Spacely + Caribou on the ZCU102

Adam Quinn

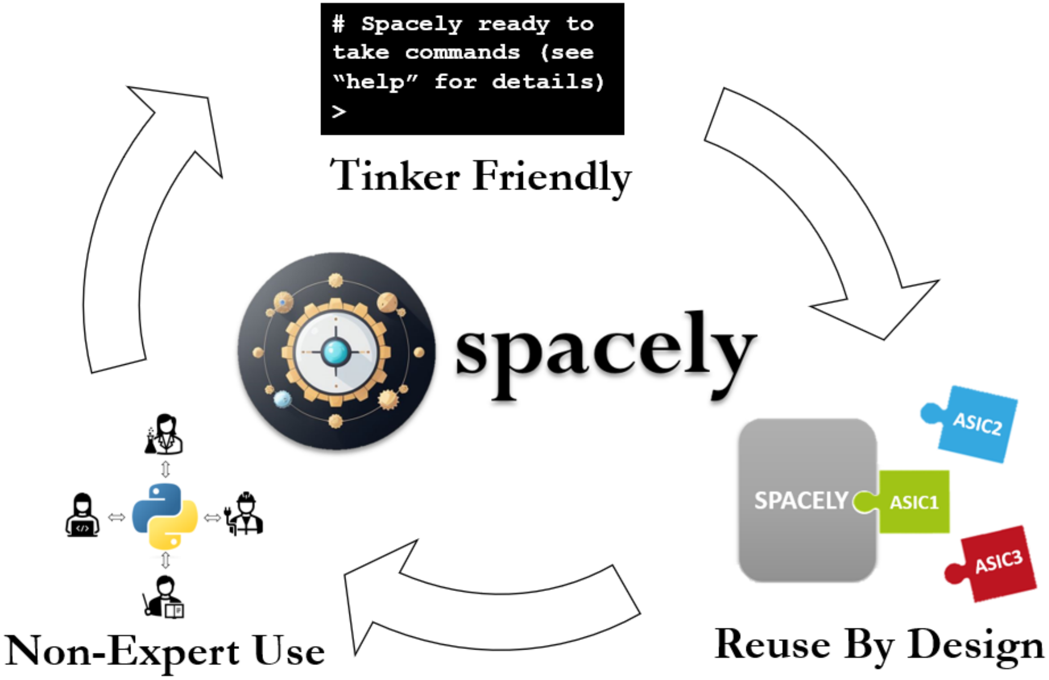
May 14, 2024

Agenda

- Brief Overview of Spacely
- Spacely-Caribou
  - Work done to make Caribou Compatible w/ ZCU102

Spacely

- Spacely is an **open-source, Python based** test automation framework
- Spacely consists of a core Python repository, supplemented by **instrument libraries, firmware**, and recommended **design flows**.
- Spacely is targeted toward small ASIC design teams in **research/academia**, and non-expert (scientific) users.

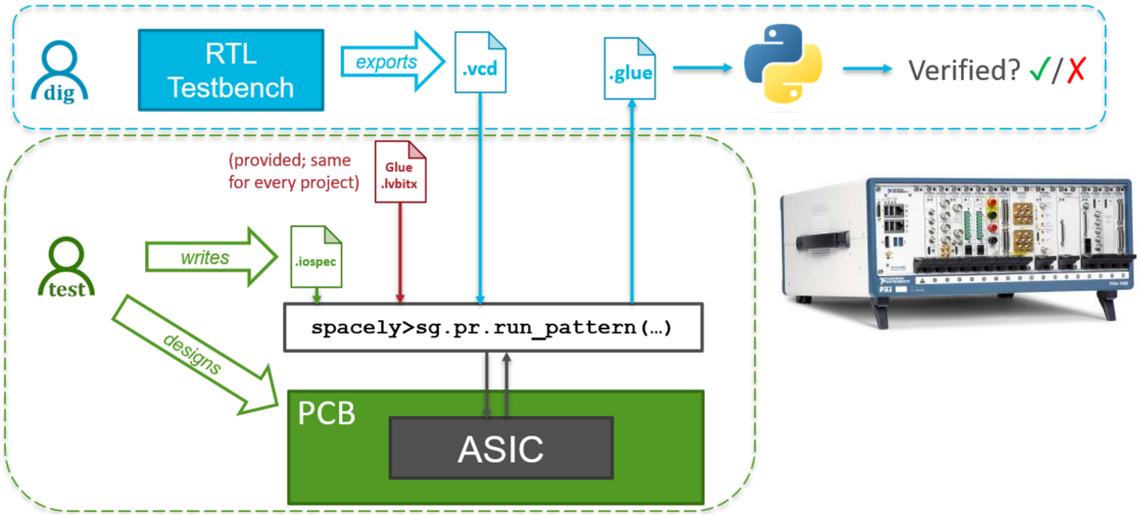


```
# Spacely ready to
take commands (see
"help" for details)
>
```

Go from Verilog to the Lab with Spacely

- Export VCD waveforms from your verification testbench and use them directly as test stimulus for your ASIC.
- Expert digital ASIC and test engineers can both participate directly in test, reducing miscommunication.

spacely



dig

RTL Testbench

exports

.vcd

.glue

Verified? ✓/✗

(provided; same for every project)

Glue .lvbitx

test

writes

.iospec

designs

```
spacely>sg.pr.run_pattern(…)
```

PCB

ASIC

Control Test Instruments with Spacely

- Spacely abstracts interfaces to test equipment
- Python test routines read like natural language; easy for scientific stakeholders to participate.

```
def ROUTINE_run_adc_sweep():
    """Test ADC while sweeping input pulse magnitude."""


    V_PORT["Vbias"].set_voltage(1.2)

    for pulse_mag_V in [0.1, 0.2, 0.3]

        sg.INSTR["awg"].set_pulse_mag(pulse_mag_V)

        result = sg.pr.run_pattern(genpattern_run_ADC())

        result_glue = sg.gc.read_glue(result)
        dout = sg.gc.get_bitstream(result_glue,"dout")

        print(f"Vin: {pulse_mag_V} Result: {parse_dout(dout)}")
```

VOLT:OFFS 1.2

PULS:MAG 0.1

<i/p vector>

<o/p vector>

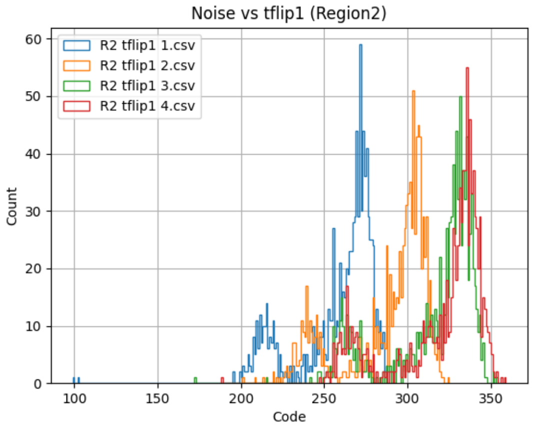
Analyze Data with Spacely

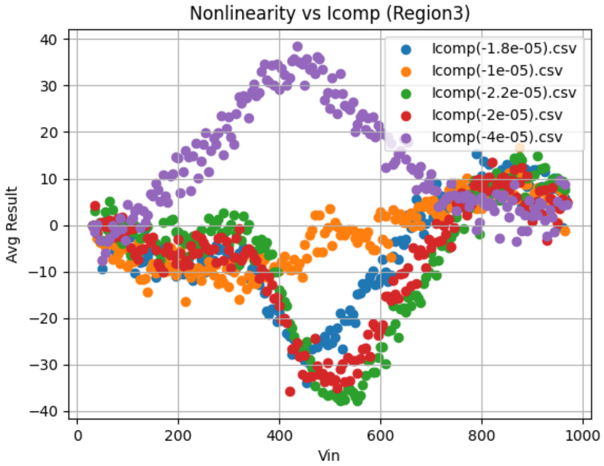
- Group data into **Experiments** and **DataFiles**, each with tagged **Metadata** corresponding to the Python variables used in the Spacely routine, which is automatically written to file.

- Calculate statistics and generate scatter plots / histograms natively in Spacely.

spacely

Ibdig(0.2).csv
Ibdig(0.3).csv
Ibdig(0.4).csv
Ibdig(0.5).csv
Ibdig(0.6).csv
Ibdig(0.7).csv
meta.txt
Nonlinearity_vs_Ibdig.png
Transfer_Function_vs_Ibdig.png

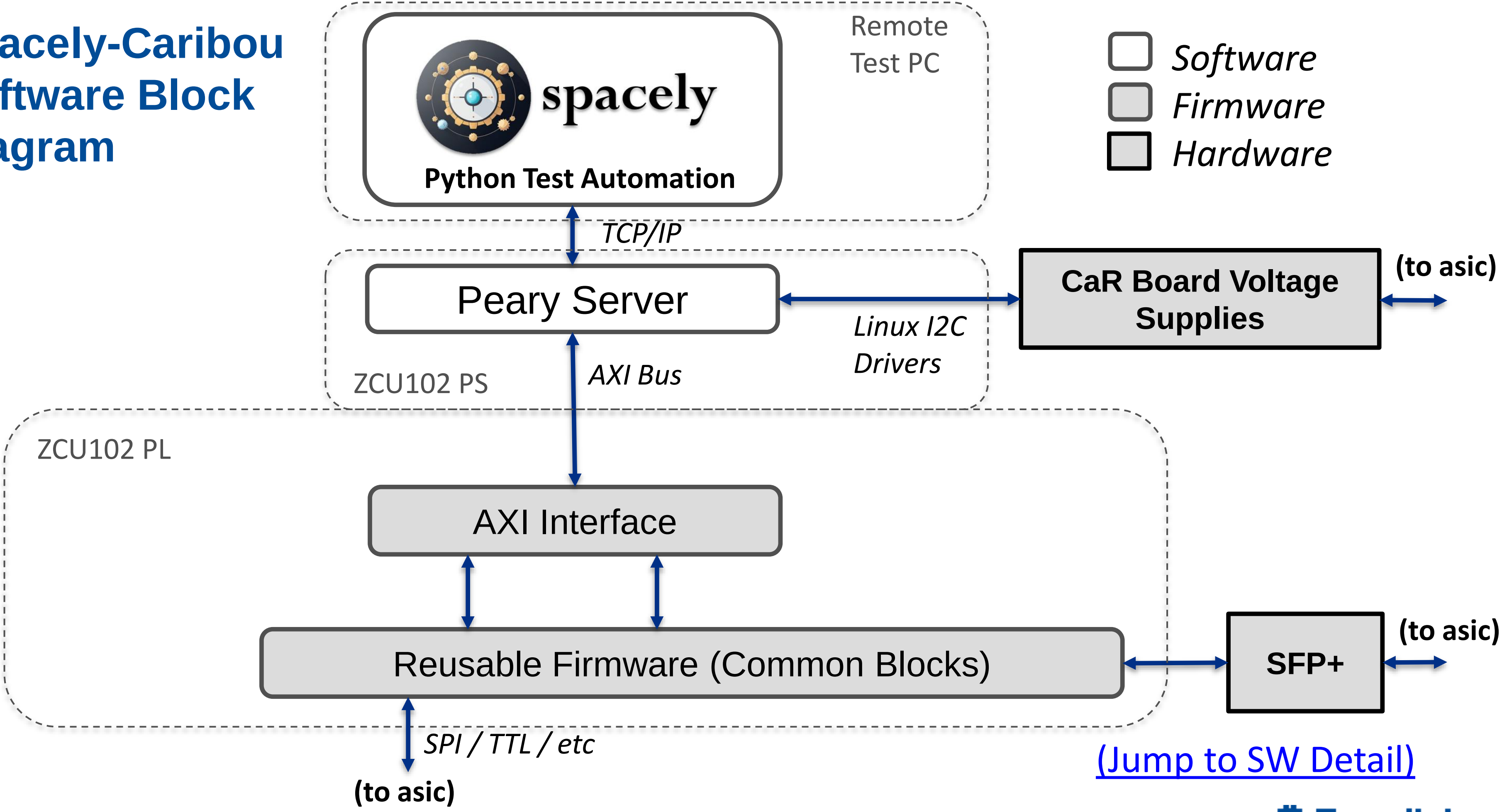




What is Spacely-Caribou?

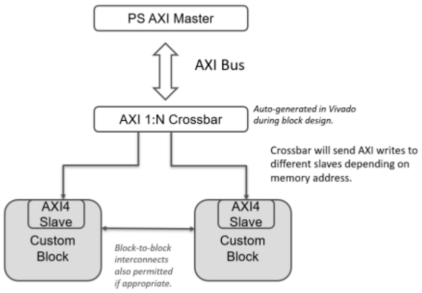
- **Spacely-Caribou** is a test strategy which uses the **Spacely** test automation framework to write Python tests which are carried out using **Caribou** test hardware and firmware.
- The primary aim is to leverage Caribou infrastructure, while shifting as much of the ASIC-specific development into the high-productivity Python environment.

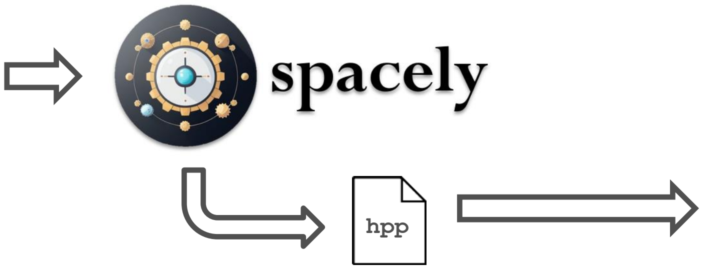
Spacely-Caribou Software Block Diagram

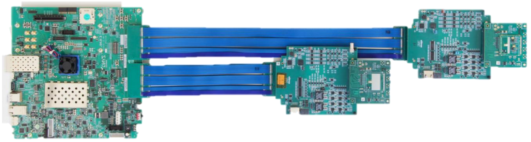


(Jump to SW Detail)

Caribou Usage Model w/ Spacely


PS AXI Master
AXI Bus
AXI 1:N Crossbar
Auto-generated in Vivado during block design.
Crossbar will send AXI writes to different slaves depending on memory address.
AXI4 Slave
Custom Block
Block-to-block interconnects also permitted if appropriate.
AXI4 Slave
Custom Block

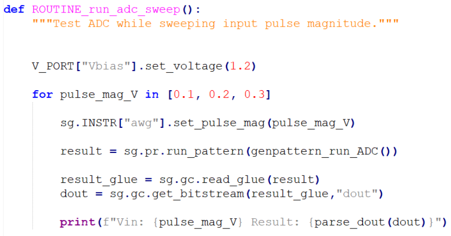




(1) Build a firmware design in Vivado by connecting pre-made AXI blocks (no RTL experience required unless you need custom blocks)

(2) Spacely generates a C++ style AXI memory map, which you include in the SpacelyCaribouBasic peary device. (All other parts of the device are generic.)

(3) Flash your ZCU102 with the provided BOOT.BIN, and build Peary with the SpacelyCaribouBasic device.



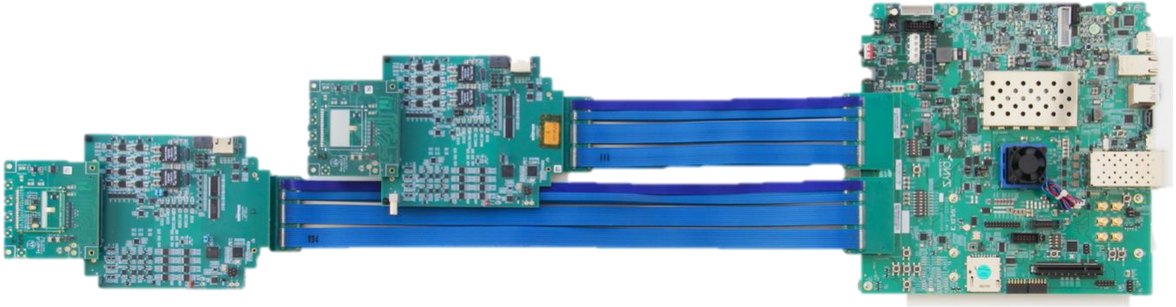
```
def ROUTINE_run_adc_sweep():
    """Test ADC while sweeping input pulse magnitude."""


    V_PORT["Vbias"].set_voltage(1.2)

    for pulse_mag_V in [0.1, 0.2, 0.3]

        sg.INSTR["awg"].set_pulse_mag(pulse_mag_V)

        result = sg.pr.run_pattern(genpattern_run_ADC())

        result_glue = sg.gc.read_glue(result)
        dout = sg.gc.get_bitstream(result_glue,"dout")

        print(f"Vin: {pulse_mag_V} Result: {parse_dout(dout)}")
```

(4) Write your test routines in Python with commands like caribou.get_memory() and *V_PORT["VDD"].set_voltage()*

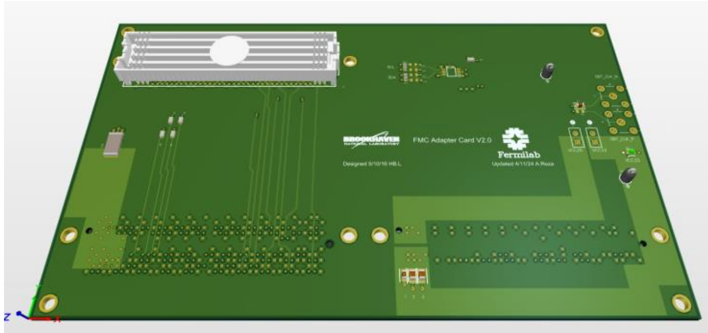
(5) Do Science!

How we Ported Caribou to ZCU102

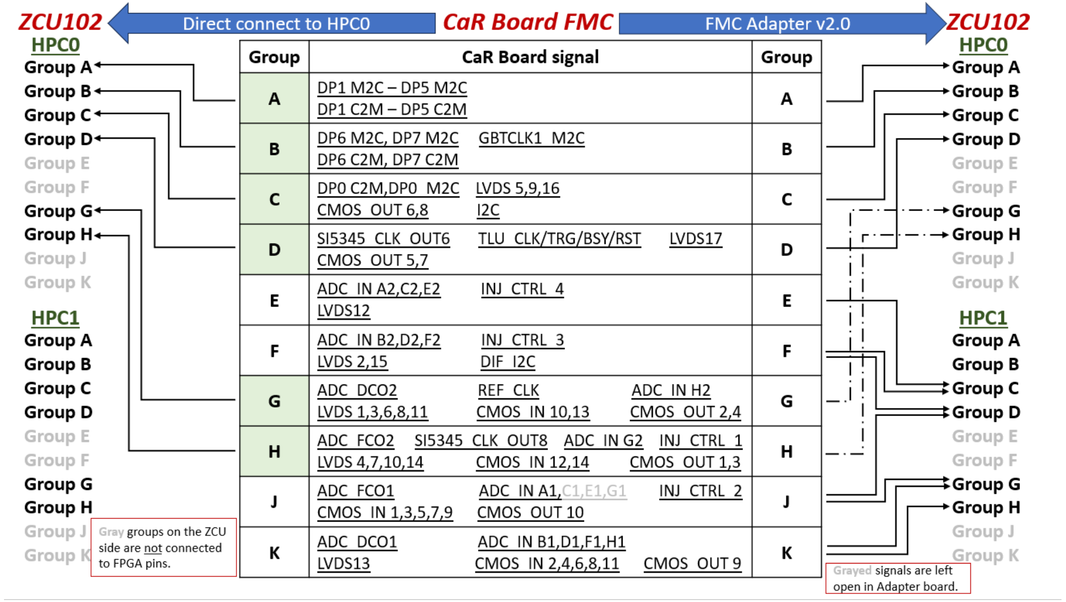
1. Software: Minor updates to Peary
2. OS: Petalinux
3. Hardware: ZCU102 Mezzanine Board



Hardware: ZCU102 Mezzanine Board



- Modification of the BNL ZC706 Mezzanine Board
- Minor component updates and signal re-routing.
- Note: A small number of connections (ADC_IN C1, E1, and G1) were sacrificed due to insufficient pinout.
- Status: **Manufacturing**



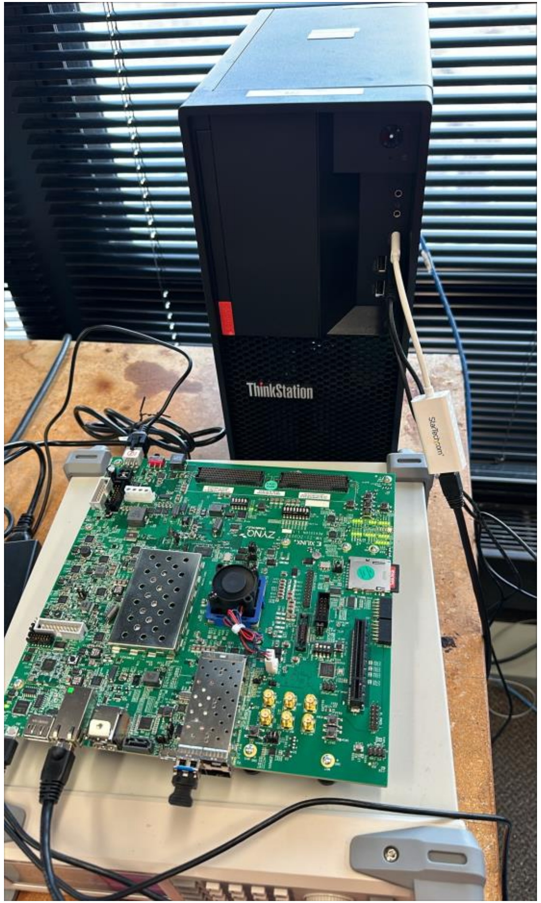
Group	CaR Board signal	Group
A	DP1 M2C – DP5 M2C DP1 C2M – DP5 C2M	A
B	DP6 M2C, DP7 M2C GBTCLK1_M2C DP6 C2M, DP7 C2M	B
C	DP0 C2M,DP0_M2C LVDS 5,9,16 CMOS_OUT 6,8 I2C	C
D	SI5345_CLK_OUT6 TLU_CLK/TRG/BSY/RST LVDS17 CMOS_OUT 5,7	D
E	ADC_IN A2,C2,E2 INJ_CTRL_4 LVDS12	E
F	ADC_IN B2,D2,F2 INJ_CTRL_3 LVDS 2,15 DIF_I2C	F
G	ADC_DCO2 REF_CLK ADC_IN H2 LVDS 1,3,6,8,11 CMOS_IN 10,13 CMOS_OUT 2,4	G
H	ADC_FCO2 SI5345_CLK_OUT8 ADC_IN G2 INJ_CTRL_1 LVDS 4,7,10,14 CMOS_IN 12,14 CMOS_OUT 1,3	H
J	ADC_FCO1 ADC_IN A1,C1,E1,G1 INJ_CTRL_2 CMOS_IN 1,3,5,7,9 CMOS_OUT 10	J
K	ADC_DCO1 ADC_IN B1,D1,F1,H1 LVDS13 CMOS_IN 2,4,6,8,11 CMOS_OUT 9	K

Gray groups on the ZCU side are not connected to FPGA pins.

Grayed signals are left open in Adapter board.

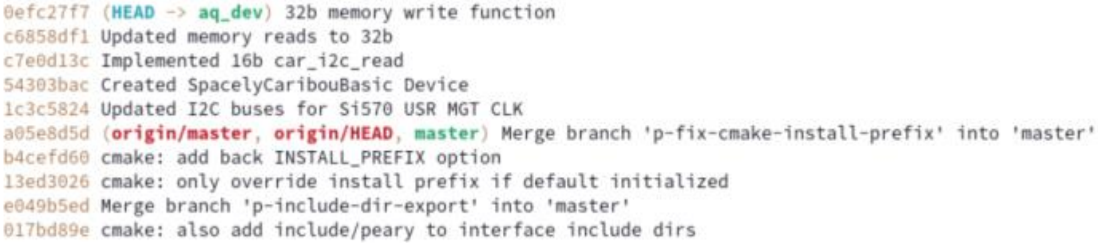
OS: Petalinux

- Built a basic image for the ZCU102 using Petalinux tools.
- Included a very basic FW image which maps 4 GB of AXI address space (0x400000000 to 0x4FFFFFFFF) so that it is addressable from Peary.
- Edited the Device Tree to include CaR board I2C multiplexers
- Booted the ZCU102 from SD card.



Software: Minor Updates to Peary

- Updated CaR board I2C addresses to match the ZCU102 Device Tree
  - Added car_i2c_read/write functions that can be accessed from pearyd for flexibility.
- Created “**SpacelyCaribouBasic**” Device
  - Keeping with Spacely-Caribou philosophy, this “device” should be sufficient for most ASICs.
  - Only the memory map must change, and this can be autogenerated from Spacely.
- Small update to Memory interface to make AXI reads 32b wide independent of platform.



```
0efc27f7 (HEAD -> aq_dev) 32b memory write function
c6858df1 Updated memory reads to 32b
c7e0d13c Implemented 16b car_i2c_read
54303bac Created SpacelyCaribouBasic Device
1c3c5824 Updated I2C buses for Si570 USR MGT CLK
a05e8d5d (origin/master, origin/HEAD, master) Merge branch 'p-fix-cmake-install-prefix' into 'master'
b4cefd60 cmake: add back INSTALL_PREFIX option
13ed3026 cmake: only override install prefix if default initialized
e049b5ed Merge branch 'p-include-dir-export' into 'master'
017bd89e cmake: also add include/peary to interface include dirs
```

Caveat: I am not a C++ developer. Pull from my branch at your risk & expect to see some warnings about integer casting… ☺

Let's work together!

Spacely is open-source: https://github.com/SpacelyProject/spacely-docs

-- Includes common firmware blocks for ZCU102 PL

Happy to push my aq_dev branch to the Peary Gitlab (need permission)

Other things we'd be happy to share (need to figure out the best way):

- Boot Image for Petalinux on the ZCU102
- Design for the ZCU102 Mezzanine Board

Possible Targets for Collaboration:

- Documentation / Guide on setting up Caribou + Peary