

Zynq 搭載 MTCA. 4 ボードへの Linux 及び EPICS の組み込み

LINUX AND EPICS EMBEDDING ON MTCA.4 FPGA(ZYNQ) BOARD

出口 久城^{#,A)}, 岩城 孝志^{A)}, 林 和孝^{A)}, 松本 隆太郎^{A)}, 漁師 雅次^{A)}

Hisakuni Deguchi^{#,A)}, Takashi Iwaki^{A)}, Kazutaka Hayashi^{A)}, Ryutaro Matsumoto^{A)}, Masatsugu Ryoshi^{A)}

^{A)} Mitsubishi Electric TOKKI System Corp.

Abstract

Our present FPGA board has been adopted as many accelerator apparatus, SuperKEKB LLRF, cERL LLRF, and so on since 2009. The board uses the Xilinx's "Virtex5" FPGA (PowerPC440 included) in MTCA.0 standard. The user friendly interface (EPICS IOC) was made by Linux embedding on PowerPC. In recent years, embedding devices and technologies have developed greatly by smartphones. The Linux-OS on ARM-CPU becomes a new mainstream in embedding systems. A new FPGA board has been developed, which is using the Xilinx's "Zynq" (ARM Cortex-A9 included) and conforming to the new MTCA.4 (MTCA for Physics) standard. The Zynq evaluation kit and documents (including application notes) were helpful for embedding. Linux and EPICS embedding work has been done on this new Zynq board. The Cortex-A9 computing ability is drastically improved with power-saving. Since each of CPU and user-logic in Zynq had been independent, the remote update using a network became very easy. The new FPGA board embedding Linux and EPICS will be the monitor and controller (LLRF, BPM, and so on) board for the next generation accelerators.

1. はじめに

STF, SuperKEKB、及び cERL の LLRF システムには、通信系のプラットフォーム規格である Micro Telecommunications Computing Architecture 0 (MTCA.0) の FPGA ボードが採用されている^[1]。このボードでは、FPGA によるリアルタイム処理を行っており、更に、FPGA 内蔵プロセッサを利用し、Linux 及び EPICS を組み込んでおり、柔軟なユーザインタフェースを実現している。すなわち、FPGA ボード単体で一つの EPICS Input Output Controller (IOC) を形成している。この柔軟な動作環境が加速器制御に適応しているため、先に述べた加速器において、様々な用途（フィードバック制御、チューナ制御、高速インターロック制御、リファレンス位相制御、ビーム位置モニタ）に採用されている。この FPGA ボードへの Linux 及び EPICS 組み込みは、SuperKEKB 制御グループの多大なる協力の下に成されたものである^[2]。

現行の（前述の）FPGA ボードは当面現役で使用されるが、一方、将来の加速器を見据えて、更なる高度かつ高速な処理、拡張性及び柔軟性も求められてきている。現行のボードでは、組み込み Linux には "Wind River Linux 2.0" を採用しているが、2012 年に Wind River Linux のライセンス形態が変更され（大量生産を前提としたライセンス形態となり 1 ライセンス当りの価格が上昇）、該当バージョン（2.0）のサポートも打ち切りとなった。従って、大量生産しない開発者にとっては、（新規開発するための、または既存アプリケーションをリビルドするための）ライセンス更新（または購入）が困難となっている（ただし、Wind River Linux 2.0 を組み込み済みのボードを継続使用することは可能）。この 3 年間は Wind River Linux 2.0 を従来のライセンス形態のまま（価格変更無し）で使用できるよう、メーカーと調整してきたが、これも永続できるものではない。また、FPGA とその内蔵プロセッサの処理能力

や省消費電力性能は、現行のボード開発時（2009 年）と比べると飛躍的に伸びており、2012 年には、近年のスマートフォンに代表される組み込み機器で実績のある高性能な ARM プロセッサ Cortex-A9 を内蔵した FPGA "Zynq"（Xilinx 社製）が登場した。そこで今回、この Zynq と、近年加速器制御で注目されている MTCA.4 規格を組み合わせた新たな FPGA ボードを開発した（Figure 1）^[3]。この新しい MTCA.4 FPGA ボードについても、現行のボードと同様に単体で一つの EPICS IOC となるよう、Linux 及び EPICS の組み込みを行なった。実際の動作環境としては、MTCA.4 FPGA ボードを MTCA.4 規格準拠シェルフへ挿入し電源供給され、MTCA Carrier Hub（MCH）を経由した Ethernet 通信となる（Figure 2）^[3]。

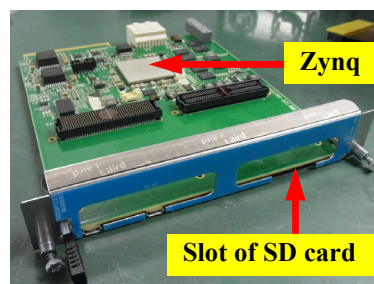


Figure 1: MTCA.4 FPGA(Zynq) Board

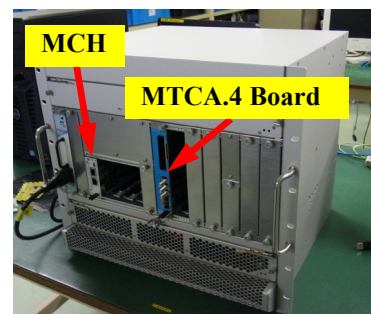


Figure 2: Installing the board into MTCA.4 shelf

[#] hi-deguchi@west.melos.co.jp

本稿では、Zynq 搭載 MTCA.4 ボードへの Linux 及び EPICS 組込みにおける実行内容、成果及び今後の課題について報告する。

2. 開発環境の構築

今回の Linux 及び EPICS の組込みは、Zynq 内蔵の ARM プロセッサ Cortex-A9 がターゲット CPU となる。従って、ARM 用クロス開発環境を構築した。

2.1 Linux

現行のボードの FPGA には "Virtex5" (Xilinx 社製) を採用しており、Virtex5 との組合せにおけるサポートの充実度により、Wind River Linux を採用していたが、今回の Linux 組込みにおいては Wind River Linux を使用せず、別の組込み Linux を使用することが求められた。理由は、1. 節で述べたように、新規開発のためのライセンス購入が困難なためであり、かつ ARM+Linux の場合、多くの事例があるためである。そこで今回採用したのが、Xilinx 社が提供している組込み Linux^[4] (以降、Xilinx Linux と仮称する) である。Xilinx Linux (カーネルバージョン: 3.9.0-xilinx) には以下の 2 つの利点がある。

- ライセンスフリー (無償提供) である。
- マニュアルが豊富である。

欠点としては、サポートが無いことであるが、今回の開発においては、豊富なマニュアルで十分カバー可能であった。

Xilinx Linux はカーネルソース (デバイスツリー含む) である。Linux 組込みには他に、ブートローダ、ルートファイルシステムが必要である。これらも共に Xilinx 社が提供しており、無償で使用することが可能である (ブートローダは u-boot である)。ビルドツールチェーンについては、Xilinx Linux のマニュアルに従い、"PetaLinux"^[5]を使用した。PetaLinux はカーネルソースだけでなく、ブートローダ (u-boot) やルートファイルシステム、デバイス・ドライバの開発環境まで含まれる統合的な組込み Linux 開発環境である。今回は、その中のツールチェーンのみを使用した。各々 Web からダウンロードし、クロス開発環境用の Linux PC へインストールする。

2.2 EPICS

EPICS の ARM 用ビルドツールチェーンには、Mentor Graphics 社の "Sourcery CodeBench Lite 2013.05-24 (gcc 4.7.3)"^[6] (以降、CodeBench と呼称する) を使用した。Xilinx Linux 同様、ライセンスフリーのビルドツールチェーンである。EPICS の make 環境においては、CodeBench のディレクトリ構成が標準であり、メイクファイルの修正が容易であったため、CodeBench を使用した。CodeBench についても、Web からダウンロードし、クロス開発環境用の Linux PC へインストールする。

2.3 デバイス・ドライバ

EPICS IOC としての機能は、現行のボードから変更が無い。すなわち、FPGA の処理に対してパラメータ設定やモニタの必要がある。これは、Linux カーネルレベルでのデバイス・ドライバ (FPGA のレジスタ・アクセス) の実装となる^[7]。基本機能は変更が無い (レジスタの物理アドレスが変更されるのみ) ため、現行のボードでのソースコードを再利用し、それを ARM 用にビルドした。ビルドツールチェーンには、カーネルの make 環境と合わせるため、PetaLinux を使用した。このデバイス・ドライバは、以降、レジスタ・アクセス・ドライバと呼称する。

3. Linux ポーティング

3.1 起動の仕組み

現行のボードは単体では Linux を起動できず、起動のために専用のサーバ PC (サービス) が必要であった。すなわち、以下の 2 点についてである。

- u-boot が TFTP により Ethernet 経由で Linux カーネルをダウンロードするため、TFTP サーバが必要であった。
- Linux カーネルがルートファイルシステムを NFS マウントするため、NFS サーバが必要であった。

新しいボードにおいては、上記内容を改善し、ボード単体で Linux 起動までを可能とした。ボード単体で Linux まで起動させるためには、u-boot、Linux カーネル及びルートファイルシステムをボード本体 (不揮発性のメモリ等) へ書き込む必要がある。今回の組込みにおいては SD カード (Figure 1) へそれらを書き込み、SD カードブートの仕組みを実装した。Flash メモリ等への書き込みとは異なり、手順が非常に簡単であり (単純なファイルコピー)、作業工数を短縮できることも大きな利点である。一方、EPICS のアプリケーション・レイヤーは、頻繁に変更することとなるため SD カードへ書き込むことは得策ではない。従って、EPICS については NFS マウントし起動することとした (Figure 3)。

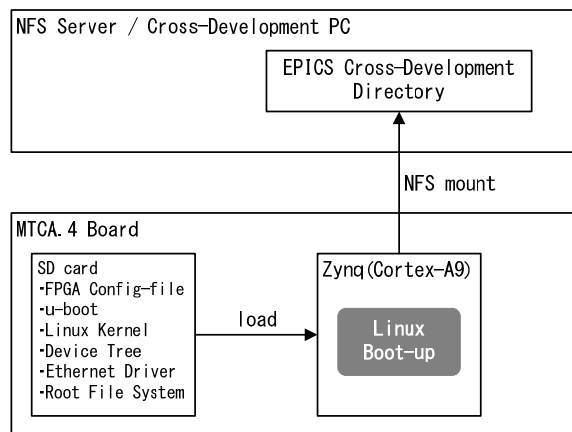


Figure 3: Boot-up configuration of MTCA.4 Board

また、ボードへ電源供給してから EPICS 起動ま

での一連の動作については、自動的にこなすものである（現行のボードと変わらない）。自動起動の仕組みについては、ルートファイルシステムを編集し実装した。実装内容としては、以下の5つである。

- Linux 自動ログイン
- Ethernet アクティブ（詳細は 3.2 小節を参照）及び IP アドレスの決定
- レジスタ・アクセス・ドライバの自動インストール
- 自動 NFS マウント（EPICS 開発ディレクトリを NFS マウントする）
- EPICS（起動スクリプト）自動起動

ソフトウェアとしての、SD カードブートの実現方法は、Xilinx 社提供のアプリケーション・ノート（リファレンス・デザイン）”XAPP1082”^[8]を利用した。ハードウェアとしては、Zynq の開発評価ボードである ”ZC706” をリファレンスとしてボード設計を行ない、SD カードブート可能な回路構成とした。

3.2 Ethernet ドライバ

ここで言う Ethernet ドライバとは、デバイス（ハードウェア）に直結するものであり、u-boot におけるドライバ及び Linux カーネルレベルのドライバである。現行のボードでは、u-boot における Ethernet ドライバ（TFTP 実行のために必要）と Linux カーネルレベルの Ethernet ドライバを実装している。一方、新しいボードは SD カードブートであるため、u-boot においては Ethernet 機能が不要、すなわち u-boot における Ethernet ドライバが不要であり、Linux カーネルレベルの Ethernet ドライバのみの実装となる。これは、開発工数の大幅な削減となる。ドライバはデバイスとのインターフェースとなるため、デバイスの回路構成により、実装もそれに合わせて変更する必要がある。新しいボードの回路構成は Extended Multiplexed I/O（EMIO）経由 1000BASE-X（Figure 4）^[8]であるため、EMIO 経由 1000BASE-X Ethernet ドライバを実装した。

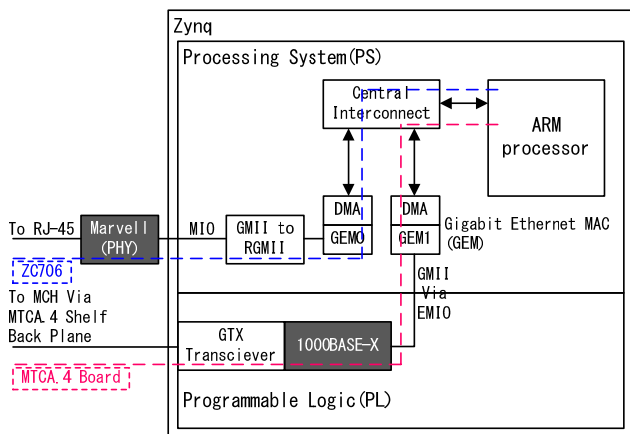


Figure 4: Zynq Ethernet interface

EMIO 経由 1000BASE-X Ethernet ドライバの実装方法は、SD カードブートと同じく、XAPP1082 を

利用した。XAPP1082 では、Linux カーネルのローダブル・モジュールとして、EMIO 経由 1000BASE-X Ethernet ドライバが実装されている。XAPP1082 には XAPP1082 環境のパッチファイルが含まれており、そのパッチを Xilinx Linux に適用しビルドすることにより、EMIO 経由 1000BASE-X Ethernet ドライバが利用可能となる。このドライバを Linux 起動後にインストールすることにより、Ethernet がアクティブとなる。また、現行のボードでは、CPU-Media Access Controller (MAC) 間の Ethernet データのインターフェースが First In First Out (FIFO) 形式であった。新しいボードでは、このインターフェースを Direct Memory Access (DMA) 転送形式 (Figure 4) に改善し、性能向上を図った。

4. 成果

4.1 起動

ここで改めて起動の仕組みを整理すると、特徴としては以下の2つにまとめられる。

- Linux 起動までを MTCA.4 ボード単体で自動的にこなす。
- EPICS のみを NFS マウントにより動作させる（アプリケーション変更時にフレキシブルな対応が可能）。

ここまで述べた内容を MTCA.4 ボードへインプリメントし、EPICS が起動することを確認できた。

4.2 設定とモニタ

EPICS IOC の機能確認として、CSS-BOY^[9]を用いて設定とモニタを行なった。アプリケーションとしてフィードバック制御の FPGA ロジックと EPICS IOC を実装し、各 EPICS レコードを CSS-BOY で実装した Operation Interface (OPI) 上から設定、モニタできることを確認した (Figure 5)。

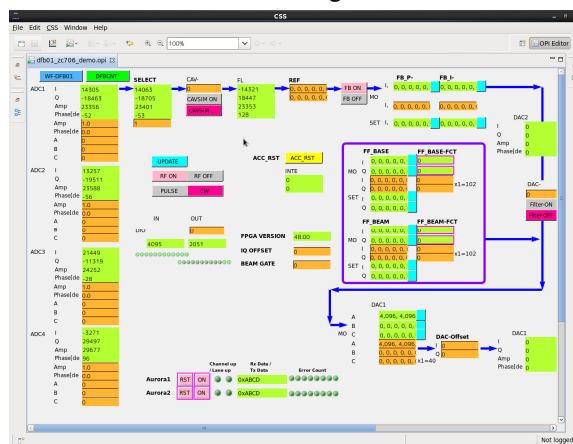


Figure 5: OPI on CSS-BOY

4.3 大容量データの転送

現行のボードでは、EPICS (CPU) がボード上のメモリから 256MBytes のデータを取得し、それを Ethernet 経由で転送し、転送先にてファイル出力す

る機能を実装していた。新しいボードにおいても同様の機能を実装し、転送性能の比較を行なった。新しいボードは、現行のボードの約 9 倍の性能という結果を得られた (Table 1)。20 秒という結果についても、運用に耐えうる性能である。

Table 1: Performance of transferring large-volume data

Kind of boards	Time necessary to transfer 256M Bytes data (typical)
Present board @SuperKEKB	180 seconds
MTCA.4 board	20 seconds

性能向上の要因として、以下の 3 つが挙げられる。

- CPU がボード上のメモリからデータを取得するインタフェース (CPU の内部バス) に、Advanced eXtensible Interface 4 (AXI-4) バスを使用した^[3]。現行のボードでは Processor Local Bus (PLB) を使用していた。
- CPU-MAC 間の Ethernet データのインタフェースを DMA 転送形式とした (Figure 4)。現行のボードでは FIFO 形式であった。
- プロセッサが高性能な ARM となった。現行のボードでは、PowerPC440 (Virtex5 内蔵) であった。

4.4 応用

新しいボードの応用・今後の課題としては、やはり、ARM プロセッサ Cortex-A9 の能力を発揮させるアプリケーション開発である。例えば、現行のボードの EPICS では、4096 点の I,Q 及び振幅位相データを Waveform レコードとしてモニタしているが、新しいボードでは、これに加えて FFT 処理を EPICS で行なうことも考えられる。また、CW 加速器の振幅位相のフィードバック制御は FPGA ロジックで行なっているが、短いパルス運転の加速器では、パルス内での制御は FPGA ロジックでも間に合わないため、パルス間でのフィードバック制御を上位システムで行なっている。これをボード上の EPICS で行ない、よりフレキシブルな環境を構築できる可能性がある。

5. リモートアップデート

最後に、Zynq の特性を活かした機能について述べる。Zynq は、Processing System (PS) と呼ばれるシステム (CPU を含めたハードウェア) 部分と Programmable Logic (PL) と呼ばれるユーザ・ロジック部分とに完全に分離された構造となっている^[10]。この特性により、CPU が起動した状態のまま、FPGA ロジックの書き換えが可能なのである。更に、SD カードブート機能と組み合わせることにより、Ethernet 経由で FPGA ロジックを簡単に書き換えることができる。FPGA ロジックはファイルとして SD カードに保存されているため、ボードへ telnet 等でログインし、ファイルシステムにマウントされ

た SD カードへ cp コマンドで FPGA ロジックのファイルをコピーするだけでよい。あとは、ログインしたシェル上で reboot コマンドを実行し、Linux を再起動すれば、書き換えた FPGA ロジックが起動する。現行のボード (Virtex5) は、Zynq のような構造ではなく SD カードブートでないため、FPGA ロジックの書き換えのためには専用のインタフェース (プログラム) を準備しなければならないが、新しいボードは、Linux の既存機能のみで、FPGA の書き換えから再起動まで全てリモートで行なうことが可能なのである。

6. まとめ

今回、Xilinx 社製の FPGA である Zynq を搭載した MTCA.4 ボードを開発し、そこへ Linux 及び EPICS を組み込んだ。ターゲット CPU は、Zynq 内蔵の ARM プロセッサ Cortex-A9 である。組み込み Linux には、Xilinx Linux (Xilinx 社無償提供) を使用した。Ethernet ドライバの実装には XAPP1082 (Xilinx 社無償提供のリファレンス・デザイン) を利用した。Linux 起動に必要なデータを、MTCA.4 ボードに実装した SD カードへ書き込み、Linux 起動までを MTCA.4 ボード単体で自動的に行なえるにした。EPICS のみ NFS マウントとすることにより、EPICS のアプリケーション・レイヤー変更時にフレキシブルな対応を可能とした。EPICS IOC としての機能は、CSS-BOY で実装した OPI により確認した。256MBytes のデータ転送性能は現行のボードの約 9 倍という結果が得られ、運用に耐えうるものであることを確認できた。また、Zynq の特性を活かし、Ethernet 経由による FPGA ロジックの書き換え (リモートアップデート) を実現した。

参考文献

- [1] M. Ryoshi et al., “LLRF Board in Micro-TCA Platform”, Proceedings of the 7th Annual Meeting of Particle Accelerator Society of Japan, Himeji, Aug., 2010.
- [2] J. Odagiri et al., “Fully Embedded EPICS-Based Control of Low Level RF System for SuperKEKB”, Proceedings of the IPAC’10, Kyoto, Japan, May, 2010.
- [3] M. Ryoshi et al., “MTCA.4 FPGA(Zynq) Board”, in this meeting.
- [4] <http://www.wiki.xilinx.com/Getting+Started>
- [5] <http://www.wiki.xilinx.com/PetaLinux>
- [6] <http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/>
- [7] H. Deguchi et al., “EPICS Embedding for SuperKEKB LLRF System”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Aug., 2011.
- [8] http://www.xilinx.com/support/documentation/application_notes/xapp1082-zynq-eth.pdf
- [9] <http://www-linac.kek.jp/cont/epics/css/>
- [10] http://www.xilinx.com/support/documentation/user_

guides/ug585-Zynq-7000-TRM.pdf