

動的再構成可能な Hw/Sw 混載システムの FPGA による PCI インターフェース 技術研究報告形式

○志賀裕介† 今中晴記† 上野貴史† 工藤健慈† 関根優年†

† 東京農工大学工学部
〒184-8588 東京都小金井市中町 2-24-16

E-mail: † {you, imanaka, ueno, k2x, sekine}@sekine-lab.ei.tuat.ac.jp

あらまし データ量が膨大な画像処理、音声処理などのリアルタイム処理を可能とするため、高速なハードウェアと柔軟なソフトウェアとを併せ持った Hw/Sw 混載システムの実現を目指している。この Hw/Sw 混載システムをオブジェクト指向で構成するため、HwObject と SwObject とを導入する。HwObject は PCI バス上に配置されたメモリ型 PCI デバイスとして認識される HwModule 上に設けられた FPGA にロードされ、SwObject と同様に実行することが可能である。それによってハードウェアとソフトウェアが強調して動作することが可能となる。HwModule とは CPU、メモリ、FPGA×3 によって構成された PCI デバイスであり、3 つの FPGA を連動させてパイプライン処理することも考えている。本稿では HwModule を制御する PCI インターフェースに重点を置き、インターフェースと内部バスの制御、FPGA の制御などについて述べる。

キーワード FPGA, Hw/Sw 混載システム, PCI インターフェース

PCI Interface by FPGA of Hardware and Software heterogeneous system The format of Technical Report

Yusuke SHIGA †, Haruki IMANAKA †, Takashi UENO †, Kenji KUDO, Masatosi SEKINE †

† Tokyo University of Agriculture & Technology
Nakamachi 2-24-16, Koganei Tokyo, 184-8588 Japan

E-mail: † {you, imanaka, ueno, k2x, sekine}@sekine-lab.ei.tuat.ac.jp

Abstract In order to make possible real-time operations, such as image processing with a lot of data, and speech processing, realization of the Hardware and Software heterogeneous system having high-speed hardware and flexible software is spent. Because this Hw/Sw heterogeneous system is constituted from object-orientation, HwObject and SwObject are introduced. It is possible for HwObject to be loaded to FPGA prepared on HwModule recognized as a PCI memory device arranged, and to perform like SwObject. It is possible for hardware and software to emphasize and operate by it. HwModule are the PCI device constituted by CPU, a memory, and FPGA×3, and it also considers carrying out pipeline processing tied to three FPGA. Control of an interface and an internal bus, control of FPGA, etc. are described with emphasis on the PCI interface which controls HwModule by this paper.

Keywords FPGA, PCI, Hardware and Software heterogeneous system

1. はじめに

コンピュータ・システムは、トランジスタ集積度ならびにマイクロプロセッサの動作周波数の長足なる向上を基盤にして、以下の手法で性能向上を図ってきた。

- マイクロプロセッサにおける命令レベル並列処理、すなわち、パイプライン処理(動作周波数向上の活用)およびスーパースカラ処理(集積度向上の活用)の採用。
- マイクロプロセッサにおけるオンチップ・キャッシュ(集積度向上の活用)の採用。
- 主記憶容量の増大化(集積度向上の活用)。

しかし、ノイマン・ボトルネックの顕著化として、マイクロプロセッサの集積度および動作周波数の向上が原因となり、(1) マイクロプロセッサと主記憶の性能差拡大(2) メモリ・バンド幅の限界などあり、マイクロプロセッサの動作周波数向上が年率 22% なのに対して、DRAM の行アクセス時間の向上は年率 7% と低いため、両者間の性能差はますます拡大していく傾向にある。また、大量の情報を処理すると同時に判断を行う画像処理などでは条件分岐が多く、ストールが頻繁に多発し、命令の処理能率は大きく低下してしまう。こうした背景の元に再構成可能な LSI を用いて可塑的に変化する能力を持ったシステムがかねてより注目されている。その主な方式として

(1) 汎用エンジン方式

汎用計算機のベクトルファシリティのように、ワークステーション等のホスト計算機に付加して計算集約的な処理や汎用プロセッサが苦手な処理を任せる方式である。代表的なものには、米 SPLASH、米 IBM 社の GANGLION、神戸大の RM、米 HP 社の TERAMAC などがある。

(2) リコンフィギャラブル・コプロセッサ方式

汎用マイクロプロセッサと組み合わせて利用する点に特徴があり、応用に適したカスタム命令の実装や処理時間の大半を費やす付加の重い計算や入出力処理のハードウェア化によって高速化を図る方式である。代表的なものには、米 Brown 大の PRISM、豪 Melbourne 大の RECON、仏 DEC-PRL の Pamette、奈良先端代の GPC-SS などがある。

(3) リコンフィギャラブル・プロセッサ方式

マイクロプロセッサ自体をリコンフィギャラブル・ロジックで構成し、対象アプリケーションの必

要に合わせてプロセッサ内部を on-the-fly で再構成する方式である。つまり、マイクロプロセッサ自体を部分再構成可能な FPGA で構成し任意に配置可能なカスタム命令モジュールを用いてアプリケーションに応じた命令セットによって処理を行う方式である。代表的なものに、米 Brigham Young 大の DISC である。

(4) SCORE、仮想ページング方式

RWCP が開発した、PC クラスタ向けの並列オペレーティングシステム。クラスタシステム上に SCore をインストールすると、ノード間の効率的な相互通信や、複数ユーザの並列プログラムの時分割実行などが可能となる。が挙げられる。

2. Hw/Sw 混載システム概要

ハードウェアの高速性とソフトウェアの柔軟性という両者の長所を兼ね備える Hw/Sw 混載システムの開発である。これは動的再構成可能な FPGA を用いて実現させるものである。画像処理、音声処理などの大量なデータをそれに対応した回路に FPGA を書き換え処理させることにより高速なデータ処理を可能にすることを目的としている。Hw/Sw 混載システムではオブジェクト指向によって、ハードウェアをオブジェクト(HwObject)として、ソフトウェアのオブジェクト(SwObject)からハードウェアとして意識させずに呼び出し利用することが可能である。HwObject は HDL によって書かれた回路データ(HwNet)および処理データとをあわせたものを示す。これに対して SwObject は通常のソフトウェアで実装されるオブジェクトである。ハードウェアのほうが高速に実行できる場合は、つまり大量のデータ(画像、音声など)を処理する場合などには HwObject に、複雑でソフトウェアに処理をさせた方がいい場合は SwObject に割り当てる。この HwObject を動作させる場として、CPU、Memory、FPGA×3 を搭載した PCI デバイスであり、以後 HwModule と呼ぶ。現時点では、この HwModule は単体での動作のみであるが、今後 HwModule 間の通信を整備することによって、複数個の HwModule を並列に並べる事によって並列分散処理の実現をめざす。

3. HwModule の設計と実装

3.1 システム構成

システムの構成として以下の通りである。

- FPGA：200K ゲート相当、456 ピンファインピッチ BGA を 4 つ使用。1 つは PCI インターフェースとして使用。残りの 3 つは HwNet として使用する。
- メモリ：アクセスタイム 10ns である SRAM をローカルバス側に 1MB を 3 つ、CPU 側に 512KB を使用。
- CPU：33MHz 動作を使用
- バス構成：CPU 側とローカルバス側と 2 つにバスを分けている。LB 側は PCI と FPGA1 と SRAM1 を、FPGA1 と FPGA2 と SRAM2 を、FPGA2 と FPGA3 と SRAM3 をバスで接続している。

図 1 に HwModule のシステム構成図を示す。

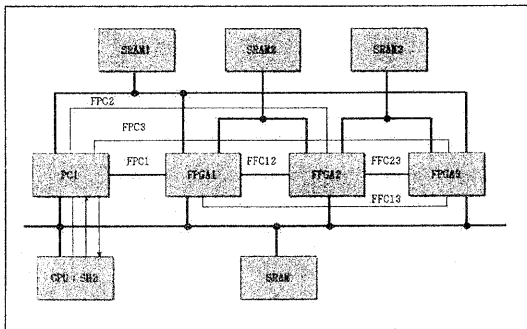


図 1 HwModule のシステム構成

ここで CPU 側と LB 側とに 2 つのバスに分けた理由として HwNet と CPU が互いに競合することによってのパフォーマンスの低下を避けるための配慮である。LB 側の SRAM には入出力データを、CPU 側の SRAM には HwNet 等を書き込む。CPU は FPGA(HwNet)の管理、および Host の通信を行う。HwObject を動的に扱えるようにするためにコマンドを作成し、Host 上の Object Manager(OM)と HwModule 間でコマンドをやり取りすることで、Host から HwModule を利用することを可能にさせる。各 FPGA 同士(FFC)および、PCI と各 FPGA(FPC)を 12bit の GPIF を結び、これらを用いることによって制御・管理する。PCI から FPGA を

制御・管理したい時には各 FPC を、FPGA 間で制御したい時には FFC を用いる。これらを用いることによって FPGA 同士を連動させて動作(パイプライン処理)させることも考慮に入れている。さらに、FPGA3 は PCI-FPGA1 と SRAM1 を結ぶバスとも接続されており、これによって処理を循環させることも可能である。

3.2 PCI の仕様

PCI バス仕様 Rev.2.2 を元に、32bit、33MHz のメモリ型の PCI デバイスとして PCI ターゲットデバイスを設計している。I/O 空間、パリティなどは設計していない。図 2 に PCI_TGT の状態遷移図を示す。

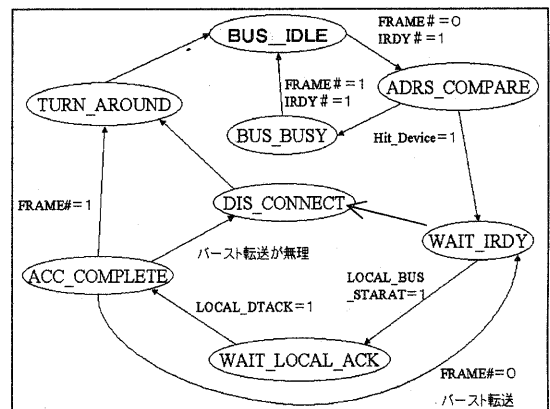


図 2 PCI_TGT の状態遷移図

ベースアドレスはローカルバス側、CPU 側とともに 16MB と設定している。PCI のローカルバス側はアドレスによって各 SRAM へのバス権を要求する。また、CPU 側は 16bit であるため、ハード的に 32bit を 16bit×2 に変更して 2 回に分け SRAM に読み書きしている。ローカルバスにある FPGA が動作している時に PCI バス側の応答があった場合には、現在では待機してしまうために PCI バスを占有してしまう。そのため、PC の全体的なオーバーヘッドになっている。これをなくすために FPGA が動作している際には PCI からのアクセスに対してはある一定の時間待機した後に、FPGA の動作を止め busy 状態にし、STATE: DISCONNECT に遷移して PCI の動作を停止する。その後、再度 PCI のアクセスを受けたら動作開始するように、アクセス要求がなくなったら

busy 状態を開放することも考慮に入れている。

3.3 ローカルバス制御

PCI インターフェースに使用する FPGA には、先に述べたようにローカルバスを制御・管理するために Module・FBC を設けている。FBC によってローカルバスは管理されている。各 FPGA 内に設けられた BUS インターフェース Module(LBS 1・3)が FBC にバス・リクエスト要求を出すことによって、各バスの使用権を得る。Host から見ると HwModule のローカルバス側にある 3 つの SRAM は 1 つの SRAM として認識されている。FBC はアドレスを見ることによって、どの SRAM に対してのアクセスかを判断し、それに対応してアクセス要求を出す。PCI はすべての SRAM に対してアクセスが可能であり、FPGA1 は SRAM1 と SRAM2、FPGA2 は SRAM2 と SRAM 3、FPGA3 は SRAM1 と SRAM 3 にアクセスが可能である。優先順位としては表 1 に示す通りである。PCI から SRAM2 に対してのアクセスは FPGA1 を経由して行い、SRAM3 に対しては FPGA3 を経由してアクセスが行われる。

表 1 バスの優先順位

優先順位	1	2	3
SRAM1	PCI	FPGA3	FPGA2
SRAM2	PCI	FPGA1	FPGA2
SRAM3	PCI	FPGA2	FPGA3

3.4 HwNet の仕様

FPGA 内のブロック図を図 3 に示す。HwNet は 2 つのバスと制御バス・インターフェース内のコンフィグレーション・レジスタからのポートを持つ。CPU から HwNet へはコンフィグレーション・レジスタを介してアクセスをする。コンフィグレーション・レジスタに置かれる情報は、HwNet へ割り当てられたアドレス情報、HwNet が処理を開始するためのフラグなどである。CPU と Host は CPU 側のバス上に置かれた SRAM の決められた領域を用いてデータの共有、通信を行う。共有されるデータには FPGA や HwObject の状態・情報に与えられたアドレス、実行状態などが挙げられる。Host はこの領域を参照することによって、HwObject の情報を得ることが可能である。

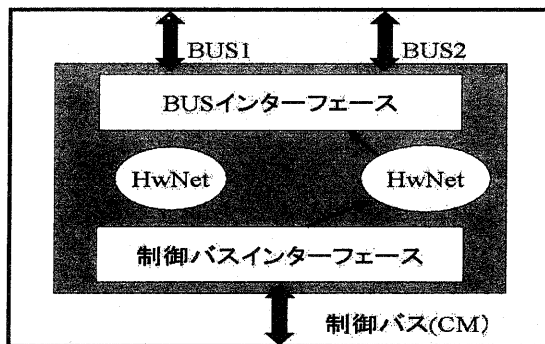


図 3 FPGA 内部ブロック図

現在 HwModule で行われる基本的な処理の流れとしては

(1) Create

LM 側の SRAM の空き領域を新しく生成する HwObject に割り当てると同時に、空いている FPGA をチェックし HwNet を FPGA に書き込む。

(2) Active

CPU から HwNet へアドレス情報の書き込みと、HwNet のその他の初期化を行う。Host はこの間にテーブルから得られたアドレスに初期データの書き込みを行う。

(3) Execute

FPGA 内の制御バス・インターフェースに設けられたコンフィグレーション・レジスタの HwNet 動作開始フラグを有効にし、HwNet が処理を開始する。

(4) Delete

HwNet の動作を必要に応じて停止させ、テーブル上のデータをクリアする。
を現時点では考えている。

4. 結果と考察

メモリを 1MB から 256B に変更して、SilosⅢ上で、ローカルバスの制御に注目してシミュレーションをした結果を載せる。

4.1 SilosⅢでのシミュレーション結果

図 4 に SilosⅢ上での PCI とローカルバスとの一連の動作として、各 FPGA、および PCI からのバス要求が同時に起こった際、各 Module からの無造作なバスアクセスが起こった際の結果を示している。その一連の動作のシミュレーション結果を示す。

また、図 5 に PCI から各メモリへのアクセスのシミュレーション結果を示す。図 4 では PCI→SRAM3, FPGA1→SRAM1, FPGA2→SRAM2, FPGA3→SRAM1 へのバス要求を出している時の動作である。図から見てもらえるとおり、SRAM1 へのアクセスは PCI, FPGA1, FPGA3 がバスの要求しているが、先に述べたように、優先順位としては PCI, FPGA3, FPGA1 の順番になっているので、その通りにアクリッジが出されているのがわかる。図 6 においては、PCI から各 SRAM へアクセスを行っているが、SRAM2, SRAM3 は FPGA を経由しなくてはアクセスできないので、その処理の流れを示す。

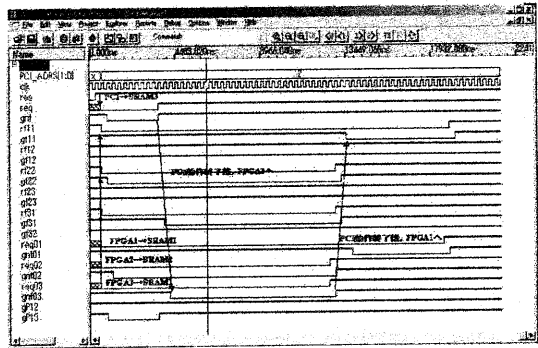


図 4 バスの一連の動作

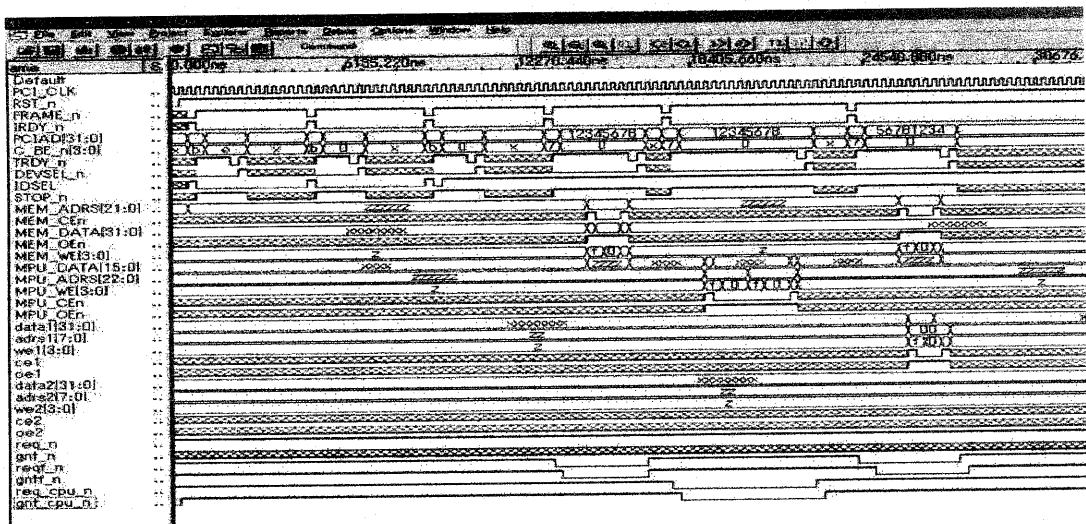


図 5 PCI から SRAM へのアクセス

4.2 Foundation でのシミュレーション結果

表 2 に Foundation に各 Module を Implementation した結果と HDL 記述量を示した。全体の処理速度が 64.859MHz であるので、PCI としての動作速度 33MHz を満たしている。

表 2 .各 Module の説明

Module	Module の説明	HDL 記述量	回路規模	処理速度
PCI_TOP	TGT と LOCAL_BUS をつなぐ Module	121 行	4,509	64.859MHz
PCI_TGT	PCI のターゲット部分の Module	243 行		
LOCAL_BUS	PCI のローカル部分の Module	472 行		
FBC	ローカルバスを制御・管理する Module	238 行	576	117.398MHz
LSB13	FPGA1 と 3 に置かれるバス・インターフェース	240 行	2,340	80.283MHz
LSB2	FPGA2 内に置かれるバス・インターフェース	204 行	1748	95.730MHz

5. むすび

PCI ターゲットとしての動作を改良して，ローカルバス，及び FPGA の動作処理を制御することを考慮に入れなくてはならない．

現在 HwModule ボードを製作中である．この HwModule ボードが完成次第，このボードを用いて本稿で述べた一連の動作を確認していく予定である．これによってシステムのオーバーヘッド，制御上の問題点を追求して，Hw/Sw 混載システムでの HwObject の有効性などを研究していきたい考えである．

文 献

- [1] 高橋義造，“並列処理機構”，丸善株式会社，平成元年．
- [2] 金子俊夫，“OPENDESIGN No. 7” CQ 出版株式会社，1995 初版発行，1999 第 7 版発行．
- [3] 山岸誠仁，“PCI デバイス設計入門”，CQ 出版株式会社，2000．
- [4] 青木忠一，伊藤元昭，川田茂，白川千洋，鈴木文雄 “パネル討論，プログラマブル論理デバイスの現状と将来動向～FPGA は ASIC を凌駕するか～” 1997，回路とシステム軽井沢ワークショップ，pp. 179-180．
- [5] 市川周一，島田俊夫：“パーソナル・コンピューティング指向の動的再構成可能 PCI カード，” 第 5 回 FPGA/PLD Design Conference & Exhibit，pp. 269-277 (1997)．