

MyC – TP Final

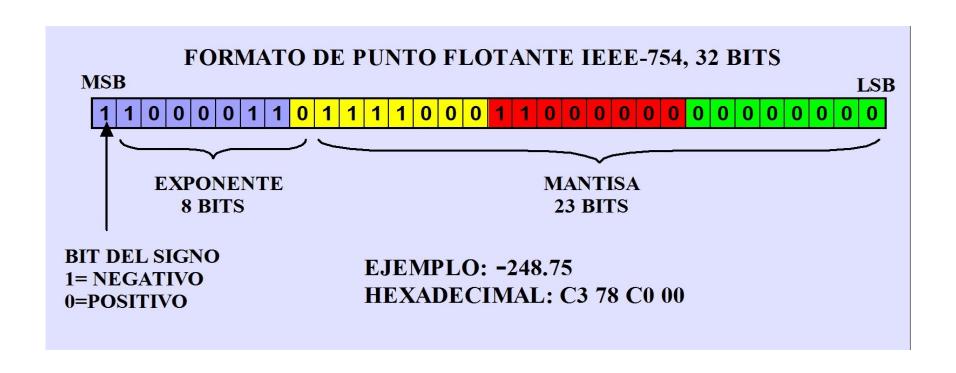
Curso de Especialización en Sistemas embebidos

Trabajo: Multiplicador de Punto Flotante

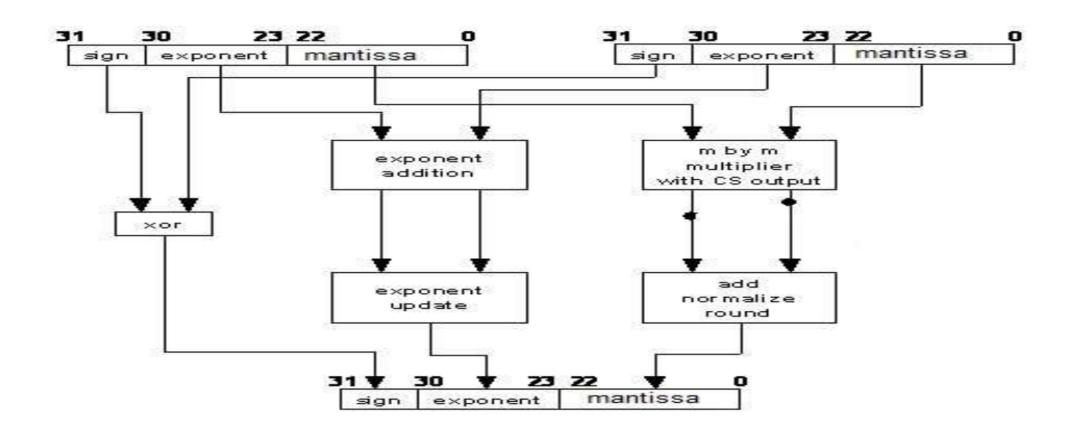
Docente: Ing. Nicolas Álvarez

Alumno: Ing. Jhonny Velasco

Introducción – Multiplicador Punto Flotante



Introducción – Multiplicador Punto Flotante



Ejemplo

```
n1 = 2.5
```

n2 = -10.5

r = -26.25

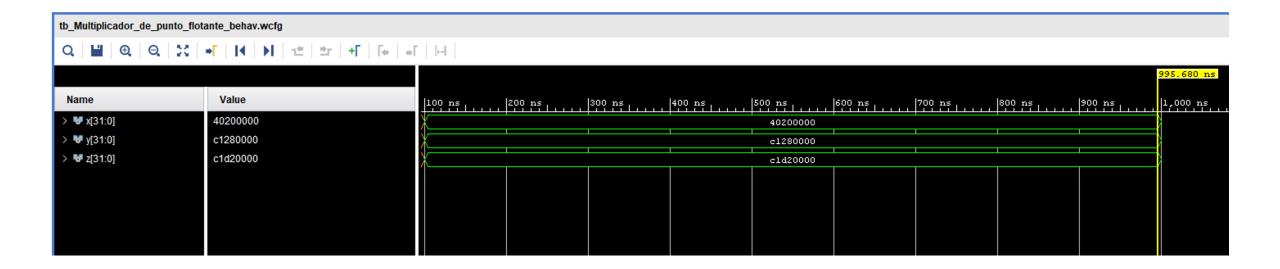
N	lúmero a convertir:	-26.25		Clear	Con Redondeo	Sin Redondeo
Re	esultados:					
Va	alor decimal del núm	nero introducido:	-26.25			

Representación binaria en simple precisión (32 bits):

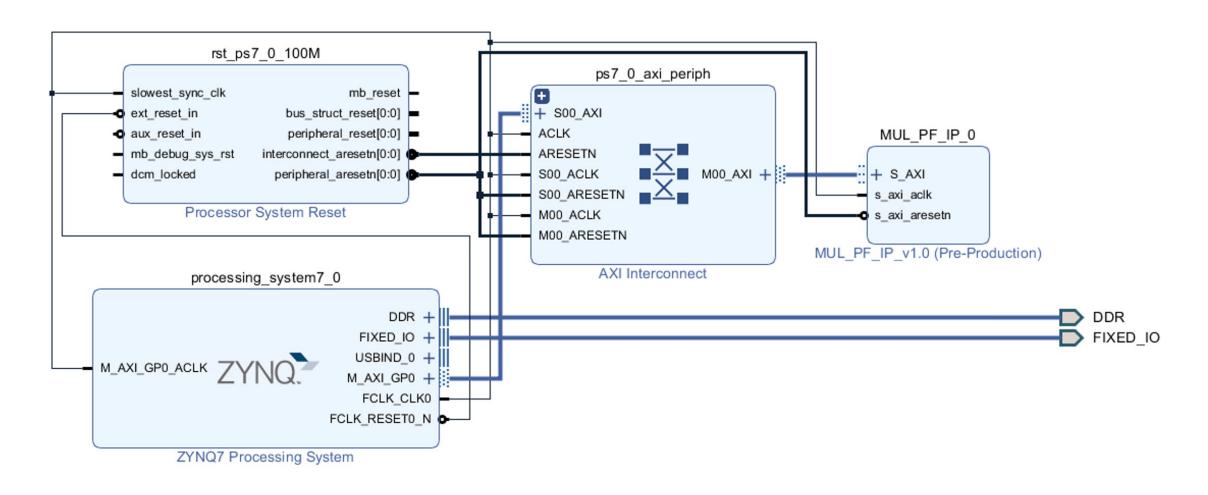


Número a co	onvertir:	2.5		Clea	ar	Con Redondeo	Sin Redondeo				
Resultados:											
/alor decimal del número introducido: 2.5											
Representación binaria en simple precisión (32 bits):											
Tipo: normal											
					Bits 22 - 0 Significando						
0	10000000 Valor decimal del exponente y su equivalente 128 - 127 = 1			1.0100000000000000000000000000Valor decimal equivalente							
(0=+/1=-)											
					1.2500000						
Valor hexadecimal: 40200000 Valor decimal: 2.5000000											
Número a convertir: -10.5 Cle						ar Con Redondeo Sin Redondeo					
Resultados:											
Valor do simo	يئيم إمام إم	mara intradi	usida. 40 F								
Valor decima	ai dei nui	nero introdi	-10.5								
Representación binaria en simple precisión (32 bits):											
Tipo: normal											
Bit 31						Bits 22 - 0					
Signo	Exponente 10000010 Valor decimal del exponente y su equiva				Significando 1.010100000000000000000000000000000000						
1 (0=+/1=-)			alente	Valor decimal equivalente							
(3 1, 1)	130 - 127 = 3				1.3125000						
Valor hexadecimal: C1280000 Valor decimal: -10.500000											

Simulación



Block Design



Código C

```
xil printf("Bienvenidos!\r\n");
while(1)
   data rec=XUartPs Recv(&uart ps,&RecvBuffer,RECV BUFFER SIZE);
    if(data rec!=0)
       data=RecvBuffer;
       xil printf("Operacion: %i ", RecvBuffer);
   switch(data){
    case 49: // data es igual a #1
       xil printf("Primera multiplicacion: \r\n");
       MUL_PF_IP_mWriteReg(XPAR_MUL_PF_IP_0_S_AXI_BASEADDR, MUL_PF_IP_S_AXI_SLV_REG0_OFFSET, opA1);
       MUL_PF_IP_mWriteReg(XPAR_MUL_PF_IP_0 S_AXI_BASEADDR, MUL_PF_IP_S_AXI_SLV_REG1_OFFSET, opB1);
       res1 = (float)MUL PF IP mReadReg(XPAR MUL PF IP 0 S AXI BASEADDR, MUL PF IP S AXI SLV REG2 OFFSET);
       xil printf("Multiplicacion: %f * %f = %f\r\n", opA1, opB1, res1);
       break:
    case 50: // data es igual a #2
       xil printf("Segunda multiplicacion: \r\n");
       MUL PF IP mWriteReg(XPAR_MUL PF IP_0 S_AXI_BASEADDR, MUL_PF_IP_S_AXI_SLV_REGO_OFFSET, opA2);
       MUL PF IP mWriteReg(XPAR MUL PF IP 0 S AXI BASEADDR, MUL PF IP S AXI SLV REG1 OFFSET, opB2);
       res2 = (float) MUL PF IP mReadReg(XPAR MUL PF IP 0 S AXI BASEADDR, MUL PF IP S AXI SLV REG2 OFFSET);
        xil printf("Multiplicacion: %f * %f = %f\r\n", opA2, opB2, res2);
       break;
    case 51: // data es igual a #3
       xil printf("Tercera multiplicacion: \r\n");
       MUL_PF_IP_mWriteReg(XPAR_MUL_PF_IP_0_S_AXI_BASEADDR, MUL_PF_IP_S_AXI_SLV_REG0_OFFSET, opA3);
       MUL PF IP mWriteReg(XPAR MUL PF IP 0 S AXI BASEADDR, MUL PF IP S AXI SLV REG1 OFFSET, opB3);
       res3 = (float)MUL PF IP mReadReg(XPAR MUL PF IP 0 S AXI BASEADDR, MUL PF IP S AXI SLV REG2 OFFSET);
       xil printf("Multiplicacion: %d * %d = %d\r\n", opA3, opB3, res3);
       break;
```

Prueba Real (SoC)

```
[artyz7-user00@lse-server-pc ~]$ minicom -D /dev/ArtyZ7-Board00-003017A4C81C -b
115200
Welcome to minicom 2.8
OPTIONS: I18n
Compiled on May 26 2022, 00:00:00.
Port /dev/ArtyZ7-Board00-003017A4C81C, 11:29:52
Press CTRL-A Z for help on special keys
Bienvenidos!
Operacion 49.
Primera multiplicacion:
```

^{*} Está fallando en esta parte, no ejecuta la operación y se queda colgado.

