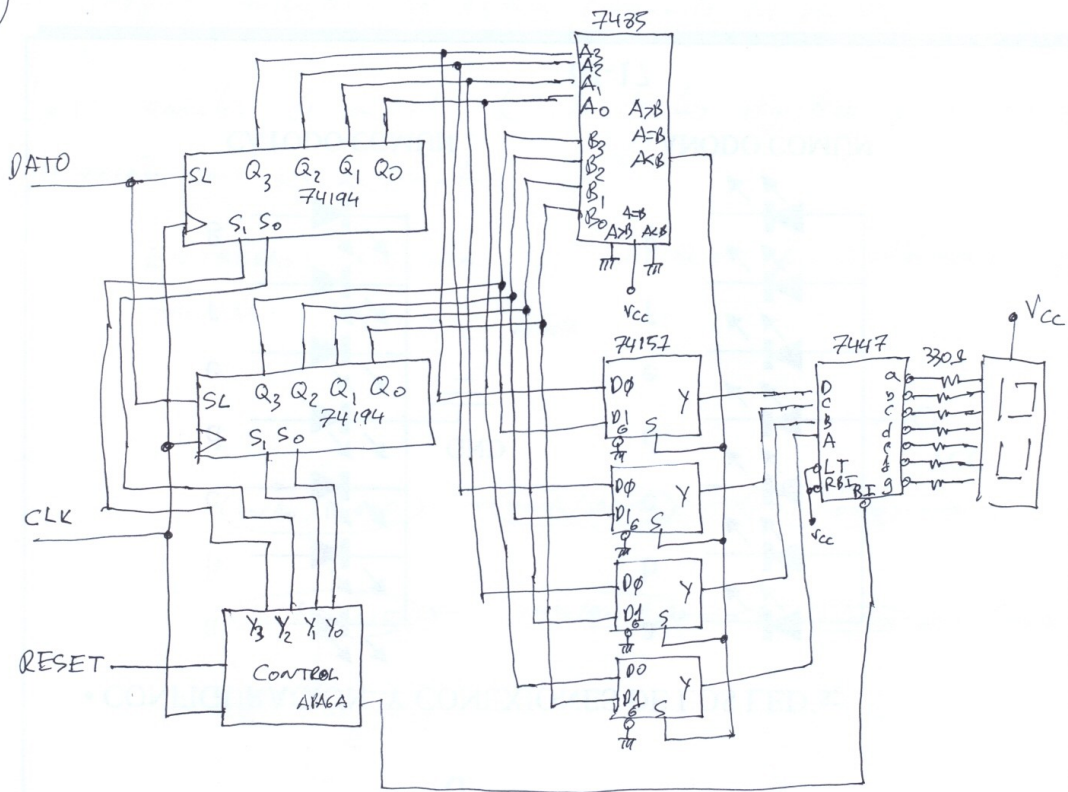


1)



Durante los 4 pulsos de reloj después del reset se cargará el dato que llega por la línea serie en el registro de desplazamiento superior. Para ello se irá desplazando a derecha de forma que A_0 empieza a entrar por la izquierda y va pasando de Q_3 hasta llegar a Q_0 . Pasados 4 pulsos de reloj tenemos el dato completo almacenado en el registro y se mantiene ya sin cambios ($S_1, S_0 = 00$).

Durante los 4 pulsos siguientes se hace lo mismo pero en el registro de desplazamiento inferior, almacenando así $B_3 \dots B_0$.

Una vez que los dos datos están almacenados se sacará el mayor.

El 7485 los compara y el cuádruple multiplexor selecciona el mayor de los dos. Además se añadirá una línea que indique

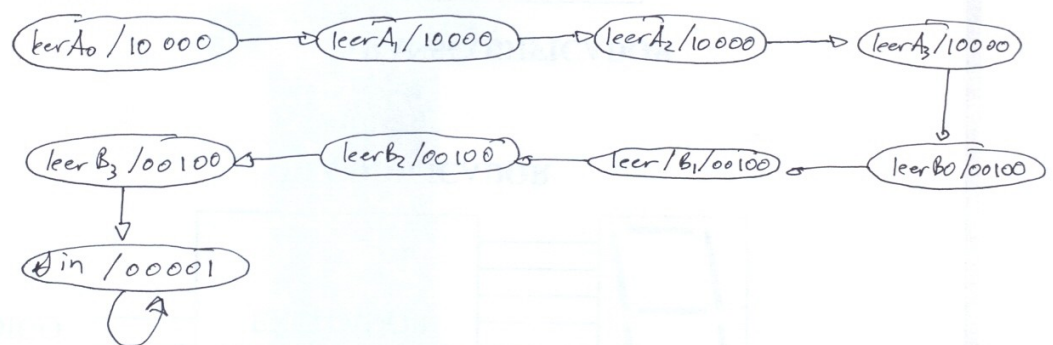
1

el display mientras se están cargando los datos.

Las señales de control (S_1, S_0 de los registros $\times$ APA6A) se generarán con un autómata.

ENTRADAS: NINGUNA (EL RESET forzará el estado inicial).

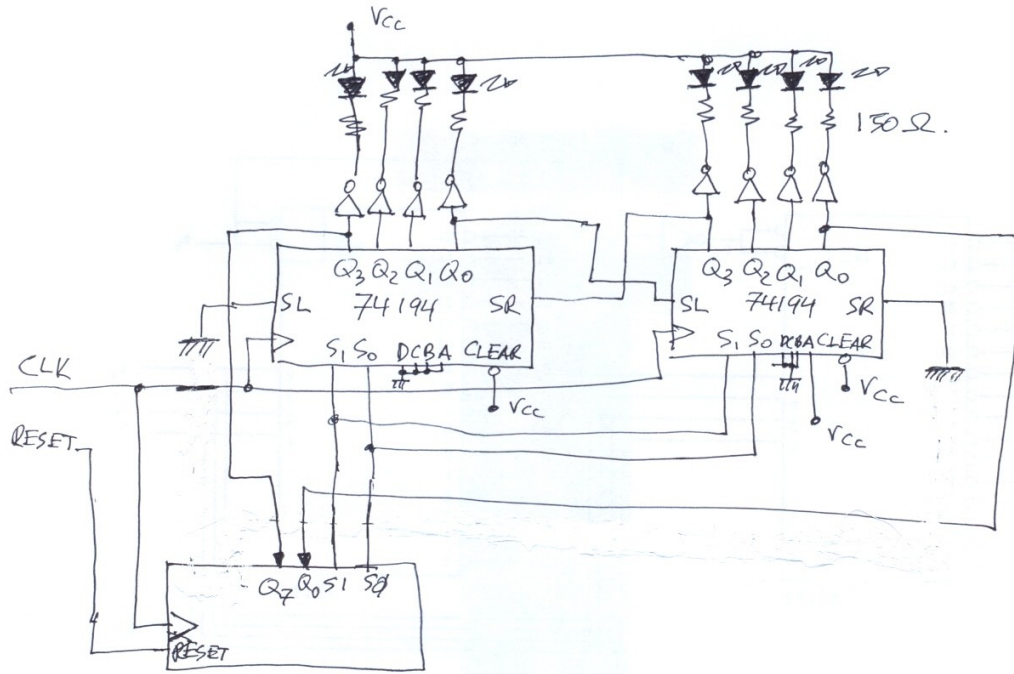
SALIDAS: Y_3, Y_2, Y_1, Y_0 APA6A



Todas las transiciones son incondicionales (ya que no hay entradas).

Durante cada estado se lee un bit y se envía al registro correspondiente y durante el último estado se mantienen los datos y se enciende el display, permaneciendo así hasta que se produzca un reset.

3) Conectaré dos 74194 en cascada de forma que el bit de mayor peso del registro de la derecha entre como bit de menor peso al registro de la izquierda (en desplazamiento a izquierda) y el bit de menor peso del registro de la izquierda entre como bit de mayor peso



al registro de la derecha (en desplazamiento a derecha)
Como las salidas del 74194 no pueden entregar 20mA
($I_{OH} = 0.4mA$ e $I_{OL} = 8mA$) utilizaré un buffer inversor

$$R = \frac{5V - V_{LED}}{20mA} = 150\Omega$$

En la inicialización los registros cargarán 0000 0001, se harán a través de sus entradas paralelo. Luego el registro debe girar a derecha (siempre entrando ceros por la izquierda) hasta que $Q_0 = 1$ en ese instante se cambia el sentido de giro y se empieza a rotar a la izquierda (siempre entrando ceros por la derecha) hasta que $Q_7 = 1$ y se vuelve a

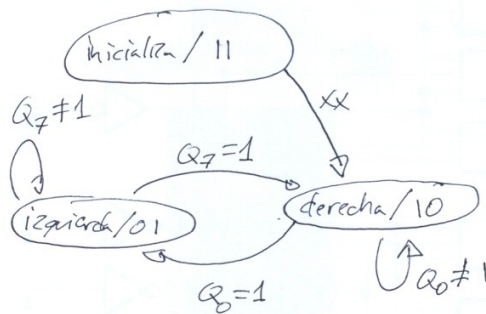
repetir la secuencia

El reset obligará al autómata a ir a su estado inicial.

El autómata tendrá como entradas y salidas:

Entradas Q_7 Q_0

Salidas S_1 S_0



La programación en abel es la siguiente:

```
MODULE CONTROL;
```

```
Q7, Q0, S1, S0 PIN; // entradas y salidas
```

```
E1, E0 NODE IS TYPE 'reg'; // estado
```

```
CLK, RESET PIN;
```

```
inicializa = [0, 0]; izquierda = [0, 1]; derecha = [1, 0]; // estados.
```

equations

```
[E1, E0].CLK = CLK;
```

```
[E1, E0].AR = RESET;
```

```
STATE_REGISTER [E1, E0]
```

```
STATE inicializa:
```

```
S1 = 1; S0 = 1;
```

```
GOTO derecha;
```

```
STATE derecha:
```

```
S1 = 1; S0 = 0;
```

```
IF (Q0 == 1) THEN izquierda ELSE derecha;
```

```
STATE izquierda:
```

```
S1 = 0; S0 = 1;
```

```
IF (Q7 == 1) THEN derecha ELSE izquierda;
```

3) b) $a b \bar{c} d \bar{a} + a b c d + a \bar{b} c e + \underbrace{[(\bar{a} + \bar{b})(a \cdot b)(c + d + e)]}_{\bar{a} + \bar{b} = \overline{a \cdot b}} =$

$$= a b c d + \overline{a \bar{b} c e} = a b c d + \bar{a} + b + \bar{c} + e =$$

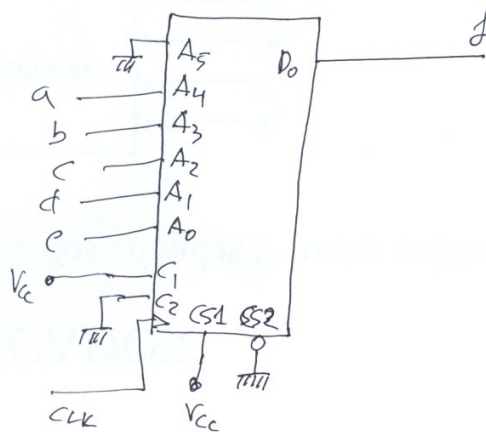
$$= \bar{a} + b + \bar{c} + e$$

a)

abcde	f
00000	1
00001	1
00010	1
00011	1
00100	1
00101	1
00110	1
00111	1
01000	1
01001	1
01010	1
01011	1
01100	1
01101	1
01110	1
01111	1
10000	1
10001	1
10010	1
10011	1
10100	0
10101	1
10110	0
10111	1
11000	1
11001	1
11010	1
11011	1
11100	1
11101	1
11110	1
11111	1

c) $f(a, b, c, d, e) = (\bar{a} + b + \bar{c} + d + e) \cdot (\bar{a} + b + \bar{c} + \bar{d} + e)$

d) La más pequeña de las tres (y sobra) es la ER2055



El contenido de la memoria (82 primeras posiciones) será la tabla de verdad donde f será el valor de D0.