Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 4 Issue 6, June - 2017

Implementation of a Sodimm Interfaced Embedded and Modular Controller Board

Sadık Arslan
Research and Development Department
Kent Kart Ege Elektronik A.Ş.
İzmir, Turkey
sadik.arslan@kentkart.com.tr

Mustafa Gündüzalp
The Department of Electrical & Electronic Eng.
Yaşar University
İzmir, Turkey
mustafa.gunduzalp@yasar.edu.tr

***Abstract*—Modular hardware structure is a very important requirement in electronics and embedded system design processes that have microprocessors. According to market demands, modularity provides the ability to support different features of an embedded device quickly. Research and development time is shortened; development costs are reduced; productivity is increased, and flexibility is raised through use of modular units. Electronic devices are composed of blocks with different functions such as microprocessors, input-output interfaces, power units etc. Each block fulfills its own special operations. In new embedded designs, if different functions are required, these blocks must be replaced. In non-modular systems, these changes take a long time and increase the product cost. In embedded electronic devices that have microprocessors, highest material and design cost are incurred with the control unit which contains microprocessors. In this study, the modular design of the control unit application was performed by using Sodimm interface. During the implementation of this project, a microprocessor that performs video and audio processing, an NAND flash, a DDR2 RAM and a power management IC are used. Thus, a control board which can support many features like playing audio and running LCD, SD cards and USBs, which runs on Linux operating system, the modularity of which is provided by Sodimm interface, and on which software can be developed was designed. The designed module will be primarily used in handheld terminals and validators found in fare collection systems. The designed card is also suitable for general use in embedded systems.**

***Keywords*—Controller card design; design verification; Linux porting; modular hardware design; Sodimm interface**

I. INTRODUCTION

Today, the modular structures in embedded systems are gaining importance with emerging technologies. Modular structures are frequently used to reduce the development time and cost in the research and development (R&D) phase. In this study, a Small Outline Dual In-line Memory Module (Sodimm) interfaced microprocessor card was designed for general purpose use. Primary use will be in validators and handheld terminals of automatic fare collection systems.

Embedded systems are systems that are designed for specific purposes. Unlike personal computers, embedded systems perform tasks specifically defined in advance. Embedded systems fulfil a specific function by combining hardware and software systems that are very dependent on each other. In embedded systems, relatively slow processors, memories and other units are used. It is required to assign an Operating System (OS) for each system that have a processor. Many OSs such as Windows CE, Embedded Linux, Java OS, LynxOS, Mobilinux are used in embedded systems. Automatic tax machines, mobile phones, network equipment, motor controllers, brake systems, home automation products, air defence systems, medical equipment, measurement systems are systems that can be given as examples of embedded systems.

In general, there are sections such as sensors, processor, power block, memory, communication interfaces in the structure of an embedded system. The processor, storage unit, Random Access Memory (RAM) and power block that supply all parts are connected to each other in embedded systems. When the processor is changed, other units may also change. For this reason, it is important to make a modular processor block for embedded systems. Basic block diagram of an embedded system that has a processor module can be seen in Figure 1.

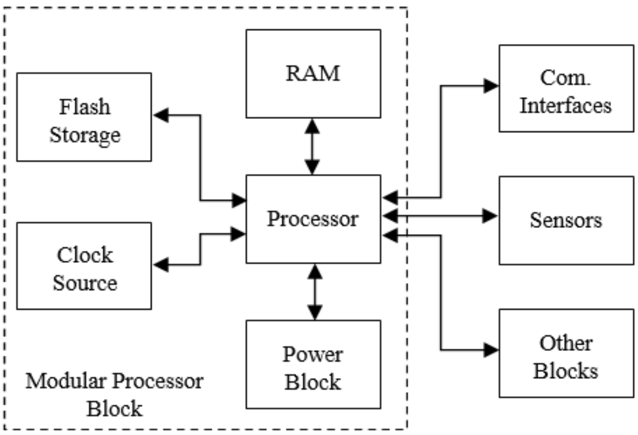

Fig. 1. *Basic block diagram of an embedded system.*

Modular processor module usage helps people who develop their new own embedded system devices. They just add some new peripherals and don't need to design a processor block, which is difficult to design. Thus, for new designs, processor part will not be designed and development time and cost can be reduced.

In our study, main subject is the design of a modular processor card considering different parameters. The module is obtained for general usage in embedded electronic systems. First applications that are designed by these modules will be validators and Personal Digital Assistant (PDA) devices which are used in fare collection systems.

As main processor of the designed board, an i.MX25 series microprocessor is preferred [1]. This integrated circuit (IC) is selected by considering cost/performance balance. In addition, this IC works with 128 MB Double Data Rate Random Access Memory (DDR2 RAM) and 512 MB Negative AND (NAND) flash module. These are main features of the designed module. The designed module is in the segment that can be considered as low cost for systems that can play video and audio.

In the designed module, 200-pin Sodimm output interface is preferred for interfaces such as UART, General Purpose Input-Output (GPIO) etc. This interface is selected according to many criteria such as cost, physical dimensions, number of pins, and usage conditions in the market. In this study, mechanical dimensions of the standard Sodimm interface are used [2].

Sodimm interface is generally used in DDR RAM types. Sodimm interfaced RAMs are widely used in many computer systems and embedded systems such as devices used in nuclear science [3 - 5], embedded systems that include Field Programmable Gate Array (FPGA) [6 - 8], devices used in astronomy [9, 10], systems in the medical field [11, 12], seismological devices [13], embedded system devices used in scientific purposes [14], in computer science [15], and signal processing fields [16].

V. Harini and S. Rajender discussed the design of a DDR3 RAM in detail [17]. P. Li et al. worked on the electrical characteristics and the mechanical requirements that affect performance in Sodimm socket designs [18]. C. J. Huang, C. Y. Chou, and K. N. Chiang published an article about the mechanical strength of Sodimm [19]. F. Abtahi et al. used a Sodimm module similar to the designed module in our study in their bio-signal measurement project [20]. Z. C. Lin and C. Y. Ho performed quality improvement processes in the production of Sodimm cards in their study [21]. Y. Cinar et al. worked on strength problems relating to soldering operations in Sodimm cards [22]. H.S. Ahn et al. used a similar module that was developed in our project in nuclear sciences. They used similar interfaces in their project that was called The Cosmic Ray Energetics and Mass Timing Charge Detector [23].

When studies that related to the Sodimm interface are examined, it is seen that Sodimm interfaced modular cards are used in new embedded designs. However, there is no research topic related to electrical aspects of designed Sodimm interfaced cards, implementation of microprocessors, Linux porting and design verification. In our study, a microprocessor-implemented Sodimm interface module design is realized by taking into consideration many different disciplines.

II. Controller Card Design

First, block system diagram is determined while designing the Sodimm interfaced card. Different interfaces that are not supported by the microprocessor were identified during designing. These interfaces can be supported by high performance series ICs [24]. According to this situation, 200 pins were distributed some of which were optional in the Sodimm interface. The layout of the pins is made according to ease of routing while drawing printed circuit board (PCB). Power Management Integrated Circuit (PMIC), Ethernet, two Universal Serial Buses (USB), two Inter-Integrated Circuits (I2C), two Serial Peripheral Interfaces (SPI), Pulse-width modulation (PWM) output, two Secure Digitals (SD), five Universal Asynchronous Receiver/Transmitters (UART), two Integrated Interchip Sound (I2S), One Wire Data (OWD), 24-bit Liquid Crystal Display (LCD), Subscriber Identification Module (SIM) interface, three Analog Inputs and twelve GPIO pins were connected to Sodimm interface in the module design. For applications that require high performance interface like High Definition Multimedia Interface (HDMI), Camera Serial Interface (CSI), Low-Voltage Differential Signaling (LVDS), Peripheral Component Interconnect Express (PCIE), Serial AT Attachment (SATA) and Sony/Philips Digital Interface Format (SPDIF), output signals may be added to the design in future works which may also contain a more powerful processor. In our current design these high performance interfaces were not added to the module because the microprocessor that is used in the module does not support them. The Ethernet PHY IC, 128 MB DDR2 RAM, 512 MB NAND Flash [25] and MC34704B IC [26] as PMIC are used in the designed module. Simplified drawing of the designed Sodimm module can be seen in Figure 2.

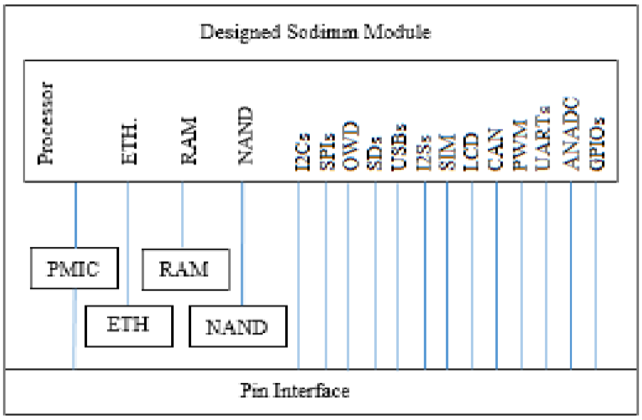

Fig. 2. *Basic block diagram of an embedded system.*

The developed Sodimm interfaced card will be primarily used in the validator and PDAs that function in fare collection systems. In addition, the module can be used in different embedded devices such as vehicle tracking devices and ticket vending machines. Many different technologies and interfaces like UART, USB, SPI and SD are used in a validator device. Basically, the validator system reads from Radio Frequency (RF) cards and writes to RF cards by using these interfaces. The remaining balance and charge fee information are written to the LCD that is used in the validator. The validator can reach the Global System for Mobile Communications (GSM) module using USB interface and send card usage information to the system server. Additionally, there are audio codecs or audio amplifiers which run with I2S interface for auditory announcement processes. The designed Sodimm module card provides all the necessary interfaces and processing power to the validator system.

PCB drawing was made after the schematic design. During the PCB design, the most important part is drawing data and address lines that are located between the DDR2 RAM and the microprocessor. For noise insulation on differential and single lines, design was made according to the appropriate values of line impedance. PCB design is done using 8 separate layers. Noise insulation at analog signals was provided with ground planes [27]. The shape of the PCB was made according to standard Sodimm mechanical dimensions. In addition, screw holes which help fix the module card in vibration environments were added to the card. The designed PCB can be seen in Figure 3. Signals and components that are located on the top layer can be seen in the Figure 3.

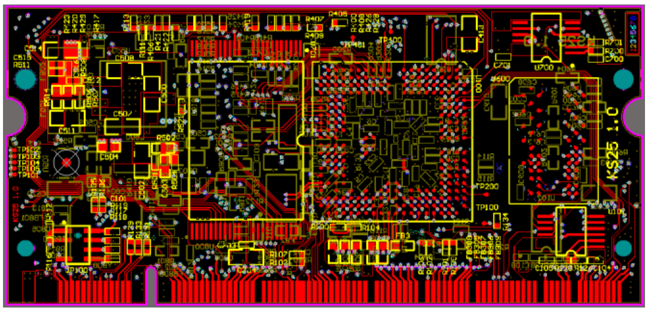

Fig. 3. *Designed module, top signal and component view.*

Power requirements of the microprocessor, NAND, DDR2 RAM and Ethernet units were determined. Required voltages were 3.3 V, 1.45 V, 1.8 V, 1.5 V for core and 1.2 V battery supply for Real Time Clock (RTC) in the module. Energy of the battery was taken from outside of the module via the Sodimm connector. There is no battery on the module. The main supply of the module can accept power source with 2.7-5.5 V interval voltage range. 3 different regulators that are located in the PMIC were used in the power block design. 3.3 V, 1.45 V and 1.8 V voltage sources were taken from these PMIC regulators. In addition, a separate regulator was used outside of the PMIC for 1.5 V source. All required supply voltages for the microprocessor came from the PMIC and 1.5 V regulator. The power diagram of the designed Sodimm module can be seen in Figure 4. VIN shows the input supply voltage that has 2.7-5.5 V interval range of the module. The Reg1/VG regulator, which is located in the PMIC, is a 5 V step up circuit for supplying inner circuits of the PMIC.

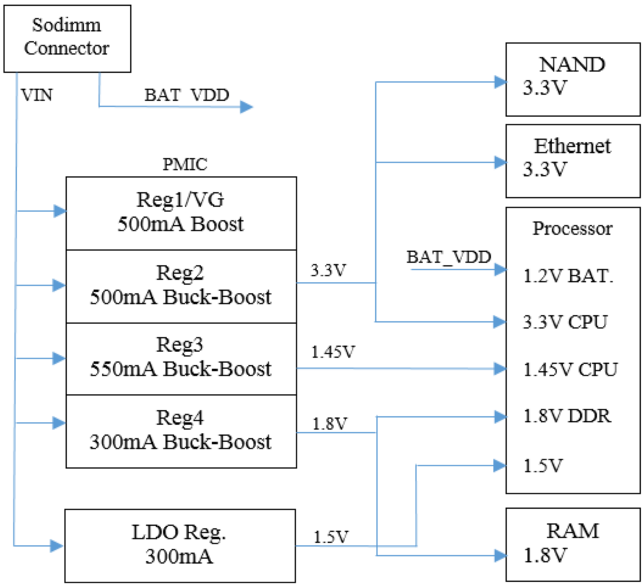

Fig. 4. *Power diagram of the module.*

In order to provide proper power-up timing after the first time the power is applied to the board some measurements and improvements are made. All supplies need to reach their own specific voltage levels before resetting the microprocessor. The timing diagrams of the reset signal (RST) coming out of the PMIC and supply voltages of the system can be seen in Figure 5. The reset signal comes from the PMIC and goes to the microprocessor 10 ms after reaching 1.8 V supply level and operation of the system starts [28].

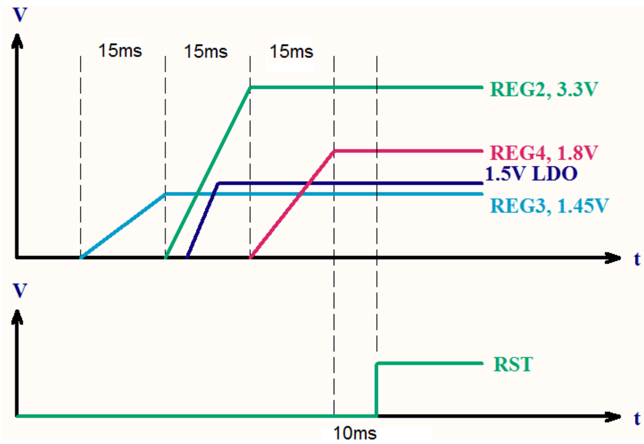

Fig. 5. *System power-up waveform.*

The reset operation of the microprocessor and all of the designed system is very important to overcome problems encountered during module operation. The On/Off and reset operations are considered while developing the module. The Reset-In, Reset-Out, Power-On Reset (POR) and PMIC Power Button (Pow_But) signals were connected to the Sodimm pin interface. The Reset-In signal coming from outside of the module directly resets the microprocessor. The reset signal from the Joint Test Action Group (JTAG) [29] interface that is located on the Sodimm card also resets the microprocessor. The Reset-Out signal provides reset signal to units that are located outside of the Sodimm module. The Power Button signal was connected to the Power On/Off input of the PMIC. This pin was placed to ensure opening of the system using a mechanical switch. A GPIO coming from the microprocessor was optionally connected to the PMIC On/off signal. The POR signal that came from outside of the module was connected to the POR input of the microprocessor. Reset ICs can be connected to this pin from outside of the Sodimm module. The reset signal coming from the PMIC was optionally connected to the POR input of the microprocessor. Simplified reset signal architecture of the Sodimm module is given in Figure 6.

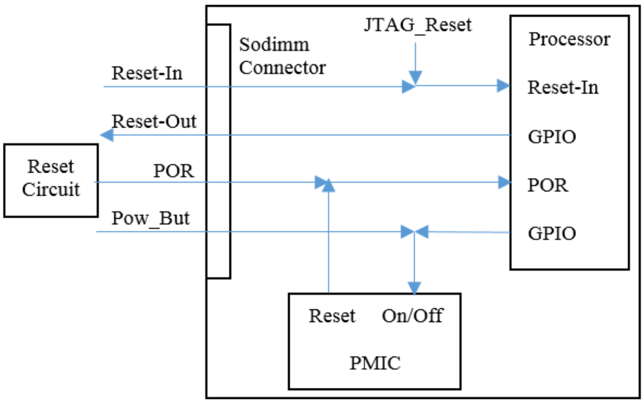

Fig. 6. *Simplified reset signal architecture.*

III. Software Development and Design Verification

First of all, hardware of the designed Sodimm module was rendered available for testing the module. The designed 8-layered module PCB was produced. All necessary main units such as the microprocessor, PMIC, NAND and RAM and peripheral components needed were soldered in the PCB and a suitable Sodimm card was obtained for test purposes. The Sodimm module that has all components can be seen in Figure 7.

Fig. 7. *The designed Sodimm module.*

Another test card design was made for testing all features which were supported by the Sodimm module. In this card, a hardware environment where all features such as USB, Ethernet, LCD, serial ports were tested was provided. The provided test set-up that included the test card and the Sodimm module card for development applications can be seen in Figure 8. The Sodimm module was inserted to the test card via the Sodimm interface connector.

Fig. 8. *Designed Sodimm module development set-up.*

Running the Linux OS on the microprocessor is necessary for testing all parts of the designed module. All features that are supported by the module were tested via applications, drivers and libraries that developed on the Linux OS and Linux porting procedure was made. The Software Development Kit (SDK) was created as a result of these studies. The SDK was required for both the design verification process and developers who used the designed Sodimm module.

Cross compiler environment was built on a Linux Ubuntu desktop computer [30]. This environment allowed the Linux Kernel 2.6.35.3 operating system to be compiled according to the suitable ARM cored microprocessor that is used in our study. The open source Eclipse editor was run on the computer for compiling the Bootloader [31] and Kernel. Basically, the bootloader allowed opening of the computer and starting of the OS. Bootloader was compiled in cross compiler environment and was rendered operable. The Kernel (uImage) that had main parts such as the file management, network management, basic drivers was also compiled in the cross compile environment. Parameter adjustments of the DDR2 SDRAM that would be used in Bootloader code were made [32]. The module card must be booted by a boot interface such as the NAND, SD, UART, and USB for first board bring-up process of the OS [33]. In our study, opening from the SD card and NAND interfaces were provided at first step. The memory units that can be reached by these interfaces had to have some binary parts which can be seen in Figure 9 to allow the operation of the embedded system. A Root File System (Rootfs) where the necessary files are located was needed in a system that has Linux OS. Linux Target Image Builder (LTIB) system was used for preparing Rootfs according to needs and the hardware of the module. The first compiled Bootloader, uImage and Rootfs were copied to the SD card and NAND flash. The size of the binary code that was copied was considered at the copy process. Binary files were placed on the

NAND with specified addresses as in Figure 9 [34]. A User File System (Userfs) that holds user files and Linux OS Configuration (Configure) binary sections were available at the NAND memory space. Linux kernel command line was reached after executing the designed Sodimm module from the NAND flash or SD card. Development operations continued via the command line [35]. The NAND boot process is more appropriate for field operation of the system. This process was studied in detail. Suitable modifications of the NAND hardware were made into the Bootloader code.

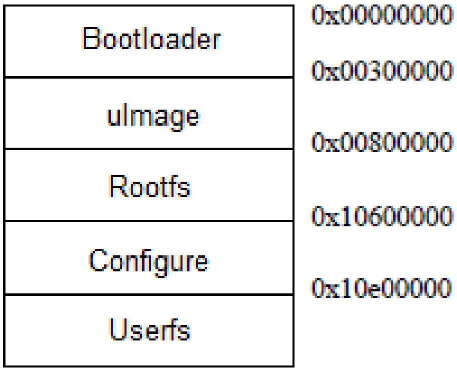

Fig. 9. *NAND Flash binary allocation for system boot [34].*

Necessary units in the Kernel and Bootloader were patched for the proper operations of the designed module. Software codes of the Kernel and Bootloader were modified in line with GPIOs and other interfaces. Necessary changes were made in drivers that are located in the Kernel code to operate all features such as UART, ADC and SPI interfaces. Required user libraries and applications were written in the Eclipse compiler using the C language [36, 37] for operation of units such as the SPI, LCD [38], I2C, Analog Input [39] and RTC [40]. All peripheral units were functionally tested. Working configuration adjustments were made to driven 18-bits parallel data during the LCD study. However, the system is able to support 24-bits LCD operation in the future with the optional structure of the designed module. General software architecture drawing of the system that was developed in the study can be seen in Figure 10.

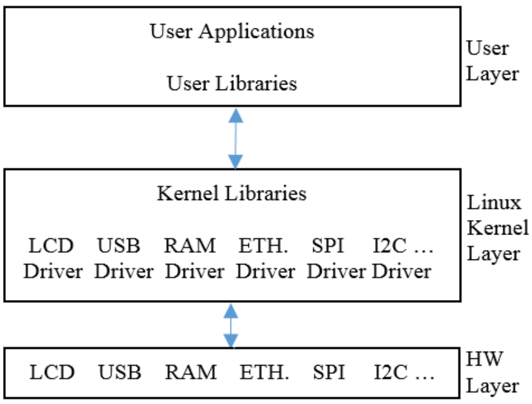

Fig. 10. *The system SW layers.*

In addition, the system is compatible with the Secure Boot process that can be made with High Assurance Boot version 3 (HABv3) [41]. The Secure Boot adaptation can be made easily by working on the Bootloader and Kernel. In all development processes, debugging operations were used to solve problems faced in the study. The JTAG and Eclipse tools that are commonly preferred in the world were used at the development process [42].

The processor and memory performance tests were also conducted when the system was running as a Validator. Tests that were performed included different operations such as recording on disc, RF card reading, printing an image on the screen, sound operations. While these tests were performed, the processor and memory utilization rates were examined by Top application. The Top application is a performance monitoring tool of the Linux OS. This application illustrates the usage of the processor activity as percentage when the Linux OS is running. The total load of the processor is approximately 3.5% in log-term usage. In short time intervals, 70~80% instant usages can be seen from the Top application. This performance, which was seen from the Top application, suited the expected performance of the designed module.

The temperature performance of the designed Sodimm module was also examined. Measurements were made using a thermal camera while the module was working as a validator. There was no problem relating to the temperature in the office environment. The maximum temperature on the card was measured as 43~45°C. The temperature image that was obtained from the thermal camera is given in Figure 11.

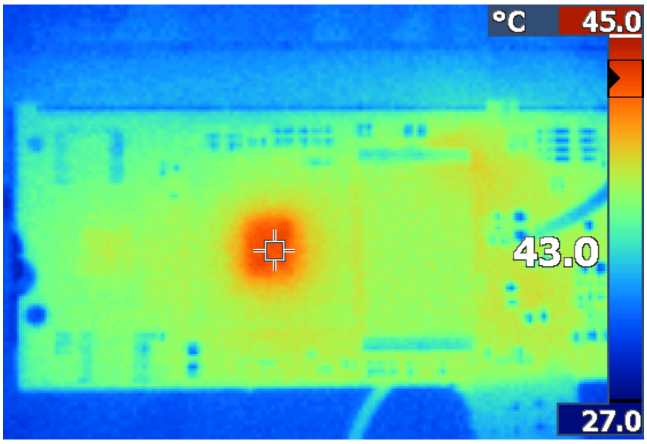

Fig. 11. *Thermal performance.*

Additionally, the power analysis of the module was done. The main power supply source that was called VIN and other sub-branches were measured. Power consumption of the VIN was 0.872 W, 1.8 V branch was 0.145 W, 1.45 V was 0.127 W, 1.5 V was 0.019 W, and 3.3 V was 0.442 W. So, the efficiency of the power unit was calculated as 84.05 %.

IV. Discussion and Conclusions

In the study, a Sodimm interfaced card that met modular hardware needs, solved the problems of modularity that has been designed. In the designed card, hardware and software works were performed and a module card that can run the Linux OS was obtained. The SDK was made for the Linux OS that can run on the designed embedded system board and software development processes. Necessary Kernel drivers, libraries, and user applications were developed. There were many tests during the development of the project and design verification process was completed. With the use of the modular card that was developed in this study, new electronic systems can be designed faster and design costs will be lower in further projects.

In the future, higher performance modules that are similar to this project can also be designed. Using these modules, the image processing and video processing operations with high performance requirements can also be performed in embedded devices. Different technologies and interfaces that are considered optional during the development can be used in a compatible manner with the designed Sodimm module pin structure. The system complies with the HABv3 Secure Boot adaptation. If necessary in the future, Secure Boot topics can be examined. Additionally, Android OS, which is commonly used by mobile devices, can run on the designed module.

Acknowledgment

We would like to thank Kent Kart Ege Elektronik A.S. Company for all equipment and laboratory conditions.

References

[1] Freescale Semiconductor, i.MX25 Applications Processor for Automotive Products Silicon Version 1.2. Austin, Texas, USA: Freescale Semiconductor Inc., 2013.

[2] TE Connectivity, DDR1 & DDR2 Sodimm Profile Standart Type Datasheet, Schaffhausen, Switzerland: Tyco Electronics Ltd., 2011.

[3] P. Rodrigues, M. Correia, A. Batista, P.R. Carvalho, B. Santos, B.B. Carvalho, J. Sousa, B. Gonçalves, C.M.B. Correia, and C.A.F. Varandas, "Intelligent platform management controller software architecture in ATCA modules for fast control systems," IEEE Trans Nucl Sci, vol. 61-4, pp. 2318-2322, 2012.

[4] A.J.N. Batista, C. Leongb, V. Bexigab, A.P. Rodriguesa, A. Comboa, B.B. Carvalhoa, J. Fortunatoa, M. Correiaa, J.P. Teixeirab, and I.C. Teixeirab, "ATCA/AXIe compatible board for fast control and data acquisition in nuclear fusion experiments," Fus Eng Des, vol. 87, pp. 2131-2135, 2012.

[5] G. Balbi, M. Bindi, D. Falchieri, A. Gabrielli, R. Travaglini, S.P. Chen, S.C. Hsu, S. Hauck, and A. Kugel, "Commissioning oftheread-outdriver (ROD) card for the ATLAS IBL detector and upgrade studies for the pixel Layers 1 and 2," Nuclear Instruments and Methods in Physics Research A, vol. 765, pp. 232-234, 2014.

[6] A.B. Mann, I. Konorov, F. Goslich, and S. Paul, "An advancedtca based data concentrator and event building architecture," 17th IEEE-NPSS 2010 Real Time Conference, pp. 1-3, 2010.

[7] R. Cupek, A. Ziebinski, and M. Fraenk, "FPGA based OPC UA embedded industrial data server implementation," J Circuit Syst Comp, vol. 22-8, pp. 1-18, 2013.

[8] P. Barrio, C. Carreras, J.A. López, O. Robles, R. Jevtic, and R. Sierra, "Memory optimization in FPGA-accelerated scientific codes based on unstructured meshes," J Syst Architect vol. 60, pp. 579-591, 2014.

[9] A. Gunst, A. Szomoru, G. Schoonderbeek, E. Kooistra, D. Schuur, and H. Pepping, "The application of UniBoard as a beam former for APERTIF," Exp Astron vol. 37-1, pp. 55-67, 2014.

[10] M. Kleifges, "Measurement of cosmic ray air showers using MHz radio-detection techniques at the Pierre Auger Observatory," Nucl Instrum Meth, vol. 718, pp. 499-501, 2013.

[11] F. Xia, Y. Dou, anf G. Jin, "The unified accelerator architecture for RNA secondary structure prediction on FPGA," J Supercomput, vol. 61, pp. 826-855, 2012.

[12] F. Xia, Y. Dou, and G. Lei, "FPQRNA: hardware-accelerated QRNA package for noncoding RNA gene detecting on FPGA," J Bioinform Comput Biol, vol. 8-4, pp. 743-761, 2010.

[13] P. Riascos, A. Racines, J. Caicedo, J. Mejia, and H. Meyer, "OSSODAS, a portable digital system for seismological signal acquisition," J Earth Sci Res, vol. 8-1, pp. 52-55, 2004.

[14] F. Tapperoa, and Y. Abe, "In situ plasma diagnostics study by means of optical emission spectroscopy for diamond chemical vapor deposition under high gravity conditions," Rev Sci Instrum, vol. 74-10, pp. 4458-4461, 2003.

[15] T. Hussain, A. Haider, S.A. Gursal, and E. Ayguadé, "AMC: advanced multi-accelerator controller," Parallel Comput, vol. 41, pp. 14-30, 2015.

[16] P. Kumar, V. Kamakoti, and S. Das, "System-on-programmable-chip implementation for on-line face recognition," Pattern Recognit Lett, vol. 28, pp. 342-349, 2007.

[17] V. Harini, and S. Rajendar, "Design, validation and correlation of characterized SODIMM modules supporting DDR3 memory interface," Journal of Electronics and Communication Engineering, vol. 6-5, pp. 01-11, 2013.

[18] P. Li, J. Martinez, J. Tang, S. Priore, K. Hubbard, and J. Xue, "Development and evaluation of a high performance fine pitch SODIMM socket package," 54th Electronic Components and Technology Conference, pp. 1161-1166, 2004.

[19] C.J. Huang, C.Y. Chou, and K.N. Chiang, "Dynamic study and structure enhancement of small outline dual-in-line memory module," 10th International Conference on Thermal, Mechanical and

Multi-Physics simulation and Experiments in Microelectronics and Microsystems, pp. 1-5, 2009.

[20] F. Abtahi, J. Snäll, B. Aslamy, S. Abtahi, F. Seoane, K. Lindecrantz, "Biosignal PI, an affordable open-source ECG and respiration measurement system," Sensors vol. 15, pp. 93-109, 2015.

[21] Z. Lin, and C. Ho, "Quality improvement by using grey prediction tool compensation model for uncoated and TiAlCN-coated tungsten carbide tools in depanel process of memory modules," Int J Adv Manuf Technol vol. 40, pp. 857–864, 2009.

[22] Y. Cinar, J. Jang, G. Jang, S. Kim, J. Jang, J. Chang, and Y. Jun, "Failure mechanism of FBGA solder joints in memory module subjected to harmonic excitation," Microelectron Reliab, vol. 52, pp. 735-743, 2012.

[23] H.S. Ahn, P.S. Allison, M.G. Bagliesi, J.J. Beatty, G. Bigongiari, P. Boyled, J.T. Childers, N.B. Conklin, S. Coutu, and M.A. DuVernois, "The Cosmic Ray Energetics and Mass (CREAM) timing charge detector," Nucl Instrum Meth, vol. 602, pp. 525-536, 2009.

[24] i.MX 6Dual/6Quad Automotive and Infotainment Applications Processors. Freescale Semi., 2014.

[25] Freescale Semiconductor. Interfacing and Configuring the i.MX25 Flash Devices. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.

[26] Freescale Semiconductor. 30704 Multiple Channel DC-DC Power Management IC. Austin, Texas, USA: Freescale Semiconductor Inc., 2014.

[27] Freescale Semiconductor. i.MX Layout Recommendations. Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[28] Freescale Semiconductor. i.MX25 Power Management Using the MC34704. Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[29] IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2001 (R2008), 2008.

[30] Freescale Semiconductor. Building a Linux Image and Downloading onto i.MX Processors Using a Virtual Machine. Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[31] Freescale Semiconductor. U-Boot for i.MX25 Based Designs, Source Code Overview and Customization. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.

[32] Freescale Semiconductor. Interfacing mDDR and DDR2 Memories with i.MX25. Freescale Semi., 2010.

[33] Freescale Semiconductor. i.MX25 Boot Options. Freescale Semi., 2012.

[34] Freescale Semiconductor. i.MX25 PDK Linux User's Guide. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.

[35] Freescale Semiconductor. Customizing the Freescale Advanced Toolkit for i.MX Based Platforms. Freescale Semi., 2010.

[36] Freescale Semiconductor. Developing an Application for the i.MX Devices on the Linux Platform. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.

[37] Freescale Semiconductor. Hello World! i.MX PDK Linux. Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[38] Freescale Semiconductor. Image Processing API for the i.MX Platform. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.

[39] Freescale Semiconductor. i.MX25 Integrated Analog-to-Digital Converter. Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[40] Freescale Semiconductor. i.MX25 Real Time Clock (RTC). Austin, Texas, USA: Freescale Semiconductor Inc., 2009.

[41] Freescale Semiconductor. Secure Boot on i.MX25, i.MX35, and i.MX51 using HABv3. Austin, Texas, USA: Freescale Semiconductor Inc., 2012.

[42] Freescale Semiconductor. Using Open Source Debugging Tools for Linux on i.MX Processors. Austin, Texas, USA: Freescale Semiconductor Inc., 2010.