AES Encryption Databus Unit with USB Protocol

Verification Plan

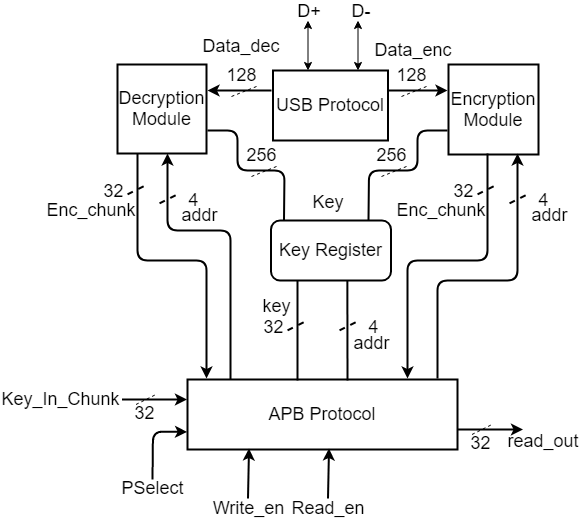
Team Members: Da Cheng, Benjamin Parker, Joe Trichak, Zihan Wang

Submission Date: 4/8/18

Thursday 11:30 Lab (Lab Section 4)

Teaching Assistant: Mochen Linghu

**Architecture**



**AES Encryption:** The top level block that handles the encryption of the 128 bit data packets received through the USB Receiver Protocol and stores the resulting encrypted data.

**AES Decryption:** The top level block that handles the decryption of the 128 bit data packets received through the USB Receiver Protocol and stores the resulting decrypted data.

**USB Receiver Protocol:** The top level block that directs the flow of the 128 bit data packets being received from the user to the encryption or decryption blocks.

**APB Protocol:** The top level block that controls the reading out of the encrypted or decrypted data as well as the writing in of the 256 bit key to the Key Register block.

**Key Register:** This is a 256 bit serial to parallel shift register that is interfaced with the APB protocol to receive an input 256 bit key. The Key Register block is being tested through the APB protocol.

**Fixed Criteria**

1. (2 points) Test benches for each top level block in the design and the top level design. These test benches exhibit full functionality and can be used to demonstrate the functional requirements given in the design specific success criteria.
2. (4 points) Entire design synthesizes completely, without any inferred latches, timing arcs, and, sensitivity list warnings.
3. (2 points) The source and mapped versions of each block and the top level design behave the same for all test cases. There are only timing errors in the mapped version at t = 0.
4. (2 points) A complete IC layout is produced that passes all geometry and connectivity checks.
5. (2 points) The entire design complies with targets for area, pin count, throughput, and clock rate. Partial credit in this criteria will be given if the targets are not met but are close enough to the target goals.
   1. Size: Framed with IO Pads 27 mm^2 / Unmapped 18 mm^2
   2. Pinout: 73 pins (power, ground, clock, active low reset, 2 USB signals, 67 APB signals)
   3. Clock Period: 96 MHz

**Design Specific Criteria**

1. (2 point) Demonstrate by simulation of a Verilog test bench and compare output to KAT (Known Answer Test) Vectors to verify the AES encryption algorithm.
2. (1 point) Demonstrate by simulation of a Verilog test bench and compare output to KAT Vectors to verify the AES decryption algorithm
3. (2 points) Demonstrate by simulation of a Verilog test bench that the USB receiver unit meets the 12 Mb/s throughput requirement (full-speed transmission), uses the CRC5 and CRC16 protocols correctly, and accounts for bit-stuffing to verify successful design.
4. (1 point) Demonstrate by simulation of a Verilog test bench that the CRC5 and CRC16 can check for errors successfully using polynomial division.
5. (1 point) Demonstrate by simulation of a Verilog test bench that the APB protocol is able to receive key data and transmit output from buffers successfully using the cyclic address - active read/write out procedure.
6. (1 point) Demonstrate by simulation of a Verilog test bench that the top level design meets all of the requirements and works together while meeting the above requirements for each protocol in unison.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| What to Verify | Design Modules Involved | Procedure Summary | DSSC Proved | Use in Final Demo | Comments |
| Successful Encryption | Top Level | Compare encryption results with known KAT vectors | 1 & 6 | Yes | Send and retrieve data from calculation block properly |
| Successful Decryption | Top Level | Compare decryption results with known KAT vectors | 2 & 6 | Yes | Send and retrieve data from calculation block properly |
| USB Receiver Protocol Interfacing | Top Level | Use test bench to provide samples of possible bus transactions | 3, 4, &6 | Yes | Check packet validity, order of transmission, CRC checking, and proper control of data flow |
| APB Protocol Interfacing | Top Level | Use test bench to provide samples possible of bus transactions | 5 & 6 | Yes | Verify with registers to control address indexing, read out and write in of storage |
| Successful Encryption | Encryption Block | Compare encryption results with known KAT vectors | 1 | Only if fails in top level | Feed in key and input, output holds constant |
| Successful Decryption | Decryption Block | Compare decryption results with known KAT vectors | 2 | Only if fails in top level | Feed in key and input, output holds constant |
| USB Receiver Protocol Interfacing | USB Receiver Protocol Block | Use test bench to provide samples of possible bus transactions | 3 & 4 | Only if fails in top level | Check packet validity, order of transmission, CRC checking, and proper control of data flow |
| APB Protocol Interfacing | APB Protocol Block | Use test bench to provide samples possible of bus transactions | 5 | Only if fails in top level | Verify with registers to control address indexing, read out and write in of storage |
| Verify CRC5 and CRC16 Correctly Validate Packets | USB Receiver Protocol - CRC5/16 Block | Send data through each verification individually with correct/incorrect CRC packets and check if the results match what is expected | 4 | Only if fails in the top level  (USB) | Transmit both correct and incorrect CRC expansions to verify error detection |

**Detailed Verification Test Breakouts**

**Demo Tests**

**Successful Top Level Encryption**

* Shown in Demo: Yes
* DSSC(s) Proved: 1 & 6
* Highest level of design involved
  + Total Design
* Test Bench Expectations/Requirements
  + Use the same input data as the individual test bench for this data
* Cross reference with <http://rijndael.online-domain-tools.com/> for comparison
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Check read out results from overall top level test bench and cross reference with the same calculation to verify the algorithm works properly

**Successful Top Level Decryption**

* Shown in Demo: Yes
* DSSC(s) Proved: 2 & 6
* Highest level of design involved
  + Total Design
* Test Bench Expectations/Requirements
  + Use the same input data as the individual test bench for this data
* Cross reference with <http://rijndael.online-domain-tools.com/> for comparison
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Check read out results from overall top level test bench and cross reference with the same calculation to verify the algorithm works properly

**Top Level USB Receiver Protocol Interfacing**

* Shown in Demo: Yes
* DSSC(s) Proved: 3, 4 & 6
* Highest level of design involved
  + Total Design
* Test Bench Expectations/Requirements
  + Use some of the sample interactions from the individual module test bench for this protocol
  + Have samples that trigger error transmissions as well, including triggering both CRC errors, wrong packet orders, etc.
* Only packet references are needed for USB
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Simulate the transaction samples used from the individual test bench
  + 2) Check for proper response against the samples used in the test bench
  + 3) Repeat 1-2 for all possible types of interactions with the USB protocol

**Top Level APB Protocol Interfacing**

* Shown in Demo: Yes
* DSSC(s) Proved: 5 & 6
* Highest level of design involved
  + Total Design
* Test Bench Expectations/Requirements
  + Use some of the sample interactions from the individual module test bench for this protocol
  + Use both write-in and read-out transactions to verify correct interaction with registers
* No external references are needed
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Test an interaction for the bidirectional APB protocol using the same stimulus in the individual module test bench
  + 2) Verify correct response from the APB read out signals
  + 3) Repeat 1-2 for all possible interactions

**Successful Module Level Encryption**

* Shown in Demo: Only if can’t show using top level
* DSSC(s) Proved: 1
* Highest level of design involved
  + Encryption Block
* Test Bench Expectations/Requirements
  + Randomly generated input keys and data
* Cross reference with <http://rijndael.online-domain-tools.com/> for comparison
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Randomly generate raw data for encryption
  + 2) Feed data as if it came from the USB protocol into the proper input register
  + 3) Check the encrypted data with the above reference model
  + 4) Repeat steps 1-3 for multiple data values

**Successful Module Level Decryption**

* Shown in Demo: Only if can’t show using top level
* DSSC(s) Proved: 2
* Highest level of design involved
  + Decryption Block
* Test Bench Expectations/Requirements
  + Randomly generate inputs for raw data
* Cross reference with <http://rijndael.online-domain-tools.com/> for comparison
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Randomly generate raw data for decryption
  + 2) Feed data as if it came from the USB protocol into the proper input register
  + 3) Check the decrypted data with the above reference model
  + 4) Repeat steps 1-3 for multiple data values

**Successful Module Level USB Protocol Interfacing**

* Shown in Demo: Only if can’t show using top level
* DSSC(s) Proved: 3 & 4
* Highest level of design involved
  + USB Receiver Module
* Test Bench Expectations/Requirements
  + Have packets pre-processed and ready for all types of bus interactions
  + Have packets that cause errors as well, including CRC errors to verify the CRC module works correctly
  + Have registers that load in data from the protocol to verify correct reception from the input lines
* Only packet references are needed for the USB protocol
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Simulate bus transactions from list of samples
  + 2) Check the receiver’s response against the correct behavior as defined by the protocol
  + 4) Repeat steps 1-2 for all possible types of interactions

**Successful Module Level APB Protocol Interfacing**

* Shown in Demo: Only if can’t show using top level
* DSSC(s) Proved: 5
* Highest level of design involved
  + APB Module
* Test Bench Expectations/Requirements
  + Have all possible interactions with this protocol defined
  + Have registers on both sides of the protocol to verify proper interactions with data
* No external references are needed
* No pre/post processing is needed
* Main Verification Test Steps:
  + 1) Simulate bus transactions
  + 2) Check the receiver’s response against the correct behavior as defined by the protocol
  + 4) Repeat steps 1-2 for all possible types of interactions

**Successful Module Level CRC Error checking**

Shown in Demo: Only if can’t show using top level or USB protocol

* DSSC(s) Proved: 4
* Highest level of design involved
  + CRC Module
* Test Bench Expectations/Requirements
  + Have streams of data for both CRC5 and CRC16 modules to feed in serially to the Galos shift register
* No external references are needed
* CRC5 and CRC16 module expansions need to be processed beforehand
* Main Verification Test Steps:
  + 1) Stream in data followed by CRC expansion
  + 2) Check the modules response to the stimulus
  + 4) Repeat steps 1-2 for various data input streams