CPU、コンパイラから、ライブラリまで、自分たちの手でコンピュータを作れ！

CPU実験
―― ほんとうのコンピュータ自作

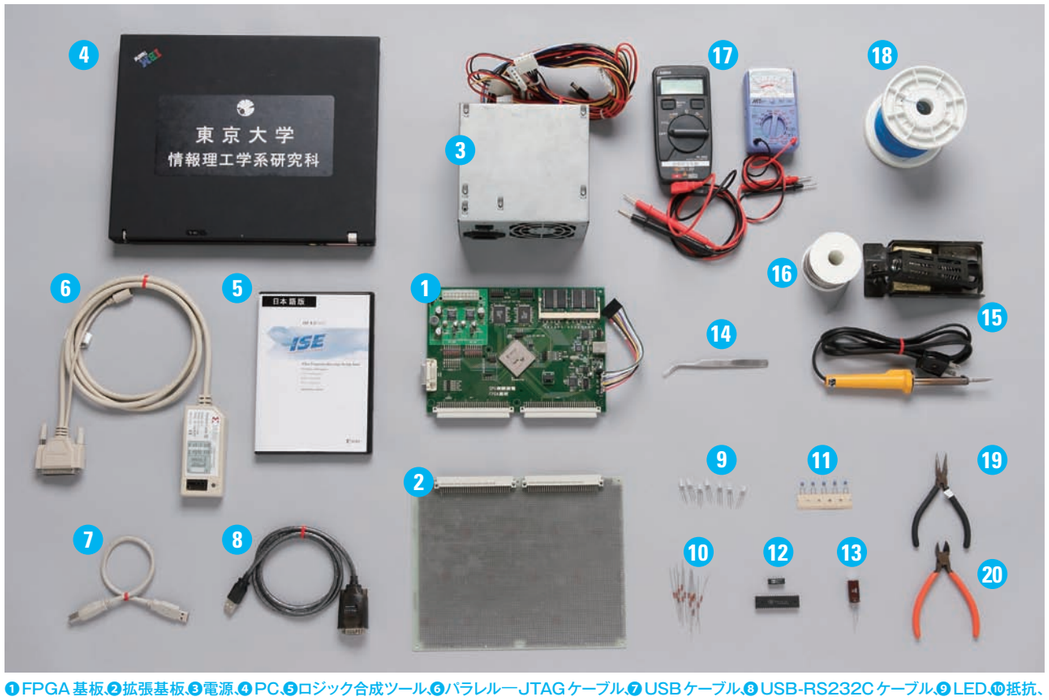
3年生の冬学期になると、情報科学科の名物、情報科学実験II――通称「CPU実験」が始まります。CPUの設計段階からコンピュータを作り、課題プログラムが動くようにするというハードなものですが、OB・OGの誰もが「楽しかった」と口を揃えます。このエキサイティングな実験の様子を紹介します。

10月のはじめ、4～6人に分けられた各チームに、FPGA基板と道具がいくつか渡されます。ミッションは「半年かけてできるだけ速いコンピュータを作れ」。それから翌年3月に開かれる発表会までのあいだに、与えられた課題プログラム（例年はCGプログラム）が動くように独自のコンピュータを設計・製作します。

実験は、まず命令セット/アーキテクチャの設計から始め、CPU、コンパイラ、アセンブラやシミュレータを分担して実装、というふうに進みます。学生実験としてはかなり難しく、動作に至らないこともありますが、いったん動きだすと、発表会でのスピードコンテストを目指してさらに高速化に工夫を凝らしていきます。

高速化のノウハウは先輩から後輩へと伝えられ、先輩は後輩たちがさらに速いものを作るのを楽しみに待ちます。

またCPU実験は、一連の課程を通してコンピュータの根底にある動作原理を体得できるだけでなく、半年にわたるプロジェクトワークがたいへん貴重な経験になります。



❶FPGA基板、❷拡張基板、❸電源、❹PC、❺ロジック合成ツール、❻パラレル―JTAGケーブル、❼USBケーブル、❽USB-RS232Cケーブル、❾LED、❿抵抗、⓫フィルムコンデンサ、⓬チップ、⓭電解コンデンサ、⓮ピンセット、⓯ハンダごてとハンダごて置き、⓰ハンダ、⓱テスタ、⓲導線、⓳ラジオペンチ、⓴ニッパー

1 全体的な設計

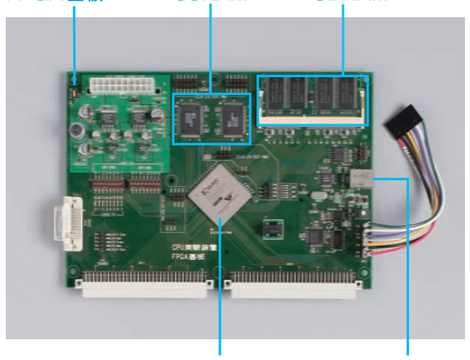
最初に、CPUの全体的な設計を行います。この段階では、まず命令セット――CPUが備える命令群などのアーキテクチャを決めます。複雑な仕様にすると完成させるのが難しくなるので、最初は既存のCPUアーキテクチャを参考に、シンプルな設計から始めることが多いようです。性能のよいコンピュータを作ろうとするとき、CPUの実装のしやすさと、コンパイラの開発のしやすさが、相容れない場合もあります。うまくそのバランスをとることがたいせつです。

スピードコンテストでの記録更新を狙うようなチームは、何度か設計しなおし、シンプルで周波数が高いもの、独特な命令や複雑な機構を採用したもの、というふうに既存の枠に囚われないアーキテクチャも登場します。

仕様が固まると、各自の興味・得意不得意を考慮に入れて開発の分担を決め、それぞれ開発にとりかかります。分担した作業が進むあいだ、仕様の解釈に違いがないように確認し、進捗状況を確かめあって必要であれば手伝い、途中で気づいた問題点をうまく回避して各自の開発物に反映させます。

2 FPGA基板について

CPUの実装には、支給されたFPGA基板を使用します。基板の中央には、内部の回路を自由にデザインして書き換えられるFPGAという半導体チップが載っています。FPGAの周りには、メモリ（SSRAM、SDRAM）、入出力端子（DVI、USB）、回路書き込み用のケーブルなどの周辺装置が備えられ、FPGAに制御回路を用意してやると利用可能になります。拡張基板には、作業用コンピュータと通信するための追加の入出力端子（RS-232C）、電圧を変換するチップ、開発初期のFPGAの動作確認に使うLED、抵抗などを、必要に応じてハンダ付けしていきます。写真は現在使用しているFPGA基板で、中央のFPGAは100万ゲート規模の回路を実装可能です。最新のFPGAを搭載した新しい基板も準備中なので、もっと多様な回路が実装可能になるでしょう。



Column コンピュータの動作原理とアーキテクチャ

「コンピュータの心臓はCPUだ」「書いたプログラムはCPUが計算してくれる」ということを聞いたことはあっても、実際にCPUがどうやってプログラムを実行しているかは知らない方も多いでしょう。

コンピュータは「0」と「1」が並んだ機械語の命令列しか理解できません。そこで、プログラミング言語で書かれたプログラムは、コンパイラというプログラムで機械語に変換します。コンパイラによっては、いったんアセンブリ言語（機械語をより読みやすい記号で記述できるようにしたもの）のプログラムに変換し、次にアセンブラというプログラムで機械語に変換するものもあります。

プログラムが書き手によって変わるように、コンパイラの変換結果もコンパイラによって差異があり、コンパイラの最適化しだいで、変換後のプログラムの実行効率は大きく変わります。プログラムが巨大化している今日では、特別な分野を除いてアセンブリ言語でプログラミングすることは少なくなり、いかに高性能なコンパイラを作るかが重要になっています。

CPUは、メモリ上に保存された機械語の命令列を順次読みだして実行することにより、プログラムを動作させます。メモリからCPUへの転送速度や数値計算の速度などは、半導体の製造技術だけでなく、メモリの大きさや配置、機械語の命令形式や種類、複数の命令を同時に実行する仕組み、性能の引き出しやすさといった設計が大きく影響します。スーパーコンピュータから携帯電話まで、コンピュータも用途が違えばかけられるコストも半導体技術も異なり、それに応じた設計が求められます。このようなことを扱うのがコンピュータ・アーキテクチャの分野で、基本設計のことを一般にアーキテクチャと呼んでいます。

3 製作開始

▶▶▶ CPU の開発

回路の規模が大きくなった現在、論理回路の実装には、回路図上でゲートを配線する代わりに、HDL（ハードウェア記述言語）を用いるのが主流です。HDLはプログラミング言語に似ていて、回路の動作を詳細に記述できます。このHDLからCADが実際の回路を生成しますが、思い通りの回路を得られるように、CADの動作を見越してHDLを書くのも腕の見せどころです。通常、生成された回路はCADがFPGA内の資源に自動配置・配線しますが、必ずしも最適とはいえないので、手動で配置・配線するツワモノも現れます。設計した論理回路は、HDLシミュレータで表示される波形図で検証します。シミュレーションには時間がかかりますが、この作業を丁寧に行うことが完成への近道です。

HDLから回路を生成、HDLシミュレータで出力信号をチェックする。

▶▶▶コンパイラの開発

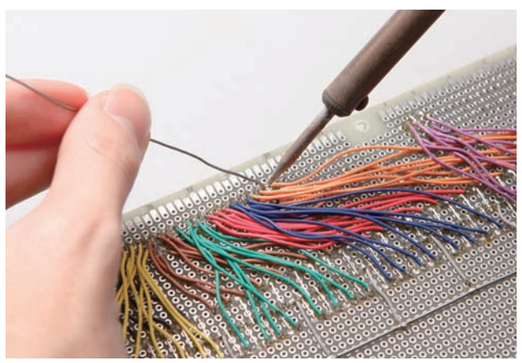
課題プログラムはMLというプログラミング言語で書かれているので、MLコンパイラが必要です。最近はMLで実装されたきれいでわかりやすいMLコンパイラがあるので、これを改造してまず動作させ、より効率の良い命令列を生成するよう最適化することが多いようです。一方、好きな言語でゼロから実装する人も例年います。最初はよく知られている最適化手法を調べて実装したりしますが、自分たちのCPUに特化した最適化には試行錯誤で独自の方法を考えなくてはいけません。三角関数のような一般的な関数ライブラリも、コンパイラとともに用意します。高速化のためにアセンブリ言語で書く人が多いのですが、コンパイラの最適化を頑張った結果、MLで記述したほうが速くなった人もいます。

シミュレーションなど、計算量の大きい作業には情報科学科のクラスタシステムなどを使う。

▶▶▶ツールの開発

CAD、HDLシミュレータ以外の開発ツールも自作します。代表的なものはCPUのシミュレータで、コンパイラが生成した命令列の検証や、実行時間を予測してアーキテクチャを改良するために使います。シミュレータがなければコンパイラを開発できないので、まず簡単なものを実装し、必要に応じて改良していきます。命令の取捨選択のために各命令の呼び出し回数の統計をとったり、アーキテクチャの改良のためにパラメータを変えて実行する機能が加わることもあります。様々な条件でまとめてデータをとるときには、性能の高い情報科学科のクラスタシステムなどを利用します。

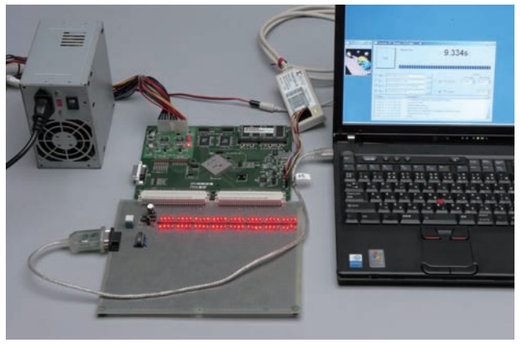
以上の作業と並行して拡張基板に追加端子やLEDをハンダ付けし、FPGAボードでのテストの準備も進みます。



拡張基板に追加の端子やLEDをハンダ付け。製作者の好みがでるところ。

4 ただいまテスト中

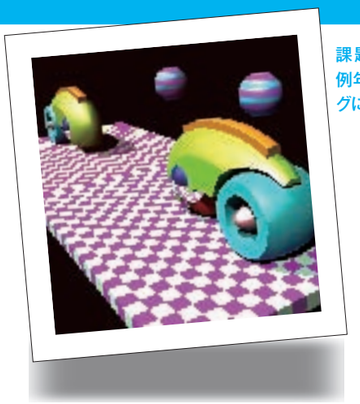
ひととおり出来上がると、実際にFPGAに回路のデータを送り込んでテストします。それぞれは一見うまく動いているようでも、不具合や仕様の解釈のずれは残っているもので、ここから完成までは思いのほか長くかかります。あまり楽しくない作業ですが、その代わり検証方法が身に付きます。PCとの通信はほんとうにうまくいっているか、機械語が適切に生成されているか、CPUは仕様どおりかなど、すべての可能性を疑って問題を探ります。ハードウェアに不具合がある可能性もあり、出力される信号をオシロスコープで調べることもあります。発熱が原因ではないかと疑ってヒートシンクをFPGAの上に載せてみたらうまく動いた、なんていうこともありました。



FPGAに回路のデータを送り込んでテストする。

5 動いた！

ついに完動する日がきます。課題プログラムのCGも正確に描画されて、これでおしまい……ではなく、実はスピードコンテストに向けたここからの高速化が本番です。CPU実験の最も楽しいところです。過去の例では、パイプライン、レジスタフォワーディング、VLIW、スーパースカラ、キャッシュ、プリフェッチ、分岐予測、スクラッチパッドメモリなどが導入されました（興味のある人はぜひWebなどで調べてみてください）。課題プログラムを徹底的に解析してコンパイラを最適化したり、それに飽きたらず、生成したアセンブリプログラムをさらに手作業で最適化する人もいます。自分たちのコンピュータがどんどん速くなっていくのはとても気分の良いものです。



課題プログラムは例年レイトレーシングによるCG。

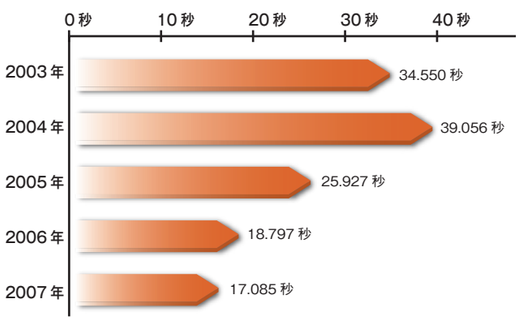
6 完成

発表会には例年、院生なども大勢訪れます。各チームが半年かけて作った自分たちのアーキテクチャやコンパイラを自慢し、無事に完成したチームは課題プログラムを実演してスピードを測定します。

年々記録が更新されていく様子が見てとれる右のグラフには、先輩から継承されたCPU実験のノウハウが形となって表れています。数年前の「夢の20秒」というフレーズも、いまではすっかり現実のものになりました。

そして、発表会が終わっても、なぜか挑戦は続きます（だって、そこにボードがあるから）。2008年にはついに、1年間の改良を経て、10秒を切ってしまった人も現れました。そこまで熱中できるのがCPU実験なのです。

（谷田直輝）



課題プログラムの実行時間の推移。2006年には夢の20秒突破を果たした。