

10_secuenciales_sincronos_mef

June 19, 2026

1 Diseño de Circuitos Secuenciales Síncronos y Máquinas de Estados (MEF)

Electrónica Digital — 2º GIERM (Mecatrónica) Apuntes propios — Tema 10. Basado en ED10-Diseño Circuitos Secuenciales Síncronos y ED10-MEF-VHDL (Universidad de Sevilla).

1.1 Índice

1. Concepto: circuito secuencial síncrono y MEF
2. Estructura: Moore vs. Mealy
3. Metodología de diseño (6 pasos)
4. Worked example 1 — Conmutador (Moore)
5. Worked example 2 — Control de temperatura (3 estados) + VHDL
6. Asignación / codificación de estados
7. Implementación de MEF en VHDL (plantilla de examen)
8. Resumen y checklist de examen

Cómo usar estos apuntes: cada bloque va *worked-example primero* — primero el ejemplo resuelto entero, luego la teoría que lo sustenta. Las figuras (diagrama de bolas y cronograma) se dibujan con `matplotlib` desde cero.

Utilidades de dibujo cargadas.

1. Concepto: circuito secuencial síncrono y MEF

Un **circuito secuencial** se diferencia de uno **combinacional** en que su salida depende no solo de las entradas actuales, sino también de la **historia** (memoria). Esa memoria se concentra en el **estado**.

Máquina de Estados Finitos (MEF / FSM): - El **estado** engloba la información actual y pasada relevante para el funcionamiento. - Evoluciona por sí mismo o en función de **entradas** externas. - Físicamente se codifica en **biestables** (flip-flops). El número de estados es **finito**. - El estado **no cambia de forma continua**, sino sincronizado con un flanco de **reloj** -> *circuito síncrono*.

Esquema general de la evolución del estado:

Entradas ->	Circuito de	Q(n+1)	Biestables	--> Estado actual Q(n)
	determinacion	----->	(registro)	
	de Q(n+1)	<-----+ <-----+ (realimentacion)		

+-----+ Estado actual $Q(n)$

El estado futuro $Q(n+1)$ es función combinacional del estado actual $Q(n)$ y de las entradas. Cada flanco de reloj, los biestables capturan $Q(n+1)$ y lo convierten en el nuevo $Q(n)$.

2. Estructura: Moore vs. Mealy

Toda MEF síncrona se descompone en tres bloques:

Bloque	Significado	Naturaleza
CCE	Circuito Combinacional de Entrada	Combinacional -> calcula $Q(n+1)$ a partir de $Q(n)$ y entradas
Biestables	Registro de estado	Síncrono -> guardan Q y avanzan con el reloj
CCS	Circuito Combinacional de Salida	Combinacional -> genera las salidas

1.1.1 Diferencia Moore / Mealy

MOORE	MEALY
-----	-----
Entradas->CCE->[Q]->CCS->Salidas	Entradas-->CCE->[Q]-->CCS->Salidas
	+-----+
Salida = f(estado)	Salida = f(estado, entradas)

- **Moore:** la salida depende **solo del estado**. La salida se escribe *dentro* de la bola. Más estable, salida síncrona con el estado.
- **Mealy:** la salida depende del **estado Y de las entradas**. La salida se escribe *sobre la transición* (arco).
- **Regla práctica:** las máquinas de Mealy tienen igual o menos estados que las de Moore para la misma funcionalidad (la salida puede “anticiparse” en la transición).

3. Metodología de diseño (6 pasos)

Esta es la receta que hay que aplicar en el examen, en orden:

#	Paso	Bloque
1	Diagrama de bolas (diagrama de estados)	Diseño
2	Tabla de transición de estados (TTE)	Diseño
3	Reducción / fusión de estados (eliminar estados equivalentes)	Diseño
4	Asignación (codificación) de estados -> binario en biestables	Implementación

#	Paso	Bloque
5	CCE — Circuito Combinacional de Entrada (ecuaciones de D/Q+)	Implementación
6	CCS — Circuito Combinacional de Salida (ecuaciones de salidas)	Implementación

Elementos del diagrama de bolas: - **Bola (círculo)** = estado. Nombre dentro; y si es **Moore**, también el valor de la salida. - **Arco** = transición. Siempre tiene origen y destino; se etiqueta con la **condición** (valor de entradas) que la dispara. - El origen puede coincidir con el destino -> **autolazo** = *acción de espera* (el estado se mantiene mientras se cumple la condición).

4. Worked example 1 — Conmutador (máquina de Moore)

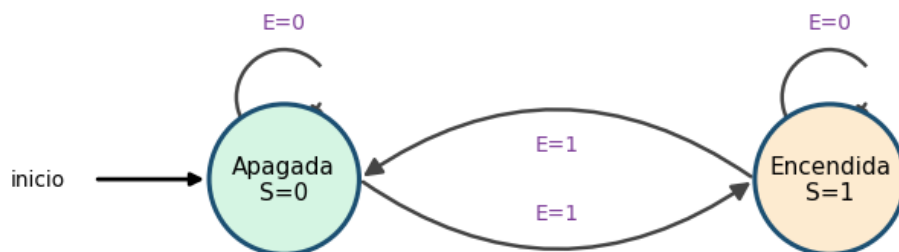
Enunciado. Un conmutador enciende o apaga la luz cada vez que se pulsa, dependiendo de si estaba apagada o encendida. Entrada E (pulsador), salida S (luz).

1.1.2 Paso 1 — Diagrama de bolas

Necesitamos recordar si la luz está **apagada** o **encendida** -> 2 estados. La salida depende solo del estado -> **Moore**.

- Apagada (S=0): con E=0 se queda; con E=1 (pulsación) pasa a Encendida.
- Encendida (S=1): con E=0 se queda; con E=1 pasa a Apagada.

Ejemplo 1 - Conmutador (maquina de MOORE)



1.1.3 Paso 2 — Tabla de transición de estados (TTE)

Estados: A = Apagada, B = Encendida. La tabla da el **estado siguiente** según E, y la salida S (Moore -> solo depende del estado).

Estado actual	E=0	E=1	S (Moore)
A (Apagada)	A	B	0
B (Encendida)	B	A	1

1.1.4 Paso 3 — Reducción

No hay estados equivalentes (tienen distinta salida), así que **no se reduce**: quedan 2 estados.

1.1.5 Paso 4 — Codificación

Con 2 estados basta **1 biestable** (Q): A -> Q=0, B -> Q=1.

1.1.6 Paso 5 — CCE (estado futuro)

Con biestable tipo D, $D = Q(n+1)$. Tabla codificada:

Q(n)	E	Q(n+1)=D	S
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

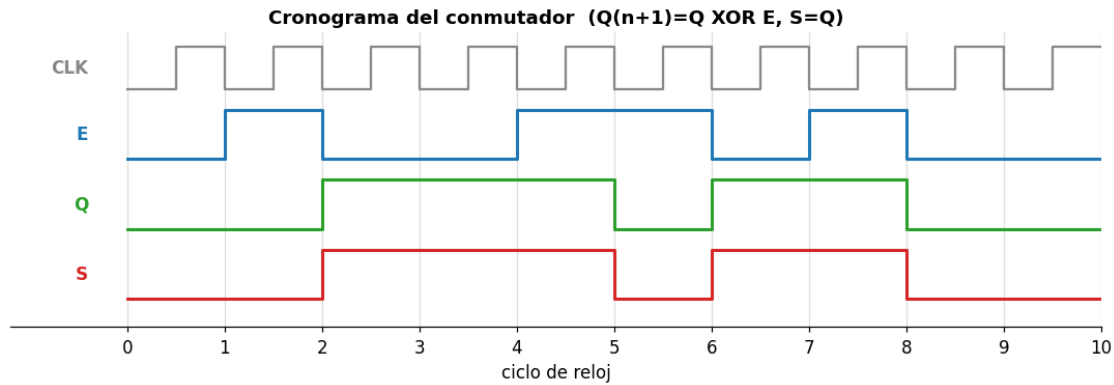
De la tabla, $Q(n+1) = Q \text{ (XOR) } E \rightarrow D = Q \text{ XOR } E$.

1.1.7 Paso 6 — CCS

$S = Q$ (la salida es directamente el estado, por ser Moore con esta codificación).

Circuito final: un flip-flop D con $D = Q \text{ XOR } E$ y salida $S = Q$. Un solo flip-flop y una XOR.

Ciclo: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
E : [0, 1, 0, 0, 1, 1, 0, 1, 0, 0]
Q : [0, 0, 1, 1, 1, 0, 1, 1, 0, 0]
S : [0, 0, 1, 1, 1, 0, 1, 1, 0, 0]



5. Worked example 2 — Control de temperatura (3 estados) + VHDL

Este es el ejemplo que se lleva a **VHDL** en los apuntes (`tempercontrol`).

Enunciado. Un sistema de control de temperatura tiene tres rangos representados por las entradas T1, T2, T3 (temperatura baja / media / alta). Hay **3 estados** Q1, Q2, Q3 (modos de actuación) y **2 salidas** C1, C2.

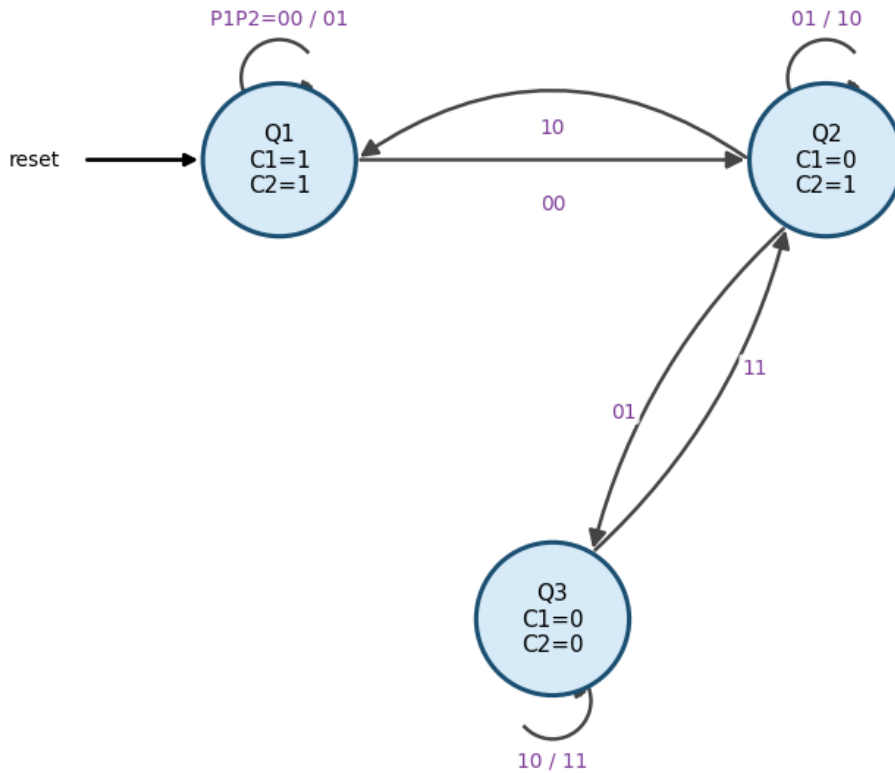
1.1.8 Codificación de las entradas

Como no todas las combinaciones de T1, T2, T3 son posibles, se codifican en **2 bits** P1, P2 mediante un codificador previo. La MEF trabaja con P1, P2.

1.1.9 Diagrama de bolas (Moore, 3 estados)

Transiciones según P1P2 (de la TTE del código VHDL; los casos imposibles van marcados):

Ejemplo 2 - Control de temperatura (MOORE, 3 estados)



1.1.10 Tabla de Transición de Estados (TTE)

Estado	P1P2=00	P1P2=01	P1P2=10	P1P2=11	C1	C2
Q1	Q1	Q1	Q2	(Q1)*	1	1
Q2	Q1	Q2	Q2	Q3	0	1
Q3	(Q1)*	Q2	Q3	(Q3)*	0	0

(.)* = combinación imposible de entradas (se rellena por seguridad / *don't care*).

Salidas (Moore): C1 = 1 solo en Q1; C2 = 0 solo en Q3 (es decir, C2=1 en Q1 y Q2).

1.1.11 Codificación de estados (paso 4)

3 estados -> **2 biestables**. Una asignación binaria típica: Q1=00, Q2=01, Q3=10 (la combinación 11 queda sin usar -> *don't care*).

Clave VHDL: como verás abajo, en VHDL **no elegimos la codificación a mano** — se declara un **type** enumerado y el sintetizador asigna los bits. Solo describimos la *lógica* (la TTE y las salidas), no las ecuaciones del CCE/CCS.

6. Asignación / codificación de estados — por qué importa

La **asignación de estados** es traducir cada estado simbólico a un patrón de bits en los biestables. Afecta a: - **Coste hardware** (complejidad del CCE/CCS según el Karnaugh resultante). - **Riesgo de glitches** y consumo.

Estrategias clásicas: - **Binaria secuencial:** 00, 01, 10, 11... -> mínimo n° de biestables ($\lceil \log_2 N \rceil$). - **One-hot:** un biestable por estado (Q1=001, Q2=010, Q3=100) -> más flip-flops pero CCE más simple y rápido; habitual en **FPGA**. - **Gray:** minimiza transiciones de bits entre estados adyacentes.

En diseño **manual** (Karnaugh) la asignación cambia las ecuaciones del CCE; en **VHDL** se delega al sintetizador (puedes sugerir el estilo con atributos, p.ej. `attribute fsm_encoding`).

7. Implementación de MEF en VHDL (plantilla de examen)

La estructura Moore/Mealy se traduce a **3 bloques de VHDL**, que reproducen exactamente el esquema CCE -> biestables -> CCS:

Bloque hardware	En VHDL	Tipo de proceso
CCE	proceso asíncrono que calcula estadosig (= Q(n+1)) a partir de estado y entradas	combinacional (sensible a entradas + estado)
Biestables	proceso síncrono que hace estado <= estadosig en el flanco de reloj (+ reset)	secuencial (sensible a clk , reset)
CCS	asignación combinacional que fija las salidas según estado	combinacional

Patrón de **tres procesos** (recomendado y el del enunciado). Lo más importante: declarar un tipo enumerado de estados.

```
type state_type is (Q1, Q2, Q3);    -- estados simbolicos: el sintetizador codifica
signal estado      : state_type;    -- estado actual (salida de los biestables)
signal estadosig   : state_type;    -- estado siguiente (= Q(n+1), salida del CCE)
```

1.1.12 Código completo tempercontrol (control de temperatura)

```
library ieee;
use ieee.std_logic_1164.all;

entity tempercontrol is
  port (p1, p2      : in  std_logic;    -- entradas codificadas
        c1, c2      : out std_logic;    -- salidas (Moore)
        clk, reset  : in  std_logic);   -- el reloj y el reset SON entradas en VHDL
```

```

end tempercontrol;

architecture temparch of tempercontrol is
    type state_type is (Q1, Q2, Q3); -- los estados posibles
    signal estado      : state_type;  -- estado actual
    signal estadorig    : state_type;  -- estado siguiente
begin

    -- (1) CCE: proceso ASINCRONO -> equivale a la TTE
    process(p1, p2, estado)
    begin
        case estado is
            when Q1 => if    p1='0' and p2='0' then estadorig <= Q1;
                        elsif p1='0' and p2='1' then estadorig <= Q1;
                        elsif p1='1' and p2='0' then estadorig <= Q2;
                        elsif p1='1' and p2='1' then estadorig <= Q1; -- caso imposible
                        end if;
            when Q2 => if    p1='0' and p2='0' then estadorig <= Q1;
                        elsif p1='0' and p2='1' then estadorig <= Q2;
                        elsif p1='1' and p2='0' then estadorig <= Q2;
                        elsif p1='1' and p2='1' then estadorig <= Q3;
                        end if;
            when Q3 => if    p1='0' and p2='0' then estadorig <= Q1; -- caso imposible
                        elsif p1='0' and p2='1' then estadorig <= Q2;
                        elsif p1='1' and p2='0' then estadorig <= Q3;
                        elsif p1='1' and p2='1' then estadorig <= Q3; -- caso imposible
                        end if;
            when others => estadorig <= Q1; -- por seguridad
        end case;
    end process;

    -- (2) BIESTABLES: proceso SINCRONO -> actualiza el estado en cada flanco
    process(reset, clk)
    begin
        if    reset='1'           then estado <= Q1;           -- reset asincrono
        elsif rising_edge(clk) then estado <= estadorig;       -- captura Q(n+1)
        end if;
    end process;

    -- (3) CCS: combinacional -> salidas como funcion del estado (Moore)
    with estado select
        c1 <= '1' when Q1, '0' when others;
    with estado select
        c2 <= '0' when Q3, '1' when others;

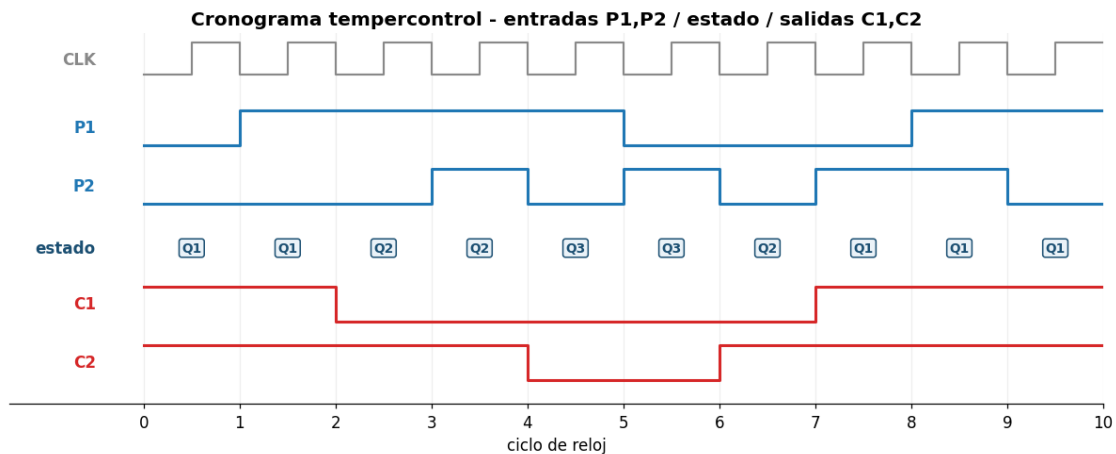
end temparch;

```


1.1.13 Notas de examen sobre el VHDL

- **clk y reset deben aparecer como entradas** en la **entity**, aunque conceptualmente “no se cuentan” como entradas del circuito.
- El **CCE** es **asíncrono**: su lista de sensibilidad incluye **todas las entradas y el estado** (p1, p2, estado). Si olvidas una señal, el simulador no la actualiza (latch indeseado).
- El **proceso síncrono** distingue **reset='1'** (reset **asíncrono** prioritario) de **rising_edge(clk)**.
- En **Mealy**, el CCS dependería también de las entradas: `c1 <= ... when (estado=Qx and entrada=...).`
- Para **simular**: `ghdl -a fichero.vhd (analiza) -> ghdl -r tb --stop-time=20us --wave=out.ghw (simula) -> gtkwave out.ghw`. El **testbench** no tiene puertos: instancia el DUT, genera **clk** (p.ej. periodo 1 us) y las entradas, y observa las salidas.

estado	P1	P2	C1	C2	->	sig
Q1	0	0	1	1	->	Q1
Q1	1	0	1	1	->	Q2
Q2	1	0	0	1	->	Q2
Q2	1	1	0	1	->	Q3
Q3	1	0	0	0	->	Q3
Q3	0	1	0	0	->	Q2
Q2	0	0	0	1	->	Q1
Q1	0	1	1	1	->	Q1
Q1	1	1	1	1	->	Q1
Q1	1	0	1	1	->	Q2



8. Resumen y checklist de examen

1.1.14 Mapa mental

Problema -> Diagrama de bolas -> TTE -> Reduccion -> Codificacion -> CCE + CCS
 (Moore: salida en bola) (manual: Karnaugh)

(Mealy: salida en arco)

(VHDL: type enumerado)

1.1.15 Checklist

- ☐ ¿Cuántos estados mínimos necesito? (¿qué historia debo recordar?)
- ☐ ¿**Moore** (salida = f(estado)) o **Mealy** (salida = f(estado, entradas))?
- ☐ Diagrama de bolas: todo arco con condición; autolazos para “espera”; estado de reset.
- ☐ TTE completa: una columna por combinación de entradas; marcar casos imposibles (*don't care*).
- ☐ Reducir estados equivalentes (misma salida y mismas transiciones).
- ☐ Codificar: `ceil(log2 N)` biestables; usar *don't cares* en Karnaugh.
- ☐ CCE: ecuaciones de D/Q+. CCS: ecuaciones de salidas.
- ☐ En **VHDL**: 3 procesos (CCE asíncrono, biestables síncronos con reset, CCS combinacional); `clk` y `reset` como entradas; `type state_type` enumerado.

1.1.16 Errores típicos

- Olvidar una señal en la lista de sensibilidad del CCE -> latches no deseados.
- Confundir Moore/Mealy al colocar la salida (bola vs. arco).
- No definir el **when others** -> estado indeterminado en simulación.
- Contar mal los biestables (recordar `ceil(log2 N)`).

Apuntes propios — Tema 10 Electrónica Digital (2º GIERM). Worked-examples: Conmutador (Moore) y tempercontrol (3 estados -> VHDL).