

温冷感の過渡変化を考慮した空調制御器の最適設計

小島 一恭*

Optimal design of air-conditioning controller considering of thermal sensation

Kazuyuki KOJIMA

Abstract:

This paper proposes an optimal design of an air-conditioning controller considering of human thermal sensation. The human thermal sensation is typically estimated referring to temperature, humidity, and air-velocity. However, the thermal sensation is not only dependent on such parameters but also a staying time, human's physical condition, and a mental state. Therefore, the thermal sensation based on temperature, humidity and air-velocity often provide incorrect index. In this paper, to solve this problem, we try to build a model of human thermal sensation. At first, we clarify human's thermal sensation is dependent on a staying time. Then, a prediction model of the thermal sensation is constructed by utilizing machine learning. Finally, we try to optimize an air-conditioning controller which is built out of FPGAs by utilizing our developed thermal sensation model.

KEY WORDS : Optimal design, machine learning, air-conditioning, control system, evolutionally computing, FPGA

要旨:

本稿では人間の温冷感を考慮した空調制御器の最適設計手法を提案する。空調機器の制御で用いられる温冷感指標は一般に室温、湿度、気流速に基づいて予測される。しかしながら、人間の温冷感はいくらのパラメータだけではなく体調や精神状態、その温熱環境の滞在時間などにも影響される。このため、空調制御器が予測した温冷感と実際の人間の温冷感とでミスマッチが生じ、不快と感ずることがしばしば起こる。そこで本研究では、このミスマッチの解消のため、温熱環境での滞在時間に注目し、空調制御器の最適化を図る。本稿では、まず、機械学習を使用して実験に基づいた温冷感の過渡応答モデルを構築する。次に、このモデルを使用して計算機上で空調制御シミュレーションを行い、空調制御器の最適化を図る。その際、空調制御器にはFPGAを用い、FPGA内部の論理回路を記述するVHDLを進化計算を利用して自動生成する。得られた制御器の性能を検証するため、種々の温熱条件でシミュレーションを実施する。

キーワード : 最適設計, 機械学習, 空調制御, 制御系, 進化計算, FPGA

1. はじめに

近年の電子技術の著しい発展に伴い、身の回りにある数多くの制御系に高機能・高性能な電子部品が組み込まれ、より複雑、高度な制御が行われるようになってきている。このような制御系の多くは、センサから得られたアナログ信号をデジタル信号に変換し、それを MPU (Micro Processing Unit) や DSP (Digital Signal Processor), ASIC (Application Specific Integrated Circuit), CPLD (Complex Programmable Logic Device), FPGA (Field

Programmable Gate Array) などを用いて演算処理し、制御量や制御シーケンスを決定している。制御器を設計する際、設計者はシステムを構成するセンサやアクチュエータ、演算器などの資源を機械的・電氣的かつ有機的に結合し機能を発現させる。複雑かつ高度な制御系の設計は、近年では、C言語やHDLなどの言語を使用してソフトウェア的に行われることが多くなってきている。

空調装置はこのようなプロセスで開発される制御系を持つシステムのうちの一つである。空調装置の制御系設計では、目的の仕様を満たすために構成すべき最適解がただひとつ存在するわけではなく、多様な設計解を持つ。このため設計者は多様な設計解

*湘南工科大学 工学部 機械工学科 准教授

の中から目的の仕様を満たす設計解を見つけ出さなければならず、制御系設計には多くの労力を要している。さらに、近年の空調制御では人間の快適性を考慮した制御系の設計が求められてきており、このような人間をも制御系に含めなければならないことが、制御系の設計をさらに難しいものとしている。

一方、近年、多様な設計解の中から発見法的に最適な設計解を探索する手法として進化計算を応用する研究が盛んになされている[1] - [3]。著者らはこれまでに進化計算を応用し、制御系の最適化をはかる手法を提案してきた[4], [5]。これらの手法では、C言語で記述される制御プログラムを進化計算を使用して適応的に進化させ、目的に合致する制御プログラムを生成する。これまでの研究では、目的の制御系を構成するプログラムの自動生成が可能であることを確認する一方で、進化計算特有の問題から、プログラムサイズが大きくなったり、制御器のリアルタイム性が損なわれたりする場合があることが明らかとなった。そこで、本稿では逐次処理的な動作をするC言語ではなく、並列動作が可能な論理回路を記述するハ

ードウェア記述言語 (HDL) を進化計算により最適化し、これらの問題を解決する。この手法は多くの制御系に汎用的に適用可能な枠組みである。入出力信号の演算処理を行う論理回路を記述する HDL を進化計算を用いて合目的に自動生成する。

本稿では、この枠組を応用し、空調制御器を自動生成する。制御器の演算部分には FPGA を使い、内部の制御回路を記述する HDL を自動生成する。

2. 温熱快適指標

表1に一般によく用いられる温熱快適指標を示す。空調機器で用いられる温冷感指標は SET* や PMV であるが、測定可能な室温、湿度、気流速を用いて計算する。しかし、人間の温冷感は温度、湿度、気流速だけではなく、放射温度、代謝量、着衣量のほか、体調や心理状態、滞在時間などにも影響されるため、これらの指標に基づく空調制御では、予測した温冷感と実際の人間の温冷感とでミスマッチが生じ、計算上は快適な温熱空間であっても不快と感ずることがし

表1 代表的な温熱快適指標

	指標	提案者	特徴
物理計算に 基づく指標	カタ冷却力	L.Hill (1916)	人体の冷却効果の関連から環境を評価する。
	グローブ温度	H.M.Vernon (1930)	温度、気流、輻射を総合的に評価する。
		Madsen (1972)	温度、気流、輻射の総合的評価をもとに着衣量、作業量、湿度を電子回路の系の定数としてあたえFMVを指示する。
生理反応に 基づく指標	P4SR	McAdele (1947)	発汗量の予測を温度、輻射、気流、作業量を変数とするモノグラムを用いて行う。
主観的経験的指標	有効温度 (ET)	Houghten, Yaglou (1923)	温度、湿度、気流の3つの温熱要素の組み合わせによる温冷感を、温度の尺度で表示。多くの被験者の主観的判断に基づく。
	不快指数 (THI)	アメリカ気象局 (1959)	温度と湿度のみを用いて有効温度を近似的に示す。
	WBGT	Yaglou, Minard (1957)	温度、気流、輻射、湿度の計測により有効温度を近似的に示す。
熱平衡式に基づく指標	作用温度 (TO, STO)	Gagge (1937)	輻射、温度、熱放射伝達率、対流熱伝達率を含む等価仮想温度である。
	平均予測温熱申告 (PMV)	Fanger (1970)	温度、湿度、気流、輻射、代謝量、着衣量から温冷感を数値として表示。その環境に滞在する人々の不満足に感じる割合は、PPDで表示する。
	新有効温度 (ET*, SET*)	Gagge, Nishi (1971)	発汗による体温調節を含む熱平衡式モデルに基づき、温度、輻射、湿度、気流、着衣、作業量、気圧、人工空気などの変数より、生理因子として皮膚温、体内温、発汗量、貯熱量等を総合的に評価する。

ばしば起こる。本研究では、このミスマッチの解消のため、先に上げた要因のうち、特に、時間的な影響と心理的な影響に注目し、制御器の最適設計を行う。

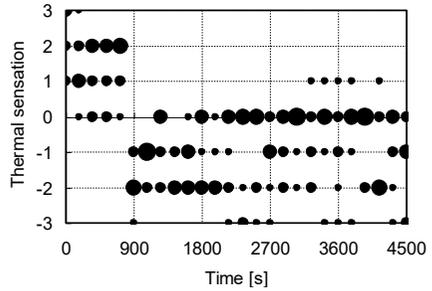
3. 温冷感の過渡変化

温冷感、快適感に及ぼす時間的・心理的な影響を調べるため、温湿度および気流速を調整可能な実験用室内環境（図1）を使用して被験者による温冷感、快適感の申告実験を行う。本稿では、被験者がある温熱環境1（室温30℃、湿度60%、代謝量1.0[met]、着衣量1.0[met]、PMV=1.97）から別の温熱環境2（気温25℃、湿度30%、代謝量1.0[met]、着衣量1.0[clol]、PMV=0.07）に移動した場合の温冷感、快適感の過渡応答を調べる。被験者が30℃、60%の環境において30分以上経過した後に実験を開始し、開始後15分間はこの環境下での温冷感、快適感の申告を3分毎に行う。15分経過後、直ちに気温25℃、湿度30%の部屋に移動し、その後、1時間にわたり3分毎に温冷感、快適感を申告する。温冷感の申告は「暑い(+3)」、「暖かい(+2)」、「やや暖かい(+1)」、「どちらでもない(0)」、「やや涼しい(-1)」、「涼しい(-2)」、「寒い(-3)」の7段階、快適感の申告は、「非常に快適(+3)」、「快適(+2)」、「やや快適(+1)」、「何も感じない(0)」、「やや不快(-1)」、「不快(-2)」、「非常に不快(-3)」の7段階とする。一般に、温冷感、快適感は主観的なものであるため、事前にどの状態が「寒い」状態でのどの状態が「暑い」状態であるということを教示せず、主観評価で申告を行ってもらう。また、快適感には温熱的快適感のみを申告し、その他の快適感には申告しない。心理負荷としては一桁の加減乗除算を使用する。

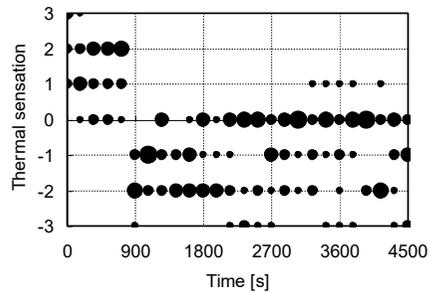
図2に温熱的快適感、温冷感申告実験の結果を示



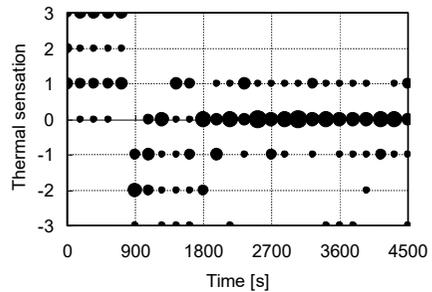
図1 実験用室内環境



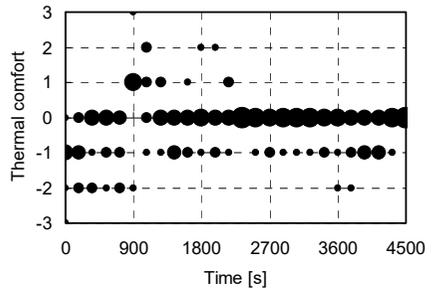
(a) 温冷感の過渡変化（心理負荷なし）



(b) 温熱的快適感の過渡変化（心理負荷なし）



(c) 温冷感の過渡変化（心理負荷あり）



(d) 温熱的快適感の過渡変化（心理負荷あり）

図2 温冷感・快適感の過渡変化

す。図 2(a), 図 2(b)はそれぞれ、心理負荷がない場合の温冷感申告の過渡応答、快適感申告の過渡応答である。また、図 2(c), 図 2(d)はその心理負荷のある場合である。黒丸の大きさは全申告に対する割合を示し、黒丸が大きいほど申告度数が高いことを示す。温冷感の過渡応答は、PMV の指標で中立な環境が保たれていたとしても、時間とともに被験者の申告は変化する。また、快適感に関しては、心理負荷がない場合は「快適」側の申告が多かったのに対し、心理負荷がある場合は、「何も感じない」が圧倒的多数になっていることがわかる。特に快適感に関しては、心理負荷がない場合の申告において個人差による影響が強く現れ、結果にばらつきがあるのに比べ、心理負荷がある場合には「何も感じない」という申告に集中していることが確認できる。

4. 快適性の過渡応答モデル

温冷感や温熱的快適感とは、前節で述べた通り、たとえば温熱環境が一定であっても、心理的・時間的要因により変化し、また、個人差があるため、同一条件下でも分布を持つといえる。したがって、このような温冷感・快適感とはたとえば室内の環境側 4 要素（温度、湿度、気流速、放射温度）と人体側 2 要素（着衣量、代謝量）が既知であっても熱流体力学的な演算で推定することが難しい。そこで、本稿では空調制御系を構成する各要素のパラメータに温熱環境での曝露時間を加えた情報から、階層型ニューラルネットワークを用いて過渡応答を考慮した温冷感・快適感の分布を推定するモデルを構築する。階層型ニューラルネットワークの構成は対象とする空調システムによって異なるが、たとえば、図 3 に示すニューラルネットワークが考えられる。このモデルでは、温冷感および快適感の分布が正規分布であると仮定し、平均値と分散を推定する。

5. 進化計算による制御器の最適設計

5.1 進化計算の概要

本稿での最適設計の枠組みの基本となる進化計算について概説する。図 4 に進化計算の枠組みを示す。進化計算のメカニズムは、生殖、突然変異、遺伝子組換え、自然淘汰、適者生存という生物の進化過程に着想を得ている。最適化問題の解の候補群が生物の個体群の役割を果たし、コスト関数によってどの解が生き残るかを決定する。初期の個体群の中から生存した複数の個体で新たな個体群を形成することを世代交代と呼び、世代交代を繰り返すことで個体群の

進化、すなわち解候補の最適化を行う。生存の適者選択を確率的にすることで、各世代の解候補群に含まれる解候補の多様性を維持し、大域探索を可能としている。

最適化問題の決定変数ベクトル x を、 N 個の記号 s_j ($j=1, \dots, N$) の列で次のように表す。

$$x : s = s_1 s_2 s_3 \dots s_N \tag{1}$$

この記号列 s を N 個の遺伝子座からなる染色体とみなす。 s_j は第 j 遺伝子座における遺伝子であり、 s_j の取り得る値が対立遺伝子である。対立遺伝子としては、ある整数の組、ある範囲の実数値、単なる記号などを考えればよい。それぞれが式(1)で表される K 個の個体からなる集合を考え、世代 n における個体集合 $P(n)$ が遺伝子の複製・変異を経て次の世代 $n+1$

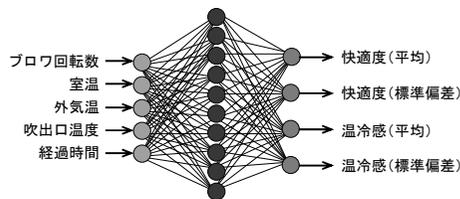


図 3 ニューラルネットワーク

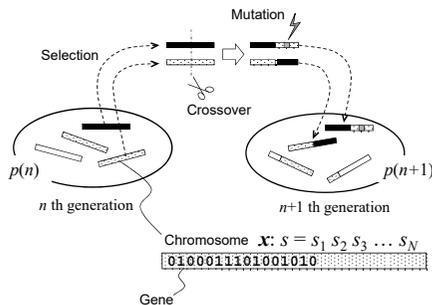


図 4 進化計算の枠組み

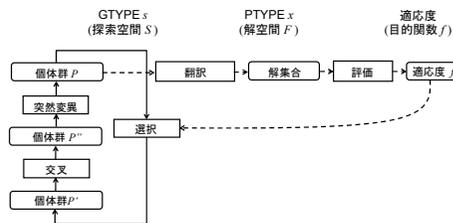


図 5 進化計算の流れ

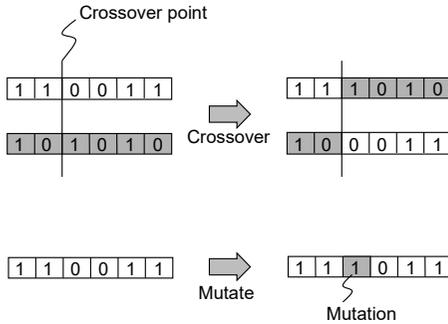


図 6 交叉・突然変異

における個体集合 $P(n+1)$ に変わるものとする。このような世代の更新が繰り返され、更新のたびに、より最適値に近い解 x を表わす個体を選択されて、増殖するようにすれば、やがて最適解が得られるであろうというのが遺伝的アルゴリズムの基本的な考え方である。この際、最適化問題での目的関数値がよい解ほど、それに対応する個体の評価値（適応度）が高くなるようにしておく。実際の生物では、染色体の変異と選択によって種の進化が実現されるが、遺伝的アルゴリズムでは主に選択、交叉、突然変異という遺伝的操作を用いる。これら 3 種類の演算子を用いた遺伝的アルゴリズムの全体的手順を図 5 に示す。選択操作は個体集合から適応度の高い個体が高い確率で選択されるように個体を選択する操作をいう。また、交叉は選択された 2 個体の記号列を途中で入替える操作である。突然変異は個体集合からある比率（突然変異率）で選択された個体の遺伝子を新たに対立遺伝子で書換える操作である。図 6 に交叉および突然変異操作の一例を示す。

5.2 CPLD/FPGA と VHDL

CPLD, FPGA はともにプログラム可能な LSI の一種で、論理ブロックの規模が比較的小さい SRAM ベースのものを FPGA、論理ブロックの規模が大きい EEPROM ベースのものを CPLD と呼ぶ。双方とも HDL (Hardware Description Language) によって内部ロジックを設計することが可能である。これらの LSI と同様に HDL を用いて設計できるデバイスとして ASIC があるが、CPLD や FPGA などのプログラマブル・デバイスは、設計した回路を直ちに実機評価できる、仕様変更にも柔軟に対応可能であるなどのメリットがあるため、開発時の用途に適している。本稿では制御器として CPLD を使用するが、提案する枠組みはデバイスを限定するわけではなく、HDL で設計できるあらゆるデバイスに適用可能であ

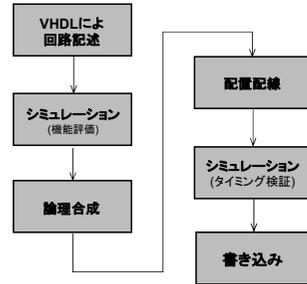


図 7 HDL を用いた回路設計

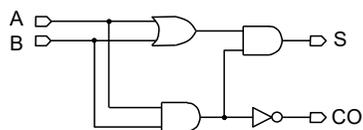
```

library IEEE;
use IEEE.std_logic_1164.all;

entity HALF_ADDER is
  port(
    A,B : in std_logic;
    S,CO : out std_logic);
end HALF_ADDER;

architecture DATAFLOW of HALF_ADDER is
  signal C, D : std_logic;
begin
  C <= A or B;
  D <= A nand B;
  CO <= not D;
  S <= C and D;
end DATAFLOW;
    
```

(a) VHDL による回路記述



(b) 対応する回路

図 8 VHDL による回路記述の一例

る。また、現在、HDL として利用されている標準的な言語には VHDL と Verilog HDL の 2 種類あるが、本稿では VHDL を使用する。2 つの HDL はほぼ同等の記述能力を持ち、ここで示す枠組みは双方に対し重複して使用できる部分が多い。図 7 に VHDL を用いた LSI 設計の流れを示す。VHDL で記述したロジックはシミュレータや論理合成ツールなどを使用して検証および変換がなされ、デバイスに書き込み可能な形式となる。CPLD や FPGA がターゲットデバイスとなっている場合には、ダウンロードケーブ

ルを介して、デバイスに論理回路を書き込み、簡単に目的の LSI を得ることができる。図 8 に簡単な VHDL の記述例と対応する論理回路を示す。

5.3 進化計算を用いた制御器最適設計

Xilinx XC9572 を例に進化計算を用いた制御器最適設計の枠組みについて説明する。図 9 に XC9572 の内部ブロックを示す[11]。XC9572 は外部端子 44 本 (ユーザ I/O 34 本)、マクロセル 72 個、内部ゲート数 1600 個と小型の CPLD である。設計者は 34 本のユーザ I/O の中から選択的に入力および出力を決定し、それらのピン配置を定義する。さらに、入出力に対して信号を定義する。制御系の場合には、CPLD の入出力ピンに対してセンサーやアクチュエータを関連付けることができる。図 10 に I/O ピンの割当ての例を示す。図 10 (a) は、センサー 2 個、アクチュエータ 1 個を関連付けた例である。この例では、センサー値をそれぞれ 8 ビットのデジタル信号として CPLD に入力し、アクチュエータへの参照信号として 8 ビットのデジタルデータを出力する。図 10 (b) では、2 つのセンサー値 (それぞれ、8 ビット、4 ビット) のほかにスイッチ入力を 2 つ加えている。アクチュエータへの参照信号としての出力は 8 ビットである。

進化計算を用いた制御器の最適設計の枠組みでは、制御器の設計から評価、検証に至る一連のプロセスを自動化することができる。設計者は判定基準のみをこのプロセスに組み込めばよい。これまでに行われているこの分野の研究ではハードウェア内部の回路構成を設定するビット列を染色体にコーディングするため、染色体の符号はハードウェアに依存して遺伝子座の意味が固定されていた。このため、デバイ

スを変更する場合には、ビット列が持つ意味を別のデバイスにあわせて変換するか、染色体への符号化を改めてしなおす必要があった。この場合、目的を限定し、内部構造をある程度決定した上で最適化がなされることが多く、使用目的を変更する際には、評価関数だけでなく染色体のビットの持つ意味づけを改めてしなおす必要があった。本論文で提案する枠組みでは、CPLD の内部ロジックを記述する VHDL を染色体上に符号化して進化計算を行う。これにより、特定のデバイスに依存することなく、制御器設計の

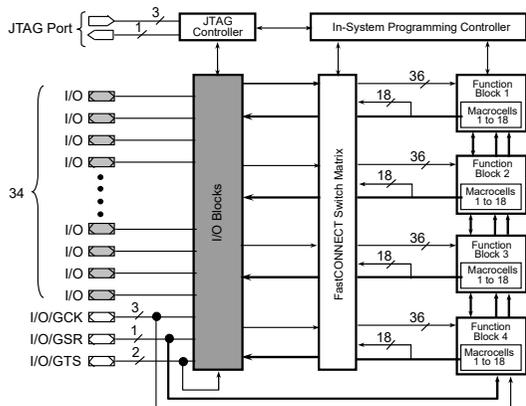
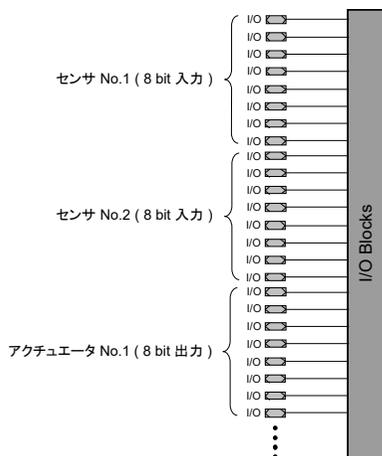
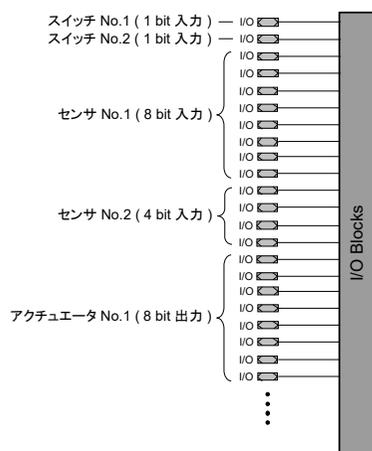


図 9 XC9572 の内部ブロック図



(a) 応用例 1



(b) 応用例 2

図 10 I/O ピンの割当て

自動化が可能となる。また、内部構造は記述の矛盾を生じない程度に自由度を残し、構造を固定することも、固定しないこともできるようにする。

5.4 VHDL の符号化と染色体構造

CPLD を用いた制御器設計の一連のプロセスを自動化するために進化計算を用いて VHDL の記述を最適化する。VHDL は記述能力の高い言語であり、さまざまなレベルでの記述が可能である。例えば、システム全体のアルゴリズムやアーキテクチャを記述したり、データの流れやモータの制御などをモデル化してシステム全体を抽象度の高いレベルで記述したり、ロジック回路の自動生成が可能なレベルで記述したりすることができる。ところが、システム全体のアルゴリズムの記述や抽象度の高いレベルで

の記述をハードウェア進化に適用すると、下位レベルの設計や抽象化のためのモデル化の作業が設計者に残される。この場合、自動設計の枠組みに設計者の関与が残され、制御器設計プロセスを一巡して自動化することが困難になる。そこで、ロジック回路が自動生成可能なレベルの記述を染色体に符号化して、設計者が関与しなくてよいようにする。

図 11 に生成される VHDL の一例を示す。図 11 の VHDL は(a)エンティティ宣言部、(b)信号宣言部、(c)アーキテクチャ宣言部の 3つの部分で構成される。(a)エンティティ宣言部では CPLD の入出力信号が定義され、また、(b)信号宣言部では CPLD の内部信号が定義される。信号の型は主に `std_logic` 型と `std_logic_vector` 型が使用される。`std_logic` 型は信号を単体で取扱う場合に使用でき、いくつか

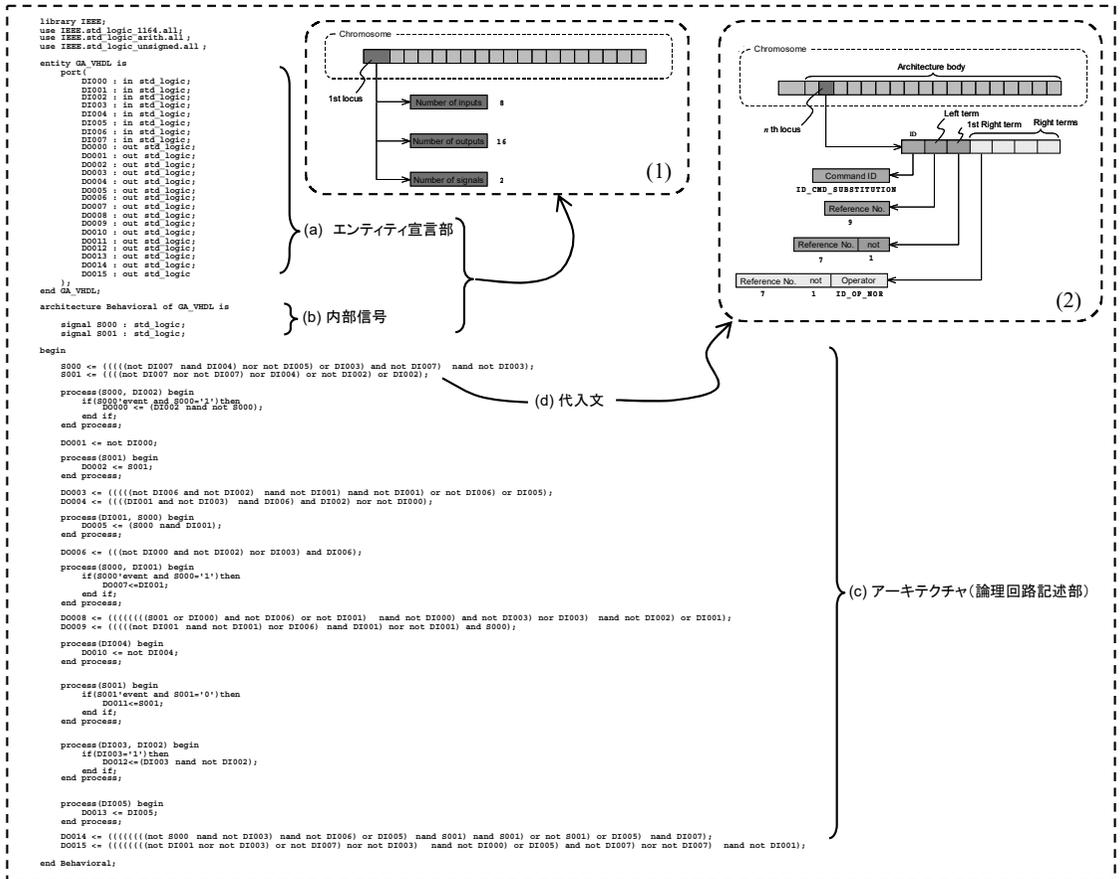


図 11 自動生成した VHDL

の信号をまとめて取扱う場合には `std_logic_vector` 型を使用できる。メンテナンス性や可読性を向上させるためには、`std_logic` 型と `std_logic_vector` 型を適宜使い分けるほうがよい。しかし、進化計算の枠組みに適用する際には、使用する型が増えると染色体の解読処理が複雑化するだけでなく、探索域も増大する。そこで、本枠組みではすべての信号を `std_logic` 型で定義する。`std_logic_vector` 型を使用した VHDL の記述は、複数の `std_logic` 型を使用した VHDL の記述で置換できるため、このようにしても本枠組みで生成される内部ロジックが制限されるわけではない。入出力信号、内部信号をすべて同一の `std_logic` 型にすると、その数のみを染色体上に符号化すれば、VHDL

の記述を復元できる。そこで、図 11 (1)に示す通り、染色体の先頭の遺伝子に入力信号数、出力信号数、内部信号数を符号化する。図 11 (a)の場合には、入力信号が 8、出力信号が 16 と設定され、内部信号がハードウェアの許す範囲で任意に設定される。図 11(b)の VHDL では 2 と設定されている。

図 11 (c)のアーキテクチャ宣言部では、代入文とプロセス文を使用して制御ロジックを記述している。VHDL においてアーキテクチャ宣言部に記述可能な表現は非常に多くあるが、ここでは、紙面の都合上、代入文の記述の符号化のみを示す。図 11 (2)は代入文の符号化を図示したものである。図に記されている値は、図 11 (d)の 2 番目の代入文 (左辺が「`s001`」の文) に相当している。染色体の第 2 番目以降の遺伝子はすべて同等の扱いであり、ここにアーキテクチャ宣言部の記述が符号化される。染色体は多重のリスト構造になっており、階層の数は特に制限していない。新たな VHDL の記述に対応させる場合には、必要な数だけ階層を設けることになる。2 階層目のリストの先頭は必ず ID になっており、この ID により、代入文であるのか、プロセス文であるのか、あるいは他の記述であるのかを判別することができる。ID が「`ID_CMD_SUBSTITUTION`」の場合は、代入文を表し、同時に、下層のリスト構造も図示する通りに一意に決まる。すなわち、2 階層目の 2 番目の遺伝子は VHDL の記述の代入文の左辺を表し、3 番目の遺伝子は代入文の右辺の初項を表し、4 番目以降は代入文の右辺の 2 項目以降を表すことが一意に決まる。リスト構造の 3 番目の階層に実際の値が格納されている。参照番号は、信号の番号を表し、入力信号、内部信号、出力信号の順に 0 番から連番で付したものである (表 2)。図 11 (b)では、左辺の参照番号が「8」となっているが、これは、1 番目の内部信号を表している。参照番号は、遺伝的操作での矛盾発生を回避するために導入したものである。右辺の初項に相当する遺伝子には参照番号と正負の論理を決定する値を格納する。右辺の 2 項目以降は参照番号と正負の論理、前の項との論理演算子を決定する値を格納する。

表 2 信号の参照番号

Reference No.	Signal name	VHDL variable name
0	Input No. 1	DI000
1	Input No. 2	DI001
2	Input No. 3	DI002
3	Input No. 4	DI003
4	Input No. 5	DI004
5	Input No. 6	DI005
6	Input No. 7	DI006
7	Input No. 8	DI007
8	Signal No. 1	S000
9	Signal No. 2	S001
10	Output No. 1	DO000
11	Output No. 2	DO001
12	Output No. 3	DO002
13	Output No. 4	DO003
14	Output No. 5	DO004
15	Output No. 6	DO005
16	Output No. 7	DO006
17	Output No. 8	DO007
18	Output No. 9	DO008
19	Output No. 10	DO009
20	Output No. 11	DO010
21	Output No. 12	DO011
22	Output No. 13	DO012
23	Output No. 14	DO013
24	Output No. 15	DO014
25	Output No. 16	DO015

5.5 可変長染色体の構造の制約と遺伝子操作

VHDL の入力信号数と出力信号数は設計者が決定する制御系の仕様によって変化する。また、内部信号も進化計算により生成されるロジックによりその必要数が増える。さらに、VHDL の記述では、代入文の右辺の長さ、プロセス文のセンシティビリティリストに列挙される信号の数、そもそも、VHDL の記述の行数自体が固定ではなく、制御器の設計に依存して大きく変化する。このため、多重リスト構造の染

染色体を VHDL のさまざまな記述に対応できるように、あらゆる階層で可変長としている。このような染色体構造とした場合に問題となるのは、遺伝的操作、特に交叉である。染色体のいたるところで染色体長の不整合が生じているため、単純に交叉することができない。リスト毎の交換を行おうとしても、VHDL の記述が代入文なのかプロセス文なのかによって、遺伝子の多重リスト構造が大きく異なるため、リストを交換してしまうと遺伝子解釈で矛盾を生じることになる。これらの問題を解決するため、本枠組みでは次のような制約を設ける。

1. 最上位層における染色体長は、内部信号数と出力信号数に 1 だけ加えた長さとする。
2. 最上位層の染色体のエンティティ宣言部には内部信号、出力信号の順で、かつ、優先度の低いものから、それぞれ一つの遺伝子上に一回だけに符号化する (図 12)。
3. すべての内部信号に対し優先度を決め、優先度の高い内部信号は優先度の低い内部信号と入力信号のみで記述する。
4. すべての信号を参照番号で符号化する。
5. 交叉は最上位層の染色体のみで行い、交叉点は選択された 2 つの染色体のうち、短い側の染色体長を超えない位置で行い、交叉点が奇数の場合は、先頭の遺伝子を入れ替える。

以上の制約を設けることで、矛盾を生じさせることなく、初期化、交叉、突然変異が可能となる。そして、これらの制約は論理合成不可能な VHDL の生成を排除することが目的であるため、進化計算の過程で生成されるロジックの複雑さを制限することはない。上述の制限を設けたとしても、十分に複雑なロジックの生成が可能である。但し、最適な制御ロジックは必ずしも複雑さに依存するものではない。

6. 快適性の過渡応答を考慮した空調制御器の最適設計

6.1 制御対象の空調モデル

本論文では、設計対象として図 13 に示す室内空調モデルを考える。本モデルは、流入口から流入した空気を一旦すべて冷却し、その後、冷却した空気の一部を再加熱して目的の温度の空気を生成し、室内に吹

出して空調を行う。室内への吐出量はブロワの回転数によって調整し、再加熱した空気と冷気との混合比はミックスドアの開度によって調整する。図 14 に示すように、本論文では、進化ハードウェアを適用する際の制御器への入力は車室内の温冷感とする。また、制御器からの出力はブロワ開度、ミックスドアの開度とし、それぞれ 8bit のデータとして取扱う。温冷感は、2 章の実験に基づいてモデルを構築し、算出する。

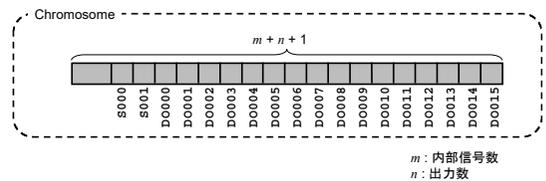


図 12 染色体長

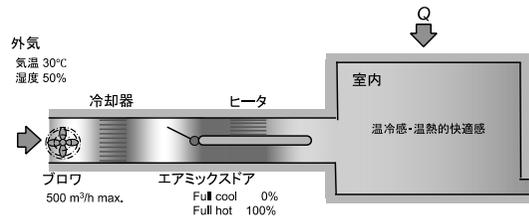


図 13 制御対象の空調モデル

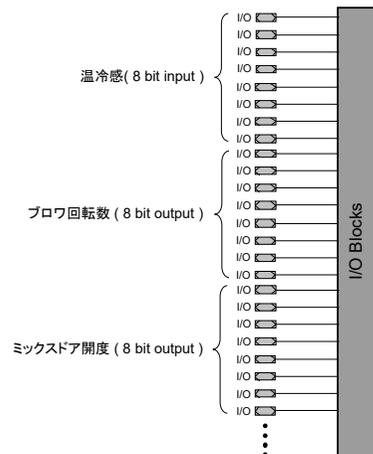


図 14 I/O ピンの割当て

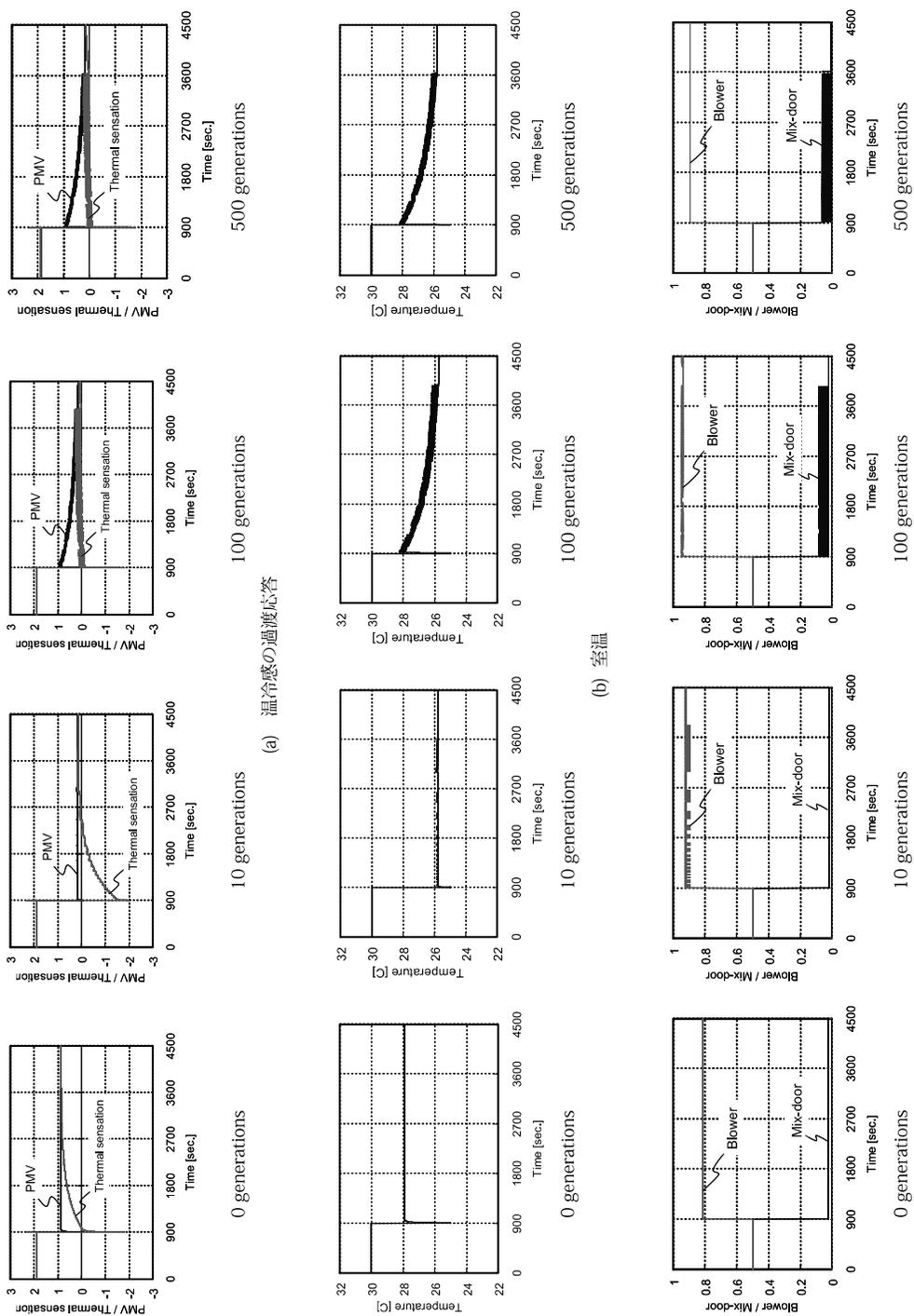


図15 シミュレーション結果(条件1)

温冷感の過渡変化を考慮した空調制御器の最適設計 (小島)

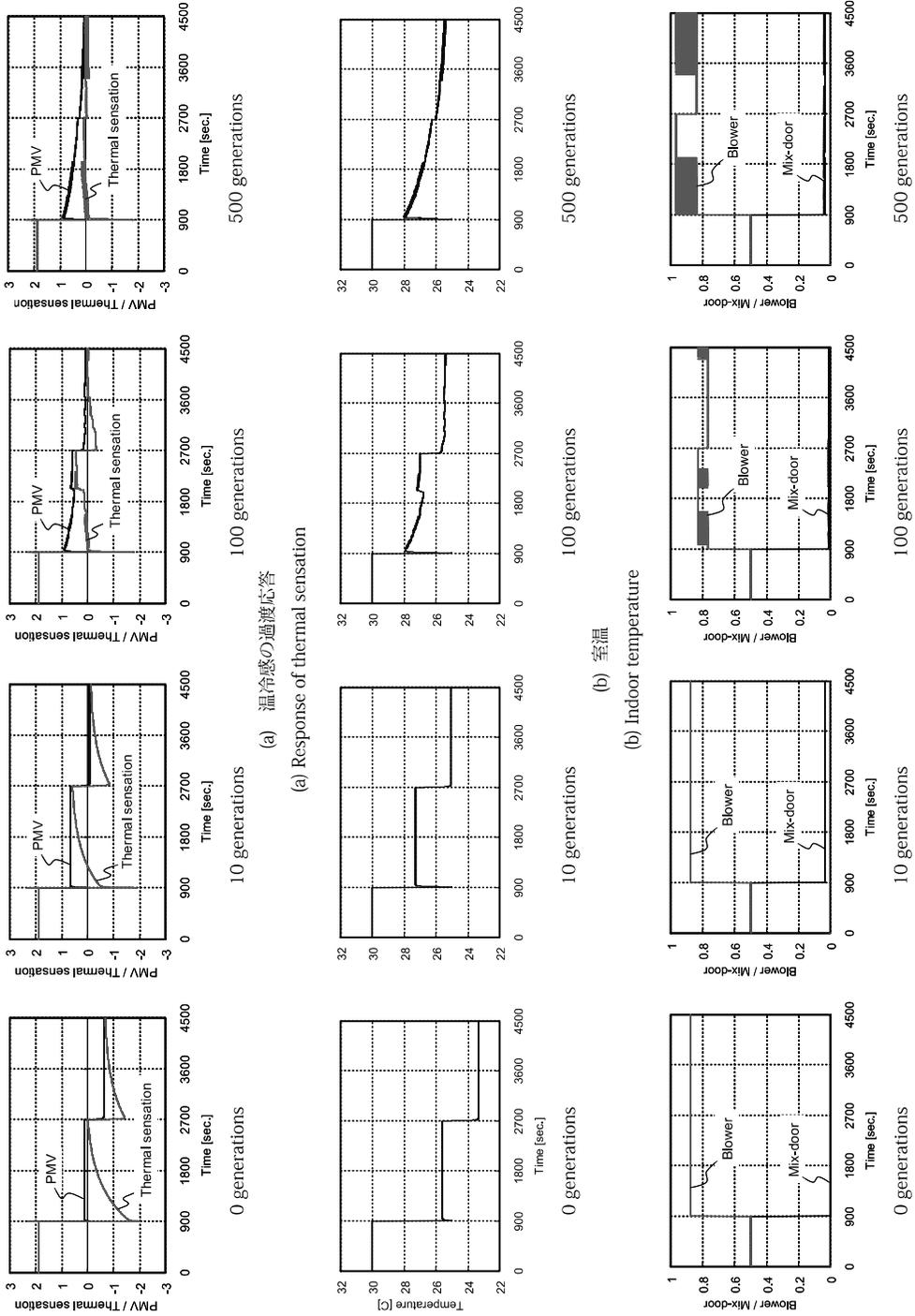


図 16 シミュレーション結果 (条件 2)

6.2 評価関数

評価関数は目標となる温冷感を 0 として、4500 秒間の空調制御シミュレーションを行い、目標値との偏差を積算し、積算値の最小のもののほど適応度が高くなるようにする。本論文では、以下の 2 通りの条件でシミュレーションを行う。

1. 室温 30°C の状態から室温を 25°C にステップ状に変化させるとともに、冷房負荷を考慮して進化ハードウェアで生成された制御器の制御シミュレーションを 3600 秒間行う。このシミュレーションは、2 章の快適性の過渡応答実験に相当する。
2. 上記のシミュレーションにおいて、外乱として冷房負荷の変動を与える。30°C から 25°C にステップ変化後から 1800 秒間は 1100[W]、その後さらに 1800 秒間は 550[W] を与える。

6.3 空調制御シミュレーション

図 15, 図 16 にそれぞれ、前述の 2 つの条件の進化学習シミュレーションの結果を示す。30°C から 25°C への変化のタイミングはシミュレーション開始から 900 秒とするが、900 秒以前のシミュレーションでは、進化学習で得られた制御器の空調制御は無効である。両グラフとも (a) 温冷感の過渡応答、(b) 室温の変化、(c) ブロワとミックスドアの開度の制御状況を示す。図 15 の前節の条件 1 の結果では、0 世代において 900 秒以降は冷房負荷の影響で温度が上昇し 28°C 程度で一定となっており、このため PMV は +1 程度を示している。10 世代、100 世代、500 世代と進化学習の世代が進むにつれて、ブロワモータとミックスドア開度が適切に制御され、500 世代では温冷感が 0 で一定になるように制御されていることが確認できる。ここで、PMV は 0 で一定とはならず、本条件下では PMV が 1 付近から徐々に 0 に収束していく様子が見て取れる。これは、2 章の実験で得られた快適性の過渡応答が考慮されているためである。図 16 は前節の条件 2 の結果で、冷房負荷に変動を与えたものである。900 秒と 2700 秒で冷房負荷が変動するため、0 世代では進化学習がまったく進んでいないために制御がうまく機能せず、900 秒と 2700 秒を境にグラフが大きく変化し不連続であることが確認できる。しかし、条件 1 の場合と同様、500 世代では温冷感を 0 とする制御がなされていることが確認できる。以上より、快適性の過渡応答を考慮した空調制御器が進化学習により自動生成されることが示唆される。また、50 試行分の VHDL を論理合成した結

果、マクロセル使用数は最大 21 個、最小 16 個、平均 17.5 個であり、XC9572 の場合、マクロセル使用率は平均 24.4% であった。本モデルではより小規模の CPLD を使用することも可能であることがわかる。また、他問題への応用の際にセル使用数が XC9572 の上限を超えてしまう場合には、より規模の大きな CPLD を選定すればよい。

7. おわりに

本稿では、進化計算を応用した空調制御器の最適設計手法を提案した。人間の快適性の過渡変化に注目し、室内の快適指標が一定であっても、環境への暴露時間や心理状態により人間の快適感・温冷感に変化することを実験的に明らかにするとともに、快適性の過渡応答モデルについて検討した。さらに、進化計算を応用し、CPLD や FPGA の内部ロジックを記述する VHDL を合目的に自動生成させるための枠組みを提案し、快適性の過渡応答を考慮した空調制御系をターゲットとして、VHDL を自動構成するシミュレーションを行い、本手法の有用性を示した。

参考文献

- [1] Koza, J : Genetic Programming ; MIT Press, (1994).
- [2] 伊庭斉志: 情報科学セミナー 遺伝的プログラミング, 東京電機大学出版局 (1996) .
- [3] 伊庭斉志: 遺伝的プログラミング入門; 東京大学出版会, pp.177 (2001) .
- [4] 小島一恭, 綿貫啓一: 予測快適度に基づく車室内空調制御プログラムの進化的自動構成; 設計工学, Vol. 40, No. 5, pp.255-262(2005).
- [5] 小島一恭, 綿貫啓一: GA による 8 脚歩行ロボットの歩容獲得; 設計工学, Vol.39, No. 1, pp.46-53(2004).
- [6] 薩本弥生: 生活・人間科学シリーズ 快適ライフを科学する; 丸善, pp. 23(2003).
- [7] 中山昭雄: 温熱生理学; 理工学社, pp.69(1981).
- [8] Fanger, P. O. : Thermal Comfort; McGraw-Hill, (1970).
- [9] Goldberg, D. E. : Genetic Algorithms in search, Optimization and Machine Learning ; Addison-Wesley, (1989).
- [10] 三宮信夫, 喜多一, 玉置久, 岩本貴司: 遺伝的アルゴリズムと最適化, 朝倉書店, pp.13 (1998) .
- [11] Xilinx : XC9572 In-System Programmable CPLD Product Specification ; Xilinx, pp.2 (1998).