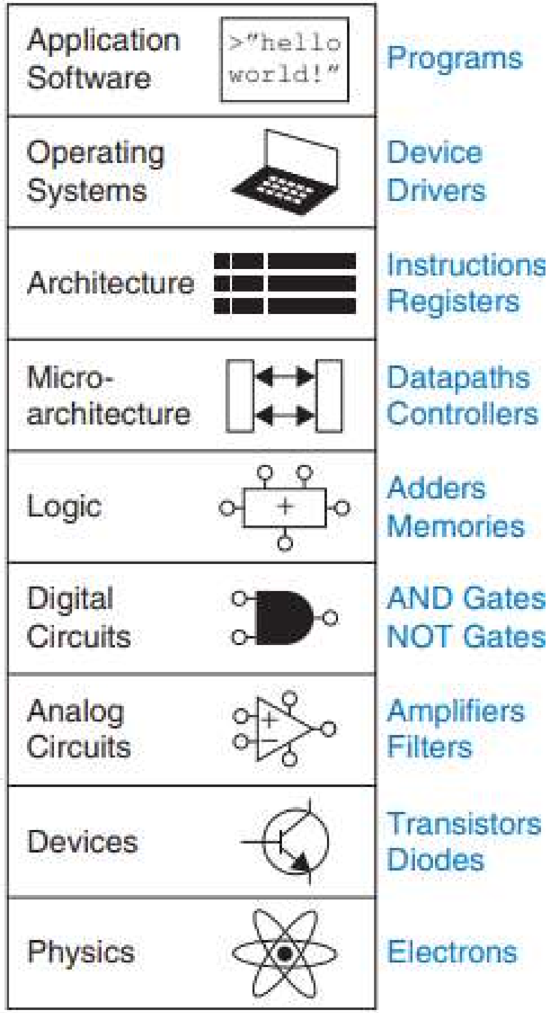
Inicio AOC_01.pptx
AOC_25 S1

El arte de gestionar la complejidad

- Abstracción.
- Disciplina o diseño
- Las tres cualidades:
  - -- Jerarquía
  - -- Modularidad
  - -- Regularidad



Abstracción

Ocultar detalles cuando no son importantes

Disciplina

- Restringir intencionalmente las opciones de diseño
- Ejemplo: Disciplina digital

  -- Voltajes discretos en lugar de continuos

  -- Más simple de diseñar que los circuitos analógicos: se puede construir sistemas más sofisticados
- Sistemas digitales que reemplazan a los predecesores analógicos: es decir, cámaras digitales, televisión digital, teléfonos celulares, CDs, etc.

Las tres cualidades

- Jerarquía

-- Un sistema dividido en módulos y submódulos

- Modularidad

-- Tener funciones e interfaces bien definidas

- Regularidad

-- Fomentar la uniformidad, para que los módulos se puedan reutilizar fácilmente

Disciplina digital: valores binarios

- Dos valores discretos:
  - -- 1’s y 0’s
  - -- 1, TRUE, ALTO (HIGH)
  - -- 0, FALSE, BAJO, (LOW)

- 1 y 0: niveles de tensión, engranajes giratorios, niveles de fluidos, etc.
- Los circuitos digitales usan niveles de voltaje para representar 1 y 0
- Bit: dígito binario

Sistemas numéricos

Números decimales

1000's column 100's column 10's column 1's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

five thousands | three hundreds | seven tens | four ones

Números binarios

8's column 4's column 2's column 1's column

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one eight | one four | no two | one one

Potencias de 2

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$

Conversión de bases

Conversión de base decimal a base binaria:

- Convert 10011_2 to decimal
- $16\times1 + 8\times0 + 4\times0 + 2\times1 + 1\times1 = 19_{10}$

Conversión de base binaria a base decimal :

- Convert 47_{10} to binary
- $32\times1 + 16\times0 + 8\times1 + 4\times1 + 2\times1 + 1\times1 = 101111_2$

Valores binarios y rango

- Número decimal de N dígitos

  -- ¿cuantos valores se pueden representar (dominio)? 10^N

  -- ¿Rango? $[0, 10^N - 1]$

  -- Ejemplo: número decimal de 3 dígitos: $10^3 = 1000$ valores posibles

  -- Rango: [0, 999]

- Número binario de n bits

  -- su dominio o cuantos valores se pueden representar: 2^N

  -- Rango: $[0, 2^N - 1]$

  -- Ejemplo: número binario de 3 dígitos:

  --- $2^3 = 8$ valores posibles

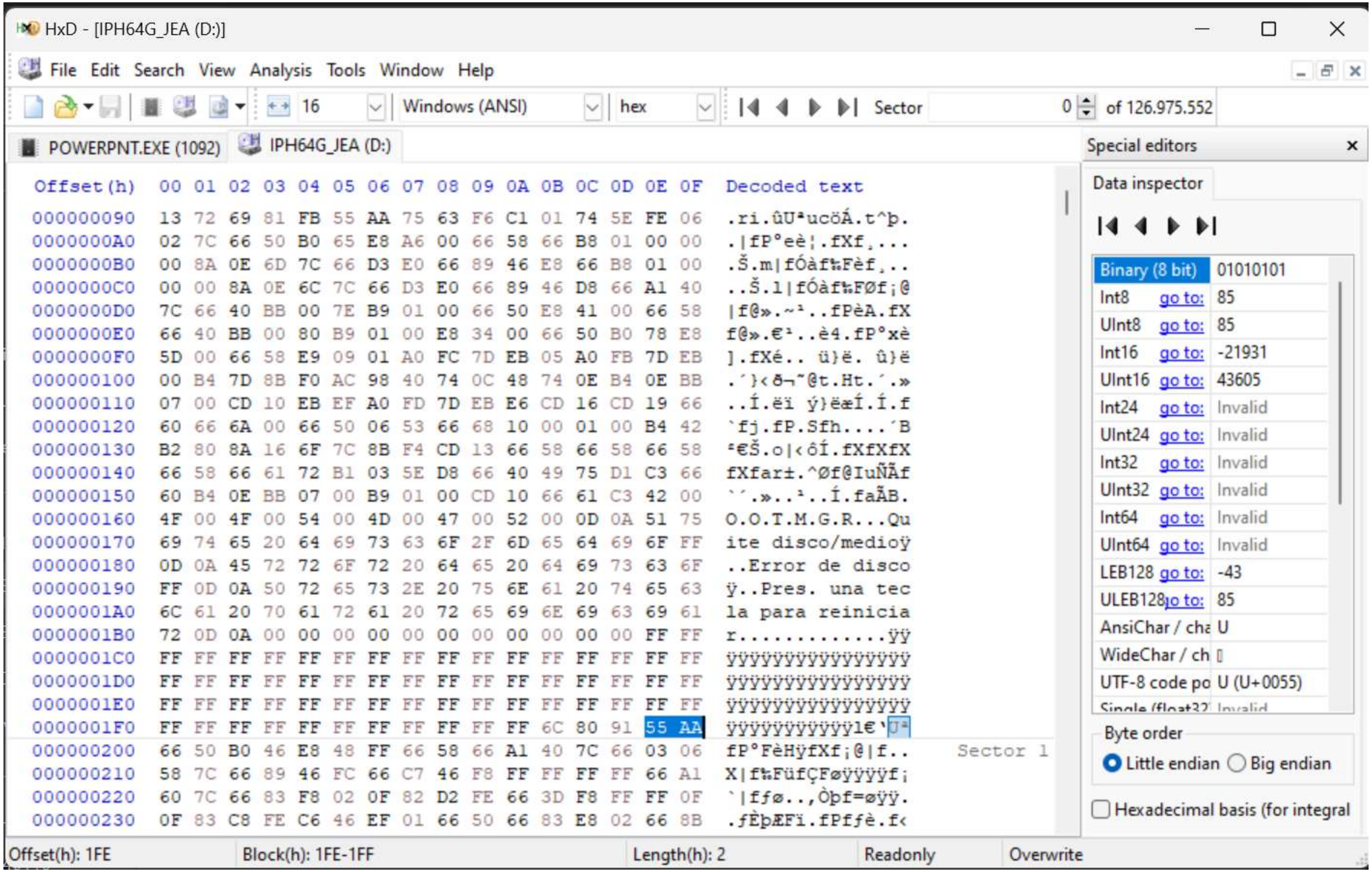
  --- Rango: $[0, 7] = [000_2$ a $111_2]$

Números hexadecimales (Base 16)

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

- Base 16
- Taquigrafía para binario

Representación en el nivel de lenguaje de máquina (Lenguaje Assembly) de la información



Ver adjunto ASCII_8bit.pdf

Conversión Hex a Bin

Hexadecimal a Binario

- Convert $4AF_{16}$ (also written 0x4AF) to binary
- $0100\ 1010\ 1111_2$

Conversión Hex a Dec

- Convert $4AF_{16}$ to decimal
- $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

Bits, Bytes, Nibbles ...

- Bits

- Bytes & Nibbles

- Bytes

Word 2 Bytes, doubleword 4 Bytes, quadword 8 Bytes, octalword 16 Bytes

Grandes potencias de 2

2^{10} = 1 Kilobit = 1024 bit
2^{20} = 1 Megabit = 1 048 576 bit
2^{30} = 1 Gigabit = 1 073 741 824 bit
2^{40} = 1 Terabit = 1 099 511 627 776 bit
2^{50} = 1 Petabit = 1 125 899 906 842 624 bit

Estos múltiplos son extensibles a la métrica del Byte, donde se usan mayormente:

Unit	Equivalent
1 kilobyte (KB)	1,024 bytes
1 megabyte (MB)	1,048,576 bytes
1 gigabyte (GB)	1,073,741,824 bytes
1 terabyte (TB)	1,099,511,627,776 bytes
1 petabyte (PB)	1,125,899,906,842,624 bytes

Observación!

Nota:
Los nombres y abreviaturas de números de bytes se confunden fácilmente con las notaciones de bits.
Las abreviaturas de números de bits utilizan una "b" minúscula en lugar de una "B" mayúscula.
Dado que un byte se compone de ocho bits, esta diferencia puede ser significativa.

Por ejemplo, si se anuncia una conexión a Internet de banda ancha con una velocidad de descarga de 3,0 Mbps, su velocidad es de 3,0 megabits por segundo o 0,375 megabytes por segundo (que se abreviaría como 0,375 MBps).

Los bits y las tasas de bits (bits a lo largo del tiempo, como bits por segundo [bps]) se usan con más frecuencia para describir las velocidades de conexión, por lo que debe prestar especial atención al comparar proveedores y servicios de conexión a Internet.

Estimando potencias de 2

Cual es el valor de 2^{24}

Cuántos valores puede representar una variable de 32 bit

SUMA

- Decimal

```
   11  ← carries
   3734
 + 5168
 ------
   8902
```

- Binary

```
   11  ← carries
   1011
 + 0011
 ------
   1110
```

Ejemplos de suma binaria

Sume los siguientes números binarios de 4 bits

$$\begin{array}{rr} & \scriptstyle 1 \\ & 1001 \\ + & 0101 \\ \hline & 1110 \end{array}$$

Sume los siguientes números binarios de 4 bits

Overflow!

$$\begin{array}{rr} & \scriptstyle 111 \\ & 1011 \\ + & 0110 \\ \hline & 10001 \end{array}$$

- Los sistemas digitales operan en un número fijo de bits.
- Desbordamiento (Overflow): cuando el resultado es demasiado grande para caber en el número de bits disponibles.

Números binarios signados

- Números signo – magnitud
- Números en complemento 2 (números C-2

Nos. Signo-magnitud

- 1 bit de signo, N-1 bits de magnitud
- El bit de signo es el bit más significativo (más a la izquierda)

-- Número positivo: bit de signo = 0 $A:\{a_{N-1}, a_{N-2}, \cdots a_2, a_1, a_0\}$

-- Número negativo: bit de signo = 1 $A = (-1)^{a_{n-1}} + \sum_{i=0}^{n-2} a_i 2^i$

Ejemplo, representaciones de signo/magnitud de 4 bits de ± 6:

+6 = 0110

- 6 = 1110

Rango de un número de signo/magnitud de N bits:

$[-(2^{N-1}-1), 2^{N-1}-1]$

Nos. Signo-magnitud

Problemas con la representación signo-magnitud:

- No trabaja con la suma de binarios
- Tiene dos representaciones para el cero
- No se usa esta representación en los microprocesadores, es solo una expresión académica para explicar como se puede representar números negativos, PERO NO SE USA EN UN PROCESADOR DE CUALQUIER ARQUITECTURA

Nos. en C-2

- Trabaja con la suma de binarios
- Tiene una representación única para el cero
- Es la representación utilizada en todos los microprocesadores y microcomputadores actuales

Nos. en C-2

- MSB tiene un valor de -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- Número de 4 bits más positivo: 0111
- Número de 4 bits más negativo: 1000
- El bit más significativo aún indica el signo (1 = negativo, 0 = positivo)
- Rango de un número en C-2 de N bits:

  $[-(2^{N-1}), 2^{N-1}-1]$

Obteniendo el C-2

- Invertir el signo de un número a complemento a dos

- Método:
  -- invertir los bits
  -- Añadir 1

Ejemplo: voltear el signo de $3_{10} = 0011_2$

$$
\begin{array}{lcr}
0011 & \longrightarrow & \mathbf{1100} \\
 & & \mathbf{+\quad 1} \\
\hline
 & & \mathbf{1101 = -3_{10}}
\end{array}
$$

Ejemplos C-2

Tome el C-2 de $6_{10} = 0110_2$

1001

+ 1

$1010_2 = -6_{10}$

¿Cuál es el valor decimal del número en C-2 1001_2?

0110

+ 1

$0111_2 = 7_{10}$, entonces $1001_2 = -7_{10}$

Suma en C-2

Sumar 6 + (-6) usando números en complemento a dos

$$\begin{array}{r} 111 \\ 0110 \\ +\ 1010 \\ \hline 10000 \end{array}$$

Sumar -2 + 3 usando números en complemento a dos

$$\begin{array}{r} 111 \\ 1110 \\ +\ 0011 \\ \hline 10001 \end{array}$$

En ambos casos el carry (hacia el quinto bit) se desecha

Aumentando el ancho de bits

Extender el número desde *N* a *M* bits (*M* > *N*) :

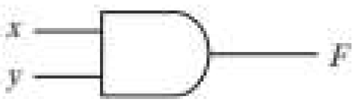
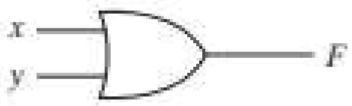
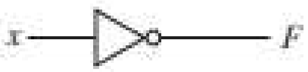
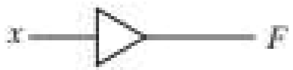
-- extensión del Signo

-- extensión del cero

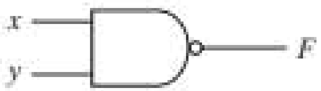
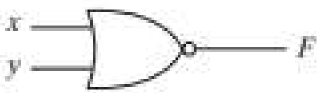
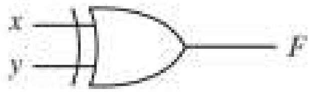
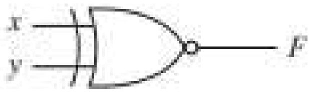
Extensión del signo:

- Bit de signo copiado a MSB
- El valor numérico es el mismo
- Ejemplo 1:
  - -- Representación de 4 bits de 3 = 0011
  - -- Valor de signo extendido de 8 bits: 00000011
- Ejemplo 2:
  - -- representación de 4 bits de -5 = 1011
  - -- Valor de signo extendido de 8 bits: 11111011

Puertas lógicas

Nombre	Símbolo gráfico	Función algebraica	Tabla de verdad
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inversor		$F = x'$	x | F 0 | 1 1 | 0
Búfer		$F = x$	x | F 0 | 0 1 | 1

Puertas lógicas elementales

Nombre	Símbolo gráfico	Función algebraica	Tabla de verdad
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
OR exclusivo (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
NOR exclusivo o equivalencia		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Puertas lógicas de múltiples entradas

NOR3

$$Y = \overline{A+B+C}$$

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

AND4

$$Y = ABCD$$

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Niveles lógicos

Rango de voltajes para 1 y 0
Diferentes rangos para entradas y salidas para superar el ruido.

¿Que es el RUIDO?

Cualquier cosa que degrade la señal. Por ejemplo, resistencia, ruido de la fuente de alimentación, acoplamiento a cables vecinos, etc.

Ejemplo: una puerta emite 5 V pero, debido a la
resistencia en un cable largo, el receptor obtiene 4,5 V. Además se puede introducir voltajes por inducción (ruido=noise) desde otros cables o fuentes eléctricas

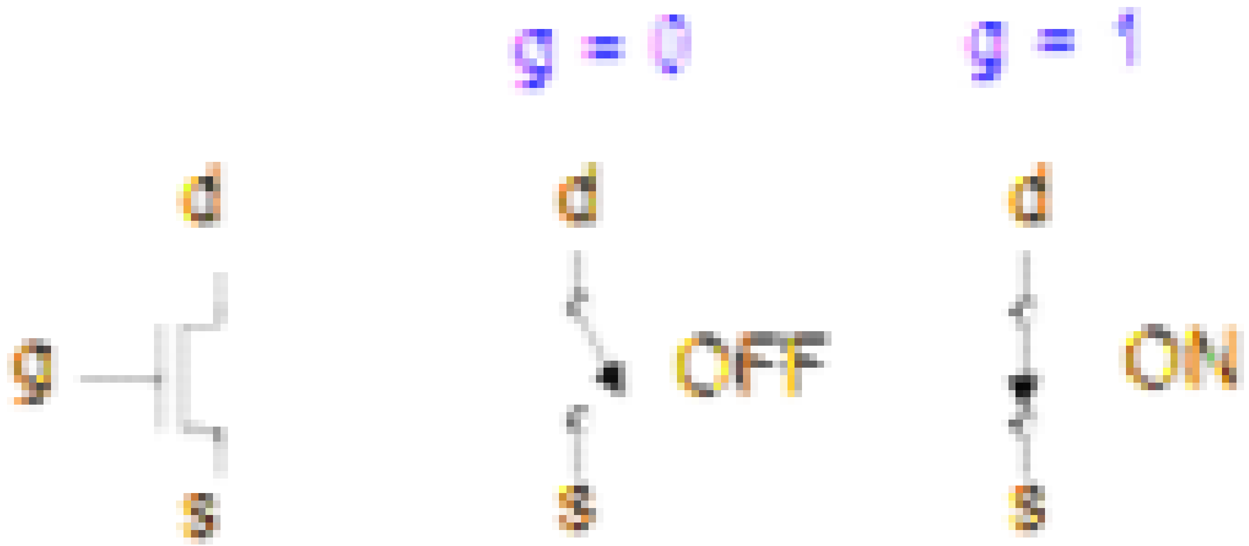
Escalamiento de la fuente de alimentación

V_{DD} representa el voltaje en el pin drenaje (Drain) de un transistor semiconductor de óxido de metal, utilizado para construir la mayoría de los chips modernos.
El voltaje de la fuente de alimentación es también llamado a veces V_{CC}, representando el voltaje en el colector de un transistor bipolar utilizado para construir chips en una tecnología más antigua.
El suelo o tierra o referencia de cero volts en el sistema, a veces se llama V_{SS} porque es el voltaje en la fuente de un transistor semiconductor de óxido de metal.

Logic Family	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
CMOS	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVCMOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7

Transistores

- Puertas lógicas construidas a partir de transistores.
- Interruptor de 3 pines controlado por voltaje
  - -- 2 pines conectados dependiendo del voltaje del 3er pin
  - -- Drain (d) y source (s) están conectados (ON) cuando gate (g) es 1



Fin AOC_01.pptx