A Solder-Defined Computer Architecture for Backdoor and Malware Resistance

Candidate: Marc W. Abel

Committee: Travis Doom, Ph.D. Dissertation Director Jack Jean, Ph.D. Michael Raymer, Ph.D. Krishnaprasad Thirunarayan, Ph.D. T.K. Prasad Vincent Schmidt, Ph.D. Air Force Research Laboratory

For slides: https://wakesecure.com

Department of Computer Science and Engineering 30 November 2022



Three walls to defend

- Software
- Personnel
- Hardware

Four kinds of hardware problems

- Outdated approaches ignore security
- Excessive complexity hides problems
- Manufacturer interests prevail
- Silicon chips can't be repaired later

Three freedoms sought

- Independence from vendors
- Full ownership rights
- Permanent security

Two enablers of success

- Surface-mount technology
- Firmware in RAM as logic





Seven Basic Logic Gates A D flip-flop only changes its output when:

- 1. told it's time to check, and
- 2. output doesn't already reflect the input.



A RAM can remember a lot of 18-bit words.

18-bit "address" where store or retrieve will occur

RAM Random-Access Memory

18-bit word to store to or retrieve from the given address



of the number 63







CPU Principal Data Paths

From page 17 in text.



© 2021 IEEE. Reprinted, with permission, from M. W. Abel, "Solder-Defined Architectures for Trusted Computing," NAECON 2021— IEEE National Aerospace and Electronics Conference, 2021, p. 251, doi: 10.1109/NAECON49338.2021.9696432.

Data Layers of ALU



ALU with carry propagation elements shown

Small digits that are not subscripts indicate number of wires. From page 88 in text.

Superposition of ALU Operations From page 76 in text.



Same board space

theta	alpha0	alphal		beta3	374.28	:: r10k	:: r10k	ii r10k		374.35		• or	and	and			
			-1-6-2		•	::	::	::	•								
			atphaz			:: r10k	:: r10k	:: r10k				• nand	 nor	and			
					374.29	• •	• •	• •		374.36							
• •	•	•	· · ·	•	••••••	:: r10k	:: r10k	:: r10k	•			• nand	and	and			
•		•															
beta0	gamma0	gammal		beta4	374.26		aod0			374.37		nor	and	nand			
			gamma2		•••••	••••			•••••								
			guinnuz									and	xor	or			
					374.27		aodl			374.38							
· ·	· ·		· · · ·	·	••••••	••••			•		•••••	nand	nand	and			
• · · · · · · · · · · · · · · · · · · ·	• • • • • • • • •	•	: :	•		•			:	1		•					
betal	alpha3	alpha4		beta5	r10k r10k r10	lk			374	. 40	flop				buf	or	nand
			alpha5		1. 1. 1.	: E							DØ				
					r10k r10k r10	lk	-		:	: flop					nand	nand	nor
					r10k r10k				or	xor	xor				inv	and	and
•	•	•	· ·						•			•					
beta2	gamma3	gamma4		giant6	giant4		giant3								nand	nand	nand
			gamma5		••••••	••••				s			D1				
			-		••••••												
				giant7	giant5		giant2										
				••••••													
MO	 r10k r10k r10k r10k r10k r10k	R	: :			••••		•••	:						t	1	
				374.31		gian						inv	xor	xor	374.	374.39 ^t	buf
			L		80mhz				с								
						••••		···•				xor	and	and			buf
				374.32			giant0					••••	••••				
	r10k r10k r10k											and	and	and			
274.22		274 21			274.11				274 17		•••••	••••	••••	••••	••••	••••	••••
3/4.22			conctru		3/4.11				•	3/4.1/		xor	xor	xor	flop	flop	and
	Ml			Р									••••	••••		••••	
274 41		274 20			274 12	••••	274 10			274 10		xor	and	and	nand	xor	
3/4.41		5/4.20	574.50		574.12	••••	574.10		•			••••	•	•			
												and	and	and			
374 13	i i 🛄 🛄	374 34	374 0	•	374 5	••••	···• ····	••••		374 15		••••	••••	•			
	flop flop flop	••••••	•		•	flop	flop	flop	•			xor	xor	xor			
	••••			zeta		••••		••••				••••	•	•			
374 14	flop flop	374 33	374 1		374.6	flop		flop		374 16		and	and	and			
	···· ··· ···		5/4.1 •	i	5/4.0		••••		•			••••	••••	••••			
	flop flop flop					flop	xor					and	and	and			
374 7	374.23	374 4	···· ···	374 2		•	•	•	•	•	•	•	•	•			
••••••	••••••	••••••	flop flop			flop	flop	flop	and	and	and	xor	xor	xor			
					374.25	•	•	•	••••	••••	••••	••••	••••	••••			
374 8	374 24	374 19		374 3		and	and	and	nor	or	or	and	and	and			
	••••••	•••••••••••••••••••••••••••••••••••••••		•		••••			•	••••	••••	••••	••••				
						and			or	xor	xor	and	and				

Circuit Board Floorplan

From page 180 in text.



Fast Enough For

- Hardened desktop apps
- Electronic mail
- Light- to moderate-use servers
- Controlling objects that move
- Process controls
- Peripheral & device controllers
- Telephony
- Modest Ethernet switches

Too Slow For

- Most Web surfing
- Machine learning
- Image and video processing
- Self-driving vehicles
- Fast raster or vector graphics
- Fast symmetric cryptography
- Fast asymmetric cryptography
- Bioinformatics

Security Improvements

- Sticky out-of-range flag for all arithmetic
- Mixed-sign variants for add, subtract, multiply, shift, abs. value
- Stack overflow unlikely, can't lead to privilege escalation
- No program access to stack except CALL and RETURN
- No branch to addresses not present in the instruction word
- No privilege escalation via the CPU
- No DRAM or DRAM-associated vulnerabilities
- No complex logic from IC manufacturers within CPU
- Every I/O device confined to its own bus and buffer
- No CPU persistent state except for one firmware IC
- No secret functionality
- No vendor lock-in
- No encrypted or closed-source firmware
- No license fees to build, use, or modify
- No purpose of use limitations
- No right to repair infringements



Demonstration

Before This Can Be Built

- Preemptive multitasking
- I/O subsystem to support SPI and I²C buses
- Firmware loader
- Resolution of clock skew concern

Ways to Get Involved

- Firmware upgrade for faster multiplication
- Support for integer division
- Floating point like IEEE 754-2019, but 36- and 72-bit formats
- Floating point for compatibility (32- and 64-bit formats)
- More assembler features
- Lightweight operating system
- Lightweight scripting language
- Lightweight programming language
- Minimalist toolchain that can be audited
- I/O device drivers
- TCP/IP stack
- TLS 1.3
- New block cipher to leverage architecture
- Formal verification (similar to seL4 or INTEGRITY-178B)

