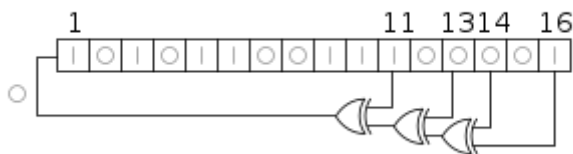


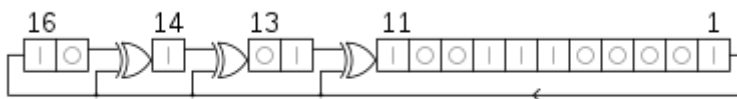
EJERCICIO 1: GENERADOR LFSR

Un generador LFSR (que significa Registro de desplazamiento de realimentación lineal, en sus siglas en inglés), se trata de un registro cuyo bit de entrada es función lineal de su estado anterior, es decir, que existe una realimentación en donde transmite los estados anteriores en la próxima iteración, que mediante compuertas (generalmente XOR) establecidos en cierto punto, generan una aleatoriedad a la salida del sistema. Existen dos técnicas de LFSR:

- **Fibonacci LFSR:** La característica de esta técnica, es que la realimentación se realiza con xor secuenciadas con el bit de salida, realimentadas al bit de la primera entrada.



- **Galois LFSR:** Esta es la técnica consiste en utilizar compuertas XOR independientes, ubicadas en lugares determinados, los cuales tomando valores de la realimentación y de iteraciones anteriores, se obtiene la aleatoriedad en la secuencia.



Actividades:

Actividad N1:

- Junto a este archivo se le proporciono el software “LFSRTestBench.exe”, el que sirve para simular la secuencia de bits que tendremos a la salida de nuestro generador LFSR:
 - 1- Se propone utilizar el software proporcionado para encontrar la combinación que genere el mayor número de combinaciones antes de que el patrón se vuelva a repetir.
 - 2. Una vez encontrada la combinación, implementar el código de verilog del generador:
 - A este generador se le debe poder configurar la seed inicial.
 - Solamente se generara una secuencia nueva cuando se obtenga una señal de valid en la entrada del módulo. El puerto de esta señal debe llamarse i_valid.
 - El módulo debe poder resetearse con un flanco positivo.
 - El módulo tendrá 2 resets:
 - El primer reset será asíncronico, en donde al accionarse, el módulo setee el valor fijado de seed a nuestro sistema. Este reset deberá llamarse i_rst
 - El segundo reset será síncronico. Al accionarse este reset, el sistema deberá registrar el valor de reset que veremos en puesto de entrada “i_seed”. Este reset deberá llamarse “i_soft_reset”.

Actividad N2:

- Se plantea realizar la verificación del código de verilog realizado anteriormente. Para ello se debe:
 - Generar un testbench donde:
 - Se utilice un clock con velocidad de 10MHz.
 - Se realice un reset, en donde el tiempo en el que se esté reseteando el módulo no debe ser un valor menor a 1us ni mayor a 250us.
 - Se genere una señal de valid, en donde aleatoriamente su valor cambie o no en cada ciclo de clock.
 - Se deberá crear task en donde se cambie el valor del puerto de entrada i_seed.
 - Se deberá crear dos tasks:

- Una que seteara por un tiempo random el reset asincrónico.
- Otra que hará lo mismo con el reset sincrónico.

Actividad N3:

- Se deberá crear tests en donde se pruebe:
 - Periodicidad del generador.
 - Periodicidad del generador con diferentes seeds random.

Actividad N4:

- Crear un “LFSR Checker”. Este será un nuevo módulo de verilog, que se conectara a la salida del generador LFSR. El propósito de este módulo es que en todos los ciclos de reloj se testeé que el valor obtenido del generador de PRBS sea el correcto:
 - Se deberán definir “thresholds” o límites para el bloqueo y desbloqueo del checker:
 - La señal que defina si el checker está bloqueado o no, se deberá llamar o_lock, y deberá ser un puerto de salida del checker.
 - El checker se bloqueara cuando detecte que al menos 5 nuevos valores del generador son válidos.
 - El checker se deberá desbloquear cuando detecte al menos 3 nuevos valores inválidos del generador.

Actividad N5:

- Una vez finalizado el checker, se deberá agregar este nuevo módulo al testbench y realizar las conexiones entre este módulo y nuestro generador inicial.
- Se deberá crear un proceso que chequee que el puerto o_lock siempre está en el mismo estado. En el caso de cambiar de estado, se deberá imprimir un mensaje informado que el estado cambio y cuál es su valor.
- Luego de realizar las conexiones, se deberán realizar los siguientes tests:
 - Generar tráfico valido y verificar que el checker se lockee y nunca se desbloquee por al menos un periodo de valores del generador.
 - Corroborar condiciones fronteras del “o_lock”:
 - Hacer un test en donde 4 datos sean válidos y 1 invalido. Verificar que nunca se lockee.
 - Hacer un test donde una vez lockeado, se generen 2 datos inválidos y 1 valido. Verificar que nunca se desbloquee.

- Hacer un test en donde genere 5 ciclos válidos y 3 ciclos inválidos. Verificar que siempre la salida transiciones entre estado de lock y unlock.
- Realizar tests en donde se reinicie en momentos random el checker. Ver que siempre luego del reset, el mismo se vuelva a lockear.