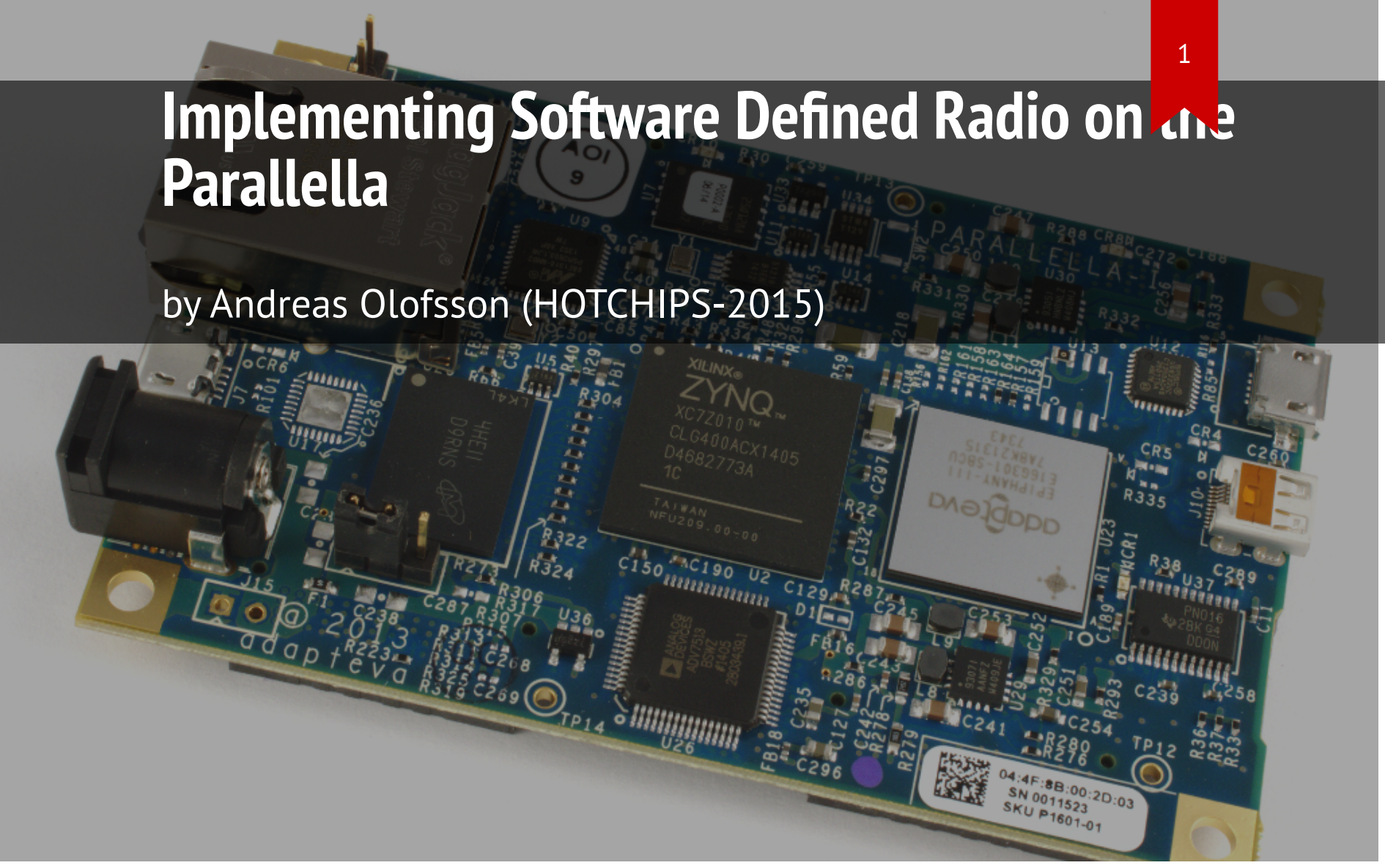
Implementing Software Defined Radio on the Parallella

by Andreas Olofsson (HOTCHIPS-2015)



What is Software Defined Radio? (SDR)

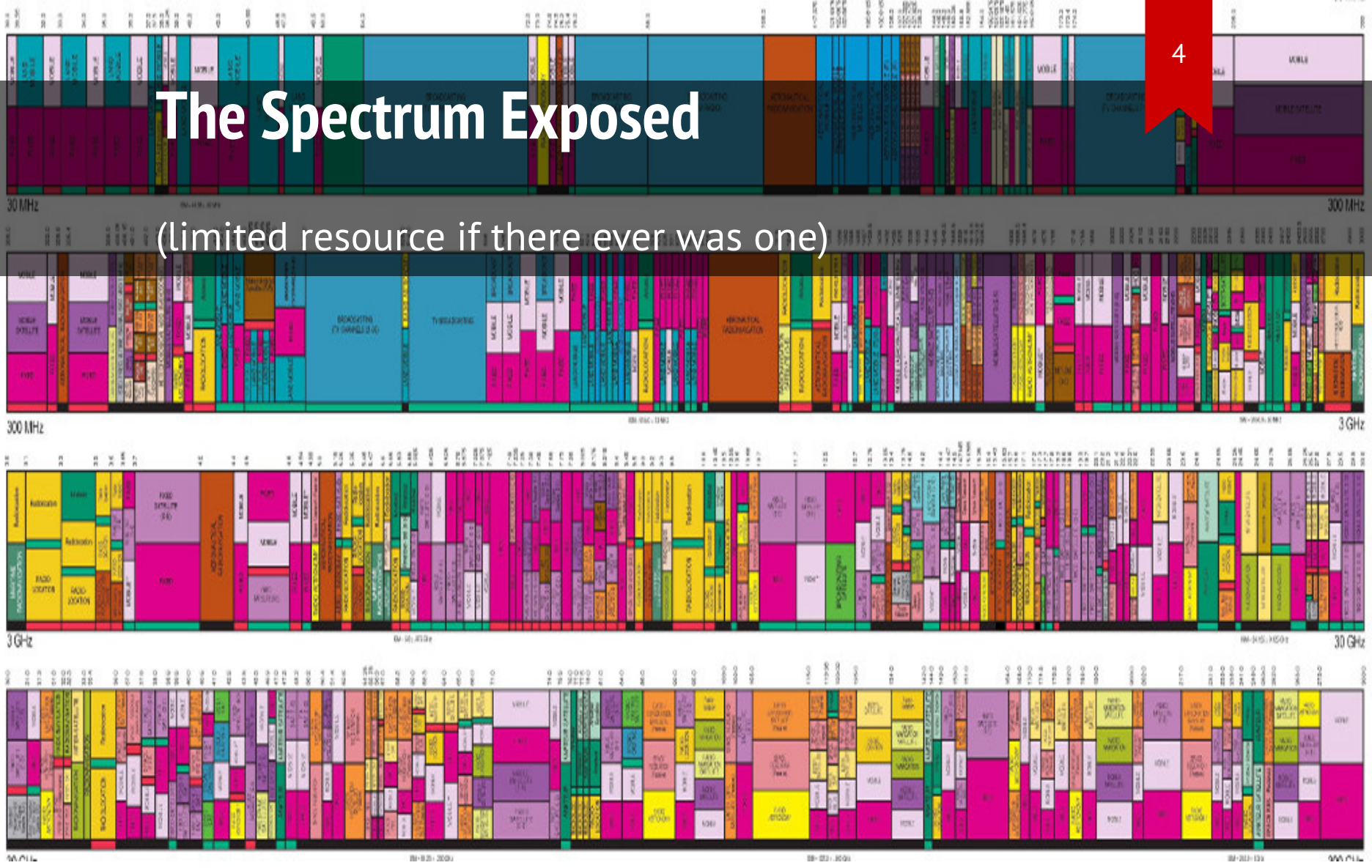
> *"Radio in which some or all of the physical layer functions are software defined" --Wireless Innovation Forum*

Why SDR is so cool (&hot)!!

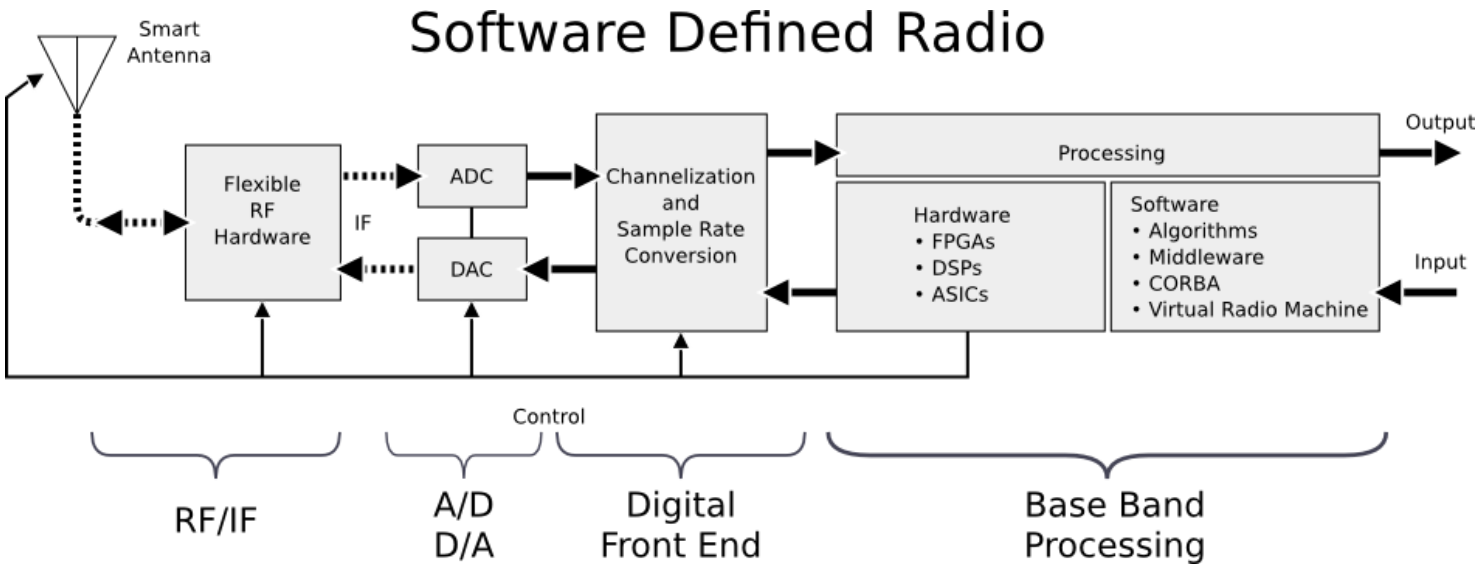
	HW	SDR
Compilation	Months	Minutes
Cost	$50K	$500
Hurdle	RF, HW, SW	SW
Real time configurable	No	Yes
Future proof	No	yes

The Spectrum Exposed

(limited resource if there ever was one)



Canonical SDR architecture

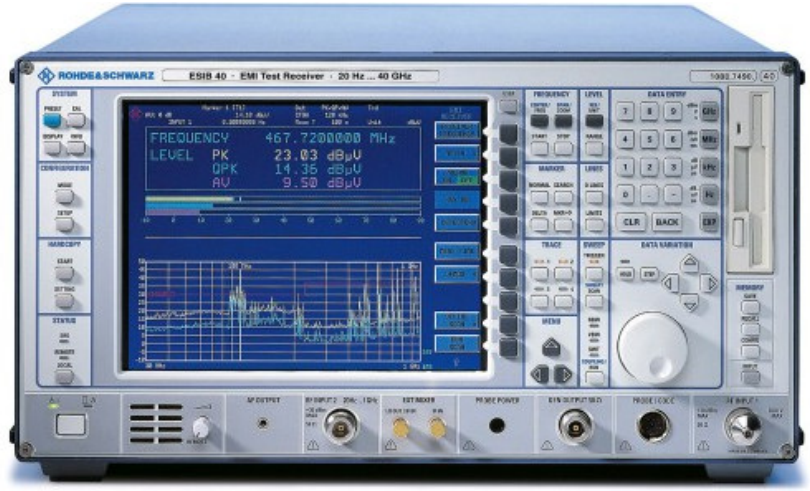


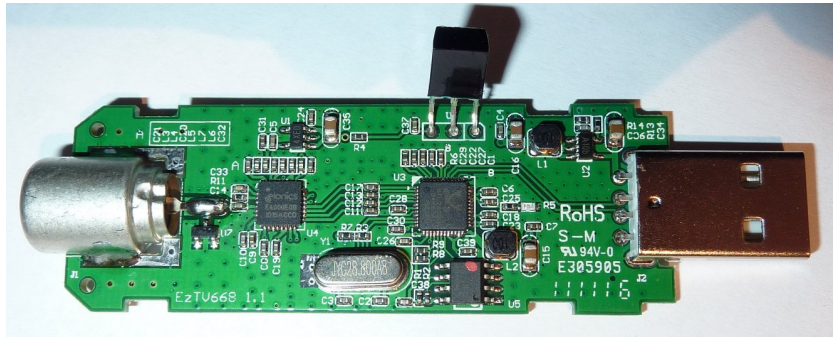
SDR Application Examples

- Amateur radio (HAM, etc)
- Radio astronomy
- Legacy modem emulation
- Wireless comms (GSM, LTE)
- Wireless research (5G)
- Spectrum analysis
- Teaching DSP

SDR Challenge

- Latency (microsecond)
- Throughput (gigabits)
- Size, Weight, Power (SWAP)
- Cost ($20-->$30,000)





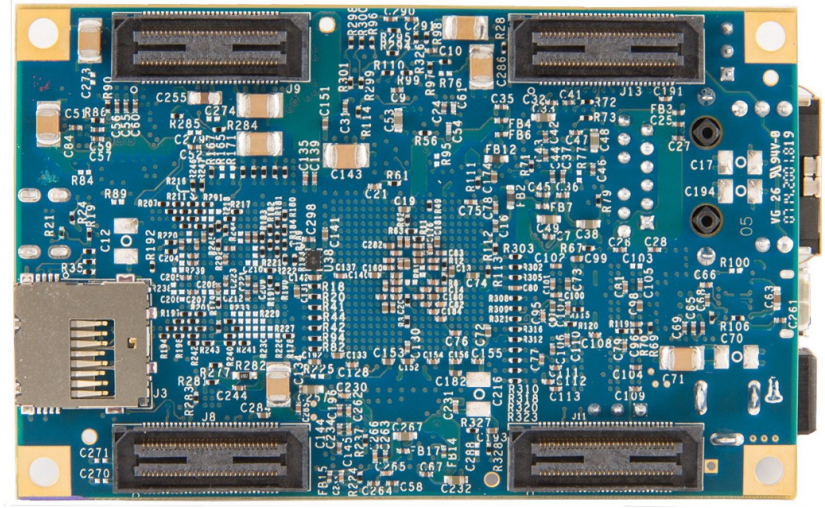
Parallella Introduction

Parallella Specs (parallella.org)

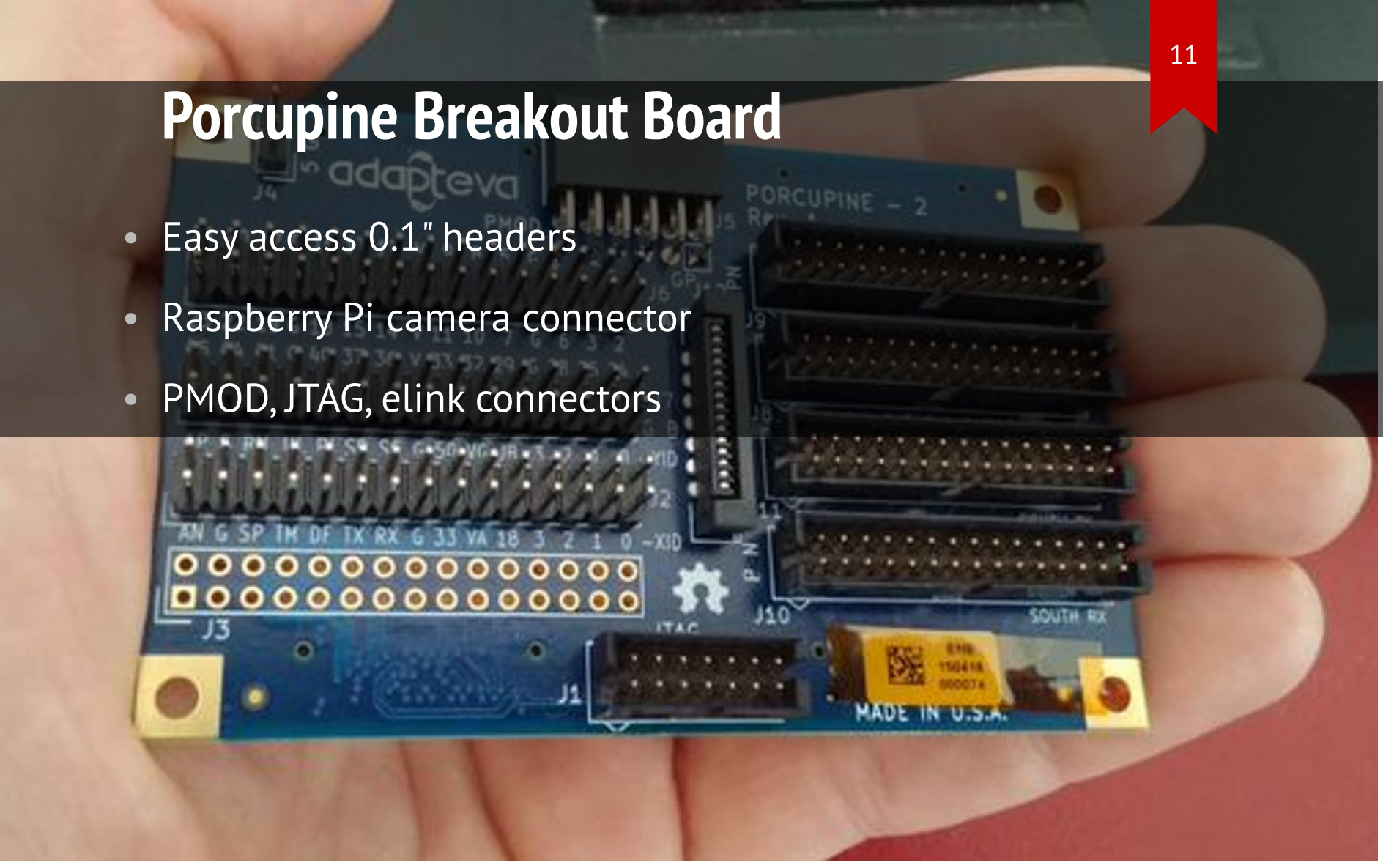
Performance	~30 GFLOPS
Memory	1GB DDR3
IO	~25 Gb/s (48 GPIO)
Size	credit-card
Weight	38g
Power	<5W
Cost	$99 -> $249

Parallella IO

- 0.5mm Samtec connectors
- 48-pin/24Gbps FPGA link
- 2 Epiphany links (20Gbps)
- JTAG, UART, I2C, SPDIF
- LVDS/CMOS
- Adjustable I/O voltage



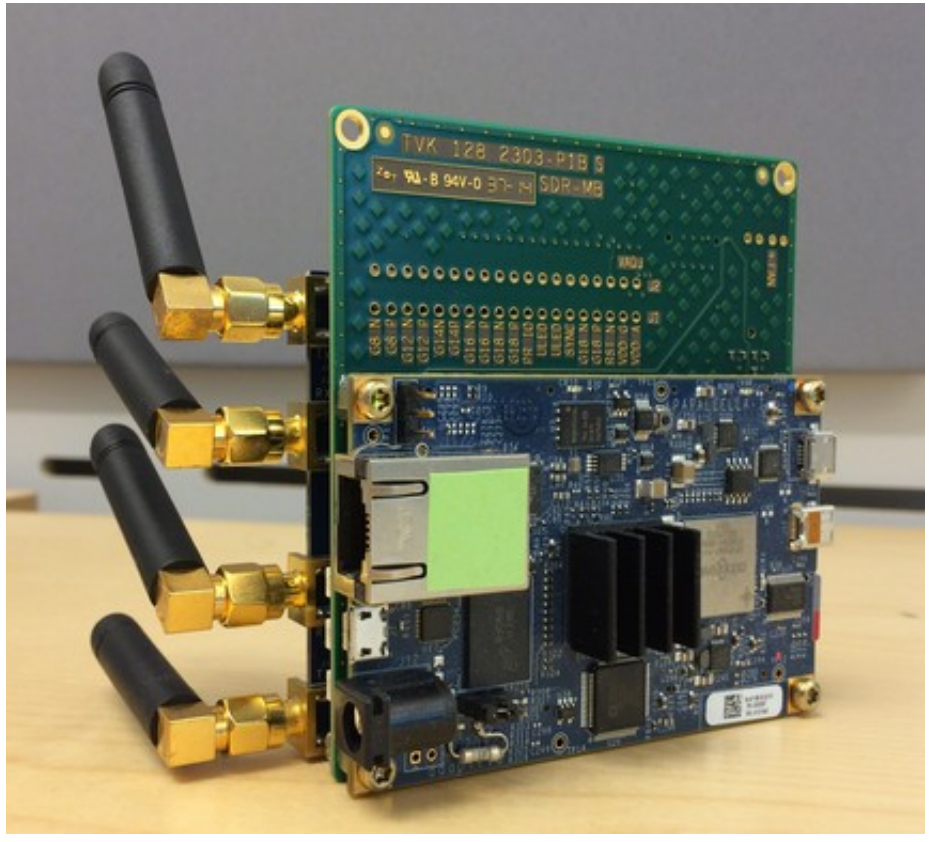
Porcupine Breakout Board



- Easy access 0.1" headers
- Raspberry Pi camera connector
- PMOD, JTAG, elink connectors

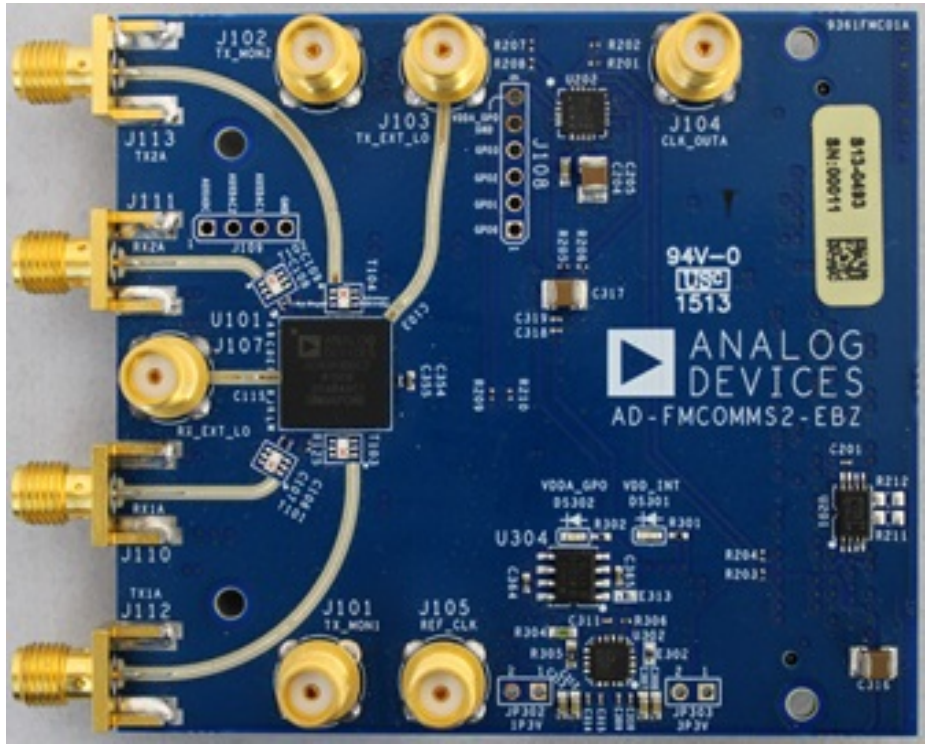
Parallella SDR Platform

- 70MHz - 6GHz
- ADI FCOMMS* board (FMC)
- Parallella carrier
- FMC adapter board
- 100% Open source SW



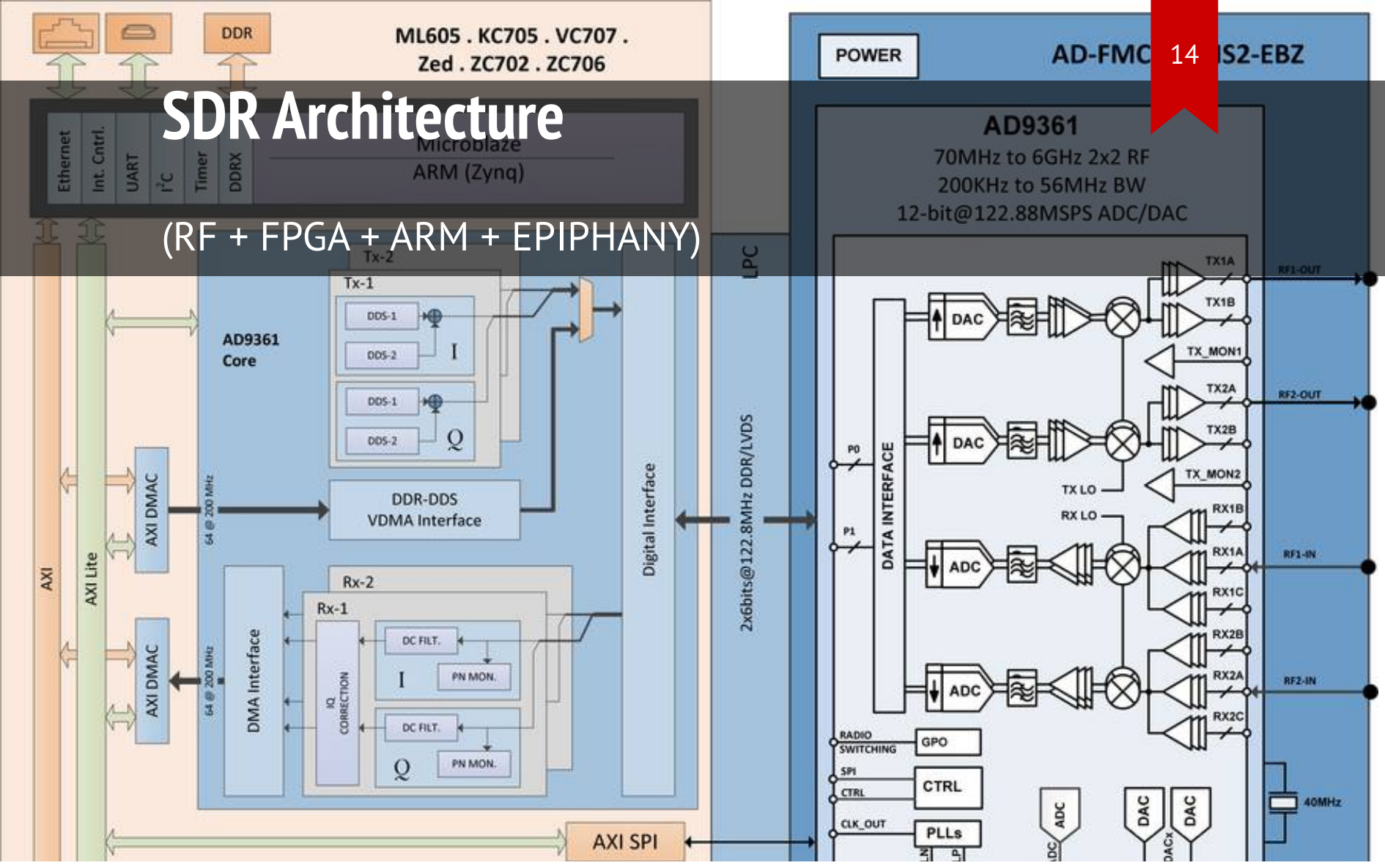
AD9361 Overview

- RF 2 × 2 transceiver
- 12-bit DACs and ADCs
- 70 MHz to 6.0 GHz
- TDD/FDD support
- BW: <200 kHz to 56 MHz
- Noise figure < 2.5 dB
- Independent AGC



SDR Architecture

(RF + FPGA + ARM + EPIPHANY)



Zynq Architecture

- **ARM SOC:**

  -Dual A9 CPUs (up to 1GHz)

  -GigE,USB,UART,CAN,I2C,SPI

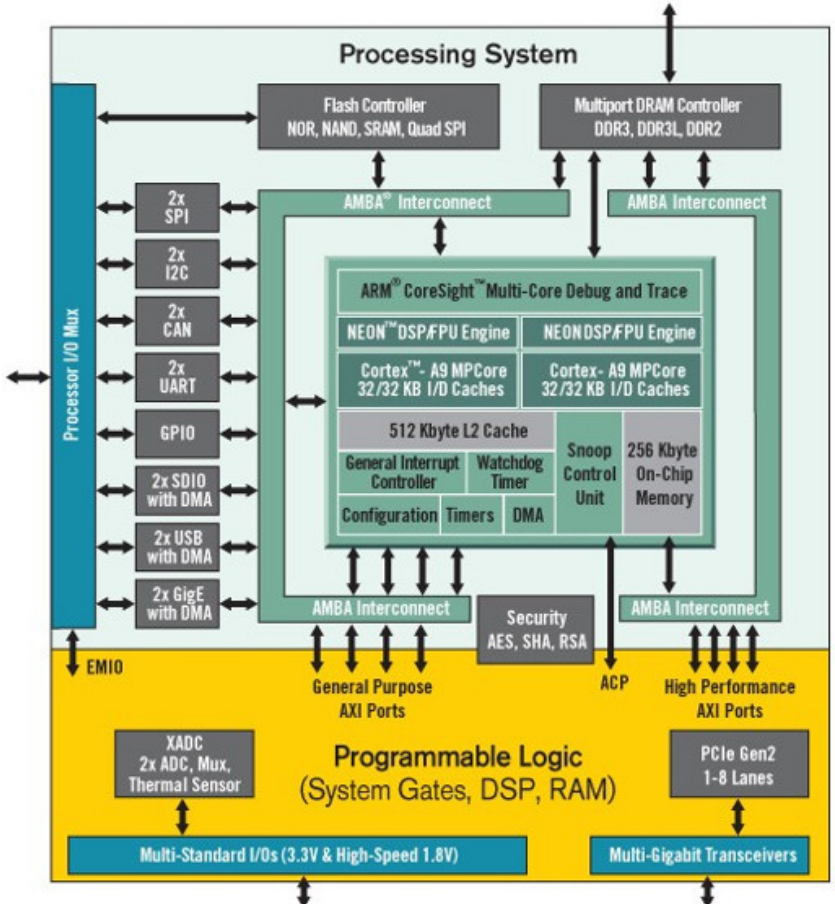
  -Flash & DDR3 controller

- **Programmable Logic:**

  -I/O: 86 --> 470

  -LUTS: 17 --> 277K

  -BRAM: 0.24MB --> 3MB



Epiphany Chip Features

- 16 1GHz RISC processors "DSP"
- C/C++ programmable
- 32 bit IEEE floating point
- 512KB on chip cache
- 128 GB/s NOC bandwidth
- 8 GB/s IO bandwidth
- 512 GB/s local memory BW

Epiphany CPU

- Dual issue 5-8 stage pipeline
- 64 general purpose registers
- IEEE754 floating point (FMADD)
- 16/32b instruction set
- Nested interrput support, 2-Channel DMA engine
- ISA: B,BL,JR,JAL, LDR, STR, TESTSET, ADD, SUB, ASR, LSR, LSL, ORR, AND, EOR, BITR, FADD, FSUB, FMUL, FMADD, FMSUB, FABS, FIX, FLOAT, MOV, MOVT, MOVFS, NOP, IDLE, RTS, RTI, GID, BKPT, TRAP,

Epiphany Memory System

- 32 bit addressing, upper 12 bits specify x,y coordinate in 2D map
- Shared flat address space, no HW caches
- 32KB per core in E16G301, readablea and writeable by all
- 4 independent 64 bit local memory transactions per clock cycle
- Fetch, load, DMA, emesh can generate 32 bytes read/write per cycle

Epiphany Network-On-Chip

- 3 separate meshes for on-chip writes, read requests, off-chip writes
- On chip writes complete in 1 clock cycle
- Non-blocking round robin routing
- 1.5ns latency / hop
- x/y static routing
- Up to 8 bytes transfered per cycle
- extends off chip to I/O (elinks)

Parallella SDR Software

Essential Software Components

- GNURadio: Open source SDR platform
- Epiphany SDK: Epiphany compiler, debugger
- Vivado: FPGA synthesis tools
- COPRTHR: OpenCL, MPI, Threads
- PAL: Optimized open source math/dsp library

Creating a Parallella SD card

- Download image
- Insert SD card in laptop

```
$ gunzip -d <releasename>.img.gz
$ df -h
$ umount <sd-partition-path>
$ sudo dd bs=4M if=<release-name>.img of=<sd-device-path>
$ sync
```

- Remove SD card and insert into Parallella

Install Vivado

- Download Vivado from Xilinx (Choose the web installer)

```
$ sudo unlink /bin/sh
$ ln -s /bin/bash /bin/sh
$ chmod u+x ./Xilinx_Vivado_SDK_2015.2_0626_1_Lin64.bin
$ ./Xilinx_Vivado_SDK_2015.2_0626_1_Lin64.bin
$ source 2015.2/settings64.csh
```

Install GNURadio Dependancies

```
$ sudo apt-get -y install git-core cmake g++ python-dev swig \
pkg-config libfftw3-dev libboost1.55-all-dev libcppunit-dev \
libgsl0-dev libusb-dev libsdl1.2-dev python-wxgtk2.8 \
python-numpy python-cheetah python-lxml doxygen libxi-dev \
python-sip libqt4-opengl-dev libqwt-dev libfontconfig1-dev \
libxrender-dev python-sip python-sip-dev
```

Building GNURadio (Be patient!)

```
$ dd if=/dev/zero bs=1MiB count=2048 of=/home/analog/swap.img
$ sudo mkswap /home/analog/swap.img
$ sudo swapon /home/analog/swap.img
$ git clone https://github.com/analogdevicesinc/gnuradio.git
$ git checkout master
$ mkdir gnuradio/build; cd gnuradio/buid
$ cmake -DENABLE_DOXYGEN:bool=false ..
$ make -j2
$ sudo make install
$ sudo make -C gr-iio install
$ sudo ldconfig
```

Installing "libiio"

```
$ git clone https://github.com/analogdevicesinc/libiio.git
$ cd libiio
$ cmake ./
$ make all
$ sudo make install
```

REFERENCES

FMCOMMS3 User Guide (ADI)

GNURadio Installation (ADI)

IIO-scope User Guide (ADI)

SD-CARD WIKI (ADI)

Epiphany demo

SDR demo