Real Time Packet Processing with FPGAs

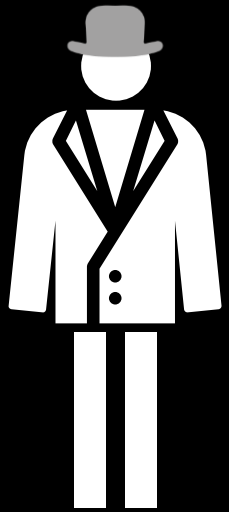
A network security toolbox with encryption features designed for FPGA logic-fabrics

Hack In The Box Dubai – 2018

Matteo Collura

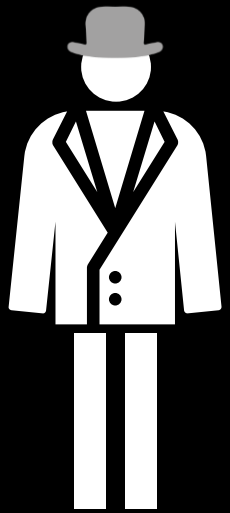
About me

- M.Sc in Micro and Nanotechnologies for Integrated systems
- B.Sc in Electronics Engineering



About me

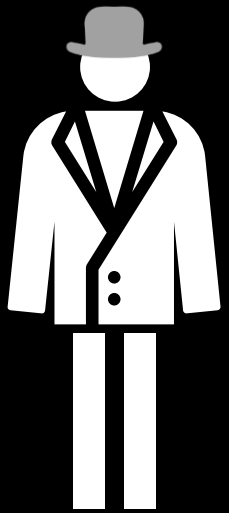
- Amateur and passionate about IT Security
- Speaker at International security conferences since 2013



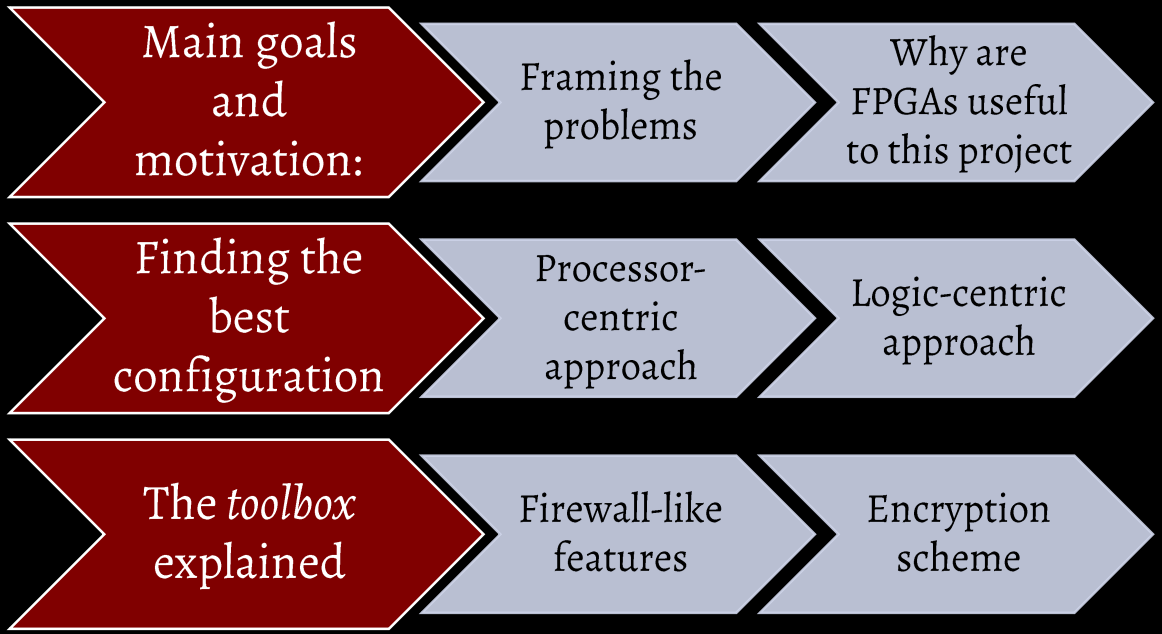
About me

- Worked for 6-month Master's Thesis Project at Knowledge Resources GmbH

www.knowres.ch



Outline

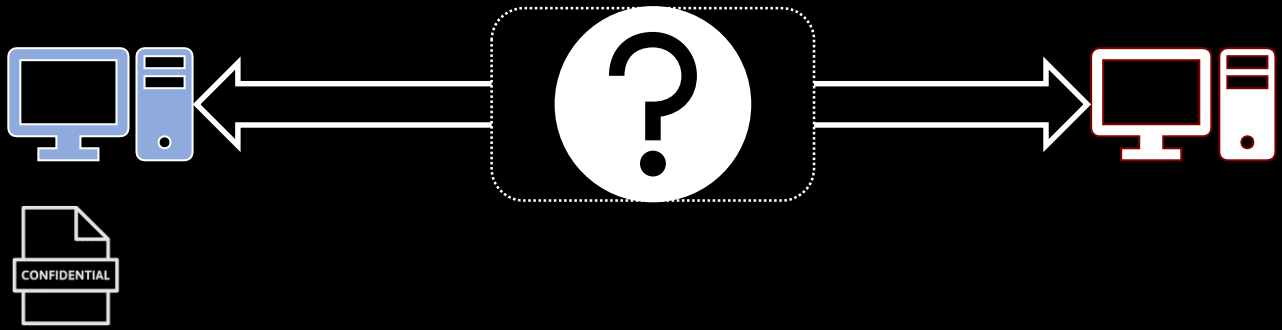

Main goals and motivation:
Framing the problems
Why are FPGAs useful to this project
Finding the best configuration
Processor-centric approach
Logic-centric approach
The *toolbox* explained
Firewall-like features
Encryption scheme

Outline

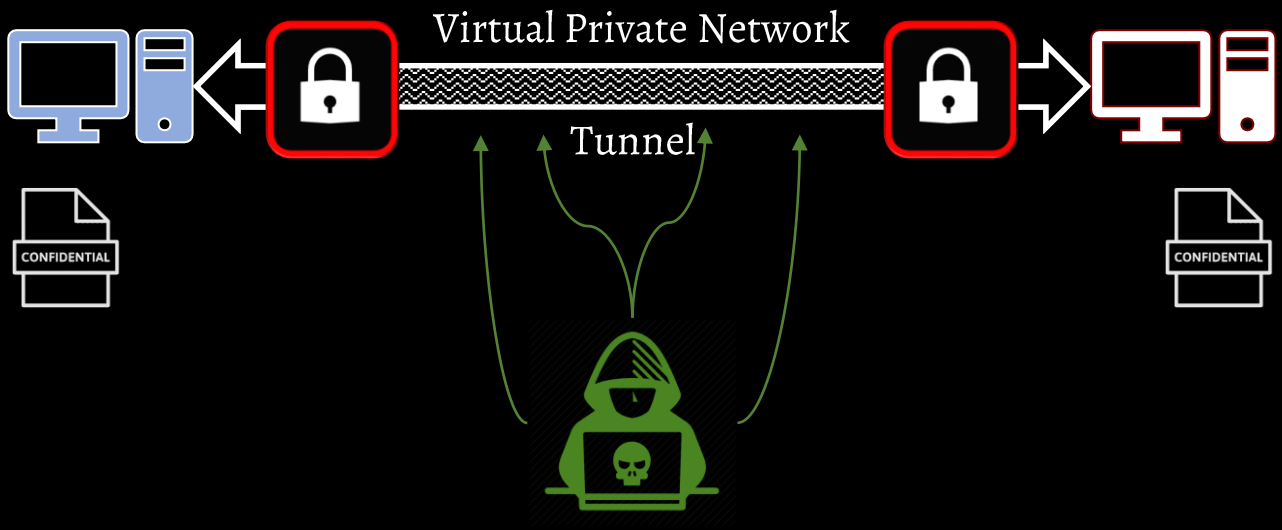
Results and Demo

Follow me @ Armory to get more insights and see the code

How to deliver confidential information securely?



VPN

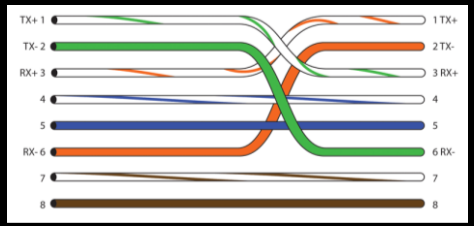

Virtual Private Network
Tunnel
CONFIDENTIAL
CONFIDENTIAL

VPN

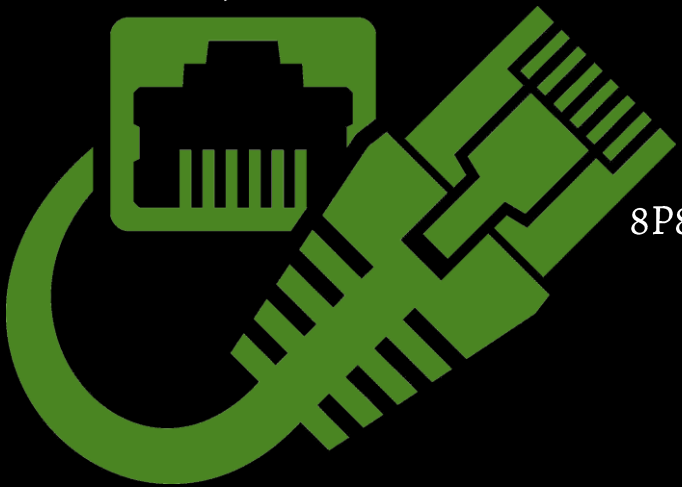
Real Time?

Ethernet connections run at 1Gbps commonly

IEEE 802.3 standard



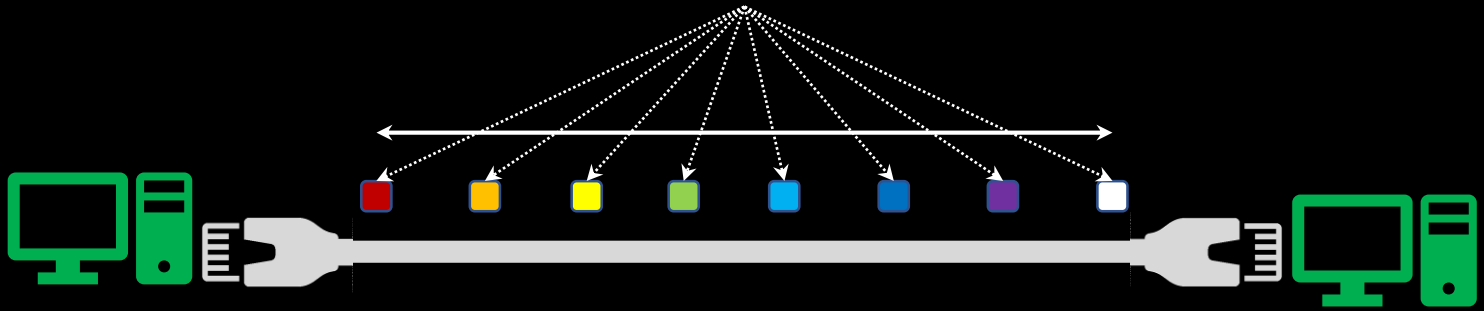
RJ45



8P8C

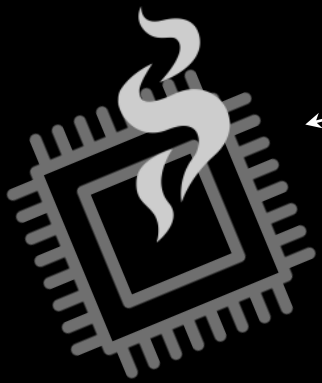
Ethernet connections run at 1Gbps commonly

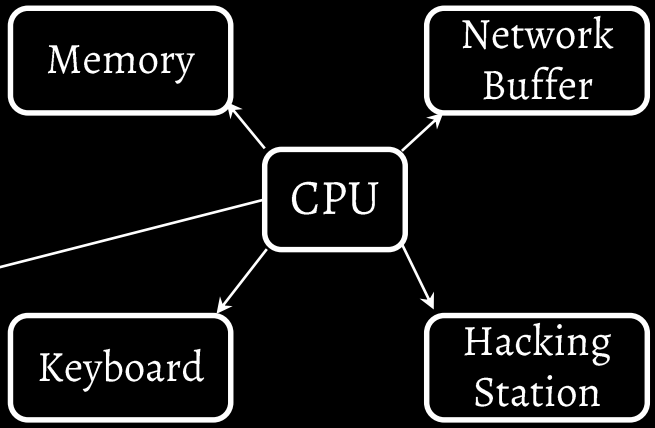
1 Byte every 8 ns



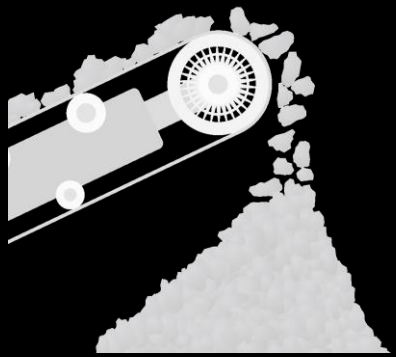
Von Neumann architecture

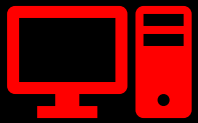
CPU speed ≈ GHz

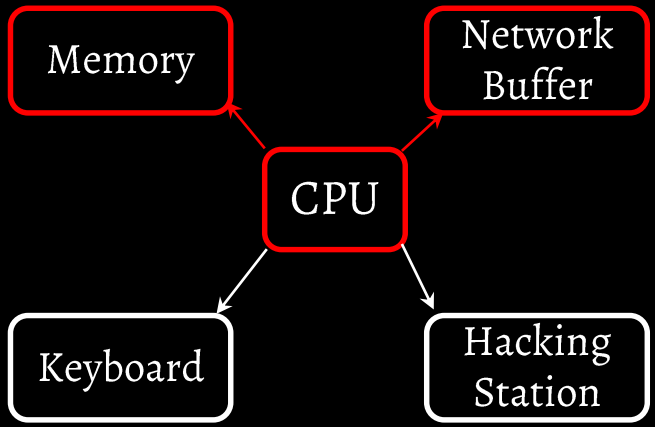




Von Neumann Bottleneck



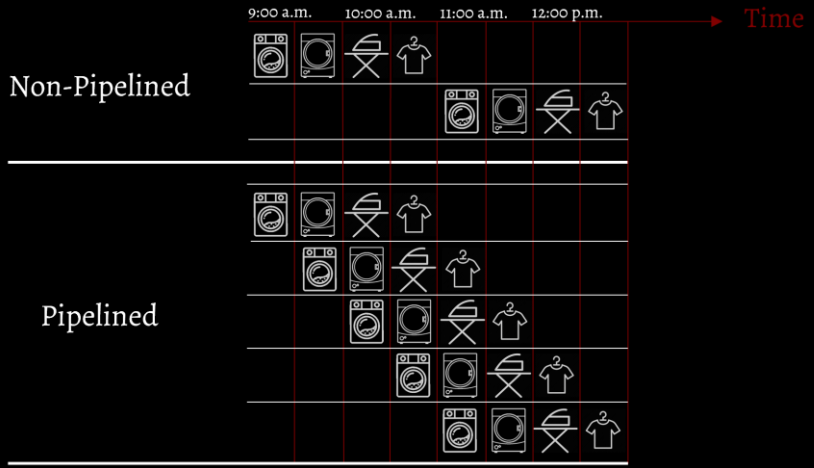




FPGAs: a Non-Von Neumann class of devices

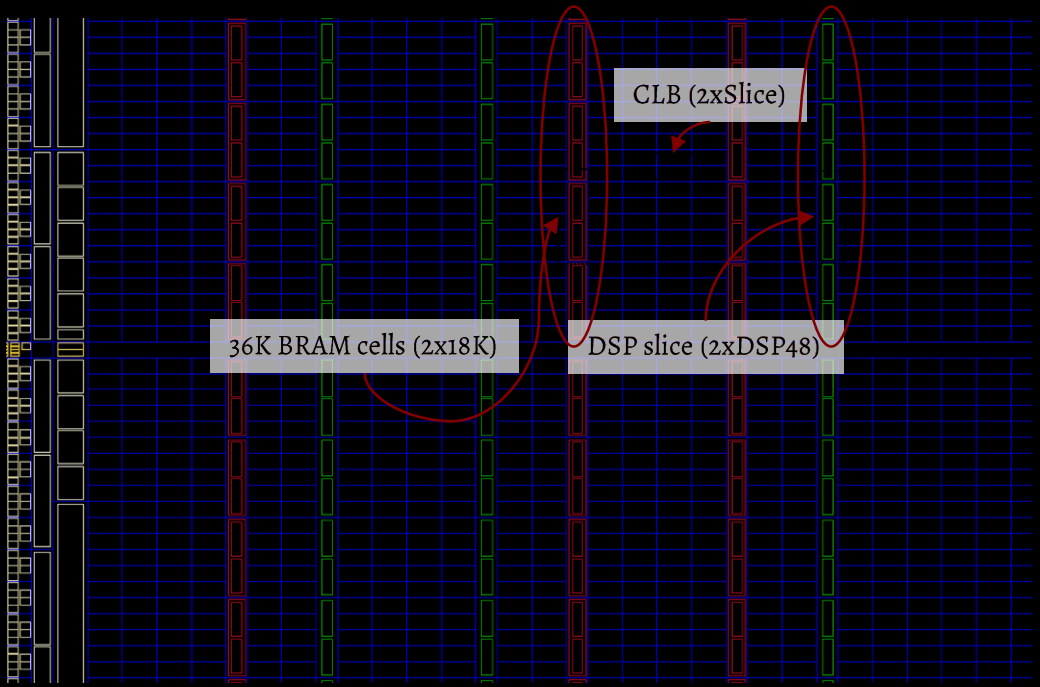
Picture taken from knowres.com

FPGAs: a Non-Von Neumann class of devices



Picture taken from knowres.com

FPGAs: what's inside



FPGAs: Xilinx fabric families

Performances

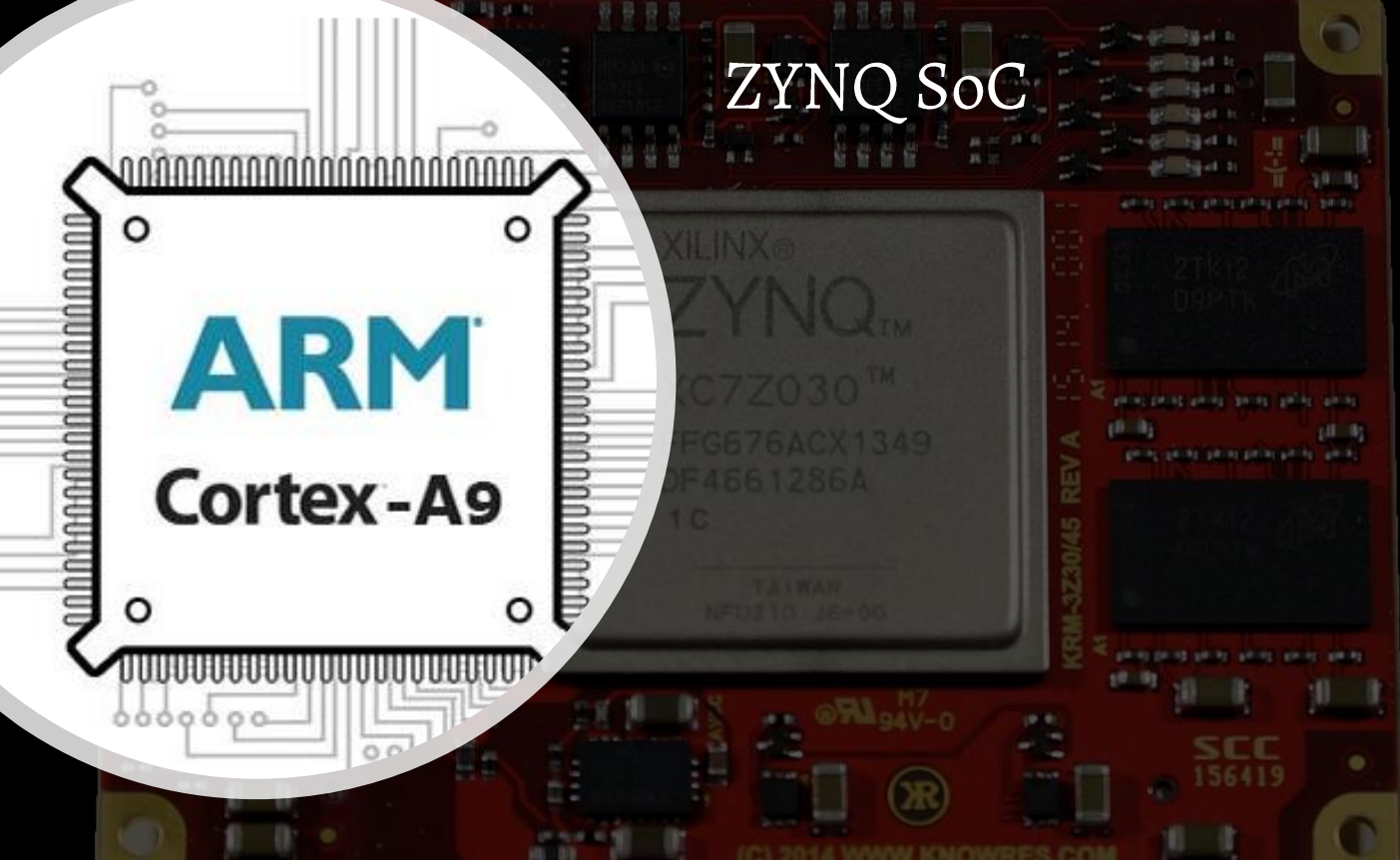
Spartan

Artix

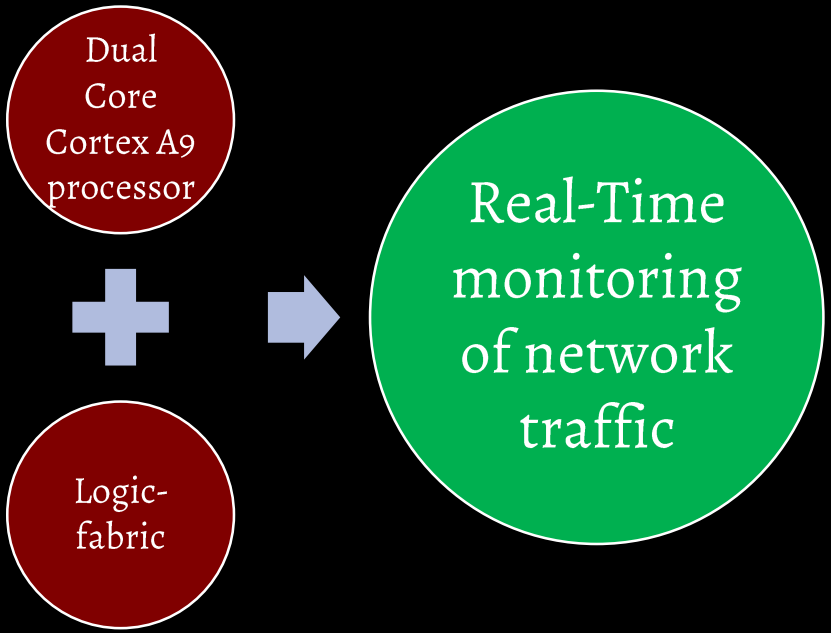
Kintex

Virtex

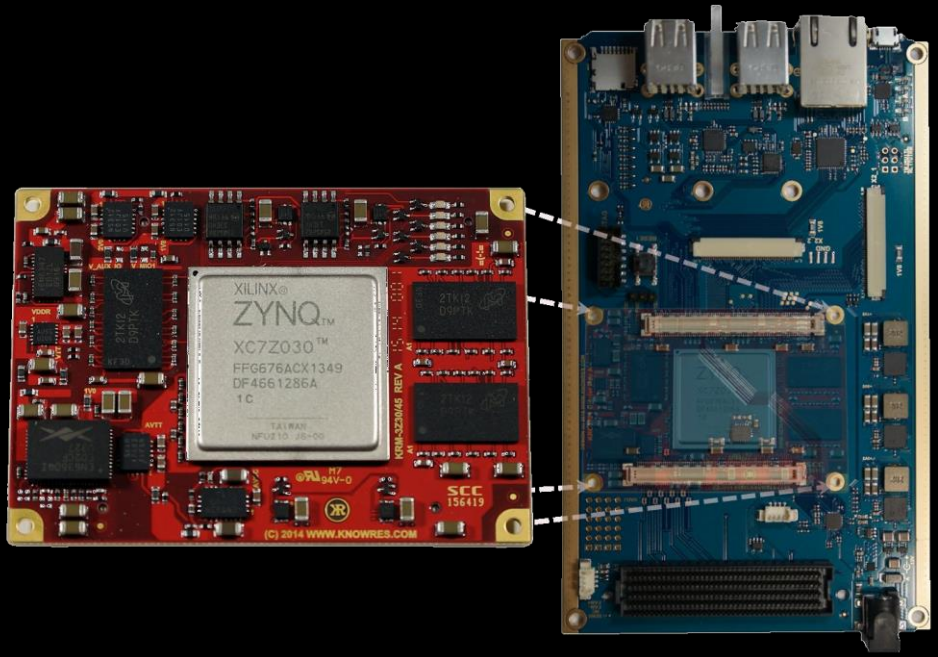
ZYNQ SoC


ARM
Cortex-A9
XILINX
ZYNQ
C7Z030
FG676ACX1349
TAIWAN
KRM-3Z30/45 REV A
SCC
156419

Summarizing

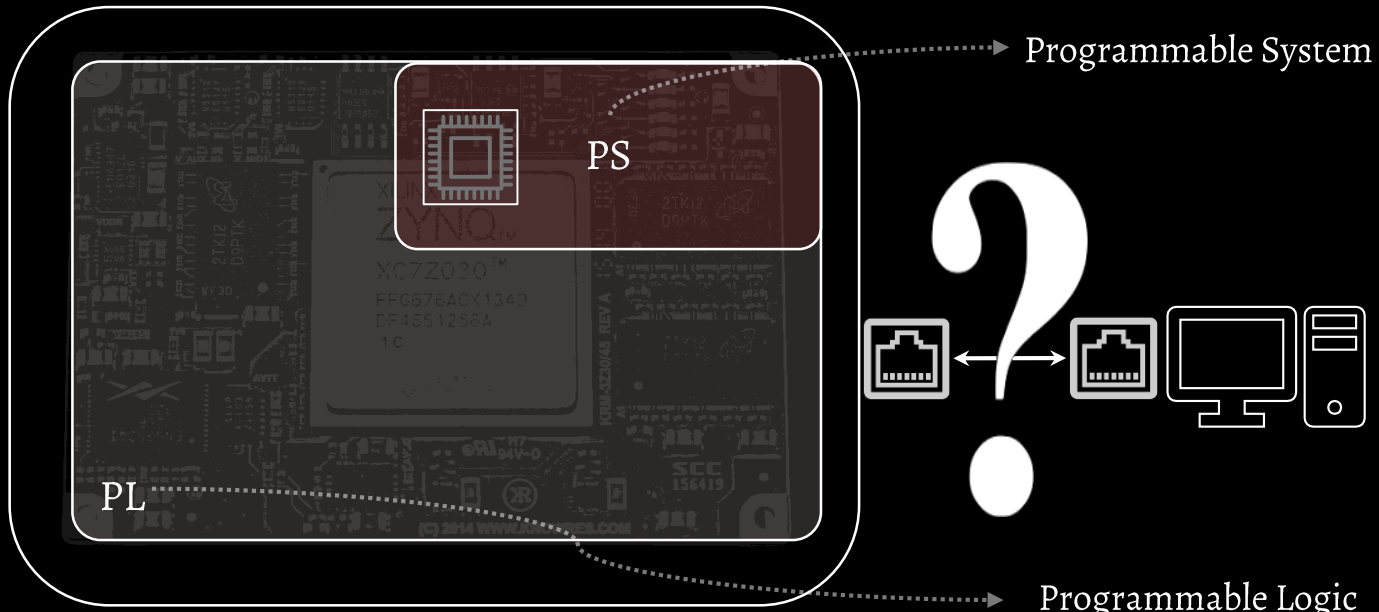


Carrier Evaluation Kit

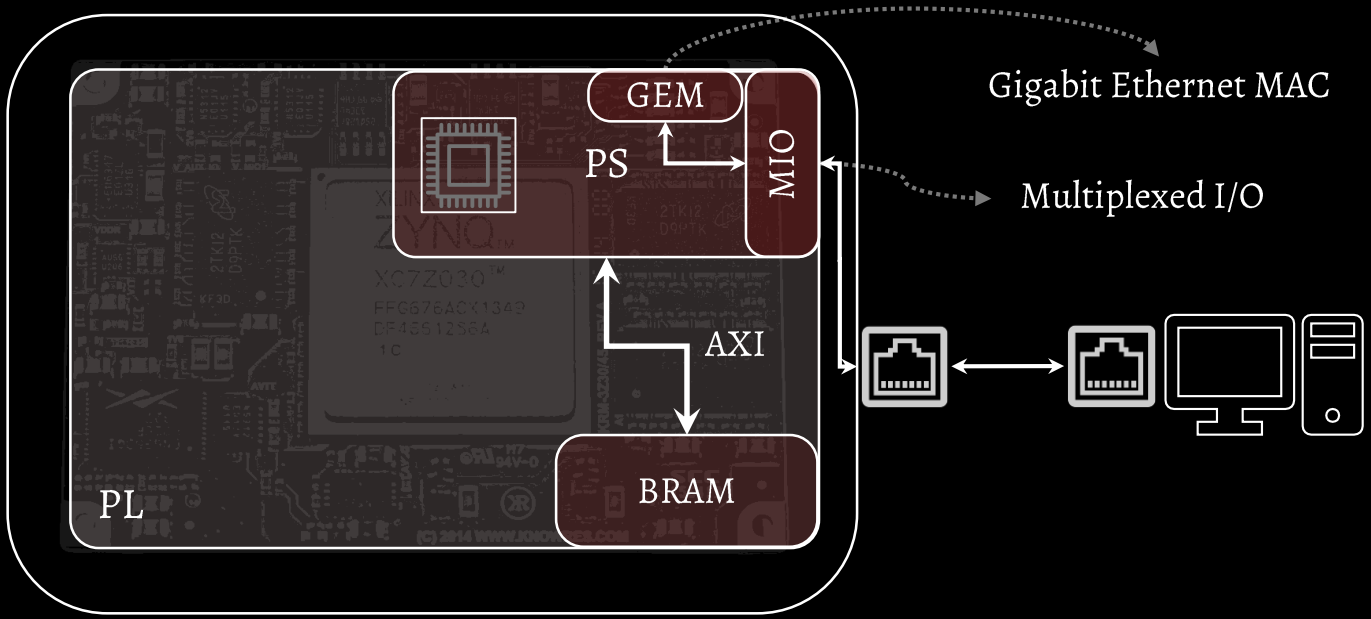


Picture taken from knowres.com

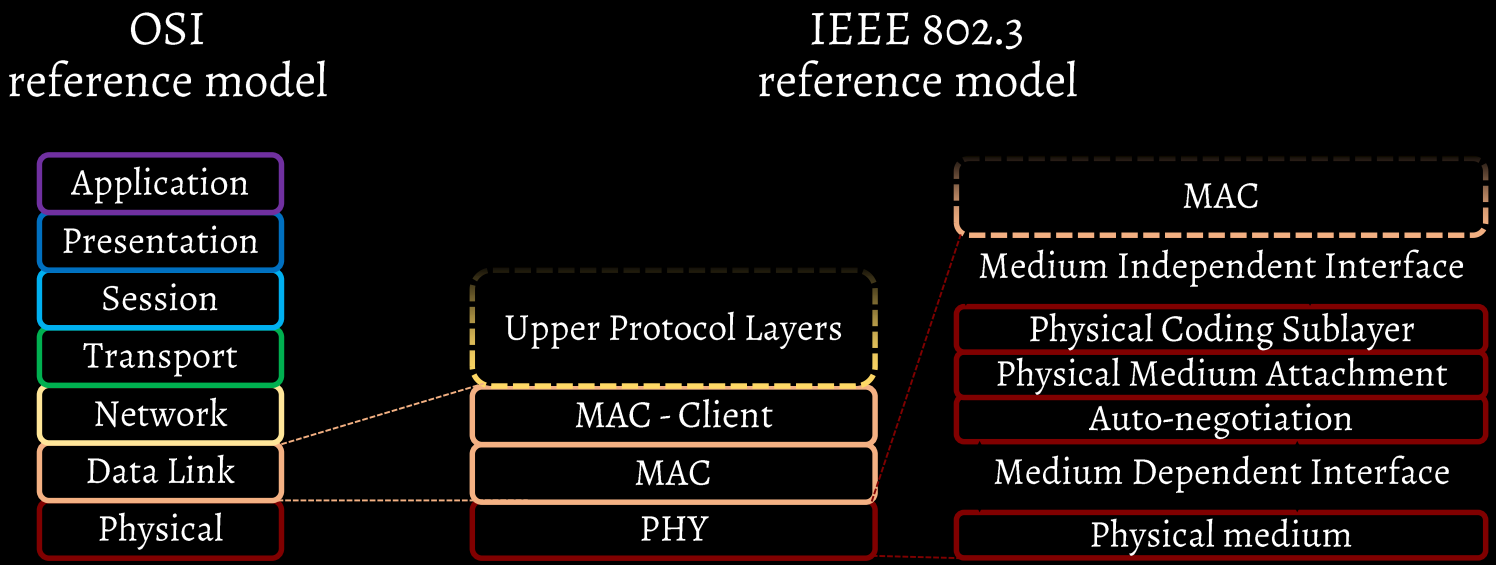
ZYNQ architecture



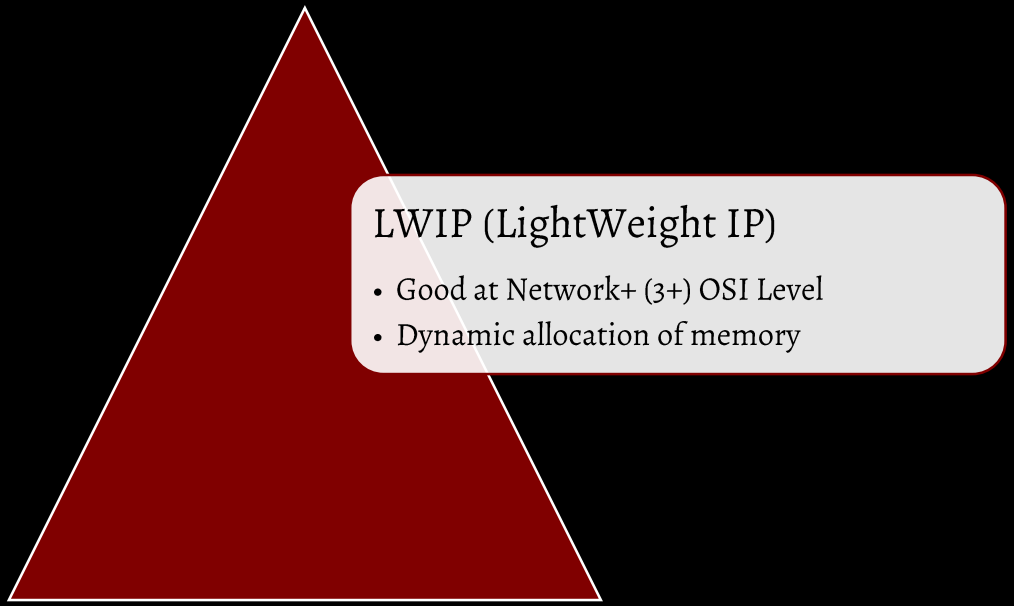
Routing Network Connections



ISO/OSI


OSI
reference model
Application
Presentation
Session
Transport
Network
Data Link
Physical
IEEE 802.3
reference model
Upper Protocol Layers
MAC - Client
MAC
PHY
MAC
Medium Independent Interface
Physical Coding Sublayer
Physical Medium Attachment
Auto-negotiation
Medium Dependent Interface
Physical medium

Useful libraries to implement network applications



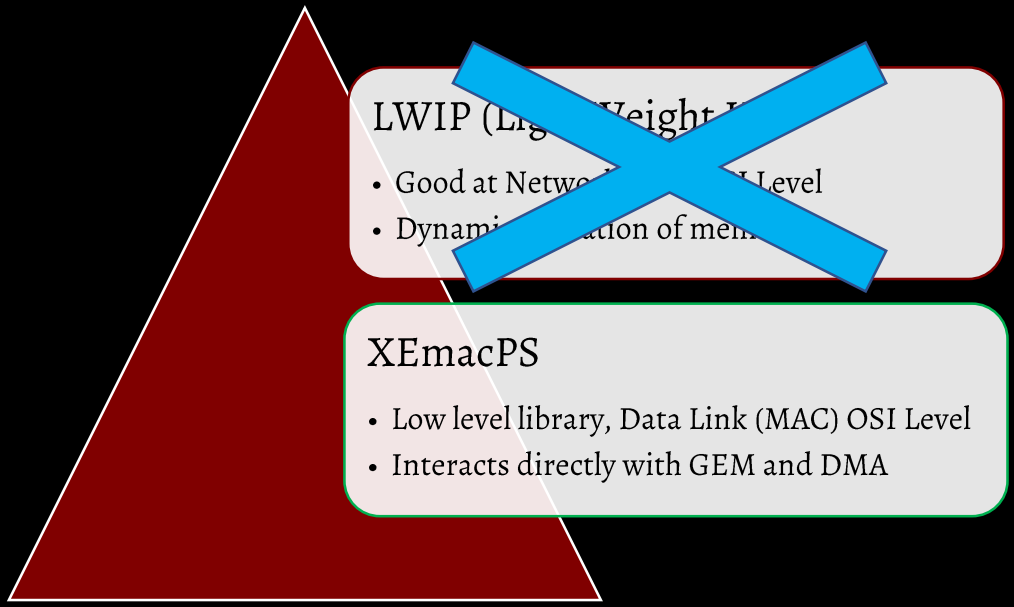
LWIP (LightWeight IP)

- Good at Network+ (3+) OSI Level
- Dynamic allocation of memory

Too high level of abstraction

The processor is always busy

Useful libraries to implement network applications



LWIP (Li eight I

- Good at Netwo Level
- Dynami ation of mem

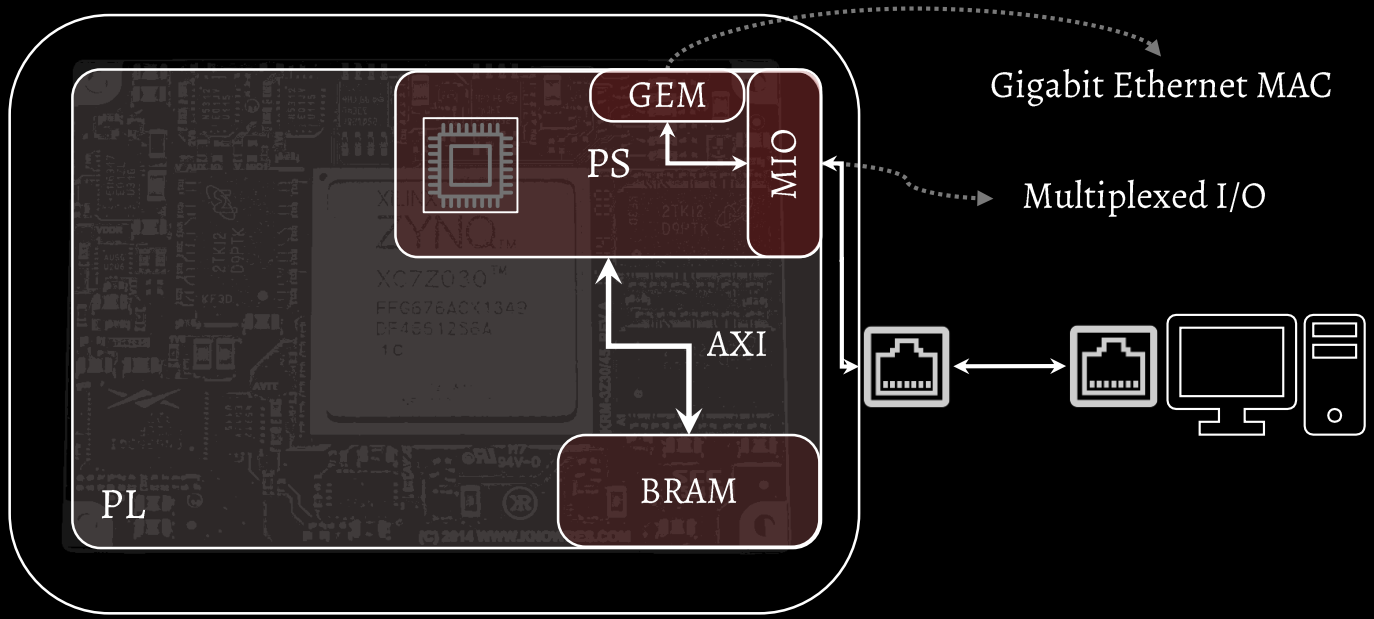
XEmacPS

- Low level library, Data Link (MAC) OSI Level
- Interacts directly with GEM and DMA

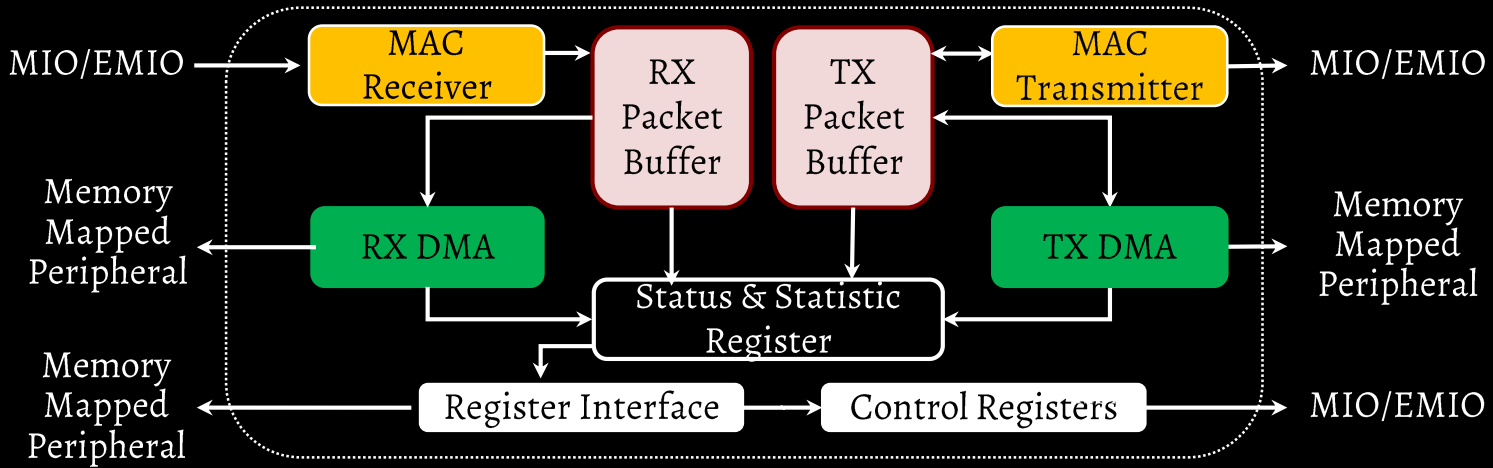
Captures ALL the packets

A bit complicated

Routing Network Connections



Gigabit Ethernet MAC



Gigabit Ethernet MAC

RX/TX Buffer

- Fills up quickly
- Packet size is overestimated but well organized structure

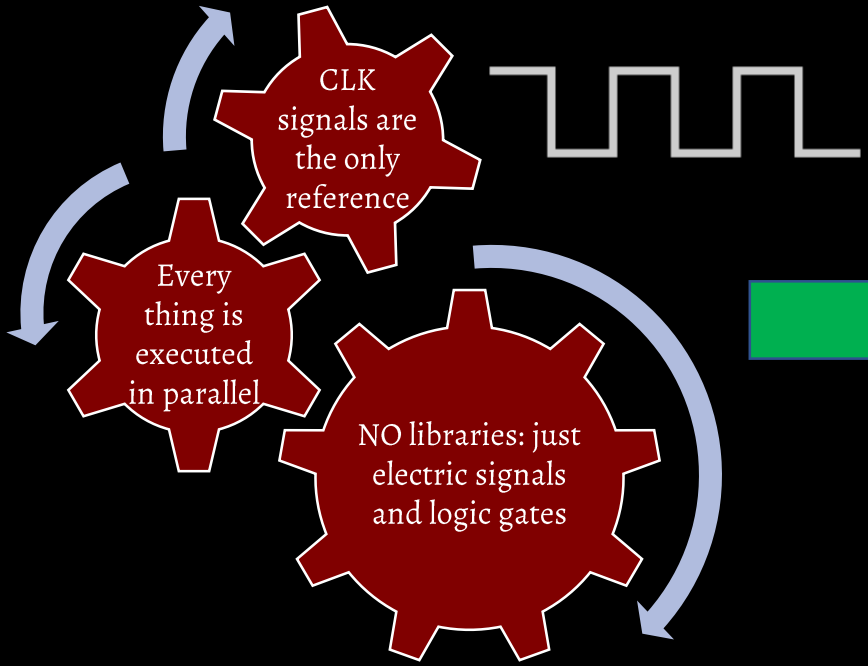
DMA Engine

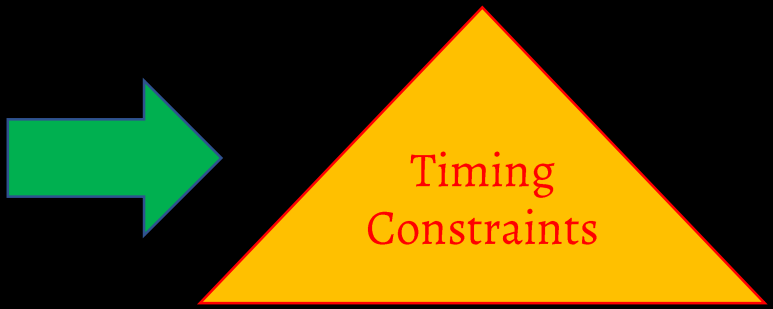
- 4-state Finite State Machine
- Driven by ARM processor

BRAM

- “Slow” (Hundreds of MHz)
- Too many blocks will mess up the timing constraints.

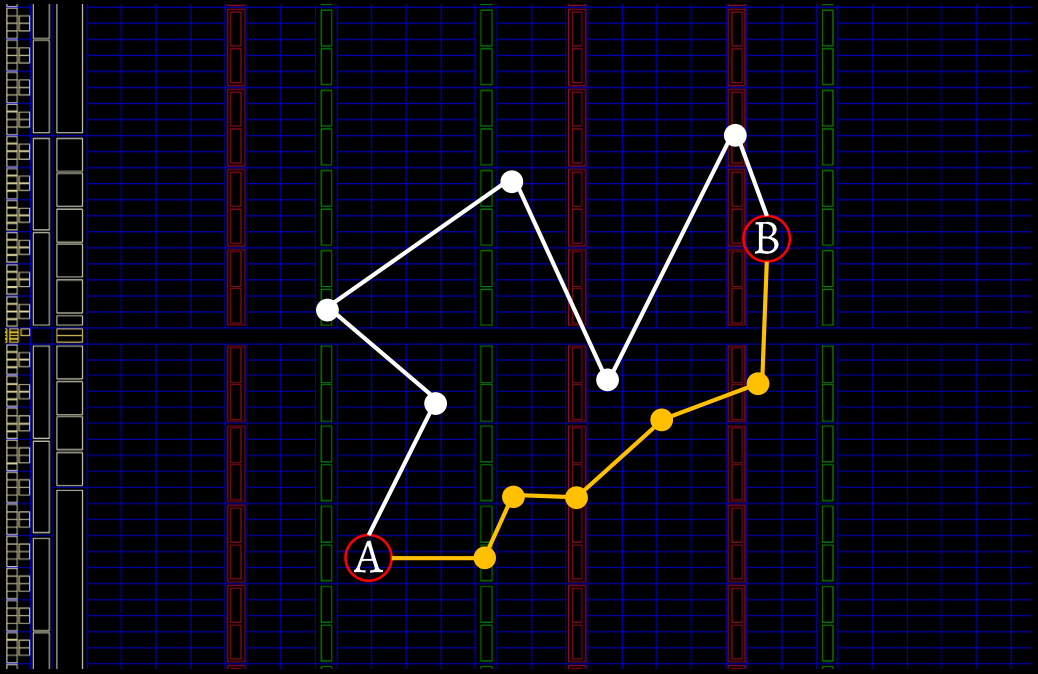
Towards a fully hardware implementation



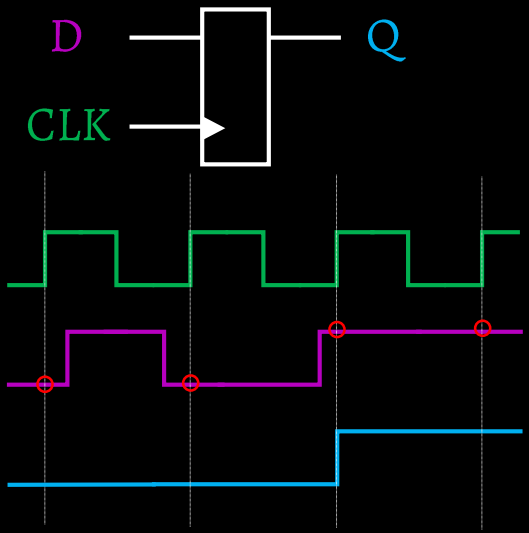


Timing Constraints

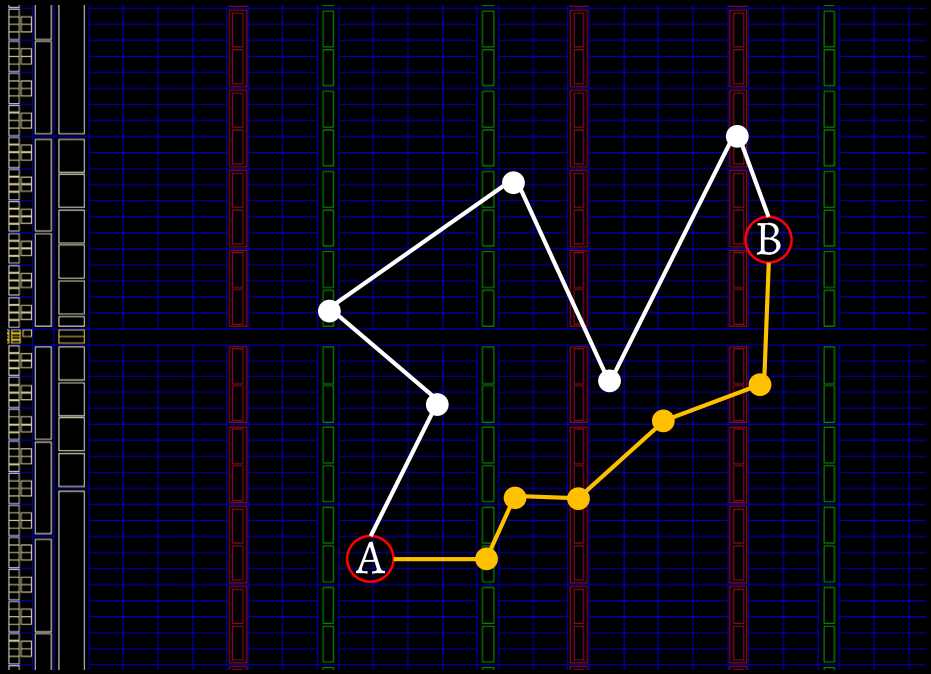
Do use memory waypoints wisely!



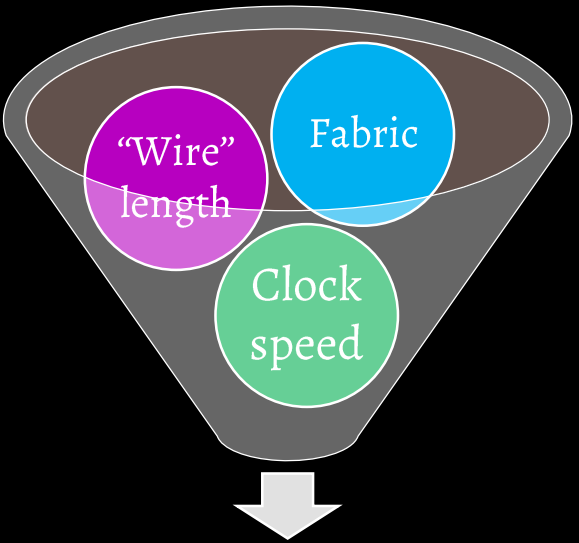
Timing Constraints

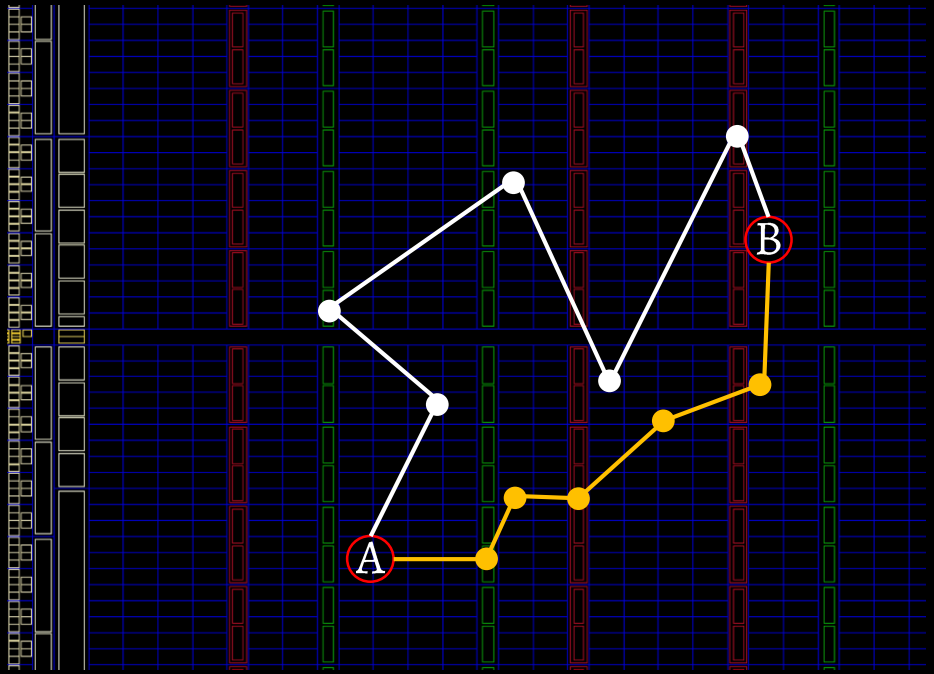


Registers / Flip Flops

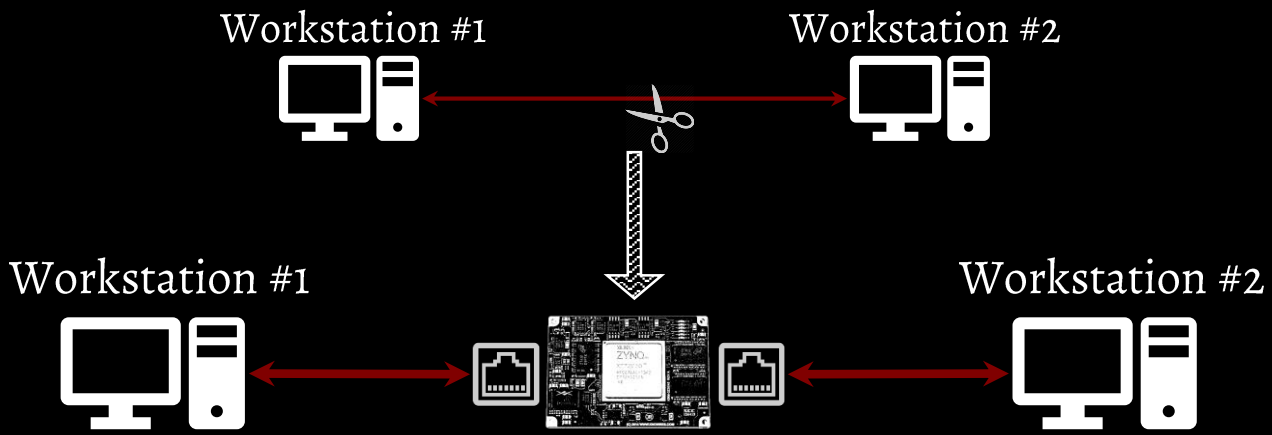


Timing Constraints – Dependencies

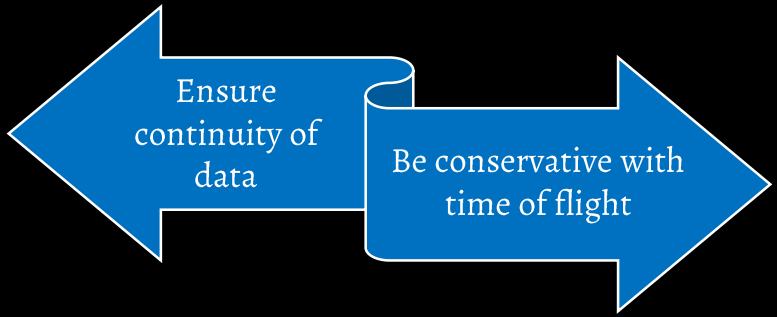




Primary goal: the Ethernet extension cord



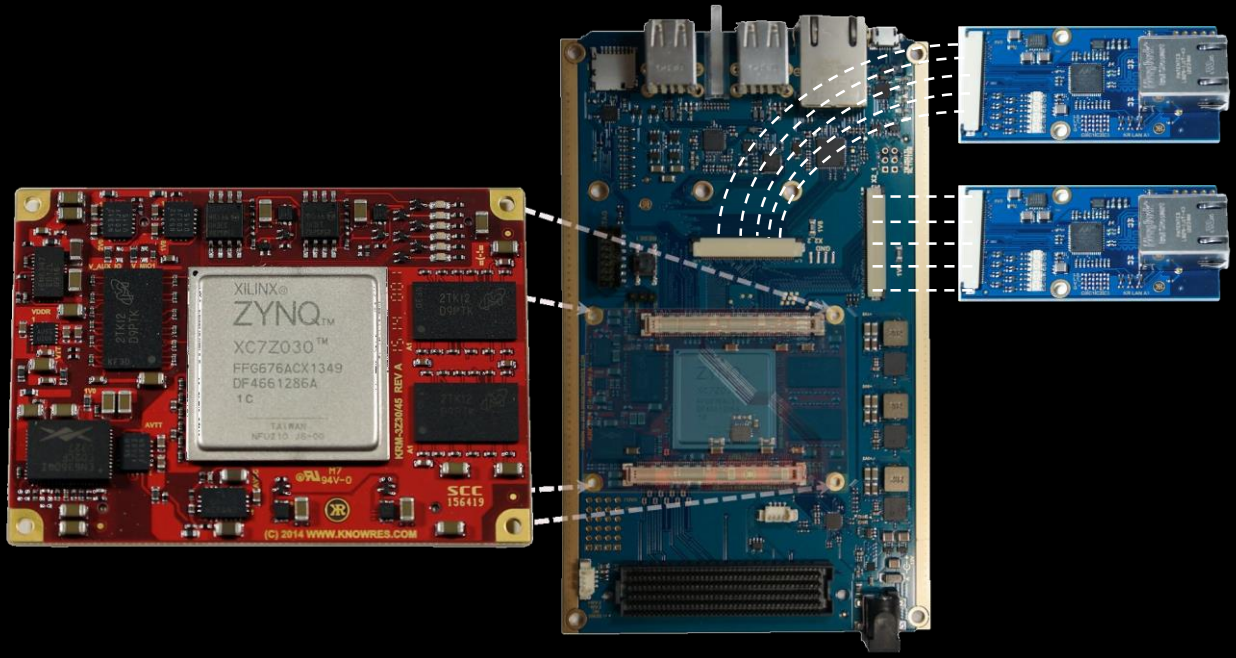
Primary goal: the Ethernet extension cord



Workstation #1

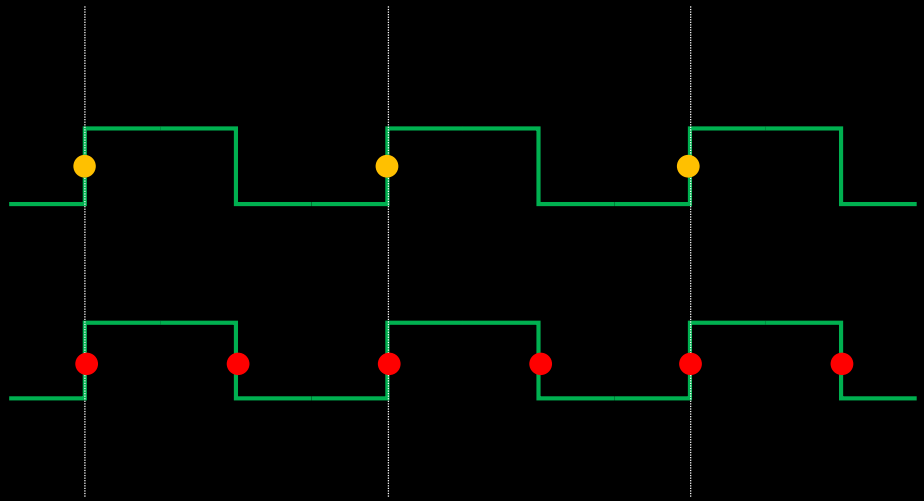
Workstation #2

FPGA expansion modules

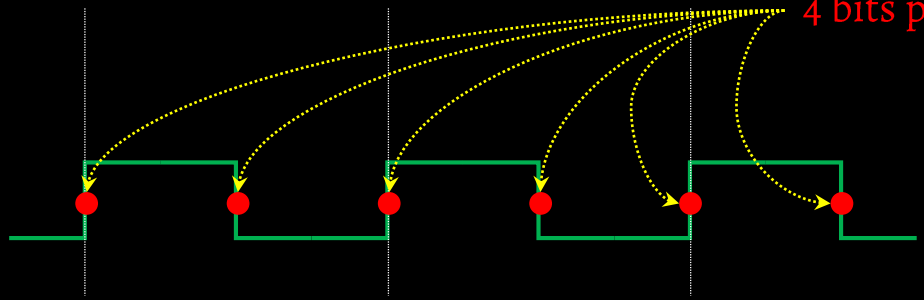


Picture taken from knowres.com

Ethernet at “ground” level



Ethernet at “ground” level



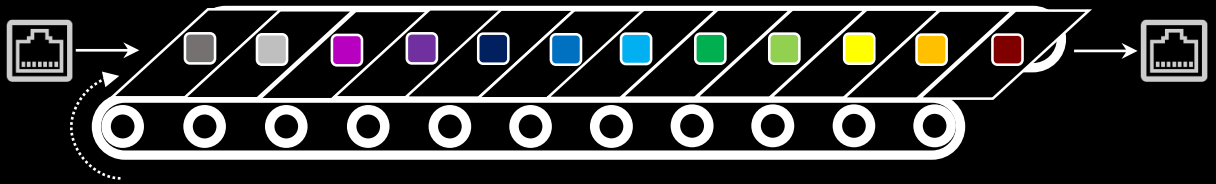
Double Data Rate (DDR)

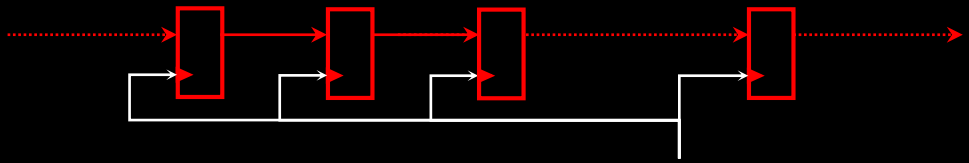
RGMII to GMII

SDR, 8 bit (1 byte) per cycle

1 cycle = 8ns (125 MHz) for 1Gbps link

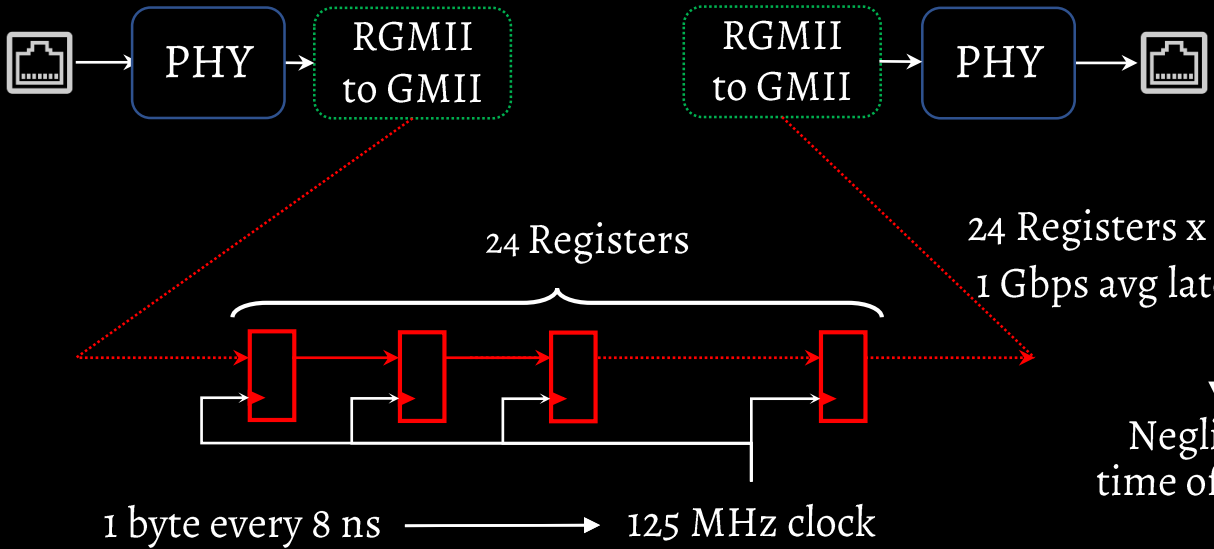
Conveyor belt model





1 byte every 8 ns → 125 MHz clock

Conveyor belt model



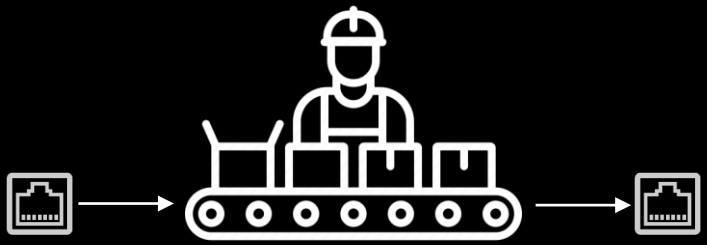
So far so good! What's next?

- Firewall-like features
- Deep packet inspection
- Encryption/ Decryption system

Network Packets

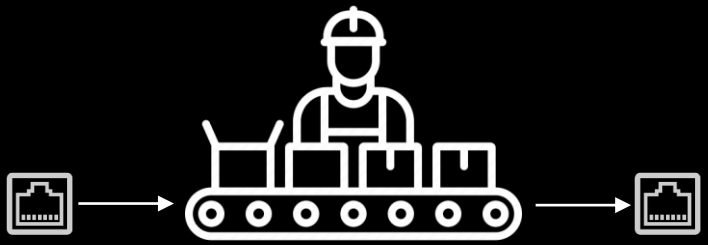
Information travels via network packets according to IEEE 802.3 standard

Assembly chain model



On-the-flight manipulation of packets

Ethernet Frame structure

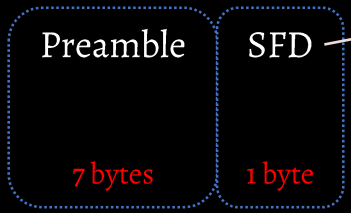


IEEE 802.3 standard

Preamble	SFD	Destination MAC address	Source MAC address	Ether Type	Payload	FCS
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes
Beginning constant		Receiver	Sender	Type of content	Content	Checksum

Preamble detector

- Constant pattern “0101010101....1101” → 0x55.....0xD5
- Helps clock synchronization



Data valid from GMII interface

Initialization of CRC engine

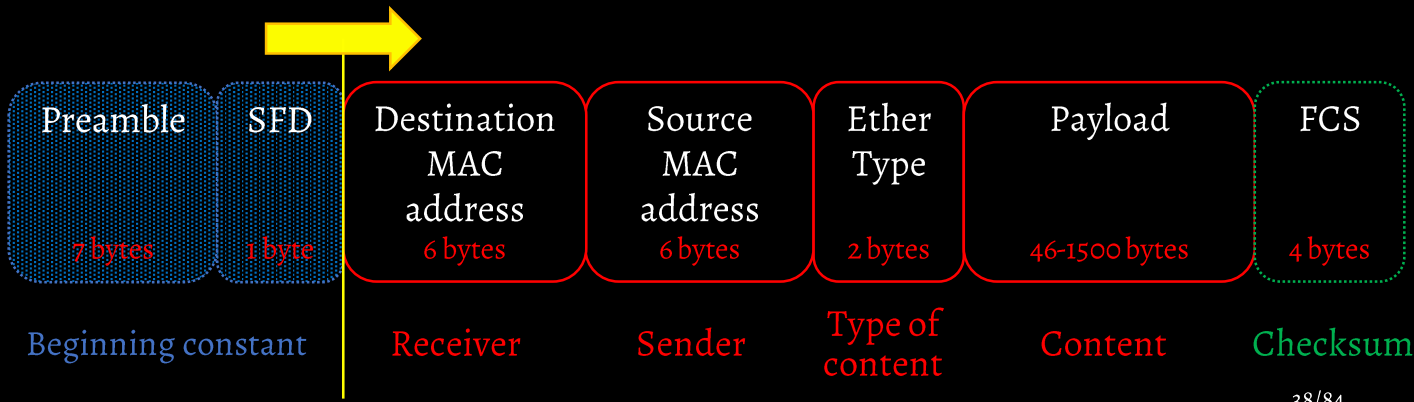
Initialization of crypto engine

```
when IDLE =>
   IS_CURRENT_STATE_IDLE <= '1';
   if (DV_IN = '1') then
         counter_next <= "001";
      state_next <= PREAMBLE_DETECT;
      crc_rst <= '1';
   else
      state_next <= IDLE;
   end if;

when PREAMBLE_DETECT =>
   if (PIPELINE(SIZE_OF_PIPE-1) = "01010101") then
      if (counter_reg = "011") then
         if (D_IN = "11010101") then
         state_next <= PREAMBLE_OK;
         crypto_start <= '1';
         else
         counter_next <= (OTHERS => '0');
         state_next <= IDLE;
         end if;
      else
         state_next <= PREAMBLE_DETECT;
      end if;
   else
      counter_next <= (OTHERS => '0');
      state_next <= IDLE;
   end if;
```

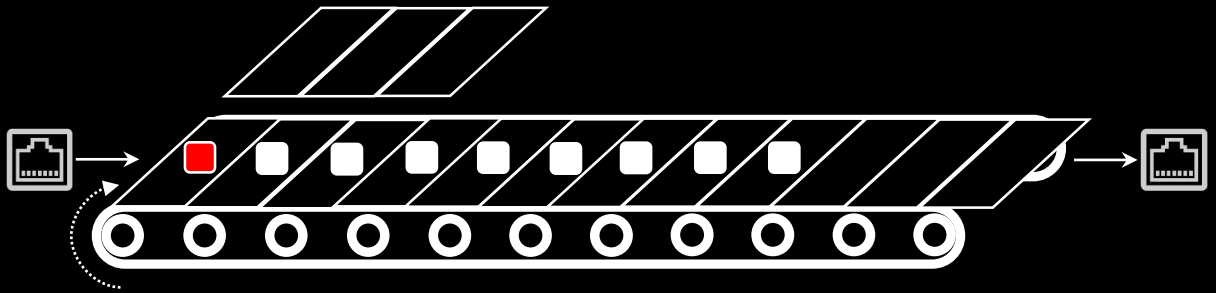
MAC and EtherType filters

Ensure continuity of data

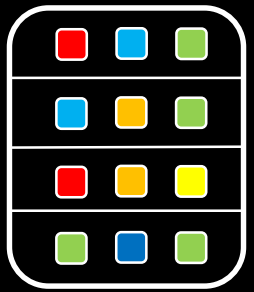
The assembly chain never stops



MAC and EtherType filters

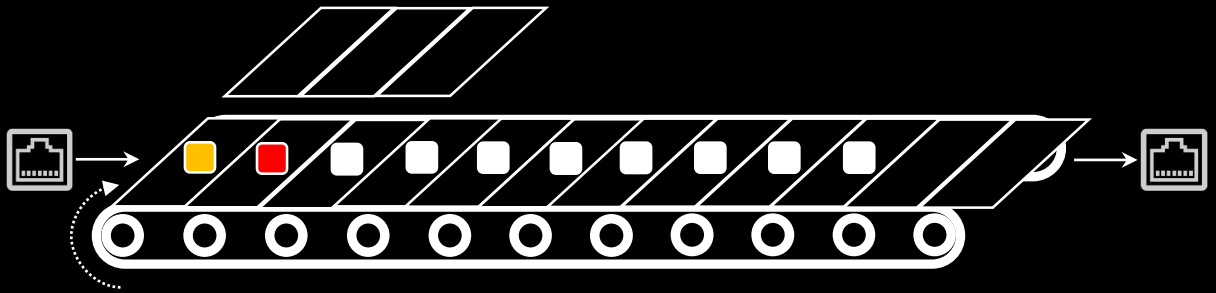


Blacklist

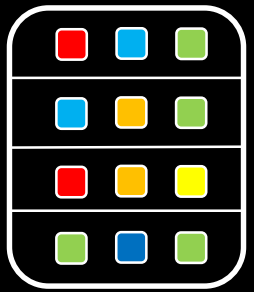


Search algorithm progress:
0%

MAC and EtherType filters

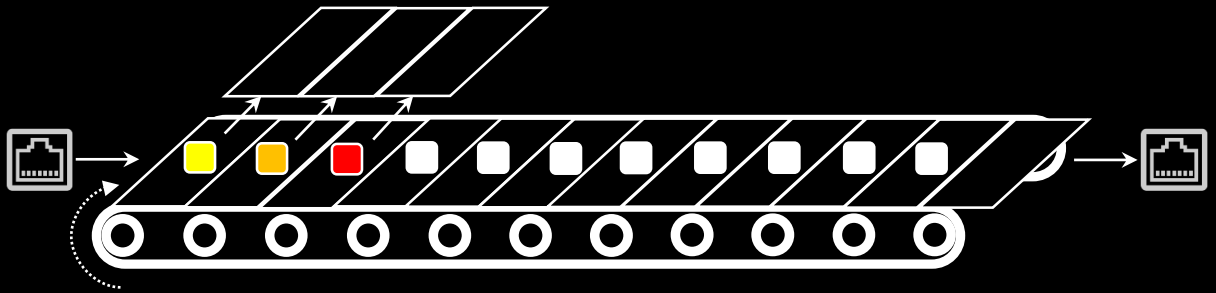


Blacklist

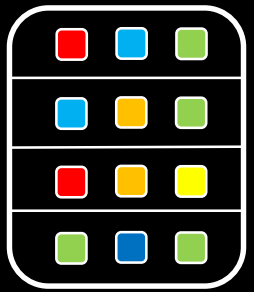


Search algorithm progress:
0%

MAC and EtherType filters

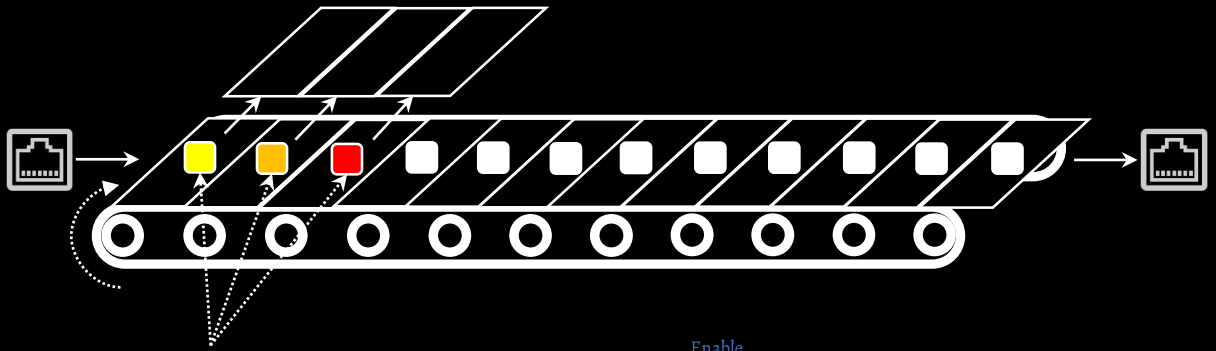


Blacklist

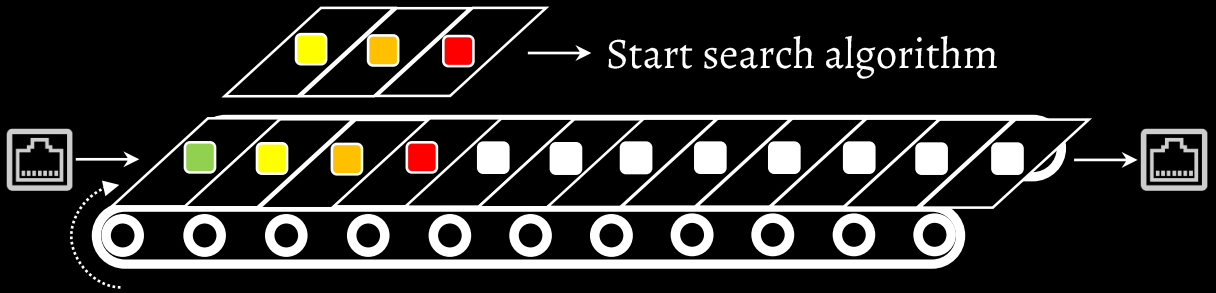


Search algorithm progress:
0%

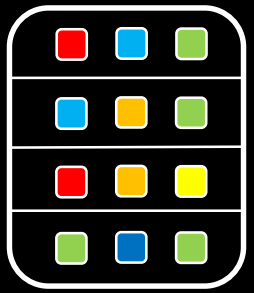
MAC and EtherType filters



MAC and EtherType filters

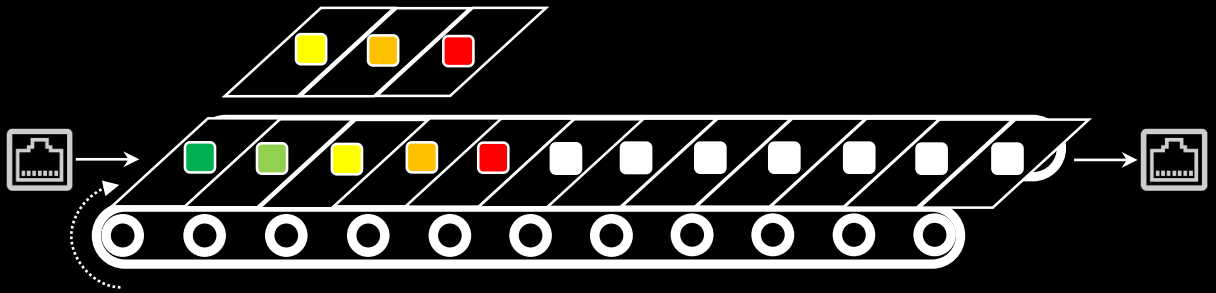


Blacklist

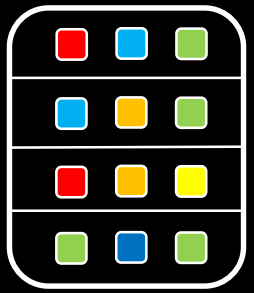


Search algorithm progress:
20%

MAC and EtherType filters

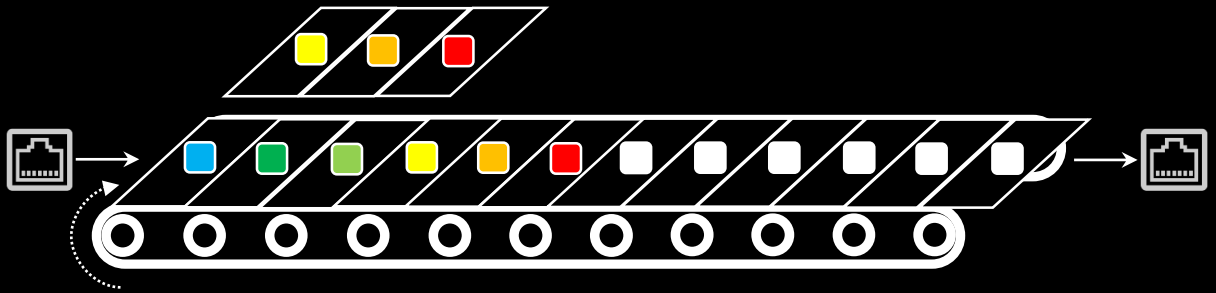


Blacklist

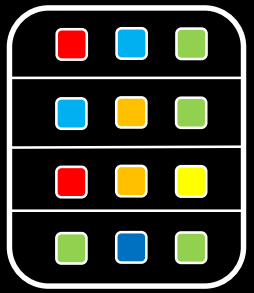


Search algorithm progress:
40%

MAC and EtherType filters

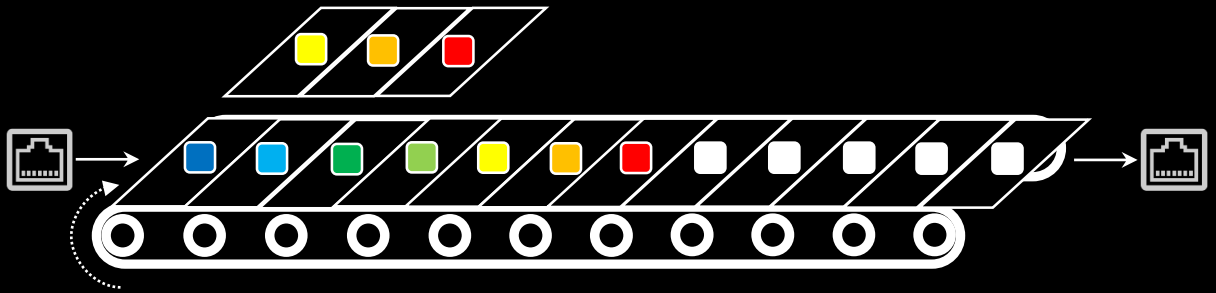


Blacklist

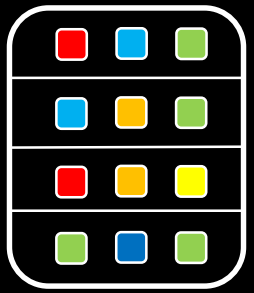


Search algorithm progress:
60%

MAC and EtherType filters

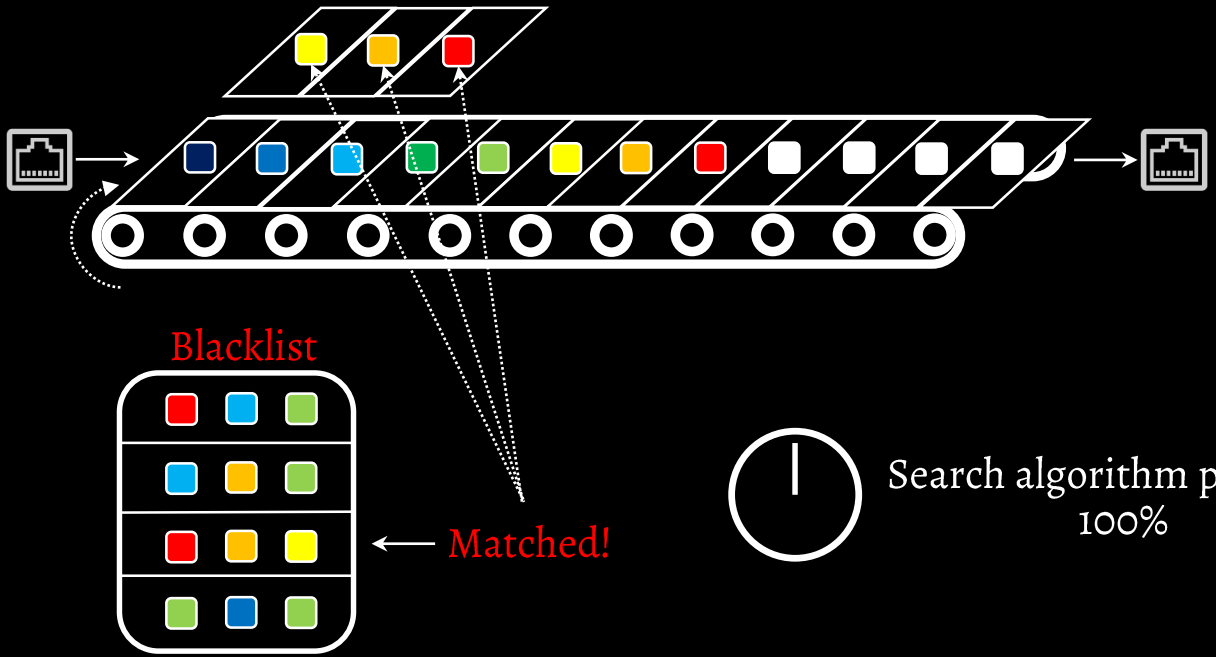


Blacklist

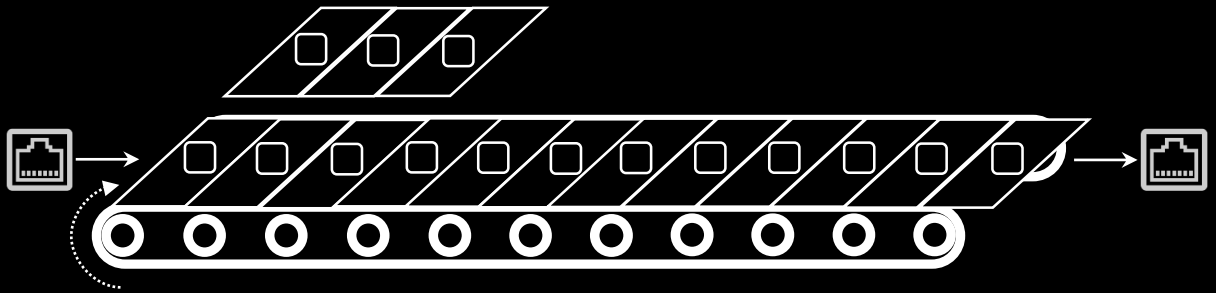


Search algorithm progress:
80%

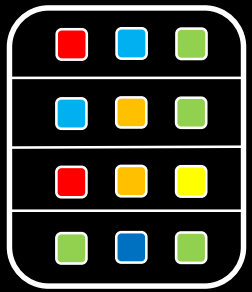
MAC and EtherType filters



MAC and EtherType filters



Blacklist



Matched!

Drop the whole frame

MAC and EtherType filters – Preparation

Smart Lookup in memory

- Arrange the Blacklist properly
- Dicotomic (Tricotomic) search

Ascending order pre-sort

- Handled by the CPU

MAC Address Library

1. 00:01:03:02:30:22
2. 02:23:43:C2:B4:FF
3. C0:CA:C0:1A:00:00
4. FF:FF:FF:FF:FF:AC

Qsort is useful!

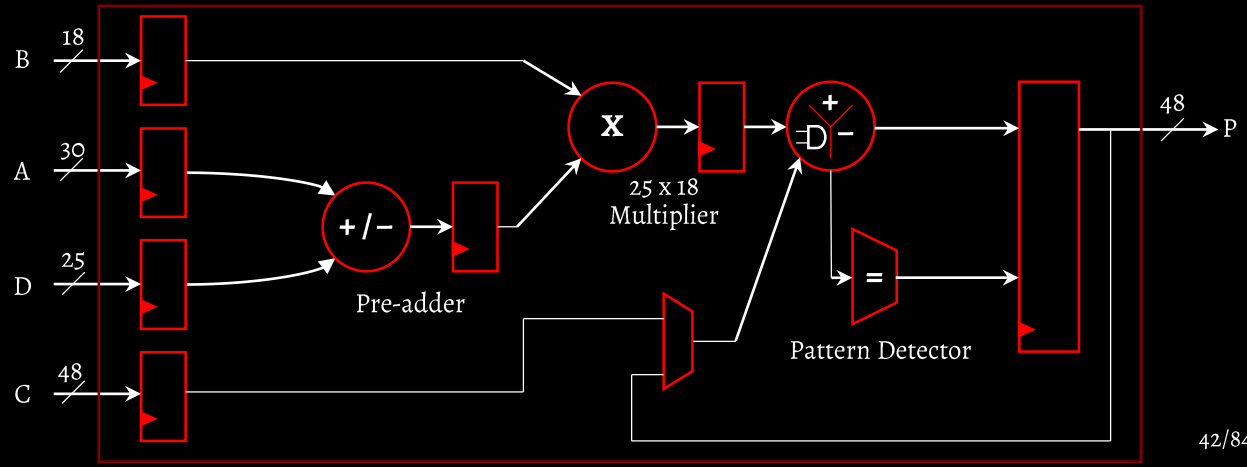
MAC and EtherType filters

Fetch and Match

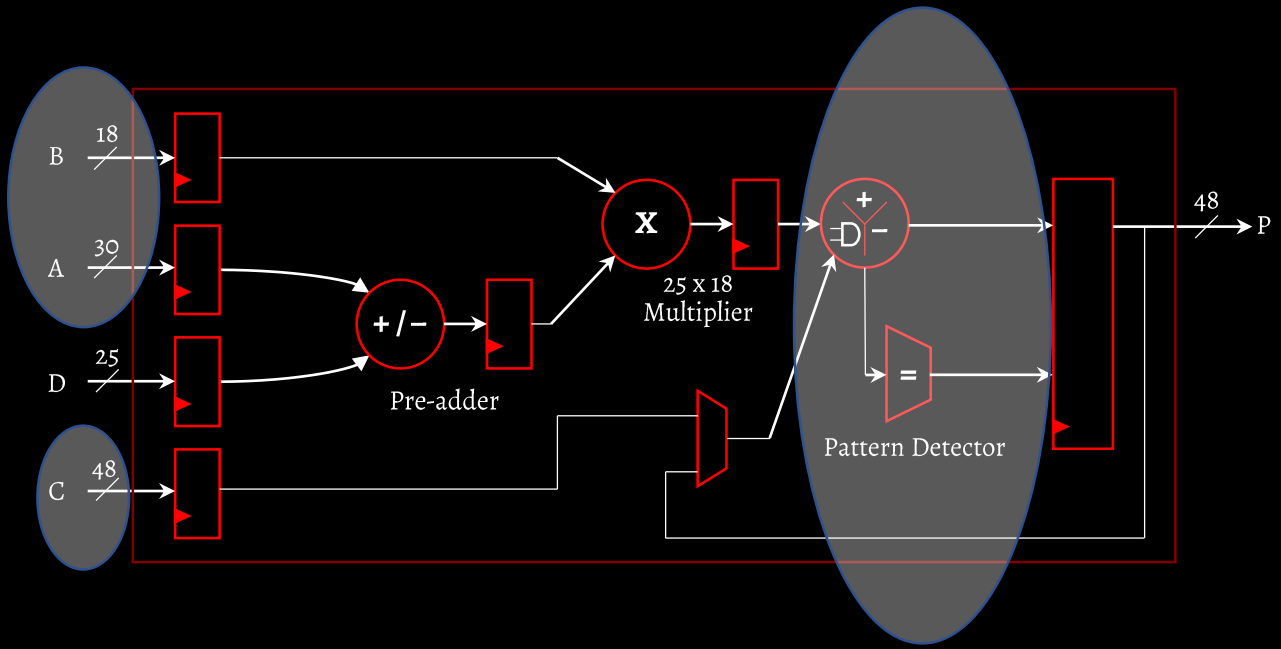
- The incoming MAC is compared with the blacklist entries

DSP makes fast comparisons

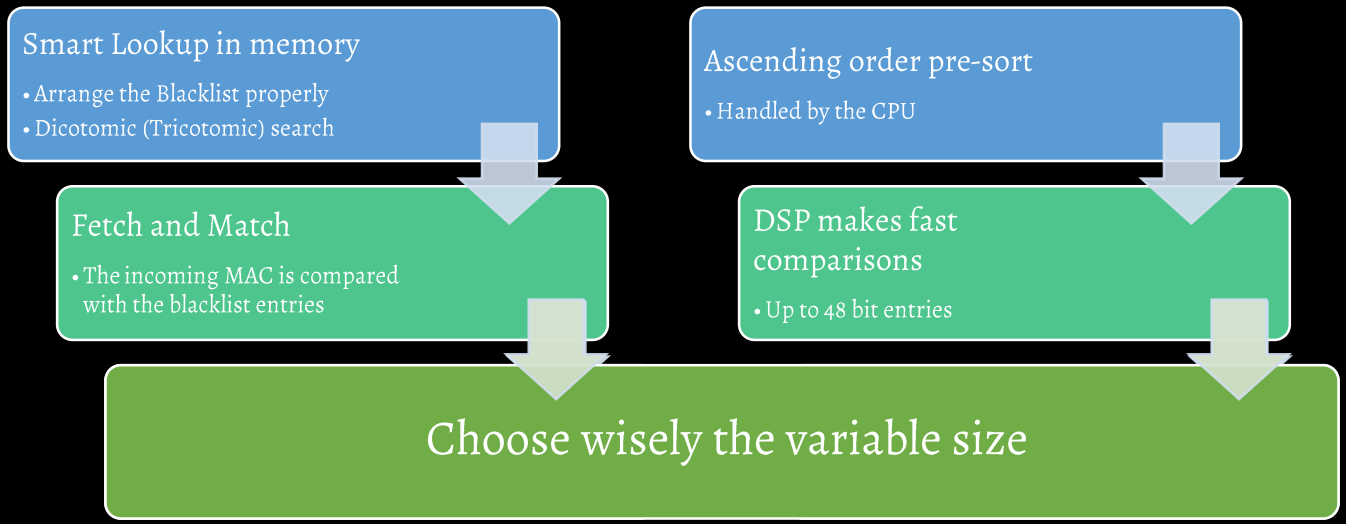
- Up to 48 bit entries



DSP48


B
18
A
30
D
25
C
48
+/−
Pre-adder
X
25 x 18
Multiplier
+
−
=
Pattern Detector
48
P

MAC and EtherType filters



MAC address

00:10:FA:CC:C0:1A

MAC address – 48 bits

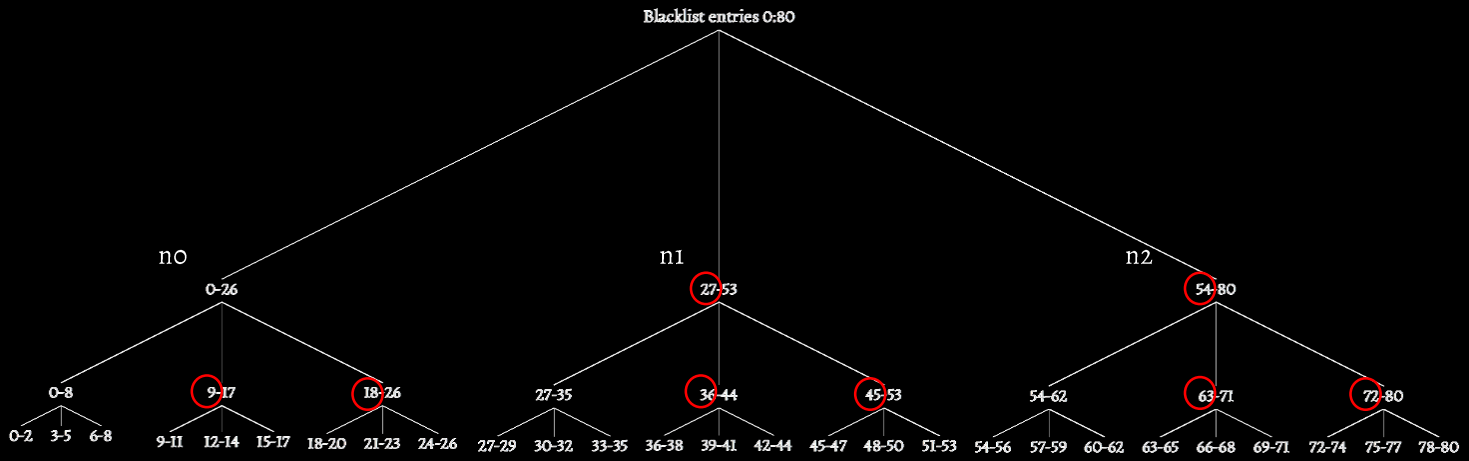
Organizationally Unique Identifier (OUI) – 24 bits

Do save time!

Compare 24-bit operands

DSP will evaluate 2 comparisons per time

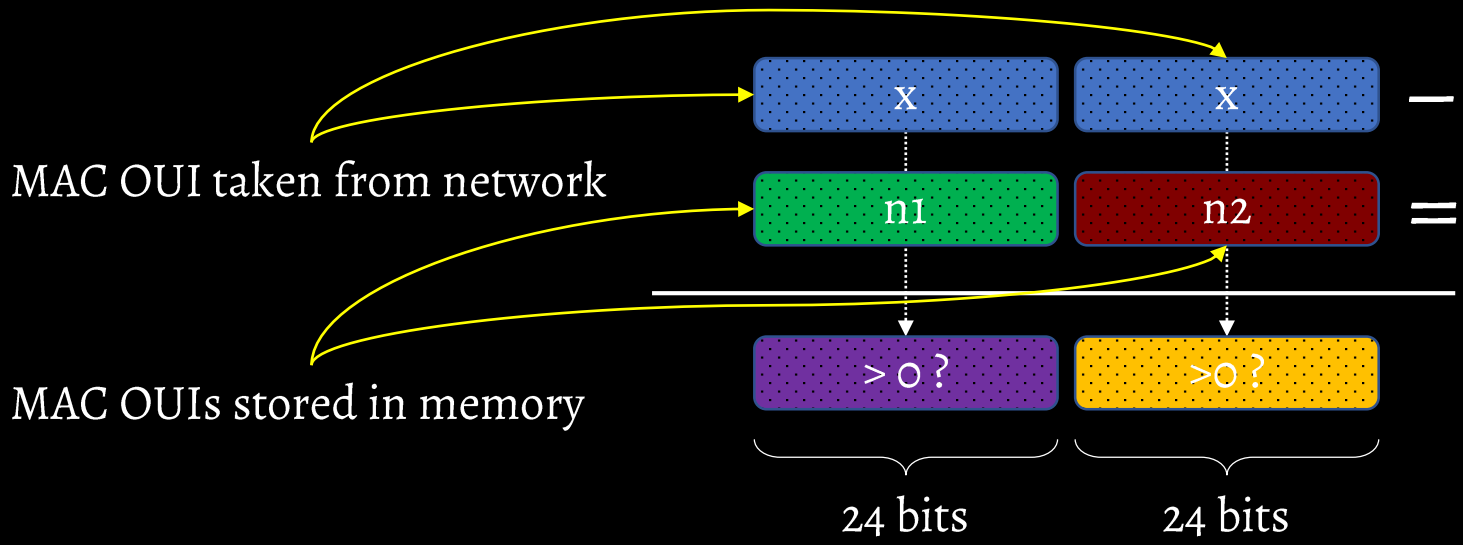
Search Algorithm version 1 – tricotomic



$x > n1\ ? \rightarrow x - n1 > 0$

$x > n2\ ? \rightarrow x - n2 > 0$

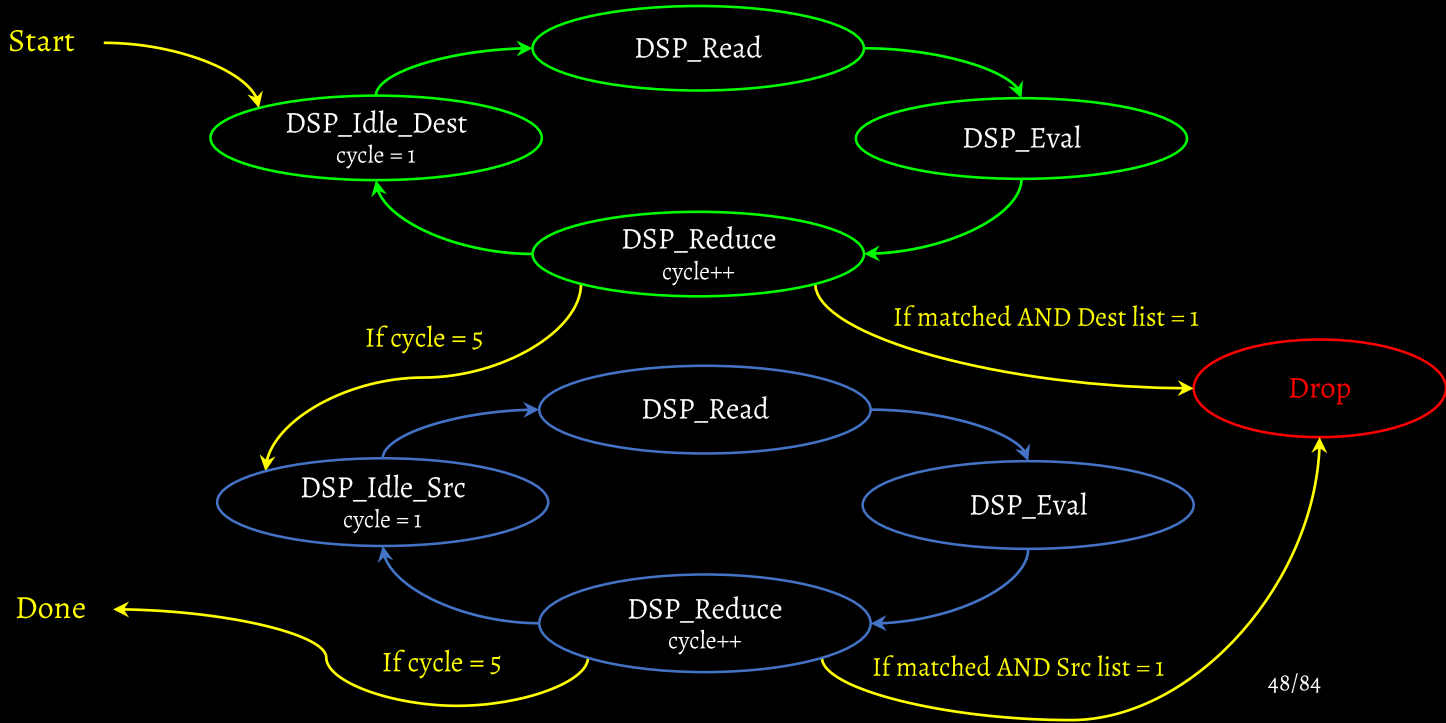
Search Algorithm version 1 – tricotomic



$x > n1 ? \rightarrow x - n1 > 0$

$x > n2 ? \rightarrow x - n2 > 0$

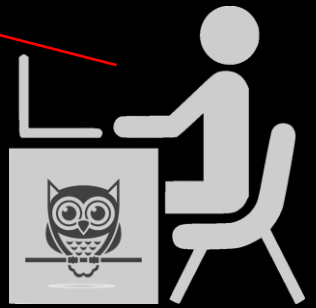
MAC filter state machine



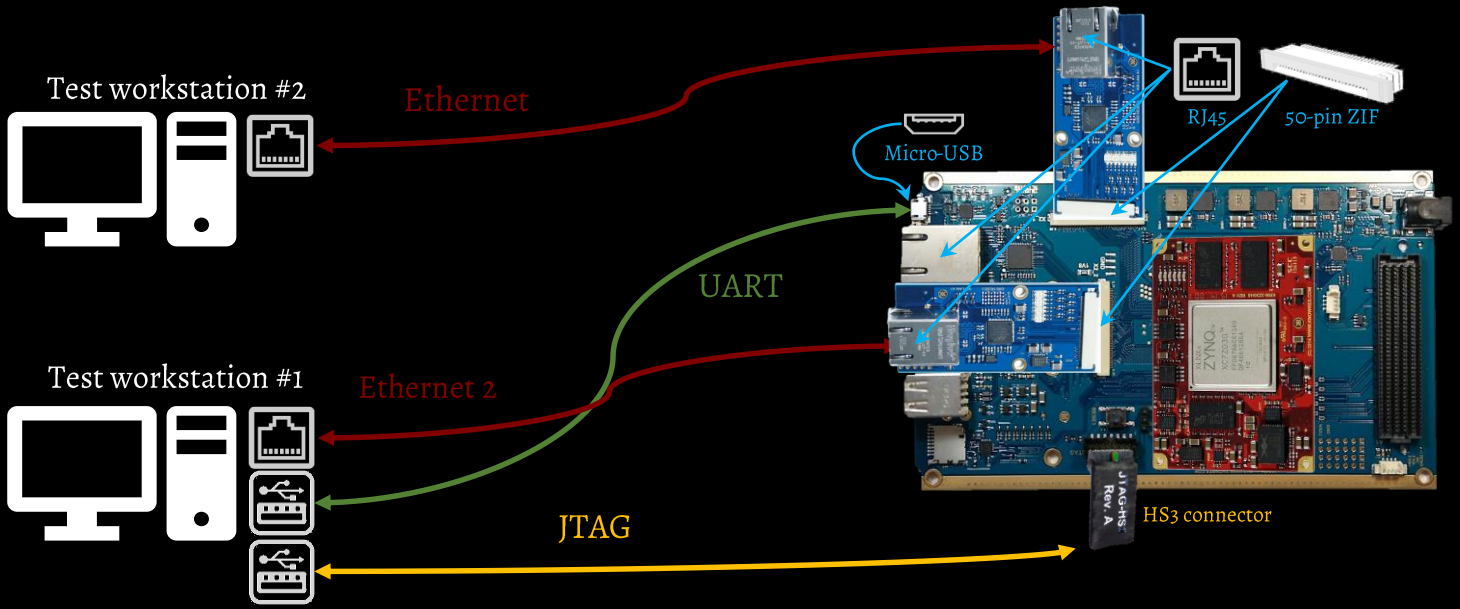
MAC and EtherType filters configuration

```
Set up MAC filter for Source and Destination address:

Source :        Packets coming from the chosen MAC are flushed
Destination :   Packets directed towards the chosen MAC are flushed
Would you like to add all the inserted MACs to:
(1) Source blacklist
(2) Destination blacklist
(3) Both Source and Destination blacklist
(4) I would like to choose one by one

Please type per each MAC:
(1) --> Source blacklist
(2) --> Destination blacklist
(3) --> Both Source and Destination blacklist

  0 | 00:0A:35:XX:XX:XX DESTINATION
  1 | 0A:0A:0A:XX:XX:XX SOURCE
  2 | 12:23:45:XX:XX:XX SOURCE & DESTINATION
  3 | 12:31:23:XX:XX:XX
```

- Up to 81 entries → 3^4
- Source and/or Destination MAC attribute is defined per each entry

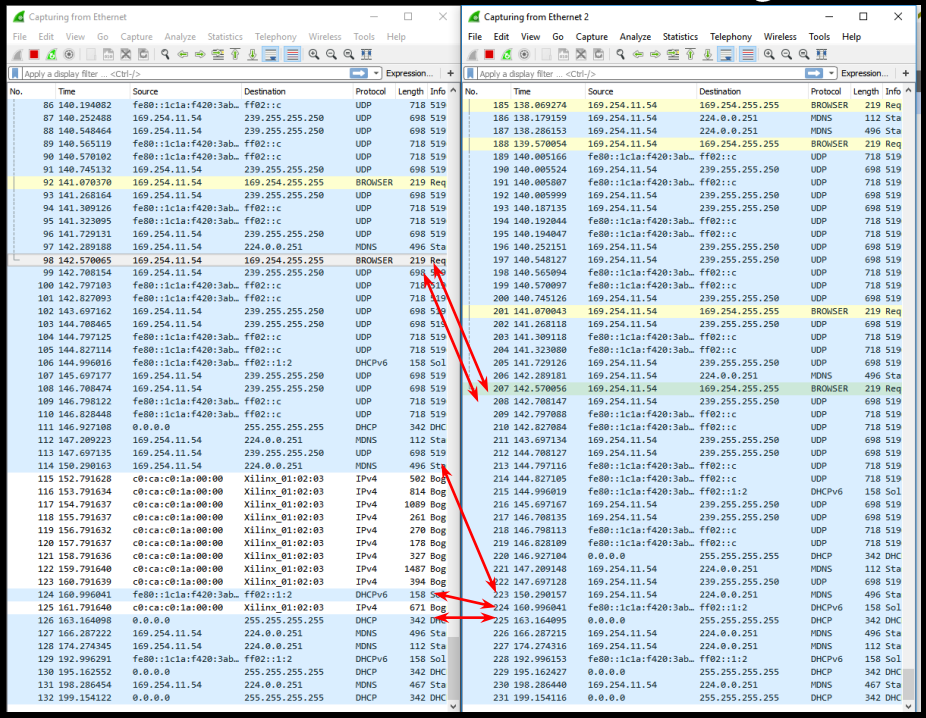


Demo : MAC filtering

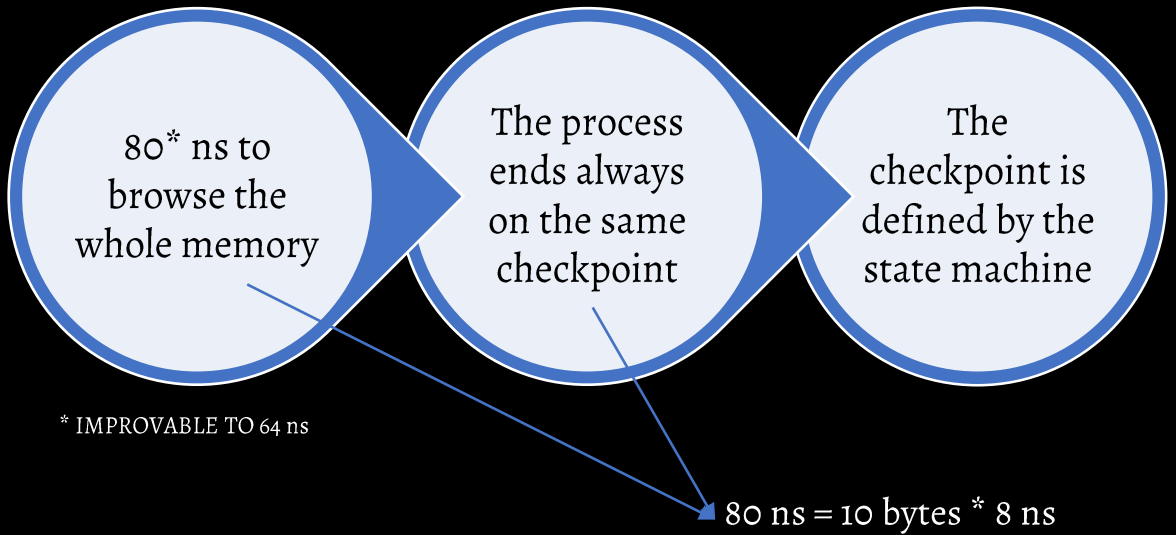


Demo : MAC filtering

Demo : MAC filtering



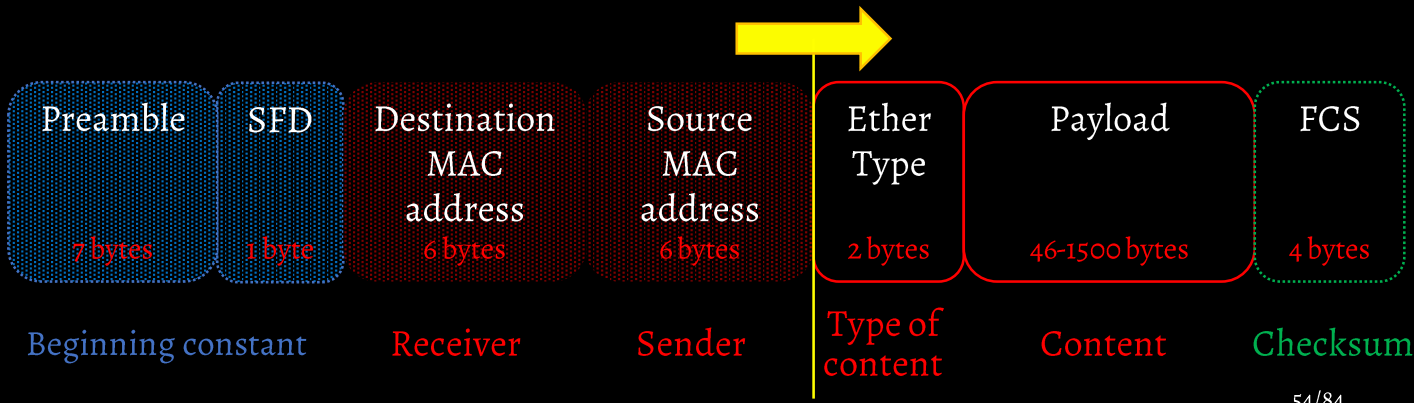
MAC filter conclusions



EtherType filter

Ensure continuity of data

The assembly chain never stops

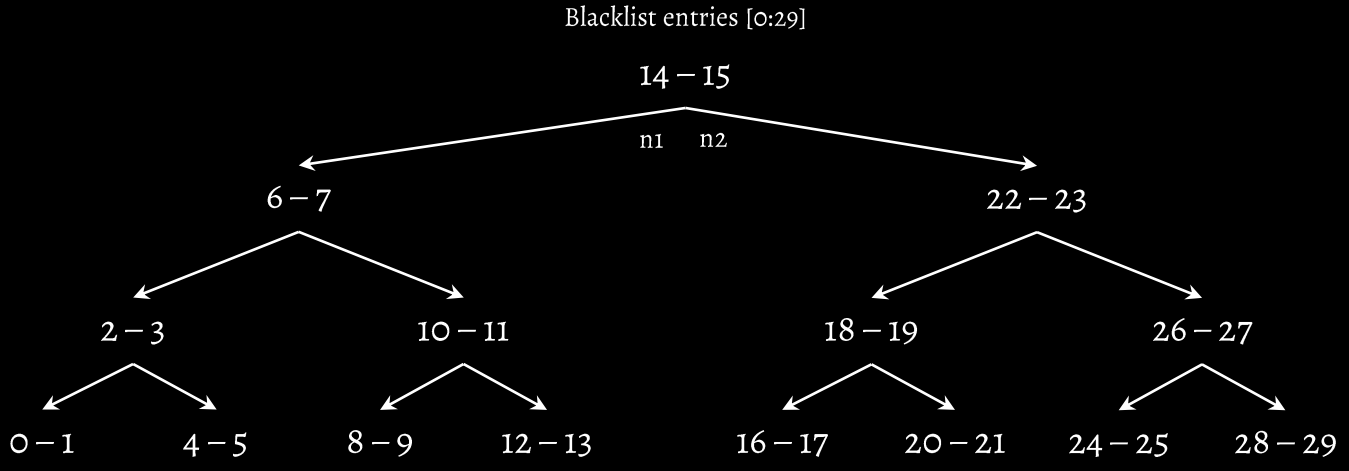


EtherType filter configuration

```
CURRENT LIST:

  # |    EtherType_ID
---------------------
  0 |    0x0800

Do you want to add a new EtherType ID to your filter?
You still have 29 entries left

(y)es, please           (n)o, that's enough for today


Ok, fine.


Set up EtherType filter to work as a Whitelist or Blacklist

Whitelist :     Only the packets with matching EtherTypes are accepted
Blacklist :     Packets with matching EtherTypes are flushed
Would you like to set up a Blacklist or a Whitelist?
(1) Blacklist
(2) Whitelist
```

- Up to 30 entries
- Blacklist/Whitelist mode

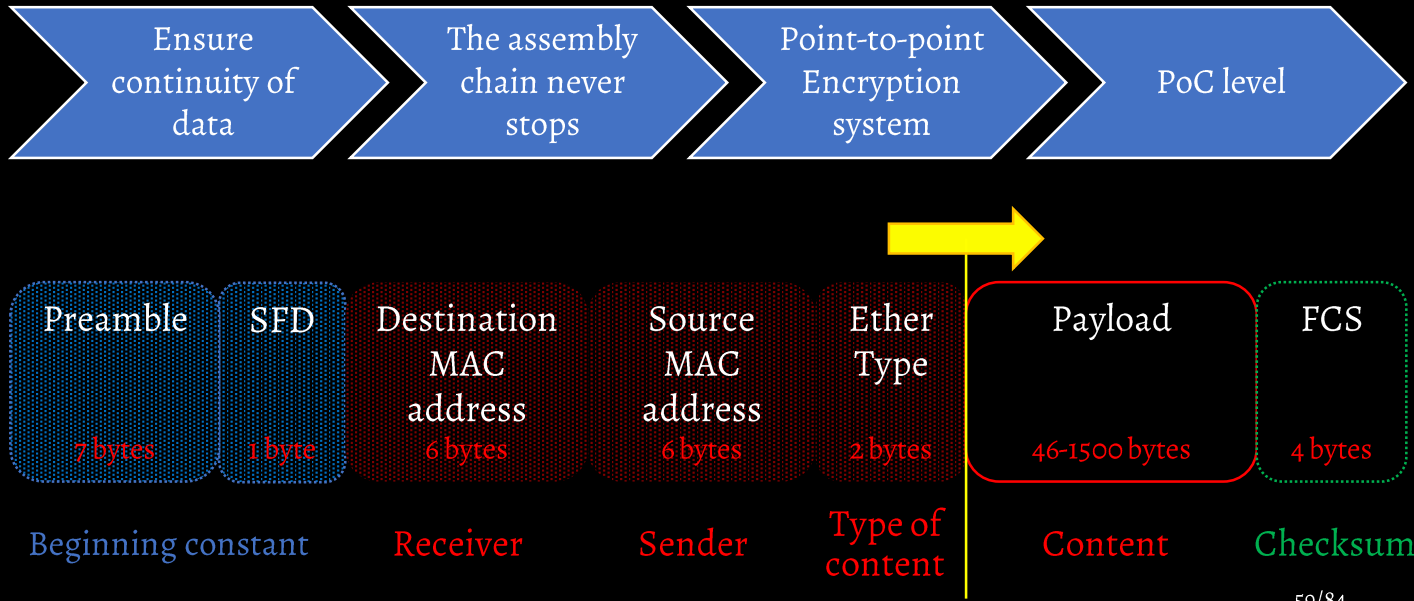
Search Algorithm version 2



x > n1 ? → x-n1 > 0

x > n2 ? → x-n2 > 0

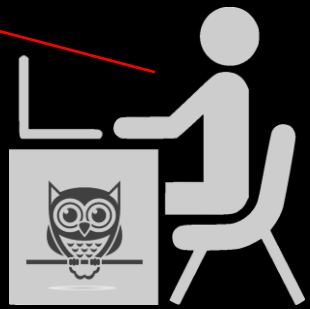
Encryption Environment



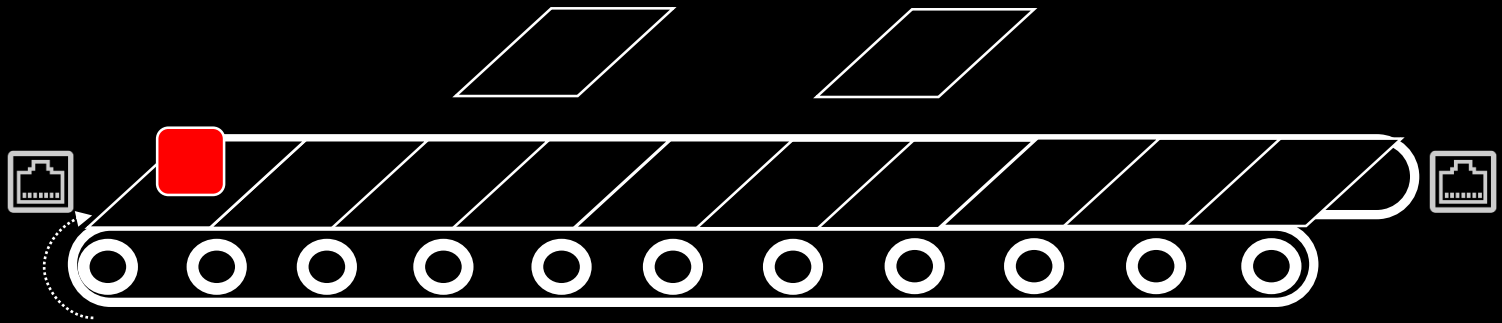
Encryption configuration

```
-----------------------------------
-----|  Crypto Engine Setup  |-----
-----------------------------------
Do you want to setup the Crypto Engine? Press (y)es or (n)o
Alright!


What is the port exposed the outer world?
Type 1 (parallel wrt board) or 2 (perpendicular)

You chose port #2

It's time to choose a password. Maximum characters allowed: 32
Type your password here below. The ENTER key will validate your password

*******
Please, type your password once again to confirm.
*******

Congratulations: it's time to generate the seed now
```

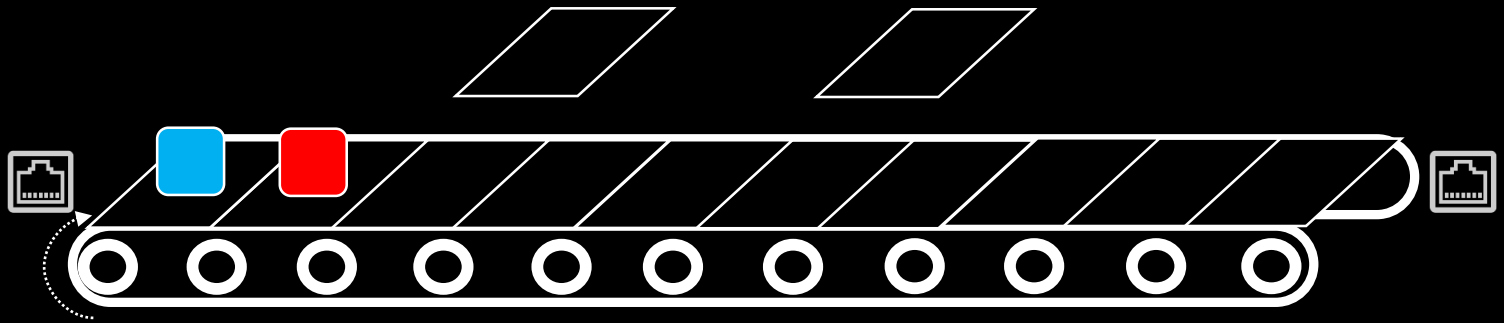
- Password input by the user
- A key is generated to encrypt data, one byte per time



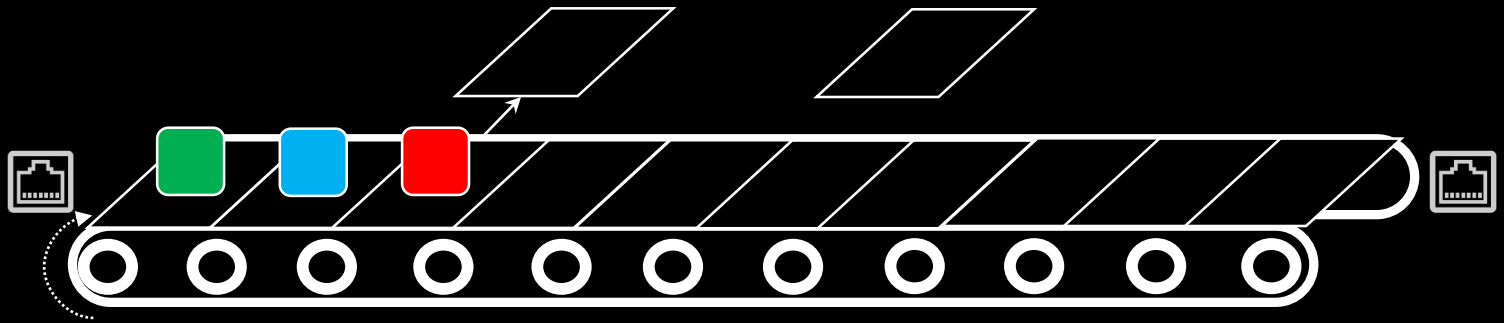
Encryption schema



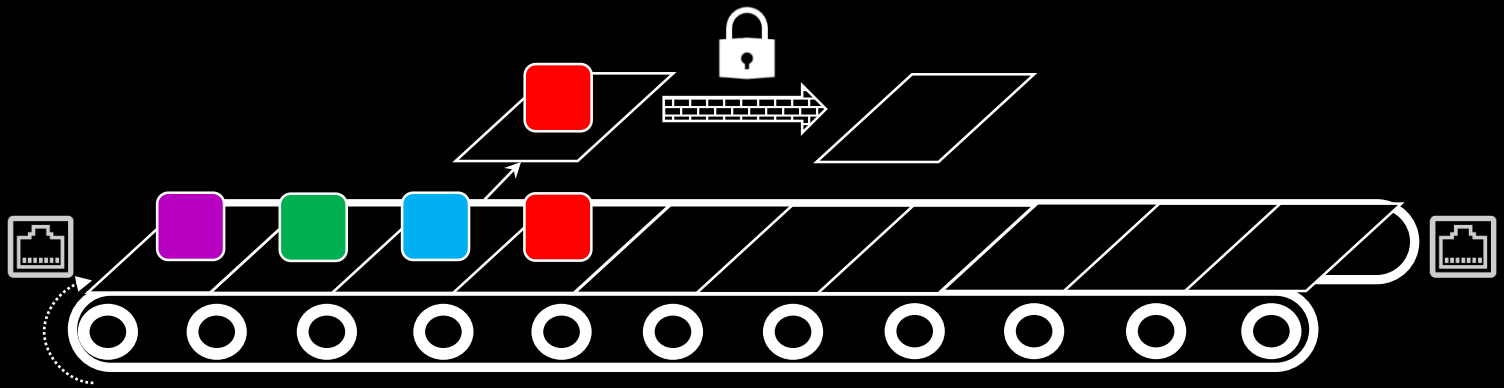
Encryption schema



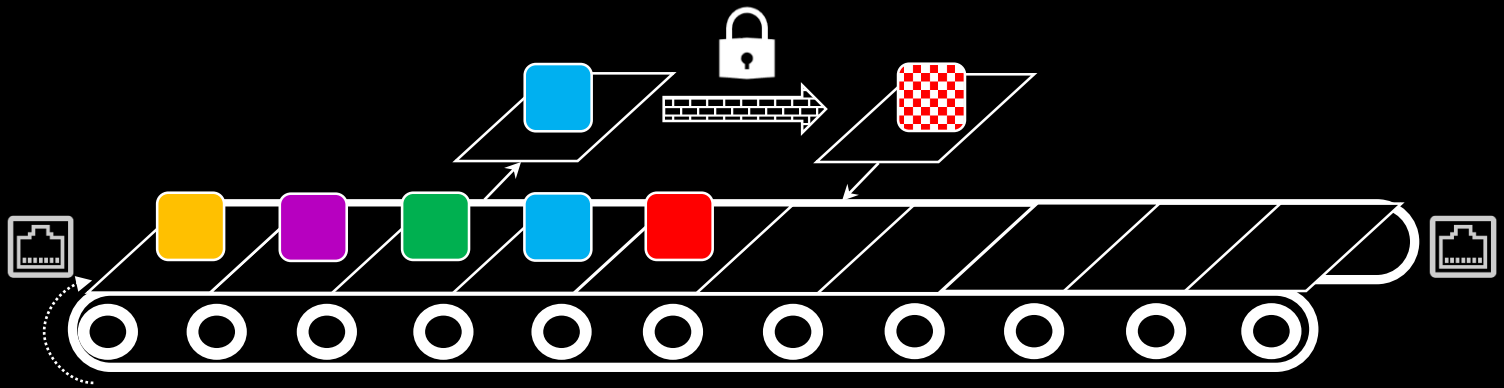
Encryption schema



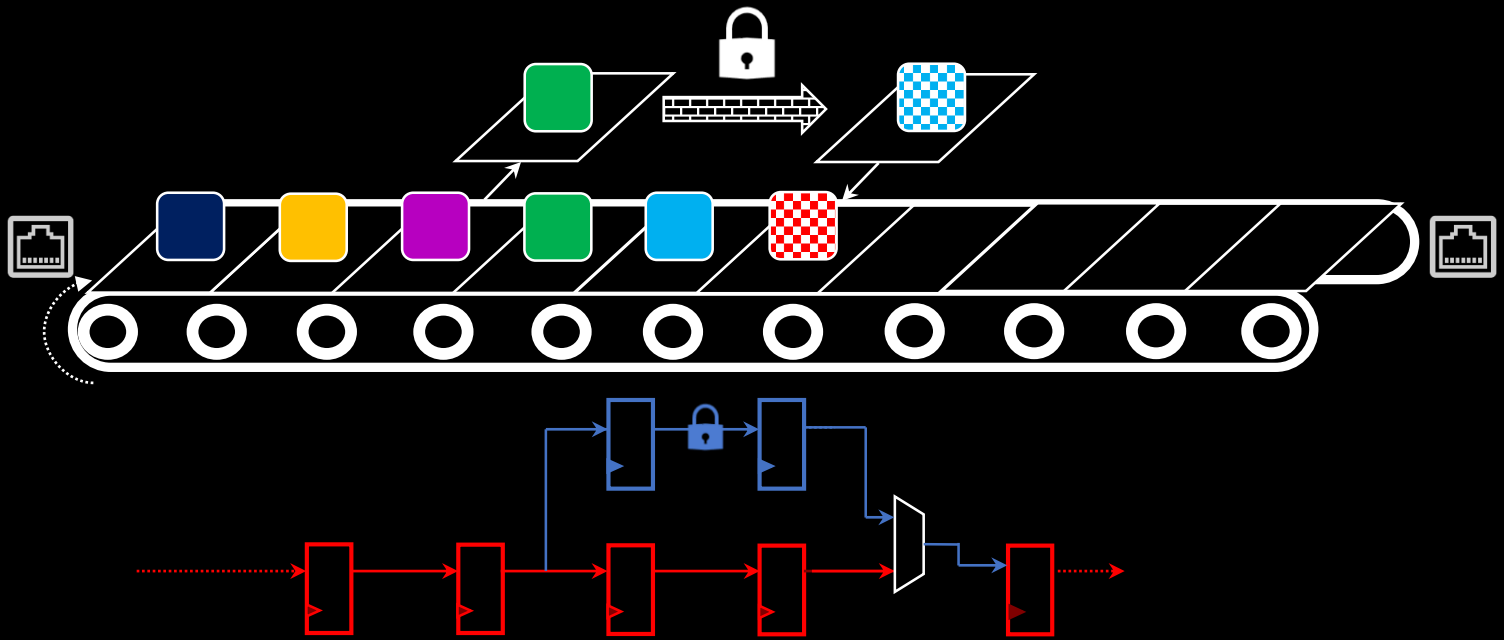
Encryption schema



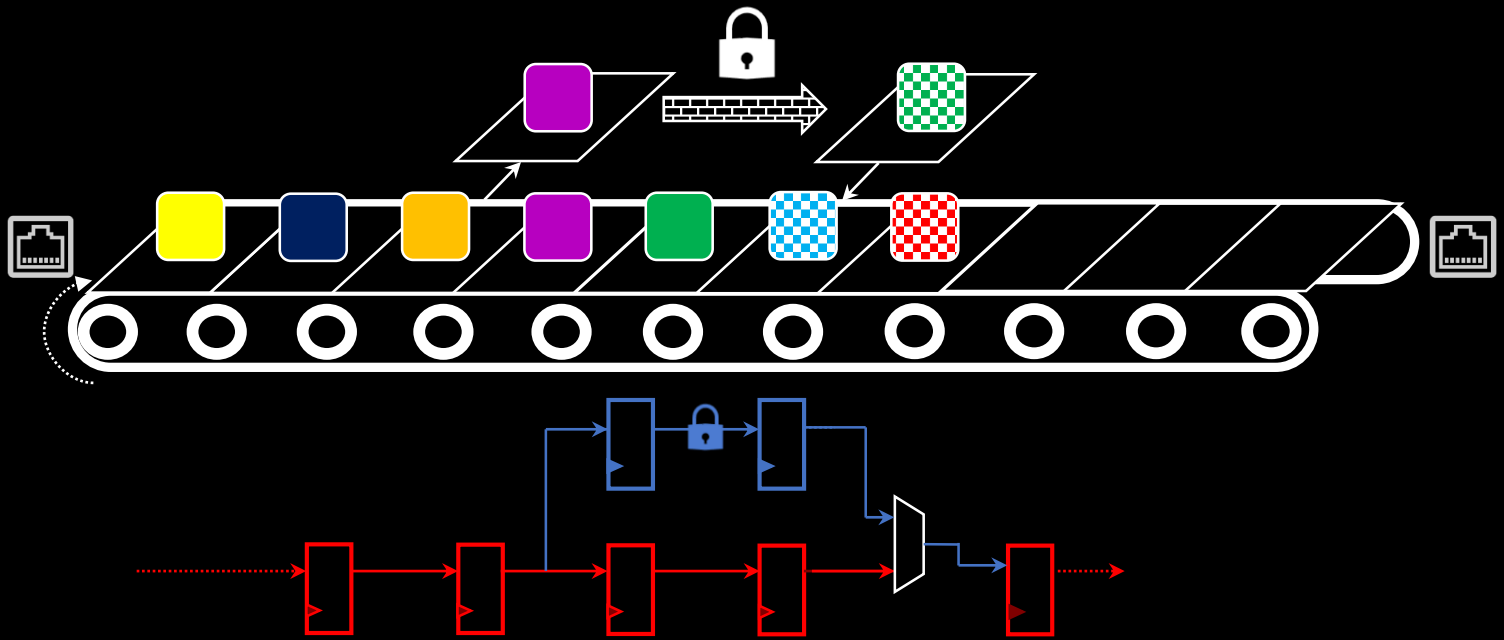
Encryption schema



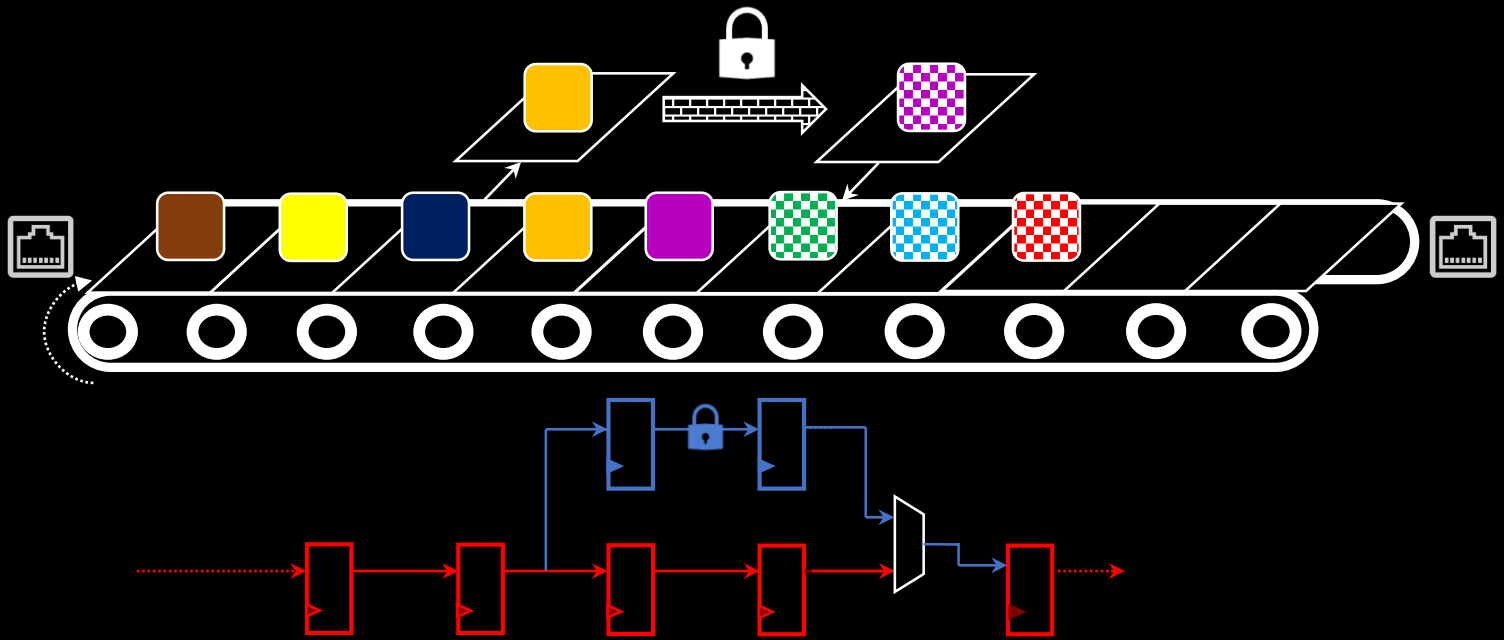
Encryption schema



Encryption schema



Encryption schema



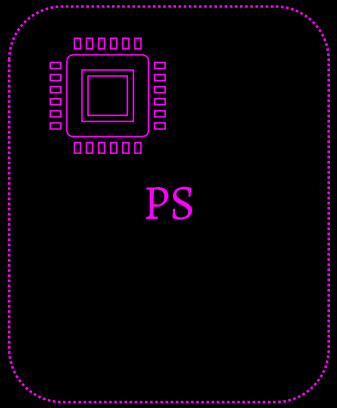
Encryption architecture

The following slides can seriously hurt Crypto specialists

Encryption architecture

Password input by user → 32-bit hash: Fowler Noll Vo → 32 8-bit subsets are cycled

Encryption architecture



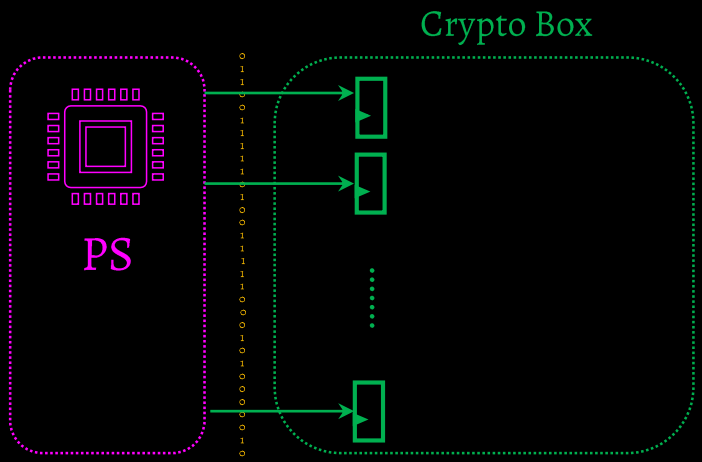
JohnDo3

Password input by user

0x67D3E282

32-bit hash: Fowler Noll Vo

Encryption architecture



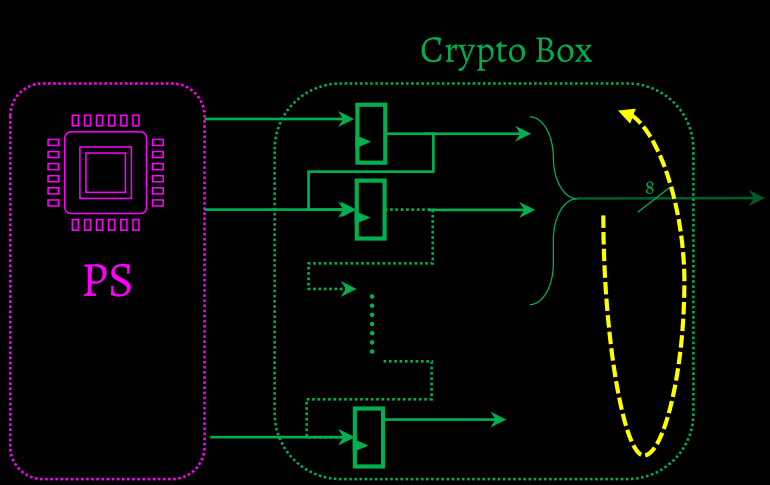
0x67D3E282

1. 0b01100111
2. 0b11001111
3. 0b10011111

8-bit subsets are cycled

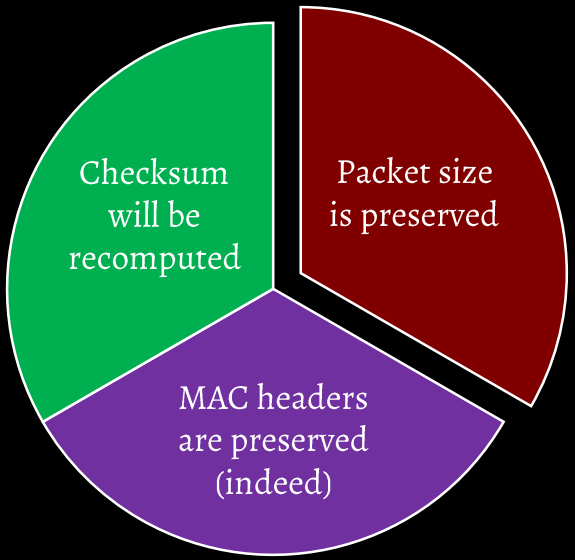
0b01100111111010011111100010100000010

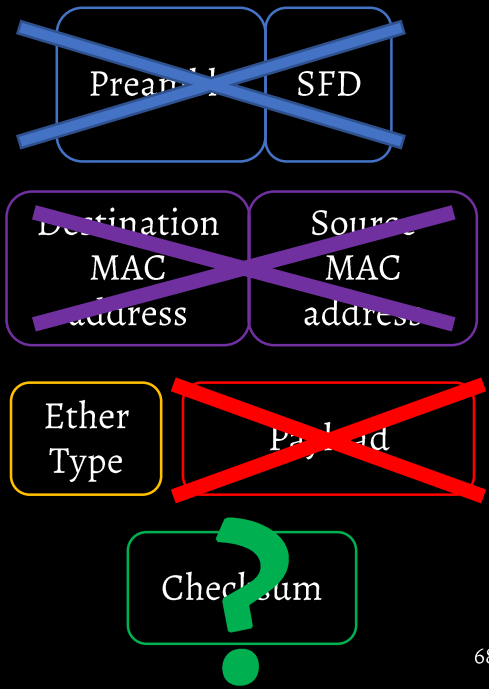
Encryption architecture



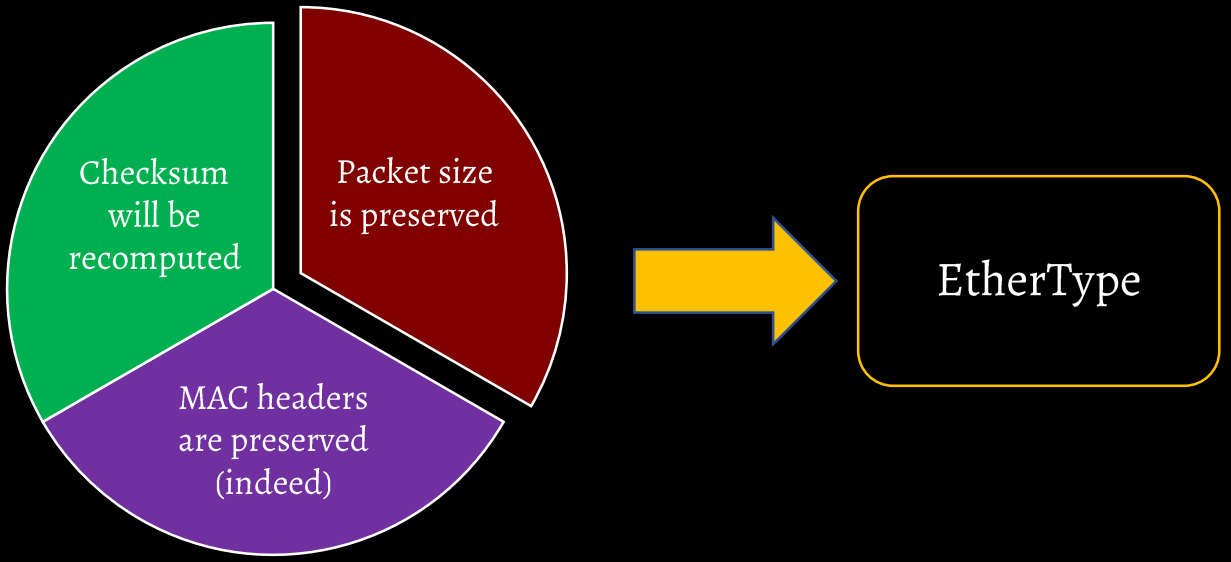
J	3	o
o	J	3
H	o	J
n	H	o
D	n	H
o	D	n
3	o	D

Encryption architecture – “Signature”





Encryption architecture – “Signature”



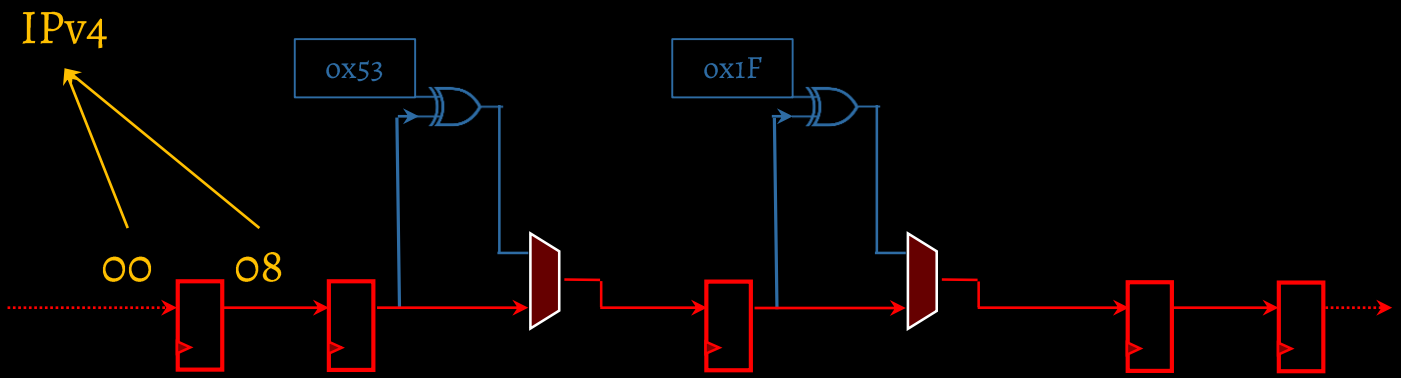
Encryption architecture – “Signature”

Match EtherTypes with another ID

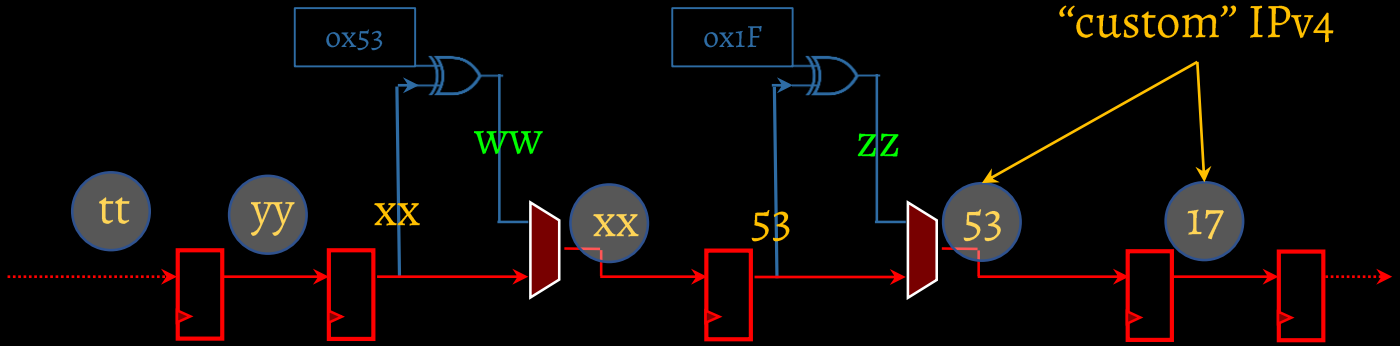
Only IPv4 is encoded atm

EtherType	Protocol
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x0842	Wake-on-LAN[9]
0x22F3	IETF TRILL Protocol
0x22EA	Stream Reservation Protocol
0x6003	DECnet Phase IV
0x8035	Reverse Address Resolution Protocol
0x809B	AppleTalk (Ethertalk)
0x80F3	AppleTalk Address Resolution Protocol (AARP)
0x8100	VLAN-tagged frame (IEEE 802.1Q) and Shortest Path Bridging IEEE 802.1aq with NNI compatibility[10]
0x8137	IPX
0x8204	QNX Qnet
0x86DD	Internet Protocol Version 6 (IPv6)
0x8808	Ethernet flow control
0x8809	Ethernet Slow Protocols[11] such as the Link Aggregation Control Protocol

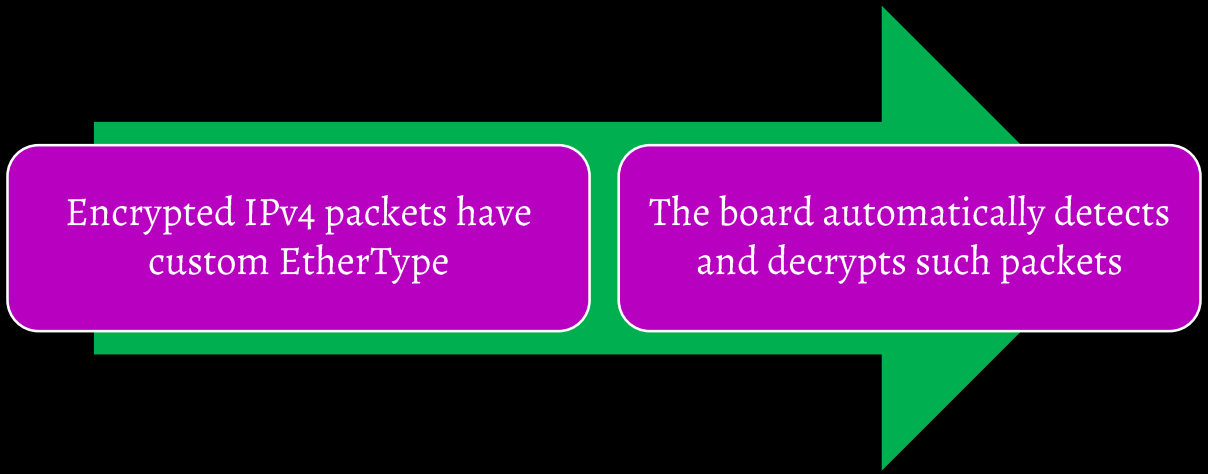
Encryption architecture – “Signature”



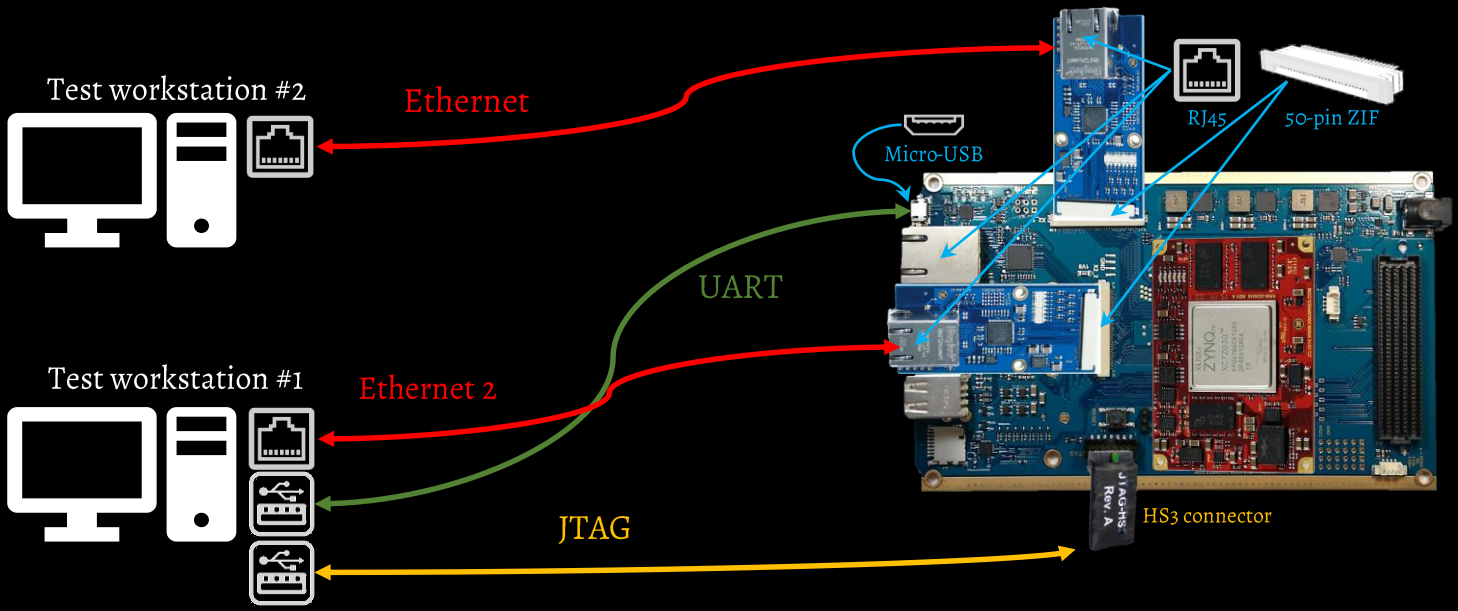
Encryption architecture – “Signature”



Encryption architecture – “Signature”

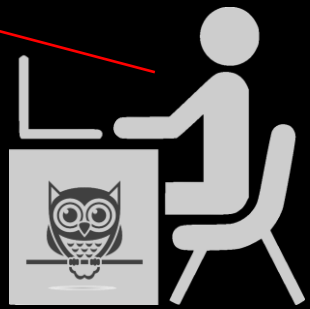


Demo : Encryption

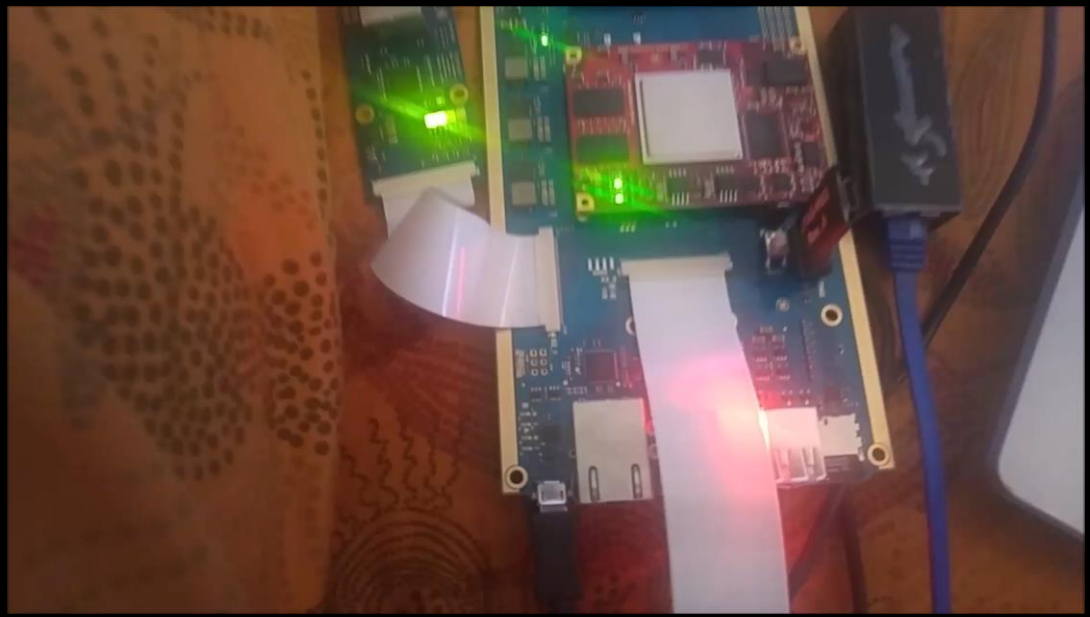


Demo : Encryption

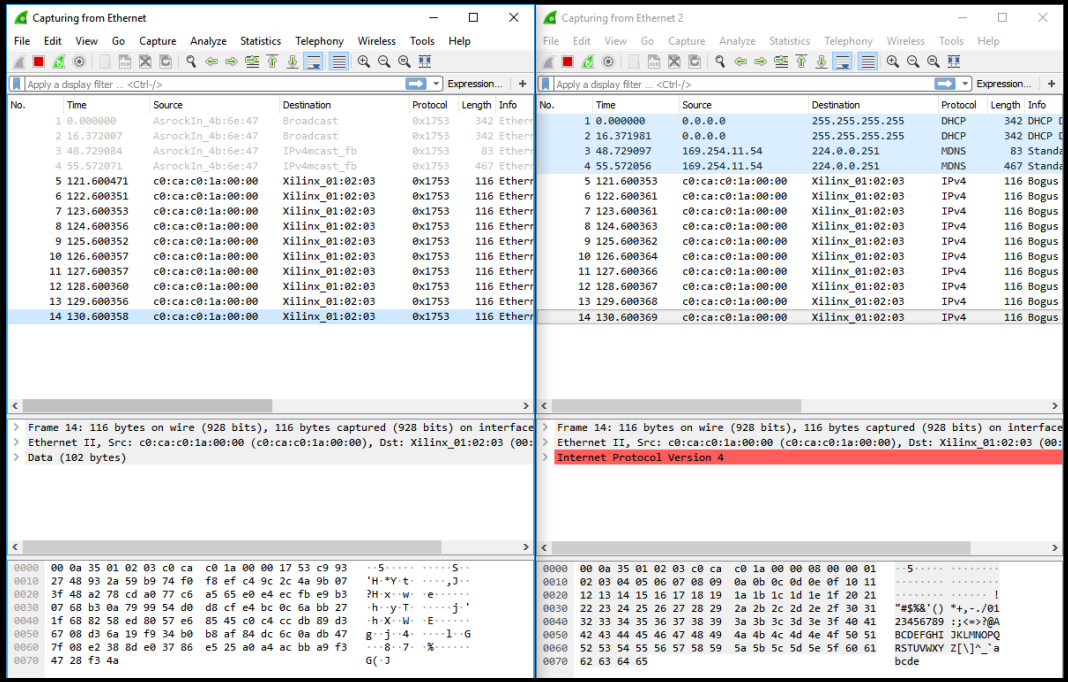
```
-----------------------------------
-----|  Crypto Engine Setup  |-----
-----------------------------------
Do you want to setup the Crypto Engine? Press (y)es or (n)o
Alright!


What is the port exposed the outer world?
Type 1 (parallel wrt board) or 2 (perpendicular)

You chose port #2

It's time to choose a password. Maximum characters allowed: 32
Type your password here below. The ENTER key will validate your password

*******
Please, type your password once again to confirm.
*******

Congratulations: it's time to generate the seed now
```



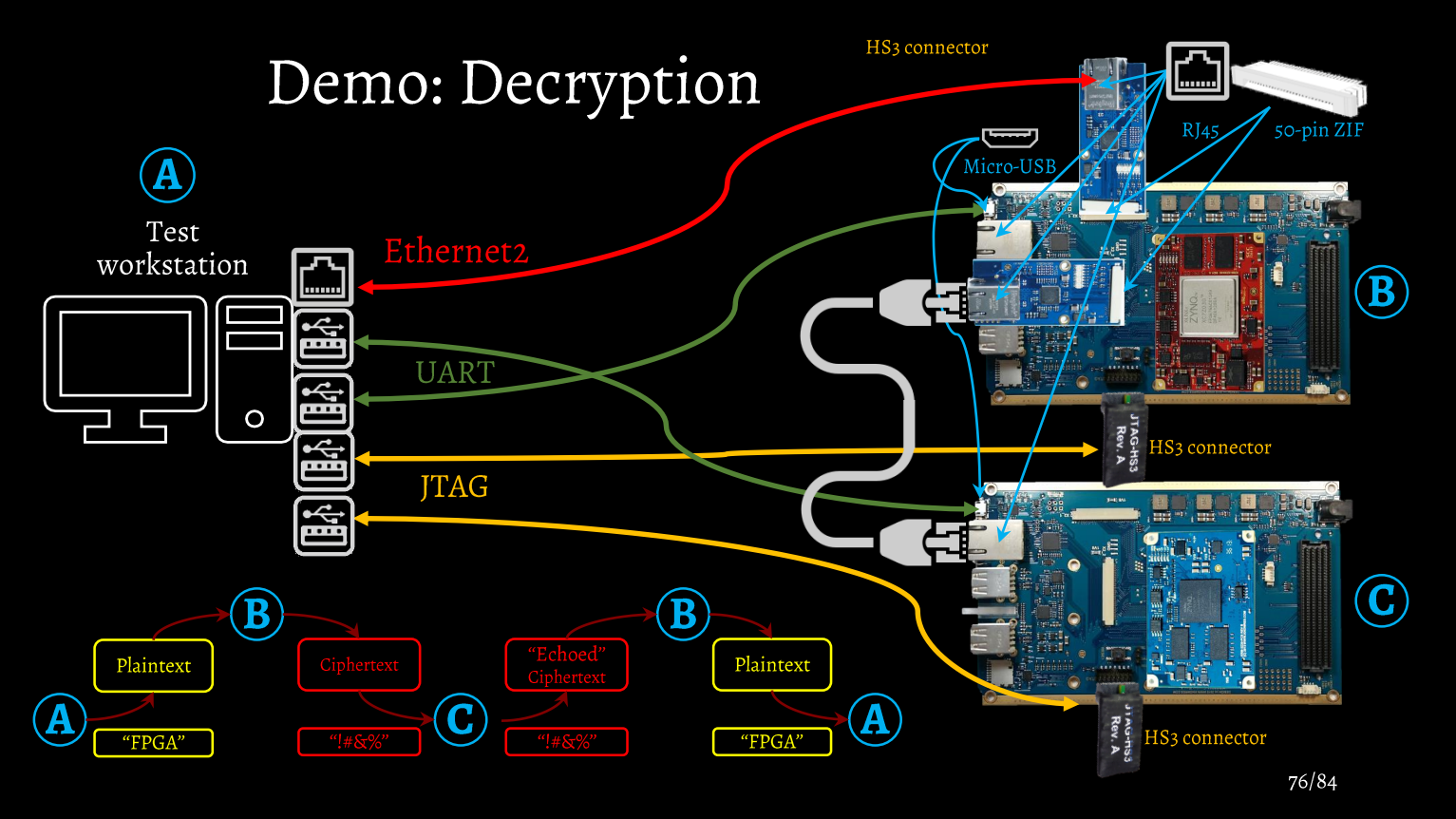
Demo: Encryption



Demo: Encryption

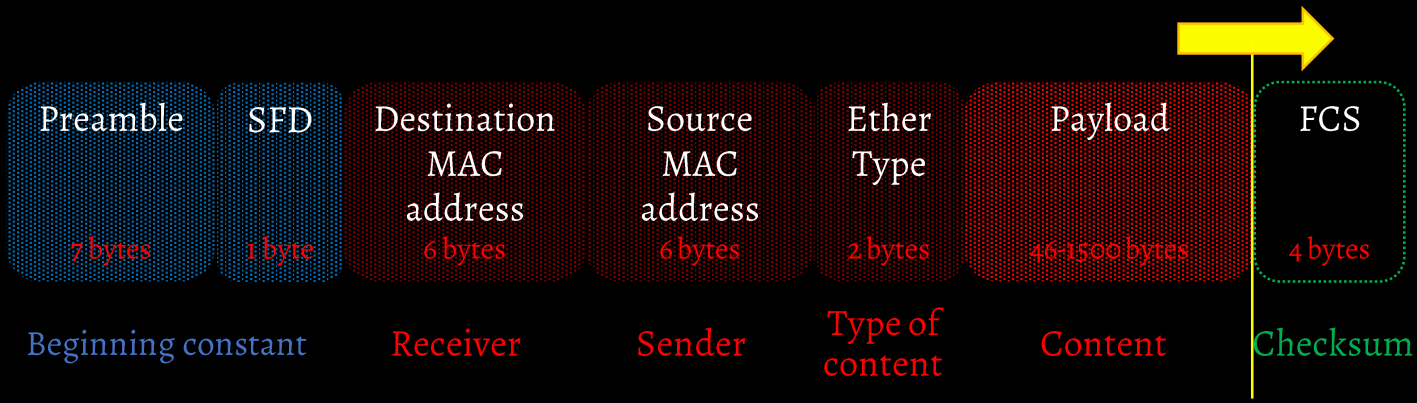


Demo: Decryption

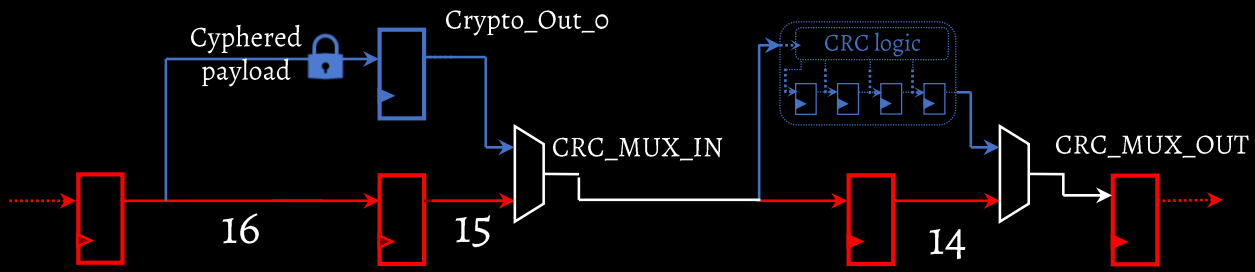


Demo: Decryption

Checksum generation



Checksum generation

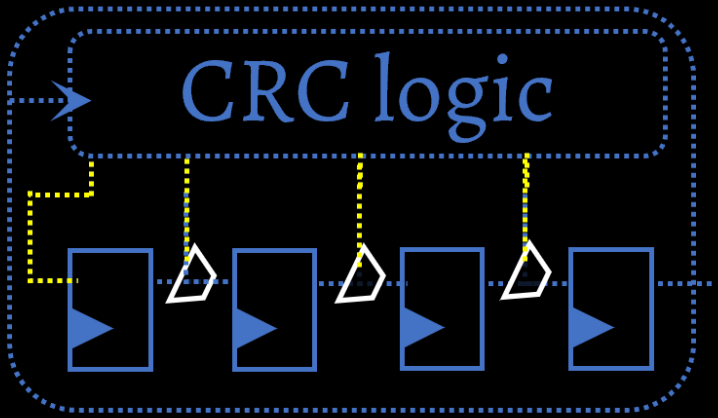



```
CRC_MUX_OUT <=  CRC_OUT_o when crc_append = '1' else PIPELINE(14);
CRC_MUX_IN <= CRYPTO_OUT_o when is_crypto_detected_reg = '1' else PIPELINE(15);
```

Flag raised when the last payload's byte crosses PIPELINE(14)

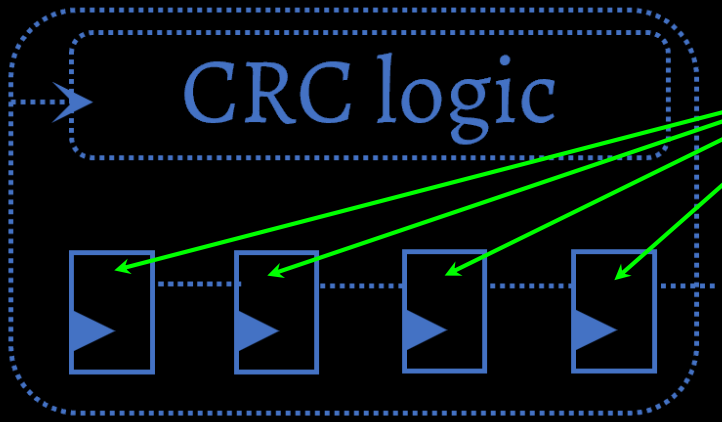
Flag raised when the custom IPv4 sequence is detected

Checksum generation




```
CRC_IN_3 <= crc_out(31 downto 24) when crc_append = '0' else (OTHERS => '0');
CRC_IN_2 <= crc_out(23 downto 16) when crc_append = '0' else CRC_OUT_3;
CRC_IN_1 <= crc_out(15 downto 8) when crc_append = '0' else CRC_OUT_2;
CRC_IN_0 <= crc_out(7 downto 0) when crc_append = '0' else CRC_OUT_1;
```

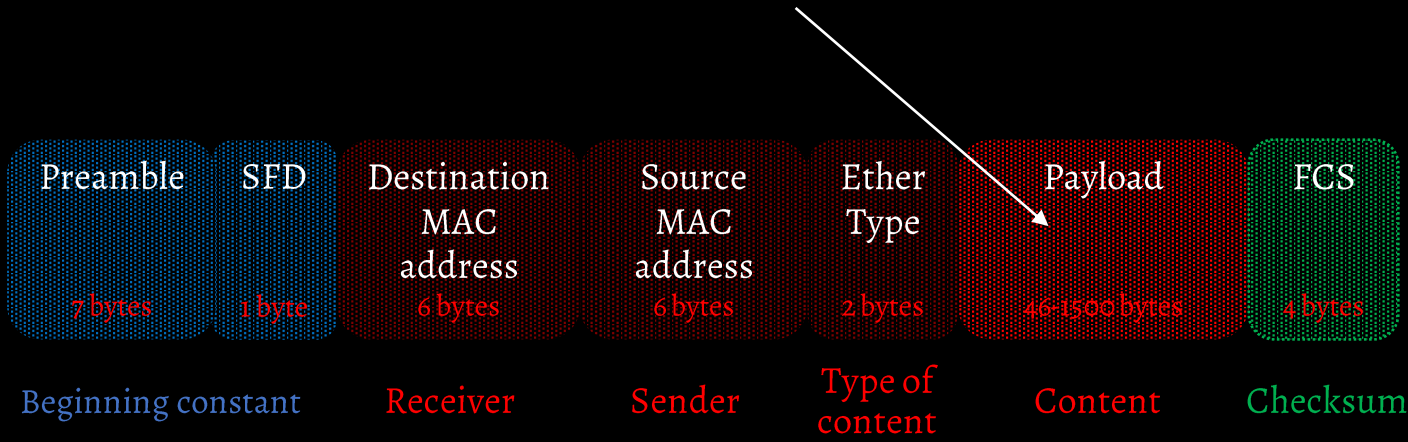
Checksum generation



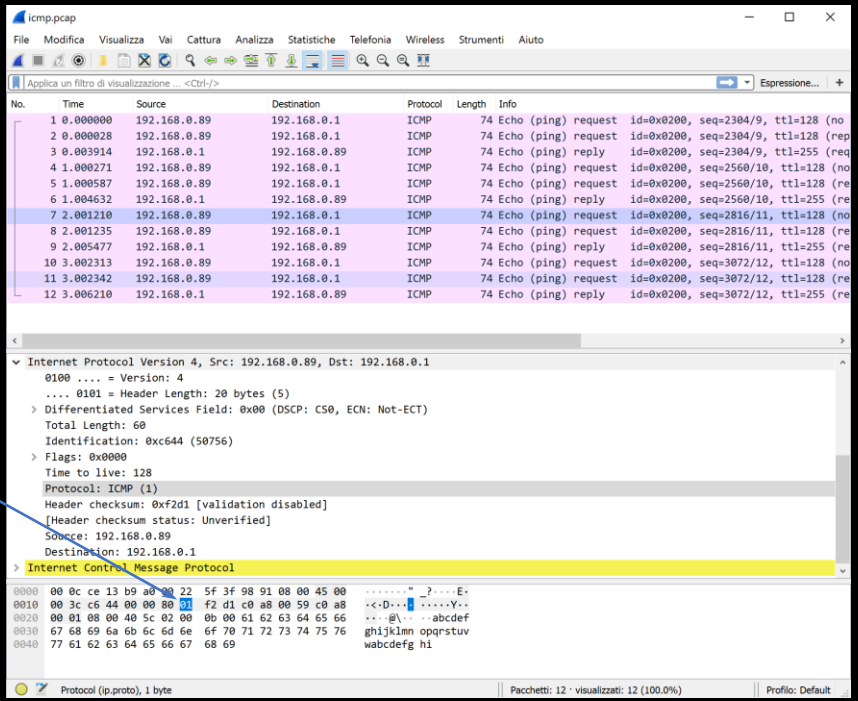

```
CRC_OUT_i : ff_gen
   generic map (NBITS =>
DEPTH)
port map(
   CLK => CLK,
   RST => RST,
   DIN => CRC_IN_i,
   DOUT => CRC_OUT_i
); (OTHERS => '0');
```

32-bit checksum stored in 4 chunks

Moving towards a higher level...



Moving towards a higher level...

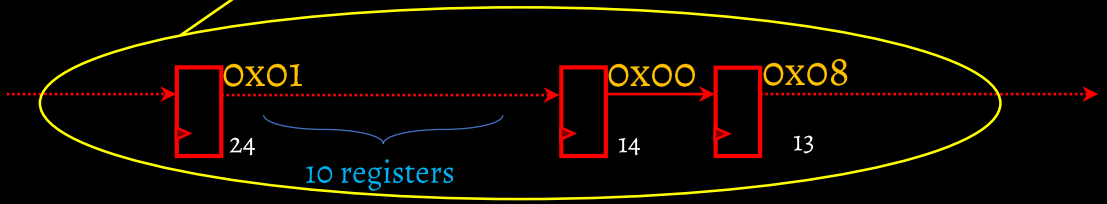


Byte 0x18

Moving towards a higher level...

Killing ICMP packets!

```
when ICMP_KILL =>
    if (is_icmp = '1') then
      state_next <= DROP;
    else
      state_next <= WAIT_TILL_DONE;
    end if;
```



Achievements

- Transparent modification of data
- Real-time equivalent
- Pipelining allows multiple firewall-feature implementation
- The whole time of flight (192 ns) is below avg latency
- P2P Encryption

Future Upgrades

- Filter upper layer protocols (IP, etc.)
- Deep Packet Inspection (custom word lookup in payload, etc.)

Questions?

Thank you for your attention!

Follow me on LinkedIn: linkedin.com/in/matteocollura