

# TinyTapeout: A Shared Silicon Tapeout Platform Accessible To Everyone

Matt Venn  
Cofounder, TinyTapeout  
Email: matt@tinytapeout.com

**Index Terms**—ASIC, Multi Project Chip, Open Source Silicon, TinyTapeout.

## I. INTRODUCTION

**TINYTAPEOUT** is a multi project chip platform that makes it easier and cheaper than ever to get ASIC designs manufactured.

Open source tools and process design kit (PDK [1]) are used, so no licenses or NDAs are needed. As the tools are run in the cloud, no software needs to be installed on the user's machine. However, as long as the template structure is followed, proprietary tools can be used.

Around 400 open source 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 chip contains every design, which can be activated and tested in turn.

Additionally, each project submits documentation for their design, collected to form a printable datasheet [2] along with an online index at [TinyTapeout.com/runs/](http://TinyTapeout.com/runs/) [3]. The datasheet helps participants to explore each other's design in addition to their own.

By separating the cost of area and the physical chip, a group can share the cost of chip packaging and PCBs, while still getting to test and measure all the designs on the chip. In a classroom setting this helps to reduce the overall price, as students can share a smaller number of PCBs while each submitting their own design.

Each tile (Fig. 1) is approximately  $160 \times 100 \mu\text{m}^2$ , enough for around 1000 logic gates. Tiles can be joined to enable larger designs. Analog and mixed signal support is being added for the next shuttle.

Community engagement has been strong with 756 designs submitted over the first 5 shuttles. Some highlights are listed in section IX. The online chat server has 1000 members with 1600 subscribed to the mailing list. Submitters tend to identify as hobbyists, students and teachers as shown in Fig. 2.

The first [4] free and experimental shuttle with 152 designs was submitted to the seventh Google sponsored [5] 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 [6] chipIgnite MPW service. Table I shows a summary of all the shuttle runs to date.

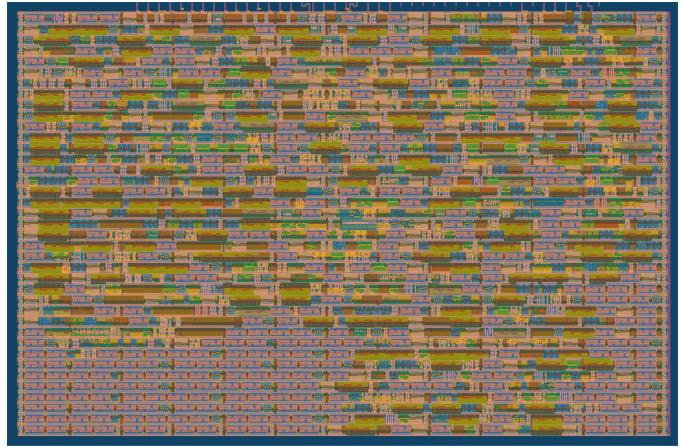


Fig. 1. 2D render of a single tile

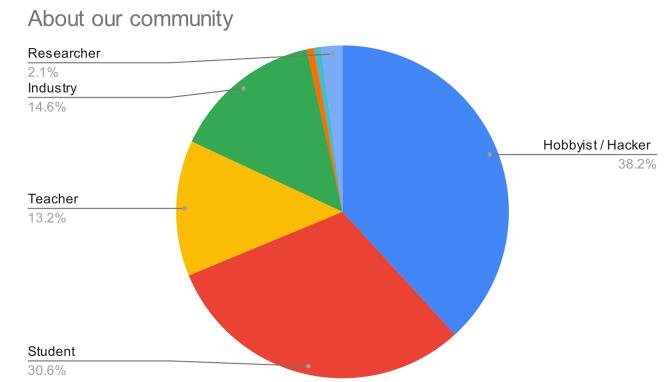


Fig. 2. How TT04 submitters identified themselves.

The rest of this paper will discuss the TinyTapeout design flow, multiplexer evolution, circuit boards, silicon results and next steps.

## II. DESIGN FLOW

Design entry is done mostly with Verilog or Wokwi [7]. Wokwi is a web based schematic based editor that is an easy way to get started for people with no prior hardware description language (HDL) experience. The TinyTapeout website [8] includes a basic getting started guide for drawing circuits available in English and Spanish.

The design flow consists of templating a GitHub [9] repository, adding a design, waiting for the tests and binary layout

TABLE I  
STATISTICS 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

files (GDS [10]) generation to complete, then submitting to a quarterly shuttle.

The GitHub templates [11] make use of GitHub Actions [12]—an automatic continuous integration system that is triggered every time the repository is updated.

There are 4 main jobs:

- 1) GDS: installs OpenLane [13] and the Sky130 [14] PDK, builds the GDS and generates a summary of the design (Fig. 3). The summary includes utilization, standard cells used, a 2D render (Fig. 1) and an interactive 3D viewer (Fig. 4). This job can also optionally 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 [15] and cocotb [16] 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 [17].

While the process can be done entirely in the browser, it's also possible to install a local copy of the tools [18], which can help to reduce iteration time, especially for tests and verification.

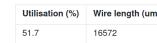
### III. SCAN CHAIN ARCHITECTURE

TinyTapeout started as an experiment in fitting as many designs as possible into the 10 mm<sup>2</sup> available on the Google lottery shuttles (Fig. 5). 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 [14], a type of flip flop with an integrated multiplexer at its input. An example showing a two-design scan chain is shown in Fig. 6.

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 [19] by sending data from the output of the previous scan flop into the next scan flop'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.

### Routing stats



### Cell usage by Category

Category	Cells	Count
Fill	decap fill	1145
Combo Logic	o21ai nand3b a2210 o31a or2b a21b0 nor2b o31ai a41a o21ci a2110 o21ba o21bai a22b20 or3b o221a a31o a22e o32a o23a a32e	249
Tap	tappywvngd	246
Flip Flops	dfltp	146
Buffer	buf clkbuf	127
AND	and2 o21boi and3 and4	97
Misc	dlygate4x4d3 dlymetal1x2d3 comb	84
OR	or3 xor2 or2 or4	81
NOR	xnor2 nor2 nor3 nor4	64
NAND	nand2 nand3 nand4 nand2b	52
Inverter	inv	37
Multiplexer	mx2x mx4	9
Diode	diode	1

947 total cells (excluding fill and tap cells)

Fig. 3. The summary table of the GDS job.



Fig. 4. The interactive 3D viewer.

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. For example, assuming a 50 MHz scan chain clock, 250 designs with 8 inputs and 8 outputs, the maximum refresh rate is  $50\text{ MHz}/(8 \times 250) = 25\text{ kHz}$ .

TT01's scan chain was embedded into each design, which meant that a user could unintentionally remove it, breaking the chain. This risk was mitigated with a formal [20] equivalence check—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.

Another concern was 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 substantially

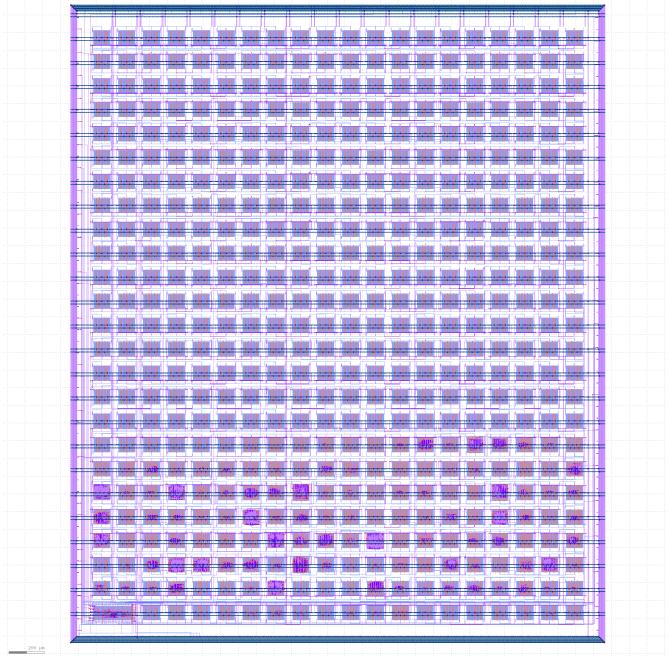


Fig. 5. 500 designs connected in a chain for TT01, with the scan chain driver in the lower left corner.

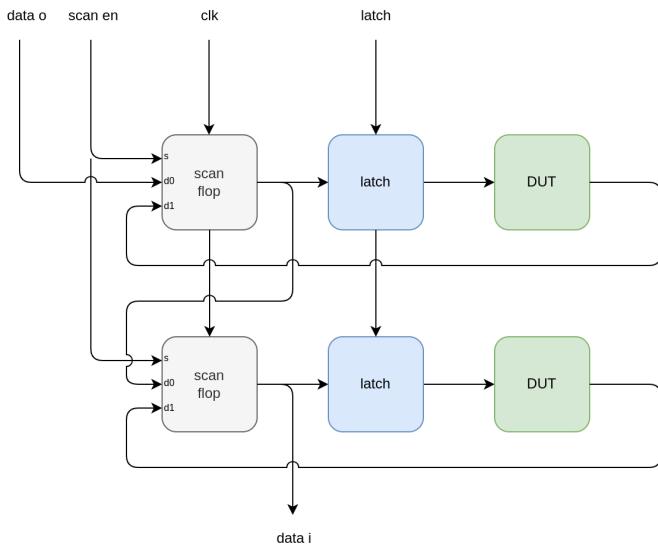


Fig. 6. A simplified view of 2 designs in the chain.

more hold margin.

After static timing analysis (STA) it was discovered that the clock duty cycle could change substantially due to the 500 sequential clock drivers. Depending on the clock buffers and capacitance between each design, the clock duty cycle could either increase or decrease, with this effect accumulated over the chain.

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. Fig. 8 shows a comparison between the TT02 and TT03 clock buffer designs. By inverting the clock between

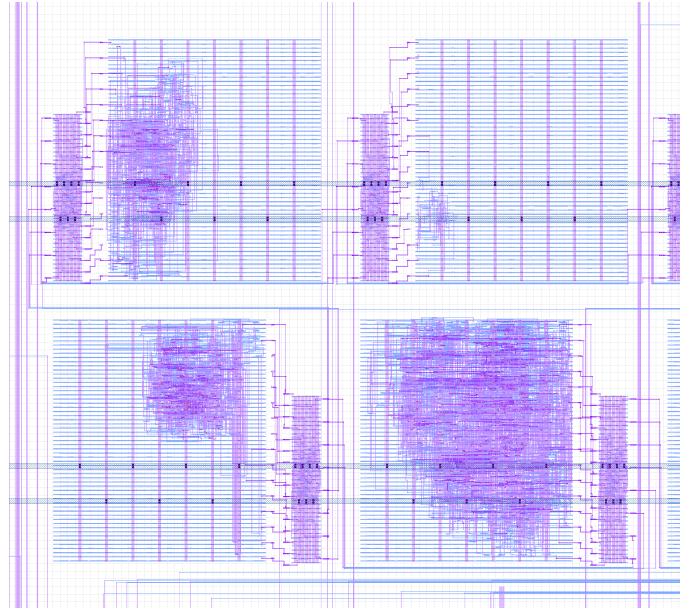
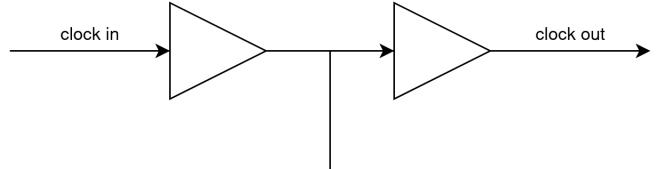


Fig. 7. TT02 designs with separate scan chain blocks.

TT02



TT03

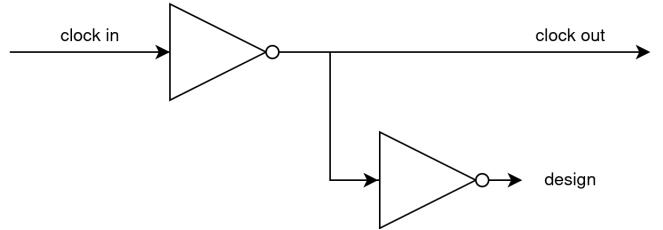


Fig. 8. TT03 buffers the output from the clock network into each design. Clock polarity is alternated between designs to minimize asymmetry between positive and negative cycles.

each design, any asymmetry in the clock pulse is evenly spread across the negative and positive cycles.

The verification effort [21] was broad and included a community review, register transfer level (RTL) and gate level (GL) simulation, Formal Verification [22], STA, layout vs schematic (LVS), DRC, and device level static verification [23].

#### IV. CIRCUIT BOARDS

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

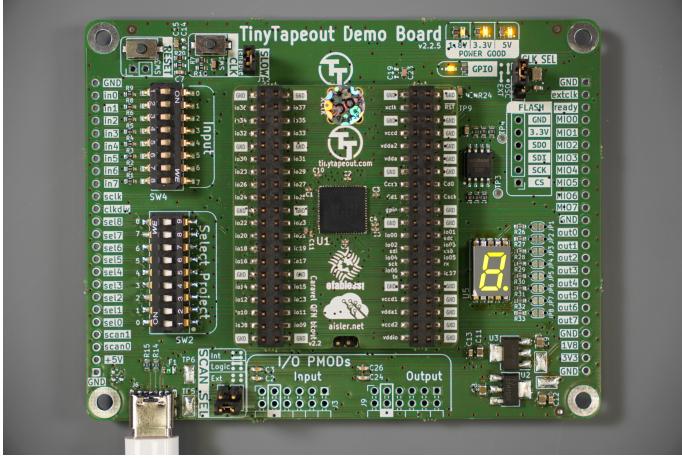


Fig. 9. The demonstration board. Certified Open Source Hardware ES000040 [24].

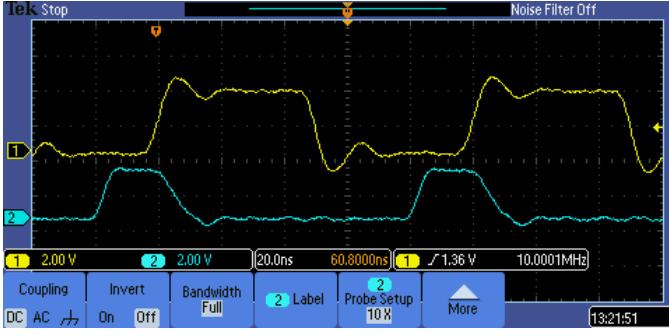


Fig. 10. Measurement from TT02 silicon, with input clock in yellow and the distorted output clock in blue.

The carrier fits onto the demonstration board shown in Fig. 9 which provides:

- USB-C for power connection,
- 1.8 V and 3.3 V power supplies for core and IO,
- 20 MHz 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.

## V. SCAN CHAIN 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 [25]. The chain was validated, and a few of the designs were shown to be working.

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

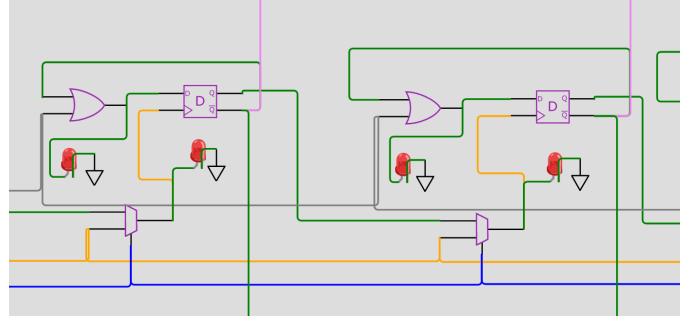


Fig. 11. Combinational logic in the clock path of one of the failed designs.

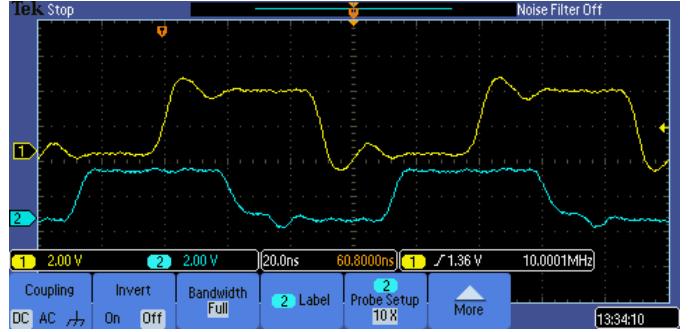


Fig. 12. Measurement from TT03 silicon.

After measuring the clock asymmetry (Fig. 10) and maximum frequency it was decided to run the production boards with a 20 MHz oscillator, resulting in a 10 MHz 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, 6 used alternative HDLs like VHDL, Amaranth [26] and Chisel [27]. Some Wokwi designs using combinational logic in clock paths (Fig. 11) worked in simulation but failed in hardware. This was due to the lack of timing data in the simulation, and wasn't detected by STA because the clock paths were not known. A detailed analysis has yet to be carried out.

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 scan chain shows a more symmetric (Fig. 12) output clock at the end of the chain. This will allow a faster scan chain clock, resulting in a faster update frequency.

## VI. BEYOND THE SCAN CHAIN

The biggest limitation of the scan chain based architecture was the IO bandwidth and latency. For TinyTapeout 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 shown in Fig. 13.

The physical layout (shown in Fig. 14) 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

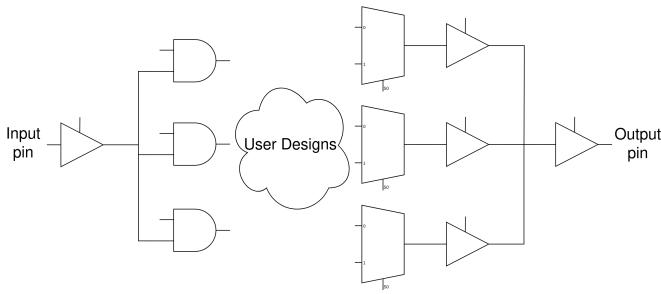


Fig. 13. Simplified diagram of the multiplexer architecture.

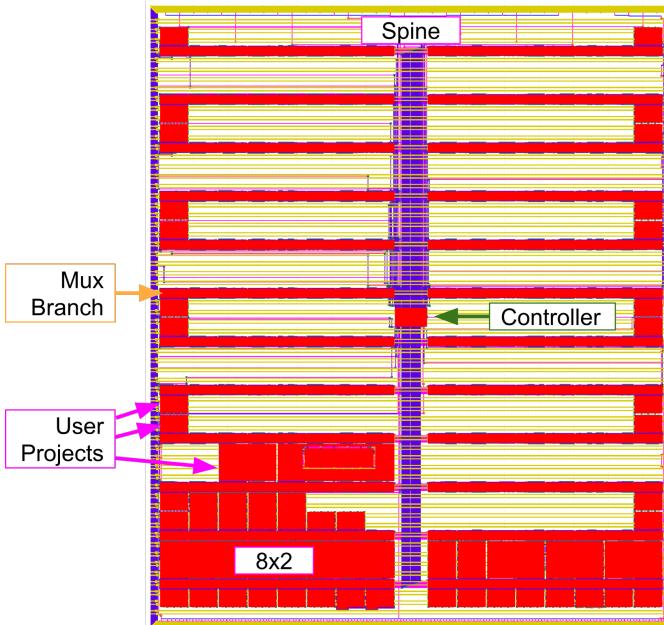


Fig. 14. The TT03.5 test design.

TABLE II  
COMPARISON BETWEEN TT03 AND TT04

Parameters	TT03	TT04
Max clock speed	12.5 kHz	50 MHz
Max design size	$150 \times 170 \mu\text{m}^2$	$1359 \times 225 \mu\text{m}^2$
Input pins	8	10
Output pins	8	8
Bidirectional I/O pins	None	8
Custom GDS file	X	✓

tile designs. Multiple tile designs were also enabled, allowing a maximum project size of  $8 \times 2$  tiles or  $1359 \times 225 \mu\text{m}^2$ —around 20 000 logic cells. Table II shows the key differences between TT03 and TT04.

Another major limitation of TT01 to TT03 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.

An invite-only experimental shuttle [28] was submitted with 32 designs to Efabless chipIgnite 2306C. Two of the designs

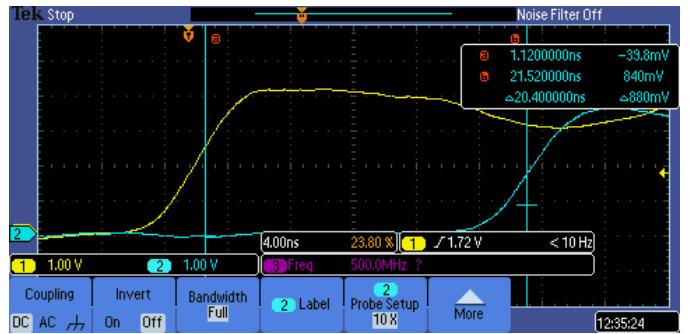


Fig. 15. Round trip latency on a rising edge of about 20 ns.



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

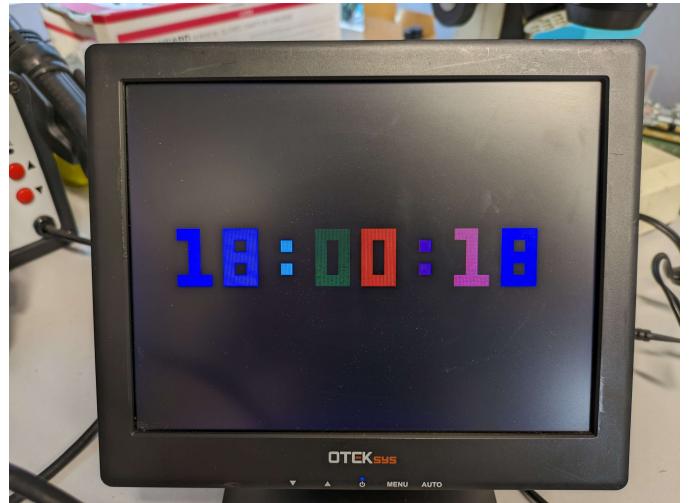


Fig. 17. VGA clock design running on TT03.5 silicon.

included a power gate as a stepping stone to supporting analog and mixed-signal designs.

## VII. MULTIPLEXER SILICON RESULTS

After silicon was received, the worst round trip latency was measured to be 20 ns as shown in Fig 15 and 16. Some designs have been validated, including a VGA clock project (Fig. 17) that takes advantage of the new higher speed IO.

The new chip pinout and serial design selection required a new demo board (Fig. 19) that included an easy way to select the design. The RP2040 microcontroller was chosen as a co-processor as it allows:

```
enabling design tt_um_test by sending 102 [0b01100110] pulses
design repo https://github.com/TinyTapeout/tt03p5-test at 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. 18. A MicroPython program [30] enabling a design, clocking it, and printing the results.

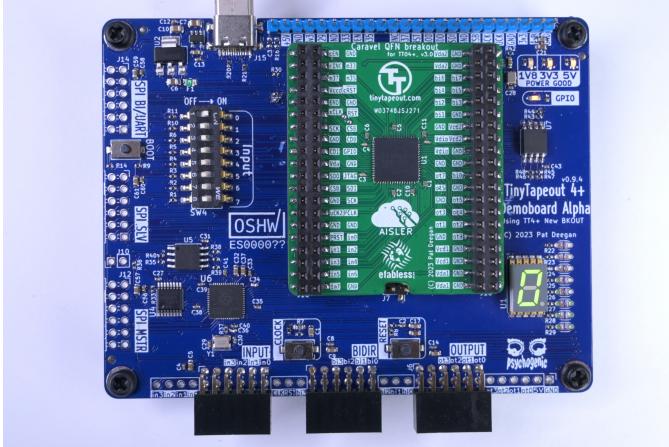


Fig. 19. The TT04+ demo board [31].

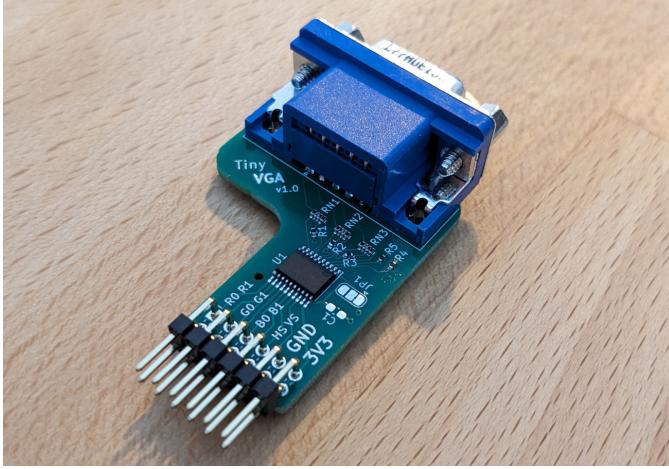


Fig. 20. A user-contributed VGA output PMOD.

- Drag and drop firmware updates on any OS,
- Runs MicroPython [29], ideal for beginners to enable and test their designs (Fig. 18),
- External memory emulation via PIO and DMA.

An additional PMOD expansion port was added for the bidirectional pins, and the community has started to standardize on pinouts [32] making it easier to test each other's designs. A new repository was created to house user-contributed PMODs [33], for example the VGA PMOD shown in Fig. 20.

### VIII. IMPROVING THE MULTIPLEXER AND MIXED SIGNAL SUPPORT

TT05 split the mux into two parts to improve performance. As each spine segment is now half as long, it will have half

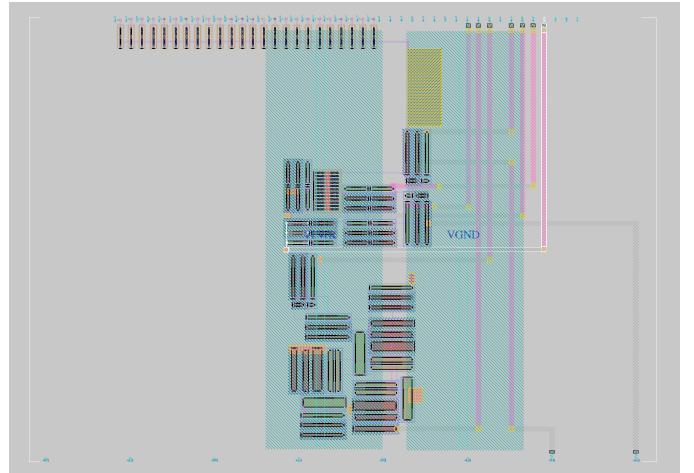


Fig. 21. Ring oscillator and DAC design submitted to TT05 with a transmission gate highlighted (auto-placed and auto-routed using an experimental analog P&R tool).

the capacitance. We expect to reduce the round trip latency to around 10 ns.

For TT06, the Caravel harness will be replaced by OpenFrame [34], an alternative harness provided by Efabless that uses the same padding but removes the RISC-V coprocessor. This results in 5 mm<sup>2</sup> more space for user designs, and 12 more pins that will be used for analog signals.

For increased safety, all designs will be power-gated, which allows 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 [35]. This allows up to 192 designs to share the analog pins between them. The transmission gates were tested as part of an experimental analog submission to TT05 shown in Fig. 21.

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.

### IX. SILICON SHOWCASE

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

- Serial FPGA (Link)
- Synthesizable Digital Temperature Sensor (shown in Fig. 22) (Link)
- 395 standard cells with mux (Link)
- FM transmitter with I2S input (Link)
- USB full speed - (Link)
- A Linux capable RISC-V CPU - (Link)

An index of all submitted designs can be found at [TinyTapeout.com/runs/](https://tinytapeout.com/runs/) [3].

### X. ACKNOWLEDGEMENTS

- 1) Uri Shaked for Wokwi development and lots more.
- 2) Sylvain Munaut for help with scan chain improvements and multiplexer design.
- 3) Mike Thompson and Mitch Bailey for verification expertise.

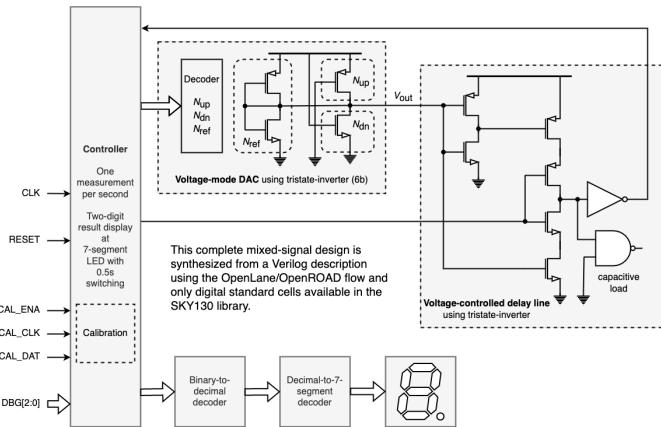


Fig. 22. The Synthesizable Digital Temperature Sensor.

- 4) Jix for formal verification support.
- 5) Proppy for help with GitHub actions.
- 6) Maximo Balestrini for the amazing renders and the interactive GDS viewer.
- 7) Harald Pretl for advice and analog support.
- 8) The team at YosysHQ and all the other open source EDA tool makers.
- 9) Efabless for running the shuttles and providing Open Lane and sponsorship.
- 10) Tim Ansell and Google for supporting the open source silicon movement.
- 11) Pat Deegan for PCB development.
- 12) The TinyTapeout community for all their contributions.
- 13) The Zero to ASIC course community for all their support.

## REFERENCES

- [1] M. Venn, “PDK,” <https://www.zerotoasiccourse.com/terminology/pdk/>, [Accessed January 28, 2024].
- [2] ——, “Datasheet PDF - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/raw/tt02/datasheet.pdf>, [Accessed January 28, 2024].
- [3] ——, “TinyTapeout Runs,” <https://tinytapeout.com/runs/>, [Accessed January 28, 2024].
- [4] ——, “First Shuttle - TinyTapeout Runs,” <https://tinytapeout.com/runs/tt01/>, [Accessed January 28, 2024].
- [5] E. Mahintorabi, “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 January 28, 2024].
- [6] Efabless, “Efabless,” <https://efabless.com/>, [Accessed January 28, 2024].
- [7] U. Shaked, “Wokwi,” <https://wokwi.com/>, [Accessed January 28, 2024].
- [8] M. Venn, “TinyTapeout,” <https://tinytapeout.com/>, [Accessed January 28, 2024].
- [9] GitHub, “GitHub,” <https://github.com/>, [Accessed January 28, 2024].
- [10] M. Venn, “GDS2 File Format - Zero to ASIC Course,” <https://www.zerotoasiccourse.com/terminology/gds2/>, [Accessed January 28, 2024].
- [11] U. Shaked, “Verilog Template - TinyTapeout Repository,” <https://github.com/TinyTapeout/tt06-verilog-template>, [Accessed January 28, 2024].
- [12] GitHub, “GitHub Actions Documentation,” <https://docs.github.com/en/actions>, [Accessed January 28, 2024].
- [13] Efabless, “OpenLane documentation,” <https://openlane.readthedocs.io/en/latest/>, [Accessed January 28, 2024].
- [14] S. P. Authors, “Skywater PDK Documentation,” [https://skywater-pdk.readthedocs.io/en/main/contents/libraries/sky130\\_fd\\_sc\\_hdll/cells/sdfxtp/README.html](https://skywater-pdk.readthedocs.io/en/main/contents/libraries/sky130_fd_sc_hdll/cells/sdfxtp/README.html), [Accessed January 28, 2024].
- [15] S. Williams, “iVerilog,” [https://github.com/steveicarus/iverilog/](https://github.com/steveicarus/iverilog), [Accessed January 28, 2024].
- [16] F. Foundation, “cocotb,” <https://www.cocotb.org/>, [Accessed January 28, 2024].
- [17] P. Deegan, “Wokwi Automated Testing - TinyTapeout,” [https://tinytapeout.com/digital\\_design/wokwi\\_automated\\_testing/](https://tinytapeout.com/digital_design/wokwi_automated_testing/), [Accessed January 28, 2024].
- [18] U. Shaked, “Locally installing the tools,” <https://docs.google.com/document/d/1aUZ1jthRpg4QURIIyzlOaPWImQzr-jBn3wZipVUPt4>, [Accessed January 28, 2024].
- [19] M. Venn, “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 January 28, 2024].
- [20] J. Harder, “TinyTapeout Scan - GitHub Repository,” [https://github.com/jix/tinytapeout\\_scan](https://github.com/jix/tinytapeout_scan), [Accessed January 28, 2024].
- [21] M. Venn, “Verification Documentation - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/blob/tt02/VERIFICATION.md>, [Accessed January 28, 2024].
- [22] YosysHQ, “SymbiYosys (SBY) - YosysHQ Repository,” <https://github.com/YosysHQ/sby>, [Accessed January 28, 2024].
- [23] M. Bailey, “CVC - Verification Documentation - TinyTapeout Repository,” <https://github.com/TinyTapeout/tinytapeout-02/blob/tt02/VERIFICATION.md#cvc>, [Accessed January 28, 2024].
- [24] OSHWA, “Open Source Hardware Certification - TinyTapeout,” <https://certification.oshwa.org/es000040.html>, [Accessed January 28, 2024].
- [25] M. Venn, “TT02 Silicon is Alive! - Zero to ASIC Course Blog,” <https://www.zerotoasiccourse.com/post/tt02-silicon-is-alive/>, [Accessed January 28, 2024].
- [26] C. ‘whitequark’, “Amaranth,” <https://amaranth-lang.org/docs/amaranth/latest/>, [Accessed January 28, 2024].
- [27] ChipsAlliance, “Chisel,” <https://www.chisel-lang.org/>, [Accessed January 28, 2024].
- [28] M. Venn, “TinyTapeout 03p5 - GitHub Repository,” <https://github.com/TinyTapeout/tinytapeout-03p5>, [Accessed January 28, 2024].
- [29] G. R. Limited, “MicroPython Official Website,” <https://micropython.org/>, [Accessed January 28, 2024].
- [30] M. Venn, “Demo Firmware Test Script - TinyTapeout Repository,” <https://github.com/TinyTapeout/tt3p5-demo-fw/blob/main/tt3p5-test/test.py#L119>, [Accessed January 28, 2024].
- [31] P. Deegan, “TT04+ Demoboard PCB - TinyTapeout Repository,” <https://github.com/TinyTapeout/tt-demo-pcb>, [Accessed January 28, 2024].
- [32] L. Moser, “Pinouts Specifications - TinyTapeout,” <https://tinytapeout.com/specs/pinouts/>, [Accessed January 28, 2024].
- [33] U. Shaked, “Awesome TinyTapeout PMODs - GitHub Repository,” <https://github.com/TinyTapeout/awesome-tinytapeout-pmods>, [Accessed January 28, 2024].
- [34] Efabless, “Caravel Openframe Project,” [https://github.com/efabless/caravel\\_openframe\\_project](https://github.com/efabless/caravel_openframe_project), [Accessed January 28, 2024].
- [35] H. Pretl, “TT05 Analog Test,” <https://github.com/iic-jku/tt05-analog-test>, [Accessed January 28, 2024].