

ゲート自動生成システム (ANGEL) を  
用いた回路設計に関する一評価

3R-5

池田昌夫

宮岸 修

遠藤 真

細田泰弘

NTT電気通信研究所

1. まえがき：機器に組み込むVLSIを効率良く設計するためには、回路設計に自動設計システムを適用する方策が有効と考えられる。そこでNTT電気通信研究所において開発されているVLSI設計システム ANGEL<sup>①</sup>を用いて、初心者がマイクロプロセッサ(μP)応用機器の回路を設計した場合と、同回路を数年の回路設計経験を持つ技術者が手設計した場合とを比較し、自動設計システムを用いた回路設計に関する定量的な評価を行ったので報告する。

2. ANGELを用いた設計手順：機能ブロック分割から論理ゲート生成までの回路設計の手順を図1及び表1に示す。二つのループ(ループ(a)：機能記述の見直し、ループ(b)：ゲート生成における最適化アルゴリズムの適用)による最適化が特徴である。

3. 設計対象回路の概要：設計対象とした回路は、μPを応用した機能ブロックにおけるブロック内部のμPと、機能ブロックを利用する外部μPとのインタフェース回路である。このブロック構成を図2に示す。

4. ANGELを用いた回路設計：ANGELを用いた設計手順の特徴を前提とし、論理ゲートの自動生成を効率化し、設計時間を短縮するために設定した条件を表1に示す。さらに表2に示す最適化手法を用い、自動生成されるゲート数の削減を行った。

(a)機能ブロック単位に、重複した部分や、不必要な回路部分を探索しそれらの冗長回路を生成している機能記述にフィードバックをかける。具体例として、IF~THEN(1) ELSE(2)の記述では(1)と(2)の機能を実現する回路が個別に生成されるため、(1)と(2)の間でファシリティを共用したい場合は、ELSE節を用いない等動作仕様記述の表現法を変える必要があった。また、I/Oバス制御や外部割込み制御等の

ブロックでは、入力信号の禁止条件をもとに、論理変換の妥当性を機能シミュレーション(ELSA-FX<sup>②</sup>)で確認しつつ、共用できる部分をくり出した動作記述に変更した。このように、冗長な動作を行う論理ゲートが生成される場合には、本手法が有効である。  
(b) ANGELに用意されている最適化アルゴリズム(定数を入力とするゲートや、ファンアウ

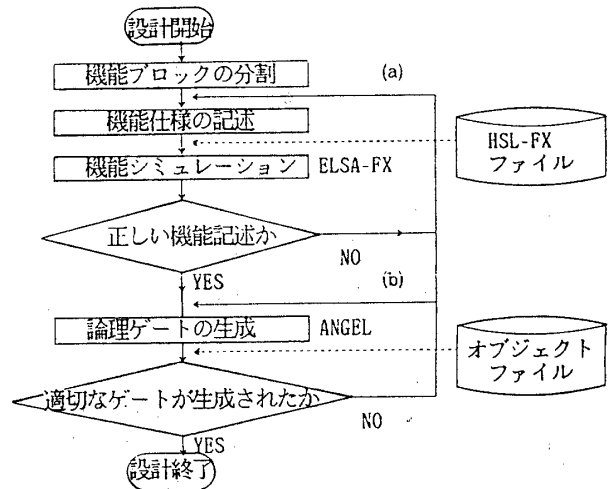


図1 ANGEL による回路設計手順

表1 回路設計の手順

設計手順	設計条件
機能ブロックの分割 ・機能の分割、設定 ・端子の設定	数百ゲート/ブロック以下で 実現できる機能に分割
機能仕様の記述 ・機能の部品化 ・可読性の向上	サブブロックの共通部品化 簡潔な機能動作表現の選択 セット条件等の記述順序を統一
機能シミュレーション ・ターンアラウンド タイムの短縮	記述の妥当性を確認するための テストブロックを作成 仮のモニタ端子による動作確認
論理ゲートの生成 ・ターンアラウンド タイムの短縮	最も下位の動作記述のレベルで 最適化の組合せを選択 生成されるゲートの妥当性を確 認するテストブロックの作成

トの無いゲートの削除等の localな最適化、論理関数のキューブ表現を利用した globalな最適化) のいくつかを組み合わせて、各々の機能ブロックのゲート数を削減する。具体例として、オーダ制御の様にデコード回路を含むブロック (HSL-FXでCASE文を用いたブロック) は、キューブ表現を用いた論理変換による globalな最適化が有効である。

5. ANGELによる設計結果: 図2に示した各機能ブロックについて論理ゲートの生成と最適化を行った。各ブロック毎のHSL-FXによる動作機能仕様の記述量、自動生成されたゲート数 (2入力NAND換算) を表3に示す。比較のため、予め手設計により得た同一機能ブロックのゲート数も同表に示す。

6. 考察: I/Oバス制御や外部割込み制御のブロックは共通回路をくり出した機能表現が行え (最適化手法②)、機能条件をそのまま論理ゲート化した手設計の約4割のゲート規模で実現できた。オーダ制御は論理変換 (最適化手法④) を用いた ANGELの最適化を適用し約半分の規模となった。同様にサブブロックの一部に ANGELの最適化 (最適化手法④) が有効であった I/Oデータ制御のブロックは約7割の規模となった。

本稿での設計対象回路は、機能動作に単純な論理式が数多く含まれ、ラッチやレジスタから構成される部分が多い等の特徴があり、ハード制御や割込み制御等のブロックは手設計の場合でも ANGELの最適化 (最適化手法③) と同等のゲートの削減が行え、ほぼ同じ規模でブロックが構成される。

一方、自動生成されたゲート数が手設計に比べ増加しているDMAバス制御ブロックを分析した結果、HSL-FXによる機能動作記述の可読性を向上させるため同期回路で機能記述した事により、非同期回路で構成した手設計よりゲート規模が増加したことがわかった。

また、これらの回路の設計から論理シミュレーションまで、手設計では3人月、ANGELを用いた自動設計では1人月と設計時間を約1/3に短縮できた。

7. あとがき: VLSI設計システム ANGELを用いた自動設計を行い、手設計による回路とほぼ同規模の回路が短期間で得られることを明らかにした。今後はさらに多くの回路設計に ANGELを適用し、その有効性の確認を進める

予定である。

最後に日頃御指導頂く宅復室 川嶋室長、中野主幹員、集応室 中島室長、唐津主幹員、星野主任員に深謝いたします。

参考文献

- ① T.Hoshino et al. "An Automatic Logic Synthesizer for Integrated VLSI Design System", Proc. CICC May 1984, pp356-360.
- ② 星野他、「混合シミュレータ ELSA-FX」、昭61信学総全大344

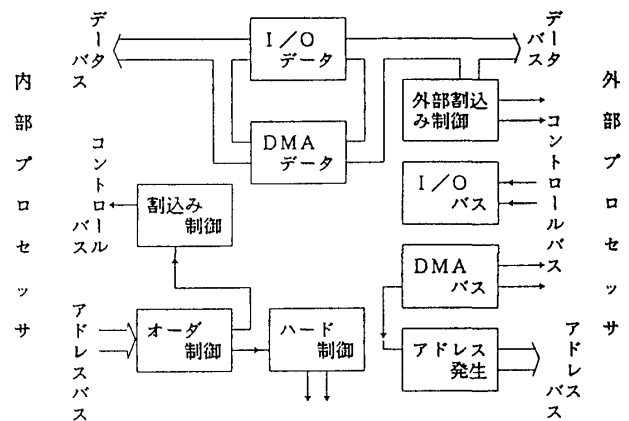


図2 対象回路のブロック構成図

表2 ゲート自動生成における最適化

手法	内容	具体例
HSL-FXによる機能動作仕様記述の最適化	回路図出力による冗長回路の発見と機能記述の修正	① WHILE文, CASE文等で不要な回路が生成される場合機能動作記述を変更する ② 入力信号の禁止条件をもとに論理変換を行い共通部をくり出す
ANGELの最適化アルゴリズムの適用	回路の性質によって最適化のアルゴリズムを選択し組み合わせる	③ 定数入力のゲートの削除等 local な最適化を行う ④ キューブ表現を用いた論理変換による global な最適化を行う

表3 設計対象回路の回路規模

ブロック名	HSL-FX 記述量(行)	ANGEL ゲート数(a)	手設計ゲート数(b)	ゲート比 (a)/(b)	表2の最適化手法
I/Oバス制御	55	1.8	4.3	40.4	①②③
外部割込み制御	21	0.6	1.4	42.9	
オーダ制御	75	1.8	3.5	49.6	①③④
I/Oデータ制御	44	4.7	6.3	73.7	①③④
ハード制御	206	11.4	11.9	96.0	① ③
割込み制御	129	15.1	15.4	97.5	
アドレス発生制御	165	32.7	31.9	102.5	① ③
DMAデータ制御	249	17.6	16.9	104.1	
DMAバス制御	287	11.0	8.0	136.7	① ③
合計	1231	96.5	100.0	96.5	

注) ゲート数(a)(b)は手設計によるゲート数の合計を1とした相対比 (%)