

デジタル変調に向けた演算機能を持つ リコンフィギャラブル・アーキテクチャ

小林 礼貴† 谷口 一徹† 坂主 圭史† 武内 良典† 今井 正治†

† 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5
E-mail: †{a-kobays,i-tanigu,sakanusi,takeuchi,imai}@ist.osaka-u.ac.jp

あらまし 近年、組み込みシステムにおいて、複数の無線通信方式の実現が重要になっている。複数の無線通信方式を実現する手段として、高性能と柔軟性を同時に実現可能なリコンフィギャラブル・アーキテクチャが注目を浴びている。本研究では、対象アプリケーションをデジタル変調とし、デジタル変調に向けた演算機能を持つリコンフィギャラブル・アーキテクチャRASKを提案する。評価実験より、提案するアーキテクチャを用いることで、使用する処理エレメント数を削減でき、また、全体の面積を約1割削減できた。

キーワード リコンフィギャラブル・アーキテクチャ、デジタル変調

Reconfigurable Architecture with Calculation Function for Shift Keying

Ayataka KOBAYASHI†, Ittetsu TANIGUCHI†, Keishi SAKANUSHI†, Yoshinori TAKEUCHI†,
and Masaharu IMAI†

† Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka Suita, Osaka 565-0871 Japan
E-mail: †{a-kobays,i-tanigu,sakanusi,takeuchi,imai}@ist.osaka-u.ac.jp

Abstract Recently, in the embedded systems, the implementation of multi wireless communications is important. As the means of implementation of multi wireless communications, a reconfigurable architecture gets a lot of attention. It has high performance and high flexibility. In this report, we propose the RASK(Reconfigurable Architecture for Shift Keying). RASK is a reconfigurable architecture with calculation function for shift keying dedicated for shift keying. Experiments show that using proposed architecture reduces the number of processing elements and reduces total area by tenth.

Key words Reconfigurable Architecture, Shift Keying

1. はじめに

近年、携帯電話、PDAなどのさまざまな小型の組み込みシステムが広く普及している。これらの組み込みシステムにおいては、無線通信機能を用いたアプリケーションが数多く使用されている。

無線通信機能を用いたアプリケーションが普及するに連れ、様々な符号化方式や通信方式、変調方式が提案されており、高速・高品質な通信を実現するため、組み込みシステムには高い処理性能が要求されている。よって、次世代のユビキタス社会における組み込みシステムにおいては、高速・高品質な無線通信を実現するための高い性能や、所望のアプリケーションに応じて無線通信方式を瞬時に切り替え可能な柔軟性を持ち、なおかつ小面積で無線通信機能を実現することが重要である。

現在、無線通信方式の実現には広くASIC(Application Specific Integrated Circuit)が使用されているが、ASICで無線通信機能を実現すると、高性能であるが柔軟性に欠けるため、複数の無線通信方式に対応することができない。一方、プロセッサやDSP(Digital Signal Processor)で無線通信機能を実現すると、複数の無線通信方式に対応できるが、高い性能を出すことができない。そこで近年、高性能と柔軟性を同時に実現可能なリコンフィギャラブル・デバイスが注目を浴びている[1]~[4]。

一般にリコンフィギャラブル・デバイスは、多数のPCE(Processing and Control Element)と呼ばれる処理エレメント、PCE間の配線、および複数のコンフィギュレーションデータを持ち、動作時にコンフィギュレーションデータの切替えを行い、所望のアプリケーションを実現する。

しかし、リコンフィギャラブル・デバイスは、柔軟性を実現す

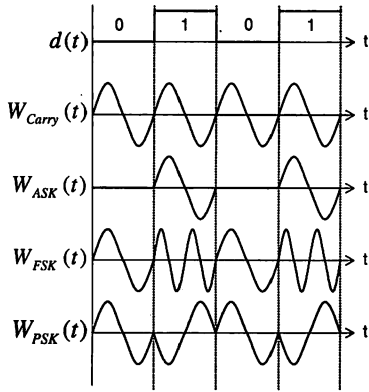


図1 搬送波と各変調波

るために搭載されている多数の PCE や PCE 同士を接続するための内部配線により、非常に多くの HW 面積が必要になるという欠点がある。

これまでに、複数の無線通信を実現するリコンフィギュラブル・デバイスが提案されている [5]。文献 [5] では複数の無線通信方式に対応できるアーキテクチャを提案しているが、携帯を目的とした組み込みシステムでの実現において、HW 面積の面で改善の余地がある。HW 面積を抑えつつ、柔軟性を発揮するためには、アプリケーションに応じて PCE の機能を適切に定め、アプリケーションに応じた最適なリコンフィギュラブル・デバイスを決定する必要がある。

そこで本研究では、デジタル変調向けの演算機能を持つリコンフィギュラブル・アーキテクチャ (Reconfigurable Architecture for Shift Keying, RASK) を提案する。RASK アーキテクチャは演算ユニットの PCE がデジタル変調向けの演算を備えており、また複数のデジタル変調に対応可能なアーキテクチャである。

以下に本稿の構成を述べる。第 2 章ではデジタル変調処理について述べる。第 3 章では RASK 構成について述べる。そして、第 4 章で評価実験について述べ、第 5 章で本研究のまとめについて述べる。

2. デジタル変調処理

デジタル変調処理は、振幅 A 、周波数 f 、位相 θ が一定の正弦波 $W_{\text{Carry}}(t)$ を搬送波とし (式 (1), 図 1), 搬送波の振幅、周波数、位相を送信データ $d(t)$ に応じて変化させ、搬送波に情報を持たせた被変調波を出力する。

$$W_{\text{Carry}}(t) = A \sin(2\pi ft + \theta) \quad (1)$$

デジタル変調は、搬送波の振幅に情報を持たせる振幅変調方式 (図 1 $W_{\text{ASK}}(t)$), 搬送波の周波数に情報を持たせる周波数変調方式 (図 1 $W_{\text{FSK}}(t)$), 搬送波の位相に情報を持たせる位相変調方式 (図 1 $W_{\text{PSK}}(t)$) の 3 つに分類できる。

2.1 直交変調器

本節では、直交変調器 [6] を用いたデジタル変調について説明し、本研究で対象とするデジタル変調方式である ASK,

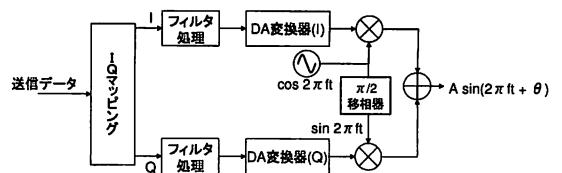


図2 直交変調器を利用した構成

$\pi/4$ shiftQPSK の IQ マッピングについて説明する。

2.1.1 直交変調器を用いたデジタル変調

図 2 は、直交変調器 [6] を用いたデジタル変調を行う回路構成を示している。直交変調器は余弦波生成器、 $\pi/2$ 移相器と乗算器で構成される回路である。直交変調器を用いたデジタル変調では、入力された送信データを IQ マッピングで I 信号と Q 信号に変換し、フィルタ処理を行った後、DA 変換される。DA 変換された I 信号と Q 信号は、それぞれ \cos 波 (余弦波) と \sin 波 (正弦波) と乗算し、加算される。

直交変調器を利用したデジタル変調では、振幅や位相の変化は \sin 波と \cos 波の足し合わせで表現可能であり、変調波の振幅 A と位相 θ は I と Q を用いて式 (2) で表すことができる。

$$\begin{aligned} A \sin(2\pi ft + \theta) &= I \cos(2\pi ft) + Q \sin(2\pi ft) \quad (2) \\ A &= \sqrt{I^2 + Q^2}, \theta = \tan^{-1} \frac{Q}{I} \end{aligned}$$

次に本研究で対象とするデジタル変調方式である ASK, $\pi/4$ shift QPSK の IQ マッピングについて説明する。

2.1.2 ASK

振幅変調である ASK (Amplitude shift keying) では、式 (2) 中の位相 θ を 0 に固定し、振幅 A を変化させて送信データを符号化する。したがって、ASK の IQ マッピングでは $Q=0$ とし、スプリットフェーズ符号化することで、送信データに応じて I を変化させる。

スプリットフェーズ符号化では、0 が入力されたとき、0, 1 を時間差で出力し、1 が入力されたとき、1, 0 を時間差で出力する。スプリットフェーズ符号化は、送信データに 0 が連続する場合でも、被変調波の振幅が常に 0 にならないという特徴を持つ。図 3 に 4 サイクルで 1 つの送信データを符号化する ASK の出力を示す。送信データが 1 のとき、0 サイクル目に 1 を出力し、2 サイクル目に 0 を出力する。送信データが 0 のとき、0 サイクル目に 0 を出力し、2 サイクル目に 1 を出力する。

2.1.3 $\pi/4$ shiftQPSK

位相変調である $\pi/4$ shiftQPSK は、連続する 2 ビットの送信データに応じて位相の変化量を決定し、搬送波 $W_{\text{Carry}}(t)$ と搬送波 $W_{\text{Carry}}(t-1)$ との位相差を情報とする。 $\pi/4$ shiftQPSK の IQ マッピングは、横軸を I 信号の値、縦軸を Q 信号の値とした IQ 平面を使用する。 $\pi/4$ shiftQPSK では振幅を固定し、送信データに応じて位相は $\pi/4$ の整数倍で変化させる。よって単位円上の点 (P_n とする) を 8 個用意し、送信データが 2 ビット入力されるたびに、 P_n の値を変化させ、そのときの I の値、 Q の値をそれぞれ I 信号、 Q 信号として出力する。

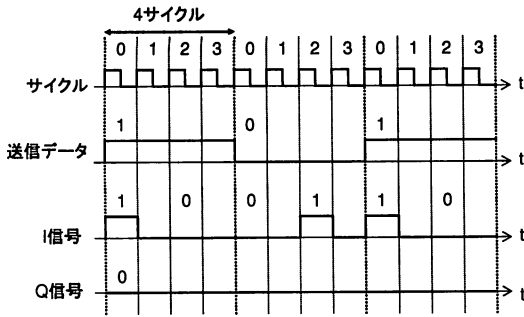


図3 ASKにおけるスプリットフェーズ符号化

IQ平面と P_x の関係を図4, 送信データ, 位相変化量と P_x の変化を表1, P_x の値と I, Q の出力を表2にそれぞれ示す.

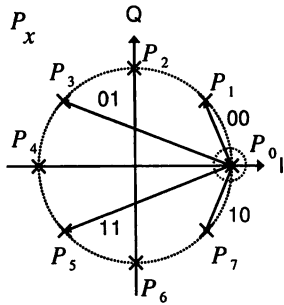


図4 $\pi/4$ shift QPSKにおけるIQ平面と P_x

表1 送信データと位相変化量と P_x の変化

送信データ	位相変化量	P_x の変化
00	$+\pi/4$	$P_x = (P_x + 1) \bmod 8$
01	$+3\pi/4$	$P_x = (P_x + 3) \bmod 8$
11	$+5\pi/4$	$P_x = (P_x + 5) \bmod 8$
10	$+7\pi/4$	$P_x = (P_x + 7) \bmod 8$

表2 $\pi/4$ shift QPSKにおける P_x の値とI信号, Q信号

P_x	I信号	Q信号
P_0	1	0
P_1	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
P_2	0	1
P_3	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
P_4	-1	0
P_5	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$
P_6	0	-1
P_7	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$

3. RASK

本研究では, デジタル変調に向けた演算機能を持つRASK (Reconfigurable Architecture for Shift Keying) を提案する. まず全体の構成について述べ, 次にRASKの動作遷移と各動作について述べる.

3.1 全体構成

RASKの構成を図5に示す. RASKは, 演算処理とスイッチボックスの制御を行うPCE(Processing and Control Element), スwitchボックスの配線制御を行うCE(Control Element), 任意の入力ポートから任意の出力ポートに配線可能なSwitchボックスSB(Switch Box), PCEと配線を接続するCB(Connection Box)から構成される.

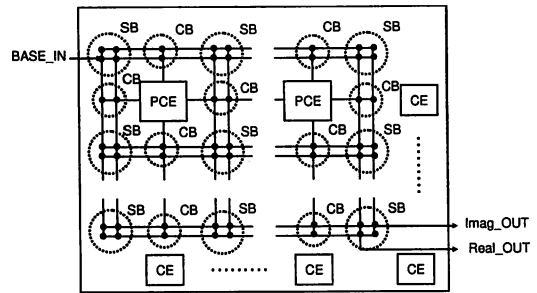


図5 RASKの全体構成

PCEとCEは内部にコンフィギュレーションデータを持ち, 設定されたコンフィギュレーションデータに応じてPCEの演算, 左上にあるSB, 左上にあるCBをそれぞれ制御する.

3.1.1 RASKの入出力インターフェース

RASKの入出力ポートを表3に示す. $BASE_IN$ を入力することで, I信号 ($Real_OUT$), Q信号 ($Imag_OUT$) を出力する回路である.

3.1.2 PCE(Processing and Control Element)

PCEの構成を図6に示す. PCEは演算を行うALU, 入力信号を選択するMA, MB, MC, MD, 値を保存するレジスタであるALU_REGとReg_R1, コンフィギュレーションデータを保存するレジスタファイルConfを持つ. PCEへの入力信号は, クロック信号clk, 演算データIn.a, In.b, コンフィギュレーションレジスタ内アドレス信号CAB, コンフィギュレーションデータバスCDB, コンフィギュレーションレジスタのアドレス指定信号Reconf, データ書き込み制御信号Dataset, コンフィギュレーションデータ書き込み許可信号EX, EYである. 出力信号は演算結果PCE_OUT, SB制御信号SB_OUT, CB制御信号CB_OUT.

Confはコンフィギュレーションデータを保存するレジスタファイルである. 1つのコンフィギュレーションデータは64ビットであり, Confはコンフィギュレーションデータを最大4つまで保存でき, 実行時に指定されたコンフィギュレーションデータによって, ALUでの演算, PCE内のデータバス, 制御するSB, CBの接続が変化する.

表 3 RASK の入出力ポート

ポート名	方向	ビット幅	意味
clk	IN	1	クロック信号
Reconf	IN	2	PCE, CE 内コンフィギュレーションレジスタのアドレス指定信号
BASE_IN	IN	8	入力ベースバンド信号
CAB	IN	4	PCE 内コンフィギュレーションレジスタ内の bit 指定信号
CDB	IN	8	PCE 内コンフィギュレーションレジスタへのデータ信号
EX	IN	9	コンフィギュレーションレジスタへの書き込み制御信号
EY	IN	9	コンフィギュレーションレジスタへの書き込み制御信号
Dataset	IN	1	PCE 内動作制御信号
Real_OUT	OUT	8	I 信号
Imag_OUT	OUT	8	Q 信号

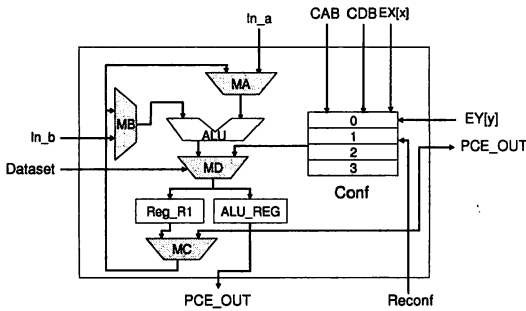


図 6 PCE の回路構成

3.1.3 CE

CE は PCE の SB, CB を制御するが、データの演算を行わないため、PCE から演算機能を省いた構成となる。

3.1.4 SB, CB

SB と CB は、配線とスイッチから構成されている。図 5 のパスの交点は図 7 のような構造をしており、任意の配線から任意の配線へデータ転送が可能である。

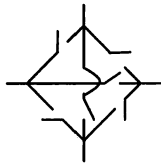


図 7 スイッチの回路構成

3.2 デジタル変調に向けた演算機能

3.2.1 デジタル変調における IQ マッピングの特徴

デジタル変調における IQ マッピングにおいて、位相 θ から出力される I, Q を計算する場合、三角関数を計算する演算器が必要となり、面積の増大や処理時間の増大を招く。そこで本研究では、デジタル変調における IQ マッピングの特徴を最大限に生かすことで、面積の増大や処理時間の増大を抑えたアーキテクチャを提案する。

表 2 が示すように、デジタル変調における IQ マッピングの特徴は、

- (1) 位相をあらわす変数 P_x に応じて出力される I と Q の値は、デジタル変調方式に応じて決定されている、
 - (2) 出力される I または Q の値が同じ P_x と P_x' が存在する、
 - (3) 出力される I または Q の値の絶対値が同じ P_x と P_x' が存在する、
- の 3 点である。

3.2.2 出力値の記憶と IsEqual 演算

提案するアーキテクチャでは、三角関数の処理時間を削減するため、(1) の特徴に着目し、出力される I と Q の値の候補をレジスタに保存し、 P_x との条件判定結果と乗算することで、 P_x の値に応じた I と Q を出力する。その例として、 $\pi/4$ shift QPSK における P_x が 1 のときの I を出力する DFG を図 8 に示す。

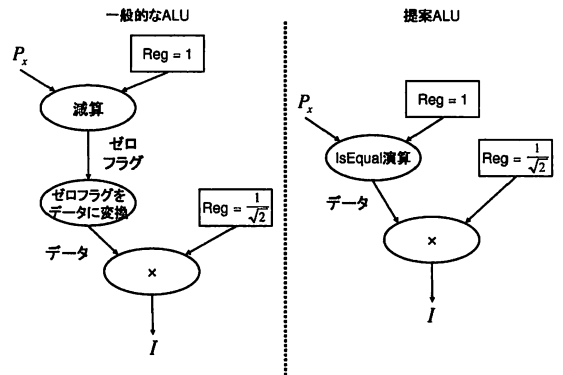


図 8 P_x が 1 のときの I を出力する DFG

一般的な ALU でこの処理を行う場合、図 8 の左側の処理を行う。まず P_x が 1 かどうか判定するために、 P_x と 1 の比較を行う。その比較結果はゼロフラグに出力されるが、ゼロフラグを演算に用いることができないため、ゼロフラグの値をデータに変換する必要がある。そして、データに変換された比較結果とレジスタに保存した値との乗算を行い、 P_x が 1 のときの I を出力する。したがって一般的な ALU を用いる場合、比較結果をデータに変換する仕組みが必要になる。そこで本研究では、比較結果をフラグではなくデータとして出力する“IsEqual”演算を提案する。

“IsEqual”演算は条件判定を行う演算で、ALUの入力 ALU_A と ALU_B の値が等しければ 1 を出力し、異なれば 0 を出力する。図 8 の右側に “IsEqual” 演算を用いるときの DFG を示す。まず P_2 が 1 が判定するために “IsEqual” 演算を行う。そしてその結果とレジスタに保存した値との乗算を行い、 P_2 が 1 のときの I を出力する。“IsEqual” 演算を使用することで、比較結果との乗算をすぐに実行できるため、PCE の個数を削減できる。また、ゼロフラグが必要ないため、ゼロフラグに必要なビットを削減することができる。

また、(2) の特徴より、複数の位相でレジスタに保存した値を共有することで、使用する PCE 数を削減することができる。 P_3 と P_6 で I を共有する例を図 9 に示す。

入力された P_2 に対して並列に “IsEqual” 演算を行い、それぞれの比較結果との論理和をとり、論理和の結果と $-\sqrt{\frac{1}{2}}$ を乗算することで、 P_3 と P_6 の I を出力できる。

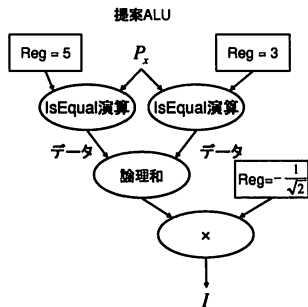


図 9 P_2 が 3 または 5 のときの I を出力する DFG

3.2.3 演算の共有と IfNOT 演算

さらに本研究では (3) の特徴に着目し、出力の絶対値のみをレジスタに保存し、 P_2 の値に応じて出力値の正負の符号を決定する機能を ALU に追加する。提案アーキテクチャでは、“IsEqual” 演算によって P_2 の比較結果がデータに出力されるため、ALU の入力データによって、出力値を反転させる “IfNOT” 演算を提案する。“IfNOT” 演算は ALU_B の値によって、 ALU_A または $-ALU_A$ を出力する演算である。

図 10 は、 P_2 の値が 1 または 7 なら I として $-\sqrt{\frac{1}{2}}$ 、 P_2 の値が 1 または 7 でなければ I として $\sqrt{\frac{1}{2}}$ を出力する DFG を表している。一般的な ALU では、 P_2 が 1 か 7 なら $-\sqrt{\frac{1}{2}}$ と乗算し、 P_2 が 1 か 7 以外なら $\sqrt{\frac{1}{2}}$ と乗算を行う。一方提案アーキテクチャでは、“IsEqual” 演算と “IfNOT” 演算を用いることで、使用する PCE 数を大幅に削減することができる。

3.3 動作遷移

RASK では、まずコンフィギュレーションデータを PCE, CE のコンフィギュレーションレジスタの *Conf* に書き込み、書き込まれたコンフィギュレーションデータを元にアプリケーションの処理を行う。RASK の動作遷移図を図 11 に示す。

3.3.1 コンフィギュレーションデータ書き込み状態

コンフィギュレーションデータ書き込み状態では、アプリケーションのコンフィギュレーションデータを PCE, CE に書き込

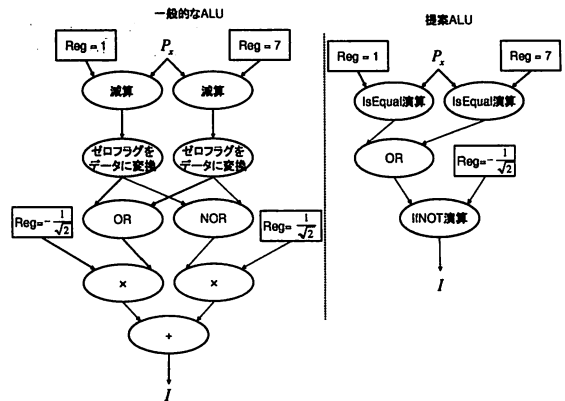


図 10 P_2 の値に応じて I を出力する DFG

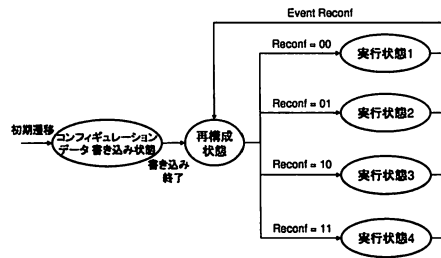


図 11 RASK の動作遷移

む処理を行う。コンフィギュレーションデータの書き込みは、書き込み制御信号の EX, EY が共に 1 の PCE または CE の、レジスタファイル内の *Reconf* で指定されたレジスタに対して行う。なお PCE, CE のコンフィギュレーションデータは 64bit であり、コンフィギュレーションデータバスの *CDB* は 8bit であるため、*Conf* にコンフィギュレーションデータを 1 回で書き込むことはできない。そこでコンフィギュレーションデータを 8 つに分割し、それぞれ *Conf* アドレス 0 番地から 7 番地とし、8 ビット単位で書き込む。その *Conf* アドレスの指定は *CAB* が行う。

RASK では PCE が 64 個、CE が 17 個、1 つの PCE, CE のコンフィギュレーションデータが 64 ビット、コンフィギュレーションデータバス *CDB* が 8 ビットなので、1 つのコンフィギュレーションの書き込みサイクル数 $Cycle_{write}$ は $(64 + 17) \times (64/8)$ で 648 サイクルである。

3.3.2 再構成状態

再構成状態は、*Reconf* で指定されたコンフィギュレーションデータにしたがって回路の構成を変更する状態である。回路の構成が変更された後、コンフィギュレーションデータで指定された初期値をレジスタ *ALU_REG*, *Reg_R1* に代入する。

3.3.3 実行状態

実行状態ではコンフィギュレーションデータに基づいて PCE で演算を行い、SB, CB を使用しデータ転送を行う。

表 4 ASK, $\pi/4$ shiftQPSK における RASK の動作周波数, PCE 数, 面積における比較

	ASK	$\pi/4$ shift QPSK	ASK で必要な PCE 数	$\pi/4$ shift QPSK で必要な PCE 数	面積
“IsEqual” 演算, “IfNOT” 演算を持たないとき	約 230MHz	NA	16 個	76 個	約 310[kgate]
RASK	約 230MHz	約 210MHz	13 個	55 個	約 280[kgate]

3.4 RASK でのデジタル変調への実現

RASK における設計フローを以下に示す。

- (1) DFG の作成
- (2) RASK への配置配線
- (3) コンフィギュレーションデータの作成

まず RASK で実行するアプリケーションを, PCE 内の ALU の演算機能をノードとした DFG(Data Flow Graph) で表現する。次に, DFG のノードを PCE に割り当て, DFG にしたがって PCE 間の配線経路を決定する。そして, 決定された配置配線情報から PCE, CE に格納するコンフィギュレーションデータを作成する。

4. 評価実験

本章では, RASK に ASK と $\pi/4$ shiftQPSK を実装し, 評価した結果を述べる。そして, RASK の柔軟性を確認するために, QPSK, 16QAM の実装した結果について述べる。

評価環境では以下の環境を使用した。

- シミュレータ: Mentor Graphics 社 Model Sim 6.2a
- 論理合成ツール: Synopsys 社 Design Compiler
- ライブラリ: 0.14 μ m CMOS スタンダードセルライブラリ

4.1 ASK, $\pi/4$ shiftQPSK の実装時の評価

RASK と, RASK と同じ回路構成で ALU が “IsEqual” 演算と “IfNOT” 演算を持たない一般的な ALU を使用したときで, ASK, $\pi/4$ shiftQPSK を実行したときの動作周波数, PCE 数, 面積を表 4 に示す。動作周波数は, PCE 間でスイッチボックスを最も多く経由した経路の遅延時間から算出した。

表 4 より RASK は, “IsEqual” 演算, “IfNOT” 演算を持たないときより, 少ない PCE 数でデジタル変調を実現できることが確認できた。特に $\pi/4$ shiftQPSK では, “IsEqual” 演算, “IsNOT” 演算を持たなければ, PCE 数が足りず実装できなかった。また表 4 より, “IsEqual” 演算と “IfNOT” 演算は動作周波数に大きな影響を及ぼさないことが確認できた。また表 4 より, RASK の面積が “IsEqual” 演算, “IfNOT” 演算を持つことで小さくなった。これは “IsEqual” 演算を追加したことで, フラグが不要になり, データバスのビット数を削減できたためである。

4.2 QPSK, 16QAM の実装

RASK の柔軟性を確認するために, QPSK, 16QAM を実装し, 動作周波数を測定した結果を表 5 に示す。QPSK, 16QAM 共に入力データによって I 信号, Q 信号を出力する処理である。

表 5 RASK における QPSK, 16QAM の動作周波数

QPSK	16QAM
約 230MHz	約 210MHz

4.3 伝送速度

測定した動作周波数に基づいて算出したデジタル変調の伝送速度と, 要求される伝送速度を表 6 に示す。

表 6 RASK の各変調方式における伝送速度

変調方式	ASK	$\pi/4$ shiftQPSK
アプリケーション	ETC	PDC
要求伝送速度	1Mbps	384kbps
達成伝送速度	約 57Mbps	約 105Mbps
変調方式	QPSK	16QAM
アプリケーション	ケーブルモデム	モデム
要求伝送速度	128K-3Mbps	2400-4800bps
達成伝送速度	115Mbps	210Mbps

本研究で提案する RASK では, いずれの変調方式においても要求される伝送速度を達成している。

5. おわりに

本研究ではデジタル変調向けの演算機能を持つリコンフィギュラブル・アーキテクチャ RASK を提案した。評価実験より, PCE がデジタル変調向けの演算機能を持つことで, 同一デジタル変調を少ない PCE 数で実現でき, 面積を削減できた。また, RASK で ASK, $\pi/4$ shiftQPSK を実現できたことを確認し, それぞれのデジタル変調方式で, 要求される伝送速度を達成できた。また, QPSK, 16QAM をそれぞれのコンフィギュレーションデータを作成することで実現できた。その際, QPSK, 16QAM のデジタル変調方式で, 要求される伝送速度を達成できた。

今後の課題はリコンフィギュラブル・アーキテクチャのための低消費電力手法とデジタル復調の実装が挙げられる。

文 献

- [1] R. Hartenstein, “A Decade of Reconfigurable Computing: a Visionary Restrospective,” Proc. of DATE2001, pp. 642-649, 2001.
- [2] 天野英晴, “最近のリコンフィギュラブルシステム, 動的リコンフィギュラブルシステムの動向,” 電子情報通信学会技術研究報告 SR2005-5, Vol.105, No.36, pp.31-36, 2005 年 5 月.
- [3] 末吉敏則, 天野英晴, “リコンフィギュラブルシステム,” オーム社, 2005.
- [4] K. Masselos, and N. S. Voros, “Introduction to Reconfigurable Hardware”, System Level Design of Reconfigurable Systems-on-Chips, pp. 15-26, Springer, 2005.
- [5] 藤沢久典, 斉藤美寿, 境裕樹, 西島誠一, 大館尚記, 杉山麻, 廣瀬住生, 吉澤英樹, “動的再構成回路に基づくソフトウェア無線向け信号処理プラットフォーム LSI の開発” 電子情報通信学会技術研究報告 CE2006-107, Vol.30, No.65, pp.49-54, 2006 年 12 月.
- [6] 電子情報通信学会, “電子情報通信ハンドブック,” オーム社, pp. 584, 1998.