

PROYECTO DE ENCRYPTACIÓN Y DESENCRIPTACIÓN EN HARDWARE

Testbenches de Encriptación/Desencriptación

TE2002B.401

ENCRYPTADORES

Líder - Angelo Segura Ibarra

AddRoundKey

Emiliano Camacho Ponce
Obiel Emiliano Rangel Moreno
Evan André Santana Pacheco
Alfredo Alejandro Soto Herrera

MixColumns

Yumee Chung
Ana Carolina Coronel Herrera
Adrián Márquez Núñez
Andrés Ignacio Zegales Taborga

ShiftRows

Dalila Fonseca Maya
Santiago Isai González Arista
Victoria Lilian Robles Vargas
Angelo Segura Ibarra

Líder - Andrés Ignacio Zegales Taborga

SubBytes

Ricardo Antonio Enguilo Sánchez
Mariana Hernández Díaz
Lizeth Niniz Corona

Key Schedule

Anell Moreno Quilatán
Mario Armando Moscoso
Fernando Hernández
Hernández

DESENCRIPTADORES

Líder- *Omar Reyes Barrueta*

Líder- *Arturo Balboa Alvarado*

Inv SubBytes

Matías Piedra Pichardo
Josemaría De Alba Arguelles
Ricardo Ruiz Cano
Samantha Erin Medina Muñoz
Fernando García Padilla

Inv ShiftRows

Socorro Alejandra Montelongo
Loredo
José Fernando Vivanco Benavides
Juan César Nuño Rivera
Gabriel Enrique Lascurain Flores
Arturo Urías Jiménez

Inv AddRoundKey

Arturo Balboa Alvarado
Anell Moreno Quilantán
Omar Reyes Barrueta
Mario Armando Moscoso
Becerra
Fernando Hernández Hernández

Inv MixColumns

Arturo Balboa Alvarado
Omar Ryes Barrueta

Key Schedule

Anell Moreno Quilatán
Mario Armando Moscoso
Becerra
Fernando Hernández Hernández

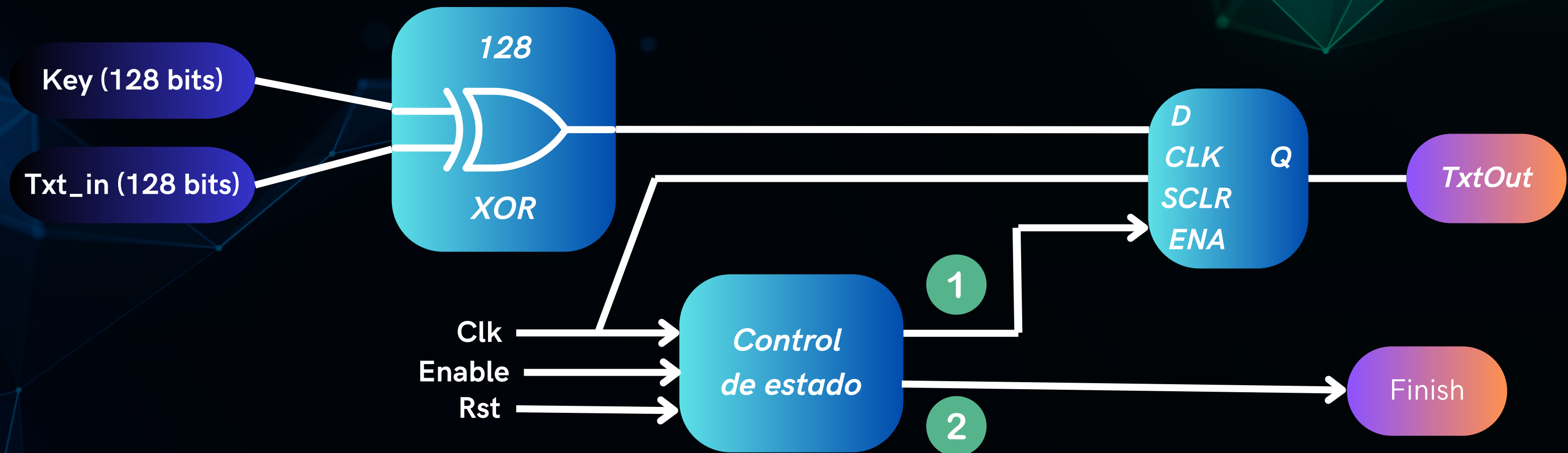
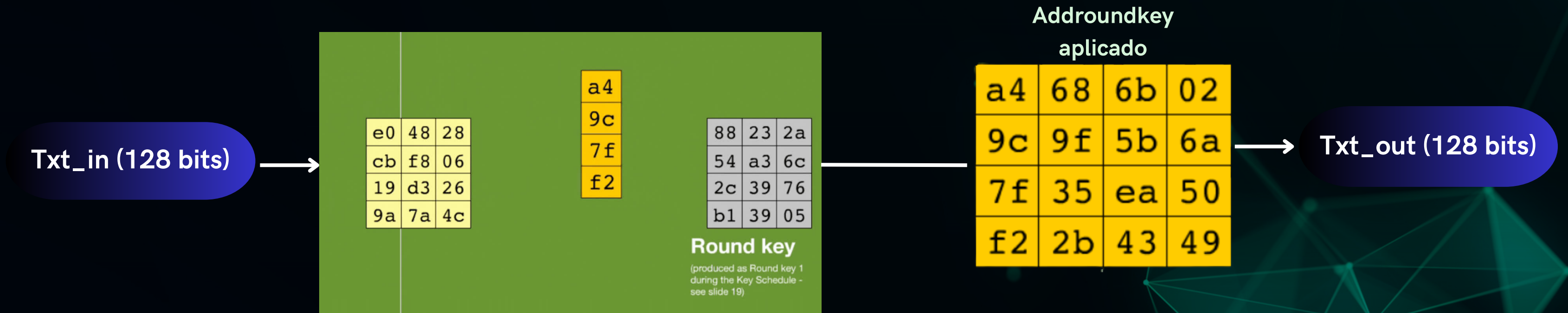
Para garantizar el correcto funcionamiento de cada módulo del sistema de encriptación y desencriptación, desarrollamos un conjunto de **testbenches**.

Cada testbench es un entorno de prueba que simula el comportamiento real del módulo, permitiéndolo verificar que la lógica implementada sea correcta antes de llevarla al hardware.

ENCRYPTACIÓN



ADDROUNDKEY



SIMULACIÓN

Addroundkey

Prueba 1

Quartus Prime Lite Edition - C:/D/AddRoundKey/AddRoundKey - AddRoundKey

File Edit View Project Assignments Processing Tools Window Help

Search Intel FPGA

Project Navigator Files

- AddRoundKey.vhd
- tb_AddRoundKey.vhd

Tasks Compilation

- Task
- Compile Design
- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate program)
- Timing Analysis
- EDA Netlist Writer
- Edit Settings

```
121         report "Test Case 3 failed: Finish not cleared by reset"
122         severity error;
123
124         rst <= '0';
125         enable <= '0';
126         wait for CLK_PERIOD;
127
128         -- Test Case 4: All zeros input
129         report "Test Case 4: All zeros input";
130         key_in <= (others => '0');
131         txt_in <= (others => '0');
132         enable <= '1';
133         wait until rising_edge(clk);
134         wait for 1 ns;
135
136         -- Verificación de ceros
137         for i in 0 to 127 loop
138             assert txt_in(i) = '0'
139                 report "Test Case 4 failed: 0 XOR 0 should be 0"
140                 severity error;
141         end loop;
142
143         enable <= '0';
144         wait for CLK_PERIOD;
145
146         -- End of tests
147         report "All tests completed successfully";
148         wait;
149     end process;
150
151 end architecture test;
```

Messages

Type	ID	Message
Info	22036	For messages from NativeLink execution see the NativeLink log file C:/D/AddRoundKey/AddRoundKey_nativelink_simulation.rpt
Info	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/21.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "AddRoundKey" "AddRoundKey")
Info	22036	For messages from NativeLink execution see the NativeLink log file C:/D/AddRoundKey/AddRoundKey_nativelink_simulation.rpt
Info	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/21.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "AddRoundKey" "AddRoundKey")
Info	22036	For messages from NativeLink execution see the NativeLink log file C:/D/AddRoundKey/AddRoundKey_nativelink_simulation.rpt

System (11) Processing (122)

Runs the specified RTL simulation tool

100% 00:00:35

SIMULACIÓN

Addroundkey

Prueba 2

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the 'Compilation Report - AddRoundKey' for the 'AddRoundKey' project. The report indicates a successful compilation on Wednesday, April 23, 2025, at 11:40:05. The project is built using Quartus Prime Lite Edition 21.1.1. The report lists the following resources and their utilization:

Resource	Used	Available	Usage (%)
MAX 10	10M50DAF484C7G		
Final	131 / 49,760		(< 1 %)
129	260 / 360		(72 %)
0	0 / 1,677,312		(0 %)
Embedded Multiplier 9-bit elements	0 / 288		(0 %)
Total PLLs	0 / 4		(0 %)
UFM blocks	0 / 1		(0 %)
ADC blocks	0 / 2		(0 %)

The 'Messages' window at the bottom shows the following messages:

- Command: quartus_eda --read_settings_files=off --write_settings_files=off AddRoundKey -c AddRoundKey
- 18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value.
- 204019 Generated file AddRoundKey.vho in folder "C:/D/AddRoundKey/simulation/modelsim/" for EDA simulation tool
- Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 13 warnings

The 'Tasks' window on the left shows the compilation process with the following tasks completed successfully:

- Compile Design
- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate program)
- Timing Analysis
- EDA Netlist Writer

The status bar at the bottom indicates 100% zoom and a runtime of 00:00:37.

SUBBYTES

Txt_in (128 bits)



d4	a0	9a	e9
27	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	16	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-BOX byte substitution table



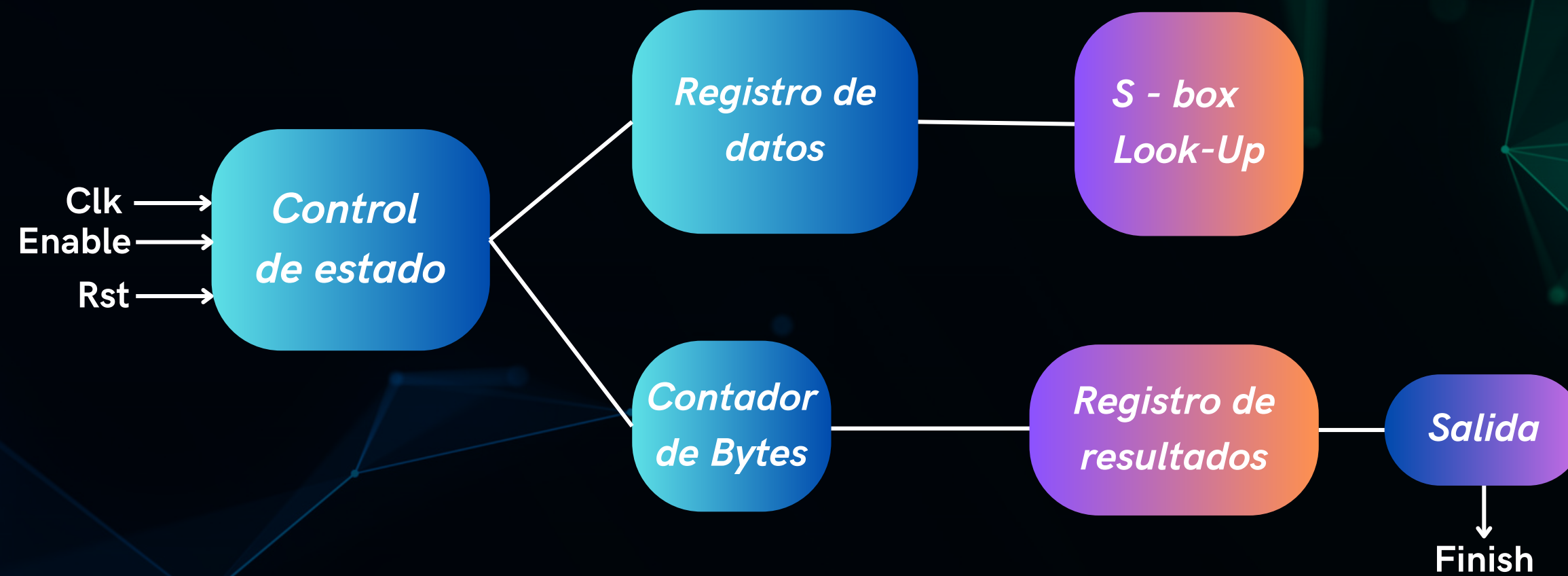
Txt_out (128 bits)

S-BOX

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

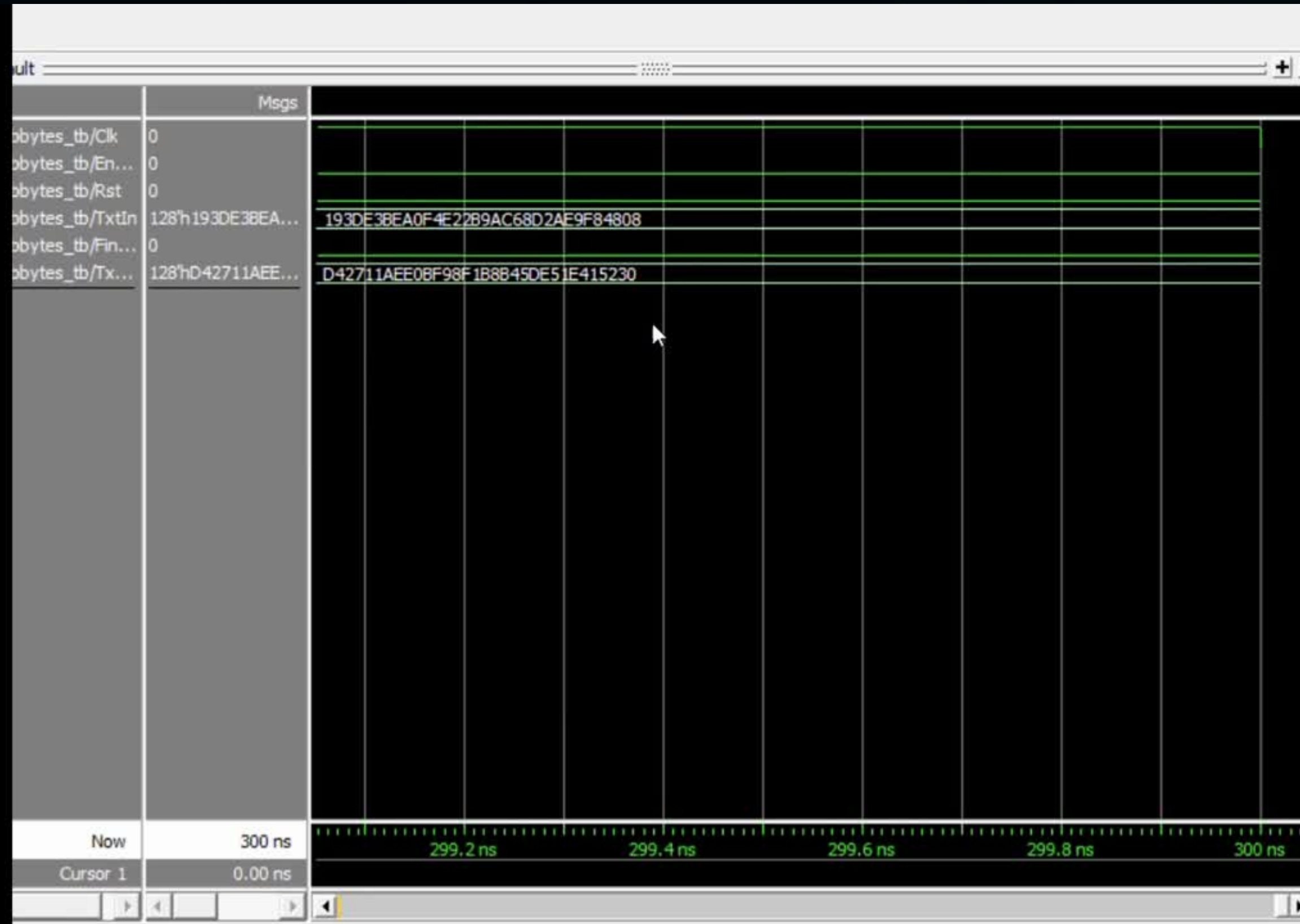
SUBBYTES

Diagrama de funcionamiento

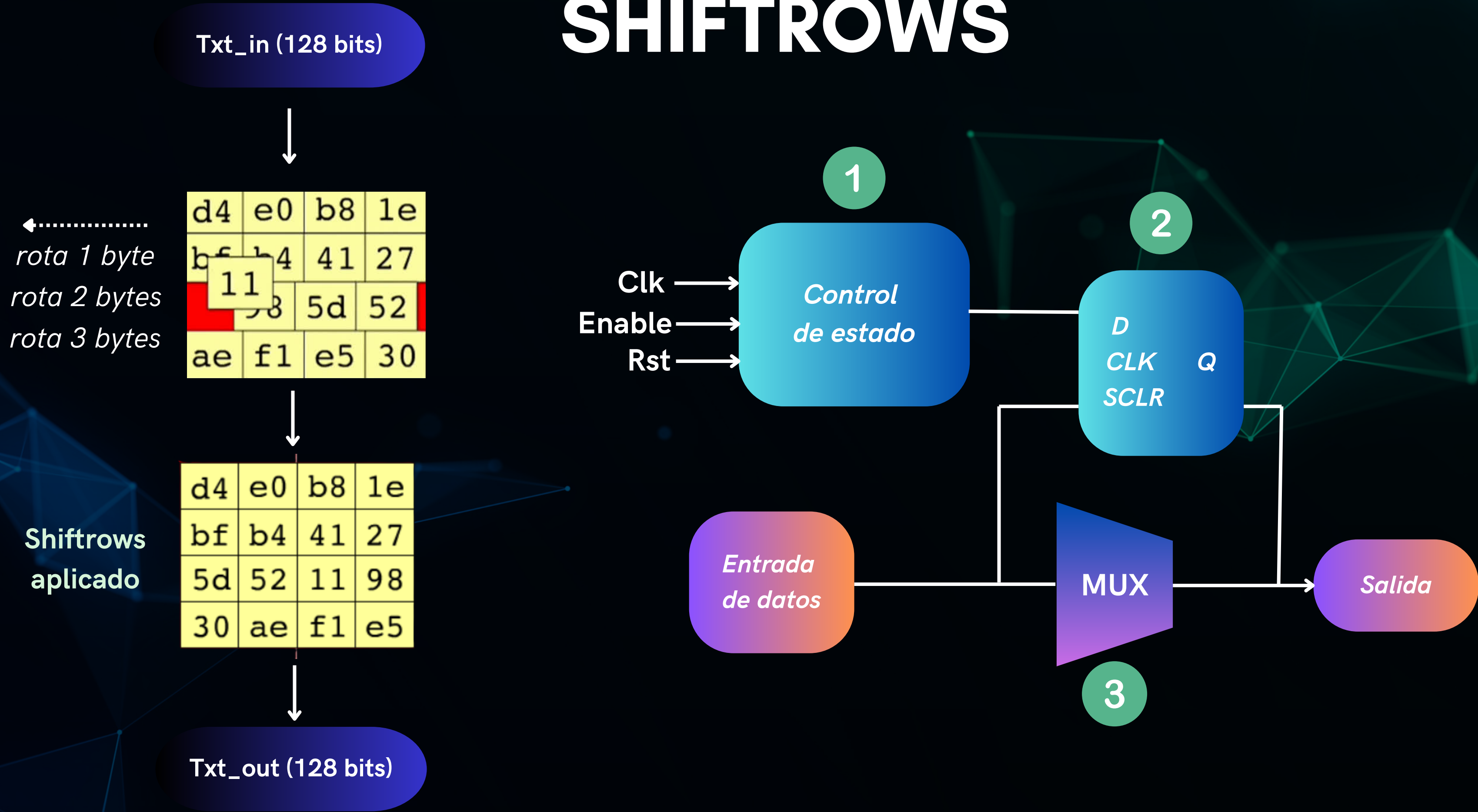


SIMULACIÓN

Subbytes



SHIFTROWS



SIMULACIÓN

Shiftrows

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the VHDL code for `ShiftRows_tb.vhd`. The code defines a testbench for the `ShiftRows` component. The `entity` and `architecture` sections are visible, along with component declarations, port definitions, and signal assignments.

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity ShiftRows_tb is
6 end ShiftRows_tb;
7
8 architecture behavior of ShiftRows_tb is
9
10     -- Component Declaration
11     component ShiftRows
12     port(
13         Clk      : in  std_logic;
14         Enable   : in  std_logic;
15         Finish   : out std_logic;
16         Rst      : in  std_logic;
17         TxtIn    : in  std_logic_vector(127 downto 0);
18         TxtOut   : out std_logic_vector(127 downto 0);
19     );
20 end component;
21
22 -- Inputs
23 signal Clk      : std_logic := '0';
24 signal Enable   : std_logic := '0';
25 signal Rst      : std_logic := '0';
26 signal TxtIn    : std_logic_vector(127 downto 0) := (others => '0');
27
28 -- Outputs
29 signal Finish : std_logic;
```

The left sidebar shows the Project Navigator with files `ShiftRows.vhd` and `ShiftRows_tb.vhd`. The Tasks pane lists compilation steps, with 'Compile Design' and its sub-tasks (Analysis & Synthesis, Fitter, Assembler, Timing Analysis, EDA Netlist Writer) marked as completed. The Status pane shows the progress of the compilation.

The bottom pane displays the Messages window, showing the following messages:

- 291001 Opening Licensing section of the Intel FPGA website at <https://www.intel.com/content/www/us/en/programmable/f/download/licensing/lic-q2web.html>
- 22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/21.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "ShiftRows" "ShiftRows")
- 22036 For messages from NativeLink execution see the NativeLink log file C:/ProyQuartus/Prueba_ShiftRows/ShiftRows_nativelink_simulation.rpt

The bottom status bar indicates the simulation is in the 'Processing' stage, with 0% completion and a time of 00:00:00.

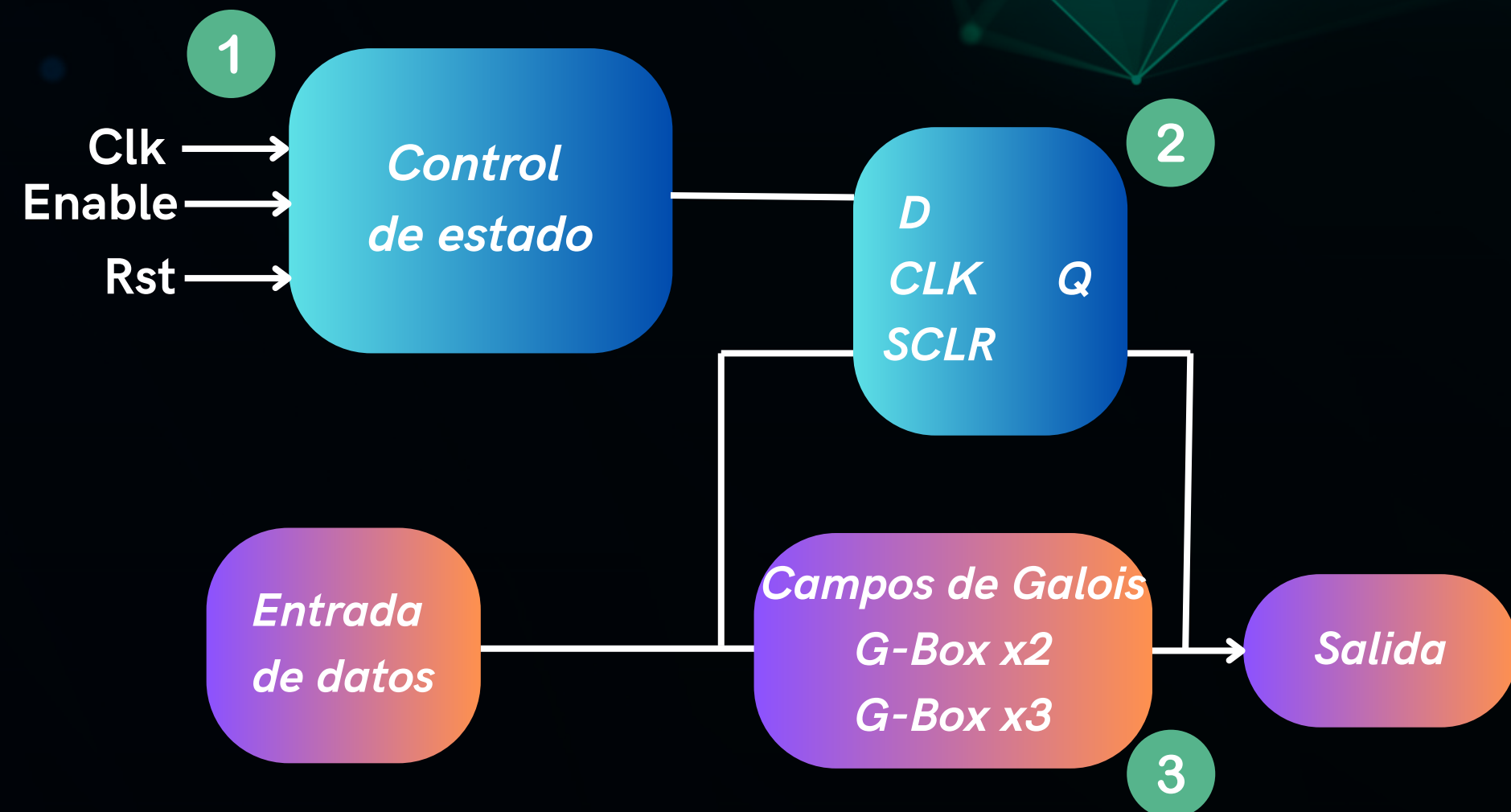
MIXCOLUMN

Txt_in (128 bits)

04	e0	b8	1e
66	b4	41	27
81	52	11	98
e5	ae	f1	e5

The MixColumns step along with the ShiftRows step is the primary source of diffusion in Rijndael.

Txt_out (128 bits)



SIMULACIÓN

Mix Column

The screenshot displays the Quartus Prime Lite Edition interface for a project named 'MixColumns'. The 'Tools' menu is open, showing options like 'Run Simulation Tool' and 'Launch Simulation Library Compiler'. The 'Project Navigator' on the left shows files 'MixColumns.vhd' and 'MixColumns_tb.vhd'. The 'Status' window at the bottom left shows the progress of various compilation steps, all of which are 100% complete.

Module	%	Progress
Full Compilation	100%	00:01
Analysis & Synthesis	100%	00:00
Fitter	100%	00:00
Assembler	100%	00:00
Timing Analyzer	100%	00:00
EDA Netlist Writer	100%	00:00

The main editor window shows the VHDL code for the 'MixColumns' module. The code includes library declarations for IEEE standard logic packages and a port declaration for 'clk', 'enable', 'finish', 'rst', and 'txtin'. The 'txtin' port is connected to a 'std_logic_vector(127 downto 0)'.

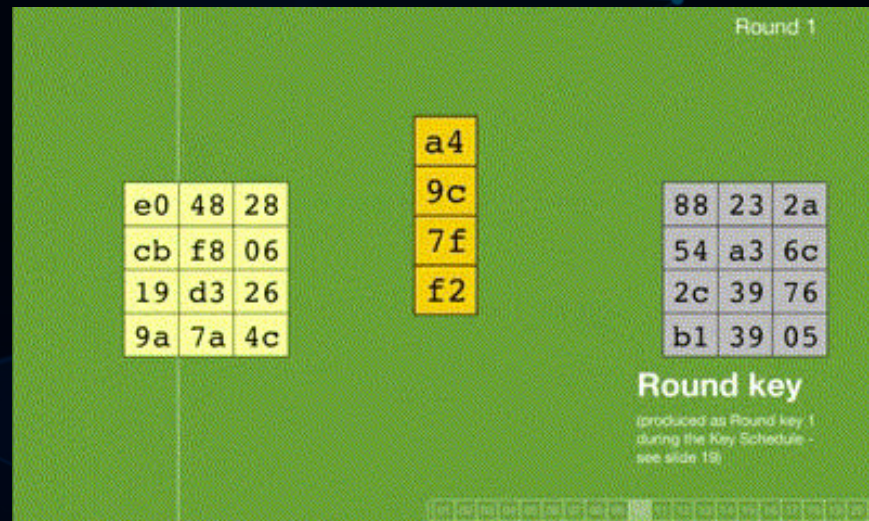
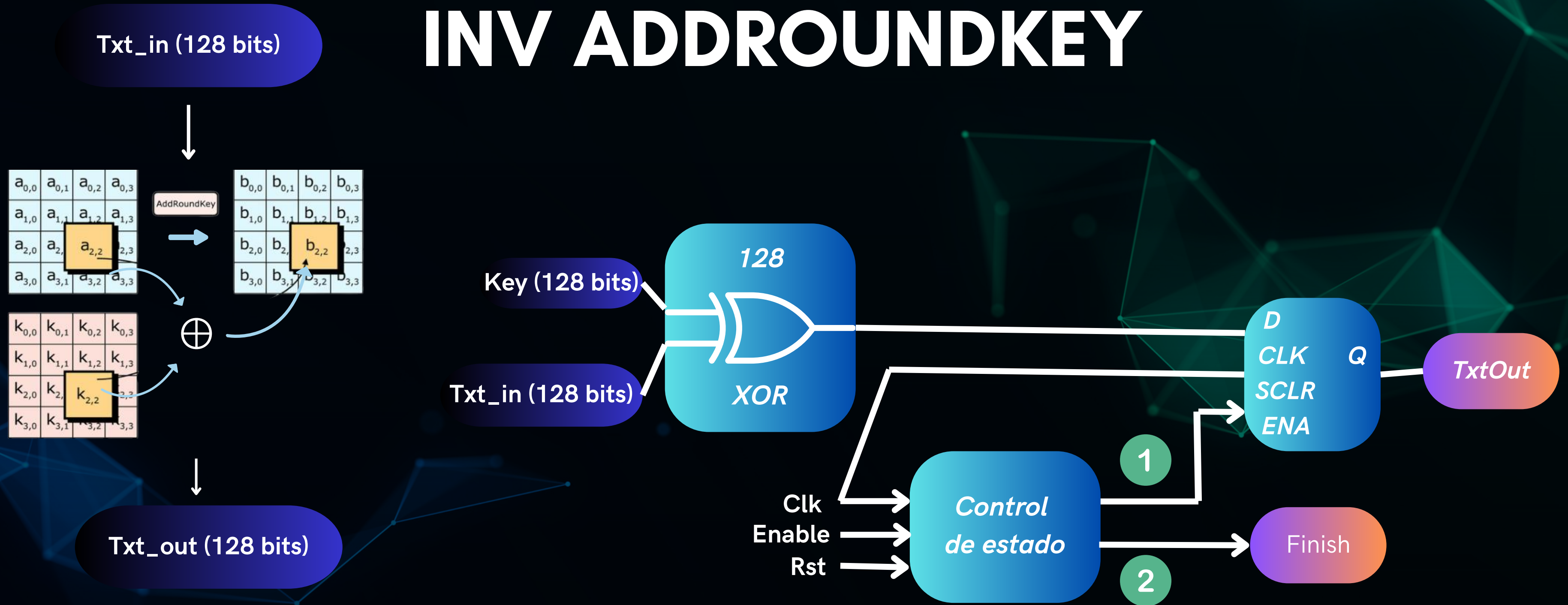
```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_unsigned.all;
4 use ieee.numeric_std.all;
5
6 entity MixColumns is
7     port (
8         clk : in std_logic;
9         enable : in std_logic;
10        finish : out std_logic;
11        rst : in std_logic;
12        txtin : in std_logic_vector(127 downto 0);
13    );
14 end entity MixColumns;
```

The 'Messages' window at the bottom shows a series of successful simulation messages, indicating that the NativeLink simulation was launched and executed successfully. The messages include the command 'quartus_sh -t "c:/intelfpga_lite/22.1std/quartus/common/tcl/internal/nativelink/qnativesim.tcl"' and the log file path 'C:/ProyQuartus/P15_MixColumns/MixColumns_nativelink_simulation.rpt'.

DESENCRIPTACIÓN

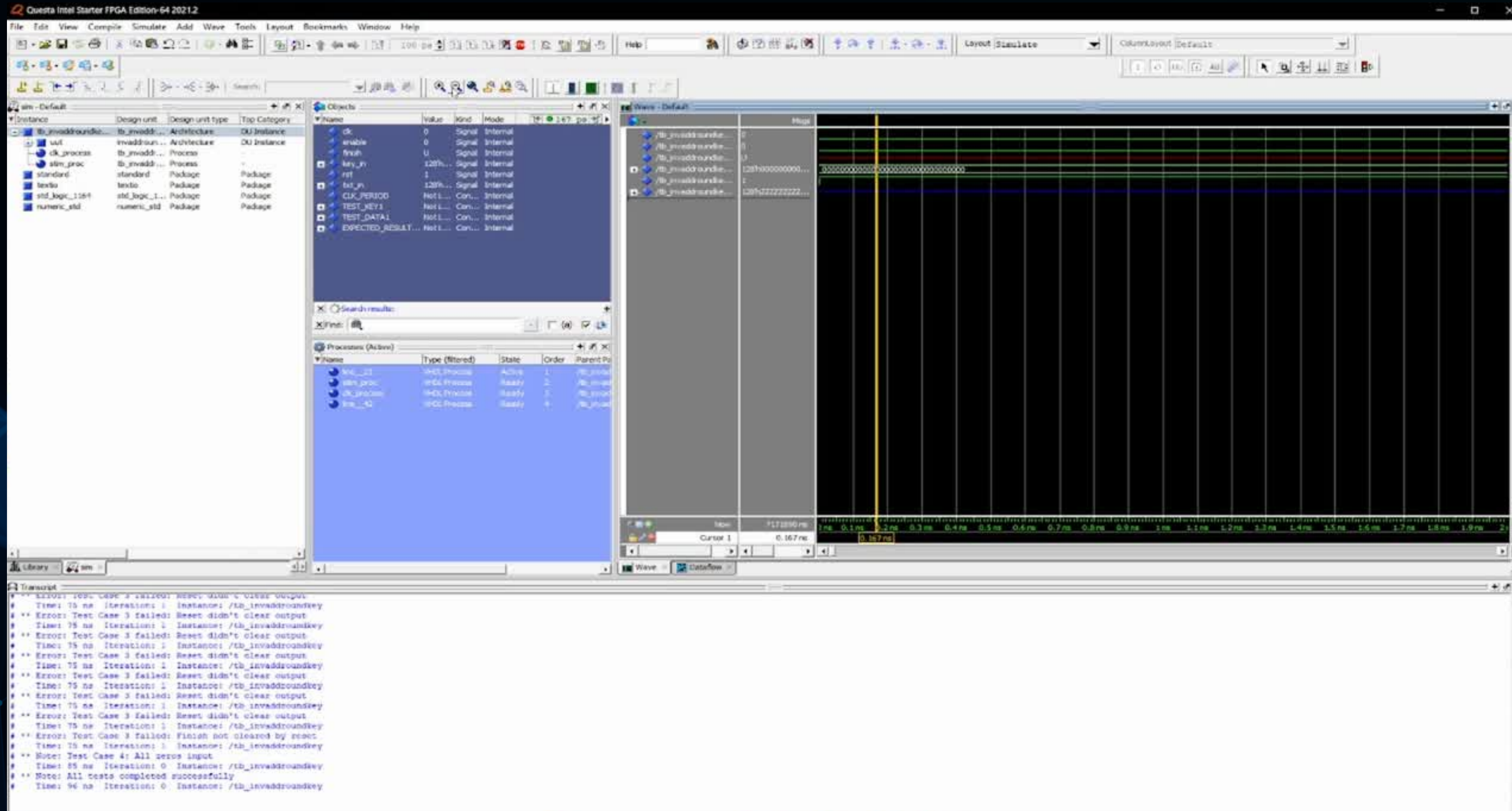


INV ADDROUNDKEY

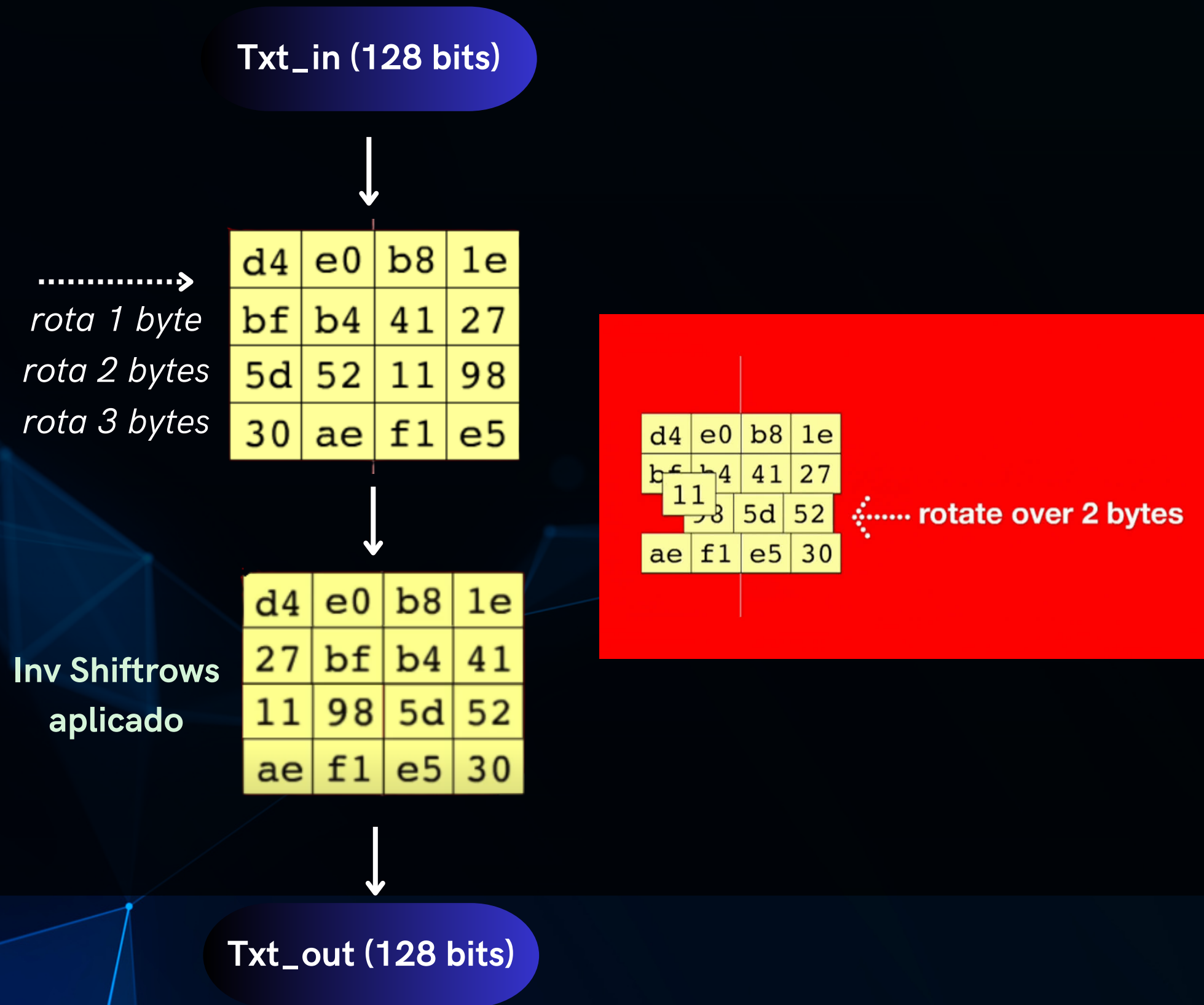


SIMULACIÓN

Inv Addroundkey

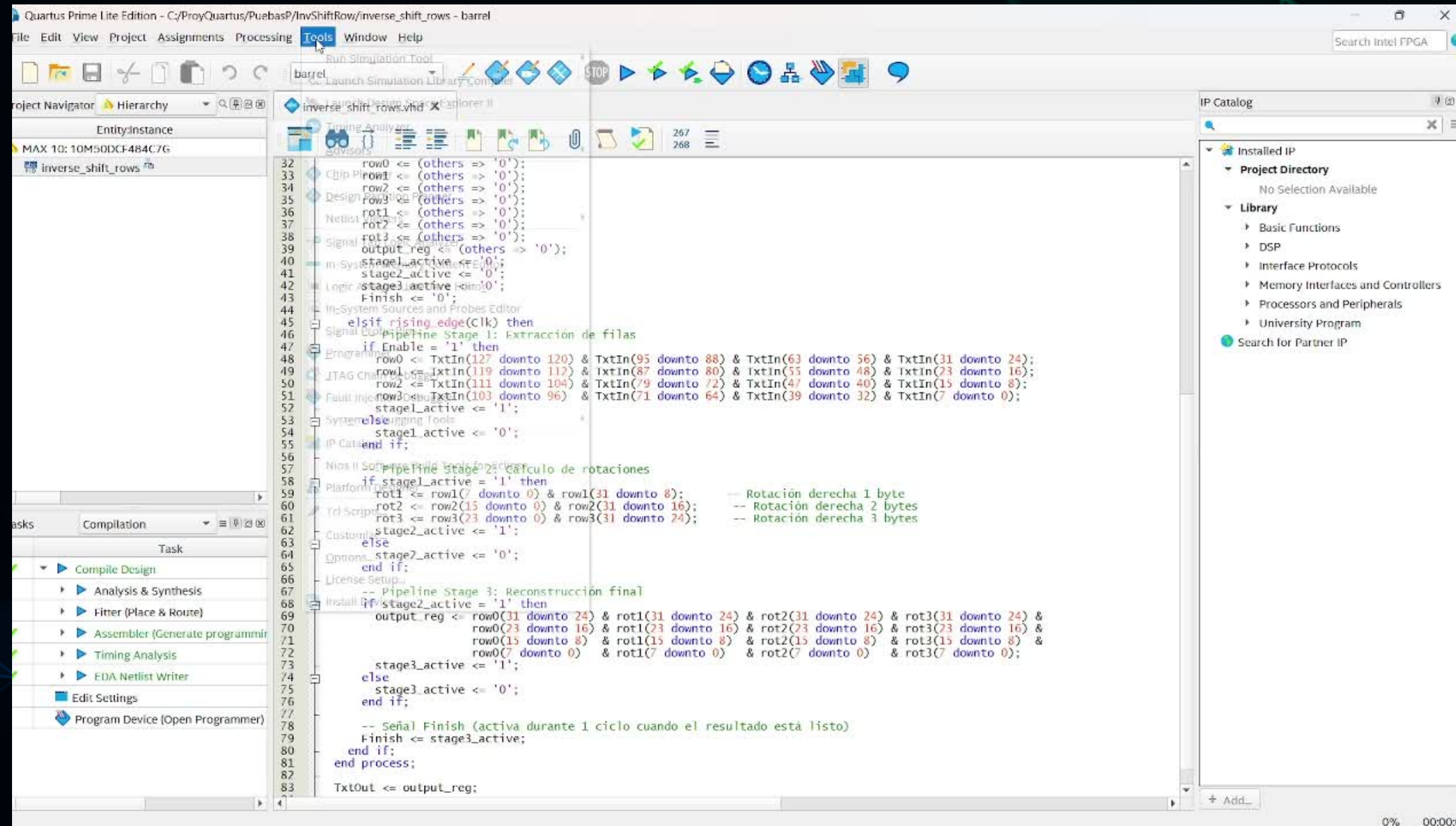


INV SHIFTRROWS



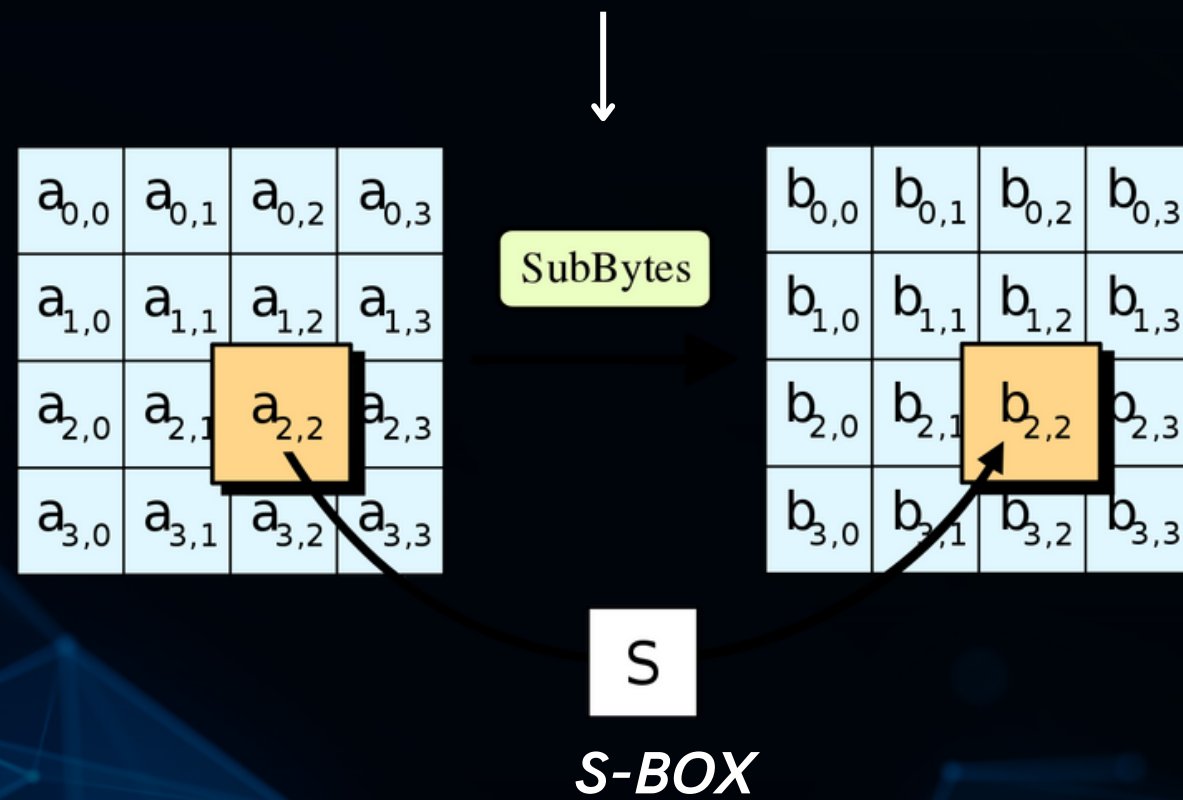
SIMULACIÓN

Inv Shiftrows



INV SUBBYTES

Txt_in (128 bits)



S-SBOX

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Txt_out (128 bits)

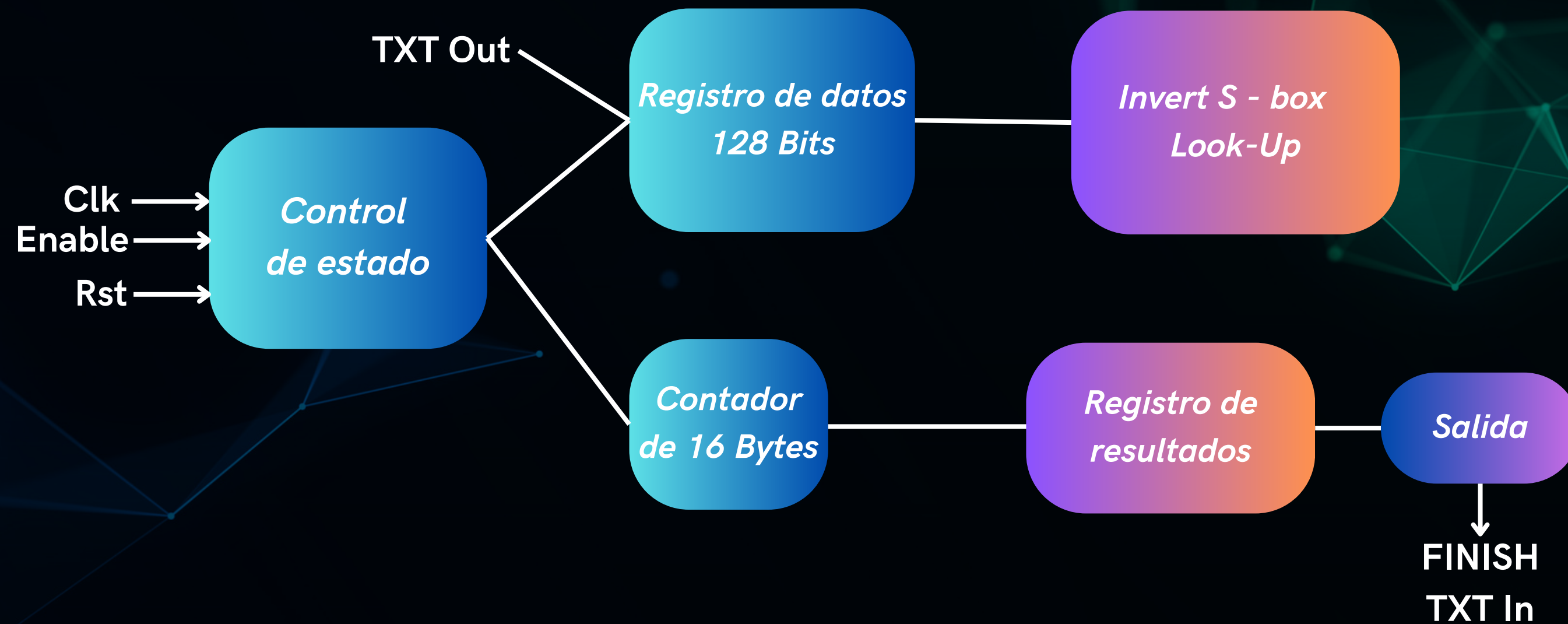
d4	a0	9a	e9
27	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

Row	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-SBOX byte substitution table

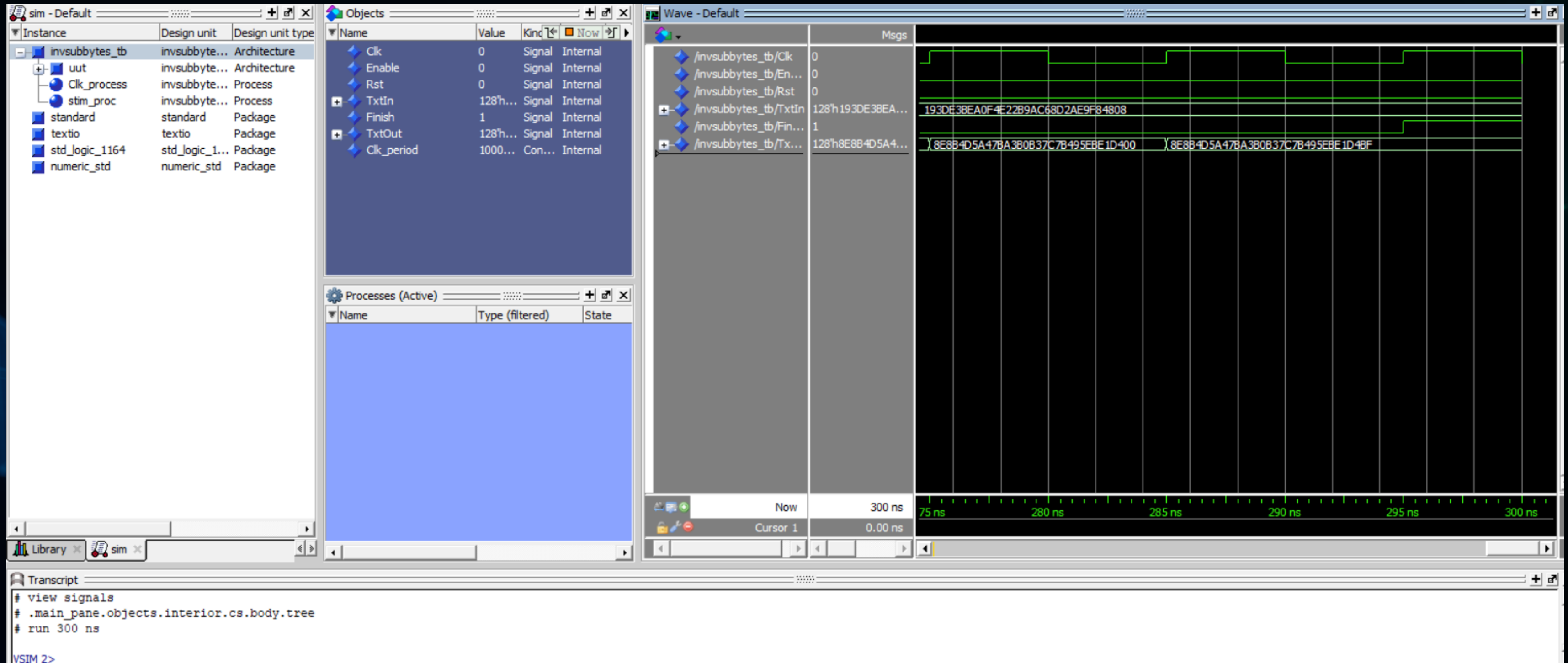
DIAGRAMA

Inv Subbytes



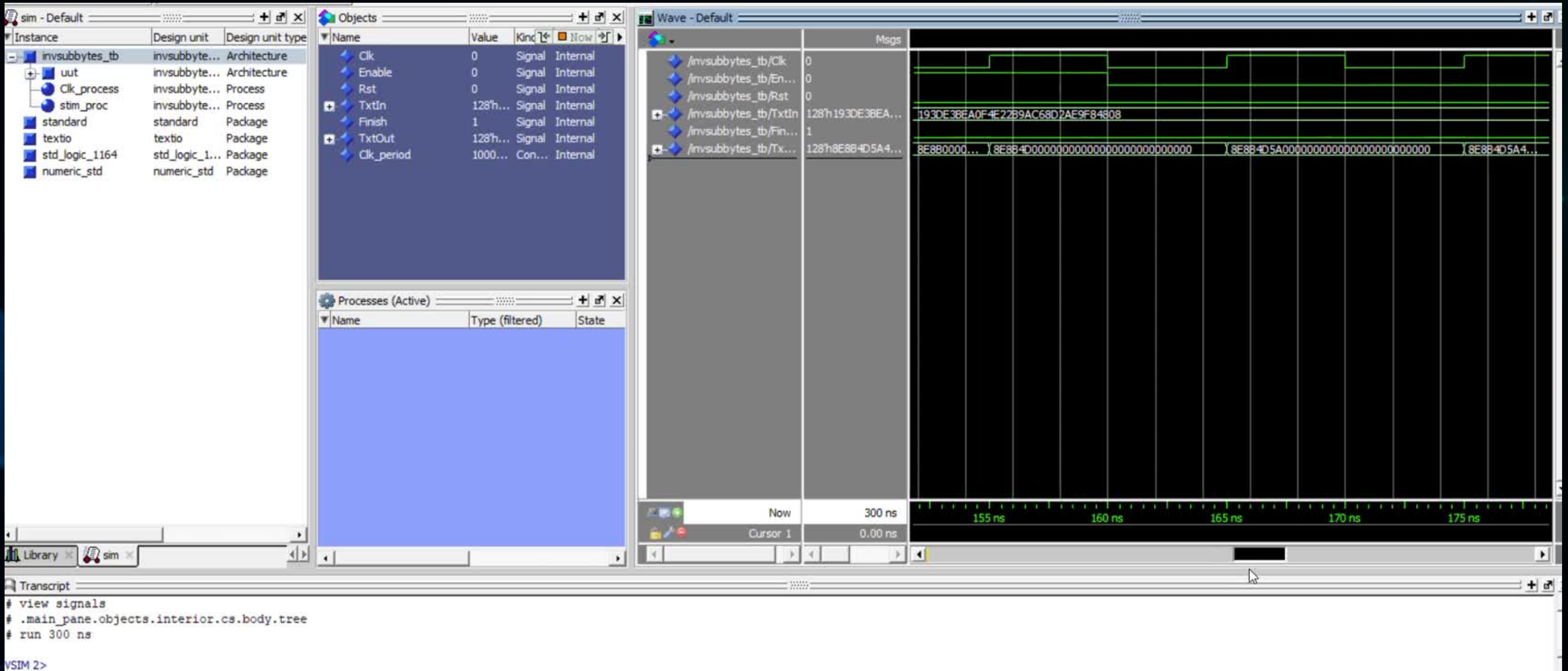
SIMULACIÓN

Inv Subbytes



SIMULACIÓN

Inv Subbytes



INV MIX COLUMN

Txt_in (128 bits)



04	e0	b8	1e
49	d4	81	92
31	36	ae	88
e5	a0	11	e5

 $\times$

14	11	13
9	14	13
13	14	11
11	13	14

 $=$

d4	66	e0	ea
bf	e8	84	04
5d	52	11	34
30	bf	81	7d

The fixed matrix
used in Rijndael's
Galois Field

Txt_out (128 bits)



046681E5E0CB199A48F8D37A2806264C				00000000
00000...		D48F5D30E0B452AEB84111F11E2798E5		

SIMULACIÓN

Inv Mix Column

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the VHDL code for the `InvMixColumns` entity. The code includes a port declaration for `Clk`, `Enable`, `Finish`, `Rst`, `DataIn`, and `DataOut`. The architecture is behavioral and implements a function `timesTwo` that multiplies a 7-bit vector by 2. The function uses a loop to calculate the result, with a comment indicating it multiplies by 2. The simulation results are shown in the Messages window at the bottom, indicating a successful launch of the NativeLink simulation.

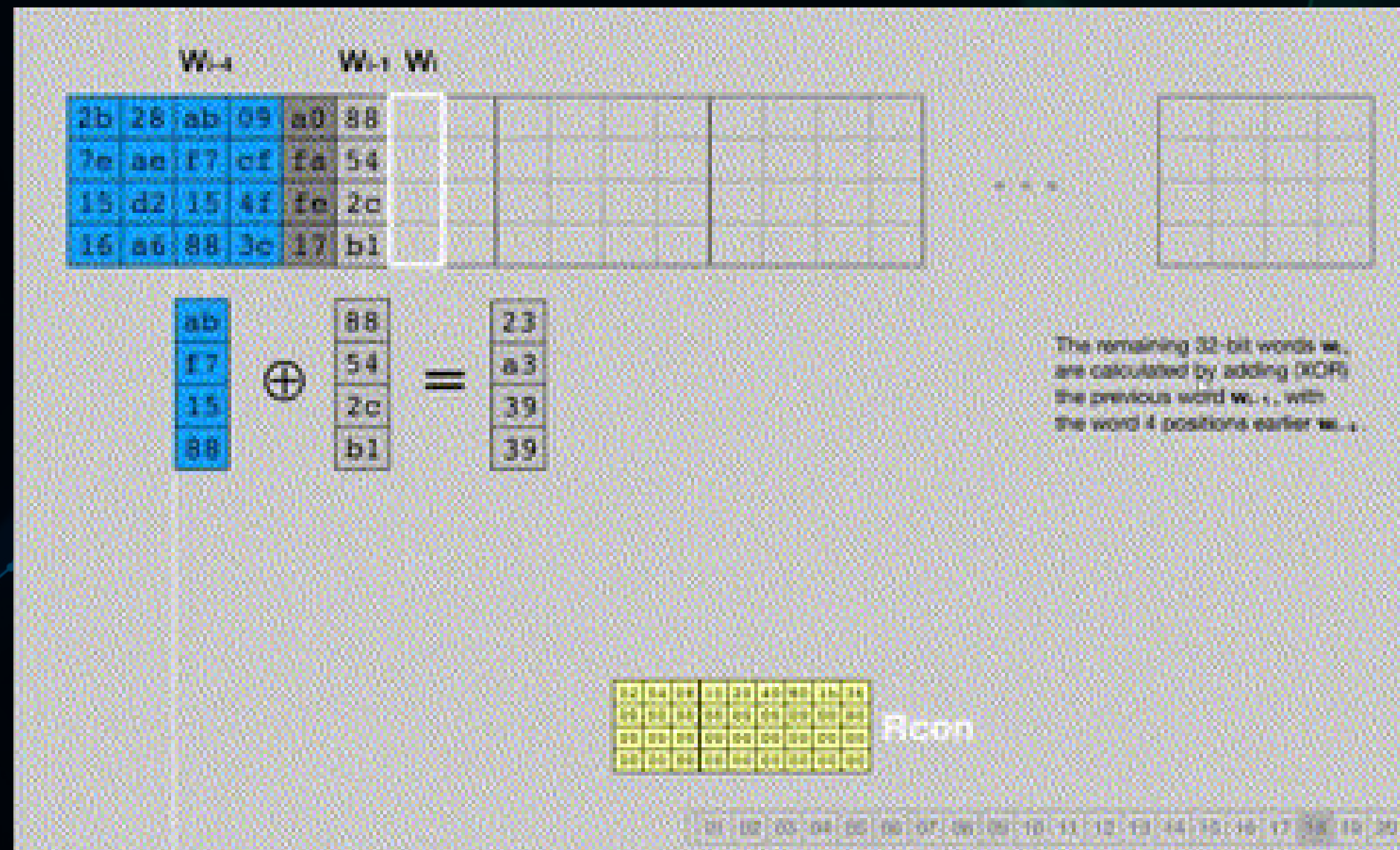
```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity InvMixColumns is
6 port (
7     Clk      : in  std_logic;
8     Enable   : in  std_logic;
9     Finish   : out std_logic;
10    Rst       : in  std_logic;
11    DataIn    : in  std_logic_vector(127 downto 0);
12    DataOut   : out std_logic_vector(127 downto 0);
13 );
14 end entity InvMixColumns;
15
16 architecture Behavioral of InvMixColumns is
17
18     --Multiplica a por 2 n veces
19     function timesTwo (a : std_logic_vector(7 downto 0); b : integer) return std_logic_vector is
20         variable result : std_logic_vector(7 downto 0);
21     begin
22         result := a;
23
24         for i in 0 to b -1 loop
25             if result(7) = '1' then
26                 result := (result(6 downto 0) & '0') xor "00011011";
27             else
28                 result := result(6 downto 0) & '0';
29             end if;
30         end loop;
31
32         return result;
33     end function;
34
35 end architecture Behavioral of InvMixColumns;
```

Messages

Type	ID	Message
Information	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/21.1/quartus/common/tcl/internal/nativeLink/qnativesim.tcl" --rtl_sim "InvMixColumns")
Information	22036	For messages from NativeLink execution see the NativeLink log file C:/Users/arbaj/Downloads/InvMixColumns/InvMixColumns/InvMixColumns_nativeLink_simulation.rpt

System (2) Processing (122)

INV KEY SCHEDULE



SIMULACIÓN

Inv Key Schedule

Objects

Name	Value	Unit	Kind
clk	Not L...	Signal	Internal
reset	Not L...	Signal	Internal
start	0	Signal	Internal
key_in	128'h...	Signal	Internal
done	0	Signal	Internal
clk_period	Not L...	Con...	Internal

Wave - Default

Msgs
/key_schedule_inve...
/key_schedule_inve...
/key_schedule_inve...



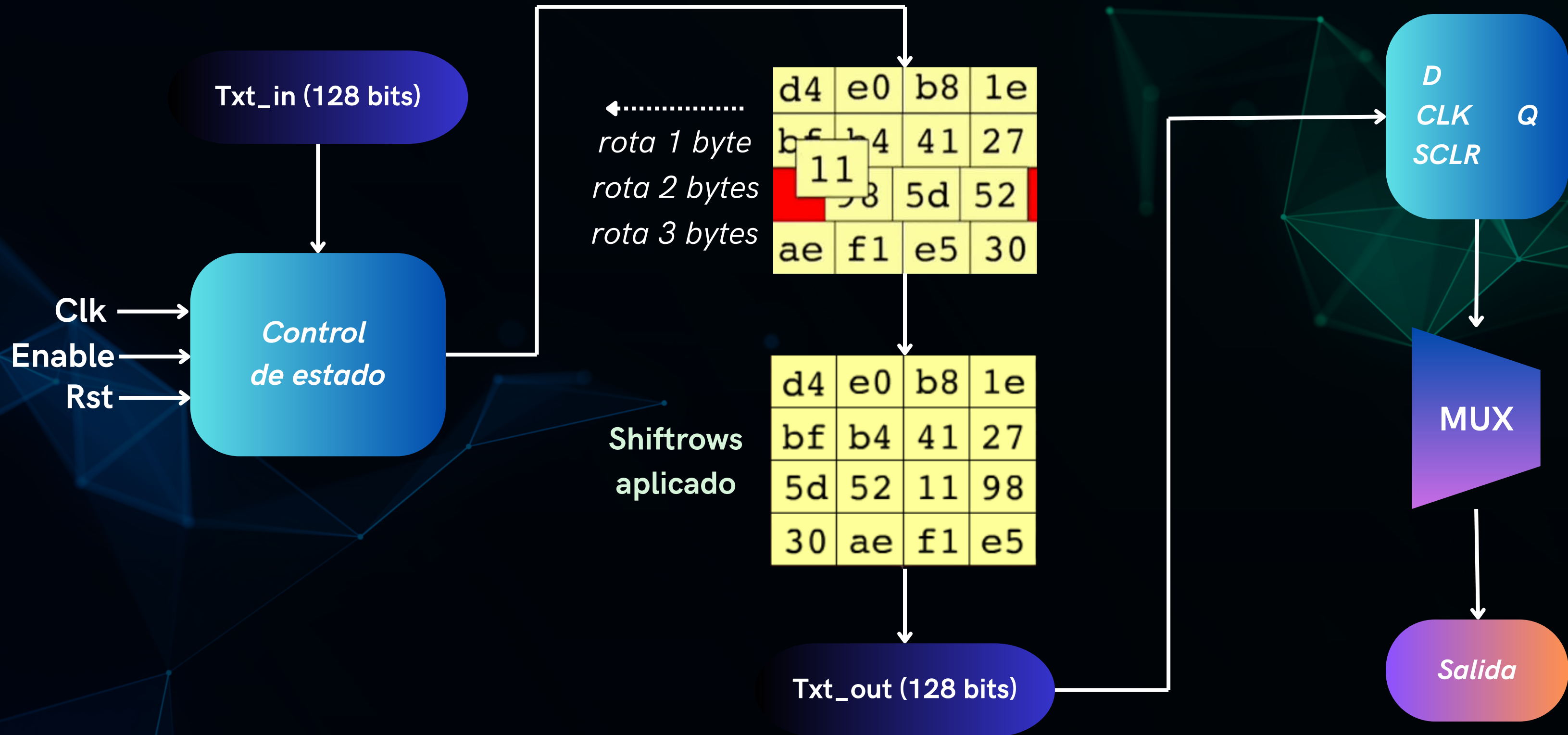
**MUCHAS
GRACIAS!**

BIBLIOGRAFIA

EITCA. (2023) ¿Cuál es el papel del polinomio irreducible en la operación de multiplicación en los campos de Galois?
<https://eitca.org/cybersecurity/eitc-is-ccf-classical-cryptography-fundamentals/aes-block-cipher-cryptosystem/introduction-to-galois-fields-for-the-aes/examination-review-introduction-to-galois-fields-for-the-aes/what-is-the-role-of-the-irreducible-polynomial-in-the-multiplication-operation-in-galois-fields/>

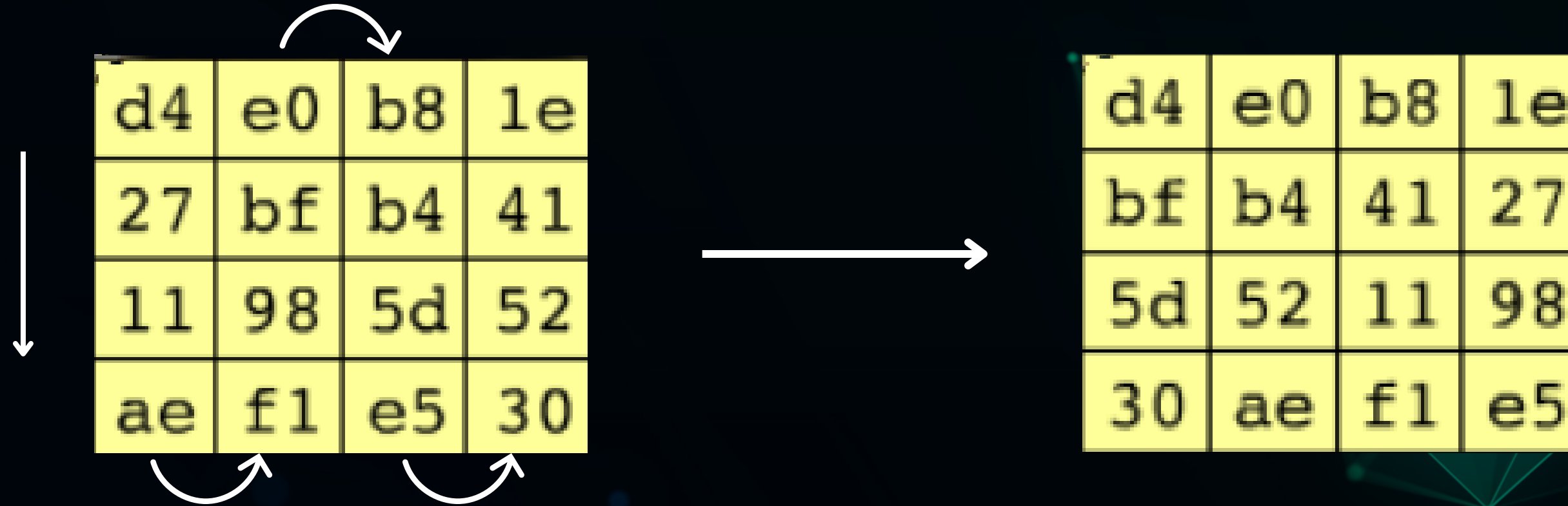
EITCA. (2023). ¿Cómo se realiza la multiplicación en Galois Fields en el contexto del algoritmo AES?
<https://es.eitca.org/cybersecurity/eitc-is-ccf-classical-cryptography-fundamentals/aes-block-cipher-cryptosystem/introduction-to-galois-fields-for-the-aes/examination-review-introduction-to-galois-fields-for-the-aes/how-is-multiplication-performed-in-galois-fields-in-the-context-of-the-aes-algorithm/>

SHIFROWS



SIMULACIÓN

Shiftrows

[illegible]