

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2016-2017

Pràctica 1 – Fase 2

LSHangman

Alumnes	Login	Nom
	Ls30759	Samuel Tavares da Silva
	Ls30652	Gabriel Cammany Ruiz

Entrega	Placa	Memòria	Nota

Data	13/01/2016
-------------	-------------------

Portada de la memòria

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2016-2017

Pràctica 1 – Fase 2

LSHangman

Alumnes	Login	Nom
	Ls30759	Samuel Tavares da Silva
	Ls30652	Gabriel Cammany Ruiz

Entrega	Placa	Memòria	Nota

Data	13/01/2016
-------------	-------------------

Portada de l'alumne

Índex

1. Síntesi de l'enunciat	4
2. Plantejament.....	5
3. Diagrama de mòduls	6
4. Disseny.....	8
4.1. Disseny del SSPE.....	8
4.1.1. Unitat de control	8
4.1.2. Unitat de procés	11
4.2. Implementació del SSS	12
4.3 Fitxers .pld de les GAL's	12
5. Esquema elèctric	20
6. Problemes observats.....	26
7. Conclusions.....	27
8. Planificació	28
9. Annex	28

1. Síntesi de l'enunciat

Es requereix la implementació del LSHangman, també conegut com el clàssic model de joc el penjat, aquest només contindrà paraules de com a màxim 8 lletres, per lo que s'ha decidit que l'usuari per poder guanyar la partida tingui que completar una paraula sencera sense cometre més de 8 intents.

El primer sistema es basa en la selecció d'una lletra, pe tant per poder comunicar-se amb el sistema s'utilitzarà un teclat matriu de mida 3x3 el qual simularà un teclat alfanumèric, típicament utilitzat en els mòbils. El sistema controlarà el temps que l'usuari es mantingui amb la mateixa tecla un temps superior o igual a 2 segons, cal destacar que cada lletra s'haurà de mostrar per un display de 16 segments.

El segon sistema seleccionarà aleatòriament una paraula, dins un conjunt de com a màxim 255 paraules, seguidament indicarà mitjançant una sèrie de 8 leds quines lletres l'usuari ha encertat, també s'utilitzarà un 16 segments amb la mateixa finalitat però mostrar les lletres que l'usuari ha endevinat, cal destacar que aquest sistema també és l'encarregat de controlar les vides de l'usuari, en cas que aquest superi els 8 intents s'il·luminarà un led que indiqui el Game Over.

Finalment el tercer sistema mitjançant l'ús d'un microcontrolador i un cable VGA, mostrarem per pantalla el número d'errors que porta l'usuari, de manera que per cada error haurem d'afegir una part del ninot i quan es completi el ninot de Hangman indicarà que s'ha acabat el joc.

2. Plantejament

Tenint una pràctica gran com aquesta es indispensable dividir la feina en diverses fases, sent aquesta l'encarregada de portar a terme la lògica del joc.

En primer lloc, tenim desades totes les 256 paraules en la EEPROM. Les quals es seleccionaran aleatòriament tot just començar la partida, es a dir, al prémer el PCI aquest activarà en enable dels comptadors i aquest comptaran en un ritme de 1Khz.

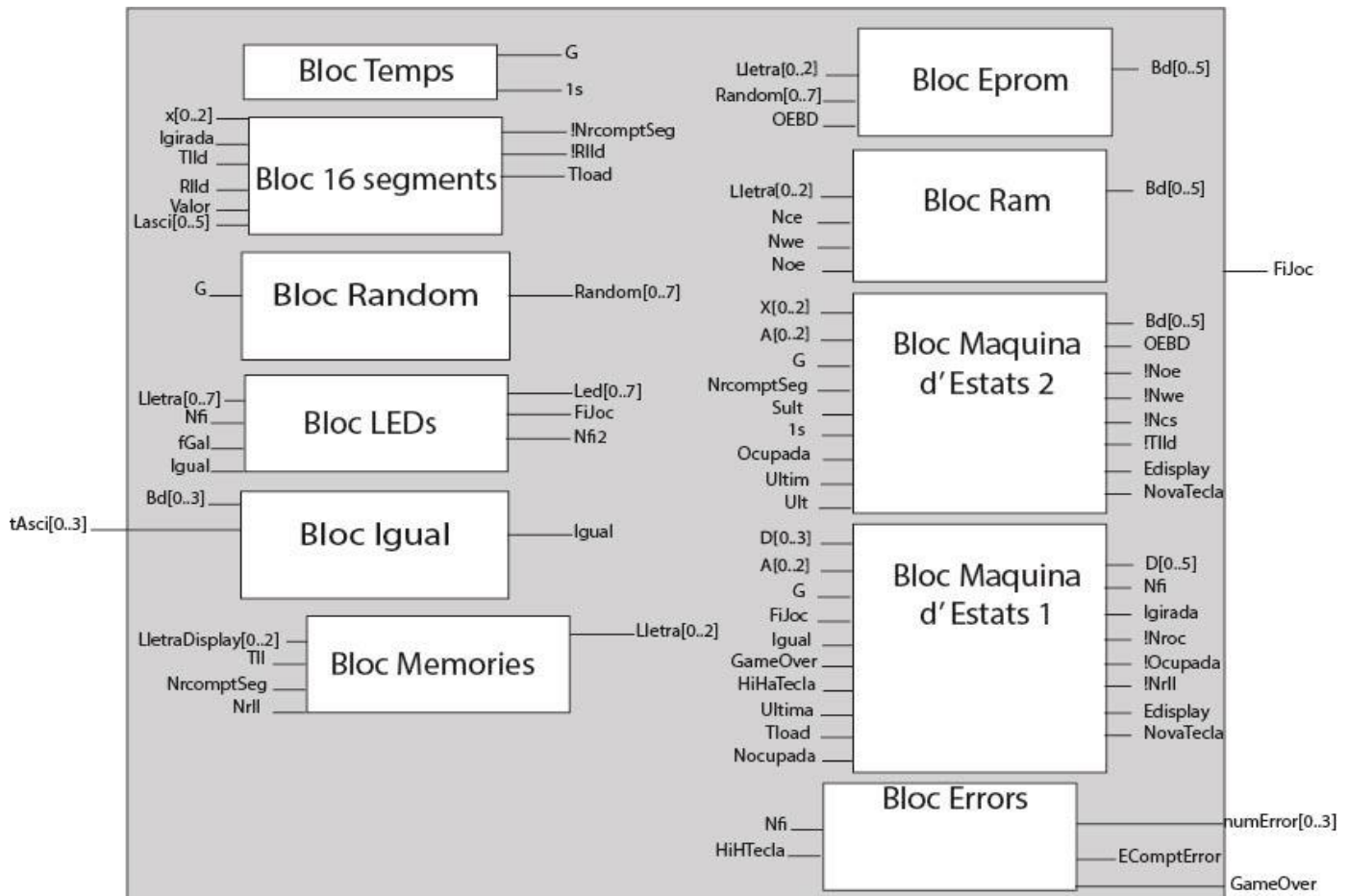
Per un altre banda, després de seleccionar el aleatori, s'escriurà en la RAM totes les lletres que anirem a utilitzar amb dos bits extrems. El bit de lletra girada i el bit de ultima girada. Aquests dos bits ens serviran per controlar en quina posicio estem de la paraula i a la vegada saber si s'ha de mostrar o no per el 16 segments la lletra corresponent.

Així a partir de la ram i la EEPROM, podem realitzar la lògica del joc juntament amb comptadors, biestables i registres. Aquests, s'encarregaran de juntament amb les dos memòries de passar la informació a les diferents gals.

Per tant, per mostrar les diferents lletres s'utilitzarà el 16 segments amb 2 gals, que convertiran el valor ascii de cada lletra a 16 segments. Cada una s'encarregarà d'una diferent mitat.

Finalment per realitzar tota el control del hardware i per tant les dues maquines d'estats utilitzarem 2 gals més, que cada una tindrà una maquina diferent. La maquina principal es la encarregada de controlar la escriptura de la RAM per a totes les lletres i també fer les corresponents comprovacions de les lletres introduïdes per la fase 1, per un altre banda la maquina d'estats 2 esta exclusivament centrada en mostrar les lletres per el display.

3. Diagrama de mòduls



- Bloc Temps- Aquest es el que s'encarrega de saber quan temps sa de mostrar una lletra per el display i de generar un senyal que permeti treballar a tots els components del sistema a una mateixa velocitat.
- Bloc 16 segments- Aquest no sols conté el display de 16 segments que ens mostrarà la lletra contínuament sinó que realitzarà la conversió de la lletra en valor ascii a 16 segments mitjançant dos gals.
- Bloc Random- Es el bloc encarregat de generar una direcció random per la eprom.
- Bloc Leds- Bloc encarregat de gestionar els leds que indicarà les lletres que encara no han sigut encertades i de generar alguns senyals per altres blocs.
- Bloc igual – Aquest es l'encarregat de comparar la lletra introduïda per l'usuari i la que es te guardada a la memòria Ram.
- Bloc control memòries- Es l'encarregat de realitzar el control de col·lisió amb la memoria ram i eprom.

- Bloc eprom- Format per la memòria eprom, es l'encarregat de emmagatzemar les paraules que tindrà el nostre sistema, ja que en funció d'una direcció aleatòria ens retornarà una paraula de com a màxim 8 lletres.
- Bloc ram- Bloc encarregat d'emmagatzemar una paraula amb l'estat en que es troba cada lletra per tal de que el display funcioni correctament.
- Bloc Maquina d'estats 2- Bloc encarregat de la maquina d'estats principal, per lo que es el SSS principal, gracies al qual aconseguim que el hardware del nostre sistema funcioni sincronament i de manera correcta en funció del nostre disseny.
- Bloc Maquina d'estats 1- Bloc encarregat de la maquina d'estats del display, per lo que en funció de l'estat de la ram, llegirà cada lletra i la mostrarà per el display.
- Bloc Errors- Bloc encarregat de gestionar els comptatge dels errors realitzats per l'usuari, generant el senyal de GameOver.

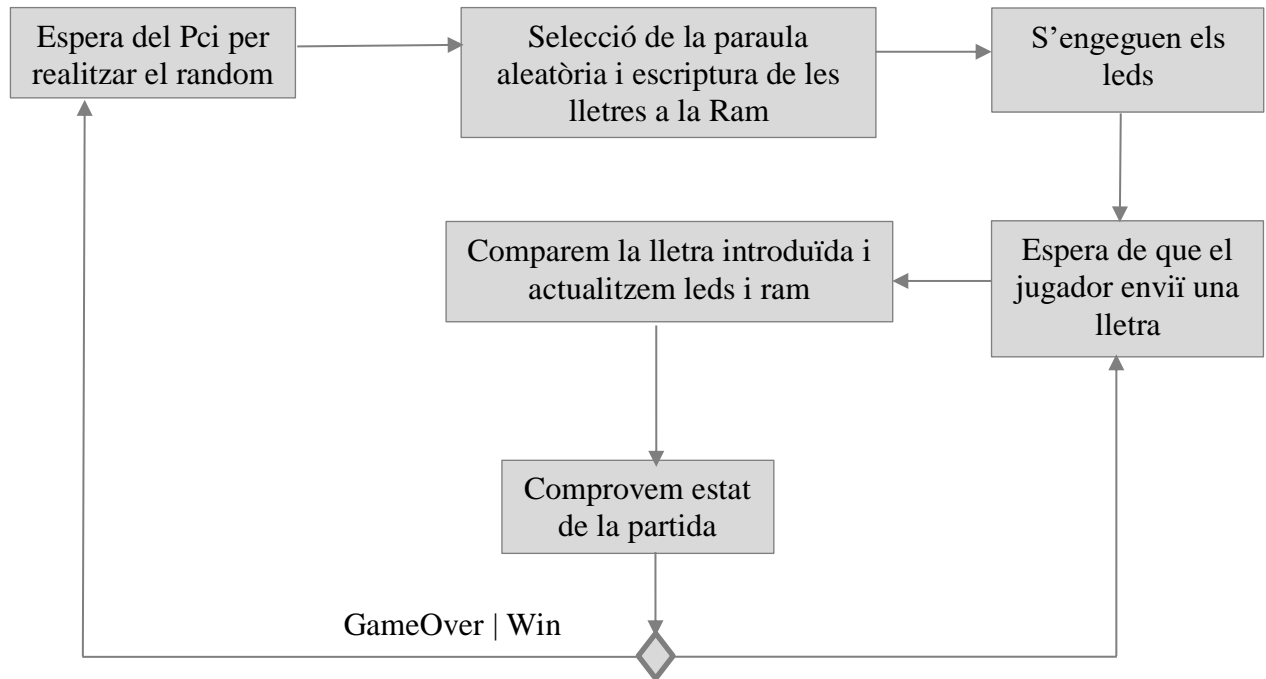
4. Disseny

4.1. Disseny del SSPE

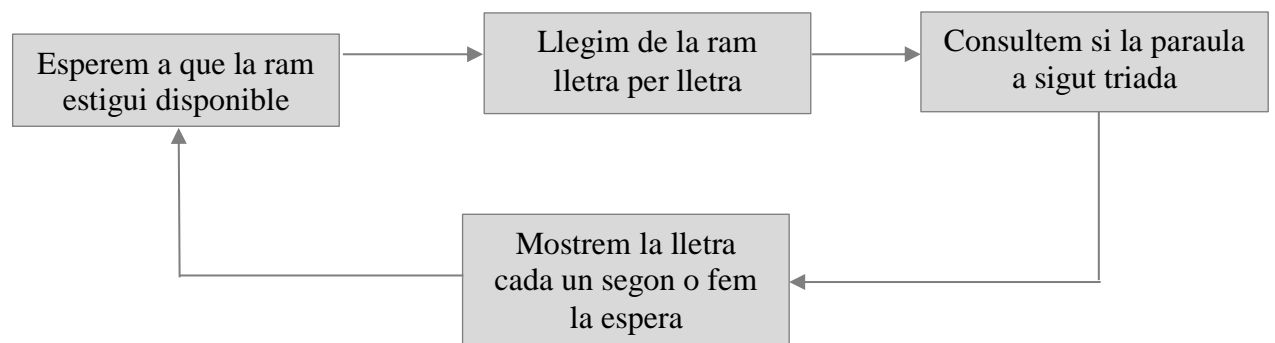
4.1.1. Unitat de control

Disseny de la unitat de sincronització

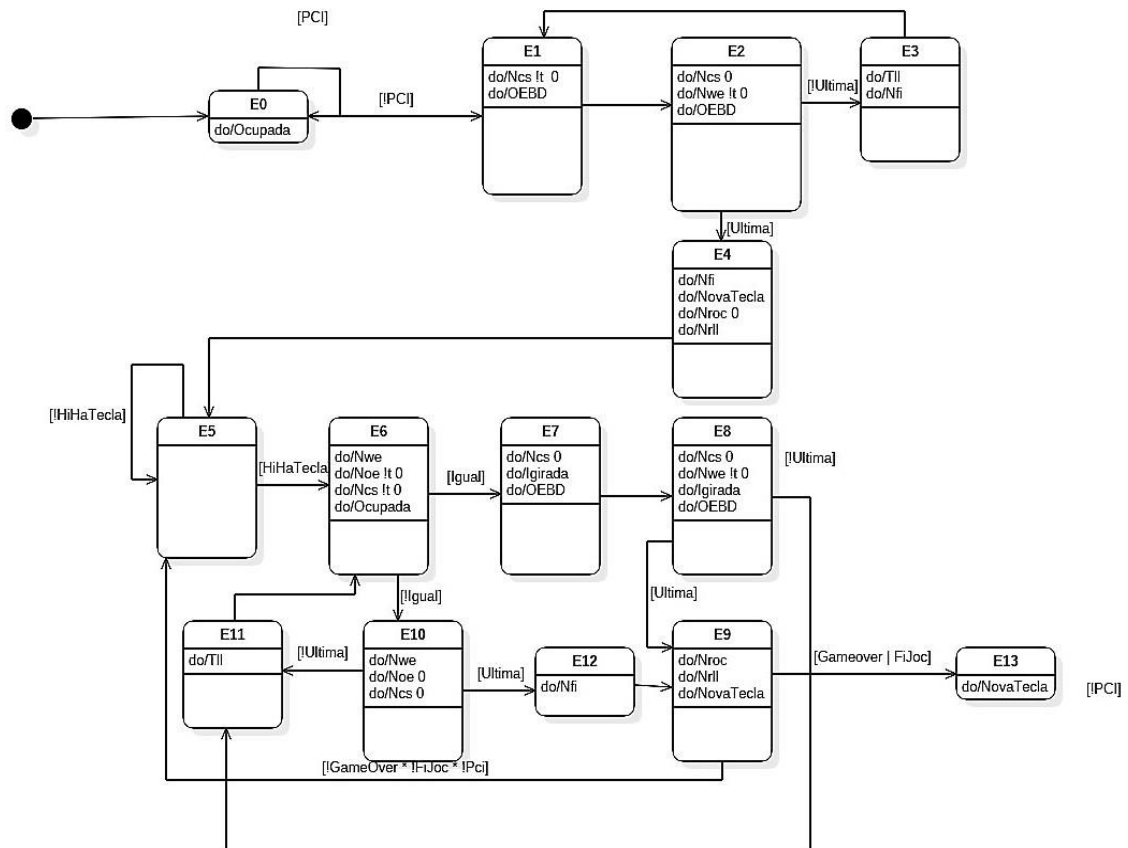
1ra unitat de sincronització



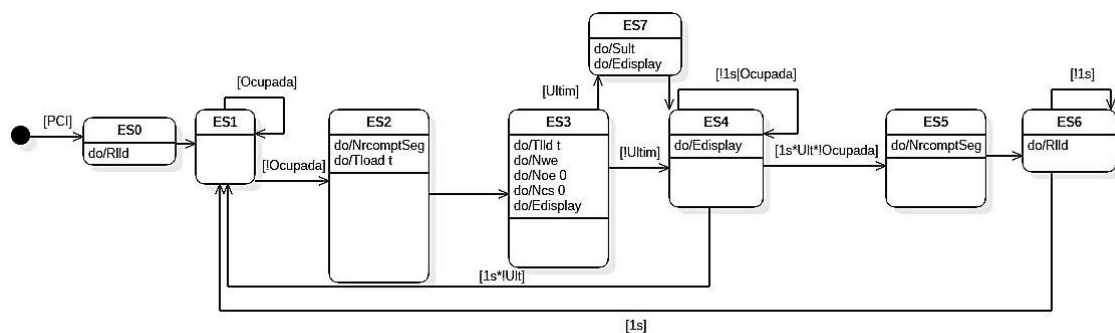
2na unitat de sincronització



Disseny del SSS Maquina d'estats principal



Maquina d'estats del display



Disseny de la interfície

Les equacions pertinents a la maquina d'estats principal:

```
estat0 = (e0 & PCI);
estat1 = ((e0 & !PCI) # e3);
estat2 = (e1);
estat3 = (e2 & !Ulti);
estat4 = (e2 & Ulti);
estat5 = (e5 & !HHT # (e4) # (e9 & !GameOver & !FiJoc & !PCI));
estat6 = (e5 & HHT) # (e11);
estat7 = (e6 & I);
estat8 = (e7);
estat9 = (e8 & Ulti) # (e12);
estat10 = (e6 & !I);
estat11 = (e10 & !Ulti) # (e8 & !Ulti);
estat12 = (e10 & Ulti);
estat13 = ((e9 & (GameOver # FiJoc)) # (e13 & !PCI));

D0 = (estat1 # estat3 # estat5 # estat7 # estat9 # estat11 # estat13) & !estat0;
D1 = (estat2 # estat3 # estat6 # estat7 # estat10 # estat11) & !estat0;
D2 = (estat4 # estat5 # estat6 # estat7 # estat12 # estat13) & !estat0;
D3 = (estat8 # estat9 # estat10 # estat11 # estat12 # estat13) & !estat0;
```

Les equacions pertinents a la maquina d'estats del display:

```
estat1 = (es0) # (es1 & Ocupada) # (es4 & 1s & !Ult) # (es6 & 1s) # (es2 & Ocupada) #
(es3 & Ocupada);
estat2 = (es1 & !Ocupada);
estat3 = (es2 & !Ocupada);
estat4 = (es3 & !Ultim & !Ocupada) # (es4 & (!1s # Ocupada)) # (es7);
estat5 = (es4 & 1s & Ult & !Ocupada);
estat6 = (es5 # (es6 & !1s));
estat7 = (es3 & Ultim & !Ocupada);

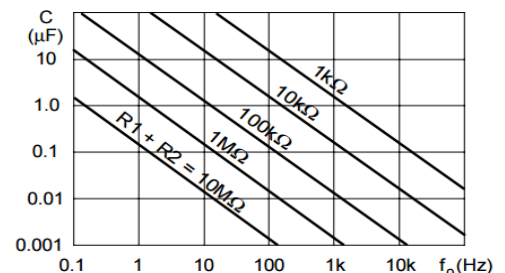
X0 = (estat1 # estat3 # estat5 # estat7);
X1 = (estat2 # estat3 # estat6 # estat7);
X2 = (estat4 # estat5 # estat6 # estat7);
```

4.1.2. Unitat de procés

Alu (Unitat Aritmètico-Lògica)

A continuació llistarem i explicarem cada xip que es considera dins de la unitat lògica del sistema:

- Ne555 - Xip encarregat de generar un senyal quadrat d'una freqüència d'1kHz amb un Duty Cycle del 50% (± 2), per tal de generar aquest senyal cal realitzar certs passos per tal de configurar el xip en mode estable, aquesta configuració es realitza a partir d'una gràfica i unes formules que ens donen les resistències necessàries i la capacitat dels condensadors que hem de posar. Tal com es mostra en les següents imatges:



$$f = \frac{1}{T} = \frac{1.44}{(R_1 + 2R_2) C_1} \quad D = \frac{R_2}{R_1 + 2R_2}$$

- 74LS14 - Xip anomenat Trigger Smith que actua com una porta not amb una característica especial, ja que quan el senyal es troba en el rang de 1.5V – 2V tel estabilitza a un '1' lògic (5V) de manera que si col·loquem dos triggers Smith aconseguim generar un senyal quadrat perfecte encara quan el senyal d'entrada no ho es. Aquest xip sol ser utilitzat especialment per els senyals generats per polsadors en el nostre cas pci.
- 74ls85 → Xip comparador de 4 bits amb una única sortida per lògica negativa que ens indicarà si A i B son iguals, aquest concretament ens permet utilitzar un output enable, en canvi aquest no inclou pull up, però en el nostre cas no ens es necessari, però al necessitar comparar 8 bits concatenem dos comparadors.

Unitat de memorització

Hem utilitzat una sèrie de memòries per tal de estabilitzar la informació del sistema, aquestes han set:

- 74LS74 → Xip que conté dos biestables de tecnologia TTL, es una unitat de memòria simple que ens permet emmagatzemar un sol bit i admet un sincronisme. Aquest xip te la peculiaritat de tenir les senyals de set i reset, en aquest cas per lògica negativa, però permeten un us asíncron.
- 74LS173 i 74LS169 → Xip que conte un registre de 4 entrades en paral·lel i 4 sortides paral·lel, pertany a la tecnologia TTL, aquest actua com quatre biestables junts ja que emmagatzema 4 bits i permet la utilització d'un sincronisme.
- 25C256 → Memòria Eprom, la qual utilitza tecnologia CMOS, es l'encarregada de rebre una direcció aleatòria on es trobi una paraula en ascii. Un cop la memòria es esborrada mitjançant un feix de llum ultra violada durant uns 20 – 45 minuts, es programarà utilitzant el loprogram i un cop guardat el .hex el carregarem al xip mitjançant una programadora i un software anomenat ic writer.
- 62c256 → Memòria Ram, de tecnologia TTL en un encapsulat Pip, aquesta l'utilitzem per emmagatzemar la lletra seleccionada de manera aleatòria i l'estat en que es troba, entenen per estat si ha sigut encertada o no anteriorment.

4.2. Implementació del SSS

4.3 Fitxers .pld de les GAL's

Gal 1-Maquina d'estats principal

```
Name P1F2G1 ;
PartNo 00 ;
Date 23/11/2016 ;
Revision 01 ;
Designer ls30652_ls30759;
Company La Salle ;
Assembly None ;
Location Barcelona ;
Device g22v10 ;

/* ***** INPUT PINS ***** */
PIN 1 = t;
PIN 8 = Q0;
PIN 9 = Q1;
PIN 10 = Q2;
PIN 11 = Q3;
PIN 6 = PCI;
PIN 7 = HHT;
PIN 4 = I;
PIN 2 = Ulti;
PIN 5 = GameOver;
PIN 3 = Fijoc;
PIN 13 = Tload;

/* ***** OUTPUT PINS ***** */
PIN 17 = D3;
PIN 15 = D2;
PIN 16 = D1;
PIN 19 = D0;
PIN 14 = TII;
PIN 18 = !NrII;
PIN 20 = !Ocupada;
PIN 22 = Igirada;
PIN 21 = !Nroc;
PIN 23 = Nfi;

/* ***** EQUACIONS ***** */

e0 = !Q3 & !Q2 & !Q1 & !Q0;
e1 = !Q3 & !Q2 & !Q1 & Q0;
e2 = !Q3 & !Q2 & Q1 & !Q0;
e3 = !Q3 & !Q2 & Q1 & Q0;
e4 = !Q3 & Q2 & !Q1 & !Q0;
e5 = !Q3 & Q2 & !Q1 & Q0;
e6 = !Q3 & Q2 & Q1 & !Q0;
e7 = !Q3 & Q2 & Q1 & Q0;
e8 = Q3 & !Q2 & !Q1 & !Q0;
e9 = Q3 & !Q2 & !Q1 & Q0;
e10 = Q3 & !Q2 & Q1 & !Q0;
e11 = Q3 & !Q2 & Q1 & Q0;
e12 = Q3 & Q2 & !Q1 & !Q0;
e13 = Q3 & Q2 & !Q1 & Q0;

D0 = (estat1 # estat3 # estat5 # estat7 # estat9 # estat11 # estat13) & !estat0;
```

```
D1 = (estat2 # estat3 # estat6 # estat7 # estat10 # estat11)    & !estat0;
D2 = (estat4 # estat5 # estat6 # estat7 # estat12 # estat13)    & !estat0;
D3 = (estat8 # estat9 # estat10 # estat11 # estat12 # estat13)    & !estat0;

estat0 = (e0 & PCI);
estat1 = ((e0 & !PCI) # e3);
estat2 = (e1);
estat3 = (e2 & !Ulti);
estat4 = (e2 & Ulti);
estat5 = (e5 & !HHT # (e4) # (e9 & !GameOver & !FiJoc & !PCI));
estat6 = (e5 & HHT) # (e11);
estat7 = (e6 & I);
estat8 = (e7);
estat9 = (e8 & Ulti) # (e12);
estat10 = (e6 & !I);
estat11 = (e10 & !Ulti) # (e8 & !Ulti);
estat12 = (e10 & Ulti);
estat13 = ((e9 & (GameOver # FiJoc)) # (e13 & !PCI));

TII = (e3 & t) # e11 # (Tload);
Ocupada = e0 # (e6 & t);
Igirada = e7 # e8;
Nroc = e4 # e9;
Nfi = (e4 & !t) # (e3 & !t) # (e7 & t) # e12;
NrII = (e4 & t) # e9 # HHT;
```

Gal 2- Maquina d'estats principal i maquina d

```
Name P1F2G2 ;
PartNo 00 ;
Date 23/11/2016 ;
Revision 01 ;
Designer ls30652_ls30759;
Company La Salle ;
Assembly None ;
Location Barcelona ;
Device g22v10 ;

/* ***** INPUT PINS ***** */
PIN 1 = t;
PIN 8 = Q0;
PIN 9 = Q1;
PIN 10 = Q2;
PIN 11 = Q3;
PIN 4 = Z0;
PIN 3 = Z1;
PIN 2 = Z2;
PIN 5 = 1s;
PIN 6 = Ocupada;
PIN 7 = Ult;
PIN 13 = Ultim;

/* ***** OUTPUT PINS ***** */
PIN 21 = X0;
PIN 22 = X1;
PIN 23 = X2;
PIN 17 = !Ncs;
PIN 18 = !Nwe;
PIN 19 = !Noe;
PIN 20 = OEBD; /*L'utilitzem tambe per fer flanc al biestable igual*/
PIN 14 = NovaTecla;
PIN 15 = Edisplay;
PIN 16 = Tlld; /*Flanc al comptador de display i flanc per agafar valor display*/

/* ***** EQUACIONS ***** */

e0 = !Q3 & !Q2 & !Q1 & !Q0;
e1 = !Q3 & !Q2 & !Q1 & Q0;
e2 = !Q3 & !Q2 & Q1 & !Q0;
e3 = !Q3 & !Q2 & Q1 & Q0;
e4 = !Q3 & Q2 & !Q1 & !Q0;
e5 = !Q3 & Q2 & !Q1 & Q0;
e6 = !Q3 & Q2 & Q1 & !Q0;
e7 = !Q3 & Q2 & Q1 & Q0;
e8 = Q3 & !Q2 & !Q1 & !Q0;
e9 = Q3 & !Q2 & !Q1 & Q0;
e10 = Q3 & !Q2 & Q1 & !Q0;
e11 = Q3 & !Q2 & Q1 & Q0;
e12 = Q3 & Q2 & !Q1 & !Q0;
e13 = Q3 & Q2 & !Q1 & Q0;

es0 = !Z2 & !Z1 & !Z0;
es1 = !Z2 & !Z1 & Z0;
es2 = !Z2 & Z1 & !Z0;
```

```

es3 = !Z2 & Z1 & Z0;
es4 = Z2 & !Z1 & !Z0;
es5 = Z2 & !Z1 & Z0;
es6 = Z2 & Z1 & !Z0;
es7 = Z2 & Z1 & Z0;

estat1 = (es0) # (es1 & Ocupada) # (es4 & 1s & !Ult) # (es6 & 1s) # (es2 & Ocupada) # (es3 & Ocupada);
estat2 = (es1 & !Ocupada);
estat3 = (es2 & !Ocupada);
estat4 = (es3 & !Ultim & !Ocupada) # (es4 & (!1s # Ocupada)) # (es7);
estat5 = (es4 & 1s & Ult & !Ocupada);
estat6 = (es5 # (es6 & !1s));
estat7 = (es3 & Ultim & !Ocupada);

X0 = (estat1 # estat3 # estat5 # estat7);
X1 = (estat2 # estat3 # estat6 # estat7);
X2 = (estat4 # estat5 # estat6 # estat7);

Ncs = (e1 # e6) & t # e2 # e7 # e8 # es3 # e10;
Nwe = (e2 # e8) & t;
Noe = (e6) & t # es3 # e10;
OEBD = !(e1 # e2 # e7 # e8 # e3);
NovaTecla = e4 # e9 # e13;
Edisplay = es3 # es4 # es2 # es7;
TlId = es3 & t;

```

Gal 3- Display part esquerra

```

Name P1F2G3 ;
PartNo 00 ;
Date 10/10/2016 ;
Revision 01 ;
Designer ls30652_XXXXX ;
Company La Salle ;
Assembly None ;
Location Barcelona ;
Device g22v10 ;

PIN [7..11] = A, B, C, D, E;
PIN 6 = Z2;
PIN 5 = Z1;
PIN 4 = Z0;
PIN 3 = Edisplay;
PIN 2 = lgirada;
PIN 13 = Valor;

/*Terminales de salida */
PIN 23 = !sa;
PIN 22 = !sk;
PIN 21 = !sm;
PIN 20 = !sh;
PIN 19 = !su;
PIN 18 = !ss;
PIN 17 = !st;
PIN 16 = !sg;
PIN 15 = !sf;

PIN 14 = !NrcomptSeg;
/*Declaracions */

```

```

/*Ecuaciones booleanas */

sa = ((Asa) # (sa)) & (Edisplay);
sm = ((Asm) # (sm)) & (Edisplay);
sk = ((Ask) # (sk)) & (Edisplay);
sh = ((Ash) # (sh)) & (Edisplay);
su = ((Asu) # (su)) & (Edisplay);
ss = ((Ass) # (ss)) & (Edisplay);
st = ((Ast) # (st)) & (Edisplay);
sg = ((Asg) # (sg)) & (Edisplay);
sf = ((Asf # (!Igirada & Valor)) # (sf)) & (Edisplay);

es2 = !Z2 & Z1 & !Z0;
es5 = Z2 & !Z1 & Z0;

NrcomptSeg = es2 # es5;

FIELD entradas = [Valor, Igirada, A,B,E,D,C];
FIELD salidas = [Asa,Asm,Ask,Ash,Asu,Ass,Ast,Asg,Asf];

TABLE entradas => salidas {
'b'1100000 => 'b' 000010000;

'b'1100001 => 'b' 100110010; /* A          */
'b'1100010 => 'b' 110001001; /* B          */
'b'1100011 => 'b' 100100011; /* C          */
'b'1100100 => 'b' 110001001; /* D          */
'b'1100101 => 'b' 100110011; /* E          */
'b'1100110 => 'b' 100110010; /* F          */
'b'1100111 => 'b' 100100011; /* G          */
'b'1101000 => 'b' 000110010; /* H          */
'b'1101001 => 'b' 110001001; /* I          */
'b'1101010 => 'b' 000000011; /* J          */
'b'1101011 => 'b' 000110010; /* K          */
'b'1101100 => 'b' 000100011; /* L          */
'b'1101101 => 'b' 010101010; /* M          */
'b'1101110 => 'b' 001100010; /* N          */
'b'1101111 => 'b' 100100011; /* O          */
'b'1110000 => 'b' 100110010; /* P          */
'b'1110001 => 'b' 100100011; /* Q          */
'b'1110010 => 'b' 100110010; /* R          */
'b'1110011 => 'b' 101000001; /* S          */
'b'1110100 => 'b' 110001000; /* T          */
'b'1110101 => 'b' 100000011; /* U          */
'b'1110110 => 'b' 000100110; /* V          */
'b'1110111 => 'b' 000100110; /* W          */
'b'1111000 => 'b' 001000100; /* X          */
'b'1111001 => 'b' 001001000; /* Y          */

'b'1111010 => 'b' 100000101; /* Z          */

}

```


Gal 4- Display part dreta

```

Name   P1F2G4;
PartNo  00 ;
Date   10/10/2016 ;
Revision 01 ;
Designer ls30652_XXXXX ;
Company La Salle ;
Assembly None ;
Location Barcelona ;
Device  g22v10 ;

PIN 1 = t;
PIN [6..2] = A, B, C, D, E;
PIN 7 = Z0;
PIN 8 = Z1;
PIN 9 = Z2;
PIN 10 = Edisplay;
PIN 11 = lgirada;
PIN 13 = Valor;

/*Terminals de sortida */
PIN 17 = !se;
PIN 18 = !sd;
PIN 19 = !sr;
PIN 20 = !sp;
PIN 21 = !sc;
PIN 22 = !sn;
PIN 16 = !sb;
PIN 14 = Tload;
PIN 15 = !RIld;
PIN 23 = !Sult;

/*Declaracions */

FIELD entradas = [Valor,lgirada,A,B,E,D,C];
FIELD salidas = [Ase,Asd,Asr,Asp,Asc,Asn,Asb];

se = ((Ase # (!lgirada & Valor)) # se) & (Edisplay) ;
sd = ((Asd # (sd)) & (Edisplay));
sr = ((Asr # (sr)) & (Edisplay));
sp = ((Asp # (sp)) & (Edisplay));
sc = ((Asc # (sc)) & (Edisplay));
sn = ((Asn # (sn)) & (Edisplay));
sb = ((Asb # (sb)) & (Edisplay));

es2 = !Z2 & Z1 & !Z0;
es0 = !Z2 & !Z1 & !Z0;
es6 = Z2 & Z1 & !Z0;
es7 = Z2 & Z1 & Z0;

Tload = es2 & t;
RIld = es6 # es0;
Sult = es7;

TABLE entradas => salidas {
'b'1100000 => 'b' 0001000;

'b'1100001 => 'b' 0101101; /*  A          */

```

```
'b'1100010 => 'b' 1101101; /* B          */
'b'1100011 => 'b' 1000001; /* C          */
'b'1100100 => 'b' 1100101; /* D          */
'b'1100101 => 'b' 1000001; /* E          */
'b'1100110 => 'b' 0000001; /* F          */
'b'1100111 => 'b' 1101001; /* G          */
'b'1101000 => 'b' 0101100; /* H          */
'b'1101001 => 'b' 1000001; /* I          */
'b'1101010 => 'b' 1100100; /* J          */
'b'1101011 => 'b' 0010010; /* K          */
'b'1101100 => 'b' 1000000; /* L          */
'b'1101101 => 'b' 0100100; /* M          */
'b'1101110 => 'b' 0110100; /* N          */
'b'1101111 => 'b' 1100101; /* O          */
'b'1110000 => 'b' 0001101; /* P          */
'b'1110001 => 'b' 1110101; /* Q          */
'b'1110010 => 'b' 0011101; /* R          */
'b'1110011 => 'b' 1101001; /* S          */
'b'1110100 => 'b' 0000001; /* T          */
'b'1110101 => 'b' 1100100; /* U          */
'b'1110110 => 'b' 0000010; /* V          */
'b'1110111 => 'b' 0110100; /* W          */
'b'1111000 => 'b' 0010010; /* X          */
'b'1111001 => 'b' 0000010; /* Y          */
'b'1111010 => 'b' 1000011; /* Z          */
}
```

Gal 5- Display part esquerra

```
Name P1F2G5 ;
PartNo 00;
Date 10/10/2016;
Revision 01;
Designer ls30652_XXXXX;
Company La Salle;
Assembly None;
Location Barcelona;
Device g22v10;

/* ***** INPUT PINS *****/

PIN 5 = Lletra1 ;
PIN 6 = Lletra2 ;
PIN 7 = Lletra3 ;
PIN 8 = Igual ;
PIN 9 = Nfi ;
PIN 10 = fGal ;
PIN 11 = PCI ;

/* ***** OUTPUT PINS *****/
PIN 15 = oLED0 ;
PIN 22 = oLED1 ; /*
PIN 21 = oLED2 ; /*
PIN 20 = oLED3 ; /*
PIN 19 = oLED4 ; /*
PIN 18 = oLED5 ; /*
PIN 17 = oLED6 ; /*
PIN 16 = oLED7 ; /*
```

```
PIN 14 = Fijoc ; /* */
PIN 23 = Nrll ;

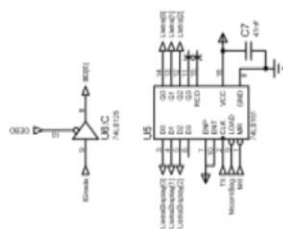
/* ***** EQUACIONS ***** */

Nrll = PCI & fGal;
Fijoc = !(oLED0 # oLED1 # oLED2 # oLED3 # oLED4 # oLED5 # oLED6 # oLED7);

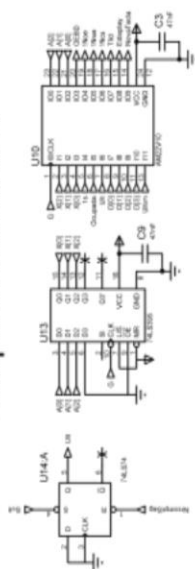
LED0 = !Lletra3 & !Lletra2 & !Lletra1;
LED1 = !Lletra3 & !Lletra2 & Lletra1;
LED2 = !Lletra3 & Lletra2 & !Lletra1;
LED3 = !Lletra3 & Lletra2 & Lletra1;
LED4 = Lletra3 & !Lletra2 & !Lletra1;
LED5 = Lletra3 & !Lletra2 & Lletra1;
LED6 = Lletra3 & Lletra2 & !Lletra1;
LED7 = Lletra3 & Lletra2 & Lletra1;

oLED0 = PCI & (((Nfi & LED0) # (oLED0)) & !(Igual & LED0 & Nfi));
oLED1 = PCI & (((Nfi & LED1) # (oLED1)) & !(Igual & LED1 & Nfi));
oLED2 = PCI & (((Nfi & LED2) # (oLED2)) & !(Igual & LED2 & Nfi));
oLED3 = PCI & (((Nfi & LED3) # (oLED3)) & !(Igual & LED3 & Nfi));
oLED4 = PCI & (((Nfi & LED4) # (oLED4)) & !(Igual & LED4 & Nfi));
oLED5 = PCI & (((Nfi & LED5) # (oLED5)) & !(Igual & LED5 & Nfi));
oLED6 = PCI & (((Nfi & LED6) # (oLED6)) & !(Igual & LED6 & Nfi));
oLED7 = PCI & (((Nfi & LED7) # (oLED7)) & !(Igual & LED7 & Nfi));
```

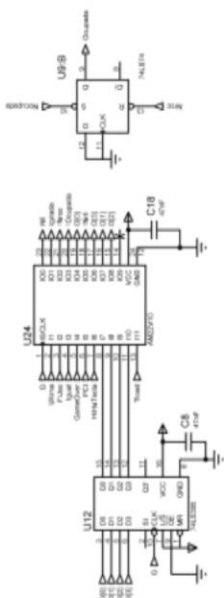
Control memories



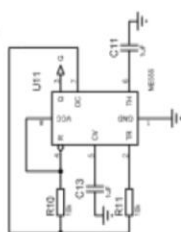
Maquina d'estats 2



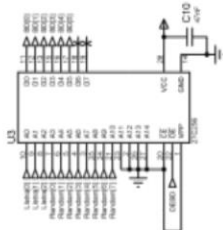
Maquina d'estats 1



Bloc de Temps

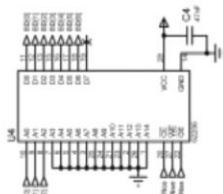


EEProm

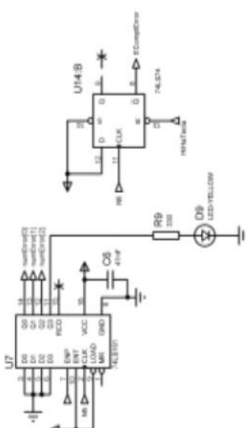


EEProm

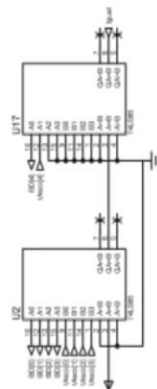
RAM



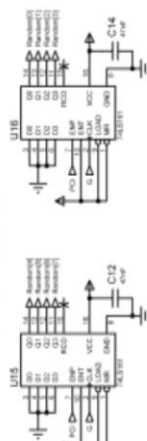
Bloc Errors



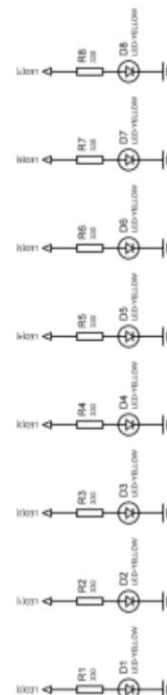
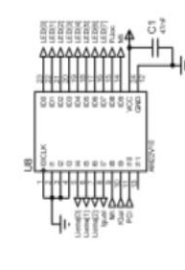
Igual



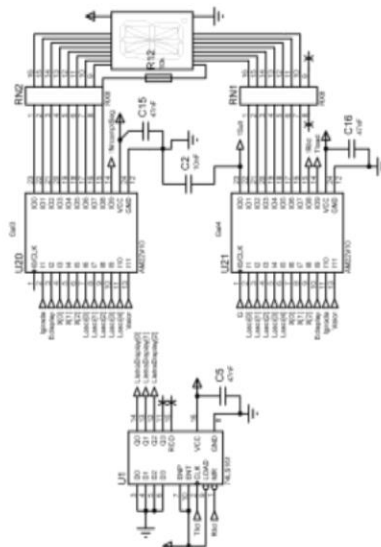
Random



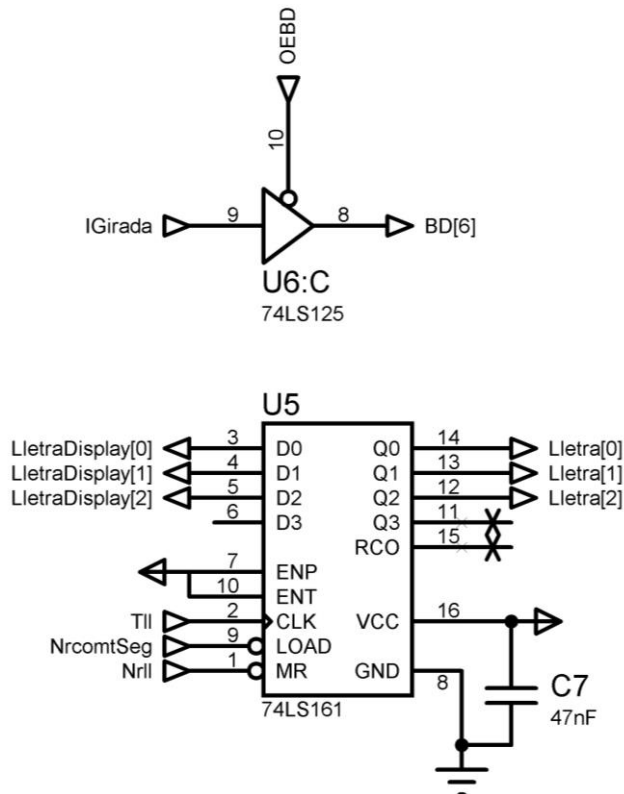
Bloc LEDs



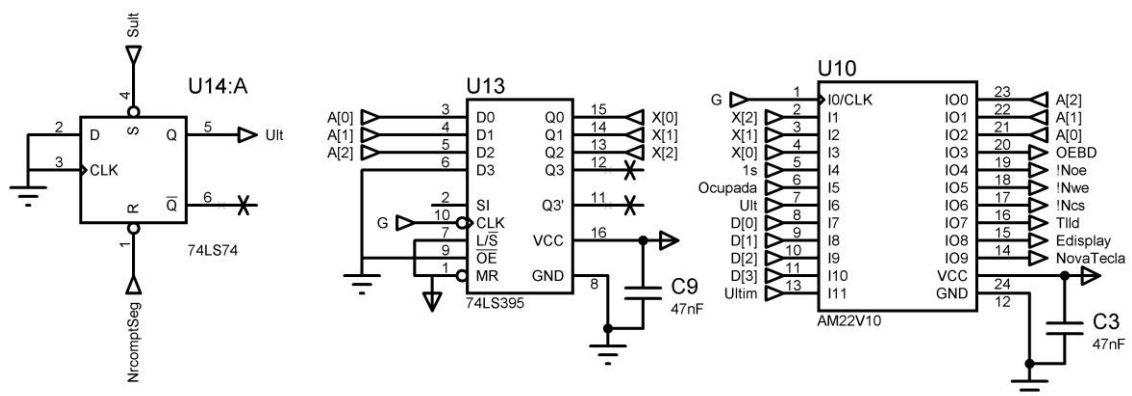
Bloc 16 segments



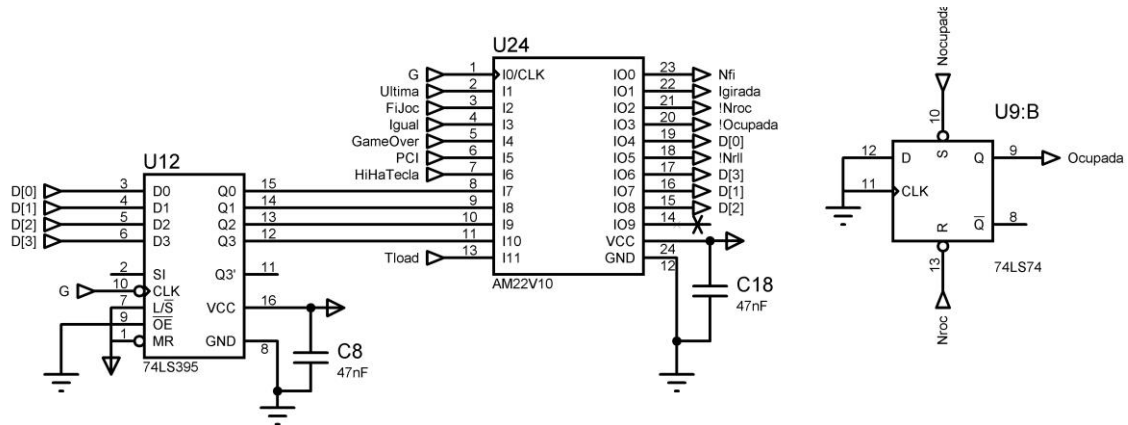
Control memòries



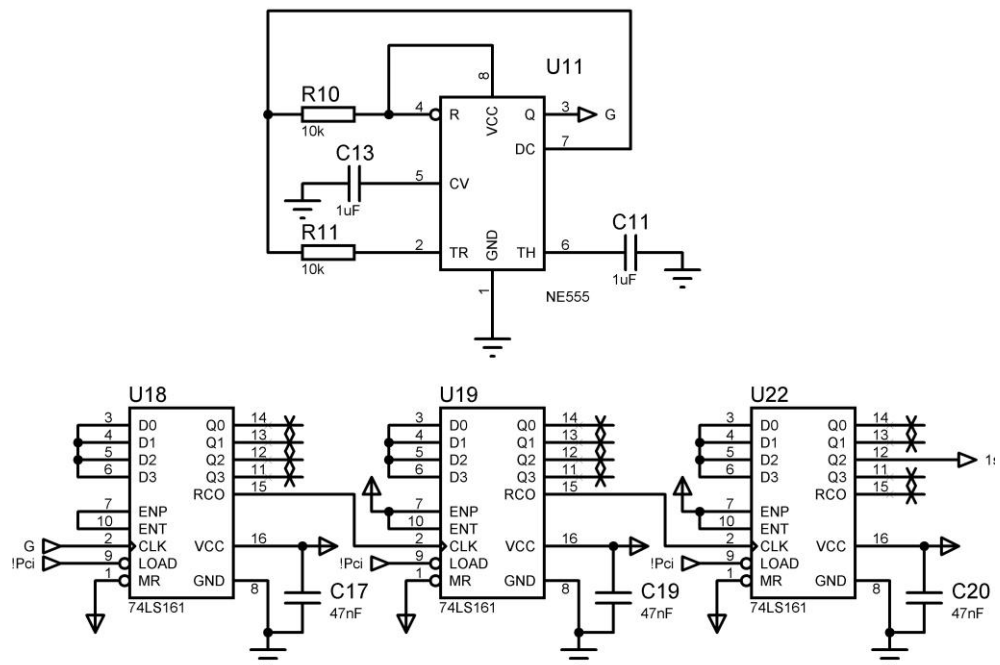
Màquina d'estats 2 (Maquina display)



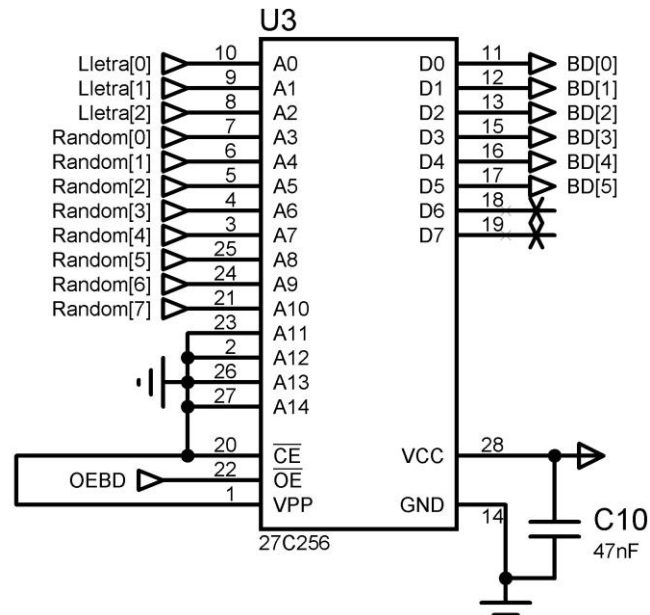
Màquina d'estats 1 (Maquina principal)



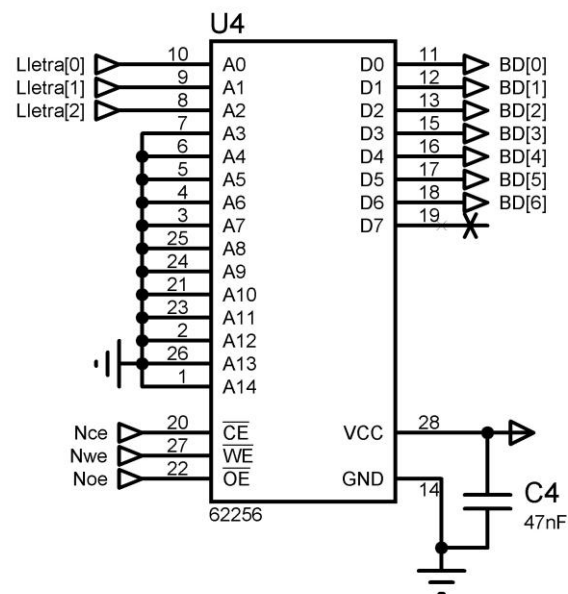
Bloc temps



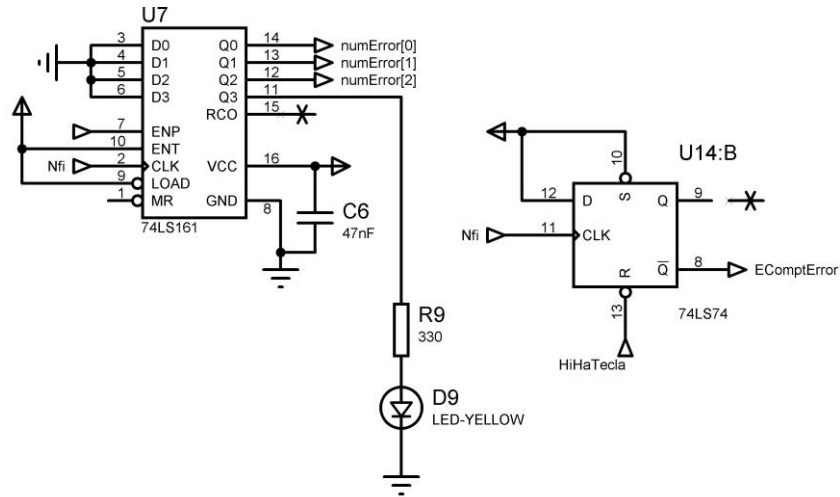
Eprom



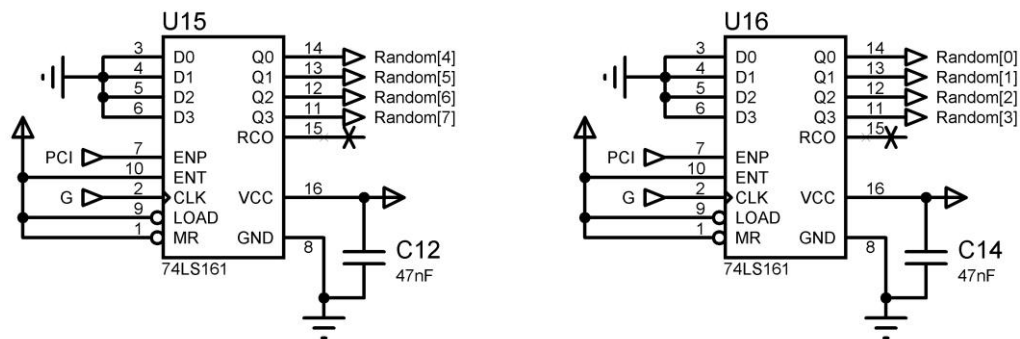
Ram



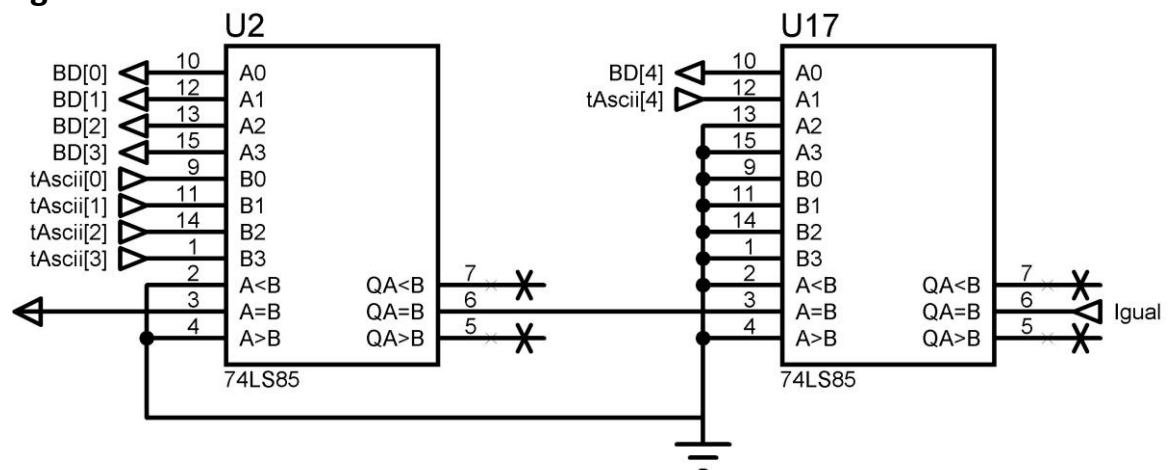
Bloc Errors



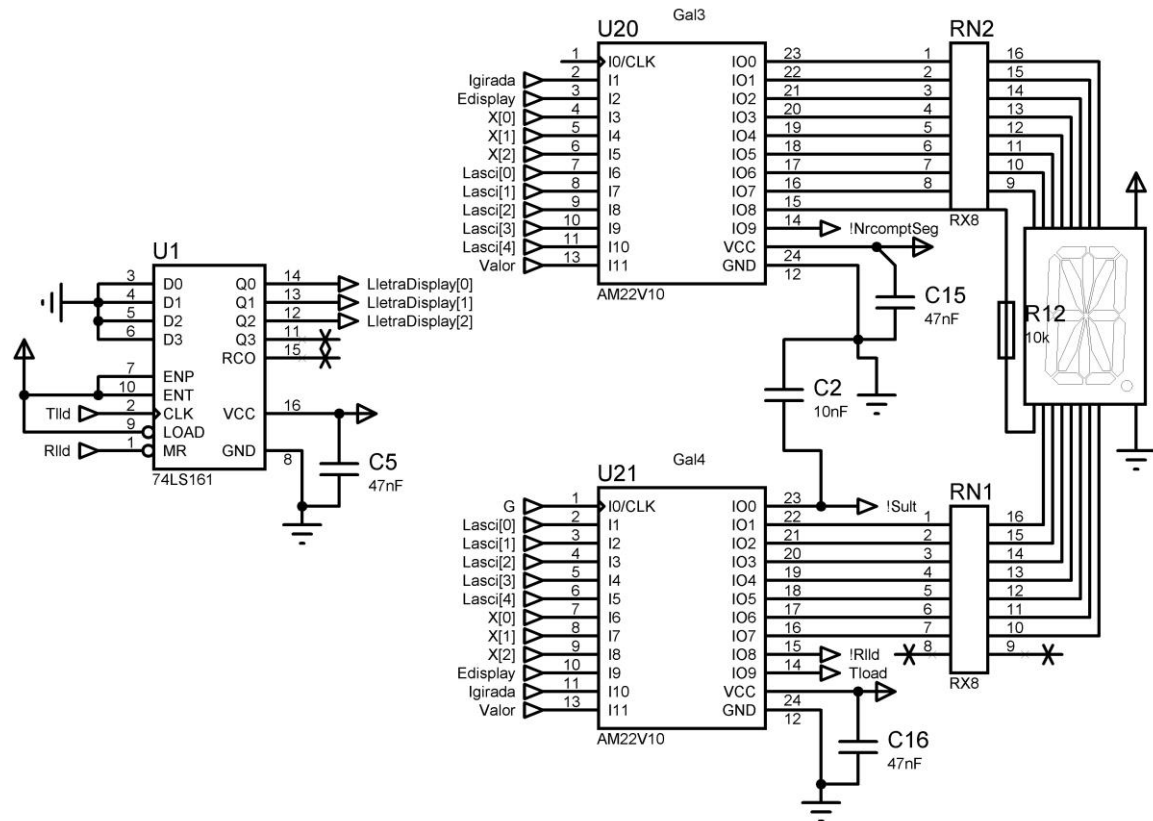
Random



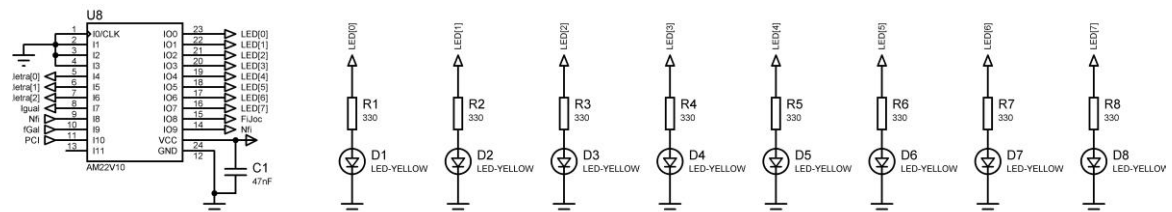
Igual



16 Segments



Bloc Leds



6. Problemes observats

Inicialment els problemes que vam tenir estaven més relacionats amb la fase 1 degut a que durant un període de temps al estar parada es van produir un parell de soldadures fredes, per el que el ne555 ens va deixar de funcionar i ens va costar bastant de detectar. També ens vàrem confondre al soldar els pins d'una placa i de l'altre per lo que no ens donava certs senyals i ens va costar adonar-nos que un bit del bus no estava connectat correctament.

Per el que fa la fase 2 íntegrament vam tenir problemes amb un model de la gal 22V10CQZ, ja que aquesta no accepta utilitzar un input/output per tal d'aprofitar els biestables que tenen a cada entrada/sortida. Con a solució vam optar per utilitzar un altre model 22V10C i aquest no ens va ocasionar cap problema com aquet.

Principalment però, la programació de les gals es una de les coses que ens ha fet perdre mes temps. Per el fet de que son molt sensibles a qualsevol cop, etc. I per tant a vegades teníem que tornar-les a programar per assegurar-nos que estaven be.

Per altre banda, un dels problemes que no ens vam donar compte fins ben al final de la practica ha sigut el que el fet de utilitzar les mateixes sortides com entrades en les gals, ens ha produït una necessitat de mes intensitat per a tota la placa. Per tant, per aquesta falta, les gals no funcionaven be amb errors diversos que no hi trobàvem la raó. Degut a aquest motiu, hem perdut molt temps fins que hem provat d'augmentar el voltatge de la font i amb això un augment de la intensitat.

7. Conclusions

Com a conclusió d'aquesta segona fase, podem destacar que hem sigut capaços de implementar qüestions teòriques que hem vist aquest curs i a cursos anteriors com són la gestió de la informació a una ram, un altre factor que també ens ha agradat i extraiem d'aquesta practica es el fet de que al cercar informació sobre dispositius que utilitzem com les gals hem pogut trobar maneres d'implementar-les per tal d'estalviar hardware com registres o portes lògiques.

Un altre factor important que extraiem d'aquesta practica es el fet de que al realitzar la connexió per bus de la fase anterior amb aquesta em pogut veure com projectes dissenyats independentment poden complementar-se i treballar conjuntament i crear un sistema que externament sembli que sempre ha estat plantejat en conjunt.

També considerem que hem pogut aprendre la finalitat d'un rigorós esquema elèctric, degut a que facilita molt més la feina tan a trobar errors com a soldar correctament i la importància de treballar conjuntament per tal de aconseguir arribar a les dates límits, que en el nostre cas ha resultat complicat.

8. Planificació

En el nostre cas no hem portat un control rigorós de les hores dedicades, però podem aproximar que:

- Plantejament inicial = 5h
- Disseny hardware i maquina d'estats = 10h
- Implementació i resolució de problemes = 40h
- Finalització del projecte = 5h

9. Annex

El vídeo de youtube amb la demostració de funcionament de la placa es aquest:

https://youtu.be/6e6fn_BFBw8