

## 自動論理合成システムLUNAの適用と評価

黒沢雄一, 増淵美生, 西尾誠一, 上田隆司, 宮田操

株式会社 総合研究所

VLSI等の大規模デジタルシステムの機能・論理設計の効率向上を目的として、我々はハードウェア設計言語 $H^2DL$  (Hierarchical Hardware Design Language) とこれに基づく自動論理合成システムLUNA (Logic synthesis system based on Universal Network Architecture) を開発した。LUNAの論理合成は、ルールベースによることが特徴である。現在LUNAは、各種の実際の設計に適用しその評価を進めている。このうち一部の製品については設計が完了し、機能・論理設計の期間を従来の1/2とすることができ、その有効性が実証された。

本報告においては、まず、 $H^2DL$ 及びLUNAの概要を述べ、次に設計の完了した実製品を含む各種回路へのLUNAの適用結果をもとにして、LUNAの定性的評価及び実行時間等に関する定量的評価を行ない、最後にLUNAを用いた有効な設計手法及びこれらの適用において明らかになった今後拡張すべき機能について論ずる。

## Application and Evaluation of the Automatic Logic Synthesis System LUNA

Yuichi KUROSAWA, Yoshio HASUBUCHI, Seiichi NISHIO, Ryuji UEDA, Hisao MIYATA  
Toshiba R&D Center, 1, Toshiba-cho, Komukai, Saiwai-ku, Kawasaki, 211, Japan

In order to do functional and logic design of VLSI more and more effectively, we developed a hardware design language  $H^2DL$  (Hierarchical Hardware Design Language) and an automatic logic synthesis system LUNA (Logic synthesis system based on Universal Network Architecture). It is a main feature of LUNA that it adopts rule-based approach. LUNA has been applied to several actual designs and has been evaluated. One of these designs was completed and the functional and logic design effort was reduced to one half of that it had ever taken.

In this paper, firstly we describe the overview of  $H^2DL$  and LUNA. Secondly we evaluate our system on the basis of the results of those applications stated above and finally we propose an effective design method using LUNA and describe the extension of our system.

## 1. はじめに

近年のLSI技術の急速な進歩により、VLSIのような大規模システムの実現が可能となってきた。このような大規模システムの機能・論理設計レベルからの設計効率を目的として、我々は、ハードウェア設計言語H<sup>2</sup>DL (Hierarchical Hardware Design Language) 及び、これを入力として論理回路を自動生成する自動論理合成システムLUNA (Logic synthesis system based on Universal Network Architecture) を開発した。現在、LUNAは数種の実製品の設計に適用し、その評価を進めている。このうち、一部の製品については適用、評価が完了した。

本稿においては、まず、H<sup>2</sup>DL [1], [2] 及びLUNAシステム [3], [4] の概要、次にLUNAの実際の設計への適用結果とその考察を述べ、最後にLUNAの機能拡張、今後の課題等について述べる。

## 2. H<sup>2</sup>DL及びLUNAの概要

### 2-1. H<sup>2</sup>DLの概要

H<sup>2</sup>DLは、LSIの機能設計レベルから階層設計を進められることを目指して開発されたハードウェア設計言語であり、内部仕様記述、ハードウェアプロセス記述、インターフェース記述、接続記述の4つの記述形式をもつ。以下では、LUNAと直接関連のある内部仕様記述、ハードウェアプロセス記述、接続記述について説明する。

#### ① 内部仕様記述

設計対象とするモジュールのレジスタ転送レベルの機能を記述するものであり、自動論理合成システムLUNAの入力となる。内部仕様記述の主な特徴は以下のとおりである。

- ・状態遷移表現を用いることができる。
- ・記述内では、回路の構成要素として、レジスタ、メモリ等の他にハードウェアプロセスと呼ばれるユーザ定義の機能モジュールを用いることができる。
- ・割込み的な機能が容易に表現できる。

#### ② ハードウェアプロセス記述

モジュールの機能のみに着目して、ハードウェアによる実現方法は意識しないレベルの記述である。モジュールの各機能は、手続的な記述であるプロセスにより、表現される。このハードウェアプロセス記述自身は、LUNAの入力とはならないが、内部仕様記述と組合せて、設計対象モジュール全体の機能シミュレーションを行なうことができる。

#### ③ 接続記述

モジュールの階層構造や接続関係を素子形式により記述するものである。

内部仕様記述におけるハードウェアプロセスの使用は、ハードウェアプロセスの各プロセスを起動させ、その出力を受け取るという形で行なわれる。図1に内部仕様記述とハードウェアプロセス記述の記述例を示す。

### 2-2. LUNAの概要

LUNAはH<sup>2</sup>DL内部仕様記述を入力とし、最終ターゲットテクノロジーにおける素子の接続関係を接続記述として出力するシステムである。LUNAのシステム構成図を図2に示す。LUNAによる論理合成は、以下のステップで進められる。

#### ステップ1 トランスレーション

H<sup>2</sup>DL内部仕様記述を入力として、ほぼ記述と一対一に対応した初期回路を生成する。初期回路は実際にどのような素子で実現するかについては考慮していない抽象的なものであり、この段階ではAND、OR等のゲート類の入力には制限がなく、内部仕様記述で用いているレジスタ、関数、ハードウェアプロセス、状態等是一个の構成要素となっている。

#### ステップ2 モジュール割付け指定

設計者から合成系への指示をコマンドの形式で受取り、それに従った処理を行なう。

##### (1) 状態フリップフロップの割当て

設計者の指定するビットパターンに従って、回路中の状態要素を状態フリップフロップにおきかえる。

##### (2) バス割当て

回路中の指定された内部信号をバス構造で実現する。

##### (3) レジスタ等への標準セル割当て

レジスタ等の構成要素にステップ5の段階で自動的に割当てられる標準セル以外を割当てたい場合の処理を、設計者の指定に従って行なう。

##### (4) ハードウェアプロセスの変換

回路中のハードウェアプロセス要素には、実際のハードウェアのイメージはない。そこで、ハードウェアプロセスの起動条件や引数と実際のハードウェアの外部端子との対応関係を指定し、ハードウェアプロセス要素とその周辺の回路を実際のハードウェアと対応するものにおきかえる。

#### ステップ3 高機能要素の処理

回路中の関数、デコーダ、セレクタ等の要素をゲートに

(INTS) CPU120;

```

<IN> RESET, "RESET SIGNAL FROM BUS CONTROLLER"
PIBSEN, "INSTRUCTION ON IBUS IS ENABLE"

IFLG8<12:0>;

<HPU>
EADDR : HEADD<31:0>; "EFFECTIVE ADDRESS ADDER"
LOCADDR : HLOCADD<31:0>; "LOCATION ADDER"
GR : HGR<127:0>; "GENERAL REGISTER"
. . .

<ENDHPU>;
<TER> "STATE OF PIPELINE REGISTER"
PT1E, "IF 1,QIR1'S STATE IS TQ1E."
PT1R, "IF 1,QIR1'S STATE IS TQ1R."

```

宣言部

```

PINT<1:0>;
<REG> POP1<7:0>, "OP CODE"
PSR1<7:0>, "SOURCE REGISTER"
. . .
LOC<31:0>; "LOCATION COUNTER"

```

ハードウェアプロセス  
使用の宣言

```

<STATIC>
PINT(1)=PEX3(0)&PT3R;
. . .
<ENDSTATIC>;

```

\*\*\*\*\*

```

<STATE> PSTATE1 : MCLK ;
TQ1E:
. . .
TQ1R:PT1E=0;PT1R=1;PT1W=0;PT1D=0;
/*-----*/
"READ OUT OF BASE AND INDEX REGISTER AT THE SAME TIME"

```

状態遷移表現  
による記述

機能記述部

```

CASE PAM1<1:0> OF
2B00::NULL;
2B01::IF PCHNGI
THEN NULL;
ELSE HGR.READI<PIX1>;
IRL=HGR<127:96>;
ENDIF;
2B10::IF PCHNGR
THEN NULL;
ELSE HGR.READB<PBR1>;
BRL=HGR<95:64>;
ENDIF;
2B11::IF PCHNGB!PCHNGI
THEN NULL;
ELSE HGR.READB<PBR1>;
IRL=HGR<127:96>;
HGR.READI<PIX1>;
BRL=HGR<95:64>;
ENDIF;
ENDCASE;

```

ハードウェアプロセス  
の起動  
結果の参照

```

/*-----*/
"THE INSTRUCTION IN QIR1 GO TO QIR2, "
"AND NEXT INSTRUCTION IS NOT READY. "

```

```

IF RESET!MCLR
!"PIBSEN
&"(PCHNGB!PCHNGI)
&"POFST1
&(PT2E
!"PCHNGS
&(PT3E
!"PMUBSY&(BUSEN!"PTRG)
&(MTHW2!MEND<4>)))
THEN NEXT TQ1E;
ENDIF;
. . .

```

```

<ENDSTATE>;
<STATE> PSTATE2 : MCLK ;
. . .
<ENDSTATE>;
. . .

```

(ENDINTS);

図 1 ( a ) 内部仕様記述の例

展開する。ここで、設計者側で用意しているセルライブラリにこれらの高機能要素と対応する要素があれば、それを割当てすることも可能である。

#### ステップ4 回路の簡単化

回路の論理的に冗長な部分の簡単化を行なう。簡単化は local transformation方式を採用している。この処理は、ステップ1, 2の各処理の後においても必要に応じて行なわれる。

#### ステップ5 自動セル割当て

回路の構成要素をセルライブラリとして用意されている実際の物理的な素子(標準セル)におきかえる。また、セルライブラリの特性データにもとづいてファンアウト調整も行なう。

#### ステップ6 接続記述変換

実際の物理的な素子から成る回路から、H<sup>2</sup>DL接続記述を生成する。生成された接続記述は、論理シミュレータ、

```

<HP> GR<127:0>;          * GENERAL REGISTER *
*      31-----16 15-----8 7-----0 *
*      (---M2---) (---M1---) (---M0---) *
* M0, M1 AND M2 ARE GLOBAL VARIABLES. *

<SIG> M0(16)<7:0>;        ← グローバル変数の宣言
      M1(16)<7:0>;
      M2(16)<15:0>;

<PROC> READB(ADRS);       ← ローカル変数の宣言
<SIG> ADRS<3:0>;
<CPART>
      GR<71:64>=M0(ADRS);
      GR<79:72>=M1(ADRS);
      GR<95:80>=M2(ADRS);
<ENDCPART>;
<ENDPROC>;

<PROC> READI(ADRS);
<SIG> ADRS<3:0>;
<CPART>
      GR<103:96>=M0(ADRS);
      GR<111:104>=M1(ADRS);
      GR<127:112>=M2(ADRS);
<ENDCPART>;
<ENDPROC>;

<PROC> WRITED(ADRS, VAL, TRMODE);
<ENDPROC>;
      . . .
<ENDHP>;
  
```

図1 (b) ハードウェアプロセス記述の例

レイアウトシステムの入力となる。

以上の処理のうち、ステップ1, 6は通常の手続き的プログラムによって処理が行なわれるが、ステップ2~5はすべてのルールベースに基づき論理合成エンジンによって処理が行なわれる。

ここで、ステップ2~5の処理の順序については、あらかじめ最適なもの定まっているわけではなく、設計者の回路設計手法、ライブラリ等を考慮して決めることが望ましい。これに柔軟に対処するためには、回路のデータ構造と処理方式を統一的なものとする必要がある。このような観点から、LUNAにおいてはトランスレーション直後の抽象的な回路から、実際の物理的な素子から成る回路までを統一的に扱うUNETというデータ構造を定め、また処理方式としてルールベース方式を採用している。図3にトランスレーション直後の回路とセル割当て後の回路の一例を示す。

この他に、本システムには、上記の論理合成過程の任意の段階の図面を出力することのできる自動図面生成システム

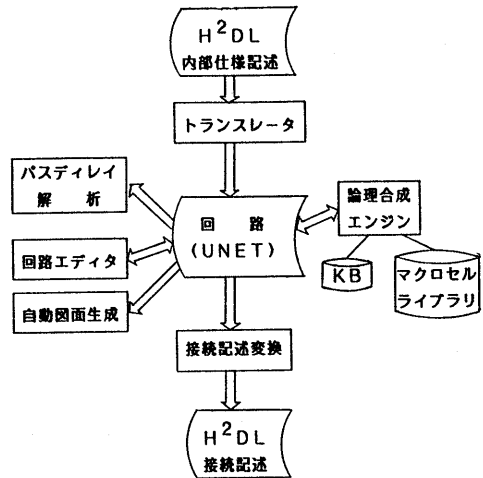


図2 自動論理合成システムLUNAの構成

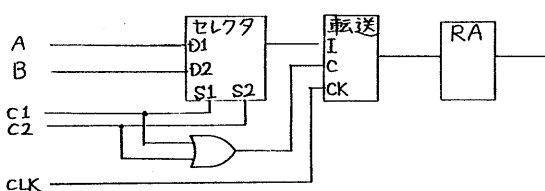


図3 (a) トランスレーション直後の回路の例

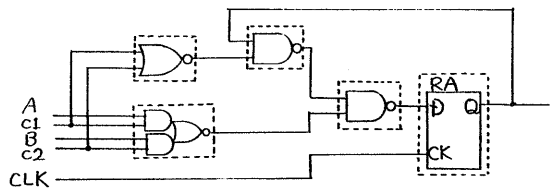


図3 (b) セル割当て後の回路の例

ム、合成された回路のバスディレイ計算を行なうシステム、合成された回路のマニュアル修正を可能とするシステムがある。

### 3. LUNAの実製品への適用結果例

本章においては、LUNAを適用して実際に設計の完了した製品の例としてスーパーミニコンの演算制御プロセッサ（以下BPHと略記）をとりあげ<sup>[5]</sup>、まず、3-1. においてその適用結果を述べ、次に、3-2. において、本適用において明らかになったH<sup>2</sup>DL及びLUNAの利点、問題点等をまとめる。

#### 3-1. LUNAのBPH開発への適用結果

BPHは8Kゲート規模のゲートアレイ4種から構成される。図4にLUNAの適用結果の要約を示す。機能シミュレーション、ゲートシミュレーション、テストデータ生成等を含めると、H<sup>2</sup>DL記述作成から接続記述完成まで設計期間約5ヶ月、作業量2.1人月を要した。これはLUNAを使用せず、人手により論理回路図及び接続記述を作成する従来の手法による設計期間の約1/2であった。

#### 3-2. BPH開発におけるLUNAとH<sup>2</sup>DLの評価

##### ① H<sup>2</sup>DLの評価

###### \*利点

言語自体の習得は容易であり、初心者でも1~2週間可能である。日本語で表現しにくい、あるいは日本語で表現すると曖昧となる仕様も明確に表現でき、ドキュメント

としても有効である。また、論理合成の対象となる部分だけでなく、外部回路の記述を行ない、接続記述等で結合しミックスシミュレーションを行なうことにより、外部環境を含めた検証ができ、単体レベルのシミュレーションでは発見できないようなバグの発見等に大きな効果がある。

###### \*記述上の留意点

H<sup>2</sup>DLでは、容易にいくらでも複雑な論理を書くことができるが、レジスタ間の論理はクロックサイクル内で動作可能であるようにする等、ある程度ハードウェアを意識しながら書いてゆくことは必要である。また、タイミング

回路	回路			
	GA1	GA2	GA3	GA4
記述	計1. 5人月			
記述量 [行] タイプ	1430 制御系	1450 制御系	600 データ系 +コード変換系	540 データ系 +コード変換系
論理合成				
処理時間 [秒]	844	881	799	516
処理時間内訳 [%]				
ステップ1	4	7	3	5
ステップ2	3	1	3	8
ステップ3	18	11	22	20
ステップ4	54	67	35	35
ステップ5	21	14	37	32
ゲート数	5065	3987	6635	5395
セル数	2250	1914	3568	2641
バスディレイ解析と回路変更	5人月			

(注) 処理時間は15MIPSの計算機を使用した場合の値。

図4 LUNAの実製品開発への適用結果

回路	回路						
	A	B	C	D	E	F	G
記述							
記述量 [行] タイプ	2404 制御系 +コード変換系	489 制御系	698 制御系	1904 コード変換系 +制御系	1866 コード変換系	861 制御系	1014 制御系 +データ系
論理合成							
処理時間 [秒]	1624	206	194	2268	3819	221	395
内訳 [%]							
ステップ1	7	6	8	3	4	9	7
ステップ2	1	3	7	0.5	0	4	2
ステップ3	1	3	15	0.5	0.5	14	4
ステップ4	87	82	62	93	95	65	83
ステップ5	4	6	8	3	0.5	7	4
セル数	1143	347	598	807	236	367	550

(注) 処理時間は15MIPSの計算機を使用した場合の値。

図5 LUNAの論理合成結果

的に非常に厳しい部分は、ハードウェアプロセスとして記述・シミュレーションを行わない設計は人手で行なう等、LUNA適用部と人手設計部との切り分けを考慮することも重要である。

## ② LUNAの評価

### \*利点

3-1. で述べたように、現状でも従来の半分の期間で機能・論理設計を行なうことができ、設計期間の短縮に対する効果は多大である。今回の実製品開発において自動生成された回路の質については、設計担当者の詳細な図面出力のチェックにより、「人手と同等」という評価を得た。また、クロック分配、ファンアウト調整法等のユーザ毎に特有な手法については、論理合成ルールの追加により、比較的容易に対応できた。これは、ルールベースシステムの大きな利点である。さらに、LUNAのパスディレイ解析プログラムにより、回路が自動生成された時点で直ちにタイミング制約のチェックが可能であり効果があった。

### \*問題点

現在のLUNAによる論理合成では、タイミング制約を満たす回路が必ず生成できるという保証はない。そのため、今回のBPHへの適用においても、タイミング解析とそれにとまなう回路修正（回路エディタによる）に約1ヶ月の期間（作業量約5人月）を要した。回路修正の目的は主として以下の2点であった。

#### (1) パスの高速化

H<sup>2</sup>DL記述時点で、見当をつけたパスについては、予想通りの回路が生成されほとんど変更を必要としなかったが、記述時点であまり注意を払わなかった部分については、ディレイの大きいパスが生成され、高速化の変更を必要とするものもあった。また、値の確定するのが遅い外部入力から内部レジスタへの信号についても伝播時間を短縮するための変更が必要な場合があった。これらは、現在の論理合成が面積最小化を評価基準としており、段数を考慮に入れていないことが原因である。

#### (2) ホールドタイム補償

使用したセルの最小遅延時間が非常に小さいため、信号の伝播が遅すぎ、レジスタ入力にディレイを増加させるための修正を必要とする部分があった。

## 4. LUNAの評価

本章においては、BPH(3.)の適用結果に他の数種の製品への適用結果を加え、LUNAの評価を行なう。

図5にLUNAを適用して開発中の回路7種の論理合成結果を示す。ここで、図4、図5の各回路のH<sup>2</sup>DL記述

はその特徴から大きくデータ系、制御系、コード変換系に大別できる。データ系の記述は、多ビットのレジスタ、内部信号等への転送、代入文が中心となる記述であり、制御系の記述は、レジスタへの転送等の制御信号を発生する1ビットの内部信号への代入文、状態遷移記述が中心となる記述である。また、コード変換系の記述では、入力ビットパターンに応じ多ビットの信号線に数値を代入する文が中心である。

### 4-1. 定性的評価

他の各種製品開発においても、ほぼ3. で述べたと同様な評価が得られた。すなわち利点としては、H<sup>2</sup>DL記述のしやすさ、ドキュメントとしての有効性、自動論理合成により論理設計期間の大幅な短縮が可能、設計対象規模の見積りが容易に可能である点等があがっており、留意点としては、ハードウェアをある程度意識したH<sup>2</sup>DL記述をすること等があがっている。

### 4-2. 定量的評価

本節においては、図5～図8のデータに基づき、LUNAの定量的な評価を行なう。

#### ① H<sup>2</sup>DL記述量、処理時間、回路規模間の関係

図6にH<sup>2</sup>DL内部仕様記述の行数とLUNAによる論理合成の処理時間の関係、図7にLUNAによる論理合成結果の回路の標準セル数とLUNAによる論理合成の処理時間の関係、図8にH<sup>2</sup>DL内部仕様記述の行数とLUNAによる論理合成結果の回路の標準セル数との関係を示す。

図6～図8により次のようなことがわかる。

・D、Eのようなコード変換系の回路を除けば、論理合成の実行時間は、データ系、制御系の回路それぞれについて記述量に対して比例に近い関係にあり、また、設計対象の回路規模に対しても爆発的には増大しない。すなわち、LUNAの論理合成において、local transformation方式を採用した効果があらわれている。

また、対象回路のタイプによる特徴としては、

・A～C、Fのような制御系の回路の方がGA3、GA4のようなデータ系の回路に比べ、記述量の割に処理時間は少ない。

・データ系の回路の方が、制御系の回路に比べ、記述量の割に合成される回路規模はかなり大きい。

・コード変換系の回路は、他と比べ、記述量の割に合成される回路規模は非常に小さく、多くの処理時間を要する。等のことがわかる。

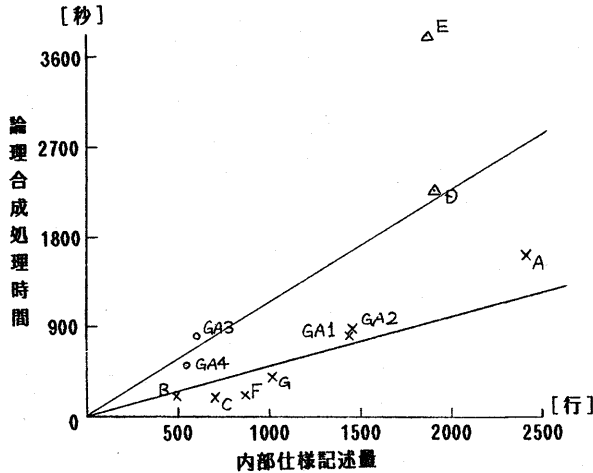


図6 内部仕様記述量と論理合成処理時間の関係

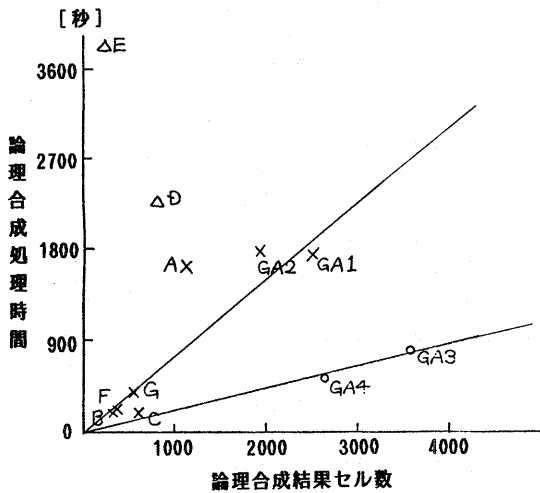


図7 論理合成回路規模の関係と論理合成処理時間の関係

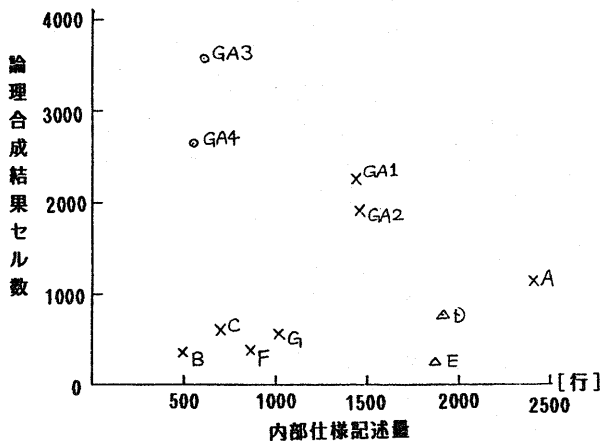


図8 内部仕様記述量と論理合成回路規模の関係

〔注〕 図6～図8において、×…制御系中心 ○…データ系中心 △…コード変換系中心

## ② LUNAによる論理合成実行時間の分析

図4、図5に論理合成全体の処理時間と共に、2-2.で述べた論理合成の各ステップの処理に要した時間の割合を示す。これらによれば、多くの場合、単純化の処理に40%以上の時間が費やされており、この処理が論理合成の大きな部分を占めていることがわかる。

また、対象回路のタイプ毎の特徴としては、以下のよう  
なことがわかる。

- ・制御系の回路では、単純化に要した時間の割合が多く、ビット展開に要した時間の割合は少ない。
- ・データ系の回路では、他の回路に比べ、単純化に要した時間の割合は比較的少なく、ビット展開及びセル割当てに要した時間の割合が多い。
- ・コード変換系の回路では、単純化に要した時間がほとんどを占め、ビット展開及びセル割当てに要した時間の割合は、他の回路に比べ、特に少ない。

これは、制御系の回路については、論理が複雑であり、状態遷移表現等の高位表現を用いてH<sup>2</sup>DL記述することによりかなり単純化が行なわれるためであり、データ系の回路については、1ビットの制御信号を受け多ビットのデータ転送を行なう比較的単純な動作が中心であるために単純化処理の占める時間の割合が低く、逆に、ビット展開及びセル割当てに要する時間の割合が多くなると考えられる。

また、コード変換系の回路の単純化の処理時間の割合が特に大きいのは、入力パターンに対応した出力パターンを記述する大きなCASE文や条件部に多ビットの数値との比較式、実行部に多ビットの数値代入文を含むようなIF文から成るといふ記述の特徴によると考えられる。それゆえ、コード変換系の回路については、論理合成ルールの使われ方をさらに詳しく調べ、この記述の特徴をうまく利用した論理合成ルールの選択と適用順を考えることも必要と考えられる。

## 4-3. 図面生成システムの評価

図9に2種の論理合成結果の回路を自動生成システムによって図面出力した結果を示す。ここで、回路Iは回路Hのうちの制御部分を抜き出したものである。(データ系の部分は、サブモジュールとしてある。)回路Hにおいては、数十ビットのレジスタ等があるために、図面枚数が非常に多く論理が追いつきにくい図面となったが、回路Iの制御回路においては、枚数も少なく比較的見やすい図面が得られた。

この例からもわかるように、セル数の多い大規模な回路では図面の質に問題があるが、規模がそれほど大きくない回路に対しては比較的、良好な図面が得られる(A1サイ

ズの場合900セル以下<sup>[6]</sup> )。

	回路	
	H	I
記述		
記述量 [行]	950	861
タイプ	データ系 +制御系	Hの制御系のみ
セル数	1958	367
図面出力		
処理時間 [秒]	531	179
枚数	35枚/A1	12枚/A1

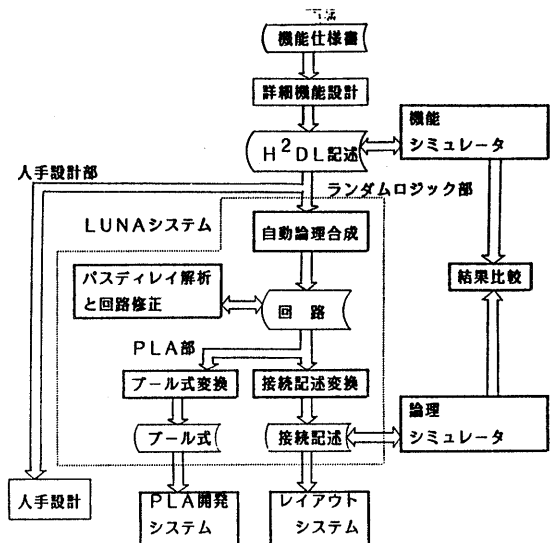
図9 図面生成出力結果

### 5. LUNAを用いた効率的な設計手法

以上述べてきたことから、LSI設計においてLUNAを利用する場合、次のような点に留意して設計を行なうことが効果的であると考えられる。

まず、制御系の回路は、一般に論理が複雑であり、人手で設計した場合に作業量が多く、設計ミスも入りやすい。これに対し、LUNAを用いた場合、H<sup>2</sup>DL記述量に対する論理合成の処理時間の増加は比例係数も小さいこと、H<sup>2</sup>DL記述において状態遷移表現が可能であり、複雑な論理も比較的容易に記述できること、誤りのない回路が合成できること等の利点がある。それゆえ、制御系の回路は、LUNAの適用による効果が最も大きく、設計期間の大幅な短縮が可能である。

一方、データ系回路については、レジスタ間の論理がクロックサイクル内で収まるかといったタイミング的な要素を考慮すること等が必要である。また、演算器等の規則性のある回路で、人手設計を行なった方が、スピード、面積の点で効率的なものが設計可能な部分は、人手設計とする等、LUNAを使用する部分と人手設計部分との切り分けを考慮することも重要である。さらに、コード変換系の回路は、PLAに適した回路と考えられ、PLAの利用により面積を大幅に減少できる可能性がある。それゆえ、このタイプの回路に対してはPLA使用部分の切り分けを考慮することが重要であると考えている。これについては、LUNAのトランスレーション結果からプール式を出力し、PLA開発システムと接続するツールが完成している。以上述べた設計手法をまとめると図10のようになる。



(注) □ は人手による処理。□ は計算機による処理。  
図10 LUNAを用いたLSIの有効な設計手法

### 6. LUNAの機能拡張

5.において、LUNAを利用して生成するランダムロジック部、PLA使用部分、人手設計部分の切り分けが重要であると述べた。現在、この切り分け作業は、H<sup>2</sup>DL記述の段階で行なう必要がある。しかし、H<sup>2</sup>DL記述の段階では、回路の詳細についてはわからないため、上記の切り分け作業を適切に行なうことは一般に困難である。それゆえ、この作業は自動論理合成を進めながら、適宜回路を評価し行なえることが望ましい。このような観点から、我々はLUNAの機能拡張を進めている。

- ① 回路 (UNET) 中の指定された一部分を抽出した回路をもとの回路とは別に生成する機能。
- ② サブモジュール (LUNAの回路中では、ブロックとして扱っている。) を含んだ回路とそのサブモジュール内部の回路から、前者のサブモジュール部分を展開した回路、すなわち前者の回路においてサブモジュールの部分を後者の回路でおきかえた回路を生成する機能。

これらの機能により、上述の切り分け作業の他に以下のような処理も可能となり、現在のLUNAの主な問題点はかなり解決できると考えている。

- ・論理合成された回路を評価して、タイミング制約を満たさない部分を切り出す。切り出した部分については、タ



タイミングを考慮した $H^2DL$ 記述を行ない再度論理合成を試みる、あるいは手設計等のタイミング修正を行なう。再設計を行なった後の回路はもとの回路にはめ込むこともできる。

・論理合成された回路の一部分のみを抽出し、自動図面生成システムにより図面出力する。小規模な回路であれば、見やすい図面が生成されるので、設計者の注目する部分のみを見やすい図面として出力することが可能となる。

## 7. おわりに

LUNA及び $H^2DL$ は各種の実製品開発に適用され、 $H^2DL$ による機能設計にしやすさ、LUNAによる論理設計期間の大幅な短縮等の利点があり、LSI設計において実用上非常に有効であるという評価を得た。現在のLUNAの問題点を解決するためには、まず6. で述べたような回路の切り出し、はめ込み機能の追加が重要であると考えているが、この他の課題としては、

- (1) タイミング設計のサポート機能の強化
- (2) 論理合成ルール作成の効率化

等があげられる。(1)については、「指定された信号についてゲート段数を少なくする。」「回路中のタイミング的に厳しい部分のみをスピードを優先した構造に変形する。」等の機能の追加、(2)については、設計者、ライブラリ作成者側が容易に論理合成ルールを記述、検証できるようなツールの開発が必要であると考えている。今後は、さらにこれらの機能拡張についても検討していく予定である。

### [参考文献]

- [1] 西尾、宮田他“階層的ハードウェア設計言語 $H^2DL$ 言語仕様—内部仕様記述とハードウェアプロセス記述—”，情報会 設計自動化，22-2，1984.
- [2] 西尾、黒沢他“階層的ハードウェア設計言語 $H^2DL$ ”，信学全，S9-2，1985.
- [3] 増淵、村井他“ $H^2DL$ に基づく自動論理合成システムの概要”情報全，第31回，1J-1，1986.
- [4] 増淵、平川他“自動論理合成システムLUNAを用いた論理回路設計手法”情報全，第33回，3R-6，1986.
- [5] 黒沢、増淵他“自動論理合成システムLUNAの適用・評価—論理設計手法—”情報全，第34回，4F-7，1987.

- [6] 西尾、真鍋他“自動論理合成システムLUNAの適用・評価—合成回路図面生成”情報全，第34回，4F-8，1987.