
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

# Aplicación de las Arquitecturas Paralelas y Secuenciales al Procesado Multitasa

## Procesado Digital de la Señal en FPGA

**Máster Universitario en Ingeniería de Sistemas Electrónicos**

# Objetivos

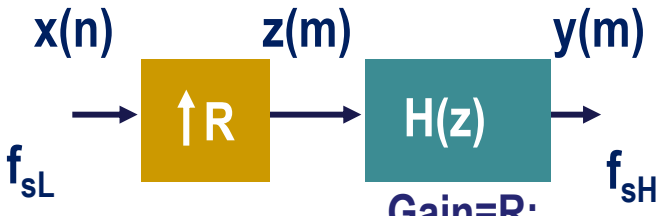
- Implementación de algoritmos multitasa aplicando aquitecturas paralelas o secuenciales

# Contenidos

- Procesado multitasa
- Arquitecturas polifásicas: Diezmadores
- Arquitecturas polifásicas: Interpoladores

# Arquitecturas FIR Multitasa

## Interpolación

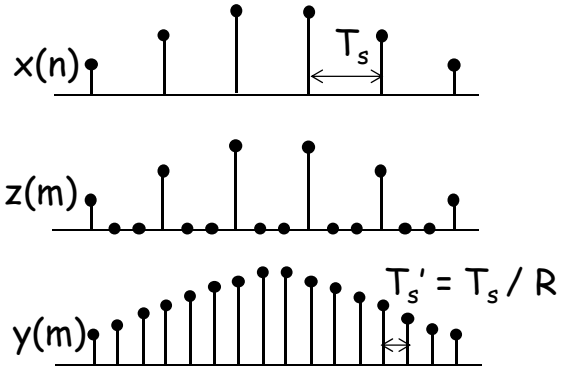


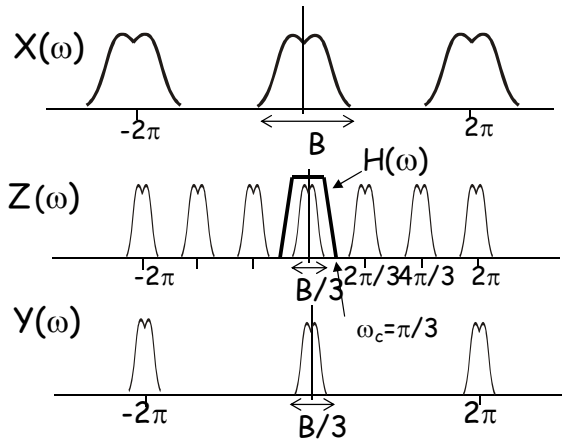

Gain=R;
$\omega_c=\pi/R$

Bloque expansor:

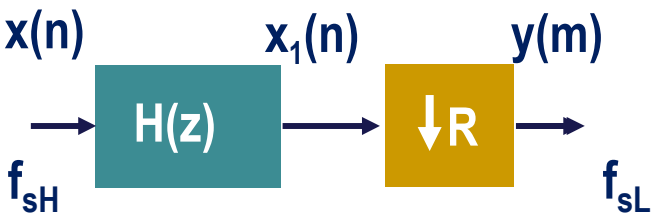
$$z(m) = \begin{cases} x(m/R) & \text{si } m=0, \pm R, \pm 2R,... \\ 0 & \text{otros} \end{cases}$$

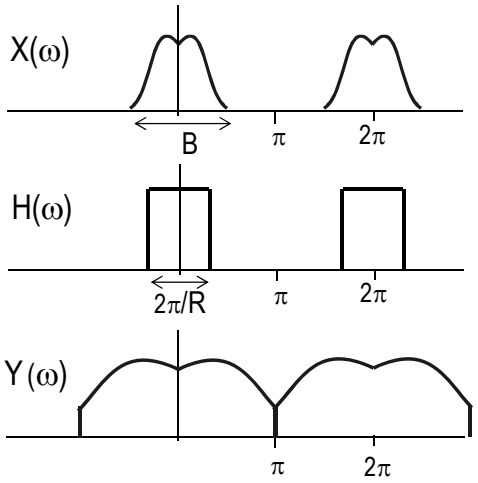
$$f_{sH} = R\, f_{sL}$$




## Diezmado



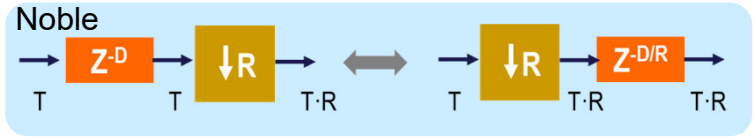



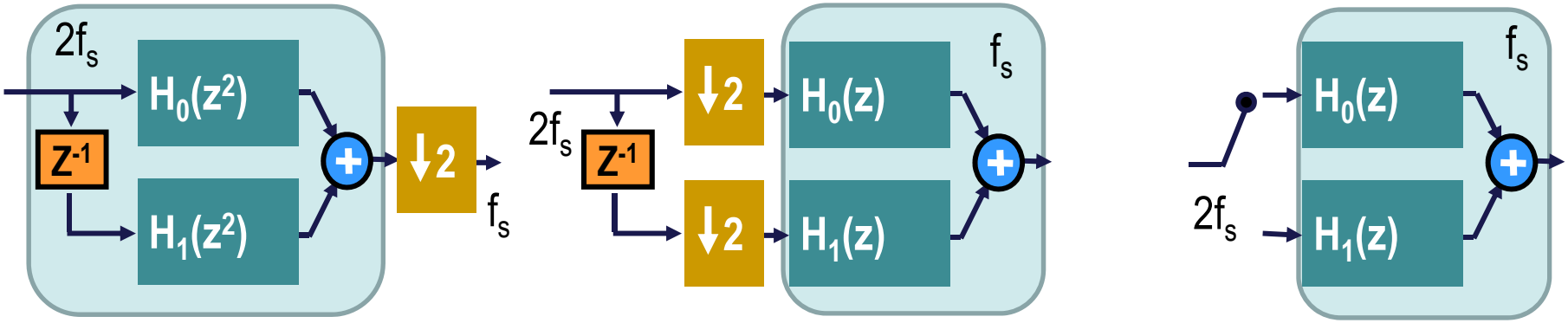
# Arquitecturas polifásicas

Descomposición polifásica en dos subfiltros:

$$H(z)=\sum_{i=0}^{M} h_i z^{-i}=\sum_{i=0}^{M/2} h_{2i} z^{-2i}+z^{-1}\sum_{i=0}^{M/2} h_{2i+1} z^{-2i}=H_0(z^2)+z^{-1}H_1(z^2)$$

## Diezmado por 2



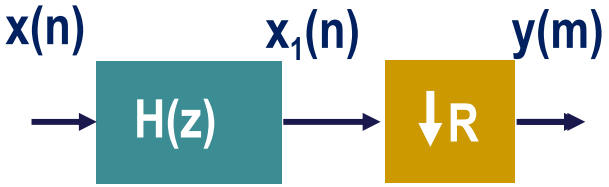



# Arquitecturas Polifásicas: Diezmado

Ej: Diezmado por R=2 con un FIR de 8 etapas

Sólo se necesita la salida en los siguientes instantes:

y(0), y(R), y(2R), ...

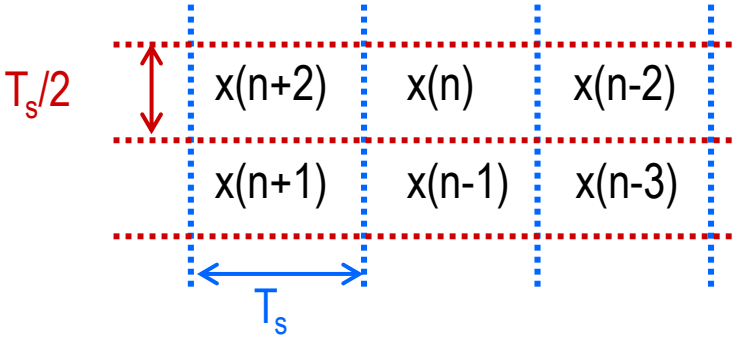


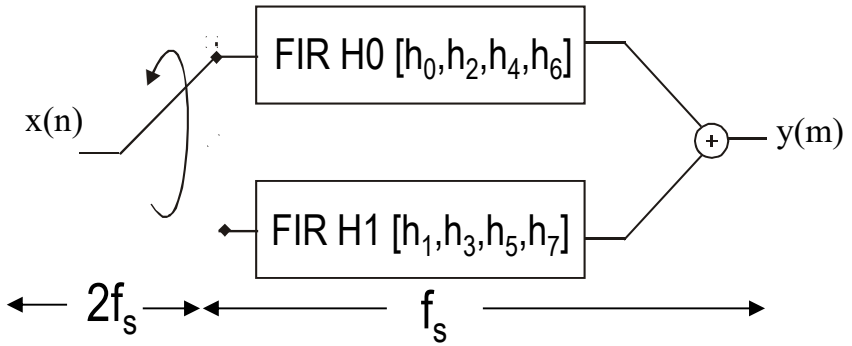

$$y(m)=x_1(n)=h_0x(n)+h_1x(n-1)+h_2x(n-2)+h_3x(n-3)+h_4x(n-4)+h_5x(n-5)+h_6x(n-6)+h_7x(n-7))=$$

$$= \underbrace{h_0x(n)+h_2x(n-2)+h_4x(n-4)+h_6x(n-6)}_{\textbf{FIR H0 } [h_0,h_2,h_4,h_6]} + \underbrace{h_1x(n-1)+h_3x(n-3)+h_5x(n-5)+h_7x(n-7)}_{\textbf{FIR H1 } [h_1,h_3,h_5,h_7]}$$

$$y(m+1)=x_1(n+2)=h_0x(n+2)+h_1x(n+1)+h_2x(n)+h_3x(n-1)+h_4x(n-2)+h_5x(n-3)+h_6x(n-4)+h_7x(n-5))=$$

$$= \underbrace{h_0x(n+2)+h_2x(n)+h_4x(n-2)+h_6x(n-4)}_{\textbf{FIR H0 } [h_0,h_2,h_4,h_6]} + \underbrace{h_1x(n+1)+h_3x(n-1)+h_5x(n-3)+h_7x(n-5)}_{\textbf{FIR H1 } [h_1,h_3,h_5,h_7]}$$



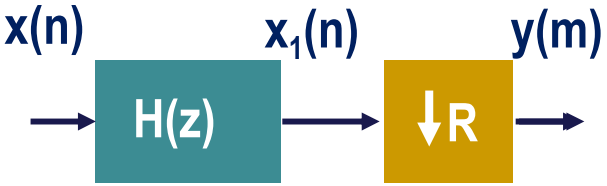



# Arquitecturas Polifásicas: Diezmado

Ej: Diezmado por R=3 con un FIR de 9 etapas

Sólo se necesita la salida en los siguientes instantes:

y(0), y(R), y(2R), ...

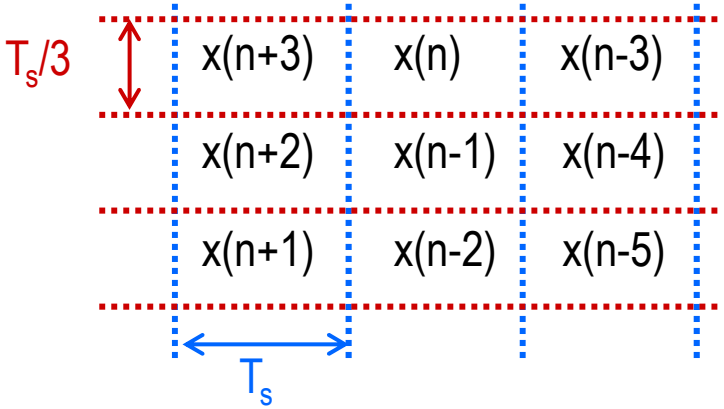


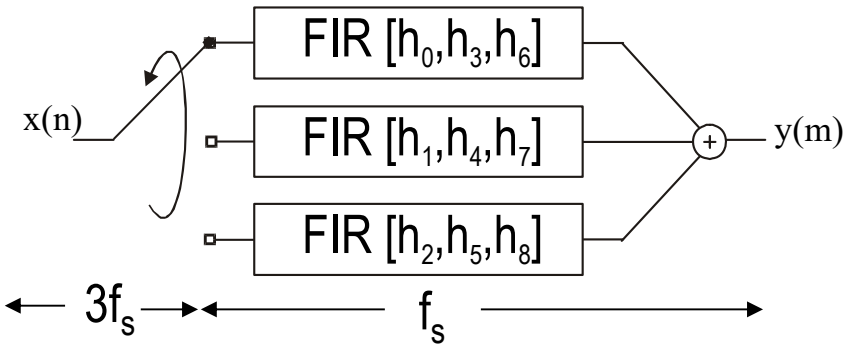

$$y(m)=x_1(n)=h_0x(n)+h_1x(n-1)+h_2x(n-2)+h_3x(n-3)+h_4x(n-4)+h_5x(n-5)+h_6x(n-6)+h_7x(n-7)+h_8x(n-8)=$$

$$= \underbrace{h_0x(n)+h_3x(n-3)+h_6x(n-6)}_{\mathbf{FIR[h_0,h_3,h_6]}} + \underbrace{h_1x(n-1)+h_4x(n-4)+h_7x(n-7)}_{\mathbf{FIR[h_1,h_4,h_7]}} + \underbrace{h_2x(n-2)+h_5x(n-5)+h_8x(n-8)}_{\mathbf{FIR[h_2,h_5,h_8]}}$$

$$y(m+1)=x_1(n+3)=h_0x(n+3)+h_1x(n+2)+ h_2x(n+1)+h_3x(n)+h_4x(n-1)+h_5x(n-2)+h_6x(n-3)+h_7x(n-4)+h_8x(n-5)=$$

$$= \underbrace{h_0x(n+3)+h_3x(n)+h_6x(n-3)}_{\mathbf{FIR[h_0,h_3,h_6]}} + \underbrace{h_1x(n+2)+h_4x(n-1)+h_7x(n-4)}_{\mathbf{FIR[h_1,h_4,h_7]}} + \underbrace{h_2x(n+1)+h_5x(n-2)+h_8x(n-5)}_{\mathbf{FIR[h_2,h_5,h_8]}}$$

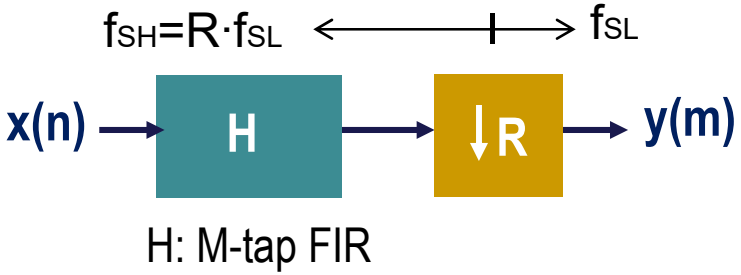


# Arquitecturas polifásicas: Diezmado

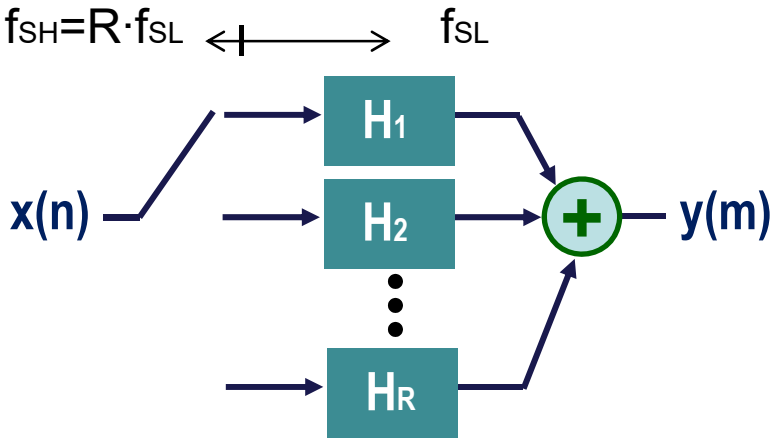
## Carga computacional (CL) del proceso de diezmado

- Fuerza bruta: $CL=M \cdot f_{SH}$



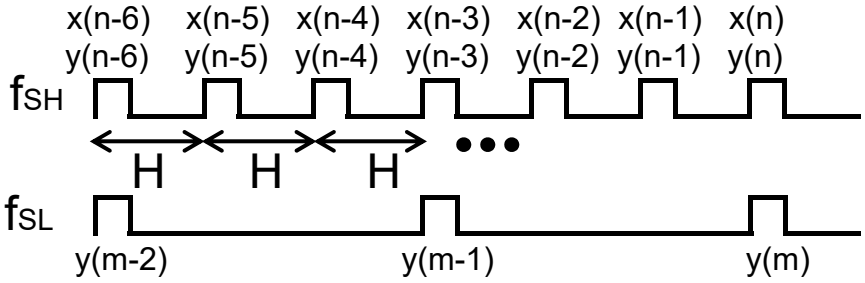

H: M-tap FIR

- Polifásico: $CL=M/R \cdot f_{SH}$

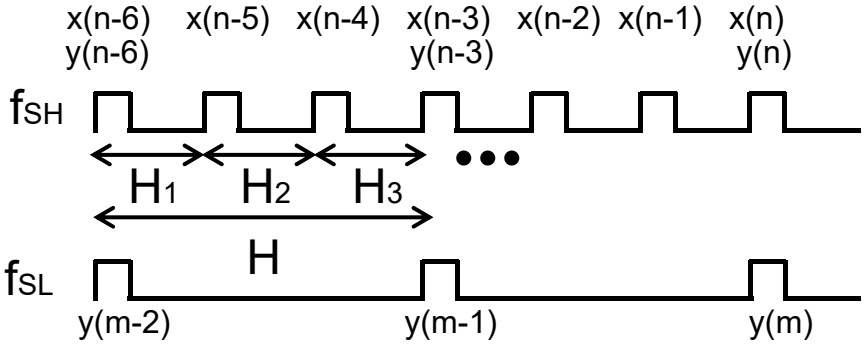



Hi: (M/R)-tap FIR

- Ej: FIR de 9 etapas, diezmador R=3




H: $\{h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8\}$
9 mults cada $T_{SH}$

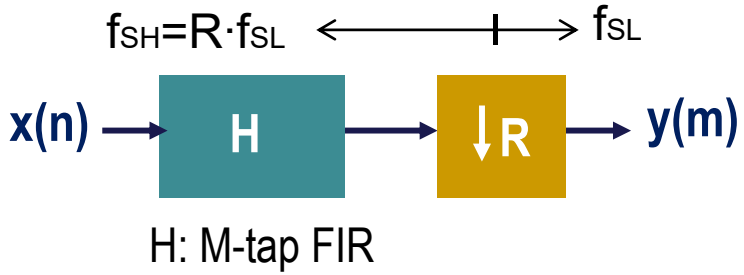



$H_1$: $\{h_0,h_3,h_6\}$, $H_2$: $\{h_1,h_4,h_7\}$, $H_3$: $\{h_2,h_5,h_8\}$
3 mults cada $T_{SH}$

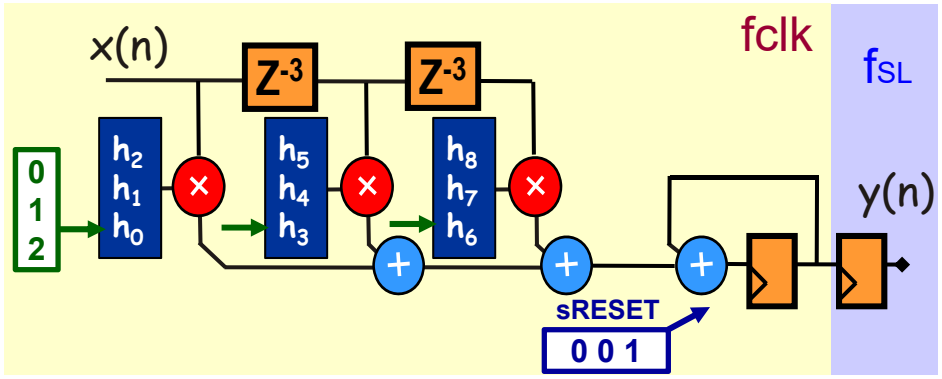
# Arquitecturas polifásicas: Diezmado

$f_{SL}=f_{clk}/3$

## Case: $f_{clk} = f_{SH}$

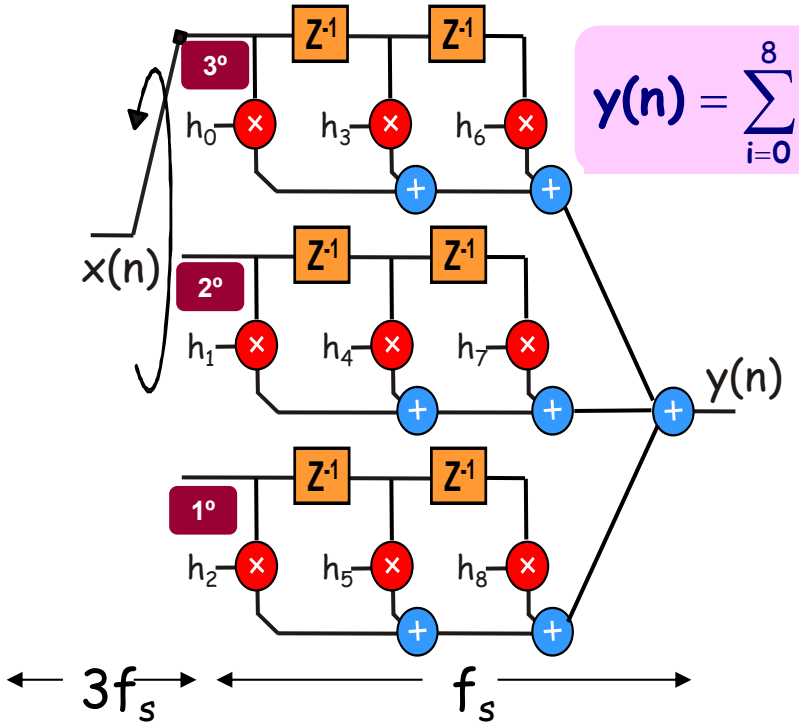



H: M-tap FIR

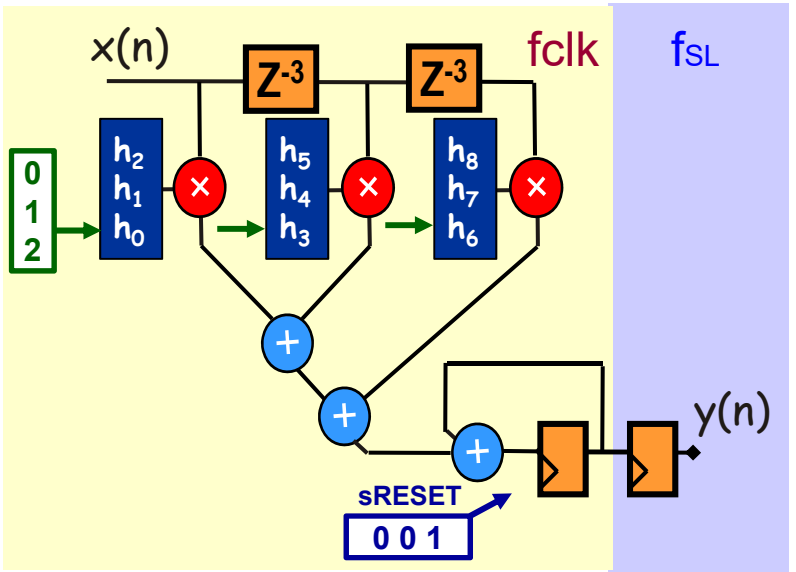



Estructura sumadores en cascada

## FIR de 9 etapas diezmador por R=3




$$y(n) = \sum_{i=0}^{8} h_i x(n-i)$$




Estructura sumadores en árbol

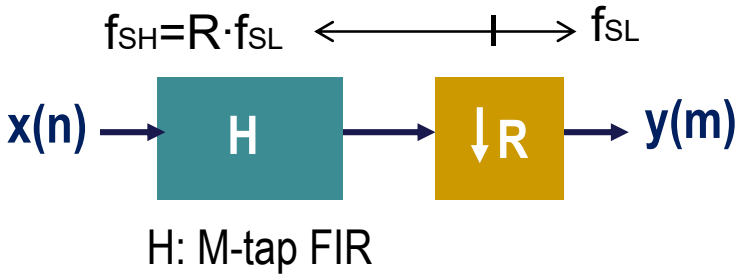
# Arquitecturas polifásicas: Diezmado

**Caso: $f_{clk} = f_{SH}$**

Cuantificación:

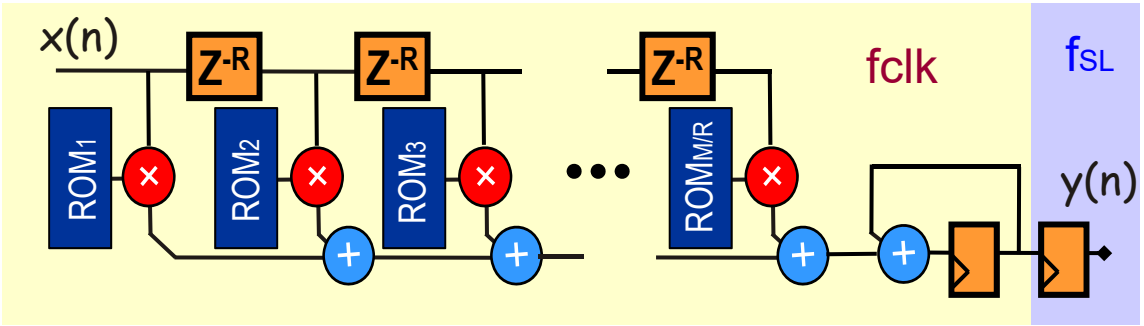
Datos Wd bits

Coef. Wc bits




## FIR de M etapas diezmador por R

$f_{SL}=f_{clk}/R$




ROM1: h(R:-1:1)
ROM2: h(2*R:-1:R+1)
ROM3: h(3*R:-1:2*R+1)
⋮

## Recursos HW:

- M registros of Wd bits
- M/R ROMs: RxWc bits
- M/R Multiplicadores: WdxWc bits
- M/R sumadores: Wd+Wc+g bits
- Registros: Wd+Wc+g bits (g:bits de guarda del acum.)
- Registro de salida: Wout bits

## Throughput:

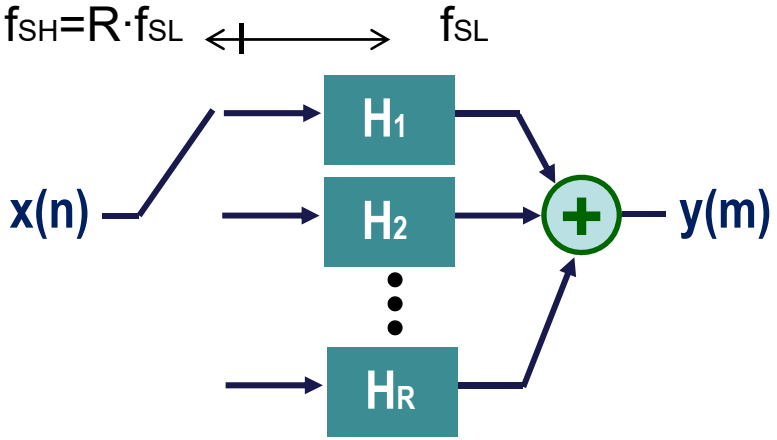
- $T_{out}$ = fclk/R
- $T_{in}$ = fclk

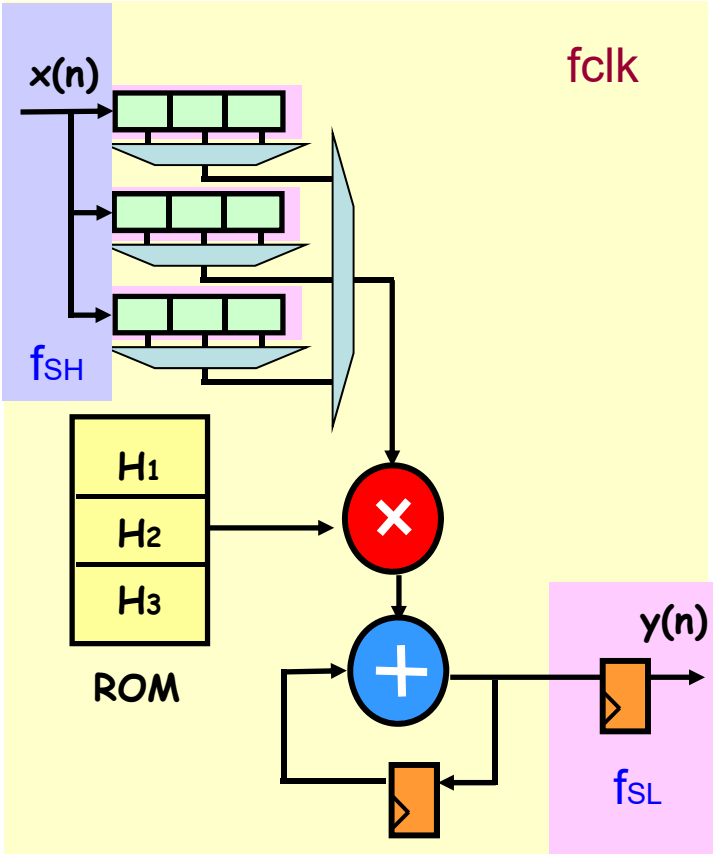
# Arquitecturas polifásicas: Diezmado

## Caso: $f_{clk}/f_{SH} \geq M/R$

**FIR de 9 etapas diezmador por R=3**

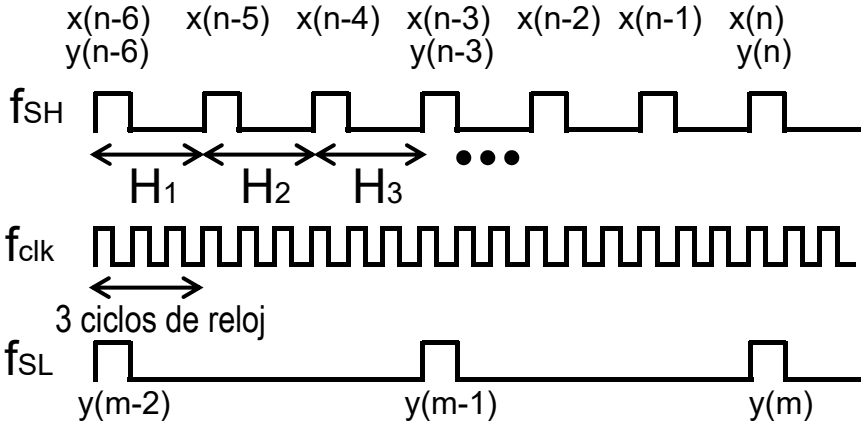
$$y(n) = \sum_{i=0}^{8} h_i x(n-i)$$



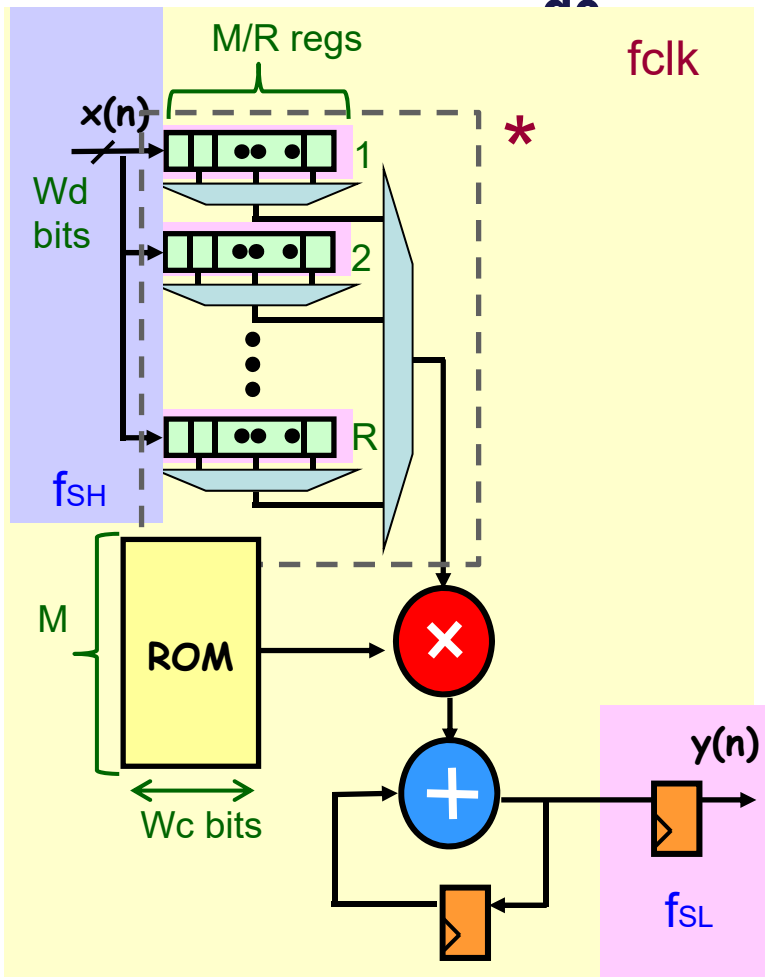

**ROM**

$H_1$: $\{h_2, h_5, h_8\}$
$H_2$: $\{h_1, h_4, h_7\}$
$H_3$: $\{h_0, h_3, h_6\}$



# Arquitecturas polifásicas: Diezmado

**Caso: $f_{clk}/f_{SH} \geq M/R$**




ROM = [h(M:-R:R) h(M-1:-R:R-1) … ]

**Filtro de M-etapas polifásico diezmador por R**

$$y(n) = \sum_{i=0}^{M-1} h_i x(n-i)$$

## Recursos HW:

- R shift-registers con M/R registros de Wd bits
- ROM: MxWc bits
- Multiplicador: WdxWc bits
- Sumador: Wd+Wc+g bits
- Registros (acum.): Wd+Wc+g bits (g: bits de guarda del acum)
- Registro de salida: Wout bits

## Throughput (max):

- $T_{out}$ = fclk/M
- $T_{in}$ = fclk/(M/R)

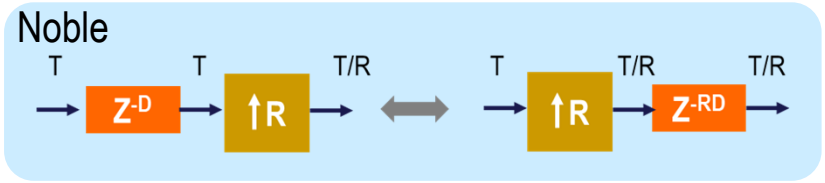
* SR se puede implementar con RAM si el orden del filtro es elevado

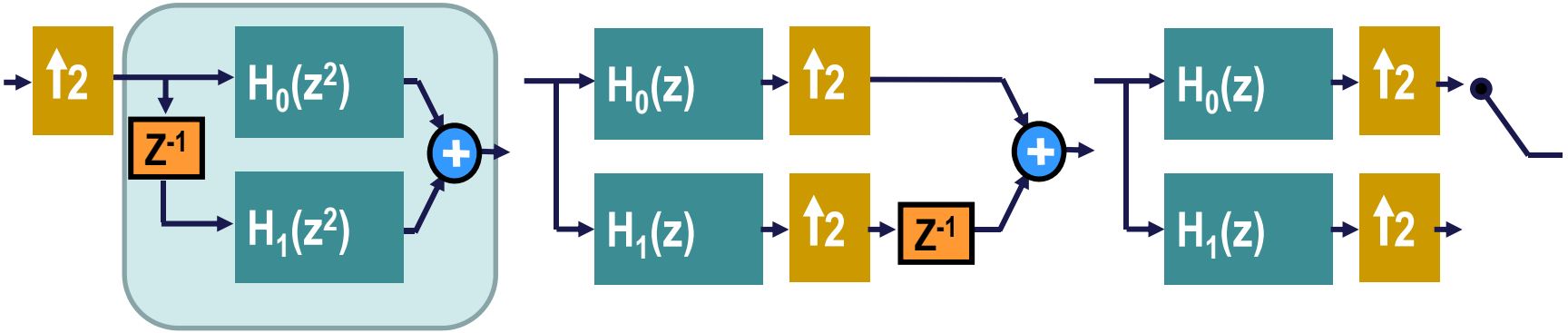
# Arquitecturas polifásicas: Interpolador

Descomposición polifásica en dos subfiltros:

$$H(z)=\sum_{i=0}^{M} h_i z^{-i}=\sum_{i=0}^{M/2} h_{2i} z^{-2i}+z^{-1}\sum_{i=0}^{M/2} h_{2i+1} z^{-2i}=H_0(z^2)+z^{-1}H_1(z^2)$$

## Interpolación por 2

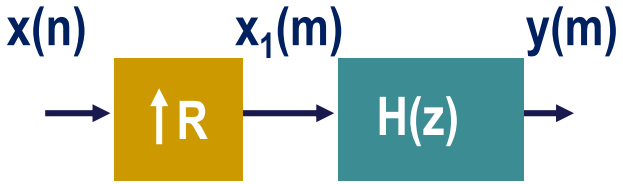


# Arquitecturas Polifásicas: Interpolación

Ej: Interpolación por 2 usando un FIR de 8 etapas

$$y(m) = \sum_{k=0}^{8} x(m-k)h_k$$

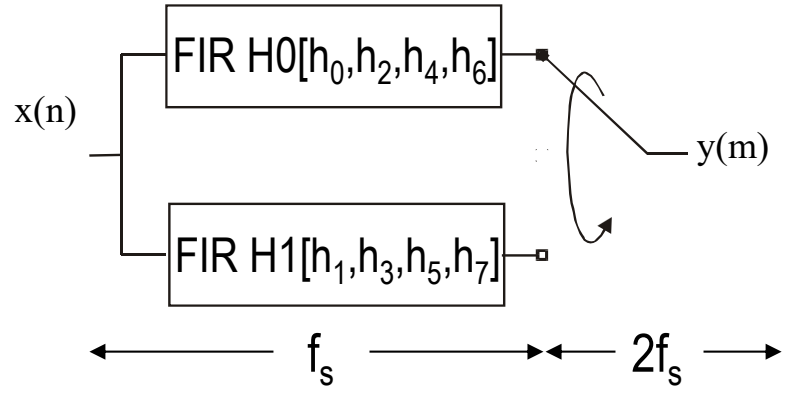



$y(m)=h_o x_1(m)+h_1 \cancel{x_1(m-1)}^{0}+h_2 x_1(m-2)+h_3 \cancel{x_1(m-3)}^{0}+h_4 x_1(m-4)+h_5 \cancel{x_1(m-5)}^{0}+h_6 x_1(m-6)+h_7 \cancel{x_1(m-7)}^{0}=$

$=h_o x_1(m)+h_2 x_1(m-2)+ h_4 x_1(m-4)+ h_6 x_1(m-6) = h_o x(n)+h_2 x(n-1)+ )+ h_4 x(n-2)+ h_6 x(n-3)$ = **FIR[$h_0$,$h_2$,$h_4$,$h_6$]**

$y(m+1)=h_o \cancel{x_1(m+1)}^{0}+h_1 x_1(m)+h_2 \cancel{x_1(m-1)}^{0}+h_3 x_1(m-2)+h_4 \cancel{x_1(m-3)}^{0}+h_5 x_1(m-4)+h_6 \cancel{x_1(m-5)}^{0}+h_7 x_1(m-6)=$

$=h_1 x_1(m)+h_3 x_1(m-2)+h_5 x_1(m-4)+h_7 x_1(m-6)=h_1 x(n)+h_3 x(n-1)+ )+ h_5 x(n-2)+ h_7 x(n-3)$ = **FIR[$h_1$,$h_3$,$h_5$,$h_7$]**



# Arquitecturas Polifásicas: Interpolación

Ej: Interpolación por 3 usando un FIR de 9 etapas

$$y(m) = \sum_{k=0}^{8} x(m-k)h_k$$

$y(m)=h_o x_1(m)+h_1 \overset{0}{\cancel{x_1(m-1)}}+h_2 \overset{0}{\cancel{x_1(m-2)}}+h_3 x_1(m-3)+h_4 \overset{0}{\cancel{x_1(m-4)}}+h_5 \overset{0}{\cancel{x_1(m-5)}}+h_6 x_1(m-6)+h_7 \overset{0}{\cancel{x_1(m-7)}}+h_8 \overset{0}{\cancel{x_1(m-8)}}=$

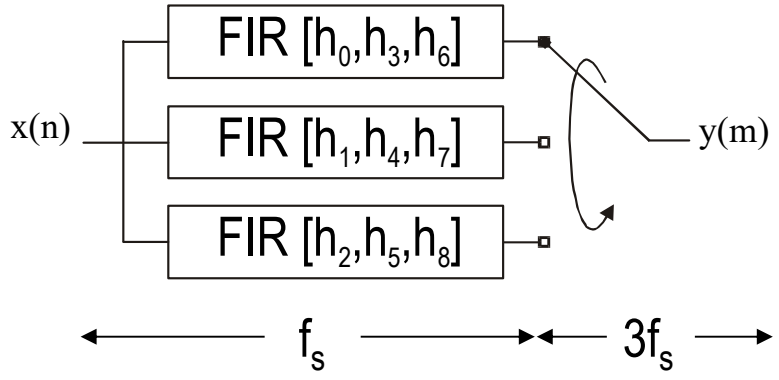
$=h_o x_1(m)+h_3 x_1(m-3)+ h_6 x_1(m-6) =h_o x(n)+h_3 x(n-1)+h_6 x(n-2) =$ **$FIR[h_0,h_3,h_6]$**

$y(m+1)=h_o \overset{0}{\cancel{x_1(m+1)}}+h_1 x_1(m)+ h_2 \overset{0}{\cancel{x_1(m-1)}}+h_3 \overset{0}{\cancel{x_1(m-2)}}+h_4 x_1(m-3)+h_5 \overset{0}{\cancel{x_1(m-4)}}+h_6 \overset{0}{\cancel{x_1(m-5)}}+h_7 x_1(m-6)+h_8 \overset{0}{\cancel{x_1(m-7)}}=$

$=h_1 x_1(m)+h_4 x_1(m-3)+h_7 x_1(m-6) =h_1 x(n)+h_4 x(n-1)+h_7 x(n-2) =$ **$FIR[h_1,h_4,h_7]$**

$y(m+2)=h_o \overset{0}{\cancel{x_1(m+2)}}+h_1 \overset{0}{\cancel{x_1(m+1)}}+h_2 x_1(m)+h_3 \overset{0}{\cancel{x_1(m-1)}}+h_4 \overset{0}{\cancel{x_1(m-2)}}+h_5 x_1(m-3)+h_6 \overset{0}{\cancel{x_1(m-4)}}+h_7 \overset{0}{\cancel{x_1(m-5)}}+h_8 x_1(m-6)=$

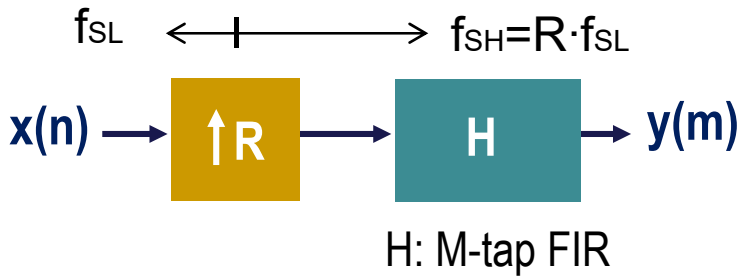
$=h_2 x_1(m)+ h_5 x_1(m-3)+ h_8 x_1(m-6) =h_2 x(n)+ h_5 x(n-1)+ h_8 x(n-2) =$ **$FIR[h_2,h_5,h_8]$**



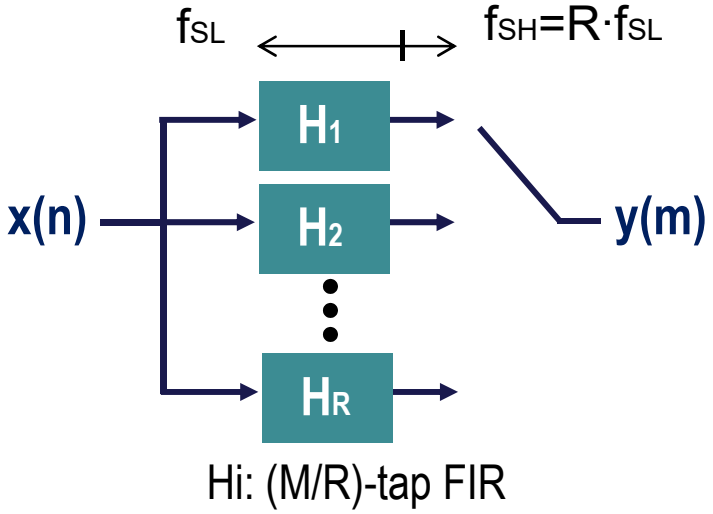
# Arquitecturas polifásicas: Interpolación

## Carga computacional del proceso de interpolación

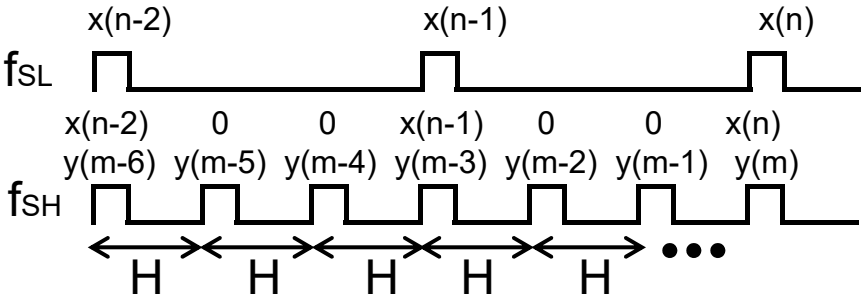
- Fuerza bruta: $CL=M \cdot f_{SH}$



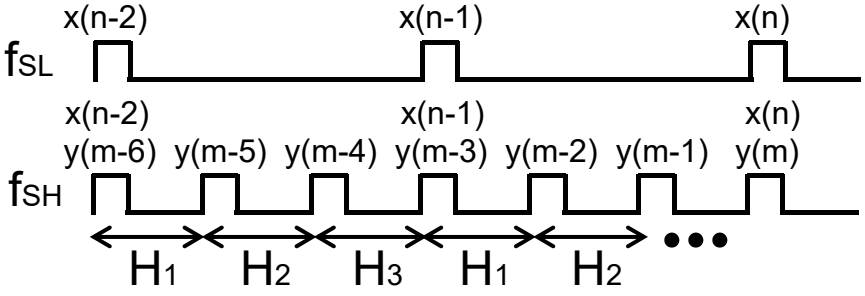

- Polifásico: $CL=M/R \cdot f_{SH}$




- Ej: FIR de 9 etapas interpolador por R=3




H: $\{h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8\}$
9 mults cada $T_{SH}$

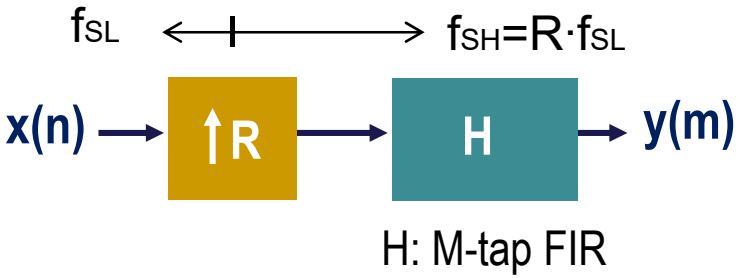



$H_1$: $\{h_0,h_3,h_6\}$, $H_2$: $\{h_1,h_4,h_7\}$, $H_3$: $\{h_2,h_5,h_8\}$
3 mults cada $T_{SH}$

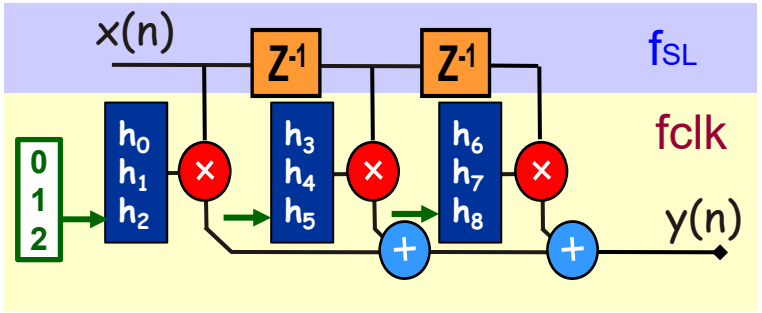
# Arquitecturas polifásicas: Interpolación

$f_{SL}=f_{clk}/3$

**Caso: $f_{clk} = f_{SH}$**



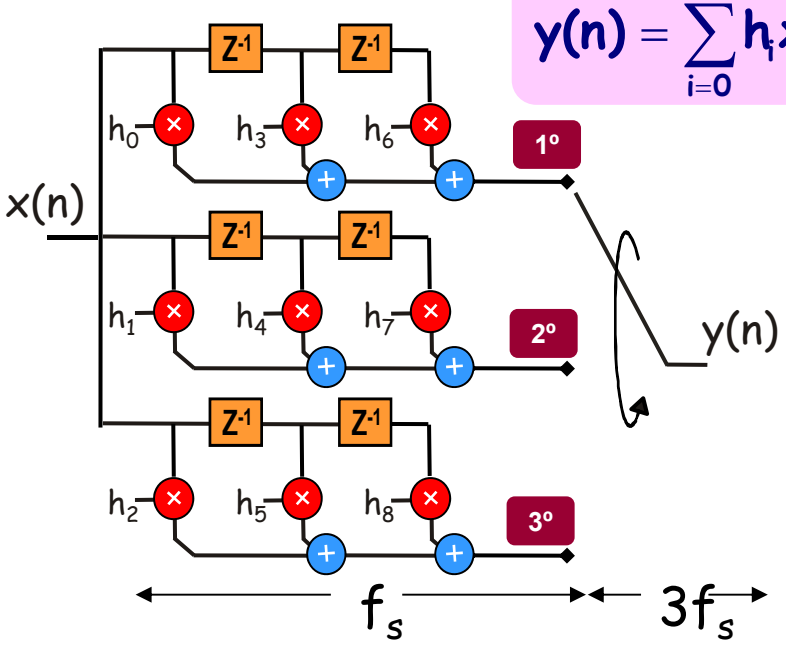

H: M-tap FIR

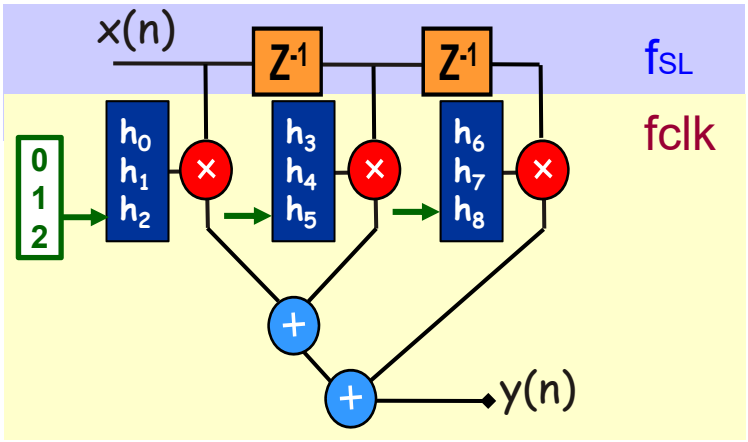



Estructura de sumadores en cascada

Filtro polifásico de 9 etapas Interpolador por 3

$$y(n) = \sum_{i=0}^{8} h_i x(n-i)$$



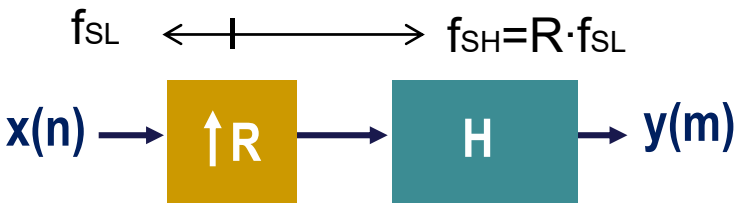




Estructura de sumadores en árbol

# Arquitecturas polifásicas: Interpolación

**Caso: $f_{clk} = f_{SH}$**

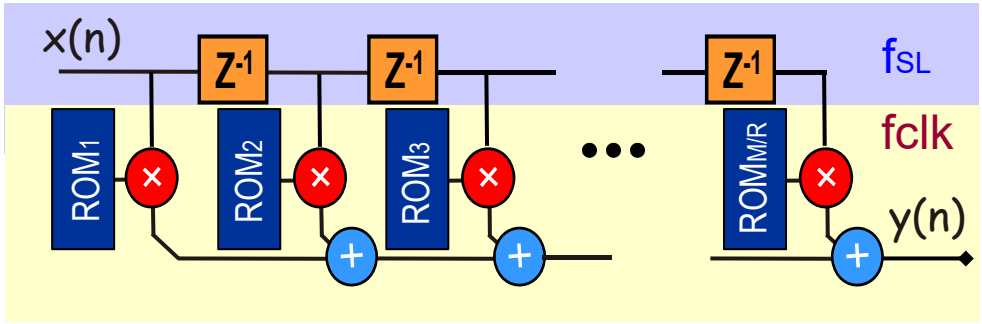
Cuantificación:
- Datos Wd bits
- Coef. Wc bits




H: M-tap FIR

## Filtro interpolador por R polifásico de M etapas

$f_{SL}=f_{clk}/R$




ROM$_1$: h(1:R)
ROM$_2$: h(R+1:2*R)
ROM$_3$: h(2*R+1:3*R)
⋮

## Recursos HW:

- M/R registros de Wd bits
- M/R ROMs: RxWc bits
- M/R Multiplicadores: WdxWc bits
- (M/R-1) sumadores: Wd+Wc+g bits
  (g: bits de guarda de la red de sumadores)

## Throughput:

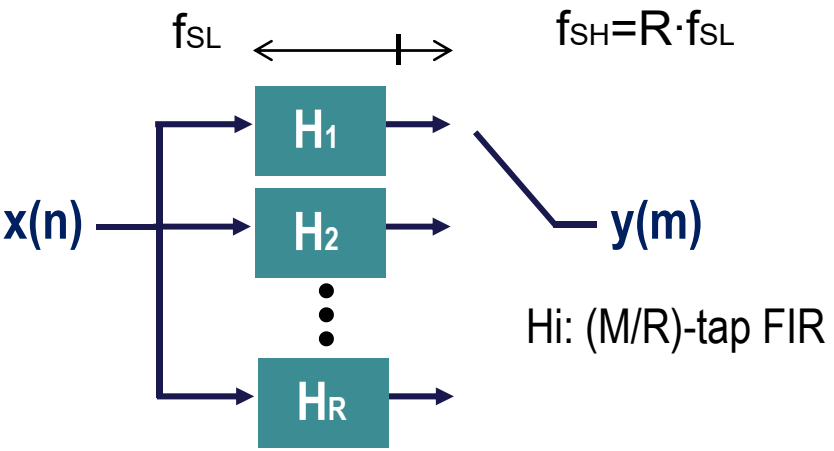
- $T_{out}$ = fclk
- $T_{in}$ = fclk/R

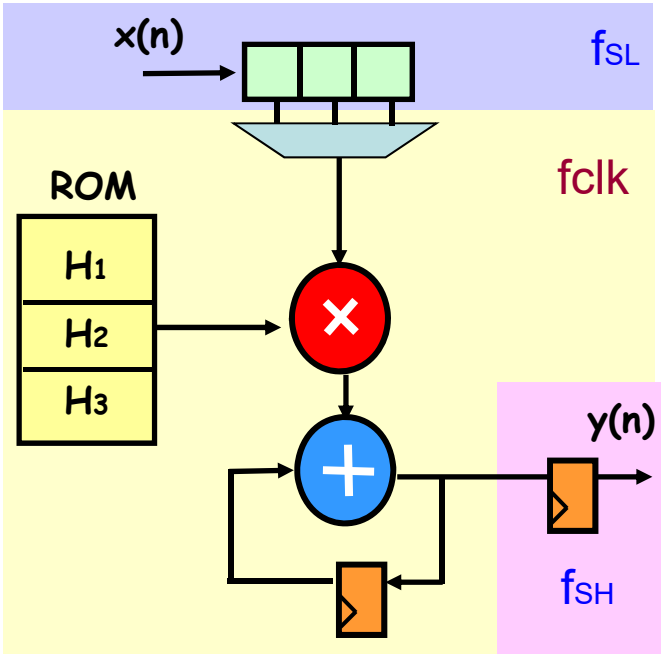
# Arquitecturas polifásicas: Interpolación

## Caso: $f_{clk}/f_{SH} \geq M/R$

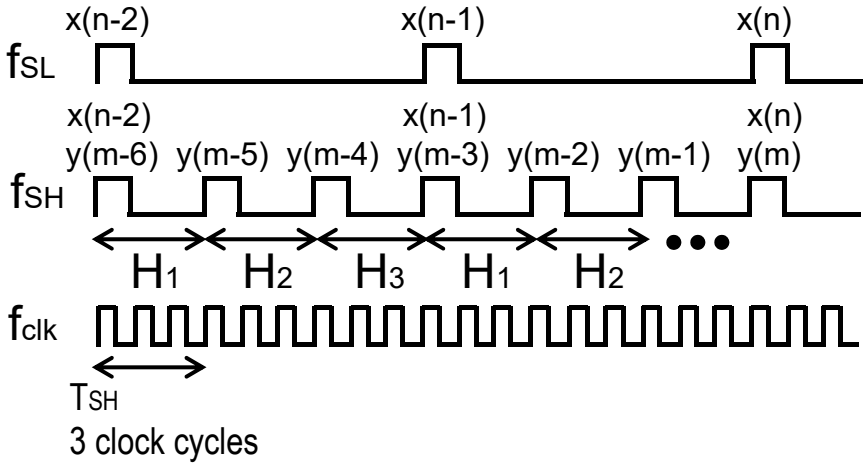
Filtro interpolador por 3 polifásico de 9 etapas

$$y(n) = \sum_{i=0}^{M-1} h_i x(n-i)$$




ROM
- $H_1$: $\{h_0, h_3, h_6\}$
- $H_2$: $\{h_1, h_4, h_7\}$
- $H_3$: $\{h_2, h_5, h_8\}$

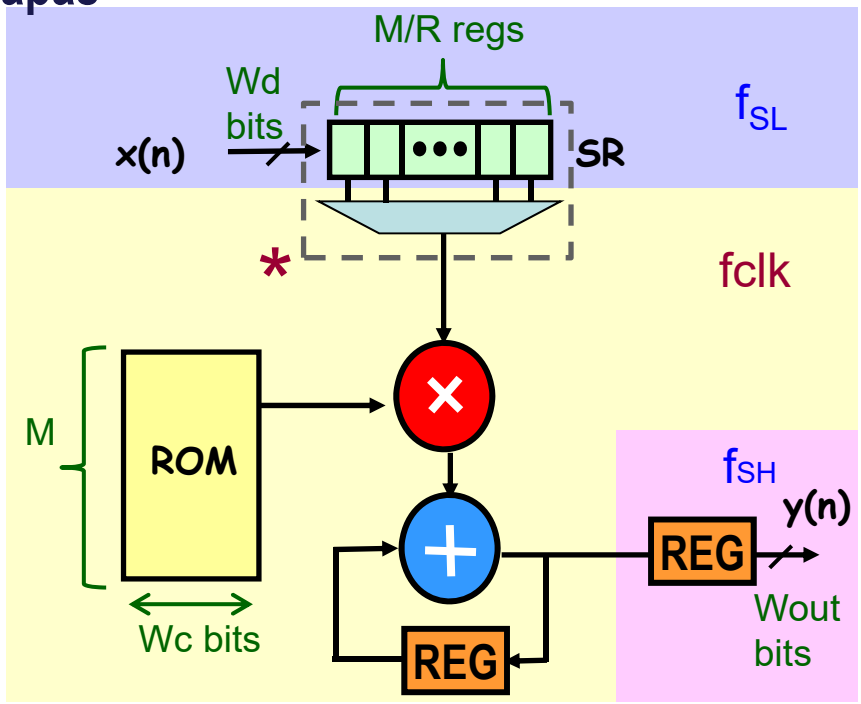



3 mults cada $T_{SH}$

# Arquitecturas polifásicas: Interpolación

## Caso: $f_{clk}/f_{SH} \geq M/R$

**Filtro polifásico interpolador por R de M etapas**




ROM = [h(1:R:M) h(2:R:M) … ]

$$y(n) = \sum_{i=0}^{M-1} h_i x(n-i)$$

## Recursos HW:

- M/R registros of Wd bits
- ROM: MxWc bits
- Multiplicador: WdxWc bits
- Sumador: Wd+Wc+g bits
- Registros (acc.): Wd+Wc+g bits (g: bits de guarda del acum.)
- Registro de salida: Wout bits

## Throughput (max):

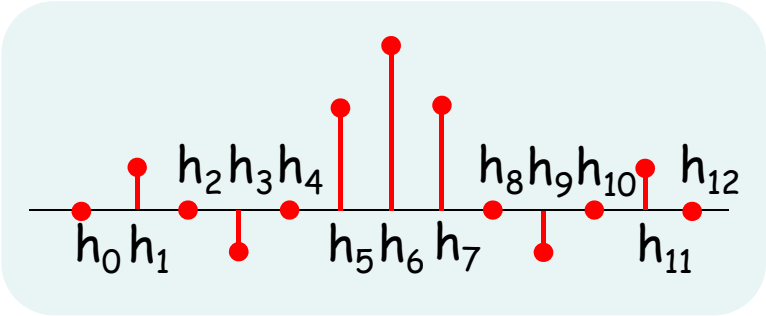
- $T_{out}$ = fclk/(M/R)
- $T_{in}$ = fclk/M

* SR se puede implementar con RAM si el orden del filtro es elevado

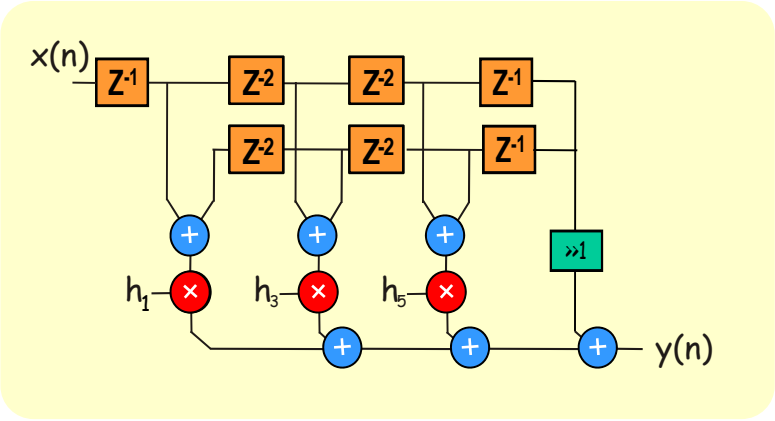
# Aprovechando las propiedades del algoritmo

Se pueden aprovechar las **simetrías** y los **coeficientes nulos**

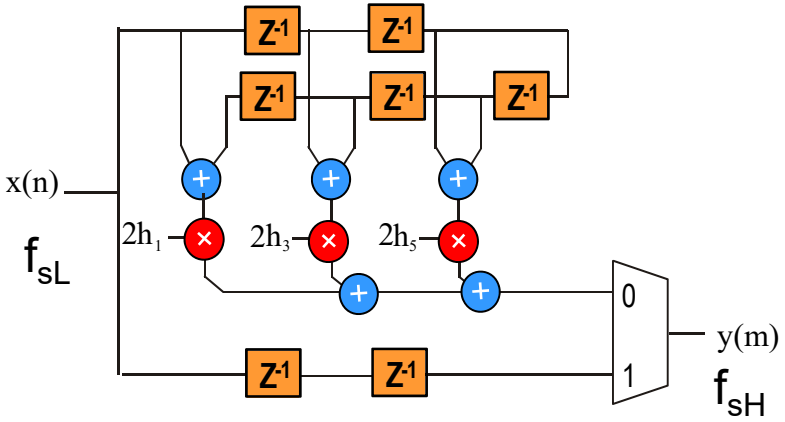
## Filtro Media-banda



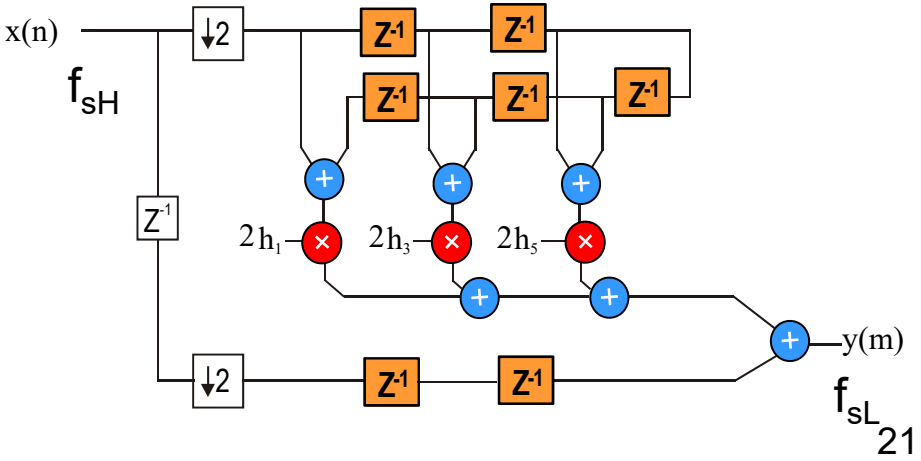

$h_1 = h_{11}$
$h_3 = h_9$
$h_5 = h_7$
$h_6 = 0.5$
$h_0 = h_{12} = 0$
$h_2 = h_{10} = 0$
$h_4 = h_8 = 0$




## Interpolador x2 media-banda




## Diezmador x2 media-banda



# Conclusiones

- Implementación algoritmos multitasa utilizando arquitecturas paralelas y secuenciales


UNIVERSITAT POLITÈCNICA DE VALÈNCIA

# Aplicación de las Arquitecturas Paralelas y Secuenciales al Procesado Multitasa

## Procesado Digital de la Señal en FPGA

**Máster Universitario en Ingeniería de Sistemas Electrónicos**