ASIC comes to Demoscene

Making demo for Tiny Tapeout competition on 130 nm

ReJ^Nesnausk^TBL aka Renaldas Zioma

160 x 225 microns
130 nm ASIC

Tiny Tapeout 08 Competition rules


DROP

clock in

nothing else!

out

VGA 60Hz / audio 1-bit

Logic gates - ??

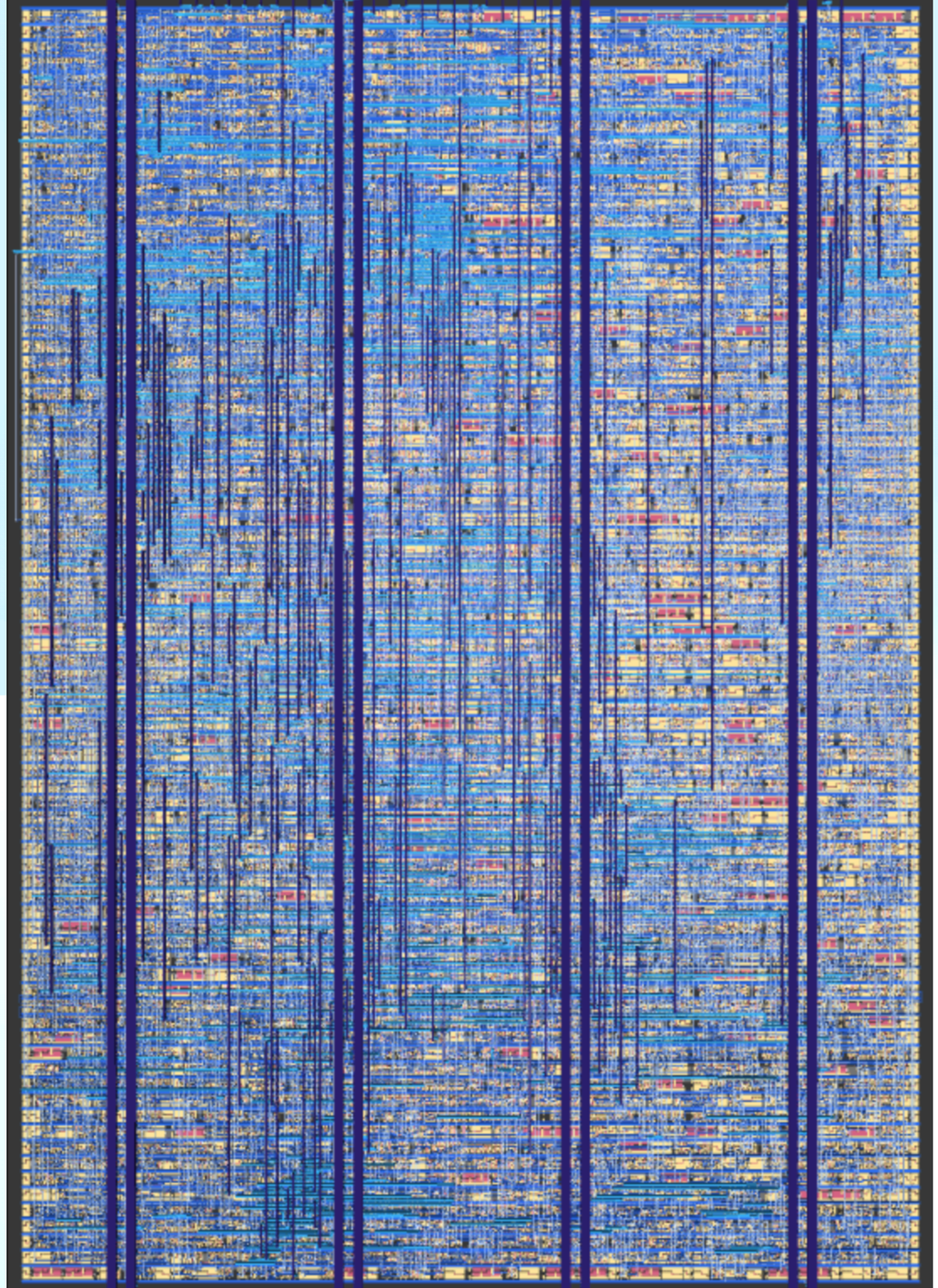
Flip-flop bits - ??

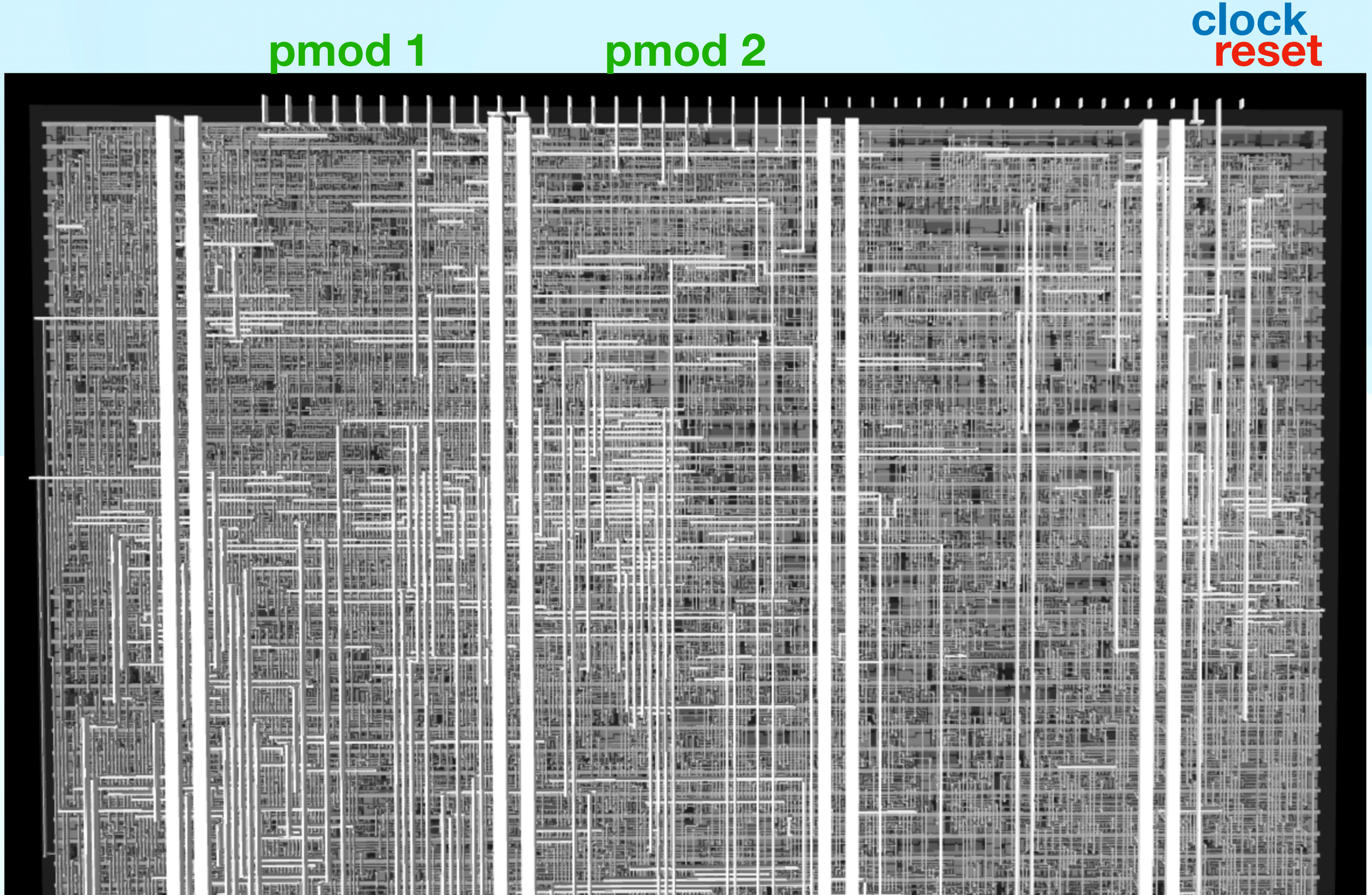
Lines of code - ??

Logic gates - 3456

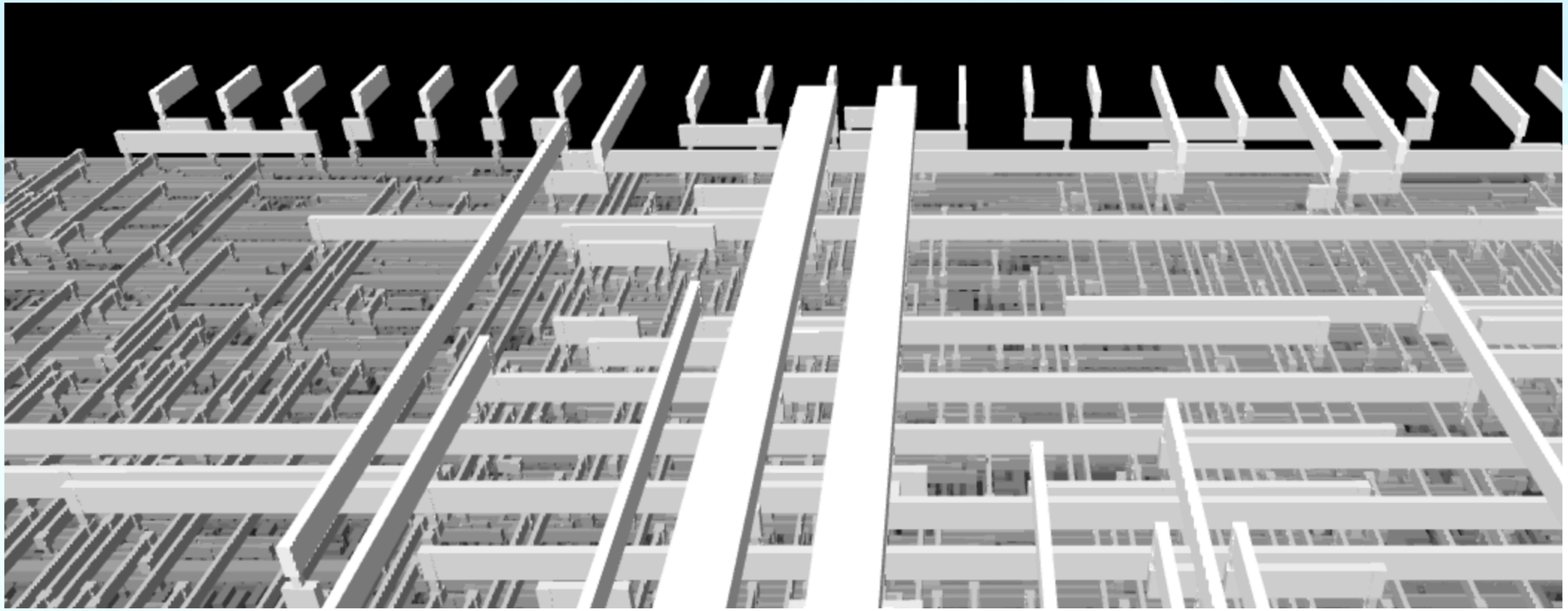
Flip-flop bits - 137

Lines of code - 250




pmod 1
pmod 2
clock
reset

VIDEO AUDIO



The most **surprising** learning?

Easier than coding graphics on existing **constrained*** platforms!

* 8-bit / 16-bit home computers, MCUs or very old consoles

What made this **possible**?

Quick **iteration** times

Open-source tools

Tiny Tapeout **community**

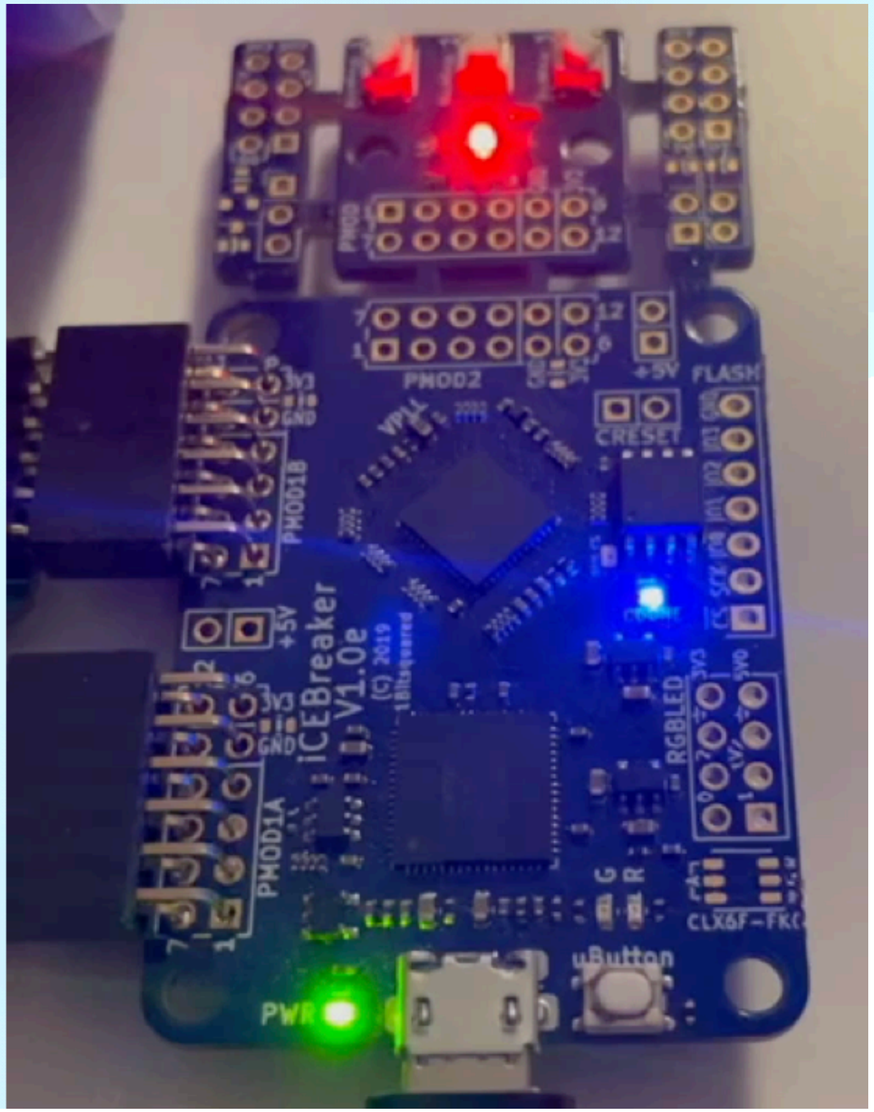
What made this possible?

VGA Playground
Verilog in the browser

Yosys + iCEBreaker FPGA

Tiny Tapeout
easy&quick, but exposes enough for tweaks

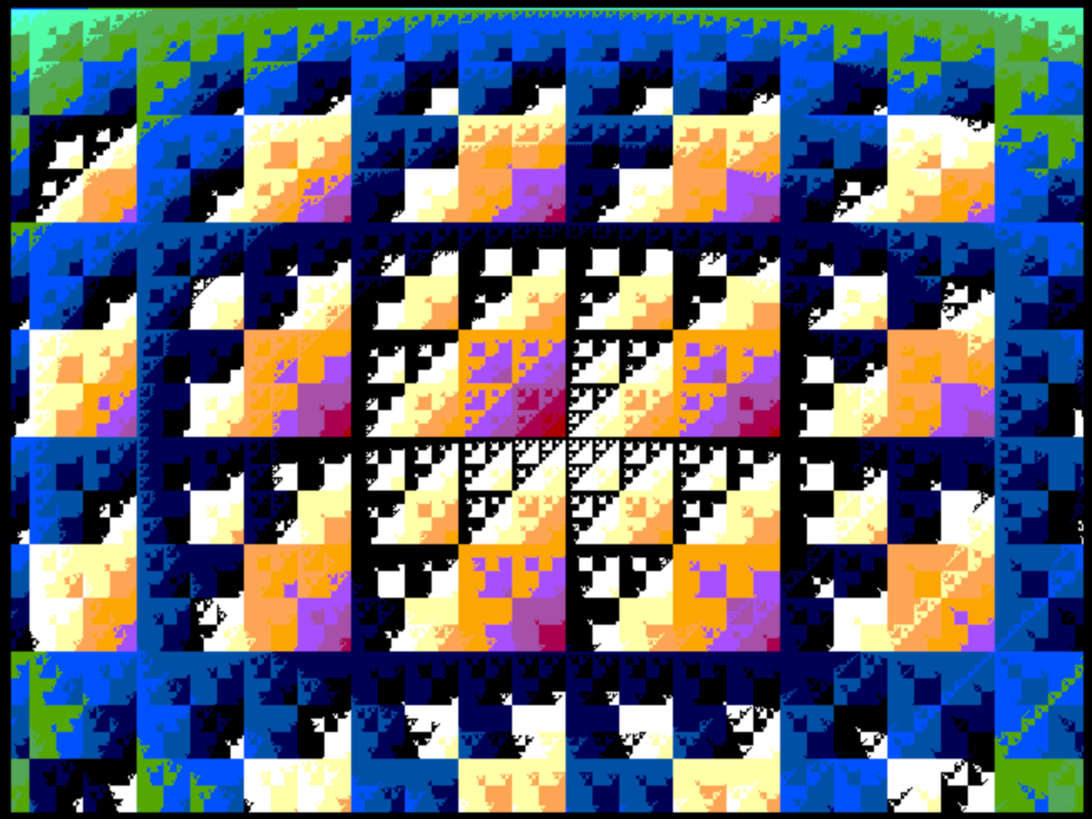
OpenLane 2



Tiny Tapeout VGA Playground Presets: Stripes | Balls | Logo | Conway | Checkers | "Drop" demo

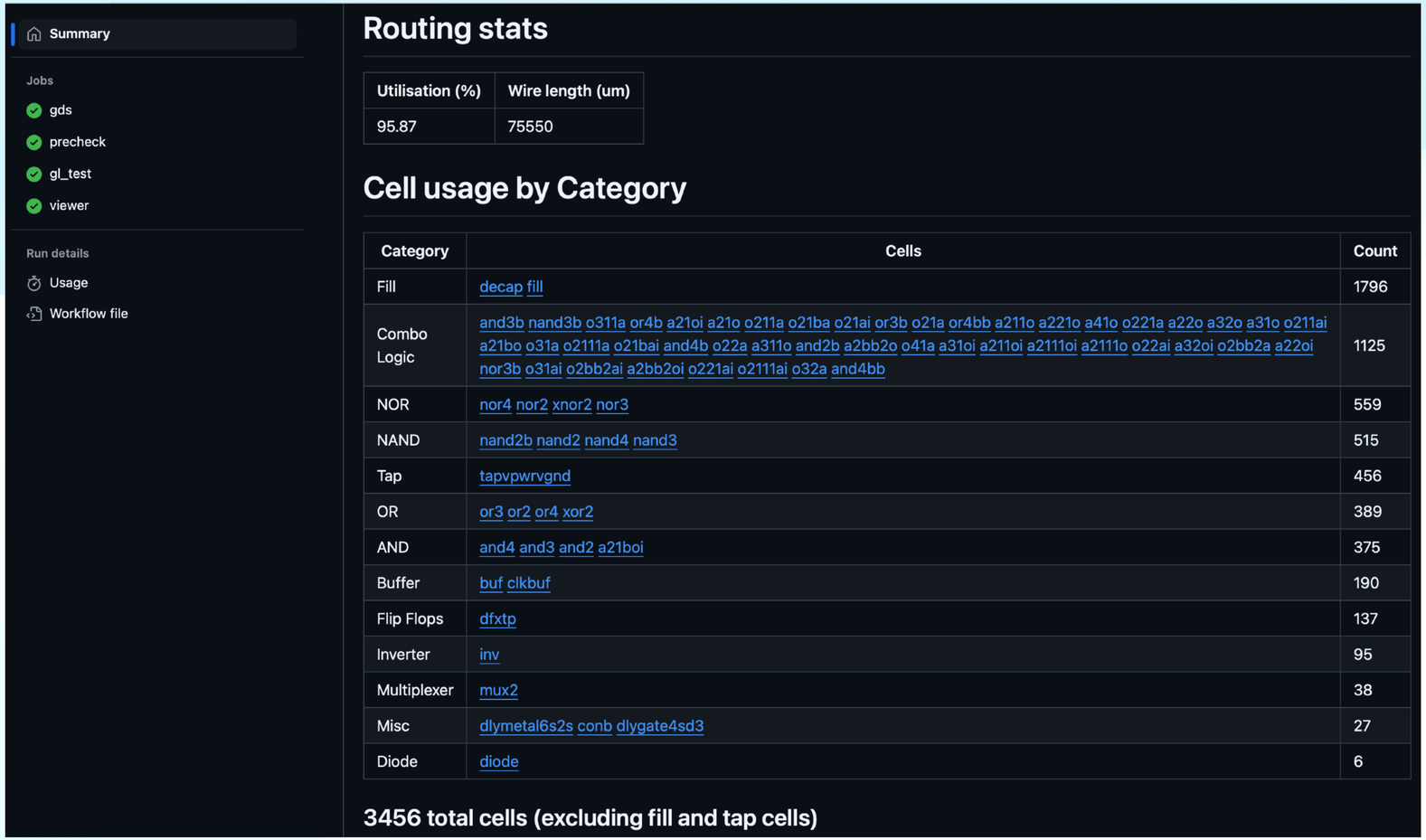
```
/*
 * "Drop" ASIC audio/visual demo. No CPU, no GPU, no RAM!
 * Racing the beam, straight to VGA 640x480@60Hz.
 * 3456 logic gates, 160 x 220 microns silicon area.
 * Entry to Tiny Tapeout Demoscene 2024 competition:
 *   https://tinytapeout.com/competitions/demoscene/
 *
 * Full version: https://github.com/rejunity/tt08-vga-drop
 * VGA video recording: https://youtu.be/jJBU0J2ceMM
 * Live recording from FPGA: https://youtu.be/rCupc2soGqo
 *
 * Copyright (c) 2024 Renaldas Zioma, Erik Hemming and Matthias Kampa
 * Code is based on the VGA examples by Uri Shaked
 * Inspired by "Memories" 256b demo by Desire
 * SPDX-License-Identifier: Apache-2.0
 */

`default_nettype none

module tt_um_vga_example(
  input  wire [7:0] ui_in,    // Dedicated inputs
  output wire [7:0] uo_out,   // Dedicated outputs
  input  wire [7:0] uio_in,   // IOs: Input path
  output wire [7:0] uio_out,  // IOs: Output path
  output wire [7:0] uio_oe,   // IOs: Enable path (active high: 0=input, 1=output)
  input  wire       ena,      // always 1 when the design is powered, so you can ignore it
  input  wire       clk,      // clock
  input  wire       rst_n     // reset_n - low to reset
);

  // VGA signals
  wire hsync;
  wire vsync;
  wire [1:0] R;
  wire [1:0] G;
  wire [1:0] B;

  // TinyVGA PMOD https://github.com/mole99/tiny-vga
  assign uo_out = {hsync, B[0], G[0], R[0], vsync, B[1], G[1], R[1]};

  // Unused outputs assigned to 0.
  assign uio_out = 0;
  assign uio_oe  = 0;
```

ui_in: 0 1 2 3 4 5 6 7 FPS: 23



Tiny Tapeout

easy & quick, but exposes enough for **tweaks**



Summary

Jobs

gds

precheck

gl_test

viewer

Run details

Usage

Workflow file

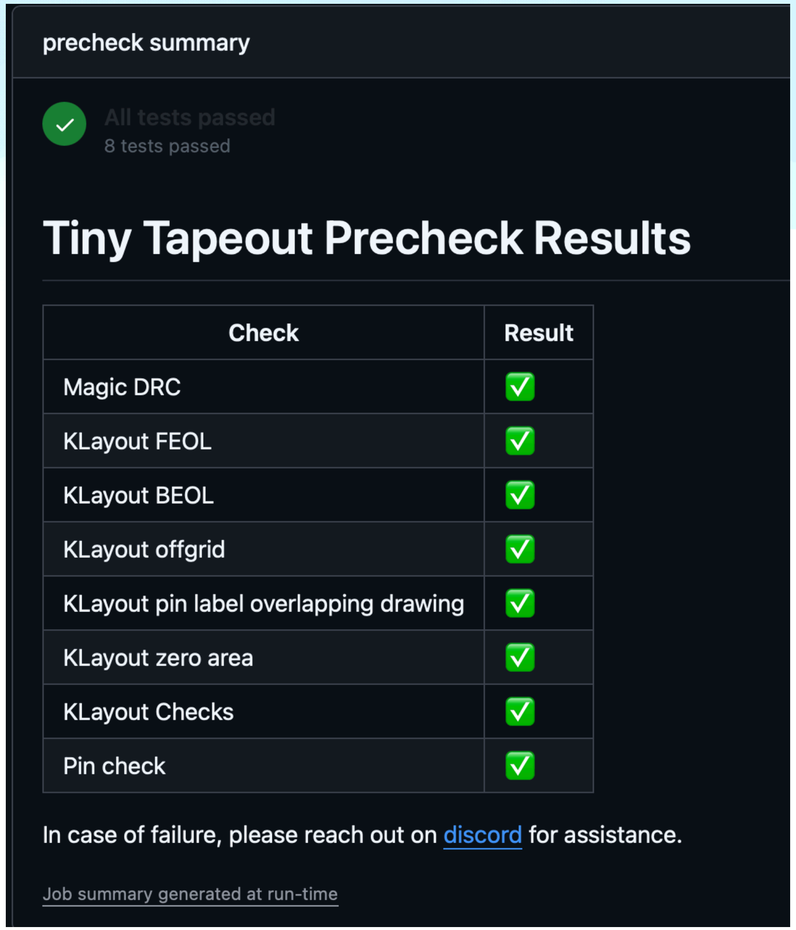
Routing stats

Utilisation (%)	Wire length (um)
95.87	75550

Cell usage by Category

Category	Cells	Count
Fill	decap fill	1796
Combo Logic	and3b nand3b o311a or4b a21oi a21o o211a o21ba o21ai or3b o21a or4bb a211o a221o a41o o221a a22o a32o a31o o211ai a21bo o31a o2111a o21bai and4b o22a a311o and2b a2bb2o o41a a31oi a211oi a2111oi a2111o o22ai a32oi o2bb2a a22oi nor3b o31ai o2bb2ai a2bb2oi o221ai o2111ai o32a and4bb	1125
NOR	nor4 nor2 xnor2 nor3	559
NAND	nand2b nand2 nand4 nand3	515
Tap	tapvpwrvgnd	456
OR	or3 or2 or4 xor2	389
AND	and4 and3 and2 a21boi	375
Buffer	buf clkbuf	190
Flip Flops	dfxtp	137
Inverter	inv	95
Multiplexer	mux2	38
Misc	dlymetal6s2s conb dlygate4sd3	27
Diode	diode	6

3456 total cells (excluding fill and tap cells)



precheck summary

All tests passed

8 tests passed

Tiny Tapeout Precheck Results

Check	Result
Magic DRC	✅
KLayout FEOL	✅
KLayout BEOL	✅
KLayout offgrid	✅
KLayout pin label overlapping drawing	✅
KLayout zero area	✅
KLayout Checks	✅
Pin check	✅

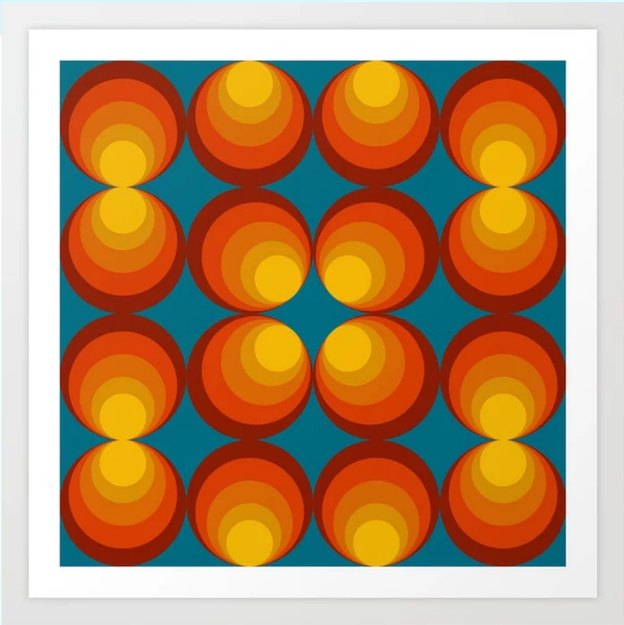
In case of failure, please reach out on discord for assistance.

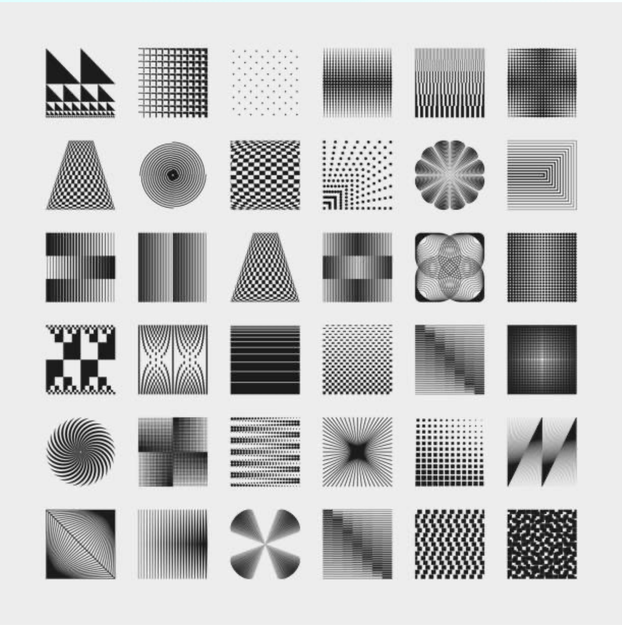
Job summary generated at run-time

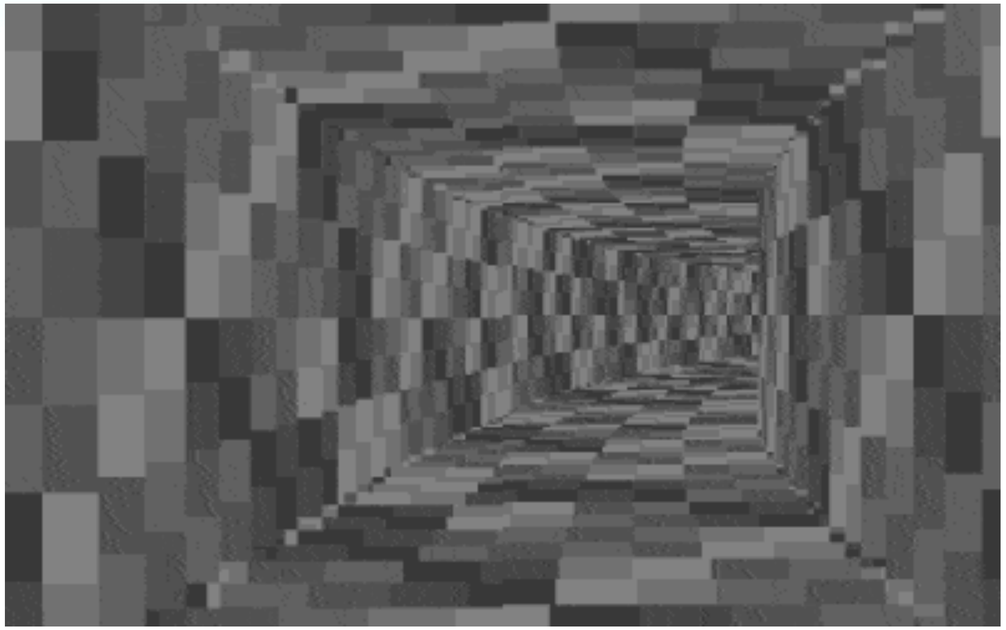
Inspirations

Memories by Desire - 9 effects in **256** byte demo for PC

1970’s print design



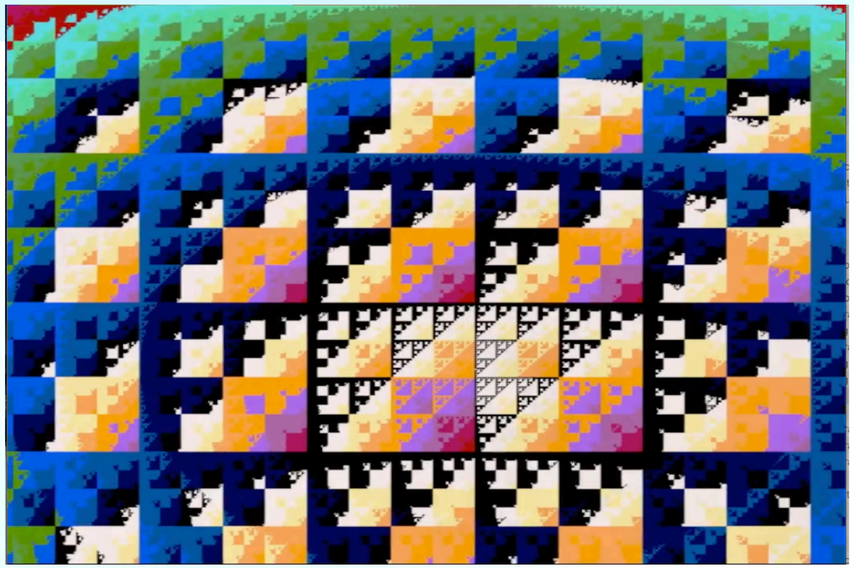


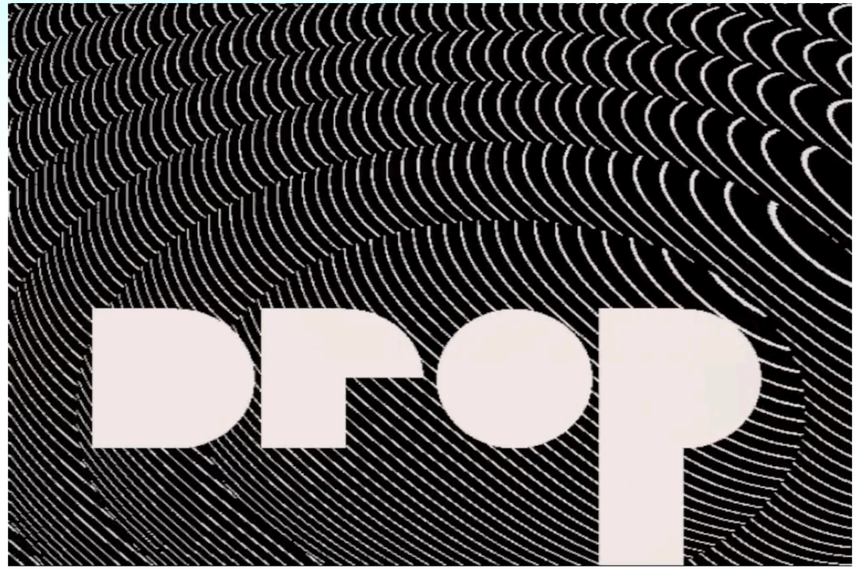


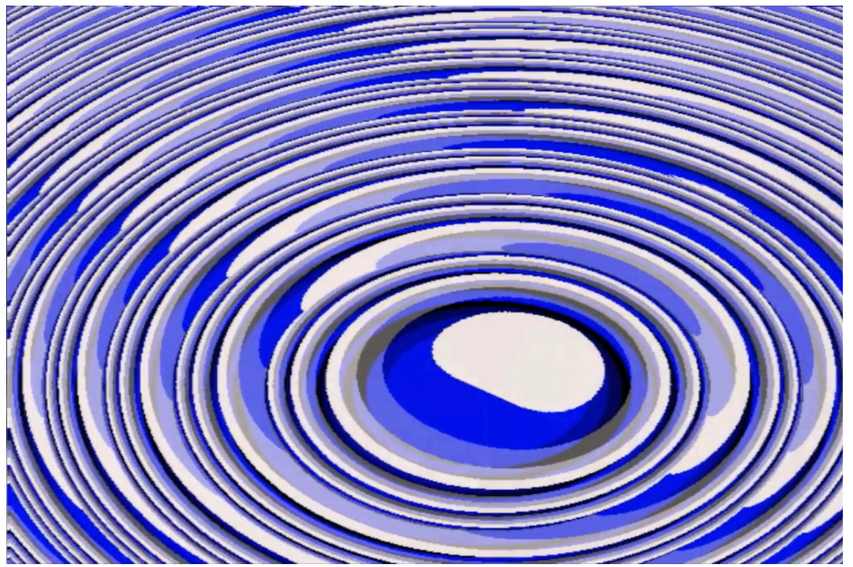
Unique look

Emphasis on a limited color palette - no dithering on purpose

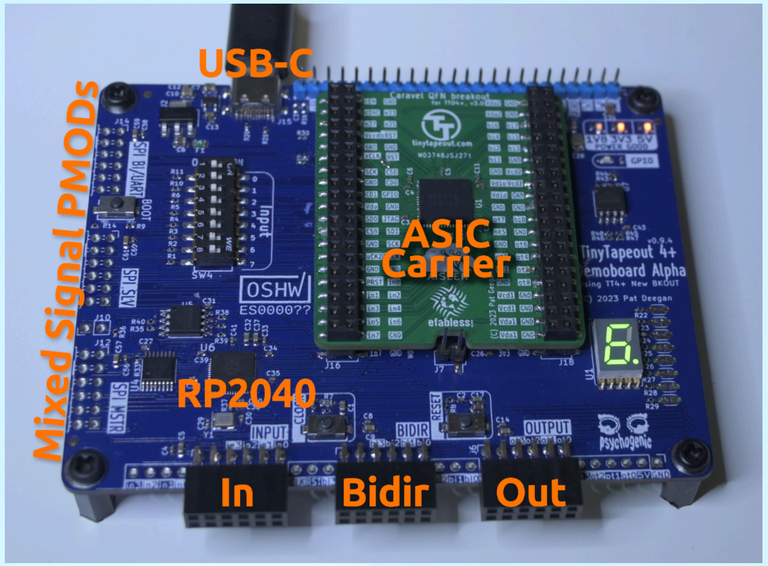
Visuals not seen in 8-bit / 16-bit demos

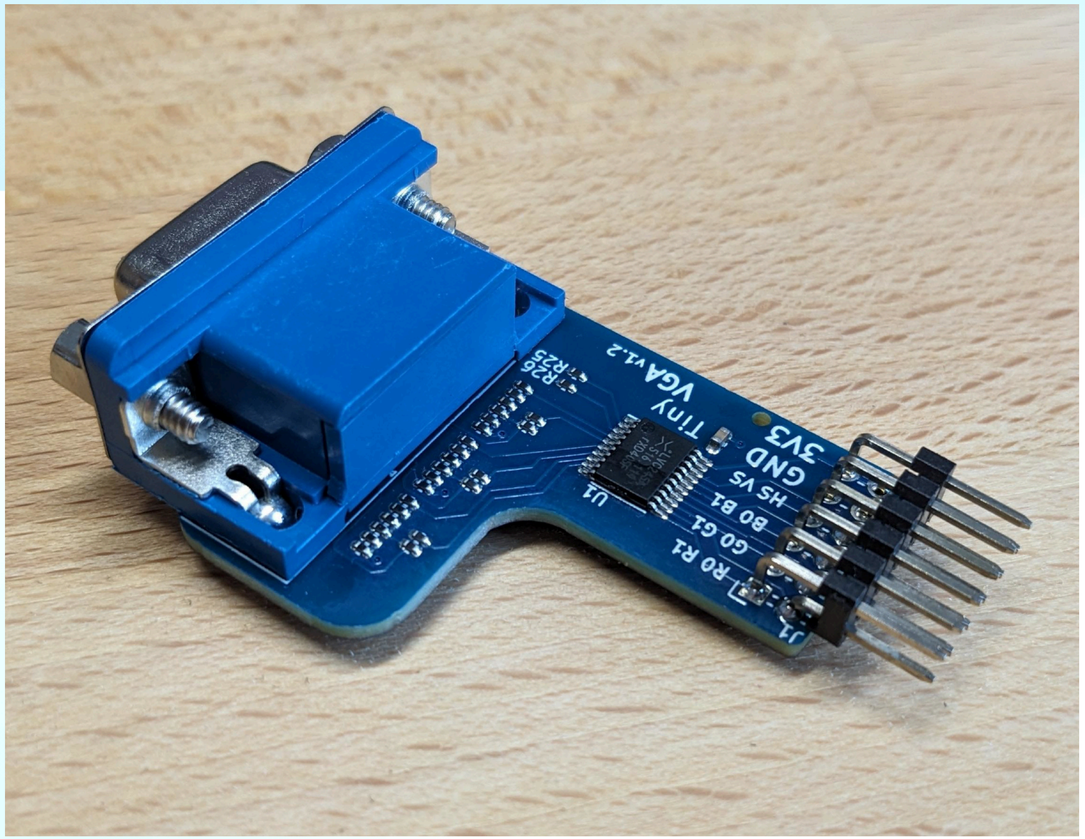


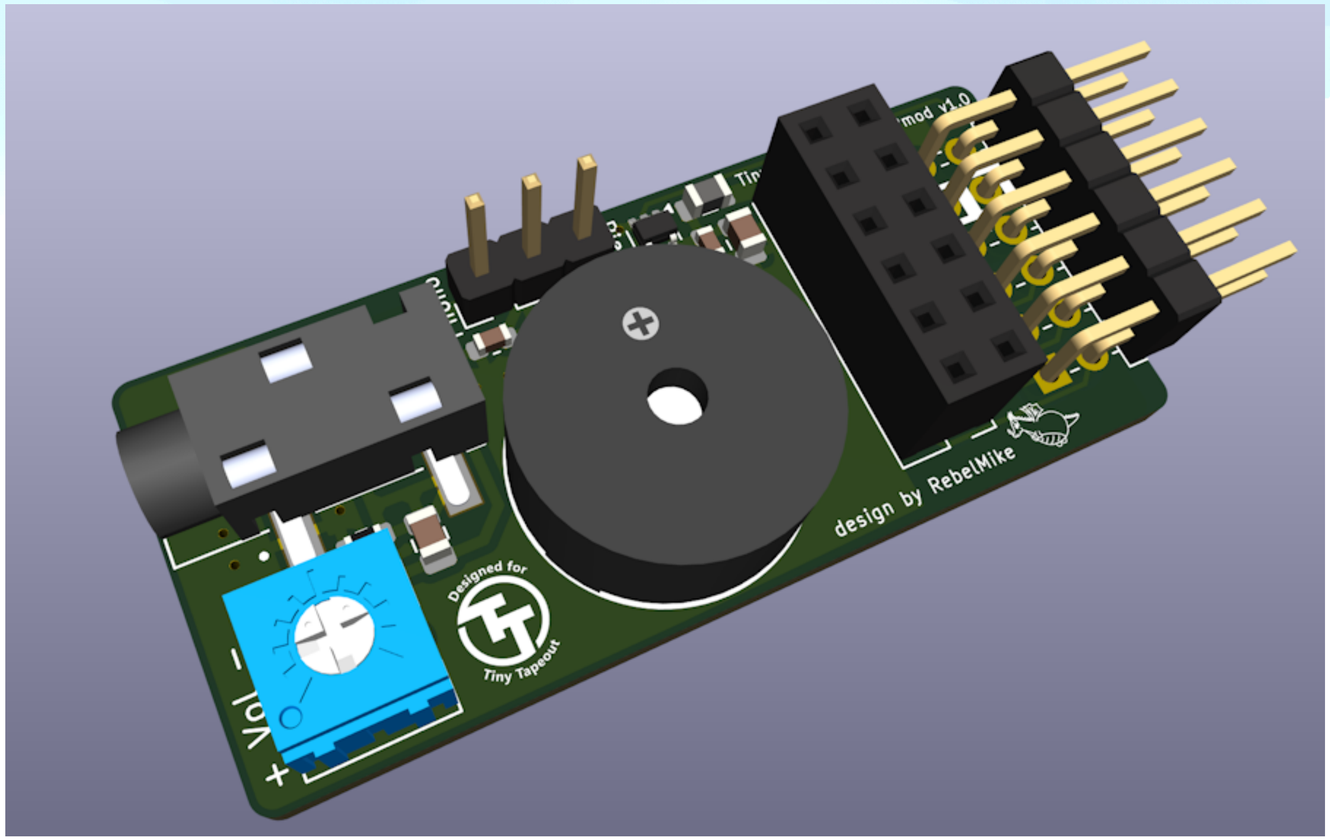




Tiny PMODs


USB-C
Mixed Signal PMODs
ASIC
Carrier
RP2040
In
Bidir
Out


Tiny VGA v1.2
3V3
GND
HS VS
B0 B1
G0 G1
R0 R1
J1
U1
R25
R25


Vol
Designed for
Tiny Tapeout
design by RebelMike

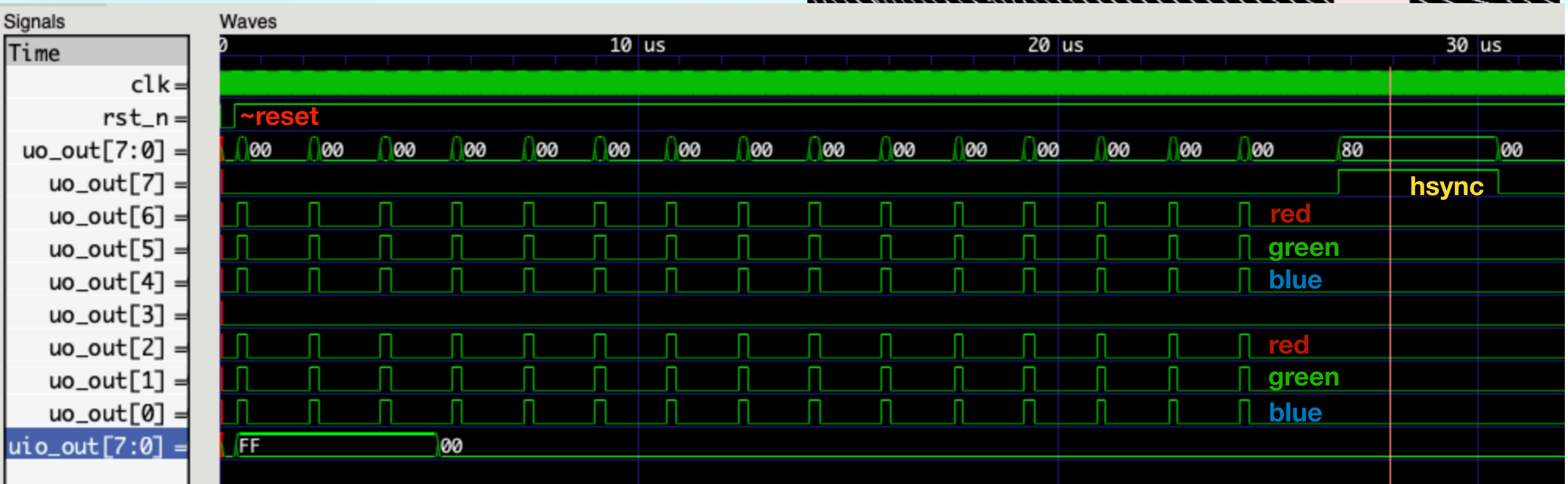
VGA video

- 640 x 480 pixels
- 60 Hz
- 25.125 MHz pixel clock
- 40 nanoseconds to generate a pixel
- 6 color bits per pixel

```
// VGA signals
wire hsync;
wire vsync;
wire [1:0] R;
wire [1:0] G;
wire [1:0] B;

// Generate VGA signal, x and y coordinates
wire [9:0] x;
wire [9:0] y;
wire video_active;
hvsync_generator hvsync_gen(
  .clk(clk),
  .reset(~rst_n),
  .hsync(hsync),
  .vsync(vsync),
  .display_on(video_active),
  .hpos(x),
  .vpos(y)
);

// TinyVGA PMOD https://github.com/mole99/tiny-vga
assign uo_out = {hsync, B[0], G[0], R[0], vsync, B[1], G[1], R[1]};
```

VGA Video


DROP


Signals
Waves
Time
clk
rst_n
uo_out[7:0]
uo_out[7]
uo_out[6]
uo_out[5]
uo_out[4]
uo_out[3]
uo_out[2]
uo_out[1]
uo_out[0]
uio_out[7:0]
10 us
20 us
30 us
~reset
00
80
FF
hsync
red
green
blue
red
green
blue

Racing the beam

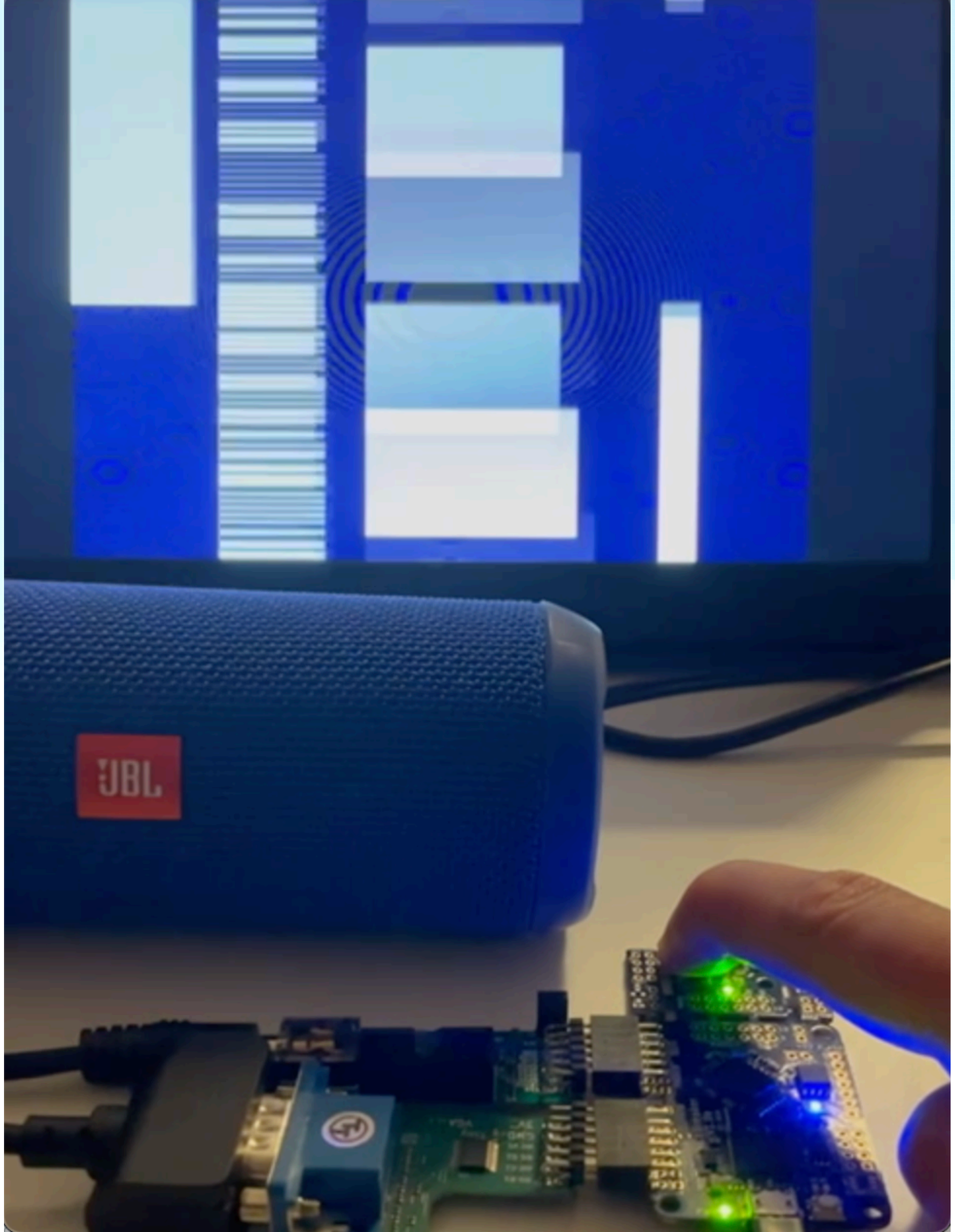
- No frame-buffer
- Hard deadline to generate a pixel - **40 nanoseconds**

- “Texture combiner” approach

```
wire [2:0] part = frame_counter[9-:3];
assign {R,G,B} =
  (~video_active) ? 6'b00_00_00 :
  (part == 0) ? { &out[5:3] | title ? 6'b11_11_11 : 6'b00_00_00 } :          // title + wakes
  (part == 1) ? { &out[5:2] * out[1-:2], &out[6:0] * out[1-:2], 2'b00 } :    // red/golden Sierpinsky
  (part == 3) ? { |out[7:6] ? {4'b11_00, dot[6:5]} : out[5:4] } :            // tunnel
  (part == 4) ? { &out[6:4] * 6'b11_00_00 | &out[6:3]*dot[7]*6'b00_00_10 } : // red wakes
  (part == 6) ? { out[7-:2], out[6-:2], out[5-:2] } :                        // multi-color Sierpinsky
  (part == 7) ? { |out[7:6] ? {4'b11_00, dot[6:5]} : out[5:4] } |
                { 6{title & (frame_counter[6:0] >= 96) }  } :                // title + tunnel,
                                                                             // the title is delayed by 1.5 beat
                { out[7-:2], out[7-:2], out[7-:2] } | {4'b0,~dot[6-:2]};
```

Audio

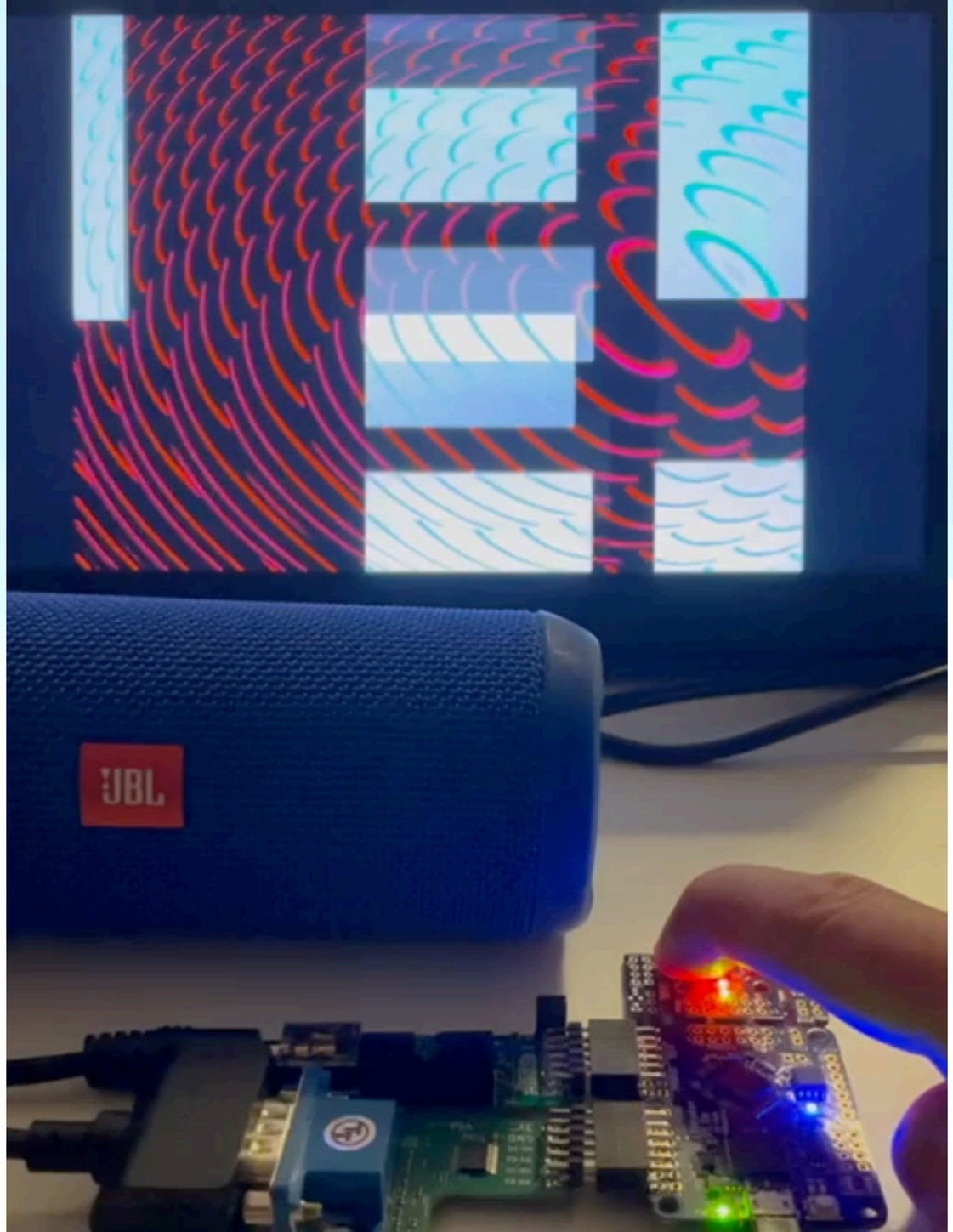
- 1-bit PWM
- 25 MHz clock



Audio

- **square** waves
- amplitude **envelope**
- 4 channels

  drum kick
  snare
  base
  lead



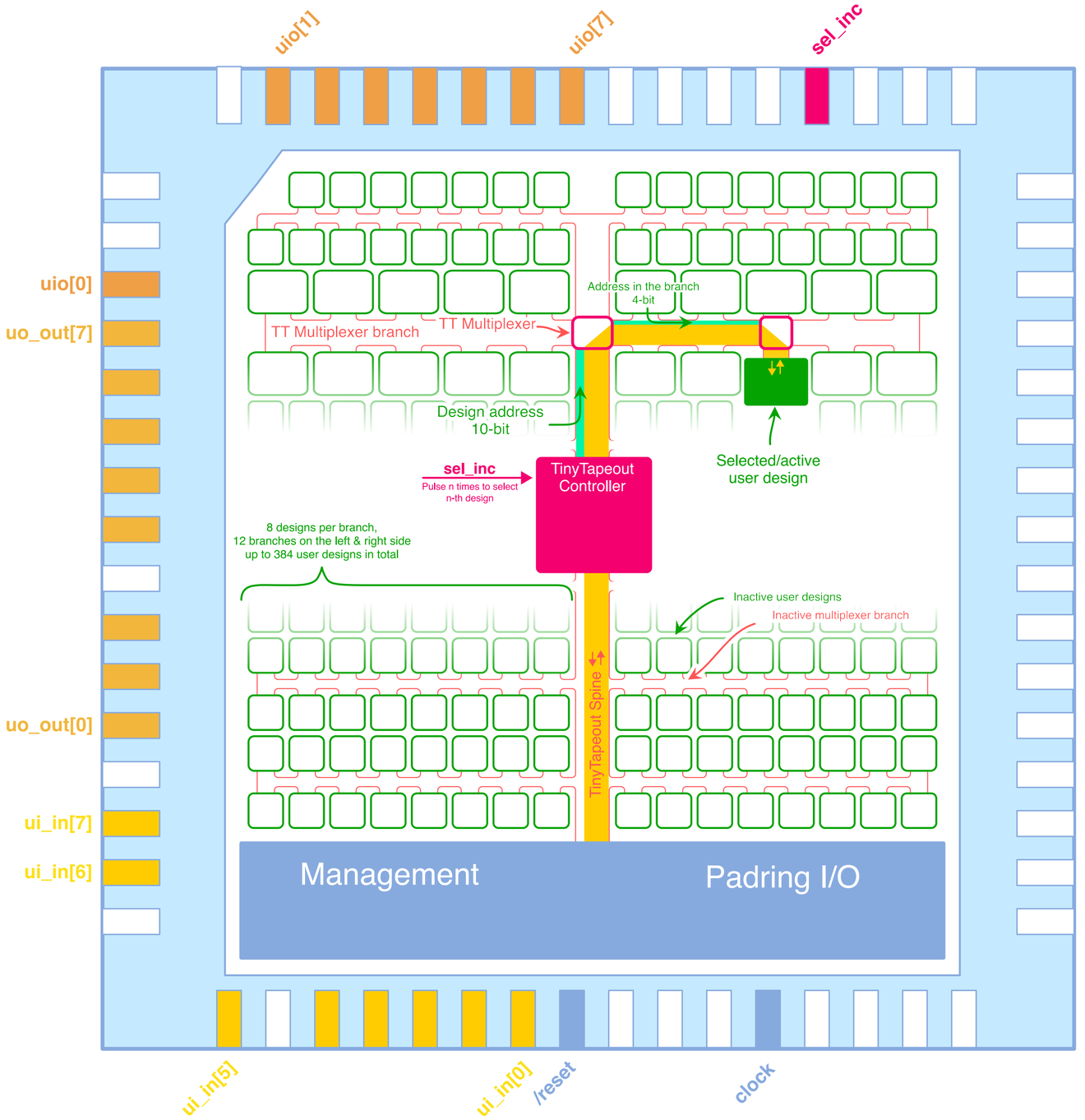
iCEBreaker FPGA utilisation

iCE40 UltraPlus 5K

```
Info: Device utilisation:
Info:            ICESTORM_LC:  1693/ 5280     32%
Info:           ICESTORM_RAM:     0/   30      0%
Info:                  SB_IO:    26/   96     27%
Info:                  SB_GB:     4/    8     50%
Info:           ICESTORM_PLL:     1/    1    100%
Info:            SB_WARMBOOT:     0/    1      0%
Info:           ICESTORM_DSP:     0/    8      0%
Info:         ICESTORM_HFOSC:     0/    1      0%
Info:         ICESTORM_LFOSC:     0/    1      0%
Info:                 SB_I2C:     0/    2      0%
Info:                 SB_SPI:     0/    2      0%
Info:                 IO_I3C:     0/    2      0%
Info:            SB_LEDDA_IP:     0/    1      0%
Info:            SB_RGBA_DRV:     0/    1      0%
Info:         ICESTORM_SPRAM:     0/    4      0%
```

Take Away

- It is ~*easy*~ to create your own ASIC
- Special purpose hardware might be **easier** than coding for constrained platform
- **Optimise** for iteration times
- I love **open source** + **demoscene!**


uio[1]
uio[7]
sel_inc
uio[0]
uo_out[7]
uo_out[0]
ui_in[7]
ui_in[6]
ui_in[5]
ui_in[0]
/reset
clock
Address in the branch
4-bit
TT Multiplexer
TT Multiplexer branch
Design address
10-bit
Selected/active
user design
sel_inc
Pulse n times to select
n-th design
TinyTapeout
Controller
8 designs per branch,
12 branches on the left & right side
up to 384 user designs in total
Inactive user designs
Inactive multiplexer branch
TinyTapeout Spine
Management
Padring I/O