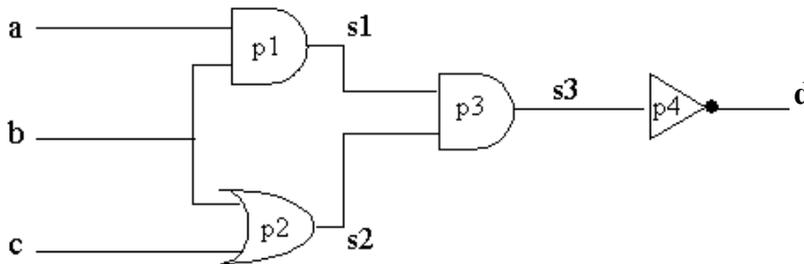


## EXAMEN DE SEPTIEMBRE DE 2.005. I. T. INFORMÁTICA DE GESTIÓN

Duración 2 horas.

### Problema 1)

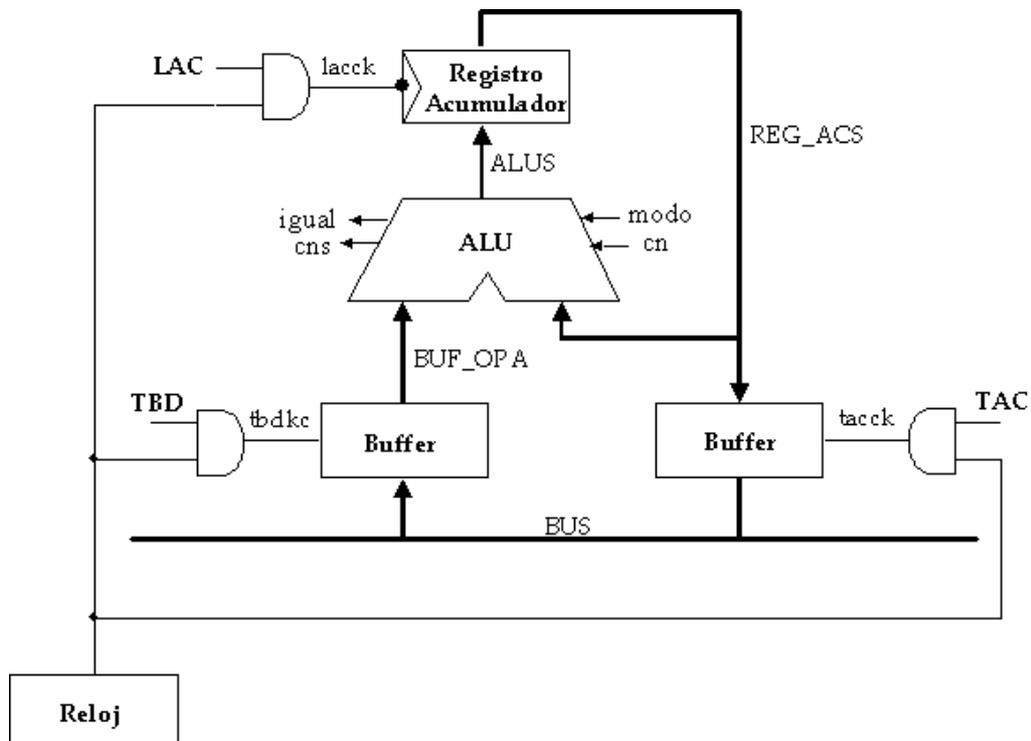
Dado el siguiente circuito combinacional:



- Desarrollar el modelo VHDL de cada una de las puertas lógicas utilizando el estilo **flujo de datos (RTL)**, suponiendo un retardo para cada una de ellas de 2 ns. (1,5 puntos).
- Desarrollar en VHDL la entidad y arquitectura del circuito propuesto utilizando el estilo **comportamental (algorítmico)**. Utilizar un proceso para representar cada una de las puertas del circuito e introducir un retardo a cada una de 2 ns. (1,5 puntos).

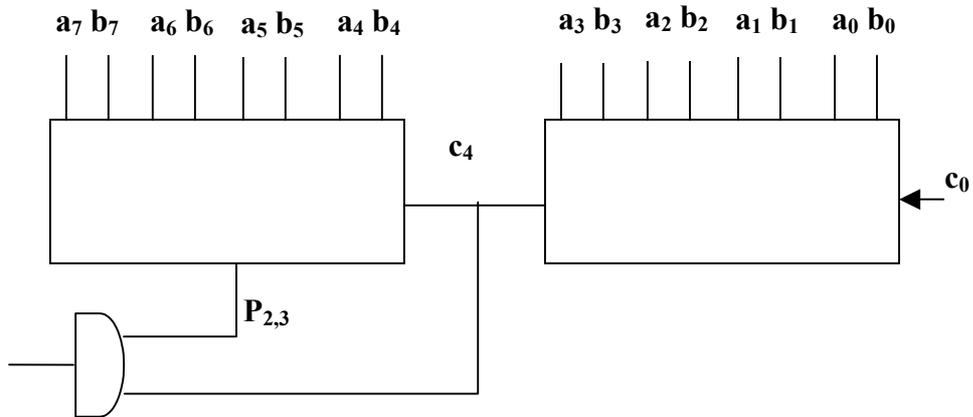
### Problema 2)

En el circuito de la figura se ilustra un circuito compuesto por una unidad aritmético-lógica de 8 bits y los elementos de transferencia necesarios.



- Desarrollar su modelo VHDL **estructural** ( 3.5 puntos)
- Averiguar la máxima frecuencia de reloj a la que es capaz de trabajar el circuito, desde que la señal sale del bus hasta que se carga en el registro acumulador (1.5 puntos).
- Se desea encontrar una alternativa al operador de suma con propagación de acarreo de la ALU para acelerar su funcionamiento, disponiendo únicamente del operador con salto de acarreo de la figura de abajo. Justificar, calculando el tiempo de retardo de este sumador, si esta opción sería la apropiada (2 puntos).

OPERADOR DE SUMA DE 8 BITS CON SALTO DE ACARREO



**MODELOS VHDL DE LOS COMPONENTES:**

```

ENTITY and2 IS
    GENERIC (retardo: TIME := 2 ns);
    PORT (e1, e2: IN BIT; sal: OUT BIT);
END and2;

ARCHITECTURE comportamiento OF and2 IS
    ...
END comportamiento;
    
```

```

USE WORK.arq_pack.ALL;
ENTITY registro8 IS
    GENERIC(retardo_carga:TIME:=10 ns);
    PORT(entrada:IN vector_bus; control:IN BIT:=0';
    salida:OUT vector_bus:=('0','0','0','0', '0','0','0','0'));
END registro8;

ARCHITECTURE comportamiento OF registro4 IS
    ...
END comportamiento;
    
```

```

USE WORK.arq_pack.ALL;
ENTITY buffer8 IS
    GENERIC(retardo_transferencia:TIME:=15 ns;retardo_Z:TIME:=10 ns);
    PORT(entrada:IN vector_bus; control:IN BIT:=0';
    salida:OUT vector_bus);
END buffer8;

ARCHITECTURE comportamiento OF buffer4 IS
    ...
END comportamiento;
    
```

```
USE WORK.arq_pack.ALL;
ENTITY alu_8bits IS
  GENERIC (retardo: TIME:=32 ns);
  PORT (a, b: IN vector_bus;
        op: IN BIT_VECTOR(1 DOWNT0 0);
        modo,cn: IN BIT;
        resultado: OUT vector_bus ;
        cns,igual: OUT BIT);
END alu_8bits;

ARCHITECTURE comportamiento OF alu_8bits IS
...
END comportamiento;
```

```
ENTITY reloj IS
  GENERIC(periodo:TIME:=?? ns);
  PORT(reloj:OUT BIT:=0');
END reloj;

ARCHITECTURE comportamiento OF reloj IS
...
END comportamiento;
```

```
PACKAGE arq_pack IS
  TYPE tri_estado IS ('0', '1', 'Z');
  TYPE vector_bus IS ARRAY (7 DOWNT0 0) OF tri_estado;
  TYPE array_vector_bus IS ARRAY (INTEGER RANGE <>) OF vector_bus;
  FUNCTION resolucion (entrada: array_vector_bus) RETURN vector_bus;
  --la señal del bus será la resuelta
  SUBTYPE bus_resuelto IS resolucion vector_bus;
  PROCEDURE logico_entero(VARIABLE vector:IN BIT_VECTOR;
                          VARIABLE entero:OUT INTEGER);
  PROCEDURE entero_logico(VARIABLE entero:IN INTEGER;
                           VARIABLE vector:OUT BIT_VECTOR);
END arq_pack;
PACKAGE BODY arq_pack IS
  .....
END arq_pack;
```