

Prueba 2: Procesado digital de la señal con FPGAs

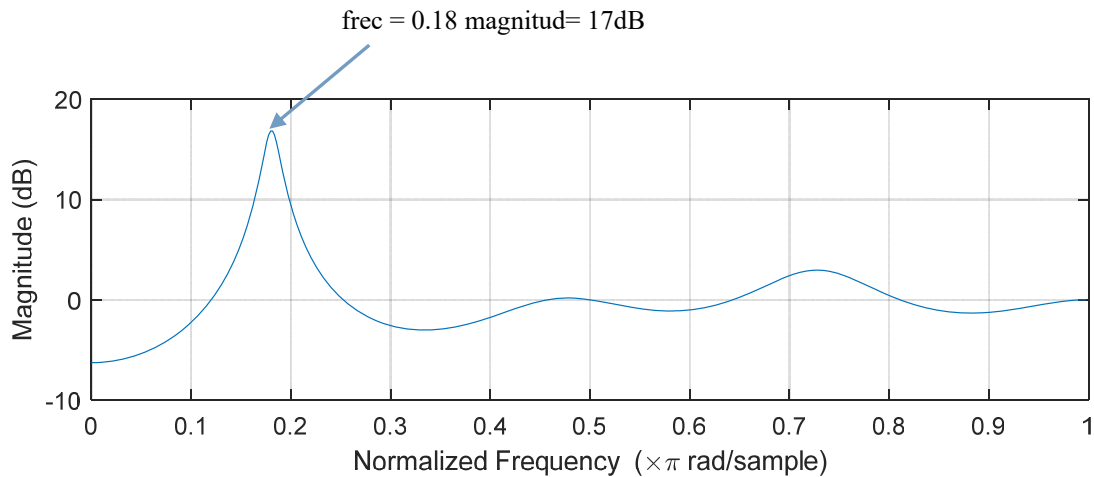
Alumno:

11/5/2018

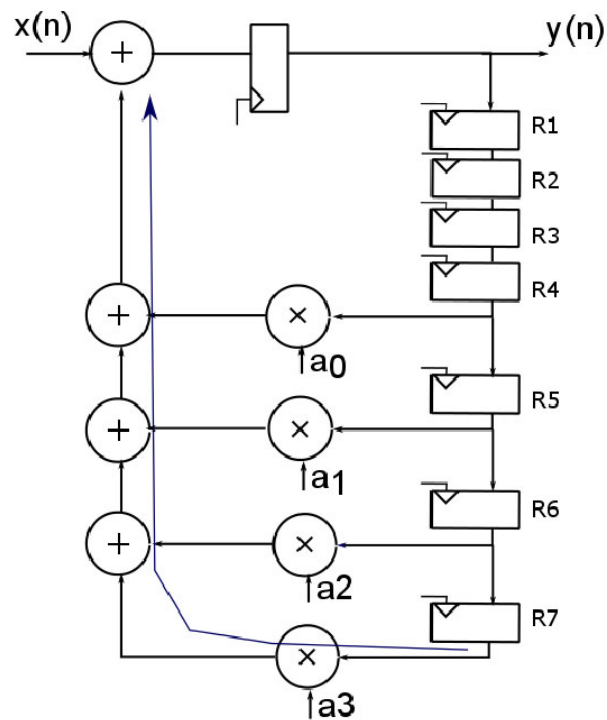
Ejercicio 1 (4 Puntos)

Se ha diseñado un filtro recursivo paso banda cuya ecuación en diferencias y respuesta en frecuencia se muestran a continuación:

$$y(n) = x(n-2) - a_0 \cdot y(n-4) - a_1 \cdot y(n-5) - a_2 \cdot y(n-6) - a_3 \cdot y(n-7)$$



a) Dibuje la arquitectura paralela con la que implementar el algoritmo anterior



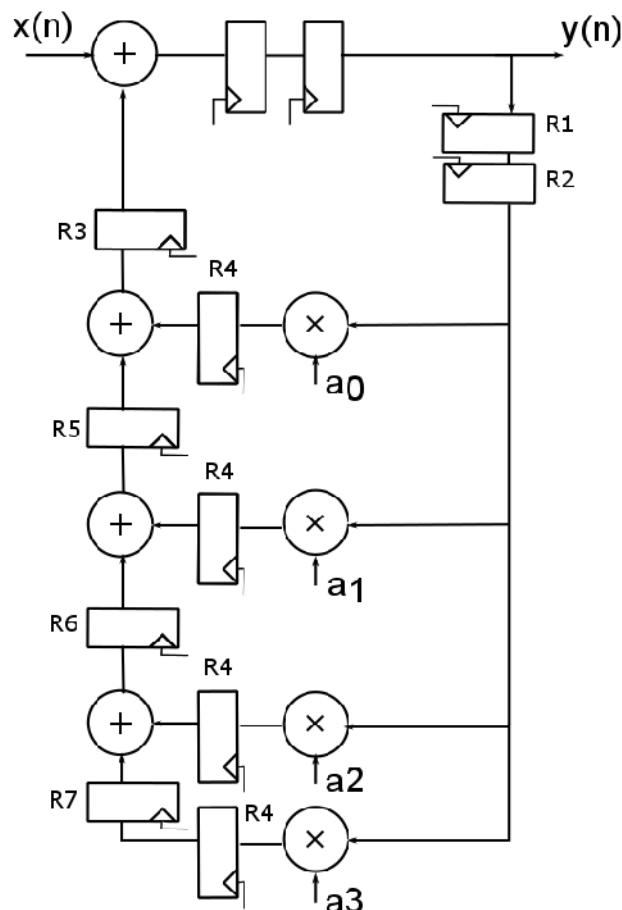
- b) Sobre el esquema anterior dibuje el camino crítico del circuito y halle la máxima frecuencia de funcionamiento sabiendo que $t_{mult}=3.5ns$ y $t_{add}=1.5ns$ (NOTA: considere nulos los tiempos de set-up y propagación de los registros para resolver este ejercicio).

El camino crítico es el que atraviesa el multiplicador cuyo coeficiente es a_3 y los cuatro sumadores en cascada.

*El tiempo de propagación de dicho camino es $t_{cc}=t_{mult} + 4 * t_{add} = 3.5 + 3 * 1.5 = 9.5ns$*

Por lo tanto, la máxima frecuencia de funcionamiento del circuito es $f_{clk}=1/9.5ns = 105.3 MHz$.

- c) ¿Se puede aumentar la frecuencia de funcionamiento del filtro mediante la aplicación de alguna técnica arquitectural? Si su respuesta es positiva, dibuje la arquitectura del filtro resultante de la aplicación de dichas técnicas e indique cuál será la máxima frecuencia de funcionamiento.



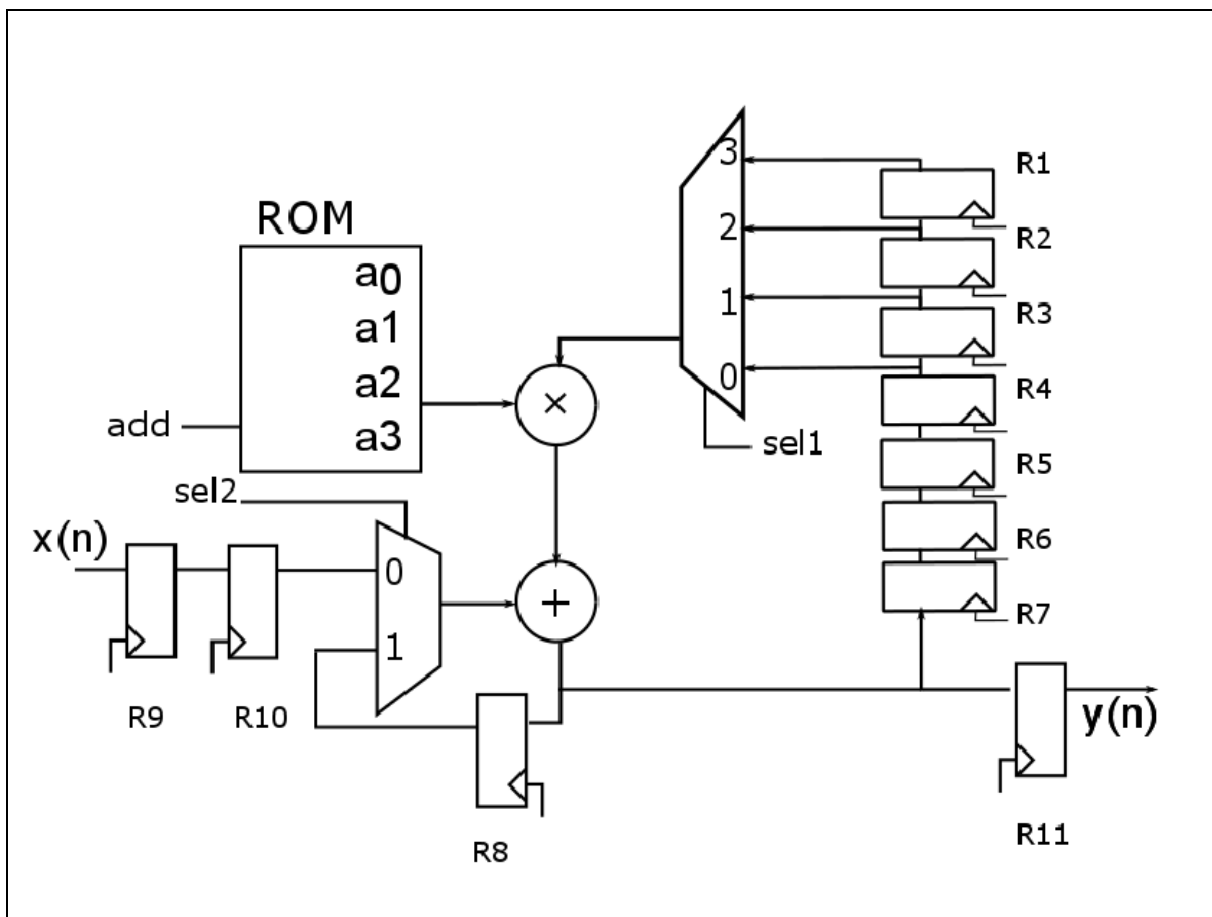
Dado que tenemos varios registros concatenados podemos utilizar “retiming” para reducir el camino crítico, tal y como mostramos en la figura anterior.

De este modo el camino crítico recorrerá un solo multiplicador, por lo que la frecuencia máxima de funcionamiento será $t_{cc}=1/t_{mult} = 1/3.5ns = 285.7MHz$

Ejercicio 2 (6 Puntos)

Partiendo de la ecuación en diferencias del ejercicio anterior, se desea realizar una implementación serie para conseguir el mínimo uso de recursos. En la siguiente figura se muestra el esquema de la implementación serie. Cada uno de los registros presentes en el circuito poseen una entrada de habilitación (ce) y una de reset (rst).

$$y(n) = x(n-2) - a_0 \cdot y(n-4) - a_1 \cdot y(n-5) - a_2 \cdot y(n-6) - a_3 \cdot y(n-7)$$



- a) Explicar el funcionamiento del circuito secuencial propuesto indicando claramente cuántos ciclos se necesitan para realizar el cómputo completo y qué operación se realiza en cada ciclo.

El circuito propuesto utilizará cuatro ciclos para computar la muestra $y(n)$.

En el primer ciclo se realizará la suma de $x(n-2) + a_0 \cdot y(n-4)$, de manera que se selecciona la entrada 0 del multiplexor controlado por $sel2$.

En los siguientes ciclos se realiza la acumulación del resultado del ciclo anterior más $a_x \cdot y(n-(4+x))$ siendo $x=[1,2,3]$. Para ello se debe seleccionar la entrada 1 del multiplexor.

- b) Cada uno de los registros presentes en el circuito poseen una entrada de habilitación (ce) y una de reset (rst). Justifique, para cada registro, cómo tienen que ser estas líneas de control

Los registros R1 a R7 se utilizan para desplazar los datos de la realimentación. Deberán trabajar a la frecuencia de muestreo por lo que necesitarán una señal de habilitación $ce1$ que controle el desplazamiento de los datos cada $f_{clk}/4$. Estos registros llevarán un reset ($rst1$) que sólo se activará al inicializar el circuito.

El registro R8 es el que se encarga de acumular los productos parciales. Este registro necesita realizar un reset ($rst2$) cuando comience el cómputo de una nueva muestra. La entrada de habilitación de este registro ($ce2$) estará siempre a nivel alto, permitiéndole funcionar a f_{clk} .

Los registros R9 y R10 se utilizan para cargar los datos de entrada $x(n)$. Estos registros deberán funcionar a $f_s=f_{clk}/4$ por lo que necesitan una señal de habilitación que lo permita ($ce3$). El reset de estos registros se utilizará para inicializar el circuito (igual que $rst1$).

El registro R11 se utiliza para obtener la salida y se controla de la misma manera que R1-R7, de hecho, podría evitarse este registro obteniendo la salida directamente del registro R7.

Resumiendo:

- R1-R7 y R11 se controlan con $ce1$ y $rst1$
- R8 se controla con $ce2$ y $rst2$
- R9 y R10 se controlan con $ce3$ y $rst1$

- c) ¿Cuál será el tamaño de la memoria ROM que almacenará los coeficientes si se cuantifican con 10 bits? ¿Cuántas LUTs se requieren si se implementase como memoria distribuida de un dispositivo FPGA Cyclone V?

Necesitamos almacenar 4 coeficientes de 10 bits cada uno, por lo que el tamaño de la memoria será $2^2 \times 10$ bits.

Dado que la memoria es pequeña se podría implementar de manera distribuida, usando 10 LUTs de 2 entradas.

- d) ¿Cuál es la frecuencia máxima de funcionamiento del circuito anterior? Considere los siguientes tiempos: $t_{add}=1.5ns$, $t_{mux}=2ns$, $t_{mult}=3.5ns$ y $t_{ROM}=2ns$. ¿Cuál será la frecuencia máxima de muestreo del circuito? Justifique la respuesta

El camino crítico recorre 1 multiplexor, 1 multiplicador y 1 sumador, por lo que la frecuencia máxima será:

$$f_{clkmax} = 1/(2ns+3.5ns+1.5ns) = 1/7ns = 142.8 \text{ MHz}$$

El circuito necesita cuatro ciclos para completar el cálculo de $y(n)$ por lo que la frecuencia máxima de muestreo será

$$f_{smax} = f_{clkmax}/4 = 35.7 \text{ MHz}$$

- e) ¿Es posible segmentar el circuito para alcanzar mayor frecuencia de muestreo?

Si segmentamos a la salida del multiplexor, la salida de la ROM y la salida del multiplicador, el camino crítico vendrá dado por el operador más lento, que en este caso es el multiplicador. De este modo obtendremos una $f_{clkmax}=1/3.5ns=285.7 \text{ MHz}$.

Con esta segmentación la latencia del circuito aumenta en dos ciclos por lo que la frecuencia de muestreo máxima será $f_{smax} = f_{clkmax}/(4+2) = 285.7/6=47.6 \text{ MHz}$

Por lo que sí que se consigue mejorar la frecuencia de muestreo al segmentar.

- f) Explique claramente el funcionamiento de todas las señales de control que se utilizarán para controlar los registros **del circuito sin segmentar**. Realice una tabla en la que indique el valor de todas las líneas de control en cada ciclo

La señal de habilitación de los registros (ce) se activará cada cuatro ciclos para permitir el desplazamiento de los datos de la realimentación ($y(n-x)$), la entrada de un nuevo dato ($x(n-2)$) y la salida del resultado ($y(n)$). Esta señal se activará durante el último ciclo, en el cual se han calculado las cuatro acumulaciones.

Por otro lado, el registro del acumulador necesitará “limpiar” la acumulación cada cuatro ciclos para evitar que se afecte al cálculo de la siguiente muestra. Esta operación se realizará utilizando el reset del registro. Se realiza este reset en el primer ciclo.

La memoria y el multiplexor de los datos se direccionarán de la misma manera, seleccionando consecutivamente los coeficientes a_0 , a_1 , a_2 y a_3 , a la vez que se seleccionan los datos $y(n-4)$, $y(n-5)$, $y(n-6)$ y $y(n-7)$. Por lo tanto, las líneas de control add y $sel1$ son idénticas.

Ciclo	add	$sel1$	$sel2$	$ce1$	$rst1$	$ce2$	$rst2$	$ce3$
0	00	00	0	0	0*	1	1	1
1	01	01	1	0	0	1	0	0
2	10	10	1	0	0	1	0	0
3	11	11	1	1	0	1	0	0

** sólo se activa para inicializar el circuito completo*

- g) Mapee las líneas de control en una memoria de forma que sea capaz de generar todo el control del circuito.

Se puede reducir el número de señales de control ya que:

- add y $sel1$ son iguales
- $ce2$ siempre está a 1
- $rst2$ y $ce3$ son iguales

Por lo tanto podemos reducir a 5 bits de control $\{add(2\text{ bits}), sel2, ce1, rst2\}$

El contenido de la memoria será el siguiente:

Posición	$\{add(2\text{ bits}), sel2, ce1, rst2\}$
0	00001
1	01100
2	10100
3	11110

- h) Trabajando con el **circuito sin segmentar**, se desea aplicar la misma cuantificación que en el apartado d del ejercicio 1. Cuantificar la salida de todos los operadores internos del filtro e indique claramente qué bits se seleccionan como salida del filtro.

La ganancia máxima del filtro es 17dB por lo que necesita $\text{ceil}(\log_2(10^{(17/20)})) = 3$ bits para evitar el desbordamiento a la salida del filtro.

El formato de la ruta de datos será sfix(16,9) (los 9 bits de precisión más los 4 bits enteros que necesitamos para evitar el desbordamiento).

Al multiplicador le llega un dato sfix[16,12] que se multiplicará por un coeficiente sfix[10,9]. Esto nos da una cuantificación de salida del multiplicador sfix[26,21]. Como se va a trabajar con 12 bits de precisión y 4 enteros nos quedaremos con los bits [24:9] obteniendo un formato sfix[16,12].

El acumulador será de 16 bits.

