

Table of Contents

| | | |
|-----------|--------------------------------|---|
| 1 | Introduction..... | 1 |
| 2 | Deliveries Description..... | 1 |
| 2.1 | cocotb/drivers/amba.py..... | 1 |
| 2.2 | doc..... | 2 |
| 2.2.1 | SystemMMU_func.pdf..... | 2 |
| 2.2.2 | TestBench_Description.pdf..... | 2 |
| 2.3 | hdl..... | 2 |
| 2.3.1 | mmu.v..... | 2 |
| 2.3.1.1 | axi_data_channel.v..... | 2 |
| 2.3.1.1.1 | fifo.v..... | 2 |
| 2.3.1.2 | ram_sp_sr_sw.v..... | 2 |
| 2.3.1.3 | mmu_parameter.v..... | 2 |
| 2.3.1.4 | tscale.v..... | 2 |
| 2.4 | tests..... | 2 |
| 2.4.1 | Makefile..... | 3 |
| 2.4.2 | *.gtkw..... | 3 |
| 2.4.3 | top_mmu.v..... | 3 |
| 2.4.4 | test_mmu.py..... | 3 |
| 2.4.5 | sim.log..... | 3 |
| 2.4.6 | mmu.cov & mmu.dcov..... | 3 |
| 2.5 | yosys..... | 3 |
| 2.5.1 | Makefile..... | 3 |
| 2.5.2 | synth.js..... | 4 |
| 2.6 | Script..... | 4 |
| 2.6.1 | aescript.py..... | 4 |

1 Introduction

All deliveries can be found at http://firmware-developments.com/WEB/P6x/RTL_MMU/ and are in plain text format except 2 source files that are encrypted (AES).

- mmu.v
- test_mmu.v

A python script is provided to decrypt those 2 files and once your payment is accepted we give you the key to decipher it by email.

2 Deliveries Description

2.1 cocotb/drivers/amba.py

Updated Amba drivers. Original file coming with Cocotb has been replaced by that one.

Original location : `$CocotbHome/cocotb/drivers`

2.2 doc

Directory containing all documents.

2.2.1 [SystemMMU_func.pdf](#)

System MMU functional specification

2.2.2 [TestBench_Description.pdf](#)

Provides an overview of the testbench structure as well as its different components (drivers, generator)

2.3 hdl

Directory containing all Verilog hdl source files.

2.3.1 mmu.v

Top level MMU.

Instantiate the below modules.

2.3.1.1 *axi_data_channel.v*

Buffered AXI4 data channel.

Instantiate the below module.

2.3.1.1.1 fifo.v

Fifo description.

2.3.1.2 ram_sp_sr_sw.v

SRAM model. Should be replaced by a model of your targeted library.

2.3.1.3 mmu_parameter.v

Contains all Verilog parameters used in the MMU description.

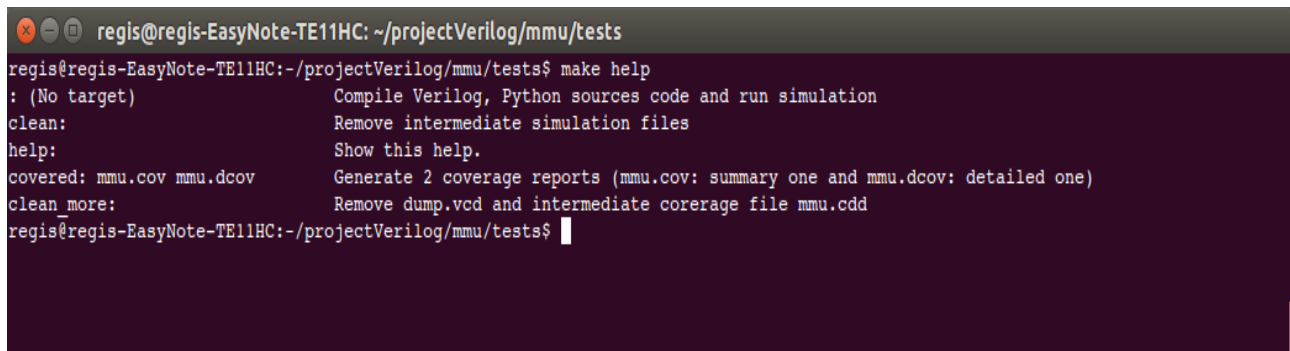
2.3.1.4 tscale.v

Include definition of timescale parameter.

2.4 tests

Contains files relative to MMU verification, testbench and test coverage.

2.4.1 Makefile



```
regis@regis-EasyNote-TE11HC: ~/projectVerilog/mmu/tests
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/tests$ make help
: (No target)          Compile Verilog, Python sources code and run simulation
clean:                Remove intermediate simulation files
help:                 Show this help.
covered: mmu.cov mmu.dcov  Generate 2 coverage reports (mmu.cov: summary one and mmu.dcov: detailed one)
clean more:           Remove dump.vcd and intermediate coverage file mmu.cdd
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/tests$
```

2.4.2 *.gtkw

GTKWAVE save files. Contains MMU signals lists to be analyzed with GTKWAVE (waveform viewer)

2.4.3 top_mmu.v

Top level mmu.

All inputs vectors from MmuTB are delayed before being provided to DUT.

All outputs vectors form DUT are delayed before being provided to MMUTB

Objective is to avoid any signals race condition.

2.4.4 test_mmu.py

Main test program developed in Python.

2.4.5 sim.log

Simulation log file (from cocotb and icarus verilog tools)

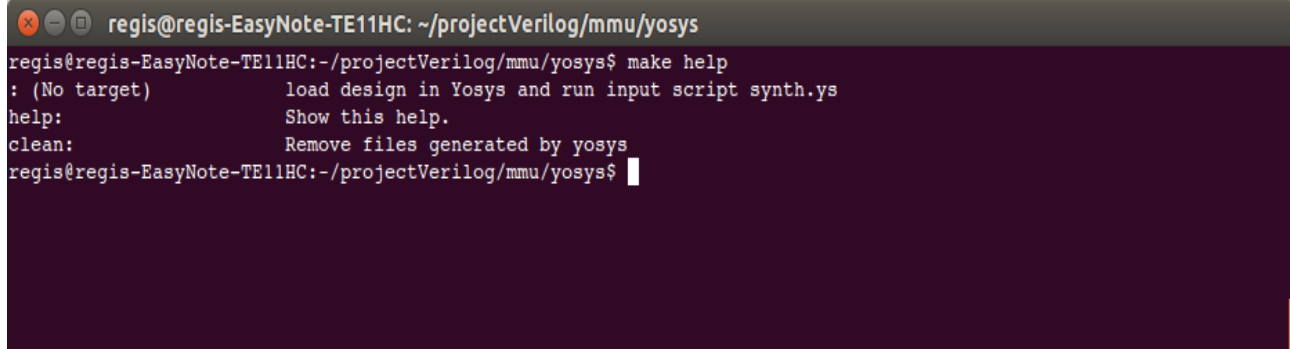
2.4.6 mmu.cov & mmu.dcov

Coverage reports (summary and detailed ones) generated from covered tool.

2.5 yosys

yosys synthesis suite is used to check that design is synthesizable.

2.5.1 Makefile



```
regis@regis-EasyNote-TE11HC: ~/projectVerilog/mmu/yosys
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/yosys$ make help
: (No target)          load design in Yosys and run input script synth.py
help:                  Show this help.
clean:                 Remove files generated by yosys
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/yosys$
```

2.5.2 synth.py

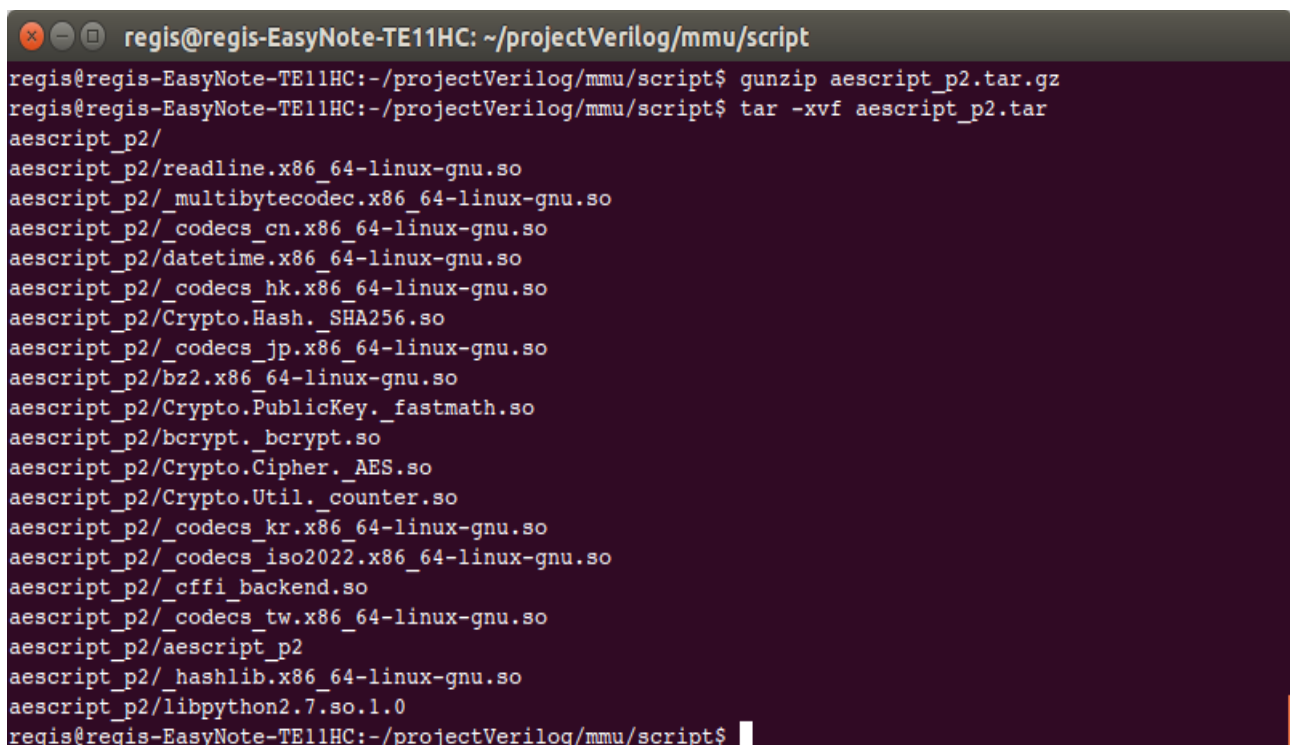
Synthesis script.

2.6 Script

An executable (Linux only) python script is provided to decrypt encrypted files.

It comes as a compress tar file.

Before being able to use it, you have to execute following commands:



```
regis@regis-EasyNote-TE11HC: ~/projectVerilog/mmu/script
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$ gunzip aescript_p2.tar.gz
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$ tar -xvf aescript_p2.tar
aescript_p2/
aescript_p2/readline.x86_64-linux-gnu.so
aescript_p2/_multibytecodec.x86_64-linux-gnu.so
aescript_p2/_codecs_cn.x86_64-linux-gnu.so
aescript_p2/datetime.x86_64-linux-gnu.so
aescript_p2/_codecs_hk.x86_64-linux-gnu.so
aescript_p2/Crypto.Hash._SHA256.so
aescript_p2/_codecs_jp.x86_64-linux-gnu.so
aescript_p2/bz2.x86_64-linux-gnu.so
aescript_p2/Crypto.PublicKey._fastmath.so
aescript_p2/bcrypt._bcrypt.so
aescript_p2/Crypto.Cipher._AES.so
aescript_p2/Crypto.Util._counter.so
aescript_p2/_codecs_kr.x86_64-linux-gnu.so
aescript_p2/_codecs_iso2022.x86_64-linux-gnu.so
aescript_p2/_cffi_backend.so
aescript_p2/_codecs_tw.x86_64-linux-gnu.so
aescript_p2/aescript_p2
aescript_p2/_hashlib.x86_64-linux-gnu.so
aescript_p2/libpython2.7.so.1.0
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$
```

2.6.1 aescript_p2

```
regis@regis-EasyNote-TE11HC: ~/projectVerilog/mmu/script
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$ aescript_p2/aescript_p2 -h
usage: aescript_p2 [-h] [-c | -d] [-k | -p] keypass infile [outfile]

AES encryption / decryption of a file

positional arguments:
  keypass      AES Key (32-Bytes expected) or password. !! Don't forget to
               enclose keypass with single quote (')
  infile       input file
  outfile      output file. If not provided output file is infile.e when
               encrypting or infile.d when decrypting

optional arguments:
  -h, --help    show this help message and exit
  -c, --crypt   File encryption
  -d, --decrypt File decryption
  -k, --key     keypass is AES key
  -p, --passwd  keypass is password to generate AES key (!! Unsupported
               option when decrypting)
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$
```

You will need to execute that script to decrypt the 2 encrypted files:

- mmu.v.e
- test_mmu.py.e

To decrypt test_mmu.py.e, use the below command (replace 'xxxx' with the provided key).

```
regis@regis-EasyNote-TE11HC: ~/projectVerilog/mmu/script
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$ \
> aescript_p2/aescript_p2 -d -k '$2b$12$17aS8jhBG8g8YDENhC1xxuwms' ../tests/test_mmu.py.e
using AES key $2b$12$17aS8jhBG8g8YDENhC1xxuwms
regis@regis-EasyNote-TE11HC:~/projectVerilog/mmu/script$
```