Implementación de AES en Hardware

Cuauhtemoc Mancillas López

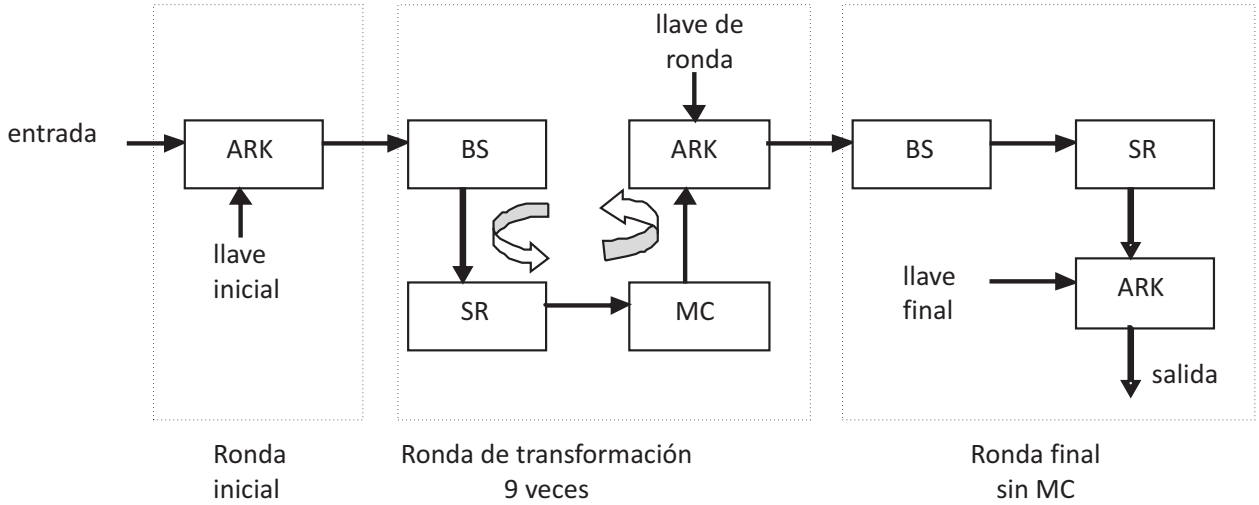
10 de febrero de 2012

Cifrador por Bloques AES

El 2 de octubre de 2000, el Instituto Nacional de Estándares y Tecnología (NIST por sus siglas en ingles) de Estados Unidos anunció el algoritmo ganador para sustituir al DES(*Data Encryption Standard*) y ser declarado como estándar de Cifrado Avanzado (AES por sus siglas en inglés). Rijndael propuesto por los belgas Vincent Rijmen y Joan Daemen, es un cifrador por bloques que opera con llaves de longitudes variables, que pueden ser especificadas independientemente a 128, 192, ó 256 bits.

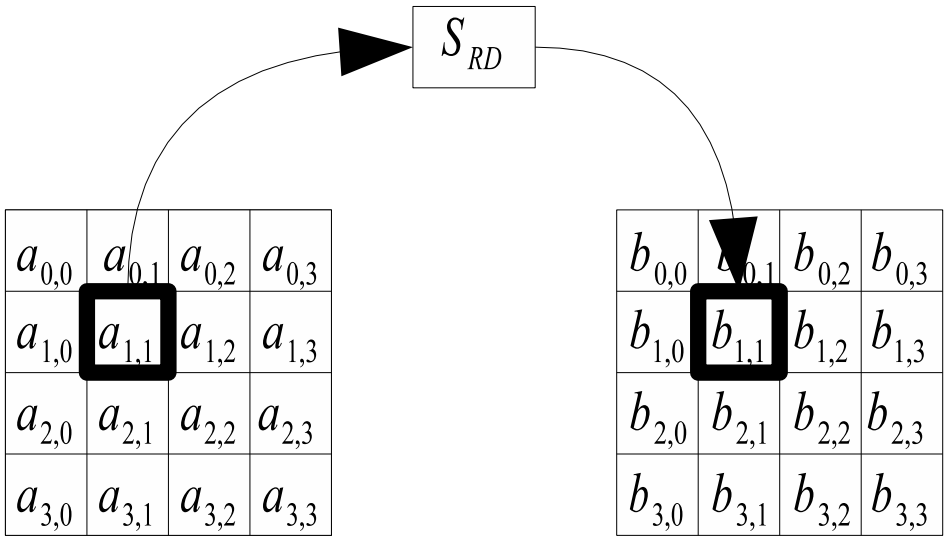
Cifrador por Bloques AES

El número de rondas definidas en el estándar es de 10 para llaves de 128 bits. La ronda de transformación se compone de 4 pasos: *SubBytes* (SB), *ShiftRows* (SR), *MixColumns* (MC)y *AddRoundKey* (ARK), la ronda final no incluye la aplicación de *MixColumns*.



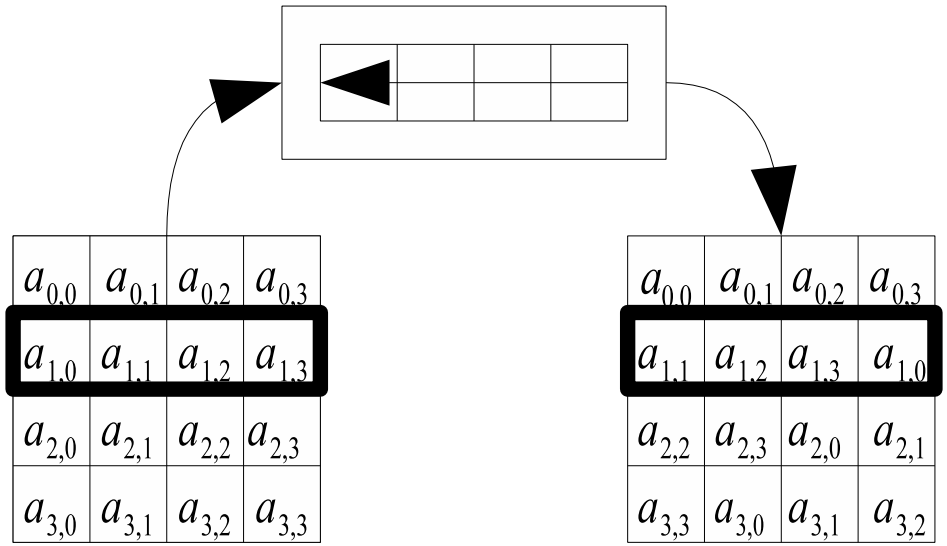
SubBytes (SB)

Consiste en la substitución de cada uno de los bytes de la matríz de estado por su correspondiente valor en la SBOX.



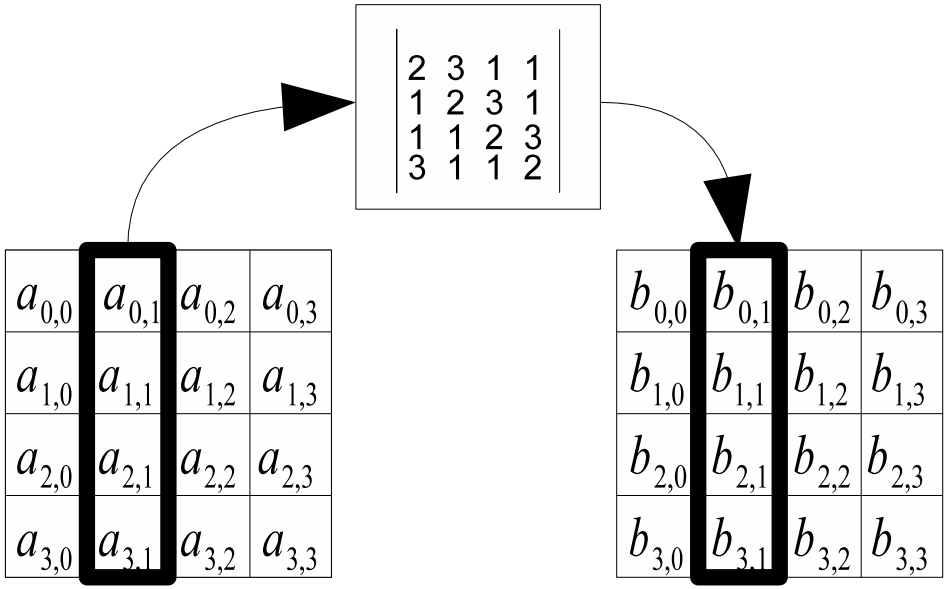
ShiftRows (SR)

Consiste en el corrimiento cíclico de las filas de la matriz de estado, en cero, uno, dos y tres posiciones respectivamente.



MixColumns (MC)

Multiplicación de la matríz de estado por una matriz constante.



Implementaciones de AES

- Microprocesadores de propósito general (Core i7, 8088).
- ARM (Cómputo móvil).
- Microcontroladores (PICs, Motorola).
- DSP.
- GPU.
- GAL.
- FPGA.
- ASIC.

Tipos de Cores

- Cifrado.
- Decifrado.
- Single-Chip (cifrado/decifrado).
- C/S generación de llaves.

Matríz de Estado

$$[P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}] \rightarrow$$

$$\begin{bmatrix} P_0 & P_4 & P_8 & P_{12} \\ P_1 & P_5 & P_9 & P_{13} \\ P_2 & P_6 & P_{10} & P_{14} \\ P_3 & P_7 & P_{11} & P_{15} \end{bmatrix}$$

Definición de la caja S

$$\begin{bmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Donde: $x = a^{-1}\ mod\ x^8 + x^4 + x^3 + x + 1$.

Definición de iBS

$$\begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Finalmente $a = x^{-1}\ mod\ x^8 + x^4 + x^3 + x + 1$.

Implementaciones de BS

- Guardar las cajas S para cifrado y decifrado en memoria.
- Guardar el inverso multiplicativo en $GF(2^8)$ en memoria.
- Realizar el cálculo completo de la caja S.

Arquitectura para BS y iBS

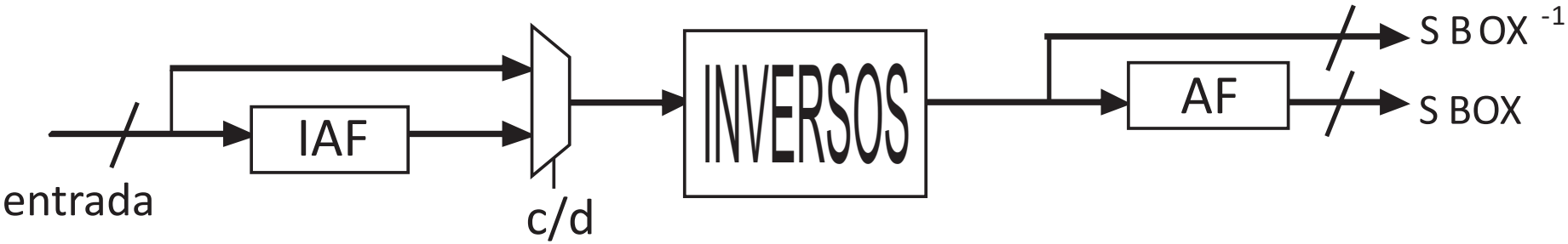


Figura: Arquitectura para la sustitución de bytes (BS) y su inversa (IBS).

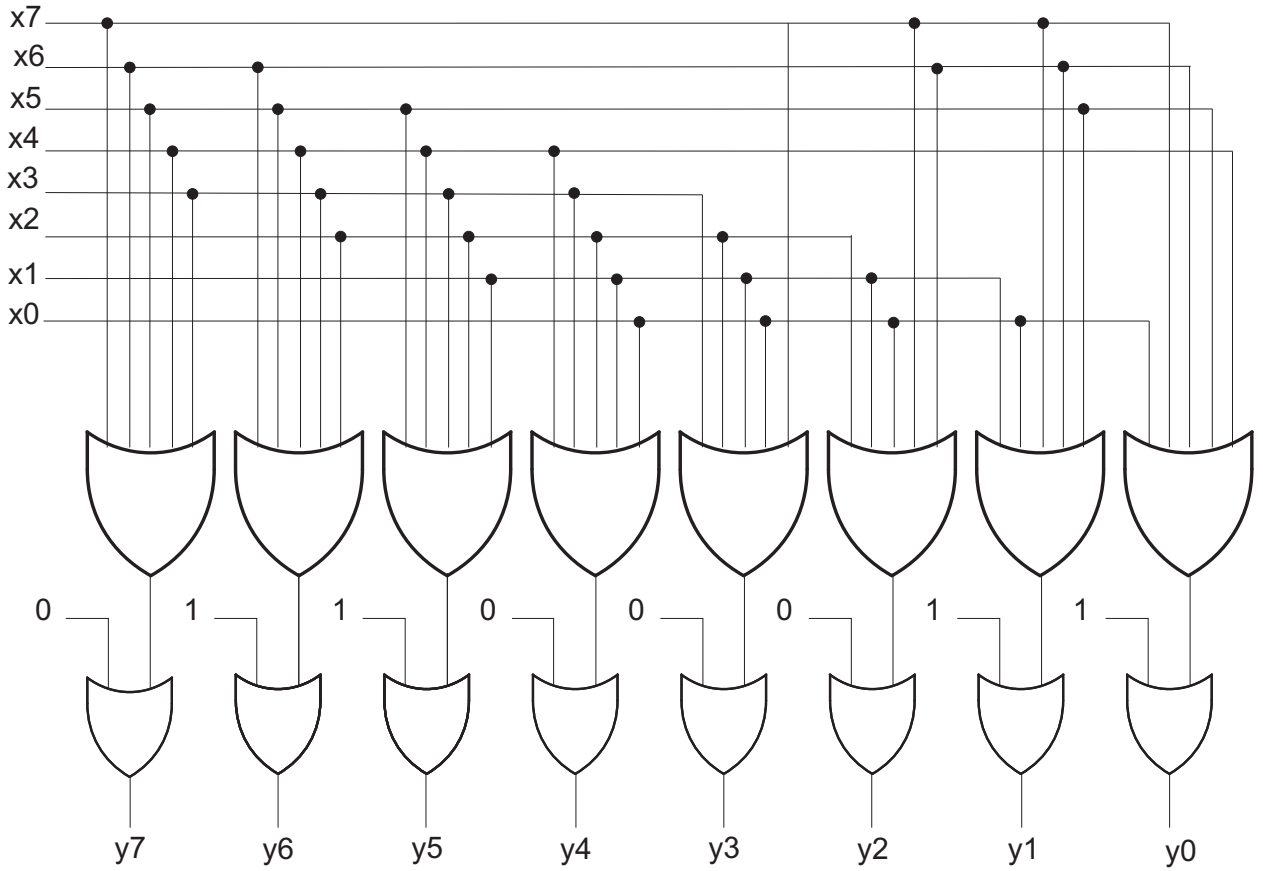


Figura: Circuito que calcula AF.

Implemetación de SR

Sólo representa recursos de ruteo.

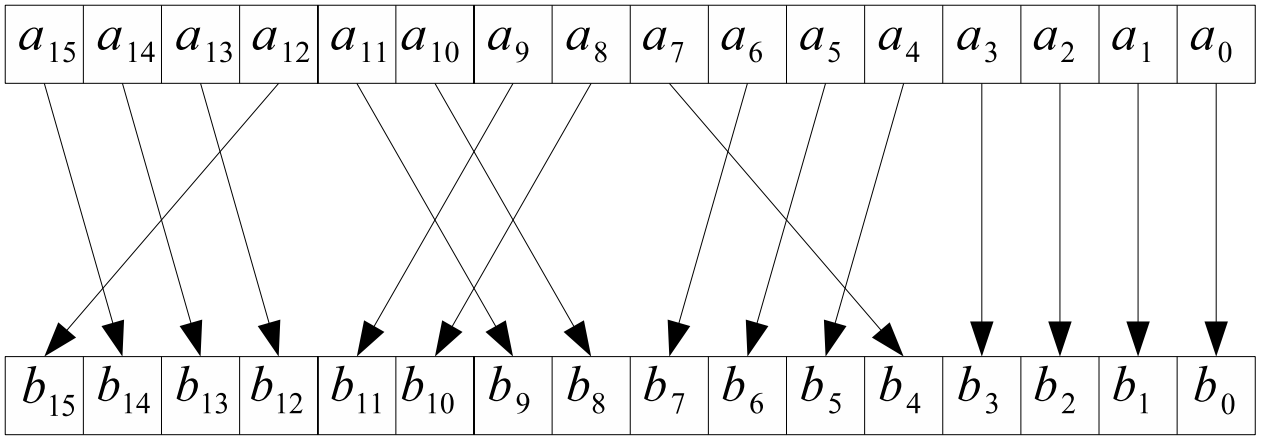


Figura: Arquitectura para el corrimiento de filas (SR).

Implemetación de SR

Sólo representa recursos de ruteo.

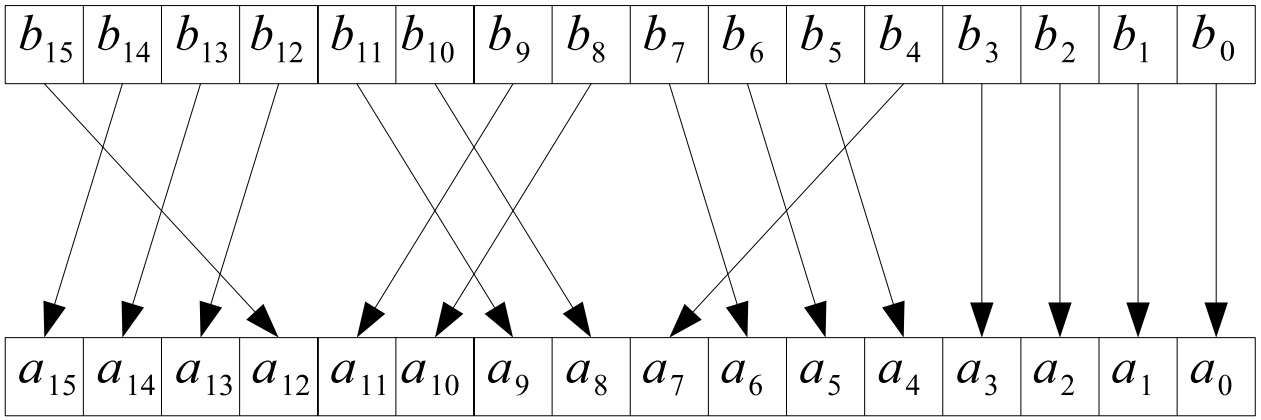


Figura: Arquitectura para el corrimiento de filas inverso (ISR).

Definición de MC

MC:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

iMC:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Reutilizando la matriz de MC para decifrar

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 01 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix}$$

Donde a la matriz,

$$\begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 01 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix}$$

se le llama *ModM*.

Arquitectura para MC e iMC

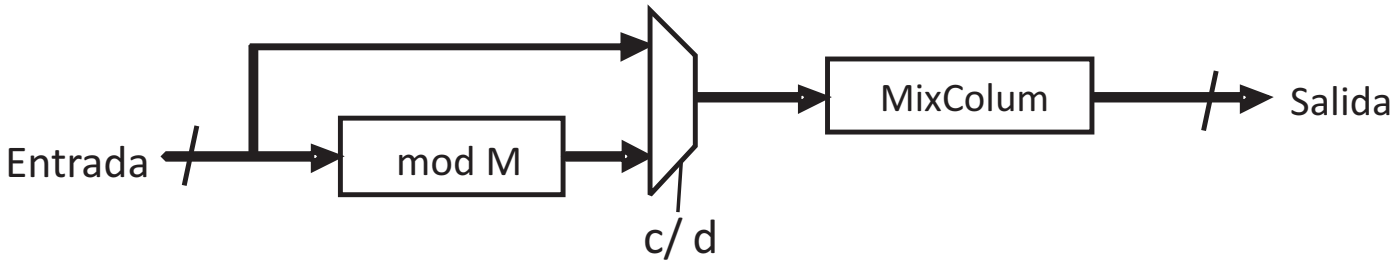


Figura: Arquitectura para el mezclado de columnas (MC) y su inversa (IMC).

Tablas T

Es una forma de optimizar AES para procesadores de 32 bits, consiste en mezclar SB y MC. Aumenta el monto requerido de memiria 4 veces. Reduce la ronda de AES a sólo $\oplus$'s.

Generación de llaves de ronda

La llave de cifrado original consiste en 128 bits acomodados como una matriz de bytes de 4×4. Sean $w[0], w[1], w[2]$ y $w[3]$ las cuatro columnas de la llave original. Entonces, estas cuatro columnas son expandidas de manera recursiva para obtener 40 columnas extras:

$$w[i] = \begin{cases} w[i-4] \oplus w[i-1] & \text{si i mod 4 } \neq 0 \\ w[i-4] \oplus T(w[i-1]) & \text{en otro caso} \end{cases}$$

Tipos de aquitecturas

- Secuencial.
- Desenrollada.
- Pipeline.
- Mezcas de las anteriores.

Ejemplos de arquitecturas

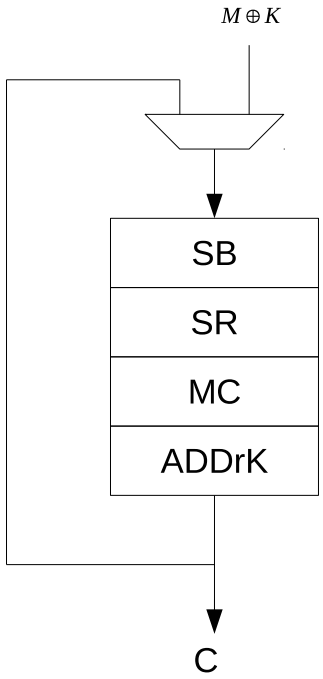
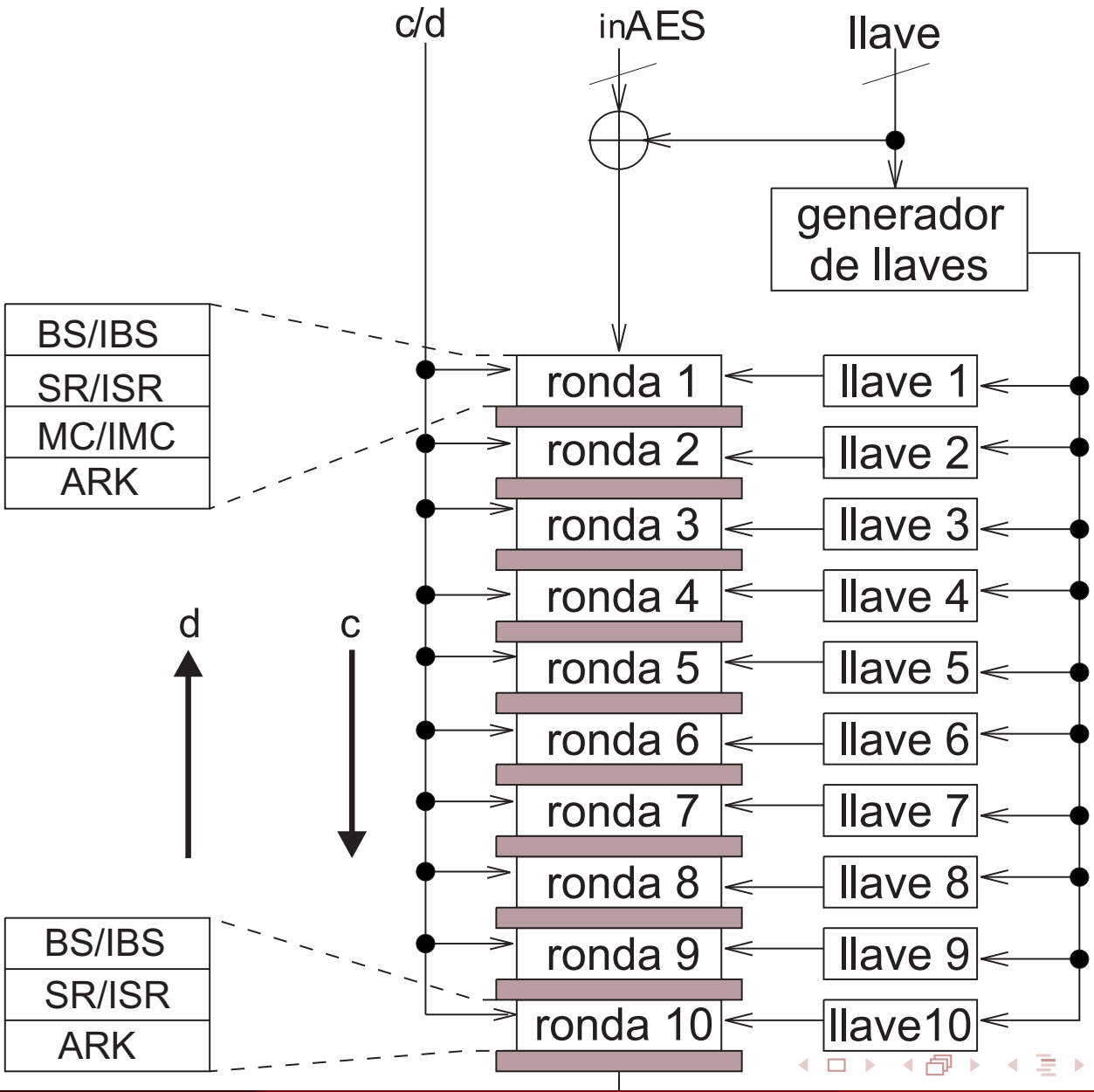


Figura: Arquitectura secuencial de AES.

Ejemplos de arquitecturas



Thanks for your attention.
Questions?