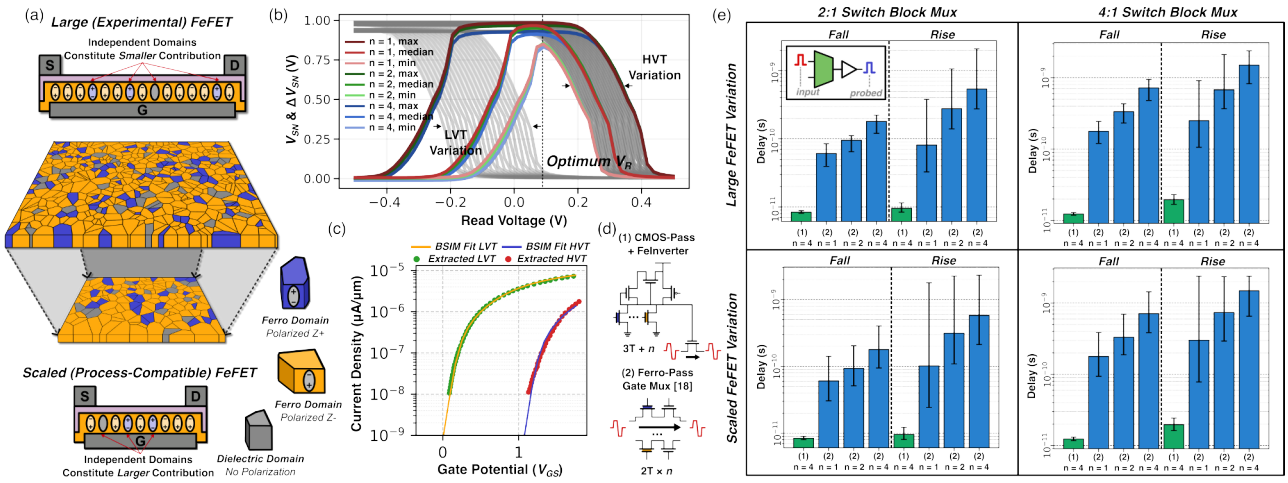

Fig. 9. (a) Effect of domain variation in scaled FeFETs. (b) V_R placement and resultant V_{SN} in scaled AOS FeFET [26]. (c) Calibration of BSIM Preisach model to experimental FEOL and BEOL FeFETs used in simulation. (d) Comparison of time-multiplexed FeFET switch designs in this work and [18] (e) Monte Carlo outcomes on CMOS + FeInverter and FeFET pass gate based multiplexers under scaled gaussian sampling with varying number of configs n.

ther degrades routing speed by orders of magnitude (depicted by the logarithmic scale). While area savings in pass-gate approaches can be traded for larger devices, this reduces density benefits [39] and does not eliminate the performance sensitivity to variation; moreover, for $n \geq 3$ the pass-gate strategy requires more transistors than the FeInverter, which maintains high speed by maintaining the use of CMOS switches.

B. Miniaturizing the Look-up-Table (LUT)

Prior pull-down ferroelectric LUTs can be extended to multi-context operation by switching multiple smaller LUTs using a LUT multiplexer and context decoder [18]. However, replicating the required multiplexing incurs significant overhead (∼58% for an optimized $K = 6$ LUT), which reduces density as the number of contexts n increases. As shown in Fig. 10, based on area-latency product optimized LUT multiplexers using COFFE/HSPICE for $K = 6$ inputs, an FeInverter-based multi-context ($n = 4$) LUT is 3.2× more compact and scales with n at a ∼15× lower rate.

VI. System-Level Performance, Power and Area

A. Benchmarking Methodology

For tile design and evaluation, we use a custom COFFE [56] flow to support M3D placement of configuration bitcells, and we modify the source code to replace ideal pass-gate models with netlisted bitcells in order to capture storage-node disturbance from signal feedthrough and isolate FEOL/BEOL area consumption on a component bases while optimizing total area–delay product. We assume two additional metal layers dedicated to cell-level connectivity, with device placement above the M4 level. [57]. Circuit-level evaluation is performed in Synopsys HSPICE using the ASAP7 PDK for 7 nm projection. For benchmark mapping, we use VTR (VPR, Yosys and ABC) [58] to netlist and map Koios deep-learning benchmarks [59], then apply COFFE-derived architectural parameters to modify a Xilinx-7 style baseline for place-and-route evaluation of area and critical-path latency with parameters $\{K, N, I, W, F_c, F_s\} = \{6, 10, 60, 300, 0.15/0.1, 3\}$ and 2 global routing-metallization layers. NeuroSim-Cache is used to estimate BRAM power, performance and area (PPA) [60].

B. Compositional Comparison

► **Baseline 1: Si SRAM FPGA (Si-FPGA):** The Si-FPGA baseline represents a conventional island-style FPGA tile in which configuration bits are implemented using FEOL SRAM, to which we make area comparisons of each subcircuit in an area-latency product optimized tile in Fig. 11. An SRAM configuration bit cannot be time-multiplexed, thus the

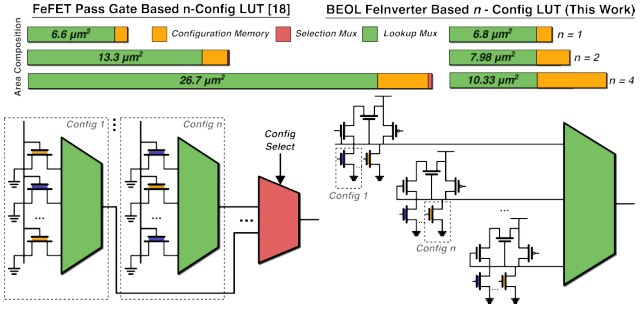

Fig. 10. Compositional comparison of time-multiplexed LUTs vs. n.

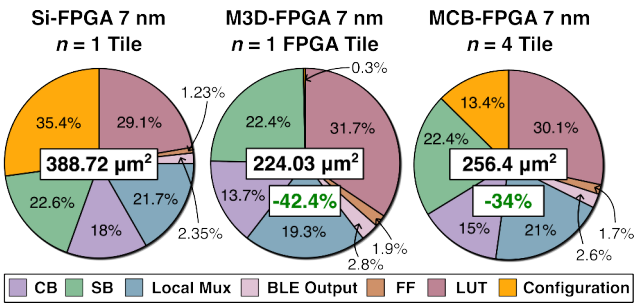

Fig. 11. Compositional comparison of 2D & M3D Tile Implementations

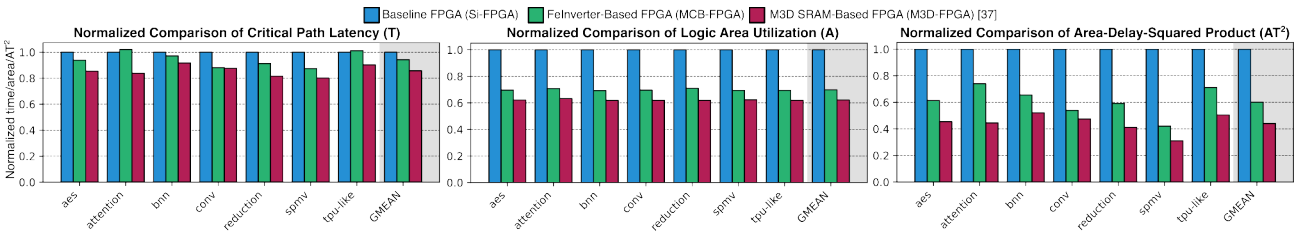

Fig. 12. Area utilization, Critical Path Delay and Area-Delay-Squared product comparison of baseline and M3D FPGA [37] designs on Koios Benchmarks

number of stored functions, unless more SRAM configuration cells are added is $n = 1$. Nevertheless, we observe that a significant fraction of the FPGA tile (35.4%) is consumed by these configuration memories.

► **Baseline 2: M3D SRAM FPGA (M3D-FPGA):** In an M3D-FPGA, the FEOL SRAM configuration memory is substituted with AOS (n-type W-doped In_2O_3 and p-type SnO) based SRAM integrated in the BEOL, modeled with routing and layout overheads derived from [37]. Relative to Si-FPGA, an M3D-FPGA minimizes the overhead of configuration, substantially reducing the total footprint of configuration to interconnect connections (42.4%), which has the additional benefit of reducing global routing R/C. However, the small β ratio in the inverter pair reduces the static noise margin (SNM) of the bitcell, thus relying on PMOS/NMOS stacking to minimize the footprint of the larger PMOS [37]. Like Si-SRAM, M3D-SRAM is not intrinsically time-multiplexable.

► **Proposed: Multi-Context BEOL FPGA (MCB-FPGA):** The MCB-FPGA implements configuration storage using the proposed low-power (LP) FeInverter bit-cell placed in the BEOL, enabling time multiplexing across multiple configurations. With the prior motivation on microrobotics in mind, we employ an FeInverter bit-cell with $n = 4$ as a baseline for this benchmark. The FEOL presence of the startup-assist PMOS and the larger number of transistors required to support four configurations reduce the area savings of the MCB-FPGA scheme by 8.4% compared to the M3D-FPGA; however, employing BEOL placement of LP FeInverters still saves 34% of the area overhead of a conventional Si SRAM-based FPGA while enabling 4× higher effective functional density.

C. On Koios Benchmarks

Using COFFE-derived parameters, we generate representative architectural XMLs and benchmark performance on deep-learning workloads from the Koios suite [59] using VPR [58] to examine a Xilinx-7 architecture island-based FPGA. By only utilizing one of four configurable functions, we show that **adopting the time-multiplexed FeInverter fabric can still improve baseline FPGA performance even for applications that do not explicitly exploit multiple contexts**: the higher tile density reduces global-routing latency, thus significantly improving both density and operating frequency. Figure 12 compares timing, area, and the area–delay-squared (AT^2) product. Consistent with tile-level results, the M3D-FPGA achieves the largest area reduction and therefore delivers the lowest critical-path delay (via reduced global routing) and the best AT^2 (-56%). The MCB-FPGA improves AT^2 by a geometric mean of 40%. Timing improvements are smallest for compute-heavy benchmarks (e.g., *attention layer*), where the slower LUT path (because configuration bits appear on the logic input of an analog multiplexer) dominates over routing-delay reductions; even then, delay increases are limited to $\leq$5% relative to the baseline. Across all benchmarks, MCB-FPGA AT^2 improves significantly (32 - 59 %).

VII. Conclusion

This work presents a time-multiplexed, non-volatile FPGA fabric based on BEOL-compatible FeFET configuration bits for low-power edge acceleration. We employ a bistable, high-gain inverting topology to achieve ultra-low standby power and robust operation under scaling-induced variation, enabling up to 12.5× lower standby power while preserving non-volatility for power gating. Using VTR integrated with experimentally calibrated BSIM+Preisach SPICE models, we show that the multi-configuration MCB-FPGA design using one of four programmable functions can improve deep-learning accelerator deployment AT^2 by up to 59% at the 7 nm node.

Appendix A
Transition Window and Operating Voltages

FeInverter switching behavior is set primarily by M_F. The transition occurs when V_R exceeds $V_{t,MF}$, so a non-volatile bit that maps the shared read voltage to V_{SN} must use a read level between V_{tl} and V_{th}. This mapping is invariant to the value of $V_{t,MF}$. Let the transition window be $[V_{IL}, V_{IH}]$, defined by $\frac{\partial V_{SN}}{\partial V_R} = -1$. Rearranging allows us to derive minimum requirement for MW that prevents aliasing of the two configuration states at V_R:

$$MW > (\frac{2|V_{t,M1}|}{\sqrt{3\beta_R}} - \frac{|V_{t,M1}|}{\sqrt{\beta_R(1+\beta_R)}})_i \forall_i \in FPGA$$

The transition region is finite due to limited gain and must be considered when selecting V_R, which must satisfy:

$$V_R \in (max(V_{IH,LVT}), min(V_{IL,HVT}))$$

When $V_R < V_{t,MF}$, M_F operates in subthreshold, and the strong pull-up to V_{DD} is dominated by M_1. When a third device, M_{F2}, is added in parallel with M_{F1}, The gate of M_{F2} is driven by a hold voltage V_H, which serves as an inhibition signal, which must satisfy:

$$V_H \leq V_{tl,MFi} \forall_i \in FPGA$$

REFERENCES

[1] S. Liu, Z. Wan, B. Yu, and Y. Wang, *Robotic computing on fpgas*. Springer, 2021, vol. 16.

[2] E. J. McDonald, "Runtime fpga partial reconfiguration," in *2008 IEEE Aerospace Conference*. IEEE, 2008, pp. 1–7.

[3] S. Liu, R. N. Pittman, A. Forin, and J.-L. Gaudiot, "Achieving energy efficiency through runtime partial reconfiguration on reconfigurable systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 3, pp. 1–21, 2013.

[4] A. Boutros, S. Yazdanshenas, and V. Betz, "You cannot improve what you do not measure: Fpga vs. asic efficiency gaps for convolutional neural network inference," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 11, no. 3, pp. 1–23, 2018.

[5] V. George, *Low-energy field-programmable gate array*. University of California, Berkeley, 2000.

[6] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," in *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, 2006, pp. 21–30.

[7] K. Vipin and S. A. Fahmy, "A high speed open source controller for fpga partial reconfiguration," in *2012 International Conference on Field-Programmable Technology*. IEEE, 2012, pp. 61–66.

[8] Q. Liu, Z. Wan, B. Yu, W. Liu, S. Liu, and A. Raychowdhury, "An energy-efficient and runtime-reconfigurable fpga-based accelerator for robotic localization systems," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2022, pp. 01–02.

[9] Y. Gan, Y. Bo, B. Tian, L. Xu, W. Hu, S. Liu, Q. Liu, Y. Zhang, J. Tang, and Y. Zhu, "Eudoxus: Characterizing and accelerating localization in autonomous machines industry track paper," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 827–840.

[10] J. Lotze, S. A. Fahmy, J. Noguera, B. Ozgul, L. Doyle, and R. Esser, "Development framework for implementing fpga-based cognitive network nodes," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE, 2009, pp. 1–7.

[11] K. Vipin and S. A. Fahmy, "Fpga dynamic and partial reconfiguration: A survey of architectures, methods, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–39, 2018.

[12] S. Liu, R. N. Pittman, A. Forin, and J.-L. Gaudiot, "On energy efficiency of reconfigurable systems with run-time partial reconfiguration," in *ASAP 2010-21st IEEE International Conference on Application-specific Systems, Architectures and Processors*. IEEE, 2010, pp. 265–272.

[13] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, pp. 1712–1724, 2005.

[14] M. Klein, "Power consumption at 40 and 45 nm," *White Paper*, vol. 298, pp. 1–21, 2009.

[15] S. Trimberger, D. Carberry, A. Johnson, and J. Wong, "A time-multiplexed fpga," in *Proceedings. The 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines Cat. No. 97TB100186)*. IEEE, 1997, pp. 22–28.

[16] S. Trimberger, "Scheduling designs into a time-multiplexed fpga," in *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*, 1998, pp. 153–160.

[17] A. M. Keller and M. J. Wirthlin, "Impact of soft errors on large-scale fpga cloud computing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2019, pp. 272–281.

[18] Y. Xu, Z. Zhao, Y. Xiao, T. Yu, H. Mulaosmanovic, D. Kleimaier, S. Duenkel, S. Beyer, X. Gong, R. Joshi *et al.*, "Ferroelectric fet-based context-switching fpga enabling dynamic reconfiguration for adaptive deep learning machines," *Science Advances*, vol. 10, no. 3, p. eadk1525, 2024.

[19] H. Kohlstedt, Y. Mustafa, A. Gerber, A. Petraru, M. Fitsilis, R. Meyer, U. Böttger, and R. Waser, "Current status and challenges of ferroelectric memory devices," *Microelectronic Engineering*, vol. 80, pp. 296–304, 2005.

[20] T. Böscke, J. Müller, D. Braeuhaus, U. Schröder, and U. Böttger, "Ferroelectricity in hafnium oxide thin films," *Applied Physics Letters*, vol. 99, no. 10, 2011.

[21] H. Mulaosmanovic, E. T. Breyer, S. Dünkel, S. Beyer, T. Mikolajick, and S. Slesazeck, "Ferroelectric field-effect transistors based on hfo2: a review," *Nanotechnology*, vol. 32, no. 50, p. 502002, 2021.

[22] M. Trentzsch, S. Flachowsky, R. a. Richter, J. Paul, B. Reimer, D. Utess, S. Jansen, H. Mulaosmanovic, S. Müller, S. Slesazeck *et al.*, "A 28nm hkmg super low power embedded nvm technology based on ferroelectric fets," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 11–5.

[23] Y. Yoshimura, K. Suzuki, R. Ichihara, K. Sakuma, K. Takahashi, K. Matsuo, M. Fujiwara, and M. Saitoh, "Understanding of polarization reversal and charge trapping under imprint in hfo2-fefet by charge component analysis," *Japanese Journal of Applied Physics*, vol. 63, no. 4, p. 04SP02, 2024.

[24] S. De, M. A. Baig, B.-H. Qiu, F. Müller, H.-H. Le, M. Lederer, T. Kämpfe, T. Ali, P.-J. Sung, C.-J. Su *et al.*, "Random and systematic variation in nanoscale hf0. 5zr0. 5o2 ferroelectric finfets: Physical origin and neuromorphic circuit implications," *Frontiers in Nanotechnology*, vol. 3, p. 826232, 2022.

[25] S. Datta, E. Sarkar, K. Aabrar, S. Deng, J. Shin, A. Raychowdhury, S. Yu, and A. Khan, "Amorphous oxide semiconductors for monolithic 3d integrated circuits," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2024, pp. 1–2.

[26] C.-C. Lu, Y.-C. Shih, C.-Y. Chang, Y.-K. Chang, Y.-C. Chiu, W.-L. Lu, C.-H. Nien, Y.-M. Hsiang, Y.-Y. Cheng, Y.-C. Liu *et al.*, "Demonstration of ferroelectric fet memory with oxide semiconductor channel to achieve smallest cell area 0.009 μm 2 and high endurance for non-volatile high-bandwidth memory applications," in *2024 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2024, pp. 1–4.

[27] Z. Chen, H.-C. Kim, W. Zheng, R. Izmailov, B. Truijen, S. Subhechha, A. M. Walke, A. Chasin, M. I. Popovici, J. Li *et al.*, "Novel design strategy for high-endurance ($> 10^{\{10\}}$) and fast-erase oxide-semiconductor channel fefet," in *2024 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2024, pp. 1–4.

[28] S. G. Kirtania, H. Park, O. Phadke, E. Sarkar, D. Chakraborty, F. G. Waqar, J. Shin, A. Khan, S. Yu, and S. Datta, "Amorphous indium oxide channel fefets with write voltage of 0.9 v and endurance texpreserve0 for refresh-free embedded memory," *IEEE Transactions on Electron Devices*, 2025.

[29] C.-K. Chen, Z. Fang, S. Hooda, M. Lal, U. Chand, Z. Xu, J. Pan, S.-H. Tsai, E. Zamburg, and A. V.-Y. Thean, "First demonstration of ultra-low d it top-gated ferroelectric oxide-semiconductor memtransistor with record performance by channel defect self-compensation effect for beol-compatible non-volatile logic switch," in *2022 International Electron Devices Meeting (IEDM)*. IEEE, 2022, pp. 6–1.

[30] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3d-fpga," in *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, 2006, pp. 113–122.

[31] C. Chen, W. S. Lee, R. Parsa, S. Chong, J. Provine, J. Watt, R. T. Howe, H.-S. P. Wong, and S. Mitra, "Nano-electro-mechanical relays for fpga routing: Experimental demonstration and a design technique," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012, pp. 1361–1366.

[32] S. Nikolić, F. Catthoor, Z. Tőkei, and P. Ienne, "Global is the new local: Fpga architecture at 5nm and beyond," in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2021, pp. 34–44.

[33] C.-H. Chang, V. Chang, K. Pan, K. Lai, J. Lu, J. Ng, C. Chen, B. Wu, C. Lin, C. Liang *et al.*, "Critical process features enabling aggressive contacted gate pitch scaling for 3nm cmos technology and beyond," in *2022 International Electron Devices Meeting (IEDM)*. IEEE, 2022, pp. 27–1.

[34] Y. Y. Liauw, Z. Zhang, W. Kim, A. El Gamal, and S. S. Wong, "Nonvolatile 3d-fpga with monolithically stacked rram-based configuration memory," in *2012 IEEE International Solid-State Circuits Conference*. IEEE, 2012, pp. 406–408.

[35] Y. Chen, J. Zhao, and Y. Xie, "3d-nonfar: Three-dimensional non-volatile fpga architecture using phase change memory," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, 2010, pp. 55–60.

[36] T. Naito, T. Ishida, T. Onoduka, M. Nishigoori, T. Nakayama, Y. Ueno, Y. Ishimoto, A. Suzuki, W. Chung, R. Madurawe *et al.*, "World's first monolithic 3d-fpga with tft sram over 90nm 9 layer cu cmos," in *2010 Symposium on VLSI Technology*. IEEE, 2010, pp. 219–220.

[37] F. Waqar, J. Zhang, A. Lu, Z. He, J. Cong, and S. Yu, "Monolithic 3d fpga design and synthesis with back-end-of-line configuration

memories," in *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2025, pp. 1–7.
[38] J. Cong and B. Xiao, "Fpga-rpi: A novel fpga architecture with rram-based programmable interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 864–877, 2013.
[39] X. Chen, K. Ni, M. T. Niemier, Y. Han, S. Datta, and X. S. Hu, "Power and area efficient fpga building blocks based on ferroelectric fets," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1780–1793, 2018.
[40] C.-T. Dai, M.-D. Ker, Y.-N. Jou, S.-C. Huang, G.-L. Lin, and J.-H. Lee, "Study on latchup path between hv-ldmos and lv-cmos in a 0.16-μm 30-v/1.8-v bcd technology," in *2018 40th Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*. IEEE, 2018, pp. 1–6.
[41] J. Babb, R. Tessier, M. Dahl, S. Z. Hanono, D. M. Hoki, and A. Agarwal, "Logic emulation with virtual wires," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 6, pp. 609–626, 2002.
[42] I. Eslick, D. Chen, E. Tau, J. B. E. Mirsky, and A. DeHon, "A first generation dpga implementation," in *FPD'95–Third Canadian Workshop of Field-Programmable Devices*, 1995.
[43] W. Chong, S. Ogata, M. Hariyama, and M. Kameyama, "Architecture of a multi-context fpga using reconfigurable context memory," in *19th IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2005, pp. 7–pp.
[44] R. BROOKS and A. FLYNN, "Fast, cheap and out of control: A robot invasion of the solar system. s. 478-485," *Journal of the British Interplanetary Society*, vol. 42, 1998.
[45] R. Hadidi, B. Asgari, S. Jijina, A. Amyette, N. Shoghi, and H. Kim, "Quantifying the design-space tradeoffs in autonomous drones," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021, pp. 661–673.
[46] C. Wang, Y. Liu, K. Zuo, J. Tong, Y. Ding, and P. Ren, "ac2slam: Fpga accelerated high-accuracy slam with heapsort and parallel keypoint extractor," in *2021 International Conference on Field-Programmable Technology (ICFPT)*, 2021, pp. 1–9.
[47] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for fpgas," in *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, 2004, pp. 33–41.
[48] A. R. Bucknall and S. A. Fahmy, "Zypr: End-to-end build tool and runtime manager for partial reconfiguration of fpga socs at the edge," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 16, no. 3, pp. 1–33, 2023.
[49] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
[50] C.-T. Tung, G. Pahwa, S. Salahuddin, and C. Hu, "A compact model of ferroelectric field-effect transistor," *IEEE Electron Device Letters*, vol. 43, no. 8, pp. 1363–1366, 2022.
[51] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "Asap7: A 7-nm finfet predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.
[52] P. Rau, A. V. Ghia, and S. M. Menon, "Configuration memory architecture for fpga," Apr. 24 2001, uS Patent 6,222,757.
[53] D. Kong, S. Prakash, J. Kufel, G. Kyriazidis, Y. Omri, D. Verity, E. Ozer, V. J. Reddi, and G. Hills, "333-edram-3t embedded dram leveraging monolithic 3d integration of 3 transistor types: Igzo, carbon nanotube and silicon fets," in *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2025, pp. 1–7.
[54] E. T. Breyer, H. Mulaosmanovic, J. Trommer, T. Melde, S. Dünkel, M. Trentzsch, S. Beyer, S. Slesazeck, and T. Mikolajick, "Compact fefet circuit building blocks for fast and efficient nonvolatile logic-in-memory," *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 748–756, 2020.
[55] O. Phadke, M. Shon, S. Wodzro, H. Mulaosmanovic, S. Dünkel, A. Khan, S. Datta, and S. Yu, "On the localized ferroelectric phase variation in scaled fefet: Experiments and modeling," in *2025 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2025, pp. 23–5.
[56] S. Yazdanshenas and V. Betz, "Coffe 2: Automatic modelling and optimization of complex and heterogeneous fpga architectures," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 1, pp. 1–27, 2019.
[57] Y.-J. Lee, P. Morrow, and S. K. Lim, "Ultra high density logic designs using transistor-level monolithic 3d integration," in *Proceedings of the International Conference on Computer-Aided Design*, 2012, pp. 539–546.
[58] M. A. Elgammal, A. Mohaghegh, S. G. Shahrouz, F. Mahmoudi, F. Koşar, K. Talaei, J. Fife, D. Khadivi, K. Murray, A. Boutros *et al.*, "Vtr 9: Open-source cad for fabric and beyond fpga architecture exploration," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 18, no. 3, pp. 1–53, 2025.
[59] A. Arora, A. Boutros, S. A. Damghani, K. Mathur, V. Mohanty, T. Anand, M. A. Elgammal, K. B. Kent, V. Betz, and L. K. John, "Koios 2.0: Open-source deep learning benchmarks for fpga architecture and cad research," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 11, pp. 3895–3909, 2023.
[60] F. Waqar, J. Kwak, J. Lee, O. Phadke, M. Shon, M. Gholamrezaei, K. Skadron, and S. Yu, "Optimization and benchmarking of monolithically stackable gain cell memory for last-level cache," *IEEE Transactions on Computers*, 2025.