

Clase 09 - Circuitos lógicos

IIC1001 - Algoritmos y Sistemas Computacionales

Cristian Ruz – cruz@uc.cl

Lunes 8-Abril-2024

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Contacto

Temas

Compuertas Lógicas

Circuitos lógicos

Contacto

Temas

Compuertas Lógicas

Circuitos lógicos

Contacto



ignaciomunoz@uc.cl

Ignacio Muñoz
Ayudante jefe

- Coordinación
- Notas de actividades, interrogaciones
- Todo lo que no sé donde más enviar



vicente.cabra@uc.cl

Vicente Cabra
Ayudante

- Materia



fernando.concha@uc.cl

Fernando Concha
Ayudante

- Materia



alejandro.tapia@uc.cl

Alejandro Tapia
Ayudante

- Materia



Contacto

Temas

Compuertas Lógicas

Circuitos lógicos

Sistemas computacionales

- Representación datos, números y codificación
- Funcionamiento hardware, procesadores y memoria.
- Funcionamiento de sistemas operativos: ejemplo scheduling
- Funcionamiento de Internet
- Herramientas computacionales: github + latex

Algoritmos

- Algoritmos y resolución de problemas
- Eficiencia algorítmica
- Estructuras secuenciales y ordenamiento
- Grafos y árboles

Lunes 8-Abril: Clase

- Circuitos lógicos
- Sumadores
- Multiplexores

Miércoles 10-Abril: Repaso en horario de clase

- Traigan preguntas
- l1 de 17:30 a 19:30, Salas B23 y BC24
- Escrito, sin apuntes. Se entregarán tablas (ASCII, compuertas)

Contacto

Temas

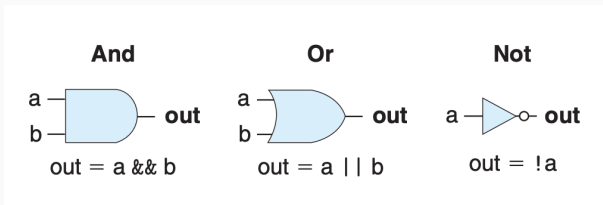
Compuertas Lógicas

Circuitos lógicos

Compuertas lógicas

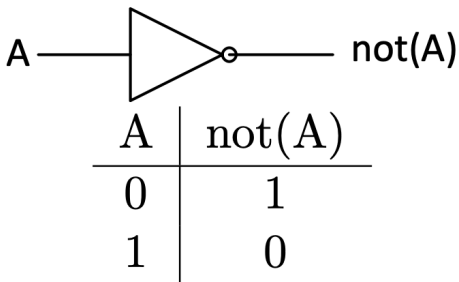
Componentes electrónicas que implementa las condiciones lógicas de Boole.

Poseen una cantidad de valores (bit) de entrada y entregan como resultado uno o más valores (bits) de salida de acuerdo a una tabla.



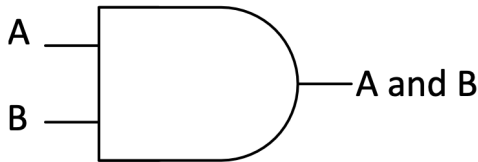
Compuerta NOT

Entrada es lo opuesto de la salida: $\text{not}(A) = \neg A$



Compuerta AND

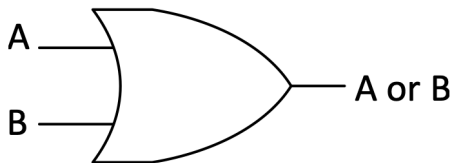
Entrada es verdadera (1) sólo cuando **ambas** entradas son verdaderas: $A \text{ and } B = A \wedge B$



A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

Compuerta AND

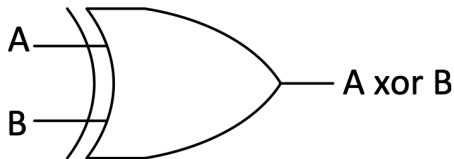
Entrada es verdadera (1) cuando **al menos una** de las entradas son verdaderas: $A \text{ or } B = A \vee B$



A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

Compuerta XOR

Entrada es verdadera (1) cuando **exactamente una** de las entradas son verdaderas: $A \text{ xor } B = A \oplus B$



A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

Contacto

Temas

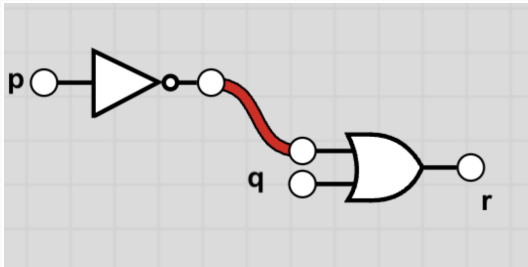
Compuertas Lógicas

Circuitos lógicos

Circuitos binarios

Mezclando entradas y salidas de diferentes compuertas podemos construir circuitos que implementan operaciones lógicas.

¿Qué salidas se obtienen del siguiente circuito? (I2, 2023-1)



Circuitos binarios

Construir un circuito cuya salida sea 1 solamente si ambos bit de entrada son iguales.

a	b	$a \text{ eq } b$
0	0	1
0	1	0
1	0	0
1	1	1

¿Cómo se escribiría en álgebra booleana?

¿Cómo se escribiría un circuito lógico?

Circuitos sumadores

Podemos empezar a construir circuitos que implementes operaciones aritméticas.

¿Cómo sería relación entrada/salida? $S(A, B) = A + B$

A	B	$A + B$
0	0	00
0	1	01
1	0	01
1	1	10

Circuitos sumadores

Podemos empezar a construir circuitos que implementes operaciones aritméticas.

¿Cómo sería relación entrada/salida? $S(A, B) = A + B$

A	B	$A + B$
0	0	00
0	1	01
1	0	01
1	1	10

¡Pero la salida de una suma de dos números de un 1 bit son dos bit!

Circuitos sumadores

INPUT: bit A y bit B

OUTPUT: dos bit. Podemos separarlos: S_1 , S_0

A	B	$A + B = S_1S_0$	S_1	S_0
0	0	00	0	0
0	1	01	0	1
1	0	01	0	1
1	1	10	1	0

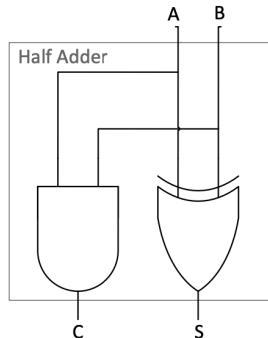
Circuitos sumadores

INPUT: bit A y bit B

OUTPUT: dos bit. Podemos separarlos: S_1 , S_0

A	B	$A + B = S_1S_0$	S_1	S_0
0	0	00	0	0
0	1	01	0	1
1	0	01	0	1
1	1	10	1	0

Half-adder



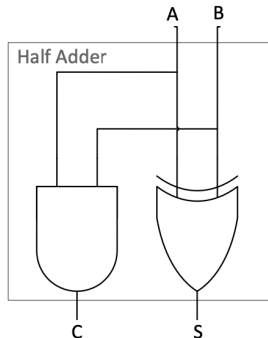
Circuitos sumadores

INPUT: bit A y bit B

OUTPUT: dos bit. Podemos separarlos: S_1 , S_0

A	B	$A + B = S_1S_0$	S_1	S_0
0	0	00	0	0
0	1	01	0	1
1	0	01	0	1
1	1	10	1	0

Half-adder



Con esto podemos dos números de 1 bit.
¿Cómo sumar números más grandes?

Circuitos sumadores

¿Cómo sumar números, por ejemplo, de 2-bit?

INPUT: dos números de dos bit: $A = A_1A_0$, $B = B_1B_0$

OUTPUT: $A + B = S$ ¿cuántos bit?

Hay que ver cómo funciona la
suma

$$\begin{array}{rcccc} & & 1 & & 1 \\ & 1 & 0 & 0 & 1 \\ + & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & \end{array}$$

Circuitos sumadores

¿Cómo sumar números, por ejemplo, de 2-bit?

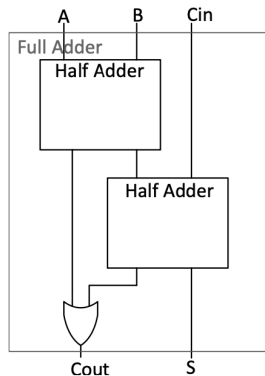
INPUT: dos números de dos bit: $A = A_1A_0$, $B = B_1B_0$

OUTPUT: $A + B = S$ ¿cuántos bit? $S = S_2S_1S_0$

Hay que ver cómo funciona la
suma

$$\begin{array}{r} \\ \\ + \\ \hline 1 \end{array}$$

Full-adder



Circuitos sumadores

¿Cómo sumar números, por ejemplo, de 2-bit?

INPUT: dos números de dos bit: $A = A_1A_0$, $B = B_1B_0$

OUTPUT: $A + B = S$ ¿cuántos bit?

A_1A_0	B_1B_0	$S = S_2S_1S_0$
00	00	000
00	01	001
00	10	010
00	11	011
01	00	001
01	01	010
01	10	011
01	11	100
10	00	010
10	01	011
10	10	100
10	11	101
11	00	011
11	01	100
11	10	101
11	11	110

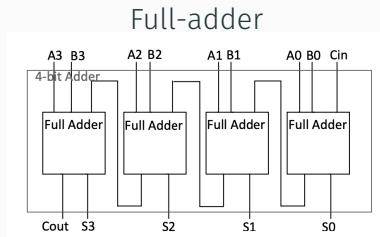
Circuitos sumadores

¿Cómo sumar números, por ejemplo, de 2-bit?

INPUT: dos números de dos bit: $A = A_1A_0$, $B = B_1B_0$

OUTPUT: $A + B = S$ ¿cuántos bit? $S = S_2S_1S_0$

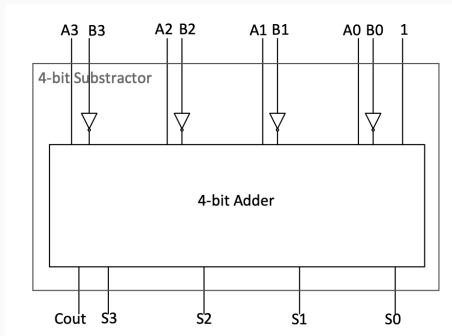
A_1A_0	B_1B_0	$S = S_2S_1S_0$
00	00	000
00	01	001
00	10	010
00	11	011
01	00	001
01	01	010
01	10	011
01	11	100
10	00	010
10	01	011
10	10	100
10	11	101
11	00	011
11	01	100
11	10	101
11	11	110



Circuitos sumadores

También se puede implementar restadores en complemento de 2.

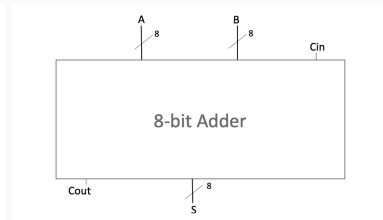
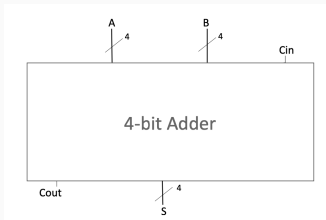
Conversión a complemento de 2 es: (1) Invertir los bits, (2) Sumar 1



Restador de 4 bits

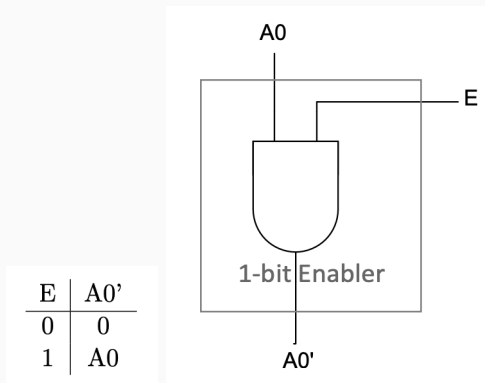
Abstrayendo componentes

Al construir circuitos más complejos podemos ocultar detalles. La notación se simplifica indicando cuántas líneas (bit) lleva cada entrada. Estas líneas se conoce como "buses".



Otros componentes: Enabler

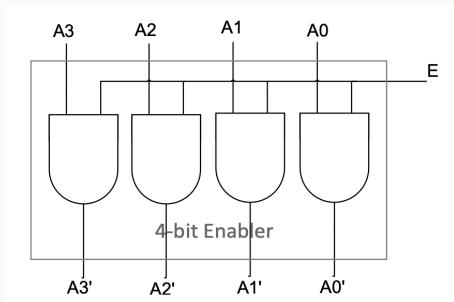
Componente que permite habilitar o deshabilitar una salida



Enabler de 1 bit

Enabler

Se puede implementar para circuitos con más bit de entrada



Enabler de 4 bits

Multiplexor

Buscamos un circuito que tenga más de una entrada, y nos permita **elegir** cuál de ellas queremos leer.

Dos entradas: $A0$, y $B0$.

Una salida: $M0$.

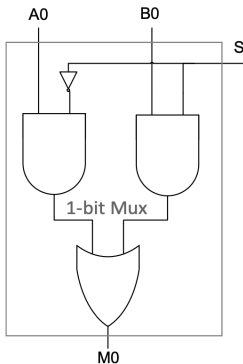
Señal de elección: S . Si $S = 0$, queremos leer $A0$. Si $S = 1$, queremos leer $B0$.

S	$M0$
0	$A0$
1	$B0$

Multiplexor

Señal de elección: S . Si $S = 0$, queremos leer $A0$. Si $S = 1$, queremos leer $B0$.

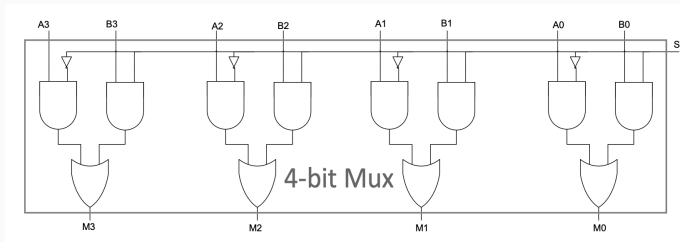
S	$M0$
0	$A0$
1	$B0$



Multiplexor de 2 entradas de 1 bit de datos

Multiplexor

Se puede implementar con más bits

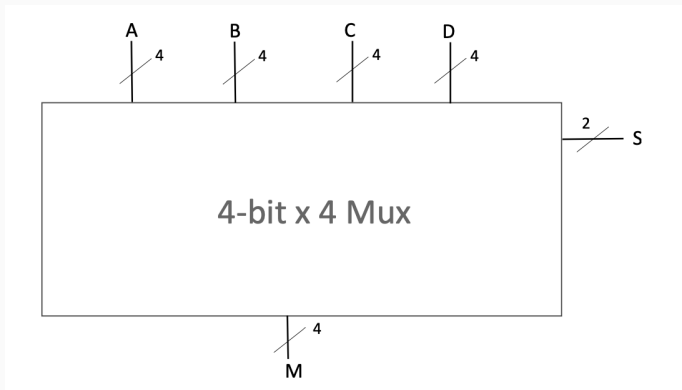


S1	S1	M
0	0	A
0	1	B
1	0	C
1	1	D

Multiplexor de 2 entradas de 4 bit de datos

Multiplexor

En notación de buses



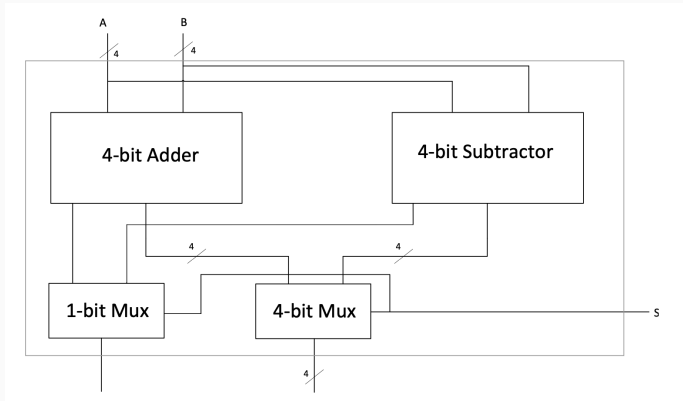
Multiplexor de 4 entradas de 4 bit de datos

Unidad de selección

Se pueden combinar operaciones y seleccionar usando multiplexores.

Es una primera aproximación a una **unidad aritmético-lógica (ALU)**

$S = 0$: Suma, $S = 1$: Resta



Sumador-restador de 4 bit de datos