The Leuschner Spectrometer

Rachel Domagalski

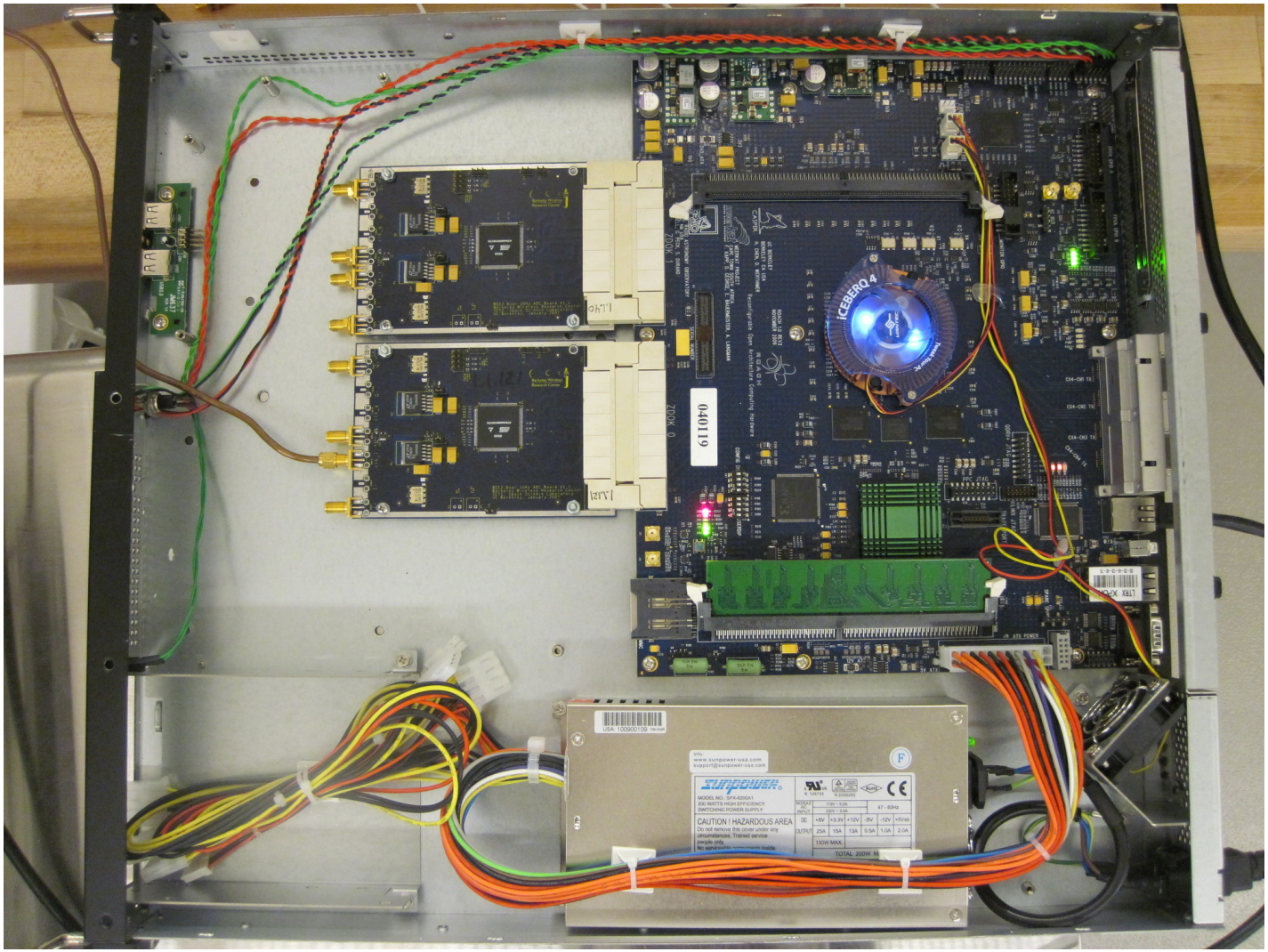
University of California, Berkeley

Hardware

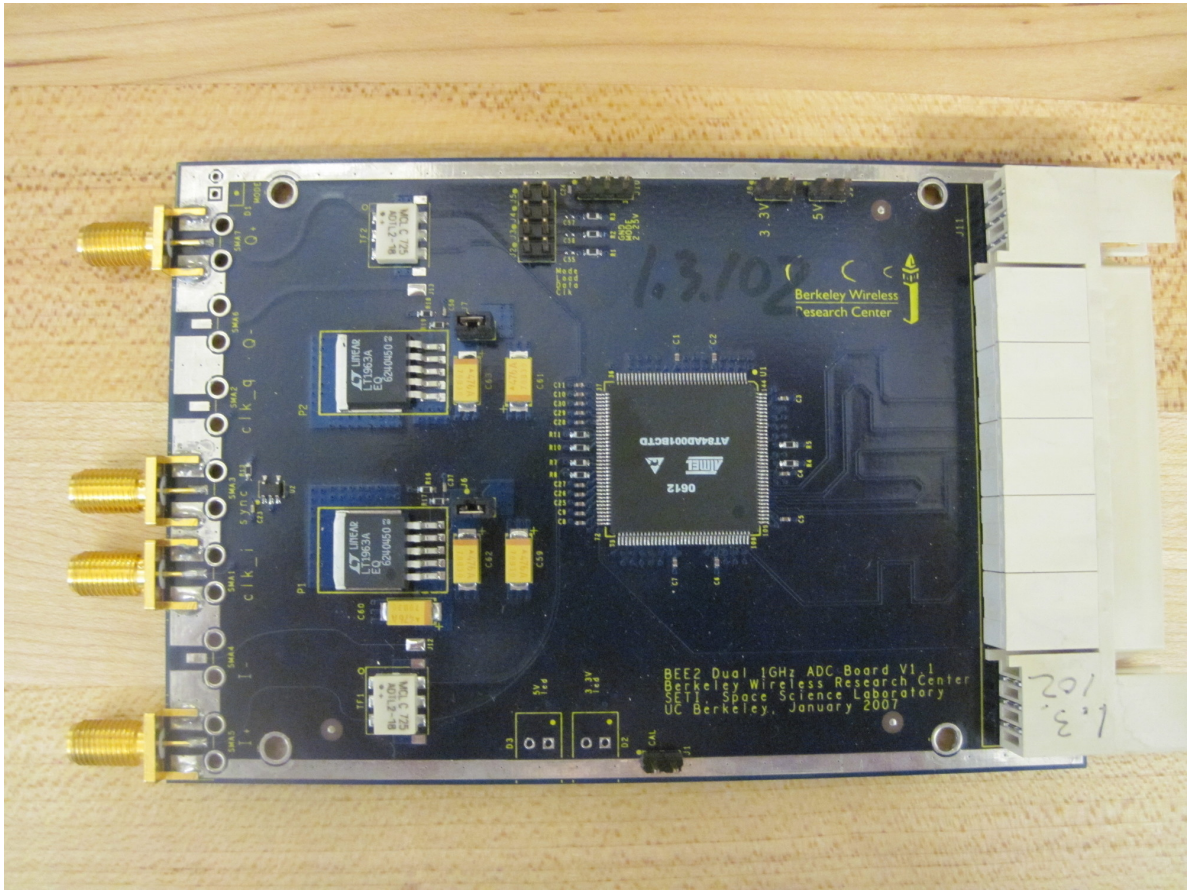
The FPGA processor

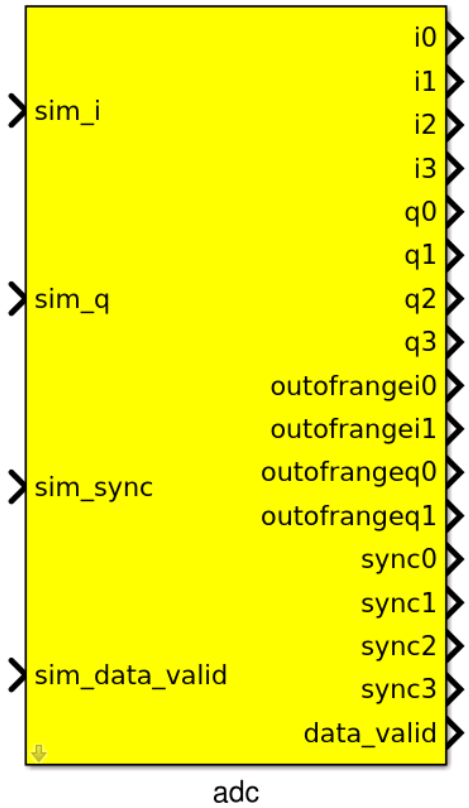
- Xilinx Virtex-5 XC5VSX95T-1FF1136 FPGA
- Can be reconfigured on the fly.
- FPGA programs are like circuit designs.
- Can run many subroutines in parallel on the same chip.
- Has a minimum clocking speed of 120 MHz.
- A ROACH is used as the FPGA processing board.

Hardware



Hardware


Berkeley Wireless
Research Center
BEE2 Dual 1GHz ADC Board V1.1
Berkeley Wireless Research Center
SETI Space Science Laboratory
UC Berkeley, January 2007

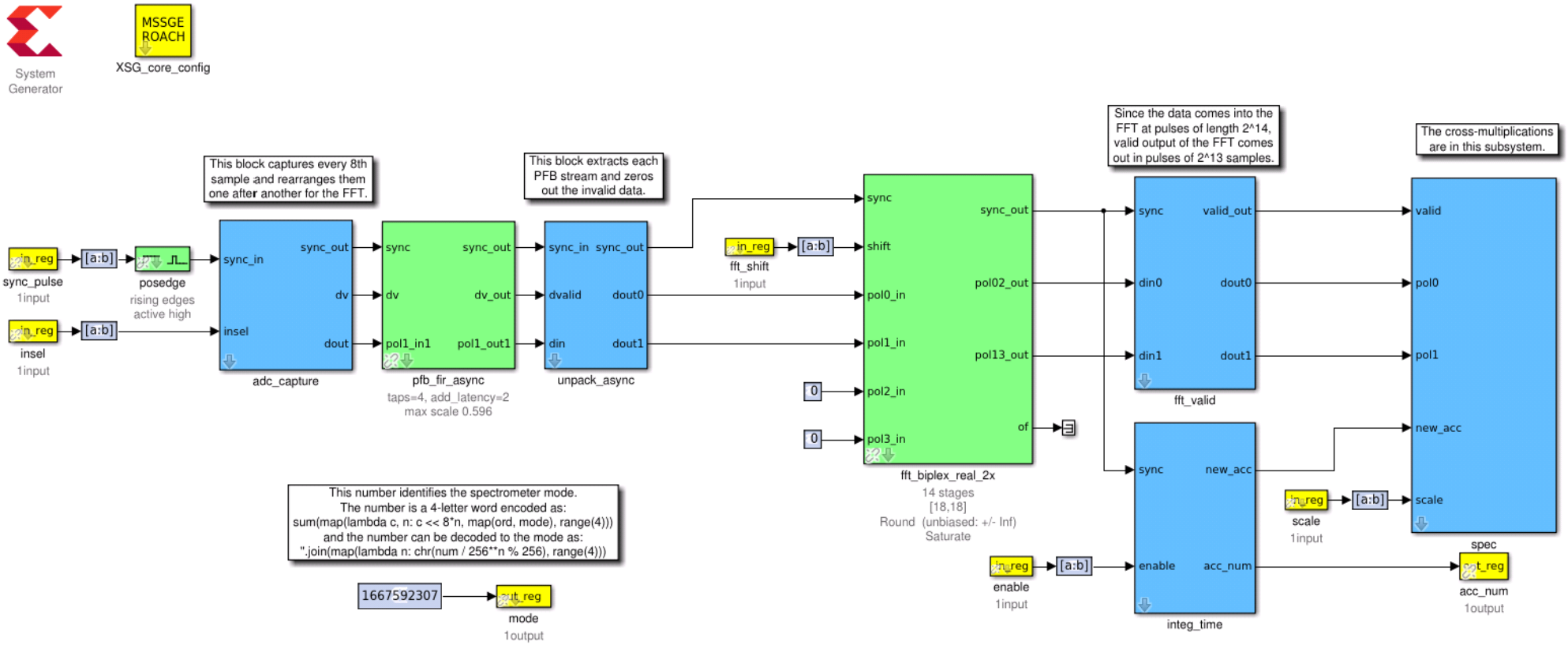

sim_i
sim_q
sim_sync
sim_data_valid
i0
i1
i2
i3
q0
q1
q2
q3
outofrangei0
outofrangei1
outofrangeq0
outofrangeq1
sync0
sync1
sync2
sync3
data_valid
adc

The Spectrometer

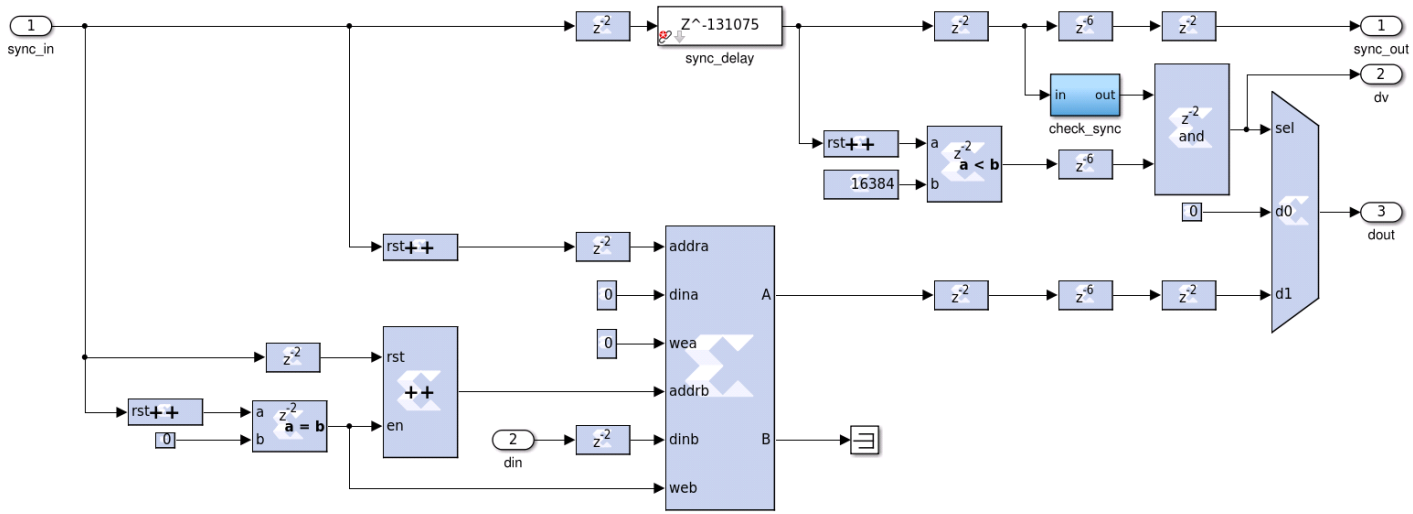
Overview:

- Created using Simulink.
- 12 MHz bandwidth.
- 8192 frequency channels ($\Delta f \approx 1.5\,\text{kHz}$).
- Two polarization spectrometer with cross-correlation.
- Runs on a ROACH board.
- Interfaces with a computer using Python.

The Spectrometer

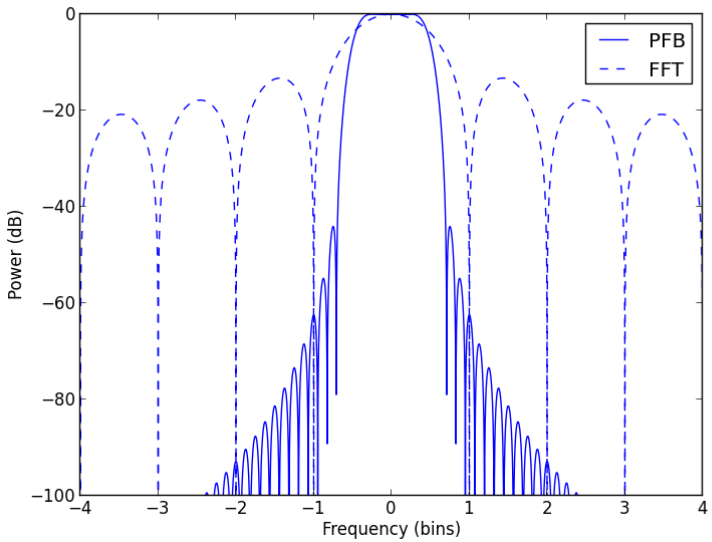


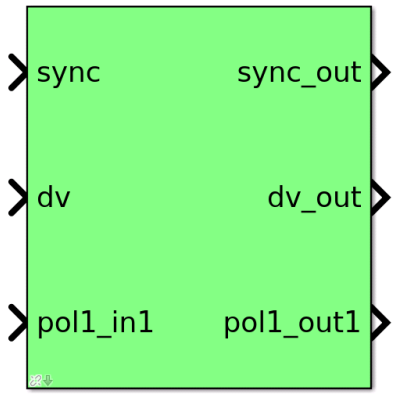
Downsampling the ADC



- The Virtex 5 has a minimum clocking speed of 120 MHz.
- A 24 MHz sampling rate can be achieved by downsampling 192 MHz by a factor of 8.

The Polyphase Filter Bank Block

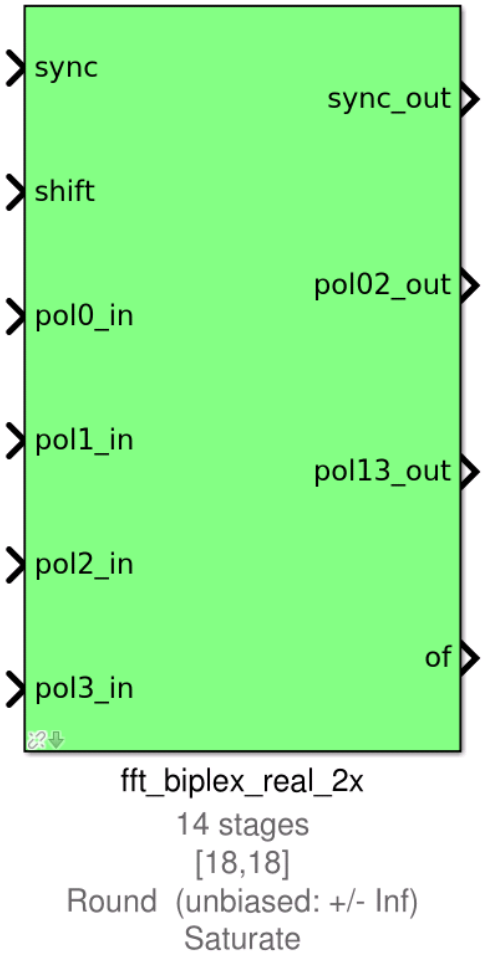




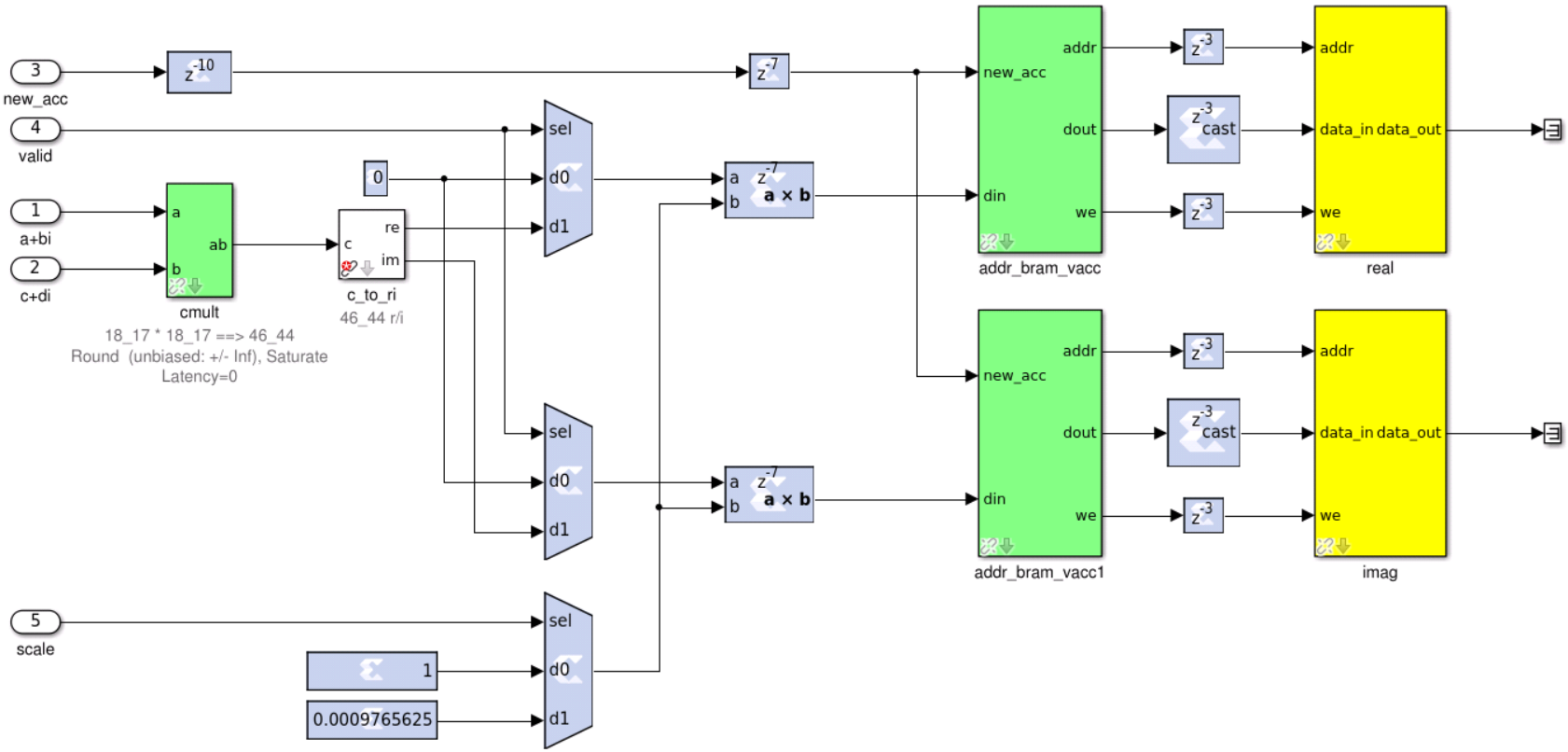
$$y(n) = \sum_{p=0}^{P-1} x(n+pN)\, h(n+pN)$$

The Biplex Real-FFT Block

- Radix-2 Cooley-Tukey FFT.
- Internal design is similar to an FFT butterfly diagram.
- Can process four real signal channels in parallel.
- Can output two spectra along the same channel.
- The `shift` input port is used to help prevent overflow.



Cross-Correlation



The Software Interface

Software communication with the ROACH in Python:

```
>>> import leuschner
>>> spec = leuscher.Spectrometer('10.0.1.2')
>>> spec.read_spec('hydrogen.fits', 100, (210, 0))
```

- Communication with the ROACH is done in Python using a KATCP wrapper in the `corr` module.
- The spectrometer receiver code is written as a Python class that can be used in Python scripts or from the shell.
- IDL or any other language can be used to wrap the Python spectrometer receiver code.

Documentation

More information and documentation can be found here:

- General documentation:
  `http://astro.berkeley.edu/~domagalski/leuschner-radio/`
- Source code and Simulink designs:
  `https://github.com/domagalski/leuschner-spectrometer/`
- More information on the hardware/software used:
  `https://casper.berkeley.edu/wiki/Main_Page`