

平成 17 年 1 月 13 日

専攻	知識情報工学専攻	学籍番号	013749	指導教官氏名	市川 周一
申請者氏名	山本 浩司				

論文要旨（修士）

論文題目	FPGA による実時間制御システムに関する研究
------	-------------------------

産業用機械などのシーケンス制御にはプログラマブル・ロジック・コントローラ (PLC) が多く用いられている。しかし、規模の拡大、要求精度の高まりに伴って PLC の速度不足が問題になりつつある。また、PLC プログラムはソフトウェアであるため解析やコピーが容易で、技術情報の流出やコピー商品の出現などの問題を招くことがあった。

近年、FPGA (Field Programmable Gate Array) を用いて制御プログラムを論理回路化する手法 (ハードワイヤード化) が検討されはじめた。FPGA はプログラム可能な LSI であるため生産現場で任意に論理を変更することができ、1 チップで数十～数百万ゲートを集積しているため大規模な制御論理も小面積で実装することができる。部品点数の削減による低コスト化も期待できるため、高性能を必要とする大規模システムだけでなく、安価な小規模システムにも応用可能である。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、制御論理の秘匿性も高い。

本研究では、制御システムを実現するための論理ライブラリを整備し、実際の制御回路を FPGA 上に実装・評価することを目的とする。

PLC プログラムはラダー図で記述されることが多く、ラダー図はリレー回路が原型であるため論理回路で実現可能である。しかし、現在の PLC は一種の計算機であるため、単純な論理回路だけでは実現することができない。また、制御システムは PLC 以外に多くの周辺回路から構成されるため、周辺回路まで FPGA 上に実装しなければ実用的システムを構成することができない。多く用いられる制御論理も、周辺回路と共にライブラリとして整備する必要がある。

市川研究室では、平成 15～16 年度に八洲熱学 (株) との共同研究で整列巻取機を試作した。本研究では、実際の制御論理の例として、この整列巻取機の制御論理の一部を FPGA に実装・評価した。論理を VHDL で記述し、Altera 社の QuartusII4.0 で FPGA の論理を設計した。ターゲットデバイスには Altera 社の EP20K200EFC484-2X を指定した。その結果、回路規模は 430LE、動作周波数は 50.38MHz となり、充分実用可能な規模と速度の制御論理が得られた。

実際に PLC と周辺回路による制御論理を FPGA ボードに置き換え、動作を確認した。これにより、FPGA による小型・高性能・高秘匿性制御論理の実現への第一ステップが実現されたといえる。本研究では、現実の制御論理を FPGA 上に実装し、動作を確認することができた。しかし現状では制御論理ライブラリの整備が十分でないため、さらに整備を進めていく必要がある。また、本研究で扱った制御論理は小規模なものであるため、より大規模な応用に関しても FPGA 技術が適用可能であるか検討を進めてゆきたい。

Name	Hiroshi YAMAMOTO	Registration Number	013749
Department of Knowledge-based Information Engineering			
Graduate Adviser	Shuichi ICHIKAWA		
Keywords	sequence control, logic control, FPGA		

Abstract (Master)

Title	A Study on Real-Time Control System with FPGA
-------	---

Programmable logic controllers (PLCs) are widely adopted for sequence control of industrial machinery. However, the performance limitation of PLC is recently becoming a serious problem, where higher precision and larger scale control are required. Another problem is that PLC program is rather easy to duplicate and to analyze. This often results in the leakage of valuable trade secrets and the rise of clone products in market.

Recently, it has been proposed to implement control programs in hard-wired logic by using reconfigurable devices such as FPGA (Field Programmable Gate Array), which can be arbitrarily re-programmed in actual application fields. An FPGA chip contains maximally ten million logic gates, and thus can contain a very large control system in a small chip. This may lead to reducing the number of components, which may lower the cost. Therefore, the application of FPGA technology is not limited to a large-scale applications that require high performance, but it is also applicable to small systems. It should be also noted that FPGA technology is better than PLC in protecting intellectual properties and trade secrets, because it is more difficult to analyze an FPGA hardware than to analyze a PLC software.

This study describes the implementation of control logic library, which is required to integrate a whole control system in a FPGA chip. A sample implementation of FPGA control system is also presented.

PLC programs are often described in ladder diagram, which originated from relay circuit and thus can be implemented in logic circuit. However, a modern PLC is a kind of computer, and it is not so easy to implement an actual PLC program in hardware. Even if PLC program could be implemented in an FPGA, it does not work without other peripheral devices. Hence, it is practically essential to prepare a library of peripheral devices and popular control primitives for FPGA implementation. The author implemented several functions for control library, but only one device is introduced here.

Ichikawa laboratory developed an experimental winder in a joint research project with Yashima Netsugaku Co. Ltd. In this study, the author implemented the simplified control logic of this experimental winder on an FPGA board. The abovementioned logic was described in VHDL and implemented on an FPGA by using Altera QuartusII 4.0 software for an EP20K200E device. The derived circuit costs 430 LE (logic elements) in logic scale, which is only 5% of an EP20K200 device. The maximal operational frequency was 50.38 MHz, which was higher than the on-board system clock (33.3 MHz).

An example of FPGA control logic was actually implemented and successfully operated. This would be the first step to the application of FPGA to control circuits. Since the functions of control logic library are not yet satisfactory, they have to be extended still more. Though this study examined a small control system, larger systems should be examined and evaluated on FPGA chips.

FPGAによる実時間制御システムに関する研究
A Study on Real-Time Control System with
FPGA

豊橋技術科学大学大学院 工学研究科 知識情報工学専攻
学籍番号 013749 氏名 山本浩司

指導教官 市川周一

目次

1	はじめに	1
1.1	背景	1
1.2	関連研究	1
1.3	目的	2
2	実現手法	3
2.1	制御論理ライブラリ	3
2.2	パルスジェネレータ (PG)	3
2.3	STPG (ステッピング用パルスジェネレータ)	4
2.3.1	評価	7
2.4	TRPG (トラバース用パルスジェネレータ)	9
2.4.1	評価	10
2.5	PLCでよく用いられる命令	11
2.6	変換案	12
2.6.1	リレー ON	12
2.6.2	計算	12
2.6.3	リレーセット	13
2.6.4	タイマーリレー	15
2.6.5	データ転送	16
3	実装	17
3.1	試作機の制御論理	17
3.2	実装時の変更部分	19
3.3	実装した制御論理	20
4	評価	22
4.1	評価環境	22
4.2	評価条件	22
4.3	結果	23
5	おわりに	25
A	パルスジェネレータ	26
A.1	STPG (ステッピング用パルスジェネレータ)	26
A.1.1	入出力	29
A.1.2	信号線の定義	29
A.1.3	内部パラメータ	31
A.1.4	設計	33

A.1.5	評価	35
A.2	TRPG（トラバース用パルスジェネレータ）	37
A.2.1	入出力	38
A.2.2	信号線の定義	39
A.2.3	設計	42
A.2.4	評価	44
B	PLC 制御論理の実装	45
B.1	現実の PLC 制御論理	45
B.2	試作機の制御論理	45
B.3	実装時の変更部分	45
B.4	実装した制御論理	47
B.4.1	設計	47
B.4.2	信号線などの定義	49
B.4.3	評価	53
B.4.4	FPGA ボードと各ドライバの接続	55

図 目 次

1	STPG のブロック図	4
2	内部パラメータの関係と STPG の動作	5
3	TRPG のブロック図	9
4	TRPG の動作	9
5	リレー ON（サンプルパターンと変換案）	12
6	計算（サンプルパターンと変換案）	13
7	PLS, PLF（サンプルパターン）	13
8	PLS, PLF の動作	14
9	1 スキャン ON（PLS, PLF）（変換案）	14
10	SET, RST（サンプルパターン）	14
11	SET, RST（変換案 1）	15
12	SET, RST（変換案 2）	15
13	タイマーリレー（サンプルパターン）	15
14	タイマーリレー（変換案）	15
15	データ転送（サンプルパターン）	16
16	データ転送（MOV）（変換案）	16
17	データ転送（DMOV）（変換案）	16
18	サンプル 3 のブロック図	17
19	サンプル 3 改のブロック図	20
20	STPG のブロック図	26

21	内部パラメータの関係と STPG の動作	28
22	STPG の入出力	29
23	STD の入出力	29
24	内部パラメータの関係と STPG の動作	32
25	STPG のブロック図 (詳細)	33
26	STPG のステート図	34
27	TRPG のブロック図	37
28	TRPG の動作	37
29	TRPG の入出力	38
30	TRD の入出力	38
31	TRPG のブロック図 (詳細)	42
32	サンプル 3 のブロック図	45
33	サンプル 3 改のブロック図	47
34	サンプル 3 改の制御部分のブロック図	48
35	サンプル 3 改の制御部分と各 PG, ドライバの接続	52
36	FPGA ピンと STD の接続	55
37	FPGA ピンと TRD の接続	56
38	FPGA ピンとドライバの接続の回路図	57

目 次

1	評価環境 (PC)	7
2	ターゲットデバイス	7
3	STPG の回路規模と動作周波数	8
4	TRPG の回路規模と動作周波数	10
5	評価環境 (PC)	22
6	ターゲットデバイス	22
7	サンプル 3 改の回路規模と動作周波数	23
8	評価環境	35
9	ターゲットデバイス (Nios 開発ボード)	35
10	STPG の回路規模と動作周波数	36
11	sel_fr と sel_lab の関係	40
12	TRPG の回路規模と動作周波数 (上段: 通常版, 下段: 機能限定版)	44
13	評価環境 (PC)	53
14	ターゲットデバイス	53
15	サンプル 3 改の回路規模と動作周波数	54

1 はじめに

1.1 背景

産業用機械などのシーケンス制御にはプログラマブル・ロジック・コントローラ (PLC) が多く用いられている。しかし，規模の拡大，要求精度の高まりに伴って PLC の速度不足が問題になりつつある。また，PLC プログラムはソフトウェアであるため解析やコピーが容易で，技術情報の流出やコピー商品の出現などの問題を招くことがあった。

近年，FPGA (Field Programmable Gate Array) を用いて制御プログラムを論理回路化する手法 (ハードワイヤード化) が検討されはじめた。FPGA はプログラム可能な LSI であるため生産現場で任意に論理を変更することができ，1 チップで数十～数百万ゲートを集積しているため大規模な制御論理も小面積で実装することができる。部品点数の削減による低コスト化も期待できるため，高性能を必要とする大規模システムだけでなく，安価な小規模システムにも応用可能である。また，ソフトウェアをハードウェアに変換した場合，一般的に高速化が期待できる。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，制御論理の秘匿性も高い。

1.2 関連研究

Adamski ら [1] は，ロジックコントローラのふるまいをペトリネット表現に変換し，次にペトリネットベースの記述言語に変換する。このペトリネットベースの言語を PLD (Programmable Logic Device) 用の記述言語 (Palasum) に IF-THEN ルールで変換し PLD に実装することを行なっている。

Wegrzyn ら [2] は，論文 [1] で述べられている，ペトリネットのモデルから IF-THEN ルールで PLD 用の言語に変換するという手法を基本とし，VHDL に変換する手法を示している。

竹馬ら [3] は「シーケンス制御方法および装置」を特許出願している。この特許では，シーケンスプログラムを外部から受信し，ハードウェアロジック用の言語に変換して再プログラム可能なハードウェアロジックに格納する方法および装置が示されている。

Wegrzyn ら [4] は，ロジックコントローラのペトリネットモデル，もしくは SFC モデルを，PNSF (Petri-Net Specification Format) と呼ばれるルールベースの明細に変換し，PNSF を Xilinx のネットリストフォーマットに変換することで，FPGA 上に実装を行う方法を報告している。

西田ら [5] は，ペトリネットやラダー図を中間言語に変換し，この中間言語から VHDL に変換する手法を示している。

宮澤ら [6] は，ラダー図を HDL による表現に変換する手法と，変換時に工夫す

ることとそのいくつかの例を示している,

Welch ら [7] は, ラダー図を直接 FPGA 上に実装する為の FPGA アーキテクチャの提案をしている.

1.3 目的

関連研究では PLC の論理を FPGA で実現するアイデアや手法が提案されている. しかし, 現実の PLC の論理の実装や評価については述べられていない.

本研究の目的は, 実用的システムの構成や, ハードワイヤード化を行なうための変換ツールで利用する論理ライブラリの整備を行なうこと. また, 実際の制御回路の一部を FPGA 上に実装し動作を確認すると共に, 回路規模, 動作周波数を評価することである.

2 実現手法

PLC プログラムはラダー図で記述されることが多く，ラダー図はリレー回路が原型であるため論理回路で実現可能である [6]．しかし，現在の PLC は一種の計算機であるため，単純な論理回路だけでは実現することができない．

PLC プログラムの命令列をハードウェアに変換する手法は変換ツールによる変換 [8] があるが，PLC プログラム部分のみを扱っており，周辺回路部分の変換は行なわれていない．

PLC プログラムを論理回路に変換するだけでは，制御システムのハードワイアード化には不十分である．制御論理には多くの周辺回路が関与しているため，周辺回路も FPGA 上に実装しなければ実用的なシステムを構成することができない．そのため，制御論理ライブラリの整備が必要となる．また，PLC でよく用いられる命令も周辺回路のパターンとあわせてライブラリとして整備することで，効率よく実装を行うことができる．

2.1 制御論理ライブラリ

本研究で用いた PLC 制御論理は，八洲熱学 (株) [9] との共同研究で試作された整列巻取機の PLC プログラムである．この整列巻取機では，周辺回路としてパルスジェネレータ (PG) が用いられていたため，まずライブラリとしての PG の実装案を述べる．

次に，PLC でよく用いられる命令について，ライブラリ整備のためのハードウェアへの変換案を述べる．

制御論理ライブラリでは周辺回路や PLC でよく用いられる命令などの汎用部品，汎用機能の提供を行なう．制御論理ライブラリは，C コンパイラがライブラリ関数を利用するのと同様に，変換ツール [8] で利用される．

2.2 パルスジェネレータ (PG)

PG を実装する上で参考にしたのは，試作された整列巻取機で用いられていた $\text{FX}_{2N}\text{-10PG}$ [10] である．

試作機では $\text{FX}_{2N}\text{-10PG}$ の機能のうち同時に使用していないものがある．同時に使用しない機能を分割して作成することで，回路規模の削減をはかることができる．また，一度も使用していない機能は本研究では実装しないことにした．実装しない機能は，アドレス指定運転，近似 S 字制御，エラー詳細の出力などである．分割する主な機能は以下の 2 つとする．

1. 指定された速度 [Hz] のパルスを台形加減速制御で出力する

2. 外部入力パルスを電子ギア比（分子／分母）倍したパルスを出力する

機能1をSTPG（ステッピング用パルスジェネレータ）、機能2をTRPG（トラバース用パルスジェネレータ）とする。

STPG, TRPG の具体的な設計などは付録 A に示す。

2.3 STPG（ステッピング用パルスジェネレータ）

STPG は指定された速度 [Hz] のパルスを台形加減速制御で出力する機能を持つ。台形加減速とは、速度を縦軸、時間を横軸にとったとき、速度の変化が台形となる加減速の方法である。

STPG を図 1 に示す。

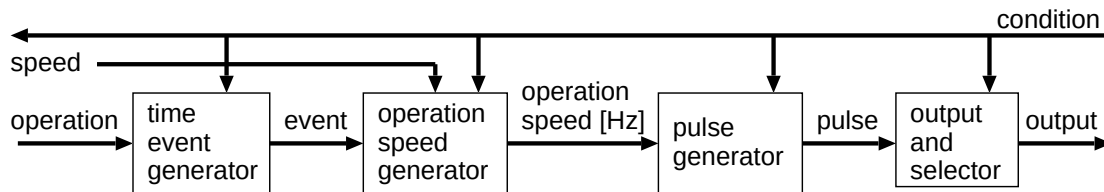


図 1: STPG のブロック図

STPG は指令を受けると、まず指定された周期でイベントを生成する。イベントが発生するごとに、指定された値を加速時には加算、減速時には減算し、動作速度を生成する。speed で指定した速度または最大速度まで加速、もしくはバイアス速度まで減速を行なう。生成された動作速度 [Hz] のパルスを生成し、出力を行なう。

time event generator で指定する周期を以下に示す。

F_ACCEL 加速周波数. 1kHz～1MHz.

F_SLOW 減速周波数. 1kHz～1MHz.

加速時には F_ACCEL で指定した周期でイベントを生成し、減速時には F_SLOW で指定した周期でイベントを生成する。

operation speed generator で指定する値を以下に示す

MAX_S 最大速度. 1Hz～1MHz.

BIAS_S バイアス速度. 0Hz～30kHz.

ACCEL_C 加速定数. 1Hz～10kHz.

SLOW_C 減速定数. 1Hz～10kHz.

加速時にはイベントごとに現在の動作速度に ACCEL_C を加算し，動作速度を加速する．動作速度が speed で指定した速度または最大速度に達したら加算を停止し，動作速度は一定となる．減速時にはイベントごとに現在の動作速度に SLOW_C を減算し，動作速度を減速する．動作速度がバイアス速度に達したら減算を指定しパルスの出力を停止する．

F_ACCEL と ACCEL_C の関係を式 1, F_SLOW と SLOW_C の関係を式 2 に示す．accelerationtime は加速時間，slowdowntime は減速時間を示す．(加速時間，減速時間は FX_{2N}-10PG に指定するパラメータであるが，F_ACCEL, ACCEL_C, F_SLOW, SLOW_C は FX_{2N}-10PG には存在しない．必要な精度で加減速を行なうために，4つのパラメータで指定を行なうことにしている.)

$$ACCEL_C = MAX_S / (F_ACCEL * accelerationtime[sec]) \quad (1)$$

$$SLOW_C = MAX_S / (F_SLOW * slowdowntime[sec]) \quad (2)$$

図 2 に内部パラメータの関係と STPG の動作を示す．

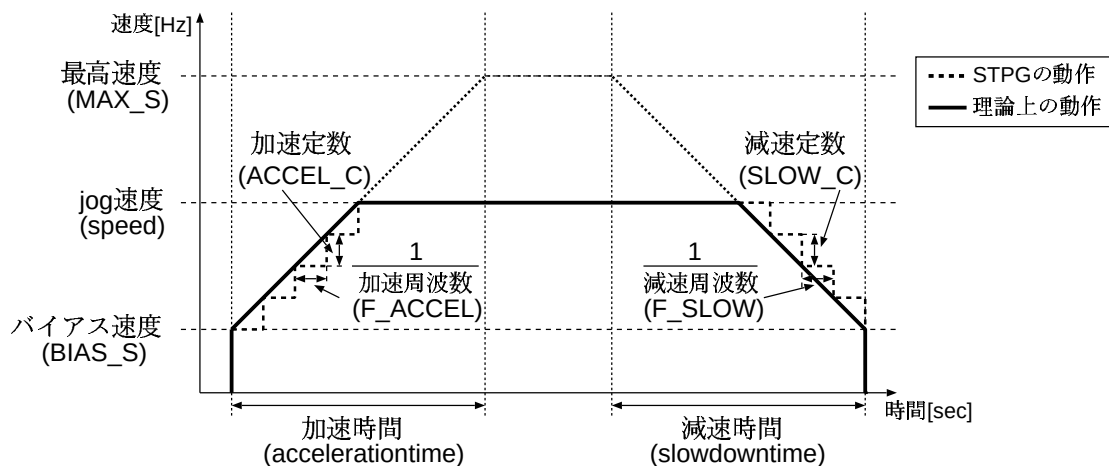


図 2: 内部パラメータの関係と STPG の動作

例) 外部クロック信号周波数 33.333MHz，最大速度 500kHz，バイアス速度 0Hz，加速時間 100msec，減速時間 100msec の場合，次のように定義する．

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000. (10usec 間隔).

F_SLOW 10,000. (100usec 間隔).

ACCEL_C 50. ($= 500,000 / (100,000 * 0.1)$).

SLOW_C 500. ($= 500,000 / (10,000 * 0.1)$).

operation speed pulse generator では、生成された動作速度のパルスを生成する。
output and selector では FP（正転用出力）、RP（逆転用出力）のうち、選択された方に生成された動作速度のパルスの出力を行なう。

2.3.1 評価

STPG 単体での回路規模，動作周波数の評価を行なった．評価環境とターゲットデバイスを表 1，表 2 に示す．

表 1: 評価環境 (PC)

CPU	AthlonXP 3200+
MEM	1GB
OS	Windows2000Pro SP4

表 2: ターゲットデバイス

シリーズ	APEX20KE
デバイス	EP20K200EFC484-2X
LEs	8320
Pin	376

論理は VHDL で記述し，Altera 社の QuartusII4.0 を用いて，Analysis&Synthesis, Fitter, Assembler, Timing Analyzer を実行し見積りを行なった．

内部パラメータは，外部クロック信号周波数 33.333MHz，最大速度 500kHz，バイアス速度 0Hz，加速時間 100msec，減速時間 100msec とした．

各定数を以下に示す．

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000.

F_SLOW 100,000.

ACCEL_C 50.

SLOW_C 50.

結果を表 3 に示す．表 3 中の LEs は回路規模を示す．LE は論理機能を実現するための基本ユニットのことである．fmax は動作周波数を示す．Pins は使用する FPGA の I/O ピンの数を示し，単体時が TRPG のみで構成した場合に使用する数，制御回路使用時が制御回路を組みこんだ場合に使用する数を示す．

表 3: STPG の回路規模と動作周波数

LEs	fmax (MHz)	Pins (単体時)	Pins (制御回路使用時)
185	61.74	32	5

2.4 TRPG (トラバース用パルスジェネレータ)

TRPGは外部入力パルスを電子ギア比(分子/分母)倍したパルスを出力する機能を持つ。(ただし 分子 $\leq$ 分母)

TRPGを図3に示す。

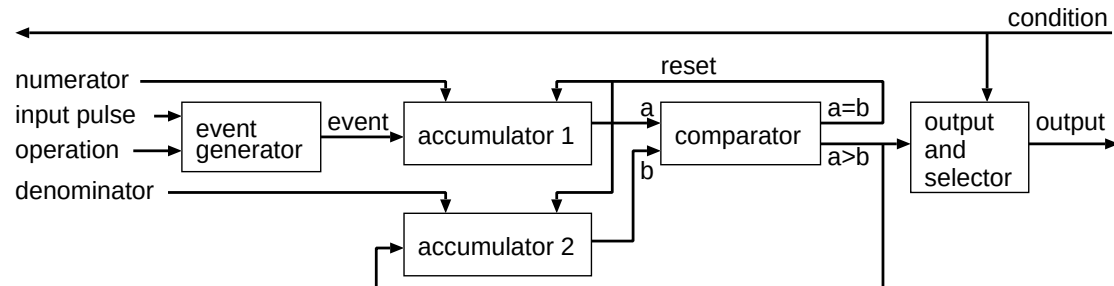


図 3: TRPG のブロック図

STPGは指令を受けると、まず入力パルスの立上り、立下りごとにイベントを生成する。イベントが発生するごとに、分子の値を加算する。分子の累算値が分母の累算値を超えたら出力パルスのレベルを反転させる。また、同時に分母の値を加算する。分子の累算値と分母の累算値が等しくなったら、累算値をリセットする。

電子ギア比の指定用パラメータを以下に示す。

numer 電子ギア比分子. 0~65535 (ただし 分子 $\leq$ 分母)。

denom 電子ギア比分母. 0~65535 (ただし 分子 $\leq$ 分母)。

例) numer=1, denom=2 としたときの動作を図4に示す。

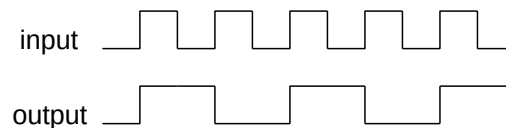


図 4: TRPG の動作

2.4.1 評価

TRPG 単体での回路規模，動作周波数の評価を行なった．評価環境とターゲットデバイスは 2.3.1 節の STPG の場合と同じである．

結果を表 4 に示す．表 4 中の LEs は回路規模を示す．LE は論理機能を実現するための基本ユニットのことである．fmax は動作周波数を示す．Pins は使用する FPGA の I/O ピンの数を示し，単体時が TRPG のみで構成した場合に使用する数，制御回路使用時が制御回路を組みこんだ場合に使用する数を示す．

表 4: TRPG の回路規模と動作周波数

LEs	fmax (MHz)	Pins (単体時)	Pins (制御回路使用時)
403	55.90	97	7

2.5 PLCでよく用いられる命令

PLCでよく用いられる命令として A 社提供の整列巻取機の制御論理の中から用いられているパターンを抽出した。この制御論理は八洲熱学(株)から提供されたものとは異なる論理である。A 社提供の整列巻取機の制御論理をサンプル 1 とする。

サンプル 1 から抽出したパターンを以下に示す。サンプル 1 からパターンを抽出した理由は、初期に入手できたこと、また、八洲熱学(株)で試作された機器も整列巻取機であったからである。サンプル 1 で用いられている PLC は A171SCPU[11] であり、論理は A171CPU に基づいて記述されている。

リレー ON 条件を満たした時にリレーが ON になるパターン [12].

スイッチ 11 例。1 つの条件リレーが ON になると動作リレーが ON になる。

OR-AND 4 例。条件が OR-AND の型で構成されているもの。

AND 2 例。条件が AND の型で構成されているもの。

計算 四則演算などを行ないデータレジスタに結果を格納する [13].

D+ 3 例。32 ビット (2 データレジスタ) 加算。

D- 3 例。32 ビット (2 データレジスタ) 減算。

D* 12 例。32 ビット (2 データレジスタ) 乗算。

D/ 9 例。32 ビット (2 データレジスタ) 除算。

リレーセット リレーの ON,OFF, 保持, 解除などの命令 [13].

PLS 18 例。入力の立ち上がり 1 スキャンサイクルのみ ON。

PLF 3 例。入力の立ち下がり 1 スキャンサイクルのみ ON。

SET 9 例。リレーを ON 状態に保持する。

RST 9 例。ON 状態に保持されたリレーを OFF にする。

タイマーリレー 入力が一定時間つづくかを確認する [12].

T 3 例。指定時間入力があった場合 ON になるリレー。

データ転送 データレジスタ間のデータ転送 [14].

MOV 1 例。16 ビット (1 データレジスタ) 転送。

DMOV 3 例。32 ビット (2 データレジスタ) 転送。

2.6 変換案

サンプル 1 から抽出したパターンを元にライブラリ整備のためのハードウェアへの変換を行なった。各パターンごとの変換案を以下に示す。

2.6.1 リレー ON

リレー ON のパターンについては図 5 に示すような論理素子に変換を行なう。

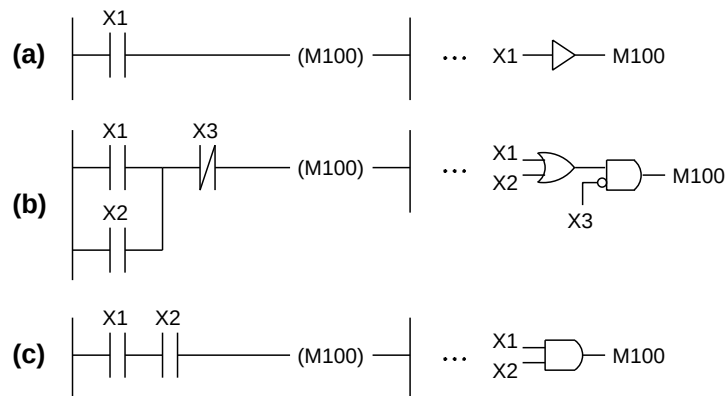


図 5: リレー ON (サンプルパターンと変換案)

図 5 の (a) は $X1$ が ON の間、 $M100$ が ON する論理を示す。

図 5 の (b) は $X1$ または $X2$ が ON のうえで $X3$ が OFF の間、 $M100$ が ON する論理を示す。

図 5 の (c) は $X1$ かつ $X2$ が ON の間、 $M100$ が ON する論理を示す。

2.6.2 計算

図 6 に示すような計算のパターンについては、加減算には `lpm_add_sub[15]` を、乗算には `lpm_mult[15]`、除算には `lpm_divide[15]` を用いて変換を行なう。

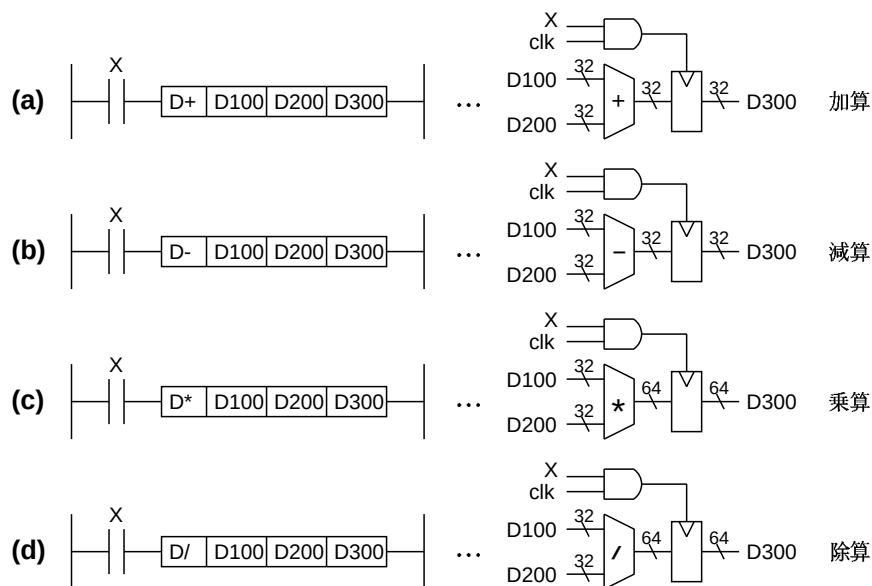


図 6: 計算（サンプルパターンと変換案）

図 6 の clk はシステムクロックを示す。

図 6 の (a) は X が ON の間、 $D100 + D200 = D300$ の動作を行なう論理を示す。

図 6 の (b) は X が ON の間、 $D100 - D200 = D300$ の動作を行なう論理を示す。

図 6 の (c) は X が ON の間、 $D100 * D200 = D300$ の動作を行なう論理を示す。

図 6 の (d) は X が ON の間、 $D100 / D200 = D300$ の動作を行なう論理を示す。

2.6.3 リレーセット

図 7 に示すような PLS, PLF のパターンについて、動作を図 8 に、変換案を図 9 に示す。

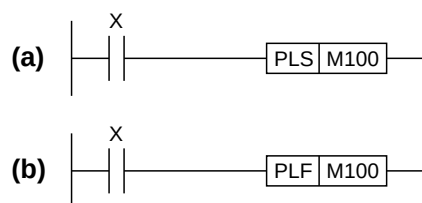


図 7: PLS, PLF（サンプルパターン）

図 8 中の clk はシステムクロックを示す。

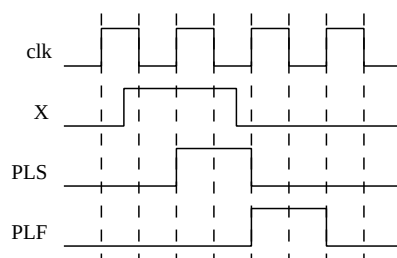


図 8: PLS, PLF の動作

図 7 の (a) は X の立上りで M100 が 1 スキャン ON になる論理を示す. 1 スキャンは PLC の 1 演算周期を示し, 変換案ではシステムクロックの 1 クロック分に相当させている.

図 7 の (b) は X の立下りで M100 が 1 スキャン ON になる論理を示す.

図 9 中の clk はシステムクロック, clr はクリア信号を示す.

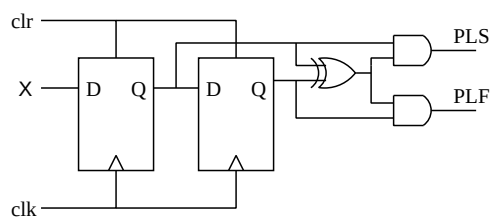


図 9: 1 スキャン ON (PLS, PLF) (変換案)

図 10 に示すような SET, RST のパターンについて, 変換案を図 11, 12 に示す.

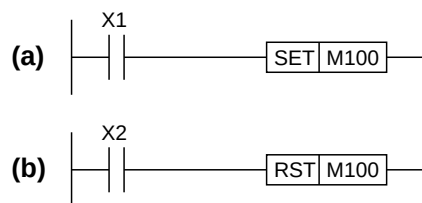


図 10: SET, RST (サンプルパターン)

図 11, 12 中の clk はシステムクロック, clr はクリア信号を示す.

図 10 の (a) は X1 が ON になったら, M100 を ON し X1 が OFF になっても M100 を ON のまま保持する論理を示す.

図 10 の (b) は X2 が ON になったら, M100 を OFF する論理を示す.

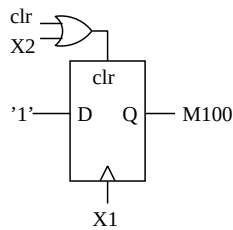


図 11: SET, RST (変換案 1)

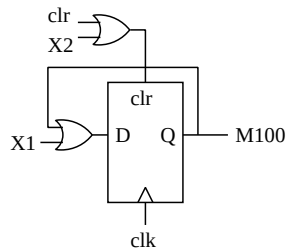


図 12: SET, RST (変換案 2)

2.6.4 タイマーリレー

図 13 に示すようなタイマーリレーのパターンについて、変換案を図 14 に示す。

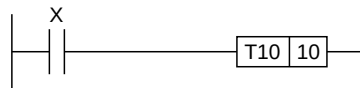


図 13: タイマーリレー (サンプルパターン)

図 13 は X が 1sec 以上 ON になっていれば、T10 を ON する論理を示す。

T10 は 100msec タイマーリレーを示し、T10 の右隣の 10 の指定で 100msec を何回待機するかを示す。図 13 では $100\text{ms} \times 10 = 1\text{sec}$ となる。

図 14 中の clk はシステムクロックを示す。比較する数 33,333,000 は、システムクロック周波数が 33.333MHz であり、待機時間が $100\text{ms} \times 10 = 1\text{sec}$ の場合を示す。待機時間が $100\text{ms} \times 1 = 0.1\text{sec}$ の場合には比較する数を 3,333,000 に変更する。

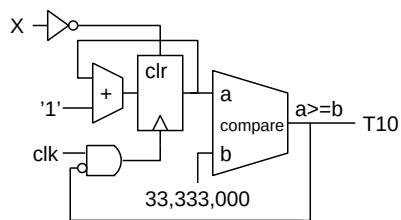


図 14: タイマーリレー (変換案)

2.6.5 データ転送

図 15 に示すようなデータ転送のパターンについて、MOV の変換案を図 16 に、DMOV の変換案を図 17 示す。

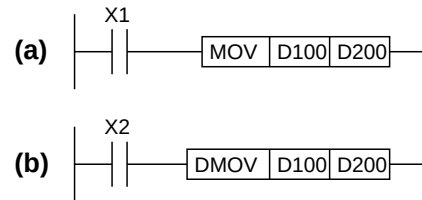


図 15: データ転送（サンプルパターン）

図 15 の (a) は X1 が ON になっていれば、D100 のデータを D200 に転送する論理を示す。

図 15 の (b) は X2 の立上りで、D100 のデータを D200 に転送する論理を示す。

図 16, 図 17 中の clk はシステムクロックを示す。図 17 中の PLS はで述べたものを用いる。

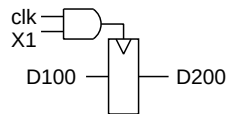


図 16: データ転送（MOV）（変換案）

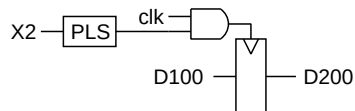


図 17: データ転送（DMOV）（変換案）

3 実装

現実の PLC 制御論理として，平成 15～16 年度に八洲熱学 (株)[9] との共同研究で試作された整列巻取機の制御論理に基づいて FPGA による実装を行なう．

本研究では制御論理のサンプルが 3 つ提供されている．提供された順に，

1. A 社の整列巻取機の制御論理
2. 八洲熱学 (株) の整列巻取機の制御論理
3. 2 番目の制御論理の改良版

となっている．

本研究では 3 番目に提供された整列巻取機の制御論理の一部を FPGA に実装した．3 番目に提供された制御論理をサンプル 3 とする．サンプル 3 で用いられている PLC は FX_{2N}-16MT[16] であり，論理は FX_{2N}-16MT に基づいて記述されている．

2.6 節で述べた PLC でよく用いられる命令の変換案は A171CPU に基づいた論理から作成したものであるが，A171CPU と FX_{2N}-16MT は製造した企業が三菱電機 (株) である．そのため，同じ命令であれば動作は同じであり，また，異なる記述の命令であっても動作が同じであれば，対応は容易である．

具体的な設計などは付録 B に示す．

3.1 試作機の制御論理

試作機の制御論理（サンプル 3）を図 18 に示す．

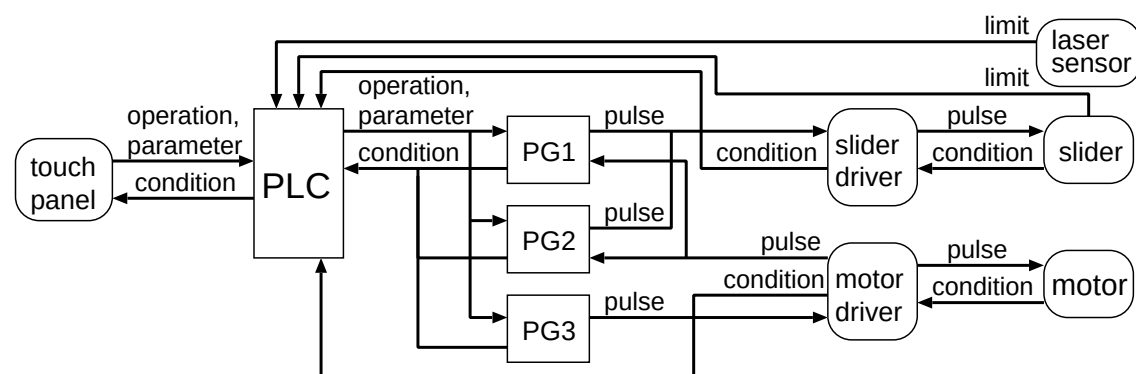


図 18: サンプル 3 のブロック図

サンプル 3 は巻取動作，jog 前後進の機能を持つ．巻取動作は，整列しながら糸の巻取をする場合に行なう．jog 前後進は，スライダの位置を変更する場合に行なう．

- 巻取動作

touch panel から巻取開始指令を出すと PG3 から motor driver に指定速度 [Hz] まで加速しながらパルスを出力する。指定速度になったら加速を終了し指定速度のパルスを出力する。motor driver はパルスの入力を受けて motor を回転させる。

motor が回転すると、回転に対応した 2 種類のパルスが motor driver から出力される。1 つ目 (ASG) はモータの回転に対応し、2 つ目 (BSG) は ASG のパルスから 90 度位相差があるパルスである。1 つ目のパルスの立ち上がりの 2 つ目のパルスのレベルによって回転方向を判別することができる。

PG1 と PG2 は PLC から手動パルス入力運転指令を受けると motor driver から出力されたパルスを入力として手動パルス入力運転を行う。手動パルス入力運転時の運転速度 (出力パルス速度) は、手動パルス入力用電子ギアにより、入力パルス列の周波数に比例した速度となる。

電子ギア比は (分子/分母) で指定し、出力パルスは入力パルスの (分子/分母) 倍になる。

サンプル 3 では分子 (巻取ピッチ) は touch panel で指定し、分母 (巻取ピッチベース) は固定パラメータである。

PG1 の出力は slider driver の前進用入力に接続されており、PG2 の出力は slider driver の後進用入力に接続されている。motor driver から出力されたパルスは ASG, BSG とともに PG1 と PG2 の入力であるが、slider driver への出力はどちらも ASG の入力のみを用いる。

巻取中に slider を前進させる場合は PG1 の出力を接続して PG2 の出力を切断する、後進させる場合は PG2 の出力を接続し、PG1 の出力を切断する。

巻取が始まると motor の回転に対応して slider が前進する。slider のセンサまたは laser sensor が前 limit を検知すると slider の移動方向を後進に切り替える。後進中に slider のセンサまたは laser sensor が後 limit を検知すると slider の移動方向を前進に切り替える。このような slider の移動と motor の回転によって整列巻取を行う。

touch panel から巻取停止指令を出すと PG3 から motor driver にバイアス速度 [Hz] まで減速しながらパルスを出力する。バイアス速度になったら減速を終了しパルス出力を停止する。

運転中に slider driver または motor driver が異常を検知した場合は、パルスの出力を停止し、touch panel に通知する。

- jog 前後進

スライダの位置を調整する場合の指令である。touch panel から jog 前進指令を出している間 PG1 から slider driver にパルスを出力する。slider driver はパ

ルスの入力を受けて slider を前進させる． slider のセンサまたは laser sensor が前 limit を検知すると slider は停止する．

touch panel から jog 後進指令を出している間 PG2 から slider driver にパルスを出力する． slider driver はパルスの入力を受けて slider を後進させる． slider のセンサまたは laser sensor が後 limit を検知すると slider は停止する．

3.2 実装時の変更部分

本研究では第一段階としてモータ，スライダの動作確認を行なうことにした．実装にあたって省略した機能と理由を以下に示す．

- LS-7500（レーザーセンサ）を含む論理
 - － モータ，スライダの動作確認には必要ないから
 - － LS-7500 とのインターフェース部分の実装ができていないから
 - － LS-7500 が研究室になかったのでテストができないから
- PLC 本体の状態を利用する論理（PLC エラーなど）
 - － PLC 本体は利用しないから

実装にあたって変更した機能と理由を以下に示す．

- 入出力をタッチパネルからボードのスイッチ類に変更
 - － モータ，スライダの動作確認では，動作指令はスイッチで可能だから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - － タッチパネルが研究室になかったのでテストができないから
- 巻取ピッチ指定を可変から固定（データ依存）に変更
 - － モータ，スライダの動作確認で動作中には変化させないパラメータであると判断したから
 - － タッチパネルが研究室になかったのでテストができないから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - * パラメータを変更する場合には回路生成を再度行なう
- 巻取速度指定を可変から固定（データ依存）に変更
 - － モータ，スライダの動作確認で動作中には変化させないパラメータであると判断したから

- タッチパネルとのインターフェース部分の実装ができていないから
- タッチパネルが研究室になかったのでテストができないから
 - * パラメータを変更する場合には回路生成を再度行なう
- 現在速度表示をタッチパネルに出力から内部で保持に変更
 - 巻取速度を固定しているから
 - タッチパネルとのインターフェース部分の実装ができていないから
 - タッチパネルが研究室になかったのでテストができないから

3.3 実装した制御論理

実装した論理をサンプル3改とし，図19に示す．

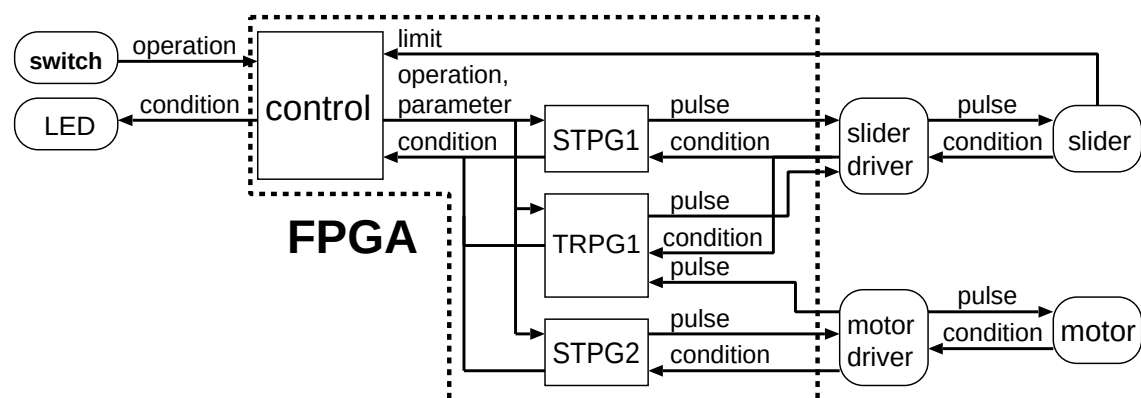


図 19: サンプル3改のブロック図

サンプル3改では，サンプル3で周辺機器であるパルスジェネレータも含めて1チップのFPGA上に実装した．モータ，スライダ，各ドライバはサンプル3と同じものを用いるので接続を変更すれば試作機での動作確認も行なうことができる．

サンプル3改はサンプル3と同様に巻取動作，jog 前後進の機能を持つ．巻取動作は，整列しながら糸の巻取をする場合に行なう．jog 前後進は，スライダの位置を変更する場合に行なう．

● 巻取動作

switchで巻取開始指令を出すと STPG2 から motor driver に指定速度 [Hz] まで加速しながらパルスを出力する．指定速度になったら加速を終了し指定速度のパルスを出力する．motor driver はパルスの入力を受けて motor を回転させる．

motor が回転すると、回転に対応した 2 種類のパルスが motor driver から出力される。1 つ目 (ASG) はモータの回転に対応し、2 つ目 (BSG) は ASG のパルスから 90 度位相差があるパルスである。1 つ目のパルスの立ち上がりの 2 つ目のパルスのレベルによって回転方向を判別することができる。

TRPG1 は PLC から手動パルサ入力運転指令を受けると motor driver から出力されたパルスを入力として手動パルサ入力運転を行う。手動パルサ入力運転時の運転速度 (出力パルス速度) は、手動パルサ入力用電子ギアにより、入力パルス列の周波数に比例した速度となる。

電子ギア比は (分子 / 分母) で指定し、出力パルスは入力パルスの (分子 / 分母) 倍になる。

サンプル 3 改では分子 (巻取ピッチ)、分母 (巻取ピッチベース) は固定パラメータである。

STPG の出力は slider driver の前進用、後進用入力に接続されている。motor driver から出力されたパルスは ASG、BSG とも STPG の入力であるが、slider driver への出力は ASG の入力のみを用いる。

巻取中に slider を前進させる場合は、slider driver の前進用入力に STPG のパルスを出力する。slider を後進させる場合は、slider driver の後進用入力に STPG のパルスを出力する。

巻取が始まると motor の回転に対応して slider が前進する。slider のセンサが前 limit を検知すると slider の移動方向を後進に切り替える。後進中に slider のセンサが後 limit を検知すると slider の移動方向を前進に切り替える。このような slider の移動と motor の回転によって整列巻取を行う。

switch から巻取停止指令を出すと STPG2 から motor driver にバイアス速度 [Hz] まで減速しながらパルスを出力する。バイアス速度になったら減速を終了しパルス出力を停止する。

運転中に slider driver または motor driver が異常を検知した場合は、パルスの出力を停止し、LED で通知する。

- jog 前後進

スライダの位置を調整する場合の指令である。touch panel から jog 前進指令を出している間 STPG1 から slider driver の前進用入力にパルスを出力する。slider driver はパルスの入力を受けて slider を前進させる。slider のセンサまたは laser sensor が前 limit を検知すると slider は停止する。

touch panel から jog 後進指令を出している間 STPG1 から slider driver の前進用入力にパルスを出力する。slider driver はパルスの入力を受けて slider を後進させる。slider のセンサまたは laser sensor が後 limit を検知すると slider は停止する。

4 評価

4.1 評価環境

サンプル3改の回路規模，動作周波数の評価を行なった．評価環境とターゲットデバイスを表5，表6に示す．

表 5: 評価環境 (PC)

CPU	AthlonXP 3200+
MEM	1GB
OS	Windows2000Pro SP4

表 6: ターゲットデバイス

シリーズ	APEX20KE
デバイス	EP20K200EFC484-2X
LEs	8320
Pin	376

論理はVHDLで記述し，Altera社のQuartusII4.0を用いて，Analysis&Synthesis, Fitter, Assembler, Timing Analyzerを実行し見積りを行なう．

4.2 評価条件

サンプル3改を実装するにあたり，設定したパラメータを以下に示す．各パラメータの値はサンプル3に准ずる．

STPG1の内部パラメータを以下に示す．

外部クロック信号周波数 33.333MHz，最大速度 500kHz，バイアス速度 0Hz，加速時間 100msec，減速時間 100msec，jog 速度 1kHz とした．

対応する各定数を以下に示す．

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000.

F_SLOW 100,000.

ACCEL_C 50.

SLOW_C 50.

speed 1000.

STPG2 の内部パラメータを以下に示す.

外部クロック信号周波数 33.333MHz, 最大速度 30kHz, バイアス速度 0Hz, 加速時間 5000msec, 減速時間 5000msec, jog 速度 30kHz とした.

対応する各定数を以下に示す.

CLK_F 33,333,000.

MAX_S 30,000.

BIAS_S 0.

F_ACCEL 1,000.

F_SLOW 1,000.

ACCEL_C 6.

SLOW_C 6.

speed 30,000.

TRPG1 の内部パラメータを以下に示す.

分子 500, 分母 3600 とした.

対応する各定数を以下に示す.

numer 500.

denom 3,600.

4.3 結果

結果を表 7 に示す. 表 7 中の LEs は回路規模を示す. LE は論理機能を実現するための基本ユニットのことである. fmax は動作周波数を示す. Pins は使用する FPGA の I/O ピンの数を示す.

表 7: サンプル 3 改の回路規模と動作周波数

LEs	fmax (MHz)	Pins
430	50.38	31

この論理回路はデバイスの最大LE数8,320, 最大ピン数376, FPGAボードのシステムクロック33.333MHzと比較して十分実用可能な規模と速度であるといえる.

この論理回路をデバイスにダウンロードし, インターフェース回路経由でモータ, スライダに接続し問題無く動作することを確認した.

5 おわりに

実際に PLC と周辺回路による制御論理の一部をを FPGA ボードに置き換え，動作を確認した．これにより，FPGA による小型・高性能・高秘匿性制御論理の実現への第一ステップが実現されたといえる．

今後の課題はまず，タッチパネル，レーザセンサを含めた試作機と同等の論理の回路を実装し動作を確認することである．次に，制御論理ライブラリの整備が十分ではないため，さらに整備を進めていく必要がある．具体的には，パルスジェネレータの他にシーケンス制御でよく用いられる周辺回路や，三角関数命令などが挙げられる．また，本研究で扱った試作機の制御論理は小規模なものであるため，より大規模な応用に関しても FPGA 技術が適用可能であるか検討を進めてゆきたい．

謝辞

最後まで根気強くご指導してくださった市川先生にお礼申し上げます．また，苦楽を共にした池田君，八反田君をはじめ研究室の皆さんに感謝いたします．最後に，高専，大学，大学院に通わせてくださった両親に感謝いたします．

A パルスジェネレータ

八洲熱学(株)[9]との共同研究で試作された整列巻取機の制御論理の一部（サンプル3改）の実装を行なうにあたり，周辺回路のパルスジェネレータを実装することにした．参考にしたのは試作機の制御論理（サンプル3）で用いられていたFX_{2N}-10PG[10]である．

サンプル3ではFX_{2N}-10PGの機能のうち同時に使用していないものがある．同時に使用しない機能を分割して作成することで，回路規模の削減をはかることができる．また，一度も使用していない機能は本研究では実装しないことにした．実装しない機能は，アドレス指定運転，近似S字制御，エラー詳細の出力などである．分割する主な機能は以下の2つとする．

1. 指定された速度 [Hz] のパルスを台形加減速制御で出力する
2. 外部入力パルスを電子ギア比（分子／分母）倍したパルスを出力する

機能1をSTPG（ステッピング用パルスジェネレータ），機能2をTRPG（トラバース用パルスジェネレータ）とする．

A.1 STPG（ステッピング用パルスジェネレータ）

STPGは指定された速度 [Hz] のパルスを台形加減速制御で出力する機能を持つ．台形加減速とは，速度を縦軸，時間を横軸にとったとき，速度の変化が台形となる加減速の方法である．

STPGを図20に示す．

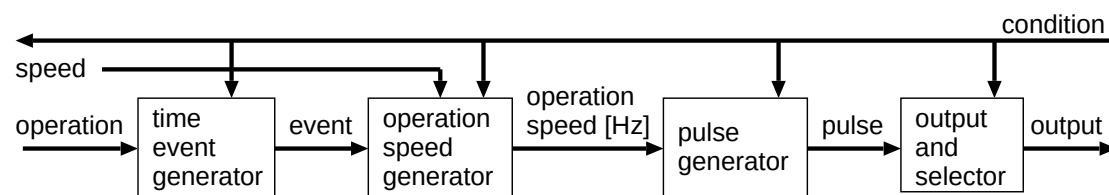


図 20: STPG のブロック図

STPGは指令を受けると，まず指定された周期でイベントを生成する．イベントが発生するごとに，指定された値を加速時には加算，減速時には減算し，動作速度を生成する．speedで指定した速度または最大速度まで加速，もしくはバイアス速度まで減速を行なう．生成された動作速度 [Hz] のパルスを生成し，出力を行なう．

time event generatorで指定する周期を以下に示す．

F_ACCEL 加速周波数. 1kHz～1MHz.

F_SLOW 減速周波数. 1kHz～1MHz.

加速時には F_ACCEL で指定した周期でイベントを生成し，減速時には F_SLOW で指定した周期でイベントを生成する.

operation speed generator で指定する値を以下に示す

MAX_S 最大速度. 1Hz～1MHz.

BIAS_S バイアス速度. 0Hz～30kHz.

ACCEL_C 加速定数. 1Hz～10kHz.

SLOW_C 減速定数. 1Hz～10kHz.

加速時にはイベントごとに現在の動作速度に ACCEL_C を加算し，動作速度を加速する．動作速度が speed で指定した速度または最大速度に達したら加算を停止し，動作速度は一定となる．減速時にはイベントごとに現在の動作速度に SLOW_C を減算し，動作速度を減速する．動作速度がバイアス速度に達したら減算を指定しパルスの出力を停止する．

F_ACCEL と ACCEL_C の関係を式 5, F_SLOW と SLOW_C の関係を式 6 に示す．accelerationtime は加速時間，slowdowntime は減速時間を示す．(加速時間，減速時間は FX_{2N}-10PG に指定するパラメータであるが，F_ACCEL, ACCEL_C, F_SLOW, SLOW_C は FX_{2N}-10PG には存在しない．必要な精度で加減速を行なうために，4つのパラメータで指定を行なうことにしている.)

$$ACCEL_C = MAX_S / (F_ACCEL * accelerationtime[sec]) \quad (3)$$

$$SLOW_C = MAX_S / (F_SLOW * slowdowntime[sec]) \quad (4)$$

図 24 に内部パラメータの関係と STPG の動作を示す.

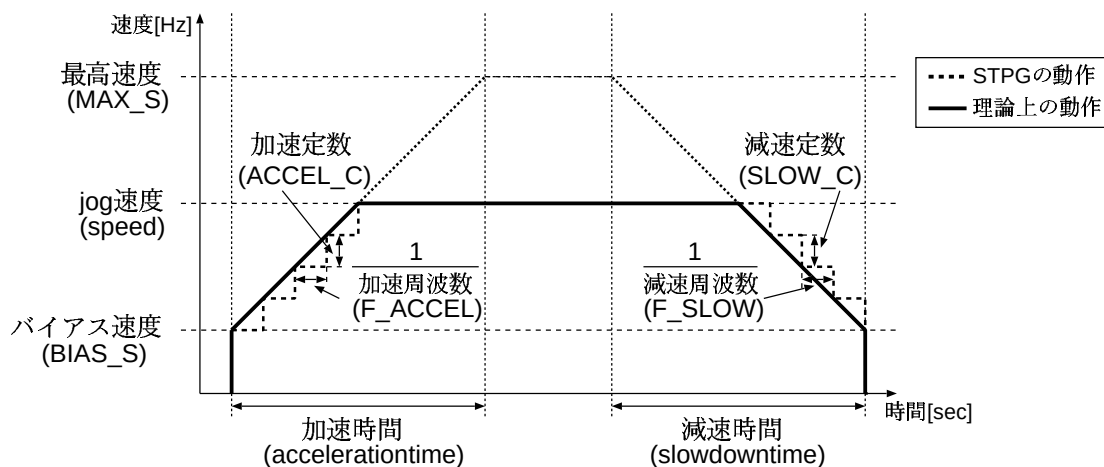


図 21: 内部パラメータの関係と STPG の動作

例) 外部クロック信号周波数 33.333MHz, 最大速度 500kHz, バイアス速度 0Hz, 加速時間 100msec, 減速時間 100msec の場合, 次のように定義する.

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000. (10μsec 間隔).

F_SLOW 10,000. (100μsec 間隔).

ACCEL_C 50. ($= 500,000 / (100,000 * 0.1)$).

SLOW_C 500. ($= 500,000 / (10,000 * 0.1)$).

operation speed pulse generator では, 生成された動作速度のパルスを生成する.
output and selector では FP (正転用出力), RP (逆転用出力) のうち, 選択された方に生成された動作速度のパルスの出力を行なう.

A.1.1 入出力

STPG の入出力を図 22 に示す.

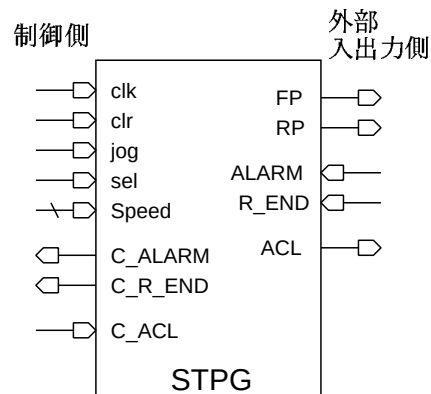


図 22: STPG の入出力

STD (ステッピング用ドライバ) [17] の入出力を図 23 に示す.

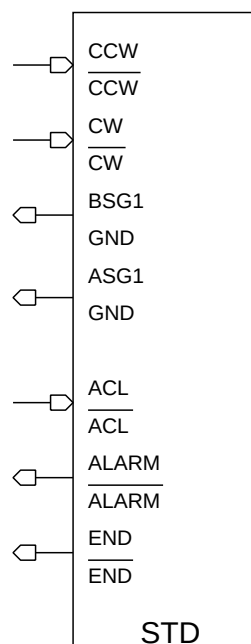


図 23: STD の入出力

A.1.2 信号線の定義

以下に STPG で用いる信号の説明を示す.

clk システムクロック入力.

clr システムクリア信号入力.

jog パルス出力信号. '0'→'1'で speed で指定した速度まで加速を行ないながらパルスを出力する. '1'→'0'でバイアス速度まで減速しながらパルスを出力し, バイアス速度まで減速したら停止する. 加減速は台形制御で行なう. バイアス速度, 加減速時間は内部パラメータで定義する.

sel FP と RP のどちらにパルスを出力するか. '0'=FP, '1'=RP に出力.

speed 出力パルス速度. 1Hz~1MHz (20ビット).

C_ALARM ドライバユニットのエラー出力.

C_R_END モーター停止状態信号出力. '0'=動作中, '1'=停止状態.

C_ACL エラークリア信号入力.

FP 正転信号出力.

RP 逆転信号出力.

ALARM ドライバユニットのエラー入力.

R_END モーター停止状態信号入力. '0'=動作中, '1'=停止状態.

ACL エラークリア信号出力.

以下に STD (トラバース用ドライバ) で用いる信号の説明を示す.

CW 正転パルス入力.

$\overline{\text{CW}}$ 正転パルス入力コモン.

CCW 逆転パルス入力.

$\overline{\text{CCW}}$ 逆転パルス入力コモン.

ALARM ドライバユニットのエラー出力. '0'=エラー, '1'=正常.

$\overline{\text{ALARM}}$ ドライバユニットのエラー出力コモン.

END モーター停止状態信号出力. '0'=動作中, '1'=停止状態.

$\overline{\text{END}}$ モーター停止状態信号出力コモン.

ACL エラークリア信号入力.

$\overline{\text{ACL}}$ エラークリア信号入力コモン.

A.1.3 内部パラメータ

サンプル 3 で FX_{2N} -10PG に対して指定するパラメータのうち、動作中に変化しないものを内部パラメータとして回路を生成する。動的にパラメータの指定を行なう場合には、対応する部分を入力信号とすることで対応できる。内部パラメータ信号を以下に示す。

CLK_F 外部クロック信号の周波数。(33.333MHz の場合は 25 ビットの信号線が必要)。

MAX_S 最大速度。1Hz～1MHz (20 ビット)。

BIAS_S バイアス速度。0Hz～30kHz (15 ビット)

F_ACCEL 加速周波数。1kHz～1MHz (20 ビット)。指定した周期でイベントが発生し、イベントごとに ACCEL_C[Hz] 加速する。

F_SLOW 減速周波数。1kHz～1MHz (20 ビット)。指定した周期でイベントが発生し、イベントごとに SLOW_C[Hz] 減速する。

ACCEL_C 加速定数。1Hz～10kHz (14 ビット)。F_ACCEL 周期ごとに指定した速度分加速する。

SLOW_C 減速定数。1Hz～10kHz (14 ビット)。F_SLOW 周期ごとに指定した速度分加速する。

F_ACCEL と ACCEL_C の関係を式 5、F_SLOW と SLOW_C の関係を式 6 に、内部パラメータの定義を図 24 に示す。accelerationtime は加速時間、slowdowntime は減速時間を示す。(加速時間、減速時間は FX_{2N} -10PG に指定するパラメータであるが、F_ACCEL、ACCEL_C、F_SLOW、SLOW_C は FX_{2N} -10PG には存在しない。必要な精度で加減速を行なうために、4 つのパラメータで指定を行なうことにしている。)

$$ACCEL_C = MAX_S / (F_ACCEL * accelerationtime[sec]) \quad (5)$$

$$SLOW_C = MAX_S / (F_SLOW * slowdowntime[sec]) \quad (6)$$

図 24 に内部パラメータの関係と STPG の動作を示す。

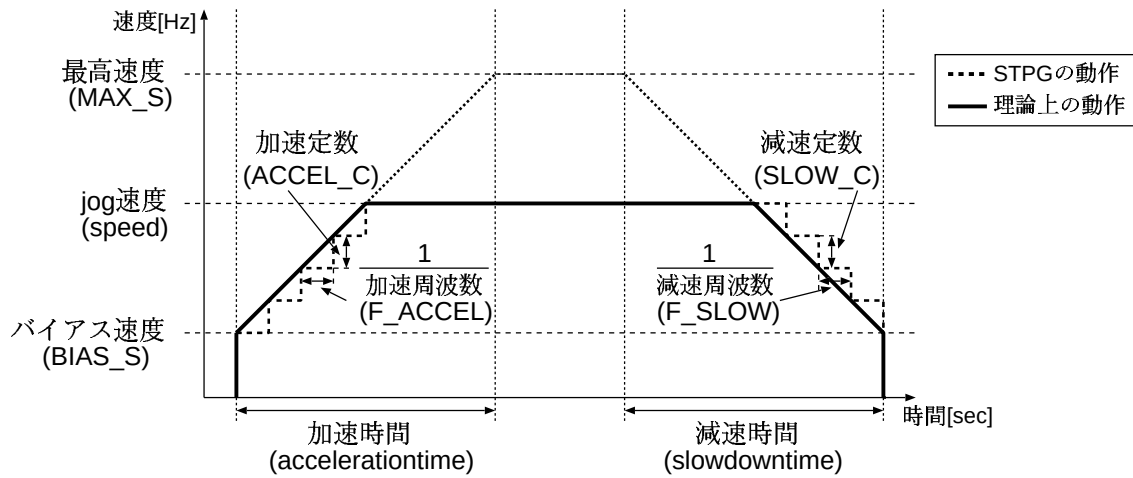


図 24: 内部パラメータの関係と STPG の動作

例) 外部クロック信号周波数 33.333MHz, 最大速度 500kHz, バイアス速度 0Hz, 加速時間 100msec, 減速時間 100msec の場合, 次のように定義する.

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000. (10μsec 間隔).

F_SLOW 10,000. (100μsec 間隔).

ACCEL_C 50. ($= 500,000 / (100,000 * 0.1)$).

SLOW_C 500. ($= 500,000 / (10,000 * 0.1)$).

A.1.4 設計

STPG のブロック図（詳細）を図 25 に、状態図を図 26 に示す。

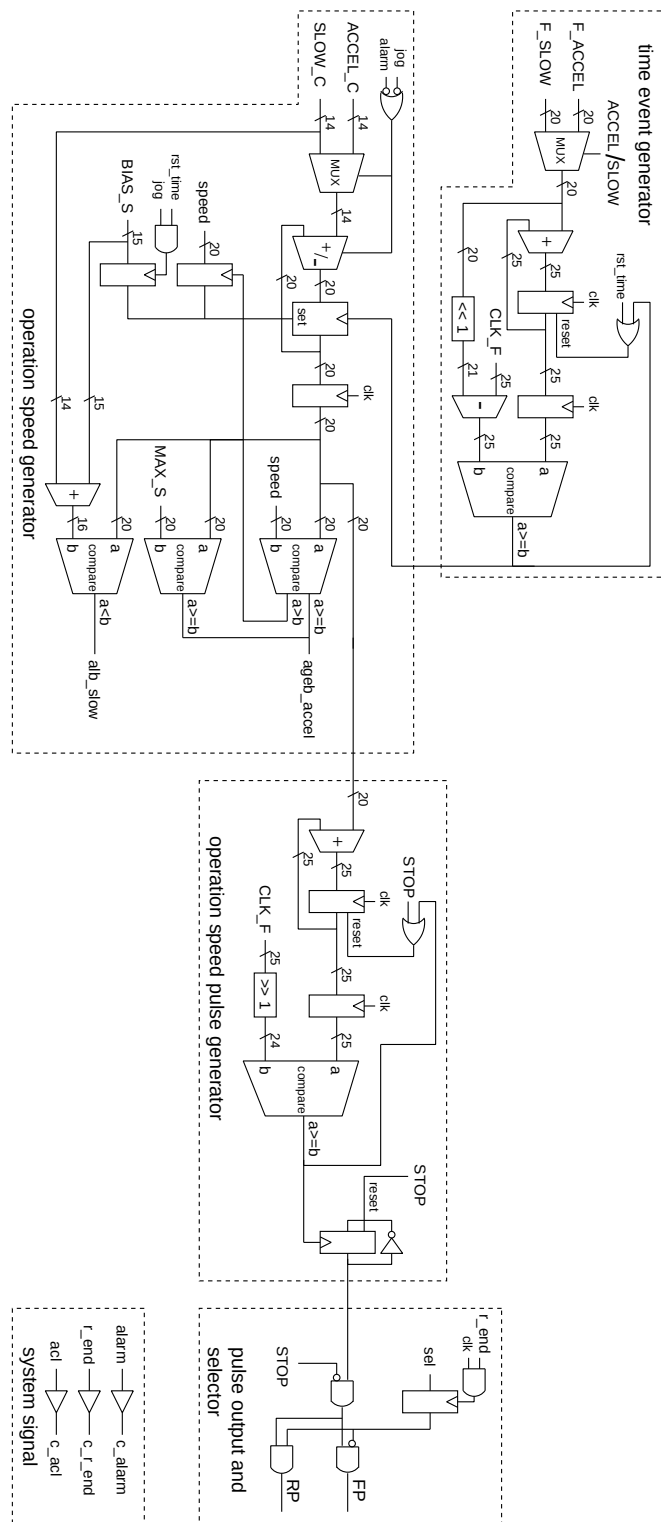


図 25: STPG のブロック図（詳細）

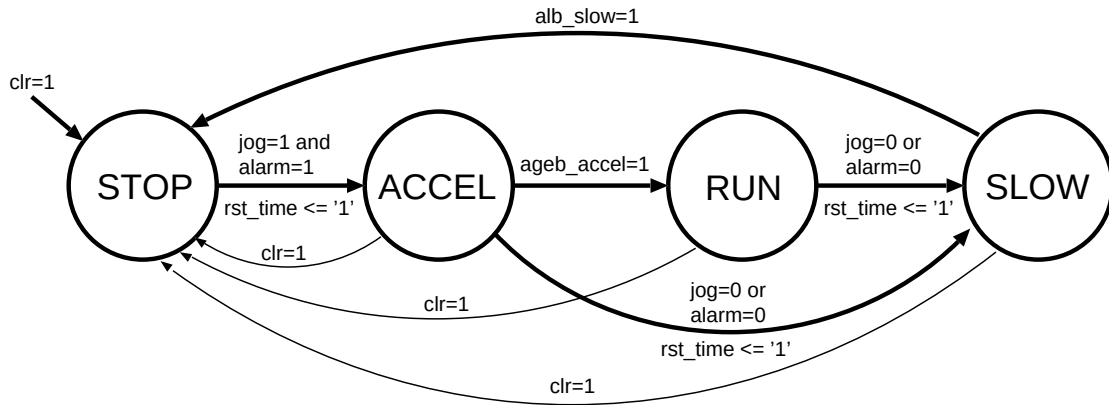


図 26: STPG の状態図

入力のうち、ageb_accel, alb_slow, 出力の rst_time は図 25 に示す内部定義信号である。

各状態は、STOP が停止、ACCEL が加速中、RUN が一定速度運転、SLOW が減速中であることを表す。

図 25 の各ブロックの説明を以下に示す。

- time event generator
 - － 加速時には F_ACCEL[Hz]、減速時には F_SLOW[Hz] でイベントを発生させる
 - * 入力：システムクロック, F_ACCEL, F_SLOW, CLK_F, rst_time.
 - * 出力：F_ACCEL[Hz] または F_SLOW[Hz] のパルス
- operation speed generator
 - － time event generator で生成したパルスイベントごとに加速時には ACCEL_C を加算、減速時には SLOW_C を減算し動作速度を生成する。指定速度、最大速度を超えた場合は ageb_accel を出力し、バイアス速度を下まわった場合には alb_slow を出力し、状態を変更する。
 - * 入力：time event generator の出力, システムクロック, jog, alarm, ACCEL_C, SLOW_C, speed, MAX_S, BIAS_S, rst_time.
 - * 出力：動作速度 [Hz], ageb_accel, alb_slow.
- operation speed pulse generator
 - － operation speed generator で生成した動作速度 [Hz] のパルスを出力する。
 - * 入力：operation speed generator の出力, システムクロック, CLK_F.
 - * 出力：生成したパルス.

- pulse output and selector
 - operation speed pulse generator で生成されたパルスを, sel 信号で選択して出力する.
 - * 入力: operation speed pulse generator の出力, sel, システムクロック, r_end.
 - * 出力: 出力パルス.
- system signal
 - 制御部分からドライバへの指令を送る. またはドライバの状態を制御部分に送る.
 - * 入力: alarm, r_end, acl.
 - * 出力: c_alarm, c_r_end, c_acl.

A.1.5 評価

図 31 の STPG 単体での回路規模, 動作周波数の評価を行なった. 評価環境とターゲットデバイスを表 8, 表 9 に示す.

表 8: 評価環境

CPU	AthlonXP 3200+
MEM	1GB
OS	Windows2000Pro SP4
ツール	QuartusII 4.0

表 9: ターゲットデバイス (Nios 開発ボード)

シリーズ	APEX20KE
デバイス	EP20K200EFC484-2X
LEs	8320
Pin	376

論理は VHDL で記述し, Altera 社の QuartusII4.0 を用いて, Analysis&Synthesis, Fitter, Assembler, Timing Analyzer を実行し見積りを行なった.

内部パラメータは, 外部クロック信号周波数 33.333MHz, 最大速度 500kHz, バイアス速度 0Hz, 加速時間 100msec, 減速時間 100msec として測定を行なった.

各定数を以下に示す.

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000.

F_SLOW 100,000.

ACCEL_C 50.

SLOW_C 50.

結果を表 10 に示す．表 10 中の LEs は回路規模を示す．LE は論理機能を実現するための基本ユニットのことである．fmax は動作周波数を示す．Pins は使用する FPGA の I/O ピンの数を示し，単体時が TRPG のみで構成した場合に使用する数，制御回路使用時が制御回路を組みこんだ場合に使用する数を示す．

表 10: STPG の回路規模と動作周波数

LEs	fmax (MHz)	Pins (単体時)	Pins (制御回路使用時)
185	61.74	32	5

A.2 TRPG (トラバース用パルスジェネレータ)

TRPG は外部入力パルスを電子ギア比 (分子/分母) 倍したパルスを出力する機能を持つ。(ただし 分子 $\leq$ 分母)

TRPG を図 27 に示す。

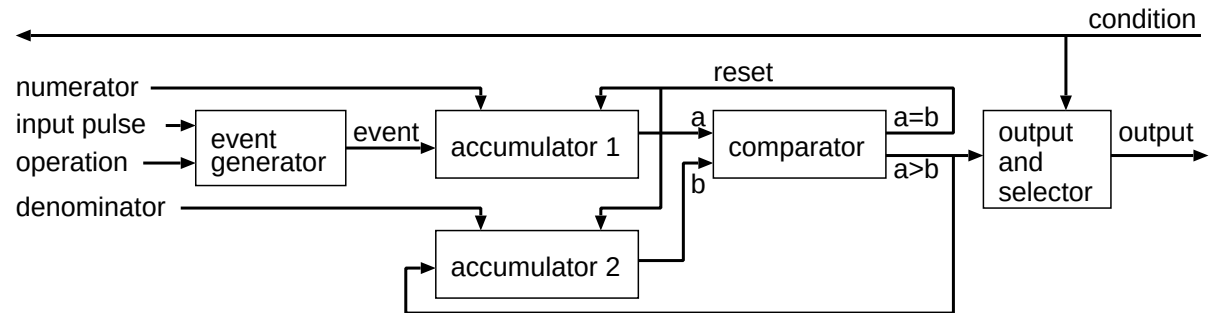


図 27: TRPG のブロック図

STPG は指令を受けると、まず入力パルスの立上り、立下りごとにイベントを生成する。イベントが発生するごとに、分子の値を加算する。分子の累算値が分母の累算値を超えたら出力パルスのレベルを反転させる。また、同時に分母の値を加算する。分子の累算値と分母の累算値が等しくなったら、累算値をリセットする。

電子ギア比の指定用パラメータを以下に示す。

numer 電子ギア比分子. 0~65535 (ただし 分子 $\leq$ 分母).

denom 電子ギア比分母. 0~65535 (ただし 分子 $\leq$ 分母).

例) numer=1, denom=2 としたときの動作を図 28 に示す。

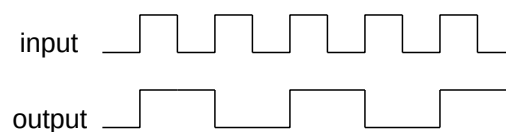


図 28: TRPG の動作

A.2.1 入出力

TRPG の入出力を図 22 に示す.

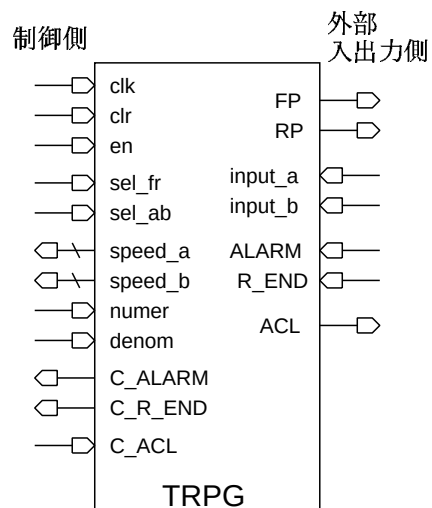


図 29: TRPG の入出力

TRD (スライダ用ドライバ) [18] の入出力を図 30 に示す.

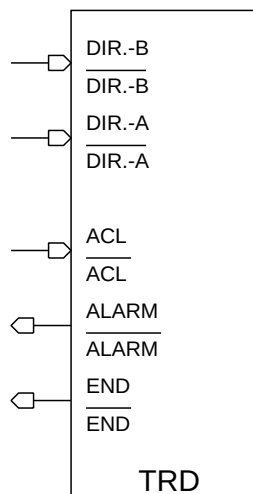


図 30: TRD の入出力

A.2.2 信号線の定義

以下に TRPG で用いる信号の説明を示す.

clk システムクロック入力.

clr システムクリア信号入力.

en 出力指令信号. '1' の場合に input_a または input_b から入力されたパルス信号が指定された比のパルス信号となり, 指定された方法で出力される.

speed_a input_a から入力されたパルス信号のシステムクロックでの分周数を出
力. システムクロックに 33.333MHz を用いた場合, input_a が 1Hz であった
ときの speed_a の出力は 33,333,000 前後であり, この数が最大値であるため,
25ビットの信号線を用いる. ビット幅はシステムクロックの周波数によって
変更する.

speed_b input_b から入力されたパルス信号のシステムクロックでの分周数を出
力. システムクロックに 33.333MHz を用いた場合, input_b が 1Hz であった
ときの speed_b の出力は 33,333,000 前後であり, この数が最大値であるため,
25ビットの信号線を用いる. ビット幅はシステムクロックの周波数によって
変更する.

numer 電子ギア比分子を入力 (0~65535 (16 ビット)), ただし 分子 ≤ 分母).

denom 電子ギア比分母を入力 (0~65535 (16 ビット)), ただし 分子 ≤ 分母).

sel_fr FP と RP のどちらにパルスを出力するか. '0'=FP, '1'=RP を出力. (表 11
参照)

sel_ab input_a と input_b のどちらを出力するか. '0'=input_a → FP, input_b →
RP. '1'=input_b → FP, input_a → RP. (表 11 参照)

C_ALARM ドライバユニットのエラー出力.

C_R_END モーター停止状態信号出力. '0'=動作中, '1'=停止状態.

C_ACL エラークリア信号入力.

FP 正転信号出力.

RP 逆転信号出力.

input_a 入力パルス A. 周波数は 1Hz~30kHz とする.

input_b 入力パルス B. 周波数は 1Hz~30kHz とする.

ALARM ドライバユニットのエラー入力.

R_END モーター停止状態信号入力. '0' = 動作中, '1' = 停止状態.

ACL エラークリア信号出力.

表 11: sel_fr と sel_ab の関係

sel_fr	sel_ab	out
0	0	input_a $\rightarrow$ FP
0	1	input_b $\rightarrow$ FP
1	0	input_a $\rightarrow$ RP
1	1	input_b $\rightarrow$ RP

以下に TRD（トラバース用ドライバ）で用いる信号の説明を示す.

DIR.-A 前進パルス入力.

$\overline{\text{DIR.-A}}$ 前進パルス入力コモン.

DIR.-B 後進パルス入力.

$\overline{\text{DIR.-B}}$ 後進パルス入力コモン.

ALARM ドライバユニットのエラー出力. '0' = エラー, '1' = 正常.

$\overline{\text{ALARM}}$ ドライバユニットのエラー出力コモン.

END モーター停止状態信号出力. '0' = 動作中, '1' = 停止状態.

$\overline{\text{END}}$ モーター停止状態信号出力コモン.

ACL エラークリア信号入力.

$\overline{\text{ACL}}$ エラークリア信号入力コモン.

A.2.3 設計

TRPGのブロック図（詳細）を図31に示す。

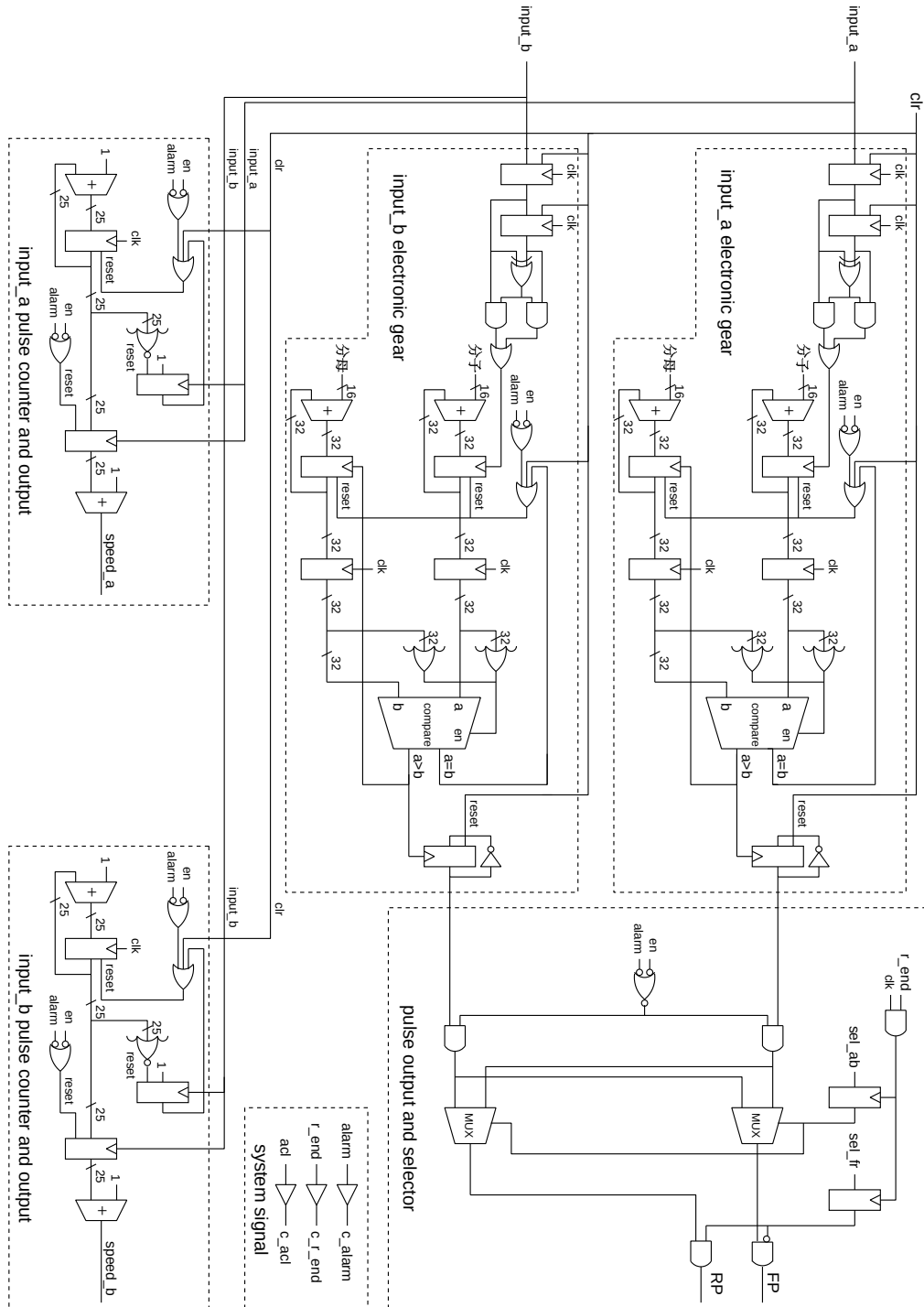


図 31: TRPG のブロック図（詳細）

図 31 の各ブロックの説明を以下に示す。

- input_a electronic gear
 - input_a のイベントごとに numer を加算する。加算した値が denom の値を超えたらパルスを出力し、denom の値を加算する。numer=denom なら加算してきた値をリセットする。
 - * 入力：システムクロック, clr, input_a, en, alarm, numer, denom.
 - * 出力：numer > denom 時のパルス.
- input_b electronic gear
 - input_b のイベントごとに numer を加算する。累算値が denom の値を超えたらパルスのレベルを反転し、denom の値を加算する。numer の累算値=denom の累算値なら累算値をリセットする。
 - * 入力：システムクロック, clr, input_b, en, alarm, numer, denom.
 - * 出力：numer > denom 時のパルス.
- input_a pulse conuter and output
 - input_a がシステムクロックで何分周できるかをカウントする。input_a の立上りから立上りまでの間システムクロックをカウントする。
 - * 入力：システムクロック, clr, input_a, en, alarm.
 - * 出力：speed_a のカウント数.
- input_b pulse conuter and output
 - input_b がシステムクロックで何分周できるかをカウントする。input_b の立上りから立上りまでの間システムクロックをカウントする。
 - * 入力：システムクロック, clr, input_b, en, alarm.
 - * 出力：speed_b のカウント数.
- pulse output and selector
 - operation speed pulse generate で生成されたパルスを duty50%に変換し、sel_ab, sel_fr で指定された出力を行なう。
 - * 入力：input_a electronic gear の出力, input_b electronic gear の出力, sel_ab, sel_fr, システムクロック, r_end, en, alarm.
 - * 出力：生成したパルス.

A.2.4 評価

図 31 の TRPG 単体での回路規模，動作周波数の評価を行なった．評価環境とターゲットデバイスは STPG と同じである．

結果を表 12 に示す．表 12 中の LEs は回路規模を示す．LE は論理機能を実現するための基本ユニットのことである．fmax は動作周波数を示す．Pins は使用する FPGA の I/O ピンの数を示し，単体時が TRPG のみで構成した場合に使用する数，制御回路使用時が制御回路を組みこんだ場合に使用する数を示す．

表 12 に示す通常版は TRPG の機能を全て含むものであり，機能限定版は，入力パルスの分周数を出力する部分，すなわち通常版の input_a pulse counter and output 部分と input_b pulse counter and output 部分を削除したものである．Pins は使用する I/O ピンの数を示し，単体時が TRPG のみで構成した場合に使用する数，制御回路使用時が制御回路を組みこんだ場合に使用する数を示す．

表 12: TRPG の回路規模と動作周波数（上段：通常版，下段：機能限定版）

LEs	fmax (MHz)	Pins (単体時)	Pins (制御回路使用時)
403	55.90	97	7
283	54.21	47	7

B PLC 制御論理の実装

B.1 現実の PLC 制御論理

現実の PLC 制御論理として，平成 15～16 年度に八洲熱学 (株) [9] との共同研究で試作された整列巻取機の PLC プログラムに基づいて FPGA による実装を行なう．本研究では制御論理のサンプルが 3 つ提供されている．提供された順に，

1. A 社の整列巻取機の制御論理
2. 八洲熱学 (株) の整列巻取機の制御論理
3. 2 番目の制御論理の改良版

となっている．

本研究では 3 番目に提供された整列巻取機の制御論理を FPGA に実装した．3 番目に提供された制御論理をサンプル 3 とする．

B.2 試作機の制御論理

試作機の制御論理（サンプル 3）を図 32 に示す．

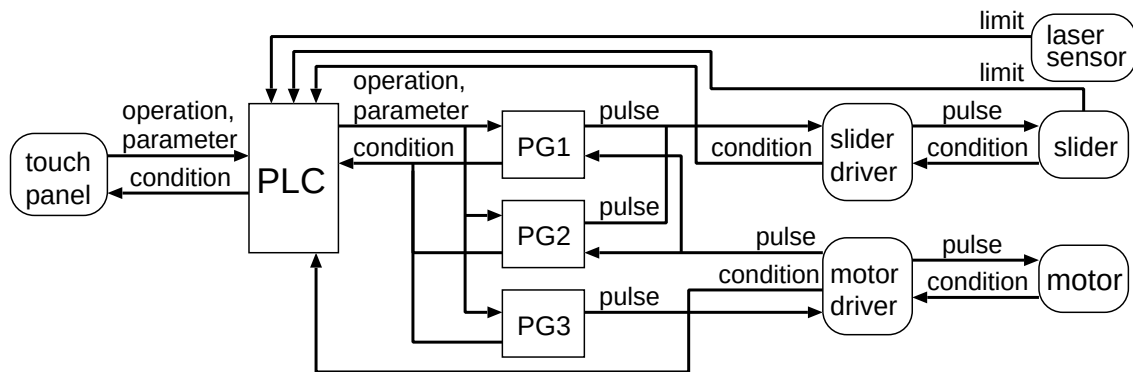


図 32: サンプル 3 のブロック図

サンプル 3 は巻取動作，jog 前後進の機能を持つ．巻取動作は，整列しながら糸の巻取をする場合に行なう．jog 前後進は，スライダの位置を変更する場合に行なう．

B.3 実装時の変更部分

本研究では第一段階としてモータ，スライダの動作確認を行なうことにした．実装にあたって省略した機能と理由を以下に示す．

- LS-7500（レーザーセンサ）を含む論理
 - － モータ，スライダの動作確認には必要ないから
 - － LS-7500 とのインターフェース部分の実装ができていないから
 - － LS-7500 が研究室になかったのでテストができないから
- PLC 本体の状態を利用する論理（PLC エラーなど）
 - － PLC 本体は利用しないから

実装にあたって変更した機能と理由を以下に示す.

- 入出力をタッチパネルからボードのスイッチ類に変更
 - － モータ，スライダの動作確認では，動作指令はスイッチで可能だから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - － タッチパネルが研究室になかったのでテストができないから
- 巻取ピッチ指定を可変から固定（データ依存）に変更
 - － モータ，スライダの動作確認で動作中には変化させないパラメータであると判断したから
 - － タッチパネルが研究室になかったのでテストができないから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - * パラメータを変更する場合には回路生成を再度行なう
- 巻取速度指定を可変から固定（データ依存）に変更
 - － モータ，スライダの動作確認で動作中には変化させないパラメータであると判断したから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - － タッチパネルが研究室になかったのでテストができないから
 - * パラメータを変更する場合には回路生成を再度行なう
- 現在速度表示をタッチパネルに出力から内部で保持に変更
 - － 巻取速度を固定しているから
 - － タッチパネルとのインターフェース部分の実装ができていないから
 - － タッチパネルが研究室になかったのでテストができないから

B.4 実装した制御論理

実装した論理をサンプル 3 改とし，図 33 に示す．

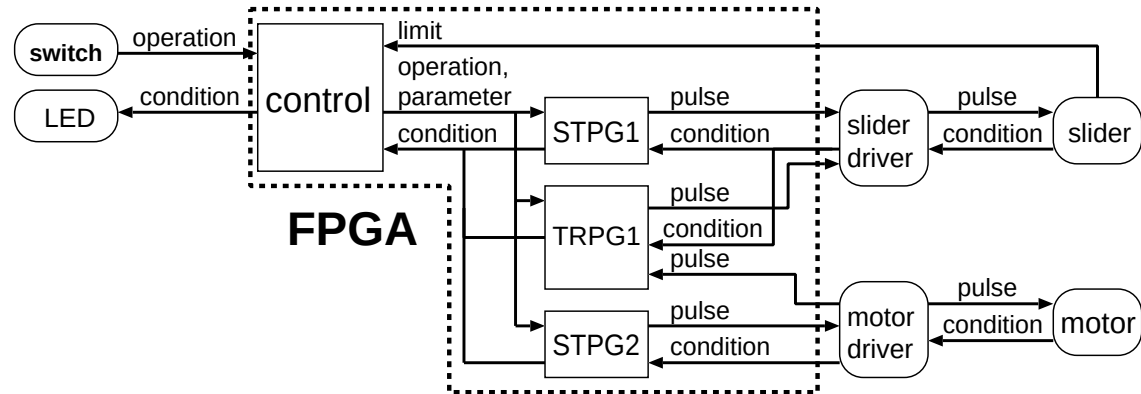


図 33: サンプル 3 改のブロック図

サンプル 3 改では，サンプル 3 で周辺機器であるパルスジェネレータも含めて 1 チップの FPGA 上に実装した．モータ，スライダ，各ドライバはサンプル 3 と同じものを用いるので接続を変更すれば試作機での動作確認も行なうことができる．

サンプル 3 改はサンプル 3 と同様に巻取動作，jog 前後進の機能を持つ．巻取動作は，整列しながら糸の巻取をする場合に行なう．jog 前後進は，スライダの位置を変更する場合に行なう．

B.4.1 設計

サンプル 3 改の制御部分のブロック図を図 34 に示す．

図 34 中のブロックは，それぞれ STPG（ステッピング用パルスジェネレータ），TRPG（トラバース用パルスジェネレータ），STD（ステッピングモータ用ドライバ）[17]，TRD（スライダ用ドライバ）[18]，STM（ステッピングモータ）[17]，TRM（スライダ用モータ）[18]を示す．

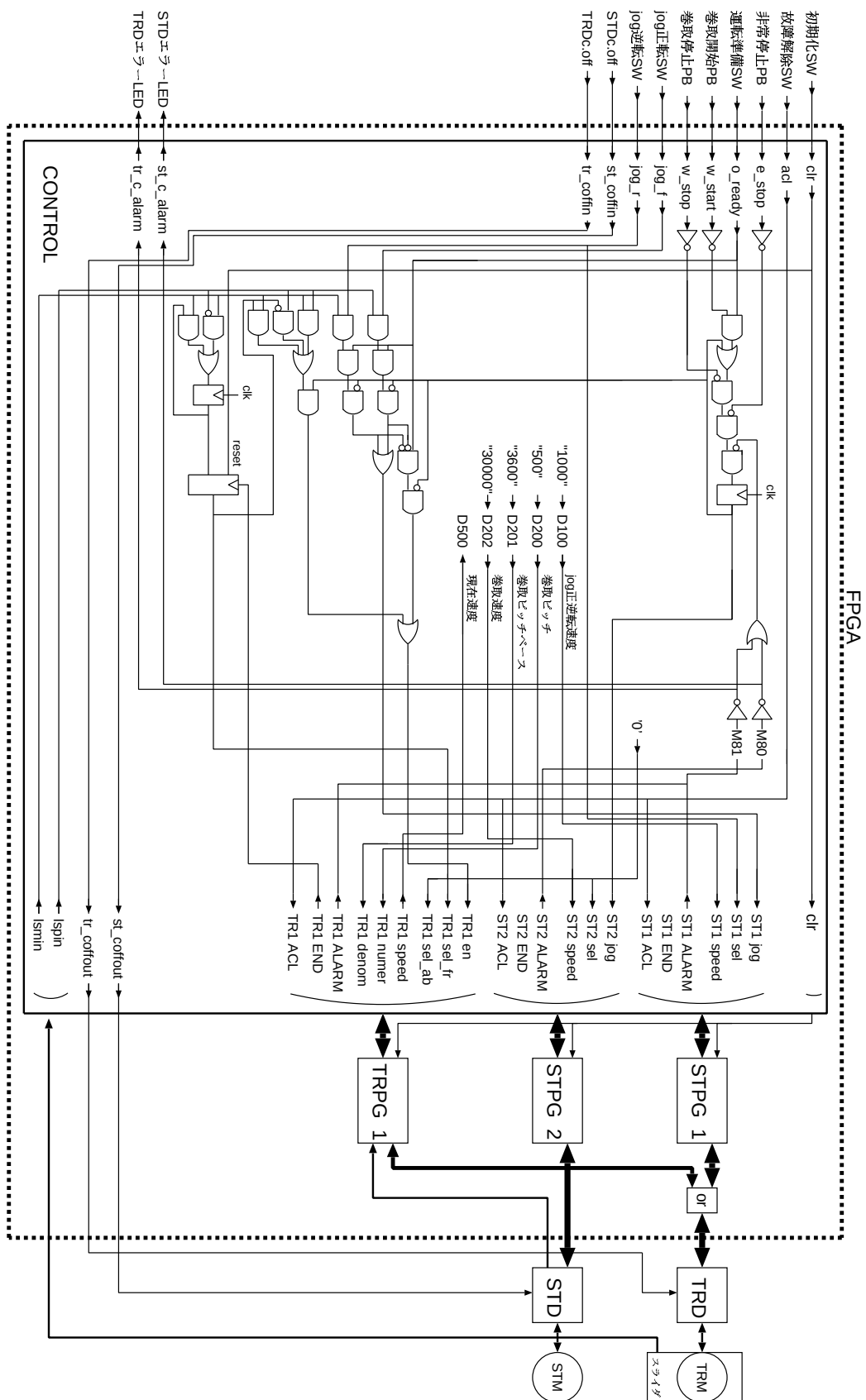


図 34: サンプル 3 改の制御部分のブロック図

B.4.2 信号線などの定義

信号線などの定義を以下に示す.

DIPSwitch は Close 側が '0', Open 側が '1' であり, PUSHSwitch は押すと '0', 離すと '1' である.

コントロール側入力信号以下に示す.

clr 初期化信号. DIPSwitch で入力.

acl 故障解除信号. DIPSwitch で入力. STD, TRD に対して送信する.

e_stop 非常停止信号. PUSHSwitch で入力. '0'=非常停止.

o_ready 運転準備信号. DIPSwitch で入力. '1'の間運転が可能.

w_start 巻取開始信号. PUSHSwitch で入力. '0'=巻取開始.

w_end 巻取終了信号. PUSHSwitch で入力. '0'=巻取終了.

jog_f jog 正転信号. DIPSwitch で入力. '1'=スライダ前進. '0'=停止.

jog_r jog 逆転信号. DIPSwitch で入力. '1'=スライダ後進. '0'=停止.

lspin スライダ位置 (前) 検出センサ信号 (LS+). '0'=検出, '1'=未検出.

lsmin スライダ位置 (後) 検出センサ信号 (LS-). '0'=検出, '1'=未検出.

st_coffin STDc.off 信号. DIPSwitch で入力.

tr_coffin TRDc.off 信号. DIPSwitch で入力.

コントロール側出力信号以下に示す.

st_c_alarm STD エラー信号. STD の状態を表示. '0'=正常, '1'=異常.

tr_c_alarm TRD エラー信号. TRD の状態を表示. '0'=正常, '1'=異常.

st_coffout STDc.off 信号. '0'=無効, '1'=有効.

tr_coffout TRDc.off 信号. '0'=無効, '1'=有効.

パルスジェネレータモジュールとの接続を以下に示す.

ST1jog STPG1 の jog と接続.

ST1sel STPG1 の sel と接続.

ST1speed STPG1 の speed と接続.

ST1ALARM STPG1 の C_ALARM と接続.

ST1END STPG1 の C_R_END と接続.

ST1ACL STPG1 の C_ACL と接続.

ST2jog STPG2 の jog と接続.

ST2sel STPG2 の sel と接続.

ST2speed STPG2 の speed と接続.

ST2ALARM STPG2 の C_ALARM と接続.

ST2END STPG2 の C_R_END と接続.

ST2ACL STPG2 の C_ACL と接続.

TR1en TRPG1 の jog と接続.

TR1sel_fr TRPG1 の sel_fr と接続.

TR1sel_ab TRPG1 の sel_ab と接続.

TR1speed TRPG1 の speed_a と接続.

TR1numer TRPG1 の numer と接続.

TR1denom TRPG1 の denom と接続.

TR1ALARM TRPG1 の C_ALARM と接続.

TR1END TRPG1 の C_R_END と接続.

TR1ACL TRPG1 の C_ACL と接続.

内部信号を以下に示す.

D100 jog 正逆転速度. 1Hz～30kHz (20ビット).

D200 巻取ピッチ. 1～999 (16ビット).

D201 巻取ピッチベース. 3600 (16ビット).

D202 巻取速度. 1Hz～30kHz (20ビット).

D500 入力パルス分周数. (25ビット).

M80 STD エラー信号. '0'=異常, '1'=正常.

M81 TRD エラー信号. '0'=異常, '1'=正常.

サンプル 3 改の制御部分と各 PG, ドライバの接続をを図 35 に示す。

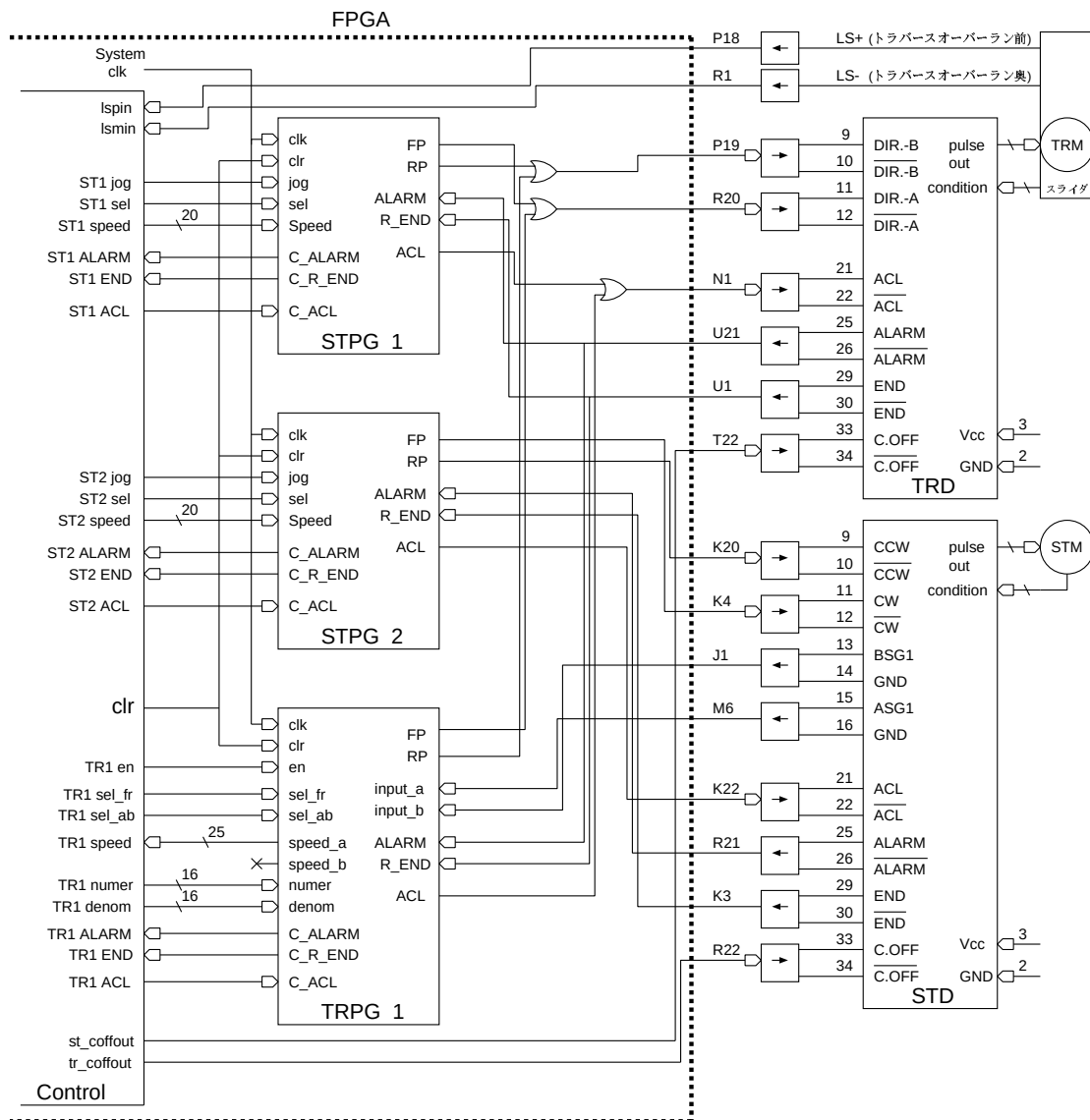


図 35: サンプル 3 改の制御部分と各 PG, ドライバの接続

図 35 中の FPGA ピンとドライバの接続部分の詳細は B.4.4 節に示す。

B.4.3 評価

サンプル3改の回路規模, 動作周波数の評価を行なった. 評価環境とターゲットデバイスを表13, 表14に示す.

表 13: 評価環境 (PC)

CPU	AthlonXP 3200+
MEM	1GB
OS	Windows2000Pro SP4

表 14: ターゲットデバイス

シリーズ	APEX20KE
デバイス	EP20K200EFC484-2X
LEs	8320
Pin	376

論理はVHDLで記述し, Altera社のQuartusII4.0を用いて, Analysis&Synthesis, Fitter, Assembler, Timing Analyzerを実行し見積りを行なう.

内部パラメータを以下に示す.

STPG1の内部パラメータを以下に示す.

外部クロック信号周波数 33.333MHz, 最大速度 500kHz, バイアス速度 0Hz, 加速時間 100msec, 減速時間 100msec, jog 速度 1kHz.

対応する各定数を以下に示す.

CLK_F 33,333,000.

MAX_S 500,000.

BIAS_S 0.

F_ACCEL 100,000.

F_SLOW 100,000.

ACCEL_C 50.

SLOW_C 50.

speed 1000.

STPG2 の内部パラメータを以下に示す.

外部クロック信号周波数 33.333MHz, 最大速度 30kHz, バイアス速度 0Hz, 加速時間 5000msec, 減速時間 5000msec, jog 速度 30kHz.

対応する各定数を以下に示す.

CLK_F 33,333,000.

MAX_S 30,000.

BIAS_S 0.

F_ACCEL 1,000.

F_SLOW 1,000.

ACCEL_C 6.

SLOW_C 6.

speed 30,000.

TRPG1 の内部パラメータを以下に示す.

分子 500, 分母 3600.

対応する各定数を以下に示す.

numer 500.

denom 3,600.

結果を表 15 に示す. 表 15 中の LEs は回路規模を示す. LE は論理機能を実現するための基本ユニットのことである. fmax は動作周波数を示す. Pins は使用する FPGA の I/O ピンの数を示す.

表 15: サンプル 3 改の回路規模と動作周波数

LEs	fmax (MHz)	Pins
430	50.38	31

この論理回路はデバイスの最大 LE 数 8,320, 最大ピン数 376, FPGA ボードのシステムクロック 33.333MHz と比較して十分実用可能な規模と速度であるといえる.

この論理回路をデバイスにダウンロードし, インターフェース回路経由でモータ, スライダに接続し問題無く動作することを確認した.

B.4.4 FPGA ボードと各ドライバの接続

FPGA ピンと STD との接続の対応を図 36 に、TRD との接続を図 37 に示す。
図 36、図 37 の括弧中の番号は FPGA 側のピン番号を示す。

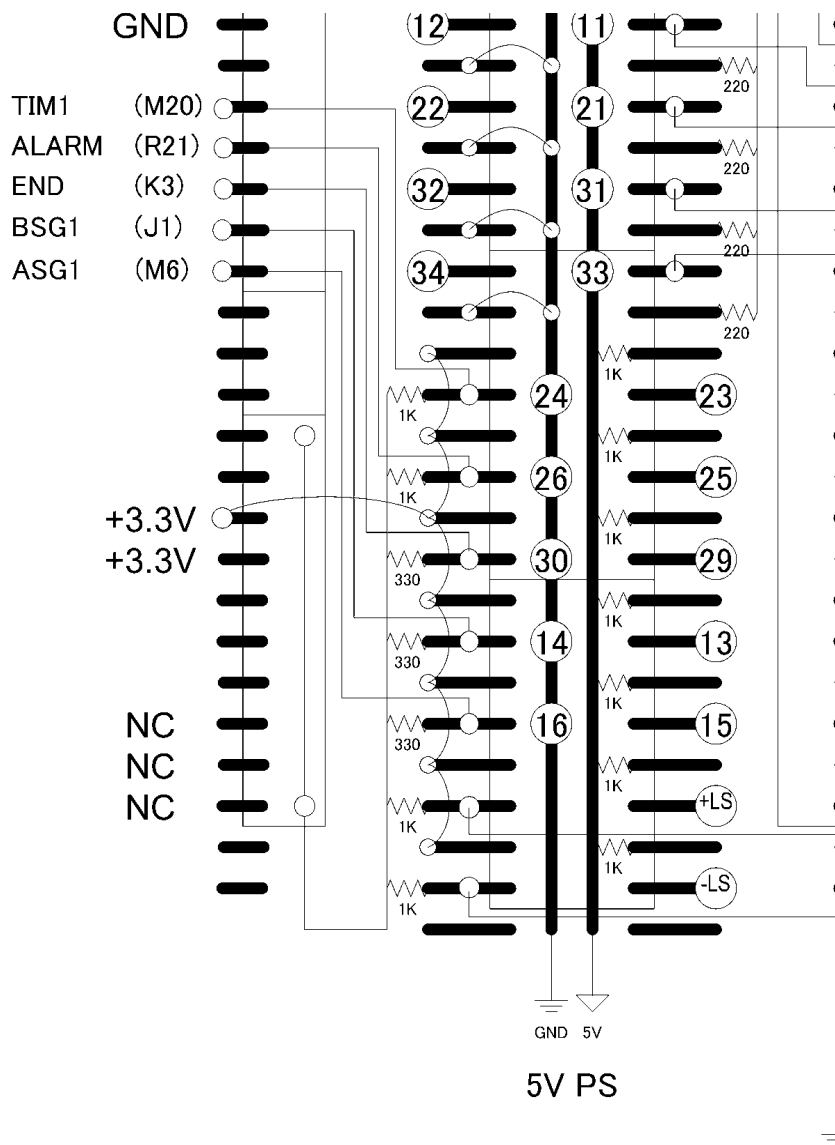


図 36: FPGA ピンと STD の接続

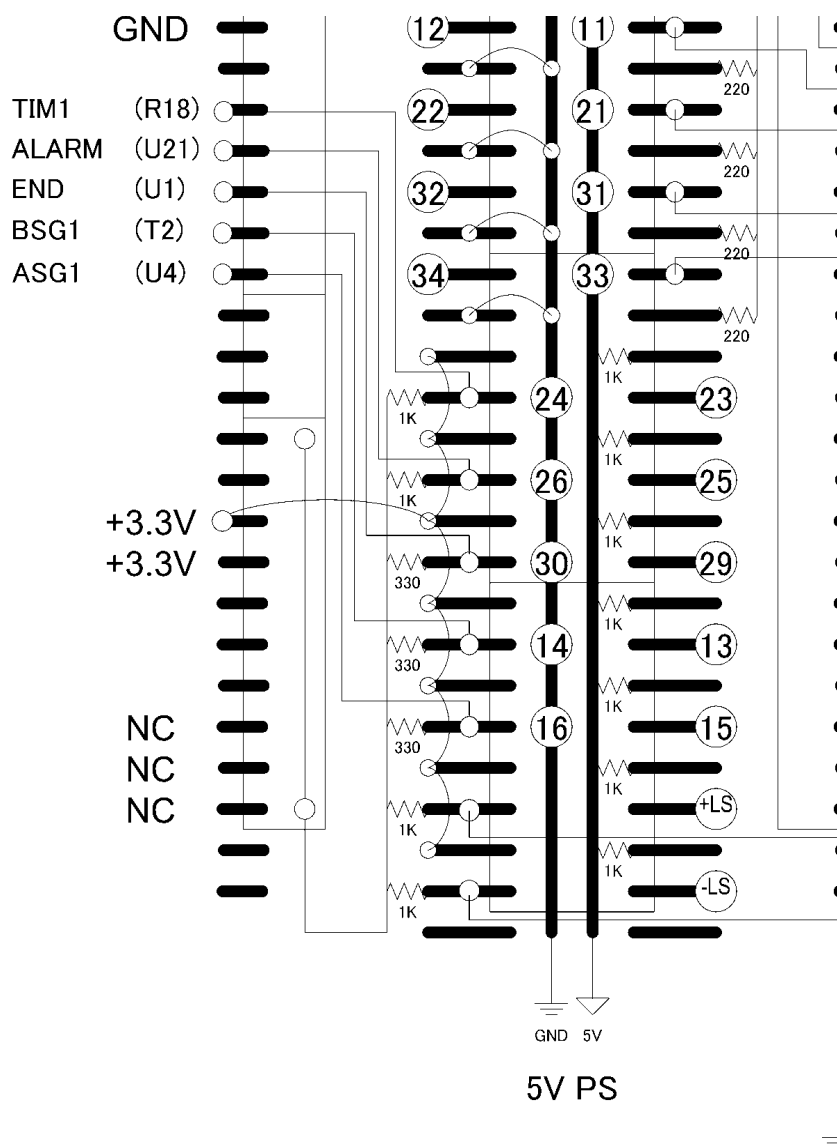


図 37: FPGA ピンと TRD の接続

FPGA ピンとドライバの接続の回路図を図 38 に示す。

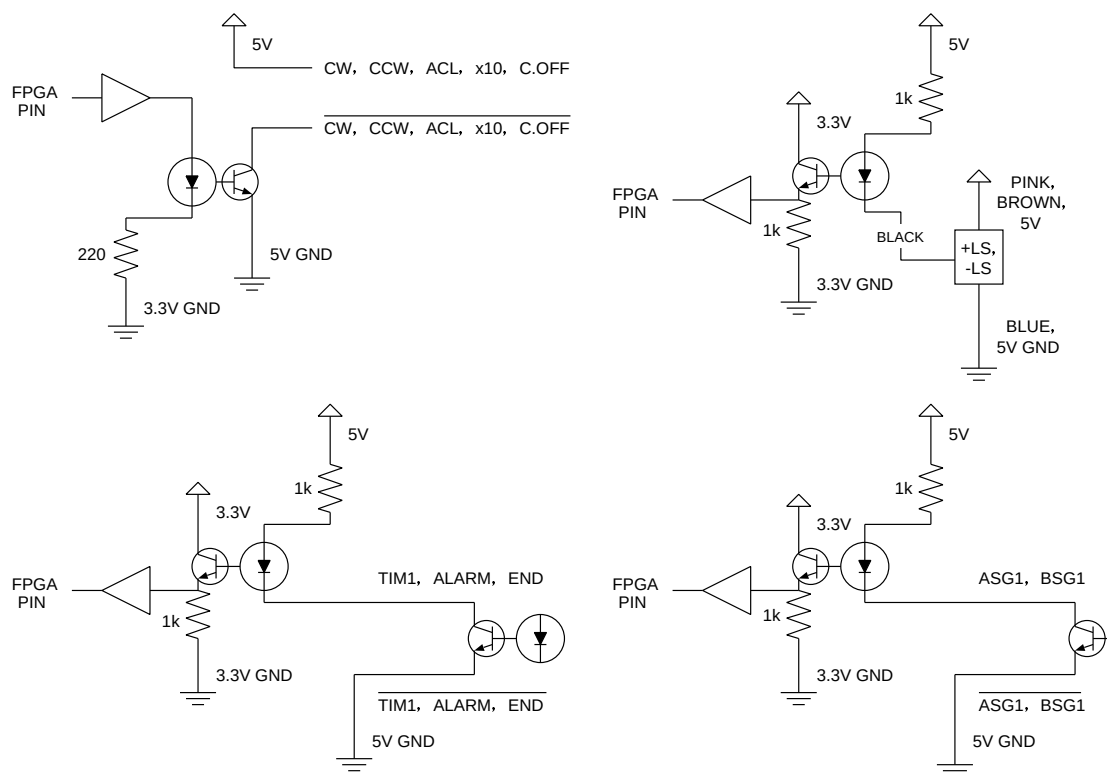


図 38: FPGA ピンとドライバの接続の回路図

参考文献

- [1] Adamski, M. A. and Monteiro, J. L.: PLD Implementation of Logic Controllers, *Proc. IEEE Int. Symp. Industrial Electronics (ISIE'95)*, Vol. 2, pp. 706–711 (1995).
- [2] Wegrzyn, M., Wolanski, P., Adamski, M. and Monteiro, J.: Field Programmable Device as a Logic Controller, *Proc. Int. Conf. Control'96*, Vol. 2, pp. 715–720 (1996).
- [3] 竹馬徹, 宮崎浩人: シーケンス制御方法および装置, 特開平 10-207711.
- [4] Wegrzyn, M., Adamski, M. A. and Monteiro, J. L.: The application of reconfigurable logic to controller design, *Control Engineering Practice*, Vol. 6, pp. 879–887 (1998).
- [5] 西田崇, 新村正明, 不破泰: 関数型言語を用いた FPGA によるシーケンスハードウェアの設計ツール, 信学技報 VLD98-142, pp. 9–16 (1999).
- [6] Miyazawa, I., Nagao, T., Fukagawa, M., Itoh, Y., Mizuya, T. and Sekiguchi, T.: IMPLEMENTATION OF LADDER DIAGRAM FOR PROGRAMMABLE CONTROLLER USING FPGA, *Proc. 7th IEEE Int. Conf. Emerging Technologies and Factory Automaton (EFTA'99)*, Vol. 2, pp. 1381–1385 (1999).
- [7] Welch, J. T. and Carletta, J.: A Direct Mapping FPGA Architecture for Industrial Process Control Applications, *Proc. Int. Conf. Computer Design 2000 (ICCD2000)*, pp. 595–598 (2000).
- [8] 池田亮, 市川周一: PLC プログラムの H/W 変換ツールに関する予備的検討, 平成 17 年電気学会全国大会 (投稿中).
- [9] : 八洲熱学株式会社, <http://www.yashima-ne.co.jp>.
- [10] 三菱電機株式会社: FX2N-10PG ユーザーズマニュアル, 三菱電機株式会社 http://wwwf3.mitsubishielectric.co.jp/members/o_manual/plc_fx/jy992d93301/jy992d93301d.pdf (2003).
- [11] 三菱電機株式会社: A171SCPU ユーザーズマニュアル, 三菱電機株式会社 <http://wwwf3.mitsubishielectric.co.jp/members/ssc/ib67200/ib67200h.pdf> (1997).
- [12] 三菱電機株式会社: ACPU プログラミングマニュアル (基礎編), 三菱電機株式会社 <http://wwwf3.mitsubishielectric.co.jp/members/plc/sh3435/sh3435q.pdf> (2003).

- [13] 三菱電機株式会社: ACPU/QCPU-A(A モード) プログラミング (共通命令編), 三菱電機株式会社 <http://wwwf3.mitsubishielectric.co.jp/members/plc/sh3435/sh3436o.pdf> (2003).
- [14] 三菱電機株式会社: SV13/22(リアルモード 編) プログラミングマニュアル (A171S/A273UH 対応), 三菱電機株式会社 <http://wwwf3.mitsubishielectric.co.jp/members/ssc/ib67259/ib67259g.pdf> (1998).
- [15] AlteraCorporation: *LPM Quick Reference Guide*, 日本アルテラ株式会社 <http://www.altera.co.jp/literature/catalogs/lpm.pdf> (1996).
- [16] 三菱電機株式会社: FX2N シリーズハンディマニュアル, 三菱電機株式会社 http://wwwf3.mitsubishielectric.co.jp/members/o_manual/plc_fx/jy992d61601/jy992d61601k.pdf (2003).
- [17] オリエンタルモーター株式会社: α STEP SRL シリーズ 取扱説明書, オリエンタルモーター株式会社 http://www.orientalmotor.co.jp/pdf_data/HM-6161-9J.pdf (2003).
- [18] オリエンタルモーター株式会社: α STEP 小型電動スライダ SRL シリーズ α STEP 仕様, オリエンタルモーター株式会社 http://www.orientalmotor.co.jp/pdf_data/HL-0104-2J1.pdf (2003).