

TinyTapeout

Alice Smith*, Bob Jones†, Charlie Brown*, David Wilson‡

*Department of Computer Science, University X

Email: {alice.smith, charlie.brown}@universityx.edu †Department of Electrical Engineering, University Y

Email: b.jones@universityy.com ‡Department of Mathematics, University Z

Email: d.wilson@universityz.org

Abstract—The abstract goes here

Index Terms—Open Source Silicon, TinyTapeout

I. INTRODUCTION

TINITAPEOUT TinyTapeout [1] is an educational project that makes it easier and cheaper than ever to get ASIC designs manufactured. The digital design flow consists of templating a GitHub [2] repository, adding a design, waiting for the tests and binary layout files (GDS [3]) generation to complete, then submitting to a quarterly shuttle.

Up to 500 designs are multiplexed to 24 general purpose input/output (GPIO) pins, and after manufacture the chips are mounted to a demonstration board for easy testing. Each design can be activated and tested in turn. Documentation submitted with each project forms a printable datasheet [4] as well as an online index at TinyTapeout.com/runs/ [5]

Design entry is done mostly with Verilog or Wokwi [6]. Wokwi is a schematic based editor that is an easy way to get started for people with no prior hardware description language (HDL) experience. The TinyTapeout website includes a basic getting started guide for drawing circuits with Wokwi available in English and Spanish.

The first [7] free and experimental shuttle with 152 designs was submitted to the seventh Google sponsored [8] lottery multi project wafer (MPW) shuttle in September 2022. The next 4 shuttles combined 582 designs and were sponsored by and manufactured with the Efabless [9] chipIgnite MPW service.

Each tile is approximately $100 \times 160\mu m$, enough for around 1000 logic gates and is priced at \$50. The physical chip and demo board are optional and cost an additional \$250. Individuals pay a reduced \$100 for their first chip and board thanks to sponsorship by Efabless [9].

By separating the cost of area and the cost of the chip, a group of 10 could submit 10 designs and share 1 board for \$600.

The GitHub templates [10] make use of GitHub Actions [11] - an automatic continuous integration system that is triggered every time the repository is updated. There are 4 main jobs:

- 1) GDS - installs OpenLane and the Sky130 process design kit (PDK), then builds the GDS, and generates a summary of the design that includes utilization, standard cells used, and a 2D and 3D model of the GDS. This job can optionally also run a gate-level verification.

- 2) Verification - installs the YosysHQ open source CAD suite which includes many common electronic design automation (EDA) tools. Then iVerilog and cocotb are used to run any testbenches included.
- 3) Documentation - generates a preview of the documentation.
- 4) Precheck - a number of tests are run to make sure that the design doesn't cause design rule check (DRC) errors after integration into the chip.

Successful GDS, Documentation, and Precheck jobs are required to submit to a shuttle. Verification is optional but highly encouraged. Wokwi designs can make use of an integrated truth table testing system [12].

While the process can be done entirely in the browser, it's also possible to install a local copy of the tools, which can help to reduce iteration time, especially for tests and verification.

Routing stats

Utilisation (%)	Wire length (um)
51.7	16572

Cell usage by Category

Category	Cells	Count
Fill	decap fill	1145
Combo Logic	o21ai nand3b a211o o31a o2b a212o nor2b o21ei a11o a21oi a211o o21ba o21a o21bai a21b2b o2b o221a a31o a22o o32a o22a a32o	249
Tap	tappyvervnd	246
Flip Flops	dfltp	146
Buffer	buf clkbuf	127
AND	and2 a21b1oi and3 and4	97
Misc	dlygate4sd3 dlymetal6sd3 comb	84
OR	or3 xor2 or2 or4	81
NOR	xnor2 nor2 nor3 nor4	64
NAND	nand2 nand3 nand4 nand2b	52
Inverter	inv	37
Multiplexer	mxu2 mxu4	9
Diode	diode	1

947 total cells (excluding fill and tap cells)

Fig. 1. The summary table of the GDS job.

Community engagement has been strong with 756 designs submitted over the 5 shuttles. The Discord community has 1000 members with 1600 subscribed to the mailing list.

II. SCANCHAIN ARCHITECTURE

TinyTapeout started as an experiment in fitting as many designs as possible into the $10mm^2$ available on the Google lottery shuttles. As a fast proof of concept, a scan chain was chosen. Each design had 8 inputs and 8 outputs. Clock and reset were optional and not treated specially. The chain was formed of scan flops [13], a type of flip flop with an integrated multiplexer at its input.

TABLE I
TABLE SHOWS THE STATS FOR EACH OF THE TINYTAPEOUT SHUTTLE RUNS.

Run	Launched	Closed	Shuttle	Designs	Chips Expected	Estimated delivery date
TT01	2022-08-17	2022-09-01	MPW7	152	2024-01-30	Not expecting to ship this test
TT02	2022-11-09	2022-12-02	2211Q	165	2023-10-17	2024-01-30
TT03	2023-03-01	2023-04-23	2304C	249*	2024-01-15	2024-02-28
TT04	2023-07-01	2023-09-08	2309	143	2024-02-28	2024-04-15
TT05	2023-09-11	2023-11-04	2311	174	2024-04-12	2024-05-12
TT06	2024-02-01	2024-04-19	2404	TBD	TBD	TBD

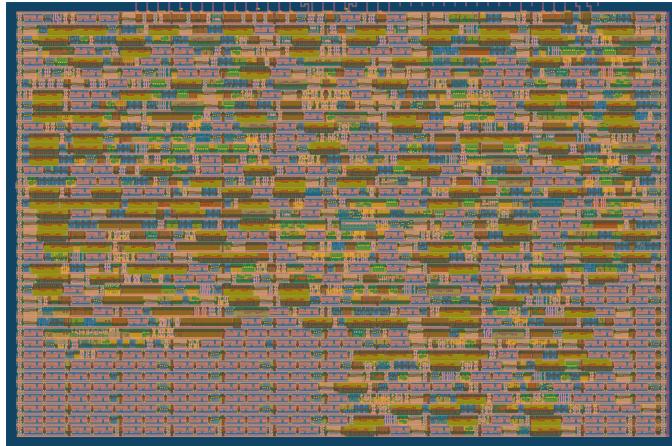


Fig. 2. The 2D render of the cells in use, with empty areas visible in the lower left corner.

About our community

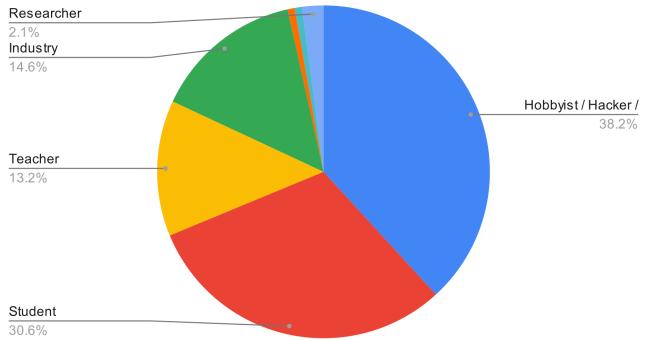


Fig. 5. How TT02 submitters identified themselves.



Fig. 3. The interactive 3D viewer.

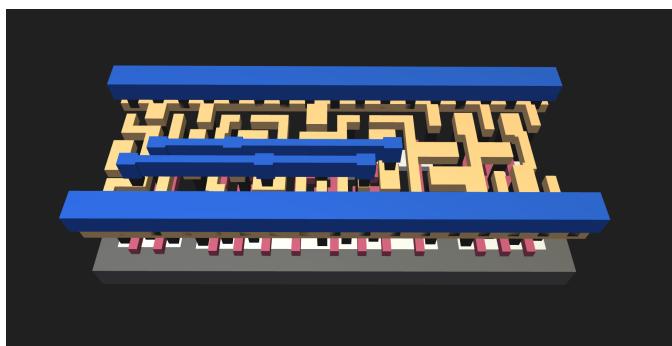


Fig. 4. A deep zoom to a D-type flip-flop, isolated from the rest of the design.

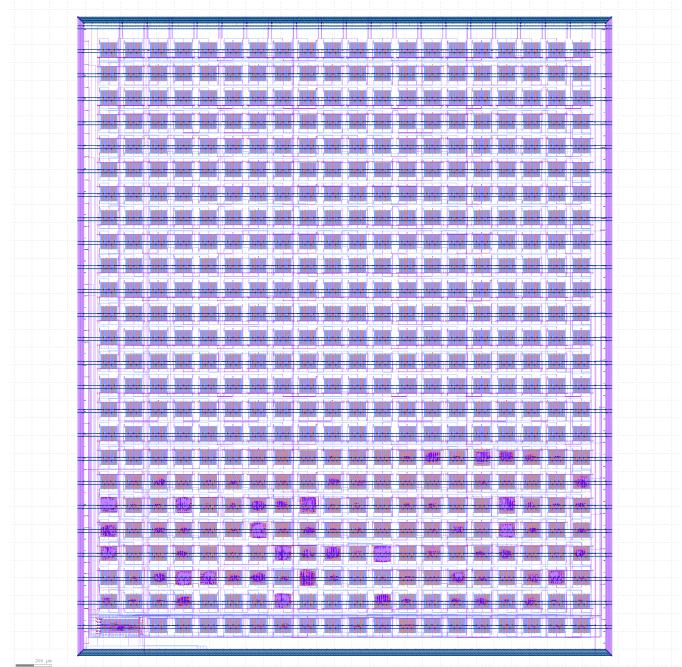


Fig. 6. 500 designs connected in a chain for TT01, with the scan chain driver in the lower left corner.

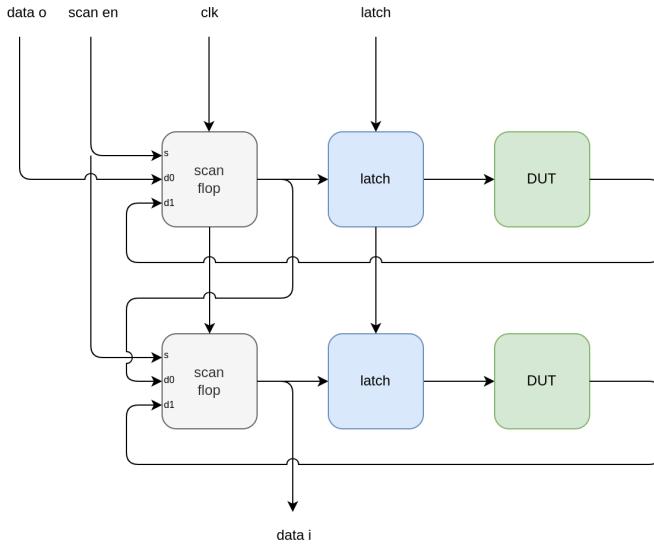


Fig. 7. A simplified view of 2 designs in the chain.

Each design sends data into the scan flops secondary input and receives input from the output of the flop via a latch. The chain is built [14] by sending data from the output of the previous scan flop into the next scan flops's primary input.

This arrangement allows the loading of data into any of the designs, and then capturing the output and clocking that through the rest of the chain to the output.

While relatively easy to implement, the downside is the latency. The more designs in the chain, the longer it takes to send and receive data.

Assuming a $50MHz$ scan chain clock, 250 designs with 8 inputs and 8 outputs, the maximum refresh rate is $50M/(8 \times 250) = 25kHz$.

Another concern is hold violations due to the large number of serially connected flops and potentially large clock skews due to long signal wires. This was mitigated by:

- Reclocking the output data with a negedge flop, providing more hold margin.
- Driving the latch and scan signals separately, and with a configurable delay.
- Keeping wires between designs short by snaking them through the grid.

To further reduce the risk of failure, three independent methods of driving the chain are provided:

- 1) External - all the signals are available as GPIO pins on the chip. This safest option is the default.
- 2) Internal - an internal scan chain driver runs the chain and copies the data to and from 16 GPIOs.
- 3) Logic analyser - The Caravel [15] harness provided by Efabless contains a RISC-V co-processor with a firmware driven logic analyser. The logic analyser was connected as a third possible way to drive the chain and hence access all designs from firmware.

After static timing analysis (STA) it was discovered that the clock duty cycle would increase slightly at each of the 500 sequential clock drivers. This is because PMOS transistors are weaker than NMOS transistors, so for a balanced drive

the PMOS needs to be substantially larger than the NMOS. To save space, the PMOS is typically undersized in standard cells, and so the rising clock edge is always slightly slower. This effect limits the maximum clock speed as eventually the clock duty cycle tends to 100%, but faster speeds are possible by starting with a lower duty cycle. With a 50% duty cycle a $30MHz$ maximum clock is expected, limiting the refresh rate to $7kHz$.

TT01's scan chain was embedded into each design, which created the risk that a user could unintentionally break it, and break the rest of the chain. This risk was mitigated by formally [16] proving the chain was present in the submitted design. For TT02 and TT03, the scan chain was separated into a separate macro block that the user can't modify.

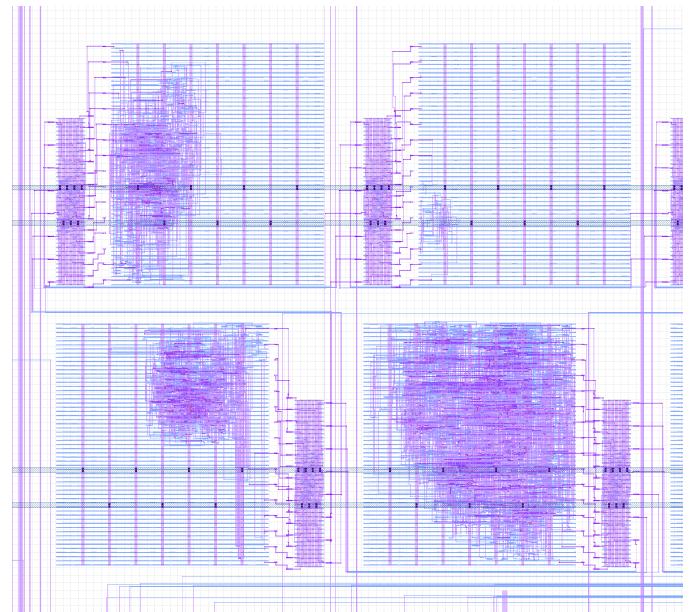


Fig. 8. TT02 designs with separate scan chain blocks.

For TT01 and TT02 each design used two clock buffers, with the internal flops driven after the first buffer.

TT03 used inverting clock buffers, with only one between the clock in and out.

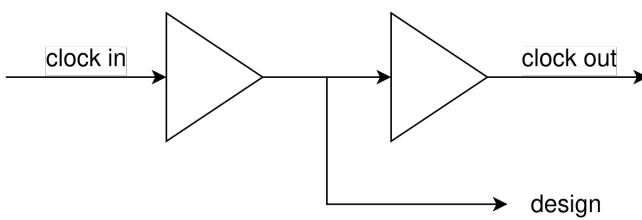
This meant that the clock is inverted between each design, so the delay caused by the PMOS transistors were evenly spread across the negative and positive cycles. With this change, the chain should be able to be driven up to around $60MHz$, for an update frequency of $30kHz$.

The verification effort [17] was broad and included a community review, register transfer level (RTL) and gate level (GL) simulation, Formal Verification [18], static timing analysis, layout vs schematic (LVS), DRC, and device level static verification [19].

III. CIRCUIT BOARDS

After manufacture, the chips are mounted onto small carrier boards with 0.1 inch headers. This allows people with limited equipment or surface mount technology (SMT) assembly experience to build their own demonstration boards.

TT02



TT03

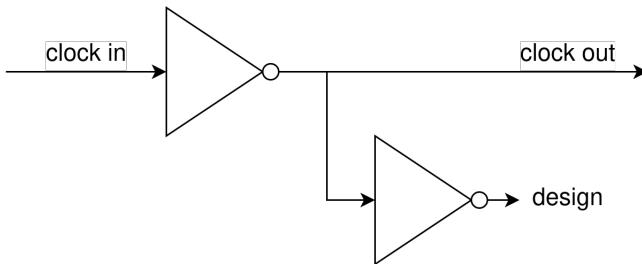


Fig. 9. Differences between TT02 and TT03 scanchain clock buffers.

The carrier fits onto the demonstration board which provides:

- USB-C for power connection,
- 1.8v and 3.3v power supplies for core and IO,
- 20MHz oscillator,
- buttons for reset and single-step clock,
- an 8-way DIP switch for inputs,
- a 9-way DIP switch for design selection,
- a 7-segment LED display for the outputs,
- headers for all IO, including 2 standard Digilent ports (PMOD),
- a header to select internal or external clock,
- a header to select internal or external scan chain driver,
- a header to engage an automatic clock divider in input pin 0.

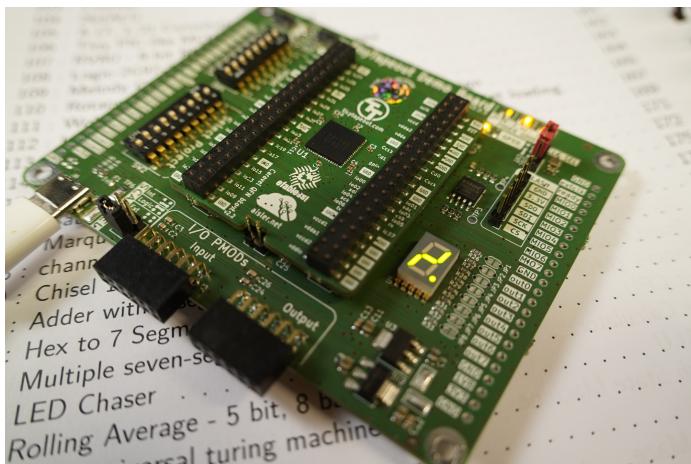


Fig. 10. The demonstration board. Certified Open Source Hardware ES000040 [20].

IV. SCANCHAIN SILICON RESULTS

TT02 chips were received in October 2023, 11 months after the chips were submitted for manufacture on Efabless chipIgnite 2211Q. The chips were tested for the first time in public on a livestream [6]. The chain was validated, and a few of the designs were shown to be working.

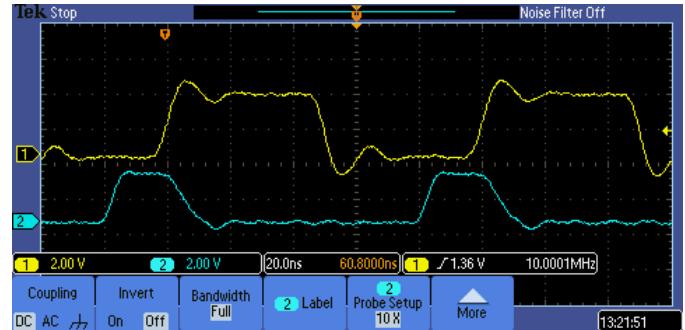


Fig. 11. Measurement from TT02 silicon, with input clock in yellow and the distorted output clock in blue.

In the following days another 30 designs were tested and shown to be working.

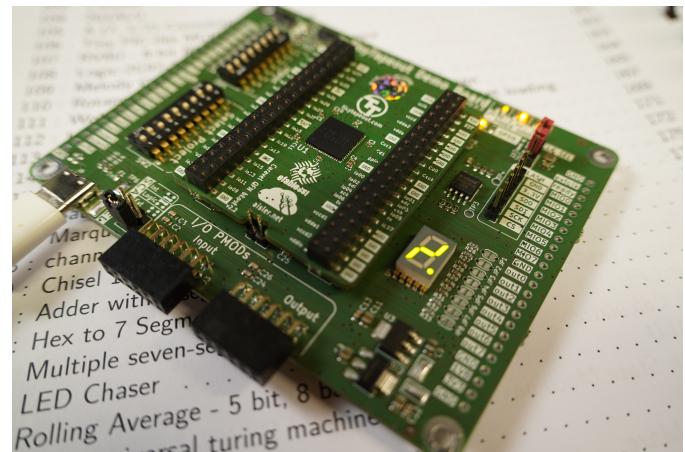


Fig. 12. The TT02 demo board running a design.

After measuring the clock skew and maximum frequency it was decided to run the production boards with a 20MHz oscillator, resulting in a 10MHz scan chain.

Some designs didn't function as expected, which in most cases was due to faults in the submitted design.

As well as 82 Verilog designs, 64 used the Wokwi graphical editor, 5 used Amaranth, and 1 used Chisel. Some Wokwi [21] designs using combinational logic in flop clocks failed in hardware but worked in simulation. This is due to the lack of timing data in the simulation, and wasn't detected by STA because the clocks were not known. A detailed analysis has yet to be carried out. The addition of SR flops to Wokwi will help to alleviate this, as well as the start of an ERC check.

At the time of writing, PCBs are in production and are expected to ship to customers by the end of January 2024.

TinyTapeout 3 silicon was received in January 2024, and the updated scanchain shows a more symmetric output clock at the end of the chain.

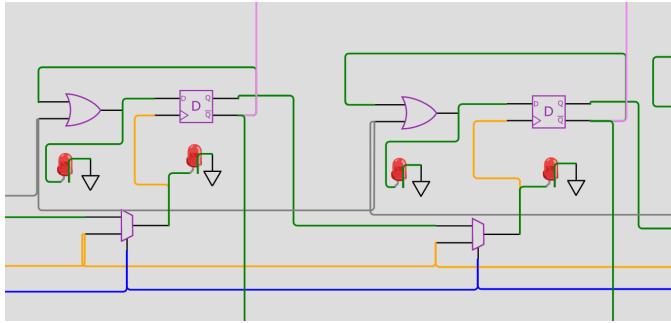


Fig. 13. Combination logic in the clock path of one of the failed designs.

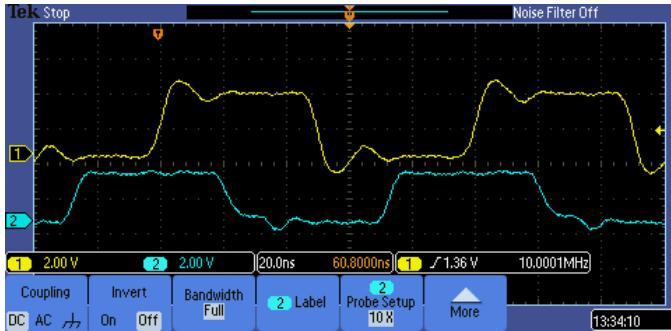


Fig. 14. Measurement from TT03 silicon.

V. BEYOND THE SCANCHAIN

The biggest limitation of the Tiny Tapeout architecture was the IO latency. For Tiny Tapeout 4 a new architecture was needed, and a series of proposals was gathered from the community. An online video call was held and the 10 proposals discussed. The winning design was a fairly straightforward multiplexer design.



Fig. 15. The multiplexer design for Tiny Tapeout 4.

The physical layout consists of a central controller connected up and down to two vertical spines. Twenty-four horizontal muxes connect to the spine with each supporting 16 designs. This allows up to 384 separate single tile designs. Multiple tile designs were also enabled, allowing a maximum

project size of 2×8 tiles or $1359 \times 225\mu m$ - around 20,000 logic cells.

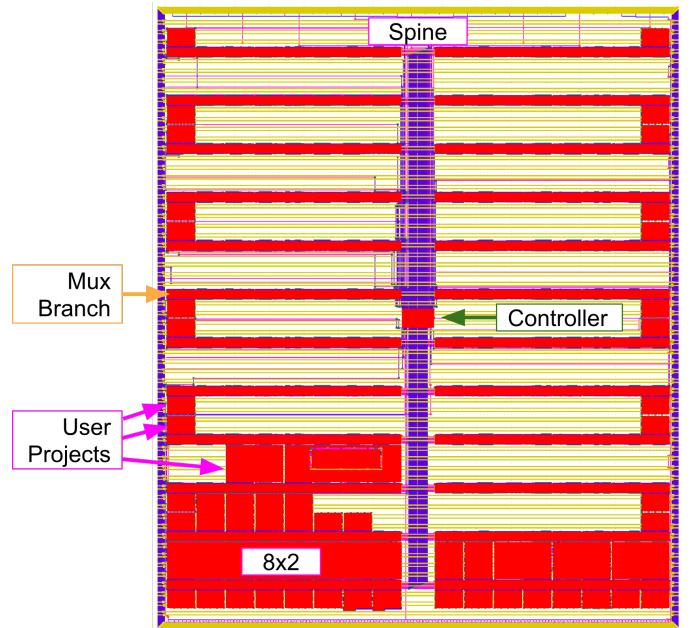


Fig. 16. The TT03.5 test design.

Another major limitation of TT1 to 3 was the small number of IO. The scan controller used 9 GPIOs to select the currently active design, which, while simplifying the demo board, wasted valuable pins. With TT04, the parallel design selection was dropped in favor of a serial protocol. The extra pins were then used as bidirectional pins, giving each design clock, reset, and 24 IO.

TABLE II
COMPARISON BETWEEN TT03 AND TT04

Parameters	Tiny Tapeout 3	Tiny Tapeout 4
Max clock speed	12.5kHz	50MHz
Max design size	$150 \times 170\mu m$	$1359 \times 225\mu m$
Input pins	8	10
Output pins	8	8
Bidirectional I/O pins	None	8
Custom GDS file	✗	✓

An invite-only experimental shuttle [22] was submitted with 32 designs to Efabless chipIgnite 2306C. Two of the designs included a power gate as a stepping stone to supporting analog and mixed-signal designs.

VI. MULTIPLEXER SILICON RESULTS

After silicon was received, the worst round trip latency was measured to be 20ns .

The new chip pinout and serial design selection required a new demo board that included an easy way to select the design. The RP2040 microcontroller was chosen as a co-processor as it allows:

- Drag and drop firmware updates on any OS,
- Runs MicroPython [22], ideal for beginners to test their designs,

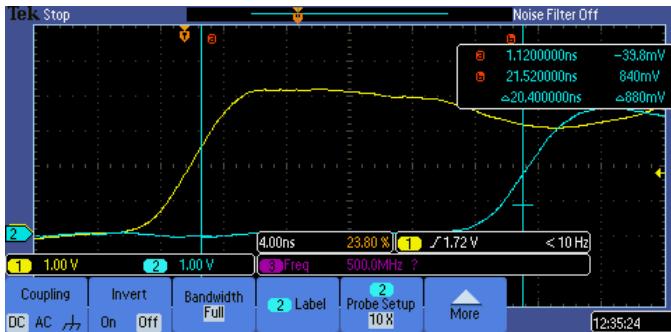


Fig. 17. Round trip latency on a rising edge of about 20ns.

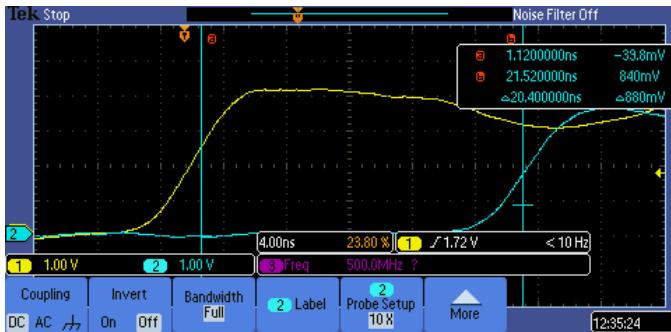


Fig. 18. Round trip latency on a falling edge of about 16ns.

- External memory emulation via PIO and DMA.

```
enabling design tt_um_test by sending 102 [0b01100110] pulses
design repo https://github.com/TinyTapeout/tt03p5-test @ 434c5d508d20053bea346881a61355f87ea1ca91
0 0 0 0
1 0 0 0
0 1 0 0
1 1 0 0
0 0 1 0
1 0 1 0
0 1 1 0
1 1 1 0
0 0 0 1
```

Fig. 19. A MicroPython program enabling a design, clocking it, and printing the results.

```
enabling design tt_um_test by sending 102 [0b01100110] pulses
design repo https://github.com/TinyTapeout/tt03p5-test @ 434c5d508d20053bea346881a61355f87ea1ca91
0 0 0 0
1 0 0 0
0 1 0 0
1 1 0 0
0 0 1 0
1 0 1 0
0 1 1 0
1 1 1 0
0 0 0 1
```

Fig. 20. The TT04+ demo board.

An additional PMOD expansion port was added for the bidirectional pins, and the community has started to standardize on pinouts [23] making it easier to test each other's designs. A new repository was created to house user-contributed PMODs [24].

VII. IMPROVING THE MULTIPLEXER AND MIXED SIGNAL SUPPORT

TT05 splits the mux into two parts to improve performance. As each spine segment is now half as long, it will have half the capacitance. We expect to reduce the round trip latency to around 10ns.

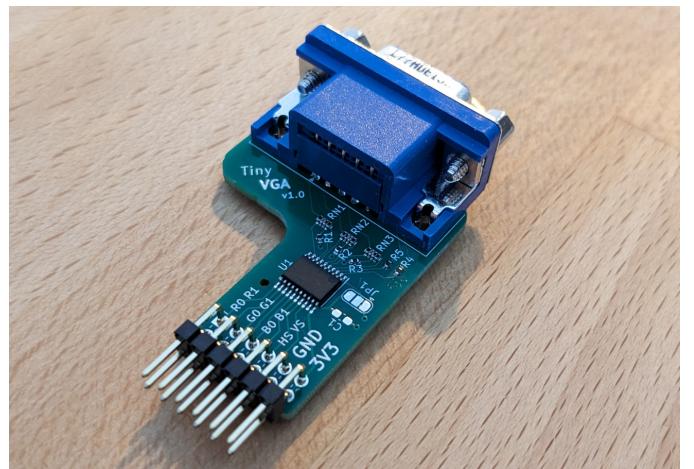


Fig. 21. A user-contributed VGA output PMOD.



Fig. 22. VGA clock design running on TT03.5 silicon.

For TT06, the Caravel harness will be replaced by OpenFrame [25], an alternative harness provided by Efabless that uses the same padring but removes the RISC-V coprocessor. This adds an extra 5mm^2 more space for user designs, and an extra 12 pins that will be used for analog.

For increased safety, all designs will be power-gated, which will allow designers to take more risks or use custom flows.

Analog and mixed-signal designs will be enabled by adding an analog multiplexer based on transmission gates [26]. This allows up to 192 designs to share the 8 analog pins between them.

TT06 is planned to open for digital designs at the end of January 2024, for analog designs at the end of February, and to close on April 19th, 2024.

VIII. SILICON SHOWCASE

A small sample of the types of designs possible with TinyTapeout are listed below:

- Serial FPGA (Link)
- Synthesizable Digital Temperature Sensor (Link)
- 395 standard cells with mux (Link)

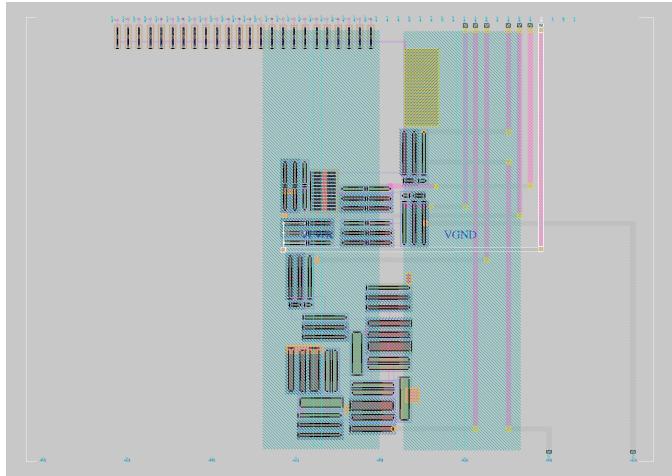


Fig. 23. Transmission gate tested in TT05 that will be used to form the analog multiplexer.

- FM transmitter with I2S input (Link)
- USB full speed - (Link)
- A Linux capable RISCV CPU - (Link)

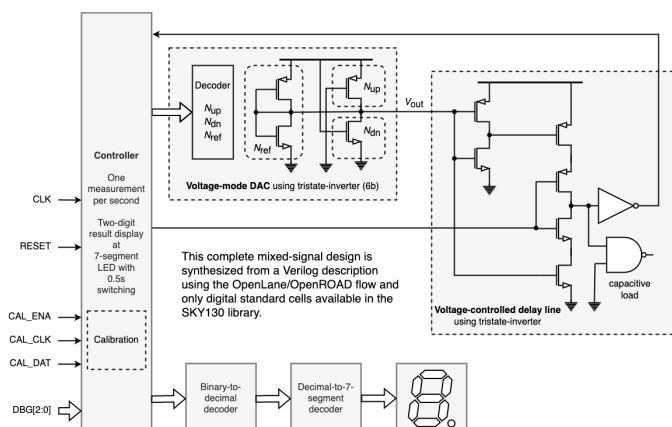


Fig. 24. The Synthesizable Digital Temperature Sensor.

IX. CONCLUSIONS

The conclusion goes here.

ACKNOWLEDGMENTS

We thank ... or not

REFERENCES

- [1] “TinyTapeout,” <https://tinytapeout.com/>, accessed: [Insert access date here].
- [2] “GitHub,” <https://github.com/>, accessed: [Insert access date here].
- [3] “GDS2 File Format - Zero to ASIC Course,” <https://www.zerotoasiccourse.com/terminology/gds2/>, accessed: [Insert access date here].
- [4] “Datasheet PDF - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/raw/tt02/datasheet.pdf>, accessed: [Insert access date here].
- [5] “TinyTapeout Runs,” <https://tinytapeout.com/runs/>, accessed: [Insert access date here].
- [6] “Wokwi,” <https://wokwi.com/>, accessed: [Insert access date here].
- [7] “First Shuttle - TinyTapeout Runs,” <https://tinytapeout.com/runs/tt01/>, accessed: [Insert access date here].
- [8] “Announcing the GlobalFoundries Open MPW Shuttle Program - Google Open Source Blog,” <https://opensource.googleblog.com/2022/10/announcing-globalfoundries-open-mpw-shuttle-program.html>, accessed: [Insert access date here].
- [9] “Efabless,” <https://efabless.com/>, accessed: [Insert access date here].
- [10] “Verilog Template - TinyTapeout Repository,” <https://github.com/TinyTapeout/tt06-verilog-template>, accessed: [Insert access date here].
- [11] “GitHub Actions Documentation,” <https://docs.github.com/en/actions>, accessed: [Insert access date here].
- [12] “Wokwi Automated Testing - TinyTapeout,” https://tinytapeout.com/digital_design/wokwi_automated_testing/, accessed: [Insert access date here].
- [13] “Skywater PDK Documentation,” https://skywater-pdk.readthedocs.io/en/main/contents/libraries/sky130_fd_sc_hdll/cells/sdfxtpl/README.html, accessed: [Insert access date here].
- [14] “Updating Inputs and Outputs of a Design - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/blob/tt02/INFO.md#updating-inputs-and-outputs-of-a-specified-design>, accessed: [Insert access date here].
- [15] “Caravel Harness - Efabless Repository,” <https://github.com/efabless/caravel>, accessed: [Insert access date here].
- [16] “TinyTapeout Scan - GitHub Repository,” https://github.com/jix/tinytapeout_scan, accessed: [Insert access date here].
- [17] “Verification Documentation - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/blob/tt02/VERIFICATION.md>, accessed: [Insert access date here].
- [18] “SymbiYosys (SBY) - YosysHQ Repository,” <https://github.com/YosysHQ/sby>, accessed: [Insert access date here].
- [19] “CVC - Verification Documentation - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/blob/tt02/VERIFICATION.md#cvc>, accessed: [Insert access date here].
- [20] “Open Source Hardware Certification - TinyTapeout,” <https://certification.oshwa.org/es000040.html>, accessed: [Insert access date here].
- [21] “Wokwi,” <https://wokwi.com/>, accessed: [Insert access date here].
- [22] “MicroPython Official Website,” <https://micropython.org/>, accessed: [Insert access date here].
- [23] “Pinouts Specifications - TinyTapeout,” <https://tinytapeout.com/specs/pinouts/>, accessed: [Insert access date here].
- [24] “Awesome TinyTapeout PMODs - GitHub Repository,” <https://github.com/TinyTapeout/awesome-tinytapeout-pmods>, accessed: [Insert access date here].
- [25] “Caravel Openframe Project,” https://github.com/efabless/caravel_openframe_project, accessed: [Insert access date here].
- [26] “TT05 Analog Test,” <https://github.com/iic-jku/tt05-analog-test>, accessed: [Insert access date here].