

Clase 03 - Operaciones aritméticas y lógicas

Profesor: **IIC2343 - Arquitectura de Computadores**

- Felipe Valenzuela González

Correo:

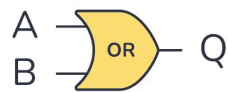
frvalenzuela@alumni.uc.cl

Resumen de la clase pasada

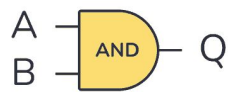
Compuertas lógicas



A	Q
0	1
1	0



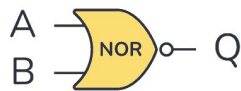
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



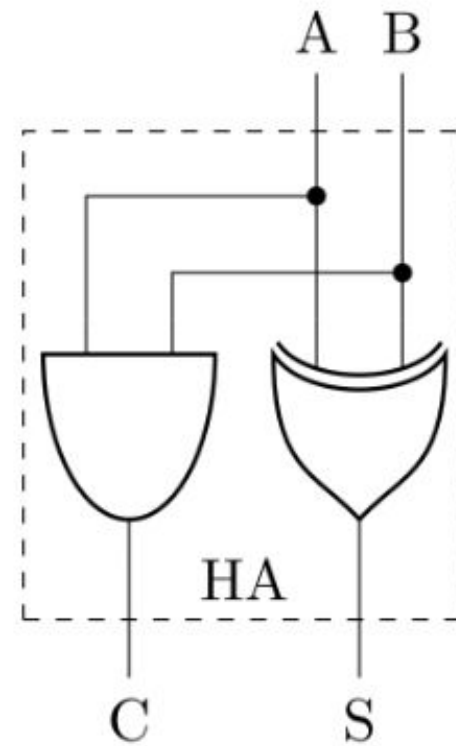
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Algebra booleana

Equivalence	Name of Identity
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity Laws <input type="checkbox"/>
$p \wedge F \equiv F$ $p \vee T \equiv T$	Domination Laws <input type="checkbox"/>
$p \wedge p \equiv p$ $p \vee p \equiv p$	Idempotent Laws
$\neg(\neg p) \equiv p$	Double Negation Law
$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$	Commutative Laws
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ $(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative Laws
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	Distributive Laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's Laws <input type="checkbox"/>
$p \wedge (p \vee q) \equiv p$ $p \vee (p \wedge q) \equiv p$	Absorption Laws
$p \wedge \neg p \equiv F$ $p \vee \neg p \equiv T$	Negation Laws

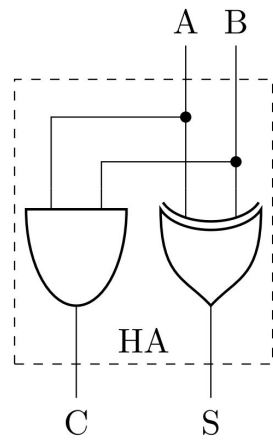
Half Adder

<i>A</i>	<i>B</i>	<i>S</i>	<i>C</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

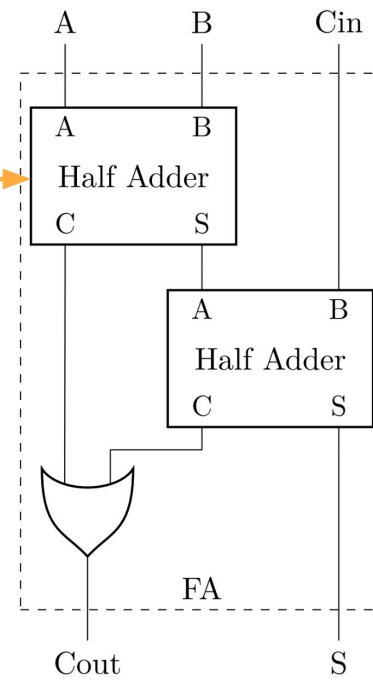


(a) Half Adder

Full Adder

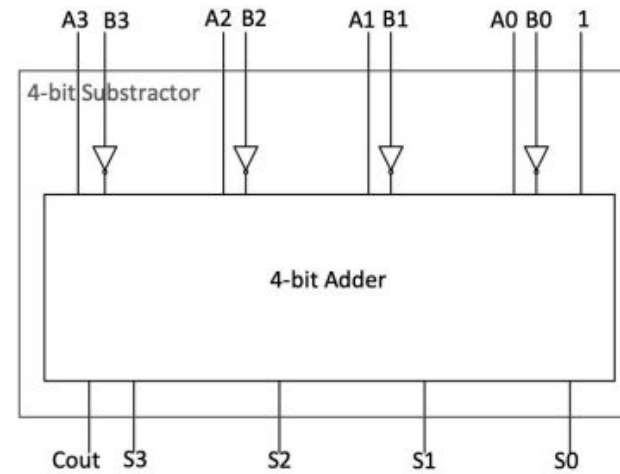
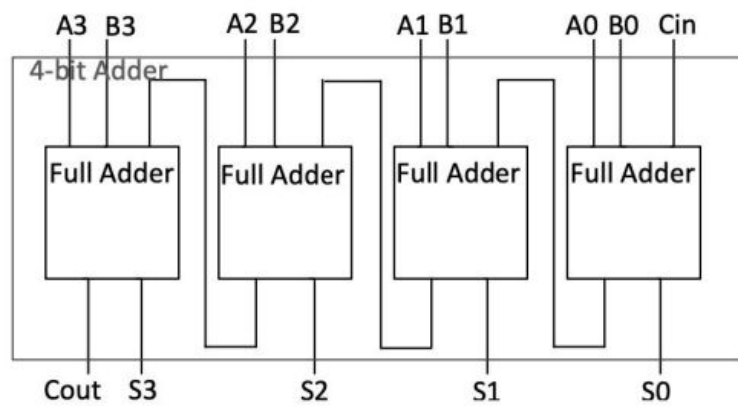


(a) Half Adder





(b) Full Adder

Sumador/Restador



¿Dudas?

Introducción:

- Un computador lo definimos como una **máquina programable que ejecuta programas.**
- Para programar necesitamos:
 - Datos: números (enteros, reales) , texto, imágenes, etc 
 - Operaciones: suma, resta, multiplicación, división, etc 
 - Variables: simples, arreglos
 - Control de flujo: comparaciones, manejo de ciclos
- La clase de hoy veremos más **operaciones, y lo básico de variables y control**

Objetivos de la clase

- Conocer más operaciones aritméticas y lógicas
- Conocer circuitos básicos de control de flujo
- Comenzar el entendimiento de almacenamiento de datos

Resolución de circuitos

Minterm

- Las señales de **input** se conectan con compuertas **AND**
- Los valores 0 se niegan con la compuerta **NOT**
- **Todas las combinaciones** se conectan con **compuertas OR**
- Las combinación de de **input** debe dar 1

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Minterm

Ejemplo:

NOT(A) AND B AND NOT(C)

OR

A AND NOT(B) AND C

OR

A AND B AND C

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Maxterm

- Las señales de **input** se conectan con compuertas **OR**
- Los valores 1 se niegan con la compuerta **NOT**
- **Todas las combinaciones** se conectan con **compuertas OR**
- Las combinación de de **input** debe dar 0

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Maxterm

Ejemplo:

A OR B OR C

AND A OR B OR NOT(C)

AND A OR NOT(B) OR NOT(C)

AND NOT(A) OR B OR C

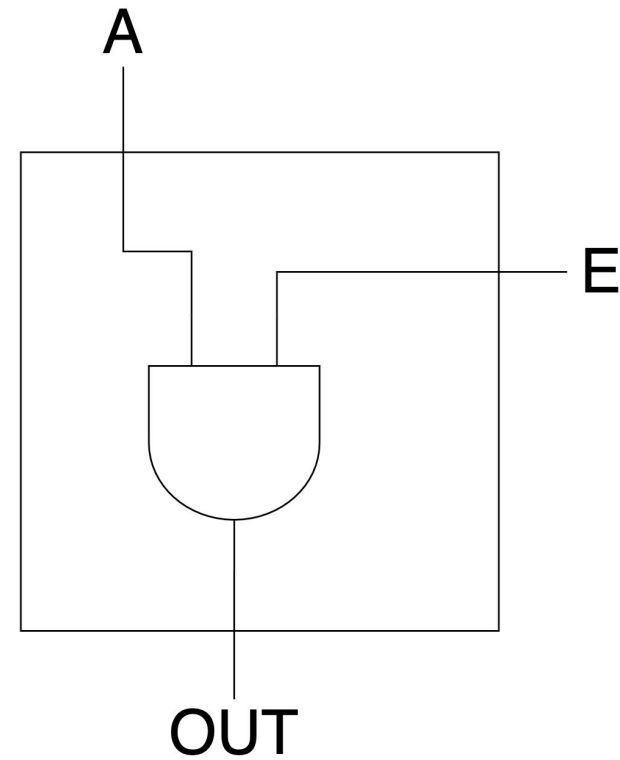
AND NOT(A) OR NOT(B) OR C

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

¿Dudas?

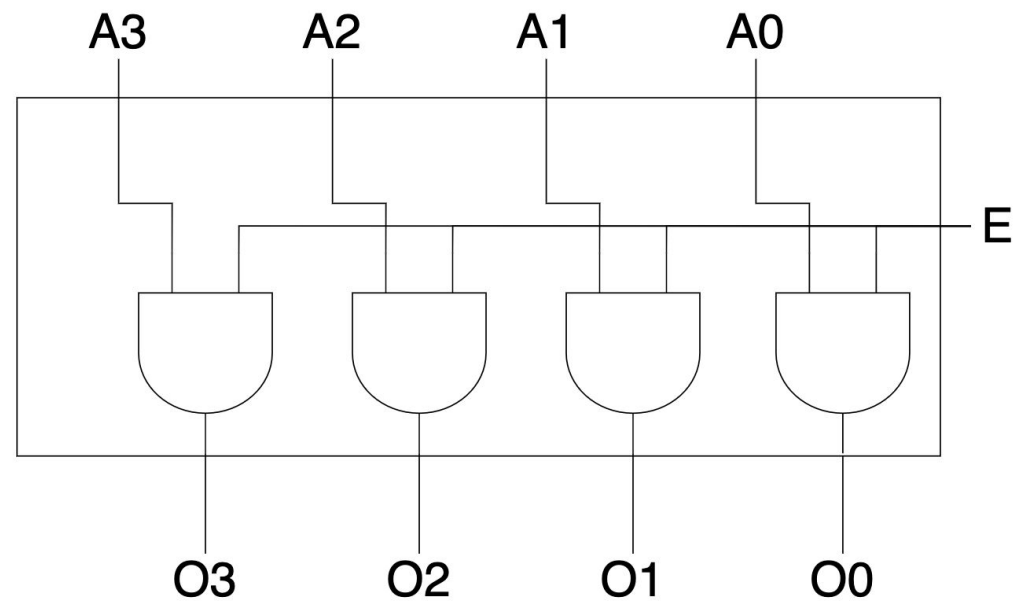
Circuitos de control - Enabler

OUT	E
0	0
A	1



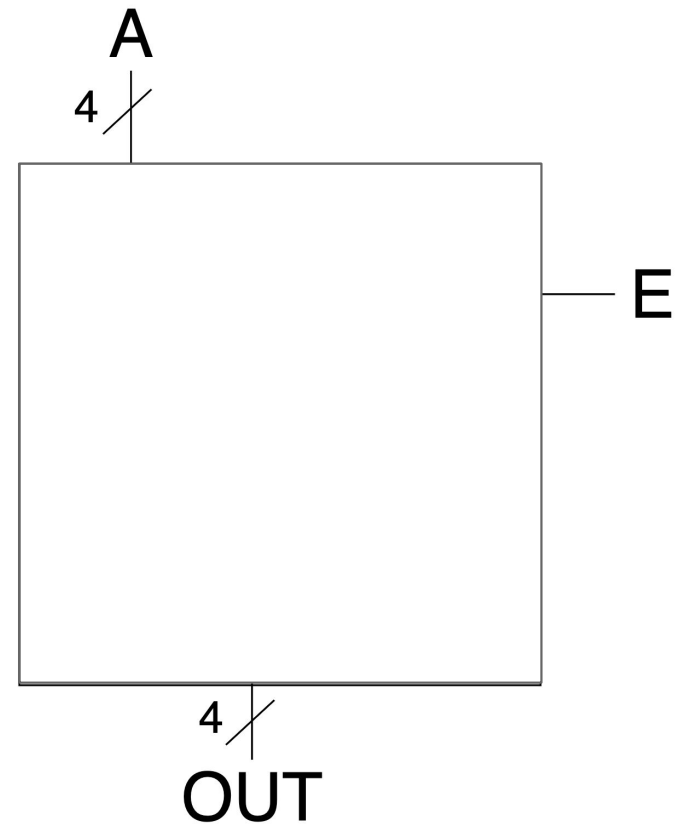
Circuitos de control - Enabler 4 bits

O3O2O1O0	E
0	0
A3A2A1A0	1



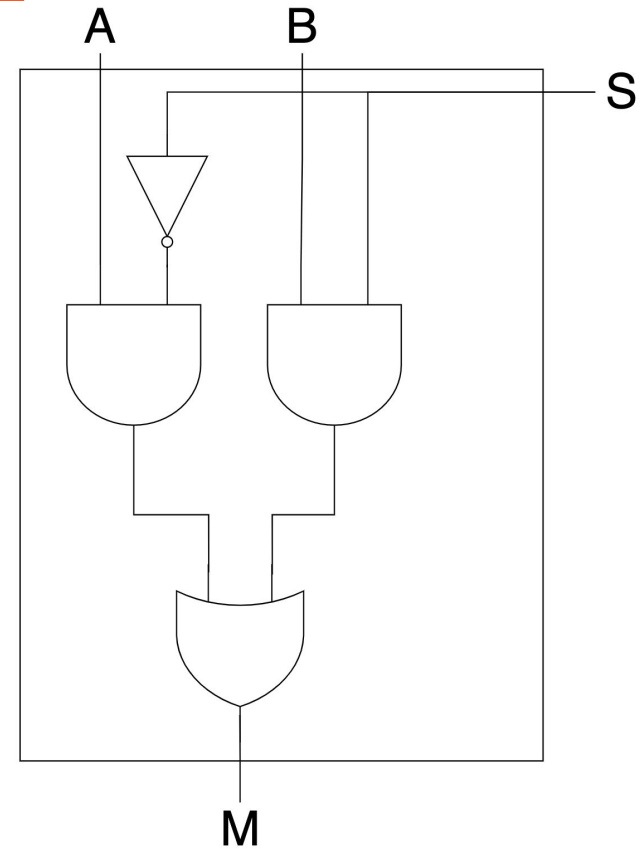
Circuitos de control - Enabler - Bus Format

OUT	E
0	0
A	1



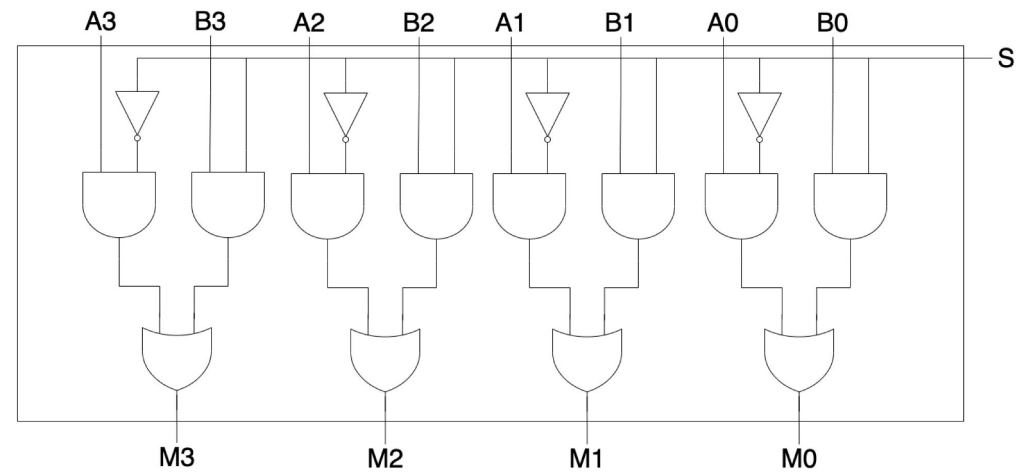
Circuitos de control - Multiplexor

S	M
0	A
1	B



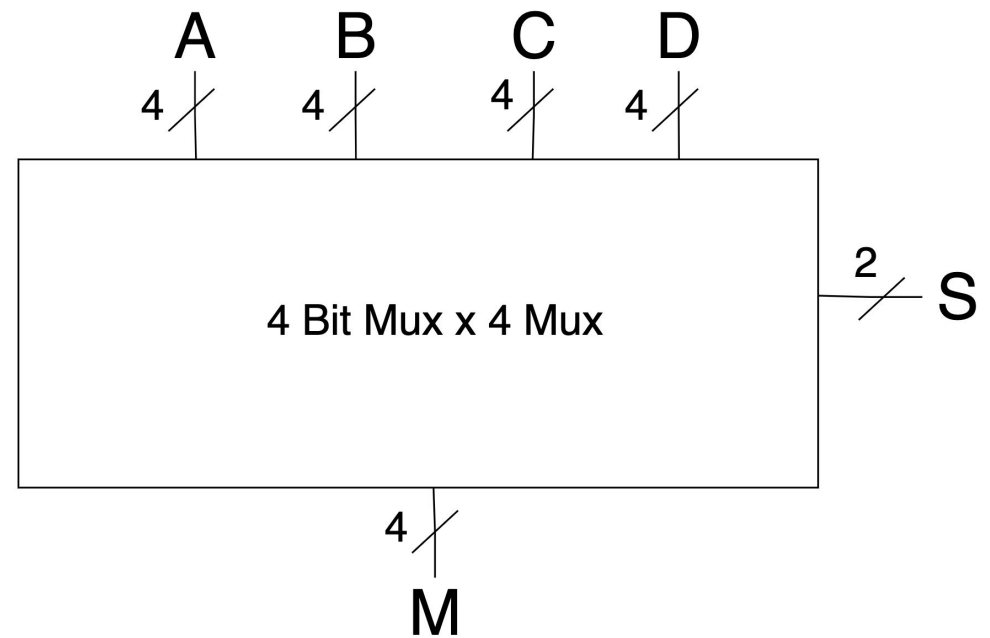
Circuitos de control - Multiplexor 4 bits

S	M
0	A
1	B



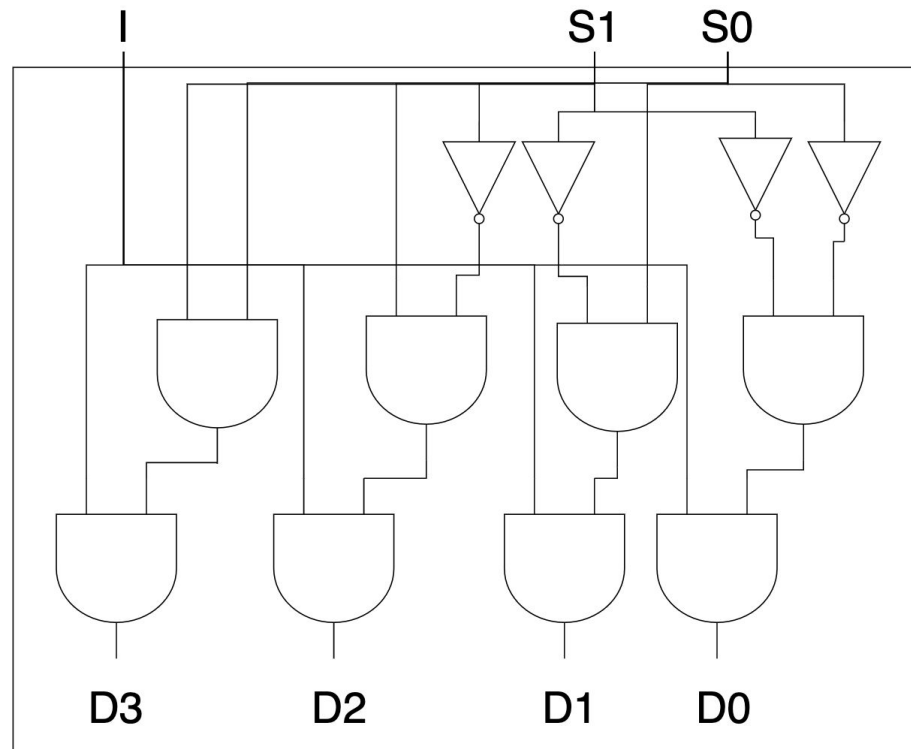
Circuitos de control - 4 bit x 4 Mux

S1	S0	M
0	0	A
0	1	B
1	0	C
1	1	D



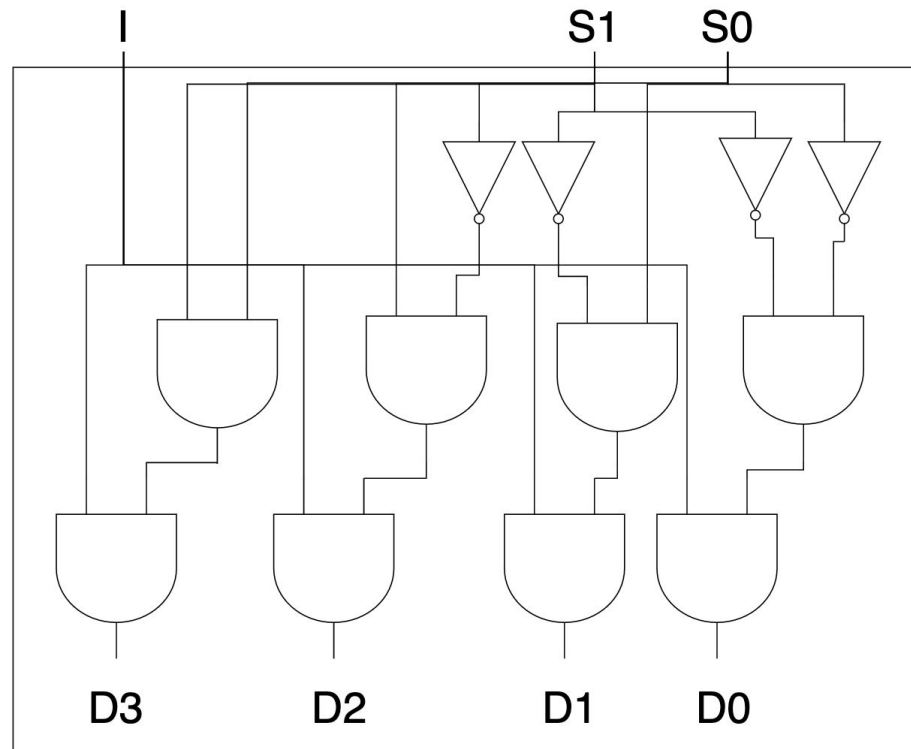
Circuitos de control - Demux

S1	S0	Output
0	0	D0 = I
0	1	D1 = I
1	0	D2 = I
1	1	D3 = I



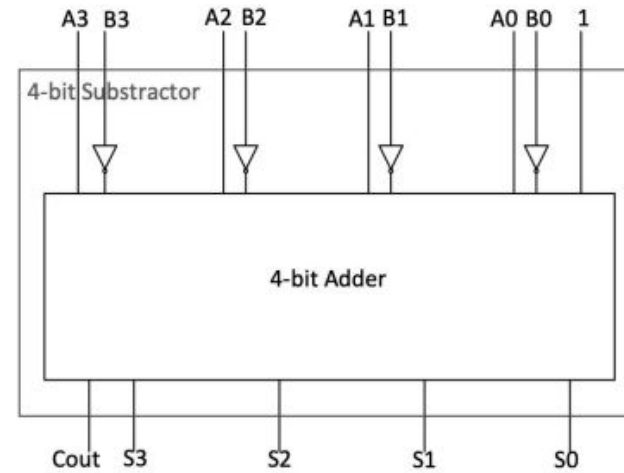
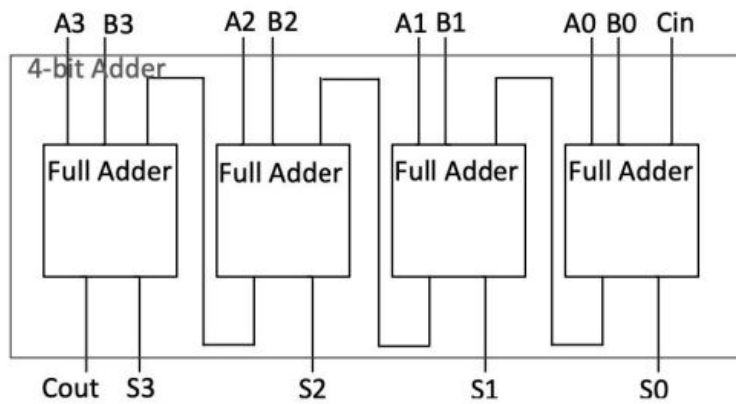
Circuitos de control - Decoder

S1	S0	Output
0	0	D0 = I
0	1	D1 = I
1	0	D2 = I
1	1	D3 = I



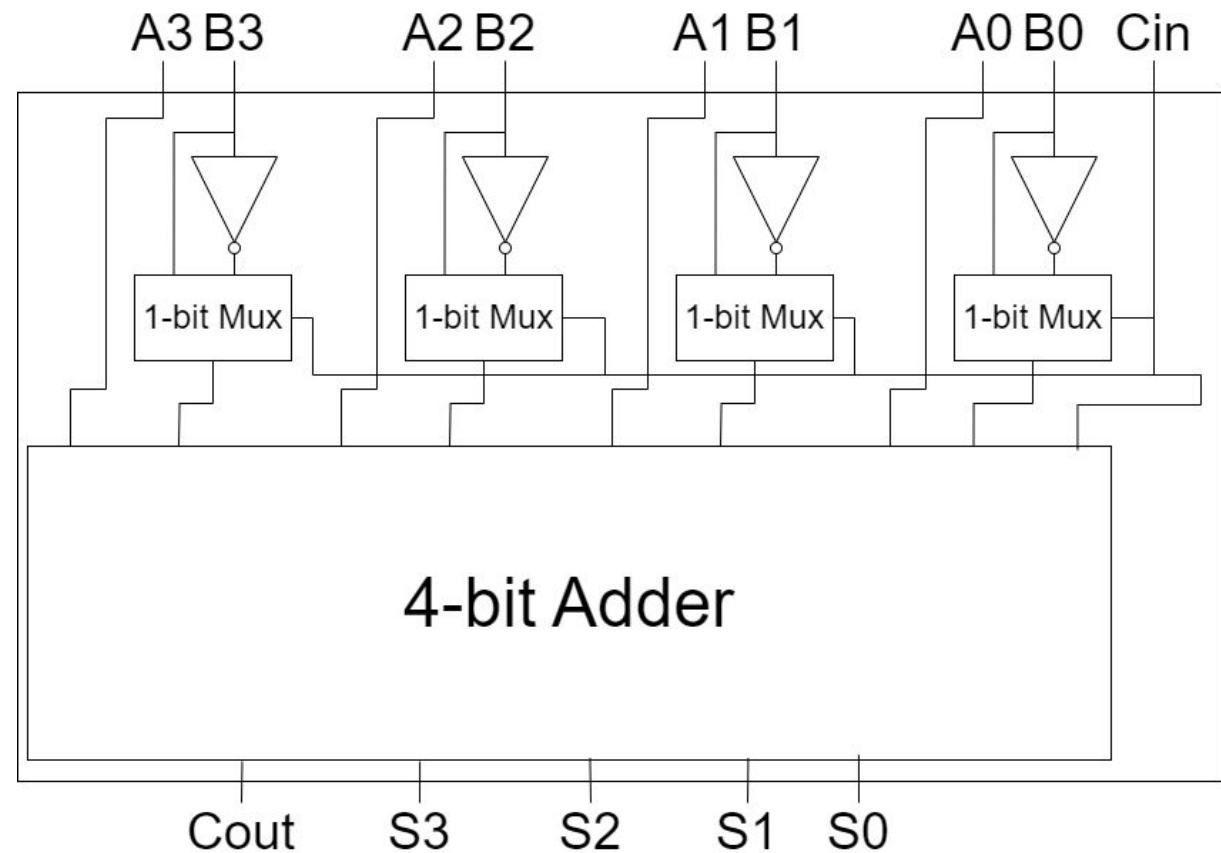
¿Dudas?

Operaciones: Sumador y Restador de 4 bits



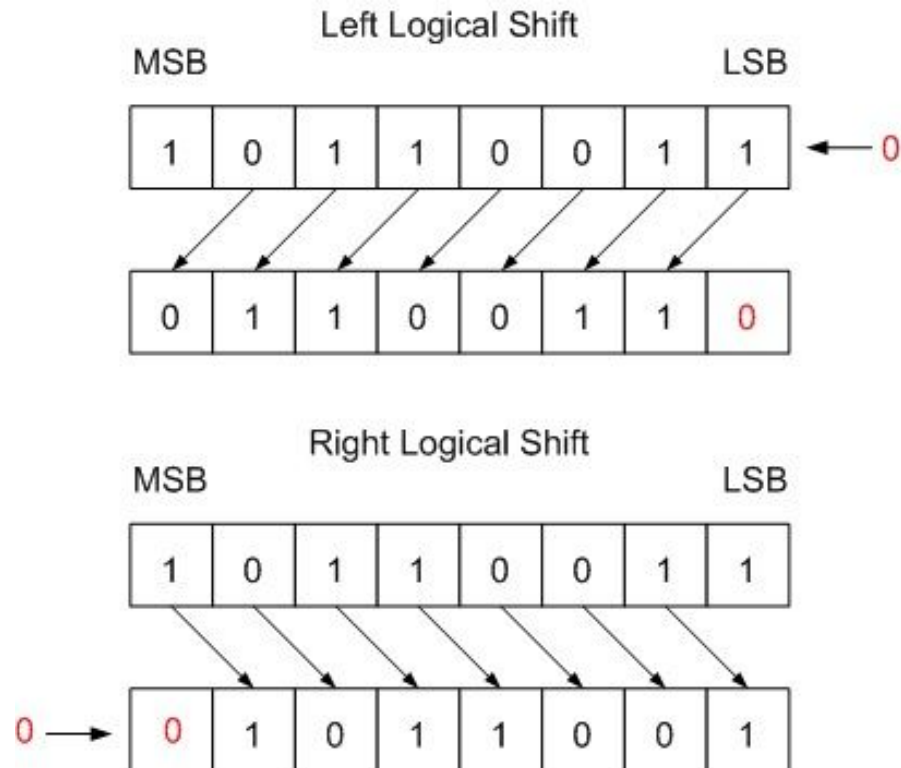
Operaciones: Sumador-Restador de 4 Bits

Cin	Result
0	A+B
1	A-B



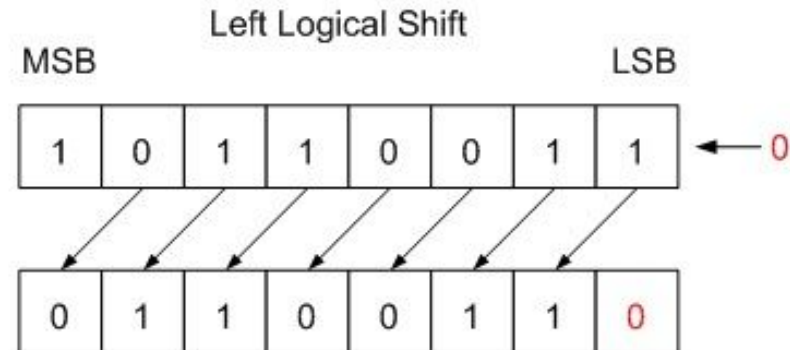
Operaciones: Lógicas - Shift Lógico

- Una operación muy ocupada es el **shifting** o desplazamiento.
- Consiste en mover todas las cifras a la izquierda o la derecha.



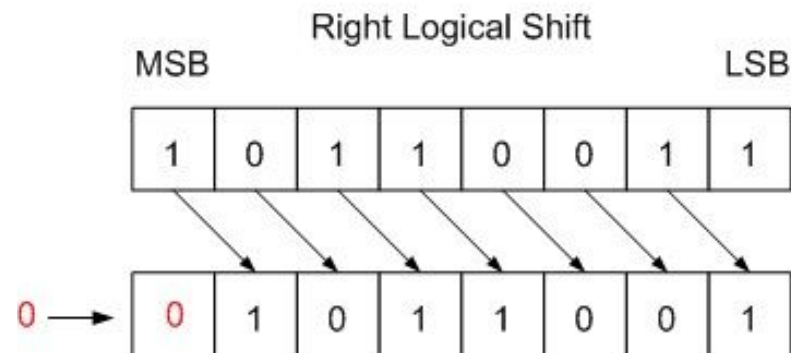
Operaciones: Lógicas - Shift Left Lógico (SHL)

- Definiremos el desplazamiento a la izquierda como **SHIFT LEFT (SHL)**
- En binario, desplazar a la izquierda es análogo a multiplicar por 2
- Se agrega un cero en el bit **menos** significativo



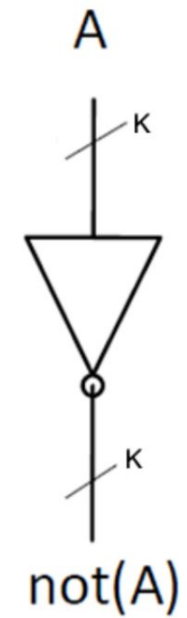
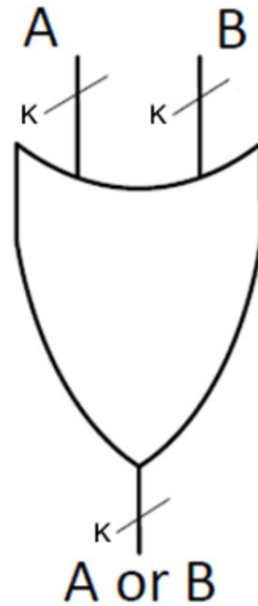
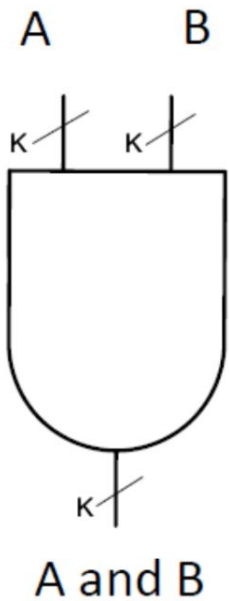
Operaciones: Lógicas - Shift Right Lógico (SHR)

- Definiremos el desplazamiento a la izquierda como **SHIFT RIGHT (SHR)**
- En binario, desplazar a la izquierda es análogo a **división entera por 2**
- Se agrega un cero en el bit **más** significativo



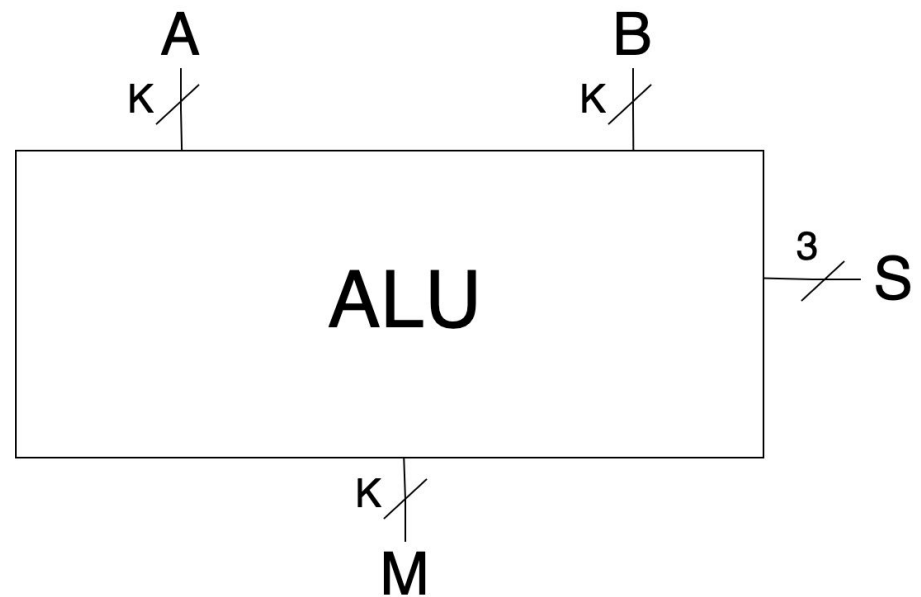
¿Dudas?

Operaciones: Operadores lógicos *bitwise*



Unidad Aritmética Lógica: ALU

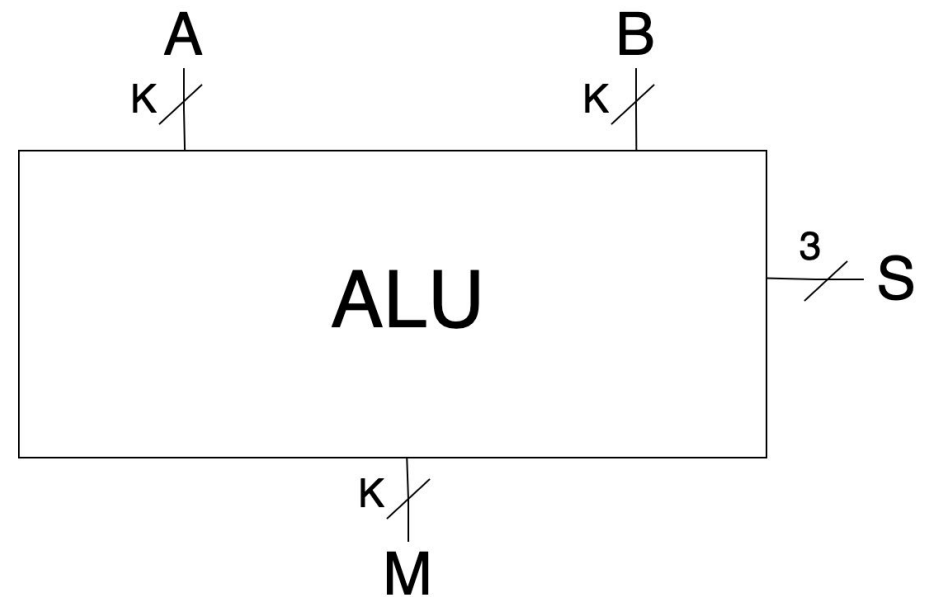
- Es el **componente principal** para hacer operaciones en un computador
- Se basa en **operaciones lógicas bitwise** y operaciones aritméticas como lo son el **sumador-restador**
- Su comportamiento interno lo verán **próxima semana** en la actividad de laboratorio



Unidad Aritmética Lógica: ALU

- Tabla de valores:

S2	S1	S0	M
0	0	0	Suma
0	0	1	Resta
0	1	0	And
0	1	1	Or
1	0	0	Not
1	0	1	Xor
1	1	0	Shift left
1	1	1	Shift right



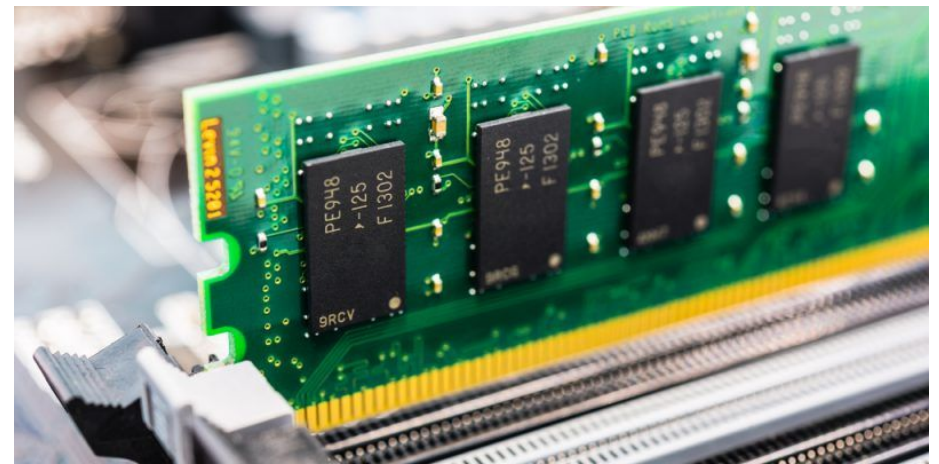
Introducción del curso:

- Un computador lo definimos como una **máquina programable que ejecuta programas.**
- Para programar necesitamos:
 - Datos: números (enteros, reales) , texto, imágenes, etc
 - Operaciones: suma, resta, multiplicación, división, etc
 - Variables: simples, arreglos
 - Control de flujo: comparaciones, manejo de ciclos
- Para terminar la clase de hoy, y veremos **lo básico para tener variables**



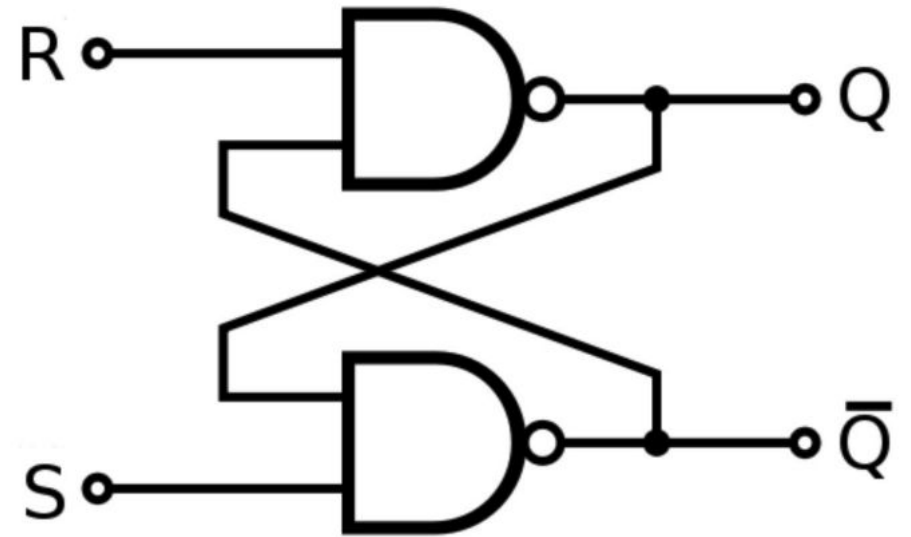
Almacenamiento de datos

- Es componente esencial de todo computador
- Es *hardware* especializado para **almacenar** un estado
- Permite realizar **un cambio** en un **instante determinado de tiempo**



Almacenamiento de datos: Latch RS

S	R	Q(t+1)
0	0	-
0	1	0
1	0	1
1	1	Q(t)



Clase 03 - Operaciones aritméticas y lógicas

Profesor: **IIC2343 - Arquitectura de Computadores**

- Felipe Valenzuela González

Correo:

frvalenzuela@alumni.uc.cl