AES Encryption on USB Platform

Charles Hansen, Dan Suciu, Mitch Bouma, Alex Dunker

Project Overview

USB 1.1

- The chip will accept USB data which includes a packet designated to be the key for encryption, encrypt the data following the AES encryption protocol standards, and send the data using the same USB 1.1 standards.
- The chip can be used to encrypt not only portable flash-drives and hard-drives, but any storage device with a USB interface
- ASIC Design allows for Improved Performance and smaller package

AES - 128 bit encryption

- Symmetric-key encryption adopted by the government in 2001
- Physical encryption is necessary in an age of exposed and sensitive information
- Chip encryption make the process much faster, and the it can be small enough to fit inside the USB device itself
- A hard-wired chip means that altering the algorithm to fit an attacker’s agenda is impossible

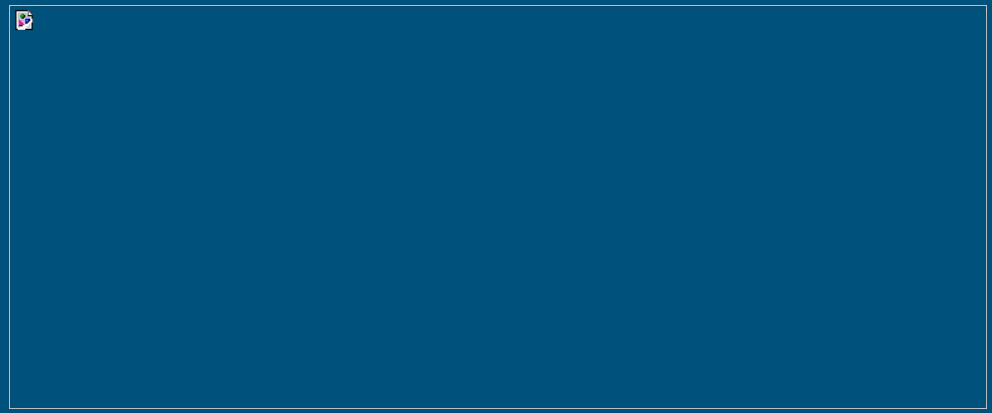
System Design

AES Encryptor Chip Top Level Diagram

System Design Continued...

USB Interface

- As packets come in, the receiver determines their types. Any packets other than data are stored in their buffers to be used in the transmit stage.
- Data packets will be put in a fifo that holds 128 bits which is sent for encryption when full.
- A 0 bit is represented by the output level staying constant whereas a 1 bit is represented by the output level changing
- Incoming data is sent most significant bit first and starts with a SYNC byte (01111111).

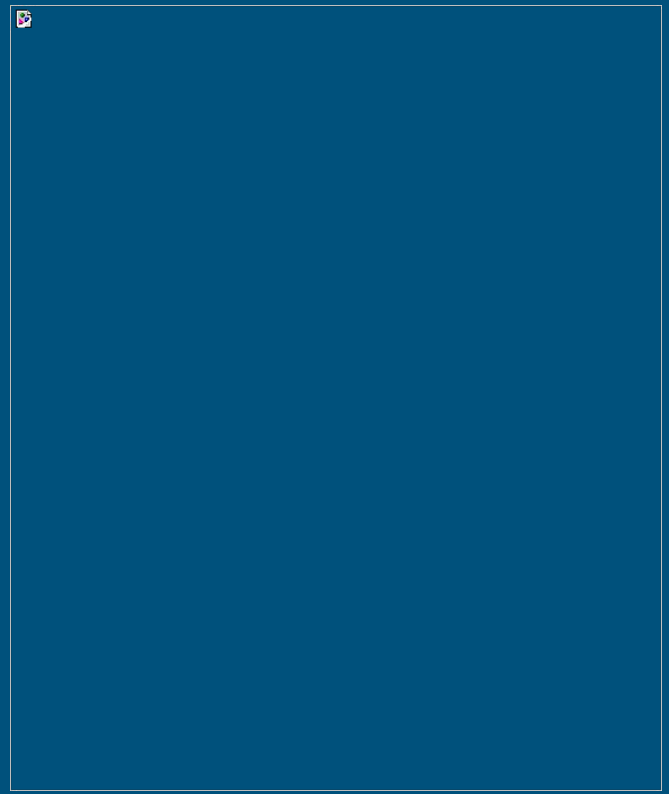


USB Receiving Block Diagram

System Design Continued...

AES Encryption

- Before first round, key is taken from first 128 bits of data and is expanded into 10 separate round keys
- 4 steps to encryption, each runs 10 times
- Sub Bytes - each byte replaced with lookup table
- Shift Rows - rows of 4 x 4 bytes are circularly shifted
- Mix Columns - each byte is a function of the others in its “column” and uses GF(2^8)
- Add Round Key - XOR with corresponding round key



AES Encryption Flowchart

System Design Continued...

USB Transmission

- Transmission follows the same guidelines as receiving USB data
- Uses PID fifo to decide what kind of information must be transmitted.
- When encrypted data fifo is full, a PID is read from the PID fifo and depending on the value, a certain number of bytes is read from each of the other fifo's (data, crc16, and non-data)

USB Transmission Block Diagram

Results

Success Criteria

1. Test benches for all top level components and the entire design
   -Shown in demonstration, **Achieved**
2. Design synthesizes completely without latches, timing arcs, and sensitivity list warnings
   -Shown in demonstration, **Achieved**
3. Sourced and mapped versions of the complete design behave the same for all test cases except for mapped version operation at time zero
   -Correct output for key expansion/parts of encryption, but not the final mapped product, **Partially Achieved**
4. Complete IC layout is produced that passes all geometry and connectivity checks
   -Show screenshots, **Achieved**
5. Entire design compiles with targets for area, pin count, throughput, and clock rate listed in requirements
   Pin count - **10** | Total cell area ~ **6 x 6 mm** | Clock rate - **24 Mhz**, **Achieved**

Results Continued...

Design Specific Success Criteria

1. Successfully deal with incorrect key size
   -Shown in demonstration, **Achieved**
2. Successfully encrypt files not multiples of 128 bit sizes
   -Shown in demonstration, **Achieved**
3. Successfully encrypt files up to 1MB
   -Shown in demonstration, **Achieved**
4. Successfully encrypt files of different extensions (png, pdf, .docx)
   -Shown in demonstration, **Achieved**

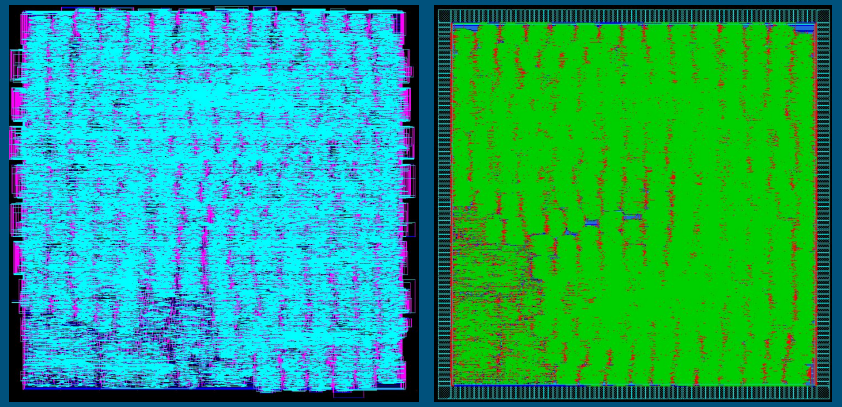
Results - IC Layout

Size (from Cadence Encounter Reports):

- # IO Pins: 10
- # Std Cells: 450110
- Total area of chip: 74,047,975 um^2 @ 60% density
- Total area from mapped report: 35,963,532 um^2
- Design budget estimated 37,120,485 um^2

Timing (from Cadence Encounter Reports):

- Critical path: Input USB -> Output USB
- Time: 99.687 ns
- Design Budget estimated 111.984 ns

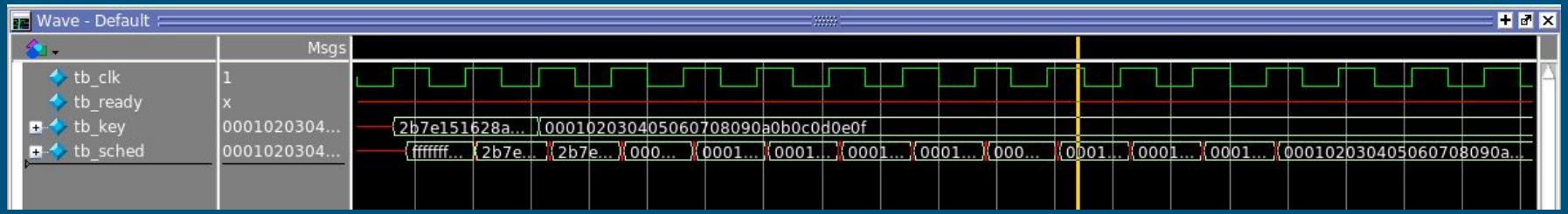


Virtuoso & Encounter IC Layout

```
############################################################
#  Generated by:      Cadence Encounter 09.11-s084_1
#  OS:                Linux x86_64(Host ID ee215lnx01.ecn.purdue.edu)
#  Generated on:      Tue May  3 13:16:10 2016
#  Command:           verifyConnectivity -type all -noWeakConnect -connLoop ...
############################################################
Verify Connectivity Report created on Tue May  3 13:16:10 2016



Begin Summary
  Found no problems or warnings.
End Summary

############################################################
#  Generated by:      Cadence Encounter 09.11-s084_1
#  OS:                Linux x86_64(Host ID ee215lnx01.ecn.purdue.edu)
#  Generated on:      Tue May  3 13:16:27 2016
#  Command:           verifyConnectivity -type all -geomLoop -error 1000 -wa...
############################################################
Verify Connectivity Report created on Tue May  3 13:17:27 2016



Begin Summary
  Found no problems or warnings.
End Summary
```

Geometry & Connectivity

Results - Test Waveforms

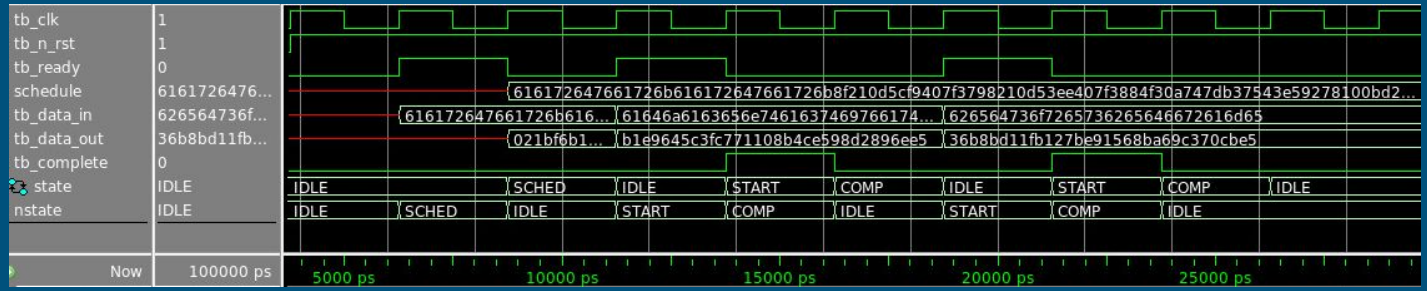


Timing of the mapped Key Schedule

Key schedule verified to Advanced Encryption Standard (AES) (FIPS PUB 197)
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Results - Test Waveforms

Timing of AES Encryption - Multiple 128 bit Sets - Verified by Python Script



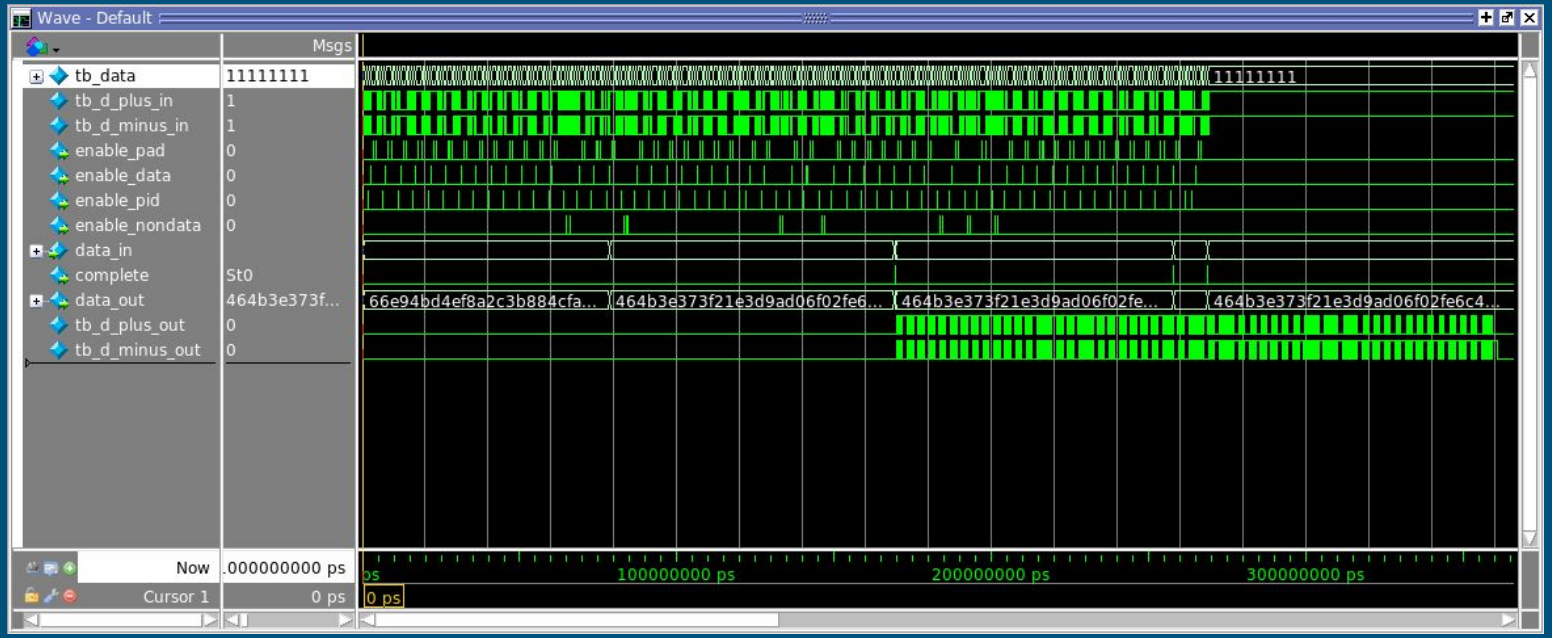
Verification of AES from NIST FIPS document 197

```
Input =       32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
```

output

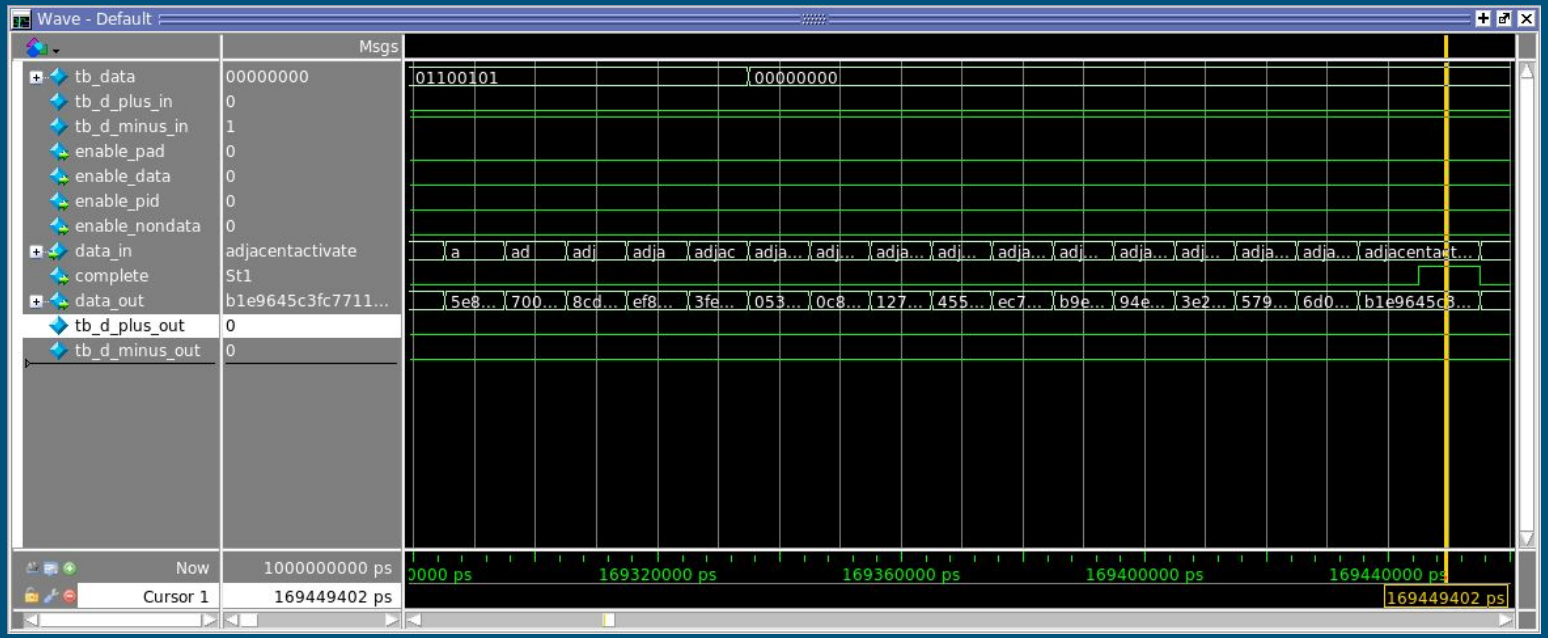
39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

Results - Test Waveforms



Encrypting Full File

Results - Test Waveforms



Encrypting a Section of 128 Bits

Conclusions

Biggest Challenges To Our Design?

- Internal timing
- Test bench simulation for USB input
- GF(2^8) multiplication

Improvements We Would Make?

- FPGA implementation to use true USB
- USB 2.0/3.0 Protocol

How Would We Do Things Differently?

- Keep everyone in the loop throughout the project
- Better naming conventions

Q&A

Thank you for your time and attention!