

# Aplicacions electromagnètiques i electròniques

Mar Roselló Canal

P08/19018/00542



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índex

<b>Introducció</b> .....	5
<b>Objectius</b> .....	6
<b>1. Línies de transmissió</b> .....	7
1.1. Introducció a les línies de transmissió .....	7
1.2. Anàlisi general d'una línia de transmissió .....	8
1.3. Disseny de línies de transmissió en plaques de circuit imprès .....	9
1.4. Càlcul de la impedància diferencial .....	11
<b>2. Disseny de plaques de circuit imprès (PCB)</b> .....	16
2.1. Eines necessàries i introducció al disseny d'un PCB .....	16
2.2. Biblioteca de components .....	20
2.3. L'esquemàtic i el <i>netlist</i> .....	22
2.3.1. L'esquemàtic .....	22
2.3.2. El <i>netlist</i> .....	23
2.4. El <i>layout</i> .....	24
2.5. Generació de <i>gerbers</i> i altra documentació .....	27
2.6. Procés d'insolació casolà .....	27
<b>3. Convertidors analògic-digital i digital-analògic</b> .....	30
3.1. La quantificació i la codificació .....	30
3.2. Convertidors analògic-digital .....	32
3.2.1. L'LM3914 .....	34
3.2.2. L'ICL7106-07 .....	36
3.3. Convertidors digital-analògic .....	38
<b>4. Disseny de circuits amb dispositius GAL</b> .....	40
4.1. Principi de funcionament dels dispositius GAL .....	40
4.2. Etapes a seguir en el procés de disseny d'un sistema amb dispositius GAL .....	43
4.3. Introducció al desenvolupament amb ispLEVER .....	44
<b>5. Els microcontroladors PIC</b> .....	50
5.1. Elements fonamentals d'un microcontrolador .....	50
5.2. La rellevància dels PIC de Microchip .....	51
5.3. Les famílies dels dispositius PIC de Microchip .....	53
5.4. Desenvolupament d'un sistema amb microcontroladors PIC .....	54
5.4.1. Desenvolupament de l'aplicació .....	57

<b>6. Els motors elèctrics</b> .....	62
6.1. Consideracions generals sobre els motors elèctrics .....	62
6.2. El motor de continu .....	63
6.2.1. Parts del motor de continu .....	63
6.2.2. Principi de funcionament del motor de continu .....	64
6.2.3. Control de la velocitat .....	65
6.2.4. Funcionament amb PWM .....	65
6.2.5. 6.2.5. Tipus de motors de continu .....	66
<b>7. Introducció als sensors</b> .....	68
7.1. Què és un sensor? .....	68
7.2. Sensors de temperatura .....	69
7.2.1. Termòmetres de dilatació .....	69
7.2.2. Sensors basats en materials resistius metàl·lics .....	70
7.2.3. Sensors basats en resistències semiconductores. Termistors .....	72
7.2.4. Termòmetres basats en unions de materials semiconductors .....	75
7.2.5. Altres tipus de sensors de temperatura .....	78
<b>Resum</b> .....	81
<b>Bibliografia</b> .....	85

## Introducció

En aquest mòdul es pretén donar una eina útil a tots aquells estudiants que vulguin desenvolupar el seu projecte final de carrera en l'àrea d'aplicacions electromagnètiques i electròniques. Així, es desenvolupen en detall alguns temes que al llarg de la carrera no s'han vist en detall.

En el primer apartat es tractarà el tema de les línies de transmissió orientat al disseny de línies *microstrip* i *stripline* per a circuits impresos (PCB).

PCB és la sigla de l'expressió anglesa *printed circuit board*.

En el segon apartat veurem quines són les fases de disseny a tenir presents durant la realització d'un PCB i quines són les eines de disseny assistit per ordinador necessàries per al seu desenvolupament.

El tercer apartat el dediquem als convertidors de senyal. Veurem alguns exemples de convertidors que ens poden ser molt útils en petites aplicacions de mesura o d'instrumentació.

El quart i el cinquè apartat estan orientats al disseny digital amb GAL i PIC respectivament. Veurem quin és el programari necessari per a la seva programació, així com els gravadors necessaris per a gravar el dispositiu.

El sisè apartat és una breu introducció als motors elèctrics. Dins dels motors elèctrics ens centrarem bàsicament en els motors de contínua que són els més utilitzats en aplicacions de baixa potència.

En darrer lloc, a l'apartat 7, parlarem de sensors i basarem la nostra explicació en els sensors de temperatura, ja que ens donen una visió molt general de les diferents tecnologies i tècniques utilitzades en la captació de paràmetres físics.

## **Objectius**

L'objectiu principal d'aquest mòdul és donar a l'estudiant una primera eina per a la introducció en el món del disseny electrònic en general. Els objectius específics són:

- 1.** Entendre la necessitat de l'ús de línies de transmissió.
- 2.** Entendre els passos bàsics per al disseny d'un PCB.
- 3.** Aprendre la metodologia de disseny amb GAL i PIC.
- 4.** Introduir l'ús de convertidors de senyal.
- 5.** Introduir l'ús de motors de contínua.
- 6.** Conèixer les diferents tecnologies utilitzades en sensors.

## 1. Línies de transmissió

En aquest apartat farem una breu aproximació pràctica al món de les línies de transmissió orientat bàsicament al disseny de línies de transmissió en plaques de circuit imprès (PCB).

PCB és la sigla de l'expressió anglesa *printed circuit board*.

En els dos primers subapartats, 1.1 i 1.2, trobareu una primera aproximació teòrica al problema i a la necessitat d'utilitzar línies de transmissió en els nostres dissenys. Al subapartat 1.3 tractarem el cas pràctic de disseny de línies de transmissió de tipus *microstrip*\* i *stripline*\*\* per a la transmissió de senyals unipolars.

\* Línies de transmissió micropista.  
\*\* Línies de transmissió de pista coplanar.

En darrer lloc, al subapartat 1.4 veurem el disseny de línies de transmissió per a la transmissió de senyals diferencials.

### 1.1. Introducció a les línies de transmissió

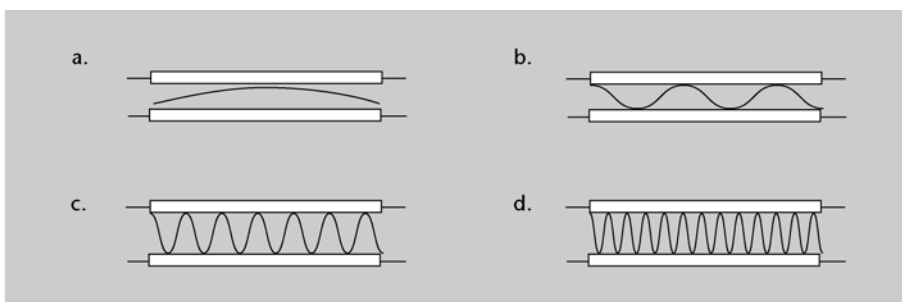
Estem molt habituats a analitzar els nostres circuits utilitzant la teoria clàssica de circuits, segons la qual, apliquem la Llei d'Ohm i, per tant, a punta i punta d'un conductor ideal considerem que hi ha la mateixa tensió. És a dir, un curtcircuit té exactament la mateixa tensió a una punta i l'altra del conductor.

Aquests principis, els apliquem correctament sempre que les dimensions del cable o la pista de circuit imprès o el conductor amb el qual estiguem treballant siguin més petites que la longitud d'ona del senyal amb el qual estiguem treballant. Si no es dona aquesta condició haurem de treballar d'acord amb les teories desenvolupades sobre línies de transmissió i deixar de banda la teoria clàssica de circuits. Podem veure-ho exemplificat a la figura 1.

#### Impedància

La impedància característica d'una línia de transmissió és el quocient entre tensió i corrent en qualsevol punt de la línia.

Figura 1. Línies de transmissió



En una línia de transmissió (o en un conductor qualsevol) podem considerar que tots els punts de la línia es troben a la mateixa tensió, sempre que la longitud d'ona del senyal sigui molt més gran que les dimensions del conductor.

Recordeu que  $\lambda = c/f$  on  $c$  és la velocitat de propagació de l'ona.

#### Figura 1

Fixeu-vos en els 4 casos de la figura; en el cas *a* ens trobem amb un senyal pràcticament continu (longitud d'ona quasi infinita) i, per tant, podem considerar que qualsevol punt del conductor es troba a la mateixa tensió. A mesura que augmenta la freqüència del senyal (casos *b*, *c* i *d*), la longitud d'ona és cada vegada més petita i, per tant, no podem considerar que tots els punts del conductor es troben al mateix potencial.

Les línies de transmissió es caracteritzen per la seva impedància característica i han d'estar carregades amb una impedància adequada per a evitar la presència d'una ona reflectida a la línia de transmissió.

Al següent subapartat farem un breu resum d'aquestes idees i ho posarem en pràctica en el cas de disseny de línies de transmissió en un PCB (*printed circuit board*, circuit imprès): *microstrip* i *stripline*.

### Impedància en el cas d'un PCB

Penseu que en el cas d'un cable: el fabricant ja ens dona la impedància característica però en el cas d'un PCB haurem de calcular-la nosaltres.

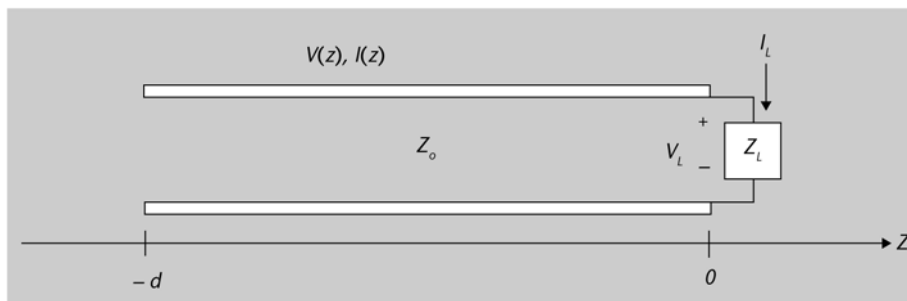
## 1.2. Anàlisi general d'una línia de transmissió

Vegem primer de tot quins són els paràmetres que ens defineixen una línia de transmissió i les condicions que s'hi han de complir.

Una línia de transmissió (ja sigui un cable o una pista en un circuit imprès) queda definida per la seva impedància característica:  $Z_0$ . La impedància característica ens defineix un valor constant en tota la línia que es correspon al quocient entre la tensió i el corrent en qualsevol punt d'aquesta.

Al final d'aquesta línia de transmissió hi tindrem connectat algun component o algun circuit que podem caracteritzar per la seva impedància d'entrada, diguem-li  $Z_L$ . La situació és la que podeu veure representada a la figura 2, on una línia de transmissió d'impedància característica  $Z_0$  està carregada amb una impedància  $Z_L$ .

Figura 2. Esquema d'una línia de transmissió carregada amb una impedància  $Z_L$  sobre la qual cau una tensió  $V_L$  i hi circula un corrent  $I_L$



Si la línia està caracteritzada per una certa impedància constant, el quocient entre tensió i corrent hauria de continuar sent el mateix al final de la línia; per tant, la impedància de càrrega hauria de coincidir amb la impedància característica de la línia ( $Z_L = Z_0$ ).

Per a no fer demostracions numèriques, podríem fer la comparativa amb una canonada d'aigua. Imagineu que teniu una canonada d'un cert diàmetre per la qual circula un cert corrent d'aigua (això equivaldria a la nostra línia de transmissió d'impedància característica  $Z_0$  i per la qual es propaga una ona). Què connectaríem al final d'aquesta canonada si volem continuar tenint una circulació normal d'aigua cap endavant sense canvis en el flux d'aigua dins del tub ni canvis de pressió?

Si continuem la canonada amb una de diàmetre més estret tindrem que aquest estreyniment ens portarà a un augment sobtat de la pressió; si tapem la canonada, el corrent d'aigua rebotarà de nou cap endarrere (ona reflectida), en canvi, si continuem la canonada amb una del mateix diàmetre (carreguem la línia de transmissió amb una impedància de valor igual que la impedància característica de la línia), tindrem una circulació normal del corrent d'aigua.



Podem veure-ho també si ens fixem en el coeficient de reflexió de la línia:

$$\rho = \frac{Z_L - Z_o}{Z_L + Z_o} \quad (1)$$

Si no volem ona reflectida, aquest coeficient de reflexió hauria de ser nul. Arribem, doncs, a la mateixa conclusió:  $Z_L = Z_o$ .

Veurem, doncs, com hauríem d'ajustar a la pràctica les impedàncies característiques dels nostres conductors a les impedàncies de les seves càrregues.

### 1.3. Disseny de línies de transmissió en plaques de circuit imprès

El fabricant d'un cable ens dona sempre la seva impedància característica, però ¿com sabem nosaltres en el disseny d'una pista de circuit imprès quina és la seva impedància característica i, per tant, si l'estem carregant convenientment?

Primer de tot fem un petit càlcul per a entendre en quines situacions seria necessari entrar en ajustos de línies de transmissió.

Quan haurem d'utilitzar les tècniques de disseny de línies de transmissió en un circuit imprès? Imagineu que tenim una placa d'uns 30 cm en la qual estem treballant amb un processador digital que utilitza un senyal de rellotge de 250 MHz. Haurem de tenir presents les tècniques de disseny de línies de transmissió per a conduir aquest senyal de rellotge d'una banda a l'altra de la placa? Calculem-ho; aquest senyal de rellotge té una longitud d'ona  $\lambda = c/f$ :

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8 \text{ m/s}}{250 \text{ MHz}} = 1,2 \text{ m} \quad (2)$$

Normalment considerem que s'han d'utilitzar línies de transmissió en el moment que les dimensions del circuit ( $d$ ) són superiors a 1/6 de la longitud d'ona. En aquest cas tenim doncs:

$$\frac{\lambda}{6} = 20 \text{ cm} \quad (3)$$

Per tant  $d = 30 \text{ cm}$  és superior a 20 cm. Haurem de tenir present en la implementació d'aquesta placa les teories de disseny de línies de transmissió en PCB.

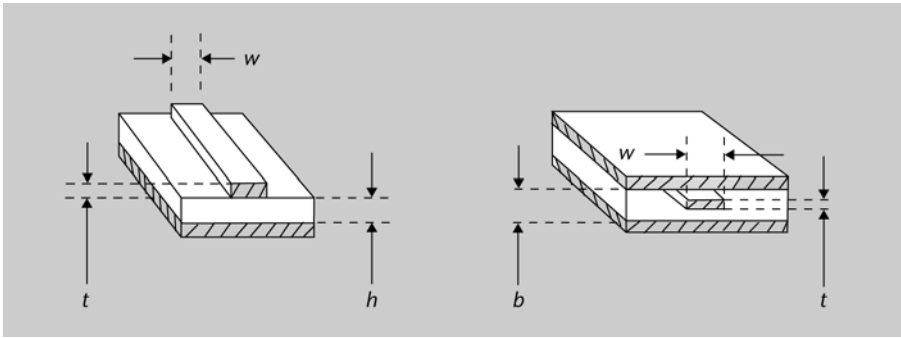
Analitzem amb quina tipologia de línia de transmissió ens podem trobar durant el disseny d'una placa de circuit imprès. Podem trobar dos tipus de disseny:

- *Microstrip*: si la línia de transmissió va per una capa externa del PCB i la capa de sota és un pla de massa.
- *Stripline*: si la línia de transmissió ens queda entre dos plans de massa. És a dir, és una capa interna del PCB envoltada per dos plans de massa.

A la figura 3 podeu veure representada una estructura tipus *microstrip* i una estructura *stripline*.

Vegeu l'apartat 2 d'aquest mateix mòdul.



Figura 3. Representació d'una estructura *microstrip* (a l'esquerra) i una estructura *stripline* (a la dreta)**Figura 3**

A la figura 3 podeu veure la diferència entre una línia *microstrip* i una línia *stripline*. Tal com comentàvem, la primera aniria per una capa externa del PCB i a la capa següent hi trobaríem un pla de massa. La segona, en canvi, queda enterrada entre dos plans de massa.

A la mateixa figura 3 podem veure també quins són els paràmetres que caracteritzen ambdues línies:

- $h$  (en el cas de *microstrip*): distància entre la pista i el pla de massa (si és una placa amb només dues capes es correspondrà aproximadament al gruix de la placa, vegeu la figura 3).
- $b$  (en el cas de *stripline*): distància entre els dos plans de massa.
- $w$ : amplada de la pista de circuit imprès.
- $t$ : el gruix de la pista de coure (pistes conductores).
- $\epsilon_r$ : constant dielèctrica relativa del material utilitzat per a fer la placa (baquelita, fibra de vidre, etc).

Paràmetres com la  $\epsilon_r$ , la  $t$  o la  $h$ , ens vindran donats per la fabricació de la placa, per tant els haurem escollit nosaltres (dins les possibilitats del fabricant) a l'hora d'encarregar la placa; o si no, el fabricant ens sabrà donar aquesta informació tècnica. El paràmetre  $w$ , el gruix de la pista, és el que nosaltres podem variar per a fixar la impedància característica de la nostra línia.

Fixeu-vos que els únics paràmetres diferents a les dues línies són  $h$  i  $b$ , que són els que estan relacionats amb la geometria que diferencia ambdós tipus de línies.

Els experts han calculat d'una forma més o menys empírica quin és el valor de la impedància característica d'una línia *stripline* o *microstrip* en funció dels paràmetres anteriors. Es donaria per les següents fórmules segons si es tracta d'una línia *microstrip* o d'una línia *stripline*:

$$\text{Microstrip: } Z_o = \frac{60}{\sqrt{0,475\epsilon_r + 0,67}} \cdot \ln \left[ \frac{4h}{0,67(0,8w + t)} \right] \Omega \quad (4)$$

$$\text{Stripline: } Z_o = \frac{60}{\sqrt{\epsilon_r}} \cdot \ln \left[ \frac{4h}{0,67\pi(0,8w + t)} \right] \Omega \quad (5)$$

Tot i així, a l'hora de dissenyar ens és més útil aïllar d'aquestes expressions el valor de  $w$  necessari per tal d'obtenir una determinada  $Z_o$  a la línia:

#### Enllaç recomanat

Podeu trobar més informació sobre aquests càlculs en notes d'aplicació de l'empresa National Semiconductors (<http://www.national.com>).

$$\text{Microstrip: } w = \frac{7,463h}{e^{\left(\frac{Z_0\sqrt{0,475\epsilon_r+0,67}}{60}\right)}} - 1,25t \tag{6}$$

$$\text{Stripline: } w = \frac{2,375h}{e^{\left(\frac{Z_0\sqrt{\epsilon_r}}{60}\right)}} - 1,25t \tag{7}$$

**Ús d'un full de càlcul**

És molt útil programar-se aquestes fórmules, per exemple, en un full de càlcul per tal d'anar variant paràmetres i poder anar fent diferents proves i iteracions.

Les unitats d'aquest paràmetre  $w$  coincidiran amb les unitats utilitzades per  $h$  i  $t$ .

**Exemple de disseny**

Suposeu que volem dissenyar una línia *microstrip* d'impedància característica 50 ohms en un PCB a doble cara amb un gruix d'1,6 mm. La cara inferior del PCB serà un pla de massa. El fabricant ens dóna les següents dades de fabricació (vegeu figura 3):

- $\epsilon_r = 4,5$ .
- $t = 0,035$  mm.

Trobem quina seria l'amplada ( $w$ ) adequada per a la pista:

$$w = \frac{7,463 \cdot (1,6 \cdot 10^{-1})}{e^{\left(\frac{50\sqrt{0,475 \cdot 4,5 + 0,67}}{60}\right)}} - 1,25 \cdot (0,035 \cdot 10^{-1}) = 0,29 \text{ cm} \tag{8}$$

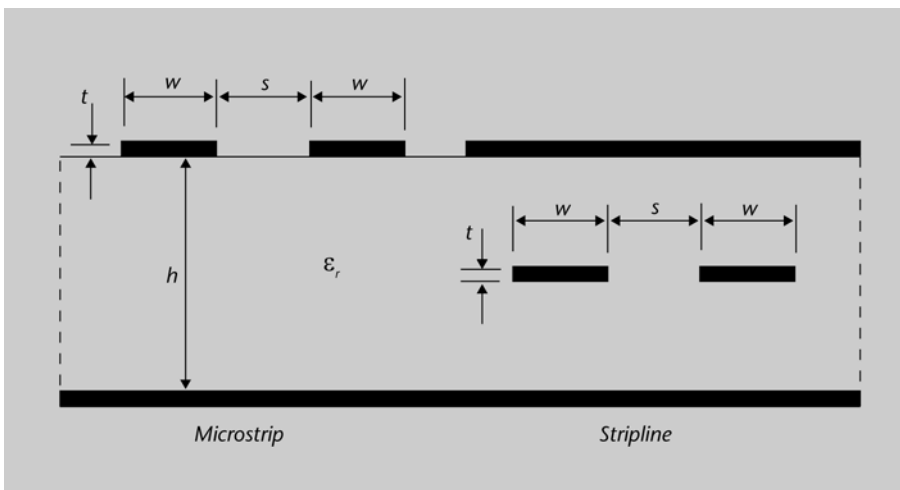
Així doncs necessitariem implementar una pista d'aproximadament uns 3 mm per a poder obtenir la impedància característica de 50 ohms fixada en aquest exemple.

Aneu amb compte amb les unitats de les diferents mides, que han d'estar en aquest cas en cm.

**1.4. Càlcul de la impedància diferencial**

Tot el que hem vist al subapartat 1.3 seria vàlid en qualsevol línia de transmissió en la qual estiguéssim transmetent un senyal unipolar referenciat a massa. Però què passa quan ens trobem davant la transmissió d'un senyal diferencial? Llavors necessitem dos conductors o, en el nostre cas, dues pistes conductores per a fer un *microstrip* o un *stripline* tal com podeu veure a la figura 4.

Figura 4. Exemple de *microstrip* i *stripline* diferencials



**Senyal diferencial**

Recordeu que un senyal diferencial s'obté per la diferència de tensions entre dues línies amb l'objectiu de disposar d'un senyal més immune al soroll i les interferències. Per aquest motiu necessitem ara dues pistes.

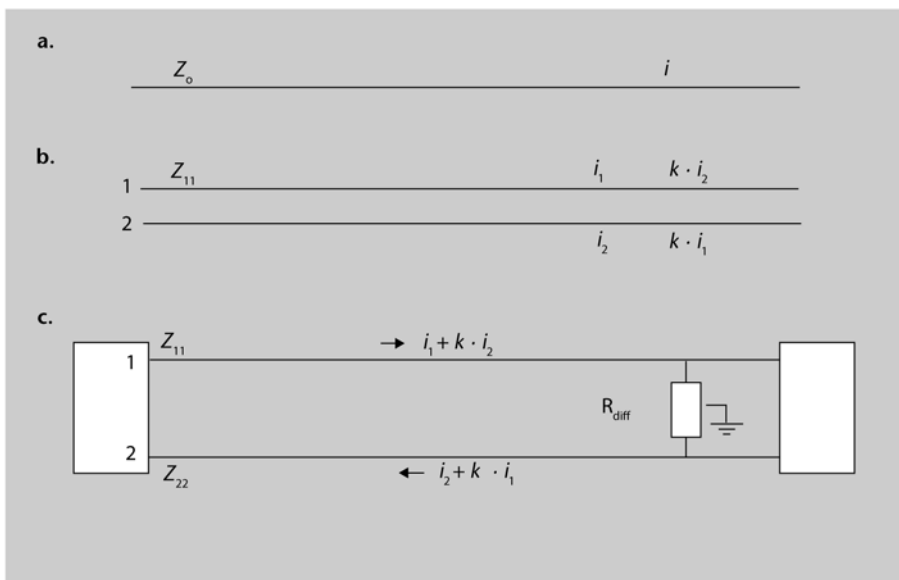
Fixeu-vos que en aquest cas se'ns afegeix un nou paràmetre  $s$  que indica la separació entre les dues pistes.

Com haurem d'adaptar en aquest cas les impedàncies? Penseu que ara la càrrega es trobarà connectada entre els dos conductors; per tant, la impedància que ha de coincidir amb la de càrrega no és ara la impedància característica ( $Z_0$ ) de cada línia per separat tal com l'havíem calculat fins ara, sinó el que anomenarem impedància diferencial entre les dues línies ( $Z_{diff}$ ).

Analitzem per què necessitem adaptar ara respecte d'aquesta nova impedància diferencial i no respecte de la impedància característica com havíem fet fins ara.

Per entendre-ho millor ens basarem en els casos  $a$ ,  $b$  i  $c$  de la figura 5.

Figura 5. Comparació d'una línia de transmissió amb una sola pista referenciada a massa (cas  $a$ ) en el cas de dues pistes diferencials (casos  $b$  i  $c$ ).



En el cas  $a$  de la figura 5 podem veure el mateix cas ja analitzat al subapartat 1.2. És a dir, una única traça o línia per la qual circula un corrent  $i$  i a la qual cau una tensió  $v$  de tal manera que la seva impedància característica  $Z_0$  és  $v/i$ .

En el cas  $b$  en canvi, s'il·lustra el cas de la transmissió per una línia diferencial (dues pistes). En aquest cas cada traça tindria la seva impedància característica  $Z_{11}$  i  $Z_{22}$  (que es correspondria a la  $Z_0$  de quan teníem una sola traça aïllada), però a més a més ara hem de tenir en compte un cert acoblament de corrent a causa del corrent que circula per l'altra traça. És a dir, a cada línia tindrem el seu corrent  $i$  i una part proporcional del corrent de l'altra.

Si considerem que aquest corrent s'acobla segons una constant de proporcionalitat  $k$  i apliquem la Llei d'Ohm en qualsevol punt de les línies, obtenim les següents equacions que ens donen la tensió de cada línia:

$$\begin{aligned}v_1 &= Z_{11} \cdot i_1 + Z_{11} \cdot k \cdot i_2 \\v_2 &= Z_{22} \cdot i_2 + Z_{22} \cdot k \cdot i_1\end{aligned}\quad (9)$$

Podem definir ara dues noves impedàncies, que anomenarem  $Z_{12}$  i  $Z_{21}$ , de manera que  $Z_{12}$  serà  $k$  vegades  $Z_{11}$  i  $Z_{21}$  serà  $k$  vegades  $Z_{22}$ . Llavors les dues expressions anteriors ens quedarien:

$$\begin{aligned}v_1 &= Z_{11} \cdot i_1 + Z_{12} \cdot i_2 = Z_{11} \cdot i_1 + Z_{11} \cdot k \cdot i_2 \\v_2 &= Z_{21} \cdot i_1 + Z_{22} \cdot i_2 = Z_{22} \cdot k \cdot i_1 + Z_{22} \cdot i_2\end{aligned}\quad (10)$$

Si estem davant d'una situació de disseny ben balancejada, és a dir amb dues traces completament simètriques per a conduir el senyal diferencial, es complirà que  $Z_{11} = Z_{22} = Z_o$  i a més a més  $i_1 = -i_2$  (imagineu que pengeu una càrrega determinada al final de les dues línies del cas  $b$  de la figura 5; llavors veureu clar que  $i_1 = -i_2$ ).

Així doncs el valor de  $v_1$  i  $v_2$  quedaria:

$$\begin{aligned}v_1 &= Z_o \cdot i_1 \cdot (1 - k) \\v_2 &= -Z_o \cdot i_1 \cdot (1 - k)\end{aligned}\quad (11)$$

Fixeu-vos, doncs, que en aquest cas la impedància de cada línia per separat, la qual s'anomena ara impedància en mode senar  $Z_{odd}$  (es correspondria a la impedància característica  $Z_o$  però tenint ara present l'acoblament entre les dues línies) prendrà ara el següent valor:

$$Z_{odd} = \frac{v_1}{i_1} = \frac{v_2}{-i_1} = Z_o \cdot (1 - k) = Z_{11} - Z_{12}\quad (12)$$

Assumirem per un moment que acabem cadascuna de les línies amb una impedància del valor calculat connectada a massa. Té això sentit? Fixeu-vos que la resposta és no. Hem parlat que  $i_1 = -i_2$ ; per tant, per aquestes resistències a massa no estaria circulant cap corrent. Quina és, doncs, la manera correcta d'acabar una línia de transmissió diferencial? La solució la tenim en el cas  $c$  de la figura 5, on el que connectem és una impedància entre les dues línies. Aquesta impedància és la que anomenem impedància diferencial  $Z_{diff}$ .

#### Observació

Fixeu-vos que en les dues equacions 9, si no existís l'acoblament,  $Z_{11}$  i  $Z_{22}$  es correspondrien amb la impedància característica de la línia ja definida al subapartat 1.2.

Podem veure que  $v_1 = -v_2$ . Ja ens ho podríem imaginar, atès que eren senyals diferencials.

Aquesta impedància connectada entre les dues línies ha de prendre per valor la suma de les impedàncies en mode senar de cada línia (ja que va entremig de les dues línies), com que aquestes són iguals tenim:

$$Z_{\text{diff}} = 2 \cdot Z_o \cdot (1 - k) = 2 \cdot (Z_{11} - Z_{12}) \quad (13)$$

Així i tot, encara que el que acabem d'explicar està perfectament fonamentat, no ens ajuda gaire a l'hora de fer el nostre disseny pràctic. De nou unes fórmules trobades de manera més o menys empírica ens ajuden al disseny.

En aquest cas tindrem les següents relacions en funció de si es tracta d'un *microstrip* o d'un *stripline*:

$$\text{Microstrip} \quad Z_{\text{diff}} = 2 \cdot Z_o \cdot \left[ 1 - 0,48 \cdot e^{\left(-0,96 \frac{s}{h}\right)} \right] \quad \Omega \quad (14)$$

$$\text{Microstrip} \quad Z_{\text{diff}} = 2 \cdot Z_o \cdot \left[ 1 - 0,48 \cdot e^{\left(-0,96 \frac{s}{h}\right)} \right] \quad \Omega \quad (15)$$

En aquestes expressions tindrem dos paràmetres a controlar, per una banda l'amplada  $w$  de les pistes (amagada dins  $Z_o$ ) i per l'altra la separació  $s$  entre pistes.

Farem finalment un petit exemple pràctic per a veure com aplicaríem aquestes equacions. Imagineu que voleu treballar amb el component que es mostra a la figura 6.

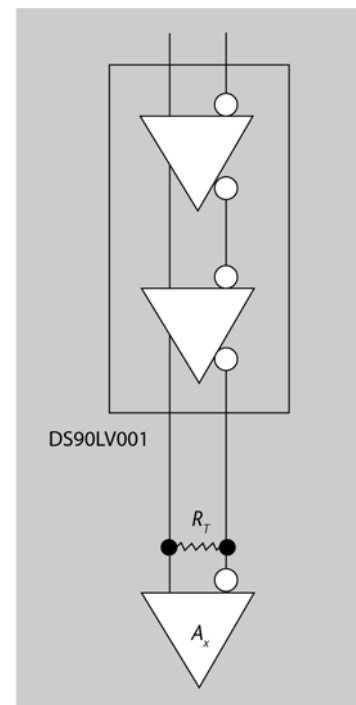
Suposeu que al final de la línia volem fer una adaptació amb una càrrega de 100 ohms (per tant, la impedància diferencial de la línia haurà de ser també de 100 ohms). A més a més, decidim posar el component en un PCB de 8 capes per al qual dissenyarem una línia de transmissió tipus *stripline* entre capes de plans de massa per tal de fer la connexió d'aquest component. Els paràmetres de disseny són en aquest cas:

- El gruix del PCB és de 2,4 mm
- $\epsilon_r = 4,5$
- $t = 0,035$  mm

Quina haurà de ser en aquest cas l'amplada  $w$  de les traces i la separació  $s$  entre elles? Fixeu-vos que aquests dos són els únics paràmetres que podem fixar nosaltres. Tot i així, com que juguem amb dos graus de llibertat (l'amplada  $w$  i la separació  $s$ ), la solució no serà única.

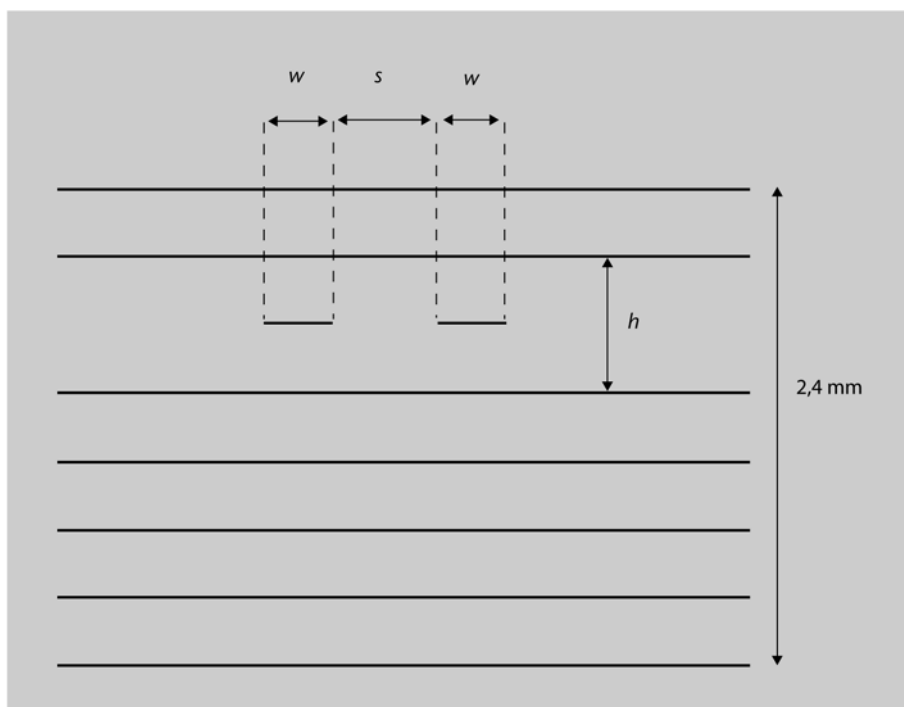
Primer de tot anem a veure quin serà el valor de  $h$  en aquest cas, ja que a diferència de l'altre exemple, ara estem en un PCB de diverses capes. Mireu la figura 7.

Figura 6. Circuit DS90LV001



El DS90LV001 és un circuit integrat que treballa amb senyals diferencials. En aquest cas si la longitud de la connexió ho requerís, hauríem de dissenyar una línia de transmissió diferencial adaptada.

Figura 7. Esquema del PCB a 8 cares amb un gruix de 2,4 mm en el qual volem dissenyar un parell de línies *stripline* en un dels plans interiors



Tal com podeu veure a la figura 7 i suposant que les diferents capes estan equiespaiades, si el PCB té un gruix de 2,4 mm, hem de dividir aquest valor entre 7 (hi ha 8 capes) i multiplicar per 2 per tal d'obtenir el valor  $h$ .

Ara doncs ja tenim tots els paràmetres per a atacar les fórmules donades:

$$\text{Stripline: } Z_o = \frac{60}{\sqrt{\epsilon_r}} \cdot \ln \left[ \frac{4b}{0,67\pi(0,8w + t)} \right] \Omega \quad (16)$$

$$\text{Stripline: } Z_{diff} = 2 \cdot Z_o \cdot \left[ 1 - 0,347 \cdot e^{\left(-2,9 \frac{s}{h}\right)} \right] \Omega \quad (17)$$

#### Observació

En tenir ara dos graus de llibertat,  $w$  i  $s$ , es fa encara més necessari programar aquestes dues fórmules amb, per exemple, un full de càlcul per a poder anar provant diferents possibilitats.

Provant valors amb un full de càlcul podeu comprovar que per exemple una solució amb  $s = 0,42$  mm i  $w = 0,2$  mm ens porta a una impedància diferencial de pràcticament 100 ohms amb una impedància característica d'uns 50 ohms.

## 2. Disseny de plaques de circuit imprès (PCB)

En aquest apartat farem una introducció al disseny de plaques de circuit imprès utilitzant les eines de disseny assistit per ordinador disponibles actualment al mercat.

Al subapartat 2.1 veureu una breu introducció a les eines necessàries així com l'enumeració de les etapes a seguir en el procés. Al subapartat 2.2 veurem com es genera la biblioteca de components necessària per a portar a terme el disseny. Finalment als subapartats 2.3, 2.4, 2.5 i 2.6 trobareu tots els detalls necessaris per entendre les diferents etapes a seguir per a obtenir una placa de circuit imprès acabada i llesta per entrar en funcionament.

### 2.1. Eines necessàries i introducció al disseny d'un PCB

Les sigles PCB es corresponen a les paraules angleses *printed circuit board* (placa de circuit imprès). Les plaques de circuit imprès són el suport que utilitzem per a fer realitat els nostres dissenys electrònics i poder-los convertir així en dispositius amb una certa finalitat.

Un PCB consisteix habitualment en una base d'un material aïllant com baquelita o fibra de vidre, en la qual inserim pistes conductores, normalment de coure, que uniran les diferents parts dels components i circuits integrats que hi decidim soldar. El PCB tindrà també connectors per a poder rebre i enviar senyals d'altres PCB o altres aparells.

Els components, els soldarem sempre a la capa superior o inferior d'aquesta base. Tanmateix, internament, pot estar formada per més capes en les quals també podem disposar pistes conductores, plans de massa, etc. A la figura 8 podeu veure un exemple del que és una placa de circuit imprès.

L'electrònica ha avançat espectacularment en els darrers anys i, a la vegada, les tecnologies emprades en la fabricació de components electrònics i circuits integrats. Com a conseqüència, ens trobem amb la necessitat de realitzar circuits cada vegada més complexos en els quals la miniaturització dels terminals de molts components, i els seus encapsulats han fet que el disseny ja no pugui realitzar-se de manera manual.

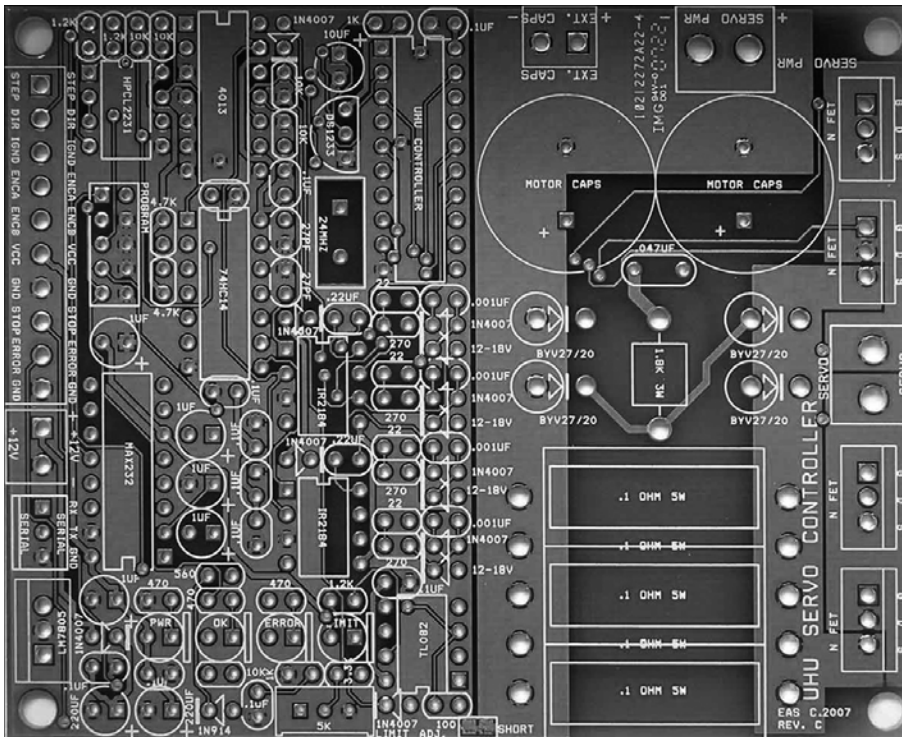
Afortunadament l'aparició d'entorns gràfics ha facilitat considerablement el disseny i fins i tot la simulació de circuits electrònics. Aquests entorns gràfics s'anomenen EDA (*electronic design automation*, que podríem traduir per 'disseny electrònic automatitzat'). D'aquesta manera podem realitzar qualsevol canvi en el disseny del nostre PCB abans que aquest sigui fabricat.

#### Simulació del disseny

Hem de tenir present que la majoria d'entorns ens permeten la simulació del nostre disseny, tant a nivell d'esquema elèctric com posteriorment al disseny del PCB.



Figura 8. Placa de circuit imprès



PCB amb múltiples connexions cap a l'exterior dissenyat amb 8 capes (2 externes i 6 internes). Les capes externes no serien suficients per a fer passar totes les pistes necessàries per a fer totes les connexions.

En els següents subapartats comentarem quins són els passos per a dissenyar un PCB. Això sí, suposarem que tenim clar que el funcionament del nostre disseny és el correcte, per la qual cosa no parlarem de la part de simulació de l'esquema electrònic del circuit. Val a dir també que no centrarem l'explicació en cap entorn EDA concret, sinó que parlarem dels passos bàsics a seguir en el disseny i, per tant, comuns a qualsevol entorn.

El disseny d'un PCB es divideix bàsicament en quatre parts:

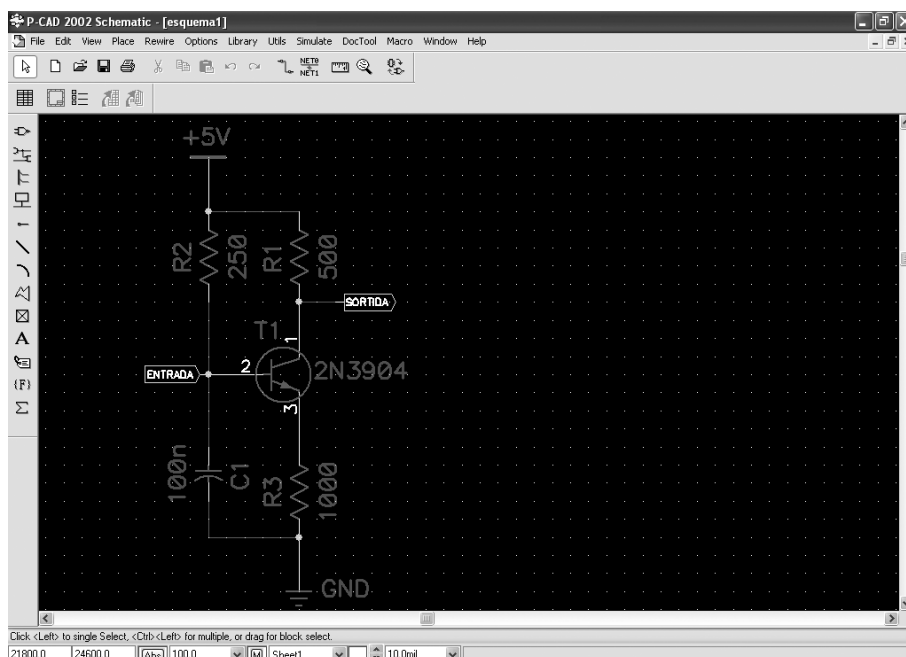
- La realització de l'esquemàtic (en aquest punt podríem tenir la possibilitat de simular).
- L'obtenció del *netlist*.
- La realització del *layout* (en aquest punt també podríem tenir la possibilitat de simular, simulació *postlayout*).
- L'obtenció de fitxers en formats estàndard per a la seva fabricació.

En aquest mòdul tractarem les quatre parts llistades i ens oblidarem del muntatge i la fabricació, no perquè no siguin una part important del procés, tot el contrari, sinó perquè habitualment els deixem en mans de professionals que disposen d'eines i maquinària especialitzada per aquests dos processos; tot i així, repassarem les passes que cal seguir si voleu fer-vos la vostra placa de circuit imprès. La fabricació és el procés químic a partir del qual podem tenir les nostres pistes conductores disposades sobre la base de material aïllant. El muntatge consisteix a soldar tots els components electrònics així com tots els connectors necessaris sobre el PCB.

Veiem ara en què consisteix cadascuna d'aquestes parts. Ho farem primer breument, per a tenir una visió de conjunt, i a continuació entrarem a fons en cadascuna.

L'esquemàtic és el conjunt de connexions elèctriques de tots els components i les parts del nostre circuit. En un esquemàtic assignem un símbol a cada component i indiquem on queda connectat cada terminal del component. Els entorns EDA tenen sempre un editor especialitzat en el dibuix de l'esquemàtic. A la figura 9 tenim un exemple d'esquemàtic.

Figura 9. Exemple d'esquemàtic



En un esquemàtic, cada tipus de components té un símbol assignat. Tot i que habitualment els símbols són força estàndards, sempre hi pot haver lleugeres diferències segons el gust del dissenyador o les biblioteques utilitzades per l'entorn EDA.

L'esquemàtic però, no representa les pistes conductores del nostre circuit, ni tampoc la forma dels terminals on han d'anar soldats els components i connectors. En el *layout* tenim la "fotografia" del nostre PCB; s'hi representa la mida exacta dels terminals dels diferents components, el traçat de les diferents pistes, la disposició de tots els components, etc. L'entorn EDA disposarà també d'un editor específic per a la realització del PCB. Hi podrem dibuixar les pistes de les diferents capes, la forma exacta dels components, indicar el contorn de la placa, etc.

L'editor d'esquemàtics i l'editor de *layout* tenen, però, una manera de comunicar-se que és a través del *netlist*. El *netlist* representa el mateix que l'esquemàtic però en format text, de tal manera que l'editor de *layout* pot carregar aquesta informació per a buscar a la biblioteca de l'entorn el *footprint* (empremta) de cada component. El *netlist* pot indicar-nos quins terminals hem de connectar amb quins a l'hora de dibuixar les diferents pistes.

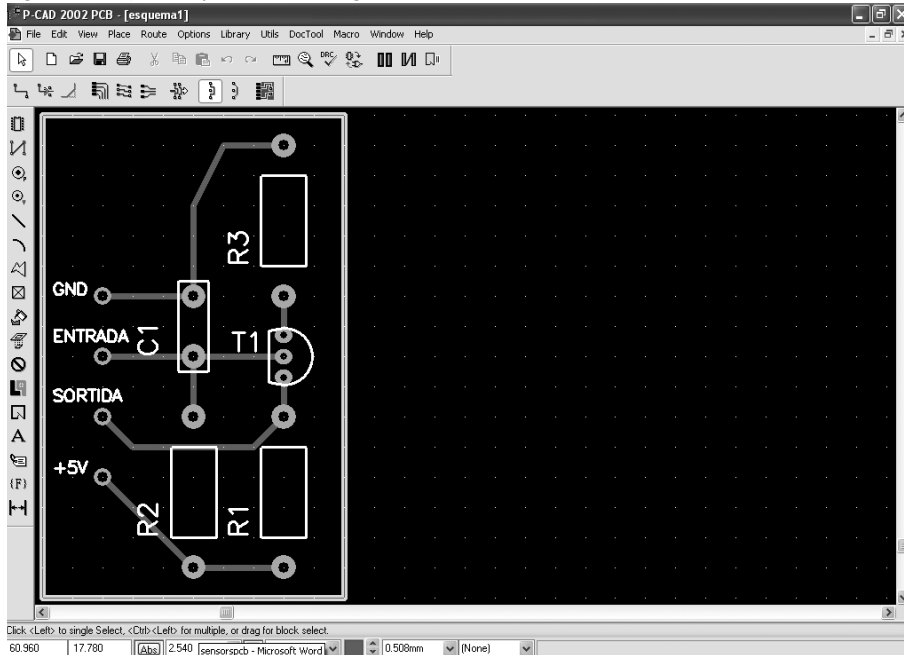
A la figura 10 podem veure un possible *layout*, que es correspondria a l'esquemàtic de la figura 9.

### Components d'un esquemàtic

Fixeu-vos que la posició dels components en un esquemàtic no és rellevant, el que és rellevant és on queda connectat cada terminal del component. No té res a veure, però, amb la disposició física posterior dels components quan obtinguem el PCB.

### Footprint

El *footprint* representa la forma exacta dels terminals i de l'encapsulat del component per la banda que aquest ha d'anar soldat al PCB.

Figura 10. *Layout* esquemàtic de la figura 9

*Layout* en el qual veiem serigrafada la forma dels diferents components i les pistes conductores que els uniran una vegada fabricat el PCB. Fixeu-vos en la referència per a veure a quin component de l'esquemàtic es correspon cadascun. Podem veure també el contorn de la placa.

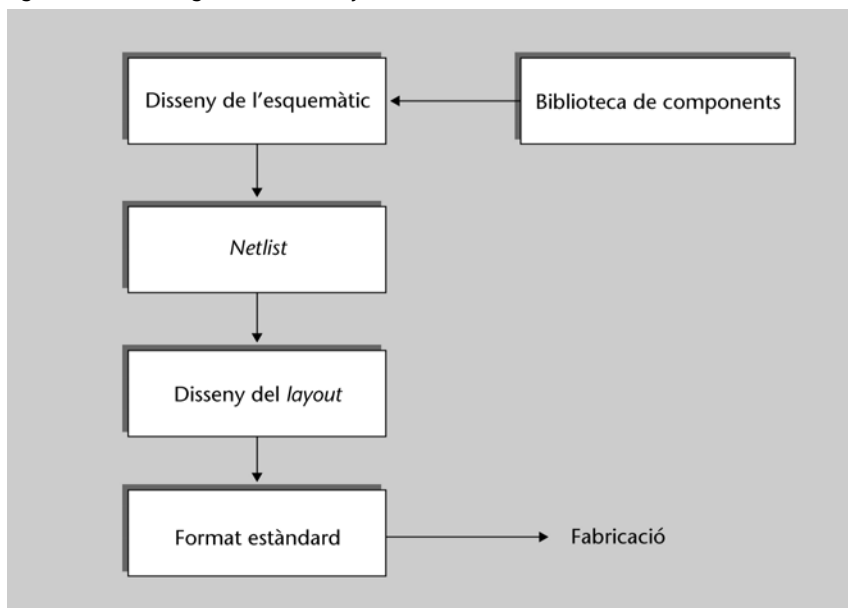
Si editéssim el *netlist* d'aquest circuit hi podríem veure en format text totes les connexions, tots els tipus de components, el seu valor, etc. Una vegada dissenyat el PCB podrem procedir ja a extreure'n la documentació necessària en formats estàndard per tal que el fabricant pugui produir allò que nosaltres hem dissenyat amb l'entorn EDA.

Podríem resumir el procés explicat en l'esquema de la figura 11. Tot seguit, en els subapartats següents, veurem més detalls de cadascuna de les parts.

#### Documentació dels entorns EDA

Per l'existència de nombrosos entorns EDA de disseny, els diferents entorns es veuen amb l'obligació d'extreure tota la documentació seguint una sèrie de formats estàndards que pugui entendre qualsevol fabricant.

Figura 11. Passos seguits en el disseny del PCB



L'entorn EDA trobarà els components a inserir en una biblioteca de components. Un cop acabat l'esquemàtic generarem el *netlist* per tal de carregar-lo a l'editor de *layout*. Una vegada acabat el disseny podrem passar ja a la seva fabricació.

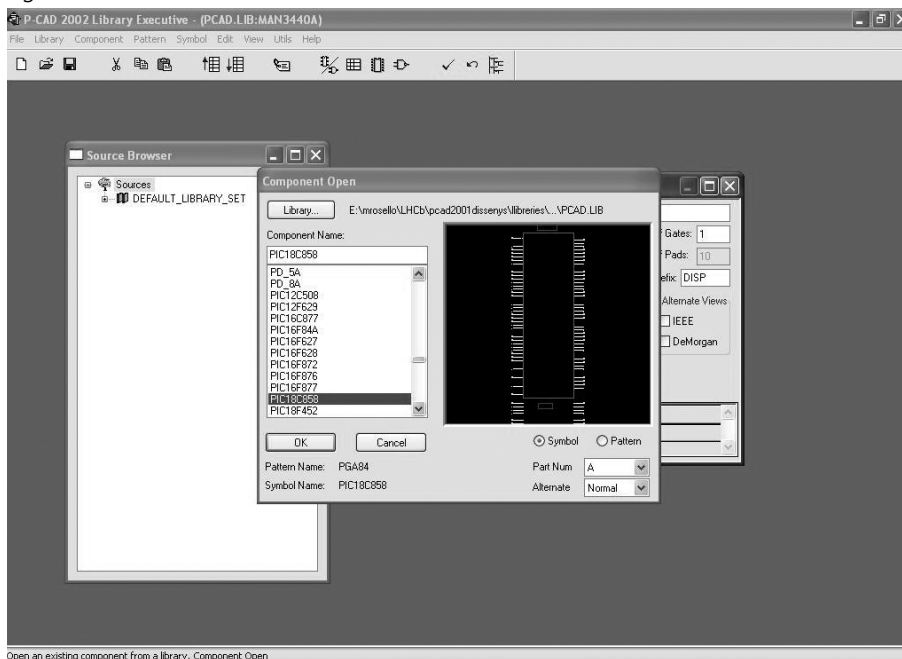
## 2.2. Biblioteca de components

Els components que inserim a través de l'editor d'esquemàtics els hem de tenir prèviament definits en una biblioteca. La biblioteca pot ser una biblioteca estàndard que ja formi part del mateix paquet de l'entorn EDA o una biblioteca que ens haguem creat nosaltres mateixos amb els nostres components personalitzats. Per aquest motiu l'entorn disposa també d'un tercer editor: l'editor de components.

Tant si els components formen part d'una biblioteca estàndard com d'una biblioteca que ens hem implementat nosaltres mateixos, estan formats per dues parts: el símbol del component i el *footprint*. El símbol és el dibuix que identifica el component en l'esquemàtic, el *footprint* és la 'petjada' sobre la qual col·locarem el component en el PCB, és a dir, la representació del component en el *layout*.

A les figures 12 i 13 tenim, respectivament, un exemple de símbol i *footprint* respectivament. Es tracta en aquest cas del símbol i el footprint del PIC18C858.

Figura 12. Símbol del PIC18C858

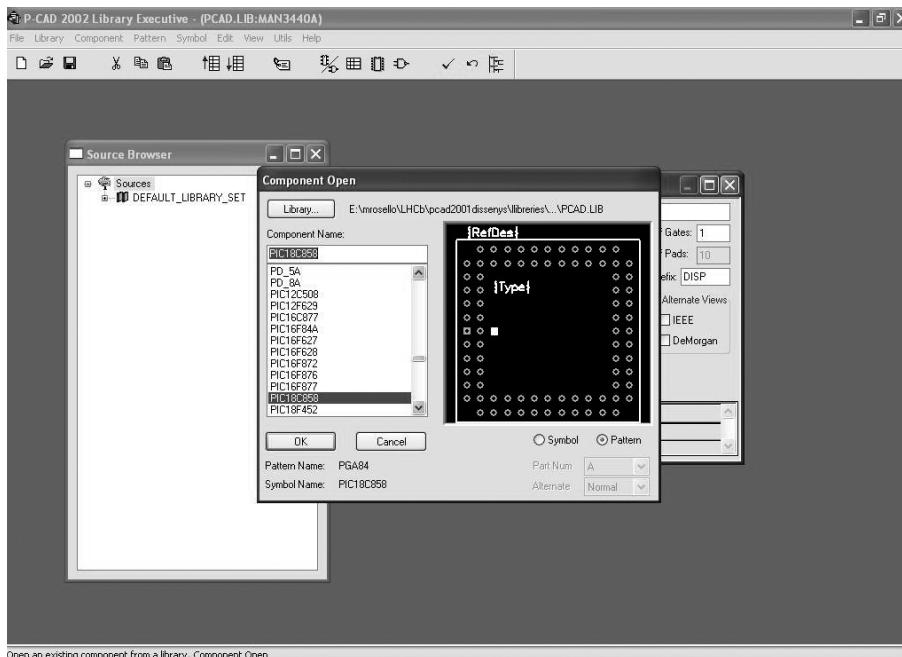


Utilitzarem el símbol tan sols en la realització de l'esquemàtic. Tots els pins aniran correctament identificats, però la seva forma i mida no s'ha de correspondre pas a la forma i mida reals del component, ja que es tracta tan sols d'un símbol.

Cada terminal de l'esquemàtic s'ha de correspondre a un *pad* del *footprint*. Els *pads* són aquelles zones per on fixarem els terminals del circuit integrat al PCB. Així doncs, el terminal i el *pad* corresponent han d'anar identificats de la mateixa manera en el símbol i en el *footprint*, respectivament. Podem trobar *pads* de moltes formes i mides diferents. Abans de dibuixar-los (en cas que ens creem nosaltres els components) haurem d'estar segurs de les mides exactes dels terminals dels components, així com de les dimensions de l'encapsulat. Pen-

seu que després el component hi ha de coincidir del tot; si no, no el podríem soldar correctament i no faria un bon contacte amb el PCB.

Figura 13. Footprint del PIC18C858



El footprint pren la forma i mides exactes de la planta del component una vegada situat sobre el PCB.

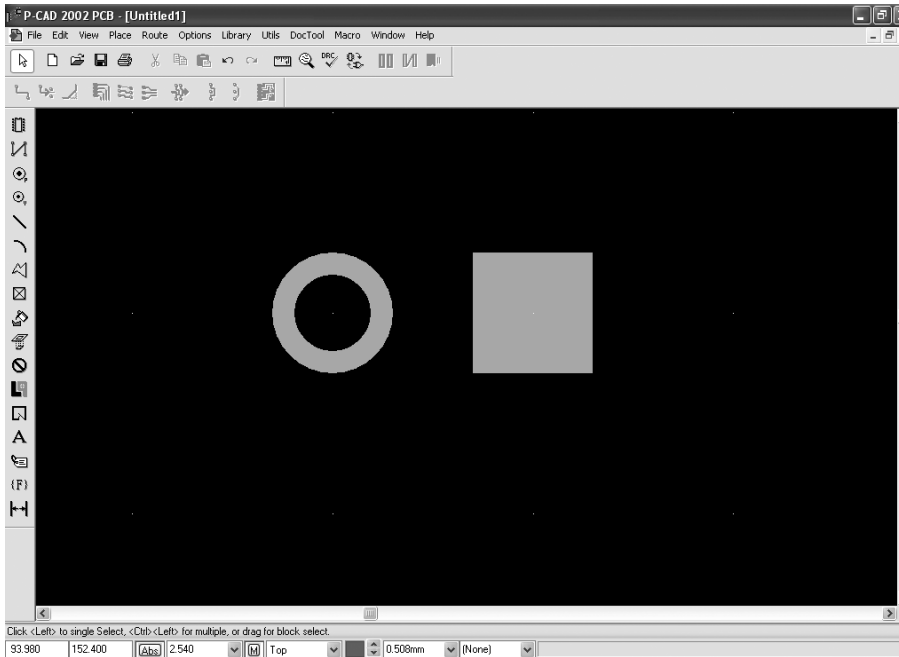
A les figures 9 i 10 tenim més exemples de símbols i *footprints*. Tot el que apareix representant els components de la figura 9 són símbols. En canvi, a la figura 10 hi podeu veure els corresponents *footprints*.

Podem diferenciar bàsicament dos tipus de *pads* per a soldar els nostres components:

- els *pads* per a components *through-hole*, els quals disposaran d'un forat ja que el terminal ha de travessar el PCB i els soldarem pel costat contrari on tenim el component,
- i els *pads* per a component de muntatge superficial (anomenats *components SMD, surface mounted device*), en els quals el terminal del component se solda a la mateixa cara on tenim el component.

A la figura 14 podem veure la diferència entre aquest dos tipus de *pads*. Fixeu-vos que en un podem veure el forat pel qual travessarà el terminal i en canvi a l'altre no.

Una vegada estiguem segurs que tots els components que hem d'utilitzar es troben en una biblioteca o una altra, ja podem obrir l'editor d'esquemàtics per a començar l'esquema del nostre disseny.

Figura 14. Pad *through-hole* i pad SMD

A l'esquerra tenim un *pad through-hole* circular (la part interior més fosca aniria perforada i seria per on passaríem el terminal del component). A la dreta tenim un *pad SMD* quadrat (el terminal aniria soldat directament sobre de la superfície del *pad*).

## 2.3. L'esquemàtic i el *netlist*

Anem a veure a continuació la manera com habitualment entrem els components i les connexions existents en el nostre disseny, l'esquemàtic. Tot i així, l'esquemàtic pot ser poc compatible entre uns entorns EDA i uns altres, ja que cadascú pot utilitzar símbols dibuixats lleugerament diferents; per aquest motiu, es fa necessària la conversió de l'esquemàtic a un fitxer de text, el *netlist*.

Als subapartats 2.3.1 i 2.3.2 s'expliquen els aspectes més importants de l'esquemàtic i del *netlist*.

### 2.3.1. L'esquemàtic

Editar l'esquemàtic no té habitualment cap secret en cap dels entorns EDA, sempre que prèviament ens haguem proveït d'una bona biblioteca de components.

Des de l'editor d'esquemàtics cridarem els components que desitgem de la biblioteca i els identificarem amb una etiqueta única per a cada component. Ens podem fixar de nou en l'esquemàtic de la figura 9 on hem identificat cada component amb un nom R1, R2, R3, T1 i C1 que no es repeteix. Una vegada inserits els components podem unir els seus terminals mitjançant cables o si es tracta de dissenys digitals podem utilitzar busos (els quals identifiquen més d'una línia sota un sol nom).

D'altra banda, haurem d'identificar també amb etiquetes els diferents ports d'entrada i sortida del disseny. Tingueu en compte també que, tot i que la posició dels components a l'esquemàtic no és rellevant mentre els terminals es-

tiguin connectats correctament, un esquema net i ordenat sempre ens pot ajudar a entendre més fàcilment el funcionament del circuit.

En l'esquemàtic, si ho desitgem, podrem editar les propietats de cada component i assignar-los un valor. Aquesta part és útil si després volem extreure una llista de tots els components amb el seu valor des del nostre entorn EDA o si per exemple volem simular l'esquemàtic.

En editar les propietats podrem escollir també quina informació ha de ser visible de cada component i quina no. Si us fixeu de nou en la figura 9, veureu que en aquest cas hem optat per no visualitzar el valor del transistor, ja que no és rellevant, però en canvi, en visualitzem el tipus (2N3904) i l'enumeració dels terminals. Per les resistències, per contra, hem fet al revés, el tipus (si és una resistència de ½watt, etc.) no el visualitzem ni tampoc la numeració dels terminals (en el fons si la soldem al revés no passa res) i en canvi en aquest cas sí que hem considerat rellevant visualitzar-ne el valor.

Una vegada hem completat l'esquemàtic hi ha molts entorns que en permeten fer una simulació directament. En aquests casos, en els components de la biblioteca, a part del símbol i el *footprint*, hi tindriem també el model de comportament del dispositiu. Si fem una simulació, serà important que no ens haguem descuidat el valor de cap component que el requereixi.

Si no volem fer una simulació del nostre circuit, podem generar directament el *netlist* per tal de poder procedir amb el *layout* del disseny.

### 2.3.2. El *netlist*

El *netlist* converteix tota la informació de l'esquemàtic en un fitxer de text. En aquest fitxer figuraran els components que utilitzem i les biblioteques a les quals pertanyen, on va connectat cada terminal de cada component, les propietats que haurem editat de cadascun, etc.

A continuació teniu petits fragments del *netlist* del circuit de la figura 9. Fixeu-vos que comença definint cada component: el nom que li hem assignat, el tipus de component, el nom del *footprint* (*patternName*) i fins i tot el valor en el cas de les resistències. Després de les definicions dels components hi tindriem la definició de cada node (*net*) on s'indica per quins terminals (i de quin component) està format cada node del circuit.

```
(netlist "Netlist_1"
(compiust "T1"
(compRef "2N3904_1")
(originalName "2N3904")
(patternName "TO92V")
)
(compiust "R1"
(compRef "RES-1/2W_1")
(originalName "RES-1/2W")
```

#### Observació

El terme *net* fa referència tant a un punt com a un cable; dit d'una altra manera, tot el que està unit a un determinat lloc (igual potencial), és un *net* o *node*, encara que físicament dibuixat no sigui un punt.

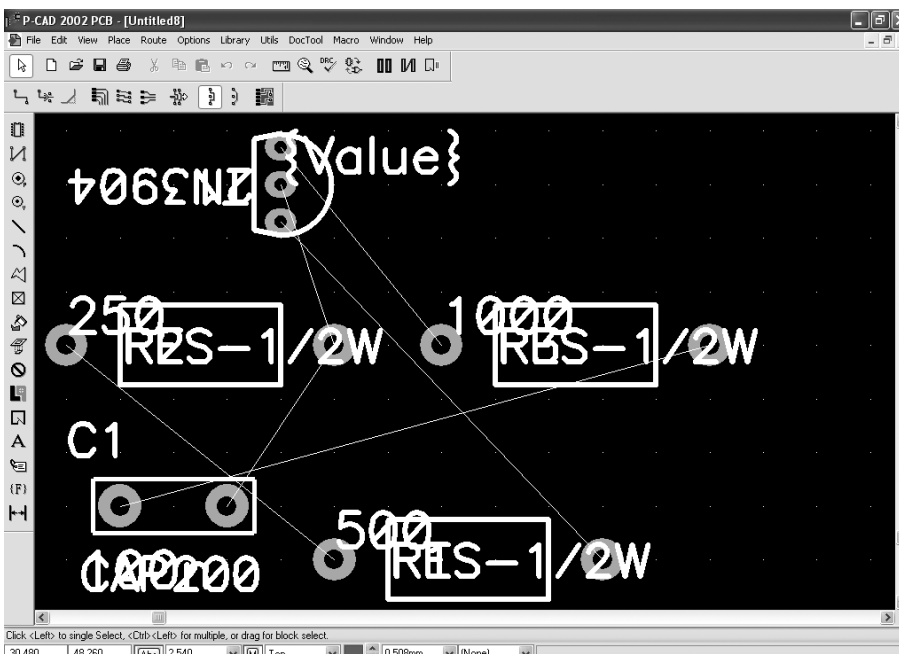
```
(compValue "500")
(patternName "RES-1/2W")
)
...
(net "ENTRADA"
(node "T1" "2")
(node "R2" "2")
(node "C1" "2")
)
(net "+5V"
(node "R2" "1")
(node "R1" "1")
)
)
)
```

## 2.4. El layout

Tota la informació continguda en el *netlist* la necessitarà l'editor de *layouts* per a començar amb el disseny. Una vegada carreguem el *netlist* a l'editor de *layouts* podrem veure els patrons (*footprint*) de cada component i unes línies que ens indicaran quins *pads* van connectats entre ells, és a dir, formen part d'un mateix node del circuit.

L'editor carrega el *netlist* d'una manera molt barroera, per exemple molts components queden els uns sobre els altres. A la figura 15 podem veure un exemple del que ens trobaríem en carregar el *netlist* de l'esquemàtic de la figura 9. És feina nostra col·locar els components allà on volem que estiguin finalment en el PCB. Hem de pensar que el *layout* és en el fons una fotografia del que serà el PCB una vegada fabricat.

Figura 15. *Layout* del disseny just després de carregar el *netlist*



Tots els components queden desordenats i amb tota la informació sobreposada. Unes fines línies (*nets*) indiquen les connexions que hi hauria d'haver entre els *pads*.

A partir d'aquí podrem col·locar els components de la manera que creiem més convenient i començar a traçar el que seran pistes conductores (pistes de cou-

### Traçat de pistes

El traçat de les pistes es pot fer de manera automàtica o manual. Les eines no optimitzen gaire encara el traçat automàtic, per això en la majoria de dissenys s'opta per un traçat manual.

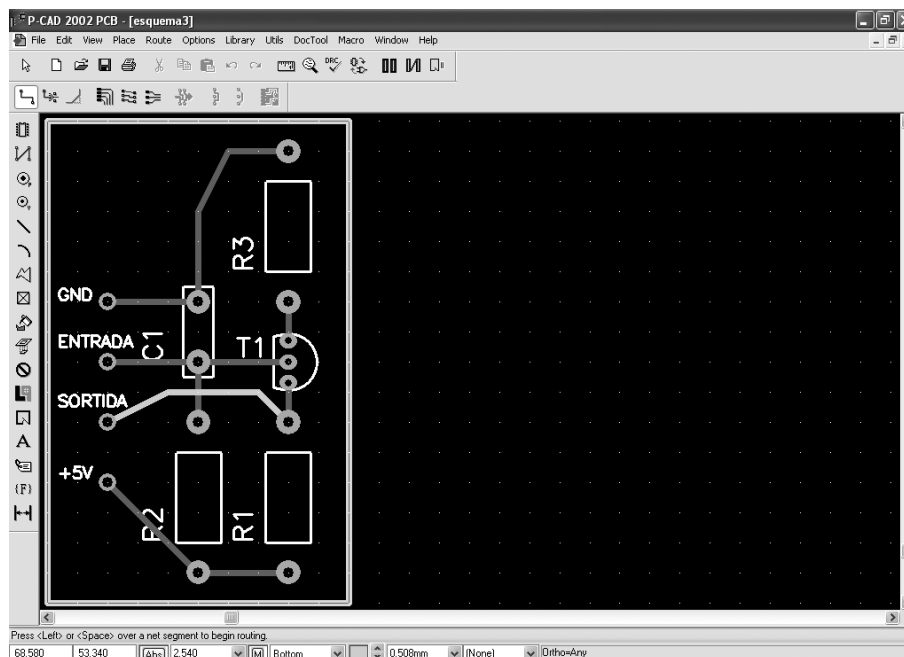


re) entre uns *pads* i els altres. Unes petites línies anomenades *nets* ens indiquen quines connexions haurem de traçar amb pistes: són les línies clares que podeu veure a la figura 15, que indiquen la unió dels diferents *pads*.

Prèviament, però, haurem de definir quantes capes conductores tindrà el PCB. Per defecte tindrem sempre les dues externes: la capa superior (anomenada *top*) i la capa inferior (anomenada *bottom*). En aquestes dues capes podrem traçar totes les pistes que ens interessen. L'editor representa en colors diferents les pistes traçades en capes diferents.

A la figura 10 hem pogut veure un possible disseny del PCB traçant totes les pistes per la mateixa cara *top*. A la figura 16 podem veure una petita mostra del resultat si decidim traçar pistes per les dues cares *top* i *bottom*, seria llavors un disseny a doble cara. A la figura podeu veure les pistes de les dues cares representades amb colors diferents.

Figura 16. Disseny de *layout* a doble cara



En aquest cas la pista que unia el *pad* de sortida amb un dels *pads* de la resistència R1 hem decidit fer-la passar per la banda on ja hi havia una pista; per tant, perquè no es curtcircuitin una de les dues ha de passar per la cara contrària (*bottom*).

Tot i que el disseny que estem utilitzant d'exemple és molt petit i no necessitem més capes, podríem decidir també afegir una capa interna. Aquesta capa interna podrà ser de dos tipus:

- Capa de senyal: capa interna per on podrem fer passar pistes igual que per *top* i per *bottom*.
- Capa de pla d'alimentació o de massa: capa interna dedicada per complet a la conducció d'algun nivell d'alimentació o massa amb l'objectiu de disminuir-ne la impedància i millorar el comportament del circuit.

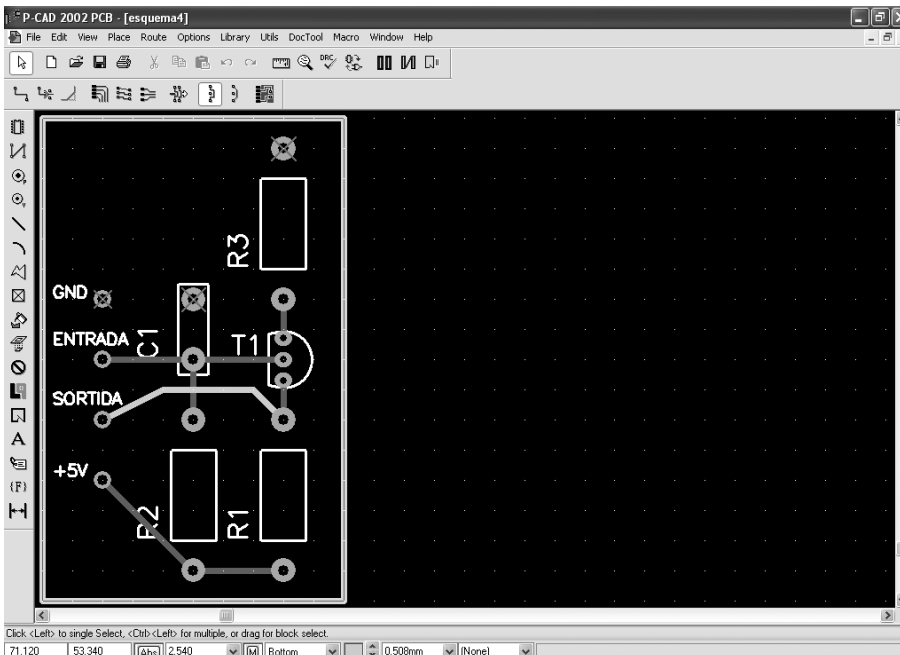
Vegem com canviaria l'aspecte del *layout* de la figura 10 si a més a més decidim afegir una capa interna connectada al node de massa (GND). Llavors (figura 17) ens podem estalviar totes les pistes que conformaven el node de GND, ja

#### Observació

Si en un moment donat decidim acabar de traçar una mateixa pista per una altra capa, podem fer el canvi utilitzant el que en el disseny de PCB s'anomena una via.

que els *pads* que havien d'estar-hi connectats ja han quedat connectats automàticament al pla de massa mitjançant una via.

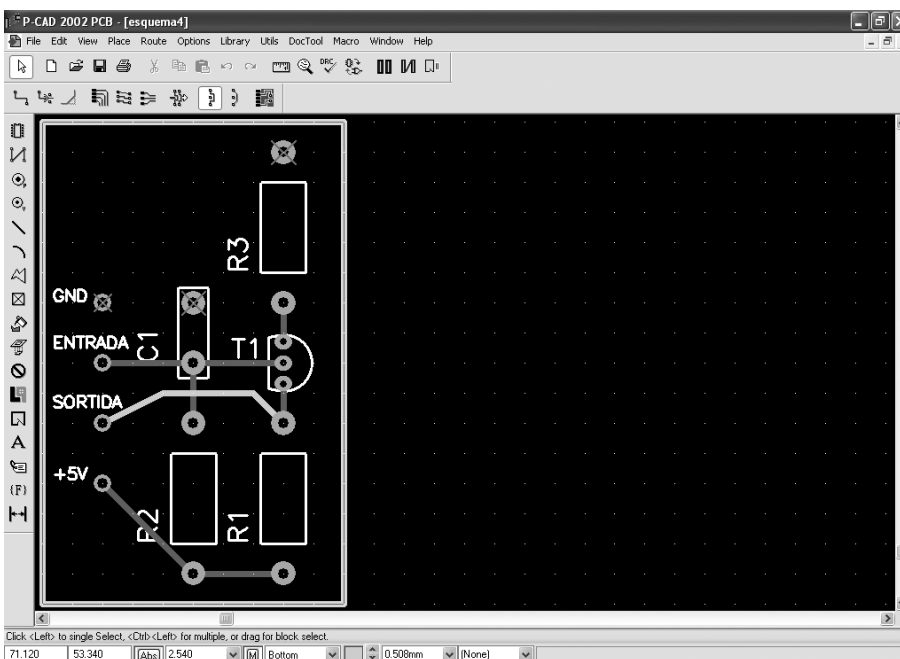
Figura 17. *Layout* afegint una capa interna connectada al node de GND



*Layout* afegint ara una capa interna connectada al node de GND (pla de massa). A la capa *top* no seran necessàries ara les pistes del node GND.

Fins i tot és habitual una vegada acabat el disseny cobrir de pla de massa tots aquells espais buits de la placa per tal de disminuir al màxim la impedància d'aquest node GND i assegurar-ne una tensió constant al llarg de tota la superfície de la placa. El resultat el podem veure a la figura 18.

Figura 18. Vista del *layout* si generem pla de massa per la capa *top*



Podem veure que els *pads* que formaven part d'aquest node GND hi queden també connectats i en canvi la resta de *pads* tan sols queden envoltats pel pla de massa.

A l'altra capa externa, *bottom*, podríem fer exactament el mateix i generar-hi un pla de massa o un pla d'alimentació. No visualitzem la imatge perquè en estar tot sobreposat no es veuria bé.

Una vegada acabat el *layout*, podríem passar ja a la fabricació i muntatge. Algunes eines EDA permeten també fer una simulació *postlayout* (després del *layout*), d'aquesta manera podríem veure si els plans que hem fet ens són necessaris o no valia la pena.

Hem de pensar que com més capes internes més cara serà la fabricació del PCB. Per tant, si no són necessàries no les hauríem de fer.

## 2.5. Generació de *gerbers* i altra documentació

En darrer lloc tan sols ens falta fabricar i muntar el PCB. Per aquesta darrera etapa l'entorn EDA ens ofereix la possibilitat de generar fitxers de documentació que faciliten aquestes tasques.

Cadascú utilitza un entorn EDA diferent (PCad, Eagle, Orcad, etc.) i tot sovint n'apareixen noves versions. Tanmateix, els fabricants necessiten una manera de poder analitzar els dissenys sigui quin sigui l'entorn EDA que utilitza cadascú. D'aquesta manera sorgeix la necessitat que tots els entorns siguin capaços d'extreure la documentació en formats estàndard i així poder treballar amb qualsevol fabricant.

El format més habitual és obtenir el que s'anomenen els *gerbers* del nostre disseny. Els *gerbers* són com una fotografia de cadascuna de les capes del nostre disseny. Així, amb un simple processament d'imatge, el fabricant pot analitzar la fotografia i procedir a la fabricació.

Els *gerbers* no són l'única documentació que necessitem extreure de la nostra eina de disseny. El muntador necessitarà un plànol de coordenades amb la posició de cada component per saber exactament on l'ha de soldar i haurà de saber també quins components i de quin valor ha de col·locar a cada lloc, etc. Per aquest motiu tots els entorns permeten extreure fitxers de text amb un format estandarditzat on els fabricants i muntadors podran trobar tota la informació necessària.

## 2.6. Procés d'insolació casolà

Tot i que l'opció més habitual és fer-nos fabricar la placa, potser en primers prototips preferim fer un parell de proves nosaltres mateixos abans d'assumir el cost econòmic que suposa la fabricació d'un PCB.

Si el PCB és a una sola cara o fins i tot a dues cares podem intentar fabricar-nos el PCB nosaltres mateixos encara que no disposem de gaires mitjans. Els passos a seguir consistirien a:

- insolar la placa,
- revelar la placa,
- i atacar-la.

Per això necessitem per exemple una base positiva de PCB, una insoladora (o alguna font de llum prou intensa), sosa càustica pel revelador, i, sulfurant, aigua oxigenada i aigua, per a l'atacador.

Per a insolar una placa positiva necessitem imprimir a partir de l'entorn EDA un fotolit (o dos si és a doble cara) en el qual tan sols hi hagi en fosc totes les zones on volem pistes conductores o *pads* conductors; la resta, ha de ser el més transparent possible.

La placa de PCB és una làmina de coure (o làmines si és a doble cara) en què el coure es troba cobert d'una resina sensible a la llum en la qual mitjançant el fotolit podem marcar per on han d'anar les nostres pistes i *pads*.

### Pas 1. Insolar la placa

Per començar desemboliquem la placa sensible positiva vigilant de no exposar-la gaire a la llum i hi enganxem el fotolit que correspongui a cada cara amb una miqueta de cinta adhesiva perquè no se'ns mogui durant el procés. Una vegada llest, ho posem a la insoladora (i si no en tenim, ho podem fer amb alguna làmpada una mica potent que faci llum blanca).

Passats uns quants minuts (entre 2 i 10 minuts, no podem concretar sense saber ni la intensitat de la llum ni el tipus de font exacta de llum) el fotolit hauria d'haver quedat imprès en la resina que cobreix la placa sensible.

### Pas 2. Revelar la placa

A continuació hem de revelar la placa per tal que aquesta resina resti tan sols en les zones on hi volem pistes i *pads* conductors. Aquesta resina cobreix tot el coure. Amb el revelat farem que es descobreixi el coure de totes aquelles zones que no volem que siguin conductores. En canvi la resina continuarà protegint el coure de les zones que finalment voldrem que siguin conductores.

Per a revelar una placa fotosensible positiva necessitem una dissolució de sosa càustica (la podem obtenir ja preparada en qualsevol botiga de components electrònics). Quan hi tinguem més pràctica i coneguem les concentracions ens la podem fer nosaltres mateixos. Ens sortirà més a compte, ja que podem obtenir de manera més econòmica la sosa càustica a qualsevol drogueria.

Si posem la placa en aquesta dissolució, en menys de 2 minuts hauria de desaparèixer la resina de les zones on el fotolit era transparent. Si en revelar ens desapareix tota la resina és símptoma que hem insolat massa estona o que les parts fosques del fotolit no eren prou opaques. Si la resina, per contra, no se'n va de cap lloc és que la dissolució està poc concentrada.

#### Base positiva de PCB

Si utilitzem una base positiva de PCB, totes aquelles zones que quedin exposades a la llum seran zones on el nostre PCB finalment no tindrà pistes de coure.

#### Proves prèvies

Tal com podeu veure, es tracta d'un procés en el qual haurem de fer moltes proves fins que no enganxem el punt exacte de tots els temps d'espera i concentracions de les dissolucions. Es recomana primer fer proves amb un trosset molt petit de placa i un petit fotolit.

#### Manipulació de productes químics

Quan treballem amb productes químics, protegiu-vos sempre adequadament la pell i no deueu d'inhal·lar els vapors que desprenen.

### Pas 3. Revelar la placa

Una vegada revelada la placa tan sols ens manca fer-ne l'atacat. L'atacat consisteix a fer saltar tot el coure que ha quedat desprotegit, és a dir, tot el coure de les zones on no volem que hi hagi conducció. Per això necessitarem submergir la placa en una dissolució a base de sulfurant, aigua oxigenada i aigua, a parts iguals.

Aneu molt alerta amb la dissolució a base de sulfurant, aigua oxigenada i aigua, ja que desprèn vapors molt tòxics.

De seguida veureu com el coure desprotegit comença a fer una escuma de color verd i es va desprendre. Quan veieu que comença a fer efecte l'atacat aneu traient la placa amb unes pinces (mai amb les mans) i aneu controlant el procés, ja que si ataqueu durant massa estona us saltarà el coure tant de les parts protegides com de les que no ho estan. En menys d'un minut tindreu ja la vostra placa.

Una vegada atacada la placa renteu-la bé sota l'aixeta. Les restes de resina de sobre les pistes les podeu treure amb una mica d'acetona o amb una mica d'alcohol.

Ja tindreu el PCB preparat per a ser foradat i muntat!

Si treballeu amb components *through-hole*, les pistes conductores i els *pads* estan per la cara contrària on vosaltres poseu el component. És a dir, si el component el posem a la cara *top*, les pistes i els *pads* els tindrem per la cara *bottom*, que és on haurem de soldar el terminal del component. Si el component és SMD, en canvi, ho tindrem tot a una mateixa cara.

Treballeu sempre en un lloc ben ventilat.

#### Atacat menys agressiu

Per fer un atacat no tan agressiu, ho podeu fer amb perclorur fèrric barrejat amb una mica d'aigua calenta. Llavors el procés pot trigar de 15 a 30 minuts.

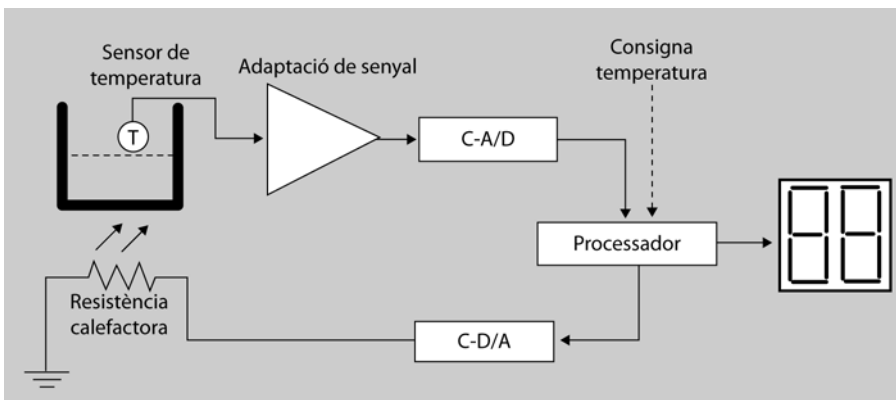
### 3. Convertidors analògic-digital i digital-analògic

Cada vegada tenim més tendència a treballar amb dades en format digital per tal de poder-les tractar amb processadors digitals. Prèviament, però, haurem de convertir els senyals analògics, que haguem obtingut (per exemple, de sensors) a senyals digitals. El dispositiu o component encarregat d'aquesta funció és el convertidor analògic-digital (C-A/D).

Així mateix, els resultats obtinguts pel processador digital els haurem de convertir de nou a un senyal analògic. El component que fa ara aquesta funció inversa a l'anterior és el convertidor digital-analògic (C-D/A).

En una simple aplicació d'instrumentació podem acabar d'entendre la necessitat d'aquests dos tipus de components.

Figura 19. Instrumentació per a controlar la temperatura d'un escalfador d'aigua



Aplicació d'instrumentació en la qual es vol controlar la temperatura d'un escalfador d'aigua. Volem visualitzar la temperatura i mantenir-la al voltant d'un nivell de consigna fixat per l'usuari.

En l'exemple de la figura 19, el sensor de temperatura obté una tensió funció de la temperatura. Aquesta tensió segurament l'haurem d'amplificar per tal de poder atacar l'entrada d'un convertidor analògic-digital (C-A/D), d'aquesta manera podrem tractar les dades de forma digital amb un processador. El processador compararà la temperatura amb la temperatura de consigna, visualitzarà la temperatura i en funció del resultat de la comparació decidirà si l'aigua s'ha d'escalfar més o menys i per fer-ho variarà la tensió que alimenta una resistència elèctrica. Aquesta tensió de sortida, però, l'haurem de convertir de nou a analògica per a poder alimentar la resistència calefactorsa.

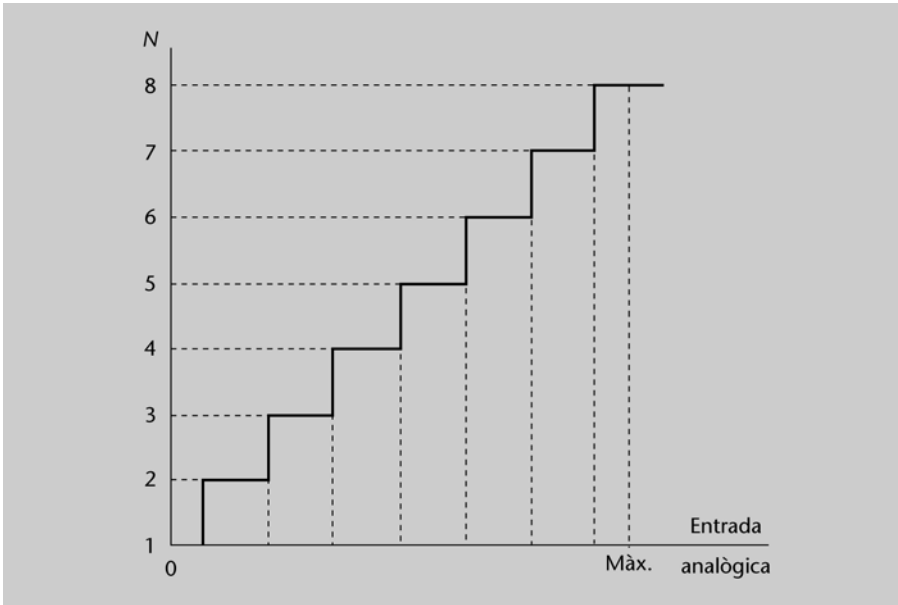
Per a entendre el funcionament dels convertidors i les seves limitacions haurem de veure primer les dues idees bàsiques en les quals es basa el pas d'anàlogic a digital: la quantificació i la codificació (subapartat 3.1). En els subapartats següents (3.2 i 3.3) analitzarem un parell de casos pràctics de cada tipus de component.

#### 3.1. La quantificació i la codificació

La quantificació d'un senyal elèctric consisteix a representar el seu valor en forma d'una sèrie finita de  $N$  nivells. Aquesta funció rep el nom de *quantificació*.

La quantificació pot ser uniforme quan, a cada nivell li correspon un marge d'igual amplitud, o pot no ser-ho. Ens centrarem només en l'estudi de quantificacions uniformes. A la figura 20 tenim un exemple de quantificació uniforme en 8 nivells diferents.

Figura 20. Quantificació uniforme amb  $N = 8$



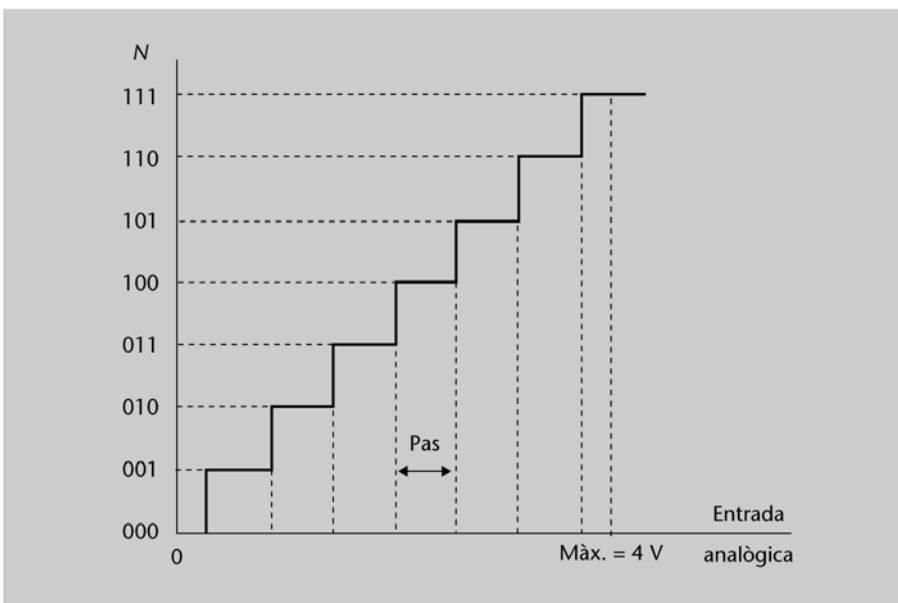
**Figura 20**  
Un senyal analògic d'entrada definit entre un valor 0 i Màx el quantifiquem en  $N$  nivells diferents de sortida. A l'eix horitzontal de la gràfica de la figura 20 hi podeu veure el senyal analògic a quantificar. A l'eix vertical la sortida en 8 nivells diferents.

La codificació consisteix a assignar a cadascun dels nivells obtinguts en la quantificació un codi digital únic. La codificació es caracteritzarà pel nombre de bits que utilitzem,  $n$ , per a obtenir un codi diferent per a cada nivell de quantificació. Així doncs s'haurà de complir que:

$$2^n \geq N \tag{18}$$

Podem veure a la figura 21 el resultat d'ajuntar la quantificació i la codificació.

Figura 21. Codificació en binari natural d'una quantificació en 8 nivells



**Figura 21**  
Fixeu-vos que en el cas de la figura 21 necessitem com a mínim 3 bits ( $n = 3$ ) per a poder obtenir els 8 nivells ( $N = 8$ ) de quantificació. El valor màxim que pot prendre la variable d'entrada analògica s'anomena fons d'escala (FS).

La unió d'aquestes dues operacions, ja sigui en un sentit o en l'altre, és el que volem que facin els convertidors A/D i D/A que analitzarem als subpartats 3.2. i 3.3 respectivament.  $n$  es correspondrà a la resolució dels convertidors (número de bits utilitzats per a la codificació digital).

Quan utilitzem un codi de  $n$  bits per a fer la conversió estem dividint el marge del senyal analògic d'entrada en  $2^n - 1$  intervals diferents (mireu la figura 21). Tenint present el valor de FS (fons d'escala, màxim valor del rang del senyal analògic que estem quantificant i codificant) podem calcular el pas de quantificació (la resolució en el procés de quantificació i codificació), que serà:

$$\text{Pas de quantificació} = \frac{\text{FS}}{2^n - 1} \quad (19)$$

Així doncs, si tornem a l'exemple de la figura 21, donada una tensió de FS de 4 V estariem parlant d'un pas de quantificació de 4/7 V.

Normalment com que  $2^n \gg 1$  aproximem directament el pas per  $\text{FS}/2^n$ .

El pas de quantificació és el que ens porta a saber si estem treballant amb un convertidor de prou resolució o per contra necessitem un convertidor de més elevada resolució ( $n$  major). Si per exemple amb el convertidor anterior volem detectar diferències entre mostres de menys de 0,1 V, necessitarem treballar amb una quantificació de més nivells ( $N$  més gran i per tant  $n$  també) fins que el pas de quantificació sigui de l'ordre de les variacions que volem detectar:

$$\frac{\text{FS}}{2^n - 1} < 0,1 \quad \frac{4}{2^n - 1} < 0,1 \rightarrow n \geq 6 \quad (20)$$

Necessitariem com a mínim un convertidor de 6 bits per a poder obtenir resolucions de 0,1 V amb un fons d'escala de 4 V.

Una vegada entès el concepte de quantificació i codificació ja podem passar a analitzar el comportament dels convertidors.


### 3.2. Convertidors analògic-digital

La feina d'un C-A/D consisteix a mostrejar una variable analògica, quantificar-la i codificar-la amb un cert nombre de bits.

Podem diferenciar dos tipus de convertidors:



- Directes: són aquells que quantifiquen i codifiquen directament la variable mostrejada.
- Indirectes: primer converteixen la variable mostrejada en algun altre tipus de variable (càrrega, freqüència, etc.) i és aquesta segona variable la que quantifiquen i mostregen.

Tot i així hi ha una sèrie de característiques que són comunes a qualsevol convertidor sigui del tipus que sigui. Les característiques principals que defineixen les prestacions d'un C-A/D són la resolució i la velocitat de conversió. Dos paràmetres que en general es troben en contraposició: és complicat obtenir a la vegada un convertidor d'alta resolució i alhora ràpid. 

La resolució, tal com hem vist al subapartat 3.1, ens ve donada pel nombre de bits que utilitzarem per a fer la codificació, ja que això ens determina el nombre de nivells en què podem quantificar. La velocitat ens indica el nombre de mostres d'entrada (i per tant conversions a la sortida) que és capaç de prendre el convertidor per unitat de temps (s'expressa normalment en mostres/segon).

Habitualment els convertidors de tipus directe es caracteritzen per poder treballar més de pressa i, en canvi, els indirectes, per obtenir resolucions més elevades.

D'altra banda, podem destacar també una sèrie de característiques de tipus més funcional:

- Trobem convertidors que mostregen senyals de tensió i uns altres que mostregen senyals de corrent.
- Tenim convertidors que treballen amb un únic canal analògic i convertidors que poden treballar amb diferents canals analògics a la vegada (llavors la velocitat quedaria dividida entre el nombre de canals operatius).
- Tenim convertidors amb sortida paral·lel (a cadascun dels  $n$  bits de la codificació els correspon un terminal de sortida diferent) o amb sortida sèrie (els  $n$  bits de cada mostra surten per un únic terminal seguint algun tipus de protocol sèrie).

Tots els convertidors, però, tindran una sèrie de terminals indispensables per al seu funcionament. Per una banda, tindrem sempre les tensions d'alimentació i la tensió que marca el fons d'escala (FS) del convertidor; per l'al-

tra, necessitem un senyal de rellotge per a fixar la freqüència de mostreig del senyal analògic d'entrada; i per últim necessitarem les entrades analògiques i les sortides digitals. Fins i tot podem trobar convertidors que interaccionen directament amb algun tipus d'etapa visualitzadora que són molt útils per a petites aplicacions de mesura.

A continuació veurem dos exemples de C-A/D compatibles amb etapes visualitzadores. Un serà de tipus directe, l'LM3914, i l'altre indirecte, l'ICL7106. Tots dos es fan servir molt en aplicacions de mesura dins del món de la instrumentació electrònica.

### 3.2.1. L'LM3914

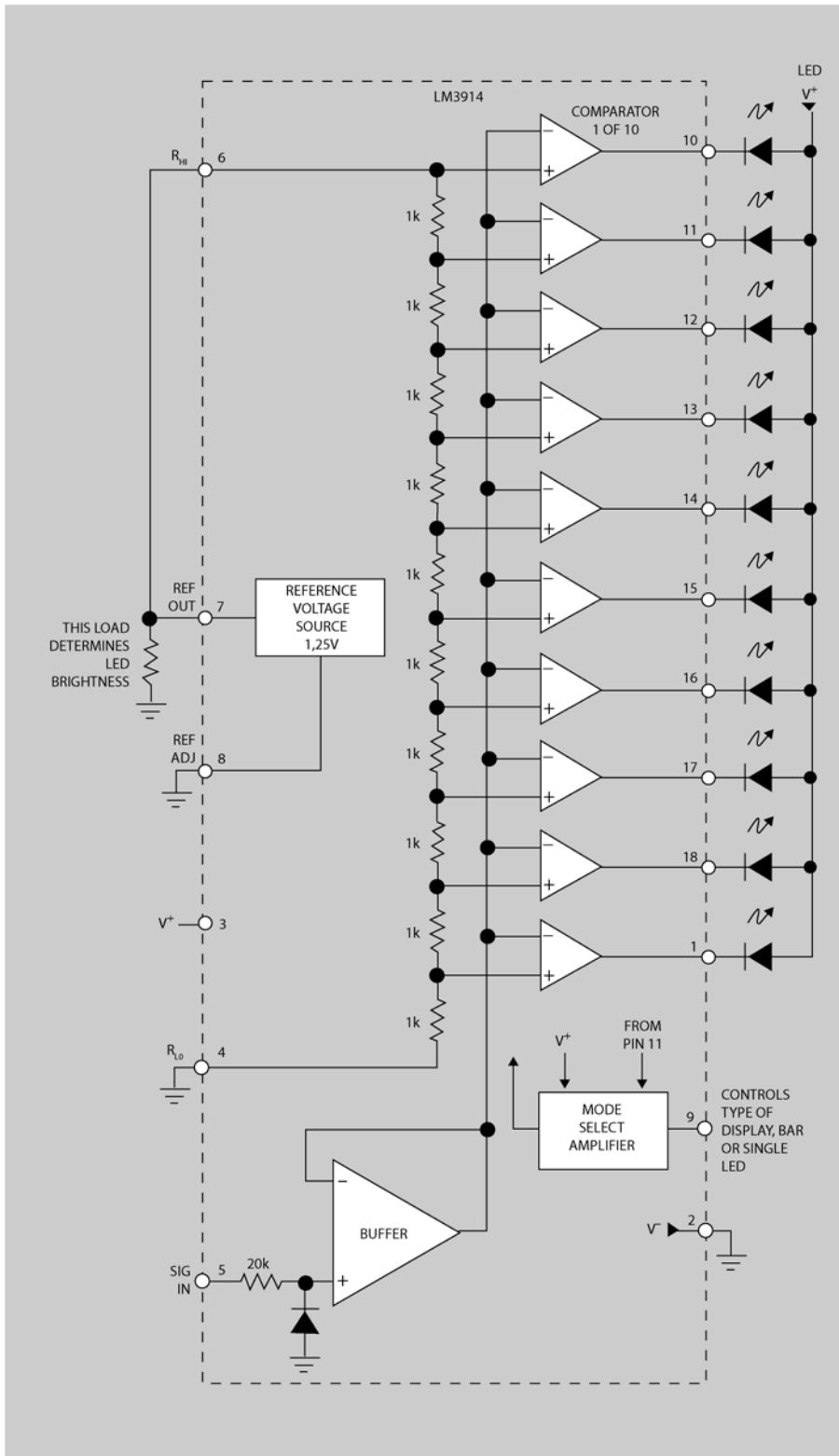
L'LM3914 és un C-A/D de tipus directe força ràpid però amb una resolució limitada. L'avantatge principal és que incorpora una barra de 10 *leds* per a visualitzar la sortida de la conversió. La visualització, però, no segueix un codi binari natural, sinó que tan sols hi ha 11 possibles combinacions: tots els *leds* apagats, només el primer encès, encès fins al segon, encès fins al tercer i així successivament fins a estar tots encesos.

És un convertidor molt útil en petites aplicacions de mesura en les quals la resolució no sigui un factor extremadament crític com per exemple la visualització del nivell de càrrega d'una bateria, la visualització del nivell d'un dipòsit, etc.

La il·luminació dels *leds* és regulable controlant-ne el corrent de manera que podem fer una millor gestió del consum del convertidor. Per a millorar-ne el consum tenim també l'opció de no visualitzar tots els *leds* fins a un cert nivell de la barra, sinó només el *led* més alt fins aquell nivell i deixar apagats tots els que queden per sota.

A la figura 22 tenim el diagrama de blocs del funcionament del convertidor. Fixeu-vos en la figura per a entendre com treballa el convertidor. L'entrada analògica del convertidor la tenim pel terminal número 5 (*SIG IN*). Aquesta tensió es compara amb cadascun dels nivells de tensió obtinguts amb un divisor de tensió format per 10 resistències iguals alimentades amb una certa tensió de referència (tensió de *FS* en aquest cas) pel terminal 6. Així doncs el primer *led* quedarà il·luminat sempre que la tensió d'entrada sigui superior a  $FS/10$ , el segon *led* quan la tensió d'entrada sigui superior a  $2 \cdot FS/10$  i així successivament fins a arribar al *led* de més pes el qual sols s'il·luminarà quan la tensió d'entrada (terminal 5) sigui superior a *FS* (terminal 6).

Figura 22. Esquema intern de funcionament de l'LM3914



Si utilitzéssim el convertidor, per exemple, per a determinar el nivell de càrrega d'una bateria, tindríem que tots els *leds* es trobarien encesos quan la bateria es trobés al 100% de càrrega. A mesura que la bateria s'anés descarregant els *leds* s'anirien apagant. Per exemple si tinguéssim els 4 *leds* baixos il·luminats voldria dir que el nivell de càrrega de la bateria es trobaria entre el 40% i el 50%. La tensió de fons d'escala hauria de ser poc més petita que la tensió de

#### Tensió d'una bateria

Tot i que en aquest mòdul no parlem de bateries, penseu que la tensió d'una bateria és només indicativa del seu nivell de càrrega, sempre que li estiguem exigint un cert corrent de descàrrega.

càrrega màxima de la bateria per tal de poder tenir il·luminats els 10 *leds* quan la bateria estigui pràcticament al 100%.

Aquest tipus de convertidors que obtenen el codi digital a partir de comparacions amb submúltiples de la tensió de fons d'escala s'anomenen convertidors paral·lels. Es caracteritzen per ser de tipus directe (no fan cap transformació al senyal d'entrada), són ràpids però per contra ens aporten baixes resolucions com hem pogut veure amb l'LM3914.

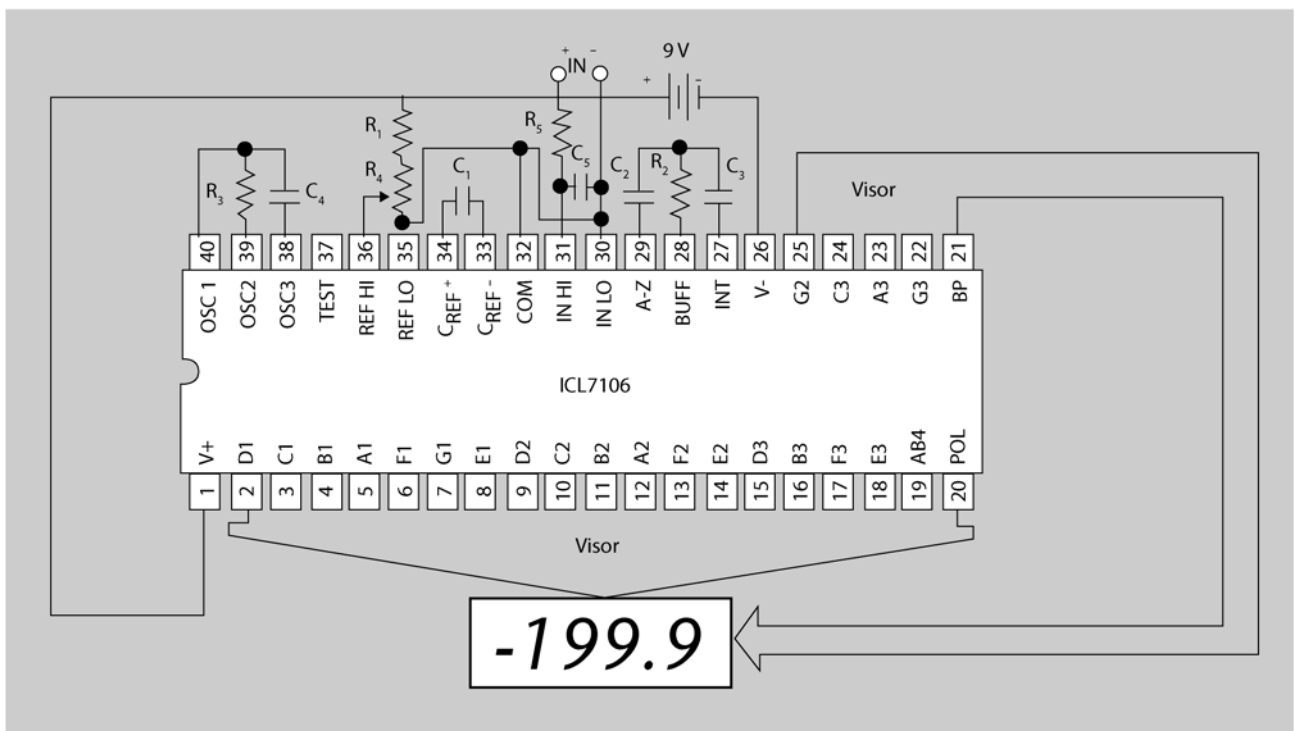
### 3.2.2. L'ICL7106-07

Un altre exemple de convertidor amb etapa de visualització, en aquest cas molt més lent que l'LM3914 però amb una resolució molt superior és l'ICL7106 o l'ICL7107.

L'ICL7106 fa la visualització de la dada digital a través d'una petita pantalla de cristall líquid (LCD) i en canvi l'ICL7107 utilitza una pantalla d'iguals característiques funcionals però en comptes de ser de cristall líquid els segments dels dígit són *leds*.

A la figura 23 podem veure quines són les característiques de la pantalla de l'ICL7106-07.

Figura 23. Imatge de la pantalla del convertidor ICL7106-07



Es tracta del que es diu una visualització en pantalla de 3 dígit i ½ ja que hi tenim tres dígit complets (unitats, desenes i centenes) i un 4t dígit (els milers) que tan sols pot valer 1 o 0 (el punt decimal a la figura 23 el podem posar on

vulguem; per tant, quan parlem d'unitats ens referim al dígit de més a la dreta i quan parlem de milers al dígit de més a l'esquerra). El valor dels dígits els controla el convertidor a partir de la tensió de fons d'escala i la tensió analògica d'entrada. El signe i el punt que separa els dígits els controla externament l'usuari independentment del resultat de la conversió.

L'ICL7106-07 és un convertidor indirecte, ja que transforma la tensió analògica d'entrada en la càrrega d'un condensador. El convertidor carrega un condensador al valor de la tensió analògica d'entrada a convertir, a partir d'aquí procedeix a la descàrrega del condensador comptant quant de temps triga a descarregar-se. Aquest recompte serà proporcional a la tensió que volíem convertir. Aquest tipus de convertidors es diuen integradors o de rampa.

Així doncs, deixant a part el signe i el punt, ambdós controlats per l'usuari, el convertidor pot extreure un valor entre 0 i 1999. Si a aquest valor li diem "lectura", el convertidor calcula el valor de la "lectura" seguint la següent fórmula:

$$Lectura = \frac{1999}{2} \frac{IN HI - IN LO}{REF HI - REF LO} \quad (21)$$

La diferència entre  $REF HI$  i  $REF LO$  és la tensió que el convertidor utilitza com a referència per a fer la conversió (veurem seguidament amb un exemple que en aquest cas no és el que en altres convertidors s'anomena FS, tot i que té també l'objectiu de marcar una referència per a la conversió).  $IN HI$  seria l'entrada analògica a convertir i  $IN LO$  és un senyal que ens serveix per a ajustar el 0 de la lectura tal com veurem seguidament.

Posem un exemple per a entendre com ajustariem tots aquests valors.

### Exemple

Imagineu que disposem d'un sensor mesurador de distàncies entre 0 i 2 metres. El sensor ens dóna una sortida ( $V_{out}$ ) lineal entre 0,5 V i 1,5 V en aquest rang de mesura. Volem visualitzar la mesura de longitud directament en el *display* de l'ICL7106-07.

$V_{out}$  es correspondria directament al terminal  $IN HI$  del convertidor. El primer pas que hem de fer sempre per a fixar la resta de terminals consisteix a ajustar el 0 de la lectura. A què ens referim per 0 de la lectura? Doncs a aquell valor de  $IN HI$  pel qual volem que la lectura de la pantalla sigui 0000.

Així doncs quan  $IN HI$  és 0,5 V, volem que la lectura = 0, per tant:

$$Lectura = \frac{1999}{2} \frac{0,5 - IN LO}{REF HI - REF LO} = 0 \rightarrow IN LO = 0,5 V \quad (22)$$

Tenim ja el valor del terminal  $IN LO$ . Ens manquen només els terminals de referència.

Per a fixar la tensió de  $REF HI$  i  $REF LO$  ens haurem de plantejar quina resolució volem que tingui la lectura de la longitud o dit d'una altra manera, amb quines unitats la volem visualitzar.

Si per exemple volem aprofitar al màxim les possibilitats del *display* tenim que la lectura estaria donada en mil·límetres (no podríem visualitzar del tot els 2 m, ja que el *display* arriba només fins a 1999). Però potser no volem tenir tanta resolució i decidim fer directament la lectura en cm, llavors quan estiguem mesurant 2 metres la lectura del *display* haurà de ser 0200 (si volem podem situar el punt entre les centenes i les desenes per a tenir la lectura en metres).

### Enllaç recomanat

Per a més detalls podeu consultar el *datasheet* a la pàgina web d'Intersil:  
<http://www.intersil.com>.

Si decidim fer-ho de la primera manera tindriem que per a una tensió IN HI d'1,5V volem que la lectura sigui 1999 (ja que com hem dit 2000 no el podem visualitzar):

$$Lectura = \frac{1999}{2} \frac{1,5 - 0,5}{REF HI - REF LO} = 1999 \rightarrow REF HI - REF LO = 0,5 V \quad (23)$$

En canvi, si ho fem de la segona manera tindriem:

$$Lectura = \frac{1999}{2} \frac{1,5 - 0,5}{REF HI - REF LO} = 200 \rightarrow REF HI - REF LO \approx 5 V \quad (24)$$

Fixeu-vos que el càlcul de  $REF HI - REF LO$  dependrà de la manera com vulguem fer la mesura. Habitualment voldrem aprofitar al màxim les prestacions de la pantalla; per tant, probablement hauríem escollit la primera opció i hauríem situat el punt entre els milers i les centenes per a tenir la lectura en metres.

$REF HI - REF LO$  es correspon a la diferència de potencial entre dos terminals. Podem fixar la tensió entre ells amb un divisor amb resistiu.

### 3.3. Convertidors digital-analògic

La funció del C-D/A és just la contrària de la que hem vist fins ara: a partir d'una mostra digital de  $n$  bits extraurà una tensió discreta (o corrent) obtinguda a partir d'una certa referència de tensió (o corrent). El convertidor, per tant, convertirà un codi binari en un senyal elèctric.

El procés de conversió consistirà en aquest cas a fer una ponderació de cada bit del codi i sumar el resultat de totes les ponderacions multiplicades pel valor del bit associat. Per tant la sortida del convertidor és única per cada codi d'entrada associat, per això diem que extreu una tensió (o un corrent) discreta.

Per exemple, si ens tornem a fixar amb el cas de la quantificació a 8 nivells utilitzant un codi de 3 bits que hem utilitzat al subapartat 3.1, podríem fer ara el procés invers.

Hem calculat que amb una tensió de FS de 4 V el pas de quantificació era de  $4/7$  V. Assignariem al bit de menys pes una ponderació de  $4/7$ , al segon bit una ponderació del doble,  $8/7$ , i al tercer bit una ponderació del doble de l'anterior,  $16/7$ . Donada aquesta ponderació fixeu-vos què faria el C-D/A amb tensió de referència a 4 V i  $n = 3$  bits:

$$b_2 b_1 b_0: b_2 \cdot 16/7 + b_1 \cdot 8/7 + b_0 \cdot 4/7 V$$

$$000: 0 \cdot 16/7 + 0 \cdot 8/7 + 0 \cdot 4/7 = 0 V$$

$$001: 0 \cdot 16/7 + 0 \cdot 8/7 + 1 \cdot 4/7 = 4/7 V$$

$$010: 0 \cdot 16/7 + 1 \cdot 8/7 + 0 \cdot 4/7 = 8/7 V$$

$$011: 0 \cdot 16/7 + 1 \cdot 8/7 + 1 \cdot 4/7 = 12/7 V$$

$$100: 1 \cdot 16/7 + 0 \cdot 8/7 + 0 \cdot 4/7 = 16/7 V$$

$$101: 1 \cdot 16/7 + 0 \cdot 8/7 + 1 \cdot 4/7 = 20/7 V$$

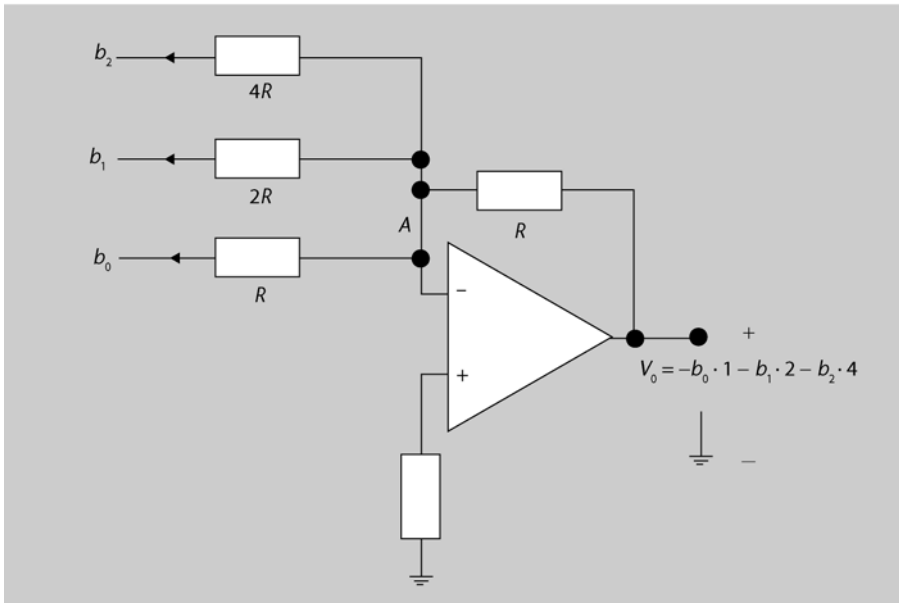
$$110: 1 \cdot 16/7 + 1 \cdot 8/7 + 0 \cdot 4/7 = 24/7 V$$

$$111: 1 \cdot 16/7 + 1 \cdot 8/7 + 1 \cdot 4/7 = 28/7 V = 4 V$$

A cada codi li correspon un únic nivell discret de sortida.

En el cas dels C-D/A és molt típic el disseny en elements ponderats. L'element ponderat pot ser, per exemple, una resistència. Analitzem ara el cas de la figura 24 per a entendre millor què volem dir amb això.

Figura 24. C-D/A de 3 bits amb tensió de referència a 4 V



Cada entrada ( $b_i$ ) prendria un valor de 0 V quan el bit corresponent és 0 i de  $-4/7$  V quan el bit corresponent fos 1.

Si tenim la ponderació de resistències de la figura 24 tindriem a la sortida les tensions que acabem de calcular.

Imagineu que quan un bit és 1 lògic l'entrada corresponent val  $-4/7$  V i quan és 0 val 0 V. Per la primera resistència de valor  $R$  estaria circulant un corrent (en el sentit que marca la figura) de valor  $-b_0/R$ . Per la resistència de valor  $2R$  tindriem un corrent  $-b_1/2R$  i per la resistència de valor  $4R$  tindriem un corrent  $-b_2/4R$ . Tenint present que per a l'anàlisi de l'amplificador operacional realimentat negativament al node A tindrem una tensió de 0 V, llavors la tensió de sortida és la suma d'aquests 3 corrents multiplicats per la resistència de realimentació  $R$ . La tensió de sortida és, doncs, la indicada a la figura. Fixeu-vos si substituïu per qualsevol codi de la taula que la sortida de tensió és l'esperada.

Aquesta ponderació es podria fer també amb d'altres elements com per exemple fonts de corrent. Sigui quin sigui l'element utilitzat ens trobem davant del cas més simple de C-D/A.

Tot i que no segueix exactament l'esquema indicat un bon exemple de C-D/A és el DAC08. El DAC08, fabricat per Analog Devices, és un C-D/A d'ús molt estès en aplicacions de relativament baixa resolució. Accepta múltiples tecnologies digitals d'entrada i el seu cost és relativament baix, d'aquí la gran extensió del seu ús.

Cal destacar que a diferència dels C-A/D, ara no necessitem mostrejar cap senyal i per tant els convertidors funcionen sense senyal de rellotge. Així doncs, en el cas del C-D/A més simple els terminals quedaran reduïts a les tensions (o corrents) de referència, les  $n$  entrades digitals i la sortida analògica en tensió o corrent.

Recordeu que un amplificador operacional ideal té una impedància d'entrada infinita i tenim un curtcircuit virtual entre el pin inversor i el no inversor.

## 4. Disseny de circuits amb dispositius GAL

Al llarg dels estudis s'han introduït els dispositius d'arquitectura configurable; recordeu que són aquells circuits integrats que canvien la seva arquitectura interna per mitjà d'una programació que físicament altera les connexions internes, com per exemple les FPGA (*field programmable gate array*), o les GAL (*generic array logic*, formació lògica genèrica).

Són, doncs, una manera sovint econòmica i ràpida de crear un maquinari específic per una aplicació electrònica. Un exemple d'aquests tipus de components d'ús molt simple i estès són, precisament, els dispositius GAL, els quals ens donen moltes possibilitats de disseny.

En aquest apartat repassarem primer de tot (subapartat 4.1) quin és el principi de funcionament d'aquests dispositius d'arquitectura configurable i a continuació analitzarem les etapes a seguir durant el procediment de disseny amb dispositius GAL (subapartat 4.2).

Al subapartat 4.3 trobareu una introducció a l'eina ispLEVER la qual ens permet el desenvolupament amb dispositius del tipus GAL.

### 4.1. Principi de funcionament dels dispositius GAL

L'origen dels dispositius GAL es remunta a l'ús de les memòries ROM per a crear circuits combinatoris, de tal manera que les línies d'adreça de la memòria s'associen als senyals d'entrada del sistema combinatori i a cadascuna d'aquestes adreces s'emmagatzema el valor de sortida que volem que presenti el sistema combinatori. Podem implementar amb cada memòria tants sistemes combinatoris com nombre de bits tingui cadascuna de les paraules emmagatzemades.

L'estructura interna d'una ROM equival a una matriu de portes AND i una matriu de portes OR (figura 25). Aquesta darrera és configurable per mitjà de la seva programació.

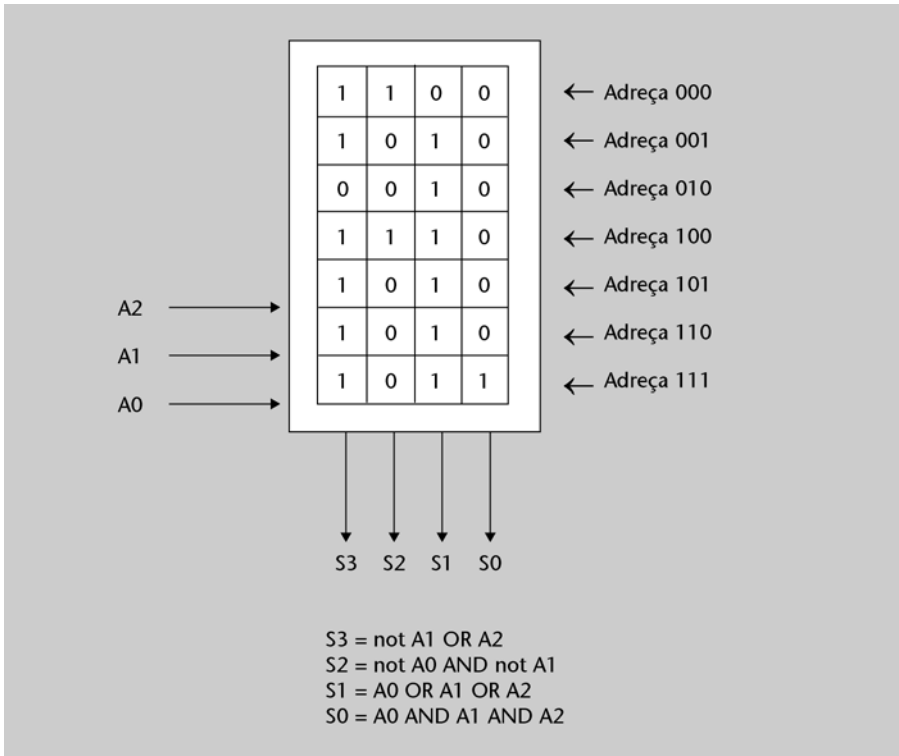
L'origen dels dispositius configurables (figura 26) el trobem als anys 70 quan l'empresa Monolithic Memories Inc. (MMI) va prendre l'estructura de les memòries ROM per a fer configurable la matriu de portes AND en comptes de la matriu de portes OR, invent que es va anomenar PAL. Posteriorment l'empresa Lattice va desenvolupar les mateixes estructures però amb l'ús de transistors MOS de porta flotant per tal de fer-les reprogramables, és a dir, que permetés anar canviant la programació, invent que es va anomenar GAL.

PAL prové de *programmable array logic*, mentre que GAL prové de *generic array logic*.



Arran d'aquest fet MMI i Lattice s'enfrontaren pels drets d'invenció, demanda que finalment va guanyar la primera empresa. Posteriorment MMI va ser adquirida per la multinacional AMD i curiosament al cap de molts anys, el 1999, Lattice va comprar la divisió d'AMD corresponent a l'original MMI. Actualment Lattice és qui comercialitza els dispositius GAL.

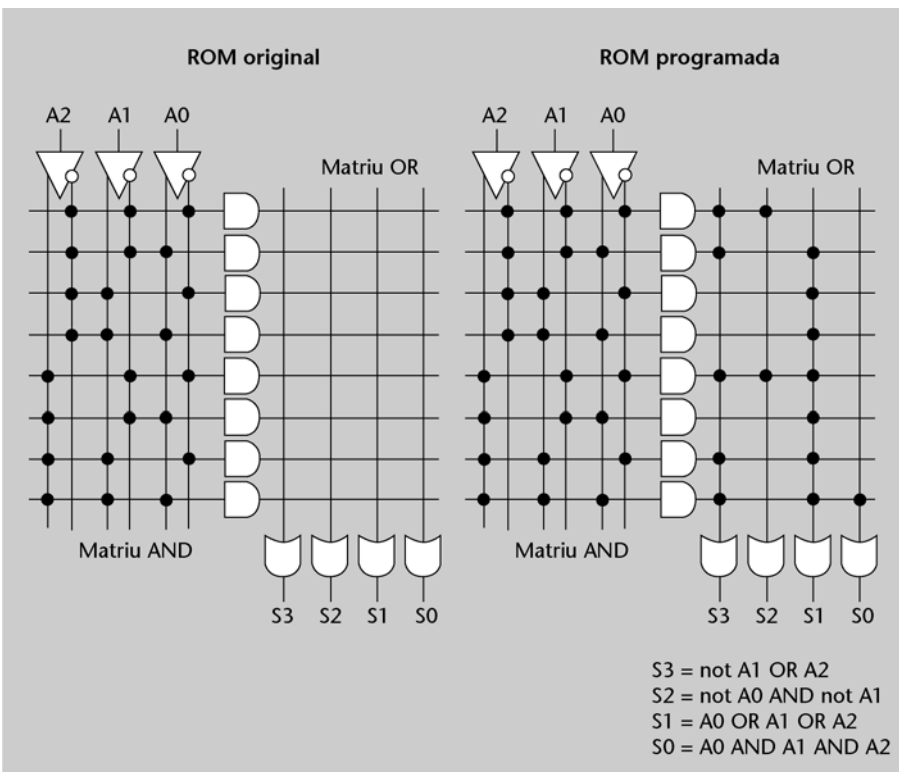
Figura 25. Exemple d'ús d'una memòria ROM de paraules de 4 bits amb tres entrades (8 adreces) per a implementar fins a 4 sistemes combinatoris



**Figura 25**

A la figura 25, podem observar com el bit de menor pes de cada paraula emmagatzemada es correspon amb la combinació de fer una AND dels tres bits de l'adreça d'entrada, i dóna lloc en aquest cas a la sortida S0. De la mateixa manera el segon bit de cada paraula emmagatzemada es correspon amb la combinació de fer una OR, dóna lloc a la sortida S1, i així successivament per a la resta de sortides.

Figura 26. Esquema de funcionament intern d'una memòria ROM

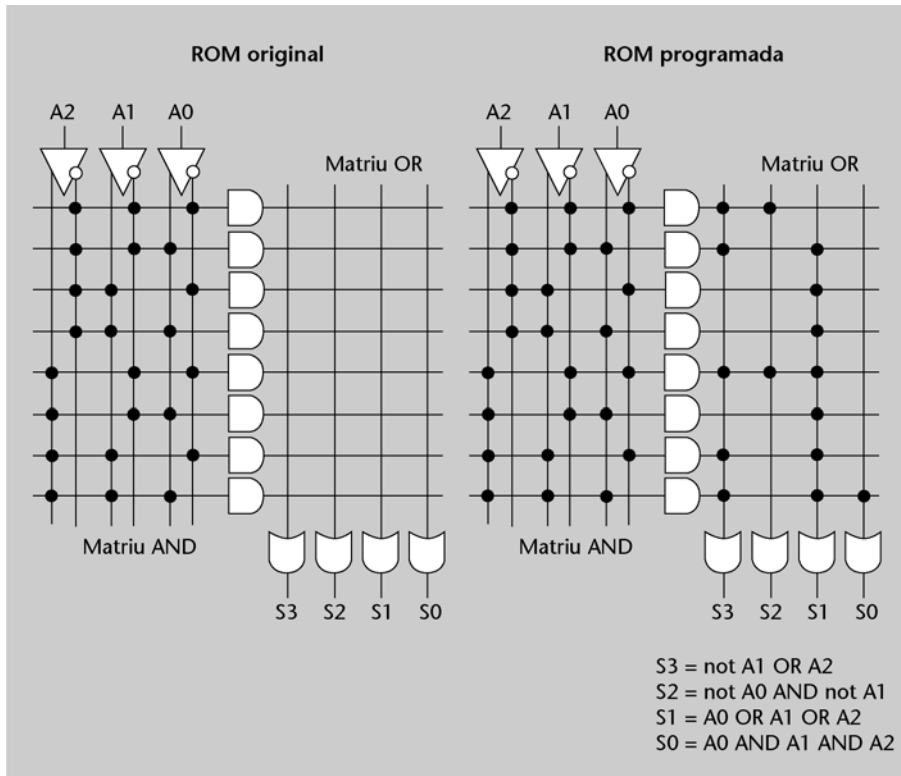


**Figura 26**

A la part esquerra de la figura 26 veiem una memòria sense gravar amb la matriu fixa de portes AND que activa, segons els senyals d'entrada, una de les línies de la matriu de portes OR de la dreta, la qual és programable. A la part de la dreta de la figura 26, veiem una memòria ROM gravada. Observeu com aleshores s'activen algunes de les interseccions de la matriu de portes OR a fi d'activar els senyals de sortida segons els valors d'entrada.

Els dispositius GAL que trobem actualment al mercat no estan formats tan sols per simples matrius de portes AND i portes OR, sinó que a continuació de les portes OR també s'hi han incorporat biestables, portes XOR, multiplexors, etc., fet que permet crear no tan sols sistemes combinatoris sinó fins i tot simples sistemes seqüencials. Tots aquests elements addicionals que s'han afegit als dispositius GAL constitueixen uns blocs que s'anomenen macrocel·les que determinen el grau de complexitat dels dispositius (figura 27).

Figura 27. Estructura de la macrocel·la del dispositiu GAL 18V10



Estructura de la macrocel·la del dispositiu GAL 18V10 on s'observa com a continuació de la porta OR s'ha incorporat tant un biestable com multiplexors per a facilitar la implementació de sistemes seqüencials.

Actualment Lattice comercialitza dispositius GAL amb diferents prestacions:

- Alimentacions estàndards de 5V o bé alimentacions de 3,3V.
- 8, 10 o 12 macrocel·les per dispositiu.
- Possibilitat de signatura electrònica del dispositiu.
- Possibilitat de regravar el dispositiu sense haver-lo d'extreure de la placa en la qual està funcionant (ISP, *in-system programming*).
- Etc.

Aquestes prestacions permeten implementar aplicacions en les quals intervenen màquines d'estats, simples processadors gràfics, controladors d'accés directe a memòria (DMA, *direct memory access*, accés directe a memòria), etc. Aquestes estructures han anat evolucionant fins al desenvolupament dels anomenats CPLD (*complex programmable logic device*, dispositiu lògic complex programable), que permeten la implementació de sistemes molt sofisticats.

#### Signatura electrònica

La signatura electrònica consisteix en uns quants bytes on l'usuari pot emmagatzemar informació referent al dispositiu resultant que faciliti la seva identificació, com per exemple l'identificador del programador, la data de programació, el número de revisió, etc.

## 4.2. Etapes a seguir en el procés de disseny d'un sistema amb dispositius GAL

La implementació d'un sistema mitjançant un dispositiu GAL és força simple i es limita a tres processos bàsics, als quals se n'hi poden afegir d'altres que són accessoris. En general aquests processos s'executen dins d'una mateixa aplicació informàtica en un ordinador personal. Els passos són els següents:

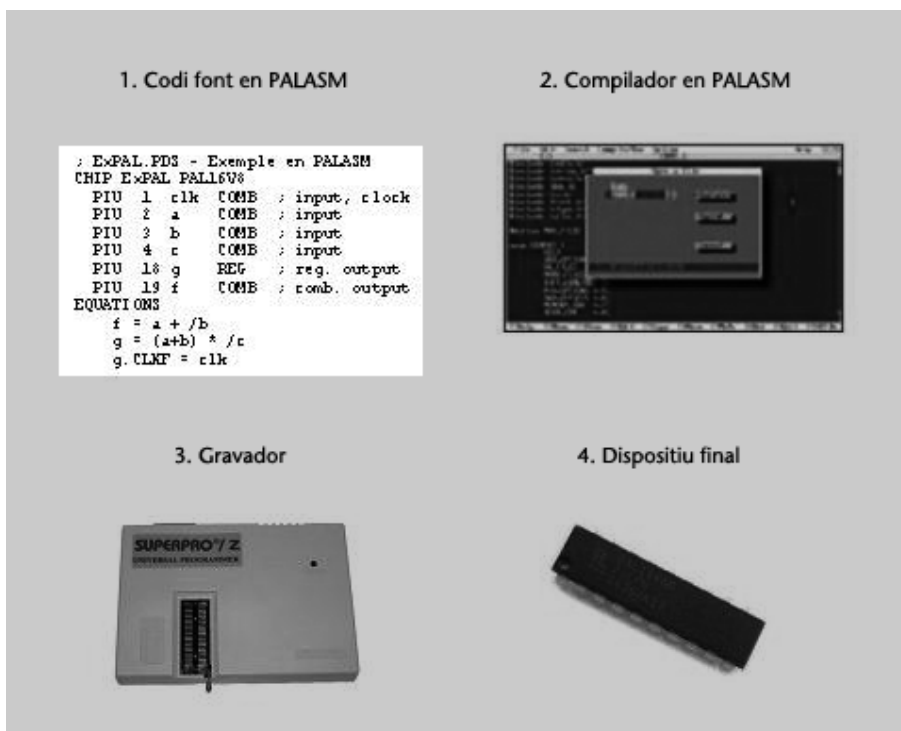
- 1) Primer de tot s'escriu el sistema que es vol implementar, en general amb qualsevol processador de textos, seguint un llenguatge descriptor específic.
- 2) Posteriorment es compila el codi escrit per a generar un fitxer binari JEDEC.
- 3) Finalment es grava al dispositiu GAL per mitjà de la transmissió del fitxer JEDEC a través d'un dels ports de l'ordinador, cap a un gravador a sobre del qual es posa el dispositiu GAL.

Quant als llenguatges caldria comentar que aquests han anat evolucionant al llarg dels anys. Així, per exemple, el primer llenguatge que es desenvolupà per tal de facilitar la programació d'aquests dispositius va ser el PALASM, creat el 1978 per John Birkner a MMI. Aquest llenguatge encara es pot emprar avui en dia juntament amb l'aplicació anomenada PALASM4 per a gravar dispositius GAL (figura 28). Posteriorment aparegueren altres llenguatges com AMAZE (1982), CUPL (1983) o ABEL (1983) amb prestacions similars al PALASM. Avui en dia els dispositius GAL es defineixen habitualment amb llenguatge ABEL o bé emprant llenguatges estructurats estandarditzats, com són el VHDL o Verilog.

### Fitxers JEDEC

Els fitxers JEDEC, sovint amb extensió .JED, són fitxers que segueixen un estàndard promogut per l'organització JEDEC, per mitjà del qual es descriuen les connexions internes que s'han d'establir en els dispositius electrònics programables.

Figura 28. Exemple del procés d'implementació d'un dispositiu GAL mitjançant programació PALASM



Les aplicacions informàtiques més sofisticades han inclòs prestacions addicionals a les tres etapes anteriorment esmentades, tals com:

- Simuladors lògics previs a la gravació per tal de verificar la correctesa del disseny.
- Processos de minimització lògica per tal d'optimitzar els recursos necessaris a l'hora de fer la implementació.
- Mòduls de comprovació d'errors.
- Etc.

### Exemple de minimització lògica

Suposem que volem implementar un sistema combinatori amb quatre entrades i una sortida on el senyal de sortida ha de prendre el valor 1 quan l'entrada pren els valors decimals 1, 3, 4, 5, 7, 9, 10, 11 o bé 15.

Una manera de fer-ho seria implementar una suma de productes on cadascun d'aquests productes fos una de les anteriors combinacions d'entrades que dona lloc a l'1 lògic. Així, doncs, si les entrades fossin  $x_3$ ,  $x_2$ ,  $x_1$  i  $x_0$  podríem implementar que la sortida fos  $y = \underline{x_3}x_2x_1x_0 + x_3x_2x_1x_0 + \underline{x_3}x_2x_1\underline{x_0} + \underline{x_3}x_2x_1x_0 + x_3x_2x_1x_0 + x_3x_2x_1x_0 + x_3x_2x_1x_0 + \underline{x_3}x_2x_1x_0 + x_3x_2x_1x_0 + x_3x_2x_1x_0 + \underline{x_3}x_2x_1x_0 + \underline{x_3}x_2x_1x_0$  (els termes que estan subratllats indiquen que estan negats). Ara bé, si prèviament realitzem una minimització lògica, per exemple mitjançant algorismes de Karnaugh o Quine-McCluskey, veuríem que el mateix sistema es pot implementar amb l'expressió següent:  $y = x_1x_0 + \underline{x_2}x_0 + x_3x_2x_0 + x_3x_2x_1$  fet que comporta una reducció de les portes lògiques necessàries.

Entre les diferents aplicacions informàtiques desenvolupades pel disseny específic de dispositius GAL, la més emprada avui en dia és l'aplicació de distribució gratuïta que la mateixa empresa Lattice ofereix a través de la seva pàgina d'Internet, anomenada ispLEVER Classic.

#### Enllaç recomanat

Visiteu <http://www.latticesemi.com> on, a més a més, podreu trobar manuals, exemples d'aplicacions, anàlisis comparatives, etc.

### 4.3. Introducció al desenvolupament amb ispLEVER

L'empresa Lattice ha desenvolupat una aplicació anomenada ispLEVER que permet treballar amb qualsevol dels seus dispositius electrònics programables, inclosos els dispositius GAL.

Aquesta aplicació és modular, de tal manera que un es pot instal·lar des del mòdul més elemental que és de distribució gratuïta i que tan sols permet desenvolupaments amb dispositius elementals com són els dispositius GAL; fins a mòduls més complexos, tant de simulació com de síntesi avançada, per dispositius molt més sofisticats com són les FPGA. En aquest subapartat ens centrarem en la part més bàsica i us adrecem a la pàgina web de Lattice per aprofundir en el tema.

Per a poder-ho practicar cal anar a la pàgina web de Lattice i descarregar-se i instal·lar-se l'aplicació ispLEVER Classic. També caldrà demanar a la pàgina web una llicència de distribució gratuïta.

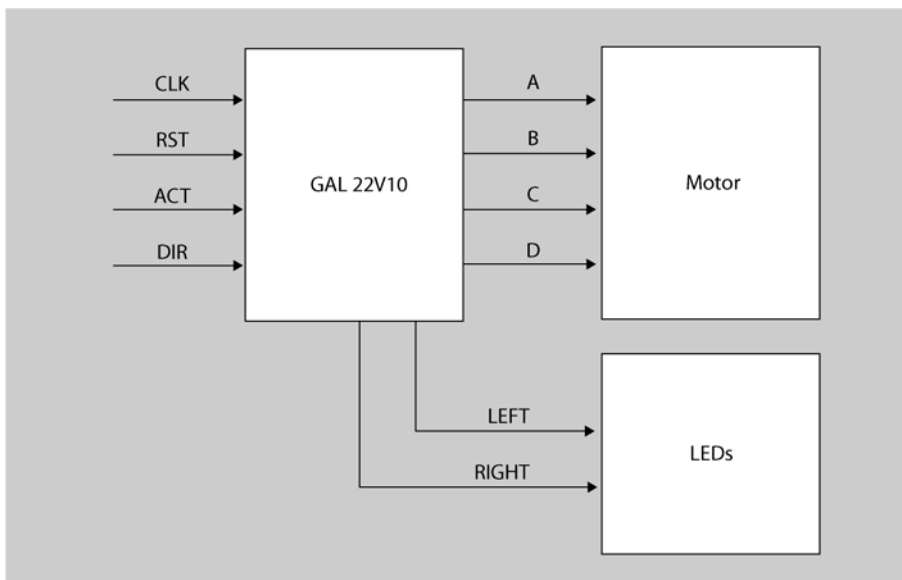
Com a exemple, anem a suposar que volem implementar el control d'un motor pas a pas mitjançant un dispositiu GAL del model 22V10. Les especificacions són les següents (figura 29):

- El dispositiu 22V10 té quatre entrades:
  - CLK que és el rellotge del sistema.
  - RST que reinicialitza el sistema.
  - ACT que valdrà 1 si volem moure el motor o bé 0 si el volem aturar.
  - DIR que valdrà 1 si el volem moure cap a l'esquerra o bé 0 si el volem moure cap a la dreta.
- El dispositiu 22V10 té sis sortides. N'hi ha quatre que són A, B, C i D, que s'envien al motor, mentre que les dues restants són LEFT i RIGHT que s'utilitzen per a indicar si el motor està activat i es mou cap a l'esquerra o bé cap a la dreta, respectivament. Se suposa que aquests dos darrers senyals activen uns *leds* per poder-ho visualitzar.
- El motor pas a pas té quatre entrades A, B, C i D, de tal manera que per a moure el motor anirem aplicant de manera seqüencial el valor 1 lògic a una d'aquestes quatre entrades, mentre les tres restants es troben a 0 lògic. Concretament, si el volem moure en el sentit de l'esquerra caldrà activar un 1 lògic seguint la seqüència ACBDACBDA, etc. mentre que per moure'l en sentit contrari caldrà seguir la seqüència inversa, és a dir, DBCADBCA, etc.

#### Observació

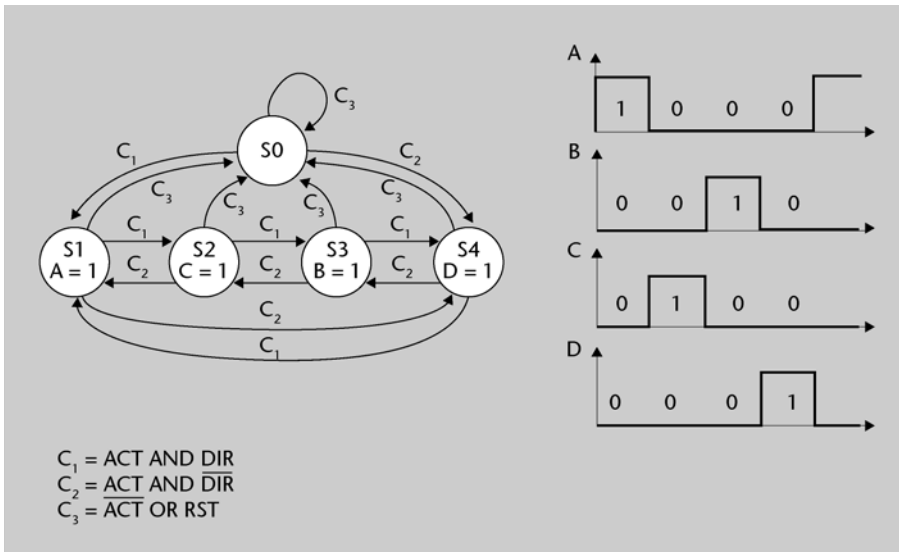
En el nostre exemple el motor pas a pas internament està format per quatre bobinats (fases), que s'han d'excitar alternadament i no pas en ordre, per això la seqüència dels senyals d'entrada del motor és ACBD o bé DBCA.

Figura 29. Esquema de l'aplicació de l'exemple



Per tal de desenvolupar el sistema es proposa implementar la màquina d'estats que es mostra a la figura 30.

Figura 30. Màquina d'estats proposada per a satisfer les especificacions



En els estats S1 a S4 s'activarà tan sols amb un 1 lògic una de les sortides, mentre la resta es mantenen a 0 lògic. El senyal DIR determina si la seqüència és ACBD o bé DBCA, de tal manera que si se satisfà la condició 1 ( $C_1$ ) el motor gira en un sentit, mentre que si se satisfà la condició 2 ( $C_2$ ) gira en sentit contrari. La tercera condició ( $C_3$ ) atura el motor o bé el posiciona a l'estat inicial en cas de *Reset*.

Abans de res haurem d'escriure la solució proposada, tot detallant-la amb qualsevol llenguatge de descripció que admeti ispLEVER. En aquest cas l'escriurem en llenguatge ABEL, tot i que també admet descripcions en forma d'esquemàtics o en VHDL:

```

1  MODULE StepMotor
2  TITLE 'Example with GAL22V10'
3
4  DECLARATIONS
5  " Input and output pins
6  CLK          PIU 1;                " Clock
7  RST,ACT,DIR  PIU 2,3,4;          " Reset, Active,
                                   " Direction
8  LEFT,RIGHT  PIU 23,22           istype 'com'; " LEDs
9  D,C,B,A     PIU 21,20,19,18     istype 'com'; " Motor signals
10 Q2,Q1,Q0    PIU 17,16,15        istype 'reg'; " State machine
11 " Other declarations
12 REGEST=[Q2,Q1,Q0];              " Definition of registers
13 S0=^B000;                        " Binary encoded state 0
14 S1=^B001;                        " Binary encoded state 1
15 S2=^B010;                        " Binary encoded state 2
16 S3=^B011;                        " Binary encoded state 3
17 S4=^B100;                        " Binary encoded state 4
18 X=.X.;                          " Don't care symbol
19
20 EQUATIONS
21 " Clock and reset
22 REGEST.CLK=CLK;
23 REGEST.ACLR=RST;
24 " State machine for A,B,C and D signals
25 STATE_DIAGRAM REGEST
26 STATE S0:
27 A=0; B=0; C=0; D=0;
28 IF (ACT&DIR) THEN S1 ELSE
29 IF (ACT&!DIR) THEN S4 ELSE S0;
30 STATE S1:
31 A=1; B=0; C=0; D=0;
32 IF (ACT&DIR) THEN S2 ELSE
33 IF (ACT&!DIR) THEN S4 ELSE S0;
34 STATE S2:
35 A=0; B=0; C=1; D=0;
36 IF (ACT&DIR) THEN S3 ELSE
37 IF (ACT&!DIR) THEN S1 ELSE S0;
38 STATE S3:

```

Podem trobar diversos manuals sobre el llenguatge ABEL per Internet.

```

39  A=0; B=1; C=0; D=0;
40  IF (ACT&DIR) THEN S4 ELSE
41  IF (ACT&!DIR) THEN S2 ELSE S0;
42  STATE S4:
43  A=0; B=0; C=0; D=1;
44  IF (ACT&DIR) THEN S1 ELSE
45  IF (ACT&!DIR) THEN S3 ELSE S0;
46  TRUTH_TABLE ([ACT,DIR] -> [LEFT,RIGHT])
47  [0,X] -> [0,0]; " LEDs off
48  [1,1] -> [1,0]; " LEFT on
49  [1,0] -> [0,1]; " RIGHT on
50
51  END StepMotor;

```

La línia 1 indica amb la directiva *MODULE* el nom de l'aplicació i la línia 2 indica el títol.

La línia 4 és una directiva *DECLARATIONS* per a indicar que a continuació es detallen constants, pius, etc., és a dir tot un conjunt de declaracions.

La línia 5 és de comentari en iniciar-se amb `"`.

De la línia 6 fins a la línia 10 es descriuen els pius d'entrada i de sortida. En el cas de les sortides s'indica si aquesta és combinatòria (*com*) o bé és la sortida d'un registre (*reg*). Més detalladament els de la línia 10 són les sortides de tres registres que utilitzarem per a implementar els estats de la màquina d'estats.

La línia 12 és una manera d'agrupar qualsevol acció sobre els registres *Q0*, *Q1* i *Q2* sota el nom de *REGEST*.

De la línia 13 fins a la línia 17 es defineixen els valors binaris corresponents als estats *S0*, *S1*, *S2*, *S3* i *S4*.

La línia 18 és la definició del símbol "indiferent".

La línia 20 indica que a continuació s'iniciarà la definició de les equacions. Per a aquestes podrem emprar les declaracions que prèviament hem definit.

La línia 22 indica que tots els senyals de rellotge de tots els registres es connectin al senyal *CLK*. Passa el mateix a la línia 21 per al senyal de posada a zero dels registres amb el senyal *RST*.

La línia 25 indica que s'iniciarà la definició d'una màquina d'estats amb els registres *REGEST*.

De la línia 26 fins a la línia 45 es defineixen els estats. En cadascun s'indiquen les sortides i les condicions de salt cap a altres estats.

La línia 46 indica que a continuació es definirà una taula de veritat que tindrà com a entrades *ACT* i *DIR* i que tindrà com a sortides *LEFT* i *RIGHT*. De les línies 47 a 49 es defineixen els valors d'aquesta taula de veritat.

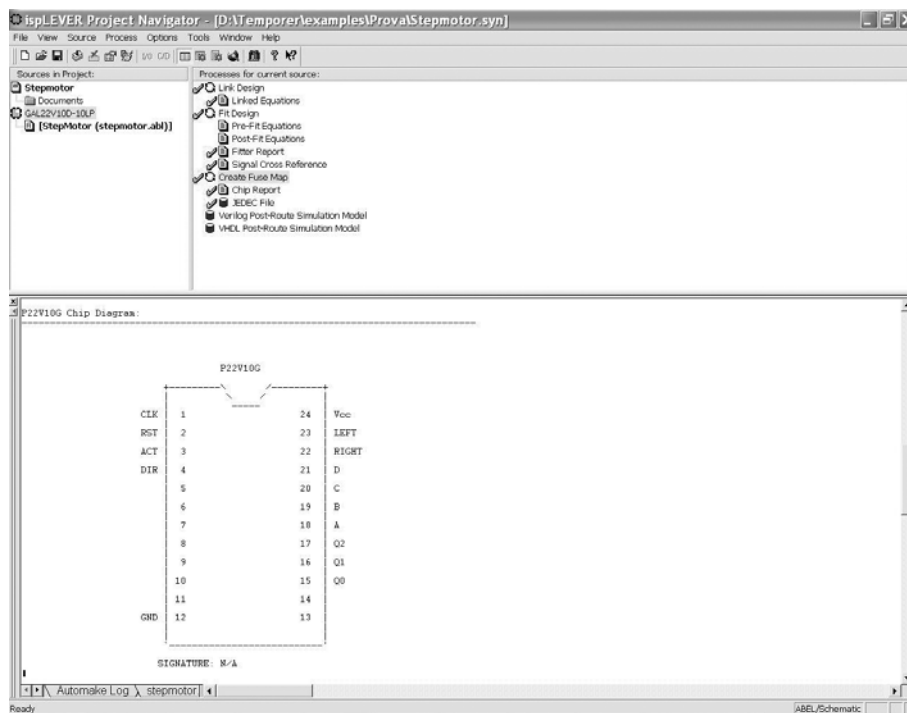
La línia 51 és una directiva que indica la finalització en la definició del sistema.

El funcionament de l'ispLEVER és molt intuïtiu. Fonamentalment els passos bàsics a seguir serien els següents:

- Petició de crear un nou projecte en el qual especificarem el dispositiu amb el qual treballarem. En el nostre cas per exemple, un dispositiu GAL22V10D-10LP.
- Inclusió en el projecte del fitxer descriptor del sistema. En el nostre cas el fitxer anteriorment entrat mitjançant qualsevol editor de text.
- Compilació del fitxer (*Link design*) perquè el programa ens mostri la viabilitat d'implementar les equacions descrites.
- Adaptació de les equacions descrites al dispositiu escollit (*Fit design*) perquè el programa determini les connexions internes en el dispositiu GAL.

- Creació del fitxer de gravació (*Create fuse map*) perquè el programa generi el fitxer JEDEC necessari per a la gravació.

Figura 31. Entorn de treball ispLEVER



A la finestra superior esquerra es mostren els documents que engloben el projecte. A la finestra superior dreta es mostren els passos que cal anar seguint per a implementar el sistema. A la finestra inferior es visualitzen els resultats de cadascuna de les etapes.

Finalment tan sols restaria emprar qualsevol gravador per a enviar aquest fitxer al dispositiu i poder verificar-ne físicament el funcionament. Hi ha molts gravadors comercials, alguns per exemple de la mateixa empresa Lattice, de tal manera que un pot adreçar-se a qualsevol comerç minorista de components electrònics per a adquirir-ne un.

Tanmateix, aquests gravadors són electrònicament força simples, per la qual cosa molta gent se'n construeix un per a ús personal. A Internet es pot trobar força informació sobre alguns d'aquests gravadors personals que els seus desenvolupadors han fet públics a la xarxa.

Al llarg de tots els passos anteriorment descrits el programa genera fitxers de text on va detallant els resultats obtinguts.

Un exemple de la seva utilitat el podem trobar quan per exemple el programa “decideix” canviar el piu assignat a una sortida per una altra perquè d'aquesta manera potser li resulta més fàcil fer la implementació, o bé quan “decideix” que genera un senyal de sortida tal com l'hem definit, però amb polaritat inversa, etc.

Pot semblar estrany que aquests programes decideixin pel seu compte aspectes com aquests, però cal entendre que a vegades podem entrar uns requeriments

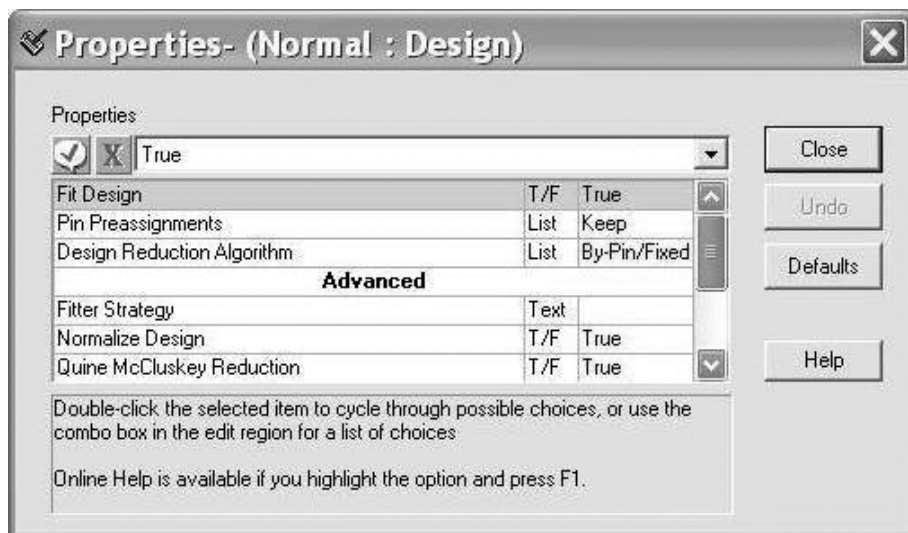
### Gravadors personals populars

Alguns gravadors personals que s'han fet populars a la xarxa són el GALBlast, el P16PRO40, el programador de Jaap Havinga, etc.



que el programa no vegi viables de poder implementar i opti per solucions properes però lleugerament diferents. En el cas de l'ispLEVER aquests graus de llibertat es determinen en el menú *Properties* (figura 32), que permet definir diferents maneres d'executar cadascun dels passos esmentats anteriorment.

Figura 32. Finestra de propietats de l'ispLEVER on es defineixen diverses opcions que afecten la implementació de cadascuna de les diferents etapes



A part d'aquests passos bàsics, ispLEVER admet altres funcionalitats, com per exemple la simulació del disseny per a verificar-ne el funcionament abans de fer la gravació. En general, però, els dissenys amb dispositius GAL solen ser relativament simples, raó per la qual no sol tenir gaire sentit destinar temps a fer simulacions que aportaran poca informació.

## 5. Els microcontroladors PIC

De manera genèrica anomenem *controlador* qualsevol sistema destinat a gestionar el correcte funcionament d'un procés. Més específicament i des del vessant electrònic, la major part d'aquests sistemes se solen implementar per mitjà de circuits integrats que incorporen tota una sèrie de prestacions que permeten adaptar-los a cada aplicació. Aquests circuits integrats són els microcontroladors, i d'exemples del seu ús en podem trobar tants com vulguem:

- per a controlar la temperatura d'un forn,
- per a controlar el sistema d'injecció d'un vehicle,
- per a controlar una porta automàtica,
- per a controlar un sistema de reg,
- per a controlar una màquina expenedora de tabac, etc.

En el subapartat 5.1 parlarem de les parts fonamentals d'un microcontrolador. En el subapartat 5.2 ens centrarem en l'estudi dels microcontroladors PIC, parlarem de les seves característiques i la seva rellevància i analitzarem un exemple de disseny. Al subapartat 5.3 analitzarem les diferències entre els diferents dispositius de la família PIC. Per a acabar, al subapartat 5.4 veurem el desenvolupament complet d'un exemple.

### 5.1. Elements fonamentals d'un microcontrolador

Els elements fonamentals d'un microcontrolador són:

- **Processador.** És l'element essencial que s'encarrega d'adreçar la memòria per a llegir la instrucció que cal executar, descodificar-la i activar els senyals necessaris per a la seva execució.
- **Memòria.** En aquesta cal diferenciar aquella part on s'emmagatzemen les instruccions i aquella part on s'emmagatzemen les dades. Si les instruccions i les dades es troben integrades en una única memòria es parla d'arquitectures Von Neumann, mentre que si es troben en dues memòries independents es parla d'arquitectures Harvard.
- **Ports d'entrada i sortida.** Permeten comunicar el microcontrolador amb la resta de dispositius del sistema global.
- **Relotge.** Marca el ritme de funcionament del microcontrolador i sincronitza els diferents elements.

#### Processadors CISC i RISC

Els processadors se solen catalogar segons la complexitat de les possibles instruccions que poden executar. Així doncs, parlem de processadors CISC si permeten executar instruccions relativament complexes que sovint requereixen de més d'un cicle de relotge; o bé processadors RISC si tan sols permeten executar instruccions simples sovint amb un únic cicle de relotge.

Tot i així els microcontroladors comercials incorporen tota una sèrie de prestacions addicionals que permeten adaptar-los més fàcilment a l'aplicació final per a la qual està pensat. Algunes d'aquestes prestacions són:

- *Timers* (temporitzadors) que permeten controlar intervals de temps.
- *Watchdogs* (“gossos guardians”) que reinicialitzen el procés si detecten que el sistema s’ha quedat bloquejat durant un cert temps.
- Convertidors A/D i D/A.
- Moduladors d’amplada de polsos (*PWM*).
- Busos de comunicacions amb diversos protocols: USART, USB, I<sub>2</sub>C, IrDA,...
- Etc.

## 5.2. La rellevància dels PIC de Microchip

Els primers circuits integrats que es van catalogar com a microcontroladors van ser fabricats per Intel, i de fet és seu un dels microcontroladors més popular, el 8051 de 8 bits. També són d’aquesta marca alguns dels microcontroladors que millor emulen el funcionament dels ordinadors portàtils, com és el cas del 80386EX. L’altre gran fabricant de microcontroladors és Motorola, del qual destaca el popular 68HC11 amb 8 bits per al bus de dades i 16 bits per al bus d’instruccions. A la figura 33 podeu veure l’aspecte del 8051C i del 68HC05 de Motorola.

El circuit integrat 8048 d’Intel dels anys 70 es considera el primer microcontrolador.

Per la seva banda, l’empresa americana Microchip és la que fabrica els microcontroladors PIC. A mitjan anys 70 l’empresa portava el nom GI i va crear un circuit integrat anomenat PIC pensat per a controlar les entrades i sortides d’un microprocessador de fabricació pròpia amb el qual estaven tenint problemes. Curiosament l’arquitectura d’aquest controlador continua essent la mateixa que utilitzen alguns dels PIC actuals. A mitjan 80 l’empresa es va vendre a un conjunt d’inversors, que la van rebatejar com Arizona Microchip Technology, que potencià el desenvolupament de nous controladors PIC amb noves prestacions.

Figura 33. Aspecte físic de populars microcontroladors com el 80C51 d’Intel i el 68HC05 de Motorola



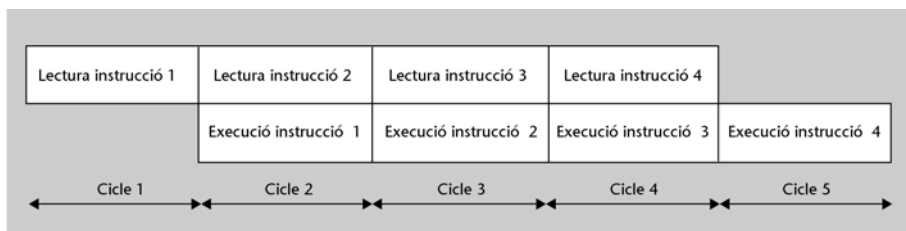
Actualment Microchip dissenya a Arizona alguns dels microcontroladors més emprats a escala mundial, tot i que la fabricació es fa en països asiàtics. Tot plegat ha relançat l’empresa fins a convertir-la en un líder del mercat.

Però, a què és deguda la rellevància del PIC? Abans de respondre la pregunta analitzem les característiques fonamentals d’un PIC:

- Processador tipus RISC que disposa d’un conjunt molt limitat d’instruccions, unes poques desenes que en permet optimitzar l’arquitectura. El fet de disposar de poques instruccions no els desautoritza per a aplicacions on calguin operacions més complexes, ja que aquestes normalment es poden implementar a partir d’un conjunt d’operacions simples.
- Arquitectura fonamentada en bancs de registres, és a dir, tots els elements del sistema tals com ports d’entrada i sortida, temporitzadors, convertidors, etc. estan mapats en registres físics.

- Memòria tipus Harvard que permet adreçar separatament les instruccions de les dades.
- Tècniques de segmentació en l'execució de les instruccions (*pipeline*), que permet estar llegint una instrucció mentre, en el cicle  $N$ , se n'executa una altra. Vegeu-ne un exemple a la figura 34: en aquesta el cicle  $N$  el processador es troba llegint de la memòria de programa la instrucció  $N$ , i simultàniament està executant la instrucció  $N - 1$  llegida en el cicle anterior.

Figura 34. Procés de segmentació



Procés de segmentació pel qual un microcontrolador llegeix una instrucció mentre n'executa una altra. D'aquesta manera en el cicle  $N$  el processador es troba llegint de la memòria de programa la instrucció  $N$  i simultàniament està executant la instrucció  $N - 1$  llegida en el cicle anterior.

Tanmateix, hi ha molts altres fabricants que segueixen polítiques similars en alguns dels seus microcontroladors. Per què són, doncs, rellevants els dispositius PIC? La resposta no està del tot clara i podem trobar múltiples explicacions.

D'una banda tenim un gran nombre de models amb múltiples recursos accessoris (ports, temporitzadors, convertidors, etc.), de tal manera que pràcticament se'n pot trobar un perfectament adaptat a les necessitats de l'aplicació concreta. D'altra banda, tenim un elevadíssim nombre de programes de suport tals com compiladors en C o BASIC, programadors, simuladors, emuladors en temps real, etc. relativament econòmics. Aquests aspectes els detallarem més endavant.

Tot i així, potser tot plegat tan sols és degut a la popularitat agafada pels PIC amb el temps, reforçada per diversos estudis i experiències de dissenyadors de sistemes. Aquests han vist com a l'hora de comparar un PIC amb altres microcontroladors de prestacions similars d'altres fabricants, s'han trobat que els primers solen ser més eficaços quant a temps d'execució o bé longitud del codi necessari.

Segurament la resposta de la rellevància del PIC no és per un aspecte concret sinó pel conjunt dels aspectes que s'han anat esmentant.

Cal ser, però, conscients que el mercat mundial de microcontroladors és molt gran i a l'hora de dissenyar una aplicació electrònica podem tenir en consideració les solucions que aporten molts altres fabricants: AMD, Atmel, Fujitsu, Hitachi, Intel, Mitsubishi, Motorola, National, Philips, ST, Texas, Toshiba, Zilog, etc.

### 5.3. Les famílies dels dispositius PIC de Microchip

Tal i com hem dit, un dels aspectes més destacats dels dispositius PIC és el gran ventall de models, que permet disposar-ne d'un que pràcticament s'adapti a les nostres necessitats. Així podem trobar microcontroladors amb diferents busos de dades segons el grau de complexitat de les operacions que vulguem fer:

- Famílies PIC amb bus de dades de 8 bits: PIC10, PIC12, PIC16 i PIC18
- Famílies PIC amb bus de dades de 16 bits: PIC24, dsPIC30 i dsPIC33
- Famílies PIC amb bus de dades de 32 bits: PIC32

Dins d'aquestes famílies podem trobar molts models que permeten incloure diferents prestacions, com per exemple:

- Fins a 512 Kbytes de memòria de programa i 32 Kbytes de memòria de dades.
- Possibilitat de protegir el codi de programació davant d'intents de còpia.
- Protocols de comunicació integrats en els dispositius: RS232, RS485, SPI, I<sub>2</sub>C, CAN, USB, TCP/IP, IrDA, ràdio UHF, etc.
- Controls de temps: comptadors, *timers*, *watchdogs*, etc.
- Compatibilització amb senyals analògics: convertidors A/D de fins a 12 bits, convertidors AD, comparadors, tensions de referència, etc.
- Adaptació a dispositius LCD i LED.
- Implementació de processadors digitals.
- Velocitat del processador fins a 40 MIPS.

El ventall de dispositius és tan elevat que, per tal d'automatitzar la cerca de les millors opcions disponibles, es fa necessari accedir a la pàgina web de Microchip. En aquesta apareix un selector de dispositius en funció de les necessitats del disseny que volem implementar. A la taula 1 teniu un resum de les prestacions més importants de les diferents famílies.

MIPS: milions d'instruccions per segon.

Web de Microchip:  
<http://www.microchip.com>

Tabla 1. Resum de les prestacions de les famílies de dispositius PIC més populars

Dispositius PIC amb busos de dades de 8 bits					
Família	Preu abril 08 (\$)	Freq. màx CPU (MIPS)	Nombre màx. pins E/S	Programa KBytes	Prestacions
PIC10	0,39 a 0,47	1 a 2	4	0,25 a 0,5	Convertidors A/D 1 <i>timer</i> de 8 bits
PIC12	0,47 a 0,80	1 a 5	6	0,75 a 3,5	Convertidors A/D 2 <i>timers</i> de 8 bits + 1 <i>timer</i> de 16 bits
PIC16	0,37 a 3,71	5 a 10	52	0,75 a 14	Convertidors A/D 2 <i>timers</i> de 8 bits + 1 <i>timer</i> de 16 bits 1 SPI/I <sub>2</sub> C + 1 USART
PIC18	1,47 a 8,06	6 a 16	70	4 a 128	Convertidors A/D 2 <i>timers</i> de 8 bits + 3 <i>timers</i> de 16 bits 2 SPI/I <sub>2</sub> C + 2 USART

Dispositius PIC amb busos de dades de 8 bits					
Família	Preu abril 08 (\$)	Freq. màx CPU (MIPS)	Nombre màx. pins E/S	Programa KBytes	Prestacions
PIC24F	1,66 a 4,61	16	85	16 a 256	Convertidors A/D 5 timers de 16 bits 3 SPI + 3 I <sup>2</sup> C + 4 UART IrDA + USB 2.0 3 comparadors 9 PWM
PIC24H	2,14 a 5,10	40	85	12 a 256	Convertidors A/D 9 timers de 16 bits + 4 timers de 32 bits 2 SPI + 2 I <sup>2</sup> C + 2 UART IrDA + 2 CAN 2 comparadors 8 PWM

#### 5.4. Desenvolupament d'un sistema amb microcontroladors PIC

Per tal d'implementar un sistema amb un microcontrolador cal seguir tot un conjunt de processos. Aquests processos es realitzen amb el suport de programes i circuits electrònics.

Pel fet d'estar implementant una solució a baix nivell, és a dir, molt propera a l'arquitectura que el dispositiu ens facilita, és pertinent tenir present el funcionament intern del PIC que utilitzarem. Aquesta tasca no és difícil i un cop ens habituem a treballar amb qualsevol d'aquests, trobarem que la resta tenen un funcionament molt similar. Ara bé, pel fet que a vegades canvia el nom d'algun dels registres, o bé s'afegeixen alguns bits addicionals per a controlar certs senyals, entre d'altres situacions, es fa pertinent que tinguem davant nostre el manual de funcionament del dispositiu emprat, que es pot descarregar des de la pàgina web de Microchip.

Per a exemplificar-ho agafarem, per exemple, un PIC16F84A. El primer que faríem serà cercar el *datasheet* d'aquest dispositiu, document en el qual es troba explicada tant la seva forma física, com la seva arquitectura i les instruccions que admet. D'aquesta manera veurem algunes particularitats que posteriorment potser cal considerar, en el moment de fer l'aplicació, com per exemple les següents:

- La memòria de programa és d'1 K amb instruccions de 14 bits que inicia l'execució des de la posició 00h. Cal considerar que a la posició 04h hi ha l'adreça a la qual salta el programa en cas d'haver-hi una interrupció.
- La memòria de dades està segmentada en dues seccions de tal manera que a les adreces baixes hi trobem registres de funcions especials, mentre que a

les adreces altes trobem 68 bytes utilitzables per l'usuari. A més a més, el conjunt de la memòria de dades està segmentada en dos blocs diferents, anomenats *bancs de memòria*, de tal manera que caldrà tenir present en cada moment a quin ens estem adreçant.

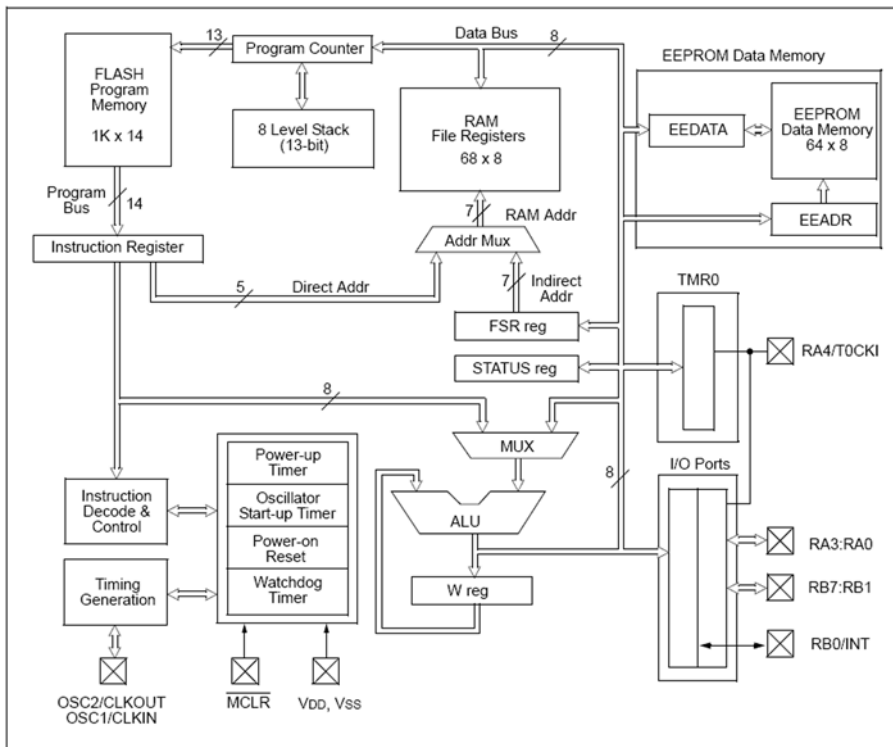
- Alguns dels registres de funcions especials permeten configurar aspectes tals com si volem que els ports siguin d'entrada o bé de sortida (registres anomenats *TRISA* o *TRISB*), com d'altres que mapen a memòria el funcionament del dispositiu, com per exemple el valor que prenen aquests ports (registres anomenats *PORTA* o *PORTB*).
- Les operacions que efectua l'ALU es poden emmagatzemar a diferents registres, des dels mateixos registres de funcions especials, fins a per exemple un registre temporal anomenat *W*.
- Les instruccions s'executen a una freqüència quatre vegades més lenta que la freqüència de l'oscil·lador extern que s'utilitzi.
- Etc.

Tot i que aparentment pot semblar complicat, cal entendre que el que s'explica tan sols evidencia la necessitat d'analitzar l'arquitectura del dispositiu, abans de la seva programació, pel fet d'estar realitzant una programació molt propera a l'arquitectura. Afortunadament les diferències entre els diferents models solen ser menors, tot i que per a cadascun d'ells cal tenir present el *datasheet* corresponent.

### **PIC16F84A**

El PIC16F84A (figura 35) presenta una memòria de programa de 1024 posicions on es desen les instruccions de 14 bits. A aquesta memòria s'hi accedeix per l'adreça emmagatzemada en el comptador de programa (*program counter*), per tal d'obtenir la instrucció corresponent que s'emmagatzema en el registre d'instrucció (*instruction register*) i que finalment es decodifica pel mòdul corresponent (*instruction decode and control*). Quan en el programa trobem una crida a una funció aleshores l'adreça actual s'emmagatzema en una pila de fins a 8 nivells (*8 level stack*) per a recuperar-la quan finalitzi l'execució de la funció que s'ha cridat. També trobem una memòria RAM de 68 bytes i una EEPROM de 64 bytes per a guardar tota una sèrie de registres que mapen i configuren el dispositiu, com ara si els ports són d'entrada o de sortida (el sentit és configurable), les interrupcions, el temporitzador TMR0, etc.

Figura 35. Diagrama de l'arquitectura interna del PIC16F84A on es mostren les seves característiques fonamentals.



Internament també observem la unitat aritmèticològica (ALU), que executa les operacions lògiques entre operands que poden procedir de diverses zones tals com del registre que emmagatzema cada instrucció, de la memòria de dades, dels registres interns o bé del registre temporal *W*. Els seus resultats es poden desar també en el registre *W* o bé directament a memòria o registres.

La resta d'elements interns són bàsicament el temporitzador *TMR0* i els registres de temporització tals com el *watchdog timer*, així com algun registre addicional que s'utilitza per a accessos indirectes a memòria (*FSR*).

Observem també 13 pins d'entrada/sortida configurables per l'usuari (*RA0* a *RA4* i *RB0* a *RB7*). N'hi ha alguns que són configurables també per a introduir el senyal amb el qual varia el temporitzador (*RA4/TOCKIN*) o bé per a introduir el senyal que genera una interrupció (*RB0/INT*).

Entre els altres pins trobem *OSC1/CLKIN*, al qual podem aplicar una xarxa resistor-condensador per a generar el senyal de rellotge i el pin *OSC2/CLKOUT* que genera un senyal de rellotge que mostra la freqüència amb la qual s'executen les instruccions i que presenta una freqüència igual a  $\frac{1}{4}$  de la freqüència del senyal generada externament a *OSC1/CLKIN*. També es pot utilitzar un oscil·lador basat en un cristall de quars aplicat entre aquests dos pins.

El microcontrolador es reinicialitza mitjançant el senyal *MCLR*, que és actiu per nivell baix.

La resta de pins són d'alimentació (*VDD* i *VSS*).

En el mateix document podem trobar també les instruccions que admet aquest dispositiu; hi podem veure que només són un total de 35 i que la majoria s'executen en un cicle de rellotge, excepte les que impliquen un salt en el comptador de l'adreça del programa (registre *PCL*), que aleshores s'executen en dos cicles de rellotge. Posteriorment aprofitarem un breu exemple per a mostrar-ne algunes.



### 5.4.1. Desenvolupament de l'aplicació

Així doncs, un cop haguem analitzat el dispositiu, podrem entrar en el desenvolupament de l'aplicació.

Primer de tot s'escriu el codi que ha d'executar el microcontrolador a partir de les especificacions del sistema amb qualsevol editor de textos. Cal recordar que estem parlant de l'execució d'un circuit integrat que finalment estarà implementat en una placa de circuit imprès, per la qual cosa cal tenir clara la interfície amb aquesta: els pins on es connecten les entrades i sortides, els senyals analògics que s'apliquen als convertidors del PIC, etc.

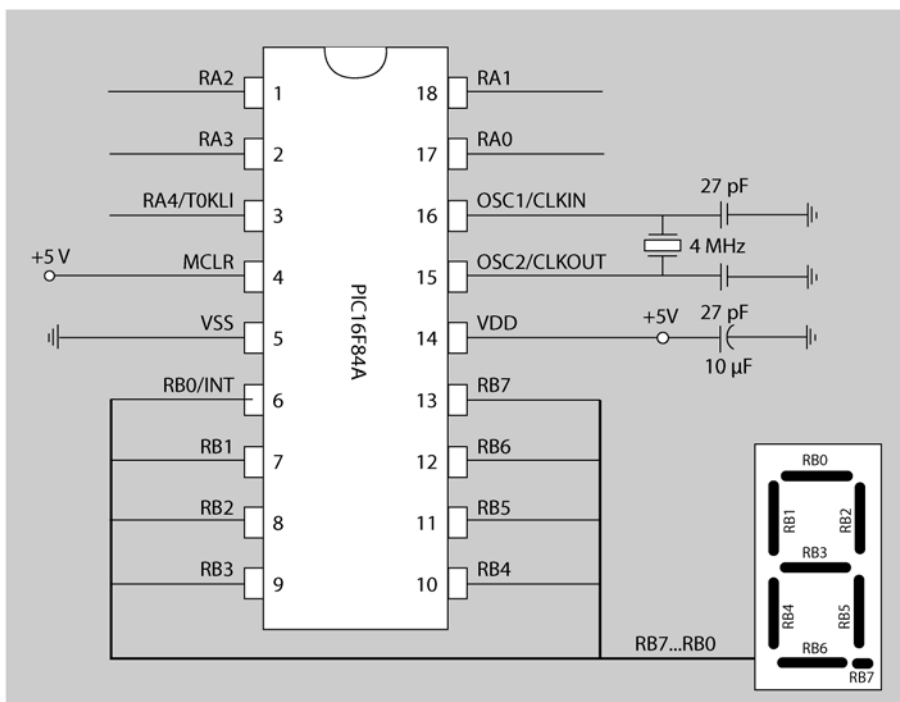
Per a escriure aquest codi podem emprar assembler, com per exemple el proposat per Microchip, o llenguatges de més alt nivell, com per exemple C.

A diferència de la programació d'aplicacions informàtiques on els llenguatges d'alt nivell són els més emprats, en el cas dels microcontroladors la programació en llenguatge assembler és força habitual, ja que sovint aquests microcontroladors han d'executar processos relativament simples i per consegüent gens difícils de programar.

#### Exemple

Com a exemple tenim el codi següent, que es correspon amb una simple aplicació amb la qual el PIC anirà traient per als bits del seu *PORTB* els senyals necessaris per a activar un visualitzador de set-segments en què anirà apareixent una lletra cada  $\frac{1}{2}$  segon aproximadament (figura 36); concretament, apareixerà primer una P, després una I, després una C, després un espai en blanc i així successivament, és a dir "PIC PIC PIC...". Com que  $\frac{1}{2}$  segon és un temps molt elevat per a un processador, veurem com el codi niua un comptador dins d'un altre per a aconseguir aquest temps d'espera.

Figura 36. Esquema de l'exemple



Esquema de la possible aplicació de l'exemple on veiem que externament tan sols necessita un cristall extern de 4 MHz i un visualitzador de set-segments (amb punt) que s'activa pel port B del microcontrolador.

La intenció de mostrar el codi és a tall d'exemple, però cal tenir present que caldria analitzar-lo amb detall i sobretot aprendre's les instruccions per a comprendre'l correctament. Tanmateix, s'han afegit alguns comentaris que s'indiquen sempre amb un ";" i posteriorment també es comenta cadascuna de les línies del programa.

```

1;*****
2 ; ProjectLED.ASM
3 ; March, 2008
4 ; Device: PIC16F84A
5 ; Oscillator: 4MHz (Instruction cycle=1 microsecond)
6;*****
7
8 list      p=16F84A; definició del dispositiu
9 #include <p16F84A.inc>; inclu alias per facilitar la programació
10
11 __CONFIG  _CP_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC
12
13 ;**** DEFINICIO DE CONSTANTS
14 TEXT_LENGTH EQU 4      ; Longitud del text "PIC "
15
16 ;**** DEFINICIO DE VARIABLES
17
18 INTVAR          UDATA 0x0C
19 inner_counter   RES 1   ; variable emprada com a comptador
20 outter_counter  RES 1   ; variable emprada com a comptador
21 letter          RES 1   ; variable per a comptar la lletra actual
22
23 ;**** CODI
24 RST            CODE    0x000
25 goto begin
26 MAINCODE 0x005
27 beginbsf      STATUS,RP0
28 clrf TRISB
29 bcf          STATUS,RP0
30 clrf PORTB
31 resetclrletter
32 displaymovfletter,w
33 call table
34 movwf PORTB
35 countermovlwd'217'
36 movwf outter_counter
37 loopoutmovlwd'255'
38 movwf inner_counter
39 loopiudecfszinner_counter,f
40 goto loopiu
41 decfsz outter_counter,f
42 goto loopout
43 resultincfletter,f
44 movf letter,w
45 xorlw TEXT_LENGTH
46 btfsc STATUS,Z
47 goto reset
48 goto display
49 tableaddwfPCL,f
50 retlw b'00011111' ; codi per la lletra P
51 retlw b'00010010' ; codi per la lletra I
52 retlw b'01010011' ; codi per la lletra C
53 retlw b'00000000' ; codi per la lletra <blanc>
54 END

```

De la línia 1 fins a la línia 6 trobem entre comentaris la descripció del projecte. Tots els comentaris s'inicien amb punt i coma.

La línia 8 és una directiva (*list*) per a indicar al compilador el dispositiu final.

La línia 9 és una directiva (*#include*) per a incloure un fitxer que inclou constants per a facilitar la programació. Així per exemple podem trobar "W EQU h0000", que equival a dir que allà on escrivim *W* equival a escriure l'adreça 00h.

La línia 11 és una directiva per a configurar alguns aspectes clau del dispositiu, tals com que funciona amb un cristall de quars (*XT\_OSC*) o bé que no té activat el *watch dog* (*WDT\_OFF*).

La línia 14 és una constant definida per nosaltres per a indicar la longitud de la frase que anirà visualitzant-se pel set-segments.

La línia 18 indica que el codi que s'escriu a continuació s'emmagatzema a l'adreça 0Ch. Aquesta adreça no és arbitrària, sinó que respon a la primera adreça de memòria EEPROM, que està lliure de registres del dispositiu mapat a memòria. Per consegüent, està lliure perquè l'usuari la faci servir per a allò que consideri oportú, típicament per a desar variables. La paraula *INTVAR* tan sols és una etiqueta.

Les línies 19, 20 i 21 són variables necessàries per al programa que reserven (*RES*) un byte de memòria.

La línia 24 indica que el següent codi s'emmagatzema a l'adreça 00h, que és l'adreça inicial d'execució després d'inicialitzar-se el sistema. La paraula *RST* tan sols és una etiqueta.

La línia 25, que serà la que s'emmagatzemarà a l'adreça de memòria 00h, indica que el programa salti a l'adreça 05h. Això és habitual, ja que a l'adreça 04h és on salta el dispositiu en el cas que s'executi una interrupció i per consegüent el codi "real" sol escriure's a partir de l'adreça 05h.

La línia 26 indica que el següent codi s'emmagatzemarà a partir de l'adreça 05h. La paraula *MAIN* tan sols és una etiqueta.

La línia 27 posa a 1 lògic el bit *RPO* del registre *STATUS*. Amb això se selecciona el banc de memòria 1, que és on es troba el registre *TRISB* i que s'utilitza per a programar el funcionament del *PORTB*.

La línia 28 posa a 0 tots els bits del registre *TRISB* per a indicar que el *PORTB* és de sortida.

La línia 29 posa a 0 lògic el bit *RPO* del registre *STATUS* per a seleccionar el banc de memòria 0, que és on es troba el registre *PORTB*, que s'utilitza per a escriure els bits d'aquest port o bé llegir-los en cas d'haver-lo definit com a port d'entrada.

La línia 30 posa a 0 tots els bits del *PORTB* i així s'apaguen els *led*.

La línia 31 posa a 0 la variable *letter*. La paraula *reset* és una etiqueta per després poder-se referenciar a aquesta posició del codi.

La línia 32 copia a la variable *W* el valor de la variable *letter*. La paraula *display* és una etiqueta per després poder-se referenciar a aquesta posició del codi.

La línia 33 fa un salt de programa on hi hagi el codi amb l'etiqueta *table*. Aquesta part de codi és una taula on s'emmagatzemen els valors que ha de prendre el *PORTB*. Aquests valors s'emmagatzemaran al registre *W*.

La línia 34 copia a *PORTB* el valor de *W*. Aquesta és la primera línia que s'executa en retornar del salt que s'ha fet a la taula on es guarda a *W* el valor que ha de prendre el *PORTB*.

La línia 35 guarda a *W* el valor decimal 217. La paraula *counter* és una etiqueta per després poder-se referenciar a aquesta posició del codi. En aquesta part del codi s'executarà un bucle 217 vegades on a cada iteració s'executarà un altre bucle 255 vegades. L'execució global d'aquests bucles és aproximadament de 0,5 segons.

La línia 36 guarda a la variable *outter\_counter* el valor del registre *W*.

La línia 37 guarda a *W* el valor decimal 255. La paraula *loopout* és una etiqueta per després poder-se referenciar a aquesta posició del codi.

La línia 38 guarda a la variable *inner\_counter* el valor del registre *W*.

La línia 39 decrementa la variable *inner\_counter* en 1 i després avalua si el resultat és 0, de tal manera que si ho és saltarà la següent instrucció (no l'executarà). La paraula *loopiu* tan sols és una etiqueta per després poder-se referenciar a aquesta posició del codi.

La línia 40 torna a la línia anterior en la mesura que tan sols s'executa si *inner\_counter* encara no ha arribat a 0.

La línia 41 decrementa la variable *outter\_counter* en 1 i després avalua si el resultat és 0, de tal manera que si ho és saltarà la següent instrucció.

La línia 42 torna a la línia on hi ha l'etiqueta *loopout* en la mesura que tan sols s'executa si *outter\_counter* encara no ha arribat a 0 i per consegüent cal repetir el bucle interior.

La línia 43 incrementa la variable *letter* en 1 un cop ha sortit dels bucles.

La línia 44 posa a *W* el valor de la variable *letter*.

La línia 45 calcula una XOR de *W* amb el valor de la constant *TEXT\_LENGTH* per veure si són idèntiques, és a dir, que *letter* valgui el mateix que la longitud del text que es vol visualitzar. Si són iguals posa a 1 el bit Z del registre *STATUS*.

La línia 46 avalua el bit Z del registre *STATUS*, que si és 0 se salta la següent instrucció.

La línia 47 salta a la línia de codi etiquetada com a *reset* per tal de reiniciar des de 0 el valor de la variable *letter*.

La línia 48 salta a la línia de codi etiquetada com a *display* i així anar a aquella part de codi encarregada d'actualitzar a *PORTB* el valor que correspongui per a veure la lletra pel set-segments.

La línia 49 suma al registre on s'emmagatzema l'adreça del programa (*program counter*) el valor de *W*. Això serveix per a saltar fins a la línia on es desi el codi que ha de prendre el *PORTB* segons la lletra actual que calgui visualitzar. La paraula *table* és una etiqueta per a poder-se referenciar a aquesta posició del codi.

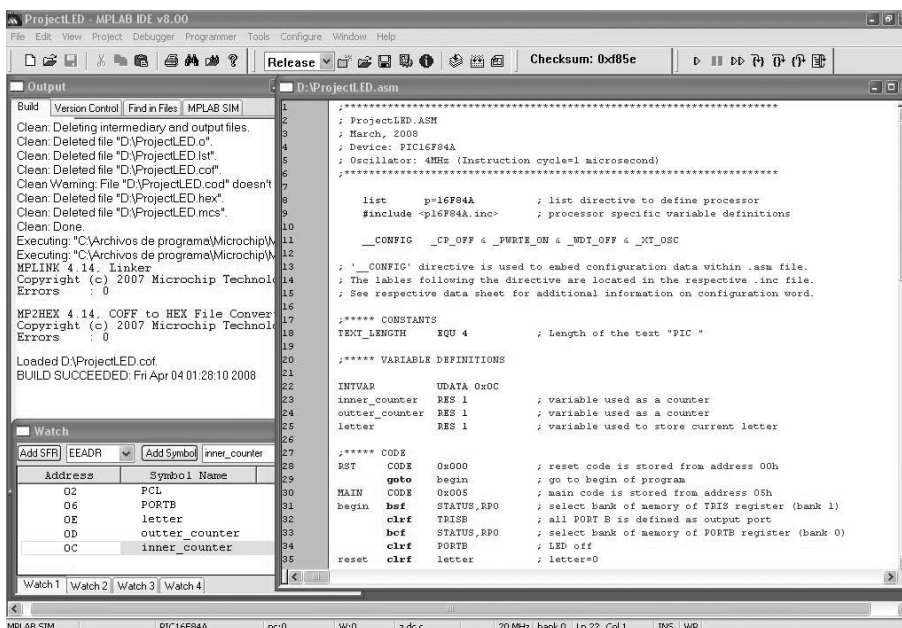
Les línies 50, 51, 52 i 53 són retorns a la posició de programa que havia fet un salt a *table* i actualitza el valor del registre *W* amb els valors que ha de prendre el *PORTB*.

La línia 54 és una directiva per a indicar la finalització del codi.

Posteriorment es compila el codi que haurem escrit amb la finalitat de generar el codi objecte. Aquesta compilació converteix el llenguatge ensamblador o d'alt nivell en el conjunt de zeros i uns que físicament es gravaran a la memòria del microcontrolador i que donarà lloc a cadascuna de les instruccions que executarà el PIC.

Per a poder generar el codi objecte cal un programa específic que faci aquesta conversió. La mateixa empresa Microchip facilita gratuïtament a través de la seva pàgina d'Internet alguns programes per a poder efectuar aquesta tasca. Així, si hem optat per escriure el programa en ensamblador podem emprar per exemple el MPASMWIN de Microchip (vegeu a la figura 37 una mostra d'aquest programari). Afortunadament, però, Microchip ha desenvolupat una aplicació de lliure distribució, anomenada MPLAB, que integra molts d'aquests programes i facilita els desenvolupaments.

Figura 37. Exemple de compilació amb MPASMWIN dins de MPLAB



Un cop escrit el codi objecte podem realitzar un procés opcional però molt recomanable, que és el de simular el codi abans de gravar-lo al microcontrolador. De fet, el camp de la simulació de sistemes electrònics ha donat lloc a un gran ventall de possibilitats durant les darreres dècades i el camp dels microcontroladors PIC no n'ha estat una excepció.

Així, per exemple, la mateixa empresa Microchip facilita a través d'Internet el programa de simulació MPLAB-SIM, el qual permet visualitzar en el nostre ordinador personal amb l'entorn MPLAB com s'aniran executant les diferents instruccions i podrem anar veient l'evolució dels valors dels diferents registres del microcontrolador a fi de validar-ne el funcionament.

Tot i així, aquest tipus de simulació no permet tenir en consideració els senyals "reals" que poden interactuar amb el microcontrolador tals com senyals de busos de comunicacions reals o bé senyals analògics aplicats als convertidors. Tampoc no permeten analitzar problemes de funcionament en temps real.

Per aquests motius hi ha altres possibilitats de simulació tals com els emuladors, com el MPLAB-ICE de Microchip, que permeten connectar un cable al sòcol de la placa de circuit imprès on finalment s'hagi de muntar el microcontrolador PIC. Per mitjà d'aquest es pot simular el funcionament real que tindrà el nostre microcontrolador un cop s'implementi físicament en aquella placa.

Un cop verificada la correctesa del codi objecte tan sols resta gravar-lo dins de la memòria del microcontrolador perquè aquest l'executi quan es reinicialitzi. Aquesta gravació s'efectua per mitjà d'un programa específic instal·lat al nostre ordinador personal que envia aquest codi cap als pins corresponents del microcontrolador, habitualment per mitjà d'un dels ports del nostre ordinador i una placa gravadora en la qual es posa el microcontrolador.

A l'igual que amb els compiladors i simuladors podem trobar tant programes i gravadors propis de Microchip, com el PICSTART o el PICKit2; com molts d'altres de moltes empreses diferents. Igualment, alguns dels esquemes elèctrics d'aquests gravadors són públics, fet que permet que hom pugui desenvolupar-se'l.

Finalment, tan sols restarà connectar el microcontrolador en el sòcol de la placa de circuit imprès de l'aplicació electrònica final i verificar-ne el funcionament.

#### Programes per a generar codis objecte

Hi ha molts programes comercials que compilen programes escrits en llenguatges d'alt nivell per a generar codis objecte per a microcontroladors PIC. N'hi ha alguns de molt elaborats que disposen de tot un conjunt de biblioteques amb funcions molt complexes que faciliten l'elaboració del codi, tècniques d'optimització de codi final a fi de minimitzar el temps d'execució de l'aplicació final, compatibilitzacions amb altres rutines escrites en llenguatges d'alt nivell, possibilitat d'incorporar rutines escrites en assemblador, etc. Alguns exemples d'aquests programes són el PBASIC de Parallax o bé el C2C de Byte Craft.

## 6. Els motors elèctrics

Podríem definir un motor com aquella màquina que pot convertir energia elèctrica en energia mecànica. Aquesta funció els dóna una especial importància ja no només dins el món de les màquines elèctriques, sinó també dins el món econòmic. Per aquest motiu se'n fa imprescindible l'estudi.

Aquesta funció dels motors, però, és sempre reversible, poden actuar també de generadors per a convertir energia mecànica en energia elèctrica. Per això, normalment en la majoria de bibliografia no es parla de motors elèctrics, sinó de màquines elèctriques en general.

Al subapartat 6.1 analitzarem algunes de les característiques generals que defineixen els motors i al subapartat 6.2 ens centrarem en l'estudi dels motors de contínua.

### 6.1. Consideracions generals sobre els motors elèctrics

Hi ha molts tipus de motors elèctrics. No es diferencien tan sols pel tipus de senyal elèctric que els alimenta (alimentació alterna o contínua), sinó també per la seva estructura i elements que els constitueixen.

Les primeres característiques en què ens fixem d'un motor són la velocitat i el parell de forces que pot arribar a subministrar. Molt sovint voldrem controlar totes dues característiques o només alguna d'aquestes. Cada tipus de motor és adequat per a un ús concret i no amb tots podrem controlar tots els paràmetres:

- Els **motors asíncrons**, especialment els trifàsics, són els més utilitzats en la indústria en aplicacions d'elevada potència. Permeten una bona regulació de la velocitat i són força econòmics.
- En aplicacions que requereixen un gran parell dinàmic (força), ja sigui en l'arrencada del motor o per variacions de la càrrega, és freqüent la utilització de **motors síncrons**.

Els motors asíncrons són motors d'alterna i n'hi ha de trifàsics i de monofàsics.

Els motors síncrons són també motors d'alterna.

Tot i així en el nostre cas ens trobem habitualment desenvolupant aplicacions de baixa potència i treballam amb senyals continus, per aquest motiu i atesa l'extensió del camp dels motors, ens centrarem només en l'estudi dels motors de contínua.

## 6.2. El motor de contínua

Els motors de contínua ens permeten una bona regulació de la velocitat i el sentit de gir tan sols modificant el valor i la polaritat de la tensió d'alimentació. Tanmateix, com veurem, aquest control es pot fer també a partir de senzilles tecnologies electròniques com el PWM (*pulse width modulation*, modulació de l'amplitud dels polsos). Molt sovint porten acoblats a l'eix engranatges reductors que faran que el moviment de rotació de l'eix sigui més lent, però amb el parell de força adequat a l'aplicació concreta.

Tot i que s'han imposat en usos de baixa potència i baixa tensió, poden actuar també fins a potències d'uns quants megawatts. La seva instal·lació és relativament econòmica però requereixen un cert manteniment sobretot pel que fa a les escobretes.

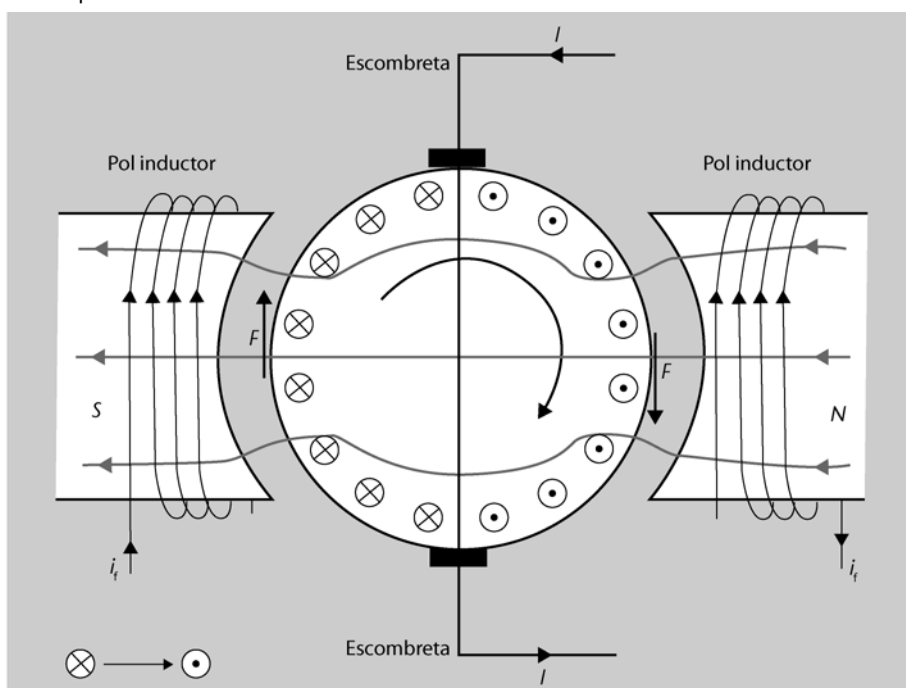
Al subapartat 6.2.1 veurem què són les escobretes.

### 6.2.1. Parts del motor de contínua

Un motor de contínua es compon dels següents elements:

- **L'inductor o estator.** És la part immòbil del motor. Es tracta d'un element del circuit magnètic que constitueix el motor al qual es bobina un cable per a produir un camp magnètic. D'aquesta manera es construeixen uns electroimants amb una cavitat cilíndrica entre els seus pols (vegeu la figura 38).

Figura 38. Constitució d'un motor de contínua. Podem veure el rotor i els pols formats per l'electroimant de l'estator.



**Figura 38**

Fixeu-vos que, com que els conductors del rotor queden en perpendicular al pla de la imatge, el sentit del corrent s'indica amb una creu o un punt. El punt indica que el sentit del corrent és cap a nosaltres (sortint del pla del full) i, en canvi, la creu indica que el corrent entra cap al full.

- **L'induït o rotor.** És la part mòbil del motor (part central de la figura 38). És un cilindre de xapes magnètiques aïllades entre si i perpendiculars a l'eix

del cilindre. Ocupa la cavitat cilíndrica formada entre els electroimants de l'estator. Tal com diu el seu nom, rotor, gira al voltant del seu eix. A la seva perifèria trobem conductors regularment repartits (vegeu la figura 38).

- **El col·lector i les escobretes.** Són els elements utilitzats per a alimentar el rotor. El col·lector es troba solidari al rotor i les escobretes, en canvi, són fixes. Les escobretes freguen els col·lectors en girar el rotor i n'alimenten els conductors (vegeu la figura 38).

### 6.2.2. Principi de funcionament del motor de contínua

Vegem ara el principi de funcionament d'un motor de contínua.

Quan alimentem l'estator es crea un camp magnètic de pol nord a pol sud de l'electroimant. Quan alimentem l'induït, el rotor, llavors els conductors situats a un mateix costat del pol de l'estator (a una banda de les escobretes) estan recorreguts per corrents del mateix sentit i a l'altra banda de les escobretes, davant de l'altre pol de l'electroimant, queden recorreguts per un mateix corrent però de sentit contrari.

A causa del camp magnètic que travessa el rotor, generat per l'electroimant, els conductors es veuen sotmesos a una força. Tanmateix, com que el corrent que recorre els conductors d'una banda és de sentit contrari al de l'altra banda, les forces a les quals queden sotmesos els conductors d'una banda i l'altra són de sentit contrari (vegeu de nou la figura 38, on teniu indicat el sentit d'aquestes dues forces).

Aquest parell de forces farà que el rotor tingui una tendència natural a girar. Per a canviar el sentit de gir és suficient amb canviar la polaritat de l'alimentació del rotor, o si no, canviar la polaritat de l'alimentació de l'estator i canviar així la polaritat dels pols de l'electroimant i, per tant, el sentit del camp magnètic generat.

Si el rotor s'alimenta amb una tensió  $V$  i hi circula un corrent  $I$ , tindrem que es genera una força electromotriu  $E$ :

$$E = V - R \cdot I \quad (25)$$

On el producte  $R \cdot I$  representa la caiguda de tensió òhmica en el rotor.

Aquesta força electromotriu es relaciona amb la velocitat angular del motor i amb el flux de camp magnètic a través de la següent equació:

$$E = k \cdot \omega \cdot \varphi \quad (26)$$



On  $k$  és una constant pròpia del motor,  $w$  la velocitat angular de rotació i  $\varphi$  el flux de camp magnètic.

El parell  $P$  generat per les forces del rotor també es pot relacionar amb el flux de camp magnètic i el corrent  $I$  del rotor de la següent manera:

$$P = k \cdot I \cdot \varphi \quad (27)$$

Podríem definir el parell  $P$  com la força que és capaç d'exercir el motor a cada gir.

Així doncs, fixeuvos que, donada una força electromotriu constant, si disminuïm el flux de camp magnètic, disminueix el parell; en conseqüència, si la força electromotriu  $E$  és constant, augmentaria la velocitat de rotació del motor.

### 6.2.3. Control de la velocitat

Bàsicament tenim dos mètodes directes que ens permeten regular la velocitat del motor:

- La primera manera consistiria a augmentar la força electromotriu  $E$  del motor; per tant, augmentant el valor de la tensió d'alimentació  $V$  de l'induït però amb excitació sempre constant. El parell es mantindria constant i per això s'anomena control de velocitat a parell constant.
- La segona manera consisteix a disminuir el flux d'excitació (corrent  $i_f$  de la figura 38) mantenint, però,  $V$  constant (per això s'anomena funcionament a potència constant), llavors baixarà el flux magnètic disminuint el parell però augmentant la velocitat a mantenir  $V$  constant.

Una cosa que hem de tenir sempre present és que el funcionament d'un motor és reversible. És a dir, si la càrrega solidària a l'eix de la màquina s'oposa al moviment, la màquina proporciona un parell que fa que funcioni de motor. En canvi, si la càrrega és tal que té tendència a fer girar la màquina, llavors s'inverteix el funcionament i passa a treballar com a generador.

### 6.2.4. Funcionament amb PWM

Tal com dèiem a la introducció d'aquest subapartat 6.2, una altra manera de controlar la velocitat d'un motor de contínua és utilitzant un PWM (*pulse width modulation*). Una de les maneres més senzilles de variar la velocitat és, tal com hem dit, variant la tensió d'alimentació  $V$  del rotor.

#### Observació

Tingueu present que força sovint un dispositiu digital no ens donarà suficient corrent per a alimentar un motor i hi hauràrem de posar entremig algun tipus de component que pugui subministrar més potència.

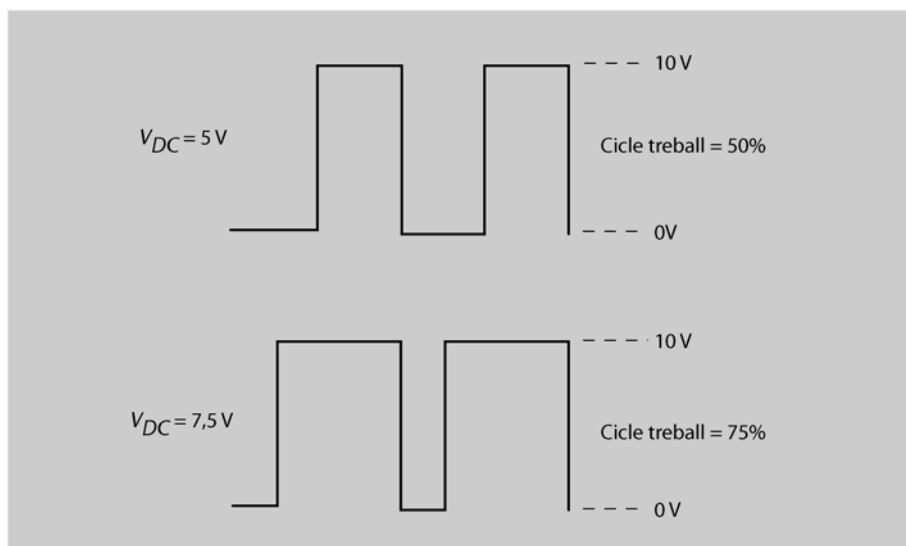
Aquesta tensió contínua pot estar molt sovint generada per algun tipus de dispositiu digital, per exemple un microcontrolador, sense possibilitats de variar l'amplitud d'aquest senyal. La tècnica del PWM ens permetrà canviar la tensió d'alimentació del motor sense necessitat que el dispositiu digital canviï l'amplitud de la seva sortida.

Si en comptes d'alimentar el motor amb un senyal continu l'alimentem amb un tren de polsos, el motor pren com a valor d'alimentació el valor mitjà d'aquest senyal. Així, doncs, augmentar la velocitat del motor consisteix a augmentar el cycle de treball del tren de polsos, és a dir que estigui més estona a nivell alt que a nivell baix. Ho podeu veure exemplificat a la figura 39.

#### Cicle de treball

Quan un tren de polsos es troba la meitat del temps a nivell alt i l'altra meitat a nivell baix, parlem d'un cycle de treball del 50%. Si per exemple es troba només  $\frac{1}{4}$  del període a nivell alt, parlariem d'un cycle de treball del 25%.

Figura 39. Variació del cycle de treball d'un tren de polsos mitjançant la tècnica del PWM



En el primer cas el tren de polsos es troba la meitat del temps a nivell alt (10 V) i l'altra meitat a nivell baix (0 V), d'aquesta manera la tensió mitjana obtinguda és de 5 V. En augmentar el cycle de treball fins a un 75%, la tensió mitjana ha augmentat fins a 7,5 V.

### 6.2.5. Tipus de motors de contínua

Per a acabar anem a veure de manera molt breu quins són els tipus principals de motors de contínua que podem trobar. Bàsicament es diferencien per la manera com s'alimenten el rotor i l'estator. Els tipus que podríem diferenciar són els següents:

- Excitació paral·lel-independent. Rotor i estator s'alimenten amb dues tensions diferents (figura 40).
- Excitació paral·lel-*shunt*. Rotor i estator alimentats amb el mateix senyal (figura 41).
- Excitació sèrie. Els dos bobinats se situen en sèrie (figura 42).

- Excitació sèrie-paral·lel (*compound*). L'inductor disposa de dos debanats, un en sèrie amb el debanat del rotor i l'altre en paral·lel, de tal manera que els fluxos d'aquests dos debanats poden quedar sumats o restats (figura 43).

Figura 40. Connexió en paral·lel independent. Rotor alimentat amb *Alim 1* i l'estator amb *Alim 2*

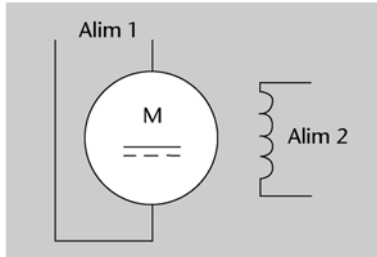


Figura 41. Connexió *shunt*. Rotor i estator alimentats amb el senyal *Alim*

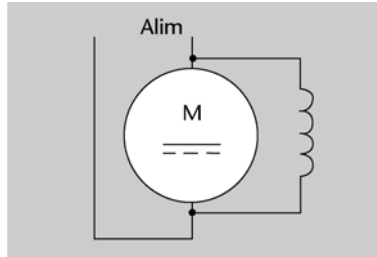


Figura 42. Motor sèrie

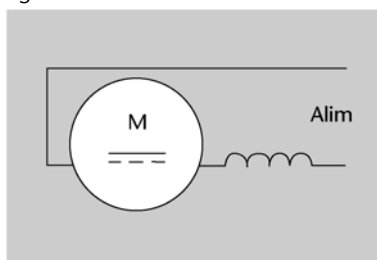
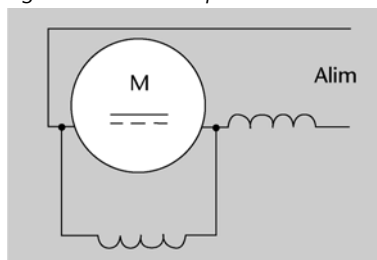


Figura 43. Motor *compound*



## 7. Introducció als sensors

En els següents subapartats farem una breu introducció al món dels sensors. Veurem que de sensors n'hi ha de tecnologies molt diferents però tots sempre amb una mateixa finalitat: la de captar una certa magnitud física per tal que la puguem mesurar, quantificar, visualitzar o senzillament saber-la. De tots els tipus de sensors que hi ha ens centrarem en l'estudi dels sensors de temperatura.

Al subapartat 7.1 introduïrem el concepte de sensor. Al subapartat 7.2 parlarem ja del cas concret dels sensors de temperatura i veurem exemples de sensors fabricats amb tecnologies molt diferents. Al subapartat 7.3 farem una breu pinzellada a altres tecnologies de mesura de temperatura menys accessibles a nivell pràctic per al món estudiantil però molt importants dins del món professional.

### 7.1. Què és un sensor?

Un sensor és aquell dispositiu que ens permet convertir el paràmetre físic que nosaltres volem mesurar (temperatura, humitat, pressió atmosfèrica, etc.) en una variable elèctrica (resistència, tensió, freqüència, etc.) que nosaltres puguem tractar per a saber la magnitud del paràmetre físic.

Així, doncs, en qualsevol muntatge d'instrumentació, ja sigui de control o només de mesura, es fa imprescindible la utilització de sensors per a poder controlar, o senzillament monitoritzar, tot allò que ens proposem.

La tecnologia dels sensors dependrà del tipus de paràmetre físic que vulguem captar; tot i així, tal com veurem, dins de cada tipus de sensors n'hi ha de basats en principis completament diferents, però tots amb el mateix objectiu: aconseguir un senyal que varii amb el paràmetre a mesurar i ens permeti treballar amb ell.

En el nostre cas ens interessarà sempre parlar d'aquell tipus de sensors compatibles amb un circuit electrònic; no tractarem aquells casos que directament no són compatibles amb un circuit elèctric.

Tot i que de sensors n'hi ha per a captar qualsevol tipus de magnitud, ens centrarem en els següents subapartats en l'estudi dels de temperatura, els quals ens donaran una visió general de les diferents tecnologies que es poden utilitzar en les mesures de magnituds. La temperatura es pot captar de maneres molt diferents, ja que l'única cosa que necessitem és algun material o dispositiu que experimenti algun canvi en sotmetre'l a un canvi de temperatura.

#### Exemples de sensors

Mirant al nostre voltant podem trobar molts exemples de sensors:

- El termòstat de la calefacció o l'aire condicionat.
- El sensor de temperatura ambient del nostre vehicle.
- El detector de presència de la porta automatitzada del garatge.
- El detector de sobrepès de l'ascensor.
- El sensor de nivell d'aigua de la cisterna del bany.

## 7.2. Sensors de temperatura

Un sensor de temperatura, tal com el seu nom indica, és aquell dispositiu encarregat de transformar la temperatura a mesurar en alguna variable observable o tractable electrònicament. És el que comunament anomenem un termòmetre.

Sensors de temperatura n'hi ha de molts tipus, tal com veurem seguidament. Tot i que cada sensor és adequat per a treballar en determinats rangs de temperatura, tan sols necessitem algun dispositiu o material capaç de variar alguna de les seves propietats amb la temperatura.

Hem de tenir present, però, a l'hora de fer mesures de temperatura, que aquestes es basen en els principis de la termodinàmica i en el fet d'assolir un cert equilibri tèrmic entre la temperatura del sensor i la temperatura a mesurar. Així doncs, sempre tindrem un cert error implícit a causa de l'assoliment d'aquest equilibri.

El camp dels sensors de temperatura és un dels més amplis dins el món de les mesures de magnituds. Podem trobar sensors basats en la dilatació de determinats materials (termòmetres de dilatació), sensors basats en materials que varien la seva resistència amb la temperatura, sensors que detecten radiació electromagnètica, etc.

En aquest subapartat, doncs, no podrem analitzar tots els tipus de sensors de temperatura, ens centrarem només en aquells que tenim més a prop nostre i sobretot en aquells que són compatibles amb un circuit electrònic de mesura o de control.

### 7.2.1. Termòmetres de dilatació

Els termòmetres de dilatació són aquells sensors de temperatura basats en el canvi de volum d'un determinat material en variar la temperatura. Molts d'aquests no són del nostre interès ja que, tal com avançàvem, no són compatibles amb un circuit electrònic.

Un bon cas de sensor no compatible electrònicament són els termòmetres de mercuri destinats, per exemple, a aplicacions de mesura de temperatures fisiològiques: fem la lectura visualment però no podem fer un tractament posterior d'aquesta mesura a través d'un circuit electrònic. Tot i així, aquest cas ens servirà per a acabar d'entendre el que comentàvem anteriorment de l'error implícit en la mesura: el termòmetre ha d'assolir un cert equilibri tèrmic amb el cos; si no, la lectura no és correcta. Per tant, abans de fer la lectura hem d'esperar una estona per a minimitzar aquest error.

#### Detecció domèstica de la temperatura

Imaginu que voleu detectar de manera aproximada la temperatura de l'aigua; amb un simple ou en tindríem prou per a saber per exemple si la temperatura és propera als 100°C.

#### Termòmetre de mercuri

En el cas del termòmetre de dilatació de mercuri, el material sensible a canvis de volum (el mercuri) es troba emmagatzemat dins de vidre. En construir un termòmetre ens hem d'assegurar sempre que el contenidor tingui un coeficient de dilatació molt més petit que no pas el del material sensible, si no s'estaria dilatant tot el termòmetre i no només el mercuri!

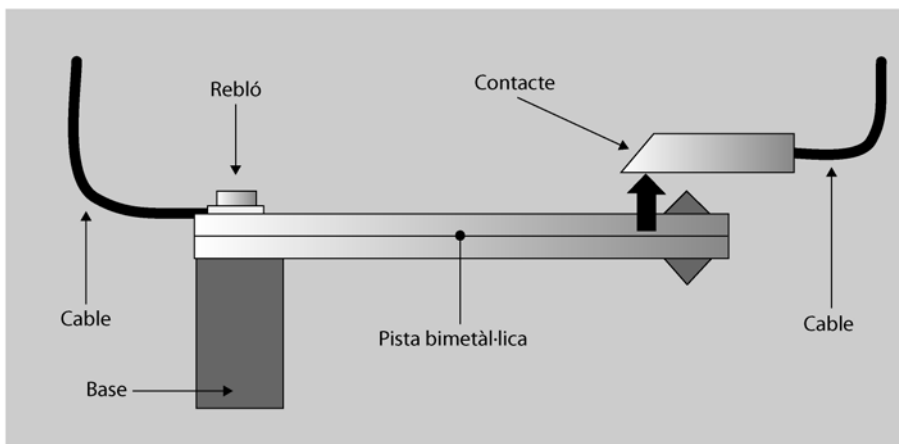
Dins dels termòmetres de dilatació, però, sí que hi ha un tipus de sensors de gran interès electrònic i del qual podreu detectar moltes aplicacions al vostre voltant: es tracta dels bimetalls.

Un bimetall consisteix en la unió de dues làmines de metalls diferents, amb coeficients de dilatació amb la temperatura molt diferents. Quan aquesta unió està sotmesa a un canvi de temperatura, els metalls es dilaten de manera desigual pel fet que el coeficient de temperatura és diferent. El resultat és una curvatura de tota l'estructura.

La curvatura de l'estructura és la que ens indica la temperatura mesurada. L'aplicació més familiar d'aquest sensor són els termòstats dels aparells d'aire condicionat o calefacció que tenim a casa.

Com que es tracta d'elements metàl·lics, podem aprofitar aquesta curvatura per a fer que un circuit entri en conducció, tal com es mostra en l'exemple de la figura 44, on podeu veure un bimetall que contactarà o no amb un cable en funció de la seva curvatura. Aquest sistema es podria correspondre perfectament al termòstat de qualsevol sistema de calefacció en el qual el sistema entraria en funcionament quan el bimetall contactés amb el cable.

Figura 44. En variar la temperatura el bimetall es corba i fa que el circuit entri en conducció, ja que el bimetall tocarà el contacte del cable.



**Figura 44**

Tot i que a la figura 44 veiem el bimetall com una petita tira metàl·lica, la veritat és que poden prendre formes molt diferents, és molt habitual veure estructures en forma d'espiral, el canvi de dilatació fa llavors que l'espiral estigui més o menys cargolat.

Els materials més habituals en la fabricació de bimetalls són el llautó, l'acer i l'invar. Els marges d'aplicació oscil·len entre els  $-75^{\circ}\text{C}$  i els  $450^{\circ}\text{C}$  amb una exactitud d' $1^{\circ}\text{C}$ .

### 7.2.2. Sensors basats en materials resistius metàl·lics

Els metalls presenten un coeficient de temperatura positiu com a resultat al canvi de dimensions i de resistivitat que experimenten en variar la temperatura. En altres paraules, la resistència d'un metall augmenta en augmentar-ne la temperatura. Podem aprofitar aquest fet per a obtenir un dels tipus de sensor de temperatura més comú, l'RTD (*resistor thermal detector*, resistència detector tèrmic).

$$\text{Graus Kelvin (K): } K = ^{\circ}\text{C} + 273$$

Si el marge de temperatures a mesurar és prou reduït, el canvi de resistència és proporcional a la temperatura:

$$R(T) = R_o (1 + \alpha(T - T_o)) \quad (28)$$

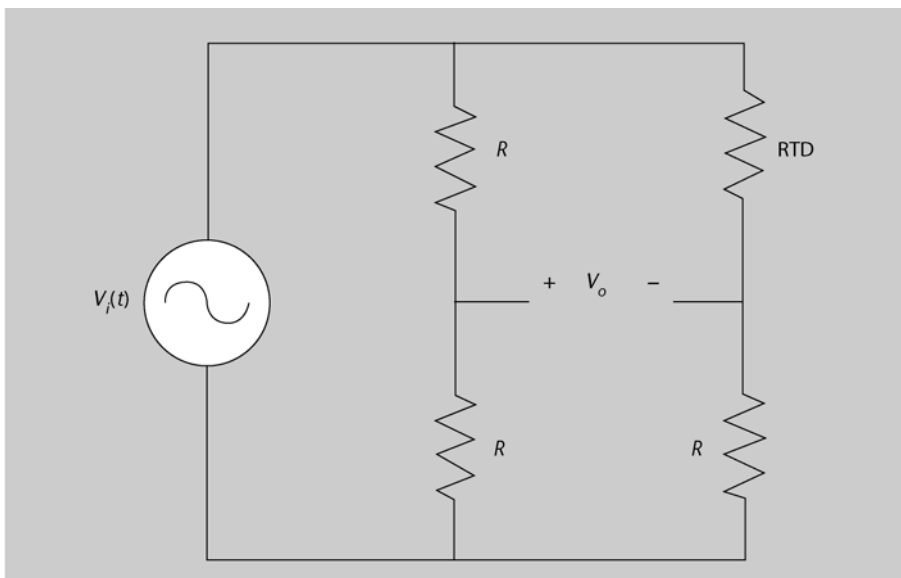
on:

- $R(T)$  representa la resistència a una temperatura  $T$  donada en graus Kelvin,
- $R_o$  és la resistència a una temperatura coneguda  $T_o$  (també en graus Kelvin) i
- $\alpha$  és el coeficient de temperatura del metall que ens indica la variació de resistència en variar la temperatura.

L'RTD més conegut és la Pt100 o sonda de platí, a causa del fet que el platí és un metall que presenta un marge de mesura lineal més ampli amb una sensibilitat acceptable. El nom li ve donat perquè a  $0^\circ\text{C}$  presenta una resistència de  $100\ \Omega$ . El marge de temperatures que pot mesurar oscil·la entre els  $-200^\circ\text{C}$  i els  $850^\circ\text{C}$  amb una exactitud de  $0,01^\circ\text{C}$ .

Com que els RTD són una resistència, els podem connectar com un element més en un pont resistiu. Així obtenim una tensió que varia amb la temperatura. Ho podem veure il·lustrat a la figura 45.

Figura 45. Pont resistiu amb un RTD. Obtindrem una tensió de sortida ( $V_o$ ) la qual serà funció de la temperatura (equació 29).



$$V_o = V_i \left( \frac{1}{2} - \frac{R}{R + \text{RTD}} \right) \quad (29)$$

#### Observació

En aquest cas el coeficient de temperatura ens indica la variació que experimentarà la resistència en variar un grau la temperatura.

#### Pont de Wheastone

El pont de resistències en el qual hem connectat l'RTD s'anomena Pont de Wheastone. És molt emprat sobretot amb sensors resistius, inductius i capacitius. Fixeu-vos que si la resta de resistències del pont l'ajustem al valor que pren l'RTD a la temperatura de referència, llavors a aquesta temperatura la tensió de sortida seria 0. És el que s'anomena el '0' de la lectura.

### 7.2.3. Sensors basats en resistències semiconductores.

#### Termistors

Els termistors són sensors de temperatura basats en elements semiconductors als quals s'ha aplicat un dopatge amb impureses de tipus 'n' o de tipus 'p'.

Si el dopatge no és intens, en augmentar la temperatura s'incrementa el nombre de portadors, cosa que redueix la resistència elèctrica. Parlem llavors d'un NTC (*negative thermal coefficient*, coeficient de temperatura negatiu).

En canvi, si aquest dopatge és intens (ja sigui de tipus 'n' o 'p') el semiconductor adquireix propietats metàl·liques. D'aquesta manera, presenta un coeficient de temperatura positiu. Llavors parlem dels PTC (*positive thermal coefficient*, coeficient de temperatura positiu).

#### Termistors NTC

Els termistors NTC són resistències amb un coeficient de temperatura negatiu i molt alt. Estan fets d'òxids del grup del ferro (Cr, Mn, Fe, Co o el Ni), els quals es transformen en semiconductors afegint-hi petites quantitats d'ions que tinguin una valència diferent. Atès el coeficient negatiu, la variació de la resistència en els termistors NTC és una exponencial decreixent:

$$R(T) = R_0 \cdot e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)} \quad (30)$$

on:

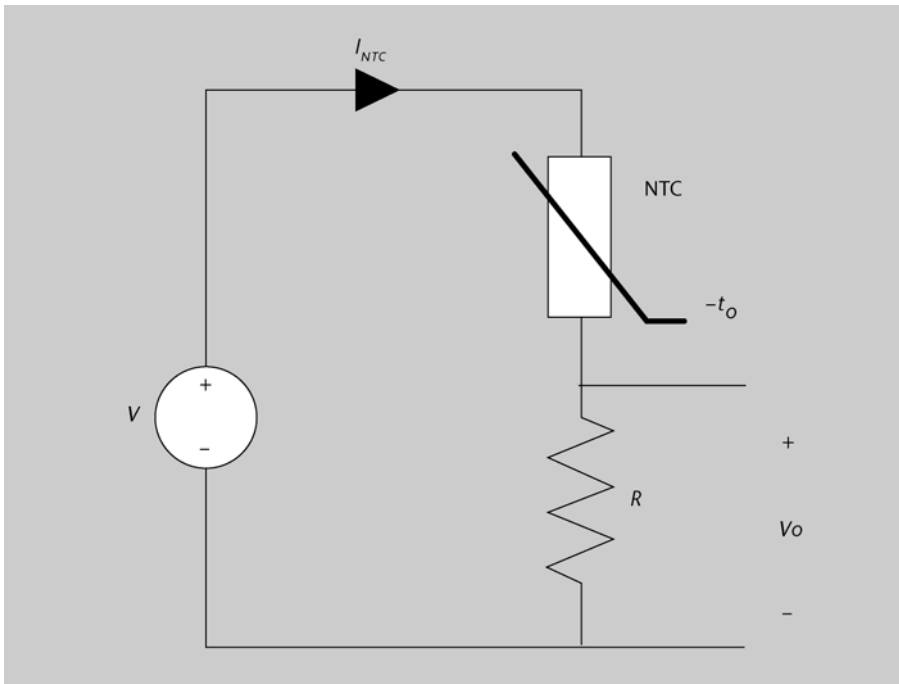
- $R(T)$  és la resistència del termistor a una temperatura  $T$  (en graus Kelvin),
- $R_0$  la resistència a una temperatura coneguda  $T_0$  (també en graus Kelvin) i
- $B$  és la constant del termistor que ens la facilitarà el fabricant del sensor.

A la pràctica, la  $B$  varia suaument amb els increments de temperatura, per aquest motiu, el fabricant ens facilita diferents valors de  $B$  segons el marge de temperatures en què vulguem treballar.

Podrem polaritzar el nostre NTC amb una font de contínua i una resistència en sèrie. L'anàlisi d'aquest circuit ens dona una tensió de sortida que dependrà de la resistència de l'NTC i, per tant, de la temperatura. Ho podeu veure representat a la figura 46, on teniu el circuit de polarització utilitzant una font i una resistència.



Figura 46. Circuit de polarització de l'NTC



Per a trobar la tensió de sortida del circuit de la figura 46 tan sols hem de fer un divisor de tensió de la tensió d'alimentació entre les dues resistències. Obtenim una tensió que depèn de la temperatura:

$$V_o = V \frac{R}{R + NTC} \quad (31)$$

Cal mantenir la tensió de treball del circuit a un valor no gaire elevat. Hem de tenir present que treballem amb resistències i que a més tensió d'alimentació, més potència dissipada i, per tant, més autoescalfament i, per tant, estarem falsejant la mesura de temperatura.

Una de les aplicacions més importants dels NTC és la mesura de nivells de líquids (la resistència de l'NTC varia segons si aquest es troba submergit o no, ja que la temperatura no és la mateixa).

### Termistors PTC

Els PTC són termistors amb un coeficient de temperatura positiu a causa d'una densitat d'impureses més elevada que en els NTC. El tipus de material més utilitzat per a la seva fabricació és el titanat de bari.

Per a entendre per què un PTC i un RTD tenen coeficient de temperatura positiu i, en canvi, un NTC negatiu, simplifiquem molt la situació: en un NTC el fet d'augmentar la temperatura fa que les poques impureses circulin més ràpidament. En un PTC i un RTD, en canvi, els portadors són més abundants; per tant, en augmentar la temperatura tendrien també a excitar-se i moure's

més de pressa, però com que n'hi ha més, topen entre ells, és a dir, es moles-ten, i, per tant, es dificulta la circulació de corrent i com a conseqüència aug-menta la resistència.

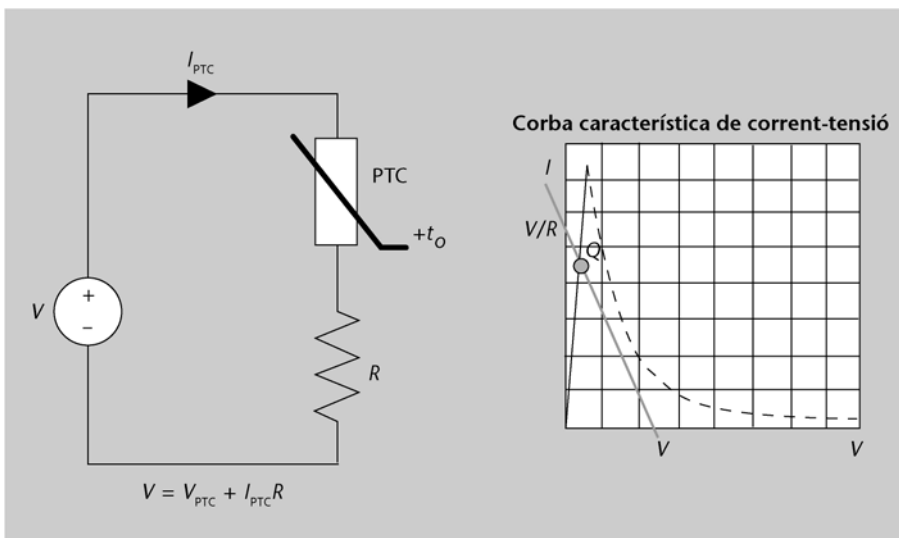
La resistència d'un PTC és donada per la següent expressió:

$$R(T) = R_0 \cdot e^{B(T-T_0)} \quad (32)$$

En aquest cas fixem-nos que en l'exponent no fem la inversa de la temperatu-ra; per tant, la resistència augmenta amb la temperatura.

El circuit de polarització pot ser igual al que hem vist per al cas de l'NTC i per a analitzar-lo podem utilitzar la característica corrent-tensió que ens dóna el fabricant. Si mirem la figura 47 veiem que el PTC es comporta com a resistèn-cia lineal només en el primer tram de la característica.

Figura 47. Circuit de polarització del PTC



Circuit de polarització del PTC i característica amb la recta de càrrega sobreposada, de manera que queda determinat un punt de treball Q. En l'eix vertical tenim ara  $I_{PTC}$  i en l'eix horitzontal  $V_{PTC}$ .

Fixeu-vos en què el símbol del NTC i del PTC es diferencien per un signe + o -.

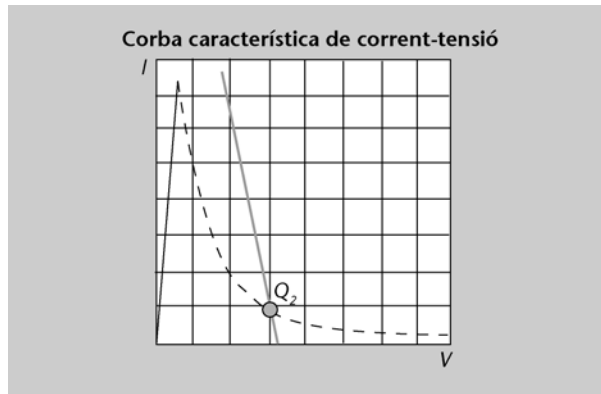
En aquest cas la forma de la característica del PTC fa que aquests es puguin uti-litzar en lloc de fusibles en circuits de protecció davant de:

- Curtcircuits.
- Augments de la tensió d'alimentació del circuit en què es troben.
- Augments de temperatura.

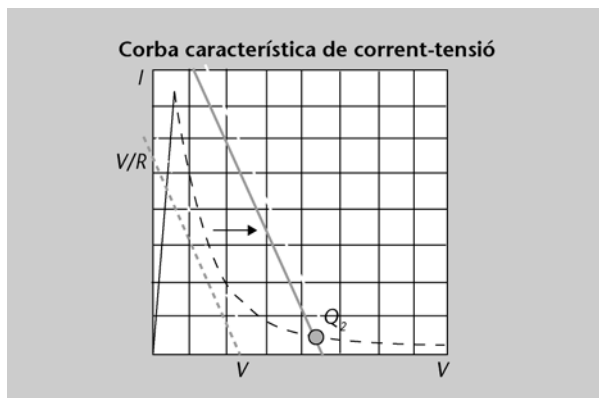
En situació normal el punt de treball seria Q (figura 47). Ara bé, davant d'un curtcircuit del circuit que estigui protegint (podem simplificar pensant que R disminueix molt), el quocient  $V/R$  augmentaria i faria augmentar el pendent de la recta de càrrega. Aleshores ens quedem amb un únic punt de treball ( $Q_2$ ) en la zona baixa de la característica en el qual el corrent que circularia pel circuit a protegir ens ha quedat limitat a un valor molt petit (figura 48).

Figura 47

Analitzant la malla del circuit de polarització obtenim el que anomenem la recta de càrrega del circuit que teniu a la figura 47. Aquesta recta la podem sobreposar a la característica corrent-tensió del fabricant. Fixeu-vos que tallaria l'eix vertical (quan  $V_{PTC} = 0$ ) a un valor  $V/R$ , i l'eix horitzontal (quan  $I_{PTC} = 0$ ) a un valor  $V$ . La intersecció amb la característica del fabricant ens dóna el punt de treball Q del PTC.

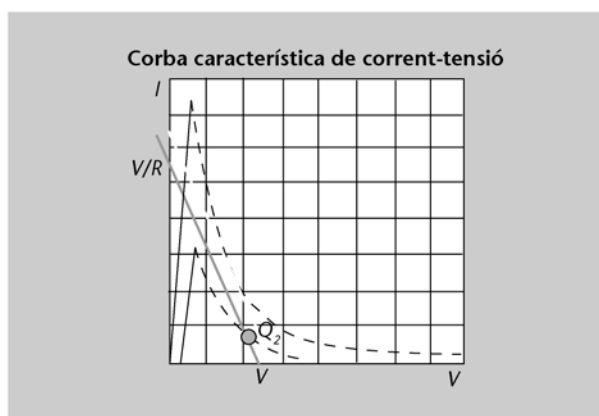
Figura 48. Pendent de la recta de càrrega en disminuir  $R$ 

Davant un augment de la tensió d'alimentació, la recta de càrrega es desplaçaria paral·lelament (figura 49), i ens quedem de nou amb el punt de treball  $Q_2$  on es limita el corrent del circuit.

Figura 49. Desplaçament de la recta de càrrega davant un augment de  $V$  (tensió d'alimentació del circuit a protegir)

Si es donés un augment de temperatura, llavors és la característica la que varia desplaçant-se cap avall. De nou, ens quedaríem amb un únic punt de treball  $Q_2$  (figura 50).

Figura 50. Desplaçament de la característica en augmentar la temperatura



#### 7.2.4. Termòmetres basats en unions de materials semiconductors

Vegem ara un altre tipus de sensors de temperatura basats en semiconductors, en aquest cas, però, en unions de materials semiconductors de tipus diferent.

Tal com podem veure en les especificacions de qualsevol component basat en unions de semiconductors, per exemple els díodes o els transistors, el comportament de les unions no és independent de la temperatura. Aquest fet, quan treballem amb díodes o transistors, és un inconvenient, ja que distorsiona el comportament del nostre circuit davant de canvis, per exemple, de la temperatura ambient. Ben gestionat, però, aquest efecte es pot aprofitar per a crear un sensor de temperatura.

Per a una unió PN polaritzada en directe amb un corrent constant, la tensió decreix de forma pràcticament lineal amb la temperatura en una proporció de  $-2,2 \text{ mV}/^\circ\text{C}$  en el cas del silici. Si la unió PN la polaritzem en inversa a tensió constant, llavors el corrent invers es duplica pràcticament cada  $10^\circ\text{C}$ .

Penseu, per exemple, en la tensió col·lector-emissor d'un transistor:

El corrent  $I_o$  té també una variació important amb la temperatura; per aquest motiu la majoria de sensors es dissenyen de manera que la tensió col·lector-emissor de dos transistors idèntics quedin restades, d'aquesta manera en restar els dos logaritmes simplifiquem el terme  $I_o$  i la tensió total varia només de forma proporcional amb la temperatura.

$$V_{ce} = \frac{KT}{q} \ln\left(\frac{I_c}{I_o}\right) \quad (33)$$

on:

- $K$  representa la constant de Boltzman,
- $T$  la temperatura,
- $q$  la càrrega de l'electró,
- $I_c$  el corrent de col·lector i
- $I_o$  el corrent invers de polarització.

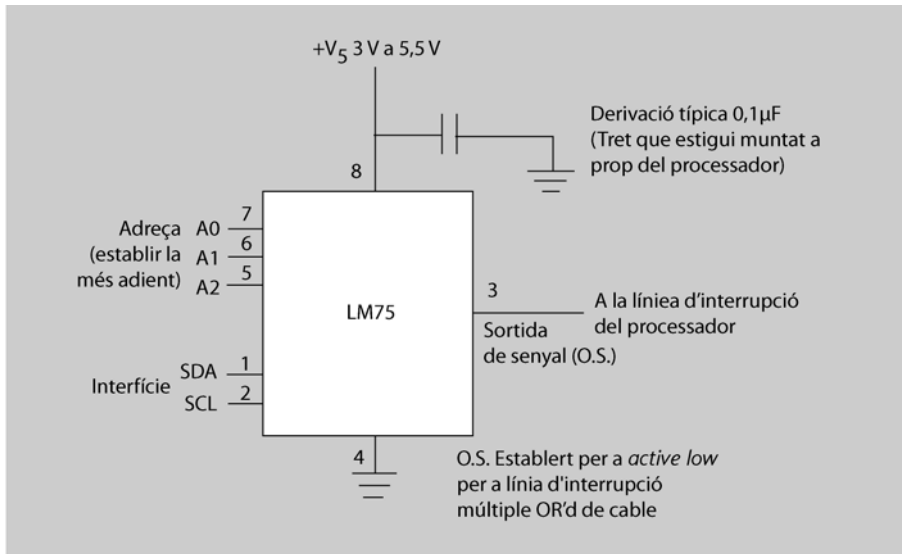
Així doncs, atesa una polarització constant del transistor ( $I_c$  fix), tindríem que la tensió entre col·lector i emissor variaria amb la temperatura.

Tot i que un simple díode es pot utilitzar com a sensor de temperatura, el mercat de semiconductors ofereix uns dispositius integrats que aprofiten les característiques anteriors i presenten un comportament molt millor com a sensor de temperatura que un simple díode o transistor.

Alguns sensors semiconductors de temperatura presenten una sortida de tensió proporcional a la temperatura, com la sèrie LM35 de National que subministra una tensió de  $10 \text{ mV}/^\circ\text{C}$ .

La tensió del LM35 pot atacar directament uns convertidors o un amplificador o qualsevol tipus d'etapa visualitzadora.



Figura 53. Senyals característics del bus I<sup>2</sup>C: SDA i SCL

Aquest petit integrat amb sensor de temperatura i interfície I<sup>2</sup>C ens pot ser molt útil per a aplicacions de temperatura a través d'un microcontrolador.

### 7.2.5. Altres tipus de sensors de temperatura

Tal com hem pogut veure en els subapartats anteriors, la mesura de temperatura es pot fer de maneres molt diferents. No podem acabar aquest mòdul sense parlar de com a mínim unes quantes maneres més que potser no són tan accessibles per a nosaltres (tant des del punt de vista econòmic com de disponibilitat) com les que hem explicat en els subapartats anteriors, però no per això són menys importants des d'un punt de vista industrial i professional.

Destacarem en aquest darrer punt els termoparells, els termòmetres de quars i els sensors de temperatura per detecció de radiació electromagnètica.

#### Termoparells

Els termoparells són transductors de temperatura que destaquen pel seu ampli camp de mesura (−270°C a 3.300°C), així com pel fet que són completament passius i no necessitem alimentar-los. Es basen en el fet que quan la unió de dos metalls diferents se sotmet a un gradient de temperatura podem detectar entre els seus terminals una tensió anomenada tensió termoelèctrica, la qual és proporcional al gradient de temperatura. Un dels inconvenients, però, d'aquests sensors és el fet que requereixen una temperatura de referència.

Es poden diferenciar molts tipus de termoparells segons els dos metalls amb què estiguin fabricats, els fabricants els identifiquen amb lletres majúscules.

Així per exemple: un termoparell de tipus E està format per cromel i constantà, un de tipus J és ferro i constantà, etc. Els fabricants faciliten sempre unes taules d'on podem extreure la tensió del termoparell quan se sotmet a un gradient donat entre la temperatura a mesurar i la de referència (normalment a 0°C). A la figura 54 teniu un exemple de termoparell.

### Termòmetres de quars

Els cristalls de quars es poden també aprofitar per a fer sensors de temperatura. Un rellotge amb cristall de quars pot tenir petites oscil·lacions en el seu comportament si el sotmetem a temperatures extremes. Aquest fet que treballant amb rellotges és un inconvenient, ara és un bon avantatge per construir sensors de temperatura. Aquesta dependència és molt lineal en el marge de -80°C a 250°C.

El mètode de mesura consisteix a comparar la freqüència del cristall sotmès a la temperatura de mesura amb la freqüència d'un altre cristall aïllat tèrmicament. L'exactitud que es pot arribar a tenir és de 0,075°C, i aquesta característica els converteix en els sensors més exactes dins del seu marge de temperatures. A la figura 55 podeu veure l'aspecte d'aquests sensors.

#### Exactitud i precisió

Convé no confondre exactitud i precisió. L'exactitud és la diferència entre el valor mesurat i el valor real. La precisió és la diferència entre valors repetitius de la mesura. Imagineu que voleu mesurar una temperatura ambient de 25°C amb dos termòmetres diferents i amb cadascun fem 4 mesures diferents:

- Les 4 mesures del termòmetre A són: 24°C, 25°C, 26°C i 25,5°C.
- Les mesures pel termòmetre B són: 30°C, 30,1°C, 30,05°C i 30°C.

Fixeu-vos que de l'A podríem dir que és força exacte, però en canvi és poc precís (tenim diferències de fins a 2° entre les mesures). El B, en canvi, és molt poc exacte (s'allunya molt de la mesura real) i per contra és molt precís (mesures repetitives donen valors molt semblants).

### Sensors per radiació electromagnètica

Tots els termòmetres que hem descrit fins aquest punt requereixen el contacte amb el cos o l'ambient del qual es vol detectar la temperatura. Per aquest motiu no podríem mesurar temperatures més enllà dels punts de fusió dels materials utilitzats en la fabricació del sensor.

Tots els cossos emeten una certa radiació electromagnètica que depèn de la temperatura i naturalesa del cos. El darrer tipus de sensors dels quals parlem, sensors per radiació electromagnètica, aprofiten aquest fet per a detectar la temperatura dels cossos.

Les tecnologies en aquest camp són també molt diverses i no entrarem en més detalls. En són exemples els piròmetres òptics, els bolòmetres, els detectors de fotons, etc.

Figura 54. Aspecte extern d'un termoparell



#### Material piezoelèctric

Un cristall de quars és el que s'anomena un material piezoelèctric: si se sotmet a una diferència de potencial experimenta una deformació o si se li aplica una deformació dóna una certa diferència de potencial. Altres tipus de sensors, com els de força, també aprofiten aquesta propietat del material piezoelèctric.

Figura 55. Aspecte d'un sensor de temperatura de quars



#### Termografia

La termografia és una ciència que es basa en la reconstrucció d'imatges a partir de la radiació emesa. És una aplicació directa del que estem explicant en aquest subapartat.

**Exemple**

Acabarem amb un exemple que ens pot ser força familiar. Es tracta de molts dels sensors de presència de les alarmes que hi ha instal·lats als habitatges. Alguns són autèntics detectors de presència que es basen en el rebot d'una ona d'ultrasons per a detectar moviment. Altres, però, no detecten moviment, sinó que el que detecten és la radiació electromagnètica emesa pels cossos; d'aquesta manera no detecten la presència del lladre sinó la seva escalfor.



## Resum

En l'apartat 1 hem pogut analitzar la necessitat del disseny utilitzant línies de transmissió en el moment que la longitud d'ona dels nostres senyals ho fa necessari.

S'han donat les eines necessàries per a poder començar el disseny d'una placa de circuit imprès on hi hagi el requeriment d'utilitzar algun tipus de línia de transmissió *microstrip* o *stripline*, ja sigui en circuits unipolars o en circuits que utilitzin senyals diferencials.

Hem après a calcular les impedàncies característiques d'aquestes línies per a tenir una adaptació correcta amb les càrregues que hi tinguem connectades al final.

En l'apartat 2 hem pogut analitzar amb força detall les diferents etapes que componen el disseny d'un PCB. Tot i que no ens hem centrat en cap eina EDA concreta, les passes vistes són comunes a qualsevol entorn de disseny.

Hem vist que les etapes bàsiques consisteixen en la creació de l'esquemàtic, la generació del *netlist*, la realització del *layout* i finalment l'obtenció de la documentació necessària perquè el fabricant ens pugui fabricar el PCB. Heu de tenir sempre present que, prèviament al que acabem de dir, hem de tenir sempre disponible una biblioteca de components on tinguem disponibles tots els símbols i *footprints* dels components del nostre disseny electrònic.

En l'apartat 3 hem pogut analitzar primer de tot les dues operacions bàsiques en les quals es basa la conversió analògic-digital i digital-analògic: la quantificació i la codificació.

Hem pogut veure també l'exemple d'un parell de convertidors analògic-digital compatibles amb etapes visualitzadores, la qual cosa els fa molt útils en aplicacions de mesura en les quals volem visualitzar el valor d'algun paràmetre.

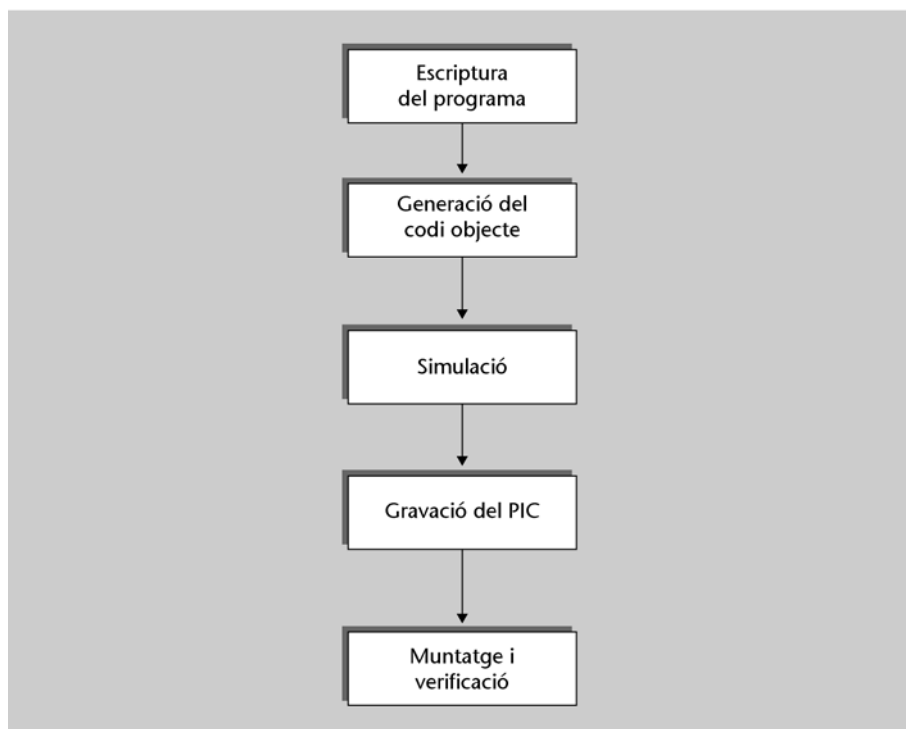
En darrer lloc hem vist el funcionament i l'estructura interna d'un convertidor digital-analògic.

En l'apartat 4 hem pogut fer un breu repàs a l'origen dels dispositius digitals programables com els GAL per a entendre'n millor el funcionament i les possibilitats de disseny que ens ofereixen.

Hem analitzat també el desenvolupament d'aplicacions amb ispLEVER, eina subministrada per Lattice, que ens permet treballar amb molts dels seus dispositius digitals.

Al llarg de l'apartat 5 s'ha evidenciat que hi ha moltes solucions en el mercat per a facilitar cadascuna de les etapes del procés de desenvolupament. Algunes són gratuïtes i unes altres són de pagament. Algunes les desenvolupa la mateixa empresa Microchip i unes altres les desenvolupen terceres empreses. Serà l'experiència la que ens farà optar per una solució o l'altra, i en qualsevol cas caldrà estar al corrent de les diverses noves aplicacions que continuen apareixent al mercat.

A la figura següent teniu un resum del procés de desenvolupament d'un sistema amb microcontrolador. El procés de desenvolupament d'un sistema amb microcontrolador PIC pot semblar molt complex i difícil; tanmateix, tal com ja hem comentat al principi de l'apartat 5, les aplicacions amb microcontroladors PIC són moltes i molt diverses. Això fa que avui dia sigui molt fàcil trobar tota la informació necessària perquè el desenvolupament de la nostra aplicació sigui tan senzill com es pugui.



Aquesta figura és un resum del procés de desenvolupament d'un sistema amb microcontrolador, on veiem la seqüència bàsica d'escripció del codi, la seva compilació per a generar el codi objecte i la gravació d'aquest en el dispositiu per a fer el muntatge final. Recordeu que també hi ha una etapa intermèdia, un cop generat el codi objecte, que és la seva simulació.

En l'apartat 6 hem analitzat el principi de funcionament dels motors de contínua i hem pogut veure quines són les parts. Hem destacat també els diferents mètodes de control que tenim per a variar la velocitat, i hem destacat, sobretot, el control amb PWM, el qual és molt senzill d'acoblar a un dispositiu digital.

Hem parlat també dels diferents tipus de motor de contínua que podem trobar al mercat segons com s'alimenten rotor i estator hem destacat, d'altra banda, que els motors són màquines reversibles que poden funcionar també com a generadors.

En l'apartat 7 hem donat una àmplia visió de les diferents tecnologies que podem trobar dins del món dels sensors de temperatura, anant des de sensors purament mecànics com són els bimetalls fins a sensors basats en la detecció de radiació electromagnètica.

Cal destacar, per a petites aplicacions de mesura, la gran versatilitat i facilitat d'ús que ens presenten en general els sensors resistius, ja sigui metàl·lics (Pt100) o d'elements semiconductors (PTC i NTC).

Hem de tenir present que l'elecció d'un sensor de temperatura haurà d'anar sempre d'acord amb el marge de temperatures que necessitem mesurar i, per tant, ens hem d'assegurar sempre que el sensor compleix els requeriments exigits consultant les especificacions facilitades pel fabricant.



## Bibliografia

*Data acquisition and control handbook* (2001). Data Acquisition and Control Handbook.

**Pérez, M. Álvarez; Campo, J.; Ferrero J.; Grillo, F.** *Instrumentación electrónica*. Thomson.

**Recasens, A.; González, J.** *OrCAD Capture y layout v.9.2*. Thomson.

**Marven, G.; Ewers, G.** (1993). *A simple approach to digital signal processing*. Texas Instruments.

**Artigas, J.; Barragán, L.; Orrite, C.** (2002). *Electrónica digital*. Prentice Hall.

**Stallings, W.** (1993). *Computer Organization and Architecture*. MacMillan.

**Cuaderno técnico** (núm. 207) Barcelona: Schneider Electric (2005).

**Broetz, D.** (1998). *Differential Impedance*. Printed Circuit Design. Miller Freeman Publications.

*National Semiconductors. Transmission line RAPIDESIGNER operation and applications guide*. AN-905.





