ChipFlow

Automotive case study

Using Open Source to take 11 chips down to 1
In 3 months!

From no requirements to tapeout in 3 months?

Reduce Software Load?

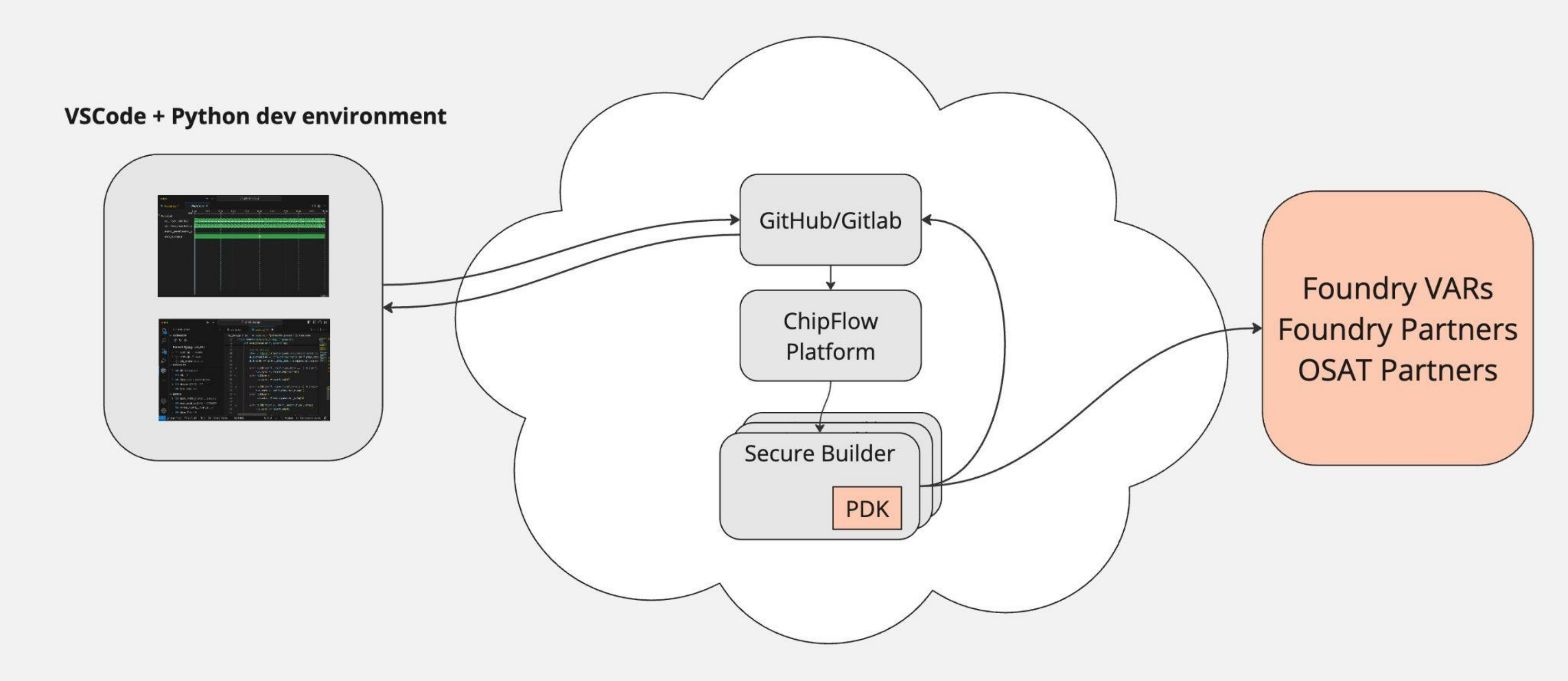
Let’s See!

From no requirements to tapeout in 3 months?

Reduce Software Load?

Let’s See!

Does our Platform help?



VSCode + Python dev environment

GitHub/Gitlab

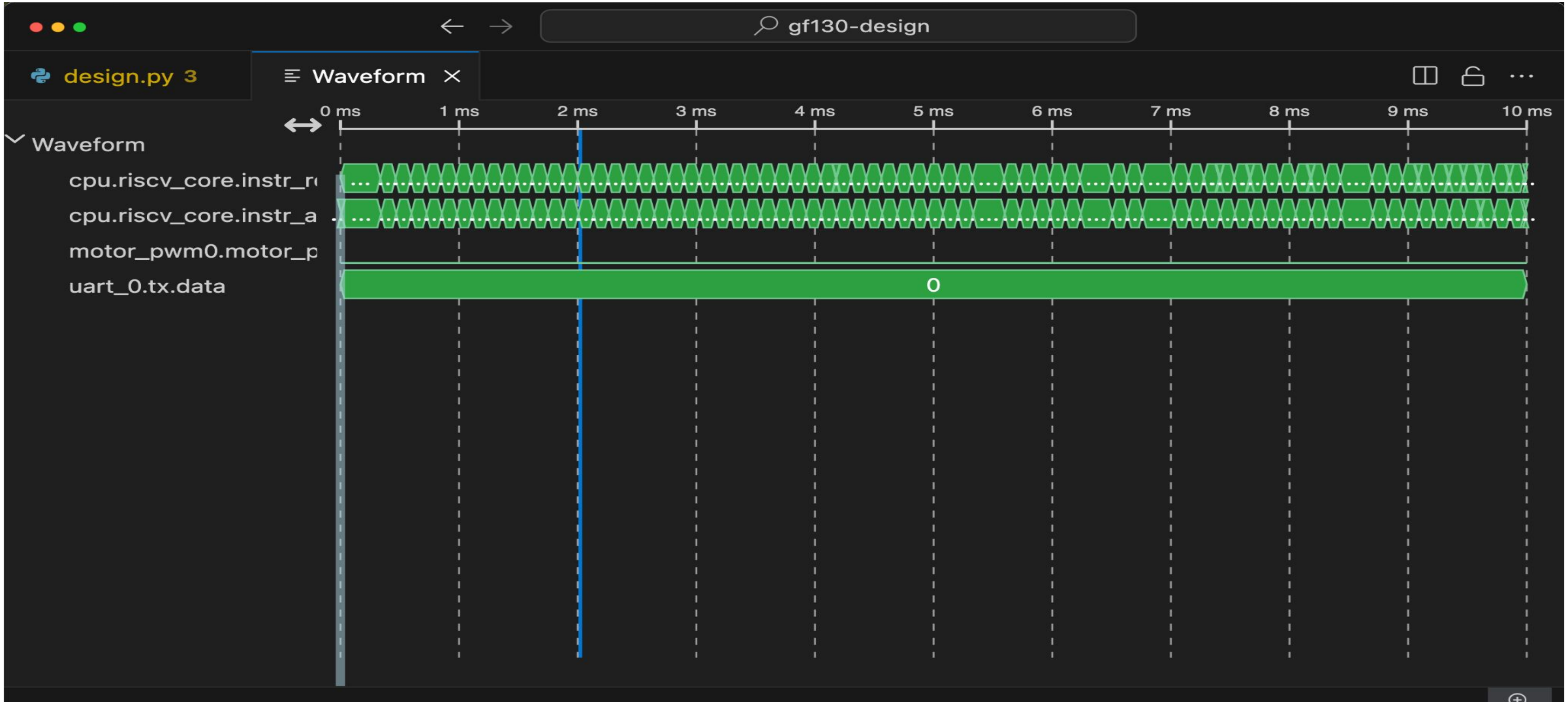
ChipFlow Platform

Secure Builder

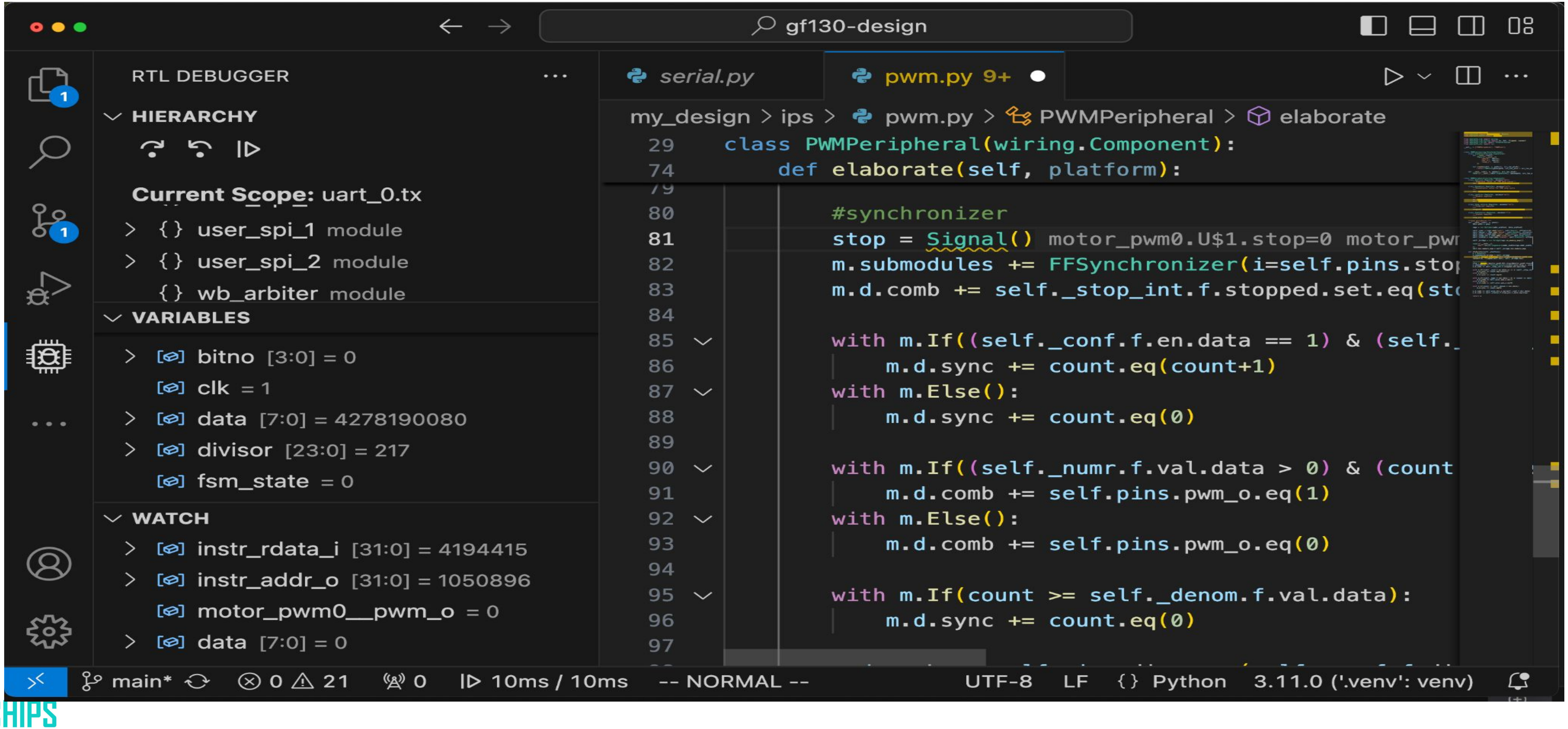
PDK

Foundry VARs
Foundry Partners
OSAT Partners

Does our Platform help?



Does our Platform help?



gf130-design

RTL DEBUGGER

HIERARCHY

Current Scope: uart_0.tx

- {} user_spi_1 module
- {} user_spi_2 module
- {} wb_arbiter module

VARIABLES

- bitno [3:0] = 0
- clk = 1
- data [7:0] = 4278190080
- divisor [23:0] = 217
- fsm_state = 0

WATCH

- instr_rdata_i [31:0] = 4194415
- instr_addr_o [31:0] = 1050896
- motor_pwm0__pwm_o = 0
- data [7:0] = 0

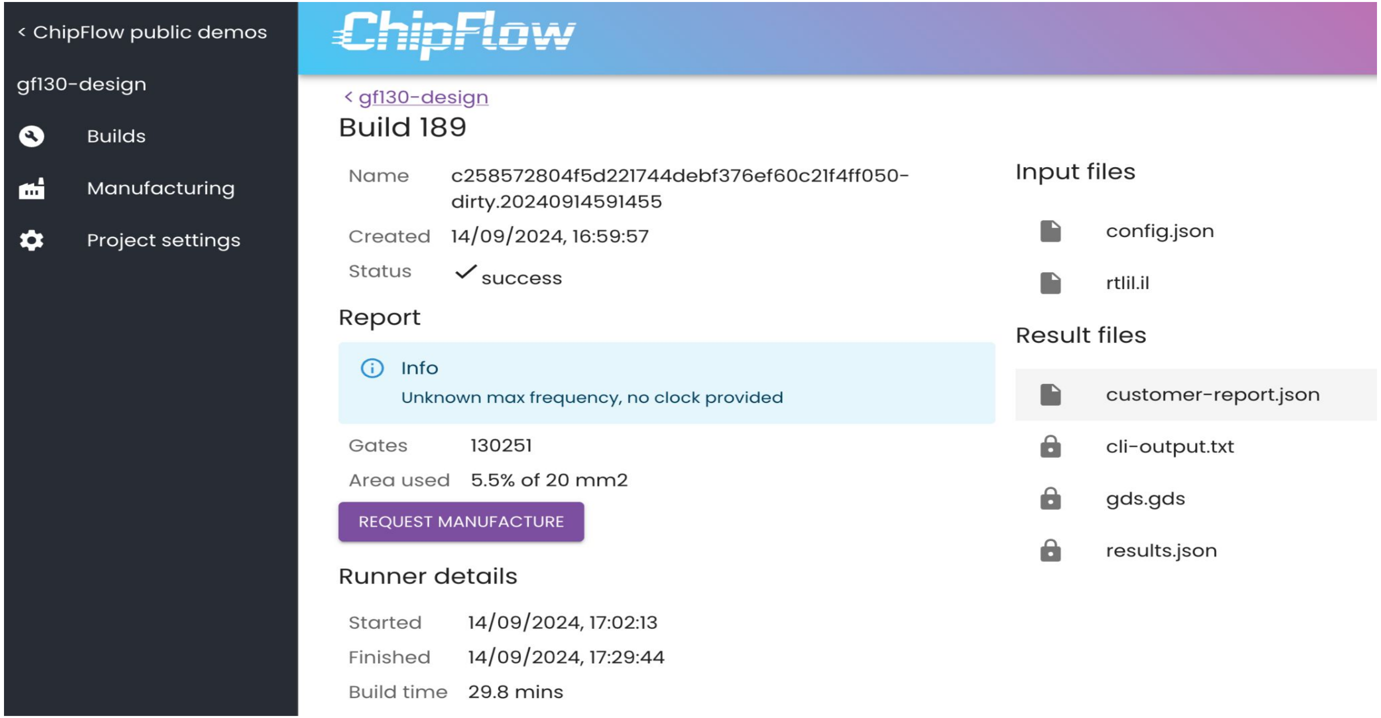
serial.py | pwm.py 9+

my_design > ips > pwm.py > PWMPeripheral > elaborate

```
class PWMPeripheral(wiring.Component):
    def elaborate(self, platform):

        #synchronizer
        stop = Signal()  motor_pwm0.U$1.stop=0 motor_pwr
        m.submodules += FFSynchronizer(i=self.pins.stop
        m.d.comb += self._stop_int.f.stopped.set.eq(stc

        with m.If((self._conf.f.en.data == 1) & (self._
            m.d.sync += count.eq(count+1)
        with m.Else():
            m.d.sync += count.eq(0)

        with m.If((self._numr.f.val.data > 0) & (count
            m.d.comb += self.pins.pwm_o.eq(1)
        with m.Else():
            m.d.comb += self.pins.pwm_o.eq(0)

        with m.If(count >= self._denom.f.val.data):
            m.d.sync += count.eq(0)
```

main* ⊗ 0 ⚠ 21 0 10ms / 10ms -- NORMAL -- UTF-8 LF {} Python 3.11.0 ('.venv': venv)

Does our Platform help?



Build 189

Name	c258572804f5d221744debf376ef60c21f4ff050-dirty.20240914591455
Created	14/09/2024, 16:59:57
Status	✓ success

Report

Info
Unknown max frequency, no clock provided

Gates	130251
Area used	5.5% of 20 mm2

Runner details

Started	14/09/2024, 17:02:13
Finished	14/09/2024, 17:29:44
Build time	29.8 mins

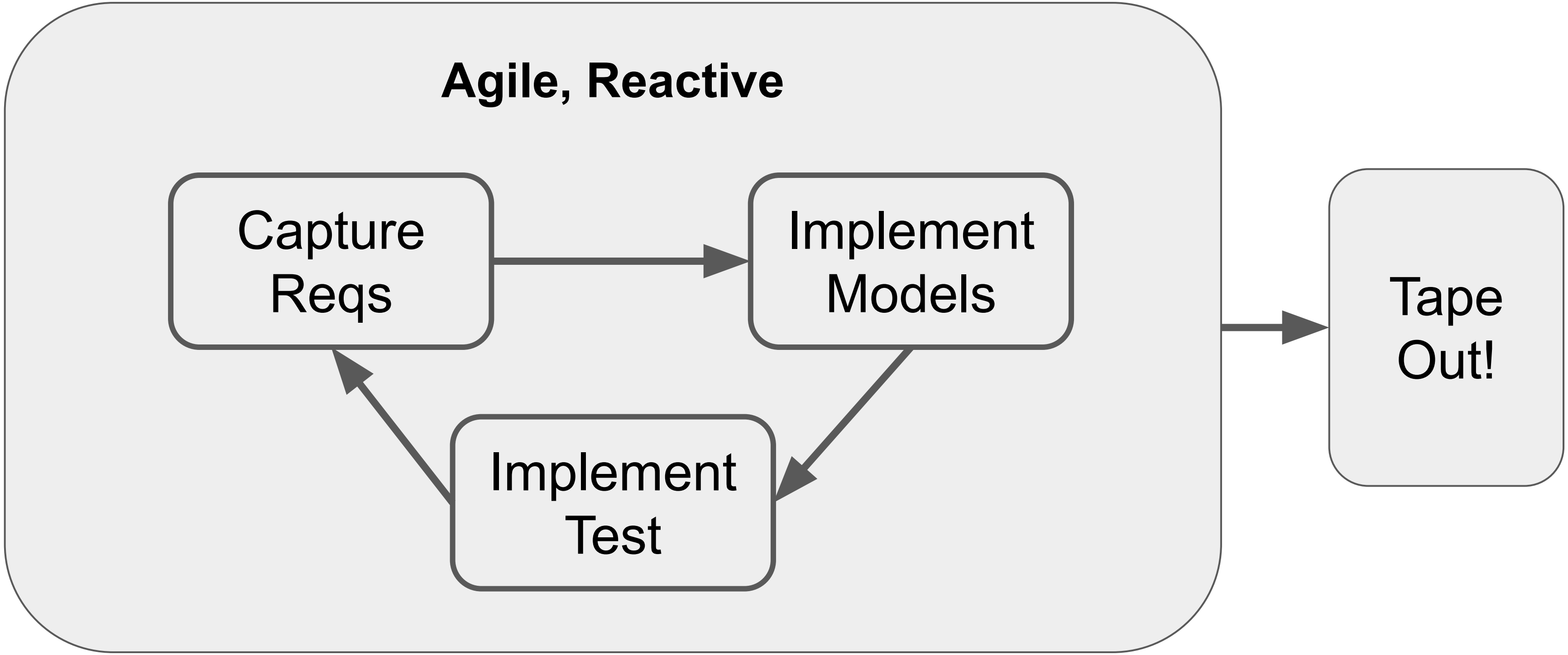
Input files

- config.json
- rtlil.il

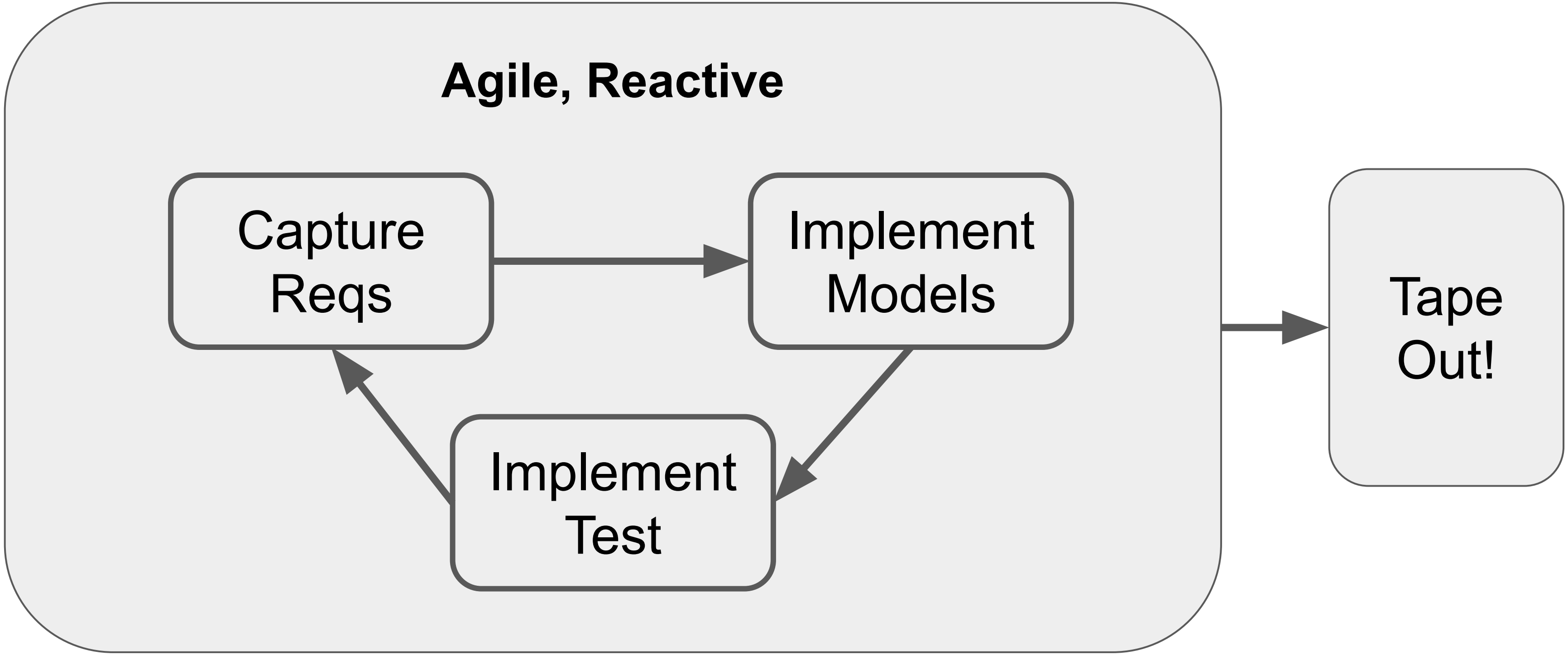
Result files

- customer-report.json
- cli-output.txt
- gds.gds
- results.json

A different approach to design house



A different approach to design house



Biggest challenges?

Biggest challenges?

GF130 BCD

Biggest challenges?

Functional Safety

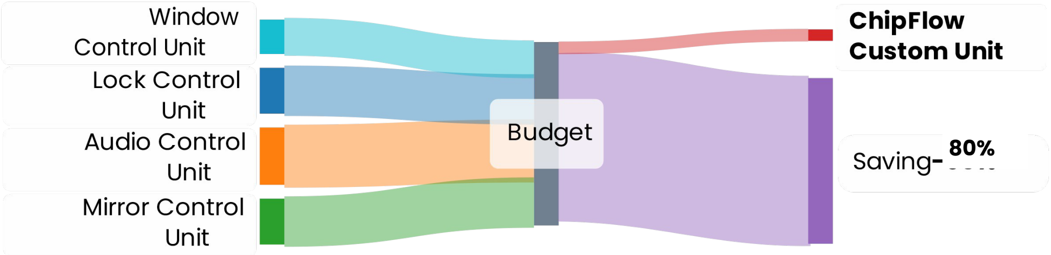
Outcomes

CASE STUDY

11 to 1 in 3 months

A leading luxury automotive OEM needed a path forward for overhauling their **electronic architecture.**

The **ChipFlow platform** was used to deliver an all-in-one custom ASIC against their requirements for **full door functionality** connected by **Ethernet T1S.**



Initial test chip was designed in **3 months**, at a cost of **$300k**, giving the customer a path to full control of their platform needs.

This custom solution replaced **4 control units** and **11 ICs** with **1 IC,** representing a saving of **80%** on the budget for harness and electronics.

Questions?

Rob Taylor <rob.taylor@chipflow.io