**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**
**OFFICE OF THE MEMBER SECRETARY OF THE COMMITTEE**
**FOR ADVANCED STUDIES & RESEARCH (CASR), RUET, RAJSHAHI.**

Application form for approval of **M. Sc. Engg**. thesis proposal by the CASR. All the items, which are applicable of the following list, must be mentioned and filled in properly. **Please submit ten (10) Copies**. [One original and other 9 photocopies in the **full script plain paper**].

**Date**: 15.10.2017

**1. Name of the student**: Nakib Aman **Status:** Part-Time

**Roll No**. 1604103001 **Session** 2016-2017

**2. Present Address**: Islam Tower, Mirzapur, Binodpur, Motihar Rajshahi-6204.

**3. Present Designation & Organization:** Lecturer, CSE Department, Varendra University

**4. Name of the Department:** CSE **Programme**: M.Sc. Engineering

**5. Name of the Supervisor:** Dr. Md. Al Mamun **Designation**: Professor

**6. Name of the Co-Supervisor (if any):** N/A **Designation**: N/A

**7. Date of First Enrolment in the Programme:** 27th March, 2017

**8. Tentative Title (Block Letters)**: SOFTWARE DEVELOPMENT EMULATION FOR EMBEDDED SYSTEMS.

**9. Background and present state of the problem:** Traditionally, embedded software development is inherently dependent on hardware availability. Hardware development includes design, simulation and testing of hardware architecture, logic, circuit schematics and finally the PCB. This is a very time-consuming process, often iterative and unpredictable due to dependency on factors like component availability and vendor support.

In embedded system software development there are two methods, bottom-up and top-down [1]. Bottom-up method consists of developing components at each abstraction level, starting at the bottom. Components at the bottom layer are then used to create higher level components, yielding a complete system. Top down on the other hand starts with a high level specification and from this specification smaller parts are specified and implemented. This iterates until a complete system is implemented. Both methods have its strengths and drawbacks. In bottom-up the developers have control over system metrics in each layer. Top-down might allow for an earlier start of software development, targeting the initial specification.

To be able to develop effective software for embedded systems it is important to know and understand the constraints hardware puts on the system. A big challenge here is the gap between hardware and software development, as the software development can usually not be started and especially not tested until hardware is available [2]. More hardware constraints can lead to extra costs and development time.

Waiting for hardware is the largest issue when it comes to embedded software development and it puts a lot of pressure on the software developers when it arrives. To change the process and let off some pressure from the developers, an emulated platform could help in several ways. Even though an emulator is not a full representation of a functional system but it could still be beneficial [3]. For example an emulator could be a way for initial testing and to find limitations before the physical hardware exists.

The goal behind hardware emulation is to construct a software model of the hardware, sometimes before it is completed in silicon, to test software functionality. This can be done in a numerous of ways with different methods and techniques. Some are whole systems designed for the purpose of emulating hardware, with FPGAs and complementary software [4]. These kind of solutions are usually commercial products. Other methods are software based and open-source. When the emulator is pure software, it has certain limitations. If hardware emulation can ease the platform transition process it is also possible that it could be a tool in daily development as well.

The ability to develop applications for a hardware without the physical device itself is a definitely a big advantage. However, there are other advantages provided by the emulator too. It can make a safe and secure environment for the testing of new and untried applications. Since modern emulators now support various kinds of hardware as well as different operating systems, the application can also be tested for scalability and reliability and not just functionality [5]. This makes virtualized environments very powerful and feasible testing platforms. Emulators also prove valuable in cases of kernel and device driver development, where a small mistake can crash the entire operating system. Developing and debugging drivers on an emulator, makes it similar to user-space applications, which at the worst can lead to the emulator crash.

Software based emulators have been around for a long time and predominantly used by developers of Operating Systems. With virtualization technology becoming more common and gaining in-processor support, these emulators have expanded to support several processor architectures and a richer feature set. Consequently, popular operating systems such as Linux have been adapted to run within virtual environments seamlessly. As part of the software development process, this virtualization technology has huge potential to increase product quality and developer efficiency [6].

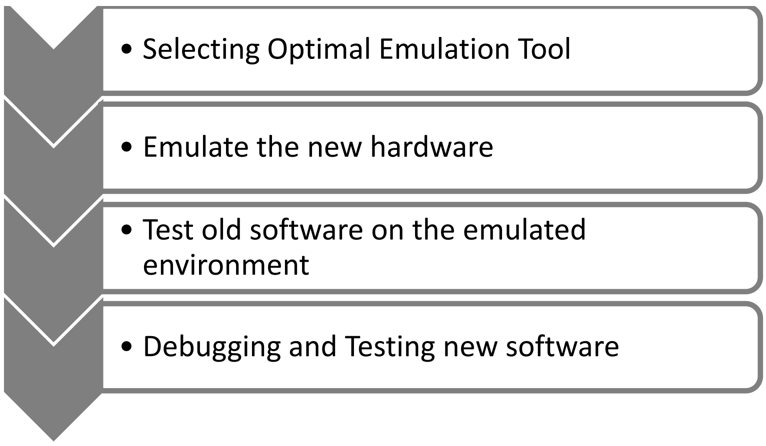
10. **Justification of the study:** Sampath and Rao conclude in a paper that it is possible to make an efficient development and testing platform for embedded systems using emulation [7]. The paper focuses much on the process of the development setup: which emulator to use, which development environment to be used and so on. QEMU is chosen as the most fitting emulation method for their system.

    A paper about testing embedded networked systems with an emulator concludes that emulation has great potential in embedded system design and development [8]. In this paper an alarm system used to detect fire and notify its surroundings is discussed. As the system is very large and wide spread it is more convenient to emulate the system to test it. However, the paper offers a good background for the possibilities to test software against virtual platforms.

A book by Boulé and Zilic brings up hardware emulation as a tool to dynamically test hardware simultaneously as developing it, which also enables software development [9]. The book also brings up the benefits of an emulator in comparison to simulation.

11. **Objectives with specific aims:** This thesis aims to investigate how hardware platform emulation affects software development. It is also possible to further develop this into a secure and efficient testing platform [10]. Another aim is to estimate if software modification is needed early. Problems that are part of the platform development might be easier to address with the emulator, as it gives the developer more control.

12. **Outline of Methodology/ Experimental Design:** A method is needed to evaluate the potential benefits of an emulator. In this case an emulation tool is needed to be chosen. The emulation in this thesis will not run at native speed, which will be taken in to consideration during the evaluation. The main issue here is compatibility between the new hardware and the old software. When the basics are in a functional state, they will be used to implement functionality similar to a real system. The applications can be run and debugged on the emulator just as they would on the real board [11].




13. **Expected outcome:** To investigate if it is possible to start the software development earlier and target an emulated platform. A way to start the software development and testing earlier would hopefully yield better systems and shorter overall development time.

## 14. References:


[1] G. Martin, "A career in system-level design research [review of "Embedded System Design: Modeling, Synthesis, and Verification (Gajski, D.D. et al; 2009)]," in *IEEE Design & Test of Computers*, vol. 27, no. 2, pp. 82-83, March-April 2010.

[2] P. Patel and M. Moallem, "Using FPGA-based platforms for embedded control applications in Mechatronics," *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Montreal, ON, 2010, pp. 1356-1361.

[3] F. Cucchetto, A. Lonardi and G. Pravadelli, "A common architecture for co-simulation of SystemC models in QEMU and OVP virtual platforms," *2014 22nd International Conference on Very Large Scale Integration (VLSI-SoC)*, Playa del Carmen, 2014, pp. 1-6.

[4] Garcia, C. Pacheco and A. Herrera, "Defining a Software Process Improvement-Based Methodology for Embedded Systems Development," *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, Morelos, 2010, pp. 120-125.

[5] H. Satria, B. Wibowo, J. B. Kwon, J. B. Lee and Y. S. Hwang, "VDEES: A virtual development environment for embedded software using open source software," in *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 959-966, May 2009.

[6] J. Horalek, R. Cimler and V. Sobeslav, "Virtualization solutions for higher education purposes," *2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA)*,Pardubice, 2015, pp. 383-388.

[7] L. Sartor, A. F. Lorenzon and A. C. S. Beck, "The Impact of Virtual Machines on Embedded Systems," *2015 IEEE 39th Annual Computer Software and Applications Conference*, Taichung, 2015, pp. 626-631.

[8] T. C. Yeh, G. F. Tseng and M. C. Chiang, "A fast cycle-accurate instruction set simulator based on QEMU and SystemC for SoC development," *Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, Valletta, 2010, pp. 1033-1038.

[9] M. Boule, J. S. Chenard and Z. Zilic, "Adding Debug Enhancements to Assertion Checkers for Hardware Emulation and Silicon Debug," *2006 International Conference on Computer Design*, San Jose, CA, 2006, pp. 294-299.

[10] M. Monton, A. Portero, M. Moreno, B. Martinez and J. Carrabina, "Mixed SW/SystemC SoC Emulation Framework," *2007 IEEE International Symposium on Industrial Electronics*, Vigo, 2007, pp. 2338-2341.

[11] H. m. Qian and C. Zheng, "A Embedded Software Testing Process Model," *2009 International Conference on Computational Intelligence and Software Engineering*, Wuhan, 2009, pp. 1-5.

**15. List of courses taken so far (To be verified and signed by the Course Advisor)**

| Sl. No. | Course No. | Course Name | Grade | Credit | Grade point | G. P. A |
|---|---|---|---|---|---|---|
| 1. | CSE 6001 | Embedded Systems | Continuing | 3 | | |
| 2. | CSE 6705 | Computer Arithmetic Analysis | Continuing | 3 | | |
| 3. | | | | | | |
| 4. | | | | | | |
| 5. | | | | | | |
| 6. | | | | | | |
| | | | | | | |

**Signature of the Course Advisor:** …………………………………

**16. Cost Estimate: (Invoice / Quotation must be provided for each and every items which cost Tk. 10,000/- and more)**

| | | |
|---|---|---|
| (a) Cost of Material (Breakup needed) | Tk | 5,000 |
| (b) Field works (if applicable) | | N/A |
| (c) Conveyance/ Data Collection (With Breakup) | Tk | 2,000 |
| (d) Typing, Drafting, Binding & Paper etc. | Tk | 5,000 |
| (e) Miscellaneous | Tk | 3,000 |
| Total: | Tk | 15,000 |

**17. Justification of having Co-Supervisor:**

Co-supervisor is not encouraged in Masters level thesis unless there is an absolute necessity. In the Ph.D level, Co-supervisor(s) are acceptable if required. In both cases it is required to justify by the supervisor that the work requires considerable knowledge of a discipline other than his own field of work.

**--------------------------------------- For Office Use Only ---------------------------------------------**

**18. Doctoral Committee/PGAC reference:**

**Meeting no. -------------------------Resolution No. -------------------------Date: -----------------------------**

**19. Appointment of Supervisor & Co-Supervisor Approved by the CASR Meeting No. (For Ph. D):-**

**Resolution No. ------------------------- Date-------------------------------------**

**20. Appointment of Doctoral Committee Approved by the CASR Meeting No. (For Ph. D)**:--------

**Resolution No. -------------------------- Date--------------------------------------**

**21. Result of the comprehensive examination for Ph. D (Photocopy of the result should be enclosed)**

**Date: -------------------------------------- Satisfactory/Unsatisfactory.**

**22. Number of Post-Graduate Student(s) working with the Supervisor at Present:**

<table>
<tr><td rowspan="9">--------------------------------<br>**Signature of the Student**<br><br>------------------------------------<br>**Signature of the Supervisor**<br><br>-----------------------------------------<br>**Signature of the Co-Supervisor**<br><br>-------------------------------------------------<br>---<br>**Signature of the Head of the Department**</td><td colspan="2">**Names and signatures of the members of the Doctoral Committee (if applicable)**</td></tr>
<tr><td>1.</td><td></td></tr>
<tr><td>2.</td><td></td></tr>
<tr><td>3.</td><td></td></tr>
<tr><td>4.</td><td></td></tr>
<tr><td>5.</td><td></td></tr>
<tr><td>6.</td><td></td></tr>
<tr><td>7.</td><td></td></tr>
<tr><td>8.</td><td></td></tr>
</table>