

実時間通信システムに対する高信頼ハードウェア合成手法の提案

高本 佳史[†] 木谷 友哉[†] 安本 慶一^{††} 中田 明夫[†] 東野 輝夫[†]

[†] 大阪大学大学院 情報科学研究科 〒560-8531 大阪府豊中市待兼山町1-3

^{††} 奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町8916-5

E-mail: †{takamoto,t-kitani,nakata,higashino}@ist.osaka-u.ac.jp, ††yasumoto@is.aist-nara.ac.jp

あらまし 本稿では、実時間通信システムを複数の機能部品で構成する場合に、コストおよび性能のバランスを考慮しつつ回路を効率よく設計する方法を提案する。提案手法では、実時間システムを、システム全体の性能と各機能部品の性能をパラメタとして用いて並行周期 EFSM 群としてモデル化する。その際、幾つかの部品の共通化、ある部品の並列化などの部品構成の変更が効率よく行えるモデル化の方法を提案する。記述したシステムの仕様から、パラメトリックモデル検査法により、並行周期 EFSM 群が指定された時間制約の下でデッドロックすることなく周期的に処理を繰り返し実行できるためのパラメタ間の関係を求め、また、その関係に従い、各種パラメタ値を決めることで、RT レベルの VHDL 記述を得ることができるためのツールを開発した。携帯電話端末の基本モジュールの設計・実装を通して、本手法の有用性を確認した。

キーワード 実時間システム、高位合成、パラメトリックモデル検査、スケジューリング

A Proposal of a High-Reliable Design of Circuits Modeled as Concurrent Periodic EFSMs

Yoshifumi TAKAMOTO[†], Tomoya KITANI[†], Keiichi YASUMOTO^{††}, Akio NAKATA[†], and
Teruo HIGASHINO[†]

[†] Graduate School of Information Science and Technology, Osaka University
1-3 Machikanyama-cho, Toyonaka, Osaka 560-8531, Japan

^{††} Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan

E-mail: †{takamoto,t-kitani,nakata,higashino}@ist.osaka-u.ac.jp, ††yasumoto@is.aist-nara.ac.jp

Abstract In this paper, we propose a high-reliable design method for real-time systems consisting of multiple functional modules where the balance between cost and performance can effectively be considered. In our method, we specify a system as a parallel composition of concurrent periodic EFSMs where the behavior of a functional module is specified as a periodic EFSM with timing constraints including performance parameters of the module. Here, we propose modeling techniques to easily treat the cases that the organization of functional modules is frequently changed for cost reduction by unifying multiple modules to one or for performance improvement by paralelizing a module to multiple ones. We have implemented tools to derive parameter conditions which must hold for the system to proceed without deadlocks, and to generate RT-level VHDL descriptions. By applying our method to development of the basic functionality of a cellular phone, we have confirmed the usefulness of our approach.

Key words real-time systems, high-level synthesis, scheduling, parametric model checking

1. まえがき

近年の高速ネットワークやマルチメディア通信システムの発展・普及に伴い、実時間通信システムをハードウェアで実現し高速化する方法が注目されている。ほとんどの実時間通信システムは、大部分が繰り返し周期的な処理を実行する時間制約付きのモジュールとして実装される。その上で適応性のある実時間通信システムを開発するためには、処理を指定された周期で繰り返し実行でき、かつ、システム全体としてデッドロックを起ささないよう、各動作を適切な時刻に実行するための時間制約のパラメタを見つけ出す必要がある。また同時に、システムの性能を高めるためには、できるだけ周期を短くできることが

望ましい。しかしながら、適切なパラメタ値を得るためには、通常、試行錯誤を繰り返してシミュレーションや分析を行い、パラメタ値を修正しなくてはならない。

以上の目的に対し、我々は文献[4]において、実時間システムの高信頼設計手法について提案している。本手法では、実時間システムを、システム全体の性能と各機能部品の性能をパラメタとして用いて並行周期 EFSM 群としてモデル化し、パラメトリックモデル検査法[5]を用いて、並行周期 EFSM 群が指定された時間制約の下でデッドロックすることなく周期的に処理を繰り返し実行できるためのパラメタ条件を求める。得られたパラメタ条件をもとに、条件を満たすパラメタ値、すなわち、適切な機能部品を決定していくことで、コストおよび性能のバ

ランスを考慮したシステムの設計が可能になっている。また、決定したパラメータ値をもとに、整数線形計画問題の手法を用い、全ての並行周期 EFSM 群が指定した周期で処理を実行するための各動作の実行時間の範囲を導出し、得られた時間範囲から、EFSM 群を制御する実時間スケジューラを生成し、最終的にレジスタ転送レベルの VHDL 記述を導出するためのツールを作成している。

一般に、実時間システムの設計においては、目標とする性能およびコストを達成するため、部品構成の変更が頻繁に行われる。例えば、コスト削減のため、2つ以上の部品 (DSP と CPU など) を一つの部品に共通化したい場合や、ボトルネックを解消するため、一つの部品で行っていた処理を複数の部品で並列処理したい場合などが存在する。

そこで、本稿では、文献 [4] の方法を拡張し、並行周期 EFSM モデルを用いて、部品の共通化および部品の並列化に伴う仕様の変更を最小限で行えるようにするモデル化手法を提案する。提案手法では、あるシステムで使われる機能部品群の動作に対応する周期 EFSM 群の並列実行として記述する。周期 EFSM 間では、マルチランデブ [1] (複数並行プロセスにおいて、それぞれ条件が満たされた時に指定した動作 (イベント) を同期実行しデータ交換する仕組み) による同期通信が可能である。部品 A と部品 B の共通化を行いたい場合には、A の各パス (各周期で実行可能な動作系列) と B の各パスが同時には実行できない (すなわち、どちらかのパスの実行後にのみ他方を実行できる)、という制約を表す周期 EFSM を追加し、これら部品に対応する EFSM とマルチランデブにより同期実行するよう仕様を記述する。一方、部品 C の並列化を行いたい場合には、C に対応する周期 EFSM のインスタンスを並列化したい数だけ増やし、それらインスタンスを並列実行するよう指定する。マルチランデブ機構を用いることで、部品 C と通信する他の部品は、C のインスタンスのうち未使用のものと同通信するようにできる。提案手法では、部品の共通化および並列化を、幾つかの周期 EFSM を仕様を追加し同期並列実行することで容易に実現でき、また、修正後の仕様もまた並行周期 EFSM モデルのクラスに入っている。このため、文献 [4] で作成した、デッドロックしないためのパラメータ条件の導出ツールと、レジスタ転送レベルの VHDL 記述を得るためのツールがそのまま使用できる。

提案手法の有用性を示すため、携帯電話の基本機能を周期 EFSM 群によりモデル化し、部品の共通化および並列化が容易に行えることを確認した。また、記述した仕様から VHDL 記述を導出し、市販の CAD ツールで合成した結果、実用に耐える性能の回路が合成可能なことを確認した。

2. ハードウェア高信頼性設計手法

本設計手法では、システムを、システムのトータル性能 (周期) と各機能部品の性能 (メモリのアクセススピード等) をパラメータとして仕様化することで、システムがデッドロックすることなく動作するためにパラメータ間で成立すべき条件式が導出される。この条件式を用いて、例えば、最終的に必要となる性能を固定し、その場合に、条件式を満たすような機能部品を選択することが容易に行える。文献 [4] では、本設計手法を用いて、QoS ルータにおける WFQ 機構を、性能優先、コスト優先で設計する事例について報告している。

我々が文献 [4] で提案したハードウェア設計手法における設計手順は以下のとおりである (図 1)。

- (1) 対象とする実時間システムの仕様を並行周期 EFSM でモデル化する。このとき、モデル中の制約条件には、使用する機能部品の性能や実現したい速度 (周期) などをパラメータとして指定することが可能である。
- (2) パラメトリックモデル検査法により、モデル化し

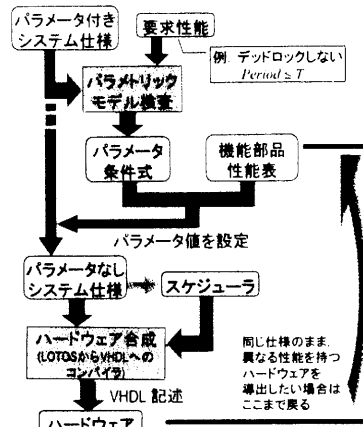


図 1 設計手順

た仕様でデッドロックしないパラメータ条件式を導出する。

- (3) 実現したい速度 (周期) や機能部品の性能表 (性能、価格、消費電力など) から得られたパラメータ条件式を満たすように、各機能部品を選定し、パラメータの値を固定する。
- (4) パラメータが固定されたシステムの仕様から、各 EFSM のイベントの実行順序を制御するスケジューラ EFSM を自動的に導出する。
- (5) 上記の手順で得られた仕様から、我々が文献 [3] で開発したツールを用いることで、レジスタ転送レベルの VHDL 記述を得る。
- (6) 同じ仕様から異なる性能を持つ回路を導出したい場合は、上記 (3)~(5) を繰り返す。

以下、これらについて簡単に述べる。(詳細については [4] 参照。)

2.1 並行周期 EFSM

EFSM は、変数に相当するレジスタ群と遷移に対する条件式 (ガード) が記述できるよう拡張された状態機械 (FSM) である。システムと外部の環境とのデータの入出力動作は、イベントと呼ばれる。イベントの実行は即座に完了すると仮定する。また、異なる EFSM で同期実行するよう指定されている同名のイベントは、同期イベントと呼ばれる。入力、出力イベントはそれぞれ $a?x$, $b!E$ と表記される。ここで a , b は環境とのインタラクションポイントであり、ゲートと呼ばれる。入力変数 x は、再び $a?x$ が実行されるまで不変である。さらに、 $a@?t$ はイベント a の実行時刻として、直前の状態遷移からの経過時間 $clock$ の値を時間変数 t に代入することを表す。この時間変数 t は、再び $a@?t$ が実行されるまで不変である。clock の値は整数として扱われる。EFSM における状態 s_{cur} から s_{next} への遷移規則は、 $s_{cur} \xrightarrow{a[guard]} s_{next}$ と定義される。ここで、guard は遷移条件と呼ばれる。もし遷移条件 guard が状態 s_{cur} において真であり、イベント a が実行されたなら、その EFSM は状態 s_{next} に遷移する。このとき、遷移条件内で時間変数を使うことによって、それぞれのイベントの実行時間に制約を与えることができる。遷移条件の式にはパラメータを含む線形不等式を指定することが可能である。

EFSM が必ず一定時間間隔 T で初期状態に戻ることができるように遷移条件が指定されているならば、その EFSM を周期 EFSM と呼ぶ。与えられた EFSM を周期的にするために、我々は初期状態からの全てのパスが、特別な遷移 ψ を持つと仮定する (図 2 の点線の遷移)。

本研究では EFSM 間の通信 (データ交換、同期など) は仕様記述言語 LOTOS [1] のマルチランデブ機構と呼ばれる高レベルな通信プリミティブを用いて実現する。マルチランデブとは、

複数の並行プロセスがそれぞれある条件を満たしたとき、これらのプロセス間で同期してデータ交換を行う仕組みである。EFSM 間のマルチランデブ関係は、以下に示す LOTOS の並列オペレータで指定される ($||gate_list||$ はオペレータの両側の EFSM が $gate_list$ のイベントを同期実行することを指定し、 $|||$ は両側の EFSM がイベントの同期実行なしに独立に動作可能なことを指定する)。

$$S ::= S ||gate_list|| S \mid S ||| S \mid (S) \mid EFSM$$

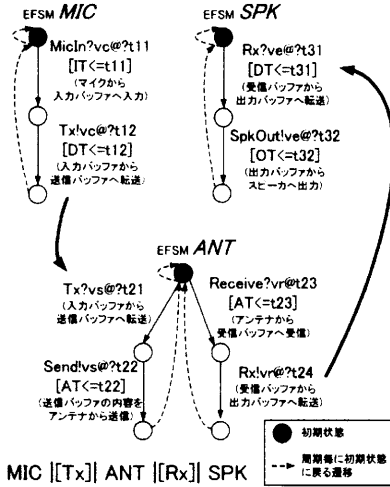


図2 携帯電話端末の一部をモデル化した並行周期 EFSM 群

ここで図2の仕様は、携帯電話端末の一部を表現したものであり、MICが音声のマイク入力、SPKが音声のスピーカ出力、ANTがアンテナを介した音声データの送受信を表現している。MIC、ANTはイベントTxで、SPK、ANTはイベントRxでマルチランデブを行うよう指定されている。

2.2 パラメトリックモデル検査

文献[5]において、我々は周期EFSMのためのパラメトリックモデル検査法を提案した。この手法では、時間的性質はRPCTL (Real-time and Parametric extension of Computation Tree Logic) で記述される。モデルを周期的であることに制限することにより、この手法では、高々3周期分の与えられた周期EFSMの振舞いを解析することで効率的にパラメータ条件を導出することができる(詳細は文献[5]参照)。

図2の並行周期EFSM群の例において、“システムはいつも周期Period毎に特別リセットランジション ψ を実行して初期状態に戻らなければならない”という性質を考える。この性質はRPCTLでは $AG_{=T}(EF(\psi \text{ true}))$ と記述される。文献[5]で提案したモデル検査手法は、このような制約から、“ $Period \geq IT + OT \text{ and } Period \geq IT + AT + DT \text{ and } Period \geq AT + DT + OT \text{ and } Period \geq 2AT + 2DT$ ”というパラメータ条件を導出する。これは、デッドロックなく周期的に実行するためには、周期Periodは上記の式を満たさなければならないことを示している(ここで、ITはマイクから入力バッファへの入力時間、DTはバスの転送時間、ATはアンテナ・バッファ間の送受信時間、OTはスピーカへの出力時間である)。

2.3 スケジューリングアルゴリズム

イベントの実行系列内のすべてのイベントが時間制約を満たしながら実行できる(スケジュール可能である)ようにするには、連続するイベント群が各々の時間制約を満たすように実行時間を決める必要がある。そこで、提案手法では、イベント系列をスケジュール可能とするような各イベントの実行可能時間範囲をあらかじめ導出する。

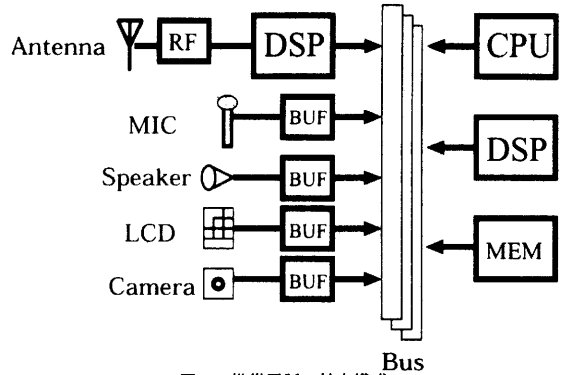


図3 携帯電話の基本構成

先行するイベントの実行時間に依存して、後続のイベントの実行可能時間範囲が変化する場合があります。提案手法では、各イベントがある範囲のどの時点で実行されても、後続のイベントの実行可能時間範囲が変化しないような範囲を求めることにする。

以上のような静的スケジューリングを行うことにより、各イベントが実際に実行された時刻を記憶する必要がなくなり、現在時刻を表すグローバルなタイマと複数個の比較器のみで各イベントの時間制約が満たされるか否かを判定することが可能となる[3]。

2.4 回路の導出

提案手法の有用性を確認するために、仕様記述言語ELOTOS[2]の構文を用いて記述された並行周期EFSMの動作仕様から、ハードウェア記述言語VHDLによるレジスタ転送レベルの回路記述を自動合成するツールを実際に作成した。導出される回路記述は大きく分けて以下の3つの部分、(1)同じクロックで動作する各EFSMに対応する順序回路、(2)マルチランデブコントローラ、(3)各時刻でEFSMのスケジュール可能なイベント系列のみを示すスケジューラ、から構成される[3]。

3. 回路構成の変更に柔軟なモデル化手法の提案

回路設計においては、幾つかの機能部品を共通化することでコストダウンをはかったり、あるいはボトルネックになっている部分を並列化することで、高速化をはかりたい場合が多い。本稿では、制約指向記述スタイル[6]を用いることにより、仕様をほとんど変更することなく、部品間の制約を付け加える、あるいは変更するだけで、部品の共通化および並列化を実現するモデル化手法を提案する。以下では、カメラ付き携帯電話の基本動作部分をモデル化する事例を使って、提案手法を説明する。

3.1 携帯電話の基本構成

携帯電話の基本構成として、図3のように、マイク、スピーカ、液晶画面、カメラがバスを介してDSP、メモリ、CPUと接続する形となるモデルを考える。また、図8では、図3の基本構成を以下の9つの部分に分けてモジュール化したものを図8に示す。

$FUNC \ ||\{S0, Call, Drcv, Dsnd, Pct, Mv\}$

$BUS \ ||\{S0, Bg, Br\}$

$(MIC|||SPK|||ANT_IN|||ANT_OUT|||LCD|||CCD)$

$||\{S0, Dg, DSP, Dr\} \ || \ DSP$

合計で、9個の周期EFSMが並列に動作する。

図8のように各モジュールをそれぞれ独立に記述することにより、使用するバスやDSPの数を容易に修正して置き換える

ことが出来るようになっている。

図 8 の仕様では、以下に示す制約パラメータを使用している。

T	全 EFSM の周期
T_{do}	音声に対する DSP 処理時間
T_{dv}	画像に対する DSP 処理時間
T_{ba}	音声のバス転送時間
T_{bd}	データのバス転送時間
T_{bv}	画像のバス転送時間

3.2 携帯電話モジュールの動き

図 8 において、全ての EFSM は時間周期 T で周期的に動作する。各 EFSM の動きを順に説明していく。

- (1) *FUNC* は、携帯電話がどのような動作をするのかを決定するモジュールである。通話 (*Call*)、データ受信 (*Drcv*)、データ送信 (*Dsnd*)、静止画撮影 (*Pct*)、動画撮影 (*Mv*) のいずれかの動作の選択を行う。
- (2) *BUS* は、バスの調停を行うモジュールである。このモジュールは、バス使用権を得るイベント B_g と使用しているバスを解放するイベント B_r を繰り返し行う。これにより、ある EFSM がバスの使用権を得たら、その EFSM がバスの解放イベントを起こすまで他の EFSM はバスを使用できないというバスの排他制御を実現している。
- (3) *DSP* は、DSP の調停を行うモジュールである。このモジュールは、DSP 使用権を得るイベント D_g と使用している DSP を解放するイベント D_r を繰り返し行う。バスの調停と同様に、これにより DSP の排他制御を実現している。
- (4) *MIC* は、マイク機能を持つモジュールである。通話イベント信号 (*Call*)、または、動画撮影信号 (*Mv*) を受け取ると、DSP の使用権を要求し (D_g)、得ることが出来たら、バスの使用権を要求する (B_g)。そして、バスを使用できるようになると、データを DSP に転送し (MtD)、バスを解放する (B_r)。DSP での処理 (*DSP*) が終わると再びバスの使用権を要求し、DSP からデータを受け取り (DtM)、バスを解放する (B_r)。最後に DSP を解放する (D_r)。
- (5) *SPK* は、スピーカ機能を持つモジュールである。通話イベント信号 (*Call*) を受け取ると、DSP の使用権を要求し (D_g)、得ることが出来たら、バスの使用権を要求する (B_g)。バスを使用できるようになると、メモリからデータを DSP に転送し (MtD)、バスを解放する (B_r)。DSP での処理 (*DSP*) が終わると再びバスの使用権を要求し (B_g)、DSP からデータを受け取り (DtS)、バスを解放 (B_r)、さらに DSP を解放する (D_r)。
- (6) *ANT_IN* は、アンテナ入力モジュールである。通話イベント信号 (*Call*)、または、データ受信信号 (*Drcv*) を受け取ると、ベースバンド部での処理を実行し (BB) バスの使用権を要求する (B_g)。バスを使用できるようになると、ベースバンドからデータをメモリに転送し (BtM)、バスを解放する (B_r)。
- (7) *ANT_OUT* は、アンテナ出力モジュールである。通話イベント信号 (*Call*)、または、データ送信信号 (*Dsnd*) を受け取ると、バスの使用権を要求する (B_g)。バスを使用できるようになると、メモリからデータをベースバンド部に転送し (MtB)、バスを解放する (B_r)。そしてベースバンド部での処理を行う (BB)。
- (8) *LCD* は、ディスプレイモジュールである。静止画撮影信号

(*Pct*)、または、動画撮影信号 (*Mv*) を受け取ると、DSP の使用権を要求し (D_g)、得ることが出来たら、バスの使用権を要求する (B_g)。バスを使用できるようになると、メモリからデータを DSP に転送し (MtD)、バスを解放する (B_r)。DSP での処理 (*DSP*) が終わると再びバスの使用権を要求し (B_g)、DSP からデータを受け取り (DtL)、バスを解放 (B_r) する。最後に DSP を解放する (D_r)。

- (9) *CCD* は、CCD カメラモジュールである。静止画撮影信号 (*Pct*) または、動画撮影信号 (*Mv*) を受け取ると、DSP の使用権を要求し (D_g)、得ることが出来たら、バスの使用権を要求する (B_g)。バスを使用できるようになると、CCD からデータを DSP に転送し (CtD)、バスを解放する (B_r)。DSP での処理 (*DSP*) が終わると再びバスの使用権を要求し (B_g)、DSP からデータをメモリに転送し (DtM)、バスを解放 (B_r) する。最後に DSP を解放する (D_r)。

以上が各 EFSM の動作である。

全体の同期通信例として、例えば、通話をする場合には、EFSM *FUNC* で、通話イベントである *Call* を実行し、通話モードにするように各 EFSM に信号を送る。すると、通話モードに関係する EFSM *MIC*, *SPK*, *ANT_IN*, *ANT_OUT* の各 EFSM が同期して動作する。次に音声等の処理を行うため、これらの EFSM はそれぞれ DSP やバスを利用する。その際、DSP やバスを 2 つ以上のモジュールが同時に利用することはできない。そこで、DSP とバスの調停を行う EFSM *BUS*、*DSP* と同期して動作することでこの問題を解決している。

3.3 部品の共通化

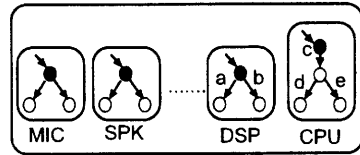


図 4 元の仕様

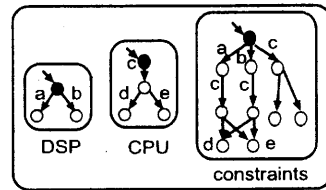
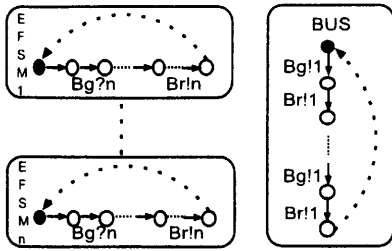
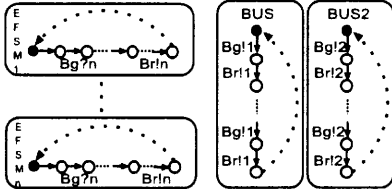


図 5 DSP と CPU を共通化するための制約

次に、部品の共通化について述べる。携帯電話に限らず、回路設計の際、消費電力や回路の規模を抑える事は大きな課題である。その対策として部品の共通化が挙げられる。もし共通化しても性能が落ちない場合は、それらの部品を共通化することで、より良い回路が設計できる。図 4 のモデルにおいては、CPU と DSP は独立に動作する。このモデルを、CPU と DSP を共通化して、一つのモジュールとして動作させる場合を考える。図 5 では、DSP、CPU で実行されるバスが、シーケンシャルに実行されるよう constraints に記述している (例えば、a の後に c-d など)。constraints と DSP、CPU の間で同名のイベントを同期実行させることで、constraints に指定している通りに順番でイベントが実行される。この例では、DSP と CPU の動作系列 (バス) をシーケンシャルに実行する全ての組み合わせについて記述しているが、デッドロックしてしまう場合もあるので、例えば、DSP → CPU の順番でしか実行しない、また、a を実行した時は c-d の系列のみ実行し、b を実行した時は c-e の系列を実行する、というように、スケジューリングのパター



each EFSMs [[Bg,Br]] BUS_ARB
図6 変更前



each EFSMs [[Bg,Br]] BUS_ARB
図7 変更後

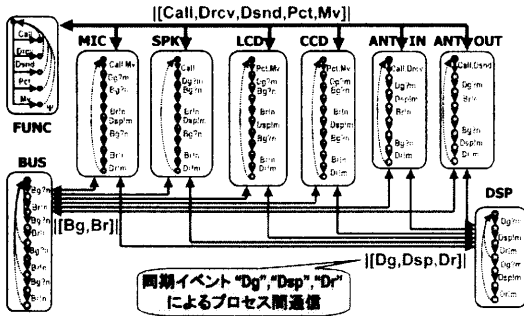


図8 携帯電話のモジュール

ンを減らすことで、システム全体の性能を向上させることも可能である。(紙面の都合で詳細は省略)

先の例では、デッドロックを起こさないためには、以下の条件を満たす必要がある。

$$\begin{aligned}
 &Ta + Tc + Td \leq T \text{ and } Ta + Tc + Te \leq T \\
 &\text{and } Tb + Tc + Td \leq T \text{ and } Tb + Tc + Td \leq T \\
 &\text{and } Tc + Td + Ta \leq T \text{ and } Tc + Te + Ta \leq T \\
 &\text{and } Tc + Td + Tb \leq T \text{ and } Tc + Te + Tb \leq T
 \end{aligned}$$

3.4 バスの本数の変更

音声や文字列のようなデータを処理する場合、バスネックになる可能性は低いが、次世代携帯電話で高画質の映像を送受信する場合など、バスを複数本用意することで性能が向上する場合がある。そこで、バスの本数を変更してモデル化を行い比較をした。このような仕様の変更は、本手法を用いる場合、制約指向でモデル記述をしているのでバスに関するEFSMの仕様を変更するだけでよく、容易である。具体的に、バス1本からバス2本に変更する際には、図6から図7のようにバスを表すモジュールを1つ追加して、もとのEFSMと同期実行させるだけで良い。このように、バスの本数を1本から3本まで変更して評価した結果は4.2章で述べる。

4. パラメータ値の決定と回路合成

図8に示す携帯電話の仕様から、2.章で示した合成法を用いて、RTレベルのハードウェア回路を合成する。

4.1 部品の選択

回路合成のためには、まず前節で導出したパラメータ条件式

に適切な値を割り当てておく必要がある。前章で述べた携帯電話のモデルでは、バスの本数やDSPの個数の変更に対応できる。そこで、バスの本数が1本と2本の場合、DSPの個数が1個と2個の場合に相当する各仕様に対してパラメータ条件式を導出した。今回の導出では、通常、携帯電話端末のCPUは各モジュールの制御を行うには十分すぎる能力を持っているので、CPUでDSPの機能を実現できるとし、DSP2個のときはCPU、DSPそれぞれをDSPとして使用し、DSP1個のときはCPUとDSPを共有化して使用するようにした(図5)。

(i) DSP1個のとき(バスの本数に依存せず)

$$\begin{aligned}
 &4Tba + 2Tbd + 4Tda \leq T \\
 &\text{and } 4Tbv + 2Tdv \leq T
 \end{aligned}$$

(ii) DSP2個、バス1本のとき

$$\begin{aligned}
 &4Tba + 2Tda + 2Tbd \leq T \\
 &\text{and } 2Tba + 2Tbd + 2Tda \leq T \\
 &\text{and } 4Tbv \leq T \text{ and } 3Tbv + Tdv \leq T
 \end{aligned}$$

(iii) DSP2個、バス2本のとき

$$\begin{aligned}
 &4Tba + 2Tda \leq T \\
 &\text{and } 2Tbd + 2Tda \leq T \\
 &\text{and } 2Tbv + Tdv \leq T
 \end{aligned}$$

以下、表1に示す部品の性能表に基づいて各パラメータの具体的な値を設定することを考える。メモリモジュールの性能については、ルネサステクノロジの携帯機器用メモリ[8]を、DSPモジュールの性能についてはTexas InstrumentsのDPS C54xシリーズ[9]を参考にした。

表1 機能部品の性能表

メモリモジュール			
	アクセス時間	転送速度	消費電力
(1)	100ns	10.0MB/s	60mW
(2)	70ns	14.2MB/s	180mW
(3)	55ns	18.2MB/s	180mW

DSPモジュール

	処理速度	消費電力
(1)	50MIPS	30mW
(2)	100MIPS	60mW
(3)	200MIPS	120mW

まず、バス幅を8bitとした場合のバスの帯域について考える。回路全体の動作周波数は、現在のFPGAに実装した場合、少なくとも50MHz以上になると考えられるが、メモリモジュールは表1より最速でも55ns(18.2MHz)でしか動作しない。このため、この携帯電話の例では、メモリモジュールの転送速度がバスの帯域となる。音声データは、3GPP(3rd Generation Partnership Project)の勧告番号TS 26.071として策定されたAMR-NBを想定する[10]。AMR-NBでは音声データを8KHz、13bitでサンプリングする。モデル化した並行周期EFSM群では1周期で音声データを送受信するため、そこで扱う音声データのサイズは、 $8 \times 13 \times T \text{ bit} = 13T \text{ bytes}$ となる。LCDの画面サイズは、ここではQCIF(176 × 144 pixels)とする。LCD1画面分の画像サイズは、 $176 \times 144 \times 3 = 76\text{Kbytes}$ となる。パケットデータの送受信では、1パケットサイズ1500bytesとする。最後に各データの符号化時間について考える。一般にAMRに必要な処理性能は20MIPS前後である。そのため、性能がnMIPSのDSPを用いた場合、T秒間の音声データを処理するには $20/n \times T$ 秒必要であるといえる。画像データのMPEG-4への符号化には、QCIFサイズ、15fpsでは約65MIPSが必要である。そのため、nMIPSのDSPを用いた場合、QCIFサイズの画像データ1枚を処理するには $65/15n$ 秒必要であるといえる。

以上をふまえ、上記の各機能部品の組合せに対して、具体的なパラメータ値をパラメータ条件式に代入する。代入した結果が表2である。

表 2 パラメータ条件式への適用結果

メモリ	DSP	条件式 (i)	条件式 (ii)	条件式 (iii)
(1)	(1)	<i>False</i>	$109ms \leq T$	$102ms \leq T$
(1)	(2)	$117ms \leq T$	$66.1ms \leq T$	$58.5ms \leq T$
(1)	(3)	$73.7ms \leq T$	$44.5ms \leq T$	$36.9ms \leq T$
(2)	(1)	<i>False</i>	$103ms \leq T$	$97.3ms \leq T$
(2)	(2)	$108ms \leq T$	$59.3ms \leq T$	$54.0ms \leq T$
(2)	(3)	$64.6ms \leq T$	$37.6ms \leq T$	$32.3ms \leq T$
(3)	(1)	<i>False</i>	$99.2ms \leq T$	$95.0ms \leq T$
(3)	(2)	$103ms \leq T$	$55.9ms \leq T$	$51.7ms \leq T$
(3)	(3)	$60.0ms \leq T$	$34.2ms \leq T$	$30.0ms \leq T$

察することができる。通話遅延が端末 1 台あたり 150msec 以内という携帯電話として要求される制約から、 $T \leq 150msec$ でなければならない。このとき、表 2 の結果より、表 1 の (1) の DSP を 1 個用いるだけでは要求される時間制約を満たせないことが分かる。また、15fps で動画像の処理をしたい場合には、周期が $1/15=66msec$ 以内でなければならないので、DSP 1 個で実現したい場合は、(2) か (3) のメモリと (3) の DSP の組み合わせでなければならないことが分かる。バスの本数については、回路規模と性能のトレードオフを考えると 2 本にするメリットは少なく、1 本で十分であると分かる。しかし、MPEG-4 に交換する画像サイズを大きくした場合 (例えば VGA (640 × 480) など)、処理は DPS ではなく専用ハードウェアにしないと追いつかなくなる (T_{dv} が大きくなる)。もし、専用ハードウェアが VGA で $T_{dm} = 10msec$ を実現したとしても、バスに流れる画像データの増加によって、アクセス時間が今の倍 (30msec) の高速なメモリモジュールを使ったとしても $T_{bv} = 27.6msec$ となる。パラメータ条件式より、バス 1 本の場合は $110msec \leq T$ 、2 本の場合は $65.2msec \leq T$ となり、バス 2 本にする優位性が見えてくる。並列に動作するモジュールが多いような仕様では、バスの本数の増減による性能の変化はより顕著になると考えられ、このようにバスの増減を簡単に行える仕様記述ができることは非常に有用であると考えられる (簡単のため、ここではバスの本数の増加による消費電力の増加は考えていないが、バスの増加による消費電力の増加の見積もりが出れば、回路全体の消費電力を考慮した部品の選択も可能である)。

このように、部品の性能・コストとめざす性能をもとにパラメータ条件式を満たす適切なパラメータ値を選択することで、実際に回路を合成する前に、適切と思われる部品の組が選択できるようにする。

4.2 回路合成

以下では、部品を共通化した場合とバスの本数を変更した場合のメインモジュールの動作周波数と回路規模を表 3.4 に載せる。

元の仕様と制約を付加した仕様の比較では、制約付加のためのモジュールがメインモジュールの動作周波数を下げることは無いということが分かった。

バスの本数を変更しても、動作周波数の点では、メインモジュールにほとんど差は無いことがわかる。これは、バスが 1 本の時にバスネックとなるような大きなデータを流さなければならない場合に、バスの本数を増やせば、バスネックを回避することができることを表している。一方回路規模の方は、バスを 1 本増やすたびに約 400LE 程度増加している。回路規模は小さいほうが良いので、この増加幅を減らすことは今後の課題である。

5. まとめ

本稿では、並行周期 EFSM 群を用いて実時間システムをモデル化する際に、複数部品の共通化やボトルネック解消のため

表 3 仕様を変更した場合の合成結果

	DSP/CPU 個別	DSP+CPU 共通化
動作周波数	101.77MHz	101.77MHz
回路規模	872LE	912LE

表 4 バスの本数を変更した場合の合成結果

	バス 1 本	バス 2 本	バス 3 本
動作周波数	101.77MHz	99.11MHz	95.63MHz
回路規模	872LE	1267LE	1737LE

の部品の並列化などの部品構成の変更に容易に対応できる仕様記述法を提案した。本手法、および我々が文献 [4] で提案した、パラメータ条件を生成するツールと指定したパラメータ値から実時間スケジューリング機構を含む RT レベル VHDL 記述を自動生成するツールを用いることによって、性能およびコストのバランスを考慮したハードウェアの設計・開発が可能になる。

提案手法をカメラ付携帯電話の基本動作部分の設計・実装に適用した結果、使用する DSP の個数やバスの本数などの部品構成の変更に対応でき、また、それぞれの部品構成に対してパラメータ条件を生成することで、目標の性能を達成するために、どのような性能の部品が必要かを効率よく決定することができた。

本稿では、複数部品を一つの部品に共通化する際に、元もとの部品に対する処理が任意の順序でシーケンシャルに実行されるための制約を付け加えることで対処した。今後の課題として、より高速な回路を得るため、共通化した部品に対する処理の効率よいスケジューリング法を考案する予定である。

文 献

- [1] ISO. "Information Processing System, Open Systems Interconnection LOTOS," *ISO 8807*, 1989.
- [2] ISO. "Final Committee Draft 15437 on Enhancements to LOTOS," *ISO/IEC JTC1/SC21/WG7*, 1998.
- [3] H. Katagiri, M. Kirimura, K. Yasumoto, T. Higashino and K. Taniguchi. "Hardware Implementation of Concurrent Periodic EFSMs," *Proc. of Joint Int. Conf. on 13th Formal Description Techniques and 20th Protocol Specification, Testing and Verification (FORTE/PSTV2000)*, pp. 285-300, 2000.
- [4] T. Kitani, Y. Takamoto, I. Naka, K. Yasumoto, A. Nakata and T. Higashino, "Design and Implementation of Priority Queuing Mechanism on FPGA using Concurrent Periodic EFSMs and Parametric Model Checking," *Proc. of 13th Int. Conf. on Field Programmable Logic and Applications (FPL 2003)*, LNCS Vol.2778, pp.1145-1148, 2003.
- [5] A. Nakata, and T. Higashino, "Deriving Parameter Conditions for Periodic Timed Automata Satisfying Real-Time Temporal Logic Formulas," in *Proc. of 21st IFIP Int. Conf. on Formal Techniques for Networked and Distributed Systems (FORTE2001)*, pp. 151-166, 2001.
- [6] C. A. Vissers, G. Scollo, M. v. Sinderen, and E. Brinksma: "Specification Styles in Distributed Systems Design and Verification," *Theoretical Computer Science*, vol. 89, no. 1, pp. 179-206, 1991.
- [7] T. Yen and W. Wolf, "Performance Estimation for Real-Time Distributed Embedded Systems," *IEEE Trans. on Parallel and Distributed Systems*, vol. 9, no. 11, pp.1125-1136, 1998.
- [8] 株式会社ルネサス テクノロジ, "専用 IC," available at <http://www.renesas.com/jpn/products/assp/>
- [9] テキサス・インスツルメンツ株式会社, "DPS 製品," available at <http://www.ti.co.jp/jsc/docs/dsps/>
- [10] CQ 出版株式会社, "Design Wave Magazine," available at <http://www.cqpub.co.jp/dwm/>