



FireSim

A Brief Tour of FireSim:
The Manager & Compiler;
Building Hardware Designs

<https://firesim.com>



@firesimproject

MICRO 2019 Tutorial

Speaker: David Biancolin (@davidbiancolin)



Berkeley Architecture Research



Agenda: What Will We Cover?

1) The Compiler → Golden Gate

- Invoke it on example RTL
- Inspect its outputs

2) The Manager → `firesim`

- Explain how it's configured
- Demonstrate how it's used to build bitstreams



Where is FireSim in Chipyard?

With the software RTL simulators!

~/chipyard-afternoon/sims/firesim

→ This has been exported as \$FDIR



Interactive:

```
# <ssh back onto your ec2 instance>
```

```
$ tmux new -s afternoon
```

```
$ cd $FDIR
```

```
$ ls
```



FireSim's Directory Structure

`sim/`

- Golden Gate lives here
- Scala & C++ sources for additional FireSim models
- Make-based build system to invoke Golden Gate

`deploy/`

- Manager lives here
- FireSim workload definitions

`platforms/` → AWS FPGA/Vivado project definitions

`sw/` → target software & FireMarshal (more on this later)



Agenda: What Will We Cover?

1) The Compiler → “Golden Gate”

- Invoke it on example RTL
- Inspect its outputs

2) The Manager → `firesim`

- Explain how it’s configured
- Demonstrate how it’s used to build bitstreams



Interactive:

```
$ cd $FDIR/sim
```

```
$ make DESIGN=FireSimNoNIC
```



An Analogy

- Golden Gate is like Verilator but for FPGA-accelerated simulation

Verilator generates C++ sources to simulate your design.

→ Compile and run on a CPU-host

Golden Gate generates C++ & Verilog to simulate your design.

→ Compile and run on a hybrid CPU & FPGA host



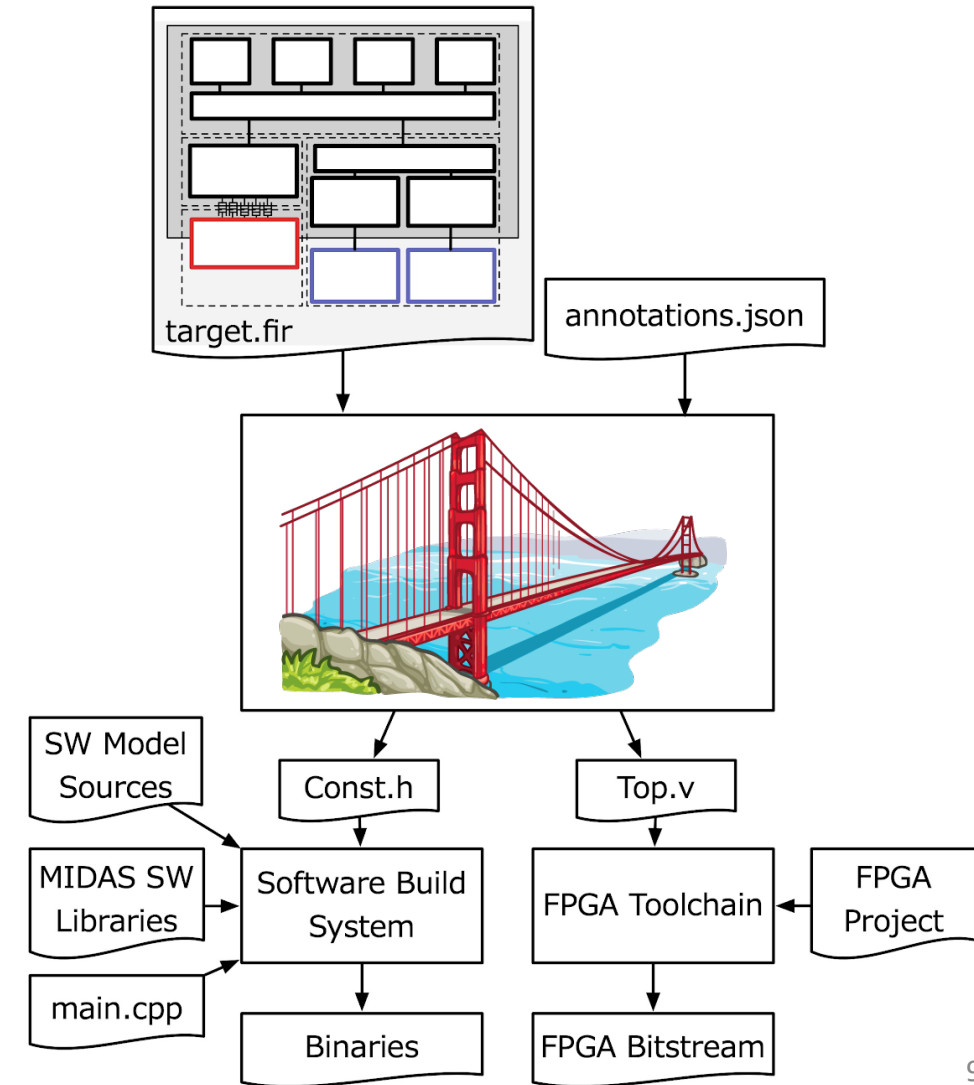
Golden Gate Compiler

Inputs:

- FIRRTL & annos from a Chipyard generator
- Compiler configuration

→ Produces sources for a simulator that are:

- deterministic
- support co-simulation of software models
- *area-optimized to fit more on the FPGA*





Plug: ICCAD 2019 Publication

GOLDEN GATE: Bridging The Resource-Efficiency Gap Between ASICs and FPGA Prototypes

Albert Magyar, David Biancolin, John Koenig, Sanjit Seshia, Jonathan Bachrach, Krste Asanović

Punchline:

- Can fit two more BOOM cores (4 -> 6)
- think: “-Os for FireSim”

These optimizations ship with FireSim 1.7.0!



Interacting with Golden Gate via Make

- Make invokes Golden Gate with three variables (the “Tuple”):

DESIGN :

- The top level module → MODEL in Chipyard

TARGET_CONFIG:

- The generator’s config → CONFIG in Chipyard

PLATFORM_CONFIG:

- Compiler options passed to Golden Gate



Interactive:

```
$ cd $FDIR/sim/generated-src/f1
```

```
# here you'll find output directories for all builds
```

```
$ cd <any-directory-here>
```

```
$ ls
```



Inspecting the Outputs

`<long-name>.fir & <long-name>.anno.json`

- Target's FIRRTL & annotations

`FPGATop.v`

- The compiled simulator

`$DESIGN-const.h`

- Simulator's memory map

`runtime.conf`

- A default runtime configuration for simulation



Agenda: What Will We Cover?

1) The Compiler → Golden Gate

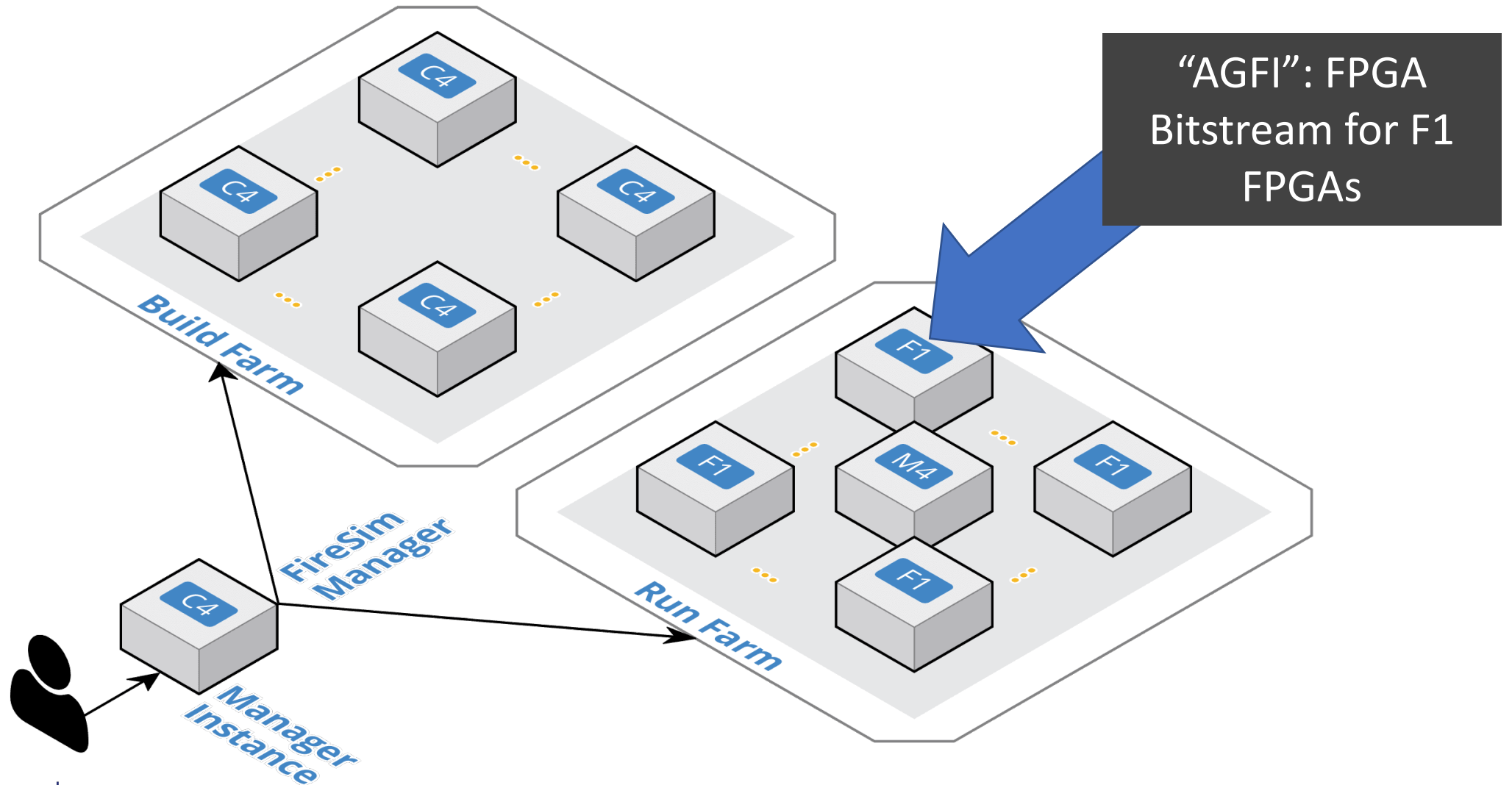
- Invoke it on example RTL
- Inspect its outputs

2) The Manager → `firesim`

- Explain how it's configured
- Demonstrate how it's used to build bitstreams



Background Terminology





Using the `firesim` Manager Command Line

- sourcing `sourceme-fl-manager.sh` puts `firesim` on your path
- can call `firesim` from anywhere on the instance
- it will always run from the directory:

```
$FDIR/deploy/
```

After a fresh clone, need to call:

```
firesim managerinit
```

→ You did this at the start of the tutorial



Interactive:

```
$ cd $FDIR/deploy
```

```
$ ls
```



Configuring the Manager. 4 files in firesim/deploy/

config_build_recipes.ini

```

1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
2 # these params.
3 # This file contains sections that describe hardware designs that /can/ be built.
4 # edit config_build.ini to actually "turn on" a config to be built when you run
5 # builddefi.
6
7 [[firesim-singlecore-no-nic-lbp]]
8 DESIGN=FireSimNoNIC
9 TARGET_CONFIG=FireSimRocketChipSingleCoreConfig
10 PLATFORM_CONFIG=FireSimConfig
11 instancetype=C4_4xlarge
12 deploytriple=None
13
14 [[firesim-quadcore-nic-ddr3-11c4mb]]
15 DESIGN=FireSim
16 TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
17 PLATFORM_CONFIG=FireSimDDR3FRFCPLL4MBConfig
18 instancetype=C4_4xlarge
19 deploytriple=None
20
21 [[firesim-quadcore-no-nic-ddr3-11c4mb]]
22 DESIGN=FireSimNoNIC
23 TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
24 PLATFORM_CONFIG=FireSimDDR3FRFCPLL4MBConfig
25 instancetype=C4_4xlarge
26 deploytriple=None
27
28 # BOOM-based targets
29 [[fireboom-singlecore-no-nic-ddr3-11c4mb]]
30 DESIGN=FireSimNoNIC
31 TARGET_CONFIG=FireSimBoomConfig
32 PLATFORM_CONFIG=FireSimDDR3FRFCPLL4MBConfig
33 instancetype=C4_4xlarge
34 deploytriple=None
35
36 [[fireboom-singlecore-nic-ddr3-11c4mb]]
37 DESIGN=FireBoom
38 TARGET_CONFIG=FireSimBoomConfig
39 PLATFORM_CONFIG=FireSimDDR3FRFCPLL4MBConfig
40 instancetype=C4_4xlarge
41 deploytriple=None

```

config_build.ini

```

1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
2 # these params.
3
4 [[agfbuild]]
5 sbucketname=firesim-72179603761
6 buildinstancemarket=ondemand
7 spotinterruptionbehavior=terminate
8 spotmaxprice=ondemand
9
10 [[builds]]
11 # this section references builds defined in config_build_recipes.ini
12 # if you add a build here, it will be built when you run builddefi
13
14 [[firesim-singlecore-no-nic-lbp]]
15 firesim-singlecore-no-nic-ddr3-11c4mb
16 fireboom-singlecore-no-nic-ddr3-11c4mb
17
18 [[agfistoshane]]
19 firesim-singlecore-no-nic-lbp
20 fireboom-singlecore-no-nic-ddr3-11c4mb
21 fireboom-quadcore-nic-ddr3-11c4mb
22
23 [[sharewithaccounts]]
24 somebodyname=123456789012

```

config_hwdb.ini

```

1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
2 # these params.
3
4 # Hardware configs represent a combination of an agfi, a deploytriple override
5 # (if needed), and a custom runtime config (if needed)
6
7 # The AGFIs provided below are public and available to all users.
8 # Only AGFIs for the latest release of FireSim are guaranteed to be available.
9 # If you are using an older version of FireSim, you will need to generate your
10 # own images.
11
12 [[firesim-singlecore-no-nic-lbp]]
13 agfi=agfi-0584a1a71df6a005a
14 deploytripleoverride=None
15 customruntimeconfig=None
16
17 [[firesim-quadcore-no-nic-ddr3-11c4mb]]
18 agfi=agfi-06b9b705ab9af1238
19 deploytripleoverride=None
20 customruntimeconfig=None
21
22 [[firesim-quadcore-nic-ddr3-11c4mb]]
23 agfi=agfi-030b49bce9b5e9f96
24 deploytripleoverride=None
25 customruntimeconfig=None
26
27 [[fireboom-singlecore-nic-ddr3-11c4mb]]
28 agfi=agfi-090491454199fb160
29 deploytripleoverride=None
30 customruntimeconfig=None
31
32 [[fireboom-singlecore-no-nic-ddr3-11c4mb]]
33 agfi=agfi-0df9101df7b7ff708
34 deploytripleoverride=None
35 customruntimeconfig=None

```

config_runtime.ini

```

1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
2 # these params.
3
4 [[runfarm]]
5
6 [[runfarm]]
7 mainrunfarm
8
9
10 [[instancemarket=ondemand]]
11 spotinterruptionbehavior=terminate
12 spotmaxprice=ondemand
13
14 [[largestconfig]]
15 topology_no_net_config
16 no_net_num_nodes=1
17 linklatency=5405
18 switchinglatency=10
19 netbandwidth=200
20 profileinterval=1
21
22 # This references a section from config_hwconfigs.ini
23 # In homogeneous configurations, use this to set the hardware config deployed
24 # for all simulators
25 defaulthwconfig=fireboom-quadcore-no-nic-ddr3-11c4mb
26
27 [[tracing]]
28 enable=no
29 startcycle=0
30 endcycle=1
31
32 [[workload]]
33 workloadname=linux-uniform.json
34 terminateoncompletion=no

```





Configuring a Build

```
0 / mosh-client (tmux) #1
config_build_recipes.ini buffers
1 # Build-time design configuration for the FireSim Simulation Manager
1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
  these params.
2
3 # this file contains sections that describe hardware designs that /can/ be built.
4 # edit config_build.ini to actually "turn on" a config to be built when you run
5 # buildafi
6
7 [firesim-singlecore-no-nic-lbp]
8 DESIGN=FireSimNoNIC
9 TARGET_CONFIG=FireSimRocketChipSingleCoreConfig
10 PLATFORM_CONFIG=FireSimConfig
11 instancetype=c4.4xlarge
12 deploytriplet=None
13
14 [firesim-quadcore-nic-ddr3-1lc4mb]
15 DESIGN=FireSim
16 TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
17 PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
18 instancetype=c4.4xlarge
19 deploytriplet=None
20
21 [firesim-quadcore-no-nic-ddr3-1lc4mb]
22 DESIGN=FireSimNoNIC
23 TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
24 PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
25 instancetype=c4.4xlarge
26 deploytriplet=None
27
28 # BOOM-based targets
29 [fireboom-singlecore-no-nic-ddr3-1lc4mb]
30 DESIGN=FireBoomNoNIC
31 TARGET_CONFIG=FireSimBoomConfig
32 PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
33 instancetype=c4.4xlarge
34 deploytriplet=None
35
36 [fireboom-singlecore-nic-ddr3-1lc4mb]
37 DESIGN=FireBoom
38 TARGET_CONFIG=FireSimBoomConfig
```

```
0 / mosh-client (tmux) #1
config_build.ini buffers
2 # BUILDTIME/AGFI management configuration for the FireSim Simulation Manager
1 # See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of
  these params.
3
4 [afibuild]
5 s3bucketname=firesim-721179603761
6 buildinstancemarket=ondemand
7 spotinterruptionbehavior=terminate
8 spotmaxprice=ondemand
9
10 [builds]
11 # this section references builds defined in config_build_recipes.ini
12 # if you add a build here, it will be built when you run buildafi
13 firesim-singlecore-no-nic-lbp
14 firesim-quadcore-no-nic-ddr3-1lc4mb
15 firesim-quadcore-nic-ddr3-1lc4mb
16 fireboom-singlecore-no-nic-ddr3-1lc4mb
17 fireboom-singlecore-nic-ddr3-1lc4mb
18
19 [agfistoshare]
20 firesim-singlecore-no-nic-lbp
21 firesim-quadcore-no-nic-ddr3-1lc4mb
22 firesim-quadcore-nic-ddr3-1lc4mb
23 fireboom-singlecore-no-nic-ddr3-1lc4mb
24 fireboom-singlecore-nic-ddr3-1lc4mb
25
26 [sharewithaccounts]
27 somebodyname=123456789012
```



Anatomy of a Build Recipe

Consists of:

```
[firesim-singlecore-no-nic-lbp]  
DESIGN=FireSimNoNIC  
TARGET_CONFIG=FireSimRocketChipSingleCoreConfig  
PLATFORM_CONFIG=FireSimConfig  
instancetype=c4.xlarge  
deploytriplet=None
```

- A label
- The tuple from before
- The EC2 instance type you'd like to build on



Defining a Build Job: config_build.ini

```
1 [afibuild]
2 s3bucketname=firesim-721179603761
3 buildinstancemarket=ondemand
4 spotinterruptionbehavior=terminate
5 spotmaxprice=ondemand
6
7 [builds]
8 # this section references builds defined in config_build_recipes.ini
9 # if you add a build here, it will be built when you run buildafi
10 firesim-singlecore-no-nic-lbp
11 firesim-quadcore-no-nic-ddr3-llc4mb
12 firesim-quadcore-nic-ddr3-llc4mb
13 fireboom-singlecore-no-nic-ddr3-llc4mb
14 fireboom-singlecore-nic-ddr3-llc4mb
15
16 [agfistoshare]
17 firesim-singlecore-no-nic-lbp
18 firesim-quadcore-no-nic-ddr3-llc4mb
19 firesim-quadcore-nic-ddr3-llc4mb
20 fireboom-singlecore-no-nic-ddr3-llc4mb
21 fireboom-singlecore-nic-ddr3-llc4mb
22
23 [sharewithaccounts]
24 somebodyname=123456789012
~
~
```

Consists of:

- More instance configurations
- A list of recipes you'd like to batch out to a build farm

Once you're done with builds:

- A list of recipes you'd like to share with other users

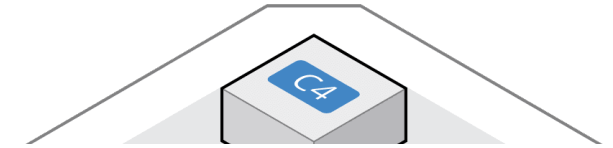


Running builds

- Once we've configured **what** we want to build, let's build it

```
$ firesim buildafi
```

- This completely automates the process. For each design, in-parallel:
 - Launch a build instance (c5.4xlarge)
 - Generate target RTL & invokes Golden Gate
 - Ship infrastructure to build instances, run Vivado FPGA builds in parallel
 - Collect results back onto manager instance
 - \$FDIR/deploy/results-build/<TIMESTAMP>-<tuple>/
 - Email you the entry to put into config_hwdb.ini
 - Terminate the build instance



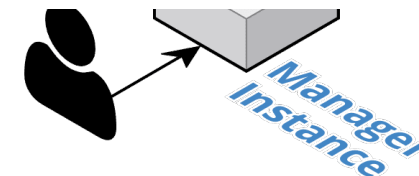
AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

Your AGFI has been created!

Add

```
[firesim-singlecore-sha3-l2-no-nic-ddr3-llc4mb]  
agfi=agfi-0679d5d17ba885886  
deploytripletoverride=None  
customruntimeconfig=None
```

to your config_hwdb.ini to use this hardware configuration.





Interactive:

```
$ cd $FDIR/deploy
```

```
# Should print the FPGA image from the AM
```

```
$ cat built-hwdb-entries/*
```



Anatomy of a HWDB Entry

```
[fireboom-singlecore-no-nic-ddr3-[1c4mb]  
agfi=agfi-0df9101df7b7ff708  
deploytripletoverride=None  
customruntimeconfig=None
```

- Same label as before
- The FPGA image

Hooks to change:

- Software models
- Runtime arguments

→ *Without FPGA recompilation*



Interactive:

```
# Prefetching for the next section  
$ cd ~/chipyard-afternoon/software/firemarshal/  
$ ./marshal -d build workloads/sha3-linux-test.json
```



Summary

- Don't fret if you didn't catch everything, everything we showed you today is documented in excruciating detail at <http://docs.fires.im>
- We learned how to:
 - Build FireSim FPGA images for a set of targets
 - <http://docs.fires.im/en/latest/Building-a-FireSim-AFI.html>



Backup Slides



Interactive:

```
$ cd $FDIR/deploy
```

```
$ cd results-build/<name>/cl_firesim
```

```
$ ls
```



Captured Build Outputs

`design/`

- The source files for the build;

`build/scripts/<timestamp>.vivado.log`

- Log of the entire vivado build process

`build/reports/`

- Timing and utilization reports from various stages

`build/checkpoints/`

- Design checkpoints (*.dcp); can reopen in Vivado to debug a build



Interactive:

```
cd $FDIR/sim
```

```
make run-asm-tests
```

```
<wait for one test to complete> ctrl-c
```



Simulating the Simulator

- Can simulate Golden Gate's output *without* doing an FPGA-build
 - Runs with all the same models you'd have on the FPGA
 - Should produce target-cycle-exact behavior as an FPGA simulation
- outputs in output/f1/<tuple>