LUND
UNIVERSITY

EDA385
Embedded Systems Design Advanced Course

Ahmed Mohammed Youssef (aso10ayo@student.lu.se)
Mohammed Shaaban (aso10mib@student.lu.se)

sEs

Secure Embedded Systems

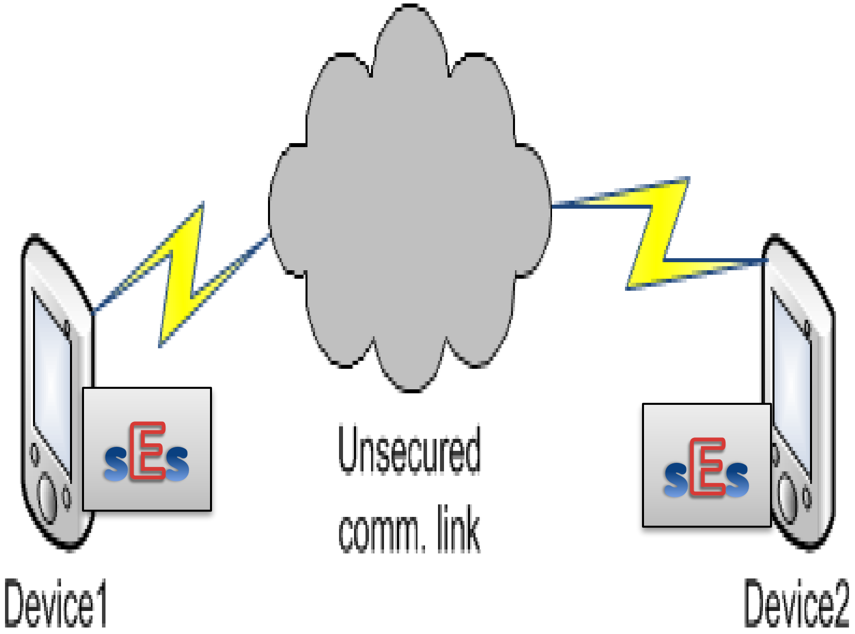
Why SES?

DKI-0079-022 [RF] © www.visualphotos.com

Mobile Phones…

PDAs…

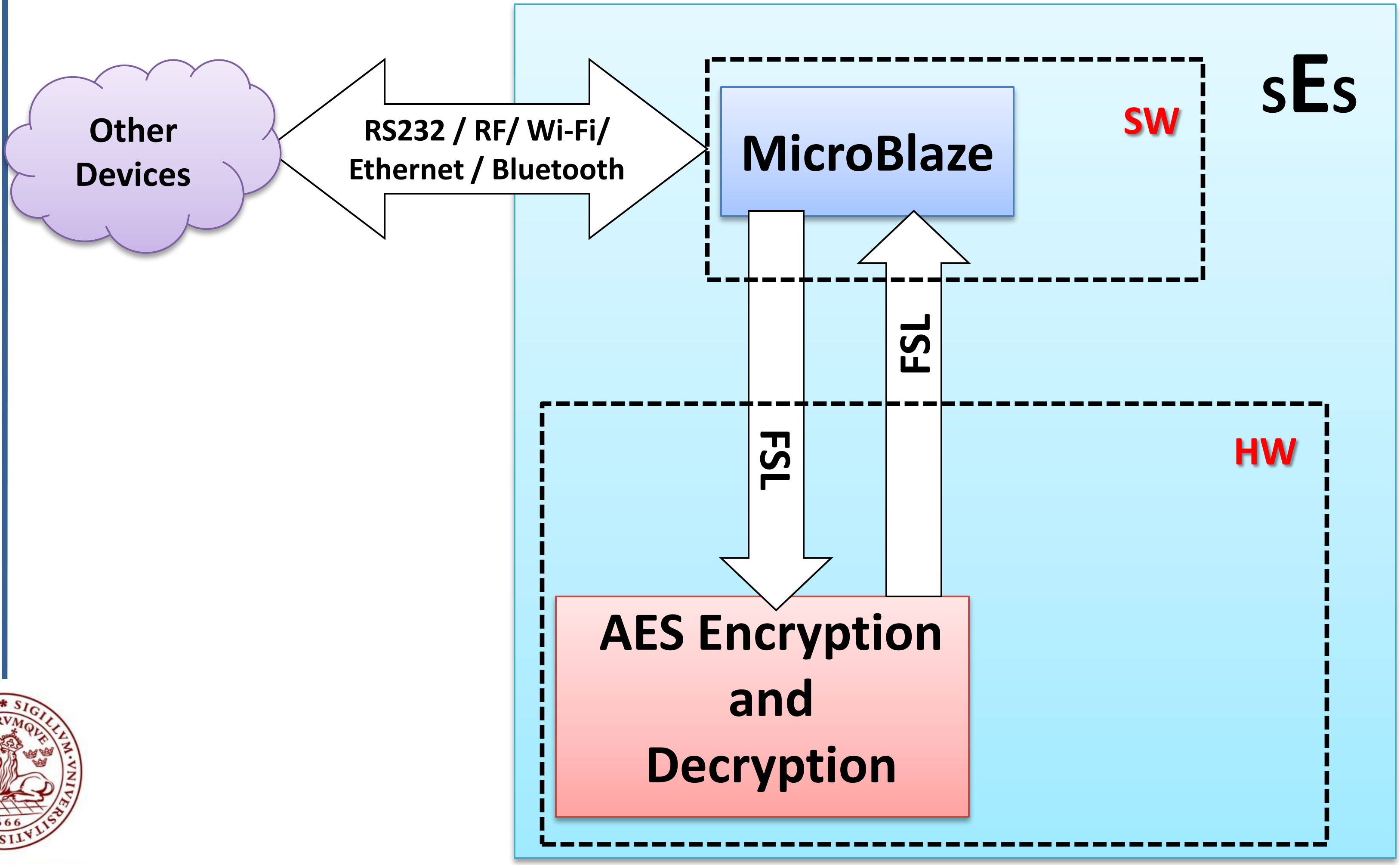
Security Cameras…



Let’s Get Started !!

- System Overview
- Message Life Cycle
- Encryption/Decryption Engine

System Overview


Other Devices
RS232 / RF/ Wi-Fi/ Ethernet / Bluetooth
sEs
MicroBlaze
SW
FSL
FSL
HW
AES Encryption and Decryption

Message Life Cycle – Encryption

Plain text or Cipher ?

Operational Mode ?

Slide Switch

Encryption/ Decryption

Receive a Message – RS232

No. of Blocks & Padding – 128 bit / block

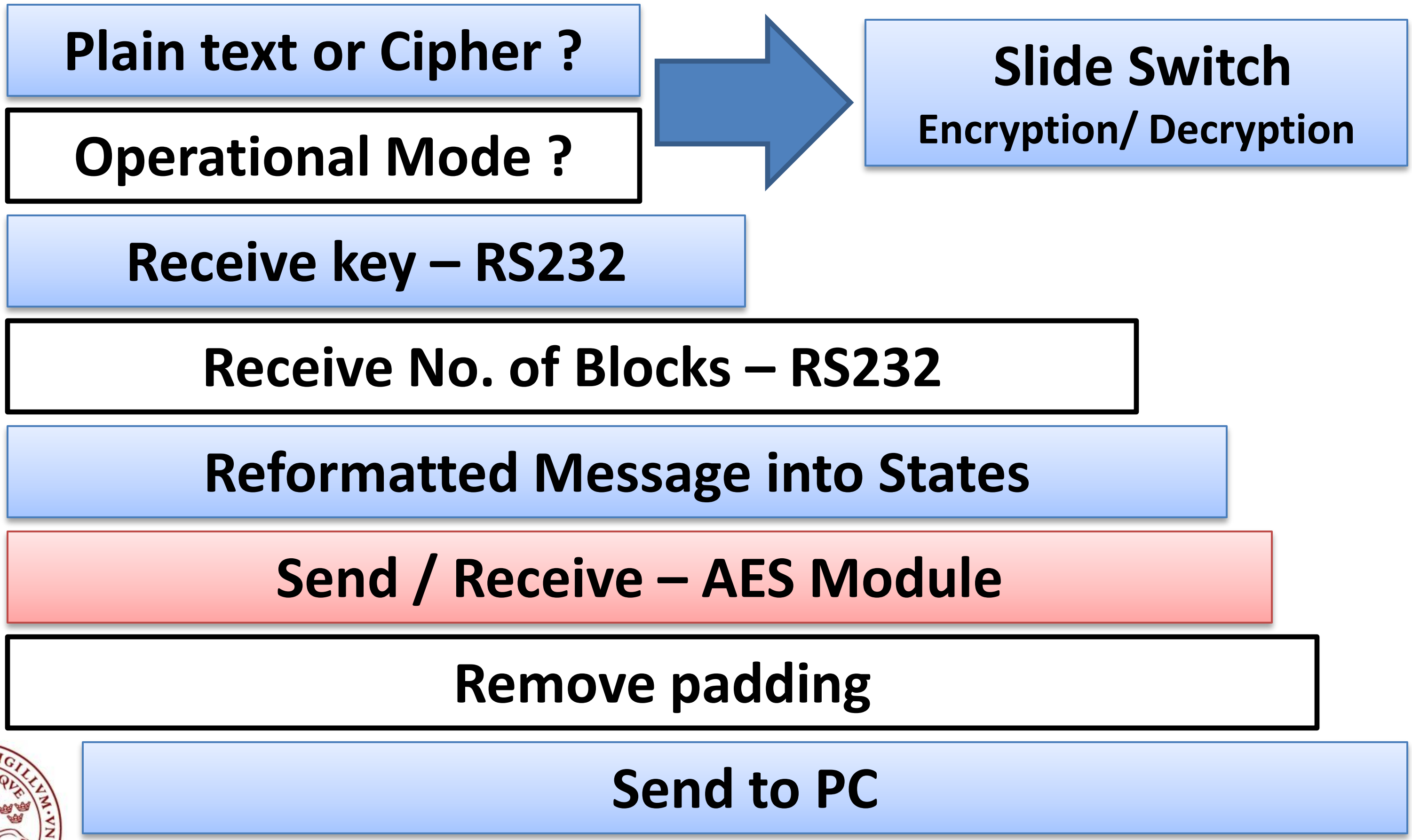
Generate Key – random – 128 bit

Reformatted Message into States

Send / Receive – AES Module

Send to PC Key – No. Blocks – Cipher text

Message Life Cycle – Decryption



Plain text or Cipher ?

Operational Mode ?

Slide Switch

Encryption/ Decryption

Receive key – RS232

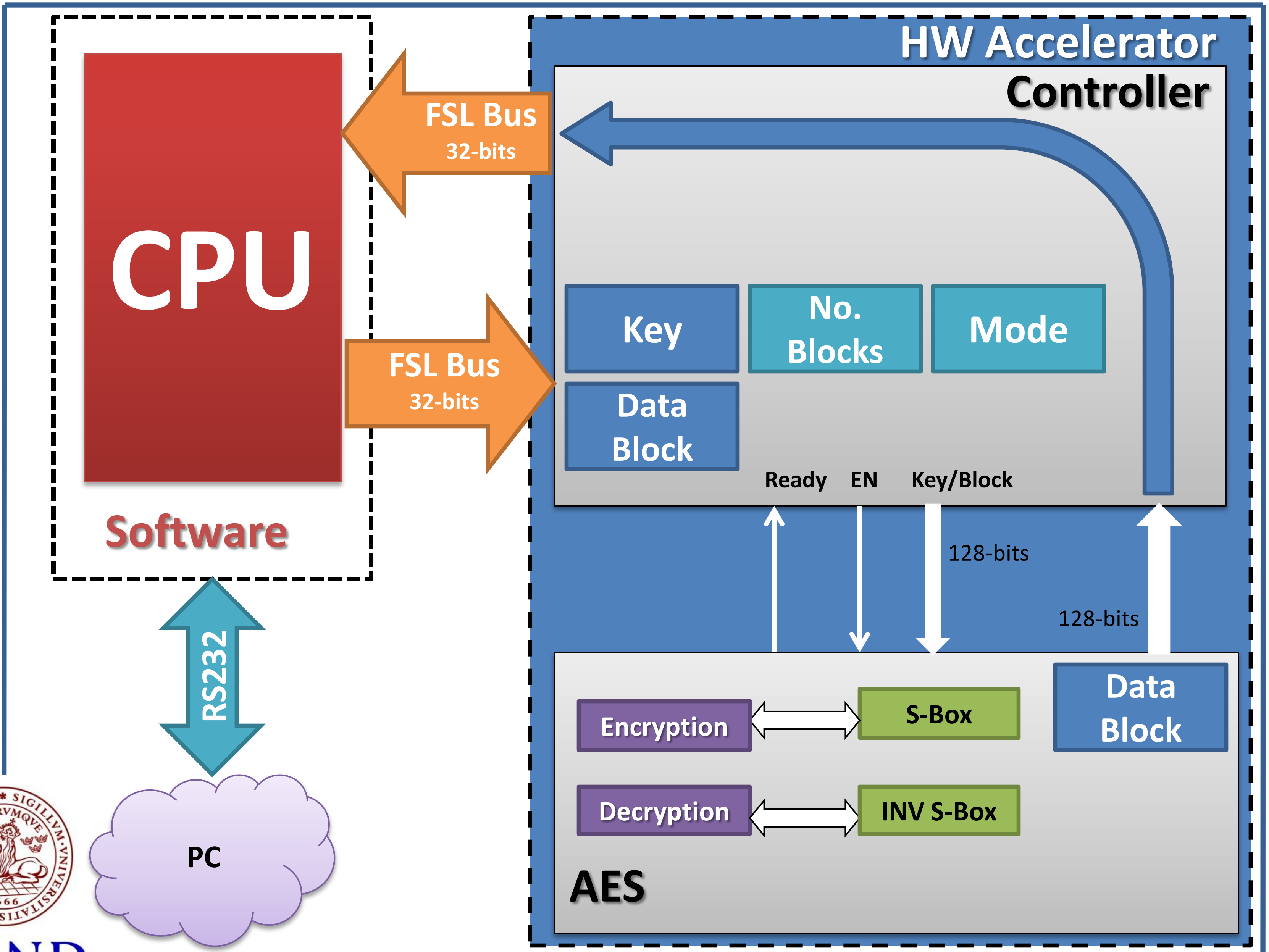
Receive No. of Blocks – RS232

Reformatted Message into States

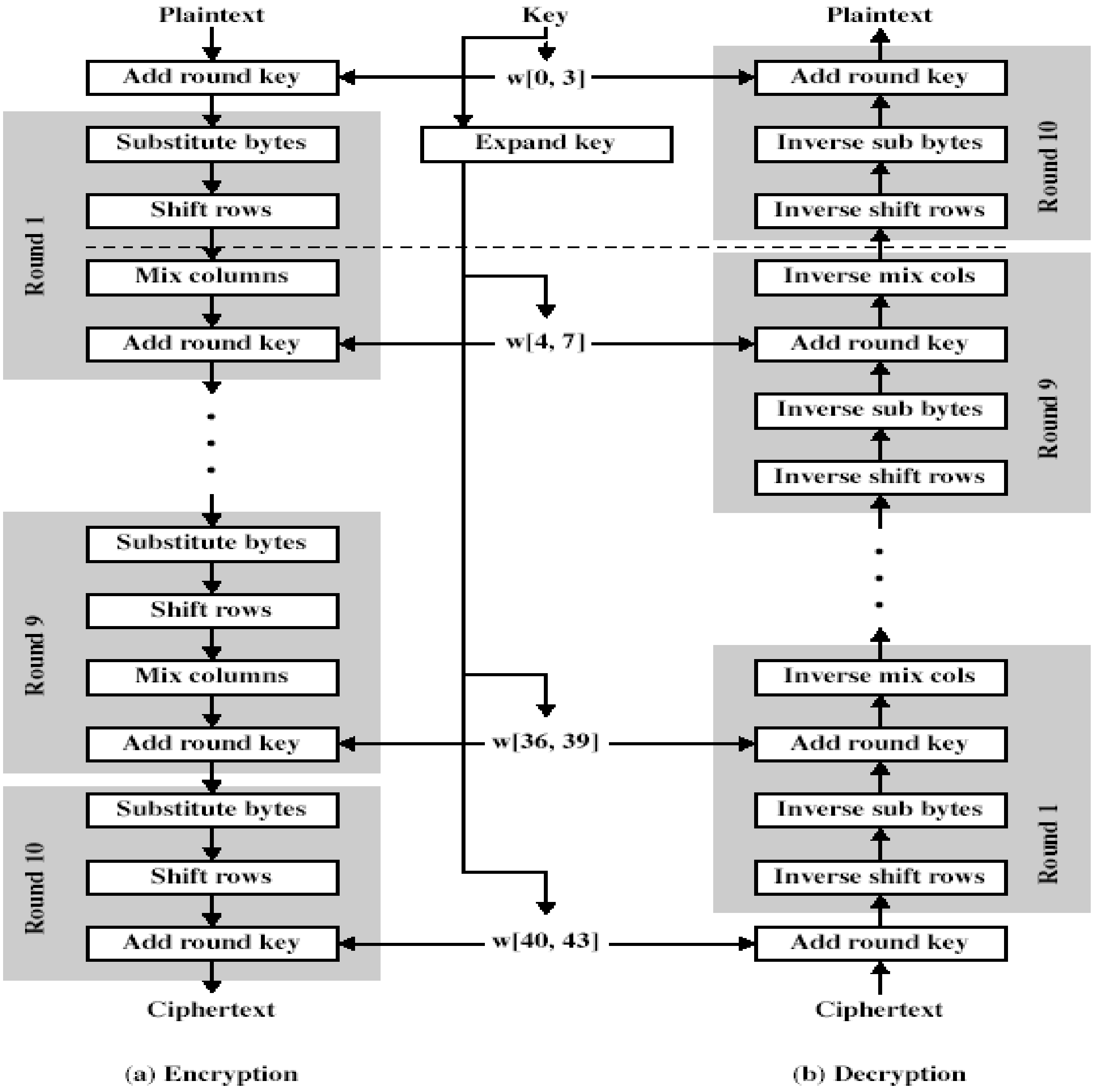
Send / Receive – AES Module

Remove padding

Send to PC


HW Accelerator
Controller
CPU
Software
FSL Bus
32-bits
FSL Bus
32-bits
Key
No. Blocks
Mode
Data Block
Ready
EN
Key/Block
128-bits
128-bits
RS232
PC
Encryption
S-Box
Decryption
INV S-Box
Data Block
AES

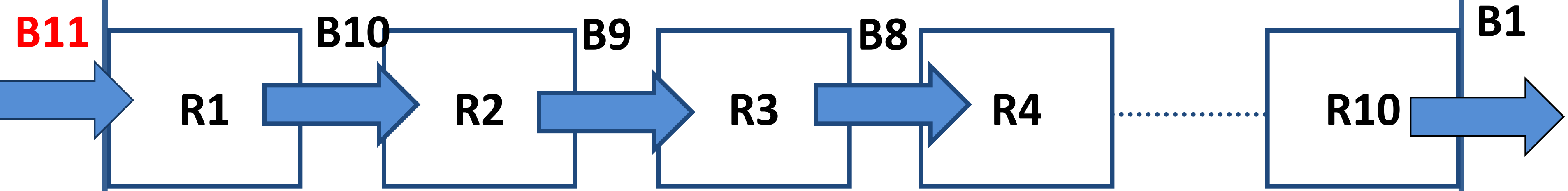
Advanced Encryption Standard (Rijndael - AES)



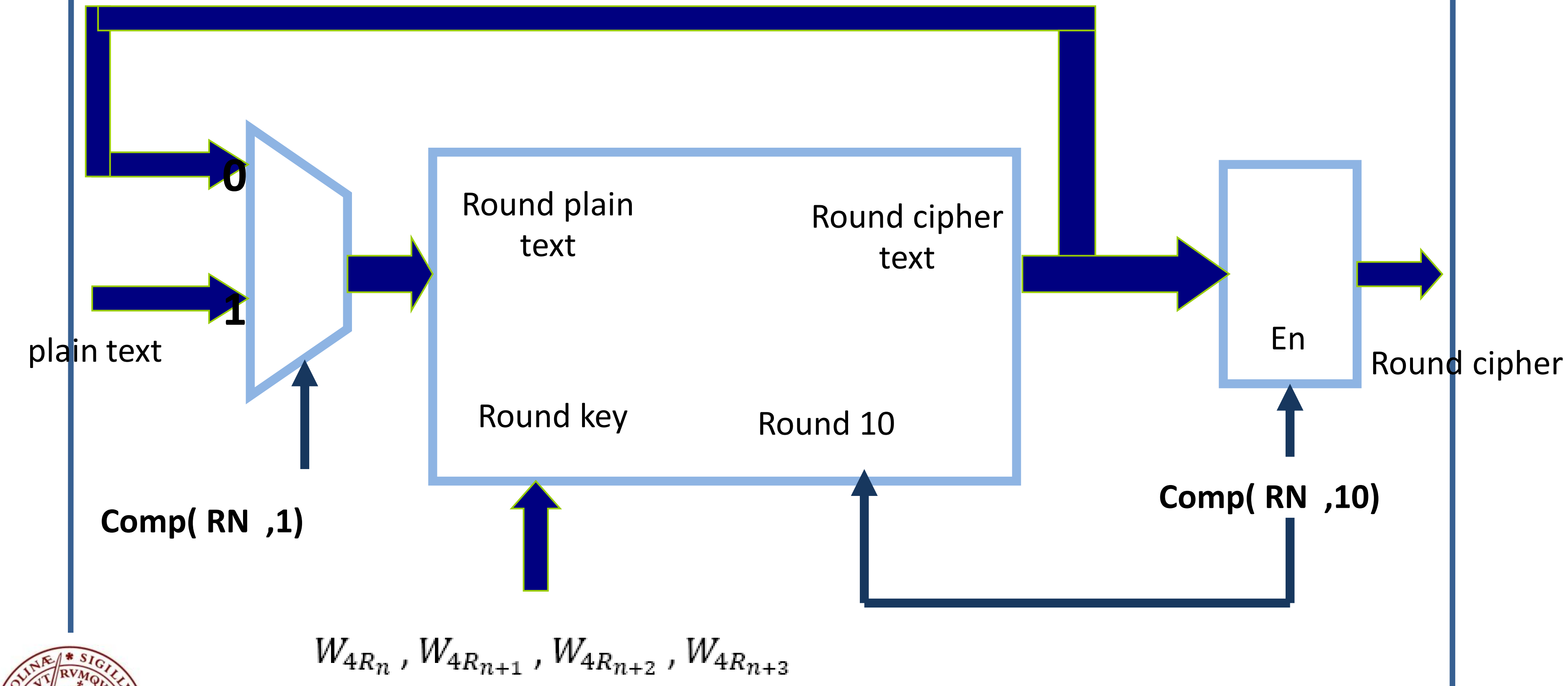
[Cryptography and Network Security Principles and Practices, Fourth Edition, By William Stallings]

Round Implementation

Speed optimized (pipelined) implementation

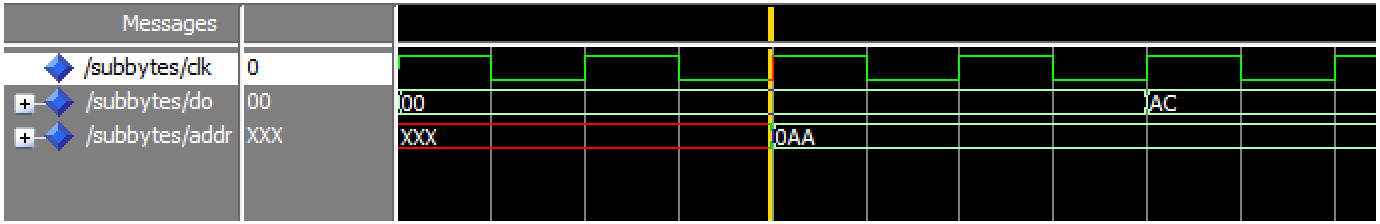


Space optimized Implementation



Different transformations Implementations

- SubBytes :-
  - We used two memories, one for the S-Box (Forward transformation), and one for the Inverse S-Box.
  - Multiple cycles to do one byte transformation



Different transformations Implementations

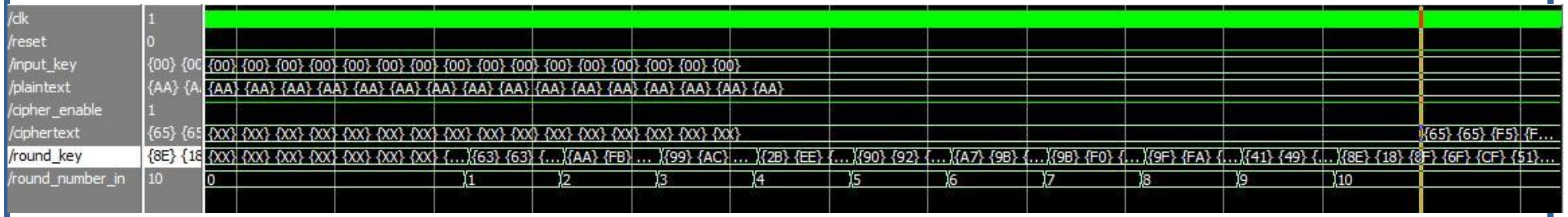
- MixColumns:-
  - Polynomials multiplications over finite field.
  - Each Byte in the state contributes in all Bytes in the output column.
  - Multiplying by two is considered as a Shift left + a conditional XOR with 0x1B.
  - Multiplying by three can be achieved by XORing the multiplied-by-two Byte with the original un-shifted Byte.
  - Inverse transformation consists of multiplication by 0x09, 0x0B, 0x0D, 0x0E.

Different transformations Implementations

- ShiftRows:-
  - Done by only mapping the input Byte in the State to the corresponding Byte in the output state.
- AddRoundKey:-
  - Done by XORing each Byte in the State with the corresponding Byte in the Round Key.

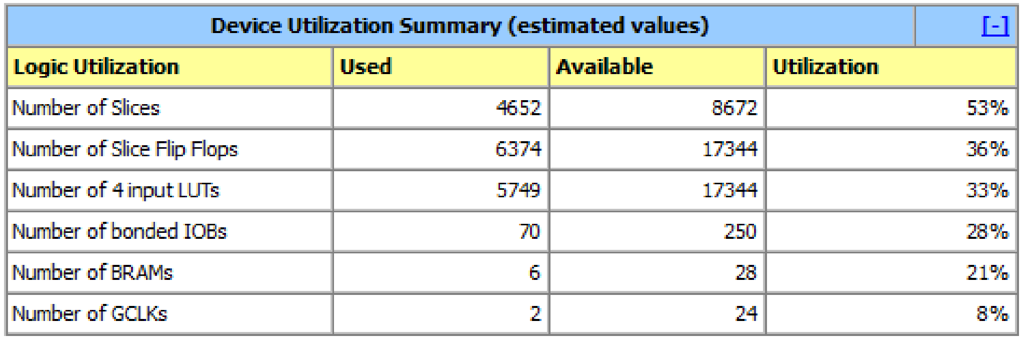
Block processing

- Processing one block took 653 CC, 360 of them are only for memory accesses in the different substitution stages, Which is a significant overhead we can avoid by implementing the S-Box transformation.


/clk 1
/reset 0
/input_key {00} {00 {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00} {00}
/plaintext {AA} {A {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA} {AA}
/cipher_enable 1
/ciphertext {65} {65 {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {XX} {65} {65} {F5} {F...
/round_key {8E} {18 {XX} {XX} {XX} {XX} {XX} {XX} {XX} {... {63} {63} {... {AA} {FB} ... {99} {AC} ... {2B} {EE} {... {90} {92} {... {A7} {9B} {... {9B} {F0} {... {9F} {FA} {... {41} {49} {... {8E} {18} {8F} {6F} {CF} {51}...
/round_number_in 10 0 1 2 3 4 5 6 7 8 9 10

Device Utilization Summary

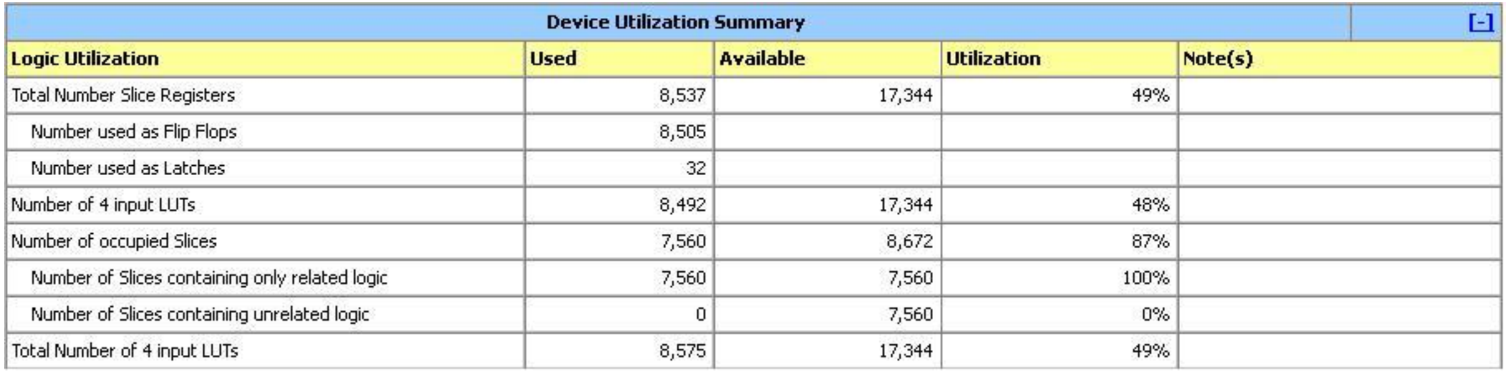
- Below is the device utilization summary for both encryption and decryption and FSL to AES controller.



Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	4652	8672	53%
Number of Slice Flip Flops	6374	17344	36%
Number of 4 input LUTs	5749	17344	33%
Number of bonded IOBs	70	250	28%
Number of BRAMs	6	28	21%
Number of GCLKs	2	24	8%

- Below is the device utilization summary for the Whole System.



Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	8,537	17,344	49%	
Number used as Flip Flops	8,505			
Number used as Latches	32			
Number of 4 input LUTs	8,492	17,344	48%	
Number of occupied Slices	7,560	8,672	87%	
Number of Slices containing only related logic	7,560	7,560	100%	
Number of Slices containing unrelated logic	0	7,560	0%	
Total Number of 4 input LUTs	8,575	17,344	49%	

Conclusion

- Adding security to embedded systems is vital for guaranteeing secure storing/communicating sensitive data.
- AES is a cryptography standard widely used.
- In our project we encrypt/decrypt messages from a PC, however this can be easily modified to encrypt/decrypt any data in a embedded device.
- AES can be implemented with regard to either space or speed optimization, we adopted the space optimized implementation.
- SubBytes stage is a bottleneck as it depends on many memory accesses.

Thank You