

Tiny Tapeout: A Shared Silicon Tapeout Platform Accessible To Everyone

Matt Venn
 Tiny Tapeout, YosysHQ
 Email: matt@tinytapeout.com

I. INTRODUCTION

TINY TAPEOUT is a multi project chip platform that makes it easier and cheaper to get application specific integrated circuit (ASIC) designs manufactured.

Open source tools and process design kits (PDK [1]) are used so no restrictive licenses or non disclosure agreement (NDAs) are required. As the tools run on remote cloud servers no software needs to be installed locally on the user's machine. As long as the template structure is followed, however, Tiny Tapeout can support the use of proprietary tools.

Each Tiny Tapeout ASIC production run has around 400 open source designs multiplexed to 24 general purpose input/output (GPIO) pins. After manufacture the resulting chip is mounted to a demonstration board for ease of testing. Each chip contains a copy of every design, which can be selected and tested in turn.

At the same time each participant submits documentation for their design, which is used to create a printable datasheet [2] alongside an online project index at TinyTapeout.com/runs/ [3]. The datasheet helps participants explore other designs on the chip in addition to their own.

By separating the cost of area on a silicon wafer and the finished physical chip, the Tiny Tapeout participant group is able to share the cost of chip packaging and of circuit board manufacture while still being able to test and measure all the designs on the chip. For use in educational settings it is possible for multiple students to submit individual designs while sharing the finished chips and circuit boards, reducing the cost still further.

Each Tiny Tapeout tile (Fig. 1) is approximately $160 \times 100 \mu\text{m}^2$. This provides enough room for around 1000 logic gates when built upon the SkyWater 130nm open source PDK. Multiple tiles can be interconnected to enable larger designs, while analog and mixed signal support is on the roadmap for the next shuttle.

Community engagement in Tiny Tapeout has been strong, with 756 designs submitted over the first five shuttles. A curated selection of projects is provided in section XI.

An online chat server for participants has 1000 members with 1600 subscribers to the project's mailing list. Individuals submitting designs to Tiny Tapeout tend to self identify as hobbyists, students, and teachers, as shown in Fig. 2.

The first [4] Tiny Tapeout production run, which was provided as a free and experimental effort with a total of 152 designs, was submitted to the seventh Google-sponsored [5]

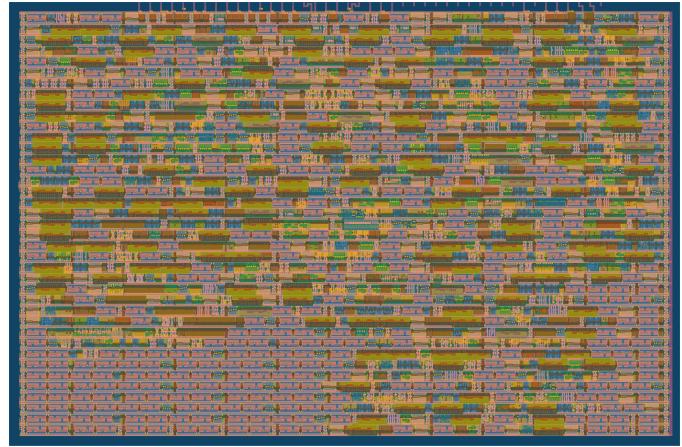


Fig. 1. A 2-D render of a single Tiny Tapeout tile.

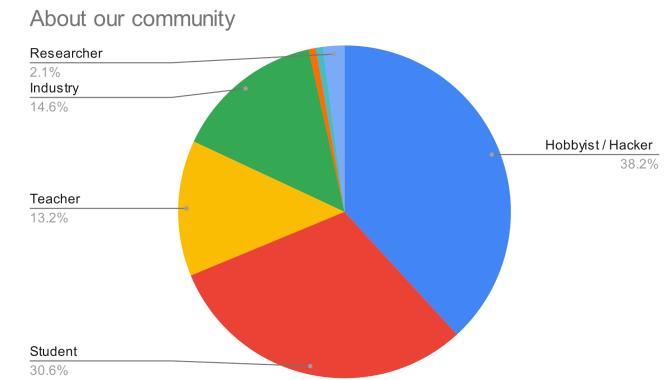


Fig. 2. Tiny Tapeout 4 participant self identification.

lottery based multi project wafer (MPW) shuttle in September 2022. The next four shuttles combined a total of 582 designs, all sponsored by and manufactured through the Efabless [6] chipIgnite MPW service. Table I shows a summary of all Tiny Tapeout shuttle runs to date.

The remainder of this paper will detail the Tiny Tapeout submission flow, multiplexer evolution, circuit board design, the results of post production silicon testing, and the project's next steps.

II. SUBMISSION FLOW

Tiny Tapeout designs are primarily developed in the Verilog hardware description language (HDL) or Wokwi [7] Wokwi is

TABLE I
TINY TAPEOUT SHUTTLE SUMMARY

Run	Launched	Shuttle	Designs	Delivery date	Architecture	Number of IOs	IO bandwidth	Analog support
TT01	2022-08-17	MPW7	152	n/a	Scan chain	16	5 kHz	no
TT02	2022-11-09	2211Q	165	2024-01-30	Scan chain	16	5 kHz	no
TT03	2023-03-01	2304C	249	2024-02-28	Scan chain inverted clock	16	10 kHz	no
TT04	2023-07-01	2309	143	2024-04-15	Mux	26	50 MHz	no
TT05	2023-09-11	2311	174	2024-05-12	Split Mux	26	50 MHz	no
TT06	2024-02-01	2404	TBD	2024-11-30	Split Mux	38	50 MHz	yes

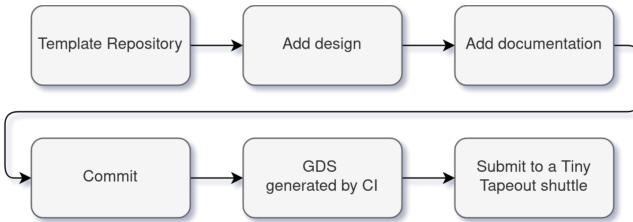


Fig. 3. Design Submission Flow

a web based visual schematic editor for hardware description, designed as an easier way for individuals with no prior HDL experience to get started. The Tiny Tapeout website [8] includes a basic getting started guide for Wokwi, demonstrating how to use the tool to draw circuits, which is made available in English and Spanish.

The process of designing an ASIC for submission to Tiny Tapeout, whether with Wokwi, Verilog, or another HDL, matches existing workflows built around the same tools and is outside the scope of this article; instead, we focus here on the aspects of the submission flow which differ from traditional ASIC design and production flows.

The submission flow starts with the participant creating a GitHub [9] source code repository based on a template provided by Tiny Tapeout. Using this template and guidance provided through the Tiny Tapeout website and broader communication channels the participant then creates their chip design in Verilog, Wokwi, or the HDL of their choice.

The participant's design must then be documented, after which the design and its documentation can be submitted to the Tiny Tapeout GitHub repository. This submission triggers automated tests and the generation of binary layout files in GDSII [10] format.

This automated testing and GDS generation is handled through the Tiny Tapeout GitHub templates [11] using GitHub Actions [12]—an automatic continuous integration system triggered every time the repository is updated. This reduces duplicated effort and makes it possible for Tiny Tapeout to support large numbers of participants without excessive technical overhead.

There are four main jobs in the continuous integration system:

- 1) GDS: installs OpenLane [13] and the SkyWater Sky130 [14] PDK, builds the binary layout files, and generates a summary of the design (Fig. 4). The summary includes

utilization, standard cells used, a 2-D render (Fig. 1) and an interactive 3-D viewer (Fig. 5). This job can also optionally run a gate-level verification of the design.

- 2) Verification: installs the YosysHQ open source computer-aided design (CAD) suite, which includes many common electronic design automation (EDA) tools; uses iVerilog [15] and cocotb [16] to run included testbenches.
- 3) Documentation: generates a preview of the documentation.
- 4) Precheck: runs design rule check (DRC) tests to ensure the design can be integrated into the multi project chip.

Successful GDS, Documentation, and Precheck job completion are all required for a design to be submitted to a shuttle for production. Verification is optional but highly encouraged. Submissions designed in Wokwi are able to make use of its integrated truth table testing system [17].

If all tests pass and the binary layout files are correctly generated the design can then be submitted to a quarterly shuttle for production in silicon, alongside all other passing designs from participants in that Tiny Tapeout run. Projects are submitted to a shuttle through the Tiny Tapeout website [8]. Participants are free to update submitted projects up until the closing date of the shuttle, providing the tests continue to pass.

While the Tiny Tapeout continuous integration system can be run entirely in the user's web browser, it is also possible to install a local copy of the tools [18] on a participant's computer. Locally installed tools can help to reduce the time between design iterations, especially for the test and verification jobs.

With its focus on automated testing Tiny Tapeout is able to minimize design errors common to those coming to ASIC design with little or no prior experience, reducing wastage at the production stage and maximizing

III. TINY TAPEOUT 1

The Tiny Tapeout project started as an experiment in fitting as many designs as possible into the 10 mm² available on the Google lottery shuttles (Fig. 6). In order to prove the concept as rapidly as possible initial designs were based on a scan chain architecture to simplify testing.

Each Tiny Tapeout 1 design has eight inputs and eight outputs. Clock and reset signals were optional and not treated specially. The chain was formed of scan flops [14], a type of flip flop with a multiplexer integrated at its input. An example showing a two design scan chain is shown in Fig. 7.

Cell usage by Category

Category	Cells	Count
Fill	decap fill	1145
Combo Logic	o21ai nand3b a221o o31a or2b a21o a21bo nor2b o31ai a41o a21oi a211o o21ba o21a o21ba1 a2bb2o or3b o221a a31o a22o o32a o22a a32o a211o and2b and3b or4b or4bb o211a a211oi o2bb2a a211oi and4bb o211ai a220i a31oi o311a	249
Tap	tapvlpwrvgnd	246
Flip Flops	dfxtp	146
Buffer	buf clkbuf	127
AND	and2 a21boi and3 and4	97
Misc	dlygate4sd3 dlymetal6s2s comb	84
OR	or3 xor2 or2 or4	81
NOR	xnor2 nor2 nor3 nor4	64
NAND	nand2 nand3 nand4 nand2b	52
Inverter	inv	37
Multiplexer	mux2 mux4	9
Diode	diode	1

947 total cells (excluding fill and tap cells)

Fig. 4. A summary table from the GDS continuous integration job.



Fig. 5. The interactive 3-D viewer.

Each design sends data into the secondary input of the scan flop and receives its own 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 primary input of the next scan flop. This arrangement allows the loading of data into any of the designs, followed by the capturing of the output and its clocking through the rest of the chain to the overall chain output.

While relatively easy to implement, a scan chain architecture has a downside: high latency. As more designs are added to the chain it takes longer to send and receive data through it. For example: assuming a 50 MHz scan chain clock with 250 designs each having eight inputs and eight outputs, the maximum refresh rate of the resulting chain is $50 \text{ MHz} / (8 \times 250) = 25 \text{ kHz}$.

The Tiny Tapeout 1 scan chain was embedded into each design, meaning a user could unintentionally remove it and break the chain. This risk was mitigated with a formal [20] equivalence check which proves the chain was present in each submitted design, preventing their submission if it had been removed.

Another concern with the scan chain design was hold violations, due to the large number of serially connected flops

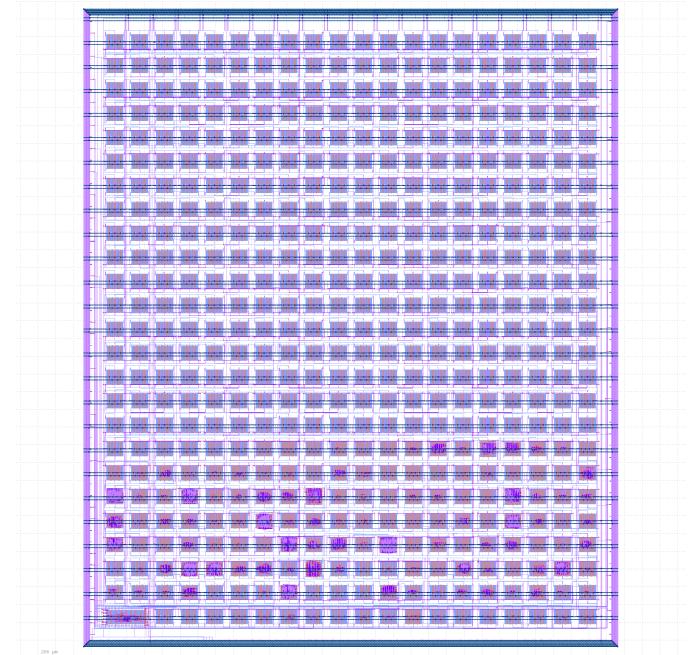


Fig. 6. 500 designs connected in a chain for Tiny Tapeout 1; the scan chain driver can be seen in the lower left corner.

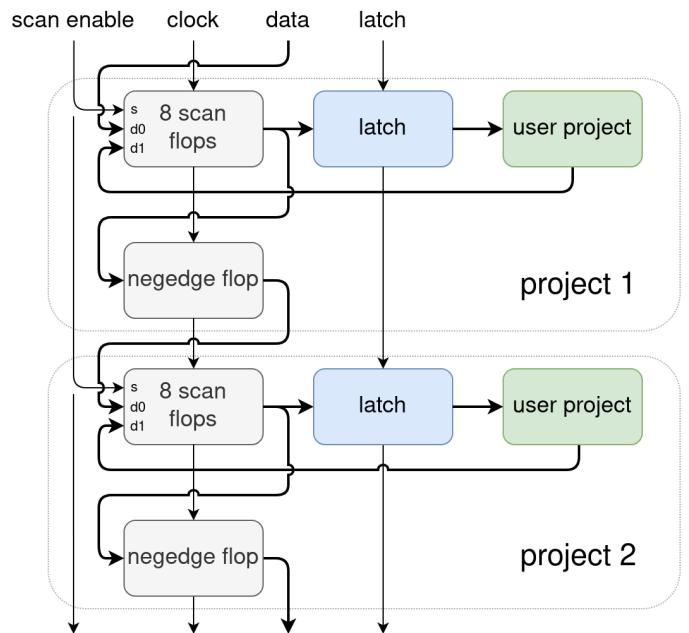


Fig. 7. A simplified view of two Tiny Tapeout 1 designs in the scan chain.

and potentially large clock skews over long signal wires. This was mitigated by reclocking the output data with a negative edge (negedge) flop, providing substantially more hold margin.

Following static timing analysis (STA) it was discovered that the clock duty cycle could change substantially due to the 500 sequential clock drivers in the chain. 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.

The verification effort [21] for the scan chain was broad

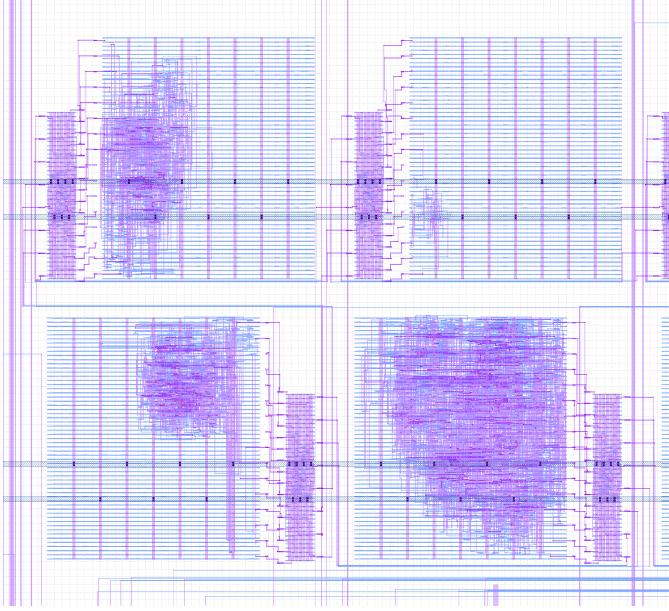


Fig. 8. Tiny Tapeout 2 designs, showing the discrete scan chain blocks.

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. TINY TAPEOUT 2

To address the issue of accidental macro block removal encountered in Tiny Tapeout 1, Tiny Tapeout 2 moved the scan chain into a discrete macro block. This was present in all designs, but could not be modified by participants.

As with Tiny Tapeout 1, designs in in Tiny Tapeout 2 used two clock buffers, with the internal flops driven after the first buffer.

V. TINY TAPEOUT 3

For Tiny Tapeout 3 the two clock buffers of Tiny Tapeout 1 and 2 were replaced by an inverting clock buffer design, with only one buffer between the clock input and output. Fig. 9 shows a comparison between the TT02 and TT03 clock buffer designs. By inverting the clock between each design any asymmetry in the clock pulse is evenly spread across the negative and positive cycles.

Following the closure of Tiny Tapeout 3, an invitational experimental shuttle dubbed Tiny Tapeout 3.5 [24] was submitted for production. This featured with 32 designs, testing and previewing some of the changes planned for Tiny Tapeout 4, detailed in the next section. These designs submitted for production in Efabless chipIgnite 2306C. Two of these designs included a power gate as a stepping stone to supporting analog and mixed signal designs.

VI. TINY TAPEOUT 4

The biggest limitations of the scan chain based architecture used in Tiny Tapeout 1 through 3 inclusive were its limited

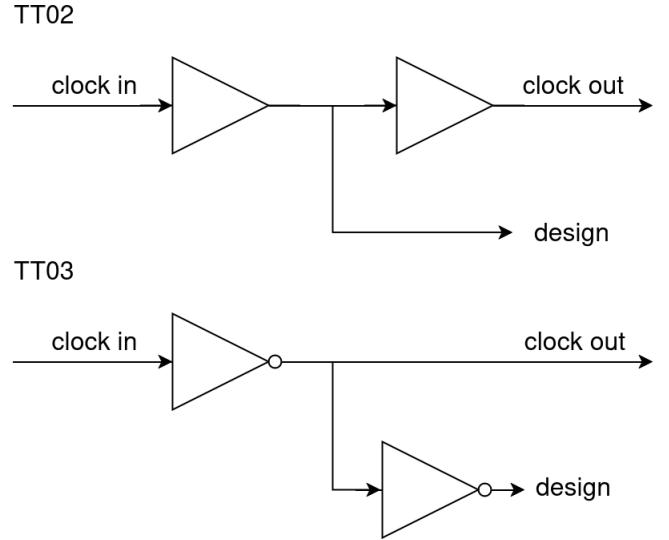


Fig. 9. The Tiny Tapeout 3 architecture buffers the output from the clock network into each design. Clock polarity is alternated between designs to minimize asymmetry between positive and negative cycles.

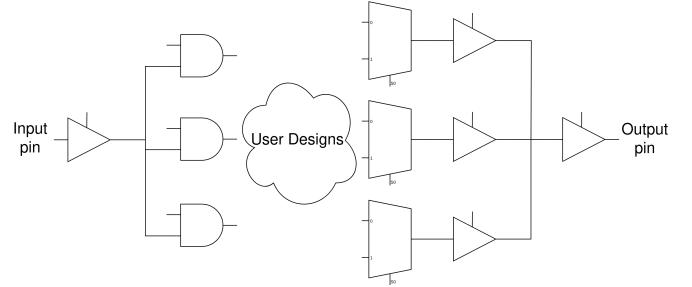


Fig. 10. A simplified diagram of the Tiny Tapeout 4 multiplexer architecture.

IO bandwidth and high latency. It was decided that a new architecture was needed for Tiny Tapeout 4 and proposals were gathered from the community. An online video call was held with community members, and the ten submitted architecture proposals discussed.

The winning architecture was a straightforward multiplexer design, shown in Fig. 10. This architecture was chosen as the simplest to implement while providing the most benefit in terms of additional IOs and higher bandwidth.

The physical layout tested in Tiny Tapeout 3.5 and finalized in Tiny Tapeout 4 (shown in Fig. 11) consists of a central controller connected up and down to two vertical wiring spines. For this experimental run we also included loopback test designs at the end of each multiplexer so we could measure performance for each multiplexer position.

Twenty-four horizontal multiplexers connect to the wiring spines, each of which supports 16 designs. This allows for up to 384 separate single tile designs. The new architecture also enabled multiple tile designs, allowing a maximum project size of 8×2 tiles or $1359 \times 225 \mu\text{m}^2$ —around 20 000 logic cells. Table II shows the key differences between Tiny Tapeout 3 and Tiny Tapeout 4.

Another major limitation of the scan chain architecture used in Tiny Tapeout 1 through Tiny Tapeout 3 was the small

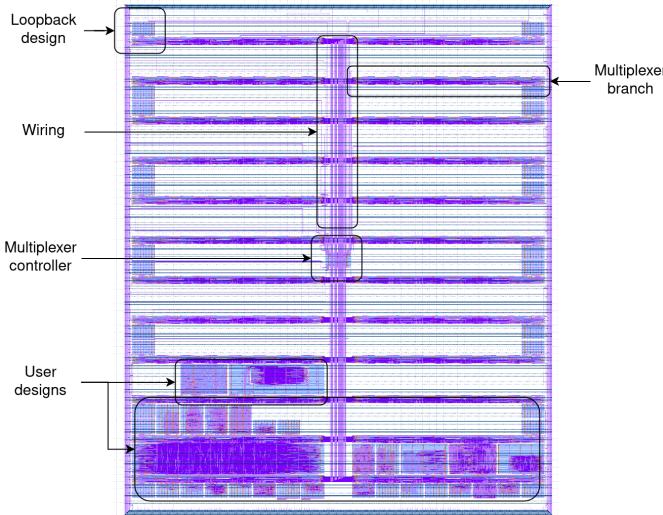


Fig. 11. The Tiny Tapeout 3.5 test design.

TABLE II
A COMPARISON BETWEEN TINY TAPEOUT 3 AND TINY TAPEOUT 4.

Parameters	TT03	TT04
Max clock speed	8 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 IO pins	None	8
Custom GDS file	X	✓

number of IO pins. The scan controller used nine GPIOs to select the currently active design. While this simplified the demonstration board board, it also wasted valuable IO pins.

Starting with Tiny Tapeout 4 the parallel design selection architecture used in previous chips was dropped in favor of a serial protocol. The extra pins thus provided were then used bidirectionally, giving each design a clock pin, reset pin, and 24 IO pins.

VII. TINY TAPEOUT 5

Tiny Tapeout 5 saw another change in the design to split the multiplexer into two parts in order to improve performance. As a result of the split the controller was also updated to multiplex between the two halves. Some other small changes to the multiplexer in Tiny Tapeout 5 included the tweaking, addition, or removal of buffers to improve the STA results.

As each spine segment in Tiny Tapeout 5 is now half as long as in Tiny Tapeout 4, it will exhibit half the capacitance. As a result, upon receipt and testing of fabricated chips we expect to see the round trip latency reduced to around 10 ns.

A bigger change in Tiny Tapeout 5 was the inclusion of an experimental analog design submission, shown in Fig. 12. This was created to test support for analog and mixed signal designs planned for Tiny Tapeout 6.

VIII. TINY TAPEOUT 6

For Tiny Tapeout 6 the Caravel harness will be replaced by OpenFrame [25], an alternative harness provided by Efabless

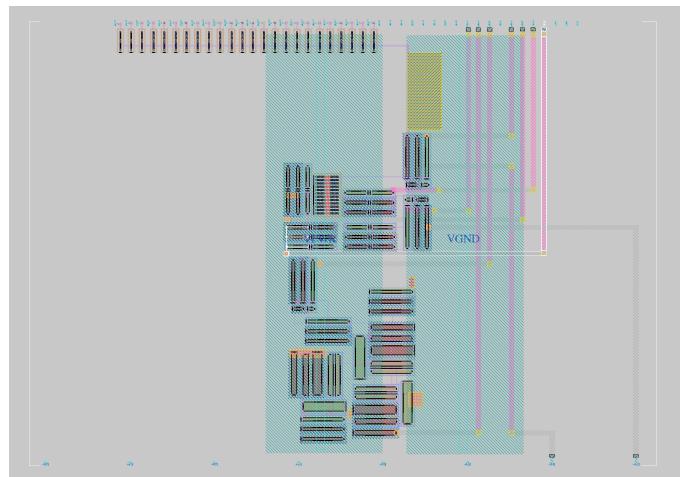


Fig. 12. A ring oscillator and digital to analog converter (DAC) design submitted to Tiny Tapeout 5, with a transmission gate highlighted. This design was automatically placed and routed using an experimental analog place and route (P&R) tool.

that uses the same padding but removes the RISC-V coprocessor. This results in an additional 5 mm^2 of space for user designs, and 12 more IO pins which will be used for analog signals.

For increased safety, all designs will be power-gated. This allows designers to take more risks in their submissions, or to use custom flows not previously used in Tiny Tapeout.

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 analog pins between them.

Noise coupling between analog and digital power domains is a known concern. However, due to other limitations of our current setup including its limited number of low bandwidth analog interfaces, we target educational low to medium performance analog and mixed signal designs where noise coupling is a lesser concern.

Tiny Tapeout 6 is planned to open for submission of digital designs at the end of January 2024, for analog designs at the end of February 2024, and to close for submissions on 19 April 2024.

IX. CIRCUIT BOARDS

After manufacture Tiny Tapeout chips are mounted onto small carrier boards with 0.1 inch pin headers. These carriers allow people with limited surface mount technology (SMT) assembly experience to build their own demonstration boards around the chips.

The carrier fits onto the demonstration board shown in Fig. 13. The demonstration board is designed primarily for ease of use by beginners, though offers enough flexibility for power users. As all signals are below 50 MHz, no special layout was needed in the design of the demonstration board.

The demonstration board provides:

- USB Type-C as a power connection,
- 1.8 V and 3.3 V power supplies for core and IO,
- 20 MHz oscillator,

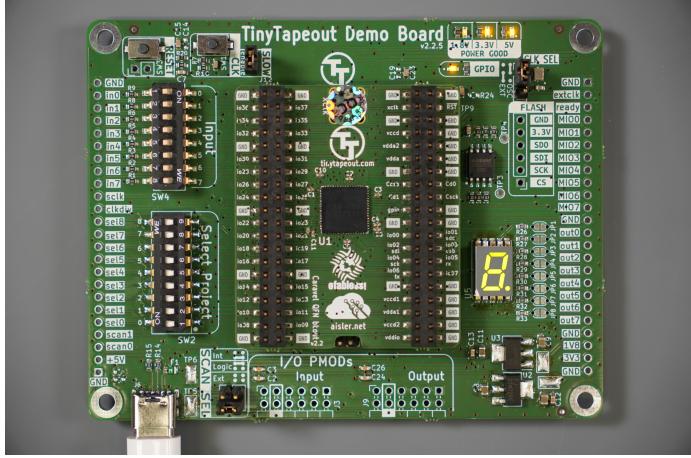


Fig. 13. The demonstration board, which has been recorded as Certified Open Source Hardware ES000040 [27].

```
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. 14. A MicroPython program [29] enabling a chosen design, clocking it, and printing the results.

- Buttons for reset and single step clock,
- An eight way DIP switch for inputs,
- A nine way DIP switch for design selection,
- A seven segment LED display for the outputs,
- Headers for all IO, including two standard Digilent ports (PMOD),
- A header to select the internal clock or to provide one externally,
- A header to select an internal or external scan chain driver,
- A header to engage an automatic clock divider in input pin zero.

The move away from the scan chain architecture in Tiny Tapeout 3.5 and Tiny Tapeout 4 necessitated the design of a new demonstration board (Fig. 15), which had to include an easy way to select a chosen design. The Raspberry Pi RP2040 microcontroller was chosen as a coprocessor on the revised demonstration board, as it offers:

- Drag and drop firmware updates on any operating system,
- MicroPython [28] support, an ideal language and programming environment through which beginners can enable and test their designs (Fig. 14),
- External memory emulation via programmable input output (PIO) and direct memory access (DMA) support.

An additional PMOD expansion port was added to the demonstration board for the bidirectional pins. The Tiny Tapeout community has begun to standardize on pinouts [31], making it easier to test each design. A new repository was created to house user-contributed PMOD [32] accessories, for example the VGA PMOD shown in Fig. 16.

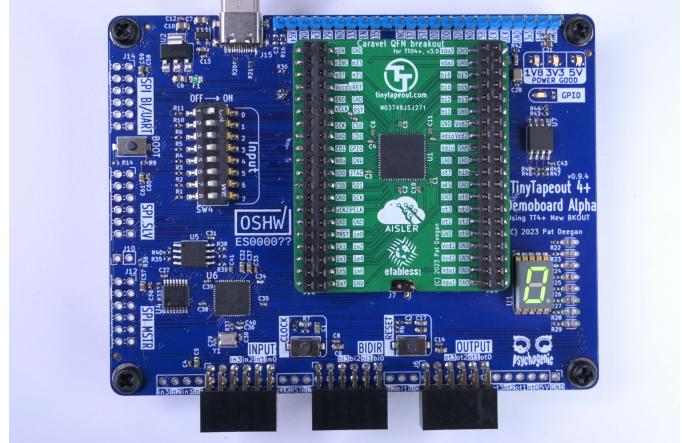


Fig. 15. The Tiny Tapeout 4+ demonstration board [30].

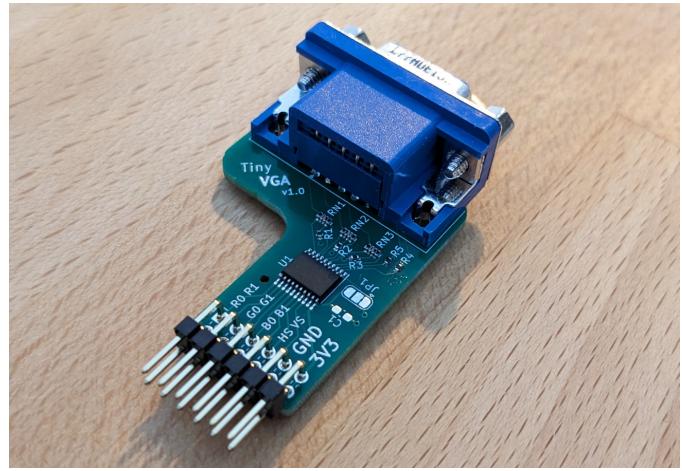


Fig. 16. A user-contributed VGA output PMOD accessory for the demonstration board.

A further set of three PMOD expansion ports were added, mixing input and outputs, which allows the most common standard PMODs to be used with the demonstration board. For more information about the circuit board, pinout, and PMOD support see the repository [30].

X. RESULTS AND MEASUREMENTS

Tiny Tapeout 2 chips were received in October 2023, 11 months after the chips were submitted for manufacture on the Efabless chipIgnite 2211Q shuttle. The chips were tested for the first time on a public livestream [33]. During this testing the chain was validated and a small number of the designs were demonstrated to be working. In further testing following the livestream another 30 designs were tested and shown to be working.

After measuring the clock asymmetry (Fig. 17) and maximum frequency it was decided to run the production boards with a 20 MHz oscillator, resulting in a 10 MHz scan chain clock and a 5 kHz IO update rate in order to maximize stability.

Some Tiny Tapeout 2 designs did not function as expected; in most cases this was determined to be due to faults in the submitted design.

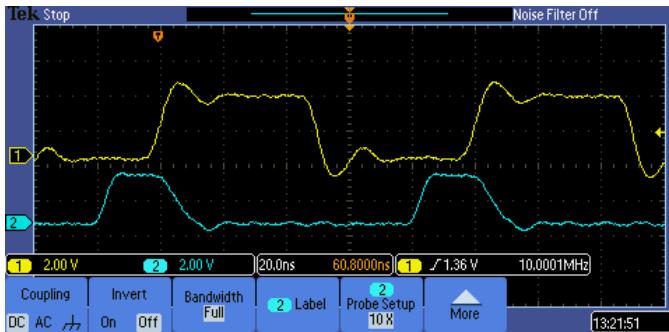


Fig. 17. Measurement from Tiny Tapeout 2 silicon, with input clock in yellow and the distorted output clock in blue.

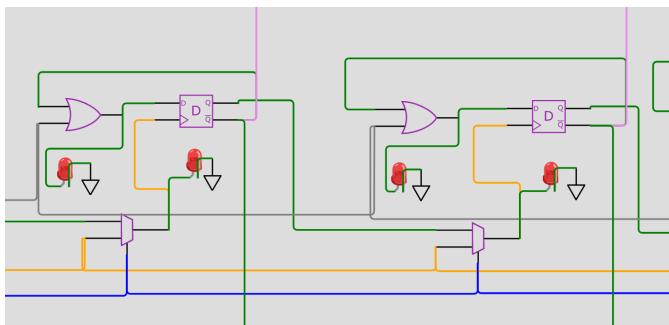


Fig. 18. Combinational logic in the clock path of one of the failed designs.

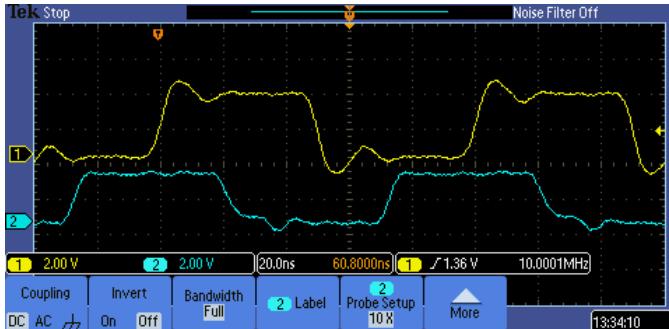


Fig. 19. Measurement from Tiny Tapeout 3 silicon.

Of the designs submitted 82 used the Verilog HDL, 64 were created using the Wokwi graphical editor, and six used alternative HDLs including VHDL, Amaranth [34], and Chisel [35].

Some of the designs created using Wokwi using combinational logic in clock paths (Fig. 18) worked in simulation but failed in hardware. This was determined to be due to the lack of timing data in the simulation, and was not detected by STA because the clock paths were not known. A detailed analysis on these failures 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.

Silicon from Tiny Tapeout 3 production was received in January 2024. The updated scan chain design shows a more symmetric output clock at the end of the chain (Fig. 19). The improved clock symmetry will allow for the use of a faster scan chain clock, resulting in a faster update frequency.

Some Tiny Tapeout 3.5 designs have already been validated,

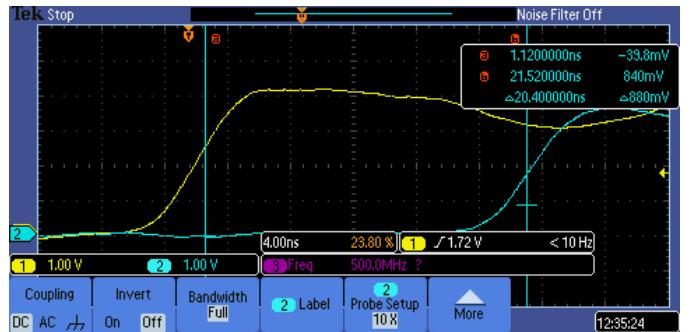


Fig. 20. The Tiny Tapeout 3.5 round trip latency on a rising edge, measured at about 20 ns.



Fig. 21. The Tiny Tapeout 3.5 round trip latency on a falling edge, measured at about 16 ns.

including a VGA clock project (Fig. 22) which takes advantage of the new higher speed IO.

The overhead of multiplexing multiple tiles makes power consumption of the infrastructure only a minor concern at this point. No direct comparison of the power consumption impact of the multiplexer over the previous scan chain architecture is available at the time of writing. The motivation of the multiplexer approach, however, was to substantially increase the bandwidth of the IOs.

After Tiny Tapeout 4 silicon was received and tested, the worst round trip latency was measured to be 20 ns as shown in Fig 20 and 21.

XI. SILICON SHOWCASE

A curated list showcasing the types of designs possible through Tiny Tapeout is provided below:

- Serial FPGA [36],
- Synthesizable digital temperature sensor [37],
- 395 standard cells with multiplexer [38],
- FM transmitter with I2S input [39],
- USB Full Speed device [40],
- A Linux capable RISC-V CPU [41].

An index of all designs submitted to Tiny Tapeout can be found at TinyTapeout.com/runs/ [3].

ACKNOWLEDGMENTS

- 1) U. Shaked for Wokwi development and lots more.
- 2) S. Munaut for help with scan chain improvements and multiplexer design.



Fig. 22. A VGA clock design running on Tiny Tapeout 3.5 silicon.

- 3) M. Thompson and M. Bailey for verification expertise.
- 4) Jix for formal verification support.
- 5) Proppy for help with GitHub actions.
- 6) M. Balestrini for 3-D renders and the interactive GDS viewer.
- 7) H. 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) T. Ansell and Google for supporting the open source silicon movement.
- 11) P. Deegan for PCB development.
- 12) The Tiny Tapeout community for all its contributions.
- 13) The Zero to ASIC course community for all its support.

REFERENCES

- [1] M. Venn, “PDK, Zero to ASIC Course,” <https://www.zerotoasiccourse.com/terminology/pdk/>, (accessed March 12, 2024).
- [2] M. Venn, “Datasheet PDF, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tintytapeout-02/raw/tt02/datasheet.pdf>, (accessed March 12, 2024).
- [3] M. Venn, “Tiny Tapeout Runs,” <https://tinytapeout.com/runs/>, (accessed March 12, 2024).
- [4] M. Venn, “First Shuttle, Tiny Tapeout Runs,” <https://tinytapeout.com/runs/tt01/>, (accessed March 12, 2024).
- [5] E. Mahintorabi, J. Euphrusine, and A. Cunningham, “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 March 12, 2024).
- [6] Efabless, “Efabless Home Page,” <https://efabless.com/>, (accessed March 12, 2024).
- [7] U. Shaked, “Wokwi home page,” <https://wokwi.com/>, (accessed March 12, 2024).
- [8] M. Venn, “Tiny Tapeout Home Page,” <https://tinytapeout.com/>, (accessed March 12, 2024).
- [9] GitHub, “GitHub Home Page,” <https://github.com/>, (accessed March 12, 2024).
- [10] M. Venn, “GDS2 File Format, Zero to ASIC Course,” <https://www.zerotoasiccourse.com/terminology/gds2/>, (accessed March 12, 2024).
- [11] U. Shaked, “Verilog Template, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tt06-verilog-template>, (accessed March 12, 2024).
- [12] GitHub, “GitHub Actions Documentation,” <https://docs.github.com/en/actions>, (accessed March 12, 2024).
- [13] Efabless, “OpenLane Documentation,” <https://openlane.readthedocs.io/en/latest/>, (accessed March 12, 2024).
- [14] S. P. Authors, “Skywater PDK Documentation,” https://skywater-pdk.readthedocs.io/en/main/contents/libraries/sky130_fd_sc_hdll/cells/sdfxtpl/README.html, (accessed March 12, 2024).
- [15] S. Williams, “iVerilog, GitHub Repository,” <https://github.com/steveicarus/iverilog>, (accessed March 12, 2024).
- [16] F. Foundation, “cocotb Home Page,” <https://www.cocotb.org/>, (accessed March 12, 2024).
- [17] P. Deegan, “Wokwi Automated Testing, Tiny Tapeout,” https://tinytapeout.com/digital_design/wokwi_automated_testing/, (accessed March 12, 2024).
- [18] U. Shaked, “Locally Installing the Tools,” <https://docs.google.com/document/d/1aUZ1jthRpg4QURIIyzlOaPWImQzr-jBn3wZipVUPt4>, (accessed March 12, 2024).
- [19] M. Venn, “Updating Inputs and Outputs of a Design, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tintytapeout-02/blob/tt02/INFO.md#updating-inputs-and-outputs-of-a-specified-design>, (accessed March 12, 2024).
- [20] J. Harder, “Tiny Tapeout Scan, GitHub Repository,” https://github.com/jix/tintytapeout_scan, (accessed March 12, 2024).
- [21] M. Venn, “Verification Documentation, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tintytapeout-02/blob/tt02/VERIFICATION.md>, (accessed March 12, 2024).
- [22] YosysHQ, “SymbiYosys (SBY), YosysHQ Repository,” <https://github.com/YosysHQ/sby>, (accessed March 12, 2024).
- [23] M. Bailey, “CVC, Verification Documentation, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tintytapeout-02/blob/tt02/VERIFICATION.md#cvc>, (accessed March 12, 2024).
- [24] M. Venn, “Tiny Tapeout 3.5, GitHub Repository,” <https://github.com/TinyTapeout/tintytapeout-03p5>, (accessed March 12, 2024).
- [25] Efabless, “Caravel Openframe Project, GitHub Repository,” https://github.com/efabless/caravel_openframe_project, (accessed March 12, 2024).
- [26] H. Pretl, “TT05 Analog Test, GitHub Repository,” <https://github.com/iic-jku/tt05-analog-test>, (accessed March 12, 2024).
- [27] OSHWA, “Open Source Hardware Certification, Tiny Tapeout,” <https://certification.oshwa.org/es000040.html>, (accessed March 12, 2024).
- [28] D. P. George, “MicroPython Home Page,” <https://micropython.org/>, (accessed March 12, 2024).
- [29] M. Venn, “Demo Firmware Test Script, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tt3p5-demo-fw/blob/main/tt3p5-test/test.py#L119>, (accessed March 12, 2024).
- [30] P. Deegan, “TT04+ Demoboard PCB, Tiny Tapeout Repository,” <https://github.com/TinyTapeout/tt-demo-pcb>, (accessed March 12, 2024).
- [31] L. Moser, “Pinouts Specifications, Tiny Tapeout,” <https://tinytapeout.com/specs/pinouts/>, (accessed March 12, 2024).
- [32] U. Shaked, “Awesome Tiny Tapeout PMODs, GitHub Repository,” <https://github.com/TinyTapeout/awesome-tintytapeout-pmods>, (accessed March 12, 2024).
- [33] M. Venn, “TT02 Silicon is Alive!, Zero to ASIC Course Blog,” <https://www.zerotoasiccourse.com/post/tt02-silicon-is-alive/>, (accessed March 12, 2024).
- [34] C. ‘whitequark’, “Amaranth Documentation,” <https://amaranth-lang.org/docs/amaranth/latest/>, (accessed March 12, 2024).
- [35] ChipsAlliance, “Chisel Home Page,” <https://www.chisel-lang.org/>, (accessed March 12, 2024).
- [36] J. Gray, “S4GA Super Slow Serial SRAM FPGA, Tiny Tapeout 2,” <https://tinytapeout.com/runs/tt02/006/>, (accessed March 12, 2024).
- [37] H. Pretl, “Synthesizable Digital Temperature Sensor, Tiny Tapeout 3,” <https://tinytapeout.com/runs/tt03/047/>, (accessed March 12, 2024).
- [38] H. T. Fab, “Microtapeout of SKY130 Cells, Tiny Tapeout 3,” <https://tinytapeout.com/runs/tt03/047/>, (accessed March 12, 2024).
- [39] J. Kral and O. Kolar, “FM Transmitter, Tiny Tapeout 4,” <https://tinytapeout.com/runs/tt03/047/>, (accessed March 12, 2024).
- [40] D. Miles, “USB Device, Tiny Tapeout 4,” <https://tinytapeout.com/runs/tt03/047/>, (accessed March 12, 2024).
- [41] H. Dabul, “KianV uLinux System on Chip, Tiny Tapeout 5,” <https://tinytapeout.com/runs/tt05/654/>, (accessed March 12, 2024).