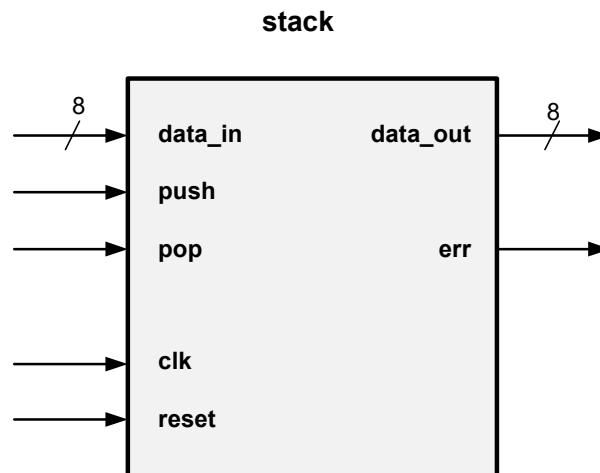
 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Ingeniería en Electrónica Técnicas Digitales I	2do parcial	Hoja 1/ ...
		Fecha: 22/11/17	
Apellido y nombre:	Leg.:	Calific.:	
Tiempo asignado al examen: 3 horas La resolución de todos los problemas debe estar justificada por tablas de verdad, expresiones lógicas o cuadros explicativos sin ambigüedades. La interpretación de la solución debe ser directa y lo escrito en el examen debe hablar por sí mismo. Si la resolución escrita de un problema requiere de aclaraciones posteriores del alumno se considerará mal resuelta. Los circuitos solo deben realizar la funcionalidad pedida y no más. Cualquier característica extra inválida la resolución salvo que sea inherente y no pueda ser eliminada.			Firma del Docente

Problema 1

Codifique en VHDL un circuito de **stack** cuya entidad es



Internamente la FIFO tiene 8 registros de 8 bits llamados **reg0**, **reg1**, **reg2**, **reg3**, **reg4**, **reg5**, **reg6** y **reg7**. El comportamiento es tal que al activarse la entrada **push** por primera vez, la palabra presente en **data_in** se almacena en **reg0**. La segunda vez que se active **push**, **data_in** se almacenará en **reg1** y así hasta llegar a **reg7**. La próxima vez que se active **push**, **data_in** no se almacena en ningún registro y se activa la salida **err**.

Aclaración. Si se deja activada la entrada **push**, esta es evaluada cada vez que se produzca un flanco ascendente en el reloj del sistema. No es necesario desactivar **push** para producir un nuevo almacenamiento.

Cuando se activa la entrada **pop** deberá presentar en la salida **data_out** la última palabra almacenada. Acto seguido la cantidad de palabras almacenadas disminuye en 1. Por ejemplo si la última palabra almacenada estaba **reg6**, luego de un pop, estará en **reg5**. Si no hubiera palabras almacenadas deberá activar la salida **err**.

Antes de codificar el circuito en VHDL dibuje un diagrama RTL.

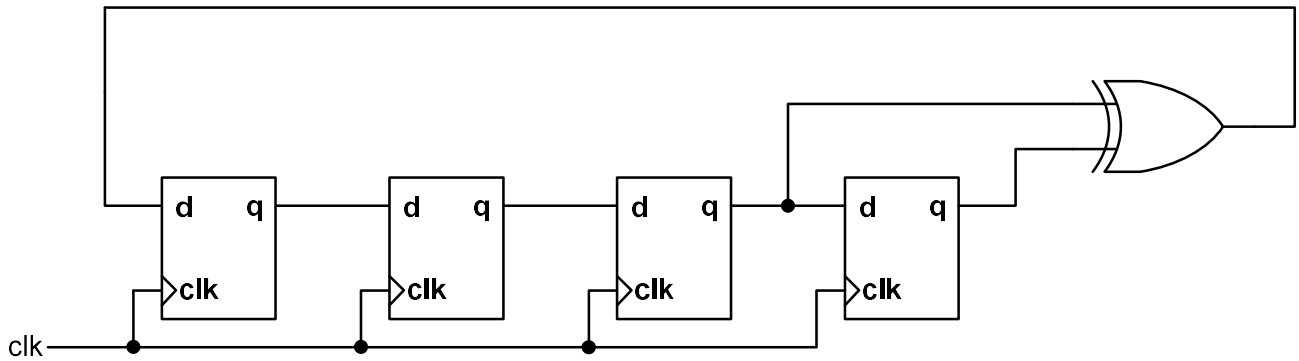
Problema 2

En el problema anterior es muy probable que deba construir un contador bidireccional. Haga el diseño en forma clásica siguiendo estos pasos

- Dibuje un contador bidireccional de 3 bits. Recuerde que necesita una entrada que controle el sentido de cuenta.
- En base a la regularidad del circuito dibuje una celda iterativa que permita construir contadores bidireccionales de la longitud de palabra deseada.
- Muestre el conexionado de varias celdas como las del punto **b)** para formar un contador bidireccional de 6 bits
- La celda del punto **b)** ¿permite obtener el **tc** sin lógica adicional? Si es afirmativa muestre como.

Problema 3

La figura muestra un contador LFSR de 4 bits



La tecnología con la que se lo fabricará se caracteriza por estos parámetros temporales

$t_{SU} = 2000 \text{ ps}$

$t_h = 800 \text{ ps}$

$t_{CQ} = 1500 \text{ ps}$

$t_{NOT} = 700 \text{ ps}$

$t_{NAND} = t_{NOR} = 1000 \text{ ps}$

$t_{AND} = t_{OR} = 1500 \text{ ps}$

$t_{XOR} = t_{XNOR} = 2500 \text{ ps}$

- a)** Confecciones las tablas de transiciones y estado. ¿Existen estados que no formen parte de la secuencia principal?
- b)** ¿Cuál es su módulo? ¿Cómo obtiene la señal de **tc**?
- c)** Si la pregunta del punto **a)** fuera afirmativa, rediseñe el LFSR para que en caso de caer en un estado inválido el circuito entre en la secuencia principal en único ciclo de reloj.
- e)** ¿Cuál es su frecuencia máxima de operación?
- f)** ¿Funciona si el circuito presenta un *clock skew* de $+600\text{ps}$.
- g)** ¿Funciona si el circuito presenta un *clock skew* de $+1500\text{ps}$.