# *Implementation of the AES-128 on Virtex-5 FPGAs*

P. Bulens[1] F.-X. Standaert[1] J.-J. Quisquater[1]
P. Pellegrin[2] G. Rouvroy[2]

[1]UCL Crypto Group

[2]intoPIX S.A.

June 11, 2008



# *Outline*

- Introduction
- Cipher Description (if required)
- FPGA
- Architecture
- Virtex-5 / Virtex-4 / Spartan-3
- Implementation Results
- Conclusion



# *Introduction*

- Well-know block cipher
- Efficiency requirements since NIST's call
  $\longrightarrow$ Tuned according to the platform
- Compliant with DCI specification
- Sound up-to-date design



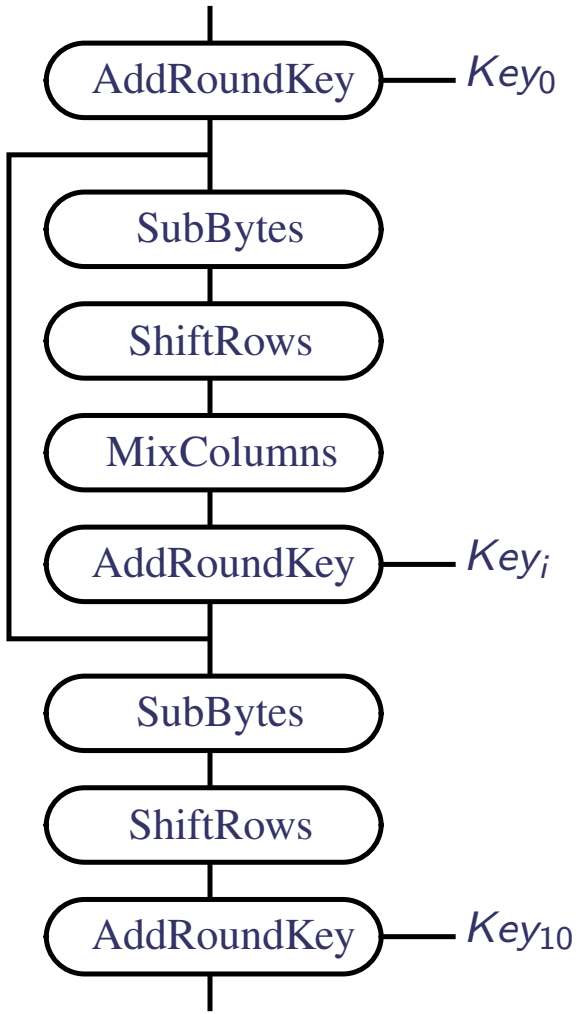
# Cipher Description

State

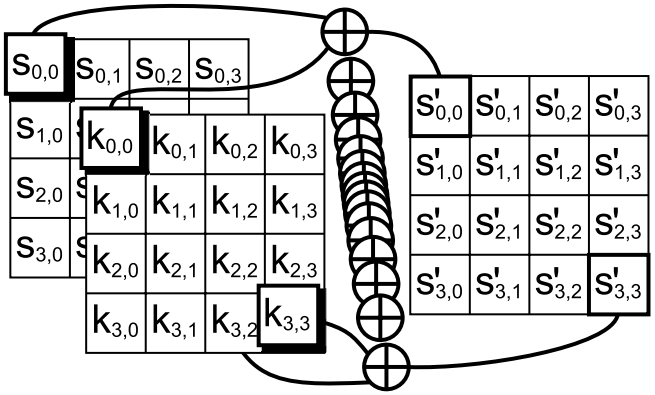
| | | | |
|---|---|---|---|
| $s_{0,0}$ | $s_{0,1}$ | $s_{0,2}$ | $s_{0,3}$ |
| $s_{1,0}$ | $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ |
| $s_{2,0}$ | $s_{2,1}$ | $s_{2,2}$ | $s_{2,3}$ |
| $s_{3,0}$ | $s_{3,1}$ | $s_{3,2}$ | $s_{3,3}$ |

Master Key

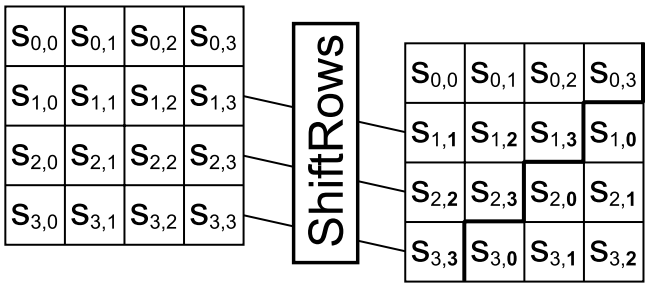
| | | | |
|---|---|---|---|
| $k_{0,0}$ | $k_{0,1}$ | $k_{0,2}$ | $k_{0,3}$ |
| $k_{1,0}$ | $k_{1,1}$ | $k_{1,2}$ | $k_{1,3}$ |
| $k_{2,0}$ | $k_{2,1}$ | $k_{2,2}$ | $k_{2,3}$ |
| $k_{3,0}$ | $k_{3,1}$ | $k_{3,2}$ | $k_{3,3}$ |



# *Cipher Description: AddRoundKey & ShiftRows*

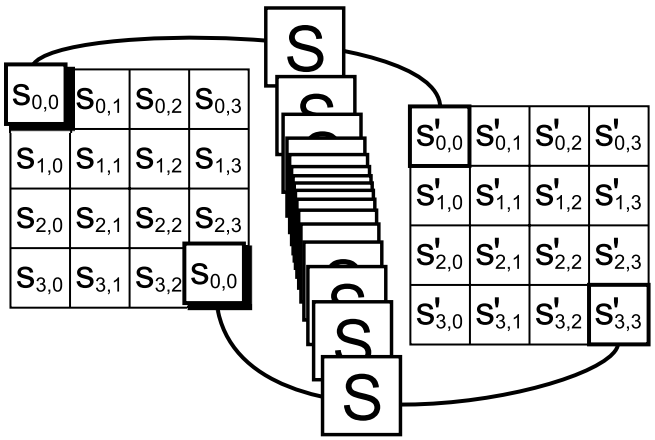



*AddRoundKey*:
$\oplus$ each state byte with corresponding key byte.




*ShiftRows*:
Cyclic shift row $i$,
$i$ bytes to the left

# *Cipher Description: SubBytes*

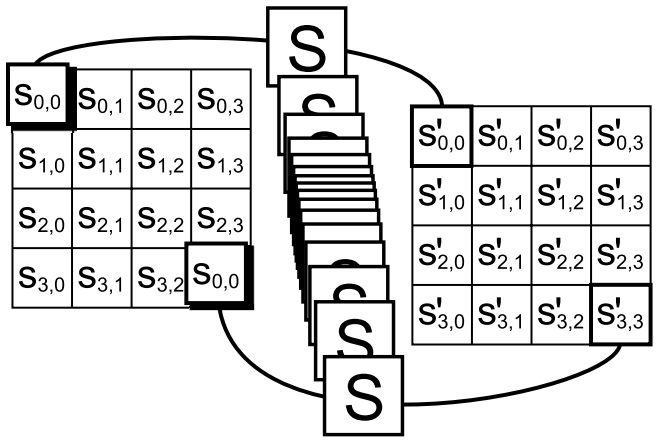



Compose

1. byte inversion in GF($2^8$)
2. affine transformation



# *Cipher Description: SubBytes*



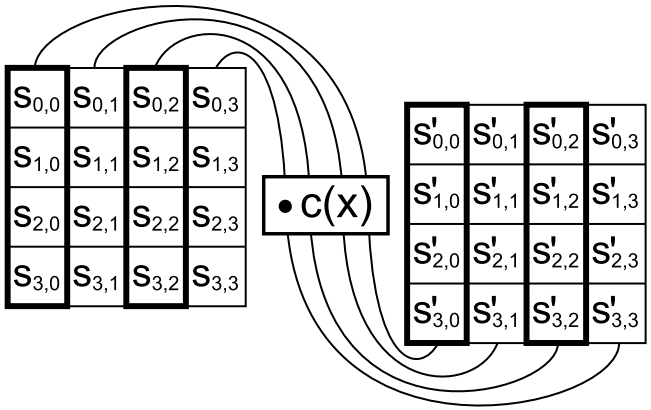

Compose

1. byte inversion in GF($2^8$)
2. affine transformation

Or through S-box:

```
char sbox[256] = {
  0x63, 0x7C, 0x77, 0x7B, ...
  0x30, 0x01, 0x67, 0x2B, ...
  0xCA, 0x82, 0xC9, 0x7D, ...
  0xAD, 0xD4, 0xA2, 0xAF, ...}
```

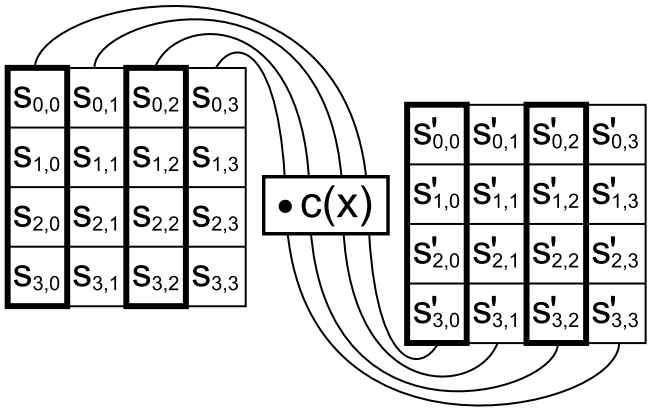
# *Cipher Description: MixColumns*




Consider each column:

1. multiply by $a(x)$
   $= 3x^3 + x^2 + x + 2$
2. reduce modulo $x^4 + 1$

# *Cipher Description: MixColumns*

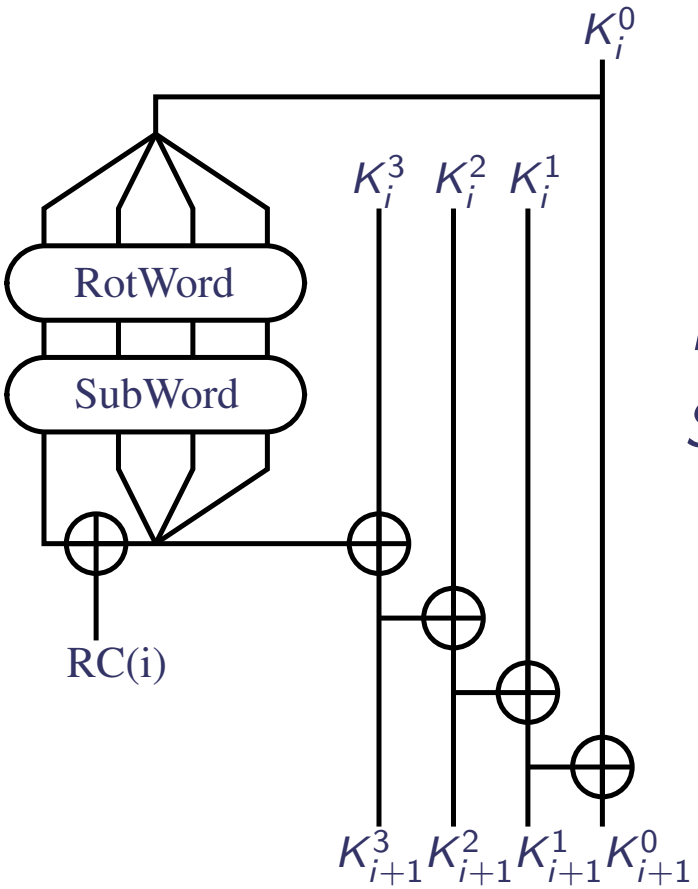



Consider each column:

1. multiply by $a(x)$
   $= 3x^3 + x^2 + x + 2$
2. reduce modulo $x^4 + 1$

$$
\begin{aligned}
s'_{0,c} &= 2s_{0,c} + 3s_{1,c} + s_{2,c} + s_{3,c} \\
s'_{1,c} &= s_{0,c} + 2s_{1,c} + 3s_{2,c} + s_{3,c} \\
s'_{2,c} &= s_{0,c} + s_{1,c} + 2s_{2,c} + 3s_{3,c} \\
s'_{3,c} &= 3s_{0,c} + s_{1,c} + s_{2,c} + 2s_{3,c}
\end{aligned}
$$

# *Cipher Description: KeySchedule*




*RotWord* : cyclic left shift of 1 byte

*SubWord* : S-box each byte

*RC(i)* : $x^{i-1}$ $(\text{mod } x^8 + x^4 + x^3 + x + 1)$

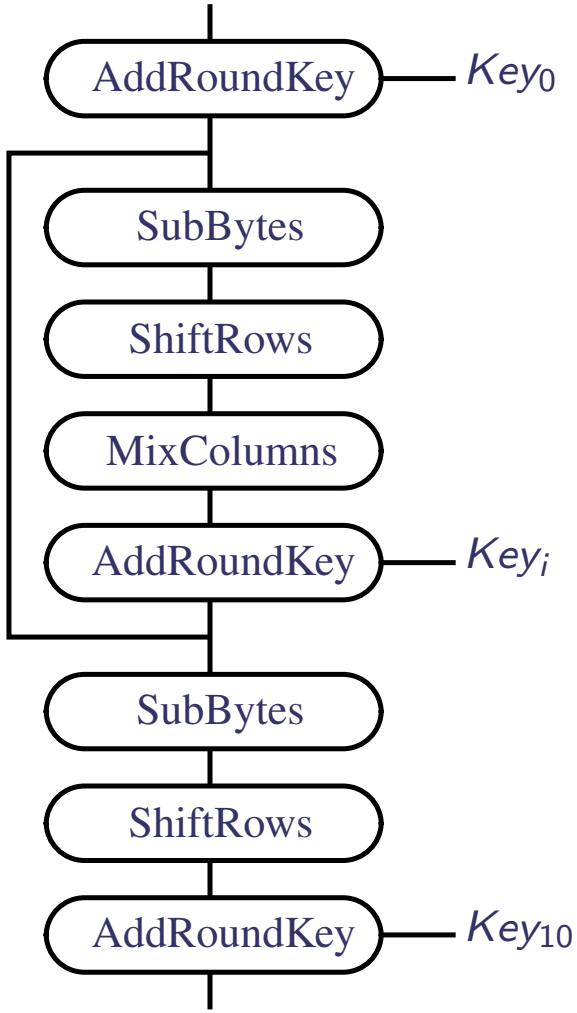
# *Cipher Description*

State

| | | | |
|---|---|---|---|
| $s_{0,0}$ | $s_{0,1}$ | $s_{0,2}$ | $s_{0,3}$ |
| $s_{1,0}$ | $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ |
| $s_{2,0}$ | $s_{2,1}$ | $s_{2,2}$ | $s_{2,3}$ |
| $s_{3,0}$ | $s_{3,1}$ | $s_{3,2}$ | $s_{3,3}$ |

Master Key

| | | | |
|---|---|---|---|
| $k_{0,0}$ | $k_{0,1}$ | $k_{0,2}$ | $k_{0,3}$ |
| $k_{1,0}$ | $k_{1,1}$ | $k_{1,2}$ | $k_{1,3}$ |
| $k_{2,0}$ | $k_{2,1}$ | $k_{2,2}$ | $k_{2,3}$ |
| $k_{3,0}$ | $k_{3,1}$ | $k_{3,2}$ | $k_{3,3}$ |



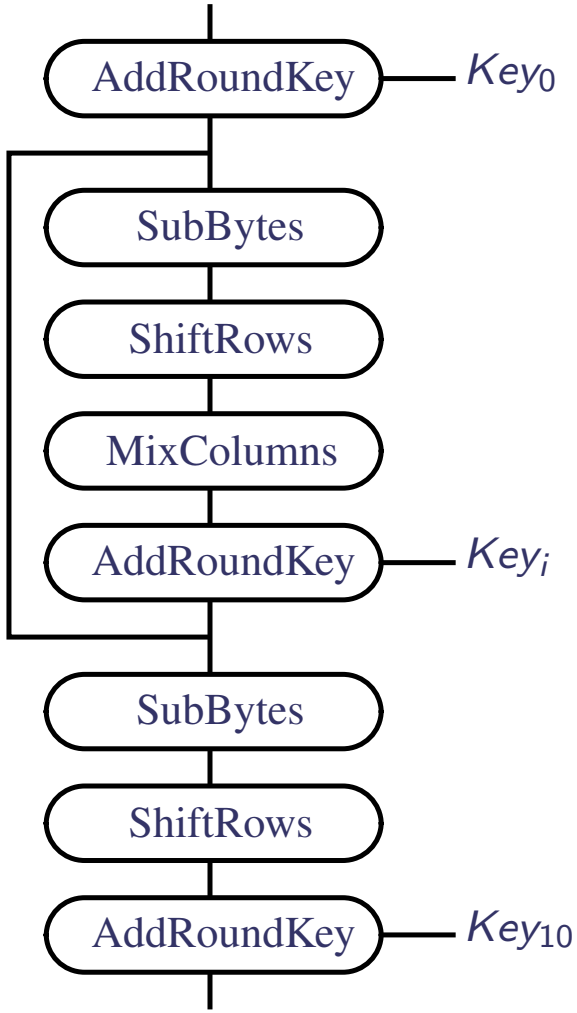
# Cipher Description

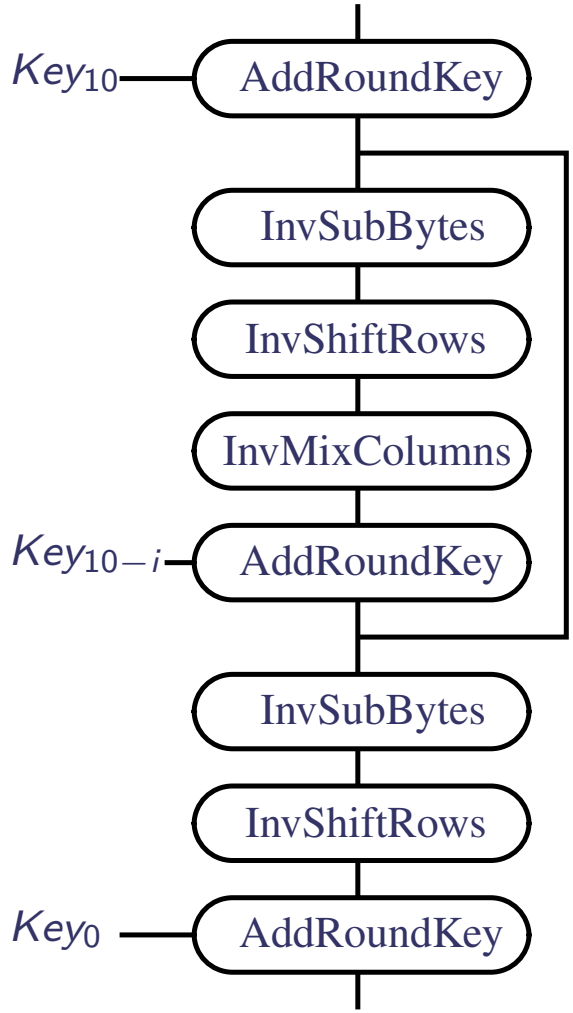
State

| | | | |
|---|---|---|---|
| $s_{0,0}$ | $s_{0,1}$ | $s_{0,2}$ | $s_{0,3}$ |
| $s_{1,0}$ | $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ |
| $s_{2,0}$ | $s_{2,1}$ | $s_{2,2}$ | $s_{2,3}$ |
| $s_{3,0}$ | $s_{3,1}$ | $s_{3,2}$ | $s_{3,3}$ |

Master Key

| | | | |
|---|---|---|---|
| $k_{0,0}$ | $k_{0,1}$ | $k_{0,2}$ | $k_{0,3}$ |
| $k_{1,0}$ | $k_{1,1}$ | $k_{1,2}$ | $k_{1,3}$ |
| $k_{2,0}$ | $k_{2,1}$ | $k_{2,2}$ | $k_{2,3}$ |
| $k_{3,0}$ | $k_{3,1}$ | $k_{3,2}$ | $k_{3,3}$ |



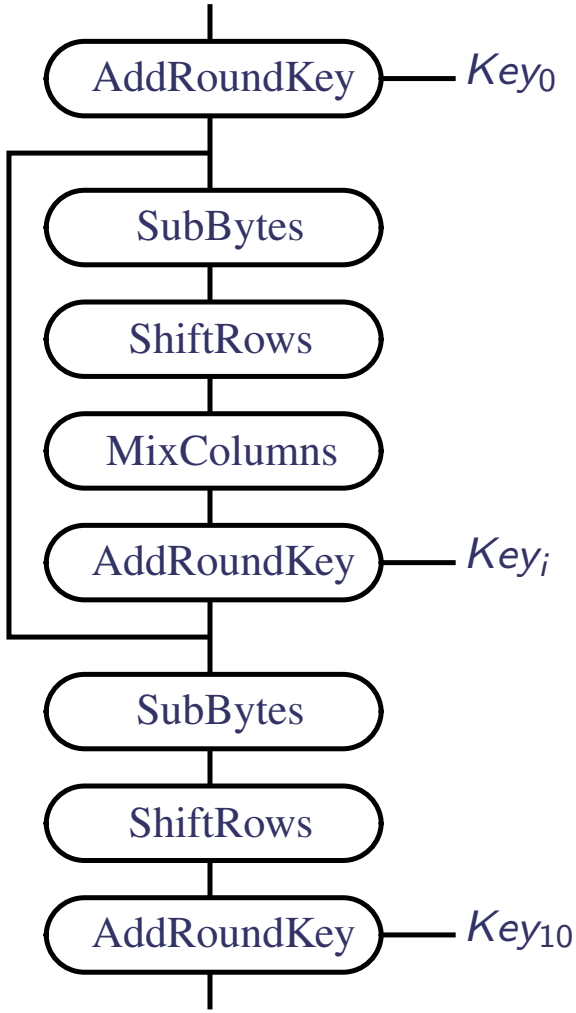
# Cipher Description

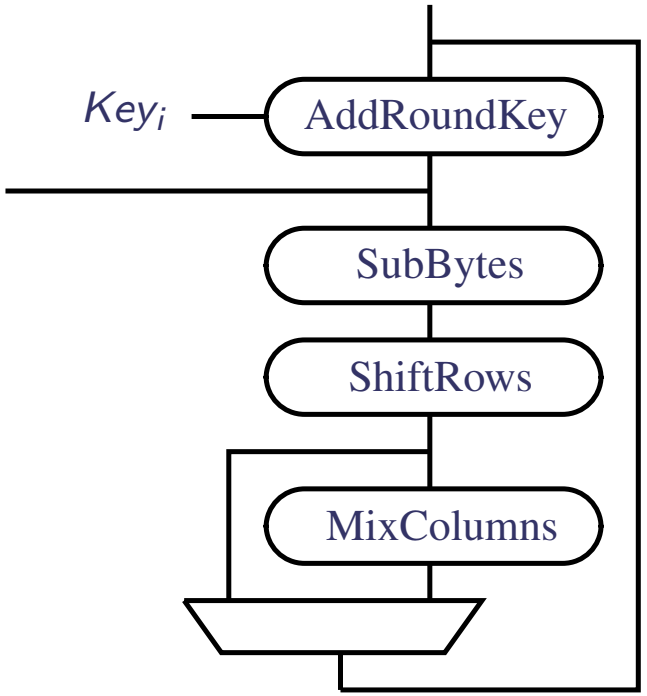
State

| | | | |
|---|---|---|---|
| $s_{0,0}$ | $s_{0,1}$ | $s_{0,2}$ | $s_{0,3}$ |
| $s_{1,0}$ | $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ |
| $s_{2,0}$ | $s_{2,1}$ | $s_{2,2}$ | $s_{2,3}$ |
| $s_{3,0}$ | $s_{3,1}$ | $s_{3,2}$ | $s_{3,3}$ |

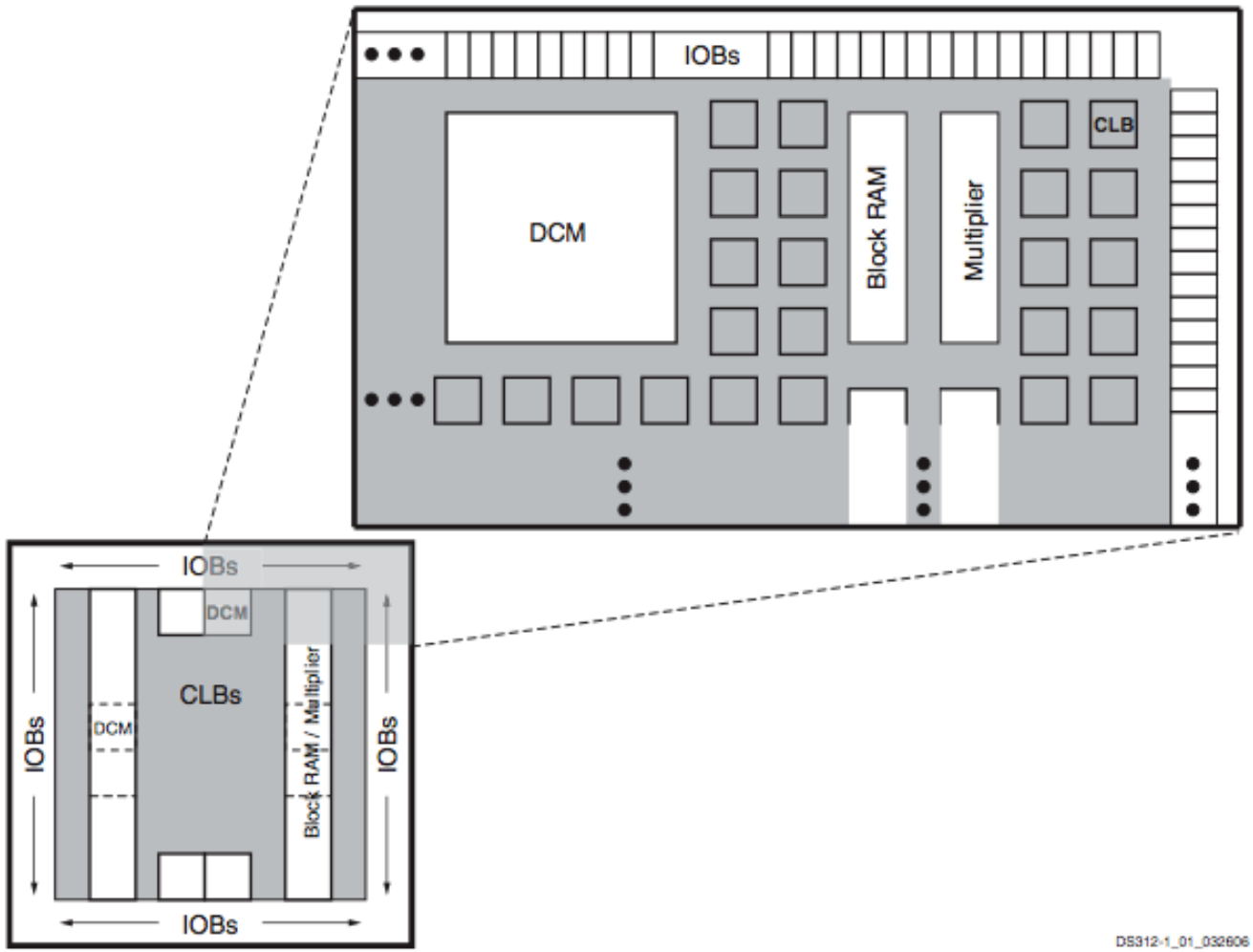
Master Key

| | | | |
|---|---|---|---|
| $k_{0,0}$ | $k_{0,1}$ | $k_{0,2}$ | $k_{0,3}$ |
| $k_{1,0}$ | $k_{1,1}$ | $k_{1,2}$ | $k_{1,3}$ |
| $k_{2,0}$ | $k_{2,1}$ | $k_{2,2}$ | $k_{2,3}$ |
| $k_{3,0}$ | $k_{3,1}$ | $k_{3,2}$ | $k_{3,3}$ |

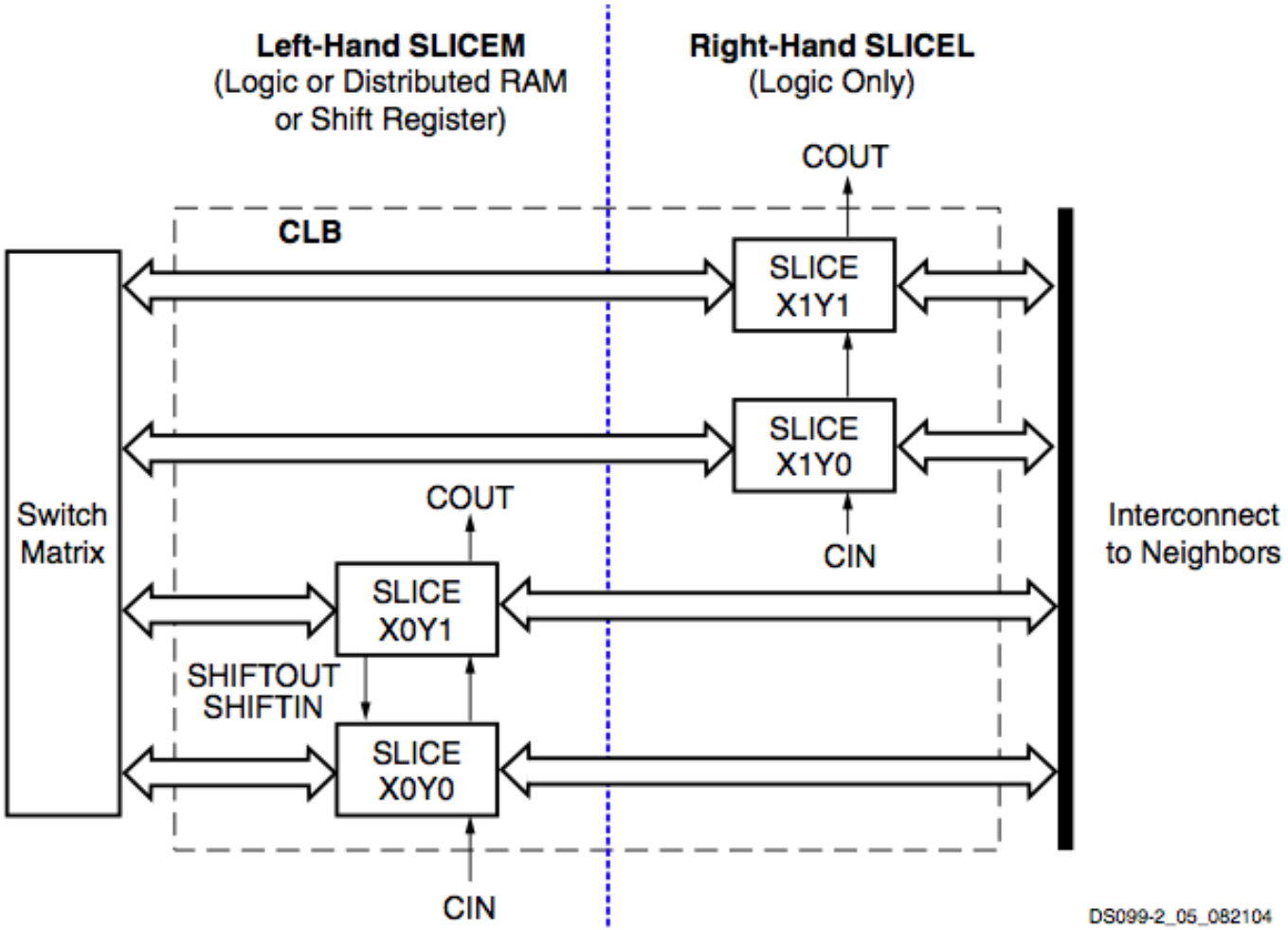


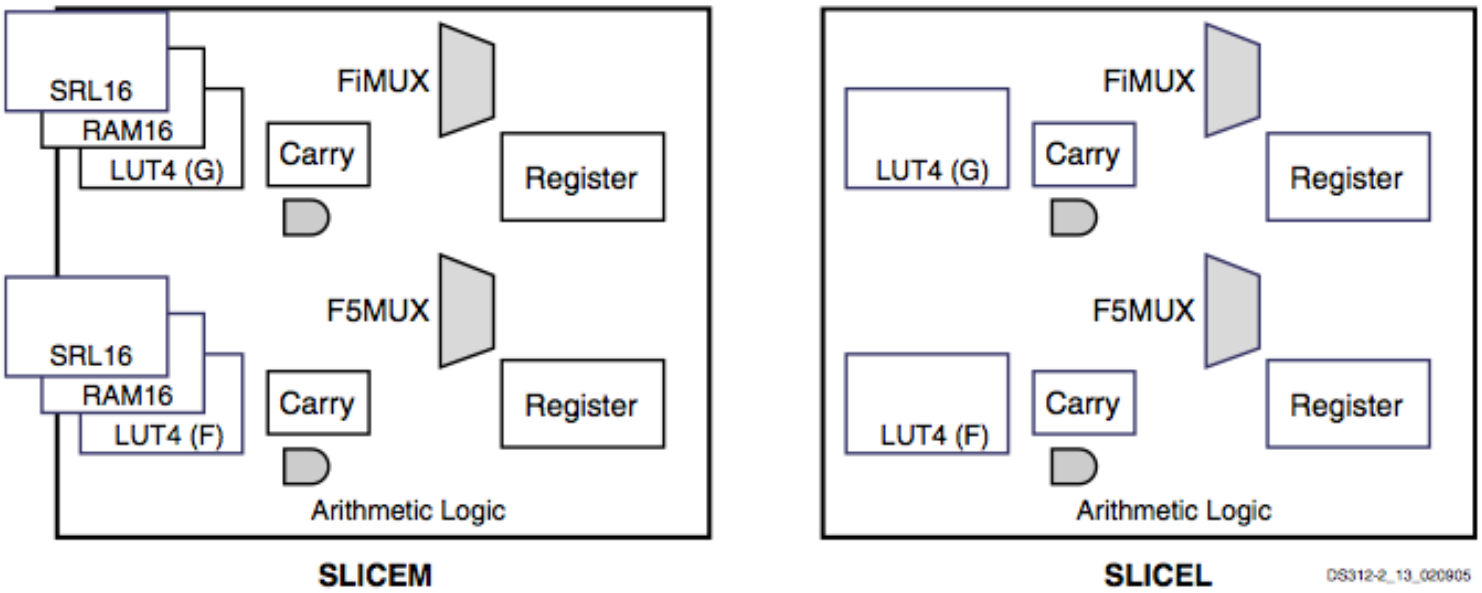
# *FPGA*


IOBs
DCM
Block RAM
Multiplier
CLB
IOBs
DCM
CLBs
DCM
Block RAM / Multiplier
IOBs
IOBs
IOBs
DS312-1_01_032606

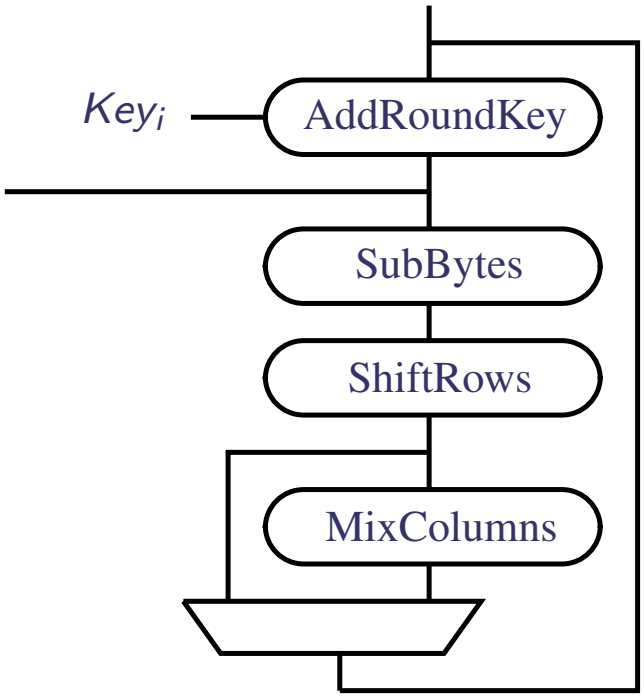
# FPGA

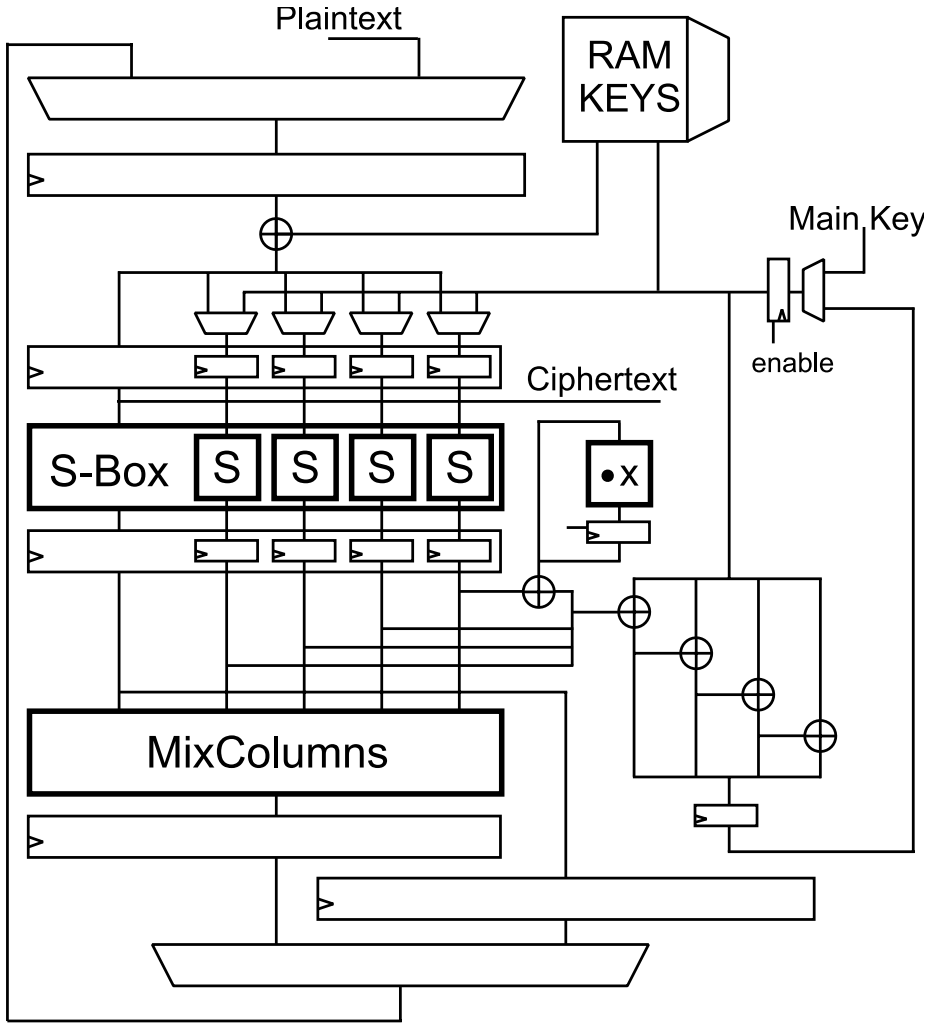

Left-Hand SLICEM
(Logic or Distributed RAM
or Shift Register)
Right-Hand SLICEL
(Logic Only)
COUT
CLB
SLICE
X1Y1
SLICE
X1Y0
Switch
Matrix
COUT
CIN
Interconnect
to Neighbors
SLICE
X0Y1
SHIFTOUT
SHIFTIN
SLICE
X0Y0
CIN
DS099-2_05_082104

# *FPGA*


SRL16
RAM16
LUT4 (G)
FiMUX
Carry
Register
SRL16
RAM16
LUT4 (F)
F5MUX
Carry
Register
Arithmetic Logic
SLICEM
LUT4 (G)
FiMUX
Carry
Register
LUT4 (F)
F5MUX
Carry
Register
Arithmetic Logic
SLICEL
DS312-2_13_020905

# Architecture



# *Architecture: S-box*



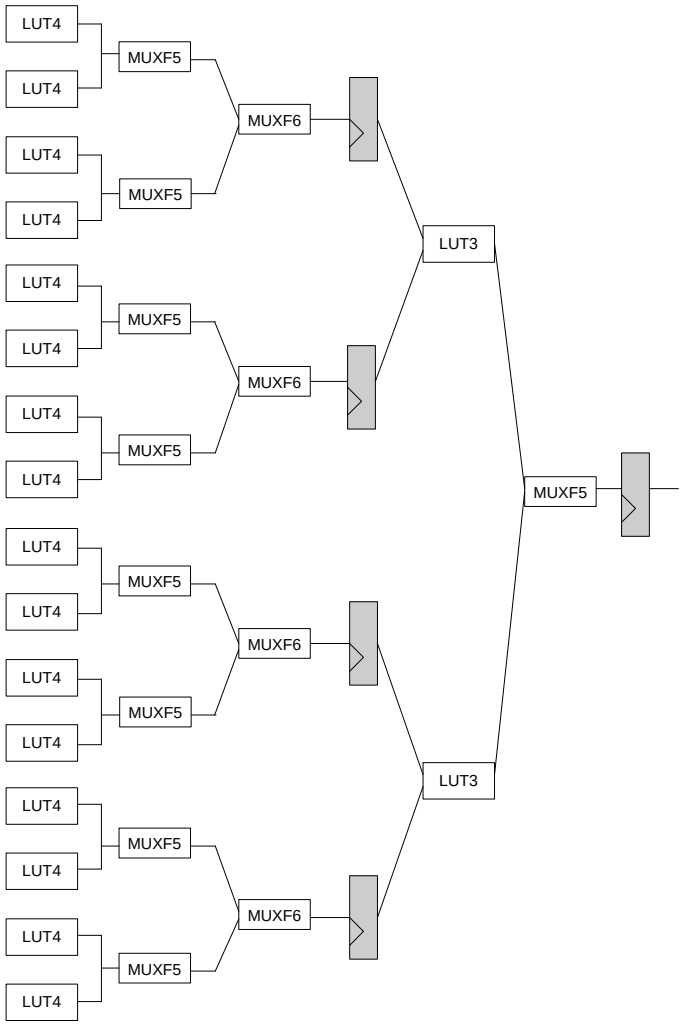
# *Architecture: S-box 1*

*Logic:*

- one S-box per state byte
- 256$x$8-bit



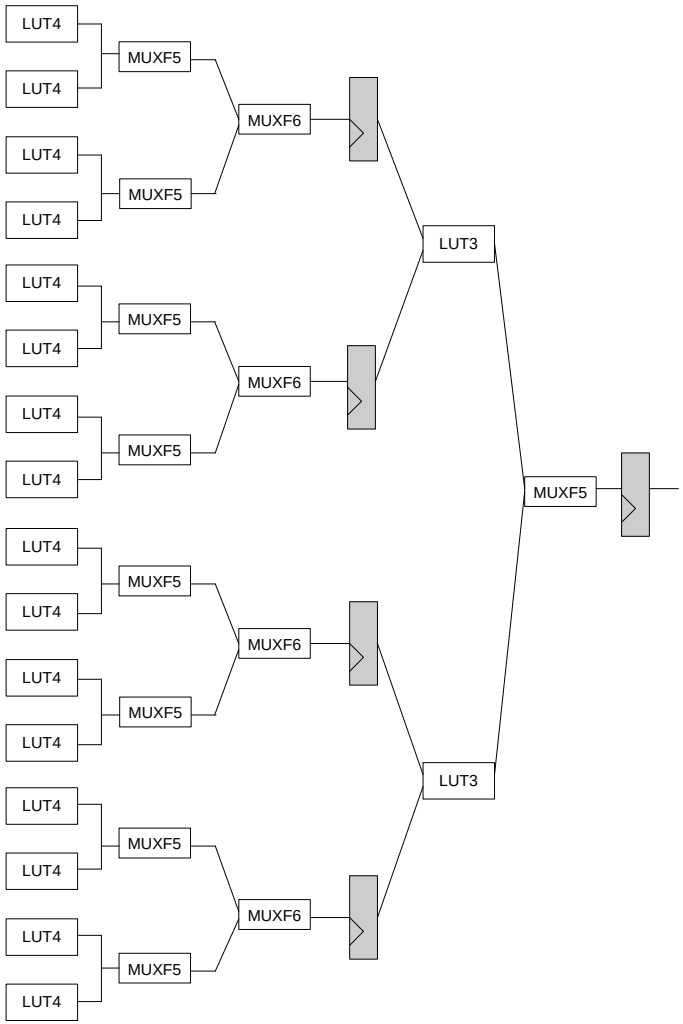
# *Architecture: S-box 1*




*Logic:*

- one S-box per state byte
- 256*x*8-bit

# *Architecture: S-box 1*




*Logic:*

- one S-box per state byte
- 256*x*8-bit
- 2 pipeline stages
- 144 LUTs per S-box

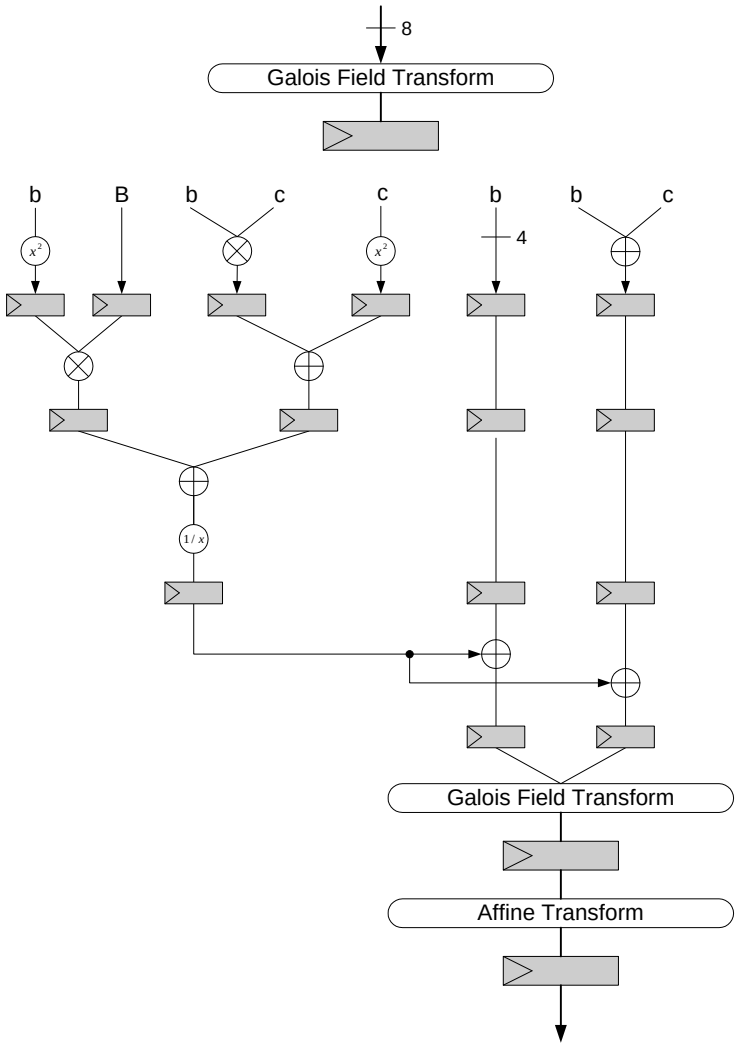
# *Architecture: S-box 2*

*Algorithmic:*

- $\mathrm{GF}(2^8) \rightarrow \mathrm{GF}((2^4)^2)$
- $2^8 \times 8 \rightarrow 2^4 \times 4$



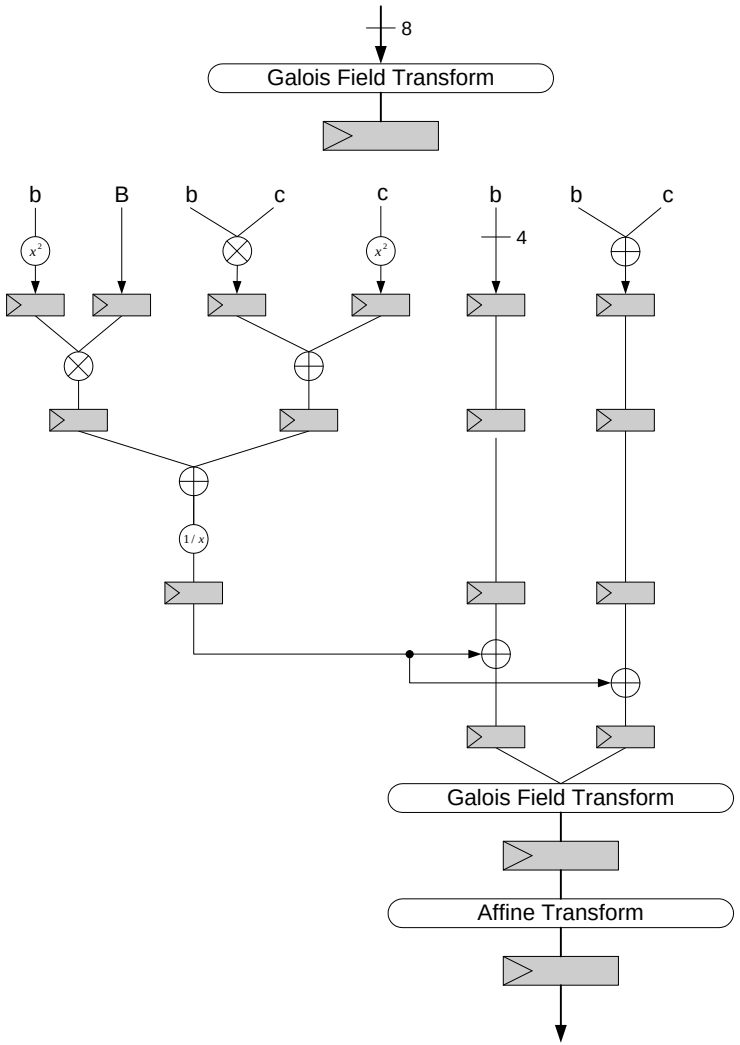
# *Architecture: S-box 2*




*Algorithmic:*

- $GF(2^8) \rightarrow GF((2^4)^2)$
- $2^8 \times 8 \rightarrow 2^4 \times 4$

# *Architecture: S-box 2*




*Algorithmic:*

- $GF(2^8) \rightarrow GF((2^4)^2)$
- $2^8 \times 8 \rightarrow 2^4 \times 4$
- 7 pipeline stages
- 84 LUTs per S-box

# *Architecture: S-box 3*

*RAM:*

- combine SubBytes and part of MixColumns



# *Architecture: S-box 3*

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} \mathrm{SB}(s_0) \\ \mathrm{SB}(s_1) \\ \mathrm{SB}(s_2) \\ \mathrm{SB}(s_3) \end{bmatrix}$$

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = T_0(s) \oplus T_1(s) \oplus T_2(s) \oplus T_3(s)$$

$$T_0(s) = \begin{bmatrix} 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \end{bmatrix}, T_1(s) = \begin{bmatrix} 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ \mathrm{SB}(s) \end{bmatrix},$$

$$T_2(s) = \begin{bmatrix} \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \end{bmatrix}, T_3(s) = \begin{bmatrix} \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \end{bmatrix}$$

*RAM:*

- combine SubBytes and part of MixColumns



# *Architecture: S-box 3*

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} \mathrm{SB}(s_0) \\ \mathrm{SB}(s_1) \\ \mathrm{SB}(s_2) \\ \mathrm{SB}(s_3) \end{bmatrix}$$

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = T_0(s) \oplus T_1(s) \oplus T_2(s) \oplus T_3(s)$$

$$T_0(s) = \begin{bmatrix} 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \end{bmatrix}, T_1(s) = \begin{bmatrix} 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ \mathrm{SB}(s) \end{bmatrix},$$

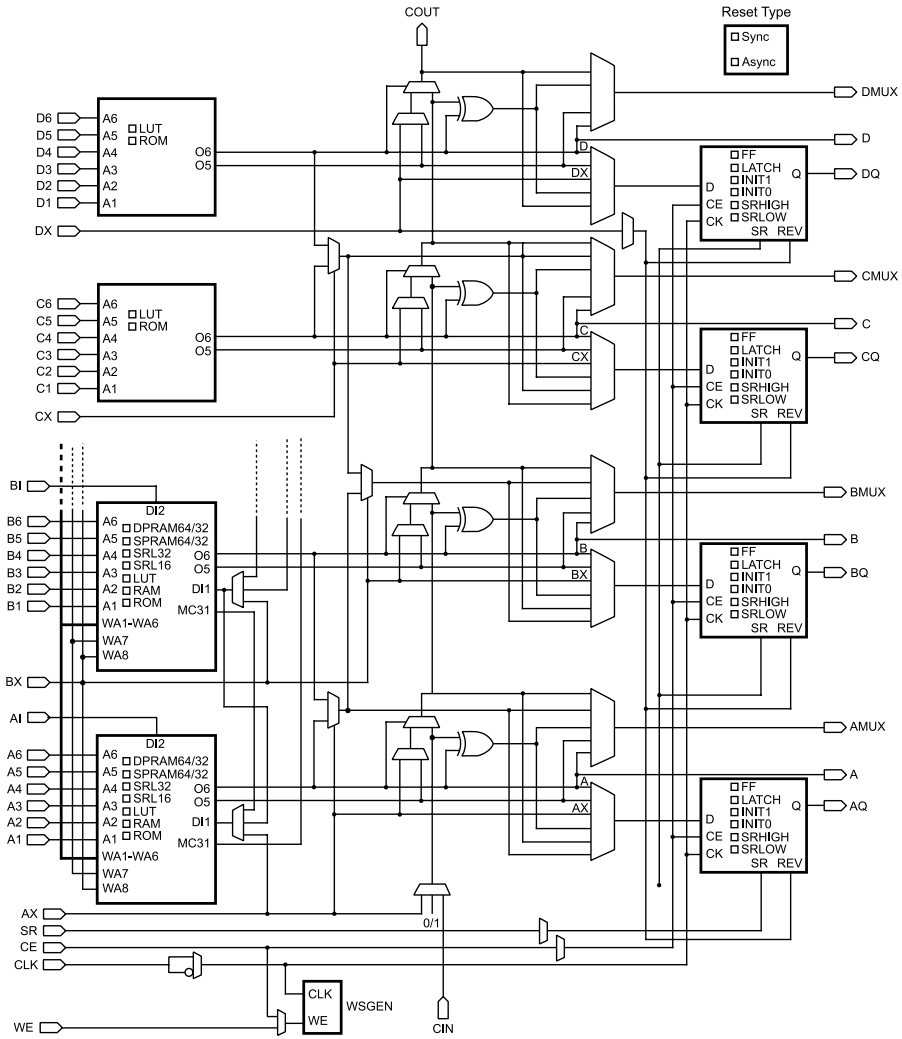
$$T_2(s) = \begin{bmatrix} \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \\ \mathrm{SB}(s) \end{bmatrix}, T_3(s) = \begin{bmatrix} \mathrm{SB}(s) \\ \mathrm{SB}(s) \\ 03 \times \mathrm{SB}(s) \\ 02 \times \mathrm{SB}(s) \end{bmatrix}$$

*RAM:*

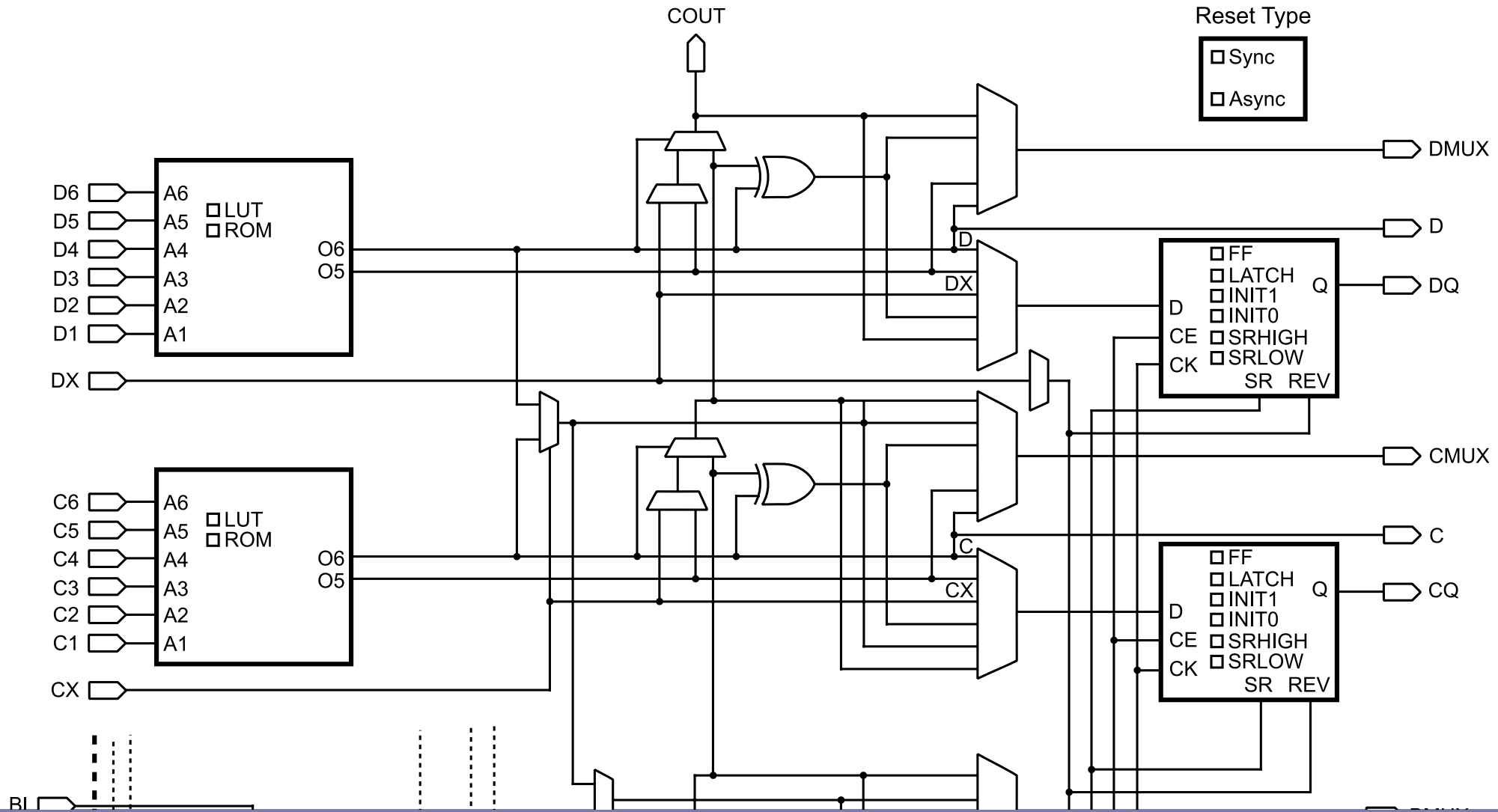
- combine SubBytes and part of MixColumns
- store $T$-tables
- xor in logic



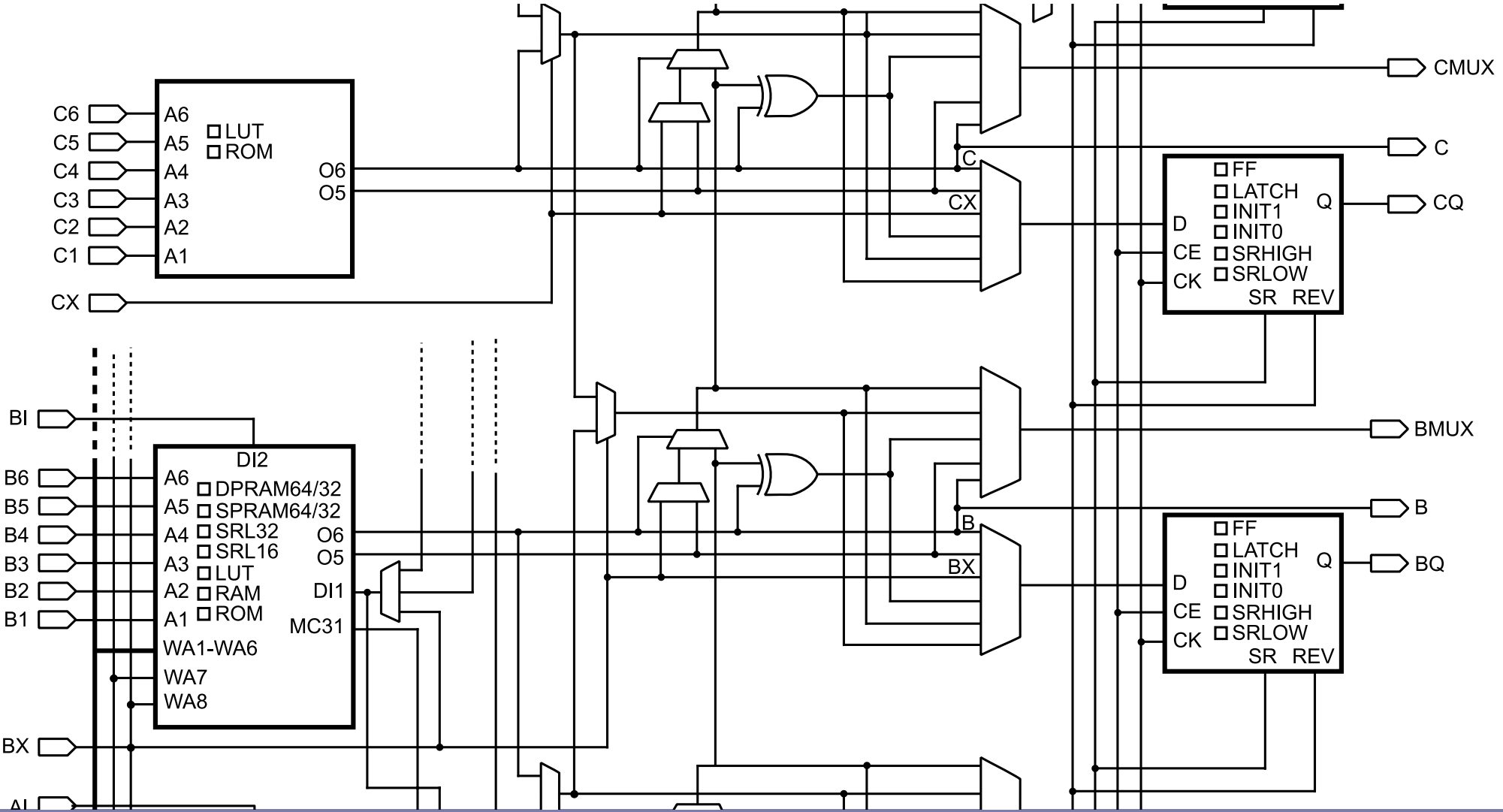
# *Platform: Virtex-5*



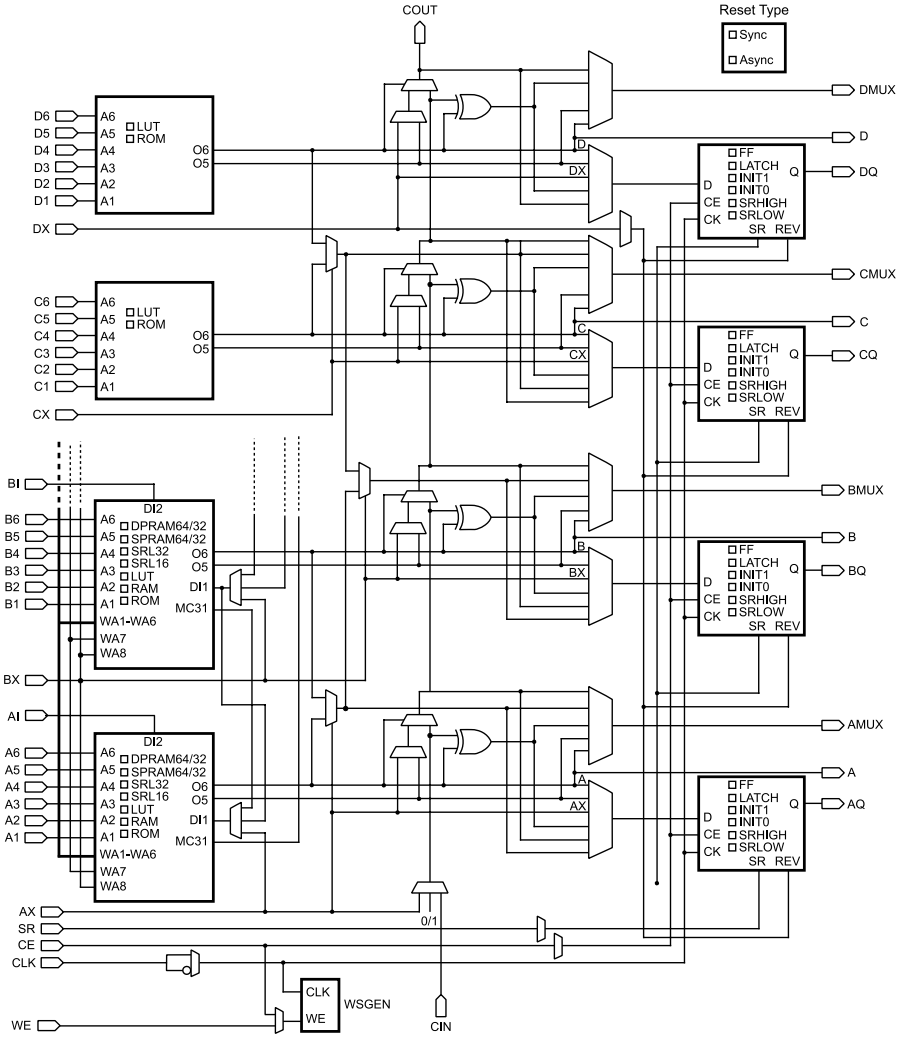
# *Platform: Virtex-5*



# *Platform: Virtex-5*



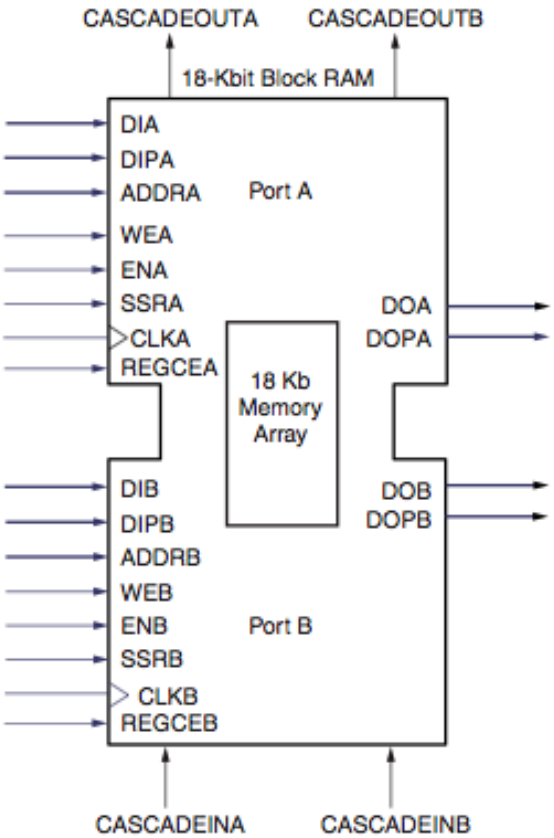
# *Platform: Virtex-5*




*Logic:*

- one S-box per state byte
- 256*x*8-bit
- 1 pipeline stage
- 32 LUTs per S-box

# *Platform: Virtex-4*




*RAM:*

- output latched
- 8 blockRAMs

| Port Data Width | Port Address Width | Depth | ADDR Bus | DI Bus / DO Bus | DI |
|---|---|---|---|---|---|
| 1 | 14 | 16,384 | <13:0> | <0> | |
| 2 | 13 | 8,192 | <13:1> | <1:0> | |
| 4 | 12 | 4,096 | <13:2> | <3:0> | |
| 9 | 11 | 2,048 | <13:3> | <7:0> | |
| 18 | 10 | 1,024 | <13:4> | <15:0> | |
| 36 | 9 | 512 | <13:5> | <31:0> | |



# *Platform: Spartan-3*

| Feature | Spartan-3/ Spartan-3E Block RAM | Spartan-3A/ Spartan-3AN Block RAM | Spartan-3A DSP Block RAM |
|---|---|---|---|
| Individual write-enables for each byte lane in x9, x18, or x36 configurations | No (single write-enable only) | Yes | Yes |
| Special routing resources between block RAM and multiplier for x36 configurations | No | Yes | General Purpose |
| Output register | No | No | Yes |
| Supported by RAMB16 primitive | Yes | Yes | Yes |
| Supported by RAMB16BWE primitive (RAMB16 with byte-level write enable) | No | Yes | Yes |
| Supported by RAMB16BWER primitive (RAMB16BWE with output register) | No | No | Yes |

Keep it in logic



# *Implementation Results*

Post Place & Route results using Xilinx ISE 9.1i

| Device | Slices | BRAM | Freq. (MHz) | Thr. (Gbps) | Thr. / Area (Mbps/slice) |
|---|---|---|---|---|---|
| Virtex-5 | 400 / 550 (800 / 1100) | 0 | 350 | 4.1 | 10.2 / 7.4 |
| Virtex-4* | 700 / 1220 | 8 | 250 | 2.9 | 4.1 / 2.3 * |
| Spartan-3 | 1800 / 2150 | 0 | 150 | 1.7 | 0.9 / 0.8 |



# *Implementation Results: Comparison*

| Device | Datapath | Slices | BRAM | Freq. (MHz) | Thr. (Gbps) | Thr./Area (Mbps/slice) |
|---|---|---|---|---|---|---|
| Spartan-2* | 8 | 124 | 2 | – | 0.002 | 0.02* |
| Virtex-2* | 32 | 146 | 3 | 123 | 0.358 | 2.45* |
| Virtex-E* | 128 | 542 | 10 | 119 | 1.45 | 2.67* |
| Virtex-E | 128 | 2257 | 0 | 169 | 2.0 | 0.88 |
| Virtex-2* | 128 | 387 | 10 | 110.16 | 1.41 | 3.64* |
| Virtex-2 | 128 | 1780 | 0 | 77.91 | 1.0 | 0.56 |
| Virtex-4 | 128 | 18400 | 0 | 140 | 17.9 | 0.97 |
| Virtex-4* | 128 | 700 | 8 | 250 | 2.9 | 4.1 |
| Virtex-5 | 128 | 349 | 0 | 350 | 4.1 | 11.67 |
| Virtex-5 | 128 | 400 | 0 | 350 | 4.1 | 10.2 |



# *Conclusion*

AES-128 on Virtex-5:

- Technology advances: 144 $\searrow$ 32 LUTs for one S-box
  - equivalent to removing RAM
  - or reducing area by 50%
- Efficient architecture suitable for Gbps applications
  - Digital Cinema



# *Discussion*

Thank you!

http://www.dice.ucl.ac.be/crypto/