PLC 命令列を論理回路に変換するツールの実装と評価

指導教員：市川 周一　　学籍番号：033701　秋中 昌訓

1　はじめに

産業用機械などのシーケンス制御には，PLC (Programmable Logic Controller) と呼ばれる制御用の計算機が広く利用されている．PLC を用いると制御論理を柔軟に変更できるが，大規模な制御では処理速度の不足が問題になる場合がある．また，PLC プログラムはソフトウェアであるため複製や解析が容易で，技術情報の流出やコピー製品の出現が懸念される．

近年，FPGA (Field Programmable Gate Array) を利用した PLC プログラムのハードワイヤード化が検討されている [1]．FPGA は再構成可能な LSI であるため，PLC の柔軟性を保ったまま処理速度の改善が可能である．また，ハードウェアである FPGA の解析は PLC プログラムの解析より困難であるため，近年の FPGA の秘匿性向上技術とあわせて，知的所有権の保護にも有効であると考えられる．

本研究では，既存の PLC プログラムの蓄積を活用し，FPGA への移行を容易にするため，PLC 命令列をハードウェア記述言語 VHDL に変換するツールを作成した．さらに，実応用の PLC プログラムを用いて FPGA 上で評価した．本研究の概要については既に報告している [2].

2　PLC プログラムの論理回路への変換手法

PLC プログラムの記述には，ラダー図が広く利用される．ラダー図表現における処理の最小単位を段と呼び，段は条件部と処理部から構成される (図 1)．処理部には，(a) 出力命令や (b) データ処理命令などが設定可能である．PLC プログラムは複数の段から構成され，それらが上から下へ繰り返し処理されることでシーケンス制御を実現している．このとき，1 回の実行時間をスキャンタイムと呼ぶ．スキャンタイムが短いほど高速なシステムといえる．

PLC の実行制御方式を論理回路で忠実に再現するには，ラダー図の各段に順序回路の 1 状態を割り当てて，各段を逐次処理すればよい．この設計を Sequential Design (SD) と呼ぶ．

SD を高速化するには，PLC プログラムの各段の依存関係を適切に維持しながら，並列実行可能な制御回路を並列に動作させればよい．依存関係の例を図 2 に示す．ある段の出力 Y002 を下の段で利用する場合，下の段は Y002 の値が定まるまで実行できない (データ依存関係)．そこで依存関係に従って各段のレベル付け行い，各レベルを 1 状態として順序回路を生成する．この設計を Levelized Design (LD) と呼ぶ．

しかし，依存性さえ守れば各レベルに 1 状態を割り当てる必要はない．LD のレベル間の記憶素子を除去してラダー図を組合せ回路で生成し，毎スキャンの終了時に入出力の更新を行うように変換することができる．この設計を Flat Design (FD) と呼ぶ．FD では，基本的に 1 スキャンを 1 状態で処理するが，プログラム中に外部機器 IO 命令が含まれている場合には事情が異なる．外部機器への入出力は BFM (バッファメモリ) を用いて即座に行われるため，FPGA では外部機器番号及びそれぞれの機器の BFM アドレスを用いてシングルポート RAM へのアクセスと同様に実装した．そのため外部機器 IO 命令があった場合，メモリへの入出力を行うための状態遷移が必要となり，FD であっても 1 つの状態での実行はできない．

以上の設計では演算命令と同数の演算器を生成していた．しかし，SD，LD では全ての演算器が同時に動作するわけではない．同時に動作しない演算器は，マルチプレクサやバスで共有することにより，資源の膨張を抑止できる可能性がある．そこで，本変換ツールでは大規模な制御論理向けに，資源制約を行うオプションを用意した．演算器の生成数に上限を設け，共有して使用することにより，回路規模の増加を抑制する．

3　ツールの実装と評価

2 節で検討した変換手法を実装し，企業から入手した Mitsubishi FX2N PLC 向けのサンプルプログラムを用いて評価を行った．ターゲットデバイスは StratixII EP2S60F672C5ES (48,352 ALUTs) とし，Altera Quartus II 6.0SP1 で論理規模と動作周波数を見積もった．論理合成のオプションはデフォルト (バランス優先) とした．演算命令 1 つに演算器を 1 個生成する回路を “固有”，資源制約を設けて各演算器 (加減算，乗算，除算) の最大生成数を 1 とし共有する回路を “共有” と示した．FPGA 上の実行時間 (スキャンタイム) は，クロックサイクル時間と状態遷移数の積で求めた．また，PLC 上での実行時間はすべての段の条件部で条件が成立したときの時間 (worst case) とした．

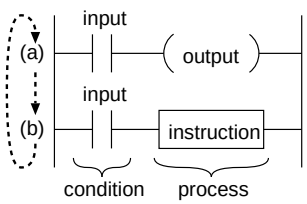

図 1: ラダー図の例

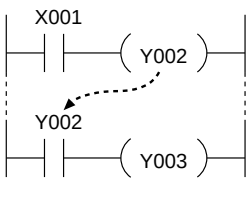

図 2: データ依存の例

表 1: A 社サンプルプログラムの評価結果

デバイス	設計	演算器	状態遷移数	動作周波数 [MHz]	論理規模 [ALUTs]	実行時間 [sec.]
PLC	—	—	—	—	—	1.61×10^{-3}
FPGA	SD	固有	74	8.21	5,848	9.01×10^{-6}
		共有	74	6.10	2,863	1.21×10^{-5}
	LD	固有	12	7.57	5,686	1.59×10^{-6}
		共有	17	6.88	2,716	2.47×10^{-6}
	FD	固有	1	4.67	4,625	2.14×10^{-7}

表 2: 八洲熱学サンプルプログラムの評価結果

デバイス	設計	演算器	状態遷移数	動作周波数 [MHz]	論理規模 [ALUTs]	実行時間 [sec.]
PLC	—	—	—	—	—	4.64×10^{-2}
FPGA	SD	固有	415	16.08	3,554	2.58×10^{-5}
		共有	415	8.04	4,148	5.16×10^{-5}
	LD	固有	76	16.02	2,864	4.74×10^{-6}
		共有	76	7.81	3,445	9.73×10^{-6}
	FD	固有	63	13.64	2,643	4.62×10^{-6}

表 1 はある会社 (A 社) で用いられている PLC プログラムの評価結果である．本プログラムは 165 命令で構成され，6 個の加減算命令と 12 個の乗算命令及び 9 個の除算命令が含まれる．演算処理を中心としたサンプルプログラムである．

まず LD(固有) では SD(固有) に対して 5.7 倍高速化し，FD では SD(固有) に対して 42.1 倍の高速化を達成した．本プログラムは複数の出力に対して制御を行っているため，処理の並行度が高く，レベル付けによる並列化で状態遷移数が大きく減少する (最大で 84%減少)．また演算器を共有した場合，論理規模は SD で 51%，LD で 48%減少した．

表 2 は，八洲熱学株式会社と共同開発中の整列巻取機の制御プログラムを評価した結果である．本プログラムは 1303 命令で構成され，5 個の加減算命令と 11 個の乗算命令及び 2 個の除算命令が含まれる．演算命令が全体に占める割合は少なく，制御を中心とした実応用のサンプルプログラムである．

まず LD(固有) では SD(固有) に対して 5.4 倍高速化した．これは A 社のプログラムと同様に並列度の高い制御を行っているためである．FD では SD(固有) に対して 5.6 倍高速化したが，LD(固有) と大差ない結果となった．これは外部機器 IO 命令が多いため，入出力用の状態遷移が必要で，FD でも状態遷移数は大きく変わらない (LD の 83%程度) ためである．また演算器を共有した場合，論理規模は SD で 17%，LD で 20%増加してしまった．これは，共有のためにマルチプレクサや，汎用的演算器が生成されることが原因であると考えられる．本プログラムのように演算の占める割合が少なく，計算する値に定数が多く含まれる場合，専用演算器を生成する方が論理規模は小さくなることがある．

4　おわりに

今回提案した 3 つの設計方法の中では FD が最も大きな高速化が得られた．ただし外部機器 IO 命令が多く含まれる場合，状態遷移数の削減が制限されるため，FD による効果は制約される．また，演算器の共有で論理規模の削減が期待できるが，プログラムの性質次第で効果の大小が大きく影響された．

今後は，提案手法を実際の機械に組み込んで効果を確認し，さらに実用的な応用に耐えるような環境整備と機能拡張を行いたい．

参考文献

[1] I. Miyazawa, T. Nagao, M. Fukagawa, Y. Itoh, T. Mizuya, and T. Sekiguchi: “Implementation of ladder diagram for programmable controller using FPGA,” in *Proc. 7th IEEE Int’l Conf. Emerging Technologies and Factory Automation (ETFA ’99)*, vol. 2, 1999, pp. 1381–1385.

[2] S. Ichikawa, M. Akinaka, R. Ikeda, and H. Yamamoto: “Converting PLC instruction sequence into logic circuit: A preliminary study,” in *Proc. IEEE Int’l Symp. Industrial Electronics (ISIE ’06)*, 2006, pp. 2930–2935.