Laboratorio 1:

Lógica combinacional

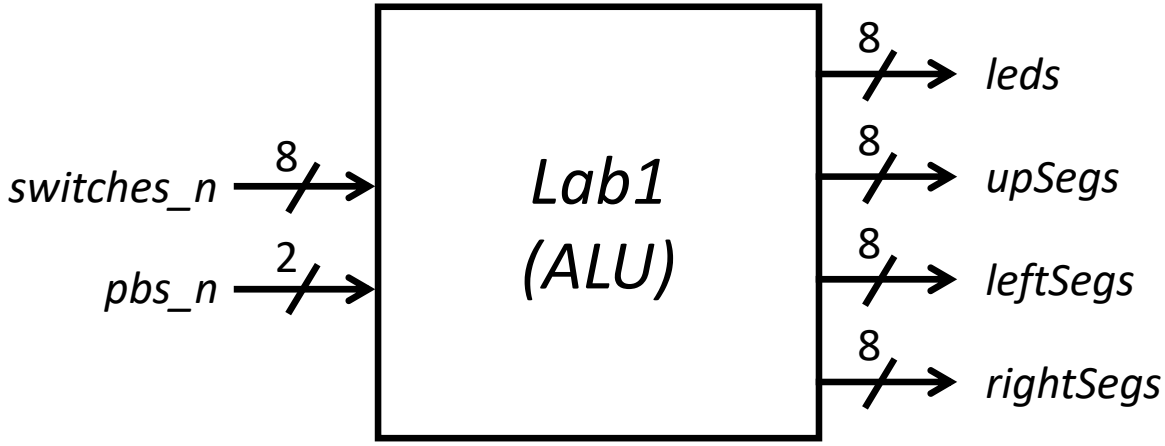
aritmética y acceso a dispositivos elementales de E/S

Diseño automático de sistemas

José Manuel Mendías Cuadros
Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Presentación

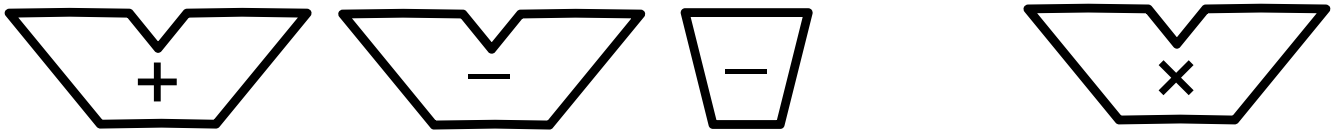
- Diseñar una ALU combinacional de 8 bits capaz de sumar, restar, multiplicar y calcular el opuesto.
- Tomará los operandos y visualizará los resultados del siguiente modo:
  - El operando derecho lo tomará de los 4 switches menos significativos de la placa XST.
  - El operando izquierdo lo tomará de los 4 switches más significativos de la placa XST.
  - El código de operación lo tomará de los 2 pulsadores de la placa XSA-3S:
    - "00" : Suma, "01" : Resta, "10" : Opuesto del operando derecho, "11" : Multiplica
  - El resultado se visualizará en hexadecimal en los displays 7-segs de la placa XST.
  - La código de operación se visualizará en decimal en el display 7-segs de la placa XSA-3S.
  - El estado de los switches se visualizará en binario en los leds de la placa XST.



J.M. Mendías
2018

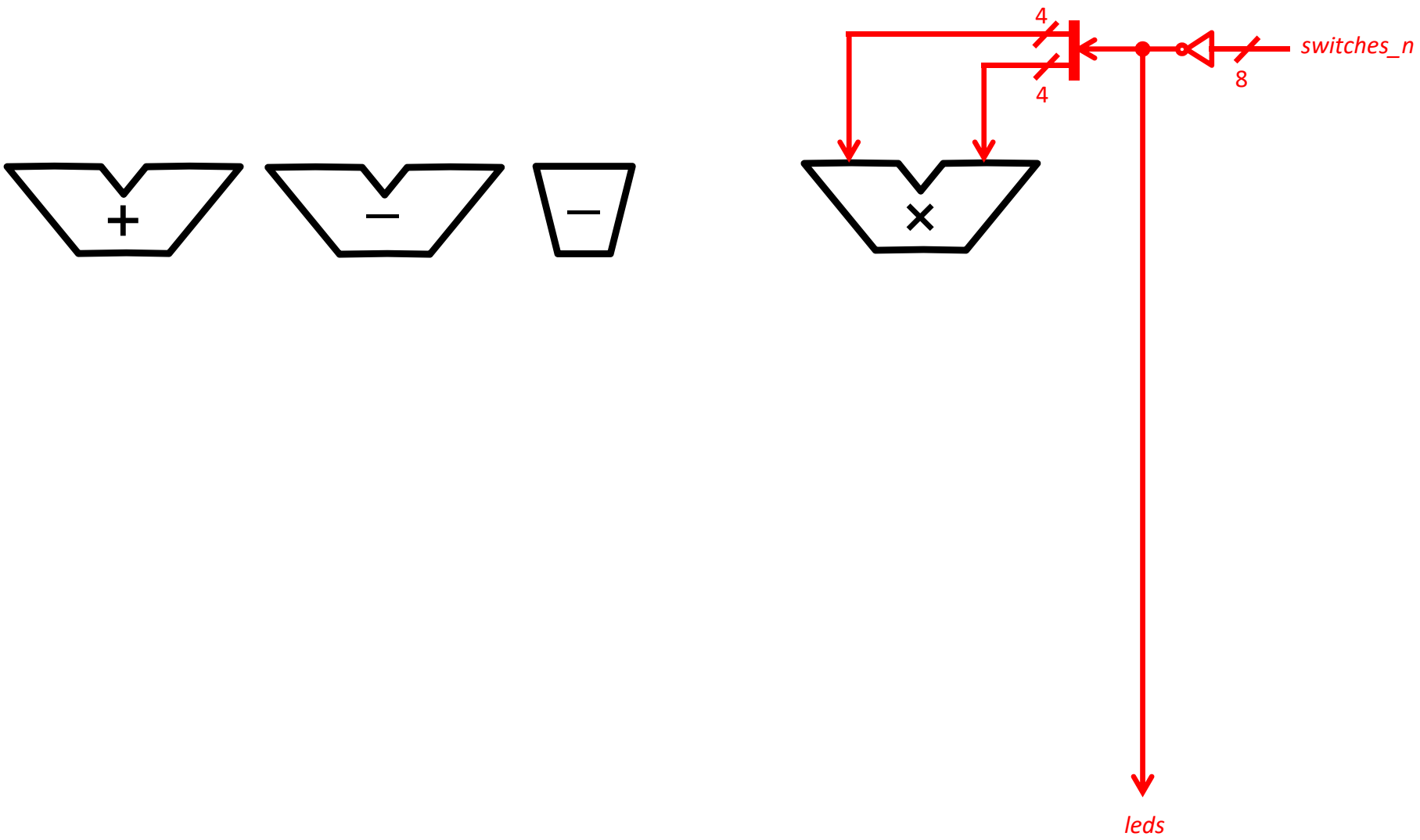
Diseño principal

diagrama RTL



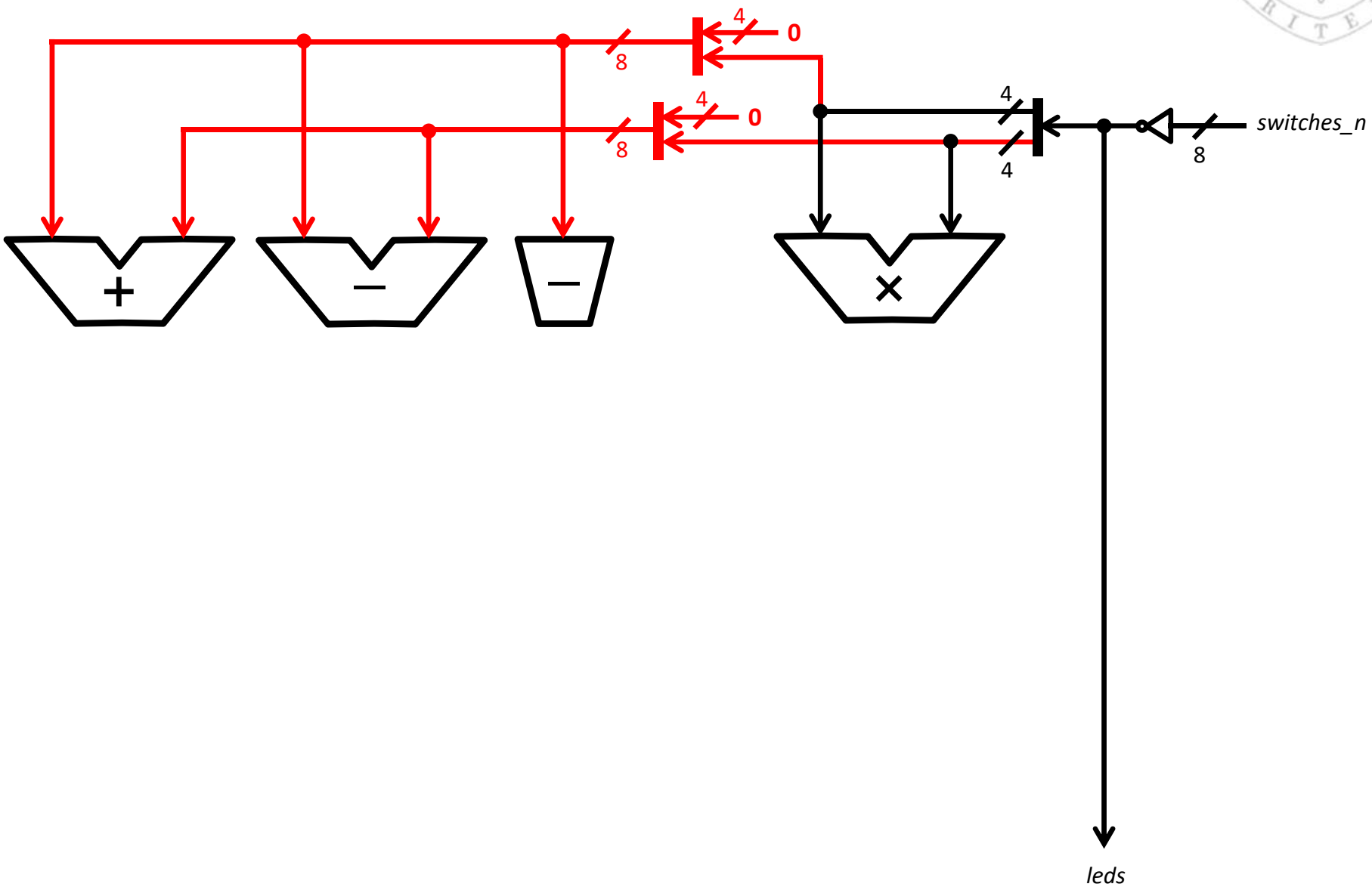
Diseño principal

diagrama RTL



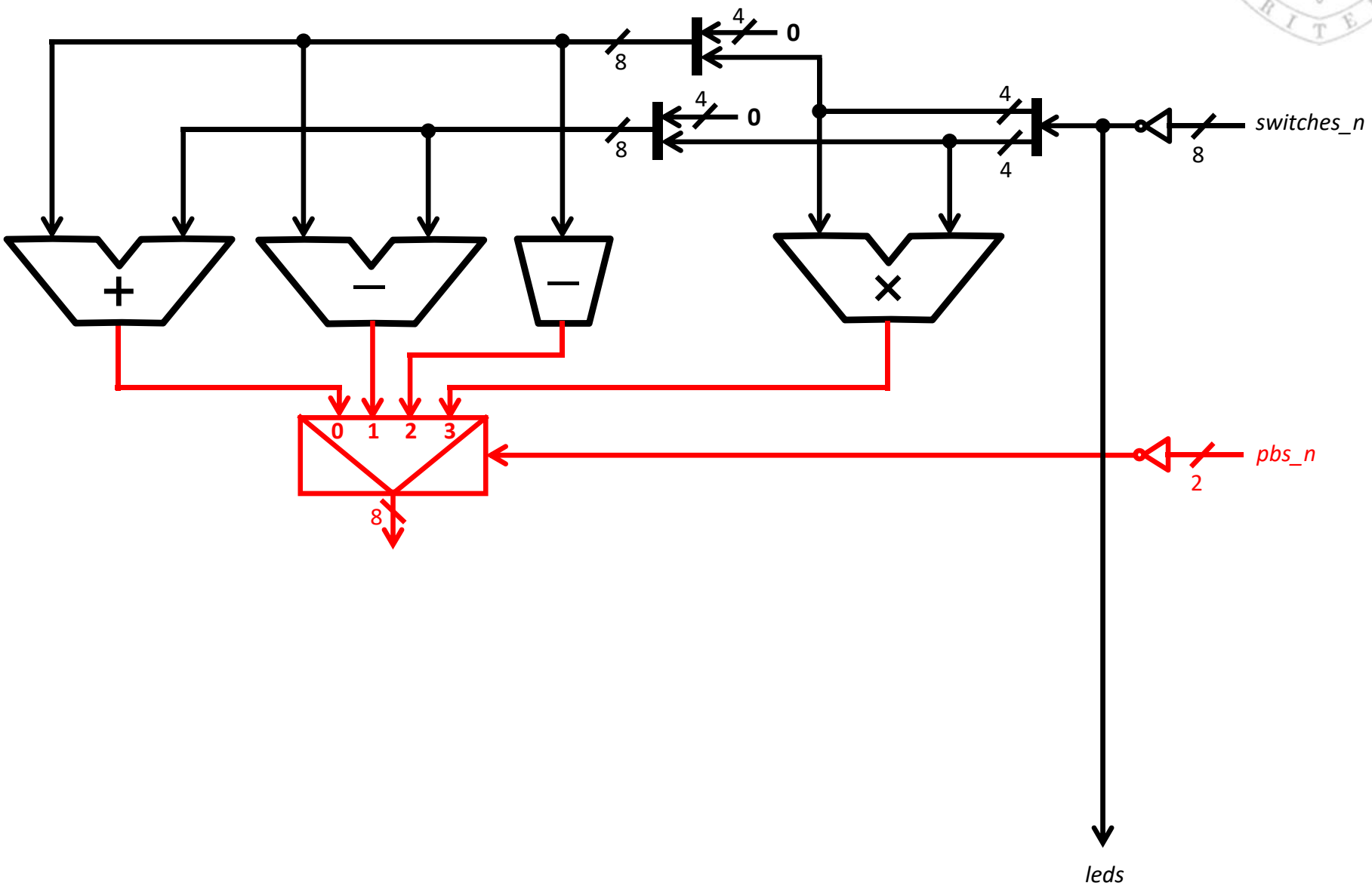
Diseño principal

diagrama RTL



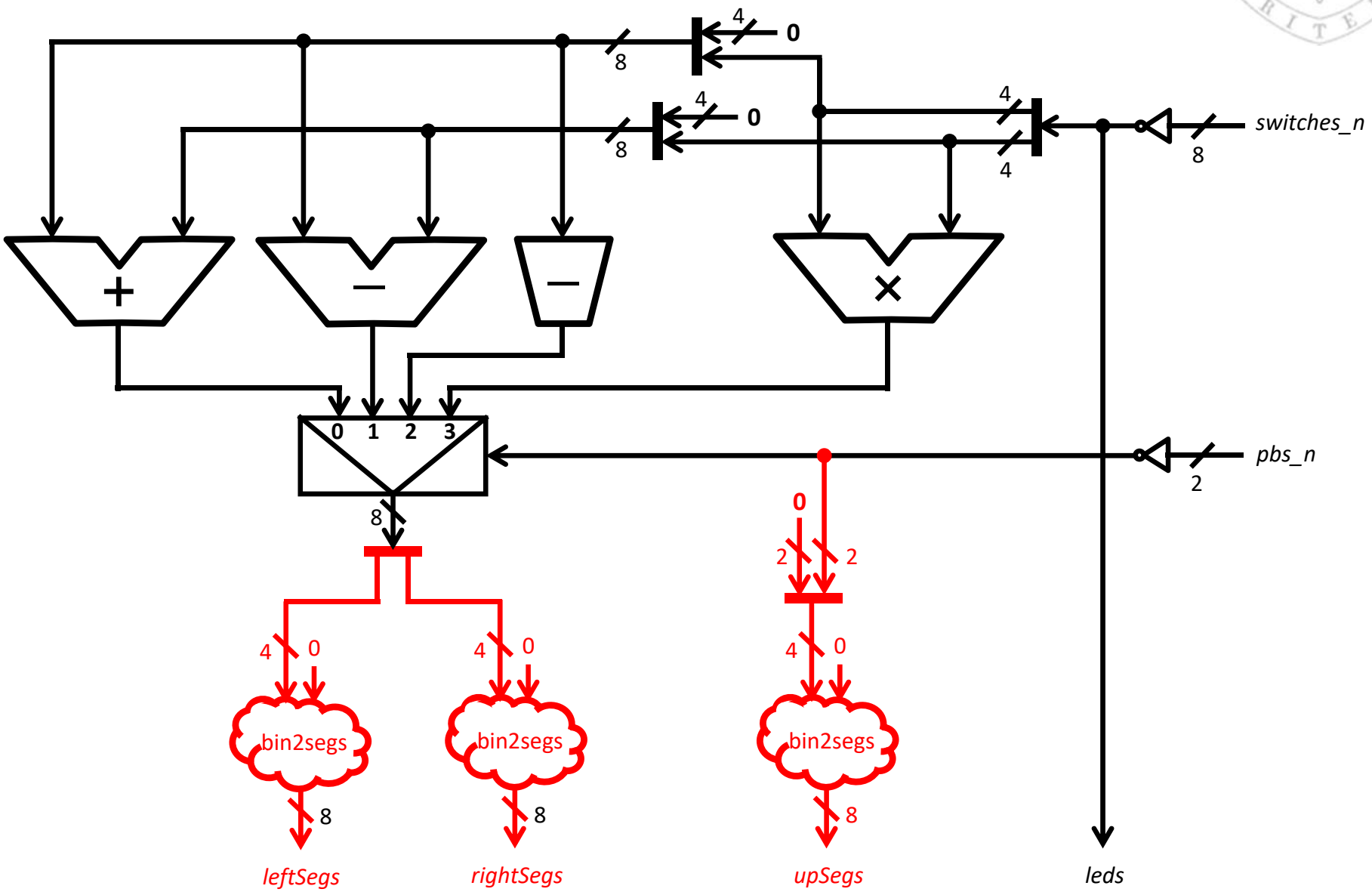
Diseño principal

diagrama RTL



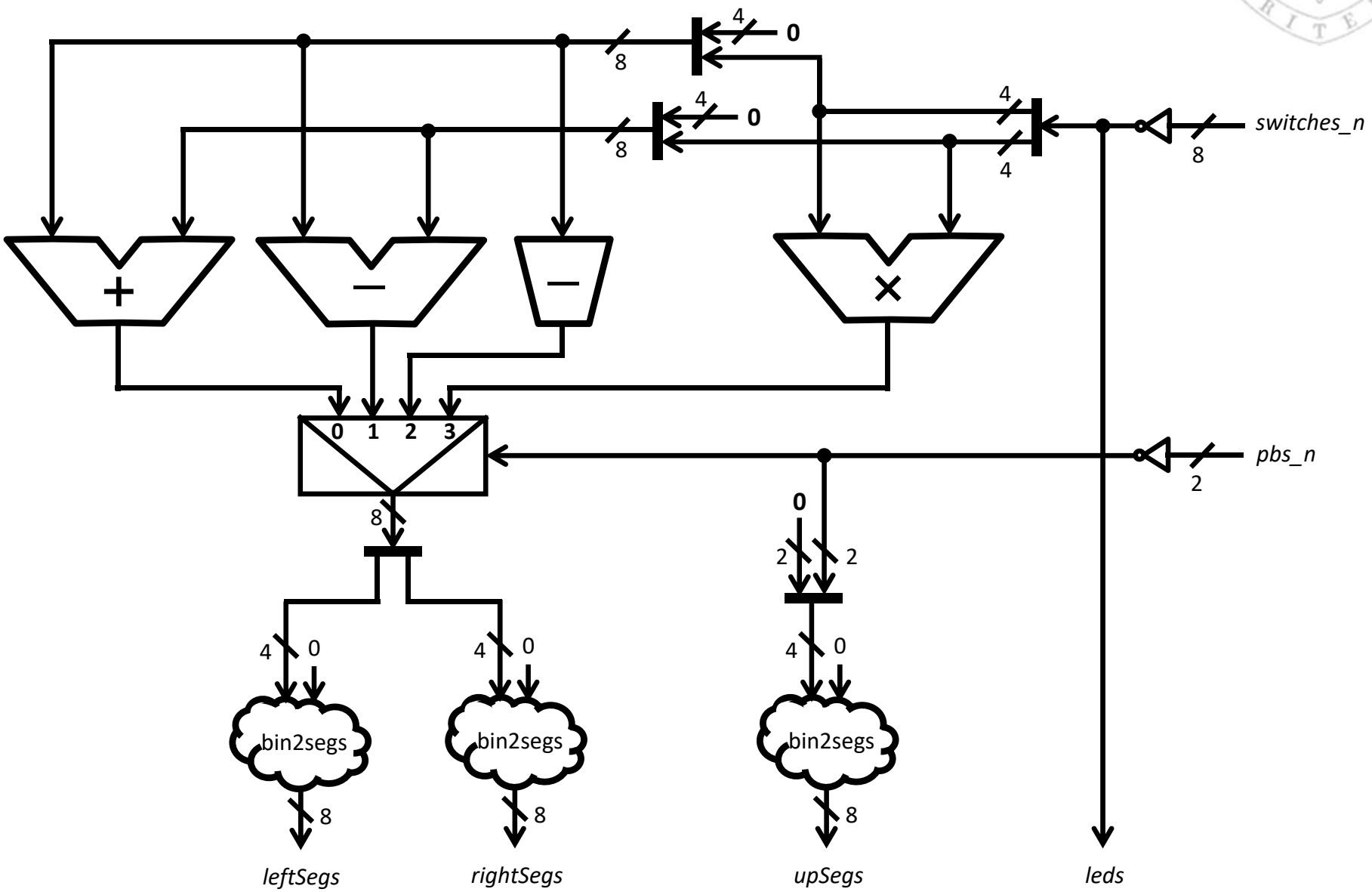
Diseño principal

diagrama RTL



Diseño principal

diagrama RTL



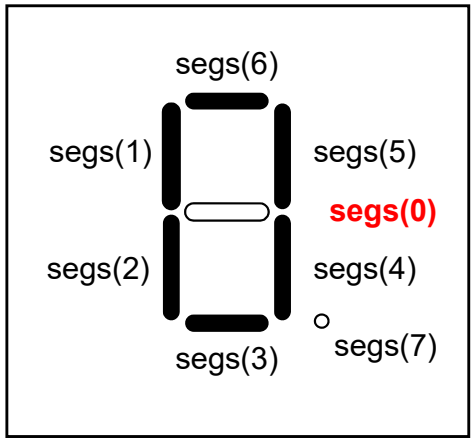
Módulo conversor 7-segmentos

bin2segs.vhd

```
library ieee;
use ieee.std_logic_1164.all;

entity bin2segs is
  port (
    -- host side
    bin  : in  std_logic_vector(3 downto 0);   -- codigo binario
    dp   : in  std_logic;                      -- punto
    -- leds side
    segs : out std_logic_vector(7 downto 0)    -- codigo 7-segmentos
);
end bin2segs;

architecture syn of bin2segs is
begin
  segs(7) <= ...;
  with bin select
    segs(6 downto 0) <=
      "1111110" when X"0",  ····· visualiza el 0
      "......." when X"1",
         ...
      "......." when others;
end syn;
```



J.M. Mendías
2018

Utilidades y componentes

common.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;

package common is

  constant YES  : std_logic := '1';
  constant NO   : std_logic := '0';
  constant HI   : std_logic := '1';
  constant LO   : std_logic := '0';
  constant ONE  : std_logic := '1';
  constant ZERO : std_logic := '0';

  -- Calcula el logaritmo en base-2 de un numero.
  function log2(v : in natural) return natural;
  -- Selecciona un entero entre dos.
  function int_select(s : in boolean; a : in integer; b : in integer) return integer;

  -- Convierte codigo binario a codigo 7-segmentos
  component bin2segs
    port
    (
      -- host side
      bin  : in  std_logic_vector(3 downto 0);   -- codigo binario
      dp   : in  std_logic;                      -- punto
      -- leds side
      segs : out std_logic_vector(7 downto 0)    -- codigo 7-segmentos
    );
  end component;

end package common;
```

declaración de componentes reusables

J.M. Mendías
2018

Utilidades y componentes

common.vhd

```
package body common is

  function log2(v : in natural) return natural is
    variable n    : natural;
    variable logn : natural;
  begin
    n := 1;
    for i in 0 to 128 loop
      logn := i;
      exit when (n >= v);
      n := n * 2;
    end loop;
    return logn;
  end function log2;

  function int_select(s : in boolean; a : in integer; b : in integer) return integer is
  begin
    if s then
      return a;
    else
      return b;
    end if;
    return a;
  end function int_select;

end package body common;
```

funciones orientadas a trabajar
con argumentos constantes
en expresiones computables

Diseño principal

lab1.vhd

```
library ieee;
use ieee.std_logic_1164.all;

entity lab1 is
  port (
    switches_n : in  std_logic_vector(7 downto 0);
    pbs_n      : in  std_logic_vector(1 downto 0);
    leds       : out std_logic_vector(7 downto 0);
    upSegs     : out std_logic_vector(7 downto 0);
    leftSegs   : out std_logic_vector(7 downto 0);
    rightSegs  : out std_logic_vector(7 downto 0)
  );
end lab1;

library ieee;
use ieee.numeric_std.all;
use work.common.all;

architecture syn of lab1 is

  signal opCode  : std_logic_vector(3 downto 0);
  signal leftOp  : signed(7 downto 0);
  signal rightOp : signed(7 downto 0);
  signal result  : signed(7 downto 0);

begin
  ...
end syn;
```

permite usar las utilidades y las componentes

declaración de los tipos de las señales internas

Diseño principal

lab1.vhd

```
begin

  opCode <= ...;
  leftOP <= ...;
  rightOp <= ...;

  ALU:
  with opCode select
    result <=
      ...;

  leftConverter : bin2segs
  port map ( bin => ..., dp => ..., segs => ... );

  rigthConverter : bin2segs
  port map ( bin => ..., dp => ..., segs => ... );

  upConverter : bin2segs
  port map ( bin => ..., dp => ..., segs => ... );

  leds <= ...;

end syn;
```

adapta puertos de entrada a señales internas

núcleo operativo

usar siempre asociación por nombre

adapta y convierte señales internas a puertos de salida

J.M. Mendías
2018

Tareas

1. Crear el directorio **DAS** y en ella el subdirectorio **common**
2. Crear el proyecto **lab1** en el directorio **DAS**
3. Descargar de la Web los ficheros:
   - **common.vhd** y **bin2segs.vhd** en el directorio **common**
   - **lab1.vhd** y **lab1.ucf** en el directorio **lab1**
4. Completar el código omitido en los ficheros:
   - **bin2segs.vhd** y **lab1.vhd**
5. Añadir al proyecto los ficheros:
   - **common.vhd**, **bin2segs.vhd**, **lab1.vhd** y **lab1.ucf**
6. Sintetizar, implementar y generar el fichero de configuración.
7. Conectar la placa y encenderla.
8. Descargar el fichero **lab1.bit**

Acerca de *Creative Commons*

- Licencia CC (Creative Commons)
  - Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:

Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.

Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: https://creativecommons.org/licenses/by-nc-sa/4.0/