computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

FROM CAMAC TO WIRELESS SENSOR NETWORKS AND TIME-TRIGGERED SYSTEMS AND BEYOND: EVOLUTION OF COMPUTER INTERFACES FOR DATA ACQUISITION AND CONTROL. PART I

Janusz Zalewski

Dept. of Software Engineering, Florida Gulf Coast University
Fort Myers, FL 33965, USA
zalewski@fgcu.edu, http://www.fgcu.edu/zalewski/

Abstract: The objective of this paper is to present a historical overview of design choices for data acquisition and control systems, from the first developments in CAMAC, through the evolution of their designs operating in VMEbus, Firewire and USB, to the latest developments concerning distributed systems using, in particular, wireless protocols and time-triggered architecture. First part of the overview is focused on connectivity aspects, including buses and interconnects, as well as their standardization. More sophisticated designs and a number of challenges are addressed in the second part, among them: bus performance, bus safety and security, and others. *Copyright © Research Institute for Intelligent Computer Systems, 2016. All rights reserved.*

Keywords: Data Acquisition, Computer Control, CAMAC, Computer Buses, VMEbus, Firewire, USB.

1. INTRODUCTION

The design and development of data acquisition and control systems has been driven by applications. The earliest and most prominent of those were applications in scientific experimentation, which arose in the early sixties of the previous century, where a number of fast changing signals had to be sampled periodically and stored over a short time interval. This need combined with “the continuing downward trend in the size and price of minicomputers and their flexibility and capacity for data handling” has led to the introduction of digital techniques in laboratory automation [1]. This has become particularly evident in the areas, such as chemistry, biochemistry, medicine, and nucleonics, where data collection and processing, often for spectral analysis, are commonplace. Especially demanding were data acquisition requirements for experiments in physics, mostly in high-energy and nuclear physics [2].

This demand, which came from multiple nuclear research laboratories around the world that worked on joint experiments and required standardized digital equipment, led to the formation of ESONE Committee (European Standards On Nuclear Electronics). The ESONE committee was established in 1961 [3] and its major accomplishment was the development of a number of standard documents for Computer Automated Measurement And Control, known as CAMAC, which later became international standards adopted by IEC and IEEE [4]-[7].

The CAMAC standards played a significant role in developing data acquisition and control instrumentation not only for nuclear research, but also for research in general and for industry as well [8]. They helped realize how important it is to have compatible equipment from different manufacturers to build and configure entire systems, tailored towards requirements specific to given applications. This trend was followed by a proliferation of other standards, which were developed either to keep up with advances in technology, such as VMEbus, or with a demand of specific application areas, such as GPIB, or just with the evolution of new concepts in engineering, such as PCI [9].

The objective of this paper is to overview the development of data acquisition and control systems from a historical perspective, focusing on the evolution of design concepts. The paper is structured as follows. The next two sections describe briefly basic concepts of data acquisition and control, and give an overview of their utilization in designing the CAMAC system. Section 4 discusses bus design principles based on the example of Multibus II design, and Section 5 continues with VMEbus case study. Section 6 covers standards relevant to serial buses: Firewire (IEEE-1394) and USB. The final section presents paper summary and conclusions.

2. PRINCIPLES OF DATA ACQUISITION AND CONTROL

General principles of data acquisition and control are widely known and additionally covered in multiple publications, including books and tutorials, so this section only overviews the basic concepts and outlines the rules of connectivity, which are essential in building systems.

A general view of a data acquisition and control system is shown in Fig. 1 [10]. The essential components include the Plant (or Experiment), with respective sensors (marked with an empty circle) providing values of measured variables (quantities) and actuators (marked with a filled circle) delivering control signals from the Controller to the Plant. Two variations of this system are common: plain data acquisition system, when there are no Command Signals from the Controller to the Plant, and a programmed control system, when there are no measured values delivered from sensors to the Controller (as in the case of numerical control or 3D printing, for example).

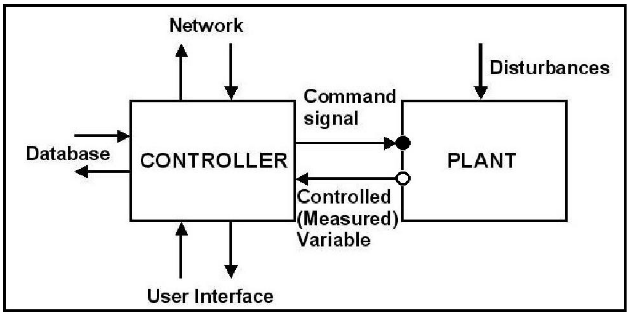


Fig. 1 – Generic Example of a Data Acquisition and Control System

The path of a signal representing the measured variable, from the sensor to the controller is not as trivial as it appears in Fig. 1, and involves normally, either discrete or integrated electronics, in a form of transducers, amplifiers, filters, multiplexers, sample-and-hold circuits and analog-to-digital converters that adjust the raw signal to be represented digitally in a form acceptable for processing by the Controller and for storage. Similarly, command signals from the Controller to the actuators require some electronic circuits to make them acceptable for the plant; this varies depending on the type of actuator and may involve digital-to-analog conversion, pulse signals, etc. Signals in both directions can be also plain digital, as opposed to analog form.

Out of many fields where data acquisition and control systems are applied, the most demanding are physics and nuclear applications, due to their massive amounts of data (for example, in high-energy physics [2],[11]) and specific concerns regarding dependability (in nuclear reactor control). Therefore, it is worthwhile to address this specific area, because challenges the designers and engineers of these systems are facing help in understanding the entire spectrum of applications and facilitate developing systems for all of those.

Historically, the first data acquisition system in physics was conceived in 1919 [12] (Fig. 2). As LeCroy stated it [13]: "even though the technology has changed a lot since 1919, the fundamental pieces of this little circuit are still employed in every high-energy physics experiment". The detector (E) plays the role of a sensor, in modern terms. As another paper explains it [14], it is connected to a programmable trigger (HR, for High Resistance) and analog front-end electronics (G, for Geiger tube), through which the signal is acquired by an amplifier (vacuum tube, which plays a role of an analog processor), and then directed to an actuator, which is a relay (R), and through a local circuit (LC) recorded on analog media (P for Pen).

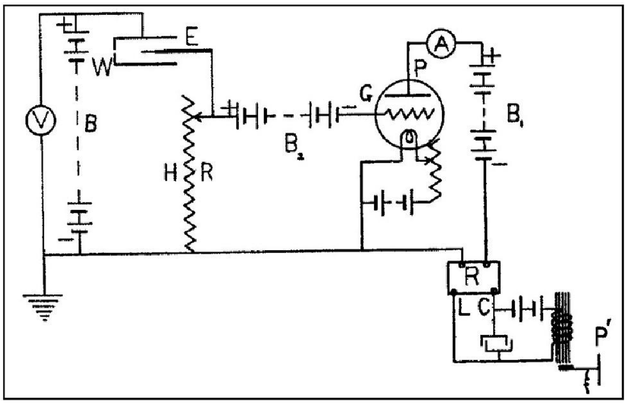


Fig. 2 – First documented example of a data acquisition system for a nuclear experiment [12]

The growing demand for acquiring and handling multiple quantities at the same time for a single experiment, associated with the progress in digital technology, led to the necessity of developing systems, which could handle multiple instruments simultaneously. Since traditional computers were designed for data processing and not for acquiring experimental data online, scientists and engineers in the field of laboratory automation began paving their own ways into designing systems that would interface measuring equipment to computers.

In contemporary experiments, depending on the number of measured variables, which can be extensive, and the speed of their changes, as well as required frequencies of recording, the sophistication of computing systems may become significant. To accommodate this need, special computer architecture was developed based on a concept of a bus, to allow handling multiple data sources at a time.

3. THE DEVELOPMENT OF CAMAC

The initial CAMAC standard [4]-[7] was developed in response to urgent need, in the sixties, to standardize transfer of digital data from measuring instruments to and between computers and data processing devices. The use of interfacing standards of individual computer manufacturers, if such standards existed at all, was no longer an option due to the rapidly increasing amounts of data, growing number of diverse instruments, and diversity of computing devices. As one author states it, the common interface was needed, which would be independent of the type of the computer and the kind of data source, whether it is “a process plant, a section of a transport network, a generator of electric power, a hospital patient, or the behavior of a nuclear particle” [15]. Thus, thanks to the efforts of European laboratories [3], the CAMAC standard was born.

Why was it beneficial to use CAMAC at that time? The ESONE committee suggested the following important reasons:

- A modular system with functional plug-in units that mount in a standard crate.
- Designed to exploit the highest packing density possible with solid state devices.
- Plug-in modules connect to data highway (Dataway) that is part of the crate and carries data, control signals and power.
- Can connect to on-line computer, even though the use of a computer is optional.
- Crates can be connected together by either parallel or serial highways.

As pointed out by Clout [8], there were several other benefits of using CAMAC:

- CAMAC is a high-level hardware, which means that it is computer independent and thus can be considered somewhat like high-level languages such as Fortran, etc.
- CAMAC means that system building and maintenance resources are used effectively. In choosing CAMAC much of the system design already exists, and thus this effort is saved, and engineers and technicians need be trained in only one system.
- CAMAC is modular, so systems can be built up and extended in easily understood units.
- CAMAC is an international standard and results in a stable system – important for people investing in systems for a long life.
- CAMAC reduces overall system costs, assuming that a professional approach is used in building the system, and labor costs are realistic.
- CAMAC facilitates graceful upgrades.

3.1 PRINCIPLE AND OPERATION

CAMAC was specified at three different levels: mechanical (physical dimensions of modules, their, housing and connectors), electrical (voltage levels and signal timing), and logical (protocol for the modules to operate together and communicate). In addition, it was a crucial decision at that time that data acquisition systems would need to be a bus structured rather than individual point-to-point connections, which led to establishing a communication principle via a bus, specified and named a Dataway.

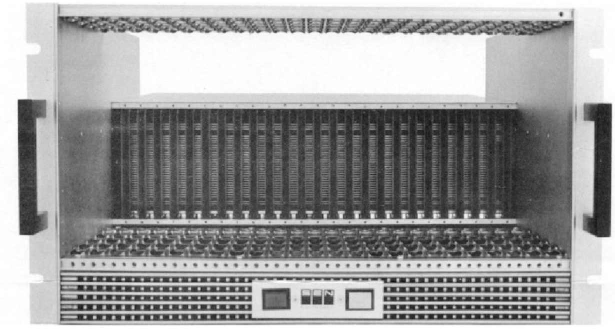


Fig. 3 – CAMAC Crate

Fig. 3 shows an appearance of the empty chassis for housing modules, called a crate, with Dataway connectors visible on the backplane in the back. A central controller placed in the rightmost slot within the crate communicated with all other plug-in modules over the Dataway bus lines, as shown in Fig. 4.

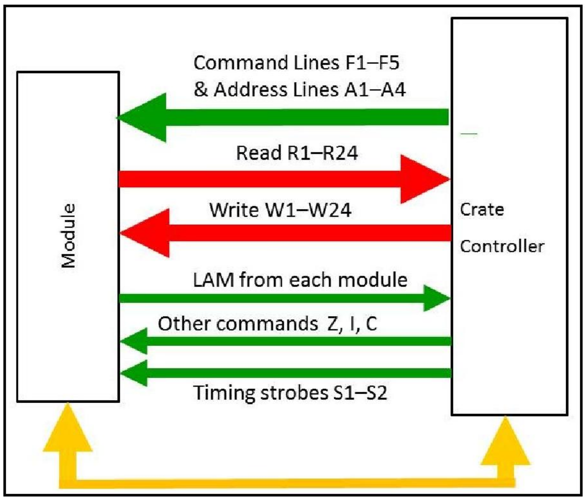


Fig. 4 – Organization of a CAMAC Dataway

Each module, including the controller, has access to bus lines organized in the following groups: addressing and program functions, 24-bit data lines (separate for Read and Write), other commands and strobes. Separate lines from each module to the controller (LAM) played a role of interrupts.

3.2 CAMAC SYSTEMS

CAMAC standard became enormously popular and very quickly complete systems have been built. Fig. 5 shows an example of a medium complexity crate filled with modules.

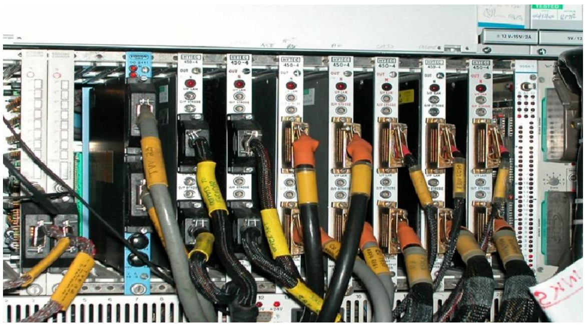


Fig. 5 – CAMAC Crate with Modules

Since the early days of CAMAC, exchange of information on designs and systems has grown almost exponentially. Multiple laboratories in Europe, United States, and around the world (including Japan, South Africa, Australia, etc.) reported on their findings on the suitability of CAMAC for data acquisition and control purposes, as well as on their accomplishments [15]-[21].

Very early in the life of the system, individual modules were designed [22], and interfacing of computers to CAMAC based instrumentation was started [21], including board design for PDP-11 [23] and IBM 360 [24], and many others. Enormous amount of papers have been published, since the birth of the CAMAC standard, with early developments documented in several bibliographies [25]-[29].

What are the lessons from developing and using this standard? First, it provided the physical and logical framework for designing and developing large parts of custom-built data handling systems for both acquisition of the data from experiments, and the parts requiring data processing and data communication. Secondly, it perfectly met the needs of the users, providing a uniform digital “bus which could be used to transfer data, control, and status information between a single host and a set of functional I/O devices” [30]. But it was quickly realized that “every system needs more than one bus”, a statement attributed to Shlomo Pri-Tal of Motorola [31].

During the time CAMAC was successful and “ruled the world” of digital instrumentation, it was also realized that its lifetime is limited by two primary factors (its limitations):

- evolution of applications and advent of new, more demanding requirements, and
- technological progress in electronics.

3.3 NOTE ON FASTBUS

The evolution of application requirements can be viewed from two ends. At a lower end, slower processes and experiments, where the cost of introducing CAMAC was somewhat prohibitive, led to nearly parallel development of the General Purpose Interface Bus (GPIB), soon standardized as IEEE Std 488 [32]. At the high end, speed and volume requirements of high-energy physics experiments soon led to the development of a Fastbus standard [33]-[34]. Since this paper focuses on the progress and evolution in instrumentation interfaces, we only make a brief note on Fastbus.

While CAMAC design was meant to meet all sorts of applications, Fastbus developers realized early on that a general purpose instrumentation bus, although desirable, was technically not necessarily a good idea. Consequently, a new bus was designed to meet high-end applications in high-energy physics only, which are both I/O and computationally intensive. Among the most important and notable features, it included the following [35]:

- no restrictions on the I/O facilities
- parallel processing on separate segments
- multiple processors on a segment
- large address space for easy communication
- physically large module cards
- ease of module configuration and exchange.

With this, additional technical details can be included in the description:

- formation of segments as a basic architectural feature (including crate and cable segments)
- common basic protocol within and between segments
- participating devices consisting of masters, slaves, and segment interconnects
- priority arbitration to support multiple masters
- 32 multiplexed signal lines for address/data
- no distinction between on- and off-segment addressing, with support for broadcast, geographical and logical addressing
- primary (device) and secondary (internal) addressing
- separation of data space from control and status register space
- random, block (async) and pipeline (sync) data transfer modes.

Because of a significant effort devoted to the design of the Fastbus standard and the development of practical systems and applications, given additional experiences and lessons from CAMAC, Fastbus by itself, even though suitable for only one specific market, from today’s perspective may form a good case study for bus design.

4. BUS DESIGN PRINCIPLES

It has been made clear, with early bus designs, that the essential characteristics of conventional bus definition must include three kinds of specs, mechanical, electrical and logical elements, which can be described as follows [36]:

- mechanical properties that concern bus wiring, connectors, their pinout, and module design and dimensions, and often the chassis/housing for module placement, as well
- electrical (or optical) properties that are related to signal levels and their dynamics to carry information, including electromagnetic characteristics concerning transmission line effects, as well as power requirements, and
- logical properties that concern the protocol of exchanging information over a bus.

Specifics of the bus protocol are the most interesting from the perspective of this paper and must include separate descriptions of the three phases of bus operation:

- bus arbitration (competing for bus access)
- data transfer, how devices exchange data once they obtain bus access, and
- fault handling (dealing with bus errors).

Although these aspects relate to all types of buses, one has to add that of interest to this paper are not that much internal (on-board or on-chip) buses, for communication inside the computer, but external buses, essential for instrumentation devices, that is, for communication with peripherals, which includes backplane buses.

4.1 HARDWARE ISSUES

Bus design requires careful consideration of mechanical and electrical properties. While mechanics are important overall, they are out of scope of this paper. It is worth noting that electrical properties of the bus used for instrumentation involve a number of issues deserving careful attention. The primary focus among them has to be given to signal propagation along the bus.

In principle, the bus line can be treated as lumped parameter circuit, but only if the round-trip signal delay is much less than the transition time (rise and fall). This translates into a requirement that the size of the circuit must be much smaller than the wavelength of the signal.

Thus, contemporary buses, due to their high speed, should be treated as transmission lines, with respective theories applied. In particular, one essential parameter has to be considered, which is signal delay. Two primary components of such delay must be taken into account:

- propagation delay along the bus line itself, and
- the response time of the interface circuits connecting the module to the bus.

The latter factor, named board (or chip) related response time, includes two primary issues related to circuit delays:

- the setup and hold time of latches and
- the skew between data and control paths.

Pure bus delay is actually attributed to the settling time (when the signal travels along the bus) and propagation delay. These two factors are interrelated and cannot be treated in isolation. Bus irregularities and discontinuities, such as unmatched loads, connection points, board-layer changes, and drivers and receivers, can disturb the transmission line, thus causing changes in propagation.

On the other hand, settling time is the function of the loaded round-trip propagation delay and the number of reflections it takes to cross the signal threshold. There are several engineering techniques to optimize the bus design regarding these aspects, most of them explained in [36]. So are the other bus design issues, some of them familiar, such as, bus driver circuits, length of traces on boards, crosstalk, signal skew, capacitance of transceivers and connectors, grounding, and others known exclusively to bus designers, examples of which are: wired-OR glitches and metastability.

4.2 BUS PROTOCOL

The essential assumption with all post-CAMAC buses (although it was raised in CAMAC, as well, with multiple controllers in a CAMAC crate [37]) is that a single bus shall handle multiple competing processors. Unlike in the first CAMAC standard [4], where only one central controller was permitted per crate, the need for increased functionality and intelligence in a crate, caused by the growing complexity of applications, made the designers think of more advanced solutions.

The same became true for the industry. Electronic and computer companies began developing, first proprietary, then standard buses, allowing multiple processor boards per chassis. That way, the market was quickly flooded with bus designs originating from the most prominent manufacturers, such as Motorola (Versabus [38] and VMEbus [39]-[40]), Intel (Multibus I and Multibus II [41]), Texas Instruments (NuBus) [36], and even Digital Equipment (BI – Backplane Interconnect). From today’s perspective, almost all these buses are outdated and some of the respective standards have been withdrawn, but for the purpose of illustrating progress in bus design, protocol description of Multibus II is discussed below, based on the author’s participation in its design project.

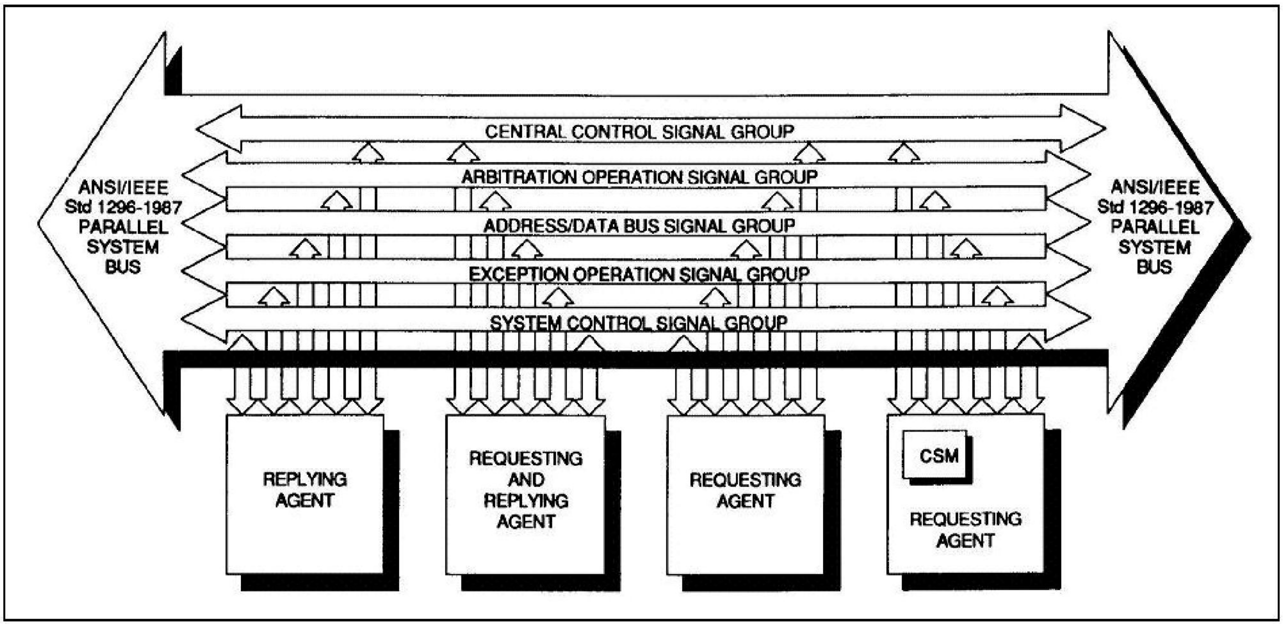

CENTRAL CONTROL SIGNAL GROUP
ARBITRATION OPERATION SIGNAL GROUP
ADDRESS/DATA BUS SIGNAL GROUP
EXCEPTION OPERATION SIGNAL GROUP
SYSTEM CONTROL SIGNAL GROUP
ANSI/IEEE Std 1296-1987 PARALLEL SYSTEM BUS
ANSI/IEEE Std 1296-1987 PARALLEL SYSTEM BUS
REPLYING AGENT
REQUESTING AND REPLYING AGENT
REQUESTING AGENT
CSM
REQUESTING AGENT

Fig. 6 – Block Diagram of the Multibus II Parallel Bus Interface [41]

The organization of Multibus II is shown in Fig. 6. Modules on the bus are called agents, with one of them having central control functions, named Central Services Module (CSM). As one can tell by comparison with the CAMAC Dataway (Fig. 4), there are very few differences in the concept:

- address and data lines are multiplexed
- there is new group of lines for arbitration (CAMAC was a single controller system)
- there is an expanded number of control lines (in CAMAC it was only Z, I, C and strobes)
- there is a group of lines for error handling (exceptions)
- an equivalent of interrupt lines (LAM in CAMAC) are missing, because in multiprocessor systems this is accomplished by message passing.

The principle of bus operation, which is synchronous (Fig. 7), involves arbitration first, then data transfer once this particular bus contention is resolved, and a new arbitration can start before current data transfer is completed. In this sense, the bus operation is parallel, since arbitration is done simultaneously with data transfers.

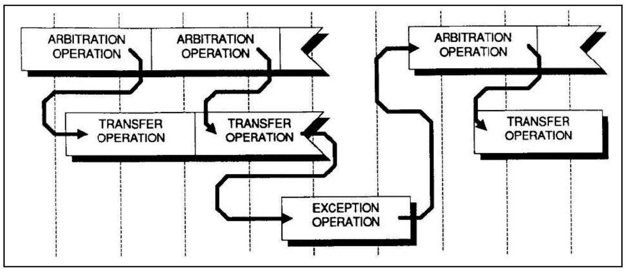


Fig. 7 – Bus Operation Principle [41]

A more detailed picture of bus operation is shown in Fig. 8 (with signals listed in Table 1).

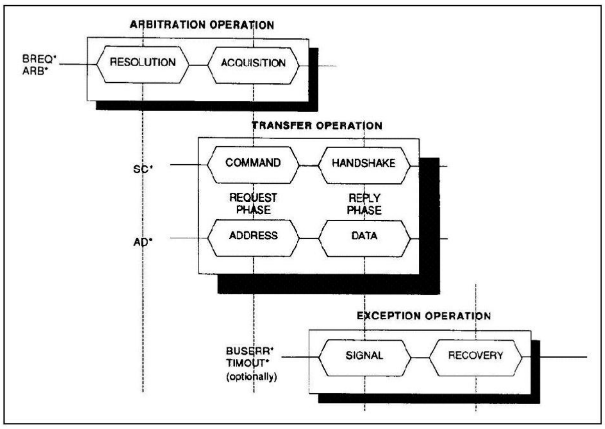


Fig. 8 – Bus Operation Details [41]

Table 1. Signals on the Multibus II Parallel Bus.

Group Name	*Signals*	#	*Description*
Arbitration	BREQ* ARB*	7	Bus request and resolution
Address/Data	AD* PAR*	36	Adrs & data read/write with parity
System Control	SC*	10	Control of adrs & data
Exceptions	TIMOUT* BUSERR*	2	Error indication
Central Control	RST* PROT* + others	7	Provide system wide services

Full description of the bus [41] may be hard to read but popular articles are available too [42]-[45].

5. VMEBUS AND ITS OFFSPRINGS

Multibus II was the product of Intel in the "bus wars" of the 1980s of the previous century, and it met expectations of the time, but actually never took off, in a sense of creating an industry base. Its major competitor at that time was VMEbus, backed by Motorola and associated companies, in particular, by nuclear laboratories. It had a more appealing history and is around in data acquisition and control systems till this day. In fact, it is celebrating its 35th birthday this year.

Fig. 9 – Example of VMEbus Crate

VMEbus evolved from the Motorola Versabus design [38] by changing the board dimensions to the Eurocard and renaming it to Versa Module Eurocard (VME). It was initially announced in 1981, with a full standardization for Revision C.1 achieved in 1987 [39]-[40], and many extensions afterwards. Conceived initially as a computer bus, it was immediately picked up by designers of data acquisition and control systems [46]-[48], and – due to its flexible design and open specification – by a host of tech hungry industries, in particular, by the military. A typical VMEbus crate (chassis) compared to that of CAMAC (Fig. 3), only with different dimensions, is shown in Fig. 9.

5.1 PROTOCOL DESIGN

The original VMEbus is an asynchronous bus with 4-edge handshaking between participating modules and 8-bit, 16-bit or 32-bit data transfers, non-multiplexed with address lines. This is somehow in contrast to Multibus II design, which is synchronous and multiplexed. The bus uses 16-bit, 24-bit or 32-bit addressing, and 6 additional Address Modifier lines, which allow for address width and cycle type definition and data transfer protection.

The bus allows multiprocessing based on a master-slave principle. There are 5 essential types of functional modules (Fig. 10): master that can initiate data transfer, slave that responds to a master request, interrupter (usually a slave) that initiates an interrupt, interrupt handler that can receive interrupts, and an arbiter that monitors the status of the bus and provides bus access to the requestor, which requested ownership of the bus. Before the master can send the data, it has to request the bus. Prioritized requests can be made using daisy chain. Based on this, the arbiter decides which module gains the bus access.

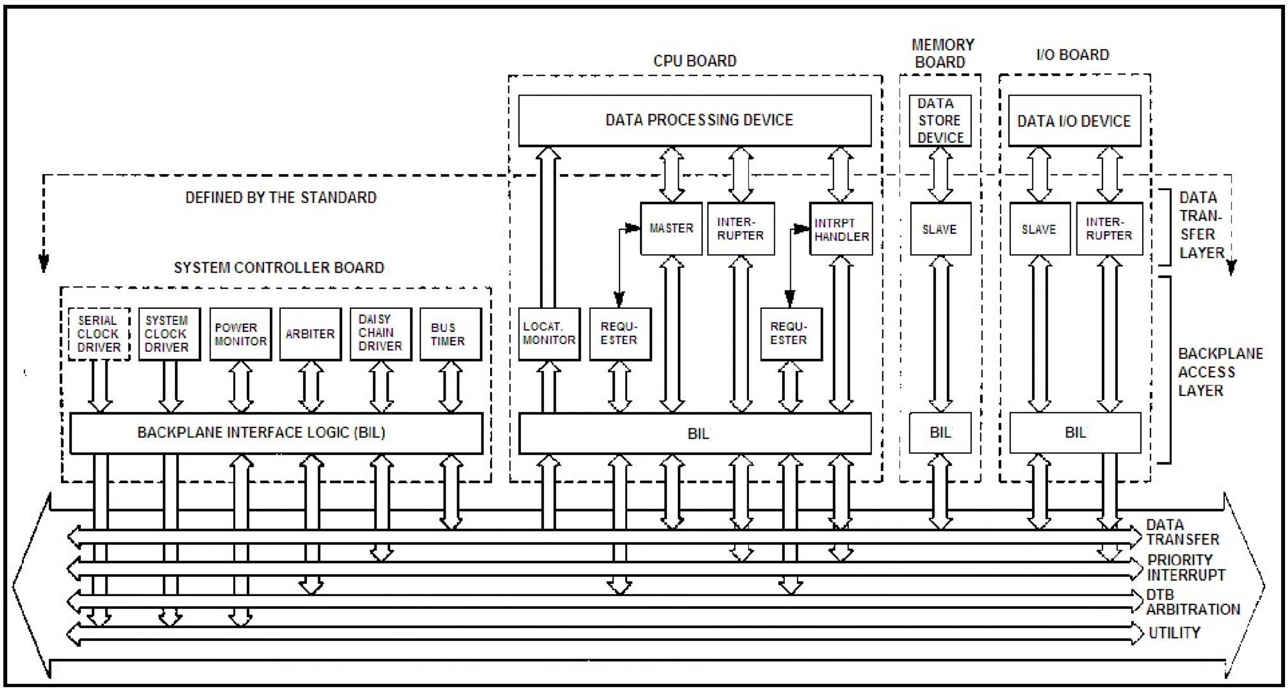


Fig. 10 – Block Diagram of the VMEbus Bus Interface [40]

This standard includes a high-speed asynchronous parallel data transfer bus (DTB), as shown in Fig. 10 [40]. It allows Master modules to direct the transfer of binary data between themselves and Slaves. The data transfer bus lines are grouped into three categories: Addressing lines, Data lines, and Control lines. The arbitration bus consists of six bused lines and four daisy-chained lines. It allows an Arbiter module and several requester modules to coordinate use of the DTB. Priority interrupt bus provides the signal lines needed to generate and service interrupts. It allows Interrupter modules to send interrupt requests to the various Interrupt Handlers. Utility bus provides utility functions such as periodic timing, system reset, initialization and diagnostic for the system, the power-up and power-down.

It is important to realize that, in addition to single word data transfer (single cycle read or write), the original VMEbus protocol also allows sequential transfers, BLT, to achieve higher speed by avoiding regular bus arbitration procedure, the feature extended in the additional specifications:

- BLT, typical block transfer to be generated by a master; normally, it is a direct memory access (DMA) transfer, when only the first transfer involves an address cycle and subsequent transfers use only data cycles
- MBLT (VME64), multiplexed block transfer, which allows 64-bit transfers by multiplexing data onto the 31 address lines and the LWORD* control line
- 2eBLT (VME64e), two edged block transfers, which means that data are driven on both the falling and rising edges of a clock, permitting the transfer of two bits of data per each cycle
- SSBLT, source synchronous block transfer, when the source of the data supplies the clock used to sample data at the destination [49]
- 2eSST (VME320), combining both two edged and source synchronous transfer.

5.2 RELATED STANDARDS

With VMEbus being an open standard and the developments in technology, as well as rapid expansion of new applications, a number of extensions were proposed, most of them standardized by an industrial organization named VITA (VMEbus International Trade Association). Chronologically, the first proposed extensions included VSB, VICbus:

- VSB [50]-[51] was designed as a local bus, to allow private access to resources that could still be shared, without the necessity to increase the traffic on a global parallel bus of a VMEbus crate;
- VICbus [52], which was created for the exactly opposite reason to VSB, to expand VMEbus systems to multiple crates, as a multiplexed, multi-master cable bus primarily intended for interconnecting backplane bus systems.

VXI (VME eXtension for Instrumentation) has been designed to meet the requirements of Automated Test Equipment (ATE) applications and improve the characteristics of GPIB. Built around VMEbus-size chassis and its parallel bus [53]-[54], it provides additional buses useful for instrumentation and allows two protocols: Word Serial Protocol to exchange ASCII-level messages, and shared memory protocol that allows device communicate only through direct register reads and writes. There are multiple successful applications of VXI in data acquisition and control [55].

A significant number of VMEbus extensions meant to improve its performance characteristics and respond to specific market demands [56]. VME64 is the original VMEbus extension standard [57], later enhanced by VME64x (1996), which added a number of new hardware features, VME64xP for applications in physics (1998), and VME 2eSST (2003) increasing the speed up to 320 Mb/s. VXS (VITA 41.0-2006) specified switched serial high speed fabric, and VPX known as VITA 46.x series, went even further by defining entirely new high-speed connectors to carry mappings for popular switched serial fabrics including Gigabit Ethernet, PCI Express, Serial RapidIO and InfiniBand.

5.3 CURRENT STATUS

Due to its open standard and non-proprietary nature, VMEbus reached an enormous users market, unprecedented for specialized computer design. VME based systems have been developed for and applied in areas such as military and aerospace, telecom, industrial control, medical, science, and many more. Around 1992, the VMEbus board market hit $1 bln in sales, and in 2011 it was still in the $600 mln range [58]. This is due to bus adaptive architecture for event-driven data acquisition and control applications, as opposed to data driven applications, in areas such as signal processing.

It is the new applications, which weigh heavily towards completely new designs, such as VPX (VME/PCI/eXtension), whose sales now overcome traditional VMEbus modules sales. But before the sophisticated designs of switched serial fabrics were even conceived, it was the much simpler serial buses, that appeared on the market in the 1990s and tended to dominate data acquisition and control systems, which appears to last till this day.

6. SERIAL BUSES: IEEE-1394 AND USB

The prototype of all serial buses, which were ever used in data acquisition and control, is an old RS232C protocol originally designed in the sixties of previous century to exchange information between a teletype and a modem. Due to its simple structure, it was quickly adopted for data transfer from measuring instruments to computers and for sending simple commands from computers to external instruments or control devices. The earliest paper on data acquisition and control with RS232 we could find dates back to 1980 [59], but there may have certainly been earlier ones.

Just for historical reasons, it is nice to look at the diagram presented in Fig. 11, which illustrates how sophisticated are the functions assigned to the RS232 interface in this project. It collects data from multiple instruments and external devices and passes them to the PDP 11/60 minicomputer.

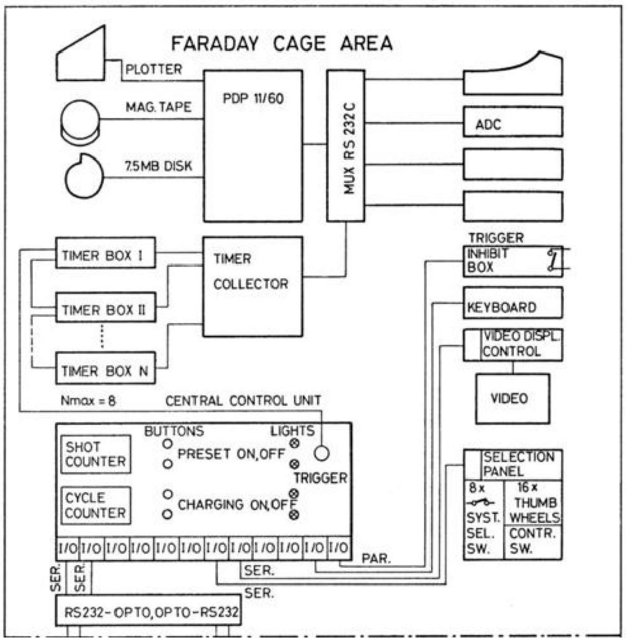


Fig. 11 – Early Data Acquisition System & RS232 [59]

While the functions and structure of data acquisition systems may not have changed that much since the 1980s, definitely the volume of data and transmission speed requirements have increased. Therefore more sophisticated bus-based serial interfaces have been designed, among them Firewire (IEEE Std 1394 [60]) and Universal Serial Bus (USB), now standardized as USB-3.1 [61].

6.1 FIREWIRE IN DATA ACQUISITION

Firewire was originally created by Apple in the late 1980's to handle traffic related to entertainment, including consumer electronics and multimedia, but it was soon realized that it can find application in other areas, such as computing clusters, networking, data acquisition, etc., which require high-speed data transfer. Indeed, it has been designed to accommodate fast data transfer (currently, up to 800 Mb/s) among computing nodes connected via a serial cable (up to approx. 70m).

The standard supports two different types of transfer modes between nodes. The transfer which requires periodic data transmission without guaranteed data delivery is called asynchronous transfer, while the kind of data transfer that is crucial to other applications and requires guaranteed delivery is called isochronous. It is isochronous transmission, which makes this bus especially interesting, because it guarantees a particular time slice each 125μs (8,000 isochronous cycles per second) dedicated to real-time traffic. Since a real-time device is guaranteed a time slot and isochronous communication takes over asynchronous, isochronous bandwidth is assured.

Modern bus protocols are typically described in terms of a layered approach, defining various aspects of bus operation according to the respective layers of the ISO/OSI Reference Model, especially Physical, Data Link and Application layers. Firewire bus has this structure very well organized and defined, which is illustrated in Fig. 12.

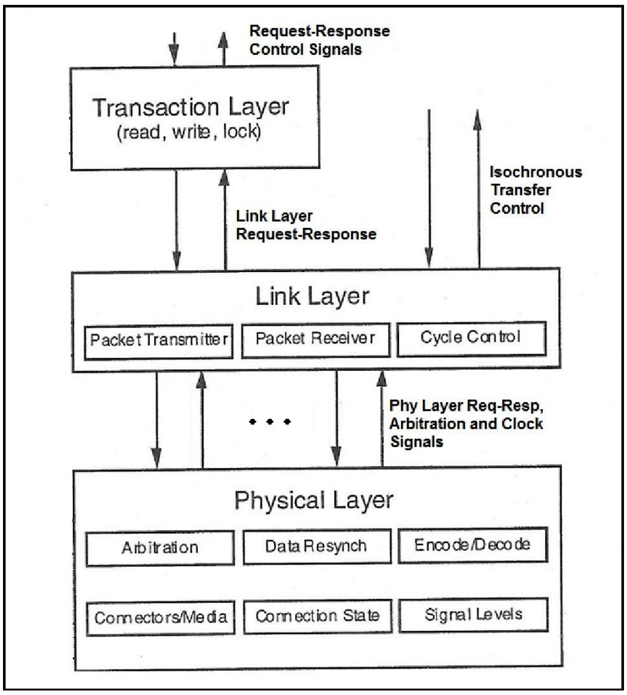


Fig. 12 – Firewire Protocol Layers [62]

The Transaction Layer defines a complete request-response bus protocol to perform transactions on the bus (read, write and lock) interacting with the application. It does not provide any services to the isochronous transfers, which bypass this layer, with signals going directly to/from the Link Layer. The Link Layer provides the acknowledge (confirmation of reception) to the Transaction Layer. It also does addressing, data checking and framing. As shown on the diagram in Fig. 12, it provides isochronous data services directly to the application. Finally, the Physical Layer translates the symbols used by the Link Layer directly into electrical signals on the bus media, which can be either backplane or cable [63].

Even though the Firewire design has not explicitly targeted data acquisition and control, its modern features and flexibility made it suitable for use in real-time systems, which was documented in a number of applications. Over the years, since its introduction, Firewire was used in many demanding data acquisition/control systems, where data from multiple sources required fast transfer to the central computer or remote units for storage or processing.

One such application [64] uses Firewire for distributed data acquisition in Relativistic Heavy Ion Collider (RHIC) beam position monitoring. All this formed a sophisticated control system, with four independent Firewire position monitoring devices connected to each of several VMEbus control systems. Another demanding application involved using Firewire as a data acquisition platform for Single-Photon Emission Computed Tomography (SPECT) project [65]. It allowed building standard PCI-based Firewire board modules for data transfer from three detector channels per module, and integrating them easily with a bigger SPECT control system. The authors stated that the use of standard Firewire platform radically reduces the cost and time of implementation of the overall system.

There were also attempts to apply Firewire technology in much bigger control systems, such as those used in factory automation [66]. Firewire high data transfer rate and guaranteed delivery times are the critical properties in applications that involve some sort of image transfer. For example, a factory quality control system may include multiple devices, all of which require high data throughput, among them: color photo scanner that provides precise imaging of fine details, an instrument measuring the outer contours of symmetrical machine parts, and an industrial camera based on a CMOS image sensor. The authors conclude that IEEE 1394 is well suited for high performance data transmission in industrial and factory automation applications.

In fact, the data transmission properties of IEEE 1394 were measured experimentally by the author [67], with industrial applications in mind, and – at that time – turned out to overpass respective properties of fast Eathernet channels. This was especially true for isochronous transmissions, however, at the cost of asynchronous channels, confirmation of the findings by other authors [68].

6.2 UNIVERSAL SERIAL BUS – USB

It turned out very soon that the IEEE 1394 bus had to compete with another serial bus design, Universal Serial Bus (USB), which soon became much more popular in data acquisition and control applications. USB was designed by a consortium of computer manufacturers and officially released in 1996. It was meant to significantly facilitate the attachment of peripheral devices to PC's, anything from as simple as a joystick to as complicated as a disk drive. Just like in the case of Firewire, users soon realized that it is perfectly meeting their needs for data acquisition and control in smaller and slower applications.

Since its original release, the USB standard has been expanded at least twice, to 2.0 (year 2000, 480 Mb/s) and 3.0 (year 2008, 5 Gb/s) and the USB interface is found in literally billions of digital devices on the market today. Its popularity in data acquisition and control is also unprecedented, due to the availability of chips and firmware, and ease of implementation of the basic functions, plus the increase of data transfer speed in successive standard updates (now 10 Gb/s in USB 3.1). USB protocol is also hierarchically layered, like that for IEEE 1394, with Physical Layer, Link Layer, and the uppermost layer named Protocol Layer, which interfaces directly to the application.

Table 2. Brief Comparison of IEEE 1394 and USB.

Feature	*IEEE 1394 (Firewire)*	*USB*
Year Standard Introduced	1995	1996
Main Design Objectives	Transfer Speed & High Perform.	Convenience and Simplicity
Standard Evolution	400 (1995) 800 (2002) S800T (2006) S1600 & S3200	USB 1.0 USB 2.0 (2000) USB 3.0 (2008) USB 3.1 (2013)
Architecture and Topology	Peer-to-peer Tree topology	Master-slave Tiered star
Data Transfer Types	Isochronous Asynchronous	Isochronous Interrupt Bulk transfers
Connectivity	Full Duplex	Half Duplex (Full since 3.0)
Maximum Theoretical Speed	400: 3 Gb/s 800: 6 Gb/s	1.0 – 12 Mb/s 2.0 – 480 Mb/s 3.0 – 5 Gb/s 3.1 – 10 Gb/s
*Sample [69] Sustained Data Transfer Rate**)	400Mb/s version Read: 38 Mb/s Write: 35 Mb/s	2.0 version Read: 33 Mb/s Write: 27 Mb/s

*) Multiple other comparisons exist, which give more insight into transfer rate, but the fact is that both buses are comparable and the choice depends on application.

A brief comparison of basic features of both serial buses is shown in Table 2. It is in fact very difficult to compare them, since they constantly evolve and designers enhance their characteristics, in particular, to make them faster, but the basic features remain the same.

Even though the USB protocols are not trivial from the electrical standpoint, the bus itself quickly became tremendously popular in data acquisition and control due its simplicity of use. Any instrument or device equipped with the USB interface can be immediately plugged into and connected with a computer or larger network for the purpose of data collection and analysis.

Applications of USB in this respect go into thousands and are hard to account for. A quick survey reveals systems as simple as those in education [70]-[72] to as complex as those (in chronological order) applied in vibration analysis [73], electrocardiogram recording [74], pulse height analysis [75], detection of heavy charged particles [76], measuring energy spectra of cosmic rays [77], spectrofluorometry [78], detector development in nuclear research [79], environmental measurements [80], electromyogram signal recording [81], Fourier mass spectrometry [82], medical ultrasound scanning [83]. Many of them include controller development to mention only the latest [84]-[86]. What was not true for other standards, USB spread over the entire globe, from the U.S. and Western Europe, to Malaysia, China, India, Japan and further.

7. SUMMARY AND CONCLUSION

The first part of this paper discussed basic concepts of standards in data acquisition and control, limited to computer interfaces, which are backbone of every such system. Following the perception [87] that until the mid-eighties: "Backlane bus standardization was a key activity and interconnecting multiple crates with point-to-point links was the way to grow beyond the limitations imposed by a single enclosure," special attention was given to the most successful interfaces and related computer buses, from CAMAC to VMEbus to serial buses.

The second part of the article will deal with advanced interfaces and interconnects and respective protocols, including wireless transmission and time-triggered architecture. It will also discuss further evolution of buses (Futurebus, PCI) and bus interfaces specific to selected applications, such as scientific measurements (PXI, LXI), power grid (Modbus), automotive (CAN), avionics (MIL-STD 1533), and aerospace, with heads-up on serial switched fabrics, increasingly popular in data-intensive applications. In this context bus safety and security will be covered as well.

8. ACKNOWLEDGEMENT

The preliminary version of this paper was presented as a keynote address at the Intelligent Data Acquisition and Advanced Computing Systems conference, IDAACS'2015, in Warsaw, Poland. The author is particularly grateful to the conference chairmen, Prof. Anatoly Sachenko and Prof. Wiesław Winiecki, for their guidance in the preparation of this talk.

Part of this work has been funded by a grant SBAHQ-10-I-0250 from the U.S. Small Business Administration (SBA). All SBA-funded projects are extended to the public on a nondiscriminatory basis.

9. REFERENCES

[1] E. J. Millett, "Digital techniques in laboratory automation," *Journal of Physics E: Scientific Instruments*, vol. 9, issue 10, pp. 794-802, October 1976.

[2] R. W. Dobinson, "Practical data acquisition problems in large high-energy physics experiments," in *Proceedings of the 6th CERN School of Computing*, Vraona-Attiki, Greece, September 14-27, 1980. Report CERN-1991-003, pp. 325-361.

[3] K. D. Müller, "ESONE activities," in *Proceedings of the International Conference on New Trends in Data and Signal Processing in Research ESONE RTD'95*, Warsaw, Poland, September 27-29, 1995, pp. 9-14.

[4] *Commission of the European Communities. CAMAC – A Modular Instrumentation System for Data Handling*, Report EUR 4100-EN, Brussels, 1969.

[5] *IEC-516-1975 A Modular Instrumentation System for Data Handling: CAMAC System*, International Electrotechnical Commission, Geneva, Switzerland, 1975.

[6] *IEEE Std 583-1975, Modular Instrumentation and Digital Interface System (CAMAC)*, Institute of Electrical and Electronics Engineers, Washington, DC, 1975.

[7] *Commission of the European Communities. CAMAC – Updates Specifications*, vol. 1. Report EUR 8500-EN, Brussels, 1983.

[8] P. Clout, *A CAMAC Primer, Report LA-UR 82-2718*, Los Alamos National Laboratory, Los Alamos, NM, December 1982.

[9] J. Zalewski, "Interconnect architectures: Old and new," in *Proceedings of the International Conference on New Trends in Data and Signal Processing in Research ESONE RTD'95*, Warsaw, Poland, September 27-29, 1995, pp. 15-21.

[10] J. Zalewski, "Real-time software architectures and design patterns: Fundamental concepts and their consequences," *Annual Reviews in Control*, vol. 25, issue 1, pp. 133-146, July 2001.

[11] J. Zalewski, "Real-time data acquisition in high-energy physics experiments," in *Proceedings of the IEEE Workshop on Real-Time Applications*, New York, May 13-14, 1993, pp. 112-115.

[12] A. Kovarik, "On the automatic registration of α-particles, β-particles and γ-ray and X-ray pulses," *Physical Review*, vol. 13, issue 4, pp. 272-280, 1919.

[13] W. LeCroy, "The history of data acquisition for high-energy physics – a corporate perspective," in *Proceedings of the 9th Fermilab industrial affiliates roundtable on applications of accelerators*, Batavia, Ill.,May 26-27, 1989, pp. 147-168.

[14] E. Meschi, "The DAQ needle in the big-data haystack," *Journal of Physics: Conference Series*, vol. 664, issue 3, pp. 2353-2360, 2015.

[15] H. Bisby, "The CAMAC interface and some applications," *The Radio and Electronic Engineer*, vol. 41, issue 12, pp. 527-537, December 1971.

[16] R. C. M. Barnes, I. N. Hooton, "The CAMAC system of modular instrumentation," *IEEE Transactions on Nuclear Science*, vol. 16, issue 5, pp. 76-80, 1969.

[17] K. Tradowsky, *CAMAC – Ein System rechnergefürter Elektronik – Prinzip und Anwendungen*, Report KFK 1241, Kernforschungszentrum Karlsruhe, Juli 1970.

[18] L. Costrell, "CAMAC instrumentation system – introduction and general description," *IEEE Transactions on Nuclear Science*, vol. 18, issue 2, pp. 3-8, 1971.

[19] R. L'Archeveque, G. Yan, *A Review of the CAMAC Concept*, Report AECL-3806, Atomic Energy of Canada Limited, Chalk River, Ont., March 1971.

[20] R. C. Furst, J. D. Wiedwald, *CAMAC System for Remote Data Acquisition*, Report UCRL-73968, Lawrence Radiation Laboratory, Livermore, Calif., June 1972.

[21] F. A. Kirsten, "Some characteristics of interfaces between CAMAC and small computers," *IEEE Transactions on Nuclear Science*, vol. 18, issue 2, pp. 39-45, 1971.

[22] F. Iselin et al., *Pattern A – Type 021, CERN NP CAMAC Note 8-00*, CERN, Geneva, Switherland, January 1969.

[23] H. Halling, K. Zwoll. K. D. Müller, "Versatile PDP-11 CAMAC crate controller for nuclear data acquisition and processing," *IEEE Transactions on Nuclear Science*, vol. 19, issue 1, pp. 699-703, February 1972.

[24] L. Babiloni, E. de Agostino, A. Ostrowicz, *CAMAC Module Interface between CAMAC System and IBM 360/44 Computer*, Report CNEN RT/EL(72)5, Comitato Nazionale Energia Nucleare, Rome, Italy, 1972.

[25] I. Tradowsky-Thal, *CAMAC Bibliographie*, Report KFK 1471, Kernforschungszentrum Karlsruhe, September 1971.

[26] I. Tradowsky-Thal, *CAMAC Bibliographie Supplement*, Report KFK 1671, Kernforschungszentrum Karlsruhe, Oktober 1972.

[27] R. C. M. Barnes, A. R. C. Whiteman, *CAMAC Bibliography Covering Period 1968-71*, Report AERE-BIB-180, Atomic Energy Research Establishment, Harwell, UK, March 1972.

[28] H. J. Stuckenberg, "CAMAC bibliography," *CAMAC Bulletin*, No. 13 (Supplement B), pp. 1-36, September 1975.

[29] J. Zalewski, *CAMAC Bibliography*, Report ESONE/BIB/01, Commission of the European Communities, Luxembourg, 1978.

[30] P. J. Ponting, "Lessons learnt from fastbus," in *Proceedings of the ESONE VMEbus Working Group Workshop on New Backplane Bus Architectures*, CERN, Geneva, Switzerland, March 22-23, 1990. Report CERN CN 90/4, pp. 5-20.

[31] Anon., "Buses at the crossroads," *CERN Courier*, pp. 5-6, June 1990.

[32] *IEEE Std 488-1975 – Standard Digital Interface for Programmable Instrumentation*, Institute of Electrical and Electronics Engineers, Washington, DC, 1975.

[33] R. S. Larsen, "Introduction to the fastbus standard data bus," *Interfaces in Computing*, vol. 1, issue 1, pp. 19-31, May 1982.

[34] *IEEE Std 960-1986 – FASTBUS Modular High-Speed Data Acquisition and Control System*, Institute of Electrical and Electronics Engineers, Washington, DC, 1986.

[35] P. J. Ponting, *Fastbus: The Development of a Standard and its Infrastructure*, Report CERN/ECP 91-2, CERN, Geneva, Switzerland, January 1991.

[36] J. Zalewski (Ed.), *Advanced Multimicroprocessor Bus Architectures*, IEEE Computer Society Press, Los Alamitos, Calif., 1994.

[37] *IEEE Std 675-1979 – Multiple Controllers in a CAMAC Crate*, Institute of Electrical and Electronics Engineers, Washington, DC, 1979.

[38] R. DeBock, "VERSAbus – A multiprocessor bus standard – and VMEbus – its Eurocard counterpart," *Microprocessors and Micro-*

systems, vol. 6, issue 9, pp. 475-481, November 1982.

[39] W. Fischer, "IEEE P1014 – A standard for the high-performance VME bus," *IEEE Micro*, vol. 5, issue 1, pp. 31-40, February 1985.

[40] *IEEE Std 1014-1987 – A Versatile Backplane Bus: VMEbus*, Institute of Electrical and Electronics Engineers, Washington, DC, 1987.

[41] *IEEE Std 1296-1987 – High-Performance Synchronous 32-Bit Bus: Multibus II*, Institute of Electrical and Electronics Engineers, Washington, DC, 1987.

[42] W. M. Clemow, "Introduction to the Multibus II architecture," *Interfaces in Computing*, vol. 3, issues 3-4, pp. 277-286, 1985.

[43] M. D. Rap, R. S. Tetrick, "P1296: The interprocess communication standard," *IEEE Micro*, vol. 6, issue 3, pp. 72-77, June 1986.

[44] J. Mahoney, "Overview of Multibus II architecture," *Supermicro Journal*, issue 4, pp. 58-67, January/February 1990.

[45] C. Beardsmore, "Multibus II: A bus architecture for fault resilient systems," in *Proceedings of the IEE Colloquium on Designing Resilient Architectures*, London, November 15, 1991, *IEE Digest*, No. 170, pp. 3/1-3/11.

[46] M. Lobelle, "VME bus interfacing: A case study," *Interfaces in Computing*, vol. 1, issue 3, pp. 193-210, August 1983.

[47] C. Parkman, *VMEbus at CERN – A Brief Review*, Report CERN DD/85/28, CERN, Geneva, Switzerland, November 1985.

[48] W. F. Jones et al., "A VMEbus based, real time sorter design for positron emission tomography," *IEEE Transactions on Nuclear Science*, vol. 33, issue 1, pp. 601-604, February 1986.

[49] J. Regula, "The proposed SSBLT standard doubles the VME64 transfer rate," *IEEE Micro*, vol. 12, issue 2, pp. 64-71, April 1992.

[50] S. Pri-Tal, "The VME subsystem bus," *IEEE Micro*, vol. 6, issue 2, pp. 66-71, April 1986.

[51] J. Alexander, "Evolution and use of the VME subsystem bus – VSB," *Microprocessors & Microsystems*, vol. 10, issue 6, pp. 307-312, July/August 1986.

[52] C. F. Parkman, "VICbus: VME inter-crate bus – A versatile cable bus," *IEEE Transactions on Nuclear Science*, vol. 39, issue 2, pp. 77-84, 1992.

[53] R. Dettmer, "The VXI bus: An open standard for modular instrumentation," *IEE Review*, vol. 35, issue 9, pp. 327-330, October 1989.

[54] *IEEE Std 1155-1992 – VMEbus Extensions for Instrumentation: VXIbus*, Institute of Electrical and Electronics Engineers, Washington, DC, 1992.

[55] C. Ender, "Has VXIbus a role in physics?" in *Proceedings of the ESONE VMEbus Working Group Workshop on New Backplane Bus Architectures*, CERN, Geneva, Switzerland, March 22-23, 1990. Report CERN CN 90/4, pp. 56-66.

[56] S. Pri-Tal, "Giving the VME system a mid-life kicker," in *Proceedings of the ESONE VMEbus Working Group Workshop on New Backplane Bus Architectures*, CERN, Geneva, Switzerland, March 22-23, 1990, Report CERN CN 90/4, pp. 44-55.

[57] *ANSI/VITA 1-1994 – American National Standard for VME64*, VMEbus International Trade Association, Scottsdale, Ariz., 1994.

[58] R. Alderman, "VME turns 30," *VME and Critical Systems*, vol. 29, issue 3, pp. 7, Fall 2011.

[59] J. B. Lister, A. Simik, "Control, acquisition and data retrieval for the TCA Tokamak," in *Proceedings of the 11th Symposium on Fusion Technology*, Oxford, UK, Sept. 15-19, 1980, vol. 2, pp. 689-694.

[60] *IEEE Std 1394-1995 – High Performance Serial Bus*, Institute of Electrical and Electronics Engineers, Washington, DC, 1995.

[61] *IEC 62680-3-1 Ed. 1.0. Universal Serial Bus Interfaces for Data and Power – Part 3-1*, International Electrotechnical Commission, Geneva, Switzerland, 2016.

[62] M. Teener, *A Bus on a Diet – The Serial Bus Alternative. An Introduction to the P1394 High Performance Serial Bus*, Advanced Multimicro-processor Bus Architectures (J. Zalewski, Ed.), IEEE Computer Society Press, Los Alamitos, Calif., 1994, pp. 180-194.

[63] G. Marazas, "IEEE 1394: Status and growth path," *IEEE Micro*, vol. 16, issue 3, pp. 75-78, June 1996.

[64] T. J. Shea et al., "Evaluation of IEEE 1394 serial bus for distributed data acquisition," in *Proceedings of the 1997 Particle Accelerator Conference*, Vancouver, BC, May 12-16, 1997, vol. 2, pp. 2501-2504.

[65] A. Rillbert et al., "A general firewire data acquisition platform applied to SPECT," in *Proceedings of the 1999 IEEE Nuclear Science Symposium Conference Record*, Seattle, Wash., October 24-30, 1999, pp. 489-493.

[66] M. Scholles et al., "IEEE 1394 firewire system design for industrial and factory automation applications," in *Proceedings of the 8th International Conference on Emerging Technologies and Factory Automation ETFA*

2001, Antibes – Juan les Pins, France, October 15-18, 2001, pp. 627-630.

[67] C. Williamsson, D. Williamsson, J. Zalewski, "A study of cluster computing over IEEE 1394," in *Proceedings of the 2nd Swedish-American Workshop on Modeling and Simulation SAWMAS-2004*, Cocoa Beach, FL, February 2-3, 2004, pp. 120-126.

[68] D. Steinberg, Y. Birk, "An empirical analysis of the IEEE-1394 serial bus protocol," *IEEE Micro*, vol. 20, issue 1, pp. 58-65, January-February, 2000.

[69] G. Ramamurthy, K. Ashenayi, "Comparative study of the firewire IEEE-1394 protocol with the universal serial bus and Ethernet," in *Proceedings of the 45th Midwest Symposium on Circuits and Systems MWSCAS-2002*, Tulsa, Oklahoma, August 3-7, 2002, pp. II-509-512.

[70] B. Murovec, S. Kocijancic, "A USB-based data acquisition system designed for educational purposes," *International Journal of Engineering Education*, vol. 20, issue 1, pp. 24-30, 2004.

[71] Z. Lucev, H. Dzapo, M. Cifrek, "Multifunctional configurable USB data acquisition system," in *Proceedings of the IEEE International Instrumentation Measurement Technology Conference IMTC 2008*, Vancouver Island, Victoria, May 12-15, 2008, pp. 1206-1209.

[72] J. P. Grinias et al., "An inexpensive, open-source USB Arduino data acquisition device for chemical instrumentation," *Journal of Chemical Education*, vol. 93, pp. 1316-1319, 2016.

[73] A. Podgorski, R. Nedwidek, M. Pochmara, "Implementation of the USB interface in the instrumentation for sound and vibration measurement," in *Proceedings of the IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2003*, Lviv, Ukraine, September 8-10, 2003, pp. 159-163.

[74] J. M. Najeb, A. Ruhullah, S. H. Salleh, "12-channel USB data acquisition system for QT dispersion analysis," in *Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing ROVISP'2005*, Penang, Malaysia, July 20-22, 2005.

[75] Z. Guzik et al., "TUKAN – An 8k pulse height analyzer and multi-channel scanner with a PCI or a USB interface," *IEEE Transactions on Nuclear Science*, vol. 53, issue 1, pp. 231-235, February 2006.

[76] Z. Vykydal, J. Jakubek, S. Posposil, "USB interface for Medipix2 pixel device enabling energy and position-sensitive detection of heavy charged particles," *Nuclear Instruments and Methods in Physics Research A*, vol. 563, pp. 112-115, 2006.

[77] J. H. Yoo et al., "Improved data acquisition system for CREAM-III," in *Proceedings of the 30th International Cosmic Rays Conference ICRC'07*, Merida, Mexico, 2007, vol. 2, pp. 401-404.

[78] D. A. Fabila et al., "Development of a spectrofluorometer with USB interface for in vivo measurements in surgical procedures," in *Proceedings of the 2nd Circuits and Systems for Medical and Environmental Applications Workshop CASME'2010*, Merida, Mexico, December 13-15, 2010, pp. 1-4.

[79] H. Jiang et al., "A USB-2 based portable data acquisition system for detector development and nuclear research," *Nuclear Instruments and Methods in Physics Research A*, vol. 652, pp. 483-486, 2011.

[80] M. Caciotta et al., "Development of an USB data acquisition system for power quality and smart metering applications," in *Proceedings of the 11th International Conference on Environment and Elctrical Engineering EEEIC'2012*, Venice, Italy, May 18-25, 2012, pp. 835-839.

[81] S. Boukhenous et al., "A USB based data acquisition system for EMG signal recording," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications WoSSPA'2013*, Algiers, Algeria, May 12-15, 2013, pp. 230-232.

[82] D. Li et al., "Design and implementation of data acquisition system based on FPGA and USB interface in Fourier-transform mass spectrometer," in *Proceedings of the 8th International Conference on Biomedical Engineering and Informatics BMEI'2015*, Shenyang, China, October 14-16, 2015, pp. 169-173.

[83] J. R. Raj, S. M. K. Rahman, S. Anand, "Micro-controller USB interfacing with MATLAB GUI for low cost medical ultrasound scanners," *Engineering Science & Technology*, vol. 19, pp. 964-969, 2016.

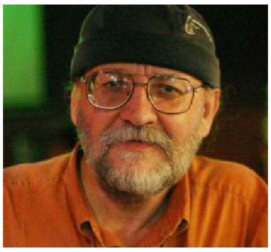
[84] D. Hercog, B. Gergic, "A flexible microcontroller-based data acquisition device," *Sensors*, vol. 14, pp. 9755-9775, 2014.

[85] K. Mroczek, "USB FIFO interface for FPGA based DAQ applications," in *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced*

Computing Systems IDAACS'2015, Warsaw, Poland, September 24-26, 2015, pp. 666-671.

[86] V. U. Patil, A. R. Kapur, "Real-time alert data acquisition system using dynamic IP embedded webserver by USB modem," *Procedia Comp. Science*, vol. 49, pp. 187-193, 2015.

[87] D. Calvet, "A review of technologies for the transport of digital data in recent physics experiments," in *Proceedings of the 14th IEEE-NPSS Real Time Conference RTC'05*, Stockholm, June 4-10, 2005, pp. 629-633.



__Janusz Zalewski__ is a professor of Computer Science and Software Engineering at Florida Gulf Coast University, in Ft. Myers, Florida, USA. He previously worked at nuclear research labs in Europe and the U.S. and consulted for the government and industry. He also held fellowships at NASA and Air Force Research Labs. He worked on the original working groups designing Multibus II (IEEE Std 1296) and Firewire (IEEE Std 1394). His current research interests include real-time embedded and cyberpysical systems, safety and security of computer systems and software, and software engineering education.