浮動小数点変数の固定小数点化ツール

FP-Fixer Ver.3のご紹介

2007 年度 ***LSI オブ ザ イヤー*** **優秀賞**受賞製品
2007 年度 ***東京都ベンチャー技術大賞*** **優秀賞**受賞製品
2008 年度 ***中小企業優秀新技術・新製品賞*** **奨励賞**受賞製品
2009 年度 ***東京都トライアル発注認定製品***

株式会社礎デザインオートメーション
http://www.ishizue-da.co.jp

現状のFP-Fixerの使われ方

コード全体の固定小数点化は困難

個々の関数毎の固定小数点化に使用

現状のFP-Fixerの問題１

- 入力テストデータの規模
  - 動画などは膨大なフレーム数で検証が必要で、数時間～数日かかる。
- 解析エラーの対策
  - 解析エラー変数は検出するがコード上のどの位置の演算でエラーか分からない。
- 高精度な解析の対応
  - 必ず整数部１ビット以上が必要で、小数部のみから成る高精度解析ができない。
- C言語の記述制約
  - キャストが動的メモリ等の扱いのルールは記述してみないと分からない

現状のFP-Fixerの問題２

- 既存のコード利用
  - 既に固定小数点化済みの関数の使用は困難
  - 浮動小数点変数のある関数の使用はできない
- テストデータの分割による解析
- 64ビットマシーン対応
- GUI

礎DA
ISHIZUE Design Automation Corp.

主な強化機能

- ●大幅な解析時間の高速化の実現
  - ●入力テストデータの圧縮
  - ●限定ビット幅指定により大幅なビット幅高速解析
- ●解析精度の改善
  - ●小数部のみのデータの解析
  - ●浮動小数点変数と固定小数点変数の混在解析
- ●解析エラーおよびC言語記述制約の対策
  - ●エラーが発生した演算位置の解析エラーレポート
  - ●Cルールチェッカと記述例のフリーダウンロード
    - ●2013年2月より弊社webより

礎デザインオートメーション

礎DA ISHIZUE Design Automation Corp. 大幅な解析時間の高速化の実現

- 入力テストデータの圧縮
  - 各変数のビット幅に影響するデータを抽出
  - 丸めを考慮したテストデータ解析アルゴリズムを開発
    - 単なる最大値、最小値データの抽出とは異なる
  - 圧縮率はデータの規模が多ければ多いほど効果的
    - ある動画処理の3種類のデータでは、1/3、1/5、1/8に
    - ある静止画データは、1/100に

- 限定ビット幅指定による解析
  - 変数のビット幅は必ずしもギリギリまで解析は不要
  - 解析したいビット幅のグループ指定が可能
  - 例えば、4,8,12,16,20,24,28,32ビットでおよそ1/10に改善

礎デザインオートメーション

解析精度の改善

- 小数部のみのデータの解析
  - 小数部の実際の値を仮数として表現可能に
  - 以下のディレクティブもサポート
    - // fpfix fixedP U 5:7 i.e
    - 0.00nnnnn　：仮数部の開始が小数点以降2桁後に仮数値を開始。

- 浮動小数点変数と固定小数点変数の混在解析
  - 浮動小数点のまま解析したい変数には以下のディレクティブの指定が可能
    - // fpfix float

礎DA ISHIZUE Design Automation Corp. FP-Fixer Ver.3改善項目一覧

No.	改善項目
1	解析データ圧縮コマンドの追加
2	ビット限定精度解析の追加
3	解析エラー(ビット不足)のエラー表示強化
4	高精度実数演算のビット幅解析処理追加
5	浮動小数点と固定小数点の混在サポート1 (浮動小数点変数の混在)
6	浮動小数点と固定小数点の混在サポート2 (浮動小数点関数の混在)
7	ブロック(ループ)単位の解析(シミュレーション)を追加
8	積和変数を明示的に指定するディレクティブ追加
9	自動発生中間変数のプレフィクス文字列の指定オプション追加
10	解析速度向上のためにDSP簡易解析モードの割算解析変更
11	16ビットCPU対応の関数ライブラリの追加
12	その他

礎デザインオートメーション

礎DA ISHIZUE Design Automation Corp. 高精度実数演算のビット幅解析処理追加1

■概要

高精度実数演算のビット幅の解析処理は、AlgorithmicC（-all_algoc）のみの対応となります。
高精度実数演算対象の変数がある場合、デフォルトで、高精度実数演算を行います。
高精度実数演算を行わない場合は、「-En」オプションを指定してください。

この機能は、以下の2つの機能をもちます。

- 小数点以下の0（ゼロ）が多い極小値
- 整数部が大きい場合

礎DA ISHIZUE Design Automation Corp. 高精度実数演算のビット幅解析処理追加2

◆小数点以下の0（ゼロ）が多い極小値

0.005のように正数部が無く、小数部のみでかつ、0で始まるような数値の場合
　整数部　　：1ビット、
　小数部　　：３1ビット
　拡張小数部：２ビット
（小数部は合計33ビット分で表現）
として、解析処理をを行います。（-max 32の場合）
拡張小数部は、小数部開始の0（ゼロ）の数により異なります。

ディレクティブで指定する場合は次のようになります。

```
// fpfix fixedP S 32:34 i.t
```

32は、変数全体のビット幅で、
整数部が1ビット、
小数部が31ビット、
拡張小数部が(34-32)で2ビットとなります。
よって、小数部は(31+2)で33ビットで解析します。

※ディレクティブ指定する場合、小数部の0（2進）の数に注意して下さい。
　0の数より大きく拡張小数部を指定すると正常に解析できません。

礎DA ISHIZUE Design Automation Corp. 高精度実数演算のビット幅解析処理追加3

◆整数部が大きい場合

小数部が無く整数部のみの変数で、maxビットを超えるような値となる場合、
maxビットに収まるように仮数表現でデータを保持します。

ディレクティブで指定する場合次のようになります。

```
// fpfix fifxedP S 32:-2 i.t
```

32+2=34ビットのデータを表現するようになります。
但し、下位2ビットは、0となります。

礎DA ISHIZUE Design Automation Corp. 解析データ圧縮起動コマンド１

■概要

本コマンドは、FP-Fixerへの入力データを精度解析に必要なデータのみに圧縮する。

■コマンド

extract_fpfdat [option's] <src_files>

<src_files> : C source files
[option's]
-sw : S/W（DSP)の解析用にデータ圧縮
-hw : H/W（AlgorithmicC)の解析用データ圧縮
-arg <args> : 入力引数
-erate <err_rate> : 評価レート (デフォルト0.1%）
-mft <file_name> : モジュール引数定義ファイル（必須）
-ctf <file_name> : 制約条件ファイル指定（FP-Fixerの制約条件ファイル同様）
その他、FP-Fixerの起動オプション

■コマンド使用例

FP-Fixerの実行ディレクトリで、次のようにコマンドを起動してください。

%extract_fpfdat -mft mod_form.txt -erate 0.3 -arg input.data *.c

①モジュール引数定義ファイル「mod_form.txt」
②入力データファイル「input.data」
③評価レート「0.3%」

礎DA 解析データ圧縮起動コマンド２

ISHIZUE Design Automation Corp.

■コーディング規約

①入力引数は、モジュール関数(//fpfix module)の入出力データのみとする。
②評価関数でNGとなる場合の、レコード番号を出力するコードを記述
　レコード番号を出力するファイルは、「./fpfdat_addrec.txt」固定
③入力引数が複数の入出力ファイルで構成する場合、fpfdatコマンドの「モジュール引数定義ファイル」の順番に注意してください。入出力引数の順番と同じになるように記述してください。

■記述例

```
typedef struct{
  double real;
  double image;
}complex;

int main( int argc, char *argv[] )
{
  complex inbuf[4];
  complex outbuf_sfft[4];
  complex outbuf_sfftfp[4];
  FILE *fp;
  int rr, r;
  int rec_no;
  FILE *rec_fp;

// レコード番号出力ファイルオープン
rec_fp = fopen( "./fpfdat_addrec.txt", "w" );

// モジュール入力データファイルオープン
 fp = fopen( argv[1], "r" );

rr = 0;
rec_no = 0;
while(1){
// データ読込み及びモジュール変数
// argv[1]のファイルから
//  inbufへデータセット

++rec_no;
// 期待値作成
fft(inbuf,outbuf_sfft);

/ モジュールコール
fftfp(inbuf,outbuf_sfftfp); //fpfix module

// 評価
r = evFunc( outbuf_sfft, outbuf_sfftfp, rate );
if( r != 0 ){
 rr = 1; // NGフラグON
 // NGレコード番号に出力
fprintf( rec_fp, "%d\n", rec_no );
      }
    }

    fclose( fp );
    fclose( rec_fp );

    return rr;
}
```

礎DA ISHIZUE Design Automation Corp. ビット限定精度解析の追加1

プロファイルで取得した値を評価レートにかけ、仮の固定小数点のビット幅を算出して、このビット幅を基に固定小数点化を行います。

評価レートは、-erateで指定した値

S/W用（-dsp)及びH/W(-algoc）で、それぞれの限定ビットは、次のようになります。

起動のオプション「-bx_mode」で動作します。

礎DA ISHIZUE Design Automation Corp. ビット限定精度解析の追加2

■S/W(-dsp)向けの限定精度解析

C言語変数の型を基本としたビット幅

char	8 ビット
short	16
int	32
long long	64
accum	-maxt で指定した値、-maxt指定なしの場合は、-maxの値

各型のビット幅は、次のオプションにより変更可能です。

char	:	-bx_byte <size>
short	:	-bx_sort <size>
int	:	-bx_int <size>
long long	:	-bx_longlong <size>
accum	:	-bx_accum <size> (-maxtより優先)

※charからacuumの指定は char < short < int < longlong <accum となるように指定してください。
なお、-max 32 の時は、それ以上の値を指定しても、accumを除いて、32以下になります。
(-maxで指定した値が優先)

礎DA ISHIZUE Design Automation Corp. ビット限定精度解析の追加3

■H/W向け限定精度解析

デフォルトは、4 8 12 16 20 24 28 32 ...
-max で指定した値以下です。

次のように、-bx_bits オプションで変更可能です。

-bx_bits 8:16:24:32（-maxで指定した値以下が有効）

磯解析エラー(ビット不足)のエラー表示強化1

ISHIZUE Design Automation Corp.

■概要

解析時のエラー(ビット不足)のエラー表示を、変数名及びエラー箇所が分かるように表示強化しました。通常は、エラーの変数名及び必要と思われるビット数をのみ表示します。
表示されるビット数はあくまでも目安です。

-erepオプション指定では、次のようなフォーマットを使用います。

<type> : <name> <file_name> <func_name> <iw>:<fw> <u_iw>:<u_fw> <fixVal>
<file_name> <func_name> <lineNo> <floatVal> <exp> <leftVal> <rightVal>
....

<type> : var 変数
cnst 定数
exp 式
<name> : varの時変数名
cnstの時定数の値
expの時 mul(乗算)又はdiv(除算)
<file_name> : ソースファイル名
<func_name> : 関数名、グローバル変数の場合「$」
<iw> : エラー時の正数部ビット幅
<fw> : エラー時の小数部ビット幅
<u_iw> : 必要と思われる正数部ビット幅 (参考)

<u_fw> : 必要と思われる小数部ビット幅 (参考)
<fixVal> : 固定小数点時の値 (参考)

<lineNo> : ソース行
<floatVal> : 浮動小数点時の値
<exp> : 演算子
<leftVal> : 浮動小数点時の入力値
<rightVal> : 浮動小数点時の入力値

※必要と思われる小数部ビット幅は、参考値です。
実際の必要ビット数とは異なる場合があります。

礎DA ISHIZUE Design Automation Corp. 解析エラー(ビット不足)のエラー表示強化2

①実行レベルオプション

-grun_level <level>

<level> の部分追加
0:プロファイルの最大値/最小値で必要ビット幅を計算し-maxまたは-maxtに収まらない場合
エラー出力します。(解析を行う前に行います)
1:解析を実行します。(-grun_levelオプション指定しない場合と同様)

②-erate オプション

-erate <error_rate>

エラーレートをパーセントで指定します。
デフォルトは、(1%)です。

上記、エラーの検出や、必要ビット幅計算に用います。
実際のエラーレートより少し小さめの指定をオススメします。

③その他のオプション

-erep : エラー表示機能を動作します。
-no_erep : エラー表示機能を動作しません。
-erep_o <file> : 上記エラー表示を指定ファイルに出力します。

礎浮動小数点と固定小数点の混在サポート1

ISHIZUE Design Automation Corp.

■概要（浮動小数点変数の混在使用）

浮動小数点変数と固定小数点変数の混在をサポートします。
次のディレクティブで指定された変数は、浮動小数点（double)のままとします。

```
// fpfix float
```

この機能は、AlgorithmicC（-algoc）及びDSP（-dsp）のみ対応です。

また、「-float」オプションで。maxビットを超えるような整数値となる場合
その変数は浮動小数点（double)のままとします。
但し、この機能は、AlgorithmicC（-algoc）のみで対応です。

■注意

DSPの場合、
//fpfix float の追加により、16ビットCPUの場合は、doubleは4バイト(float型）で解析します。
従って、4バイト（float型）で解析するするオプション「-grun_float」を追加しました。

礎浮動小数点と固定小数点の混在サポート2

ISHIZUE Design Automation Corp.

■概要（浮動小数点関数の混在使用）

浮動小数点関数（固定化しない:**unmodule float）及びユーザが作成した関数**をコールする場合、コールする前に、固定小数点化しない(//fpfix float)変数に代入式を記述してください。
この記述をより、浮動小数点関数の呼び出しが可能となります。

■例

```
typedef struct _ss {
  double MA;
  double MB;
} SS_STR;

void func_ummodule( SS_STR *in, double *Y )
{
    ---
}
```

```
void test_fp( double A, double B, double *Y )
{
    SS_STR in_tmp; // fpfix float
    double y_tmp;  // fpfix float

    in_tmp.MA = A;
    in_tmp.MB = B;
    func_unmodule( in_tmp, &y_tmp ); // fpfix unmodule float
    *Y = y_tmp;
}
```

礎DA ISHIZUE Design Automation Corp. ブロック(ループ)単位の解析

■概要

「-bsim_mode」オプションで、ループを1つのブロックとして、解析を実行します。
最初のブロックに関係する変数のビット幅が決定したら、そのブロックからの出力をログテーブルとして作成します。
ログテーブルを作成したブロックは、ログを参照しブロック内は実行しません。

■注意

①このブロック単位の解析の高速化は、負荷のかかるループが平行している場合などに有効です。
②負荷が余りかからないループや、ループが1つの場合などは、解析時間は余計にかかる場合があります。

礎デザインオートメーション

積和変数を指定するディレクティブの追加

■概要

積和変数を明示的に指定するディレクティブ「//fpfix muladd_var」を追加しました。

礎DA 自動発生中間変数のプレフィクス文字列

ISHIZUE Design Automation Corp.

■概要

多項演算の2項式化及びキャストで発生するテンポラリ変数のプレフィクス文字列を指定できるようにしました。

-p_tvar <prefix> : 多項演算の2項式
-p_cvar <prefix> : キャスト

■例

fpfix .... -p_tvar _tmp_ ..

_tmp_0 となります。

製品別機能サポート状況

機能	SW向け	Alogi-C向け	HW-Ansi-C	特記事項
標準実行モード	○	○	○	―
Express実行モード	○	×	×	-ex_mode
ビット限定実行モード	○	○	×	-bx_mode
math関数テーブル生成	○	○	×	―
高精度実数解析	× 予定	○	×	ビット限定モード時は不可
浮動小数点混在	○	○	×	―

価格と出荷時期

■価格

◆年間ライセンス

FP-Fixer Professional ： 120万円
FP-Fixer Express ： 55万円 (FPGA版とSW版それぞれ用意)

◆永久ライセンス

FP-Fixer Professional ： 350万円（年間保守料は、15%）
FP-Fixer Express ： 95万円 (FPGA版とSW版それぞれ用意、
バージョンアップ費用は、35万円)

■出荷

2012年12月21日より

FP-Fixer お問合せ

図研エルミック株式会社　東日本営業部
TEL:045-624-8002　　FAX:045-476-1102
E-mail:info@elwsc.co.jp

株式会社礎デザインオートメーション　営業部
TEL:03-6869-9458　　FAX:03-5501-9054
E-mail(営業部):sales@ishizue-da.co.jp
E-mail(サポート):fpf_support@ishizue-da.co.jp
ホームページ:http://ishizue-da.co.jp/

MEMO