

FPGA で作成する電子回路

生命・情報等教育支援室（電子・情報工学系） 小野 雅晃

はじめに

私は、先生方より電子回路（主に論理回路）の作成を請け負うことがある。10 数年前までは、電子回路を作成する場合には、標準 TTL(Transistor-Transistor-Logic)を使用し、配線はラッピングで作成していた。その後、電子回路の要求性能も向上し、標準 TTL を使用しては、要求されている性能を満足できない状態になってしまった。高度に回路が集積された専用の IC(Integrated Circuit)を使用すれば性能は向上できるが、そのためには、専用 IC を手に入れないといけない。専用 IC は何万個単位で購入することが出来れば、簡単に購入できるが、1,2 個しか購入できなければ、入手が困難なことが多い。また、自分の目的に完全に合致した部品がないこともある。そこで、もう大学では、ある程度性能の優れた電子回路を作成するのは無理であると考えていた。しかし、5,6 年前から、FPGA が柔軟に論理回路を作成できることを知り、使用してみた。その結果、専用 IC を FPGA(Field Programmable Gate Array)で置き換えが可能であることがわかった。FPGA の利点はプログラムにより、柔軟性のある機能ブロックを配線可能なため、ほとんどの論理機能を実現できることである。よって、FPGA を使用すれば、いろいろな専用 IC を作ることが出来る。また、複数の専用 IC の機能を 1 個に集約することも出来る。専用 IC を使用すると、どうしても使わない機能が出てきてしまう。FPGA を使用して電子回路を作成すると、目的の電子回路に不必要な機能は搭載する必要がない。しかも、FPGA は汎用部品であるので、専用 IC よりも購入することが容易である。小さな規模の FPGA は通信販売でも買うことが出来る。大規模の FPGA は販売代理店で購入することになるが、大体の場合は 1 個からも購入することが出来る。よって、少量多品種生産の私たちのような場合に、メリットが大きい。本ポスターセッションでは、FPGA のプログラム方法を実機によりデモンストレーションする。

以前の電子回路作成

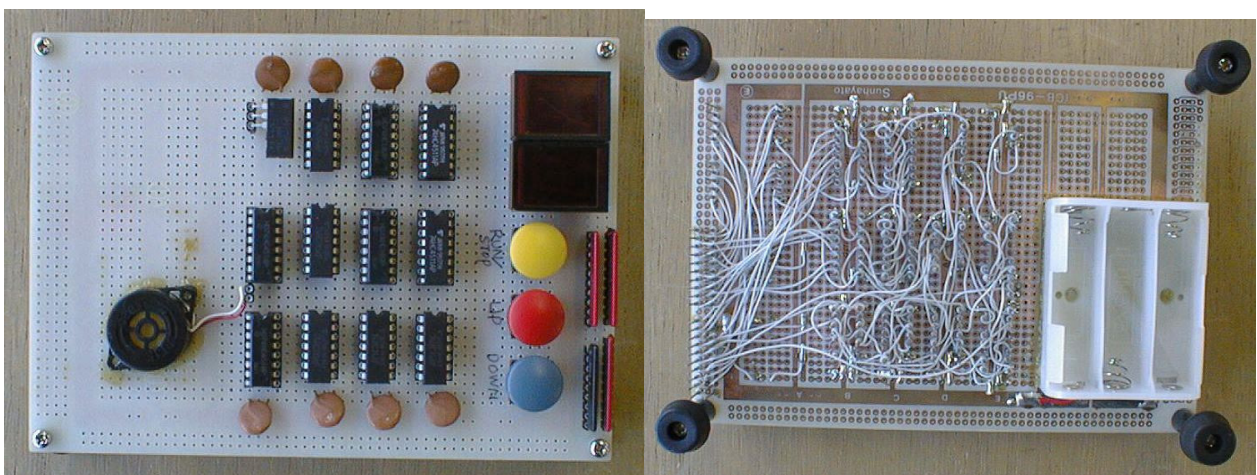


図 1 標準 TTL とラッピングで作成した電子回路(タイマー)表、裏

以前の電子回路は図 1 のように、標準 TTL とラッピングで作成していた。図 1 はタイマーの例である。小さな集積度の標準 TTL の端子間をラッピングで接続することによって回路を構成する。標準 TTL は DIP (Dual In-line Package) だったため、ラッピングソケットに直接挿入できた。この場合は、設計ミスなどによって、回路が動作しない場合には配線をやり直したり、場合によっては標準 TTL を追加したりして、回路を修正しなければならない。

その後、10 数年前には、1 部に Lattices Semiconductor 社の GAL (Generic Array Logic) を使用して、電子回路を作成した。GAL は加法標準形の組み合わせ回路に、レジスタなどが内蔵されたマクロセルを基本回路としている。GAL は EEPROM (Electrically Erasable and Programmable Read Only Memory) で作られていて、一度回路情報を書き込めば、電源を切っても回路情報は残っている。その構造を図 2 に示す。(Lattices Semiconductor 社の GAL16V8 データブックより 1 部分を転載)

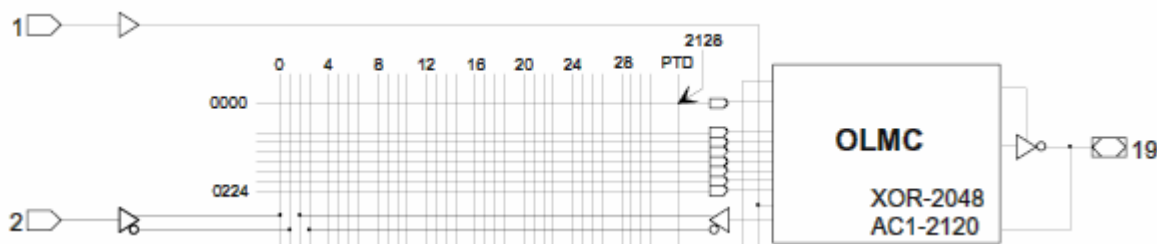


図 2 Lattices Semiconductor 社の GAL16V8 の 1 部分

図 2 で OLMC がマクロセルで、レジスタモードや、各種の組み合わせ回路モードに切り替えることが出来る。その手前の線が交差している回路は組み合わせ回路である。デザインの入力は、ブール式の加法標準形の論理式で記述する。その論理式を PALASM というソフトでコンパイルし、JEDEC (Joint Electron Device Engineering Council) フォーマットで出力する。JEDEC フォーマットのファイルをライターに入力して、GAL に書き込む。GAL を使用したことで、設計ミスが発生した時に配線をつなぎなおさずに、論理式を変更してコンパイルし、GAL に書き込んで修正することが出来た。しかし GAL は小規模なので、あまり大きな回路を構成できないのが、残念なところである。

FPGA, CPLD の内部構造

FPGA と CPLD (Complex Programmable Logic Device) はどちらも回路情報を書き込むことによって、いろいろな機能を実現できる論理 IC である。しかし、構造上の違いや論理規模の違いが存在する。CPLD は GAL を複数内蔵して、GAL のマクロセルの出力を内部配線で接続できるようにしたものである。ほとんどの CPLD は EEPROM セルで出来ているので、電源を切断しても回路の配線を記憶している。CPLD は FPGA に比べて小規模である。図 3 に Altera 社の MAX7000 シリーズ CPLD のアーキテクチャを示す。(Altera 社 MAX7000 シリーズのデータシートより転載)

FPGA は CPLD よりも構成単位の粒度が細かい。1 つの構成単位は 4 入力の LUT (Look Up Table) と 1 つのレジスタで構成されている。その構成単位同士をプログラム可能な配線で接続することにより、回路を構成する。FPGA の例を図 4 に示す。(Altera 社 FLEX10K シリーズのデータシートより転載) 図 4 でロジック・エレメントが 1 つの構成単位になっている。EAB は FPGA に内蔵された RAM、カラム、ロウ・インターコネクトは内部配線、I/O エレメントは入出力ブロックである。

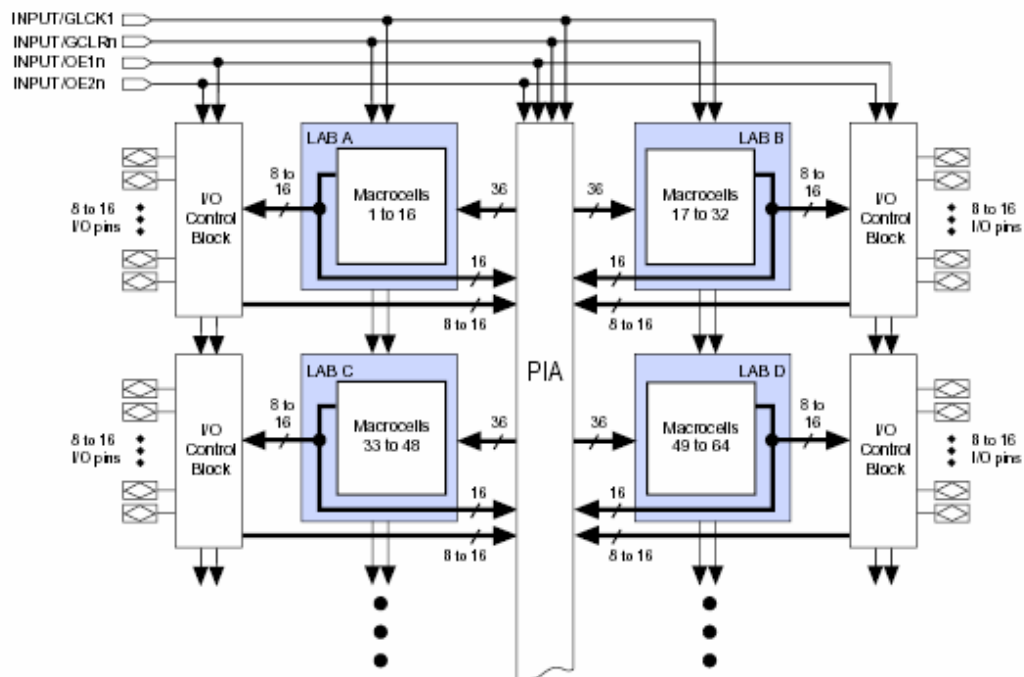


図 3 Altera 社 EPM7032、EPM7032V、EPM7064、EPM7096 のアーキテクチャ

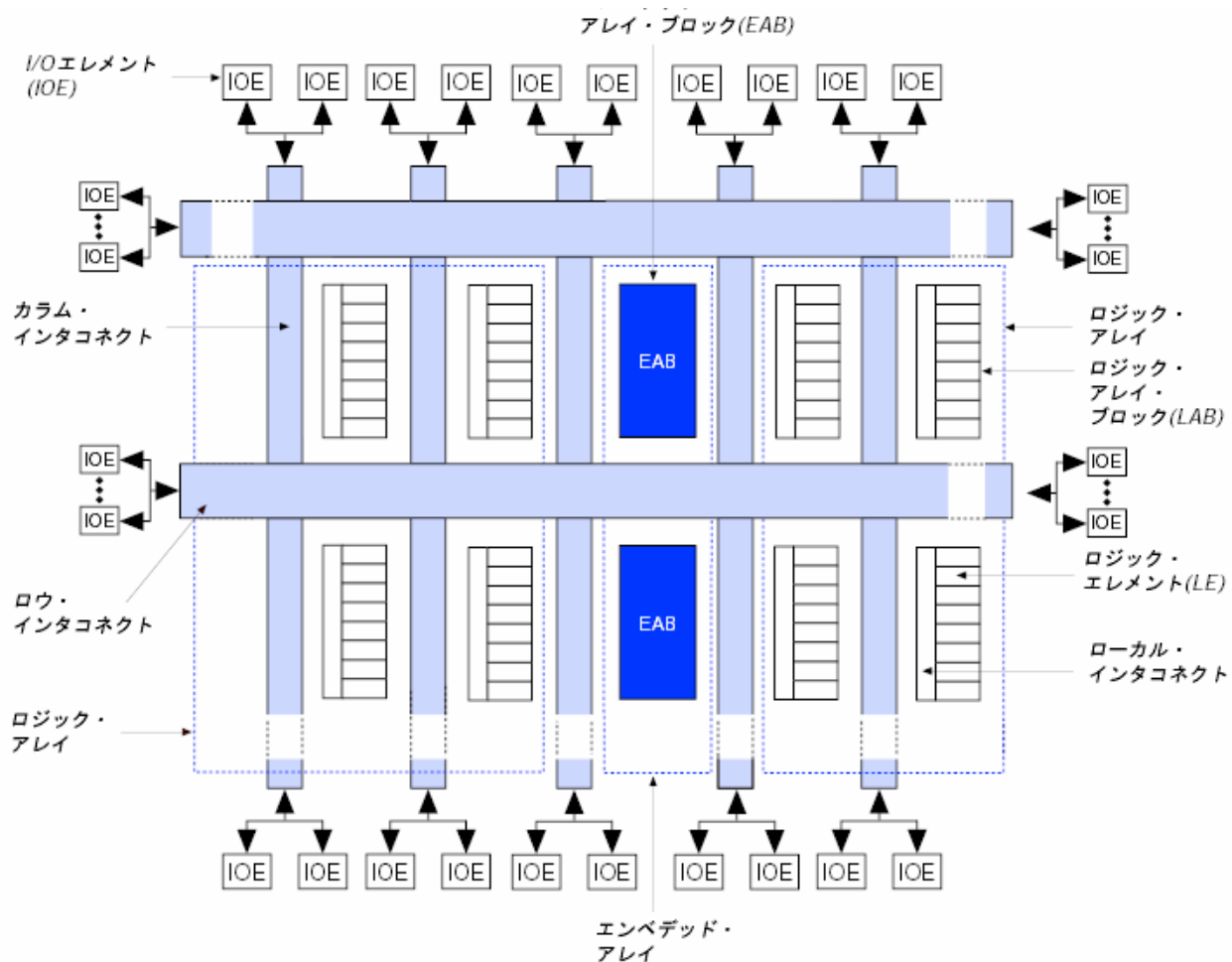


図 4 Altera 社 FLEX10K デバイスブロック図

FPGA は 30 本程度の入力がある CPLD に比べて粒度が小さいので、アドレスビットを比べるアドレスでコードや大きな組み合わせ回路を作るのには向かない。しかし、CPLD よりも一般的に集積度が大きい。FPGA は CPLD より粒度が細かいので無駄が少なくなる。また、FPGA は SRAM 構造が多い、よって、電源 ON 時に ROM などから、回路情報をダウンロードしなければならない。

FPGA, CPLD の開発手順

FPGA, CPLD の開発環境として、専用のソフトウェアが必要となる。FPGA, CPLD の製造会社独自の開発ソフトウェアがあるので、それを活用する。Xilinx 社で言えば Foundation ISE とか Alliance と呼ばれるソフトウェアである。Altera 社で言うところの Quartus II や MAX+PLUS II である。これらのソフトウェアはデザイン入力からコンパイル、そしてデバイスへのプログラミングをすべて 1 つで行うことが出来るようになっている。これらのソフトウェアにはフリーで利用できるバージョンがある。小さな規模の FPGA, CPLD はそれらのフリーソフトを使用すれば開発することができる。

ここで FPGA, CPLD の開発手順を図 5 に示す。

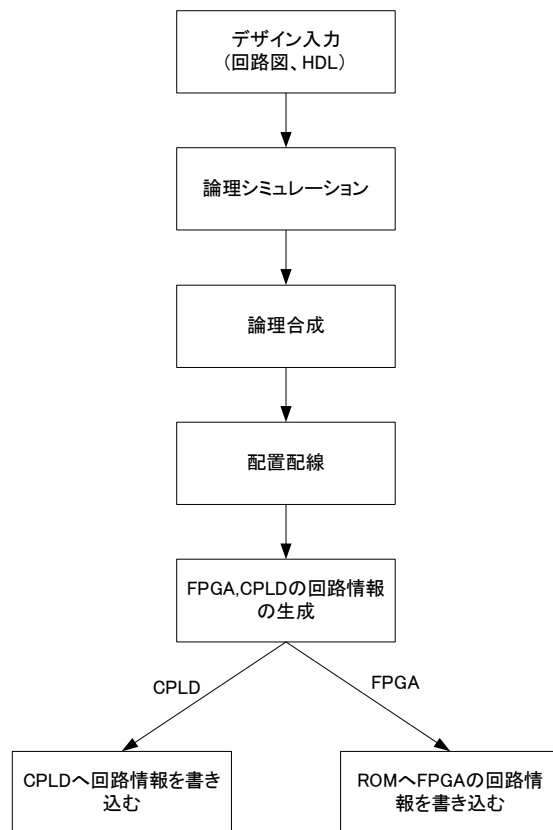


図 5 FPGA, CPLD の開発手順

図 5 において、最初のデザイン入力は、回路のデザインを回路図や HDL などを入力する。次に、入力した回路が本当に動作するかを論理シミュレーションで確認する。動作しない場合は最初に戻って、回路図や HDL を修正する。大きな回路の場合には、いくつかのブロックに分割する。その場合、各ブロックごとにデザイン入力して、論理シミュレーションを行う。すべてのブロックの論理シミュレーションが成功したら、今度はすべてのブロックを接続し、論理シミュレーションを行う。すべての論理シミュレーションが終了したら、今度は論理合成を行う。論理合成は単体の論理合成ツールを用いる場合と FPGA メーカーの開発環境でそのまま論理合成できる場合がある。VHDL (VHSIC Hardware Description Language) や VerilogHDL (Verilog Hardware Description Language) を使用する場合は、単体の論理合成ツールを使用する場合が多い。回路図などでは FPGA メーカーの開発環境でそのまま論理合成する。

その後、FPGA, CPLD の回路構成に合わせて、配置配線を行う。実際にどの LE を使用してどの配線を使うかをここで決定する。配置配線の出来不出来が動作周波数や信号のセットアップタイム、出力信号のクロックからの遅延に影響する事が多い。性能に非常に影響する部分である。ここで所定の性能に達しなければデザインを修正し、最初からやり直すこともある。その後、CPLD は直接チップに回路情報を書き込む。CPLD は EEPROM 構造なのでチップに直接書き込むことが出来る。FPGA は専用の ROM や FLASH ROM へ回路情報を書き込んで使用することが出来る。

FPGA による電子回路作成手順

次に FPGA による電子回路の作成手順を示す。まずは回路規模を概算する。小さい回路規模であるならば CPLD を使用すると良い。CPLD は EEPROM 構造なので、電源を消しても回路情報が消えないので使いやすい。大きい回路規模の場合には FPGA を選択することになる。その場合、電源 ON 時に回路情報をダウンロードするために ROM などの付加回路をつけなければならない。専用の ROM があるので、それが一番使いやすいが、汎用の FLASH ROM 等にダウンロード回路を付加しても大丈夫である。CPLD も FPGA もいろいろな回路規模のチップが用意されている。CPLD ならば数千論理ゲート規模、FPGA ならば数万論理ゲート規模から数百万論理ゲート規模のものまでラインナップされている。

ターゲットが簡単なステートマシンや組み合わせ回路で構成できるならば、FPGA や CPLD だけで構成できる。プロセッサが必要な場合は複数の選択肢がある。8 ビットから 32 ビットまで位のプロセッサが必要でそれほど性能を必要としない場合は、プロセッサの IP (Intellectual Property) を購入するか、またはプロセッサを自分で FPGA 内に作りこむ方法が使用できる。キャッシュや 300MHz 以上の動作周波数のプロセッサが必要な場合は、外部に単体のプロセッサを付けることになる。PCI インターフェースなども FPGA 内に作りこむことが可能である。SDR (Single Data Rate) SDRAM インターフェースは十分に FPGA で構成可能だが、DDR (Double Data Rate) SDRAM も構成可能な FPGA も出てきた。この FPGA は IOB (Input Output Block) に DDR レジスタを内蔵している。

最近の FPGA は複数の入出力レベルをサポートしている。LVTTTL (Low Level Transistor Transistor Logic) から SSTL (Stub Series Terminated Transceiver Logic)、GTL (Gunning Transceiver Logic)、LVDS (Low Level Differential Signaling) までサポートしている FPGA もある。また、プログラマブルなインピーダンス制御回路を内蔵している FPGA もある。(Xilinx 社 Virtex II)

使用するデバイスが決定したら、前述の FPGA, CPLD の開発手順でデザインを考え、入力し、論理合成、配置配線してチップにダウンロードする。その場合に論理合成後に回路の規模がわかるので、出来れば、ここまでチップの規模の選定を待ったほうが良い。

FPGA で電子回路を作成する場合には、市販のボードを購入して、それに付加回路を付けるという方法でも良い。大体の FPGA のトレーニングボードは機能が拡張できるように、使用していないピンをコネクタやプリント基板のパッドに引き出してある。その部分を使用して自分の目的とする回路を構成すればよい。こうすると規模の大きな FPGA を使用しやすくなる。規模の大きな FPGA は一般的に QFP (Quad Flat Package), BGA (Ball Grid Array), FCBGA (Flip Chip Ball Grid Array) 等のパッケージに入っているので、自分で半田付けするのが難しい。また、既製品を利用する場合は、基本的な部分 (FPGA の電源投入時のコンフィギュレーション回路など) は動作することが保障されているので、安心感がある。市販の FPGA ボードは CQ 出版社、ヒューマンデータ、エスケーエレクトロニクス、三菱電機マイコン機器ソフトウェアなどで製品化されている。

自分で基板を起こす必要があるならば、PCBE などの基板 CAD ツールを使用して基板のレイアウトを設計して、基板作製メーカーに作製してもらう方法もある。PCBE はフリーのプリント基板 CAD ツールで Vector などからダウンロードできる。その他、Eagle など、制限はあるがフリーのプリント基板 CAD ツールがある。また、かなり困難ではあるがラッピングでも基板を作製できる。ラッピングで作製した基板の例を図 6 に示す。

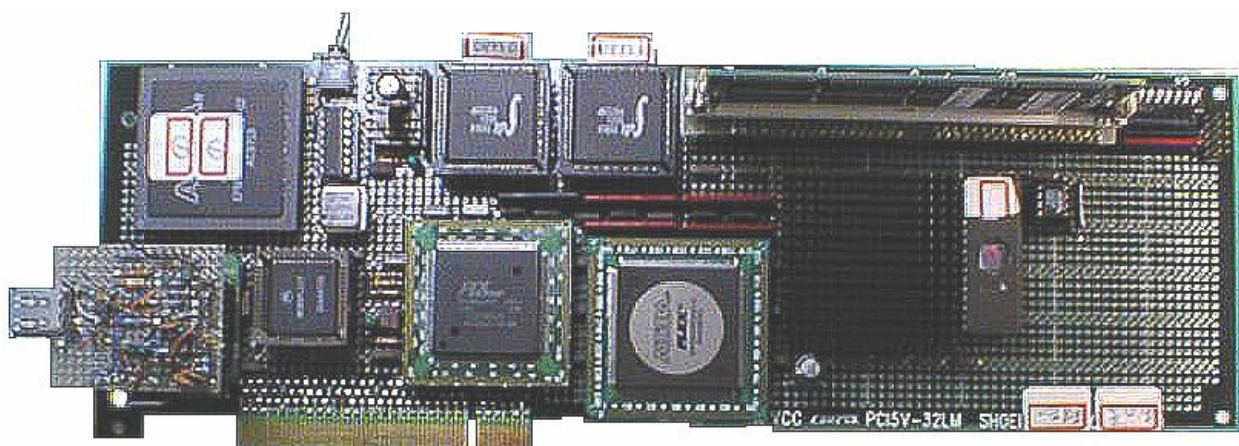


図 6 ラッピングで作製した基板の例 (PCI ボード)

この基板では、QFP パッケージを PGA ピンに変換する変換基板を使用している。このような変換基板はサンハヤト、昭英電機などが扱っている。図 6 の基板でも変換基板により、QFP パッケージを 2.54mm ピッチのピンに変換して、ユニバーサルボードに取り付けている。これらのメーカーに QFP パッケージなどの半田付けを頼めばやってもらうことも出来る。PLCC パッケージはラッピング用のソケットが市販されているので、そのソケットを使用してユニバーサルボードに取り付ける。このように表面実装用のパッケージでもユニバーサルボードに取り付けることが出来る。ただし、240 ピン等の QFP パッケージをピンに直すと基板の裏側はまるで剣山ようになってしまう。そのため、ラッピングでもワイヤをつなぎなおすということが極めて困難となってしまふ。通常、ラッピングは、間違えてワイヤを接続しても、もう一度ワイヤをはずして、正しい配線に接続しなおすのが容易という利点がある。しかし、この場合には、ワイヤの下に何本ものワイヤが埋もれて、前に接続したワイヤを掘り出すのが非常に難しい。

ポスターセッション時のデモンストレーション

ポスターセッション時には、情報学類 3 年実験に使用している FPGA 搭載のメデューサボードを使用してデモンストレーションを行う。今回のデモンストレーションに使用するメデューサボードは Altera 社の FLEX10K30 を搭載した市販のテスト用ボードである。メデューサボードを図 7 に示す。

情報学類 3 年生の実験では、このトレーニングボードを使用して、FPGA 設計の実験を行っている。今回はその課題の中から、野球ゲームと MIPS プロセッサをデモンストレーションする。

どちらも、デザイン開発環境ソフトウェアは Altera 社の MAX+PULS II を使用している。野球ゲームはメデューサボードの 7 セグメント LED を使用して、2 チームが対戦する形式のゲームを FPGA 内に回路を組んで作りこむ。野球ゲームのデザイン入力回路図と AHDL である。MIPS プロセッサは、MIPS プロセッサのサブセットを FPGA 内に作りこんでいる。FPGA 内に制御回路と RS-232C のインターフェース回路を内蔵し、パーソナル・コンピュータからシリアル・インターフェース経由で制御が可能である。MIPS プロセッサシステムは、中山技官とともに開発し、平成 12 年度の技術報告に発表済みである。MIPS プロセッサシステムのデザイン入力回路図は AHDL のみである。

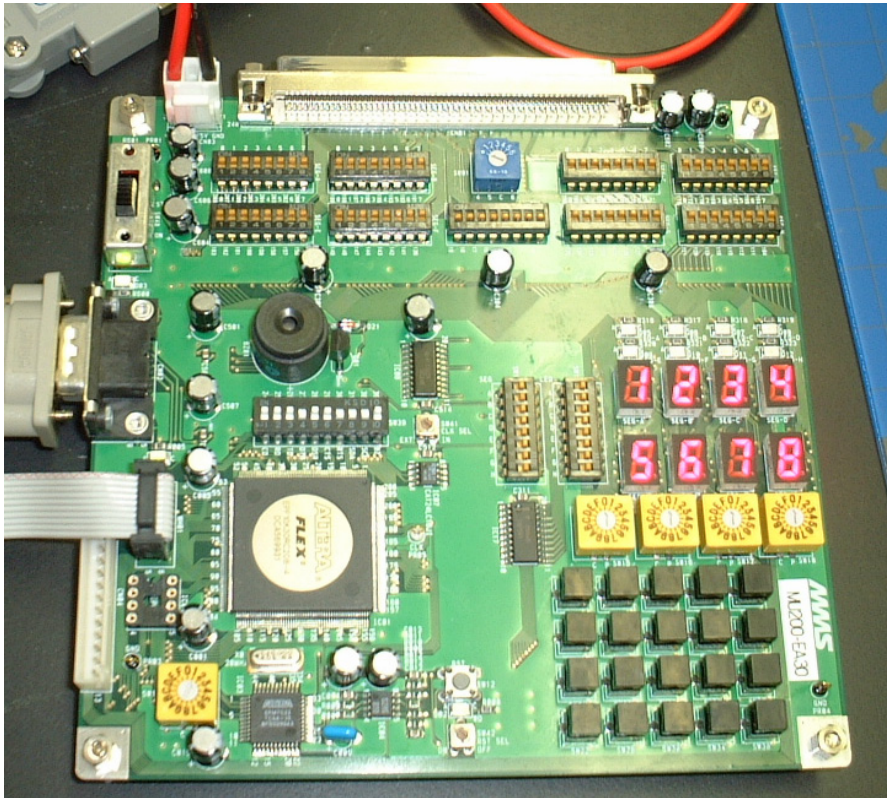


図 7 メデューサボード

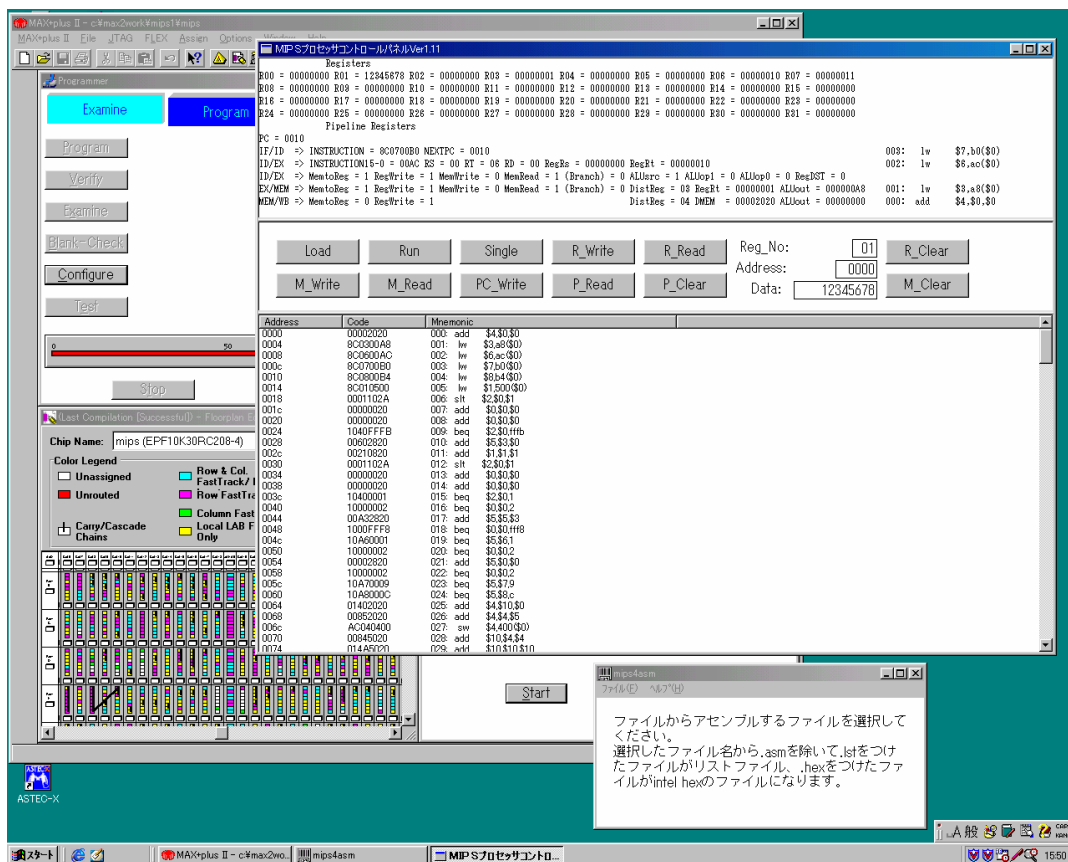


図 8 MIPS プロセッサ実験用ソフトウェア群

図 8 に MIPS プロセッサシステムのデザイン開発環境ソフトウェア MAX+PULS II と MIPS プロセッサを PC からコントロールするためのソフトウェア MIPS プロセッサコントロールパネル、そして MIPS アセンブラ・ソフトウェアの画面を示す。

デモンストレーションでは、デザイン開発環境ソフトウェア MAX+PULS II の使用方法の説明、実際にトレーニングボード上の FPGA チップにダウンロード、動作の展示を行う。

最後に

ここまで、電子回路工作で FPGA を使用方法、注意点を述べた。FPGA は、大学で電子回路を作成する上で主要な選択肢になると思う。VDEC や民間のメーカーでカスタムまたはゲートアレイで IC を作ってもらうことがあるかもしれない。その場合でも FPGA は専用 IC チップを作る試作段階で、動作確認用として使用することが出来る。これを機会に、実際に業務で FPGA を使用する人が増えることを願っている。

引用文献、参考文献、参考ホームページ

- [1] Lattice Semiconductor Corporation : GAL16V8 Data Sheet (図 2 を引用)
- [2] 日本アルテラ株式会社 : MAX7000 プログラマブル・ロジック・ファミリ データシート : 1998 年 2 月 ver. 5.01 (図 3 を引用)
- [3] 日本アルテラ株式会社 : FLEX10K エンベデッド・プログラマブル・ロジック・ファミリ データシート : 1998 年 1 月 ver. 3 (図 4 を引用)
- [4] 小野 雅晃、中山 勝 : MIPS プロセッサ実験システムの開発 : 筑波大学 技術報告 No. 21 p33~40 2001 年 3 月
- [5] Xilinx Inc. : Virtex II 1.5V フィールド プログラマブル ゲート アレイ データシート : DS031(v1.1) 2000 年 12 月 6 日
- [6] Xilinx Inc. ホームページ : <http://www.xilinx.co.jp/>
- [7] 日本アルテラ株式会社ホームページ : <http://www.altera.co.jp/>
- [8] Vector ホームページ : <http://www.vector.co.jp/>
- [9] Cad Soft ホームページ : <http://www.cadsoft.de/> (Eagle のフリーバージョンがあります。)
- [10] サンハヤトホームページ : <http://www.sunhayato.co.jp/>
- [11] 昭英電機ホームページ : <http://www.shoeidenki.co.jp/>
- [12] CQ 出版社ホームページ : <http://www.cqpub.co.jp/>
- [13] ヒューマンデータ・ホームページ : <http://www.hdl.co.jp/home.html>
- [14] エスケーエレクトロニクス・ホームページ : <http://www.sk-el.co.jp/>
- [15] 三菱電機マイコン機器ソフトウェア・ホームページ : <http://www.mms.co.jp/>