

COTS-Based High-data-throughput Acquisition System for a Real-Time Reflectometry Diagnostic

J. Santos, M. Zilker, L. Guimarães, W. Treutterer, C. Amador, M. Manso, and the ASDEX Upgrade Team

Abstract—Achieving higher levels of plasma performance control in present fusion experiments requires that diagnostics be upgraded to deliver processed physical parameters in real time (RT). A key element in a diagnostic RT upgrade is the data-acquisition system (DAS) that should be capable of delivering the acquired data to the data-processing resources with very low latencies and in the shortest possible time. Adequate standard commercial solutions with these characteristics are not easily found in the market, which leads most of the time to the development of complex custom high-performance designs from ground-up. A mixed solution, partially based on commercial off-the-shelf (COTS) components has been developed to upgrade the existing ASDEX upgrade broadband reflectometry diagnostic so that a full demonstration of plasma position control using RT reflectometry density profile measurements can be performed. The designed 8-channel (12-bit/105 MSPS) DAS features a PCI Express 1.1 (PCIe) x8 interface to enable direct memory access (DMA) data transfers with an effective throughput in excess of 1 GB/s. The use of COTS components resulted in a faster hardware design cycle without compromising system performance and exibility. The architecture of the system and its main design constraints are herein discussed. Benchmark results for data throughput and overall latency measurements are also presented

Index Terms—Commercial off the shelf (COTS), data acquisition (DAQ), field-programmable gate array (FPGA), high data throughput, real-time diagnostics.

I. INTRODUCTION

IN controlled fusion experiments, the access to processed physical parameters in real-time (RT) opens the way to higher levels of plasma performance control. Consequently, upgrading diagnostics for RT operation is a growing trend in experimental devices like the ASDEX Upgrade (AUG). A key element in such diagnostic upgrades is the data acquisition system (DAS), which must be capable of delivering the acquired data to the data processing resources with very low latencies and in the shortest possible time. When large volumes of acquired data or a high number of acquisition channels are involved, and/or the processed measurement cycle is very demanding, high-performance custom built designs are in general required. However, many diagnostics do not have such demanding requirements and therefore can live with solutions partially based on commercial off-the-shelf (COTS) components.

Manuscript received June 15, 2010; revised March 30, 2011.

J. Santos, L. Guimarães, C. Amador and M. Manso are with the Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear - Laboratório Associado, Lisboa, Portugal (telephone: +351.218419080, e-mail: jsantos@ipfn.ist.utl.pt).

M. Zilker, W. Treutterer and the ASDEX Upgrade Team are with Max-Planck-Institut für Plasmaphysik, EURATOM Association, Garching, Germany.

In this paper we present the DAS developed for the demonstration of the reflectometry based plasma position control technique, an ITER¹ relevant demonstration presently underway in AUG [1]. In this context, the goal of the RT reflectometry diagnostic is producing density profile measurements and separatrix position estimates for plasma position feedback control in the AUG fastest position control cycle, i.e. ≈ 1 ms [2]. For this purpose, a DMA capable PCIe 1.1 (x8) DAS was designed to remove all possible hardware related latencies from the complete RT measurement cycle, using mainly COTS components. The 8-channel (12-bit/105 MSPS) DAS uses a Xilinx Virtex-5 SX series FPGA to guarantee data burst transfers, between the ADC's local buffer memory and the data processing server's RAM, with effective bandwidths higher than 1 GB/s. The targeted control cycle duration and the very high data-throughput of the DAS relieved the requirement of having a commercial hard-RT operating system (OS) to manage both the acquisition hardware and the RT data processing tasks. In fact, the continuous integration of hard RT capabilities into standard open-source OS like Linux makes them ideal candidates for these applications. The gained system development flexibility also resulted in an easier integration of the designed system in the AUG RT software framework [3].

In the following sections we will briefly describe the reflectometry RT measurements and resulting system requirements/constraints, the proposed DAS architecture and system benchmark results.

II. DIAGNOSTIC CHARACTERISTICS AND REAL-TIME OPERATIONAL REQUIREMENTS

The AUG RT reflectometry diagnostic will produce two types of on-line results: high-field side (HFS) and low-field side (LFS) density profiles, and estimates for the inner and outer separatrix position. These measurements are obtained using two broadband O-mode reflectometers [4] probing the plasma HFS and LFS at the equatorial plane. Interference signals, resulting from the swept operation of the K, Ka, Q and V band microwave sources of both reflectometers, are sampled to produce density profiles covering the $0.3 - 6.0 \times 10^{19} \text{ m}^{-3}$ density range. In total, eight signals are synchronously acquired in frames of N samples, as sketched in the Fig. 1 (a). As the microwave sources used to probe the plasma are swept in $25 \mu\text{s}$ (requiring a settling time of $10 \mu\text{s}$), in this application the DAS is to be operated at one of the two following sampling frequencies: 40 MSPS to acquire $N=1K$

¹The future International Thermonuclear Experimental Reactor.

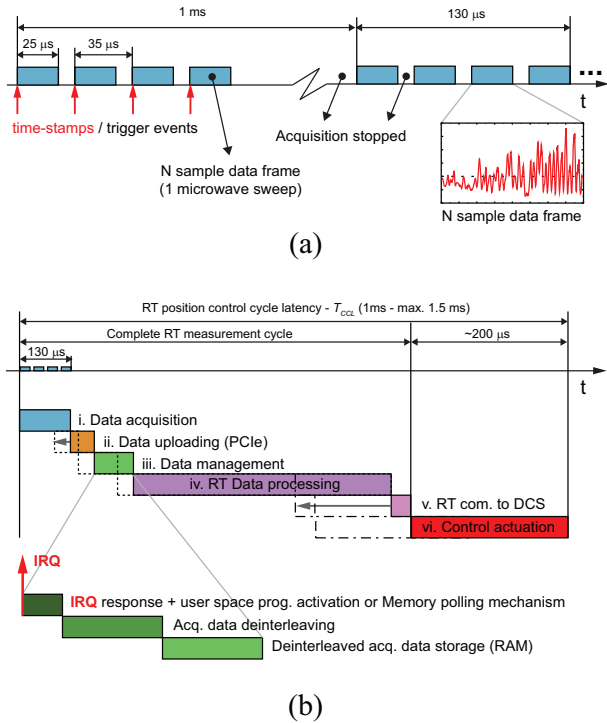


Fig. 1. (a) Burst acquisition and (b) complete RT measurement and position control cycle timing diagrams.

samples per sweep/frame, and 80 MSPS for doubling the amount of data acquired per sweep.

A burst of four consecutive broadband sweeps is required for the RT calculation of single HFS and LFS density profiles and their corresponding separatrix position estimations. To complete the position control cycle these results need to be sent to the AUG discharge control system (DCS) where the control actuation is calculated and generated. Although the targeted cycle, the fastest AUG control cycle, is now 1 ms its length, if needed, can be increased up to 1.5 ms to safely accommodate fluctuations in the calculation/data delivery times.

Fig. 1 (b) shows a simplified schematic diagram of the complete position control cycle. This is a two-phase process that starts with the acquisition and RT data processing of the physical relevant parameters to be delivered to the DCS. These tasks, running locally in the RT diagnostic, are followed by the position control actuation processing, running in the DCS.

The control cycle can be broken down in the following sub-tasks: (i) data acquisition (temporary local storage of data in the acquisition system), (ii) data uploading (to a linear buffer allocated in the host RAM via a DMA transfer), (iii) data management (including IRQ or data polling based task activation and buffered data adaptation and replication), (iv) RT data processing (to calculate density profiles and separatrix position estimates), (v) communication of processed data to the DCS, and (vi) position control actuation processing. The duration of these tasks is either fixed, e.g. (i), bonded and experiment imposed (v) and (vi), dependent of the host system performance, e.g. (iv) or on the choices made for the OS, (iii)/IRQ response, or of the acquisition hardware design,

e.g. (ii). This cycle duration is simultaneously the maximum acceptable latency between the actual measurement (status of the plasma at probing/acquisition time) and control actuation. Thus, any form of sub-task overlap or pipelining must result in a deterministic ≤ 1 -1.5 ms overall latency, irrespective of the obtainable measurement rate.

As the degree of sophistication of the implementable RT data processing algorithms depends directly on the available computation time, all the performed system optimizations were geared towards maximizing the time available to spend in this step of the control cycle. To improve the performance and deterministic behavior of the RT data processing task it is important to eliminate all possible sources of memory bus contention during the calculations phase. One possibility is simply to prevent acquired data upload and data processing phases from overlapping in time. At the hardware level, this strategy involves maximizing the DAS data uploading bandwidth in order to shorten the duration of step (ii). As mentioned, in the reflectometry application, a maximum of 128 KiB needs to be transferred from the DAS every 1 ms. At the maximum theoretical PCIe bandwidth, a 1x PCIe bus (250 MB/s) takes $\approx 524 \mu\text{s}$ to transfer the acquired block of data from the DAS to the computation host, consuming in the transfer $> 50\%$ of the complete targeted control cycle latency (T_{CCL}). Increasing the used number of lanes in the PCIe bus up to eight reduces the memory bus allocation to a mere $\approx 65 \mu\text{s}$ ($\approx 6.5\%$ of T_{CCL}), i.e. below the time taken to actually acquire the 128 KiB data block (130 μs). However, as we will see in section V, in host systems similar to the used one, the attainable effective bandwidth can drop to less than 70% of the referred values (due to protocol header overheads, implemented transaction layer protocol data payload sizes, and various other cumulative system latencies), justifying the choice of the wider x8 PCIe bus for the developed DAS.

On the software side, a multithreaded neural network based algorithm, developed [5] to calculate reflectometry density profiles and produce separatrix position estimates in RT, was implemented using OpenMP. Early benchmarking of this code shown that at least half of the targeted 1 ms measurement period should be available for all the non-data processing activities involved in the complete control measurement cycle. Considering that the data acquisition accounts for just $\approx 130 \mu\text{s}$ of the time ($\geq 500 \mu\text{s}$) available for OS IRQ response and other data management and communication tasks, great flexibility was gained in what respects the choice of RT OS and system hardware architecture. This was the main reason to choose a RT enabled Linux kernel² (standard kernel with RT_PREEMPT patches applied) [6] over more latency optimized RT OS implementations like RTAI (Real-Time Application Interface) [7], Xenomai [8] or even commercial solutions like VxWorks [9]. As writing software for a RT Linux kernel based system requires no special API, no or only minor adaptations of the RT processing codes or hardware device drivers are required. Since, in this application, an IRQ response latency of a few tens of micro-seconds could be

²As the supported/favored Linux flavor on AUG is OpenSuse, an OpenSuse 11.2 (x86_64) distribution and a 2.6.31.12-rt21 (RT enabled) kernel were installed in the host.

afforded, a trade-off between code simplicity and optimal IRQ latency was made, resulting in large benefits in terms of software development and debugging times. In what concerns the complete RT position control cycle, achieving a $\approx 1 - 1.5$ ms mark will depend essentially on the final implementation of the RT data processing code (step iv). However, if required, further optimizations such as converting some calculations to fixed point or increasing the number of CPUs (the original OpenMP code is scalable) are still possible. Additionally, some parts of the data processing can always be implemented in the FPGA in order to further improve the cycle time.

III. COTS BASED SYSTEM ARCHITECTURE

To satisfy the operational requirements mentioned in the previous section, the acquisition system was design to have the following characteristics: 8 channels, 12-bit resolution, switchable 40-80-100 MSPS operation, 128 KiB burst FIFO memory, x8 PCIe bus interface. Using the standard PCIe interface format, rather than the more industrially adopted compact PCIe format (cPCIe) minimizes the number of data bus switches between the acquisition board PCIe endpoint and the microprocessor and RAM buses. By using the larger PCIe slots, directly connected to the used motherboard memory controller hub (Intel 5100 MCH), DMA data transfer latencies are minimized and data throughput maximized. Besides optimizing system performance, bringing the complete acquisition system into the server's rack mount case allows for an "all-in-a-box", compact and self contained RT diagnostic data acquisition and processing system.

The two main hardware building blocks were easily found on the COTS market: an ADC evaluation board featuring a serial LVDS interface and a quad-channel 12-bit, 105 MSPS ADC and a PCIe FPGA development board with an 8-lane 1.1 PCIe bus. Among the many available options, the later was chosen to feature a Xilinx Virtex-5 SX FPGA (XC5VSX95T) since these devices integrate a built-in hardware x8 PCIe endpoint. This FPGA family also has enough internal memory resources to implement the large FIFO required to temporally store the bursts of data, and DSP specialized units to allow for future in-FPGA data processing. Such development boards are also filled with extra functionality such has SFP connectors, Gigabit Ethernet PHYs, USB and RS232 ports, on-board and slotted DDR2 SDRAM memory and multiple programmable clock sources. Above all, they feature special expansion connectors, for customized user application daughter cards, whose pins are directly routed to the FPGA single ended or differential IO pins and IO clock resources.

In practice, the only hardware that had to be developed to build the described acquisition system was one such piggy back daughter card, used essentially to interface the FPGA development board to the ADC evaluation boards. The remaining components, i.e. the centrally synchronized timing device and the low latency RT network interface board, are items required to integrate the diagnostic in the AUG RT diagnostic network. Fig. 2 shows the referred components and how the system interfaces with the reflectometry microwave circuitry and with the DCS via either the low or higher latency RT networks.

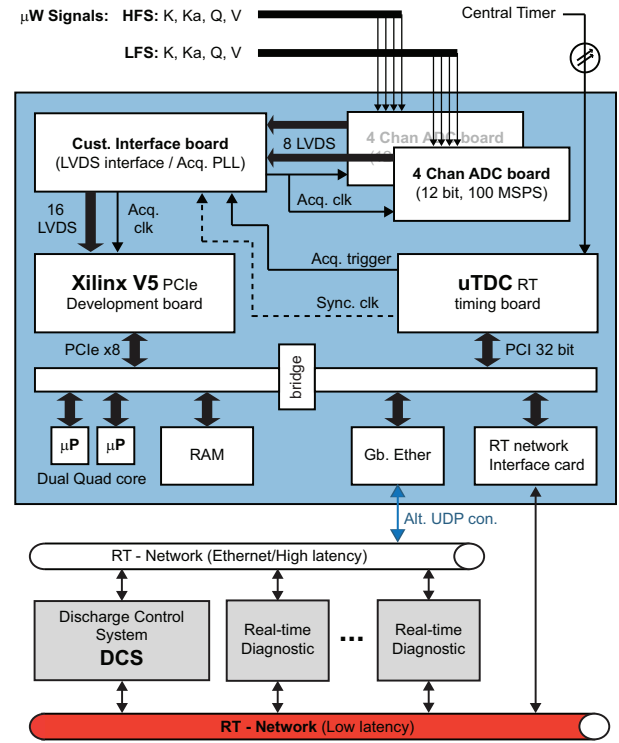


Fig. 2. Block diagram of the main components of the Acquisition system.

All data acquisition systems connected to the AUG RT diagnostic network need to be tightly synchronized with the DCS. The timing synchronization of all participants is achieved using an uTDC board [10], an IPP (Max-Planck-Institut für Plasmaphysik) in-house developed timing device. To guarantee that all uTDC timing nodes, in this distributed system, always share the same (64-bit) time count, all are connected to one central timer via an unidirectional optical fiber network in a star topology. On every millisecond the central timer distributes the actual system time and synchronization information to which each uTDC phase-locked loop (PLL) circuit can lock. The present accuracy of this timing device, available as standard PCI or compact PCI board, is 20 ns. These devices can produce complex timing signal patterns to control ADC boards, using two onboard independent programmable pulse generators (PPGs).

In a first step the RT-Reflectometry diagnostic will be connected to the ASDEX Upgrade control network by a standard Gigabit Ethernet link. This solution provides latencies in the range of a few hundreds of microseconds [11] which will be sufficient for the system integration testing phase. Later, a VMIC reflective memory (GE Fanuc 5565 in a ring topology), guaranteeing true hard real-time operation with bonded latencies of only a few tens of microseconds [11], will be used to connect the reflectometry diagnostic to the low-latency AUG RT network.

IV. HARDWARE AND FIRMWARE DEVELOPMENT

Using mainly COTS components to build the described DAS helped to limit the complexity and the amount of hardware to be developed. As previously mentioned, this system only

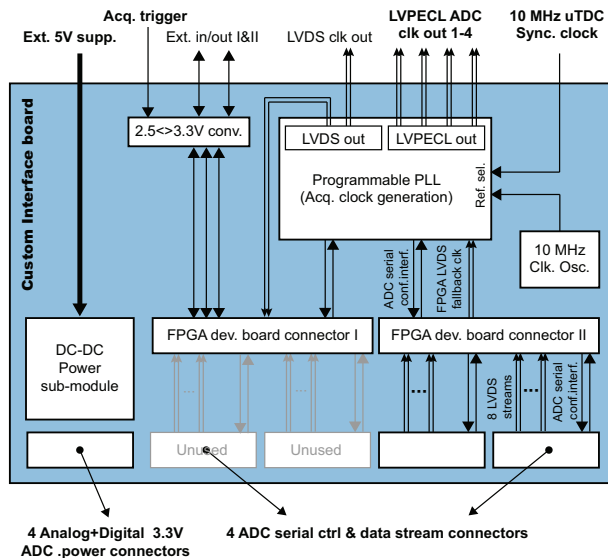


Fig. 3. Custom interface board block diagram.

required a custom-built daughter card to provide an interface between the PCIe FPGA board and the ADC evaluation boards. The main “hardware” development effort, however, was put in programming the FPGA so that the acquired data buffering and very high speed transfer requirements were properly satisfied. A careful planning and an adequate architecture design also helped to limit the complexity of the operations performed inside the FPGA. The sheer performance of recent multi-core processors (the used Tyan-S5375AG2NR motherboard was populated with two quad-core 3.0 GHz Xeon X5450 processors, with 12 MB of L2 cache each, and 8GB of DDR2 RAM) allows the migration of most of the low-level data management functionality, such as sample grouping and reordering or even data filtering, from FPGA to the RT data processing tasks without severe penalties in terms of overall performance. On the other hand, precious development and debugging time is gained since programming applications in C or C++ is a much easier task than programming FPGA in behavioral languages such as VHDL or Verilog. This is particularly true when placing and routing complex designs, with very large data buses working at several hundreds of MHz. In the next subsections a more detailed description of the custom interface board and FPGA firmware functionality will be made.

A. Custom Interface Board Design

As can be seen in the block diagram of Fig. 3, the custom interface board has four main functions. First of all it is used to route the 8 LVDS DDR data streams (two per channel) and frame and data clocks from each ADC evaluation board to the FPGA development board. This board supports the connection of up to 4 ADC boards, i.e. 16 12-bit/105 MSPS acquisition channels, via 4 high speed socket strips. The interconnection between the board and the ADC boards is made through 50Ω high speed cable assemblies that also carry the single ended signals used in the ADC serial programming interface.

A 5V powered DC-DC conversion sub-module generates locally digital and filtered analog 3.3V supply voltages to feed simultaneously all 4 ADC boards and the in-board circuitry. Three single-ended IO connectors are usable for feeding in/out trigger signals that can be locally converted between 2.5-3.3 V voltage levels. Finally, a fully programmable PLL was implemented to produce a high quality low jitter sample clock to drive the ADCs in phase with the 10 MHz synchronization clock generated by the uTDC board. The chosen PLL has LVDS and LVPECL output stages. All four LVPECL outputs are used to drive the ADCs, whilst two of LVDS outputs are connected one to an output plug and the another to one of the FPGA input clock differential buffers. The jitter characteristics of the PLL/internal VCO were evaluated using the PLL supplier own simulator/loop filter calculator. Although the ADC boards could be equipped with the 14-bit versions of the ADC (also natively supported by the interface board), the jitter characteristics of the output LVPECL acquisition clocks only allow a maximum of ≈ 11.7 effective number of bits (ENOB) to be obtained in the application targeted 40-80 MSPS sampling frequency range. The PLL can lock on one of the two available clock reference sources: the external uTDC 10 MHz synchronization clock and an on-board low jitter 10 MHz oscillator. In case of failure of the selected reference, usually the external uTDC sync clock, the PLL automatically commutes to the fall back clock reference.

B. FPGA Embedded Functionality

The choice of the FPGA family was critical to guarantee that the level of firmware development was maintained as low as possible. As stated, the Virtex 5 family integrates an hardware implementation of an x8 PCIe endpoint. By using a third-party DMA IP core, the full functionality of a DMA capable x8 PCIe interface was unlocked. Fig. 4 shows a simplified block diagram of the logic programmed into the FPGA. The complete functionality can be grouped in two main macro blocks charged of the: i) acquisition data flow and ii) generic logic control.

The acquisition data flow macro block contains the blocks required to receive, store and format the data blocks to be uploaded to the host by the DMA management module through the DMA and PCIe EP cores. The LVDS frontend receives the ADC differential DDR data stream pairs, the bit (*dclk*) and frame (*felk*) clocks, and uses the FPGA built-in delay and deserializer resources to reconstruct each 12-bit sample. Since the system synchronously acquires data from 8 microwave channels, the samples are grouped in a 96-bit bus running at a maximum frequency of 100 MHz (when the maximum allowed sampling rate is used). As the ADC clocks run continuously, the frames of samples are formatted and synchronized with the acquisition triggers received from the uTDC in the Acquisition data buffering control logic block. This block can be programmed to accept different frame sizes and number of frames per burst. It also zero pads each channel’s samples from 12 to 16 bits (upgrading the 96-bit sample bus to 128-bit) and provides the required signals to the write interface of the local data buffering block. This memory

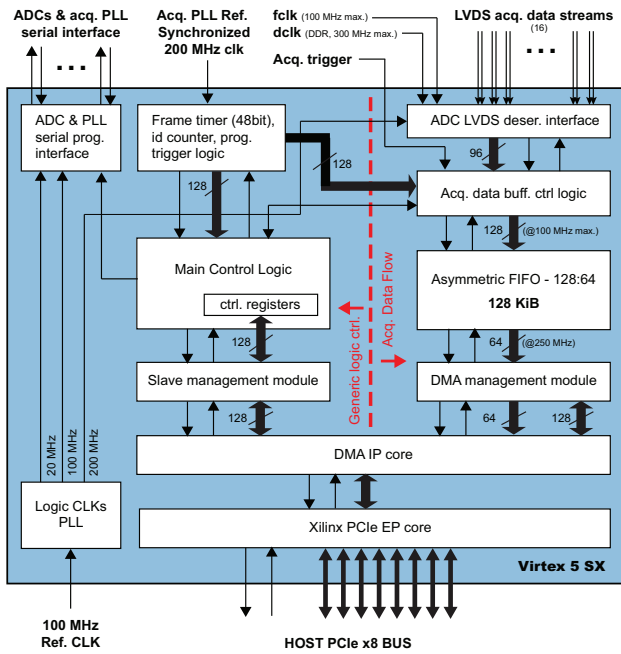


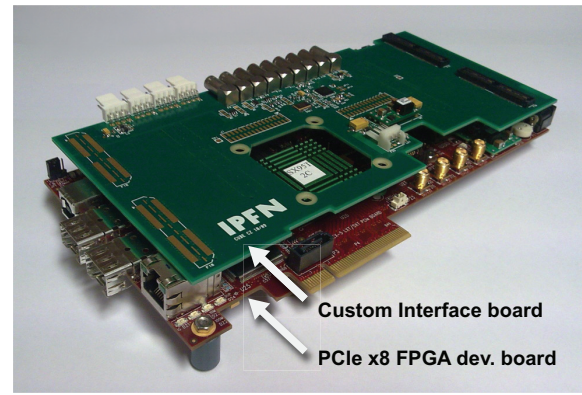
Fig. 4. Simplified FPGA logic block diagram.

block implements a 128 KiB asymmetric FIFO with a 128-bit write interface, operating at a maximum of 100 MHz, and a 64-bit read interface, operating at a fixed 250 MHz rate. The read side is connected to the DMA management module, responsible for the DMA data upload transfer to the host's main memory.

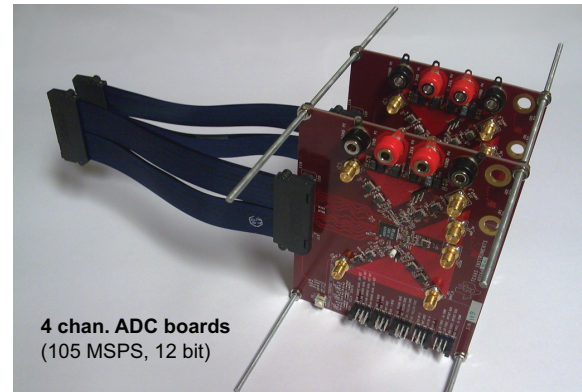
The generic control logic macro block handles all the remaining functions such as synchronizing the frame id and timer registers with the frame acquisition triggers, handling the programmable internal trigger subsystem and delivering PLL and ADC configuration data to the serial program interface block. To perform these functions the Main Control Logic block uses a set of IO registers handled by the DMA core Slave management module. These registers are used to bring configuration data and logic level and triggering signals to the board, and to return the status of the various system block components. One of these registers is used to access the 128-bit contents of the frame timers and id counter registers. The acquisition PLL programmable LVDS output fed to the FPGA is the clock source of the 48 bit time counter used to internally timestamp each acquired frame of samples. As such, this clock, programmed in run-time to have 200 MHz, is always in phase with the uTDC board timer's own 10 MHz clock reference and hence in perfect sync with the received frame trigger signals. Fig. 5 shows pictures of (a) the developed interface board in "piggy-back" with the PCIe FPGA board and (b) of the ADC module sub-assembly and respective high speed cabling.

V. SYSTEM BENCHMARKING

At the time of writing, the reflectometry diagnostic was not yet fully integrated in the AUG RT diagnostic framework [12], and therefore, only the functionality of blocks (i) to (iii) of the control cycle task diagram, depicted in Fig. 1-b),



(a)



(b)

Fig. 5. (a) Custom built interface card piggy backed to the PCIe x8 FPGA development board, (b) 4 channel ADC boards and CTRL & LVDS high speed cable assemblies.

are operational. The system is presently being operated as a basic acquisition system, gathering experimental data with the timings shown in Fig. 1.a). For this reason, only acquisition hardware related benchmarking and preliminary data processing benchmarking are possible at this moment. An extrapolation for the achievable complete control cycle duration will be made at the end of this section based on benchmarks and published measurements of the remaining cycle steps, which are common to other AUG RT diagnostics but not tested here.

The following histograms were obtained in one hour³ runs of continuous 1 ms acquisition cycles (3.600.000 128 KiB data blocks are acquired in each run) in four different setups. The used data block size (common to the remaining benchmarks) corresponds to the acquisition of bursts of 4 microwave sweeps at a 80 MSPS sampling frequency, i.e. to the acquisition of $8 \text{ channels} \times 4 \text{ sweeps} \times 2048 \text{ samples} = 128 \text{ KiB}$ per burst, the maximum allowable size. The four setups consist in running the system using an IRQ or data polling mechanism to start/activate the data management user domain task, running in a high priority level, in either an idle or loaded system. The idle system setup corresponds to the optimal situation where there is no overlap in time between the DMA transfer

³In comparison a typical AUG discharge lasts for at most 10 s.

and task activation process and the data management and RT data processing (RTDP) tasks execution. A loaded system corresponds to the situation where this overlap occurs and the data processing task consumes all its allocatable resources (1 CPU / 4 cores are the computing resources that will be exclusively allocated for this purpose in the host’s final operational configuration).

The test load was simulated using the “Calibrator v0.9e” [13], a cache-memory and translation lookaside buffer (TLB) calibration tool [14], commonly used by the RT community to perform system stress benchmarks. This tool, basically a loop executing a million memory reads of different sizes and changing offsets from a large memory pool, is used to force varying cache miss rates for CPU benchmark purposes. By saturating the access to the memory bus it is ideal to test the RT system handling of hardware generated DMA transfers, task prioritization, IRQ responses, etc. As an extreme load condition, four endless loops continuously running the tool (set for a CPU clock frequency of 3.0 GHz and a memory pool size of 512 MiB) were started in the 4 CPU cores reserved for running the RT data processing codes. Additionally, a continuous “ping” to an neighboring machine in the AUG intranet was started to generate Ethernet related interrupts on a 1 ms time frame (similar to the acquisition repetition period).

A. Effective DMA bandwidth

An eight lane (x8) PCIe 1.1 bus is theoretically capable of transferring data at a 2 GB/S rate. This value, however does not account for transaction overhead, such as packet headers, sequence numbers, CRCs, ERCs and other protocol packets involved in the transfer of large chunks of data. The maximum effective bandwidth (MEB) expectable from our x8 bus, when transferring 128 KiB data chunks, drops to $\approx 85\%$ of this value, i.e. ≈ 1.7 GB/s, due just to the extra protocol header overhead (24-28 B) [15], added to the 128 B transaction layer protocol (TLP) data payload implemented by the host motherboard chipset. In practice, the verified effective bandwidth is much lower due to the contribution of PCIe endpoint and DMA core latencies, switch latencies, data path throttling, availability of buffer credits, size of packet buffers, round-trip system read latency and system memory buffer performance, among others.

To measure the acquisition system DMA transfer rate, we used the 200 MHz 48 bit frame timer implemented in the FPGA firmware. As mentioned before, for every triggered burst of acquired frames, the frame number and the 48 bit timestamp of the beginning of the burst are always automatically registered. A second register can be programmed to store one of the following timestamps: the start time of the DMA transfer, the stop time of the acquisition of a complete burst of data, the stop time of the actual DMA transfer. As the start of the DMA transfer is programmed as a timed offset with respect to the start of the burst acquisition, the actual DMA transfer duration can be easily calculated using this mechanism within a 5 ns precision.

Fig. 6 shows four histograms of the DMA transfer duration. As can be seen, on an unloaded system and in 99.8% of

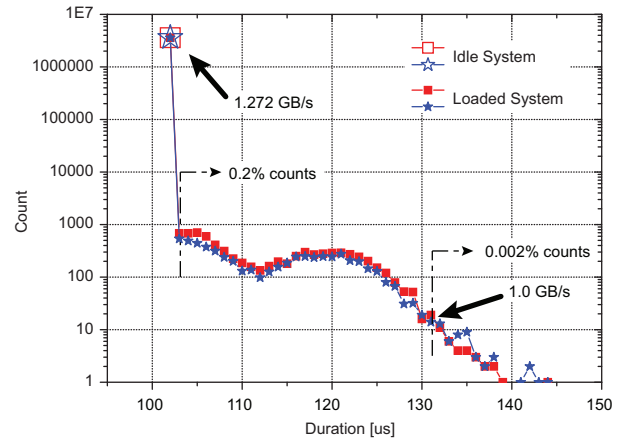


Fig. 6. Histograms (4) of the measured DMA transfer duration (128 KiB block).

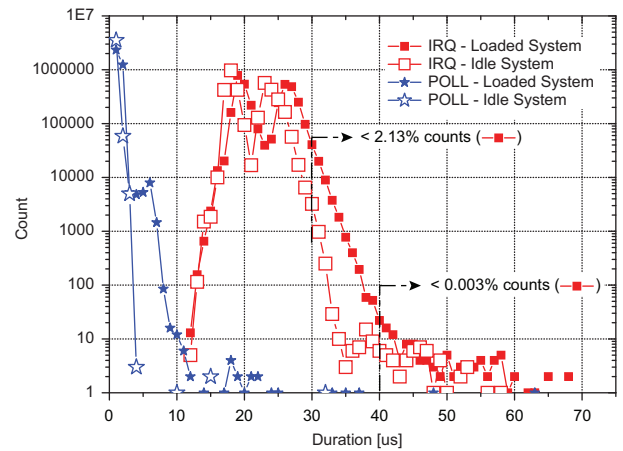


Fig. 7. Histograms (4) of the measured user space task activation delay.

the cases in a loaded system, the measured transfer duration was 103 μ s, what corresponds to an effective bandwidth of 1.272 GB/s, i.e. 75% MEB. The intense use of the memory bus by the four “calibrator” instances affected only 0.2% of the data transfers that, nevertheless, were performed in 99.998% of the cases at a rate ≥ 1 GB/s and never below 908 MB/s.

B. Data management task activation latency

Two methods were tried to activate the data management (DM) task, running in a segregated CPU core in an high priority level: a) IRQ+Unix signal and b) data polling. In the first case, the user task registers itself with the DAS PCIe device driver to receive an Unix signal whenever the later is called to serve an hardware generated IRQ (the FPGA DMA core automatically generates an interrupt after a DMA transfer is completed). In the second case, the DM task continuously polls the shared linear buffer into which the DMA transfer is directly made. Fig. 7 shows four histograms obtained with the two methods in a loaded and an idle host. The IRQ initiated method naturally resulted in the longer delays but in the worst case scenario, a loaded system, the DM task could start processing the acquired raw data, in $\approx 99.28\%$ of the times, just 30 μ s after the end of the DMA transfer. In

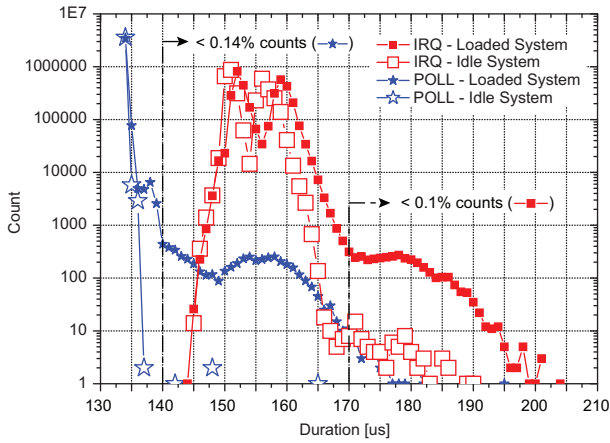


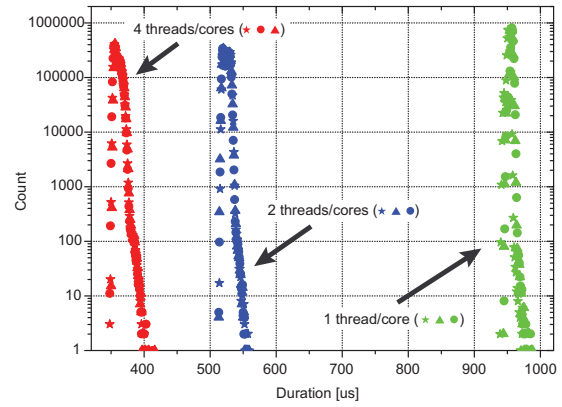
Fig. 8. Histograms (4) of the user space activation delay counted from the trigger of the first frame of each acquired burst.

the interval $[40, 70] \mu\text{s}$ occurred the highest activation times collectively accounting for just $< 0.003\%$ of the measured delays. The polling mechanism, in the unloaded case, reacted in $< 5 \mu\text{s}$ with just 4 measurements being observed in the interval $[5, 35] \mu\text{s}$. In the loaded case this mechanism provides a response in $< 10 \mu\text{s}$. The jitter is higher with a negligible amount of measurements reaching, nevertheless, $\approx 65 \mu\text{s}$.

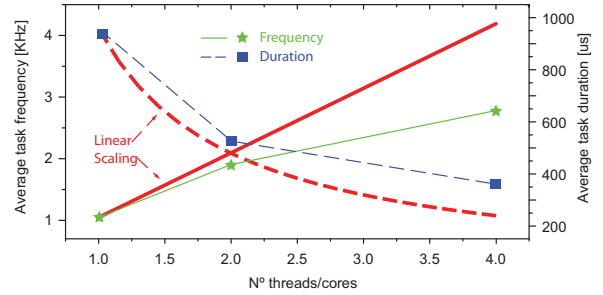
Thanks to the programmable DMA start time mechanism, it was possible to overlap the DMA transfer with the actual data acquisition. Using this feature, implemented in the FPGA firmware, the transfer was adjusted to finish $1 \mu\text{s}$ after the acquisition of the last sample in a burst (what was observed to actually happen in 99.8% of the times - Fig. 6). Fig. 8 shows the overall delay from the beginning of the control cycle (trigger of the first frame of a burst) up to the point where the DM task is started, i.e. corresponding to (i), (ii) and first sub-block of (iii) in the diagram of Fig. 1. Opting for a data polling mechanism seems to be the obvious choice as even in a loaded system data processing can be started $140 \mu\text{s}$, T_{TA_min} , after the burst's initial acquisition trigger in 99.86% of the times. Using IRQs increases the response time up to $170 \mu\text{s}$ in 99.9% of the times. In any case no overall DM task activation latencies were ever registered above $T_{TA_max} = 210 \mu\text{s}$.

C. Real-time data processing task duration

At this time, the density profile and separatrix estimation RTDP multithreaded code is still being optimized as part of the on-going process of system integration in the AUG RT diagnostic framework. The code has been written in OpenMP and takes advantage of the large 12 MB L2 cache of the chosen 4 core Xeon CPU to exploit the spacial and temporal localities allowed by the self-contained small sized input data block (max. 128 KiB). As during each processing cycle no access to data other than the one just acquired is required, both data and processing code easily fit the CPU cache, avoiding penalizing cached page misses. Additionally, no IO system calls are made inside the RTDP calculation loop, consequently a bonded deterministic behavior was observed when running the existing code in a high priority level on a dedicated CPU.



(a)



(b)

Fig. 9. (a) Histograms (9) of the RT data processing duration using 1, 2, and 4 threads/cores; (b) scaling of the average duration and frequency of the RT data processing iterations using 1, 2, and 4 threads/cores (thick lines represent a linear scaling extrapolated from the duration/frequency of the mono-threaded case).

Fig. 9 shows the obtained performance scaling when enabling up to 4 threads (as many as available cores in the same CPU). The performance scaled well from one to two threads. However, the Xeon 5450 large L2 cache architecture, in fact two 6 MB caches serving a group of 2 core each, might have been responsible for the weaker scaling observed when moving from 2 to 4 threads. Anyhow, computation times higher than $T_{RTDP_max} = 420 \mu\text{s}$ were never observed when running the present version of the code using 4 threads, as is exemplified in the histogram of Fig. 10. In 99.966% of the calculation cycles the time required to produce the results to be sent to the DCS were obtained in $\leq 380 \mu\text{s} = T_{RTDP_min}$.

D. Complete control cycle latency extrapolation

Based on the results obtained so far, RT reflectometry measurements can be produced with a maximum latency of $630 \mu\text{s} = T_{TA_max} + T_{RTDP_max}$. To calculate the complete control cycle latency one must add to this value the delays involved in the communication of the RT results to the DCS and the actual control actuation processing ($T_{CTRL} \approx 200 \mu\text{s}$). If the higher latency UDP connection is used, a measured maximum latency of $T_{COM} \approx 200 \mu\text{s}$ [11] must be added to this value. The use of VMIC reflective memory cards guarantees communication latencies of the order of a few tens of μs for the amount of processed data to transfer (a maximum of ≈ 50

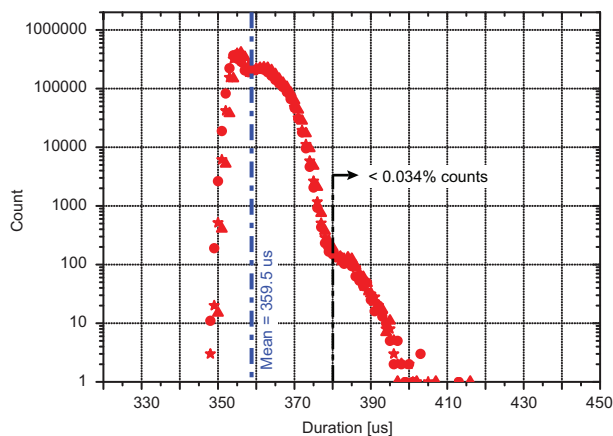


Fig. 10. Histograms (3) of the RT data processing duration using 4 threads/cores.

DW)⁴. It is then safe to assume that the complete control cycle latency, T_{CCL} , will rest somewhere inside the $900 \mu s < T_{TA_max} + T_{RTDP_max} + T_{COM} + T_{CTRL} < 1030 \mu s$, well below the maximum allowed control latency (1.5 ms) and very close to the targeted fastest AUG control cycle (1 ms). Even better timings can be obtained if fallback solutions are found to compensate, at the control stage, the $< 0.2\%$ measurements that correspond to the tails of the shown histograms. If a polling solution is applied, the same extrapolation interval would fall down to $790 \mu s < T_{CCL} < 920 \mu s$. Because the last stage of the control cycle is taking place on the DCS itself, and hence can be overlapped in a pipeline fashion with the remaining tasks running in the diagnostic host, maintaining the same control cycle latency a maximum theoretical control cycle rate of $1/(T_{TA_min} + T_{RTDP_min}) = 1/(520 \mu s) = 1.923 \text{ kHz}$ could be reached in the later proposed scenario.

VI. CONCLUSION

It was shown that, by using available COTS components, compact low latency and high throughput data acquisition systems can be built with limited hardware development. This option implies that the main development effort is displaced to the programming of the embedded FPGA devices. However, the high performance of cheap and widely available multi-core multiprocessor host servers directly contributes to limit the complexity programmed into these FPGAs. In fact, if this complexity is moved to RT user-space tasks running on the acquisition system host, the use of parallel programming paradigms such as OpenMP and optimized parallel digital signal processing function libraries allows for a quicker development and prototyping cycle of the required data processing and management algorithms. In the end, a solution capable of satisfying demanding RT measurement cycle rates, without the need for complex and time consuming development of FPGA based data processing codes, is achievable.

For RT diagnostics compatible with overall system response times greater than $\approx 50\text{-}100 \mu s$, the standard RT Linux kernel

⁴For comparison purposes, $70 \mu s$ is the observed latency for the transfer of a 39×69 element matrix of magnetic flux [11].

(mainstream kernel with RT_PREEMPT patches) implementation has the required characteristics to guarantee an adequate deterministic behavior. By using it, overall system implementation simplicity and access to a much broader hardware and software support is gained. These advantages alone justify, in these cases, its choice over a standard hard-real time OS.

The hardware prototype of this system was commissioned at the experimental site. After complete integration in the ASDEX Upgrade RT network, during the 2011 experimental campaign, a demonstration of plasma position feedback control using reflectometry measurements will be made in both H-mode and ELM free regimes.

ACKNOWLEDGMENT

This work, supported by the European Communities and the Instituto Superior Técnico, has been carried out within the Contract of Association between EURATOM and IST. Financial support was also received from the Fundação para a Ciência e Tecnologia in the frame of the Contract of Associated Laboratory. The views and opinions expressed herein do not necessarily reflect those of the European Commission, IPP, IST and FCT.

REFERENCES

- [1] J. Santos, et al., "Status of the demonstration of reflectometry based plasma position control on ASDEX Upgrade", in *9th International Reflectometry Workshop*, Lisbon, 2009.
- [2] W. Treutterer, et al., "Real-time diagnostic at ASDEX Upgrade - integration with MHD feedback control system", in *Proc. 6th IAEA TCM on Control Data Acquisition and Remote Participation on Fusion Research*, Japan, 2007.
- [3] K. Behler et al., "Real-time standard diagnostic for ASDEX Upgrade", *Fusion Engineering and Design*, vol. 85, no. 3, pp. 313-320, Jul. 2010.
- [4] A. Silva, et al., "Microwave reflectometry diagnostic for density profile and fluctuation measurements on ASDEX Upgrade", *Rev. Sci. Instrum.*, vol. 70, no. 1, pp 1072-1075, Jan. 1999.
- [5] J. Santos, "Reflectometry measurements for plasma position control purposes", Ph.D. dissertation, Dept. Elect. Eng., Instituto Superior Técnico, Lisboa, Portugal, 2008.
- [6] Open Source Automation Development Lab. (2011, March). Real-time Linux project [Online]. Available: <https://www.osadl.org/Realtime-Linux.projects-realtime-linux.0.html>
- [7] RTAI Team. (2010, February 4). RTAI - the real-time application interface for Linux [Online]. Available: <https://www.rtai.org/>
- [8] Xenomai Team. (2010, July 3). Xenomai real-time framework for Linux [Online]. Available: <http://www.xenomai.org>
- [9] Wind River Systems Inc. (2011, March). VxWorks real-time operating system [Online]. Available: <http://www.windriver.com/products/vxworks/>
- [10] A. Lohs, K. Behler, K. Lddecke, G. Raupp and ASDEX Upgrade Team, "The ASDEX Upgrade UTDC and DIO cards - a family of PCI/cPCI devices for real-time DAQ under Solaris", *Fusion Engineering and Design*, vol. 81, pp 1859-1862, Jul. 2006.
- [11] W. Treutterer, et al., "Real-time signal communication between diagnostic and control in ASDEX Upgrade", *Fusion Engineering and Design*, vol. 85, pp 466-469, Jul. 2010.
- [12] M. Reich, et al., "Real-time diagnostics and their applications at ASDEX Upgrade", *Fusion Science and Technology*, vol. 58, no. 3, pp. 727-732, Nov. 2010.
- [13] S. Manegold. (2004, June 24). The Calibrator (v0.9e), a cache-memory and TLB calibration tool [Online]. Available: <http://www.cwi.nl/~manegold/Calibrator/>
- [14] S. Manegold and P. Boncz, "Cache-memory and TLB calibration tool", Available: <http://homepages.cwi.nl/~manegold/Calibrator/doc/calibrator.pdf>
- [15] K. Lund, D. Naylor, M. DiPaolo, and S. Trynosky, "Virtex-5 FPGA integrated endpoint block for PCI Express designs: DDR2 SDRAM DMA initiator demonstration platform", XAPP859 - v1.1 Application Note, Xilinx Inc., 2008.