

**EX9** d'Electrònica Digital (1BT4)**Com funciona un indicador de panell?**

*(Exemple de problema estructurat per ser desenvolupat per l'equip cooperatiu base)*

**Temps total d'estudi estimat: 8h**

**A) Objectiu:** Disseny d'un indicador de direcció basat en una matriu de punts. Aplicació del mètode canònic per dissenyar FSM (mínims 7) en aplicacions reals.

**B) La teoria:** Imprimiu i estudeu les la unitat 2.5: el mètode canònic de disseny de FSM de Moore. Obriu el projecte Proteus-VSM del panell que acompanya aquest exercici (vegeu la Fig. 1 i assageu-lo per la freqüència de rellotge de a) 10kHz; b) 1Hz. Expliqueu com funciona la matriu de punts.

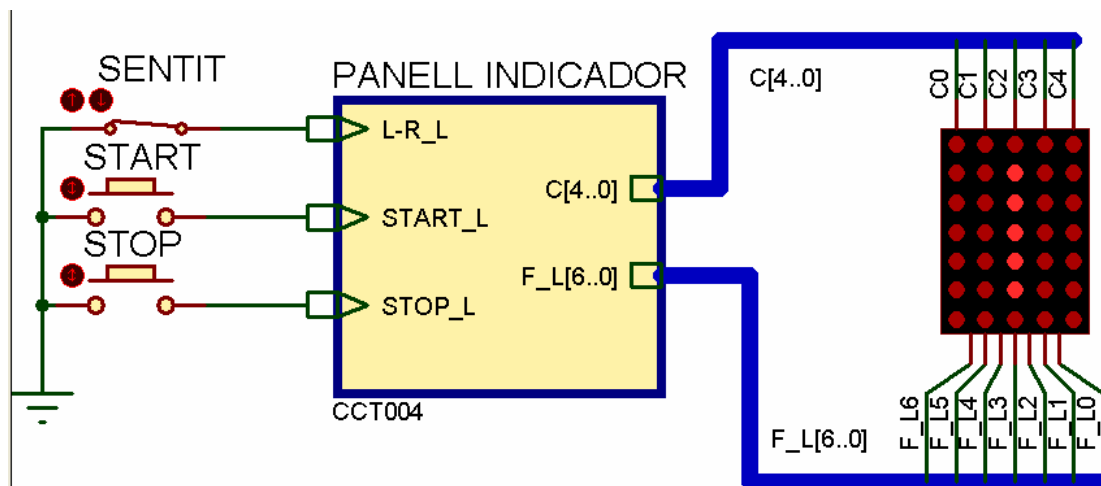


Fig. 1 Esquema elèctric del panell funcionant i mostrant la il·luminació de LED's de la columna C3

**C) El mètode:**

**1) Treball cooperatiu del grup base a la sessió de treball TGA (1,5h):**

El prof. repassa les línies generals del mètode canònic de disseny de FSM indicades a la Unitat 2.5 i explica l'anàlisi del funcionament i el plantejament del disseny intern del circuit de l'indicador representat a la Fig. 2

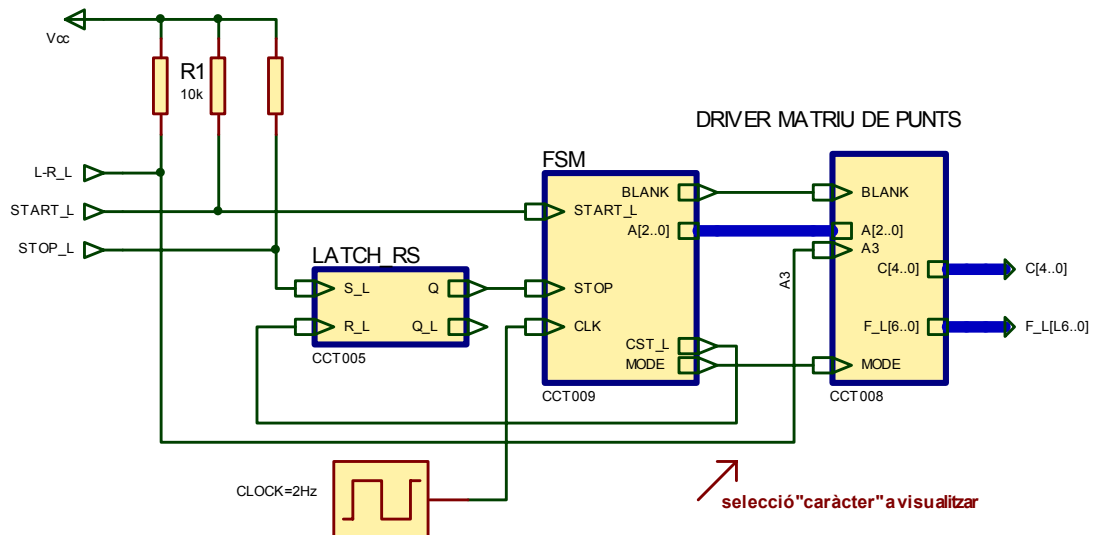


Fig. 2 Disseny intern del controlador de panell

- El “driver” d’excitació de la matriu de punts de 5×7 representat a la Fig. 3.

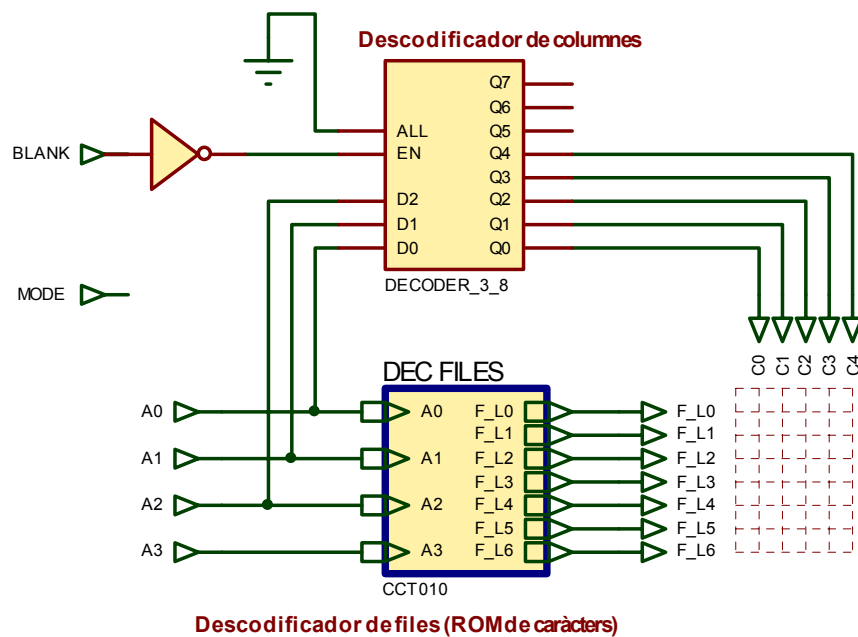


Fig. 3 Esquema del driver de la matriu de punts

- El condicionament dels pulsadors d’entrada
- La feina que ha de resoldre la FSM i el mètode canònic per dissenyar-la:
  - Especificacions i diagrama d’estats
  - Arquitectura general de la FSM de Moore pel mètode canònic (vegeu la Fig. 4)
  - Codificació d’estats (binary, Gray, etc.)
  - Disseny del SC2
  - Dibuix de la memòria d’estat amb *flip-flops* D

- f) Disseny del SC1 a través de la taula de transició d'estats; obtenció de les funcions de sortida a través de la taula de disseny del flip-flop D; realització amb qualsevol mètode dels explicats en el Tema 1
- g) Dibuix del circuit general en Proteus-VSM
- h) Verificació del funcionament.

Els estudiants, treballant en grup base realitzen el disseny intern del bloc *driver* de la matriu de punts i del *latch* RS amb entrades actives a nivell baix. També proposen un cronograma que indica com funciona el *latch*.

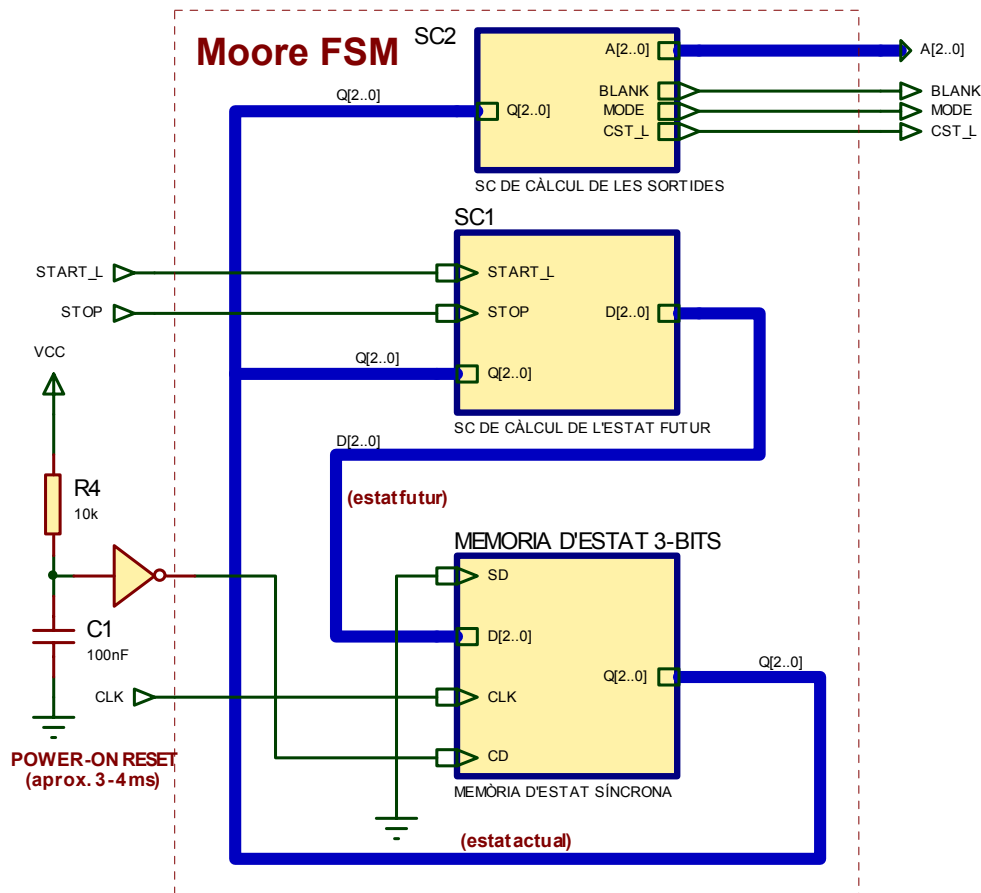


Fig. 4 FSM síncrona

## 2) Treball cooperatiu del grup base a les sessions TGC (2h):

Els estudiants, treballant en grup base realitzen el disseny de la FSM des de l'apartat b) fins al f).

El grup realitzarà una proposta alternativa del disseny intern del sistema segons:

- Estudiant A: Disseny amb *flip-flops* JK
- Estudiant B: Disseny amb *flip-flops* T
- Estudiant C: Codificació d'estats en binari i disseny amb D

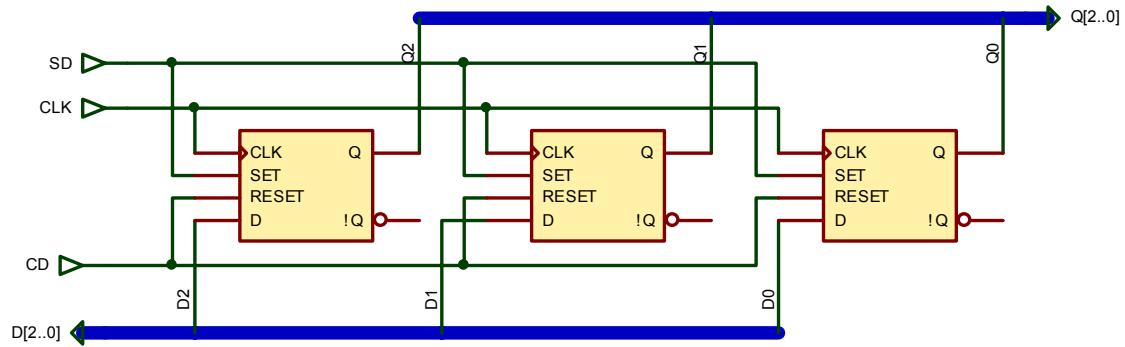


Fig. 5 Cel·la de memòria d'estat síncrona

Cada estudiant explicarà als seus companys de grup com funciona i es dissenya el seu circuit. Heu d'apuntar els dubtes que han sortit.

L'estudiant A documenta el treball realitzat per B; l'estudiant B documenta el treball realitzat per C; i l'estudiant C documenta el treball realitzat per A.

### 3) Treball cooperatiu del grup base a la sessió de treball TGB (1h):

Durant la sessió es resolen els dubtes que han sortit dissenyant la FSM.

El prof. explica la forma alternativa que s'ha usat en aquest projecte exemple per dissenyar el descodificador de files a través del llenguatge VHDL (vegeu la Fig. 7) i gravant un xip programable GAL22V10 (Fig. 8). Aquests conceptes correspondrien a la realització en VHDL de sistemes combinacionals: Unitats 1.14 – 1.15 (sessions de demostració).

El prof. explica com es dissenyaria la FSM d'una forma alternativa a través del VHDL i com es gravaria en un sPLD GAL22V10. Aquests conceptes correspondrien a la realització en VHDL de sistemes seqüencials: Unitats 2.6 – 2.7 (sessions de demostració).

### 4) Treball cooperatiu del grup base a la sessió de treball TGA (1,5h):

El grup realitzarà la tasca d'ampliació del sistema dissenyant el senyal MODE que permet visualitzar mentre dura l'estat S7 totes les columnes de LEDS simultàniament permetent així observar tot el caràcter representat complet abans d'acabar el cicle.

A la vista del projecte dissenyat, cerqueu informació de com es podrien visualitzar més símbols a la pantalla de punts, per exemple tot els caràcters numèrics d'un teclat telefònic.

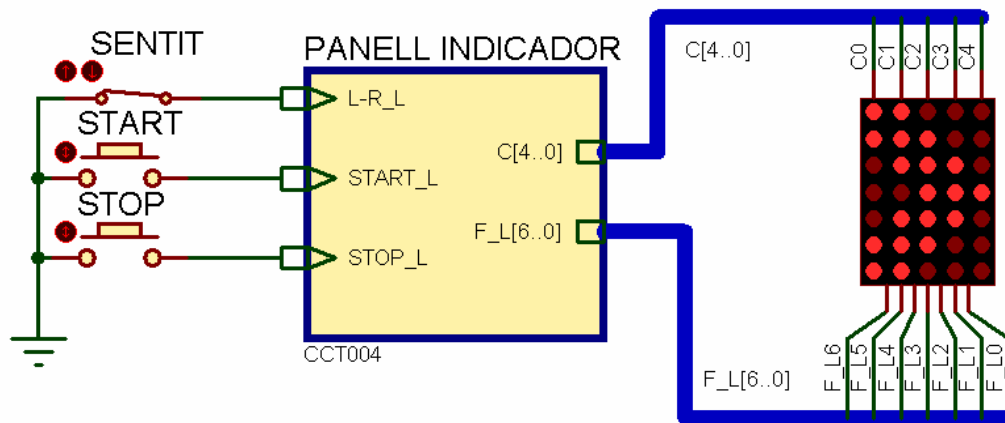


Fig. 6 Visualització de tot el caràcter complet

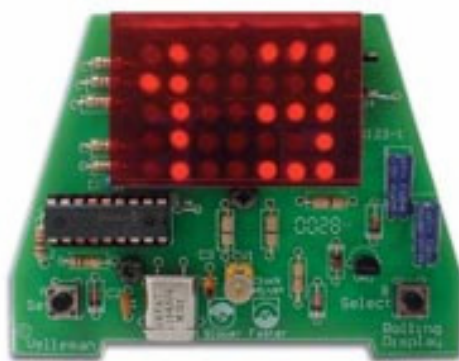
#### D) Els resultats:

##### 5) Treball cooperatiu del grup base a les sessions TGC (2h):

La verificació en Proteus-VSM d'un dels vostres dissenys i la documentació final de l'exercici.

#### NOTA D'APLICACIÓ:

A l'assignatura SED del 2A (troncal a Telemàtica i optativa a Sistemes de Telecomunicació) podreu realitzar un parell de versions més avançades d'aquest projecte: amb xips programables PLD (Tema 1) i amb microcontroladors PIC (Tema 2). És a dir, dissenyareu tot el projecte en un sol xip tal com podeu trobar en comerços de productes electrònics. Aquí teniu un parell d'exemples



Kit de rellotge rodant (Diotronic S.A., Codi MK123, 17,75 €). Permet la visualització de la data i l'hora gràcies a la matriu de 35 LED's.



Kit de missatge rodant (Diotronic S.A., Codi MK124, 17,75 €). Pots crear els teus caràcters

## Annex: El fitxer VHDL del descodificador de files i el xip GAL22V10 gravat

```

dec_files.vhd*
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY dec_files IS
    PORT (
        A      : IN    STD_LOGIC_VECTOR(3 DOWNTO 0);
        F_L    : OUT   STD_LOGIC_VECTOR(6 DOWNTO 0)
    );
END dec_files;
ARCHITECTURE Arch_Taula_Veritat OF dec_files IS
BEGIN
    PROCESS (A)
    BEGIN
        CASE A IS
            -- A3,A2,A1,A0   F6 F5 F4 F3 F2 F1 F0
            -- Caràcter Direcció dreta ->
            WHEN "0000" => -- Columna 0
                F_L <= NOT("1100011");
            WHEN "0001" => -- Columna 1
                F_L <= NOT("1110111");
            WHEN "0010" => -- Columna 2
                F_L <= NOT("0111110");
            WHEN "0011" => -- Columna 3
                F_L <= NOT("0011100");
            WHEN "0100" => -- Columna 4
                F_L <= NOT("0001000");
            -- Caràcter direcció esquerra <-
            WHEN "1000" =>
                F_L <= NOT("0001000");
            WHEN "1001" =>
                F_L <= NOT("0011100");
            WHEN "1010" =>
                F_L <= NOT("0111110");
            WHEN "1011" =>
                F_L <= NOT("1110111");
            WHEN "1100" =>
                F_L <= NOT("1100011");
            WHEN OTHERS =>
                F_L <= NOT("0000000"); --Apagat
        END CASE;
    END PROCESS;
END Arch_Taula_Veritat;

```

Fig. 7 Fitxer que defineix el descodificador de files en llenguatge VHDL

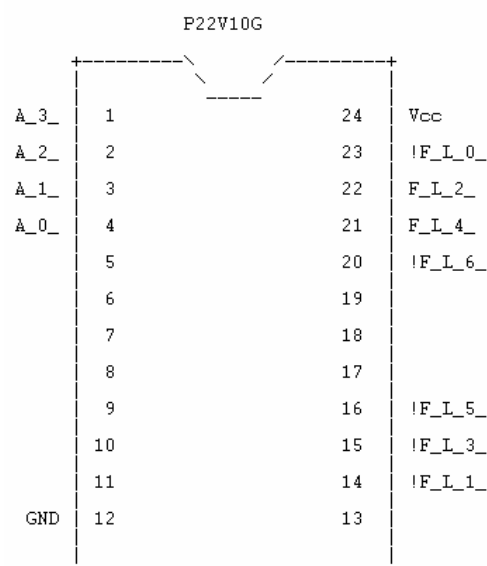


Fig. 8 Encapsulat de la sPLD GAL22V10 que conté el descodificador de files una vegada compilat amb el programari ispLEVER