The Embedded Hour

Where Embedded minds come together

Hour #1
April 10, 2020

ADIUVO
ENGINEERING AND TRAINING, LTD.

The Embedded Hour

- Welcome! – The first Embedded Hour
- Objective is to feel like a community, talk about our projects and any other questions which may come up
- Be excellent to each other – we are all here to share, develop and learn.
- Two great engineers talking today
  - Alex Forencich - FPGAs, Datacenters, and Optical Switching
  - Tom Verbeure – Repurposing electronic boards as FPGA development boards
- Following the talks we will have an Open Q & A discussion

ADIUVO

FPGAs, Datacenters, and Optical Switching

Alex Forencich

Embedded Hour
April 10, 2020

About myself

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Postdoc in systems and networking group at UCSD
  - Joined group as an undergrad in 2011
  - We build hardware (not just simulations)
  - https://circuit-switching.sysnet.ucsd.edu
- Research focus: optical switching in datacenter networks
  - Goal: use optical switches to improve efficiency

Topics for today

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Motivation and background
- IBM burst-mode receiver/silicon photonic switch integration
  - Published in OFC 2018, IEEE JLT 2020
- Corundum open source NIC
  - To appear at FCCM 2020

Motivation

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Web applications and cloud services run in datacenters
- Bandwidth within datacenters increasing rapidly

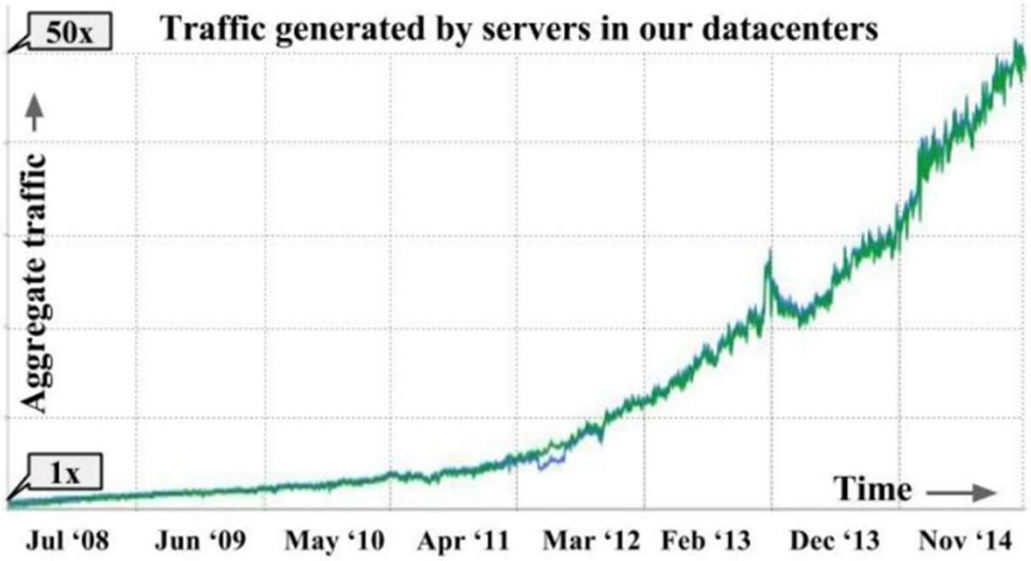


Figure from Google's Jupiter Rising paper

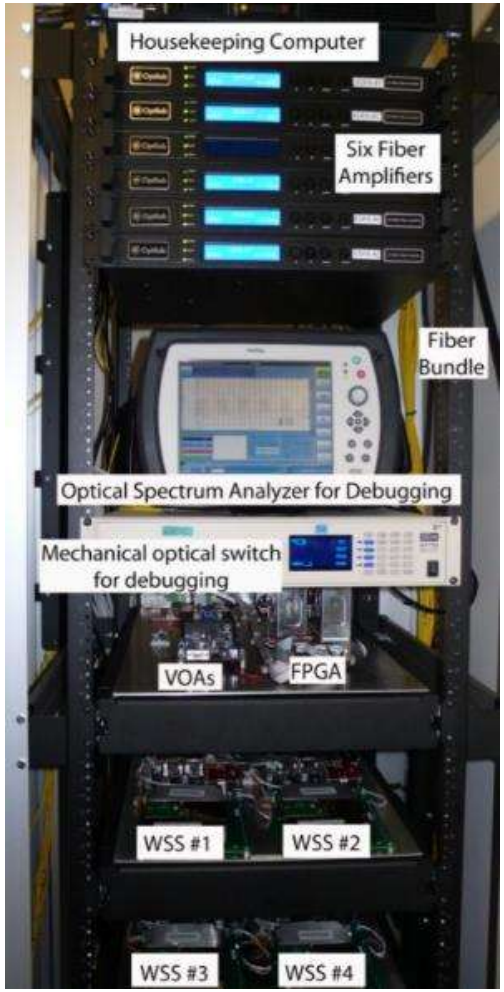
Motivation

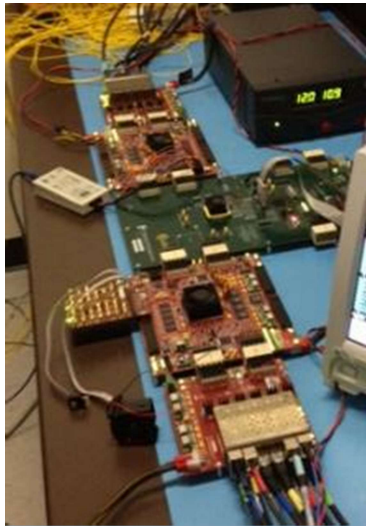
UC San Diego
JACOBS SCHOOL OF ENGINEERING

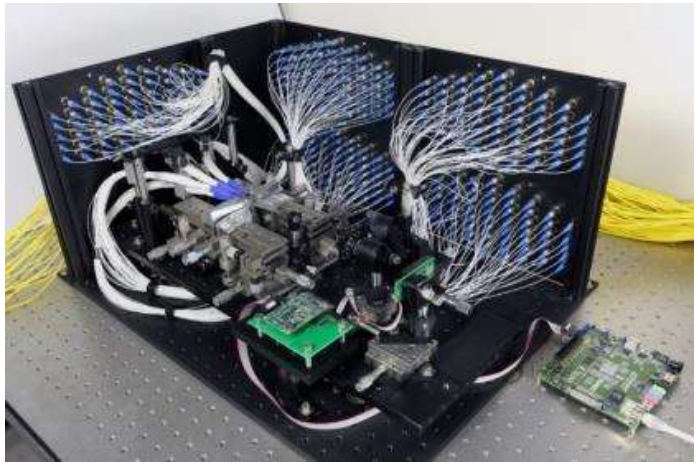
- Datacenter networks traditionally built with packet switches
  - Distributed, operate on a packet-by-packet basis
- Packet switch BW and link BW not increasing fast enough
- Possible solution: optical circuit switching
  - Operates at the level of links instead of packets
  - Link rate agnostic; high data rate per port
  - Potential for significant power and cost savings
- Challenges
  - Fundamentally different paradigm
  - Requires different network architectures, time synchronization, precise transmission control, fast locking receivers, new protocols, etc.

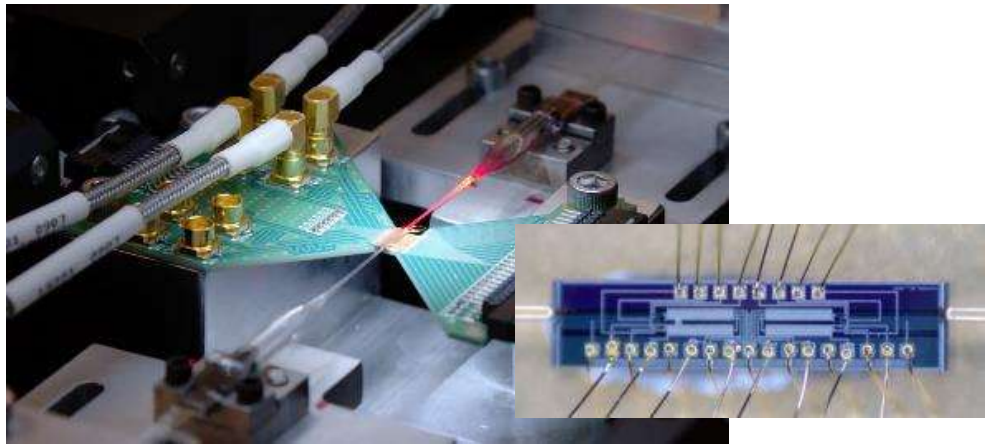
Background

UC San Diego
JACOBS SCHOOL OF ENGINEERING









IBM Burst-Mode Receiver Integration

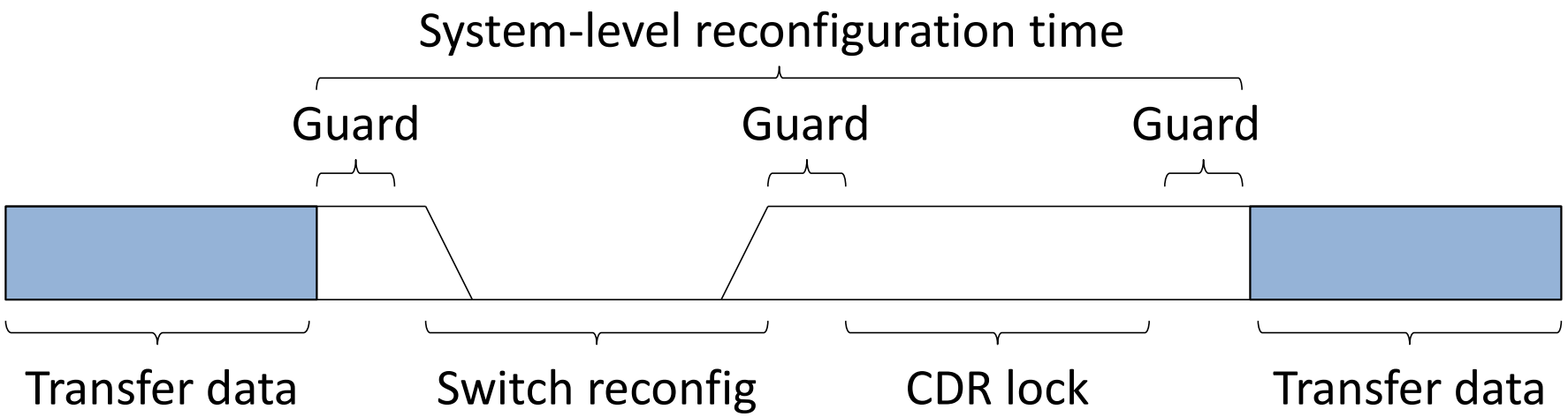
UC San Diego
JACOBS SCHOOL OF ENGINEERING

- 5 ns silicon photonic switch
- 31 ns burst-mode receiver

System-Level Reconfiguration Time

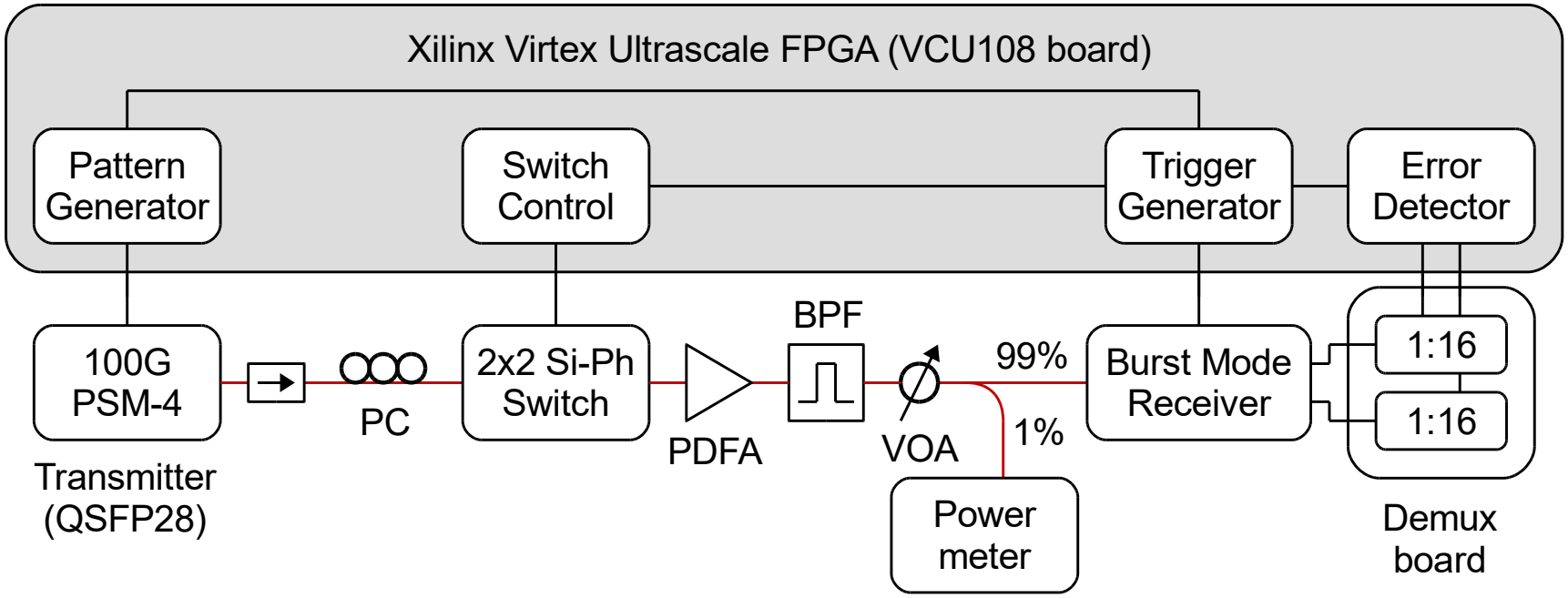
UC San Diego
JACOBS SCHOOL OF ENGINEERING

- The time required to change the configuration of the switch
  - The time during which data cannot reliably transit the network
- Includes:
  - Switch reconfiguration time
  - Clock Data Recovery (CDR) locking time
  - Synchronization guard delays



Burst-Mode Link Test Setup

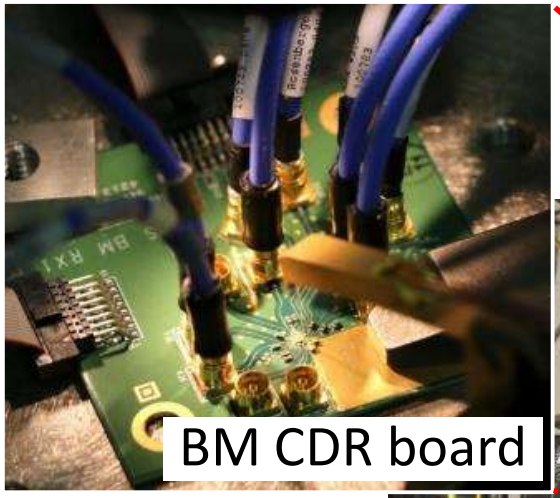
UC San Diego
JACOBS SCHOOL OF ENGINEERING

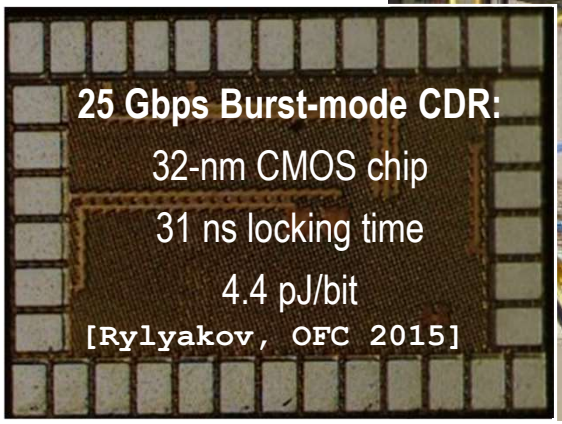


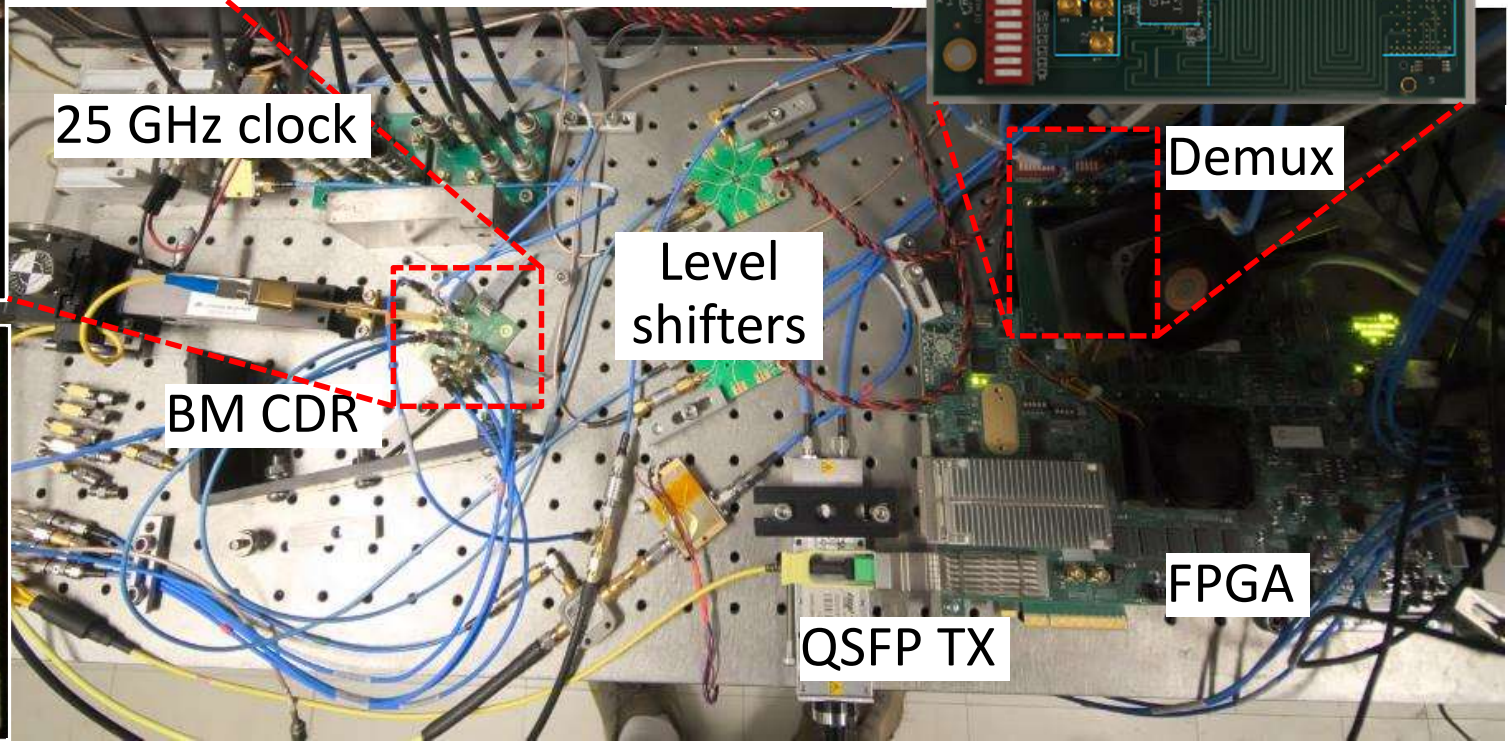
- Data Plane
  - 100G PSM-4 QSFP28 TX
  - 2x2 Silicon-Photonic switch
  - Burst-mode receiver
- Control Plane
  - Xilinx XCVU095 FPGA
  - 25 Gbps pattern generator
  - 25 Gbps error detector

Testbed: FPGA, TX, BM CDR, Demux

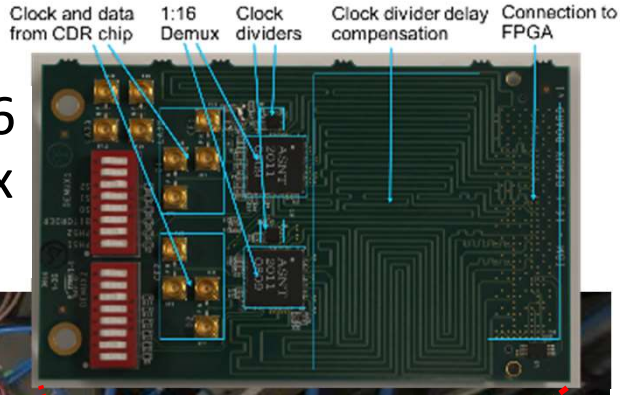
UC San Diego
JACOBS SCHOOL OF ENGINEERING





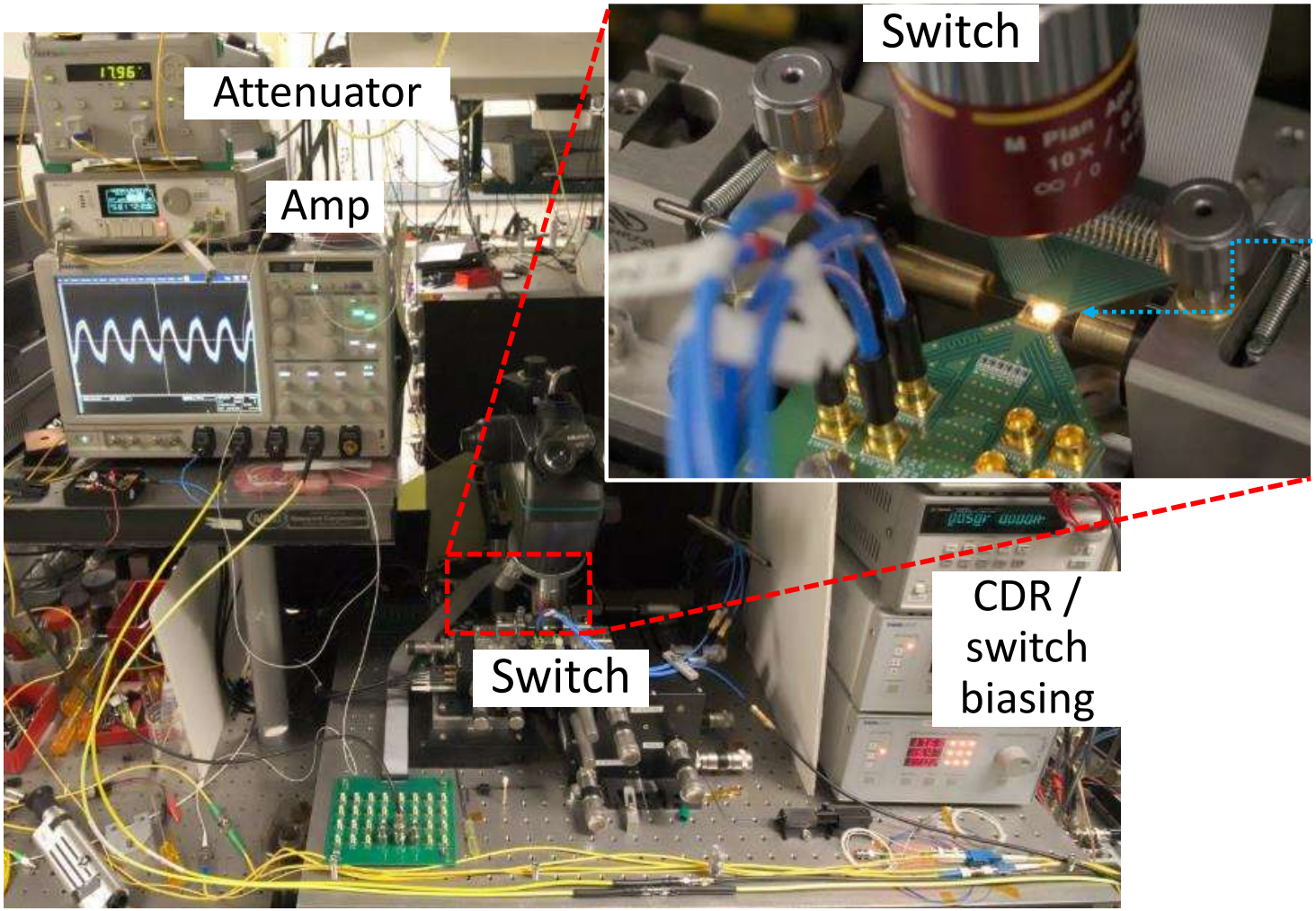


2x 1:16 Demux board

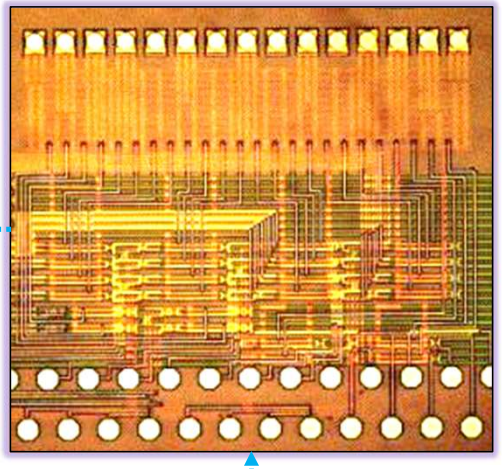


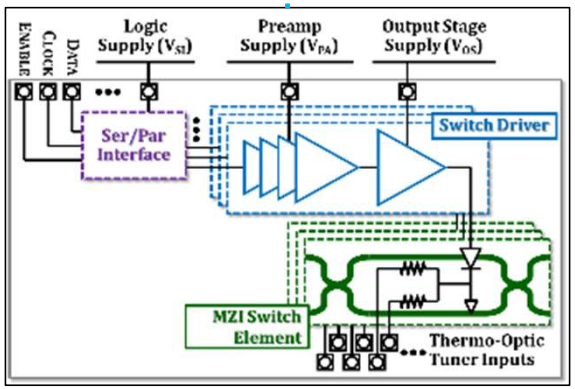
Testbed: Silicon Photonic Switch Chip

UC San Diego
JACOBS SCHOOL OF ENGINEERING



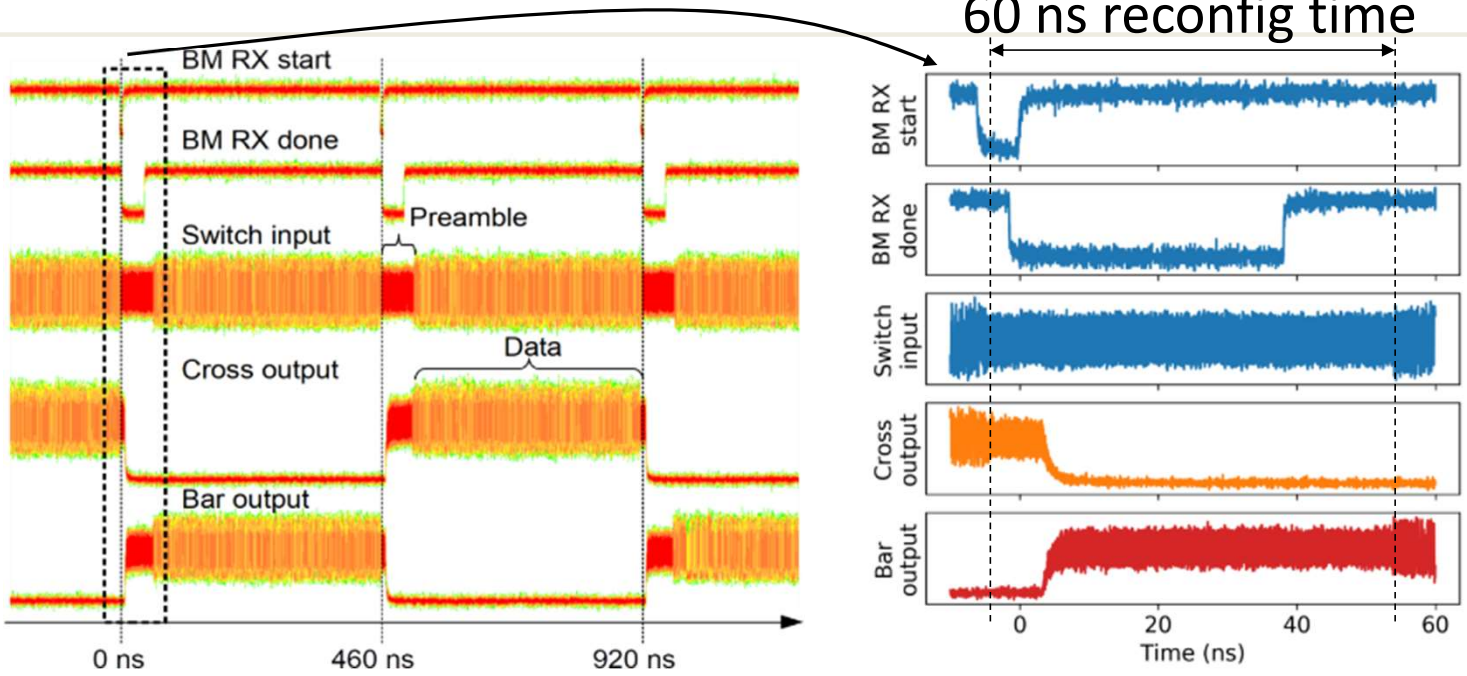
Switching times << 5 ns





[N. Dupuis et al., IEEE JLT 33, 3597-3606, 2015]

Measured Waveforms

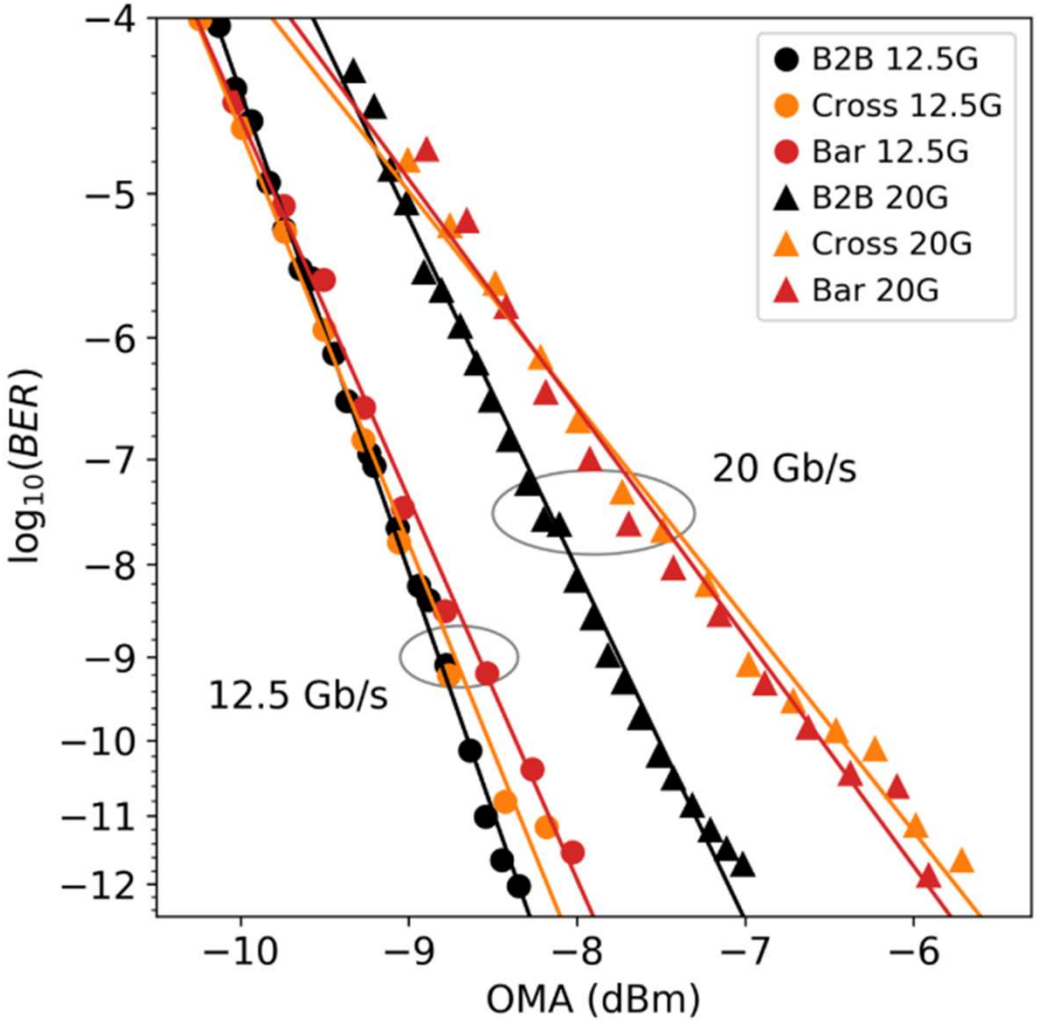


Payload size (B)	2048	1024	2048	1024
Data rate (Gbps)	12.5	12.5	20	20
Cycle time (ns)	1366	730	858	460
Switch time (ns)	90	90	60	60
Duty cycle (%)	93	87	93	87

Results: Switch burst-mode link with low BER

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- CDR in burst mode
- Si-Photonic Switch
  - Bypassed
  - Switching, cross state
  - Switching, bar state
- Data rates
  - 12.5 Gbps (●)
  - 20 Gbps (▲)
- FPGA pattern gen
  - 1010 preamble
  - 2048 bytes of PRBS15
- FPGA error detection
  - Burst mode frame sync, PRBS15 payload



Summary

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- System-level reconfiguration time is only part of the story
- These measurements are idealized
  - Direct hardware control
  - No variance from time synchronization
- For operation in a datacenter environment
  - Need precision synchronization and transmission control
  - As we have learned from Mordia, REACToR, Rotornet, this is not feasible with software or with commercial NICs

Problem

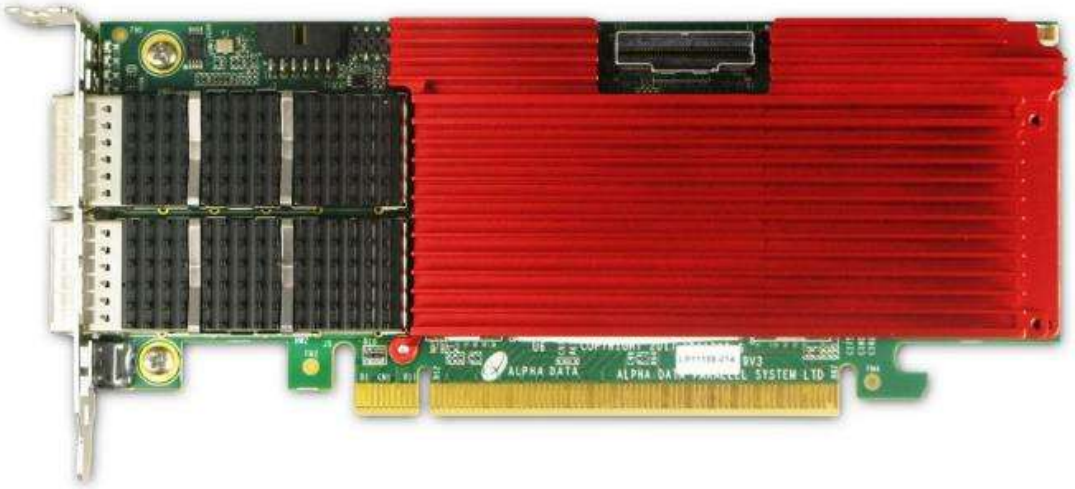
UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Precise control of packet transmission from servers has many applications
  - Rate limiting, flow control, congestion control, TDMA, circuit switching, etc.
- Sub-microsecond control not feasible in software
- Little hardware support in commercial NICs
  - Limited number of queues, limited control over transmit scheduling
  - Commercial SmartNICs designed for packet processing, not flow control
- Possible to do on an FPGA
  - But there are no extensible high-performance FPGA-based NIC reference designs to use as a foundation

Solution: Corundum NIC

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- A reference open-source FPGA-based NIC!
  - A high performance “reference” NIC
  - Extensible: open-source permits additional custom features
  - Key application: hardware prototyping of experimental networks/protocols



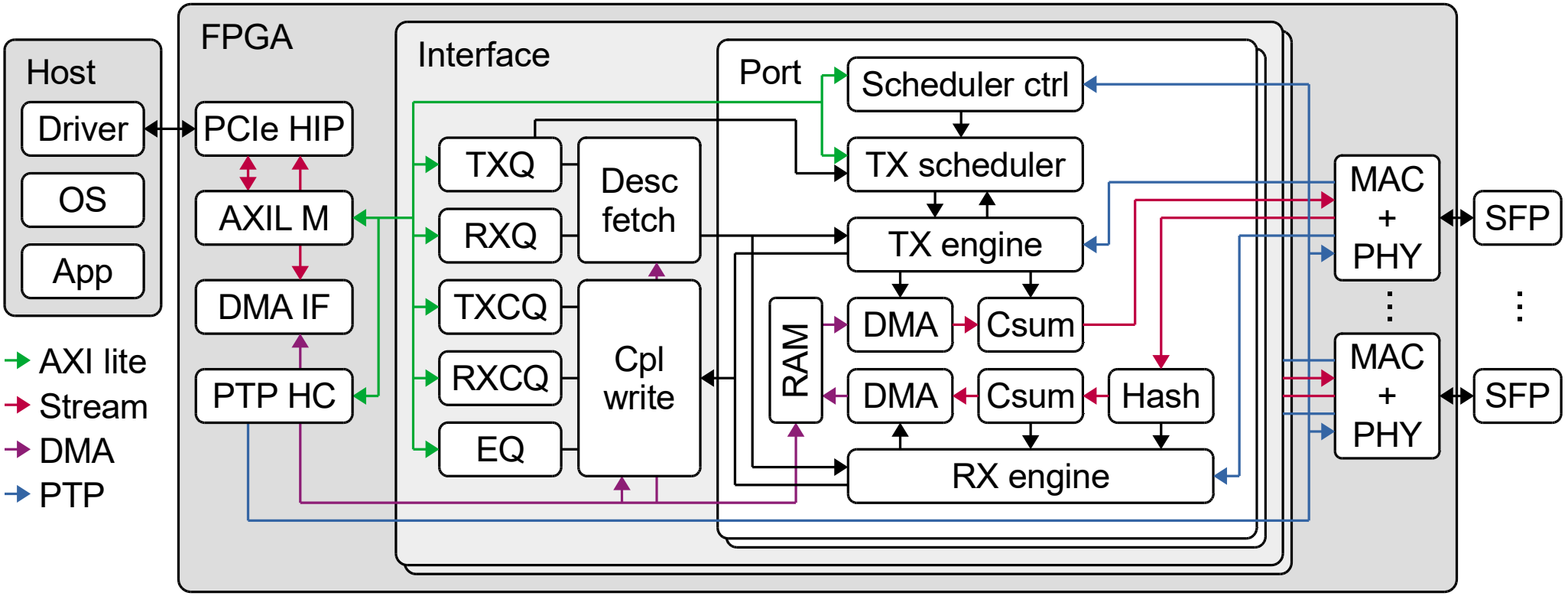
High-Level Features of Corundum

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Open-source, high-performance, FPGA-based NIC
  - PCIe gen 3 x16
  - Multiple 10G/25G/100G Ethernet ports
  - Fully custom, high-performance DMA engine
  - Linux driver
- Control data transmission on a per-destination basis
  - 10,000+ independent hardware queues
- Microsecond precision TDMA
  - Synchronized via precision time protocol (PTP)
  - TDMA PHY-level BER measurement capability
- Source code: https://github.com/ucsdsysnet/corundum

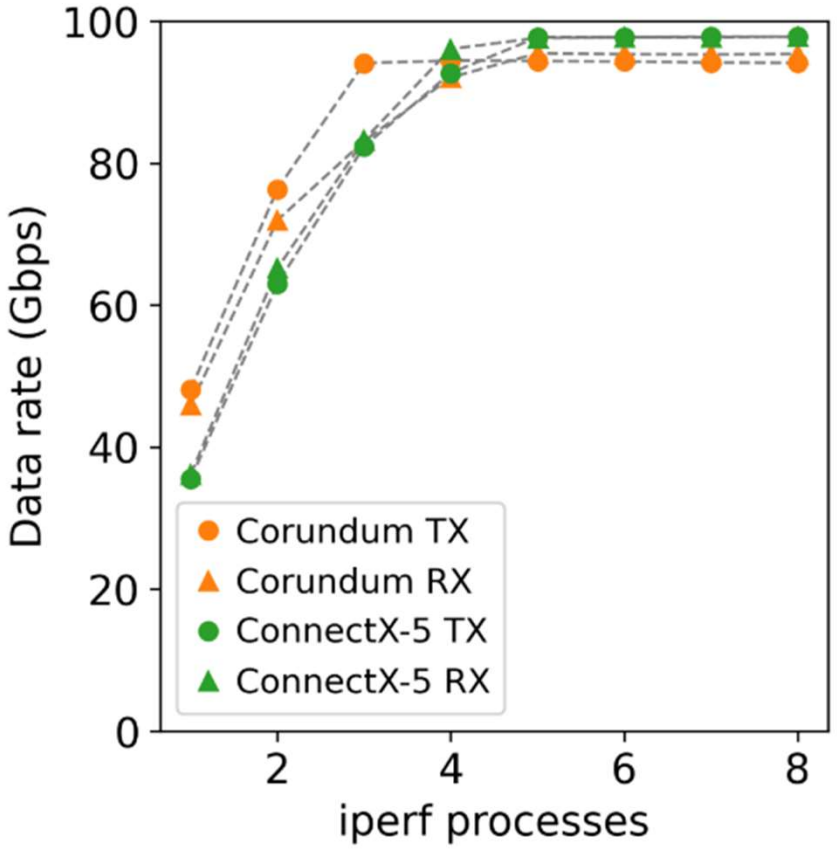
Corundum Block Diagram

UC San Diego
JACOBS SCHOOL OF ENGINEERING

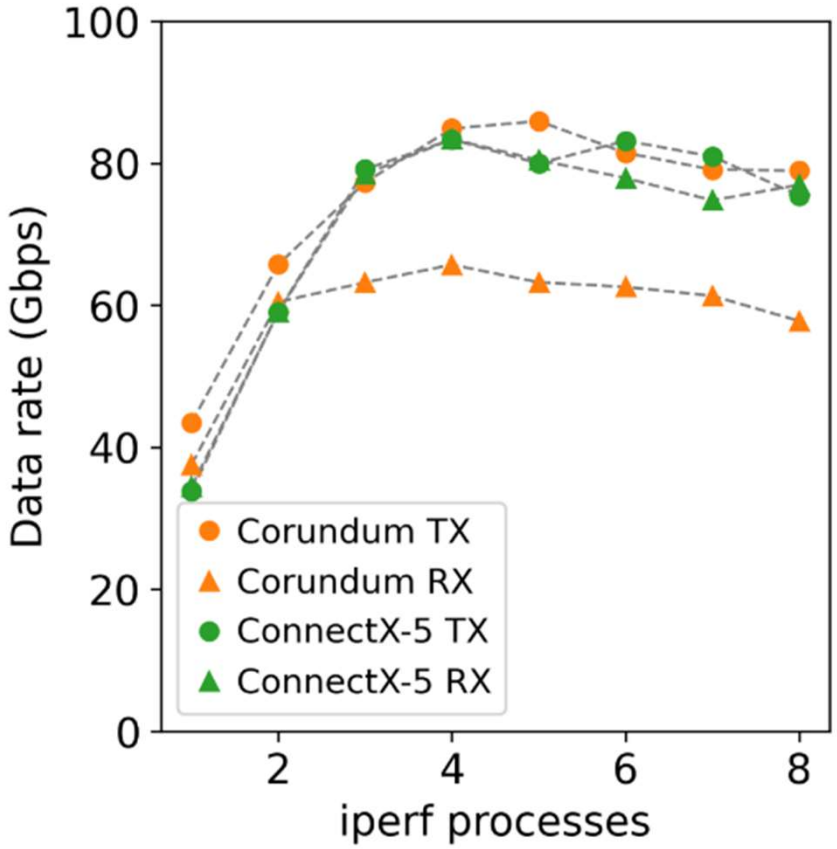


Performance: 9 KB MTU

UC San Diego
JACOBS SCHOOL OF ENGINEERING



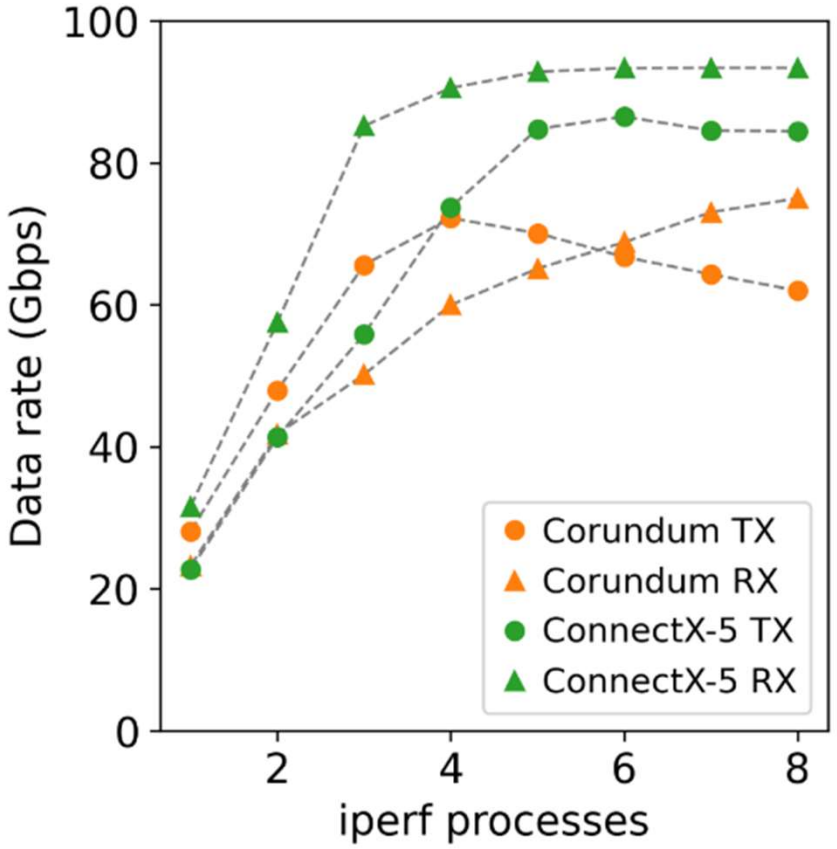
Intel host, 9KB MTU, separate



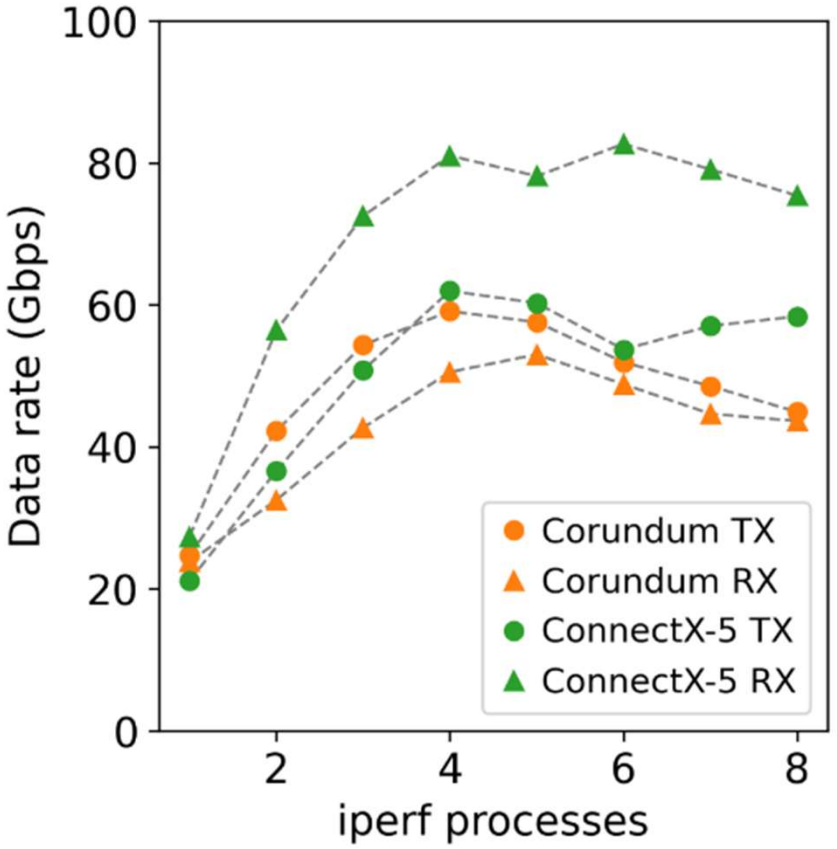
Intel host, 9KB MTU, simultaneous

Performance: 1.5 KB MTU

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Intel host, 1.5KB MTU, separate

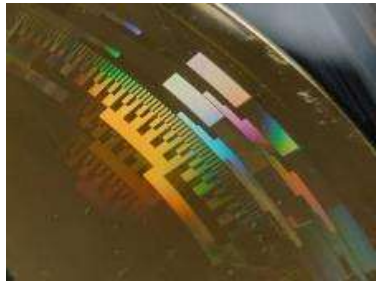


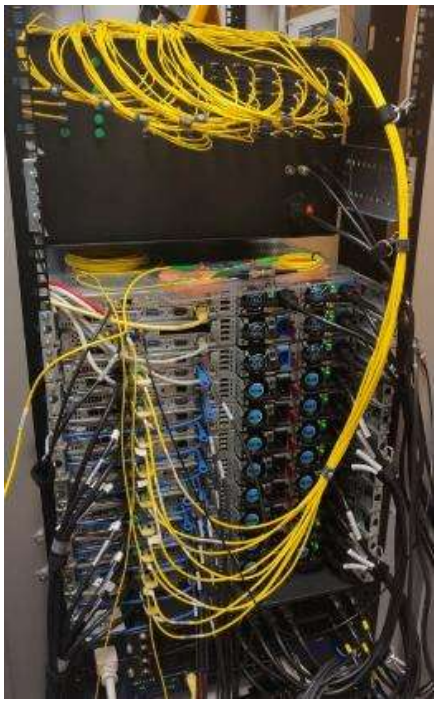
Intel host, 1.5KB MTU, simultaneous

TDMA

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- NIC scheduler can control queues based on PTP time
  - Enables sub-microsecond-resolution TDMA
- TDMA off
  - 94 Gbps
- 200 us period, 50% duty cycle TDMA schedule
  - Guard time 2 us
- Using TDMA schedule, can run iperf through pinwheel switch
  - Initial test with old FW: 3 Gbps on 10 Gbps link with low packet loss

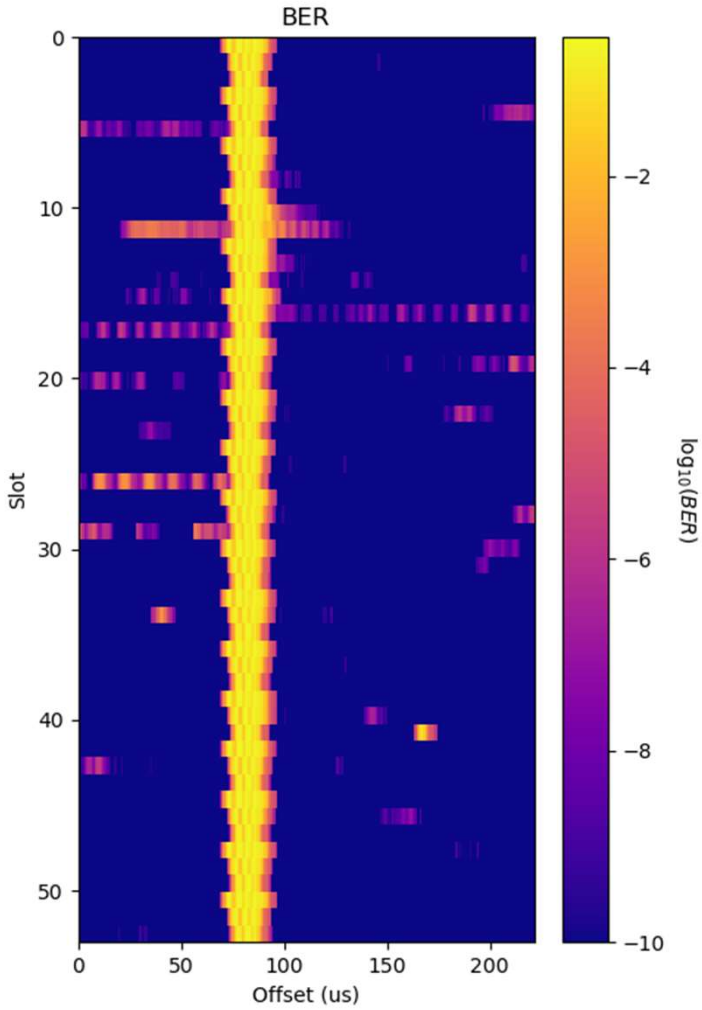




In situ link-level measurements

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- BER measurement capability integrated into NIC
- Measure link-level performance from vantage point of every NIC in datacenter
- Supports time-domain BER
  - Synchronized over network via PTP
  - Measure every path through switch
- Heat map represents signal at one receiver through pinwheel switch



Where to go from here

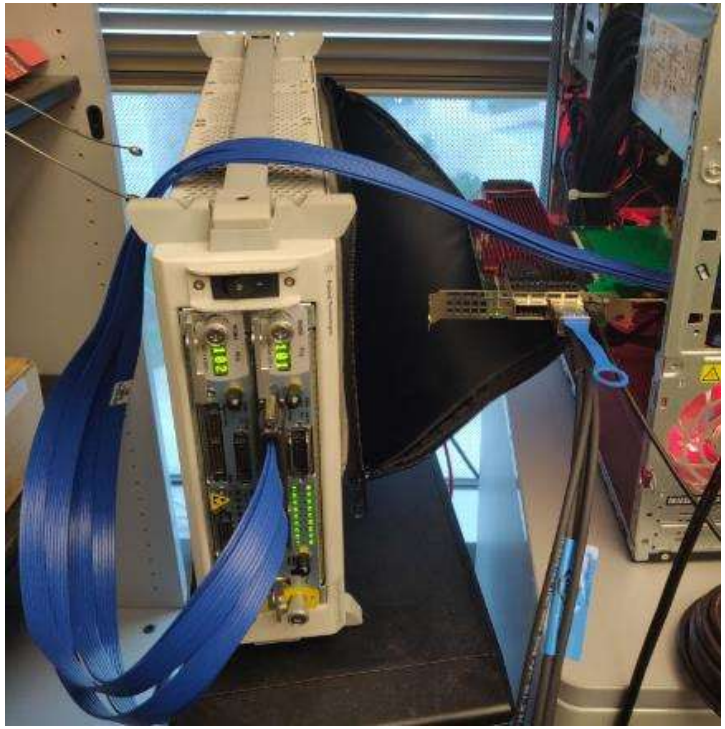
UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Current design functional at 100 Gbps on Alpha Data board
  - Linux driver working
  - Can saturate 10G and 25G links on both TX and RX
  - Can achieve 94 Gbps on a 100 G link on both TX and RX

- Near-term Objectives:
  - Improve throughput for small packet sizes
  - Improve TDMA implementation/schedulers
  - Develop/validate novel hardware-based networking protocols
    - Rotornet, Opera
  - DPDK driver

Thank you

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Cisco Modem as Cheap FPGA Development Kit

Tom Verbeure

Commercial FPGA Boards as Development Kit

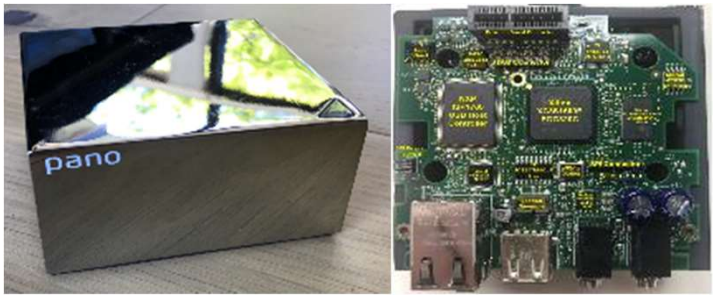
Pros:

- Price!!!
- FPGA capacity
- Speciality components and features
- Satisfaction of hacking

Cons:

- Ugly hacks may be required
- Tedious reverse engineering
- Lack of generic GPIOs
- Little to no support online

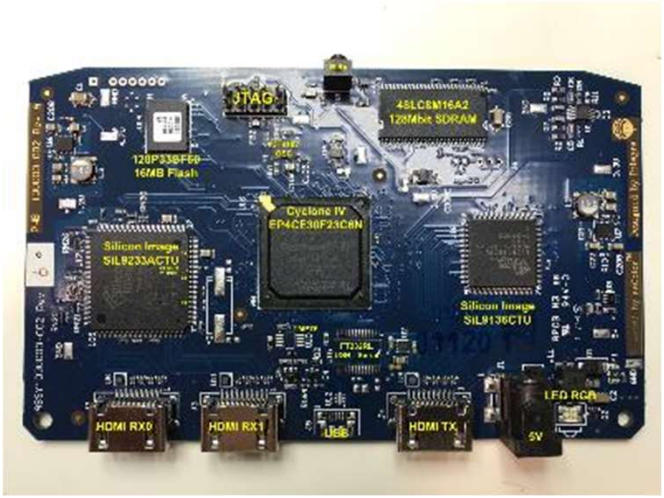
Examples (1)

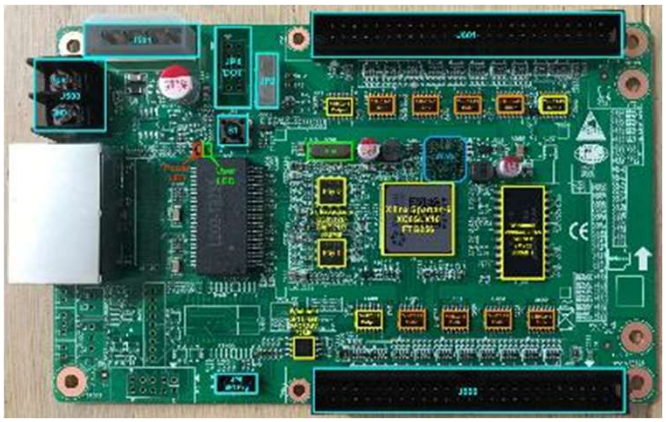


- Pano Logic G1 and G2
  - Obsolete FPGA-only thin clients
  - Huge FPGAs (Spartan 3E-1500 / Spartan 6 LX150)
  - All possible PC-like IOs: USB, VGA (G1) or DVI/HDMI (G2), Ethernet, DRAM, …
  - Easy accessible JTAG port
  - ~$20 on eBay
  - https://github.com/tomverbeure/panologic and https://github.com/tomverbeure/panologic-g2
- Colorlight 5A-75B
  - Current commercially available LED matrix controller
  - Large ECP5 FPGA
  - Fully supported by open source tools: Yosys, Nextpnr
  - 2x GigE, DRAM, lots of output pins
  - Easy accessible JTAG port
  - $15 on AliExpress
  - https://github.com/q3k/chubby75

Examples (2)

- eeColor Color3
  - Obsolete HDMI color processor
  - 2x HDMI in, HDMI out
  - Cyclone IV EP4CE30
  - Easy access JTAG port
  - https://github.com/tomverbeure/color3
  - $20 on Amazon
- RV901T
  - Spartan 6 LX16 version of Colorlight 5A-75B
  - Requires some surgery for JTAG port
  - $18 on AliExpress
    https://github.com/q3k/chubby75





See FPGA Board Hack on Hackaday.io for more candidates!

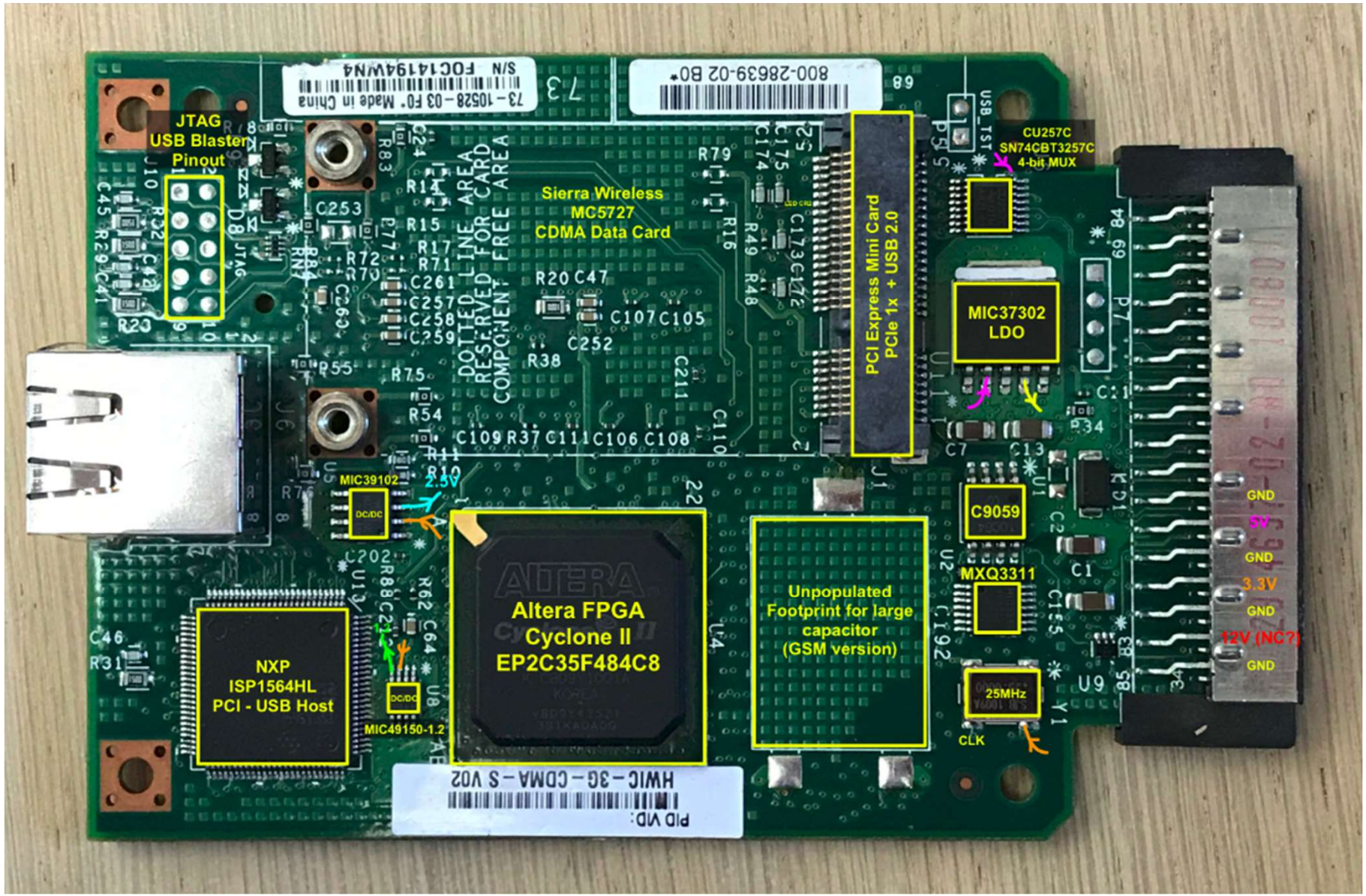
Cisco Uses a Lot of FPGAs

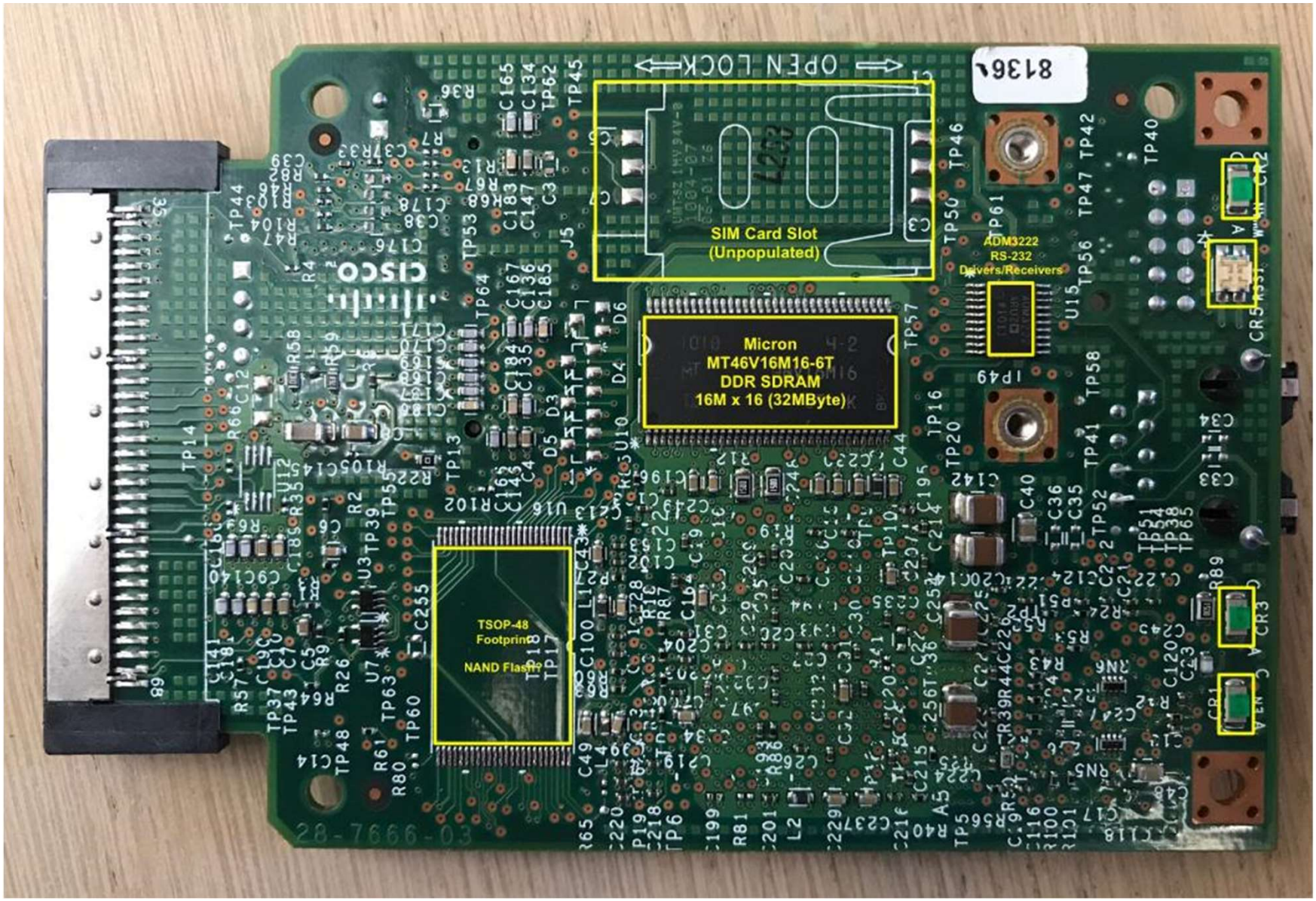
- Products with high margins are good for FPGAs
- FPGAs allow for in-the-field upgrades through IOS, the Cisco router OS
- Many Cisco extensions boards with FPGA, but only a few are interesting
- HWIC connector for router expansion -> lots of GPIOs going to FPGA!
- Bitstream loaded by router -> not self-hosting!

VWIC3-MFT1-T1/E1	Stratix II EP2S15	$4	SRAM, Quartus Lite, No JTAG
VWIC3-MFT2-T1/E1	Stratix II EP2S30	$5	SRAM, Quartus ($$$), No JTAG
HWIC-3G-CDMA (GSM)	Cyclone II EP2C35	$8	DDR2, Quartus Lite, NOR Flash(*), JTAG
EHWIC-3G-HSPA+7-A	Cyclone II EP2C35	$20	SRAM, Quartus Lite, JTAG

CISCO HWIC-3G-CDMA

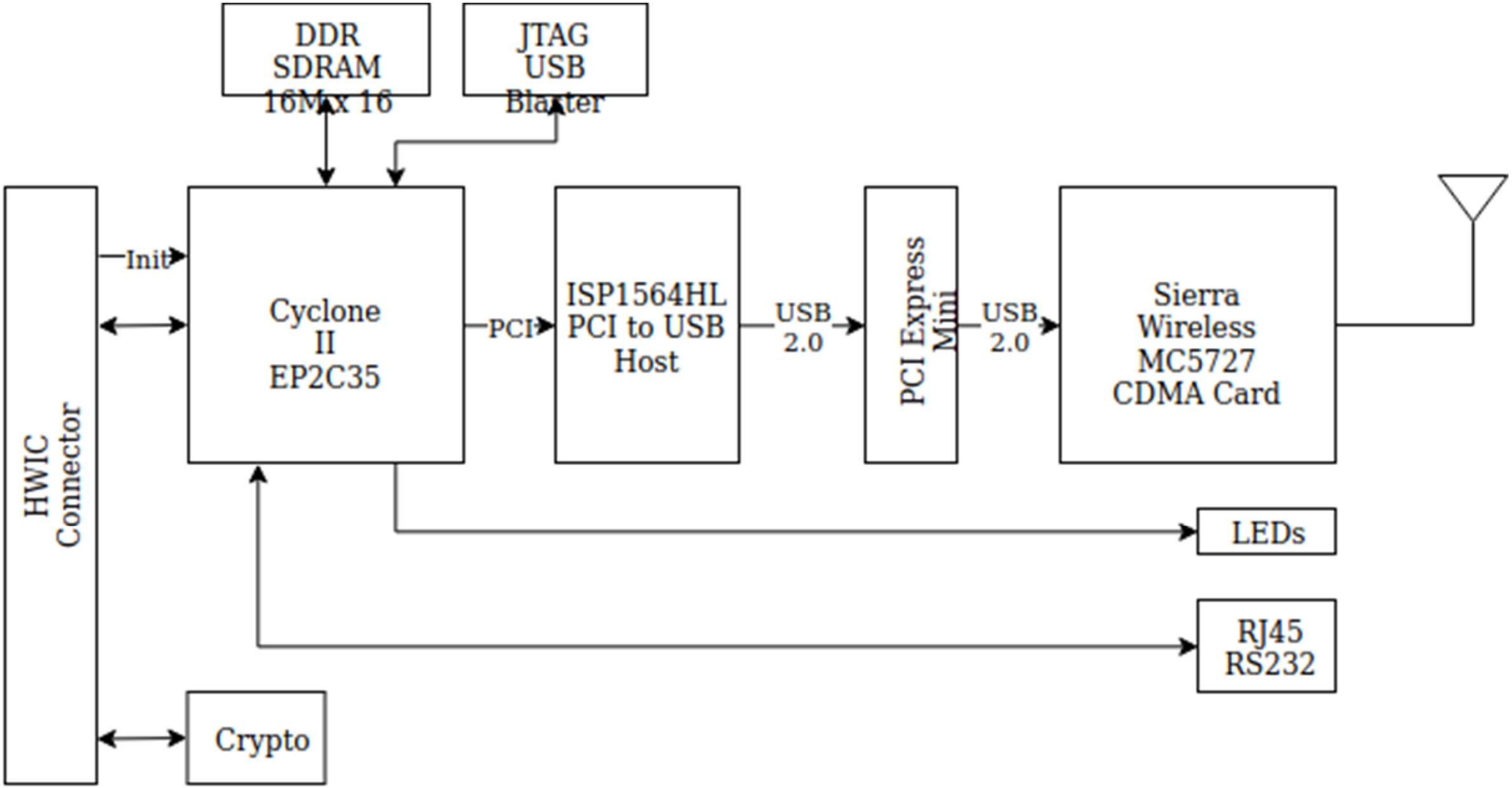

SIERRA WIRELESS
MC5727
74-4928-03
AUX
MAIN
ALTERA
Cyclone II
D43484064


JTAG
USB Blaster
Pinout
Sierra Wireless
MC5727
CDMA Data Card
DOTTED LINE AREA
RESERVED FOR CARD
COMPONENT FREE AREA
PCI Express Mini Card
PCIe 1x + USB 2.0
CU257C
SN74CBT3257C
4-bit MUX
MIC37302
LDO
C9059
MXQ3311
MIC39102
2.5V
Altera FPGA
Cyclone II
EP2C35F484C8
Unpopulated
Footprint for large
capacitor
(GSM version)
NXP
ISP1564HL
PCI - USB Host
MIC49150-1.2
25MHz
CLK
GND
5V
GND
3.3V
GND
12V (NC?)
GND


SIM Card Slot
(Unpopulated)
Micron
MT46V16M16-6T
DDR SDRAM
16M x 16 (32MByte)
ADM3222
RS-232
Drivers/Receivers
TSOP-48
Footprint
NAND Flash
OPEN LOCK
8136
28-7666-03

Components

- Intel Cyclone II EP2C35
  - 33K LEs
  - 484Kbits of BRAMs
  - 35 HW multipliers
  - 4 PLLs
- USB Blaster pin header footprint - easy to solder!
- RS232 port (+-12V) with RJ45 connector
- 32MB DDR2 DRAM
- PCI Express Mini socket
- NXP ISP1564HL USB PCI Host Controller
  - Goes to USB pins of PCI Express Mini socket
- NOR flash socket (empty!)
- Power Regulators


DDR
SDRAM
16Mx 16
JTAG
USB
Blaster
HWIC
Connector
Init
Cyclone
II
EP2C35
PCI
ISP1564HL
PCI to USB
Host
USB
2.0
PCI Express
Mini
USB
2.0
Sierra
Wireless
MC5727
CDMA Card
LEDs
RJ45
RS232
Crypto

Minimal Cost of Dev Kit

- Main Board: $8 on eBay
- JTAG:
  - USB Blaster (FT245 + CPLD based): $8
  - Solder Iron: $5
  - Solder: $1
  - 2x6 Pin Header: $1
  - Desoldering wick: $1
  - Desoldering hollow needles: $1

Total: $25 + Power ($5 or $60)

- Nice to Have
  - USB to RS232 RJ45 console cable: $5

Power

- HWIC has 12V, 5V, and 3.3V connections
  - 3.3V is sufficient to get most logic working

Best generic solution:

- Bench power supply: ~$60

Cheap solution:

- DC-DC convertor based on LM2596: ~$5 to $8 (with display)

High Quality USB Blaster Clone: FT245+CPLD

$8

$2

Development Tools

- PC
- Quartus 13.0sp1 Web version
  - This is the last one that supports Cyclone II devices (2013)
  - Free, as in beer
  - GUI and functionality almost identical to today's Quartus

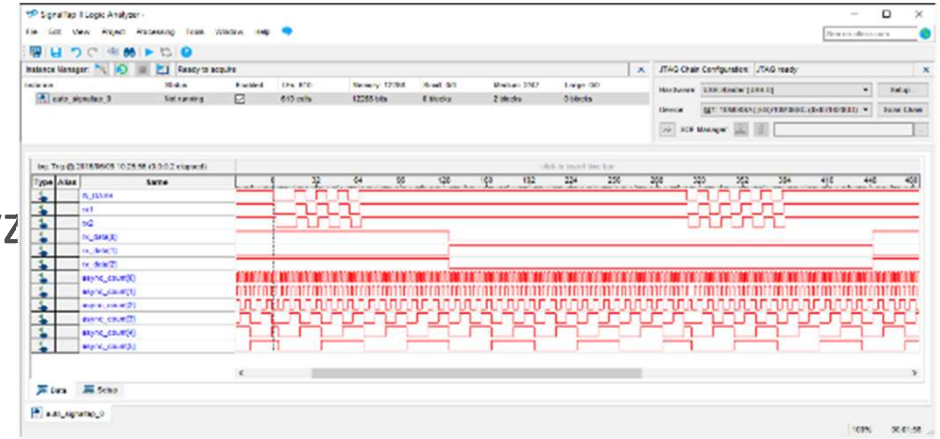
Intel/Xilinx Proprietary Tools vs Yosys/NextPNR

Benefits of open source:

- Feels good!
- Very fast synthesis and place & route
  - Usually quite a bit faster than commercial tools
- ...

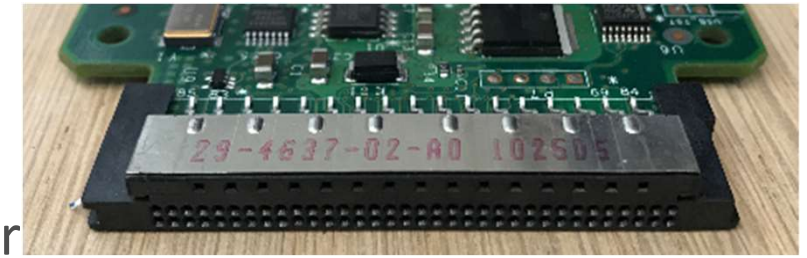
Biggest disadvantage:

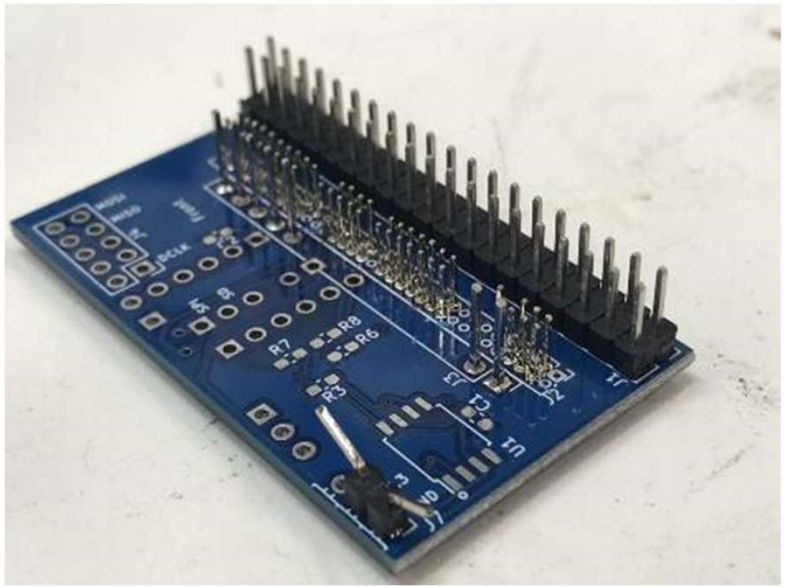
- No support for on-chip logic analyz
  - Xilinx: ChipScope
  - Intel: SignalTap



Adapter Board

- HWIC connector has 50mil pin spacing
- Converts from 50min to 100mil pin header
- ATTiny + SPI flash for self-hosting support
  - Work in Progress
- ~$10 for 20 PCBs
  - Ask me: I'll send one to you for free!
- Improvements possible:
  - PMODs
  - Mounting holes
  - On-board power regulator





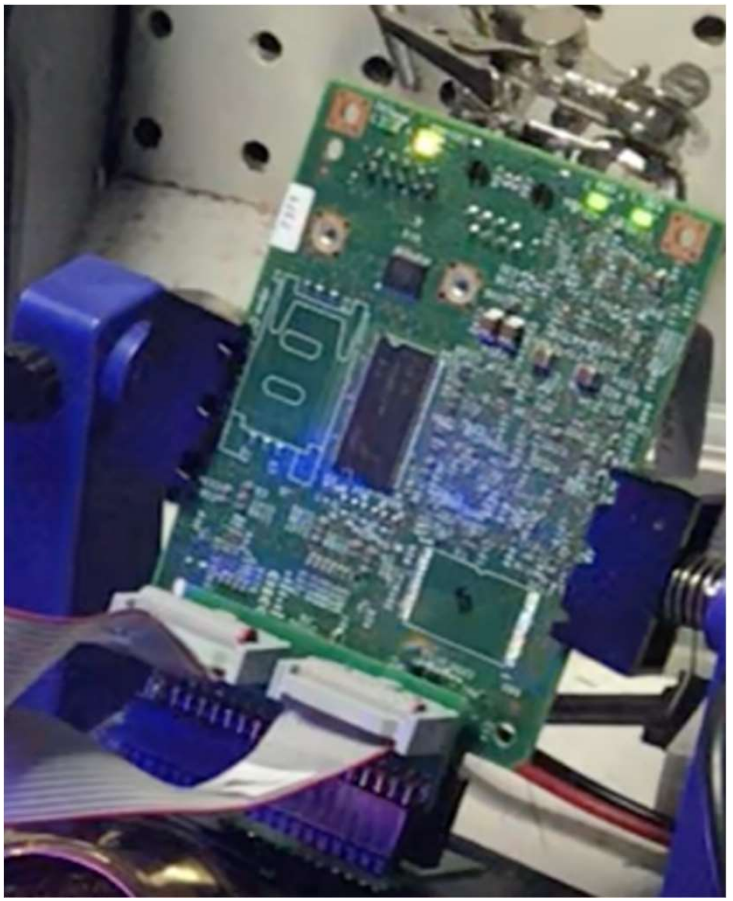
Still to Do

- Make self-hosting with ATtiny
  - Reading from SDcard already works
- Bring up NOR Flash
  - Parallel flash -> fast and easy to use for CPU code
  - Already soldered on board
- Bring up DDR2 DRAM
  - Supported by Quartus Free version, but only in tethered mode.
  - Requires $$$ license for untethered usage
  - Use Litex DDR2 Controller?
- Bringup NXP PCI to USB
  - Requires PCI interface
  - Lots of SW work…
  - Will probably never happen...

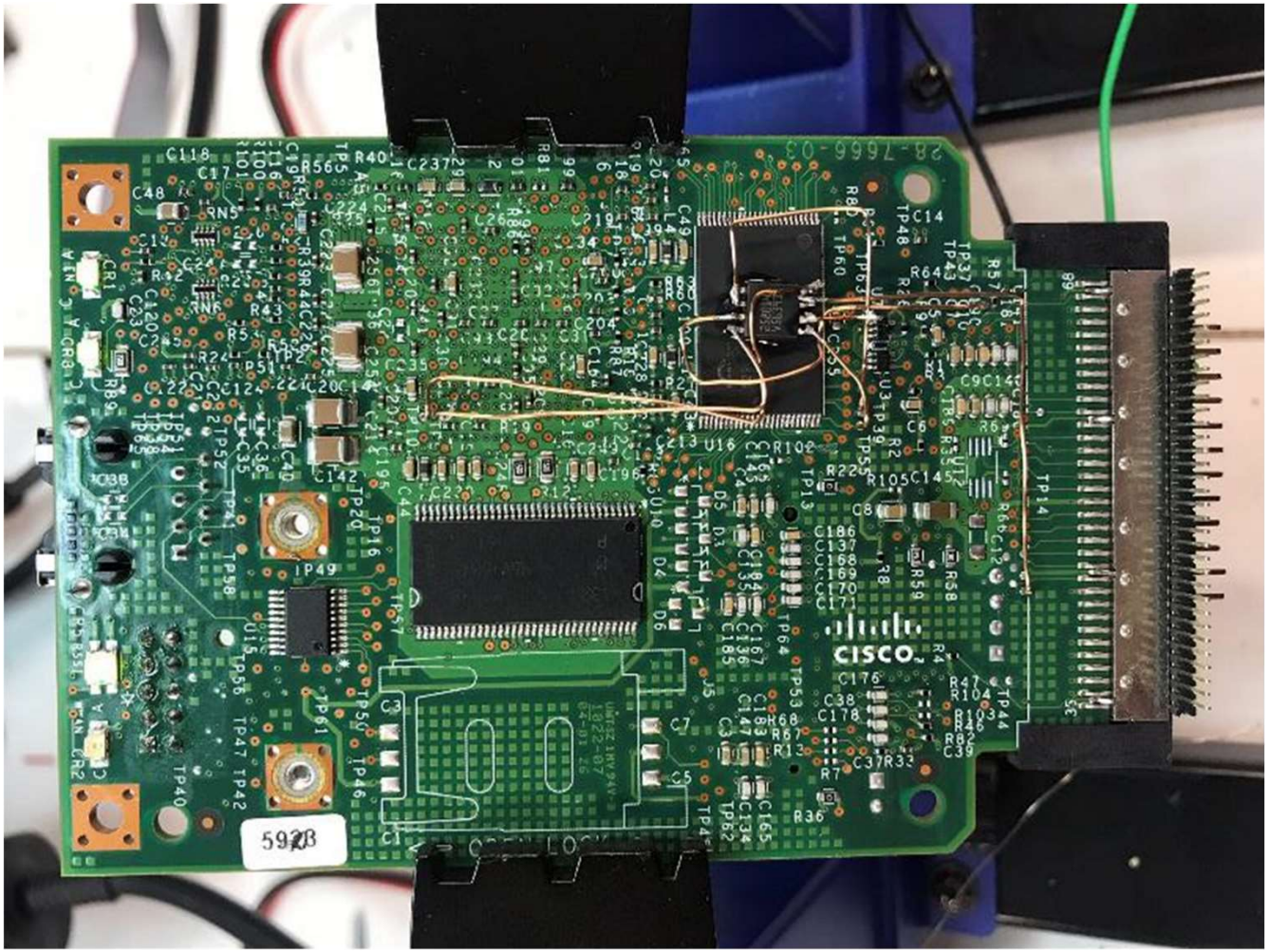
LED Cube Example Project

- Modem board
- 12 32x16 LED Matrices
  - Recovered from 12 Project MC2 Pixel Purses
- Adapter Board HWIC to GPIO + Adapter Board GPIO to 2x HUB75
- FPGA
  - VexRiscv CPU to generate content to display
  - LED DMA engine
    - Fetch LED values in right order
    - Binary encoding for color pallette
    - Gamma Correction
  - HUB75 timing controllers
- Still requires USB Blaster and laptop to download bitstream


898D
ENTER



Active Serial Retrofit



More Info

- My blog: https://tomverbeure.github.io
- My github: https://github.com/tomverbeure
- Twitter: @tom_verbeure
- HWIC-3G-CDMA specific info
  - Reverse Engineering the Cisco HWIC-3G-CDMA
    https://tomverbeure.github.io/2019/11/11/Cisco-HWIC-3G-CDMA.html
  - Reverse engineering info: pinout, adapter PCB, ...
    https://github.com/tomverbeure/cisco-hwic-3g-cdma
  - LED Cube project repo
    https://github.com/tomverbeure/cube
  - Cisco Purses Cubed presentation at !!Con West:
    https://www.youtube.com/watch?v=0tBU5-lJYmU

FPGA / Embedded Engineers

- Want to submit your project to talk about? Email
  - embeddedhour@adiuvoengineering.com

- Next one is on the 8th May 2020
  - Speakers are on PYNQ
  - Development of Hardware, FPGA board and supporting ecosystem