TEMA 9

LÓGICA DE CIRCUITOS. CIRCUITOS COMBINACIONALES Y SECUENCIALES

INDICE:

Realizado por Cayetano Borja Carrillo

Tiempo de escritura: 2 horas y 10 minutos

1. INTRODUCCIÓN

A finales del siglo XIX, el matemático británico George Boole desarrolló unas reglas de cálculo para aplicar a la lógica deductiva. Dichas reglas forman el Álgebra de Boole y las operaciones se realizan en base a 2 valores; verdadero y falso.

Como un ordenador digital funciona con impulsos eléctricos y solo entiende 2 estados: cuando pasa corriente eléctrica (se interpreta como “1”) y cuando no (como “0”), se puede aplicar la misma lógica matemática en las ciencias de la computación, donde “falso” se representa con el valor “0” y “verdadero” con el valor “1”.

En este tema se desarrollan los principales conceptos de la lógica de circuitos y los circuitos combinacionales y secuenciales más importantes. Se trata de un tema de gran importancia dentro del campo de estudio del *hardware*, ya que todos los componentes de un ordenador digital están construidos a partir de circuitos digitales.

2. LÓGICA DE CIRCUITOS

La lógica de circuitos es un campo de estudio dentro de la electrónica e informática que se centra en el diseño y análisis de los circuitos digitales aplicando los principios de la lógica booleana. Estos principios lo forman las tablas de verdad, las operaciones lógicas básicas y los axiomas del Álgebra de Boole.

2.1. Tablas de verdad

Una tabla de verdad es la representación del comportamiento de una función lógica. En ella se muestra el valor de salida que tendrá la función para todas las combinaciones de entrada posibles.

Ejemplo: La tabla de verdad de una función “F” con 2 entradas “A” y “B” es la siguiente:

A	B	F (a, b)
F	F	Resultado cuando A y B son falsas
F	V	Resultado cuando solo se cumple B
V	F	Resultado cuando solo se cumple A
V	V	Resultado cuando se cumplen tanto A como B

2.2. Operaciones lógicas básicas

En el Álgebra de Boole se definen las siguientes operaciones básicas:

- Negación: Función que invierte el valor lógico de la variable, es decir, si la función tiene como entrada el valor “Falso”, devuelve “Verdadero” y viceversa. Se interpreta como “NO” (NOT) y su tabla de verdad es la siguiente:

A	F (a) = $\overline{A}$
F	V
V	F

- Unión o suma: Función cuyo resultado es “Verdadero” si se cumple al menos una condición y “Falso” si no se cumple ninguna. Se interpreta como “O” (OR) y su tabla de verdad es la siguiente:

A	B	F (a, b) = A+B
F	F	F
F	V	V
V	F	V
V	V	V

- Intersección o producto: Función cuyo resultado es “Verdadero” si se cumplen todas las condiciones y “Falso” en el resto de los casos. Se interpreta como “Y” (AND) y su tabla de verdad es la siguiente:

A	B	F (a, b) = A · B
F	F	F
F	V	F
V	F	F
V	V	V

- Disyunción o desigualdad material: Función cuyo resultado es “Falso” si todas sus entradas son iguales y “Verdadero” en el resto de casos. Se interpreta como “O-exclusiva” (XOR) y su tabla de verdad es la siguiente:

A	B	F (a, b) = A $\oplus$ B
F	F	F
F	V	V
V	F	V
V	V	F

2.3. Axiomas del Álgebra de Boole

En el Álgebra de Boole se definen una serie de leyes y reglas (axiomas) que se pueden aplicar sobre las funciones lógicas para simplificarlas y reducir la expresión al menor número de términos. Aplicar estos conceptos sobre los circuitos digitales, permite reducir su complejidad y abaratar costes. Dichos axiomas son los siguientes:

Reglas fundamentales

O (OR)	Y (AND)	NO (NOT)
A + 0 = A	A · 0 = 0	$\overline{\overline{A}}$ = A
A + 1 = 1	A · 1 = A	
A + A = A	A · A = A	
A + $\overline{A}$ = 1	A · $\overline{A}$ = 0	

Leyes conmutativas	Leyes distributivas	Leyes asociativas
$A+B = B+A$	$A\cdot(B+C) = (A\cdot B)+(A\cdot C)$	$A+(B+C) = (A+B)+C$
$A\cdot B = B\cdot A$	$A+(B\cdot C) = (A+B)\cdot(A+C)$	$A\cdot(B\cdot C) = (A\cdot B)\cdot C$

Factor común	Teoremas DeMorgan	Absorción
$(A\cdot B)+(A\cdot C) = A\cdot(B+C)$	$\overline{A\cdot B} = \overline{A} + \overline{B}$	$A+(A\cdot B) = A$
$(A+B)\cdot(A+C) = A+(B\cdot C)$	$\overline{A+B} = \overline{A}\cdot\overline{B}$	$A+(\overline{A}\cdot B) = A+B$

Aplicar estos axiomas sobre una función compleja, nos permite obtener una función equivalente más simple. Ejemplo: Simplificar la función "F(a,b) = A·($\overline{A}$+AB) + AB.

$$F(a,b) = A\cdot(\underbrace{A+\overline{A}B}_{A+B}) + AB \rightarrow A(A+B) + AB \rightarrow \underbrace{AA}_{A} + \underbrace{AB + AB}_{AB} \rightarrow \underbrace{A + AB}_{A} \rightarrow A$$

3. CIRCUITOS COMBINACIONALES

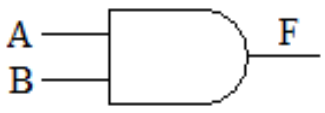
Un circuito combinacional es un circuito digital donde el valor de sus salidas depende exclusivamente de la "combinación" de los valores de sus entradas. Se construyen a partir de unos dispositivos sencillos llamados puertas lógicas.

3.1. Puerta lógica

Una puerta lógica es un chip electrónico que realiza una operación lógica básica. Las puertas lógicas existentes son las siguientes:

- Puerta AND: Realiza el producto lógico de 2 o más valores. Permite el paso de corriente (1) si por todas sus entradas circula electricidad (1) y corta el paso de la corriente (0) en el resto de los casos.

Símbolo



Expresión

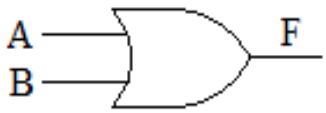
$F(a,b) = A\cdot B$

Tabla de verdad

A	B	F(a,b)
0	0	0
0	1	0
1	0	0
1	1	1

- Puerta OR: Realiza la suma lógica. Devuelve "0" si todas sus entradas tienen el valor "0" y devuelve "1" en el resto de los casos.

Símbolo



Expresión

$F(a,b) = A+B$

Tabla de verdad

A	B	F(a,b)
0	0	0
0	1	1
1	0	1
1	1	1

- Puerta XOR: Realiza la disyunción lógica. Devuelve “0” cuando todas sus entradas tienen el mismo valor y “1” en el resto de casos.

Símbolo

Expresión

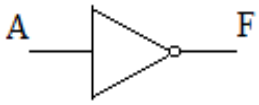
$F(a,b) = A \oplus B$

Tabla de verdad

A	B	F(a,b)
0	0	0
0	1	1
1	0	1
1	1	0

- Puerta NOT: Realiza la negación lógica. Devuelve el valor contrario al valor de su entrada.

Símbolo



Expresión

$F(a) = \overline{A}$

Tabla de verdad

A	F(a)
0	1
1	0

- Puerta NAND: Representa una puerta AND invertida. Los valores de salida son los opuestos de los de una puerta AND.

Símbolo

Expresión

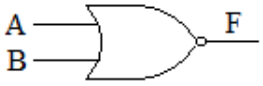
$F(a,b) = \overline{A \cdot B}$

Tabla de verdad

A	B	F(a,b)
0	0	1
0	1	1
1	0	1
1	1	0

- Puerta NOR: Representa una puerta OR invertida.

Símbolo



Expresión

$F(a,b) = \overline{A + B}$

Tabla de verdad

A	B	F(a,b)
0	0	1
0	1	0
1	0	0
1	1	0

- Puerta XNOR: Es la versión invertida de una puerta XOR.

Símbolo

Expresión

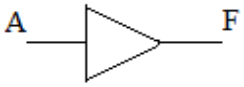
$F(a,b) = \overline{A \oplus B}$

Tabla de verdad

A	B	F(a,b)
0	0	1
0	1	0
1	0	0
1	1	1

- Puerta IF: La puerta lógica IF (SI) devuelve el mismo valor que el de su entrada. La función principal de esta puerta lógica es la de amplificar la corriente suministrada para adaptar impedancias.

Símbolo



Expresión

$F(a) = A$

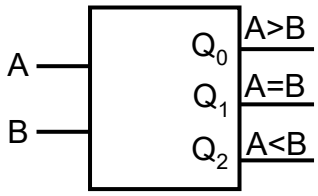
Tabla de verdad

A	F(a)
0	0
1	1

3.2. Comparador

Un comparador es un circuito combinacional compuesto de 2 entradas "A" y "B" y 3 líneas de salida "Q_i". Su función es la de comparar el valor de sus 2 entradas y dirigir el flujo de corriente por una de sus 3 salidas, dependiendo de si "A>B", "A=B" o "A<B".

Bloque funcional



Circuito combinacional

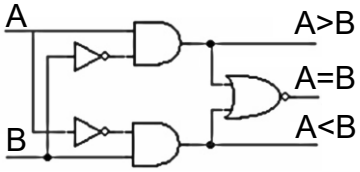


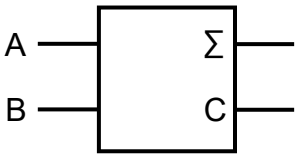
Tabla de verdad

A	B	Q_0	Q_1	Q_2
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

3.3. Semisumador y sumador

Un semisumador es un circuito combinacional que realiza una suma binaria. Se compone de 2 entradas "A" y "B" y dos líneas de salida "∑" y "C". La salida "∑" devuelve el resultado de la suma de sus entradas y la salida "C" el acarreo.

Bloque funcional



Circuito combinacional

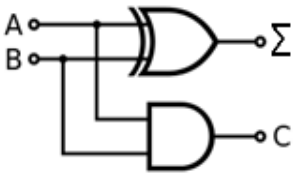


Tabla de verdad

A	B	C	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

El sumador completo es un semisumador que incluye como entrada, además de A y B, el acarreo de la operación anterior (C_{IN}). Con varios sumadores completos en serie se pueden hacer sumas de varios bits, ya que contempla los acarreos.

Bloque funcional

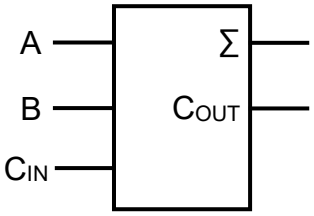


Tabla de verdad

A	B	C_{IN}	C_{OUT}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

3.4. Decodificador y codificador

Un decodificador es un circuito que transforma un código binario a decimal. Se compone de "n" entradas "E_i" y puede tener "2^n" líneas de salida "Q_i" como máximo. El decodificador dirigirá el flujo de corriente por una única salida. La salida activa será la equivalente decimal correspondiente al valor binario que forman sus entradas.

Ejemplo de un decodificador de 2 entradas y 4 salidas:

Bloque funcional

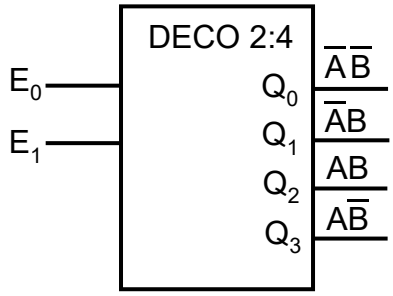


Tabla de verdad

E_0	E_1	Q_0	Q_1	Q_2	Q_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

Como se puede observar en la tabla de verdad, la combinación de entradas "10" activa únicamente la salida "2", ya que $10_{(2} = 2_{(10}$.

Un codificador hace la acción opuesta de un decodificador, transforma un código decimal a binario. Se compone de "2^n" entradas como máximo y "n" líneas de salida. Ejemplo de un codificador 4:2.

Bloque funcional

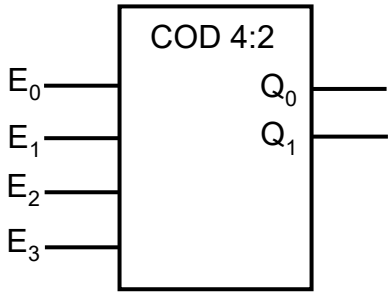


Tabla de verdad

E_0	E_1	E_2	E_3	Q_0	Q_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

3.5. Multiplexor y demultiplexor

Un multiplexor es un circuito compuesto de "n" entradas de selección "S_i", "2^n" entradas de datos "E_i" y una única salida "Q". Dependiendo del valor de las entradas de selección "S", se transmitirá hacia la salida "Q" el valor de una y solo una entrada de datos "E". Ejemplo de un multiplexor de 2 entradas de selección y 4 de datos.

Bloque funcional

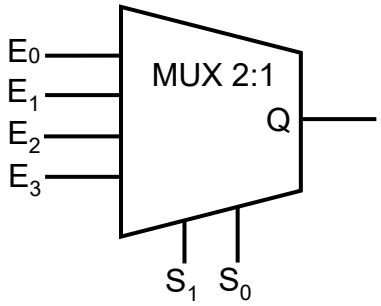


Tabla de verdad

S_1	S_0	Q
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

Un demultiplexor realiza la operación contraria a un multiplexor, es decir, tiene una única entrada de datos "E", "n" entradas de selección "S_i" y "2^n" líneas de salida "Q_i". Dependiendo del valor de las entradas de selección "S", se transmitirá el valor de la entrada de datos "E" por una salida u otra. Ejemplo de un DEMUX 2:4.

Bloque funcional

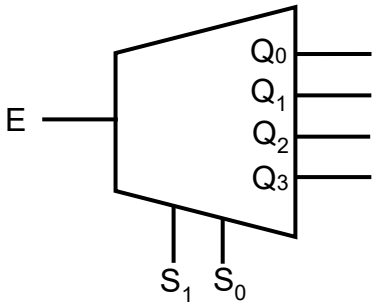


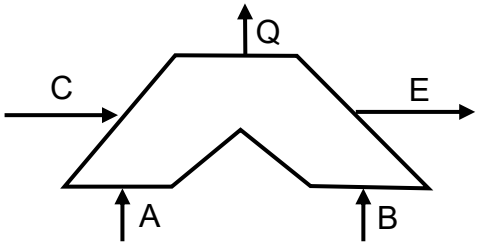
Tabla de verdad

S_0	S_1	Q_0	Q_1	Q_2	Q_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

3.6. Unidad aritmético-lógica

La unidad aritmético-lógica o ALU (*Arithmetic-Logic Unit*) es un circuito combinacional compuesto de puertas lógicas y otros circuitos más sencillos como semisumadores y multiplexores. Se encarga de realizar operaciones aritméticas (suma, restas, etc.) y lógicas simples (AND, OR, etc.) entre uno o 2 valores.

A continuación, se muestra el bloque funcional de una ALU y su descripción:



- Las entradas “A” y “B” transmiten los operandos. Pueden ser de varios bits.
- La entrada de control “C” indica la operación a realizar. También de varios bits.
- La salida “Q” transporta el resultado de la operación.
- La salida “E” transmite condiciones que se quedaron pendientes en la anterior operación, como el acarreo.

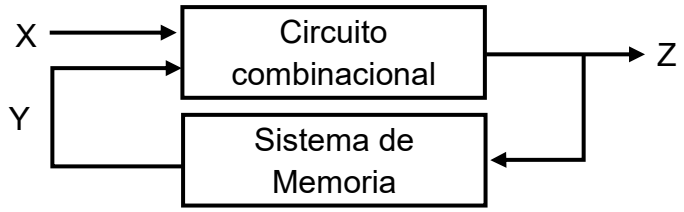
Ejemplo de funcionamiento: Supongamos una ALU de 2 bits donde el tipo de operaciones que puede realizar, dependiendo del valor de “C”, son las siguientes:

Entrada “C”	Operación
00	XOR
01	AND
10	OR
11	Suma

La ALU en un determinado momento recibe los valores “A=00”, “B=01” y “C=11”. Realiza la operación “00+01” y los valores de salida serán “Q=01” y “E=00” ya que no existe acarreo.

4. CIRCUITOS SECUENCIALES

Un circuito secuencial es un circuito digital donde el valor de sus salidas depende de la combinación de los valores de sus entradas y del resultado de operaciones anteriores. Se construye a partir de un circuito combinacional al que se le añade un sistema de memoria que permite la retroalimentación.



Un circuito secuencial puede ser síncrono si el cambio de estado lo determina una señal de sincronización generada por un reloj o asíncrono si el cambio de estado está determinado por la propia naturaleza de las puertas lógicas que se utilizan.

Para el estudio de los circuitos secuenciales se utiliza el cronograma, que es la representación gráfica de la evolución en el tiempo de las entradas y salidas, mostrándose el valor de cada variable en cada cambio de estado.

4.1. Biestable

Un biestable es un circuito secuencial capaz de mantener uno de dos estados posibles (un cero o un uno) durante un periodo de tiempo. Se utilizan varios biestables conectados en cascada para construir memorias, por ejemplo, las memorias RAM estáticas (SRAM) que se encuentran en la memoria caché y en los registros internos del microprocesador se crean a partir de biestables.

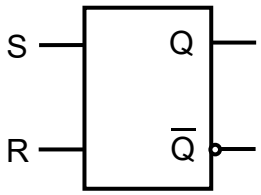
Dependiendo de su sincronización, un biestable se puede clasificar en 2 tipos: cerrojos y flip-flops.

Cerrojo o *latch*

Un cerrojo es un biestable asíncrono, es decir, está retroalimentado por sí mismo.

Existen 2 tipos de cerrojos: el cerrojo D (*Delay*) y el cerrojo SR (*Set-Reset*). El SR es el más simple y se construye a partir de puertas lógicas NOR. La salida "Q" representa al bit almacenado, mientras que "$\overline{Q}$" se usa solamente para la retroalimentación. Ejemplo:

Bloque funcional



Circuito secuencial

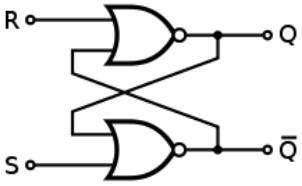


Tabla de verdad

S	R	Q
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	Error

"Q_{n-1}" significa que se mantiene el valor anterior.

Flip-Flop

Un *flip-flop* es un biestable síncrono compuesto de un cerrojo más una señal sincronizadora, que determina el cambio de estado.

Existen varios tipos de *flip-flop*, siendo el más sencillo el SR. Este *flip-flop* consiste en un cerrojo SR al que se le añaden 2 puertas AND para crear una tercera entrada "C" que actúa como señal sincronizadora. Ejemplo:

Bloque funcional

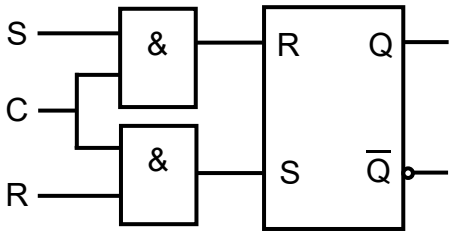


Tabla de verdad

C	S	R	Q
0	X	X	Q_{n-1}
1	0	0	Q_{n-1}
1	0	1	1
1	1	0	0
1	1	1	Error

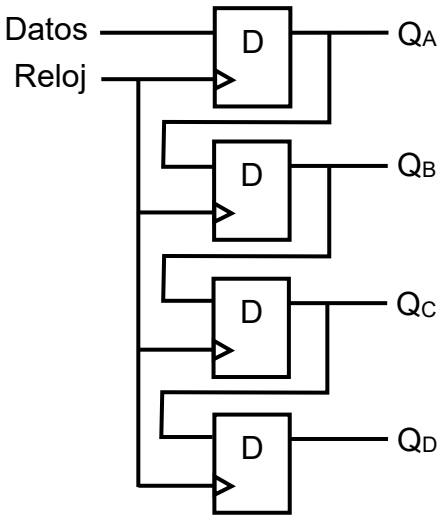
"X" significa que no importa su valor ya que al ser C=0, no hay cambio de estado.

4.2. Registro de desplazamiento

Un registro de desplazamiento es un circuito secuencial compuesto de varios biestables conectados en cascada, que basculan de forma sincrónica con la misma señal de reloj.

Según las conexiones de los biestables, un registro puede tener un desplazamiento a la izquierda, a la derecha o en ambos sentidos. El desplazamiento a la izquierda se utiliza para multiplicar por 2 y a la derecha para dividir entre 2.

A continuación, se muestra un registro de desplazamiento de 4 bits construido a partir de 4 *flip-flops* de tipo D.



Supongamos que las salidas “Q” contienen los datos “0100” (4 en decimal). Si se desplaza a la izquierda, hace una multiplicación y se obtiene “1000” (8 en decimal).

4.3. Contador

Un contador es un circuito compuesto de varios *flip-flops* conectados en cascada, cuya función es la de contar y almacenar el número de impulsos eléctricos que recibe en su entrada.

El número máximo de impulsos que puede contar es de “2^n”, siendo “n” el número de bits o cantidad de *flip-flops* que contiene. Una vez llega al máximo valor, se empieza a contar un ciclo nuevo desde 0.

El bloque funcional, la tabla de verdad y el cronograma de un contador de 2 bits construido con *flip-flops* de tipo JK es el siguiente:

Circuito funcional

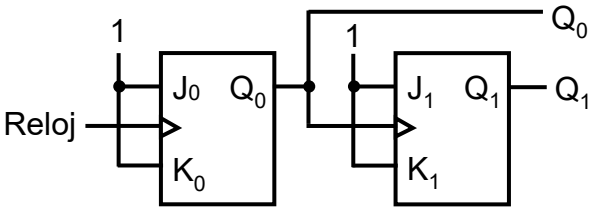
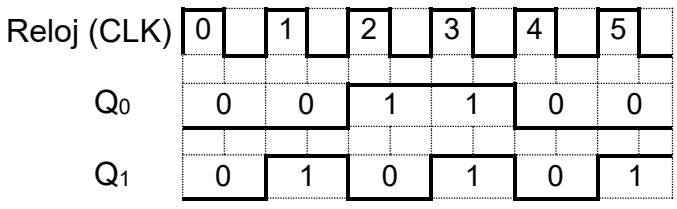


Tabla de verdad

Q_0	Q_1	Valor decimal
0	0	0
0	1	1
1	0	2
1	1	3

Cronograma



5. CONCLUSIÓN

La lógica de circuitos y las operaciones y reglas que se definen en el Álgebra de Boole son fundamentales en el diseño de los circuitos digitales, ya que permite obtener circuitos lo más simplificados y optimizados posibles.

Gracias a las puertas lógicas, que realizan una operación booleana, es posible crear circuitos combinacionales y secuenciales capaces de realizar operaciones simples como sumas, descodificar datos o multiplexarlos.

La combinación y conexión de varios de estos dispositivos sencillos permite construir componentes más complejos como una unidad aritmético-lógica, memorias y, en definitiva, cualquier dispositivo *hardware* como puede ser un microprocesador o una tarjeta gráfica.

6. BIBLIOGRAFÍA

- López Ureña, L. A. et al. (1997). *Fundamentos de Informática (1ª ed.)*. Ra-ma.
- Prieto Espinosa, A. et al. (2006). *Introducción a la informática (4ª ed.)*. McGraw-Hill.
- Brookshear, J. G. (2012). *Introducción a la computación (11ª ed.)*. Pearson Educación.
- Castro Gil, M. et al. (2002). *Estructura y Tecnología de computadores I (1ª ed.)*. Universidad Nacional de Educación a Distancia (UNED)
- Dormido, S. et al. (2000). *Estructura y Tecnología de computadores (2ª ed.)*. Editorial Sanz y Torres
- Canalfdet. (2019). *Circuitos combinacionales*. Youtube. Recuperado de https://www.youtube.com/@FdetEs

7. NORMATIVA

Para el desarrollo de este tema, se ha tenido en cuenta la siguiente normativa, donde se especifican los contenidos, competencias y criterios de evaluación de los Ciclos Formativos y Bachillerato en Andalucía:

- Orden 7 de julio de 2009 (SMR).
- Orden 19 de julio de 2010 (ASIR).
- Orden 16 de junio de 2011 (DAW/DAM).
- Instrucción 13/2022 (Bachillerato).