C-006

システムレベル設計のシーケンス制御システム設計への応用

System Level Design Applied to Sequence Controller Systems

小林憲貴† 吉田紀彦† 楢崎修二‡
Kazutaka Kobayashi Norihiko Yoshida Shuji Narazaki

1. はじめに

SoC(System-on-Chip)の設計において生産性ギャップを埋めるべく開発されたシステムレベル設計は，ソフトウェア(SW)とハードウェア(HW)の協調設計を可能にする手法である[1][2][3]．このシステムレベル設計は，高位の抽象モデルからの設計が可能であるという優れた特徴を持つ．

我々は以前にASIC(遺伝的アルゴリズムVLSI)をHW記述言語(SFL)とシステムレベル設計言語(SpecC)で設計・比較検討し，そのシステムレベル設計の設計効率を検証した[4]．この検証により，システムレベル設計の特徴の有効性を確認した．

そこで我々は，現在，低位抽象度の設計理論を用いて設計が行われている分野においても，システムレベル設計の応用が有効であると考え，これまでSoC設計のみに考案，使用されてきたシステムレベル設計の他分野への応用を検証する．

本研究では，このシステムレベル設計のシーケンス制御システム[5][6][7]への応用を考える．つまり，システムレベル設計の各段階にシーケンス制御システムを設計するための手法を導入する．本論文では，システムレベル設計導入の第一歩として，システムレベル設計の最終段階である実装モデルの作成（ライブラリの完備，シミュレーションによる動作確認）に焦点をあて，実験・評価を行う．第一歩として実装モデルに焦点をあてた理由は，システムレベル設計の仕様や，アーキテクチャ探索の段階がシーケンス制御システムへ応用できたとしても，実装の段階が設計できなければ，システムレベル設計がシーケンス制御に応用ができないからである．つまり，本論文の実験・評価では，シーケンス制御システムの設計において，我々が実装段階に導入する手法の有効性を確認し，それにより，システムレベル設計のシーケンス制御システム全体への応用の可能性を検証する．

設計言語にはC言語ベースのSpecCを採用した．その理由は，合成ツールがライブラリ・ベース（SystemCなど）に比べ理解しやすいことに加え，C言語ベースであるため馴染み深いからである．

2. 関連研究

システムレベル設計の他分野への応用例として，カリフォルニア大学アーバイン校で電力システムとそのエミュレータのSpecCによるシステムレベル設計が行われている[8][9]．この２つの研究は，システムレベル設計をSoC設計ではなく他分野に応用する研究であり，本研究と同じ目的であるが，応用分野が電力システムとエミュレータなので，本研究とは応用分野が異なる．他の応用例として(株)インターデザイン・テクノロジーで自動車の制御システムの検証，解析に応用した“車載ネットワークシステムの高速・高精度シミュレータ Venet”というグラフィカルツールが存在する[10]．

3. シーケンス制御

シーケンス制御は自動制御の一種で，予め定められた手順に従って，入力に応じてシステムの状態を制御するものである．シーケンス制御システムの設計フローはSoCの設計フローに似ているが，詳細化の段階でSoCがSWとHWに切り分けられるのに対して，シーケンス制御システムはSWによる制御とPLCによる制御に切り分けられるところが異なる．よって，PLCによる制御のシステムレベル設計が可能であるならば，シーケンス制御システム全体のシステムレベル設計が可能となる．ここでPLCとは，シーケンス制御に適したHWであり，PLC独自の機能，制約を持つ．またPLC上で実際の処理を行うプログラムをラダープログラムという(図１)．ラダープログラムは論理演算と類似しており，接点(入力)とリレーコイル(出力)とその接続関係によってプログラミングされる．

4. シーケンス制御システムとSoCの相違点

PLCはSoCのHW論理とは異なる特徴を持ち，それがPLCのシステムレベル設計を行う際にも反映される．その特徴を以下に挙げる．

1：入力専用素子(接点)の出力端子への接続禁止
2：接点に未接続である出力専用素子(リレーコイル)の使用禁止
3：出力素子の使用回数制限
4：入出力素子の個数制限
5：反復処理(1回の反復を1スキャンと定義)
6：接続による論理演算の区別
7：PLC特有の特殊な機能を持つ素子の存在
8：並列処理プログラムから順序回路の作成

また，上記の７には以下の特殊機能素子がある．

・時間の概念
この概念を持つ代表的な素子は，信号が設定時間以上入力されると動作を行うという素子である．
・信号微分
代表的な素子として立上り微分がある．これは入力信号の立ち上がり時に１スキャンの間信号を出す素子である．

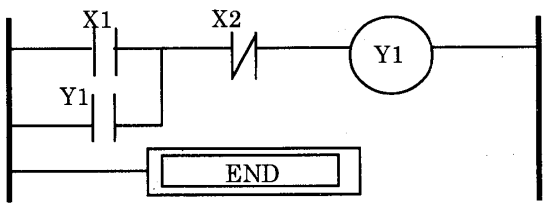

図１ ラダーロジック(自己保持回路)

† 埼玉大学 Saitama University
‡ 長崎大学 Nagasaki University

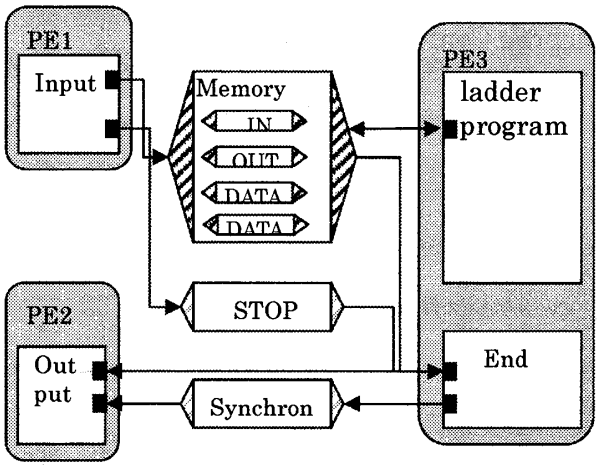

図２　PLC 環境

・カウンタ機能
内部にカウンタを持ち，信号の入力回数が設定回数以上になると信号を出力する素子である．また，外部からカウンタのリセットが可能である．
・内部的入出力素子
PLC の外部に対して入出力ができない内部的な入出力素子である．

5. シーケンス制御のシステムレベル設計

一般的なシステムレベル設計は以下のような流れになる．
① 要求を満たす実行可能な仕様モデルの作成
② 仕様モデルの動作部分を具体化したモデル作成
③ ②と同様に通信部分を具体化したモデル作成
④ クロック精度の実装モデルの作成

このシステムレベル設計をシーケンス制御システムに応用する際，各段階に導入される手法を詳述すると以下のようになる．
①仕様モデルの作成
シーケンス制御システムの設計では，所望のシステムの要求分析を行う前に，PLC の機能，制約を分析する必要がある．これは４章で述べた 1～5 に対応し，設計ルールとして定められる．
②アーキテクチャ探索
ここでは，最終実装がラダープログラムであることを念頭に置いた上で具体化を行う．言い換えると，ラダープログラムの最適化である．ここでは４章の６に対応する必要がある．理由は，アーキテクチャ探索では様々な接続パターンが考えられるため，評価・見積もりを行いながらモデルの形態を決定する必要があるからである．またこの段階で，SW/PLC の切り分けを評価するが，PLC の設計可能な範囲，PLC で設計すべき範囲を理解した上で，システムの要求に合う切り分けを行う必要がある．
③通信合成
システムレベル設計における通信合成の手順は，PLC による制御の設計においては大幅に削減可能であると考えられる．これはラダープログラムが PLC に付随する SW であり，

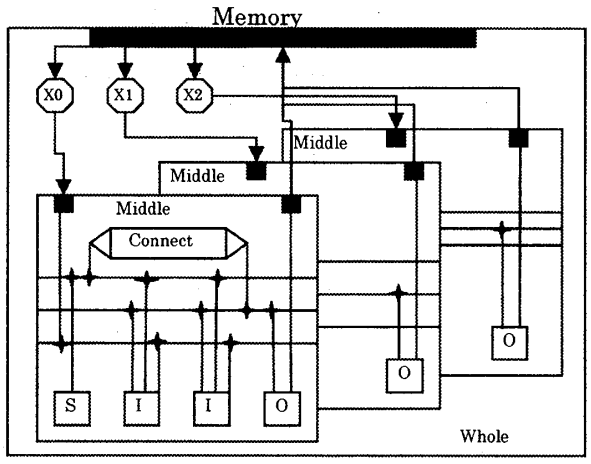

図３　モジュール図

ラダープログラムの素子間の通信は PLC によって制御されるため詳細化する必要がなくなるためである．
④実装モデルの作成
ここで得られる実装モデルは，ラダープログラムと１対１に対応する．つまり，実装モデル作成には PLC 素子のライブラリが必要となるため，４章の７にある特殊素子にも対応しなければならない．この特殊素子は，上位の段階では容易に設計可能であるが，実装段階ではライブラリとして作成する必要がある．さらに，４章の８に対しては，システムレベル設計の並列処理を FSM に変換する手順と対応しており，PLC の詳細な処理手順が FSM で表現できるため容易である．また，シーケンス制御では SoC 設計のようにクロック精度を考慮する必要はない．この実装モデル作成が可能か否かによって，システムレベル設計のシーケンス制御システムへの応用が可能であるかを検証することができる．

以上によりシーケンス制御システムのシステムレベル設計を行うが，実際には作成した実装モデルをラダープログラムへ変換するコンパイラの設計が必要となる．

6. 設計

6. 1　PLC 環境設計

本研究では，忠実にラダープログラムを再現するため PLC の仮想 HW を SpecC で構築した．具体的には，PLC の内部コンポーネント（入力ユニット，出力ユニット）とメモリ，ラダープログラムに対応する処理プログラムをマルチスレッドアプリケーション（図２）として作成した．このように仮想 HW を作成したことによって，PLC の出力におけるスキャン単位のズレを表現することが出来る．この出力におけるズレとは，PLC のメモリアクセスを伴う入出力にリフレッシュ方式（一括入出力）とダイレクト方式（必要時の入力，値の変更時の出力）があり，その違いによって生じる出力のズレを表す．

6. 2　処理モジュール

ラダープログラムに対応するモジュールは三階層に設計した（図 3）．それぞれの階層は上位層から，処理全体を表すモジュール，一連の処理を担当する中間モジュール，ラ

ダープログラムの素子に１対１に対応するリーフモジュールである．また上記三層の他に，PLCの処理プログラム（図２のladder program）内の通信部分は，チャネルを用いて設計した．チャネルとは，演算と通信を切り分けるために通信をカプセル化するものである．これらのモジュール，チャネルの全てをライブラリ化した．以下に各モジュールを詳述する．

・全体を表すモジュール

このモジュールはプログラム作成のためのフレームワークを提供するものである．そのため，メモリアクセスのためのポートを持つ．これはPLCでは，プログラムとデータが異なるメモリに格納されており，そのメモリ間で通信を行っているのを表現するためである．また，処理内でカウンタを使用する場合はモジュール内にカウンタリセットチャネルの追加を行う必要がある．これはカウンタモジュールとカウンタリセットモジュールが異なる中間モジュール内に存在することがあり，そのため，その中間モジュール間での通信を行う必要があるからである．

・中間モジュール

このモジュールではリーフモジュール間の接続とメモリアクセスを管理する．

モジュール間の直列接続は実際には配線のみであるが，改良・拡張を容易に行うために，我々はチャネルで実装し，直列接続を表現した．

一方で，並列接続の再現には，３つのパターンが考えられる．第１にチャネル内部にorまたはORメソッドを持たせ，そのチャネルからデータを引き出す素子と組み合わせるパターン．第２に仮想的なorまたはORモジュールを作成し，接続チャネルの個数を増加させるパターン．第３に第２と同様に仮想的orまたはORモジュールを作成し，チャネルが複数接続できるようにチャネル内のメモリを増やすパターンである．どのパターンにおいてもアーキテクチャ探索において見積もり，評価が必要となる．

なお，メモリアクセスにおいてはリーフモジュールの引数を整合させる必要がある．またリーフモジュールに特殊素子がある場合，それぞれを初期設定する必要がある．

・リーフモジュール

ラダープログラムの各素子と対応するモジュールをリーフモジュールとする．これは，変更せずに利用可能なライブラリとして用意した．リーフモジュールとして以下に示すように７つ作成した．

1：a接点(通常入力)　　5：Pulse（信号微分）
2：b接点(否定入力)　　6：カウンタ
3：出力リレー　　7：スタート
4：Timer（時間の概念）

1，2は前の素子からの信号と素子に接続された入力信号のandまたはANDを出力する．3は入力データをメモリに書き込む．書き込むタイミングは出力方式ごとに異なる．4～6の特殊素子は高抽象度設計をカプセル化することで実現する．カウンタ素子では，カウンタのリセットを外部から操作可能とするためリセット用のチャネルを高位の階層に作成した．7は処理開始を示すとなる仮想的なモジュールである．

6．3　スケジューリング

まず，処理プログラム（図２中のladder program）のスケジューリングは，全体・中間モジュールの内部をFSMでモデル化することによって，ラダープログラムと等しいスケジュールとなる．

ラダープログラムの反復表現は，PLCの停止ボタンを設定し，状況に応じて各コンポーネント（図２中のPE1～3）が割り込み処理を行うことで実現できる．処理プログラムと出力ユニットでは同期が必要となることから，ボタン状況の確認と出力ユニットとの同期を行うendまたはENDモジュールを処理コンポーネント(PE3)内に作成した．PLCは反復によって処理を実現するため，Endモジュールでは１スキャン終了後，次スキャン開始のための割り込み信号を発信する．割り込み処理は上位の処理コンポーネントで行う．

7．シミュレーション

シミュレーションは，(株)インターデザイン・テクノロジーのVisualSpecを使用して，図１の自己保持回路と，３入力（センサ２個とスタートボタン）によるベルトコンベア２個の制御をシミュレートした．

まず，我々が構築した仮想HWについて，コンポーネント間の同期やイベントのタイミングなどHWとしての通信動作を確認し，SWが行うメモリアクセスによって生じる出力データのスキャン単位でのズレがPLCで生じるズレと等しくなることも確認した．このズレとは，6.1に記述した４つの入出力方式の組み合わせにより，出力データのスキャン単位で生じるズレである．次に処理モジュールについては，メモリアクセス，モジュール間通信，各リーフモジュールの動作の確認を行った．いずれの場合においても

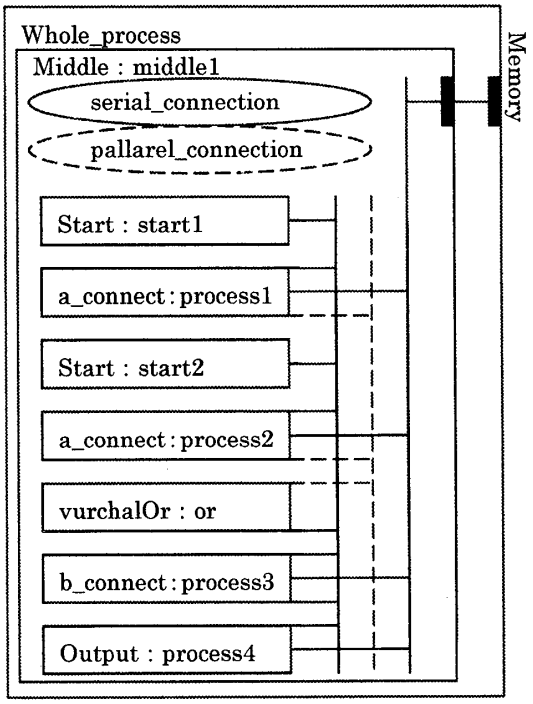

図４：自己保持回路のモデル図

VisualSpecのグラフィックツールにより容易にシミュレーションを行うことが出来た．

図４に自己保持回路のモデル図を示す．

8. 他の段階の改良

本論文では，システムレベル設計の実装段階でのシーケンス制御システムへの応用を実験・評価した．次に他の段階でのシーケンス制御への応用を考える．

仕様：最も抽象度が高いモデルの作成であるため，５章で述べた設計ルールを制限事項として追加するだけで良いと考えられる．

アーキテクチャ探索：ラダープログラムの最適化を目標とする．これは，６章で述べた実装段階の中間モジュールとなる処理毎に分割し，評価・見積もりを行う必要がある．PLCとSWの切り分けについては，シーケンス制御システムに精通した者が行った方が良いが，複雑な演算以外はPLCで制御可能と思われる．

通信合成：手順の大幅な削減を行う．これは５章で述べたように，ラダープログラムで表現される処理間の通信はPLCによって定められており，設計者は気にする必要がないからである．つまり，システムレベル設計の通信合成で行われる手順である，異なる通信プロトコル間の橋渡しをするトランスデューサの合成は不要となる．また，処理間の通信をチャネルで表現できるため，インライン化も不要である．しかし，PLCにおける通信が単純な通信であることから，チャネル内のメソッドを処理として作成し，マスタ・スレイブ方式のプロトコルの挿入を行う．

以上のことが現在考えられる．

9. おわりに

本研究では，システムレベル設計によるシーケンス制御システムの設計のへ応用を行っている．つまり，シーケンス制御システムへの応用に必要な手法の導入である．本論文では，ラダープログラムに対応する実装モデルの作成の可能性を検証のため，実際に仮想HWを構築し，実験・評価を行った．その結果，実装モデルの作成が可能であったため，シーケンス制御システムへのシステムレベル設計の応用における可能性の第一歩を見極めることができた．今後，仕様設計，その後の合成フローを手法に則って作成していき，最終的には大規模プラントの設計で検証を行う予定である．

参考文献

[1] Daniel D. Gajski, Jianwen Zho, Rainer Domer, Andreas Gerstlauer, Shoqing Zhao 著，木下常雄，富山宏之訳，“SpecC，仕様記述と方法論”，CQ出版社(2000)

[2] Andreas Gerstlauer, Rainer Domer, Junyu Peng, Daniel D. Gajski 著，木下恒夫訳，“システム設計，SpecCによる実現”，SpecC Technology Open Consortium(2002)

[3] http://www.ics.uci.edu/~specc/

[4] 小林，VLSIシステムレベル設計の制御システム設計への応用，長崎大学大学院修士論文（2004）

[5] 関口隆志著，“新しいプログラマブルコントローラのプログラミング，IEC 61131-3による効率的プログラミング”，コロナ社(1999)

[6] 熊谷秀樹著，“シーケンス制御プログラム定石集”，日刊工業新聞(2003)

[7] 熊谷英樹著，“ゼロからはじめるシーケンス制御”，日刊工業新聞(2001)

[8] Slim Ben Saoud, Daniel D. Gajski, Andreas Gerstlauer, “Co-design of Enbedded Controllers for Power Electronics and System”, International Synposium on Intelligent Control, 2002

[9] Slim Ben Saoud, Daniel D. Gajski, Andreas Gerstlauer, “Co-design of Emulators for Power electric Processes Using SpecC Methodology”, 28th Annual Conference of the IEEE Industrial Electronics Society, 2002

[10] http://www.interdesigntech.co.jp/vn1209.htm