High-Speed Computing & Co-Processing with FPGAs

FPGAs (Field Programmable Gate Arrays) are slowly becoming more and more advanced and practical as high-speed computing platforms. In this talk, David will provide an in-depth introduction into the guts and capabilities of modern day FPGAs and show how you can take your current algorithms and efficiently convert them to gate logic and run them on hardware. This presentation will also introduce a set of open source cores (jawn v1.0) that will implement the basic functionality of john the ripper on FPGAs and allow you to crack password hashes as fast as 100+ PCs using FPGA PCMCIA cards on your laptop.

David Hulton <dhulton@picocomputing.com>

Founder, Dachb0den Labs
Chairman, ToorCon Information Security Conference
Embedded Systems Engineer, Pico Computing, Inc.

Disclaimer

- Educational purposes only
- Full disclosure
- I'm not a hardware guy

Goals

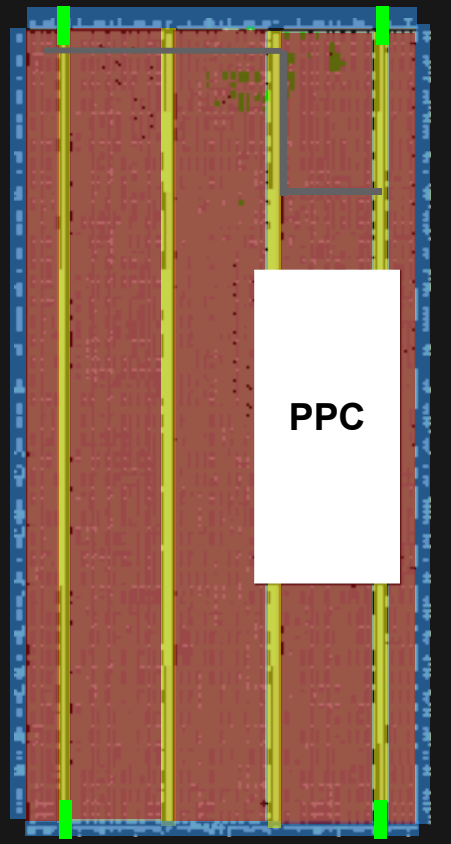
- This talk will cover:
  - Introduction to FPGAs
    - Verilog
    - Optimization Concepts
  - Cryptography
    - History
    - Password File Cracker (jawn v0.1)
  - Artificial Intelligence
    - Neural Networks

Introduction to FPGAs

- Field Programmable Gate Array
  - Lets you prototype IC's
  - Code translates directly into circuit logic

Introduction to FPGAs

- Configurable Logic Blocks (CLBs)
  - Registers (flip flops) for fast data storage
  - Logic Routing
- Input/Output Blocks (IOBs)
  - Basic pin logic (flip flops, muxs, etc)
- Block Ram
  - Internal memory for data storage
- Digial Clock Managers (DCMs)
  - Clock distribution
- Programmable Routing Matrix
  - Intelligently connects all components together



FPGA Pros / Cons

- Pros
  - Common Hardware Benefits
    - Massively parallel
    - Pipelineable
  - Reprogrammable
    - Self-reconfiguration
- Cons
  - Size constraints / limitations
  - More difficult to code & debug

Introduction to FPGAs

- Common Applications
  - Encryption / decryption
  - AI / Neural networks
  - Digital signal processing (DSP)
  - Software radio
  - Image processing
  - Communications protocol decoding
  - Matlab / Simulink code acceleration
  - Etc.

Introduction to FPGAs

- Common Applications
  - Encryption / decryption
  - AI / Neural networks
  - Digital signal processing (DSP)
  - Software radio
  - Image processing
  - Communications protocol decoding
  - Matlab / Simulink code acceleration
  - Etc.

Types of FPGAs

- Antifuse
  - Programmable only once
- Flash
  - Programmable many times
- SRAM
  - Programmable dynamically
  - Most common technology
  - Requires a loader (doesn't keep state after power-off)

Development Platform

- ROAG
  - PCMCIA Form Factor
  - Virtex II-Pro (XC2VP4-5)
  - Embedded PowerPC 405
  - 128MB RAM
  - 32MB Flash
  - 10/100 Ethernet
  - Synchronous Serial Port
  - 2 RS232 Ports
  - CANBus
  - Satellite Radio Controller

Development Platform

- Virtex II-Pro (XC2VP4-5)
  - 6,768 Logic Cells
    - 12KB of Registers (Distributed RAM)
    - ~ 180,000 Gates
  - 64KB of Block RAM
  - PowerPC 405
    - 300mhz Max Clock Speed

Development Platform

- FPGA Programming
  - PCMCIA
  - JTAG
- Embedded System
  - Xilinx's Microkernel
  - Linux
  - OpenBSD / NetBSD / etc ?

Creating Your Project

- Tools
  - ISE 6.3i
  - Chipscope 6.3i
  - Modelsim 5.8c
  - EDK 6.3i
- Installation date + 60-day trials available on xilinx.com

Verilog

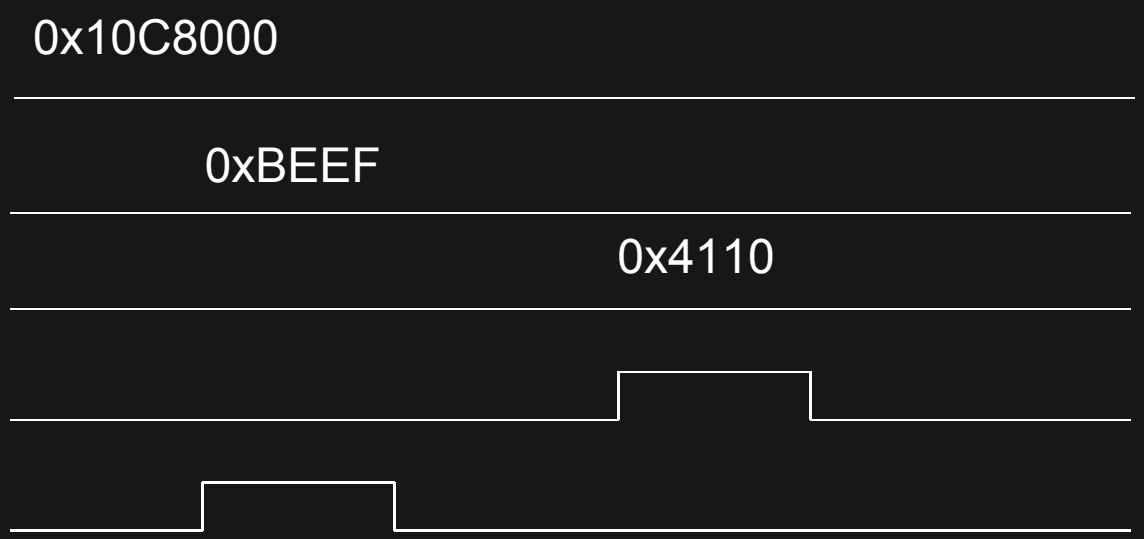
- Hardware Description Language
- Simple C-like Syntax
- Like Go - Easy to learn, difficult to master

Demonstration

- Interfacing with the PCMCIA bus
- Creating your design
- Building
- Running

PCMCIA Bus

- Lines
  - Address
  - Data In
  - Data Out
  - Read
  - Write



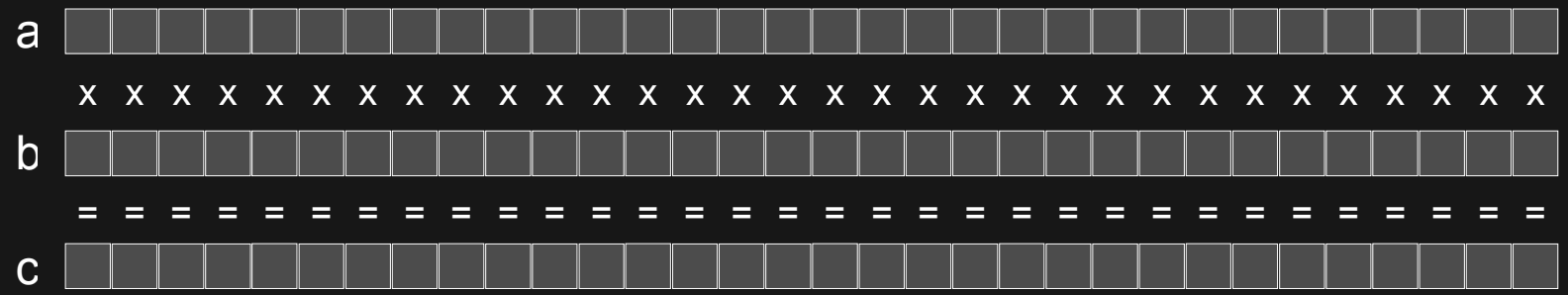
- Example
  - Read in input from PCMCIA bus
  - Invert bits and return it

Massively Parallel Example

- PC (32 * ~ 7 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < 32; i++)
    c[i] = a[i] * b[i];
```

- Hardware (1 clock cycle) @ 300Mhz



Massively Parallel Example

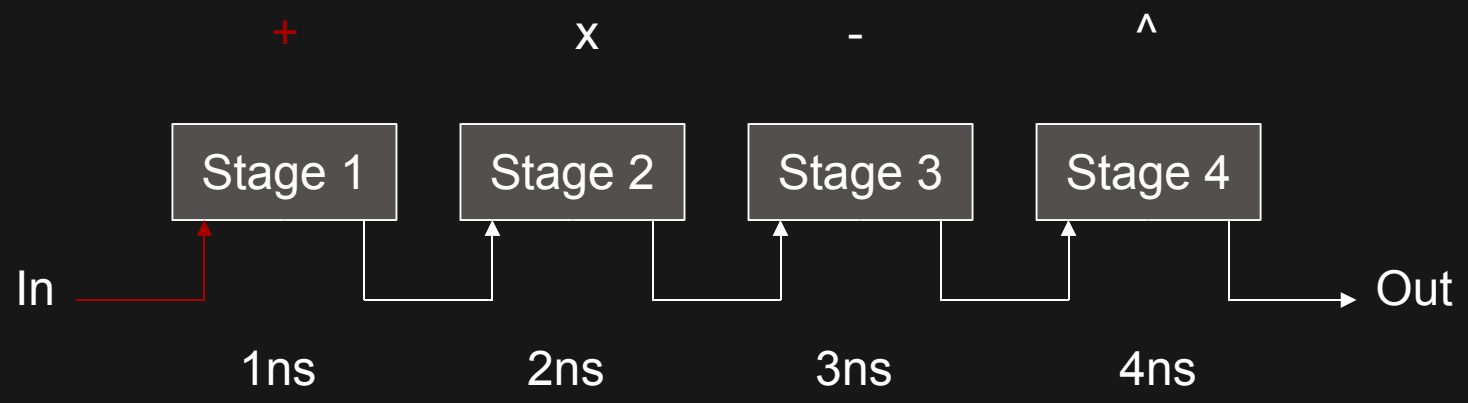
- PC
  - Speed scales with # of instructions & clock speed
- Hardware
  - Speed scales with FPGA's:
    - Size
    - Clock Speed

Pipeline Example

- PC (x * ~ 10 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < x; i++)
    f[i] = a[i] + b[i] * c[i] – d[i] ^ e[i]
```

- Hardware (x + 3 clock cycles) @ 300Mhz

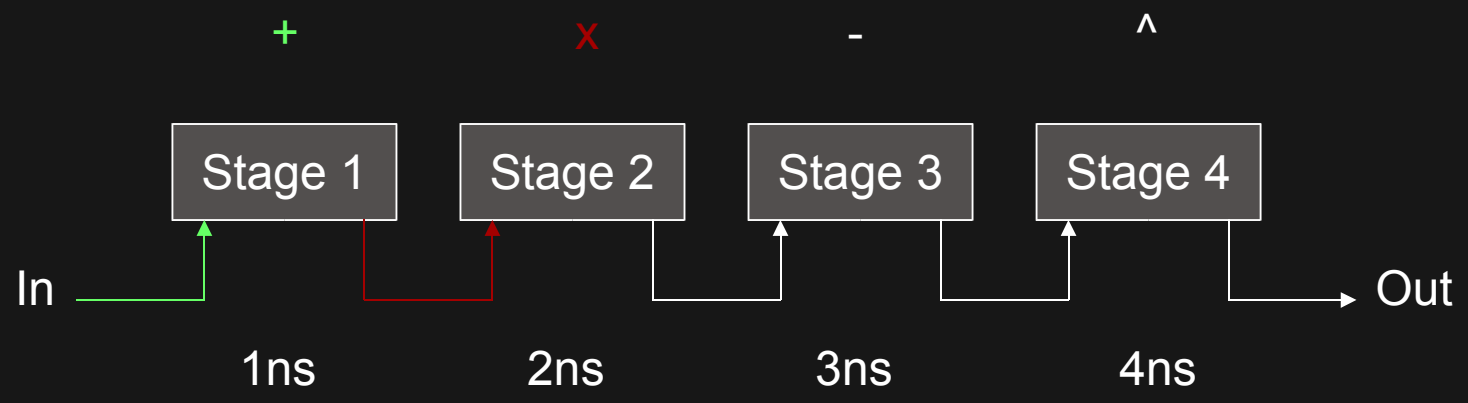


Pipeline Example

- PC (x * ~ 10 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < x; i++)
        f[i] = a[i] + b[i] * c[i] – d[i] ^ e[i]
```

- Hardware (x + 3 clock cycles) @ 300Mhz

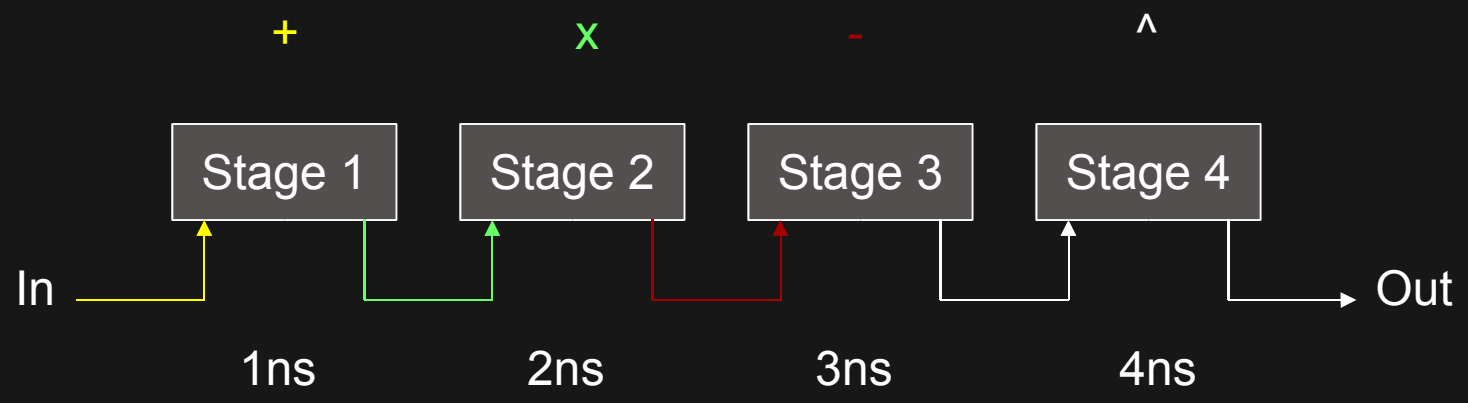


Pipeline Example

- PC (x * ~ 10 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < x; i++)
    f[i] = a[i] + b[i] * c[i] – d[i] ^ e[i]
```

- Hardware (x + 3 clock cycles) @ 300Mhz

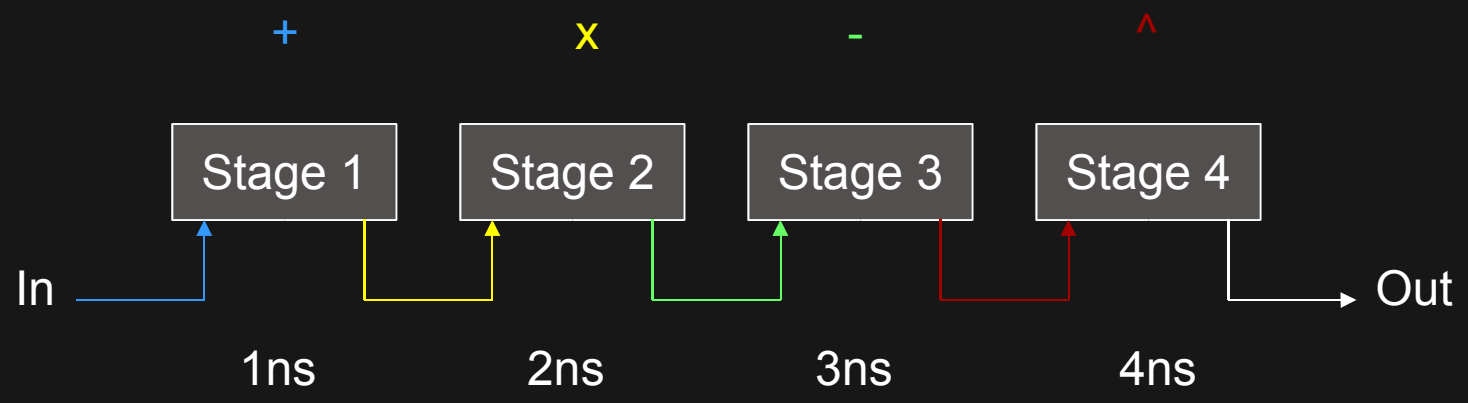


Pipeline Example

- PC (x * ~ 10 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < x; i++)
    f[i] = a[i] + b[i] * c[i] – d[i] ^ e[i]
```

- Hardware (x + 3 clock cycles) @ 300Mhz

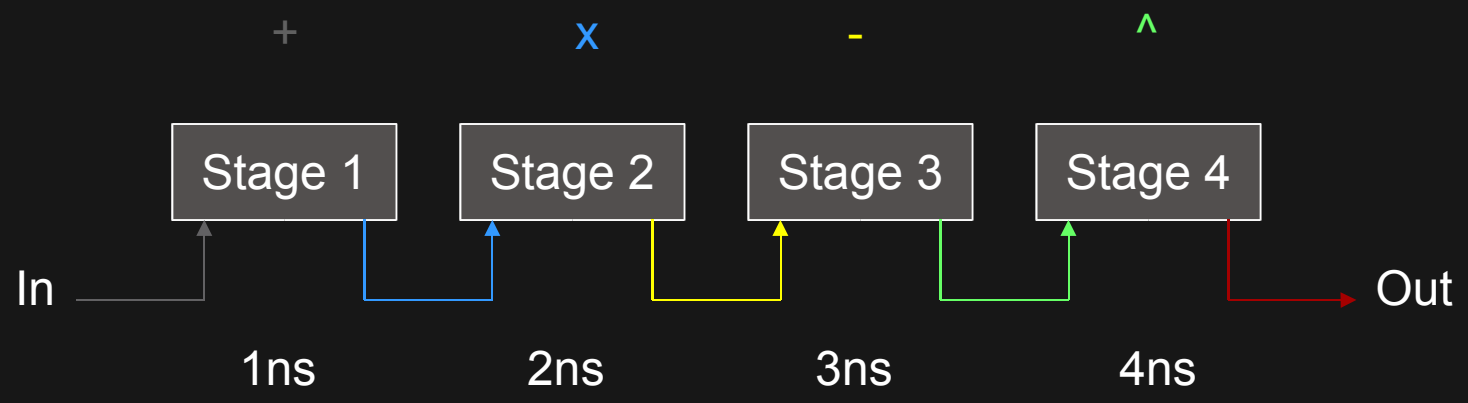


Pipeline Example

- PC (x * ~ 10 clock cycles ?) @ 3.0Ghz

```
for(i = 0; i < x; i++)
    f[i] = a[i] + b[i] * c[i] – d[i] ^ e[i]
```

- Hardware (x + 3 clock cycles) @ 300Mhz



Pipeline Example

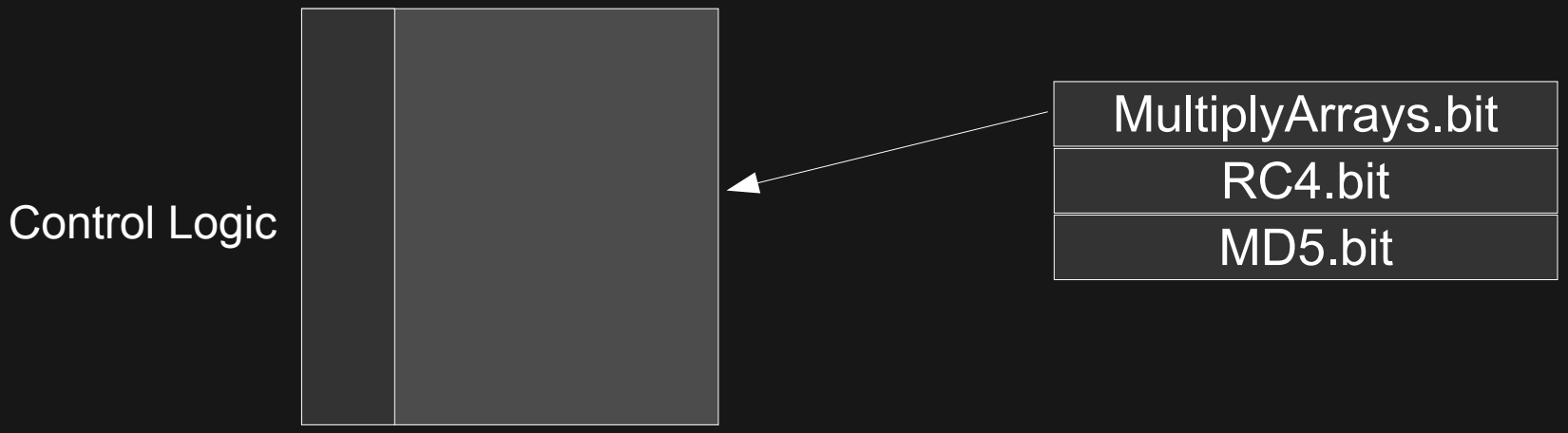
- PC
  - Speed scales with # of instructions & clock speed
- Hardware
  - Speed scales with FPGA's:
    - Size
    - Clock speed
    - Slowest operation in the pipeline

Self-Reconfiguration Example

- PC

```
data = MultiplyArrays(a, b);
RC4(key, data, len);
m = MD5(data, len);
```

- Hardware

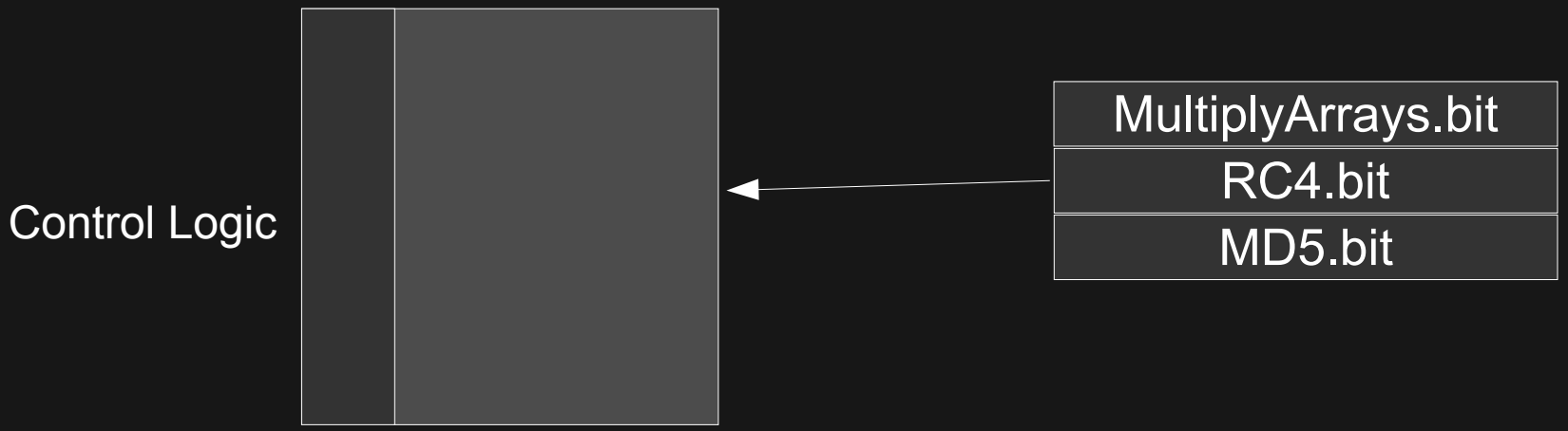


Self-Reconfiguration Example

- PC

```
data = MultiplyArrays(a, b);
RC4(key, data, len);
m = MD5(data, len);
```

- Hardware

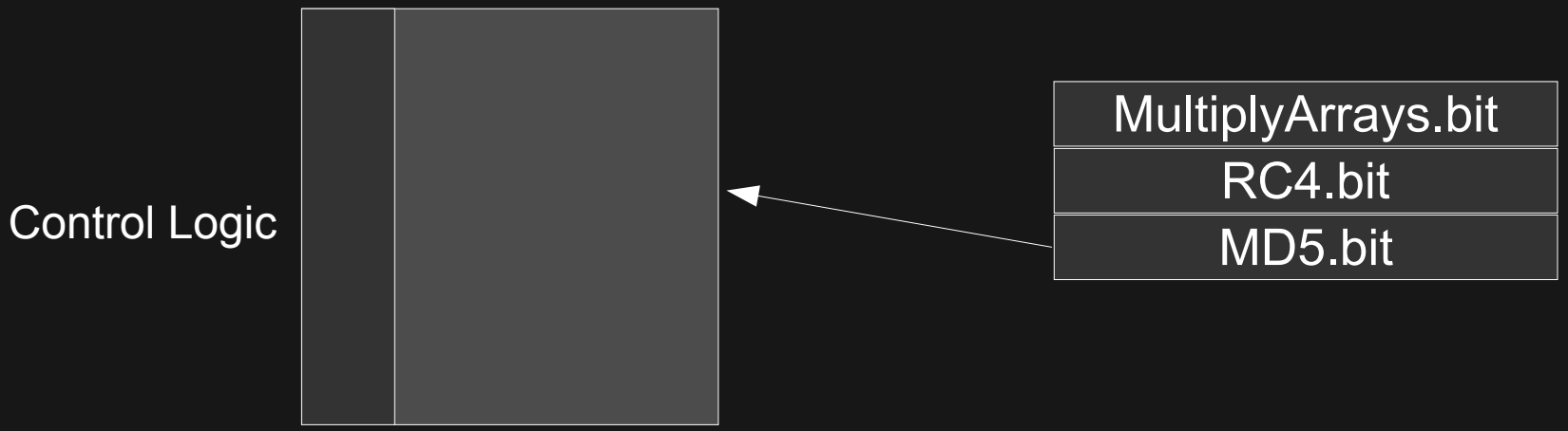


Self-Reconfiguration Example

- PC

```
data = MultiplyArrays(a, b);
RC4(key, data, len);
m = MD5(data, len);
```

- Hardware



History of FPGAs and Cryptography

- Minimal Key Lengths for Symmetric Ciphers
  - Ronald L. Rivest (R in RSA)
  - Bruce Schneier (Blowfish, Twofish, etc)
  - Tsutomu Shimomura (Mitnick)
  - A bunch of other ad hoc cypherpunks

History of FPGAs and Cryptography

Budget	Tool	40-bits	56-bits	Recom
Pedestrian Hacker				
Tiny	Computers	1 week	infeasible	45
$400	FPGA	5 hours	38 years	50
Small Company				
$10K	FPGA	12 min	556 days	55
Corporate Department				
$300K	FPGA	24 sec	19 days	60
	ASIC	0.18 sec	3 hrs	
Big Company				
$10M	FPGA	0.7 sec	13 hrs	70
	ASIC	0.005 sec	6 min	
Intelligence Agency				
$300M	ASIC	0.0002 sec	12 sec	75

History of FPGAs and Cryptography

- 40-bit SSL is crackable by almost anyone
- 56-bit DES is crackable by companies
- Scared yet?

This paper was published in 1996

History of FPGAs and Cryptography

- 1998
  - The Electronic Frontier Foundation (EFF)
  - Cracked DES in < 3 days
  - Searched ~9,000,000,000 keys/second
  - Cost < $250,000
- 2001
  - Richard Clayton & Mike Bond (University of Cambridge)
  - Cracked DES on IBM ATMs
  - Able to export all the DES and 3DES keys in ~ 20 minutes
  - Cost < $1,000 using an FPGA evaluation board

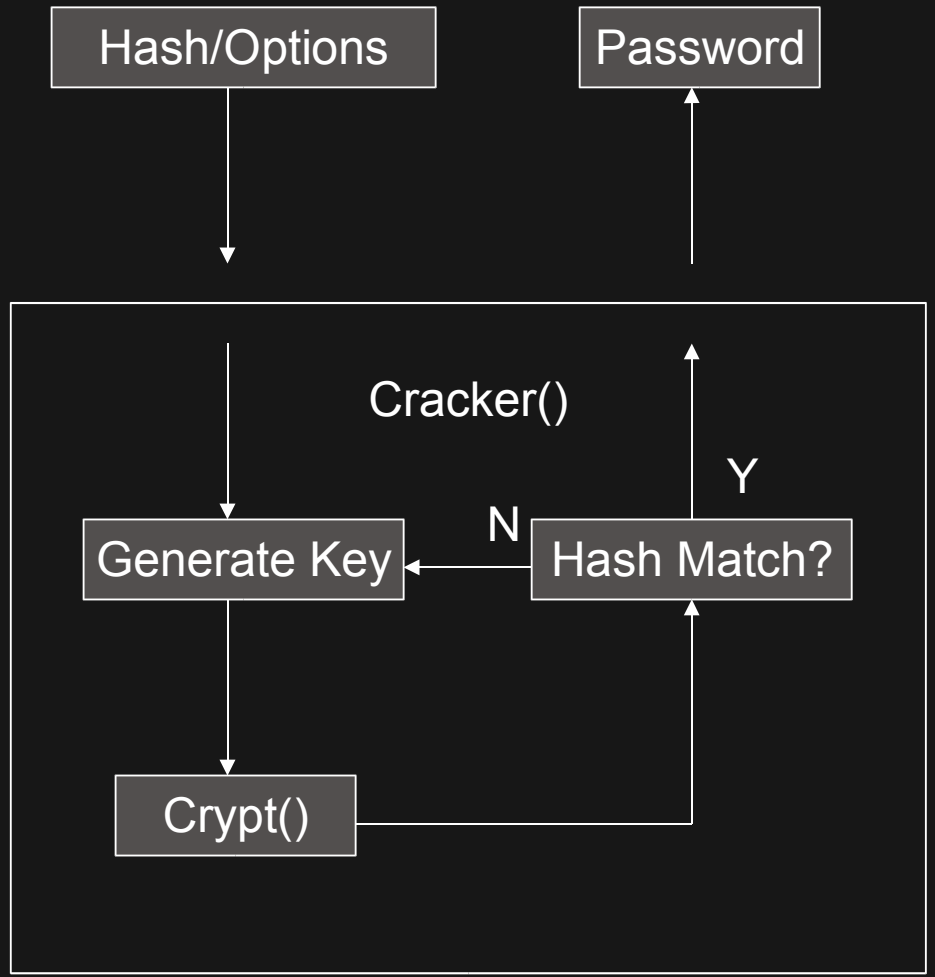
History of FPGAs and Cryptography

- 2004
  - Philip Leong, Chinese University of Hong Kong
  - IDEA
    - 50Mb/sec on a P4 vs. 5,247Mb/sec on Pilchard
  - RC4
    - Cracked RC4 keys 58x faster than a P4
    - Parallelized 96 times on a FPGA
    - Cracks 40-bit keys in 50 hours
    - Cost < $1,000 using a RAM FPGA (Pilchard)

Password File Cracker

- Design
  - Pipeline design
  - Internal cracking engine
    - password = des_crack(hash, options);
  - Interface over PCMCIA
  - Can specify cracking options
    - Bits to search
      - e.g. Search 55-bits (instead of 56)
    - Offset to start search
      - e.g. First card gets offset 0, second card gets offset 2**55
    - Typeable/printable characters
    - Alpha-numeric
    - Allows for basic distributed cracking & resume functionality

Password File Cracker



Password File Cracker

- PC (3.0Ghz P4 \w john)
  - ~ 300,000 c/s
- Hardware (Low end FPGA \w jawn)
  - 100Mhz/25 = ~4,000,000 c/s
  - When timing issues are resolved it should run at 200Mhz

Type	P4	ROAG	8 ROAGs
56-bits	3808 Y	292 Y	36 Y
Typeable / printable	381 Y	28 Y	3.5 Y
Alpha-numeric	14 Y	1.1 Y	50 D

Up & Coming

- Pico (PCMCIA)
  - 20k CLBs (~ 600k gates) @ ~ 350Mhz
  - (3x250Mhz)/25 = ~30m c/s
- Picomon (Compact Flash)
  - 30k CLBs (~ 1m gates) @ ~ 400Mhz
  - (5x300Mhz)/25 = ~60m c/s
- Nest (PCI)
  - 16 Picomons
  - 480k CLBs (~ 16m gates) @ ~ 400Mhz
  - (16x5x300Mhz)/25 = ~960m c/s
  - NOTE: Straight DES cracking is ~ 24b c/s (> 2.5x faster than the EFF DES cracker)

Up & Coming Real Performance

Type	Pico	Picomon	Nest
56-bits	36Y	19Y	1.2Y
Typeable / printable	3.8Y	1.9Y	43D
Alphanumeric	54D	27D	41H
Straight DES	1.5Y	277D	17.4D

Artificial Intelligence

- Back Propegation Neural Network
- Applications
  - Handwriting Recognition
  - Character Recognition
  - Voice Recognition
  - FFTs
  - Automatic Protocol Emulation
  - Pattern Matching
  - Etc.

BP Neural Networks

- Running

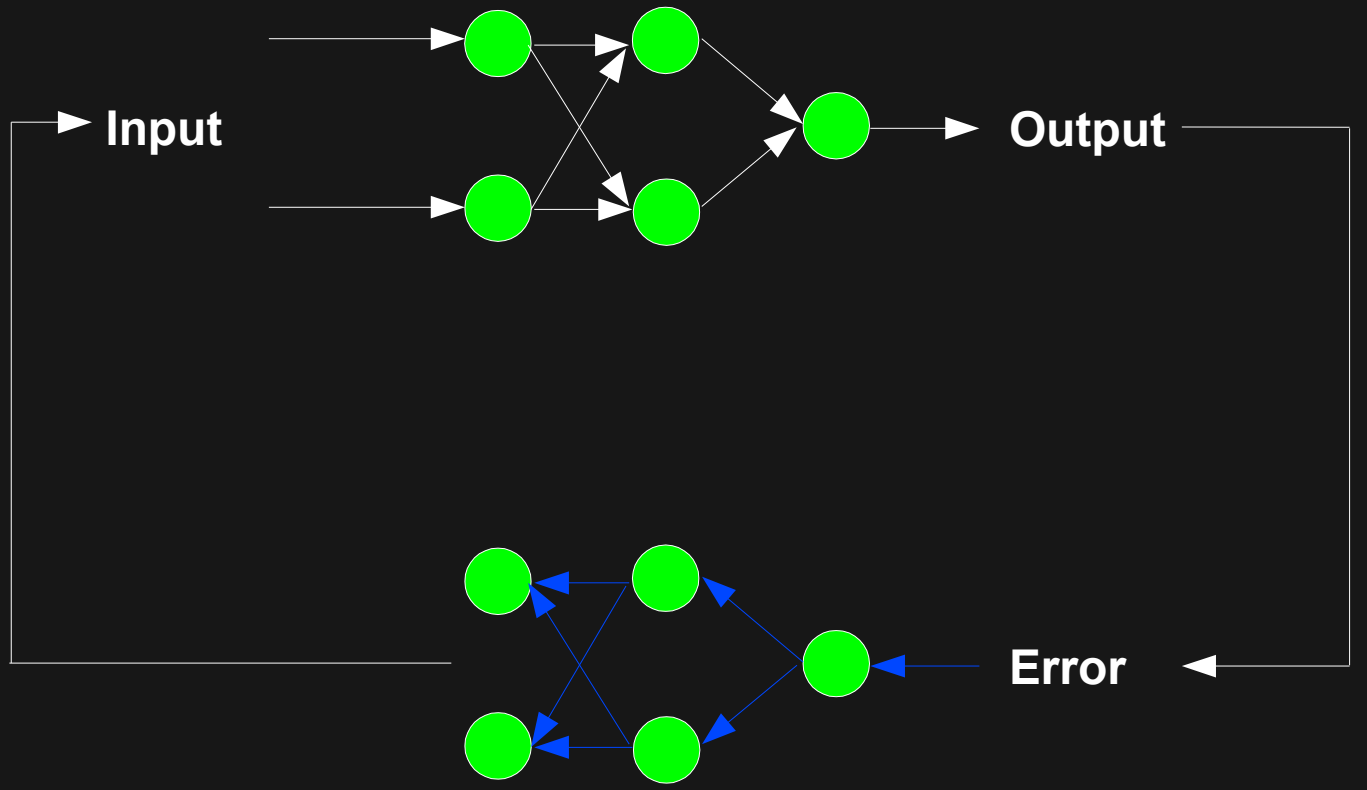
```
for(i=0; i<NEURONS; i++) {
        for(j=0, x=0; j<LayerDimms[i]; j++)
            x += y[j]*w[j][i];
        y[i] = x - t[i];
}
```

- Training

```
do {
        e += Train(y, x);
} while (e > ERRMIN);
```

BP Neural Networks

- Running (XOR)



- Training

Feedback?

- What do you think?
- Possible Applications?
- Questions?

Conclusions / Shameful Plugs

- ToorCon 7
  - End of September, 2005
  - San Diego, CA USA
  - http://www.toorcon.org
- ShmooCon
  - Super Bowl Weekend, 2005
  - Washington DC, USA
  - http://www.shmoocon.com
- LayerOne
  - June, 2005
  - Los Angeles, USA
  - http://www.layerone.info

Questions ? Suggestions ?

- David Hulton
  - 0x31337@gmail.com
  - h1kari@dachb0den.com Will be back up soon!
- OpenCores
  - http://www.opencores.org
- Xilinx
  - ISE Foundation (Free 60-day trial)
- Pico Computing, Inc.
  - http://www.picocomputing.com
  - Products will be available around March, 2005