

Hw/Sw 混載システムに於ける

HwObject モデルとその制御手法

HwObject-Model on an Hw/Sw Heterogeneous System and its controls

工藤 健慈
Kenji KUDO

今中 晴記
Haruki IMANAKA

志賀 裕介
Yusuke SHIGA

関根 優年
Masatoshi SEKINE

1. まえがき

FPGA を用いた再構成可能なシステムの研究が行われている([1],[2])。Davis 等は、アプリケーションの開発がもっと容易になる必要があると考え、オブジェクト指向の Java Hardware Objects モデルを提案している[3]。この考えを進めて、動的に再構成可能である Hw/Sw (ハードウェア/ソフトウェア) 混載システムを実現する HwModule を提案した([4],[図 1])。この HwModule の PCI 動作は通常の PCI デバイスと同様で、例えば、256KB ブロックモードで 6[μ s]であり、今後、バス・ネックや FPGA の速度改善の余地がある。

本研究では、オブジェクト指向を適用した Hw/Sw 混載システムの開発法を、市販の C++言語のコンパイラを用いて、アプリケーションにライブラリを付加することで簡便に実現する提案をしている。従来の Sw による実装のオブジェクト(SwObject)に対して、FPGA による回路(Hw)をアプリケーションに適した大きさの機能単位にオブジェクト化したものを HwObject と呼ぶ。これら Object ライブラリを用いて、Hw と Sw との境界を無くし、オブジェクト指向による設計生産性の向上を考えている。本稿では、HwObject を実装して制御する上で、必要となる基盤構成部分の提案を行う。

2. Hw/Sw 混載システム

2.1. 全体の構成

システムは図 1 に示すように単一プロセッサとメイン・メモリ(MM)からなる通常構成を Host とし、SwObject の実体は MM に置かれる。HwModule は、HwObject を実装するためのメモリ型 PCI デバイスで、FPGA、ローカル・メモリ(LM1,…)、Host から隠蔽されているマイクロプロセッサ、PCI-Bus インターフェイスで構成される。HwObject は、仮想的な回路(HwNet と呼ぶ)として HwModule 上の FPGA に実装され、そのデータ部は MM と LM に蓄えられ、MM にはインターフェイスを構成する。複数の FPGA の中で動作していない FPGA の HwNet は他が動作中でも入れ換えが可能である。マイクロプロセッサは Host との通信、FPGA への回路の書き込み、HwObject への LM の割り当てと解放を行う。

HwObject を使うアプリケーションは、回路特有の割り込み処理を含めて、HwObject 制御に関する基盤構成部をコンポーネントして組み込むプロジェクトにより通常通り生成

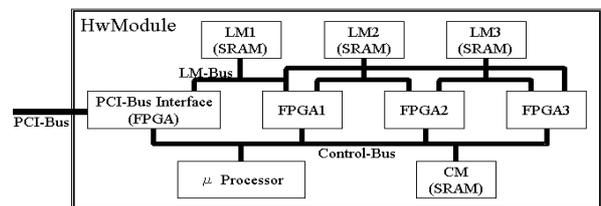
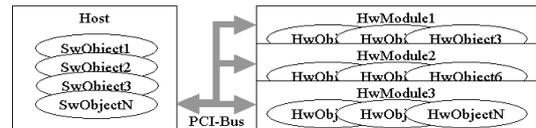


Fig. 1 Hw/Sw 混載システム

することが出来る。SwObject と同様にライブラリから HwObject を読み込み、ユーザー用のオブジェクトを派生することでアプリケーションを開発することが出来る。

2.2. HwObject と SwObject

HwObject

ビット処理、信号処理など回路が得意とする機能を割り当てる。目的とする処理は、仮想回路として HDL を用いて設計される。

SwObject

命令の解析や解釈のような回路には向かない複雑な条件分岐などの機能を割り当てる。従来の Sw と同様に C++言語を用いて設計される。

SwObject が HwObject にアクセスする際、MM にあるインターフェイスにアクセスすることで、Hw を意識する必要が無い。更に、このインターフェイスは、HwNet からの割り込みによる SwObject へのアクセスを行う手段も実装している。SwObject には、この HwObject のインターフェイスそのものが、HwObject に見える。複数の HwObject、SwObject を実装し、一連の処理を行う本システムのアプリケーションについても従来と変わらない記述が可能である。

HwObject と SwObject が大きく異なる点として、目的の処理に於けるメンバー関数の実装である。HwObject では、HwModule 上の FPGA に実装される HwNet により、目的の処理が行われる。よって、HwObject に於ける Host 上のメンバー関数は、対象の仮想回路と Host との結合を解決して動作させるための手続きのみである。このようにして、従来では Host 上のアプリケーションからは複雑であった Hw の扱いを、従来の C++言語による Sw の記述を用いて、容易に扱えるようにしている。

(学)東京農工大学

Tokyo University of Agriculture & Technology

3. HwObject の実装

図 2 に HwObject を用いたシステムを構成する際に必要な要素を示す。

3.1. HwNetDriver

HwNetDriver は、Hw 依存の Sw の手続きとして HwObject の提供者が記述を行い、HwNet と共にライブラリ化する。そして、アプリケーションの実行時に、HwManager にアタッチされ、HwNet がロードされる。

HwObject が HwNet を制御する基本的な手続きとして、生成、初期化、実行、削除があり、HwManager はこれらの手続きに HwNetDriver を介して行う。HwNetDriver は、FPGA や LM の配置やバス接続など、デバイス構成の異なるタイプ、或いは同じ構成の HwModule に於いて、デバイス特有の HwNet の制御手続きを提供するために必要となる。アプリケーションの目的に応じた HwNet 固有の処理、及び HwNet からの割り込みを HwObject に伝える処理を関数として実装する。

3.2. HwManager

Host と接続されている PCI-Bus には、複数の HwModule を実装することが可能である。この HwModule を管理するオブジェクトを HwManager と呼び、Host 上の API として実装する。HwManager は、HwNetDriver とは異なり、OS にアタッチされる通常のデバイス・ドライバである HwModuleDriver を介して HwModule にアクセスする。Hw 間のデータ通信は OS の管理下で HwModuleDriver が行う。要素として、HwObject と HwNet、HwNetDriver を結びつける管理テーブルを持ち、機能として、(1).HwModule の LM、FPGA の使用状況や空き状態などの Hw リソースの管理、(2).FPGA への HwNet のロードや破棄、(3).LM や HwNet にアクセスする手段を対応する Host 上の HwObject に提供する、(4).HwModule からの割り込みを受け付けて、HwNetDriver の割り込み処理関数を呼び出すことで、適切な処理を行う、などがある。

3.3. ObjectManager

HwObject や SwObject、Hw リソースの管理やアクセスをする HwManager は互いに協調動作を行う。これらオブジ

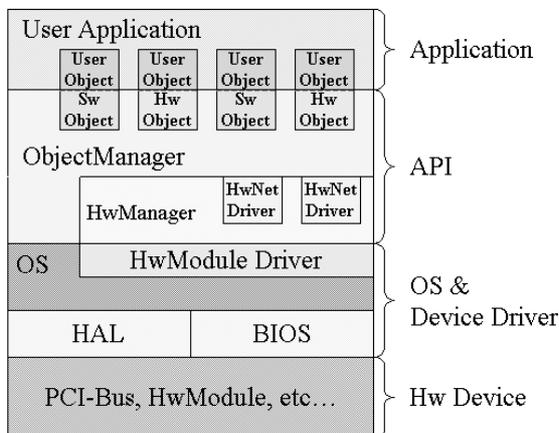


Fig. 2 Host に於けるシステムの構成要素

ェクトに対するプロセッサの割り当て、オブジェクト間の同期を含めた管理を ObjectManager が行い、API として提供する。ObjectManager は通常のアプリケーションとして、OS に管理される。要素として、システムの処理サイクル、オブジェクトの処理要求をイベントとして受け付けるイベントキュー、生成されたオブジェクトに ID を割り当てて管理を行うテーブル等を保持する。機能として、(1).処理要求のあるオブジェクトの実行、(2).システムサイクルの制御、(3).オブジェクトの生成と破壊、(4).HwManager の実行管理と HwObject の登録と破棄、(5).オブジェクトのメンバー関数を呼び出すためのイベント、オブジェクトの状態を示すイベントフラグ、イベントの伝達関係を保持するシグナルを用いたオブジェクト間の依存関係の構成、などがある。

4. むすび

本提案により、オブジェクト指向による再利用が期待でき、Hw へのアクセスは Sw 的に抽象化された記述で行えることから、Hw を用いたアプリケーションの設計・生産性の向上が図れる。

今後の課題として、HwModule 上の FPGA に動的に再構成可能な HwNet を実装し、HwNet のロードに関するオーバーヘッド、複数の HwObject、SwObject を実装した並列実行による性能についての検証が挙げられる。

付録

List.1 は、PCI-Bus のデータ転送のサイズ別に 1[byte]当たりのオーバーヘッドがどの程度になったかを示す。Host には、CPU を Pentium (600[MHz])、キャッシュ・メモリを 256[kbyte]、メイン・メモリを 128[Mbyte]実装した PC を用いた。

Direction	Byte	Word	Dword	Block-Dword		
	1	2	4	64	256	1024
Read	8.78	7.30	6.44	5.93	5.91	5.90
Write	8.82	7.21	6.51	5.87	5.87	5.87

List. 1 API から PCI-Bus 間のオーバーヘッド[μs]

参考文献

- [1] D. A. Buell, J. M. Arnold, and W. J. Kleinfelder ED., "SPLASH2: FPGAs in a Custom Computing Machine", IEEE Computer Society Press, Los Alamitos, CA (1996).
- [2] J.E.Vuillemin, et al., "Programmable Active Memories: Reconfigurable Systems Come of Age", IEEE Trans. onVLSI systems, Vol. 4, No. 1, pp. 56-69 (1996).
- [3]D.Davis, et al., "A Java Development and Runtime Environment for Reconfigurable Computing", Proc.5th Reconfigurable Architectures Workshop(RAW'98), Orlando, USA(March 1998)
- [4] 今中 晴記, 志賀 裕介, 工藤 健慈, 上野 貴史, 関根 優年, "仮想回路をカプセル化した HwObject を実行する HwModule", 第 15 回 回路とシステムワークショップ予稿集