

# L'EDIF, UN FORMAT ESTÀNDAR PER A L'INTERCANVI DE DADES DE DISSENY ELECTRÒNICS

SALVADOR MIR, XAVIER MIR, JOSEP ANTONI RUBIO

Universitat Politècnica de Catalunya

*En els darrers anys, el ràpid increment dels sistemes CAE/CAD en el camp de l'electrònica ha fet que la transferència de informació d'uns sistemes a uns altres fos un problema creixent. En conseqüència, s'ha fet necessària una estandarització dels llenguatges de disseny i dels formats d'intercanvi de dades. L'EDIF (Electronic Design Interchange Format) neix per respondre a aquesta necessitat d'estandarització.*

*Aquest article descriu la implementació d'un reconeixedor del format EDIF i de traduccions entre l'EDIF i el llenguatge LUCIE de descripció de les màscares d'un circuit integrat. Amb aquestes traduccions es pretén mostrar una metodologia per a la construcció de traductors entre l'EDIF i d'altres eines.*

**EDIF: A standar format for the interchange of electronic design data**

**Keywords:** Electronic Design Interchange Format. EDIF. Lexica', syntactic and semantic analyzers. Mask layout description languages. CIF. LUCIE. Compiler generators. YACC. LEX.

## 1. INTRODUCCIÓ

Les aplicacions CAE/CAD involucren processos de disseny a diferents nivells: disseny estructural, verificació del disseny utilitzant simulacions lògiques o a nivell de circuit, disseny del layout (placement i routing), testabilitat,... . Alguns formats d'intercanvi de dades i llenguatges de descripció del hardware ja

---

—Salvador Mir, Xavier Mir, Josep Antoni Rubio - Universitat Politècnica de Catalunya - Departament d'Enginyeria Electrònica de l'E.T.S.E.I.B. - Av. Diagonal, 647 - Barcelona.  
—Article rebut el maig de 1987.

existents es refereixen a un d'aquests aspectes concrets del disseny electrònic. D'altres són propietat d'una companyia de CAD o de semiconductors, limitant-se molt la seva utilització. En general, són difícils de implementar degut a l'evolució del format o llenguatge, o bé són difícils d'extendre per a desenvolupar els nous mètodes de disseny electrònic que van apareixent.

L'EDIF va ser concebut com un format no executable per a l'intercanvi de dades de disseny de circuits integrats, abarçant els diferents aspectes que tenen lloc en el disseny i superant les limitacions anteriors. L'interés en el seu desenvolupament ha estat portat per una sèrie de companyies americanes vinculades a CAD i al disseny de circuits integrats: Motorola Inc., Daisy Systems Corp., Mentor Graphics Corp., National Semiconductor Corp., Texas Instruments Inc. i Tektronix Inc.

A l'esser l'EDIF un llenguatge no executable, sino simplement un format d'intercanvi de dades, cal una traducció a eines especialitzades en el tractament d'un determinat aspecte del disseny.

Com a primer pas, s'ha de implementar un reconeixedor del format EDIF que permeti aplicar la informació rebuda en aquest format a eines ja existents no estandaritzades però dedicades eficaçment a un aspecte concret del disseny d'un circuit integrat. La implementació d'aquest reconeixedor implica la construcció d'un analitzador lèxic, un analitzador sintàctic i un analitzador semàntic pel format EDIF guardant-se la informació en un format intern que permeti, en un segon pas, que aquesta pugui ser utilitzada per una traducció específica.

## **2. EL FORMAT EDIF**

El desenvolupament de l'EDIF s'inicià a finals de 1983 i és avui un format en evolució, ampliant-se la seva capacitat de descripció en alguns dels nivells en els quals pot treballar. Malgrat això, les ampliacions de l'EDIF no modifiquen bàsicament la seva estructura apareguda en versions anteriors. Aquest article es refereix a la versió 1.1.0, que apareix el desembre de 1985.

L'EDIF deriva del llenguatge LISP i d'aquest ha heretat les seves característiques més representatives pel que fa a la sintaxi. Consisteix en ítems bàsics o àtoms tals com strings, constants, enters, ... i en llistes que contenen elements que són àtoms o altres llistes. Aquestes llistes estan delimitades per parèntesi i sempre comencen amb una paraula reservada que la defineix i que li dóna sentit. El format tipus LISP és relativament senzill de ser analitzat i l'ús de les paraules reservades significa que és senzill d'extendre l'EDIF per incorporar noves construccions.

L'especificació admet tres nivells de complexitat:

- . Nivell 0: nivell bàsic on sol s'admeten constants.
- . Nivell 1: extensió de l'anterior a l'afegir variables i expressions.

- . Nivell 2: extensió de l'anterior amb procediments i tipus de dades complexes.

La versió actual de l'EDIF 1 1 0 defineix completament el nivell 0 i el nivell 1, i inclueix una indicació de les extensions que tindrà el nivell 2.

Les descripcions utilitzant l'EDIF poden arribar a ser molt llargues. Per tal de reduir el tamany d'aquestes es preveu en el format la capacitat de definir noves keywords en termes de les ja existents. Això compleix un paper semblant a un preprocessat. Obviament, el sistema receptor de la descripció haurà d'expandir les noves keywords en termes de les "reals" de l'EDIF per a realitzar el processat.

Actualment, hem treballat amb el nivell 0 de la versió 1 1 0 de l'EDIF sense desenvolupar la fase de preprocessat. Una mirada a les construccions del nivell 1 permet veure que l'expansió a aquest nivell ha d'ésser fàcil, però no adequada fins a l'aparició de la nova versió 2 0 0.

### **3. DESCRIPCIONS UTILITZANT EDIF**

Les eines de CAD en el camp electrònic cobreixen un ampli rang d'aspectes de disseny incloent la descripció estructural, simulació, especificació de màscares, representacions esquemàtiques,... L'EDIF reflexa aquest rang al permetre recollir diferents representacions o vistes d'un disseny en una única descripció. Existeixen actualment set vistes diferents: MASKLAYOUT, NETLIST, DOCUMENT, BEHAVIOR, SCHEMATIC, SYMBOLIC i STRANGER. Es poden interpretar les vistes com una manera particular de veure una cel.la del disseny. Per exemple, la vista MASKLAYOUT pot contenir figures geomètriques que representen les màscares; una vista NETLIST s'encara a la connectivitat entre cel.lles; una vista BEHAVIOR especifica el comportament lògic; una vista DOCUMENT presenta una descripció textual del disseny; la vista STRANGER s'utilitza per informacions que no segueixen cap de les convencions de les altres vistes, resultant com un calaix de sastre.

Una descripció en EDIF està construïda d'una manera jeràrquica, començant pels nivells més abstractes i concretant-se a mesura que es descendeix en la jerarquia. L'estructura de l'EDIF és cel.lular, constituint cada cel.la una unitat dins del disseny. Cada cel.la podrà tenir diferents vistes, especificant aquestes diferents aspectes de la cel.la.

El nivell més alt del disseny consisteix en l'especificació d'una cel.la arrel, origen del disseny, a partir de la qual es relacionen d'altres cel.lles que formen part de la descripció EDIF. En una mateixa descripció EDIF poden existir diverses cel.lles arrel donant lloc a dissenys diferents.

Les cel.lles que tenen característiques comunes, com el tipus de tecnologia, s'agrupen en llibreries. Dins de cada vista d'una cel.la normalment s'especifica

una secció INTERFACE, que defineix la comunicació entre la cel.la i les cel.les que referencia, i una secció CONTENTS que conté la implementació detallada segons la perspectiva de la vista que es tracta.

La figura 1 il.lustra part de la definició d'un disseny representat esquemàticament en la figura 2.

```
(EDIF exemple
  (STATUS (EDIFVERSION 1 1 0)(EDIFLEVEL 0)
  (WRITEN (TIMESTAMP 1987 05 14 18 00 00)))
  (DESIGN exemple (QUALIFY cellLib exemple))
  (COMMENT "cel.la arrel del disseny")
  (LIBRARY cellLib
  (TECHNOLOGY nmosMC
  (NUMBERDEFINITION SI (SCALE DISTANCE 100 (E 1-6)))
  (DEFINE OUTPUT FIGUREGROUP (MULTIPLE NP NM NC))
  (FIGUREGROUPDEFAULT
  (FIGUREGROUPSPECIFICATION NP
  (PATHTYPE TRUNCATE TRUNCATE)
  (WIDTH 400)
  (COLOR 100 0 0))
  (FIGUREGROUPSPECIFICATION NM
  (PATHTYPE TRUNCATE TRUNCATE)
  (WIDTH 600)
  (COLOR 100 0 0))
  (FIGUREGROUPSPECIFICATION NC
  (PATHTYPE TRUNCATE TRUNCATE)
  (WIDTH 0)
  (COLOR 0 0 0 ))))
  (CELL MPContact
  (VIEW MASKLAYOUT Physical
  (CONTENTS
  (FIGUREGROUP NP (RECTANGLE
  (POINT -400 -400) (POINT 400 400)) )
  (FIGUREGROUP NM (RECTANGLE
  (POINT -600 -600) (POINT 600 600)))
  (FIGUREGROUP NC (RECTANGLE
  (POINT -200 -200) (POINT 200 200))) ) ) )
  (CELL exemple
  (VIEW MASKLAYOUT Physical
  (CONTENTS
  (INSTANCE MPContact Physical MPC1
  (TRANSFORM (TRANSLATE -1000 1000)))
  (INSTANCE MPContact Physical MPC2
  (TRANSFORM (TRANSLATE 2000 -2000)))
  (FIGUREGROUP NM (PATH
  (POINT 2000 - 2000) (POINT 2000 1000)
  (POINT -1000 1000))))))
```

FIGURA 1. Descripció EDIF de dos contactes NMOS.

L'exemple dona una descripció parcial d'un DESIGN anomenat "exemple" que té com a cel.la arrel la CELL "exemple" que està en la LIBRARY "cellLib". D'aquesta cel.la s'especifica la VIEW MASKLAYOUT, anomenada "Physical" en la qual no s'indica la secció INTERFACE i si la secció CONTENTS que defineix el layout. La cel.la utilitza la vista "Physical" de la cel.la "MPContact" que és referenciada en dues ocasions mitjançant el mecanisme formal de INSTANCE. Aquestes instàncies s'anomenen "MPC1" i "MPC2". També s'utilitza un conjunt de figures FIGUREGROUP anomenat "NM" que estableixen el camí PATH d'unió entre les cel.les referenciades tal i com es veu en la figura 2. Dintre de la llibreria s'especifica una secció TECHNOLOGY, en aquest cas "nmosMC", que conté informació relativa a les cel.les de la llibreria. En particular, FIGUREGROUPs que defineixen les màscares que són declarades OUTOUT, significat que són màscares per ser fabricades. Cada FIGUREGROUP es especifica en la secció FIGUREGROUPSPECIFICATION definint les característiques d'ella.

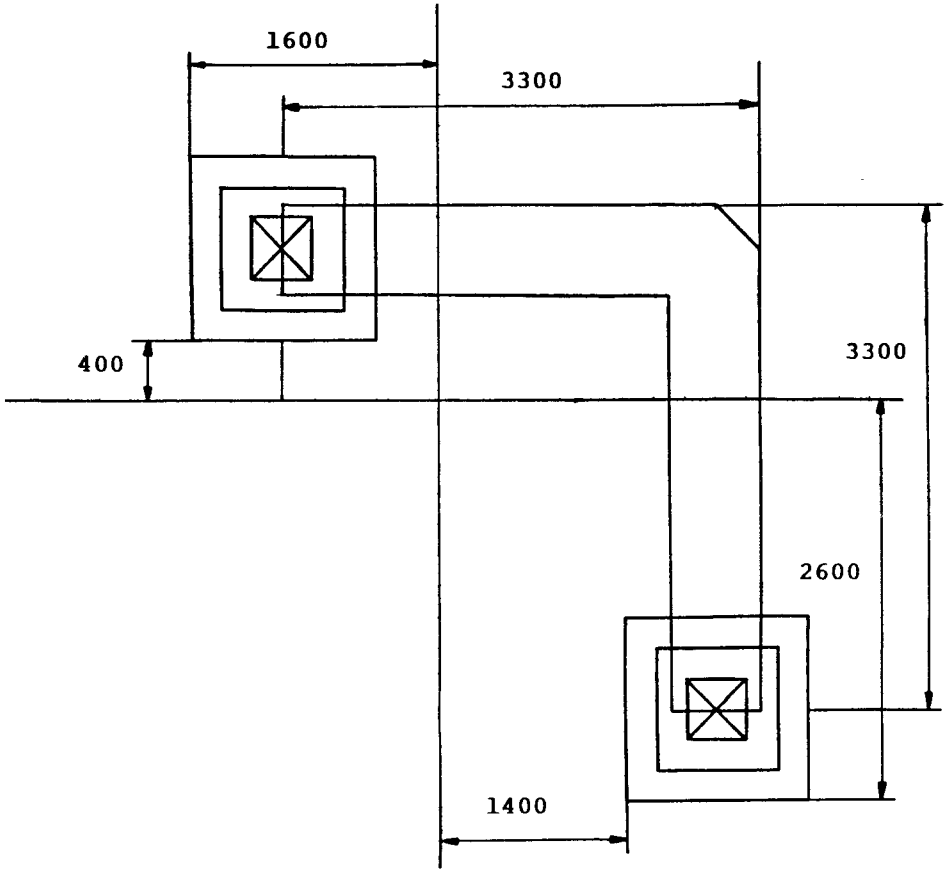


FIGURA 2

FIGURA 2. Layout de dos contactes NMOS.

#### 4. EL RECONEXEDOR D'EDIF

L'EDIF és un format en evolució i per tant el reconeixedor s'ha de dissenyar de la manera més flexible possible, de mode que ens permeti introduir les variacions del llenguatge que poden aparèixer sense modificar substancialment el que ja està fet.

Existeixen llenguatges dissenyats expressament per a l'escriptura de compiladors anomenats "compiladors de compiladors". L'aparició d'aquests generadors és deguda a que algunes fases de la compilació estan altament mecanitzades a l'estar molt formalitzada part de la definició dels llenguatges.

La utilització d'aquests generadors permet flexibilitzar molt la implementació del reconeixedor d'EDIF. S'han emprat dos generadors, LEX (Lexical Analyzer Generator) i YACC (Yet Another Compiler Compiler), per a desenvolupar els analitzadors lèxic i sintàctic. La figura 3 il·lustra un esquema general del reconeixedor.

El reconeixedor, preparat per a generar un codi diferent depenent de les eines a les quals es vol aplicar la informació EDIF, treballa en dues fases consecutives: mentre la primera involucra l'anàlisi lèxic, sintàctic i semàntic de les dades EDIF, la segona correspon a la generació de codi.

Durant la primera fase, o fase d'anàlisi, la informació EDIF és guardada en una estructura interna de dades. Aquesta estructura de dades reflexa la jerarquia d'una descripció en EDIF, anant-se detallant progressivament en els nivells més baixos. La segona fase, o fase de síntesi, utilitza aquesta estructura per a generar un codi objecte. Cada analitzador del reconeixedor disposa d'una opció de DEBUGGER per a purgar els errors que puguin aparèixer en el reconeixedor.

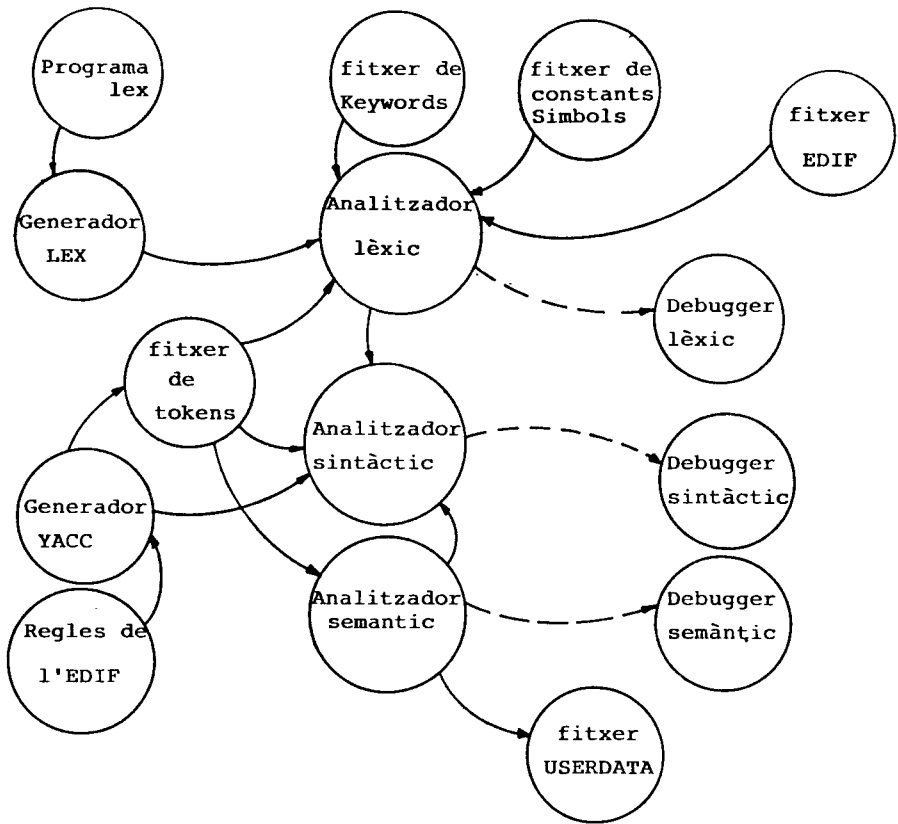


FIGURA 3

FIGURA 3. Estructura global del reconeixador d'EDIF.



#### 4.1. GENERADORS LEX I YACC

LEX genera analitzadors lèxics en llenguatge C i es troba disponible sobre UNIX. Els programes en LEX reconeixen expressions regulars o patrons. YACC accepta la definició sintàctica d'un llenguatge expressada en BNF (Backus Naur Form) i genera un analitzador sintàctic en llenguatge C i sobre UNIX, al igual que el LEX.

Per a generar un analitzador sintàctic el YACC accepta en general gramàtiques del tipus LALR(1), requerint un analitzador a baix nivell per a reconèixer els tokens d'entrada. LEX està especialment dissenyat per a simplificar la interfície amb YACC i una combinació d'ambdós resulta apropiada. La figura 4 il·lustra com treballen ambdós generadors.

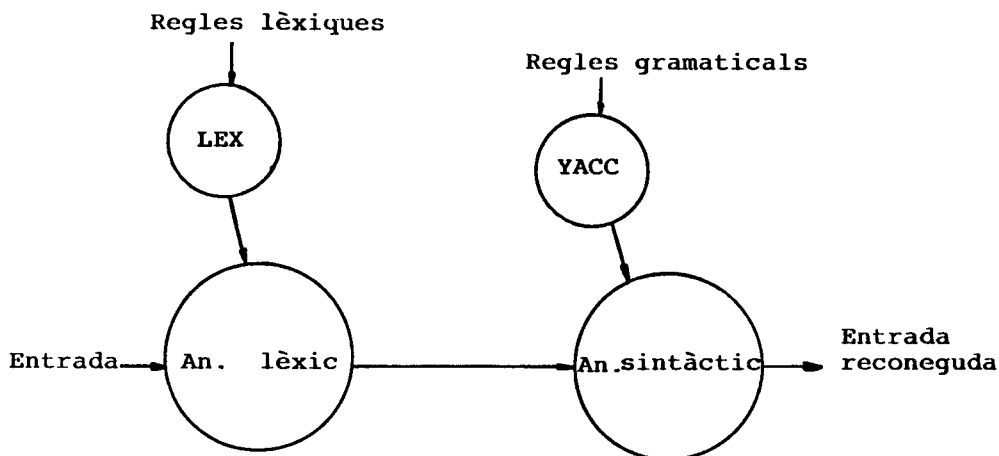


FIGURA 4

FIGURA 4. LEX amb YACC.

#### 4.2. ANALITZADOR LÈXIC

L'analitzador lèxic és la part més senzilla d'un traductor. Va explorant els caràcters del programa font d'esquerra a dreta i construeix els símbols del programa (enters, identificadors, paraules reservades, strings, delimitadors, ...) que s'anomenen tokens. Aquests símbols es van passant a l'analitzador sintàctic o parser amb un format intern: cada token es representarà per un nombre sencer. Això permetrà a la resta del traductor treballar d'una manera eficient.

L'analitzador lèxic utilitza dos fitxers on s'introdueixen les keywords i les constants simbòliques del format. Això permet flexibilitzar l'analitzador, de manera que si s'introdueixen nous tokens en aquest solament cal afegir-los als fitxers corresponents.

#### 4.3. ANALITZADOR SINTÀCTIC (PARSER) I SEMÀNTIC

Descomposen el programa font en les seves parts constitutives, realitzant una comprovació sintàctica i semàntica del programa, construint la forma interna del programa i posant informació en la taula de símbols i d'altres.

Normalment, i aquest és el cas del reconeixador d'EDIF, l'anàlisi semàntic va controlat per l'anàlisi sintàctic. Quan l'analitzador sintàctic reconeix una determinada construcció del programa font, crida una rutina semàntica que verifica si és correcta semànticament i emmagatzema la informació en la taula de símbols o la forma interna del programa.

El YACC genera l'analitzador sintàctic a partir d'un fitxer que conté la gramàtica de l'EDIF conjuntament amb accions que s'han d'invocar després de reconèixer una determinada construcció. Futures ampliacions de la gramàtica s'han d'introduir en aquest fitxer i generar de nou l'analitzador.

La presència d'errors en el programa font pot fer que l'anàlisi del programa es descontrolï o s'aturi. El problema està doncs en contruir un parser suficientment robust davant dels errors, de manera que aquest s'assumeixi com a tal i l'anàlisi pugui continuar normalment. En el nostre cas, la recuperació d'errors s'ha d'adequar al generador del parser.

L'anàlisi semàntic i la generació de codi requereixen conèixer els atributs dels identificadors del codi font. La taula de símbols és el lloc central on es guarda tota la informació associada amb aquells noms. EDIF, encara que no sigui executable, s'assembla a un llenguatge d'alt nivell amb estructura de blocs. Així, la taula de símbols s'ha organitzat com un stack amb accés per hashing per tal d'accelerar les cerques.

Dins de l'EDIF, es defineix la construcció especial USERDATA per tal de poder transmetre dades no compatibles amb l'EDIF. Aquesta construcció no pot ser, per tant, validada pel reconeixedor, i s'emmagatzema en un fitxer separat que pot ser utilitzat pel sistema CAD de recepció de la informació.

#### 4.4. ESTRUCTURA INTERNA DE DADES

La informació continguda dins del programa EDIF es va recollint i emmagatzemant en una estructura interna durant tot el procés d'anàlisi. L'estructura de dades interna està construïda amb arbres binaris de forma que reflexa la jerarquia de la informació que conté una descripció en EDIF.

Aquesta estructura és la que després s'emprarà per tal d'obtenir la traducció de la informació d'un aspecte concret de l'EDIF o vista a un llenguatge específic i per tant especialitzat en una feina concreta.

## 5. EL FORMAT EDIF I EL SISTEMA LUCIE

L'EDIF és un format d'intercanvi de dades i no un llenguatge executable. Per tant, per utilitzar la seva informació cal fer una traducció a un sistema especialitzat en un aspecte del disseny. En aquest article es comenta un traductor EDIFaLUCIE i l'invers LUCIEaEDIF, construïts amb una sistemàtica que permet ser utilitzada per a la construcció de d'altres traductors entre l'EDIF i una eina concreta.

El llenguatge LUCIE ("Language Universitaire de Conception de Circuits Intégrés pour l'Enseignement") és un llenguatge de descripció de les màscares dels circuits integrats desenvolupat en l'institut IMAG de Grenoble. Forma part d'un sistema, sistema LUCIE, que a més del llenguatge disposa d'unes eines de manipulació de la informació LUCIE: un traductor, que tradueix el llenguatge LUCIE a una forma intermitja que interpreten les altres eines; un editor gràfic, que permet la representació gràfica de tota o una part del circuit sobre una pantalla gràfica Tektronix (així com certes operacions de correcció i extracció dels paràmetres del disseny), i la seva sortida per una impressora Benson; i finalment un digitalitzador que permet l'entrada del disseny a través d'una tauleta digitalitzadora Tektronix que retornarà una descripció en llenguatge LUCIE.

L'EDIF permet també transmetre informació del disseny de les màscares utilitzant la vista MASKLAYOUT de les cel·les. La descripció en LUCIE pot ser traduïda a EDIF i inversament. La figura 5 il·lustra la descripció LUCIE corresponent a la descripció EDIF de la figura 1.

```

NIV MP, MM, MC
* arrel de la descripció del disseny
FIG exemple
    *declaració d'un contacte
    FIG MPContakt
        REC (200,200,1000,1000,MP)
        REC (0,0,1200,1200,MM)
        REC (400,400,800,800,MC)
    FFIG
    *generació dels contactes
    FIG MPContakt (3000,0)
    FIG MPContakt (0,3000)
    *generació del cami
    REC (3300,600,600,3300)
    REC (600,3300,3300,600)
FFIG

```

FIGURA 5. Descripció LUCIE de dos contactes NMOS.

El llenguatge LUCIE té una estructura imbricada de les figures, a diferència de l'EDIF que presenta una estructura cel.lular.

La traducció de la informació MASKLAYOUT d'un disseny escrit en EDIF a LUCIE s'obté a partir de l'estructura interna de dades creada pel reconeixedor d'EDIF. El traductor comença a processar pel bloc de CONTENTS de la vista MASKLAYOUT de la cel.la DESIGN extraient la informació de totes les FIGUREGROUP, processant cada INSTANCE i expansionant així la jerarquia del disseny.

Aquest pas d'EDIFaLUCIE permet llavors la visualització en un plotter o pantalla gràfica de la geometria de les cel.les.

La figura 6 esquematitza el traductor EDIFaLUCIE. Aquest pas invers és també atractiu: el llenguatge LUCIE permet treballar interactivament a l'hora de desenvolupar el disseny i és per tant una eina adequada pel disseny de les màscares. Llavors, després del disseny gràfic s'obtenen fitxers escrits en llenguatge LUCIE de les característiques del disseny que podran ser traduïts a EDIF.

Per a la implementació del traductor de LUCIE a EDIF es precisa també d'uns analitzadors lèxic i sintàctic, anàlogament al reconeixedor d'EDIF. Malgrat que el llenguatge LUCIE sigui particularment molt senzill, s'han obtingut aquests analitzadors utilitzant els generadors LEX i YACC per tal de presen-

tar una sistemàtica a l'hora de construir traductors de l'EDIF a d'altres eines que tractin la informació de d'altres vistes (simuladors elèctriques per la vista NETLIST, lògics per la vista BEHAVIOR, ...).

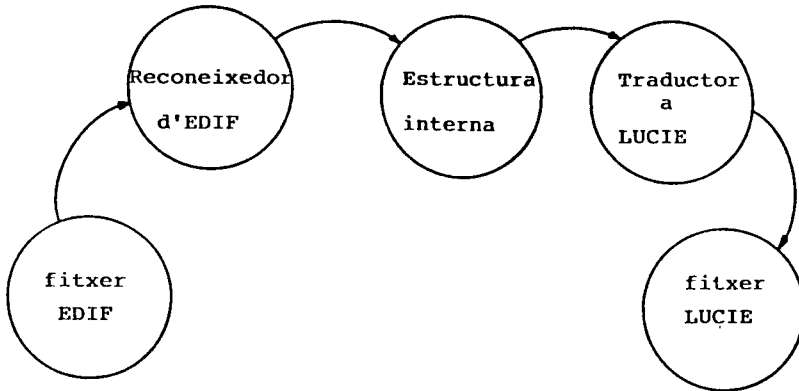


FIGURA 6

FIGURA 6. Estructura global del traductor LUCIEaEDIF.

## 6. CONCLUSIÓ

El format EDIF es presenta com un format estàndard que està essent adoptat tant per indústries com per centres universitaris. Es constata l'existència de llenguatges de descripció de hardware que descriuen dissenys electrònics a diversos nivells i que estan essent utilitzats amb molta generalitat pels grups de disseny dels centres de recerca d'ensenyament de circuits electrònics. En particular, per les descripcions a nivell de màscara de circuits integrats, en el Departament d'Enginyeria Electrònica de la Universitat Politècnica de Catalunya s'utilitzen, entre d'altres, els formats LUCIE i CIF.

En aquest treball s'ha presentat un reconeixedor de l'EDIF on part dels seus elements s'han obtingut utilitzant els generadors LEX i YACC. També s'ha presentat un traductor d'EDIF a LUCIE i inversament. Actualment, el reconeixedor soporta la versió 1 1 0 i nivell 0 sense desenvolupar la fase del preprocessat. Un examen del nivell 1 de l'EDIF permet veure que l'expansió a aquest nivell ha d'ésser fàcil, però no viable fins l'aparició de la nova versió 2 0 0. Totes les eines han estat escrites en llenguatge C corrent sota un entorn VAX/VMS o UNIX.

## REFERÈNCIES

- [1] **EDIF Steering Committee.** EDIF Specification: Electronic Design Interchange Format Version 1 1 0. 1985.
- [2] **Equipe de recherche d'Architecture des Ordinateurs du Laboratoire IMAG de Grenoble.** LUCIE: Language Univesitaire de Conception de Circuits Integrés pour l'Enseignement. 1985.
- [3] **S.C. Johnson.** YACC: Yet Another Compiler Compiler. UNIX Programmer's Manual. 1978.
- [4] **M.E. Lesk and E. Schmidt.** LEX: A Lexical Analyzer Generator UNIX Programmer's Manual. 1978.
- [5] **A.T. Schreiner, H.G. Friedman, Jr.** Introduction to Compiler Construction with UNIX. Prentice-Hall, 1985.
- [6] **Xavier Mir, Salvador Mir, J. Rubio and J. Figueras.** TRANEDIF: "A flexible EDIF translator and its application to the EDIF Mask layout view". European EDIF Forum. Brusseles, 30 setembre 1987.