Serial RISC-V (SERV) with SPI memory access

Kavish Ranawella
Bhasitha Dharmasena

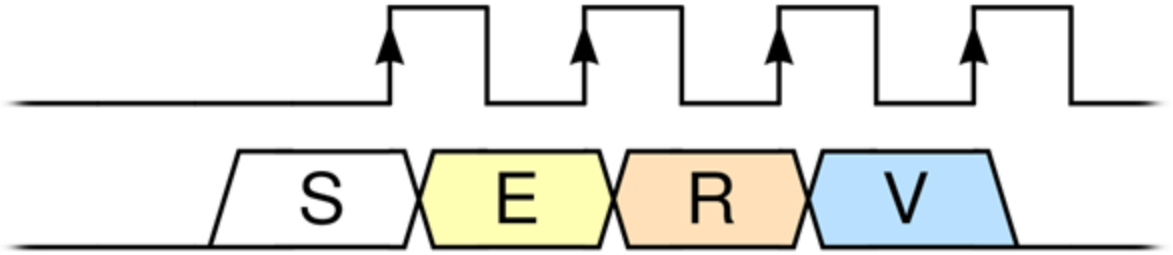
May 02 2025

SERV

- SERV is a 32 **bit-serial** RISC-V core, the world's **smallest RISC-V CPU**
- Open source (under BSD license)
- Wishbone interface for Data and Instruction Buses
- Compatible with Zephyr OS (light-weight, open-source OS by Linux Foundation)

Servant

- Reference platform for SERV
- Contains memory, GPIO and timers
- Bus interface with Wishbone



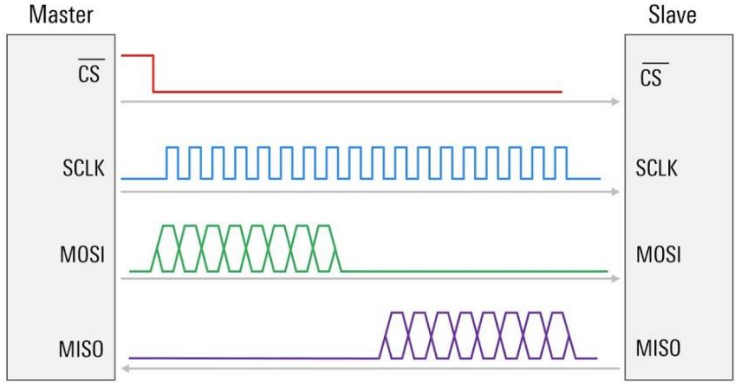
SPI (Serial Peripheral Interface) and Wishbone Interface

Wishbone

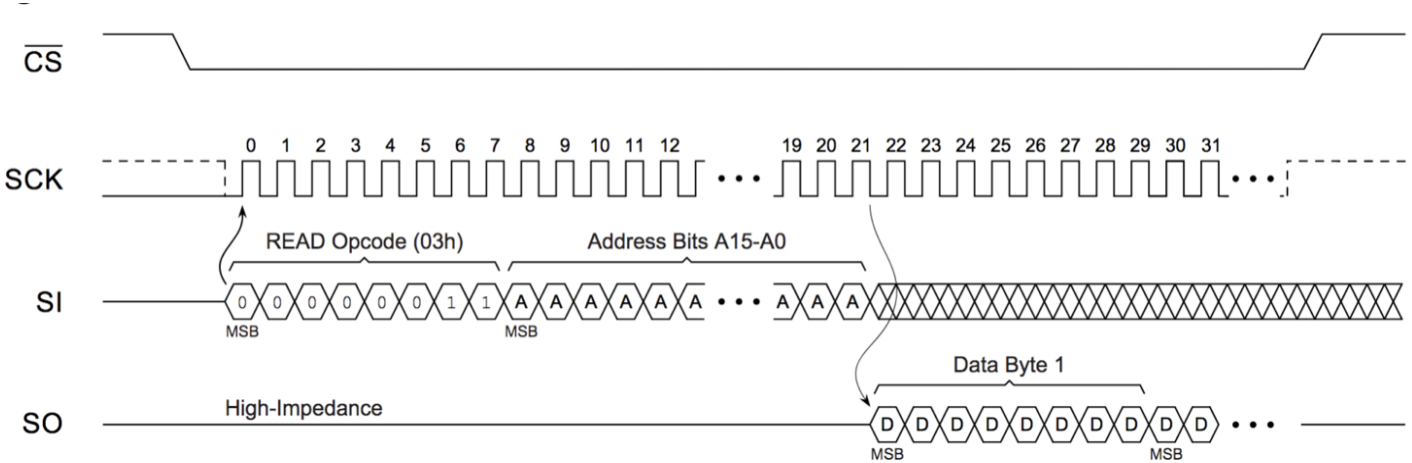
- A parallel synchronous protocol
- Relatively high speed - **Can access 1 word using 1 clock cycle**
- Requires high wire count **(100+ in total)**

SPI

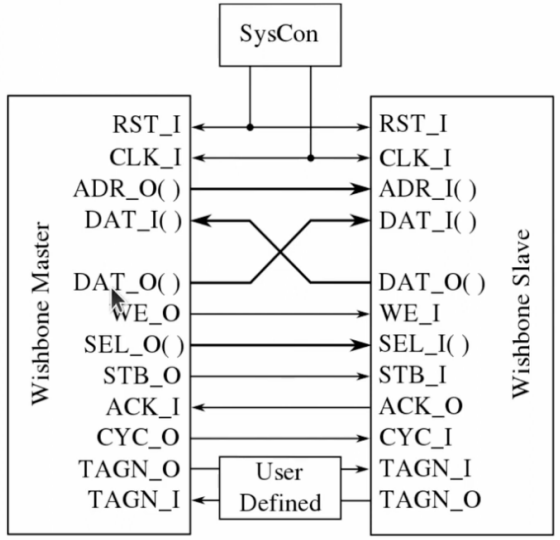
- A synchronized serial communication protocol
- Can integrate with **4 wires total**
- A Master-Slave Architecture
- Relatively slow - **require 64 clock cycles to access 1 word**



SPI protocol



SPI protocol



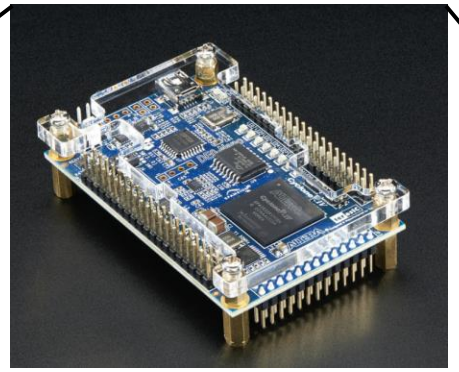
Wishbone protocol

** Rohde & Schwarz
** netburner.com
** wikipedia

Why integrate SPI with SERV?

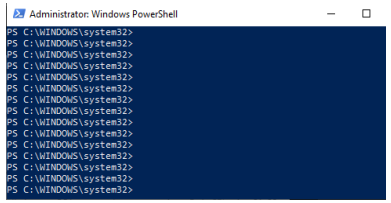
- **Decouples memory from CPU core**: Enables flexible memory placement and simplifies physical design for fabrication
- **Minimal pin count**: Communicates with external RAM using only 4 wires (MISO, MOSI, SCK, CS), reducing I/O complexity
- **Aligns with SERV's bit-serial philosophy**: Maintains SERV's ultra-minimal, bit-serial architecture by extending serial design principles to memory access
- **Shrinks logic footprint**: Removes internal RAM, reducing FPGA resource usage and improving area efficiency for ASIC targets

SPI SERV with FPGA



Altera DE0-Nano

SPI FRAM



NIOS-II Terminal

NIOS-II CPU

UART TX decoder

JTAG

UART RX

timer

gpio

mux

Data

servile

Mem

I-D RAM

RF

RF RAM

Servant

Wishbone Slave

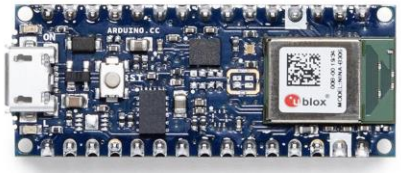
SPI Master

WB-SPI convertor

SPI Slave

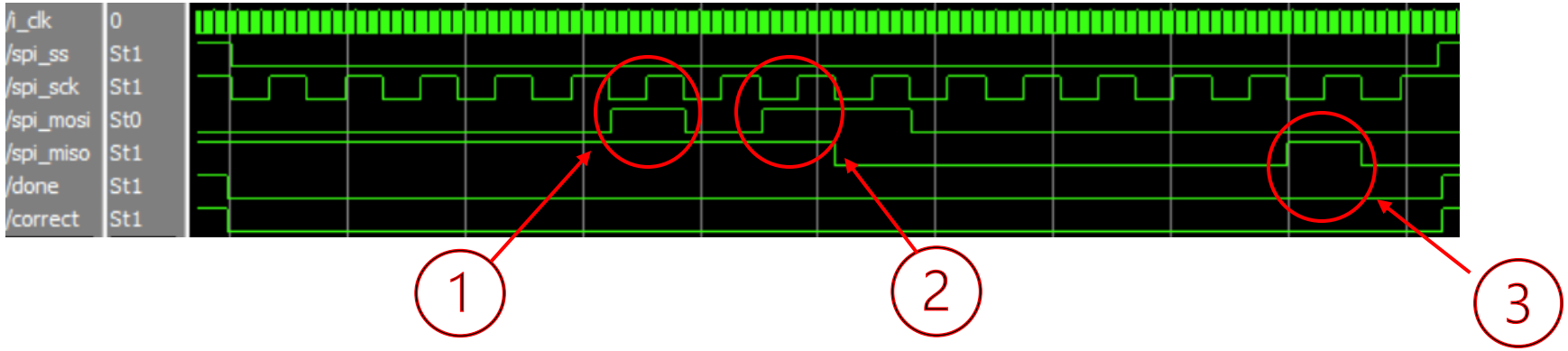
FRAM

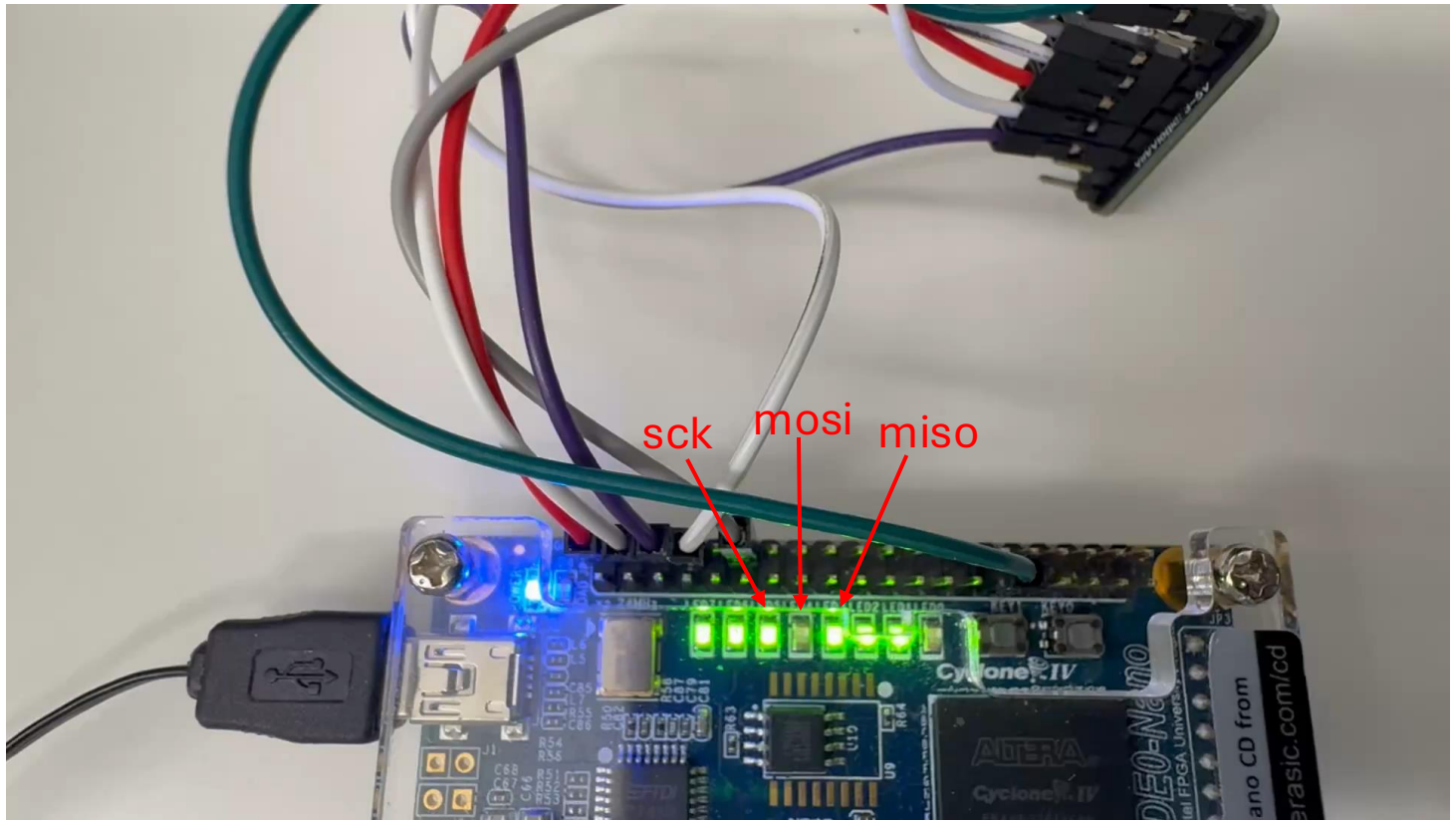
FPGA



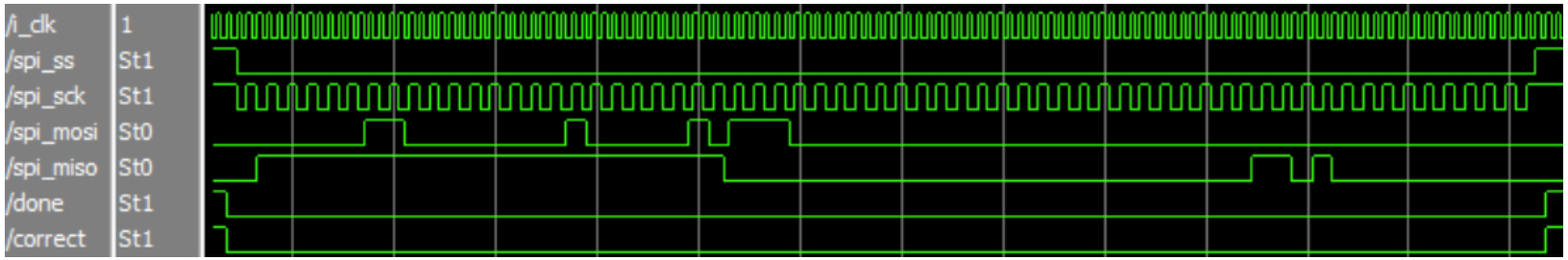
Nano 33 BLE

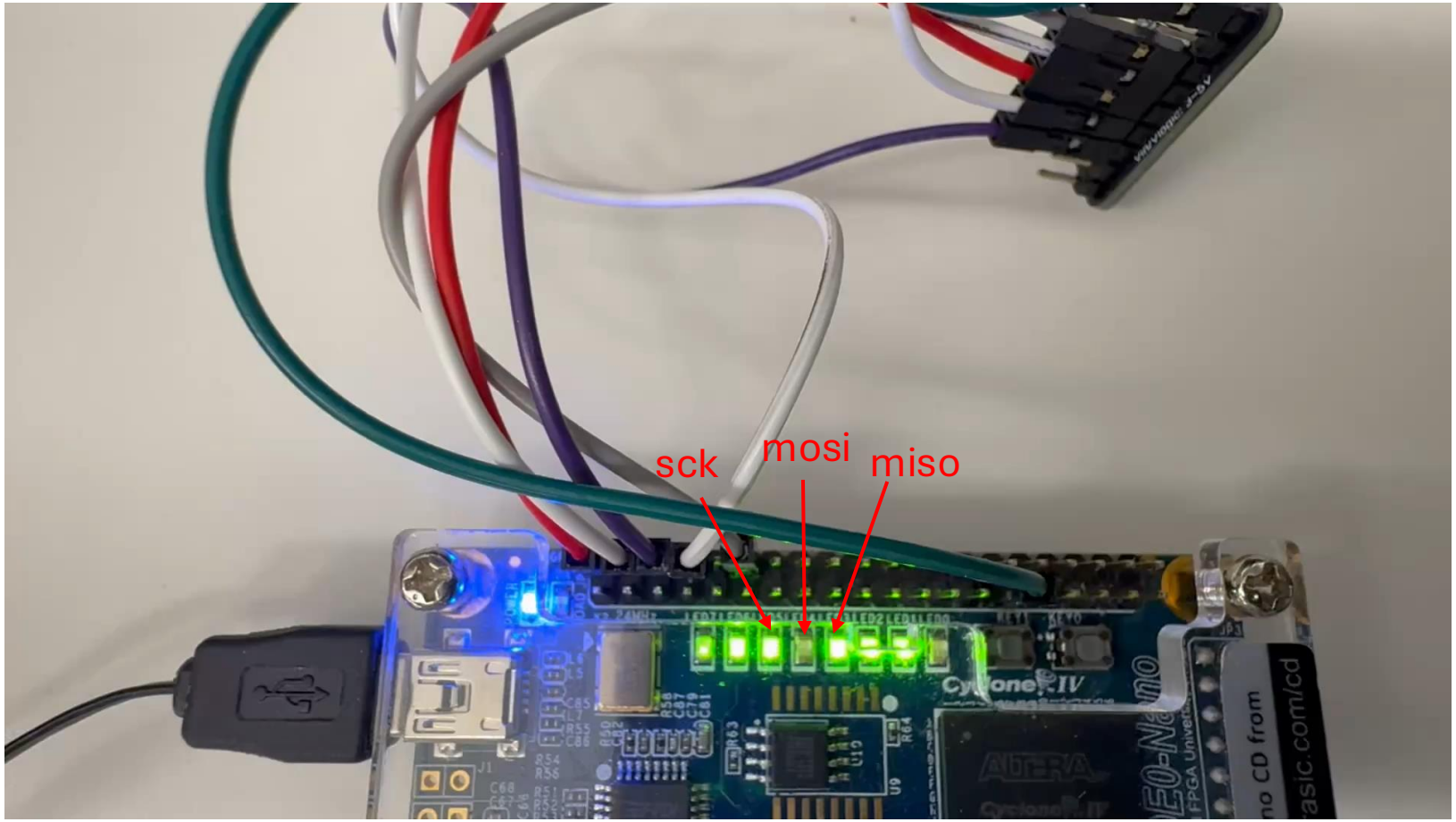
SPI memory accessing – status register read!



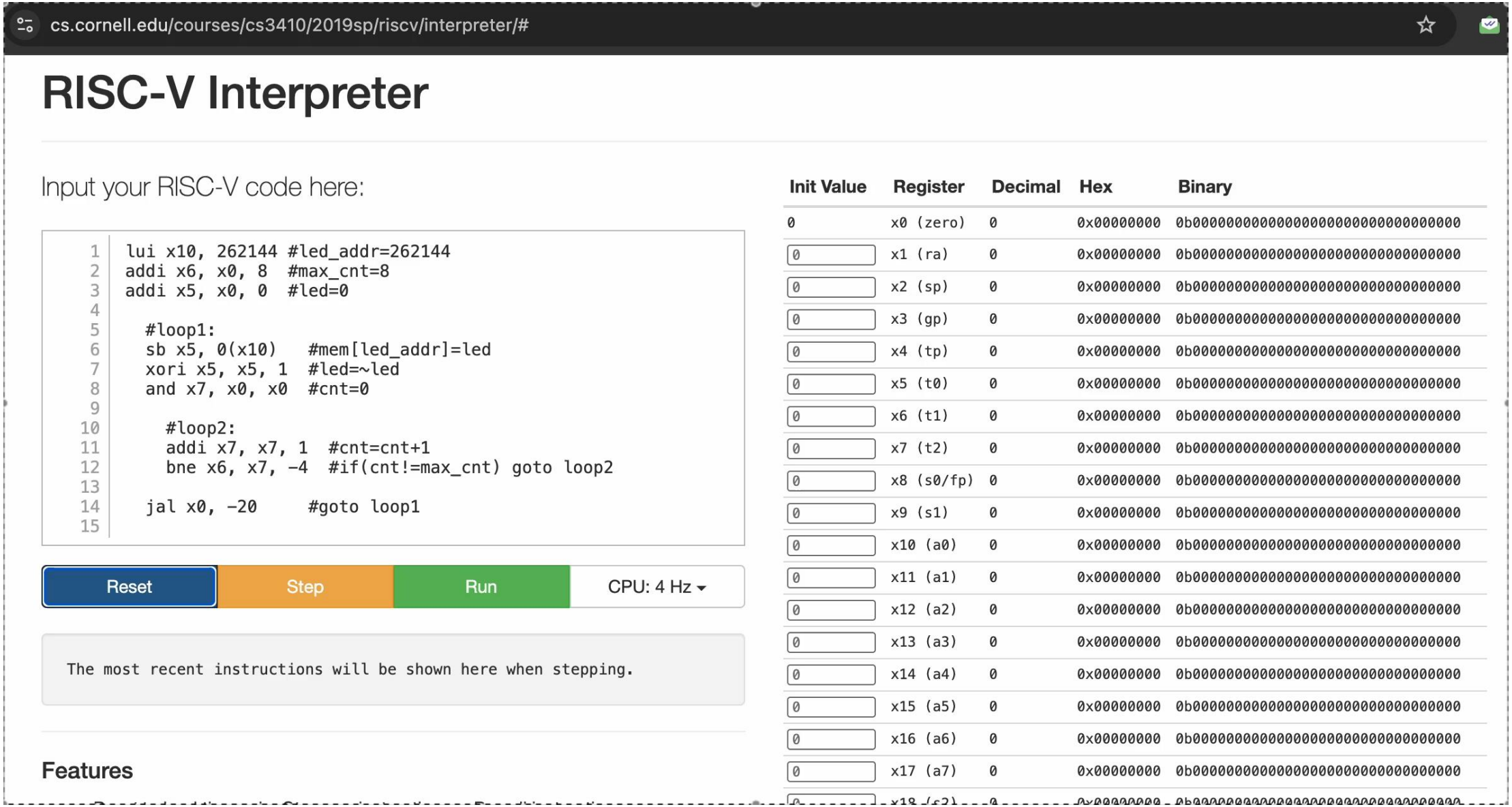


SPI memory accessing – Full Read!



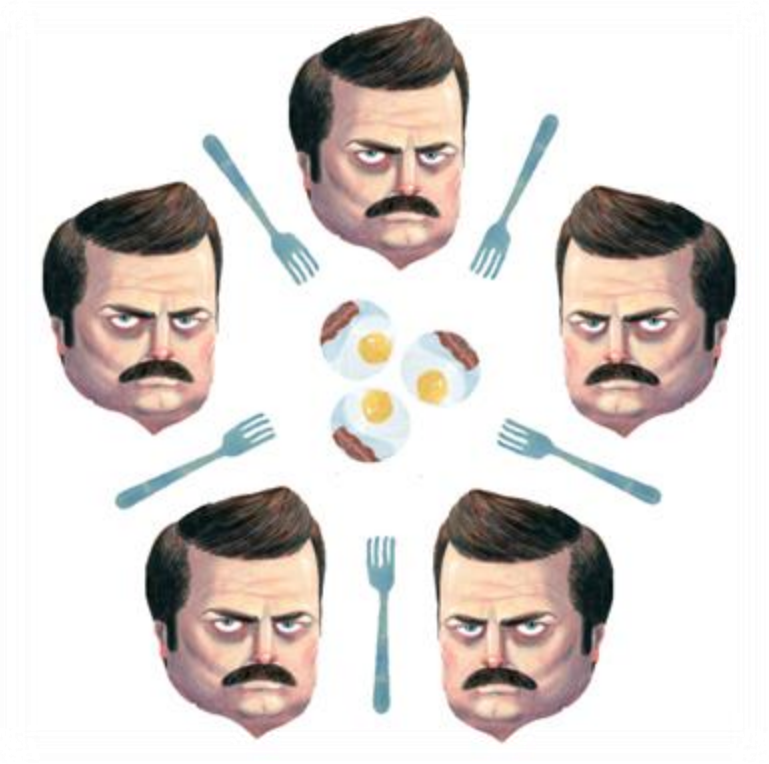


Video Demo – RISC-V Interpreter



The Dining Philosopher Problem

Scenario

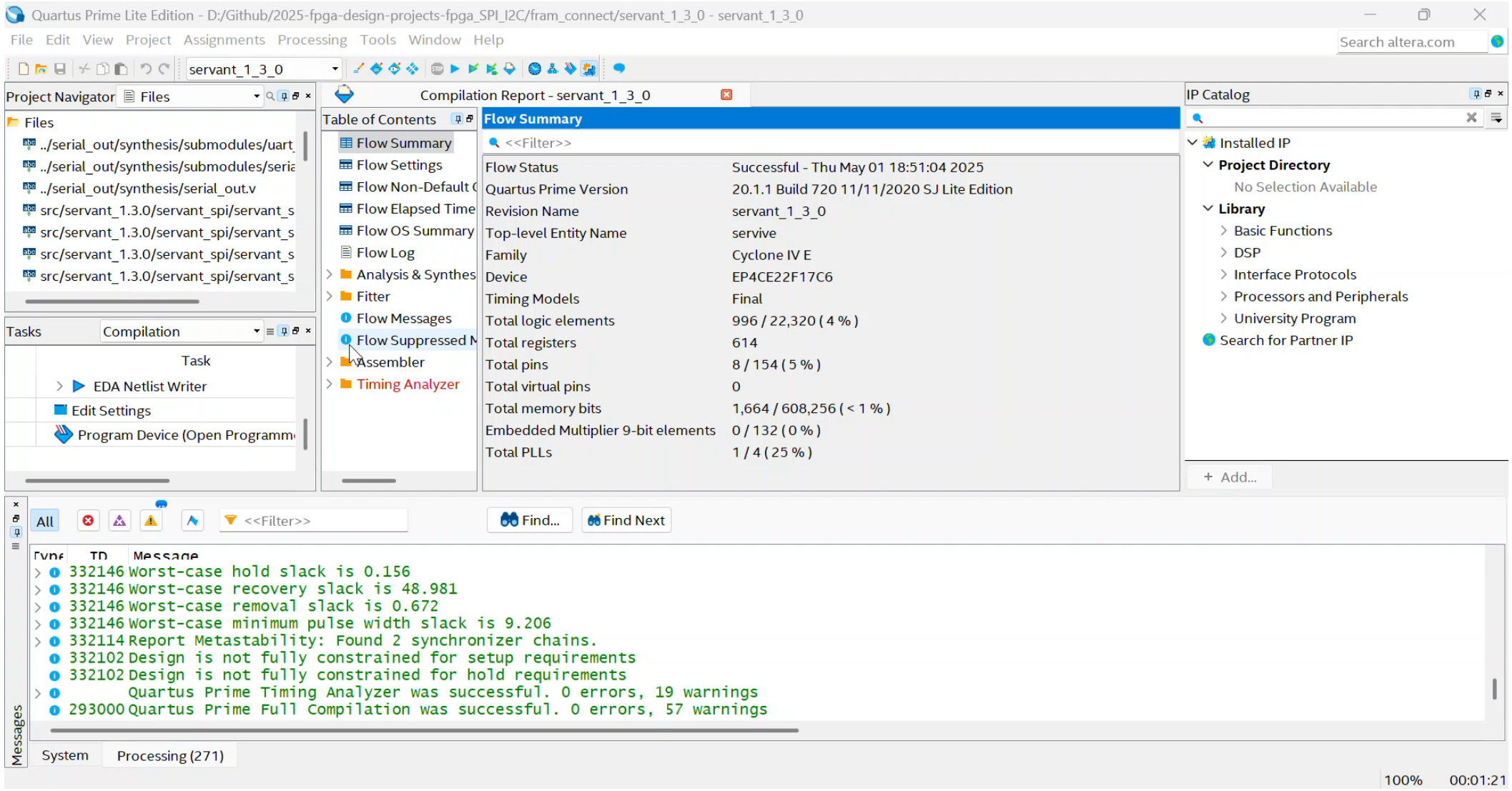


- There are **five philosophers** sitting around a circular dining table.
- Each philosopher does only two things:
    - **Think**
    - **Eat**
- Between each pair of philosophers is **one fork** (total of five forks).
- To eat, a philosopher needs **both the left and right fork**.

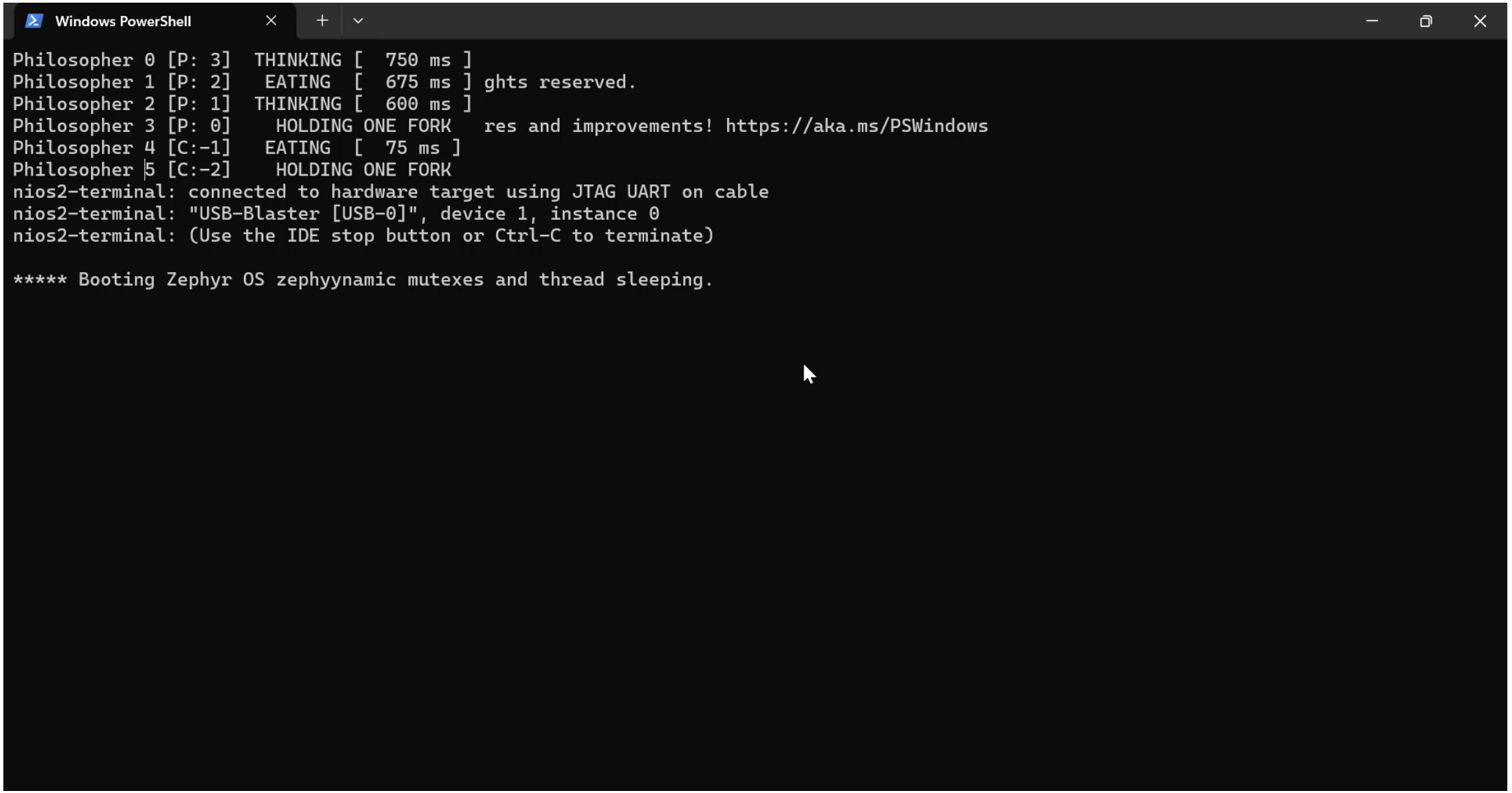
Rules

- **A philosopher must pick up both the left and right forks to eat.**
- **A philosopher can only use the two forks that are next to them.**
- **A philosopher must put down both forks after eating.**
- Philosophers alternate between **thinking and eating**.
- Forks are **shared resources** and can be held by only one philosopher at a time.

Video Demo – The Dining Philosopher Problem!



Error!! - means it works as expected!



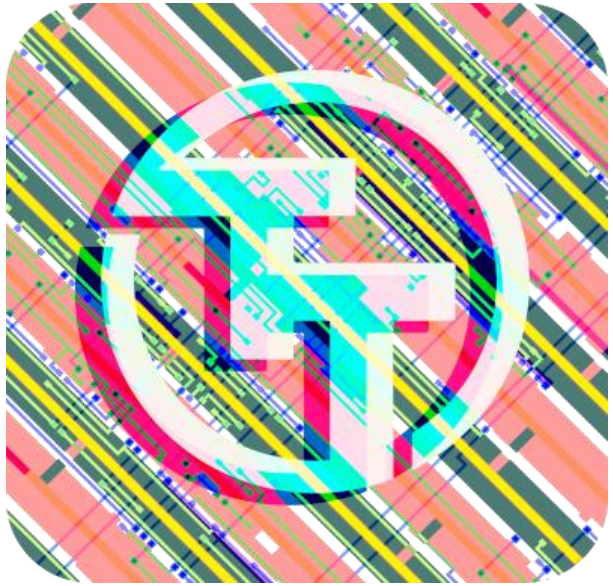
Resource Utilization

Architecture	Combinational ALUTs	Dedicated Logic Registers	Memory bits
Without SPI	458	252	263296
With SPI	574	333	1152

- With the use of SPI we have removed the Instruction and Data memory from the design, hence the huge reduction in the number of memory bits
- In contrast, we require more utilization of Combinational ALUTs and Dedicated Logic Registers but that is comparatively a small increase

SPI SERV with ASIC

- The SPI SERV was implemented as an ASIC design as well
- We have hardened the design with the TinyTapeout platform
- The design was verified using Verilator and cocotb



cocotb

VERILATOR

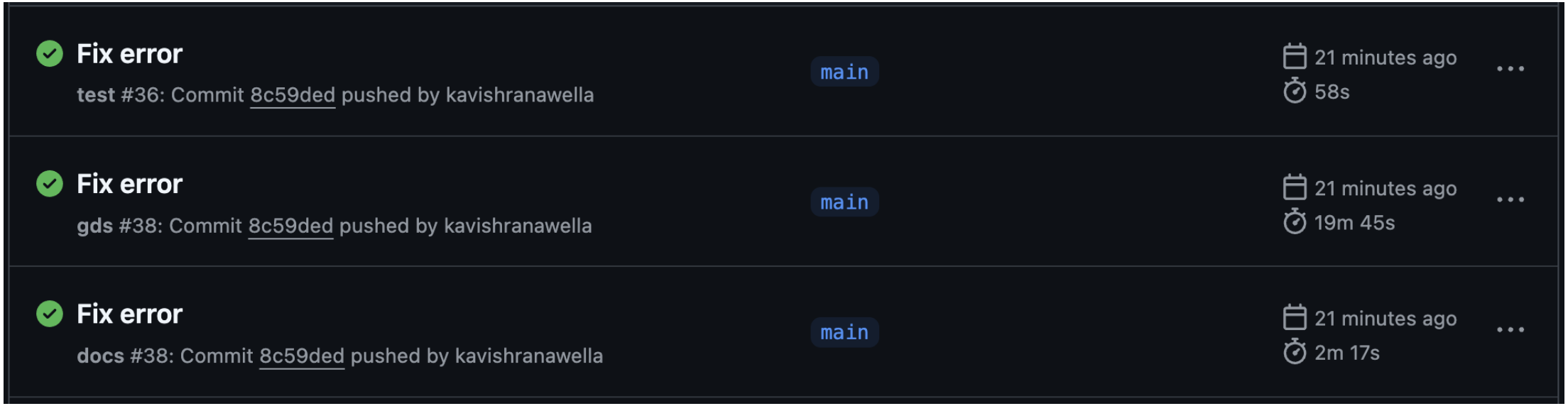
SERV with memory doesn’t fit !!

- Tiles: 8x2
- Placement density: 80%

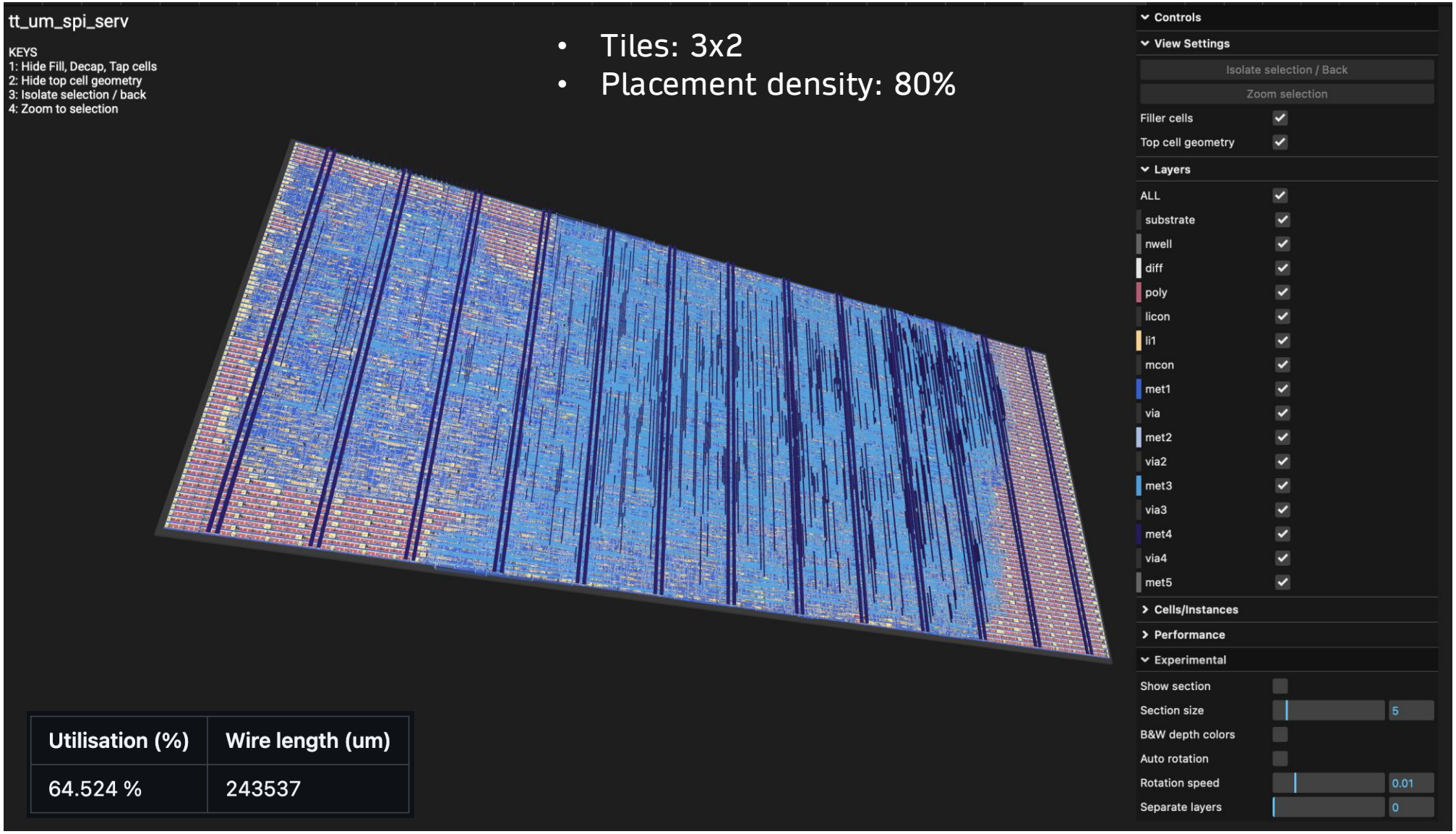
```
106924    2025-05-02T11:17:30.7674682Z [INFO GPL-0002] DBU: 1000
106925    2025-05-02T11:17:30.7682355Z [INFO GPL-0003] SiteSize: (  0.460  2.720 ) um
106926    2025-05-02T11:17:30.7689533Z [INFO GPL-0004] CoreBBox: (  2.760  2.720 ) ( 1375.400 223.040 ) um
106927    2025-05-02T11:17:34.0323680Z [INFO GPL-0006] NumInstances:           745467
106928    2025-05-02T11:17:34.0329655Z [INFO GPL-0007] NumPlaceInstances:      740906
106929    2025-05-02T11:17:34.0335046Z [INFO GPL-0008] NumFixedInstances:        4561
106930    2025-05-02T11:17:34.0341398Z [INFO GPL-0009] NumDummyInstances:           0
106931    2025-05-02T11:17:34.0347827Z [INFO GPL-0010] NumNets:                740925
106932    2025-05-02T11:17:34.0353390Z [INFO GPL-0011] NumPins:               2905063
106933    2025-05-02T11:17:34.0360747Z [INFO GPL-0012] DieBBox:  (  0.000  0.000 ) ( 1378.160 225.760 ) um
106934    2025-05-02T11:17:34.0367380Z [INFO GPL-0013] CoreBBox: (  2.760  2.720 ) ( 1375.400 223.040 ) um
106935    2025-05-02T11:17:34.0373346Z [INFO GPL-0016] CoreArea:            302420.045 um^2
106936    2025-05-02T11:17:34.0379722Z [INFO GPL-0017] NonPlaceInstsArea:    6112.112 um^2
106937    2025-05-02T11:17:34.0385319Z [INFO GPL-0018] PlaceInstsArea:     10651862.230 um^2
106938    2025-05-02T11:17:34.0391115Z [INFO GPL-0019] Util:                  3594.862 %
106939    2025-05-02T11:17:34.0397759Z [INFO GPL-0020] StdInstsArea:       10651862.230 um^2
106940    2025-05-02T11:17:34.0403303Z [INFO GPL-0021] MacroInstsArea:           0.000 um^2
106941    2025-05-02T11:17:34.0415365Z [11:17:34] ERROR    [GPL-0301] Utilization 3594.862 % exceeds    openroad.py:233
106942    2025-05-02T11:17:34.0416015Z                     100%.
106943    2025-05-02T11:17:34.5166291Z Error: gpl.tcl, 75 GPL-0301
```

*tile=167x108 uM

TinyTapeout Hardening

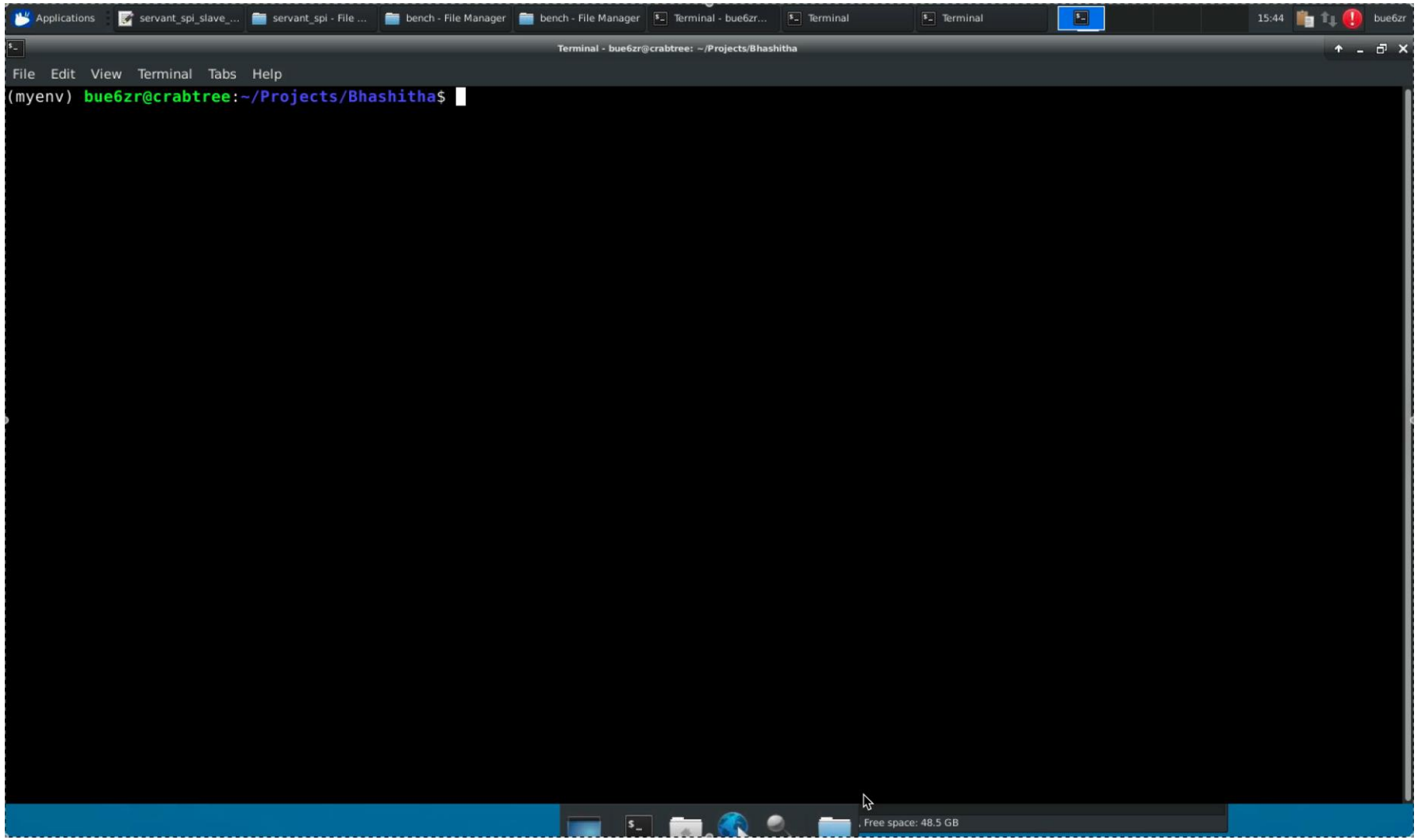


TinyTapeout Hardening

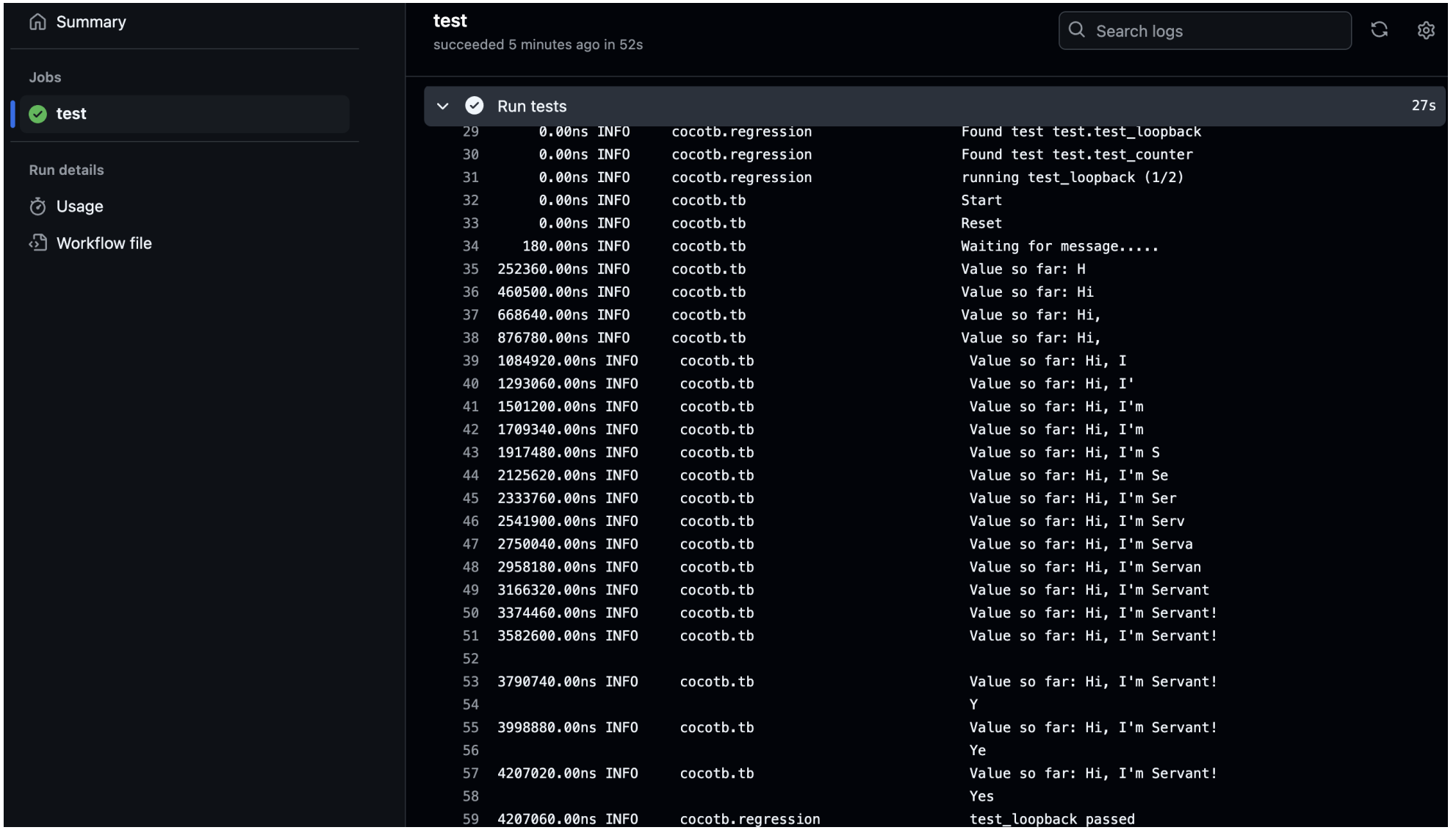


Dining Philosopher Demo using Verilator

- Verilator is a lightweight verification tool that uses C++/SystemC wrapping
- We use that to verify our ASIC design (one of the tests is demonstrated in this video)



Sanity Tests with cocotb



Resource Utilization

Cell usage by Category

Category	Cells	Count
Fill	decap fill	4083
Tap	tapvpwrvgnd	1577
Flip Flops	dfxtp dfstp dfrtp	1485
Multiplexer	mux4 mux2	1472
Misc	conb dlymetal6s2s dlygate4sd3	1375
Combo Logic	nor3b or3b o311ai a22o a2bb2o a211o a221o and2b a21o o22a o21ba a21oi and4b and4bb a32o o22ai o221a and3b o2bb2a o211a a31o a22oi a211oi o211ai o21a o21ai a2111o a31oi o311a o31a a311o o32a a221oi o2111a or4b a21bo	1157
Buffer	clkbuf buf bufinv	683
NOR	nor2 nor3 xnor2 nor4	668
OR	or4 or2 or3 xor2	184
AND	and2 and4 and3 a21boi	116
Inverter	inv	71
NAND	nand2 nand2b nand3	69
Latch	dlxtp dlxtn	63
Diode	diode	53
Clock	clkinv clkinvlp	10

FPGA & ASIC: What did we achieve?

- **Decouples memory from CPU core**: Enables flexible memory placement and simplifies physical design for fabrication - ACHIEVED
- **Minimal pin count**: Communicates with external RAM using only 4 wires (MISO, MOSI, SCK, CS), reducing I/O complexity - ACHIEVED
- **Aligns with SERV's bit-serial philosophy**: Maintains SERV's ultra-minimal, bit-serial architecture by extending serial design principles to memory access - PARTIALLY ACHIEVED
- **Shrinks logic footprint**: Removes internal RAM, reducing FPGA resource usage and improving area efficiency for ASIC targets - PARTIALLY ACHIEVED

Future work

- **Eliminate Wishbone**: Replace the Wishbone bus with a fully bit-serial interconnect to further reduce logic complexity and align with SERV's serial architecture.

- **Add bootloader support**: Enable loading programs such as **Zephyr RTOS** from SPI RAM or other sources at startup.

- **Integrate basic peripherals**:
  - **GPIO**: Provide general-purpose I/O for basic hardware interfacing.
  - **UART (RX)**: Allow serial communication for debugging or basic shell interaction.

- **Implement I2C-based memory access**: Use **I2C** as an alternative to SPI for connecting external RAM — reducing wire count even further in ultra-minimal systems.

Thank You !

Questions ??

Backups

Resource Utilization

with SPI

Memory Bits
0
0
0
1152
1152
1152
1152
0
0
0

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins
2	|servant_spi_top:servant|	574 (0)	333 (0)	1152	0	0	0	0	0
1	|serv_rf_ram:rf_ram|	2 (2)	1 (1)	1152	0	0	0	0	0
1	|altsyncram:memory_rtl_0|	0 (0)	0 (0)	1152	0	0	0	0	0
1	|altsyncram_qvg1:auto_generated|	0 (0)	0 (0)	1152	0	0	0	0	0
2	|servant_gpio:gpio|	1 (1)	2 (2)	0	0	0	0	0	0
3	|servant_mux:servant_mux|	1 (1)	1 (1)	0	0	0	0	0	0
4	|servant_spi_master_if:spi_master_if|	135 (135)	82 (82)	0	0	0	0	0	0
5	|servant_timer:timer|	98 (98)	65 (65)	0	0	0	0	0	0
6	|servile:cpu|	337 (0)	182 (0)	0	0	0	0	0	0
1	|serv_rf_ram_if:rf_ram_if|	40 (40)	18 (18)	0	0	0	0	0	0
2	|serv_top:cpu|	291 (0)	164 (0)	0	0	0	0	0	0
1	|serv_alu:alu|	14 (14)	2 (2)	0	0	0	0	0	0
2	|serv_bufreg2:bufreg2|	83 (83)	32 (32)	0	0	0	0	0	0
3	|serv_bufreg:bufreg|	9 (9)	33 (33)	0	0	0	0	0	0
4	|serv_csr:gen_csr.csr|	29 (29)	10 (10)	0	0	0	0	0	0
5	|serv_ctrl:ctrl|	45 (45)	34 (34)	0	0	0	0	0	0
6	|serv_decode:decode|	16 (16)	13 (13)	0	0	0	0	0	0
7	|serv_immdec:immdec|	38 (38)	27 (27)	0	0	0	0	0	0
8	|serv_mem_if:mem_if|	5 (5)	1 (1)	0	0	0	0	0	0
9	|serv_rf_if:rf_if|	16 (16)	0 (0)	0	0	0	0	0	0
10	|serv_state:state|	36 (36)	12 (12)	0	0	0	0	0	0
3	|servile_arbiter:arbiter|	3 (3)	0 (0)	0	0	0	0	0	0
4	|servile_mux:mux|	3 (3)	0 (0)	0	0	0	0	0	0
3	|servive_clock_gen:clock_gen|	10 (9)	11 (10)	0	0	0	0	0	0
1	|altpll:pll|	1 (0)	1 (0)	0	0	0	0	0	0

without SPI

Memory Bits
0
263296
1152
1152
1152
0
0
262144
0

Compilation Report - D:/Github/2025-fpga-design-projects-fpga_SPI_I2C/without_spi/servant_1_3_0 - servant_1_3_0

File Edit Tools Window Help

Search altera.com

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthes
  - Summary
  - Settings
  - Parallel Compila
  - Source Files Rea
  - Resource Usage
  - Resource Utilizat
  - RAM Summary
  - IP Cores Summa
  - Optimization Re
  - Source Assignm
  - Parameter Settir
  - LPM Parameter
  - Connectivity Che
  - Post-Synthesis
  - Elapsed Time Pe
  - Messages
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins
2	|uart_rx:uart_rx_inst|	65 (65)	55 (55)	0	0	0	0	0	0
2	|servant:servant|	458 (0)	252 (0)	263296	0	0	0	0	0
1	|serv_rf_ram:rf_ram|	2 (2)	1 (1)	1152	0	0	0	0	0
1	|altsyncram:memory_rtl_0|	0 (0)	0 (0)	1152	0	0	0	0	0
1	|altsyncram_qvg1:auto_generated|	0 (0)	0 (0)	1152	0	0	0	0	0
2	|servant_gpio:gpio|	1 (1)	2 (2)	0	0	0	0	0	0
3	|servant_mux:servant_mux|	1 (1)	1 (1)	0	0	0	0	0	0
4	|servant_ram:ram|	7 (7)	1 (1)	262144	0	0	0	0	0
5	|servant_timer:timer|	98 (98)	65 (65)	0	0	0	0	0	0
6	|servile:cpu|	349 (0)	182 (0)	0	0	0	0	0	0
1	|serv_rf_ram_if:rf_ram_if|	39 (39)	18 (18)	0	0	0	0	0	0
2	|serv_top:cpu|	293 (0)	164 (0)	0	0	0	0	0	0
1	|serv_alu:alu|	14 (14)	2 (2)	0	0	0	0	0	0
2	|serv_bufreg2:bufreg2|	83 (83)	32 (32)	0	0	0	0	0	0
3	|serv_bufreg:bufreg|	9 (9)	33 (33)	0	0	0	0	0	0
4	|serv_csr:gen_csr.csr|	29 (29)	10 (10)	0	0	0	0	0	0
5	|serv_ctrl:ctrl|	45 (45)	34 (34)	0	0	0	0	0	0
6	|serv_decode:decode|	17 (17)	13 (13)	0	0	0	0	0	0
7	|serv_immdec:immdec|	39 (39)	27 (27)	0	0	0	0	0	0
8	|serv_mem_if:mem_if|	5 (5)	1 (1)	0	0	0	0	0	0
9	|serv_rf_if:rf_if|	16 (16)	0 (0)	0	0	0	0	0	0
10	|serv_state:state|	36 (36)	12 (12)	0	0	0	0	0	0
3	|servile_arbiter:arbiter|	14 (14)	0 (0)	0	0	0	0	0	0
4	|servile_mux:mux|	3 (3)	0 (0)	0	0	0	0	0	0
3	|servive_clock_gen:clock_gen|	10 (9)	11 (10)	0	0	0	0	0	0

Note: For table entries with two numbers listed, the numbers in parentheses indicate the number of resources of the given type used by the specific entity alone. The numbers listed outside of parentheses indicate the total resources of the given type used by the specific entity and all of its sub-entities in the hierarchy.

100% 00:00:55

Title