

## PARTHENONを適用した大規模通信用LSIの開発

東海林 敏夫    山下 正秀    中村行宏

shoji@isl.ntt.jp  
NTT情報通信研究所  
〒238-03 横須賀市武1-2356

NTTが開発した高位論理合成システムPARTHENONを適用して大規模通信用LSIの開発を行った。その結果、設計開始からLSIの試作まで、約8カ月で完了し、その回路規模は約120KGであった。また、本LSIの開発を通して判明した、大規模、高速のLSI実現へ向けたPARTHENONに対する改善要望事項を示す。

## Development of Large Scale Communication Control LSI Using PARTHENON

Toshio SHOJI    Masahide YAMASHITA    Yukihiro NAKAMURA

NTT Information and Communication Systems Laboratories  
1-2356 Take Yokosuka-Shi Kanagawa 238-03 Japan

We have developed a large scale communication control LSI using the high level synthesis system PARTHENON developed by NTT. As a result, it took about 8 months from beginning of the functional design to making a LSI chip, and the amount of logic gate was about 120KG. And furthermore, we raise some issues of PARTHENON for development of the fast and large scale LSI.

## 1. はじめに

NTTが開発した高位論理合成システムPARTHENONを大規模通信用LSIの開発に適用した結果について述べる。

本通信用LSIは、ホスト・端末間の通信プロトコル処理をアクセラレートするもので、プロセッサのコプロセッサとして位置付けられる。

対象とする通信プロトコルは、インターネット等で幅広く普及しており、現在デファクトスタンダードとなっているTCP/IPプロトコルである。

PARTHENON上の動作記述言語SFL (Structured Function description Language) を用いて本LSIの動作記述を行い、実際にLSIチップを作成した。

本論文では、試作した通信用LSIの機能概要とPARTHENONを使用した設計及びその論理合成結果を示す。

更に本通信用LSIの開発を通して、大規模、高速のLSI実現へ向け明確となったPARTHENONに対する改善要望事項等を述べる。

## 2. ターゲットシステムの概要

情報化社会の高度化とともに、TV会議やVOD (Video On Demand) といった動画像通信や、マルチメディアデータベースのようなより高速大容量のデータ通信への要求がますます高まりつつある。このような要求を実現するためネットワーク速度は、FDDIやATMなどの出現により、高速化が図られつつある。

一方、このようなネットワークの高速化に対して、通信処理性能は必ずしも高速化されているとは言えず、上記高速ネットワークを充分使い切るためには、通信プロトコル処理性能がネックとなっている。

ネック解消策としては、高速プロトコルの新規開発あるいは既存 (TCP/IP) プロトコルの高性能実装がある。

ここでは、後者の方向でTCP/IPプロトコル処理の高速化をねらったATM通信ボードを試作し、その一環として同ボード上の汎用プロセッサのコプロセッサとし

て動作し、TCP/IPプロトコル処理をアクセラレートする通信用LSIを試作した。図1にATM通信ボードのブロック構成及び本LSIの位置付けを示す。

図2に本LSIのブロック構成概要を示す。

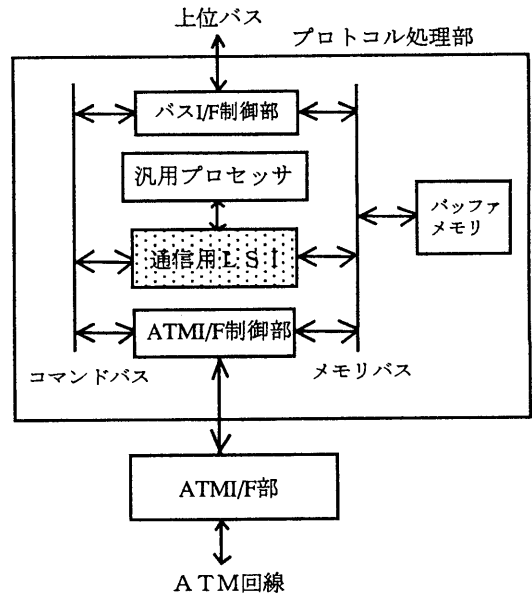


図1 ATM通信ボードブロック構成

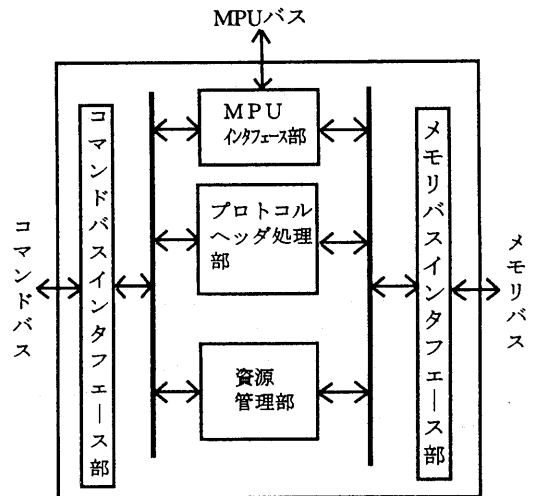


図2 通信用LSIブロック構成

### 3. PARTHENONによる通信用LSIの設計

#### 3.1 設計システム

本LSIの製造テクノロジーは、VLSIテクノロジー社の0.6 $\mu$ mCMOSゲートアレイである。それに伴い、PARTHENONのネットリストを受けてLSIレイアウト等の下流工程を担当するEDAツールには、Compass Design Automation社のDesign Navigator (以下Compass Toolと称する)を用いた。今回使用したPARTHENONのバージョンは、2.2.2、プラットフォームはSPARC Station10 (実装メモリは224MB)である。

PARTHENONでは、機能検証～論理合成～テクノロジー・マッピングまでを行い、それ以降は、ネットリストのフォーマット変換を経てCompass Toolに移管した。

図3にCompass Tool移管後の工程を示す。

Compass Toolにおいて、PARTHENONが出力したネットリストに対して、再度論理最適化を行っている。これは、Compass Toolの方がよりテクノロジーを意識したマッピングが実行可能な上に、遅延制約等をより細かく設定する必要があったためである。

#### 3.2 SFLの構成

SFLは、本LSIのハードウェア設計仕様に基づいてハードウェア構成に近い形でモジュール分割して構成した。従って、最上位階層モジュールは、接続記述となっているが、分割された各モジュール内部では、タスクやステートを用いて高位動作記述がなされている。このようなSFL構成としたのは、多人数で設計する際のインタフェースを統一化し易く、最上位階層の保守が容易となるためである。

本LSIは、多数の双方向内部バスを持っている。双方向バスをPARTHENONにより合成した場合、バス・クロックが必要になるなど、回路上のオーバーヘッドが大

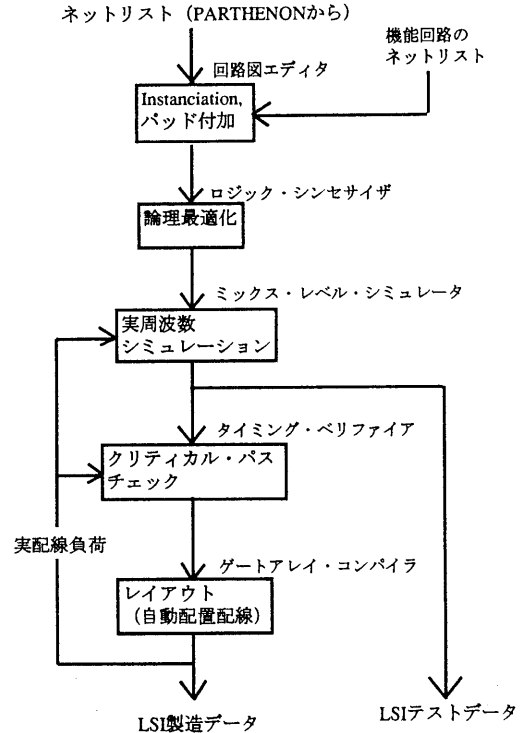


図3 Compass Toolでの設計工程

きくなる。更に、階層毎に双方向バスのトランシーバが合成されるため、設計者のイメージと合わなくなる。即ち、目標とする電力、動作周波数を達成するためには、LSI内部双方向バスは、設計上注意すべき構成要素の一つとなる。そこで、今回は双方向バスを機能回路として最上位階層に配置することにより、適切なモジュール間接続を実現した。更に、データバス部分については、SFLが提供する演算子を用いて合成した場合、速度、論理規模とも満足できる結果を得ることが困難であると予想されたため、これらについても予め用意されていた特定テクノロジー(VLSIテクノロジー社0.6 $\mu$ mCMOSゲートアレイ)のデータバス・ライブラリ(演算器マクロ等)を用いて機能回路記述を行った。これらの機能回路は、Compass Tool移管後にその実体

をマージした。

### 3. 3 論理合成結果

表1にモジュール別の論理合成結果を示す。

OPTMAP後のゲート数が約14万5千ゲート、Compass ToolでのOptimizer後で、約12万4千ゲートであった。OPTMAPにおいては、Compass ToolのOptimizerをかけることを前提としていたために、複合ゲートへのマッピングをするためのスクリプトを記述しなかった。このため、Optimizer後の方が、総ゲート数が少なくなったものと思われる。

一方、逆にOptimize後の方がゲート数が多くなったモジュール（モジュール12）がある。これは、目標とする性能を満たすための最適化により、ゲートを並列化して高速化を図ったため、結果的にゲート数が増加したためである。

なお、本LSIの開発に要した期間は、設計仕様の検討からLSIチップの試作完了まで、約8カ月であった。

## 4. PARTHENONへの改善 要望事項

本通信LSIの開発にPARTHENONを使用した結果、大規模・高速のLSI実現に向けて以下に示すような改善すべき事項が明確となった。

### 4. 1 算術演算子の直接合成

四則演算やビットシフトなどの演算子も論理合成の対象とする。この場合、単純に論理式から基本ゲートで合成するのではなく、ターゲットライブラリに用意されている多ビットの加算器セルを使用することが望ましい。これは、基本ゲートで構成されたものは、専用セルで構成されたものに比べて、速度、面積が劣るためである。

### 4. 2 エラー表示内容の詳細化

SFLの文法チェックの際に、エラー内容を詳細に表示し、かつエラーを複数箇所

表1 論理合成結果

モジュール名	OPTMAP後	Optimize後	圧縮率*
モジュール1	5,100	4,800	0.94
モジュール2	20,000	15,800	0.79
モジュール3	13,200	9,200	0.70
モジュール4	4,100	3,200	0.56
モジュール5	14,500	12,600	0.87
モジュール6	4,700	4,300	0.91
モジュール7	30,400	26,000	0.86
モジュール8	11,600	9,300	0.80
モジュール9	17,500	15,200	0.87
モジュール10	8,100	7,400	0.91
モジュール11	2,700	2,400	0.89
モジュール12	2,700	3,300	1.22
モジュール13	10,100	10,100	1
合計	144,700	123,600	—

\*:圧縮率=Optimize後ゲート数  
/OPTMAP後ゲート数

まとめて表示する必要がある。

ひとつずつ修正していくのでは、めんどろであり効率的でない。

### 4. 3 大規模・高速へ向けた合成系の 能力向上

論理合成系の能力不足により、大規模・高速の要求に応えられていない。回路が大規模になると、マシンのリソースがネックとなり、チップ全体のネットリストに対してOPTMAPを適用することができない。

このため、モジュールして処理せざるを得ない。しかしながら、モジュール分割した

場合には、要求された速度を満足するためには、チップの動作周波数を入力するだけでは不十分で、モジュールの各信号毎に遅延時間を細かく制御してやる必要がある。

現状では、このようなクリティカルパスのチューニングが出来ないため、大規模、高速になった場合、PATHENONの合成系だけでは、世の中の標準の性能（対プロセス性能）を達成することが困難となっている。特にPin-to-Pin Delayの指定による遅延速度アップが必要となる。

#### 4. 4 論理合成パラメータのSFL内記述

論理合成を制御するパラメータは、OPTMAP時のスクリプトファイルではなく、SFLソースファイル中に記述できる方が便利である。

設計者（SFL記述者）は、記述しながら各信号の遅延値が想像できるからである。

論理合成の制御パラメータ例を以下に示す。

- ・ 仮想配線容量
- ・ 動作周波数（最大遅延時間）
- ・ 外部入力信号のドライブ能力
- ・ 入力ピンのファンインの制限
- ・ 入力ピンのクロックからの遅延時間
- ・ 全信号の最大遅延時間（ランプディレイ）
- ・ 出力ピンの外部負荷
- ・ 出力ピンのクロックからの最大遅延時間
- ・ Pin-to-Pinの最大遅延時間

#### 4. 5 クロックツリー合成

4. 3項で示したようにチップ全体に対するOPTMAPの適用が困難な場合、各モジュールに対してクロックスキューを制御する手法が求められる。ファンアウトに基づく単純なツリー分割では、回路が大規模化した場合、クロックスキューの問題が大きくなる。このような問題に対応するためには、設計者に予めきめ細かなクロックツリー合成の制御を指示させることが望ましい。具体的には、ツリーの段数、バッ

ファサイズ、最大ファンアウト数等を指示させるようにすべきである。

また、配線容量、配線抵抗はレイアウトのフロアプラン後に決定される値であるが、これらを最初のクロックツリー合成へ取り込む（フィードバックする）方法が必要である。ただし、これらの値から合成をやり直すのではなく、バッファサイズの変更のみで対応できるような、余裕を持ったツリー構成を最初に生成できるようにする必要がある。

#### 4. 6 大規模へ向けた効率的共通項括りだしアルゴリズムの開発

これは、他の論理合成ツールにも言えることかも知れないが、一般に、論理ゲートを圧縮しすぎて、逆に配線を増やしてしまうことがある。

例えば、アドレスデコーダの場合（図4参照）は、アドレス線をモジュール内に分布させて、デコード出力が必要となる位置にゲートを配置しておいた方が、共通項を括り出してその共通項出力を各位置まで分配するよりも、チップ上での面積が少なくできることが良くある。

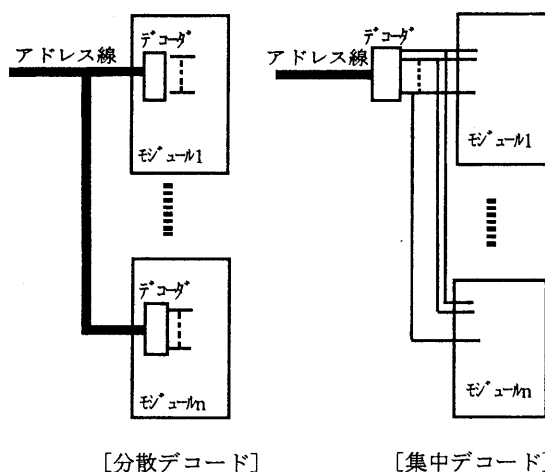


図4 共通項括りだし(アドレスデコーダの例)

ゲートアレイやスタンダードセル方式のASICの自動配置配線では、ゲート数を削減するよりも配線数（信号線の総数）を削減した方が面積効率が良い。従って、共通項の括りだしにおいては、例えば括りだしを行うためのグループ作成（モジュール内を2000～3000ゲート位にグルーピングし、グループ内だけで括りだしを行う）等の効率的な共通項括りだしアルゴリズムの開発を行う必要がある。特にこのような点でPARTHENONと他の論理合成ツールとの差別化が可能になる。

## 5. まとめ

本論文では、大規模通信用LSIの開発に、高位論理合成システムPARTHENONを適用した結果について述べた。結果的には、100Kゲートを越える規模の論理合成に成功したが、最終段階までPARTHENONだけを適用した訳ではなかった。それは、LSIが大規模、高速になった場合、現状のPARTHENONの論理合成系の能力では、目標とする条件を満たすことが、困難であると予想されたためであり、論理最適化に他社のToolを使用した。しかしながら、今後、4章で示した要求項目について、その実現を図っていけば、更に大規模、高速へ対応できる唯一の高位論理合成システムとして発展していくことが大いに期待できる。

## 謝辞

今回のLSIの開発に対して、SFL記述からLSIチップ製造まで、一貫して携わって頂き、更にPARTHENONに対する要望について有益なご意見を頂いた（株）LSIシステムズ森田部長始め関係諸氏に深謝致します。

## 参考文献

- [1] NTTデータ通信株式会社：  
” PARTHENON User's Manual”
- [2] NTTデータ通信株式会社：  
” PARTHENON Reference Manual”
- [3] RFC791 Internet Protocol
- [4] RFC793 Transmission Control Protocol
- [5] RFC792 Internet Control Message Protocol
- [6] RFC1483 Multiprotocol Encapsulation over ATM Adaptation Layer