社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－SLDM－134／2008－EMB－8 (22)
2008 / 3 / 28

非同期式設計によるFPGA向けIEEE754準拠単精度浮動小数点除算器

廣本 正之† 越智 裕之† 中村 行宏††

† 京都大学大学院 情報学研究科 通信情報システム専攻, 〒606-8501 京都市左京区吉田本町
†† 立命館大学 総合理工学研究機構, 〒525-8577 滋賀県草津市野路東 1-1-1
E-mail: †reconf@easter.kuee.kyoto-u.ac.jp, ††y-nakamr@fc.ritsumei.ac.jp

あらまし 非同期式回路は同期式回路と比較し，高性能かつ低消費電力な回路が実現できるとして期待されている．また，特定のクロック周波数に対する最適化を行うことなく高性能な回路を実現できるため，再利用性の高いIPコアを実現可能である．本研究ではこのような非同期式回路に着目し，特にプロトタイピングデバイスとして頻繁に利用されているFPGAを対象とし，高性能かつ再利用性に優れた非同期算術演算器を提供することを目的とする．本稿では対象回路としてIEEE754準拠単精度浮動小数点除算器を取り上げ，これの非同期設計を行った．提案非同期除算器は，様々な動作周波数の同期式システムに組み込み可能な外部インタフェースを備えつつ，内部は外部システムとは非同期に高速かつ低消費電力に動作する構成とした．また本稿ではXilinx社Virtex-4 FPGAを対象とし，非同期式除算器の論理合成，配置配線を行い，同期式回路と性能比較を行った．その結果，提案非同期式除算器は同期式回路に比べ，面積，消費電力，スループットにおいて優れた性能を達成することを示した．
キーワード 算術演算回路，IP再利用性，低消費電力，減算シフト型除算器

An Asynchronous IEEE754-standard Single-precision Floating-point Divider for FPGA

Masayuki HIROMOTO†, Hiroyuki OCHI†, and Yukihiro NAKAMURA††

† Dept. of Communications and Computer Eng., Graduate School of Informatics, Kyoto Univ., Yoshida-Honmachi, Sakyou-ku, Kyoto 606-8501 Japan
†† Research Organization of Science and Engineering, Ritsumeikan Univ., Noji Higashi 1-1-1, Kusatsu, Shiga 525-8577 Japan
E-mail: †reconf@easter.kuee.kyoto-u.ac.jp, ††y-nakamr@fc.ritsumei.ac.jp

Abstract *Synchronous* design methodology is widely used for today's digital circuits. However, it is difficult to reuse a highly-optimized synchronous module for a specific clock frequency to other systems with different global clocks, because logic depth between FFs should be tailored for the clock frequency. In this paper, we focus on *asynchronous* design, in which each module works at its best performance, and apply it to an IEEE754-standard single-precision floating-point divider. Our divider is ready to be built into a system with arbitrary clock frequency and achieves its peak performance and area- and power-efficiency. This paper also reports an implementation result and performance evaluation of the proposed divider on a Xilinx Virtex-4 FPGA. The evaluation results shows our divider achieves smaller area, lower power consumption, and higher throughput than the synchronous dividers.
Key words Arithmetic operation circuit, IP reusability, low power design, digit-recurrence divider

1. 序 論

近年VLSIの設計においては，設計の容易さや各種CADツールが整っていることもあり，単相同期式システムが主流となっている．しかしながら同期式システムにはいくつかの問題点がある[1]．1つは，高い周波数のグローバルクロックをチップ全域に分配する必要があるため，そのクロックツリー自体や末端に接続されているフリップフロップ群によって消費される電力が非常に大きいことである．これは，システムによっては全体の45%に及ぶとの報告もある[2]．2つ目は，チップ内での製造ばらつきへの対応が困難なことが挙げられる．これは，プロセスの微細化に伴うトランジスタのばらつきによってもたらされ

た，クロックスキューの増大によるものである．このために設計マージンを大きくとる必要性が生じ，プロセスの性能を最大限に発揮することが難しくなってきている．この問題に対し，製造後にばらつきを緩和する回路技術 [3] や設計方法 [4]，局所的には同期設計し大域的には非同期設計するアプローチ [5] などが提案されている．3つ目として，同期システムでは設計対象のクロック周波数毎に回路の最適化が必要であることが挙げられる．高スループットな同期式回路を実現するためにはクロック周波数に応じ，レジスタ間の組み合わせ回路の段数を最適化しなければならない．このためには，回路記述の段階でレジスタ挿入を最適に行う必要がある．しかし上流工程で配置配線後の遅延時間を精密に見積もることは困難であり，高スループットな回路の実現のためには相当の工数をかけて回路記述と遅延解析の繰り返しを行う必要がある．越智らにより開発された IEEE-754 準拠の同期式単精度浮動小数点除算器ライブラリ [6] では，動作周波数別に最適化された HDL 記述を提供しているが，このようなライブラリの開発には多大な工数を要する．

これらの問題を解決するべく，我々は非同期式システムに注目する．非同期式回路の主な特徴としては以下のものがある [7]．

（1） 低消費電力である

（2） クロックスキューの影響を受けない

（3） グローバルクロック周波数に依存した最適化が不要

（1）は主にクロック分配系による消費電力が削減できることによる．この特徴から，低消費電力が要求される携帯機器向け LSI への採用実績がある．（2）は，非同期式システムではグローバルクロックを用いないため，長距離配線が減りクロックスキューの影響をほとんど受けないことが期待される．（3）は，各モジュールがグローバルクロックに同期して動作する必要がないため，モジュール内の回路はグローバルクロックとは独立に設計可能なことによる．各モジュールは最も性能を発揮できる最適な回路構成を取ることができる．

本研究では，これらの非同期式回路の利点に着目し，グローバルクロック周波数に依存しない高性能な演算器の設計資産，IP コアの提供を試みる．ここでは，越智ら [6] が開発した IEEE-754 準拠の同期式単精度浮動小数点除算器を取り上げ，これの非同期化を行う．文献 [6] の設計では動作周波数に応じ，組み合わせ回路段数を調整したものを複数種類設計する必要があった．本稿では最も性能の良い構成で除算器の内部を設計し，外部とのインターフェースを非同期化することにより，1 種類の設計のみで多様な周波数のシステムに組込める除算器を提供する．

なお著者らは [8] において上記の特徴を持つ非同期浮動小数点除算器を提案しているが，本稿では提案回路のデザインフローを自動化し，より高性能な回路を得ることができたことについて述べる．また，FPGA ボードを用いた実機検証を行った結果についても併せて報告する．

本稿ではまず，2. 章で非同期式システムならびに浮動小数点除算について述べる．次に 3. 章で本稿で提案する非同期浮動小数点除算器の構成およびそのデザインフローについて述べる．その後，4. 章で FPGA 実装および提案回路の性能評価を行い，同期式と非同期式の比較を行なった結果について述べ，最後に

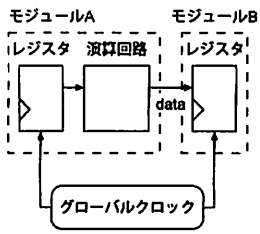

図 1 同期式データ転送

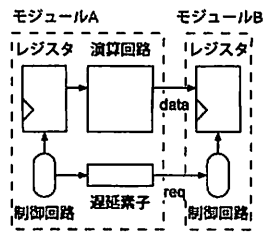

図 2 非同期式データ転送

5. 章でまとめとする．

2. 非同期化回路および単精度浮動小数点除算

本章では，まず一般的な非同期式システムについて述べる．次に今回の実装対象である単精度浮動小数点形式での除算について述べる．

2.1 非同期式システム

本稿では，回路の逐次的な動作が全て単一のグローバルクロックに同期して行われる回路を同期式回路，そのようなグローバルクロックを持たない回路を非同期式回路と定義する．

同期式回路では図 1 のように，全レジスタは単一のグローバルクロックに同期してデータの送受信を行う．一方，非同期式回路ではグローバルクロックが存在しないため，データを送受信する際に各モジュール間でタイミングを決める必要がある．その方式として，データと別にタイミング信号を持たせるハンドシェーク方式や，データ 1 ビット毎にデータの有効・無効の情報を持たせて送る 2 線式などが挙げられる．ハンドシェーク方式は一度に多ビットのデータが送られるデータ系に適しており，2 線式は一度に 1 ビットのデータが送られる制御系に適している．本稿で対象とする除算器では一度に複数ビットのデータを送るため，ハンドシェーク方式を採用する．ハンドシェーク方式では図 2 のように，送信側のモジュール A はデータとは別に req 信号を送信する．モジュール B は req 信号を受け取ると，データ信号線上のデータをレジスタに取り込む．データがモジュール B に到着するまで req 信号を遅らせる必要があるため，req 信号線には遅延回路が設けられる．

2.2 単精度浮動小数点除算

2.2.1 単精度浮動小数点形式

本研究で扱う浮動小数点数は，IEEE-754 準拠の単精度浮動小数点形式とする．単精度浮動小数点数で表わされる数値は，1 ビットの符号フィールドを s，8 ビットの指数フィールドを e，23 ビットの仮数フィールドを f とすると，下の式の通りである [9]．

$$R = (-1)^s \times (1.f) \times 2^{(e-127)} \qquad (1)$$

このような形式により，漸進アンダーフローを行わない場合でも，絶対値が 2^{-126} から $(1-2^{-24}) \times 2^{128}$ までの広範囲な数値を表現できる．

2.2.2 単精度浮動小数点除算の手順

図 3 に浮動小数点除算回路のブロック図を示す．浮動小数点除算の大まかな流れは次の通りである．

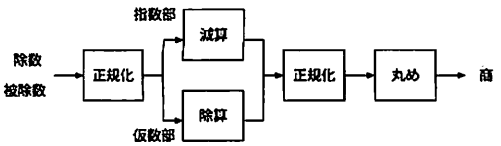

図 3　浮動小数点除算回路のブロック図

X=01111
Y=01011
-Y=10101

01111　X
+ 10101　-Y
100100 … q_0=1
01000
+ 10101
11101 … q_1=0
10000
+ 10101
100101 … q_2=1

図 4　引き戻し法アルゴリズム

まず，回路に除数と被除数を入力する．それが非正規化数であった場合は，バレルシフタを用いて正規化する．次に，仮数部同士の除算と指数部同士の減算を行う．その後，正規化，丸めを行い，商を出力する．この他に，商の符号を生成したり除数が 0 であった場合等の例外処理を行う仕組みも必要である．

2.3　除算アルゴリズム

浮動小数点除算の中で，仮数部の除算を行うアルゴリズムについて述べる．

除算アルゴリズムは，大きく乗算型と減算シフト型に分類される．本稿では，回路規模が小さく抑えられる減算シフト型の引き戻し法を取り上げる．以下，n 桁の減算シフトを行うものとし，被除数 X，及び除数 Y は正規化されているものとする．また，商を $Q = \sum_{j=0}^{n-1} q_j 2^{-j}$ とし，q_j を決定した後の部分剰余を R_{j+1} とする．

引き戻し法のアルゴリズムは，除算を筆算で行う場合の手順とほぼ同じである．$j = 0$ の場合，$X \geq Y$ であれば $q_0 = 1$, $R_1 = X - Y$ とし，さもなくば $q_0 = 0$, $R_1 = X$ とする．$j \geq 1$ の場合，$2R_j - Y \geq 0$ ならば $q_j = 1$, $R_{j+1} = 2R_j - Y$ とし，さもなくば $q_j = 0$, $R_{j+1} = 2R_j$ とする操作を繰り返す．実際の数値を用いた例を図 4 に示す．丸い点線で囲まれた部分が部分剰余，四角の点線で囲まれた部分が符号ビットである．$Y = 01011$, $X = 01111$ とすると，2 の補数を用いて，$-Y = 10101$ となる．まず，$X - Y$ を計算した結果の符号ビットが 0 なので $q_0 = 1$ を立て，$R_1 = X - Y$ とする．次に，$2R_1 - Y$ 計算すると，結果の符号ビットが 1 であるため，$2R_0$ を R_1 に格納する．以下同様である．引き戻し法において必要となる主なハードウェアは，$2R_j - Y$ を計算するための減算器と，R_j に代入する値を選択するための 2 入力マルチプレクサである．

3.　非同期単精度浮動小数点除算器

本研究では，前章で述べた単精度浮動小数点除算を行う非同期式回路の設計を行う．本章では提案非同期式除算器の構成および動作について述べる．また，非同期式除算器を設計するためのデザインフローについても解説する．

3.1　全体構成

提案する非同期浮動小数点除算モジュールのブロック図を図 5 に示す．図 3 に示した浮動小数点除算を構成する各機能ブロックに対し，同期クロックを用いずにそれぞれ適切な遅延素子によりタイミングを取るようになっている．外部とは非同期のハンドシェーク方式で接続し，任意の動作周波数で動作する同期式または非同期式システムに容易に組み込むことが可能である．

なお，仮数部除算については計 24 回の減算シフトが必要であるため，n 個の減算シフト回路を用意し，$24/n$ 回反復を繰り返すことにより実現している．反復を行うためにこの部分の遅延素子はリングオシレータとして機能し，ローカルクロックを生成する．減算シフトの段数 n は回路規模とスループットのトレードオフを考慮して決定する必要があるが，ここでは [8] の検討を踏まえ $n = 2$ とした．

3.2　提案除算器の動作

提案浮動小数点除算モジュールの動作は以下の通りである．

まず外部システムのタイミングに同期して入力除数および被除数がそれぞれレジスタに格納される．仮数部および指数部レジスタに格納された値が減算や正規化の組み合わせ回路を通り，仮数部除算モジュール内のレジスタに到着した後に，トリガレジスタの信号 `req` も遅延回路 `req_delay` を通り仮数部除算モジュールに到着する．その信号を受け，仮数部除算モジュールはリングオシレータの動作を開始させ，$n = 2$ 段の減算シフト回路により 12 回の反復処理，つまり計 24 回の減算シフト演算を繰り返し，除算を行う．除算が終了すると仮数部除算モジュールは次段へのリクエスト信号 `nreq` を立て，それと同時に除算結果のデータが正規化，丸めを行うモジュールへと渡される．正規化および丸め処理が終了した後に遅延回路 `nreq_delay` を通過した `nreq` 信号が到達し，それを受けて最終的な浮動小数点除算の結果を出力レジスタに格納する．

3.3　デザインフロー

上記のような非同期除算回路の実現には，リングオシレータに用いる遅延回路 `clk_delay`，前段の処理に対する遅延回路 `req_delay`，後段の処理に対する遅延回路 `nreq_delay` がそれぞれ必要である．本研究では FPGA 向けの非同期除算回路を対象としているため，FPGA の基本構成要素である LUT を多段接続することによりこれらの遅延回路を実現した．遅延時間の調整は LUT の接続段数を変化させることにより対応可能である．

本研究ではそれぞれの遅延回路の LUT 段数を決定するために，図 6 に示すデザインフローにより遅延回路の調整を行った．まず，各遅延回路の LUT 段数を指定し，それらを含む非同期除算回路の RTL を生成する．この際，接続関係のみの指定では FPGA ベンダから提供される CAD ツールによりどのように配置配線が行われるか不明であるため，それにより遅延時間が必ずしも LUT 段数に比例しない現象が生じる．そこで本研究では相対位置制約 (RLOC) を用い，FPGA 上に LUT が一直線に並ぶように指定することにより，上記の問題を回避した．次

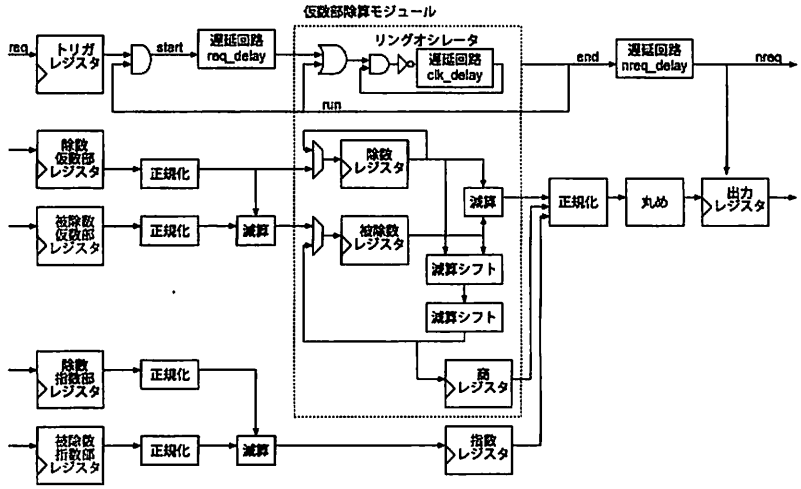

図 5　非同期浮動小数点除算器のブロック図

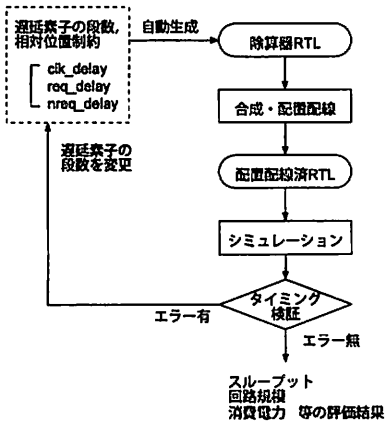

図 6　非同期除算回路のデザインフロー

に CAD ツールを用いて合成および配置配線を行い，配置配線後のタイミング情報が含まれたシミュレーション用 RTL を得る．この配置配線後 RTL を用い，複数のテストパターンに対しタイミングエラーを生じずに正しく演算が行われるかを論理シミュレーションにより検証を行う．遅延回路による遅延時間が不十分であった場合は該当箇所の LUT 段数を増加させ，十分な遅延回路が得られるまで再び合成，配置配線，シミュレーションを繰り返す．

上記の手順により適切な遅延回路を持つ除算器の探索を行うが，LUT 段数を変更する度にこれらの作業を手動で行うのは多大な工数を要する．そこで本研究では，各遅延回路を構成する LUT 段数を指定するのみで，シミュレーションによるタイミング検証までのフローを自動的に実行可能な設計環境の構築を行った．本設計環境では，指定された LUT 段数を元に LUT を縦続接続した遅延回路を自動生成し，浮動小数点除算器の該当箇所への埋め込みを行う．この際，上述の相対位置制約も自動的に付加し，適切に配置配線が行われるようにする．得られた除算器の RTL を用い，順次 CAD ツールを起動し，最終的にシミュレーション結果の検証を行う部分まで自動的に行えるようになっている．

以上のようなデザインフローに従って遅延回路を決定し，タイミングエラー無く動作する RTL が得られると，シミュレーション結果を元にそのデザインのスループット，回路規模，消費電力等の性能評価を行いレポートとして出力する．

4. FPGA への実装および性能評価

前章で述べた非同期浮動小数点除算回路を FPGA に実装し，同期式回路との性能比較を行った．また実際の FPGA ボード上に組み込み，同期式システム内においても正常に動作することを確認した．以下，それらについて述べる．

4.1 浮動小数点除算回路の性能評価

提案浮動小数点除算回路を FPGA に実装し，回路規模，消費電力，スループットの評価を行った．実装対象の FPGA としては Xilinx 社 Virtex-4 (XC4VFX12) を用いた．シミュレーションには Menter Graphics 社 ModelSim SE 6.2e を，論理合成，配置配線には Xilinx 社 ISE 9.2i を，消費電力については，配置配線後のネットリストと遅延情報を用いたミュレーションを行い，その結果を ISE に付属の消費電力解析ツール XPower を用いて見積った．

同期式設計との性能比較を行うため，比較対象として文献 [6] の同期式浮動小数点除算回路についても同様に実装を行い，性能を評価した．ただし，この同期式設計はローム社 0.35μm プロセスで 50，75，100，125，150 MHz で動作するように作られた 5 種類の回路であるが，今回実装対象とした FPGA ではその周波数で動作しないため，それぞれ 40，60，90，110，130 MHz に周波数を落として実装，評価を行った．非同期式回路については，同一の回路を上記 5 種類の周波数で動作する同期システムに組込むことを想定し，外部クロックを 5 種類に変化させて評価を行った．

同期式，非同期式の各モジュールの性能を表 1 に示す．以下，回路規模，消費電力，スループットについて同期，非同期の比較を行う．

4.1.1 回路規模

同期式，非同期式の周波数毎の回路規模を図 7 に示す．回路規模については，評価を行った全ての周波数に渡り，非同期式が同期式より小さくなっている．同期式は各周波数で減算シフト段数の異なる設計となっているため，回路規模にばらつきが見られる．それに対し，非同期のものは同一の設計を 5 種類の外部周波数で動作させているため回路規模は一定である．同期式に比べ，非同期式では回路規模を小さく抑えることができた．

4.1.2 消費電力

各設計の消費電力を評価するため，1 データの処理に要するエネルギーを求めたものを図 8 に示す．

消費エネルギーについても同期式設計には周波数により変化が見られるが，非同期式ではどの周波数においても一定である．同期式がばらつく原因については，1 データあたりのエネルギーとして正規化しているため，スループットの違いによるものである．スループットが周波数により大きく異なる理由については後述する．

非同期式設計ではグローバルクロックを用いていないため，実際にデータが通過するフリップフロップ，配線のみで電力が消費されるため，常に全ての部分がクロックにより駆動されている同期式設計より消費エネルギーを小さく抑えることができたと考えられる．

4.1.3 スループット

各設計のスループットを図 9 に示す．これまでと同じく，非同期式回路については動作周波数によらずほぼ一定の性能を示している．一方，同期式は周波数により性能のばらつきがみられる．これは同期式設計の場合，減算シフトやレジスタなどの取りうる回路構成が限られており演算に要するサイクル数が離散値となることによる．そのため動作周波数とうまく調和すれば高いスループットが得られるが，そうでなければ性能が低下してしまい，設計によりばらつきが生じることになる．

全周波数に渡り，非同期式回路が同期式回路より高いスループットを達成している．これは，同期式設計においては全体の動作周波数がクリティカルパスに律速されるが，非同期式設計では各組み合わせ回路で実質的に必要な遅延時間のみで演算が可能であることによると考えられる．同期式回路は動作周波数に対してさらに最適化することによりスループットの向上が可能であると考えられるが，上と同じ理由により必ずしも周波数と調和する回路が得られるとは限らない．

スループットと回路規模の関係を評価するため，回路規模あたりのスループットを求めたものを図 10 に示す．非同期式回路が同期式回路を大きく上回っており，面積，スループット共にバランスの取れた優れた性能を示していると言える．

4.2 FPGA ボードでの動作検証

前節の性能評価に加え，実デバイス上での動作確認のため，FPGA ボードを用いた実機検証を行った．評価に使用したボードは上の評価で用いたものと同一の FPGA である Virtex-4 (XC4VFX12) が搭載された Xilix 社の Virtex-4 評価ボード ML403 である．ボードの外観を図 11 に示す．

この評価ボードには周辺 I/O や FPGA に内蔵されている CPU PowerPC を制御するためのリファレンスデザインが付属

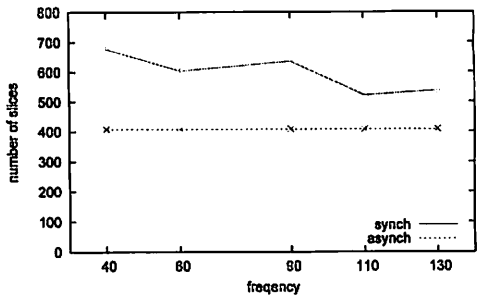

図 7 浮動小数点除算器の回路規模

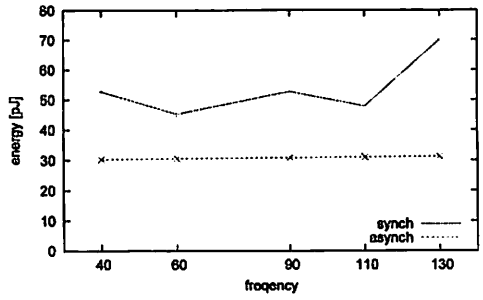

図 8 浮動小数点除算器の消費エネルギー

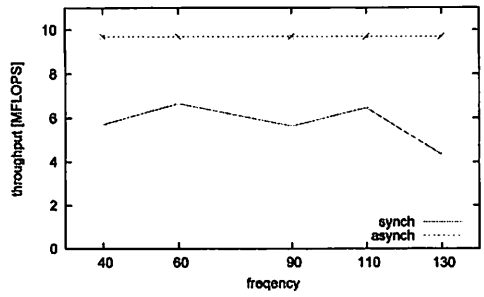

図 9 浮動小数点除算器のスループット

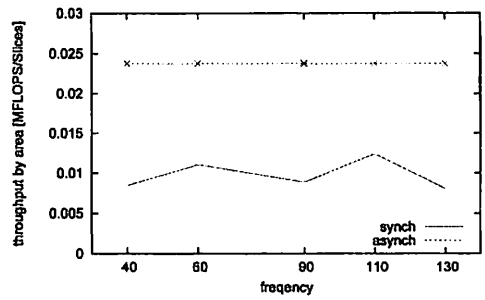

図 10 浮動小数点除算器の回路規模あたりのスループット

表 1　同期式及び非同期式浮動小数点除算回路の性能

	同期式					非同期式				
動作周波数 [MHz]	40	60	90	110	130	40	60	90	110	130
回路規模 [Slices]	677	604	635	522	538	408				
消費電力 [mW]	301	301	296	309	303	293	295	298	300	302
1 データあたりの 消費エネルギー [pJ]	52.8	45.2	52.7	47.8	70.0	30.3	30.5	30.8	31.0	31.2
スループット [MFLOPS]	5.71	6.67	5.66	6.47	4.33	9.69	9.69	9.69	9.69	9.69
スループット/回路規模 [MFLOPS/Slices]	0.0084	0.0110	0.0089	0.0124	0.0081	0.0237	0.0237	0.0237	0.0237	0.0237

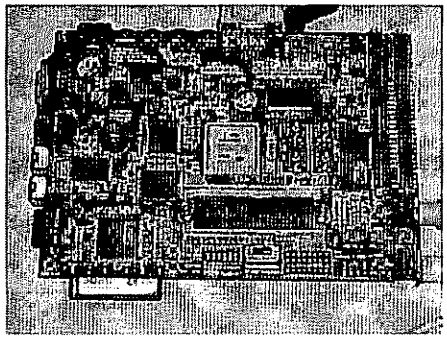

図 11　Xilinx Virtex-4 FPGA ボード

している．今回は非同期式回路の IP コアを同期式システムに組み込むことを想定し，上記のリファレンスデザインに提案非同期回路除算回路を追加して検証環境を構成した．FPGA 上の PowerPC 上で実行させたソフトウェアから非同期除算の制御，データ転送を行えるようにし，ランダムに発生させた 100 万パターンに対し浮動小数点除算を行い，演算結果を検証した．その結果，全ての演算が正しく行われ，同期式システム内で非同期式回路が正常に動作していることが確認できた．

5. 結　　論

本稿では，非同期式設計を用いることで，単一の IP コアで任意の動作周波数の同期式システムに組み込むことのできる，IEEE-754 単拠の浮動小数点除算モジュールを提案した．提案する浮動小数点除算モジュールは，外部の同期式システムとデータ入出力可能なインタフェースを持ち，内部回路は外部クロックとは独立に高速に演算を行う．非同期式回路を構成する際に必要となる遅延回路については FPGA の LUT を多段接続することにより実現し，その遅延時間の調整を行うためのデザインフローならびに自動設計環境を構築した．設計した非同期式除算回路を Xilinx Virtex-4 FPGA を対象として論理合成，配置配線し，性能評価を行った結果，5 種類の動作周波数に最適化されたそれぞれの同期式回路よりも回路規模，消費電力，スループットにおいて優れた性能を達成可能であることを示した．また，FPGA ボードを用い，CPU やその他の周辺回路から構成された同期式システム中に実際に提案非同期除算器を組み込み，正常に動作することを確認した．

今後の課題としては，自動設計フローをさらに改良し，対象 FPGA デバイスが変化した場合にも対応できるようにすることが挙げられる．また，他の種類の算術演算回路についても非同期式設計を行い，同期式回路より高性能な IP コアライブラリを構築してゆく予定である．

謝辞　本研究は一部，日本学術振興会科学研究費補助金基盤研究 (C)18500036，および，グローバル COE プログラム「光・電子理工学の教育研究拠点形成」(拠点番号 C09) による．また，本研究は東京大学大規模集積システム設計教育研究センターを通し，メンター株式会社の協力で行われたものである．

文　　献

[1] 南谷："非同期式マイクロプロセッサの動向"，情報処理，Vol39, No.3, pp. 181–186 (1996).

[2] T. Sakurai and T. Kuroda: "Tutorial on Low-Power Design Methodology", Proceedings of the Synthesis and System Integration of Mixed Technologies (SASIMI), pp. 3–10 (1996).

[3] T. Kawanami, M. Hioki, H. Nagase, T. Tsutsumi, T. Nakagawa, T. Sekigawa and H. Koike: "Preliminary Evaluation of Flex Power FPGA: A Power Reconfigurable Architecture with Fine Granularity", IEICE Transactions on Information and Systems, **E87-D**, 8, pp. 2004–2010 (2004).

[4] K. Katsuki, M. Kotani, K. Kobayashi and H. Onodera: "Extracting a Random Component of Variation from Measurement Results of a 90 nm LUT Array", Proceedings of the Synthesis and System Integration of Mixed Technologies (SASIMI), pp. 197–200 (2006).

[5] D. M. Chapiro: "Globally-Asynchronous Locally-Synchronous Systems", PhD thesis, Stanford University (1984).

[6] H. Ochi, T. Suzuki, S. Matsunaga, Y. Kawano and T. Tsuda: "Development of an IP Library of IEEE-754-Standard Single-Precision Floating-Point Dividers", IEICE Transactions on Fundamentals, **E86-A**, 12, pp. 3020–3027 (2003).

[7] J. Sparsø and S. B. Furber: "Principles of Asynchronous Circuit Design: A Systems Perspective", Kluwer Academic (2001).

[8] M. Hiromoto, S. Kouyama, H. Ochi and Y. Nakamura: "An Asynchronous Single-Precision Floating-Point Divider and its Implementation on FPGA", Proceedings of the 14th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), pp. 294–301 (2007).

[9] IEEE: "IEEE Standard for Binary Floating-Point Arithmetic", ANSI/IEEE Std 754-1985 (1985).