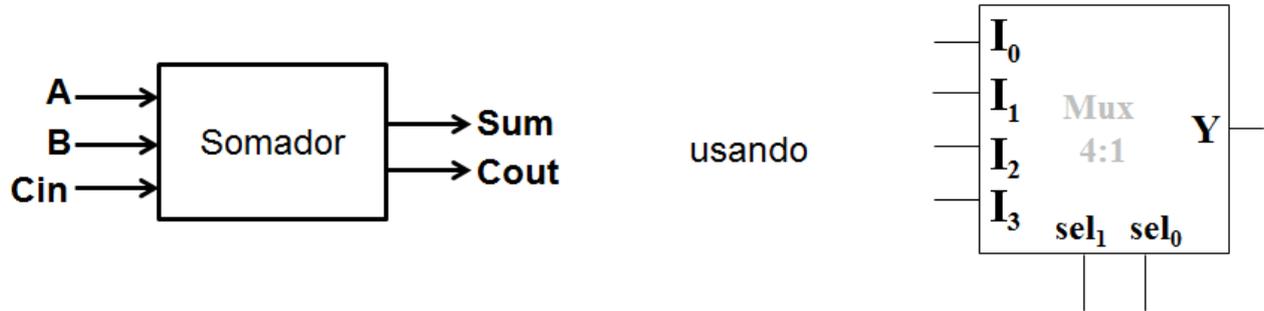
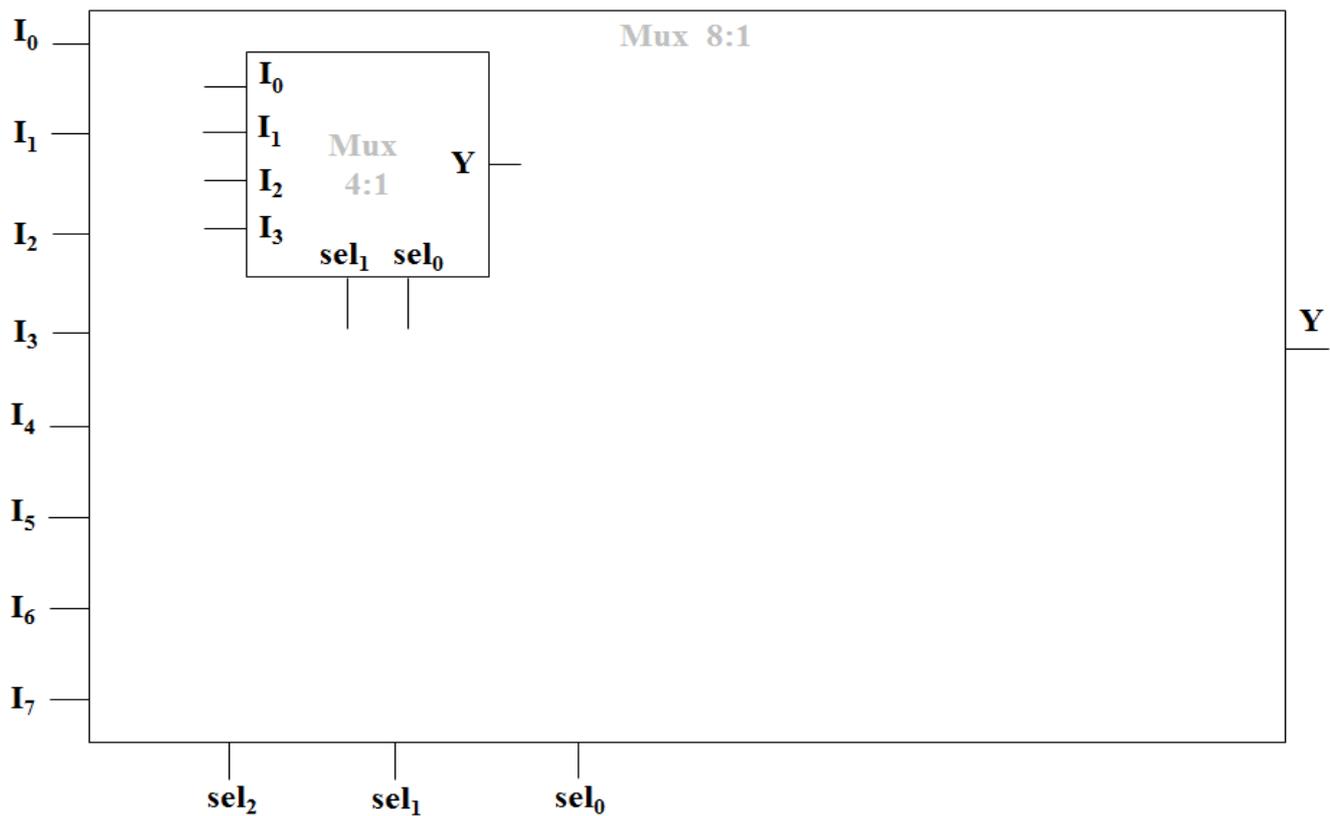


P3 - 2017

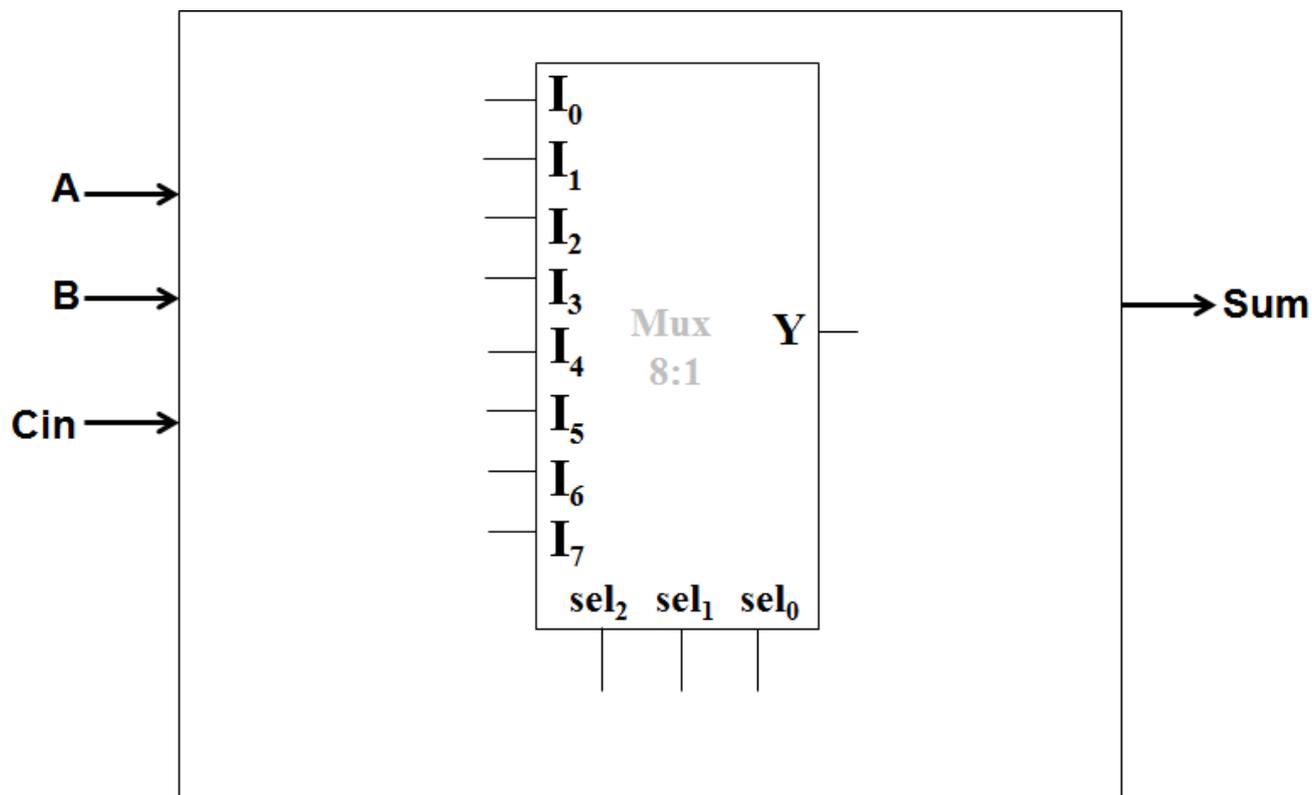
[2.5 pontos] Você foi designado para ajudar o MacGyver, da ULABRAS, que precisa de um somador completo de 1 bit para construir um certo hardware, como o ilustrado abaixo: **Sum** é resultado da soma binária entre A, B e Cin, enquanto Cout é o “vai-um” dessa soma. Entretanto, você só tem a sua disposição MUXes 4:1. Dado o prazo exíguo, você precisa tomar algumas decisões de projeto.



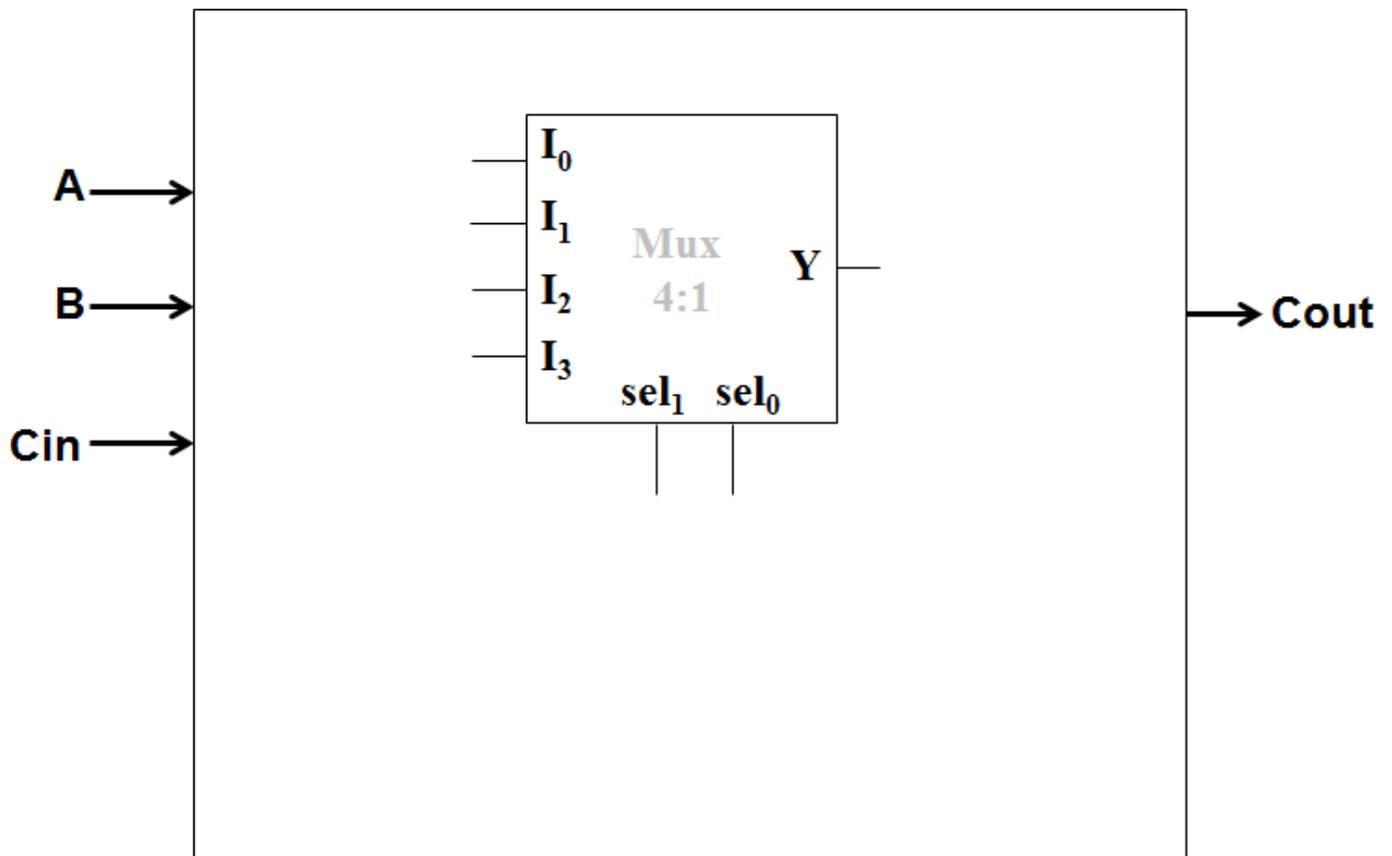
a) [0.8 ponto] Como o somador tem 3 entradas, seria conveniente que o MUX também tivesse 3 entradas de seleção, ou seja, que fosse um MUX 8:1. Construa um MUX 8:1, usando somente MUXes 4:1 na quantidade que julgar necessário.



b) [0.6 ponto] De posse de um MUX 8:1, construa o circuito para o cálculo de Sum, mostrando os passos intermediários que levaram à solução .



c) [0.8 ponto] Após construir o circuito para Sum, você percebe que lhe resta apenas um MUX 4:1. É possível usá-lo para construir o circuito para o cálculo de Cout? Em caso positivo, construa o circuito, mostrando os passos intermediários que levaram à solução. Em caso negativo, justifique.



d) [0.3 ponto] Após finalizar o projeto do somador, você percebe que seus MUXes 4:1 têm uma entrada adicional, “ent”, que você não havia notado. Seu colega lhe sugere que “ent” seja aterrado, ou seja, que seu valor lógico seja fixado em ‘0’, mas antes disso você analisa a documentação do circuito e descobre que o MUX corresponde ao seguinte código VHDL.

```

library IEEE;
use IEEE.std_logic_1164.all;

entity mux4_1 is
    port (sel
           : in  STD_LOGIC_VECTOR (1 downto 0);
          I0,I1,I2,I3
           : in  STD_LOGIC;
          ent
           : in  STD_LOGIC; -- entrada "misteriosa"
          Y
           : out STD_LOGIC);
end mux4_1;

architecture arch_mux of mux4_1 is
begin
    Y <=   'Z' when (ent = '0'
                  ) else
          I0 when (ent = '1' and sel = "00") else
          I1 when (ent = '1' and sel = "01") else
          I2 when (ent = '1' and sel = "10") else
          I3 when (ent = '1' and sel = "11") else
          '0'
          ; -- "catch all"
end arch_mux;

```

Se “ent” for aterrado, o circuito terá o comportamento desejado? Em caso positivo, justifique. Em caso negativo, explique o que deveria ser feito com a entrada “ent”.

P3 – 2017 (Q1)

Questão 1 (2,5) - O gerente da ULABRAS pediu um projeto para o monitoramento de oito sensores, todos com saídas digitais ativas em “0”, da planta industrial para o escritório dele. O gerente já tem um display de sete segmentos enorme e sem uso em sua sala, e precisa dos circuitos para que o display mostre o número do alarme quando acionado. Como vários alarmes podem disparar ao mesmo tempo, quando isso acontecer deve ser mostrado apenas o mais importante. Para construir o sistema, são disponibilizados cabos de conexão e portas lógicas simples, bem como alguns codificadores 74x148, um codificador BCD para 7 segmentos, e decodificadores 74x138 (os detalhes de funcionamento desses 3 últimos são mostrados ao final desta questão).

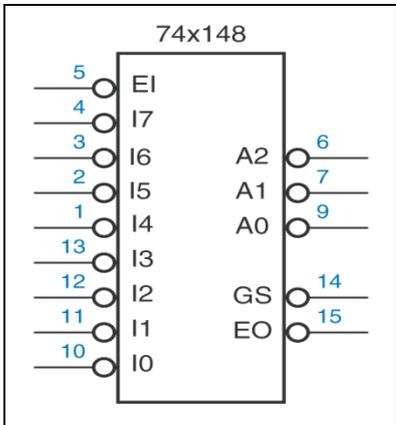
1.1 Considerando que o alarme número “7” é o mais importante e o número “0” é o menos importante, faça um esboço do projeto:

(dica: usar codificador 74x148 e decodificador de 7 segmentos)

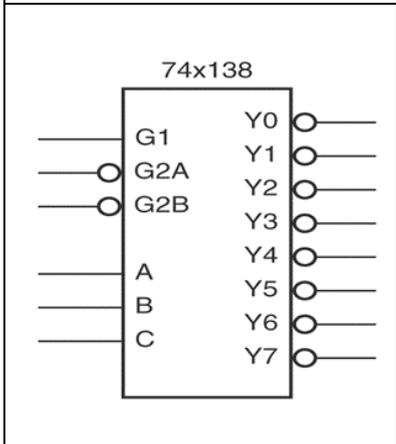
Ao testar seu projeto, você descobriu que o Codificador de 7 segmentos disponibilizado não funcionava. Para poupar tempo, o gerente preferiu alterar o projeto para que o alarme “simplesmente” acenda 1 dentre 8 LEDs (i.e., coloque sua entrada em “1”). Assim, cada LED identifica qual o alarme acionado, ainda respeitando a prioridade entre os alarmes. Ele disponibilizou os cabos de conexão, mas pediu que você economize pois, são caros. Altere seu projeto para acomodar esse novo pedido.

(dica: usar codificador 74x148 e decodificador 74138)

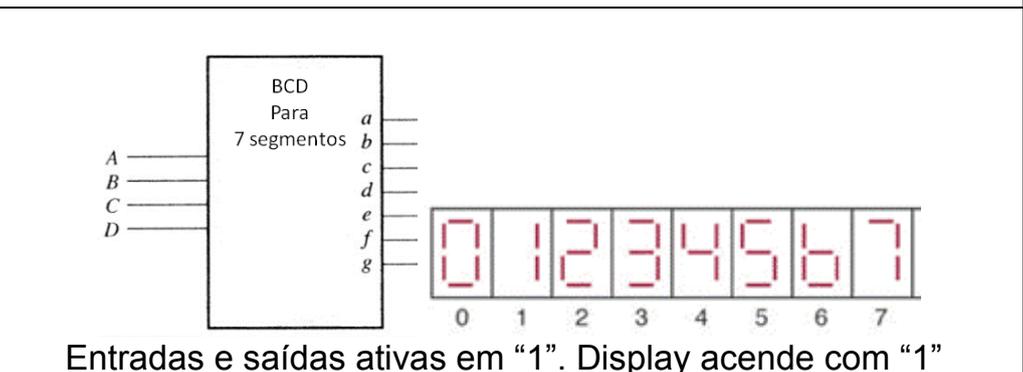
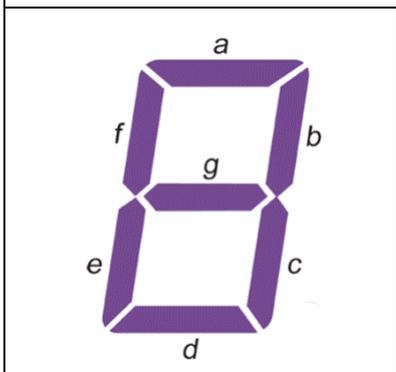
Como você não gostou da solução, considerando o material disponível, você decidiu implementar um codificador de 7 segmentos simples para, assim, impressionar o gerente. Implemente a parte do codificador que trata os segmentos “a” e “e” do display. Use apenas portas lógicas simples



EI	INPUTS								OUTPUTS				
	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H



G1	Inputs					Outputs							
	G2A_L	G2B_L	C	B	A	Y7_L	Y6_L	Y5_L	Y4_L	Y3_L	Y2_L	Y1_L	Y0_L
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1



Entradas e saídas ativas em “1”. Display acende com “1”

PREC – 2016

(1 ponto) Projete um somador completo de 1 bit (Figura 3), usando um multiplexador de 8 entradas (8 x 1) para gerar a saída s e um decodificador 3 x 8 para gerar a saída COUT. Use portas lógicas adicionais (AND, OR, NOT) caso seja necessário.

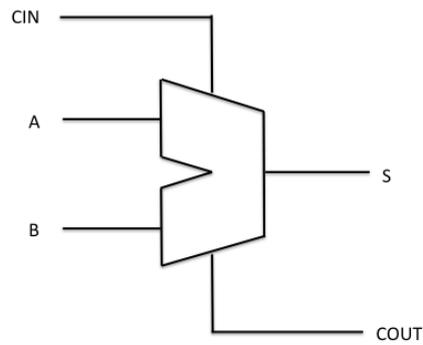
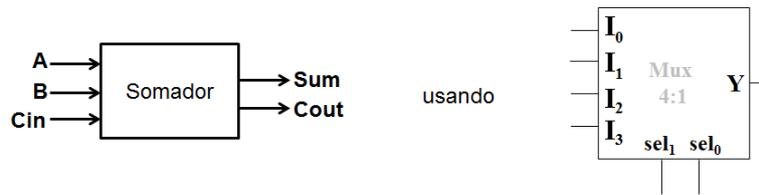


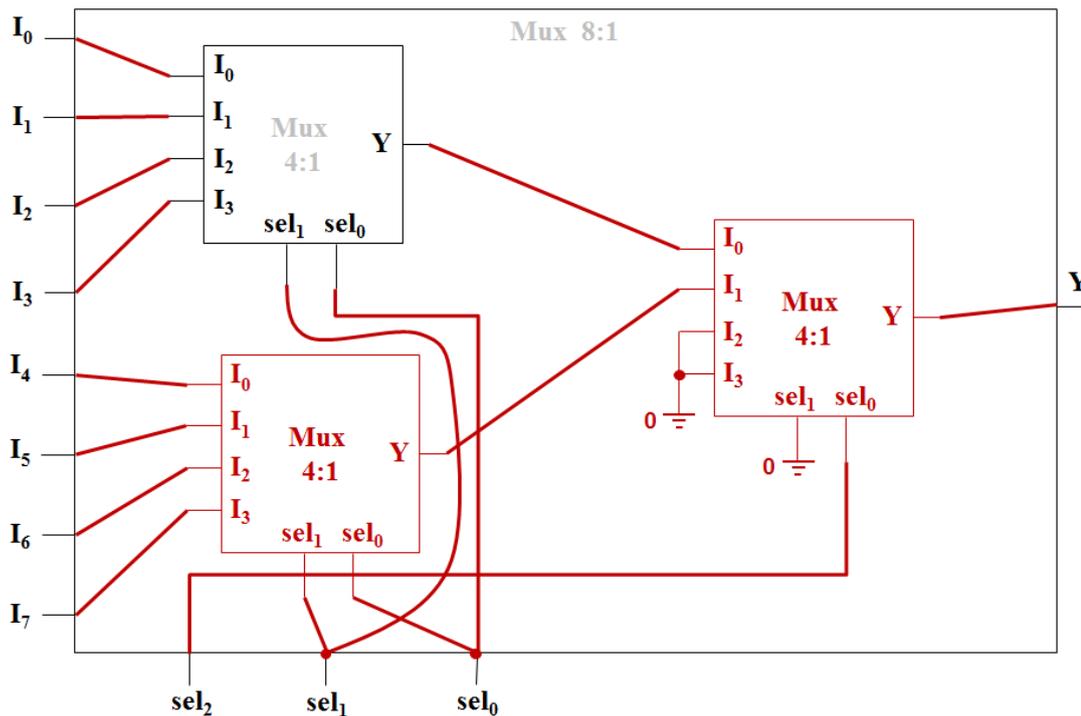
Figura 3 – Somador Completo de 1 bit

GABARITO – P3 2017

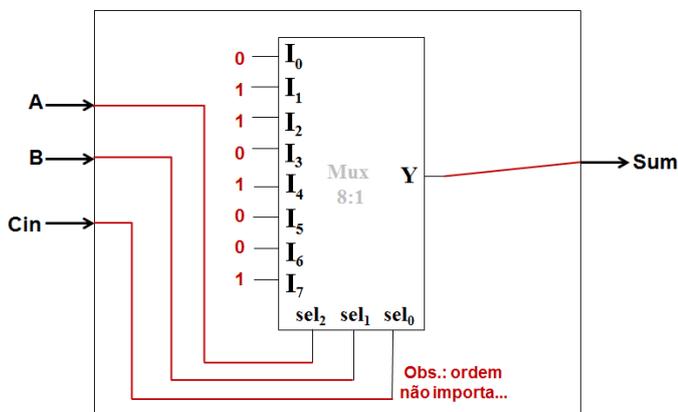
[2.5 pontos] Você foi designado para ajudar o MacGyver, da ULABRAS, que precisa de um somador completo de 1 bit para construir um certo hardware, como o ilustrado abaixo: **Sum** é resultado da soma binária entre A, B e Cin, enquanto **Cout** é o “vai-um” dessa soma. Entretanto, você só tem a sua disposição MUXes 4:1. Dado o prazo exiguo, você precisa tomar algumas decisões de projeto.



a) [0.8 ponto] Como o somador tem 3 entradas, seria conveniente que o MUX também tivesse 3 entradas de seleção, ou seja, que fosse um MUX 8:1. Construa um MUX 8:1, usando somente MUXes 4:1 na quantidade que julgar necessário.

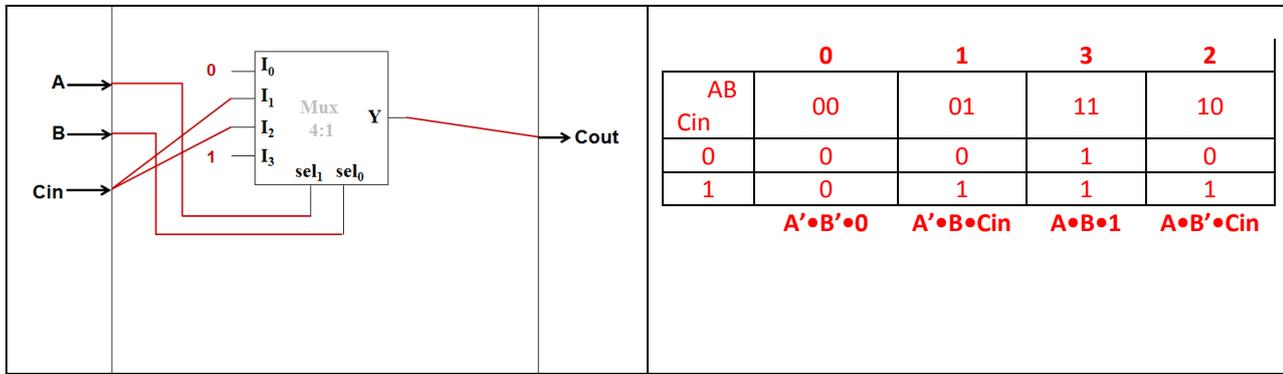


b) [0.6 ponto] De posse de um MUX 8:1, construa o circuito para o cálculo de Sum, mostrando os passos intermediários que levaram à solução .



A	B	Cin	Sum
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

c) [0.8 ponto] Após construir o circuito para Sum, você percebe que lhe resta apenas um MUX 4:1. É possível usá-lo para construir o circuito para o cálculo de Cout? Em caso positivo, construa o circuito, mostrando os passos intermediários que levaram à solução. Em caso negativo, justifique.



d) [0.3 ponto] Após finalizar o projeto do somador, você percebe que seus MUXes 4:1 têm uma entrada adicional, “ent”, que você não havia notado. Seu colega lhe sugere que “ent” seja aterrado, ou seja, que seu valor lógico seja fixado em ‘0’, mas antes disso você analisa a documentação do circuito e descobre que o MUX corresponde ao seguinte código VHDL.

```

library IEEE;
use IEEE.std_logic_1164.all;
entity mux4_1 is
    port (sel          : in  STD_LOGIC_VECTOR (1 downto 0);
          I0,I1,I2,I3 : in  STD_LOGIC;
          ent          : in  STD_LOGIC; -- entrada "misteriosa"
          Y           : out STD_LOGIC);
end mux4_1;
architecture arch_mux of mux4_1 is
begin
    Y <= 'Z' when (ent = '0' ) else
        I0 when (ent = '1' and sel = "00") else
        I1 when (ent = '1' and sel = "01") else
        I2 when (ent = '1' and sel = "10") else
        I3 when (ent = '1' and sel = "11") else
        '0' ; -- "catch all"
end arch_mux;

```

Se “ent” for aterrado, o circuito terá o comportamento desejado? Em caso positivo, justifique. Em caso negativo, explique o que deveria ser feito com a entrada “ent”.

Se “ent” for colocado em 0, a saída do MUX fica em alta impedância (Z). O correto é fazer “ent = 1” (conectar ao Vcc)