FireSim

A Brief Tour of FireSim: The Manager & Compiler; Building an FPGA image

https://fires.im

@firesimproject

ISCA 2021 Tutorial

Speaker: Sagar Karandikar

Berkeley Architecture Research

Agenda: What will we cover?

1) The Compiler → Golden Gate

- Invoke it on example RTL
- "Simulate the simulator" using Verilator

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/sims/firesim
```

→ We will reference this as `$FDIR`

Example commands:

```
$ 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
- "Simulate the simulator" using Verilator

2) The Manager → `firesim`

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

Example commands:

```
$ cd $FDIR/sim

$ make DESIGN=FireSim
```

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

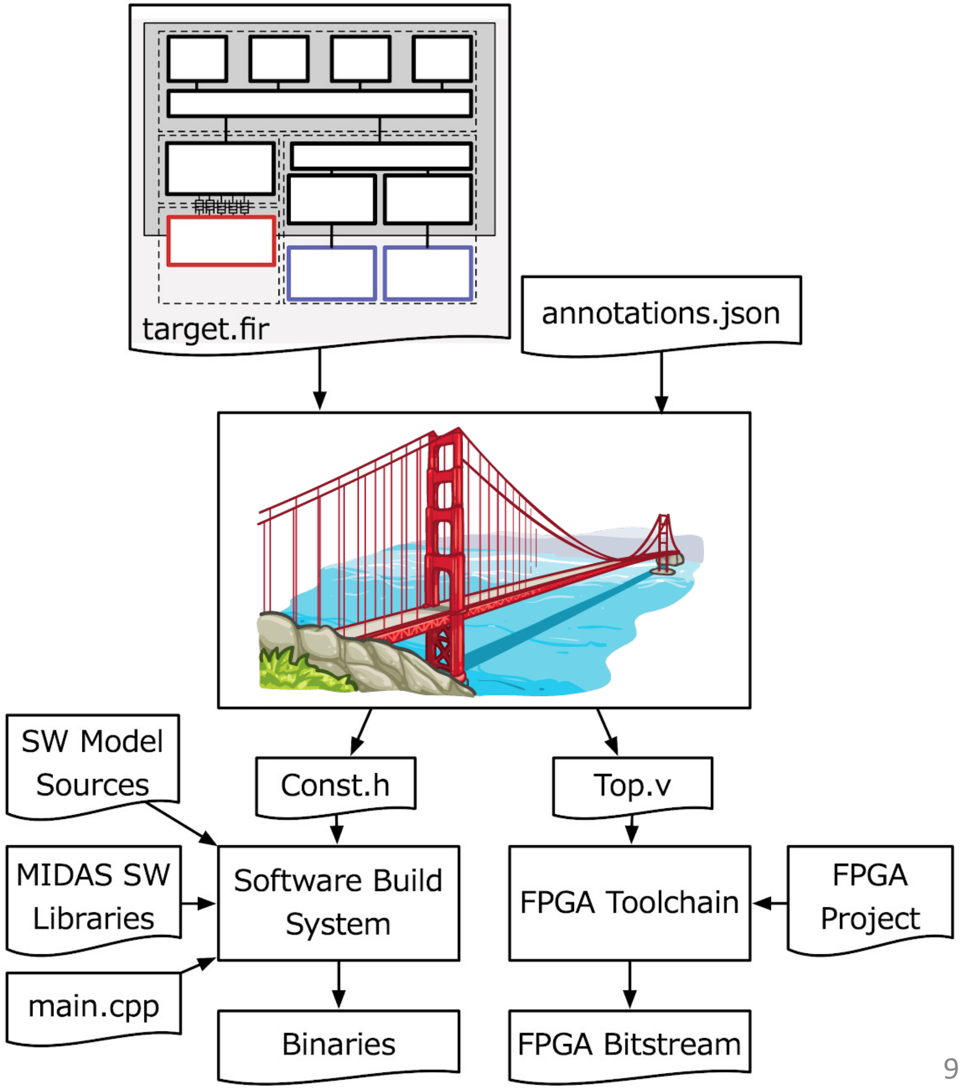
Golden Gate Compiler

Inputs:

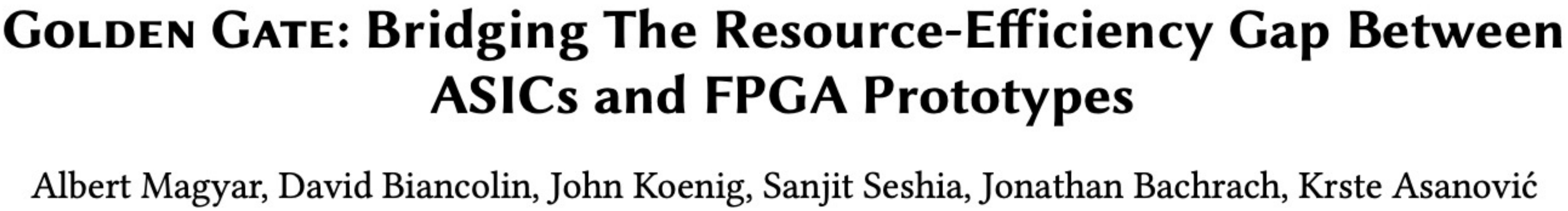
- FIRRTL & annos from a Chipyard generator
- Compiler configuration

→ 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"

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

Example commands:

```
$ 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?

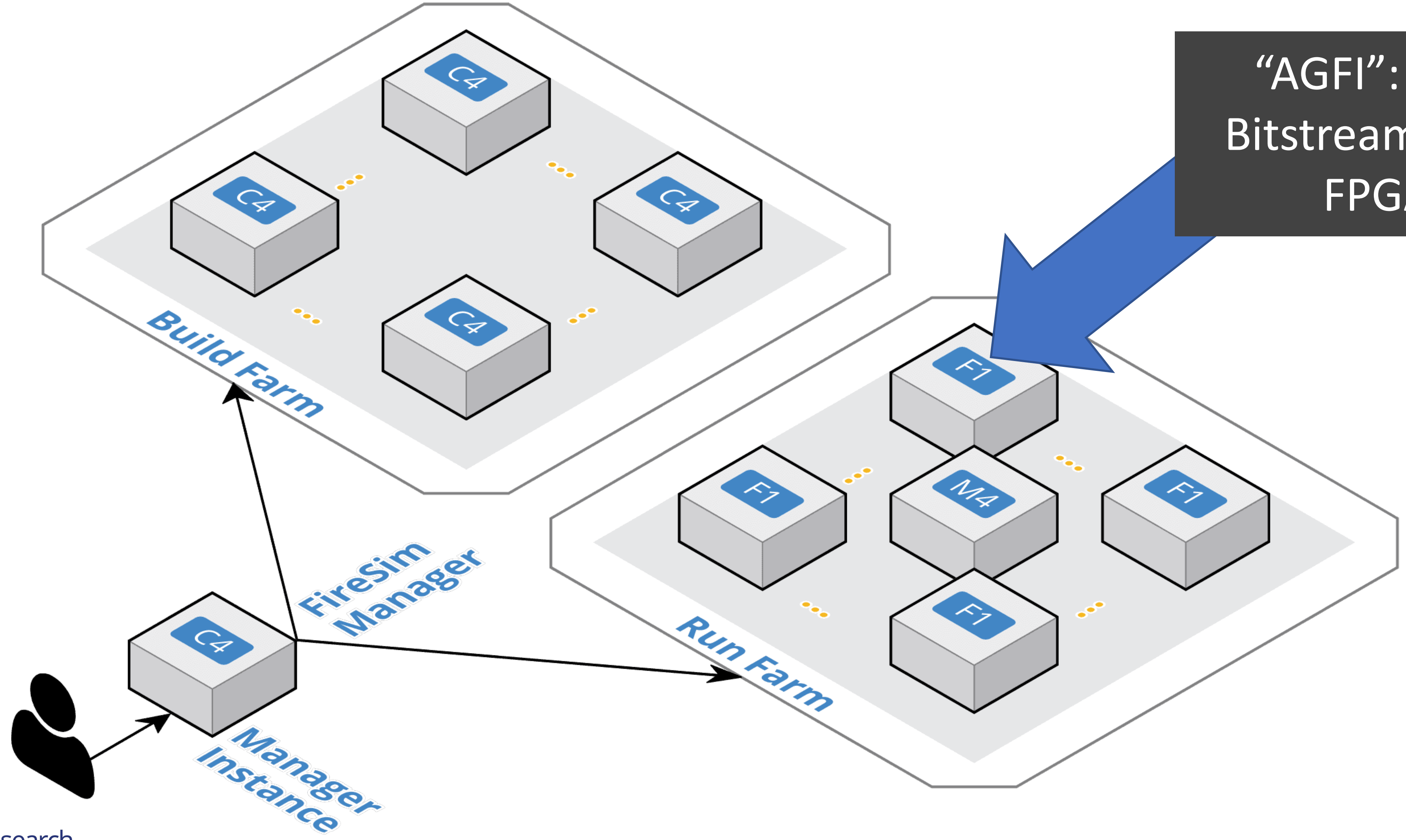
1) The Compiler → Golden Gate

- Invoke it on example RTL
- Simulate the output in an RTL simulator

2) The Manager → `firesim`

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

Background Terminology



"AGFI": FPGA Bitstream for F1 FPGAs

Using the `firesim` manager command line

- sourcing `$FDIR/sourceme-f1-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:

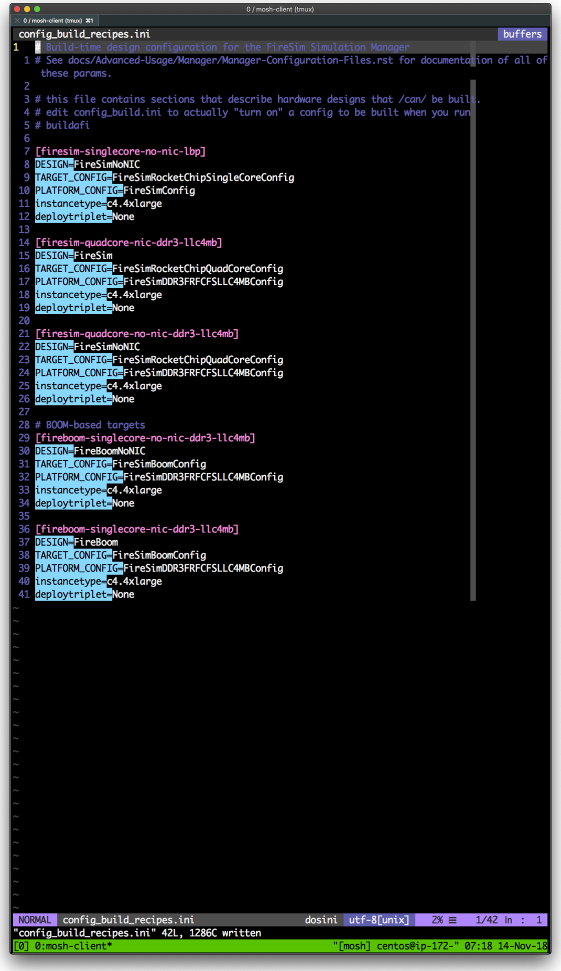
```
firesim managerinit
```

Example commands:

```
$ cd $FDIR/deploy

$ ls
```

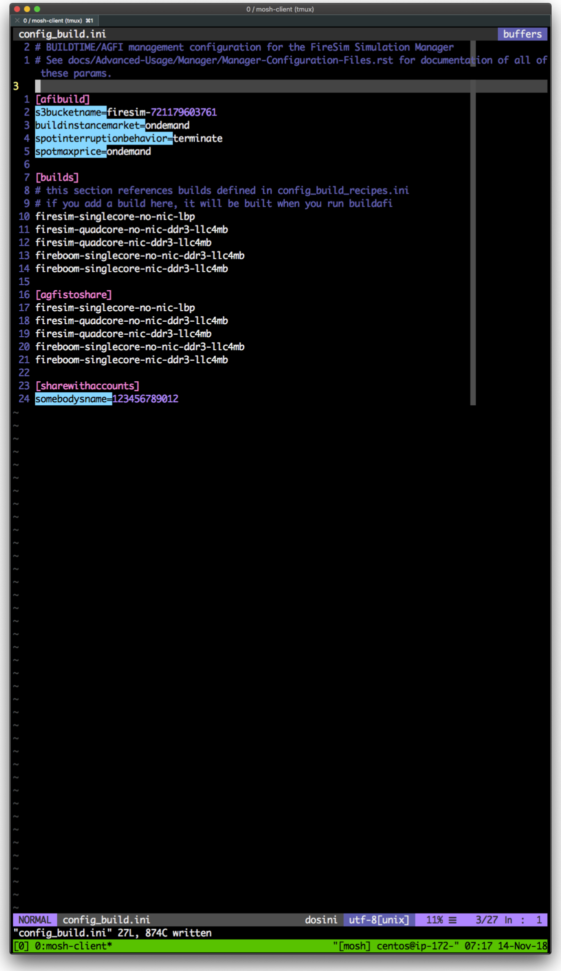
Configuring the Manager. 4 files in `firesim/deploy/`

`config_build_recipes.ini`



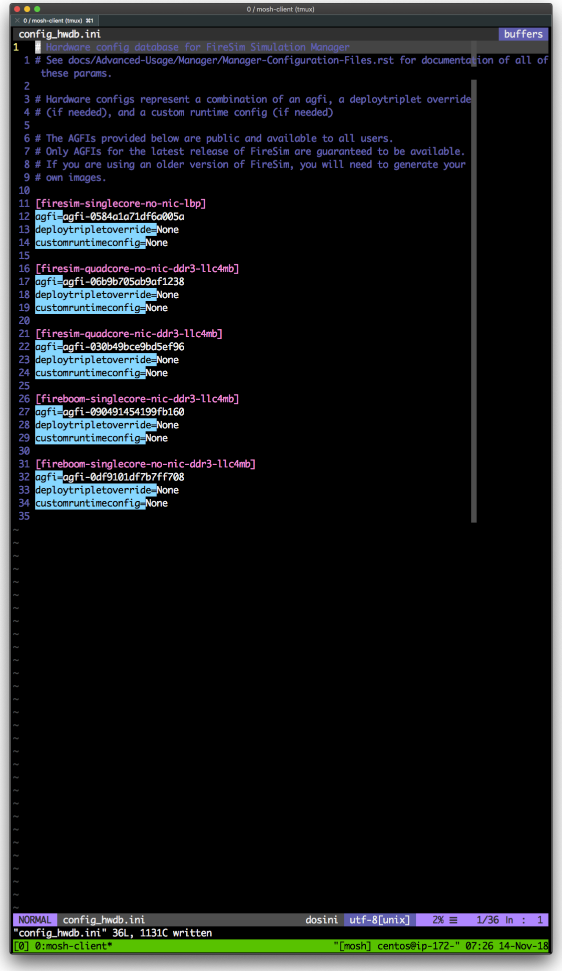
```
0 / mosh-client (tmux)
config_build_recipes.ini                                                buffers
# Build-time design configuration for the FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

# this file contains sections that describe hardware designs that /can/ be built.
# edit config_build.ini to actually "turn on" a config to be built when you run
# buildafi

[firesim-singlecore-no-nic-lbp]
DESIGN=FireSimNoNIC
TARGET_CONFIG=FireSimRocketChipSingleCoreConfig
PLATFORM_CONFIG=FireSimConfig
instancetype=c4.4xlarge
deploytriplet=None

[firesim-quadcore-nic-ddr3-llc4mb]
DESIGN=FireSim
TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

[firesim-quadcore-no-nic-ddr3-llc4mb]
DESIGN=FireSimNoNIC
TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

# BOOM-based targets
[fireboom-singlecore-no-nic-ddr3-llc4mb]
DESIGN=FireBoomNoNIC
TARGET_CONFIG=FireSimBoomConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

[fireboom-singlecore-nic-ddr3-llc4mb]
DESIGN=FireBoom
TARGET_CONFIG=FireSimBoomConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

NORMAL  config_build_recipes.ini        dosini  utf-8[unix]  2% ≡  1/42 ln : 1
"config_build_recipes.ini" 42L, 1286C written
[0] 0:mosh-client*                    "[mosh] centos@ip-172-" 07:18 14-Nov-18
```

`config_build.ini`



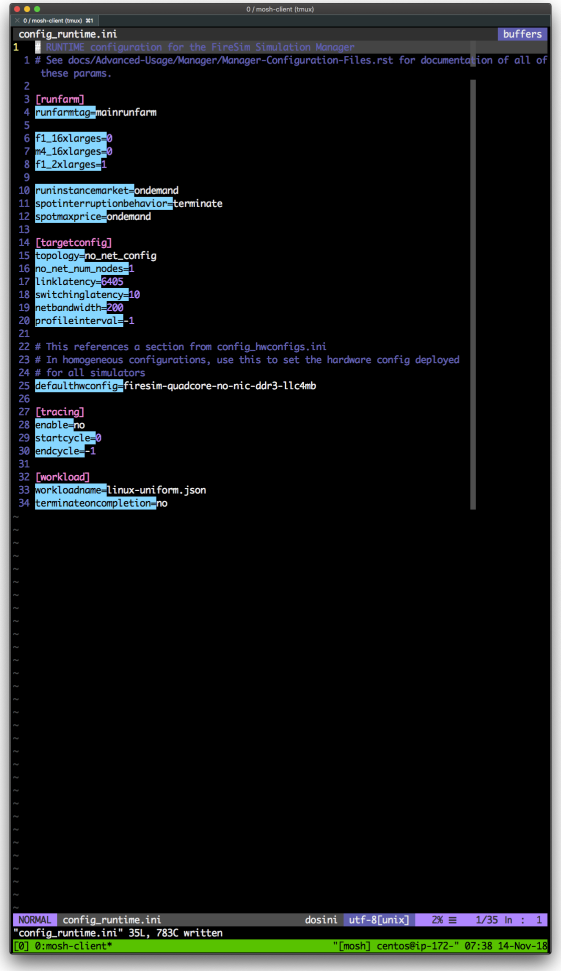
```
0 / mosh-client (tmux)
config_build.ini                                                        buffers
# BUILDTIME/AGFI management configuration for the FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

[afibuild]
s3bucketname=firesim-721179603761
buildinstancemarket=ondemand
spotinterruptionbehavior=terminate
spotmaxprice=ondemand

[builds]
# this section references builds defined in config_build_recipes.ini
# if you add a build here, it will be built when you run buildafi
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[agfistoshare]
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[sharewithaccounts]
somebodysname=123456789012

NORMAL  config_build.ini        dosini  utf-8[unix]  11% ≡  3/27 ln : 1
"config_build.ini" 27L, 874C written
[0] 0:mosh-client*                    "[mosh] centos@ip-172-" 07:17 14-Nov-18
```

`config_hwdb.ini`



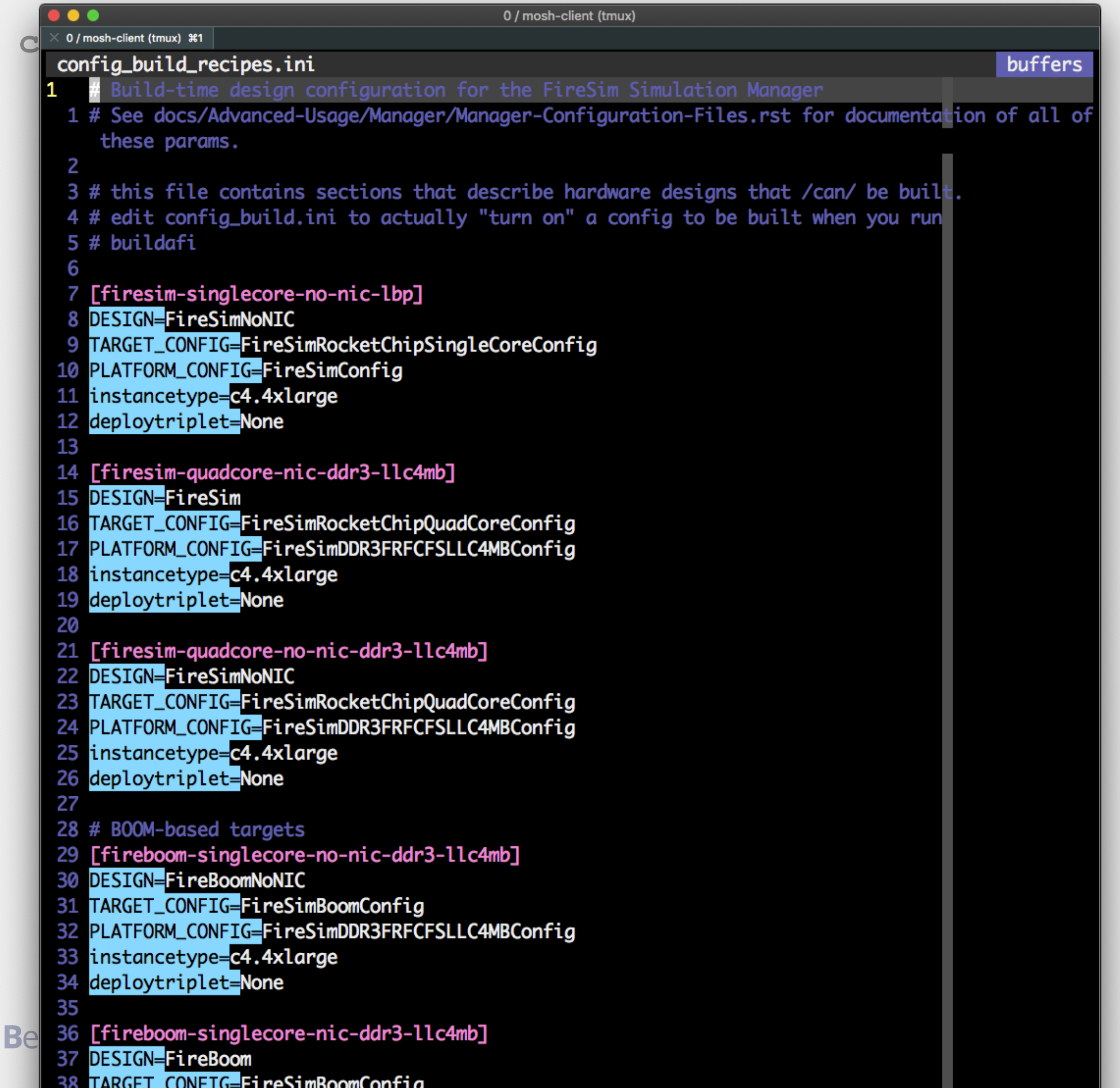
```
0 / mosh-client (tmux)
config_hwdb.ini                                                         buffers
# Hardware config database for FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

# Hardware configs represent a combination of an agfi, a deploytriplet override
# (if needed), and a custom runtime config (if needed)

# The AGFIs provided below are public and available to all users.
# Only AGFIs for the latest release of FireSim are guaranteed to be available.
# If you are using an older version of FireSim, you will need to generate your
# own images.

[firesim-singlecore-no-nic-lbp]
agfi=agfi-0584a1a71df6a005a
deploytripletoverride=None
customruntimeconfig=None

[firesim-quadcore-no-nic-ddr3-llc4mb]
agfi=agfi-06b9b705ab9af1238
deploytripletoverride=None
customruntimeconfig=None

[firesim-quadcore-nic-ddr3-llc4mb]
agfi=agfi-030b49bce9bd5ef96
deploytripletoverride=None
customruntimeconfig=None

[fireboom-singlecore-nic-ddr3-llc4mb]
agfi=agfi-090491454199fb160
deploytripletoverride=None
customruntimeconfig=None

[fireboom-singlecore-no-nic-ddr3-llc4mb]
agfi=agfi-0df9101df7b7ff708
deploytripletoverride=None
customruntimeconfig=None


NORMAL  config_hwdb.ini        dosini  utf-8[unix]  2% ≡  1/36 ln : 1
"config_hwdb.ini" 36L, 1131C written
[0] 0:mosh-client*                    "[mosh] centos@ip-172-" 07:26 14-Nov-18
```

`config_runtime.ini`

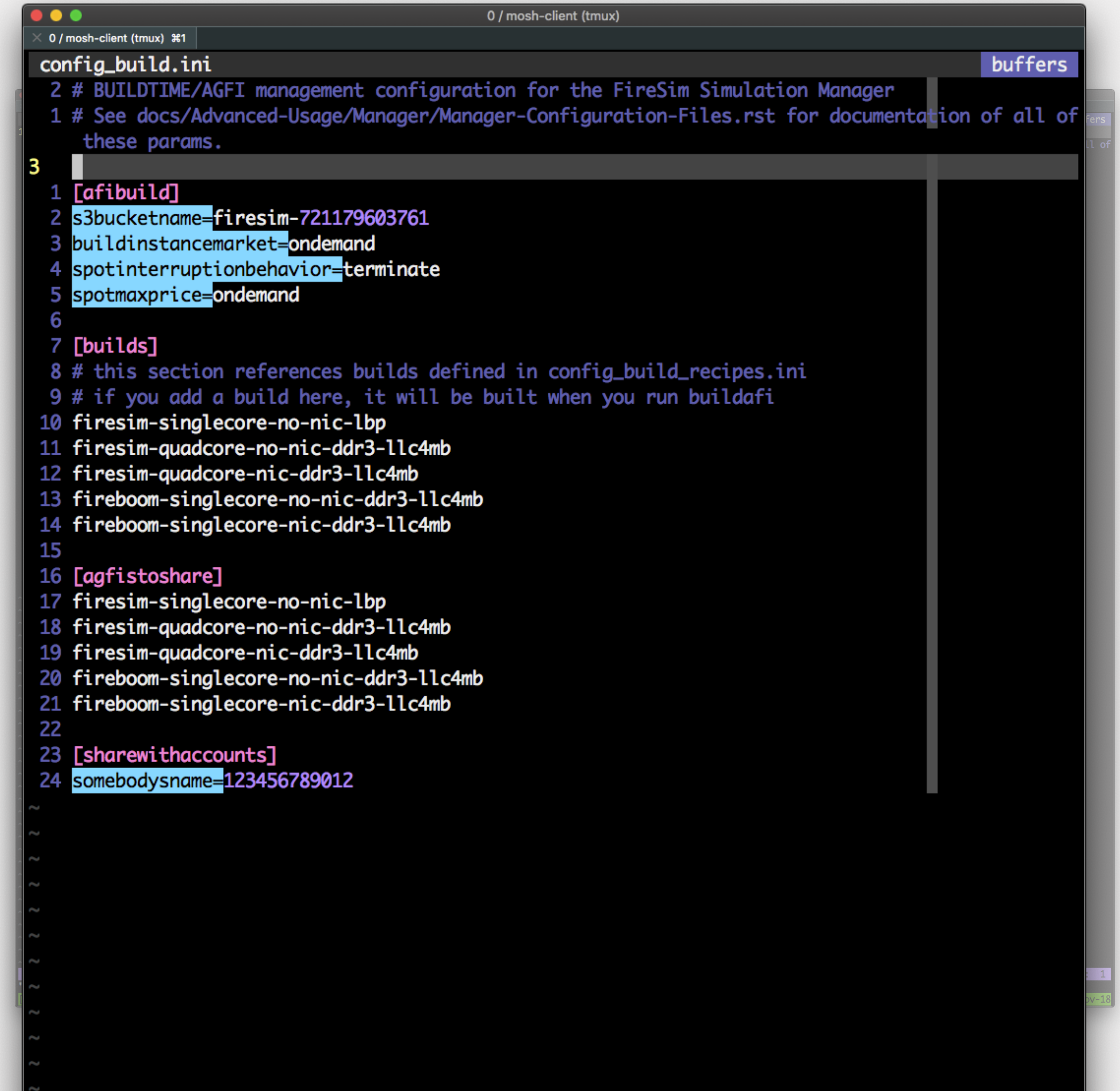


```
0 / mosh-client (tmux)
config_runtime.ini                                                      buffers
# RUNTIME configuration for the FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

[runfarm]
runfarmtag=mainrunfarm

f1_16xlarges=0
m4_16xlarges=0
f1_2xlarges=1

runinstancemarket=ondemand
spotinterruptionbehavior=terminate
spotmaxprice=ondemand

[targetconfig]
topology=no_net_config
no_net_num_nodes=1
linklatency=6405
switchinglatency=10
netbandwidth=200
profileinterval=-1

# This references a section from config_hwconfigs.ini
# In homogeneous configurations, use this to set the hardware config deployed
# for all simulators
defaulthwconfig=firesim-quadcore-no-nic-ddr3-llc4mb

[tracing]
enable=no
startcycle=0
endcycle=-1

[workload]
workloadname=linux-uniform.json
terminateoncompletion=no

NORMAL  config_runtime.ini        dosini  utf-8[unix]  2% ≡  1/35 ln : 1
"config_runtime.ini" 35L, 783C written
[0] 0:mosh-client*                    "[mosh] centos@ip-172-" 07:38 14-Nov-18
```

Configuring a Build

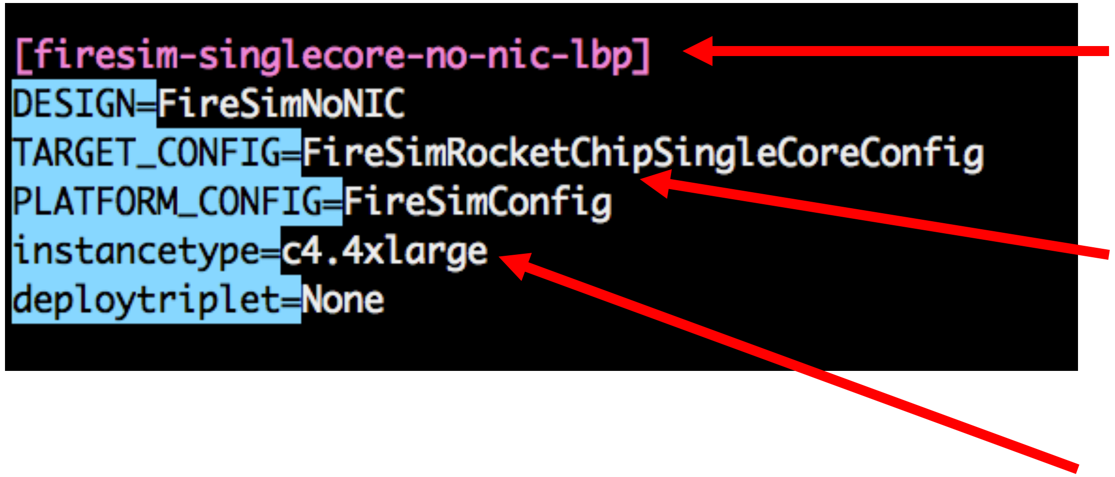


```
# Build-time design configuration for the FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

# this file contains sections that describe hardware designs that /can/ be built.
# edit config_build.ini to actually "turn on" a config to be built when you run
# buildafi

[firesim-singlecore-no-nic-lbp]
DESIGN=FireSimNoNIC
TARGET_CONFIG=FireSimRocketChipSingleCoreConfig
PLATFORM_CONFIG=FireSimConfig
instancetype=c4.4xlarge
deploytriplet=None

[firesim-quadcore-nic-ddr3-llc4mb]
DESIGN=FireSim
TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

[firesim-quadcore-no-nic-ddr3-llc4mb]
DESIGN=FireSimNoNIC
TARGET_CONFIG=FireSimRocketChipQuadCoreConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

# BOOM-based targets
[fireboom-singlecore-no-nic-ddr3-llc4mb]
DESIGN=FireBoomNoNIC
TARGET_CONFIG=FireSimBoomConfig
PLATFORM_CONFIG=FireSimDDR3FRFCFSLLC4MBConfig
instancetype=c4.4xlarge
deploytriplet=None

[fireboom-singlecore-nic-ddr3-llc4mb]
DESIGN=FireBoom
TARGET_CONFIG=FireSimBoomConfig
```



```
# BUILDTIME/AGFI management configuration for the FireSim Simulation Manager
# See docs/Advanced-Usage/Manager/Manager-Configuration-Files.rst for documentation of all of these params.

[afibuild]
s3bucketname=firesim-721179603761
buildinstancemarket=ondemand
spotinterruptionbehavior=terminate
spotmaxprice=ondemand

[builds]
# this section references builds defined in config_build_recipes.ini
# if you add a build here, it will be built when you run buildafi
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[agfistoshare]
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[sharewithaccounts]
somebodysname=123456789012
~
~
~
~
~
~
~
~
~
~
~
```

Anatomy of a Build Recipe

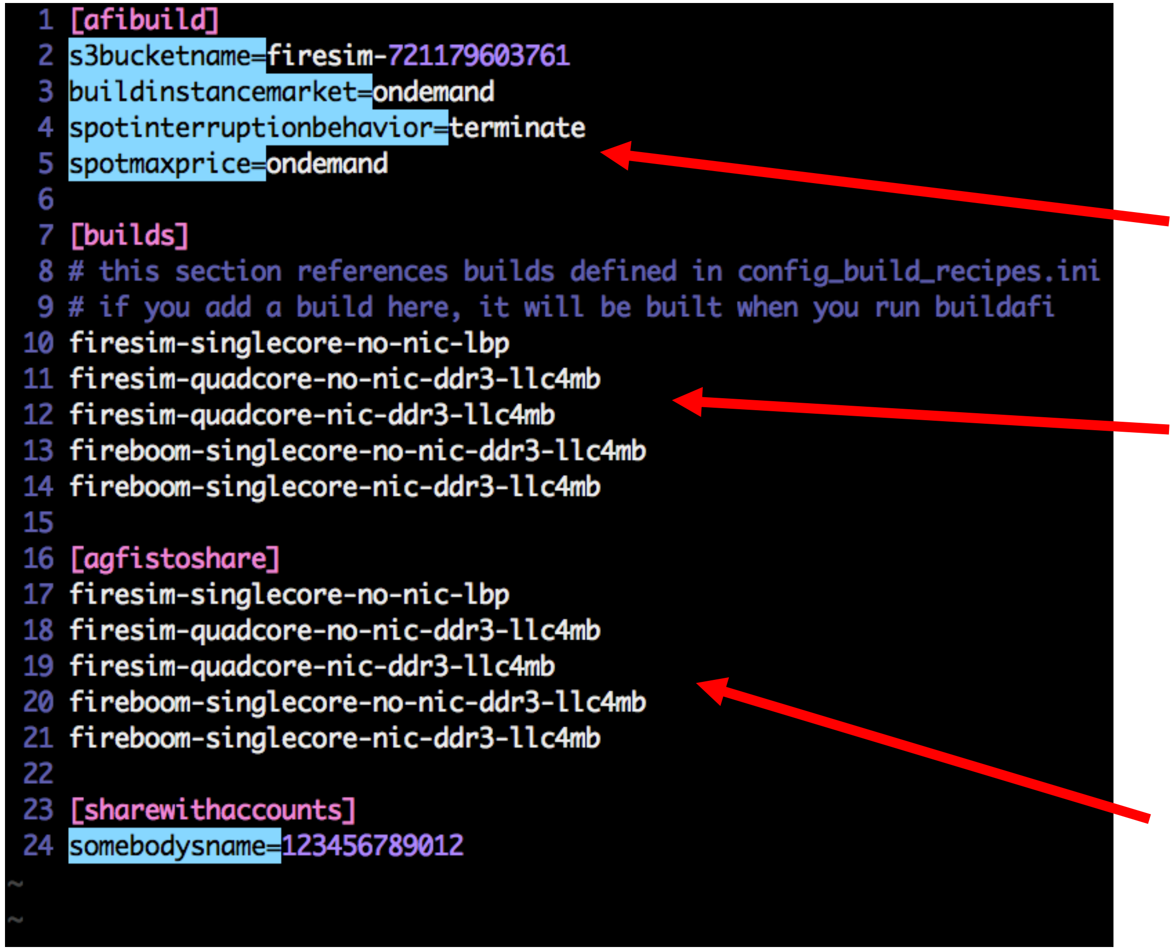


```
[firesim-singlecore-no-nic-lbp]
DESIGN=FireSimNoNIC
TARGET_CONFIG=FireSimRocketChipSingleCoreConfig
PLATFORM_CONFIG=FireSimConfig
instancetype=c4.4xlarge
deploytriplet=None
```

Consists of:

- A label
- The tuple from before
- The EC2 instance type you’d like to use

Defining a Build Job: `config_build.ini`



```
[afibuild]
s3bucketname=firesim-721179603761
buildinstancemarket=ondemand
spotinterruptionbehavior=terminate
spotmaxprice=ondemand

[builds]
# this section references builds defined in config_build_recipes.ini
# if you add a build here, it will be built when you run buildafi
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[agfistoshare]
firesim-singlecore-no-nic-lbp
firesim-quadcore-no-nic-ddr3-llc4mb
firesim-quadcore-nic-ddr3-llc4mb
fireboom-singlecore-no-nic-ddr3-llc4mb
fireboom-singlecore-nic-ddr3-llc4mb

[sharewithaccounts]
somebodysname=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:

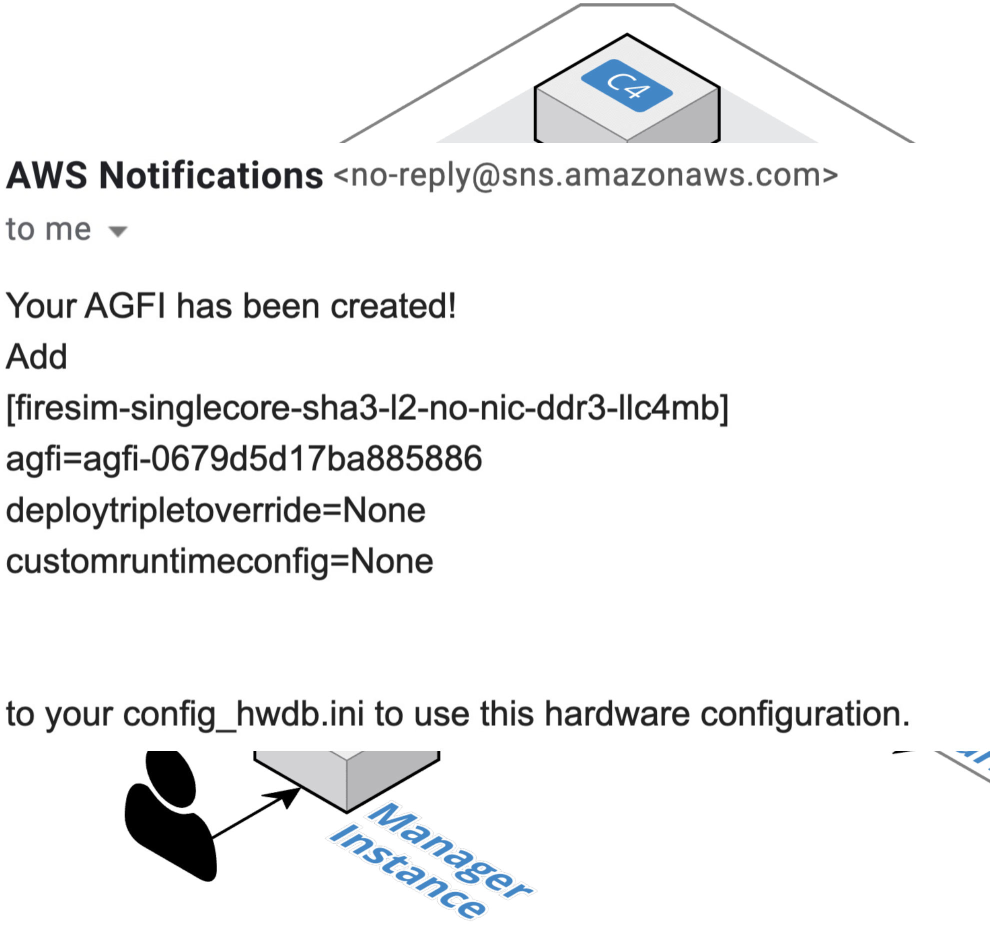
- A list of recipes you’d like to share with other users

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
    - 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.

Example commands:

```
$ 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

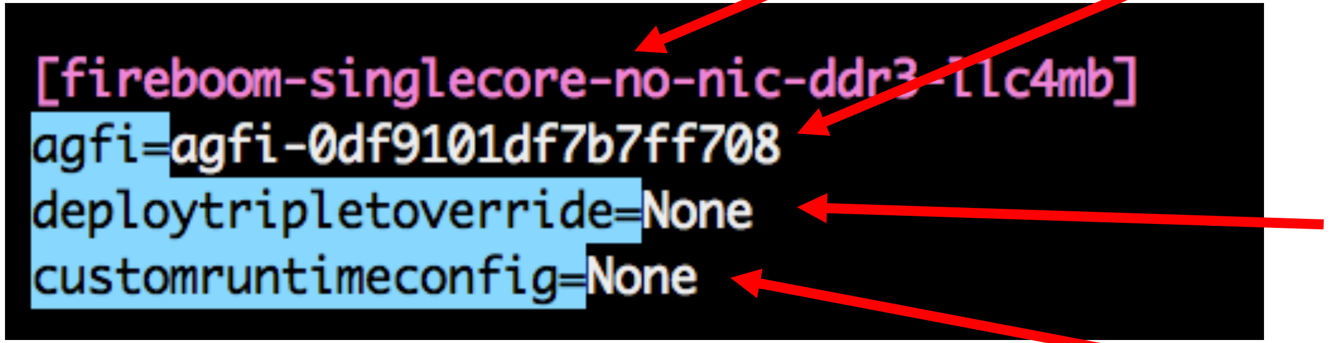
`build/checkpoints/`

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

Example commands:

```
$ cd $FDIR/deploy

$ cat built-hwdb-entries/* >> config_hwdb.ini

$ tail config_hwdb.ini
```

Anatomy of a HWDB Entry



```
[fireboom-singlecore-no-nic-ddr3-llc4mb]
agfi=agfi-0df9101df7b7ff708
deploytripletoverride=None
customruntimeconfig=None
```

- Same label as before
- The FPGA image

Hooks to change:

- Software models
- Runtime arguments

→ *Without FPGA recompilation*

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>

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