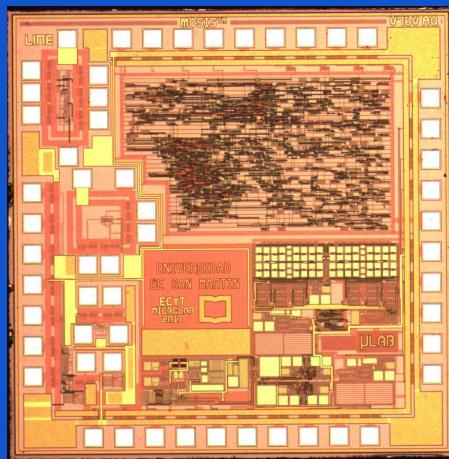


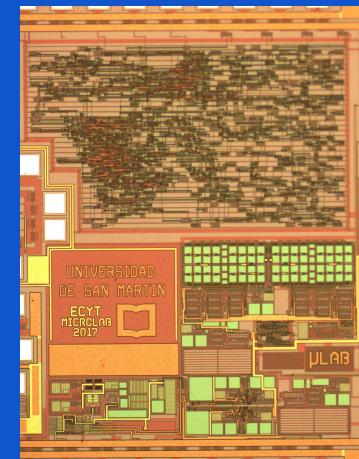


Educational Design Kit for Synopsys Tools with a set of Characterized Standard Cell Library



Members:

- Leandro Tozzi
- Yao-Ming Kuo
- Leandro Arana
- Cristian Marchese
- Luis Seva





Analog Design Flow

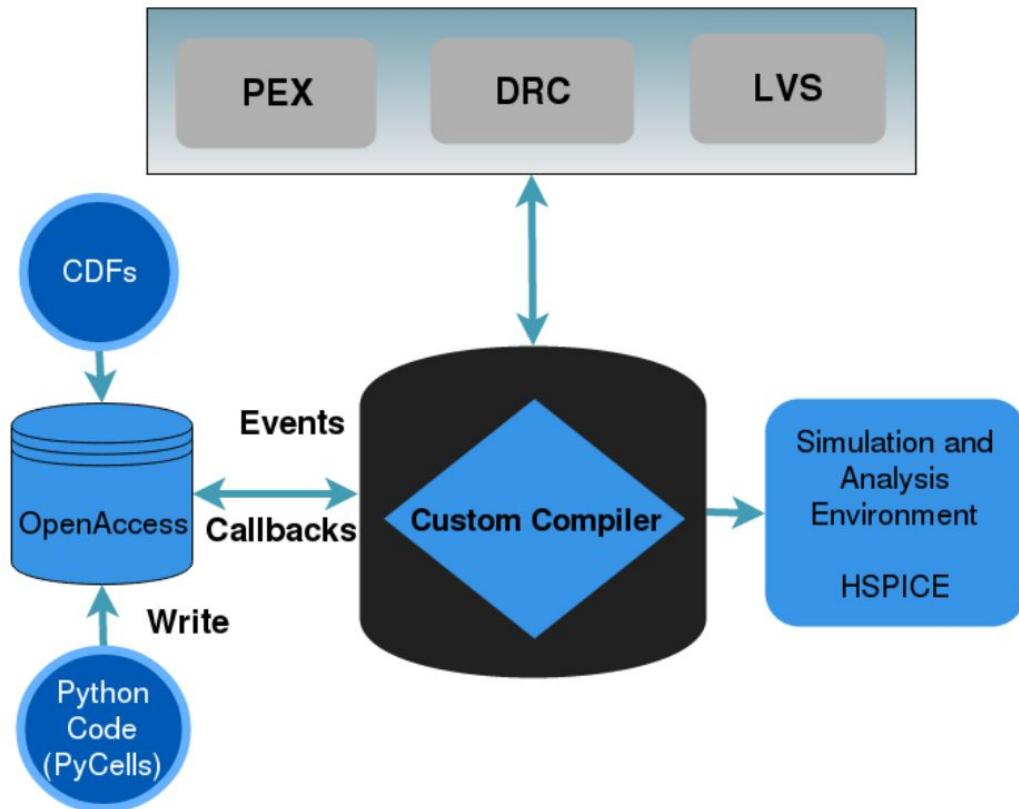
Open Design Kit

- ❖ Pycells
- ❖ Callbacks
- ❖ CDFs

Technology files/rules

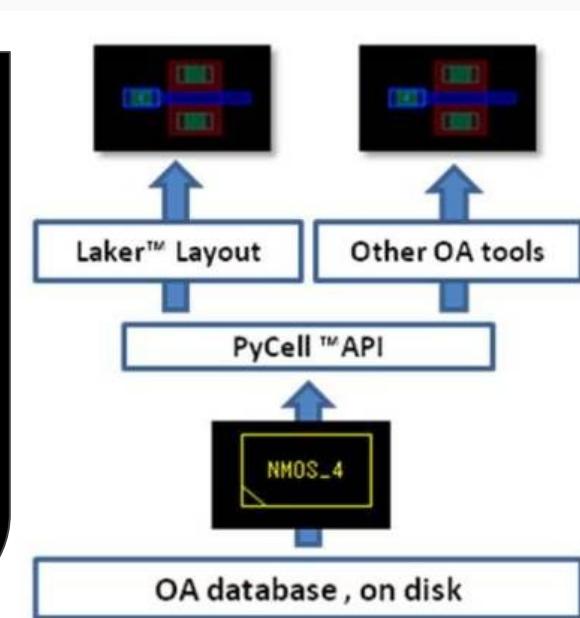
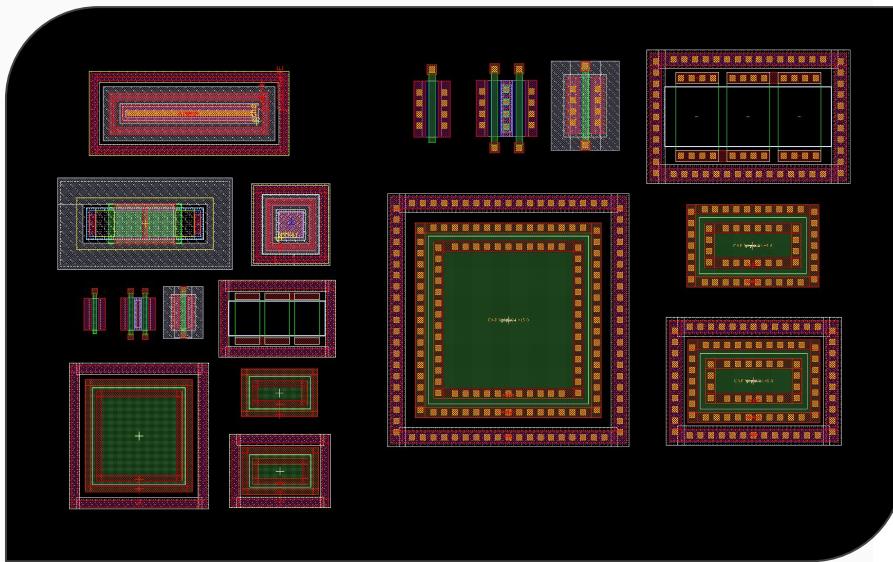
- ❖ Layermap
- ❖ Simulation SPICE models
- ❖ DRC/LVS rules

IC Analog Flow



Pycells

- ❖ Python code →
- ❖ CDFs and callbacks
- ❖ OA Database generation
- ❑ Abutment
- ❑ Multiplicity / Fingers
- ❑ High voltage MOS



22 different pycells!

- Resistors
- Capacitors
- Schottky diodes
- PNP transistor
- NMOS / PMOS
- Power transistors



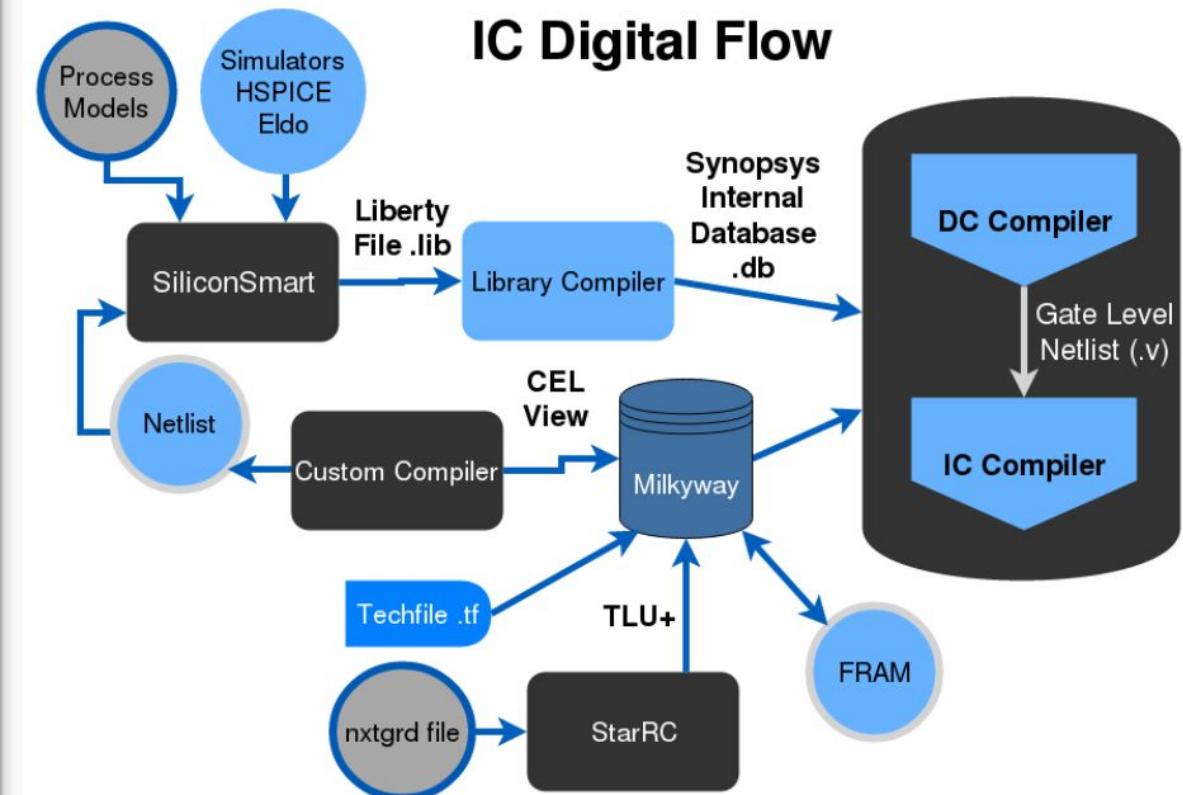
Digital Design Flow

Open Design Kit

- ❖ Liberty Files (.lib)
- ❖ Milkyway Database
- ❖ Example scripts

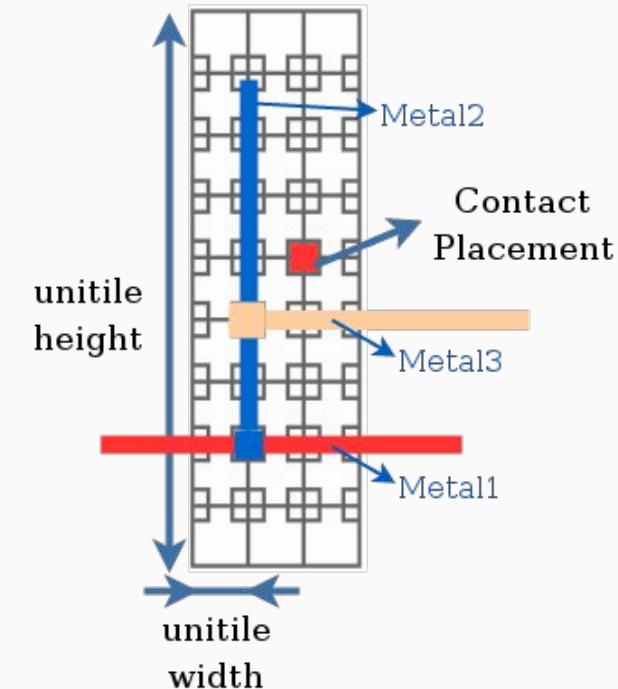
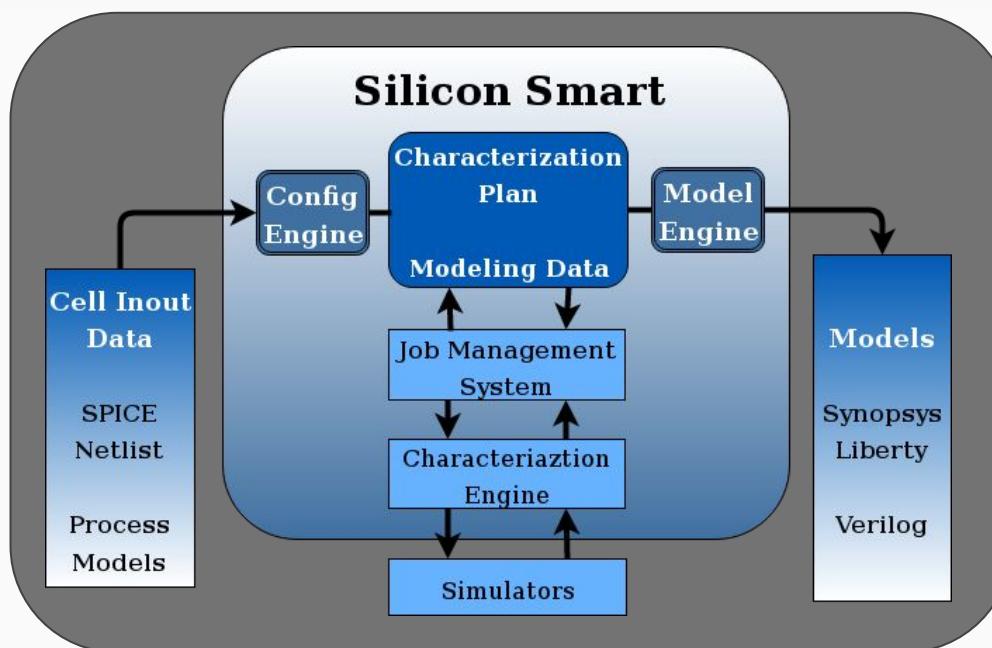
Technology files/rules

- ❖ Techfile
- ❖ TLUplus
- ❖ Simulation SPICE models



Standard Cells

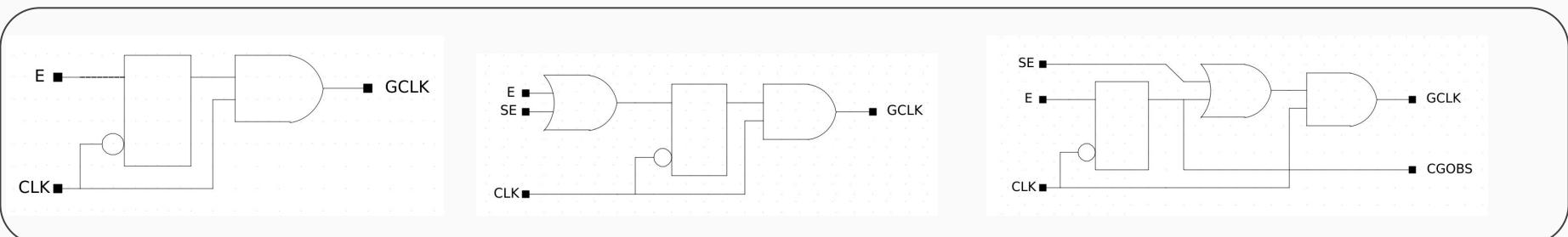
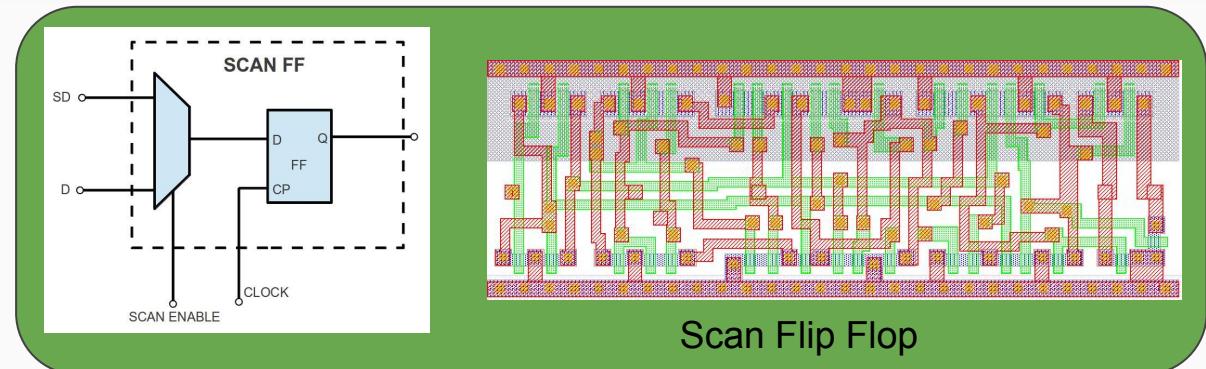
- ❖ Simulation & Layout
- ❖ Characterize
- ❖ Milkyway database generation
- ❖ 100 Characterized standard cells!



Special Cells

- ❖ Simulation & Layout
- ❖ Characterize
- ❖ Milkyway database generation
- ❖ 100 Characterized standard cells!

- ❖ Clock Gating
- ❖ Scan Flip Flop
- ❖ Antenna Cells
- ❖ Feed Cells
- ❖ Logic Cells





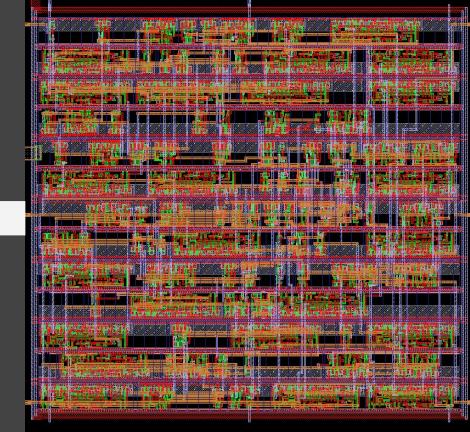
Synthesis, Place & Route

```
1 module uart;
2   input clk; // The master clock for this module
3   input rst; // Synchronous reset
4   input rx; // Incoming serial line
5   output tx; // Outgoing serial line
6   input transmit; // Signal to transmit
7   output tx_byte; // Byte to transmit
8   output received; // Indicated that a byte has been received.
9   output [7:0] rx_byte; // Byte received
10  output is_receiving; // Low when receive line is idle.
11  output is_transmitting; // Low when transmit line is idle.
12  output recv_error; // Indicates error in receiving packet.
13  );
14
15 parameter CLOCK_DIVIDE = 2604; //1302 for 50MHz; // clock rate (25MHz) / (baud rate (9600) * 4)
16
17 // States for the receiving state machine.
18 // These are not constants, not parameters to override.
19 parameter RX_IDLE = 1;
20 parameter RX_CHECK_START = 1;
21 parameter RX_READ_BITS = 2;
22 parameter RX_CHECK_STOP = 3;
23 parameter RX_DELAY_RESTART = 4;
24 parameter RX_ERROR = 5;
25 parameter RX_RECEIVED = 6;
26
27 // States for the transmitting state machine.
28 // These are not constants, not parameters to override.
29 parameter TX_IDLE = 0;
30 parameter TX_SENDING = 1;
31 parameter TX_DELAY_RESTART = 2;
32
33 reg [10:0] rx_clk_divider;
34 reg [10:0] tx_clk_divider;
35
36 reg [2:0] recv_state;
37 reg [5:0] rx_countdown;
38
39 endmodule
```

RTL (.vhdl .v .sv)

```
8 module uart_receiver ( clk, reset, rx, ready, pout );
9   output [7:1] pout;
10  input clk, reset, rx;
11  output ready;
12  wire N23, N24, N25, N26, N27, N28, n10, n11, n13, n14, n15, n16, n17, n19,
13   n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37,
14   n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75,
17   n76, n77, n78, n79, n80, n81, n82, n83;
18  wire [5:0] clk1x25_rx;
19  wire [1:0] state_reg;
20  wire [3:0] s_reg;
21  wire [2:0] n_reg;
22  wire [1:0] state_next;
23
24
25  N3X1 U13 (.A(n20), .B(n82), .C(n21), .OUT(state_next[1]));
26  N3X1 U14 (.A(n22), .B(n23), .OUT(n21));
27  N3X1 U15 (.A(n24), .B(n25), .C(n22), .OUT(n20));
28  N3TDX1 U16 (.A(n26), .B(n27), .C(n28), .OUT(state_next[0]));
29  AN21X1 U17 (.A(n22), .B(n23), .C(n29), .OUT(n28));
30  NO2X1 U18 (.A(rx), .B(n81), .OUT(n29));
31  N3TDX1 U19 (.A(n23), .B(n30), .C(n79), .OUT(n27));
32  NO2X1 U20 (.A(n25), .B(n80), .C(n78), .OUT(n26));
33  A022X1 U21 (.A(pout[7]), .B(n75), .C(rx), .D(n31), .OUT(n53));
34  A022X1 U22 (.A(pout[6]), .B(n75), .C(pout[7]), .D(n31), .OUT(n54));
35  A022X1 U23 (.A(pout[5]), .B(n75), .C(pout[6]), .D(n68), .OUT(n55));
36  A022X1 U24 (.A(pout[4]), .B(n75), .C(pout[5]), .D(n67), .OUT(n56));
37  A022X1 U25 (.A(pout[3]), .B(n75), .C(pout[4]), .D(n31), .OUT(n57));
38  A022X1 U26 (.A(pout[2]), .B(n75), .C(pout[3]), .D(n68), .OUT(n58));
39  A022X1 U27 (.A(pout[1]), .B(n75), .C(pout[2]), .D(n31), .OUT(n59));
40  A022X1 U28 (.A(pout[0]), .B(n75), .C(pout[1]), .D(n68), .OUT(n60));
41  NO2X1 U29 (.A(n32), .B(n31), .C(n33), .D(n34), .OUT(n35));
42  NO2X1 U30 (.A(n35), .B(n34), .C(n36), .D(n37), .OUT(n38));
43  NO2X1 U31 (.A(n36), .B(n35), .C(n37), .D(n38), .OUT(n39));
44
45 endmodule
```

Gate Level Netlist



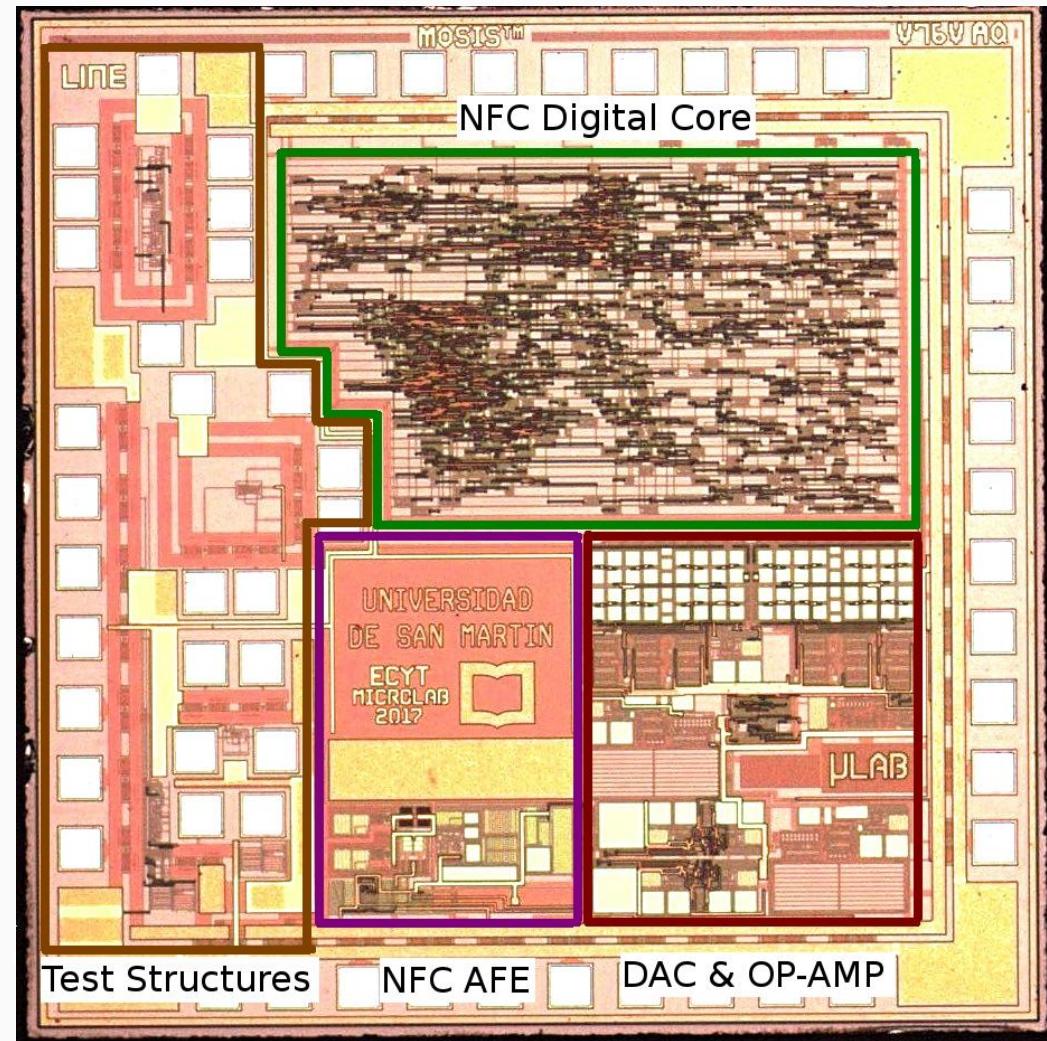
Place & Route



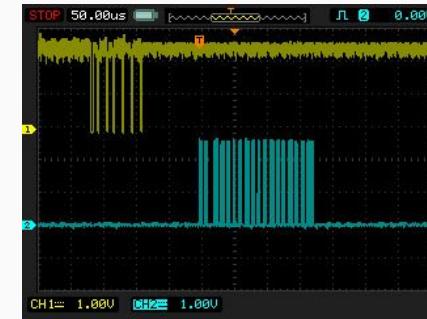
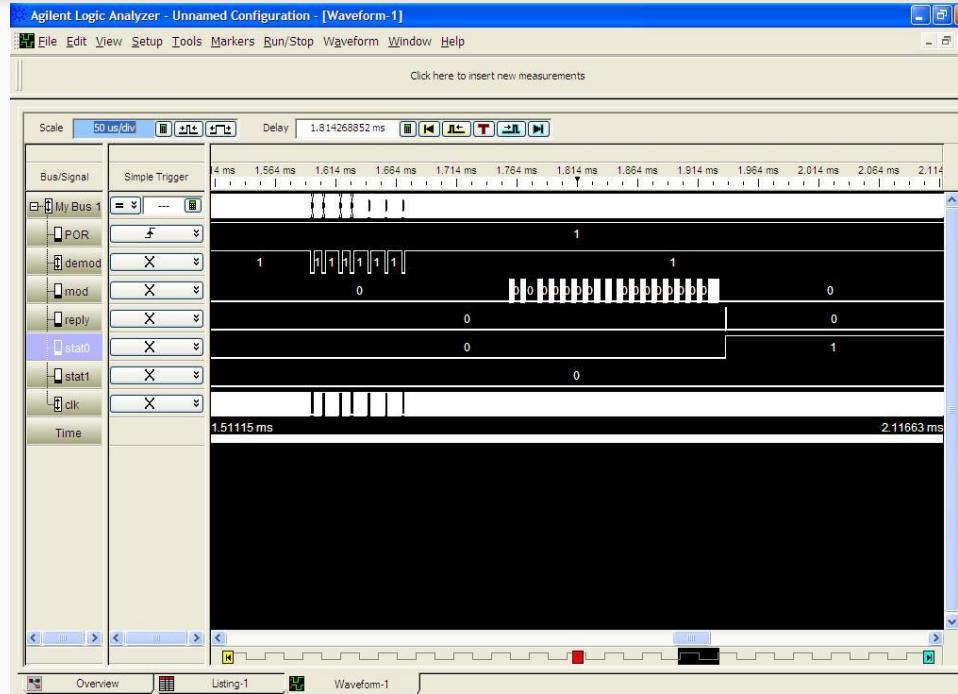
MOSIS MPW

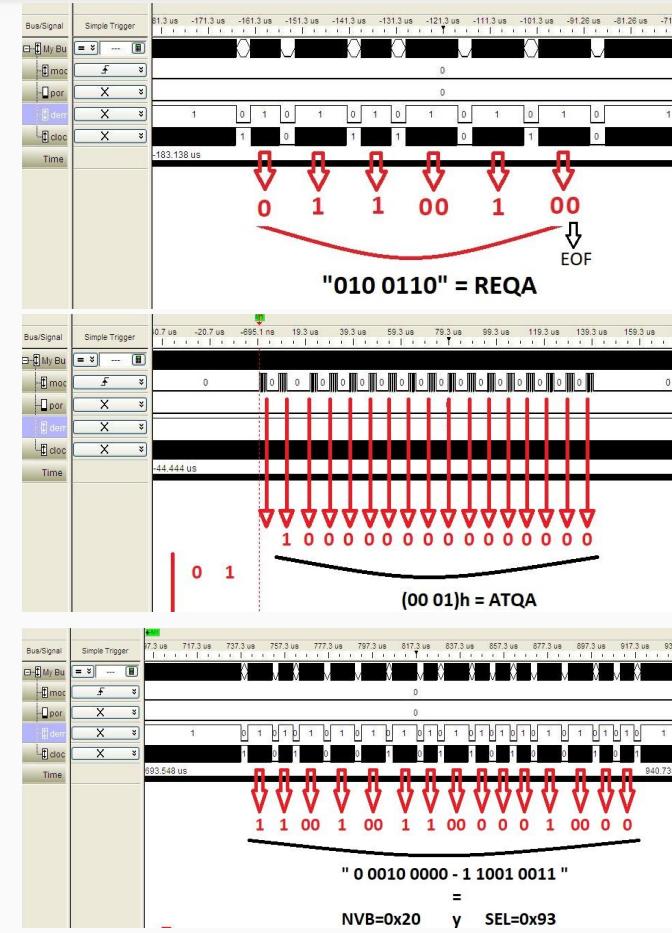
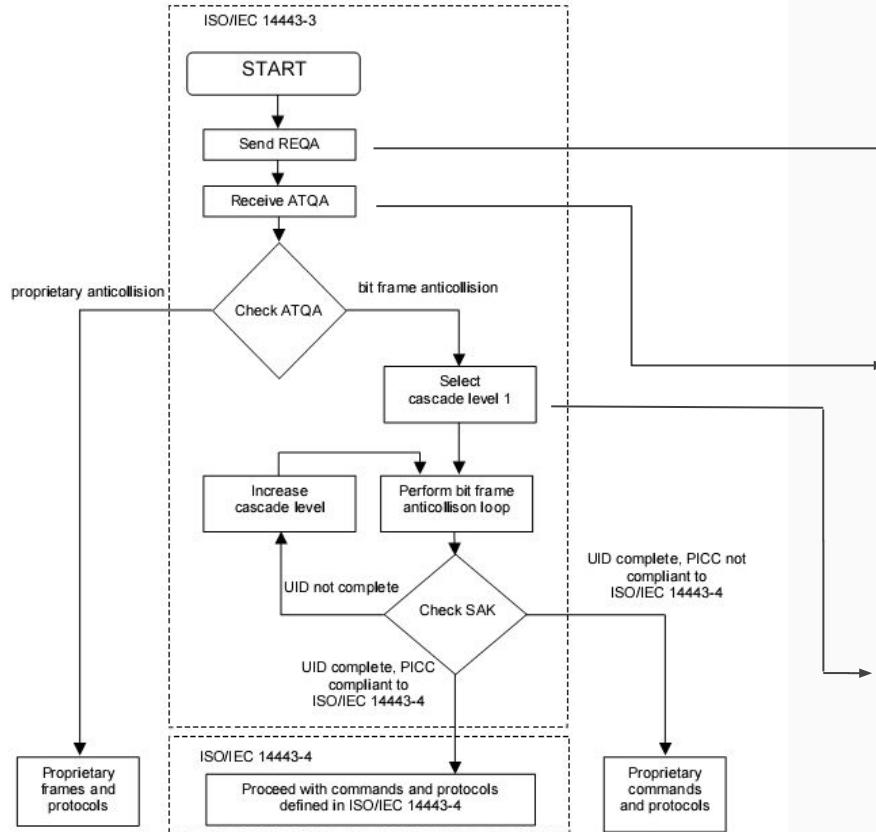
On Semiconductor 500 nm

- Operating Voltage: 3.3 ~ 5 V
- 3 Metal
- 2 Poly
- poly-poly capacitors



Measurement & Validation (NFC Core)







Download link

Contact & Support:

- Leandro Tozzi

ltozzi@unsam.edu.ar

- Yao-Ming Kuo

ykuo@frba.utn.edu.ar

The screenshot shows a web browser window with the URL https://github.com/UNSAMDCI/PDK_ONC5. The page title is "PDK_ONC5/README.md". The main content area features the logo of "UNSAM Educational PDK ONC5" and the text "Educational Design Kit for Synopsys Tools with a set of Characterized Standard Cell Library". Below this, there is a section titled "Authors" with a bulleted list of names and emails. Another section titled "Structure" contains two code snippets: one for the DC Compiler and one for the IC Compiler (ICC).

UNSAM Educational PDK ONC5

Educational Design Kit for Synopsys Tools with a set of Characterized Standard Cell Library

Authors

- Leandro Tozzi ltozzi@unsam.edu.ar
- Yao-Ming Kuo ykuo@frba.utn.edu.ar
- Leandro Arana larana@fi.uba.com.ar
- Cristian Marchese cmarchese@est.frba.utn.edu.ar
- Luis Seva lseva@est.frba.utn.edu.ar

Structure

```
dc : DC Compiler
      the simple.tcl script synthesizes an uart with clock gating, which is then used in ICC
dc / testcase_dft :
      another script that uses DFT on an adder to test the scan flip flops

icc : IC Compiler
icc / uart_example :
      run in bash > icc_shell -gui
      then in icc_shell: source icc_run.tcl > log.txt (Place & Route of the UART)
icc / icc_onc5:
      milkyway db
icc / io_nands
```

https://github.com/UNSAMDCI/PDK_ONC5
Github: UNSAMDCI/PDK_ONC5

