Exploring the Potential of OpenTitan as a Control-Flow Integrity Coprocessor

E. Parisi, A. Musa, S. Manoni, M. Ciani, D. Rossi, F. Barchi, A. Bartolini, A. Acquaviva

Department of Electrical, Electronic, and Information Engineering (DEI) – University of Bologna, Italy

emanuele.parisi@unibo.it

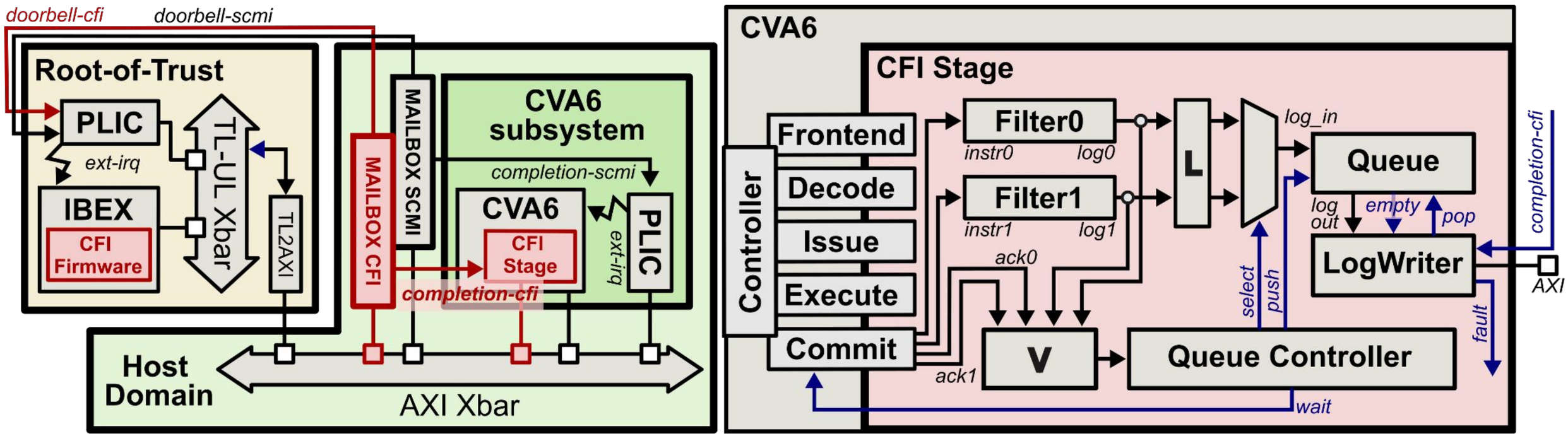

TABLE I
CYCLES REQUIRED TO IMPLEMENT THE RETURN ADDRESS PROTECTION POLICY IN OPENTITAN

		Op.	Instructions [#]			Cycles [#]			Cycles [%]		
			IRQ	CFI	TOT	IRQ	CFI	TOT	IRQ	CFI	TOT
IRQ	CALL	Logic	8	15	23	59	27	86	23	10	33
		Mem. RoT	14	5	19	74	28	102	29	9	38
		Mem. SoC	2	4	6	22	48	70	10	19	29
		TOT	24	24	48	155	103	258	62	38	100
	RET.	Logic	8	15	33	59	45	104	21	16	37
		Mem. RoT	14	5	19	74	28	102	27	10	37
		Mem. SoC	2	4	6	22	48	70	9	17	26
		TOT	24	34	58	155	121	276	57	43	100
Polling	CALL	Logic	—	15	15	—	27	27	—	26	26
		Mem. RoT	—	5	5	—	28	28	—	27	27
		Mem. SoC	—	4	4	—	48	48	—	47	47
		TOT	—	24	24	—	103	103	—	100	100
	RET.	Logic	—	25	25	—	45	45	—	37	37
		Mem. RoT	—	5	5	—	28	28	—	23	23
		Mem. SoC	—	4	4	—	48	48	—	40	40
		TOT	—	34	34	—	121	121	—	100	100
Optimized	CALL	Logic	—	15	15	—	27	27	—	42	42
		Mem. RoT	—	5	5	—	5	5	—	08	08
		Mem. SoC	—	4	4	—	32	32	—	50	50
		TOT	—	24	24	—	64	64	—	100	100
	RET.	Logic	—	25	25	—	45	45	—	55	55
		Mem. RoT	—	5	5	—	5	5	—	06	06
		Mem. SoC	—	4	4	—	32	32	—	39	39
		TOT	—	34	34	—	82	82	—	100	100

Motivation and Contribution

- **Open-hardware platforms** are becoming widespread in **safety and security-critical systems** [1]. Modern designs enhances platform security implementing **Control-Flow Integrity (CFI)** extensions to address threats posed by code reuse attacks.
- This paper introduces an innovative co-processor-based architecture where the **OpenTitan Root-of-Trust (RoT)** encodes custom CFI policies in its 32-bit Ibex core.
- This approach avoids the area overhead associated with the integration of a separate security monitor, and it **maximizes the utilization of the RoT**.

Security Implications & Assumptions

- We assume the **CFI Mailbox cannot be tampered** by other entities in the SoC, and only the CFI stage and OpenTitan can access it. Additionally, we assume the **RoT private memory is intrinsically secure** and inaccessible by any party other than the Ibex microcontroller.
- Our system uses the internal RoT scratchpad to store sensible information, such as the shadow stack and it exploits its **cryptographic accelerators to secure metadata that are spilled to insecure memory locations**.

Summary of Findings

- This work presents a study to investigate the feasibility of **using the RoT as a coprocessor to enforce CFI policies**, enabling the possibility of implementing any CFI scheme without designing and integrating custom hardware monitors.
- CFI coprocessors embedded in the RoT **take advantage of tamper-proof memory and cryptographic accelerators** to enhance the security guarantees provided by the CFI enforcement scheme. We prove our solution by extending the architecture of a SoA RISC-V SoC [1].

[1] Maicol Ciani et al. “Cyber Security aboard Micro Aerial Vehicles: An OpenTitan-based Visual Communication Use Case”. In: 2023
[2] Ryan Roemer et al. “Return-Oriented Programming: Systems, Languages, and Applications”. In: ACM Trans. Inf. Syst. Secur. (2012)
[3] F. Zaruba et al. “The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology”. In: IEEE VLSI (2019)
[4] Nathan Burow et al. “SoK: Shining Light on Shadow Stacks”. In: 2019 IEEE S&P

RISC-V SUMMIT EU
JUNE 24 – 28 | MUNICH 2024