Cell/B.E. Based Robot Controller

Matthias Fritsch
IBM Deutschland Entwicklung GmbH
Schoenaicherstr. 220 71034 Boeblingen, Germany
Matthias.Fritsch@de.ibm.com

Ronny Klauck
IBM Deutschland Entwicklung GmbH
Schoenaicherstr. 220 71034 Boeblingen, Germany
ronny.klauck@gmail.com

Marc Tritschler
IBM Deutschland Entwicklung GmbH
Schoenaicherstr. 220 71034 Boeblingen, Germany
mtritsch@de.ibm.com

Abstract

The processor Cell Broadband Engine* (Cell/B.E.) is used inside the Playstation 3*. The Cell/B.E. is a multi-core processor with heterogenous cores. Each core has its own memory. But all cores can communicate with each other using a high speed bus.

State of the art controllers for automation technology use Digital Signal Processors (DSPs), Field Programmable Gate Arrays (FPGAs) or PC-like hardware. A combination of PC-like hardware using DSPs or FPGAs as accelerators is a good design for building up controllers for automation. A Cell/B.E. can be used for a similar design of robot controllers. The Cell/B.E. has advantages compared to the combined design of PC-like hardware and DSPs or FPGAs. It is discussed how a Cell/B.E. can be used for real time applications.

The Cell/B.E. Based Robot Controller Project has the goal to implement a functional prototype of a robot controller using a Cell/B.E. based computer. A controller that is able to control up to 15 robot arms simultaneously is already implemented. Performance measurements show that the Cell/B.E. works fine for automation purposes.

The Cell/B.E. offers features that couldn't be realized with state of the art controllers for automation technology.

1 Introduction

The Cell/B.E. is used as processor inside the Playstation 3. It is known to have a very good calculation performance compared to other CPUs available. Thus it is used also for high performance computing. This paper discusses another field of application for the Cell/B.E. Following question is answered: Can the Cell/B.E. be used for embedded systems and automation technology?

2 Cell/B.E. Fundamentals

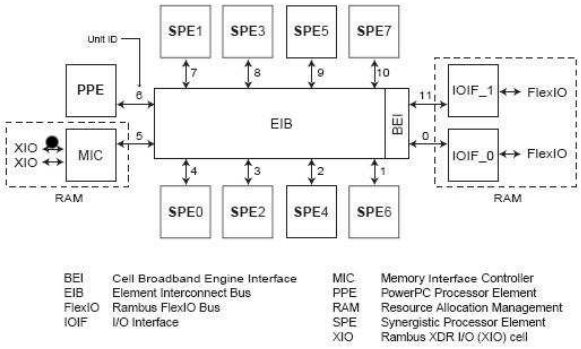

FIGURE 1: *Block Diagram of Cell/B.E. [1]*

The Cell/B.E. is used in the Playstation 3. It is a multi-core processor with nine cores. These cores are designed heterogenous. One Core is a 64 Bit PowerPC* that is used to run Linux on it. This core is called PowerPC Processing Element (PPE). The remaining eight cores are indented to run single threads on them. These cores are called Synergistic Processing Elements (SPEs). The Cell/B.E.s used in the IBM BladeCenter QS21* have a core frequency of 3,2 GHz.

2.1 PPE

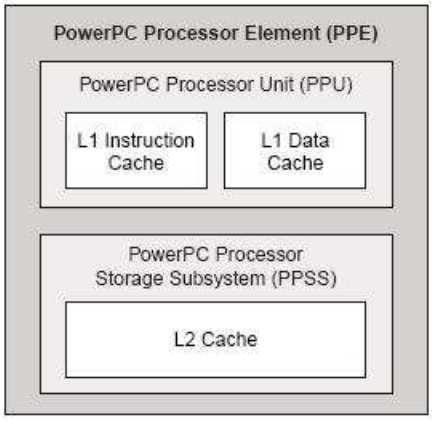

FIGURE 2: *Block Diagram of PPE [1]*

The PPE is the processor core executing the operating system. It is used to start threads in the SPEs, too. The PPE has full access to the main memory. The PPE cannot access the SPE's memory directly. Data is transfered using DMA, Mailboxes or Signal Notification. Short, each core has its own memory that can be accessed from other cores using DMA or related techniques. There is no shared memory for all cores. [1]

2.2 SPE

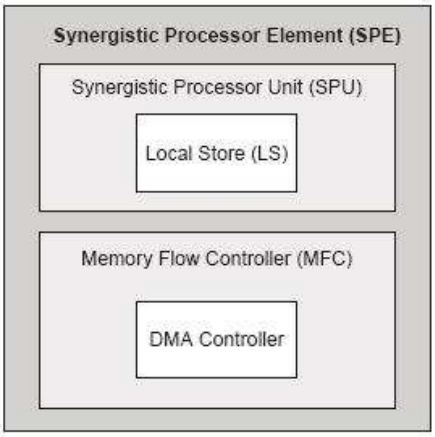

FIGURE 3: *Block Diagram of SPE [1]*

Each SPE has its own memory (Local Store). Program code and data are processed there. Each SPE can execute code independently for all other SPEs. The size of the local store is 256 kByte for each SPE. Each SPE has an own DMA controller. The SPEs can handle multiple operations per cycle (vector/SIMD).

2.3 EIB

All inter-core communication is accomplished through a high performance bus called Element Interconnect Bus (EIB). Since all cores have their own DMA controller communication can be handled very fast between all participants.

3 Cell/B.E.'s Advantages for Automation

Controllers for automation have several requirements. Real time capability is one of the main requirements. Other requirements are ease of use for customers (Programming interface, Human Machine Interface) and reliability. Real time can be handled very good using Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs). They aren't good to adapt human machine interfaces (HMI). The resources are limited. Another approach is to use PC-like hardware and mainstream operating systems. There you have to use real time extensions to reach the real time requirements. Attaching a HMI to a PC is easy because of the resources of the computer.

3.1 PC and DSP/FPGA-like Controller Design

A combined design approach of PC and DPS/FPGA design would be a very good choice. The operating systems running on the PC processor can be used for the HMI. The real time controller can be realized on the DSPs or FPGAs. But still there are disadvantages. The PC processor and the DSPs/FPGAs are programmed in different programming languages. Different technologies are combined to a complex system of processors. There are at least two tool chains needed to implement the controller software.

3.2 Cell/B.E.-like Controller Design

If we now take the Cell/B.E. and apply the previous discussed controller design on it then the PPE handles the operation system and the HMI. The real time controller parts are handled by the SPEs instead of DSPs or FPGAs. The complex system of processors is reduced to single chip with multiple cores. Only one tool chain is needed to implement the controller software. Linux-like programming can be used on PPE and SPEs (C/C++).

4 Real Time Behavior

The real time behavior of the Cell/B.E. is not different to any other processors used to build up PCs.

4.1 PPE and Real Time

You need to run a real time operating system on it to have real time capabilities. In case of the Cell/B.E. this would be a real time Linux*.

4.2 SPE and Real Time

The SPEs have their own local store. They can execute code independently from other cores. The SPE does not have an operating system running on it. The SPE can be compared to the DSP or FPGA used for state of the art robot controllers. By this the SPE can be seen as an acceleration processor for the PPE. The PPE starts the thread on the SPE. Once the SPE executes the code it is nearly independent of the PPE. That is the fact that makes the Cell/B.E. perfectly usable for real time applications like robot controllers.

4.3 Restrictions

But there are restrictions that have to be handled so that the SPE can be used for hard real time applications.

- SPE can receive interrupts
- SPE must access the PPE to handle main memory pages (once per page)
- PPE handles input/output
- System calls are handled from PPE not from SPE even if you issue it on SPE (e.g. printf)

All of these restrictions can be weakened or eliminated by an appropriate software design.

5 Project: Cell/B.E. Based Robot Controller

Among the range of applications that can be handled by the Cell/B.E. in automation technology we chose the robotic.

5.1 Goal

Goal of this project is to implement a functional prototype of a robot controller using a Cell/B.E. based computer. The show case implemented with that functional robot controller prototype shows features that are not realizable with state of the art in robot controllers.

Among these features are:

- One controller for multiple robot arms
- Collision control
- Adaptive continuous path movement (Integrating feedback)
- Balancing objects while moving
- Ability to control super fast robot arms (e.g. populate printed boards, low latency)

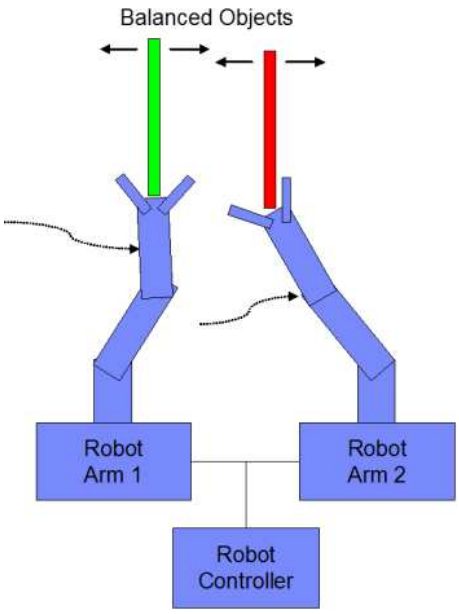

FIGURE 4: *Block Diagram - Robot Arms Balancing Objects*

5.2 Hardware

The Cell/B.E. based computer is an IBM BladeCenter QS21 Prototype board (Cell/B.E. Blade Server, QS21). QS21 has two Cell/B.E.s on board and two GByte main memory [2].

FIGURE 5: *IBM BladeCenter QS21 (Cell/B.E. Blade)*

The robot arms are attached using a serial connection. For the very early development two educational robot arms are used. These robot arms are built up using standard model making servo engines.

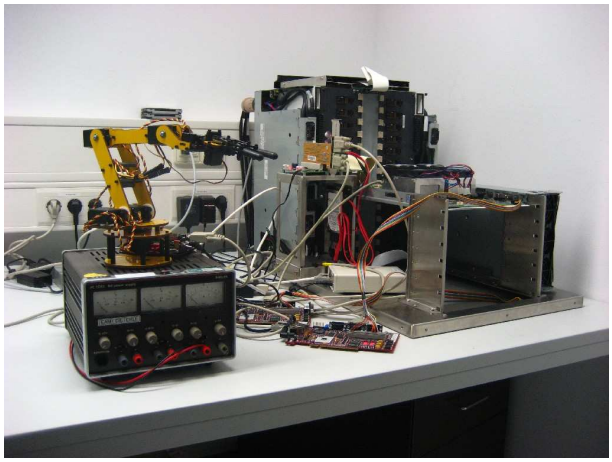

FIGURE 6: *Robot arm and Opened IBM BladeCenter H*

5.3 Status

Currently it is possible to control up to 15 robot arms with four degrees of freedom and continuous path movement using one QS21.

5.4 Performance Measurements

The first performance measurements show that 450 nanoseconds are needed for a SPE to calculate the next iteration step for the robot arm movement. Good state of the art robot controllers have 1000000 nanoseconds cycle times.

5.5 Open Issues

To be competitive to the market six degrees of freedom are needed to be controlled. A controller software that is able to achieve that is currently under development. Additional features like collision control will take benefit of the calculation performance of the Cell/B.E.

6 Conclusions

State of the art robot controller designs show that a combination of PC-like computer combined with DSP or FPGA accelerator processors are a good method to build up controller hardware. Instead of building up complex systems having different processor types a single chip design can be made by using a Cell/B.E. The Cell/B.E. offers additional advantages like having one tool chain to program or its using the calculation performance for new features. When using the SPEs for real time execution, there is not necessarily a real time Linux operating system needed to build up real time embedded systems.

References

[1] ,*IBM. Cell Broadband Engine Programming Handbook, Version 1.0 edition, April 2006.*

[2] ,*IBM BladeCenter QS21 - http://www-03.ibm.com/systems/bladecenter/qs21*

[3] ,*Cell/B.E. Technology - http://www-03.ibm.com/technology/cell*

[4] ,*Cell/B.E. Resource Center - http://www-128.ibm.com/developerworks/power/cell*

[5] ,*Cell/B.E. Documentation on Power.org - http://www.power.org/resources/devcorner/cellcorner*