

MyC – TP Final

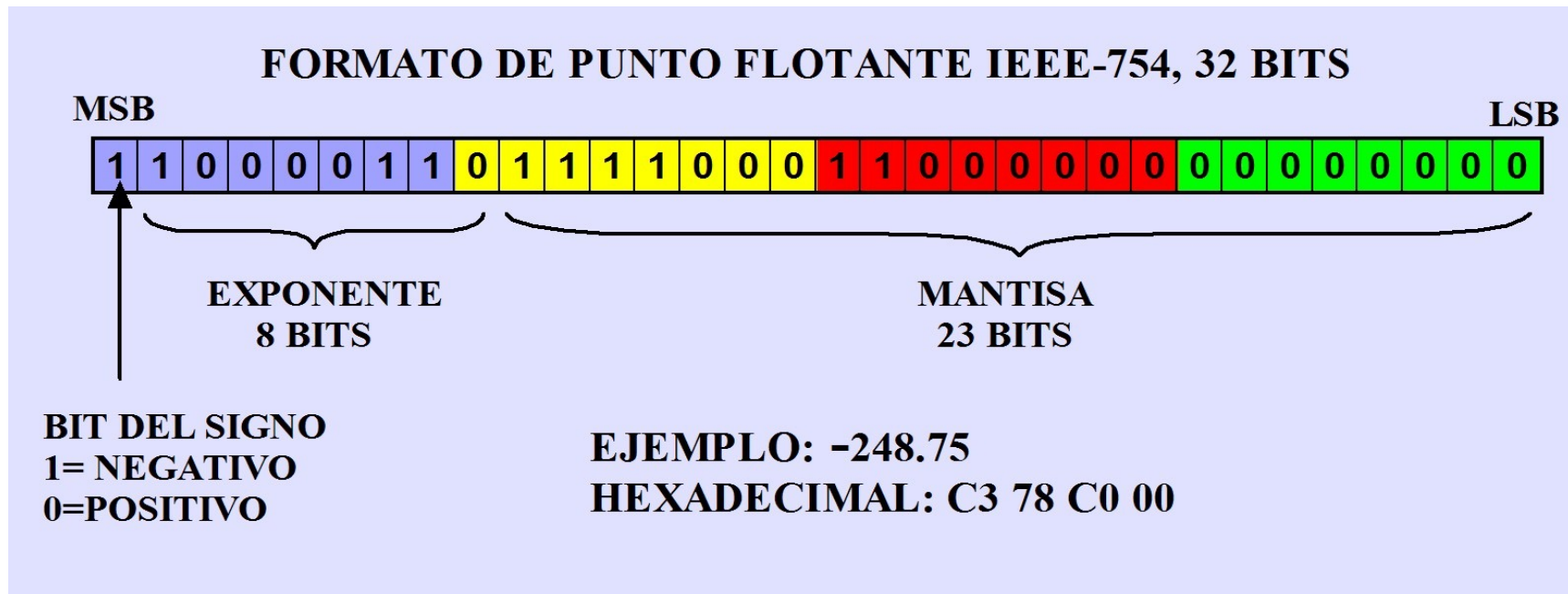
Curso de Especialización en Sistemas embebidos

Trabajo : Multiplicador de Punto Flotante

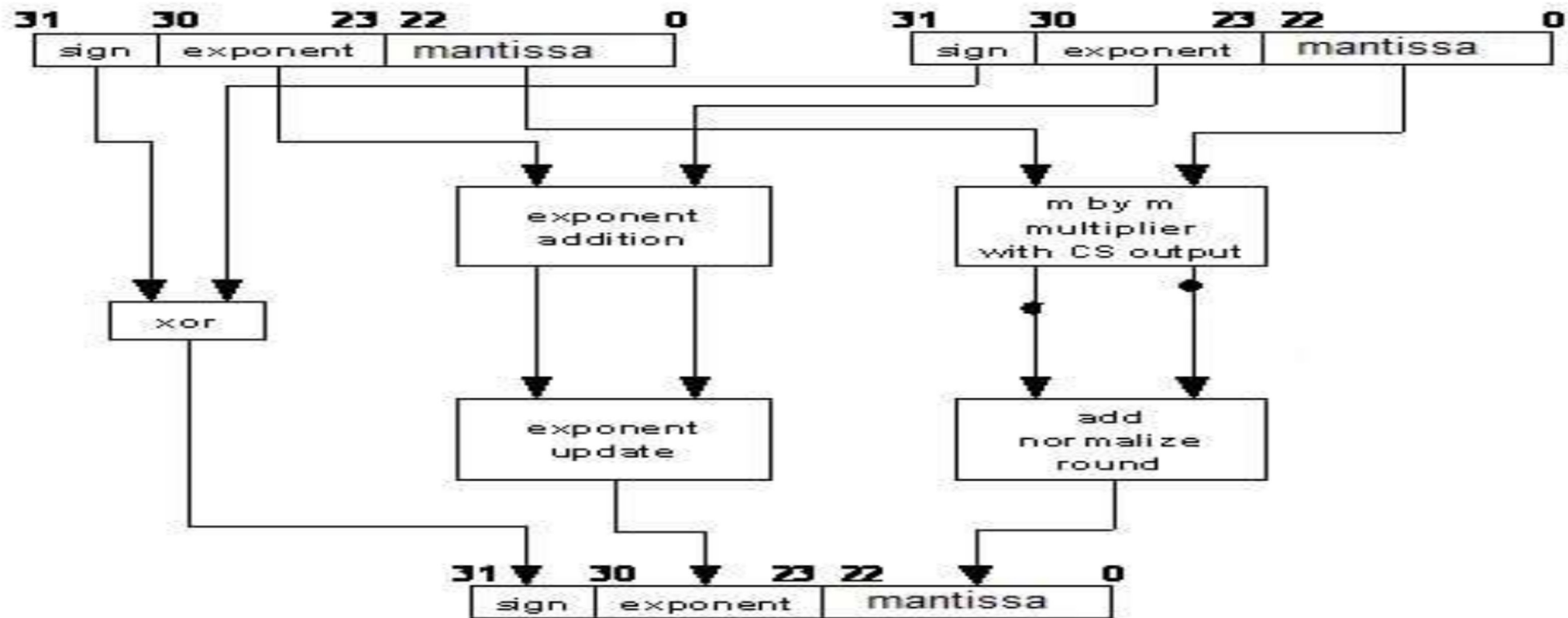
Docente: Ing. Nicolas Álvarez

Alumno : Ing. Jhonny Velasco

Introducción – Multiplicador Punto Flotante



Introducción – Multiplicador Punto Flotante



Ejemplo

$n1 = 2.5$
 $n2 = -10.5$
 $r = -26.25$

Número a convertir:

Clear

Con Redondeo

Sin Redondeo

Resultados:

Valor decimal del número introducido:

Representación binaria en simple precisión (32 bits):

Tipo:

Bit 31 Signo <input type="text" value="1"/> (0=+/1=-)	Bits 30 - 23 Exponente <input type="text" value="10000011"/> Valor decimal del exponente y su equivalente <input type="text" value="131"/> - 127 = <input type="text" value="4"/>	Bits 22 - 0 Significante <input type="text" value="1.101001000000000000000000"/> Valor decimal equivalente <input type="text" value="1.6406250"/>
-----------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Valor hexadecimal: Valor decimal:

Número a convertir:

Clear

Con Redondeo

Sin Redondeo

Resultados:

Valor decimal del número introducido:

Representación binaria en simple precisión (32 bits):

Tipo:

Bit 31 Signo <input type="text" value="0"/> (0=+/1=-)	Bits 30 - 23 Exponente <input type="text" value="10000000"/> Valor decimal del exponente y su equivalente <input type="text" value="128"/> - 127 = <input type="text" value="1"/>	Bits 22 - 0 Significante <input type="text" value="1.010000000000000000000000"/> Valor decimal equivalente <input type="text" value="1.2500000"/>
-----------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Valor hexadecimal: Valor decimal:

Número a convertir:

Clear

Con Redondeo

Sin Redondeo

Resultados:

Valor decimal del número introducido:

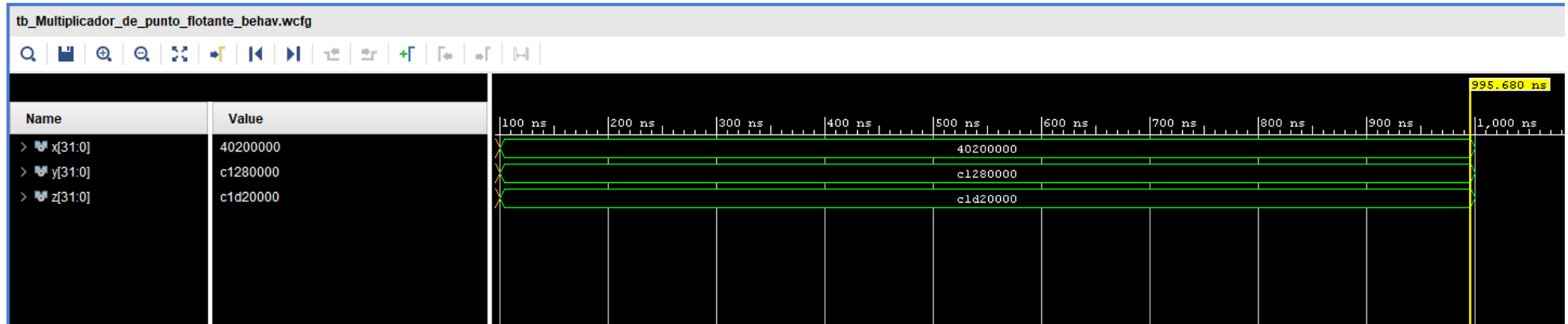
Representación binaria en simple precisión (32 bits):

Tipo:

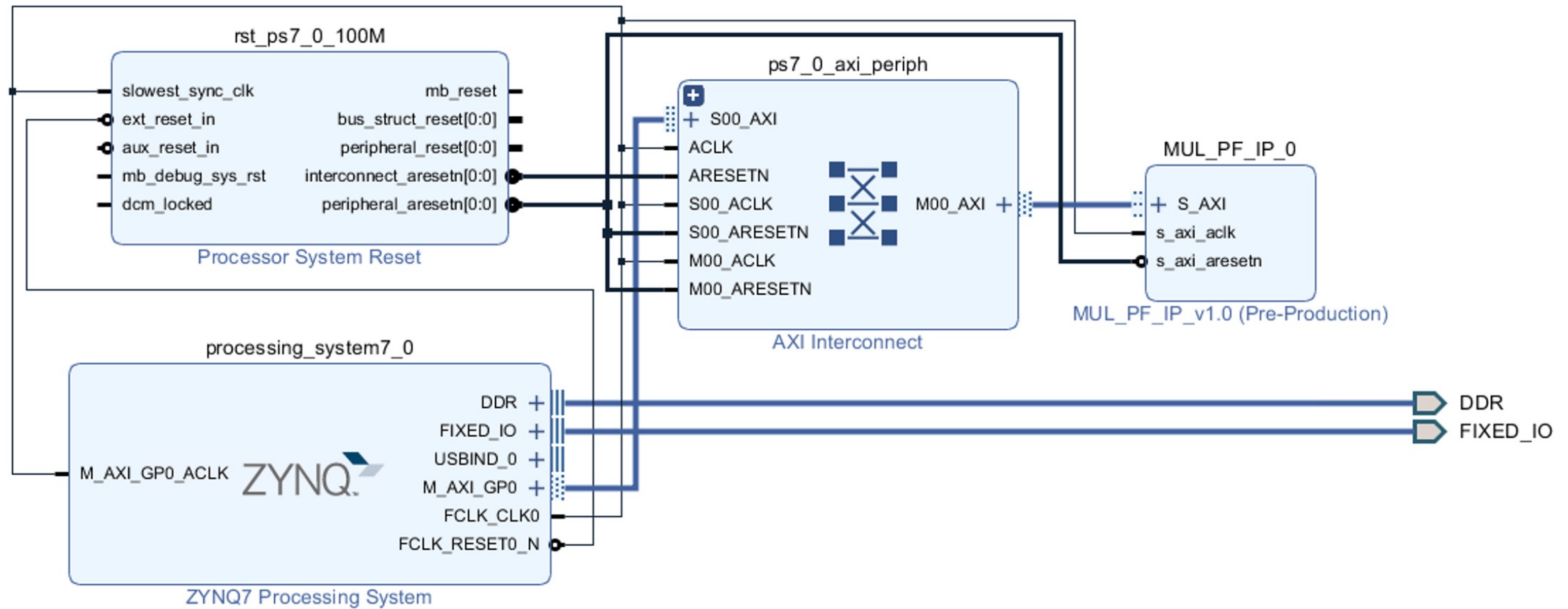
Bit 31 Signo <input type="text" value="1"/> (0=+/1=-)	Bits 30 - 23 Exponente <input type="text" value="10000010"/> Valor decimal del exponente y su equivalente <input type="text" value="130"/> - 127 = <input type="text" value="3"/>	Bits 22 - 0 Significante <input type="text" value="1.010100000000000000000000"/> Valor decimal equivalente <input type="text" value="1.3125000"/>
-----------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Valor hexadecimal: Valor decimal:

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_REG0_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.

