

Internet on Chip

Gigabit Ethernet and Programmable Hardware

Dr. Johannes Wolkerstorfer

www.xFace.at

Realraum Graz, January 26th 2010

x Face

Where hardware meets software

Motivation

How does Internet data look like?

- On the wire
- In the chip

How can packets be processed?

- What is the advantage of hardware

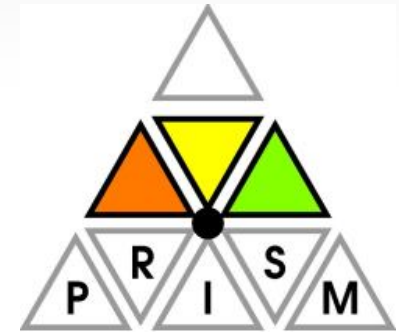
Real world example: PRISM project

- Privacy for Internet monitoring

PRISM application

PRivacy-aware Secure Monitoring

- <http://www.fp7-prism.eu/>
- European research project
 - Austrian partner: FTW (Forschungszentrum Telekommunikation Wien)



Goals

- Monitor network traffic at ISP
 - Detection of attacks
- Maintain privacy of user information (law)
 - e.g. IP address

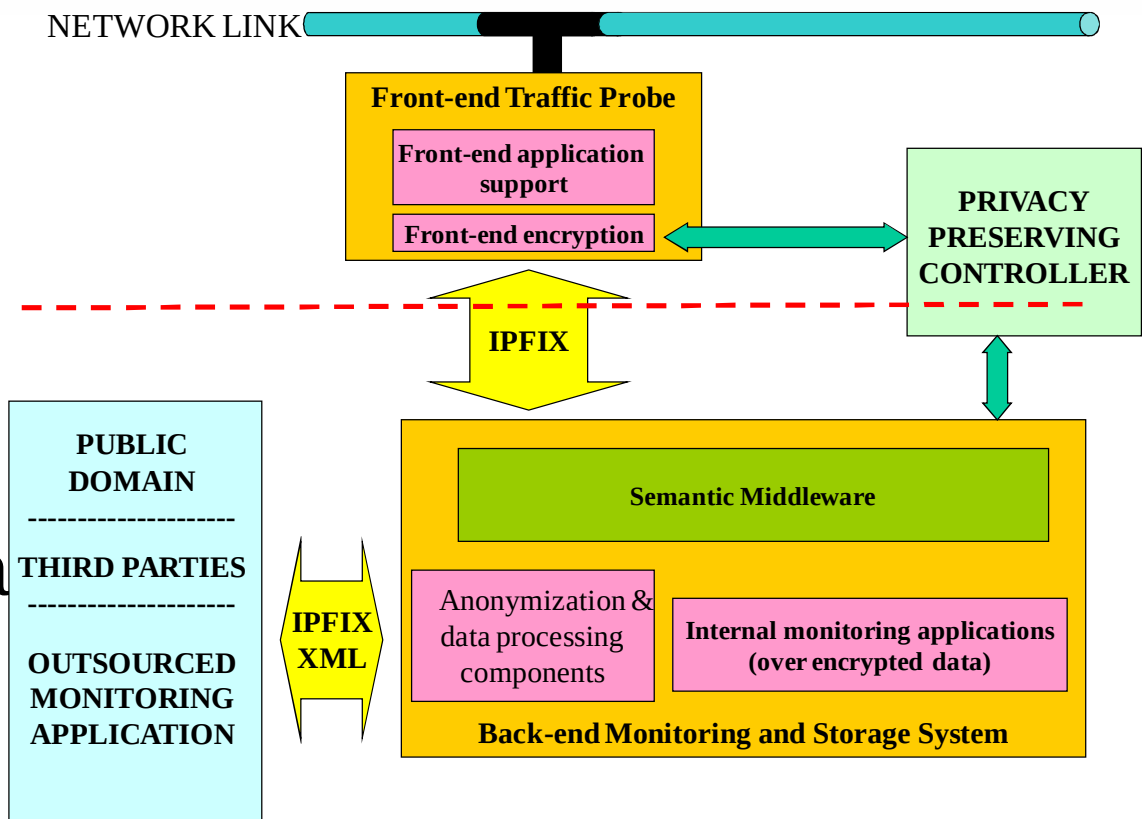
PRISM architecture

Frontend

- Monitoring of Internet traffic

Backend

- Storage of encrypted data
- interface to monitoring applications



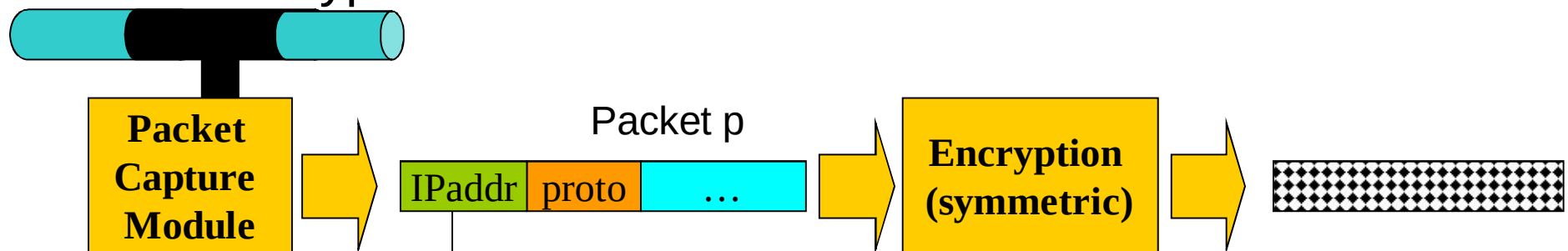
PRISM frontend

Functionality

- Flow extraction
 - classification of packets
 - normal | flow | anomaly
- Encryption

Requirements

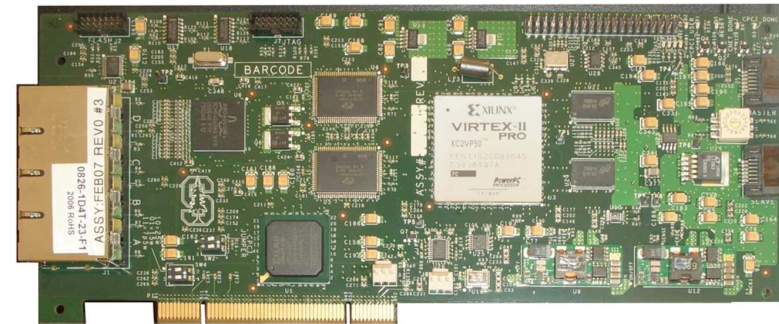
- Throughput: 1 Gbps
- Per-packet basis
 - stateless



NetFPGA project

NetFPGA

- FPGA board with
4 Gigabit Ethernet ports
- Opensource HW & SW
 - provided by Stanford Univ
- Applications
 - Router, switch, traffic gen.
 - Network security
 - Intrusion detection



NetFPGA Board

NetFPGA system

Host PC

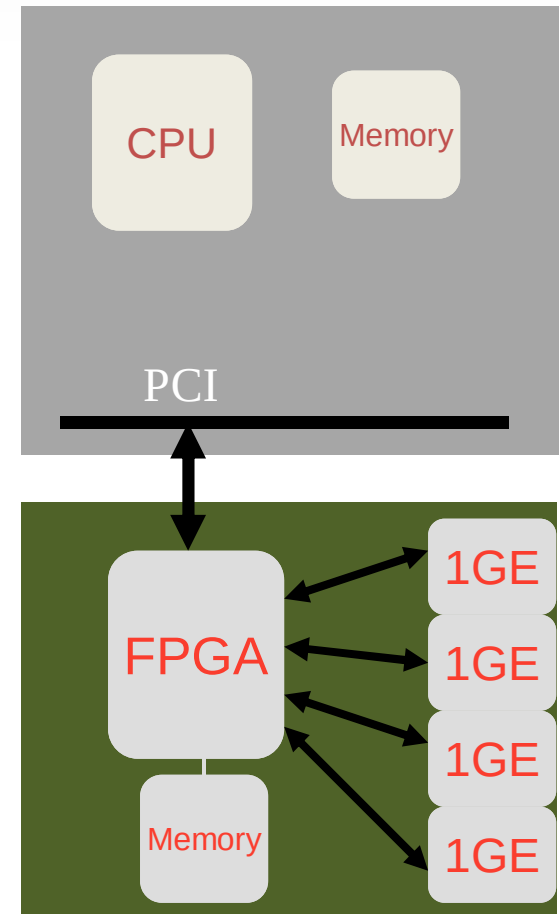
- Linux PC
- Kernel-mode driver
 - 4 additional network interfaces



PC with NetFPGA

PCI interface

- Network traffic
- Management of NetFPGA



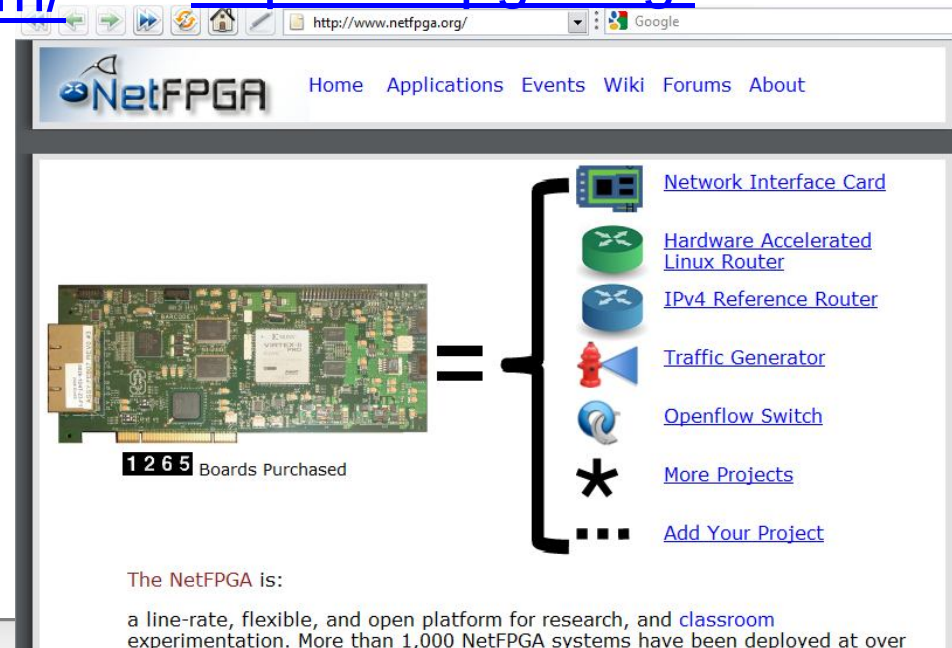
NetFPGA Where to get?

Hardware

- From Digilent
- 600 US\$ for research inst.
- <http://www.digilentinc.com/>

Software / HDL

- From Stanford University
- <http://netfpga.org/>



x Face

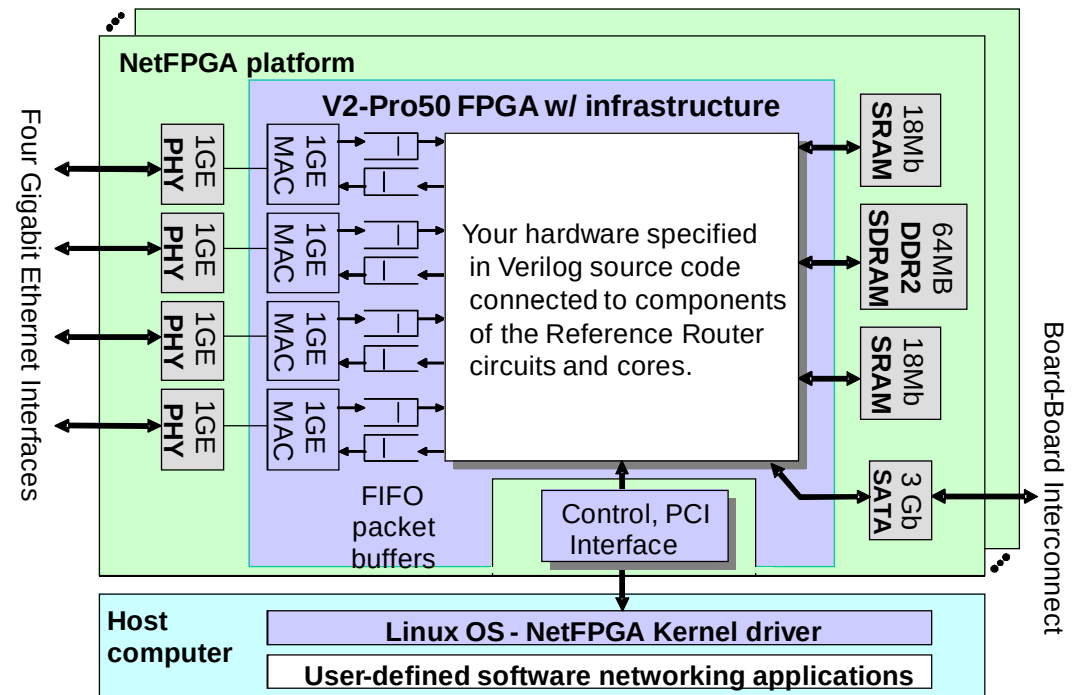
Where hardware meets software

Internet on Chip

NetFPGA architecture

NetFPGA components

- Xilinx Virtex-2 Pro FPGA for User Logic
- Xilinx Spartan for PCI Host Interface
- Cypress: 2 * 2.25 MB ZBT SRAM
- Micron: 64MB DDR2 DRAM
- Broadcom: 4 Gigabit Ethernet PHYs



FPGA (Xilinx Virtex-II Pro)

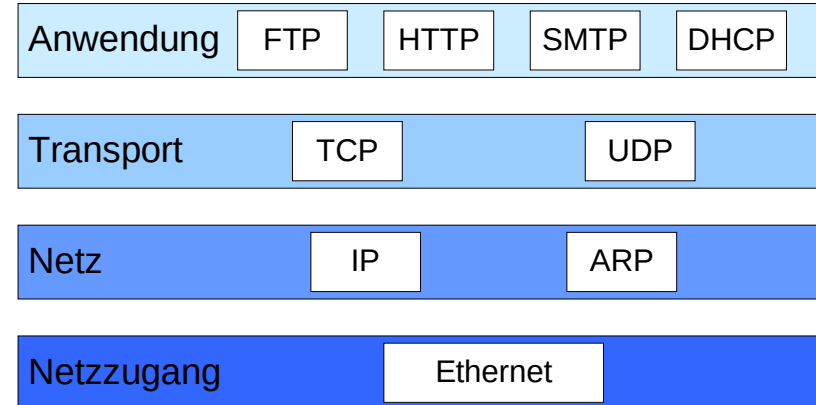
- Accommodates own project
- FPGA configured via PCI bus

Gigabit Ethernet

Packet switching network

OSI Schichtmodell

- TCP/IP: Transmission Control Protocol
 - Transport-Layer (Layer 4)
 - Gesicherte Datenverbindung
- IP: Internet Protocol
 - Network-Layer (Layer 3)
 - Datenvermittlung im Internet
- Ethernet (IEEE 802.3)
 - Data-Link Layer (OSI Layer 2)
 - Sicherung
 - Physical Layer (OSI Layer 1)
 - Bitübertragung



Interfaces: GMII for PHY

Ethernet frames

Protocol / Packet / Frame

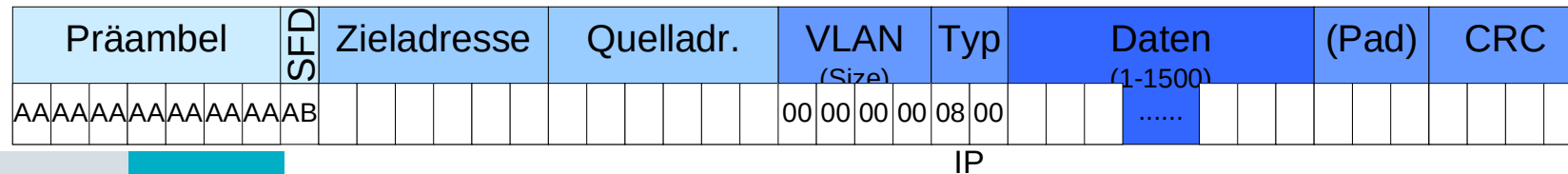
- Protocol: standard for communication
- Packet: Part of data
- Frame: Packet in transmission

Ethernet Frame

- Preamble
 - 56 Bit: alternating 0 und 1
 - For Synchronization
- SFD
 - Start of Frame Delimiter

Ethernet Frame (ctd)

- Source and target address
 - 48-Bit MAC-Address
 - Unique for every NIC
 - Manufacturer prefix
 - Do not mix up with IP address
- Paket-Typ
 - z.B. 0x0800: IP
- Prüfsumme
 - CRC



Ethernet MAC

FIFO: First in – First out

- Buffering of network data

MAC: Media access control

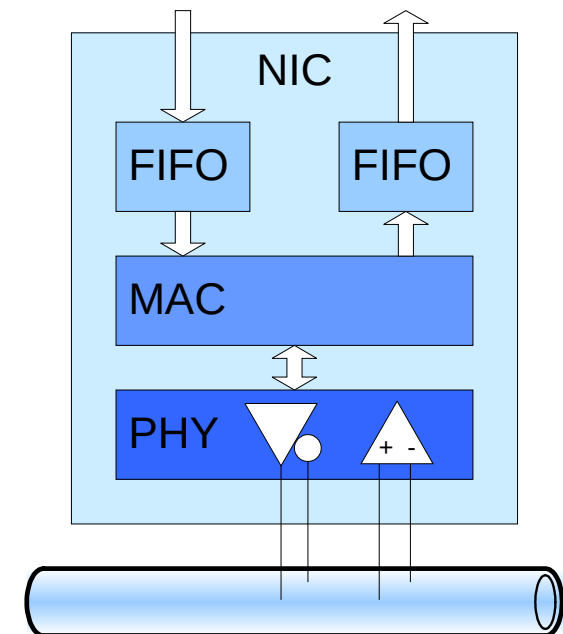
- Controls access to PHY
- Framing of packets
- CRC Checksum

PHY: Physical layer

- Electric standard
- Encoding of signals
- 1000 Base-T GbE: 4 twisted pairs



Foto ©
Realtek



Ethernet packets: Wireshark

Realtek PCIe GBE Family Controller: Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
61	6.141370	192.168.2.128	194.232.105.164	TCP	63621 > http [FIN, ACK] Seq=633 Ack=284 win=65410 Len=0
62	6.145878	194.232.105.164	192.168.2.128	TCP	http > 63620 [ACK] Seq=320 Ack=711 win=7090 Len=0
63	6.146116	194.232.105.164	192.168.2.128	TCP	http > 63621 [ACK] Seq=284 Ack=636 win=6974 Len=0
64	15.189001	192.168.2.128	88.167.9.172	UDP	Source port: 52083 Destination port: 38060
65	15.585956	88.167.9.172	192.168.2.128	UDP	Source port: 38060 Destination port: 52083
66	16.000572	194.232.104.29	192.168.2.128	TCP	http > 63617 [FIN, ACK] Seq=9572 Ack=1198 win=8276 Len=0
67	16.000659	192.168.2.128	194.232.104.29	TCP	63617 > http [ACK] Seq=1198 Ack=9573 win=65700 Len=0
68	16.006583	194.232.104.29	192.168.2.128	TCP	http > 63619 [FIN, ACK] Seq=1184 Ack=590 win=7020 Len=0
69	16.006636	192.168.2.128	194.232.104.29	TCP	63619 > http [ACK] Seq=590 Ack=1185 win=64516 Len=0
70	16.006837	194.232.104.29	192.168.2.128	TCP	http > 63618 [FIN, ACK] Seq=7030 Ack=597 win=7032 Len=0
71	16.006876	192.168.2.128	194.232.104.29	TCP	63618 > http [ACK] Seq=597 Ack=7031 win=65700 Len=0
72	16.100880	192.168.2.128	194.232.104.29	TCP	63617 > http [FIN, ACK] Seq=1198 Ack=9573 win=65700 Len=0
73	16.105238	194.232.104.29	192.168.2.128	TCP	http > 63617 [ACK] Seq=9573 Ack=1199 win=8276 Len=0
74	16.106901	192.168.2.128	194.232.104.29	TCP	63619 > http [FIN, ACK] Seq=590 Ack=1185 win=64516 Len=0
75	16.107831	192.168.2.128	194.232.104.29	TCP	63618 > http [FIN, ACK] Seq=597 Ack=7031 win=65700 Len=0
76	16.111555	194.232.104.29	192.168.2.128	TCP	http > 63619 [ACK] Seq=1185 Ack=591 win=7020 Len=0
77	16.112731	194.232.104.29	192.168.2.128	TCP	http > 63618 [ACK] Seq=7031 Ack=598 win=7032 Len=0

Frame 1 (153 bytes on wire, 153 bytes captured)

- Ethernet II, Src: AsustekC_a5:9e:e9 (00:26:18:a5:9e:e9), Dst: IPv6mcast_00:01:00:02 (33:33:00:01:00:02)
 - Destination: IPv6mcast_00:01:00:02 (33:33:00:01:00:02)
 - Source: AsustekC_a5:9e:e9 (00:26:18:a5:9e:e9)
 - Type: IPv6 (0x86dd)
- Internet Protocol Version 6
 - 33 33 00 01 00 02 00 26 18 a5 9e e9 86 dd 60 00 33.....&.....t
 - 00 00 00 00 63 11 01 fe 80 00 00 00 00 00 dd 74 ...c.....t
 - 0020 8e d6 a4 87 ee 97 ff 02 00 00 00 00 00 00 00 ..c.....t
 - 0030 00 00 00 01 00 02 02 22 02 23 00 63 fd 65 01 27".#.c.e.t
 - 0040 a8 a4 00 08 00 02 18 9c 00 01 00 0e 00 01 00 01t
 - 0050 12 3d 97 64 00 26 18 a5 9e e9 00 03 00 0c 10 00 ..=.d.&.....t
 - 0060 26 18 00 00 00 00 00 00 00 00 00 27 00 0d 00 0b &.....t
 - 0070 58 46 41 43 45 2d 4e 38 31 56 50 00 10 00 0e 00 XFACE-N8 1VP....t
 - 0080 00 01 37 00 08 4d 53 46 54 20 35 2e 30 00 06 00 ..7..MSF T 5.0...t
 - 0090 08 00 18 00 17 00 11 00 27t

Frame (frame), 153 bytes Packets: 77 Displayed: 77 Marked: 0 Profile: Default

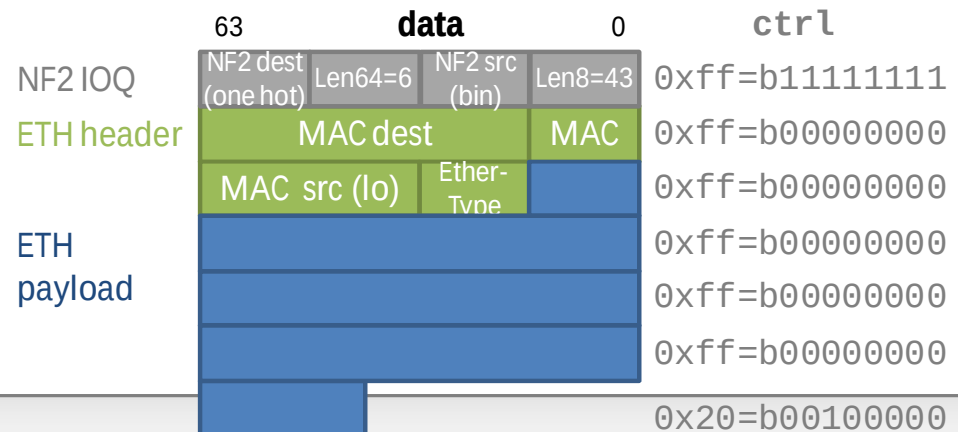
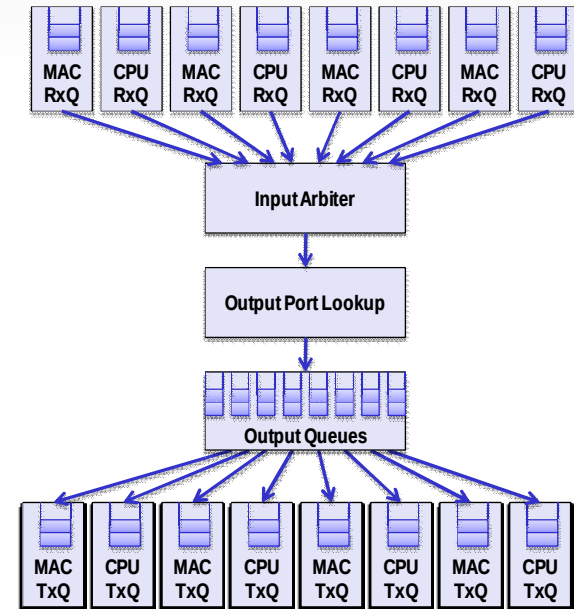
Ethernet packets in NetFPGA

GMII: Gigabit media independent interface

- connects to the PHY
- 8-bit interface; 125 MHz

user_data_path interface

- 64-bit interface, 125 MHz
- FIFO handshake



x Face

Where hardware meets software

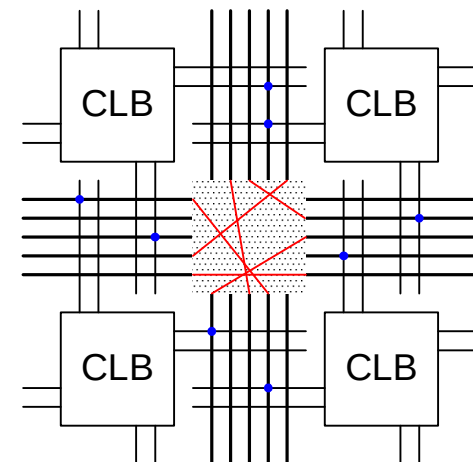
Internet on Chip

14

FPGA

Field programmable gate array (FPGA)

- Manufactured digital hardware
- Configurable hardware resources (CLB)
 - Lookup-table based logic
 - Flipflop for data storage
- Memory resources
 - block ram, distributed RAM
- Programmable interconnect



FPGA: Xilinx Virtex-II Pro

NetFPGA

– Xilinx Virtex-II Pro 50

Device ⁽¹⁾	RocketIO Transceiver Blocks	PowerPC Processor Blocks	Logic Cells ⁽²⁾	CLB (1 = 4 slices = max 128 bits)		18 X 18 Bit Multiplier Blocks	Block SelectRAM+		DCMs	Maximum User I/O Pads
				Slices	Max Distr RAM (Kb)		18 Kb Blocks	Max Block RAM (Kb)		
XC2VP2	4	0	3,168	1,408	44	12	12	216	4	204
XC2VP4	4	1	6,768	3,008	94	28	28	504	4	348
XC2VP7	8	1	11,088	4,928	154	44	44	792	4	396
XC2VP20	8	2	20,880	9,280	290	88	88	1,584	8	564
XC2VPX20	8 ⁽⁴⁾	1	22,032	9,792	306	88	88	1,584	8	552
XC2VP30	8	2	30,816	13,696	428	136	136	2,448	8	644
XC2VP40	0 ⁽³⁾ , 8, or 12	2	43,632	19,392	606	192	192	3,456	8	804
XC2VP50	0 ⁽³⁾ or 16	2	53,136	23,616	738	232	232	4,176	8	852
XC2VP70	16 or 20	2	74,448	33,088	1,034	328	328	5,904	8	996
XC2VPX70	20 ⁽⁴⁾	2	74,448	33,088	1,034	308	308	5,544	8	992
XC2VP100	0 ⁽³⁾ or 20	2	99,216	44,096	1,378	444	444	7,992	12	1,164



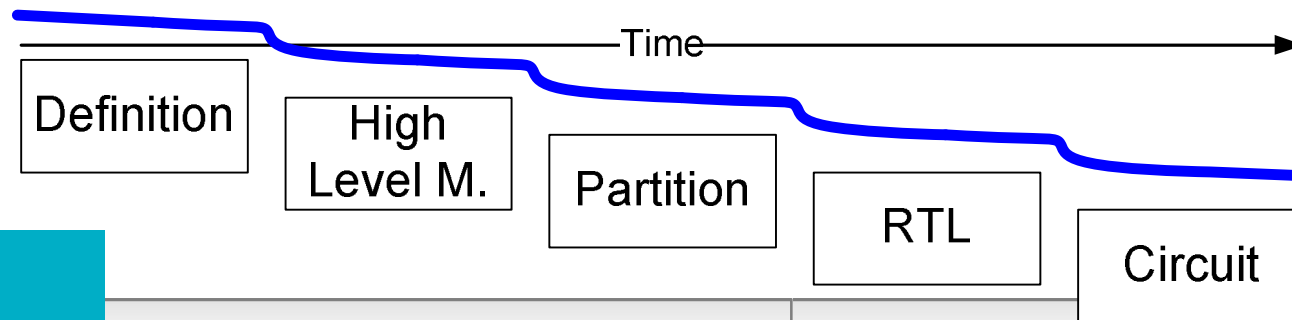
Design methodology

Top down design

- Informal specification
- High-level model: C/C++, Java
- HDL simulation
- Synthesis, place & route

No hardware debugging

- Sufficient testing of HDL model



Design tools: high level

High-level model: C/C++

- Functional model
- Performance not important
- Evaluation of algorithms

```
-----  
---  
---   Passing Ethernet packet through enckeyshare  
---  
  
Status of enckeyshare  
EncKeyshare Unit  
enc-S* = KEY1S= 0x00000000000000000000000000000000  
enc-S**= KEY2S= 0x00000000000000000000000000000000  
enc-AES IV =   0x00000000000000000000000000000000  
enc-ADP RAND = 0x00000000000000000000000000000000  
enc-COEFFS  =  0x0  
enc-XCOORD  =  0x0  
  
108 void enckeyshare::encryptPay  
109 //dbg  cout << "Encrypting  
110 payloadByte_t dil[ethe  
111 unsigned int dilLen = di  
112 // Output IV  
113 dout.addPayload(_keyStor  
114 printHex(_keyStore[IV],  
115 // Padding of input data  
116 din.getPayloadBytes(dil,  
117 //dbg  printHex(dil, " loa  
118 unsigned int blocks = _a  
119 printHex(dil, "enc-pad",  
120 // AES-128 CBC encryptio  
121 _aesCbc.StartEncryption(eth-ETHtype = 0800 (IP)  
122 _aesCbc.Encrypt(dil, dol  
123 _aesCbc.getIV(_keyStore[  
124 dout.addPayload(dol, blo  
125 }  
  
---> Input packet  
eth-MACdest = 01:02:03:04:05:06  
eth-MACsrc = aa:bb:cc:dd:ee:ff  
eth-ETHtype = 0800 (IP)  
eth-Payload = 4500002eb3fe000080110540c0a8002c  
c0a8000404000400001a2de800010203  
0405060708090a0b0c0d0e0f1011  
eth-CRC32 = ab187387 (Verified)  
  
enckeyshare processes Packet packet:  
  
---> Output packet --->  
eth-MACdest = 01:02:03:04:05:06  
eth-MACsrc = aa:bb:cc:dd:ee:ff  
eth-ETHtype = 0800 (IP)  
eth-Payload = 4500002eb3fe000080110540c0a8002c  
c0a8000404000400001a2de800010203  
0405060708090a0b0c0d0e0f1011  
eth-CRC32 = ab187387 (Verified)  
  
Passed.
```

Design tools: Verilog

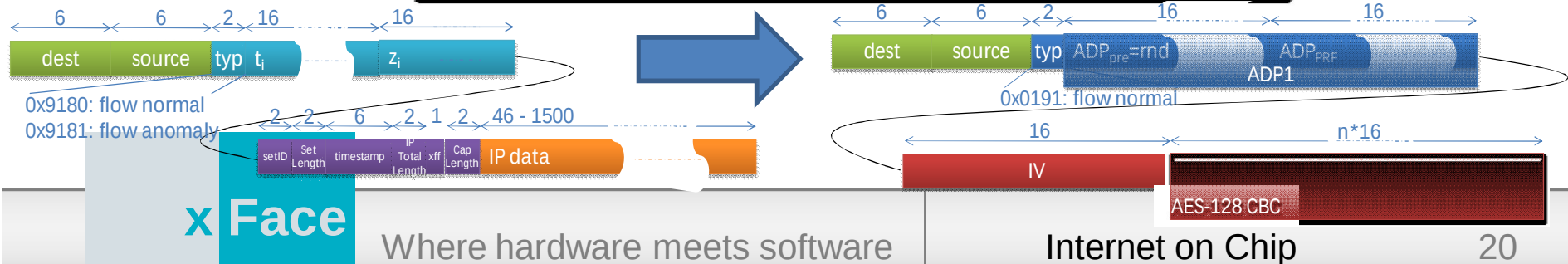
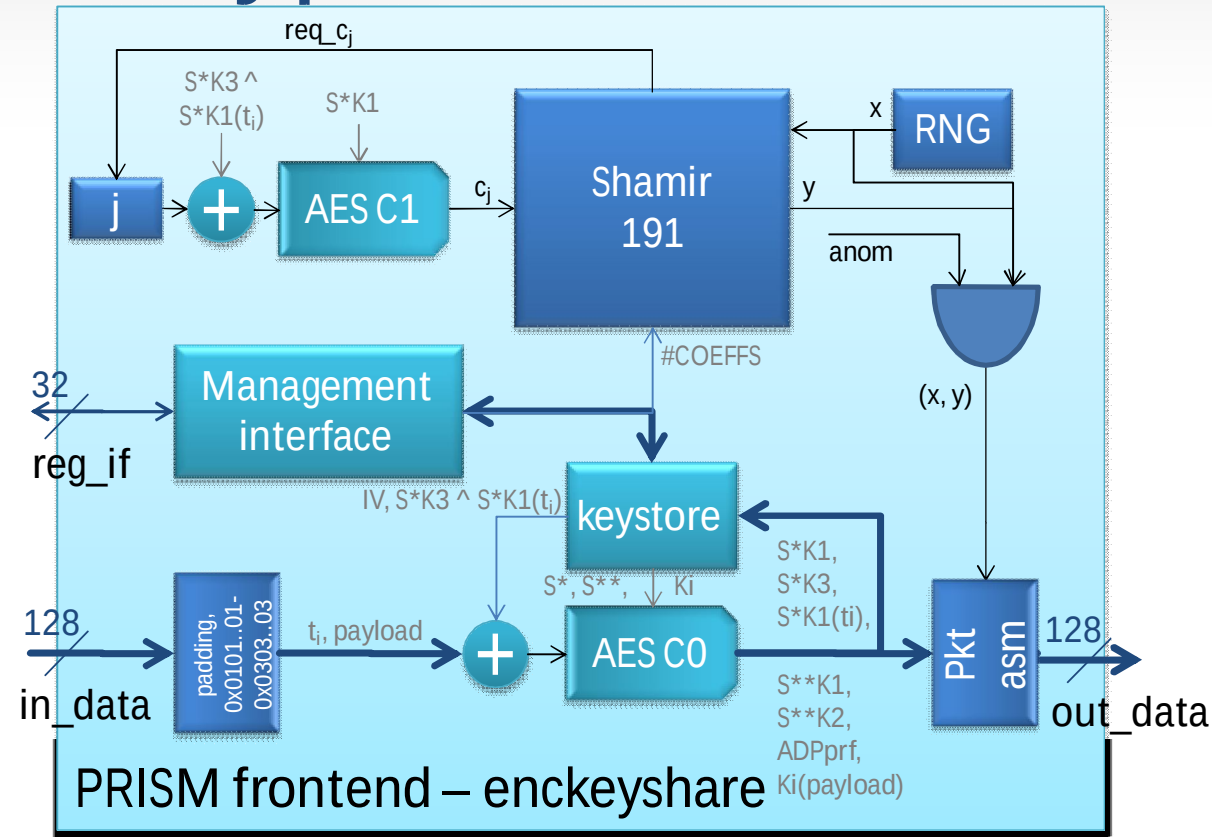
Modelling of digital hardware

- Hardware description language (HDL): Verilog

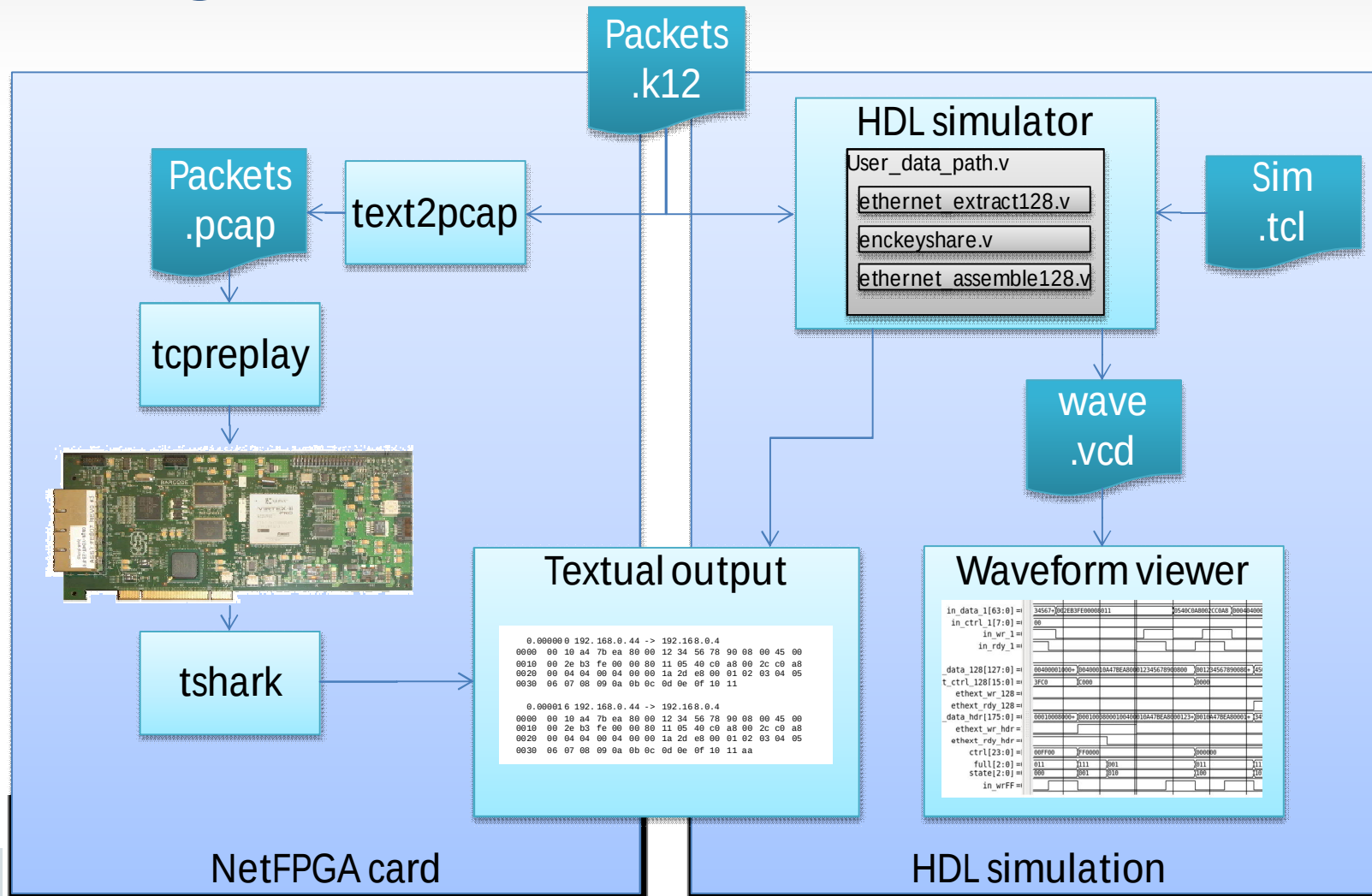
```
47
48 assign {sbyte[15],sbyte[14],sbyte[13],sbyte[12],sbyte[11],sbyte[10],sbyte[9],sbyte[8],sbyte[7],sbyte[6],sbyte[5],sbyte[4],sbyte[3],sbyte[2],sbyte[1],sbyte[0]} = {mixcol[3],mixcol[2],mixcol[1],mixcol[0],mixcol[3],mixcol[2],mixcol[1],mixcol[0],mixcol[3],mixcol[2],mixcol[1],mixcol[0],mixcol[3],mixcol[2],mixcol[1],mixcol[0]};
49
50
51 assign state = {sbyte[15],sbyte[14],sbyte[13],sbyte[12],sbyte[11],sbyte[10],sbyte[9],sbyte[8],sbyte[7],sbyte[6],sbyte[5],sbyte[4],sbyte[3],sbyte[2],sbyte[1],sbyte[0]};
52
53
54 always @(posedge clk or posedge reset) begin
55     if (reset) begin
56         rlastFF <= 0;
57         out_wrFF <= 0;
58     end else begin // posedge clk
59         if (rlastFF) dataout <= state;
60         rlastFF <= rlast;
61         out_wrFF <= out_wrFF && ~out_ack || rlast;
62     end // posedge clk
63 end // always
```

```
in_wr_data = 0;
state[4:0] = 0;
in_key[127:0] = 0;
in_data[127:0] = 0;
in_wr_iv = 0;
in_iv[127:0] = 0;
rcnt[3:0] = 0;
c0_rlast = 0;
rkey[127:0] = 0;
state[127:0] = 0;
out_data[127:0] = 0;
out_wr = 0;
out_ack = 0;
in_key[127:0] = 0;
k[127:0] = 0;
in_ack = 0;
in_wr = 0;
cnt[3:0] = 0;
```

PRISM: Crypto hardware



Design tools: Verification



Conclusions

Gigabit Ethernet and digital hardware

- Can be done with FPGAs
- Hardware and Software nearly affordable

Results

- 1 Gigabit Ethernet encryption: AES-128 CBC
- Computation of Shamir keyshares

Feature	Value	Remark
BRAM	18 (of 232: 7%)	two AES cores (c0, c1) and one round-key generator
slices	3749 (of 23616: 15%)	mostly caused by Shamir unit (191*16 multiplier)
f_max	165.4 @ MHz	XST synthesis result
Throughput	1 GBit/s	maximum throughput @ f_clk=125 MHz
Latency	75 cycles (600 ns)	delay caused by enckeyshare; required interframe gap