

TEMA 6: Circuitos aritmético / lógicos**OBJETIVOS**

Experimentar con diferentes circuitos aritmético / lógicos contruidos mediante puertas convencionales o bien con dispositivos integrados comerciales. Al final experimentaremos con una unidad aritmético lógica (ALU) diseñada expresamente para este curso.

RELACIÓN DE MATERIALES

- Entrenador Universal Trainer o plataforma equivalente para experimentación de circuitos digitales; cables de conexión.
- C. Integrados: 1 x 74xxx04; 1 x 74xxx08; 1 X 74xxx32; 1 x 74xxx47; 1 x 74xxx85; 1 x 74xxx86; 1 x 74xxx280; 2 x 74xxx283; 1 x MKE-3.0 (ALU).

INDICE GENERAL**P1: SUMADORES BINARIOS**

P1-1 Semisumador

P1-2 Sumador completo

P1-3 Sumador de 4 bits con el 74xxx283

P1-4 Sumador de 4 bits con visualización

P1-5 Sumador con corrección BCD

P1-6 Sumador con corrección BCD y visualización

P2: RESTADORES BINARIOS

P2-1 Semirestador

P2-2 Restador completo

P2-3 Circuito sumador/restador de 4 bits

P3: COMPARADORES DE MAGNITUD

P3-1 Comparador básico de 1 bit

P3-2 Comparador de 4 bits con el 74xxx85

P4: GENERADORES/DETECTORES DE PARIDAD

P4-1 Generador/Detector de paridad de 9 bits con el 74xxx280

P5: UNIDAD ARITMÉTICO / LOGICA; LA ALU

P5-1 Nuestra ALU: MKE-3.0

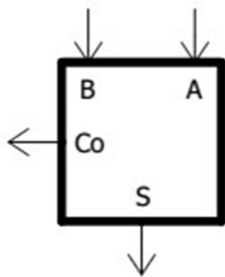
TEMA 6: Circuitos aritmético / lógicos

P1: SUMADORES BINARIOS

Son circuitos capaces de sumar números binarios de n bits cada uno. Realmente son circuitos combinacionales en los que las salidas dependen de las combinaciones de entrada. En este caso las salidas se producen en función de las entradas según los algoritmos o reglas propias de la suma en binario. Te recomiendo que las repases.

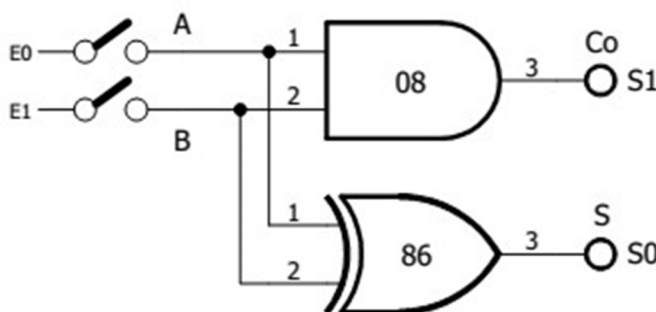
$$\begin{array}{r}
 A = 0111 \\
 B = 0101 \\
 \hline
 S = 1100
 \end{array}$$

P1-1 Semisumador



Es el circuito más básico que se puede simbolizar como un circuito que recibe dos números de 1 bit, el A y el B, y obtiene la suma S junto con una salida de llevada Co (Carry out) si la hubiera.

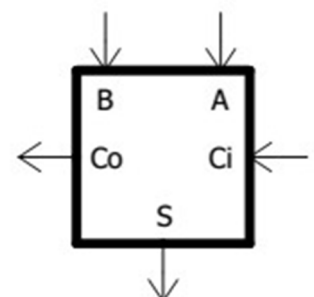
Monta el circuito de la figura y completa la tabla de la verdad. Comprueba que responde a los algoritmos de la suma que se estudian en la aritmética binaria.



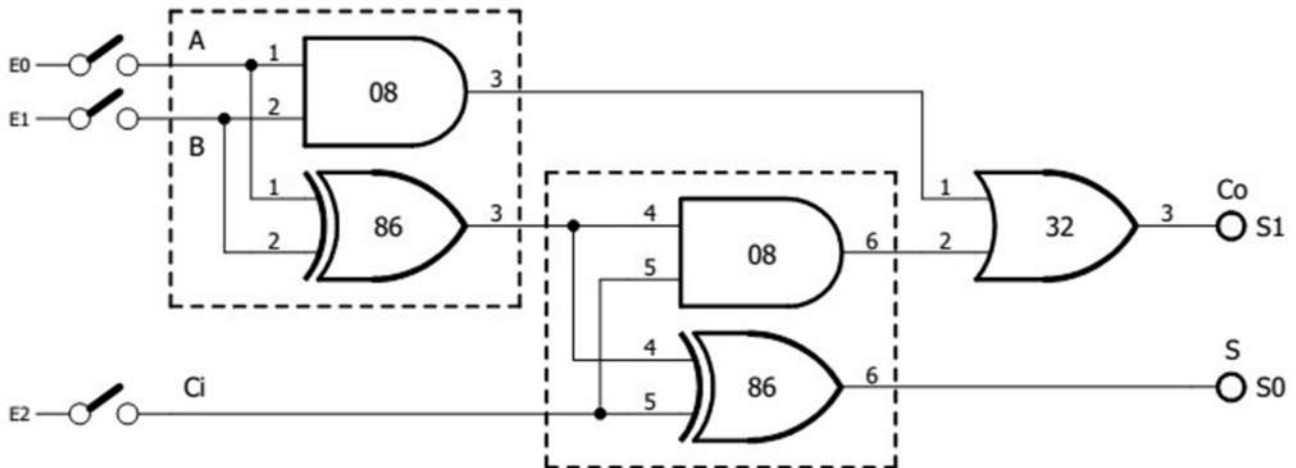
B	A	S	Co
0	0		
0	1		
1	0		
1	1		

P1-2 Sumador completo

El problema que presenta un semisumador es que no tiene en cuenta las llevadas que pudiera haber procedentes de otros semisumadores previos. La solución la encontramos en los sumadores completos que podemos simbolizar como en la figura.



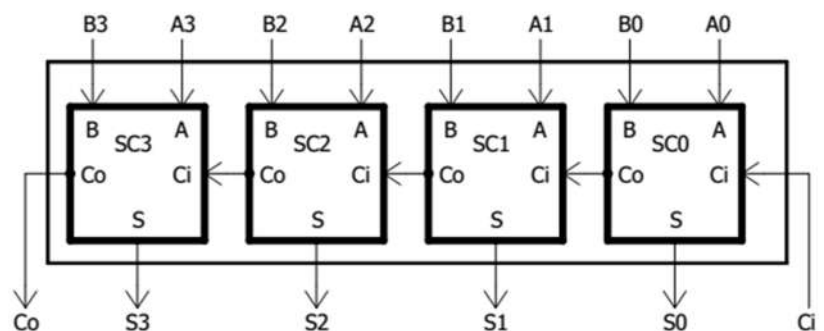
Al igual que en el semisumador tenemos los dos números de 1 bit a sumar, el A y el B. También tenemos la salida S resultante de la suma y la salida de llevada si lo hubiera, Co. Sin embargo ahora tenemos una entrada de acarreo, Ci, que estará activa si hay una llevada previa. Un sumador completo se construye a partir de dos semisumadores como puedes ver en el esquema de la figura.



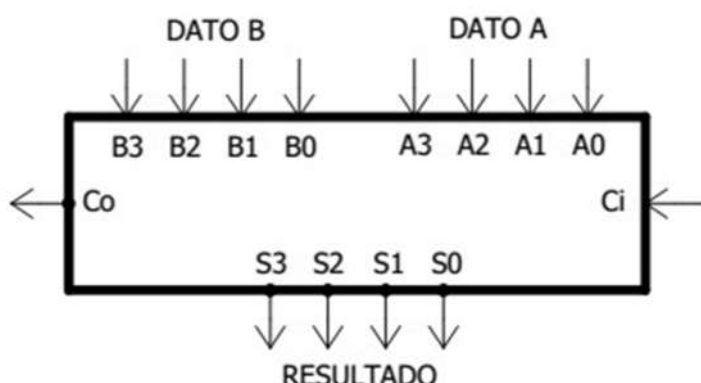
ENTRADAS			SALIDAS	
Ci	B	A	S	Co
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

El semisumador de la izquierda realiza la suma de los bits A y B. El resultado se aplica al semisumador de la derecha que lo suma con la entrada Ci de llevada previa. Ahora sí que tenemos una salida S de suma, que tiene en cuenta los bits de datos y el de la llevada. Una vez que montes el circuito completa la tabla de la verdad y comprueba nuevamente que responde a las leyes de la suma estudiadas en la aritmética binaria.

Podemos encadenar varios sumadores completos para realizar sumas de números de tantos bits como sea necesario. En la figura tienes un sumador de 4 bits. Puedes apreciar como a la entrada Ci de cualquiera de ellos se le conecta la salida Co del previo.



Todo ello se puede representar de forma más resumida como puedes ver en la figura. Tanto el dato A como el dato B están compuesto de 4 bits: A3:A0 y B3:B0, siendo

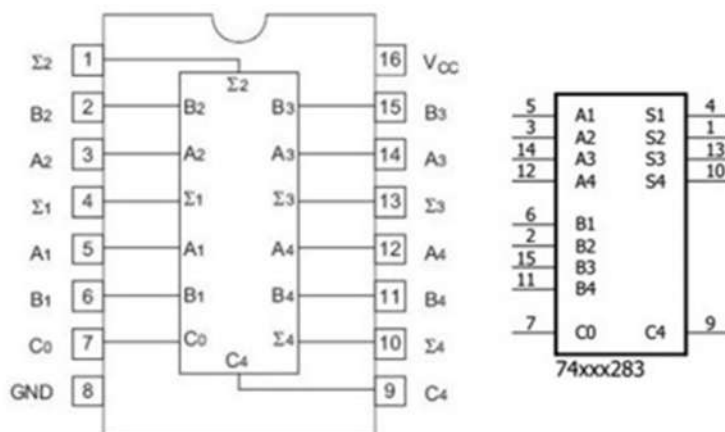


el bit 0, el de la derecha de cada dato, el bit de menos peso o LSB (Least Significant Bit), y el bit 3, el de la izquierda, el bit de más peso o MSB (Most Significant Bit).

El resultado de la suma también es de 4 bits: S3:S0, al que hay que añadir la salida de llevada Co.

P1-3 Sumador de 4 bits con el 74xxx283

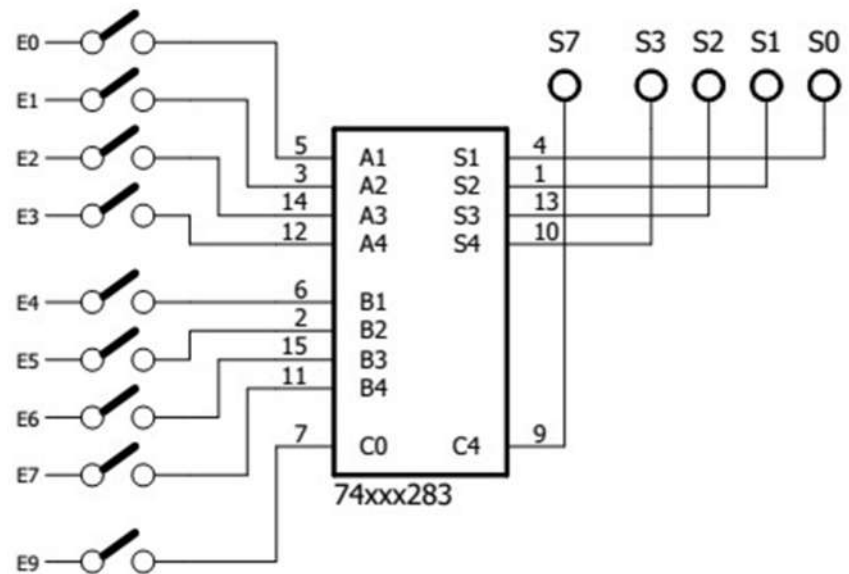
No te preocupes. Si precisas sumar dos números de 4 bits no tendrás que encadenar 4 sumadores completos en cascada. Dispones de circuitos integrados comerciales que lo implementa. Es el caso del modelo 74xxx283 que puedes ver en la figura.



Dispone de entradas para los datos A (A4:A1) y B (B4:B1) que son los sumandos. También tienes una entrada de acarreo previo, C0 (antes Ci), que te permitirá encadenar este sumador con otros precedentes. El resultado de la suma lo tienes en las salidas S4:S1 y el acarreo, si lo hubiera, está disponible en la patilla C4 (antes Co).

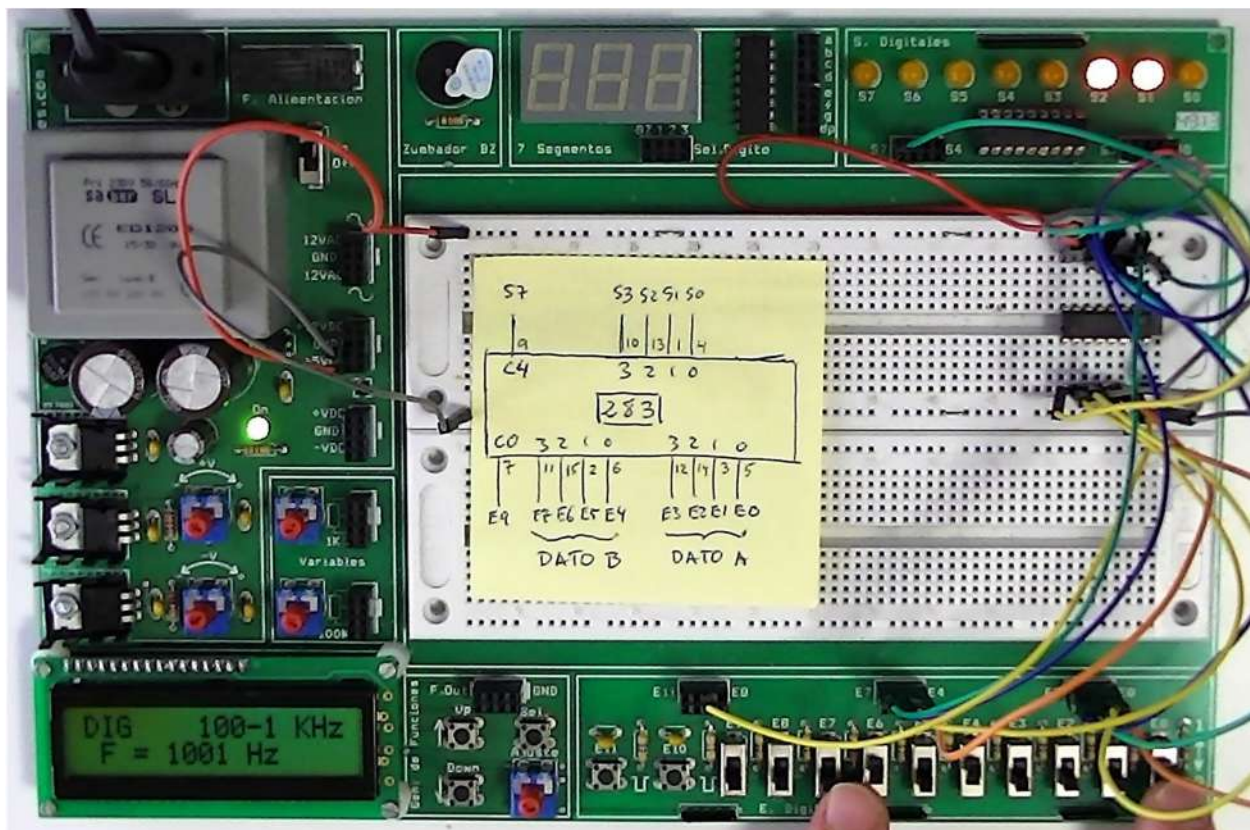
Te recuerdo que un sumador de 4 bits es capaz de sumar cualquier dígito BCD del 0 al 9 o hexadecimal del 0 a la F. Si encadenas varios en cascada podrás hacer sumas de 8 bits, 12, 16, 32, 64, etc...

En la figura tienes el esquema de la práctica. Los interruptores E3:E0 se emplean para introducir los cuatro bits del dato A, A4:A1. Los interruptores E7:E4 introducen los bits del dato B, B4:B1 mientras que con el interruptor E9, conectado con la entrada C0, simulas una posible llevada previa. El resultado de la suma lo puedes visualizar, en binario, mediante los leds S3:S0 y la salida de llevada, C4, con el led S7.



Debes de tener presente que los bits de menos peso (LSB), tanto de los datos de entrada A y B como de los de salida, deben situarse a la derecha respecto a los demás.

En la imagen tienes el montaje práctico realizado sobre el entrenador Universal Trainer.



Para que te "entregas" un poco te propongo que completes la tabla de la verdad, pero no lo hagas por hacer. Analiza los resultados de las diferentes sumas. Comprueba la influencia de la entrada de acarreo previo, C0 (E9), y el momento en que se produce acarreo de salida en C4 (S7).

ENTRADAS									SALIDAS				
C0 (E9)	DATO B				DATO A				C4 (S7)	SUMA			
	E7	E6	E5	E4	E3	E2	E1	E0		S3	S2	S1	S0
0	0	1	0	0	0	0	1	0					
0	1	0	0	0	0	0	0	1					
0	0	0	0	0	0	1	0	1					
0	0	0	1	1	0	1	1	1					
0	1	0	0	0	1	0	0	0					
0	0	1	1	1	1	0	0	1					
0	0	0	0	0	0	0	0	0					
0	1	1	1	1	1	1	1	1					
1	0	1	0	0	0	0	1	0					
1	1	0	0	0	0	0	0	1					
1	0	0	0	0	0	1	0	1					
1	0	0	1	1	0	1	1	1					
1	1	0	0	0	1	0	0	0					
1	0	1	1	1	1	0	0	1					
1	0	0	0	0	0	0	0	0					
1	1	1	1	1	1	1	1	1					

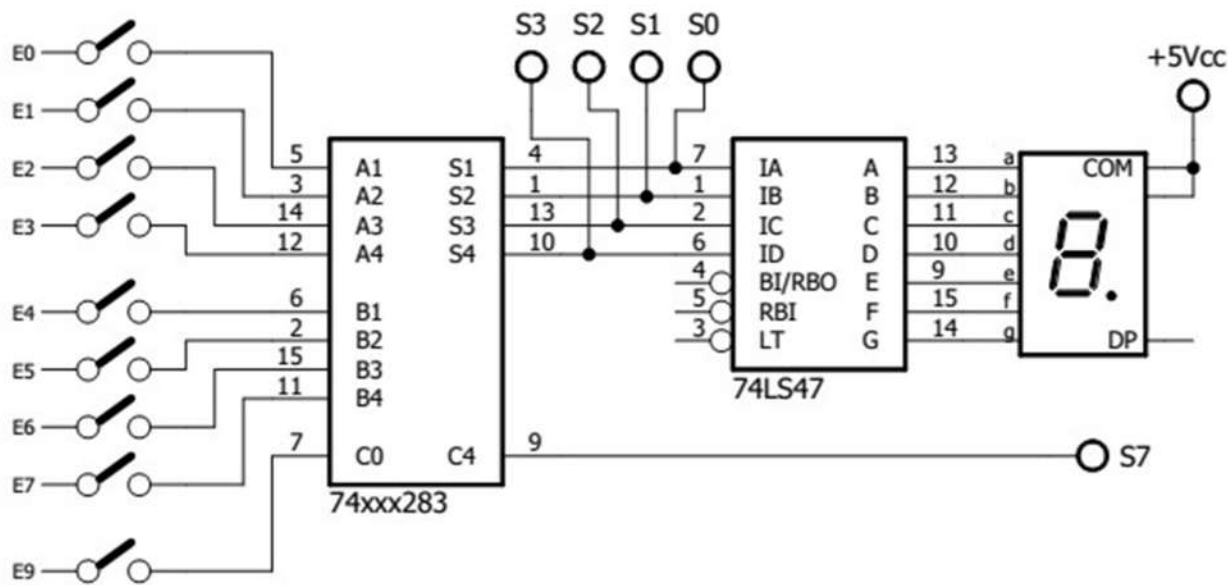
P1-4 Sumador de 4 bits con visualización

Rizando el rizo, vamos a recurrir a un viejo conocido que no es otro que el decodificador BCD a 7 segmentos que ya usaste en el tema anterior, el 74LS47. Monta el circuito de la figura.

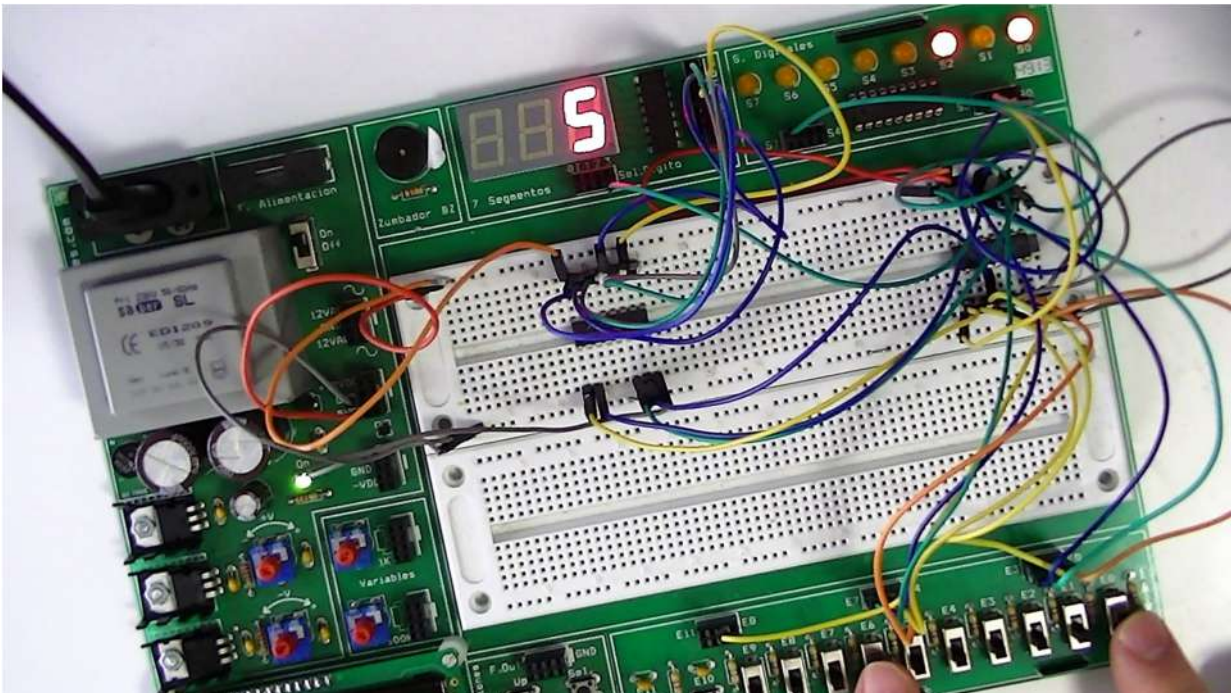
El resultado de la suma se aplica a ese decodificador que lo muestra de forma numérica sobre el display de 7 segmentos del entrenador. Puedes usar cualquiera de los tres, pero no olvides de conectar a +5Vcc el correspondiente ánodo común.

Nuevamente te propongo que completes la siguiente tabla en la que se proponen diferentes sumandos de entrada para el dato A y para el dato B. Anota tanto el resultado binario que muestran los leds S3:S0, como el símbolo que muestra el display.

Podrás observar que ciertos resultados no son visualizados correctamente sobre el display. Explica el por qué.



ENTRADAS									SALIDAS					
C0 (E9)	DATO B				DATO A				C4 (S7)	SUMA				DIS- PLAY
	E7	E6	E5	E4	E3	E2	E1	E0		S3	S2	S1	S0	
0	0	0	0	0	0	0	0	0						
0	0	0	1	1	0	0	1	0						
0	1	0	0	0	0	0	0	0						
0	0	1	1	1	0	1	0	1						
0	1	0	0	1	1	0	0	0						
0	0	0	1	0	0	1	1	1						
0	0	1	0	1	0	1	0	1						
0	0	1	1	1	0	1	1	1						
1	0	0	0	0	0	0	0	0						
1	0	0	1	1	0	0	1	0						
1	1	0	0	0	0	0	0	0						
1	0	1	1	1	0	1	0	1						
1	1	0	0	1	1	0	0	0						
1	0	0	1	0	0	1	1	1						
1	0	1	0	1	0	1	0	1						
1	0	1	1	1	0	1	1	1						



P1-5 Sumador con corrección BCD

En los anteriores montajes habrás podido comprobar que como consecuencia de sumar dos números BCD el resultado puede **NO** ser BCD. Recuerda, en BCD solo se pueden representar números del 0 (0000) al 9 (1001).

DEC.	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Mira estos ejemplos de izquierda a derecha:

9 = 1001	2 = 0010	6 = 0110	5 = 0101	5 = 0101	5 = 0101
8 = 1000	6 = 0110	7 = 0111	2 = 0010	5 = 0101	4 = 0100
17 = 0001	8 = 1000	13 = 1101	7 = 0111	10 = 1010	9 = 1001

- Al sumar 8 + 9 el resultado **SI** es BCD (0001), pero **NO** es el esperado.
- Al sumar 2 + 6 el resultado **SI** es BCD (1000), OK.
- Al sumar 6 + 7 el resultado **NO** es BCD (1101).
- Al sumar 5 + 2 el resultado **SI** es BCD (0111), OK.
- Al sumar 5 + 5 el resultado **NO** es BCD (1010).
- Al sumar 5 + 4 el resultado **SI** es BCD (1001), OK.

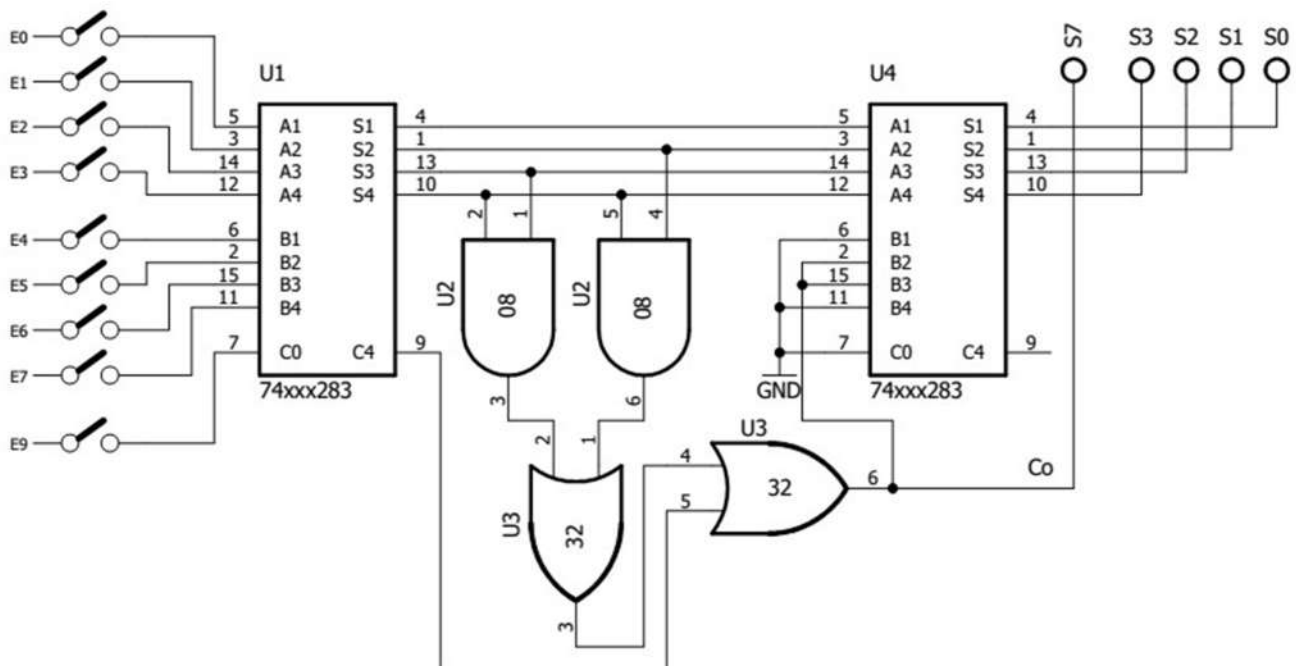
Si lo que deseas es ver un resultado BCD tendrás que hacer un ajuste decimal o "corrección BCD" cuando en el resultado se den una de estas dos circunstancias:

- Que ese resultado sea mayor de 9 (1010, 1011, 1100, 1101, 1110 o 1111)
- Que ese resultado haya producido una llevada o desbordamiento ($C4=1$).

¿Qué se hace en esos casos? Al resultado de la suma se le vuelve a sumar 6 (0110). Fíjate en las mismas sumas de antes. En aquellas ocasiones en las que se hacía necesario el ajuste decimal, el resultado tras sumar 6 ofrece una corrección a BCD.

9 = 1 0 0 1	2 = 0 0 1 0	6 = 0 1 1 0	5 = 0 1 0 1	5 = 0 1 0 1	5 = 0 1 0 1
8 = 1 0 0 0	6 = 0 1 1 0	7 = 0 1 1 1	2 = 0 0 1 0	5 = 0 1 0 1	4 = 0 1 0 0
17 = 0 0 0 1	8 = 1 0 0 0	13 = 1 1 0 1	7 = 0 1 1 1	10 = 1 0 1 0	9 = 1 0 0 1
+6 = 0 1 1 0		+6 = 0 1 1 0		+6 = 0 1 1 0	
17 = 1 0 1 1 1		13 = 1 0 0 1 1		10 = 1 0 0 0 0	

Monta el siguiente circuito y experimenta con él.



El primer sumador (U1) suma ambos datos A y B. El resultado se aplica al segundo sumador (U4) y a las puertas lógicas AND y OR contenidas en U2 y U3. Estas puertas detectan en qué casos hay que hacer la corrección BCD:

si C4=1 o si las salidas S4:S1 de U1 son = 1010, o 1011, o 1100, o 1101, o 1110 o 1111

Si la señal de corrección que generan es "0", el segundo sumador (U4) suma 0 (0000) al resultado de A + B, que se queda como está. Si la señal de corrección es "1" entonces el segundo sumador (U4) le suma 6 (0110) al resultado de A + B, haciendo así el ajuste BCD.

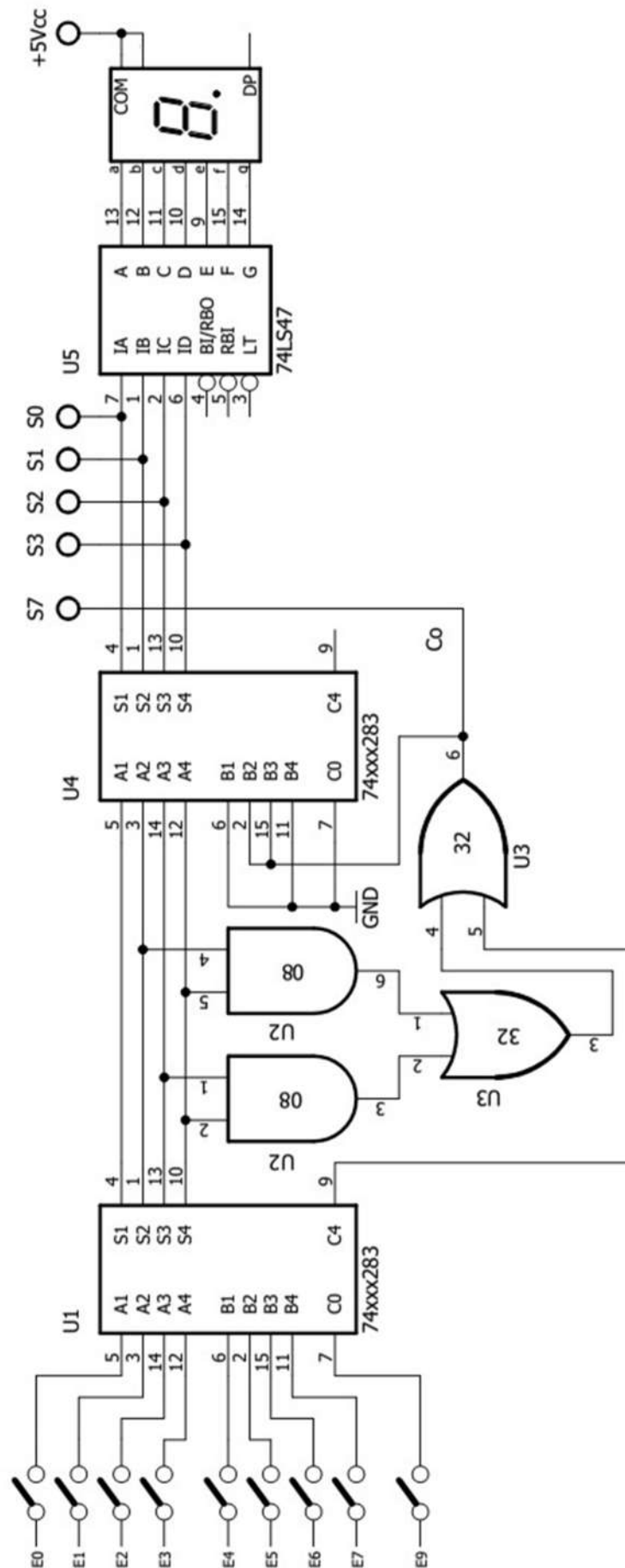
Completa la tabla de la verdad y analiza los resultados con atención.

ENTRADAS									SALIDAS				
Co (E9)	DATO B				DATO A				Co (S7)	SUMA			
	E7	E6	E5	E4	E3	E2	E1	E0		S3	S2	S1	S0
0	0	1	0	0	0	0	1	0					
0	1	0	0	0	0	0	0	1					
0	0	0	0	0	0	1	0	1					
0	0	0	1	1	0	1	1	1					
0	1	0	0	0	1	0	0	0					
0	0	1	1	1	1	0	0	1					
0	0	0	0	0	0	0	0	0					
0	1	1	1	1	1	1	1	1					
1	0	1	0	0	0	0	1	0					
1	1	0	0	0	0	0	0	1					
1	0	0	0	0	0	1	0	1					
1	0	0	1	1	0	1	1	1					
1	1	0	0	0	1	0	0	0					
1	0	1	1	1	1	0	0	1					
1	0	0	0	0	0	0	0	0					
1	1	1	1	1	1	1	1	1					

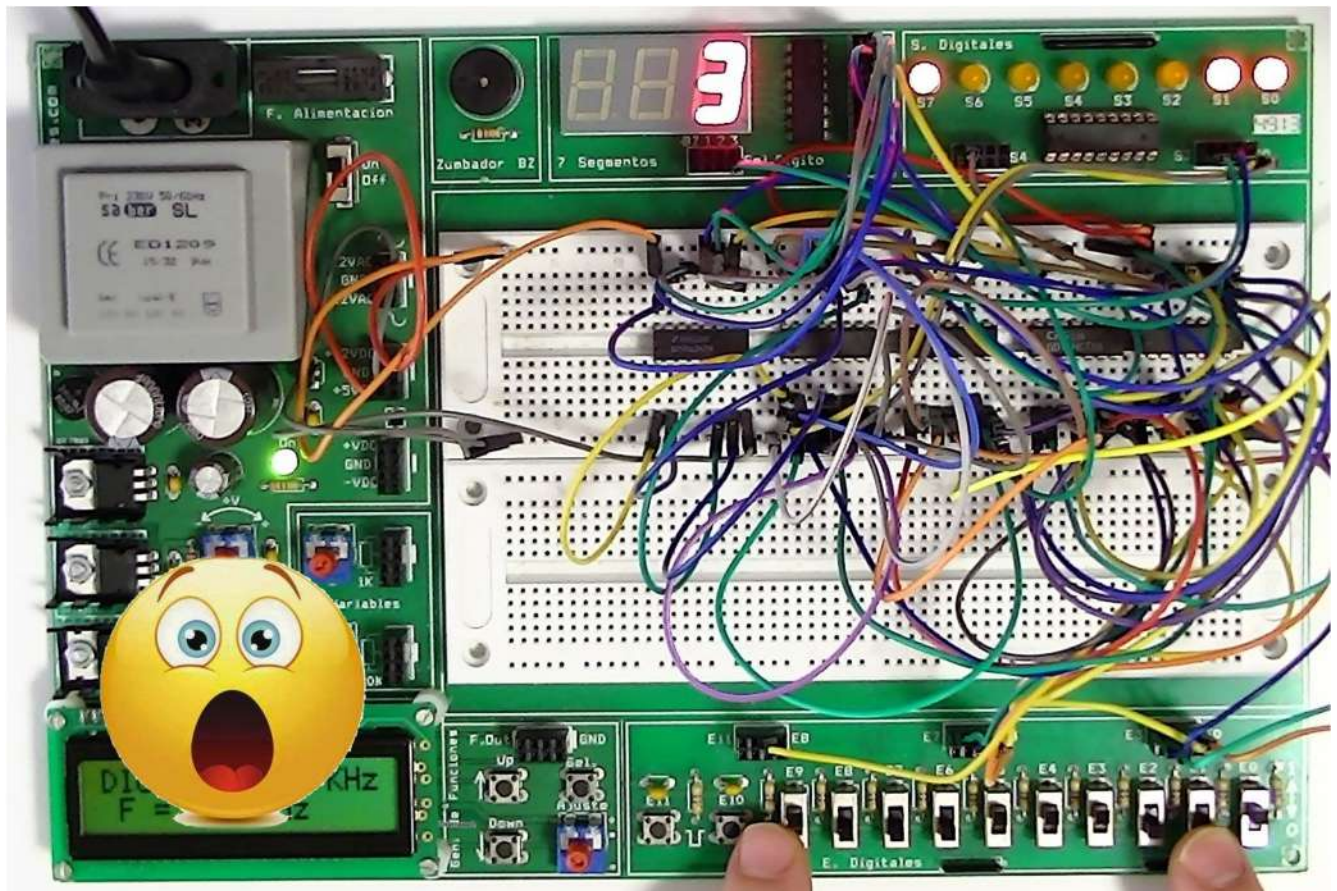
P1-6 Sumador con corrección BCD y visualización

Acabamos esta primera práctica dedicada a los circuitos sumadores con este circuito. Como puedes ver se trata del mismo sumador con corrección BCD y visualización del resultado sobre el display de 7 segmentos.

El circuito, aunque no es difícil de entender, es complicado de montar y te puedes equivocar con facilidad. Te recomiendo que empieces alimentando a cada uno de los cinco circuitos integrados empleados. El montaje merece la pena, animo.



No sirve de mucha ayuda, pero aquí tienes el montaje práctico. En cualquier caso te invito a que completes la tabla de la verdad.



ENTRADAS									SALIDAS					
C0 (E9)	DATO B				DATO A				C4 (S7)	SUMA				DIS- PLAY
	E7	E6	E5	E4	E3	E2	E1	E0		S3	S2	S1	S0	
0	0	0	1	1	0	0	1	0						
0	1	0	0	0	0	0	0	0						
0	0	1	1	1	0	1	0	1						
0	1	0	0	1	1	0	0	0						
0	0	0	1	0	0	1	1	1						
0	0	1	0	1	0	1	0	1						
0	0	1	1	1	0	1	1	1						
1	0	0	1	1	0	0	1	0						
1	0	1	1	1	0	1	0	1						
1	1	0	0	1	1	0	0	0						
1	0	0	1	0	0	1	1	1						
1	0	1	0	1	0	1	0	1						
1	0	1	1	1	0	1	1	1						

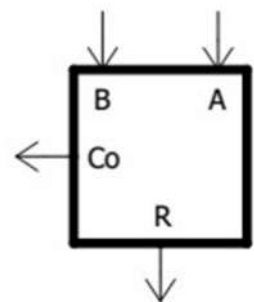
P2: RESTADORES BINARIOS

Son los circuitos capaces de realizar la otra operación de la aritmética binaria, la resta. Reciben dos operandos de n bits, el A que actúa como minuendo y el B que actúa como sustraendo. Nuevamente te recomiendo que repases los algoritmos correspondientes.

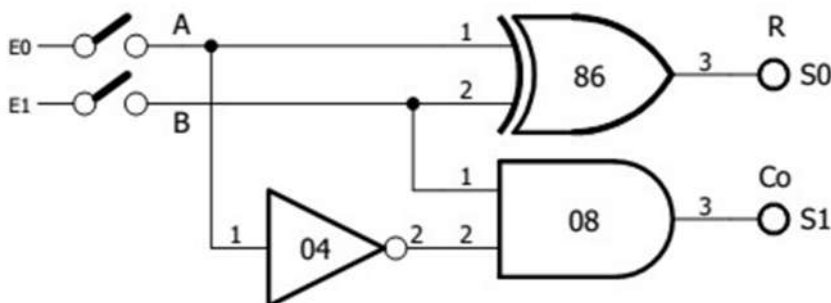
$$\begin{array}{r} A = 1100 \\ B = 0101 \\ \hline R = 0111 \end{array}$$

P2-1 Semirestador

Es el equivalente al semisumador. Resta dos números de 1 bit cada uno, A y B , y produce un resultado de resta R y de salida de llevada o acarreo Co . Se puede simbolizar como en la figura:



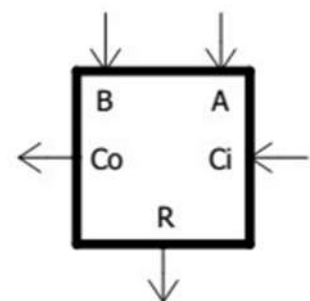
Monta el circuito de la figura y completa su tabla de la verdad verificando que responde a los algoritmos de la resta.



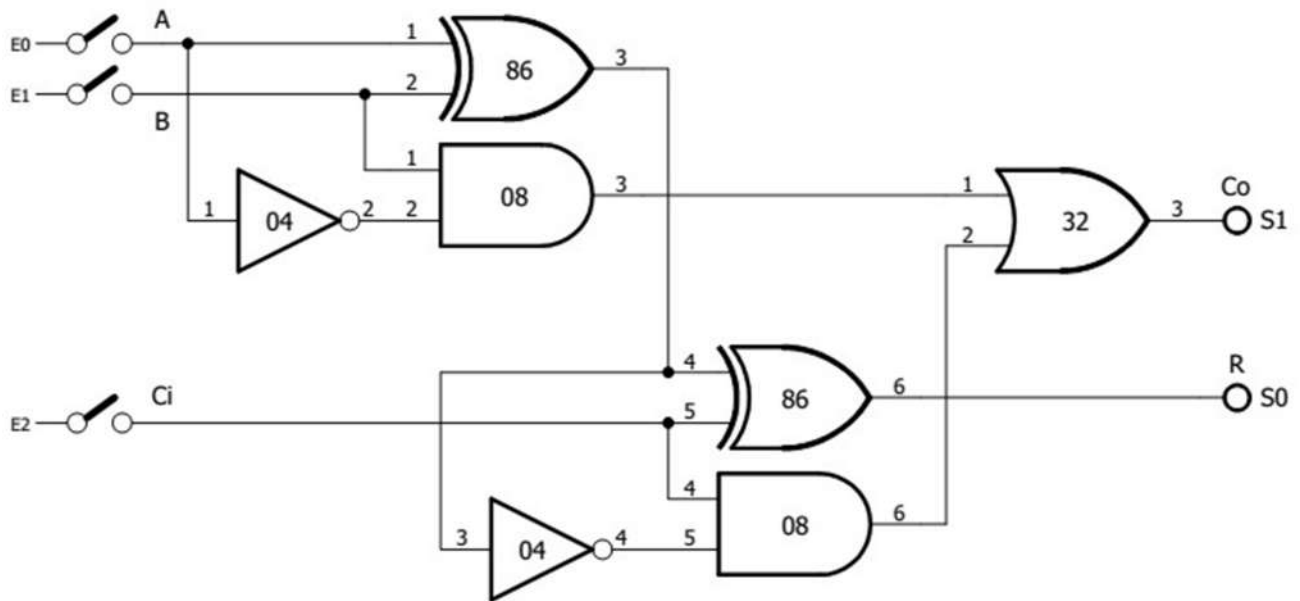
A	B	R	Co
0	0		
0	1		
1	0		
1	1		

P2-2 Restador completo

Al igual que con el sumador completo, el restador completo debe contemplar la entrada de una llevada, Ci , procedente de otros restadores previos. Podríamos simbolizarlo así:



Está compuesto de dos semirestadores. El primero resta $A - B$ y el segundo resta a ese resultado la entrada de llevada previa Ci . Monta el circuito de la figura, completa la tabla de la verdad y verifica que responde a la resta aritmética.



ENTRADAS			SALIDAS	
Ci	A	B	R	Co
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

A partir del restador completo, puedes plantearte el diseño de circuitos restadores con el número de bits que necesites. Basta con que conectes esos restadores completos en cascada, donde la salida de acarreo Co de uno de ellos será la entrada de acarreo Ci del siguiente. Lo mismo hacíamos con los sumadores.

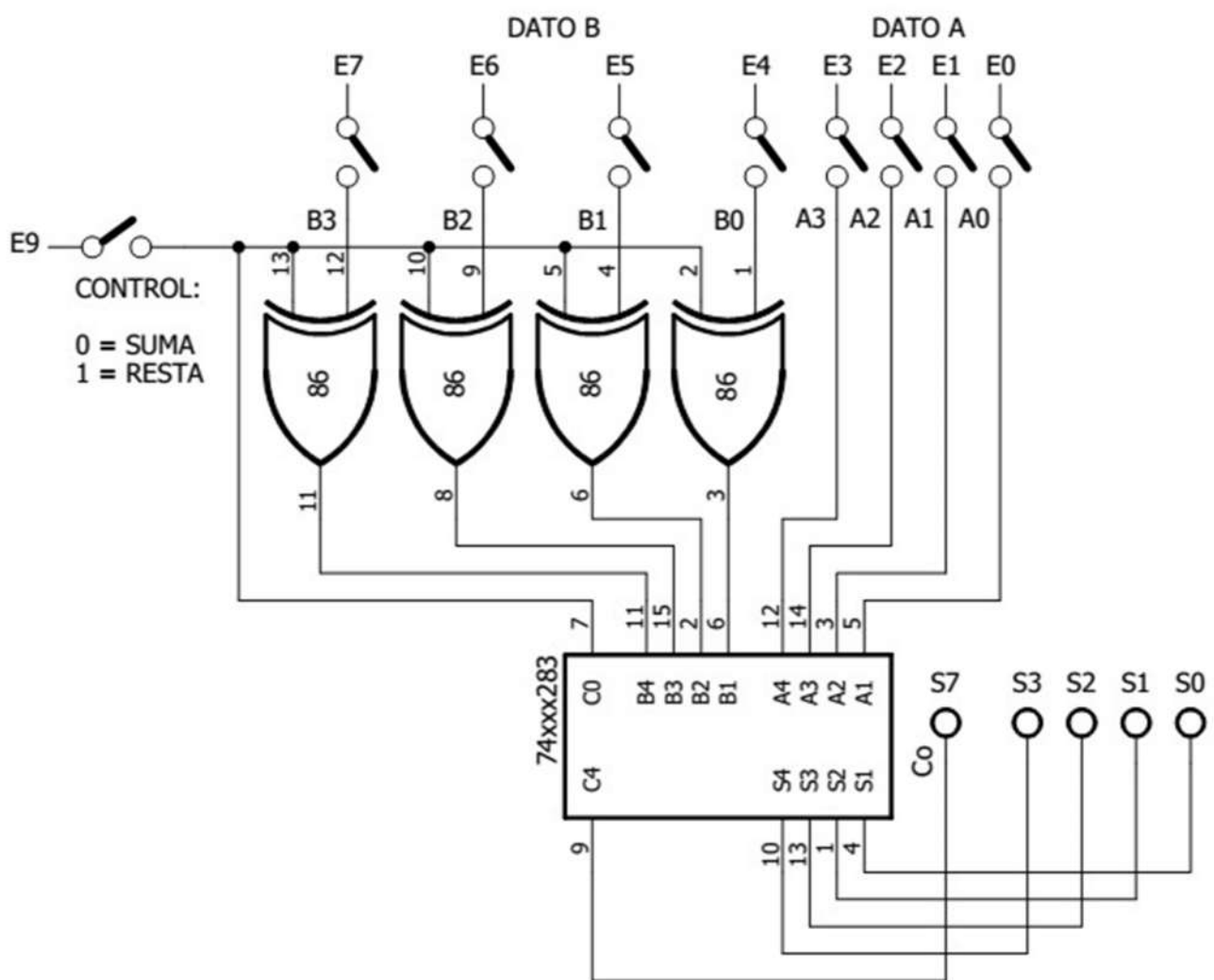
P2-3 Circuito sumador/restador de 4 bits

Si sigues repasando tus conocimientos sobre aritmética binaria quizás recuerdes que mediante la suma de complementos puede calcular la resta de dos números. En binario se suele emplear el complemento a 1 o el complemento a 2. Para calcular un complemento a 1 basta con cambiar los niveles "1" por niveles "0" y viceversa. El complemento a 2 de un número binario se obtiene a partir del complemento a 1 al que se le suma uno.

En la imagen se muestran varias restas resueltas mediante la suma del minuendo con el complemento a 1 del sustraendo y añadiendo uno.

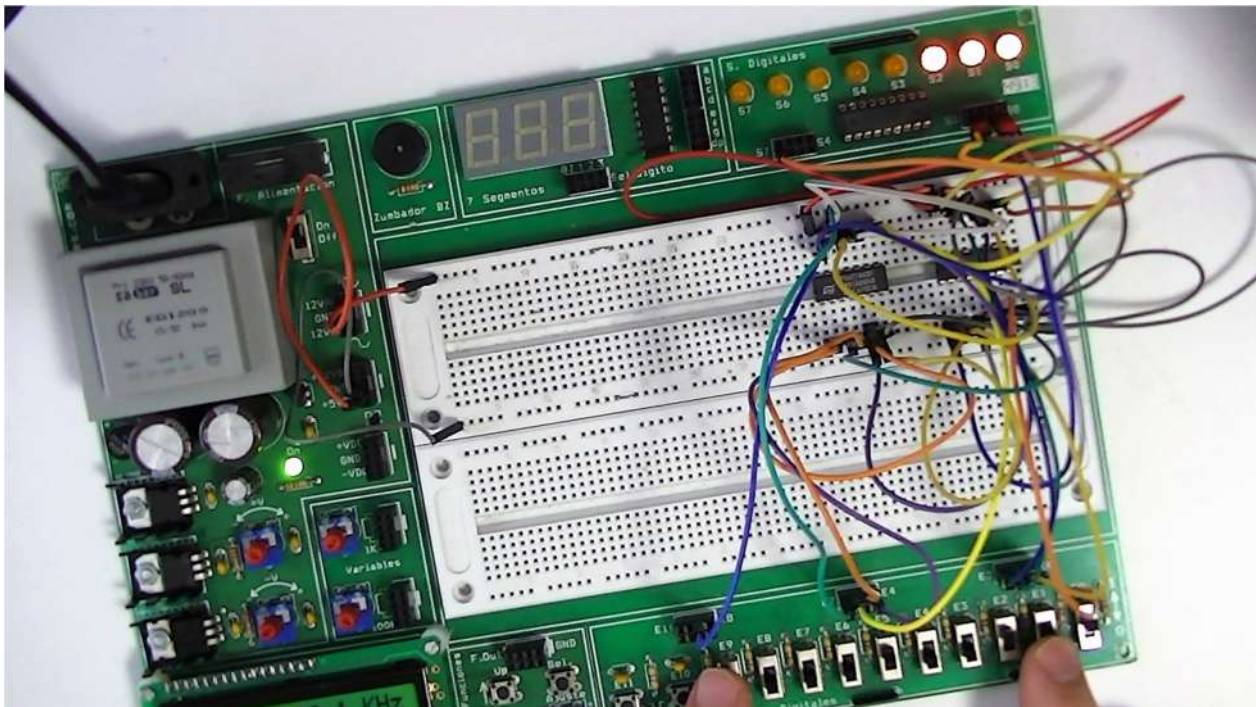
0101	0101	0111	0111	1001	1001	0011	0011
- 0011	+ 1100	-0010	+1101	- 0001	+ 1110	- 0011	+ 1100
0010	1 0001	0101	1 0100	1000	1 0111	0000	1111
	+ 1		+ 1		+ 1		+ 1
	0010		0101		1000		1 0000

Esto nos lleva a pensar que realmente los circuitos restadores no tienen mucho sentido. Nos basta con un circuito sumador y un sistema complementador para realizar restas mediante sumas. Analiza el siguiente circuito y lo montas sobre el entrenador.



Observa que empleamos el mismo circuito sumador 74xx283. Los bits del dato A se introducen directamente. Los del dato B se introducen a través de unas puertas XOR. Si la señal de control, el interruptor E9, está a nivel "0", los bits del dato B entran directamente al sumador. Se produce la suma de $A + B$. Sin embargo, si la señal de

control está a nivel "1", las funciones XOR complementan los bits del dato B. Se realiza la suma de $A + \text{el complemento del dato B}$, obteniendo así la resta. Completa la tabla.



ENTRADAS			SALIDAS	
CONTROL	OPERANDO A	OPERANDO B	RESULTADO	Co
0	0100	0011		
0	0110	0111		
0	1000	1001		
0	0011	0001		
0	1011	0101		
0	1101	1100		
1	0101	0100		
1	0010	0110		
1	1001	0011		
1	0100	0100		
1	0000	0010		
1	0101	1000		

Una observación. La salida de llevada Co se debe interpretar de dos formas. En las sumas, cuando hay llevada, es activa por nivel "1". En las restas al contrario. Cuando hay llevada es activa por nivel "0". Esto también es muy típico en los microprocesadores y los microcontroladores

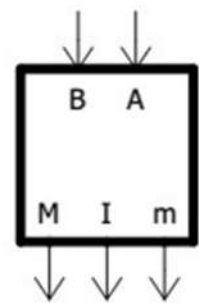
P3: COMPARADORES DE MAGNITUD



Dentro de los circuitos aritmético / lógicos tampoco pueden faltar los comparadores de magnitud. Reciben dos números de n bits, al A y el B, y sus salidas indicarán si $A > B$, o si $A < B$ o si $A = B$.

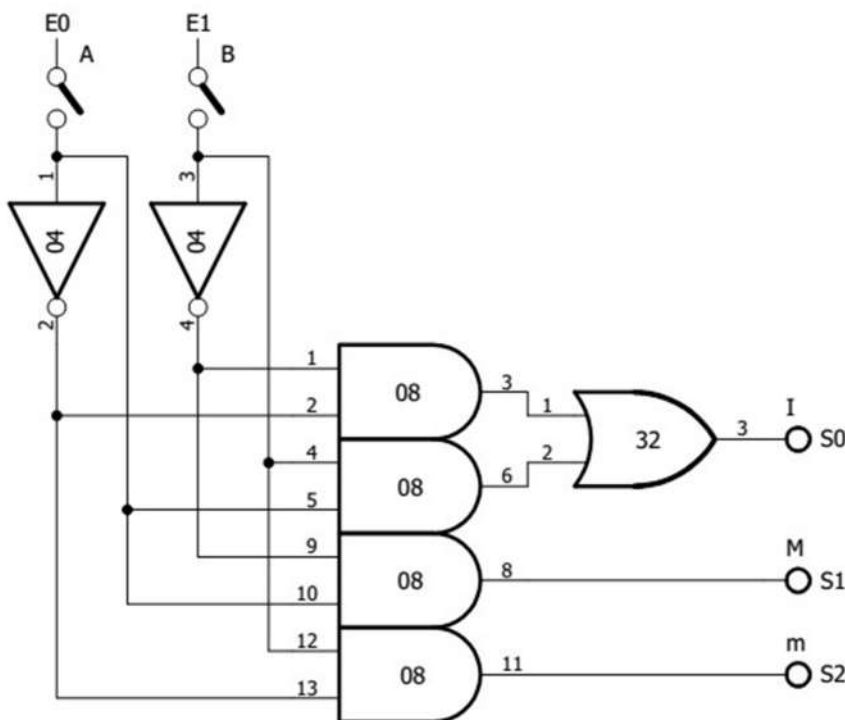
P3-1 Comparador básico de 1 bit

El circuito más sencillo (pero poco útil) sería un comparador de dos números de 1 bit cada uno, el A y el B. Sus salidas representarían si $A > B$ (M), si $A < B$ (m) o si $A = B$ (I) y respondería a las siguientes ecuaciones lógicas:



$$M = \bar{B} \cdot A; m = B \cdot \bar{A}; I = \bar{B} \cdot \bar{A} + B \cdot A$$

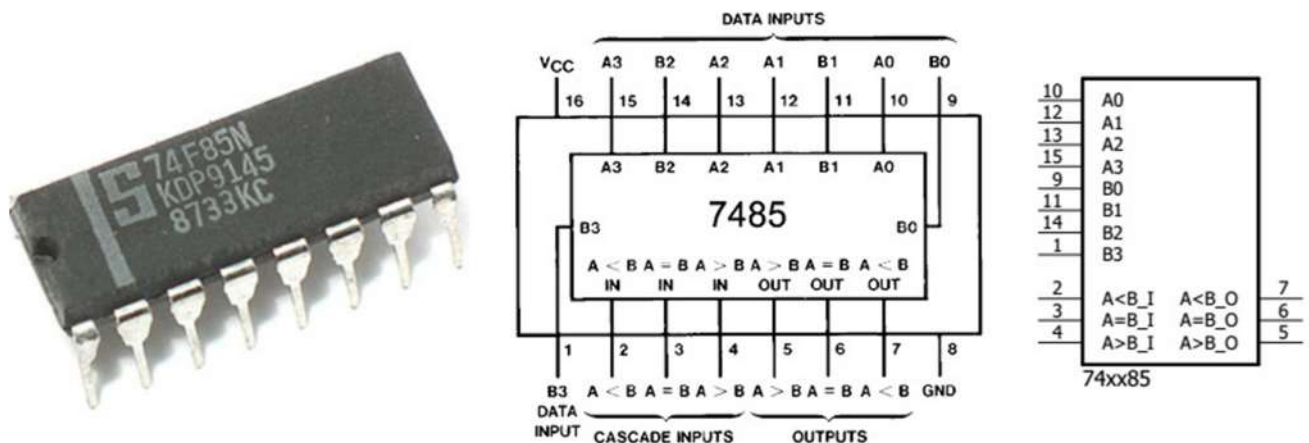
Puedes montar el siguiente circuito experimental que responde a las anteriores ecuaciones, y luego completar la tabla de la verdad.



ENTRADAS		SALIDAS		
B	A	M	I	m
0	0			
0	1			
1	0			
1	1			

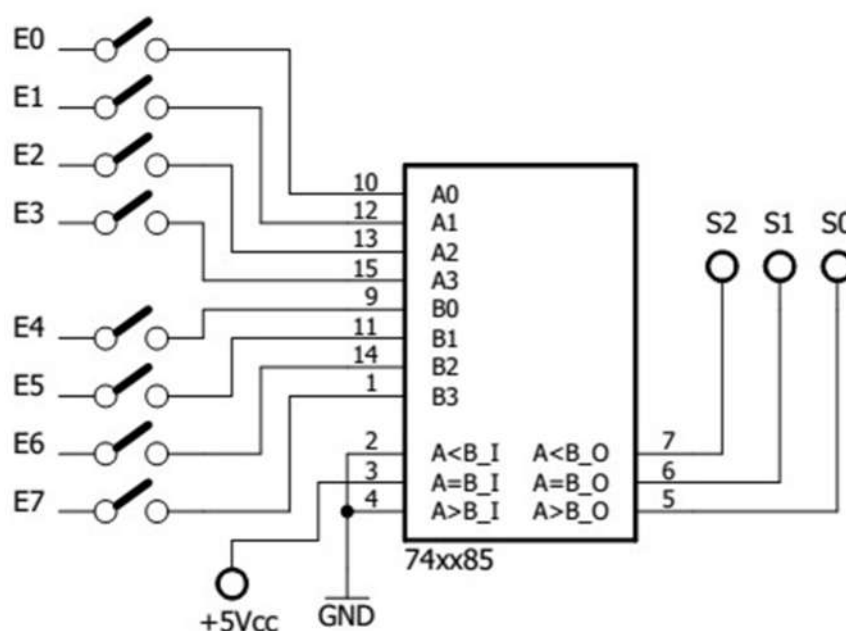
P3-2 Comparador de 4 bits con el 74xx85

Un circuito que compara un bit con otro bit no tiene demasiado interés. Lo ideal sería disponer de circuitos que comparen dos números de como mínimo 4 bits. Es el caso del dispositivo integrado 74xx85 que puedes ver en la imagen.

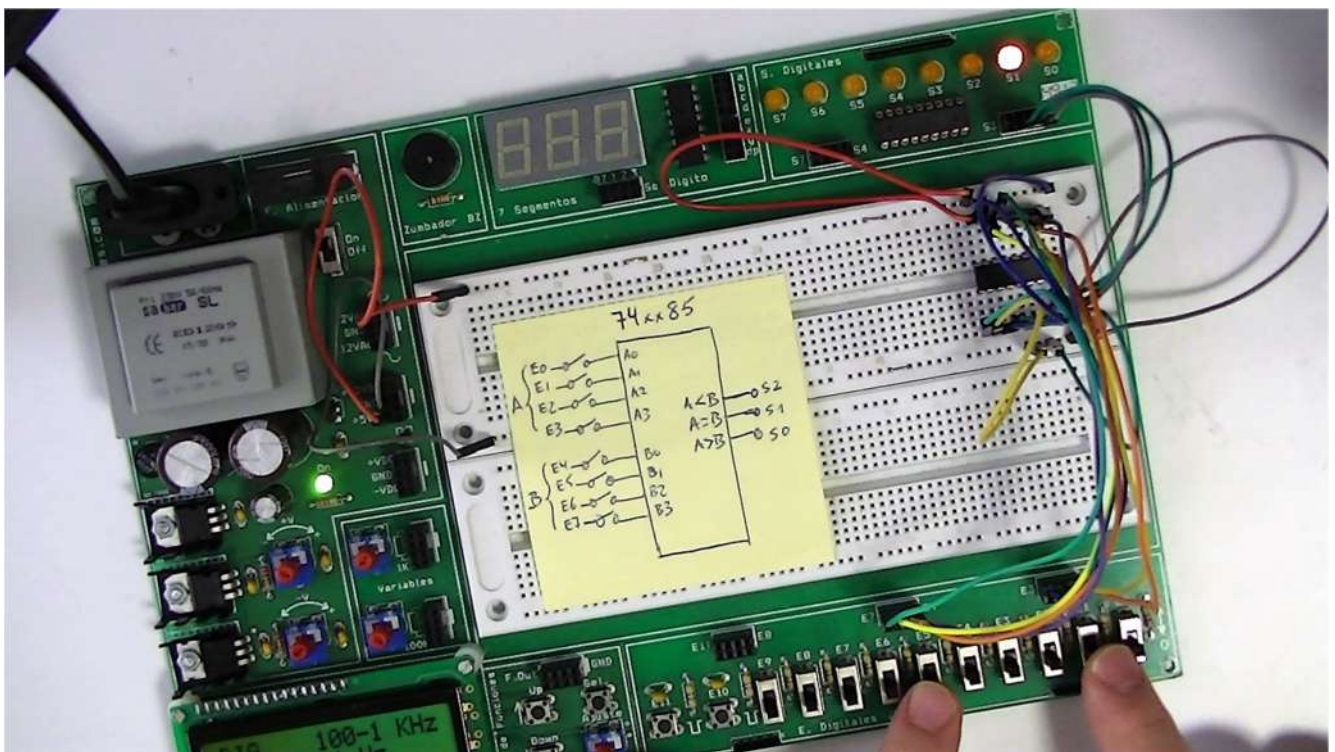


Compara los bits del dato A (A3:A0) con los del dato B (B3:B0) y genera las salidas $A < B_O$, $A > B_O$ y $A = B_O$. También dispone de entradas $A < B_I$, $A > B_I$ y $A = B_I$ que permiten conectar en cascada varios de estos comparadores y poder así comparar magnitudes numéricas todo lo grande que precises.

Te propongo que realices el montaje de la figura y completes la tabla de la verdad.

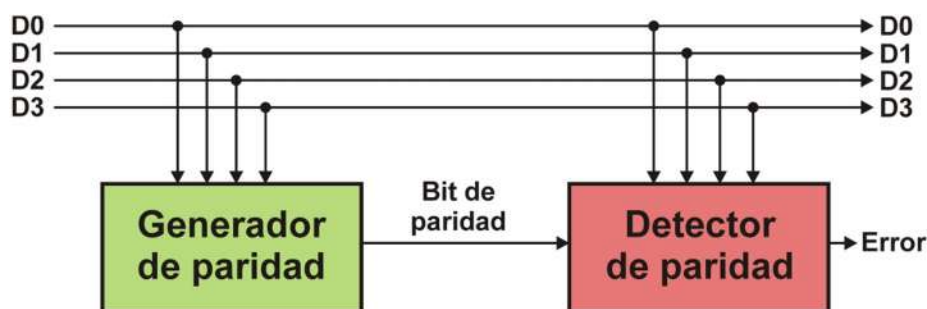


ENTRADAS								SALIDAS		
DATO B				DATO A						
E7	E6	E5	E4	E3	E2	E1	E0	A < B	A = B	A > B
1	0	0	0	0	0	0	0			
0	0	1	1	0	0	1	1			
1	1	0	0	1	1	1	0			
0	0	0	0	0	0	0	0			
1	1	0	0	0	0	1	1			
1	1	1	1	1	1	1	1			
1	0	1	0	0	1	0	1			
0	1	0	1	1	0	1	0			
0	1	1	0	0	1	0	1			
0	0	0	1	0	0	1	0			



P4: GENERADORES/DETECTORES DE PARIDAD

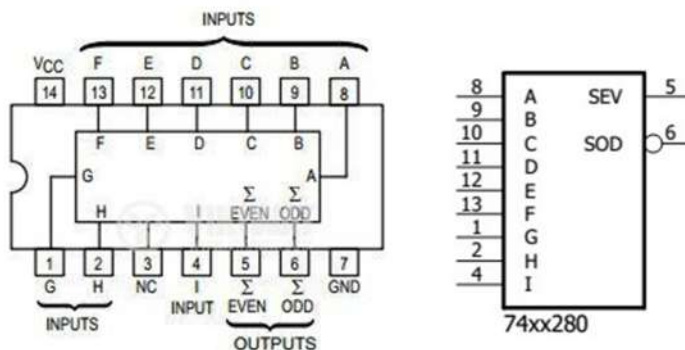
Son circuitos que se emplean en la transferencia de datos para detectar posibles errores en la comunicación. En el esquema se resume un sistema en el que se transmiten 4 bits de datos (D3:D0) más otro de paridad (generador) y se reciben otros tantos bits más otro de paridad (detector).



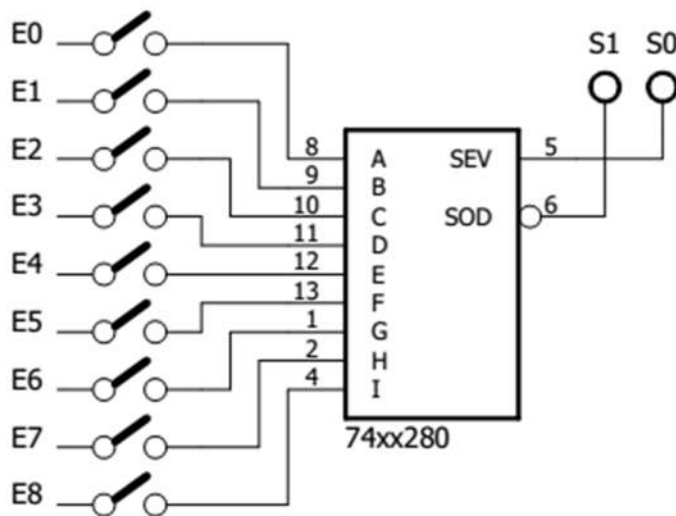
Si se decide trabajar con paridad par, el número TOTAL de niveles "1" transmitidos debe de ser par, al igual que el número TOTAL de bits recibidos. Si se trabaja con paridad impar lo contrario. Tanto el número TOTAL de bits transmitidos como recibidos debe ser impar.

P4-1 Generador/Detector de paridad 9 bits con el 74xxx280

Es un circuito muy sencillo. Tiene 9 bits de entrada, de la A a la I, y genera dos salidas que reflejan si el número de niveles "1" es par (SEV) o impar (SOD). Míralo en la figura.



El montaje del circuito de experimentación es muy sencillo. Lo tienes en el siguiente esquema.



La comprobación es más sencilla aún. Basta con que con los interruptores E0-E8 vayas metiendo niveles lógicos cualesquiera.

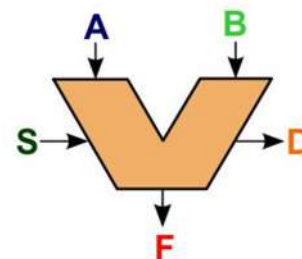
Si el número de niveles "1" de entrada es par se activa la salida S0 (even = par). Si el número de niveles "1" introducidos es impar se activa la salida S1 (odd = impar)

Tú mismo puedes deducir las salidas a partir de los bits de entrada completando esta tabla de la verdad.

ENTRADAS									SALIDAS	
I	H	G	F	E	D	C	B	A	S1	S0
1	0	0	0	0	1	0	1	0		
0	1	1	0	1	0	1	0	0		
0	0	0	0	0	0	0	0	0		
1	1	1	1	1	1	1	1	1		
1	0	1	0	1	0	1	0	1		
0	0	0	1	1	0	1	0	1		
1	1	1	1	1	0	0	0	1		
0	0	0	1	0	0	0	0	0		
1	0	0	0	0	0	0	0	1		
0	0	1	1	1	1	0	0	0		
0	0	0	1	1	1	1	1	1		
0	1	0	0	0	0	1	0	1		
1	1	1	0	0	0	0	1	0		
0	0	0	1	0	1	0	0	0		
1	1	1	1	0	1	0	0	0		
0	0	0	1	1	1	0	0	0		

P5: UNIDAD ARITMÉTICO / LÓGICA; LA ALU

Todos los circuitos aritmético / lógicos con los que acabas de experimentar, y alguno más, se pueden encontrar agrupados en un único dispositivo conocido como ALU (Unidad Aritmético Lógica). Se puede simbolizar como en la figura.

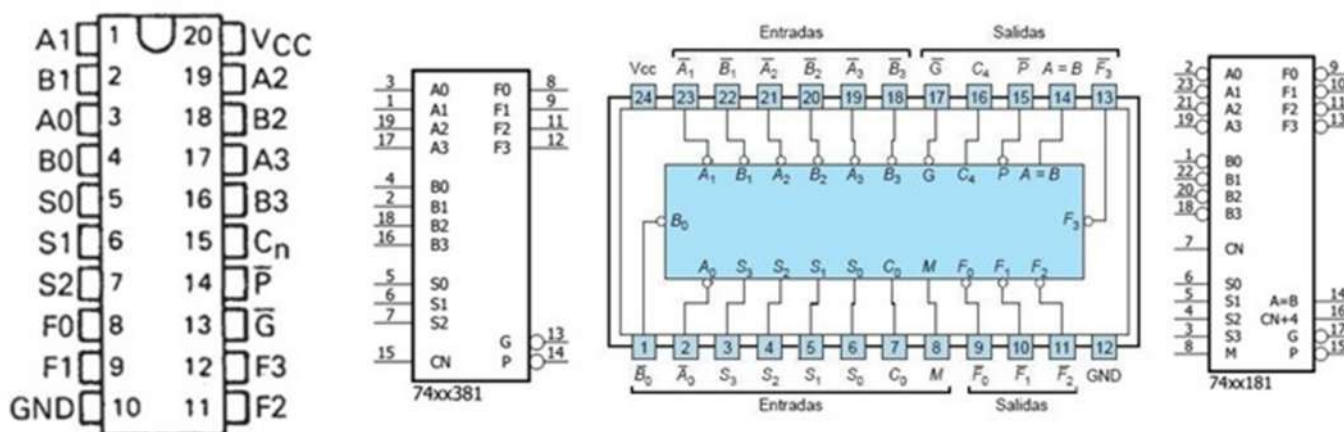


Las entradas **A** y **B** representan los operandos o datos con los que vamos a operar. Si decimos que una ALU es de 4 bits es porque esos datos también serán de 4 bits. Las ALU's serán de 4 bits o múltiplos de 4.

Las señales **D** representan señales auxiliares como pueden ser la entrada de llevada procedente de otras ALU's previas, salida de llevada, salidas de igualdad, mayor qué, menor qué, etc... Las salidas **F** muestran el resultado de la operación realizada entre los datos A y B. Habrá tantas señales F de salida como bits sea la ALU.

Por último, en las señales de entrada **S**, se encuentra el "kid de la cuestión". Decíamos al principio que la ALU integra la capacidad de realizar diferentes funciones aritmético/lógicas. Pues bien, mediante estas entradas podrás seleccionar qué operación quieres ejecutar en un momento dado. A cada una de esas operaciones le corresponde un código binario único para su selección y que tú debes de aplicar con estas señales.

Te puedo dar las referencias de un par de modelos de ALU's comerciales como pueden ser la 74LS381 y/o la 74LS181. Sus encapsulados y patillaje los puedes ver en la figura:



En la red puedes encontrar abundante información de las mismas, sin embargo adquirirlas no es tan fácil. En todos los fabricantes que hemos consultado nos indican que son dispositivos obsoletos. Los hemos encontrado en tiendas del tipo Amazon, eBay, Alibaba, Aliexpress y similares, pero o sus precios son elevados, o no hay garantías de un suministro continuo. Por eso no los proponemos para las prácticas de este curso.

P5-1 Nuestra ALU: MKE-3.0

Sin embargo consideramos que experimentar con una ALU y ejecutar las diferentes funciones que integra, es importante en un curso práctico de Electrónica Digital como es este. Esto nos ha llevado a crear nuestra propia ALU didáctica que nos permita experimentar con ella, la MKE-3.0.

En realidad el circuito integrado consiste en un microcontrolador al que le hemos programado para que ejecute una serie de funciones emulando el funcionamiento de una completa ALU de 4 bits.



Te puedes olvidar de que se trata de un microcontrolador. Para ti es un circuito integrado como cualquier otro al que le tienes que conectar las entradas del operando A, las del operando B, las de selección de función y las salidas con el resultado. Es como si hubiésemos diseñado un circuito integrado "a medida".

Así, en "petit comite", te puedo adelantar que aprovechándonos de las infinitas posibilidades que ofrece un microcontrolador, vamos a utilizar este mismo circuito integrado para hacer también algunas funciones analógicas. Todo a su debido tiempo.

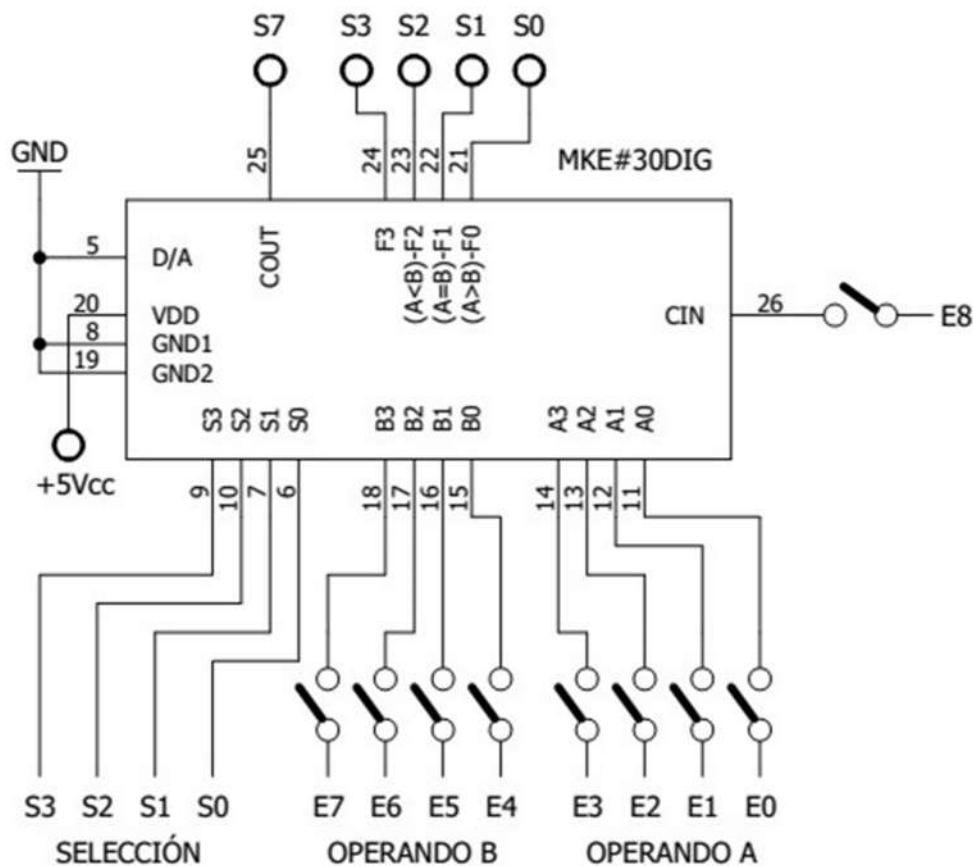
En la tabla adjunta tienes una descripción del patillaje...

NOMBRE	Nº Patilla	DESCRIPCIÓN
A3, A2, A1, A0	14, 13, 12, 11	Los 4 bits del dato A
B3, B2, B1, B0	18, 17, 16, 15	Los 4 bits del dato B
S3, S2, S1, S0	9, 10, 7, 6	Entradas para seleccionar la función a ejecutar
D/A	5	Selección de modo digital o analógico
F3, F2, F1, F0	24, 23, 22, 21	Salidas del resultado de la función seleccionada
A<B, A=B, A>B	23, 22, 21	Salidas del comparador: menor que, igual y menor que
COUT	25	Salida de acarreo en operaciones aritméticas. En las sumas activo por "1" y en las restas activo por "0".
CIN	26	Entrada de acarreo previo en operaciones aritméticas. En las sumas activo por "1" y en las restas activo por "0".
VDD, GND	20, 8, 19	Entradas de tensión de alimentación

Y en esta otra se describe cada una de las funciones aritmético/lógicas que es capaz de ejecutar según las entradas de selección.

D/A	SELECCIÓN S3 S2 S1 S0	Nº	DESCRIPCIÓN
1	xxxx	x	Modo analógico. No usar de momento.
0	0000	0	Borrar todas las salidas: F3:F0 = 0000
0	0001	1	Activar todas las salidas: F3:F0 = 1111
0	0010	2	Resta B - A - Cin; F3:F0 = (B3:B0) - (A3:A0) - Cin
0	0011	3	Resta A - B - Cin; F3:F0 = (A3:A0) - (B3:B0) - Cin
0	0100	4	Suma A + B + Cin; F3:F0 = (A3:A0) + (B3:B0) + Cin
0	0101	5	Suma A + A + Cin; F3:F0 = (A3:A0) + (A3:A0) + Cin
0	0110	6	Suma B + B + Cin; F3:F0 = (B3:B0) + (B3:B0) + Cin
0	0111	7	Compara A con B con salidas por: A > B, A = B y A < B
0	1000	8	Salida del dato A; F3:F0 = (A3:A0) (función SI de A)
0	1001	9	Salida del dato B; F3:F0 = (B3:B0) (función SI de B)
0	1010	10	Salida del complemento de A; F3:F0 = $\overline{(A3:A0)}$ (NOT A)
0	1011	11	Salida del complemento de B; F3:F0 = $\overline{(B3:B0)}$ (NOT B)
0	1100	12	Función AND entre A y B; F3:F0 = (A3:A0) & (B3:B0)
0	1101	13	Función OR entre A y B; F3:F0 = (A3:A0) + (B3:B0)
0	1110	14	Función XOR entre A y B; F3:F0 = (A3:A0) $\oplus$ (B3:B0)
0	1111	15	Función XNOR entre A y B; F3:F0 = $\overline{(A3:A0) \oplus (B3:B0)}$

El montaje práctico es el que tienes en la imagen.



Con los interruptores E3:E0 introducimos el valor del operando A (A3:A0) y con los interruptores E7:E4 los del operando B (B3:B0). El interruptor E8 se conecta con la entrada CIN para simular entradas de llevadas previas.

Los leds S3:S0 se conectan con las salidas del resultado F3:F0 de la función ejecutada. S7 se conecta con la salida COUT que representa una llevada de salida si la hubiera.

Como en el entrenador ya no quedan interruptores, proponemos que las entradas de selección S3:S0 las conectes directamente a GND (nivel "0") o a +5Vcc (nivel "1") según la combinación binaria que quieras formar para ejecutar una determinada operación.

Te hago notar que la señal D/A, la patilla 5, la hemos conectado a GND o nivel "0". El circuito se configura como una ALU digital. Más adelante pondrás esta patilla a +5Vcc, nivel "1", con lo que el mismo circuito quedará configurado como analógico.

[illegible]

Esta es la imagen del montaje de la ALU sobre el entrenador Universal Trainer, y con esto acabamos el área de prácticas del tema 6.

