AES Encryption over PCI Express

Project Proposal

Zach Curosh
Matt Swanson
Jevin Sweval

ECE 337
Tuesday, 11:30AM Lab

2009-03-13

GTA: Paul Griffin

Signed:
Zach Curosh
Matt Swanson
Jevin Sweval

Executive Summary

AES encryption has become the new preferred cryptographic cipher of the United States government after a detailed standardization process. This vetting has caused it to be used throughout industry wherever data privacy is a concern. Our project goal is to design and implement AES encryption for a standard cell ASIC process. PCI Express will interface the AES core with the rest of the system for its performance and widespread use. Implementing AES encryption inside hardware will give users high-throughput and reduced power consumption versus a software implementation. This hardware acceleration will offload the encryption from the processor, allowing it to perform other tasks simultaneously. An ASIC implementation will offer large cost benefits over an FPGA from reduced per unit cost that mass-production provides. Additionally, an ASIC implementation will be much faster than a microcontroller implementation due to multi-byte data paths. This proposal will describe in detail the design specifications (including diagrams), project timeline, and success criteria for our project.

Design Specifications

Operational Characteristics

Signal Descriptions

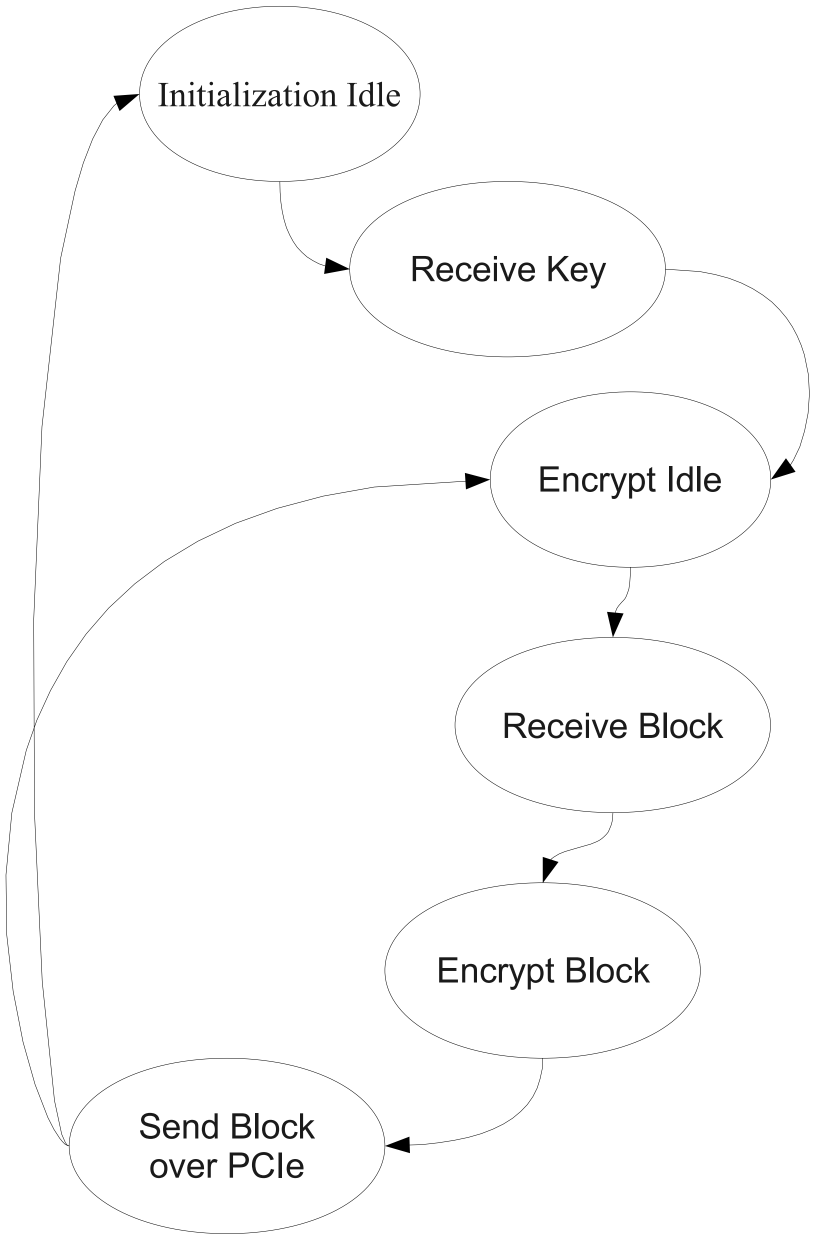
Signal Name	Type	Number of Pins	Description
NRST	IN	1	Asynchronous active-low Reset signal for all signals in the PCIe and AES blocks.
PCLK	CLK	1	The system clock (250 MHz) that clocks the AES blocks. Generated by the external PCIe PHY IC.
TXDATA[7:0]	OUT	8	8-bit transmit data input from the MAC to the PHY .
TXDATAK	OUT	1	selection input for the symbols of transmit data; LOW = data byte; HIGH = control byte
RXDATA[7:0]	IN	8	8-bit receive data output from the PHY to the MAC .
RXDATAK	IN	1	selection output for the symbols of receive data; LOW = data byte; HIGH = control byte
COMMAND[6:0]	OUT	7	Controls the operation and modes of the PCIe PHY IC.
STATUS[5:0]	IN	6	PCIe PHY IC status and fault signals.
VCC	PWR	1	Bus to provide power to the chip.
GND	PWR	1	Bus to provide ground to the chip.

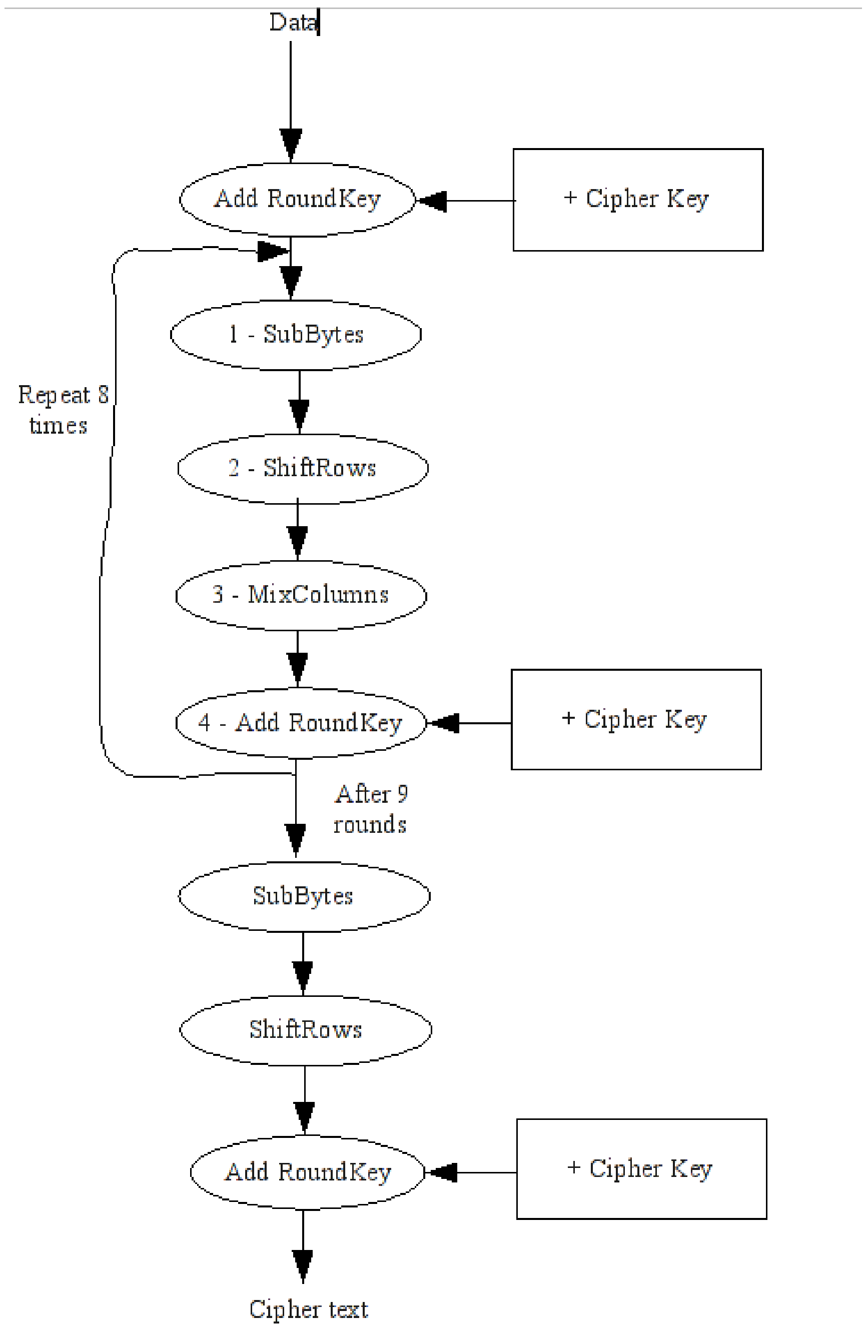
External Hardware and Protocols

The ASIC will interface with the host device using the industry standard PCI Express 1.1 interface and protocol. This allows for high throughput and compatibility with modern systems. To handle the high-speed 2.5 Gbps PCIe bitstream, the ASIC will interface with a NXP PX1011B PCIe PHY IC. This IC handles all of the physical layer for PCI Express, including the 8b/10b encoding/decoding, clock recovery, and data/control differentiation. The IC uses differential signals to communicate with the PCIe MAC device (our ASIC) but our ASIC standard cell library only contains CMOS drivers. It is assumed that there will be some transceiver between the ASIC and PCIe PHY IC.

Overall Operation

The above state diagram displays the overall operation of our design. Once the key has been received, we may begin reading in data blocks to encrypted. The data is then passed to the AES core where it is encrypted. Once the encryption is finished, the ASIC will then either signal to the host that it is complete or write out the encrypted block to memory. At this point, the ASIC will then wait for more data or change the key, depending on the host's commands.





The above flowchart describes the AES process, step-by-step. This cipher is specified as a certain number of repetitions of certain steps (transformations) that converts the input data into encrypted cipher text. For the full details and mathematical formulas, please refer to the AES standard.

The first step is the Add RoundKey step where the subkey is derived from the Key Scheduler and then this subkey is added by combining each byte of the AES state (data) with the corresponding byte of the subkey using a bitwise XOR.

The next four steps are repeated nine times. The first step of these repeated steps is the SubBytes step. In this step each byte in the data array is replaced by another byte according to the S-box. The second step in this process is the ShiftRows step. This step shifts the rows of the current AES state by a certain amount. The first row is not shifted, the second row is shifted by one, and so on. The third step in this process is the MixColumns step. During this step, the columns of the current AES state are mixed by multiplying each column by a fixed polynomial. The final step is the Add Round Key step again.

After these four steps have been repeated nine times, the SubBytes step is performed again followed by the ShiftRows step. The final step in the process is the Add RoundKey step one final time. After this process has been completed, the input data has been transformed into cipher text**.**

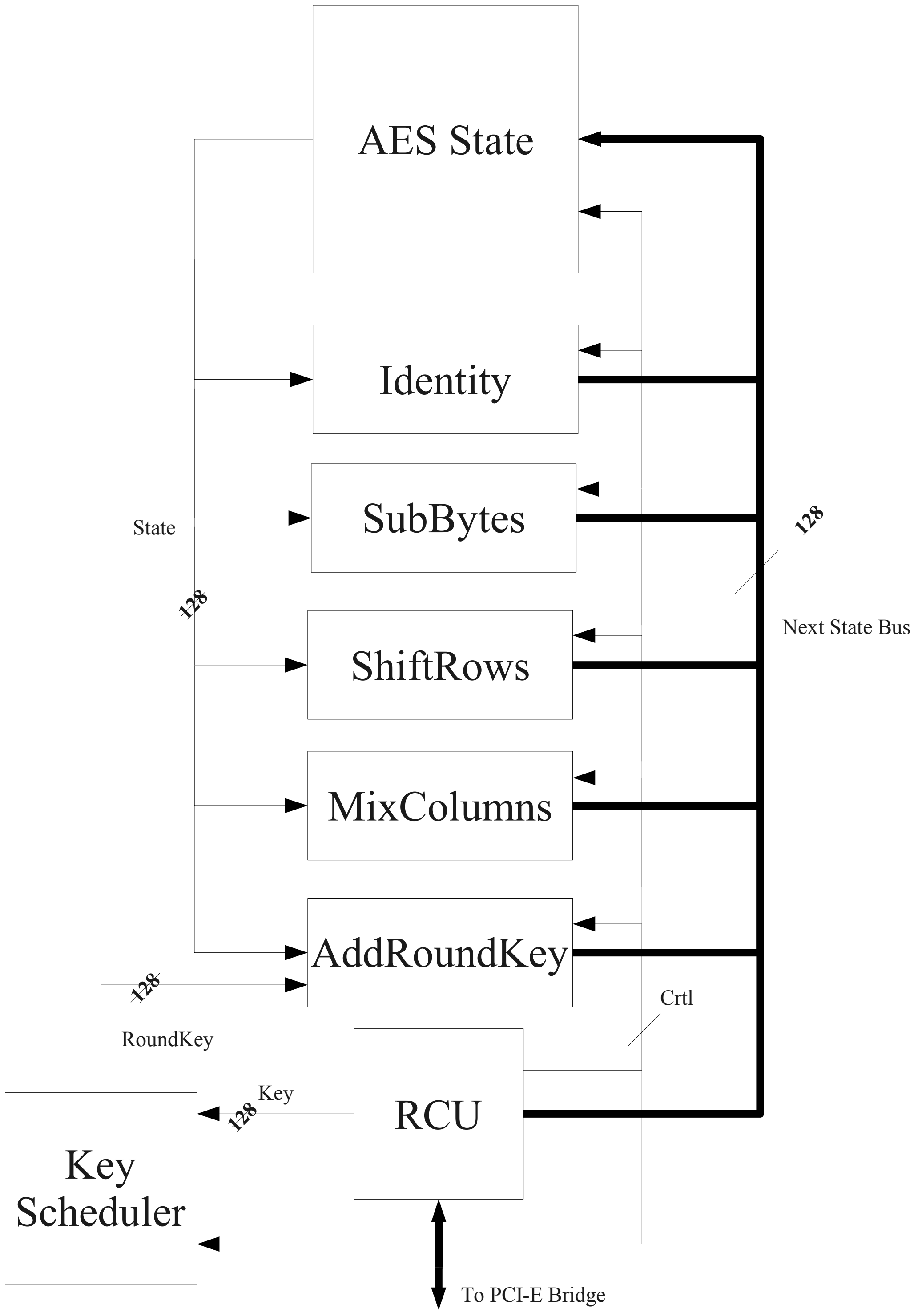
Requirements

The primary optimization for the AES core is speed and high-bandwidth. We are aiming for a 250 MHz clock and as few clocks per byte as possible. This requirement is important in achieving its primary advantage over a software implementation. Having met these requirements, we will target small area, aiming for 3mm x 3mm. The intended use of the AES core dictates that it must use an interface that is common in today's systems; we chose PCI Express because of its widespread adoption, low pin count, and longevity in the near future. PCI Express conformance will be pursued to the extant where the basics of the interface are implemented. The external PCI Express PHY IC takes care of the 2.5Gb/s bitstream, letting the ASIC use a parallel data interface at 250 MHz. The external IC uses differential signaling so some transceiver will be used to interface it with the CMOS ASIC. Some subset of the full PCI Express logical layer will be implemented that lets our AES core communicate with the host device.

Initial Design

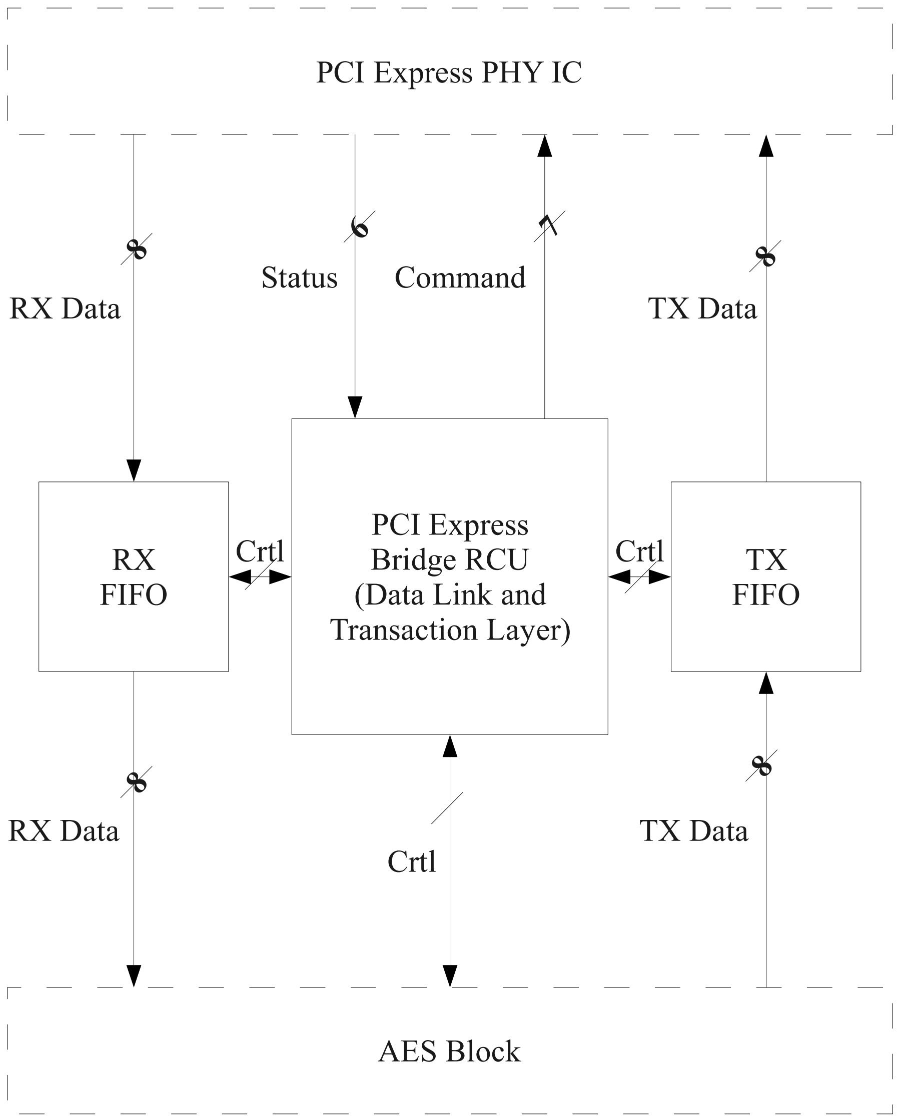
Block Diagrams

AES Core



The AES core is the central block of the ASIC. The AES state contains 128 DFF with the current AES state and clocks in the next AES state each clock cycle. This state is output to all of the sub blocks. These sub blocks are controlled via the RCU. When the RCU instructs them to do so, they output the new AES state onto the bus. The Identity block simply mirrors its input and is used for wait states. In addition to controlling the AES sub blocks, the RCU implements the overall operation of the ASIC according to the state diagram outlined in section 2.1. To do so, the RCU interfaces with the PCIe block in the ASIC, transmitting and receiving data as necessary. The key scheduler block expands the key to all of the round keys as specified in the AES standard. The sub blocks SubBytes, ShiftRows, MixColumns, and AddRoundKey perform the operations specified in the AES standard and outlined in section 2.1.

PCI Express Core



The PCI Express Core is the portion of the ASIC that bridges the AES core with the externa PCIe PHY IC. The main function of the block is to control the Data Link layer and Transaction layer of the PCIe protocol. The Physical layer is handled by the external IC. The exact manner in which the PCIe frames are handled has yet to be determined. Implementing the whole PCI Express protocol would be too laborious for this project so a smaller subset of PCI Express will be implemented. This subset will still allow for the needed functionality between the host and the AES core. FIFOs will be used to help buffer the data so that the AES core can operate asynchronously from the PCIe transmission and receiving.

Timeline

Week 1 (3/9/09)	-Full proposal due -meet as group to write proposal -research PCIexpress protocol
Week 2 (3/16/09)	Spring Break -everyone research/read about PCIe, AES
Week 3 (3/23/09)	-have all AES sub-blocks coded -Jevin: SubBytes, Bus -Zack: ShiftRows, AddRoundKey, Identity -Matt: KeyScheduler, MixColumns -write individual block test benches -Jevin: KeyScheduler, MixColumns -Zack: SubBytes, Bus -Matt: ShiftRows, AddRoundKey, Identity -finalize RCU logic, begin coding -Zack: RCU coding
Week 4 (3/30/09)	-finalize PCIe “base” spec -Jevin will do primary research, group will decide -begin coding PCIe blocks -Jevin: Bridge block -Zack: FIFO blocks -Matt: Bridge block -finish RCU coding -Matt will assist Zack if RCU is not yet done -test bench top-level AES core -Matt: write test bench -write Python script to verify AES correctness -Zack will do this, make sure it passes published test vectors -begin preparing Design review diagrams/documents
Week 5 (4/06/09)	-Week of Design Review -finish all diagrams and documents -each person will make diagrams for the blocks they coded -group will write other various sections -finish coding PCIe, begin test benches
Week 6 (4/13/09)	-finish PCIe test benches -each person will finish assignments from last week -combine PCIe and AES blocks -begin overall top-level test benching -Jevin will write main test bench -everyone will generate test cases -preliminary layout, routing -Matt will do this

Week 7 (4/20/09)	-review preliminary layout, optimization if needed to meet specs -debugging by everyone, slack time -begin preparing final presentation materials -Matt and Zack will focus on AES materials -Jevin will focus on PCIe materials
Week 8 (4/27/09)	-Week of Final Presentation -complete all final presentation materials -make sure all code is completed, working, and commented

Assigned Tasks

Matt: -monitor overall progress and task distribution
-code KeyScheduler, MixColumns blocks
-test benches for ShiftRows, AddRoundKey, Identity blocks
-top-level AES test bench
-assist Zack in finishing RCU
-prepare Design Review materials for his contributions
-code PCIe Bridge block
-generate top-level test cases
-Aid Zack in preparing AES materials for final project

Zack: -code ShiftRows, AddRoundKey, Identity blocks
-test benches for SubBytes, Bus blocks
-code RCU
-write Python script to be used for outside verification of encryption
-prepare Design Review materials for his contributions
-code PCIe FIFO blocks
-generate top-level test cases
-Aid Matt in preparing AES materials for final project

Jevin: -code SubBytes, Bus blocks
-test bench for KeyScheduler, MixColumns blocks
-research and make recommendations for PCIe base spec
-prepare Design Review materials for his contributions
-code PCIe Bridge block
-write top-level test bench
-prepare PCIe materials for final project

Success Criteria

Fixed Criteria

1. (2 points) Test benches exist for all top level components and the entire design. The test benches for the entire design can be demonstrated or documented to cover all of the functional requirements given in the design specific success criteria.
2. (4 points) Entire design synthesizes completely, without any inferred latches, timing arcs, or sensitivity list warnings.
3. (2 points) Source and mapped version of the complete design behave the same for all test cases. The mapped version simulates without timing errors except at time zero.
4. (2 points) A complete IC layout is produced that passes all geometry and connectivity checks.
5. (2 points) Your total area, including I/O pads, is no more than 3mm x 3mm and your required pin count (including power and ground) is no more than 40. For half credit on these criteria, your constraints are 10mm x 10mm and 100 pins.

Design Specific Criteria

1. (1 point) AES block encrypts the standard test vectors correctly.
2. (2 points) AES block encrypts random data identically to a PC-hosted reference AES implementation.
3. (2 points) Can correctly transmit and receive data using the subset of PCI Express that we implement.
4. (2 points) A host can initiate encryption of a data block using PCI Express and read back the correctly encrypted data.
5. (1 point) AES block area is within an order of magnitude of the smallest, published ASIC AES implementation.