EE542
Lecture 24: Internet and Cloud

Internet and Cloud Computing

Young Cho
Department of Electrical Engineering
University of Southern California

Internet Security History

- Robert Tappan Morris
  - Launched a worm (grad student at Cornell) that had a major impact
  - Application designed to count the number of systems on the Internet
    - a conceptual flaw in how rlogin, rsh, and rexec authenticate connections
    - the archaic remote debug feature in Sendmail
    - a buffer overflow in the finger daemon
    - the Worm attempted far more propagation attempts than were necessary, causing targeted machines to slow dramatically from resource starvation
  - He was arrested, found guilty, and sentenced
    - 400 hours of community service
    - Fine of $10,050
- Today
  - Tenured Professor at MIT
  - Co-founder of a venture incubator (Y-Combinator)

Internet Security History

- David L. Smith
  - Melissa Worm recognized on 03/26/1999
  - First major mail worm
  - Word macro virus – attacked Microsoft's Outlook and Word programs (semi-active)
    - First 50 addresses in the recipients' address book
    - Shut down Internet mail systems that got clogged with infected e-mails
  - Received a 10 year sentence
  - Reduced to 20 month prison sentence and $5000 fine by working for FBI
- Today
  - Catching perpetrators of internet-related crimes

Internet Security History

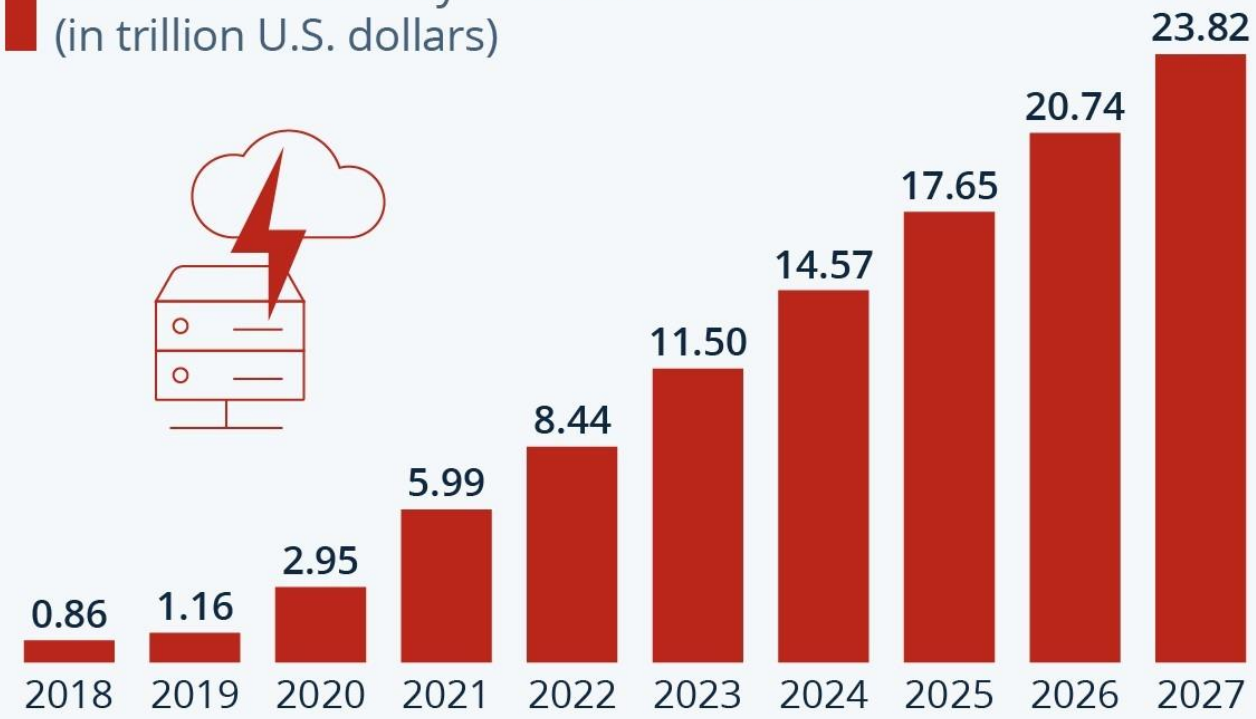
- Code Red (2001)
  - exploited buffer overflow attack in Microsoft's Internet Information Server
  - Very quickly spreading
  - 2nd version scanned logically adjacent IP addr
- Nimda (2001)
  - E-mail worm
  - Bug in Explorer and Outlook
  - spread through Windows shares, and an old buffer overflow in IIS

Estimated Financial Damages

- Morris, 1988 - 60,000 computers - $10 to $100M damage
- VBS/Loveletter, 2000 - $5.5B damage
- Code Red, 2001 - $2.6B damage
- Nimda, 2001 - $635B damage.
- SQL Slammer, 2003 -150-200k computers
- MS Blaster, 2003 - DoS (denial-of-service)
- MyDoom, 2004 - distributed DoS attack with 1M infected machines
- Witty, 2004 – limited damage but demonstrated that a worm could affect a population of machines and networks whose administrators were actively taking steps to improve security.
- Today??? Multiple Billions with No Clear Solution

Cybercrime Expected To Skyrocket in the Coming Years

Estimated cost of cybercrime worldwide
(in trillion U.S. dollars)

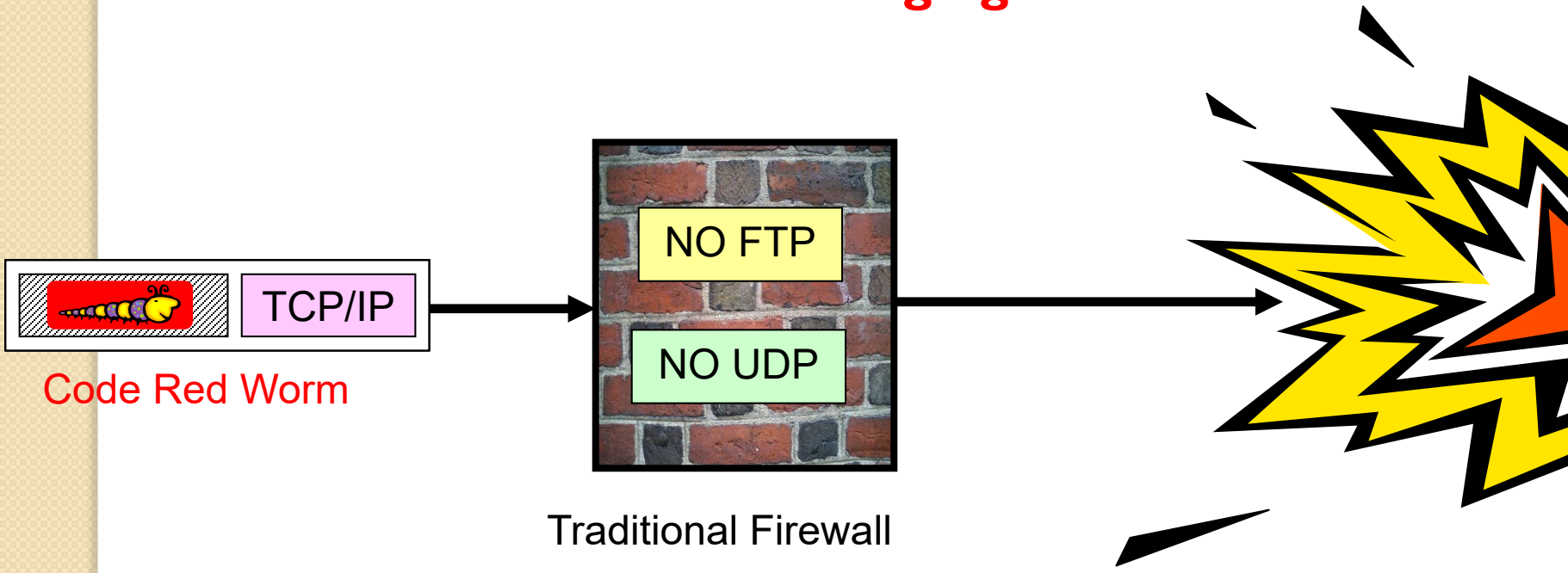


As of November 2022. Data shown is using current exchange rates.
Sources: Statista Technology Market Outlook,
National Cyber Security Organizations, FBI, IMF

statista

Problem

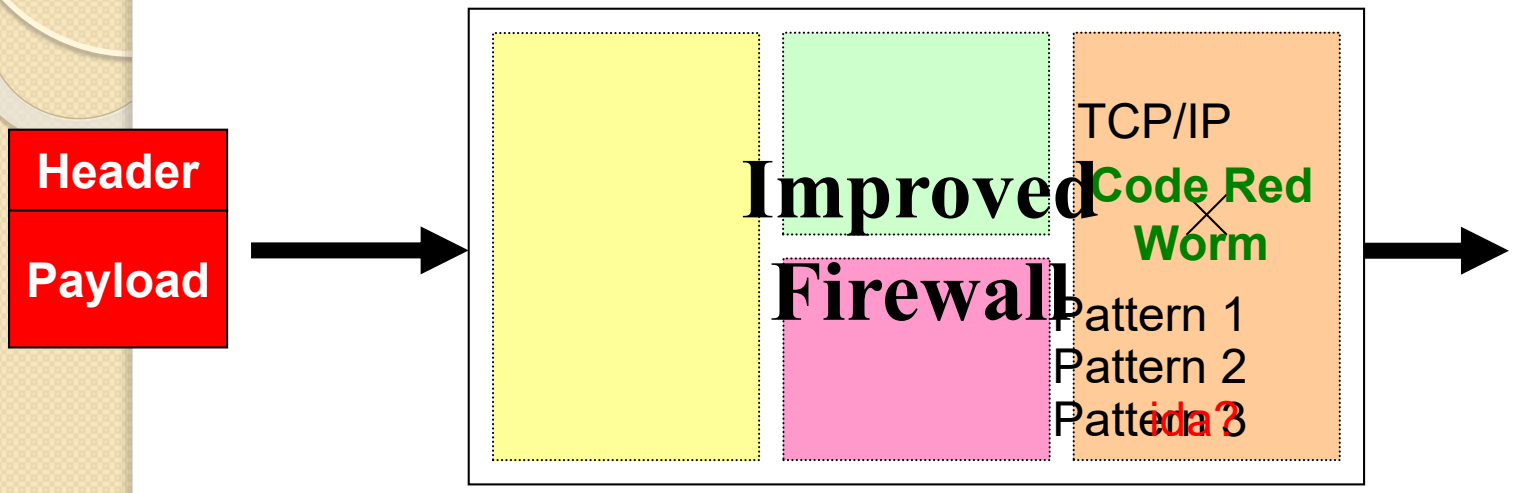
- Traditional Firewalls
  - Only examine packet headers
- Current Attacks
  - Embedded in the packet payload
- **No Fixed Pattern for Emerging Attacks**



Payload Detection Techniques

- 1+ Gbps Reconfigurable Search Engine
- Inexpensive Reconfigurable Search Engine
- Novel Fixed Hardware Search Engine
- Hybrid Pattern Search Engine
- Architecture for Advanced Pattern Search

An Effective Solution



- Packet Normalization
- Static Header Inspection
- Dynamic Payload Inspection
- Intrusion Detection/Prevention

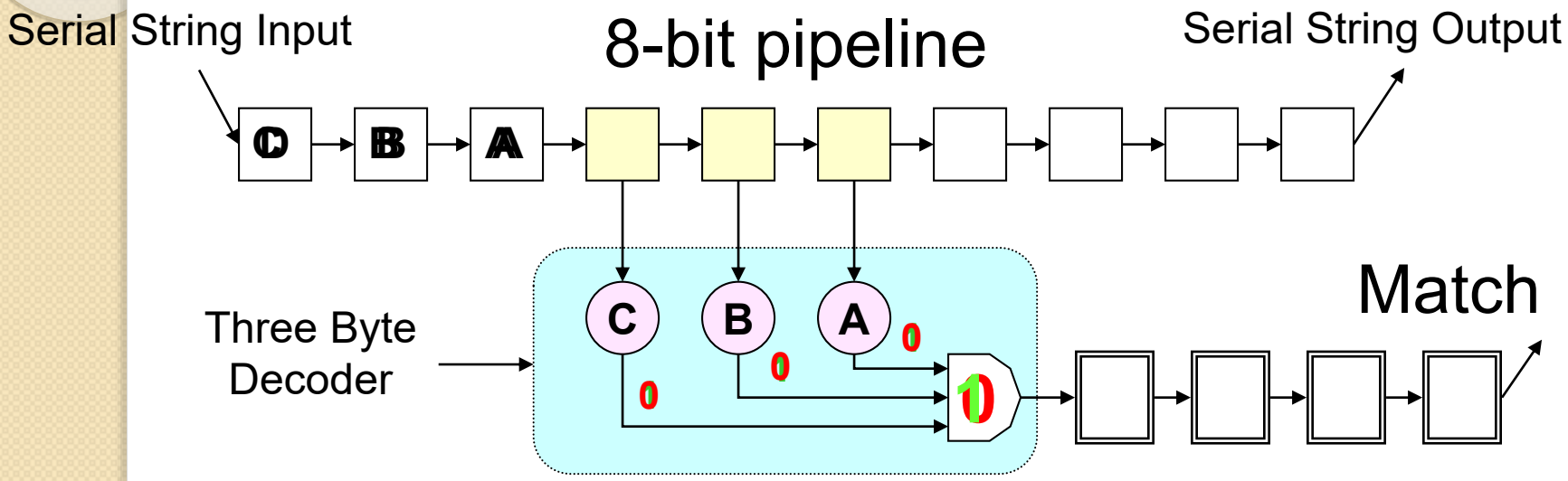
Snort Network Intrusion Detection

- Software Solution
  - Open source network intrusion detection system
  - Updatable signature database (current: 2851 rules)
  - Effective current solution
- Signature
  - action: `alert`
  - static: `tcp any 80 -> any 80`
  - dynamic: `(content: ".ida?"; dsize: >239)`
- Performance
  - Pruned down to 845 rules + dual 1Ghz PIII
  - Drop packets beyond 40 Mbps
  - Not sufficient for Gbps networks

Research

- Leverage FPGA Technology
  - Reconfigurable hardware design
  - Automatic generation of HDL
  - Take advantage of common logic
  - Use embedded memories
- Transition to Custom Hardware
  - Develop ASIC architecture
  - Develop hybrid architecture

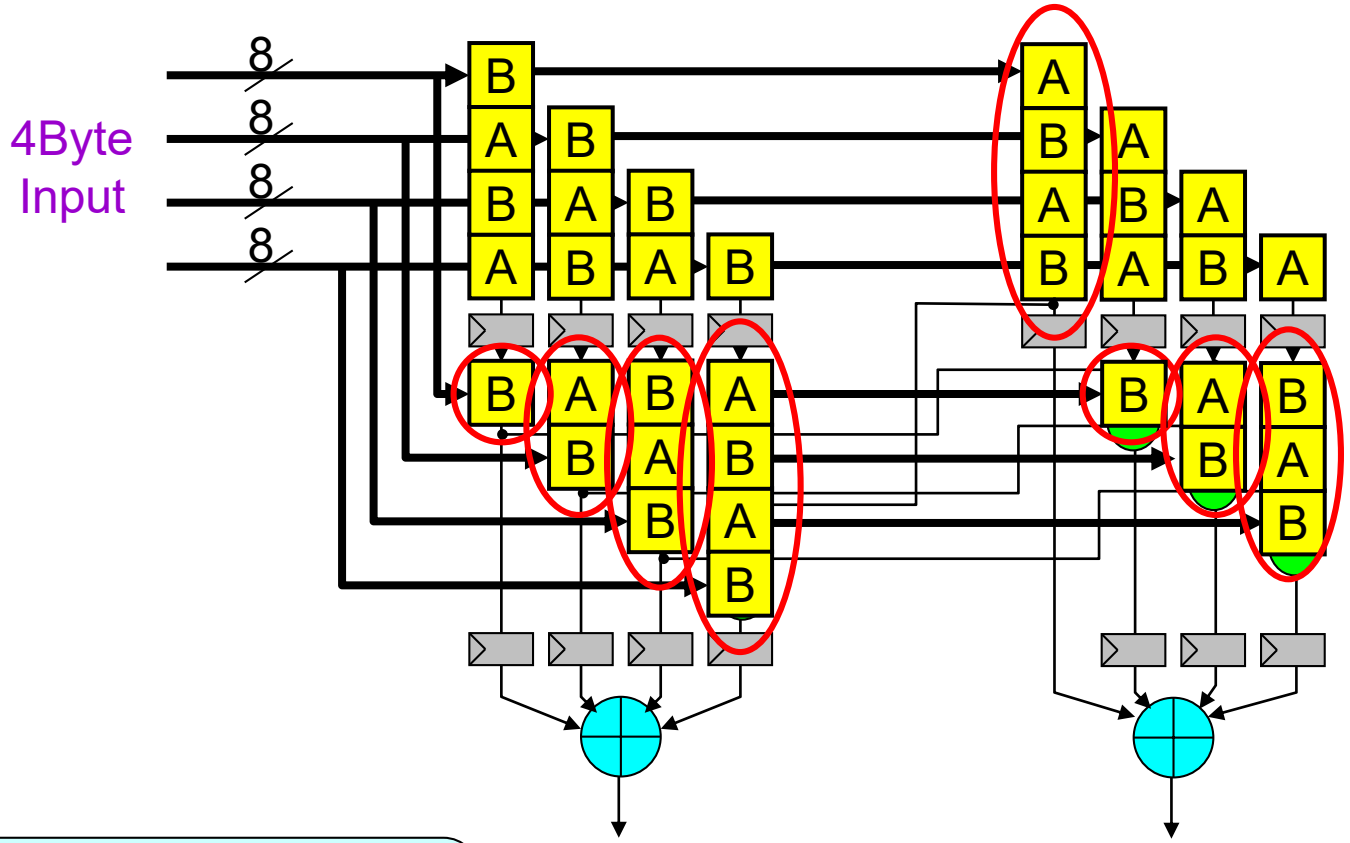
Hardware Pattern Search



Dynamic Payload Inspection Engine

Reuse common sub-structure

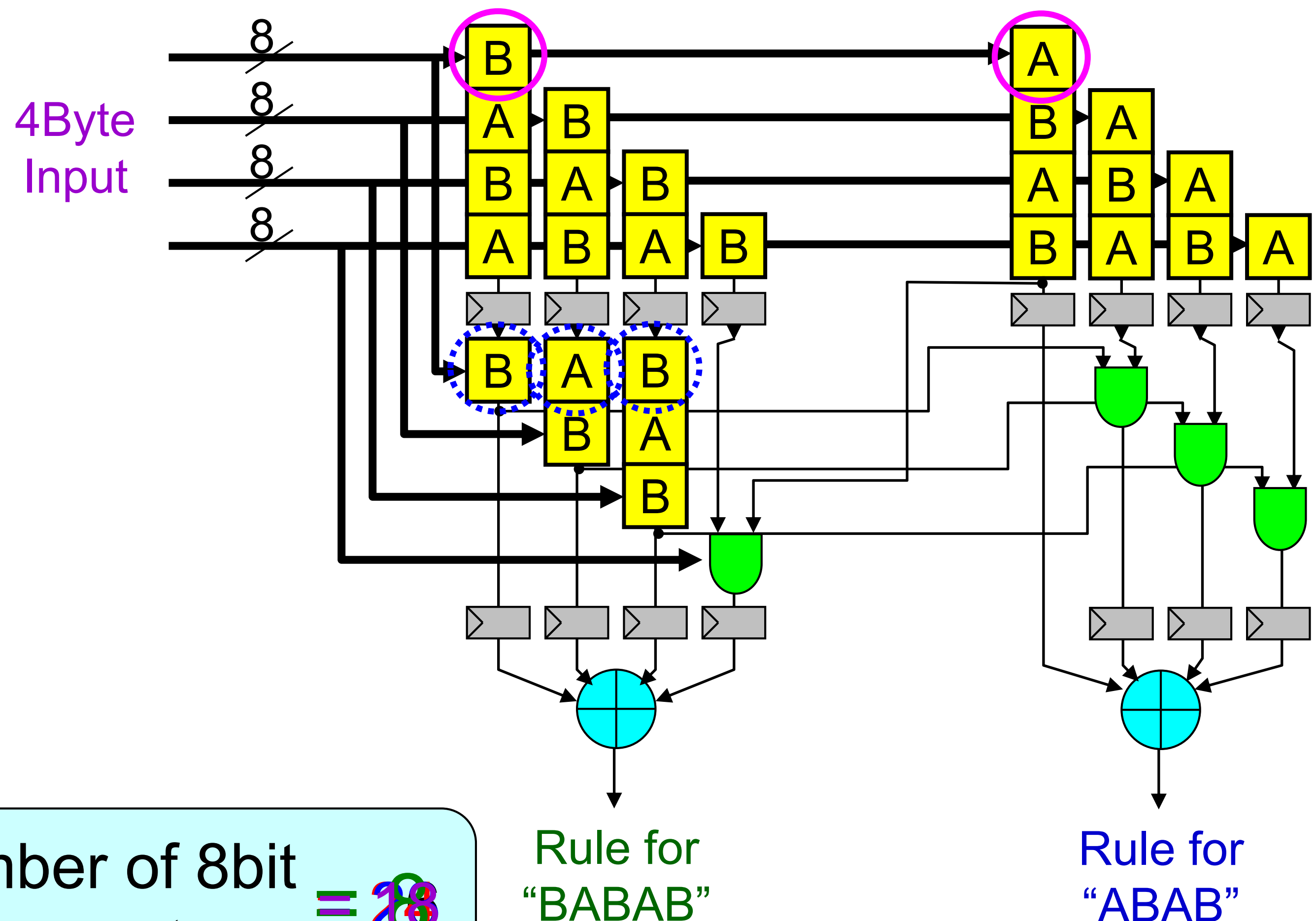
- Eliminate Duplicate Substring Comparators



Number of 8bit Comparators = 26

Reuse common sub-structure

- Eliminate Duplicate 8-bit Comparators

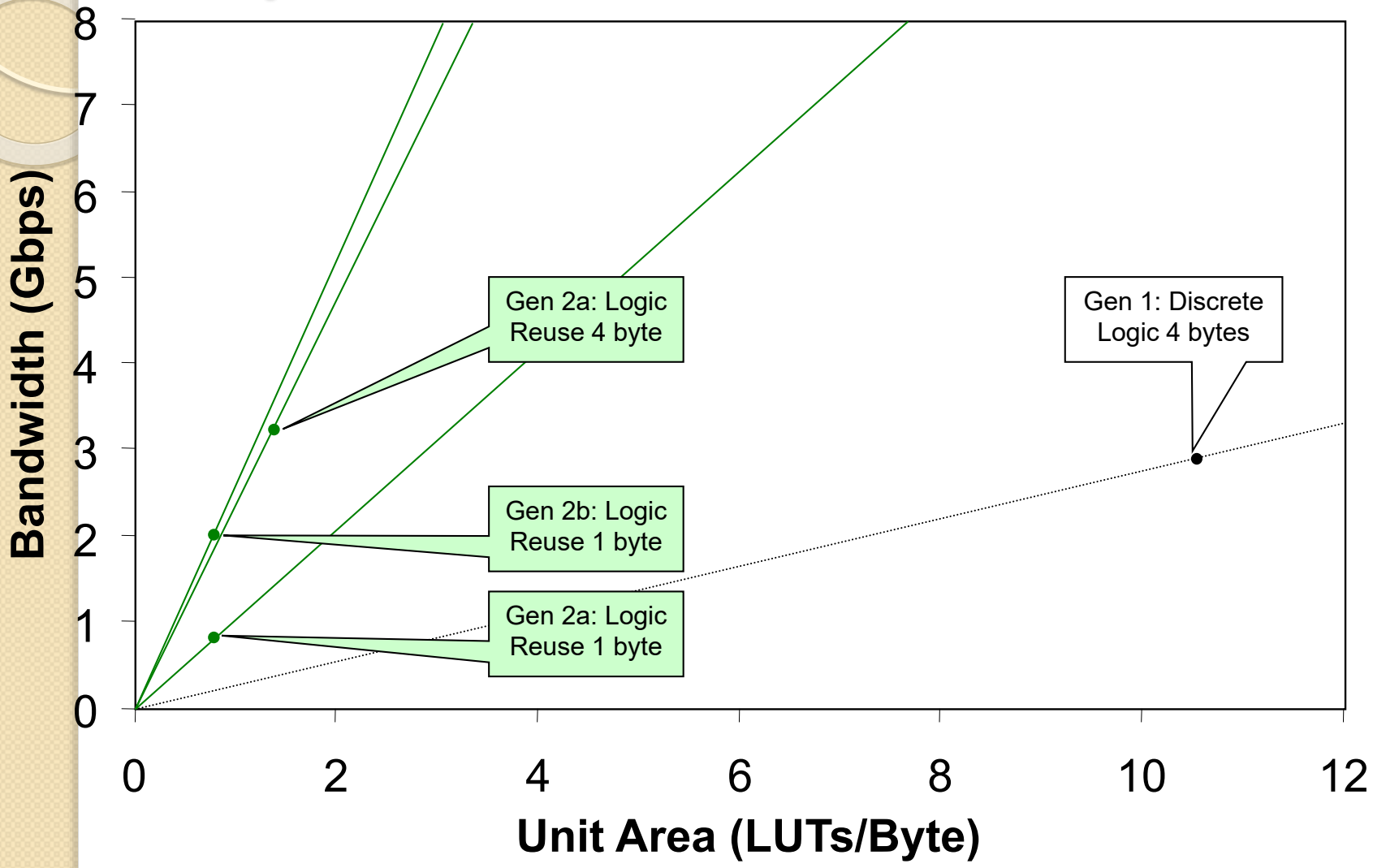


Number of 8bit Comparators = 28 18 8

Second Generation Results

- Logic Reuse Design and Encoder
  - 4 bytes: 1,519 unique patterns (19,021 bytes)
  - 1 byte: 1,661 unique patterns (20,800 bytes)
- Area
  - 4 bytes: 26,607 LUT (~1.40 LUT/byte)
  - 1 byte: 16,930 LUT (~0.81 LUT/byte)
  - Single Xilinx Spartan 3 – 1500 FPGA (< $20)
- Performance
  - 4 bytes: 3.2 – 8 Gbps
  - 1 byte: 0.8 – 2.0 Gbps
- Cost and Performance Trade-off

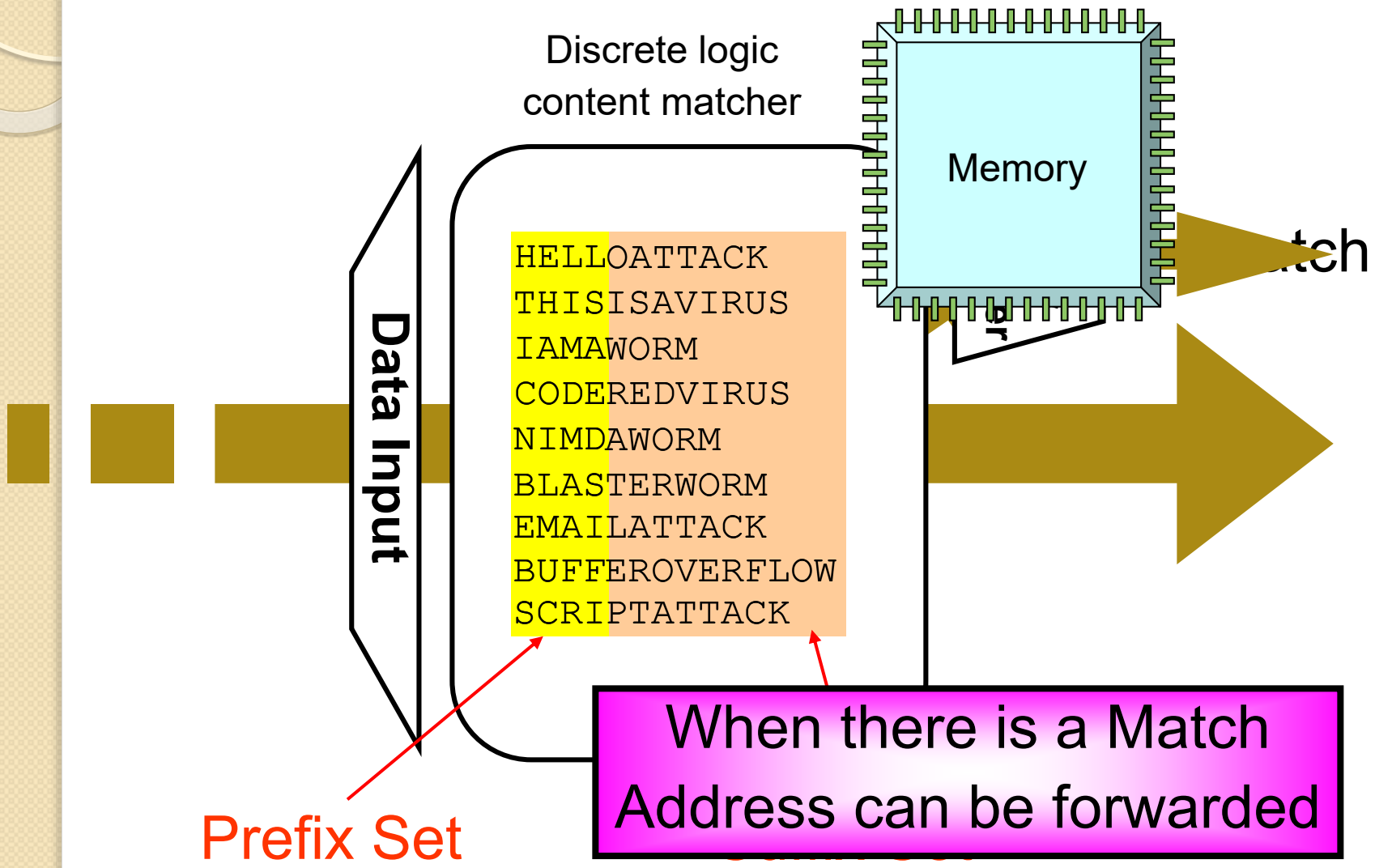
Implementation Result Chart



Embedded Memories in FPGAs

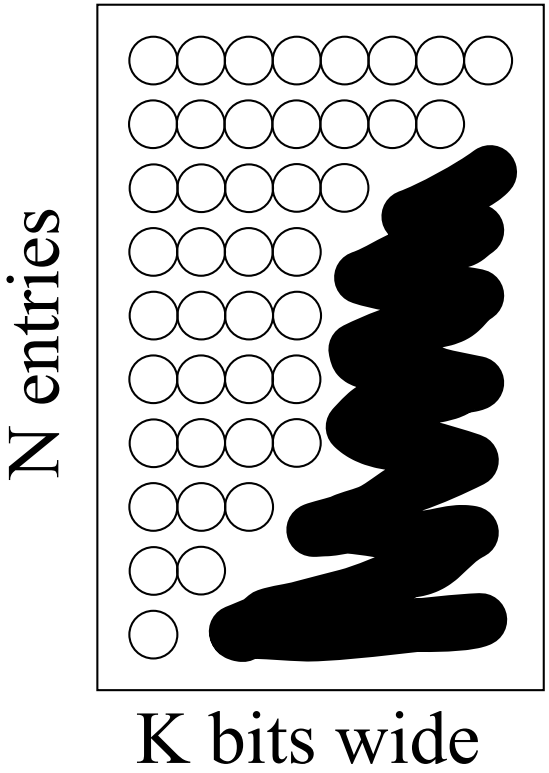
- FPGA Resources
  - Memories are distinct from the logic
  - If not included in the design, it remains unused
- Memory Based Design
  - Off-load reconfigurable logic resource
  - Enable the design to fit in smaller FPGA
- Design Methodology
  - Filtering the pattern based on the substring
  - Verifying filtered pattern with exact match

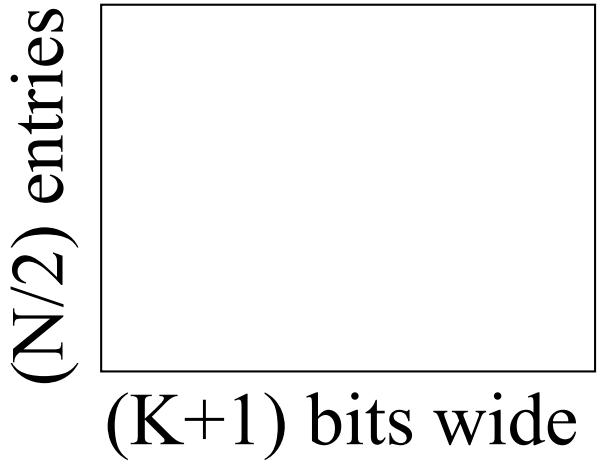
Memory Based Content Match



Increasing Memory Utilization

- Patterns are Different Lengths



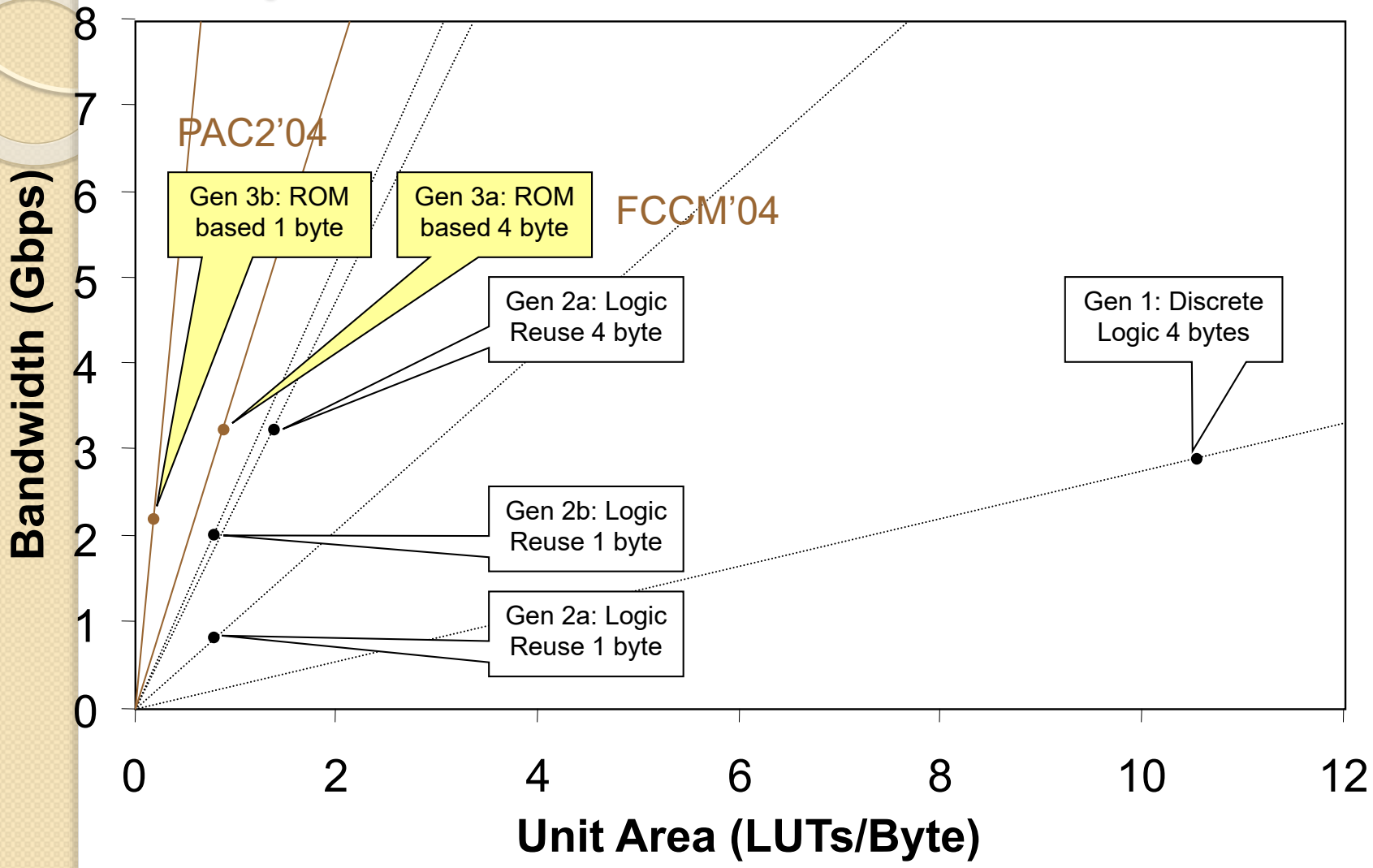


- Increased Memory Utilization

Third Generation

- ROM Based Design
  - 4 bytes: 495 Patterns (6,805 bytes)
  - 1 byte: 1,625 unique patterns (20,800 bytes)
- Area
  - 4 bytes: 6,136 LUT & 90 kbits of memories
  - 1 byte: 4,415 LUT & 162 kbits of memories
  - Maps in Spartan 3 – 400k FPGA (< $6.50/chip)
- Performance
  - 4 bytes: 3.2 Gbps – 8 Gbps
  - 1 byte: 1.6 Gbps – 2.17 Gbps

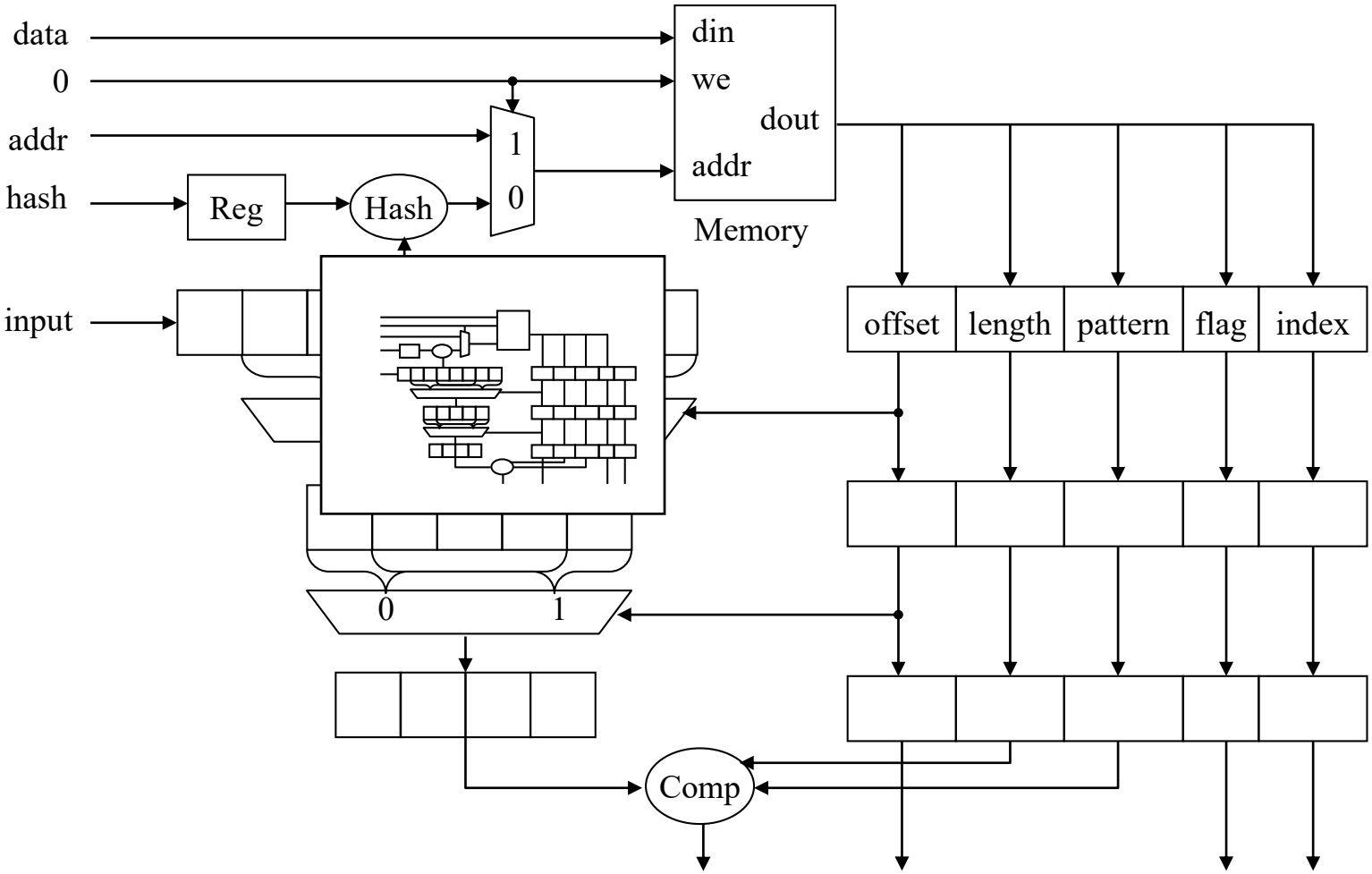
Implementation Result Chart



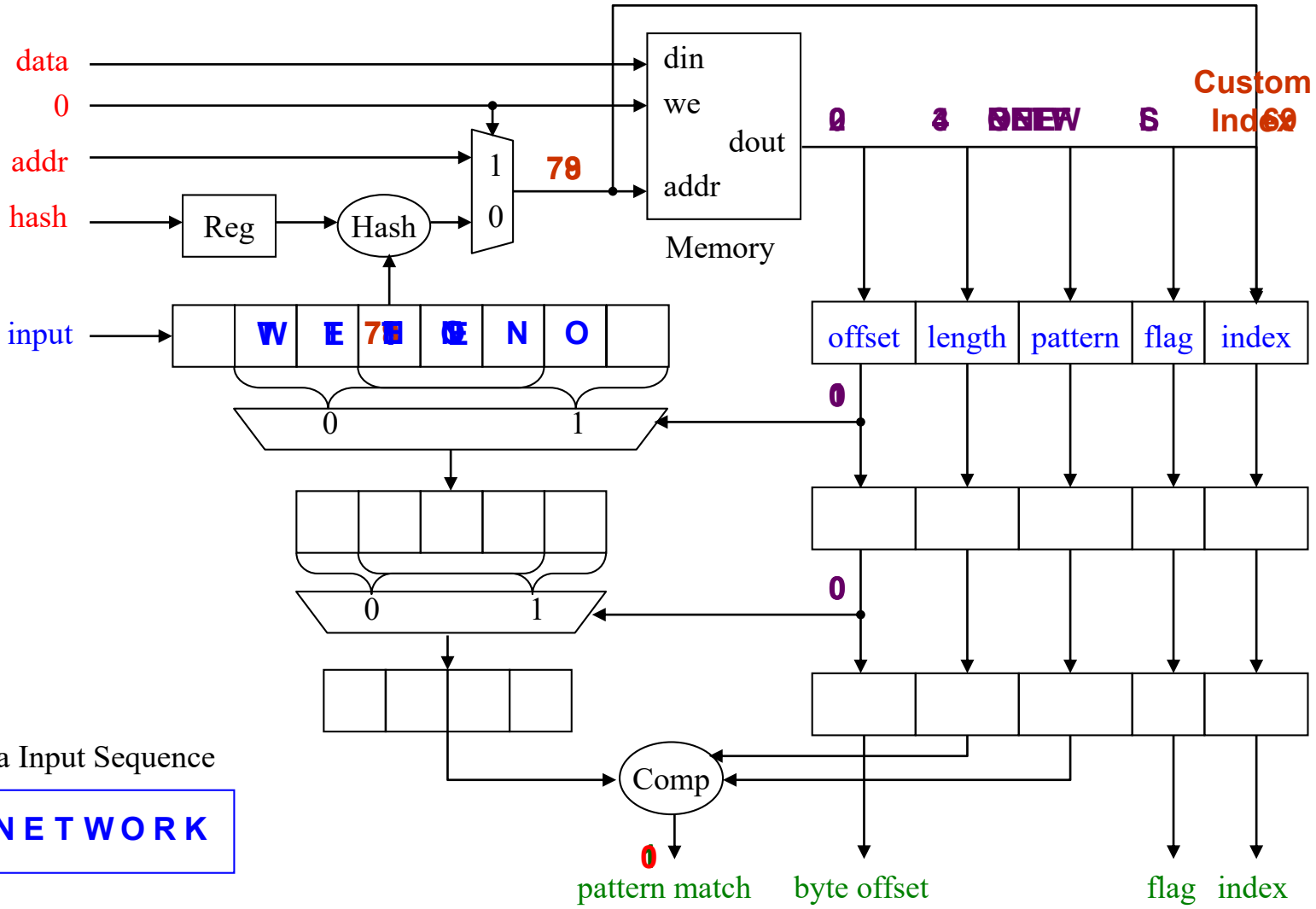
Concerns for Reconfigurable Designs

- Signature Updates
  - New signature produces new design
  - Recompilation time can be minutes to days
  - 30 patterns added between Oct'04 and Jan'05
  - Average of 1 pattern added every 1 to 3 days
- Expectations
  - UCSD – Automatic Worm Fingerprinting
  - New attacks identified at a faster rate
  - Design update should keep up
  - Software programmable solution?

Pattern Detection Module



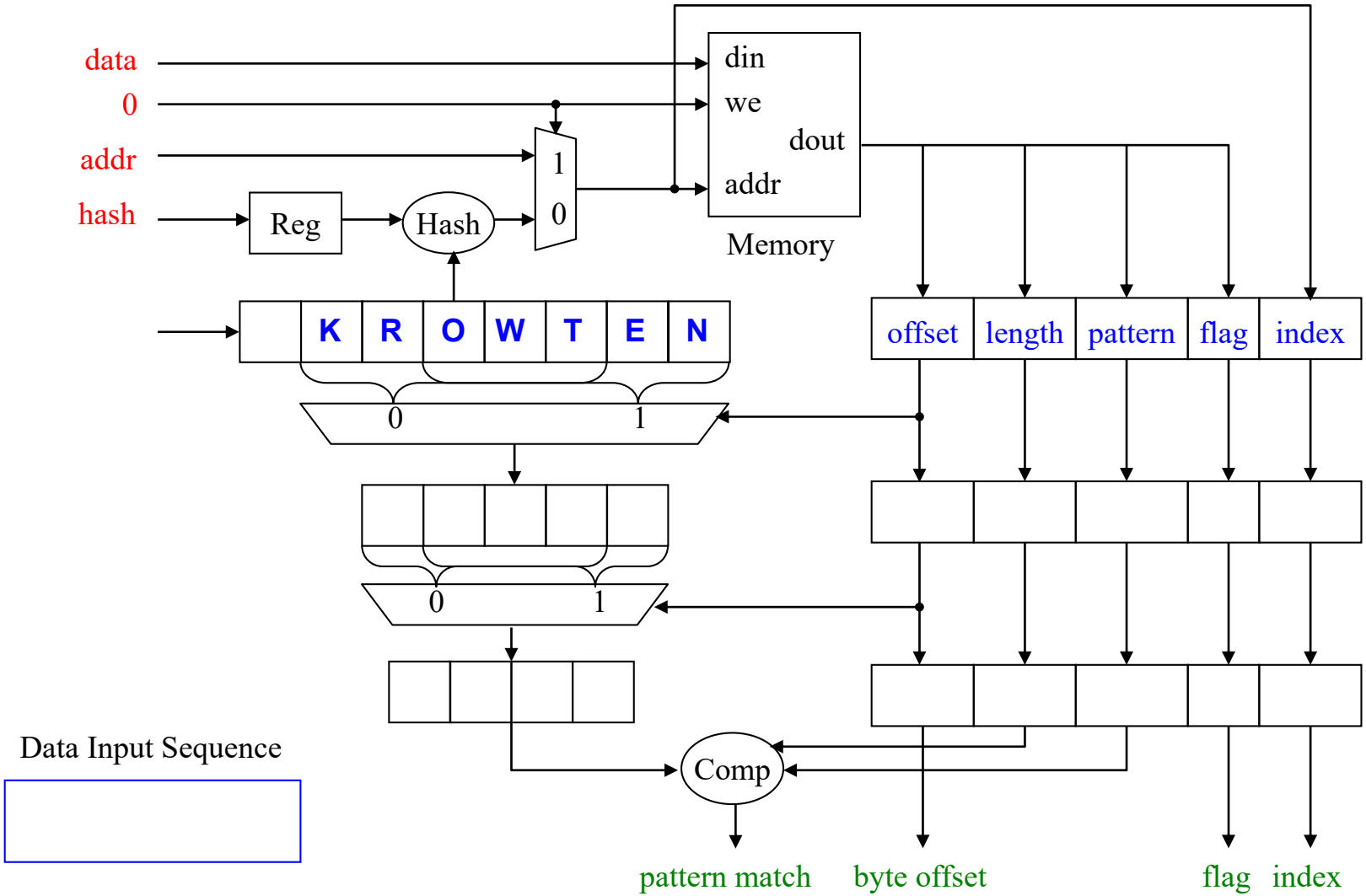
Pattern Detection Module



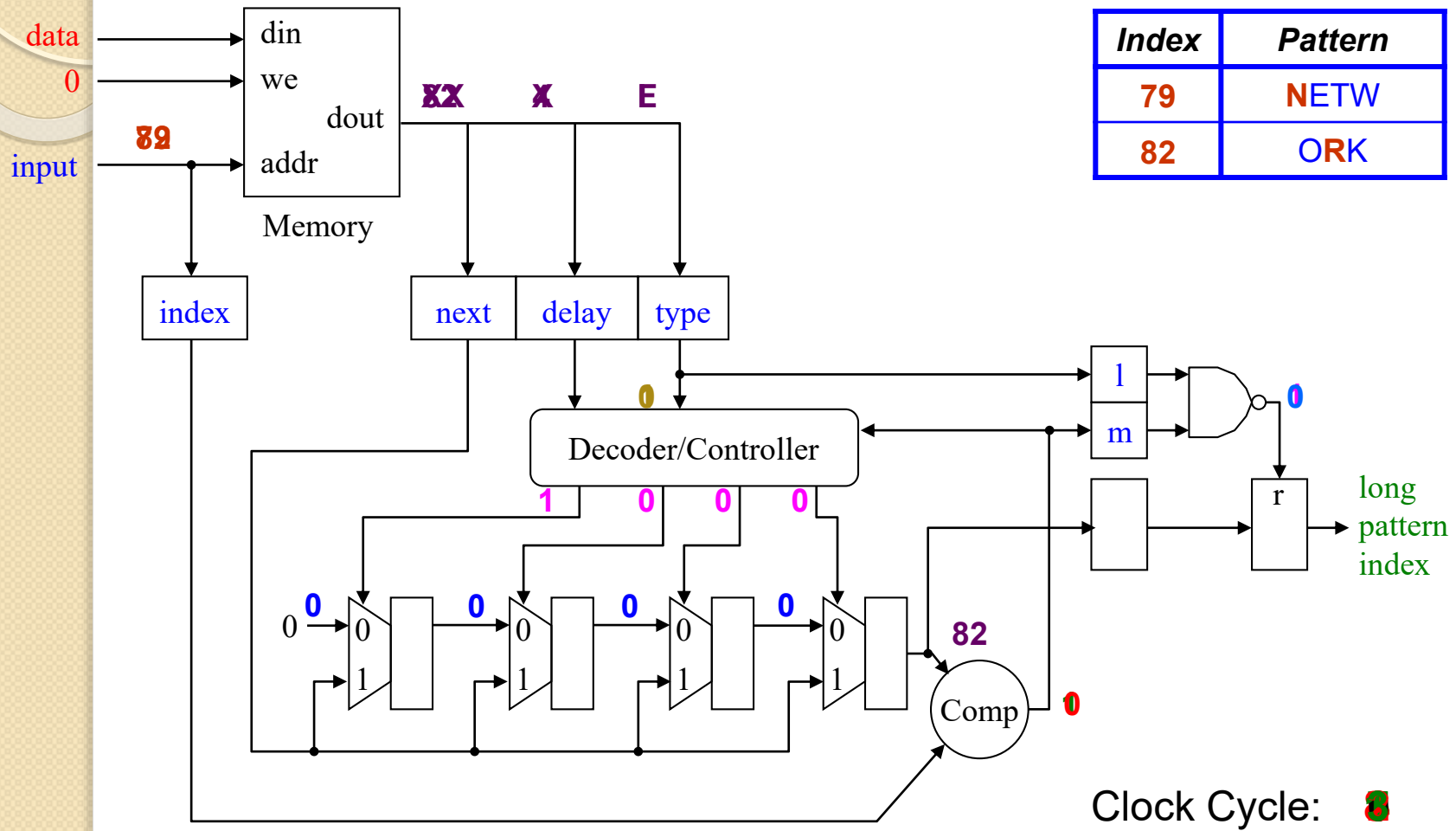
Data Input Sequence

ONETWORK

Pattern Detection Module

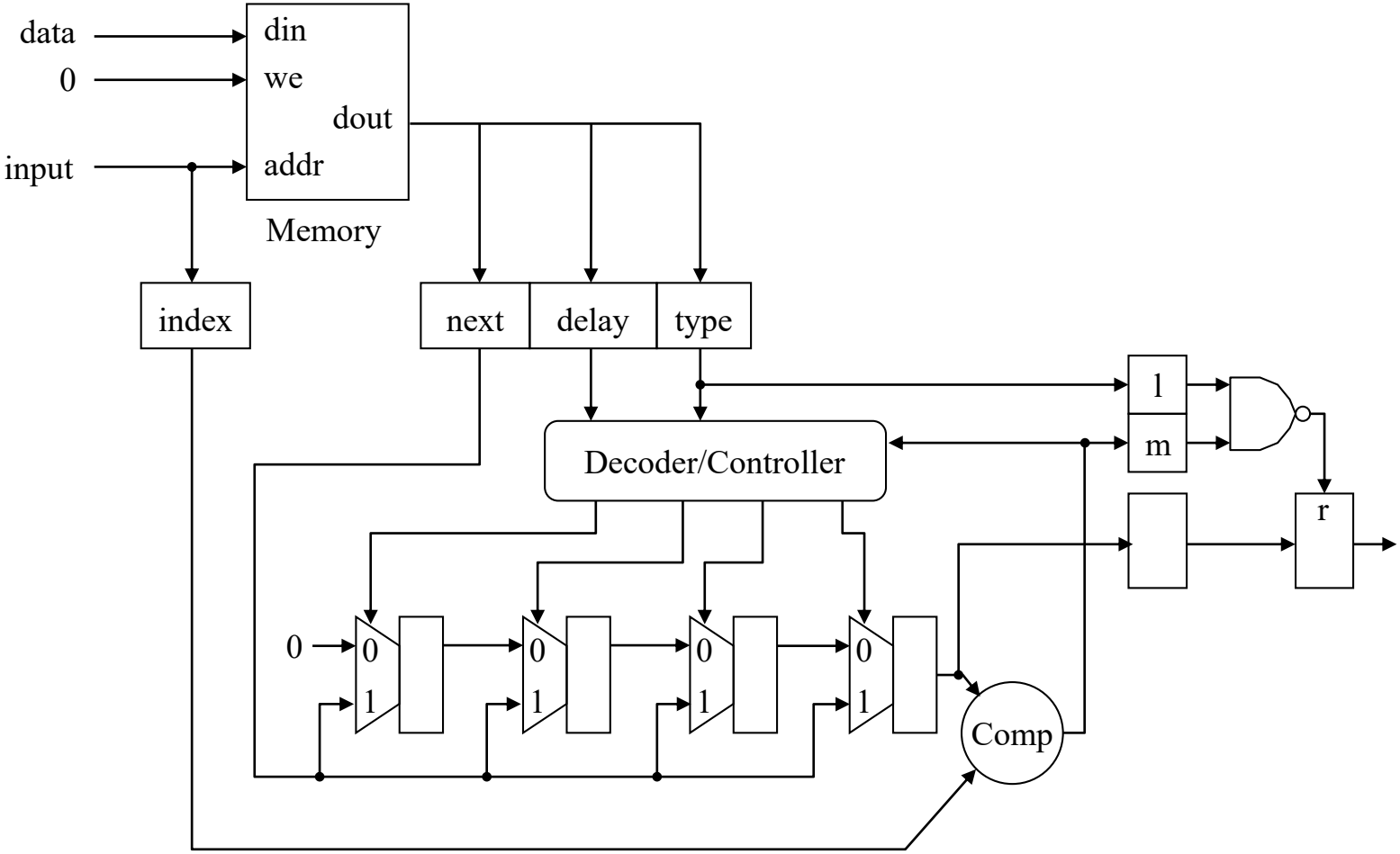


Predictive Long Pattern State Machine

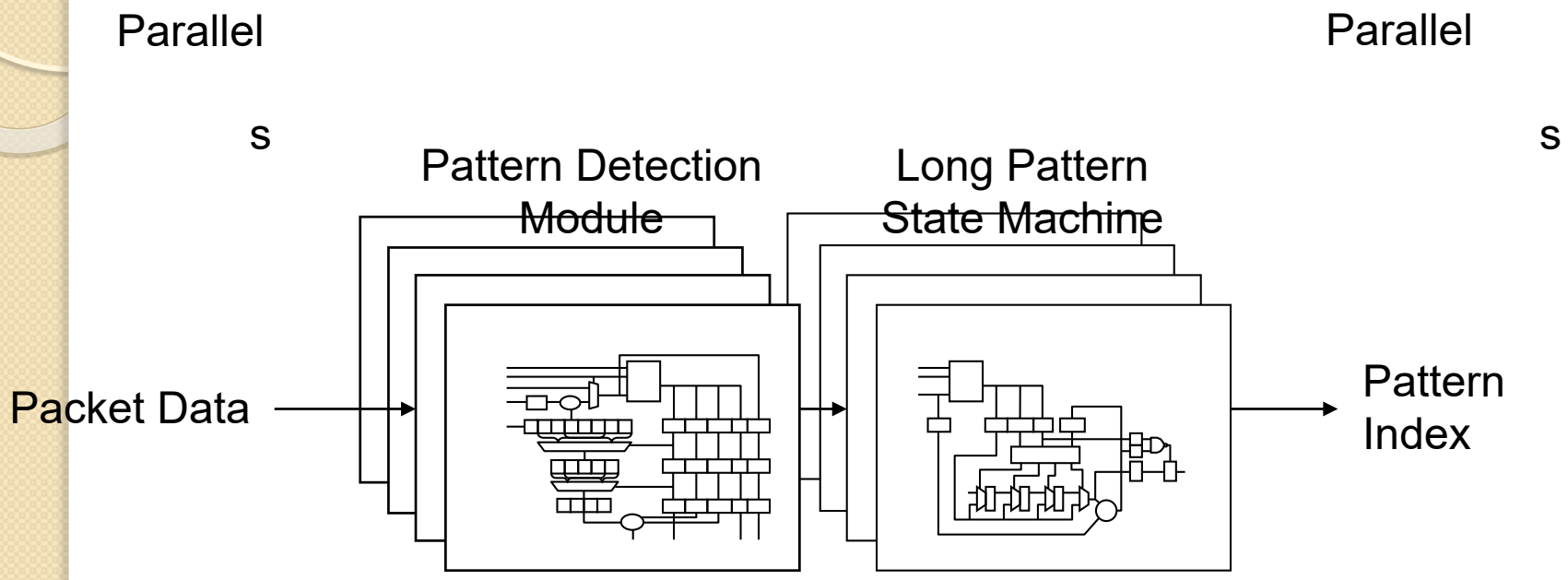


Index	Pattern
79	NETW
82	ORK

Predictive Long Pattern State Machine



Pattern Match Co-processor



- Issue: Patterns may Share Hash Value
- Regular Expressions

Pattern Mapping and Programming

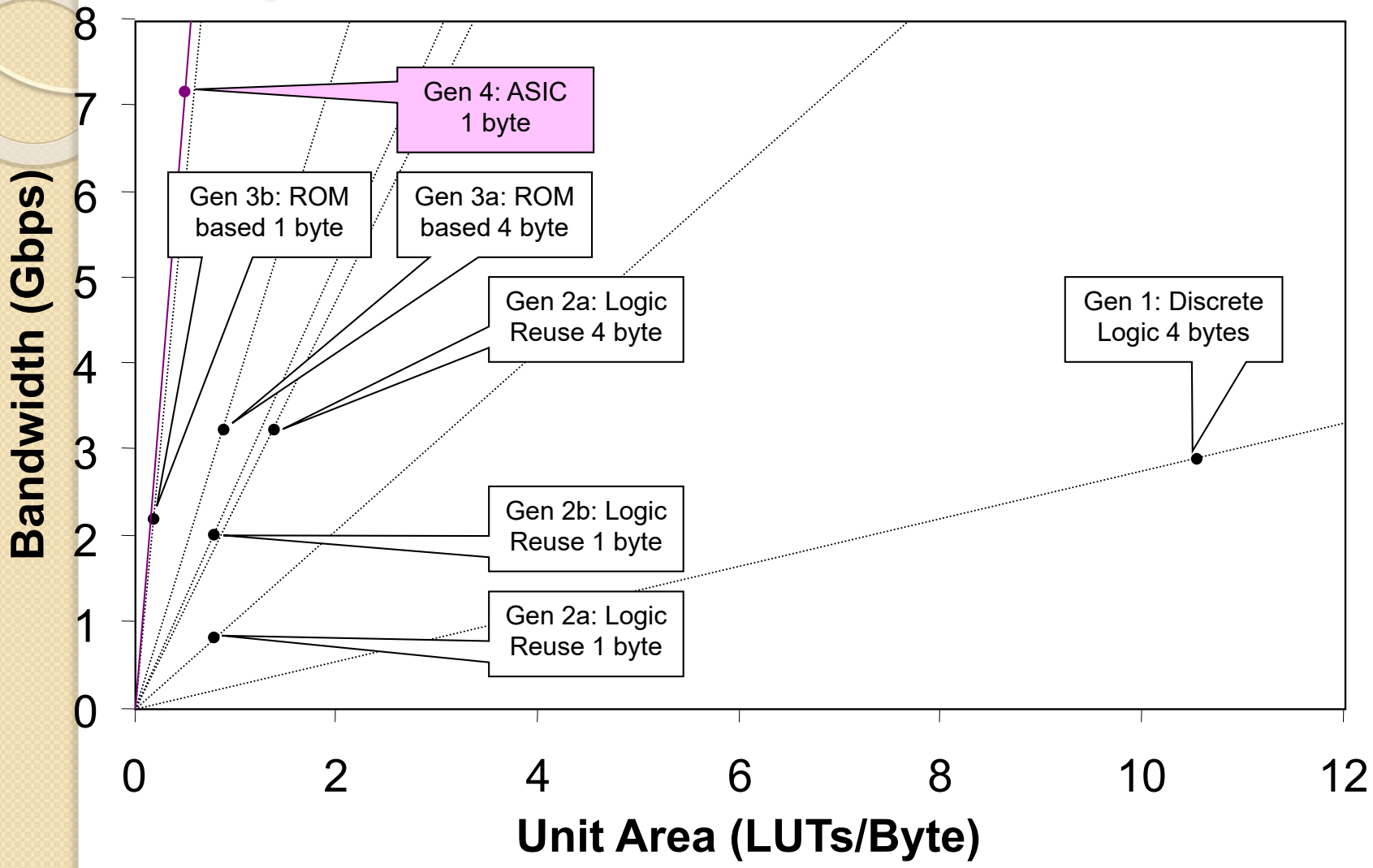
- Mapping Algorithm
  - Break long patterns into short segments
  - Greedy mapping heuristics
- ASIC Design
  - Eight PDM (8 units 512x146bits SRAM = 75KB)
  - Eight LPSM (8 units 512x29bits SRAM = 15KB)
- Execution Time
  - Mapping: 800 msec for entire Snort patterns
  - Programming: SRAM latency

Fourth Generation Result

Module	Area	Units x Area	Critical Path
PDM Logic	0.075 mm^2	0.600 mm^2	< 1.0 ns
LPSM Logic	0.024 mm^2	0.188 mm^2	< 1.0 ns
PDM Memory	0.844 mm^2	6.752 mm^2	1.12 ns
LPSM Memory	0.168 mm^2	1.342 mm^2	1.12 ns
Total Area	**-**	**8.882 mm^2**	**1.12 ns**

7.144 Gbps ← ~ 893 Mhz

Implementation Result Chart

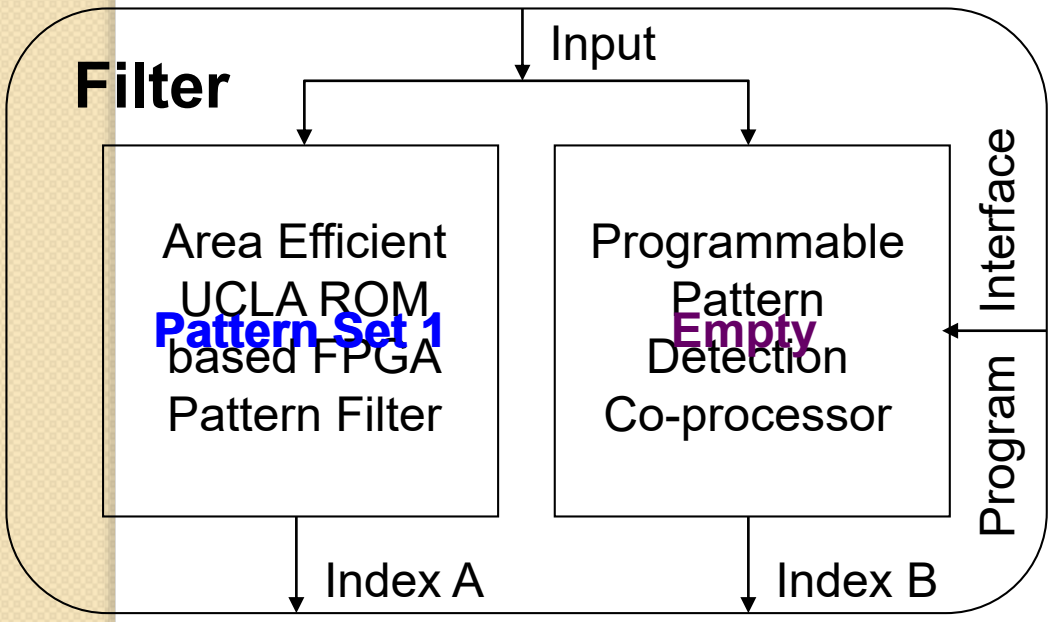


Best of Both Platforms

- Compactness of FPGA design
- Fast Re-programmability of Co-processor
- Option 1: Single FPGA
  - Large reconfigurable filter design
  - Small programmable co-processor
- Option 2: FPGA and small ASIC
  - FPGA for reconfigurable filter
  - Small ASIC for new signatures

Hybrid Filter

New Pattern Set − = Difference



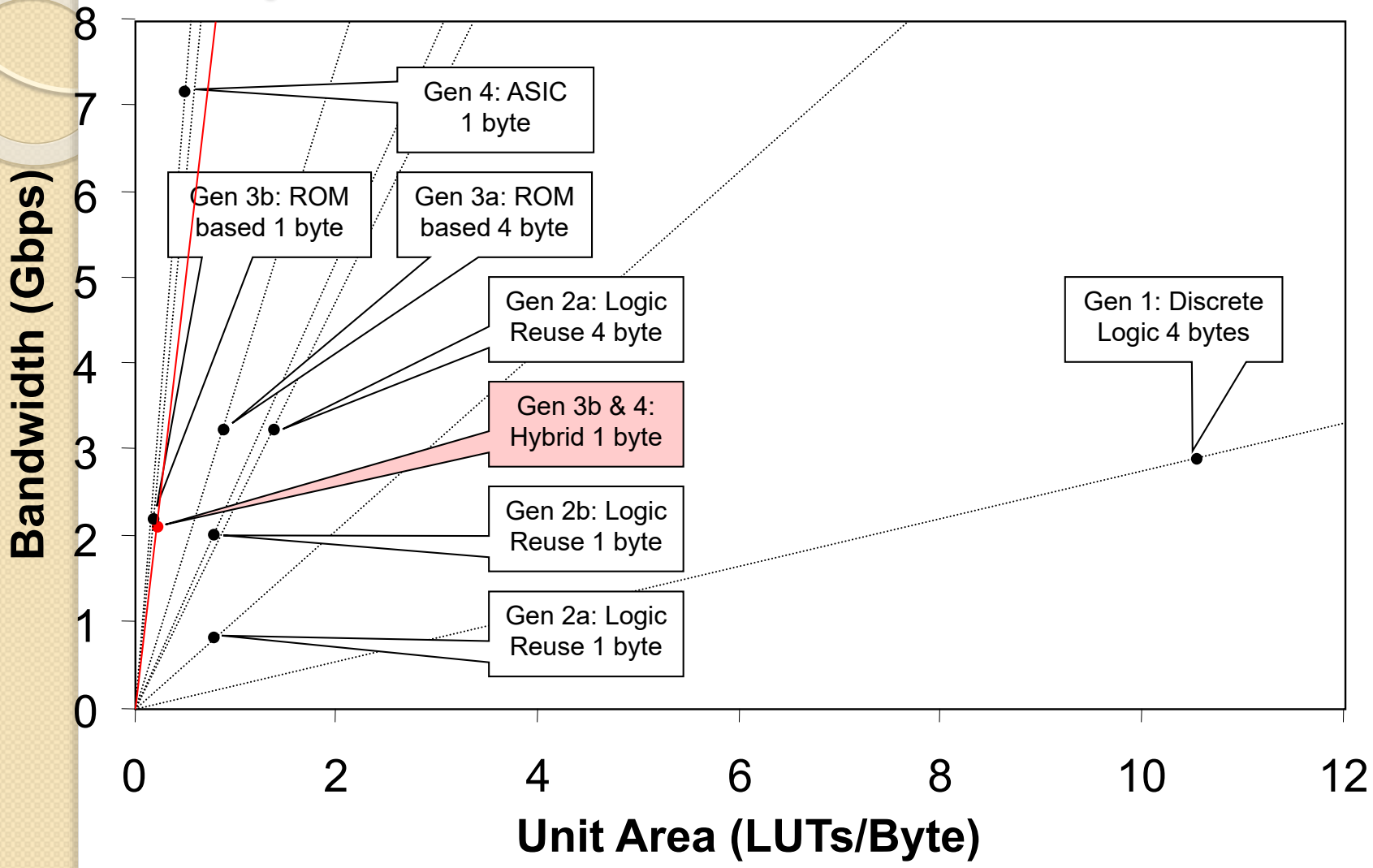
Hybrid Design Implementation

- Single FPGA Option
  - Virtex 4 LX15
  - Custom and Fixed Hardware
- ROM Based Design
  - Filters most of the patterns
  - Automatic hardware generation
- Co-processor
  - Filters updated patterns
  - Four PDM (4 – 256x72bits)
  - One retrospective LPSM (1024x18bits)
  - Relatively small pattern set (less than 100)
  - Simple and fast mapping algorithm

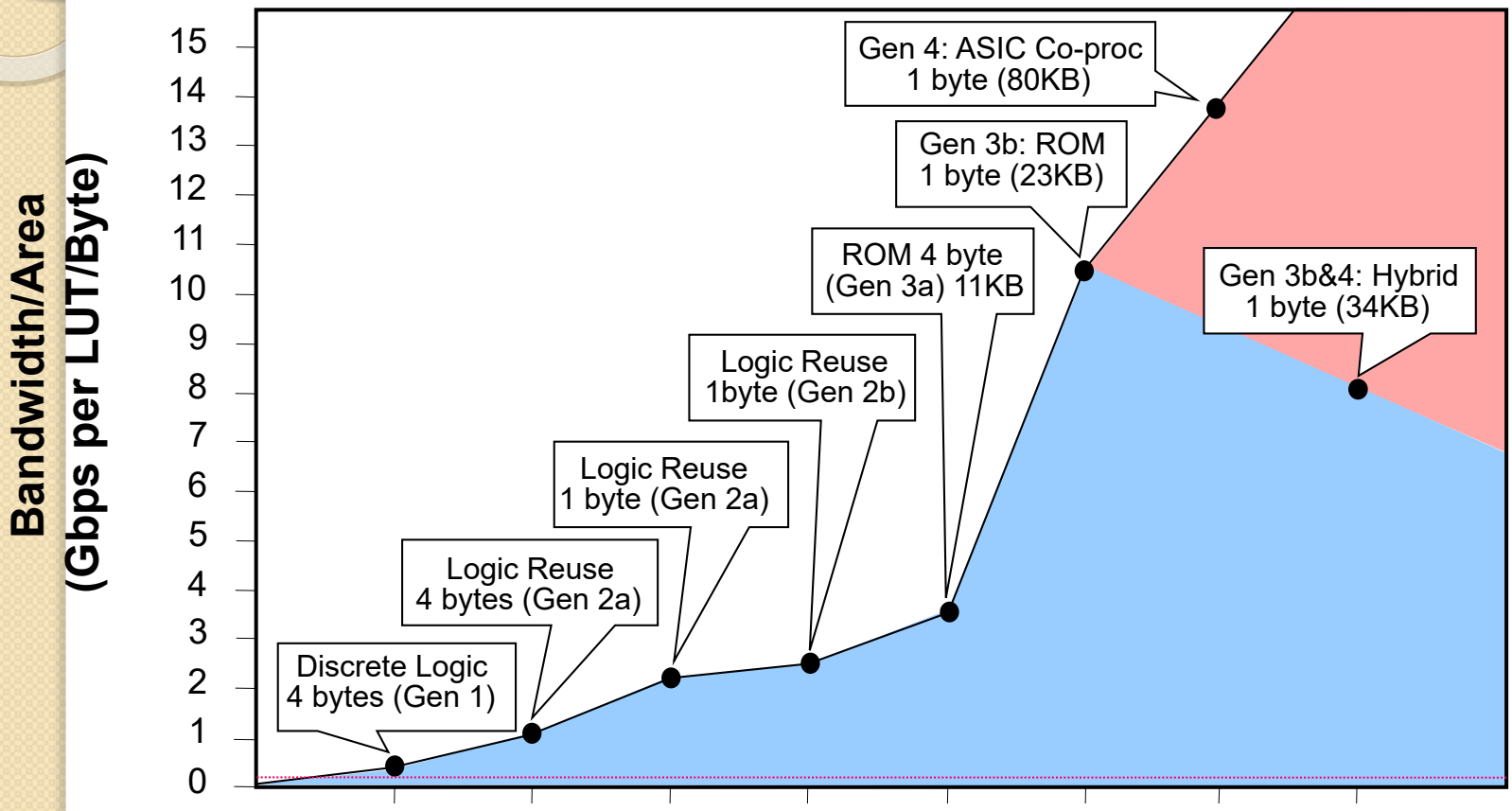
Hybrid FPGA Result

- Design in Single FPGA
  - Oct 19, 2004: 2,031 unique patterns (32,168 bytes)
  - Jan 13, 2005: 2,044 unique patterns (32,384 bytes)
  - Eliminated patterns: 17 patterns (91 bytes)
  - Added patterns: 30 patterns (307 bytes)
- Area (Oct 19 and Small Co-processor)
  - 8,480 LUT (~0.26 LUT/byte) – 2,031 patterns
  - 270 kbits (15 block memories) – 30 patterns
  - Xilinx Virtex 4 LX15
- Performance
  - 2.08 Gbps

Implementation Result Chart



Bandwidth Per Unit Area



Emerging Network Attacks

- Undetectable Attacks
  - Polymorphic attacks
  - No fixed pattern
- Examples already exist in Virus
  - Cascade – self encrypting attack
  - Mutation Engine (MtE)
  - V2P6 – Vary instruction sequences
- Faster and more complex network

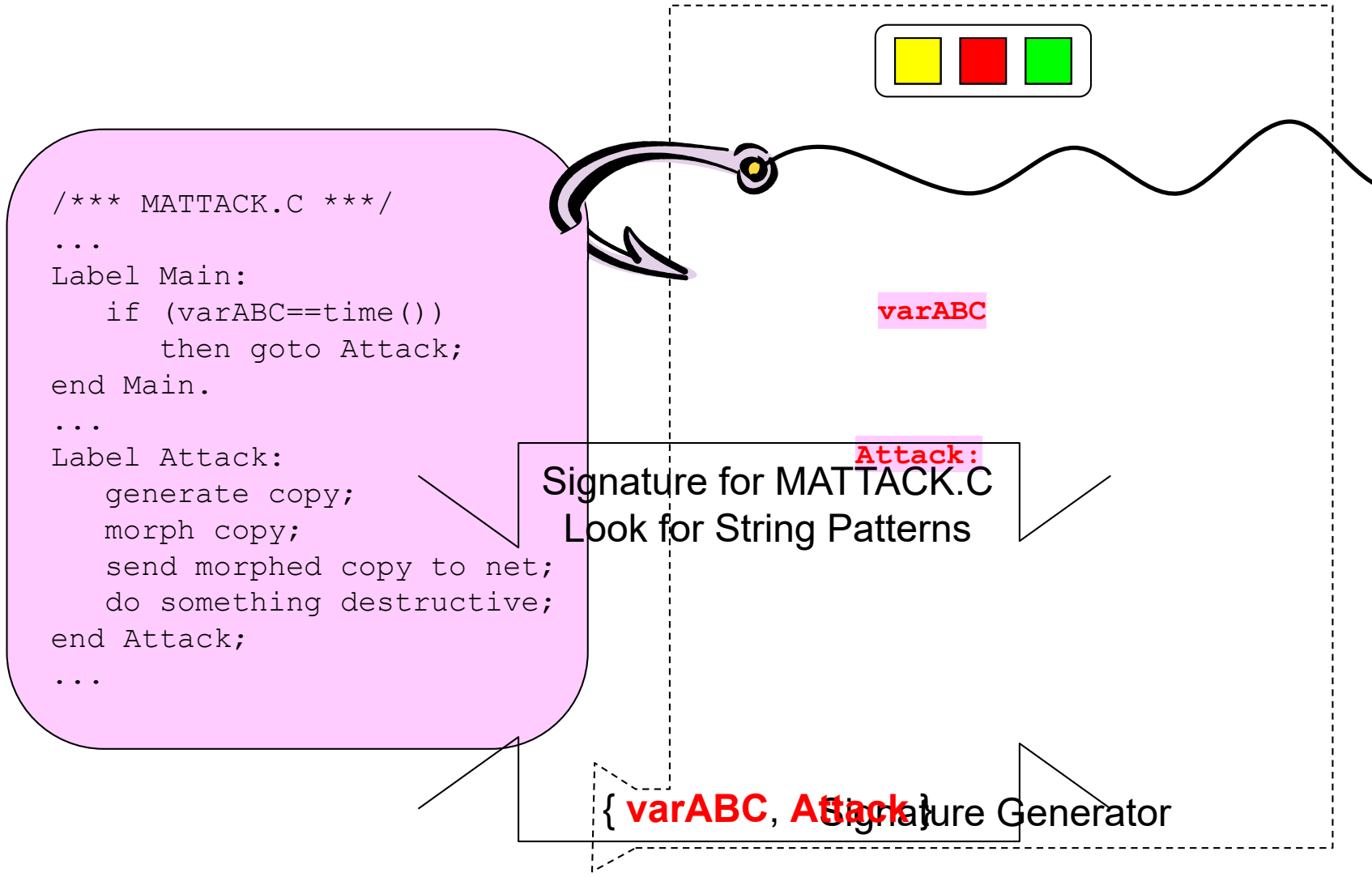
Example of Polymorphic Attack

```
/*** MATTACK.C ***/
...
Label Main:
   if (varABC==time())
      then goto Attack;
end Main.
...
Label Attack:
   generate copy;
   morph copy;
   send morphed copy to net;
   do something destructive;
end Attack;
...
```

```
/*** MATTACK_NEXTGEN.C ***/

      varXYZ
                Destroy;

    Destroy:


  Destroy;
```

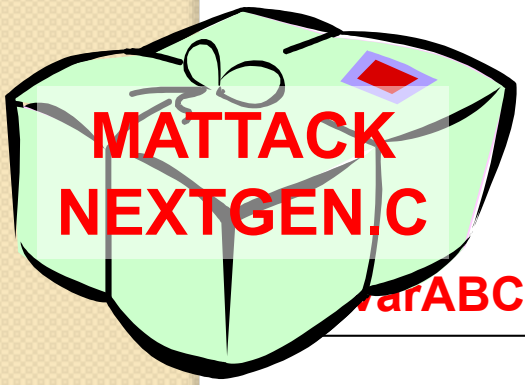
Example of Polymorphic Attack

```
/*** MATTACK.C ***/
...
Label Main:
   if (varABC==time())
      then goto Attack;
end Main.
...
Label Attack:
   generate copy;
   morph copy;
   send morphed copy to net;
   do something destructive;
end Attack;
...
```

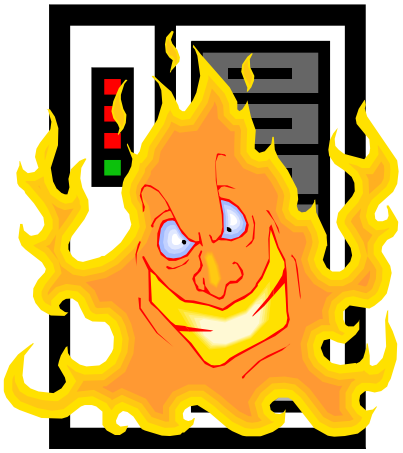


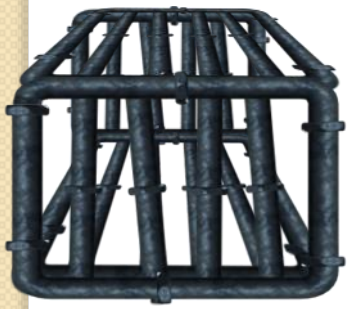
Example of Polymorphic Attack

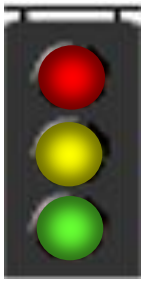
Signature for MATTACK.C
Look for String Patterns



{ **VarABC**, **Attack** }







Polymorphic Intrusion Detection

- Context Free Grammar
  - Language syntax
  - Most computer languages
  - Higher level of expression
- Leverage Existing Compiler technology
  - Lexical analysis
  - Syntactic analysis
- Critical Issue
  - Matter of expressing polymorphic attacks

Current Technology Trend

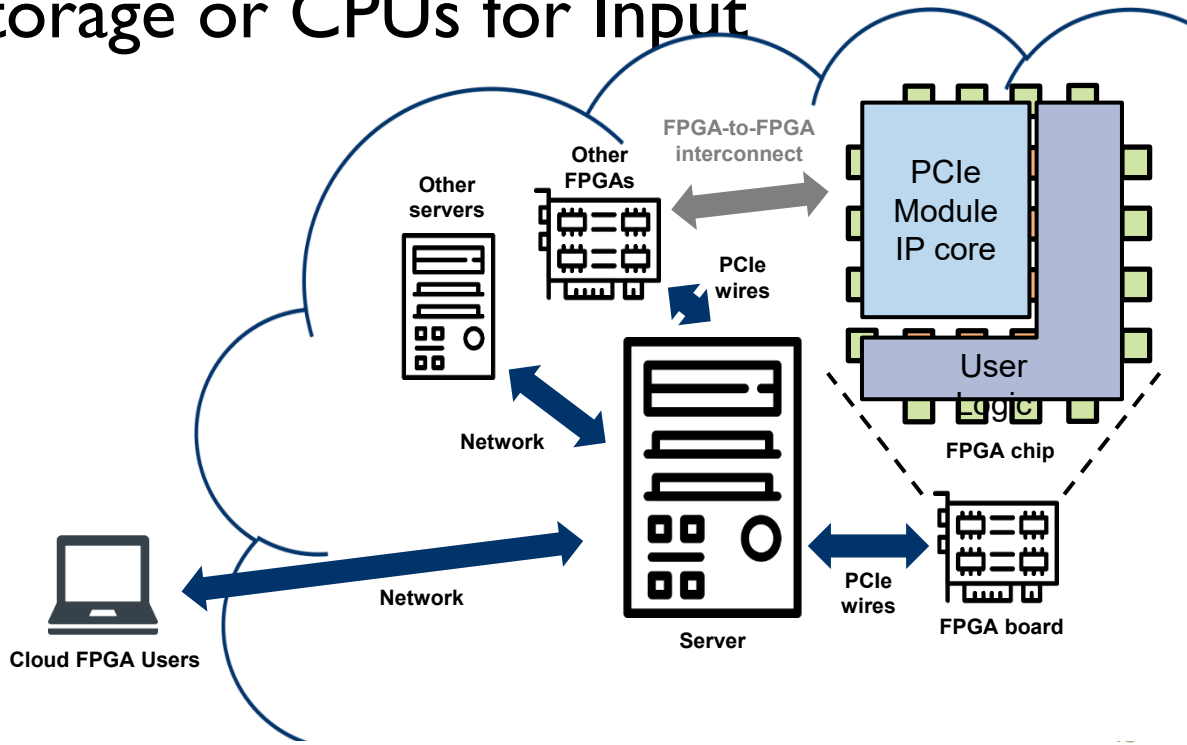
- It's ALL about the Cloud and the Internet
- Everything uses the Internet
  - Commoditized part of our lives
  - Knowledgebase
  - Commercial Hub
  - Expected to be available everywhere reliably
  - Growing Security Problems – because There's nearly NONE!
- All Bets are on the Cloud
  - Computing Offloaded from Personal Device to Cloud
  - Most Hardware and Software Developments are for the Cloud
  - Intel, AMD, and Nvidia are examples of HW Giants
  - Google, Microsoft, and Amazon are examples of SW Giants
  - New Network developments are for the Cloud

Back to Security Problem

- Security Problems of the Internet
- Cloud is connected to the Internet
- Cloud has Hardware Accelerators
  - GPU, FPGA, and Application Specific Infrastructures are Actively being Developed and Integrated

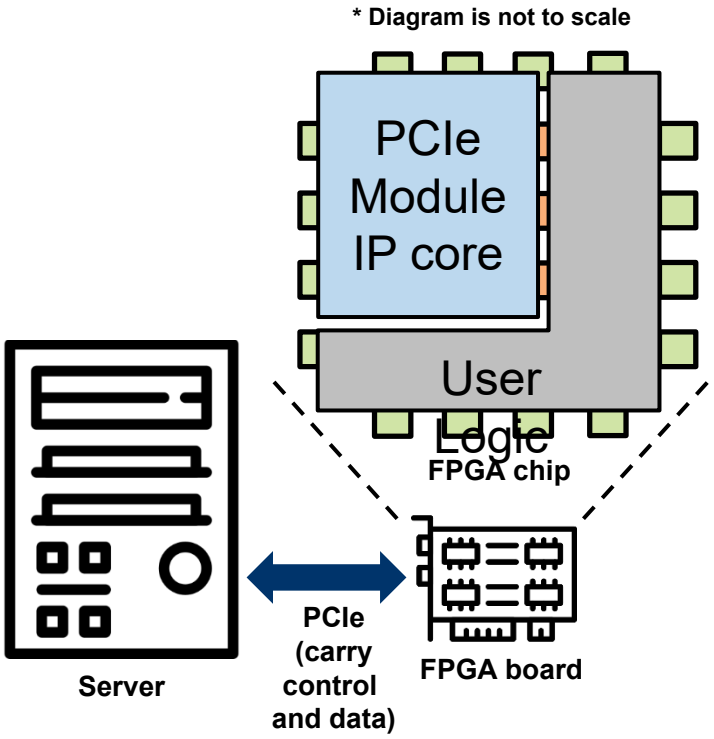
Cloud FPGA System

- FPGA with different modules
- One or more FPGAs per server
- Multiple servers with FPGAs
- Leverage storage or CPUs for Input



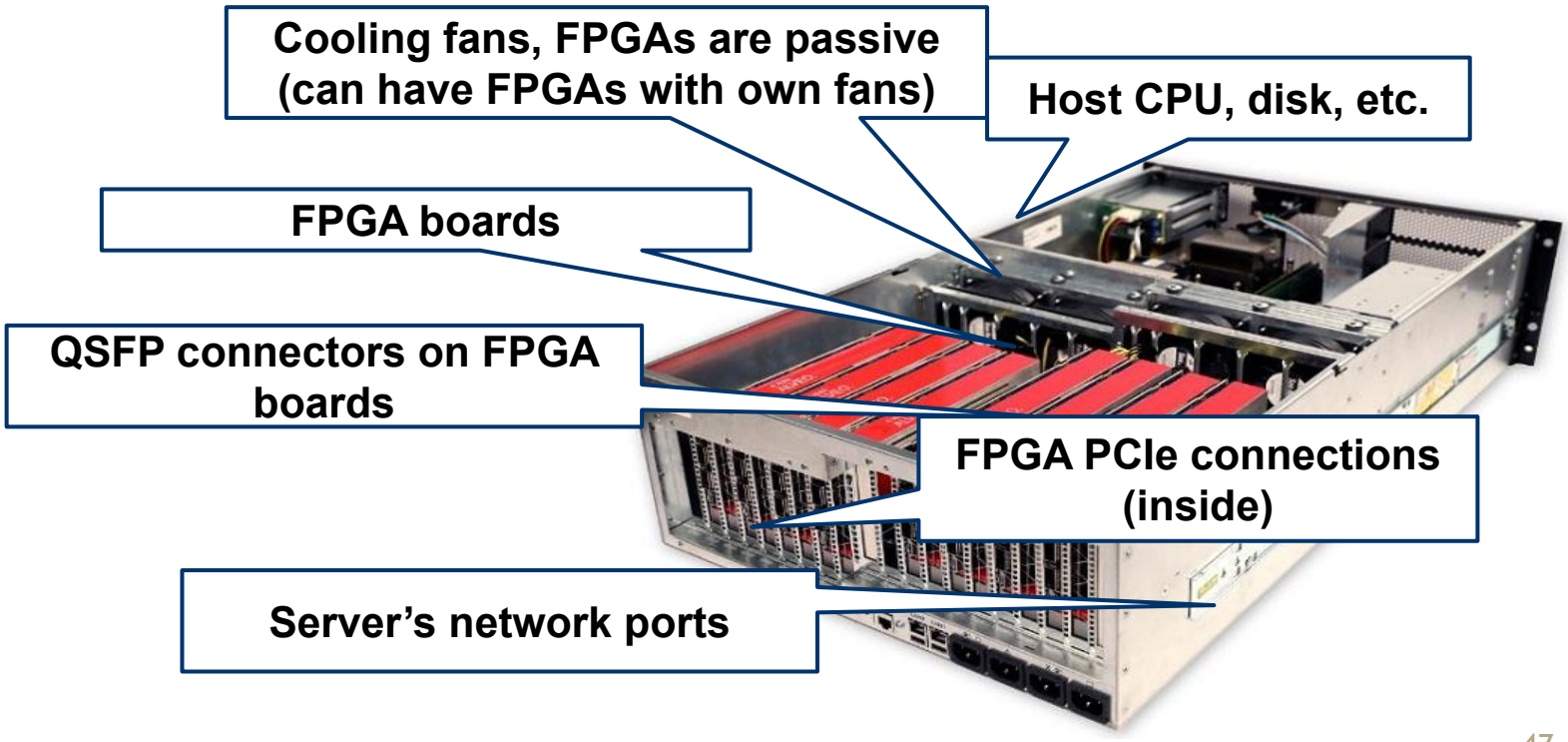
FPGA Interface

- FPGA Communication Logic
- Input and output to FPGA after processing
  - Many interface standards exist:
  - Serial port, USB
  - Ethernet, QSFP, QSFP+, SATA
  - PCI Express (PCIe)
  - Custom
- PCIe was the standard
- RDMA Taking Over



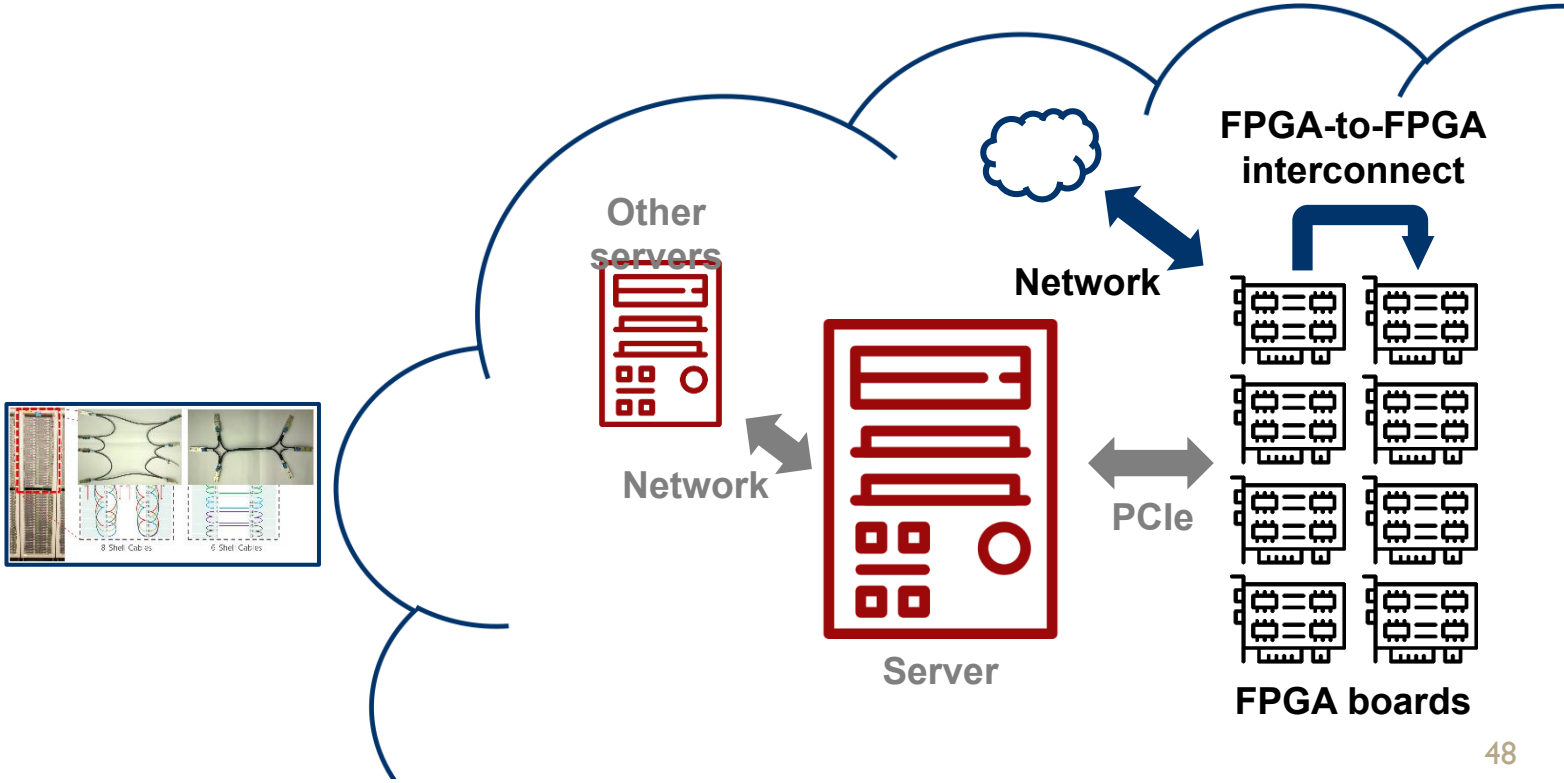
Server with AMD FPGAs

- Main CPU for Server Control
- Network interfaces for communication
- Multiple Xilinx Alveo U250 accelerator cards



FPGA Interconnections

- FPGA boards have QSFP connectors
- Some with Direct Connections



FPGAs used in Cloud

- Xilinx Virtex UltraScale+ FPGA VCU1525
  - Very similar to card used in **Amazon F1**
  - 2,500,000 logic cells
  - Thermal Design Power (TDP) of 225W
  - Up to PCIe 4.0 and DDR4 and QSFP networking

- Xilinx Alveo U200/U250/U280 Accelerator Cards
  - Likely cards for **Amazon F1 SDAccel**
  - 800,000 to 1,000,000 LUTs
  - Thermal Design Power (TDP) of 225W
  - Up to PCIe 4.0 and DDR4 and QSFP networking

- Catapult FPGA Accelerator Card (Microsoft + Intel FPGAs)
  - Altera Stratix V GS D5
  - 172,000 ALMs
  - PCIe 3.0 and DDR3

FPGA Cards

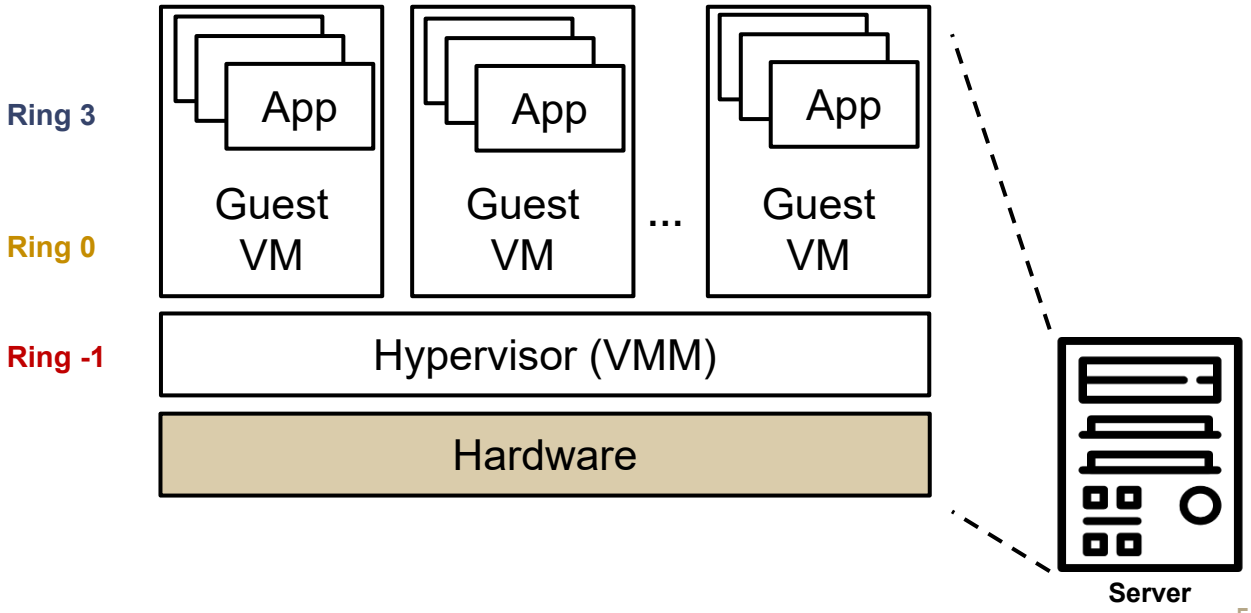
- Xilinx Alveo U50 Accelerator Card (2019)
  - Uses Xilinx's 16nm UltraScale+ FPGA archite
  - 800,000 LUTs
  - Thermal Design Power (TDP) of 75W
  - PCIe 4.0 and HBM2
  - QSFP networking

- Intel D5005 Programmable Acceleration Card (2019)
  - Uses Intel's 14nm Stratix 10 SX FPGA architecture
  - 2,800,000 logic elements
  - Thermal Design Power (TDP) of 215W
  - PCIe 3.0 and DDR4
  - QSFP networking

Images and information from [2] and [3]

Cloud FPGA Server Software

- Hypervisor, guest VMs, and applications running on the VMs
- Hypervisor is controlled by the cloud provider
- Users provide guest VM images
- Users provide applications and software that run in VMs
- VM images with FPGA development and programming



Guest VM and Libraries

- FPGA programming tools
  - Verilog, VHLD, SystemVerilog, etc.
  - High-Level Synthesis
  - E.g. Xilinx Vivado or Intel Quartus
- FPGA programming tools can be run: locally by user, on VM (without FPGA), on VM (with FPGA)
- Tools (often command line) for checking status, programming, clean up, etc.
- Libraries for programming languages (e.g., C or Python) for sending and receiving data from the FPGA
  - (slow) Read or write data word by word – user initiates each read or write
  - (faster) Bulk copy of data – copy data word by word, but under control of a library function
  - (fastest) Direct Memory Access – copy data in large chunks between DRAM and FPGA

Hardware Development Kits

- Create and compile a design for the FPGA
  - Design must include required parts such as PCIe, which is part of the *shell* in AWS terminology
  - Other parts are optional and up to the user, e.g. DRAM and user logic; all must pass design check rules
- The FPGAs are loaded with FPGA Images (i.e. AFIs),
  - Once loaded, user can interact with the hardware

Development Options

- Possible to Directly Program
  - But Too Difficult
  - CUDA is much easier because it is SW
- Prepackaged Development Solution
  - Needed to take over the market against GPU
  - Language of Choice seems to be OpenCL
- Learn to Use OpenCL
  - Used for GPU and other Accelerators
  - Including Nvidia Products
  - FPGA packages also use OpenCL