Available online at www.sciencedirect.com
ScienceDirect
Fusion Engineering and Design 83 (2008) 215–219
Fusion Engineering and Design
www.elsevier.com/locate/fusengdes

The integration of the new advanced digital plasma control system in TCV

N. Cruz [a,*], A.P. Rodrigues [a], B. Santos [a], C.A.F. Varandas [a], B.P. Duval [b], J.-M. Moret [b], J. Berrino [b], Y. Martin [b], X. Llobet [b]

[a] *CFN-IST, Centro de Fusão Nuclear, Associação Euratom-IST, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*
[b] *CRPP-EPFL, Centre de Recherches en Physique des Plasmas, Association Euratom-Confédération Suisse, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

Available online 4 January 2008

Abstract

A new advanced digital plasma control system was developed aiming at a larger flexibility and enhanced performance controlling the TCV plasma shape, position, current and density. The system is a complex grid of 32 acquisition processing and control channels (APCCs) that can go up to 25 μs for the slow control cycle and a grid of 4 APCCs that can go up to 5 μs for the fast control cycle. Each APCC is composed of analogue input, digital signal processor (DSP) and analogue output. All APCCs are interconnected broadcasting data in each cycle.

A suitable interface between the system and the TCV plant enabling the complete integration into the existing schema running and controlling the TCV was designed and implemented. The system state-machine is presented. The advantages of using a structured integrated tool such as MDSPlus is evaluated, as well as the results on the final performance, usability and stability of the system.

Keywords: MDSPlus; Plasma control; Digital signal processor; Parallel processing; System integration

1. Introduction

The new advanced digital plasma control system (APCS) is a complex grid of 32 acquisition processing and control channels (APCCs) for the slow control cycle (25 μs) and a grid of 4 APCCs for the fast control cycle (5 μs) aiming at a larger flexibility and enhanced performance controlling the TCV plasma shape, position, current and density.

Each APCC has one analogue input channel (ADC), one analogue output channel (DAC) and one digital signal processor (DSP) connected to three other APCCs, to the VME bus and capable of connecting to APCCs on other hardware modules [1].

Digital signal processors (DSPs) and field programmable gate arrays (FPGAs) have been employed in a variety of systems due to its high processing capability together with low power consumption and low price. To increase the performance and cope with distributed processing systems (like the APCS), complex parallel systems have been developed in several scientific areas [2].

Interface to such complex systems is a key issue on its usability, integration and algorithm development speed. Moreover, in fusion research experiments, an effort has been made to structure and ease the integration of new systems not only for data acquisition [3], but also to control systems [4].

A suitable interface between the APCS system and the TCV plant control software enabling the complete integration into the existing schema running and controlling the TCV was designed and implemented. This interface enables the abstraction from the hardware implementation when building the physics control algorithm.

2. Host software overview

Fig. 1 depicts the host software structure.

The host software is built on top of the Linux VME device driver [5]. The hardware access is made using a shared library built for the module (RTPro library) and a set of commands that can be called from the Linux command line (Coreutils). Then a specific shared library was built to deal with the APCS system (DPCS library) that makes calls to the RTPro library but also uses some system commands from the Coreutils. On top

* Corresponding author. Tel.: +351 239410108; fax: +351 239829158.
E-mail address: nunocruz@lei.fis.uc.pt (N. Cruz).

0920-3796/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2007.11.005

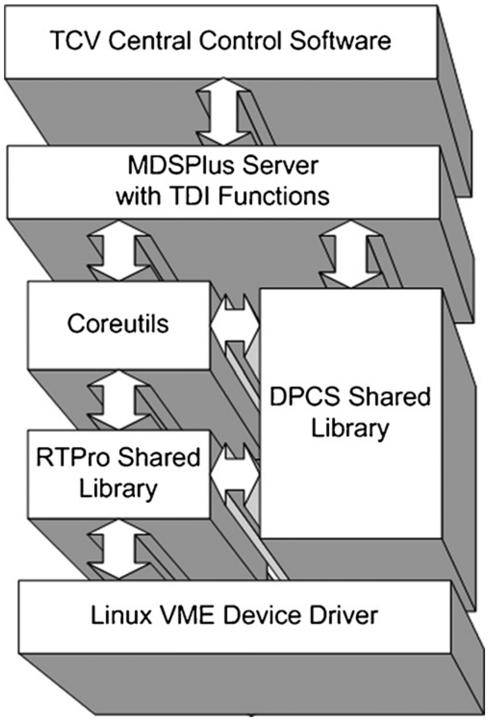

Fig. 1. The APCS software block diagram.

of these libraries a set of tree data interface (TDI) functions [6] were developed to interface the TCV central control software.

3. DSP programming interface

The DSP programming interface appears as a need to develop a simple way of programming the real-time parallel applications with an abstraction layer from the hardware. Without this tool algorithm developers and physicists would have to become experts on the hardware platform before they could take advantage of the system features. Moreover, when applications require timing constraints and real-time interaction with the environment the difficulties are even greater.

The usual approach has been that the system designers provide the user with an API and it is up to the application developers to satisfy the final requirements, checking the time constraints and the quality of service. In this case a different approach is being tried. The user provides the mathematical algorithm while the timing and data transferring requirements are handled by the DSP operating system. With this approach, application programmers can concentrate on improving the algorithms and mathematical specific code, letting the OS guarantee the best possible performance for the hardware.

One of the common difficulties in real-time systems is the scheduling of the several components. When it comes to a synchronous real-time parallel processing system with several CPUs (such as the APCS system) this problem becomes even bigger than in one CPU systems. It is very hard (impossible, most of the times, etc.) for the application programmers to coordinate the use of all hardware resources, providing the desired system performance.

The deeper system designers knowledge when building the low-level software (the OS) is an obvious advantage. Moreover, algorithm developers may be provided with a layer of abstraction from the hardware making it easier to develop the algorithms. In the future, new systems may be designed using the same philosophy for different platforms, providing code portability without spending time learning the new hardware architectures.

3.1. The DATAMOVER

The DSP OS has two main algorithms when in the control cycle:

- *DATAMOVER algorithm*: Responsible for read, format and broadcast the data to all DSPs.
- *TCV control algorithm* (*TCV-ALGO*): Responsible for data processing.

It is the independence between the data transfer (using the DATAMOVER algorithm) and the TCV control algorithm that permits the desired hardware abstraction. The TCV-ALGO developer knows that the data is available for use without worrying with how it was transferred between DSPs. Therefore all his attention may be put into the control algorithm implementation.

3.2. Integrating the TCV-ALGO in the DSP code

To easily interface the DSP OS structure a simple method was found to include the algorithm developed by the physicists in the DSP. The tools available from Texas Instruments are the C and Assembler compiler for the TMS320C3X/4X [7]. Therefore, the solution is to include predefined C and H files into the project where the algorithm can be coded.

This approach permits a higher level programming environment such as a visual language (similar to MatLab/Simulink) to be used. Using a pre-compiler the real-time programming blocks are converted into the C and H code files that are included in the DSP project [8].

The DSP OS interfaces the TCV algorithm by calling three C functions in defined time slots during the control process: (i) `void RT_init(void)`—called when the system is armed, before the first trigger; (ii) `void RT_loop(void)` —called at each real-time control cycle, triggered by the real-time loop clock; (iii) `void RT_post(void)` —called at the end of the shot, after last trigger, when the gate closes or on error as a safety procedure.

3.3. Compiling and loading the DSP software

The TCV-ALGO functions of the previous section are coded in tcvalgo.c and tcvalgo.h in the DSP project. To compile and load the DSPs some problems had to be solved:

(i) Texas Instruments uses the Code Composer Studio as the standard IDE for software development, without Linux version, and with limited scripting capabilities.