3A-06

TCI 結合による計算機システム向け ビルディングブロック OS について

並木 美太郎† 小柴 篤史† 濱田 槙亮† 大城 研治† 天野 英晴‡
東京農工大学† 慶應義塾大学‡

１．はじめに

本稿では、チップを TCI 結合で 3 次元結合した計算機システム向けの OS についての基本設計と試作について述べる。本プロジェクトでは、SOTB による低電力なプロセッサ、主記憶、入出力などの個々のチップを、TCI により結合してシステム設計が容易な柔軟性の高い計算機システムを構築する手法について考察している[1]。

本研究では、

1) TCI により結合されたプロセッサ群の仮想化手法と効率的な実行機構
2) SOTB の省電力を活用する電力制御などを課題とする OS などのシステムソフトウェア

の研究を行った。本発表では、OS の基本設計と試作した機能を概観する。

２．ビルディングブロック OS の基本概念

本プロジェクトでは、チップを積層し、TCI でチップ間通信を行う。最終目標としては、チップを組み合わせることで望む計算機システムを構築できる方式を探ることである。このときに、ソフトウェア構成はどうあるべきかを探るのが、本研究の目的である。

計算機システムの構成要素を組み合わせるのと同様に、OS や応用プログラムも適宜組合わせることでソフトウェアシステムを構築したい。

最終目標としては、

- チップ上に小規模 OS(nano-kernel)と応用プログラムを搭載
- TCI 結合のチップ間通信により nano-kernel 群が、自律分散かつ協調しながら、応用プログラム制御や資源管理を行う
- このようなチップ上の nano-kernel の集合体をビルディングブロック OS と呼ぶ

このようなソフトウェアシステムの方式を明らかにすることが本研究の最大の目標である。

過去には、分散 OS、マイクロカーネルの研究などで、協調しあう計算機システム群で処理を行う方式は研究されてきた。本研究は、従来の計算機で行ってきた処理を、個々のチップに分割し、イーサネットの代わりに TCI などで接続する。チップレベルまで小型化されたビルディングブロック方式の中での、OS に必要な機能と構成を明らかにする。

ビルディングブロック方式による計算機システムの構成法については、ハードウェアも研究段階にある。そこで、ビルディングブロック OS も最初は比較的基本的な機能と構成を対象とした OS を研究する。

３．試作した積層チップによる計算機システムの構成

本研究で試作した計算機システムの構成を図１に示す。ビルディングブロックによる計算機システム構築はこのような構成に限定されないが、典型的な機能を含んだものとした。

TCI 結合された積層チップは、各種アクセラレータ、メモリを想定しているが、全体の制御に CPU は不可欠である。TCI 結合された積層チップは、Geyser のアドレス空間の一部に配置される。したがって、積層されるチップは、Geyser からは外部バスに接続された装置と同様に、主記憶または I/O として読書き可能である。

CPU については、MIPS R3000 互換の Geyser を用い、同時にチップ上の内部バスに、DMAC(DMA Controller)、TCI 結合のルータ I/F、また外部バスへの I/F が接続されている。なお外部バス I/F は、今回の試作では FGPA で AXI バスに変換しており、AXI_GP のマスターとして動作することから、Zynq や AXI スレーブデバイスを利用できる。

CPU と DMAC はバスマスターとして、TCI 結合された各種チップと外部バスに接続された装置を操作できる。アクセラレータについては、外部バスに接続された入出力装置や主記憶とデータ転送が必要となるが、CPU によるプログラム転送、DMAC による DMA 転送のいずれも可能であり、並列動作も可能とした。なお、CPU からの転送については、データキャッシュのライン上のデータをそのまま TCI のルータに送る命令を追加した。

Basic Design of Building Block OS for Computer Systems using TCI
†Mitaaro Namiki, Atsushi Koshiba, Shinsuke Hamada, Kenji Ohshiro, Tokyo University of Agriculture and Technology
‡Hideharu Amano, Keio University

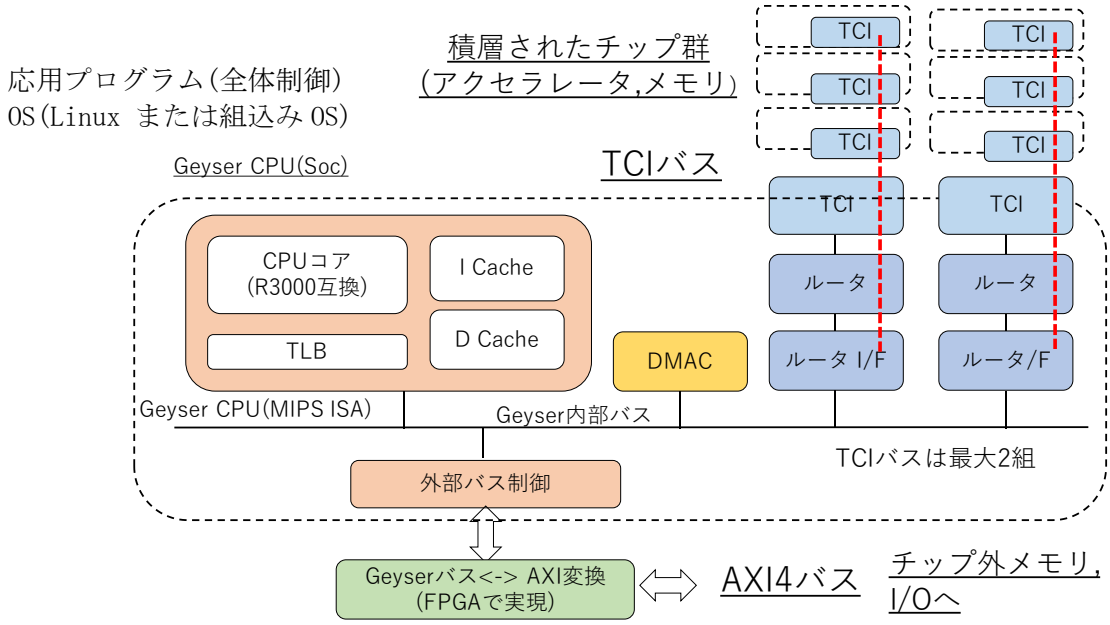

図 1 試作した計算機システムの構成

4. OS の課題と機能

このように、ビルディングブロックによる計算機では積層されたアクセラレータなどのチップを、通常の主記憶と同様に利用できる。

実際には次の処理が必要となる。

1) 積層されたチップの種別などのコンフィグレーション管理
2) 各種アクセラレータの効率の良い実行制御
   アクセラレータの実行開始と終了検出
3) データ転送管理
   アクセラレータのバッファ管理、DMAC 制御
4) TCI 通信路管理：エラー率は低いがゼロではないので、TCI の通信エラー検出と再送。
5) 実行効率の最適化
   データ転送オーバヘッドの隠蔽とそのパラメータ決定
6) 省電力制御：SOTB によるチップでの適切なバイアス制御

上記 1)～6)の処理を、試作版ビルディングブロック OS で行う。特に、積層チップの持つ機能をモジュール化するような OS アーキテクチャを考察する。

現時点では、アクセラレータを積層するので、アクセラレータの仮想化を目標とする。積層されるアクセラレータを OpenCL により仮想化し、多様な積層チップに共通のプログラミング環境を提供する。アクセラレータのプログラム、データ転送、実効制御は OpenCL の API として提供される。

OS カーネルは OpenCL の実行基盤を提供する。特に、アクセラレータの処理能力を有効活用するために、OS カーネルのオーバヘッドを排除し、可能な限りユーザレベルであるプロセスでアクセラレータを制御する。アクセラレータへのコマンド発行などはユーザプロセスの OpenCL ライブラリ内で行う。ビルディングブロック OS は、アクセラレータ処理について、割込みの検知、仮想アドレス空間とアクセラレータの実アドレス空間の対応付けなど最低限の処理だけ行う。アクセラレータのローカルメモリや制御レジスタ群は実アドレスに配置されているが、OS だけでなく、ユーザプロセスからも読書きできるように仮想アドレス空間にマップされる。

アクセラレータでストリーム処理などのパイプライン処理を行うことで実行効率の向上を可能とする。OpenCL に対して、パイプライン処理を書けるようにしている。詳細は別の稿を参照されたい。現在、Geyser で稼動している Linux、また、組込み OS として Toppers を移植している。

5. 終りに

3 次元積層によるビルディングブロック向け計算機の OS について紹介した。評価用システム、パイプライン並列ライブラリ、電力評価の環境については、本全国大会の別稿にて説明する。

参考文献

[1] 天野ほか：ビルディングブロック型計算システムプロジェクトの報告，情報処理学会第 80 回全国大会(2018)