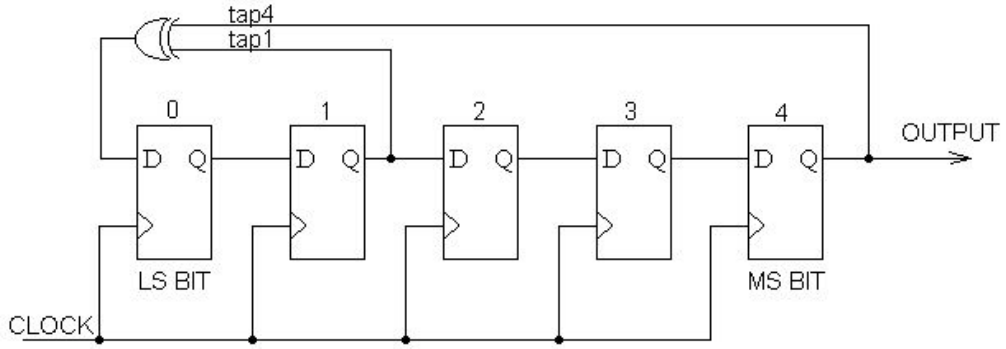
# LFSR, números pseudo aleatorios

*José Galaviz*

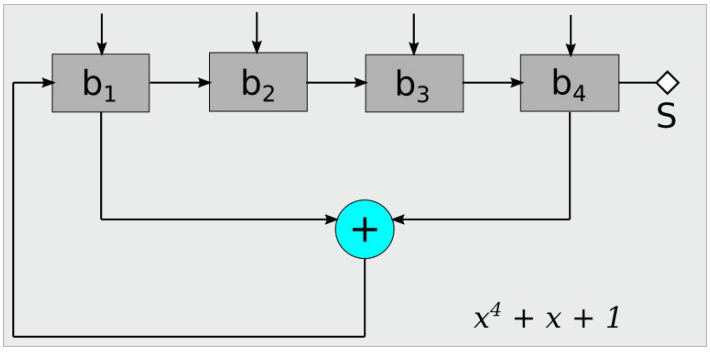
## ¿Qué son?

- Registros de desplazamiento que producen una salida secuencial regulada normalmente por un reloj. Un bit en cada pulso.
- Luego de cada pulso también se actualiza el estado del registro usando una función lineal de su estado previo.
- Se parte de un estado inicial arbitrario.
- Típicamente algunos de los bits del registro (taps) se conectan a un XOR, cuya salida regresa al registro por un extremo.
- En el extremo opuesto se produce la salida.

## Ejemplo




## Ejemplo




Para identificar a los taps se usa un modelo de polinomio

## Uso, características

- Como la secuencia de salida no parece tener un patrón, se puede pensar en usarlos en un “stream cipher”.
- Sin embargo existe un patrón:
  - Un LFSR de $n$ bits puede generar una secuencia de hasta $2^n-1$ bits, que luego se repite cíclicamente.
  - El estado inicial sólo determina donde se comienza la secuencia.
  - No todo polinomio genera secuencias máximas.

## Ejemplo de periodo corto

Polinomio $x^2+x+1$.

La secuencia de salida tiene periodo 3.

¿Cómo debe ser el polinomio de alambrado?

| | b1 | b2 | b3 | b4 | b1 xor b2 | S |
|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 1 | 1 | 1 | 1 |
| **3** | 1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 0 | 1 | 0 | 1 |
| 5 | 0 | 1 | 1 | 0 | 1 | 0 |
| **6** | 1 | 0 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 | 0 | 1 |
| 8 | 0 | 1 | 1 | 0 | 1 | 0 |
| 9 | 1 | 0 | 1 | 1 | 1 | 1 |

# Polinomio irreducible

No puede factorizarse como el producto de polinomios de grado menor no constantes.

Polinomio: $x^4+x+1$.

Periodo: $15 = 2^4-1$.

Polinomio irreducible.

| | b1 | b2 | b3 | b4 | b1 xor b4 | Salida | $x^4+x+1$ |
|---|---|---|---|---|---|---|---|
| **1** | **1** | **1** | **1** | **1** | 0 | **1** | S es igual a b1, es el bit que sale cada clock |
| 2 | 0 | 1 | 1 | 1 | 1 | **1** | b2 del renglón 1 |
| 3 | 1 | 0 | 1 | 1 | 0 | **1** | b3 del renglón 1 |
| 4 | 0 | 1 | 0 | 1 | 1 | **1** | b4 del renglón 1 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 | 1er resultado no establecido como entrada |
| 6 | 1 | 1 | 0 | 1 | 0 | 1 | |
| 7 | 0 | 1 | 1 | 0 | 0 | 0 | |
| 8 | 0 | 0 | 1 | 1 | 1 | 1 | |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 | |
| 10 | 0 | 1 | 0 | 0 | 0 | 0 | |
| 11 | 0 | 0 | 1 | 0 | 0 | 0 | |
| 12 | 0 | 0 | 0 | 1 | 1 | 1 | |
| 13 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 14 | 1 | 1 | 0 | 0 | 1 | 0 | |
| 15 | 1 | 1 | 1 | 0 | 1 | 0 | |
| 16 | 1 | 1 | 1 | 1 | 0 | 1 | Se repite el renglón 1, periodo 16-1=15 |

# Pero...

- Berlekamp (1967, para códigos BCH) y Massey (1969).
- Algoritmo para recuperar el polinomio de alambrado a partir de una secuencia del polinomio.
- En el ejemplo previo la secuencia de salidas observadas será:
  - 11110101

# Análisis

Se construye una transformación lineal a partir de una secuencia de tamaño 2n.

La transformación inversa proporciona el polinomio de alambrado.

Matriz M

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | | 0 |
| 1 | 1 | 1 | 0 | | 1 |
| 1 | 1 | 0 | 1 | | 0 |
| 1 | 0 | 1 | 0 | | 1 |

Inversa de M

| | | | | | |
|---|---|---|---|---|---|
| -1 | 0 | 1 | 1 | | |
| 0 | 1 | 0 | -1 | | |
| 1 | 0 | -1 | 0 | | |
| 1 | -1 | 0 | 0 | | |

Inversa de M (mód 2)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | | 0 | | 1 | 1 | X^4 |
| 0 | 1 | 0 | 1 | | 1 | | 2 | 0 | X^3 |
| 1 | 0 | 1 | 0 | | 0 | | 0 | 0 | X^2 |
| 1 | 1 | 0 | 0 | | 1 | | 1 | 1 | X |

## Números pseudo aleatorios

- Por definición, si existe un método, un algoritmo para generar algo, el algo no es aleatorio.
- Lo que generamos en la computadora es *pseudo-aleatorio*.

## LCG

Linear Congruential Generator.

$$X_{n+1} = (a\, X_n + c)\ (\text{mód}\ m)$$

Partiendo de una semilla $X_0$.

Secuencia determinista, finita, repetida en un ciclo que puede ser grande, en función del tamaño del tipo de dato usado.

No son suficientemente robustos para aplicaciones criptográficas.

## Requisitos

Un CSPRNG (generador de números pseudo-aleatorios criptográficamente seguro) debe:

- Con muy baja probabilidad generar secuencias idénticas de números.
- Generar secuencias estadísticamente indistinguibles de una verdaderamente aleatoria.
- Dada una secuencia de números debe ser imposible determinar el término siguiente.
- Dado un estado del generador, debe ser imposible calcular el estado previo del generador y los términos previos de la secuencia.

## Pruebas estadísticas

- Monobit test. En la secuencia de números, en binario, el número de ceros debe ser aproximadamente igual al de unos.
- Poker test. No debe haber repeticiones frecuentes del mismo dígito en los números: 959, 633, 882, etc.
- Kolmogorov-Smirnov. La distribución acumulativa de los términos de la secuencia debe corresponder a la de la distribución teórica con la que son generados.
- Run test. No debe haber corridas (subsecuencias con un patrón). P. ej. 3, **39**, 5, **25**, 13, **12**, 24, una creciente dentro de otra decreciente.
- Autocorrelación. Los términos de la secuencia deben ser independientes.

## Criterio de Yao

No debe haber un algoritmo polinomial para calcular el siguiente bit de la secuencia con probabilidad mayor a 0.5.

## ¿Qué se usa?

- Usar una función de hash criptográfica.
- O cifrados de bloque.
- A la función o cifrado se le pone en modo de operación “de contador” y se le suministra una semilla y luego la secuencia “1”, “2”, “3”, …
- Hay diseños más complejos basados en problemas matemáticos.
- Edward Snowden (2013) reveló que en uno de esos: NIST-800-90A (2007), la NSA puso una puerta trasera para predecir la secuencia.