## Open-Source Radio Microcontroller for Fabrication

ECPE Senior Design May 2025, Team 27

Presenters: Nolan Eastburn, Nathan Stark, Ibram Shenouda, Ethan Kono, Will Custis, Noah Thompson

Faculty Advisor and Client: Dr. Henry Duwe

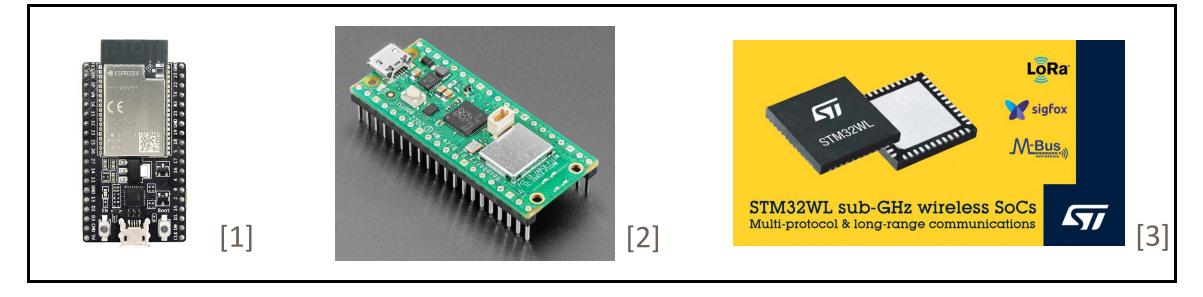
*Team Website: https://sdmay25-27.sd.ece.iastate.edu/* 

References to Design Document on slides as DD <page number>

### **Problem Statement**

The lack of open-source radio microcontroller designs makes learning about radio design for fabrication difficult for students and hobbyists.

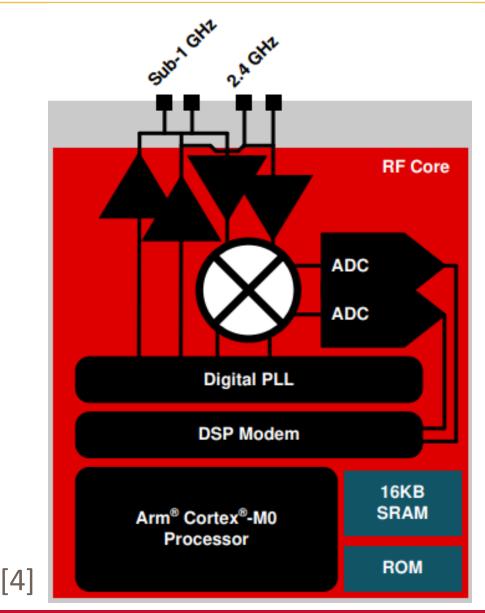
**Closed Source Radio Microcontrollers** 



DD 13

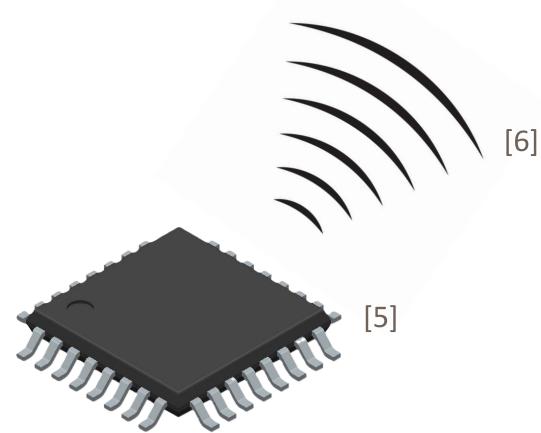
### **Market Research – Texas Instruments CC1352**

- Example of RF core architecture
- Provides a variety of wireless communication protocols
  - Thread
  - Zigbee
  - Bluetooth
- Multiple transmission frequencies
  - Ultra High (Sub GHz)
  - Super High (2.4 GHz)



## Requirements

- MCU shall contain a radio subsystem
- MCU shall implement an open standard wireless protocol stack
- MCU shall contain two independent RISC-V cores to execute user programs
- All the artifacts produced throughout the design of the MCU shall be open source.
- Thorough documentation usable by students with only basic knowledge of circuits and digital logic.
- DD 14



# **Technologies Used**

- Skywater 130 nm process
  - Provided by Efabless
  - Constraint from client
- Caravel Harness
  - Proven wrapper for project
- Openlane
  - Open-source tool for hardening digital designs

[7]

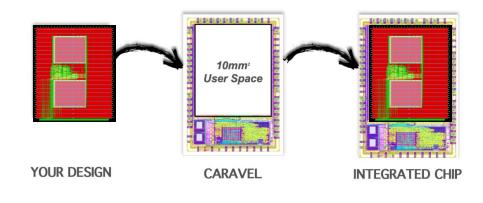
- GNU Compiler Collection (GCC)
  - Compile C programs for RISC-V processors
- Icarus Verilog
  - Simulate digital components to verify functionality
- NGSPICE
  - Simulate analog components to verify functionality



[10]

[9]





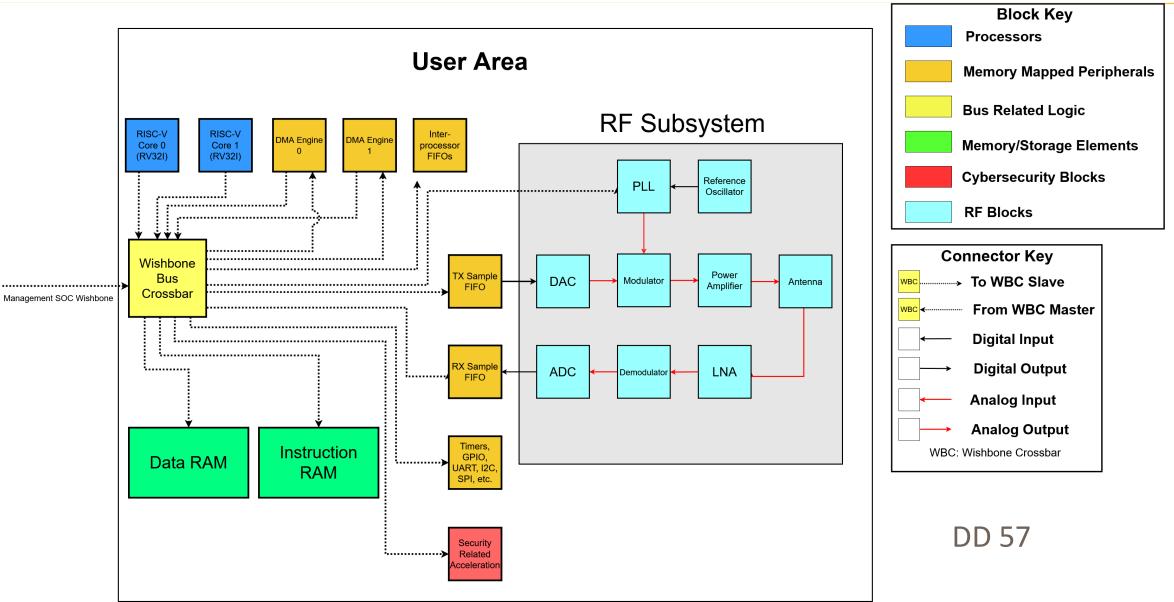
## **Choosing a Wireless Communication Standard**

- What wireless communication standard will we implement?
- Two major contenders:
  - Bluetooth and Zigbee
- Bluetooth operates at 2.4 GHz while ZigBee operates at ~915 MHz
- ZigBee is an open standard while Bluetooth is not.
- DD 40

**Zigbee** 



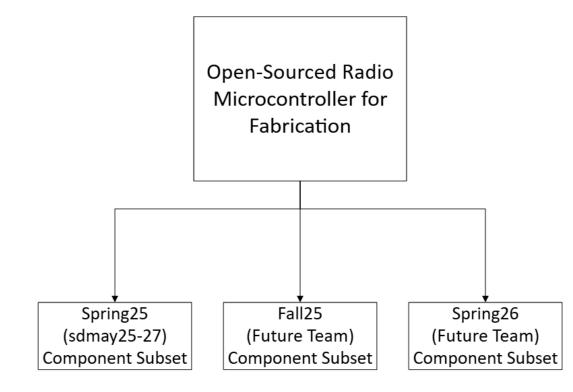
### **Detailed Design**



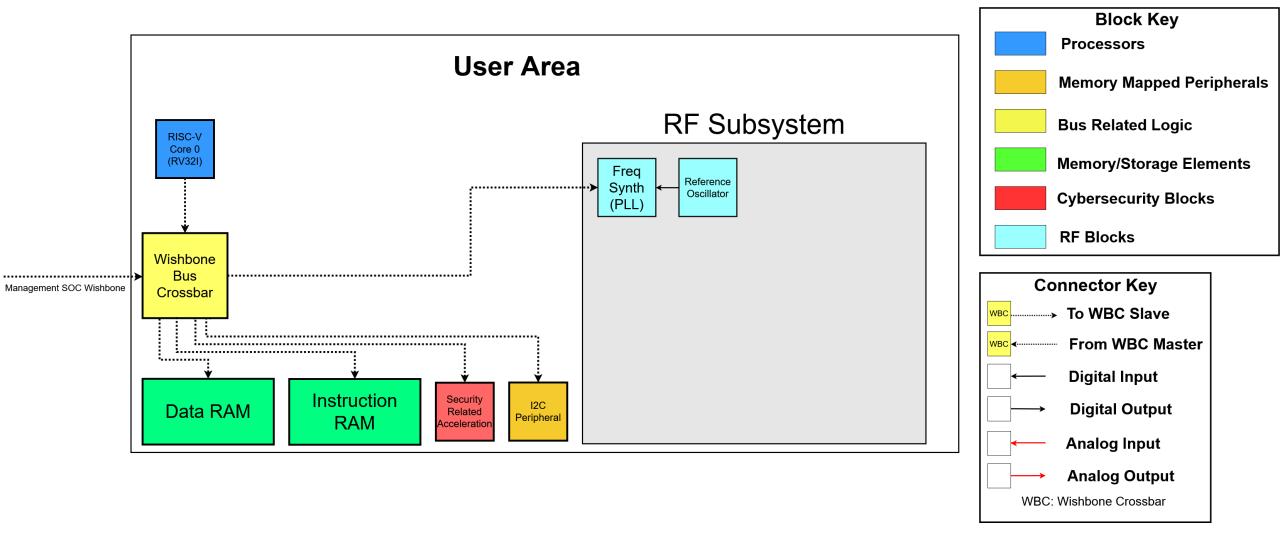
## **Potential Risks and Mitigation**

#### Unknown open-source tools

- Risks
  - Open-source tooling is different than tools team members have experience with
  - Less support and documentation
- Mitigation
  - We have created a design outline for the complete project
  - We have chosen a subset of components to implement
  - Weekly meetings with client/advisor to update on status and keep scope reasonable
- DD 25



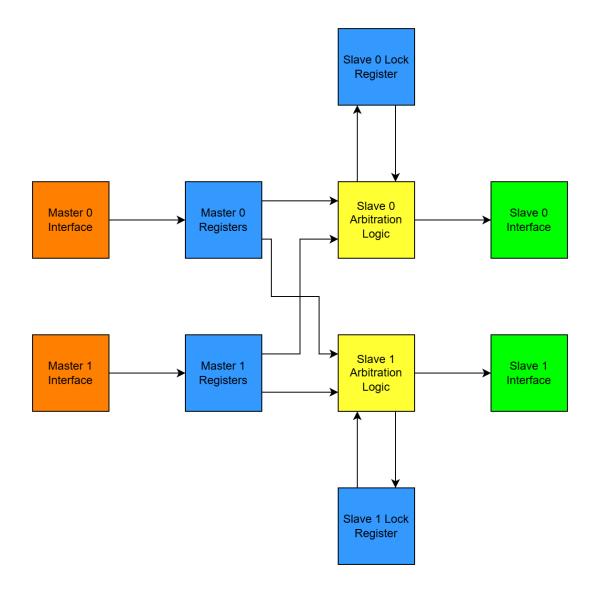
### May 2025 Component Subset



#### DD 57

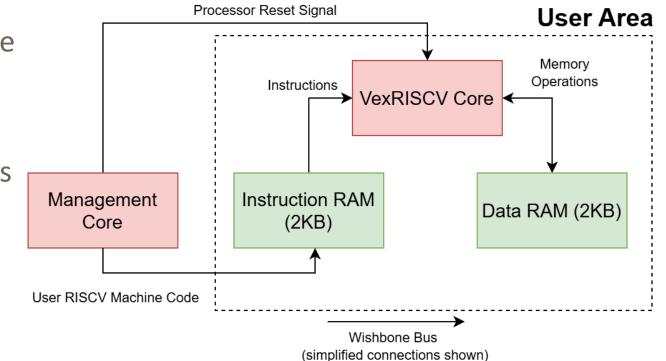
## **Digital Design – Wishbone Crossbar**

- Need a way to control access to memory while also maintaining high performance
- Wishbone crossbar allows parallel access to different modules
- Wishbone crossbar prevents simultaneous access to same module
- Prototyped in first semester as proof of concept
- A script was written to make the crossbar generic, allowing easy expansion to accommodate new modules
- DD 70, 101



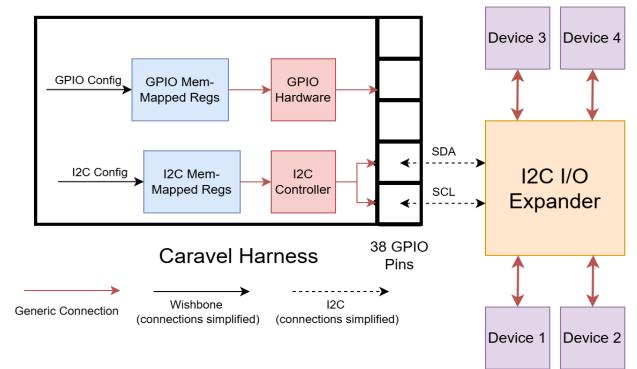
## **Digital Design – RISCV Core and RAM**

- A RISCV core in the user area will run user programs
  - Generated using VexRISCV, which is open source and Wishbone compatible
- Instruction and data RAMs must be available to the user area RISCV core
  - Used existing Efabless DFFRAM macros and wrote a Wishbone slave interface for them
    - 2KB for instruction RAM
    - 2KB for data RAM
- DD 72, 92



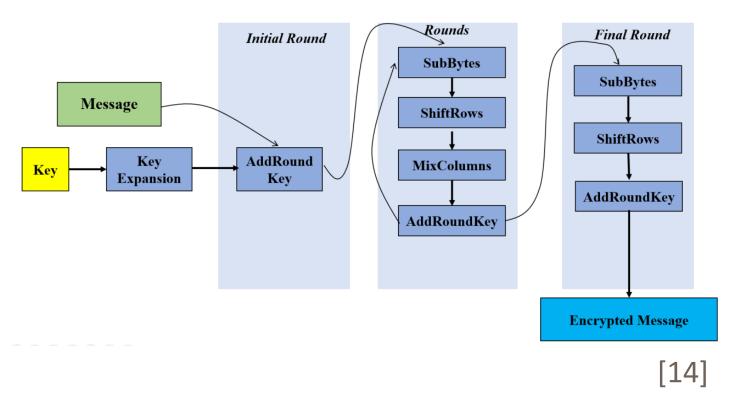
## **Digital Design - Peripherals**

- General purpose input/output (GPIO)
  - Provided by Caravel harness
- Inter-integrated circuit (I2C)
  - Configured via memory-mapped registers
  - Uses GPIO with pullup resistors for SDA and SCL
    - I2C controller implemented in hardware
- DD 70, 92

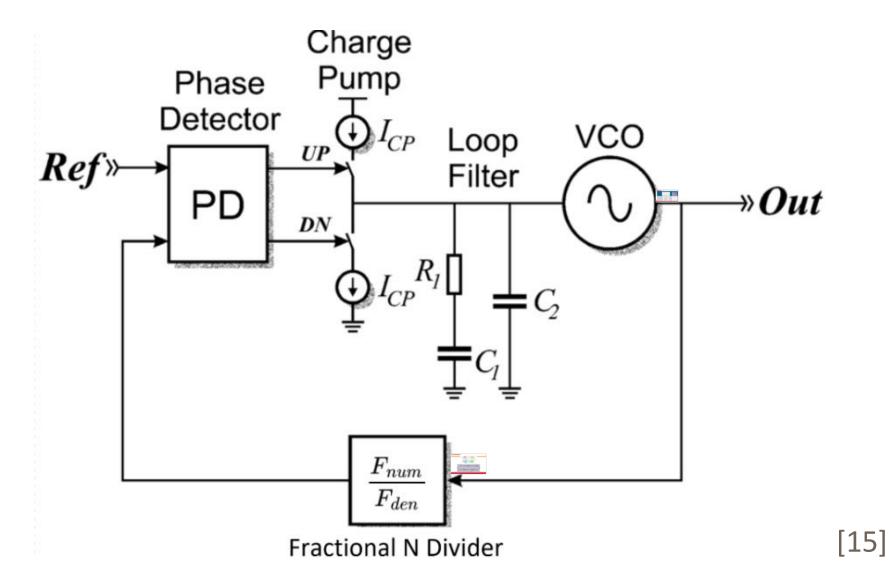


#### **Digital Design – AES Accelerator**

- Evaluated potential security threats in radio microcontrollers (DD 133)
- Encryption is a crucial feature in wireless communication.
- AES was chosen due to its efficiency and being part of ZigBee.
- We chose to implement an open-source option from the Efabless marketplace. (DD 105)



### **Analog Design – Frequency Synthesizer**



## **Unexpected Risk – Efabless Shutdown**

- We found out in late March that Efabless has shutdown
- We will not be able to fabricate our chip
- We still can implement our project and run tests
- Layouts can also still be generated so if Efabless comes back, we will be set to send our design off for fabrication

• DD 31

#### **Shutdown Notice**

[17] & [18]

Due to funding challenges, Efabless has shut down operations until further notice.

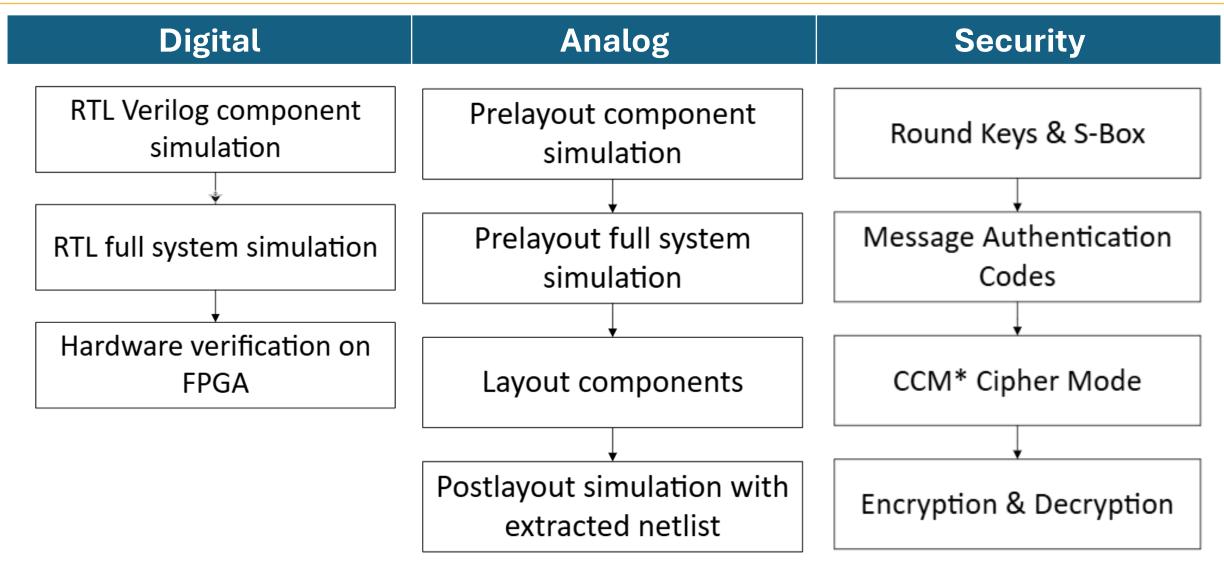
We regret any inconvenience and will provide updates as available.

#### IOWA STATE UNIVERSITY

[16]

efabless:

### Testing Plan DD 65

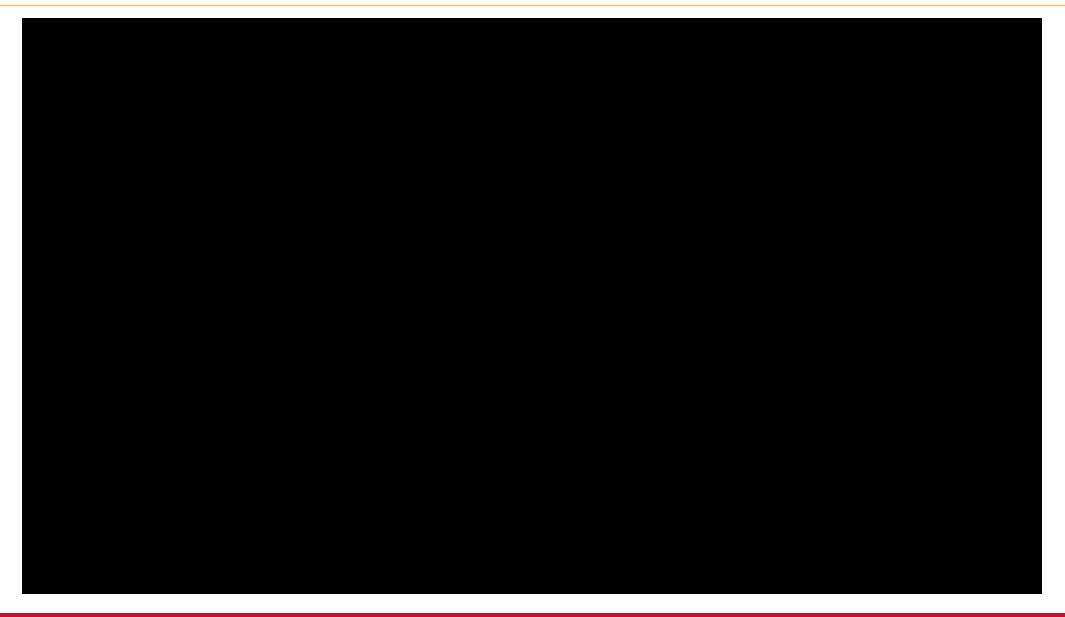


## **Digital Results**

Wishbone Crossbar	Second RISCV Core	DFF RAM	I2C	AES
<ul> <li>Parallel access verified</li> <li>Arbitration verified</li> <li>Address mapping verified</li> </ul>	<ul> <li>Reset verified</li> <li>Program execution verified</li> </ul>	<ul> <li>Reads and writes verified</li> <li>Half word and byte reads and writes verified</li> <li>Confirmed to synthesize and function on FPGA</li> </ul>	<ul> <li>Addressing verified</li> <li>Reads and writes verified</li> <li>Multi-byte reads and writes verified</li> </ul>	<ul> <li>Message Authentication Verified</li> <li>Encryption &amp; Decryption Verified</li> </ul>

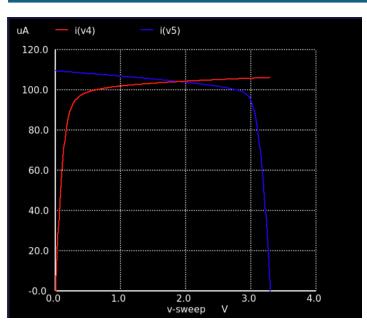
DD 78-86

### **Digital Demo Video**

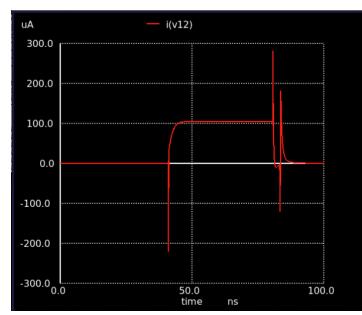


### Analog Results DD 84

**Charge Pump** 



#### PFD and Charge Pump

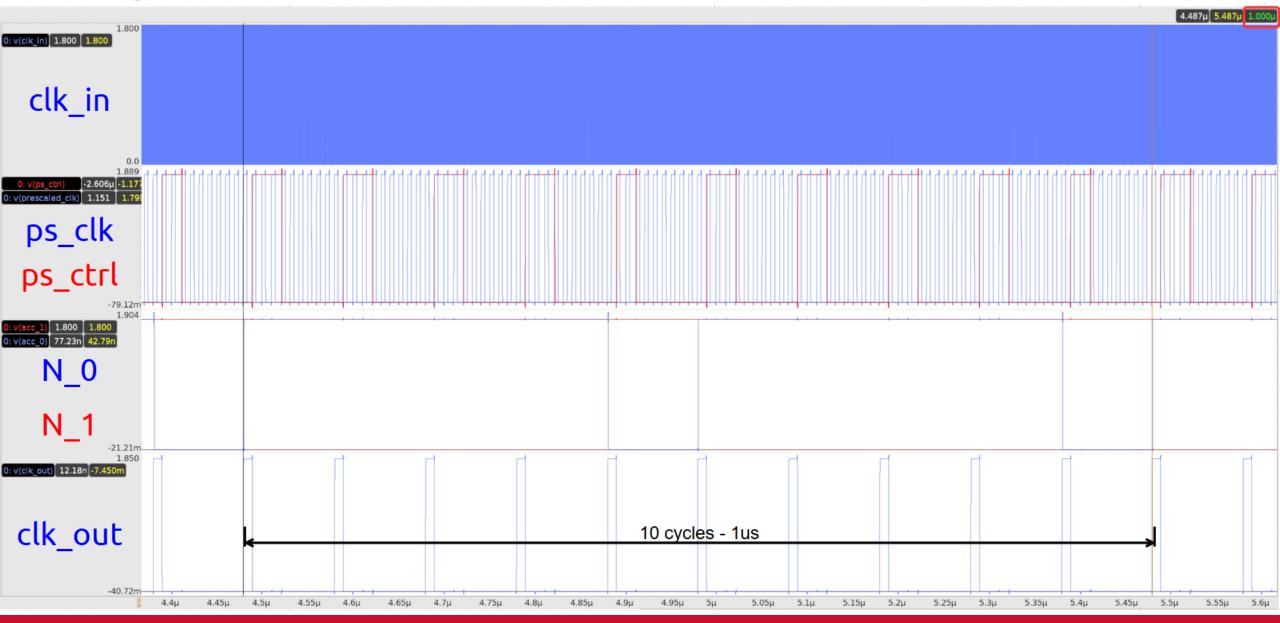


Frequency vs. Vctrl 1e9 Kvco @ Vctrl=0.9V: 3.3075 GHz/V 1.6 1.4 1.2 Frequency (GHz) 0.6 0.4 0.2 0.8 1.0 1.2 1.4 Vctrl (V) Duty Cycle vs. Vctrl 50 40 Duty Cycle (%) 8 20 10 0.8 1.0 1.2 1.4 Vctrl (V)

VCO

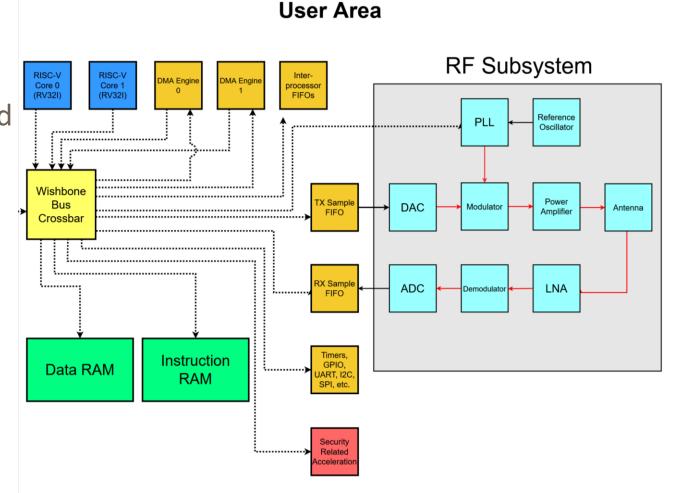
#### sdmay25-27 – Open-Sourced Radio Microcontroller for Fabrication 19

### Analog Results – Fractional Divider DD 87



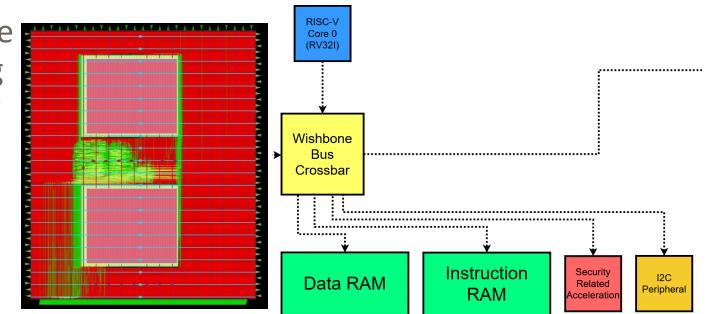
### **Future Work**

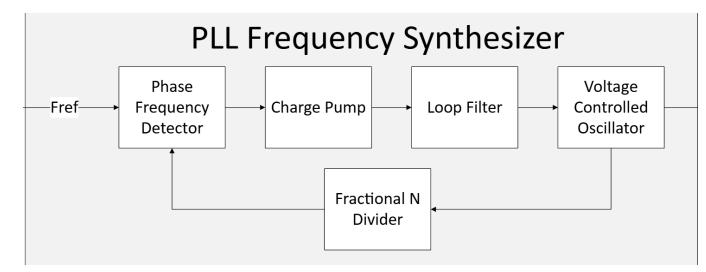
- Continue to expand on the design with the plan of developing it into diagram shown in the detailed design that includes a more well outlined digital and RF subsystem.
- Add and continue to refine documentation as project develops.
- Expand software libraries as digital and analog components are added.
- Integration of both digital and analog subsystems.
- Fabrication of the MCU



## **Conclusions – What We Have Accomplished**

- Substantial progress in creating the components for a fully functioning open-source radio microcontroller
- Second RISCV core
- DFF RAM
- Wishbone Bus Crossbar
- I2C
- AES 128-bit encryption
- Divider
- PFD
- Charge-pump
- VCO





# Questions?

ECPE Senior Design May 2025, Team 27

### **Image References**

[1] "ESP 32" Accessed: 5/6/2025. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/425/MFG\_ESP32-DEVKITC-VE.jpg

[2] "PI Pico" Accessed: 5/6/2025. [Online]. Available: https://cdn-shop.adafruit.com/970x728/5544-02.jpg

[3] "STM" Accessed: 5/6/2025. [Online]. Available: https://newsroom.st.com/wp-content/uploads/2020/12/STM32WL\_MM\_launch\_P4314S\_big.jpg

[4] "TI CC1352" Accessed: 5/6/2025. [Online]. Available: https://www.ti.com/content/dam/ticom/images/products/ic/wireless-connectivity/evm-board/launchxl-cc1352r1-angled.png

[5] "Generic Microcontroller" Accessed: 5/6/2025. [Online]. Available: https://www.electronics-lab.com/wp-content/uploads/2020/06/TQFP-32-Photo.jpg

[6] "Wifi Waves" Accessed: 5/6/2025. [Online]. Available: <u>https://static.vecteezy.com/system/resources/previews/010/518/706/non\_2x/sound-radio-wave-icon-wifi-sound-signal-connection-for-graphic-design-logo-website-social-media-mobile-app-ui-illustration-vector.jpg</u>

[7] "Skywater Logo" Accessed: 5/6/2025. [Online]. Available: https://chambermaster.blob.core.windows.net/images/customers/7798/members/22311/logos/MEMBER\_PAGE\_HEADER/SkyWater\_Logo2.png

[8] "Ngspice Logo" Accessed: 5/6/2025. Available: <u>https://ngspice.sourceforge.io/images/nglogo.jpg</u>

[9] "Icarus Verilog Logo" Accessed: 5/6/2025. [Online]. Available: <u>https://upload.wikimedia.org/wikipedia/en/c/cb/Icarus\_Verilog\_logo2.png</u>

[10] "Caravel Harness Description" Accessed: 5/6/2025. [Online]. Available: <u>https://umsousercontent.com/lib\_CUsguFEVafmoKCKW/6wufw19k0ioc1fwu.png</u>

[11] "GNU Logo" Accessed: 5/6/2025. [Online]. Available: https://mediaresource.sfo2.digitaloceanspaces.com/wp-content/uploads/2024/04/29171315/gnu-logo-146F1DDF52-seeklogo.com.png

[12] "ZigBee Logo" Accessed: 5/6/2025. [Online]. Available: https://www.tuv.com/content-media-files/master-content/services/products/p03-electrical/1163-tuv-rheinland-zigbee-testing-and-certification/tuv-rheinland-zigbee-logo\_core\_1\_x.jpg

[13] "Bluetooth Logo" Accessed: 5/6/2025. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2019/04/cropped-Bluetooth\_FM\_Color-1-223x300.png

[14] "AES Block Diagram" Accessed 5/7/2025. [Online]. Available: https://labs.dese.iisc.ac.in/embeddedlab/aes-encryption/

[15] R. C. H. van de Beek, C. S. Vaucher, D. M. W. Leenaerts, E. A. M. Klumperink, and B. Nauta, "A 2.5-10-GHz clock multiplier unit with 0.22-PS RMS jitter in standard 0.18-/SPL MU/M CMOS," IEEE Journal of Solid-State Circuits, vol. 39, no. 11, pp. 1862–1872, Nov. 2004. doi:10.1109/jssc.2004.835833

[16] "Shutdown Notice." Efabless Notice. 5/6/2025. [Online]. Available: https://efabless.com/notice

[17] "Efabless Logo" Accessed: 5/6/2025. [Online]. Available: https://store.efabless.com/cdn/shop/files/logo\_large\_4fb22d13-6eef-4eac-9cd6-c7cd23939317.png?height=628&pad\_color=ffffff&v=1702973184&width=1200

[18] "Explosion Clipart" Accessed: 5/6/2025. [Online]. Available: https://www.clker.com/cliparts/k/X/G/n/A/J/explosion.svg

### **Extra Slides**

ECPE Senior Design May 2025, Team 27

## **Quantitative Technical Requirements**

- The radio signal's phase noise will be below -41dBm/100Hz at a 32MHz offset per IEEE standard 802.15.4
- The radio signal's spurious emissions will be below -20dBc per IEEE standard 802.15.4
- The design shall contain 4 KiB of RAM accessible to the user
  - 2 KiB for data
  - 2 KiB for instructions for second RISCV core
- All Wishbone masters and slaves will use a 10 MHz reference clock which will be shared with the management SoC
- Die space limited to 2.92 mm x 3.52 mm

### **Security Analysis**

Vulnerabilities	Countermeasures
<ul> <li>Control flow manipulation</li> <li>Fault attacks</li> <li>Side-channel attacks</li> <li>Sniffing data</li> <li>Replay attacks</li> <li>Repairing attacks</li> </ul>	<ul> <li>Encryption</li> <li>Authentication</li> <li>Code obfuscation</li> </ul>

Due to the scope of the project not every possible vulnerability could be addressed. Encryption made the most sense to be the first security measure implemented due to over the air communications being incredibly easy to intercept. As the project continues to develop with future groups, we hope that more of these security measures will be implemented.

### **Crossbar Simulation Waveform**

Signals	Waves		<u></u>																			
Time			l us			2 u.	15				<b>3</b> u.	15				4 us					5 us	
clk=0					TITES.								<b>The second seco</b>		The second s							
wb_m0_ack_o=0	The second se		<b>The second se</b>			<b>T</b>					<b>T</b>											
wb_m0_adr_i[31:0]=0	0000000 30123400	00+ 30123500	00+ 301234	3480 00+ 301235	3580 00+ 30123	3400	00000000										3012340	10 OL	30000000	3012346	100	00000000
wb_m0_cyc_i=0			TITIS.																			
wb_m0_dat_i[31:0]=0	0000000 01234567	00+ DEADBEEF	00+ BA5EBA	3A11 00+ 89ABCD	CDEF 00000000												DEADBEE	F 00	30000000	DEADBEE	TEF	00000000
wb_m0_dat_o[31:0]=0			01+ 00000000	01+ 00000000	01+ 00000000	01+	+ 00000000											01+ 06	30000000		01+	0000000
wb_m0_read_data[31:0] =x: x							01234															
wb_m0_we_i=0																						
wb_m1_ack_o=0												<b>T</b>										
wb_m1_adr_i[31:0]=0	0000000						30123	3400	00+ 30123500	90 01	00+ 3012	23480	00+ 3012358	80	00+ 30123400	9 06	00+ 3012340	10		00+ 3012356	00	0000000
wb_m1_cyc_i=0										<b>T</b>	<b>T</b>					TĨ.					<b>T</b>	
wb_m1_dat_i[31:0]=0	0000000						01234	4567	00+ DEADBEEF	EF OL	00+ BASE	EBA11	00+ 89ABCDE	IEF .	00000000		BA5EBA1	11		00+ BA5EBA1	11	0000000
wb_m1_dat_o[31:0]=0								01+	L+ 00000000	01+ 00	00000000	0 01+	+ 00000000	01+	00000000	01+ 06	00000000		01+	00000000	01+	00000000 01+
wb_m1_read_data[31:0] =x: x			<b>A REAL</b>													خر	0123456	57			<b>T</b>	
wb_m1_stb_i=0																						
wb_m1_we_i=0											Ti.										<b>T</b>	
wb_s0_ack_i=0						<b>T</b>																
wb_s0_adr_o[31:0]=00	70000000 30+	90000000		30+ 00000000		30+	+ 00000000	30+	00000000			30+	+ 00000000			30+ 06	00000000	30+ 00	0+ 30+	00000000	30+	0000000
wb_s0_cyc_o=0																TÍÌ.						
wb_s0_dat_i[31:0]=0	01234567																				<b>T</b>	
wb_s0_dat_o[31:0]=0		1+ 00000000		BA+ 00000000				01+	L+ 00000000			BA+	+ 00000000					DE+ 00	0+ BA+	00000000	DE+	0000000
wb_s0_stb_o=0			<b>A Reg</b>									<b>T</b>										
wb_s0_we_o =0												<b>T</b>										
wb_s1_ack_i=0																						
wb_s1_adr_o[31:0]=0	0000000		30+ 00000000		30+ 00000000					30+ 00	00000000	7		30+	00000000						30+	0000000 30+
wb_s1_cyc_o=0																						
wb_s1_dat_i[31:0]=0	01234567																					
wb_s1_dat_o[31:0]=0			DE+ 00000000		89+ 00000000					DE+ 0	00000000	1		89+	00000000						BA+	00000000 BA+
wb_s1_stb_o=0																						
wb s1 we o=0																						

### **DFF RAM Waveforms**

Signals	Waves													
Time	Ð								1	us				
clk=1													1	
wbmacko=0					L						1			
wb_m_adr_i[31:0]=3	00000000	3000000	00000000	30000004		00000000	300006	08	00000	000	3000000	С		00000000
wb_m_cyc_i=1														
wb_m_dat_i[31:0]=0	00000000			00000001	6	00000000	000006	02	00000	000	0000000	3		00000000
wb_m_dat_o[31:0] =0		00000000		00000000			000006	00			0000000	θ		
wb_m_sel_i[3:0]=F	θ	F	)	F	(	)	F		θ		F			θ
wb_m_stb_i=1														
wb_m_we_i=0														
Signals	Waves													
	154 us											155	us	
Time												100	45	
clk=	1													
wb_m_ack_o=	•0 <b>1</b>													1
wb_m_adr_i[31:0]=	3 30000000	00000000	3000000	4	000000	3000	0008		00000000	) 30	)00000C			00000000
wb_m_cyc_i=	1								]					
wb m dat i[31:0]=	00000000													
wb m dat o[31:0]=	0000000		0000000	00000001	]	0000	0000	00000002	}	- 00	000000	00000	003	}
wb m sel i[3:0]=		Θ	F		Θ	F			0	F				0
wb m stb i=														
wb_m_we_i=														

### **RISC-V Instruction Fetch**

Signals	Waves																	
Time	1802500 ns	1802600 ns	1802700 ns	1802800	ns 1802900 ns	1803	us i	1803100 ns	1803200 n	ns 180330	00 ns 180.	3400 ns	1803500 ns	s 180360	90 ns	1803700 ns	1803800	0 ns
wb_clk_i=1																		
wb_dff_ram_0_ack_i=0																		
wb_dff_ram_0_adr_o[31:0]=3			3000000		0000000		306	00004		0000000	9	ja ja	30000020			00000000		
wb_dff_ram_0_cyc_o=1																		
wb_dff_ram_0_dat_i[31:0]=0			00000000	0200006F			006	00000	00000013			e	00000000	0000111	7			
wb_dff_ram_0_dat_o[31:0] =x			xxxxxxxx		00000000		xxx	xxxxxx		0000000	P		xxxxxxxx			00000000		
wb_dff_ram_0_sel_o[3:0] =F			F		Θ		F			Θ		F				9		
wb_dff_ram_0_stb_o=1																		
wb_dff_ram_0_we_o=0																		
wb_riscv_i_ack_i=0																		
wb_riscv_i_adr_o[31:0]=3					3000004					30000026	7				X	30000024		
wb_riscv_i_cyc_o=1																		
wb_riscv_i_dat_i[31:0]=0	00000000			0200006F	00000000				00000013	0000000	2			0000111	7	00000000		
wb_riscv_i_dat_o[31:0]=x																		
wb_riscv_i_sel_o[3:0]=F	F																	
wb_riscv_i_stb_o=1																		
wb_riscv_i_we_o=0																		

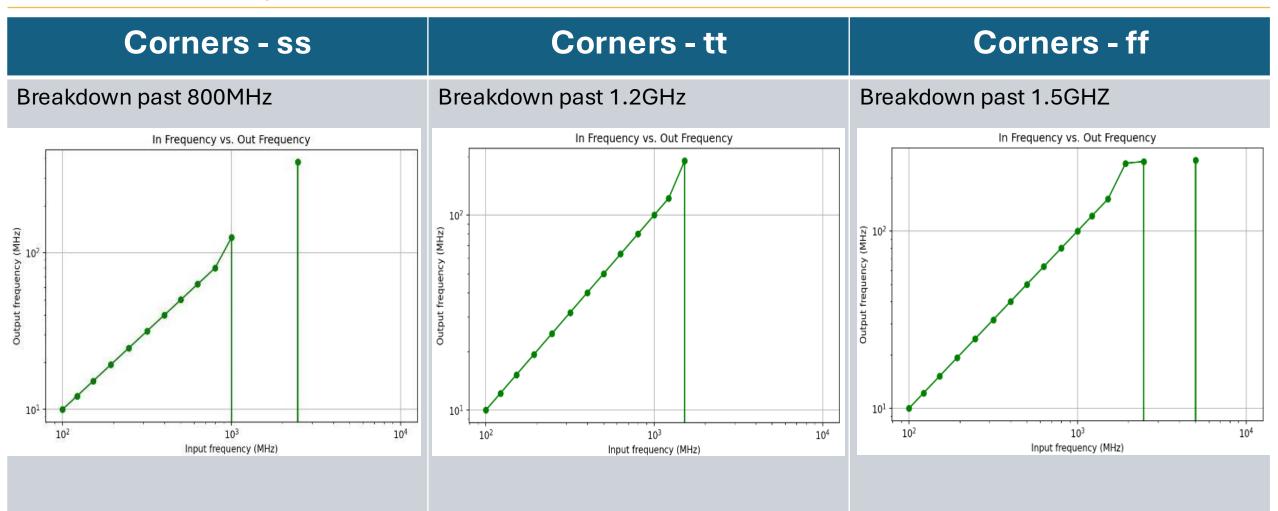
### **RISC-V Data Write**

Signals	Waves		
Time	1815 us	1816	us
wb_clk_i=1			
wb_dff_ram_1_ack_i=z			
wb_dff_ram_1_adr_o[31:0]=0	0000000	30000800	0000000
wb_dff_ram_1_cyc_o=0			
wb_dff_ram_1_dat_i[31:0]=z		00000000	
wb_dff_ram_1_dat_o[31:0]=0	0000000	00000001	0000000
wb_dff_ram_1_sel_o[3:0]=0	0	F	Θ
wb_dff_ram_1_stb_o=0			
wb_dff_ram_1_we_o=0			
wb_riscv_d_ack_i=0			
wb_riscv_d_adr_o[31:0]=3	300+ 00000000 30000800		
wb_riscv_d_cyc_o=1			
wb_riscv_d_dat_i[31:0]=0	0000000		
wb_riscv_d_dat_o[31:0]=0	00000000 000000000000000000000000000000		
wb_riscv_d_sel_o[3:0]=F	1 )2 )1 F		
wb_riscv_d_stb_o=1			
wb_riscv_d_we_o=1			

### **RISC-V Data Read**

Signals	Waves		
Time	2157400 ns 215	57500 ns 2157600 ns 215776	90 ns 2157800 ns
wb_clk_i=1			
<pre>wb_dff_ram_1_ack_i=z</pre>			
wb_dff_ram_1_adr_o[31:0]=0	00000000	30000800	00000000
wb_dff_ram_1_cyc_o=0			
wb_dff_ram_1_dat_i[31:0]=z		0000000 0000000	
wb_dff_ram_1_dat_o[31:0]=0	00000000		
<pre>wb_dff_ram_1_sel_o[3:0] =0</pre>	8	F	0
<pre>wb_dff_ram_1_stb_o=0</pre>			
<pre>wb_dff_ram_1_we_o =0</pre>			
wbs_ack_o=0			
wbs_adr_i[31:0]=3	30000800		
wbs_cyc_i=1			
wbs_dat_i[31:0]=0	0000000		
wbs_dat_o[31:0]=0	0000000	0000000	0000000
wbs_sel_i[3:0]=F	F		
wbs_stb_i=1			
wbs_we_i=0			

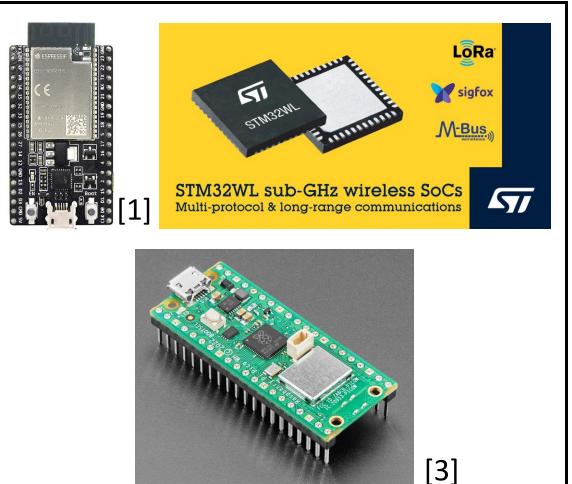
### **Divider Sweep Results**



## **Problem Statement**

- Existing <u>radio</u> microcontroller units (MCUs) have closed source designs
- Difficult for ISU students, ChipForge (ASIC design ISU club) members, and radio hobbyists to learn about how <u>radio</u> MCUs work
- Need an open-source MCU design that can be fabricated (silicon proven)

#### **Closed Source Radio MCUs**

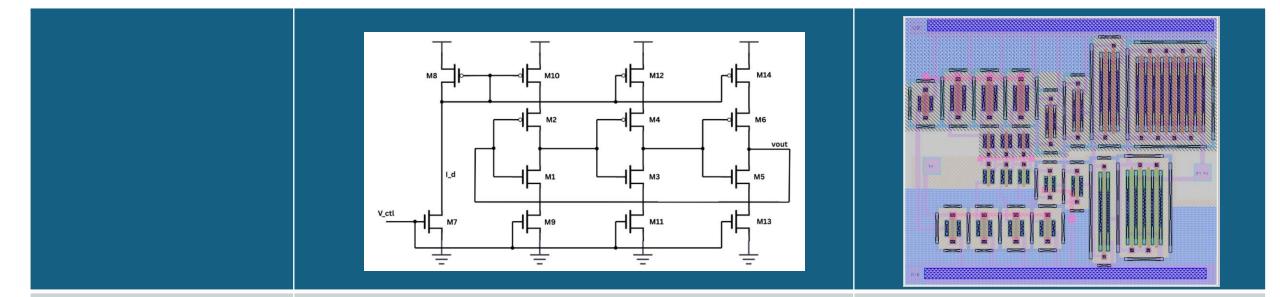


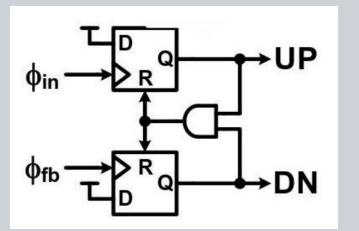
[2]

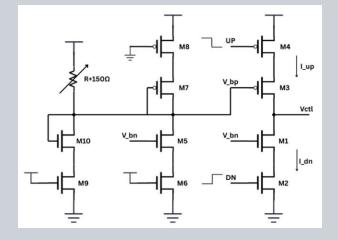
### **Security Testing Results**

Name	Value	0.000000 us	5.000000 us	10.000000 us	15.000000 us	20.000000 us	25 000000 us	30.000000 us	35.000
🕌 dk	1		hninin	hnnnn	hnnn	hnnnn	hhnnn	hnnnn	hn
> 💔 data_in[127:0]	01234567890123456789	•		012	45678901234567	89012345678901			
> 😻 key[127:0]	0000000009876543210	•		0000	00000098765432	10987654321098			
> W data_out[127:0]	000000000000000000000000000000000000000		×		000000000000000000000000000000000000000	boc		856191522bcl	c5tall
> 😻 key_s[127:0]	0000000009876543210	•		0000	00000098765432	10987654321098			
> 👽 key_s0[127:0]	22ca4620225230741042			22ci	46202252307410	42a8024470b89a			
> W key_s1[127:0]	71a6fe3b53f4ce4f43b666	•		71a	5 te3b53 t4ce4 t433	b6664d07c6ded7			
> W key_s2[127:0]	c1bbf0fe924f3eb1d1f958	•		c1b1	f0fe924f3eb1d1	1958fcd63f862b			
> 😻 key_s3[127:0]	bcff01082eb03fb9ff49674			i bef	01082eb03fb9ff	4967452976e16e		1	
> W key_s4[127:0]	94079eadbab7a11445fee	•		940	1 79eadbab7a11445:	tec6516c88273f			
> V key_s5[127:0]	70cbebfdca7c4ae98f828d			1 70c1	ebfdca7c4ae98f	328cb8e30aab87			
> 👽 key_s6[127:0]	57a9fcec9dd5b60512573	•		1 57a	fcec9dd5b60512	573abdf15d913a		1	
> 👽 key_s7[127:0]	9b287c4d06fdca4814aaf			т 9Ъ21		aaf0f5e5f761cf			
> 👽 key_s8[127:0]	e8c7f694ee3a3cdcfa90cd			1 e8c	f694ee3a3cdcfa	90cc291167ade6		1	
> W key_s9[127:0]	5b527854b56844884ff88	•		т 5Ъ53	278545568448841	1888a150912547			
> V r_data_out[127:0]	01234567899955115599			012	45678999551155	99995511559999			
> V r0_data_out(127:0)	35065cc3327d71e6656ff	1000000000		1	35065cc3327d71	e6656110a6950c	c4d5	1	
> V r1_data_out[127:0]	e6c43fb6140e602b21981	3000000000000	00000		e6c43fb614	0e602b219812b7	930 ebét		
> V r2_data_out[127:0]	43a357d86d3057572fdfc	200000000000000000000000000000000000000	0000000000000000		43a357	748643057572£4£	c3055614c98b		
> V r3_data_out[127:0]	c1f9a1cd11fc8b91ac6edt	20000000000	000000000000000000000000000000000000000	0000000	e	1f9alcdllfc8b91	Lace edb02022be	10	
> V r4_data_out[127:0]	e5a68da738083d665190	2000000	000000000000000000000000000000000000000	00000000000		e\$a69da7380	834 65190cb822	491c331	

### VCO, PFD, CP







### **Fractional Divider**

