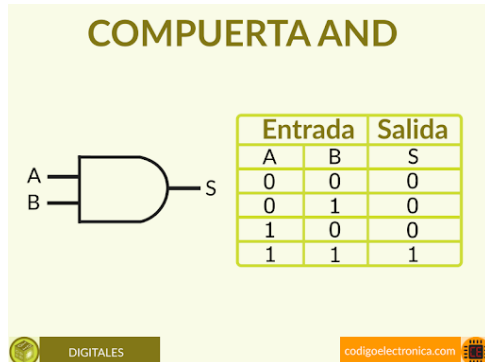


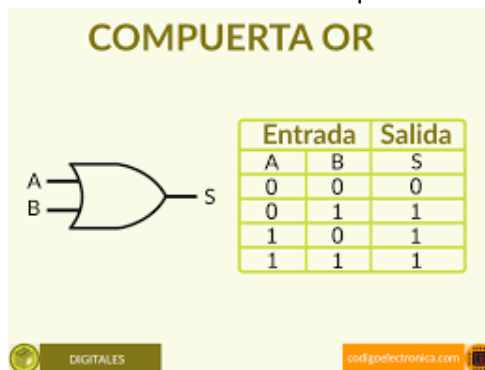
En la primera clase presencial se repasaron los conceptos de compuertas lógicas, la AND, la OR y la NOT. Ahora se establecerán relaciones entre estas compuertas básicas y las condiciones que puede tener un programa a la hora de su desarrollo.

Comenzamos por la compuerta AND:



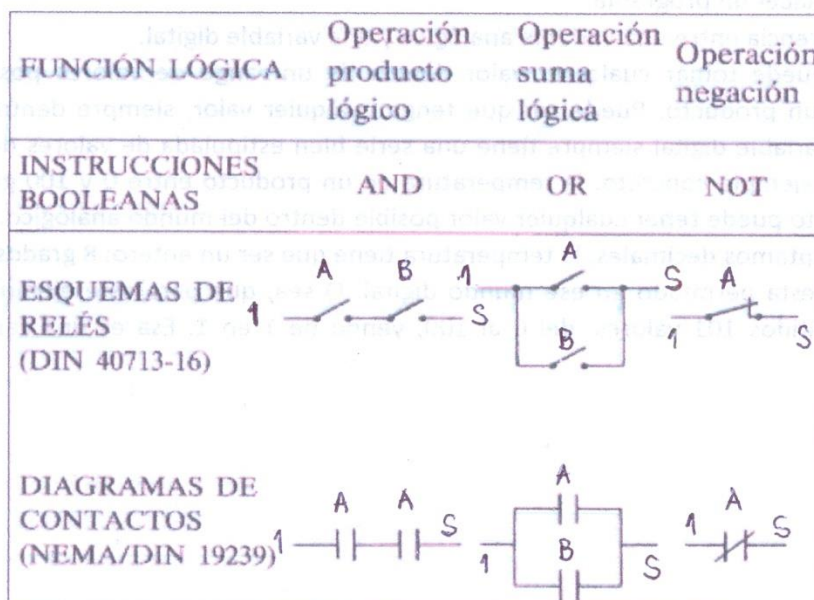
Por lo que se puede ver, es necesario que A y B sean 1 para que S sea 1, dado que en caso contrario, S es 0. O sea que dicho de otra manera, deben cumplirse las dos condiciones simultáneamente: A y B, para que S sea 1.

Veamos ahora la compuerta OR:



Aquí, es suficiente que A o que B sean 1, para que la salida S sea 1. O sea que, con que cualquiera de las dos sea 1, la salida S ya dará 1.

Veremos a continuación otra manera de representar esta idea:



En el caso de la compuerta AND, podemos observar que para que el 1 que está a la izquierda, llegue a S, es necesario que A esté cerrada (se la observa abierta en el esquema) y que B esté cerrada. En esa situación, el 1 llega a S. A y B están en serie, es decir, una a continuación de la otra.

En el caso de la compuerta OR, podemos observar que para que el 1 llegue a S, basta con que A o B estén alguna de las dos (o las dos) cerradas. El 1 tiene dos caminos para llegar a S. A y B están aquí en paralelo.

Resumiendo: cuando una serie de condiciones deben cumplirse en conjunto (todas simultáneamente), deberán estar en serie. Cuando en una serie de condiciones, basta con que se cumpla una para el sistema funcione, deberán ir en paralelo. Lo dicho vale para cualquier cantidad de señales presentes. Las combinaciones que pueden presentarse son ilimitadas.

Veamos un ejemplo de tabla para resolver a modo de ejercicio (que deberá presentarse el próximo viernes 24-ABR-2020):

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

Resolver el ejemplo de cualquier manera, aplicando los conocimientos que se recuerden y hacer el esquema con contactos, o esquema de relés, tal como se muestra en la figura anterior.

NOTA: se pueden consultar unos con otros, pero si no se aprende a poner al cerebro en marcha, no estará en funcionamiento cuando deban resolver algo por ustedes mismos.

Vayamos al esquema de contactos de la compuerta AND. Cada uno de los contactos, A y B se ven normalmente abiertos. O sea en condición de 0 lógico en condiciones normales. Por lo tanto el contacto cerrado, representa en este caso una situación de 1 lógico. Vale esta misma explicación para el caso de la compuerta OR.

Algo más complicado de comprender es la simbología de la compuerta NOT. En una situación normal (0 lógico) el 1 presente en la entrada, llega a la salida. Cuando A está en 1 (está abierto), el 1 de la entrada no alcanza la salida S. O sea que las condiciones son una cosa, y el resultado otra. Esto debe tenerse muy en claro a la hora de hacer un programa.

Por último, veamos la diferencia entre una variable analógica y una variable digital.

Una variable analógica, puede tomar cualquier valor dentro de un rango de valores posibles. Supongamos, la temperatura de un producto. Puede ser que tenga cualquier valor, siempre dentro del rango aceptado. En cambio una variable digital siempre tiene una serie bien estipulada de valores dentro de ese mismo rango. Veamos un ejemplo concreto: la temperatura de un producto entre 0 y 100 grados centígrados. Realmente, el producto puede tener cualquier valor posible dentro del mundo analógico. Pero si en nuestro mundo digital no aceptamos decimales, la temperatura tiene que ser un entero: 8 grados o 25 grados, sin decimales. El xx,x no está permitido en ese mundo digital. O sea, que para este ejemplo en particular, solamente están permitidos 101 valores, del 0 al 100, yendo de 1 en 1. Esa es una variable digital.