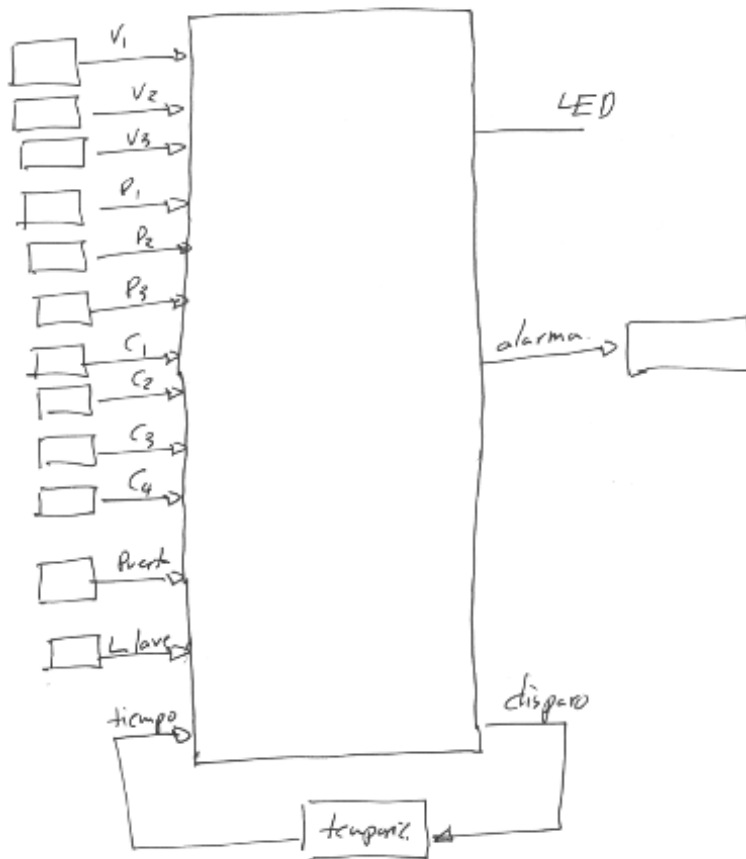


1 El diagrama de bloques del sistema ser :



Como en el caso de ~~ventana~~ apertura de una ventana, da igual que ventana sea o si hay m s de una abierta a la vez, pueden hacer

$$V = V_1 + V_2 + V_3$$

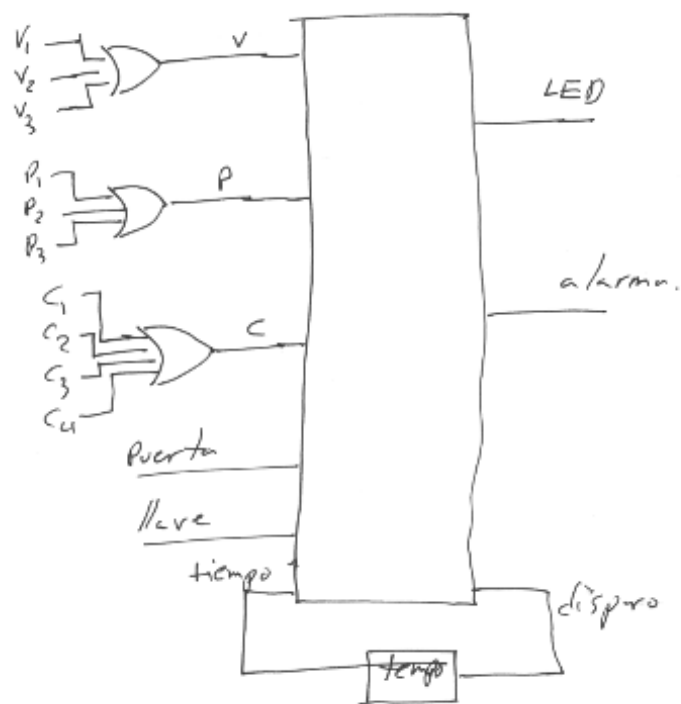
una se al que indica si hay alguna ventana abierta.

Igual con las persianas y los cristales:

$$P = P_1 + P_2 + P_3$$

$$y \quad C = C_1 + C_2 + C_3 + C_4$$

Y el circuito queda simplificado como sigue:



Vamos a ver cuando se activa cada una de las tres señales de salida:

- LED. Se activará solo en el caso de que se cierre la llave ($llave = 1$) y haya alguna ventana abierta ($V = 1$) o alguna persiana subida ($P = 1$)

$$LED = llave \cdot (V + P)$$

- Alarma: Se activará ~~de~~ en el caso de que la llave este cerrada (sistema activado) $llave = 1$ y que hayan pasado 30s o más desde la activación de l sistema (es decir, $tiempo = 1$) y se abra una ~~ventana~~ ventana ($V = 1$) o se suba una

persiana ($P=1$) o se rompa un cristal ($C=1$)

esto implica $Alarma = llave \cdot tiempo \cdot (V+P+C)$

pero también se activará si se abre la puerta (puerta=1) ~~pasados 30s de la apertura de la puerta (es decir tiempo=1) aún no se ha desactivado la llave (llave=1)~~. Por lo tanto:

$$Alarma = llave \cdot tiempo \cdot (V+P+C) + llave \cdot tiempo \cdot Puerta$$

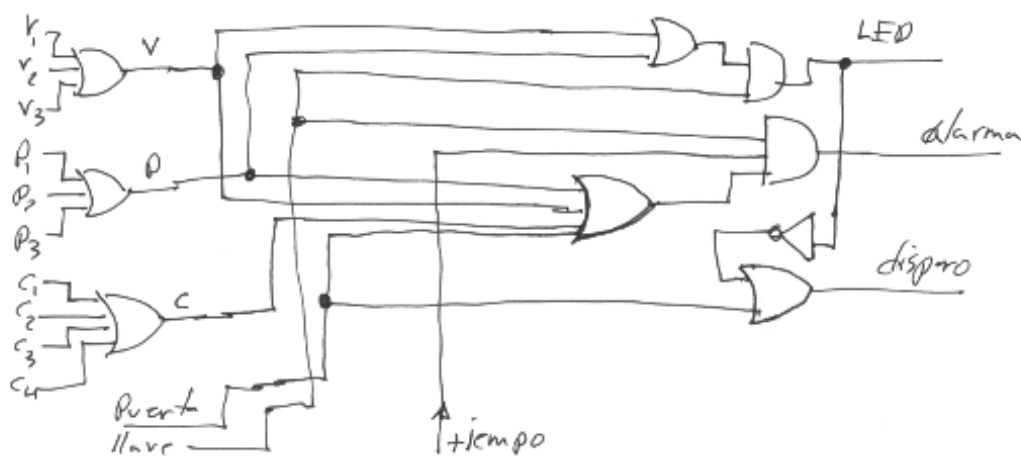
⇓

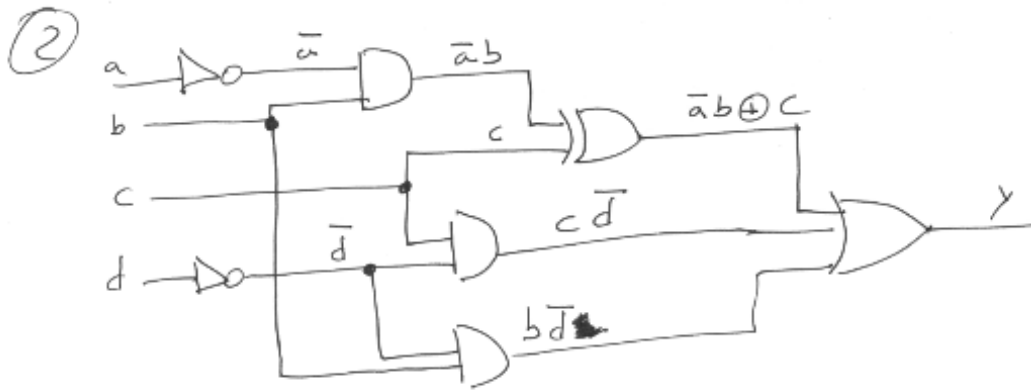
$$Alarma = llave \cdot tiempo \cdot (V+P+C+Puerta).$$

- Disparo: La temporización de 30s se lanza (con $disparo=1$) cuando se gira la llave y todo está correcto (que sería cuando no se enciende el led) o bien cuando se abre la puerta.

$$disparo = \overline{LED} + Puerta.$$

El circuito quedaría entonces:





1)
$$y = \bar{a}b \oplus c + c\bar{d} + b\bar{d} = \bar{a}b \oplus c + \bar{a}b\bar{c} + c\bar{d} + b\bar{d}$$

$$= (a + \bar{b})c + \bar{a}b\bar{c} + c\bar{d} + b\bar{d} = \boxed{ac + \bar{b}c + \bar{a}b\bar{c} + c\bar{d} + b\bar{d}}$$

2) ac produce 4 minterms en: $1x1x$

↓
1010
1011
1110
1111

$\bar{b}c$ produce 4 minterms en: $x01x$

↓
0010
0011
1010
1011

$\bar{a}b\bar{c}$ produce 2 minterms en: $010x$

↓
0100
0101

$c\bar{d}$ produce 4 minterms en: $xx10$

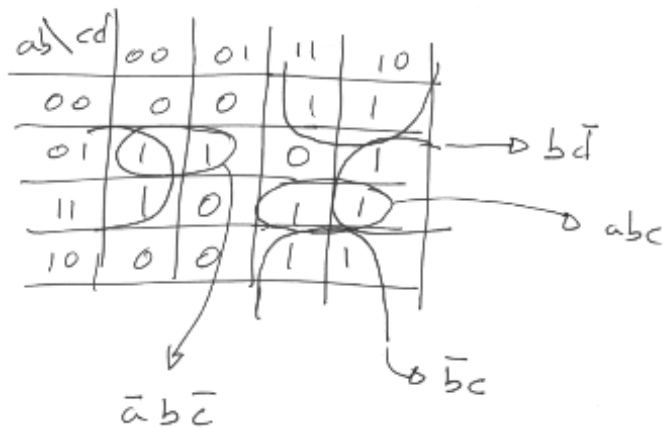
↓
0010
0110
1010
1110

$b\bar{d}$ produce 4 minterms en: $x1x0$

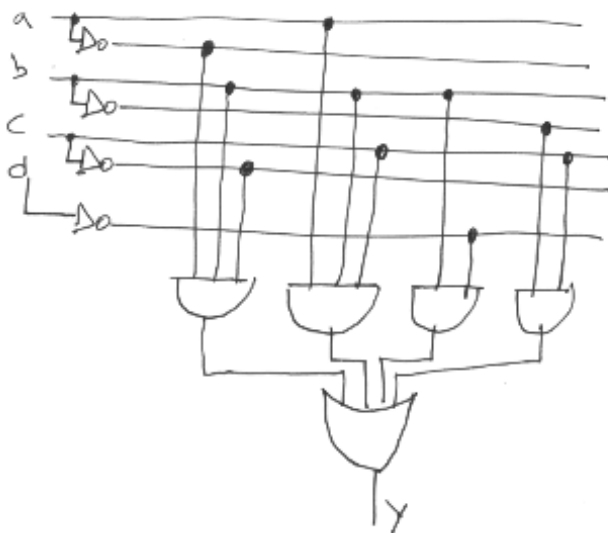
↓
0100
0110
1100
1110

abcd	y
0000	0
0001	0
0010	1
0011	1
0100	1
0101	1
0110	1
0111	0
1000	0
1001	0
1010	1
1011	1
1100	1
1101	0
1110	1
1111	1

3) La simplificación por Karnaugh queda:



4)

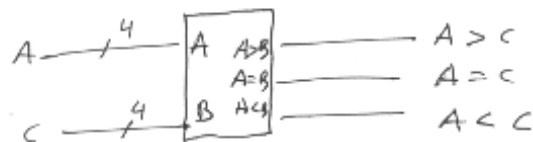
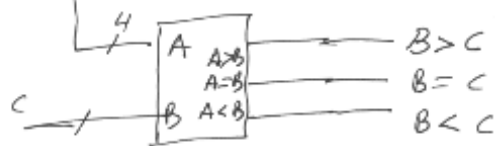
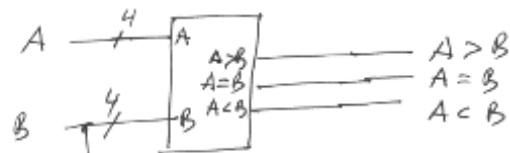


5) Sería igual pero sustituyendo todas las puertas ^{OR y AND} por puertas NAND ya que



Las negadores se sustituyen por

③ Lo que se va a hacer es una comparación. Como no hay comparadores de 3 números, se van a comparar dos a dos de la siguiente forma:



A partir de estas salidas, se van a generar 6

señales: $A_{es\ mayor}$, $B_{es\ mayor}$, $C_{es\ mayor}$,
 $A_{es\ menor}$, $B_{es\ menor}$, $C_{es\ menor}$

En principio parece ser:

$$A_{es\ mayor} = A > B \cdot A > C$$

$$B_{es\ mayor} = B > C \cdot A < B$$

$$C_{es\ mayor} = A < C \cdot B < C$$

$$A_{es\ menor} = A < B \cdot A < C$$

$$B_{es\ menor} = A > B \cdot B < C$$

$$C_{es\ menor} = A > C \cdot B > C$$

pero en esas ecuaciones no se tiene en cuenta el caso en el que dos números sean iguales, por ejemplo si

$A=B > C$ no se activa ni A es mayor ni B es mayor ni C es mayor. Para evitar ese problema, voy a hacer que si $A=B > C$ se active A es mayor, si $B=C > A$ se active B es mayor (y lo mismo para A es menor y B es menor).

$$A \text{ es mayor} = \overline{A < B} \cdot \overline{A < C}$$

$$B \text{ es mayor} = \overline{B < C} \cdot \overline{A < B}$$

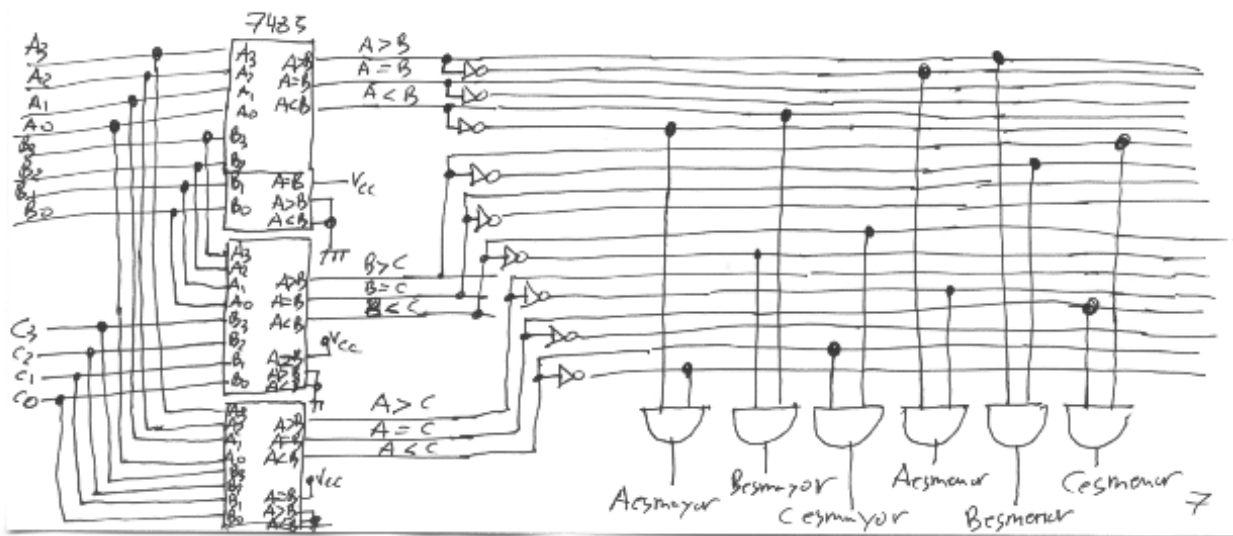
$$C \text{ es mayor} = \overline{A < C} \cdot \overline{B < C}$$

$$A \text{ es menor} = \overline{A > B} \cdot \overline{A > C}$$

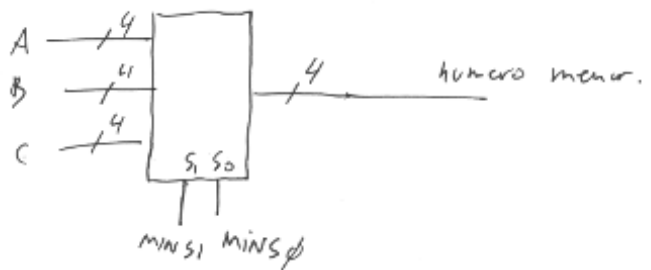
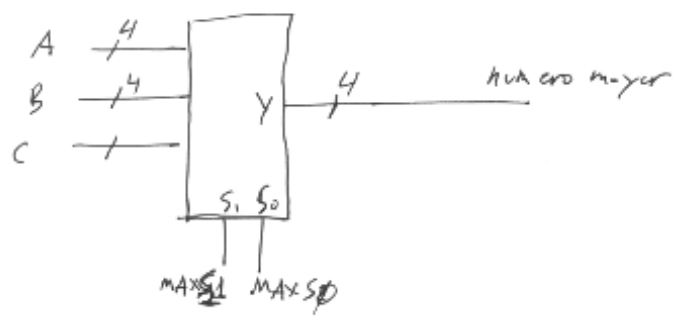
$$B \text{ es menor} = \overline{A > B} \cdot \overline{B > C}$$

$$C \text{ es menor} = \overline{A > C} \cdot \overline{B > C}$$

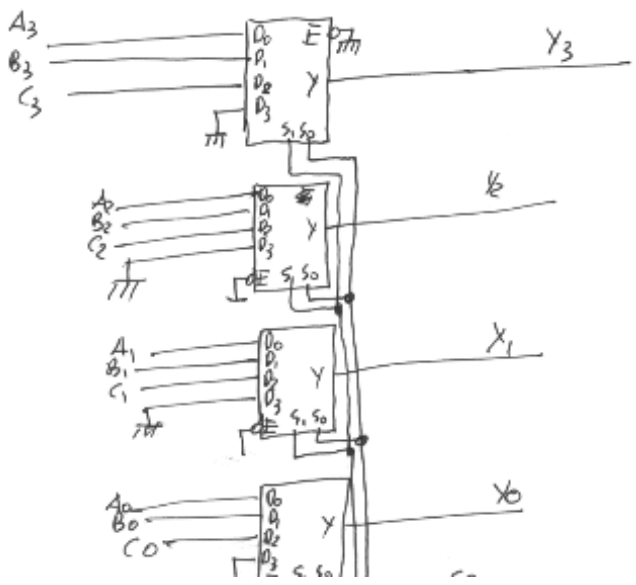
El comparador total queda:



Una vez que se sabe que número es mayor y cuál es menor, se seleccionará estos, con un multiplexor:



Las ~~señales~~ Cada uno de estas multiplexores será un multiplexor múltiple 4:1, realizado a partir de 74153, como sigue:



Las señales de selección $MAXSD$, $MAXSI$, $MINSD$ y $MINSI$ se seleccionarán según las tablas de verdad:

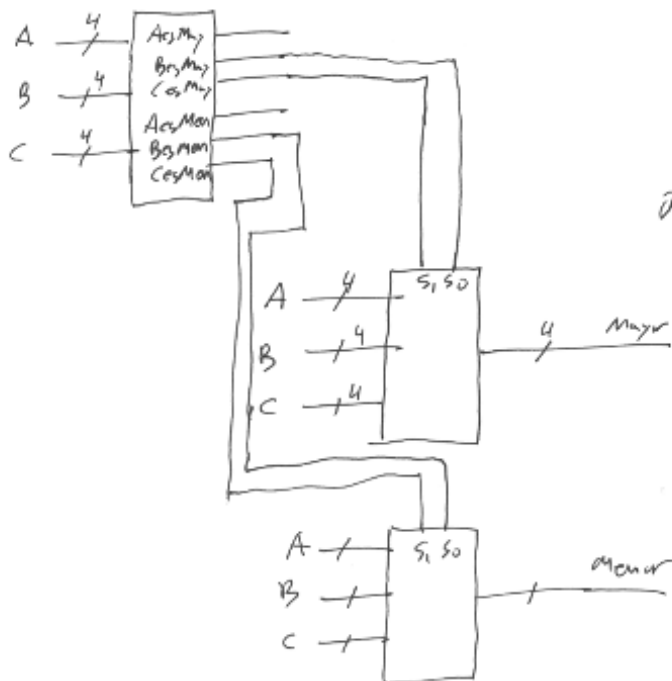
Acsmayor	Besmayor	Cesmayor	$MAXSI$	$MAXSD$
1	x	x	0	0
x	1	x	0	1
x	x	1	1	0

\Rightarrow $MAXSI = Cesmayor$
 $MAXSD = Besmayor$

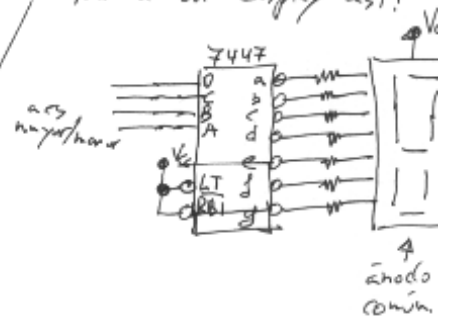
Acsmenor	Besmenor	Cesmenor	$MINSI$	$MINSD$
1	x	x	0	0
x	1	x	0	1
x	x	1	1	0

\Rightarrow $MINSI = Cesmenor$
 $MINSD = Besmenor$

El circuito queda:

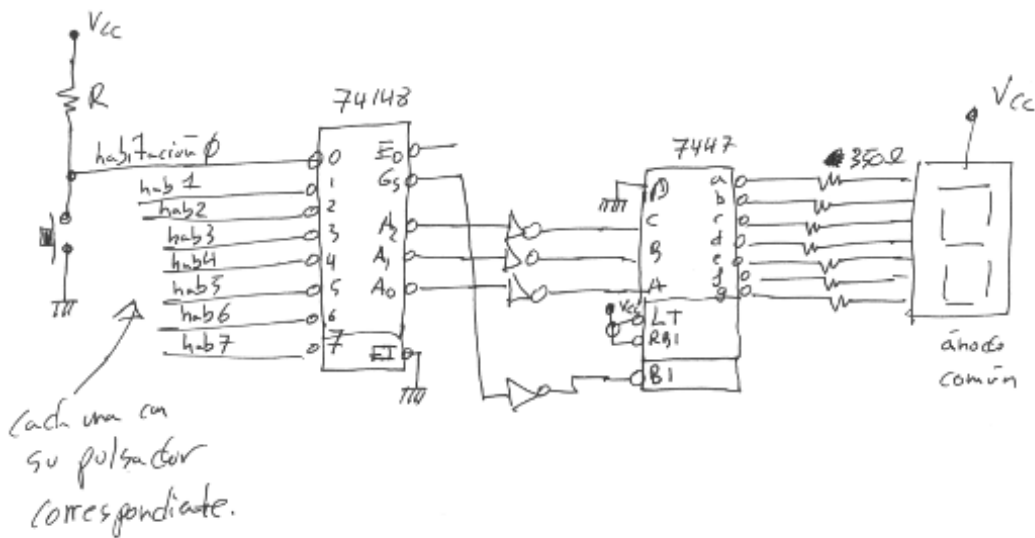


Cada uno de estas salidas irá a un display así:



4) Lo que se está pidiendo coincide casi totalmente con el comportamiento de un codificador 8:3. Voy a usar el

74148



El codificador dará a su salida el código de la habitación a la que se ha activado el pulsador, pero en lógica negativa (invertido). Con 3 inversores se convierte a lógica positiva. Para que cuando no haya ningún pulsador activado no aparezca el ϕ en el display, la entrada BI se activa cuando $\phi S = 1 \Rightarrow$ se apaga el display.



Las resistencias se calculan para 10mA suponiendo que a el led

$$\text{cae } 115V \Rightarrow R_0 = \frac{5V \cdot 115V}{10mA} = 350.$$

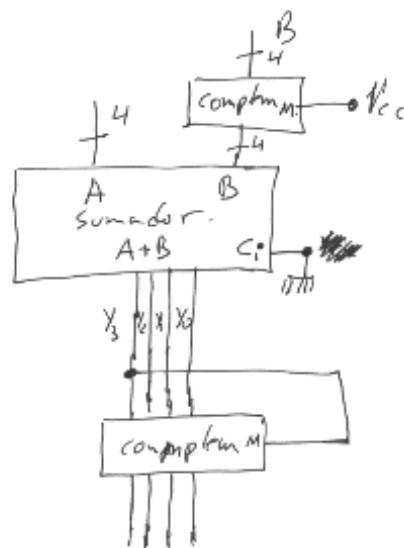
La R de los pulsadores serán de 10K, que con una $I_{IH} = 40\mu A$ para el 74LS148 produce una caída de tensión de 0.4V, lo que es perfectamente válido.

⑤ Lo que se va a hacer es restar los números, con lo cual el resultado serán la diferencia entre ambos, pero como este resultado puede ser positivo o negativo, habrá que invertirlo si es positivo. Es decir.

$$X = A - B$$

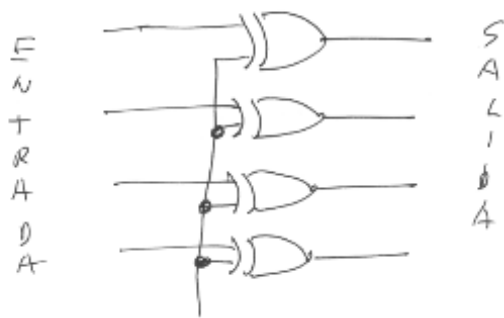
$$Y = X \quad \text{si } X \text{ positivo}$$

$$Y = -X \quad \text{si } X \text{ negativo.}$$



El circuito complementador complementa a 1 el número para $M=1$, o lo deja sin complementar para $M=0$. Para simplificar el circuito, se dibuja en complemento a 1 ($Y = A - B$ en complemento a 1).

El complementador seri:



El sumador seri en 7483.