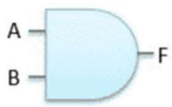
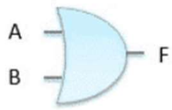
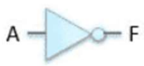
CAPÍTULO 2

ALGEBRA DE BOOLE, LÓGICA Y CIRCUITOS DIGITALES

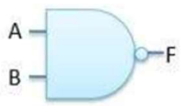
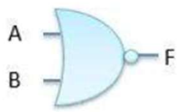
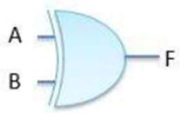
1. Puertas Lógicas

Las puertas lógicas más usadas son las puertas NOT, OR y AND. En la siguiente tabla se muestra su nombre, el símbolo más comúnmente empleado en la representación de circuitos, la notación algebraica o fórmula lógica y finalmente la tabla de verdad.

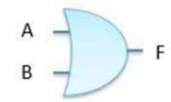
Las puertas OR y AND podrían tener más de dos entradas. No debemos olvidar que las operaciones que realizan son las que vimos en la unidad anterior como "operaciones del álgebra de Boole" . Si tenemos esto en cuenta podemos concluir que las puertas lógicas tienen por tanto las mismas propiedades vistas del Algebra de Boole.

Nombre	Símbolo	Operación	Tabla de verdad
AND		$F = A \bullet B$ ----------- $F = AB$ ----------- F = A AND B	A B F 0 0 0 0 1 0 1 0 0 1 1 1
OR		$F = A + B$ ----------- F = A OR B	A B F 0 0 0 0 1 1 1 0 1 1 1 1
NOT		$F = \bar{A}$ ----------- F = NOT(A)	A F 0 1 1 0

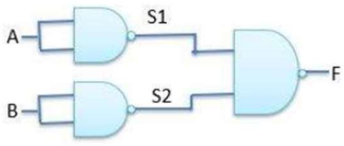
Además de las puertas lógicas vistas en el apartado anterior existen las puertas: NOR (que es la negación de la OR), NAND (que es la negación de AND), XOR (que es la función OR exclusivo el cual toma el valor 1 cuando las entradas son distintas). Es conveniente fijarse que por ejemplo el símbolo del NAND es como el del AND + "la bolita" del NOT.

Nombre	Símbolo	Operación	Tabla de verdad
NAND		$F = \overline{(A \cdot B)}$ ----------- F = A NAND B	A B F 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$F = \overline{(A + B)}$ ----------- F = A NOR B	A B F 0 0 1 0 1 0 1 0 0 1 1 0
XOR		$F = (A \oplus B)$ ----------- $F = A \cdot \bar{B} + \bar{A} \cdot B$ ----------- F = A XOR B	A B F 0 0 0 0 1 1 1 0 1 1 1 0

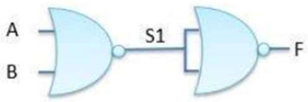
La agrupación y el encadenado de puertas lógicas da lugar a la generación de los circuitos electrónicos complejos que conforman un computador. Es incluso posible implementar el funcionamiento natural de unas puertas lógicas empleando otras. Por ejemplo: vamos a ver cómo podríamos implementar el funcionamiento de una puerta OR sólo con puertas NAND y por otro lado sólo con puertas NOR:



A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



A	B	S1= A NAND A	S2=B NAND B	F= S1 NAND S2
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1



A	B	S1= A NOR B	F=S1 NOR S1
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

2. Circuitos combinacionales

Los circuitos combinacionales son aquellos que se componen de un conjunto de puertas interconectadas cuya salida depende del valor de las entradas. Llamamos ínfimo retardo a aquel causado por los retardos en las puertas lógicas del circuito. Definimos un circuito combinacional mediante su tabla de verdad, su símbolo y su función booleana.

Implementación en su tabla de verdad:

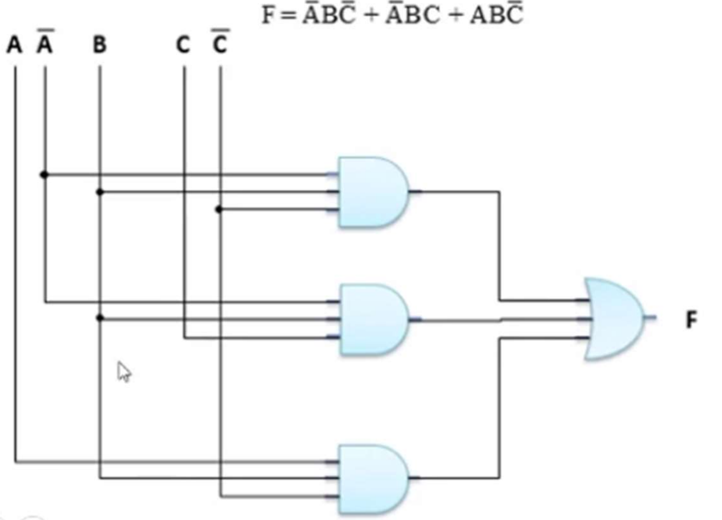
Se buscan en la tabla las combinaciones que dan F = 1.

Cada combinación se separa mediante una suma para escribir la **función booleana.**

$$F = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$$F = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$$



La complejidad de un circuito viene dada por la cantidad de puertas y conexiones que lo compongan. Hay que simplificar la función booleana, para así simplificar el circuito que la codifica para obtener un circuito equivalente, pero más sencillo. Para ello usaremos mapas de Karnaugh y el método de simplificación algebraica.

Primeramente, debemos aclarar que la simplificación de una fórmula booleana por mapas de Karnaugh se puede realizar de dos formas:

- Minimización de una suma de productos.
- Minimización de un producto de sumas.

Diremos que nuestra formula es una suma de productos si cada uno de sus términos, están formados exclusivamente por productos y se relacionan por la operación suma.

$$F = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$$

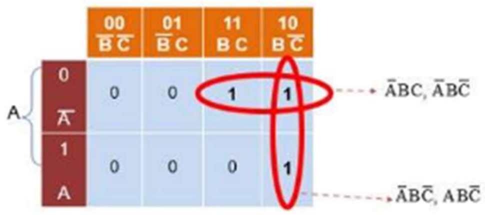
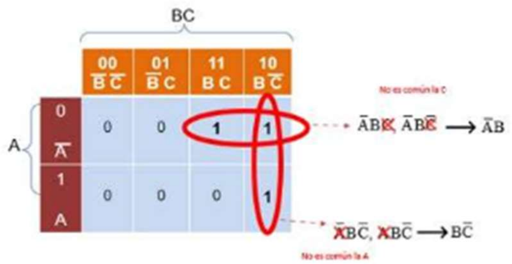


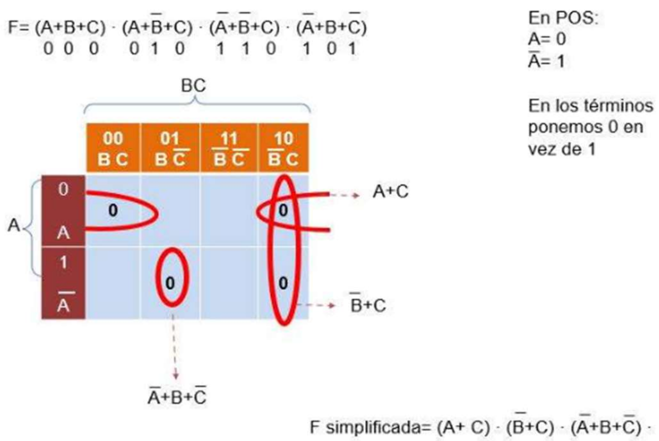
Imagen 1



Por lo tanto la función Booleana simplificada equivalente a la dada sería

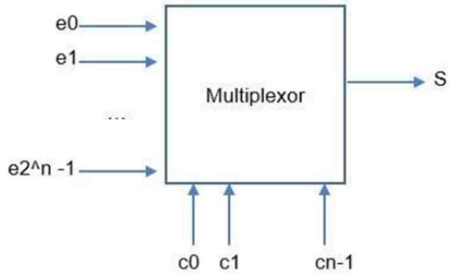
$$F = \bar{A}B + B\bar{C}$$

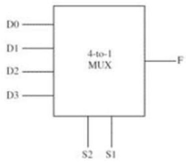
Diremos que nuestra fórmula booleana es un producto de sumas si cada uno de sus términos están formados exclusivamente por sumas y éstos a su vez se relacionan entre ellos mediante la operación de producto. El producto de sumas se conoce como POS (por su abreviación en inglés de "Product of sums").



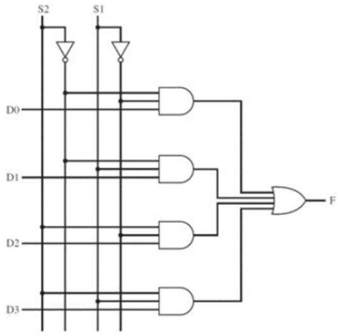
Circuitos Combinacionales típicos

Multiplexor: Circuito que conecta varias entradas con una única salida y un mecanismo de selección determina que entrada pasa a la salida.



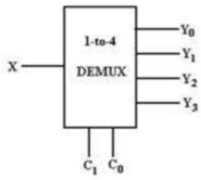


S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

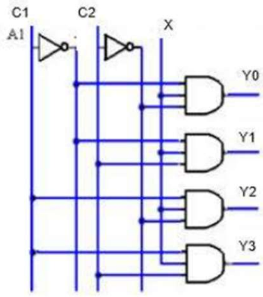


Se usan en circuitos digitales para enrutamiento de señales, existe un registro cuyo valor puede aumentar debido a 4 causas, siendo el MUX 4 a 1 el que elige en cada instante cual de estas causas es la que modificará el registro.

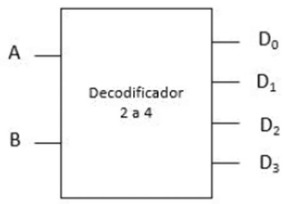
Demultiplexor: Es el inverso del multiplexor conecta varias salidas con una única entrada en un instante dado el mecanismo de selección determina que salida se activa. Usándose normalmente como enrutador de señales.



C_0	C_1	Y_0	Y_1	Y_2	Y_3
0	0	X	0	0	0
0	1	0	X	0	0
1	0	0	0	X	0
1	1	0	0	0	X

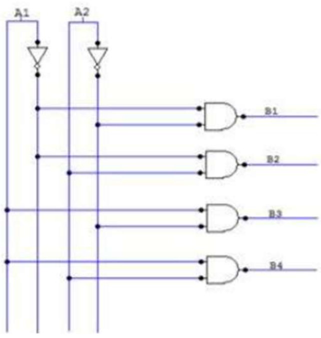


Decodificador: Es un circuito que dispone de n señales de entrada y 2n señales de salida, en un instante dado y dependido del valor de las entradas solo una de las salidas es seleccionada.



A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Uno de sus usos mas frecuentes es para decodificar direcciones, hay que elegir uno de los cuatro módulos de memoria para almacenar un dato, siendo el modulo de destino el que se codifica en las entradas A y B del decodificado, que transforma este código en la dirección del módulo destino correcto.



El decodificador funciona como un demultiplexor si contamos los selectores como una entrada.

Sumador: La suma binaria funciona de la misma manera que la decimal, el acarreo ("Carry" en inglés) es lo que aprendimos en primaria como "lo que me llevo" .

0 + 0 = 0

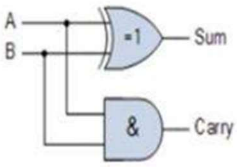
0 + 1 = 1

1 +1 = 10 (el 1 sale del acarreo)

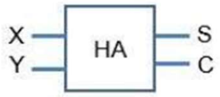
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$C = x \cdot y$

$S = x \cdot \underline{y} \; \underline{+ \; x} \cdot \underline{y} = X \text{ XOR } Y = x \oplus y$



De forma simplificada se representa como

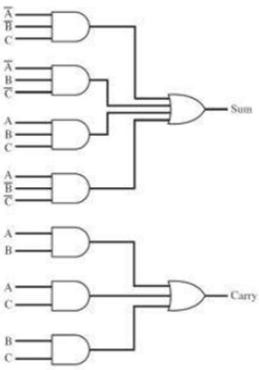


Half adder

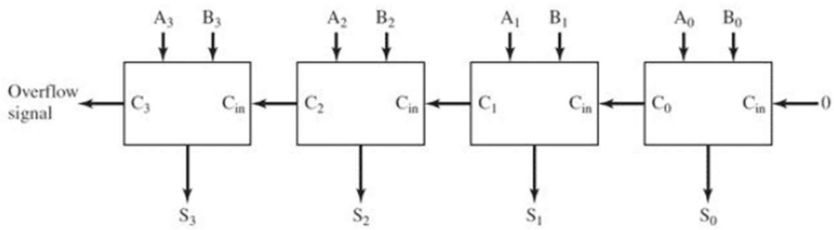
Podemos definir la sima binaria de dos bits como dos funciones de las que llamamos suma y carry pudiendo así construir la semi suma como se ve en la imagen.

$\text{Suma} = \boxed{\bar{A}\bar{B}C} + \boxed{\bar{A}B\bar{C}} + \boxed{ABC} + \boxed{A\bar{B}\bar{C}}$

$C_{out} = AB + AC + BC$



C_{in}	A	B	Suma	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

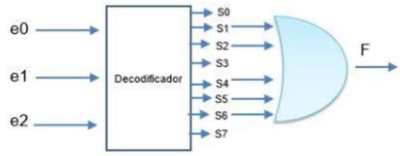


Cualquier función de n variables puede implementarse como la salida de un decodificador de n a 2n y puertas OR. Además, a veces resulta más económico construir una función con decodificadores que hacerlo directamente con puertas.

Ejemplo:

A partir de un decodificador de 3 entradas generar la función definida por la siguiente tabka de verdad. Equivale a decir que F =∑ (M1, M2, M4, M5 , M6) Siendo MX la salida x del decodificador. De la misma forma que acabamos de ver con decodificadores, con multiplexores se pueden también construir funciones lógicas. L diferencia es que para generar la función no vamos a necesitar añadir puertas lógicas.

e2	e1	e0	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



3. Circuitos Secuenciales

Un circuito secuencial es aquel en el que sus salidas dependen del estado actual de sus entradas y de sus estados anteriores. En los circuitos combinacionales, la única noción temporal que interviene es el presente. En cambio, en los circuitos secuenciales se tiene en cuenta la evolución temporal de las señales.

Esto implica que para valores idénticos en la entrada se podrían obtener valores distintos en la salida según el momento en el que estemos.

Biestables

Son circuitos que son capaces de retener información como si fueran unidades de memoria que guardan su ultimo estado indefinidamente hasta que sus entradas cambien.

- Es un circuito con dos estados. Dependiendo del valor de sus entradas, cambia de un estado a otro. En ausencia de entradas, el circuito permanece en el último estado alcanzado (memoria de un bit).
- Tiene dos salidas complementarias: Q y NOT(Q)).
- Son circuitos dependientes del tiempo (t).
- El estado actual se denomina Qt.
- El estado siguiente se denomina Qt+1.

El cerrojo (latch) S-R (Set – Reset)

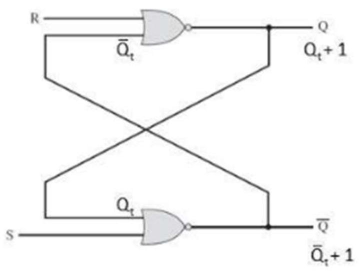
El biestable S-R asíncrono es un circuito a base de puertas lógicas con dos entradas (S (set) y R (Reset) y dos salidas Q1 y Q2.

- **El cerrojo (latch) S-R** (Set – Reset):

S	R	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

ENTRADA PROHIBIDA

El biestable RS, con los valores de R y S 1, 1 puede oscilar, dependiendo del tipo de puertas empleadas

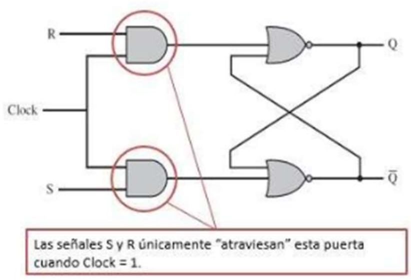


NOR		
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

El cerrojo (latch) S-R con reloj

- Tabla simplificada:

Clock	S	R	Q_{t+1}
1	0	0	Q_t
1	0	1	0
1	1	0	1
1	1	1	-
0	0	0	Q_t
0	0	1	Q_t
0	1	0	Q_t
0	1	1	Q_t



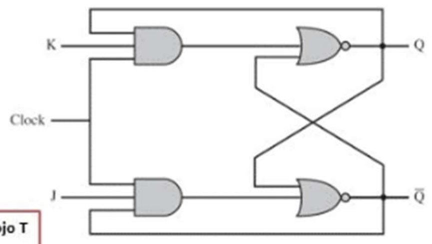
El cerrojo (latch) J-K con reloj

Similar al S-R, pero todas las entradas son válidas, las entradas 0-0, 0-1 y 1-0 funcionan igual que el S-R con reloj y la entrada 1-1 da como resultado la negación del estado anterior.

- Tabla simplificada:

Clock	J	K	Q_{t+1}
1	0	0	Q_t
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_t$

Cuando J = K = 1 siempre, se convierte en un **cerrojo T (toggle)**, cuya salida es siempre la inversa de la anterior.

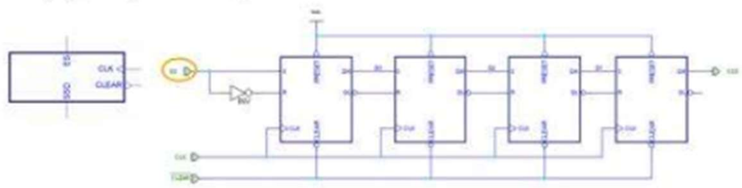


Registros

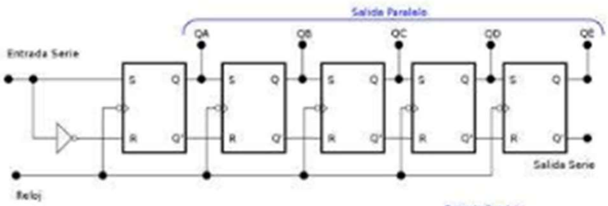
Un biestable permite guardar valores de bit, los registros son por tanto circuitos secuencuiales capaces de almacenar información y comparten sus señales de control, pudiendo ser la transferencia de información en serie, cuando van a continuación de otro o en paralelo cuando lo hacen de manera simultánea.

Ejemplo: registro de desplazamiento hacia la derecha con biestables SR.

Registro SERIE

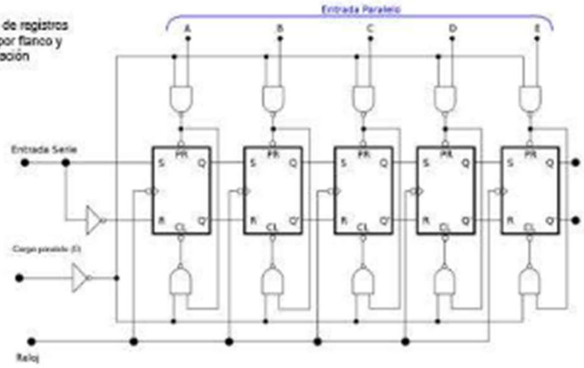


Registro SERIE-PARALELO (Se pueden obtener todos los datos a la vez. Por otro lado, también se puede obtener una salida en serie de cualquier salida Q)



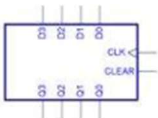
Para explicar el funcionamiento de registros síncronos se necesita: disparo por flanco y considerar el retardo de propagación de los biestables

Registro PARALELO - SERIE

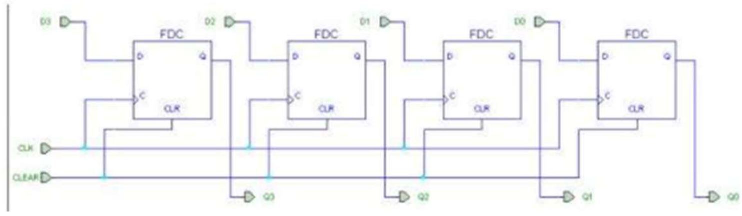


Carga Paralelo (0)
Preset (A..E=1) y
Clear (A..E=0)

Registro PARALELO - PARALELO



Ejemplo con biestables D síncronos por flanco de subida.



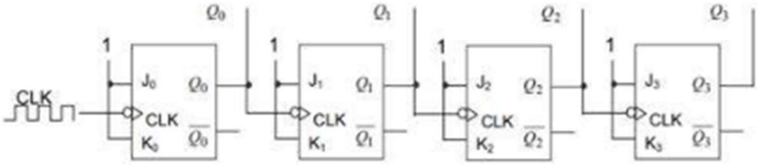
Contadores

Un contador es un circuito que genera una secuencia ordenada de salidas, repetidas en el tiempo que coinciden con el estado de sus biestables, pudiéndose ver también como una combinación de biestables que se rigen por una señal de reloj que aumentan de uno en uno el valor de la salida, siendo comúnmente utilizados en fases de proyectos.

Tipos:

- Síncronos: todos los biestables comparten la misma señal de reloj, producida por un oscilador
- Asíncronos: no todos los biestables comparten la misma señal de reloj. Se aplica una señal externa a la entrada de reloj del primer biestable y a los siguientes se les aplica como señal de reloj la salida del biestable anterior.
- Ascendente: la cuenta es creciente.
- Descendente: la cuenta es decreciente
- Reversible: la cuenta puede ser ascendente o descendente en función de una entrada de control.

Ascendente:

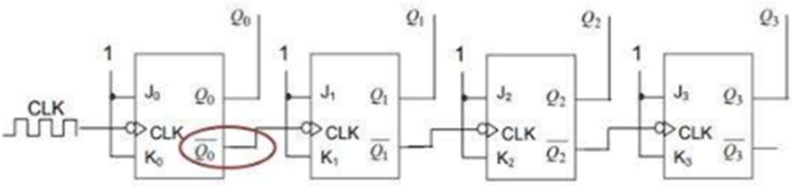


Clock	J	K	Q_{t+1}
1	0	0	Q_t
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_t$

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Cada vez que hay un flanco de bajada, la salida se invierte.
No sucede en paralelo, son secuenciales

Descendente:



Valor inicial 1111

Q_3	Q_2	Q_1	Q_0	
1	1	1	1	15
1	1	1	0	14
1	1	0	1	13
1	1	0	0	12
1	0	1	1	11
1	0	1	0	10
1	0	0	1	9
1	0	0	0	8
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0