

論理回路図用

エンジニアリング・データベースの構築

宇田川 佳久 (Yosihisa Udagawa)

溝口 徹夫 (Tetsuo Mizoguchi)

(三菱電機・情報電子研究所)

1. はじめに

設計図面を再利用することができたなら、設計作業の飛躍的な効率向上が期待できる。設計図面の管理は、エンジニアリング・データベースの主要な機能になると考えられる。設計図面データベースを構築する有力な方法に、汎用データベースを用いる方法がある。とりわけ、関係データモデルは、応用プログラムとの独立性が高い、使い易いなどの理由から注目されている [6]。本稿で述べる ADAM は、図形・線画像情報及び抽象化機能を強化した関係データベースである。

一般に設計図面は複数のリレーションによって表現されているために、図面を編集するとき、複数のリレーションにまたがるデータ操作が必要となる。従来の関係データベース操作言語は、リレーションに対する演算をいかにわかり易く表現するかということを主眼に開発されたものである [4]。従って、これらの言語を図面編集のたびに”プログラム”することは、図面作成者にとって繁雑すぎるものである。そこで、我々は図面作成者に対して、設計対象単位の操作を提供する一方で、ADAM に対するデータ操作を行なうエディタ EVE を開発した。図 1 は ADAM & EVE の構成図である。

以下では、標準 IC を用いた論理回路図の編集作業と、それに伴って必要となるデータ操作及びその処理時間を中心に報告する。

2. 回路図の階層と構造表現

標準 IC は、ゲートやフリップ・フロップなどの基本回路を組合せて構成されて

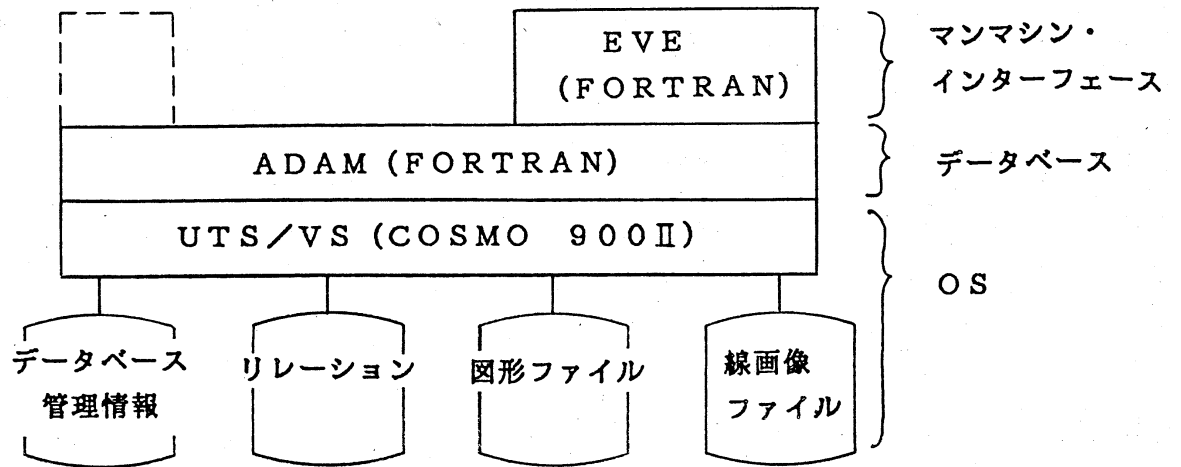


図1. ADAM&EVEの構成図.

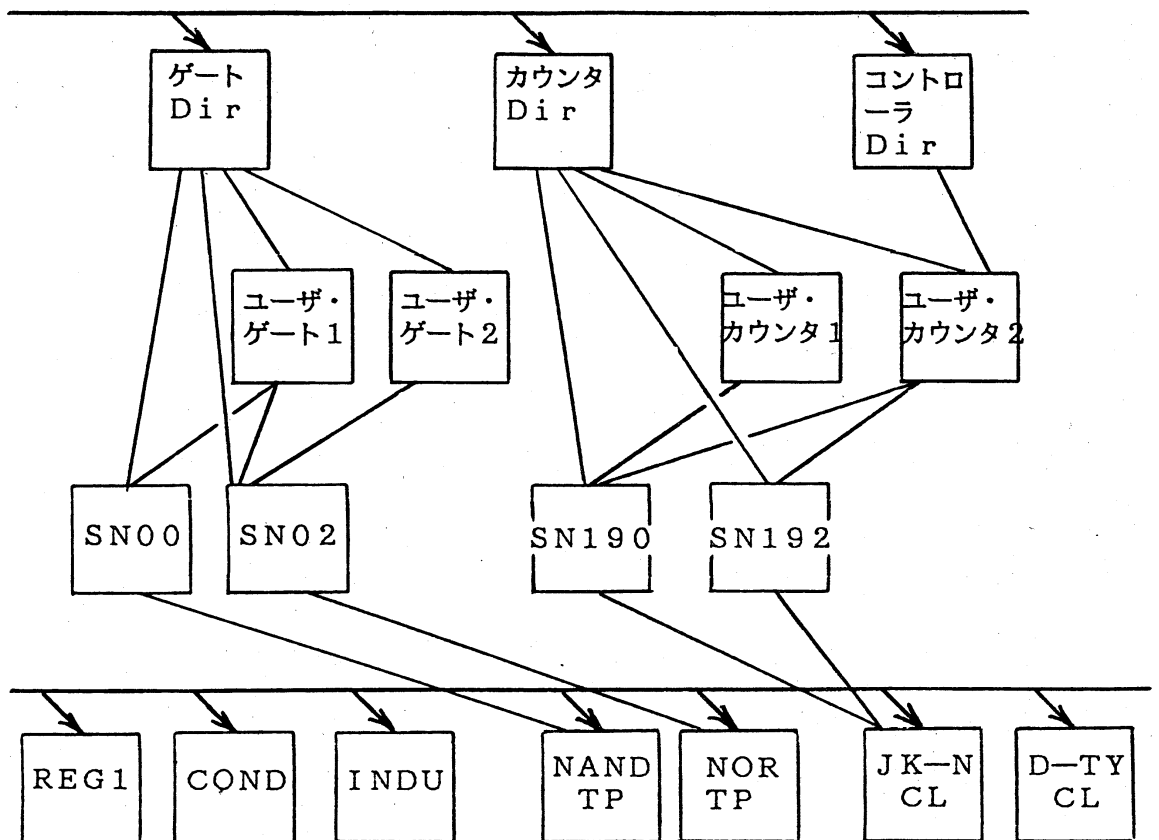


図2. 回路図の階層

いる。さらに、ユーザが作成する回路図は、基本回路と標準IC、あるいはすでにユーザが作成した図面を引用しながら作られる。このような、回路図の参照関係に基づいて回路図の階層を定義することが自然である。図2は、今回実現した回路図データベースの階層を図示したものである。図中の正方形は、回路図に相当するもので、ADAMでは引数付き実現値によって識別される単位である。同図の最下層は、基本回路のレベルである。

標準ICのレベルは基本回路よりも1階層上に位置付けられている。基本回路及び標準IC図は、いわばADAM&EVEの初期データであり、データベース・エンジニアによって作成され、ユーザに提供される。標準IC図より1階層上が、ユーザが作成した回路図の階層である。ユーザが作成した回路図と標準IC図は同じスキーマで表現されている。標準IC図とユーザが作成した回路図は、ゲート/ドライバ、演算器、レジスタなど約10種類に分類できる[3]。図2の最上位の階層は、回路図の分類に対応するもので、各分類に属する回路図の識別名を管理している（この階層の各エントリはリレーションしか含まれていない）。回路図を特徴付ける属性集合は、回路図の分類によって異なる。ADAM&EVEでは、分類ごとにエントリ（リレーション）を用意することによって属性集合の違いを吸収している。

データベース・スキーマを設計するためには、設計図面に対する操作、とりわけ回路図の妥当性チェックを考慮する必要がある。以下、妥当性チェックを念頭においてスキーマを設計する。

TTL-IC回路設計で要望される妥当性チェックの主なものは、

- (A) 未接続端点の指摘、
- (B) 未接続入力端点を含む部分回路の指摘、
- (C) 不正接続の指摘、
- (D) ファンアウト数のチェック、

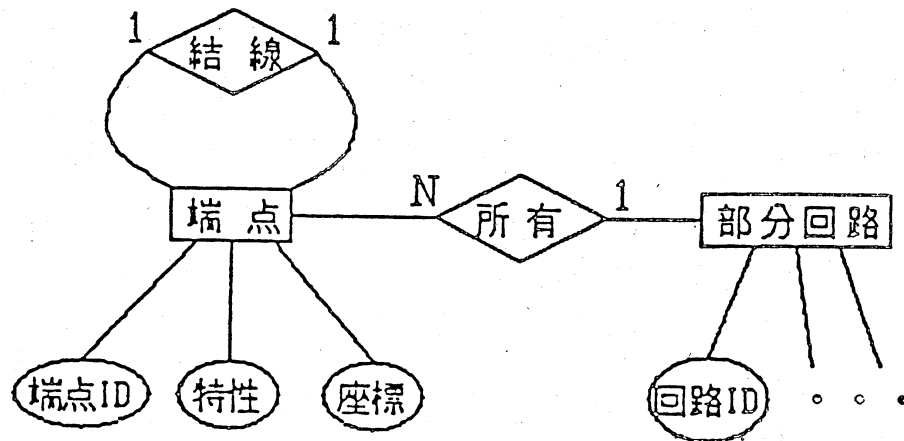


図3. 回路図の実体一関連図

である [2 , 3] 。 (A) を実現するためには、端点を識別できることと、端点同志の結線情報を保持する必要がある。 (B) を実現するためには、端点とそれが属する回路の情報が必要である。 (C) を実現するためには、端点の特性情報（トータムポール、オープンコレクタ、トライステート及び入力／出力端点の区別）が必要である。 (D) のためには、結線情報と端点の属性情報が必要である。図3は、以上の考察を基に作られた回路図の実体一関連図である。これに対応する関係データベース・スキーマを以下に示す。

COMP(DIV_ID/ID ,DIV/ABS)

TERM(T_ID/ID ,DIV_ID/ID,CHR/ID,XC/AD,YC/AD)

CONN(L_ID/ID,S_ID/ID,DIV_S/ID, D_ID/ID,DIV_D/ID)

リレーションCOMPは作成中の回路図の構成部品である回路図の識別名、パラメータの値などを保持している。リレーションTERMは端点、及び端点が属している構成部品の識別名、端点の特性、表示座標を保持している。リレーションCONNは結線の状態を表わすリレーションで、結線の識別名と始点と終点の識別名を保持している。

なお、結線の図形表現は、線画像ファイルに記憶されており、結線の識別名をキーにして操作される。構成部品が操作されるたびにリレーションCOMPとTERMの内容が更新される。また、結線が行なわれるたびにリレーションCONNと線画像ファイルの内容が更新される。

3. 関係演算による妥当性チェック

(A) 未接続端点とは回路図中の端点のうち、結線の始点、終点のいずれにもなっていない端点である。関係TERMには、回路図中のすべての端点が登録されている。また、関係CONNには、結線に関与している始点と終点が登録されている。未接続端点を検出するアルゴリズムは SQL流の表現で以下のようなになる [4]。

```
TR <--- SELECT T_ID , DIV_ID FROM TERM
        DIFF
        ( SELECT S_ID , DIV_S FROM CONN
          UNION
          SELECT D_ID , DIV_D FROM CONN );
```

こうして選択された端点を編集集中の画面に表示するためには、端点に対応する表示位置を求め、これを図形表示すれば良い。

```
DISPLAY 'MARK' AT
        ( SELECT XC , YC
          FROM TERM , TR
          WHERE TERM . T_ID = TR . T_ID ;
          AND TERM . DIV_ID = TR . DIV_ID )
```

第1行目は、第2から5行目の式で検索された端点の座標 (XC , YC)に 'MARK'なる印を表示するADAMの図形表示関数である。

(B) 未接続入力端点を含む部分回路とは、回路図を構成する入力端点 (回路図のインターフェースの端点を含まない) から結線の終点を取り除いた端点が属する部分回路である。

```
TR (--- SELECT T_ID , DIV_ID
      FROM TERM
      WHERE DIV_ID /='D0'
      AND T_ID = 'I*'
      DIFF
      SELECT D_ID , DIV_D
      FROM CONN ;
```

上式の第3行目は、端点が編集中の回路図のインターフェースの端点でないことを示す ('D0' は回路図エディタEVEによって生成される識別名である)。現在のEVEでは、先頭の1文字が 'I' であるとき入力端点を意味している。従って第4行目は、端点が入力端点であるものだけを選択することを指定している。第6, 第7行目は、結線の終点を選択している。こうして選ばれた端点を含む部分回路を編集中の画面に表示するためには以下の演算を実行する。

```
DISPLAY 'INTERFACE' OF
( SELECT DIV
  FROM COMP , TR
  WHERE COMP . DIV_ID = TR . DIV_ID );
```

上式の第1行目は、属性DIVに含まれる部分回路のインターフェースの図形を表示するADAMの図形表示関数である。

(C) 不正接続の検証の例として、次のチェックを考える。“トータムポール出力のとき、2つ以上の出力線が交わることはない。しかし、トライステート、オープンコレクタの出力は、2つ以上の出力線が交わることがある。”このチェックを行なうためには、終점에接続している結線の数が2以上である結線の始点を求め、その始点がトータムポール出力であることを確認すればよい。

```
TR (--- SELECT S_ID , DIV_S , L_ID
        FROM CONN
        GROUP BY D_ID , DIV_D
        HAVING COUNT (*) > 1 ;
TS (--- SELECT L_ID
        FROM TERM , TR
        WHERE TERM . T_ID = TR . S_ID
        AND TERM . DIV_ID = TR . DIV_S
        AND TERM . CHR = 'TP' ;
```

不正接続を行なっている結線を図形表示するためには、

```
DISPLAY 'IMAGE' OF ( TS );
```

を実行すればよい。EVEでは、結線の表示イメージを画像として扱っているの
で、'IMAGE'なる指定をしている。

(D) ファンアウト数のチェックをするためには、結線の始점에接続している終点の

数をカウントし、指定したファンアウト数Nより多ければ、その回路の図形を表示する。

```
TR <--- SELECT DIV_S
          FROM CONN
          GROUP BY S_ID , DIV_S
          HAVING COUNT (*) > N ;
          DISPLAY 'INTERFACE' OF
          ( SELECT DIV
            FROM COMP , TR
            WHERE COMP . DIV_ID = TR . DIV_S ) ;
```

4. 回路図編集とデータ操作

図4は、EVEによる回路図編集の流れを示している。EVEが起動されたとき、回路図を表現するためのリレーションCOMP, TERM, CONNはタプルを含んでいない。回路図の編集を行なうためには4個の対象物、すなわち構成回路、端点、結線、概形を定義する。回路図の編集が終了した後に、妥当性チェックを行なう。不具合が発見されたときは再度図面編集を行なう。完成した回路図はデータベースに登録する。ある回路図が複数の回路分類に属することがある。図2に示したように、ADAM&EVEでは1枚の回路図を複数の分類に登録することにより、この要求に答えている。図5は編集した回路図の一例であり、図6は図5に対応するデータである。以下、回路図編集の各機能と性能について述べる。

4.1 回路図の引用

回路図の引用は基本回路の引用と、標準IC図またはユーザが定義した回路図の引用に大別できる。しかし、いずれの場合でも最終的には1枚の回路図が選択される。ここでは、回路図が選択された後の処理について述べる。回路図を引用すると

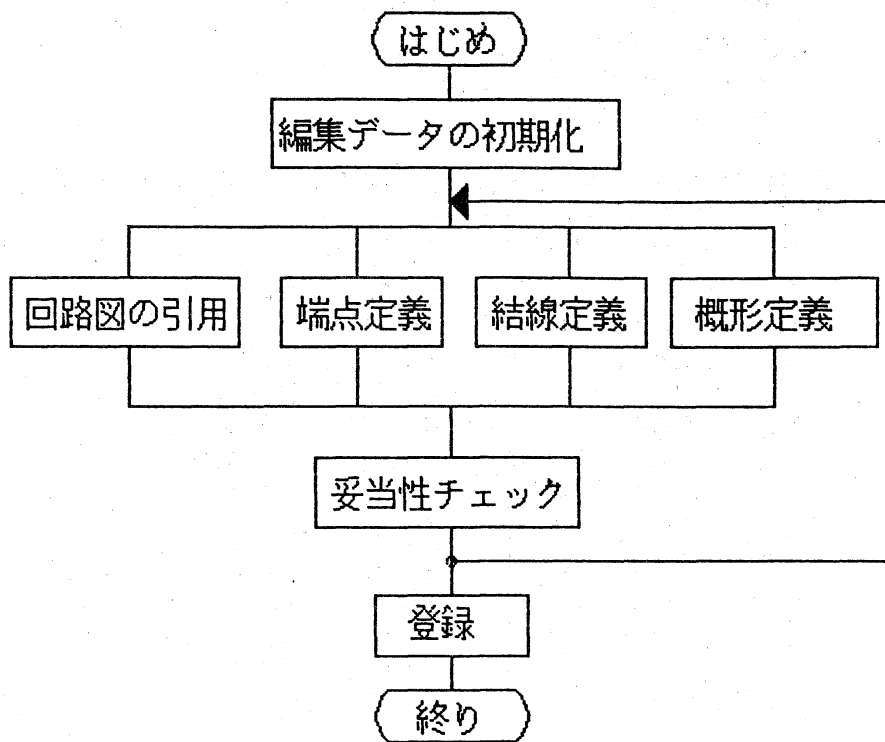


図4. 回路図編集の流れ

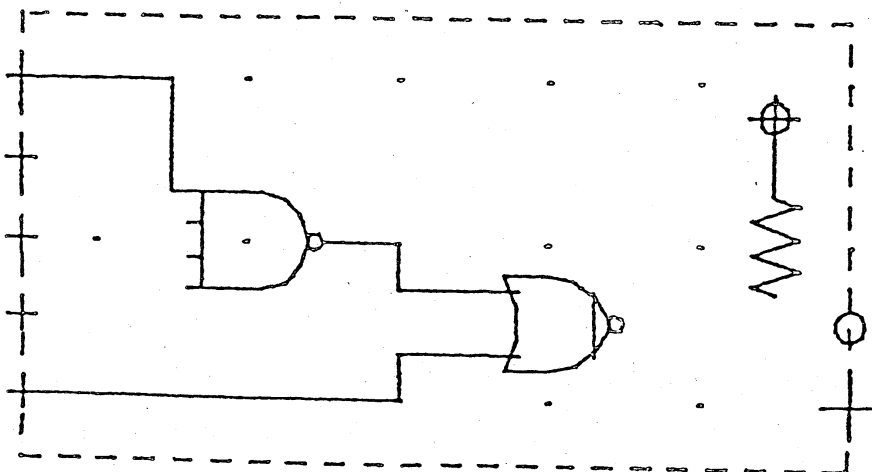


図5. 回路図の例

```

COMP ( DIV ID/ID, DIV/ABS ).
D001, REGV(-5.0,+10.0)
D002, VCC(-5.0,+18.0)
D003, NOO2(-15.0,+5.0)
D004, NAT4(-35.0,+10.0)
D0, TEST(X,Y)

```

```

TERM ( T_ID/ID, DIV ID/ID, CHR/ID, XC/AD, YC/AD ).

```

```

-I0, D001, TP_N, -5.0, +7.0
I1, D001, TP_N, -5.0, +13.0
Q0, D002, TP_N, -5.0, +18.0
Q0, D003, OC_N, -15.0, +5.0
I0, D003, TP_N, -24.0, +7.0
I1, D003, TP_N, -24.0, +3.0
Q0, D004, TP_N, -35.0, +10.0
I0, D004, TP_N, -44.0, +13.0
I1, D004, TP_N, -44.0, +11.0
I2, D004, TP_N, -44.0, +9.0
I3, D004, TP_N, -44.0, +7.0
I000, D0, TP_N, -55.0, +20.0
I001, D0, TP_N, -55.0, +15.0
I002, D0, TP_N, -55.0, +10.0
I003, D0, TP_N, -55.0, +5.0
I004, D0, TP_N, -55.0,
Q000, D0, TP_I, , +5.0

```

```

CONN ( L_ID/ID, S_ID/ID, DIV_S/ID, D_ID/ID, DIV_D/ID ).

```

```

L001, I000, D0, I0, D004
L002, Q0, D004, I0, D003
L003, I004, D0, I1, D003
L004, Q0, D002, I1, D001

```

図6. 回路図に対応するリレーション

き回路のインターフェースに関する情報のコピーと回路の図形表示を行なう。例えば、4入力NOR ゲートを引用したとき、以下の処理をする。

(1) ゲートに対応するCOMPとTERMの内容を編集集中の回路図のリレーションCOMPとTERMに追加する。このときDIV-ID内の識別子"D0"を編集集中の回路図中での識別子(EVEが生成する)に変える。また、引数付き実現値の引数に、編集図中での座標値を代入する。

(2) 回路図の図形を表示するためには引数付き実現値名と引数の値をADAMに送り、図形表示を依頼する。

初めて回路図を引用するときは、ディスクとの入出力が必要となるため処理が遅くなるが(4入力 NORで250ms)、次の引用からはすべて主記憶内で処理されるので高速である(4入力 NORで60ms)。

4・2 端点の定義

端点を定義したときは、編集集中の回路図のリレーションTERMに端点に対応するタプルを追加する。この処理はすべて主記憶内で処理される(30ms/端点定義)。

4・3 結線定義

結線を定義したときは、編集集中の回路図のリレーションCONNに、<結線の識別子, 始点, 終点>なるタプルを追加する(ただし、結線を行なうとき、始点及び終点の識別子がすでに定義されているか否かを制約演算によって判定している)。さらに、結線の表示図形を線画像ファイルに登録する。典型的な結線に要する時間は80msである。

4・4 概形図形定義

概形図形を定義したときは、入力された線画像に識別名(現在のインプリメントでは"GF1")を付け、線画像ファイルに登録する。さらに、回路図のインターフェースを図形表示するコマンドにDISP(GF1)なるコマンドを追加する。この処理に要する時間は100msである。

4・5 妥当性チェック

回路図を構成部品、端点、結線に分解して表現しておくこと、未結線端点の指摘、ドライブ信号のない回路の指摘などの回路図妥当性チェックを行うことができる。

```

#4 ( T_ID/ID, DIV_ID/ID ).
  I0, D001
  I1, D004
  I2, D004
  I3, D004

```

(1) 未接続入力端点

```

#2 ( T_ID/ID, DIV_ID/ID, DIV_ID/ID, DIV/ABS ).
  I0, D001, D001, REGV(-5.0,+10.0)
  I1, D004, D004, NAT4(-35.0,+10.0)
  I2, D004, D004, NAT4(-35.0,+10.0)
  I3, D004, D004, NAT4(-35.0,+10.0)

```

(2) 未接続入力端点とそれを含む部分回路

```

#1 ( DIV/ABS ).
  REGV(-5.0,+10.0)
  NAT4(-35.0,+10.0)

```

(3) 部分回路

図7. 処理の中間結果

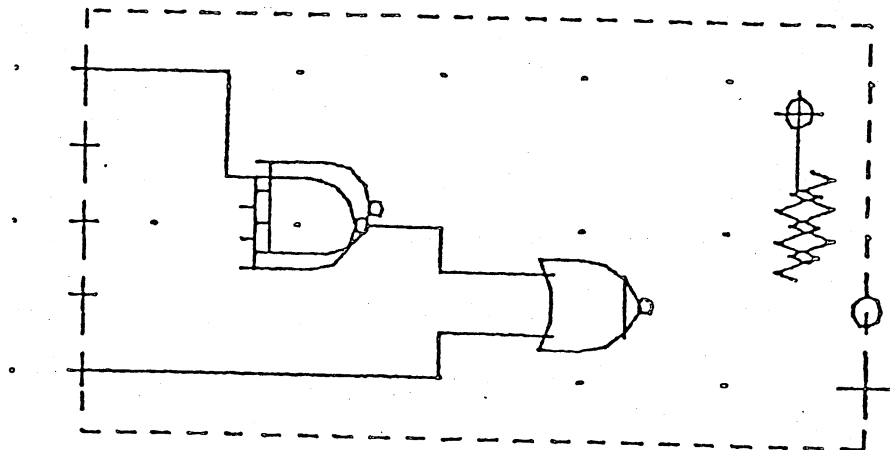


図8. チェックの結果の図形表示

図6に示したリレーションに未接続入力端点を含む部分回路を指摘するアルゴリズムを適用した結果を図7に示す。図7(1)は、回路図を構成する入力端点から結線の終点を取り除いた端点の集合を表している。図7(2)は、これらの端点が属している部分回路をJOIN演算によって検索したものである。図7(3)は、未接続ドライブ信号を含む部分回路の集合である。こうして得られた部分回路を図5に重ね合せて表示したものを図8に示した。以上の処理に150ms弱のCPU時間を要した。チェックに要する時間は、ほぼ端点の数に比例するものと思われる。その理由として、このアルゴリズムは、1回のJOIN演算を除けば、すべてタプル数Nに比例（あるいは、 $\log N$ に比例）する処理時間を要する演算で構成されている。さらに、上記のJOIN演算は処理の最終段階で行なわれるために、関与するタプルはかなり少なくなっているからである。

4・6 登録

回路図をADAMに登録するためには、利用者は以下の操作を行なう。

- (1) 作成した回路図をどの回路分類に登録するかを指定する。
- (2) 分類に応じた属性に対し、付加したい値を指定する。

上記の利用者の操作に対応して、ADAMでは回路の分類に応じたリレーションに対し、指定された属性値を有するタプルを挿入する。その後、図2に示した回路図の階層情報の変更を行なう。次に、登録したい回路図のリレーションCOMP, TERM, CONN, 図形ファイル, 線画像ファイルをディスクに書き出す。この処理に要する時間は約3000msある。

謝 辞

今回の発表の機会を与えて下さいました東京大学・大須賀 節雄教授に感謝します。日頃より有益なコメントを賜わります当社情報電子研究所ソフトウェア部長・首藤 勝博士に感謝します。

参考文献

- [1] Katz , R.H. : Information Management for ... ,Springer , 1985 .
- [2] Ravid , E. et. al. : Problems in Logic-Array ... VLSI DESIGN Sept. 1984. p.46-51 .
- [3] The Bipolar Digital IC ... , Texas Instruments, 1985 .
- [4] Date , C.J. : An Introduction to DB ... , ADDISON-WESLEY , 1981 .
- [5] 宇田川・溝口 : 拡張リレーショナル・データベースADAM... 情報処理学会, DB研究会, DB-51 , 1986 .
- [6] 桑原 : 各種CADシステムで実用化が始まったエンジニアリング・データベース、日経エレクトロニクス、1985年12月25日号、P.143-160 .