A 130nm 32-bit RISC-V microcontroller

ONCHIP

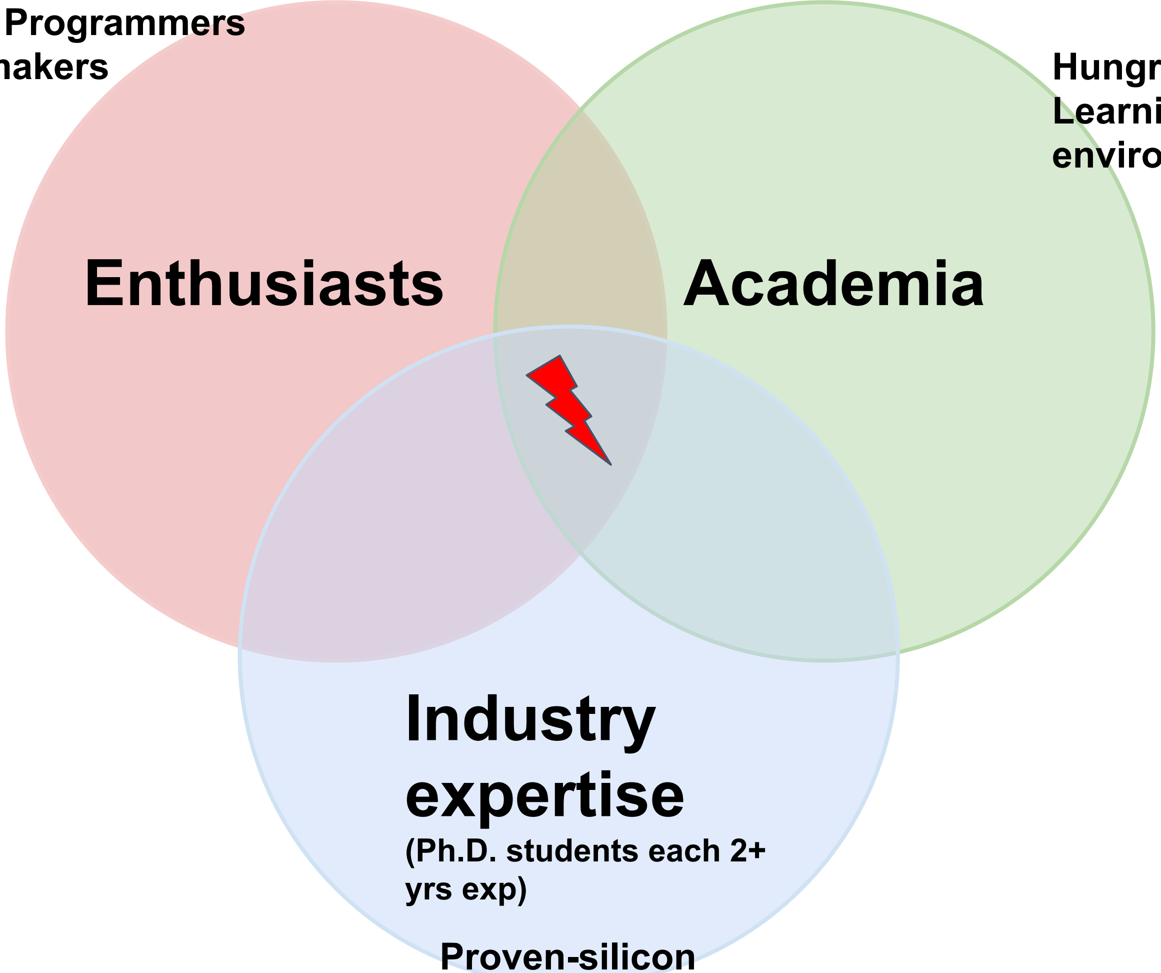
Universidad
Industrial de
Santander

UIS

@onchipUIS

Who we are

FPGA Programmers
PCB makers



Enthusiasts

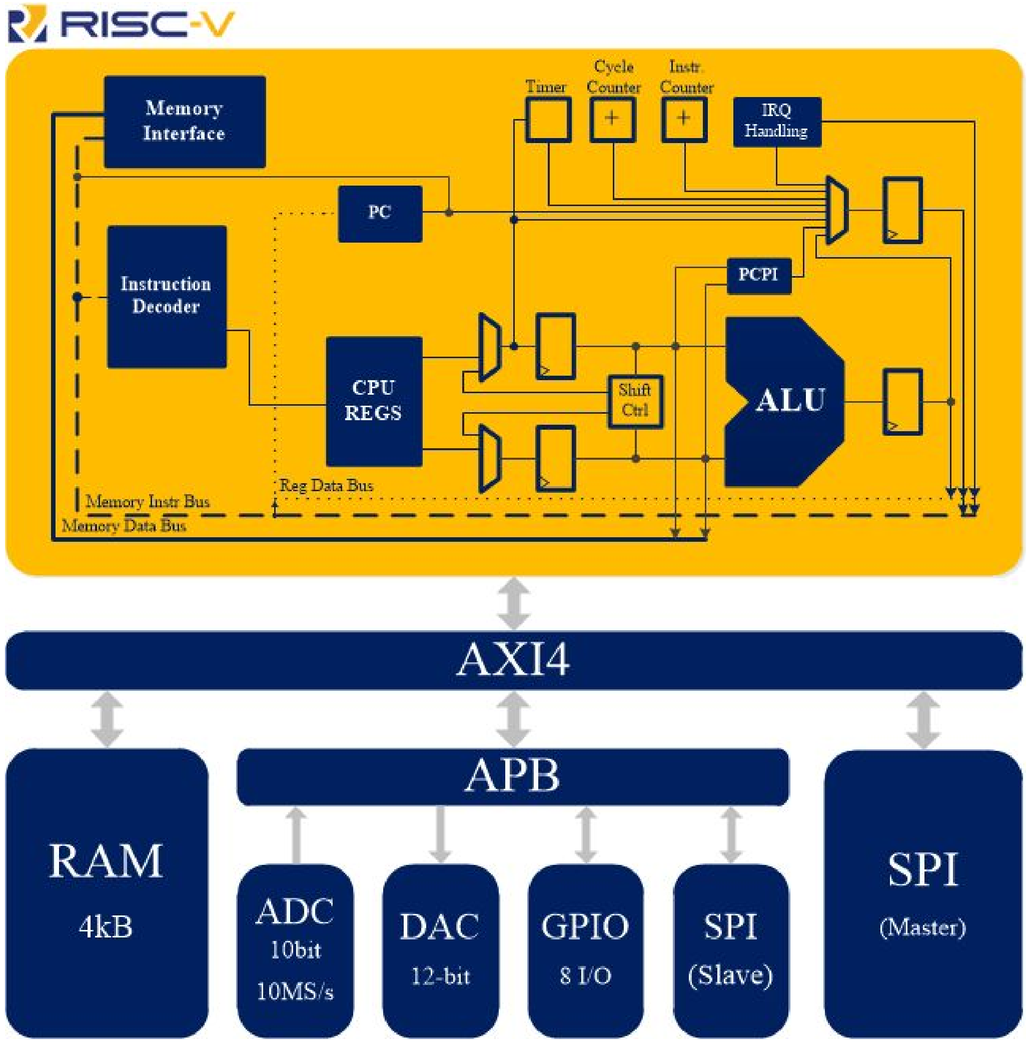
Academia

Industry expertise
(Ph.D. students each 2+ yrs exp)

Hungry-learn students
Learning and research environment

Proven-silicon
Circuit-design skills

A 32-bit RISC-V based microcontroller



RISC-V

Memory Interface

Timer

Cycle Counter

Instr. Counter

IRQ Handling

PC

PCPI

Instruction Decoder

CPU REGS

Shift Ctrl

ALU

Reg Data Bus

Memory Instr Bus

Memory Data Bus

AXI4

APB

RAM 4kB

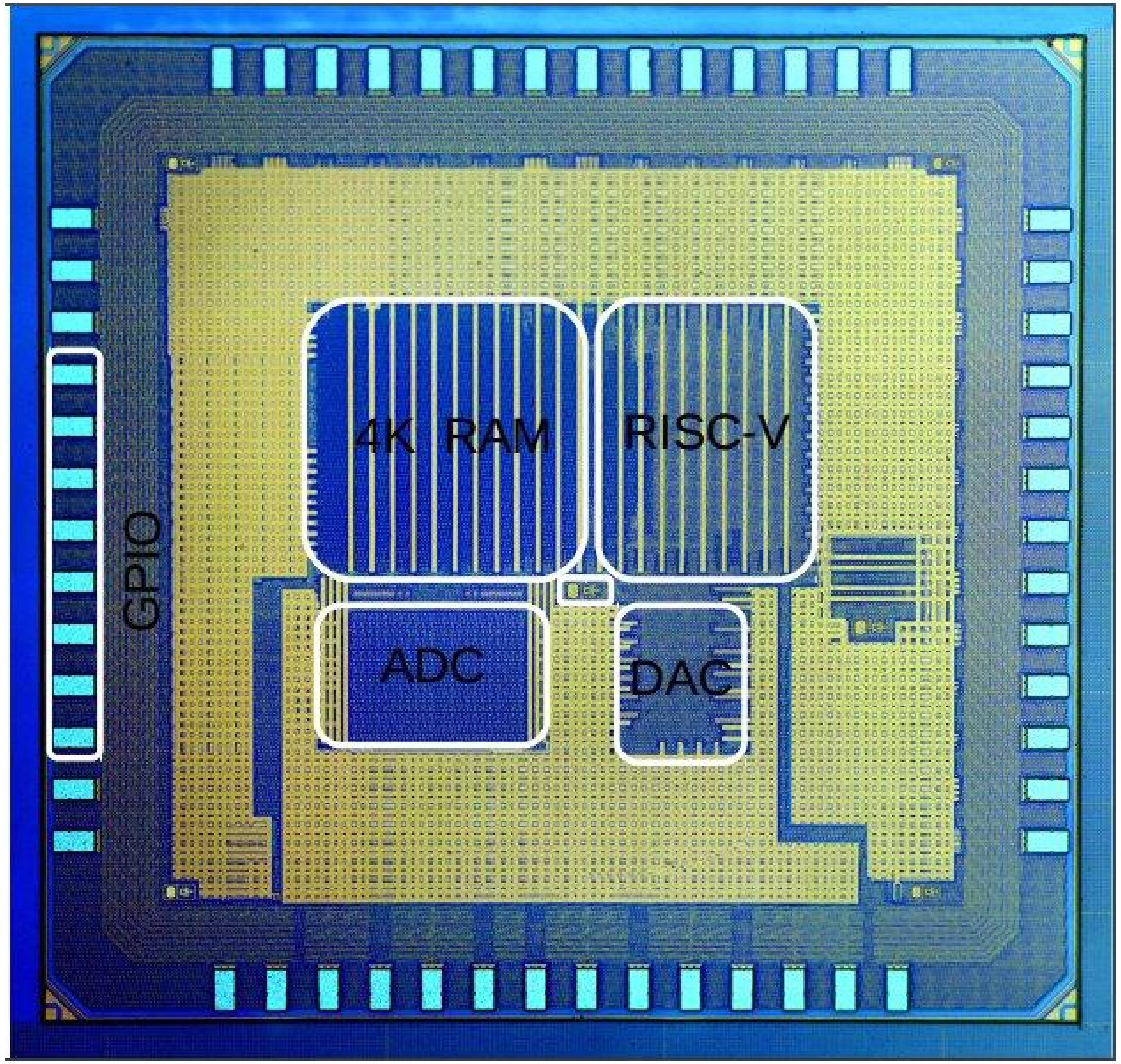
ADC 10bit 10MS/s

DAC 12-bit

GPIO 8 I/O

SPI (Slave)

SPI (Master)



Some more details

- TSMC 130nm GP

- I/O standard lib 2.5V

- Regular VT cell lib

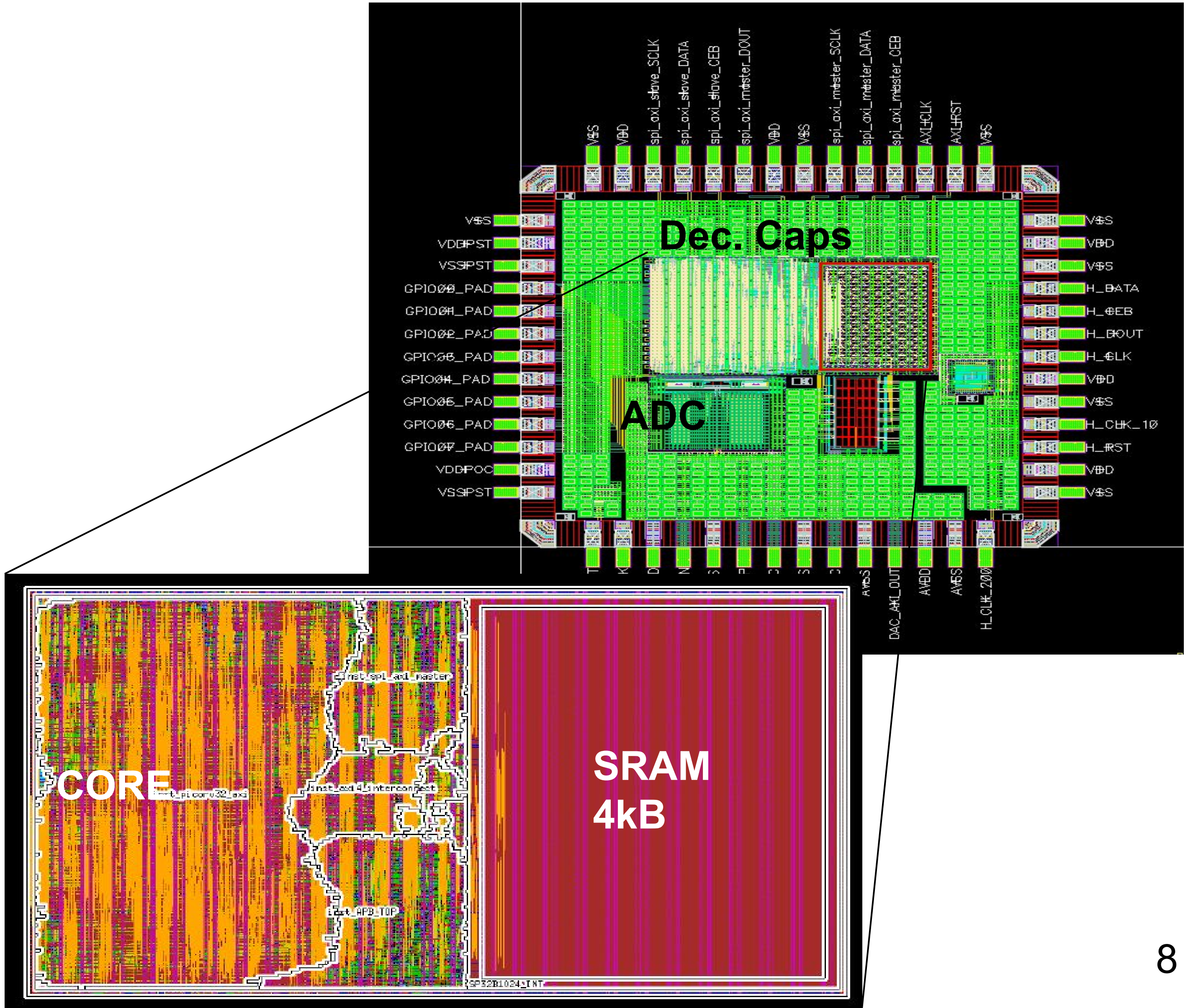
□+Some crafted cells

- Tested from 100kHz
- to 200MHz

□

- 64 pins

- Core+interfaces area:
- 800µm x 480µm



Performance

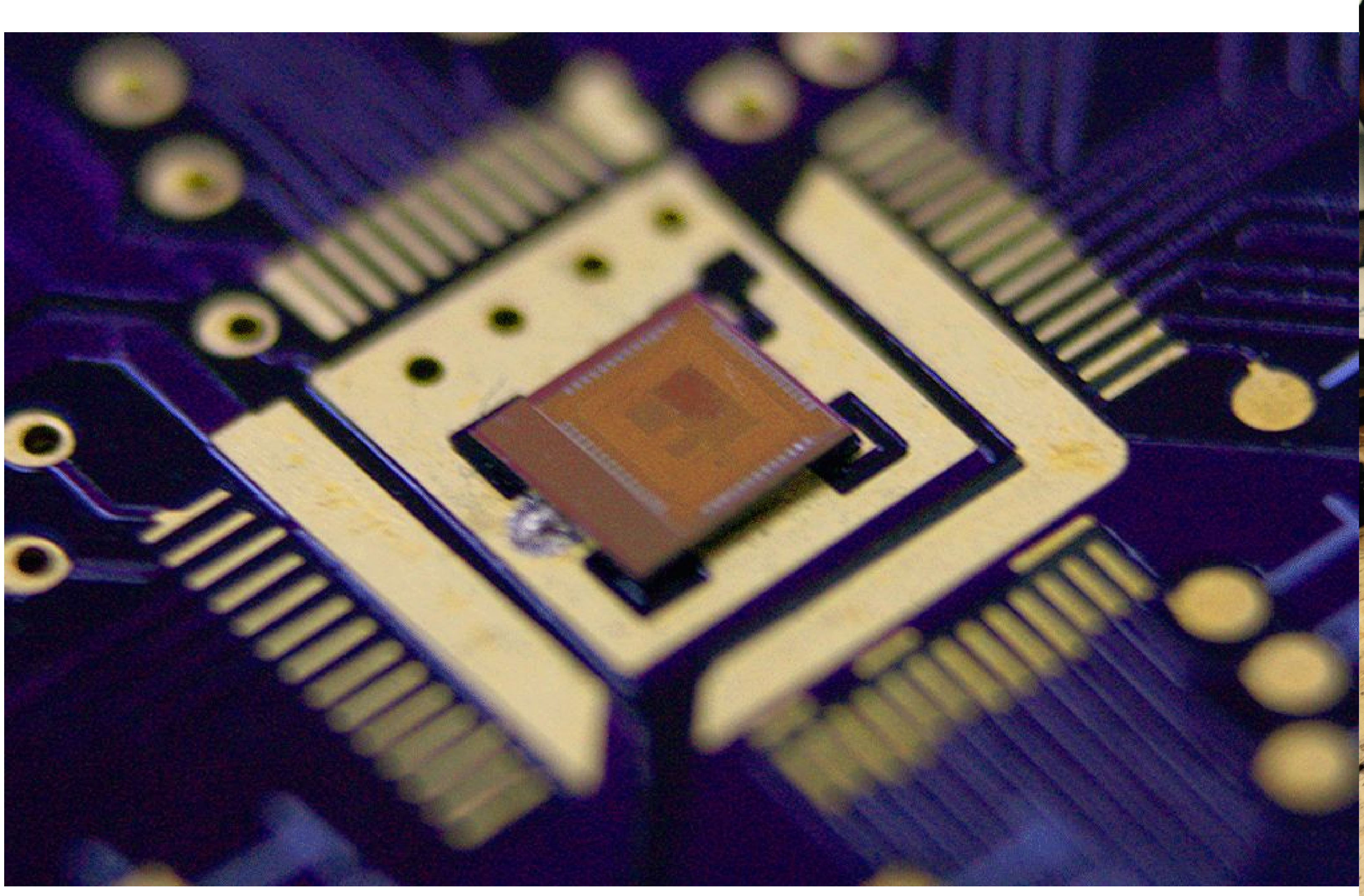
	µRISC-V
ISA	RV32IM
Architecture	Single-Issue In-Order 3-stage
Process	TSMC 130nm GP
Die Area	2.1mm x 2.1mm
uP core area	0.12 mm^2 !
Max. Frequency	100MHz
Core Voltage	1.2 V
I/O Voltage	2.5V
Core Dynamic Power	132 µW/MHz *
1.2V Current @2.5MHz	1.8mA

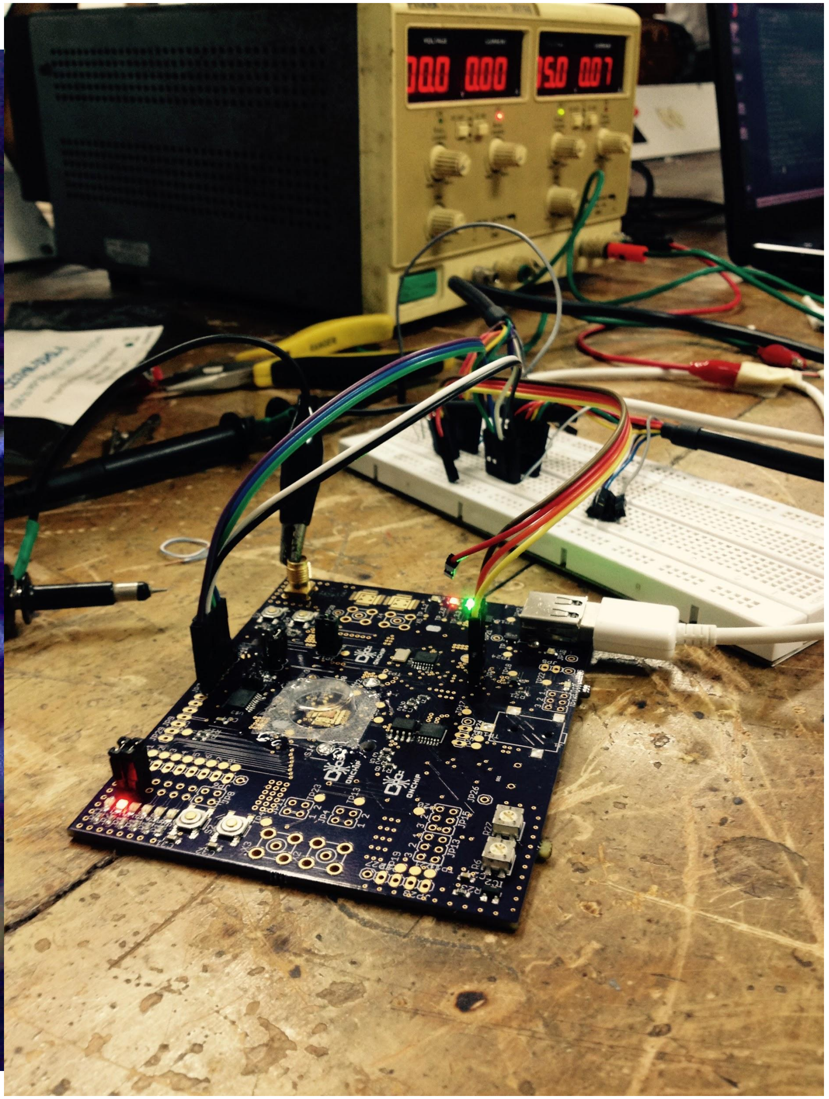
! Fully constrained clock and loads for SSLHH corner. Zero skew CTS.

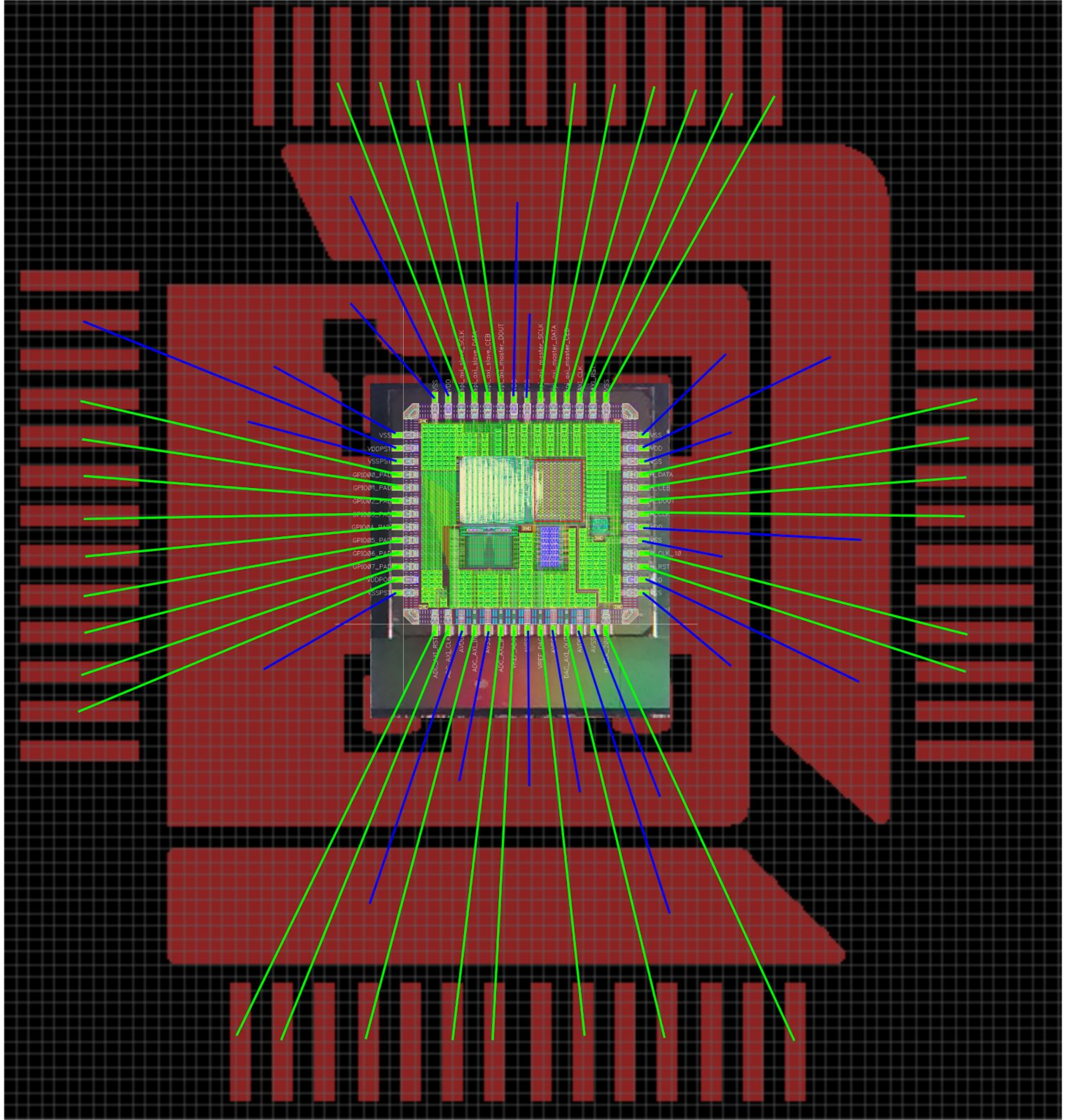
*** Measured at 100MHz running three while loops.**

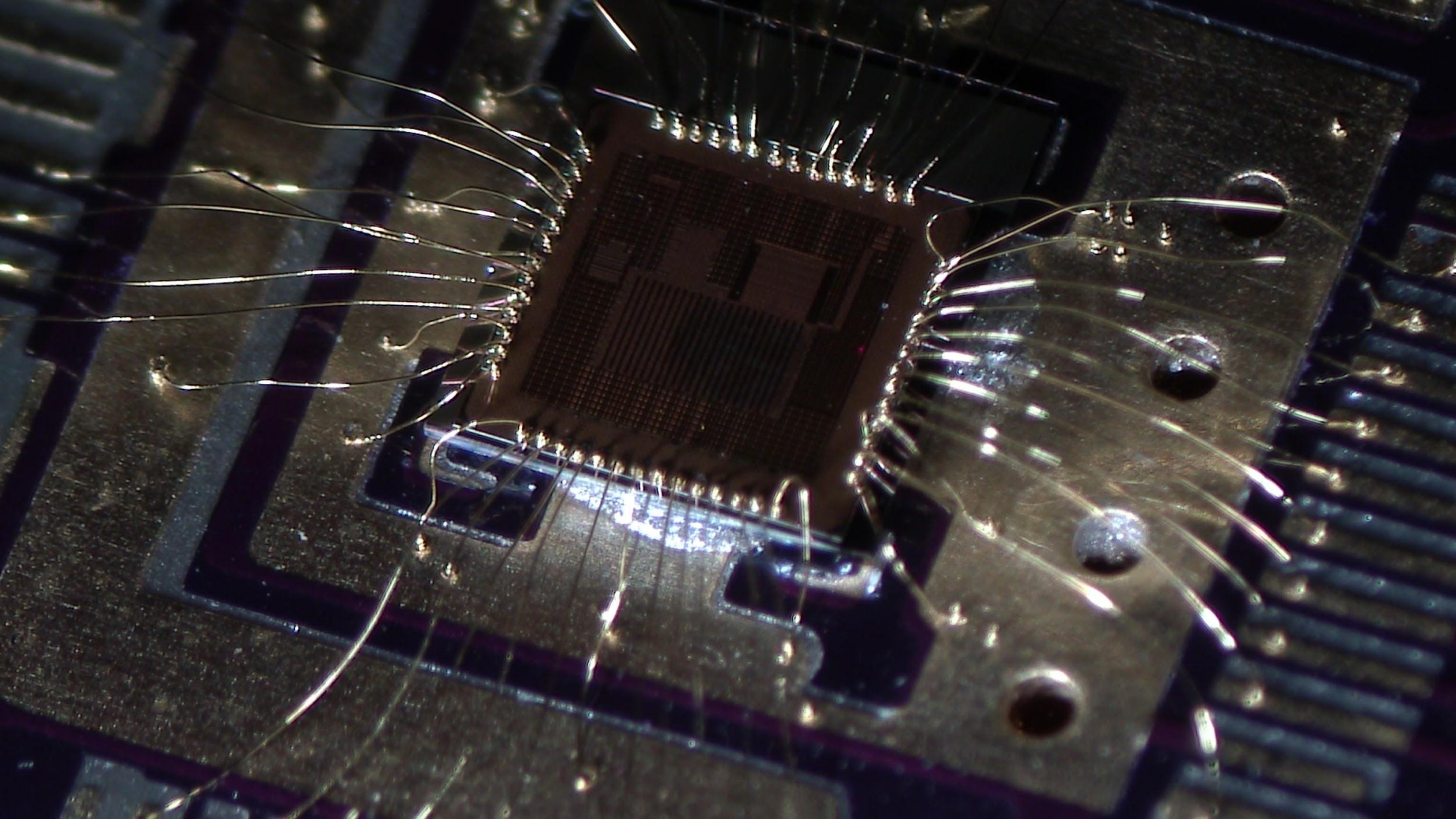
Clock Frequency [MHz]`	**Total Current (*1.2V and 2.5V supplies*) [mA]**
2.5	**2.5**
5	**4.62**
8.8	**5.01**
10	**5.12**
20	**6.13**
40	**8.10**
60	**10.20**
80	**12.18**
100	**14.13**
120	**16.14**
140	**18.09**
160	**20.06**
180	**21.89**
200	**23.76**

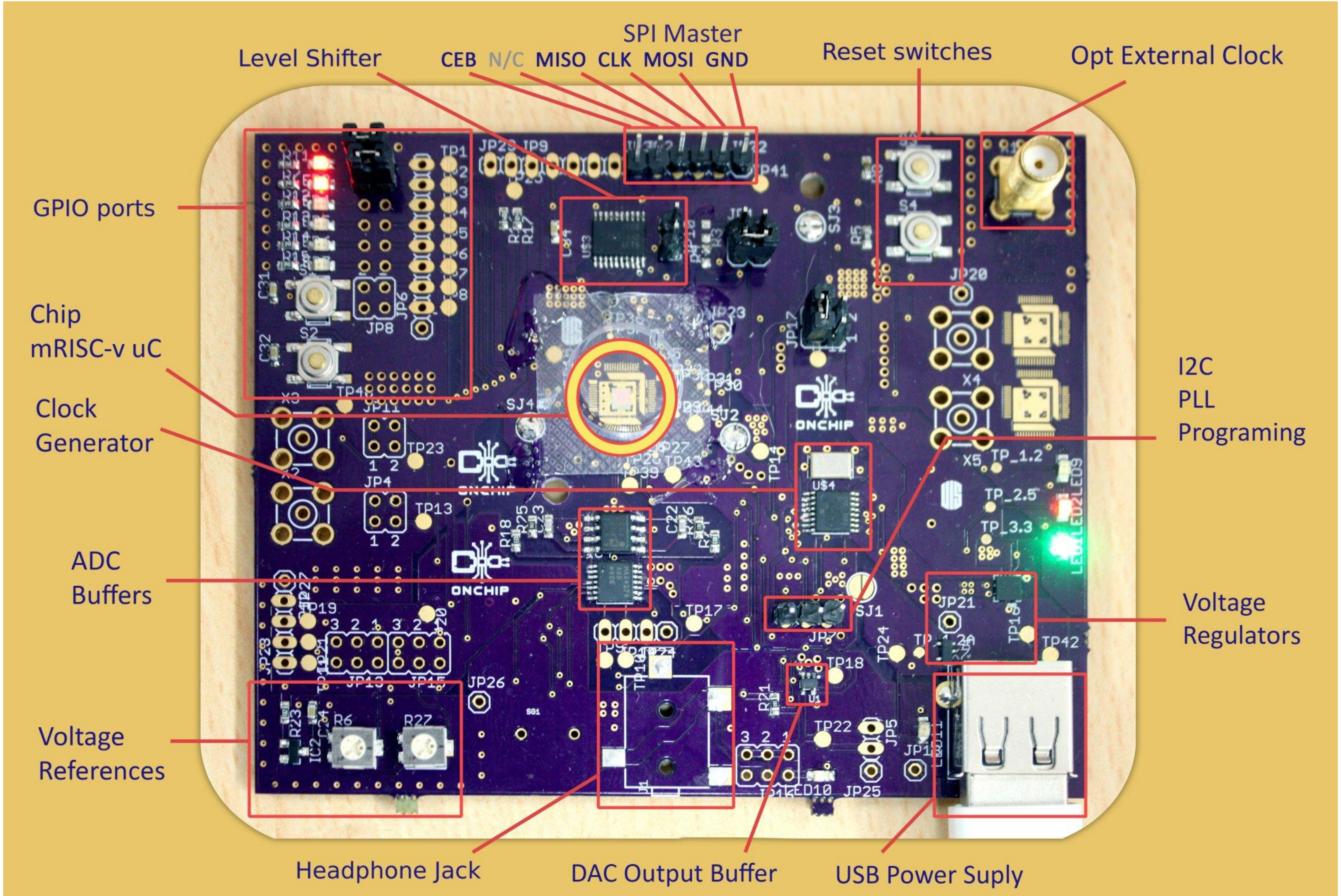
Chip-on-board





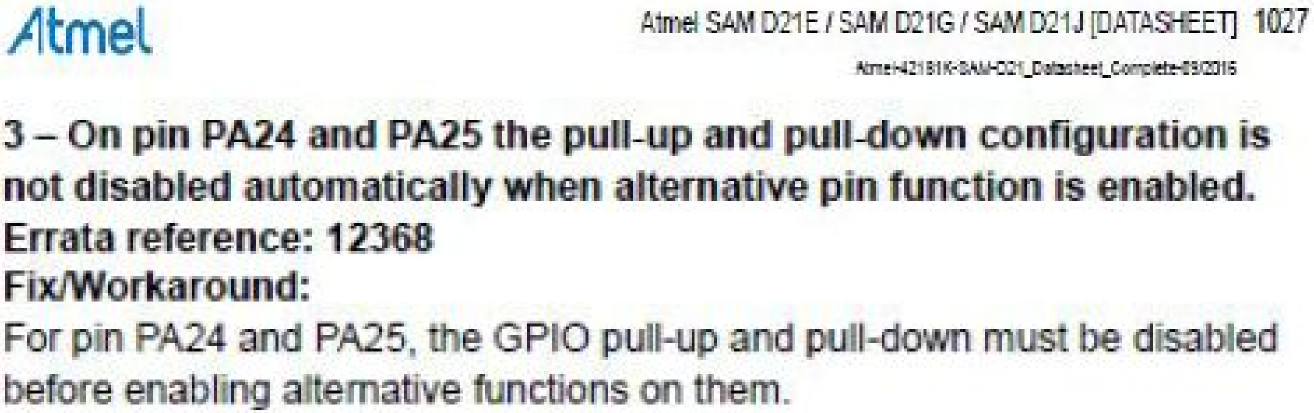

VSS
VDD
spi_axi_slave_SCLK
spi_axi_slave_DATA
spi_axi_slave_CEB
spi_axi_master_DOUT
VDD
VSS
spi_axi_master_SCLK
spi_axi_master_DATA
spi_axi_master_CEB
AXI_CLK
AXI_RST
VSS
VSS
VDDPST
VSSPST
GPIO00_PAD
GPIO01_PAD
GPIO02_PAD
GPIO03_PAD
GPIO04_PAD
GPIO05_PAD
GPIO06_PAD
GPIO07_PAD
VDDPOC
VSSPST
VSS
VDD
VSS
H_DATA
H_CEB
H_DOUT
H_CLK
VDD
VSS
H_CLK_10
H_RST
VDD
VSS
ADC_AXI_RST
ADC_AXI_CLK
AVDD
ADC_AXI_IN
AVSS
ADC_AXI_IN
VREF_ADC
AVSS
VREF_DAC
AVSS
DAC_AXI_OUT
AVDD
AVSS
H_CLK_200




SPI Master
Level Shifter
CEB N/C MISO CLK MOSI GND
Reset switches
Opt External Clock
GPIO ports
Chip
mRISC-v uC
Clock
Generator
I2C
PLL
Programing
ADC
Buffers
Voltage
Regulators
Voltage
References
Headphone Jack
DAC Output Buffer
USB Power Suply

Give sth that has no bugs

ATMEL
SAMD21 ERRATA



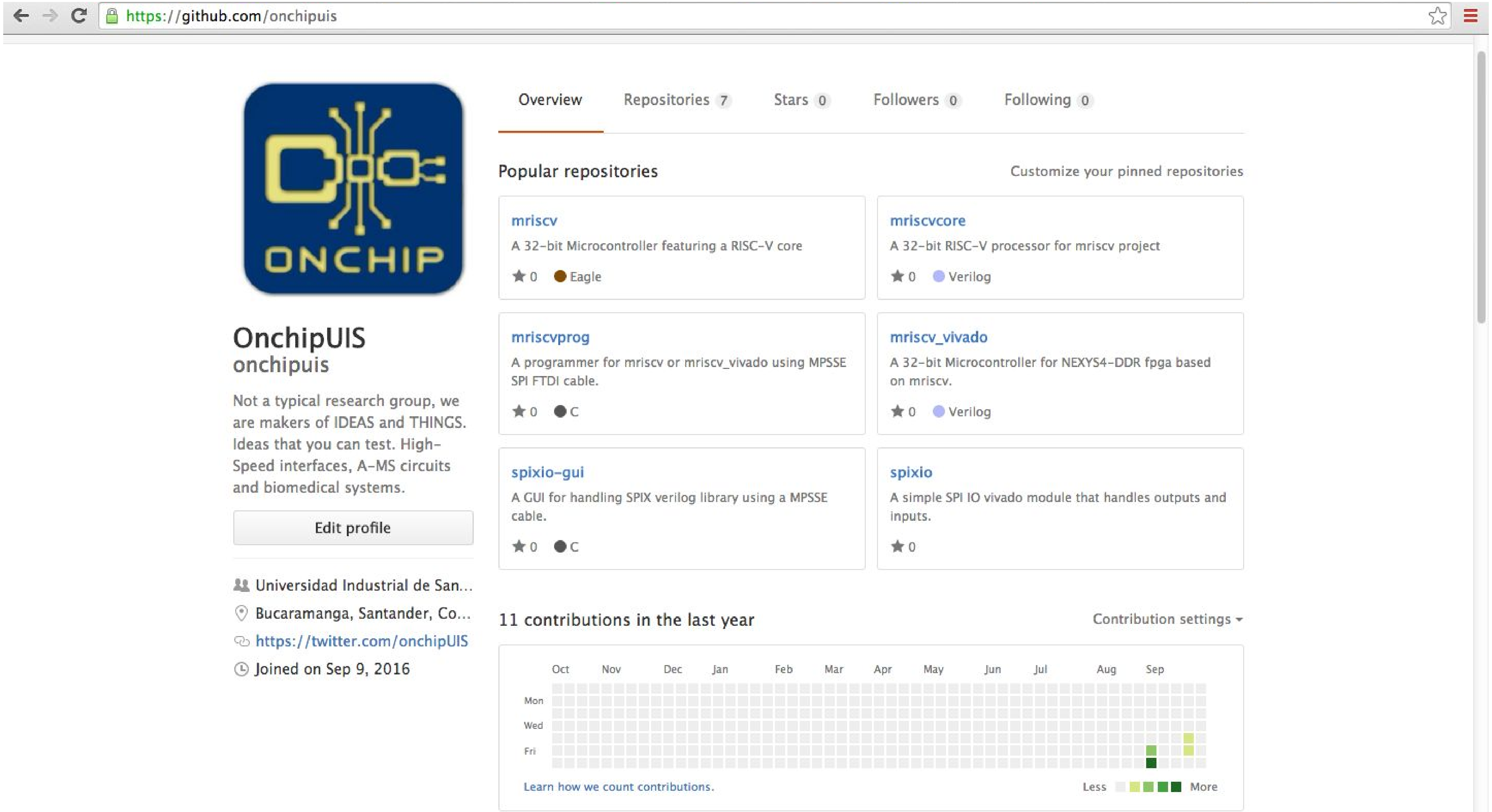
Atmel

Atmel SAM D21E / SAM D21G / SAM D21J [DATASHEET] 1027

Atmel-42181K-SAM-D21_Datasheet_Complete-09/2016

3 – On pin PA24 and PA25 the pull-up and pull-down configuration is not disabled automatically when alternative pin function is enabled.
Errata reference: 12368
Fix/Workaround:
For pin PA24 and PA25, the GPIO pull-up and pull-down must be disabled before enabling alternative functions on them.

https://github.com/onchipuis



OnchipUIS
onchipuis

Not a typical research group, we are makers of IDEAS and THINGS. Ideas that you can test. High-Speed interfaces, A-MS circuits and biomedical systems.

Universidad Industrial de San...
Bucaramanga, Santander, Co...
https://twitter.com/onchipUIS
Joined on Sep 9, 2016

Popular repositories

mriscv
A 32-bit Microcontroller featuring a RISC-V core
★ 0 ● Eagle

mriscvcore
A 32-bit RISC-V processor for mriscv project
★ 0 ● Verilog

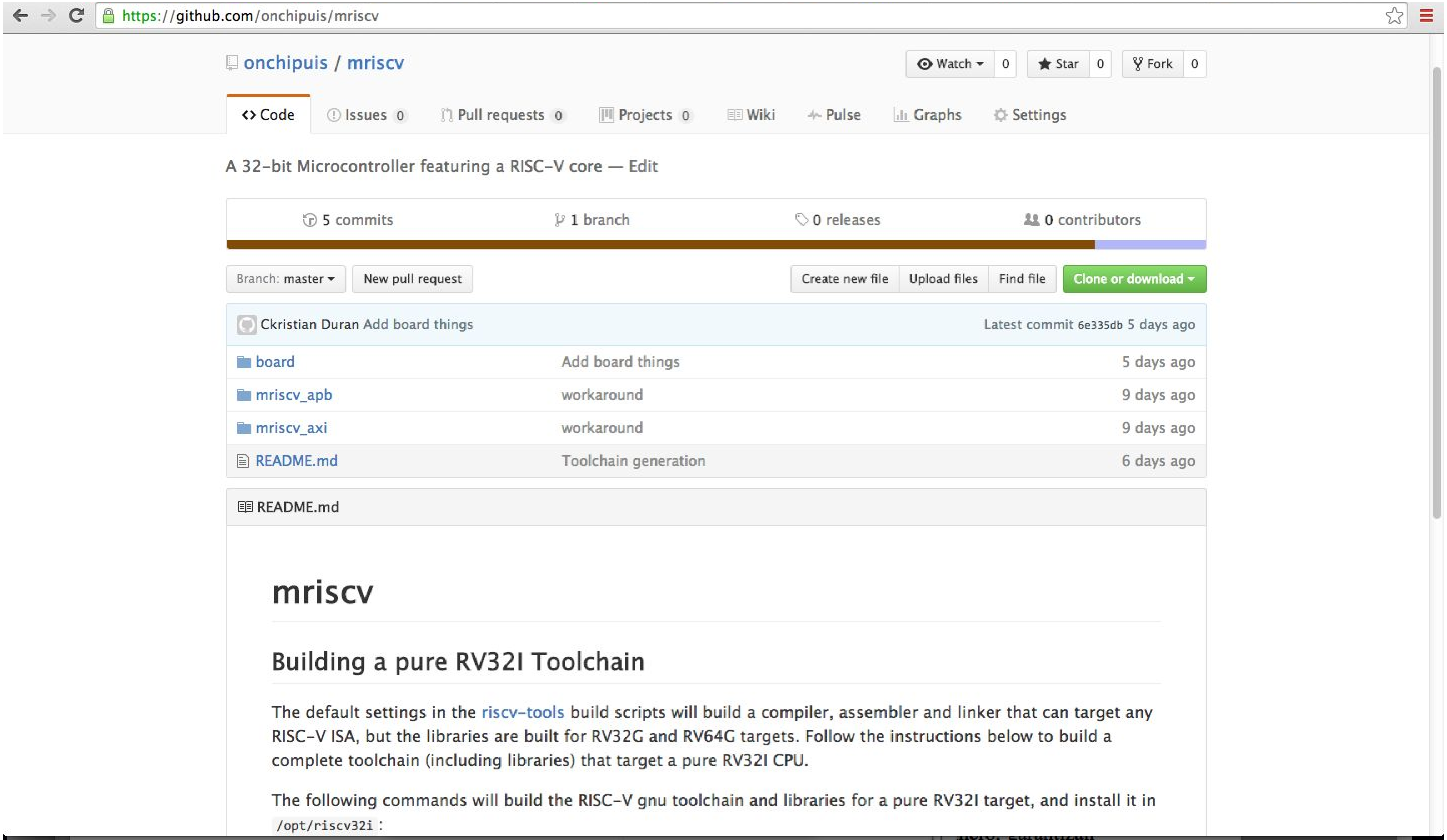
mriscvprog
A programmer for mriscv or mriscv_vivado using MPSSE SPI FTDI cable.
★ 0 ● C

mriscv_vivado
A 32-bit Microcontroller for NEXYS4-DDR fpga based on mriscv.
★ 0 ● Verilog

spixio-gui
A GUI for handling SPIX verilog library using a MPSSE cable.
★ 0 ● C

spixio
A simple SPI IO vivado module that handles outputs and inputs.
★ 0

11 contributions in the last year



A 32-bit Microcontroller featuring a RISC-V core — Edit

Ckristian Duran Add board things — Latest commit 6e335db 5 days ago

board	Add board things	5 days ago
mriscv_apb	workaround	9 days ago
mriscv_axi	workaround	9 days ago
README.md	Toolchain generation	6 days ago

README.md

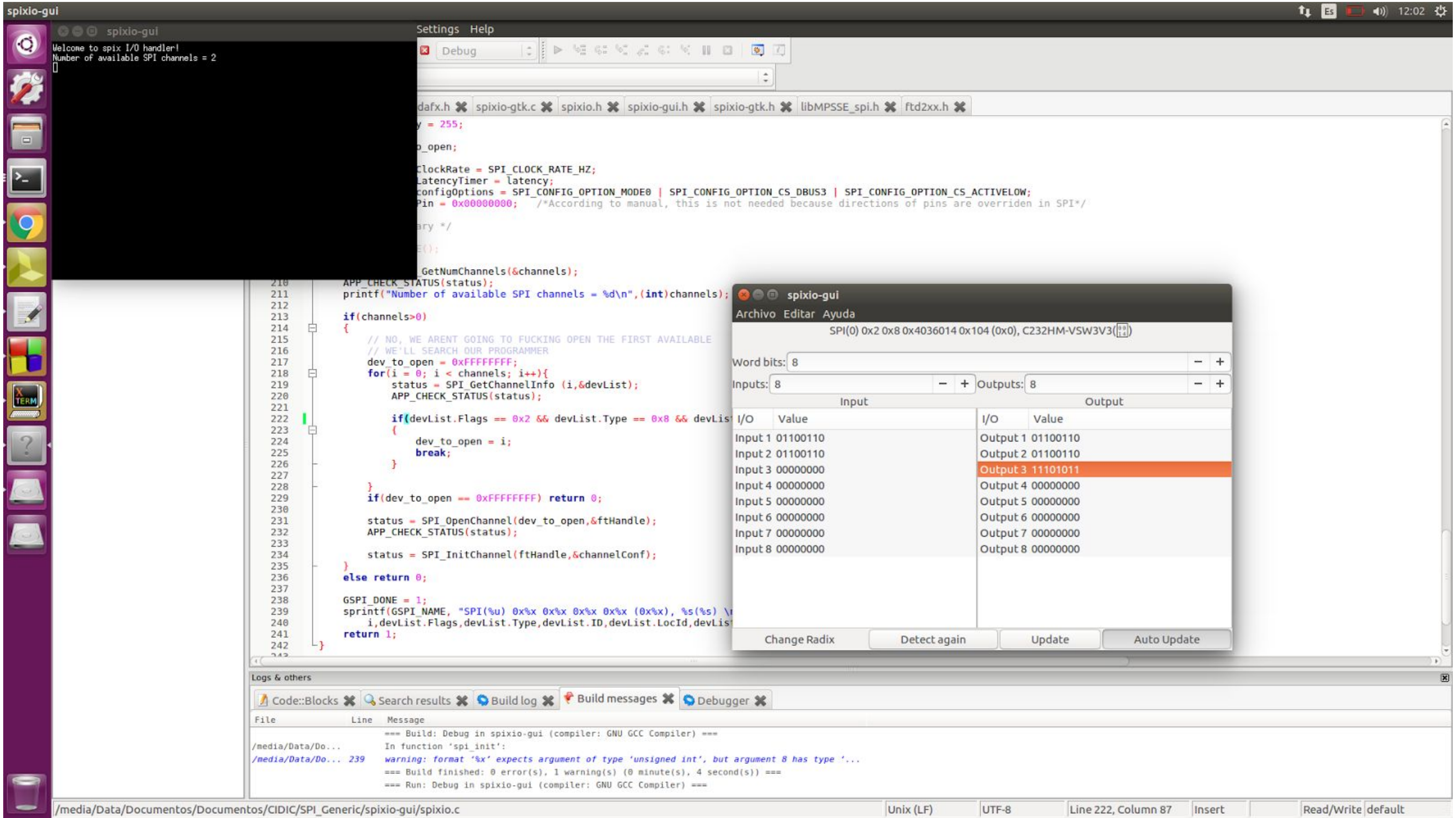
mriscv

Building a pure RV32I Toolchain

The default settings in the riscv-tools build scripts will build a compiler, assembler and linker that can target any RISC-V ISA, but the libraries are built for RV32G and RV64G targets. Follow the instructions below to build a complete toolchain (including libraries) that target a pure RV32I CPU.

The following commands will build the RISC-V gnu toolchain and libraries for a pure RV32I target, and install it in `/opt/riscv32i` :

Tools for VLSI grad students



```
Welcome to spix I/O handler!
Number of available SPI channels = 2
```

```
y = 255;

o_open;

ClockRate = SPI_CLOCK_RATE_HZ;
LatencyTimer = latency;
configOptions = SPI_CONFIG_OPTION_MODE0 | SPI_CONFIG_OPTION_CS_DBUS3 | SPI_CONFIG_OPTION_CS_ACTIVELOW;
Pin = 0x00000000;   /*According to manual, this is not needed because directions of pins are overriden in SPI*/

ary */

E();

 GetNumChannels(&channels);
210      APP_CHECK_STATUS(status);
211      printf("Number of available SPI channels = %d\n",(int)channels);
212
213      if(channels>0)
214      {
215          // NO, WE ARENT GOING TO FUCKING OPEN THE FIRST AVAILABLE
216          // WE'LL SEARCH OUR PROGRAMMER
217          dev_to_open = 0xFFFFFFFF;
218          for(i = 0; i < channels; i++){
219              status = SPI_GetChannelInfo (i,&devList);
220              APP_CHECK_STATUS(status);
221
222              if(devList.Flags == 0x2 && devList.Type == 0x8 && devLis
223              {
224                  dev_to_open = i;
225                  break;
226              }
227
228          }
229          if(dev_to_open == 0xFFFFFFFF) return 0;
230
231          status = SPI_OpenChannel(dev_to_open,&ftHandle);
232          APP_CHECK_STATUS(status);
233
234          status = SPI_InitChannel(ftHandle,&channelConf);
235      }
236      else return 0;
237
238      GSPI_DONE = 1;
239      sprintf(GSPI_NAME, "SPI(%u) 0x%x 0x%x 0x%x 0x%x (0x%x), %s(%s) \
240          i,devList.Flags,devList.Type,devList.ID,devList.LocId,devLis
241      return 1;
242  }
```

spixio-gui

Archivo Editar Ayuda

SPI(0) 0x2 0x8 0x4036014 0x104 (0x0), C232HM-VSW3V3(⍰)

Word bits: 8 − +

Inputs: 8 − + Outputs: 8 − +

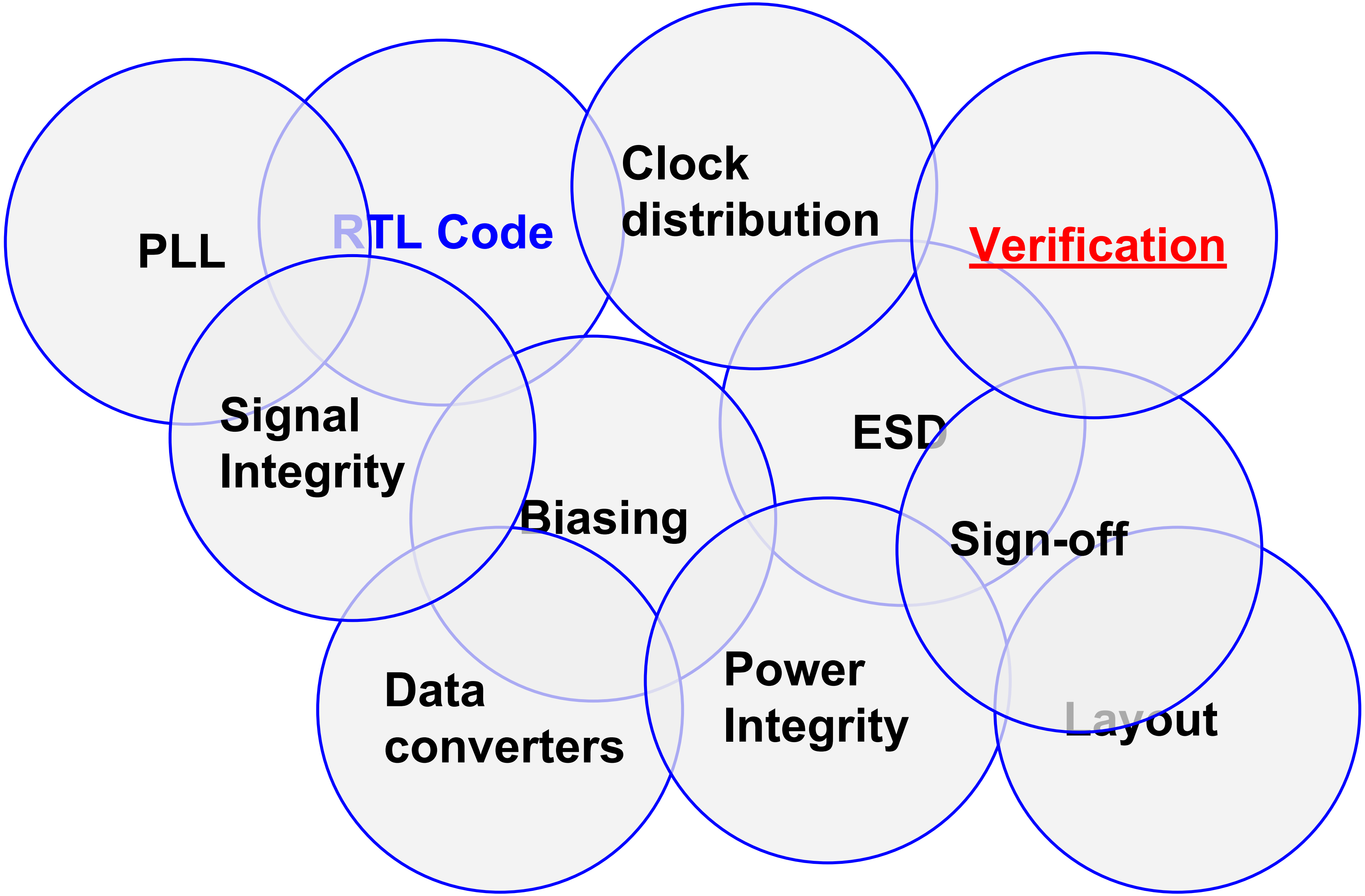
Input		Output	
I/O	Value	I/O	Value
Input 1	01100110	Output 1	01100110
Input 2	01100110	Output 2	01100110
Input 3	00000000	Output 3	11101011
Input 4	00000000	Output 4	00000000
Input 5	00000000	Output 5	00000000
Input 6	00000000	Output 6	00000000
Input 7	00000000	Output 7	00000000
Input 8	00000000	Output 8	00000000

Change Radix | Detect again | Update | Auto Update

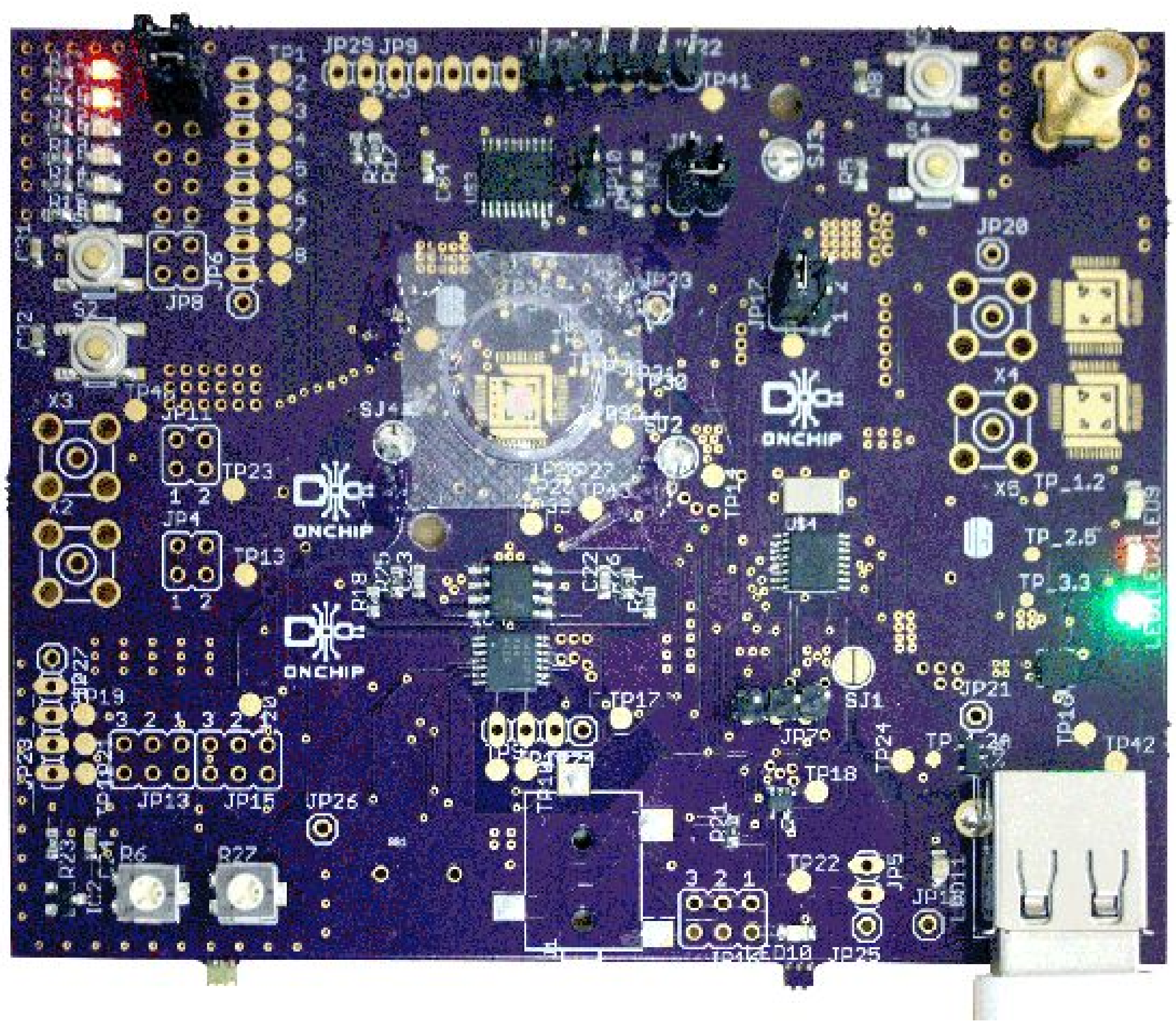
```
File                 Line  Message
                           === Build: Debug in spixio-gui (compiler: GNU GCC Compiler) ===
/media/Data/Do...          In function ‘spi_init’:
/media/Data/Do...    239   warning: format ‘%x’ expects argument of type ‘unsigned int’, but argument 8 has type ‘...
                           === Build finished: 0 error(s), 1 warning(s) (0 minute(s), 4 second(s)) ===
                           === Run: Debug in spixio-gui (compiler: GNU GCC Compiler) ===
```

/media/Data/Documentos/Documentos/CIDIC/SPI_Generic/spixio-gui/spixio.c | Unix (LF) | UTF-8 | Line 222, Column 87 | Insert | Read/Write | default

Lot of circuits, lot of skills

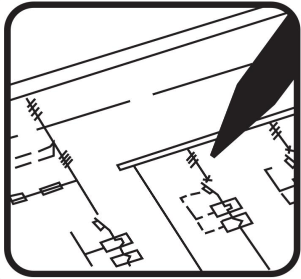


Diving into...

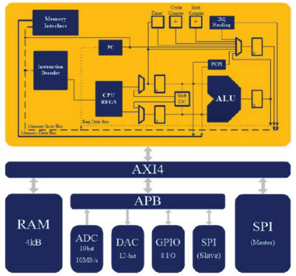


ONCHIP
ONCHIP
ONCHIP
TP1
JP29
JP9
TP41
S4
JP20
X4
X5
TP_1.2
TP_2.5
TP_3.3
S2
JP8
X3
X2
JP11
TP23
JP4
TP13
SJ4
SJ2
SJ1
JP17
TP17
JP7
TP18
TP24
JP21
TP42
TP22
JP25
D10
JP13
JP15
JP26
R6
R27
TP19

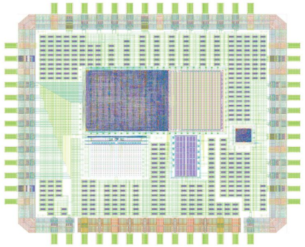
Some more details



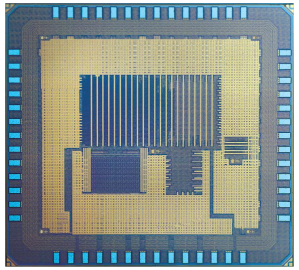
Q2 2015
Microcontroller Specs



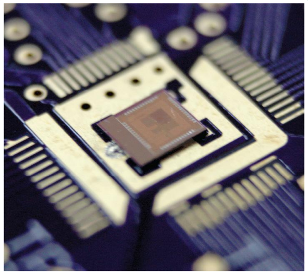
Q3-Q4 2015
RTL + Analog Design



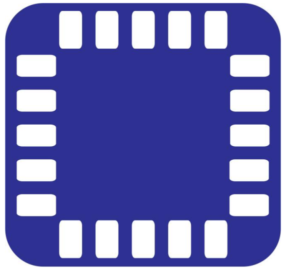
Q4 2015
Tape Out



Q2 2016
Chip is back



Q3 2016
Testing

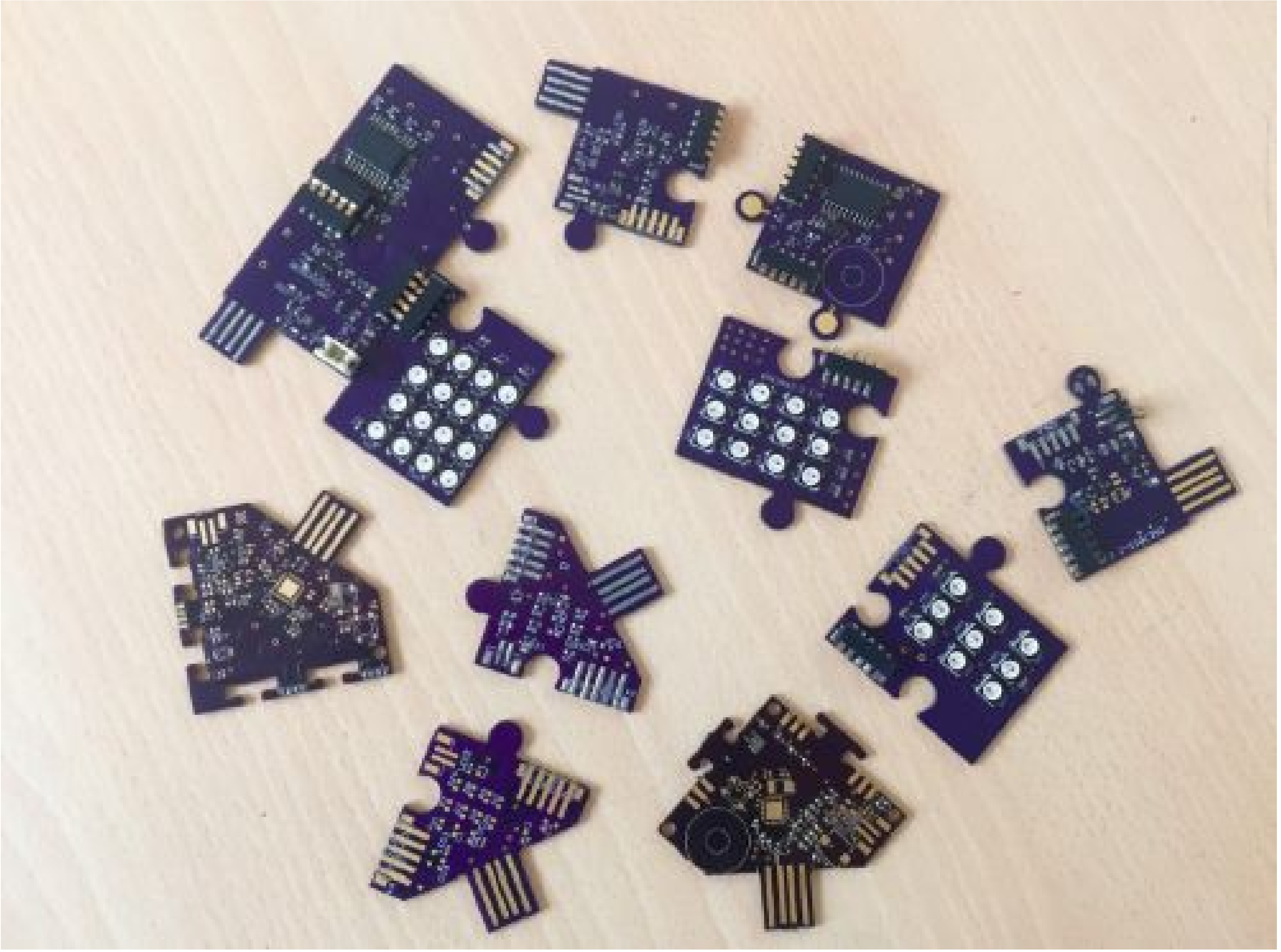


Q1 2017
Tape Out V.2
+CNN +NVRAM

Q2 2018
Puzzle Boards

A New Kind of Market

Logo programming →
New programming model



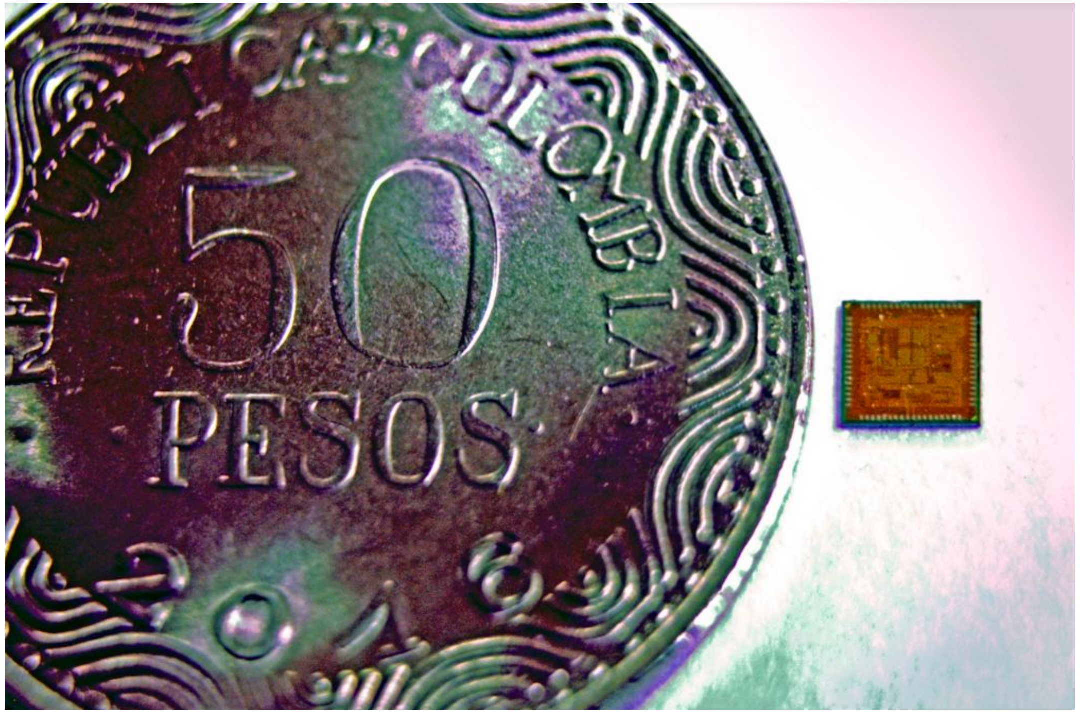
Thanks to:

- SAMPA Project
- ALICE-CERN
- Clifford Wolf
- RISC-V foundation
- Oshpark
- TPT

What else we are doing:

- LPDDR3/LPDDR4 PCS + PMA
- USB 3.1 PCS + PMA (10Gb/s)
- LPDDR3/4 UVM VIP
- Circuits for Security: Lite key exchanging, TRNGs
- Research on CDR architectures
- Research on lite CNN
- Research on DFE + on-fly offset correction

New chip has just arrived!



Funding is welcome

- Kickstarter microcontroller for a try to replace Atmel SAMD21? → $350k for 70k packaged and tested chips, 2 revisions!
- Share MPW costs with more teams. **Any school or private group want to team up?**
- Foundries should start looking at these initiatives and fund “free” prototype silicon. **High-level contacts are welcome.**
- USB 2.0/3.1 PHY effort on indiegogo?
- Maybe RasberryPi or Arduino guys are willing to get some really open silicon?

IDEAS ARE WELCOME

Demo before dinner, tonight!