インタプリタ・コプロセッサのLSIの設計

9L−8

泉 哲・平井 輝久・金本 好司・板野 肯三
筑波大学

1 はじめに

解析木インタプリタCPATIE-0のレジスタ転送レベルの設計[1][2]をもとに、高機能なLSI設計用のCADであるシリコンコンパイラGenesil[3]を使用し、LSIの設計を行なった。このシリコンコンパイラGenesilは、ハードウェアモジュールの機能レベル記述からLSIのレイアウトを自動的に作成するツールであり、ALU、RAM、PLA、乗算器、シフトレジスタなどのプリミティブを使って設計する。これらのプリミティブは2相クロックで動作しているので、クロック属性を念頭に置きながら設計を進めなければならない。以下シリコンコンパイラGenesil上でのチップ作成における一連の作業に従い、CPATIE-0のLSI設計について説明し、性能の評価を行なう。

2 CPATIE-0のLSI設計

Genesil上で一つのチップを設計するには、チップを構成するオブジェクトを幾つかのモジュールに分割し、そのモジュールをさらに階層化していくというモジュール設計の概念がサポートされている。各モジュール設計の過程には、大きく分けて、モジュール毎のパラメータ定義とモジュール間の信号を接続する論理設計、モジュールのシリコン上の配置とモジュール間の配線をする物理設計、そしてタイミング解析や機能シミュレーションを行ない性能や動作を確認する作業がある。

2.1 CPATIE-0の論理設計

CPATIE-0を構成するユニットにはTCU、RCU、BCUの3つがあり、それぞれレジスタ部、スタック部、シーケンサ部とインタフェースレジスタ部に分けることができる。但し、BCUにはスイッチ部とシーケンサ部のみ存在する。

(1) TCU

TCUのレジスタ部であるPC、NODE、OPERAND 1は、32ビット幅のパラレルデータパスで実現されており、NODEはncode(8ビット)、child1(12ビット)、child2(12ビット)に分割され、出力される。PCとOPERAND 1は、child1やchild2とスタック部からの入力を選択するマルチプレクサを備えている。これら3つのモジュールへのデータ転送は、制御信号が発せられてからデータ入力までを1クロックで終るようにしている。

TCUのスタック部であるPC_STACKは1ワード13ビットで128ワード長のRAMで、TSPは8ビット幅のパラレルデータパスで実現されており、TSPはスタックポインタの増減を行なう加減算器を備えている。PC_STACKにデータをプッシュ、ポップする際、TSPの増減を行なう必要があるので、スタック操作の終了には2クロック必要である。

TCUのシーケンサ部であるCTABLEとTCU_CTRはPLAで実現されており、CTABLEとTCU_CTRは入力ラッチを持つことによって、CTABLEの出力を受けとるTCU_CTR間でクロック属性を合わせている。CTABLEは33タームであり、TCU_CTRの状態遷移数は10の、74タームから成り立っている。

TCUのインタフェースレジスタ部であるNCODEとOPERANDは、RSフリップフロップによりハンドシェイクの制御を行なっているランダムロジックブロックで実現されている。NCODEは8ビット、OPERANDは32ビットのデータレジスタを保持しており、このブロックは送り側と受け側のデータアクセスを非同期に行なわせることを許しているので、お互い独立なクロックで動作し、ハンドシェイク方式でデータ転送できる。

(2) RCU

RCUの論理設計方針もTCUとほぼ同じなので、モジュールのパラメータ値と異なる部分のみを説明する。

RCUのレジスタ部のMPC、OPERAND2とRPOは、それぞれ10ビット、32ビット、32ビット幅のパラレルデータパスで実現されており、入力側にマルチプレクサを持っている。またRPOとOPERAND2の値は加算器に入力されており、両者の加算値が再びOPERAND2に入力されるというマイクロ命令も1クロックで実行できる。

スタック部のMPC_STACKは、1ワード11ビットで128ワード長のRAMで実現されており、TCU同様、スタック操作に2クロック要する。

シーケンサ部のETABLEとRCU_CTRも入力ラッチを持つPLAで実現されおり、ETABLEは249タームであり、RCU_CTRの状態遷移数が11の、35タームから成り立っている。

インタフェースレジスタ部のCONDもハンドシェイク機能を持つ、1ビットのランダムロジックブロックで実現されている。

(3) BCU

BCUのシーケンサ部であるBCU_CTRもTCU、RCUと同様に設計されており、状態遷移数が6の、27タームから成り立っている。

BCUのADDR_SWとDATA_SWは32ビット幅のパラレルデータパスで実現されており、ADDR_SWは入力にマルチプレクサを持ち有効なアドレスを選択している。DATA_SWはデータパスの出力側の2つのパスをトライステートに設定し、パスへの入力イネイブル信号を排他的に動作させスイッチの動作を行なっている。

また単相クロックで動いているSPARC[4]とのデータの受渡しは、SPARCがハンドシェイク機能を持たず入力クロックに同期してるため、RCUの2相クロックの一方のフェーズをSPARCに入力し、SPARCからのデータのクロック属性をもう一方のフェーズと判断し行なっている。さらにSPARCがメモリアクセスを行なう際、SPARCのアドレスパスとデータパスは同時には有効にならず、データパスがアドレスパスの1クロック後に有効になるので、GenesilのRAMにタイミングを合わせるために、アドレスパスをラッチに通しメモリに接続している。

しかしデータパスは、Genesilとのロードとストアに半クロックのタイミングの狂いが出ているので、ロード時のデータパスはラッチを通して、ストア時のデータパスはラッチを通さずに接続している。

2.2 CPATIE-0の物理設計

Genesil上での物理設計は、ブロックコンパイル、フロアプラン、ルーティングの3つの段階に分けられる。

ブロックコンパイルとは、LSIを構成する各モジュールの最下位レベルのブロックについて物理的なレイアウトを生成することである。ブロックコンパイルは論理設計時に定義されたモジュールの設定値をもとに、ブロックコンパイラが完全に自動的に行なうので設計者の作業は存在しない。ブロックコンパイルが終了すると次に行なう設計はフロアプランであり、シリコンのチップを想定した長方形の領域に複数のモジュールを配置することを行なう。物理設計の最終作業はルーティングであり、Genesilがフロアプランで配置されたモジュール間の配線を行なう。このルーティングは信号線の通路を確保するために、モジュール間の間隙が広げられ、フロアプランで設計したものとはかなり違った配置結果が生じる。

このようにGenesil上の物理設計で、設計者に自由度が与えられている設計はフロアプランのみであり、ルーティングの結果生成されたモジュールの面積が小さくなるように設計を行なわねばならない。このため各モジュールの信号線の接続にも注意を払いながらフロアプランを進め、ルーティング後の結果と確かめながら対話的に作業を繰り

LSI Design of an Interpreter Coprocessor
Satoshi Izumi,Teruhisa Hirai,Yosiji Kanamoto,Kouzou Itano
University of Tsukuba

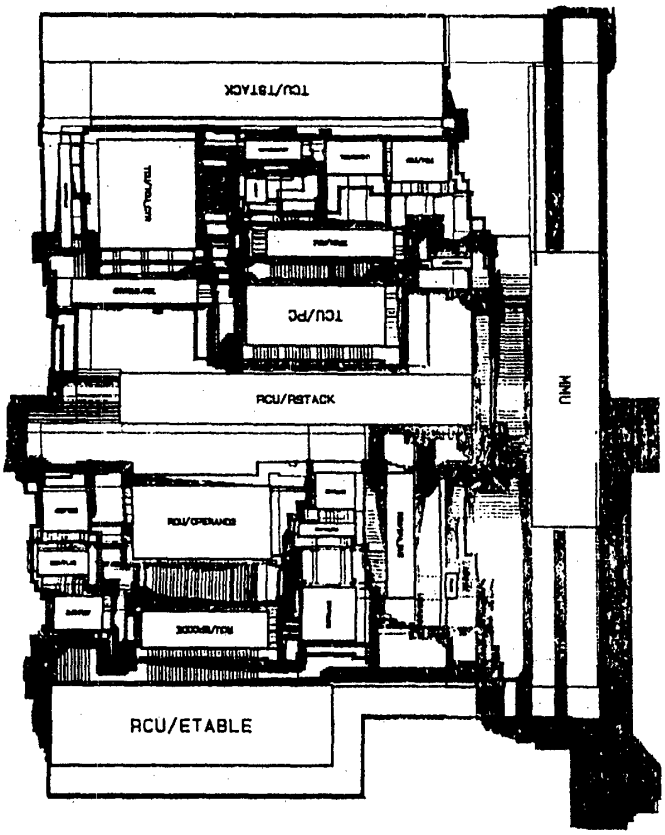

図 1: 構文木インタプリタ CPATIE-0 のレイアウト

返さねばならない。また物理設計はモジュールの階層と密接な関係があり、モジュール設計の階層を深くし単位モジュール数を少なくすると、フロアプラン時に最適な結果を容易に見つけ出せ局所的な最適値が求められるが、上位のモジュールに行くにつれ細かい調整ができず、チップレベルの全体としては逆効果になる恐れがある。

このようなことを踏まえ CPATIE-0 は、チップレベル、ユニットレベル、ブロックレベルの 3 つの階層に分け、物理設計を行なった。ユニットレベルには TCU、RCU、BCU の 3 つのモジュールがあり、それぞれのブロックレベルにはスタック、テーブル、インタフェースレジスタなどのモジュールがある。TCU のモジュール数は 1 1、RCU は 1 2、BCU は 4 であり、各ユニットのモジュールで、大きな面積を占有するのはスタックとテーブルであり、TCU では PC_STACK、PC、TCU_CTR が、RCU では MPC_STACK、ETABLE、OPERAND2 が、BCU では ADDR_SW、DATA_SW が各ユニットの半分以上を占めていた。チップレベルのフロアプランの領域が正方形に近く設計できるように、ユニットレベルのフロアプランでは、なるべく細長い長方形に配置した。そのために各ユニットで基準になるモジュールは、TCU、RCU ではスタックが、BCU ではスイッチであり、特に BCU は信号線の数が少ないため、かなり細長い長方形を実現できた。それぞれのフロアプランのサイズは、TCU が 9 8 * 1 7 7 (mils)、RCU が 1 2 4 * 1 7 7 (mils)、BCU が 3 4 * 1 4 3 (mils) になり、ルーティング後のサイズは、TCU が 1 5 6 * 2 0 5 (mils)、RCU が 1 9 3 * 2 2 6 (mils)、BCU が 3 4 * 1 4 3 (mils) になった。この結果をもとにチップレベルでのフロアプランは、TCU と RCU の長辺同士を隣接させ、BCU を TCU と RCU でできた長方形の長辺に合わせて配置した。最終的に仕上がったチップレベルのフロアプランのサイズは 2 6 1 * 3 5 0 (mils) になり、ルーティング後は 3 3 1 * 4 2 4 (mils) になった。

3 設計評価

図 1は CPATIE-0 のレイアウトを示しており、表 1は PATIE-0[5][6] とのフロアプラン面積の比較を示している。TCU については、OPERAND1 の追加にも関わらず、オブジェクトの配置や信号回線の配線が効率良く設計でき、約 4 0 %ほど縮小することができたが、RCU は予想外に大きくなった。その原因と思われるのは、ETABLE のターム数が CMEM のターム数の約 3 . 4 倍に増えたこと、オブジェクトの数が 9 から 1 2 に増えたこと、マイクロ命令を積むスタックの容量を 2 倍に増やしたこと、OPERAND2 に演算機能を付けたことなどが挙げられる。しかしチップレベルで面積を比べると、PATIE-0 面積の半分以下に押えることはできなかったものの、BCU の影響も少なく約 4 0 %の縮小が実現された。このことからオブジェクトの数や大きさより、オブジェクトの配置や信号回線の配線にチップ面積に多大な影響を及ぼすことがわかった。

表 1: FLOORPLAN のサイズの比較

単位 : *mils* ($mils^2$)

	PATIE-0	cpatie-0	対比
TCU	193 * 265 (51145)	156 * 205 (32136)	1 : 0.63
RCU	121 * 147 (17787)	193 * 226 (43618)	1 : 2.45
チップ	452 * 505 (228260)	331 * 424 (140344)	1 : 0.61

表 2: 動作クロックの最小周期

単位 : *nsec*

	PATIE-0	CPATIE-0	対比
TCU	54.2	38.7	1 : 0.71
RCU	32.1	42.6	1 : 1.33

表 2には各ユニットの動作に必要とされるクロックの最小周期を比較している。制御信号を出す PLA の状態数を 3 分の 1 に減らしたことにより TCU は速くなったが、RCU は PLA の状態数を 3 分の 1 に減らしたにも関わらず、ETABLE の状態数に影響されかなり遅くなった。まだ個々のモジュールの動作をシミュレーションしている段階なので、CPATIE-0 全体の機能シミュレーションは行なっていない。しかし TCU、RCU も状態遷移の状態数を減らしたことにより、ノード毎に必要とされるクロック数が減少すると思われるので、全体の実行速度は気になるほどには低下しないと予測される。

4 おわりに

PATIE-0 の約半分の面積を占めている DU を既存の CPU に置き換え、PATIE-0 のチップ面積の縮小化を計るとともに、さらにコプロセッサ向きのハードウェア構文木インタプリタ CPATIE-0 として再設計を行なった。CPATIE-0 のチップ面積の縮小化が実現されたが、まだモジュール配置には問題があり、PATIE-0 チップの 4 0 %までに縮小可能が期待される。また CPATIE-0 をコプロセッサとして位置付け、新たな構文木インタプリタのアーキテクチャをハードウェア設計し、その実現性を確かめることができた。

今後、より上位の高級言語へのコプロセッサ向きハードウェア構文木インタプリタを設計する際、貴重な参考資料になると思われる。

参考文献

[1] 平井 輝久, “構文木インタプリタのコプロセッサ向きアーキテクチャの設計”, 筑波大学情報学類 卒業論文 (1991).
[2] 平井 輝久, 金本好司, 板野肯三, “コプロセッサインタプリタの設計”, 情報処理学会第 4 6 回全国大会, 9L-07, (1993).
[3] Silicon Compiler Systems Corporation, “Genesil System Compiler Library,Vol.1,2,3”(1989).
[4] Cypress Semiconductor Corporation, “SPARC RISC USER'S GUIDE”,(1990).
[5] 酒井 仁, “構文木インタプリタの LSI 向き構成法”, 筑波大学工学研究科 電子情報通信学会論文, CPSY91-3 (1991).
[6] 酒井 仁, “構文木インタプリタの LSI 向き構成法に関する研究”, 筑波大学工学研究科 修士論文 (1991).