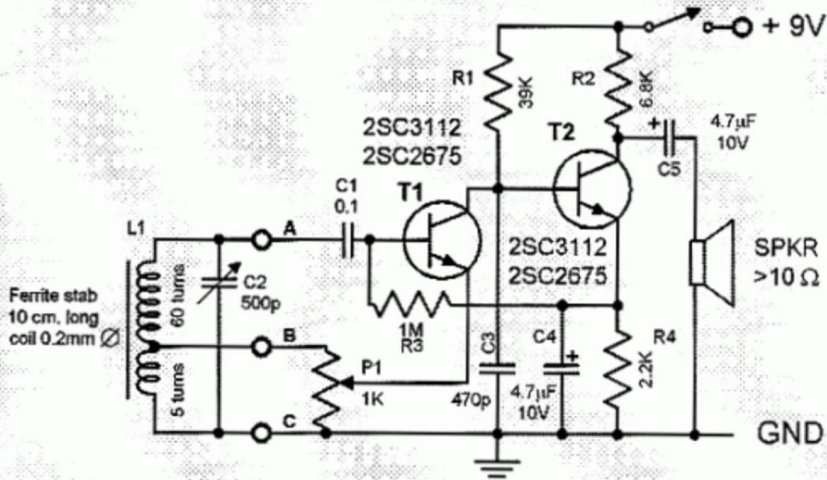


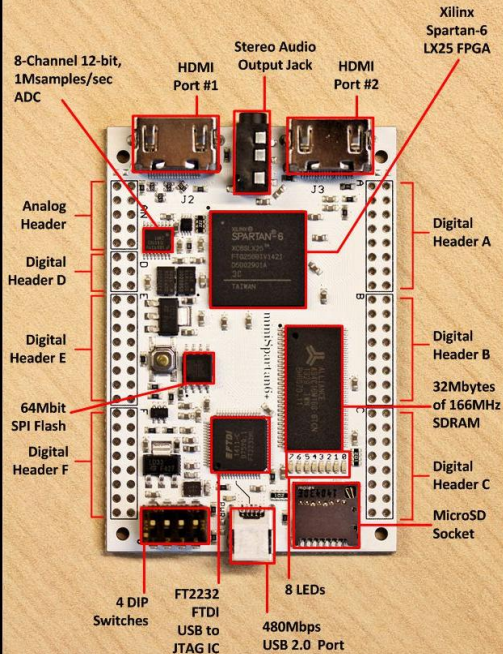
CuFlow - code not CAD

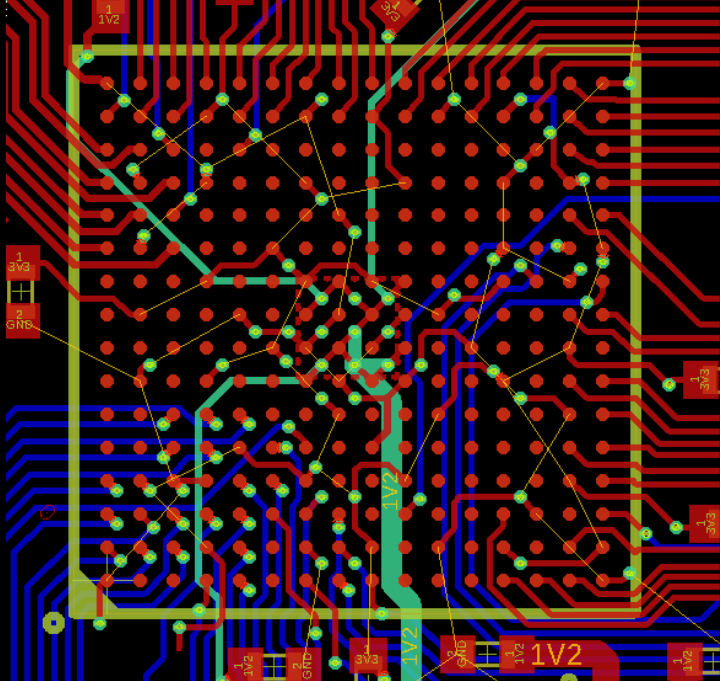
James Bowman - Excamera Labs

June 24, 2021

About the slides







ADC

GPIOA[15]	to	USART2_RXD2	USART2_RXD2_IN[15]
GPIOA[14]	to	USART2_RXD1	USART2_RXD1_IN[14]
GPIOA[13]	to	USART2_TXD1	USART2_TXD1_OUT[13]
GPIOA[12]	to	USART2_TXD2	USART2_TXD2_OUT[12]

DIP Switch

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2

micro SD Card

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2

LEDs

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

POWER & GROUND

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

JTAG

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2

FLASH

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2

GPIOs

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

Not Used

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

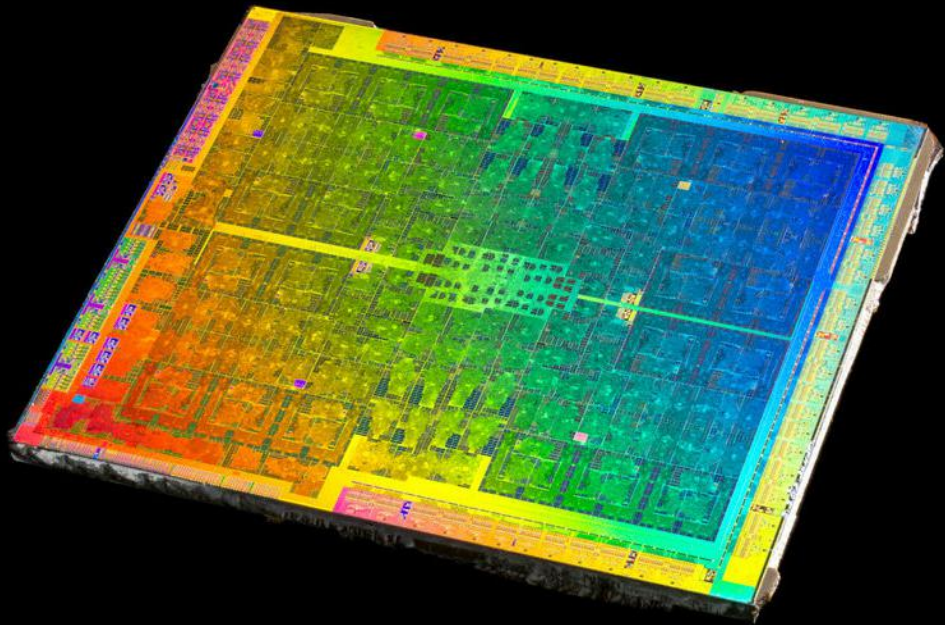
FTDI

GPIOA[15]	to	USART2_RXD2
GPIOA[14]	to	USART2_RXD1
GPIOA[13]	to	USART2_TXD1
GPIOA[12]	to	USART2_TXD2
GPIOA[11]	to	USART2_TXD3
GPIOA[10]	to	USART2_TXD4
GPIOA[9]	to	USART2_TXD5
GPIOA[8]	to	USART2_TXD6
GPIOA[7]	to	USART2_TXD7
GPIOA[6]	to	USART2_TXD8
GPIOA[5]	to	USART2_TXD9
GPIOA[4]	to	USART2_TXD10
GPIOA[3]	to	USART2_TXD11
GPIOA[2]	to	USART2_TXD12
GPIOA[1]	to	USART2_TXD13
GPIOA[0]	to	USART2_TXD14

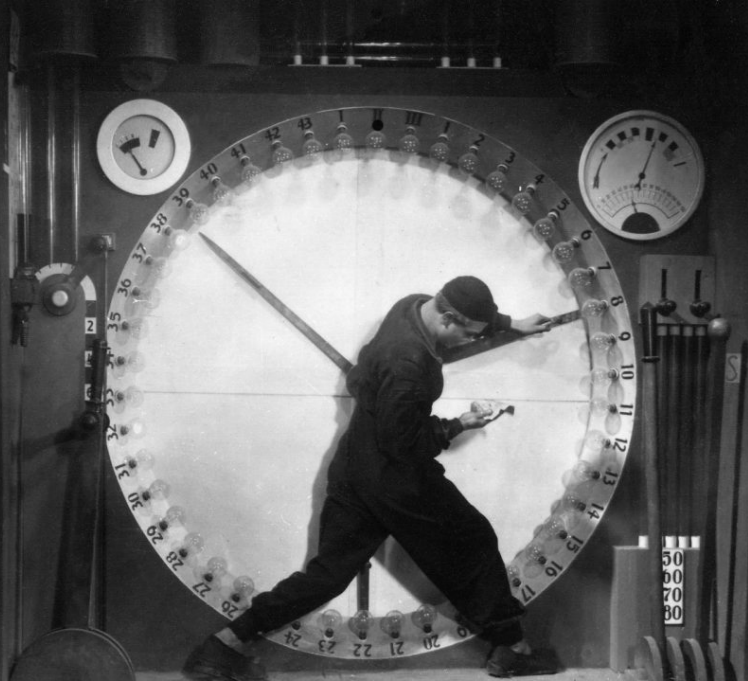
Copyright Scarab Hardware 2014
 With some parts from Saanlima Electronics
 Under
 Creative Commons Attribution Sharealike 4.0

- ▶ on paper
- ▶ in Excel!
- ▶ janky script

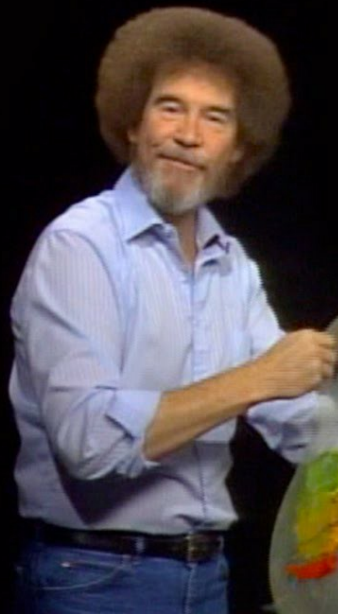
Schematics are
unsuitable for digital
circuits

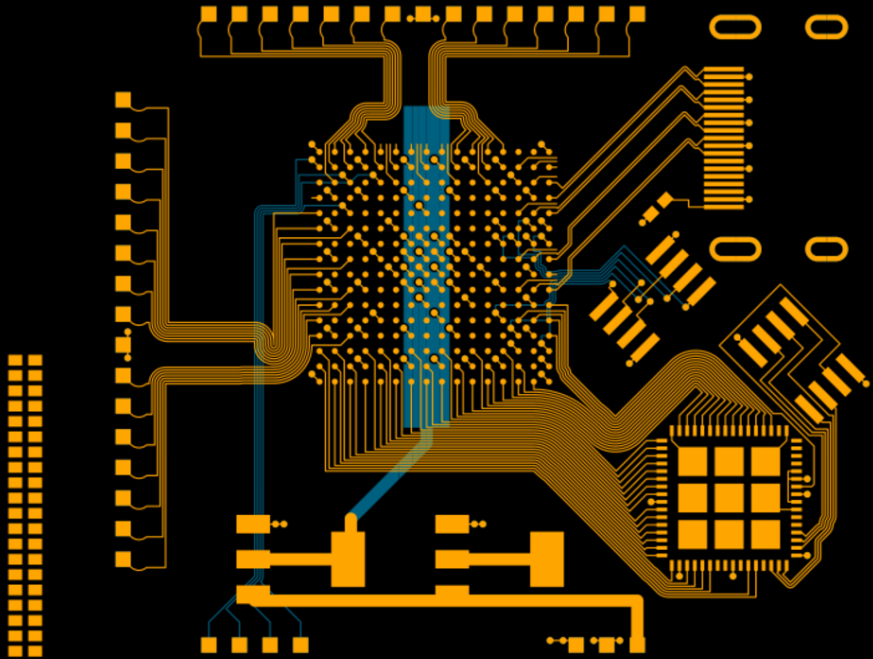


CAD tools are a lot
of work



What would be
better?

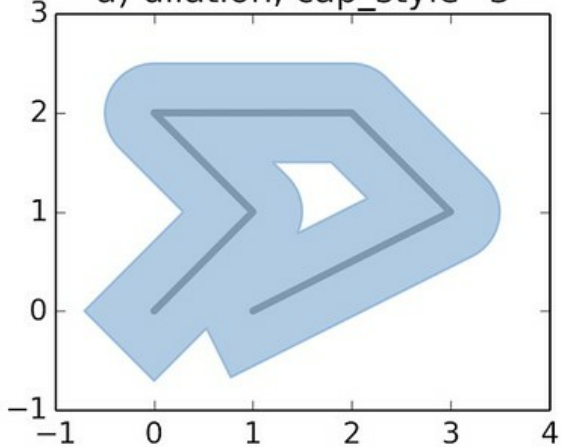




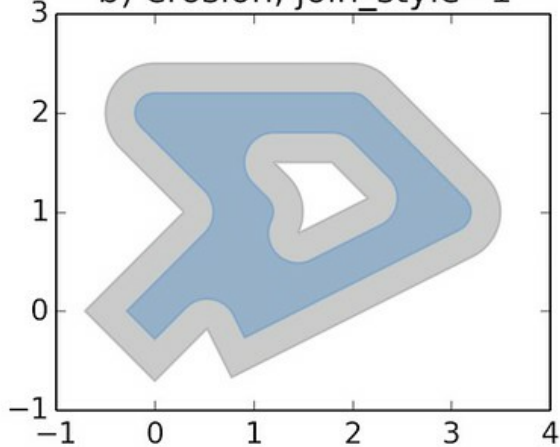
Python, shapely,
Gerbers

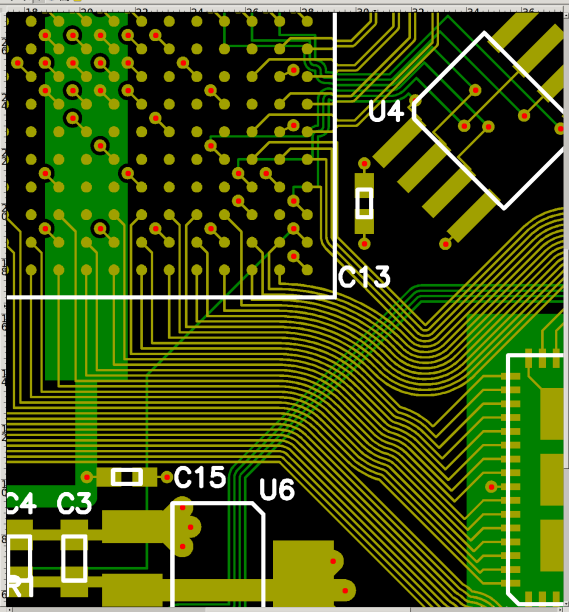


a) dilation, cap_style=3



b) erosion, join_style=1





```

osc = osc_6MHz(brd.DC((9.9, 30.7)).right(0))
clk = osc.escape()
fpga_ep0.goto(clk).wire()
brd.addnet(fpga_ep0, clk)

p_fl_f = cu.W25064J(brd.DC((35, 23)).left(45))
fl2_qsipi = p_fl_f.escape1()

def ldo(p):
    r = cu.S0T223(p)
    p.goxy(-2.3/2, -5.2).w("r 180")
    cu.C0603(p, val = '4.7 uF', source = {'LCSC': 'C19666'})
    p.forward(2)
    pa = cu.C0603(p, val = '22 uF', source = {'LCSC': 'C159801'}).pads
    pa[0].w("l 90 f 3").wire(width = 0.4)
    pa[1].w("r 90 f 3").wire(width = 0.4)
    return (r, r.escape())

(ldo12, (t, _)) = ldo(brd.DC((12, 6)).right(90))
t.outside().fan(1.0, 'GL3')
ldo12.mfr = 'LM1117S-1.2'
ldo12.source = {'LCSC': 'C126025'}

p = brd.DC((25, 6)).right(90)
(ldo33, (_, vcc)) = ldo(p)
ldo33.mfr = 'ZLD011170G33TA'
ldo33.source = {'LCSC': 'C326523'}

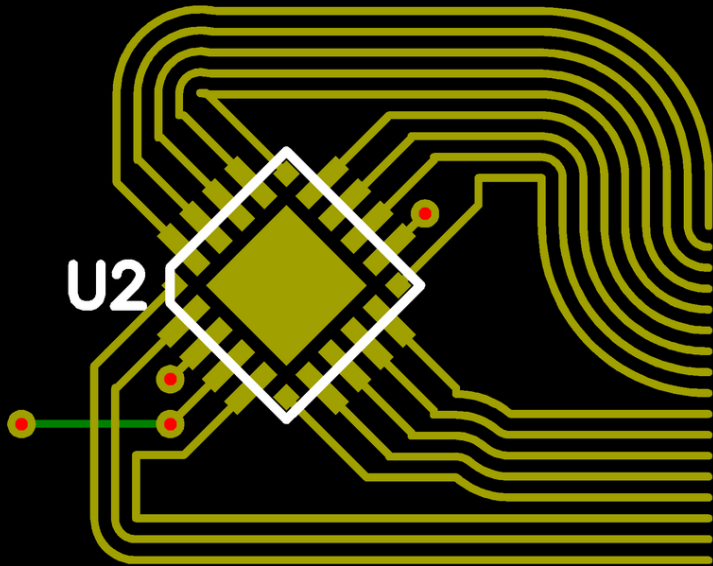
vcc.fan(1.5, 'GL3')

# Connect the LVDS pairs
for b in range(4):
    fpga_lvds[b].w("f 0.5 l 45").forward(b + 1).w("l 45 f 3").wire()
    hdmi_lvds[b].w("f 0.5").wire()
# [h.meet(f) for (h, f) in zip(hdmi_lvds, fpga_lvds)]
[f.meet2(h) for (h, f) in zip(hdmi_lvds, fpga_lvds)]

fpga_p0.w("f 2.5").meet(p0)
fpga_p1.w("f 2.5").meet(p1)

fpga_p23.right(90).wire()
(fpga_p2, fpga_p3) = fpga_p23.split(8)
# fpga_p2.w("r 45 f 1 l 45").wire()
fpga_p3.w("f 2").wire()
    
```

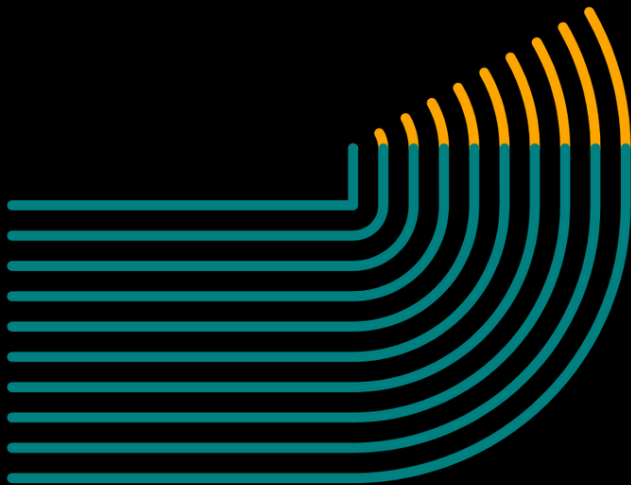
Escape Rivers

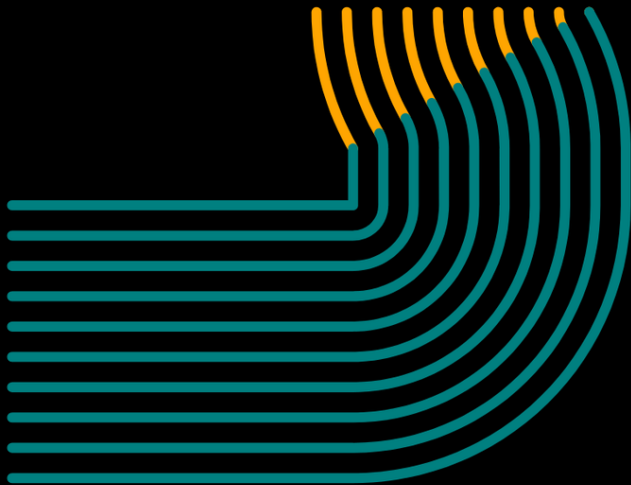


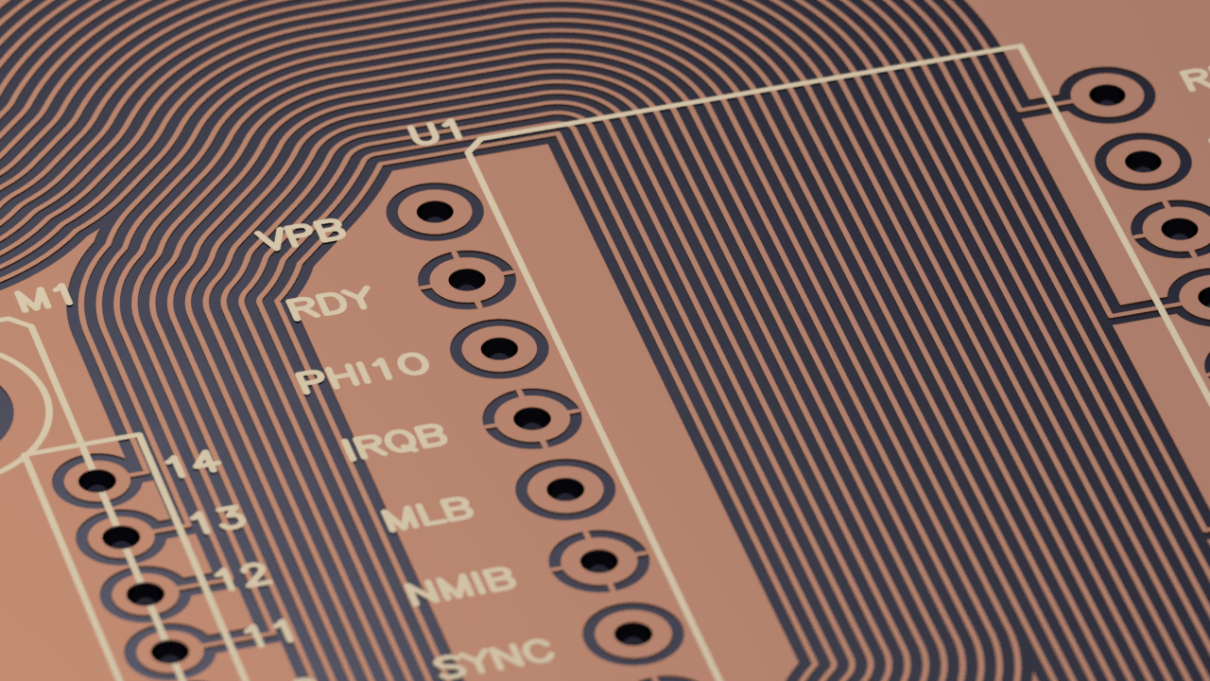












U1

VPB

RDY

PHI10

IRQB

MLB

NMIB

SYNC

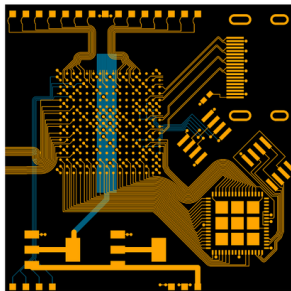
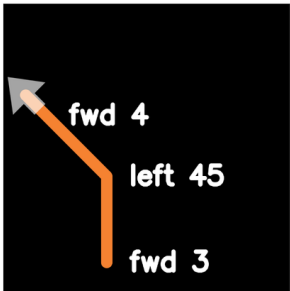
M1

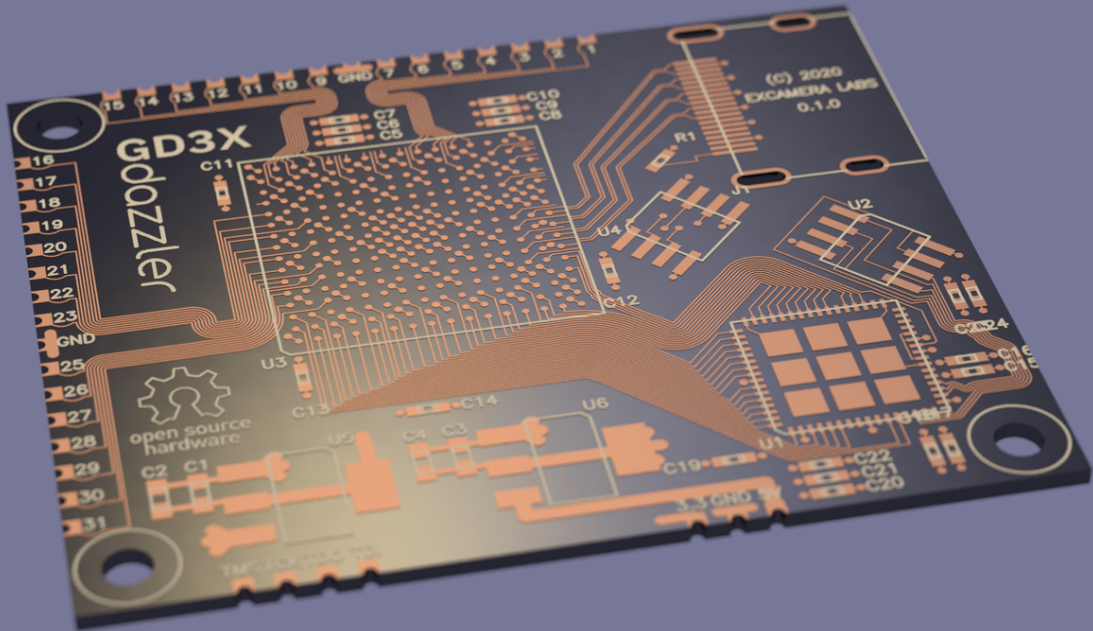
14

13

12

11





You have no
autorouter,
you loser

