CIFER: A 12nm, 16mm², 22-Core SoC with a 1541 LUT6/mm², 1.92 MOPS/LUT, Fully Synthesizable, Cache-Coherent, Embedded FPGA

Ting-Jung Chang*[1], Ang Li*[1], Fei Gao[1], Tuan Ta[2], Georgios Tziantzioulis[1], Yanghui Ou[2], Moyang Wang[2], Jinzheng Tu[1], Kaifeng Xu[1], Paul J. Jackson[1], August Ning[1], Grigory Chirkov[1], Marcelo Orenes-Vera[1], Shady Agwa[2], Xiaoyu Yan[2], Eric Tang[2], Jonathan Balkind[3], Christopher Batten[2], David Wentzlaff[1]

*Equal Contribution, [1]Princeton University, [2]Cornell University, [3]University of California, Santa Barbara

Embedded FPGAs (eFPGA) are increasingly being used in SoCs, enabling post-silicon hardware specialization. Existing CPU-eFPGA SoCs have three deficiencies. First, their low core count hinders efficient execution of thread-level-parallel workloads. Second, non-coherent or partially coherent CPU-eFPGA integration inhibits dynamic, random memory sharing. Third, the use of full-custom circuits makes proprietary eFPGAs technology-dependent, inflexible in physical layout, and lacking architectural customizability.

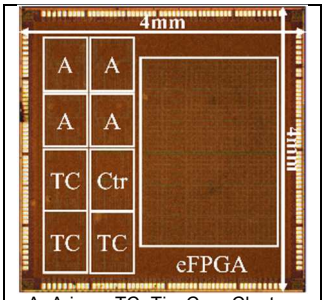
CIFER is the world's first open-source, many-core, synthesizable, cache-coherent CPU-FPGA SoC. CIFER was designed in seven months during the pandemic by a team of graduate students and postdocs collaborating across two institutions, due in part to the use of many open-source projects, including OpenPiton, BYOC, PyMTL, PyOCN, Ariane, and PRGA. The 4×4mm² chip is fabricated in 12nm FinFET and targets intelligent edge devices such as robots and edge servers. CIFER addresses the aforementioned deficiencies with the following novelties: First, CIFER integrates parallel tiny-core clusters, OS-capable processors, and an eFPGA, enabling efficient execution of various workloads across the parallelism-specialization spectrum. Second, CIFER implements a heterogeneous, bi-directional cache coherence scheme, enabling low-latency, byte-granular data sharing between all the processors and the eFPGA. Third, the eFPGA is fully synthesizable with standard cells and off-the-shelf EDA tools.

Architecture: The CIFER architecture (Fig. 1) integrates a 2×4 mesh of tiles and an eFPGA into the distributed, coherent, OpenPiton P-Mesh cache system over three packet-switched, on-chip networks (OCN) designed with PyOCN. Each tile consists of a shard of the coherence system and one of the following: an Ariane core, a TinyCore cluster, or an eFPGA controller. Each coherence shard contains a private, 8KB, L2 cache and a 64KB slice of the shared, 512KB, last-level cache (LLC). Coherence between the L2s and the LLC is maintained in hardware with a directory-based MESI protocol.

Ariane is a Linux-capable, 64-bit, RISC-V processor with a 16KB L1 instruction cache (L1I), an 8KB L1 data cache (L1D), and a double-precision floating-point unit (FPU). Coherence between the L2 and Ariane's L1I/L1D is maintained in hardware. Each TinyCore cluster contains six 32-bit, RISC-V cores organized into three pairs. Each core has a private, 4KB L1D, while each pair of cores share a 4KB L1I, an integer multiply-divide unit (MDU), and a single-precision FPU. TinyCore clusters use a MIMD execution model and a software coherence scheme, where cache flush and invalidation are managed in software. Sharing long-latency arithmetic units and reducing coherence hardware maximize computation density in each cluster.

eFPGA: The eFPGA (Fig. 2) has 6720 multi-mode, 6-input LUTs and 18 24Kbit, dual-port, block RAMs. Emulated accelerators can be built with an open-source, RTL-to-bitstream toolchain consisting of Yosys, VPR, and PRGA's bitstream assembler. The eFPGA is integrated with the system through two interfaces in the eFPGA controller: the control register interface allows the CPUs to access the eFPGA via memory-mapped I/O; the coherent memory interface is configurable at runtime to enable non-coherent, IO-coherent, or bidirectionally coherent memory accesses of the eFPGA. Atomic requests from the eFPGA are also supported, enabling low-overhead synchronization in user mode. The flexibly cache-coherent, fault-tolerant integration maximizes the programmability of the SoC.

The eFPGA contains two key novelties: First, the switch blocks implement a cycle-free connection pattern [1], facilitating automated, constraint-driven, area/timing optimization at the array level using off-the-shelf EDA tools. Compared to previous synthesizable FPGAs in which locally optimized blocks are tessellated in a predefined grid, our approach narrows the LUT density, performance, and energy efficiency gaps between full-custom and synthesizable FPGAs down to 1.3×, 3.4×, and 2.1×, respectively (Fig. 5). Second, the configuration memory is organized as multiple single-bit scanchains interconnected via an 8-bit, packet-switched, 2D-mesh network and uses an analog, multi-source clock mesh running in the same clock domain as the CPUs. This enables fast and partial reconfiguration of the eFPGA at GHz clock frequency.



A=Ariane; TC=TinyCore Cluster; Ctr=eFPGA Controller

Die micrograph

Evaluation: Fig. 3 shows our chip testing setup. Fig. 4 shows the maximum operating frequency (F_{max}) of each component across the range of functional supply voltages. Note that the eFPGA's F_{max} depends on the emulated design, and Fig. 4 shows the F_{max} of a 64-bit LFSR.

Fig. 5 compares CIFER with other state-of-the-art CPU-FPGA SoCs targeting the edge/IoT domain. The SoC runs up to 1195MHz at 1.1V. The CPUs' aggregate peak performance and energy efficiency are 15.54 GFLOPS at 1.1V and 53.18 GFLOPS/W at 0.7V (estimated power dissipation, excluding the eFPGA's configuration clock power based on post-layout power analysis), outperforming the next best SoC by 6.5× and 1.4×. The eFPGA achieves an area efficiency of 1541 LUT6/mm², outperforming the other synthesizable eFPGAs by 11.2×, and is only 1.3× worse than the best full-custom eFPGA. The eFPGA's peak performance (1.92 MOPS/LUT, 126MHz at 1.1V) and energy efficiency (148.1 GOPS/W at 0.7V) are measured with a 64-point FFT that reaches 97% utilization of the eFPGA. The 3.4× performance gap and the 2.1× energy efficiency gap between the best full-custom eFPGA and this work can be attributed to three factors: (1) CIFER is synthesized with standard cells; (2) our eFPGA has no hardware multiply-accumulate units; and (3) this work uses an open-source FPGA CAD toolchain. The last two rows show the peak memory bandwidth when the CPUs and the eFPGA (running at 10% of the CPU clock frequency) access shared memory in a random pattern. C→F shows the bandwidth when an Ariane core accesses data owned by the eFPGA's private cache, and F→C shows the opposite. Note that SMIV [5] implements the AXI4 ACP protocol that only supports I/O-coherence in which CPU accesses do not trigger cache invalidation on the eFPGA side.

Fig. 6 shows the throughput and energy efficiency gains by offloading four representative edge applications to their preferred compute unit. SORT and SHA-256 use eFPGA-emulated accelerators, while GEMM and JACOBI2D use the TinyCore clusters. The measured runtime includes all the control overhead, while the data transfer overhead is mitigated by overlapping compute with ad hoc, coherent memory accesses. To fairly compare the energy efficiency of individual components, full-chip idle power (static power and clock power) is excluded. At nominal voltage, the eFPGA outperforms the Ariane-only baseline by up to 9.29x in throughput and 10.62x in energy efficiency; the TinyCore clusters improve the performance and energy efficiency by up to 7.95x and 7.75x, respectively.

Acknowledgements: This material is based on research sponsored by the Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement No. FA8650-18-2-7852. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.

References:

[1] A. Li *et al*. FPL, 2020, pp. 208-213
[2] F. Renzini *et al*. TCAS-I, vol. 67, no. 2, pp. 489-501, 2020
[3] M. Natsui *et al*. ISSCC, 2019, pp. 202-204
[4] P. D. Schiavone *et al*. TVLSI, vol. 29, no. 4, pp. 677-690, 2021
[5] S. K. Lee *et al*. JSSC, vol. 57, no. 2, pp. 639-650, 2022
[6] MicroChip PolarFire®, Embedded World Exhibition & Conference, 2019

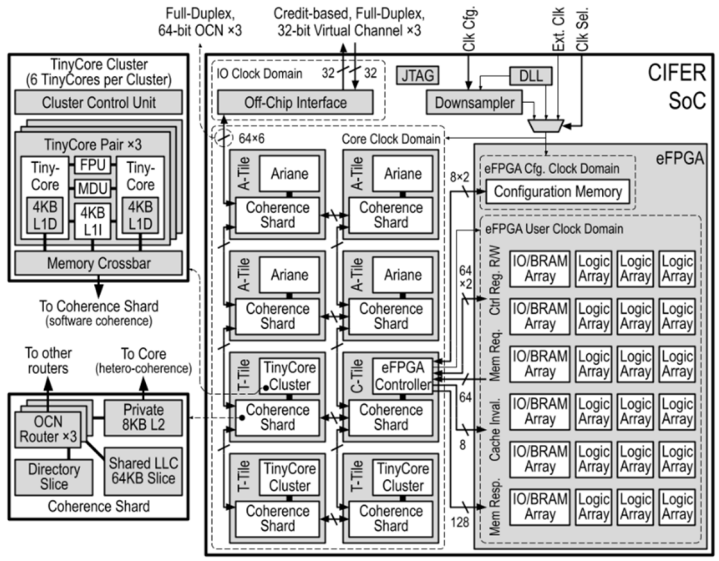


Fig. 1. SoC Architecture

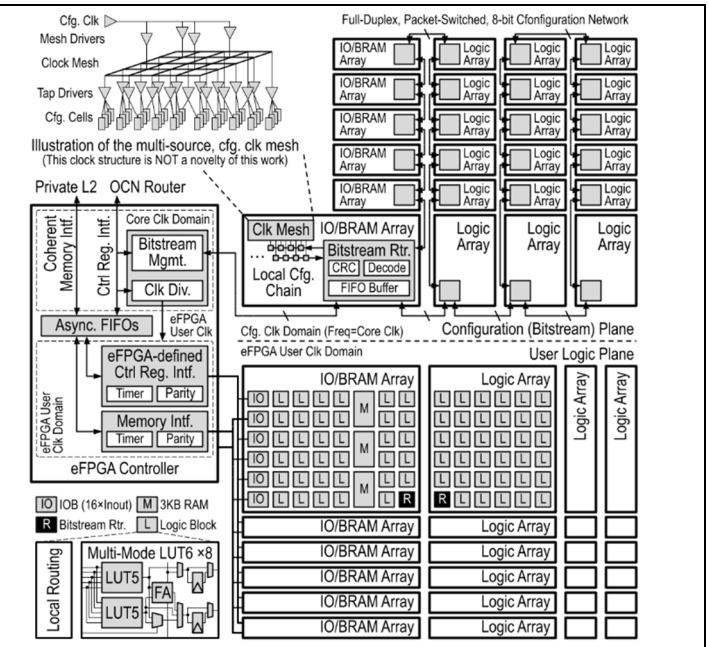


Fig. 2. eFPGA Architecture

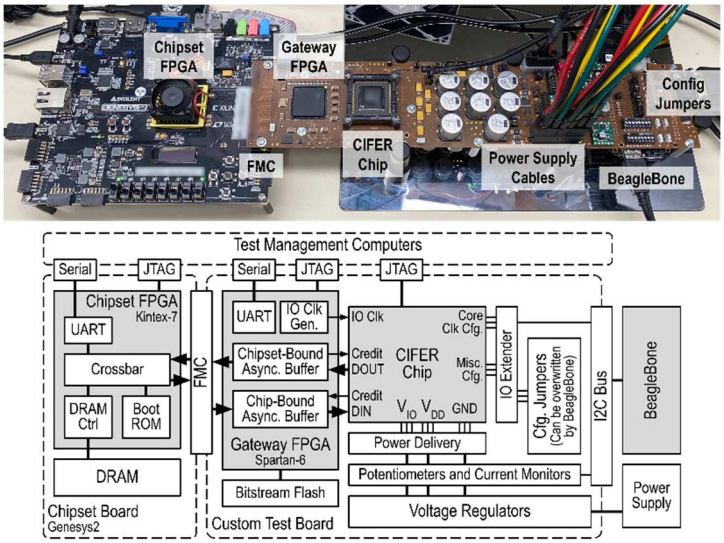


Fig. 3. Lab Evaluation Setup

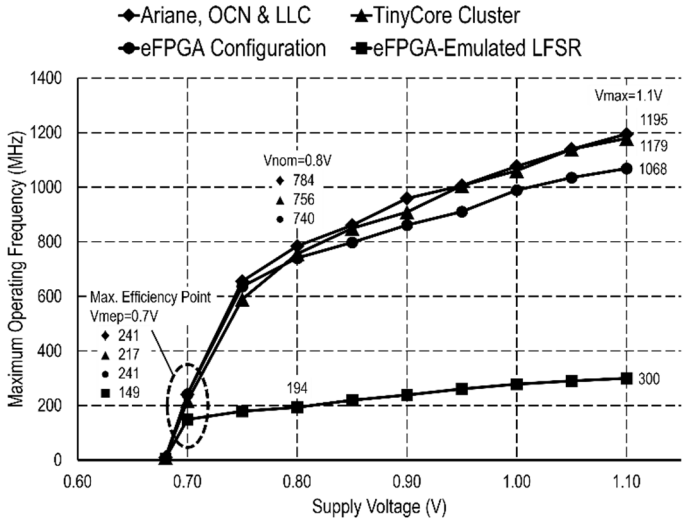
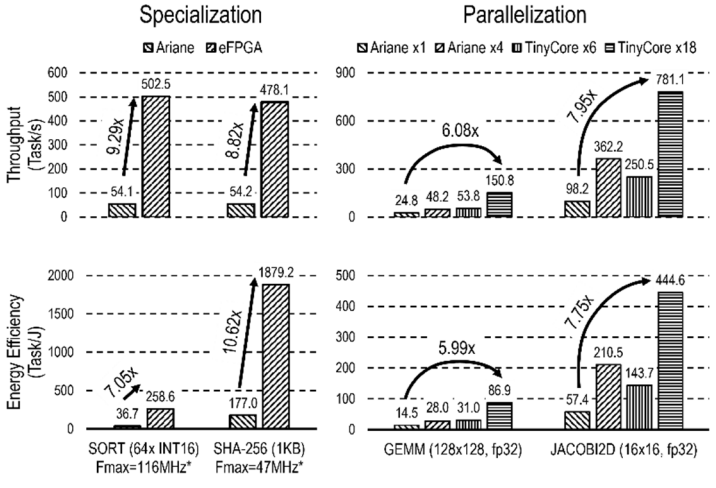


Fig. 4. Maximum Operating Frequency vs. Supply Voltage

			This Work	TCAS'20 [2]	ISSCC'19 [3]	TVLSI'21 [4]	JSSC'22 [5]	PolarFire [6]
Chip	Technology		12nm FinFET	90nm BCD	40nm CMOS + 39nm MRAM	22nm FD-SOI	16nm FinFET	28nm CMOS + 28nm SONOS
	Die Area (mm²)		16	1.78	22.09	9	25	-
	Vnom (Vmin - Vmax)		0.8 (0.68-1.1)	1.2 (-)	- (1.1-1.3)	0.8 (0.5-0.8)	0.8 (0.5-1.05)	1.0 (1.0-1.05)
	Active Pwr (mW)	Vnom	1792	1.2	5.34	24.95	918	2980
	Fmax (MHz)	Vmax	**1195**	10	200	600	972	600
CPU Host	Core Type		4x Ariane	RI5CY	Cortex-M0	RI5CY	2x Cortex-A53	4x SiFive U54
	ISA		RV64GC	RV32I	ARMv6-M	RV32IMFC	ARMv8-A	RV64GC
	CoreMark Score	Vmax	**7918**	31.9	466	1914	6376	7584
CPU Other	Core Type		18x TinyCore	N/A	N/A	N/A	Cortex-M0	SiFive E51
	ISA		RV32IMAF				ARMv6-M	RV64IMAC
	Function		Parallel Compute				Monitor	Monitor
	CoreMark Score	Vmax	**19198**				2265	1896
Total	Peak GFLOPS	Vmax	**15.54**	No HW FPU	No HW FPU	No HW FPU	1.94	2.4
	Peak GFLOPS/W	Vmep	6.63 **[53.18†]**				38.03	-
eFPGA	IP		**Synthesizable w/ Std. Cells**	Synthesizable w/ Std. Cells	Unknown	Fully-Custom Hard Macro	Fully-Custom Hard Macro	Fully-Custom Hard Macro
	Min. Prog. Time (us)		**239.4 - 1274.8**	-	-	-	450	-
	LUT Type & Count		6720 LUT6	48 LUT6	1176 LUT6	6000 LUT4	8760 LUT6	150K LUT4
	Density (LUT/mm²)		1541	137	36	1505	1991	-
	Fmax (MHz)	Vmax	300**	1.25	200	193	747	400
	MOPS/LUT	Vmax	1.92‡ (INT8)	-	-	0.02 (INT32)	6.45 (INT8)	2.03 (INT8)
	GOPS/W	Vmep	148.1‡* (INT8)	-	-	29.1 (INT32)	312.4 (INT8)	102.0 (INT8)
Shared-Memory BW (MB/s) [Vmax]		C→F	**201**	Non-Coherent	Non-Coherent	Non-Coherent	-	Non-Coherent
		F→C	**558**				486	

† Estimated power dissipation, excluding the eFPGA's configuration clock power based on post-layout power analysis
** Measured when the eFPGA emulates a 64-bit LFSR
‡ Measured when the eFPGA emulates an INT8-precision, complex, 64-point FFT
* Measured power dissipation of the eFPGA's user clock domain

Fig. 5. Comparison to the State of the Art



* The CPUs, OCN, cache system, and the eFPGA controller run at full speed (740MHz at 0.8V). F_{max} indicates the maximum operating frequency of the eFPGA-emulated design

Fig. 6. Performance and Efficiency Gains of Offloaded Benchmarks