

---

# Capítol 6

---

## Cobertura de la Xarxa Booleana

*En aquest capítol s'estudiarà el procés de cobertura d'una Xarxa Booleana bo i minimitzant una funció de cost que implica un compromís, proposat per l'usuari, entre els diferents objectius de disseny. S'analitzaran diverses estratègies de cobertura de la Xarxa Booleana i es compararan els resultats d'aplicar-ne unes o altres.*

### Introducció

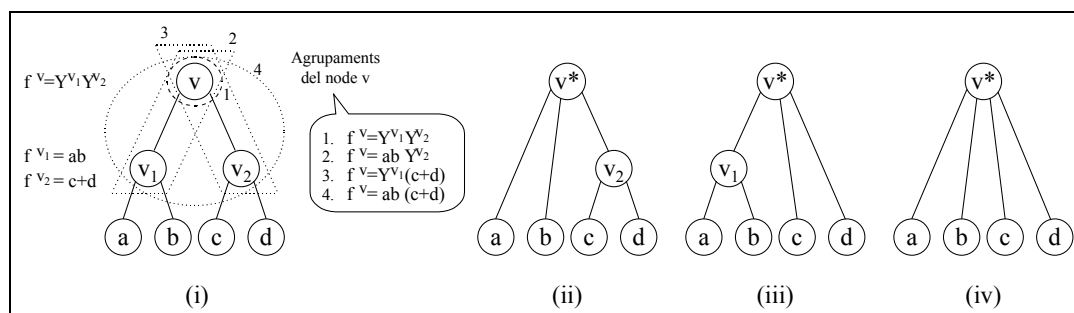
El procés de mapatge tecnològic d'una Xarxa Booleana consisteix bàsicament en la generació d'un conjunt d'agrupaments pels nodes de la xarxa, i en la posterior selecció i implementació d'un subconjunt d'aquests agrupaments, de manera que s'aconsegueixi un circuit equivalent a la Xarxa Booleana original que minimitzi una funció de cost depenent dels paràmetres tecnològics i dels objectius de disseny. L'operació de selecció dels agrupaments necessaris per implementar la Xarxa Booleana es coneix amb el nom de cobertura de la Xarxa Booleana.

El procediment de cobertura de la Xarxa Booleana s'ha realitzat de maneres ben diverses. L'optimització en base a transformacions locals ha estat aplicada en [LBK88]. DAGON [Keu87] va aplicar idees sorgides de l'optimització de codi en els compiladors al problema de cobertura d'una Xarxa Booleana. L'algoritme proposat divideix la Xarxa Booleana en un bosc d'arbres, on l'arrel de cada arbre es correspon amb un node que té un ventall d'entrada més gran que un. Utilitzant un aparellament estructural i programació dinàmica, cada arbre pot ser cobert òptimament. L'optimalitat, però, es perd en els límits dels arbres. Aquest tipus d'algoritme sol requerir una fase d'optimització global de la fase per tal de reduir el número d'inversors introduïts en el circuit. MISII [BRSW87, DGRSW87] utilitza un algoritme similar, però introdueix dos inversors en les connexions de la Xarxa Booleana per tal de permetre una millor assignació de fase en les funcions de cada node.

En aquest capítol s'estudiarà el problema de la cobertura de la Xarxa Booleana des d'un punt de vista totalment diferent. Es reduirà a un problema d'optimització d'una funció de cost. El capítol està organitzat de la següent manera. En primer lloc, es presentarà una definició formal del problema. A continuació s'estudiaran diferents alternatives per la cobertura de la Xarxa Booleana i es compararan els resultats obtinguts usant cadascuna d'elles. Finalment, s'extrauran les conclusions pertinents.

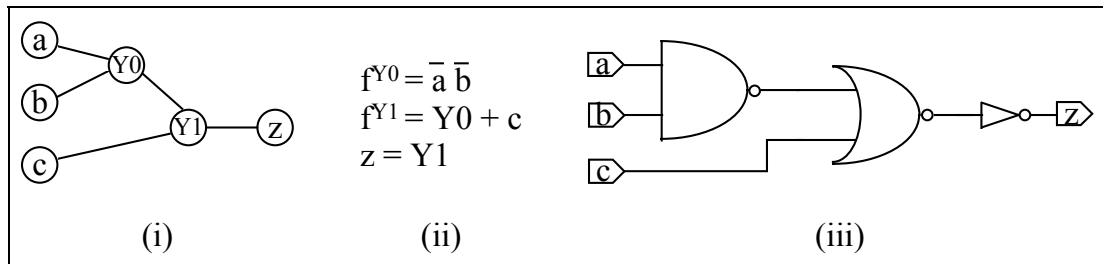
## Definició del problema

La cobertura d'una Xarxa Booleana consisteix, bàsicament, en definir una partició de la Xarxa Booleana de tal manera que: a) per tot node de la partició s'esculli un agrupament i una fase d'implementació determinats; b) els agrupaments i la fase assignada als nodes en la partició cobreixin tots els nodes de la Xarxa Booleana original, i produeixin les mateixes funcions de sortida i c) es minimitzi una funció de cost dependent dels objectius de disseny.



**Figura 6-1** Els agrupaments definits per una Xarxa Booleana en (i) permeten obtenir diferents cobertures (ii), (iii), (iv).

Un exemple de la cobertura d'una Xarxa Booleana senzilla el podem veure en la Figura 6-1. En la subfigura (i) es poden veure els possibles agrupaments del node  $v$ , suposant que tots ells siguin implementables. Els nodes del ventall d'entrada de  $v$  només depenen d'entrades primàries, per la qual cosa l'únic agrupament possible per ells correspon al seu agrupament bàsic, és a dir, a la funció que realitzen. Les possibles cobertures de la Xarxa Booleana, considerant únicament la implementació directa dels nodes, sense canvis de fase, es mostren en la Figura 6-1. Si es selecciona l'agrupament 2 per  $v$ , s'obté la cobertura mostrada en (ii). Seleccionant el 3, (iii). Finalment, si s'escull l'agrupament 4 per  $v$ , s'obté (iv). És evident que cada cobertura de la xarxa suposarà un compromís diferent entre els objectius de disseny.



**Figura 6-2** Cobertura d'una Xarxa Booleana (i), considerant la implementació directa i complementària de les funcions en els nodes de la xarxa (ii). En (iii) es mostra una implementació on  $Y0$  i  $Z$  s'implementen directament i  $Y1$  de forma complementària.

En la Figura 6-1, i per facilitar la visió global del problema de la cobertura d'una Xarxa Booleana sense perdre's en els detalls, no s'han mostrat les cobertures que s'obtidrien al considerar les combinacions d'implementacions directes o complementàries dels agrupaments dels nodes i la possible introducció d'inversors en la xarxa. Considerem l'exemple de la Figura 6-2 i suposem que anem a implementar els agrupaments bàsics en els nodes de la Xarxa Booleana de la subfigura (i). Observem les funcions realitzades en cada node de la xarxa mostrades en la subfigura (ii). L'agrupament bàsic pel node  $Y0$  correspon a  $f^{Y0}$  i resulta ser una funció de fase negativa, per la qual cosa es pot implementar directament en una porta CMOS, representada per la porta NAND de la subfigura (iii). L'agrupament bàsic pel node  $Y1$  correspon a  $f^{Y1}$  i resulta ser una funció de fase positiva. Per a seva implementació directa, caldria transformar les variables de fase positiva (les dues en el suport) en variables de fase negativa bo i introduint inversors a les entrades. Aquesta implementació requereix dos inversors i la implementació de la nova funció de fase negativa. Alternativament, es pot generar la funció complementària en el node  $Y1$ . Si  $f^{Y1}$  està representada en un BDD, obtenir la funció complementària equival simplement a intercanviar els nodes terminals; en l'expressió mostrada en la subfigura (ii) caldria fer el complement i aplicar les lleis de De Morgan. Sigui com sigui, la funció resultant és de fase negativa i pot ser implementada directament per la porta NOR visible en la subfigura (iii). Finalment, cal implementar la funció del node de sortida  $Z$ . El seu agrupament bàsic correspon a  $f^z$ . En aquest cas,  $f^z$  era igual a la sortida del node  $Y1$ , per la qual cosa es podria haver implementat com una connexió directa, sense requerir cap porta. Però la implementació escollida per  $Y1$  ha estat la complementària, i per tant no es disposa de la variable  $Y1$ , sinó de  $\bar{Y1}$ . La implementació de  $f^z$  requerirà l'inversor final mostrat en la subfigura (iii).

A la vista dels exemples mostrats, és evident que la selecció dels agrupaments i l'elecció de la seva forma directa o complementària, a implementar en cada node de la Xarxa Booleana, donarà lloc a circuits amb diferents valors pels objectius de disseny: àrea, retard i consum de potència. La selecció d'una o altre cobertura haurà de venir determinada per una funció de cost que tingui en compte aquests paràmetres de disseny i els requeriments del sistema on s'hagi d'utilitzar el circuit. Aquesta funció de cost haurà de guiar el procés de partició de la Xarxa Booleana i també la selecció dels agrupaments i fase d'implementació dels nodes que formin part de la partició.

Formalment, una *coberta* d'una Xarxa Booleana  $G=(V, E)$  consisteix en una quartet  $(U, \phi, W, G^*)$ , on  $U$  representa el subconjunt dels nodes de  $G$  que apareixen implementats en  $G^*$ ;  $\phi$  és una funció injectiva que relaciona els nodes implementats de  $G$  amb alguns dels nodes de  $G^*$ ;  $W$  correspon al conjunt de nodes per als quals es requereix un inversor en  $G^*$ ; finalment,  $G^*$  és una implementació de  $G$ , que no és altra cosa que una Xarxa Booleana funcionalment equivalent a  $G$  en la qual cada node conté una funció de fase negativa que pot ser directament implementada en una porta CMOS d'una sola etapa. Una coberta de  $G$  ha de satisfer les següents condicions:

- i)  $W \subseteq U \subseteq V$
- ii)  $G^*=(V^*, E^*)$  és una Xarxa Booleana funcionalment equivalent a  $G$ .
- iii) és una funció injectiva que identifica els nodes  $u \in U$  amb nodes  $u^* \in V^*$  que implementen l'agrupament i la fase escollits per al node  $u$ . Cal destacar que els nodes  $u^*$  es defineixen de tal manera que siguin sempre directament implementables en la tecnologia destí. Per això, si l'agrupament del qual s'originen no és ja una funció de fase negativa, caldrà convertir-l'ho, introduint noves variables corresponents al complement de les variables de fase positiva o binària presents en l'agrupament original. Els nodes  $u^*$  es poden classificar en dos subconjunts, segons la fase implementada de l'agrupament escollit pel node  $u$ :

$$V_+^* = \left\{ u^* \mid \phi(u) = u^*, u \in U \text{ s'implementa en fase positiva} \right\}$$

$$V_-^* = \left\{ u^* \mid \phi(u) = u^*, u \in U \text{ s'implementa en fase negativa} \right\}$$

on  $V_+^*$  és el conjunt de nodes implementats en forma directa i  $V_-^*$  és el conjunt de nodes implementats en forma complementària. Aquests conjunts han de complir les següents propietats:

$$V_+^* \cap V_-^* = \emptyset$$

$$\forall u \in U \exists ! u^* = \phi(u) \mid u^* \in V_+^* \cup V_-^*$$

- iv)  $V^* = V_+^* \cup V_-^* \cup I$ , on  $I$  representa el conjunt d'inversors necessaris per generar la fase correcta en les entrades dels nodes  $v^* \in V_+^* \cup V_-^*$ . Aquest conjunt es defineix com:

$$I = \left\{ I^{\phi(v)} \mid v \in W, \phi(v) \in V_+^* \cup V_-^*, Y^{I^{\phi(v)}} = \overline{Y^{\phi(v)}} \right\}$$

on  $I^{\phi(v)}$  és el node associat a l'inversor, i  $Y^{I^{\phi(v)}}$  correspon a la variable associada al node, que és justament el complement de la variable associada al node  $Y^{\phi(v)} \in V_+^* \cup V_-^*$ .

- v) Els conjunt d'entrades,  $X$ , i sortides,  $Z$ , primàries de  $G$  han d'estar presents en  $G^*$  i han de complir les següents propietats:

$$X \subseteq U, \forall x \in X \exists! \phi(x) \in V_+^*$$

$$Z \subseteq U, \forall z \in Z \exists! z^* = \phi(z) \left| \left( z^* \in V_+^* \right) \text{ o } be' \left( z^* \in V_-^*, \exists I^* \in I \right) \right.$$

- vi) Pel que fa als arcs  $e \in E^*$ , aquests han de complir que:

$$\forall e = (u, v), e \in E^* \Rightarrow \begin{cases} i) u \in V_+^* \cup V_-^*, v \in I \\ \text{ o } be' \\ ii) u \in V_+^* \cup V_-^*, v \in V_+^* \cup V_-^*, Y^{\phi^{-1}(u)} \in \text{Suport}\left(Y^{\phi^{-1}(v)}\right) \\ \text{ o } be' \\ iii) u \in I, v \in V_+^* \cup V_-^*, u = I^{\phi(w)}, w \in \text{Suport}\left(Y^{\phi^{-1}(v)}\right) \end{cases}$$

on el cas *i*) correspon a l'arc que uneix un node de  $G^*$  i l'inversor que aquest necessita.

El cas *ii*) correspon un arc que uneix dos nodes de  $G^*$  quan els dos provenen directament de nodes de  $G$ ;  $\phi^{-1}(v)$  correspon al node del qual s'ha implementat un agrupament que es guarda en  $v$ ;  $\phi^{-1}(u)$  identifica una variable del suport de l'agrupament implementat en  $v$ .

El cas *iii*) correspon a l'arc que uneix un inversor a un node de  $G^*$  que prové d'un node de  $G$ ;  $\phi^{-1}(v)$  correspon al node del qual s'ha implementat un agrupament que es guarda en  $v$ ;  $u$  correspon a un inversor d'una entrada de l'agrupament implementat en  $v$ .

Un cop definit el que és una coberta d'una Xarxa Booleana podem definir una *cobertura* d'una Xarxa Booleana  $G=(V, E)$  com el procés de trobar una coberta  $(U, \phi, W, G^*)$  que minimitzi una funció de cost determinada de la implementació  $G^*$ . Habitualment, el cost de la implementació estarà en funció del cost associat a cada node i a cada arc de la Xarxa Booleana  $G^*$ :

$$\text{Cost}(G^*) = \sum_{i=0}^{|V^*|} \text{cost}(v_i) + \sum_{j=0}^{|E^*|} \text{cost}(e_j) \quad (6-1)$$

La funció de cost associada a cada node i/o arc de la implementació tindrà en consideració els paràmetres de disseny: àrea, retard i dissipació de potència. És evident que aquests paràmetres suposen sovint objectius contradictoris, en el sentit que reduir l'àrea implicarà sovint augmentar el retard; i disminuir el retard, implicarà un increment d'àrea i un augment de la potència dissipada, etc... Per tant, el pes associat a cadascun dels paràmetres de disseny dependrà de l'aplicació específica que es vulgui donar al circuit implementat. Si el circuit ha d'anar integrat en un sistema alimentat per bateries, el paràmetre crític serà el consum de potència, mentre que per

a aplicacions en temps real el paràmetre crític serà probablement el retard màxim i, en totes les aplicacions es voldrà reduir l'àrea fins on sigui possible.

## **Modelat dels paràmetres tecnològics**

És evident que la funció de cost associada a una implementació dependrà també de la tecnologia destí i fins i tot de la metodologia de disseny. Efectivament, els compromisos entre els diferents paràmetres de disseny que es poden assolir al fer un mapatge tecnològic varien considerablement si aquest està orientat a un generador de mòduls o si s'utilitza una biblioteca de cel·les estàndard. Fins i tot en el cas d'utilitzar una biblioteca de cel·les, els compromisos assolibles variaran segons si la implementació final del circuit es fa en un mar de cel·les predifoses (*Sea-of-Gates*) o seguint una metodologia de cel·les estàndard predissenyades i precharacteritzades. A efectes de comparació, considerarem únicament dues alternatives de disseny: la síntesi orientada a generadors de mòduls i la síntesi usant una biblioteca de cel·les estàndard en metodologia *semi-custom*, ja que som les més populars.

És notori el fet que els models que poden ser utilitzats en el mapatge tecnològic orientat a generadors de mòduls són més senzills que els requerits en un mapatge tecnològic clàssic sobre una biblioteca de cel·les estàndard, tal com es podrà comprovar. És més, els models proposats són en gran mesura independents de tecnologia, la qual cosa els fa aptes per ser aplicats durant les primeres fases de la síntesi lògica. En concret, els models podrien ser utilitzats durant l'optimització lògica de la Xarxa Booleana per tal d'assolir millores relatives que proporcionessin un millor punt inicial pel mapatge tecnològic. En el cas de considerar un mapatge tecnològic sobre una biblioteca de cel·les, els models que poden ser utilitzats durant les primeres etapes de la síntesi lògica no estan tant ben correlacionats amb la implementació final i sovint es basen en mesures estadístiques [RRVC95, RRVC96].

En aquest apartat es presentaran les funcions de cost associades als paràmetres de disseny (àrea, retard màxim i dissipació de potència), pel cas del mapatge tecnològic orientat a generadors de mòduls. Es descriurà la motivació que porta a la utilització dels models proposats i es compararan amb els models utilitzats en el cas d'un mapatge tecnològic restringit a una biblioteca de cel·les.

### **Model per l'àrea**

En la funció de cost associada a l'equació (6-1) es poden distingir clarament dues contribucions al cost global. D'una banda, el primer sumatori fa referència al cost que suposen els nodes de la Xarxa Booleana, o el que és igual, les portes de la implementació. D'altra banda, el segon sumatori fa referència al cost associat als arcs de la Xarxa Booleana, que s'identifiquen a les connexions entre portes en la implementació del circuit. Per tant, el nostre estudi sobre el modelatge de l'àrea dividirà també aquesta en dues parts, la primera associada a les portes i la segona a les connexions.

A diferència del que passa en un mapatge tecnològic clàssic sobre una biblioteca de cel·les on l'àrea ocupada per cada cel·la és coneguda durant la cobertura de la Xarxa Booleana, en el mapatge tecnològic orientat a generadors de mòduls aquesta àrea encara no està fixada. Mentre que les cel·les d'una biblioteca han estat prèviament dissenyades i, per tant, les seves dimensions estan fixades i són conegudes durant el mapatge tecnològic, les cel·les que es generen durant el mapatge tecnològic orientat a generadors de mòduls no han estat encara dissenyades. Amb posterioritat al mapatge tecnològic, l'estructura de les cel·les serà definitivament establerta i els transistors que les formen seran dimensionats correctament per tal de poder atacar les capacitats visibles a la sortida. No serà fins després d'aquest procés que la dimensió real de les cel·les podrà ser coneguda. Per tant, durant el mapatge tecnològic orientat a generadors de mòduls, caldrà utilitzar una mesura indirecta de l'àrea.

Tradicionalment, quan l'àrea d'una porta no era coneguda en alguna etapa del procés de síntesi lògica, per exemple durant l'optimització de la Xarxa Booleana, ha estat estimada en base al número de literals d'una representació en suma de productes de la funció realitzada per la porta. Aquesta estimació es basa en el fet que el número de literals mesura, en certa manera, la complexitat de la funció i està relacionat amb el número de transistors requerits per implementar la porta [BHS90]. Però en el cas de considerar un mapatge tecnològic orientat a generadors de mòduls, el número de transistors necessaris per implementar una porta serà conegut, com hem vist en el capítol 4, i podrà ser utilitzat com a estimació de la complexitat de la porta i, per tant, de l'àrea ocupada per aquesta. És evident que aquesta estimació estarà molt més correlacionada amb l'àrea real que la basada en el número de literals.

Pel que fa a l'àrea ocupada per les connexions, cal dir que aquesta és difícil d'avaluar, ja que no podrà ser coneguda fins que les cel·les hagin estat col·locades i connectades en el circuit final. La complexitat d'estimació d'aquest paràmetre no depèn de si es considera un mapatge tecnològic orientat a generadors de mòduls o sobre una biblioteca de cel·les, ja que en els dos casos l'àrea ocupada pel connexionat dependrà de processos similars de posicionament i connexionat de cel·les. Diferents estratègies han estat proposades per fer una estimació de l'àrea ocupada per les connexions:

- **Estimació zero.** Aquesta tècnica simplement ignora l'àrea ocupada per les connexions. La síntesi lògica tradicional [BHS90, BRSW87] ha ignorat l'àrea ocupada per les connexions i s'ha concentrat en minimitzar l'àrea ocupada per les portes. La minimització de l'àrea global del circuit no està, en principi, garantida. L'optimització de la Xarxa Booleana identifica i extreu subfuncions comunes per tal de reduir l'àrea, però això augmenta el ventall de sortida per aquestes funcions i, possiblement, faci augmentar l'àrea ocupada pels connexionat. De totes maneres, l'evolució de les tecnologies proporciona noves i bones raons per ignorar l'àrea dedicada al connexionat. De fet, les noves tecnologies submicròniques solen permetre un número creixent de nivells de metall. Tres i quatre actualment, més en el futur. Noves eines de disseny aprofiten els nivells addicionals de metall i tendeixen a fer que l'àrea ocupada per les connexions es redueixi dràsticament, ja que la major part del connexionat es pot ser damunt de l'àrea ocupada per les cel·les. Els resultats presentats en [EUGM95] mostren que per una tecnologia de 0.5 $\mu$ m amb tres nivells de metall, l'àrea

ocupada per les connexions és menyspreable en comparació a l'àrea ocupada per les cel·les. Aquest fet serà més i més cert a mesura que nous nivells de metall i noves eines de disseny estiguin disponibles. En conclusió, per tecnologies amb més de dos nivells de metall, l'àrea dedicada a les connexions pot ser ignorada sense que això suposi un errors significatius.

- **Estimació usant el ventall de sortida.** Aquesta estimació relaciona el ventall de sortida d'una porta amb la complexitat del connexionat associat al senyal de sortida de la mateixa. Si s'incrementa el número de nodes en el ventall de sortida, s'incrementarà també la complexitat del connexionat associat. Hi haurà més portes a connectar i, el que és pitjor, algunes d'elles poden acabar posicionades lluny de la porta considerada. Una variació interessant d'aquest estimador [PP89] proposa predir el futur posicionament i connexionat de les portes usant no només el ventall de sortida de la porta considerada, sinó també el connexionat associat a les portes que es troben en aquest ventall. Els autors afirmen que la desviació respecte dels valors reals no excedeix el deu per cent. De totes maneres, l'algoritme és difícilment aplicable a la síntesi lògica degut a la seva complexitat en temps d'execució.
- **Estimació en base a dades estadístiques.** Aquesta estimació es basa en l'extracció estadística de dades sobre el connexionat de circuits ja fabricats i la seva aplicació als circuits que es volen implementar. Es construeixen diferents corbes que relacionen el ventall de sortida d'una porta amb la longitud de connexionat de la sortida de la mateixa, a partir dels circuits analitzats i variant la dimensió, tipus i característiques de disseny d'aquests circuits. Durant la síntesi lògica, aquestes corbes s'utilitzen per fer una estimació de la longitud de cada connexió. El ventall de sortida d'una porta i una estimació de la complexitat global del circuit permeten un procés d'interpolació que donarà l'estimació de l'àrea ocupada per cada connexió. Aquesta tècnica resultarà especialment eficient quan les dades estadístiques i el circuit que s'està sintetitzant usin les mateixes eines de posicionament i connexionat i s'implementin en la mateixa tecnologia o una de molt similar [RRVC95].
- **Estimació en base a un pre-posicionament de les cel·les.** Durant el mapatge tecnològic sobre una biblioteca de cel·les, [PB91] proposa la utilització d'una eina de posicionament global de les cel·les per tal de poder predir la longitud del connexionat associat a cada senyal. El mapatge tecnològic es fa sobre una Xarxa Booleana que conté només un conjunt reduït de cel·les bàsiques (p.e. *nands* de dues entrades i inversors). Les cel·les bàsiques són posicionades inicialment usant una eina de posicionament global. Aquest primer posicionament es modifica dinàmicament a mesura que el mapatge tecnològic avança i més nodes són assignats a cel·les d'una biblioteca. L'algoritme obté un guany d'un 5% en l'àrea global ocupada pel circuit i d'un 7% en la longitud de les connexions, comparat a les tècniques tradicionals [BRSW87].



En *synthetic* hem decidit d'ignorar l'àrea dedicada al connexionat. D'una banda, les noves tecnologies i eines de disseny tendeixen a fer que aquesta àrea sigui menyspreable respecte de l'àrea ocupada per les portes [EUGM95]. D'altra banda, els resultats obtinguts per eines de síntesi que incorporen estimacions precises de l'àrea de connexionat [PB91] no milloren significativament els resultats obtinguts quan aquesta àrea és ignorada [BRSW87] i en canvi el temps de càlcul es dispara.

En conclusió, el model d'àrea utilitzat en *synthetic* es basarà en el número de transistors requerits per implementar cada porta de la Xarxa Booleana que representa el circuit. És important destacar que el número de transistors requerits per implementar els nodes d'una Xarxa Booleana pot ser aproximat en base a la dimensió dels BDDs que representen les funcions en els nodes de la xarxa (veure *Del BDD a la implementació* Òptima, pàgina 50). És evident que aquesta aproximació pot ser usada durant la reestructuració de la Xarxa Booleana per tal de guiar molt més eficientment el procés d'optimització d'aquesta, obtenint així una millor Xarxa Booleana inicial pel mapatge tecnològic.

## Model pel retard

El retard màxim d'una implementació d'un circuit no pot ser calculat directament usant una equació com la (6-1), ja que el retard no és un valor acumulatiu sobre les portes i connexions del circuit. En concret, el retard màxim d'una implementació vindrà establert pel màxim retard a les sortides primàries. A més, un dels problemes crítics a l'hora de calcular el màxim retard a la sortida d'una porta és que aquest no depèn només de l'estructura interna de la porta, sinó també del temps d'arribada de les entrades de la porta. El temps d'arribada de les entrades dependrà del retard que suposen les portes que es troben en el ventall transitiu d'entrada de la porta que estem considerant. I aquest, al seu torn, depèn de la combinació de valors a les entrades primàries del circuit, que fixen un camí determinat.

Habitualment, per tal de reduir la complexitat del càlcul, s'elimina la dependència dels valors de les entrades bo i considerant el pitjor temps d'arribada possible. Aquesta simplificació accelera en gran mesura els càlculs, però pot fer que el retard màxim avaluat per un circuit es correspongui a un camí pel qual no existeixi cap combinació d'entrades primàries que l'activi. Aquesta mena de camins es coneixen amb el nom de *camins falsos*. Diverses tècniques han estat proposades per descartar els camins falsos que puguin aparèixer en un circuit [SBS94], algunes d'elles exactes. Aquest tipus de tècniques, però, són computacionalment costoses i no solen ser aplicades en la síntesi lògica.

Quan s'optimitza la Xarxa Booleana per reduir el retard màxim en el circuit, la síntesi tendeix a produir un circuit balancejat en temps, on tots els camins que van de les entrades a les sortides primàries tenen un retard aproximadament igual. Per tant, en el cas que el retard màxim del circuit vingui donat per un camí fals és molt probable que existeixi un camí real amb un retard molt similar, la qual cosa fa innecessària l'aplicació de tècniques de detecció i eliminació de camins falsos.

El retard màxim d'una Xarxa Booleana  $G=(V, E)$  amb un conjunt d'entrades primàries  $X \subset V$  i de sortides primàries  $Z \subset V$  pot ser calculat com:

$$t_{\max} = \max(t_{\max}^v), \forall v \in Z \quad (6-2)$$

on  $t_{\max}^v$  és el màxim retard a la sortida del node (o de la porta)  $v$ . Aquest, al seu torn, pot ser calculat com:

$$t_{\max}^v = \max(t_{\max}^w) + t^v, \forall w \in V_e^v \quad (6-3)$$

on  $t^v$  representa el temps de propagació de la porta associada al node  $v$ , i  $t^w, \forall w \in X$  estarà fixada per condicions externes a l'entorn del circuit.

Per poder calcular el retard màxim d'un circuit, usant les equacions (6-2) i (6-3), només ens queda per definir el model pel temps de propagació en una porta,  $t^v$ . Durant un mapatge tecnològic sobre una biblioteca de cel·les estàndard, aquest paràmetre formarà part de les dades disponibles en les fulles de dades de les cel·les i podrà ser usat. Però en el cas de considerar un mapatge orientat a un generador de mòduls, aquest valor no quedarà fixat fins que s'hagi fet el dimensionament dels transistors de les portes. Així, caldrà definir un model de propagació simplificat per les portes del circuit. Aquest model ha de basar-se únicament en la informació disponible de l'estructura de les portes del circuit. A continuació es descriuran breument els dos models que han estat implementats en *synthetic*: l'unitari i el RC simplificat.

### Model unitari

El model unitari consisteix en assignar un retard unitari a cada porta del circuit:

$$t^v = 1, \forall v \in V - X$$

El retard màxim del circuit, quan s'utilitza el model unitari per les portes del mateix, ens mesura el número màxim de nivells de lògica que han de travessar els senyals d'entrada fins a arribar a les sortides primàries del circuit.

### Model RC simplificat

El model RC d'una porta CMOS es basa en identificar aquesta a un interruptor controlat per la funció de la porta. La porta ha de carregar (descarregar) la capacitat visible a la sortida a través d'una resistència equivalent a la xarxa de transistors P (N) que es troben en el camí des de l'alimentació (terra) a la sortida i que dependrà bàsicament del número de transistors en sèrie. Es distingirà entre el temps de propagació quan la sortida passa de nivell baix a nivell alt,  $t_{lh}^v$ , i quan passa de nivell alt a nivell baix,  $t_{hl}^v$ , ja que les xarxes de transistors involucrats en el càlcul són diferents (el pla P en el primer cas i el pla N en el segon). Per al desenvolupament

del model RC simplificat, es considerarà només  $t_{lh}^v$ . El càlcul de  $t_{hl}^v$  seria equivalent. Així:

$$t_{hl}^v = t_{hl,fix}^v + \Delta t_{hl}^v C$$

on  $t_{hl,fix}^v$  és el temps de propagació intrínsec a la porta,  $\Delta t_{hl}^v$  l'increment del temps de propagació per unitat de capacitat i  $C$  és la capacitat externa a la porta.

La capacitat externa a la porta,  $C$ , depèn de la capacitat de connexionat i de la capacitat que suposen les portes dels transistors controlats pel senyal de sortida de  $v$ . Aquesta capacitat no serà coneguda fins que s'hagin efectuat les etapes de posicionament, dimensionant i connexionat de transistors. De totes maneres, l'existència d'una etapa de dimensionament de transistors posterior al mapatge tecnològic, lluny de ser un inconvenient, pot esdevenir un avantatge. Efectivament, el dimensionament dels transistors de la porta  $v$  tendirà a compensar d'alguna manera les possibles variacions de  $C$  variant també  $\Delta t_{hl}^v$  i  $t_{hl,fix}^v$ . Malgrat això, existeix un límit a partir del qual l'increment en la dimensió dels transistors de la porta  $v$  no aconseguen de reduir el temps de propagació  $t_{hl}^v$  associat a la porta [RPRCT91]. Aquest límit està relacionat amb el número màxim de transistors de tipus P en sèrie, ja que el temps de propagació apareix multiplicat per la resistència que suposen els transistors en sèrie i un terme que depèn de resistències i capacitats d'implantació.

D'aquesta manera es pot utilitzar el següent model simplificat:

$$\begin{aligned} t_{lh}^v &= k_P \max(\text{transistors P en s'erie en } v) \\ t_{hl}^v &= k_N \max(\text{transistors N en s'erie en } v) \end{aligned} \quad (6-4)$$

on  $k_P$  i  $k_N$  són constants que depenen bàsicament de la diferent mobilitat dels portadors i tensions d'alimentació i llindar, en els transistors P i N.

En conclusió, el model de retard RC simplificat es basarà en el número màxim de transistors en sèrie requerits per tal d'implementar cada porta de la Xarxa Booleana que representa el circuit. És important destacar que el número màxim de transistors P (N) en sèrie requerits per implementar els nodes d'una Xarxa Booleana pot ser aproximat en base al número màxim d'arcs de tipus  $P_i$  ( $N_i$ ) que cal travessar en els BDDs que representen les funcions en els nodes de la xarxa per arribar des de l'arrel al terminal  $I$  ( $0$ ) (veure *Conclusions* sobre *Generació de Portes Complexes*, pàgina 55). És evident que aquesta aproximació pot ser usada durant la reestructuració de la Xarxa Booleana per tal de guiar molt més eficientment el procés d'optimització d'aquesta, obtenint així una millor Xarxa Booleana inicial pel mapatge tecnològic.

## Model per a la dissipació de potència

La potència dissipada en un circuit s'està convertint cada cop més en un paràmetre de disseny crític. La contínua disminució de l'amplada mínima de canal de les

tecnologies permet un increment tant de la freqüència de rellotge com de la densitat d'integració de dispositius. El resultat és una dissipació de potència cada cop més elevada. La Taula 6-1 mostra el consum d'alguns dels microprocessadors actuals, segons [Naj94]. És evident que l'únic camí viable per reduir el consum en un circuit integrat és considerar la dissipació de potència com un factor crític, al costat del retard màxim i l'àrea, durant totes les etapes del procés de disseny [DM95].

MICROPROCESSADOR	POTÈNCIA DISSIPADA
Power PC (Motorola)	8.5 W
Pentium (Intel)	16 W
Alpha (DEC)	30 W

**Taula 6-1** Potència dissipada per alguns dels microprocessadors actuals, segons [Naj94].

Abans d'entrar en més consideracions sobre la dissipació de potència, caldria clarificar la nomenclatura. Quan es parla de dissipació de potència, normalment es fa referència a la *mitjana de la dissipació de potència* en un circuit, que no ha de ser confosa amb la *màxima dissipació instantània de potència* que es refereix a un problema totalment diferent [CB88, CB90, KNH92].

La mitjana de la potència dissipada en un circuit es pot calcular com la suma de la mitjana de potència dissipada en cadascuna de les portes del circuit. La potència dissipada en una porta es pot dividir bàsicament en tres components. D'una banda hi ha la potència dissipada estàticament. En el cas d'una porta CMOS, aquesta es deu als corrents de fuga dels transistors i, típicament, no arriba al 3% de la potència total consumida en la porta [DHNTB95]. D'altra banda, hi hauria la potència consumida durant la commutació de la porta, degut a un curt-circuit temporal entre alimentació i terra. Aquest consum pot ser pràcticament reduït fins a zero evitant que nodes del circuit puguin quedar en un estat intermig, ni alt ni baix, i dissenyant les portes per tal d'evitar transicions lentes. Entre altres qüestions, això implica limitar el número màxim de transistors en sèrie en una porta per tal d'evitar un temps de transició excessiu a la sortida. Finalment, la major part de la potència dissipada en una porta CMOS es deu als corrents necessaris per carregar o descarregar la capacitat visible tan interna com a la sortida de la porta quan aquesta commuta [DHNTB95]. Per tant, la potència dissipada, en mitjana, per una porta  $v$  del circuit serà:

$$P_{promig}^v \cong \frac{1}{2} C^v V_{dd}^2 f N^v \quad (6-5)$$

on  $V_{dd}$  és la tensió d'alimentació;  $f$  és la freqüència d'operació del circuit, normalment fixada per les especificacions; la capacitat de sortida,  $C^v$ , es deu a capacitats internes de la porta, capacitats associades al connexionat i capacitats associades a les entrades de les portes en el ventall de sortida de  $v$ ; finalment,  $N^v$  és l'*activitat de commutació* de la porta, és a dir, el número de transicions efectuat, en mitjana, per cicle de rellotge.

Reduir la dissipació de potència a la vista de l'equació (6-5) implica reduir algun dels paràmetres que intervenen en l'equació. La freqüència d'operació  $f$  vindrà fixada per les especificacions. Tampoc es pot fer gran cosa per reduir  $C^v$  des de la síntesi lògica. Només seria possible limitar el ventall de sortida de les portes. Però això comportaria

alguna mena de duplicació de lògica i un increment d'àrea. I, el que és pitjor, la duplicació de lògica suposaria noves portes en el circuit i, per tant, més potència dissipada. Possiblement molta més de la que s'estalviaria limitant el ventall de sortida. Finalment, reduir la potència dissipada passa per reduir  $V_{dd}$  o  $N^v$ .

Observant la dependència quadràtica respecte de  $V_{dd}$ , és clar que un gran estalvi en la dissipació de potència pot venir donat per una reducció de la tensió d'alimentació. Efectivament, reduir l'alimentació de 5V a 3.3V redueix el consum en un 56.44%; si l'alimentació es redueix encara fins a 2V, la potència decreix en un 84%; finalment, una alimentació de 1.5V estalvia un 91% del consum respecte del mateix circuit alimentat a 5V. Naturalment, aquest important benefici no ve de franc. En [SPK95] s'analitzen els efectes de la disminució de la tensió d'alimentació. Si aquesta no s'acompanya d'una disminució de les tensions llindar dels transistors, es produeix una significativa pèrdua de velocitat d'operació. Per compensar aquests efectes, es proposen noves metodologies de disseny basades en la paral·lelització de mòduls (comptadors, memòries, etc...) [SPK95]. Cal recordar, però, que la paral·lelització implica un increment notable en àrea. Diferents tècniques han estat proposades en [Mus96], que cobreixen solucions arquitecturals i els processos involucrats en la síntesi a nivell de sistema i la síntesi lògica.

En el cas, d'altre banda habitual, on no sigui possible reduir la tensió d'alimentació (per exemple si ja s'ha escollit una tecnologia), l'única via de reduir la dissipació de potència passa per reduir l'activitat de commutació,  $N^v$ , de les portes del circuit. Per això caldria comptar amb estimacions precises de l'activitat de commutació de cada porta  $i$ , malauradament, el càlcul de  $N^v$  no és senzill i depèn de molts i diferents factors [DKW92, Naj94, Naj95, DP95], els més importants dels quals seran breument descrits a continuació:

- **Model de retard de les portes.** El model de retard de les portes no només afecta a l'exactitud amb que són calculats els retards en el circuit, sinó també a la dissipació de potència. En efecte, entre un 20% i un 70% del total de la potència total dissipada en un circuit es deu a polsos espuris a la sortida de les portes [SGDK92]. En un model de retard zero, totes les portes commuten alhora i, per tant, cap pols espuri no apareix en el circuit. En el model unitari, la sortida de la porta canvia al cap d'un cert temps d'haver-se produït un canvi en les entrades; tots els polsos espuris són propagats a través del circuit, produint-se una sobreestimació de la potència dissipada, ja que alguns d'aquests polsos podrien ser filtrats en un circuit real pel retard inercial<sup>1</sup> de les portes. El model general considera el retard inercial de les portes i, segons [EB95], els polsos espuris es redueixen en un 50%. Cal fer atenció, però, al fet que alguns circuits són especialment sensibles a variacions en els retards interns als circuits quan es considera un model de retard general pur. Això es deu al fet que petites variacions en el retard d'alguns camins interns poden fer canviar l'amplada dels polsos espuris fent que alguns no siguin filtrats quan abans ho eren. Per evitar aquest tipus de

---

<sup>1</sup> Retard inercial d'una porta és el mínim temps que un senyal d'entrada ha d'estar estable després d'una transició, per tal que pugui ser propagat a la sortida.

problema, cal considerar models generals que considerin l'efecte d'atenuació de polsos espuris, com el presentat en [SGDK92].

- **Correlació entre senyals.** Aquest problema està d'alguna manera relacionat amb el problema de la dependència dels vectors d'entrada. Al considerar els possibles vectors d'entrada al circuit cal tenir present que aquests poden estar correlacionats tant espacialment com temporalment. Parlarem de *correlació espacial* quant els valors de dues senyals estan relacionats d'alguna forma (p.e. si una senyal no pot agafar el nivell baix mentre una altre està al mateix temps a nivell alt). Parlarem de *correlació temporal* quan els valors d'un senyal en dos cicles de rellotge consecutius estan relacionats d'alguna manera. Aquest fenomen es deu sovint a realimentacions en els circuits seqüencials. Cal comentar, però, que les correlacions poden aparèixer també en nodes interns del circuit degut a un ventall transitiu de sortida que reconvergeix sobre un node determinat. Si aquestes correlacions no es tenen en compte, l'estimació de l'activitat de commutació en el circuit pot resultar invalidada, com també pot resultar-ho si no es consideren els efectes de diverses entrades d'una porta commutant al mateix temps [ChRP94, MBOI95].
- **Dependència dels vectors d'entrades.** L'activitat en els nodes del circuit depèn dels vectors d'entrada que són aplicats al circuit. Efectivament, les transicions que es produeixen a la sortida de les portes del circuit depenen dels valors de dos vectors d'entrada consecutius. L'estimació de la potència dissipada esdevé complicada perquè no depèn només de l'estructura del circuit, sinó també dels vectors d'entrada. Aquest problema ha estat atacat usant tècniques ben diverses. Les més simples es basen en la simulació explícita del circuit per un conjunt representatiu de vectors d'entrada. Inicialment es basaven en simulacions elèctriques [Kan86, YK89], però aquestes resultaven ser massa lentes, pel que s'han anat proposat diferents tipus de modelat pels transistors així com simulacions a nivell d'interruptor o lògiques [DSL88, Tja89, Kro91, BFOR93, DLRB93, AA94, ChRP94, LLS94, Sch95, DHNTB95]. Les correlacions internes entre senyals seran considerades pel simulador, però les externes s'han de tenir presents a l'hora de generar els vectors d'entrada.

Per evitar la generació dels vector d'entrada, s'han proposat tècniques estadístiques basades en el mètode Monte Carlo que generen vectors d'entrada aleatoris [BNYT93, XN94], però es requereixen molts vectors si es vol calcular la potència dissipada en cada porta.

La utilització de probabilitats per descriure tots els possibles vectors d'entrada sense haver-los de simular explícitament va ser introduïda en [Cir87]. Les probabilitats de senyal i transició de les entrades primàries del circuit han de ser especificades. Després, aquestes probabilitats seran propagades en el circuit usant teoria de probabilitats per tal d'establir les probabilitats en cada node del circuit i estimar a partir d'elles l'activitat de commutació. Aquest tipus de tècniques són sensibles a la correlació entre senyals. Correlacions espacio-temporals han estat considerades en [MMP94] pel model de retard zero i ampliades en [MMP95]. Un altre

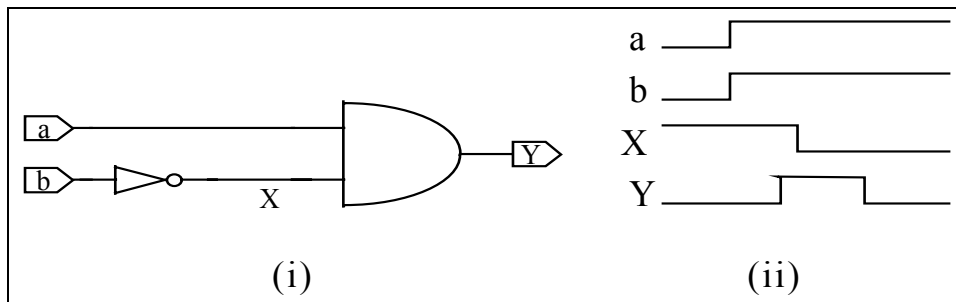
treball interessant va ser presentat en [NBYH88] i extès en [NBYH90], on es proposa modelar el comportament de les entrades per ones de probabilitat que són propagades en el circuit com en una simulació per events. Però segurament el treball més conegut és el presentat en [Naj93] on s'introdueix el concepte de densitat de transició (que és e la mitjana de transicions del senyal, per segon). L'algoritme utilitza la diferència Booleana per propagar les probabilitats de les entrades a la resta del circuit.

Finalment, existeixen tècniques exactes per un determinat model de retard i unes certes estadístiques d'entrada. De fet, utilitzen una enumeració exhaustiva implícita dels vectors d'entrada, per la qual cosa en el pitjor cas presenten complexitat exponencial i la seva aplicació queda reduïda a circuits mitjans i petits. Les tècniques exactes pel model de retard zero utilitzen BDD i posen les funcions implementades en cada node de la Xarxa Booleana en funció de les entrades primàries [Cha89]. Recentment, s'ha proposat una tècnica exacta pel model de retard zero que té en compte les possibles correlacions temporals entre senyals duplicant el número de variables d'entrada [SS95]. Per acabar, comentar que ha estat proposada una tècnica exacta pel model general de retard que es basa en la simulació simbòlica del circuit [GDKW92].

Les tècniques d'estimació de l'activitat de commutació en els nodes de la Xarxa Booleana poden ser aplicades per guiar l'optimització d'aquesta en una direcció tal que la dissipació de potència es redueixi. Efectivament, les subfuncions extretes en la Xarxa Booleana poden ser seleccionades en base a una baixa activitat de commutació [PR93, IP95]. En [TPD93] aquesta tècnica ha estat aplicada, considerant un model de retard zero i entrades no correlacionades, obtenint una disminució de la potència dissipada, després del mapatge tecnològic, de més d'un 20%. El mapatge tecnològic de circuits guiat per la dissipació de potència pot reduir aquesta en més d'un 10% [TAM93] per un model de retard zero i entrades no correlacionades. En les dues referències comentades, però, no s'ha considerat la dissipació de potència produïda per polsos espuris, que pot arribar a ser un percentatge elevat del total. Recentment, l'estudi de l'efecte dels polsos espuris ha estat analitzat en la literatura, proposant-se nous models [EB95, MFR95] i estudiant la capacitat de filtrar-los incrementant el retard inercial d'algunes portes [FM95]. De totes maneres, tal com s'apunta en les conclusions de [TAM93], els polsos espuris han de ser considerats explícitament per tal de balancejar el retard dels camins que arriben en cada porta i evitar la seva generació. Efectivament, la majoria de tècniques de mapatge tecnològic guiat per la dissipació de potència es basen en *amagar* els nodes amb més activitat de commutació en nodes interns de les portes, normalment menys capacitius. Donat que els models utilitzats no consideren la potència dissipada en els nodes interns, és difícil de saber l'efectivitat real d'aquesta tècnica, però el 10% anunciat en [TAM93] no és massa espectacular, sobretot considerant que la potència dissipada per polsos espuris pot ser entre el 20% i el 70% del total [SGDK92].

Al nostre entendre, la reducció de l'activitat de commutació ha de considerar-se en totes les etapes de la síntesi lògica, però els objectius en cada etapa poden ser diferents. Durant l'optimització de la Xarxa Booleana, l'estructura d'aquesta pot ser transformada i modificada en gran mesura en una direcció o altra, segons quina sigui la funció de cost que guii aquestes transformacions. Creiem que és en aquesta etapa

on s'ha de mirar d'optimitzar la Xarxa Booleana per tal d'aconseguir que les funcions que intervenen en el circuit tinguin la mínima activitat de commutació possible. D'altra banda, durant el mapatge tecnològic, la reestructuració de la Xarxa Booleana està molt més limitada. Els agrupaments disponibles per a cada node permeten una reducció molt limitada de l'activitat de commutació d'aquests, però en canvi poden molt bé balancejar adequadament els camins que arriben al node, reduint així la possible generació de polsos espuris.



**Figura 6-3** Els camins que arriben a la porta AND del circuit (i) tenen un retard diferent, la qual cosa provoca l'aparició d'un pols espuri en la sortida (iii).

En la Figura 6-3 es pot veure com es pot generar un pols espuri en un circuit degut a la diferència en el temps d'arribada de les entrades d'una porta. Tot i que les entrades del circuit estiguin balancejades, les entrades de la porta que genera la sortida no ho estan. És evident que un correcte balanceig dels camins presents en els circuits pot reduir en gran mesura la possibilitat de generació de polsos espuris. Per tant la funció de cost que guiarà el mapatge tecnològic en *synthetic* serà de la forma:

$$Cost(G^*) = \sum_{\forall v \in V^*} [\max(t^w) - \min(t^w), \forall w \in V_e^v] \quad (6-6)$$

on  $t^w$  representa el retard de l'entrada  $w$  del node  $v$  considerat.

La minimització de l'equació (6-6) implica reduir la generació de polsos espuris en el circuit, evitant així la seva propagació i reduint d'aquesta manera la dissipació de potència global.

## Mètodes de cobertura d'una Xarxa Booleana

El problema de cobertura d'una Xarxa Booleana ha estat formulat com un problema d'optimització d'una funció de cost que depèn de paràmetres tecnològics, de la metodologia de disseny i de la implementació escollida per a un circuit determinat. La minimització de la funció de cost associada a la implementació està subjecta als agrupaments disponibles en els nodes de la Xarxa Booleana i a la funcionalitat d'aquesta. Diverses tècniques poden ser aplicades per a resoldre el problema d'optimització plantejat. En aquest apartat se'n proposen tres. La primera d'elles correspon a la cobertura directa de la Xarxa Booleana, on els nodes són implementats usant el seu agrupament bàsic. Aquesta tècnica proporciona un mapatge directa de la



Xarxa Booleana i servirà de referència a l'hora de comparar els resultats per les altres tècniques. La segona tècnica estudiada correspon a una optimització local de la funció de cost mitjançant l'aplicació d'un algoritme àvid. En aquest cas, la funció de cost s'optimitza localment per a cada node de la Xarxa Booleana, sense considerar la minimització global de la funció de cost i portarà a un mínim local de la funció. Finalment, es proposarà una tècnica de cobertura òptima de la Xarxa Booleana basada en un estudi complet de les possibles implementacions. Les possibilitats d'implementació seran analitzades usant un algoritme que en cada node bifurca sobre totes les solucions possibles. Per reduir la complexitat de l'algoritme, en cada pas s'acota el cost mínim que suposarà arribar a una solució, de tal manera que les solucions més costoses que la millor solució trobada fins al moment seran descartades. D'aquesta manera moltes branques de l'arbre de l'espai de solucions poden ser tallades ben aviat sense que això suposi en cap moment la pèrdua de la solució òptima al problema.

### Cobertura directa

La cobertura directa de la Xarxa Booleana consisteix en implementar l'agrupament bàsic de cada node de la xarxa i en la seva forma directa o positiva. D'aquesta manera, tots els nodes de la xarxa original apareixen en la implementació, i cap d'ells pot ser implementat en la seva forma complementària. També poden aparèixer en la implementació altres nodes que corresponen als inversors necessaris per a les variables que són requerides en la seva fase positiva en algun node de la Xarxa Booleana. És a dir, són els inversors necessaris per a transformar totes les funcions que apareixen en els nodes de la Xarxa Booleana en funcions de fase negativa.

Formalment, la cobertura directa suposa una simplificació del problema general de cobertura d'una Xarxa Booleana (veure l'apartat de *Definició del problema*), ja que es compleixen les següents propietats:

- i)  $U=V$
- ii)  $V_{-}^{*} = \emptyset$

Cal comentar que la propietat (i) no garanteix la implementació de l'agrupament bàsic de cada node de la Xarxa Booleana, ja que un node amb un ventall de sortida de més d'un node podria ser col·lapsat en un agrupament complex d'un node  $v$  del seu ventall de sortida i aparèixer en la Xarxa Booleana degut a un altre node  $w \neq v$  del seu ventall de sortida. Per tant, cal afegir explícitament la condició que només l'agrupament bàsic de cada node pot ser implementat.

```

funció coberturaDirecta(G)
   $G^* = \emptyset$ 
  1) ordenamentTopològic(G)
     Per tot node de G fer
  2)   grup = agrupamentBàsic(node)
  3)    $v^* = \text{implementacióDirecta}(\text{grup})$ 
  4)    $G^* = G^* + v^*$ 
  retorna  $G^*$ 

```

**Algoritme 6-1** Algoritme per la cobertura directa d'una Xarxa Booleana.

En l'Algoritme 6-1 es pot veure la realització de l'algoritme de cobertura directa d'una Xarxa Booleana  $G=(V, E)$ . L'algoritme actua de la següent forma. En primer lloc, s'inicialitza la Xarxa Booleana,  $G^*$ , que ha de guardar la implementació del circuit representat per la Xarxa Booleana que es vol cobrir,  $G$ . A continuació (línia 1) es procedeix a realitzar un ordenament topològic dels nodes de  $G$ , des de les entrades a les sortides. Un cop fet això, es realitza la cobertura directa dels nodes de la Xarxa Booleana. Donada l'ordenació prèvia d'aquests, la cobertura es realitza des de les entrades primàries a les sortides primàries, seguint l'ordre topològic. De cada node, s'obté el seu agrupament bàsic (línia 2), que no és altre que la funció realitzada en el node. Aquest agrupament és implementat en la seva forma directa en una porta CMOS d'una sola etapa, aplicant els mètodes proposats en el capítol 4 d'aquesta tesi (línia 3). La implementació directa d'agrupaments que no siguin funcions de fase negativa implicarà necessàriament afegir inversors pels senyals d'entrada no fase negatius de la porta que implementa l'agrupament. En aquest aspecte, la cobertura en ordre topològic dels nodes de la Xarxa Booleana facilitarà aquesta operació, així com la incorporació de la nova porta en la implementació  $G^*$  (línia 4). Finalment, la funció retorna la implementació  $G^*$  resultant de la cobertura directa de  $G$ .

És evident que una cobertura directa de la Xarxa Booleana no té opció a minimitzar la funció de cost associada a la implementació, ja que només existeix una implementació possible al no poder escollir entre els diferents agrupaments de cada node. Precisament per això, la cobertura directa de la Xarxa Booleana ha estat implementada en *synthetic* per mesurar el nivell d'optimització respecte dels paràmetres de disseny de les Xarxes Booleanes que arriben al procés de mapatge tecnològic. Usant aquest valor com a referència, podrem comparar com diferents estratègies de cobertura de la Xarxa Booleana assoleixen compromisos diferents entre els paràmetres de disseny durant el mapatge tecnològic.

## Cobertura amb optimització local de la funció de cost

La cobertura de la Xarxa Booleana amb optimització local de la funció de cost consisteix en seleccionar la millor implementació possible per a cadascun dels nodes de la xarxa, basant-se únicament en la informació de cost local disponible per cada agrupament. Les implementacions que suposen els diferents agrupaments de cada node de la Xarxa Booleana són provats, tant pel que fa a la seva implementació directa com pel que fa a la complementària. S'escull la implementació que minimitzi el cost local al node. El cost local d'un agrupament  $v^*$  d'un node  $v$  es calcula com la

suma del cost d'implementar l'agrupament (en forma directa o complementària) sumat al cost de la millor implementació (respecte al respectiu cost local) dels senyals que es troben en el ventall d'entrada de l'agrupament  $v^*$ :

$$\text{Cost Local}(v^*) = \text{cost}(v^*) + \sum_{\forall w \in \mathcal{I}_{et}^{v^*}} \text{cost}(w^*) \quad (6-7)$$

on  $w^*$  representa la millor implementació d'una variable  $w$  del ventall d'entrades transitiu de  $v$ . La funció  $\text{cost}(v^*)$  haurà de tenir en consideració els possibles inversors necessaris per generar la fase negativa de totes les variables en el suport de  $v^*$ .

La minimització de l'equació (6-7) en tots i cadascun dels nodes de la Xarxa Booleana no garanteix la minimització global de la funció de cost proposada en l'equació (6-1), però permet assolir un mínim local amb un simple recorregut, en ordre topològic, de la Xarxa Booleana.

```

funció coberturaOptimitzacióLocal(G)
  implÒptima[ $\forall v \in G$ ] =  $\emptyset$ 
  1) ordenamentTopològic(G)
  1) generaAgrupaments(G)
  3) Per tot node de G fer
    millorImpl =  $\emptyset$ 
  4) Per tot agrupament del node fer
     $v+$  = implementacióDirecta(agrupament)
     $v-$  = implementacióComplementària(agrupament)
  5) Si  $\text{costLocal}(v+, \text{implÒptima}) < \text{costLocal}(\text{millorImpl}, \text{implÒptima})$  fer
    millorImpl =  $v+$ 
  5) Si  $\text{costLocal}(v-, \text{implÒptima}) < \text{costLocal}(\text{millorImpl}, \text{implÒptima})$  fer
    millorImpl =  $v-$ 
  fiPer
  6) implÒptima[node] = millorImpl
  fiPer
   $G^*$  =  $\emptyset$ 
  7) Per tot node de sortida primària de G fer
     $v^*$  = implÒptima[node]
  8) verificaEntradesTransitives( $v^*$ ,  $G^*$ , implÒptima)
  9)  $G^*$  =  $G^* + v^*$ 
  fiPer
  retorna  $G^*$ 

```

**Algoritme 6-2** Algoritme per la cobertura d'una Xarxa Booleana amb optimització local de la funció de cost.

L'Algoritme 6-2 mostra la realització, en *synthetic*, del procediment de cobertura d'una Xarxa Booleana  $G$  amb una optimització local de la funció de cost. Un algoritme d'aquestes característiques, que explora àvidament totes les solucions possibles en cada pas i en selecciona la millor basant-se en una funció de cost local, es coneix amb el nom d'algoritme àvid. L'Algoritme 6-2 comença inicialitzant una

taula que guardarà la millor implementació, respecte al cost local, de cada node de la Xarxa Booleana. A continuació (línia 1) es procedeix a l'ordenament topològic dels nodes de la xarxa. Donat que l'avaluació del cost local de cada node necessita del cost local dels nodes que es troben en el seu ventall d'entrades transitiu (veure l'equació (6-7)), la millor implementació per a aquests darrers ha d'haver estat calculada prèviament per l'algoritme. La generació de tots els possibles agrupaments pels nodes de la Xarxa Booleana es produeix en la línia 2, usant els algorismes descrits en el capítol 5 d'aquesta tesi. Tot seguit es calcula la millor implementació de cada node de la Xarxa Booleana, seguint l'ordre topològic, en la iteració iniciada en la línia 3. En cada node de la xarxa s'itera sobre el seu conjunt d'agrupaments per tal d'escollir-ne el de menys cost (iteració iniciada en la línia 4). Per això, es comprova la implementació positiva i complementària de cada agrupament. El cost d'aquestes implementacions s'avalua usant l'equació (6-7), la qual cosa requerirà les implementacions òptimes (respecte el cost local) de les variables que es troben en el ventall d'entrades transitiu de l'agrupament que s'està calculant. El processament dels nodes en ordre topològic garanteix que les implementacions localment òptimes d'aquestes variables estaran disponibles en la taula *implÒptima*, que es passa com a paràmetre juntament amb la implementació de l'agrupament a la funció que realitza el còmput del cost local. Si alguna de les implementacions de l'agrupament actual resultés ser menys costosa que la millor implementació trobada fins al moment pel node seria recordada com la millor trobada fins al moment. Quan la iteració sobre tots els agrupament d'un node acaba, es guarda en la taula *implÒptima* la millor implementació trobada pel node (línia 6). El procediment de selecció de la millor implementació respecte al cost local es repeteix per a cada node de la Xarxa Booleana. Finalment, cal construir el circuit que suposa les millors implementacions pels nodes de sortida, cosa que es fa en la iteració iniciada en la línia 7. De cada sortida de la Xarxa Booleana, s'obté la seva implementació òptima respecte al cost local i posteriorment es verifiquen i s'implementen les seves entrades transitives (línia 8). Per acabar s'afegeix la porta a la implementació (línia 9).

*funció* verificarEntradesTransitives( $v^*$ ,  $G^*$ , *implÒptima*)

Per tot  $w$  en  $V_e^{v^*}$  fer

$w^* = \text{implÒptima}[w]$

Si  $w^* \notin G^*$  fer verificarEntradesTransitives( $w^*$ ,  $g^*$ , *implÒptima*)

$G^* = G^* + w^*$

**Algoritme 6-3** Algoritme que verifica l'existència de les entrades transitives d'una porta  $v^*$  de la implementació  $G^*$ .

La implementació de les entrades transitives d'una porta  $v^*$  es pot fer recursivament usant l'Algoritme 6-3. De cada node  $w$  que figura en el ventall d'entrada de la porta  $v^*$ , se n'obté la seva implementació  $w^*$ . A continuació es verifiquen recursivament les entrades transitives de  $w^*$  i s'afegeix aquesta porta a la implementació  $G^*$ . La implementació recursiva de l'algoritme assegura que totes les entrades d'una porta estaran disponibles en la implementació  $G^*$  abans d'afegir la porta.

És important comentar el tractament dels circuits amb més d'una sortida. Un cop s'ha cobert un node de sortida, la seva implementació queda fixada (observi's la iteració en la línia 7 de l'Algoritme 6-2 i també l'Algoritme 6-3). És a dir, part de la implementació  $G^*$  ja

queda definida al cobrir un node de sortida, amb independència de com es cobreixin la resta de nodes que quedin per processar. Aquest fet ha de ser tingut en consideració, ja que a l'hora de cobrir la resta de nodes, les portes que ja estan fixades i són necessàries per implementar la primera sortida poden ser utilitzades amb un cost zero (simplement afegint una nova connexió). Això implica que l'equació (6-7) no ha de ser aplicada tal qual, sinó que ha de tenir en compte que algunes portes poden ser usades amb cost zero. D'aquesta manera s'afavoreix el reaprofitament de les portes ja presents en la coberta parcial de la Xarxa Booleana:

$$Cost Local(v^*) = cost(v^*) + \sum_{\forall w \in V_{et}^{v^*}, w^* \notin Z_t^*} cost(w^*) + \sum_{\forall u^* \in Z_t^*} cost(u^*) \quad (6-8)$$

on el conjunt  $Z_t^*$  està format per totes aquelles portes necessàries per a la implementació dels nodes de sortida que ja han estat coberts en la Xarxa Booleana.

El segon sumatori en l'equació (6-8) és el cost fix degut a les sortides ja cobertes. Aquest cost serà sempre el mateix, independentment de l'agrupament  $v^*$  que s'estigui considerant. Ara bé, a diferència del que passava en l'equació (6-7), el segon sumatori no inclou totes les portes que generen els senyals transitius d'entrada de  $v^*$ , sinó només aquelles portes que no figuren a  $Z_t^*$ . Per tant, la minimització del cost local dels nodes  $v$  que es cobreixin a partir del moment que alguna de les sortides de la Xarxa Booleana ha estat coberta passarà per un reaprofitament natural de les portes ja existents.

L'optimització local de la funció de cost, tal com s'ha descrit en aquest apartat permet optimitzacions en una direcció prioritària. És a dir, si considerem els tres paràmetres crítics a minimitzar (àrea, retard màxim i sensibilitat a polsos espuris), la cobertura amb optimització local optimitzarà un d'aquests paràmetres com a prioritari; en cas d'igualtat s'optimitzarà un dels restants i, només en el cas que la igualtat es mantingui, s'optimitzarà el tercer. No pot ser d'altra manera, ja que només es conserva una implementació per cada node de la Xarxa Booleana. Si el que es vol és assolir compromisos entre els diferents paràmetres de disseny, caldrà utilitzar una estratègia de cobertura capaç de recordar diferents solucions i permetre a l'usuari l'elecció d'una d'elles. L'algoritme de cobertura òptima que es proposa a continuació reuniria aquestes característiques.

## Cobertura Òptima

L'algoritme de cobertura òptima de la Xarxa Booleana ha d'explorar tot l'espai de solucions disponible. Totes les possibles combinacions de selecció d'agrupaments en combinació amb implementacions directes o complementàries d'aquests, han de ser provades, implícita o explícitament. Quan l'espai de solucions és gran, com és el cas del problema de la cobertura òptima d'una Xarxa Booleana, i només interessa un grup petit de solucions, es poden aplicar tècniques encaminades a reduir la recerca de solucions en zones de l'espai on es sap que no es pot arribar a cap solució interessant. Per això, cal definir *funcions d'acotació* que, a partir d'un punt en l'espai de solucions, puguin calcular un límit inferior al cost necessari per arribar a una solució. D'aquesta manera, si el cost actual calculat efectivament, sumat al mínim cost

necessari per arribar a una solució, predit per la funció d'acotació, ja excedeix el cost d'una altre solució prèviament trobada, es pot descartar el subespai de solucions a partir del punt actual.

Aquesta tècnica permet reduir l'espai de solucions efectivament explorat per l'algoritme sense perdre cap solució, si la funció d'acotació proporciona un límit inferior correcte. Els algoritmes que apliquen aquestes tècniques s'anomenen algoritmes de bifurcació amb acotació. Es diu que en cada pas de l'exploració de l'espai de solucions, l'algoritme obre una sèrie de branques que corresponen a les possibles alternatives disponibles en aquell punt. L'algoritme fa una estimació del cost que suposaria arribar a una solució per cadascuna de les branques, talla les que excedeixen el cost de la millor solució trobada fins al moment i bifurca recursivament per a cadascuna de les que queden.

La funció d'acotació serà, per tant, decisiva per l'eficàcia de l'algoritme de bifurcació amb acotació. És indispensable que la funció d'acotació sigui exacta, en el sentit que mai no pot predir en una branca un cost superior al cost real que suposaria arribar a la solució en aquella branca, ja que això podria suposar la pèrdua de solucions correctes. D'altra banda, si la predicció és dolenta, és a dir, es prediuen valors excessivament baixos, molt poques branques són eliminades i l'algoritme perd molt de temps explorant branques que no acaben portant a cap solució. Per tant, és molt important definir una bona funció d'acotació. En [CM95] es mostra com l'aplicació d'una funció d'acotació computacionalment costosa, però que acota eficientment el cost real, acaba produint algoritmes molt més eficients, tant pel que fa al temps de càlcul com als requisits de memòria.

*funció cobertura*Òptima( $G$ )

1) generaAgrupaments( $G$ )

1)  $nodesPerCobrir = \emptyset$

$portesCobertes = \emptyset$

$millorCost = \infty$

$coberta = \emptyset$

3) Per tot node de sortida primària de  $G$  fer

$nodesPerCobrir = nodesPerCobrir \cup node$

4)  $cobrirNodes(nodesPerCobrir, portesCobertes, millorCost, coberta)$

1) retorna  $coberta$

**Algoritme 6-4** Cobertura òptima d'una Xarxa Booleana  $G$ .

L'algoritme (6-4) mostra la implementació d'un cobertura òptima d'una Xarxa Booleana  $G$ . L'algoritme genera en primer lloc tots els agrupaments possibles pels nodes de la Xarxa Booleana (línia 1). Tot seguit s'inicialitzen els paràmetres de la coberta actual i també la millor cobertura trobada fins al moment, que és nul·la (línia 2). A continuació, estableix el conjunt de nodes que cal cobrir com el conjunt de sortides primàries de la Xarxa Booleana (iteració en la línia 3). Aquestes s'han de cobrir forçosament, mentre que la resta de nodes de la Xarxa Booleana només seran coberts si són requerits. Els nodes a cobrir són coberts amb una crida a la funció *cobrirNodes* (línia 4), que retorna la millor cobertura trobada i el cost associat. Per acabar, es retorna la millor cobertura trobada (línia 5).

```

funció cobrirNodes(nodesPerCobrir, portesCobertes, millorCost, coberta)
1) costActual = costReal(portesCobertes)
   Si nodesPerCobrir =  $\emptyset$  fer
2)   Si costActual < millorCost fer
       millorCost = costActual
       coberta = portesCobertes
   fiSi
   sinó
3)   node = escollirNode(nodesPerCobrir)
4)   Per tot agrupament del node fer
5)     nodesAcobrir = nodesPerCobrir - node
5)     Per tot  $v \in V_e^{\text{agrupament}}$  fer nodesAcobrir = nodesAcobrir  $\cup$  v
       v+ = implementacióDirecta(agrupament)
       v- = implementacióComplementària(agrupament)
6)     Si (costActual + costMínim(v+, nodesAcobrir)) < millorCost fer
       cobrirNodes(nodesAcobrir, portesCobertes  $\cup$  v+, millorCost, coberta)
6)     Si (costActual + costMínim(v-, nodesAcobrir)) < millorCost fer
       cobrirNodes(nodesAcobrir, portesCobertes  $\cup$  v-, millorCost, coberta)
   fiPer
7)   acotaCost(node)
   fiSi

```

**Algoritme 6-5** Cobertura òptima d'un conjunt de nodes d'una Xarxa Booleana  $G$ .

En realitat l'Algoritme 6-4 només es dedica a preparar els paràmetres necessaris per cridar a la funció *cobrirNodes*, que serà la que acabarà fent tota la feina. La realització d'aquesta funció es pot veure en l'Algoritme 6-5. L'algoritme procedeix recursivament, cobrint un node en cada pas de la recursivitat. Per a cada node a cobrir s'exploren tots els seus possibles agrupaments, tant per la implementació directa com per la complementària, i si l'estimació de costos no excedeix el cost de la millor coberta trobada fins al moment, l'algoritme continua cobrint, recursivament, els nodes pendents, afegint-hi els nodes del ventall d'entrada de l'agrupament considerat.

L'Algoritme 6-5 de cobertura òptima de nodes està implementat recursivament, usant una estratègia de bifurcació i acotació. En primer lloc, l'algoritme calcula el cost real de les portes implementades fins al moment en la coberta parcial que s'està analitzant (línia 1). Si ja no queden més nodes per cobrir, es comprova si el cost de la coberta actual està per sota del cost de la millor coberta trobada fins al moment (línia 2). En el cas que sigui així, es guarda la coberta actual i el seu cost com la millor coberta i cost trobats fins al moment, i es surt de la recursivitat. Si, pel contrari, quedaven nodes per cobrir, se n'escull un, sobre el qual es bifurcarà en aquest pas de l'algoritme (línia 3).

L'ordre en el qual es processen els nodes pot afectar a les possibilitats d'acotació del cost de la branca. En general, els nodes s'haurien de processar en un ordre topològic invers, és a dir, primer els nodes més allunyats de les entrades, per tal de disposar d'una acotació respecte de les entrades primàries. Un cop escollit el node sobre el qual es bifurcarà, es procedeix a estudiar totes i cadascuna de les possibles

implementacions pel node (iteració iniciada en la línia 4). Cada agrupament del node es considerat, tant en la seva forma positiva com en la complementària. Abans, però, es calcula el nou conjunt de nodes a cobrir, que es correspondrà als nodes que quedaven pendents, units als nodes que es troben en el ventall d'entrada del node escollit per a la bifurcació en el pas actual de l'algoritme (línies etiquetades amb 5). Tot seguit, es calculen les dues implementacions possibles per l'agrupament. Immediatament, es fa una acotació del cost que representa arribar a la solució, bifurcant per cadascuna de les dues implementacions de l'agrupament. Aquesta acotació pot fer-se en base al mínim cost fix d'implementació de cada node dels que queden per cobrir. Si aquest cost, sumat al cost actual real, és inferior al cost de la millor cobertura trobada fins al moment (línies 6), la funció procedeix cridant-se recursivament amb els nous paràmetres calculats. Quan totes les bifurcacions sobre el node escollit acaben, el mínim cost fix d'implementació del node pot ser acotat de nou (línia 7) en base a les noves dades obtingudes per als nodes en el seu ventall transitiu d'entrada. Aquesta nova acotació serà usada en nivells superiors de la recursivitat, possibilitant així un descartament més ràpid d'algunes branques que no porten a una solució millor.

L'algoritme de cobertura òptima de la Xarxa Booleana, tal com s'ha proposat, arribaria a l'òptim d'una funció de cost on els paràmetres de disseny (àrea, retard i sensibilitat a polsos espuris) estan ordenats prioritàriament segons la funció de cost, tal com passava en la cobertura amb optimització local. Si, pel contrari, el que es vol es obtenir compromisos entre els paràmetres de disseny o, fins i tot, permetre a l'usuari l'elecció d'una cobertura entre un conjunt de cobertures possibles, es pot modificar lleugerament l'algoritme. Bàsicament, les modificacions serien dues. La primera és que caldria recordar un conjunt de cobertes solució, enlloc d'una sola. La segona, fa referència a la manera en com es descarten les branques que no porten a cap solució (línies 7 de l'Algoritme 6-5). Ja no es pot fer simplement per la comparació amb un sol valor de cost, sinó que l'usuari hauria de definir uns límits per a cadascun dels paràmetres de disseny per als quals seria acceptable una solució. És important destacar el fet que, en el cas que aquests límits fossin excessivament permissius, es podria generar un conjunt considerable de cobertes solució, moltes d'elles amb compromisos similars per als objectius de disseny. Aquesta situació pot ser resolta habilitant mecanismes que permetin descartar cobertes "massa properes" a altres cobertes solució, o bé establint un objectiu de disseny a optimitzar, mentre la resta s'han de mantenir per sota d'uns certs límits.

## Resultats

En aquest apartat, els resultats obtinguts pels mètodes de cobertura de la Xarxa Booleana implementats en *synthetic* es compararan entre ells i també amb els obtinguts usant SIS [SIS92]. SIS és l'evolució de MISII [BRSW87] i incorpora també síntesi lògica de circuits seqüencials, tot i que nosaltres només hem considerat la part combinacional. SIS, com MISII és un paquet de síntesi lògica que cobreix les etapes d'optimització de la Xarxa Booleana i posterior mapatge tecnològic.



Per tal de fer comparacions únicament pel que fa a l'etapa de mapatge tecnològic, s'ha procedit a optimitzar, usant SIS, les Xarxes Booleanes corresponents als circuits de prova. Els circuits així optimitzats han estat subministrats tant a *synthetic* com a SIS per fer el posterior mapatge tecnològic. El nostre principal interès és demostrar l'eficàcia d'una metodologia orientada a generadors de mòduls en comparació amb la metodologia clàssica de mapatge sobre biblioteques de cel·les. Per això, el mapatge tecnològic en SIS s'ha realitzat considerant una biblioteca de cel·les, concretament la *lib2.genlib*, de dimensions mitjanes, i que incorpora cel·les CMOS complexes amb un número d'entrades relativament gran. El cost associat a les cel·les de la biblioteca ha estat modificat per tal de reflectir les funcions de cost proposades en aquest capítol, de manera que tant SIS com *synthetic* manipulen les mateixes funcions de cost. El model considerat pel retard és el model RC simplificat. Les modificacions realitzades al fitxer original, distribuït juntament amb SIS, poden veure's en el llistat corresponent a la descripció de la biblioteca, en l'Apèndix B.

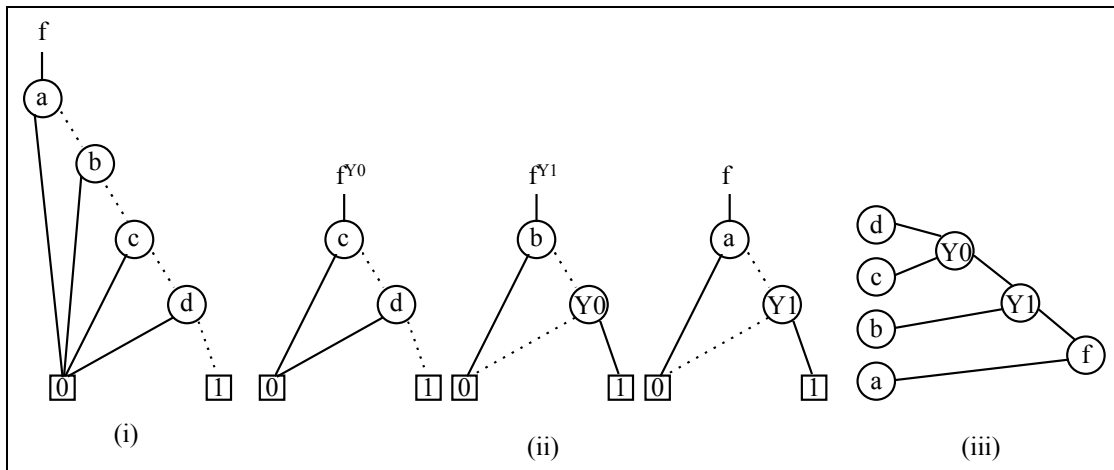
Pel que fa a *synthetic*, la cobertura de la Xarxa Booleana s'ha realitzat amb una configuració màxima de quatre transistors en sèrie, que és el màxim que apareix en la biblioteca de cel·les subministrada a SIS. La generació d'agrupaments pels nodes de la Xarxa Booleana s'ha fet sense considerar cap heurística d'eliminació d'agrupaments per tal de poder comparar les dues metodologies treballant sobre el seus respectius espais de solucions complets.

Els resultats que es presentaran a continuació corresponen a l'optimització prioritària de l'àrea i el temps, respectivament, sent l'altre paràmetre optimitzat amb prioritat secundària i sense considerar l'optimització de la dissipació de potència, degut a que SIS no incorpora mesures d'aquest paràmetre. La primera fila de la taula indica el mètode de mapatge tecnològic considerat: mapatge directe en *synthetic*, mapatge sobre biblioteca de cel·les optimitzant l'àrea en SIS, o mapatge amb optimització local d'àrea+retard màxim (o retard màxim+àrea) en *synthetic*. Per cada mètode es recull, en tres subcolumnes diferents, l'àrea final del circuit en número de transistors així com el retard màxim de les sortides del circuit, tant pel que fa al temps de pujada com pel que fa al temps de baixada. Les tres darreres files de la taula resumeixen els resultats. La fila etiquetada amb *Total* indica el total acumulat per cada mètode de mapatge, tant pel que fa al número de transistors com pel que fa al retard màxim en cada circuit (del temps de pujada i baixada). La fila etiquetada com *X/Directa* compara els resultats del mapatge amb SIS i amb *synthetic* usant optimització local, indicant l'increment o disminució del cost en comparació amb el mapatge directe de *synthetic*. Finalment, la darrera fila, etiquetada amb *X/SIS* compara els resultats del mapatge amb *synthetic* usant optimització local, indicant l'increment o disminució del cost en comparació amb el mapatge usant SIS.

Mapatge Circuit	Synthetic - Mapatge Directa			SIS - Area			Synthetic - Opt. Local Àrea		
	Àrea	Pujada	Baixada	Àrea	Pujada	Baixada	Àrea	Pujada	Baixada
cm82a	64	15,5	12,5	64	19	14	68	12	9
cm151a	78	17	18	84	19,5	20,5	66	15,5	14
parity	180	24	20	180	24	20	180	24	20
cmb	134	31	22,5	150	15	21	164	40,5	35
cm163a	140	35	33,5	118	36,5	39	134	16,5	16,5
cm162a	124	24	21,5	120	20,5	23,5	130	17,5	20,5
cm150a	118	17,5	16	132	19,5	18	110	16,5	21,5
cm85a	108	21	17	140	26,5	18,5	116	22,5	25
cu	144	30,5	23	158	28,5	19	176	23	18,5
pm1	110	20,5	21,5	126	23	24	130	17	10,5
pcler8	232	35	28	268	29,5	32,5	246	20	19,5
cc	172	24	13,5	182	18,5	15,5	182	13	9,5
Totals	1604	297		1722	296,5		1702	248,5	
X/Directa				7,36%	-0,17%		5,76%	-19,52%	
X/SIS							-1,16%	-16,19%	

**Taula 6-2** Resultats obtinguts per SIS i synthetic a l'optimitzar per àrea+retard.

La Taula 6-2 mostra els resultats obtinguts quan el criteri d'optimització prioritari és l'àrea, amb el retard com a criteri secundari. Els resultats mostren que la cobertura directa és la més eficient. Això indica que l'optimització de la Xarxa Booleana realitzada per SIS com a pas previ al mapatge tecnològic és efectiva, ja que la millor implementació en quant a àrea, correspon a implementar directament les funcions en cada node (l'agrupament bàsic del node). Cal recordar que l'optimització de la Xarxa Booleana té com a objectiu prioritari minimitzar l'àrea, cosa que aconseguix gràcies al fet de compartir subfuncions de la xarxa. Malgrat això, aquests resultats han de ser matisats. La cobertura directa no té en compte el límit en el número de transistors en sèrie imposat per la tecnologia i el dissenyador. Si aquest límit fos pres en consideració durant la cobertura directa, alguns dels circuits de prova no podrien ser coberts (cm150a, cmb, cm85a, cu, pm1, pcler8), ja que el seu agrupament bàsic resultaria ser no implementable. Tant SIS com *synthetic*, usant la cobertura amb optimització local, han de tenir presents aquests límits, per la qual cosa hauran de descompondre els nodes l'agrupament bàsic dels quals resulti ser no implementable. Això pot explicar el lleuger increment en l'àrea ocupada respecte de la cobertura directa, que és de poc més del 5% per *synthetic* amb optimització local i de poc més del 7% pel cas de SIS.



**Figura 6-4** Descomposició estructural del BDD en (i) amb limitació del suport a dues variables. En (ii) es pot veure el resultat, que suposa la introducció de dues variables intermèdies. L'ordenació de les variables en el BDD induïx una ordenació de les noves variables introduïdes en la Xarxa Booleana (iii).

En els resultats presentats en la Taula 6-2 cal destacar que el mapatge tecnològic minimitzant l'àrea de SIS no assegura una disminució del retard màxim en el circuit i presenta casos on aquest disminueix i altres on augmenta. En el còmput global, el resultat resulta aproximadament el mateix que pel mapatge directe de *synthetic*. Aquest aspecte és ben diferent en el cas de considerar el mapatge tecnològic amb optimització local de *synthetic*. Aquest, presenta una reducció del retard màxim en tots els circuits (excepte cmb, cm150a i cm85a), aconseguint disminuir-lo globalment en gairebé un 20% respecte de la cobertura directa, cosa que mostra que l'optimització respecte del segon criteri és efectiva. L'explicació de perquè el retard màxim empitjora en alguns circuits, es troba en el fet que aquests circuits necessiten descompondre algun dels seus nodes abans de procedir al mapatge tecnològic. Actualment, en *synthetic*, aquesta descomposició es fa estructuralment, és a dir, directament a partir del BDD que representa la funció implementada en el node, segons el mètode descrit en [Jac93]. Aquesta tècnica s'aprofita de l'estructura intrínsecament multi-nivell del BDD per descompondre la funció, de tal manera que, si és possible, es comparteixen en la Xarxa Booleana les mateixes subfuncions que ja eren compartides en BDD. D'aquesta manera s'indueix en la xarxa resultant una ordenació parcial en les variables del suport del BDD que s'ha descompost. En la Figura 6-4 es mostra el resultat de la descomposició estructural d'una NAND de quatre entrades entrades, suposant que la limitació de número de transistors en sèrie és dos. És evident que el procés d'agrupament de *synthetic* mai no podrà reduir la profunditat del camí entre  $f$  i les que abans eren les seves entrades, ja que les funcions dels nodes intermedis creats estan en el límit tecnològic. Però una descomposició balancejada de  $f$  hauria reduït la profunditat del camí (per exemple fent  $Y0 = \bar{c}\bar{d}$ ,  $Y1 = \bar{a}\bar{b}$ ,  $f = Y1Y0$ ). La utilització d'una descomposició balancejada en temps, aplicada enlloc de la descomposició estructural que s'utilitza actualment, permetrà reduir el retard màxim que obté *synthetic* usant optimització local en els circuits que contenen nodes amb agrupament bàsic no implementables.

Per acabar la comparació dels resultats obtinguts en el mapatge tecnològic optimitzant l'àrea, cal afegir que, els guanys en àrea de *synthetic* usant optimització local respecte de SIS són marginals, poc més d'un 1%. En canvi, el guany en temps és considerable, de més del 16%. Aquests resultats indiquen que la major flexibilitat del mapatge tecnològic orientat a generadors de mòduls permet, sinó reduir l'àrea del circuit final, sí reduir la profunditat global de la Xarxa Booleana. És de preveure que aquesta més gran flexibilitat facilitarà l'acompliment de compromisos agressius entre els diferents paràmetres de disseny.

Mapatge Circuit	Synthetic - Mapatge Directa			SIS - Area			Synthetic - Opt.Local Retard		
	Àrea	Pujada	Baixada	Àrea	Pujada	Baixada	Àrea	Pujada	Baixada
cm82a	64	15,5	12,5	84	18	17,5	70	12	9
cm151a	78	17	18	86	18,5	16	84	14	12,5
parity	180	24	20	184	22,5	23,5	180	24	20
cmb	134	31	22,5	180	14,5	17,5	190	27,5	27,5
cm163a	140	35	33,5	150	38,5	39	138	17	17,5
cm162a	124	24	21,5	148	21	18,5	128	20	20,5
cm150a	118	17,5	16	156	19,5	18	136	15,5	15,5
cm85a	108	21	17	148	21	18,5	118	21	23
cu	144	30,5	23	202	26	18,5	222	20,5	20,5
pm1	110	20,5	21,5	158	23	19,5	142	18,5	15,5
pcler8	232	35	28	306	28	26	248	20	19,5
cc	172	24	13,5	182	15	18,5	198	12	9,5
Totals	1604	297		1984	273,5		1854	225	
X/Directa				23,69%	-7,91%		15,59%	-24,24%	
X/SIS							-6,55%	-17,73%	

**Taula 6-3** Resultats obtinguts per SIS i *synthetic* a l'optimitzar per retard+àrea.

La Taula 6-3 mostra els resultats obtinguts quan el criteri d'optimització prioritari és el retard, amb l'àrea com a criteri secundari. Els resultats mostren que la cobertura directa ja no és la més eficient. Aquest fet es deu a que la Xarxa Booleana s'ha optimitzat prèviament per àrea. Comparem els resultats obtinguts per SIS i per *synthetic*, usant l'optimització local de la funció de cost, respecte del mapatge directe de *synthetic*. SIS aconsegueix reduir el retard en gairebé un 8% de mitjana, a costa d'un increment en àrea de més d'un 23%. Aquest fet indica que la reducció del retard sovint es fa a costa de duplicació de lògica en els circuits. En el cas del mapatge amb optimització local de *synthetic*, la reducció del retard és de més d'un 24%, amb un cost en àrea de prop d'un 16%. La comparació amb SIS indica que *synthetic* estalvia més d'un 6% en àrea i aconsegueix una reducció del retard màxim de més d'un 17%. Creiem que la major varietat de portes que es poden implementar en el cas de considerar un mapatge tecnològic orientat a generadors de mòduls aconsegueix que la reducció del retard es fonamenti en la reducció de la profunditat de la Xarxa Booleana gràcies a la implementació de portes més complexes en el circuit final, sense que això impliqui necessàriament la introducció de lògica duplicada.

Finalment, es presenten els resultats corresponents a l'optimització de la potència dissipada en els circuits. Donat que SIS no incorpora aquest objectiu de disseny, no ha estat possible fer una comparació semblant a la feta anteriorment per l'àrea i el

retard màxim. En aquest cas, la comparació es realitzarà entre diferents mètodes de cobertura i objectius de disseny en *synthetic*. La primera fila de la taula de resultats indica el mètode de mapatge tecnològic considerat: mapatge directe en *synthetic*, mapatge amb optimització local de retard màxim+àrea en *synthetic*, i mapatge amb optimització local de la dissipació de potència i del retard com a criteri secundari. Recordem que la minimització de la potència dissipada en el circuit es durà a terme en base a la reducció de la sensibilitat a generació de polsos espuris del circuit. Aquesta reducció dels polsos espuris s'aconseguirà balancejant, en la mesura del possible, les longituds de tots els camins presents en el circuit. Com que una optimització del retard màxim també podria tendir a balancejar els retards dels camins, com a mínim per a les sortides, hem cregut convenient realitzar la comparació entre l'optimització directa de polsos espuris i la minimització implícita que resulta durant l'optimització del retard màxim. Per cada mètode proposat de mapatge es mesuren els diferents objectius de disseny: àrea, retard màxim (pel temps de pujada i baixada) i sensibilitat a la generació de polsos espuris. Les tres darreres files de la taula resumeixen els resultats. La fila etiquetada amb *Total* indica el total acumulat per cada mètode de mapatge. La fila etiquetada com *X/Directa* compara els resultats del mapatge amb *synthetic* usant optimització local del retard màxim o de la sensibilitat a la generació de polsos espuris, indicant l'increment o disminució del cost en comparació amb el mapatge directe de *synthetic*. Finalment, la darrera fila, etiquetada amb *X/Retard* compara els resultats del mapatge amb *synthetic* usant optimització local de la sensibilitat a la generació de polsos espuris, indicant l'increment o disminució del cost en comparació amb l'optimització del retard màxim.

Mapatge	<i>synthetic</i> - Mapatge Directe				<i>synthetic</i> - Retard Màx.+Àrea				<i>synthetic</i> - Polsos Espuris+Àrea			
	Àrea	Puj.	Baix.	P. Es.	Àrea	Puj.	Baix.	P. Esp.	Àrea	Puj.	Baix.	P. Esp.
cm82a	64	15,5	12,5	50,5	70	12	9	20	72	13,5	9,5	14,5
cm151a	78	17	18	38,5	84	14	12,5	30,5	66	15,5	14	38
parity	180	24	20	135	180	24	20	135	190	31,5	27	128
cmb	134	31	22,5	54,5	190	27,5	27,5	307,5	178	32,5	28,5	218,5
cm163a	140	35	33,5	236	138	17	17,5	106,5	148	14	19	75
cm162a	124	24	21,5	162,5	128	20	20,5	127,5	150	20	23	116
cm150a	118	17,5	16	45,5	136	15,5	15,5	30	110	16,5	21,5	51,5
cm85a	108	21	17	56	118	21	23	104,5	114	21,5	25	106,5
cu	144	30,5	23	115,5	222	20,5	20,5	196	196	25,5	20,5	139,5
pml	110	20,5	21,5	88	142	18,5	15,5	72	154	22	11	61
pcler8	232	35	28	275	248	20	19,5	161	246	19,5	23,5	180
cc	172	24	13,5	157,5	198	12	9,5	103	222	12	6,5	38,5
Totals	1604	297		1414,5	1854		225	1393,5	1846		264,5	1167
X/Directa					15,59%	-24,24%		-1,48%	15,09%	-10,94%		-17,50%
X/Retard									-0,43%	17,56%		-16,25%

**Taula 6-4** Comparativa dels resultats obtinguts, quant a la generació de polsos espuris, segons el tipus de cobertura: directa o amb optimització local, amb el retard o la minimització de polsos espuris com a objectiu prioritari.

La Taula 6-4 mostra la comparació dels resultats obtinguts en *synthetic* usant diferents estratègies de mapatge tecnològic i objectius d'optimització. Els valors

obtinguts usant el mapatge directa de la Xarxa Booleana seran presos com a referència. La utilització d'una cobertura amb optimització local del retard màxim només optimitza marginalment la sensibilitat del circuit a polsos espuris, ja que aquesta es redueix poc més d'un 1%. En canvi, si l'objectiu prioritari de l'optimització és la sensibilitat a la generació de polsos, aquesta es redueix significativament, en més d'un 17%. Aquesta reducció no és gratuïta, ja que l'àrea s'incrementa en un 15%. Pel que fa al retard, aquest disminueix prop d'un 11%. De totes maneres, cal recordar que l'optimització de l'àrea com a objectiu prioritari aconseguia, com a valor afegit, una reducció de més del 19% en el retard màxim del circuit, i que l'optimització directa del retard màxim aconseguia un guany de més del 24% en terme mig. Des d'aquest punt de vista, és clar que el fet de balancejar els camins d'arribada a totes les portes del circuit suposa també un cost pel que fa al retard màxim del circuit. De totes maneres, cal esmentar que, tal com passava en el cas de l'optimització del retard màxim, la descomposició estructural realitzada en alguns nodes del circuit afecta notablement a la sensibilitat del circuit a la generació de polsos espuris. Només cal observar l'exemple de la Figura 6-4 per adonar-se que la descomposició no balancejada introdueix una alta sensibilitat a la generació de polsos, si les entrades originals arribaven al mateix temps. D'altra banda, si les entrades originals arribaven a la porta per camins de longitud diferent, la descomposició pot aprofitar-se per compensar aquestes diferències, introduint les entrades més lentes en les portes més properes a la sortida original de la funció. Aquesta optimització, tot i que prevista, no ha encara introduïda en *synthetic*.

Finalment, la comparació entre els mapatges amb objectius d'optimització local del retard màxim, o bé de la sensibilitat a la generació de polsos espuris, dona els resultats que caldria esperar. La variació en l'àrea és insignificant, ja que no arriba al 1%. Pel que fa a la sensibilitat a la generació de polsos espuris, aquesta es redueix en un 16% si es considera l'objectiu prioritari enlloc del retard màxim. Al mateix temps, el retard màxim es veu incrementat en més d'un 17%. Es fa evident, per tant, que l'optimització del retard màxim en un circuit no només no és un criteri suficient per reduir la sensibilitat del circuit a la generació de polsos espuris, sinó que pot ser un objectiu clarament contraposat.

## Conclusions

La darrera etapa de la síntesi lògica és el mapatge tecnològic, que s'encarrega de transformar una Xarxa Booleana independent de la tecnologia en un circuit implementable que minimitza una funció de cost associada als objectius de disseny. El mapatge tecnològic porta associades diferents problemàtiques. En primer lloc, cal generar la llista de possibles implementacions de cada node de la Xarxa Booleana original: és el procés d'agrupament, que s'ha estudiat en el capítol 5 d'aquesta tesi. Els possibles agrupaments generats de cada node de la Xarxa Booleana vindran determinats per la disponibilitat d'un ventall prou ampli de portes, ja que només poden ser tinguts en consideració aquells agrupaments implementables en la tecnologia destí. En un mapatge tecnològic clàssic, sobre una biblioteca de cel·les, el conjunt de portes disponibles és limitat. Per tant, per cada agrupament, cal obtenir el subconjunt de les cel·les de la biblioteca que el poden implementar. Aquest procés és coneix amb el nom d'aparellament. En el cas que ens ocupa, però, la problemàtica és

diferent. El conjunt de portes disponibles és molt més ampli, ja que cobreix totes les portes realitzables en una certa tecnologia, en aquest cas CMOS, on l'única restricció vindrà imposada per una limitació en el número màxim de transistors en sèrie. Donat que el ventall de possibles cel·les és molt gran, no es disposa d'una biblioteca de cel·les de la qual caldrà escollir-ne un subconjunt, sinó que directament es generarà la porta que suposi la millor implementació per a cada agrupament. El procediment per a la generació de portes complexes CMOS d'una etapa ha estat estudiat en el capítol 4 d'aquest treball.

Finalitzat el procés d'agrupament, cada node disposarà d'una o més possibles implementacions, amb les corresponents portes associades. El que queda per fer és la selecció d'un conjunt d'agrupaments a ser implementats, de tal manera que tots els nodes de la Xarxa Booleana quedin coberts i es minimitzi una funció de cost dependent dels objectius de disseny. Aquesta etapa es coneix amb el nom de cobertura de la Xarxa Booleana i ha estat objectiu d'estudi en aquest capítol.

Després de revisar els diferents mètodes de cobertura d'una Xarxa Booleana presents en la literatura, s'ha procedit a una descripció formal del problema. Donat que l'objectiu final del procés és cobrir la Xarxa Booleana amb un cost mínim, s'ha pensat en resoldre el problema aplicant algoritmes d'optimització d'una certa funció de cost. Aquesta funció de cost a minimitzar dependrà clarament dels requeriments de disseny exigits al circuit que s'està sintetitzant.

Per tal de definir la funció de cost que ha de guiar la cobertura de la Xarxa Booleana, s'ha procedit a l'estudi del modelat de cadascun dels objectius de disseny (àrea, retard, dissipació de potència). El fet de considerar un mapatge tecnològic orientat a un generador de mòduls, i no a una biblioteca de cel·les estàndard, permet de treballar amb models de cost més senzills i relativament independents de la tecnologia. Així, l'àrea es pot modelar en base al número de transistors en la porta; el retard d'una porta segons el número màxim de transistors en sèrie; i la dissipació de potència es pot reduir per mitjà de la reducció de la sensibilitat a la generació de polsos espuris en el circuit, és a dir, gràcies al correcte balanceig en temps dels camins que arriben a les portes del circuits. El motiu bàsic que permet aquestes simplificacions cal buscar-lo en la presència de l'etapa de dimensionament dels transistors del circuit en el procés de generació de mòduls. El dimensionament permet ajustar les cel·les del circuit als requeriments de les especificacions inicials, gràcies a la millor informació de la qual disposa, al ser aplicat amb les cel·les ja col·locades en el circuit. Tècniques similars han estat aplicades per optimitzar en temps el mapatge tecnològic sobre biblioteques de cel·les estàndard [LMK90, Fis92]. Per poder aplicar aquestes tècniques, les biblioteques han de disposar de diferents versions de cada cel·la, amb compromisos diferents dels objectius de disseny. L'optimització temporal es realitza després del mapatge tecnològic, procedint a la substitució selectiva de cel·les per versions diferents de la mateixa cel·la que minimitzin el cost del circuit. La utilització d'un generador de mòduls, però, permet molta més flexibilitat, ja que l'etapa de dimensionament pot canviar les dimensions de cada transistor individualment, la qual cosa li permet un moviment molt suau en l'espai de solucions.

Cal destacar també que les funcions de cost proposades per a cadascun dels objectius de disseny són en gran mesura independents de la tecnologia, la qual cosa permet la

seva aplicació durant les etapes d'optimització i reestructuració de la Xarxa Booleana. Creiem que l'aplicació de les mateixes funcions de cost en les etapes d'optimització lògica i mapatge tecnològic millorarà l'acoblament dels dos processos, de tal manera que serà possible assolir en el circuit final compromisos cada vegada més estrictes respecte dels objectius de disseny.

Un cop definida la funció de cost a minimitzar, s'ha procedit a l'estudi del procés de cobertura de la Xarxa Booleana. En el present capítol s'han proposat tres alternatives de resolució diferents. La primera d'elles consisteix en una cobertura directa dels nodes de la Xarxa Booleana, de manera que per cadascun d'ells s'implementa una porta en el circuit final. És evident que aquest tipus de cobertura és incapaç d'optimitzar la funció de cost, però ha estat estudiat perquè ens indica quin és el cost inicial de la Xarxa Booleana que arriba al procés de mapatge tecnològic. Aquest cost inicial ens servirà com a referència a l'hora de comparar la capacitat d'optimització dels objectius de disseny per part d'altres tipus de cobertures.

El segon mètode de cobertura proposat es basa en una optimització local de la funció de cost en cada node de la Xarxa Booleana. Aquesta tècnica no garanteix l'obtenció del mínim global de la funció de cost, únicament un mínim local. Malgrat això, els resultats obtinguts mostren la viabilitat i efectivitat d'aquesta tècnica.

Finalment, s'ha proposat un mètode de cobertura òptima de la Xarxa Booleana. Aquest es basa en un algoritme recursiu de bifurcació amb acotació, que explora tot l'espai de solucions del problema. L'acotació del cost necessari per assolir una solució, a partir d'un nivell qualsevol de la recursivitat, permet descartar ràpidament zones de l'espai de solucions que no portaran a una millora de la millor solució trobada fins al moment. També s'han proposat les modificacions necessàries per tal que l'algoritme sigui capaç d'obtenir un conjunt de solucions que suposin compromisos diferents entre els objectius de disseny, per tal que l'usuari pugui escollir la cobertura més adient a les seves necessitats.

La cobertura directa i la cobertura amb optimització local de la funció de cost han estat implementades en *synthetic*. Per tal de comparar la metodologia de mapatge tecnològic orientat a generadors de mòduls amb el mapatge clàssic sobre biblioteques de cel·les estàndard, s'han comparat els resultats obtinguts per SIS amb els obtinguts per *synthetic*, per a un mateix conjunt de circuits de prova i usant les mateixes funcions de cost. Quan el criteri principal és l'optimització de l'àrea, la millor cobertura de la Xarxa Booleana s'obté amb una cobertura directa, la qual cosa demostra l'efectivitat de l'optimització prèvia al mapatge tecnològic. Malgrat això, cal comentar que la cobertura directa no és possible per alguns circuits, ja que s'excedeixen els límits tecnològics en el número de transistors en sèrie. Considerant això, i comparant els resultats obtinguts per SIS i per *synthetic* amb optimització local, es pot veure que aquest darrer aconsegueix un estalvi d'àrea marginal respecte SIS, però en canvi redueix el retard màxim en més d'un 16%, la qual cosa demostra que la major flexibilitat en la implementació dels nodes de la Xarxa Booleana, si bé no pot ser molt aprofitada per a la reducció d'àrea, sí que ho és per la reducció del retard màxim.

Quan l'objectiu prioritari a optimitzar és el retard màxim en el circuit, *synthetic* amb optimització local aconsegueix una reducció del 17% en mitjana sobre els resultats



aconseguits per SIS, acompanyat per una reducció de més del 6% en àrea. El fet que la reducció del retard màxim s'acompanyi d'una reducció en l'àrea, en comparació amb SIS, sembla indicar que l'optimització del retard màxim en el mapatge tecnològic sobre biblioteques de cel·les estàndard introdueix duplicació en la lògica, mentre que si es considera un mapatge orientat a generadors de mòduls, aquesta reducció del retard màxim pot fer-se sovint en base a una reducció de la longitud dels camins gràcies a la introducció de funcions més complexes en els nodes de la Xarxa Booleana.

Finalment, al no incorporar SIS mètodes que permetin fer estimacions sobre la dissipació de potència o la sensibilitat dels circuits a la generació de polsos espuris, els resultats en aquest cas corresponen únicament a *synthetic*. D'una banda, s'ha observat que la minimització de la sensibilitat del circuit a la generació de polsos redueix aquesta en un 17% respecte de la cobertura directa del circuit, però amb un cost en àrea del 15% i en temps de prop del 11%. Aquests resultats també han estat comparats amb els que s'obtenen a l'optimitzar el retard màxim per tal d'estudiar la possible relació entre els dos problemes. Els resultats mostren que l'optimització específica de la sensibilitat del circuit a la generació de polsos espuris redueix aquests en un 16% en comparació amb els resultats obtinguts quan s'optimitza el màxim retard, mentre que aquest últim es veu incrementat en més d'un 17%. Els resultats demostren que l'optimització del retard màxim no necessàriament implica un balanceig en els camins presents en el circuit, que suposarien una reducció de la sensibilitat a la generació de polsos espuris.

Cal destacar finalment que la major flexibilitat que implica el mapatge tecnològic orientat a un generador de mòduls en contraposició a un mapatge clàssic sobre biblioteques de cel·les estàndard es tradueix en l'obtenció d'una reducció sensible en el cost dels circuits obtinguts. També és important assenyalar que no només es redueix el cost que suposa el criteri principal sobre el que s'optimitza, sinó que els criteris secundaris també poden ser reduïts significativament. Creiem que aquest fet fa especialment idònia la metodologia de síntesi lògica orientada a la generació de mòduls quan les restriccions imposades al disseny són més estrictes, com és el cas de les noves tecnologies submicròniques.

---

# Capítol 7

---

## Conclusions

*En aquest capítol es presenten les conclusions i contribucions del present treball en el camp de la síntesi lògica orientada a generadors de mòduls, així com les possibles línies de treball futur.*

### Introducció

El present treball s'emmarca dins d'una metodologia de síntesi lògica lligada a un generador de mòduls guiat per prestacions. És evident que el generador de mòduls disposarà d'informació tecnològica molt detallada i, per tant, serà capaç d'assolir compromisos molt més restrictius sobre els objectius de disseny. En concret, l'etapa de dimensionament de transistors és capaç de generar qualsevol versió implementable d'una porta. Aquest fet contrasta amb el número limitat de versions d'una porta que poden estar disponibles en una biblioteca de cel·les estàndard. D'altra banda, caldrà disposar d'una eina de síntesi lògica capaç de generar qualsevol porta implementable en una tecnologia determinada. D'aquesta manera la flexibilitat i suavitat de moviments en l'espai de solucions augmenta considerablement, cosa que garanteix l'acompliment de restriccions de disseny molt més estrictes que les que són assolibles en una metodologia basada en biblioteques de cel·les estàndard.

### Contribucions

L'estudi previ de la problemàtica associada a la creació, caracterització, manteniment i verificació de biblioteques de cel·les estàndard sobre diferents tecnologies, que ha estat publicat en diverses conferències [**RPRCT91**, **RRPCT91**, **RRPCT91a**, **RRPCT91b**, **RRPCT92**, **RRPSCT92**, **CPRST92**, **SRR92**, **RRPSCT92a**, **RRPSCT93**, **RRSPCT93**, **RRCT93**, **RRPSCT93a**, **CRF95**], ha motivat l'interès de l'autor pels processos de síntesi lògica automàtica guiada per prestacions. L'estudi de l'aplicació de funcions de cost més properes a la tecnologia durant totes les etapes de la síntesi lògica tradicional, usant biblioteques de cel·les estàndard, ha mostrat la

dificultat d'obtenir funcions de cost que siguin a l'hora generals i lligades a la tecnologia. Els resultats obtinguts han estat publicats en [RRVC95, RVRVC95, RRVC96]. Aquest estudi va motivar plantejar una metodologia diferent, on la síntesi lògica ja no es fes sobre una biblioteca de cel·les estàndard, sinó sobre un generador de mòduls guiat per prestacions. Aquesta metodologia ha estat proposada en [VMRPC96] i resultats parcials sobre la generació eficient de mòduls han estat publicats [VRRPC96, VMRPC96]. La part més específica de la síntesi lògica orientada a un generador de mòduls guiat per prestacions és el mapatge tecnològic, que és substancialment diferent al mapatge tecnològic sobre una biblioteca de cel·les estàndard.

El treball presentat en aquesta tesi contribueix a la síntesi lògica orientada a un generador de mòduls guiat per prestacions justament en el camp del mapatge tecnològic i concretament en els següents aspectes:

- L'estudi de la *implementació de portes CMOS* d'una sola etapa directament a partir de l'estructura del BDD que representa la funció de la porta.
- L'estudi, definició i avaluació de la complexitat del *procés d'agrupament*, que consisteix en la generació de les possibles implementacions de cada node de la Xarxa Booleana.
- L'estudi i definició de *funcions de cost* pels diferents objectius de disseny, que tant poden ser usades durant el mapatge tecnològic com en l'etapa prèvia d'optimització de la Xarxa Booleana.
- L'estudi, definició i avaluació de diferents *algoritmes de cobertura* de la Xarxa Booleana.
- La *comparació de la metodologia* proposada amb la metodologia basada en una biblioteca de cel·les estàndard.

A continuació, s'analitzaran les contribucions del treball en cadascun dels aspectes esmentats anteriorment.

## **Implementació de portes CMOS**

A ben segur, el fet bàsic que diferencia la metodologia clàssica, basada en síntesi lògica sobre biblioteques de cel·les estàndard, de la metodologia proposada de síntesi lògica orientada a generadors de mòduls, és justament el fet que la primera disposa d'un conjunt més o menys reduït de portes a escollir a l'hora d'implementar el circuit final, mentre que la segona disposa de totes les portes implementables en la tecnologia. Aquestes portes, però, no estan disponibles prèviament a l'inici del mapatge tecnològic, sinó que s'aniran generant a mesura de les necessitats de la cobertura de la Xarxa Booleana.

En aquest aspecte, el treball desenvolupat ha contribuït d'una banda en proporcionar algoritmes per a la implementació de qualsevol funció Booleana en una porta CMOS d'una sola etapa, considerant possibles inversions a les entrades, i de l'altra en establir els resultats teòrics que permeten saber quan aquesta implementació resulta

ser l'òptima. Donat que les funcions Booleanes realitzades en els nodes de la Xarxa Booleana són representades per BDDs en *synthetic*, com també ho són en la majoria d'eines de síntesi lògica, s'ha desenvolupat un algoritme que permet la generació directa de la porta que implementa una funció a partir de l'estructura del propi BDD que la representa. S'ha demostrat la validesa d'aquest algoritme per a totes les funcions monofàsiques, també anomenades de fase no binària. La generació de la implementació de les funcions de fase binària s'ha realitzat usant algorismes clàssics de factorització, basats en les operacions de divisió algebraica i generació de nuclis d'una expressió.

Per tal d'estudiar l'aplicabilitat de l'algoritme proposat i la seva eficàcia, s'han establert els límits teòrics que permeten saber el número mínim de transistors requerits per implementar una funció Booleana qualsevol en una porta CMOS d'una sola etapa, i s'han analitzat un conjunt de circuits de prova per tal de veure el percentatge en el qual aquestes implementacions òptimes apareixen inicialment en els circuits. Els resultats obtinguts mostren que:

- La gran majoria de funcions presents inicialment en els circuits de prova poden ser implementades òptimament. En concret, el 99% de les funcions, que suposen el 95% dels transistors requerits per implementar les portes del circuit i el 98% dels transistors que podrien aparèixer en els inversors necessaris per generar la fase correcta de les entrades de cada porta.
- L'associació directa de transistors als arcs dels BDDs que representen les funcions en els nodes de la Xarxa Booleana produeix la implementació òptima de la porta corresponent per un 95% dels nodes en els circuits de prova. Pel que fa a transistors, aquestes funcions suposen el 86% dels requerits per la implementació de les portes i el 88% dels que podrien aparèixer en els inversors necessaris per generar la fase correcta de les entrades de cada porta.
- La informació estructural disponible en els BDDs que representen les funcions dels nodes de la Xarxa Booleana original pot ser utilitzada durant les primeres etapes de la síntesi lògica com a mesura del cost del circuit. No cal, per tant, implementar les funcions dels nodes de la Xarxa Booleana per tal de fer una avaluació prou acurada del cost del circuit. El número de nodes del BDD pot usar-se directament com a mesura de l'àrea (cada node representarà dos transistors). El número màxim d'arcs de tipus  $V(F)$  que es trobin en els camins que van des de l'arrel del BDD fins al node terminal  $0(I)$  estimen el número de transistors  $N(P)$  en sèrie de la porta que implementa la funció, i pot ser utilitzat com a indicació del retard de la porta. Al seu torn, la dissipació de potència pot ser avaluada en base a l'activitat de commutació de les funcions dels nodes de la Xarxa Booleana, activitat que es pot obtenir a partir de les probabilitats (de senyal i/o de transició) de les entrades primàries i de l'estructura del BDD que representa la funció. També la sensibilitat de la Xarxa Booleana a la generació de polsos espuris pot ser avaluada usant les estimacions del retard dels nodes que es troben en el ventall d'entrada de cada node de la Xarxa Booleana.

## El procés d'agrupament

Si bé el procés d'agrupament també apareix en el mapatge tecnològic sobre biblioteques de cel·les, la seva complexitat es veu altament incrementada quan es considera un mapatge tecnològic orientat a generadors de mòduls. Efectivament, el procés d'agrupament per un node de la Xarxa Booleana consisteix en obtenir el conjunt de portes que poden cobrir el node  $i$ , potser, també alguns nodes en el seu ventall transitiu d'entrada. La complexitat d'aquest procés depèn del número màxim d'entrades que pot tenir una porta. En el cas del mapatge tecnològic sobre biblioteques de cel·les, el número d'entrades és relativament baix, cosa que simplifica el procés d'agrupament, però en el cas de considerar un mapatge tecnològic orientat a generadors de mòduls, aquest número d'entrades pot ser relativament alt.

En el capítol 5 d'aquest treball s'ha descrit el problema de la generació d'agrupaments per tots els nodes de la Xarxa Booleana i se n'ha estudiat la complexitat. Aquesta, augmenta exponencialment per cada nou nivell d'entrades transitives que és considerat. El fet que el mapatge tecnològic orientat a generadors de mòduls permet un més gran ventall d'entrades en les portes provoca que aquest procés esdevingui el punt crític del mapatge tecnològic orientat a generadors de mòduls, mentre que no ho era pel mapatge tecnològic sobre biblioteques de cel·les estàndard.

El procés de generació de nous agrupament implica una composició de funcions, que és una operació complexa, sobretot quan les funcions involucrades són grans. A més, cal tenir present que molts dels agrupaments generats amb aquestes composicions de funcions resulten ser no implementables en la tecnologia destí (ja que superen el número màxim de transistors en sèrie permesos) i són descartats tot seguit. Més de la meitat dels agrupaments presents en els circuits són descartats. Aquest fet ha motivat una altra contribució del present treball, que és el desenvolupament d'heurístiques encaminades a detectar agrupaments que resultaran no implementables i descartar-los abans de fer la composició de funcions. Les heurístiques proposades es poden classificar en dos grans grups segons la idea en la qual es fonamentin:

- Estimació de la complexitat de l'agrupament resultant. Tracten de predir quina serà la complexitat de la funció resultant de la composició de funcions, usant únicament la informació disponible sobre els grups a compondre. Engloba les estratègies *maxSerial*, *addSerial* i *incSupport*.
- Definició jeràrquica dels agrupaments. Estableixen una jerarquia entre els diferents agrupaments que pot tenir cada node de la Xarxa Booleana i tracten de predir la complexitat dels nivells inferiors de la jerarquia en base als nivells superiors. Engloba les estratègies *List* i *Hierarchy*.

El present treball ha proposat un algorisme per resoldre el procés de generació de tots els possibles agrupaments dels nodes d'una Xarxa Booleana, que ha estat implementat en *synthetic* juntament amb les heurístiques proposades. L'eficàcia d'aquestes últimes ha estat comprovada gràcies a la seva aplicació en el procés de generació d'agrupaments en un conjunt de circuits de prova. Els resultats obtinguts mostren que:

- L'aplicació d'estratègies de filtrat d'agrupaments no implementables permet eliminar un elevat percentatge d'aquests sense haver-los de generar (estalvia la composició de funcions).
- Les estratègies basades en una estimació de la complexitat aconseguixen eliminar un elevat percentatge d'agrupaments incorrectes (entre el 70% i el 80%), però també descarten erròniament un percentatge significatiu d'agrupaments correctes (entre el 34% i el 42%).
- Les estratègies basades en la definició jeràrquica dels agrupaments són més conservadores. La seva capacitat de filtrat és reduïda, però a canvi el nombre d'agrupaments correctes que són descartats és també relativament petit. D'altra banda, aquests tipus d'heurístiques resulten ser més sensibles a la propagació d'errors, en el sentit que un agrupament descartat erròniament pot implicar la no generació de tot un altre conjunt d'agrupaments.

Per últim, destacar que aquestes estratègies podrien ser aplicades igualment en el mapatge tecnològic sobre biblioteques de cel·les estàndard.

## **Funcions de cost**

Les funcions de cost associades als objectius de disseny són veritablement importants, ja que són elles les que guiaran les successives optimitzacions de la Xarxa Booleana. En aquest aspecte, creiem que la contribució del treball ha estat prou notable, i en destacariem dos aspectes. En primer lloc, s'han proposat funcions de cost senzilles que poden ser aplicades directament en totes les etapes de la síntesi lògica, tant en l'optimització de la Xarxa Booleana com en el mapatge tecnològic. En segon lloc, aquestes funcions de cost, encara que senzilles, estan ben correlacionades amb els objectius de disseny i es fonamenten en la utilització d'un generador de mòduls guiat per prestacions, que incorpora una etapa de dimensionament de transistors.

És important insistir en el fet que les funcions de cost proposades poden ser utilitzades durant l'optimització de la Xarxa Booleana. A més, un percentatge molt important de les funcions presents originalment en els nodes dels circuits permeten el càlcul directe d'aquestes funcions de cost a partir de la informació estructural disponible en els BDDs que les representen, tal com s'ha comentat anteriorment. Creiem que l'aplicació d'aquestes funcions de cost en l'etapa d'optimització de la Xarxa Booleana facilitarà l'obtenció de compromisos restrictius en els objectius de disseny, al poder proporcionar una millor Xarxa Booleana inicial al procés de mapatge tecnològic.

## **Algoritmes de cobertura de la Xarxa Booleana**

La cobertura d'una Xarxa Booleana consisteix en la selecció d'un conjunt de nodes dels quals se n'escull un dels agrupaments i una implementació per a aquest, bé en la forma directa o en la complementària, de tal manera que quedi coberta la Xarxa Booleana amb un cost mínim. El cost de la coberta vindrà determinat per les funcions de cost definides sobre els objectius de disseny.

El mapatge tecnològic orientat a generadors de mòduls, objecte de l'estudi d'aquest treball, es redueix a la cobertura d'una Xarxa Booleana de la qual se n'ha obtingut el conjunt d'agrupaments implementables per a cadascun dels seus nodes. Un aspecte rellevant del treball presentat ha estat el fet de formular el procés de cobertura com un problema d'optimització d'una funció de cost. En aquest sentit, s'han proposat tres diferents estratègies de cobertura de la Xarxa Booleana.

La primera d'elles és la cobertura directa de la Xarxa Booleana, on cada node és implementat usant la forma directa del seu agrupament bàsic. D'aquesta manera, a cada node de la Xarxa Booleana li correspon una porta equivalent en el circuit final. Únicament per a algunes funcions realitzades en els nodes de la Xarxa Booleana s'introdueixen els inversors necessaris per tal de generar la fase correcta d'aquelles variables que són requerides inicialment en fase positiva. És evident que aquesta cobertura serà incapaç d'optimitzar la funció de cost, ja que no té opció a escollir la implementació dels nodes de la xarxa però, en canvi, serà usada com a referència per comparar els resultats obtinguts usant altres algoritmes de cobertura.

El segon mètode proposat es basa en l'optimització local de la funció de cost. En cada node de la Xarxa Booleana, es selecciona el millor agrupament i la millor implementació per a aquest, basant-nos en les millors implementacions dels nodes que es troben en el suport de l'agrupament considerat. Aquest mètode permet assolir un mínim local de la funció de cost, però no garanteix l'obtenció del mínim global.

Finalment, el tercer mètode de cobertura de la Xarxa Booleana que ha estat proposat consisteix en la cobertura òptima d'aquesta gràcies a la utilització d'un algoritme de bifurcació amb acotació de la funció de cost.

Tant la cobertura directa com la cobertura amb optimització local han estat implementades en *synthetic*, mentre que la cobertura òptima ha quedat pendent per a un futur. Una comparació dels resultats obtinguts pels dos mètodes de cobertura en el mapatge tecnològic d'un mateix conjunt de circuits de prova ha permès extreure les següents conclusions:

- Quan el principal criteri d'optimització és l'àrea, la millor cobertura de la Xarxa Booleana s'obté usant la cobertura directa. Aquests resultats indiquen que l'optimització de la Xarxa Booleana realitzada per SIS com a pas previ al mapatge tecnològic, és eficient i aconsegueix una molt alta reutilització de les subfuncions que apareixen en la Xarxa Booleana. Cal dir, però, que la cobertura directa no és implementable en alguns circuits, ja que algunes funcions presents en els nodes superen el límit tecnològic pel que respecte al nombre màxim de transistors en sèrie.
- L'optimització de l'àrea com a criteri principal quan s'utilitza una cobertura amb optimització local suposa un increment del 5% respecte d'una cobertura directa, però al mateix temps aconsegueix una reducció del 19% en el retard màxim del circuit. Aquests resultats mostren una optimització important en el criteri secundari, en aquest cas el retard màxim, la qual cosa indica que la flexibilitat en la selecció de la implementació dels nodes de la Xarxa Booleana és ben aprofitada per l'algoritme.

- Si es comparen els resultats obtinguts pels dos mètodes quan el criteri principal és el retard, es veurà que aquest és reduït en un 24% usant la cobertura amb optimització local. Aquesta reducció té un cost important en àrea: un increment del 15%.
- Per que fa a l'optimització de la sensibilitat del circuit a la generació de polsos espuris, s'ha pogut comprovar que aquesta no es redueix significativament, respecte dels valors obtinguts usant la cobertura directa, si l'objectiu principal de l'optimització és el retard màxim en el circuit. En canvi, una optimització explícita de la sensibilitat a la generació de polsos aconseguix reduir-la en un 17%, amb un increment en àrea del 15% respecte de la cobertura directa. Els resultats indiquen que l'optimització del retard màxim no significa necessàriament un balanceig dels camins que arriben a cadascuna de les portes del circuit, més aviat el contrari. És evident que cal considerar explícitament la sensibilitat a la generació de polsos espuris per tal de reduir aquest paràmetre, i no es pot suposar la seva reducció implícita a l'optimitzar el retard màxim en el circuit.

## Comparació metodològica

Segurament la contribució més important del present treball consisteix en la comparació dels resultats obtinguts usant una metodologia clàssica de mapatge tecnològic sobre biblioteques de cel·les, i la nova metodologia proposada de mapatge tecnològic orientat a generadors de mòduls.

El mapatge tecnològic sobre biblioteques de cel·les s'ha fet utilitzant SIS, i el mapatge tecnològic orientat a generadors de mòduls s'ha fet usant *synthetic* amb el mètode de cobertura amb optimització local. Per tal de fer una comparació en igualtat de condicions, s'han modificat les funcions de cost associades a les cel·les de la biblioteca subministrada a SIS, *lib2.genlib*, de tal manera que reflectissin les mateixes funcions de cost manipulades per *synthetic*. D'aquesta manera, el que es compara bàsicament és la metodologia i no l'eficàcia de les funcions de cost.

Abans de fer el mapatge tecnològic, els circuits han estat prèviament optimitzats usant SIS. La comparació dels resultats obtinguts permet extreure les següents conclusions:

- Quan el criteri d'optimització és l'àrea, els dos mètodes obtenen pràcticament els mateixos resultats, encara que *synthetic* estalvia un 1% de l'àrea. Al nostre entendre, aquest resultat únicament indica que l'optimització de la Xarxa Booleana feta per SIS com a pas previ del mapatge tecnològic és bona. Així, la Xarxa Booleana subministrada al procés de mapatge tecnològic probablement sigui mínima en àrea, com així ho semblen indicar els resultats obtinguts per la cobertura directa de la Xarxa Booleana. D'aquesta manera, les optimitzacions només poden ser realitzades en aquells nodes que no poden ser implementats directament i han de ser descomposts. Cal destacar, però, que tot i que l'àrea no es redueix significativament en el mapatge tecnològic orientat a generadors de mòduls, sí que ho fa el retard màxim, que es redueix en un 16%. Per tant, queda clar que la major flexibilitat del mapatge tecnològic orientat a generació de



mòduls és aprofitada. Si bé l'àrea no pot ser reduïda, sí que ho és retard màxim, com a criteri secundari.

- Si el criteri d'optimització és el retard màxim, la metodologia de mapatge tecnològic orientat a generadors de mòduls supera àmpliament la metodologia basada en biblioteques de cel·les estàndard, ja que redueix tant el retard màxim, en un 17%, com l'àrea, un 6%. Al nostre entendre, aquests resultats indiquen que l'optimització del retard màxim suposa sovint la introducció de redundància lògica en el mapatge tecnològic sobre biblioteques de cel·les, mentre que en el cas del mapatge tecnològic orientat a generació de mòduls es pot aprofitar una més gran flexibilitat en la selecció de la implementació de cada node de la Xarxa Booleana, la qual cosa permet reduir la longitud dels camins sense incrementar l'àrea. Val a dir que aquests resultats encara podrien ser millors si la descomposició implementada en *synthetic*, i requerida en alguns circuits, fos balancejada en temps i no estructural.

Finalment, cal comentar que SIS no disposa de cap avaluació sobre la dissipació de potència, pel que l'optimització d'aquest objectiu no ha pogut ser comparada. Malgrat això, creiem que la major flexibilitat en l'elecció de la implementació dels nodes de la Xarxa Booleana ha de donar a la força els seus fruits, traduïts en una optimització millor dels objectius de disseny i en la possibilitat d'assolir compromisos restrictius d'aquests objectius.

## Treball futur

Molts són els camins que queden oberts a la vista del treball presentat, que cobreixen des de l'extensió de la metodologia cap als nivells superiors de la síntesi lògica fins a l'estudi de l'evolució tecnològica futura i els possibles canvis que aquesta produirà, especialment pel que fa a les funcions de cost que guien els processos de síntesi lògica i física.

Si ens concentrem, però, en un futur proper podríem citar els següents aspectes com a objectius prioritaris:

- Incorporació d'algoritmes de descomposició balancejada de funcions. Aquests algoritmes suposarien una millora immediata en els resultats obtinguts en els circuits on la descomposició de nodes és necessària per a la seva implementació.
- Implementació de la cobertura òptima d'una Xarxa Booleana. Deixant de banda el fet que es podrien obtenir les cobertes òptimes dels circuits, la implementació de la cobertura òptima d'una Xarxa Booleana permetria l'obtenció de compromisos reals, definits per l'usuari, entre els diferents objectius de disseny.
- Optimització de la Xarxa Booleana usant les funcions de cost proposades. L'optimització de la Xarxa Booleana guiada per les mateixes funcions de

cost que guien el mapatge tecnològic, proporcionaria un més bon acoblament entre els dos processos, pel que els compromisos restrictius entre els diferents objectius de disseny serien més fàcilment assolibles.

A mig termini, un objectiu de recerca que apareix com a evolució directa cap a nivells d'abstracció més elevats és el traslladar aquesta metodologia de resolució basada en funcions de cost a la síntesi a nivell arquitectural, podent arribar fins i tot al codisseny *hardware/software*. Aquesta afirmació es fonamenta en el fet que tant les estructures de dades com les funcions de cost manipulades en aquest nivell són molt similars a la dels diagrames de decisió binaris dels quals se n'ha fet un ús i estudi extens en aquest treball.

## Referències Bibliogràfiques

- [AA94] J.Akita, K.Asada, *A Method for Reducing Power Consumption of CMOS Logic Based on Signal Transition Probability*, Proc. of the EDAC, pp. 420-424, 1994.
- [Ake78] S.B.Akers, *Binary decision diagrams*, IEEE Transactions on Computers, vol. C-27, pp. 509-516, 1978.
- [AS90] M.Afghahi, C.Svenson, *A Unified Single Phase Clocking Scheme for VLSI Systems*, IEEE Journal of Solid State Circuits, vol. 25, no. 1, pp. 225-233, 1990.
- [Bak90] H.B.Bakoglu, *Circuit Interconnections and Packaging*, Addison Wesley, 1990.
- [BBFS93] T.Besson, H.Bouzouzou, I.Floricica, G.Saucier, *Input Order for ROBDDs Based on Kernel Analysis*, Proc. of the Euro-ASIC, pp. 266-272, 1993.
- [BCA88] L.Brocco, S.P.McCormick, J.Allen, *Macromodellin CMOS circuits for Timing Simulation*, IEEE Trans. On CAD, vol. 7, no. 12, pp. 1237-249, December 1988.
- [BCGH86] K.Bartlett, W.Cohen, A.DeGeus, G.Hachtel, *Synthesis and Optimization of Multilevel Logic under Timing Constraints*, IEEE Trans. On CAD, vol. CAD-5, no. 4, pp. 582-596, October 1986.
- [BCL95] Bradley S. Carlson, Suh-Juch Lee, *Delay Optimization of Digital CMOS VLSI Circuits by Transistor Reordering*, IEEE Trans. on CAD, vol. 14, pp. 1183-1192, Oct. 1995.
- [Ber88] R.A.Bergamaschi, *Automatic Synthesis and Technology Mapping of Combinational Logic*, Proc. of the ICCAD, pp. 466-469, 1988.
- [BFOR93] L.Benini, M.Favalli, P.Olivo, B.Ricco, *A Novel Approach to Cost-Effective Estimate of Power Dissipation in CMOS ICs*, Proc. of the EDAC. pp. 354-360, 1993.
- [BHS90] R.K.Brayton, G.D.Hachtel, A.L.Sangiovanni-Vincentelli, *Multilevel Logic Synthesis*, Proc. of the IEEE, vol.78, no.2, February 1990.
- [BL92] J.R.Burch, D.E.Long, *Efficient Boolean Function Matching*, Proc. of the ICCAD, pp. 408-411, 1992.
- [BM82] R.K.Brayton, C.McMullen, *The Decomposition and Factorization of Boolean Expressions*, Proc. of the ISCAS, 1982.

- [BMK87] G.Buurma, P.Michel, T.Turoda, *A High Performance Scalable Standard Cell Library with true Secon Sourcing*, Proc. of the CICC, pp. 241-244, 1987.
- [BNYT93] F.N.Najm, *A Monte Carlo approach for Power Estimation*, IEEE Trans. on VLSI Systems, vol. 1, no. 1, pp. 63-71, 1993.
- [Boo1847] G.Boole, *An investigation of the laws of thought*, Capitol 5, Walton (Londres), 1847. (Reeditat per Dover Books, NY, 1954).
- [Bra87] R.K.Brayton, *Factoring of Logic Functions*, IBM Journal of Research and Development, vol. 31, no. 2, March 1987.
- [Brace] K.S.Brace, *bdd - a BDD package*, Technical Report at Carnegie Mellon University.
- [BRB90] K.S.Brace, R.L.Rudell, R.E.Bryant, *Efficient Implementation of a BDD Package*, Proc. of the 27<sup>th</sup> DAC, pp. 40-45, 1990.
- [BRKM91] K.M.Butler, D.E.Ross, R.Kapur, M.R.Mercer, *Heuristics to Compute Variable Orderings for Efficient Manipulation of Ordered Binary Decision Diagram*, Proc. of the 28th DAC, pp. 417-420, 1991.
- [Bro90] F.M.Brown, *Boolean reasoning : the logic of Boolean equations*, Kluwer Academic Publishers, 1990.
- [BRSW87] R.K.Brayton, R.Rudell, A.Sangiovanni-Vincentelli, A.R.Wang, *MIS: A Multiple-Level Logic Optimization System*, IEEE Trans. On CAD, vol. 6, no. 6, November 1987.
- [Bry86] R.E.Bryant, *Graph-Based Algorithms for Boolean Function Manipulation*, IEEE Trans. on Computers, vol. C-35, no. 8, August 1986.
- [BSH92] Bo Wu, N.A. Sherwani, N.D. Holmes, M. Sarrafzadeh, *Over-the-Cell Routers for New Cell Model*, Proc of the 29<sup>th</sup> DAC, pp. 604-607, 92.
- [Bur91] J.R.Burch, *Using BDDs to Verify Multipliers*, Proc. of the 28th DAC, pp. 408-412, 1991.
- [Cap96] R.Capillas, *C++ Front-End for Binary Decision Diagram Packages*, Personal Communication, UAB, 1996.
- [CB88] S.Chowdhury, J.S.Barkatullah, *Current Estimation in Currents in MOS IC Logic Circuits*, Proc. of the ICCAD, 1988.
- [CB90] S.Chowdhury, J.S.Barkatullah, *Estimation of Maximum Currents in MOS IC Logic Circuits*, IEEE Trans. on CAD, vol. 9, no. 6, pp. 642-654, June 1990.

- [CD93] J.Cong, Y.Ding, *On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping*, Proc. of the 30<sup>th</sup> DAC, pp. 213-218, 1993.
- [CG92] S.Carlson, E.Girczic, *Understanding synthesis begins with knowing the terminology*, EDN, pp. 125-131, 1992.
- [Cha89] S.Chakravarty, *On the Complexity of using BDDs for the Synthesis and Analysis of Boolean Circuits*, Proc. of the 27th Annual Allerton Conf. on Communication, Control and Computing, pp. 730-739, 1989.
- [Cha93] S.Chakravarty, *A Characterization of Binary Decision Diagrams*, IEEE Trans. on Computers, vol. 42, no. 2, pp. 129-137, Feb. 1993.
- [Chen88] H.Y.Chen et al., *iCOACH: A Circuit Optimization Tool for CMOS High-Performance Circuits*, Proc. of the ICCAD, pp 372-375, 1988.
- [ChM93] S.Ch.Chang, M.Marek-Sadowska, *BDD Representation of Incompletely Specified Functions*, Proc. of the IWLS, 1993.
- [ChRP94] T.Chou, K.Roy, S.Prasad, *Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching*, Proc. of the ICCAD, pp. 300-303, 1994.
- [Cir87] M.A.Cirit, *Estimating dynamic power consumption of CMOS circuits*, Proc. of the ICCAD, pp. 534-537, 1987
- [CM95] O.Coudert, J.C.Madre, *New Ideas for Solving Covering Problems*, Proc. of the 32nd DAC, 1995.
- [COM95a] COMPASS, *Mercury library Development tools. Product Information*, 1995.
- [COM95b] COMPASS, *Deep Submicron Design. Product Information*, 1995.
- [Cor95] J.Cortadella, *Mapping BDDs into DCVSL gates*, Technical Report No. RR/95/04, UPC/DAC, 1995.
- [CP87] J.P. Cohoon and W.D. Paris, *Genetic Placement*, IEEE Trans. on CAD, vol. 6, pp. 956-964, November 1987
- [CP95] K.Chaudhary, M.Pedram, *Computing the Area versus Delay Trade-Off Curves in Technology Mapping*, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 14, no. 12, pp. 1480-1489, Dec. 1995.
- [CPRST92] J.Carrabina, J.M.Pérez, Ll.Ribas, J.Riera, J.Sáiz, Ll.Terés, *Nuevas Metodologías de diseño full-custom: concepción de un multiplicador paralelo*, VII DCIS, pp. 499-500, 1992.

- [CR92] K.Chung, J.Rose, *TEMPT: Technology Mapping for the Exploration of FPGA Architectures with Hard-Wired Connections*, Proc. of the 29<sup>th</sup> DAC, pp. 361-367, 1992.
- [CRC96a] R.Capillas, J.Riera, J.Carrabina, *Front-End C++ para paquetes de BDDs*, Actas del II Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, (en prensa), 1996.
- [CRC96b] R.Capillas, J.Riera, J.Carrabina, *bddDraw: Mostrador de Estructuras BDD*, Actas del XI DCIS, (en prensa), 1996.
- [CRF95] J.Carrabina, J.Riera, D.Flandre, *CMOS SOI & Bulk: Cell Level Comparison*, X DCIS, pp. 20-23. 1995.
- [CT90] J.Carrabina, L.Terés, *Diseño de una Biblioteca de Celdas Estándar CMOS*, V Escuela de Microelectrónica, pp 347-366, ISBN: 84-338-1279-3, Granada, 1990.
- [CWL96] K.N.Chen, T.S.Wang, Y.T.Lai, *A New Strategy of Performance-Directed Technology Mapping Algorithm for LUT-Based FPGAs*, Proc. of the ISCAS, pp. 822-825, 1996.
- [Dev93] S.Devadas, *Comparing Two-Level and Ordered Binary Decision Diagrams Representations of Logic Functions*, IEEE Trans. on CAD of Integrated Circuit and Systems, vol.12, no.5, pp. 722-723, May 93.
- [DGRSW87] E.Detjeus, G.Gannot, R.Rudell, A.Sangiovanni-Vincentelli, A. Wang, *Technology Mapping in MIS*, Proc. of the ICCAD, pp. 116-119, 1987.
- [DHNTB95] A.C.Deng, X.Huang, S.Napper, J.Tuan, J.Benkoski, *Simulation Algorithms, Power Estimation and Diagnostics in PowerMill*, Proc. of the PATMOS'95 Conf., pp. 399-410.
- [DKW92] S.Devadas, K.Keutzer, J.White, *Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation*, IEEE Trans. on CAD, vol. 11, no. 3, pp. 373-383, March 1992.
- [DLRB93] F.Dresig, Ph.Lanches, O.Rettig, U.G. Baitinger, *Simulation and Reduction of CMOS Power Dissipation at Logic Level*, Proc. of the EDAC, pp. 341-346, 1993.
- [DM95] S.Devadas, S.Malik, *A Survey of Optimization Techniques Targeting Low Power*, Proc. of the 32nd DAC, pp. 242-247, 1995.
- [DNS94] H.N.Nguyen, L.Ducouso, D.Schnapp, *Design of a Standard Cell Library for Synthesis*, Proc. of the PATMOS, pp. 194-199, 1994.
- [DP95] C.S.Ding, M.Pedram, *A Comparative Study of Switching Activity Estimation Techniques*, Proc. of the PATMOS'95 Conf., pp. 377-396.

- [DSL88] A.C.Deng, Y.C.Shiau, H.H.Loh, *Time Domain Current Waveform Simulation of CMOS Circuits*, Proc. of the ICCAD, pp. 208-211, 88.
- [EB95] M.Eisele, J.Berthold, *Dynamic Gate Delay Modelling for Accurate Estimation of Glitch Power at Logic Level*, Proc. of the PATMOS'95 Conf., pp. 190-201.
- [EBE93] S.Embabi, A.Bellaouar, M.Elmasry, *Digital BiCMOS Integrated Circuit Design*, Kluwer Academic Publishers, 1993.
- [EM91] S.Ercolani, G.DeMichelli, *Technology Mapping for Electrically Programmable Gate Arrays*, Proc. of the 28<sup>th</sup> DAC, pp. 234-239, 1991.
- [ESS07] European Silicon Structures, *0.7  $\mu$ m CMOS process and Cell Library Databook*.
- [ESS10] European Silicon Structures, *1.0  $\mu$ m CMOS process and Cell Library Databook*.
- [EUGM95] Notes of the Compass European Users Group Meeting, 1995.
- [Fis92] J.P.Fishburn, *LATTIS: An Iterative Speedup Heuristic for Mapped Logic*, Proc. of the 29th DAC, pp. 488-491, 1992.
- [FM95] M.Favalli, C.Metra, *The effect of glitches on CMOS Buffer Optimization*, Proc. of the PATMOS'95 Conf., pp. 202-212.
- [FMK91] M.Fujita, Y.Matsunaga, T.Kakuda, *On Variable Ordering of Binary Decision Diagrams for the Application of Multi-Level Logic Synthesis*, Proc. of the EDAC, pp. 50-54, 1991.
- [FRV91] R.Francis, J.Rose, Z.Vranesic, *Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGA's*, Proc. of the 28<sup>th</sup> DAC, pp. 227-233, 1991.
- [FYBS93] E.Felt, G.York, R.Brayton, A.Sangiovanni-Vincentelli, *Dynamic Variable Reordering for BDD Minimization*, Proc. of the Euro-DAC, pp. 130-135, 1993.
- [Gaj88] D.Gajski, *Silicon Compilation*, Addison Wesley, 1988.
- [GD85] G.Glasser, D.Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Addison Wesley, 1985.
- [GDKW92] A.Gosh, S.Devadas, K.Keutzer, J.White, *Estimation of Average Switching Activity in Combinational and Sequential Circuits*, Proc. of the 29th DAC, pp. 253-259, 1992.
- [GNU] *g++ Reference Manual*, GNU.

- [Hei86] P.Heinbuch, *3 Micron CMOS Cell Library*, Addison Wesley, 1986.
- [HOIW94] T.Hwang, R.M.Owens, M.J.Irwin, K.H.Wang, *Logic Synthesis for Field-Programmable Gate Arrays*, IEEE Trans. On CAD of Integrated Circuits and Systems, Vol. 13, no. 10, October 1994.
- [HS92] W.Hsu, W.Shen, *Coalgrebraic Division for Multilevel Logic Synthesis*, Proc. of the 29<sup>th</sup> DAC, pp.438-442, 1992.
- [HSA93] M.Hermann, U.Schlitchmann, K.J.Antreich, *Fast Technology Mapping for Multiplexor-based Architectures with Area/Delay Tradeoff*, Proc. of the EURO-ASIC, pp. 300-304, 1993.
- [HXKCH93] X. Hong, T. Xue, E.S. Kuh, C. Cheng, J. Huang, *Performance-Driven Steiner Tree Algorithms for Global Routing*, Proc. Of the 30<sup>th</sup> DAC, pp. 177-181, 1993.
- [IP95] S.Iman, M.Pedram, *Logic Extraction and Factorization for Low Power*, Proc. of the 32nd DAC, pp. 248-253, 1995.
- [Jac93] R.Jacobi, *A Study of the Application of BDDs in Multi-Level Logic Synthesis*, Ph.D.Dissertation, Université Catholique de Louvain, Belgium, 1993.
- [JCT91] R.Jacobi, N.Calazans, Ch.Trullemans, *Incremental Reduction of Binary Decision Diagrams*, Proc. of the ISCAS, pp. 3174-3177, 1991.
- [JPHS92] S.W.Jeong, B.Plessier, G.D.Hachtel, F.Somenzi, *Variable Ordering for Binary Decision Diagrams*, Proc. of the EDAC, pp. 447-451, 1992.
- [Kan86] S.M.Kang, *Accurate Simulation of Power Dissipation in VLSI circuits*, IEEE Journal on Solid State Circuits, vol. SC-21, no. 5, pp. 889-891, 1986.
- [Kar91a] K.Karplus, *Xmap: a Technology Mapper for Table-lookup Field-Programmable Gate Arrays*, Proc. of the 28<sup>th</sup> DAC, pp. 240-243, 1991.
- [Kar91b] K.Karplus, *Amap: a Technology Mapper for Selector-Based Field-Programmable Gate Arrays*, Proc. of the 28<sup>th</sup> DAC, pp. 244-247, 91.
- [KDN92] D.S.Kung, R.F.Damiano, T.A.Nix, *BDDMAP: a technology mapper based on a new covering algorithm*, Proc. of the 29<sup>th</sup> DAC, pp. 484-487, 1992.
- [Keu87] K.Keutzer, *DAGON: Technology Binding and Local Optimization by DAG mapping*, Proc. of the 24<sup>th</sup> DAC, pp. 341-347, 1987.
- [KG85] A.J.Kessler, Ganesan, *Standard Cell VLSI Design: a Tutorial*, IEEE Circuit and Devices, vol. 1, no. 1, pp. 17-34, Janary 1985.



- [KGV83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by Simulated Annealing*, Science, vol. 220, pp. 671-680, May 1983.
- [KNH92] H.Kriplani, F.Najm, I.Hajj, *Maximum Current Estimation in CMOS Circuits*, Proc. of the 29th DAC, pp. 2-7, 1992.
- [KP93] V.Kommu, I.Pomeranz, *GAFFGA: Genetic Algorithm for FPGA Technology Mapping*, Proc. of the EURO-DAC, pp. 300-305, 1993.
- [Kro91] T.H.Krodel, *PowerPlay: Fast Dynamic Power Estimation based on Logic Simulation*, Proc. of the ICCD, pp. 96-100, 1991.
- [Lav94] L.Lavagno, *Modelin, Verification, Synthesis and Testing of Asynchronous Circuits*, Tutorial, UPC, 1996.
- [LB94] W.K.C.Lam, R.K.Brayton, *Timed Boolean functions: a unified formalism for exact timing analysis*, Kluwer Academic Publishers, 1994.
- [LBK88] R.Lisanke, F.Brglez, G.Kedem, *McMap: A Fast Technology Mapping Procedure for Multi-Level Logic Synthesis*, Proc. of the ICCD, pp. 252-256, 1988.
- [LEDA96] S.Naher, C.Uhrig, *The LEDA User Manual, Version R-3.3*, Max-Planck-Institut fur Informatic, Germany, 1996.
- [Lee59] C.Y.Lee, *Representation of switching circuits by binary-decision diagrams*, Bell Systems Technical Journal, pp. 985-999, 1959.
- [Lee61] C.Y.Lee, *An Algorithm for path connections and its applications*, IRE Trans. on Electronic Computers, vol. 10, pp. 346-393, 1961.
- [Lin94] H.R. Lin et al., *Cell Height Driven Transistor Sizing in a Cell Based Module design*, Proc. Of the EDAC, pp. 425-429, 1994.
- [LL92] H.Liaw, Ch.Lin, *On the OBDD-Representation of General Boolean Functions*, IEEE Trans. on Computers, vol. 41, no. 6, pp. 661-664, June 1992.
- [LLS94] J.Y.Lin, T.Ch.Liu, W.Z.Shen, *A Cell-Based Power Estimation in CMOS Combinational Circuits*, Proc. of the ICAD, pp. 304-309, 1994.
- [LMK90] S.Lin, M.Marek-Sadowska, E.S.Kuh, *Delay and Area Optimization in Standard Cell Design*, Proc. of the 27<sup>th</sup> DAC, pp. 349-252, 1990.
- [LMK90] S.Lin, M.Marek-Sadowska, E.S.Kuh, *Delay and Area Optimization in Standard-Cell Design*, Proc. of the 27th DAC, pp. 349-352, 1990.
- [Long] D.E.Long, A Binary Decision Diagram (BDD) Package, long@research.att.com.

- [LPV94] Y.T.Lai, M.Pedram, S.B.K.Vrudhula, *EVBDD-Based Algorithms for Integer Linear Programming, Spectral Transformation, and Functional Decomposition*, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 13, no. 8, pp. 959-975, August 1994.
- [Lsynth91] S.Yang, *Logic Synthesis and Optimization Benchmarks User Guide*, Version 3.0, Technical Report at the MCNC, 1991
- [MAA94] M.Mellah, N.Azemard, D.Auvergne, *Standard Cell Performance Modelling*, Proc. of the PATMOS, pp. 158-167, 1994.
- [Mas91] J.M.Masgonty et al., *Technology and Power Supply Independent Cell Library*, Proc. of the CICC, 1991.
- [MBO195] H.Mehta, M.Borah, R.M.Owens, M.J.Irwin, *Accurate Estimation of Combinational Circuit Activity*, Proc. of the 32nd DAC, pp. 618-622, 95.
- [MBS92] R.Murgai, R.K.Brayton, A.Sangiovanni-Vincentelli, *An Improved Synthesis Algorithm for Multiplexor-based FPGA's*, Proc. of the 29<sup>th</sup> DAC, pp. 380-386, 1992.
- [MC80] C.Mead, L.Conway, *Introduction to VLSI Systems*, Addison Wesley, 1980.
- [MC96] E. Musoll and J. Cortadella, *Optimizing CMOS Circuits for Low Power Using Transistor Reordering*, Proc. Of the ED&TC, pp 219-223, 1996.
- [MDB87] A.M.Martinez, S.Dholokia, S.Bush, *Compilation of Standard Cell Libraries*, IEEE Journal of Solid State Circuits, vol. 22, no. 2, pp. 190-198, April 1987.
- [MFR95] C.Metra, M.Favalli, B.Riccò, *Glicth Power Dissipation Model*, Proc. of the PATMOS'95 Conf., pp. 176-189.
- [MH90] R.L. Maziasz and J.P. Hayes, *Layout Optimization of Static CMOS Functional Cells*, IEEE Trans. CAD, vol. 9, pp 708-719, July 1990.
- [MIE05] Alcatel - Mietec, *0.5  $\mu$ m CMOS process and Cell Library Databook*.
- [Min93] S.Minato, *Zero-Supressed BDDs for Set Manipulation in Combinatorial Problems*, Proc. of the 30th DAC, pp. 272-277, 1993.
- [MIY90] S.Minato, N.Ishiura, S.Yajima, *Shared Binary Decision Diagrams with Attributed Edges for Efficient Boolean Function Manipulation*, Proc. of the 27th DAC, pp. 52-57, 1990.
- [MJH89] C.R.Morrison, R.M.Jacoby, G.D.Hachtel, *TECHMAP: Technology Mapping with Delay and Area Optimization*, Logic and Architecture Synthesis for Silicon Compilation, Elsevier Science Publishers, 1989.

- [MKR92] M.R.Mercer, R.Kapur, D.E.Ross, *Functional Approaches to Generating Orderings for Efficient Symbolic Representations*, Proc. of the 29th DAC, pp. 624-627, 1992.
- [MM90] F.Mailhot, G.DeMicheli, *Technology Mapping using Boolean Matching*, Proc. of the EURO-DAC, pp. 180-185, 1990.
- [MM93] F.Mailhot, G.DeMicheli, *Algorithms for Technology Mapping Based on Binary Decision Diagrams and on Boolean Operations*, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 12, no. 5, pp. 599-620, May 1993.
- [MMP94] R.Marculescu, D.Marculescu, M.Pedram, *Switching Activity Analysis Considering Spatiotemporal Correlations*, Proc. of the 12th ICCAD, pp.294-299, 1994.
- [MMP95] R.Marculescu, D.Marculescu, M.Pedram, *Efficient Power Estimation for Highly Correlated Input Streams*, Proc. of the 32nd DAC, pp. 628-634, 1995.
- [Mor93] F. Moraes et al., *A Transparent Macrocell Layout Methodology*, Proc. Of the PATMOS, pp. 42-54, 1993.
- [MPP95] N. Menezes, S. Pullela, L.T. Pileggi, *Simultaneous Gate and Interconnect Sizing for Circuit-Level Delay Optimization*, 32<sup>nd</sup>. DAC, pp. 690-695, 1995
- [Mus96] E.Mussoll, *High Level and Logic Synthesis Techniques for Low Power*, Ph.D. Dissertation, UPC, 1996.
- [Naj93] F.N.Najm, *Transition Density: A new measure of Activity in Digital Circuits*, IEEE Trans. on CAD, vol. 12, no. 2, pp. 310-323, 1993.
- [Naj94] F.N.Najm, *A Survey of Power Estimation Techniques in VLSI Circuits*, IEEE Trans. on VLSI Systems, vol. 2, no. 4, pp. 446-455, Dec. 1994.
- [Naj95] F.N.Najm, *Feedback, Correlation, and Delay Concerns in the Power Estimation of VLSI Circuits*, Proc. of the 32nd DAC, pp. 612-617, 1995.
- [NBYH88] F.N.Najm, R.Burch, P.Yang, I.Hajj, *CREST - A Current Estimator for CMOS Circuits*, Proc. of the ICCAD, pp. 204-207, 1988.
- [NBYH90] F.N.Najm, R.Burch, P.Yang, I.Hajj, *Probabilistic Simulation for Reliability Analysis of CMOS Circuits*, IEEE Trans. on CAD, vol. 9, no. 4, pp. 439-450, (errata in July), 1990.
- [NS87] A.R.Newton, A.Sangiovanni-Vincentelli, *CAD tools for ASIC Design*, Proc. of the IEEE, vol. 75, no. 6, pp. 765-776, June 1987.

- [NZ95] F.Najm, M.Zhang, *Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits*, Proc. of the 32nd DAC, pp. 623-627, 1995.
- [PB91] M.Pedram, N.Bhat, *Layout Driven Technology Mapping*, Proc. of the 28th DAC, pp. 99-105, 1991.
- [Pel95] D.Pellerin, *Electronic Design Automation for Windows*, Prentice-Hall, 1995.
- [Pes93] R.Peset, *A Performance-Driven Micro-cell Compiler for CMOS Sea-of-Gates Arrays*, Ph.D. Dissertation, University of Twente, 1993.
- [Pes94] R.Peset, *Real Performance-Driven Module Generation*, Personal Communication, UAB, 1994.
- [Pig90] C.Piguet, *Technology Independent Layout Tools*, CMOS VLSI Design, Swiss Federal Institute of Technology, Lausanne, 1990.
- [Pig94] C.Piguet et al., *Low-Power, Low-Voltage Digital CMOS Cell Design*, Proc. of the PATMOS, pp. 132-139, 1994.
- [PP89] M.Pedram, B.Preas, *Interconnection Length Estimation for Optimized Standard Cell Layouts*, Proc. of the ICCAD, pp. 390-393, 1989.
- [PR93] S.Prasad, K.Roy, *Circuit Activity Driven Multilevel Logic Optimization for Low Power Reliable Operation*, Proc. of the EDAC, pp. 368-372, 93.
- [PWH89] H.J.Fleiderer, A.W.Wieder, K.Hart, *BiCMOS for High Performance Analog and Digital Circuits*, Proc. of the ESSIRC, 1989.
- [RC94] J.Riera, J.Carrabina, *Diseño de una biblioteca de Celdas Digitales CMOS en tecnología SOI*, IX DCIS, pp. 214-215, 1994.
- [Rie91] J.Riera, *Biblioteques de cel·les CMOS: Implementació semi-automàtica i independent de la tecnologia a nivell físic*, Treball Experimental, UAB, 1991.
- [RP93] R.Roane, F.Poirot, *Multilevel Logic Synthesis Using Binary Decision Diagrams*, Proc. of the IFIP Workshop on Logic and Architecture Synthesis, pp. 61-72, 1993.
- [RPRCT91] Ll.Ribas, J.M.Pérez, J.Riera, J.Carrabina, Ll.Terés, *Efectos de resistencias i capacitades paràsitas sobre la resposta temporal de los MOSFETS*, VI DCIS, pp. 87-91, 1991.
- [RPRVC95] Ll.Ribas, J.Pérez, J.Riera, A.J.Velasco, J.Carrabina, *Verificación y generación de patrones de test de circuitos jerárquicos a nivel de transistor empleando simulación simbólica*, X DCIS, pp. 285-290, 95.

- [RRAR95] A.Reis, M.Robert, D.Auvergne, R.Reis, *From TBDD Functional Representation to CMOS Complex Gates*, X DCIS, pp. 159-164, 1995.
- [RRCT93] Ll.Ribas, J.Riera, J.Carrabina, Ll.Terés, *Verificación formal de circuitos digitales basada en simulación simbólica*, VIII DCIS, pp. 37-41, 1993.
- [RRPCT91] J.Riera, Ll.Ribas, J.M.Pérez, J.Carrabina, Ll.Terés, *CLIP: software para facilitar el desarrollo de bibliotecas de celdas CMOS*, VI DCIS, pp. 381-386, 1991.
- [RRPCT91a] Ll.Ribas, J.Riera, J.M.Pérez, J.Carrabina, Ll.Terés, *CIRCUS: Circuit Timing Characterization Using SPICE*, VI SBCCI, pp. 51-58, Brasil, 1991.
- [RRPCT91b] J.Riera, Ll.Ribas, J.M.Pérez, J.Carrabina, Ll.Terés, *Layout Generation for CMOS Standard Cell Libraries*, VI SBCCI, pp. 59-76, Brasil, 1991.
- [RRPCT92] Ll.Ribas, J.Riera, J.M.Pérez, J.Carrabina, Ll.Terés, *CLIP: CAD tools for CMOS Standard Cell Library Generation*, IFIP Workshop on Synthesis, Generation, ... pp. 103-107, Grenoble, 1992.
- [RRPSCT92] J.Riera, Ll.Ribas, J.M.Pérez, J.Sáiz, J.Carrabina, Ll.Terés, *Independencia tecnológica en entornos de diseño VLSI*, VII DCIS, pp. 519-520, 1992.
- [RRPSCT92a] Ll.Ribas, J.Riera, J.M.Pérez, J.Saiz, J.Carrabina, Ll.Terés, *Generación automática de vectores para la caracterización eléctrica de circuitos digitales*, VII DCIS, pp. 197-202, 1992.
- [RRPSCT93] Ll.Ribas, J.Riera, J.M.Pérez, J.Saiz, J.Carrabina, Ll.Terés, *CLIP: generación asistida de células de bibliotecas estándar en entornos de diseño VLSI*, *Microelectrónica 92: Tecnologías, diseño, aplicaciones*, ISBN 84-8102-01401, Ed. Publicaciones de la Universidad de Cantabria, pp. 196-198, 1993.
- [RRPSCT93a] Ll.Ribas, J.Riera, J.M.Pérez, J.Saiz, J.Carrabina, Ll.Terés, *Automatic Pattern Generation for the Electrical Characterization of Digital Modules*, *Microprocessing and Microprogramming 39*, pp. 255-258, 1993.
- [RRSPCT93] Ll.Ribas, J.Riera, J.Saiz, J.M.Pérez, J.Carrabina, Ll.Terés, *CirCUS - Automatic Characterization of Digital Cells*, PATMOS, ISBN 3-929814-04-8, pp. 113-125, 1993.
- [RRVC95] J.Riera, Ll.Ribas, A.J.Velasco, J.Carrabina, *Performance Driven Logic Synthesis*, Proc. of the Int. Workshop on Logic and Architecture Synthesis, Grenoble, 1995.

- [RRVC96] J.Riera, Ll.Ribas, A.J.Velasco, J.Carrabina, *Deriving Cost Functions from Cell Libraries and Real IC's to allow Real Area-Power-Delay Trade-off in Early Stages of Logic Synthesis*, Journal of Systems Architecture, 1996 (in press).
- [Rud93] R.Rudell, *Dynamic Variable Ordering for Ordered Binary Decision Diagrams*, Proc. of the IWLS, 1993.
- [RVRVC95] J.Riera, J.Vidal, Ll.Ribas, A.J.Velasco, J.Carrabina, *Derivación de funciones de coste para las primeras etapas de la síntesis lógica*, X DCIS, pp. 240-245, 1995.
- [S1400] European Silicon Structures, *SOLO1400 Reference Manual*.
- [Sah88] C.Sah, *Evolution of MOS Transistor - From Conception to VLSI*, Proc. of the IEEE, vol. 76, no. 10, pp. 1280, 1326, October 1988.
- [SBS94] A.Saldanha, R.K.Brayton, A.Sangiovanni-Vincentelli, *Circuit Structure Relations to Redundancy and Delay*, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 13, no. 7, pp. 875-883, 1994.
- [Sch93] U.Schlitchmann, *Boolean Matching and Disjoint Decomposition for FPGA technology Mapping*, Proc. of the IFIP Workshop on Logic and Architecture Synthesis, 1993.
- [Sch95] P.H.Schneider, *PAPSAS: A Fast Switching Activity Simulator*, Proc. of the PATMOS'95 Conf., pp. 350-360.
- [Ser94] F.Serra, *Noves tecnologies de la informació: la Microelectrònica*, EUIS Sabadell 1994.
- [SGDK92] A.Shen, A.Gosh, S.Devadas, K.Keutzer, *On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks*, Proc. of the ICCAD, pp. 402-407, 1992.
- [Sha49] C.E.Shannon, *The synthesis of two-terminal switching circuits*, Bell Systems Technical Journal, vol. 28, no. 1, pp. 59-98, 1949.
- [She93] N.Shermann, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers, 1993.
- [Shi91] T.Shiple, *Binary Decision Diagram (BDD) Package, Version 2.3*, University of California, Berkeley, 1991.
- [SIS92] E.M.Sentovich, K.J.Singh, L.Lavagno et al., *SIS: A System for Sequential Circuit Synthesis*, Technical Report UCB/ERL M92/41, University of California, Berkeley, 1992.

- [Smi89] M.Smith et al., *Cell Libraries and Assembly Tools for Analog/Digital CMOS and BiCMOS Application Specific Integrated Circuit Design*, IEEE Journal on Solid State Circuits, vol. 34, no. 5, pp. 1419-1431, October 1989.
- [Somenzi] F.Somenzi, *CUDD: CU Decision Diagram Package Release 1.1.0*, Department of Electrical and Computer Engineering, University of Colorado at Boulder, <Fabio@Colorado.EDU>.
- [SPK95] T.Schneider, C.Piguet, V. von Kaenel, *Low-Voltage / Low-Power Parallelized Logic Modules*, Proc. of the PATMOS'95 Conf., pp. 147-160.
- [SRR92] J.Saiz, LL.Ribas, J.Riera, J.M.Pérez, T.Osés, J.Carrabina, Ll.Terés, *Desarrollo de una biblioteca de PADS para la tecnología CMOS del CNM*, VII DCIS, pp. 485-486, 1992.
- [SS95] P.Schneider, U.Schlitchmann, *Decomposition of Boolean Functions for Low-Power based on a new Power Estimation Technique*, Proc. of the Intl. Workshop on Low Power Design, pp. 123-128, 1994.
- [ST93] P.Sawkar, D.Thomas, *Performance Directed Technology Mapping for Look-Up Table Based FPGAs*, Proc. of the 30<sup>th</sup> DAC, pp. 208-212, 1993.
- [Str93] B.Stroustrup, *El C++*. *Lenguaje de Programación. Segunda Edición*, Addison-Wesley/Díaz de Santos, 1993.
- [Sun87] S.Sunter, *Designing a Standard Cell Library*, Proc. of the CICC, pp. 237-240, 1987.
- [SY93] C.Svenson, J.Yuan, *Ultra High Speed CMOS Design*, Elsevier Science, 1993.
- [TAA95] S. Turgis, N. Azemard, D. Auvergne, *Design and Sizing of tapered buffers for minimum power-delay product*, Proc. Of the PATMOS, pp. 74-90, 1995.
- [TAM93] V.Tiwari, P.Ashar, S.Malik, *Technology Mapping for Low Power*, Proc. of the 30th DAC, pp. 74-79, 1993.
- [Ter86] Ll.Terés, *Generadores automáticos de módulos para el diseño de CI's (VLSI)*, Ph.D. Dissertation, UAB, 1986.
- [Tja89] R.Tjarnstrom, *Power Dissipation Estimate by Switch Level Simulation*, Proc. of the ISCAS, pp. 881-884, 1989.
- [TPD93] C.Y.Tsui, M.Pedram, A.Despain, *Technology Decomposition and Mapping Targeting Low Power Dissipation*, Proc. of the 30th DAC, pp. 68-73, 1993.

- [Tsa88] C.Tsareff et al., *An Expert System Approach to Parametrized Module Synthesis*, IEEE Circuits and Devices magazine, pp 28-35, Jan 1988.
- [TVT86] L.Terés, A.Vandemeulebroecke, Ch.Trullemans, *Design Environment for Integrated Circuits Based on Pascal*, 2<sup>nd</sup> Symposio de Electronica des Telecomunicações, Lisboa, 1986.
- [TZ96] A.M.Trullemans, Q.Zhang, *Rapid Gate Matching with Don't Cares*, Proc. of the ED-TC, pp. 407-410, 1996.
- [UC81] T. Uehara, W. vanCleemput, *Optimal Layout of CMOS Functional Arrays*, IEEE Trans. On Computers, vol. 30, n<sup>o</sup> 5, pp 305-312, May 1981.
- [VMRPC96] A.J.Velasco, X. Marín, J.Riera, R.Peset, J. Carrabina, *Performance Driven Layout Synthesis :Optimal Pairing & Chaining*, Proc. of the 5<sup>th</sup> ACM/SIGDA Physical Design Workshop, pp. 176-182, 1996.
- [VRRPC96] A.J.Velasco, L.Ribas, J.Riera, R.Peset, J. Carrabina, *An Efficient Algorithm for Optimal Pairing and Chaining in layout Generation*, Proc. of the ISCAS, pp. 775-779, 1996.
- [WCA94] Q.Wu, C.Y.R.Chen, J.M.Acken, *Efficient Boolean Matching Algorithms for Cell Libraries*, Proc. of the ICCD, pp. 36-39, 1994.
- [WE93] N.Weste, K.Eshraghian, *Principles of CMOS VLSI Design*, Addison Wesley, 1<sup>st</sup> Edition 1985, 2<sup>nd</sup> Edition 1993.
- [WH95] K.Wang, T.Hwang, *Boolean Matching for Incompletely Specified Functions*, Proc. of the 32nd DAC, pp. 48-53, 1995.
- [Wil90] G.R.Wilson, *Advances in Bipolar VLSI*, Proc. of the IEEE, vol. 78, no. 11, pp. 1707-1719, November 1990.
- [XN94] M.Xakellis, F.N.Najm, *Statistical Estimation of the Switching Activity in Digital Circuits*, Proc. of the 31st DAC, pp. 18-23, 1994.
- [YK89] G.Y.Yacoub, W.H.Ku, *An Accurate Simulation Technique for Short-Circuit Power Dissipation based on Current Component Isolation*, Proc. of the ISCAS, pp. 1157-1161, 1989.
- [ZT94] Q.Zhang, A.M.Trullemans, *Controlling Value Analysis and Observation Function Deduction for Rapid Matching in Technology Mapping*, Proc. of the IFIP Workshop on Logic and Architecture Synthesis, 1994.



---

# Apèndix A

---

## Glossari de Tecnicismes i Traduccions

*Aquest Apèndix conté una taula, ordenada alfabèticament, dels tecnicismes anglesos més corrents relacionats amb el tema de la present tesi; per cadascun d'ells es dona la traducció cap al català escollida, així com una breu descripció del seu significat.*

<i>Tecnicisme</i>	<i>Traducció i Descripció</i>
<b>Branch &amp; Bound</b>	<i>Bifurcació amb acotació</i> Algoritmes d'optimització que en cada pas bifurquen sobre totes les possibles solucions. Algunes d'aquestes solucions poden ser eliminades gràcies a una funció d'acotació del cost d'arribar a la solució òptima a partir d'un punt donat.
<b>Cluster</b>	<i>Grup, agrupació o agrupament</i> Grup de nodes de la Xarxa Booleana que seran implementats en una única porta complexa.
<b>Clustering</b>	<i>Procés d'Agrupament</i> Procés de creació de tots els possibles grups de nodes de nodes de la Xarxa Booleana que poden ser implementats en portes complexes.
<b>Cover</b>	<i>Coberta</i> Coberta d'una funció (en suma de productes) o també coberta d'una Xarxa Booleana (circuit que la implementa).
<b>Covering</b>	<i>Cobertura</i> Cobertura d'una Xarxa Booleana per un circuit que la implementi i minimitzi una certa funció de cost.
<b>Cube-free</b>	<i>No cub-factoritzable</i> Una expressió és no cub-factoritzable si no pot ser dividida per un cub en dues subexpressions (amb resta nul·la).
<b>DC-set</b>	<i>Conjunt d'indiferents</i> Conjunt de minterms pels quals no importa el valor de la funció.
<b>Fanin</b>	<i>Ventall d'entrada</i> Conjunt d'entrades de les quals depèn un node de la Xarxa Booleana.
<b>Fanout</b>	<i>Ventall de sortida</i> Conjunt de nodes de la Xarxa Booleana que depenen d'un node determinat.

<b>Glitch</b>	<i>Pols espuri</i> Pols lògicament innecessari que es produeix a la sortida d'una porta abans que aquesta arribi al seu estat estable.
<b>Greedy algorithm</b>	<i>Algoritme àvid</i> Algoritme d'optimització en el qual en cada pas la funció de cost es minimitza localment en base a la informació disponible.
<b>Kernel</b>	<i>Nucli</i> Subexpressió, divisor no cub-factoritzable d'una funció Booleana.
<b>Matching</b>	<i>Aparellament</i> Operació que consisteix en comprovar si una funció pot ser implementada per una cel·la determinada, considerant possibles permutacions i/o inversions de les entrades i/o de la sortida.
<b>Off-set</b>	<i>Conjunt passiu, conjunt negatiu</i> Conjunt de mínterms pels quals la funció val zero.
<b>On-set</b>	<i>Conjunt actiu, conjunt positiu</i> Conjunt de mínterms que satisfan la funció.
<b>Over-The-Cell Routing</b>	<i>Connexionat damunt les cel·les</i> Es diu del procés de connexionat que utilitza un número tal de nivells de metall que permet efectuar connexions damunt les cel·les del circuit, i no únicament en canals reservats al connexionat.

---

# Apèndix B

---

## Biblioteca LIB2\_AOI modificada

*Llistat de la biblioteca Lib2\_AOI modificada per incloure les funcions de cost proposades en el Capítol 5.*

GATE inv1x 2 O = ! a;  
PIN a INV 0 0 2.5 0 1 0

GATE xor 12 O = ((!a \* b) + (a \* !b));  
PIN \* UNKNOWN 0 0 6 0 4.5 0

GATE xnor 12 O = ((!a \* !b) + (a \* b));  
PIN \* UNKNOWN 0 0 6 0 4.5 0

GATE nand2 4 O = ! (a \* b);  
PIN \* INV 0 0 2.5 0 2 0

GATE nand3 6 O = ! (a \* b \* c);  
PIN \* INV 0 0 2.5 0 3 0

GATE nand4 8 O = ! (a \* b \* c \* d);  
PIN \* INV 0 0 2.5 0 4 0

GATE nor2 4 O = ! (a + b);  
PIN \* INV 0 0 5 0 1 0

GATE nor3 6 O = ! (a + b + c);  
PIN \* INV 0 0 7.5 0 1 0

GATE nor4 8 O = ! (a + b + c + d);  
PIN \* INV 0 0 10 0 1 0

GATE aoi21 6 O = ! ((a1 \* a2) + b);  
PIN \* INV 0 0 5 0 2 0

GATE aoi31 8 O = !((a1 \* a2 \* a3) + b);  
 PIN \* INV 0 0 5 0 3 0

GATE aoi22 8 O = !((a1 \* a2) + (b1 \* b2));  
 PIN \* INV 0 0 5 0 2 0

GATE aoi32 10 O = !((a1 \* a2 \* a3) + (b1 \* b2));  
 PIN \* INV 0 0 5 0 3 0

GATE aoi33 12 O = !((a1 \* a2 \* a3) + (b1 \* b2 \* b3));  
 PIN \* INV 0 0 5 0 3 0

GATE aoi211 8 O = !((a1 \* a2) + b + c);  
 PIN \* INV 0 0 7.5 0 2 0

GATE aoi221 10 O = !((a1 \* a2) + (b1 \* b2) + c);  
 PIN \* INV 0 0 7.5 0 2 0

GATE aoi222 12 O = !((a1 \* a2) + (b1 \* b2) + (c1 \* c2));  
 PIN \* INV 0 0 7.5 0 2 0

GATE oai21 6 O = !((a1 + a2) \* b);  
 PIN \* INV 0 0 5 0 2 0

GATE oai31 8 O = !((a1 + a2 + a3) \* b);  
 PIN \* INV 0 0 7.5 0 2 0

GATE oai22 8 O = !((a1 + a2) \* (b1 + b2));  
 PIN \* INV 0 0 5 0 2 0

GATE oai32 10 O = !((a1 + a2 + a3) \* (b1 + b2));  
 PIN \* INV 0 0 7.5 0 2 0

GATE oai33 12 O = !((a1 + a2 + a3) \* (b1 + b2 + b3));  
 PIN \* INV 0 0 7.5 0 2 0

GATE oai211 8 O = !((a1 + a2) \* b \* c);  
 PIN \* INV 0 0 5 0 3 0

GATE oai221 10 O = !((a1 + a2) \* (b1 + b2) \* c);  
 PIN \* INV 0 0 5 0 3 0

GATE oai222 12 O = !((a1 + a2) \* (b1 + b2) \* (c1 + c2));  
 PIN \* INV 0 0 5 0 3 0

GATE zero 0 O=CONST0;  
 GATE one 0 O=CONST1;

---

# Apèndix C

---

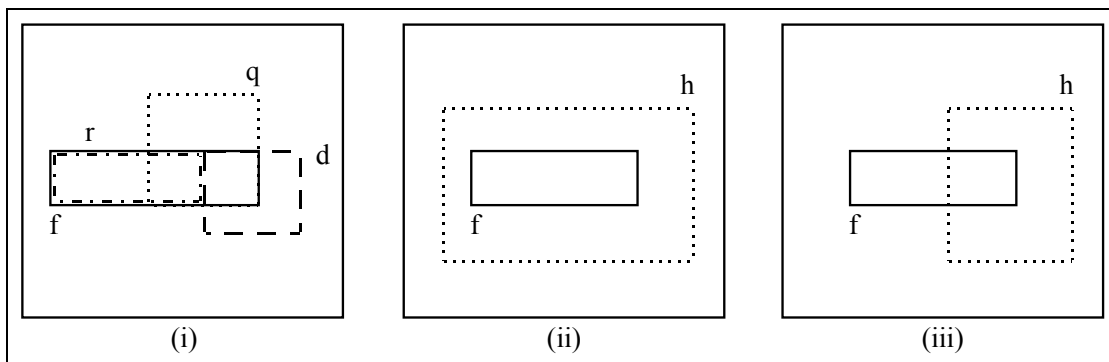
## Divisió Booleana/algebraica i Factorització

*L'operació de divisió Booleana/algebraica és definida. Aquesta operació serà utilitzada per a la factorització de funcions.*

**Definició:** Siguin  $f$  i  $d$  dues funcions Booleanes. La *divisió lògica* de  $f$  per  $d$  consisteix en trobar dues funcions  $q$  i  $r$  tal que:

$$f = qd + r$$

on  $q$  és no nul i s'anomena el quocient, i  $r$  la resta, de la divisió. Sovint s'utilitza la notació  $f/d$  per denotar el quocient  $q$ . La funció  $d$  s'anomenarà un divisor de  $f$  si  $r$  és una funció no nul·la o un factor de  $f$  si  $r$  és nul·la. Si  $q$  i  $d$  tenen suports disjunts, l'operació es coneix amb el nom de *divisió algebraica*; *divisió Booleana* en cas contrari.



**Figura C-1** Representació gràfica de l'operació de divisió lògica (i).  $h$  és un factor de  $f$  (ii).  $h$  és un divisor de  $f$  (iii).

És clar que l'operació de divisió no és única. És més, depèn fins i tot de la representació de  $f$  i  $d$ . El número de divisors o factors d'una funció determinada pot ser molt elevat, ja que tota funció Booleana  $h$  que compleixi que  $f \subseteq h$  o, el que és igual,  $f \bar{h} = \emptyset$ , serà un factor de  $f$ . I  $h$  serà un divisor de  $f$  si compleix que  $f h \neq \emptyset$ . La Figura C-1 mostra una representació gràfica de les situacions descrites. Tota funció  $h$

que contingui a  $f$  serà un factor de  $f$ , i tota funció  $h$  que tingui com a mínim un mínim en comú amb  $f$  serà un divisor de  $f$ . És clar que el conjunt dels divisors de  $f$  engloba també el conjunt dels factors de  $f$ .

*funció factorització(f)*  
*Si  $|f| \leq 1$  retorna  $f$*   
 1)  $d = \text{escullDivisor}(f)$   
 2)  $(q,r) = \text{divisió}(f,d)$   
*retorna  $\text{factorització}(q)\text{factorització}(d)+\text{factorització}(r)$*

**Algoritme C-1** Factorització d'una expressió  $f$ .

L'operació de divisió de funcions permet implementar fàcilment un algoritme de factorització tal com es veu en l'Algoritme C-1. L'eficàcia de l'algoritme dependrà dels mètodes usats per escollir un bon divisor de l'expressió (línia 1) i per realitzar l'operació de divisió (línia 2).

Brayton i altres autors [BM82, Bra87, BHS90] han desenvolupat algoritmes eficients per al càlcul de divisors algebraics d'una expressió  $F$ . Els algoritmes es basen en la generació del que els autors anomenen *nuclis* i *co-nuclis* d'una expressió algebraica. Totes les operacions involucrades en el càlcul dels nuclis i co-nuclis són operacions algebraiques.

Sigui  $f$  una funció Booleana amb suport  $X$ , i  $F$  una coberta de  $f$ . Sigui  $x_i$  una variable del suport. Sigui  $b_i$  la fase d'un literal; 0 indicarà fase negativa, 1 positiva. Sigui  $x_i^{b_i}$  un literal associat a la variable  $x_i$  amb fase  $b_i$ .

**Definició:** Direm que una expressió  $F$  és *no cub-factoritzable* si no existeix cap cub que la divideixi en dues subexpressions (sense resta):

$$F \text{ e's no cub factoritzable} \Leftrightarrow \nexists D \in F \mid F = QD$$

Una expressió no cub-factoritzable ha de tenir dos cubs com a mínim. L'expressió  $ab+cd$  és no cub-factoritzable, mentre que  $ab+ac$  és factoritzable, ja que pot ser dividida pel cub  $a$ .

**Definició:** Els conjunt dels *divisors primaris* d'una expressió algebraica  $F$  està format per les expressions:

$$\mathcal{D}(F) = \{F/c \mid c \text{ e's un cub}\}$$

**Definició:** Els conjunt dels *nuclis* d'una expressió  $F$  està format per les expressions:

$$\mathcal{X}(F) = \{G \mid G \in \mathcal{D}(F), G \text{ e's no cub factoritzable}\}$$

**Definició:** Els *co-nuclis* d'una expressió són els quocients de l'expressió respecte dels nuclis de l'expressió, és a dir, són els cubs que originen els nuclis.

Els nuclis es poden classificar per nivells, que s'identifiquen a la profunditat de l'expressió en forma factoritzada que els representa. Formalment, un nucli és de nivell zero si no conté cap altre nucli excepte ell mateix. Un nucli serà de nivell  $n$  si conté com a mínim un nucli de nivell  $n-1$ , però cap nucli de nivell  $n$  o superior.

Nucli	co-Nucli	Nivell
$a+b$	$de, df$	0
$e+f$	$ad, bd$	0
$(a+b)d+g$	$f$	1
$(a+b)(e+f)$	$d$	1
$((a+b)d)(e+f)+fg$	$1$	2

**Taula C-1** Nuclis i co-nuclis de l'expressió  $F=ade+adf+bde+ddf+fg$ .

En la Taula C-1 es poden veure els nuclis i co-nuclis associats a l'expressió  $F=ade+adf+bde+ddf+fg$ . Tal com es pot veure en la taula, un nucli pot tenir més d'un co-nucli. També cal destacar que un possible co-nucli pot ser el cub trivial  $1$ , si l'expressió original ja era no cub-factoritzable.

El càlcul de tots els nuclis d'una expressió algebraica  $F$  pot ser una tasca difícil que implica subtileses, però els nuclis de nivell zero (i alguns de nivell superior) poden ser calculats eficientment mitjançant la intersecció dels conjunts de dos o més cubs de  $F$  [BM82]. Aquests nuclis poden ser utilitzats com a divisors de  $F$  en l'Algoritme C-1 (línia 1) per obtenir una factorització de  $F$ .

*funció* factorització( $F$ )

Si  $|f| \leq 1$  retorna  $f$

$D = \text{escullDivisor}(F)$

Si  $D=0$  fer retorna  $F$

$Q = \text{divisió}(F, D)$

1) Si  $|Q| > 1$  fer  $D = \text{ferNoCubFactoritzable}(Q)$

2) sinó  $D = \text{millorLiteral}(F, Q)$

3)  $(Q, R) = \text{divisió}(F, D)$

retorna  $(Q)(D)+(R)$

**Algoritme C-2** Algoritme de factorització que garanteix una forma màximament factoritzada per  $F$  si les operacions involucrades són algebraiques.

La factorització generada per l'Algoritme C-1, però, no garanteix l'obtenció d'una forma màximament factoritzada, ja que pot existir un factor comú entre el quocient i la resta de la divisió. Aquest fet es té en compte en l'Algoritme C-2 [BRSW87]. L'algoritme obté en primer lloc un divisor candidat, que pot ser un nucli de nivell zero o superior, i calcula el quocient de l'expressió respecte del divisor. Donat que es factoritzarà  $F$  usant  $Q$  i  $D$ , l'algoritme intenta recollir tot el que estigui multiplicat per  $Q$  en  $F$ . Si  $Q$  conté més d'un cub, s'eliminen els literals que puguin ser factors de  $Q$  i, en cas contrari, s'escull el literal de  $Q$  que més apareix en  $F$  (línia 2). Aquest nou divisor s'utilitza per generar un nou quocient (línia 3), que com a mínim inclourà el vell quocient  $Q$ . Aquesta segona divisió és especialment efectiva quan l'elecció del divisor inicial no és prou bona.

La utilització d'operacions algebraiques en l'Algoritme C-2 garanteix la generació d'una forma màximament factoritzada. La divisió algebraica, que intervé també en el procés de generació de nuclis de nivell zero, necessita d'un ordenament lexicogràfic dels literals en els cubs de l'expressió  $F$  per tal que la implementació de l'algoritme sigui eficient. En *synthetic*, cada cub es representa en un BDD, i una suma de productes es redueix a una llista de BDDs. L'ordenació global de les variables dels BDDs garanteix l'ordenament lexicogràfic requerit per les operacions involucrades en la factorització, pel que la implementació dels algoritmes de factorització proposats en [BM82, Bra87, BHS90] resulta eficient.



---

# Apèndix D

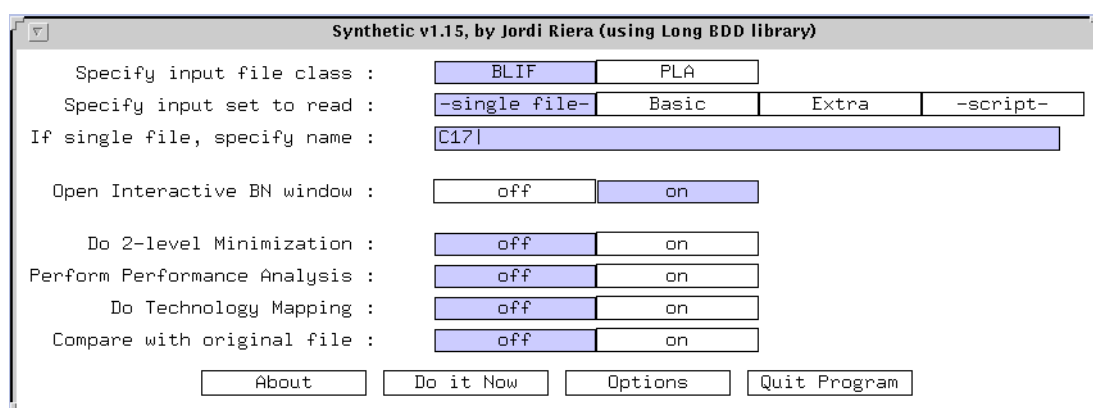
---

## Synthetic

*En aquest apèndix es mostrarà breument el funcionament de l'eina de síntesi lògica orientada a generadors de mòduls synthetic, on s'han implementat els algoritmes presentats en aquest treball.*

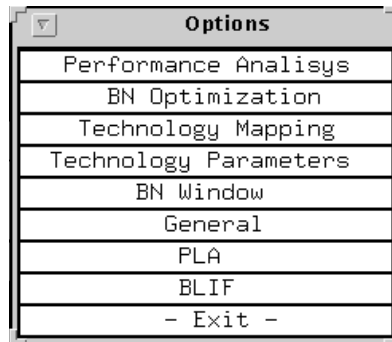
L'aplicació *synthetic* ha estat desenvolupada en llenguatge C++ [Str93], usant el compilador de domini públic de GNU [GNU], basant-se en la llibreria de funcions LEDA [LEDA96], gratuïta per l'ambient universitari, i les rutines d'encapsulació en C++ per diferents paquets de BDDs [Cap96].

L'aplicació *synthetic* ha estat dissenyada per mostrar l'efectivitat de la síntesi lògica guiada per prestacions i orientada a generadors de mòduls i també per ser utilitzada en ambients universitaris per a l'ensenyament. De moment, cobreix l'etapa de mapatge tecnològic i d'estimació de prestacions, però en un futur inclourà també l'etapa d'optimització de la Xarxa Booleana.



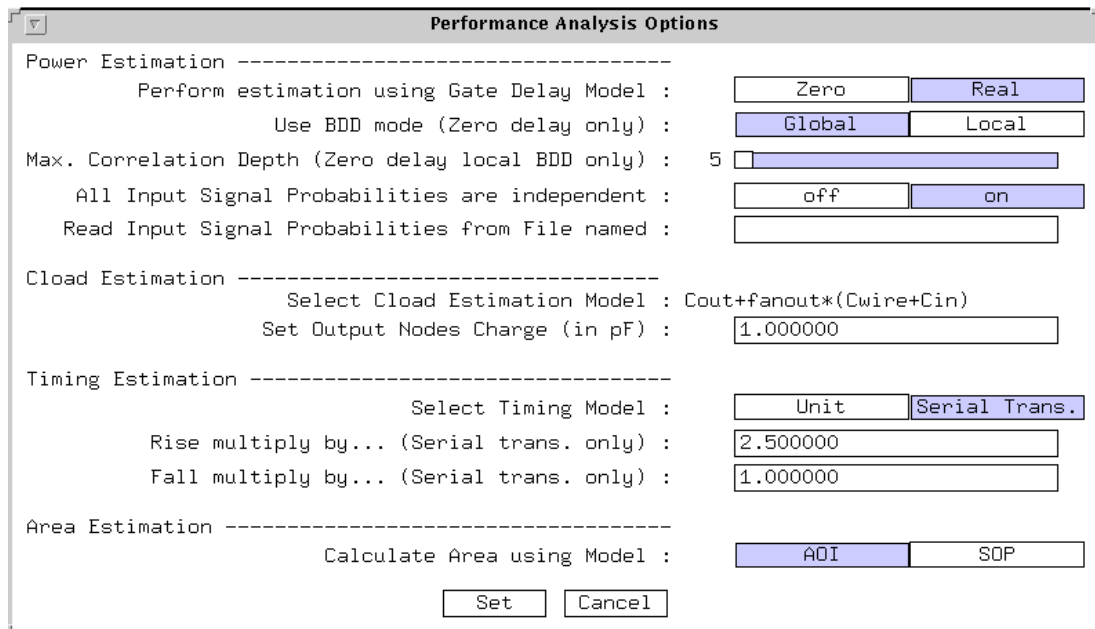
**Figura D-1** Menú principal de l'aplicació synthetic.

La Figura D-1 mostra l'aspecte del menú interactiu de l'aplicació, encara que les comandes també estan disponibles com a paràmetres en la crida a *synthetic*. Els circuits poden venir descrits en un format a dos nivells (PLA) o multi-nivell (BLIF). Aquests formats es troben especificats en [Lsynth91].



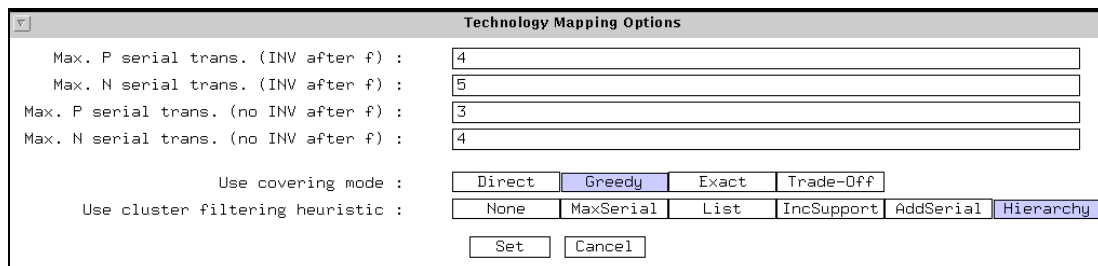
**Figura D-2** Menú d'opcions.

El funcionament de *synthetic* pot ser controlat a través de diferents opcions. El menú d'opcions es pot veure en la Figura D-2. Diferents opcions controlen el tipus d'estimacions realitzades, la definició de la tecnologia, el tipus de mapatge tecnològic, etc...



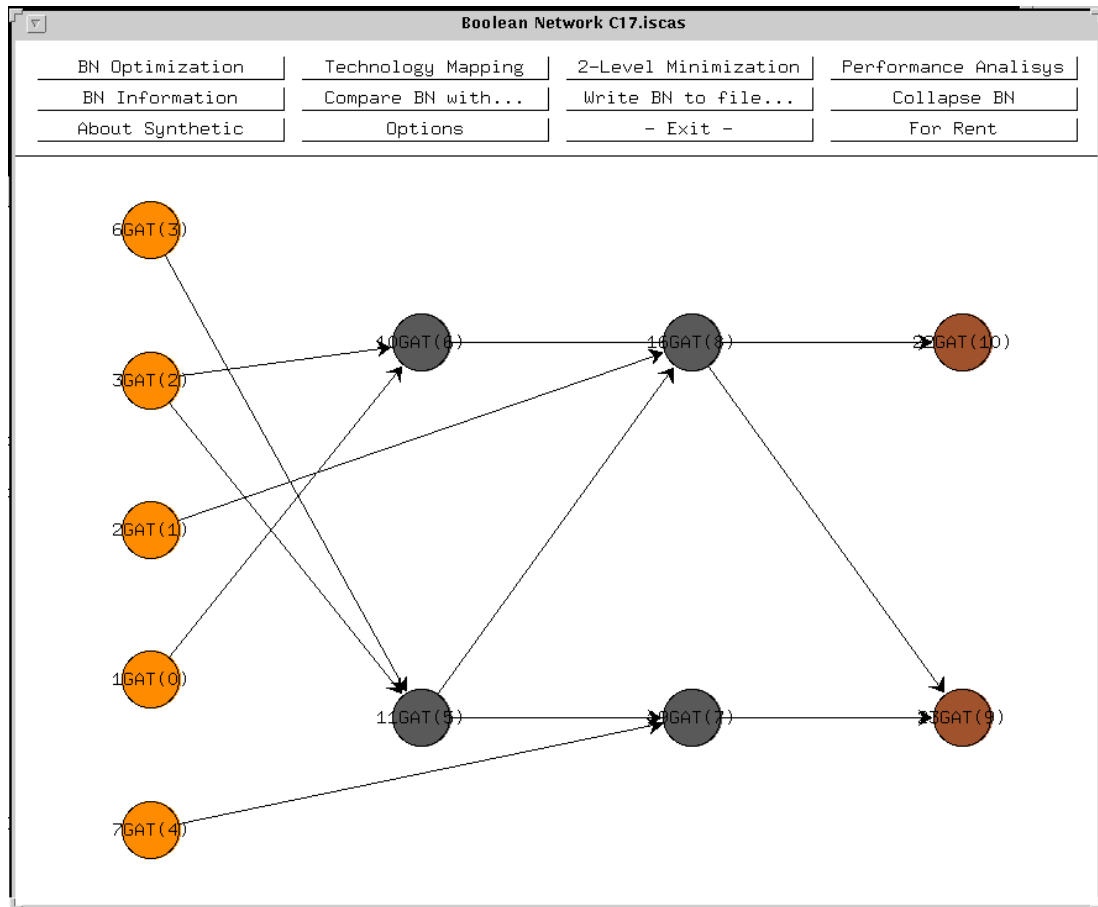
**Figura D-3** Menú d'opcions per a l'estimació d'àrea, retard i dissipació de potència.

Les opcions per a l'estimació dels diferents objectius de disseny es pot veure en la Figura D-3.



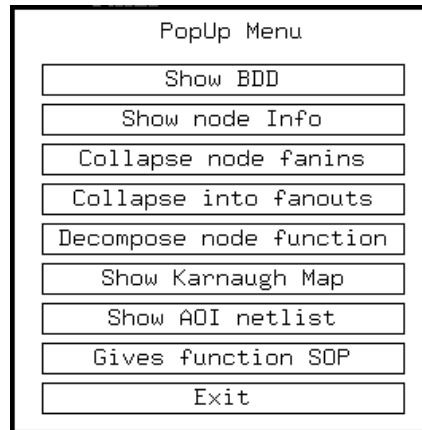
**Figura D-4** Menú d'opcions que controlen el mapatge tecnològic.

La Figura D-4 mostra les opcions que controlen el mapatge tecnològic. Es defineixen els límits imposats per la tecnologia en el número màxim de transistors en sèrie permesos en una porta CMOS d'una sola etapa. També es pot escollir el tipus de cobertura de la Xarxa Booleana i el tipus d'heurística de filtrat d'agrupaments.



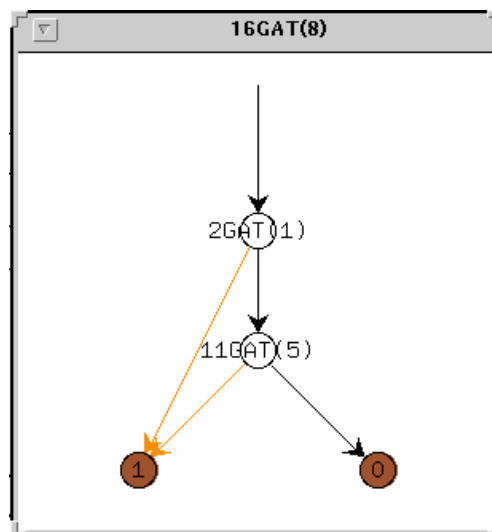
**Figura D-5** Finestra de manipulació interactiva d'una Xarxa Booleana.

En la Figura D-5 es mostra la finestra de manipulació interactiva d'una Xarxa Booleana. Diferents operacions són possibles, entre elles l'avaluació de prestacions i el mapatge tecnològic. Cada node de la Xarxa Booleana conté una funció del seu ventall d'entrades. Aquesta funció està representada internament per un BDD, però pot ser mostrada a l'usuari de diferents formes.



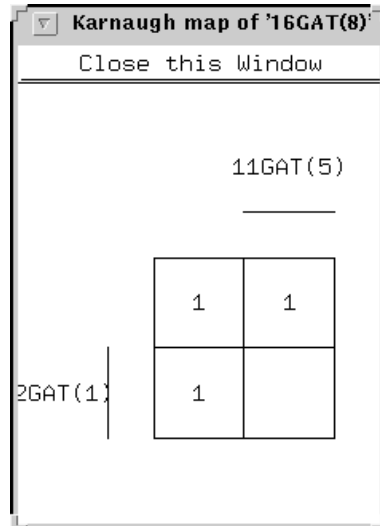
**Figura D-6** Menú d'operacions sobre nodes de la xarxa.

Diferents operacions poden ser realitzades sobre els nodes de la Xarxa Booleana, a nivell individual. La Figura D-6 mostra el menú corresponent. Les funcions que es poden realitzar sobre el node inclouen el fet de mostrar diferents representacions de la funció que conté el node, però també operacions que modifiquen la Xarxa Booleana, com son la descomposició de la funció o el fet de col·lapsar els nodes del ventall d'entrada, o col·lapsar el node actual sobre el seu ventall de sortida.



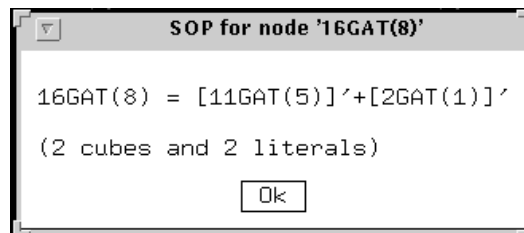
**Figura D-7** BDD corresponent a un node de la xarxa.

En la figura anterior es mostra el BDD corresponent a la funció emmagatzemada en un node de la Xarxa Booleana. Aquest BDD també es pot mostrar usant els arcs complementaris, i/o dins de l'espai global de BDDs que representa el circuit. Diferents opcions de *synthetic* permeten seleccionar com es mostraran els BDDs.



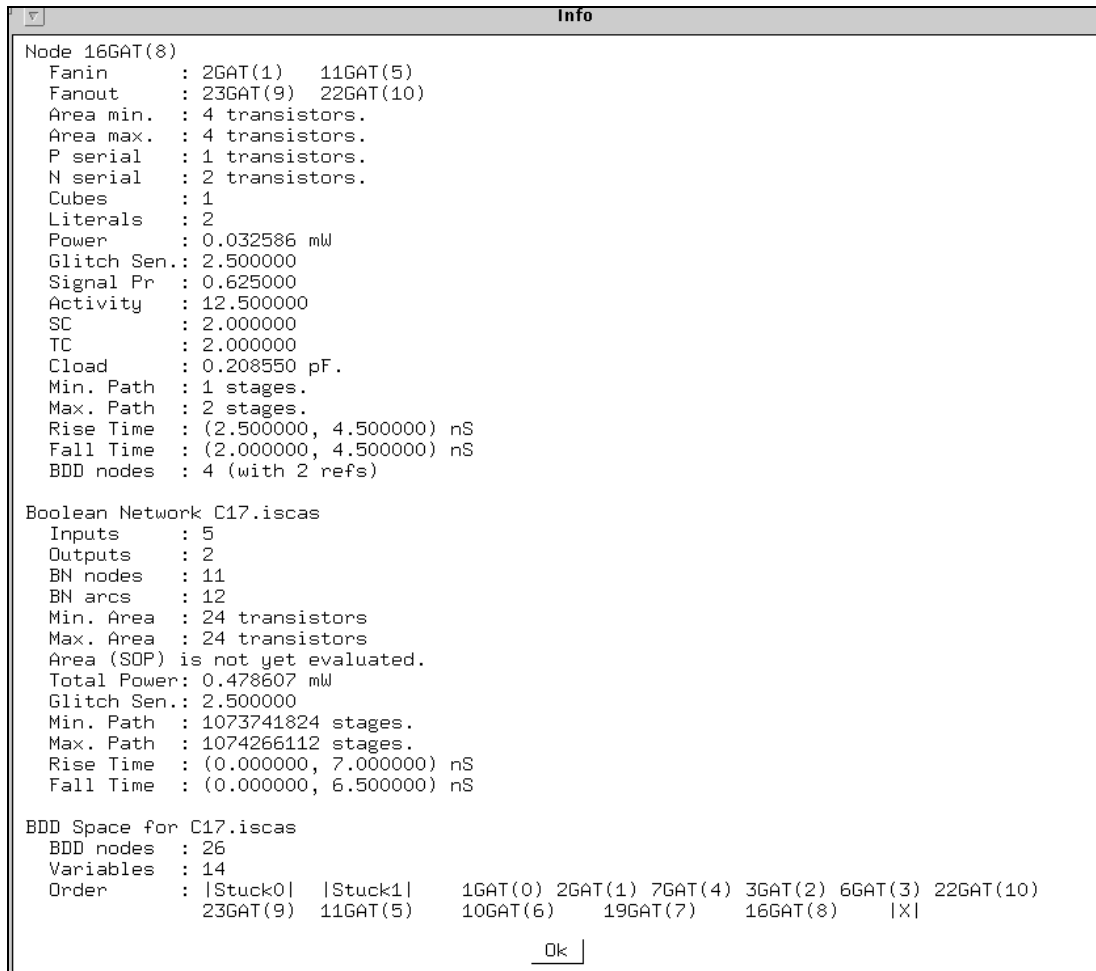
**Figura D-8** Mapa de Karnaugh corresponent a la funció d'un node de la xarxa.

La funció corresponent al mateix node pot ser mostrada també mitjançant el seu mapa de Karnaugh, sempre i quan el número de variables d'entrada no sigui superior a sis. La Figura D-8 en mostra un exemple.



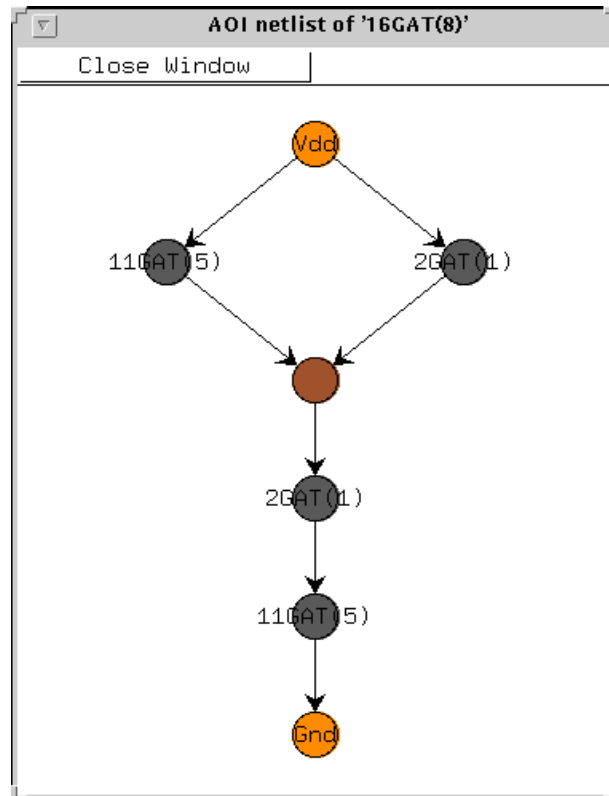
**Figura D-9** Suma de productes corresponent a un node de la xarxa.

Finalment, les funcions dels nodes de la Xarxa Booleana també poden ser mostrats en forma de suma de productes, tal com es pot veure en la Figura D-9.



**Figura D-10** Informació disponible sobre la Xarxa Booleana i sobre un node concret.

Sobre cada node de la Xarxa Booleana, i sobre aquesta globalment, se'n pot obtenir informació detallada si s'ha fet una avaluació de prestacions sobre el circuit. En la Figura D-10 es pot veure la informació extreta, que inclou informació sobre els diferents objectius de disseny i informació general sobre els BDDs.



*Figura D-11* Porta CMOS d'una sola etapa corresponent a la implementació de la funció d'un node de la Xarxa Booleana.

Per acabar, el mapatge tecnològic del circuit produirà una Xarxa Booleana on cada node té associada una porta CMOS d'una sola etapa. Aquesta porta pot ser visualitzada, tal com es pot veure en la Figura D-11.