演習2
回路実装 & Iyokan 入門

L-III 松本直樹

Chisel の基本入門

- Chisel: 回路を記述するための Scala DSL
- 最低限の使い方のみ説明する
- 以下の資料が参考になる
- chisel3-cheatsheet
  (https://inst.eecs.berkeley.edu/~cs250/sp17/handouts/chisel-cheatsheet3.pdf)
- chisel-book
  (https://github.com/schoeberl/chisel-book)
  (https://raw.githubusercontent.com/wiki/schoeberl/chisel-book/chisel-book.pdf)
- chisel-tutorial
  (https://github.com/ucb-bar/chisel-tutorial)

実装例(カウンター回路)

- 1サイクル毎に1ずつカウントアップし7に到達すると止まる

```scala
class Counter8bit extends Module {
    val io = IO(new Bundle{
        val out     = Output(UInt(8.W))
    })

    val cnt = RegInit(0.U(8.W))

    io.out := cnt

    when(cnt === 7.U){
        cnt := cnt
    }.otherwise{
        cnt := cnt + 1.U
    }
}
```

入出力の設定
Input と Output を指定する
UInt(x.W) はx bit幅のUInt型整数であることを示す
Boolや SInt も指定できる

値を保存するためのレジスタ(D-FF)を宣言する
RegInit(x) x の内容に初期化される
0.U(8.W) は 8bit 幅のUInt であることを表す

io(入出力) の out ポートにレジスタを接続する
→ レジスタの値が out ポートに出力される

実装例(カウンター回路)

- 1サイクル毎に1ずつカウントアップし7に到達すると止まる

```
class Counter8bit extends Module {
    val io = IO(new Bundle{
        val out     = Output(UInt(8.W))
    })

    val cnt = RegInit(0.U(8.W))

    io.out := cnt

    when(cnt === 7.U){
        cnt := cnt
    }.otherwise{
        cnt := cnt + 1.U
    }
}
```

cnt レジスタの値によって処理を分岐

cnt レジスタの値が 7 である場合、
今現在の値を cnt レジスタに書き込む

それ以外の場合、
今現在の値+1 を cnt レジスタに書き込む

回路実装演習

- 以下の回路を実装
  - 算術演算回路
  - 総和計算回路

算術演算回路(ALU8bit.scala)

- 入力される符号なし8bit整数について以下の操作を行う
  - 和(ADD)
  - 差(SUB)
  - ビットAND
  - ビットXOR
  - ビットOR
- 操作は opcode として指定される

算術演算回路(ALU8bit.scala)

- 擬似コード

```
#include <stdint.h>
#define ALUOPCODE_ADD 0
#define ALUOPCODE_SUB 1
#define ALUOPCODE_AND 2
#define ALUOPCODE_XOR 3
#define ALUOPCODE_OR  4

uint8_t ALU8bit(uint8_t a, uint8_t b, uint8_t opcode)
{
  uint8_t res;
  if (opcode == ALUOPCODE_ADD) {
    res = a + b;
  } else if (opcode == ALUOPCODE_SUB) {
    res = a - b;
  } else if (opcode == ALUOPCODE_AND) {
    res = a & b;
  } else if (opcode == ALUOPCODE_XOR) {
    res = a ^ b;
  } else if (opcode == ALUOPCODE_OR) {
    res = a | b;
  }

  return res;
}
```

算術演算回路(ALU8bit.scala)

- **演習内容**
  - 前述の要件を満たす回路を ALU8bit.scala に追記してください
  - 実装後、回路の生成とテストをパスすることを確認してください

```scala
import chisel3._
import chisel3.util.BitPat

object ALUOpcode {
  def ADD = BitPat("b000")
  def SUB = BitPat("b001")
  def AND = BitPat("b010")
  def XOR = BitPat("b011")
  def OR  = BitPat("b100")
}

class ALU8bit extends Module {
    val io = IO(new Bundle{
        val inA    = Input(UInt(8.W))
        val inB    = Input(UInt(8.W))
        val opcode = Input(UInt(3.W))
        val out    = Output(UInt(8.W))
    })

    io.out := DontCare

    when(io.opcode === ALUOpcode.ADD){
        io.out := io.inA + io.inB
    }
    // ここから続きを記述してください

}
```

回路の生成確認

- sbt run で生成できる

```
naoki@LAPTOP-SMJ75B8R:~/seccamp2021-ex/ex2$ sbt run
[info] welcome to sbt 1.4.9 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/naoki/seccamp2021-ex/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/naoki/seccamp2021-ex/ex2/)
[info] compiling 5 Scala sources to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/classes ...
[info] running Main
[info] [0.011] Elaborating design...
[info] [0.586] Done elaborating.
Total FIRRTL Compile Time: 2746.1 ms
[info] [0.000] Elaborating design...
[info] [0.026] Done elaborating.
Total FIRRTL Compile Time: 183.7 ms
[success] Total time: 20 s, completed Aug 9, 2021, 12:08:11 PM
```

回路のテスト

- sbt “testOnly ALU8bitSpec” でテストできる

```
naoki@LAPTOP-SMJ75B8R:~/seccamp2021-ex/ex2$ sbt "testOnly ALU8bitSpec"
[info] welcome to sbt 1.4.9 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/naoki/seccamp2021-ex/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/naoki/seccamp2021-ex/ex2/)
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/test-classes ...
[info] [0.002] Elaborating design...
[info] [0.130] Done elaborating.
Total FIRRTL Compile Time: 1448.6 ms
Total FIRRTL Compile Time: 31.2 ms
End of dependency graph
Circuit state created
[info] [0.002] SEED 1628479507776
[info] [0.007] EXPECT AT 0   io_out got 204 expected 0 FAIL
test ALU8bit Success: 1 tests passed in 5 cycles taking 0.026984 seconds
[info] [0.009] RAN 0 CYCLES FAILED FIRST AT CYCLE 0
[info] ALU8bitSpec:
[info] - tester should returned values with interpreter *** FAILED ***
[info]   false was not true (ALU8bit.scala:52)
[info] ScalaTest
[info] Run completed in 2 seconds, 680 milliseconds.
[info] Total number of tests run: 1
[info] Suites: completed 1, aborted 0
[info] Tests: succeeded 0, failed 1, canceled 0, ignored 0, pending 0
[info] *** 1 TEST FAILED ***
[error] Failed: Total 1, Failed 1, Errors 0, Passed 0
[error] Failed tests:
[error]         ALU8bitSpec
[error] (Test / testOnly) sbt.TestsFailedException: Tests unsuccessful
[error] Total time: 9 s, completed Aug 9, 2021, 12:25:10 PM
```

回路のテスト

- 正しく実装すれば成功する

```
naoki@LAPTOP-SMJ75B8R:~/seccamp2021-ex/ex2$ sbt "testOnly ALU8bitSpec"
[info] welcome to sbt 1.4.9 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/naoki/seccamp2021-ex/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/naoki/seccamp2021-ex/ex2/)
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/classes ...
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/test-classes ...
[info] [0.002] Elaborating design...
[info] [0.141] Done elaborating.
Total FIRRTL Compile Time: 1540.7 ms
Total FIRRTL Compile Time: 60.9 ms
End of dependency graph
Circuit state created
[info] [0.002] SEED 1628479613164
test ALU8bit Success: 325125 tests passed in 5 cycles taking 6.008939 seconds
[info] [5.992] RAN 0 CYCLES PASSED
[info] ALU8bitSpec:
[info] - tester should returned values with interpreter
[info] ScalaTest
[info] Run completed in 8 seconds, 912 milliseconds.
[info] Total number of tests run: 1
[info] Suites: completed 1, aborted 0
[info] Tests: succeeded 1, failed 0, canceled 0, ignored 0, pending 0
[info] All tests passed.
[info] Passed: Total 1, Failed 0, Errors 0, Passed 1
[success] Total time: 18 s, completed Aug 9, 2021, 12:27:01 PM
```

総和計算回路(Sum8bit.scala)

- **1サイクル毎**にROM(Read Only Memory) から値を読み出し、総和を計算する
- ROM の値を x_iとして $\sum_{i=0}^{N} x_i$を計算する
- 擬似コード

```
#include <stdint.h>
#define N 5

const static uint8_t ROM[N] = {34, 5, 63, 12, 7};

uint8_t sum(){
  uint8_t res = 0;
  for (int i = 0; i < N; i++)
  {
    res += ROM[i];
  }

  return res;
}
```

総和計算回路(Sum8bit.scala)

- **演習内容**
  - カウンター回路を参考に、前述の要件を満たす回路を Sum8bit.scala に追記してください
  - 実装後、回路の生成とテストをパスすることを確認してください
- 回路について
  - sum が総和を保存するレジスタ
  - cnt が ROM のアドレス (= 配列の添字) を保存するレジスタ
  - ROM から読みだした値は io.romData から入力される

```scala
import chisel3._

class Sum8bit(val num:Int, val romWidth:Int) extends Module {
    val io = IO(new Bundle{
        val romAddr = Output(UInt(romWidth.W))
        val romData = Input(UInt(8.W))

        val out     = Output(UInt(8.W))
    })

    val sum = RegInit(0.U(8.W))
    val cnt = RegInit(0.U(romWidth.W))

    io.romAddr := cnt
    io.out := sum

    when(cnt === num.U){
      // ここに処理を記述
    }.otherwise{
      // ここに処理を記述
    }
}
```

回路のテスト

- sbt “testOnly Sum8bitSpec” でテストできる

```
naoki@LAPTOP-SMJ75B8R:~/seccamp2021-ex/ex2$ sbt "testOnly Sum8bitSpec"
[info] welcome to sbt 1.4.9 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/naoki/seccamp2021-ex/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/naoki/seccamp2021-ex/ex2/)
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/classes ...
[info] [0.001] Elaborating design...
[info] [0.178] Done elaborating.
Total FIRRTL Compile Time: 1851.5 ms
Total FIRRTL Compile Time: 68.9 ms
End of dependency graph
Circuit state created
[info] [0.002] SEED 1628481191870
[info] [0.016] 249
[info] [0.020] EXPECT AT 19   io_out got 0 expected 249 FAIL
test TopSum8bit Success: 0 tests passed in 24 cycles taking 0.039752 seconds
[info] [0.022] RAN 19 CYCLES FAILED FIRST AT CYCLE 19
[info] Sum8bitSpec:
[info] - tester should returned values with interpreter *** FAILED ***
[info]   false was not true (Sum8bit.scala:38)
[info] ScalaTest
[info] Run completed in 3 seconds, 346 milliseconds.
[info] Total number of tests run: 1
[info] Suites: completed 1, aborted 0
[info] Tests: succeeded 0, failed 1, canceled 0, ignored 0, pending 0
[info] *** 1 TEST FAILED ***
[error] Failed: Total 1, Failed 1, Errors 0, Passed 0
[error] Failed tests:
[error]         Sum8bitSpec
[error] (Test / testOnly) sbt.TestsFailedException: Tests unsuccessful
[error] Total time: 9 s, completed Aug 9, 2021, 12:53:14 PM
```

回路のテスト

- 正しく実装すれば成功する

```
naoki@LAPTOP-SMJ75B8R:~/seccamp2021-ex/ex2$ sbt "testOnly Sum8bitSpec"
[info] welcome to sbt 1.4.9 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/naoki/seccamp2021-ex/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/naoki/seccamp2021-ex/ex2/)
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/classes ...
[info] compiling 1 Scala source to /home/naoki/seccamp2021-ex/ex2/target/scala-2.12/test-classes ...
[info] [0.002] Elaborating design...
[info] [0.167] Done elaborating.
Total FIRRTL Compile Time: 1627.4 ms
Total FIRRTL Compile Time: 70.8 ms
End of dependency graph
Circuit state created
[info] [0.001] SEED 1628480874052
[info] [0.036] 249
test TopSum8bit Success: 1 tests passed in 24 cycles taking 0.055351 seconds
[info] [0.039] RAN 19 CYCLES PASSED
[info] Sum8bitSpec:
[info] - tester should returned values with interpreter
[info] ScalaTest
[info] Run completed in 3 seconds, 66 milliseconds.
[info] Total number of tests run: 1
[info] Suites: completed 1, aborted 0
[info] Tests: succeeded 1, failed 0, canceled 0, ignored 0, pending 0
[info] All tests passed.
[info] Passed: Total 1, Failed 0, Errors 0, Passed 1
[success] Total time: 11 s, completed Aug 9, 2021, 12:47:56 PM
```

回路実装演習について

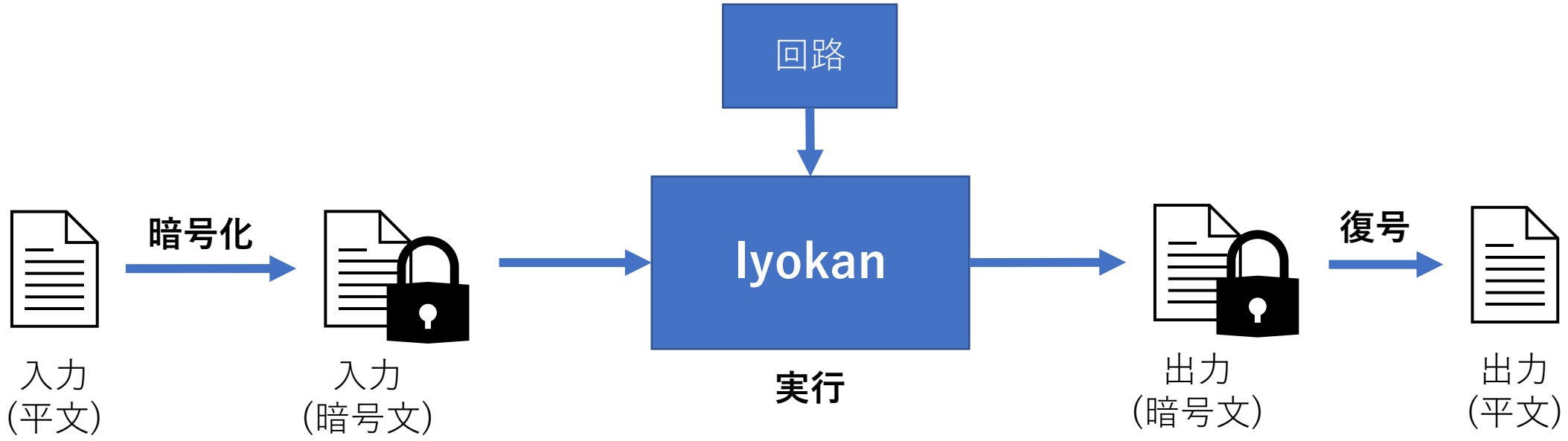
- **普通のプログラミングとは異なるパラダイム**であり、慣れるまでは大変ですが頑張ってください
- 質問やエラーの意味がわからない場合は適宜質問してください

Iyokan 入門

- 論理回路を TFHE 上で評価する実行エンジン
- CPU と GPU による評価をサポート(演習は CPU のみ利用)
- 暗号学的に最適化された ROM や RAM も提供
- 入力はモジュールのポート毎に TOML で記述し暗号化
- 出力も復号すると TOML で得られる

Iyokan 入門

- 以下のような流れで利用する



lyokan のインストール

- install_toolchain.sh を実行
- エラーが出たら教えて下さい

```
ubuntu@seccamp-2021:~/seccamp2021-l3$ ./install_toolchain.sh
(Reading database ... 81165 files and directories currently installed.)
Preparing to unpack packages-microsoft-prod.deb ...
Unpacking packages-microsoft-prod (1.0-ubuntu20.04.1) over (1.0-ubuntu20.04.1) ...
Setting up packages-microsoft-prod (1.0-ubuntu20.04.1) ...
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 https://packages.microsoft.com/ubuntu/20.04/prod focal InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1132 kB]
Hit:5 https://scala.jfrog.io/artifactory/debian all InRelease
Ign:6 https://scala.jfrog.io/artifactory/debian  InRelease
Get:9 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [844 kB]
Hit:10 https://scala.jfrog.io/artifactory/debian  Release
Fetched 2304 kB in 3s (666 kB/s)
Reading package lists... Done
```

Iyokan のインストール

```
In file included from /home/ubuntu/seccamp2021-l3/Iyokan/iyokan_tfhepp.cpp:1:
In file included from /home/ubuntu/seccamp2021-l3/Iyokan/iyokan_tfhepp.hpp:6:
/home/ubuntu/seccamp2021-l3/Iyokan/iyokan.hpp:1181:14: warning: unused variable '[item, inserted]' [-Wunused-variable]
        auto [item, inserted] =
             ^
/home/ubuntu/seccamp2021-l3/Iyokan/iyokan_tfhepp.hpp:784:13: note: in instantiation of member function 'NetworkBuilderBa
se<TFHEppWorkerInfo>::registerTask' requested here
    builder.registerTask("ram", ramPortName, indexBit, taskRAMUX);
            ^
3 warnings generated.
3 warnings generated.
[ 95%] Linking CXX executable bin/iyokan-packet
[ 95%] Built target iyokan-packet
4 warnings generated.
4 warnings generated.
7 warnings generated.
[ 97%] Linking CXX executable bin/iyokan
17 warnings generated.
[100%] Linking CXX executable bin/test0
[100%] Built target iyokan
[100%] Built target test0
ubuntu@seccamp-2021:~/seccamp2021-l3$
```

Iyokan のインストール

- バイナリが生成されていることを確認する

```
ubuntu@seccamp-2021:~/seccamp2021-l3$
ubuntu@seccamp-2021:~/seccamp2021-l3$ ls -l Iyokan/build/bin/iyokan
-rwxrwxr-x 1 ubuntu ubuntu 95003456 Aug  9 05:06 Iyokan/build/bin/iyokan
ubuntu@seccamp-2021:~/seccamp2021-l3$ ls -l Iyokan/build/bin/iyokan-packet
-rwxrwxr-x 1 ubuntu ubuntu 29281016 Aug  9 05:05 Iyokan/build/bin/iyokan-packet
```

Iyokan の使い方(回路の生成)

- sbt run で回路を生成
- ALU8bit.v (または Sum8bit.v) が生成されていることを確認

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2$ sbt run
[info] Updated file /home/ubuntu/seccamp2021-l3/ex2/project/build.properties: set sbt.version to 1.5.5
[info] welcome to sbt 1.5.5 (Ubuntu Java 11.0.11)
[info] loading project definition from /home/ubuntu/seccamp2021-l3/ex2/project
[info] loading settings for project ex2 from build.sbt ...
[info] set current project to seccamp-l3-ex2 (in build file:/home/ubuntu/seccamp2021-l3/ex2/)
[info] Updating
https://repo1.maven.org/maven2/edu/berkeley/cs/chisel-iotesters_2.12/maven-metadata.xml
  No new update since 2021-04-01 19:28:04
https://repo1.maven.org/maven2/edu/berkeley/cs/chisel3_2.12/maven-metadata.xml
  No new update since 2021-04-01 19:26:55
[info] Resolved  dependencies
[info] compiling 5 Scala sources to /home/ubuntu/seccamp2021-l3/ex2/target/scala-2.12/classes ...
[info] running Main
[info] [0.005] Elaborating design...
[info] [0.199] Done elaborating.
Total FIRRTL Compile Time: 1715.3 ms
[info] [0.000] Elaborating design...
[info] [0.011] Done elaborating.
Total FIRRTL Compile Time: 45.2 ms
[success] Total time: 13 s, completed Aug 9, 2021, 5:13:38 AM
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2$ ls -l ALU8bit.v
-rw-rw-r-- 1 ubuntu ubuntu 2403 Aug  9 05:13 ALU8bit.v
```

Iyokan の使い方(回路の生成)

- ALU8bit.v で回路の入出力が
  io_inA, io_inB, io_opcode, io_out になっていることを確認

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2$ cat ALU8bit.v
`ifdef RANDOMIZE_GARBAGE_ASSIGN
`define RANDOMIZE
`endif
`ifdef RANDOMIZE_INVALID_ASSIGN
`define RANDOMIZE
`endif
`ifdef RANDOMIZE_REG_INIT
`define RANDOMIZE
`endif
`ifdef RANDOMIZE_MEM_INIT
`define RANDOMIZE
`endif

module ALU8bit( // @[:@3.2]
  input        clock, // @[:@4.4]
  input        reset, // @[:@5.4]
  input  [7:0] io_inA, // @[:@6.4]
  input  [7:0] io_inB, // @[:@6.4]
  input  [2:0] io_opcode, // @[:@6.4]
  output [7:0] io_out // @[:@6.4]
);
  wire  _T_9; // @[ALU8bit.scala 22:20:@10.4]
  wire [8:0] _T_10; // @[ALU8bit.scala 23:26:@12.6]
  wire [7:0] _T_11; // @[ALU8bit.scala 23:26:@13.6]
```

Iyokan の使い方(回路の合成・変換)

- tests ディレクトリに移動し、 ./build.sh ../ALU8bit.v を実行
- Iyokan 専用の回路フォーマットに変換される
- ALU8bit_converted.json が生成されていることを確認

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ./build.sh ../ALU8bit.v

 /-------------------------------------------------------------------------\
 |                                                                         |
 |  yosys -- Yosys Open SYnthesis Suite                                    |
 |                                                                         |
 |  Copyright (C) 2012 - 2019  Clifford Wolf <clifford@clifford.at>        |
 |                                                                         |
 |  Permission to use, copy, modify, and/or distribute this software for any |
```

```
      $_XOR_                            6

End of script. Logfile hash: 79c8de2992
CPU: user 0.18s system 0.00s, MEM: 17.12 MB total, 11.77 MB resident
Yosys 0.9 (git sha1 1979e0b)
Time spent: 17% 9x opt_clean (0 sec), 15% 10x opt_expr (0 sec), ...
Build Done
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ls -l ALU8bit
ALU8bit.json            ALU8bit_converted.json
```

Iyokan の使い方(回路情報の定義)

- Iyokan で使う回路情報を定義する
- [[file]] で利用する回路を指定
- [connect] で回路の入出力ポートを Iyokan 側と接続
  - @A, @B, @opcode, @out は Iyokan 側で扱うポート

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ cat test-alu-8bit.toml
[[file]]
type = "iyokanl1-json"
path = "ALU8bit_converted.json"
name = "ALU"

[connect]
"ALU/io_inA[0:7]" = "@A[0:7]"
"ALU/io_inB[0:7]" = "@B[0:7]"
"ALU/io_opcode[0:2]" = "@opcode[0:2]"
"@out[0:7]" = "ALU/io_out[0:7]"
```

Iyokan の使い方(入力値の定義)

- 入力する値の定義
- Iyokan 側の入力ポート毎に値を指定できる
  - A, B, opcode について指定
  - size は bit 幅, bytes は入力するデータ

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ cat test-alu-8bit-01.in
[[bits]]
name = "A"
size = 8
bytes = [0xc]

[[bits]]
name = "B"
size = 8
bytes = [0xa]

[[bits]]
name = "opcode"
size = 3
bytes = [0x0]
```

Iyokan の使い方(鍵の生成)

- 暗号化に利用する鍵を生成
- iyokan-packet で
  秘密鍵(_test_sk) とブートストラッピング鍵(_test_bk) を生成

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet genkey --type tfhepp --out _test_sk
[2021-08-09 05:30:31.787] [iyokan-packet] [info] Starting genkey...
[2021-08-09 05:30:31.789] [iyokan-packet] [info] genkey done. (0 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet genbkey --in _test_sk --out _test_bk
[2021-08-09 05:30:52.867] [iyokan-packet] [info] Starting genbkey...
[2021-08-09 05:31:21.726] [iyokan-packet] [info] genbkey done. (28 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$
```

Iyokan の使い方(入力の暗号化)

- 入力を専用フォーマットに変換し、暗号化
- 入力を定義した TOML ファイルを専用形式に変換
- 専用形式の入力を秘密鍵を用いて評価

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet toml2packet --in test-alu-8bit-01.in --out _test_req_packet
[2021-08-09 05:40:30.964] [iyokan-packet] [info] Starting toml2packet...
[2021-08-09 05:40:30.970] [iyokan-packet] [info] toml2packet done. (0 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet enc --key _test_sk --in _test_req_packet --out _test_req_packet_enc
[2021-08-09 05:41:19.813] [iyokan-packet] [info] Starting enc...
[2021-08-09 05:41:19.815] [iyokan-packet] [info] enc done. (0 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$
```

Iyokan の使い方(実行)

- Iyokan を実行する
- 回路仕様, 入力, 出力, 実行するサイクル数を指定する
- サイクル数は組み合わせ回路の場合 1 とする

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan tfhe --blueprint test-alu-8bit.toml --bkey _test_bk -i _test_req_packet_enc
-o _test_res_packet_enc -c 1
[2021-08-09 05:44:18.647] [iyokan] [info] Options
[2021-08-09 05:44:18.647] [iyokan] [info]       Blueprint: test-alu-8bit.toml
[2021-08-09 05:44:18.647] [iyokan] [info]       # of cycles: 1
[2021-08-09 05:44:18.647] [iyokan] [info]       BKey file: _test_bk
[2021-08-09 05:44:18.647] [iyokan] [info]       Input file (request packet): _test_req_packet_enc
[2021-08-09 05:44:18.647] [iyokan] [info]       Output file (result packet): _test_res_packet_enc
[2021-08-09 05:44:18.647] [iyokan] [info]       --verbose: false
[2021-08-09 05:44:18.647] [iyokan] [info]       --quiet: false
[2021-08-09 05:44:18.658] [iyokan] [info] Run Parameters
[2021-08-09 05:44:18.658] [iyokan] [info]       Mode: TFHEpp
[2021-08-09 05:44:18.658] [iyokan] [info]       Blueprint: test-alu-8bit.toml
[2021-08-09 05:44:18.658] [iyokan] [info]       # of CPU workers: 8
[2021-08-09 05:44:18.658] [iyokan] [info]       # of cycles: 1
[2021-08-09 05:44:18.658] [iyokan] [info]       BKey file: _test_bk
[2021-08-09 05:44:18.658] [iyokan] [info]       Input file (request packet): _test_req_packet_enc
[2021-08-09 05:44:18.659] [iyokan] [info]       Output file (result packet): _test_res_packet_enc
[2021-08-09 05:44:23.992] [iyokan] [info] #1
[2021-08-09 05:44:24.613] [iyokan] [info]       done. (620925 us)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$
```

Iyokan の使い方(出力の復号)

- 結果を復号し、専用形式から TOML へ変換する
- 結果として out ポートに 22 が得られていることがわかる
- 12(0xc) + 10(0xa) = 22 であることから正しく計算出来ていることがわかる

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet dec --key _test_sk --in _test_res_packet_enc --out _test_res_packet
[2021-08-09 05:47:06.056] [iyokan-packet] [info] Starting dec...
[2021-08-09 05:47:06.058] [iyokan-packet] [info] dec done. (0 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan-packet packet2toml --in _test_res_packet
[2021-08-09 05:47:19.176] [iyokan-packet] [info] Starting packet2toml...
rom = []
ram = []
cycles = 1
bits = [
{bytes=[22],size=8,name="out"},
]
[2021-08-09 05:47:19.176] [iyokan-packet] [info] packet2toml done. (0 seconds)
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$
```

Iyokan の使い方(ROM の利用)

- Sum8bit で ROM を利用する場合は以下のように設定する

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ cat test-sum-8bit.toml
[[file]]
type = "iyokanl1-json"
path = "Sum8bit_converted.json"
name = "sum"

[[builtin]]
type = "rom"
name = "rom"
in_addr_width = 9
out_rdata_width = 8

[connect]
"rom/addr[0:8]"         = "sum/io_romAddr[0:8]"
"sum/io_romData[0:7]" = "rom/rdata[0:7]"
"sum/reset" = "@reset"
"@out[0:7]" = "sum/io_out[0:7]"
```

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ cat test-sum-8bit-01.in
[[rom]]
name = "rom"
size = 4096            # 512 * 8
bytes = [30, 75, 89, 42, 65, 16, 85, 43, 60]
```

Iyokan の使い方(ROM の利用)

- Sum8bit は**複数サイクル**で動くため、適切なサイクル数を設定し実行する

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ../../Iyokan/build/bin/iyokan tfhe --blueprint test-sum-8bit.toml --bkey _test_bk -i _test_req_packet
 -o _test_res_packet -c 15
[2021-08-09 06:02:17.476] [iyokan] [info] Options
[2021-08-09 06:02:17.476] [iyokan] [info]       Blueprint: test-sum-8bit.toml
[2021-08-09 06:02:17.476] [iyokan] [info]       # of cycles: 15
[2021-08-09 06:02:17.476] [iyokan] [info]       BKey file: _test_bk
[2021-08-09 06:02:17.476] [iyokan] [info]       Input file (request packet): _test_req_packet
[2021-08-09 06:02:17.476] [iyokan] [info]       Output file (result packet): _test_res_packet
[2021-08-09 06:02:17.476] [iyokan] [info]       --verbose: false
[2021-08-09 06:02:17.477] [iyokan] [info]       --quiet: false
[2021-08-09 06:02:17.505] [iyokan] [info] Run Parameters
[2021-08-09 06:02:17.505] [iyokan] [info]       Mode: TFHEpp
[2021-08-09 06:02:17.505] [iyokan] [info]       Blueprint: test-sum-8bit.toml
[2021-08-09 06:02:17.505] [iyokan] [info]       # of CPU workers: 8
[2021-08-09 06:02:17.505] [iyokan] [info]       # of cycles: 15
[2021-08-09 06:02:17.506] [iyokan] [info]       BKey file: _test_bk
[2021-08-09 06:02:17.506] [iyokan] [info]       Input file (request packet): _test_req_packet
[2021-08-09 06:02:17.506] [iyokan] [info]       Output file (result packet): _test_res_packet
[2021-08-09 06:02:19.458] [iyokan] [info] #1
[2021-08-09 06:02:19.838] [iyokan] [info]       done. (380057 us)
[2021-08-09 06:02:19.838] [iyokan] [info] #2
[2021-08-09 06:02:20.260] [iyokan] [info]       done. (422234 us)
[2021-08-09 06:02:20.260] [iyokan] [info] #3
[2021-08-09 06:02:20.609] [iyokan] [info]       done. (348933 us)
[2021-08-09 06:02:20.609] [iyokan] [info] #4
[2021-08-09 06:02:20.956] [iyokan] [info]       done. (347077 us)
[2021-08-09 06:02:20.956] [iyokan] [info] #5
[2021-08-09 06:02:21.330] [iyokan] [info]       done. (373418 us)
[2021-08-09 06:02:21.330] [iyokan] [info] #6
[2021-08-09 06:02:21.716] [iyokan] [info]       done. (386717 us)
[2021-08-09 06:02:21.717] [iyokan] [info] #7
```

lyokan の使い方(自動テスト)

- build.sh で ALU8bit.v と Sum8bit.v を変換し、
  test.sh で lyokan を利用したテストを行うことができる

```
ubuntu@seccamp-2021:~/seccamp2021-l3/ex2/tests$ ./test.sh
Preparing skey and bkey...
Done.
Testing ALU8bit-01...
Done.
Testing ALU8bit-02...
Done.
Testing ALU8bit-03...
Done.
Testing ALU8bit-04...
Done.
Testing ALU8bit-05...
Done.
Testing Sum8bit-01...
Done.
```

Iyokan の演習

- これまでの手順を確認して実際に動かしてみてください
- test.sh を動かしてテストをパスすることを確認してください