

PRÀCTICA 3. INTRODUCCIÓ AL CAD DIGITAL (isp LEVER)

3.1.Introducció

En aquesta pràctica s'introdueix l'ús del programa de CAD isp LEVER mitjançant la realització d'un generador de seqüència pseudoaleatòria, un detector de seqüència programable i la verificació de la seva funcionalitat. La informació bàsica del funcionament i opcions d'aquest programa es troba en el manual publicat a part. En ell s'hi descriu globalment el programa i les seves comandes i opcions necessàries per desenvolupar les pràctiques. A més, en cas necessari, es poden consultar els manuals de referència complets disponibles al Laboratori en format PDF.

La durada d'aquesta pràctica és de 1 sessió

Els objectius de la pràctica són:

- Introduir les eines de disseny de sistemes digitals amb dispositius lògics programables (isp LEVER com a exemple).
- Introducció al procés de disseny d'un sistema digital mitjançant la captura d'esquemàtics i la descripció hardware en llenguatge d'alt nivell (ABEL-HDL de isp LEVER, **Hardware Description Language**).
- Seleccionar la millor alternativa en funció del disseny: Entrada d'esquemàtic o descripció HDL.
- Posar de manifest la importància de la simulació i verificació en el disseny digital.

3.2. Treball previ

TREBALL PREVI

Recordeu que cal presentar una còpia del treball previ a l'inici de la sessió de pràctiques.

- a) Llegiu aquest manual i els manuals de *Introducció a isp LEVER* i *Abel*.**
- b) Dibuixeu el diagrama d'estats del detector de seqüència síncron de 3 bits que s'especifica en aquest apartat.**
- c) Descriviu la màquina d'estats en llenguatge ABEL.**
- d) Escriviu un programa de vectors de test en format ABEL que comprovi el pas del detector de seqüència per totes les transicions.**

a) En el **manual d'Introducció a isp LEVER** es resumeixen alguns procediments necessaris per poder realitzar la pràctica. **Es recomana llegir-lo detingudament** i consultar-lo sempre que sigui necessari durant el treball de laboratori.

b) Per al disseny del detector de seqüència s'empraran circuits simples digitals genèrics (portes i biestables sense senyals asíncrons) o una descripció Abel-HDL.

El detector de seqüència té quatre entrades i una sortida (Figura 1).

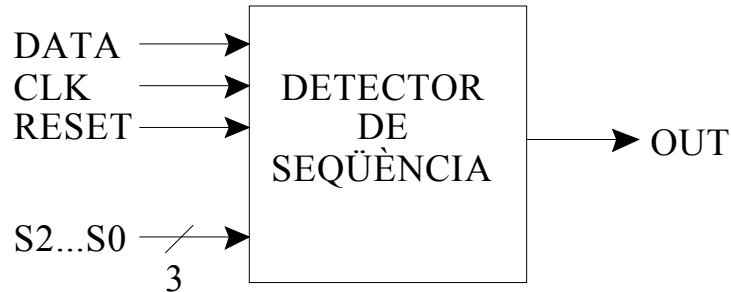


Figura 1. Detector de seqüència programable

Entrades:

CLK: Senyal de rellotge del sistema.

DATA: Seqüència de bits sèrie sincronitzada pel flanc de pujada de CLK. Els bits arriben per la línia DATA de menys significatiu a més significatiu.

RESET: Inicialitza de forma síncrona el sistema a un estat conegut. Aquest senyal és síncron amb el flanc de pujada del rellotge (és a dir, el *reset* ha de tenir efecte en el flanc de pujada de CLK).

S2 · · S0: Entrades que indiquen la seqüència que es vol detectar.

Sortida:

OUT: Senyal que ha de posar-se a '1' únicament durant el cicle de rellotge següent a l'últim bit de la seqüència programada. No es pot tornar a activar fins com a mínim tres cicles de rellotge després de l'activació (és a dir, els bits d'una seqüència vàlida no poden formar part de la següent seqüència vàlida).

Els senyals **DATA** i **RESET** es sincronitzen amb el flanc de pujada de CLK.

Dibuixeu el diagrama d'estats del detector de seqüència. Si cal, definiu les variables que creieu convenient.

IMPORTANT: Dissenyeu el detector com una màquina d'estats de tipus MOORE (no MEALY), és a dir, que la sortida només depengui directament de l'estat present, i no de les entrades.

c) Feu la descripció en llenguatge ABEL-HDL del detector de seqüència a partir del seu diagrama de màquina d'estats. En el **manual de referència d'introducció al llenguatge ABEL-HDL** trobareu l'estructura bàsica que ha de tenir un fitxer de descripció hardware d'un sistema digital. També hi trobareu exemples de màquines d'estats descrites amb aquest llenguatge d'alt nivell. Podeu també consultar des de l'ordinador del vostre lloc de treball tot el manual de referència del llenguatge ABEL-HDL de isp LEVER llegint el fitxer en format *pdf* "ABEL-HDL_ReferenceManual"

amb el programa Acrobat Reader. Aquest fitxer es troba a la carpeta H:\leii\Manuals_isplayver_Lattice.

d) Definiu un fitxer de vectors de test (formes d'ona dels senyals d'entrada i valors esperats dels senyals de sortida) que permetin comprovar que el disseny realitzat es comporta de la forma esperada, passant per totes les transicions definides al diagrama d'estats. Aquests vectors s'introduiran mitjançant un fitxer en llenguatge ABEL. (Veure manual del isp LEVER). Teniu en compte que isp LEVER pressuposa que els biestables tenen les sortides a "0" quan comença la simulació dels vectors de test, però que no obligatòriament ha de ser així a la realitat. Cal doncs fer un *reset* inicial. Verifiqueu tant l'activació de la sortida, si la combinació és l'adequada, com la no activació si la seqüència és incorrecta.

3.3. Treball al laboratori

➤ PART I: Disseny mitjançant la captura d'esquemàtics

Algunes operacions que s'indiquen es troben descrites amb més detall al **manual d' isp LEVER**. És important seguir l'ordre que s'indica.

3.3.1. Creació de nou projecte.

Convé que per al projecte creeu un subdirectori dins el vostre directori de pràctiques. Executeu l'icona de ISP lever, i creeu un nou projecte dins aquest subdirectori.

Podem associar el concepte de projecte a la base de dades que conté la informació del desenvolupament d'un sistema. Un projecte pot contenir un nombre indefinit d'esquemes però només un fitxer de vectors de test actiu (el fitxer es pot anar canviant per simular diferents esquemes).

Si volem fer simulacions parcials en un projecte cal que el subsistema que volem simular sigui el de major jerarquia. Desplegant l'opció *Source* del menú de ISP lever apareixen les comandes que permeten mantenir la llista dels esquemàtics actius i del fitxer de vectors de test. Així, amb la comanda *Remove* es poden treure de la llista *Sources in Project* fitxers esquemàtics i el fitxer de vectors de test (tot i que no s'esborren del directori). La comanda *Import* permet tornar a incorporar a la llista *Sources in Project* fitxers d'esquemàtic o un fitxer de vectors de test.

3.3.2. Exemple d'entrada d'esquema: Generador de seqüència pseudoaleatòria.

Com a exemple de disseny amb entrada d'esquemes, dissenyarem un circuit generador de seqüència pseudoaleatòria. Un exemple típic és el LFSR (Linear Feedback Shift Register). Aquest tipus de circuit genera seqüències pseudoaleatòries de longitud $2^n - 1$, on n és el nombre de biestables que conté. A la figura 2 es mostra l'esquema corresponent a un LFSR de 7 bits. En aquest cas, la descripció com a màquina d'estats pot ser més complicada que la seva descripció mitjançant un logigrama.

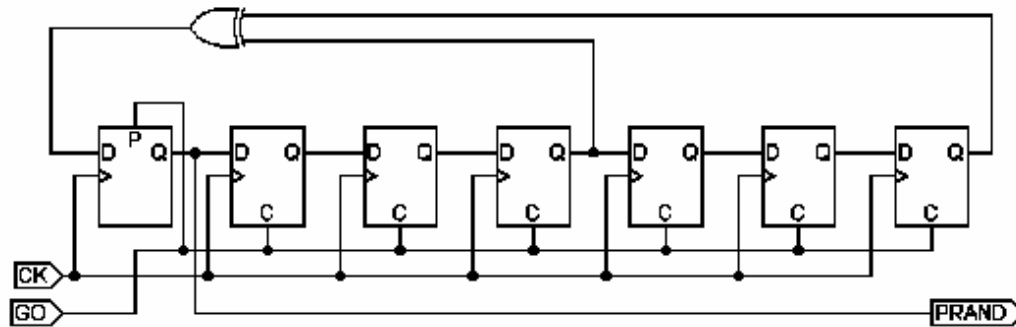


Figura 2. LFSR de 7 etapes

Introduïu l'esquema de la figura 2 en un nou fitxer esquemàtic dibuixant els components adjacents i les interconnexions. És imprescindible que totes les línies d'entrada o de sortida del mòdul tinguin una etiqueta associada i estiguin definides com a entrada o sortida.

3.3.3. Simulació.

Creeu un fitxer de vectors de test en format ABEL per comprovar que efectivament es genera una seqüència pseudoaleatòria. Per iniciar el circuit cal posar a '1' el senyal GO durant un període de rellotge. Tot seguit es deixa a '0' per a que el circuit evolucioni. Verifiqueu que la seqüència pseudoaleatòria es repeteix amb un període de $2^n - 1$.

Traieu un llistat de l'esquema, el fitxer ABEL, i les formes d'ona de simulació.

3.3.4. Generació de símbol.

Genereu un símbol per al disseny de l'LSFR, de forma que es podrà utilitzar com a element de la llibreria d'usuari en futurs dissenys.

➤ PART II: Disseny mitjançant llenguatges de descripció hardware (ABEL)

La descripció amb llenguatges HDL (*Hardware Description Language*) és una alternativa que s'està imposant a l'entrada d'esquemàtics. Entre moltes raons una és la facilitat per descriure sistemes complexos mitjançant descripció d'alt nivell i funcional que combinat amb eines de síntesi permet la realització de circuits complexos en un temps molt menor que dissenyant a baix nivell. En aquesta part es comprovarà la senzillesa amb que es pot realitzar el detector de seqüència a partir del diagrama d'estats.

3.3.5. Descripció en llenguatge ABEL.

En el mateix directori de projecte que heu creat, introduïu la descripció en llenguatge ABEL del detector de seqüència en un nou fitxer *ABEL_HDL Module*. Per fer això, en *Sources in Project* feu *New... - ABEL-HDL Module* i us sortirà una nova finestra. Doneu un nom al mòdul i el mateix al fitxer. Premeu *Ok* i entrareu a l'editor d'ABEL-HDL. Ara ja podeu entrar el vostre fitxer amb la descripció del detector de seqüència, que **heu fet com a estudi previ**, en llenguatge ABEL.

3.3.6. Simulació del detector de seqüència programable.

Feu servir per a la simulació els vectors de test preparats a l'estudi previ. Obriu un nou fitxer *ABEL Test Vectors* i copieu-hi els vectors de test o bé importeu-lo fent un *Source* → *Import...*

Observeu les formes d'ona temporals. Comproveu mitjançant les formes d'ona si teniu errors al vostre disseny. En cas d'errors, refeu la descripció.

Traieu un llistat de la descripció en format ABEL i de les formes d'ona de simulació.

3.3.7 Disseny global.

Sense sortir del projecte en curs, importeu el mòdul que heu fet a la part I del generador de seqüència pseudoaleatòria fent *Source - Import...* i el nom del fitxer de l'esquemàtic. Després obriu un nou esquemàtic (*New - Schematic*), que contindrà el disseny global (li podeu dir, per exemple, "top"), feu *Add - Symbol* i carregueu el símbol del generador de seqüència pseudoaleatòria. A continuació heu de crear i carregar el símbol del detector de seqüència que acabeu de fer. Per a això, feu dins del full de l'esquemàtic *Add - New Block Symbol* i us sortirà una finestra nova. Heu d'entrar, on posa *Block Name*, el nom del mòdul que heu dissenyat prèviament en format ABEL. Després, escriureu els noms dels pins d'entrada i sortida que heu fet servir en el disseny del detector a les caselles *Input Pins* i *Output Pins*, respectivament. Un cop fet això, premeu *Run* i us sortirà en el full de l'esquemàtic el símbol del detector.

Nota: Recordeu que isp LEVER diferencia entre majúscules i minúscules en els noms dels pins.

Ja tan sols resta interconnectar el generador de seqüència amb el detector i fer la simulació del conjunt escrivint un fitxer de vectors de test que posi en marxa el generador de seqüència pseudoaleatòria per poder comprovar un gran nombre de seqüències d'entrada al detector. Simuleu la detecció de diferents seqüències, per exemple, 010, 110 i 001.

PRÀCTICA 4: CONTROL DEL COMANDAMENT A DISTÀNCIA

4.1.Introducció

En aquesta pràctica s'utilitza el programa de ISP lever per dissenyar el sistema digital programable del comandament a distància. La descripció bàsica del funcionament del comandament a distància es troba a l'apartat 4.2 i l'esquema de blocs del comandament a distància i el de cadascun dels subblocs a l'apartat 4.3.

La durada d'aquesta pràctica és de 3 sessions

Els objectius de la pràctica són:

- Dissenyar un sistema digital,
- Gravar el disseny en un dispositiu programable del tipus ispLSI i
- Verificar el disseny amb un analitzador lògic.

Aquest dispositiu ens permetrà controlar el teclat matricial del comandament a distància i enviar el senyal sèrie adequat a l'emissor d'infrarojos. El teclat consta de 16 tecles (fig. 1) i per tant s'haurà de detectar quan una tecla ha estat polsada, identificar-la i enviar un senyal sèrie modulad en amplitud a 32.768Hz que pugui ser desmodulat pel receptor d'infrarojos i enviat directament a la UART del microcontrolador.



Fig1: Esquema extern del teclat

La pràctica consta de dues parts.

- **Part I:** S'haurà de dissenyar el circuit lògic que controlarà el comandament a distància a partir de les especificacions del circuit, comprovar el seu funcionament per parts i de forma global amb simulacions. La durada d'aquesta part és de 2 sessions de pràctiques.

- **Part II:** Es genera el fitxer de programació del dispositiu en format JEDEC. A partir del fitxer JEDEC es programarà el circuit ispLSI i es comprovarà físicament el seu funcionament amb l'analitzador lògic. La durada d'aquesta part és de 1 sessió de pràctiques.

TREBALL PREVI

a) Part I

Llegir amb deteniment els apartats 4.1, 4.2, 4.3, 4.4 i 4.5 d'aquest manual

Sessió 1: Realitzar els dissenys corresponents als subblocs de l'esquema de blocs del comandament a distància (qüestions Q1 i Q2 de l'apartat 4.4.)

Sessió 2: Respondre a les qüestions Q3..Q8 de l'apartat 4.4.

b) Part II

Llegir amb deteniment els apartats 4.6 i 4.7 d'aquest manual

a) Per al disseny dels subblocs del comandament a distància haureu de seleccionar la millor alternativa per cadascun: en uns serà més convenient entrar l'esquemàtic mentre que en d'altres serà millor fer una descripció HDL. A l'hora d'escriure el fitxer de vectors de test cal fer una verificació funcional que permeti assegurar la funció electrònica digital encomanada a cada subbloc. Es convenient que la verificació del funcionament del subbloc sigui completa, que no exhaustiva, abans de passar al següent subbloc.

b) La verificació de la PLD la realitzareu des del vostre lloc de treball. Adquiriu amb l'analitzador lògic les mateixes formes d'ona que haguéssiu simulat, captureu-les i compareu-les.

4.2 Funcionament del comandament a distància

A la següent figura es pot veure un esquema simplificat del comandament a distància

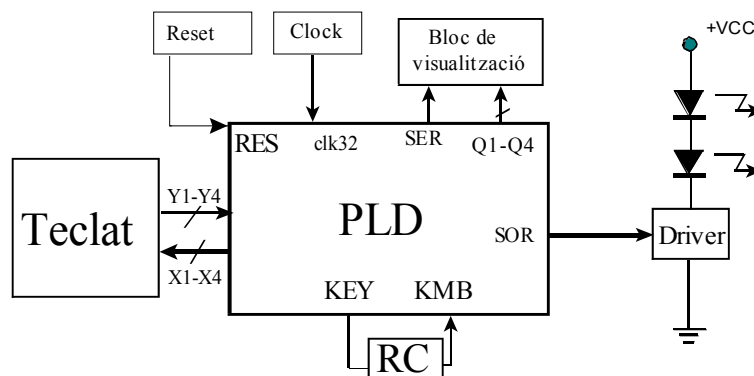


Fig.2 : Esquema simplificat del comandament a distància

El teclat internament és una matriu d'interruptors amb vuit terminals (figura 3).

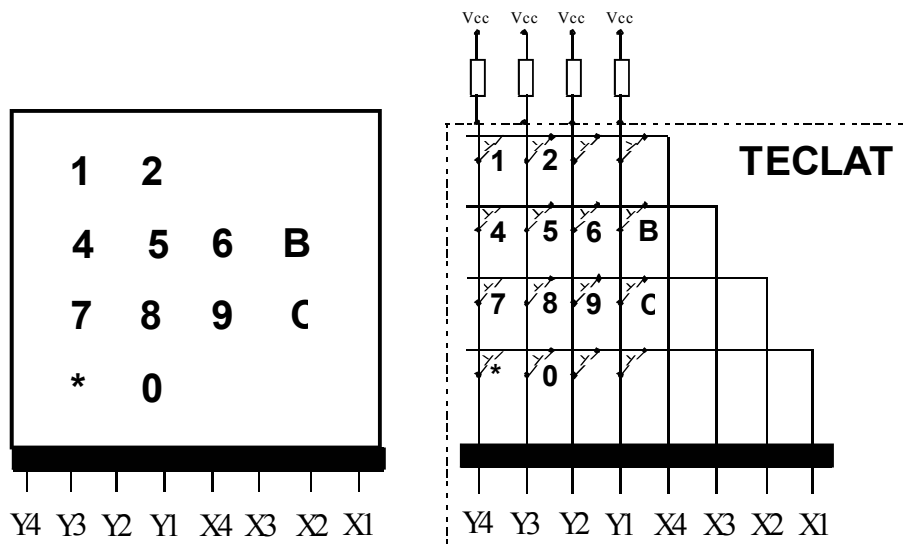


Fig.3 : Esquema intern del teclat que inclou les resistències de PULL-UP externes

Per detectar si s'ha pitjat una tecla i identificar-la cal realitzar un escombrat en els senyals d'entrada al teclat (X_i) (files) i veure quin valor de tensió ('1' ó '0') trobem en els terminals de sortida del teclat (Y_i) (columnes). Si només un dels senyals X_i és igual a '0' (p.e. X_3), el valor dels senyals Y_i només serà '0' en les columnes que tinguin una tecla pulsada al nivell de la fila "marcada" pel zero present al senyal X_i (seguint l'exemple, a la fila 3).

És a dir:

- Si no hi ha cap tecla pulsada els terminals X_i i Y_i estan aïllats. Els senyals Y_i valen tots '1'.
- Si posem als senyals X_i la combinació '1101', per exemple, i no hi ha cap tecla pulsada els senyals Y_i continuen valen tots '1'.
- Si als senyals X_i continuem posant la combinació '1101', quan fem la lectura dels senyals Y_i llegirem un '0' a totes les columnes que tinguin una tecla pulsada a la fila 3. Si la tecla '6' està pulsada llegirem als Y_i : '1101'. Si '4' i '5' estan pulsades llegirem '0011'.

Aquesta estructura ens permet anar activant seqüencialment fila per fila (posant un '0' a aquesta fila i deixant la resta de X_i a '1') i identificar totes les columnes que tenen una tecla pulsada, dins d'aquesta fila. Aquesta manera de realitzar l'escombrat del teclat pressuposa que a un instant determinat tan sols es prem una tecla, o a tot estirar, més d'una però d'una única fila.

La part central del comandament a distància és la PLD que hauréu de programar en aquesta pràctica. Aquest dispositiu haurà de realitzar seqüencialment un escombrat del teclat i quan sigui el cas, identificar la tecla pulsada. Un cop identificada haurà de proporcionar un senyal sèrie modulats amb un rellotge de 32.768Hz i enviat directament a l'emissor d'infrarojos.

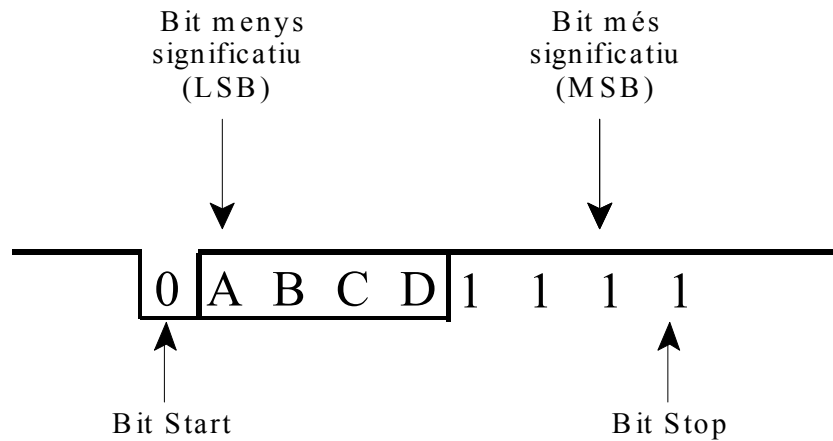


Fig 4: Format de transmissió sèrie. Els bits ABCD identifiquen la tecla pulsada.

El format sèrie serà el següent (figura 4):

- 1 bit de START,
- 1 bit de STOP,
- 7 bits de DADES: dels quals 4 ens serviran per identificar la tecla pulsada (bits ABCD) i la resta prendran el valor '1'.
- La velocitat de transmissió serà de 512 baud.

4.3 Descripció del sistema a nivell de blocs

L'esquema de blocs del comandament a distància es pot veure a la següent figura:

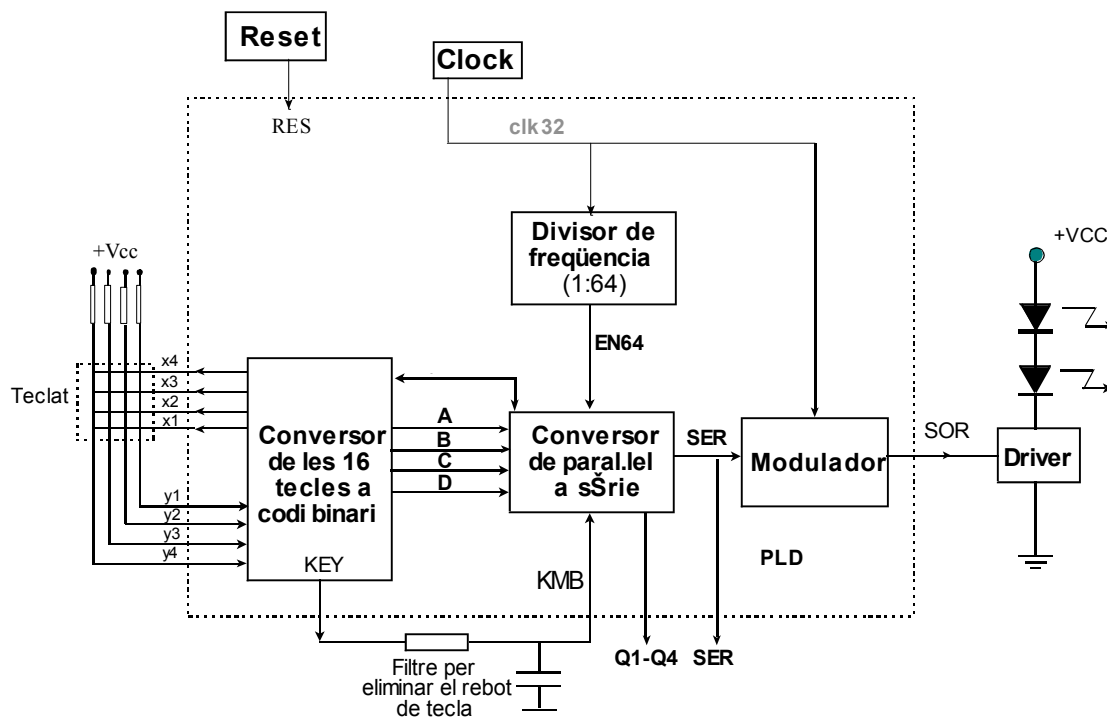


Fig. 5 : Esquema de blocs del comandament a distància

4.3.1 Reset i Oscil·lador

Tant el rellotge com el circuit de *reset* són externs. El rellotge és generat mitjançant un cristall de quars realimentat a través d'una xarxa RC proveïda d'un inversor, i fixa el període d'oscil·lació en 32.768Hz. S'afegeix un altre inversor a la sortida per evitar efectes de càrrega. La freqüència d'aquest rellotge (clk32) és la de modulació (32.768Hz).

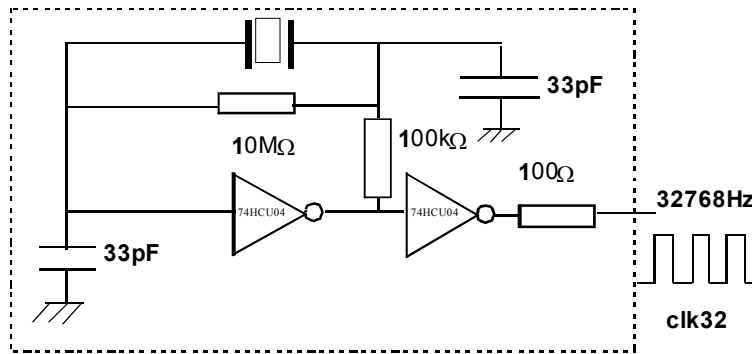


Fig. 6 : Oscil·lador

Sortida: CLK32

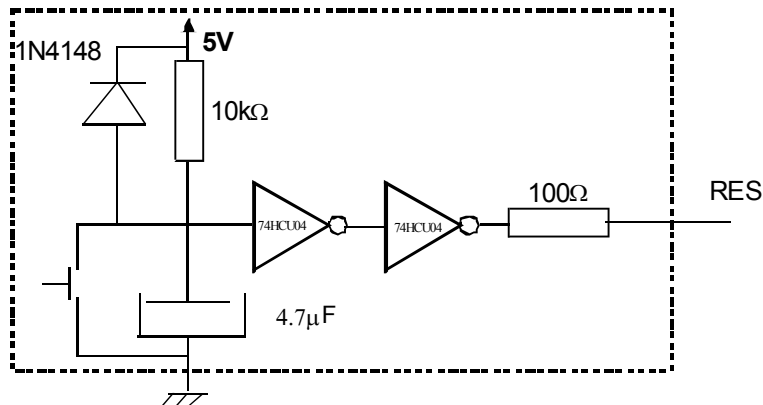


Fig. 7 : Circuit de reset

Sortida: RES

Aquest senyal de *reset* està connectat al *pin* de Reset de la PLD i per tant en prémer el polsador s'inicialitza de manera asíncrona tota la PLD, incloent-hi òbviament els biestables del nostre disseny . Tanmateix en la simulació del disseny no cal inicialitzar els biestables, atès que el valor que agafen per defecte tots els biestables del disseny quan es fa la inicialització és '0'. És per aquest motiu que no heu d'incloure l'entrada *reset* en cap disseny i que els següents esquemes no l'esmenten, a diferència del senyal de rellotge, que sí que l'heu d'incloure i que els següents esquemes l'esmenten.

4.3.2 Divisor de freqüència (÷ 64)

Aquest bloc generarà un senyal (EN64) que farà d'habilitador de part dels biestables del sistema. Aquest senyal valdrà '1' només un de cada 64 cicles del rellotge clk32. Quan sigui igual a '1', indicarà a una part dels biestables del sistema que poden canviar el seu valor. Així s'aconsegueix dividir la freqüència de funcionament de part del sistema per 64 mentre la resta funciona amb el rellotge principal a 32.768Hz, deixant que el rellotge principal (clk32) estigui connectat a les entrades de rellotge de tots els biestables. És a dir, tindrem un disseny síncron.

L'esquema de blocs del divisor es pot veure a la següent figura:

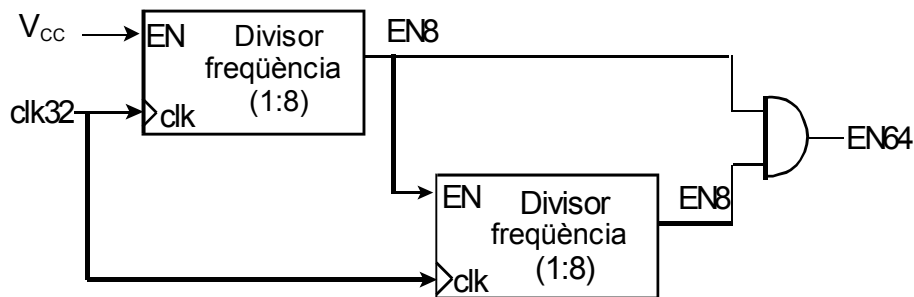


Fig. 8 : Esquema del divisor de freqüència per 64

El divisor de freqüència per 8 el farem a partir d'un comptador de 3 bits cíclic (0-7). El senyal EN8 haurà de ser igual a '1' quan el comptador tingui el valor 7.

Tots dos comptadors només podran incrementar el seu valor quan els seus respectius senyals d'entrada EN siguin igual a '1'. Aquest senyal és d'habilitació.

El senyal EN64 serà el producte lògic dels dos senyals EN8, i només valdrà '1' un de cada 64 cicles de rellotge.

Entrades: clk32

Sortida: EN64

4.3.3 Teclat

Consta de 16 tecles i 8 terminals accessibles, 4 corresponents a les files (X1-X4), i uns altres 4 corresponents a les columnes (Y1-Y4).

4.3.4 Convertidor 16-tecles a binari

Aquest bloc genera la identificació de la tecla polsada en els quatre bits de sortida ABCD. Mitjançant el senyal KEY s'indicarà a la resta del sistema si s'està polsant alguna tecla.

Aquest bloc anirà escombrant les files del teclat mitjançant el comptador de 2 bits i el descodificador 2 a 4. El comptador treballarà a 32.768Hz i s'haurà d'incrementar sempre que el seu senyal d'entrada EN sigui igual a '1'.

Els senyals Y1-Y4 que provenen del teclat passaran a un codificador amb prioritat, 4 a 2, que ens donarà la codificació en binari de la columna de la tecla que estigui pulsada. La combinació dels 4 bits ABCD ens dona la codificació completa de la tecla que s'ha pulsat (AB: fila, CD: columna). La indicació que efectivament hi ha una tecla pulsada la donarem amb el senyal KEY, que llavors haurà de prendre el valor '1'. Aquest senyal aturarà el comptador de 2 bits.

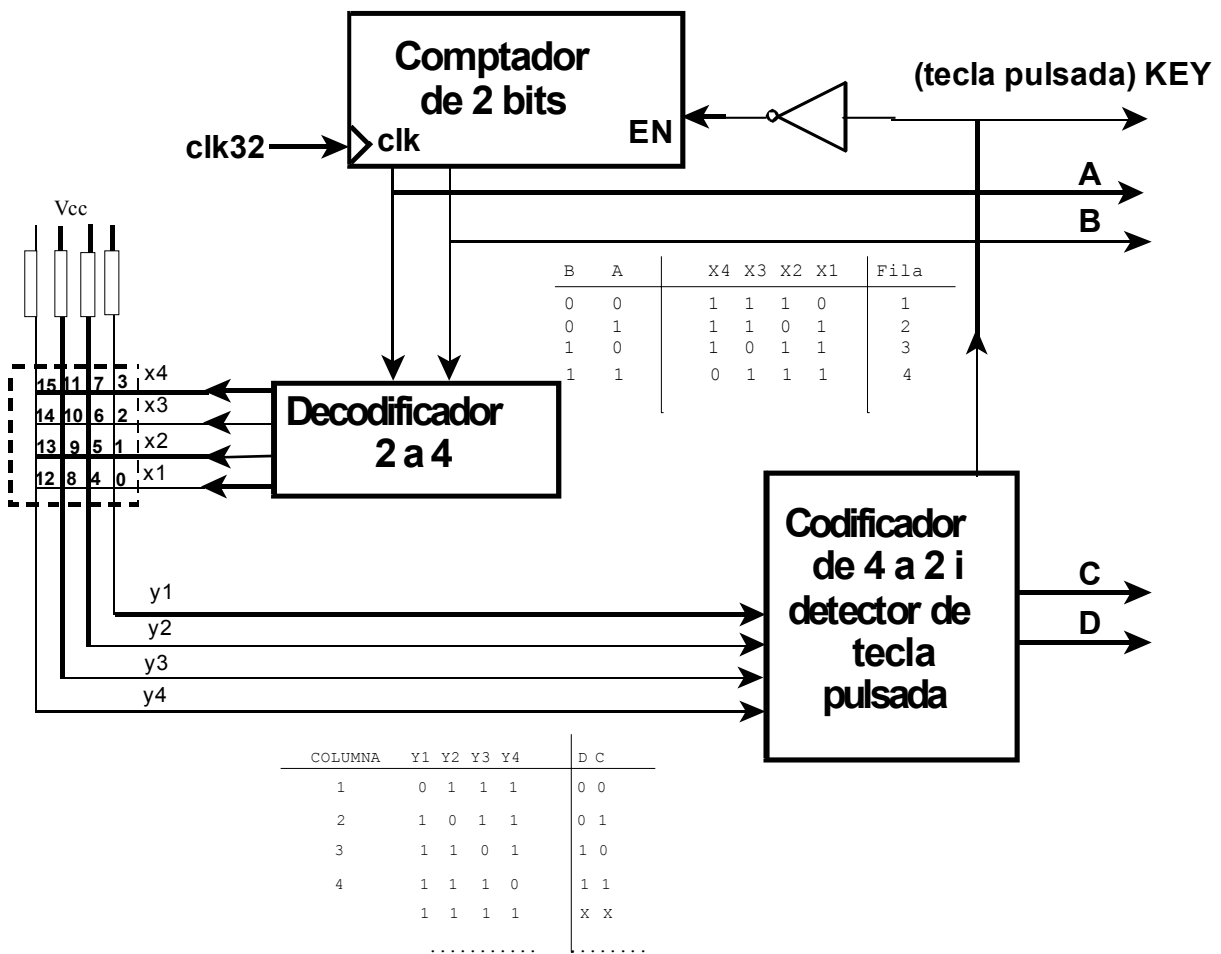


Fig. 9 : Esquema del convertidor de 16 tecles a binari

Entrades: clk32,

Sortides: KEY, A, B, C, D

4.3.5 Convertidor Paral·lel - Sèrie:

La part central d'aquest bloc estarà constituïda per un registre de desplaçament que convertirà les dades amb format paral·lel a format sèrie. Farem una càrrega paral·lela del registre sempre que haguem detectat una tecla pulsada, tenint en compte que no es poden sobreposar emissions successives de dades i que mentre es mantingui la tecla

polsada durant un cert temps haurem de repetir la dada enviada (com si s'hagués polsat varies vegades la tecla). L'esquema proposat es pot veure a la següent figura (on KMB és el senyal KEY un cop filtrat externament amb una xarxa RC per evitar falses deteccions degudes als rebots de les tecles).

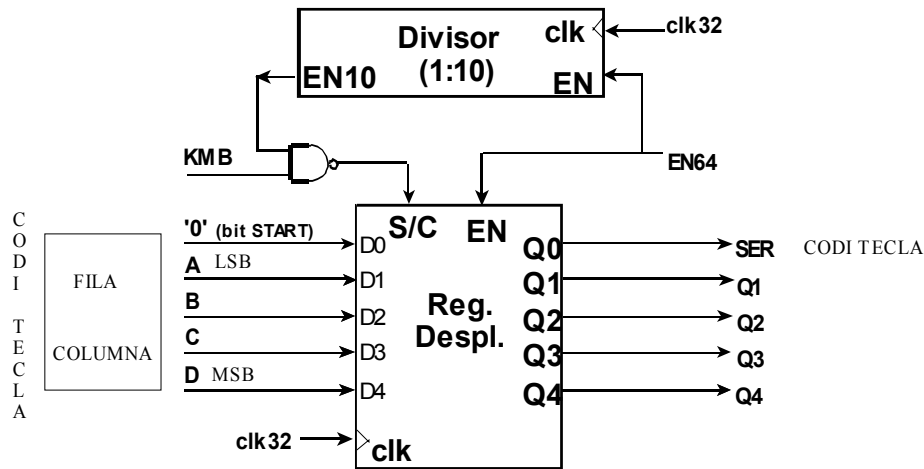


Fig. 10 : Convertidor paral·lel - sèrie

A la taula següent es pot veure la taula de veritat del registre de desplaçament.

CLK	EN	S/C	$D_i=D_4...D_0$	Q_4	$Q_i=Q_3..Q_0$
0 ó 1	X	X	X	Q_4	Q_i
↑	0	X	X	Q_4	Q_i
↑	1	1	X	1	Q_{i+1}
↑	1	0	D_i	D_4	D_i

Taula 1: Taula de veritat del registre de desplaçament

Les sortides Q1-Q4 només s'utilitzaran per verificar el bon funcionament del sistema i seran sortides de la PLD només per aquest motiu.

Entrades: clk32, KMB, EN64, A,B,C,D

Sortides: SER,Q1,Q2,Q3,Q4

El divisor de freqüència per 10 el farem a partir d'un comptador de 4 bits que haurà d'anar comptant des de 0 fins a 9 cíclicament. Quan el valor del comptador sigui 9 posarem el senyal de sortida EN10 a '1'. D'aquesta manera aquest senyal és igual a '1' un de cada 10 cicles de rellotge

4.3.6 Modulador

Aquest bloc haurà de modular el senyal SER a 32.768Hz. La sortida, SOR, serà lliurada directament a l'emissor d'infrarojos.

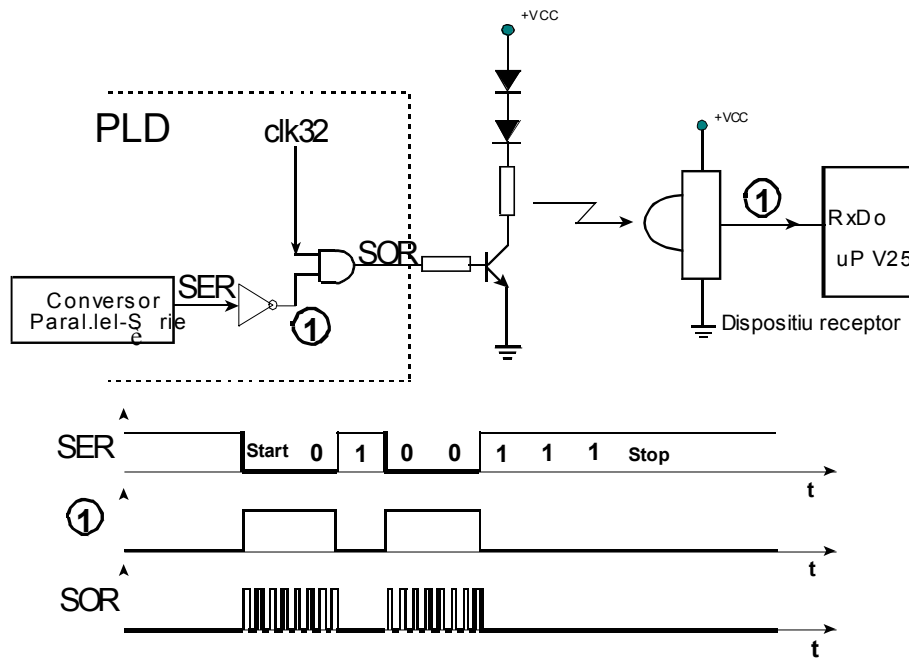


Fig. 11 : Esquema del modulador

Entrades: clk32, SER

Sortida: SOR

4.4 Estudi previ

Q1: Feu una taula amb les codificacions binàries que farà el comandament a distància per cada tecla. És a dir, considerant que el bit de menor pes dels 7 bits de dades que s'envien és el corresponent al bit 'A' (fig. 4), identifiqueu els codis que rebrà la UART del microcontrolador per cadascuna de les tecles (veure fig. 10 i fig. 11).

Q2: Dissenyau els següents blocs de l'esquema de blocs del comandament a distància, proposats en el punt 4.3 (veure fig. 5):

- Divisor de freqüència ($\div 64$)
- Convertidor de 16 tecles a binari
 - Comptador de 2 bits
 - Descodificador de 2a4
 - Codificador de 4a2 i codificador de tecla polsada (KEY = 1)
- Convertidor paral·lel - sèrie
 - Divisor de freqüència ($\div 10$)
 - Registre de desplaçament
- Modulador

Q3: Tenint en compte els diferents tipus de configuració de les sortides de la PLD (drenador obert, *push-pull*) indiqueu quines característiques elèctriques hauran de tenir els senyals Xi a la sortida de la PLD pel bon funcionament del teclat. Què pot passar si les sortides Xi són configurades com a sortides *push-pull* ?

Q4: Quin(s) efecte(s) provocaria el que no s'utilitzés el senyal KEY a l'esquema de la figura 8 per parar el comptador de dos bits ?

Q5: A partir de l'esquema proposat en els punts anteriors, determineu el temps mínim que transcorre entre dues emissions successives de caràcters per part del comandament a distància.

Q6: Quin(s) efecte(s) provocaria que no es fes servir el divisor de freqüència per 10 a l'esquema de la figura 10, suposant que es connecta directament el senyal KMB a l'entrada S/C del registre de desplaçament ?

Q7: Si es manté una tecla polsada durant 10 segons, quants caràcters hauran estat enviats a la UART del microcontrolador ?

Q8: Quin seria el temps de bit si a l'esquema de la figura 5 canviéssim el divisor de freqüència per 64 per un altre divisor per 120 (és a dir si en comptes de tenir un senyal EN64 tinguéssim un senyal EN120 a tot el sistema que fos igual a '1', un de cada 120 cicles de rellotge) ? Es veurien afectats els temps calculats a les qüestions Q5 i Q7 ? En cas afirmatiu calculeu-los.

4.5 Entrada i simulació del sistema dissenyat

A la pràctica 3 s'ha introduït l'ús del programa de CAD ispLEVER, i s'ha utilitzat el Manual d'ús d'aquesta eina. Es tracta ara d'entrar, sigui en HDL, sigui l'esquemàtic, cadascun dels blocs anteriors i simular-lo. Després de comprovar el bon comportament de cadascun dels blocs, es crea un nou projecte que ha d'incloure'ls a tots, important els símbols de cadascun dels blocs, per finalment simular el funcionament global de sistema dissenyat.

Cadascun dels blocs per separat com el sistema global ha de satisfer totes les especificacions anteriorment anunciades. Comproveu cadascuna d'aquestes especificacions tot triant els vectors de test més convenients, i no passeu a una nova simulació fins haver completat l'anterior. El sistema global tan sols funcionarà si ho fan cadascun dels blocs.

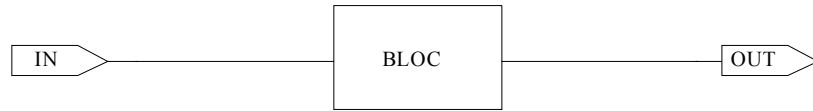
És obligatori fer un disseny modular, és a dir, un disseny estructurat en blocs de funcionalitat simple que estiguin integrats els uns a dins dels altres segons una jerarquia (per exemple, el divisor de freqüència integra dos divisors de freqüència, el convertidor de les 16 tecles a codi binari integra un comptador de 2 bits, i aquest 2 comptadors e 1 bit, etc.).

El disseny final s'ha de correspondre a l'esquema de la fig. 5 de manera que a cadascun dels 4 blocs li correspongui un símbol de la llibreria local de ispLEVER, creat per vosaltres. Juntament amb aquests 4 noms de símbols, a la llibreria local hi ha d'aparèixer tots els altres noms de símbols corresponents a blocs jeràrquicament inferiors, creats per vosaltres, i importats al disseny final.

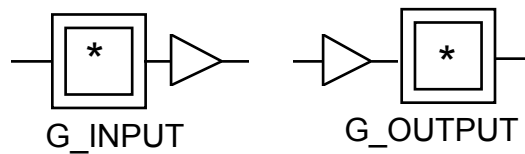
4.5.1. Assignació de pins del circuit integrat

Un cop validat el disseny, cal assignar les entrades/sortides del disseny a pins físics del circuit integrat ispLSI2032, marcats de color negra a la figura 12.

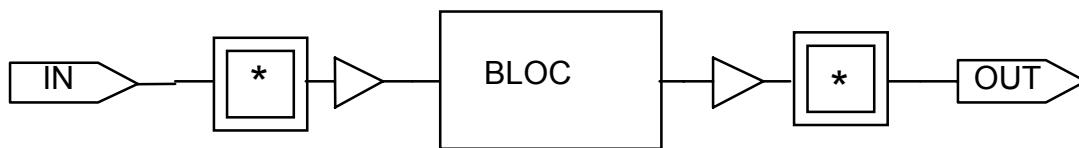
Per a això s'utilitzen els símbols de la llibreria IOPADS. En concret, si tenim un disseny com l'indicat a la següent figura:



per assignar els pins s'insereixen els elements de llibreria C_INPUT i G_OUTPUT



a les entrades i sortides tal com es mostra en el següent circuit:



D'aquesta forma s'indica al programa que els senyals IN i OUT s'han d'assignar a pins.

L'integrat ispLSI 2032 és un dispositiu programable en mode sèrie, sense desconnectar-lo de la resta de circuits on doni servei (isp: *in system programmable*), amb 2 pins de només entrada, 3 entrades de rellotge, un pin d'habilitació de les sortides i 32 pins d'entrada/sortida, que hem de fer correspondre a les entrades i sortides del circuit que acabem de senyalar.

Després de senyalar les entrades i les sortides del circuit, dins de cada *pad* dibuixat hi ha un asterisc (*). Això indica que ispLEVER pot escollir qualsevol pin disponible del ispLSI per assignar aquests senyals d'entrada i de sortida. Per fixar un pin d'entrada o de sortida, cal canviar l'asterisc per un número de pin. Per a això s'usa ADD SYMBOL ATTRIBUTE sobre el símbol de *pad* d'entrada o de sortida i es canvia l'atribut SYNARIO PIN que per defecte és (*). Les assignacions són les indicades a la figura 12.

Els pins marcats de color negra corresponen a les entrades i sortides del sistema dissenyat. Els pins marcats en gris corresponen a les alimentacions i als pins de programació del dispositiu. Finalment, els pins en blanc són entrades/sortides no emprades.

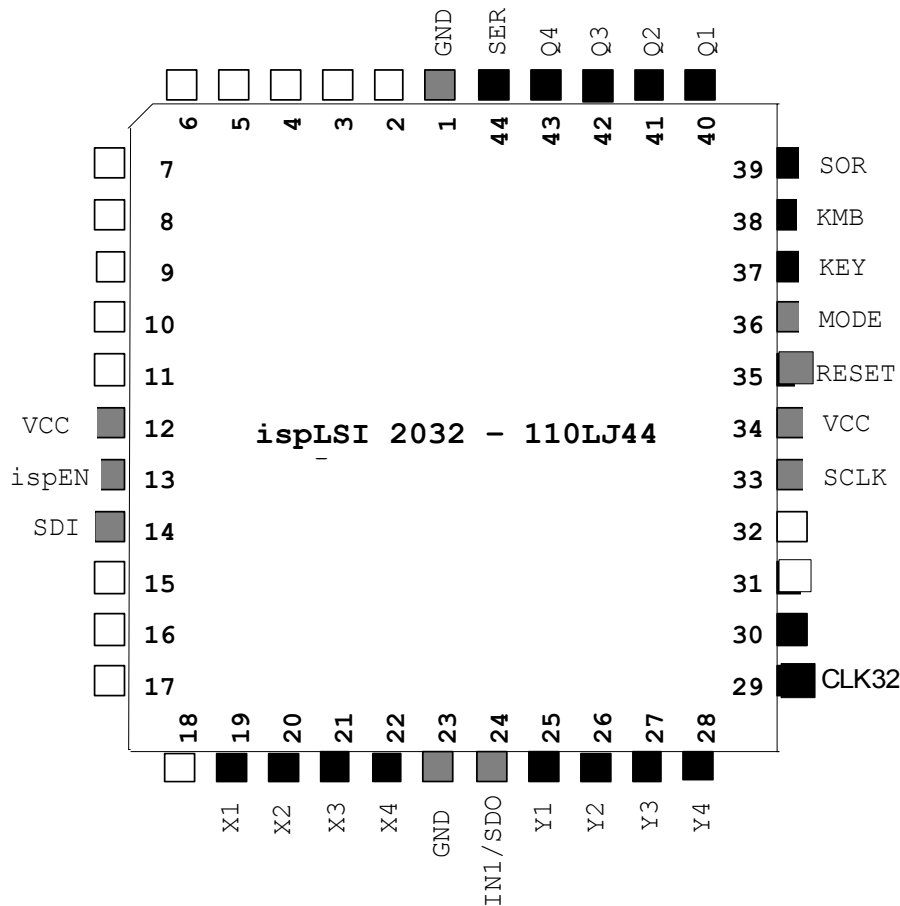


Fig. 12: Assignació de pins

NOTA:

Les propietats que poden assignar-se al procés de *fitting* poden fer que ispLEVER optimi l'ús del dispositiu sense fer cas dels pins assignats. Si això succeeix caldrà canviar les propietats corresponents.

4.5.2. Generació del fitxer JEDEC

Aquest és el darrer pas abans de gravar el circuit integrat ispLSI amb el contingut del disseny. El dispositiu que es fa servir és el ispLSI 2032-110 amb encapsat PLCC44 de Lattice. Les instruccions estan detallades en el Manual d'ús de ispLEVER. Tingueu present que l'arxiu JEDEC inclou els vectors de test del sistema global, amb els que el gravador realitzarà un test funcional del dispositiu. Certament cal que la simulació que haguéssiu realitzat prèviament hagués estat del tot satisfactòria, per tal que el dispositiu que graveu funcioni correctament i realitzi la tasca per a la qual està essent gravat.

4.6 Gravació del sistema dissenyat

El programa ispVM permet la gravació sense treure el circuit integrat del seu sòcol, mitjançant la connexió de cinc pins del dispositiu al port paral·lel de l'ordinador.

Verifiqueu que la vostra placa està alimentada i executeu el programa ispVM. A continuació seguiu les següents passes:

Selecció del dispositiu

La manera més senzilla de seleccionar el dispositiu és fer que el programa el detecti automàticament. Per a això executeu *scan chain* al menú ispTools.

Sortirà una finestra amb el tipus de dispositiu detectat (2032) i l'operació a realitzar: *erase&program&verify*. Tot clicant 2 cops apareix una finestra on mitjançant el botó *browse* podrem seleccionar el nostre arxiu JEDEC.

Gravació del dispositiu

Per gravar el dispositiu tan sols cal seleccionar GO dins la barra d'eines. Si el programa retorna *successful* vol dir que la gravació ha reeixit.

4.7 Verificació del sistema dissenyat

Per comprovar el funcionament del dispositiu que acabem de gravar s'utilitza un analitzador lògic. L'instrument 54622D tant fa les funcions d'oscil·loscopi com d'analitzador lògic de 16 canals. L'analitzador lògic permet capturar formes d'ona digitals i visualitzar-les, oferint diverses opcions a l'usuari per poder comprovar i depurar el funcionament de circuits lògics. Configureu l'analitzador i guardeu la configuració adequada per adquirir, representar i mesurar senyals.

4.7.1. Comprovació del funcionament del dispositiu

Un cop gravat l'integrat, cal comprovar que físicament funciona de la forma esperada. Aquesta comprovació es realitza connectant a la sortida del dispositiu tants canals com senyals vulguem analitzar. És convenient capturar les pantalles de l'analitzador amb les mateixes formes d'ona que s'havien prèviament simulat, per comparar els resultats de la simulació amb els de la verificació. Per dur a terme aquesta comparació de forma senzilla és convenient que les etiquetes dels senyals mesurats hagin estat ordenades a la pantalla de l'analitzador seguint l'ordre de les gràfiques obtingudes a les simulacions. La placa de proves disposa de connectors on caldrà col·locar les pinces de l'analitzador lògic.

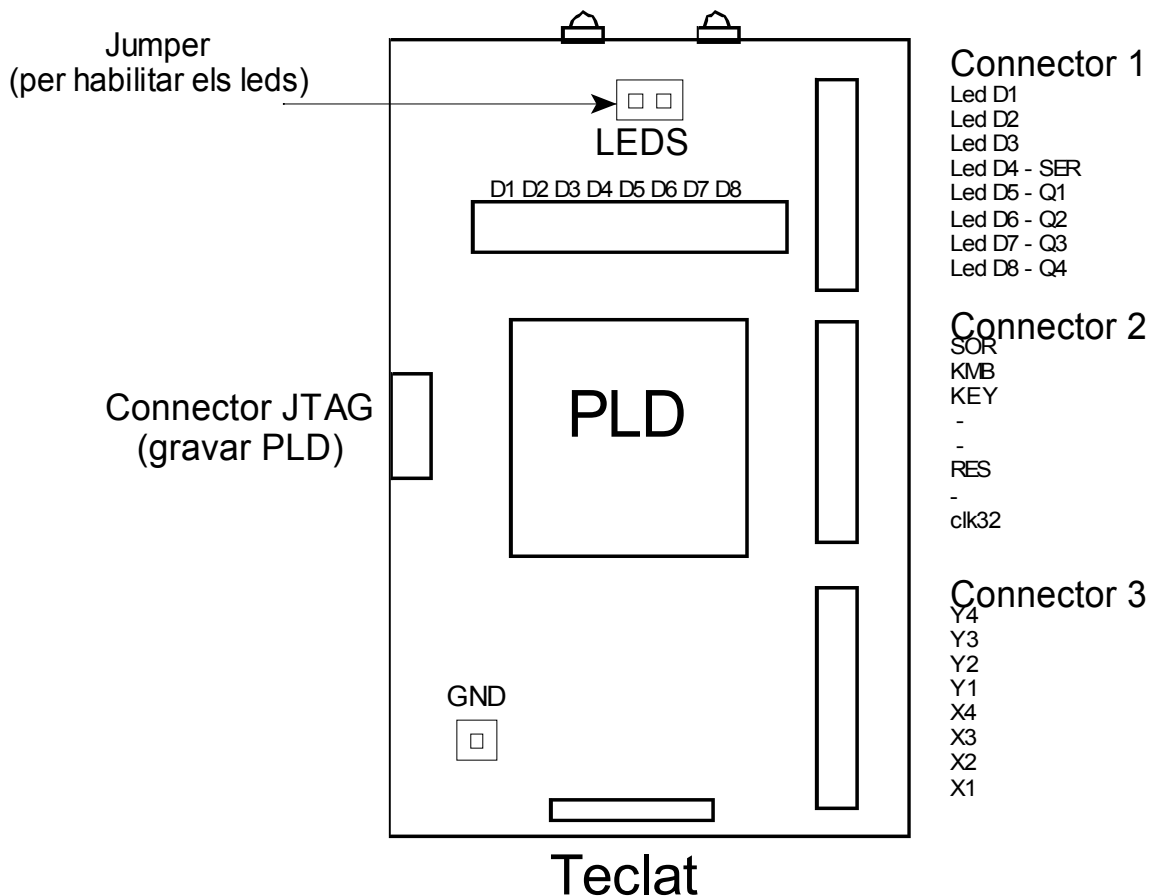


Fig. 13: Esquema de blocs de la placa de comandament a distància

Posarem com a condició del disparador (*trigger*) de l'analitzador el flanc de baixada del senyal SER (indicarà començament de transmissió) i observarem tots els senyals (registre de desplaçament, sortida modulada, escombrat teclat, etc.). Provarem també d'altres condicions, en particular es realitzaran les següents mesures:

M1 Mesura la freqüència del senyal de rellotge.

M2 Adquireix el senyal corresponent a cadascuna de les 12 tecles: obrir i tancar la pinça, pujar i baixar el braç, enrera i endavant, gir a l'esquerra i a la dreta, enregistrar, reproduir, esborrar i inici.

Utilitzant el programa *scope control sample application* podrem comunicar-nos a través del port sèrie de l'ordinador amb l'analitzador. Per capturar pantalles fixeiu les mateixes condicions de transmissió al 54622D (Utility->I/O->p.e.[Controller:RS232,Baud:57600, DTR] que a l'aplicació (Instrument->Set I/O).