PRD for UEFI Applications and Diagnostics

Kadin Brooks, Ivaylo Kozhuharov, Halla Tuaum

Sponsor: Intel

By Halla Tuaum hallatea@uw.edu

Revision History Block

Date	Version	Section	Author
10/31/2022	0.1	All	Halla
10/31/2022	0.1	All	Kadin
10/31/2022	0.1	5	Ivaylo
11/12/2022	0.2	0, 2, 8, 9, 10, 11	Kadin
11/12/2022	0.2	0, 2, 8, 9, 10, 11	Ivaylo
11/12/2022	0.2	0, 2, 8, 9, 10, 11	Halla
11/27/2022	0.3	0, 1, 2, 4, 6, 7, 8	Kadin
11/27/2022	0.3	0, 1, 2, 4, 6, 7, 8	Ivaylo
4/20/2023	0.3	3, 4, 5, 6, 7	Kadin

Table of contents

0. Project Stakeholder

The sponsor of this senior design project is the Intel corporation. The main stakeholder of this senior design project is Harry Hsiung. Harry is a Technical Marketing Engineer for UEFI/Tiano at Intel. Laurie Jarlstrom is another Technical Marketing Engineer for UEFI/BIOS at Intel who is working with Harry Hsiung in offering this design project for senior engineering undergraduate students in computer engineering and electrical engineering. Laurie Jarlstrom develops training for firmware engineering in the UEFI/EDK II platforms. Harry Hsiung supports marketing efforts on promoting UEFI and ACPI for Intel, UEFI industry public events, training sessions, presentations, and products that support public UEFI specifications and the Tianocore code base. One of the main deliverables of our senior design project is to produce UEFI Shell applications which we may contribute to the open source Tianocore repository.

1. Project Goals

The goal of this senior design project is to create UEFI shell applications and diagnostic tools for our company sponsor, Intel. More specifically, our group has agreed with our sponsor to design a UEFI shell application for CPU identification and CPU feature enumeration, a UEFI shell application for describing memory topology, and a UEFI application for reporting SMM handler latency and size.

This project is sponsored by Intel, an American multinational corporation and technology company headquartered in Santa Clara, California. Intel is the largest semiconductor chip manufacturer by revenue and is one of the developers of the x86 series of instruction sets [1]. This senior design project was created by Intel for university students for the purpose of training senior engineering students in Intel system firmware and operating system design. The technical objectives of this project involves developing UEFI shell executable programs that will be contributed to the Tianocore open source repository. These programs should be able to be used by other engineers in the field that are working in areas related to CPU features, memory profiling, or the SMI handler profiling.

By designing these UEFI shell applications and diagnostic tools for our senior design project, we plan to address the main needs of our sponsor by obtaining important experience and skills that our sponsor and other related companies desire from recent graduates in the field of computer engineering. We also plan to contribute programs to the open source Tianocore repository that may be useful to other engineers working in this field.

2. Current State of the Art

TianoCore is an open source repository that contains implementations of the Unified Extensible Firmware Interface (UEFI) [5]. Professionals across the world contribute to the Tianocore repository for the purpose of enabling the development of UEFI firmware and tools through various open source projects. Most projects in the repository are currently related to the EDK II project. EDK II is a cross-platform development environment for the UEFI and UEFI Platform Initialization (PI) specifications. In 2004, Intel announced that it would release the Extensible Firmware Interface (EFI) under an open source license. This code was developed by Intel as part of a project named Tiano. This code base eventually evolved into EDK, EDK II, and other open source projects contained in the Tianocore.

There are two implementations of the CPU diagnostics tool that we found. One is CPU-Z [10], and it is a free software that gathers information on the system's devices. They use their own code that is not related to UEFI. There is also a UEFI application contained in the Tianocore code base that displays CPUID leaf information [9].

The TianoCore github [6] repository contains an existing implementation of the memory leak detection feature [8] that we are trying to implement. This implementation has the ability to find which specific line of code is asking for memory allocation, and then display the total memory that has been requested by that line of code. Our project will differ from this current implementation, as we will be showing the memory topology for devices connected to the system, and using that to find memory leaks.

Finally, the TianoCore github repository contains an existing implementation of a system management interrupt (SMI) handler profile feature [7]. This program reports the function name of the SMI handler, the name of the dispatcher the handler is registered with, the source file name and line number, and the SMI context (the SWSMI number) [7].

3. Project Objectives

This senior design project has three major objectives to complete. We will list the major objectives of this project in order of priority.

3.1 UEFI Shell Application for CPUID output processing for CPU Feature Enumeration

The goal of this objective is to learn how to detect CPU features by processing the various leaves of the CPUID instruction and model-specific registers and control-status registers for an Intel processor and the x86 based instruction set architecture. The validation of this project will be performed by designing and executing a UEFI shell application that processes and then dumps the contents of the CPUID instruction to the shell terminal and parses the attributes of the CPU in a human readable format.

3.2 UEFI Shell Application for Memory Profiling

The goal of this objective is to learn about designing a UEFI shell application that allows a developer to analyze the hardware memory reservation for a UEFI firmware implementation. The validation of this project will be performed by designing and executing a UEFI shell application that describes the memory locations and memory sizes reserved for various components of the UEFI firmware implementation.

3.3 UEFI Shell Application for System Management Mode Interrupt Handler Profiling

The goal of this objective is to learn about the SMI handler by designing a UEFI shell application which describes the function name of the SMI handler, the name of the dispatcher the handler is registered with, the source file name and line number, and the SMI context.

4. Project Constraints

4.1 Hardware Constraints

Our project must be completed using a student kit for firmware training provided from our sponsor. The kit includes an 8th generation Intel Core processor, a 19 volt seven ampere power supply, a DediProg flash programmer, a 128 GB NVMe SSD storage device, a 4 port USB hub, a FTDI USB cable, and USB to Host cable.

4.2 Software Constraints

The applications designed for our project must be executable on the UEFI shell and hardware provided by sponsor. We must use the C programming language and Python language for implementing our applications. We will use EDK II as a firmware development environment for UEFI specifications.

4.3 Sponsor Constraints

We must use resources from the Tianocore github repository for designing our UEFI applications and training materials provided from our sponsor. We also must use the UP Xtreme platform that will be provided by our Intel sponsor.

4.4 Self Imposed Constraints

We must complete this senior design project in a group of three team members. We must store our application files in a git repository. We must track development progress over the course of the project.

4.5 Tianocore Repository Constraints

Our UEFI shell executable programs must conform to the Tianocore standards before being contributed to the open source repository.

5. Project Attributes/Features

This senior design project has three major objectives to complete. We list the attributes and features of each major objective in sorted order based on priority.

5.1 UEFI Shell Application for CPUID output processing and CPU Feature Enumeration

Our sponsor desires that the application we design be able to list the number of cores the processor has, the number of threads the processor has, the name of the processor, the design specification of the processor, the process technology used to implement the processor, the maximum junction temperature of the processor, the core speed, the instruction sets compatible with the processor, and the cache topology of the processor.

5.2 UEFI Shell Application for Memory Profiling

Our sponsor desires that the application we design profile the memory reserved for various components of the system firmware.

If time permits, a second requirement for this application is that it will contain a memory leak detection feature that can track allocated memory by various routines included in the UEFI system firmware.

5.3 UEFI Shell Application for SMI Handler Profiling

Our sponsor desires that the application we create be able to report the function name of the SMI handler, the name of the dispatcher the handler is registered with, the source file name and line number, and the SMI context.

6. Release Criteria

6.1 CPUID output processing and CPU Feature Enumeration

(Alpha) Our goal will be to design a shell executable program that calls the CPUID instruction and processes CPUID instruction output to obtain CPU attributes.
(Beta) Our goal will be to print CPU attributes obtained from CPUID instruction data processing program to the shell terminal.
(Final Release) The final goal of this objective is to create a software tool that outputs the attributes of the CPU in a human readable format using a UEFI shell executable program that calls the CPUID instruction and processes the data output from this instruction.

6.2 Memory Profiling Feature

(Alpha) Our goal will be to design a shell executable program that reports memory allocation for system firmware and operating system use.
(Beta) Our second goal will be to add memory leak detection functionality to our UEFI application.
(Final Release) The final goal of this objective is to design a UEFI shell executable program that tracks memory allocated for system firmware and the operating system. A second goal is to extend the functionality of this program to track memory allocation by various UEFI service calls.

6.3 SMI Handler Profiling Feature

(Alpha) Our goal will be to design a shell executable program that reports information about the SMI handler to the shell terminal.
(Beta) Our second goal will be to process the raw text file output by the SMI handler application using a Python script to a text file containing information presented in a human readable format.
(Final Release) The final goal of this objective is to validate the UEFI shell executable program that combines each of these features before release.

7. Timeline

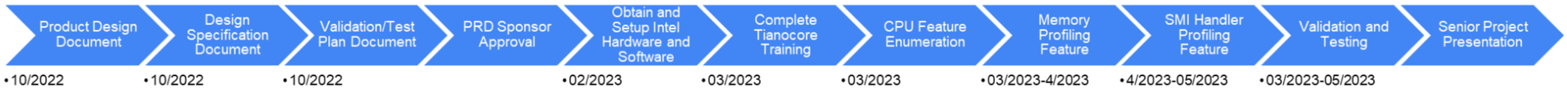

Figure 1. 2022-2023 Senior Design Project Timeline

10/2022 - 12/2022:

During this period, we will obtain hardware and software tools for the project, establish communication with our sponsors, and start writing core documentation to detail key features of the design project and track progress for the remainder of the year.

12/2023 - 03/2023

During this period, we will focus primarily on developing each deliverable of the senior design project in the following order: CPUID output processing, Memory Profiling, and finally SMI Handler Profiling. These deliverables will be developed sequentially with the higher priority deliverables coming before the lower priority deliverables.

03/2023 - 05/2023

During this period, we will focus primarily on validation and testing of each deliverable we were able to complete during the previous period and prepare for the Senior Project Presentation. We will also complete the documentation that will be submitted along with our deliverables.

8. Project Budget

Description	Cost	Link
SUT - Up Xtreme Celeron	$323.00	https://www.mouser.com/ProductDetail/AAEON-UP/UPX-WHLCR-A20-04064?qs=B6kkDfuK7%2FC5jq4xpW8wPA%3D%3D
Up Xtreme Power Supply and power cables	$11.99	https://www.mouser.com/ProductDetail/AAEON-UP/EP-PS19V342A65W?qs=%2Fha2pyFaduiqfanly%2FCwaF%2F%2FjQUqcttLfh8v91unCEev66gluJN7JQ%3D%3D
USB and Serial cable (mouser)	$9.99	https://www.mouser.com/ProductDetail/AAEON-UP/EP-CBUSB10PFL01?qs=%2Fha2pyFadujIhdWT%2FBgIH%252B8eLhJc3tYwKVNFRQzH3zKWZ4kE8rH3iQ%3D%3D
FTDI USB 3.3V to Serial Cable	$15.29	https://www.amazon.com/Converter-Terminated-Galileo-BeagleBone-Minnowboard/dp/B06ZYPLFNB/ref=sr_1_1?dchild=1&keywords=ftdi+3.3v+max&qid=1600360299&sr=8-1
Test lead wire for FTDI 2 10 Pin USB Up Xtreme		
USB Thumb drive(amazon)	$4.83	https://www.amazon.com/SanDisk-SDCZ50-008G-10PK-Everything-Stromboli-Lanyard/dp/B00FVWVDSK/ref=sr_1_13?dchild=1&keywords=usb+thumb&qid=1600360666&refinements=p_89%3ASanDisk&rnid=2528832011&s=electronics&sr=1-13
USB debug 3.0 cable	$15	https://www.datapro.net/products/usb-3-0-super-speed-a-a-debugging-cable.html
400gig 3710 ssd's (2.5" sata). Or 256gb nvme ssd	$23	https://www.amazon.com/Acclamator-Internal-Performance-Laptop-Desktop/dp/B08V8PM6WF/ref=asc_df_B08V8PM6WF?tag=bngsmtphsnus-20&linkCode=df0&hvadid=80264475882810&hvnetw=s&hvqmt=e&hvbmt=be&hvdev=c&hvlocint=&hvlocphy=&hvtargid=pla-4583863999438162&th=1
4 Port Gigethernet Switch 8 Port Netgear	$30	https://www.amazon.com/dp/B07PFYM5MZ/ref=redir_mobile_desktop?_encoding=UTF8&aaxitk=10y6uwuAgElsVt1xGEs0Ag&hsa_cr_id=2888142090101&pd_rd_plhdr=t&pd_rd_r=74fc7888-0b0f-45b2-b160-bc8fd64213db&pd_rd_w=yqDqD&pd_rd_wg=dL7fw&ref_=sbx_be_s_sparkle_mcd_asin_1_img
4 Port USB 3.0 hub	$15	https://www.amazon.com/Anker-Extended-MacBook-Surface-Notebook/dp/B07L32B9C2/ref=sr_1_4?dchild=1&keywords=anker+usb+3.0+hub&qid=1611189256&sr=8-4

9. Engineering Standards

To make a contribution to the Tianocore repository, we must create a change description in a specified format to use in the source control commit log. The commit message must include our signature in a specified format. The code must be submitted to the Tianocore project using the process that the project documents on its web page. If the process is not documented, then we must submit the code on the development email list for the project. The project must be submitted using the same copyright license as the base project. The remaining requirements for submissions to the Tianocore repository are provided in the github link provided in our list of references at the end of this document.

10. Impact of Engineering Solutions

10.1 Public Health

We do not expect there to be any public health impacts that arise from this project.

10.2 Safety Impacts

We need to ensure that our code conforms to standards of the Tianocore repository.

10.3 Public Welfare Impacts

We do not expect there to be any public welfare impacts to arise from this project.

10.4 Global Impacts

We plan to contribute our work to the Tianocore repository for open source access.

10.5 Cultural Impacts

We do not expect there to be any cultural impacts to arise from this project.

10.6 Societal Impacts

We do not expect there to be any societal impacts to arise from this project.

10.7 Environmental Impacts

We do not expect there to be any environmental impacts to arise from this project.

10.8 Economic Impacts

We plan to contribute our work to the Tianocore repository for open source access.

11. Ethical Considerations

For this project, we need to make sure that our work meets the standards provided by our sponsor. We need to acknowledge our sources whenever possible since our work will be building off previous work in the Tianocore repository. We need to assure that our UEFI Shell applications do not disrupt the functionality of the hardware resources that we will be using for this project. We do not have other ethical considerations at this time.

12. References

1. "Intel," *Wikipedia*, 28-Nov-2022. [Online]. Available: https://en.wikipedia.org/wiki/Intel. [Accessed: 27-Nov-2022].
2. V. Zimmer, M. Rothman, and S. Marisetty, *Beyond bios: Developing with the Unified Extensible Firmware Interface*. De G Press, 2017.
3. M. Rothman, V. Zimmer, and T. Lewis, *Harnessing the UEFI shell moving the platform beyond DOS, second edition*. Boston, MA: Walter de Gruyter, 2017.
4. *Unified Extensible Firmware Interface (UEFI) Specification*, Release 2.10. UEFI Forum, Inc. From the official UEFI Forum
5. *What is TianoCore?*, 12-Jan-2022. [Online]. Available: https://www.tianocore.org/. [Accessed: 27-Nov-2022].
6. Tianocore, "Tianocore/EDK2: EDK II," *GitHub*, 27-Nov-2022. [Online]. Available: https://github.com/tianocore/edk2. [Accessed: 27-Nov-2022].
7. Tianocore, "SMI handler profile feature · tianocore/tianocore.github.io wiki," *GitHub*, 15-Mar-2017. [Online]. Available: https://github.com/tianocore/tianocore.github.io/wiki/SMI-handler-profile-feature. [Accessed: 27-Nov-2022].
8. Tianocore, "Memory leak detection with memory profile feature · tianocore/tianocore.github.io wiki," *GitHub*, 10-Jul-2016. [Online]. Available: https://github.com/tianocore/tianocore.github.io/wiki/Memory-leak-detection-with-memory-profile-feature. [Accessed: 27-Nov-2022].
9. Tianocore, "Edk2/cpuid.c at master · tianocore/EDK2," *GitHub,* 7-Dec-2021. [Online]. Available: https://github.com/tianocore/edk2/blob/master/UefiCpuPkg/Application/Cpuid/Cpuid.c. [Accessed: 27-Nov-2022].
10. "CPU-Z: Softwares," *CPUID*, 2022. [Online]. Available: https://www.cpuid.com/softwares/cpu-z.html. [Accessed: 27-Nov-2022].