GRAPE-7 インストールガイド

for GRAPE-7 software package version 2.1

株式会社 K & F Computing Research
E-mail: support@kfcr.jp

概要

この文書では GRAPE-7 システムのインストール方法を説明します。

目 次

1 GRAPE-7 の概要

1.1 アーキテクチャ

GRAPE-7 は FPGA を搭載した、内部回路の書き換えが可能なアドインカードです。カードをホスト計算機 (PC) の I/O スロット (PCI-X スロットもしくは PCI Express スロット) に挿すことにより、ホスト計算機の計算能力が強化されます。

本節では GRAPE-7 のアーキテクチャを概観します。GRAPE-7 には Model 100, Model 300, Model 600, Model 800 の 4 種類のモデルがあります (図 1 参照)。

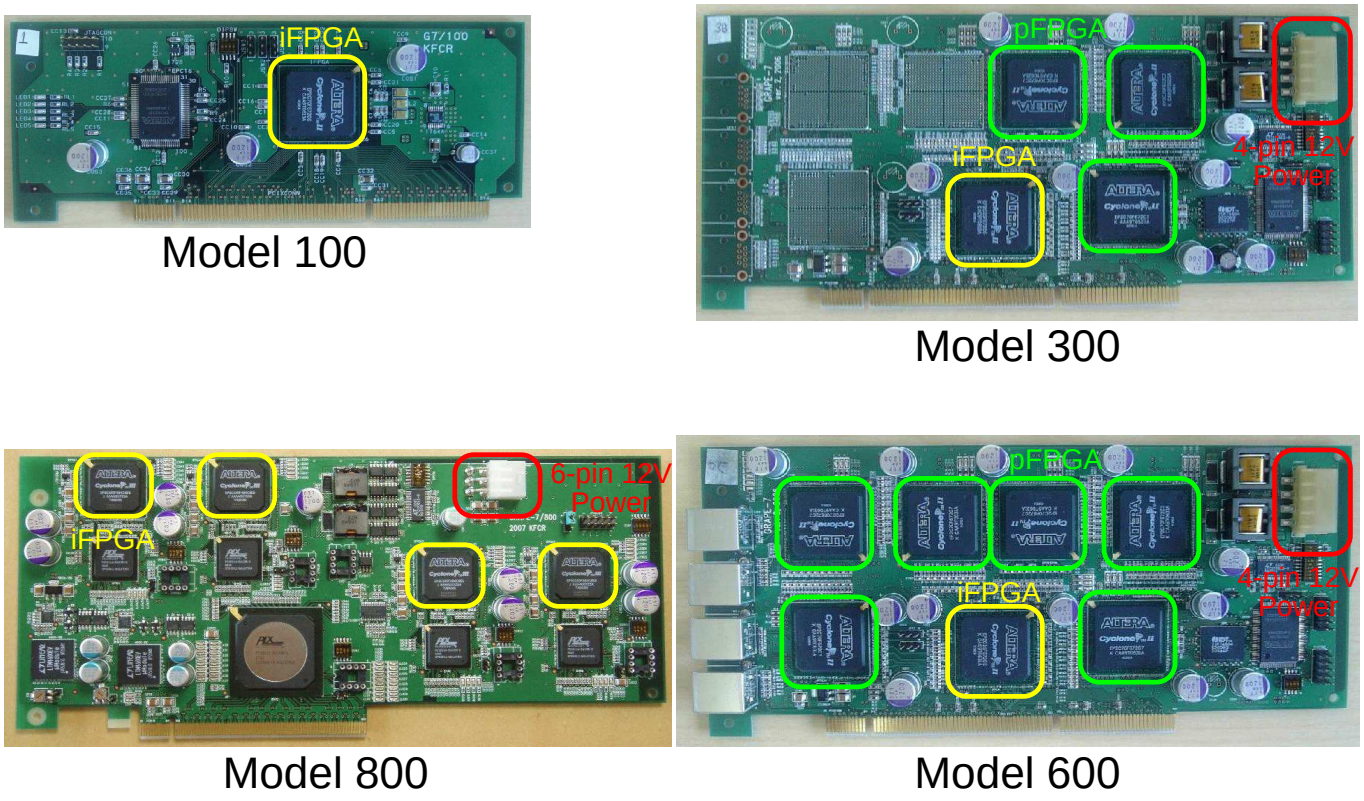

図 1: GRAPE-7 の写真。

1.1.1 Model 100

図 2 に GRAPE-7 Model 100 のブロック図を示します。アドインカードは FPGA をひとつだけ搭載しています。これを **iFPGA** と呼ぶことにします。iFPGA には重力相互作用の計算を行うパイプライン回路 G5PIPE (後述) と、ホスト計算機へのインタフェース回路 (HIB) があらかじめ書き込まれています。ユーザは必要に応じてこの回路を書き換えることが出来ますが、書き換えには別途専用の機器 (Altera 社 *USB Blaster*) が必要です。

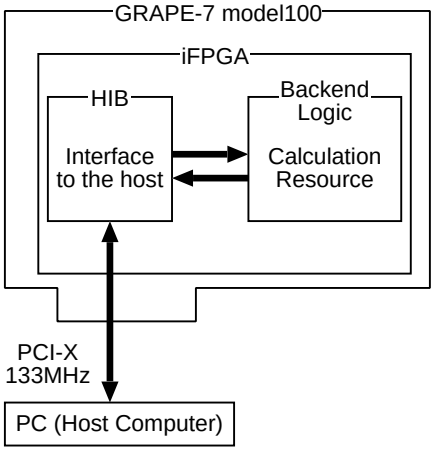

図 2: GRAPE-7 Model 100 の内部構成。

1.1.2 Model 300 および Model 600

図 3 に GRAPE-7 Model 300 および Model 600 のブロック図を示します。アドインカードは **pFPGA** を 3 個 (Model 300) ないし 6 個 (Model600) と、iFPGA をひとつ搭載しています。

pFPGA には相互作用の計算を行うバックエンド回路を書き込んで使用します。カードの電源を入れ直すたびにホスト計算機から書き込みを行う必要があります。書き込みの手順については「GRAPE-7 インストールガイド」の第 3.1 節 「初期設定ユーティリティ」を参照してください。

iFPGA にはホスト計算機へのインタフェース回路が書き込まれています。iFPGA の内部回路は当社によりあらかじめ書き込み済みであり、ユーザがこれを上書きすることは出来ません。

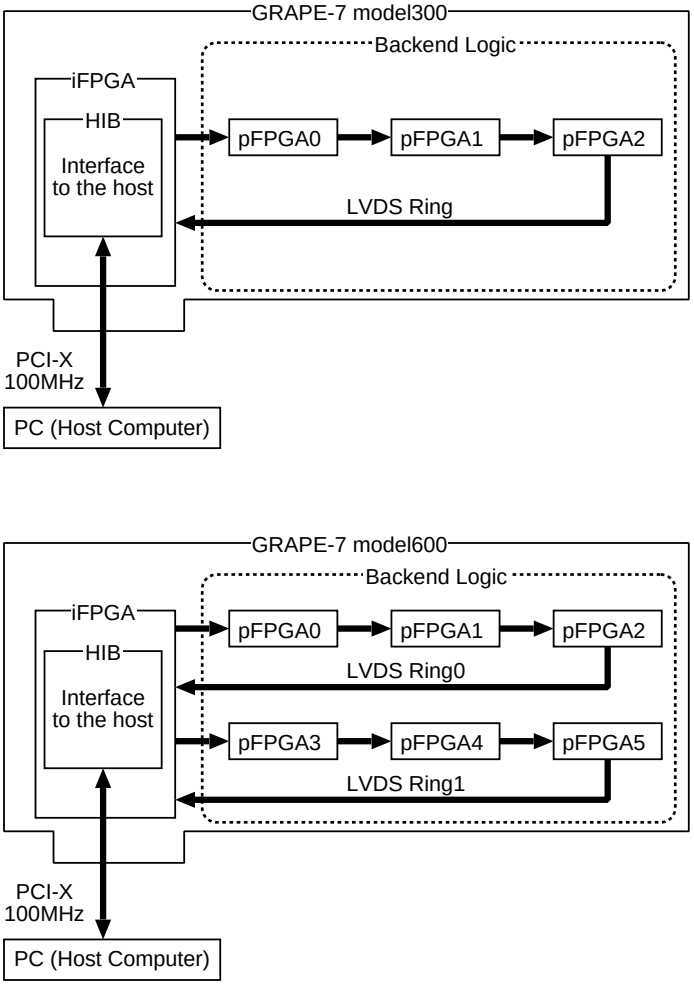

図 3: GRAPE-7 Model 300 および Model 600 の内部構成。

1.1.3 Model 800

図 4 に GRAPE-7 Model 800 のブロック図を示します。アドインカードは iFPGA を 4 個搭載しています。iFPGA には重力相互作用の計算を行うバックエンド回路 G5PIPE (後述) とインタフェース回路があらかじめ書き込まれています。ユーザは必要に応じてこの回路をホスト計算機から書き換えることが出来ます。

4 個の iFPGA は透過型 PCIe スイッチおよび PCIe/PCI-X ブリッジを経由してホスト計算機に接続されています。このためホスト計算機からは、それぞれの iFPGA が別個の PCI-X デバイスとして認識されます。なおユーザはユーザライブラリを通じて、これらを 4 つの独立したデバイスとして扱うことも、また 1 つのデバイスとして扱うことも可能です。

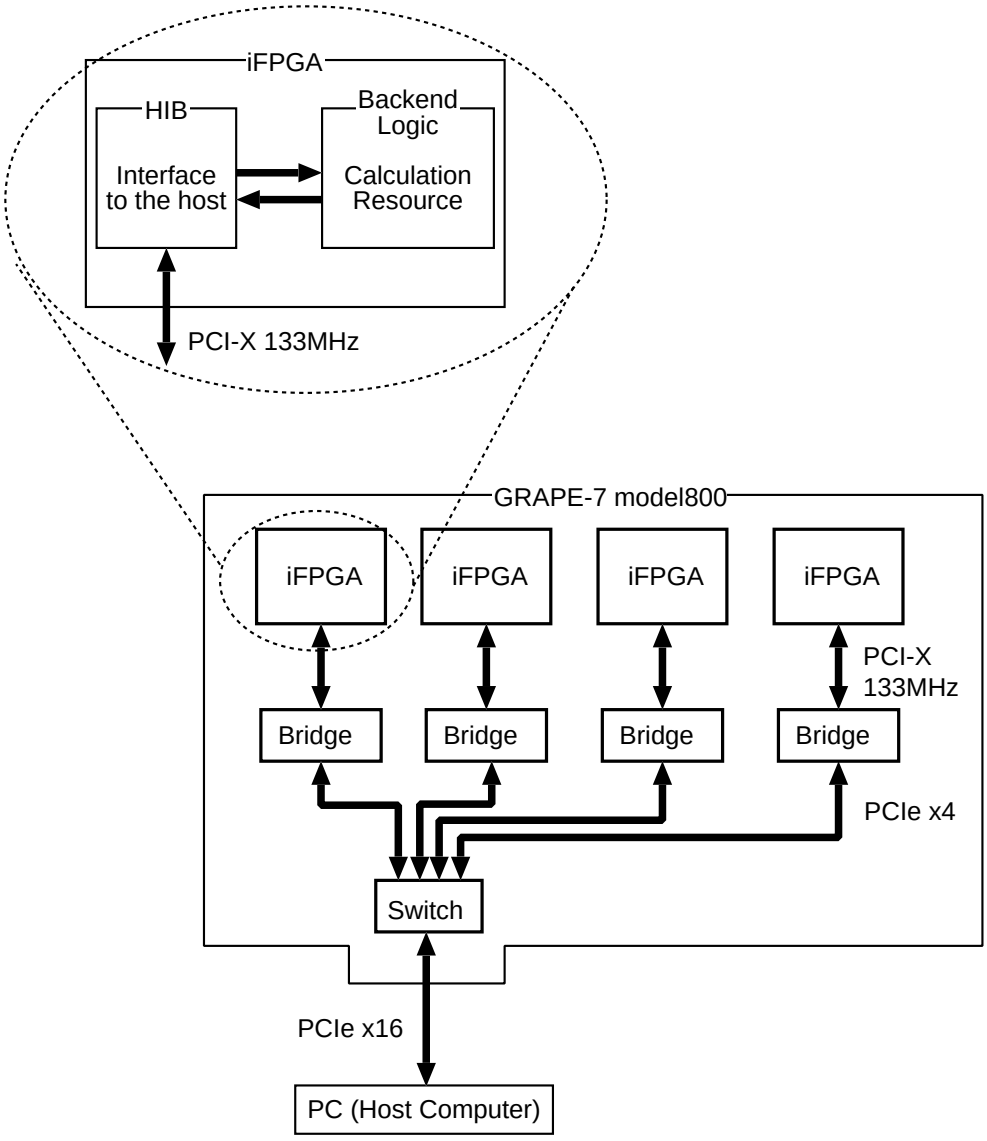

図 4: GRAPE-7 Model 800 の内部構成。

1.2 バックエンド回路

GRAPE-7 にはバックエンド回路として **G5PIPE** および **G5nbPIPE** が付属します。G5PIPE は従来の GRAPE と同様に粒子間の重力を計算する回路です。G5nbPIPE は G5PIPE に近傍粒子リストの作成機能を付加した回路です。

G5PIPE、G5nbPIPE 以外のバックエンド回路は当社 Web サイト[1] にて順次公開されます。またユーザ独自のパイプラインを設計し、バックエンド回路として書き込むことも可能です。バックエンド回路の書き込み手順については第 4 節を参照してください。

以下では GRAPE-7 付属の 2 つのバックエンド回路、G5PIPE および G5nbPIPE について、概要を説明します。

1.2.1 G5PIPE

G5PIPE はパイプラインを用いて粒子間の重力を計算します (図 5)。重力以外の計算、例えば粒子軌道の時間積分などは、ホスト計算機が行います。G5PIPE のパイプラインは GRAPE-5[1] のそれと基本的には同じものです。ただし、高速化のために、重力ポテンシャルの計算機能や近傍粒子リストの作成機能などの一部の機能を省いてあります。

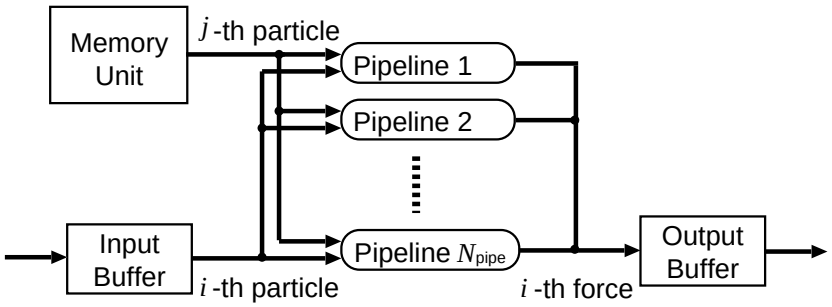

図 5: G5PIPE の内部構成

G5PIPE を用いた重力計算は以下の手順で行われます。なお以下の説明では、重力を受ける側の粒子を i 粒子、重力を及ぼす側の粒子を j 粒子と呼ぶことにします。

手順 (1) ホスト計算機が G5PIPE のメモリユニットへ j 粒子を送ります。送る j 粒子の個数はメモリユニットの大きさを超えてはなりません。

手順 (2) ホスト計算機が G5PIPE の入力バッファへ i 粒子を送ります。送る i 粒子の個数は入力バッファの大きさを超えてはなりません。

手順 (3) ホスト計算機が計算開始コマンドを G5PIPE へ送ります。

手順 (4) パイプライン本数 $N_{\rm pipe}$ ぶんの i 粒子が入力バッファから取り出され、各パイプラインへ 1 粒子ずつ配布されます。

[1] http://www.kfcr.jp/

手順 (5) パイプラインが稼働し、メモリユニット内の全ての j 粒子から、各パイプラインに配布された i 粒子への重力を計算します。計算中は 1 クロックサイクルにつき 1 個の j 粒子がメモリユニットから全 $N_{\rm pipe}$ 本のパイプラインへ放送されます。各パイプラインは j 粒子が自身の i 粒子へ及ぼす重力を計算し、結果を内部のレジスタに積算します。

手順 (6) 各パイプラインは、メモリユニット内のすべての j 粒子から自身の i 粒子への重力を積算し終えると、その結果を出力バッファへ送信します。出力バッファの内容は随時ホスト計算機に回収されます (Model 300, Model 600 の場合には、全ての pFPGA の出力が加算されてからホスト計算機へ回収されます)。

手順 (7) 入力バッファが空になるまで、手順 (4)〜(6) が繰り返されます。

手順 (8) ここまでで、手順 (1) においてメモリユニットへ送信されたすべての j 粒子から、手順 (2) において入力バッファへ送信されたすべての i 粒子への重力が求められました。ホスト計算機はシミュレーションで扱うすべての i 粒子に対して重力を求め終えるまで、手順 (2)〜(7) を繰り返します。

手順 (9) ここまでで、手順 (1) においてメモリユニットへ送信されたすべての j 粒子から、シミュレーションで扱うすべての i 粒子への重力が求められました。ホスト計算機はこれらの重力をメインメモリに保存します。そして次に、ここまでとは別の j 粒子に対して、手順 (1)〜(8) を実行し、得られた力を先にメインメモリに保存した力へ加算します。ホスト計算機はシミュレーションで扱うすべての j 粒子からの重力を求め終えるまで、この手順を繰り返します。

GRAPE-7 ソフトウェアパッケージには、G5PIPE を操作するための関数ライブラリ `g7pkg/lib/libg75.a` が含まれています。このライブラリには GRAPE-5 ライブラリとの後方互換性があります。使用法は「G5PIPE ユーザガイド」(`g7pkg/doc/g5user-j.pdf`) を参照してください。

1.2.2 G5nbPIPE

G5nbPIPE は、G5PIPE に近傍粒子リストの作成機能と、リストを保存するためのメモリを付加した回路です (図 6)。G5nbPIPE のピーク演算速度は G5PIPE に比べて低速です。これは機能追加に伴いパイプライン 1 本の実装に必要な回路資源が増加し、FPGA 内に実装されるパイプラインの本数 $N_{\rm pipe}$ が減少したためです。

GRAPE-7 ソフトウェアパッケージには、G5nbPIPE を操作するための関数ライブラリ `g7pkg/lib/libg75nb.a` が含まれています。このライブラリには GRAPE-5 ライブラリとの後方互換性があります。重力計算機能の使用方法は G5PIPE のそれと同一です。使用法は「G5PIPE ユーザガイド」(`g7pkg/doc/g5user-j.pdf`) を参照してください。近傍粒子リスト作成機能の使用方法については「G5nbPIPE ユーザガイド」(`g7pkg/doc/g5nbuser-j.pdf`) を参照してください。

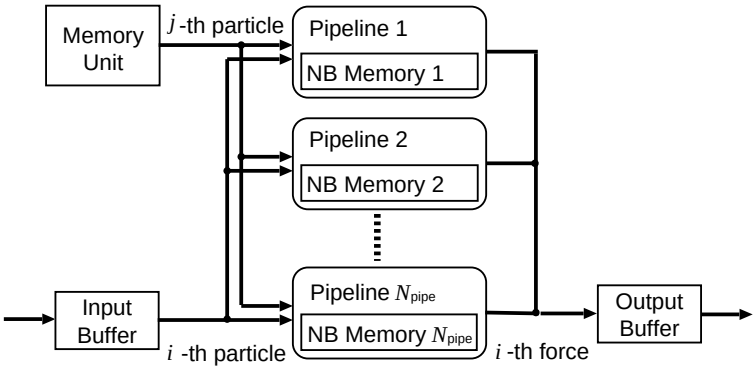

図 6: G5nbPIPE の内部構成

1.3 HIB

Host Interface Bridge (HIB) は GRAPE-7 とホスト計算機との間のデータ転送を制御する回路です。GRAPE-7 ソフトウェアパッケージには HIB を直接に操作するための関数ライブラリ `g7pkg/lib/libhib.a` が含まれています。使用法は「HIB Library Functions Reference Manual」(`g7pkg/doc/hibref.pdf`) で説明されています。

G5PIPE や G5nbPIPE のユーザは HIB ライブラリに関する情報を一切必要としません。HIB ライブラリ関数は G5PIPE/G5nbPIPE ライブラリ関数によって完全に隠蔽されています。

G5PIPE や G5nbPIPE 以外の、ユーザ自身の開発した独自のバックエンド回路を使用する場合には、HIB ライブラリ関数を用いてバックエンド回路の制御をおこなってください。この場合には、G5PIPE ライブラリ関数に相当する上位階層のライブラリを、HIB ライブラリ関数を用いてユーザ自身が開発する必要があります。

2　インストレーション

2.1　ハードウェアのインストール

2.1.1　Model 100

GRAPE-7 アドインカードを、ホスト計算機の PCI-X 133MHz スロットへ挿入します。Model 100 のカードは 133MHz 以下の任意のクロック周波数で動作しますが、性能を最大限に発揮するためには、133MHz スロットを使ってください。ホスト計算機の電源ボックスから出ている 4 ピン 12V の電源ケーブルを、図 7 に示すように接続してください。**冷却に最大限の注意をはらってください。**

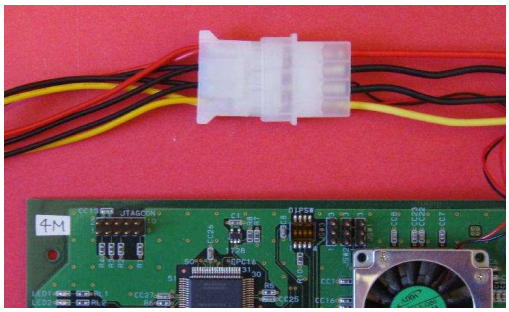

図 7: 電源ケーブルの接続 (Model 100)。

2.1.2　Model 300 および Model 600

GRAPE-7 アドインカードを、ホスト計算機の PCI-X **100MHz** スロットへ挿入します。Model 300 および Model 600 のカードは 100MHz でのみ動作します。周波数が 100MHz より高くても低くても動作しません。ホスト計算機の電源ボックスから出ている 4 ピン 12V の電源ケーブルを、図 8 に示すように接続してください。**冷却に最大限の注意をはらってください。**

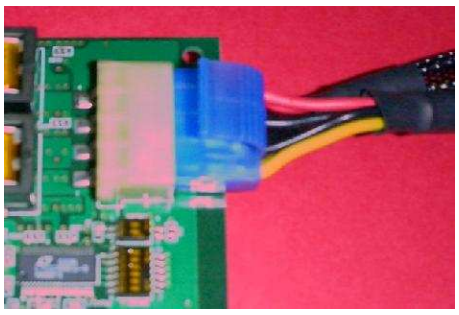

図 8: 電源ケーブルの接続 (Model 300 および Model 600)。

2.1.3 Model 800

GRAPE-7 アドインカードを、ホスト計算機の 16 レーン PCI Express スロットへ挿入します。ホスト計算機の電源ボックスから出ている 6 ピン 12V の電源ケーブルを、図 9 に示すように接続してください。**冷却に最大限の注意をはらってください。**

図 9: 電源ケーブルの接続 (Model 800)。

2.2 ソフトウェアのインストール

ソフトウェアはすべて「GRAPE-7 ソフトウェアパッケージ」(`g7pkg.tar.gz`) にまとめられています。このパッケージの最新版は `http://www.kfcr.jp/support/` から入手できます。

2.2.1 対応プラットフォーム

GRAPE-7 のソフトウェアは現在のところ Linux システム (kernel 2.6 x86_64) のみに対応しています。**ソフトウェアのインストールには、Linux カーネルの完全なソースツリーが必要**なことに注意してください。以下のプラットフォーム上での動作が確認されています。

- Fedora Core 4, 5
- WhiteBox Enterprise Linux 4
- CentOS 5
- Red Hat Enterprise Lunux 4, 5

また CentOS 4 でも動作するはずですが、確認は行っていません。Fedora Core 6 にも近日中に対応する予定ですが、現段階では動作しないことが確認されています。

2.2.2 ソフトウェアパッケージの展開

ソフトウェアパッケージ g7pkg.tar.gz をローカルマシンのディスクにダウンロードし、以下のコマンドを実行すると、パッケージが展開されます。

```
tar xvfz g7pkg.tar.gz
```

この操作によって g7pkg という名のディレクトリが生成され、この中にすべてのソフトウェアが展開されます。以降ではソフトウェアは/usr/g7pkg に展開されているものとして説明を行います。パス名がこれと異なる場合には、説明を適宜読みかえてください。

パッケージには以下のファイルが含まれています。これらすべてが存在するか、インストール作業を始める前に確認してください。

```
./00readme-j --  パッケージの概要説明
./00readme   --  00readme-j の英語版
./doc/       --  ユーザガイド、リファレンスマニュアル、その他の文書
./scripts/   --  インストール用、およびバックアップ用のスクリプト
./include    --  ヘッダファイル
./lib        --  ライブラリ
./driver/    --  デバイスドライバ
./g5util/    --  G5PIPE ユーザライブラリとユーティリティ。それらのソース
                 コード、および pFPGA へ書き込むための回路データ
                 ファイル（.ttf）。
./hibutil/   --  Host Interface Bridge (HIB) ユーザライブラリと
                 ユーティリティ。それらのソースコード。
./direct     --  サンプルコード（計算アルゴリズム:直接法、言語:C)
./directf77  --  サンプルコード（計算アルゴリズム:直接法、言語:Fortran77)
./vtc        --  サンプルコード（計算アルゴリズム:Barnes-Hut ツリー、言語:C)
```

2.2.3 ソフトウェアのインストール

インストールを行うには、ディレクトリ /usr/g7pkg へ移動し、コマンド ./scripts/install.csh を実行します。

コマンドを実行し、カードの枚数等のいくつかの質問に答えると、すべてのソフトウェアが自動的にインストールされます。

```
localhost>./scripts/install.csh
-----------------------------------------------
GRAPE-7 software package installation program
```

```
version 0.3 22-Jan-2007
-----------------------------------------
Using Linux kernel version 2.6 or later? (y/n): y
How many GRAPE-7 cards are you installing?: 1
.................
done
-------------------------------------------------------------------------------
All installation process has been completed.
Now you can check the functions of GRAPE-7 card(s)
following the procedure below:
.................
NOTE THAT STEP (1)-(3) ARE NECESSARY EVERYTIME YOU RESTART THE HOST COMPUTER.
-------------------------------------------------------------------------------
```

インストールが完了すると、以下のライブラリが生成されます。

```
/usr/g7pkg/lib/libhib.a
/usr/g7pkg/lib/libg75.a
/usr/g7pkg/lib/libg75nb.a
```

ここで libhib.a、libg75.a、および libg75nb.aは、順に HIB 関数ライブラリ、G5PIPE 関数ライブラリ、G5nbPIPE 関数ライブラリです。ヘッダファイルは以下に生成されます。

```
/usr/g7pkg/include/grape7x.h
/usr/g7pkg/include/hibutil.h
/usr/g7pkg/include/g5util.h
/usr/g7pkg/include/g5nbutil.h
/usr/g7pkg/include/typedef.h
/usr/g7pkg/include/gp5util.h
```

ここで hibutil.h、g5util.h 、g5nbutil.h はそれぞれ HIB 用、G5PIPE 用、G5nbPIPE 用のヘッダです。grape7x.h はデバイスドライバのヘッダであり、typedef.h は他のヘッダ内部で使用されるヘッダです。gp5util.h は g5util.h へのシンボリックリンクです。このファイルは後方互換性のために存在します。

カードの追加・削除に伴う再インストール: コマンド install.csh は、パッケージを初めてインストールする場合だけでなく、GRAPE-7 アドインカードをホスト計算機の I/O スロットへ追加した場合や、一部のカードをスロットから取り外した場合にも実行して下さい。

3 システムの設定

3.1 初期設定ユーティリティ

ここまでの作業でインストールは完了しました。次にデバイスドライバの Linux カーネルへの組み込み、ホスト計算機の MTRR レジスタ設定、GRAPE-7 アドインカード上の pFPGA へのバックエンド回路の書き込み、の 3 つの作業を行います。以下の手順に従ってこれらの作業を行ってください。**以下の手順 1〜3 はシステムの電源を入れ直すたびに必要です。**

手順 1 (要 root 権限): 手順 1 では GRAPE-7 のデバイスドライバを Linux カーネルへ組み込みます。ディレクトリ `/usr/g7pkg/driver` へ移動し、以下のコマンドを実行してください。

```
make installmodule
```

このコマンドによって、デバイスドライバは Linux カーネルへ組み込まれます。

```
root@localhost# make installmodule
./install.csh grape7x
-- install module grape7x --
1 GRAPE-7(s) found.
rm -f /dev/grape7x[0-9]
/sbin/insmod -f grape7x.ko
mknod /dev/grape7x0 c 253 0
chgrp wheel /dev/grape7x0
chmod 666 /dev/grape7x0
ls -l /dev/grape7x0
crw-rw-rw-  1 root wheel 253, 0 Feb 12 21:14 /dev/grape7x0
-- done --
```

ドライバが正しく組み込まれたかどうかを確認するために、コマンド `/sbin/lsmod` を実行してください。以下の出力が得られるはずです。

```
Module                  Size   Used by
grape7x                 XXXX   0
```

手順 2 (要 root 権限): 手順 2 ではホスト計算機の MTRR (memory type range register) を設定します。ディレクトリ `/usr/g7pkg/hibutil` へ移動し、コマンド `../scripts/setmtrr.csh` を実行してください。

```
root@localhost# cd ../hibutil/
root@localhost# ../scripts/setmtrr.csh
Searching for HIB(s)...Found 1 PCI-X HIB(s).
Trying to set MTRR(s)...
    echo "base=0xfeaf8000 size=0x1000 type=write-combining" > /proc/mtrr
Done.
current setting of MTRRs:
.................
reg02: base=0xfeaf8000 (4074MB), size=   4KB: write-combining, count=1
```

このコマンドによって MTRR が ”write-combining” モードに設定されます。この設定により、ホスト計算機から GRAPE-7 への Programmed I/O Write (PIOW) 方式によるデータ転送の速度が向上します。

> **コマンド `setmtrr.csh` が失敗した場合:** ホスト計算機の設定によっては MTRR を設定できないことがあります (メモリを 4GB 以上搭載している場合や、8 個すべての MTRR が他の PCI デバイスによって既に使用されている場合など)。この場合には MTRR の設定は行わないまま、以降すべての動作テストを行って下さい。MTRR を設定した場合に比べるとホスト計算機から GRAPE-7 への転送性能が 50% 以下に低下しますが、それ以外のすべての機能は正常に動作します。
> バックエンド回路として G5PIPE を使用する場合には、環境変数 `G5_SENDFUNC` の設定によって、転送速度の低下をある程度回避できます。詳細は「G5PIPE ユーザガイド」第 3.2 節「環境変数」を参照してください。
> なお MTRR の設定はホスト計算機から GRAPE-7 へのデータ転送速度のみを向上させ、逆方向 (GRAPE-7 からホスト計算機) のデータ転送速度には無関係です。

手順 3 (Model 300 および Model 600 のみ): 手順 3 では Model 300 および Model 600 に搭載されている pFPGA にバックエンド回路 G5PIPE を書き込みます。この作業は Model 100 および Model 800 には不要です。

ディレクトリ `/usr/g7pkg/g5util/config` へ移動し、回路データの置かれたディレクトリを確認してください。

```
localhost> ls -F
Makefile  boardinfo@  config.c  g5p16nb24c100MHz/  g5p20c100MHz/
p1_grav.ttf@  p2_grav.ttf@  p6_grav.ttf@  sample.c
```

ディレクトリ `g5p20c100MHz` には pFPGA チップ当り 20 本の、100MHz で動作するバックエンド回路 G5PIPE が納められています。ディレクトリ `g5p16nb24c100MHz` には pFPGA チップ当り 16 本の、100MHz で動作する、近傍粒子リスト作成機能つきバックエンド回路 G5nbPIPE が納められています。以下では G5PIPE を書き込むものとして説明を行います。

ディレクトリ g5p20c100MHz へ移動し、以下のコマンドを実行してください。

/usr/g7pkg/scripts/config [*device_id*]

このコマンドにより、デバイス ID として *device_id* を持つカードの pFPGA に内部回路が書き込まれます。引数 *device_id* を省略した場合にはデバイス ID 0 番のカードへ書き込みます。システムにカードを 1 枚しかインストールしていない場合には、そのカードのデバイス ID は 必ず 0 となるため、引数は省略できます。複数のカードをインストールしている場合には、書き込みを行うカードのデバイス ID を確認し、コマンド config への引数として指定する必要があります。デバイス ID の確認にはコマンド /usr/g7pkg/scripts/lsgrape を用います。コマンド lsgrape の利用法については第 5 節を参照してください。

```
localhost> cd g5p20c100MHz
localhost> /usr/g7pkg/scripts/config 0
devid 0 model 600 npipe 16
**** Configuration for model600 [1-6] ****
fin0: 1789902
fin1: 1789902
fin2: 1789902
nconf = 0x3f
datacount = 80000 524288
.................
nconf = 0x24
datacount = 80000 524288
datacount = 100000 1048576
datacount = 180000 1572864
configuration finished
```

書き込んだ回路の詳細情報は、コマンド lsgrape に-v スイッチを与えると得られます。

```
localhost> /usr/g7pkg/scripts/lsgrape -v
devid grape(model)              backend-logic
 0    GRAPE-7(Model600)         G5
        number of chips             : 6
        number of pipelines/chip    : 20
        j-particle memory size/chip : 4096 particles
        fout fifo size              : 8192 bytes
        P3M cutoff                  : available
        neighbor search             : available
        number format               : floating-point
        gravitational potential     : not available
```

ここではパイプライン回路G5PIPEの書き込み手順を例に説明を行いましたが、G5nbPIPEの場合も手順は同様です。また上記2種以外のパイプライン回路も当社Webサイト[2]に順次追加してゆく予定です。それらの回路を使用する場合には、ダウンロードした回路データをディレクトリ /usr/g7pkg/g5util/config 以下へ展開し、同様の手順で書き込みを行ってください。

3.2 動作テスト用プログラム

ここまでの作業で GRAPE-7 システムはバックエンド回路 G5PIPE を用いて多体計算を行える状態となりましたが、ユーザ自身のアプリケーションコードを実行する前に、GRAPE-7 の各機能が正しく動作しているかどうかを以下の手順に従って確認してください。

手順 1 (複数枚のカードを使う場合のみ): テストを行う前に、テストの対象とするカード 1 枚を、環境変数 G5_CARDS によって指定しておく必要があります。例えば、デバイス ID 2 番のカードをテストするには、環境変数を以下のように設定します。

```
csh> setenv G5_CARDS 2
 sh> export G5_CARDS=2
```

カードのデバイス ID はコマンド lsgrape によって得られます。このコマンドの利用法については第 5 節を参照してください。

手順 2: テストを開始するには、ディレクトリ /usr/g7pkg へ移動し、以下のコマンドを実行します。

```
./scripts/check.csh
```

このコマンドは多体計算の時間積分シミュレーションを、指定のカードを使って実行します。シミュレーションを数通りの粒子数について行い、各シミュレーション終了時の最終的な粒子分布を一時ファイルに保存します。これらのファイルを、ディレクトリ /usr/g7pkg/direct/snapshots 内の対応するファイルと比較します。テストには数分の時間を要します。テスト中に何らかのエラーメッセージが出力された場合には support@kfcr.jp へ連絡して下さい。

出力例:

```
localhost>./scripts/check.csh
-----------------------------------------------------------------
 GRAPE-7 functionality test program
```

[2] http://www.kfcr.jp/

```
 version 0.3    08-Feb-2007
.................
---------- in: pl1k out: tmp3052_pl1kT100.snap endt: 100 ----------
nj: 1024
n: 1024 outfile: tmp3052_pl1kT100.snap endtime: 100.000000 n: 0
.................
OK
---------- in: pl12k out: tmp3052_pl12kT100.snap endt: 100 ----------
nj: 12288
n: 12288 outfile: tmp3052_pl12kT100.snap endtime: 100.000000 n: 0
.................
OK
Passed all tests.
```

すべてのテストが正常に終了した場合には、“Passed all tests.” というメッセージが出力されます。これで GRAPE-7 とバックエンド回路 G5PIPE の動作を確認できました。

以上ですべてのインストール作業は完了です。ユーザ自身のコードから G5PIPE を使用できる状態となりました。G5PIPE の使用方法については「G5PIPE ユーザガイド」(`/usr/g7pkg/doc/g5user-j.pdf`) を参照してください。

4 FPGA 内部回路の書き換え

GRAPE-7 には重力相互作用の計算を行うバックエンド回路 G5PIPE が付属しますが、FPGA の内部回路を書き換えることによって、G5PIPE 以外の回路を使用することも可能です。この節では FPGA の書き換え手順について説明します。

4.1 Model 100

書き込みには Altera 社の専用機器 *USB Blaster* が必要です。以下で説明する 2 通りの方法いずれに関しても、書き込みは Altera 社のソフトウェア *QuartusII* の *Programmer* から USB Blaster を操作することによって、JTAG 経由で行います。実際に操作を行う前に、USB Blaster を Model 100 の JTAG コネクタへ接続しておいてください (図 10)。QuartusII の操作手順については、QuartusII のマニュアルを参照してください。

方法 1 (暫定的な書き込み): QuartusII Programmer を用いて .sof 形式の回路データを iFPGA に書き込みます。書き込み終了後、ホスト計算機を再起動 (hot reboot) します。電源を落とさずに再起動することに注意してください。電源を落とすと書き込んだデータは消去されてしまいます。その場合は改めて QuartusII Programmer を用いて書き込みを行い、その後ホスト計算機を再起動 (hot reboot) してください。

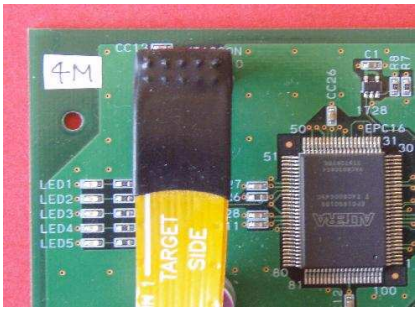

図 10: USB Blaster の接続 (Model 100)。

方法 2 (恒常的な書き込み): QuartusII Programmer を用いて .pof 形式の回路データを configuration ROM に書き込みます。書き込み終了後、Model 100 の電源を落とし、起動 (cold boot) しなおすと、ROM に書き込んだ回路データが iFPGA へと自動的にロードされます。ROM に書き込んだデータは電源を落としても消去されませんので、起動のたびに書き込み操作を行う必要はありません。

4.2 Model 300 および Model 600

Model 300 および Model 600 に関しては、G5PIPE 以外の回路の書き込みも G5PIPE 回路の場合と同じ手順で行えます。手順の詳細は第 3.1 節「手順 3」を参照してください。

4.3 Model 800

書き込みに特別な機器は不要です。4 個の iFPGA それぞれに対応する計 4 個の configuration ROM へ、ホスト計算機から、.rpd 形式のデータを直接書き込みます。書き込みには `/usr/g7pkg/scripts/config800` コマンドを用います。例えばシステムに 1 枚の Model 800 のみがインストールされている場合に

```
/usr/g7pkg/scripts/config800 g5.rpd 0 1 2 3
```

を実行すると、Model 800 に搭載されているすべての iFPGA (デバイス ID 0〜3) の ROM に、カレントディレクトリにおかれた回路データ g5.rpd が書き込まれます。コマンド `config800` の詳細については第 5 節を参照してください。

書き込み終了後に Model 800 の電源を落とし、起動しなおす (cold boot) と、書き込んだ回路データが ROM から iFPGA へと自動的にロードされます。ROM に書き込んだデータは電源を落としても消去されませんので、起動のたびに書き込み操作を行う必要はありません。

5 コマンドリファレンス

GRAPE-7 ソフトウェアパッケージに含まれる各種コマンドの使用方法を説明します。これらのコマンドはすべてディレクトリ /usr/g7pkg/scripts/ に置かれています (一部のコマンドについては、後方互換性を保つために/usr/g7pkg/hibutil/ や /usr/g7pkg/g5util/config/ 内にもコピーが置かれています)。

install.csh — ソフトウェアパッケージのインストール

書式: install.csh

説明: GRAPE-7 ソフトウェアパッケージのすべてのソースコードをコンパイルします。パッケージを初めてインストールする場合だけでなく、GRAPE-7 アドインカードをホスト計算機の I/O スロットへ追加した場合や、一部のカードをスロットから取り外した場合にもこのコマンドを実行して下さい。

bakup.csh — ソフトウェアパッケージのバックアップ

書式: bakup.csh [*package_name*]

説明: GRAPE-7 ソフトウェアパッケージのスナップショット (tarball) を生成します。ディレクトリ /usr/g7pkg へ移動し、コマンド ./scripts/bakup.csh を引数なしで実行すると、g7pkg という名前のパッケージが生成されます。つまり生成される tarball のファイル名は/usr/g7pkg/g7pkg.tar.gz となり、この tarball を展開したときの最上位ディレクトリ名は ./g7pkg となります。引数を与えた場合には g7pkg の代わりにその引数がパッケージ名として使用されます。

```
localhost>./scripts/bakup.csh
-----------------------------------------------
GRAPE-7 software package back up program
version 0.3 22-Jan-2007
-----------------------------------------------
Specify the root name for the package (default g7pkg): g7pkg20070212
./00readme
./g5util/Makefile
./g5util/g5util.h
.................
./g7pkg20070212/scripts/check.csh
./g7pkg20070212/scripts/gensnap.csh
```

```
Created ./g7pkg20070212.tar.gz
```

なお、GRAPE-7 ソフトウェアパッケージをアンインストールするには、単に/usr/g7pkg 以下のすべてのファイルを削除してください。

```
localhost> cd /usr
localhost> rm -rf g7pkg
```

setmtrr.csh — MTRR レジスタの設定

書式: `setmtrr.csh`

説明: ホスト計算機の MTRR (memory type range register) を”write-combining” モードに設定します。ホスト計算機から GRAPE-7 への Programmed I/O Write (PIOW) 方式によるデータ転送の速度が向上します。

lsgrape — GRAPE アドインカード情報の取得

書式: `lsgrape [-l] [-v] [-d` *device_id*`] [-h]`

説明: システムにインストールされている GRAPE アドインカードに関する情報を出力します。

オプション:

`-l`
パイプライン本数、粒子メモリサイズ等の詳細な情報を表示します。

`-v`
`-l` と同じです。

`-d` *device_id*
デバイス ID として *device_id* を持つカードの情報のみを表示します。省略すると、システムにインストールされている全てのカードの情報を表示します。

使用例:

```
localhost>/usr/g7pkg/scripts/lsgrape
devid grape(model)       backend-logic
  0   GRAPE-7(model100)  G5
```

```
  1   GRAPE-7(model300)  G5
  2   GRAPE-7(model300)  empty
```

この出力は、システムに Model 100 が 1 枚と Model 300 が 2 枚インストールされており、デバイス ID 0 の Model 100 とデバイス ID 1 の Model 300 には既に G5PIPE 回路が書き込まれていることを意味します。

```
localhost> /usr/g7pkg/scripts/lsgrape -v -d 0
devid grape(model)              backend-logic
 0    GRAPE-7(Model600)         G5
        number of chips             : 6
        number of pipelines/chip    : 20
        j-particle memory size/chip : 4096 particles
        fout fifo size              : 8192 bytes
        P3M cutoff                  : available
        neighbor search             : not available
        number format               : floating-point
        gravitational potential     : not available
```

この出力は、システムにデバイス ID 0 番のカードとして Model 600 がインストールされていることを意味します。またその Model 600 には G5PIPE 回路が書き込まれており、各 pFPGA につき 20 本のパイプラインと 4096 粒子ぶんのメモリユニットが内蔵されていることなどの情報も読み取れます。

`config` — Model 300 および Model 600 内部回路の書き換え

書式: `config` [*device_id*]

説明: Model 300 および Model 600 に搭載されている pFPGA へ、.ttf 形式の回路データを書き込みます。回路データはカレントディレクトリに置かれている必要があります。引数 *device_id* は書き込みを行うカードのデバイス ID です (カードのデバイス ID はコマンド `lsgrape` を用いて得られます)。デバイス ID を省略すると、デバイス ID 0 番のカードに書き込みます。

`config800` — Model 800 内部回路の書き換え

書式: `config800` <*rpd_file*> [*ifpga_ids*]

説明: Model 800 に搭載されている 4 個の iFPGA それぞれに対応する configuration ROM へ、.rpd 形式の回路データ *rpd_file* を書き込みます。引数 *ifpga_ids* は書き込みを行う ROM のデバイス ID を、空白で区切って並べた文字列です。省略すると、0〜3 のデバイス ID を持つ 4 個の ROM に書き込みます。

使用例:

システムに 1 枚の Model 800 のみがインストールされている場合に

```
/usr/g7pkg/scripts/config800 g5.rpd 0 1 2 3
```

を実行すると、Model800 に搭載されているすべての iFPGA (デバイス ID 0〜3) の ROM に、カレントディレクトリにおかれた回路データ g5.rpd が書き込まれます。

参考文献

[1] Kawai A., Fukushige T., Makino J., and Taiji M.,
GRAPE-5: A Special-Purpose Computer for N-Body Simulations,
Publ. Astron. Soc. Japan (2000), Vol. 52, p. 659,
http://xxx.lanl.gov/abs/astro-ph/9909116.

更新履歴

version	date	description	author(s)
2.1	09-Feb-2008	Model 800 に対応。 G5nbPIPE に対応。 FPGA 内部回路の書き換え方法を追記。 コマンドリファレンスを追記。	AK、TF
1.0	13-Feb-2007	初版作成	AK、TF