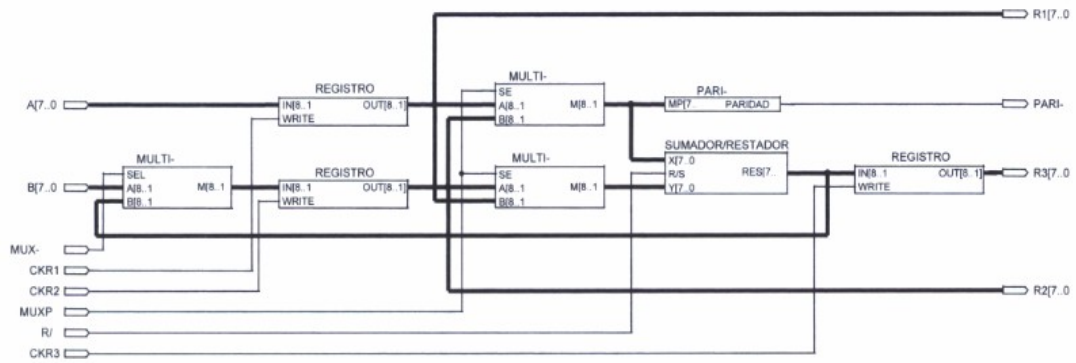
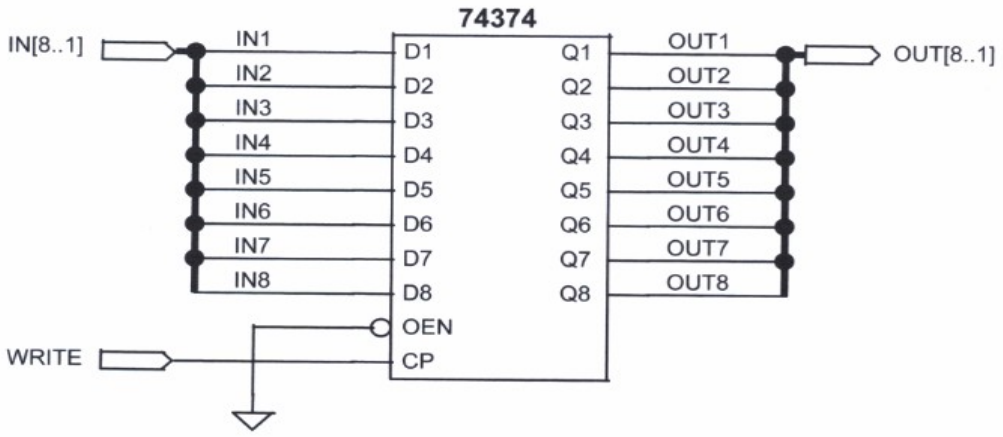
Problema 5.8 La arquitectura del sistema debe estar formada por una unidad de control que guíe la secuencia del ciclo de operaciones y una unidad de proceso que pueda realizar dichas operaciones. Dicha unidad de proceso deberá contar con tres registros (R1, R2 y R3), un sumador y un restador o un sumador/restador, un circuito comprobador de paridad y los multiplexores necesarios para guiar el flujo de datos. Un posible diagrama de bloques de esta unidad de proceso se puede ver en la siguiente figura:

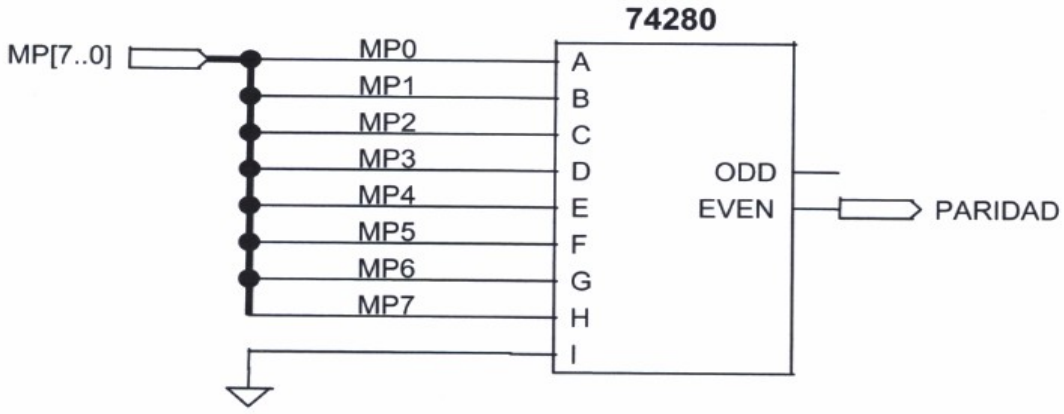


Como se puede ver necesitamos tres multiplexores: Uno para acceder al registro R2 desde el exterior o desde la salida del sumador/restador y los otros dos para disponer los datos A y B en el orden deseado. De esta forma podemos realizar A-B o B-A con el mismo circuito sumador/restador.

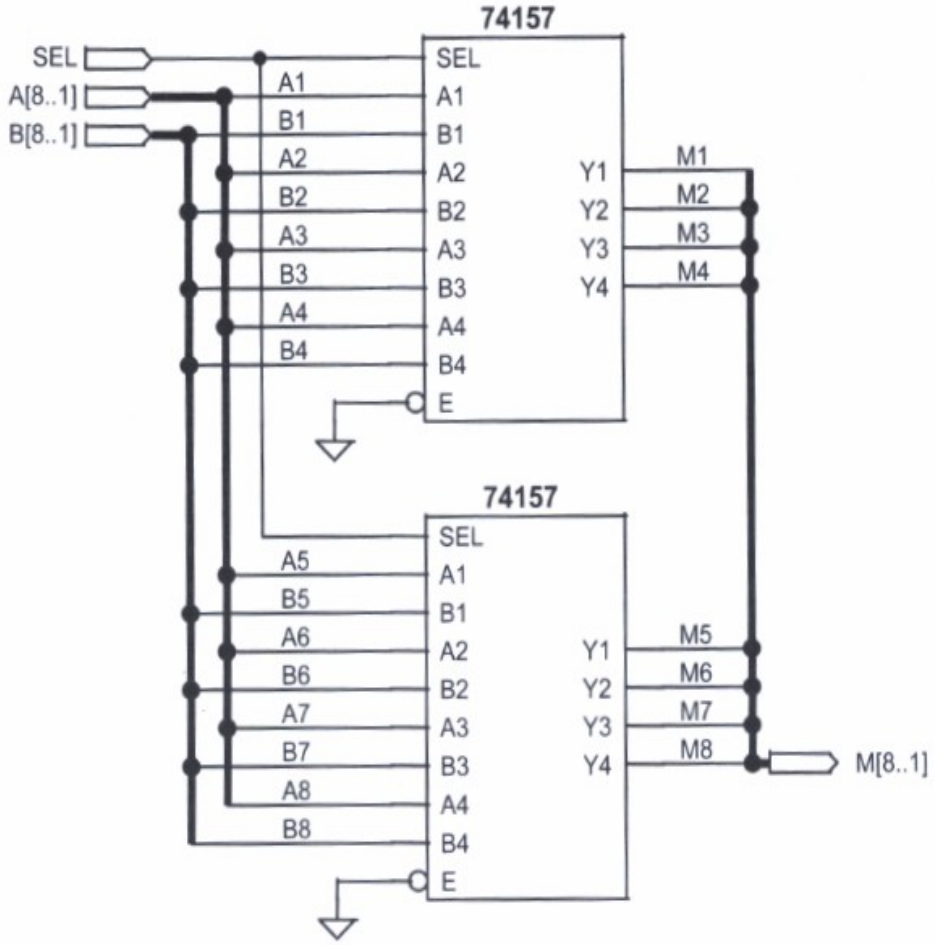
Vamos a ver ahora en detalle cada uno de estos bloques. Para realizar los registros se han utilizado los 74HC374. El esquema de los tres es el mismo:



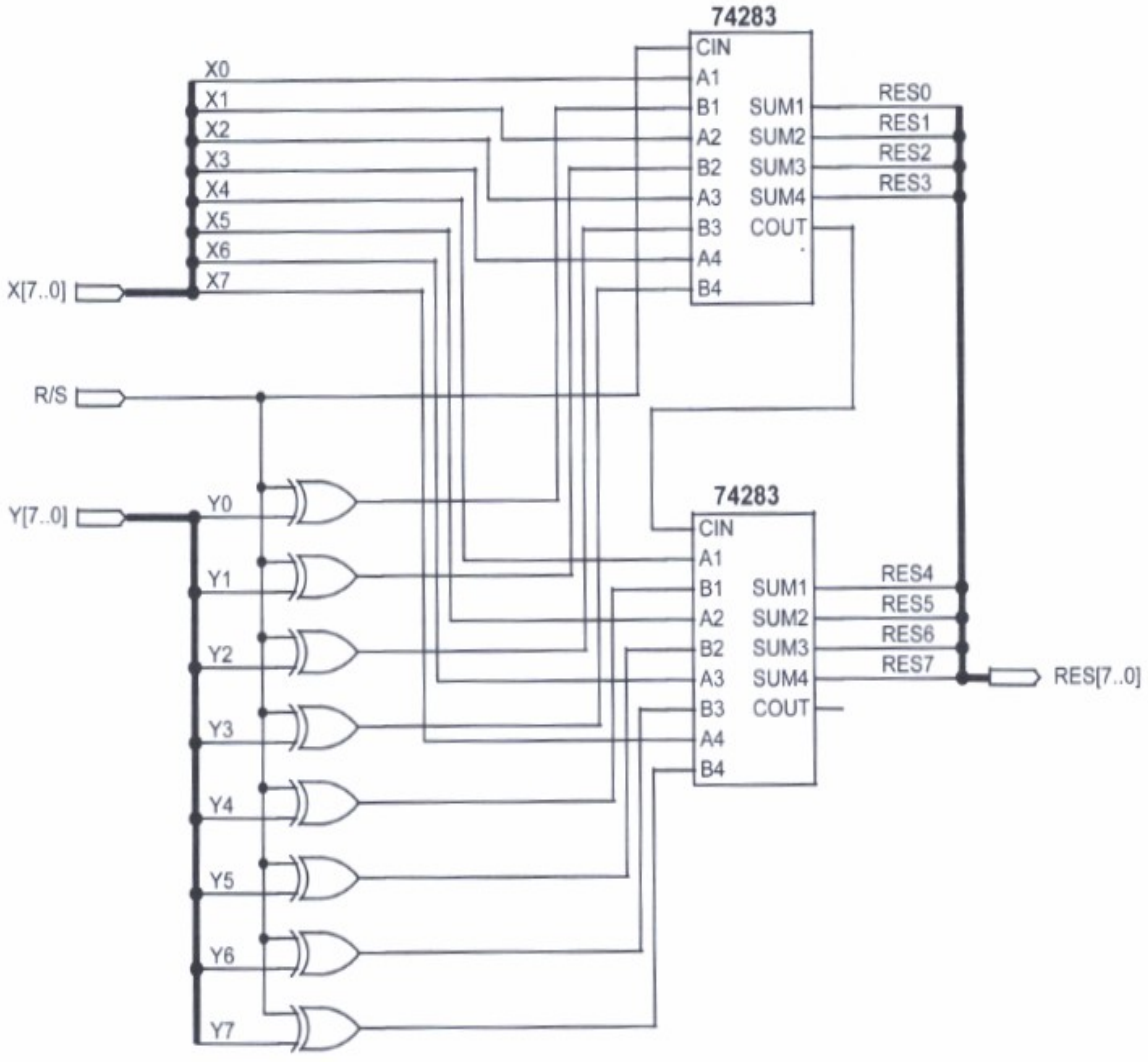
El generador de paridad se basa en el circuito 74HC280 que se debe conectar de la siguiente forma:



Para los multiplexores se han utilizado dos circuitos 74HC157 agrupados de forma que obtenemos un multiplexor de dos entradas de 8 bits tal y como se puede ver en la siguiente figura:

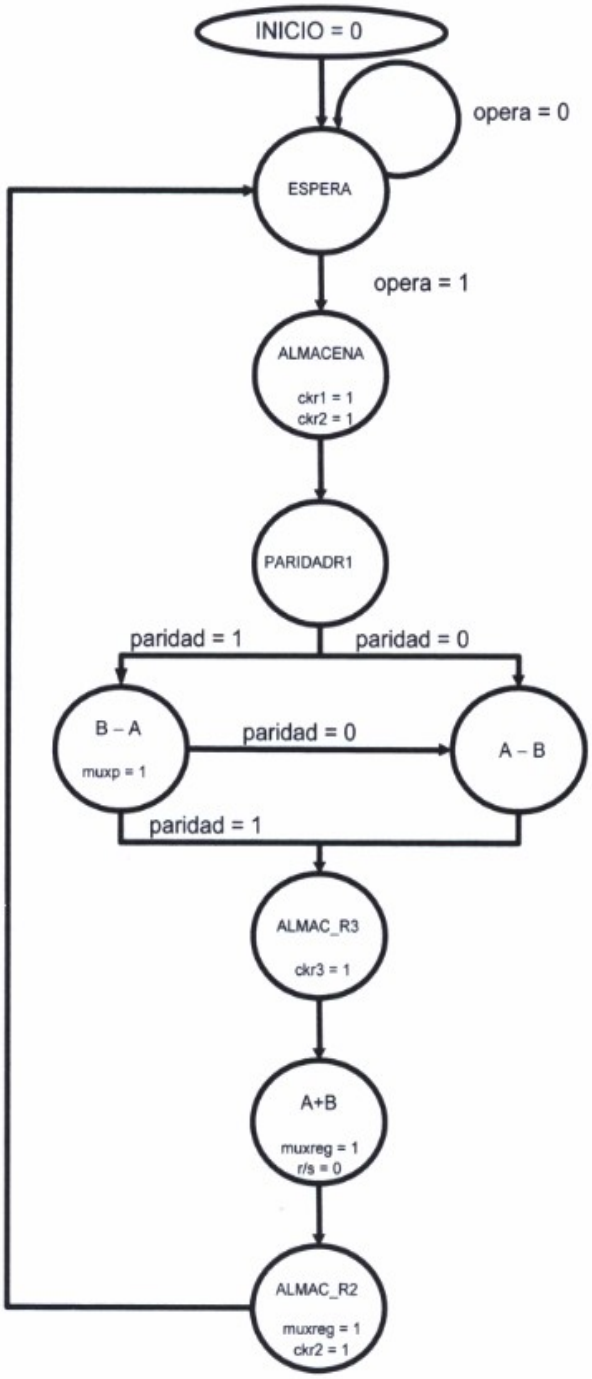


Finalmente, el sumador/restador está basado en el sumador 74HC283 y las puertas XOR necesarias para hacer el complementador.



se cambia el control de selección de los multiplexores del generador de paridad y del sumador/restador. De esta forma tendríamos el contenido de R2 a la entrada del generador de paridad.

De nuevo se comprueba la paridad que en este caso es la del dato almacenado en R2, es decir, B. Tanto si ésta es impar como si lo fue en el estado anterior la de A, se pasa al estado A-B en el que la selección de los multiplexores está como al principio y el sumador/restador está realizando la operación A-B. Tras este estado o si en el estado B-A la paridad fue par, se pasa al estado ALMAC_R3 en el que se almacena el resultado del sumador restador, sea A-B o B-A, en el registro R3.

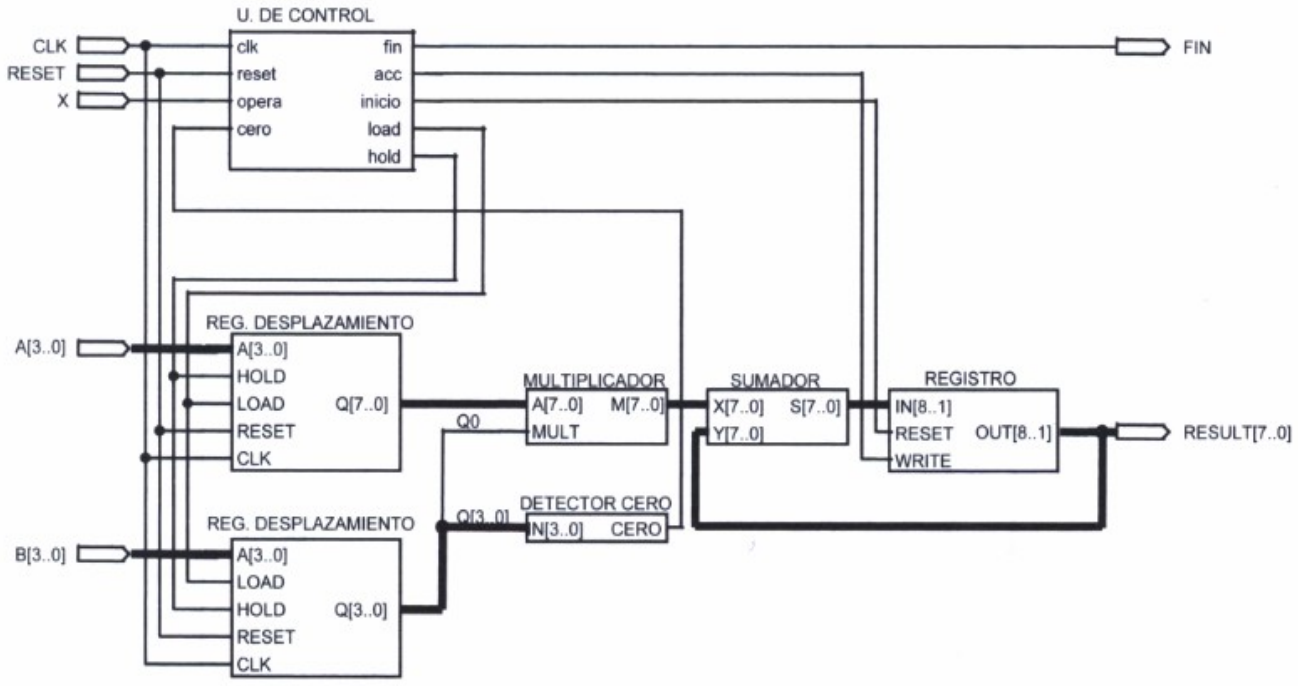


De este estado se pasa al estado A+B en el que se pone la señal de control del sumador restador en suma (r/s = 0), ya que hasta ahora había estado en resta. También se cambia el control de la selección del multiplexor R2 de forma que se pueda introducir en R2 la salida del sumador/restador.

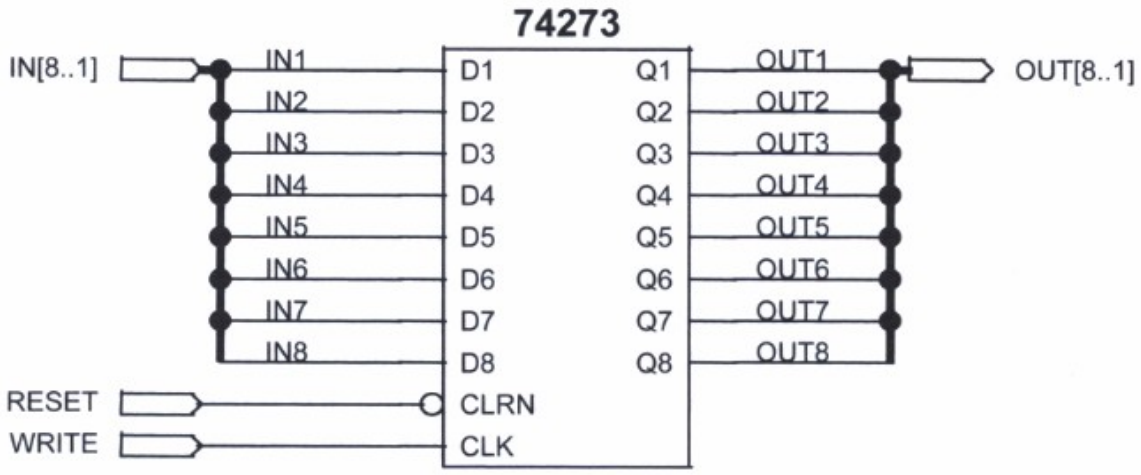
Finalmente se llega al estado ALMAC_R2 en el que, manteniendo el control del multiplexor, se almacena su salida, es decir A+B, en el registro R2, finalizando el ciclo completo y volviendo al estado ESPERA.

Problema 5.9 Para realizar la multiplicación mediante desplazamientos y sumas, necesitaremos dos registros de desplazamiento donde guardaremos el multiplicando y el multiplicador. Uno de ellos, por ejemplo A, lo desplazaremos a la izquierda, incluyendo ceros por la derecha, por lo que para no perder los datos será necesario que el registro tenga 8 bits, lo que nos permitirá realizar hasta cuatro desplazamientos sin perder ningún dato. El otro dato, es decir B, lo desplazaremos hacia la derecha y utilizaremos solamente el bit menos significativo disponible tras cada desplazamiento. Este bit será el que se multiplique con el dato A disponible en cada momento. Para ello bastará con unas puertas AND. Cuando se realice la primera multiplicación, el resultado se almacenará en un registro cuyo contenido deberá sumarse a los resultados de las multiplicaciones que se vayan obteniendo posteriormente.

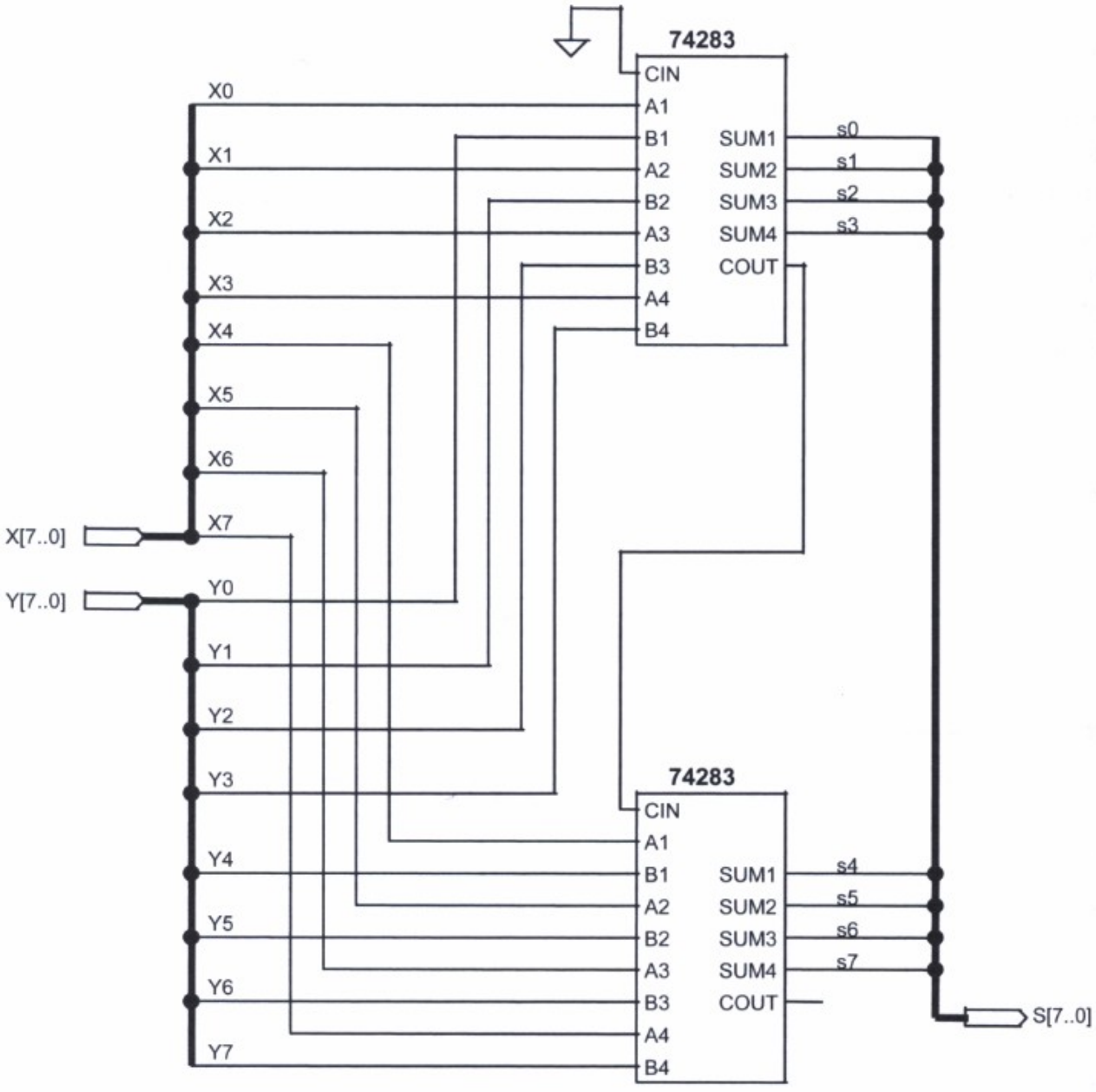
La unidad de control se encargará de controlar los registros de desplazamiento y el registro acumulador para que el proceso se realice correctamente. Para saber cuando ha terminado este proceso, bastaría con realizar 4 multiplicaciones parciales. Sin embargo, se puede utilizar una solución más general, en la que una puerta OR, de tantas entradas como bits tiene el dato B, detecta cuando este dato es 0, indicando a la unidad de control que el proceso ya puede finalizar. En la siguiente figura se puede ver el diagrama de bloques de esta posible solución.



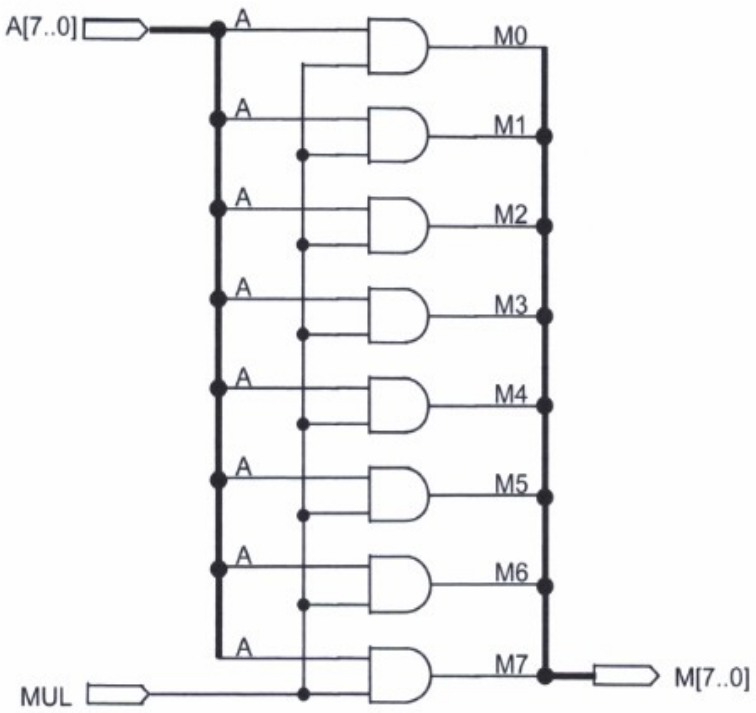
Vamos a ver ahora cómo están construidos cada uno de estos bloques. El registro acumulador estará formado por el circuito 74HC273 tal y como aparece en la siguiente figura:



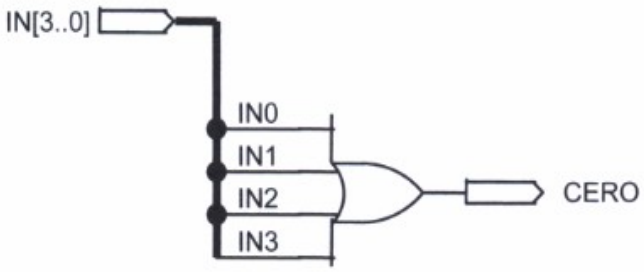
El sumador de 8 bits lo realizaremos con dos sumadores de 4 bits 74HC283 conectados tal y como se puede ver en la siguiente figura:



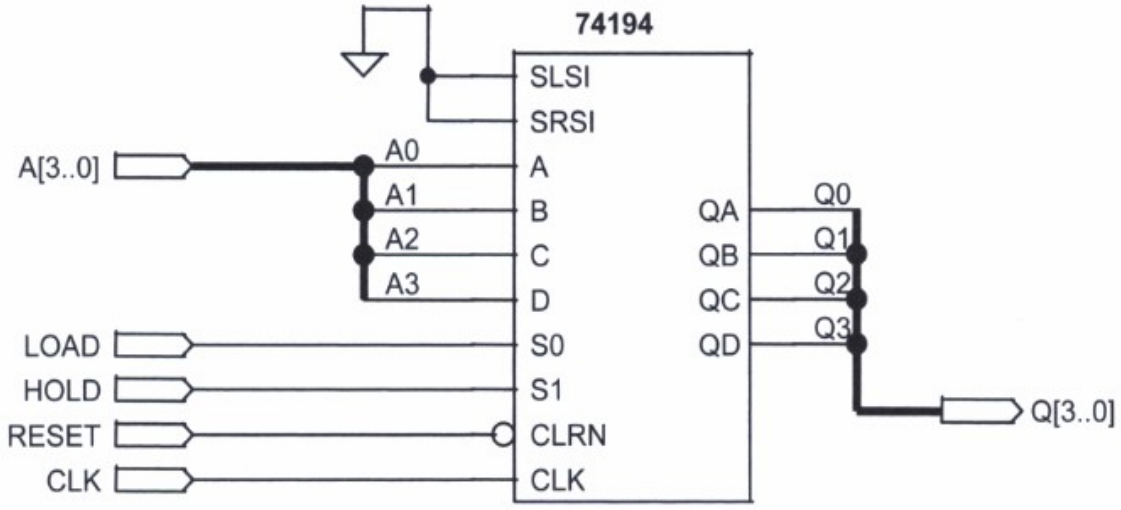
El multiplicador lo haremos con 8 puertas AND de dos entradas de la siguiente forma:



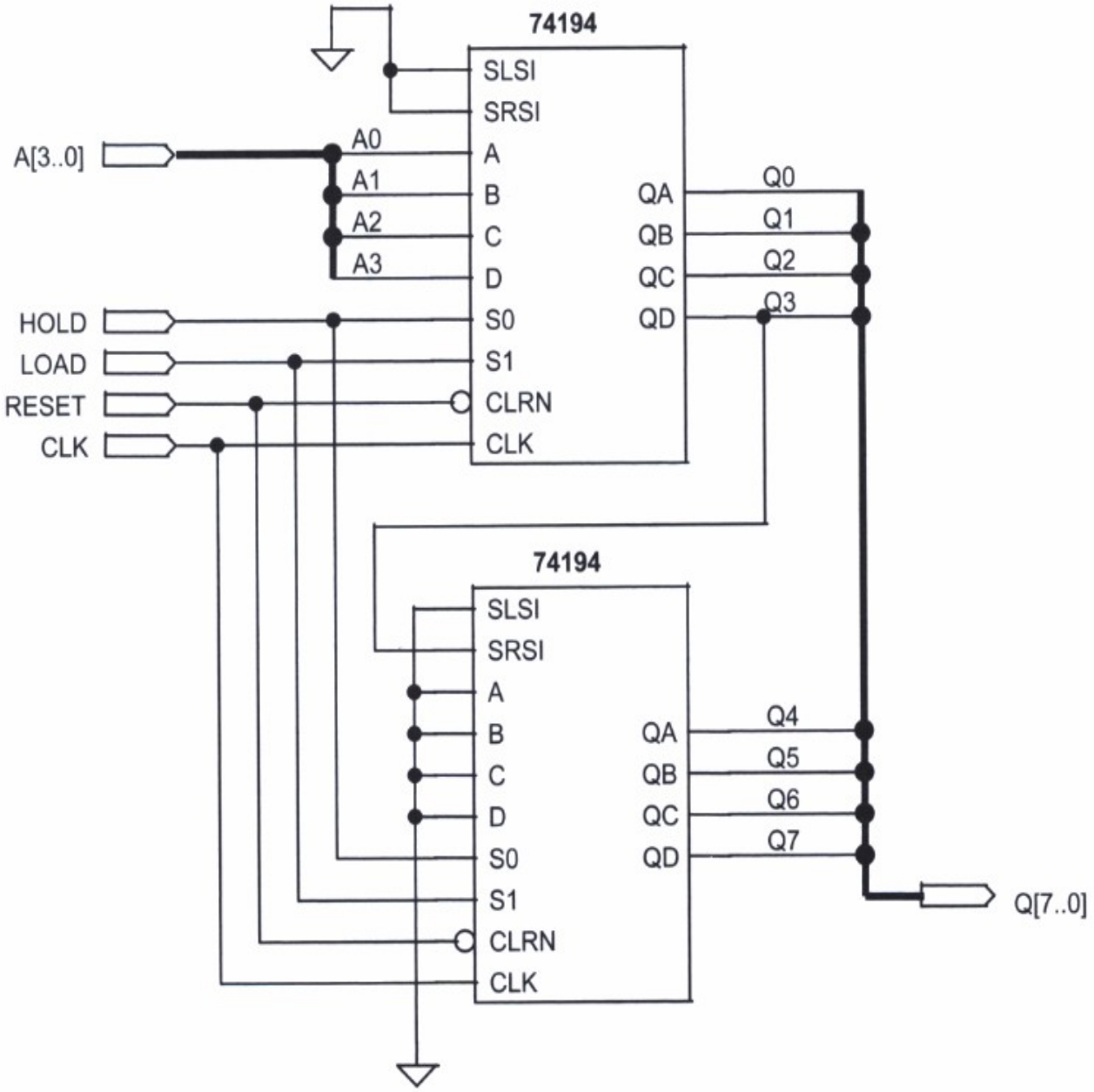
El detector de cero es una simple puerta OR de cuatro entradas.



El registro donde se guarda B está formado por un registro de desplazamiento de 4 bits 74HC194 conectado de la siguiente forma.



Finalmente, el registro donde se guarda A deberá estar formado por dos registros de desplazamiento 74HC194 conectados tal y como se puede ver en la siguiente figura:



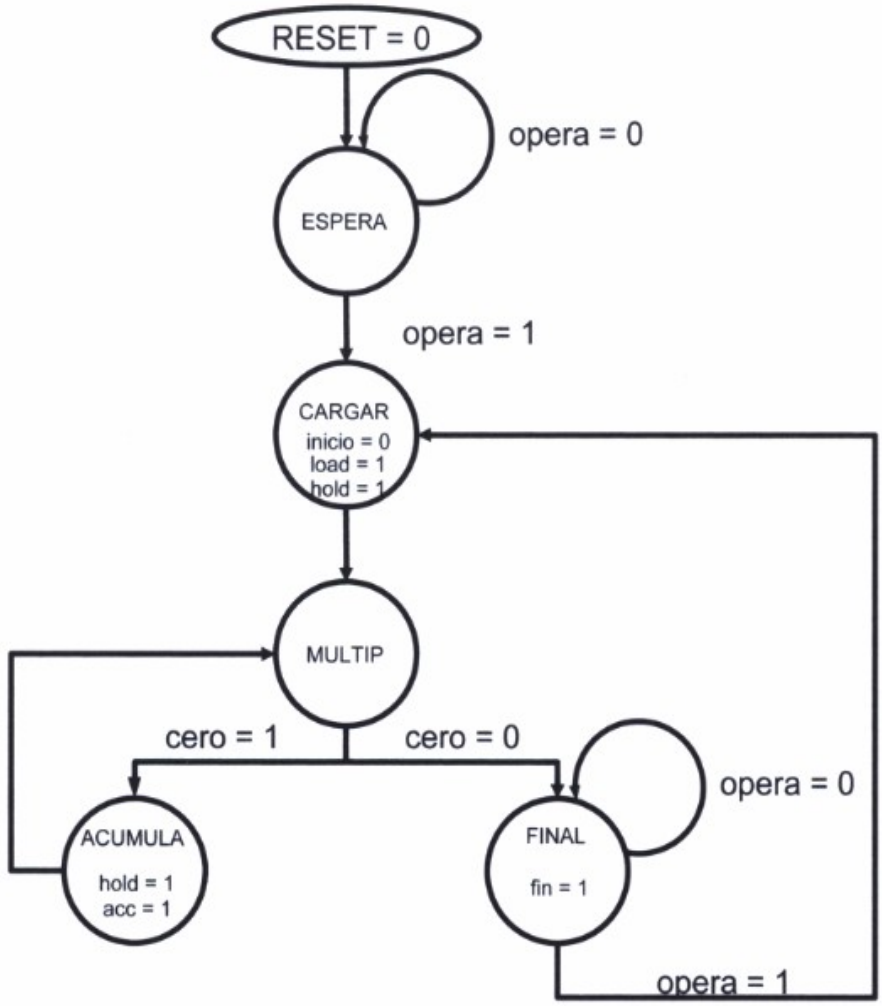
Nótese que las dos entradas de control del registro de desplazamiento de 4 bits y el de 8 bits están intercambiadas, de tal forma que cuando el de 8 bits se desplace hacia la izquierda con las mismas señales de control el de 4 bits lo hará hacia la derecha. La carga de ambos registros de desplazamiento se realiza con las dos señales de control a 1 mientras que el mantenimiento se hace con ambas a 0. Para desplazar se debe poner una a 0 y otra a 1 (ver tabla de verdad del circuito 74HC194).

Tal y como aparece en el diagrama de bloques general, la unidad de control tendrá las entradas y salidas que se indican a continuación:

Entradas: *OPERA, CERO*

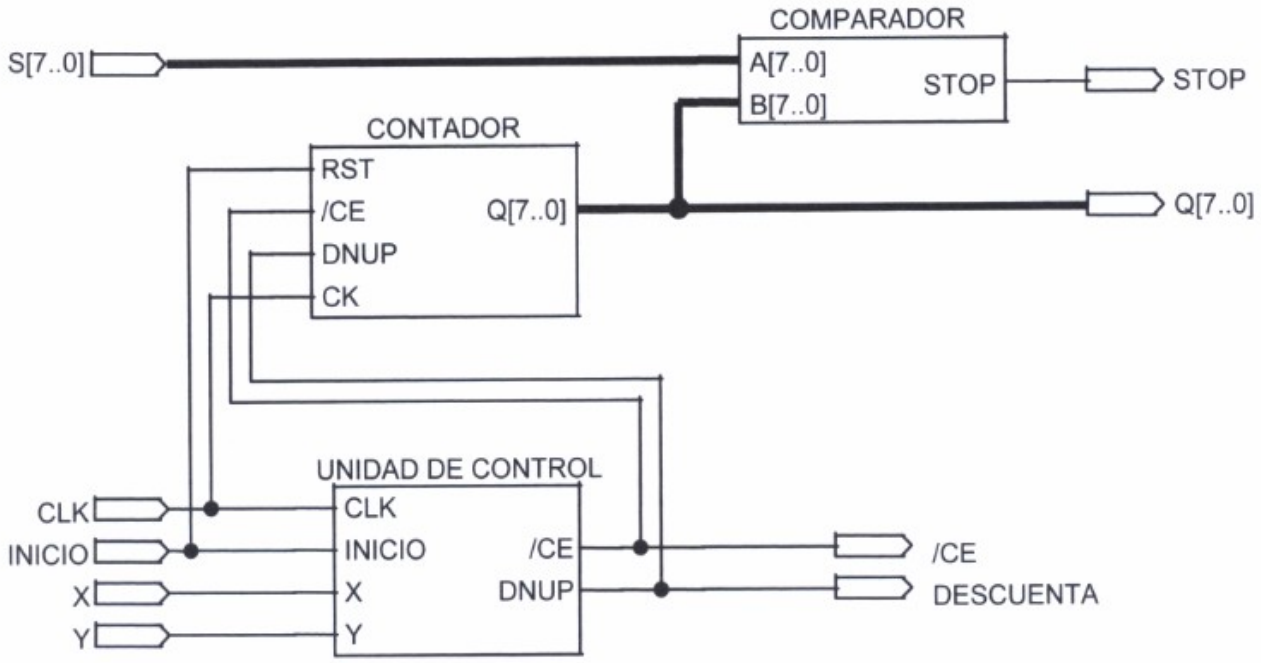
Salidas: *LOAD (0), HOLD (0), INICIO (1), ACC (0), FIN (0)*

Los valores entre paréntesis en las salidas indican los valores por defecto de forma que, en el diagrama de estados, solamente indicaremos las salidas que tengan un valor diferente. El diagrama podría ser el siguiente:

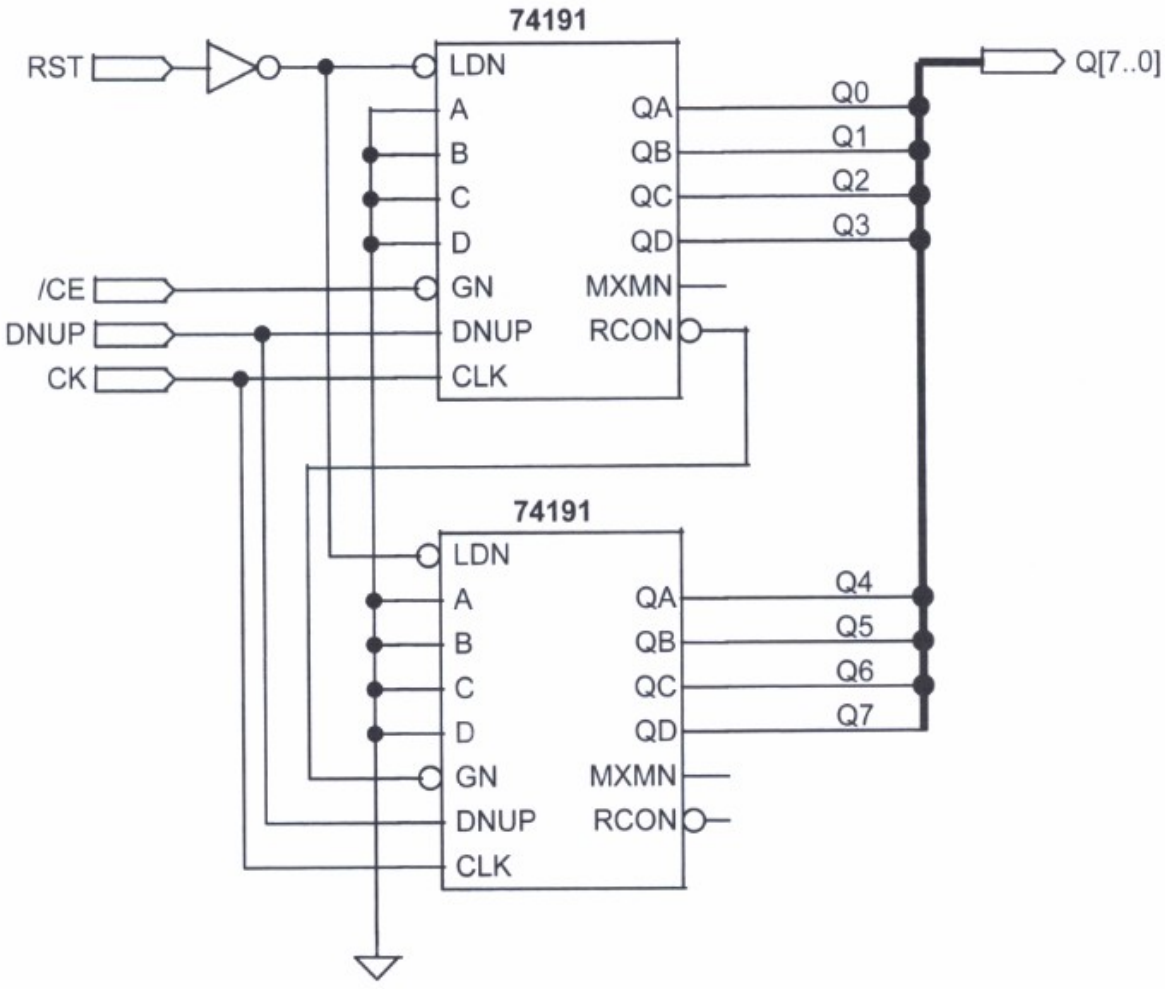


Con la activación de la entrada de *reset*, el circuito entrará en el estado ESPERA, en el que permanecerá hasta que se active la señal X, que en la máquina de estados se ha llamado *opera*. Cuando se activa esta señal, el circuito pasa al estado CARGAR, en el que se activan las señales correspondientes de los registros de desplazamiento para que se produzca la carga en paralelo de los datos. También se inicializa el registro acumulador. Tras este estado, se pasa al estado MULTIP, en el que se pregunta por la entrada *cero*. Si vale 1, es decir, si el operando de 4 bits no es 0, se guarda el resultado en el acumulador y se desplazan los registros, volviendo al estado anterior donde de nuevo se preguntará por la entrada *cero*. Si vale 0, es decir, si el operando de 4 bits es 0, el circuito ha finalizado la operación, por lo que pasa al estado FINAL en el que se genera la señal *fin*. En este estado permanecerá hasta que de nuevo se active la señal X, en cuyo caso pasará al estado CARGAR iniciándose un nuevo proceso.

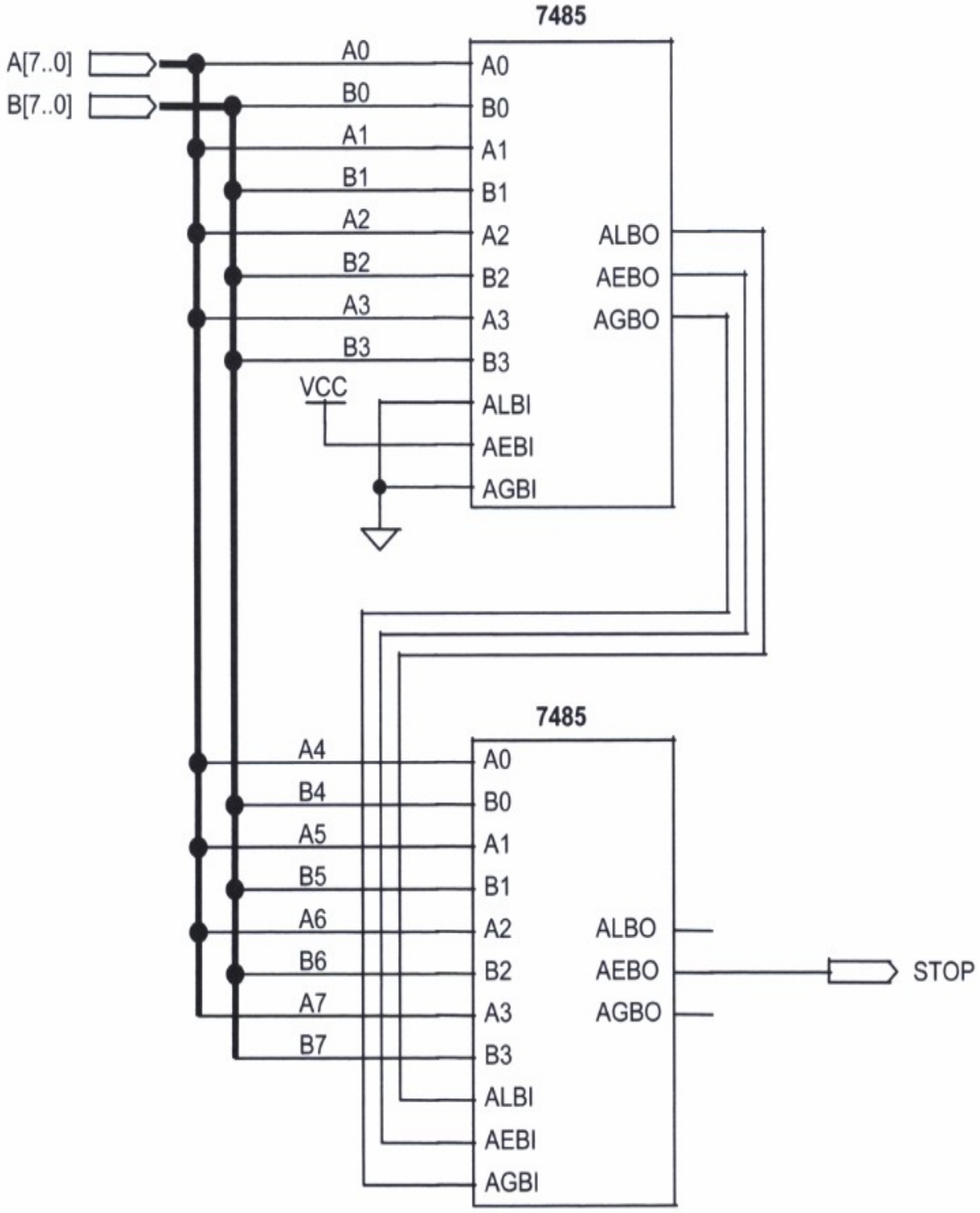
Problema 5.10 Para resolver el problema planteado nos hará falta que el circuito disponga de un contador de 8 bits, donde estará almacenado el número de personas presentes en el local, y un comparador de 8 bits que compare dicho valor con el máximo establecido mediante los microinterruputores. Ese es precisamente el material disponible para la unidad de proceso por lo que el diagrama de bloques del circuito solicitado puede ser el siguiente:



Para diseñar el contador de 8 bits basta con utilizar los dos contadores de 4 bits conectados de manera que formen uno de 8 bits que pueda contar hacia arriba cuando alguien entra y hacia abajo cuando alguien sale. El circuito es el siguiente.



De forma similar el comparador de 8 bits estará formado por dos comparadores de 4 bits conectados de la siguiente forma.

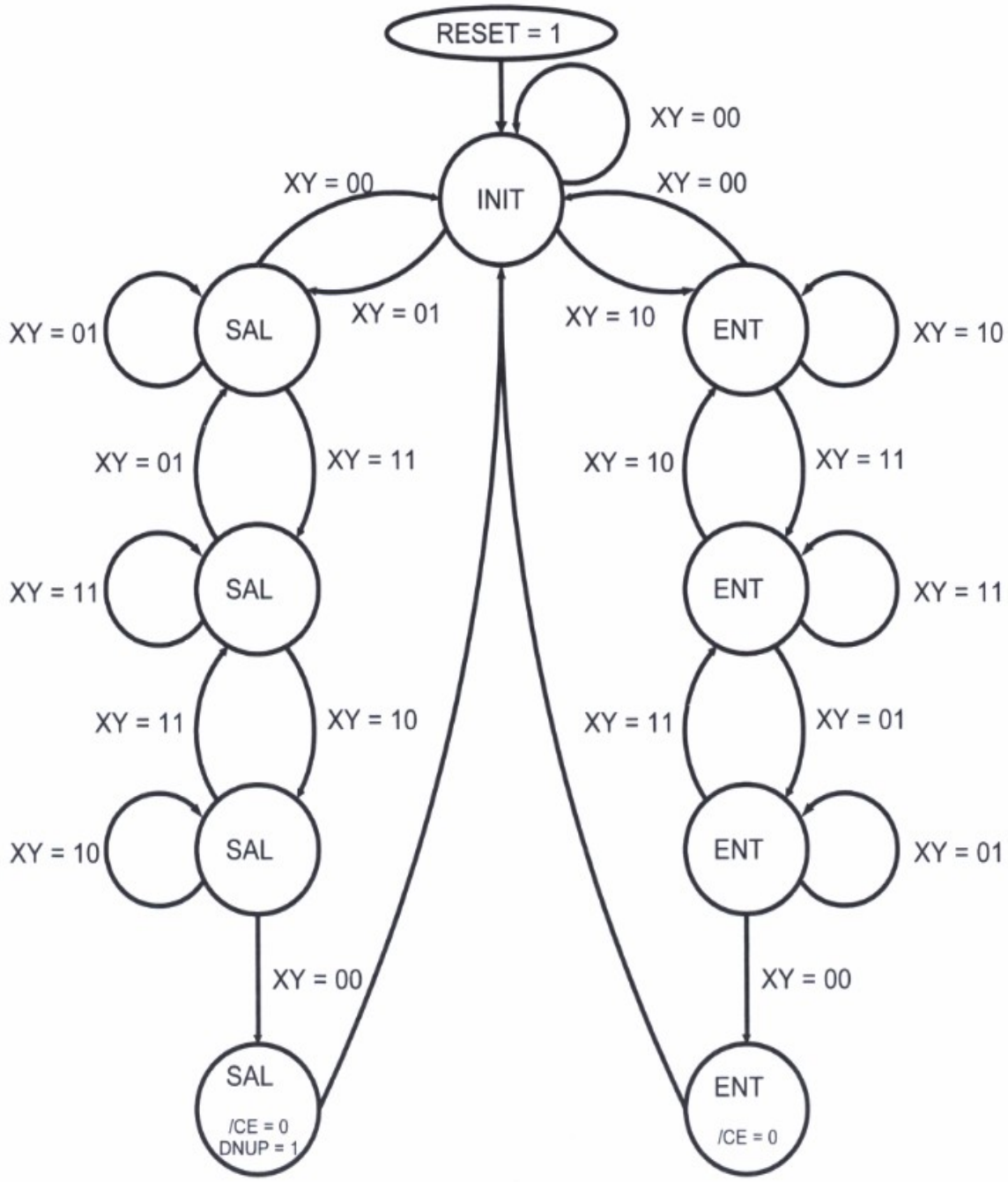


En cuanto al diagrama de estados, como siempre, indicamos en primer lugar sus entradas y salidas, así como los valores por defecto asignados a éstas últimas:

Entradas: *X, Y*

Salidas: */CE (1), DNUP(0)*

El diagrama puede ser el siguiente:

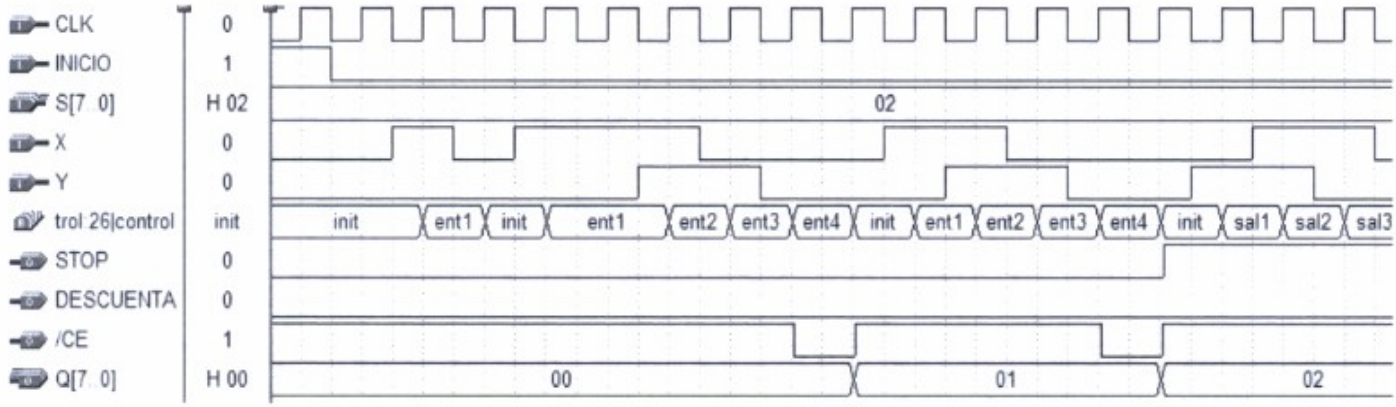


En el estado *init*, el circuito espera a que se active alguna de las dos células. Aunque el enunciado no lo especifica, vamos a considerar que la célula X es la que está en el exterior y la Y en el interior. Si se activa la célula X en primer lugar, eso indica que alguien está entrando por lo que pasamos al estado *ent1*, en el que el sistema se quedará mientras X siga activa y no se active Y. Si se desactivara X, el estado volvería a ser el anterior, indicando que la persona que iba a entrar no ha completado tal acción. Si se activa la célula Y, se pasa al estado siguiente *ent2* indicando que la persona sigue en su acción de entrar al local. Lo mismo sucede en los estados *ent2* y *ent3* hasta llegar al estado *ent4*, momento en el cual se considera que la persona está dentro, activándose el contador que se incrementará con el siguiente flanco de reloj y volviendo al estado *init*.

En el caso de que inicialmente se active en primer lugar la célula Y, la secuencia de estados es similar pero indicando que una persona va a salir. En el estado final, el contador se decrementará en lugar de incrementarse, volviendo también al estado *init*.

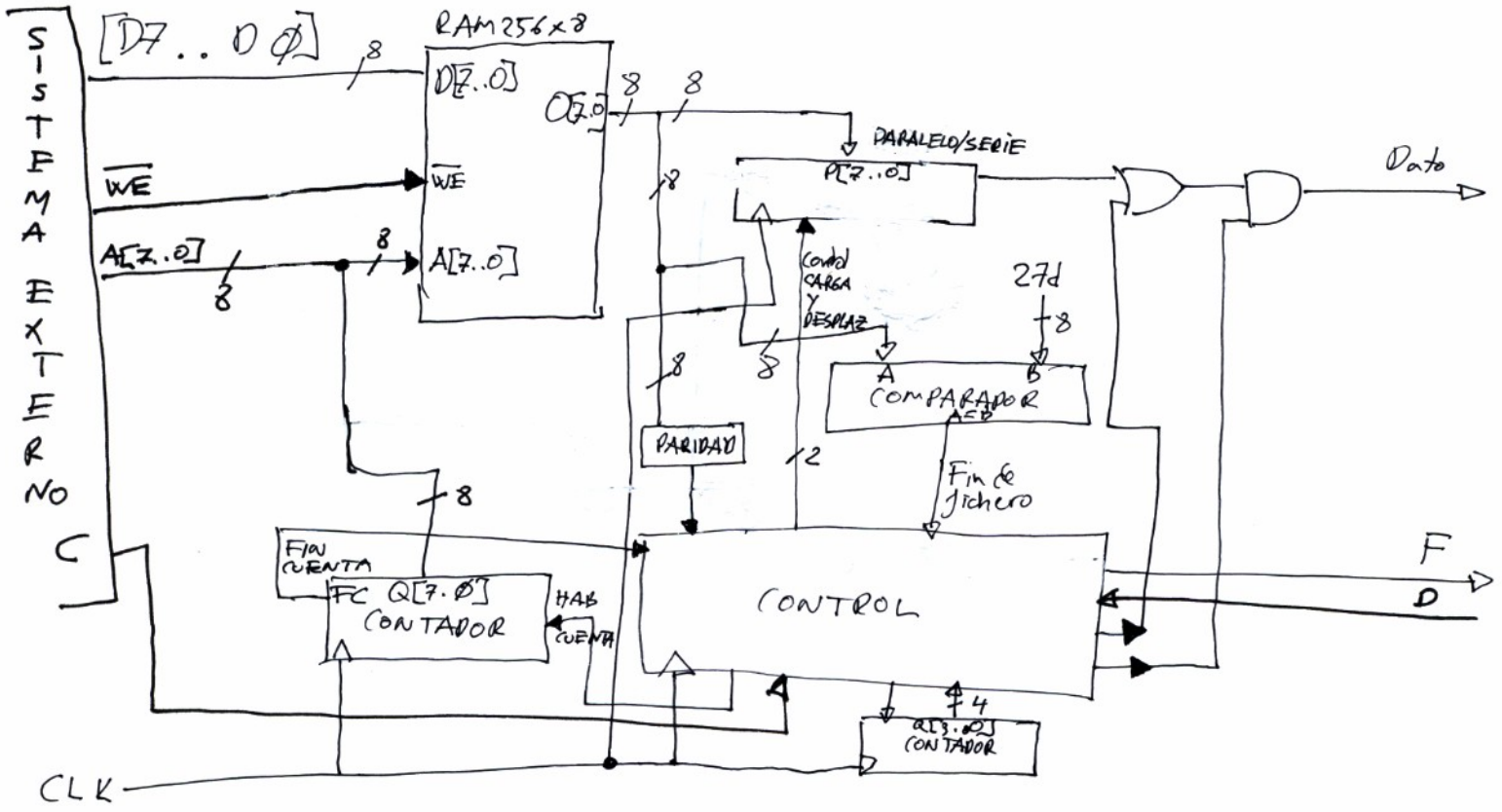
La señal STOP se activará en el momento el que el contador tenga el mismo valor que la entrada de los microinterruptores ya que está directamente conectada a la salida del comparador que compara en todo instante el valor del contador con el valor de los microinterruptores.

En la siguiente figura se puede ver el resultado de la simulación de esta solución para la secuencia de entrada propuesta en el enunciado:



5.13

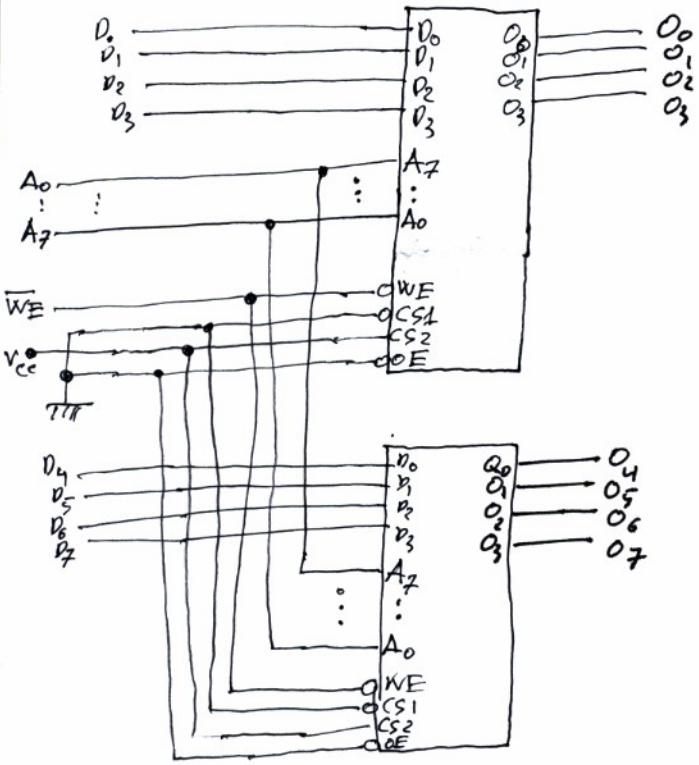
El diagrama de bloques del sistema será el siguiente:



El contador irá haciendo un barrido de la memoria tanto durante la primera fase, en la que el sistema externo envía los datos a la memoria en paralelo por D[7..0] como cuando el sistema envía los datos en serie a través de la línea "dato". En el primer caso se escribirá un dato cada pulso de reloj y en el segundo se leerá cada 12 pulsos (ya que luego hay que enviar el dato en serie). El registro paralelo/serie cargará el dato y lo irá enviando en serie por la línea DATO. Las puertas AND y OR en su salida permiten enviar por la línea DATO un

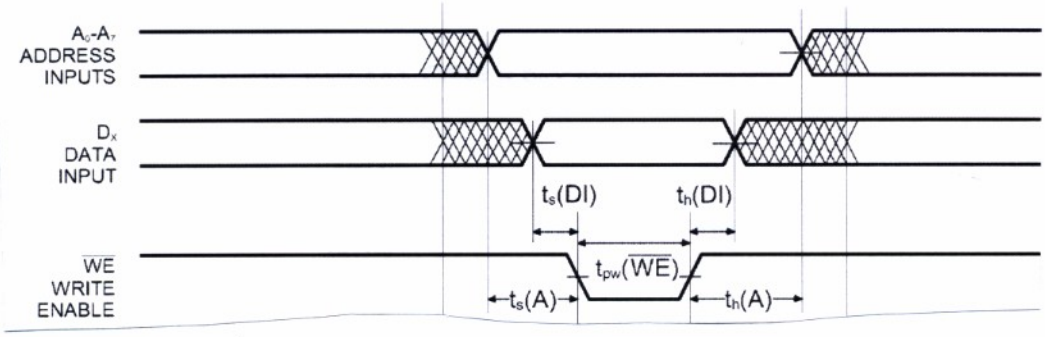
cero o un uno, para el envío de los bits de inicio, fin y paridad. El comparador compara el dato con 27d para ver cuando ha llegado el Fin de fichero.
El contador de 4 bits sirve para contar el número de bits que se van enviando en serie y saber cuando hay que enviar el bit de paridad y el de fin.

- El bloque RAM, hecho con integrados P930422 quedará así:



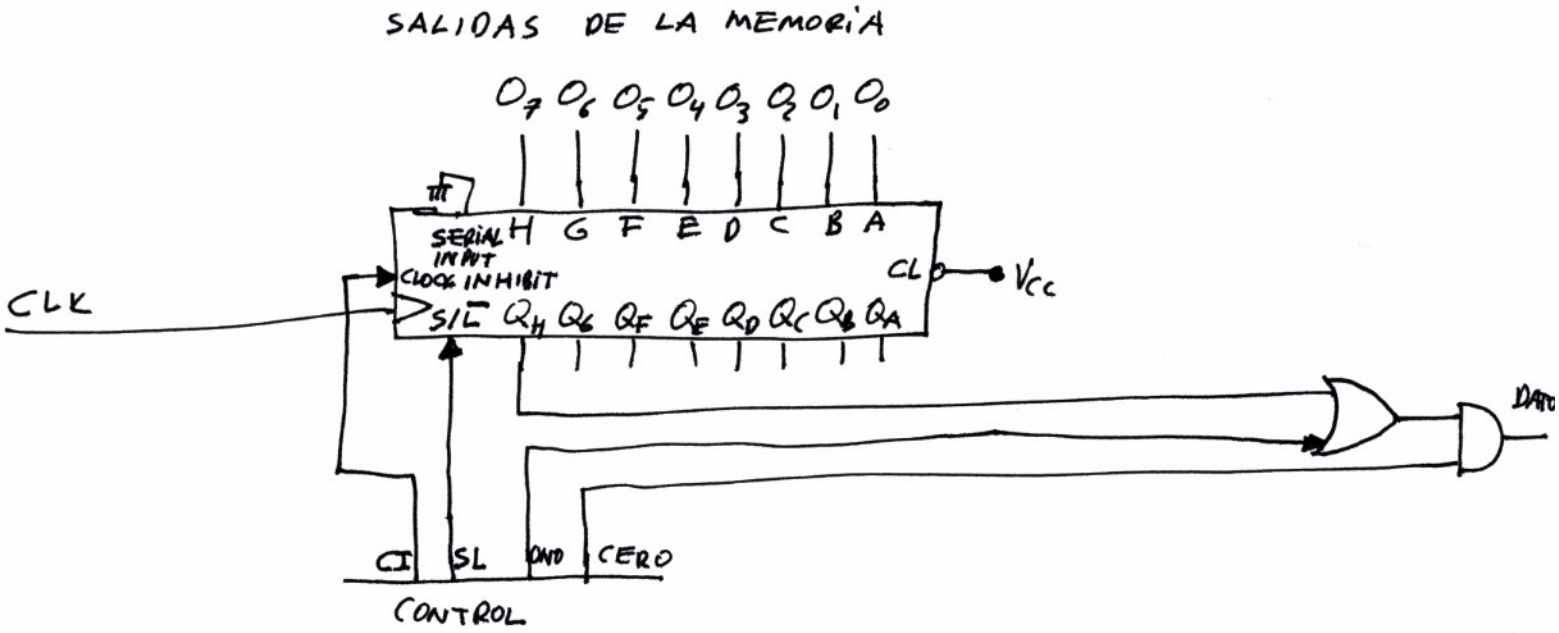
Dejando siempre activos CS y OE de forma que la escritura en la memoria se controle con $\overline{WE}$. La memoria estará siempre leyéndose ($O_0..O_7$ siempre tendrán un dato, es decir, nunca estarán en alta impedancia).

Al tener $\overline{OE}$ y CS activados, la escritura sigue el cronograma de la figura 1 (he dejado sólo lo que nos interesa ahora)

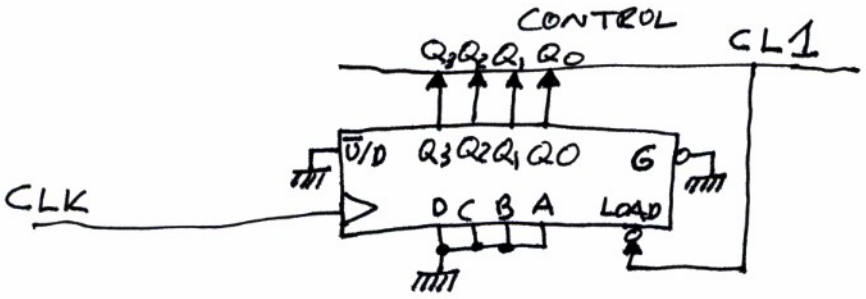


Donde vemos que tanto el tiempo de Setup (para datos y para direcciones) y el tiempo de HOLD (también para datos y direcciones) son ambos de 5 ns. Será importante tener esto en cuenta cuando generemos la señal $\overline{WE}$ desde el control.

- El registro de desplazamiento irá conectado así:



- El contador de 4 bits será un 74191 conectado así:

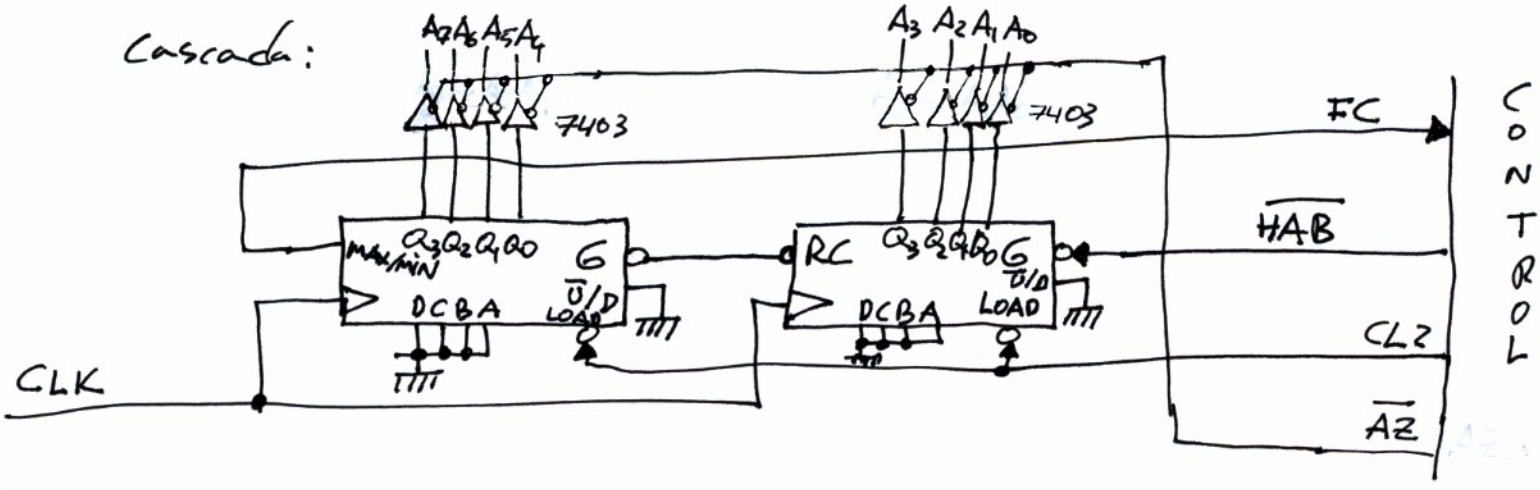


$\bar{U}/D$ está a ϕ para que cuente ascendentemente

$\overline{G} = 0$ para permitir la cuenta.

La señal $\overline{LOAD}$ se usa para inicializar el contador a cero (es asíncrona, lo que habrá que tener en cuenta para el control)).

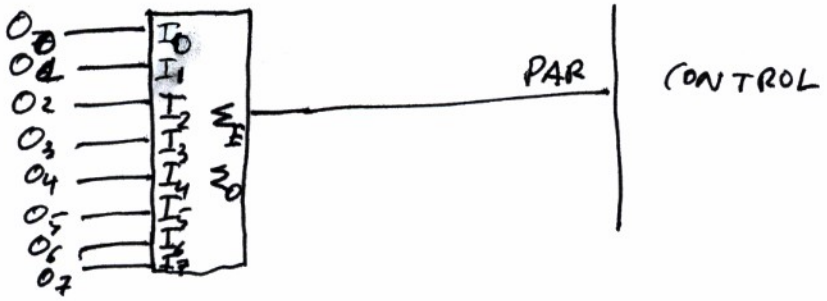
- El contador de 8 bits se hará con dos 74191 en cascada:



Las señales $\bar{U}/D$ se conectan a ϕ para que la cuenta sea ascendente. Las señales $\overline{LOAD}$ se usan para inicializar el contador (a 0000). La señal $\overline{HAB}$ habilitará la cuenta. Cuando el contador llegue a fin de cuenta (11111111) dará un pulso a nivel alto en MAX/MIN (señal FC que va al control).

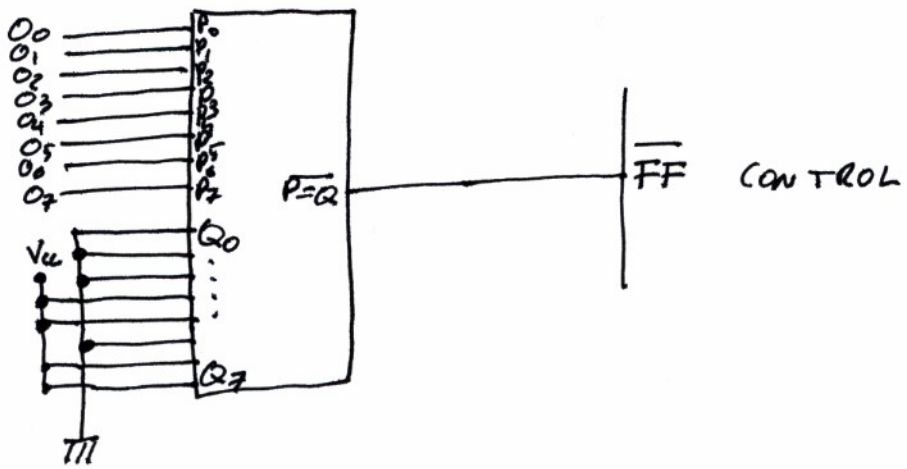
Los buffers 7403 sirven para poner las salidas $A_7 .. A_0$ en alta impedancia

- El control de paridad se hará con un 74280 así:



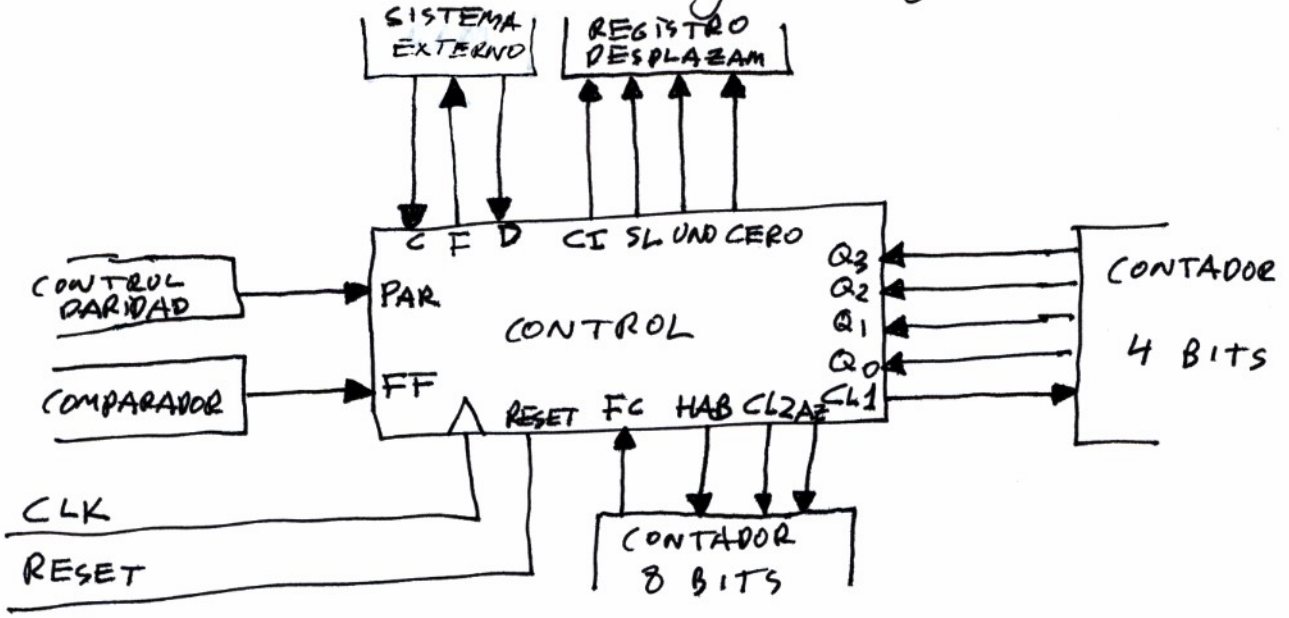
como no se especifica si la paridad es par o impar, se ha conectado para paridad PAR.

- El comparador se hará con un 74688 así:



- Diseño del sistema de control

— Con todo lo anterior, el bloque de control queda configurado de la siguiente forma:



— El funcionamiento es el siguiente:

1) El sistema espera a que el sistema externo escriba la memoria. Durante este tiempo las señales de dirección A[7..0] las pondrá el sistema externo, lo mismo que la señal $\overline{WE}$, Por lo que las salidas del contador de 8 bits permanecerán en alta impedancia (señal AZ = 1). Seguirá así hasta que el sistema externo indique que ha terminado de escribir la memoria (C = 1).
2) Se envía un 1 por la línea de DATO (UNO=1, CERO=1)
3) Se direcciona la memoria (AZ = 0), se lee el dato y se carga en el registro de desplazamiento. (CI=1, SL=0). Se borra el contador de 4 bits (CL1=0)
4) Se envía en serie durante los 8 siguientes pulsos de reloj. (CI=1, SL=1, UNO=0, CERO=1). Sabemos que se ha enviado el último bit cuando $Q_3Q_2Q_1Q_0 = 1000$

5) Se envía el bit de paridad.

6) Se envía un 1 por DATO.

7) Se envía un Ø por DATO (línea en reposo).

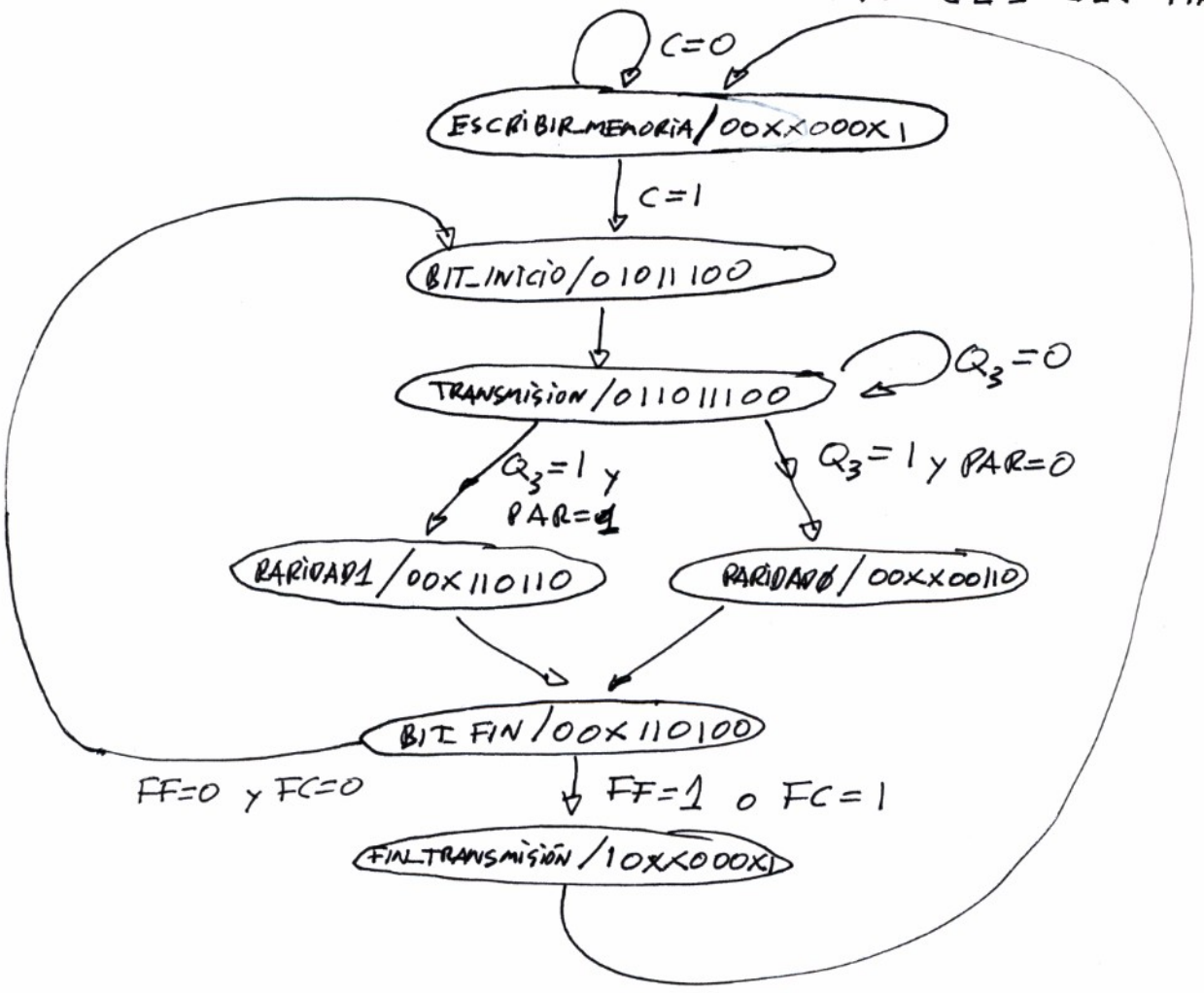
8) Se hace avanzar el contador de 8 bits ($\overline{HAB} = 0$) para dirección la siguiente posición de memoria

9) Se repiten los pasos 2 a 8 hasta que FC = 1 (se han enviado todos los datos de la memoria) o FF = 1 (se ha llegado a un fin de fichero). En este caso se activa F = 1.

- El diagrama de estados del control será el siguiente:

ENTRADAS: C D PAR FF FC $Q_3 Q_2 Q_1 Q_0$

SALIDAS: F CI SL UNO CERO CL1 CL2 HAB AZ



En el estado ESCRIBIR_MEMORIA se envía un Ø por la línea de datos (reposo) y se inicializan los contador, manteniendo la salida del contador de direcciones en alta impedancia.

En el estado BIT_INICIO el contador de 8 bits está a cero y no cuenta. El de 4 bits está a Ø y se habilita para que avance en el próximo flanco del reloj (al pasar al estadosiguiente).

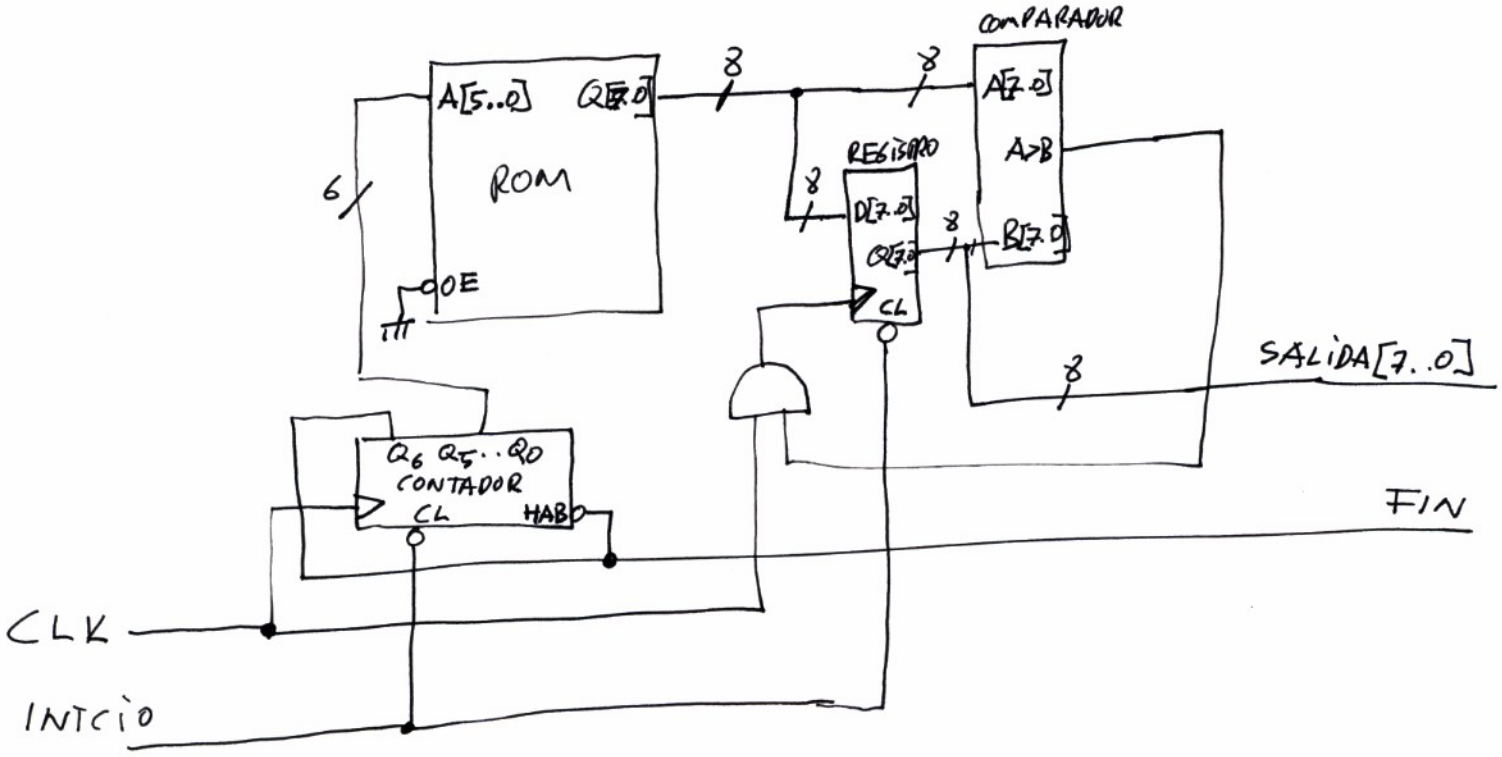
En el estado TRANSMISION se hace que el registro de desplazamiento desplace para ir enviando los distintos bits por la línea DATO. También se permite que el contador de 4 bits cuente el número de bits transmitidos, hasta llegar a 8.

En los estados PARIDAD1 y PARIDADØ se envía el bit de paridad, forzándolo con las señales UNO y CERO.

En el estado BIT_FIN se envía un uno y se habilita la cuenta del contador de 8 bits para que en el próximo flanco direccione la siguiente posición de memoria.

En el estado FIN_TRANSMISIÓN se envía un 1 por F y se reseteam los contadores para volver a reposo.

5.34 El diagrama de bloques del circuito podría ser el siguiente.



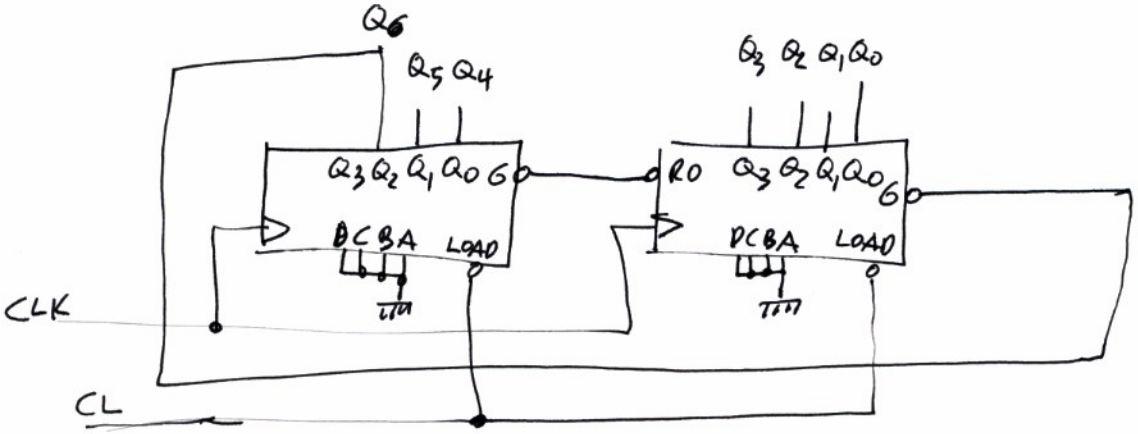
En el diagrama de bloques se incluye ya el CONTROL porque es muy sencillo y no es necesario un autómata para realizarlo. El funcionamiento del circuito es el siguiente:

Mientras la señal INICIO está a nivel bajo, el registro y el contador permanecen borrados (sus salidas a $\emptyset$). Al activar INICIO (a 1) el contador comienza a contar con cada flanco de reloj, haciendo un barrido de la memoria. Cada posición de memoria se compara con el valor almacenado en el REGISTRO y si

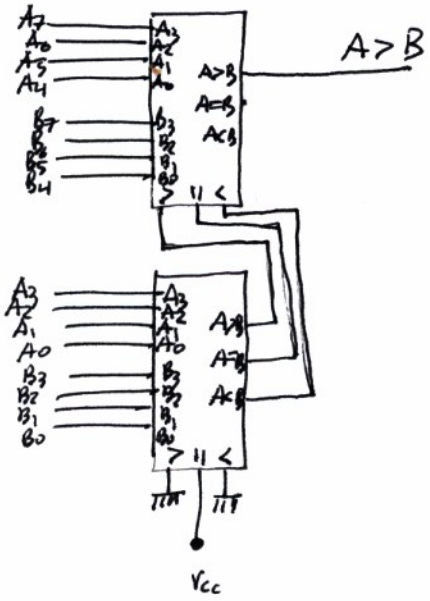
es mayor (A>B) activa, mediante la puerta AND, la entrada de reloj del registro de forma que el flanco de reloj siguiente capture el dato en el registro.

El contador cuenta hasta 63. Al pasar a 64 "daría la vuelta" a la memoria, pero en ese caso Q_6 se pone a 1, activando FIN y haciendo que el propio contador se quede parado al desactivar su habilitación de cuenta.

- El contador se hará con dos 74191 así:



- El comparador se hará con dos 7485 así:



- El registro será el 74273 donde la señal que viene como $\overline{CL}$ en el diagrama de bloques será MR.