The CoSMoS multi-FPGA Simulation Facility

Jack Whitham

11th September 2008

The multi-FPGA Simulation Facility

1. The facility is a central shared resource for running FPGA-based applications.

The multi-FPGA Simulation Facility

1. The facility is a central shared resource for running FPGA-based applications.
2. It is accessed securely via the Internet.

The multi-FPGA Simulation Facility

1. The facility is a central shared resource for running FPGA-based applications.
2. It is accessed securely via the Internet.
3. It is already being used for research.

The multi-FPGA Simulation Facility

1. The facility is a central shared resource for running FPGA-based applications.
2. It is accessed securely via the Internet.
3. It is already being used for research.
4. Next term, it will be used for teaching.

Why FPGAs?

- A CPU runs *sequential* applications efficiently.

Why FPGAs?

- A CPU runs *sequential* applications efficiently.
- A cluster of CPUs runs *coarse-grained parallel* applications efficiently.

Why FPGAs?

- A CPU runs *sequential* applications efficiently.
- A cluster of CPUs runs *coarse-grained parallel* applications efficiently.
- But *no* CPU runs *fine-grained parallel* applications efficiently.

Why FPGAs?

- A CPU runs *sequential* applications efficiently.
- A cluster of CPUs runs *coarse-grained parallel* applications efficiently.
- But *no* CPU runs *fine-grained parallel* applications efficiently.

FPGAs are good at running applications that contain a high degree of concurrency.

Why a central shared FPGA resource?

Consider the alternative:

- One FPGA per researcher?

Why a central shared FPGA resource?

Consider the alternative:

- One FPGA per researcher?
- What if the application is too large for one FPGA?

Why a central shared FPGA resource?

Consider the alternative:

- One FPGA per researcher?
- What if the application is too large for one FPGA?
  - *n* FPGAs per researcher?
  - Issues: physical space, £s, interconnections, expertise.

Why a central shared FPGA resource?

Consider the alternative:

- One FPGA per researcher?
- What if the application is too large for one FPGA?
  - *n* FPGAs per researcher?
  - Issues: physical space, £s, interconnections, expertise.
- Solution: *n* FPGAs for *m* researchers.

Why a central shared FPGA resource?

Consider the alternative:

- One FPGA per researcher?
- What if the application is too large for one FPGA?
  - *n* FPGAs per researcher?
  - Issues: physical space, £s, interconnections, expertise.
- Solution: *n* FPGAs for *m* researchers.
  - A time-shared computing resource for multi-FPGA applications!

Requirements

- Secure remote access.

Requirements

- Secure remote access.
- Good communication links between FPGAs, so that applications can span multiple FPGAs.

Requirements

- Secure remote access.
- Good communication links between FPGAs, so that applications can span multiple FPGAs.
- Ease of programming: no need to learn VHDL/Verilog.

Solutions

- **Secure remote access** is implemented using the Secure Shell (SSH) protocol, with RSA authentication. This allows us to provide access over the public Internet.

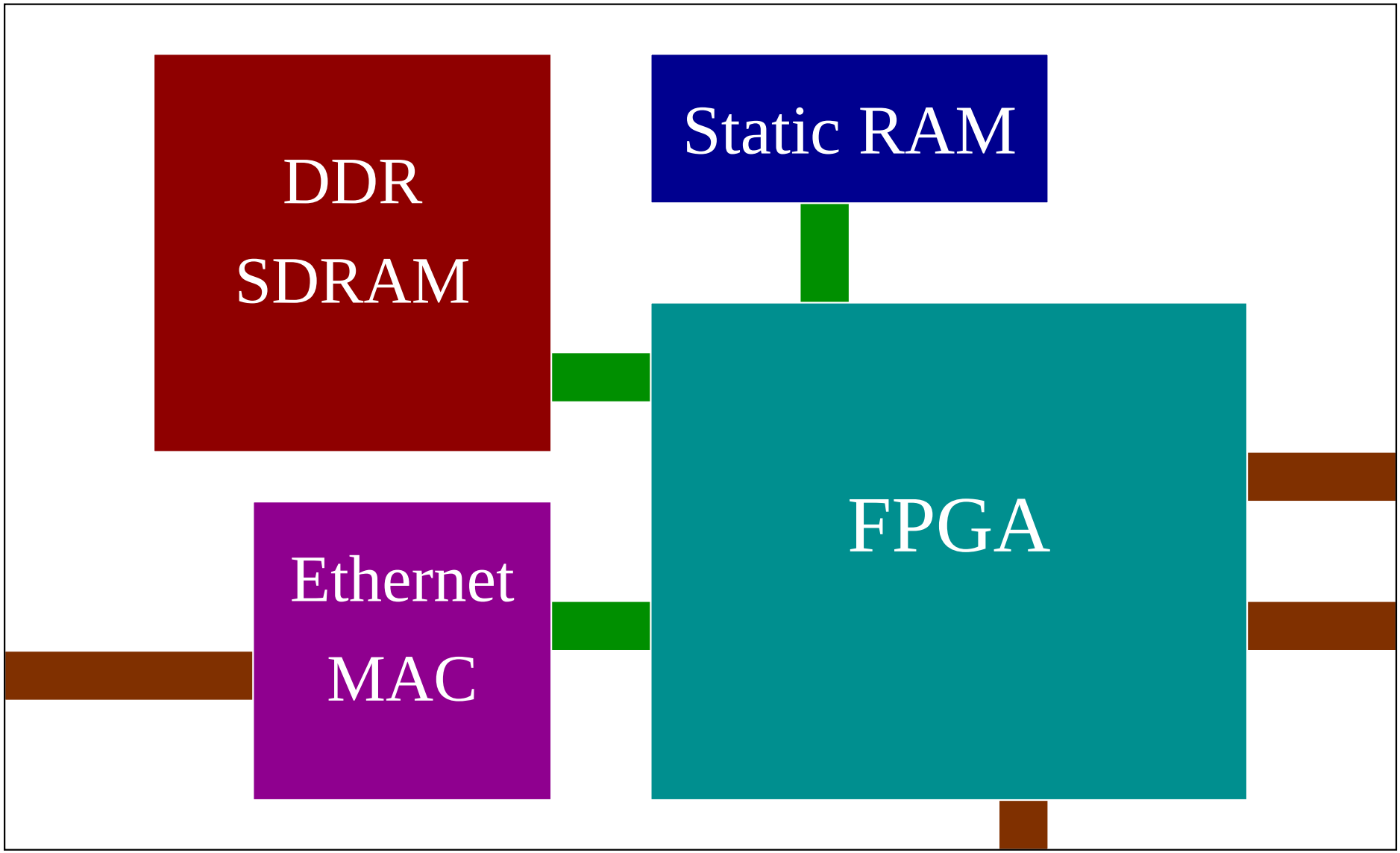
Solutions

- **Secure remote access** is implemented using the Secure Shell (SSH) protocol, with RSA authentication. This allows us to provide access over the public Internet.
- **Good communication links between FPGAs** are possible with a hierarchical structure:
  - FPGAs are organised into clusters of four, with fast point-to-point links within each cluster.
  - Each FPGA has an Ethernet connection for communication between clusters.

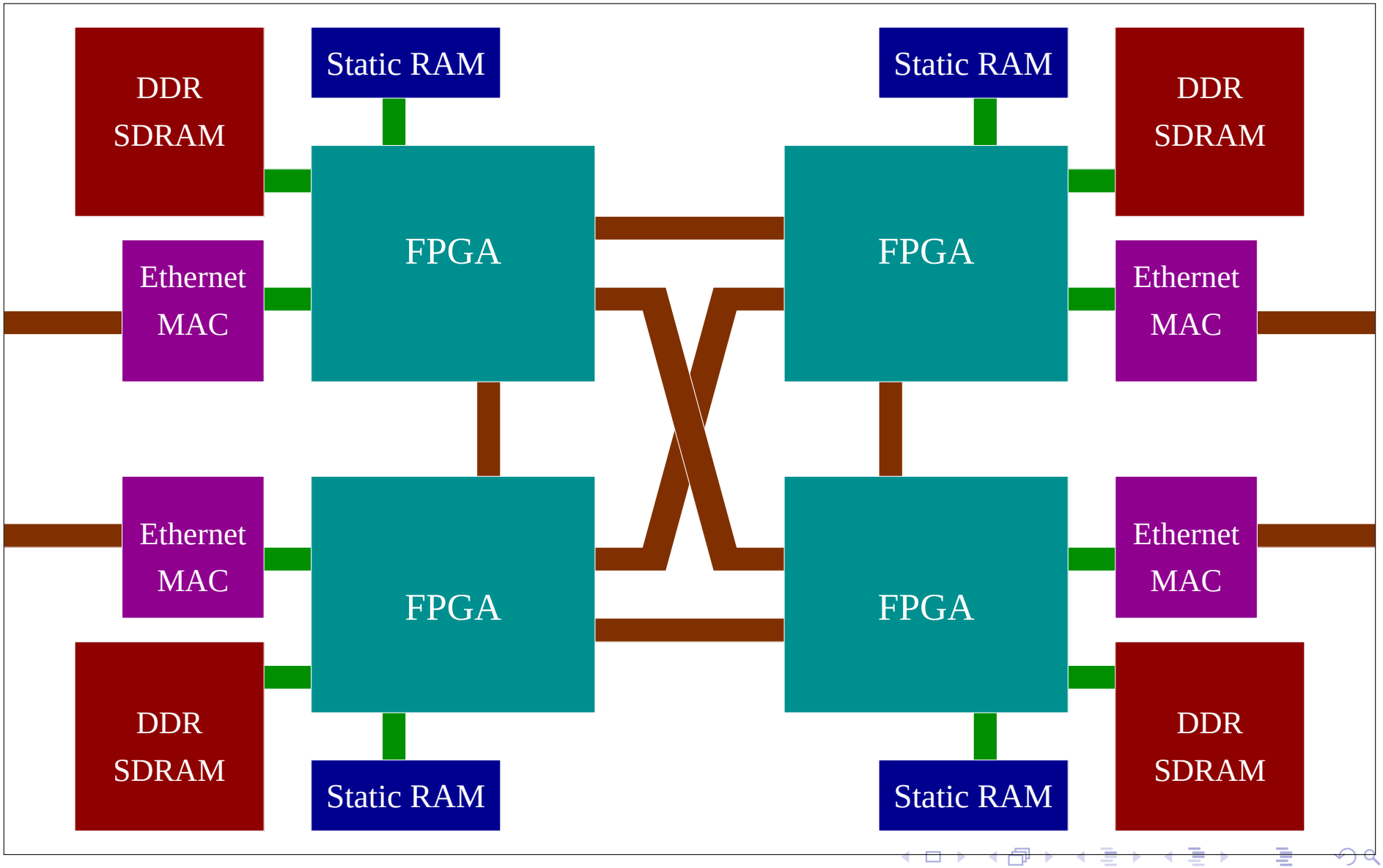
Solutions

- **Secure remote access** is implemented using the Secure Shell (SSH) protocol, with RSA authentication. This allows us to provide access over the public Internet.
- **Good communication links between FPGAs** are possible with a hierarchical structure:
  - FPGAs are organised into clusters of four, with fast point-to-point links within each cluster.
  - Each FPGA has an Ethernet connection for communication between clusters.
- **Ease of programming** is helped by our Handel C environment. This allows you to use the Handel C language in place of VHDL and Verilog.
  - Handel C is a C-like language for hardware design: includes syntax for parallel statements, and occam-like channels.

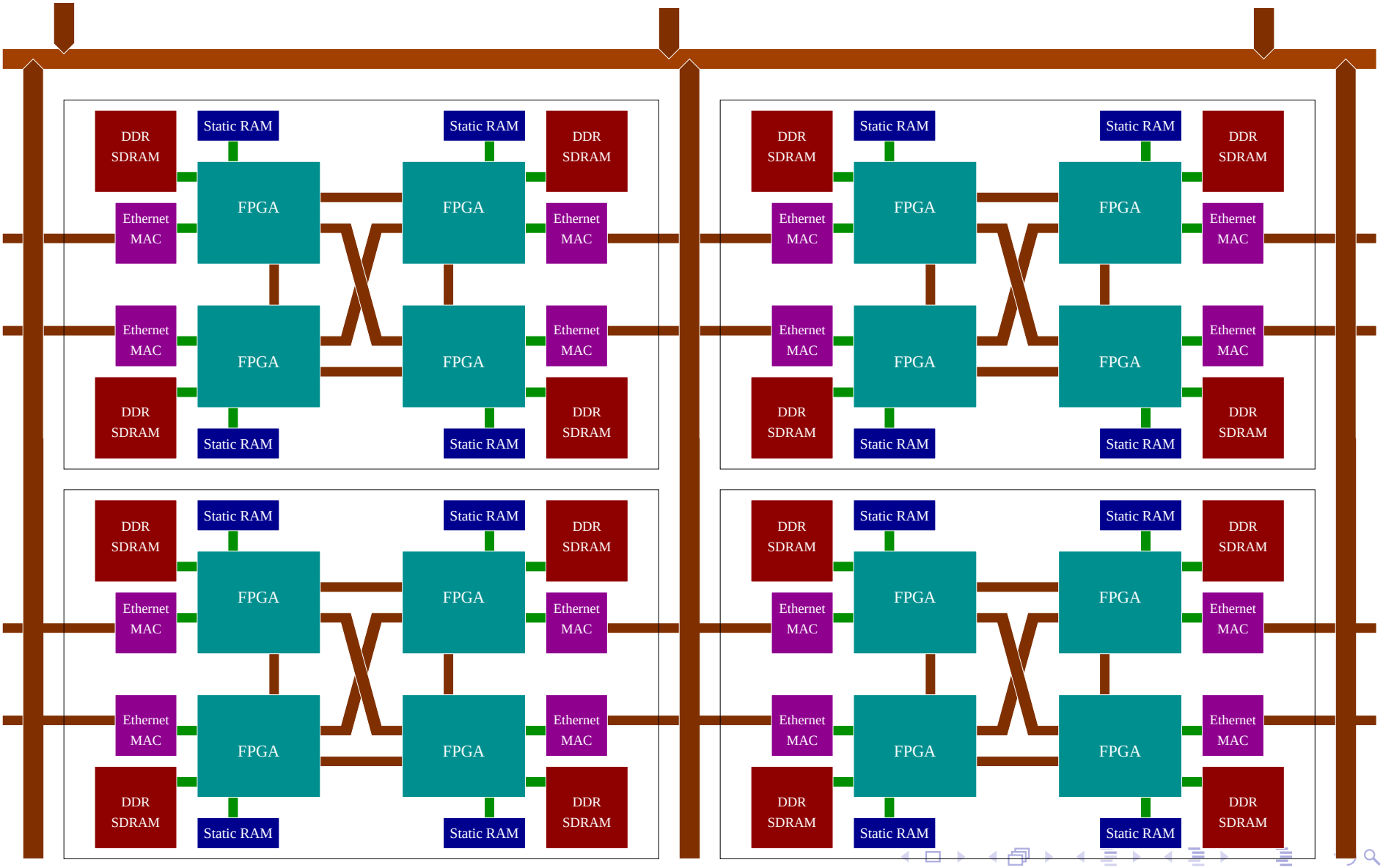
One FPGA



One cluster


DDR SDRAM
Static RAM
Static RAM
DDR SDRAM
FPGA
FPGA
Ethernet MAC
Ethernet MAC
Ethernet MAC
Ethernet MAC
FPGA
FPGA
DDR SDRAM
DDR SDRAM
Static RAM
Static RAM

Many clusters



Using the Resource

- A traditional computer would be remotely accessed by: SSH, telnet, 3270, remote desktop, NX, etc...

Using the Resource

- A traditional computer would be remotely accessed by: SSH, telnet, 3270, remote desktop, NX, etc...
- These paradigms do not apply to FPGAs: no interactive operating system is present.

Using the Resource

- A traditional computer would be remotely accessed by:
  SSH, telnet, 3270, remote desktop, NX, etc...
- These paradigms do not apply to FPGAs:
  no interactive operating system is present.

However... these protocols make no assumptions about the applications that will be used.

Using the Resource

- A traditional computer would be remotely accessed by:
  SSH, telnet, 3270, remote desktop, NX, etc...
- These paradigms do not apply to FPGAs:
  no interactive operating system is present.

However... these protocols make no assumptions about the applications that will be used.

- What is the FPGA equivalent?

Minimal Access

Remote FPGA access requires a minimum of two operations:

- Load a *bit file* onto an FPGA.

Minimal Access

Remote FPGA access requires a minimum of two operations:

- Load a *bit file* onto an FPGA.
- Open a channel to communicate with the FPGA.

Minimal Access

Remote FPGA access requires a minimum of two operations:

- Load a *bit file* onto an FPGA.
- Open a channel to communicate with the FPGA.

Q. What is the simplest way for one piece of hardware to communicate with another?

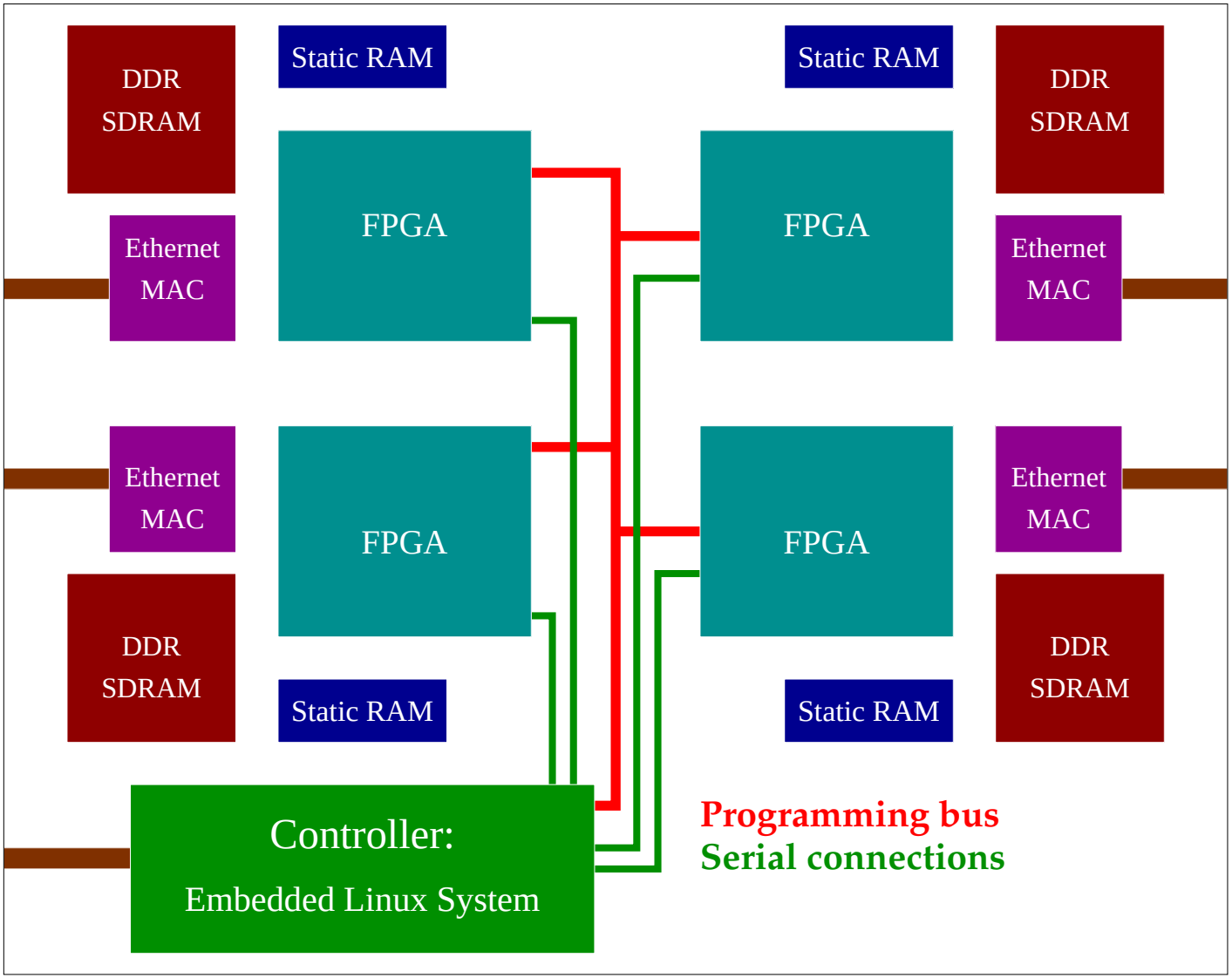
Minimal Access

Remote FPGA access requires a minimum of two operations:

- Load a *bit file* onto an FPGA.
- Open a channel to communicate with the FPGA.

Q. What is the simplest way for one piece of hardware to communicate with another?
A. A serial connection (RS232, etc.). The serial protocol is standard, requires very little hardware, and is supported almost everywhere.

Controller


DDR SDRAM
Static RAM
Static RAM
DDR SDRAM
FPGA
FPGA
Ethernet MAC
Ethernet MAC
Ethernet MAC
FPGA
FPGA
Ethernet MAC
DDR SDRAM
DDR SDRAM
Static RAM
Static RAM
Controller:
Embedded Linux System
Programming bus
Serial connections

Basic Services

At the most basic level, the multi-FPGA resource provides:

- An online service allowing you to (remotely):
  - program an FPGA, then
  - communicate with it via a serial connection.

Basic Services

At the most basic level, the multi-FPGA resource provides:

- An online service allowing you to (remotely):
  - program an FPGA, then
  - communicate with it via a serial connection.
- Software written in Python that allows you to do the above from a program or script.

Example

Python code

```
 1  import vlab
 2  from twisted.internet import reactor, defer
 3
 4  @defer.inlineCallbacks
 5  def Run():
 6      print 'Loading bit file'
 7      bits = file('test.bit', 'rb').read()
 8      print 'Connecting to the board'
 9      auth = vlab.loadAuthorisation("vluser.key")
10      vlf = vlab.VlabClientFactory(auth)
11      reactor.connectTCP(auth.relay_server_name, 22, vlf)
12      vl = yield vlf.getChannel()
13      yield vl.connect("s3esk")
14      print 'Sending bit file'
15      bid = yield vl.sendBitfile(bits)
16      print 'Bid is', bid
17      yield vl.programFPGA(0, bid)
18      print 'Done'
19      vl.disconnect()
20      reactor.stop()
```

Improved Services

We build on the basic services to provide:

- Handel C drivers to make FPGA programming simpler.

Improved Services

We build on the basic services to provide:

- Handel C drivers to make FPGA programming simpler.
- vlabifhw: Hardware that multiplexes the serial connection to provide multiple "ports" into your design, including:
  - a debugging channel providing an n-bit wide parallel interface to your design,
  - and up to 14 bidirectional 8-bit parallel channels.

Improved Services

We build on the basic services to provide:

- Handel C drivers to make FPGA programming simpler.
- vlabifhw: Hardware that multiplexes the serial connection to provide multiple "ports" into your design, including:
  - a debugging channel providing an n-bit wide parallel interface to your design,
  - and up to 14 bidirectional 8-bit parallel channels.
- ethernet: a driver for the Ethernet chip.

Improved Services

We build on the basic services to provide:

- Handel C drivers to make FPGA programming simpler.
- vlabifhw: Hardware that multiplexes the serial connection to provide multiple "ports" into your design, including:
  - a debugging channel providing an n-bit wide parallel interface to your design,
  - and up to 14 bidirectional 8-bit parallel channels.
- ethernet: a driver for the Ethernet chip.
- Software drivers for vlabifhw.

Demonstration

Embedded Systems (EMS) course.

How it works

On the FPGA (Handel C)

```
#include "vlab.hch"
char rot13(char c)
{
    if(c > 'z') return c;
    if(c >= 'n') return c - 13;
    if(c >= 'a') return c + 13;
    return c;
}
void main(void)
{
    char c;
    par {
        vlab_uart_driver(&uart);
        while(1) {
            uart ? c;
            c = rot13(c);
            uart ! c;
        }
    }
}
```

How it works

On the PC (Python):

Switch event

```
switch_val = event.switch_state
switch_reg = self.dbg_chan.getAsDict()[ 'switches' ]
switch_reg.setOutput(switch_val)
self.dbg_chan.uploadChain()
```

Rotary control event

```
motion = event.motion
ccw = ( motion < 0 )
for i in xrange(abs(motion)):
    self.control_chan.virtualButtons(rot_is_ccw=ccw, rot_rotation=True)
    self.control_chan.virtualButtons()
```

Roadmap

- Our "virtual lab" resource currently features a variety of FPGA boards (Spartan-3E Starter Kit, ML401).

Roadmap

- Our "virtual lab" resource currently features a variety of FPGA boards (Spartan-3E Starter Kit, ML401).
- It has already been used for some research (JEOPARD project).

Roadmap

- Our "virtual lab" resource currently features a variety of FPGA boards (Spartan-3E Starter Kit, ML401).
- It has already been used for some research (JEOPARD project).
- The multi-FPGA board is in the final design phases; prototypes are expected around Christmas.

Roadmap

- Our "virtual lab" resource currently features a variety of FPGA boards (Spartan-3E Starter Kit, ML401).
- It has already been used for some research (JEOPARD project).
- The multi-FPGA board is in the final design phases; prototypes are expected around Christmas.
- Handel C drivers are being written.

Conclusion

- Much of the infrastructure is already complete and working.

Conclusion

- Much of the infrastructure is already complete and working.
- But some of the hardware is not yet finished.

Conclusion

- Much of the infrastructure is already complete and working.
- But some of the hardware is not yet finished.
- Handel C drivers are also currently incomplete.

Conclusion

- Much of the infrastructure is already complete and working.
- But some of the hardware is not yet finished.
- Handel C drivers are also currently incomplete.
- You'll be able to experiment with the system soon, at least on Spartan-3E Starter Kit FPGAs.

Conclusion

- Much of the infrastructure is already complete and working.
- But some of the hardware is not yet finished.
- Handel C drivers are also currently incomplete.
- You'll be able to experiment with the system soon, at least on Spartan-3E Starter Kit FPGAs.

Find the manual online at `http://www.jwhitham.org.uk/vl2/`.
Email me (see website) for access information.