

Índice general

1. Introducción	1
1.1. Características	1
1.2. Requerimientos	2
2. Placa de programación	3
2.1. Introducción	3
2.2. Circuito de alimentación	3
2.3. Circuito de programación	5
2.4. Circuito de comunicación serie RS232	7
2.5. Circuito de prueba	9
2.6. Circuito generador de frecuencias	9
2.7. Designación de los terminales del CPLD en la placa:	10
3. Placa expansión	13
3.1. Introducción	13
3.2. Alimentación de la placa y conector DB37	13
3.3. Circuito de Pulsadores	14
3.4. Circuito con llaves o <i>switches</i>	15
3.5. Circuito de tiempo (CI 4047)	15
3.6. Señales de reloj externas	16
3.7. Circuito con diodos <i>leds</i>	16
3.8. Circuito de salidas optoacopladas	17
3.9. Circuito con <i>displays</i> BCD de siete segmentos	17
A. Información de programación	19
A.1. La Entidad (<i>Entity</i>)	19
A.2. El archivo UCF	20
B. Mediciones	21
B.1. Optoacopladores	21

Capítulo 1

Introducción

En el presente informe se realizará la descripción del dispositivo de entrenamiento con CPLD formado por dos placas (que llamaremos placa programadora y placa de expansión) que fueron realizadas por integrantes del grupo de investigación CUDAR¹ cuyo interés estuvo centrado en brindar la mayor cantidad de herramientas para el uso del CPLD, se verá que el mismo cumple con las expectativas del programador o diseñador que se esté iniciando en el lenguaje de descripción de hardware. Éste equipo CPLD llega a la cátedra de Técnicas Digitales I y la materia electiva Técnicas Digitales IV, en su versión 3.0, la cual queda a disposición de los alumnos para que hagan uso de sus recursos.

A continuación se tratará de comenzar una explicación de todos los bloques de circuito de cada placa, también se dejarán a disposición algunas mediciones realizadas en el apéndice B, para que el usuario tenga parámetros a los que atenerse al momento de realizar un proyecto.

1.1. Características

- CPLD XC9572XL PLCC44.
 - PLCC44 - 44 pines.
 - 34 pines de I/O disponibles para el usuario.
 - 5ns de retardo entre pines.
 - Frecuencia hasta 178MHz.
 - 72 macroceldas.
 - 1600 compuertas lógicas.
 - Baja potencia de operación.
 - Pines I/O tolerantes a señales de 5V, 3.3V y 2.5V.
 - Capacidad de salida de 3.3V o 2.5V.
 - ISP (*In System Programmable*).
 - Más de 90 términos producto por macrocelda.
 - *Full IEEE Standard 1149.1 B.S.* (JTAG).
- Conector DB37 mapeado los puertos I/O.
- Cuatro *displays* siete segmentos ánodo común.
- Cuatro salidas optoacopladas.
- Dos pulsadores para señales de entrada.
- *Switches* de ocho *bits* para señales de entrada.
- Ocho *leds* para visualización -para observar registro de ocho *bits*-.
- Tres relojes de diferentes frecuencias gobernados por *jumpers*.
- Un reloj con frecuencia variable.

¹Centro Universitario de Desarrollo en Automatización y Robótica.

- Dos entradas para reloj externo.
- *Hardware* interfaz para comunicación serie RS232.
- Operación a 5V.
- Voltaje de alimentación 9 - 12V².
- Cable de programación JTAG para puerto paralelo.
- *Software* de escritura y verificación.
- Analizador lógico en *software* a través del puerto JTAG.

1.2. Requerimientos

- Sistema operativo *Windows* 98/XP/2000.
- Cable con conector DB25 para PC -programador-.
- Driver para puerto paralelo.
- Adaptador de alimentación 9 -12V.

²Este voltaje está limitado por el CI 7805C véase en hoja de datos.

Capítulo 2

Placa de programación

2.1. Introducción

Ésta placa de programación soporta los CPLD de la familia XC9572XL y XC9570XL, tiene además una comunicación con el CPU mediante el puerto paralelo utilizando el protocolo JTAG el cual será descrito mas adelante.

En la figura 2.1 se tiene un diagrama en bloques de los distintos circuitos de la placa.

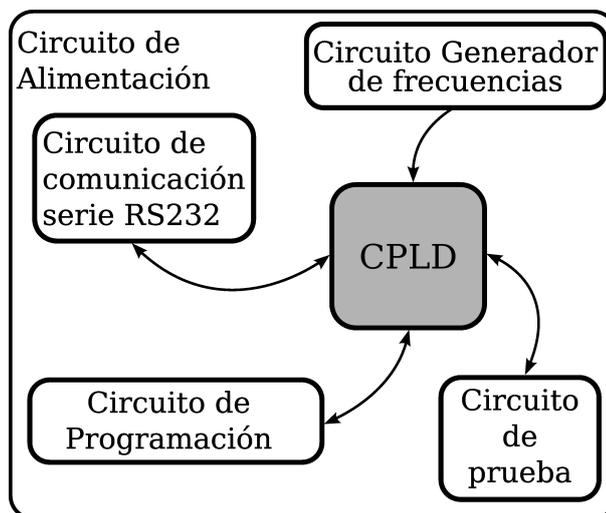


Figura 2.1: Diagrama que representa los bloques de la placa programadora.

2.2. Circuito de alimentación

Éste dispositivo de entrenamiento debe ser conectado para su correcto funcionamiento a una fuente de tensión continua que no supere los 12V, el regulador de tensión utilizado para alimentar ambas placas es el LM7805C, el mismo puede entregar una corriente máxima de 1A siempre y cuando se dispongan de condiciones óptimas de enfriamiento.

Se debe tener en cuenta que la tensión de entrada debe superar los 7.5V para una correcta estabilización de la tensión de salida, de todas formas es recomendable utilizar 9V y como dijimos, no es aconsejable que supere los 12V, sin embargo puede trabajar sin problemas a esa tensión en parte, gracias al disipador elegido.

El consumo de corriente máximo del equipo es de aproximadamente 600mA (varía muy poco de uno a otro), de todas formas el regulador de tensión incluye una protección interna de área segura para la salida de alimentación, a fin de limitar la disipación interna de potencia. Si la disipación de potencia interna se vuelve muy alta para el disipador dispuesto, se activa el circuito de apagado térmico previniendo el sobrecalentamiento del dispositivo.



Figura 2.2: Placa programadora de CPLD.

El regulador LM 7805C es el que se encarga de suministrar corriente a los circuitos de ambas placas, por lo tanto es de suponer que la potencia disipada por el mismo será de consideración. En nuestro caso el disipador está colocado en forma vertical como se puede observar en la figura 2.2. Se optó por ésta disposición para ahorrar espacio físico, pues se sabe que un disipador posee mayor rendimiento si está en posición horizontal.

Los CPLD que se utilizarán son de tipo "XL" que significa alimentación de 3,3V. Es importante tener esto en cuenta a la hora de adquirir un CPLD; no es lo mismo un XC9572XL que un XC9572 que trabaja a 5V. En la figura 2.3 se puede observar el circuito de alimentación.

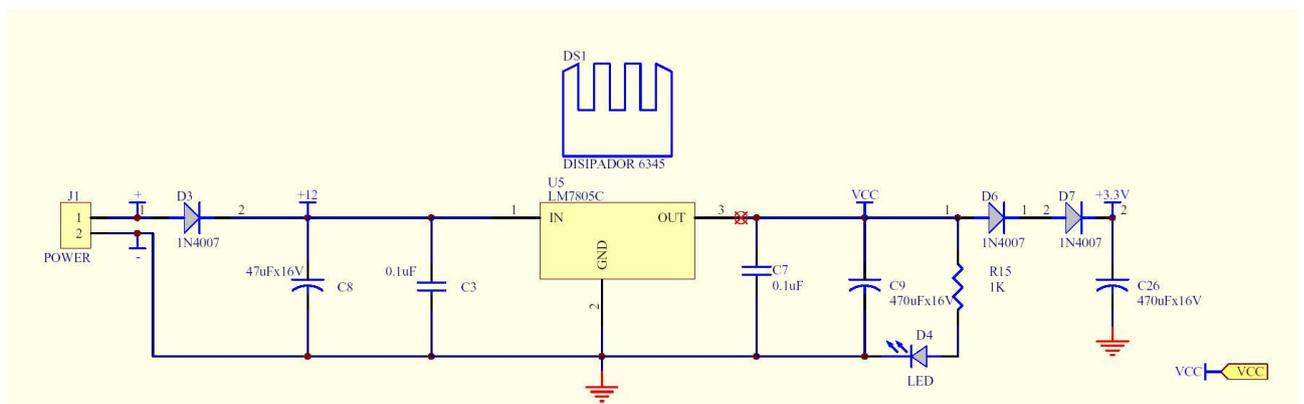


Figura 2.3: Etapa de alimentación.

2.3. Circuito de programación

Se utiliza el conector DB25 para conectar la placa a la PC a través del puerto paralelo.

Para la programación del CPLD se dispone del protocolo JTAG y del circuito mostrado en la figura 3.2. Sólo cuatro señales de entrada / salida son utilizadas para montar el puerto de programación, éstas señales están conectadas al circuito programador y éste al puerto paralelo.

Luego de conectar el cable paralelo al CPLD, el *software Impact* gestiona la configuración del dispositivo¹.

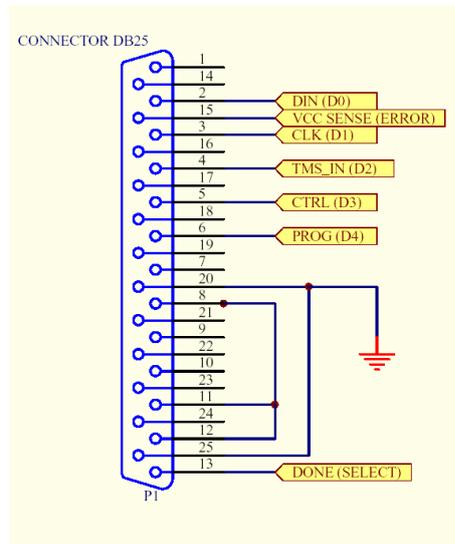


Figura 2.4: Disposición de los terminales del conector DB25.

Al sintetizar el código VHDL obtenemos un *bitstream* de configuración que es serializado por el *software Impact* a través del protocolo JTAG, con el cual se recorre el dispositivo CPLD secuencialmente según un registro de desplazamiento, realizando la configuración correspondiente en el mismo.

Recuerde previamente simular su descripción de *hardware* para asegurar que el CPLD realmente realice la tarea deseada y así minimizar las posibilidades de error².

Las señales que se toman desde el puerto paralelo y van hacia el circuito programador son:

- VCC SENSE (ERROR) [Pin 15].
- DONE (SELECT) [Pin 13].
- DIN (D0) [Pin 2].
- CLK(D1) [Pin 3].
- TMS_ IN (D2) [Pin 4].
- CTRL(D3) [Pin 5].
- PROG(D4) [Pin 6].

Éstas señales se conectan al circuito programador formado principalmente por una combinación de *buffers* en lógica negativa (74HC125). El 74HC125 es un cuádruple *buffer tri-state* su simbología se puede apreciar en la figura 2.5.

¹Nótese que **no** utilizamos el término **programar** que es más común cuando nos referimos a memorias. Usamos el término **configurar** que se utiliza tanto en el manejo de CPLDs como en FPGAs, éstas últimas poseen una memoria que tiene la descripción que queremos cargar. Cada vez que se aplica alimentación a la FPGA ésta se configura con la descripción ubicada en ésta memoria estable tipo *flash*, distinto es el caso de los CPLDs ya que éstos mantienen su configuración aún después de quitarles alimentación.

² Es muy común cometer errores de sincronismo debido a las transiciones de nivel de entrada a salida que no ocurren en forma instantánea obteniéndose valores incorrectos, es decir se produce lo que se llama carrera crítica. Por ésta razón es conveniente que todos los procesos vengán sincronizados por una misma señal de reloj.

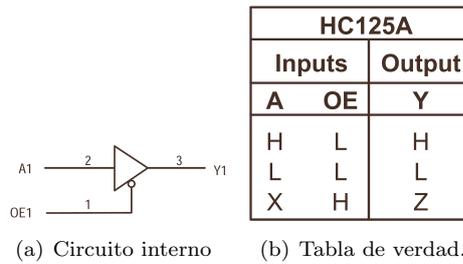


Figura 2.5: Simbología y tabla de funcionamiento de 74HC0125

Reseña histórica del JTAG

Fue el producto del trabajo realizado por un grupo de compañías interesadas en resolver el problema de controlar el conexionado de los pines de los circuitos integrados colocados en una tarjeta de circuito impreso (PCB), éste grupo, el *Join Test Action Group*, comenzó a trabajar en 1980 en el diagnostico de PCB mediante la exploración de contorno (*boundary-scan testing, BST* ó formalmente conocida como IEEE standard 1149.1[10]), es por eso que ésta norma se conoce abreviadamente como BST y también como JTAG, por el nombre del grupo creador.

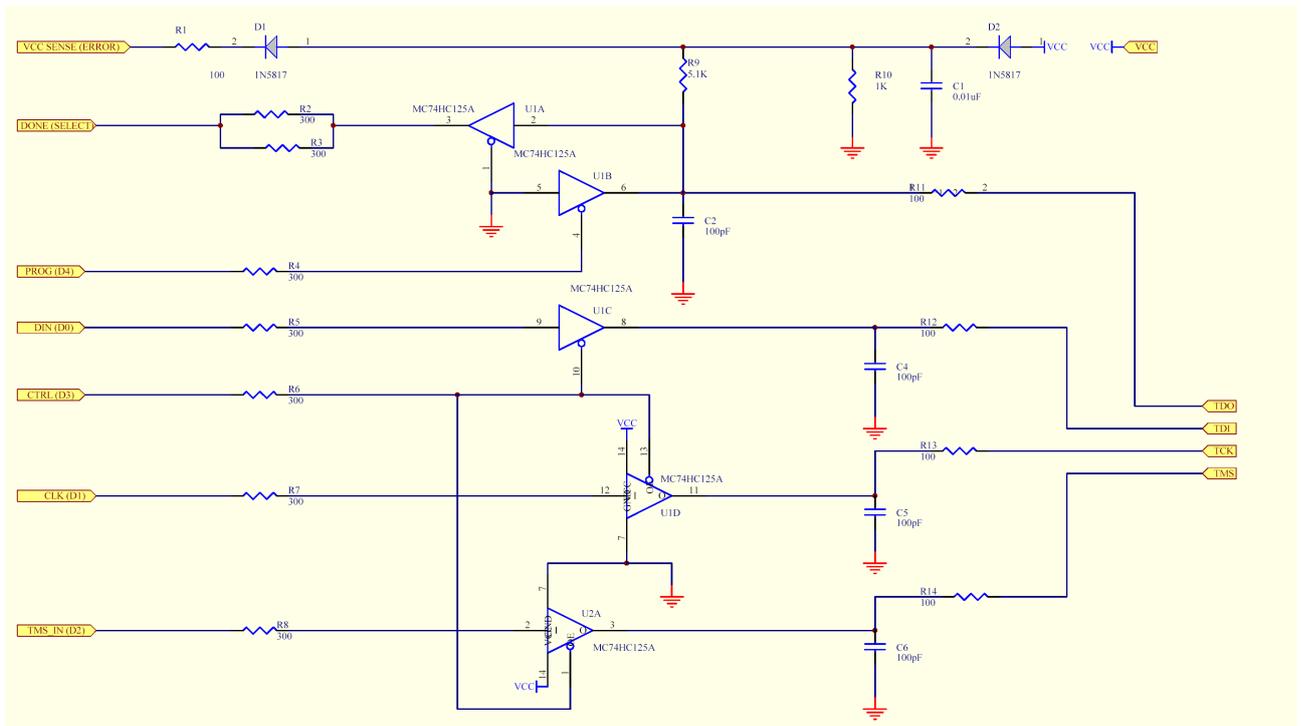


Figura 2.6: Circuito de programación JTAG[10].

El circuito programador se conecta al CPLD a través de los terminales: TDI, TD0, TCK y TMS señales que pertenecen al puerto JTAG hacer una llamada.

TDI: (*Test Data Input*) Es una entrada serial de datos, los cuales son desplazados con cada cambio al nivel alto de TCK.

TD0: (*Test Data Output*) Es una salida serial de datos, los cuales son desplazados con cada cambio al nivel bajo de TCK. Cuando no está activado el desplazamiento de datos, su estado es de alta impedancia.

TMS: (*Test Mode Select*) Entrada de control, las transiciones se producen en el cambio al nivel alto de TCK.

TCK: (*Test Clock Input*) Entrada de reloj.

TRST: (*Test Reset Input*) Activo en el nivel bajo, inicializa procesos, es opcional, en nuestra placa no contamos con ésta señal.

Las ventajas de contar con un puerto de éstas características radica en su poder para controlar el proceso de grabación, borrado, identificación del CPLD, etc.

2.4. Circuito de comunicación serie RS232

Este circuito está compuesto principalmente por el CI MAX232 que convierte las señales del estandar RS232 en señales digitales admitidas por nuestro dispositivo lógico y el circuito es como se muestra en la figura 2.7.

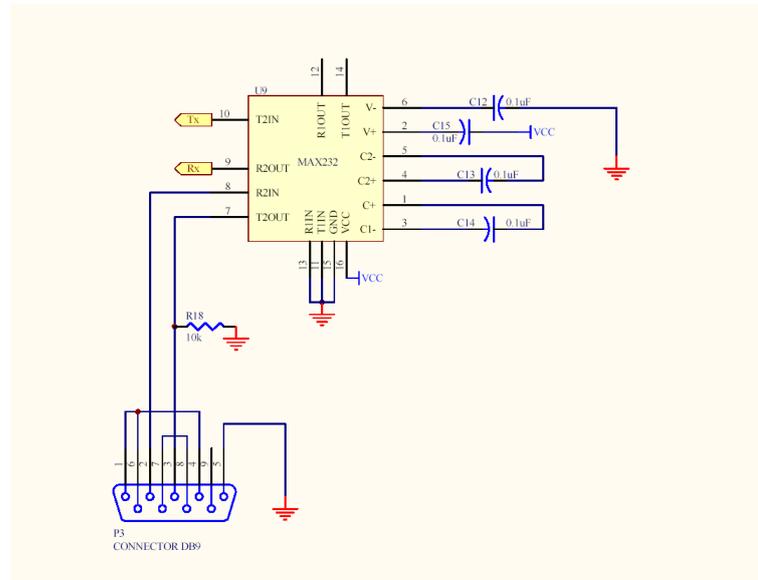


Figura 2.7: Circuito transmisión y recepción serial.

Para realizar la comunicación del CPLD con el puerto de comunicación serie, utilizamos sólo el canal 2 de la MAX232, como vemos en la figura el canal 1 tiene sus entradas (R1IN, T1IN) colocadas a potencial masa de referencia y sus salidas (R1OUT, T1OUT) no son conectadas.

Al canal 2 lo utilizamos de la siguiente forma:

R2IN: conectadas al puerto serie para la entrada de datos.

T2OUT: conectadas al puerto serie para la salida de datos.

R2OUT: conectada al pin 9 del CPLD, a través del JP1.

T2IN: conectada al pin 8 del CPLD, el cual por defecto está siempre en '1' lógico.

La ventaja de realizar la comunicación serie a través de la MAX232, es que ésta nos proporciona los niveles de las señales ya normalizadas según la EIA-232.

Éste circuito de comunicación fue puesto a prueba a una velocidad de 9600 baudios, sin embargo la hoja de datos del CI MAX232, nos garantiza velocidades de comunicación por encima de los 120Kbps.

Algunos datos útiles para realizar comunicación serie:

Sabiendo que 9600 baudios equivalen a 9600bits/seg.

Implica un tiempo de duración de $1,04167 \times 10^{-4}$ seg/bit.

Esto equivale aproximadamente a $104 \mu s$ de duración de estado (alto o bajo) de cada *bit* para la comunicación a 9600 baudios.

El ejercicio que se realizó para probar el *hardware*, creaba la señal de comunicación serie valiéndose de la entrada de reloj de 1MHz con la que se obtenía un pulso de subida cada $1 \mu s$, es por eso que, con un contador de 104, se implementó el retardo de los $104 \mu s$ necesarios para mantener establecido el nivel del *bit* de comunicación.

Breve descripción de una sección de la norma RS-232

Ésta norma de comunicación serial cubre:

- Características Eléctricas de las señales.
- Características mecánicas de los conectores.
- Descripción funcional de las señales usadas.

Existen cuatro grupos de señales; datos, control, temporización y masas. A continuación describiremos sólo las dos señales de datos y las señales de control para la configuración *Null Modem* que utiliza la placa programadora.

Señales de datos:

TXD: Datos a transmitir (en norma es BA).

El ETD³ debe poner ésta señal a nivel de la marca⁴, entre la transmisión de caracteres y también cuando no se transmitan datos.

Deben estar en estado abierto; CTS, DSR y DTR.

RXD: Recepción de datos ó datos a recibir (en la norma es BB)

Deberá estar en condición de marca mientras DCD esté en estado cerrado.

En *half-duplex* estará habilitada cuando RTS esté en estado abierto.

Señales de control:

RTS: *Request to Send* ó petición de transmitir. (en la norma es CA)

Es enviada desde el ETD hacia el ETCD para indicarle (con estado abierto) que quiere realizar una transmisión en *half-duplex*. Un estado abierto inhibe la recepción y cuando cambia de estado de cerrado a abierto, el ETCD responde cambiando CTS a estado abierto, los datos a transmitir pueden ser enviados sólo después de que ETD detecte esto en CTS.

Si RTS cambia a estado cerrado; no cambiará al abierto hasta que ETCD responda cambiando CTS a cerrado.

CTS: *Clear to Send* ó preparado para transmitir. (en la norma es CB)

Enviada por ETCD a ETD; el estado de ésta señal, indica si el ETCD está o no preparado para transmitir datos por el canal de datos.

El estado cerrado, indica ETCD en condiciones de transmitir datos, el estado cerrado indica lo contrario.

DSR: *Data Set Ready* ó aparato de datos preparado. (en la norma es CC)

Enviada por ETCD a ETD, el estado de ésta señal indica si el ETCD está o no preparado para funcionar, el estado es abierto sólo si el ETCD ha intentado establecer comunicación después que realizó las temporizaciones necesarias y generó los tonos de respuesta.

DTR: *Data Terminal Ready* ó terminal de datos preparado. (en la norma es CD)

Enviada de ETD a ETCD. El estado abierto es necesario para mantener la comunicación entre ETCD (local) y ETCD (remoto).

Su cambio a estado cerrado indica al ETCD que deberá suspender la comunicación con el ETCD remoto al finalizar la comunicación que se está realizando en ese momento.

RI: *Ring Indicator* ó indicador de llamada. (en la norma es CE)

DCD: *Data Carrier Detect* ó detección de portadora de datos. (en la norma es CF)

Éstas dos últimas no están incluidas en el equipo CPLD.

³EDT es equipo terminal de datos y ETCD es equipo de terminación del circuito de datos.

⁴Para las señales de datos la norma considera como marca ('1' lógico) cuando en ellas hay una tensión negativa y espacio ('0' lógico) cuando hay una tensión positiva. Para las señales de control y temporización se consideran estado abierto, cuando están a tensión positiva y un estado cerrado, cuando están a tensión negativa.

2.5. Circuito de prueba

Se trata de un circuito cuya función es probar que el CPLD se encuentre en estado óptimo para trabajar. Éste circuito consta de un simple pulsador y un *led*, para que sin la necesidad de conectar la expansión se puedan realizar pequeñas pruebas que utilicen éstos componentes y de ésta forma verificar nuestro CPLD; también pueden cumplir (tanto el pulsador como el *led*), alguna función dentro de un proyecto o descripción.

Se puede observar en la figura B.1 que el pulsador comparte el puerto de entrada al CPLD con el pin RX del CI MAX232 a través del JP1.

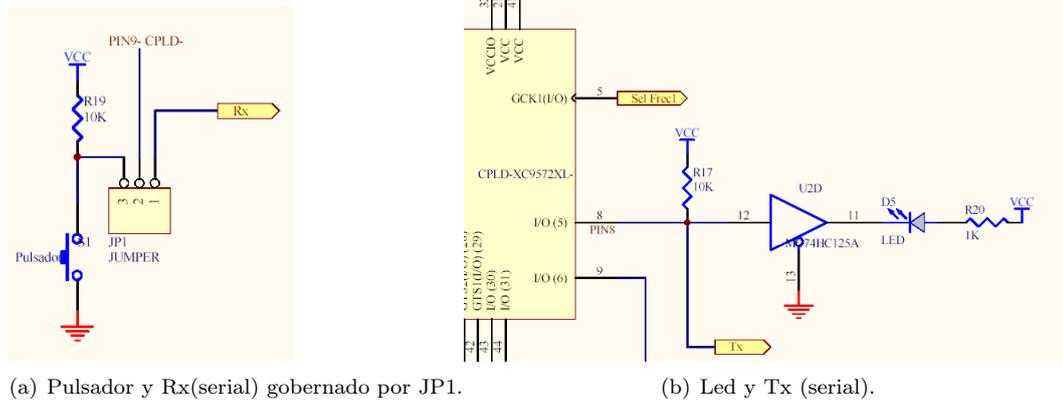


Figura 2.8: Circuito de prueba.

2.6. Circuito generador de frecuencias

Se dispone de un bloque generador de frecuencias con las cuales se pueden dar señales de reloj para sincronizar las descripciones de circuitos digitales.

El hecho de disponer de diferentes frecuencias, le permite al programador, por ejemplo, realizar un barrido para el multiplexado de los *displays* BCD de 7 segmentos, destinando una de las señales de reloj a tal fin, de lo contrario debería sintetizarse una salida de *clock* extra en alguno de los terminales del CPLD.

Para generar las distintas frecuencias se utilizará el CI 4060, cuyas frecuencias de salida se conectan a los pines 5, 6 y 7 del CPLD, ver Cuadro 3.2.

El CI 4060 nos proporciona diez divisiones de frecuencia tomando como referencia la frecuencia de oscilación del cristal de 16 MHz acoplado a sus terminales *PI* y \overline{PO} . Se obtiene así, en la salida *PO* los 16 MHz como frecuencia más alta y luego Q4 con 1MHz a partir de la cual se van realizando las correspondientes subdivisiones⁵.

Las distintas frecuencias se pueden seleccionar con los *jumper* dispuestos en los conectores JP3, JP4 y JP5. Así se tiene:

JP3 (sel_frec_1): 1MHz, 500KHz, 250KHz y FExt1 accesible desde bornera CLK IN 1.

JP4 (sel_frec_2): 125KHz, 62.5KHz, 31.25KHz, 15.625KHz y FExt2 desde CLK IN 2.

JP5 (sel_frec_3): 3.9KHz, 1.95KHz, 976.56KHz y Frec.Var⁶.

⁵Sobre éste punto hay que tener en cuenta que desde la frecuencia de 125KHz en Q7, las siguientes subdivisiones no darán señales de período entero, como resultado obtenemos en Q14 una frecuencia de salida de 976.56Hz (aproximada a 1KHz), que a los fines prácticos nos puede servir para realizar ejercitación, pero no es recomendable a los fines de diseño de circuitos de temporización crítica referida a 1 KHz.

⁶La frecuencia variable proviene del circuito oscilador de la placa de expansión, construido a partir del multivibrador astable / monoestable CI 4047 y regulada por la resistencia variable del multivuelta. Éste circuito será descrito en el capítulo siguiente.

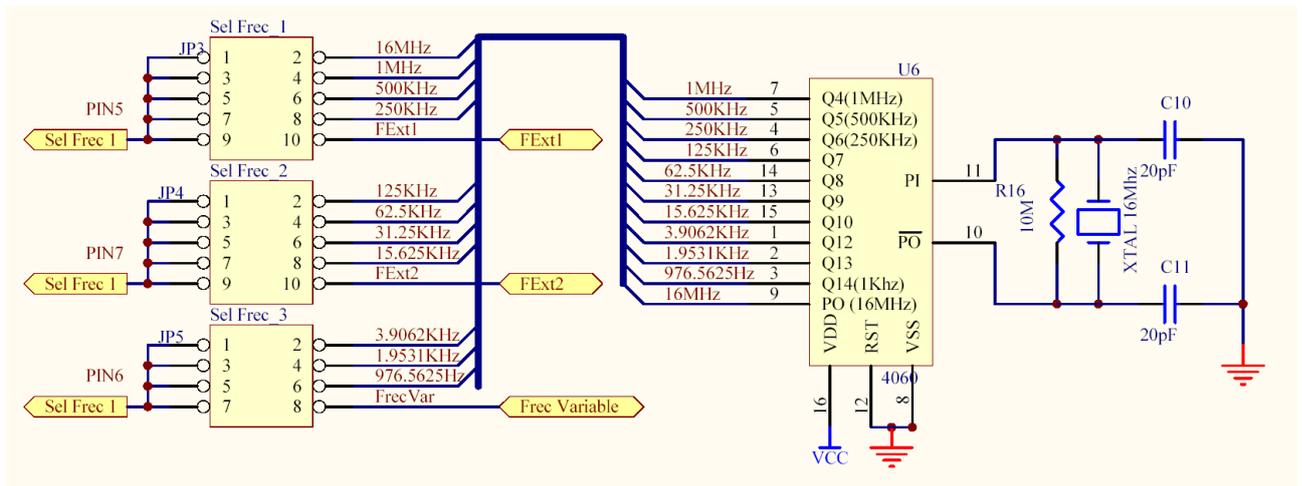


Figura 2.9: Circuito generador de frecuencias.

2.7. Designación de los terminales del CPLD en la placa:

En el cuadro 3.2 se podrá apreciar la forma en que se han dispuesto los terminales del CPLD para ser utilizados en forma conjunta con la placa de expansión. Véase que en el terminal 32 (VCC I/O), se tiene la opción de alimentación de los bloques de entrada y salida del CPLD, la misma está conectada a la salida de 5V del LM 7805C, según nivel de tensión TTL.

Para terminar con lo que se refiere a la placa de programación, se observa en la figura 2.10 un poco de lo que se está describiendo en el cuadro 3.2. En el mismo se observa como se disponen los terminales de entradas / salidas, señales de reloj y demás, en el conector DB37.

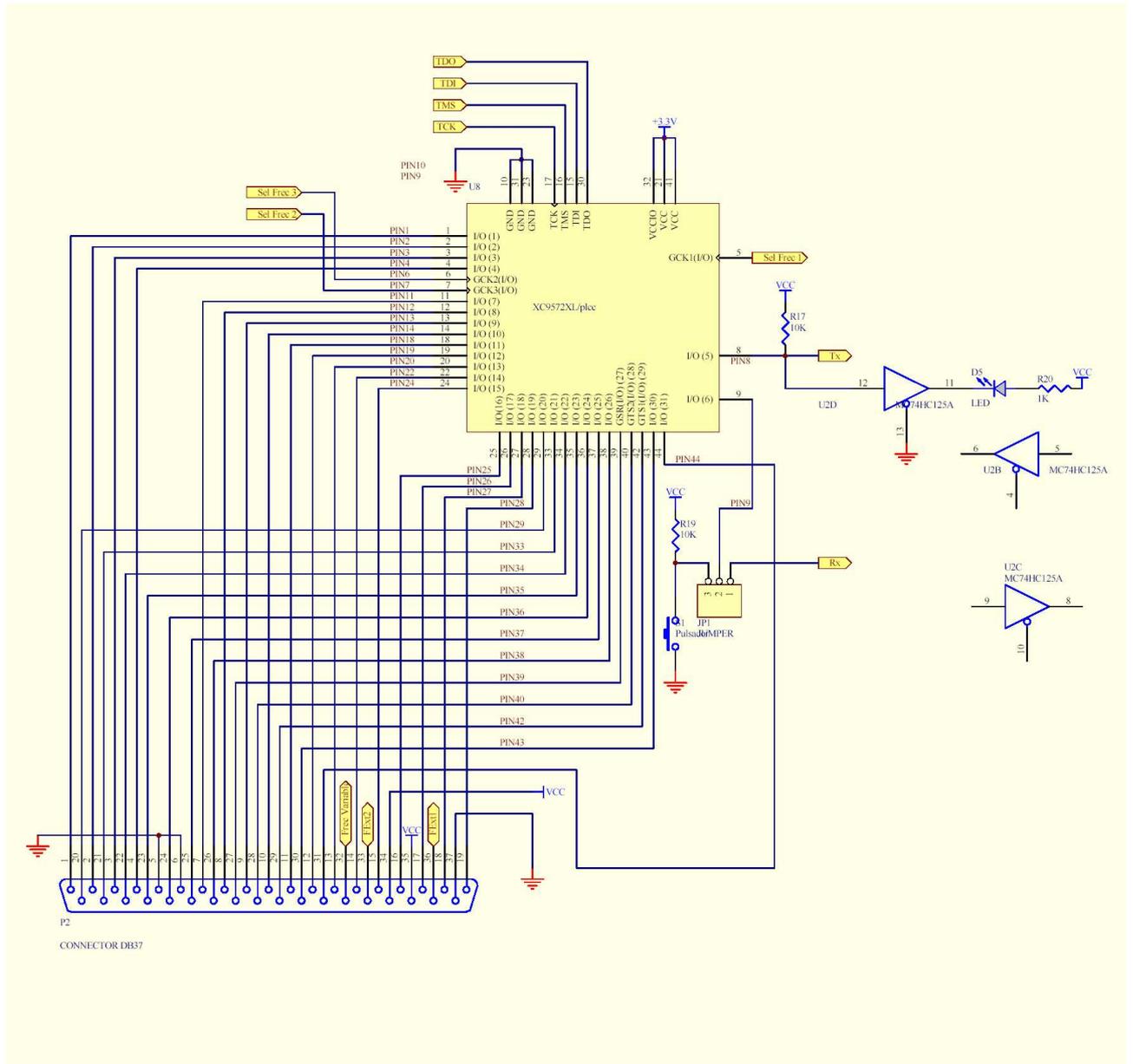


Figura 2.10: Disposición de los terminales del CPLD en la placa de programación.

Pines del CPLD	Característica de pin	Disposicion en placa
1	I/O (1)	Led[0] LSB
2	I/O (2)	Led[1]
3	I/O (3)	Led[2]
4	I/O (4)	Led[3]
5	GCK1 (I/O)	JP3(Sel.Frec.1)
6	GCK2 (I/O)	JP5(Sel.Frec.3)
7	GCK3 (I/O)	JP4(Sel.Frec.2)
8	I/O (5)	MAX232(T2IN)
9	I/O (6)	JP1=>PSR0 o MAX 232(R2OUT)
10	GND	GND
11	I/O (7)	Led[4]
12	I/O (8)	Led[5]
13	I/O (9)	Led[6]
14	I/O (10)	Led[7] MSB
15	TDI	TDI
16	TMS	TMS
17	TCK	TCK
18	I/O (11)	siete_seg[0]
19	I/O (12)	siete_seg[1]
20	I/O (13)	siete_seg[2]
21	VCC	VCC
22	I/O (14)	siete_seg[3]
23	GND	GND
24	I/O (15)	siete_seg[4]
25	I/O (16)	siete_seg[5]
26	I/O (17)	siete_seg[6]
27	I/O (18)	PSR2
28	I/O (19)	transistor_1
29	I/O (20)	transistor_2
30	TDO	TDO
31	GND	GND
32	VCC(I/O)	VCC(I/O)
33	I/O (21)	PSR1
34	I/O (22)	switch[7]
35	I/O (23)	switch[6]
36	I/O (24)	switch[5]
37	I/O (25)	switch[4]
38	I/O (26)	switch[3]
39	GSR I/O (27)	switch[2]
40	GTS2 I/O (28)	switch[1]
41	VCC	VCC
42	GTS1 I/O (29)	switch[0]
43	I/O (30)	transistor_4
44	I/O (31)	transistor_3

Cuadro 2.1: Conexión de los terminales del CPLD en la placa.

Capítulo 3

Placa expansión

3.1. Introducción

Ésta placa tiene como objetivo brindar al programador una cantidad considerable de dispositivos digitales con los cuales el usuario puede generar estímulos al CPLD y de ésta forma explotar los recursos del mismo.

Posee elementos tales como *leds*, *displays* de siete segmentos y optoacopladores con los cuales podemos corroborar el funcionamiento de nuestro diseño de *hardware* y la respuesta de la descripción que se está ejecutando.

El diseño estuvo basado en la utilización de todos los recursos de entrada / salida que dispone el CPLD, es por eso que se hace énfasis en la designación de los terminales, dejando a disposición del programador el cuadro , y un modelo en el apéndice A.2 para realizar la disposición de pines en el archivo “.ucf”, por lo tanto se recomienda que el mapeo de los pines se realice con pleno conocimiento de éste reporte, para el correcto funcionamiento de la descripción de *hardware* realizada y sobretodo para evitar cortocircuitos.

En la figura 3.1, se presenta el diagrama de la placa de expansión.

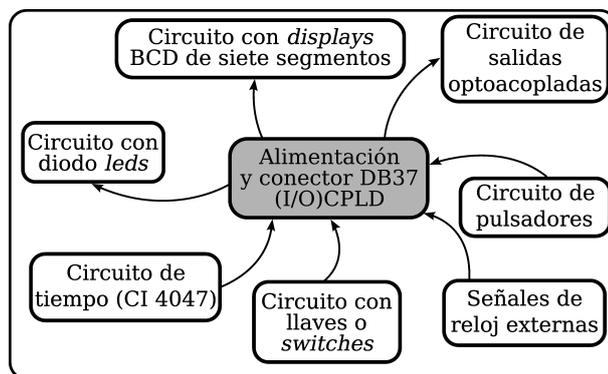


Figura 3.1: Diagrama que representa los bloques de la placa de expansión.

3.2. Alimentación de la placa y conector DB37

La alimentación de ésta placa se obtiene desde la placa programadora a través del conector DB37, es decir que recibe 5V del regulador de tensión LM 7805C que se encuentra en la placa programadora y que fue previamente descrito en éste informe.

El conector DB37 se utiliza para acoplar ambas placas sin necesidad de usar cables, se eligió éste conector debido a que la cantidad de terminales a conectar es importante. Recordemos que el CPLD posee 44 terminales, de los cuales podemos usar 31 como entrada / salida (de hecho están todos utilizados en ésta placa de expansión), el resto de los terminales se utiliza para alimentación, temporización y para el puerto JTAG. El mapeo de los pines se puede apreciar en el cuadro .

Respecto del conector DB37 cabe resaltar que proporciona como se puede apreciar en la figura , dos salidas de VCC a la placa de expansión en sus pines 34 y 35 provenientes de la placa programadora, la división de la línea de alimentación tiene como objetivo el de no sobrecargar las pistas.

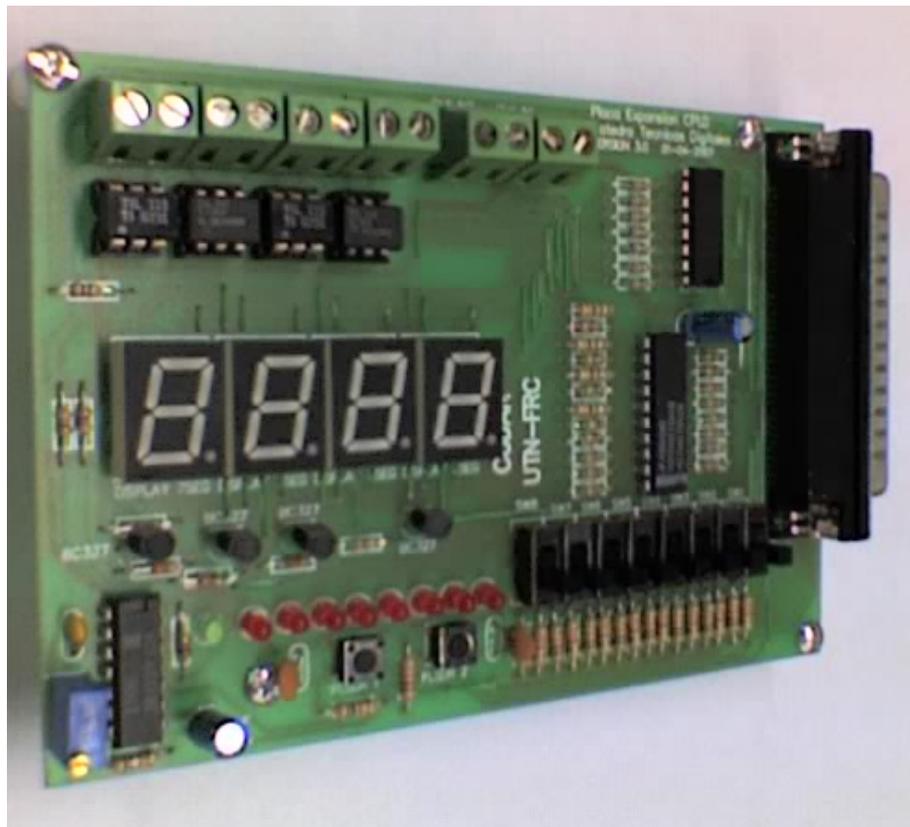


Figura 3.2: Placa de expansión.

3.3. Circuito de Pulsadores

Se cuenta con dos pulsadores que están normalmente en nivel alto, los que podrán ser utilizados en cualquier descripción realizada por el programador.

Los capacitores acoplados a los terminales P33 y P27, constituyen antirrebotes, es decir, mientras los pulsadores no sean accionados se tendrá un '1' lógico, oficiando los capacitores de memoria del nivel alto en el terminal del CPLD, de ésta manera sólo existirá un nivel bajo cuando los pulsadores sean accionados por un tiempo superior al tiempo dado por su correspondiente $\tau = RC$, y el nivel será establecido como curva creciente o decreciente.

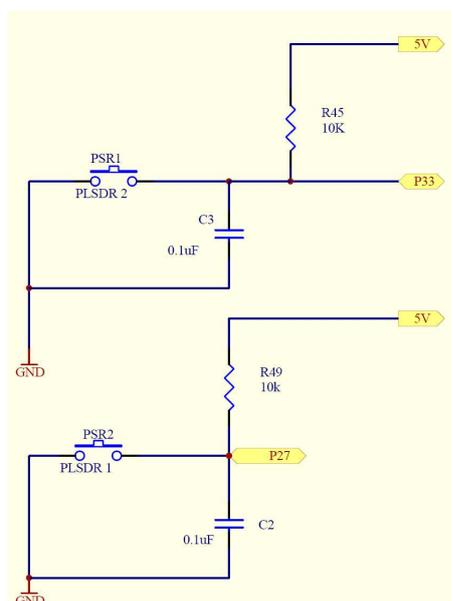


Figura 3.3: Circuito de Pulsadores.

3.4. Circuito con llaves o *switches*

En la placa de expansión contamos con ocho llaves o *switches* que serán de gran utilidad cuando se quieran emplear señales variables en tiempo de ejecución.

Se recomienda utilizarlas para ingresar registros o *bytes* de datos en alguna descripción que así lo requiera.

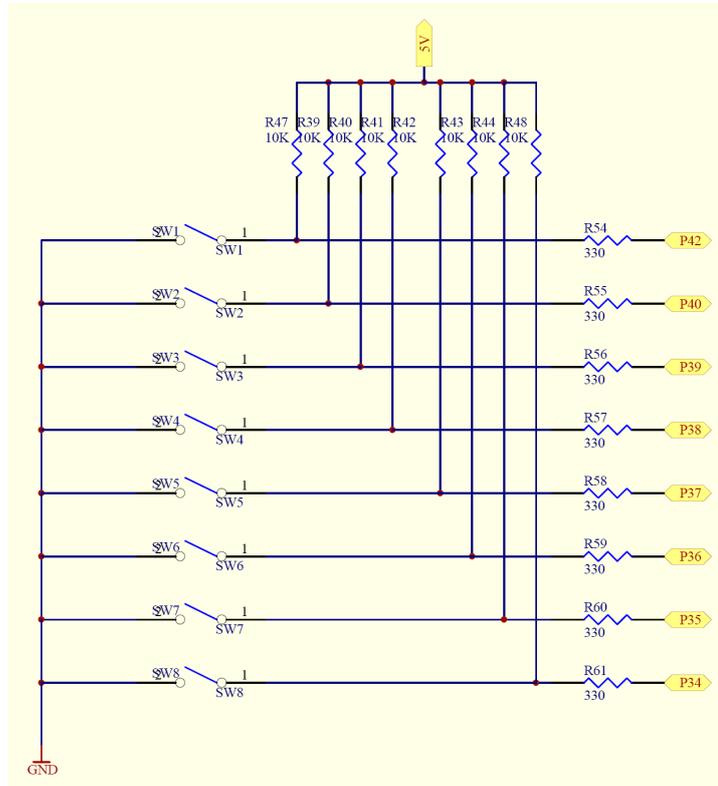


Figura 3.4: Circuito con llaves o *switches*.

3.5. Circuito de tiempo (CI 4047)

Como se describió con anterioridad, en la placa programadora se encuentran tres señales de reloj de entrada para el CPLD, obtenidas de un divisor de frecuencias constituido por un cristal de 16 MHz.

En la placa de expansión, se implementó el circuito formado por el multivibrador astable / monoestable CI 4047, con el cual el alumno ya estará familiarizado. Éste circuito posee una frecuencia ajustable y tiene como finalidad llevar el sincronismo de eventos de período extenso. Como bien se detalló en el capítulo posterior 2.6, la señal de reloj de éste circuito, puede seleccionarse con el *jumper* dispuesto en el conector JP5.

En la figura 2.9 del capítulo anterior aparece con la denominación “Frec. Var.”.

A continuación se muestra el circuito oscilador completo.

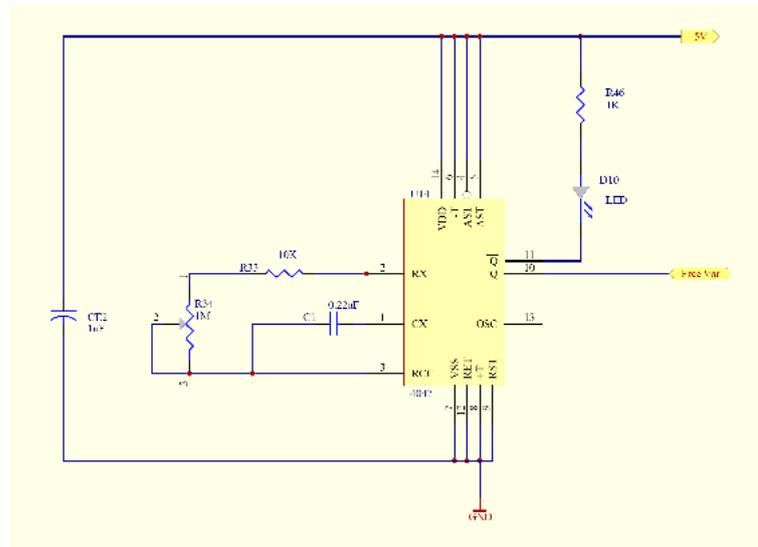


Figura 3.5: Circuito de tiempo (CI 4047).

3.6. Señales de reloj externas

La designación de éste puerto le permitirá al programador utilizar otro tipo de señales de sincronismo externo, un sensor óptico por ejemplo podría oficiar de reloj. Se podría decir que se trata de un reloj asíncrono porque el período del próximo pulso no estaría definido, si bien ésta terminología no es técnicamente correcta, sería la mas adecuada.

De ésta manera se podrán ingresar al CPLD señales que permitirán controlar los tiempos de proceso en diferentes circunstancias.

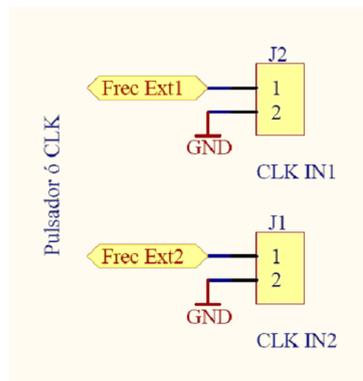


Figura 3.6: Señales de reloj externas.

3.7. Circuito con diodos *leds*

Para la visualización de datos o registros se dispone de ocho diodos *leds* en configuración cátodo común.

En la figura 3.7, se puede observar que cuatro de los diodos están conectados en paralelo con los optoacopladores de modo que, por ejemplo:

Un '1' lógico en P11 enciende el *led* para visualización y apaga el *led* del optoacoplador por lo que el transistor de acoplamiento se encontrará en corte.

Un '0' lógico en P11 apaga el *led* para visualización pero coloca el transistor en saturación a través de su *led* de acople óptico cuyo cátodo se une a masa a través del 74HC541.

En el apéndice B se puede encontrar información referida al manejo de corriente por parte de los optoacopladores.

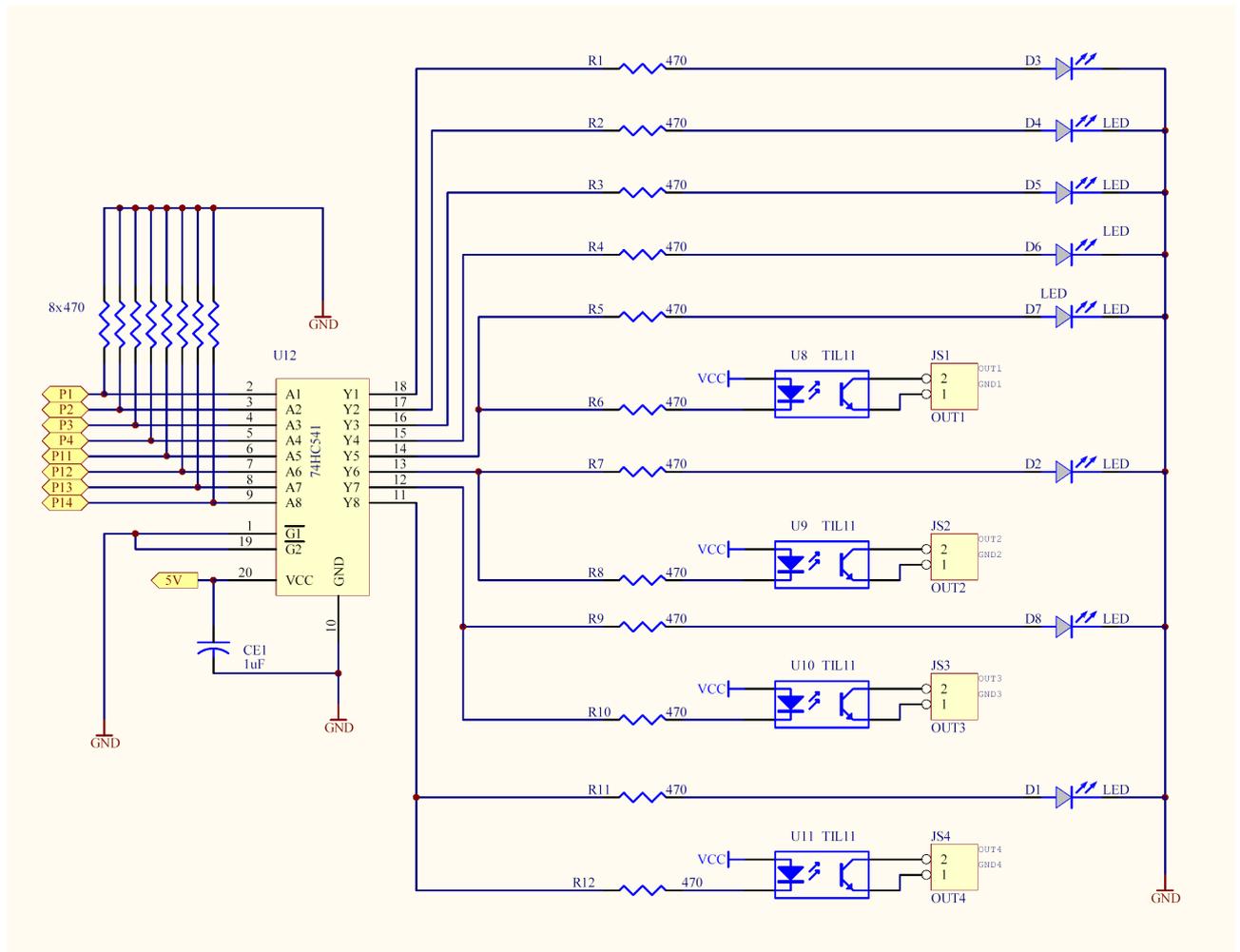


Figura 3.7: Circuito con diodos *leds* y optoacopladores.

3.8. Circuito de salidas optoacopladas

Con la utilización de optoacopladores, (en este caso se eligió en su mayoría el CI TIL116), se realiza un circuito que permite tener cuatro salidas con protección de la placa cuando se quiera trabajar con otros niveles de tensión y corriente, por ejemplo para excitar motores o relés.

Sobre estas salidas se efectuaron pruebas que se describen en el apéndice .

Éste circuito está ligado a los ocho leds de visualización como ya se explicó anteriormente, y se puede apreciar en la figura 3.7.

3.9. Circuito con *displays* BCD de siete segmentos

Uno de los bloques más importantes de la placa de expansión está constituido por cuatro displays con los cuales se realiza la visualización de datos BCD.

Los displays poseen sus ánodos conectados a los transistores PNP BC327 y son manejados por éstos al ser polarizados desde sus bases con cuatro terminales del CPLD.

Se puede apreciar en la figura 3.8 que los segmentos de los cuatro displays están conectados en paralelo, de forma que se puedan multiplexar¹.

¹De hecho es recomendable que sean multiplexados para que nunca estén encendidos al mismo tiempo mas de siete segmentos, disminuyendo así el consumo de corriente.

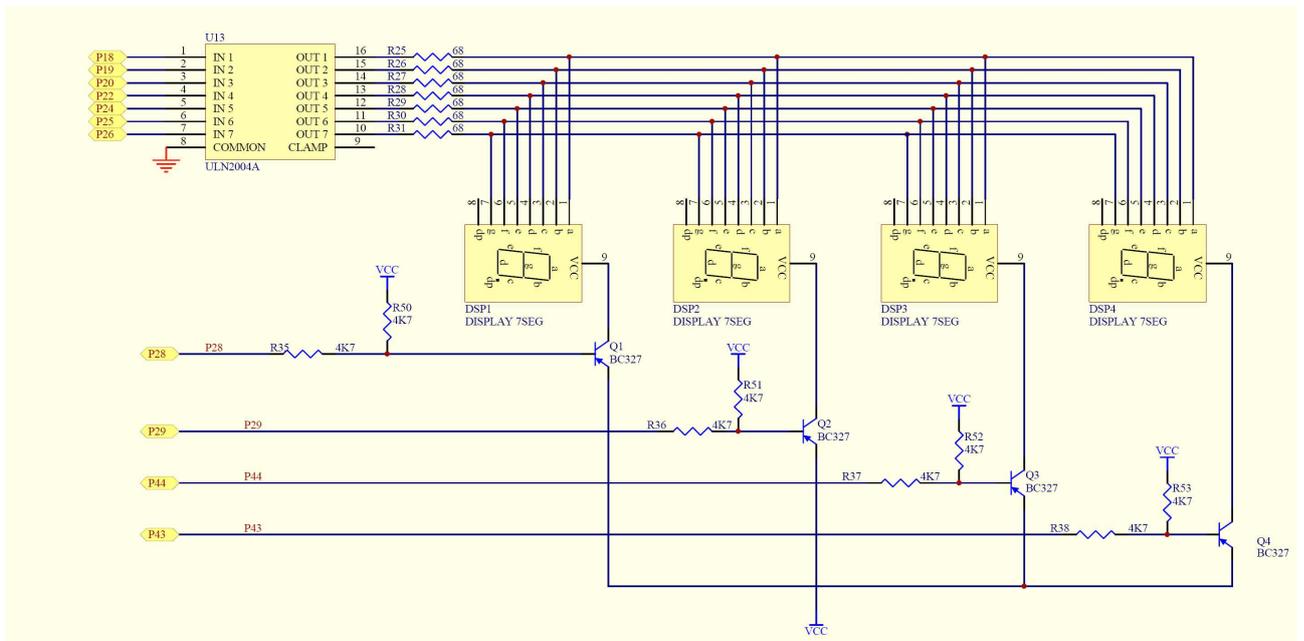


Figura 3.8: Circuito con *displays* BCD de siete segmentos.

Apéndice A

Información de programación

A.1. La Entidad (*Entity*)

La entidad es una parte del programa de gran importancia, en el mismo se describe con detalle los puertos de comunicación del CPLD con el exterior, así también los componentes creados en el mismo. Se describe con mayor detalle en el “Tutorial de práctica sobre CPLD” que será editado a continuación de este informe. Se espera que los estudiantes tomen como práctica sana de programación una entidad similar a la proporcionada a continuación, no solo por estandarización sino también por que esta entidad especifica todos los puertos disponibles en el equipo.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity top_prueba_expansion is
7 Port (
8 -- Definimos las entradas al CPLD
9 clk1 : in STD_LOGIC;
10 clk2 : in STD_LOGIC;
11 clk3 : in STD_LOGIC;
12 pulsador_rx_prog : in STD_LOGIC;
13 pulsador1 : in STD_LOGIC;
14 pulsador2 : in STD_LOGIC;
15 llaves : in STD_LOGIC_VECTOR(7 downto 0);
16-- Definimos las entradas al CPLD
17 led_tx_prog : out STD_LOGIC;
18 transistor1 : out STD_LOGIC;
19 transistor2 : out STD_LOGIC;
20 transistor3 : out STD_LOGIC;
21 transistor4 : out STD_LOGIC;
22 leds : out STD_LOGIC_VECTOR(7 downto 0);
23 siete_seg : out STD_LOGIC_VECTOR(6 downto 0));
24 end top_prueba_expansion;
25
26 architecture Behavioral of top_prueba_expansion is
27 begin
28
29--Descripción.....
30
31 end Behavioral;
```

A.2. El archivo UCF

Los archivos UCF son creados por el *software* pero el mismo es definido por el programador por lo tanto es aconsejable seguir éste modelo el cual podrá, de alguna forma, generalizar o estandarizar los proyectos. Éste UCF responde al proyecto de prueba que se utilizó el mismo se presentará mas adelante en la sección.

```
#PACE: Start of Constraints generated by PACE
#PACE: Start of PACE I/O Pin Assignments
NET "clk1" LOC = "P5" ;
NET "clk2" LOC = "P7" ;
NET "clk3" LOC = "P6" ;
NET "led_tx_prog" LOC = "P8" ;
NET "leds<0>" LOC = "P1" ;
NET "leds<1>" LOC = "P2" ;
NET "leds<2>" LOC = "P3" ;
NET "leds<3>" LOC = "P4" ;
NET "leds<4>" LOC = "P11" ;
NET "leds<5>" LOC = "P12" ;
NET "leds<6>" LOC = "P13" ;
NET "leds<7>" LOC = "P14" ;
#NET "pulsador1" LOC = "P27" ;
#NET "pulsador2" LOC = "P33" ;
NET "pulsador_rx_prog" LOC = "P9" ;
NET "siete_seg<0>" LOC = "P18" ;
NET "siete_seg<1>" LOC = "P19" ;
NET "siete_seg<2>" LOC = "P20" ;
NET "siete_seg<3>" LOC = "P22" ;
NET "siete_seg<4>" LOC = "P24" ;
NET "siete_seg<5>" LOC = "P25" ;
NET "siete_seg<6>" LOC = "P26" ;
NET "llaves<0>" LOC = "P42" ;
NET "llaves<1>" LOC = "P40" ;
NET "llaves<2>" LOC = "P39" ;
NET "llaves<3>" LOC = "P38" ;
NET "llaves<4>" LOC = "P37" ;
NET "llaves<5>" LOC = "P36" ;
NET "llaves<6>" LOC = "P35" ;
NET "llaves<7>" LOC = "P34" ;
NET "transistor1" LOC = "P28" ;
NET "transistor2" LOC = "P29" ;
NET "transistor3" LOC = "P44" ;
NET "transistor4" LOC = "P43" ;
#PACE: Start of PACE Area Constraints
#PACE: Start of PACE Prohibit Constraints
#PACE: End of Constraints generated by PACE
```

Apéndice B

Mediciones

B.1. Optoacopladores

Se realizó mediciones sobre las salidas de todos los optoacopladores de los diez equipos que se van a poner a disposición. Esto se hizo para observar el comportamiento y las limitaciones que se tendrá en el momento de cargar las salidas de los mismos. El circuito de prueba se puede ver en la figura B.1 y el cuadro B.1 presenta las mediciones.

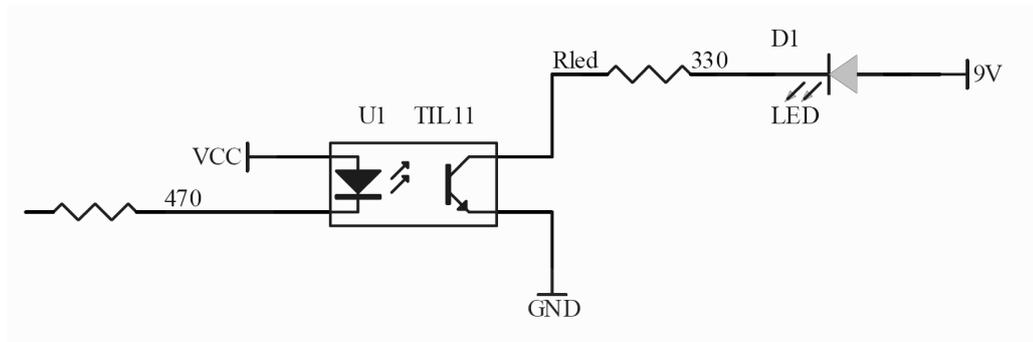


Figura B.1: Circuito de prueba.

equipo CPLD: CUDAR			
salida	V	mA	optoacoplador
<i>out1</i>	5.03	26.11	TIL115 - TI035
<i>out2</i>	5.31	24.8	TIL115 - TI036
<i>out3</i>	5.3	25.2	TIL115 - TI037
<i>out4</i>	5.04	24.48	TIL115 - TI038

Cuadro B.1: Mediciones sobre el equipo CPLD del CUDAR.

Colaboraron para realizar el presente informe

Edición y redacción¹:

Gaspar Acevedo.

Luis Alberto Guanuco.

Revisión de redacción:

Ing. Pablo Cayuela.

Ing. Sergio Olmedo.

Ing. José Martínez.

Héctor Daniel Duran.

Aportes técnicos y revisión general:

Ing. Sergio Olmedo.

Ing. José Martínez.

Daniel Durán.

Agradecimientos

Se agradece a los becarios:

Esteban Pereyra Toledo.

Héctor Daniel Durán.

Marcelo Cassanovas.

Ing. Darío Chans.

Gaspar Acevedo.

Luis Alberto Guanuco.

Por su desinteresada participación en la elaboración de éstos dispositivos de entrenamiento.

Se le agradece a los ingenieros Sergio Olmedo y Pablo Cayuela, los cuales pusieron todo su esfuerzo para cumplir con éste objetivo.

Se le agradece especialmente al ingeniero Rodolfo Cavallero, quién es el impulsor de éste entrenador, el cual será utilizado principalmente en el dictado de la materia Técnicas Digitales I.

¹La edición de éste reporte fue realizado integramente bajo el editor LyX[9], el cual es una herramienta L^AT_EX conocido en el ambiente de la edición de documentos. LyX es una de los editores mas utilizados por su facil manejo como también de las capacidades que posee.

Bibliografía

- [1] "VHDL Lenguaje para síntesis y modelado de circuitos" - Fernando Pardo Carpio y José A. Boluda Grau.
- [2] "DIGITAL DESIGN Principles & practices" - John F. Wakerly.
- [3] "Digital Design with CPLD Applications & VHDL" - DUECK.
- [4] "The designer's guide to VHDL" - Peter J. Ashenden.
- [5] "Síntesis de circuitos digitales un enfoque algorítmico" - JEAN-PIERRE DESCHAMPS.
- [6] "Fundamentos de lógica digital con diseño VHDL" - Stephen Brown-Zvonko Vranesic.
- [7] "FPGA-BASED SYSTEMS" - Trabajo presentado por cudar en SURLABS.
- [8] <http://www.xilinx.com/xc9500>
- [9] <http://www.lxy.org/>
- [10] Artículo IEEE Standard 1149.1.
- [11] "Interconexión de periféricos a microprocesadores - Serie: Mundo Electrónico" - Editorial Marcombo.