Cross-layer optimization in terminals

Valerio Frascolla[§], Jonathan Ah Sue[§], Muhammad Mudussir Ayub[§], Krzysztof Miesniak[§],
Ralph Hasholzner[§], Jürgen Englisch[§], Amal Ben-Ameur[*]
[§]Intel Deutschland, Neubiberg, Germany
[*]Intel France SA, Sophia Antipolis, France

***Abstract*—The innovation pace of the wireless communication world is breathtaking, not only due to the fierce competition, but also due to the yearly cadence with which standards bodies deliver a new set of functionalities and services to be supported. In this very dynamic context, optimizing products and differentiate against the competitors is key for all those who want to be successful in a make-or-break market. This paper therefore describes some key cross-layer optimization techniques of mobile phones, focusing on cellular protocol stack access stratum enhancements, power optimizations to memory system and finally cross-layer impact on tools for SW development.**

Keywords—Cross-layer optimization; Cellular protocol stack; Memory subsystem; Power optimization; Power modeling; Tooling.

I. Introduction

The cellular communication business is one of the most dynamic among all the technology-related markets. The pace of innovation is breathtaking [1], new set of features, defined from standard bodies, like the 3rd generation partnership project (3GPP), are added each year [2]-[5], the competition is fierce and the market is changing dramatically each few months due to mergers and acquisitions. In this rather hostile environment, it is key that system engineers are given all the needed means to design products that are both competitive and cost-effective, while at the same time capable to add at each new 3GPP Release a new set of features. To cope with such an environment, proper innovation and optimization management is essential for keeping and enlarging companies' market share.

In this context, cross-layer optimization refers to techniques that target more than one layer into which a mobile terminal SW and HW architecture can be split. In a broader sense, cross-layer optimization can be interpreted as methods that affect more than one domain or design block of a mobile terminal platform.

In the last years the concept of cross-layer optimization in wireless systems has gained attention in literature from the network side [6]-[8], and only to a lesser extent from the terminal side [9]-[11]. This paper surveys some recent enhancements of cross-layer optimizations of a cellular modem, touching on the HW/SW co-design perspective on the lower layers of the cellular protocol stack, on the design perspective at the memory system in a terminal chipset, and finally on the impact on professional SW development tools.

The rest of the paper is described in the following. Section II focuses on protocol stack access stratum optimizations, Section III on memory system optimization techniques, Section IV provides an overview of a challenge in the professional SW development, test and verification environment, and finally Section V concludes the paper and hints at future works.

II. Protocol stack Access Stratum optimizations

In this section two exemplary cross-layer optimization techniques are described. The Long Term Evolution (LTE) is chosen as the Radio Access Technology (RAT) of choice, but our proposals are general and scalable enough to be successfully applied to other RATs as well.

A. Data plane Smart engine

The data plane is the RAT-specific part of the access stratum of a cellular protocol stack, in charge of receiving and sending all information created, in the up-link (UL) case, or to be received, in the down-link (DL) case, by the SW applications running in the Application Processor (AP) of mobile phones. An abstract representation of the main logical blocks, or layers according to the 3GPP standards, of the cellular access stratum is sketched in Fig.1. In blue the SW data plane blocks, i.e. the Packet Data Convergence Protocol, (PDCP), the Radio Link Control (RLC), and the Medium Access Control (MAC). Additional SW functional blocks are the Radio Resource Control (RRC), and the SW part of the layer 1 (L1). In green the HW-related blocks dedicated to accelerate some specific protocol stack functionalities, and the physical layer (PHY).

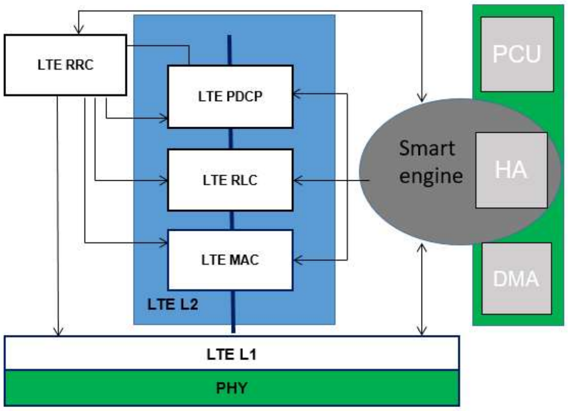

Fig. 1. *Abstract representation of the access stratum of a mobile terminal modem*

A common cross-layer optimization technique in a cellular data plane makes use of some sort of *Smart engine*, composed of both SW and HW parts, which speeds-up some computational heavy operations. This engine can be considered

ISBN 978-90-827970-1-5 © EURASIP 2018

as an abstract interface between PHY, link layer, Power Control Unit (PCU), Direct Memory Access (DMA) unit, and may also contain Hardware Accelerator (HA). Inputs to the engine can be instantaneous DL data from the PHY layer, UL data and high-level scenarios information from higher layers.

The HA block can work as a co-processor to offload the CPU from data plane specific as well as computational intensive tasks. The HA block is often needed as the data plane not only implements the layer 2 basic data handling, i.e. the data link layer of the OSI layering model, but also has to handle several, sometime implementation-specific, important functionalities like confidentiality, ciphering, security, redundancy, and access to channels, so to ensure a stable connection in unfavorable environmental conditions, like low signal and high noise level, and high interference. Each functionality increases the burden and the length of a data packet that the PHY will then have to send, in the UL case, to the network, as shown in Fig. 2.

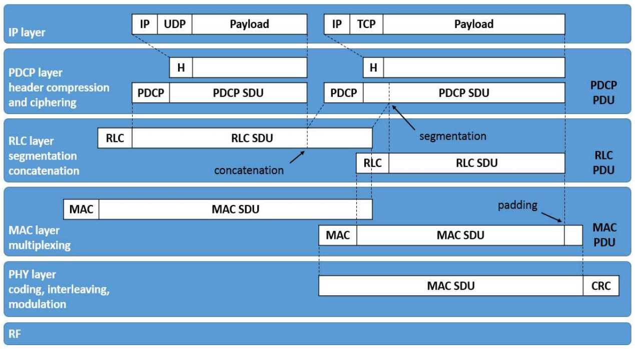

Fig. 2. *Packet processing in the data plane of a terminal phone in UL case*

Cross-layer optimization with the help of the *Smart engine* can be obtained in different ways, e.g. for bundling, activity alignment, and efficient shared resource allocation.

- Bundling is the operation of accumulating the data from the PHY before sending it to the link layer, and in this context optimization is achieved by obtaining a reduced rate of interrupts that would occur otherwise between two layers.
- Activity alignment means bringing multiple CPU cores in idle or active state at the same time, by scheduling the activity accordingly, which in return reduces CPU active time load and improves power key performance indicators (KPI).
- Once the high-level scenarios information is available, the *Smart engine* can configure HA resources accordingly, e.g., to deliver the required performance for a specific scenario at the lowest possible power consumption.

In any case, the overhead for controlling and interfacing to the *Smart engine*, as well as the additional die area of the dedicated HA, are the optimization costs to be paid in order to implement cross-layer optimizations.

B. *Context aware optimization*

From the cellular modem perspective, signaling messages of a specific layer give insights on the different reactions that will happen at that same layer in the next time windows. However, the layer connectivity and architecture of the different layers also introduce a strong dependency among them and their reactions. Therefore, signaling messages of a specific layer can be used as context information to find correlations with lower layer behavior. In the following, we present an example of context-aware optimization of power consumption of PHY layer processing engines, by taking advantage of context information extracted from higher-layer signaling messages.

As depicted in Fig. 2, when processing the communication payload in a UL communication scenario, i.e. going from upper layers to lower layers, the role of layer N is to perform operations (e.g., concatenation, segmentation, adding redundancy) on the data received from its higher neighbouring layer, i.e. layer N+1, adding its own header to indicate which operations have been done and how the data should be retrieved and verified (e.g., forward error correction mechanism and its parameters) by the equivalent layer at the receiver side in the wireless link. From the layer N perspective, the data unit coming from the upper layer N+1 is called a service data unit (SDU) and the transformed data unit forwarded to the lower layer N-1 is called a protocol data unit (PDU). Moreover, re-transmission mechanisms are defined at all layers to improve the robustness of communication in case of unreliable radio channels. Hybrid automatic repeat request (HARQ) are triggered at the receiver MAC layer when the data is not correctly decoded. In such cases, the transmitter receives a non-acknowledgment (NACK) message as response from the receiver. If the data is correctly decoded, the receiver responds with an acknowledgment (ACK) to the transmitter. Re-transmission mechanisms are also defined at RLC layer in acknowledged mode (AM) and at transmission control protocol (TCP) layer. The cost of these needed signaling messages is as a matter of fact an increase of DL traffic when transmitting UL payload (the same considerations would apply also to the DL case).

Indeed, in a TCP/IP data UL scenario, i.e., when TCP/IP packets are sent by a client on the terminal at the near endpoint, most of the DL events are triggered by the re-transmission mechanisms mentioned above. In this scenario, DL IP packets received by the terminal only carry TCP ACKs from, e.g., a server at the far end point. Therefore, there is a causal relationship from UL TCP/IP packets to DL TCP ACKs which relies on a correct data transmission at lower layers. Indeed, if an UL re-transmission is triggered at the MAC layer, the DL TCP ACK information, contained in the payload data of lower layers, is delayed. By exploiting the knowledge of UL re-transmissions, it is possible to evaluate no-DL data duration and occurrences. As depicted in Fig. 3, UL physical HARQ re-transmissions prevent any upper layer DL events until the UL transport block has been correctly received by the base station. Once all transport blocks needed for UL IP packet reconstruction have been successfully decoded, the IP packet is routed through the packet switched network to the far endpoint where TCP is terminated. A DL TCP ACK is then sent back to the terminal at the near endpoint where lower layer DL events are observed, i.e., LTE DL grants at MAC layer for DL RLC PDU payloads.

Fig. 3 shows an example of UL HARQ re-transmissions and “no-DL holes” spanning several milliseconds indicated by black arrows. It depicts UL and DL events of various LTE protocol stack layers at the terminal side as multi-dimensional time series. Colored bars represent events and intense colors correspond to a high number of bytes or packets. Time series are depicted and numbered in the causal order (bottom-up) at a millisecond granularity: (1) Number of bytes in TCP/IP UL packets. (2) Number of UL RLC PDUs. (3) PHY UL grant received from NW. (4) PHY UL ACK received from the NW. (5) PHY UL NACK received from the NW. (6) PHY DL grant received from NW. (7) Number of DL RLC PDUs. (8) Number of bytes in TCP/IP DL packets. From (1) to (5): the UL IP data is transformed following the procedures described in Fig.2 in a top-down manner. From (6) to (8): once the DL PHY response has been received, the data is reconstructed following the bottom-up procedure in Fig.2.

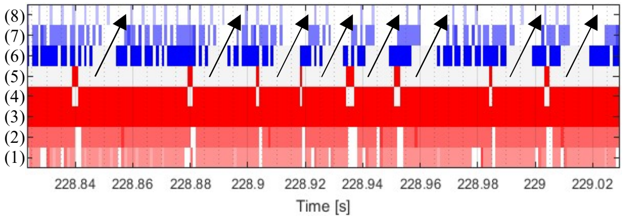

Fig. 3. *Time series of cross-layer metrics*

A terminal aware of such UL only TCP/IP traffic could adapt its DL processing chains taking into account a-priori knowledge of such no-DL holes. In LTE networks, the entire communication between base station and terminal in UL and DL is scheduled by the base station in time transmission intervals (TTI) of 1ms. After a connection is established, LTE terminals have to continuously monitor the physical downlink control channel (PDCCH) containing scheduling decisions of the base station. A non-negligible amount of power is spent in PDCCH monitoring, although it does not carry any useful information from the terminal perspective during the no-DL holes. The following optimizations might be envisioned in order to reduce modem power consumption and increase terminal battery lifetime [12]:

- From RF and PHY layer processing perspective, PDCCH decoding could be suspended during no-DL holes and the receive chains could be turned off.
- From the upper layer data plane perspective, the DL processing elements can be switched to low power states during the idle periods introduced by the no-DL holes.

As these two optimization examples might be applicable for specific use cases, i.e., UL-only data transfer such as file, image or video uploads, the upper layer context has to be known to make a reliable prediction on DL holes duration and occurrences. Indeed, as soon as a DL TCP synchronization request is received, the DL packets received at the IP layer might not only contain TCP ACKs and therefore the holes might not be deterministic anymore. Moreover, as this behavior highly depends on the TCP round trip time (RTT), this parameter could be leveraged to evaluate the expected latency between UL NACKs and DL holes.

Finally, taking advantage of cross-layer scenario specific traffic patterns might not only be restricted to LTE-Advanced UE. In fact, due to the similarities between LTE and 5G NR standards in scheduling and controlling data transmissions, similar cross-layer optimizations can be applied to future 5G NR systems as well.

III. SoC modeling aspects for memory system optimizations

After illustrating some cross-layer optimizations between protocol stack layers of a mobile phone, in this section the focus is at chipset level, where memory system power optimization, a key optimization in the design of mobile terminals, is taken into consideration. The section surveys some broad used memory technology, introduces and explains the need for powerful power-modeling tools and finally describes the System-on-Chip (SoC) modeling approach proposed by our work.

The problem of power management in terminals has been broadly discussed in literature [10], [11] and has mainly identified processor power consumption as the most important contributor to the overall system power consumption. However, recent analyses show that the power consumption caused by the memory subsystem represents a significant part of the total power. In [13] authors report that about 35% of the total energy of a Samsung Galaxy S3 I9300 is due to data movements in the memory system on video-playback applications. As a consequence one can say that the memory hierarchy of mobile phone chipset, and more in general the overall memory system, do require a higher attention when power optimizations are to be looked for. Hence, designers of memory system would need easy-to-use and high-precision power models that estimate power and energy consumption of the different operations and state transitions of the memory system. Among those, DRAMPower [14] is an open source Dynamic Random Access Memory (DRAM) power and energy tool for estimating the power consumption of different DRAM operations at the cycle-accurate level.

Fig. 4 represents in an abstract and general view the most important functional blocks of a cellular modem SoC, in the left side the Application Processor (AP), the Communication Processor (CP), and the bus (Interconnection), and on the right side the memory system.

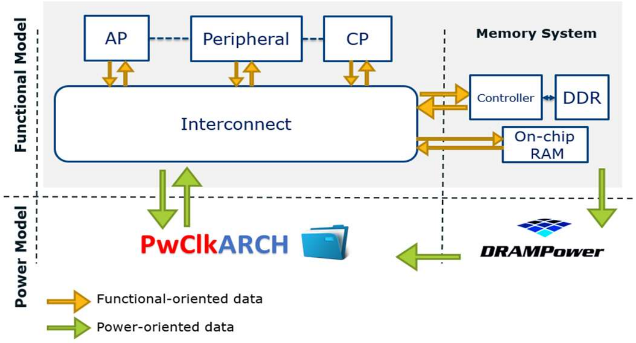

Fig. 4. *Abstract view of a mobile SoC, highlighting the memory system and the related power and simulation frameworks*

The memory system is composed of a controller, potentially some cache, and different kinds of memories, e.g. on-chip RAM and a generic external memory that is used as the main memory of the processor, which could be implemented via Double Data Rate (DDR) DRAM, which is known to be the traditional main memory, or using other more recent technologies like the Low Power DDRx (LPDDRx) family.

The problem of performance and energy optimization of the memory system is widely discussed in literature [13]-[18]. There are several memory technologies that are currently used in mobile SoC, the most traditional one being DRAM. As DRAM memories are approaching fundamental technology limitations, e.g. poor scaling and high power consumption to refresh the stored information, the industry is turning towards new main memory types, like the non-volatile memory (NVM). In [15] NVM and its many attractive aspects are discussed, such as low leakage current and high density. Though NVM also has drawbacks, like the increased dynamic power during write operation, reliability and the high production cost. That is why a new type of NVM, the spin transfer torque magnetic random access memory (STT-MRAM) is widely considered as one of the most promising candidates for next generation main memory. But if STT-MRAM is to be used as main memory, there is the problem that no agreed-upon standard interface to link the STT-MRAM with the controller has been defined, therefore designers have two options: either design a STT-MRAM specific memory controller, or adapt an existing DDRx standard [16], the latter being adopted by the majority of the designers. For example [17] introduces an optimized STT-MRAM interface which is totally compatible to the state-of-the-art LPDDR3 specification, originally designed for DRAM. Particularly interesting and at the same time rather complex is the designing of hybrid memory architectures, so to take advantage of the different types of technology [18].

Memories performance and power consumption do not depend only on memory design but also on workload (e.g. the cellular protocol stack executed on the CP), the kind of memory controller in use and on the overall system configuration. The interactions between the memory system and the different components of the mobile SoC need to be accurately considered, to achieve a satisfactory estimation of power consumption. The PwClkARCH library [19] presents a framework that helps designers to model the impact of the SoC activities on power consumption by means of Virtual Prototyping (VP), describing the SoC using SystemC-Transaction Level Modeling (TLM). The library also allows for exploration of different power management strategies, like Dynamic Frequency Scaling (DFS), Dynamic Voltage Frequency Scaling (DVFS), and Clock gating.

The PwClkARCH library by itself is still not sufficient to give a precise view on main memory power consumption, especially at the refinement level of each operation. In fact it is such information that is needed to properly explore different configurations and compare different designs of memory systems. Therefore we need to make work together the PwClkARCH library with the DRAMPower tool.

Our work [20], [21] focuses on incorporating STT-MRAM timing and power parameters given in [17] to the DRAMPower memory simulator and to the PwClkARCH framework, in order to explore different memory configurations and study the impact of memory parameters on power and performance. DRAMPower can provide information about power-refresh of the main memory and other specific operations (Read, Write, etc.), which are needed to compare different platform memory technologies and platform designs, so to find out which one has the lowest request on power.

This approach is to be used at a very early stage of the SoC design flow, as it focuses on a pre-silicon simulation environment at a high level, before SW and HW have been designed. Joint optimization of, e.g., cellular protocol stack SW architecture, mobile SoC HW architecture as well as HW/SW partitioning is enabled by this approach. In fact our work allows to analyze the impact of different architecture options on mobile SoC power consumption as well as on the performance of cellular protocol stack SW, executed on the CP.

IV. SW DEVELOPMENT TEST AND VERIFICATION ASPECTS

Product development of complex embedded systems, like cellular modems for mobile devices, requires large engineering organizations with hundreds of embedded SW developers. The layering of communication protocol stacks allows to manage complexity by separating the concerns of different subject matter among different teams of experts. As protocol stack optimizations are more and more adopting the cross-layer approach, product SW development have to be enhanced to cope with that. In fact a traditional SW development tool would consider the different layers, into which a wireless communication protocol stack is split (see blocks of Fig.1), as almost independent entities, with their own features and optimization roadmap. Such silo-approach to SW development mainly stems from the fact that 3GPP standards create technical specification targeted at a single layer, each one with its own KPIs, features and parameters, which are unique among the different layers, thus implying a parallel and almost independent evolution of the SW in each block.

As a consequence, project managers have to carefully plan the additional efforts required for adding cross-layer optimizations capability to the normal SW development flow, in order to stay within project schedule and budget constraints. Therefore, a development environment supporting the required cooperation between embedded SW developers from different teams, in an efficient and effective manner, is critical for successful introduction of cross-layer optimization in commercial products.

Complex SW development is based on the concept of mainline, i.e. the bulk of features that constitute the core of the SW functionality. Such mainline is continuously updated and enhanced, adding new features and functionalities as mandated by customer requests, optimizations, or by standards bodies. For complex SW enhancements, a branch is used, i.e. a derivation from the mainline, which later on could converge back into the mainline or could create an independent final product.

Depending on the size of the change and the number of developers, cross-layer optimization (indicated by orange arrows in Fig.5 and Fig.6) could be implemented on a separate branch or directly on the mainline of SW development together with other changes on the mainline (indicated by blue arrows in Fig.5 and Fig.6). In both cases all individual changes need to be verified together.

Fig. 5 and Fig. 6 show the different approaches that SW verification and testing managers could follow when dealing with cross-optimization enhancements.

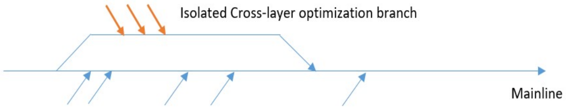

Fig. 5. *Isolated cross-layer optimizazion on different branches*

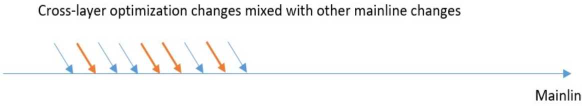
Cross-layer optimization changes mixed with other mainline changes
Mainline

Fig. 6. *Joint cross-layer and mainline changes*

Implementing cross-layer optimizations on the isolated branch makes verification easier, but has a higher risk of creating merge conflicts during the integration to the mainline. Therefore it is recommended to implement pre- and post-commit testing. Testing scope should cover regression as well as special test cases designed for cross-layer changes. Another good practice for implementing cross-layer optimization is to ask SW developers from different layers for peer reviews.

Finally, it is worth mentioning that when implementing cross-layer optimizations it is important to be aware of the security implications. One way to ensure that when a change spanning across different layers is introduced it does not compromise security, is to use static code analysis.

V. Conclusion and Future Work

This paper surveys some implications of adopting cross-layer optimizations techniques in cellular phones. Design aspects of the memory system at SoC level, and of the access stratum of a cellular protocol stack are discussed. Finally also SoC power and performance modeling as well as SW test and verification issues are touched.

Future work goes in the direction of exploring novel power-efficient memory architectures for mobile SoC, in refining the SW test and verification tools, and in further tuning the cellular protocol stack access stratum optimizations.

References

[1] B. Raaf, M. Faerber, B. Badic, and V. Frascolla, “Key technology advancements driving mobile communications from generation to generation,” *Intel Technology Journal*, 18(1), 2014.

[2] V. Frascolla, M. Faerber, E. Calvanese-Strinati, et. al., “mmWave Use cases and Prototyping: a way towards 5G Standardization,” In *Proc. 24th European Conference on Networks and Communications (EUCNC)*, Paris, Jun. 2015, pp. 128-132.

[3] V. Frascolla, M. Faerber, L. Dussopt, et al., "Challenges and opportunities for millimeter-wave mobile access standardisation,", *IEEE Global Communications Conference (GLOBCOM)*, Austin, Dec 2014.

[4] M. Tercero, P. von Wrycza, A. Amah, et al, “5G systems: The mmMAGIC project perspective on use cases and challenges between 6–100 GHz,” *In Proc. IEEE Wireless Communications and Networking Conference (WCNC*), Doha, 03-06 Apr. 2016, pp. 200-205.

[5] A Morgado, A. Gomes, V. Frascolla, et al., “Dynamic LSA for 5G networks the ADEL perspective”, *In Proc 24th European Conference on Networks and Communications (EuCNC)*, Paris, Jul. 2015, pp.190-194.

[6] C. She, C. Yang, and T. Quek, “Cross-layer optimization for ultra-reliable and low-latency radio access networks,” In *IEEE Transaction on wireless communications*, 17(1), Jan 2018, pp. 127-141.

[7] Y, Teng, and M. Song, “Cross-layer optimization and protocol analysis for cognitive ad hoc communications,” In *IEEE Access*, vol. 5, 2017.

[8] N. Baldo, and M. Zorzi, “Fuzzy logic for cross-layer optimizazion in cognitive radio networks,” In *IEEE Communication Magazine*, Apr. 2008, pp.64-71.

[9] D. Szczesny, S. Hessel, A. Showk, at al., “Joint uplink and downlink performance profiling of LTE protocol processing on a mobile platform,” *International Journal of Embedded and Real-Time Communication Systems*, 1(4), Oct. 2010, pp. 21-39.

[10] S. Trabulsi, V. Frascolla, N. Pohl, et al., “A versatile low-power ciphering and integrity protection unit for LTE-advanced mobile devices”, *IEEE 10th International New Circuits and Systems Conference (NEWCAS)*, Montreal, Jun. 2012.

[11] S Traboulsi, N Pohl, J Hausner, et al., “Power analysis and optimization of the ZUC stream cipher for LTE-advanced mobile terminals,” *IEEE 3th Latin American Symposium on Circuits and Systems (LASCAS)*, Playa del Carmen, Feb. 2012.

[12] J. Ah Sue, P. Brand, J. Brendel, et al. “A predictive dynamic power management for LTE-Advanced mobile devices,” *In Proc. IEEE Wireless Communications and Networking Conference (WCNC),* Barcelona, Apr. 2018.

[13] D. Pandiyan and C. J. Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," *In Proc 2014 IEEE International Symposium on Workload Characterization (IISWC)*, Raleigh, NC, 2014, pp. 171-180.

[14] K. Chandrasekar, B. Akesson and K. Goossens, "Improved Power Modeling of DDR SDRAMs," *In Proc. 14th Euromicro Conference on Digital System Design (DSD)*, Oulu, 2011, pp. 99-108.

[15] J. Hu, Q. Zhuge, C. J. Xue, et al., "Optimizing Data Allocation and Memory Configuration for Non-Volatile Memory Based Hybrid SPM on Embedded CMPs," *In Proc. 2012 IEEE* 26th International Parallel and Distributed Processing Symposium (IPDPS), Shanghai, 2012, pp. 982-989.

[16] K. Asifuzzaman, R. Sánchez-Verdejo, and P. Radojković, “Enabling a reliable STT-MRAM main memory simulation,” *In Proc. International Symposium on Memory Systems (MEMSYS)*, Oct. 2017, pp. 283-292.

[17] J. Wang, X. Dong, and Y. Xie, “Enabling High-performance LPDDRx-compatible MRAM,” *International Symposium on Low Power Electronics and Design (ISLPED)*, La Jolla, Aug. 2014.

[18] O. Mutlu, and L. Subramanian, “Research Problems and Opportunities in Memory Systems,”, *Supercomputing Frontiers and Innovations: an International Journal*, 1(3), Oct. 2014, pp. 19-55.

[19] H. Affes, M. Auguin, F. Verdier and A. Pegatoquet, "A methodology for inserting clock-management strategies in transaction-level models of systemon- chips," *Forum on Specification and Design Languages (FDL)*, Barcelona, 2015, pp. 1-7.

[20] A. Ben Ameur, D. Martinot, P. Guitton-Ouhamou, et al., "Power and Performance aware Electronic System Level Design," *13th International Symposium on Industrial Embedded Systems (SIES)*, Toulouse, Jun. 2017.

[21] A Ben Ameur, M. Auguin, F. Verdier, and V. Frascolla, "Mobile Terminals System-level Memory Exploration for Power and Performance Optimization,", *28th International Symposium on Power and Timing Modeling, Optimization and Simulation* (PATMOS), Costa Brava, Jul. 2018.

ISBN 978-90-827970-1-5 © EURASIP 2018