

IPMI Controller Based on Dedicated Microcontroller for ATCA Carrier Board

Piotr Perek, Dariusz Makowski, *Member, IEEE*, Paweł Prędko, and Andrzej Napieralski, *Senior Member, IEEE*

Abstract—The Advanced Telecommunications Computing Architecture (ATCA) specification allows to meet the newest trends in high speed communication technologies. Furthermore, it provides manageability, availability and exceptional reliability at 99.999% level. Therefore, this architecture is perfect to use in control systems of complex projects like the Free-Electron Laser in Hamburg (FLASH) or the X-ray Free Electron Laser (X-FEL) that work with high-frequency signal processing, use high-speed communication protocols such as PCIe, Gigabit Ethernet or RocketIO and require continuous stable operation. Modular construction allows for flexible system configuration. What is worth emphasizing, in contrast to previous solutions like, VME (Versa Module Eurocard), it is possible to change the system configuration without the need for power shutdown. Hot-plug functionality is delivered by Intelligent Platform Management Interface (IPMI) system. For this reason, all ATCA units, which can be replaced in the field should have Intelligent Platform Management Controller (IPMC) that provides IPMI functionality.

The IPMC is responsible for management and control of module on which it is placed. Due to the fact that the IPMC operates independently of another components, it allows to activation and deactivation other functional blocks of the module. For this purpose it has to communicate with the Shelf Manager according to IPMI standard. The IPMC monitors also vital operational parameters such as temperature, voltage, current and reports all abnormalities to the Shelf Manager.

This paper presents the IPMC for ATCA Carrier Board with three AMC Bays. All IPMC functionalities required by ATCA specification are in this case fulfilled by microcontroller dedicated for IPMI management in xTCA systems.

Index Terms—Intelligent Platform Management Controller, Advanced Telecommunications Computing Architecture, Carrier Board, IPMI

I. INTRODUCTION

THE Advanced Telecommunication Computing Architecture (ATCA) is the newest open architecture of computing components, which offers high levels of modularity and configurability [1], [2]. It also allows to use high frequency communication protocols. The basic elements of the ATCA are Front Boards or Rear Boards which can have variety functionalities and may be combined in various configurations according to the needs of the system. The characteristic type of modules is the Carrier Board based on the Front Board which may manage up to four Advanced Mezzanine Cards (AMCs). It allows to extend the basic functionalities of the Board and gives the possibility of quick and easy system reconfiguration [3]. The main tasks of the ATCA system are power management, providing connections between modules and devices supervision. However, this specification has no influence on the functionality of the whole system,

so it is possible to use proprietary modules designed to perform specific tasks [4]. Therefore, this architecture gains increasing popularity in non-telecommunication fields e.g. physics machine like: accelerators, detectors and similar applications [5], [6], [7], [8]. A good example is the X-Ray Free Electron Laser (XFEL), which is being built in the research center Deutsches Elektronen-Synchrotron (DESY) in Hamburg. ATCA is one of the candidates taken into account as a base platform for its LLRF control system [9], [10], [11]. The tasks of such a system include acquisition and processing of data from the accelerator with frequency of about 100 MHz and controlling the feedback circuits. Therefore, it is important to ensure continuous, fast and reliable connections between modules. The ATCA specification gives the possibility of placing all links on the backplane. It allows to avoid cables and related problems such as the necessity of connecting/disconnecting modules during each insertion/extraction or damaged wires. It also protects against human errors, which are easy to make when connecting a lot of modules with hundreds of cables. An important issue is also to ensure stable operation and the required control system uptime. It is usually assumed that such machines should be available 24/7 [5]. To increase the reliability of the ATCA system hot-swap functionality, allowing exchange of the module without power shutdown and continuous monitoring of operating parameters is introduced. The vital parameters include temperature, voltage, current, clock and trigger signals, but also the state of communication with System Manager. It is worth emphasizing that each system should have two Shelf Managers. At any given time only one is active and oversees the system but it can be replaced by the second in case of failure. ATCA specification also requires the use of two redundant buses to communicate with the Shelf Manager which increased safety.

All the functionality described above caused that it became necessary to introduce a special management layer based on the Intelligent Platform Management Interface (IPMI) standard. This imposes an obligation of equipment for most of the modules in the Intelligent Platform Management Controller (IPMC) [12], [13], [14]. The IPMC should always be operational, when management power is on, which gives the possibility to activate (turn on payload power, open connections) the modules after insertion to the shelf and their deactivation before extraction. Thanks to this solution the modules exchange during the operation of the system is safe.

II. ARCHITECTURE OF ATCA DEVELOPMENT BOARD

In order to allow development and tests of IPMC functionalities was designed prototype ATCA Carrier Board. It is typical ATCA Front Board with three AMC Bays. It also meets all the requirements of a Hot-Swap FRU [4] i.e. it has an embedded IPMC that enables control of the board and the installed AMC Modules through the IPMI infrastructure and provides the hot-swap functionality. However, due to the fact that described board was prepared in order to develop of IPMC functionalities on new microcontrollers family, it contains only elements required by ATCA specification for IPMI management, but it does not have other subsystems e.g. interconnections for AMC communication interfaces or devices for signal processing such as DSPs or FPGAs.

The most important element of this board is microcontroller which realizes IPMC functionalities. However, on account of significant role which it fulfills, it will be described in more detail in subsequent section.

The aforementioned Hot-Swap mechanism is possible thanks to separation of power to Management (3.3 V) and Payload (12 V) as shown in Figure 1. Because the supply voltage of the Carrier Board equals 48 V a DC/DC converter, which produces both voltages, is used. On the board both voltages are available immediately after inserting. However, because the role of this Carrier Board is limited to the supervision of AMC modules and does not imply any other functionality only the Management voltage is used. Payload Power is needed to ensure the power supply of AMC modules. For safety reasons buffers for AMC modules, which allow the IPMC to turn the voltages on and off, have been applied. The Module Management Controller (MMC) and all the sensors on the AMC are supplied with the Management Power. As was the case with the Carrier Board, here too this power is applied immediately after the module is inserted. Thus, the AMC is visible in the system immediately after inserting which allows the Carrier Board to manage it, control it and, most importantly, carry out the activation and deactivation processes. During activation, the Payload Power is switched on which provides full functionality of the equipment. On the

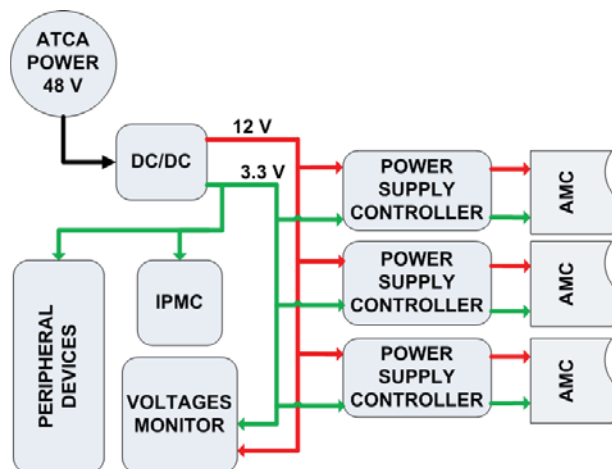


Fig. 1. Distribution of power supply on Carrier Board

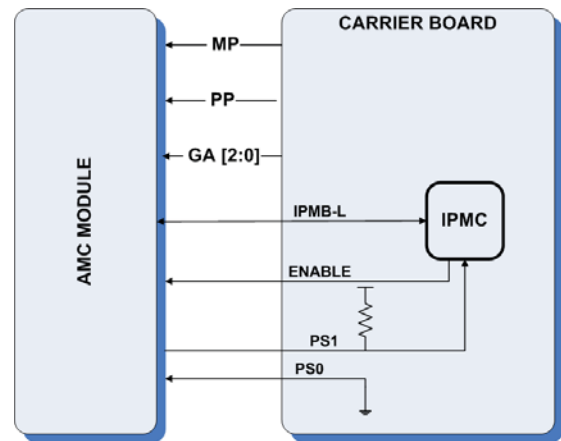


Fig. 2. Interconnects required to management of AMC modules

other hand, in case of module deactivation and extraction all the voltages are turn off. It is worth emphasizing that the power supply controllers continuously monitor states of management and payload voltages. The feedback is given for IPMC by two Power Status Outputs, one for each voltage, which inform about actual state of power output.

To support the AMC modules, in addition to power, it is necessary to ensure a few other connections, defined in the specification [15] and shown in Figure 2. Two signals (PS0 and PS1) are used to detect the AMC's presence in the connector. On the Carrier Board PS0 is connected to ground and PS1 is pulled-up, while on the module these two pins are shorted. Therefore a low level on pin PS1 signalizes a presence of the AMC module in the slot. The ENABLE signal is a kind of feedback to AMC Module. It is an active low input which informs the MMC that the module is properly inserted. The state of ENABLE signal is controlled by IPMC of Carrier Board which sets low state on it after detection of module presence and power switching on.

Local Intelligent Platform Management Bus (IPMB-L) is an I2C-compatible bus and is used for communication between the Module and the Carrier. On the described board it has been designed as three independent I2C buses, each attached to a separate channel in the microcontroller, so that the communication with the modules is fast enough and stable. The signals of IPMB-L on AMCs with signals on Carrier Board are connected using I2C buffers which are switched on only when ENABLE signal is active. This solution ensures a high impedance state when given slot is not occupied and protects against disruptions on the bus. Because the I2C standard requires that all devices have their own unique addresses it is necessary to inform each module in the ATCA system what address it should use. In case of the Carrier Board this address is assigned on the basis of Hardware Address signals according to ATCA specification [4]. However AMC Modules have no connection to the backplane and the Shelf Manager, so the addresses for them should be given by the Carrier Board. Three signals consisting of the Geographic Address (GA[2:0]) are used for this purpose. Each of these pins can be connected to Logic Ground (G), Management Power (P), or be left unconnected (U). All combinations of

the states of these pins are assigned to a specific address [15]. Examples of combinations used in the described board are present in Table I.

TABLE I
GEOGRAPHIC ADDRESSES USED IN THE CARRIER BOARD
AND CORRESPONDING IPMB-L ADDRESSES

Geographical Address [2:0]	IPMB-L address
UGU	7Ah
UUG	7Ch
UUP	7Eh

Moreover, two additional I2C buses used to communicate with the Shelf Manager defined in the ATCA specification as IPMB-0, are present on the Carrier. As already mentioned, they are redundant. Under normal conditions communication takes place in parallel over two buses, but in case of problems with one of them, the whole transmission may resume with use of the second one only. As already mentioned, in accordance to requirements of ATCA specification, during communication over IPMB-0, the IPMC should use individual address. It is calculated on the basis of states of eight pins called Hardware Address. The most often HA pins are directly connected to ground or power supply on the backplane and the Shelf Manager has no influence on their states. In case of described board, this address is read by IPMC using I2C-bus I/O expander.

Two temperature sensors and an advanced system monitor which allows to measure voltages, current and temperature, and report any cases of exceeding the critical thresholds have also been used on the board. They all communicate over the I2C interface, so an additional bus for them was intended. It is also possible to access all of them from the level of IPMI. This allows remote supervision of the entire system and detection of anomalies in real time.

III. INTELLIGENT PLATFORM MANAGEMENT CONTROLLER FOR ATCA CARRIER BOARD

A. Requirements for IPMC

As already mentioned, the described board is used to develop IPMC and test of its functionalities required by ATCA specification. Therefore the most important element on this board is the microcontroller which fulfills the role of the IPMC. Its main task is to ensure communication with the Shelf Manager. Messaging in ATCA is based on the I2C standard and uses a request-response protocol i.e. each request message should be confirmed by the target device. If the sending device does not receive the confirmation it should retry sending the request. Such a solution significantly improves the quality of communication, because it protects against losing messages. The IPMI standard defines the format of messages sent between modules in the system. Each of them consists of a header and data. The header always has a fixed length and includes fields such as addresses of the requester and the responder, and a command identifier while the data are of varying length depending on the command. All messages have two checksums calculated for the header

and the entire message separately. This mechanism protects the communication against transmission errors.

The IPMC shall also use addressing specified by ATCA. One of its components the Hardware Address, is mentioned above. It consists of seven bits which inform directly about address assigned to the module and another one, parity bit, which allows to detect if any of the address pins was damaged (e.g. bent or broken).

In order to ensure full functionality the IPMC should implement all the mandatory commands described in the ATCA specification [4]. They are divided into groups based on their functionality. One of them is associated with the Hot-Swap mechanism. It gives the possibility of inserting or extracting the module from the system during its normal operation without switching the power off. After insertion each module has to be activated in order to turn on the Payload Power while it should be deactivated before extraction. Both processes are described using eight operational states called M-States. Each module has to go through all the states, from insertion, the start of activation, up to full activity and vice versa, through deactivation to the state in which it can be safely removed from the system. The IPMC should report every transition between the states to the Shelf Manager using "Event Messages". However these commands always concern sensors, so the ATCA specification introduces a virtual "Hot Swap Sensor" that stores information about the current M-State of the board. The Hot-Swap mechanism requires also an introduction of two additional elements of operator interface: the Handle Switch and the Blue LED. The Handle Switch is connected to handle used to insert/extract Board into/from the Shelf and signals its opening or closing to the IPMC, while the Blue LED informs an operator about current operational state of the board. Its state is strictly associated with the current operational state of the board e.g. if the Blue LED is turned off that means the board is in the normal operational state and cannot be extracted safely. However, if the Blue LED is turned on, the board may be extracted safely, because Payload Power is switched off.

Another group of commands provides the ability to read data from the FRU Inventory Area. It is a special structure containing information about the board, such as manufacturer ID, product ID, serial number, as well as power supply or management data. This structure should be stored in a non-volatile memory e.g. in an external EEPROM supported by IPMC, which provides commands to read/write this information by other units. The data from this structure is collected by the Shelf Manager during activation process and for example on this basis it establishes connections between modules into the shelf.

Furthermore, the IPMC monitors the work of the Carrier Board through the use of numerous sensors e.g. temperature, voltage or current and informs the Shelf Manager about their current states. To allow the representation of the sensors in the system structures called Sensor Data Records (SDRs) were introduced. Each sensor which is available in the system has a corresponding entry in the SDR. In case of the described Carrier Board structures of Full Sensor Record type (all types of SDRs are described in specification of IPMI [16]) are used.

Each of them contains information about:

- owner of the sensor, i.e. I2C address of board,
- sensor number – unique for each sensor behind a given address,
- sensor capabilities and type, e.g. temperature, voltage, etc.,
- sensor units and coefficients needed to convert from raw to unit-based value,
- values of sensor thresholds and hysteresis.

It is worth emphasizing that the SDR does not include information about the current sensor value. It can be read only using one of the required commands. The IPMC also provides commands to read data from SDRs, setting/getting values of thresholds and hysteresis.

Besides the entries associated with sensors of physical properties the described IPMC has three additional SDRs. The first contains information about the aforementioned Hot Swap Sensor. The second describes another sensor required by ATCA. It is the Physical IPMB-0 Sensor, whose task is to monitor of the state of IPMB-0. The last of them is of the Management Controller Device Locator Record type and stores information about the controller and device capabilities.

An important functionality is also power and cooling management [17]. The IPMC is responsible for power negotiation during module activation. During this process the Shelf Manager collects information about power requirements of activated module (Carrier Board or AMC). These information are stored in FRU Information structure provided by IPMC. On this basis the Shelf Manager calculates whether the shelf can provide required power and informs the IPMC on which power level given module can operate. If the shelf cannot provide required power, the module can operate with reduced power if it supports that functionality. As already mentioned in the case of AMC modules the IPMC is responsible for turning the Payload and Management Power on and off. Moreover it should monitor parameters such as voltage, temperature or current and inform the Shelf Manager about any abnormalities.

It should also be noted that the controller of the Carrier Board is an intermediary in communication between AMC Modules and the Shelf Manager. This means that incoming messages from the Shelf Manager must be adapted and forwarded to AMCs, and then responses from modules must be converted to appropriate frames for the Shelf Manager.

B. Structure of IPMC

Communication of all three modules with the Shelf Manager, especially during activation and deactivation, causes significant traffic on both IPMB-L and IPMB-0 buses. In the previous version of Carrier Boards for XFEL, the role of IPMC is fulfilled by Atmel Atmega1281 microcontroller cooperating with Xilinx Spartan 3 Field Programmable Gate Array [18], [19], [20]. This microcontroller has only one built-in I2C interface, so all AMC Modules were connected to one bus. It resulted in frequent collisions and blocking the bus by one of the modules. Furthermore, it was necessary to use external parallel bus to I2C bus converters for IPMB-0 and

peripheral buses. They were potential sources of errors during communication. Besides, in fact, the functionality of IPMC is fulfilled by two devices: microcontroller and FPGA, which acts as I/O ports expander and is responsible for monitoring on-board peripherals.

The described solution of the Carrier Board uses Renesas H8S/2166 microcontroller that was developed especially for communication applications such as ATCA or μ TCA. Its central processing unit with an internal 16-bit architecture works with a 32 MHz clock. It has embedded 512 kbytes of ROM and 40 kbytes of RAM. The peripheral functions offered by this microcontroller are: five various timers including watchdog timer, serial communication interface which supports among other things UART, LPC interface and also ADC and DAC converters. About one hundred of I/O pins allow for the elimination of the FPGA device. But most importantly, it is equipped with six independent I2C channels, which allow to connect to all the required IPMI buses without using any external I2C bus controllers. Figure 3 shows the connections of all I2C modules of the microcontroller. Two channels are connected to IPMB-A and IPMB-B, three consecutive, consisting of IPMB-L, are used to communication with AMC Modules while the last of them ensures the communication with peripheral devices. To the last, peripheral bus the external devices are connected such as sensors, expanders, etc.

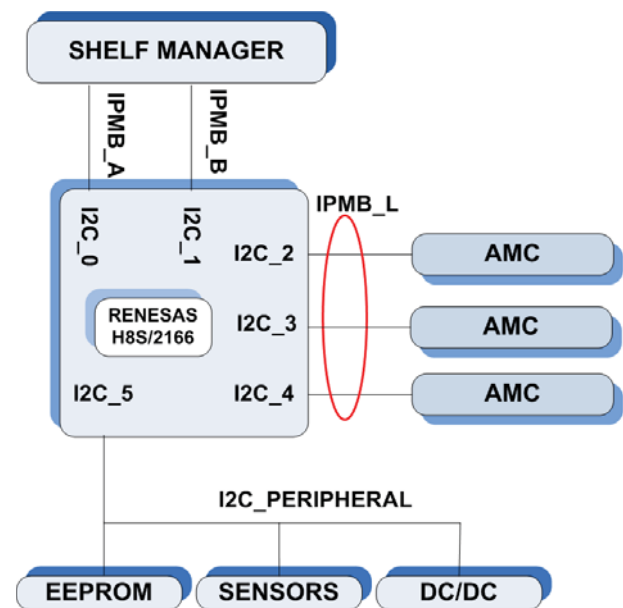


Fig. 3. Connections of microcontroller's I2C interfaces

Besides communication with the Shelf Manager and the AMC Modules, the IPMC handles many other peripheral devices. These include LEDs and switches, e.g. Hot Swap Handle and Blue LED. The controller also has access to an external EEPROM, which can store FRU Information and possibly other crucial data and an IO expander connected to the peripheral I2C bus, which is used to read the Hardware Address. It also handles external supervisory circuits with watchdog and UART interface for sending debug information.

As previously mentioned it is possible to communicate with all the sensors and also the DC/DC converter over the I2C interface.

C. Software and Development tools

The program for the microcontroller was written using High-performance Embedded Workshop (HEW) [21] and compiled with free GNU toolchains - KPIT GNU Tools [22]. HEW is an IDE provided by the manufacturer of the microcontroller - Renesas Company. It allows to cooperate with the E10A-USB programmer/debugger. Thus, it is possible not only to program the microcontroller, but also to preview the internal registers of the processor and the peripherals, reading data from RAM, setting breakpoints and perform step by step debugging.

The algorithm of the whole program is presented in Figure 4. It consists of two basic parts: the initialization part performed only once after microcontroller startup and the main

loop in which appropriate actions are continuously repeated.

The initial part of program begins immediately after reset of microcontroller. Its task is to prepare all the peripherals and software structures to use during operation. The first step is configuration of I/O pins as inputs or outputs. Then, the external interrupts are configured. All of these actions take place during configuration of microcontroller, still before starting of 'main()' function. The program execution begins from initialization of peripheral devices e.g.: timers, I2C peripheral bus, UART interface. Next, structures needed to manage the sensors are configured. Afterwards, the microcontroller reads the Hardware Address, validates it and initializes the I2C interfaces to communicate with the Shelf Manager and AMC Modules. Next, the SDR structures and FRU Information are initialized, because they contain information about current address of the Carrier Board. This address must be overwritten in these structures and the checksums must be updated because of that.

Other important elements, which also need to be initialized, are FIFO queues. They are implemented in the form of circular buffers and are used to store incoming and outgoing messages. The entire program uses six such queues. Two of them store frames from and to the Shelf Manager (see Figure 5) and other from and to AMC Modules (see Figure 6).

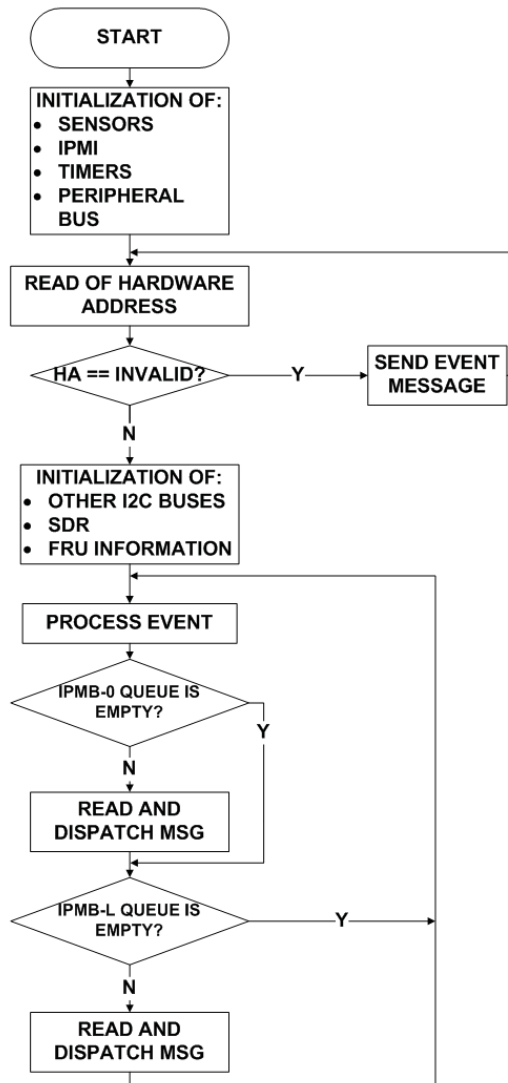


Fig. 4. Algorithm of the IPMC program

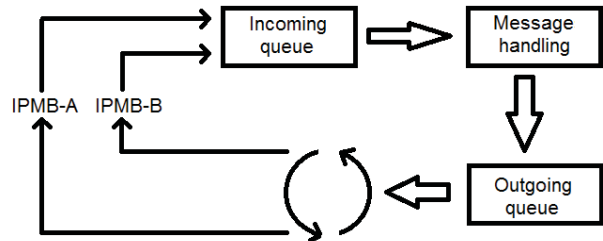


Fig. 5. Queues for messaging over IPMB-0

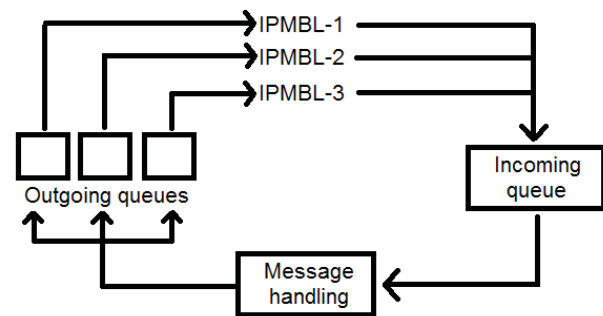


Fig. 6. Queues for messaging over IPMB-L

The second part of the program is the Main Loop. It consists of three basic parts: processing of events which occurred on the board, handling messages from Shelf Manager and handling messages from AMC Modules. They are executed in an infinite loop.

The flow of the IPMC program is determined by events according to rules of event-driven programming. The events

can be generated in various points of program. For example during the interrupt handling procedure of the timer, the state of the Hot Swap switch is controlled and on this basis the flags which inform about opening/closing of the Hot Swap are set. In the same function, states of PS1 signals for all AMC Modules are tested and events which inform about insertion/extraction AMC Module are generated. However, events indicating messages waiting in the outgoing queues are generated in the function which adds the message to the queue. The events are represented by bits in a 32-bit variable (each bit is a flag of one event) and on this basis they are identified and specific actions are taken. All events supported in the current software version are listed below:

- *HOTSWAP_OPENED* - informs about opening of the Hot Swap Handle;
- *HOTSWAP_CLOSED* - informs about closing of the Hot Swap Handle;
- *TIMER_10_MS* - generated every 10 ms and is used to control the LEDs;
- *TIMER_100_MS* - generated every 100 ms and is used to read values from sensors;
- *PS_ACTIVE_AMC1*,
PS_ACTIVE_AMC2,
PS_ACTIVE_AMC3 - inform about the insertion of AMC Modules (active states of PS1 signals);
- *PS_INACTIVE_AMC1*,
PS_INACTIVE_AMC2,
PS_INACTIVE_AMC3 - inform about the extraction of AMC Modules (inactive states of PS1 signals);
- *BUS_ERROR* - generated when an error on IPMB-0 occurs, e.g. one of the buses is busy all the time and communication is not possible;
- *IPMB0_QUEUED_MSG*,
IPMBL1_QUEUED_MSG,
IPMBL2_QUEUED_MSG,
IPMBL3_QUEUED_MSG - generated if any message waits to be sent in the outgoing queues;
- *PING* - generated periodically; in the procedure responsible for handling of this event a "Get Device ID" command is sent which is mainly used to check the status of communication with devices, in this case with AMC Modules.

In the first stage of the Main Loop the subroutine for events handling is called. It checks which flags are set and takes appropriate actions. For instance if any of flags indicating queued messages is set, in function processing events the transmission process is started etc. The events were introduced in order to shorten the interrupt handling routines. Thanks to them, all the actions related to the events are executed in the Main Loop and the processor can handle interrupts which are generated very often in such large numbers of communication interfaces, much faster.

After events processing begins the handling of messages. As already mentioned, for this purpose the queues for incoming and outgoing messages were introduced. Handling of the messages to the Shelf Manager can be described as follows: each outgoing frame is added to the outgoing queue, the

same for both buses of IPMB-0. Messages from it are sent alternately over these buses while responses are written to the second, incoming queue, whence they are collected and analyzed. It is worth adding that according to the ATCA specification messages from the outgoing queue are not deleted until a response is received. However, IPMC responses to a requests from Shelf Manager are sent directly over IPMB-0, without adding to the outgoing queue.

The process looks similarly for AMC Modules, however due to the fact that each AMC is connected to the separate I2C channel of microcontroller, for outgoing messages three separate queues were implemented, one for each AMC. Thanks to that they can independently communicate with IPMC. Used solution ensures that request messages for each one of modules are collected in defined queue and even its blocking has no influence to requests addressed to remaining modules. Furthermore, it makes easier removing messages waiting for transmission to the given module e.g. when this module was extracted.

The task of functions in the Main Loop responsible for messages handling is to read messages from the queue, if any are waiting, check their validity, prepare a response and take appropriate actions.

IV. CONCLUSIONS

This new solution with the Renesas microcontroller greatly improves the communication process with the Shelf Manager and AMC Modules compared to the one used previously. Thanks to apply Renesas microcontroller in the role of IPMC it was possible to eliminate the external parallel bus to I2C bus converters used in previous solutions. The connection of all Intelligent Platform Management Buses on the board directly to the microcontroller significantly improves the reliability of communication, simplifies management and eases the development of software for IPMC. Using an independent I2C channel for each module protects against collisions and, in case of failure of one module and blocking of the bus, the other two may still communicate with the IPMC. Thanks to a large number of I/O pins it is possible to connect all the necessary signals from AMC Modules directly to the microcontroller without using an FPGA device. It allows to focus the whole functionality of the IPMC in one device which reduces the number of potential sources of errors. Higher frequency of the CPU enables efficient service of I2C interfaces even during simultaneous communication with all the modules. The more stable communication and smaller number of devices that the logical IPMC comprises significantly improve the reliability of the Carrier Board, increase the speed of work and simplify configuration and maintenance.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Commission under the EuCARD FP7 Research Infrastructures grant agreement no. 227579 and Polish National Science Council Grant 642/N-TESLA-XFEL/09/2010/0. The author is a scholarship holder of project entitled "Innovative education ..." supported by European Social Fund.

REFERENCES

- [1] Eder J., Just what is ... ATCA?, Electronics Systems and Software Volume 2, Issue 5, Oct.-Nov. 2004
- [2] Karlsson A.D.O., Martin B., ATCA: its performance and application for real time systems, IEEE Transactions on Nuclear Science, vol. 53, pp. 688–693, June 2006
- [3] Larsen R.S., ATCA for Machines, Stanford Linear Accelerator Center – Electronics & ILC Divisions, Menlo Park California, USA, 2007
- [4] PICMG® 3.0 Revision 2.0 AdvancedTCA® Base Specification, 28 October 2005
- [5] Larsen R.S., Advances in Developing Next-Generation Electronics Standards for Physics, 16th IEEE-NPSS Real Time Conference, 2009
- [6] Liu M., et al., ATCA-based computation platform for data acquisition and triggering in particle physics experiments, The International Conference on Field Programmable Logic and Applications, 8-10 September 2008
- [7] Xu H., et al., Application of ATCA in Trigger and DAQ System for Experimental Physics, 16th IEEE-NPSS Real Time Conference, 2009
- [8] Gonçalves B., et al., ATCA Advanced Control and Data acquisition systems for fusion experiments, 16th IEEE-NPSS Real Time Conference, 2009
- [9] Jezynski T., Simrock S., The Low Level Radio Frequency System Architecture for the European X-FEL, MIXDES 2006
- [10] Koch G., et al., Modelling and Identification of a Free Electron Laser RF System, Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany, October 4-6, 2006
- [11] Simrock S., et al., Evaluation of an ATCA Based LLRF System at FLASH, MIXDES, 2009
- [12] Lang J., et al., Intelligent Platform Management Controller for ATCA Compute Nodes, 16th IEEE-NPSS Real Time Conference 2009
- [13] Zhilou Yu, Hua Ji, Research of IPMI Management based on BMC SOC, International Conference on Management and Service Science (MASS 2010), 24-26 August 2010
- [14] Leangsuksun C., et al., IPMI-based Efficient Notification Framework for Large Scale Cluster Computing, Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops CCGrid06, 16-19 May 2006
- [15] PICMG®/AMC.0 R1.0 ECR-002 Draft 0.8 Advanced Mezzanine Card Base Specification, 28 April 2006
- [16] Intelligent Platform Management Specification v2.0 Document Revision 1.0, 12 February 2004
- [17] Overgaard M., The quickest way to a compliant and interoperable Intelligent Platform Management Controller for AdvancedTCA, CompactPCI Systems, April 2004
- [18] Zawada A., et al., Prototype AdvancedTCA Carrier Board with three AMC bays, Nuclear Science Symposium Conference Record, 2008
- [19] Zawada A., et al., ATCA Carrier Board with IPMI supervisory circuit, MIXDES, 2008
- [20] Wychowaniak J., et al., Diagnostic Application for Development of Custom ATCA Carrier Board for LLRF, MIXDES 2009
- [21] http://www.renesas.eu/products/tools/ide/ide_hew/ide_hew_tools_product_landing.jsp
- [22] <http://www.kpitgntools.com>



Andrzej Napieralski received the M.Sc. and Ph.D. degrees from the Technical University of Lodz (TUL) in 1973 and 1977, respectively, and a D.Sc. degree in electronics from the Warsaw University of Technology (Poland) and in microelectronics from the Université de Paul Sabatié (France) in 1989. Since 1996 he has been a Director of the Department of Microelectronics and Computer Science. In 2002 he has been elected and in 2005 reelected as the Vice-President of TUL. He is an author or co-author of over 900 publications and editor of 18 conference proceedings and 12 scientific Journals. He supervised 44 PhD theses; four of them received the price of the Prime Minister of Poland. In 2008 he received the Degree of Honorary Doctor of Yaroslavl the Wise Novgorod State University (Russia).



Piotr Perek received the Master of Science degree in the field of Electronics and Telecommunications from Technical University of Łódź in 2010. He continues his education as a PhD student at the Department of Microelectronics and Computer Science TUL. His interests include digital electronics and software development both for computers as well as for embedded systems. His recent research has involved the development of controllers for IPMI management layer for control systems based on ATCA and AMC standards.



Dariusz Makowski received PhD degree in electrical engineering at the Department of Microelectronics and Computer Science Technical University of Łódź in 2006. His main areas of interests are digital electronics, embedded systems and programmable devices. He is engaged in the development of xTCA standards for High Energy Physics. He is involved in the design of distributed data acquisition and control systems based on ATCA, MTCA and AMC standards.



Paweł Prędko graduated from the International Faculty of Engineering at the Technical University of Łódź in 2009 with an MSc degree in Telecommunications and Computer Science. He continues his education as a PhD student at TUL and is involved in many projects. His main fields of interest are embedded systems and digital signal processing. He also deals with control systems and data acquisition systems based on the ATCA and uTCA architectures.