

Project Presentation

Project name: AES-256 Accelerator for custom processor

Project number: 22-1-1-2495

Students' names:

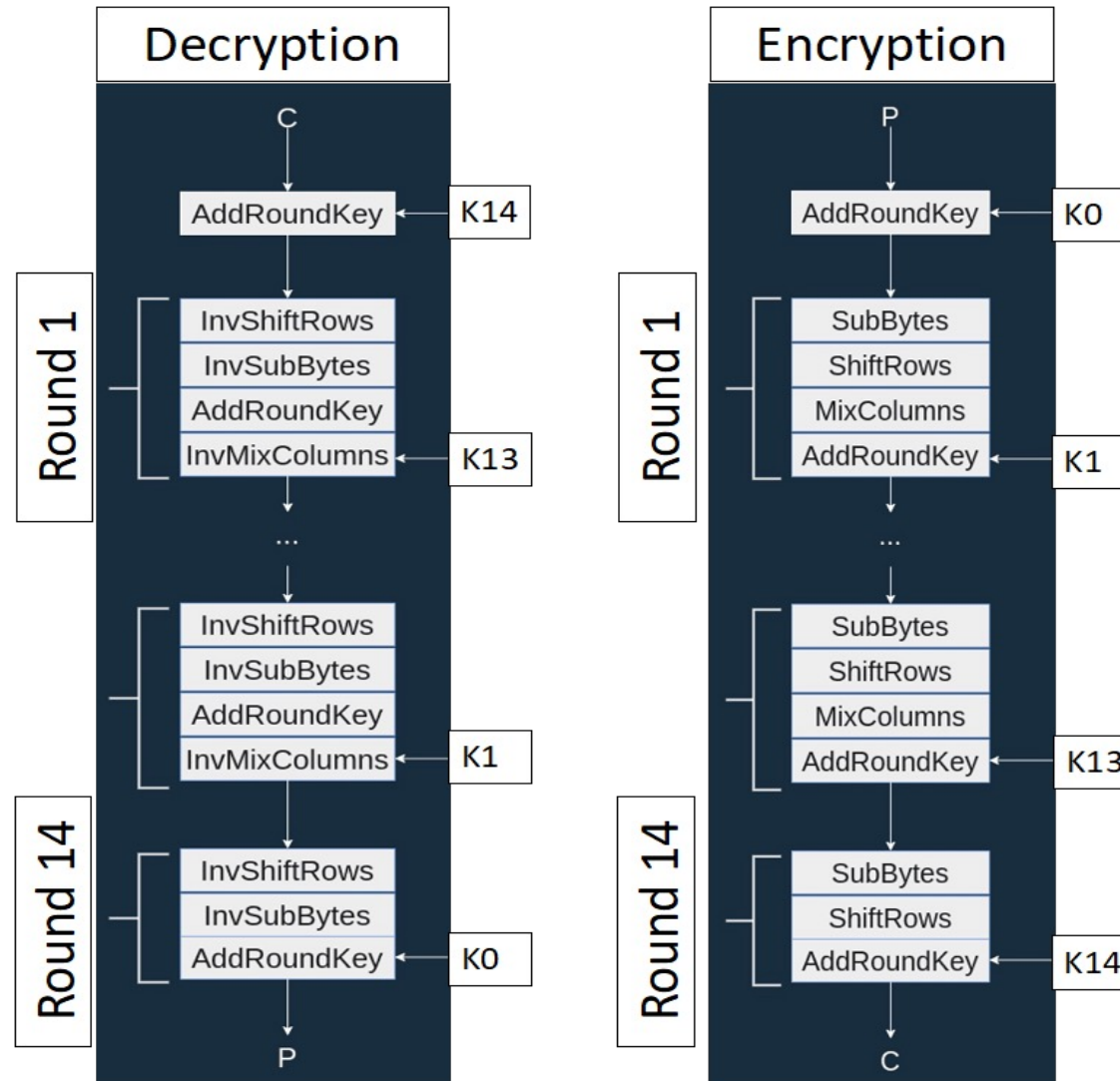
Tzvi Steinberg	209924711
Shahar Levi	313470593

Supervisor: Oren Ganon

Project Carried Out at: University

Supervisor's approval for the presentation: _____

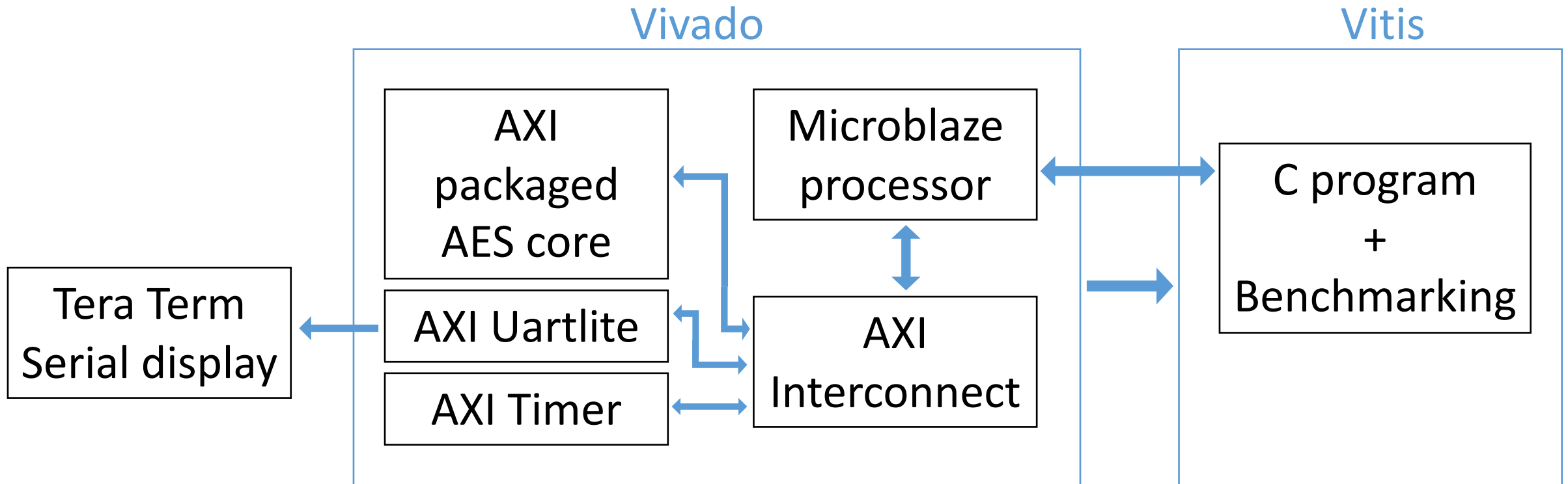
Introduction – Algorithm Structure



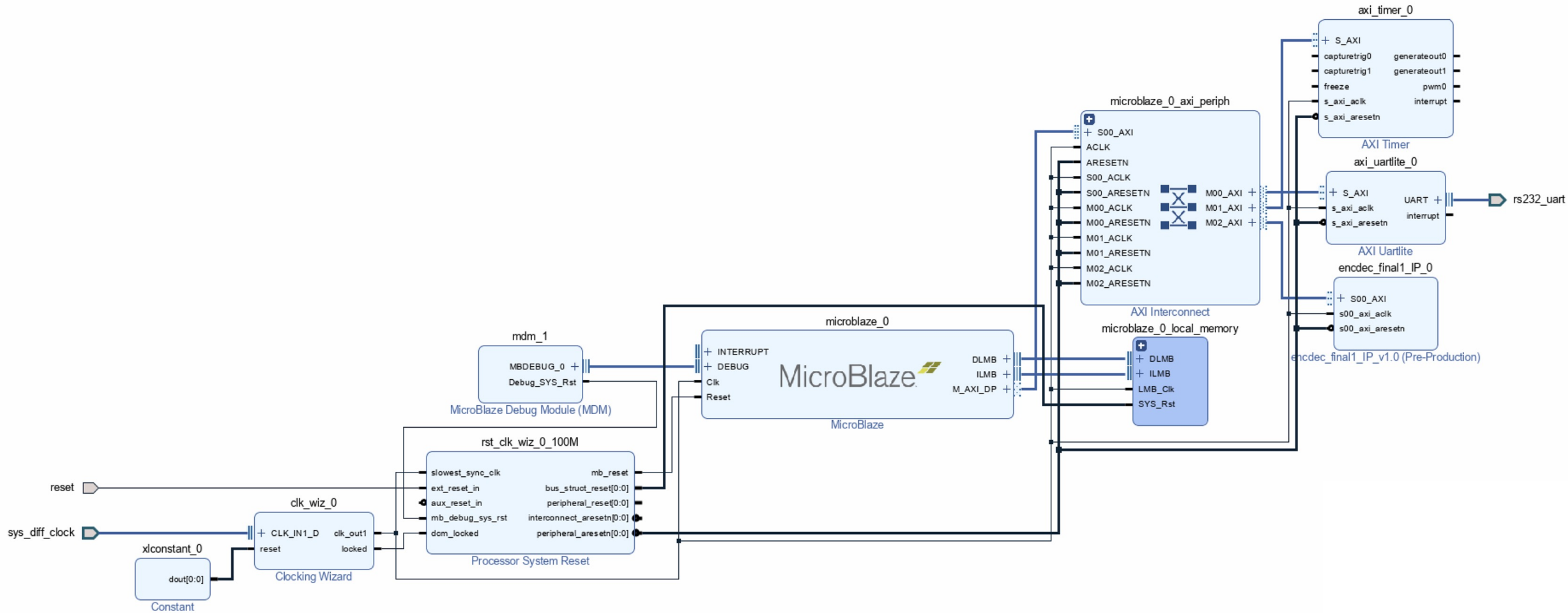
What we did

- Studied the steps
- Reviewed options to parallel and to improve the algorithm while learning about the AXI connection
- Created naive implementation
- Planned multiple designs with different levels of hardware/software balance
- Run simulations, synthesis and implementations
- Analyze results and designed improved implementations

System block diagram



Vivado's block diagram



Total Implementations

"Only C"

1

2

3

4



- Software
- This is the benchmark we wish to accelerate

- Hardware
- Loop unrolling
- Supports pipelining

- Hardware
- Single round module
- Reduction in area and power consumption

- Hybrid
- Key expansion offloaded to software
- Increases transfer delays

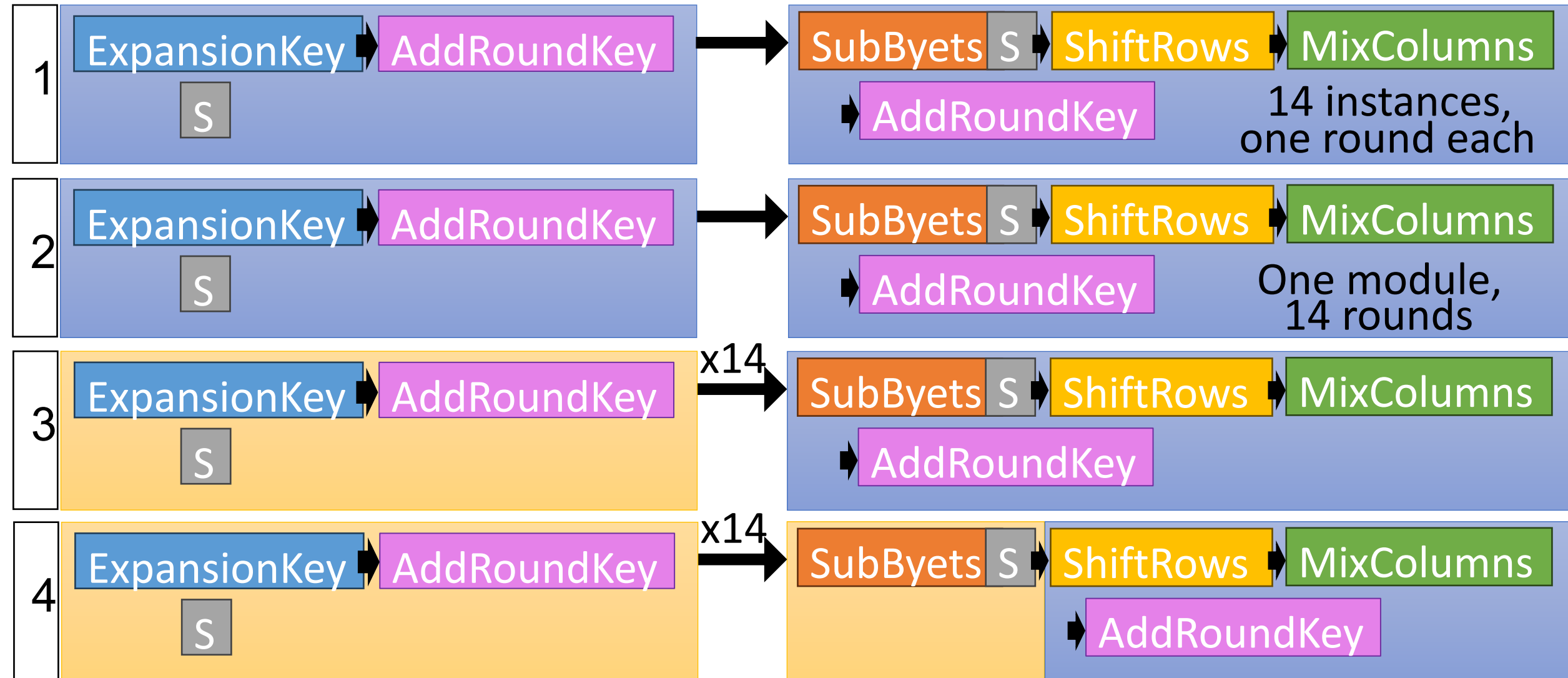
- Hybrid
- Offloads SubBytes transformation to software

Software 
Hardware 

Encryption implementations diagram



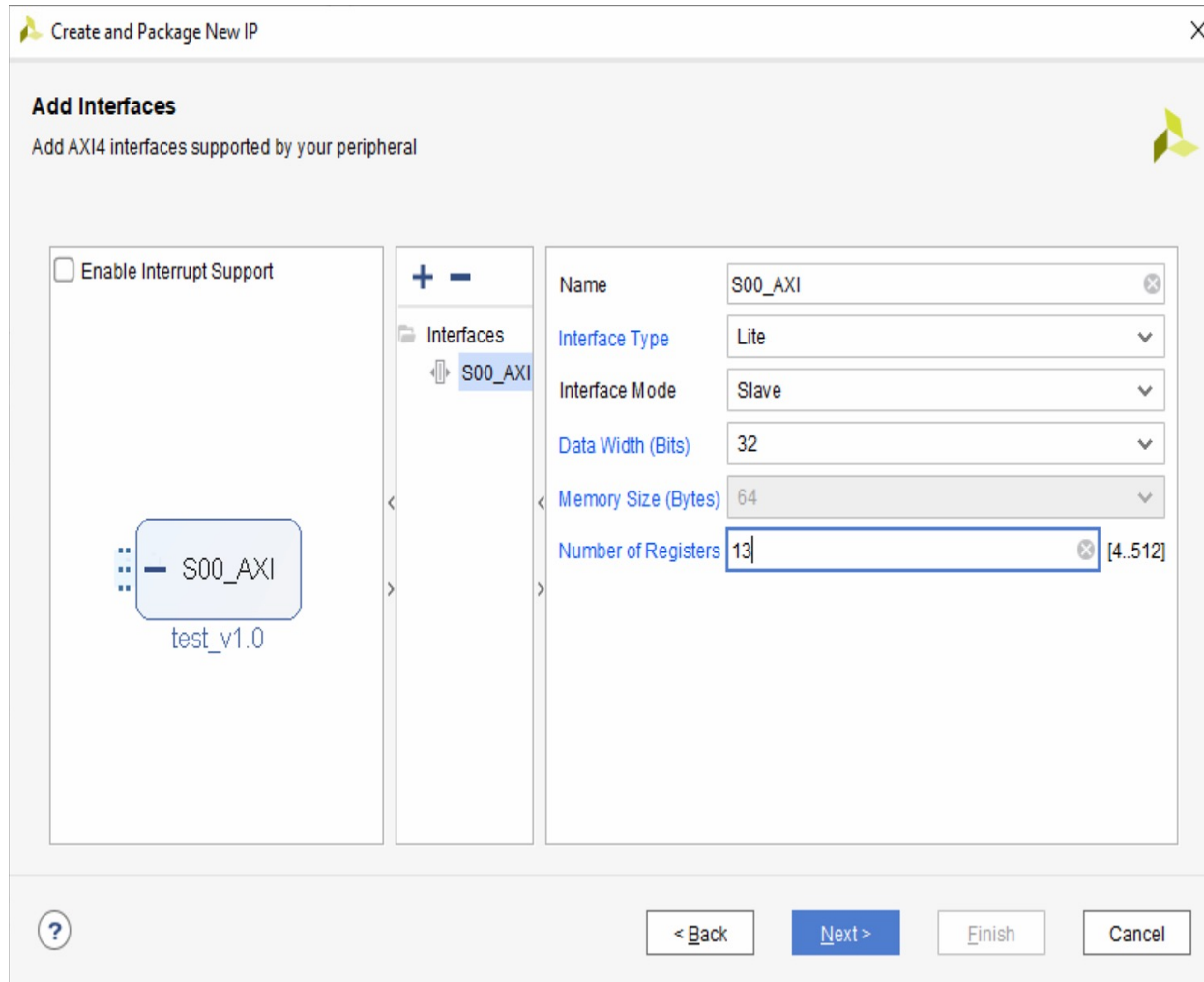
The Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University



*Final round doesn't include MixColumns

*S = Sbox LUT

Implementations - custom IP



Create and Package New IP

Add Interfaces
Add AXI4 interfaces supported by your peripheral

☐ Enable Interrupt Support

Interfaces

- S00_AXI

S00_AXI Configuration:

- Name: S00_AXI
- Interface Type: Lite
- Interface Mode: Slave
- Data Width (Bits): 32
- Memory Size (Bytes): 64
- Number of Registers: 13 [4..512]

test_v1.0

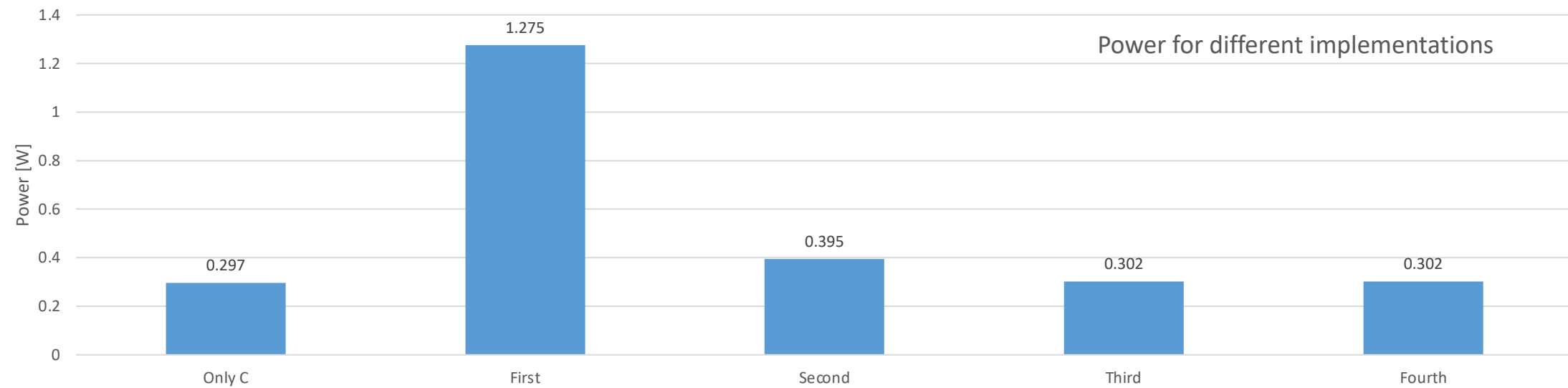
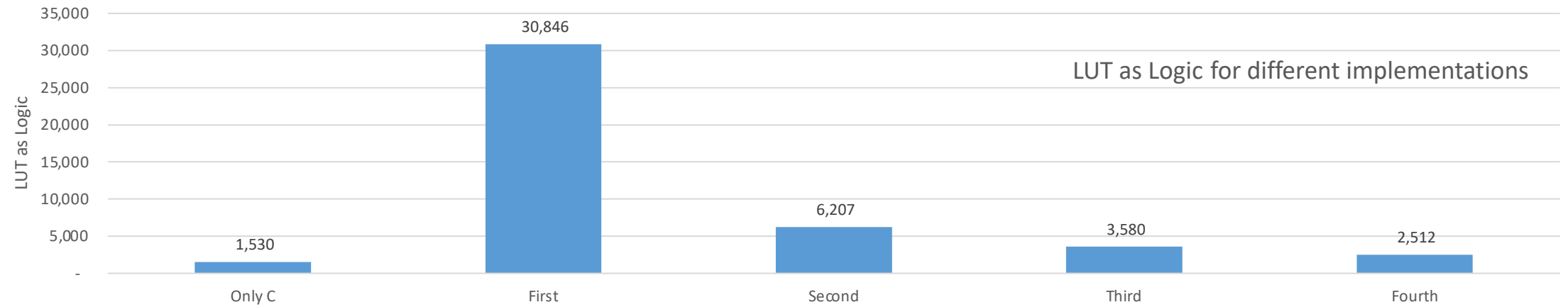
Buttons: ? < Back Next > Finish Cancel

- "Only C" – without IP
- 1 + 2 - 18 registers: 4 for input output each, 8 for key, 1 for parametes (load, reset, enc_en), 1 for done
- 3 + 4 - 13 registers: 4 for input, key and output each, 1 for parameters (enc_en, f_rnd_en)

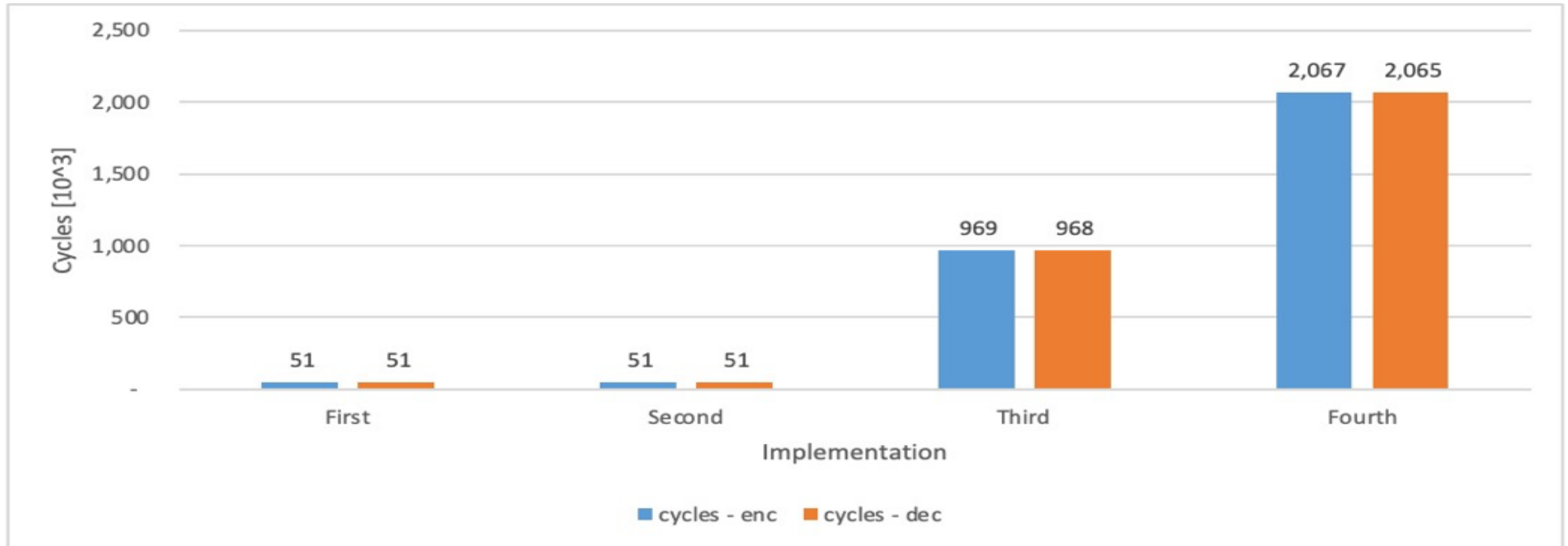
Real Time results

	LUT as Logic	power w	length	cycles - enc	cycles - dec
only c	1530	0.297	100	92,993,825	93,039,875
			200	185,987,650	186,079,750
			300	278,981,475	279,119,625
			400	371,975,300	372,159,500
First	30846	1.275	100	12,925	12,925
			200	25,450	25,450
			300	37,975	37,975
			400	50,500	50,500
Second	6207	0.395	100	12,925	12,925
			200	25,450	25,450
			300	37,975	37,975
			400	50,500	50,500
Third	3580	0.302	100	248,665	248,315
			200	488,815	488,115
			300	728,965	727,915
			400	969,115	967,715
Fourth	2512	0.302	100	523,065	522,715
			200	1,037,615	1,036,915
			300	1,552,165	1,551,115
			400	2,066,715	2,065,315

Comparison of results

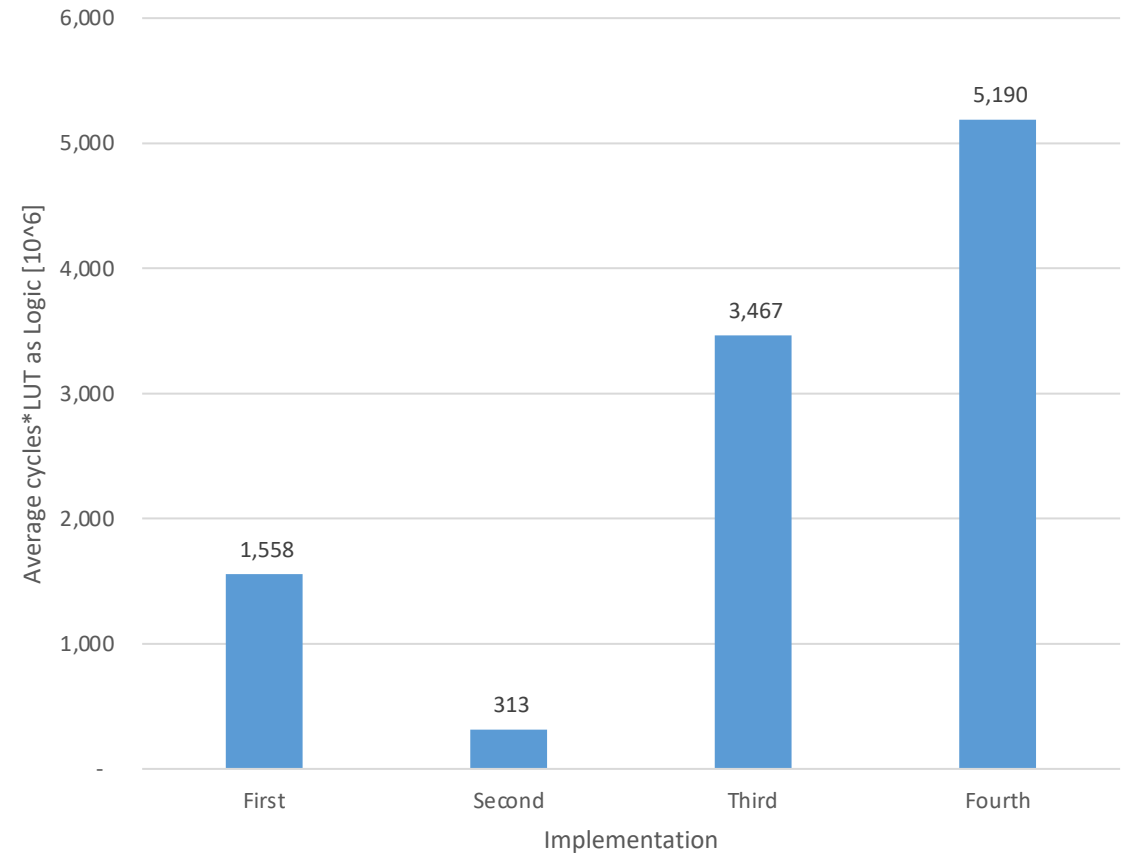
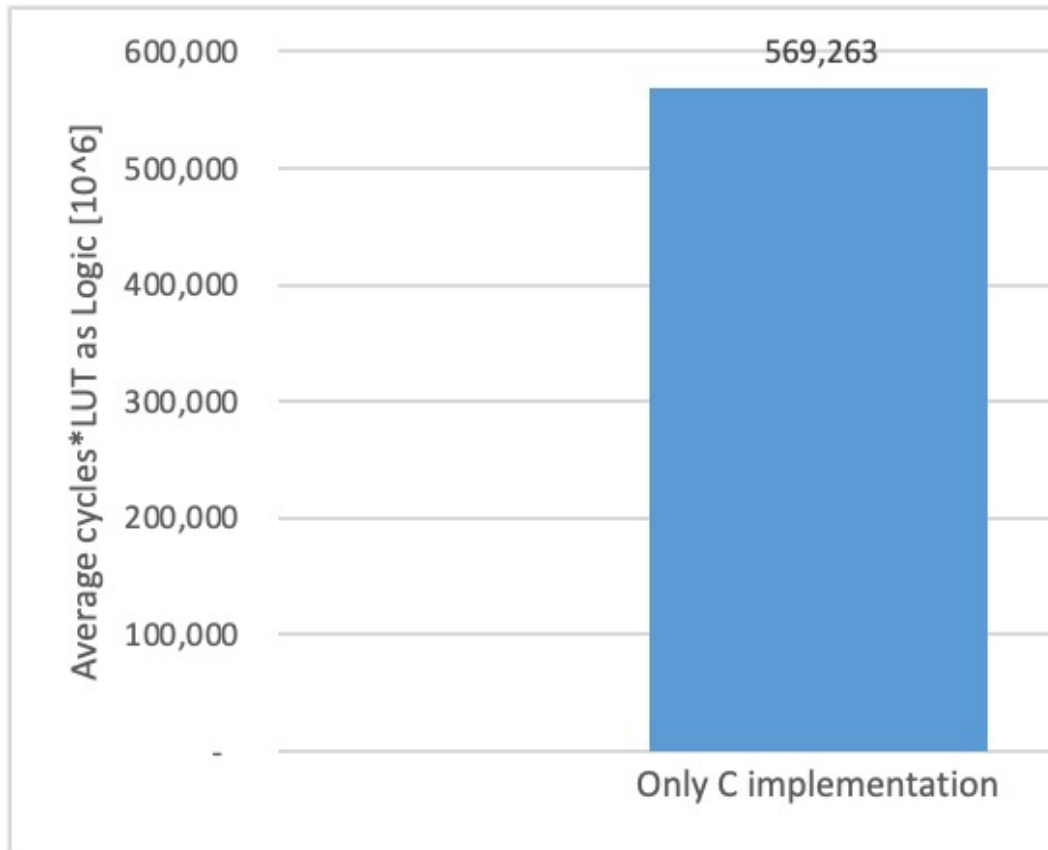


Comparison of results



Results for a data size of 400 words

Tradeoff comparison – minimizing $(\text{clock cycles}) \cdot (\text{area use})$



Conclusions

- Four designs were created and evaluated.
- Second implementation was deemed the best.
- Several goals were not met:
 - TIE Extension functionality and pipelining – due to time constraints.
 - Encryption and decryption of a 128-bit block in a single clock cycle.

Further work

- Adding TIE custom instructions
- AXI improvements
- Pipelining
- Key expansion storage

Project documentation

Documentation uploaded to Github. Includes annotated Verilog and C files stored and categorized by design.

Explanatory notes and simulation results are uploaded there as well.

https://github.com/tzvins/AES_256_hardware_acceleration.git