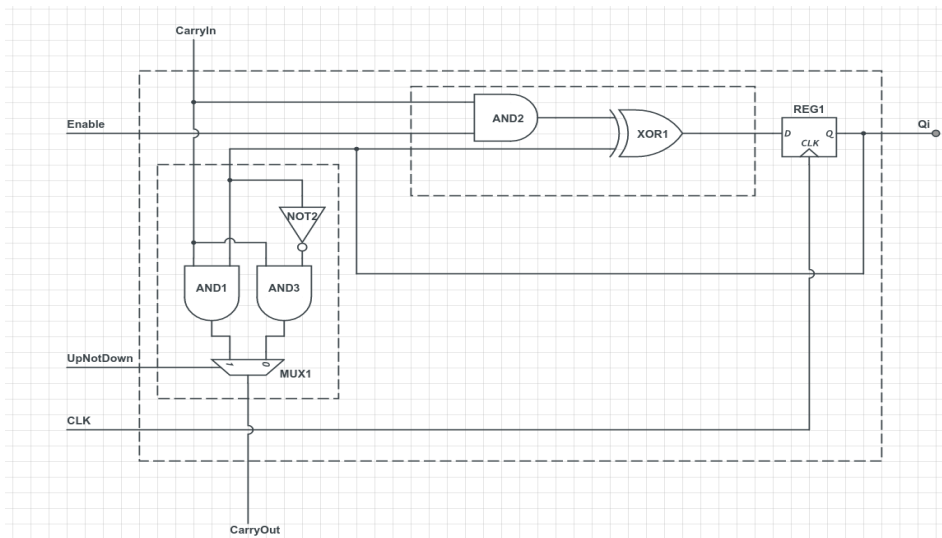


Actividad 7.1

1) El análisis de las etapas combinacionales se hizo en la *Actividad_2_1_2* . Aquí copio lo realizado en ese trabajo.

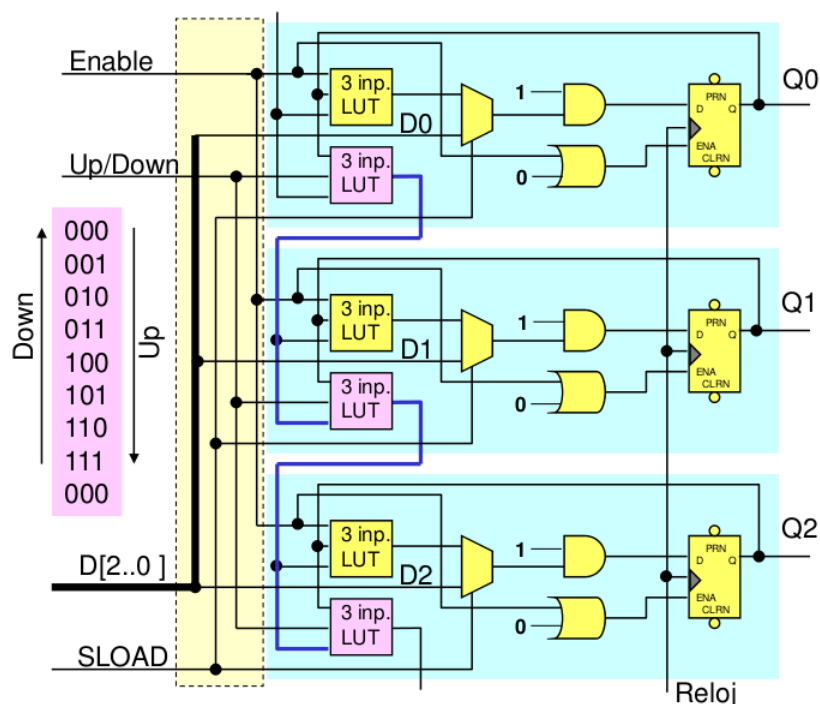
“ La etapa combinacional debe decidir dependiendo de los cambios de la etapa anterior si debe cambiar su estado, y de la misma forma comunicarla a la etapa siguiente si debe cambiar su estado. La entrada UP/DOWN indica la condición en que se debe indicar a la etapa siguiente que debe cambiar su estado en el próximo pulso de reloj.

En la siguiente imagen implemento una posible opción de la etapa combinacional, lo cual se comprobo mediante simulación

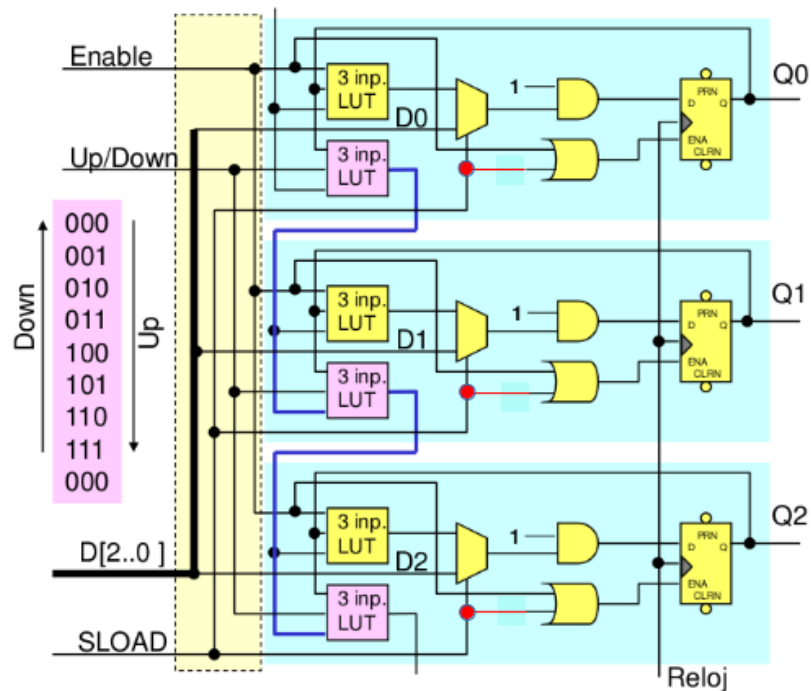


Donde si CarryIn = '1' indica que la etapa debe cambiar su estado (con lo que la etapa LSB siempre está en esa combinación). Por otro lado, cuando la etapa se encuentre en estado para que la siguiente etapa cambie de estado pondrá CarryOut = '1'. En toda otra condición, CarryIn y CarryOut se mantendrán en '0' sin modificar el estado de la etapa.”

Ahora, respecto a las prioridades de ENA y LOAD. Circuito con prioridad en ENA:



Circuito con prioridad en LOAD:



No se han encontrado diferencias significativas. Favor de notificar la respuesta al momento de corregirlo, gracias

2) A continuación se muestra la implementación del bloque combinacional BCD/CSA, en el cual se muestra a C/B como única dependencia del Reloj proporcionado.

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4
5 entity En_Head is
6   Port (
7     CLK      : in  STD_LOGIC;
8     DataIn   : in  STD_LOGIC_VECTOR(3 downto 0);
9     Contar   : in  STD_LOGIC;
10    U_D      : in  STD_LOGIC; -- '1' = Up, '0' = Down
11    Zero     : in  STD_LOGIC;
12    Unidades : in  STD_LOGIC;
13    DataOut  : out STD_LOGIC_VECTOR(3 downto 0)
14  );
15 end En_Head;
16
17 architecture Arq_Head of En_Head is
18   signal Aux: STD_LOGIC_VECTOR(4 downto 0);
19   signal C_B: STD_LOGIC;
20 begin
21   Aux <= "00000" when Zero = '1' else
22         '0' & DataIn when contar = '0' else
23         "10000" when U_D = '1' and Unidades = '1' and DataIn = "1001" else
24         "11001" when U_D = '0' and Unidades = '1' and DataIn = "0000" else
25         "10000" when U_D = '1' and C_B = '1' and DataIn = "1001" else
26         "11001" when U_D = '0' and C_B = '1' and DataIn = "0000" else
27         DataIn + "00001" when U_D = '1' and Unidades = '1' else
28         DataIn - "00001" when U_D = '0' and Unidades = '1' else
29         DataIn + "00001" when U_D = '1' and C_B = '1' else
30         DataIn - "00001" when U_D = '0' and C_B = '1';
31   DataOut <= Aux(3 downto 0);
32   C_B <= Aux(4) when rising_edge(CLK);
33 end Arq_Head;

```