

16bit CPU コンテスト課題

概要

CPU とコンパイラはデジタルシステムの基本技術であり、これらの基本的な動作を理解することはデジタルシステム設計者としてのスタートアップとして必須といっても過言ではない。第 11 回 ASIC デザインコンテストでは、エンジニアを目指す学生諸君が容易に取り組める初級コンテスト課題として 16 ビット CPU を提示する。

この課題は初級コンテストであるため、対象を大学院修士（博士前期）課程以下の学生に限定する。

この課題で要求される技術は下記のものとなる。

- ソースコードで定義される課題プログラムを実行する CPU の開発
- 同プログラムをコンパイル可能なコンパイラ開発
- 論理シミュレーションによる動作検証
- 時間・エネルギー積を最適化するハードウェア・コンパイラ改良

プログラムの実行サイクル数と実行結果の検証は論理シミュレーションによって行い、エネルギー・実行時間は PARTHENON (ライブラリは DEMO 社の demo ライブラリ) による論理合成結果の消費電力・最大遅延時間を用いて算出する。実行サイクル数はシミュレーションスタート時点から停止命令実行までのクロック数でカウントする。

論理シミュレーションツールに関しては規定しないが、課題ソースコードをコンパイルし、論理シミュレーションを実行するまでの一連の流れをコマンドラインから make コマンド等一括実行可能なこと。(審査用非公開課題プログラム (foo.sc) のシミュレーション実施方法も明示すること) また、プログラムの実行結果を検証するため、論理シミュレーションにおいて、適切なデータ表示がなされていること。

コンテストチャレンジャは、審査用設計ドキュメントの他に、CPU の SFL ファイル、コンパイラ・アセンブラのソースファイル、Makefile を電子ファイルにて提出し、シミュレーション・論理合成に必要な環境を審査用ドキュメント中に明記すること。(審査において再現性がないと判断したときは、採点ポイントを大きく減点する)

課題プログラムの実行時間・消費エネルギー積、アーキテクチャ上の工夫、ドキュメントと設計の整合性などを審査対象物とする。

1 課題プログラム

二つのプログラム (バブルソート・再帰呼び出し) をコンテスト課題プログラムとする。課題プログラムのソースコードは変更してはならない。コンテストは二つの課題プログラムと、審査委員が用意する非公開のプログラム計 3 本のプログラムの実行時間・消費エネルギー積の相乗平均値で評価し、この値が小さいほど優秀な設計とする。

コンテスト参加者は提示されたソースコードをコンパイル可能なコンパイラを作成し、コンパイラが出力する機械語を論理シミュレーションに用いる。

初級課題であることを考慮し、コンパイラ、アセンブラを含む例題パッケージを用意した。例題パッケージは CPU (SN/X) の論理ファイルとシミュレーション環境、およびアセンブラ、コンパイラのソースコード群から構成され、IP ARCH 社のホームページ [1] からダウンロード可能とする。例題のプロセッサアーキテクチャおよびコンパイラの詳細は文献 [2] を参照のこと。

例題コンパイラ・アセンブラは yacc と lex で文法を記述し、C でコード生成を記述したもので、ターゲット言語は C 言語の簡単なサブセットである。

コンパイラの言語仕様を次にまとめる。

CPU アーキテクチャ

- 命令・データのアドレス空間を分離したハーバードアーキテクチャー
- 命令・データとも 1 ワード (16 ビット) 1 アドレスのワードマシン

変数 メモリ上の 1 番地から順に格納される大域変数の a~z までと、関数引数の arg だけが許される。変数には添字を付けることができ、a[i] は a から i 番地上のメモリ上の変数を表す。

関数 関数は整数型か void 型の foo という関数名一つだけが定義でき、局所変数はない。引数は arg という名前の仮引数を 1 つだけ利用可能。

演算 + - < > >= <= == ++ -- が利用可能。(++ --は後置のみ)

構文 while if else for return halt が利用可能。halt 実行時にはプロセッサを停止させ、シミュレーションをストップすること。

課題 1 : 再帰呼び出し

```
a=foo(5);
halt;
int foo(int arg) {
    if(arg < 1) return(arg);
    return arg + foo(arg-1);
}
```

図 1: 再帰関数課題のソースコード

課題 2 : バブルソート

```
n=20;
for(i=0; i<n; i++) z[i]=n-i;
for(i=0; i<n-1; i++)
    for(j=n-1; j>i; j--)
        if(z[j]<z[j-1]) {
            w=z[j];
            z[j]=z[j-1];
            z[j-1]=w;
        }
halt;
```

図 2: バブルソート課題のソースコード

2 コンテスト審査数値の計算手順

1. 合成した結果の最大遅延時間でプロセッサを動作させた時の各プログラムの実行時間を求める。¹
2. 最高クロック周波数で動作する時の各消費電力と実行時間から各プログラムの実行に必要なエネルギーを求める。
3. 実行時間ならびに実行エネルギーの相乗平均を求める。

各プログラムの実行クロック数、合成結果が表 1、図 3 のようになっていたとする。最大立上がり遅延時間が 63.6nS とすると実行時間・エネルギー積の計算は次のようになる。

表 1: 課題各プログラムの実行クロック数例

課題	クロック数
for	156
recur	8402
sort	26216

```
--- statistics summary -----
position = /
type = NLD
class_name = snx
power = 5680.1
area = 810.06
gates = 2995
```

図 3: SN/X の論理合成結果

クロック周期は 63.6nS であるため、実行時間 T の相乗平均値は

$$T = \sqrt[3]{156 \times 8402 \times 26216} \times 63.6 \times 10^{-9} = 0.21mS$$

実行エネルギーの相乗平均値は、消費電力 P が $5680.1\mu W/MHz$ なので、

$$T \times P = 0.21 \times 10^{-3} \times 5680.1 \times 10^{-6} / (63.6 \times 10^{-9} \times 10^6) = 18.5\mu J$$

したがって、エネルギー・時間積は

$$0.21 \times 10^{-3} \times 18.5 \times 10^{-6} = 3.9nJ \cdot S$$

となる

参考文献

- [1] アイピー・アーク・インコーポレイテッド <http://www.ip-arch.jp/>
- [2] 清水, "コンピュータ設計の基礎知識," 共立出版, 2003

¹ここでは仮に審査用プログラムとして、「for」を加え、3つのプログラムで計算する