

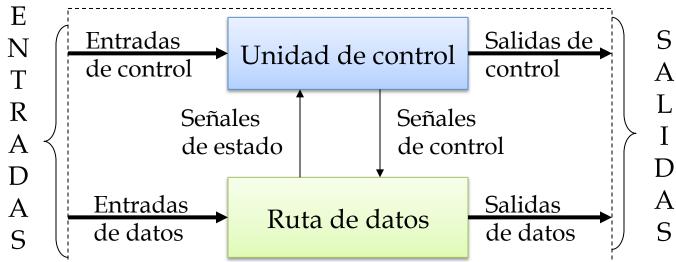
#### Práctica 6

Multiplicación de números en formato IEEE 754

Última actualización: 20/12/2017

### Especificaciones

- T E
- Diseñar un sistema algorítmico que implemente la multiplicación "simplificada" de dos números en formato simple IEEE 754
  - Números normalizados
  - Operandos de entrada (op1 y op2) nunca un NaN
- El diseño se implementará como un sistema algorítmico



#### **Puertos**

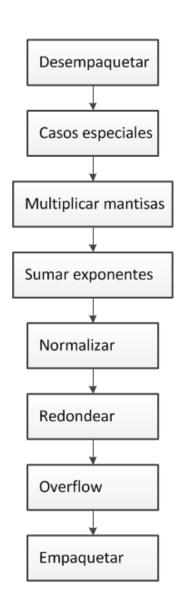


Definición entidad

```
entity mult_ieee754 is
  port (
    clk : in std_logic;
    rst : in std_logic;
    -- operandos entrada/salida
    op1 : in std_logic_vector(31 downto 0);
    op2 : in std_logic_vector(31 downto 0);
    res : out std_logic_vector(31 downto 0);
    -- control
    ini : in std_logic;
    fin : out std_logic);
end entity mult_ieee754;
```

- Funcionamiento síncrono. Todos los registros activos por flanco de subida.
- La señal de reloj será c1k.
- La señal de reset, rst, asíncrona activa a nivel alto.
- Datos de entrada (op1 y op2) y salida (res) registrados.

1. Extraer signo, exponente y mantisa de cada operando





# Desempaquetar Casos especiales Multiplicar mantisas Sumar exponentes Normalizar Redondear Overflow Empaquetar

#### Algoritmo

1. Extraer signo, exponente y mantisa de cada operando

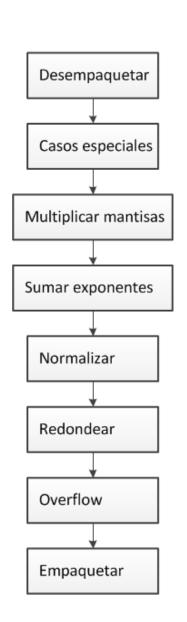


op1	op2	res
0	any	0
∞	0	NaN
∞	any	∞
∞	∞	∞

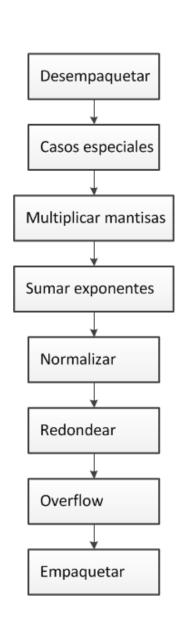
En la tabla anterior:

- Los valores de las columnas op1 y op2 son intercambiables
- Hay que tener en cuenta los signos





- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- Cada mantisa es un número de 24 bits: a los 23 bits de la mantisa hay que concatenar a su izquierda un 1
- El resultado es un número de 48 bits
- El punto decimal estará entre los bits 45 y 46 (siendo el 0 el menos significativo)



- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- 4. Los exponentes se suman como números en binario puro
- El resultado es un número de 9 bits (8+1), al que hay que restar un exceso a 127

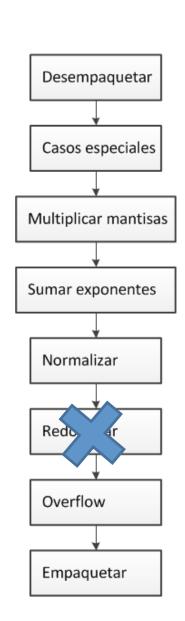
7

# Desempaquetar Casos especiales Multiplicar mantisas Sumar exponentes Normalizar Redondear Overflow Empaquetar

- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- 4. Los exponentes se suman como números en binario puro
- 5. Se desplaza el punto decimal de la mantisa
- Dicho punto debe quedar a la derecha del primer uno más significativo.
- Por cada desplazamiento a izquierda (solo hará falta un desplazamiento en el caso peor) se debe incrementar el exponente en +1

# Desempaquetar Casos especiales Multiplicar mantisas Sumar exponentes Normalizar Redo Overflow Empaquetar

- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- 4. Los exponentes se suman como números en binario puro
- 5. Se desplaza el punto decimal de la mantisa
- 6. Nuestro multiplicador simplificado no redondea



- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- 4. Los exponentes se suman como números en binario puro
- 5. Se desplaza el punto decimal de la mantisa
- 6. Nuestro multiplicador simplificado no redondea
- 7. Si el exponente (de 9 bits) fue 255 o mayor, overflow
  - El campo mantisa se debe poner a 0 y el exponente, a 255 para representar el infinito
  - Si el exponente (de 9 bits) fue negativo, underflow
  - El campo mantisa y exponente se deben poner a 0

- 1. Extraer signo, exponente y mantisa de cada operando
- 2. Gestión operandos entrada igual a 0 o infinito
- 3. Las mantisas se multiplican como numero en binario puro
- 4. Los exponentes se suman como números en binario puro
- 5. Se desplaza el punto decimal de la mantisa
- 6. Nuestro multiplicador simplificado no redondea
- 7. Si el exponente es > 255 o < 0
- 8. Se concatena el bit de signo, 8 bits menos significativos del exponente y los 23 bits más significativos de la mantisa excluido el primer 1



## Ejemplo 1

```
Desempaquetar
               op1 = 110010001010001001000001101011
                                                            v1 = -332419.34375
               op2 = 00010001010001000100011000110
                                                            v2 = 1.550382E-28
 Casos especiales
               exp1 = 10010001 mantisa1 = 01000100101000001101011
                                mantisa2 = 10001001000100011000110
               exp2 = 00100010
Multiplicar mantisas
               m1 = 101000100101000001101011
                                                   Localización punto decimal
               m2 = 110001001000100011<u>000110</u>
               Sumar exponentes
              exp1 = 10010001
 Normalizar
              exp2 = 00100010
                                     Restar exceso a 127
              exp_parcial = 010110011
                                                      exp_res= 000110100
 Redo
                                                No ha sido necesario normalizar.
               m_res = 11110010011100010010100
                                                Quitado el primer 1 de m res
              exp res = 000110100
 Overflow
                               No hay overflow ni underflow
               res=1 001101001111001001110001001001 res = -5.1537697E-23
 Empaquetar
                      Ajustar el bit de signo
```

## Ejemplo 2

