

# FPsed: A Streaming Content Search-and-Replace Module for an Internet Firewall

James Moscola, Michael Pachos  
John Lockwood, Ronald Loui  
August 21, 2003



Applied Research Laboratory  
Department of Computer Science and Engineering



Department of Computer Science and Engineering  
Applied Research Laboratory



Washington University in St. Louis  
SCHOOL OF ENGINEERING & APPLIED SCIENCE

## Outline

- Motivation
- Background
  - String Matching
  - Field Programmable Port Extender
- Packet Payload Search-and-Replace
- Automated Hardware Generation
  - Web interface
- Results
- Summary



Department of Computer Science and Engineering  
Applied Research Laboratory



Washington University in St. Louis  
SCHOOL OF ENGINEERING & APPLIED SCIENCE

## Motivation

- **Problem:**
  - Network speeds are continually increasing
    - Software packet processing techniques cannot keep up with the network
      - Software may ...
        - » Throttle the network to process everything
        - » Not process all packets
  - Need to be able to do deep packet processing at backbone speeds
- **Solution:**
  - Use custom FPGA hardware to process packets as they traverse across the network
    - Fast
    - Reconfigurable



## FPsed Module

- **Hardware module that executes a search-and-replace operation**
  - Filtering applications
  - Spam markup
  - Virus Removal
  - HTML/XML tag removal

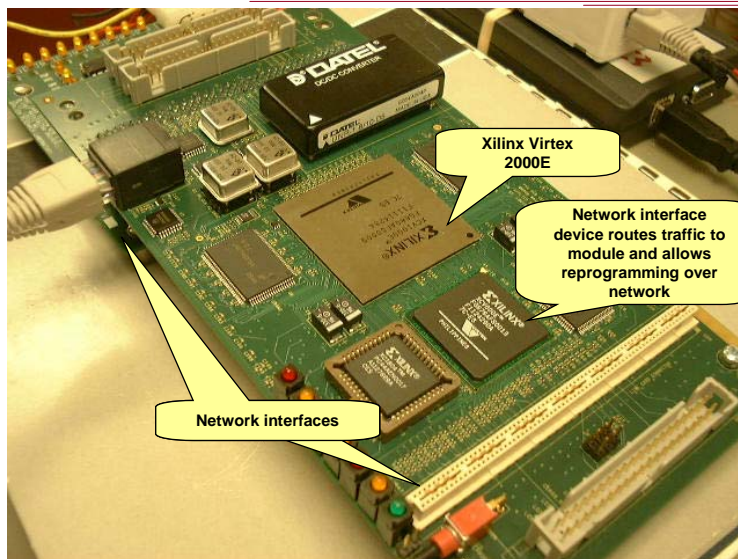


## Requirement for FPsed Module

- Need the ability to...
  - Scan every character of every packet's payload
    - To find regular expressions
  - Determine the boundaries of a given expression
    - 1<sup>st</sup> through last character
  - Replace any occurrence of a given expression
    - To remove a match from a packet payload
  - Easily reconfigure the scanner
    - To search for a new set of expressions

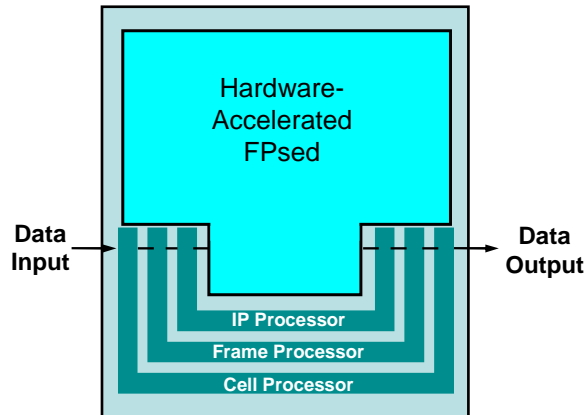


## Photograph of the FPX



## Module with Infrastructure

- **Processing Module**
  - Processes Data passing through the module
- **Protocol Wrappers**
  - Segment and reassemble Internet packets
  - Compute packet headers, lengths, and checksums
- **Interfaces**
  - Read and write packets to network



## Regular Expressions in Hardware

- **A couple different approaches ...**
  - Nondeterministic Finite Automata (NFAs)
    - Sidhu, Prasanna ; Franklin, Carver, Hutchings
    - Natural parallelism fits nicely into hardware
    - Easy construction
    - Small size
  - Deterministic Finite Automata (DFAs)
    - Theoretically large; but small in practice
    - One active state makes binary encoding possible
    - Compact state representation suitable for network
      - Context of a flow must be loaded / unloaded every packet.



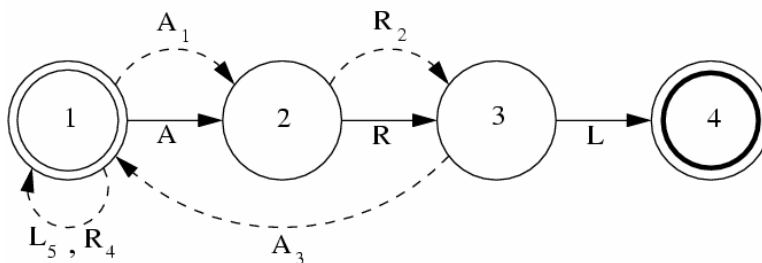
## Regular Expression Searching

- Need to detect the presence of an expression
- Do NOT need to know the boundaries of matching expressions
  - Prepend a “.” to the beginning of each string that we are looking for
- Search for “.\***ARL**” instead of “**ARL**”



## Regular Expression Searching

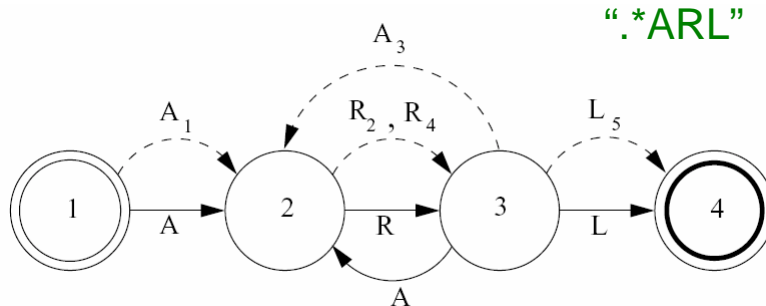
“ARL”



- Searching “ $A_1R_2A_3R_4L_5$ ” for “ARL” fails



## Regular Expression Searching



- Searching “ $A_1R_2A_3R_4L_5$ ” for “.\*ARL” succeeds



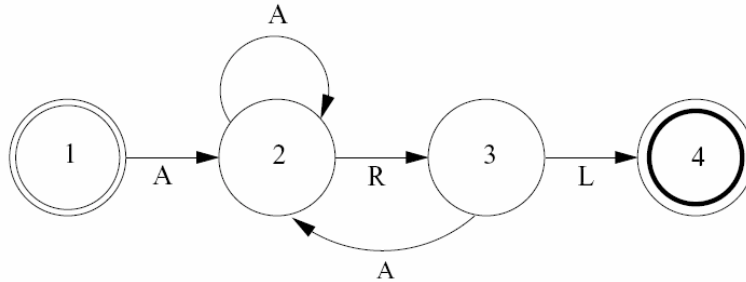
## Regular Expression Search-and-Replace

- Must determine beginning of matching substring
  - Cannot prepend a “.”
    - Difficult to determine the first character of a match
- Must determine end of longest complete matching substrings
  - Example:
    - Search for: “37F43(B+|7\*)”
    - Replace with: “Virus Pattern Detected”
    - Could replace up to where first “B” appears, but want to replace all



## Cannot Prepend a “.\*”

“.\*ARL”

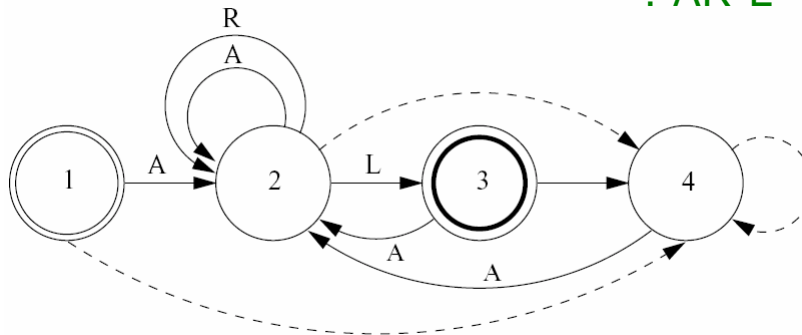


- If “.\*” is prepended, “AAARL” would be replaced when intention is to replace “ARL”
  - Fix: count characters after entering state 2



## Cannot Prepend a “.\*”

“.\*AR\*L”



- Fix: count characters after entering state 2 unless input character is “R”

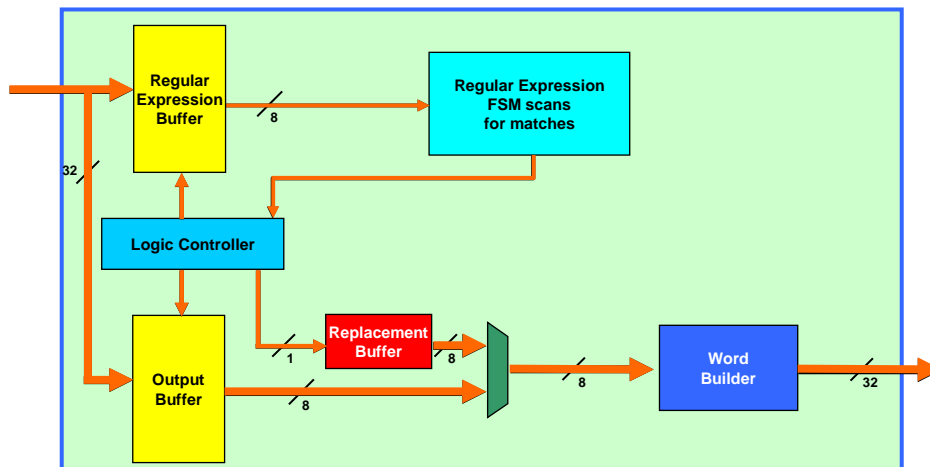


## Use Backtracking

- Every character is processed as possible start of matching substring
  - Start processing at character  $c_i$
  - Process expression until mismatch occurs
  - Continue processing at character  $c_{i+1}$
- Worst case running time:  $O(nm)$

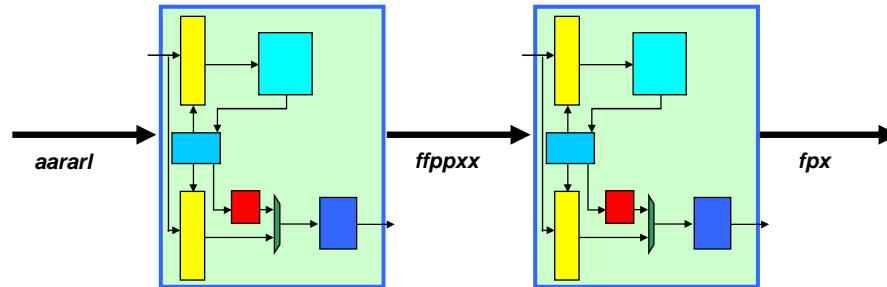


## FPsed Module



## Daisy-Chained FPsed Modules

**Specification:**  $s / a(ar)^* l / ffp\text{p}xx / g$   
 $s / f^*p+ x^* / fp\text{x} / g$

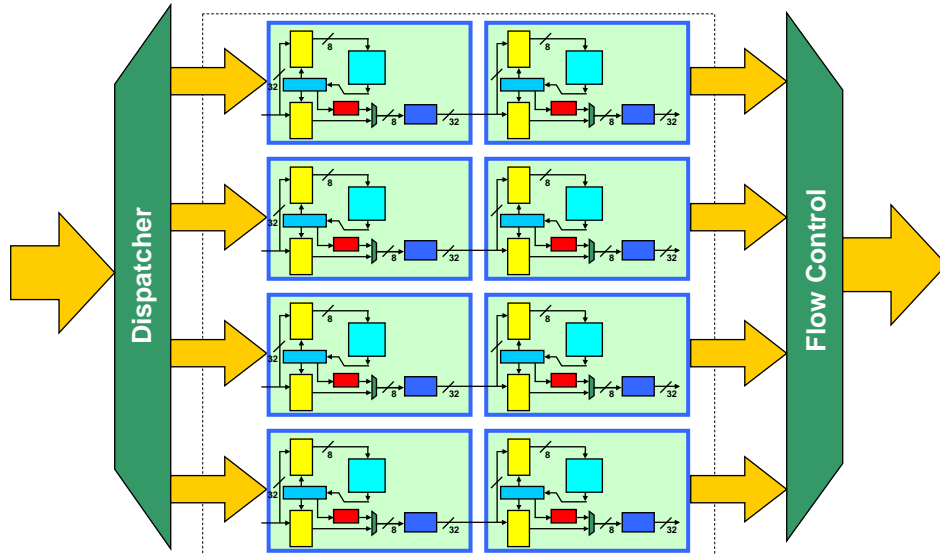


## Increased Throughput via Parallelism

- Processing packets at the full line rate
  - Problem:
    - Single scanner only processes 8 bits per clock cycle
    - Input rate is 32 bits per clock cycle
  - Solution:
    - Four parallel content scanners



## Increased Throughput via Parallelism



## Generating the Hardware

- FPsed requires new hardware for each set of expressions
- Easily reconfigurable
  - Command-line scripts
  - Web Interface
- Completely automated



## Generating the Hardware

- Read input specification
  - FPsed Syntax:
    - `s / expression / replacement string /`
  - Example:
    - `s / <[^>]*> / /`
- Parse with JLex
  - Create optimized DFA
- Generate VHDL
  - Convert JLex DFA to VHDL
  - Create replacement buffers for all replacements
  - Create structural component to connect all DFAs and replacement buffers
- Synthesize and place and route
- Dynamically reprogram the FPGA on the FPX

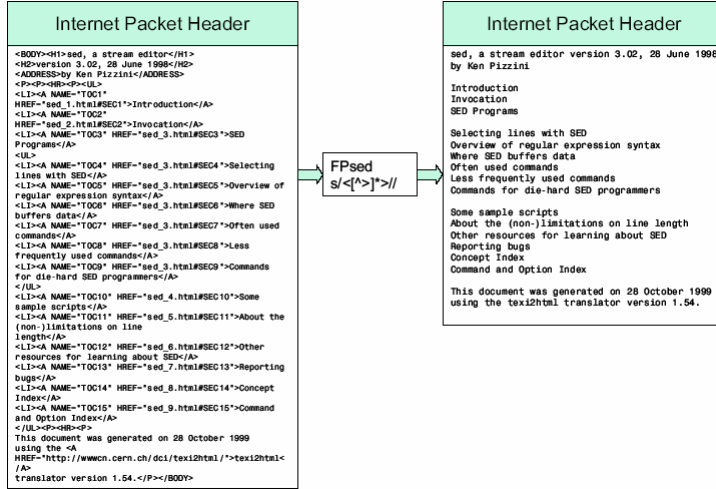


## FPsed Results

-HTML Filter



# HTML Filter



# FPsed Device Utilization



## Infrastructure and Protocol Wrappers

Resources	Virtex XCV2000E Device Utilization	Utilization Percentage
Logic Slices	<b>2410 of 19200</b>	<b>12%</b>
Flip Flops	<b>2870 of 38400</b>	<b>7%</b>
Block RAMS	<b>19 of 160</b>	<b>11%</b>
External IOBs	<b>142 of 512</b>	<b>27%</b>

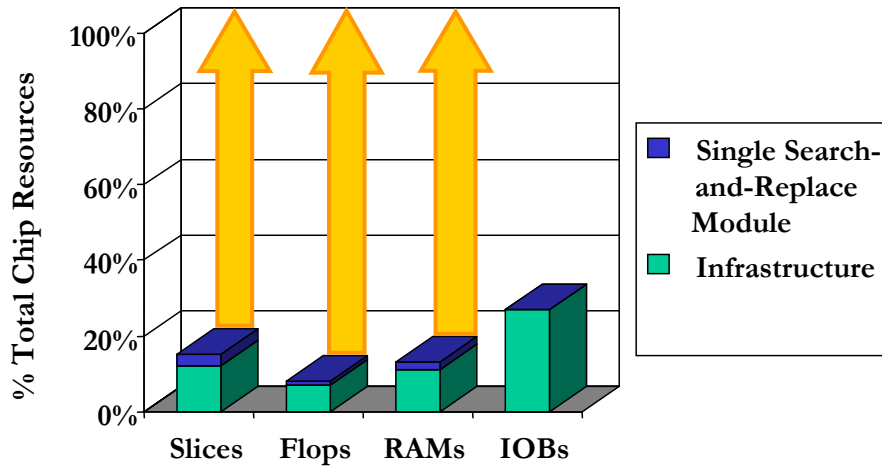


## FPsed with Single Module

Resources	Virtex XCV2000E Device Utilization	Utilization Percentage
Logic Slices	<b>2922 of 19200</b>	<b>12+3=15%</b>
Flip Flops	<b>3223 of 38400</b>	<b>7+1=8%</b>
Block RAMS	<b>21 of 160</b>	<b>11+2=13%</b>
External IOBs	<b>142 of 512</b>	<b>27%</b>



## Device Utilization – HTML Filter

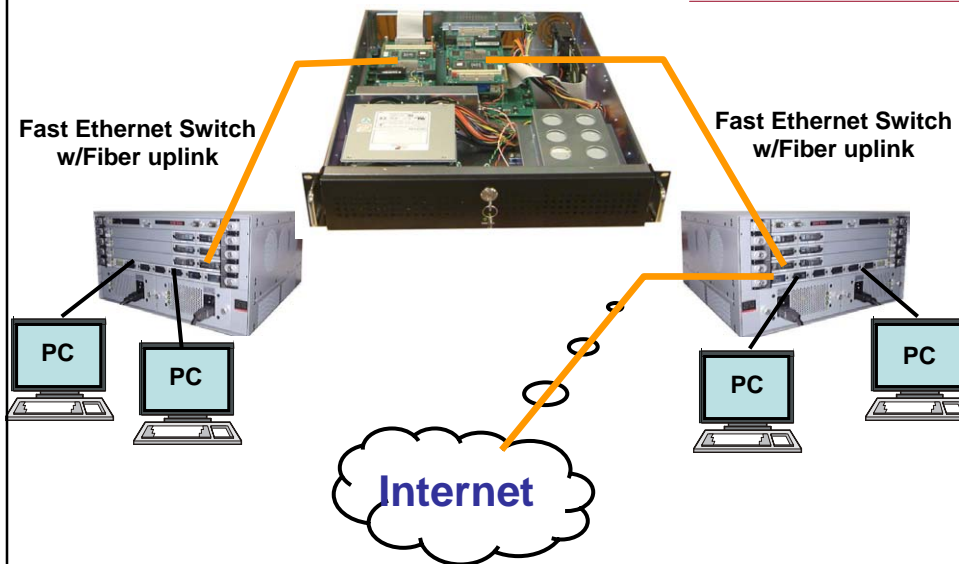


Department of Computer Science and Engineering  
Applied Research Laboratory



Washington University in St. Louis  
SCHOOL OF ENGINEERING & APPLIED SCIENCE

## Test Environment



Department of Computer Science and Engineering  
Applied Research Laboratory



Washington University in St. Louis  
SCHOOL OF ENGINEERING & APPLIED SCIENCE

## Throughput

- FPsed processes 8-bits per clock cycle
  - HTML filter
    - $8 \text{ bits} * 64 \text{ MHz} = 512 \text{ Mbps}$
- Four parallel engines increased throughput
  - HTML filter
    - $4 \text{ Modules} * 8 \text{ bits} * 64 \text{ MHz} = 2.048 \text{ Gbps}$

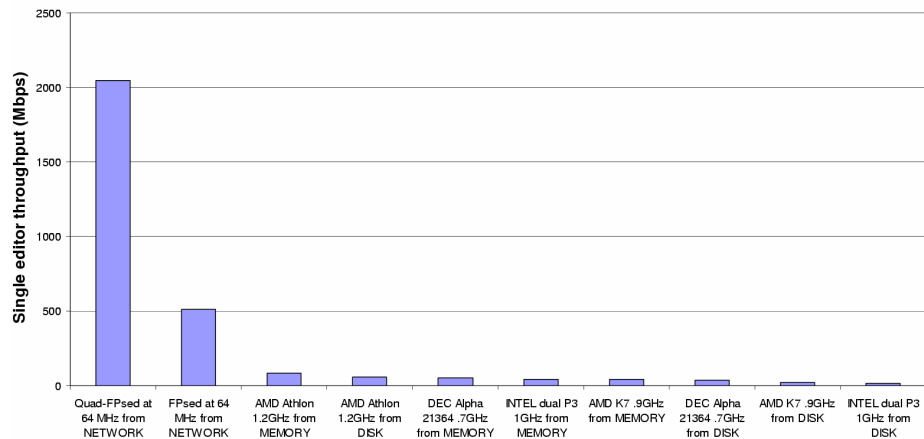


## Software Throughput Comparison

- Sed was run on different Linux PCs
  - Dual Intel Pentium III @ 1 GHz
    - 13.7 Mbps when data is read from disk
    - 32.72 Mbps when data is read from memory
  - Alpha 21364 @ 667 MHz
    - 36 Mbps when data is read from disk
    - 50.4 Mbps when data is read from memory
- PC results are 10x slower than FPsed



## Software Throughput Comparison



## Summary

- Implemented FPsed Modules on the FPX
  - Enable full processing of packet payloads
  - Modules can integrate into a firewall to add functionality
  - Determines boundaries of regular expressions
  - Actively replaces matching expressions with a predefined string
- Design Flow has been created
  - Automatically generates bitfiles
- Modules have been tested in lab
  - Operates with UDP traffic
  - Module operates at speeds of 512 Mbps – 2 Gbps



## Acknowledgements



- Washington University
  - **Faculty**
    - John Lockwood
    - Ron Cytron
    - Ronald Loui
    - Jon Turner
  - **Graduate Students**
    - David Taylor
    - Todd Sproull
    - Sarang Dharmapurikar
    - David Lim
    - David Schuehler
    - Chris Neely
    - Chris Zuver
    - Haoyu Song
    - Henry Fu (Now at Stanford)
    - Bharath Madhusudan
  - **Undergraduate Students**
    - Harvey Ku (at CMU)
    - Eliot Sinclair
    - Mike Attig
    - Doug Stirrut
    - Tucker Evans (Now at General Dynamics)
    - Mike Wrighton (Now at CalTech)
- Industry Research Partners
  - David Parlour (Xilinx)
  - Matthew Kulig (Global Velocity)
- University Research Partners
  - Prabhu Kuttiam (University of Kentucky)
  - Ken Calvert (University of Kentucky)
  - Matt Sanders (Georgia Tech)
  - Ron Srodawa (Oakland University)
  - Haiyan Qiao (NDSU)
  - William Perrizo (NDSU)
  - Kuo-Tung Kuo (University of Maryland)
  - Cary Colwell (Naval Postgraduate School)
  - John Gibson (Naval Postgraduate School)
  - Huaiyu Liu (University of Texas at Austin)
  - Qing Tan (University of Toledo)
  - Sachin Shetty (University of Toledo)
  - Rajanikanth Batchu (Mississippi State)
  - Ravi Sankar (USF)
  - Simon Wong (UCLA)
  - Sven Shepstone (University of Cape Town, South Africa)
- Visiting Faculty and Students
  - Edson Horta (Univ. de Sao Paulo, Brazil)
  - Florian Braun (University of Stuttgart)
  - Carlos Macian (University of Stuttgart)



## More Information

- <http://www.arl.wustl.edu/arl/projects/fpx/>
- <http://www.globalvelocity.info>

