Ferroelectric FET-based context-switching FPGA enabling dynamic reconfiguration for adaptive deep learning machines

Yixin Xu1†, Zijian Zhao1‡, Yi Xiao1†, Tongguang Yu1, Halid Mulaosmanovic2, Dominik Kleimaier3, Stefan Duenkel3, Sven Beyer3, Zhao Gong4, Rajiv Joshi5, Xiaobo Hu2, Shixian Wen6,7, Amanda Sofie Rios6‡, Kiran Lekkala6, Laurent Itti6, Eric Homan1, Sumitha George8, Vijaykrishnan Narayanan1, Kai Ni9‡

Field programmable gate array (FPGA) is widely used in the acceleration of deep learning applications because of its reconfigurability, flexibility, and fast time-to-market. However, conventional FPGA suffers from the trade-off between chip area and reconfiguration latency, making efficient FPGA accelerations that require switching between multiple configurations still elusive. Here, we propose a ferroelectric field-effect transistor (FeFET)–based context-switching FPGA supporting dynamic reconfiguration to break this trade-off, enabling loading of arbitrary configuration without interrupting the active configuration execution. Leveraging the intrinsic structure and nonvolatility of FeFETs, compact FPGA primitives are proposed and experimentally verified. The evaluation results show our design shows a 63.0%/74.7% reduction in a look-up table (LUT)/connection block (CB) area and 82.7%/53.6% reduction in CB/switch box power consumption with a minimal penalty in the critical path delay (9.6%). Besides, our design yields significant time savings by 78.7 and 20.3% on average for context-switching and dynamic reconfiguration applications, respectively.

INTRODUCTION

Deep neural networks (DNNs) have dominated artificial intelligence (AI) applications due to their cutting-edge performance in a wide range of applications in many domains, such as image classification (1, 2), object detection (3, 4), and natural language processing (5, 6). However, with more sophisticated models and more voluminous data to process (7), these DNN workloads are becoming more compute-intensive and data-intensive, requiring hardware accelerators to achieve lower latency, higher throughput, and higher energy efficiency. Field programmable gate array (FPGA) devices, with the capabilities of flexible reconfiguration for arbitrary logic functions while maintaining high performance, are gaining popularity as accelerators for such complex deep learning applications (8–10). The reconfigurability of FPGA is enabled by its unique architecture, as illustrated in Fig. 1A, which consists of a sea of configuration logic blocks (CLBs), connection blocks (CBs), switch boxes (SBs), configuration memory, and I/O blocks (11). In particular, CLBs are the main components that can be programmed to perform different logic operations and CBs and SBs are controlled by configuration bits loaded from the configuration memory. A variety of routing networks can be achieved by loading different configuration bits. Above all, FPGA's aforementioned properties including reconfigurability, flexibility, high performance, and fast time-to-market make it a promising choice for DNN accelerators. The basic structure of FPGA and mechanisms of primitives are shown in fig. S1.

As a concrete and highly important example of DNN acceleration on FPGA, a two-stage super-sub network is adopted for image classification (12). In this model, a superclass is first inferred using a generalist superclass-level network, and the network output is then passed to a specialized network for final subclass-level classification. In this way, the overall classification accuracy has been proved to increase over that of common inference methods when evaluating the “Superclassing ImageNet dataset”, which is a subset of ImageNet (13) and consists of 10 superclasses, each containing 7 to 116 related subclasses (e.g., 52 bird types and 116 dog types) (12). Figure 1D shows one specific example of this framework. In the first stage, the superclass "Dog" is identified by the generalist superclass network. Then, fine-grain inference in the subclass network is performed in the second stage and outputs the final result “husky” of the target image.

Numerous hardware accelerators have been proposed to implement DNNs, such as customized application-specific integrated circuits (14–16), application-driven optimization on graphics processing units (17, 18), and FPGAs (19, 20). However, among these various types of DNN accelerators, FPGA, which can provide more flexibility while maintaining high performance, is particularly suitable for implementing the accelerators of DNNs such as the super-sub network model. Figure 1E shows two main approaches when considering implementing this super-sub network into FPGA. One distinguished feature of the implementation is the requirement of multiple configurations in FPGA to map the superclass and subnetworks, respectively. The straightforward approach is to use more than one chip to process different networks (i.e., configurations). As shown in Fig. 1E, chip 1 is configured to process the general inference task for superclasses, whose outputs are then sent to chip 2 which is configured to map the subclass.
networks to identify the specific subclass. This approach, although fast, incurs penalties in chip area and cost. Another compact and cost-efficient approach is to leverage the reconfiguration capability of FPGA by simply reconfiguring chip 1 to the subclass network after it finishes execution of the superclass network. In this way, contexts, i.e., FPGA configurations, can be swapped in or out of the FPGA upon the demands of application requirements without the need for additional chips (21). Therefore, this approach saves the area cost but comes with a penalty in the reconfiguration latency. Above all, although FPGA offers an attractive choice for the acceleration of the super-sub network model (Fig. 1E), an ideal implementation with high area efficiency and low latency is still elusive with current FPGA technologies and architectures.

Fig. 1. Overview of the proposed context-switching FPGA and potential applications. Architectures of (A) a conventional static random-access memory (SRAM)-based FPGA, (B) SRAM-based FPGA supporting partial reconfiguration, and (C) Our proposed ferroelectric field-effect transistor (FeFET)-based context-switching FPGA supporting dynamic reconfiguration. (D) An example of a deep learning network: two-stage super-sub network (12), where at first the superclass Dog is identified and then the subclass “husky” is identified. (E) Conventional FPGA incurs area overhead or significant reconfiguration latency. This figure shows two main approaches to implementing the super-sub network in conventional FPGA. (F) Our approach provides fast reconfiguration speed and compact solutions. LUT, look-up table.
Many relevant works have explored design options to address the aforementioned issues at different granularities of reconfiguration and from different angles of applications. However, all of them are still limited by the dilemma or might incur other overheads. For example, a full context-switching FPGA was first proposed as a time-multiplexer FPGA based on the Xilinx XC4000E FPGA in 1997 (22), where eight configurations of the FPGA are stored in on-chip memory and the contexts can be switched in a single cycle. With preloaded contexts, reconfiguration is not needed but it comes with a large area penalty. The more configurations to be supported, the more area overhead to store those configurations. To save area while still speeding up the reconfiguration process, dynamic partial reconfiguration appears as another solution to support multiple configurations, by which only a portion of the hardware region (called the reconfigurable region) can be configured while the remainder is static (23). Partial reconfiguration brings several advantages over conventional context-switching FPGA (24), including less reconfiguration time compared to full-region reconfiguration and a smaller area with its increased logic density. However, partial reconfiguration only provides a compromised solution between the area cost and the reconfiguration latency, incapable of fundamentally solving the problem. In the end, it is possible to support fine-grain reconfiguration at the bit level, as demonstrated by consecutive works on the “NATURE” FPGA architecture to support fine-grain temporal logic folding (25, 26), which is either based on complementary metal-oxide semiconductor (CMOS) [e.g., logic and static random-access memory (SRAM)] and carbon nanotube random-access memory (NRAM) (25) or based entirely on CMOS circuits (26). In the former work, NRAM and SRAM work together to support dynamic reconfiguration for temporal logic folding of circuits, which is to realize different logic functions in the same logic elements through dynamic reconfiguration every few cycles, thereby significantly increasing the logic density. In the latter work, the dynamic reconfiguration delay is hidden behind the computation delay through the use of shadow SRAM cells (i.e., two SRAM copies). However, both works suffer from high area costs which are mainly caused by extra NRAM cells and 10 T-SRAM cells, respectively. Therefore, to date, a context-switching FPGA that can break the trade-off between the area cost and the reconfiguration latency remains elusive and the goal of the proposed research is to bridge the gap.

To mitigate the aforementioned issues in terms of area, latency, and power, we propose a dynamic context-switching FPGA architecture based on ferroelectric field-effect transistors (FeFETs) which can implement DNN accelerators more efficiently. With joint innovations from technology, circuit, and architecture levels, our proposed design has several advantages over prior context-switching works. First, from a technology perspective, FeFET is unique in that it behaves both as a transistor switch and a nonvolatile memory cell such that FPGA basic logic circuits [e.g., look-up tables (LUTs)] and routing elements (e.g., CBs and SBs) can be implemented compactly. Moreover, these FPGA basic elements have no leakage power dissipation because of the nonvolatility of FeFET, which hugely reduces the total power consumption of the entire FPGA. Second, from the circuit’s perspective, a CB composed of two parallel branches is proposed, which stores two configurations while still consuming much less area than a single-configuration SRAM-based CB. Third, the proposed FPGA is dynamically reconfigurable with the capability to load one configuration without interrupting the execution of another configuration. As a result, the reconfiguration time can be completely hidden as long as it is smaller than the computation time of the current active configuration. Therefore, our proposed solution can achieve dynamic context-switching with zero penalty in reconfiguration latency and significant area reduction compared to SRAM-based design, breaking the trade-off between area cost and reconfiguration latency the existed in conventional CMOS implementations.

With the proposed context-switching FPGA, the aforementioned super-sub network can be efficiently implemented, as shown in Fig. 1E. Considering one case in which we are interested in having an accurate classification of one specific superclass (e.g., Dog), the proposed design can perfectly fit in it and reduce the reconfiguration latency. Specifically, these two configurations including the superclass network and subclass network can be preloaded into the FPGA. First, the general inference with the superclass network is performed. As long as the output of the general inference is Dog, the configuration corresponding to Dog’s subclass network would be activated and executed for further inference. In this way, compared to long reconfiguration time, the switching time is much less or even negligible, which leads to almost zero latency overhead. In addition, the total area cost could also be heavily reduced by leveraging dense FeFETs. Note that the proposed context-switching FPGA enables applications in various domains that need switching between different contexts, beyond the super-sub network discussed here. The reconfiguration functionality is especially helpful in various dynamic adaptation applications such as changing communication encoders or decoders on demand to the appropriate protocols (27), changing the data rates to vary bandwidths (28), scaling the computation based on available energy needs (29). Moreover, with no limitation on the number of configurations, our design can also be scaled to implement multiple configurations depending on the demand of applications. Some potential applications are illustrated in fig. S3.

RESULTS

Overview of the proposed FPGA architecture

For a deeper look into the design of the proposed context-switching FPGA, details of the architecture and components to support multiple configurations are shown in Fig. 2. Figure 2A shows primitive components of the proposed context-switching FPGA which supports dual configurations, including CLBs, CBs, and SBs. For each component, it is controlled by the configuration information stored in configuration memory. By loading the configuration bits, the logic (LUT) and routing elements (CB/SB) can be connected to form a functional circuit to perform the desired computation. In the proposed context-switching FPGA, there are two local copies of each LUT, CB, and SB, which correspond to two configurations. In this way, when one configuration is active for computation, any other configuration can be loaded without interrupting the execution, thereby significantly reducing the reconfiguration latency. In contrast, in conventional context-switching FPGA, they would either require hardware resources for supporting multiple configurations on-chip or require long serial reconfiguration time. To support run-time reconfiguration and reduce the area cost incurred by the need for an extra copy of FPGA primitive components, FeFET technology, due to its programmability, nonvolatility, and compactness, is chosen in this work to implement basic programmable FPGA components such as LUTs, CBs, and SBs.
In recent years, the switches in FPGA can be realized with various embedded memory technologies as the basic elements of routing elements (CBs and SBs). Figure 2B presents existing mainstream memory technology–based single-configuration switches including SRAM, spin transfer torque magnetic RAM (STT-MRAM), flash memory, resistive RAM (ReRAM), phase change memory (PCM), and FeFET. Because of its logic compatibility, superior write and read performance, and excellent reliability, SRAM is the most straightforward memory to use by combining an SRAM cell with an n-type pass transistor. However, SRAM-based switches suffer from two crucial overheads. One is low area density due to its complex cell structure; the other is high leakage power, which accounts for 60 to 70%
of total FPGA power dissipation due to long routing tracks (30–33). Recently, emerging embedded nonvolatile memory technologies have been actively investigated as promising alternatives to SRAM due to their density, energy, and performance advantages. However, each of them comes with its own challenges. For example, a flash memory–based switch is nonvolatile and compact (34), but memory programming is slow (in approximately milliseconds) and requires a high programming voltage (~10 V). Two terminal resistive memories, including ReRAM, PCM, and STT-MRAM, are nonvolatile and dense but usually require large conduction current to program the devices, consuming a significant write power. In addition, the limited ON/OFF resistance ratio (~100 for ReRAM/PCM and ~5 for STT-MRAM) usually requires additional circuitry, such as the 1T2R structure for ReRAM/PCM (35, 36) and an even more complex supporting structure for STT-MRAM (37) to realize a single switch.

In this regard, we propose the FPGA architecture which adopts FeFETs to implement logic and routing elements. Ever since the discovery of ferroelectricity in doped HfO2, significant progress has been made in the integration of HfO2-based FeFET due to its nonvolatility, high density, large ON/OFF ratio, and excellent CMOS compatibility (38, 39). In addition, switching of ferroelectric polarization is induced by an applied electric field, rather than a large conduction current, making FeFET a highly energy-efficient nonvolatile memory (40). Since the ferroelectric film is integrated into the gate stack of a FeFET, when its polarization is set to point at the channel/metal gate, the FeFET threshold voltage (V_{TH}) will be programmed to the low-V_{TH}/high-V_{TH}, respectively, thus realizing a compact nonvolatile routing element (41). Leveraging this technology, a mixed FeFET/CMOS switch unit (i.e., 1T-1FeFET) has been proposed as a routing element in FPGA (42), which takes advantage of but does not fully exploit FeFET. In this work, leveraging the intrinsic nonvolatile switch structure of FeFET, we propose a 1FeFET routing switch for single-configuration FPGA and a 2T-2FeFET routing switch for dynamic reconfiguration context-switching FPGA, as shown in Fig. 2C, which achieve optimal area efficiency. The critical design difference in our FeFET switch compared to the flash switch and prior FeFET switch (42), despite their similarities in the device structure, is that our switch is composed of only one FeFET, which significantly improves the integration density. The flash switch requires a pair of n-type and p-type flash devices controlling one normal n-type metal-oxide semiconductor (NOMOS) pass transistor. By applying proper biases on the word line (WL) and bit line (BL), only one of the flash devices would be conducted to turn ON/OFF the pass transistor. The reason why it cannot be replaced with one flash transistor might be its relatively poor pass gate performance due to its thick gate stack. Compared to flash devices, FeFET shows great scalability and compatibility with Si CMOS, making a single FeFET feasible as a one-pass transistor. Moreover, FeFET allows lower operation voltages for both writes and reads. Besides, for the 1T-1FeFET switch (42), in addition to FeFET, they need an access transistor to coordinate with operation and programming. However, in our FeFET switch design, we leverage a novel program mechanism to write through gate and body terminals and program disturb inhibition schemes (43). In this way, our design can eliminate the access transistor at a lower area cost. For the context-switching FPGA, a serial CMOS transistor is added to each branch, which is used to cut off the branch that is loading a new configuration to minimize the disturbance to the other active branch. Figure 2D shows our proposed circuit of LUT array for dual configuration. A compact LUT cell can be efficiently implemented using a single FeFET such that the high-V_{TH}/low-V_{TH} states (HVT/LVT) of FeFET store bit “1”/“0” for the LUT cell, respectively. Besides, as shown in Fig. 2D, the proposed LUT can support dynamic reconfiguration—when the branch of configuration 1 is operating, the branch of configuration 2 can load a new configuration.

**Block design and functional verification**

In this section, experimental verification of the proposed LUT and routing elements (CB/SB) for context-switching FPGA is performed. For experimental demonstration, FeFET devices integrated into the 28-nm high-k metal gate (HKMG) technology are tested. Figure 3 (A and B) shows the transmission electron microscopy (TEM) and schematic cross section of the device, respectively. The device features an 8-nm-thick doped HfO2 as the ferroelectric layer and around 1-nm SiO2 as the interlayer in the gate stack. The FeFET memory performance is characterized by standard pulsed I_{p}-V_{G} measurements after applying ±4 V, 1-μs write pulses on the gate. Figure 3C shows a memory window of about 1.2 V, i.e., the V_{TH} separation between the low-V_{TH} and high-V_{TH} states, which enables a large ON/OFF conductance ratio. It also exhibits a well-tempered cycle-to-cycle variation. Figure 3D shows the switching dynamics of the FeFET under different pulse amplitudes and pulse widths, which also shows a trade-off between the write speed and pulse amplitude and that it is possible to program FeFET with sub-10 ns with 4-V write amplitude. It follows the classic nucleation-limited switching model in the thin film polycrystalline HfO2 (38, 44), where domain switching is mainly limited by the nucleation process and the nucleation time follows an exponential dependence on the applied electric field. These results suggest that HfO2-based FeFET exhibits high performance, showing great promise of this technology in many applications including the context-switching FPGA in this work. The endurance and retention characteristics of FeFETs are measured in fig. S2.

Figure 3 (E and F) shows the operation principle of our proposed LUT cells that store bits 1 and 0, respectively. Each cell consists of one single FeFET and one p-channel metal-oxide semiconductor (PMOS) transistor, where the PMOS is shared among all the cells and is part of the sense amplifier used to convert the read current to logic voltage levels. Bits 1 and 0 are stored by programming the FeFET into the high-V_{TH} and low-V_{TH} states, respectively. Then, in the LUT read mode, the stored bit can be read by asserting the appropriate read voltage, V_{READ}, to the gate terminal of the FeFET, as shown in Fig. 3E. Because of the large ON/OFF resistance ratio of FeFET at V_{READ}, the output voltage will be close to V_{DD} and ground for bits 1 and 0, respectively. This is achieved by choosing an appropriate PMOS gate bias (V_{B}) such that its resistance is between the FeFET high-V_{TH} and low-V_{TH} states, thereby setting the output voltage rail-to-rail. Figure 3G demonstrates the main structure of the single-configuration LUT integrated with 2^k FeFET-based bit cells (cell 0/1 cell), different logic functions can be successfully achieved by applying different combinations of select signals. In this structure, a sense amplifier composed of one pull-up PMOS transistor and two inverters is used for converting FeFET read current to voltage and amplifying the output voltage to full swing. The LUT cell operation is then verified in an experiment using the setup shown in Fig. 3H, which includes the major components in Fig. 3G.
Fig. 3. Experimental verification of the LUT cell operation. (A and B) TEM and schematic cross section. (C and D) $I_D - V_G$ characteristics for FeFET measured after ±4 V, 1-µs write pulses and the switching dynamics of FeFET under different pulse amplitudes and pulse widths. (E and F) Operations of the LUT cell for storage with bit “0”/“1” by exploiting the dynamic high-V TH/low-V TH states (LVT/HVT) programming capability. (G) Proposed k-bit LUT. (H) The experimental setup of functional verification of the LUT cell operation. (I) Experimental waveforms of proposed LUT cells in (E) and (F). (J) The circuitry of a LUT array for multiple configurations.
operation waveforms are presented in Fig. 3I, which shows the write and read phases of the LUT cell. After programming the FeFET into high-$V_{TH}$/low-$V_{TH}$ states using $-4 V$/+4 V, 1-$\mu$s write pulse, the output voltage shows a logic high and low, respectively. This verifies the successful cell operation, but due to the discrete experimental setup, performance is limited by the parasitics. To predict the fully integrated FeFET LUT performance, SPICE simulations using a calibrated FeFET model (45) and 45-nm predictive technology model (PTM) for logic transistors (46) are performed. Figure S4 shows the simulated waveform, indicating that for a six-input LUT cell, the read delay is 124.3 ps and consumes 13.1-$\mu$W power. In the subsequent section, FeFET-based primitive components, including LUTs, CBs, and SBs, are also compared with other technology implementations using consistent SPICE simulations, as will be studied in Fig. 5.

To support dynamic reconfiguration, two LUTs forming an array are designed and an additional multiplexer is used to select which configuration should be active in the current operating period, as shown in Fig. 3J. Programming in a bulk planar single FeFET array has been extensively investigated (43, 47). The applicable programming schemes depend on the number of accessible terminals during memory writing. In the proposed FPGA architecture, the source/drain terminals are not simultaneously accessible from outside, which limits the possibility of applying write schemes that need to apply the source/drain voltages. In this case, a convenient solution is shown in Fig. 3J, where the gate and the body terminals are used for programming. The WL is shared among all FeFETs in a configuration block and the body is shared across different configuration blocks. Two-step programming will then be performed where all the FeFETs in a configuration are set to the low-$V_{TH}$ states first by applying a positive write voltage (i.e., $V_W$) on the WL and keeping all the other terminals grounded. Then, those FeFETs that need to be in the high-$V_{TH}$ states are applied with a negative gate-to-body voltage (i.e., $-V_W$). To avoid write disturb to those low-$V_{TH}$ states FeFETs during the second step, the standard inhibition bias scheme (e.g., $V_{W/2}$) can be applied, which is verified in Fig. S5.

Next, the functionality of the routing elements is verified, as shown in Fig. 4. Using CB as an example, Fig. 4A shows the array structure, where BL and source line (SL) route the actual signal, and WL and the column-wise body contact are used to program FeFETs. As introduced in Fig. 2C, to support the run-time reconfiguration of one branch without interrupting the normal operation of the other branch, a serial transistor is added to each branch and is off/on during configuration loading/executing, respectively. The swap between configurations can be easily and swiftly conducted by applying corresponding read gate biases, as shown in Fig. 4B, such that when one configuration is deactivated, the FeFET will be cut off, irrespective of its state. Figure 4C shows an example waveform applied on a testing unit (Fig. 4D), where branch 1 is first configured to be the low-$V_{TH}$ state while branch 2 is executed, and then branch 1 is activated while branch 2 is configured to the high-$V_{TH}$ state using the two-step programming. Figure 4E shows the experimental results applied to the voltage sequence shown in Fig. 4C for three repeated cycles. The zoomed-in programming waveforms for branches 1 and 2 are shown in Fig. 2 (F and G, respectively). Because of the configurations used in this testing scenario, where the branch 1/branch 2 is in the low-$V_{TH}$/high-$V_{TH}$ states, respectively, the output signal will therefore switch between 0.7 V (i.e., when branch 1 is active) and 0 V (i.e., when branch 2 is active). The experimental results therefore confirm successful operations. Figures S6 to S8 show experimental results of the other three configuration combinations of two branches, which further verify the successful runtime reconfiguration operation. Similar to the LUT cell case, SPICE simulations are conducted to predict the speed of a fully integrated CB, where the simulated transient waveform of the proposed multi-configuration CB is shown in Fig. S9.

**Evaluation and application case study**

To evaluate the feasibility and performance of the proposed FeFET-based context-switching FPGA architecture, simulations are performed and a comprehensive comparison with other relevant works based on different memory technologies is shown in terms of area, delay, and power consumption. Moreover, at the system level, the capability of the proposed architecture to successfully achieve dynamic reconfiguration is demonstrated and the evaluation results show that the design presents a significant power reduction and area efficiency improvement with slightly increased critical path delay as the trade-off. To estimate the area of FeFET-based CB and LUT cells and compare them with other works, the layouts are drawn and the area is calculated using the design rules of the GPDK 45-nm library in fig. S10. All relevant area numbers are shown in Fig. 5A. Our layout analysis shows that the proposed CB and LUT cells are more compact compared to SRAM CBs and LUT cells. For example, the proposed FeFET-based single-configuration CB and LUT cells occupy an area that is only 12.6 and 18.5% of their respective SRAM-based counterparts, while the prior FeFET-based CB and LUT cells (42) require 77.0 and 97.0% of that area, respectively. Even the proposed multi-configuration FeFET CB and LUT cell area is only 25.3 and 37.0% of that of the SRAM-based single-configuration design. Therefore, the proposed design shows a significant area reduction compared to the SRAM-based design and previous FeFET-based design (42).

Figure 5B summarizes the basic structures of six-input LUT/ CB/SB based on existing memory technologies (SRAM, STT-MRAM, RRAM, and FeFET) and compares their corresponding read delay and read power consumption. All circuits are simulated with HSPICE. The 45-nm PTM (46) is adopted for all MOSFETs in this work and a calibrated FeFET model (45) is used for the proposed design. For resistive memories, the corresponding low-resistance and high-resistance levels are used for simulation (48, 49). According to the simulation results (Fig. 5B), we observe that for a six-input LUT, the proposed single-configuration LUT shows the smallest read power consumption, which is 13.1 $\mu$W, and for multiple configurations, this number increases slightly but still less than the power consumed by magnetic tunnel junction (MTJ)-based single-configuration LUT. This is due to the large ON/OFF ratio of FeFET obviating the need for a high read current to differentiate its two states, unlike MTJ designs. As for the read delay, RRAM-based single-configuration LUT has the longest latency. The proposed FeFET-based single-configuration LUT shows the second-best latency in all considered nonvolatile LUTs. Besides, the delay of the proposed FeFET-based multi-configuration LUT is less than that of RRAM-based single-configuration LUT even though considers one extra multiplexer for selecting configurations. The switching current through the sense amplifier for FeFET is larger than RRAM due to its higher on/off ratio (lower $R_{on}$), resulting in less LUT delay than RRAM. For CBs, our 1FeFET single-configuration CB and 2T-2FeFET multi-configuration CB show much less power...
consumption during operation, which consume \( \sim 95\% / \sim 85\% \) less power than the SRAM-based CB. For SBs, both FeFET-based single-configuration SB and multi-configuration SB show much less power consumption than others since our circuit contains fewer transistors. However, the delay of 1FeFET CB is around two times that of an SRAM-based CB. The delay of FeFET-based SB is the worst among different memory technology–based designs. That is because FeFET’s transmission speed is not as high as a conventional MOSFET, resulting in poorer performance than CB. In conclusion, the proposed FeFET-based designs (CB/SB) show significant advantages in power consumption over SRAM/STT-MRAM/RRAM–based designs but with a slight penalty in delay. Note that the penalty in
the routing elements’ (CB/SB) delay does not necessarily mean that the overall system will be affected as the routing delay may be a small portion of the overall system delay, which is investigated below (Fig. 5C).

To investigate the impact of the primitive (i.e., LUT/SB/CB) delay on the latency of the whole FPGA, the critical path delay is studied with the Verilog-to-Routing (VTR) tool (50). The VTR tool is a popular open-source CAD tool for FPGA architecture development and evaluation. For fair comparison, all the SRAM-/RRAM-/STT-MRAM-/FeFET–based FPGAs use a well-optimized and commercial FPGA architecture using 45-nm technology in VTR. To get the critical path delay of different memory technology–based

![Fig. 5. Area comparison and simulation results. (A) Area impact of FeFET LUT cells (storage) and CBs over SRAM-based structures. (B) Delay and power comparison of main components of FPGA based on different memory technologies. (C) Critical path delay of different memory technology–based FPGA designs.](image-url)
FPGAs, seven circuitry benchmarks (stereovision0, blob_merger, sha, spree, boundtop, diffeq2, and or1200) included in VTR are conducted (50, 51). These represent popular applications in diverse domains, such as image processing, math, cryptography, and computer vision. Figure 5C compares the critical path delay measured from SRAM-/RRAM-/STT-MRAM-/FeFET–based FPGAs. Compared with SRAM-based FPGA, the FeFET-based single-configuration FPGA presents an 8.6% reduction in the critical path delay on average, and it is also better than RRAM-based architecture. However, the proposed FeFET-based multi-configuration FPGA shows a 9.6% increment in the critical path delay compared to SRAM-based FPGA. The simulation confirms that the delay of LUTs is dominant in the overall delay of the entire FPGA, therefore explaining the aforementioned performance of these FPGAs. More details on the simulation are explained in fig. S11.

In addition, to show the feasibility of implementing the whole design in deep learning applications, three case studies under different scenarios are investigated. The first case is presented to show the benefit provided by dynamic reconfiguration in image classification. In the evaluation, two approaches of inference are considered—static inference and dynamic inference. For static inference, the input image is classified by the generalist classifier. However, for dynamic inference, the input image is first classified by the superclass classifier to identify the superclass. If the superclass is supported by the specialist subclass classifier network, then the configuration of the subclass classifier would be switched and executed for enhanced accuracy. Otherwise, a generalist classifier is invoked to complete the subclass identification. The whole workflow is shown in Fig. 6A. Figure 6B shows that dynamic inference for superclass classification improves the accuracy by up to 3.0% over static inference. Only context-switching FPGA can efficiently realize dynamic inference. In the last two cases, the feasibility and advantages of the proposed design over the conventional FPGA design are evaluated in terms of timing when considering various application scenarios. Basically, three neural networks (ResNet50, CNV, and MobileNetv1) are deployed into FPGA through the Xilinx Vitis AI platform (52). In the second case study, a case scenario that needs to switch between two neural networks frequently (Fig. 6C) is considered. In conventional FPGA, it is necessary to load new configurations before switching contexts, which is time-consuming. However, for this context-switching design, our approach can pre-load two configurations, and then freely switch between them without the reconfiguration latency. The switch time of the proposed design is less than 1 ns which is much smaller than the reconfiguration time and the proposed design shows significant speed up (from 39.0 to 97.5%; Fig. 6D). The last case study is related to dynamic reconfiguration. It is assumed that there are three different neural networks to implement and switch between. Thus, in this case, there would be six situations corresponding to six combinations of these three networks (ResNet50→CNV→MobileNetv1, ResNet50→MobileNetv1→CNV, CNV→ResNet50→MobileNetv1, CNV→MobileNetv1→ResNet50, MobileNetv1→ResNet50→CNV, and MobileNetv1→CNV→ResNet50).

DISCUSSION
In summary, we propose a FeFET-based context-switching FPGA architecture with the capability of dynamic reconfiguration, which can mitigate the trade-off in conventional FPGA between the chip area cost and reconfiguration latency. In addition, we experimentally verify the functionality of the primitive blocks of the proposed FPGA. The simulation results reveal that by leveraging FeFETs, the proposed primitives of the FPGA show huge area and power reduction compared to conventional SRAM-based design. Moreover, three representative application scenarios are investigated and studied. The evaluation results show the proposed context-switching FPGA supporting dynamic reconfiguration offers significant time-saving in these application scenarios. Our design provides an efficient solution to bridge the gap and makes FPGA more competitive in accelerating complex deep learning applications.

MATERIALS AND METHODS
Device fabrication
Here, the fabricated FeFET features a polycrystalline Si/TiN/doped HfO2/SiO2/p-Si gate stack. The devices were fabricated using a 28-nm node gate-first HKMG CMOS process on 300-mm silicon wafers. Detailed information can be found in (45, 53). The ferroelectric gate stack process module starts with removing the native oxide through wet etch, then the growth of a thin SiO2-based interfacial layer through wet chemical oxidation, followed by the deposition of the doped HfO2 film through atomic layer deposition. A TiN metal gate electrode was deposited using physical vapor deposition, on top of which the poly-Si gate electrode was deposited. The source and drain n+ regions were activated by rapid thermal annealing (RTA) at approximately 1000°C. The reason that 1000°C is used is because the source/drain dopant activation and the ferroelectric phase stabilization are performed at the same step. This is the gate-first process. Of course, lower temperature can be used if the gate last process is adopted. With Hf1-xZrO2, annealing at the back-end-of-line compatible temperature is even possible (≤450°C). This step also results in the formation of the ferroelectric orthorhombic phase within the doped HfO2. After RTA, the HfO2 becomes polycrystalline, where multiple crystalline phases can coexist, including the monoclinic dielectric phase, orthorhombic ferroelectric phase, and tetragonal
antiferroelectric phase. For future suppression of device variation, further optimization for phase-pure orthorhombic HfO₂ is necessary. For all the devices electrically characterized, they all have the same gate length and width dimensions of 0.5 μm by 0.5 μm, respectively.

**Electrical characterization**

The experimental verification was performed with a Keithley 4200-SCS Semiconductor Characterization System (Keithley system), a Tektronix TDS 2012B Two Channel Digital Storage Oscilloscope (oscilloscope), and a Keysight 81150A Pulse Function Arbitrary Generator (waveform generator). Two 4225-PMUs (pulse measurement units) were used to generate proper waveforms. The FeFETs used in experimental verification were connected with devices (inverters, p-type MOSFET, and/or n-type MOSFET) externally on a breadboard. In the experimental verification of the LUT cell operation, V_DD was given by the waveform generator. Output pulses

![Diagram](image-url)
were captured by the oscilloscope. Write and read operations were provided by the Keithley system. In the experimental verification of the multi-configuration CB operation, input voltage was given by the waveform generator. Output pulses were captured by the oscilloscope. WL and Enable (EN) signals were generated by the Keithley system. Three repeated cycles were performed for each configuration. State initialization (±4 V or −4 V to both WL1 and WL2 with a pulse width of 1 μs) was added at the beginning of the waveforms to generate the desired output in the first cycle.

Supplementary Materials
This PDF file includes:
− Figs. S1 to S12
+ configuration. State initialization (4 V to both WL1 and WL2 with a pulse width of 1 μs) was added at the beginning of the waveforms to generate the desired output in the first cycle.

REFERENCES AND NOTES
1. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778.
2. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21 July 2017, pp. 4700–4708.
3. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149 (2015).
4. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp. 779–788.
5. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805 [cs.CL] (2018).
6. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 [cs.CL] (2019).
7. A. Mehonian, A. J. Kenyon, Brain-inspired computing needs a master plan. Nature 604, 255–260 (2022).
8. J.-W. Chang, K.-W. Kang, S.-J. Kang, An energy-efficient fpga-based deconvolutional neural network accelerator for single image super-resolution. IEEE Trans. Circuits Syst. Video Technol. 30, 281–295 (2020).
9. J. Li, K.-F. Un, W.-H. Yu, P.-J. Mak, R. P. Martins, An fpga-based energy-efficient reconfigurable convolutional neural network accelerator for object recognition applications. IEEE Trans. Circuits Syst. II: Express Briefs 68, 3134–3147 (2021).
10. K. Guo, S. Zeng, J. Yu, Y. Wang, H. Yang, A survey of fpga-based neural network inference accelerators. ACM Trans. Reconfig. Technol. Syst. 12, 1–26 (2019).
11. S. W. Kim, S. Kang, H. Cho, D. Shin, S. Kang, A low active leakage and high reliability phase change memory (pcm) based non-volatile fpga storage element. IEEE Trans. Circuits Syst. II: Express Briefs 61, 2605–2613 (2014).
12. W. Zhang, G. Shi, J. Zhang, J. Yuan, H. Mulaosmanovic, A. Mallick, L. Sun, R. Joshi, X. Li, N. Shukla, V. Pavlovic, Proceedings of the 19th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, FPGA'04, Monterey, CA, February 22–24, 2004, pp. 23–30.
13. J. Shang, A. S. Kaviani, K. Bathala, Proceedings of the 2002 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays, FPGA'02, Monterey, CA, February 24–26, 2002, pp. 157–164.
14. F. Li, D. Chen, L. He, J. Cong, Proceedings of the 2003 ACM/SIGDA Eighth International Symposium on Field-Programmable Gate Arrays, FPGA'03, Monterey, CA, February 23–25, 2003, pp. 175–184.
15. P. F. O’Boyle, Proceedings of the Reconfigurable Computing Is Going Mainstream, 12th International Conference on Field-Programmable Logic and Applications (Springer-Verlag, 2002).
16. J. Greene, S. Kaptanoglu, W. Feng, V. Hecht, J. Landry, F. Li, A. Kruglyanskii, M. Morosan, P. Vezzner, Proceedings of the 19th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, Feb 27–March 1, 2011, pp. 87–96.
17. S. Tanachutiwat, M. Liu, W. Wang, Fpga based on integration of cmos and ram. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19, 2023–2032 (2011).
18. K. Huang, Y. Ha, R. Zhao, A. Kumar, Y. Lian, A low active leakage and high reliability phase change memory (pcm) based non-volatile fpga storage element. IEEE Trans. Circuits Syst. I: Regul. Pap. 61, 2605–2613 (2014).
19. W. Zhao, E. Belhaire, C. Chappert, P. Mazoyer, Spin transfer torque (stt)-mram-based runtime configuration fpga circuit. ACM Trans. Embed. Comput. Syst. 9, 1–16 (2009).
20. H. Mulasmanovic, E. T. Breyer, S. Dünkel, T. Meyer, Ferroelectric field-effect transistors based on In0.2: A review. Nanotechnology 32, 101810/136-6528/ ac189f (2021).
21. U. Schroeder, M. H. Park, T. Mikolajek, C. S. Huang, The fundamentals and applications of ferroelectric HfO2. Nat. Rev. Mater. 7, 653–669 (2022).
22. A. I. Khan, A. Keshavari, S. Datta, The future of ferroelectric field-effect transistor technology. Nat. Electron. 3, 588–597 (2020).
23. Y. Du, Y. Xu, S. Deng, Z. Zhao, N. Jiao, Y. S. Kim, S. Duensek, S. Beyer, K. Ni, S. George, V. Narayanan, Hardware functional obfuscation with ferroelectric active interconnects. Nat. Commun. 13, 2235 (2022).
24. X. Chen, K. Ni, M. T. Nienier, H. Yan, S. Datta, X. S. Hu, Power and area efficient fpga building blocks based on ferroelectric fets. IEEE Trans. Circuits Syst. I: Regul. Pap. 66, 1780–1793 (2019).
25. Z. Jiang, Z. Zhao, S. Deng, Y. Xiao, Y. Xu, H. Mulasmanovic, S. Duensek, S. Beyer, S. Meninger, M. Mahamed, R. Joshi, G. Song, K. Nienier, V. Narayanan, K. Ni, On the feasibility of 11 ferroelectric FET memory array. IEEE Trans. Electron Devices 69, 6722–6730 (2020).
26. H. Mulasmanovic, J. Ocker, S. Müller, U. Schroeder, J. Müller, P. Polakowski, S. Flachowsky, R. van Bentum, T. Mikolajek, S. Slesazeck, Switching kinetics in nanoscale hafnium oxide based ferroelectric field-effect transistors. ACS Appl. Mater. Interfaces 9, 3792–3798 (2017).
27. S. Deng, G. Yin, W. Chakraborty, S. Datta, X. S. Hu, 2020 IEEE Symposium on VLSI Technology. June 14–19, 2020, pp. 1–2.
28. Y. K. Cas, What is predictive technology model (ptm)? SIGDA Newslett. 39, 1 (2009).
29. Y. Xiao, Y. Xu, Z. Jiang, S. Deng, Z. Zhao, A. Mallick, L. Sun, R. Joshi, X. Li, N. Shukla, V. Narayanan, K. Ni, IEEE International Electron Devices Meeting, San Francisco, CA, USA, Dec. 3–7, 2022.

Xu et al., Sci. Adv. 10, eadd1525 (2024) 17 January 2024

12 of 13
48. J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, A. Raychowdhury, A 40-nm 118.44-tops/w voltage-sensing compute-in-memory ram macro with write verification and multi-bit encoding. *IEEE J. Solid-State Circuits* **57**, 845–857 (2022).

49. C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu, W. Chen, Y. Lin, M. Nowak, N. Yu, L. Tran, 2009 IEEE International Electron Devices Meeting (IEDM), Baltimore, MD, USA, December 7–9, 2009, pp. 1–4.

50. K. E. Murray, O. Petelin, S. Zhong, J. M. Wang, M. ElDafrawy, J.-P. Legault, E. Sha, A. G. Graham, J. Wu, M. J. P. Walker, H. Zeng, P. Patros, J. Luu, K. B. Kent, V. Betz, Vtr 8: High performance cad and customizable fpga architecture modelling. *ACM Trans. Reconfigurable Technol. Syst.* **13**, 1–55 (2020).

51. J. Rose, J. Luu, C. W. Yu, O. Densmore, J. Goeders, A. Somerville, K. B. Kent, P. Jamieson, J. Anderson, Proceedings of the ACM/SGDA International Symposium on Field Programmable Gate Arrays, FPGA’12, Monterey, CA, USA, February 22–24, 2012, p. 77–86.

52. Vivado design suite user guide: Partial reconfiguration. https://docs.xilinx.com/v/vivado/qpanel/filebrowser/doc/ug909-2018.1-english/ug909-vivado-partial-configuration.pdf (2018).

Acknowledgments

Funding: This work is mainly supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences Energy Frontier Research Centers program under Award Number DESC0021118. The experimental characterization is partially supported by the Army Research Office under grant number W911NF-21-1-0341 (to K.N.). The architecture benchmarking is partially supported by ND EPSCoR (to S.G.) and partially by NSF 2132918 (to V.N.) and 2008365 (to V.N.). The device fabrication is funded by the German Bundesministerium für Wirtschaft (BMWi) and by the State of Saxony in the frame of the “Important Project of Common European Interest (IPCEI).”

Author contributions: V.N., S.G., and K.N. proposed and supervised the project. Y. Xu, Y.X., T.Y., and E.H. conducted circuit and architectural simulations. Z.Z. performed experimental verification. H.M., D.K., and S.B. fabricated the FeFet devices. X.G. helped with FPGA array programming. R.I. and X.H. helped with FPGA benchmarking. S.W., A.S.R., K.L., and L.I. conducted the benchmarking of FPGA implementation of super-sub networks. All authors contributed to the write-up of the manuscript. Competing interests: A patent application was submitted for this work on 9 November 2023 with the names of Y. Xu, Z.Z., Y.X., V.N., K.N., and S.G. on it. It has been issued by the Office of Technology Management of Pennsylvania State University. A cover page patent was applied on 30 November 2023 for Penn State invention disclosures (2023-5715) to V.N. The other authors declare that they have no competing interests.

Data and materials availability: All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials.

Submitted 4 August 2023
Accepted 18 December 2023
Published 17 January 2024
10.1126/sciadv.adk1525