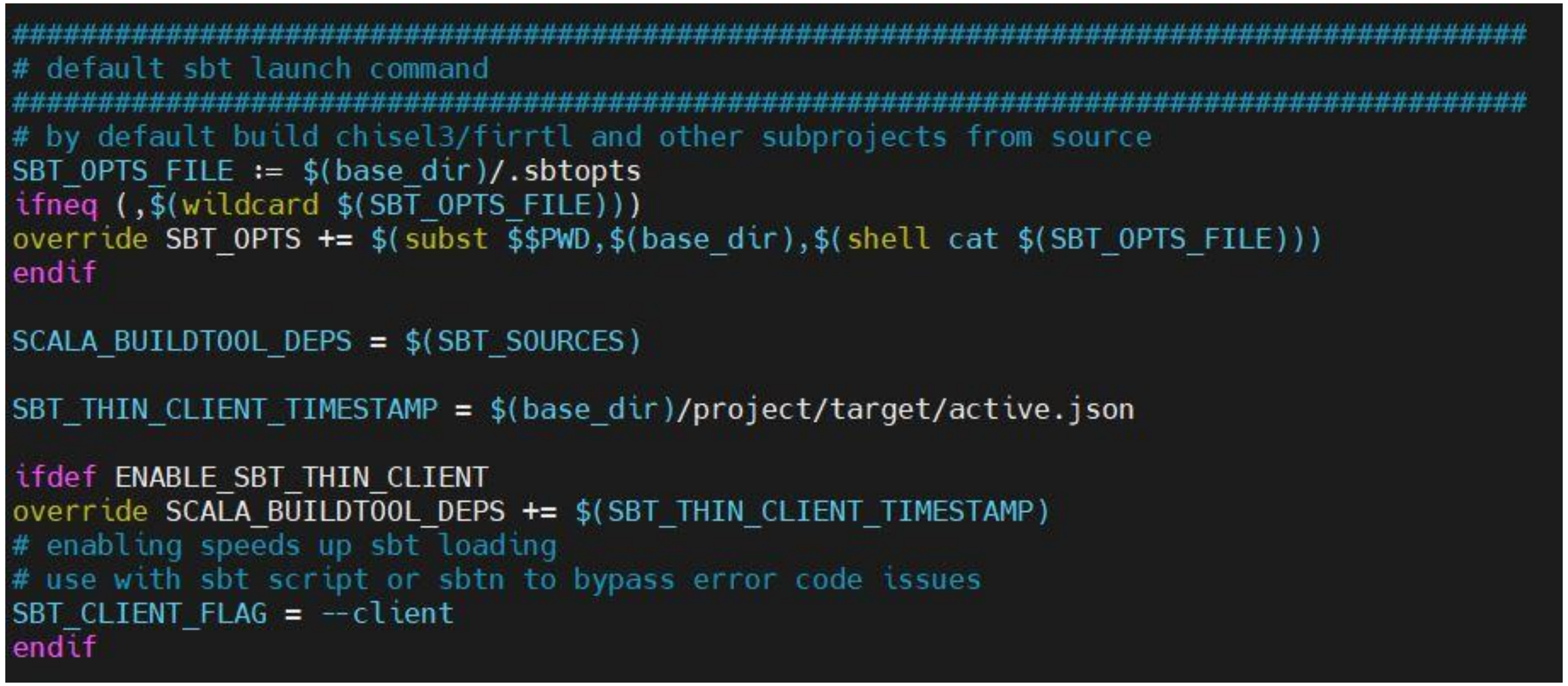
chipyard 환경에서의 오류 수정

권형서 김종엽 정지용

(23/01/25 기준)

1. /chipyard/variables.mk 파일 수정

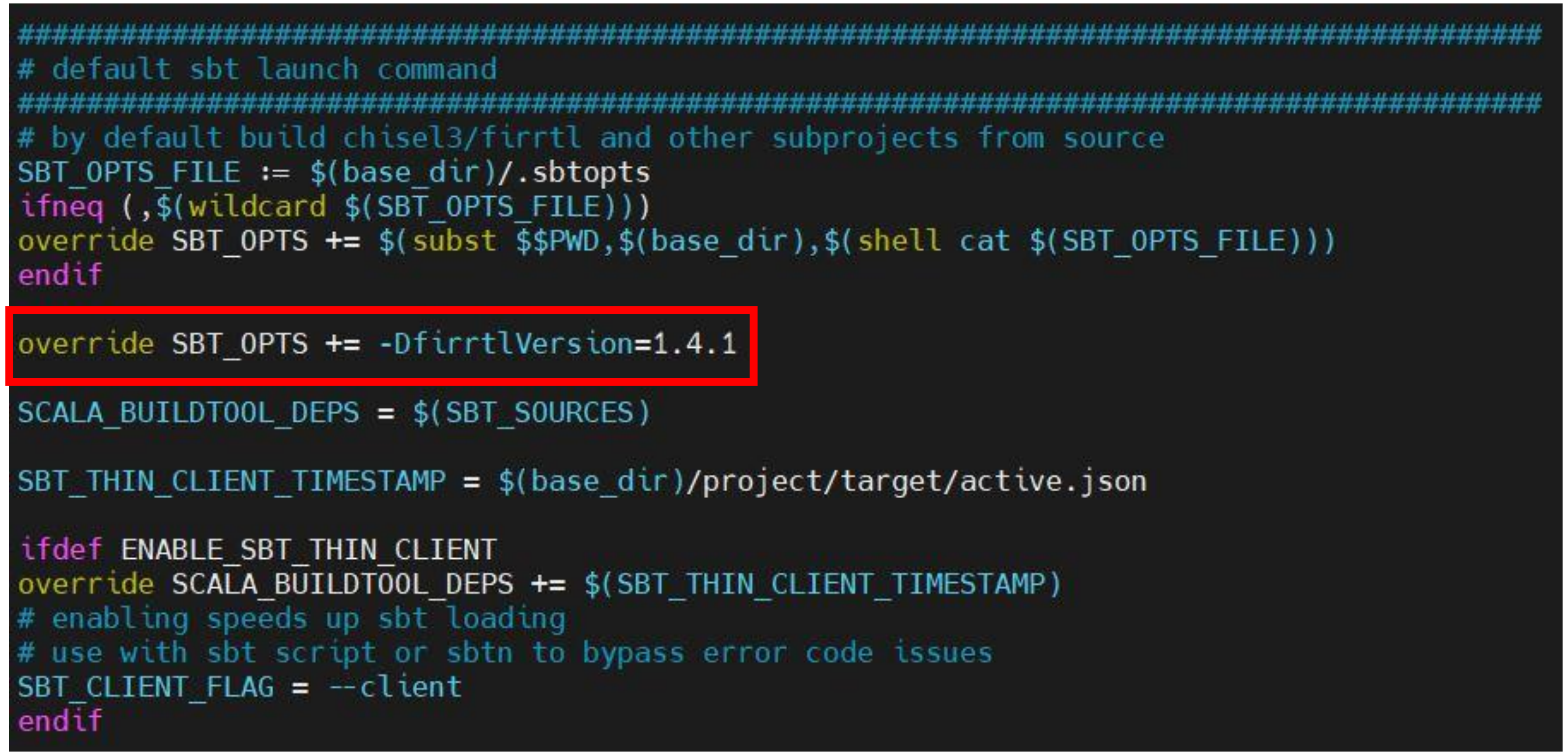
Before



```
#########################################################################################
# default sbt launch command
#########################################################################################
# by default build chisel3/firrtl and other subprojects from source
SBT_OPTS_FILE := $(base_dir)/.sbtopts
ifneq (,$(wildcard $(SBT_OPTS_FILE)))
override SBT_OPTS += $(subst $$PWD,$(base_dir),$(shell cat $(SBT_OPTS_FILE)))
endif

SCALA_BUILDTOOL_DEPS = $(SBT_SOURCES)

SBT_THIN_CLIENT_TIMESTAMP = $(base_dir)/project/target/active.json

ifdef ENABLE_SBT_THIN_CLIENT
override SCALA_BUILDTOOL_DEPS += $(SBT_THIN_CLIENT_TIMESTAMP)
# enabling speeds up sbt loading
# use with sbt script or sbtn to bypass error code issues
SBT_CLIENT_FLAG = --client
endif
```

1. /chipyard/variables.mk 파일 수정

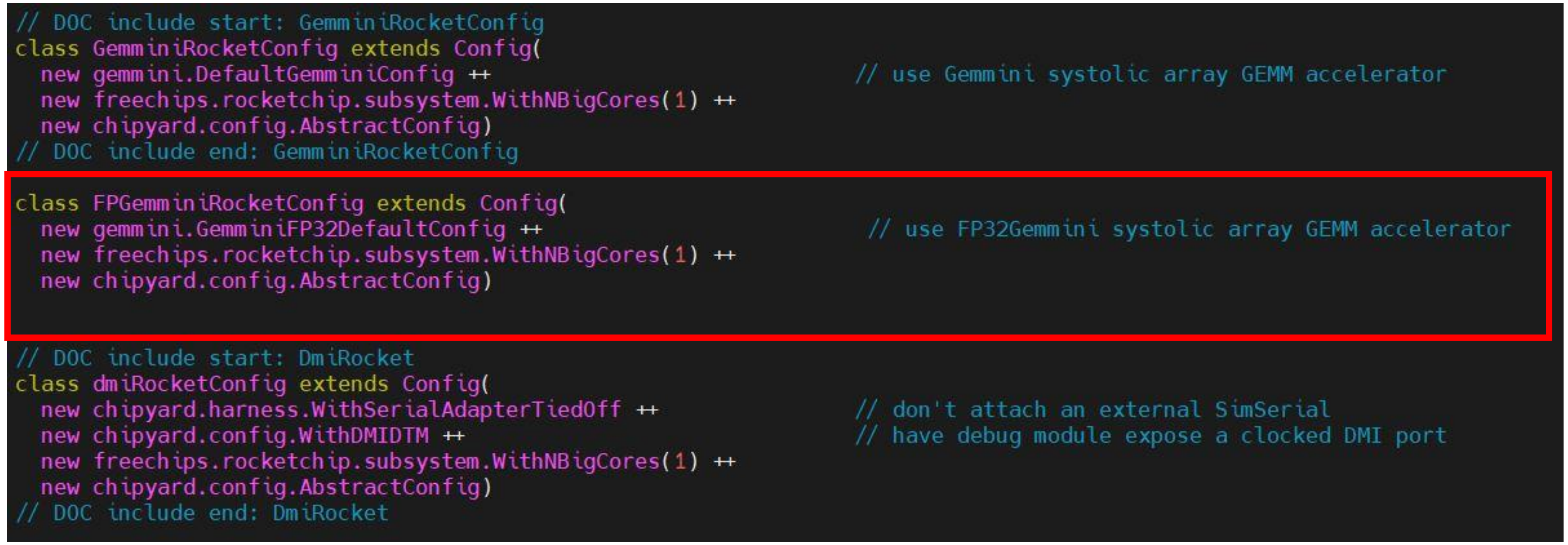
After



```
#########################################################################################
# default sbt launch command
#########################################################################################
# by default build chisel3/firrtl and other subprojects from source
SBT_OPTS_FILE := $(base_dir)/.sbtopts
ifneq (,$(wildcard $(SBT_OPTS_FILE)))
override SBT_OPTS += $(subst $$PWD,$(base_dir),$(shell cat $(SBT_OPTS_FILE)))
endif

override SBT_OPTS += -DfirrtlVersion=1.4.1

SCALA_BUILDTOOL_DEPS = $(SBT_SOURCES)

SBT_THIN_CLIENT_TIMESTAMP = $(base_dir)/project/target/active.json

ifdef ENABLE_SBT_THIN_CLIENT
override SCALA_BUILDTOOL_DEPS += $(SBT_THIN_CLIENT_TIMESTAMP)
# enabling speeds up sbt loading
# use with sbt script or sbtn to bypass error code issues
SBT_CLIENT_FLAG = --client
endif
```

2. RocketConfigs.scala 파일 수정

/chipyard/generators/chipyard/src/main/scala/config/RocketConfigs.scala 파일 수정

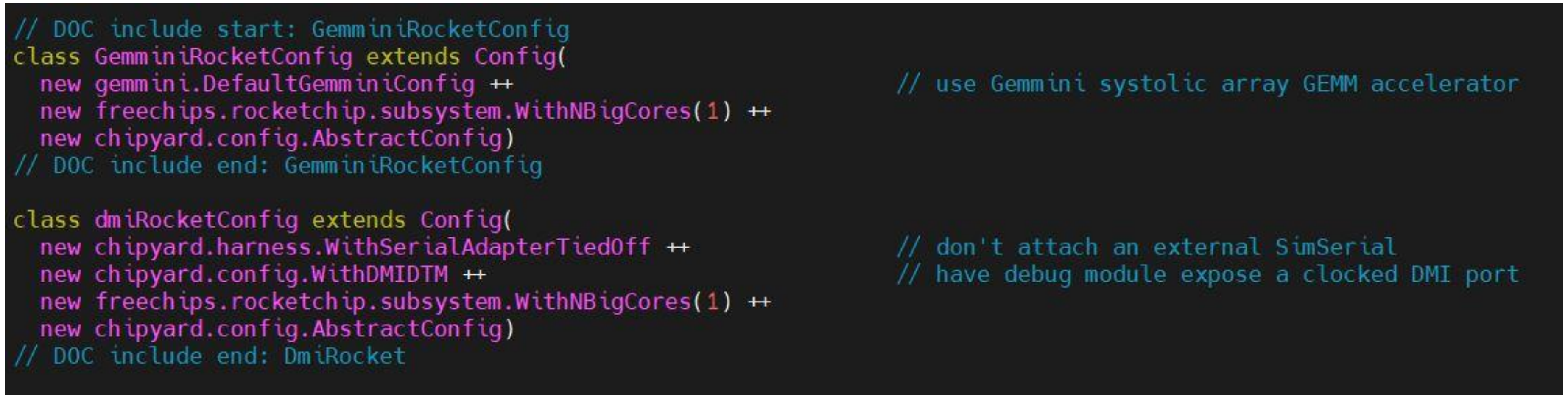


```
// DOC include start: GemminiRocketConfig
class GemminiRocketConfig extends Config(
  new gemmini.DefaultGemminiConfig ++                          // use Gemmini systolic array GEMM accelerator
  new freechips.rocketchip.subsystem.WithNBigCores(1) ++
  new chipyard.config.AbstractConfig)
// DOC include end: GemminiRocketConfig

class FPGemminiRocketConfig extends Config(
  new gemmini.GemminiFP32DefaultConfig ++                       // use FP32Gemmini systolic array GEMM accelerator
  new freechips.rocketchip.subsystem.WithNBigCores(1) ++
  new chipyard.config.AbstractConfig)

// DOC include start: DmiRocket
class dmiRocketConfig extends Config(
  new chipyard.harness.WithSerialAdapterTiedOff ++             // don't attach an external SimSerial
  new chipyard.config.WithDMIDTM ++                             // have debug module expose a clocked DMI port
  new freechips.rocketchip.subsystem.WithNBigCores(1) ++
  new chipyard.config.AbstractConfig)
// DOC include end: DmiRocket
```

Remove

2. RocketConfigs.scala 파일 수정

/chipyard/generators/chipyard/src/main/scala/config/RocketConfigs.scala 파일 수정

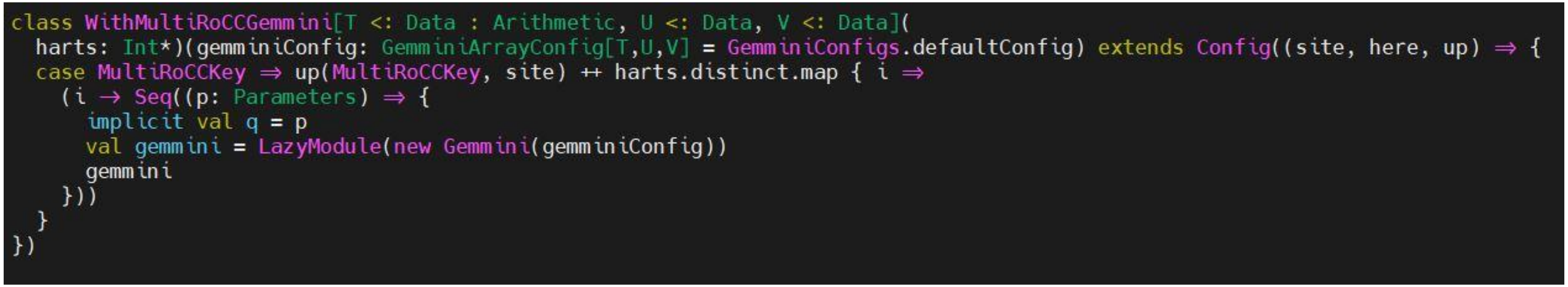


```
// DOC include start: GemminiRocketConfig
class GemminiRocketConfig extends Config(
  new gemmini.DefaultGemminiConfig ++                          // use Gemmini systolic array GEMM accelerator
  new freechips.rocketchip.subsystem.WithNBigCores(1) ++
  new chipyard.config.AbstractConfig)
// DOC include end: GemminiRocketConfig

class dmiRocketConfig extends Config(
  new chipyard.harness.WithSerialAdapterTiedOff ++             // don't attach an external SimSerial
  new chipyard.config.WithDMIDTM ++                            // have debug module expose a clocked DMI port
  new freechips.rocketchip.subsystem.WithNBigCores(1) ++
  new chipyard.config.AbstractConfig)
// DOC include end: DmiRocket
```

3. ConfigFragments.scala 파일 수정

/chipyard/generators/chipyard/src/main/scala/ConfigFragments.scala 파일 수정

Before

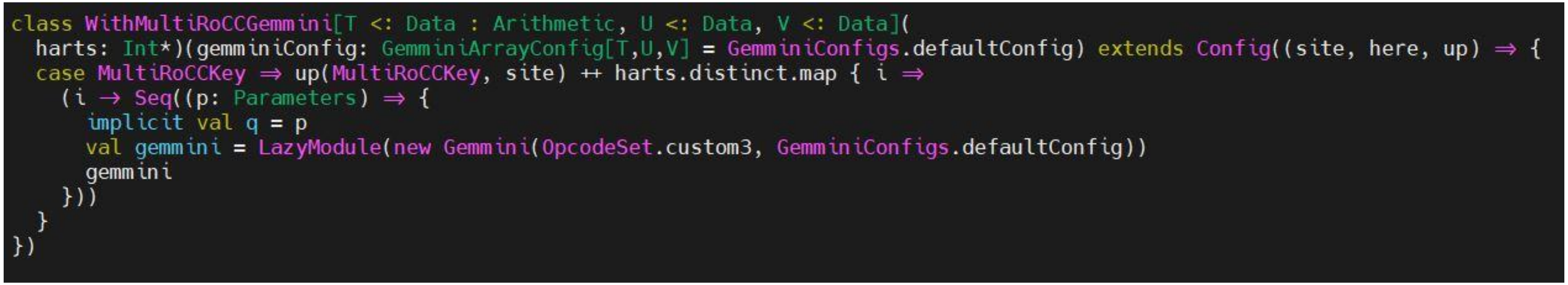


```
class WithMultiRoCCGemmini[T <: Data : Arithmetic, U <: Data, V <: Data](
  harts: Int*)(gemminiConfig: GemminiArrayConfig[T,U,V] = GemminiConfigs.defaultConfig) extends Config((site, here, up) => {
  case MultiRoCCKey => up(MultiRoCCKey, site) ++ harts.distinct.map { i =>
    (i -> Seq((p: Parameters) => {
      implicit val q = p
      val gemmini = LazyModule(new Gemmini(gemminiConfig))
      gemmini
    }))
  }
})
```

3. ConfigFragments.scala 파일 수정

/chipyard/generators/chipyard/src/main/scala/ConfigFragments.scala 파일 수정

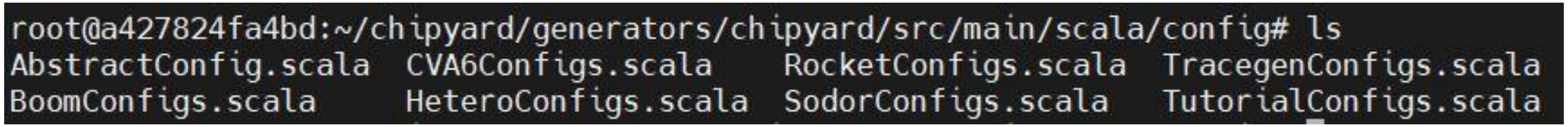
After



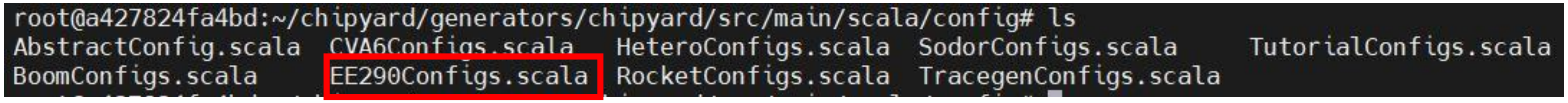
```
class WithMultiRoCCGemmini[T <: Data : Arithmetic, U <: Data, V <: Data](
  harts: Int*)(gemminiConfig: GemminiArrayConfig[T,U,V] = GemminiConfigs.defaultConfig) extends Config((site, here, up) => {
  case MultiRoCCKey => up(MultiRoCCKey, site) ++ harts.distinct.map { i =>
    (i -> Seq((p: Parameters) => {
      implicit val q = p
      val gemmini = LazyModule(new Gemmini(OpcodeSet.custom3, GemminiConfigs.defaultConfig))
      gemmini
    }))
  }
})
```

4. EE290Configs.scala 파일 만들어주기

chipyard/generators/chipyard/src/main/scala/config/
경로에 EE290Configs.scala 파일 만들어 줘야함



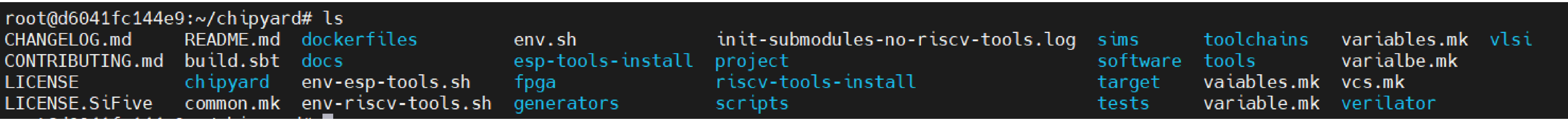
```
root@a427824fa4bd:~/chipyard/generators/chipyard/src/main/scala/config# ls
AbstractConfig.scala  CVA6Configs.scala    RocketConfigs.scala  TracegenConfigs.scala
BoomConfigs.scala     HeteroConfigs.scala  SodorConfigs.scala   TutorialConfigs.scala
```



```
root@a427824fa4bd:~/chipyard/generators/chipyard/src/main/scala/config# ls
AbstractConfig.scala  CVA6Configs.scala    HeteroConfigs.scala  SodorConfigs.scala     TutorialConfigs.scala
BoomConfigs.scala     EE290Configs.scala   RocketConfigs.scala  TracegenConfigs.scala
```

https://github.com/ucb-bar/chipyard/blob/ee290-sp21-labs/generators/chipyard/src/main/scala/config/EE290Configs.scala

5. common.mk 파일 수정



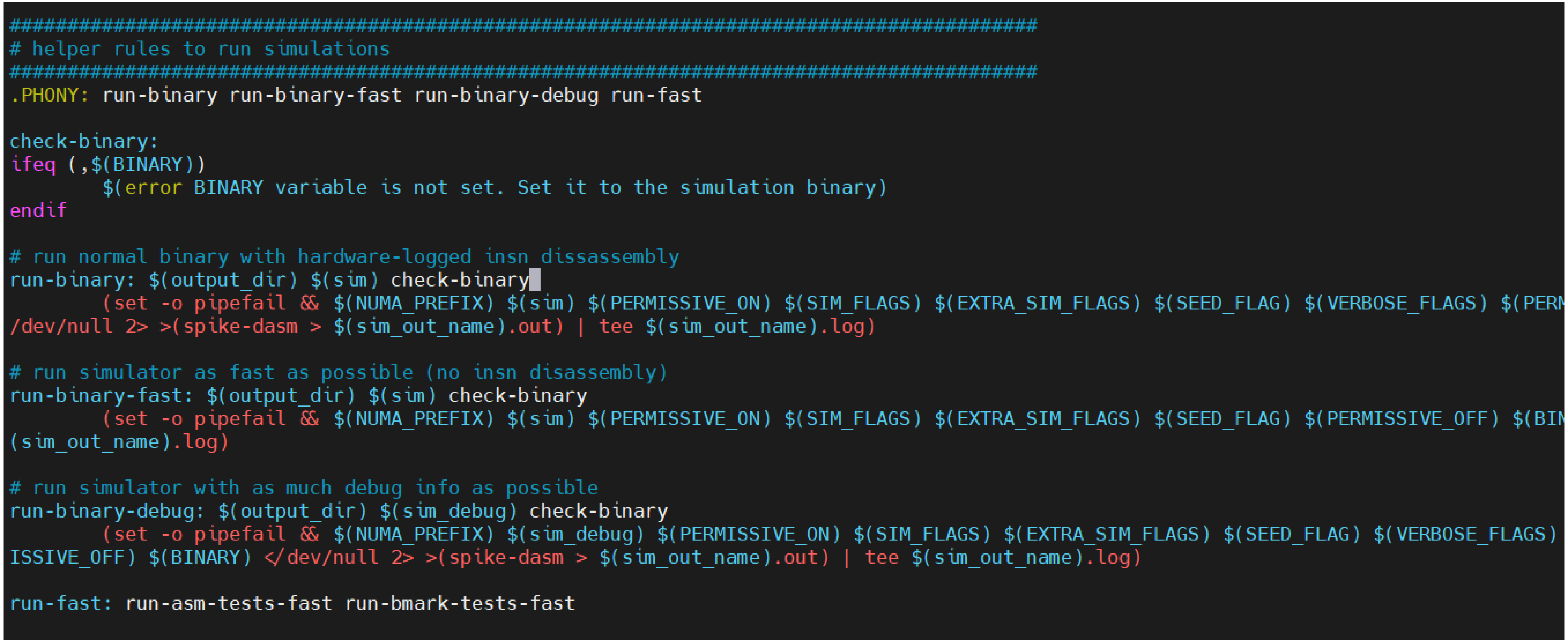
```
root@d6041fc144e9:~/chipyard# ls
CHANGELOG.md     README.md  dockerfiles         env.sh               init-submodules-no-riscv-tools.log  sims      toolchains   variables.mk  vlsi
CONTRIBUTING.md  build.sbt  docs                esp-tools-install    project                             software  tools        varialbe.mk
LICENSE          chipyard   env-esp-tools.sh    fpga                 riscv-tools-install                 target    vaiables.mk  vcs.mk
LICENSE.SiFive   common.mk  env-riscv-tools.sh  generators           scripts                             tests     variable.mk  verilator
```

# vim common.mk

그런 다음 약 174,178,182번째 줄에 있는 check-binary 지우기 (다음 장 그림 참고)

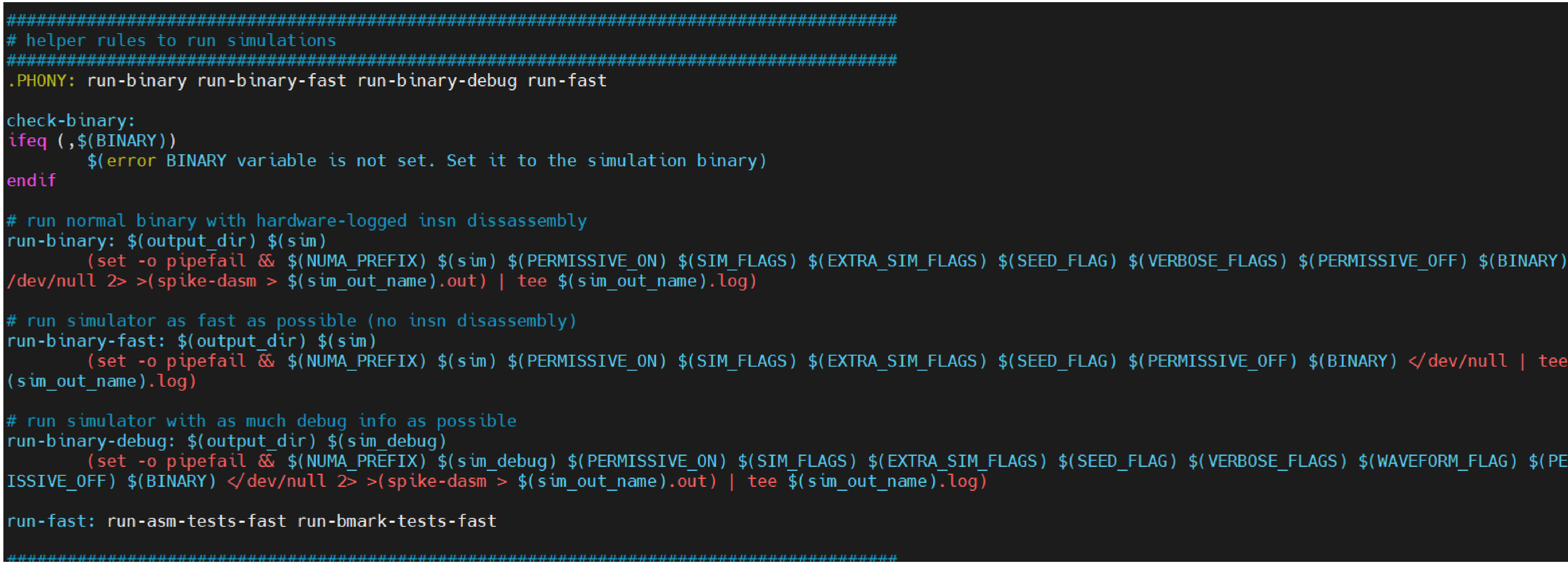
수정 하고 나서 :wq 로 저장 후 종료

BEFORE



```
#########################################################################################
# helper rules to run simulations
#########################################################################################
.PHONY: run-binary run-binary-fast run-binary-debug run-fast

check-binary:
ifeq (,$(BINARY))
        $(error BINARY variable is not set. Set it to the simulation binary)
endif

# run normal binary with hardware-logged insn dissassembly
run-binary: $(output_dir) $(sim) check-binary
        (set -o pipefail && $(NUMA_PREFIX) $(sim) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(VERBOSE_FLAGS) $(PERM
/dev/null 2> >(spike-dasm > $(sim_out_name).out) | tee $(sim_out_name).log)

# run simulator as fast as possible (no insn disassembly)
run-binary-fast: $(output_dir) $(sim) check-binary
        (set -o pipefail && $(NUMA_PREFIX) $(sim) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(PERMISSIVE_OFF) $(BIN
(sim_out_name).log)

# run simulator with as much debug info as possible
run-binary-debug: $(output_dir) $(sim_debug) check-binary
        (set -o pipefail && $(NUMA_PREFIX) $(sim_debug) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(VERBOSE_FLAGS)
ISSIVE_OFF) $(BINARY) </dev/null 2> >(spike-dasm > $(sim_out_name).out) | tee $(sim_out_name).log)

run-fast: run-asm-tests-fast run-bmark-tests-fast
```

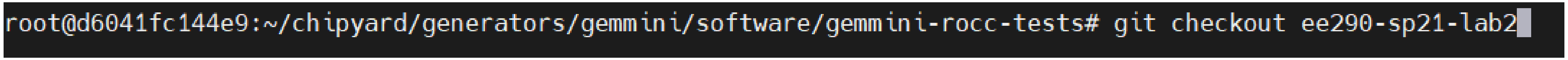
AFTER



```
#########################################################################################
# helper rules to run simulations
#########################################################################################
.PHONY: run-binary run-binary-fast run-binary-debug run-fast

check-binary:
ifeq (,$(BINARY))
	$(error BINARY variable is not set. Set it to the simulation binary)
endif

# run normal binary with hardware-logged insn dissassembly
run-binary: $(output_dir) $(sim)
	(set -o pipefail && $(NUMA_PREFIX) $(sim) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(VERBOSE_FLAGS) $(PERMISSIVE_OFF) $(BINARY)
/dev/null 2> >(spike-dasm > $(sim_out_name).out) | tee $(sim_out_name).log)

# run simulator as fast as possible (no insn disassembly)
run-binary-fast: $(output_dir) $(sim)
	(set -o pipefail && $(NUMA_PREFIX) $(sim) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(PERMISSIVE_OFF) $(BINARY) </dev/null | tee
(sim_out_name).log)

# run simulator with as much debug info as possible
run-binary-debug: $(output_dir) $(sim_debug)
	(set -o pipefail && $(NUMA_PREFIX) $(sim_debug) $(PERMISSIVE_ON) $(SIM_FLAGS) $(EXTRA_SIM_FLAGS) $(SEED_FLAG) $(VERBOSE_FLAGS) $(WAVEFORM_FLAG) $(PE
ISSIVE_OFF) $(BINARY) </dev/null 2> >(spike-dasm > $(sim_out_name).out) | tee $(sim_out_name).log)

run-fast: run-asm-tests-fast run-bmark-tests-fast

#########################################################################################
```

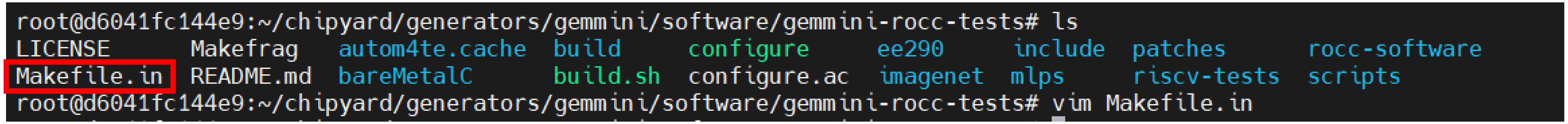
6. Git checkout

gemmini-rock-tests 디렉토리에서 git checkout 해주기



```
root@d6041fc144e9:~/chipyard/generators/gemmini/software/gemmini-rocc-tests# git checkout ee290-sp21-lab2
```

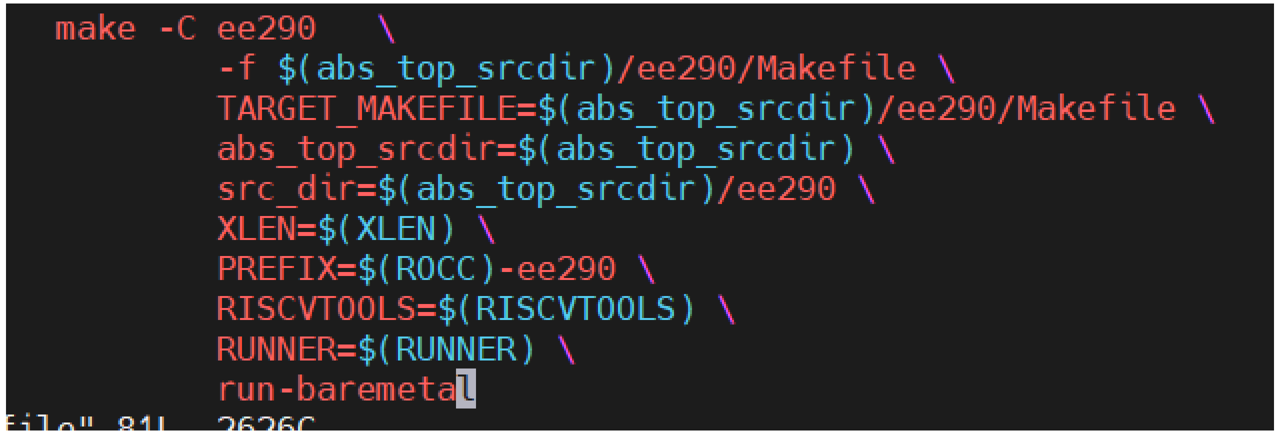
7. Makefile.in 파일 수정



```
root@d6041fc144e9:~/chipyard/generators/gemmini/software/gemmini-rocc-tests# ls
LICENSE      Makefrag   autom4te.cache  build     configure     ee290     include  patches       rocc-software
Makefile.in  README.md  bareMetalC      build.sh  configure.ac  imagenet  mlps     riscv-tests   scripts
root@d6041fc144e9:~/chipyard/generators/gemmini/software/gemmini-rocc-tests# vim Makefile.in
```

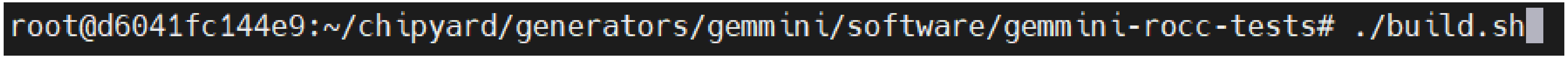
https://github.com/ucb-bar/gemmini-rocc-tests/blob/ee290-sp21-lab2/Makefile.in

여기 들어가서 맨 아래에 make –C imagenet부터 복사 한 다음에 밑에 사진 처럼 imagenet
을 전부 ee290으로 수정



```
  make -C ee290   \
          -f $(abs_top_srcdir)/ee290/Makefile \
          TARGET_MAKEFILE=$(abs_top_srcdir)/ee290/Makefile \
          abs_top_srcdir=$(abs_top_srcdir) \
          src_dir=$(abs_top_srcdir)/ee290 \
          XLEN=$(XLEN) \
          PREFIX=$(ROCC)-ee290 \
          RISCVTOOLS=$(RISCVTOOLS) \
          RUNNER=$(RUNNER) \
          run-baremetal
File" 81L  2626C
```

8. build.sh



```
root@d6041fc144e9:~/chipyard/generators/gemmini/software/gemmini-rocc-tests# ./build.sh
```

# ./build.sh

(오류가 안떠야함)

9. 테스트 돌려보기

```
cd sims/verilator
#make CONFIG=GemminiEE290Lab2RocketConfig BINARY=../../generators/gemmini/software/gemmini-rocc-tests/build/ee290/identity-baremetal run-binary
```