Building Loosely-coupled RISC-V Accelerators

Using Chisel/FIRRTL to build accelerator templates and collateral for the ESP SoC platform

Schuyler Eldridge

2020-02-01

Why Accelerators?

The Golden Age of Computer Architecture is about Accelerators (Assuming you don't hit the Accelerator Wall...[1])

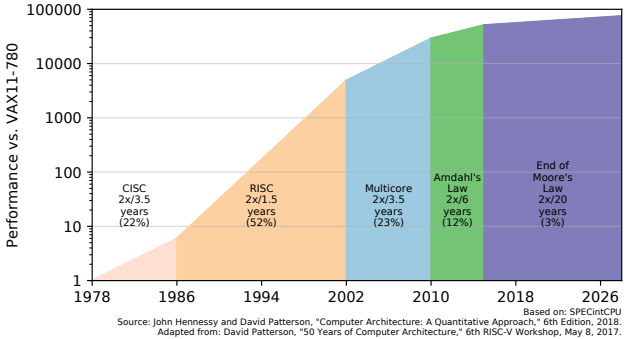


Figure 1: 50 Years of Process Performance (Measured as SPECInt Score)

[1] Fuchs, A. and Wentzlaff, D., *The accelerator wall: Limits of chip specialization*, 2019

Accelerator Types: Tightly vs. Loosely-coupled

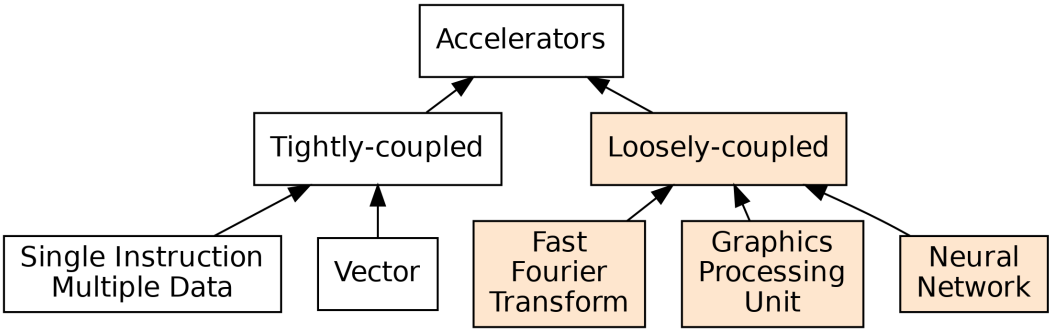


Figure 2: A type hierarchy of accelerators

Loosely-coupled: Config, Read, Compute, Write

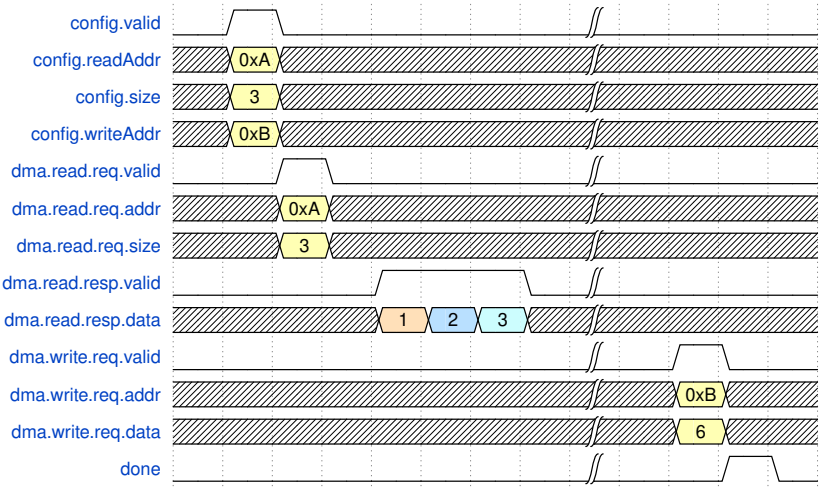


Figure 3: Waveform showing loosely-coupled accelerator timing[2]

[2]Waveform generated using using Wavedrom (github.com/wavedrom/wavedrom)

Embedded Scalable Platform (ESP) Accelerator “API”

ESP Capabilities

- Bring your own accelerator in any language
- Accelerator can be easily integrated with a Leon3 or Ariane System-on-Chip

Embedded Scalable Platform (ESP) Accelerator “API”

ESP Capabilities

- Bring your own accelerator in any language
- Accelerator can be easily integrated with a Leon3 or Ariane System-on-Chip

Clarifications

Embedded Scalable Platform (ESP) Accelerator “API”

ESP Capabilities

- Bring your own accelerator in any language
- Accelerator can be easily integrated with a Leon3 or Ariane System-on-Chip

Clarifications

- How is ESP made aware of the accelerator? (An XML file)

Embedded Scalable Platform (ESP) Accelerator “API”

ESP Capabilities

- Bring your own accelerator in any language
- Accelerator can be easily integrated with a Leon3 or Ariane System-on-Chip

Clarifications

- How is ESP made aware of the accelerator? (An XML file)
- How does the user know what to write? (An example module)

Example: “AdderAccelerator”

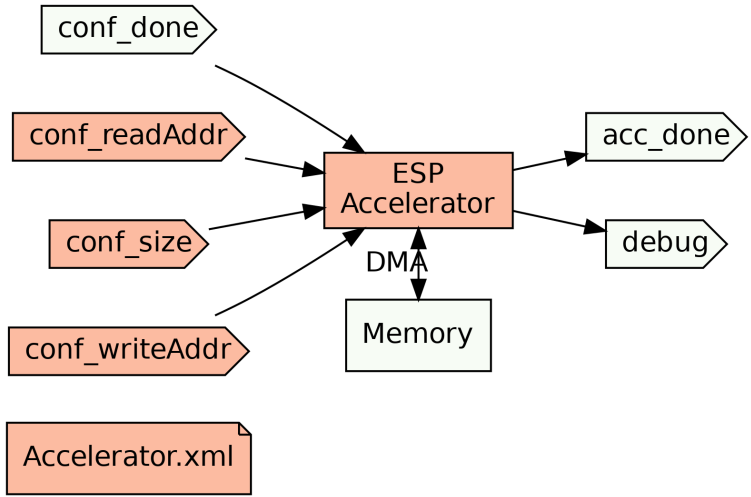
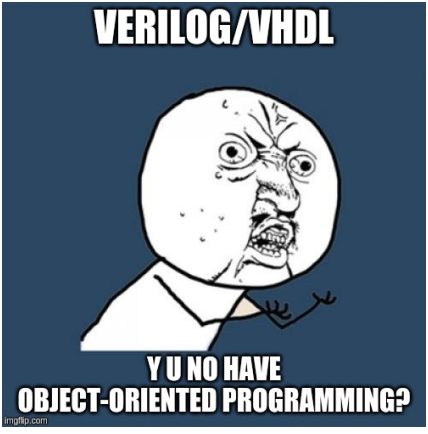


Figure 4: ESP Accelerator Socket

Just Code It Up in ((System)?Verilog)|VHDL

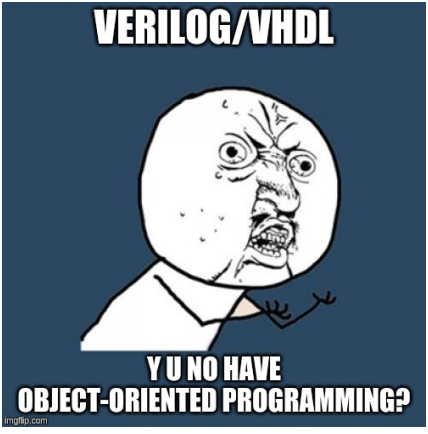
Just Code It Up in ((System)?Verilog)|VHDL

Just Code It Up in ((System)?Verilog)|VHDL



[3] Taylor, M.B., *Basejump stl: Systemverilog needs a standard template library for hardware design*, 2018

Just Code It Up in ((System)?Verilog)|VHDL



SystemVerilog is *hard* to use to build libraries.[3]

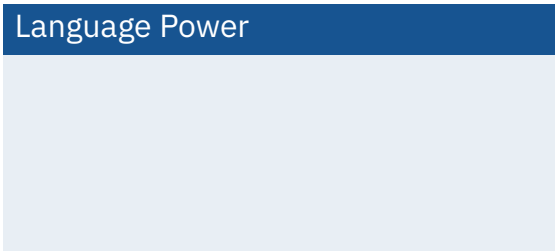
[3] Taylor, M.B., *Basejump stl: Systemverilog needs a standard template library for hardware design*, 2018

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler



Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric
1. Parametric Polymorphic

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric
1. Parametric Polymorphic
1. First Class Functions

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric
1. Parametric Polymorphic
1. First Class Functions
1. Functional Programming

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric
1. Parametric Polymorphic
1. First Class Functions
1. Functional Programming
1. Object-oriented Programming

Enter Chisel/FIRRTL

- Chisel is a Scala hardware DSL
- FIRRTL is a circuit IR and an optimizing circuit compiler

Language Power

1. Parametric
1. Parametric Polymorphic
1. First Class Functions
1. Functional Programming
1. Object-oriented Programming

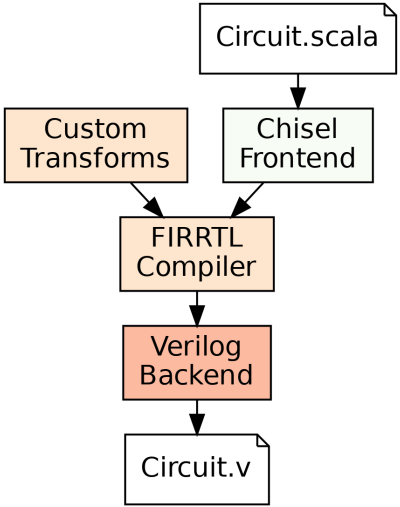


Figure 5: Chisel/FIRRTL Verilog compilation

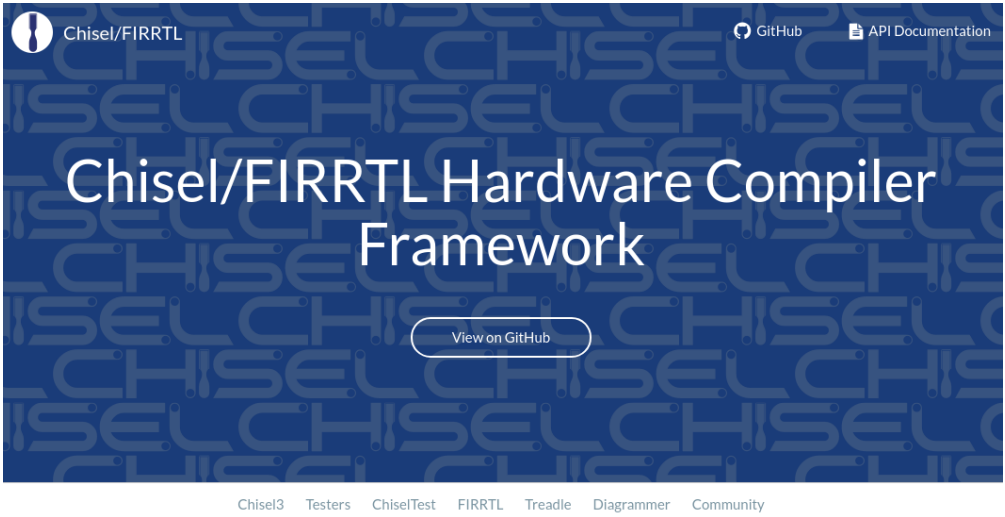


Figure 6: Chisel Website

ESP Chisel Accelerators[4]

```
trait Specification {
  /** Accelerator Config */
  def config: Config
}

abstract class Implementation
  extends Module
  with Specification {

  /** Accelerator Name */
  def implementationName: String
}
```

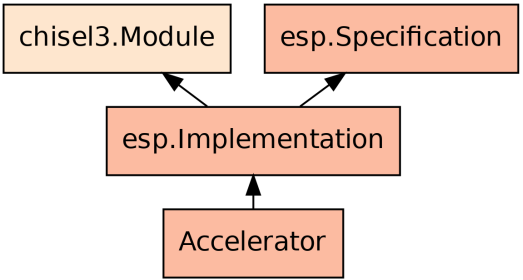


Figure 7: Composition of an ESP accelerator

An ESP accelerator is composed of an `Implementation` and a `Specification`.

[4] github.com/IBM/esp-chisel-accelerators

Example: Specification

```
trait AdderSpec extends Specification {

  override lazy val config = Config(
    name = "AdderAccelerator",
    description = "Reduces a vector via addition",
    memoryFootprintMiB = 1,
    deviceId = 0xF,
    param = Array(
      Parameter( name = "readAddr" ),
      Parameter( name = "size" ),
      Parameter( name = "writeAddr" )
    )
  )

}
```

Example: Implementation

```
class Adder extends Implementation with AdderSpec {

  override val implementationName = "AdderAccelerator"

  /** Implement me! */
}
```

Canned Demo

```
# sbt run
# tree build
build
`-- AdderAccelerator
    |-- AdderAccelerator_Default_dma32
    |   |-- AdderAccelerator_AdderAccelerator_dma32.anno.json
    |   |-- AdderAccelerator_AdderAccelerator_dma32.fir
    |   `-- AdderAccelerator_AdderAccelerator_dma32.v
    `-- AdderAccelerator.xml
```

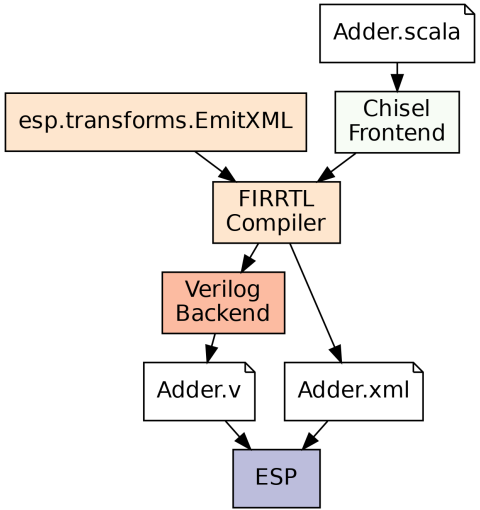


Figure 8: ESP Chisel Accelerators Flow

XML Output

```
<sld>
  <accelerator name="AdderAccelerator"
               desc="Reduces a vector via addition"
               data_size="1" device_id="15">
    <param name="readAddr"/>
    <param name="size"/>
    <param name="writeAddr"/>
  </accelerator>
</sld>
```

Wrapping Up

[5] https://github.com/grebe/ofdm

Wrapping Up

Current `esp-chisel-accelerators`

- CounterAccelerator (“hello world”)
- AdderAccelerator
- FFTAccelerator[5]

[5] https://github.com/grebe/ofdm

Wrapping Up

Current `esp-chisel-accelerators`

- CounterAccelerator (“hello world”)
- AdderAccelerator
- FFTAccelerator[5]

Future Work

- Additional collateral generation including:
  - Basic bare metal and Linux test programs
  - Drivers
- New accelerators

[5] https://github.com/grebe/ofdm

More Info

- ESP ..... github.com/sld-columbia/esp
  - Chisel Accelerators ..... github.com/IBM/esp-chisel-accelerators
- Chisel3 ..... github.com/freechipsproject/chisel3
  - Twitter ..... @chisel_lang
- FIRRTL ..... github.com/freechipsproject/firrtl

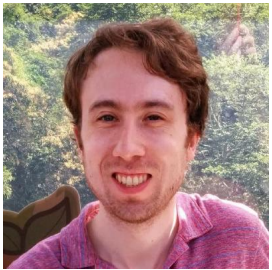


Figure 9: github:@seldridge

Acknowledgments

The research reported in this talk was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. government.

Distribution Statement A: Approved for public release, distribution unlimited.

