Martin Haubenwallner

# Finding and Analyzing Undocumented Model-Specific Registers on x86_64

**Bachelor's Thesis**

Graz University of Technology

Institute for Applied Information Processing and Communications

Advisor: Michael Schwarz

Graz, October 2020

# Abstract

Model-specific registers (MSRs) are a fundamental part of the x86_64 instruction set architecture. They are used, among others, to toggle CPU features, debugging, and monitoring. As an example, the IA32_LSTAR MSR shall be noted. It contains the instruction pointer of the system-call handler, that is executed by the `syscall` instruction. Model-specific in this context means that each micro-architecture may have a different set of MSRs. Both AMD and Intel provide a substantial list of documented MSRs as part of their developer manuals.

However, already conducted research has shown that undocumented MSRs may provide security-critical information or functionality, for example, enabling debugging modes or reading CPU internal states [6]. On Linux, a toolset is available for working with MSRs, consisting of the `msr` kernel module and the relating msr-tools which can be used for reading from and writing to MSRs from userspace. Unfortunately, there was no high-performance scanning tool to find undocumented MSRs and no tools for analyzing data from dynamic MSRs.

In this thesis, we lay the foundation for further research of both undocumented and documented model-specific registers (MSRs) on x86_64 systems. We first design and then implement software to find and analyze both documented and undocumented MSRs. In the next step, we scan for and find undocumented MSRs on various x86_64 CPUs (Intel and AMD). We then statistically analyze the found static undocumented MSRs and formulate a hypothesis based on the found anomalies in the data. Using our sampling and analyzing tools, we then monitor the found dynamic MSRs and conclude as to what their purpose might be and what their data might mean.

**Keywords:** MSR, Model-Specific Register, Undocumented, x86_64, x64, Intel, AMD, Scanning

# Contents

# Chapter 1

# Introduction

Developing highly sophisticated devices such as modern x86_64 CPUs is a very complicated task that requires an immense amount of testing to make sure all parts with their respective features work reliable and correct. Documents such as the errata sheets of various vendors, e.g. Intel or AMD, show that despite thorough testing, bugs still happen.

A fundamental tool for testing these processors during development are so-called model-specific registers or MSRs for short. These special registers are used, for example, for monitoring CPU states, and enabling and disabling CPU features [9]. Each microarchitecture has its own set of MSRs.

x86_64 CPU developers such as Intel or AMD document a lot of their CPU's MSRs. However, previous experiments, such as the one by Christopher Domas [6], have shown that there are also undocumented ones. In software development, we can see that debug functionality, that is not removed before the product is released, often leads to security vulnerabilities [23]. As it may very well be that the same is true for hardware, undocumented MSRs will become of high interest for information security researchers since undocumented MSRs may give further insight into the inner workings of CPUs, or even pose security vulnerabilities themselves.

To lay a foundation for more efficient research in the field of MSRs, we develop a software toolset that allows easy and efficient scanning for MSRs that are present on an x86_64 system. Additionally, we conduct initial analytical experiments and present first methods to categorize undocumented MSRs. We conduct all of our experiments on multiple x86_64 microarchitectures from two manufacturers, namely Intel and AMD. We discuss the different behavior and techniques of these manufacturers we observed as well as the found undocumented dynamic MSRs and their respective properties we discovered.

Due to the very complex nature of CPUs, this thesis mainly focuses on undocumented dynamic MSRs, which are undocumented MSRs that do not have fixed values but rather change over time. Undocumented static MSRs are statistically analyzed. Furthermore,

we discuss the various other aspects of MSRs as future work, such as reverse-engineering flags or monitoring CPU behavior when specific values are changed.

To summarize we make the following contributions:

- We design and implement a software toolset to find and analyze undocumented MSRs.
- We analyze four Intel CPUs and one AMD CPU and find 5880 undocumented MSRs.
- We statistically analyze the undocumented static MSRs on four Intel CPUs and one AMD CPU and find unexpected anomalies.
- We present techniques on how to categorize undocumented dynamic MSRs and apply these to our found MSRs.

# Chapter 2

# Background

In this chapter, we give a basic introduction to microarchitectures, model-specific registers, CPU privilege rings, and other topics needed to understand the work done in this thesis.

## 2.1 Microarchitectures

Each microarchitecture is based on a so-called instruction set architecture (ISA) such as x86_64. ISAs define in an abstract way how the CPU works [7]. Specifically, they define the available instructions, registers, and addressing formats, to name a few. A specific implementation of such an ISA is called a microarchitecture. There may be multiple microarchitectures implementing the same ISA. As a consequence, all microarchitectures based on the same ISA can execute the same binaries. Additionally, it does not make a difference for programmers on which microarchitecture of a certain ISA they are working on as long as they do not wish to use microarchitecture specific extensions. However, there may be some differences between microarchitectures that are allowed by the ISA. For instance, there are model-specific registers which may be different on each microarchitecture [9].

## 2.2 CPU Rings

x86_64 CPUs implement four different privilege levels, so-called rings. These rings are implemented to increase security. For example, they prevent potentially malicious user programs which are running in ring three, which is the least privileged ring, from executing potentially harmful instructions such as `hlt`, which halts the CPU. Rings three to one are for user-mode, whereas ring zero is for kernel-mode. While ring zero has full privileges and can thus execute all instructions, rings three to one can only use certain subsets of the instruction set [9]. In theory, ring three is for user-mode, ring two is for system-mode, ring one is for drivers, and ring zero is for kernel-mode. In practice, only

ring three and zero are used with the operating system kernel running in ring zero and the user programs running in ring three.

Ring zero is usually referred to as kernelspace, while ring three is commonly referred to as userspace.

## 2.3 Model-Specific Registers (MSRs)

CPUs tend to get more and more complex, embedding more and more features and functionality. With this rising complexity, there arises a need to control and monitor these features. x86_64, leverages model-specific registers, which are 64-bit wide registers with a 32-bit address. As can be seen in CPU manufacturer documentations, these registers are used, among others, for toggling CPU features, monitoring system performance, and debugging [1,9]. "Model-specific" describes the fact that each microarchitecture may have its own set of MSRs, each with their individual addresses and functionalities. Though the address-functionality-mapping usually correlates between microarchitectures, this is not necessarily the case [9].

Christopher Domas [6] has shown that there exist undocumented MSRs which may expose security-critical functionality or information.

There are over one thousand different documented MSRs that provide hundreds of different functions and settings on Intel processors alone [9].

### 2.3.1 Instructions for Modifying MSRs

x86_64 offers two instructions, `rdmsr` and `wrmsr`, to read from and write to MSRs respectively. These instructions can be executed either in real mode or protected mode [9]. In protected mode, these instructions are only available in ring 0 and thus require kernel privileges to execute.

For both instructions, the address of the target MSR is specified in the RCX register, whereas the higher 32 bits are ignored [9]. `rdmsr` reads the values of the respective MSR into the registers RDX:RAX with RDX storing the higher 32 bits and RAX storing the lower 32 bits of the 64-bit MSR value. The higher 32 bits of the RDX and RAX registers are cleared. `wrmsr` writes the value in RDX to the higher 32 bits and the value in RAX to the lower 32 bits of the MSR. The higher 32 bits of both RDX and RAX are ignored.

If one executes either of the two instructions on an MSR that is reserved or unimplemented, the CPU generates a General Protection (GP) exception [9]. Additionally, a GP is generated if `wrmsr` would change reserved bits in an implemented MSR.

## 2.4 Syscall

As described in Section 2.2, there are two primary privilege levels, or rings, on modern x86_64 CPUs (namely three and zero). To escalate from ring three to ring zero, the

`syscall` instruction is used [9]. This instruction invokes a system handler that was previously defined by the operating system by loading the respective address from the IA32_LSTAR MSR (0xc0000084) into the instruction pointer register (RIP) and setting the privilege level to zero. This way, the operating system can provide necessary privileged services, such as reading from a file or writing to a network socket, to unprivileged user-processes in a safe way. Once the operating system served the request of the user-process it deescalates back to ring three via the `iret` instruction and the user-process resumes right after the `syscall` instruction [9].

## 2.5 Linux Kernel Module

As stated above, the `wrmsr` and `rdmsr` instructions cannot be executed in userspace as they require kernelspace privileges. With Linux kernel modules, Linux offers a way to dynamically extend the Linux kernel at runtime. In principle, a Linux kernel module (LKM) is a program that is compiled to a kernel object, which may then be loaded into the Linux kernel and is executed in ring 0. Usually, the programming language of choice for creating an LKM is C.

## 2.6 Input/Output Control (ioctl)

On Linux, the number of syscalls has already exceeded 330 [32]. Adding a new syscall is usually discouraged as it requires it to be implemented in every architecture the kernel supports and is less flexible than ioctls [22]. Instead, it is recommended to create a character device and implement ioctl calls. The abbreviation ioctl stands for **i**nput/**o**utput **c**on**t**ro**l** and was added to manipulate the parameters of the underlying device [18]. It is basically a way to implement device-specific "syscalls" that are called ioctl calls. In short, a handler for the unlocked ioctl operation [5] on the respective device is added, which gets invoked when an ioctl syscall with an open file descriptor to the underlying device as a parameter is made. With this technology, it is possible to implement dynamic driver-specific syscall-like services that are executed in ring zero. This means that it is possible for kernel modules to provide syscall like interfaces to communicate with the driver.

**Listing 2.1:** The ioctl call function signature [18]

```
int ioctl(int fd, unsigned long request, ...);
```

The function signature of the ioctl call can be seen in Listing 2.1. It has to contain a valid file descriptor to the respective desired device, a request number, as well as an untyped pointer to memory [18].

## 2.7 Misc Device

In Linux, each device driver has a major number assigned to it [21]. Each individual device of this device driver gets a minor number assigned and can thus be identified by the major minor value pair. Allocating a major device number makes sense when the driver has multiple devices and thus entry points. Every action on an individual device falls back to the responsible device driver, which is the driver with the respective major number. As it is not feasible to allocate a new major number for each small service, Linux offers a simplified interface that allows for single devices to be registered. This interface is realized as the misc device driver, which has the major number 10 assigned [21]. It is possible to register a new device on this driver and then override all desired file operations of this device.

## 2.8 Exception Handling in the Linux Kernel

As described before, the `rdmsr` and `wrmsr` instructions may throw different exceptions during execution. Unhandled exceptions in an LKM usually cause the executing process to be killed. Fortunately, there exists a way to handle exceptions in the Linux kernel that has similar semantics to try/catch of higher-level languages like C++ or Python [16]. One of the key differences is that this mechanism works on the assembly level instead of being accessible through a higher level language like C itself.

To achieve try-catch behavior on certain instructions an entry containing the address of the respective instruction that may generate an exception, the address of the fixup code, and the address of the handler function that shall be executed when an exception is thrown is added to the __ex_table section. The so-called fixup code is stored in the "fixup" section and is called by the handler if an exception is raised on the specific instruction specified in the __ex_table. As this fixup code is called in the interrupt context, it should be as performant as possible. The handler has to have the function type depicted in Listing 2.2.

**Listing 2.2:** The ex_handler_t type [31]

```
typedef bool (*ex_handler_t)(
const struct exception_table_entry *,
struct pt_regs *,
int,
unsigned long,
unsigned long);
```

As one might expect, this kind of exception handling is often used in the Linux kernel, and thus, there are convenient macros that make working with this mechanism easier. Listing 2.3 shows the macro that creates the entry in the __ex_table. The `from` parameter specifies the address at which an exception might be thrown. The `to` parameter specifies

the location of the fixup code. The `handler` parameter specifies the location of the handler that shall be called to handle exceptions.

**Listing 2.3:** The _ASM_EXTABLE_HANDLE macro [31]

```
#define _ASM_EXTABLE_HANDLE(from, to, handler)    \
.pushsection "__ex_table","a" ;                   \
.balign 4 ;                                       \
.long (from) - . ;                                \
.long (to) - . ;                                  \
.long (handler) - . ;                             \
.popsection
```

There already exists a default handler, that calls the corresponding fixup code when an exception was generated, and a respective extable macro, see Listings 2.4 and 2.5.

**Listing 2.4:** The ex_handler_default function [31]

```
__visible bool ex_handler_default(
const struct exception_table_entry *fixup,
struct pt_regs *regs,
int trapnr,
unsigned long error_code,
unsigned long fault_addr)
{
regs->ip = ex_fixup_addr(fixup);
return true;
}
```

**Listing 2.5:** The _ASM_EXTABLE macro [31]

```
# define _ASM_EXTABLE(from, to)   \
_ASM_EXTABLE_HANDLE(from, to, ex_handler_default)
```

## 2.9 `rdmsr_safe` and `wrmsr_safe`

With the building blocks described in Section 2.8 it is trivial to create a safe version of both `rdmsr` and `wrmsr`. As reading from and writing to MSRs is a quite common task to perform for an x86_64 operating system these functions have already been written and merged into the kernel upstream [31], see Listings 2.6 and 2.7. These implementations provide an error value that can be used to check if the respective read or write operation was successful or whether an exception was generated.

**Listing 2.6:** The native_read_msr_safe function [31]

```
static inline unsigned long long native_read_msr_safe(
unsigned int msr, int *err)
```

```
{
DECLARE_ARGS(val, low, high);

asm volatile("2: rdmsr ; xor %[err],%[err]\n"
"1:\n\t"
".section .fixup,\"ax\"\n\t"
"3: mov %[fault],%[err]\n\t"
"xorl %%eax, %%eax\n\t"
"xorl %%edx, %%edx\n\t"
"jmp 1b\n\t"
".previous\n\t"
_ASM_EXTABLE(2b, 3b)
: [err] "=r" (*err), EAX_EDX_RET(val, low, high)
: "c" (msr), [fault] "i" (-EIO));
if (msr_tracepoint_active(__tracepoint_read_msr))
do_trace_read_msr(msr, EAX_EDX_VAL(val, low, high),
*err);
return EAX_EDX_VAL(val, low, high);
}
```

**Listing 2.7:** The native_write_msr_safe function [31]

```
static inline int notrace
native_write_msr_safe(unsigned int msr, u32 low, u32 high)
{
int err;

asm volatile("2: wrmsr ; xor %[err],%[err]\n"
"1:\n\t"
".section .fixup,\"ax\"\n\t"
"3: mov %[fault],%[err] ; jmp 1b\n\t"
".previous\n\t"
_ASM_EXTABLE(2b, 3b)
: [err] "=a" (err)
: "c" (msr), "0" (low), "d" (high),
[fault] "i" (-EIO)
: "memory");
if (msr_tracepoint_active(__tracepoint_write_msr))
do_trace_write_msr(msr, ((u64)high << 32 | low),
err);
return err;
}
```

## 2.10 Pearson and Spearman Correlation Coefficients

For this section, let two variables $X$ and $Y$ be defined as:

$$X = \{x_1, x_2, x_3, ..., x_n\} \tag{2.1}$$

$$Y = \{y_1, y_2, y_3, ..., y_n\} \tag{2.2}$$

With $x_i$ and $y_i$ each denoting the $i$th data sample.

As Xu and Deng describe in their paper [33], the Pearson correlation coefficient is used to determine the correlation of two normal continuous variables. It yields a value $r$ in the interval [-1, 1]. The meaning of $r$ for two variables $X$ and $Y$ can be expressed with the following function:

$$\text{Meaning of } r = \begin{cases} \text{X and Y have a complete positive correlation} & \text{if } r = 1 \\ \text{X and Y have a positive correlation} & \text{if } 0 < r < 1 \\ \text{X and Y have no obvious correlation} & \text{if } r = 0 \\ \text{X and Y have a negative correlation} & \text{if } -1 < r < 0 \\ \text{X and Y have a complete negative correlation} & \text{if } r = -1 \end{cases} \tag{2.3}$$

The Spearman rank correlation coefficient is described by Chok in his thesis [3] as a rank-based version of the Pearson correlation coefficient. Similarly to Pearson's correlation coefficient, the Spearman rank correlation coefficient yields a value $r_s$ for two variables X and Y in the interval [-1, 1]. The meaning of $r_s$ can be expressed through the same function as the result of Pearson, see Equation 2.3. Chok further notes, that Spearman's coefficient can not only find complete correlations for linearly related variables but also for variables that are related in a non-linear monotonic way. However, Spearman's correlation coefficient can still be 0 for variables that have a non-monotonic relationship.

As the Pearson correlation coefficient is more outlier sensitive than the Spearman correlation coefficient, Pearson seems to better reflect the overall concordance if there are only a few outliers [3]. Both of these coefficients are used in this thesis and the resulting framework to offer more conclusive results.

# Chapter 3

# Design

To fulfill our goal of templating undocumented MSRs, we develope a software toolset that is split into two main parts, the kernel part, and the userspace part, which are explained in this chapter. For an abstract visualization of this software toolset see Figure 3.1.

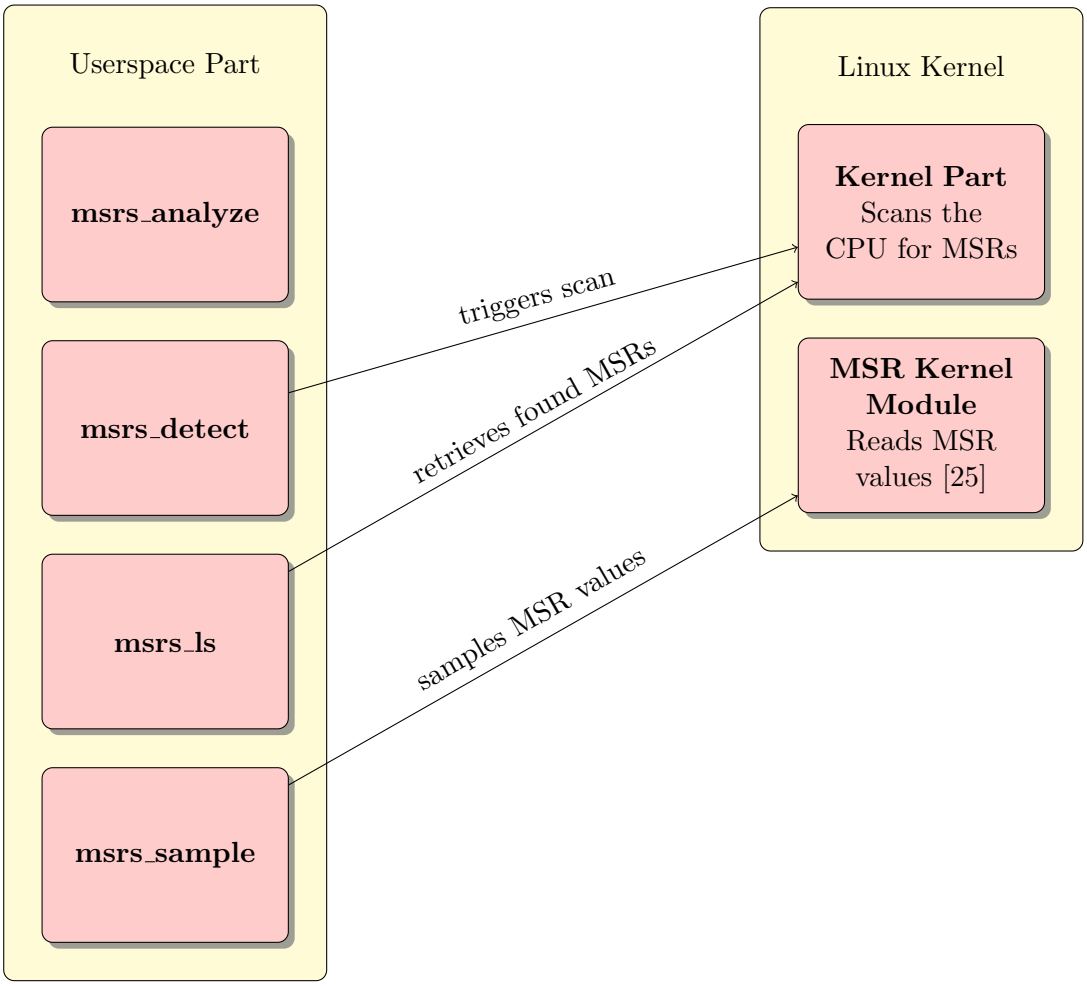



**Figure 3.1:** This figure shows an abstract visualization of the software toolset.

## 3.1 Overview

Fulfilling our main goal of finding undocumented MSRs and analyzing the dynamic ones requires multiple steps:

1. Get a list of the MSRs that exist on the respective CPU.
2. Extract all undocumented MSRs from this list.
3. Check for each MSR if it is static or dynamic.
4. Analyze the dynamic MSRs.

## 3.2 Kernel Part

MSRs can only be read from and written to in ring zero. Hence, the scanning for MSRs has to be done in kernelspace.

The kernel part is responsible for two things:

**scanning** scan the whole range of possible MSRs

**exposing** expose a list of the found MSRs to userspace

The platform of choice for the toolset is Linux due to it being open-source and easily modifiable. There are two ways to extend the functionality of the Linux kernel. One can either create a custom kernel, thus actually modifying the kernel itself by extending the functionality, or one can create a kernel module that can be loaded into the kernel at runtime. Creating a custom kernel comes with the upside of having all functions of the kernel available but with the downside of being far less flexible as each user of the toolset would have to install and boot the custom kernel. Additionally, a custom kernel would come with a far more significant maintenance overhead as all upstream changes of the Linux kernel would have to be implemented into the custom kernel if one does not want it to go severely out of date quite quickly. A kernel module is far more flexible as it can be loaded into and out of the running kernel and poses less maintenance overhead. However, it needs to be recompiled for each new kernel version as APIs may have changed. Due to the nature of our application being a tool and the anticipated maintenance overhead, we decide to implement the functionality via a kernel module.

The whole scanning process is carried out solely in kernelspace in order to gain performance by avoiding costly and, in this case, unnecessary inconveniences like context switches.

To achieve even more performance, the scanning is carried out multithreaded. While this poses a quite high CPU load on multiple cores, it also drastically decreases the run time.

Linux offers multiple ways to communicate between kernelspace and userspace, including sockets, procfs, sysfs, syscalls, and ioctls [31]. We decide to use ioctls as it poses the least overhead and is quite easy to implement. Our kernel module creates a misc device that allows triggering a scan via ioctl and reading the list of found MSRs via the pread syscall [26]. As there already exists an in-tree kernel module for reading and writing from and to MSRs, called simply msr [25], we decide against implementing this feature.

## 3.3 Userspace Part

We also need a set of userspace tools that enable us to communicate with our kernel module, sample MSRs, as well as analyze the collected data.

The needed functionality is split into four tools: msrs_detect, msrs_ls, msrs_sample, and msrs_analyze.

### 3.3.1 msrs_detect

msrs_detect is a simple program that triggers a new scan with a user-defined number of workers via an ioctl call.

### 3.3.2 msrs_ls

msrs_ls prints all MSRs that were found by the module.

### 3.3.3 msrs_sample

msrs_sample samples MSRs specified by the user on all available cores for a user-specified amount of time and prints the results to stdout. All values are represented in hexadecimal notation without a prefix or postfix. Lines starting with # are comments and do not denote a sample value. The output format is described in Table 3.1.

**Table 3.1:** The format of the output generated by msrs_sample.

| Line | Description | Example |
|---|---|---|
| 1 | The list of sampled CPUs separated by a comma. | f,e,d,c,b,a,9,8,7,6,5,4,3,2,1,0 |
| 2 | The list of sampled MSRs separated by a comma. | 102,108,10a,110,118,11f |
| 3 | The sampling time. | 12c |
| 4 - N | Each line represents one read. Each individual line, or read, contains a list of MSR values per CPU, which are separated by a comma, separated by a semicolon. The order of MSRs as well as CPUs is the same as in the respective lines 1 and 2. | MSR102_CPU0,MSR102_CPU1, ...,MSR102_CPUf;MSR108_CPU0, ...,MSR108_CPUf;...;MSR11f_CPU0, ...,MSR11f_CPUf\n |

### 3.3.4 msrs_analyze

msrs_analyze is a tool to analyze the data samples generated by msrs_sample. For this purpose, it provides the two main functions "plot" and "corr".

The "plot" function checks, per MSR, for changes on each CPU and plots the data of all CPUs in a single line plot. If data changes are present on a CPU, the tool additionally generates a line plot containing only the data of this specific CPU, a bit heatmap, that shows the bit changes of all reads, as well as a line plot which shows the data changes as well as the reverse data changes.

The "corr" function calculates the Pearson as well as the Spearman correlation coefficients, see Section 2.10, between a specified MSR and other MSRs in the individual sample file on the raw sample data as well as on the data changes. If a user-defined similarity threshold is reached, the tool creates a plot visualizing the similarity.

The "changes" function analyzes the values of each MSR per CPU and looks for changes in the sampled values. Changes means that there are different values read from the same MSR over time. If changes are found the address of the respective MSR is printed on stdout.

# Chapter 4

# Implementation

In this chapter, we will discuss the various implementation details of the kernel module and the userspace tools.

## 4.1 Kernel Module (msrs)

The kernel module is written in C. Multithreading is implemented by using a concurrency managed workqueue [8]. For communication purposes with the userspace, a misc device, see Section 2.7, with the name `msrs` is created. The path to the driver device file is `/dev/msrs`. This device implements four file operations as shown in Table 4.1.

**Table 4.1:** Implemented file operations.

| Operation | Description |
|---|---|
| open | Used by the `open` [27] syscall. This call will lock the device preventing it being opened multiple times at the same time. |
| read | Used by the `pread` [26] syscall. It is used to receive the found MSRs by reading from the device. Each 64-bit value represents an MSR address that was found. |
| unlocked_ioctl | Used for `ioctl` calls. `unlocked_ioctl` [5] does, in contrast to an `ioctl` call, not run under the Big Kernel Lock (BKL) [4]. |
| release | Used to release the device. It is used by the `close` [28] syscall. Unlocks the device. |

The `ioctl` interface provides one call, which has to be provided with an integer that specifies how many workers to use for starting an MSR scan. To avoid race conditions, only one process at a time is allowed to have a valid open file descriptor of the driver device file. If a process attempts to open the device file while another process already

holds an open file descriptor, the device will return -EBUSY indicating that the resource is busy [30].

## 4.2 Userspace

Except for the msrs_analyze tool, which is written in Python 3, all other userspace tools are written in C++17.

### 4.2.1 msrs_detect

**Listing 4.1:** Usage of msrs_detect

```
./msrs_detect thread_count
```

Listing 4.1 shows the usage of the msrs_detect tool. All arguments are described in the table 4.2.

**Table 4.2:** Command line arguments of msrs_detect.

| Argument | Description |
|---|---|
| thread_count | This mandatory positional argument specifies the number of workers to be used for the scanning process. The value has to be in $\{n \in \mathbb{N} \mid n \leq \text{UINT\_MAX}\}$. A thread_count too high may cause the system to freeze. |

### 4.2.2 msrs_ls

**Listing 4.2:** Usage of msrs_ls

```
./msrs_ls
```

Listing 4.2 shows the usage of the msrs_ls tool. It takes no arguments but directly reads all found MSR addresses and prints them, separated by a `\n`, in hexadecimal format to the standard output.

### 4.2.3 msrs_sample

**Listing 4.3:** Usage of msrs_sample

```
./msrs_sample duration msr [msr]...
```

Listing 4.3 shows the usage of the msrs_sample tool. All arguments are described in Table 4.3. The tool prints the sample data in the format described in Section 3.3.3 to the standard output. As all values are printed in hexadecimal, neither prefix, as for example `0x`, nor a postfix, as for example `H`, is added to the values.

**Table 4.3:** Command line arguments of msrs_sample.

| Argument | Description |
|---|---|
| duration | This mandatory positional argument specifies the number of seconds the tool shall sample the specified MSRs. It has to be provided in decimal. The time value has to be in $\{n \in \mathbb{N} \mid n < 2^{32}\}$. |
| msr [msr]... | At least one MSR has to be specified. Additional MSRs may be specified. The MSR addresses may be provided in decimal, octal (prefix: 0), or hexadecimal (prefix: 0x). Each provided MSR value has to be in $\{n \in \mathbb{N} \cup \{0\} \mid n < 2^{32}\}$. |

### 4.2.4 msrs_analyze

As noted above, this tool is written in Python 3. The reason for this is that Python offers a large variety of libraries for scientifically analyzing and plotting data, like NumPy [19], SciPy [24], and matplotlib [29], and additionally has integrated support for unlimited range integers [20].

**Listing 4.4:** Usage of msrs_analyze

```
python3 msrs_analyze.py [-h] [-a ACTION] [-m MSR] [--msrs
    MSRS] [-s SIMILARITY_THRESHOLD] [-c CPUS] sample_file
```

Listing 4.4 shows the usage of the msrs_analyze tool. All arguments are described in Table 4.4.

**Table 4.4:** Command line arguments of msrs_analyze.

| **Argument** | **Description** |
|---|---|
| sample_file | This mandatory positional argument specifies the file containing the sampled data that shall be analyzed. |
| -h, –help | Shows help information. |
| -a, –action | Specifies the action to execute (one of: plot, corr, changes). The default is plot. |
| –m, –msr | Specifies the main MSR used for comparison. Mandatory for action "corr". |
| –msrs | Specifies the MSRs that shall be used for the selected action. The default is: "all". |
| -s, –similarity-threshold | Specifies the similarity threshold which is a value between 0 and 1 with 0 being no connection at all and 1 stating complete equality, see also Section 2.10. This threshold has to be reached for the tool to generate a plot. The default is: 0.75 |
| -c, –cpus | Specifies the CPUs that shall be used for the selected action. The default is "all". |

# Chapter 5

# Fundamental Assessment

In this chapter, we lay the needed foundation for our experiments. We specify the experimentation setup, gather information, and conduct first fundamental experiments that provide us with the necessary data needed for more specialized experiments and the final general assessment of this thesis.

## 5.1 Experimentation Setup

In Table 5.1, the CPUs used for experimentation are depicted.

**Table 5.1:** Table of x86_64 CPUs used for experimentation.

| CPU | Manufacturer | Microarchitecture |
|---|---|---|
| Ryzen Threadripper 1920X [2] | AMD | Zen |
| Core i7-6700K [11] | Intel | Skylake |
| Core i7-8700K [12] | Intel | Coffee Lake |
| Core i9-9900K [13] | Intel | Coffee Lake |
| Xeon Silver 4208 [15] | Intel | Cascade Lake |

## 5.2 MSR Scans

We use the tools described before to scan for MSRs on each CPU depicted in Table 5.1. For each individual CPU, all documented MSRs are extracted from the respective Intel [9] and AMD [1] manuals. Please note that for AMD, we exclusively take into account the documented MSRs and not the specified ranges they provide in their document. We use this data to determine which found MSRs are documented and which are undocumented.

The extraction of the undocumented MSRs is done in a semi-automatic way using the tools described in Chapter 3, as well as basic command line tools such as `comm`. On our

CPUs the MSR scans took between 1 minute 19 seconds (AMD Ryzen Threadripper 1920X) and 8 minutes 43 seconds (Intel Core i7-6700K with hyper-threading disabled). Based on the results of multiple scans conducted on each CPU the scans seem to be fully deterministic, thus yield the same results each time. A summary of the results of this first scan can be found in Table 5.2.

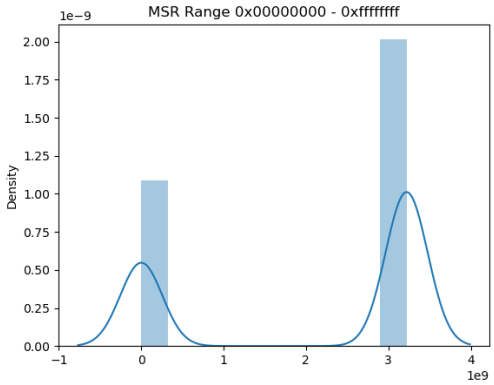
**Table 5.2:** Summarized results of the fundamental scan.

| CPU | Number of Found MSRs | Number of Undocumented MSRs |
|---|---|---|
| Ryzen Threadripper 1920X [2] | 5240 | 4875 (93.03%) |
| Core i7-6700K [11] | 477 | 104 (21.80%) |
| Core i7-8700K [12] | 520 | 124 (23.85%) |
| Core i9-9900K [13] | 532 | 134 (25.19%) |
| Xeon Silver 4208 [15] | 1098 | 643 (58.56%) |

A list of all detected MSRs can be found in the appendix in Section 10.1.

### 5.2.1 MSR Ranges

**Figure 5.1:** The distribution of MSRs over the whole range.




Based on the combined results of the found MSRs on all analyzed CPUs we found that there are apparently three used value ranges which we call $R_1$, $R_2$, and $R_3$ . They are defined as follows:

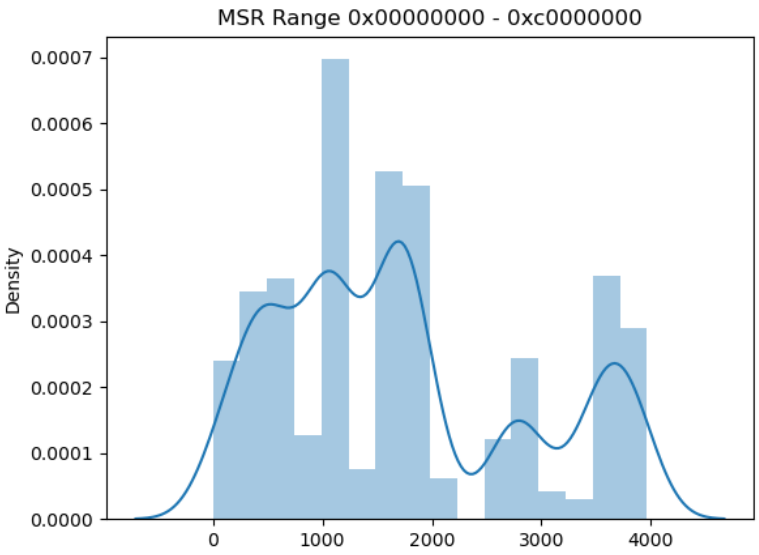
$$R_1 = \{n \in \mathbb{N} \cup \{0\} \,|n < 0xc0000000\} \tag{5.1}$$

$$R_2 = \{n \in \mathbb{N} \cup \{0\} \,|0xc0000000 \leq n < 0xc0010000\} \tag{5.2}$$

$$R_3 = \{n \in \mathbb{N} \cup \{0\} \,|n \geq 0xc0010000\} \tag{5.3}$$

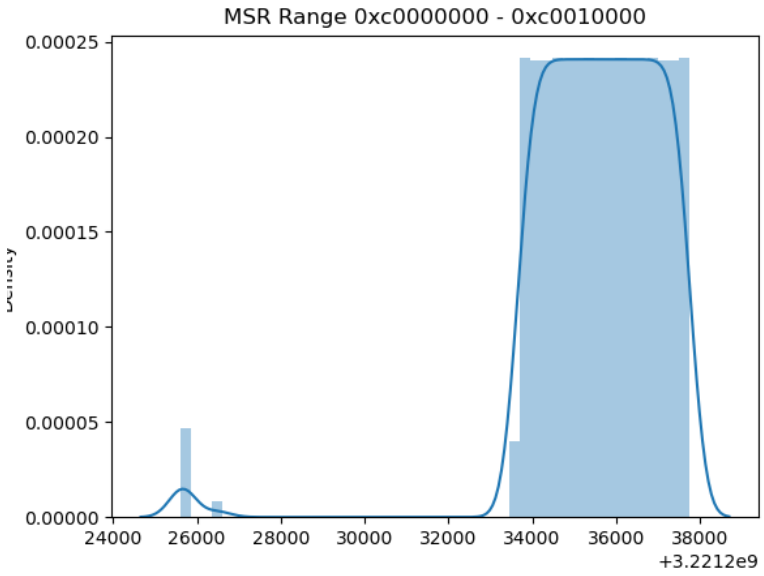
Empirically, using the actually existing MSRs on our CPUs, we found the ranges $R_1^{emp}$, $R_2^{emp}$, and $R_2^{emp}$, see the respective Figures 5.2, 5.3, and 5.4, on our machines to be:

**Figure 5.2:** The distribution of MSRs in $R_1^{emp}$.



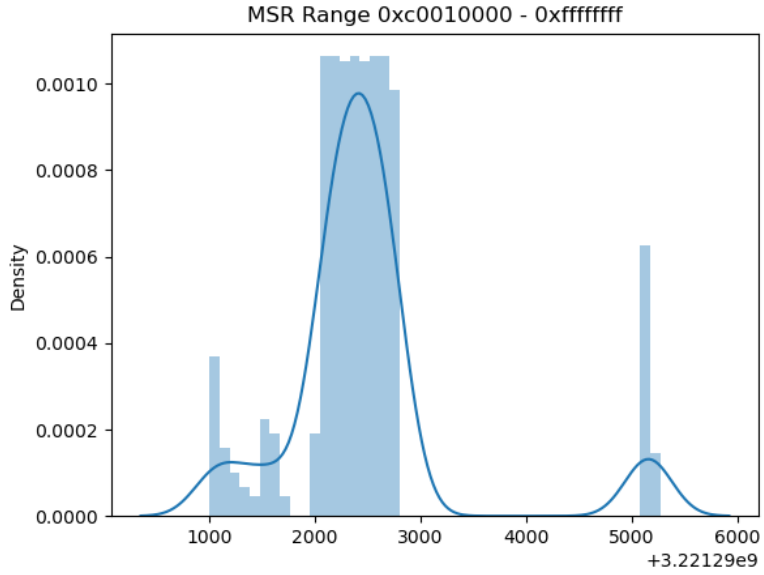

$$R_1^{emp} = \{n \in \mathbb{N} \cup \{0\} \,|n \leq 0xf7b\} \tag{5.4}$$

**Figure 5.3:** The distribution of MSRs in $R_2^{emp}$.




$$R_2^{emp} = \{n \in \mathbb{N} | 0xc0000080 \leq n \leq 0xc0002fff\} \tag{5.5}$$

**Figure 5.4:** The distribution of MSRs in $R_3^{emp}$.



$$R_3^{emp} = \{n \in \mathbb{N} | 0xc0010000 \leq n \leq 0xc00110a2\} \tag{5.6}$$

## 5.3 Differences AMD and Intel

We see a vast difference in the number of existing MSRs with the AMD CPU which has with 5240 MSRs nearly five times as many MSRs as the Intel processor with the most MSRs in our setup, which is the Intel Xeon Silver 4208 with 1098 MSRs. When comparing the number of existing MSRs divided into two groups, specifically documented and undocumented we see that our scanned AMD and Intel processors do not differ significantly in terms of existing documented MSRs but differ significantly in terms of existing undocumented MSRs. In numbers this means that the AMD CPU has 4875 undocumented MSRs whereas the Intel CPU with the most undocumented MSRs, which is again the Intel Xeon Silver 4208, has only 643 undocumented MSRs. For a detailed overview see Table 5.2.

# Chapter 6

# Analyzing Statistical Parameters of Static Undocumented MSRs

In this chapter we evaluate statistical parameters of the static undocumented MSRs that we found on the tested CPUs and discuss potential anomalies.

## 6.1 AMD Ryzen Threadripper 1920X

As we can see in Table 6.1 all 12 CPUs, except for CPU 0 and CPU 6, have 4852 undocumented static MSRs that are zero and 21 undocumented static MSRs that are non-zero. CPUs 0 and 6 have 4804 undocumented static MSRs with the value zero and 69 undocumented static MSRs that are non-zero as can be seen in Table 6.2. In Table 6.2 we can additionally see that at least 19 of the 48 additional MSRs have the value d01a000000000000. This is also visible when comparing Figures 6.1 and 6.2.

**Table 6.1:** Static MSRs with value zero compared to static MSRs with a non-zero value on the AMD Ryzen Threadripper 1920X.

| CPU | Number of Zero MSRs | Number of Non-Zero MSRs |
|---|---|---|
| 0 | 4804 | 69 (0.014%) |
| 1 | 4852 | 21 (0.004%) |
| 2 | 4852 | 21 (0.004%) |
| 3 | 4852 | 21 (0.004%) |
| 4 | 4852 | 21 (0.004%) |
| 5 | 4852 | 21 (0.004%) |
| 6 | 4804 | 69 (0.014%) |
| 7 | 4852 | 21 (0.004%) |
| 8 | 4852 | 21 (0.004%) |
| 9 | 4852 | 21 (0.004%) |
| 10 | 4852 | 21 (0.004%) |
| 11 | 4852 | 21 (0.004%) |

**Table 6.2:** Statistical parameters of static non-zero MSRs on the AMD Ryzen Threadripper 1920X.

| CPU | Minimum | Maximum | Median | Mode |
|---|---|---|---|---|
| 0 | 0000000000000001 | d400005d51505800 | 00000000feb00000 | d01a000000000000 (x20) |
| 1 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 2 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 3 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 4 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 5 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 6 | 0000000000000001 | d400005d51505800 | 00000000feb00000 | d01a000000000000 (x20) |
| 7 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 8 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 9 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 10 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |
| 11 | 0000000000000001 | d400005d51505800 | 000000001808cc17 | 0000000000000001 (x1) |

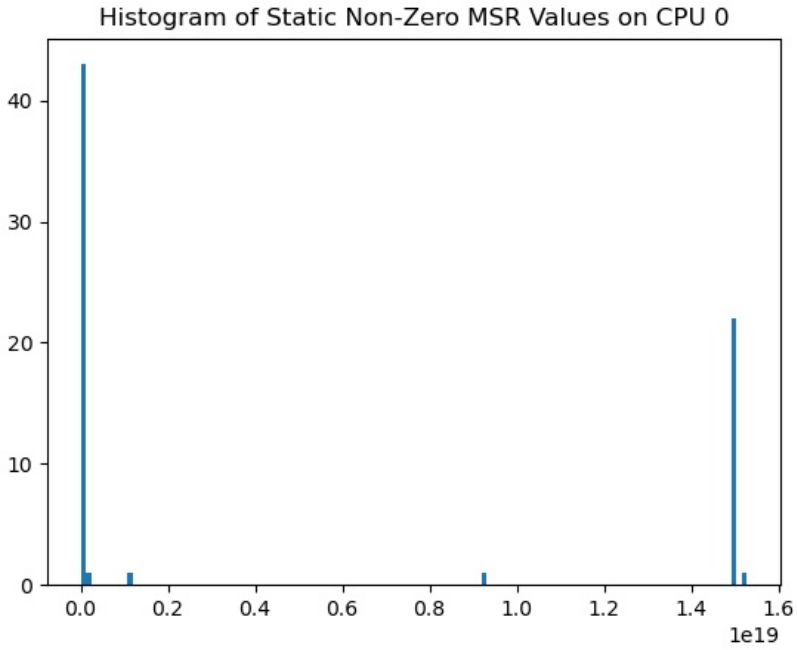



**Figure 6.1:** Histogram of the static non-zero MSR values on the AMD Ryzen Threadripper 1920X on CPU 0.

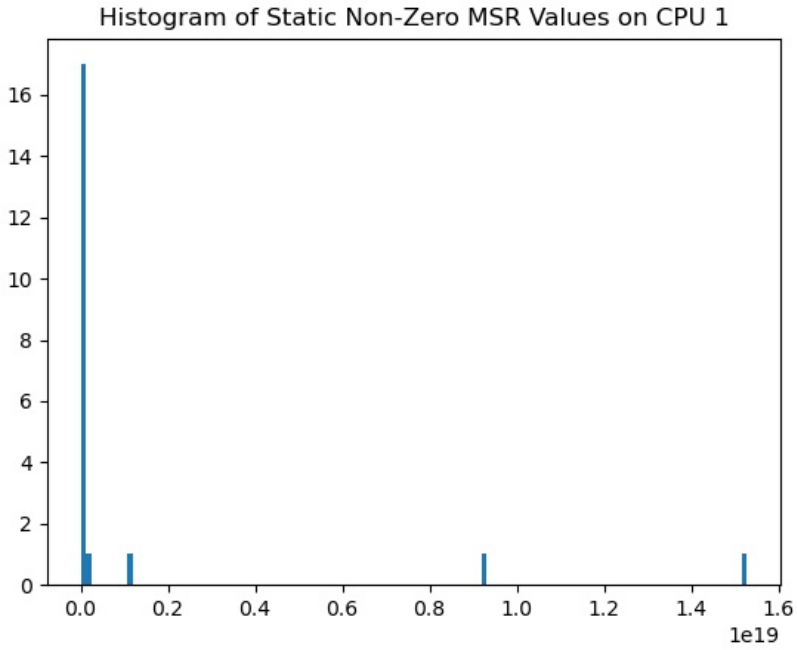



**Figure 6.2:** Histogram of the static non-zero MSR values on the AMD Ryzen Threadripper 1920X on CPU 1.

## 6.2 Intel Core i7-6700K

The Intel Core i7-6700K has a total of 70 undocumented static MSRs that are zero and 29 that are non-zero, see Table 6.3. In Table 6.4 can be seen that the most common value among the undocumented static non-zero MSRs is the same as the minimum, namely 1. The histogram of the non-zero values can be found in Figure 6.3.

**Table 6.3:** Static MSRs with value zero compared to static MSRs with a non-zero value on the Intel Core i7-6700K.

| CPU | Number of Zero MSRs | Number of Non-Zero MSRs |
|---|---|---|
| 0 | 70 | 29 (0.414%) |
| 1 | 70 | 29 (0.414%) |
| 2 | 70 | 29 (0.414%) |
| 3 | 70 | 29 (0.414%) |

**Table 6.4:** Statistical parameters of static non-zero MSRs on the Intel Core i7-6700K.

| CPU | Minimum | Maximum | Median | Mode |
|---|---|---|---|---|
| 0 | 0000000000000001 | fd0cd1679876a800 | 0000000000401cc0 | 0000000000000001 (x6) |
| 1 | 0000000000000001 | fd0cd1679876a800 | 0000000000401cc0 | 0000000000000001 (x6) |
| 2 | 0000000000000001 | fd0cd1679876a800 | 0000000000401cc0 | 0000000000000001 (x6) |
| 3 | 0000000000000001 | fd0cd1679876a800 | 0000000000401cc0 | 0000000000000001 (x6) |

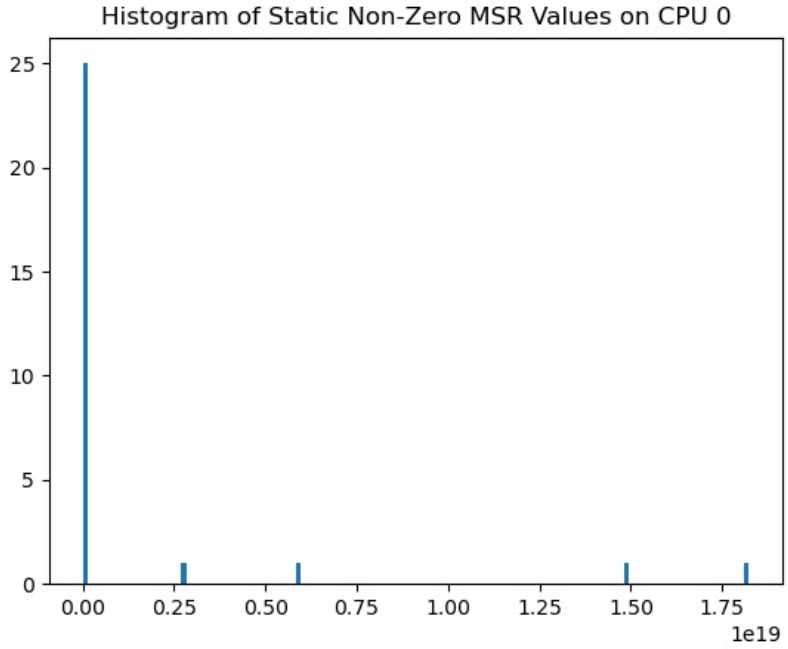



**Figure 6.3:** Histogram of the static non-zero MSR values on the Intel Core i7-6700K on CPU 0.

## 6.3 Intel Core i7-8700K

The Intel Core i7-8700K has a total of 89 undocumented static MSRs that are zero and 32 that are non-zero, see Table 6.5. In Table 6.6 can be seen that the most common value among the undocumented static non-zero MSRs is the same as the minimum, namely 1. The histogram of the non-zero values can be found in Figure 6.4.

**Table 6.5:** Static MSRs with value zero compared to static MSRs with a non-zero value on the Intel Core i7-8700K.

| CPU | Number of Zero MSRs | Number of Non-Zero MSRs |
|---|---|---|
| 0 | 89 | 32 (0.360%) |
| 1 | 89 | 32 (0.360%) |
| 2 | 89 | 32 (0.360%) |
| 3 | 89 | 32 (0.360%) |
| 4 | 89 | 32 (0.360%) |
| 5 | 89 | 32 (0.360%) |
| 6 | 89 | 32 (0.360%) |
| 7 | 89 | 32 (0.360%) |
| 8 | 89 | 32 (0.360%) |
| 9 | 89 | 32 (0.360%) |
| 10 | 89 | 32 (0.360%) |
| 11 | 89 | 32 (0.360%) |

**Table 6.6:** Statistical parameters of static non-zero MSRs on the Intel Core i7-8700K.

| CPU | Minimum | Maximum | Median | Mode |
|---|---|---|---|---|
| 0 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 1 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 2 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 3 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 4 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 5 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 6 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 7 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 8 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 9 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 10 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |
| 11 | 0000000000000001 | fd0cd1679876a800 | 000000000003c362 | 0000000000000001 (x6) |

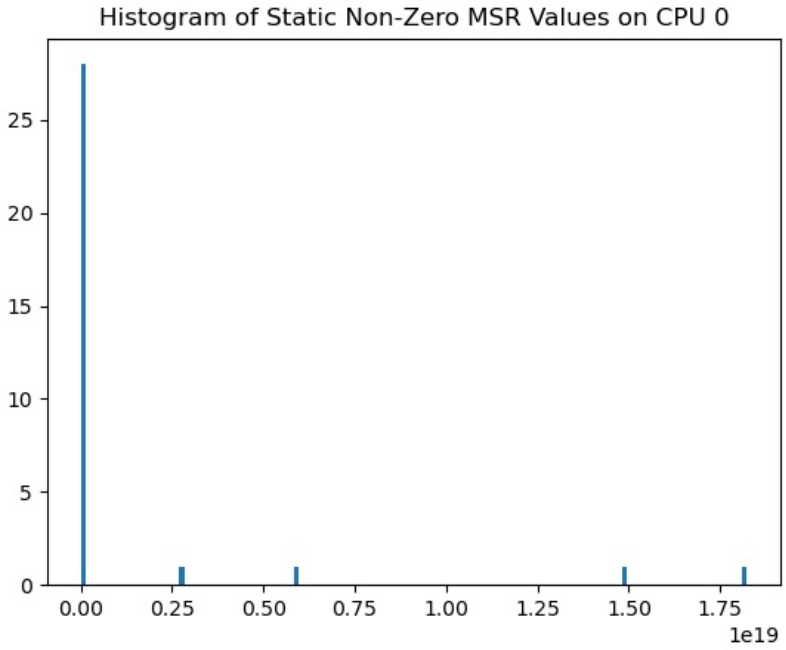



**Figure 6.4:** Histogram of the static non-zero MSR values on the Intel Core i7-8700K on CPU 0.

## 6.4 Intel Core i9-9900K

The Intel Core i9-9900K has a total of 99 undocumented static MSRs that are zero and 33 that are non-zero, see Table 6.7. In Table 6.8 can be seen that the most common value among the undocumented static non-zero MSRs is the same as the minimum, namely 1. The histogram of the non-zero values can be found in Figure 6.5.

**Table 6.7:** Static MSRs with value zero compared to static MSRs with a non-zero value on the Intel Core i9-9900K.

| CPU | Number of Zero MSRs | Number of Non-Zero MSRs |
|---|---|---|
| 0 | 99 | 33 (0.333%) |
| 1 | 99 | 33 (0.333%) |
| 2 | 99 | 33 (0.333%) |
| 3 | 99 | 33 (0.333%) |
| 4 | 99 | 33 (0.333%) |
| 5 | 99 | 33 (0.333%) |
| 6 | 99 | 33 (0.333%) |
| 7 | 99 | 33 (0.333%) |
| 8 | 99 | 33 (0.333%) |
| 9 | 99 | 33 (0.333%) |
| 10 | 99 | 33 (0.333%) |
| 11 | 99 | 33 (0.333%) |
| 12 | 99 | 33 (0.333%) |
| 13 | 99 | 33 (0.333%) |
| 14 | 99 | 33 (0.333%) |
| 15 | 99 | 33 (0.333%) |

**Table 6.8:** Statistical parameters of static non-zero MSRs on the Intel Core i9-9900K.

| CPU | Minimum | Maximum | Median | Mode |
|---|---|---|---|---|
| 0 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 1 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 2 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 3 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 4 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 5 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 6 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 7 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 8 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 9 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 10 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 11 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 12 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 13 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 14 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |
| 15 | 0000000000000001 | fd0cd1679876a800 | 00000000000187ad | 0000000000000001 (x6) |

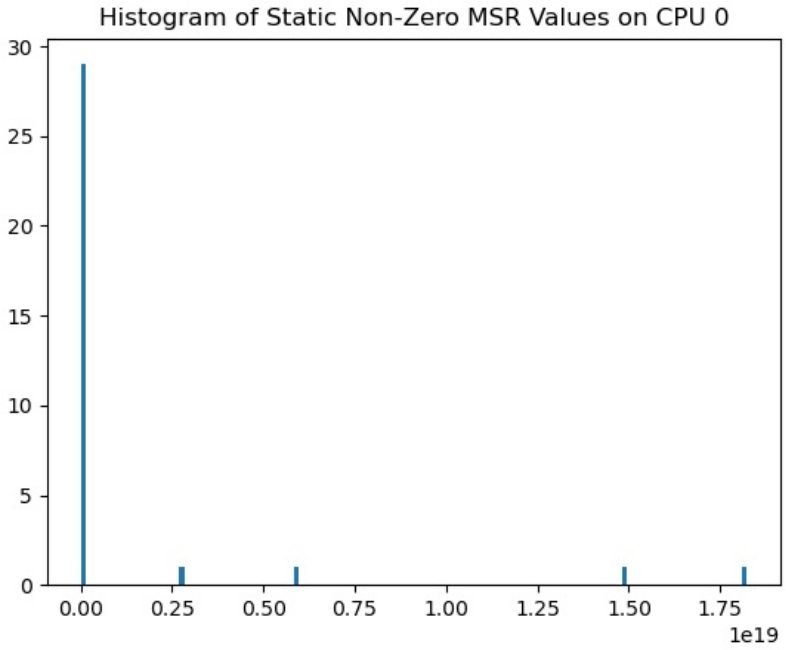



**Figure 6.5:** Histogram of the static non-zero MSR values on the Intel Core i9-9900K on CPU 0.

## 6.5 Intel Xeon Silver 4208

As we can see in Table 6.9 from the 16 CPUs total, 1 CPU has 490 undocumented static MSRs with value zero and 111 non-zero MSRs, 10 CPUs have 488 undocumented static MSRs with value zero and 113 non-zero MSRs, and 5 CPUs have 489 undocumented static MSRs with value zero and 112 non-zero MSRs. Even though there are quite a few irregularities in respect to the amount of zero and non-zero undocumented static MSRs between the different CPUs on the Intel Xeon Silver 4208 the statistical parameters minimum, maximum, median, and mode are the same on all 16 CPUs, see Table 6.10. Figures 6.6, 6.7, and 6.8 are representative histograms of the three different zero to non-zero undocumented static MSR value correlations.

**Table 6.9:** Static MSRs with value zero compared to static MSRs with a non-zero value on the Intel Xeon Silver 4208.

| CPU | Number of Zero MSRs | Number of Non-Zero MSRs |
|---|---|---|
| 0 | 490 | 111 (0.227%) |
| 1 | 488 | 113 (0.232%) |
| 2 | 488 | 113 (0.232%) |
| 3 | 488 | 113 (0.232%) |
| 4 | 488 | 113 (0.232%) |
| 5 | 489 | 112 (0.229%) |
| 6 | 488 | 113 (0.232%) |
| 7 | 488 | 113 (0.232%) |
| 8 | 488 | 113 (0.232%) |
| 9 | 488 | 113 (0.232%) |
| 10 | 489 | 112 (0.229%) |
| 11 | 489 | 112 (0.229%) |
| 12 | 489 | 112 (0.229%) |
| 13 | 489 | 112 (0.229%) |
| 14 | 488 | 113 (0.232%) |
| 15 | 488 | 113 (0.232%) |

**Table 6.10:** Statistical parameters of static non-zero MSRs on the Intel Xeon Silver 4208.

| CPU | Minimum | Maximum | Median | Mode |
|---|---|---|---|---|
| 0 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 1 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 2 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 3 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 4 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 5 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 6 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 7 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 8 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 9 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 10 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 11 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 12 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 13 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 14 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |
| 15 | 0000000000000001 | d7aac184a9664800 | 0000000000030000 | 0000000000030000 (x18) |

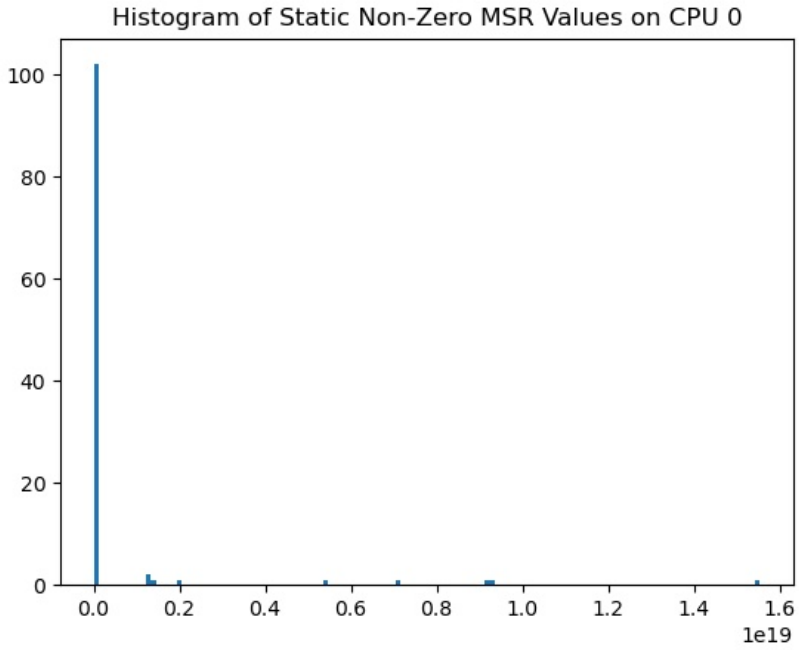



**Figure 6.6:** Histogram of the static non-zero MSR values on the Intel Xeon Silver 4208 on CPU 0.

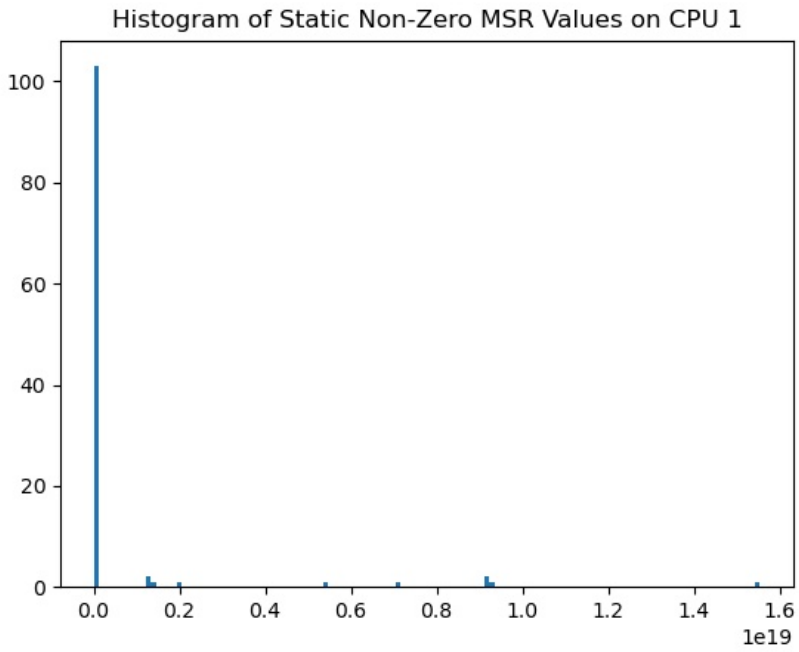



**Figure 6.7:** Histogram of the static non-zero MSR values on the Intel Xeon Silver 4208 on CPU 1.

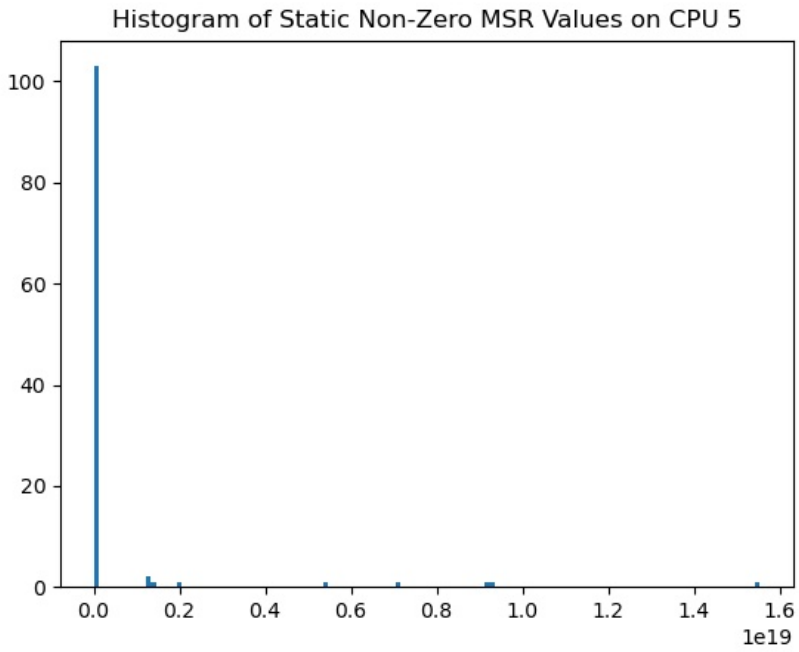



**Figure 6.8:** Histogram of the static non-zero MSR values on the Intel Xeon Silver 4208 on CPU 5.

## 6.6 Findings

The tested Intel Core series CPUs do not have any anomalies in the evaluated statistical parameters. However, the tested Intel Xeon Silver 4208 and the tested AMD Ryzen Threadripper 1920X show certain irregularities when it comes to the amount of zero and non-zero values of their undocumented static MSRs. The AMD Ryzen Threadripper 1920X even shows irregularities in the statistical parameters of it's non-zero undocumented static MSR values. These differences might indicate that the individual CPUs may not have the exact same properties or may even have differences on the transistor layer.

# Chapter 7

# Analysis of Undocumented Dynamic MSRs

In this chapter, we will explore all undocumented MSRs with dynamic values and draw conclusions from the data we gather.

## 7.1 Initial Data Gathering

To find out which MSRs are dynamic, we sample, per CPU, all undocumented MSRs at once for 180 seconds. We then filter for MSRs, which yielded different values during the scan by using the changes function of msrs_analyze. The results can be found in Table 7.1.

Although the AMD Ryzen CPU has the most undocumented MSRs, 4875 in total, it has only two undocumented MSRs with dynamic values, which is a similar number as the tested Intel Core series CPUs implement (2 to 4). It is quite notable that the undocumented dynamic MSRs of the tested Intel Core series CPUs seem to be a subset of the undocumented dynamic MSRs of the Intel Core i7-6700K CPU. By far the most undocumented dynamic MSRs, 42 to be exact, are implemented in the Intel Xeon Silver 4208 server CPU.

**Table 7.1:** Summarized results of the search for dynamic MSRs.

| CPU | Number of Undocumented MSRs | Number of Undocumented Dynamic MSRs | Undocumented Dynamic MSRs |
|---|---|---|---|
| Ryzen Threadripper 1920X [2] | 4875 | 2 (0.0004%) | c0010293<br>c0011083 |
| Core i7-6700K [11] | 104 | 4 (0.0385%) | 4e2<br>1a1<br>621<br>637 |
| Core i7-8700K [12] | 124 | 3 (0.0242%) | 1a1<br>621<br>637 |
| Core i9-9900K [13] | 134 | 2 (0.0149%) | 1a1<br>637 |
| Xeon Silver 4208 [15] | 643 | 42 (0.0653%) | see appendix<br>Section 10.2 |

## 7.2 Cross Correlation

From the addresses of the documented and undocumented dynamic MSRs, we draw conclusions from their values by correlating them to documented dynamic MSRs. For this, we sample all documented and undocumented dynamic MSRs together for 60 seconds. During the sampling, we execute a CPU stress test that spins on a square root function for a few seconds, which will, among others, trigger spikes in electricity and temperature sensor readings, which in turn makes the correlation between such registers more clear.

## 7.3 Intel Core CPUs

As mentioned in Section 7.1, we found that the tested Intel Core processors share addresses of undocumented dynamic MSRs. We found that they seem to have the same functionality, although some have different value offsets. In this section, we analyze them in more detail.

### 7.3.1 A Package Thermal Margin MSR (0x1a1)

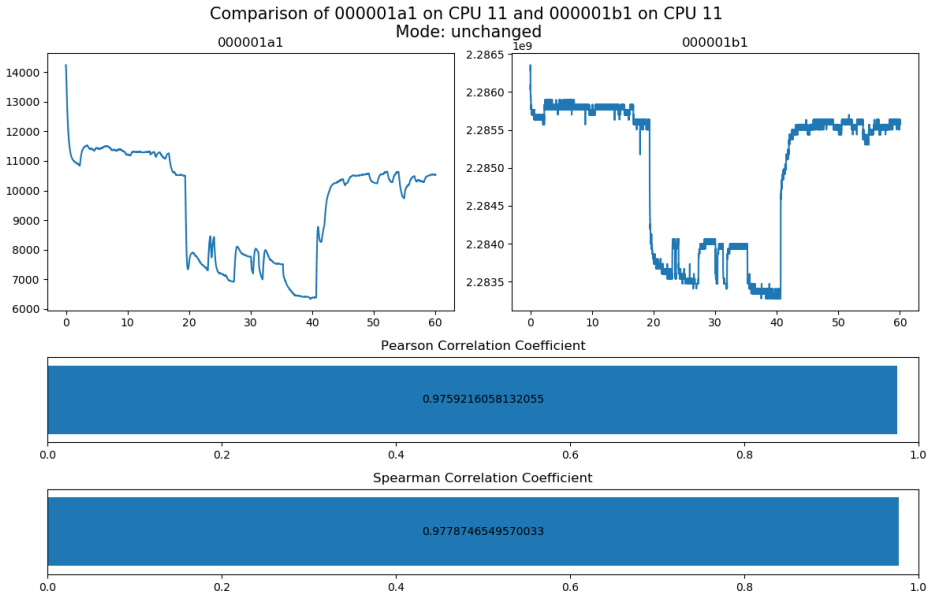



**Figure 7.1:** Correlation of MSR 0x1a1 and the IA32_PACKAGE_THERM_STATUS MSR (0x1b1) on the Intel Core i7-8700K.

The MSR 0x1a1 has an obvious correlation with the IA32_PACKAGE_THERM_STATUS MSR (0x1b1) on all tested CPUs as shown in Figure 7.1 and Appendix Section 10.3.2. Additionally, we observe other, less strong, correlations to thermally, and performance-related MSRs on different Intel Core CPUs such as IA32_PERF_STATUS (0x198) on the Core i7-6700K, and IA32_THERM_STATUS (0x19c) on the Core i7-8700K. We can, therefore, conclude that the 0x1a1 MSR is affected by thermal changes within the CPU. However, it does not seem to be a direct temperature sensor reading as the value decreases when the CPU load increases and the CPU generates more heat. This conclusion matches a statement made in an Intel forum post [34] claiming the 0x1a1 register being the IA32_PACKAGE_THERM_MARGIN MSR.

### 7.3.2 A State Register (0x621)

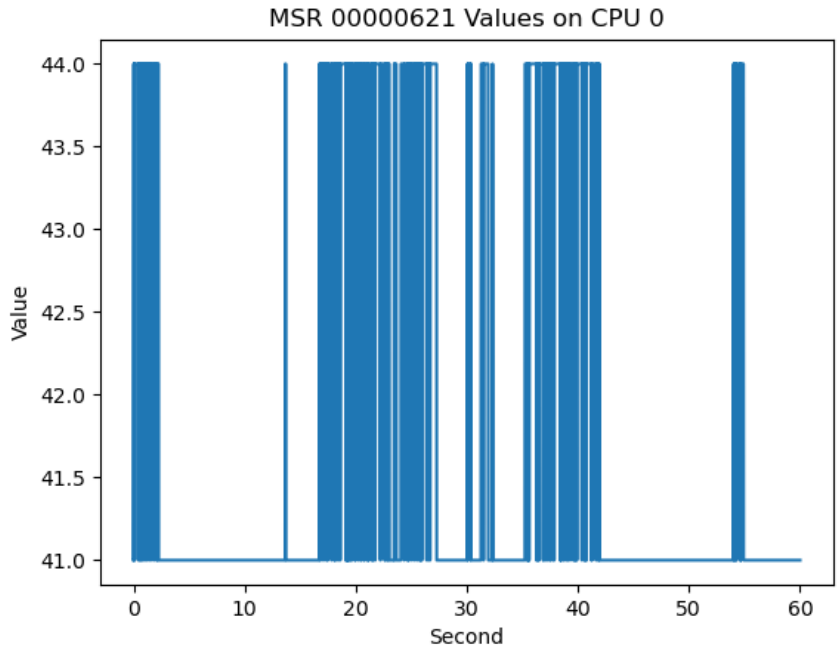



**Figure 7.2:** Values of MSR 0x621 on the Intel Core i7-8700K.

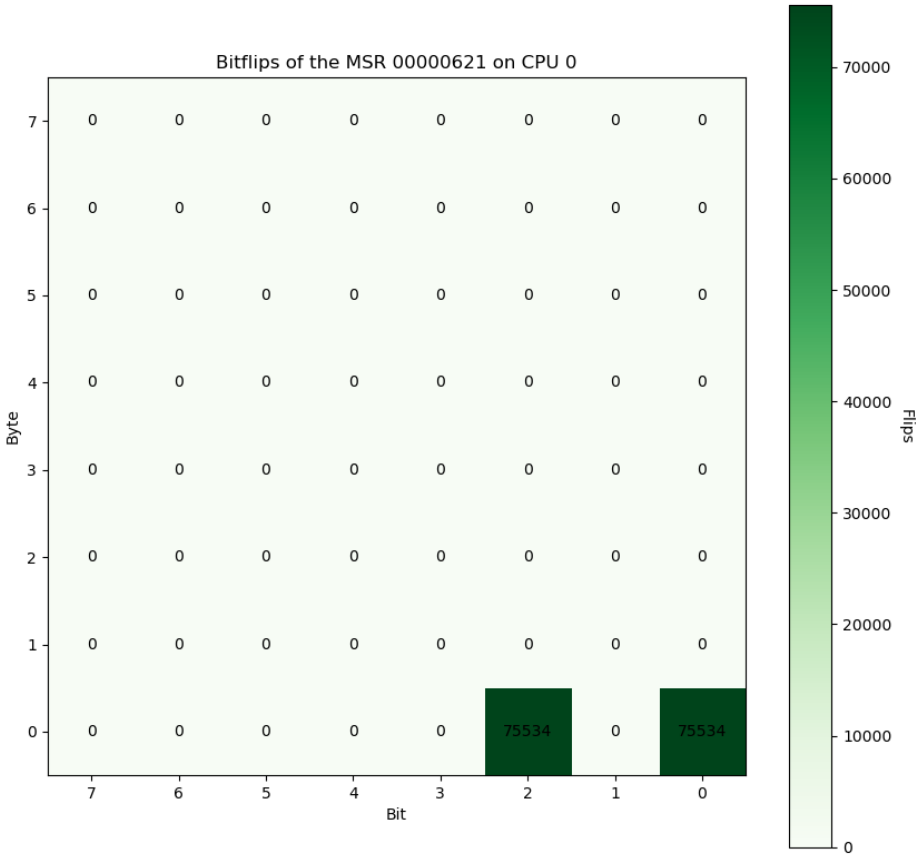



**Figure 7.3:** Bit-flip map of MSR 0x621 on the Intel Core i7-8700K.

MSR 0x621 correlates with two performance counter state registers, but only on Core i7-8700K. The correlation with MSR_RING_PERF_LIMIT_REASONS (0x6b1) is nearly a complete negative. The second correlation with MSR_PERF_STATUS (0x198) is weaker but still above 71% for Pearson and above 82% for Spearman. In Figure 7.2 we can see that the sampled values of MSR 0x621 alternate between 41 and 44, indicating alternating bit flips of bit zero and bit two, see Figure 7.3. This behavior suggests that this register represents at least two states that are mutually exclusive.

### 7.3.3 A RAPL or Performance Counter MSR (0x637)

MSR 0x637 correlates very strongly with the same nine documented MSRs on all Intel Core CPUs: MSR 0x10, MSR 0x659, MSR 0x619, MSR 0x658, MSR 0x611, MSR 0x639, MSR 0xe7, MSR 0xe8, and MSR 0x64e. For a full set of correlations with these MSRs on all tested Intel Core CPUs see Section 10.3.2 in the appendix. On the Core i7-9900K, it

additionally correlates with MSR IA32_FIXED_CTR1 (0x30a), which is a performance counter, see figure 7.5. Out of these nine MSRs, six are performance counters, and three are RAPL (Running Average Power Limit) registers [9]. We can see, for example, in Figure 7.4 that the curve has the same spike as the RAPL registers when the stress test begins and thus power consumption increases. For this reason, we think that this register is a RAPL register instead of a performance counter. However, we do not have any evidence of this register not being a performance counter.

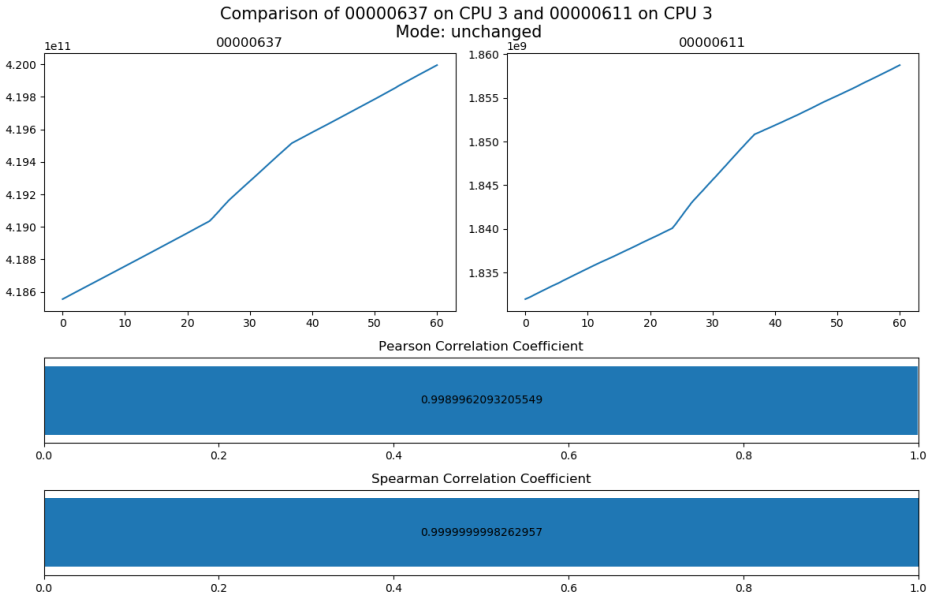



**Figure 7.4:** Correlation of MSR 0x637 and MSR_PKG_ENERGY_STATUS (0x611) on the Intel Core i7-6700K.

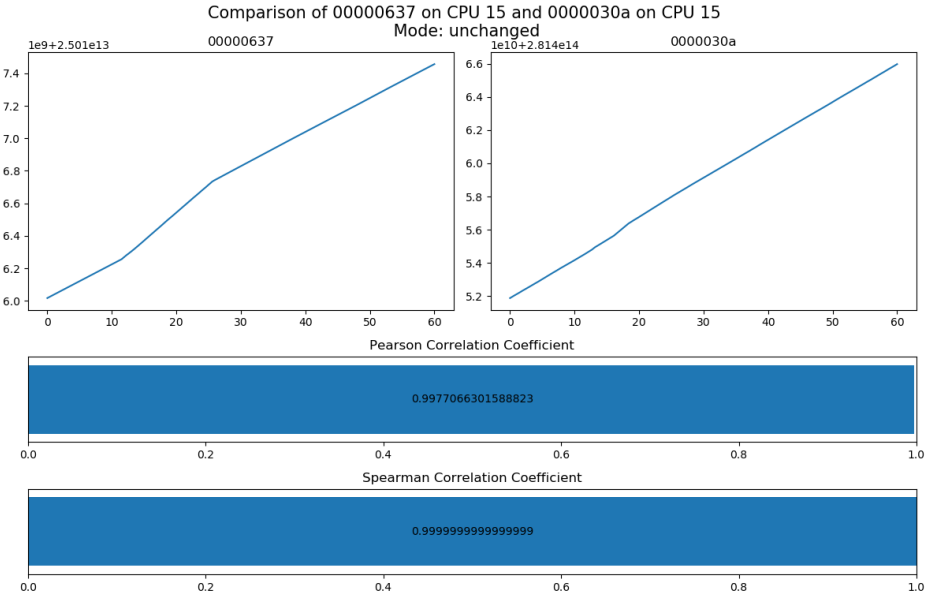



**Figure 7.5:** Correlation of MSR 0x637 and MSR IA32_FIXED_CTR1 (0x30a).

### 7.3.4 A System Management Mode MSR (0x4e2)?

On the Intel Core i7-6700K, we found a dynamic MSR that has no notable correlation to other MSRs. As of the Intel manual [9], the family of this particular CPU, specifically with the DisplayFamily_DisplayModel signatures of 06_5EH, has its documented MSRs, besides the architectural MSRs, listed in Tables 2-20, 2-21, 2-25, 2-29, 2-35, 2-39, and 2-40 [9]. The only occurrence of 0x4e2, however, is in Tables 2-12 [9] and 2-30 [9], thus it is only documented for other microarchitectures. It is also mentioned in Section 34.17.2 SMI Delivery Delay Reporting of the Intel manual [9]. In these occurrences, the register is described as being a state indicator if the servicing of a system management interrupt is delayed. It is further noted that this register may only be queried in system management mode, and attempts to query it outside of system management mode will result in a general protection fault.

In the Plots 7.6, 7.7, and 7.8 we see that we have bit flips of the lowest four bits which correspond to the logical cores of the system, as hyper-threading was disabled in the BIOS during the sampling. This behavior matches the register's description for other microarchitectures, except that it is accessible from kernelspace. To verify that this undocumented MSR is indeed the MSR_SMM_DELAYED described in the Intel manual [9], further experiments are needed that would be beyond the scope of this thesis.

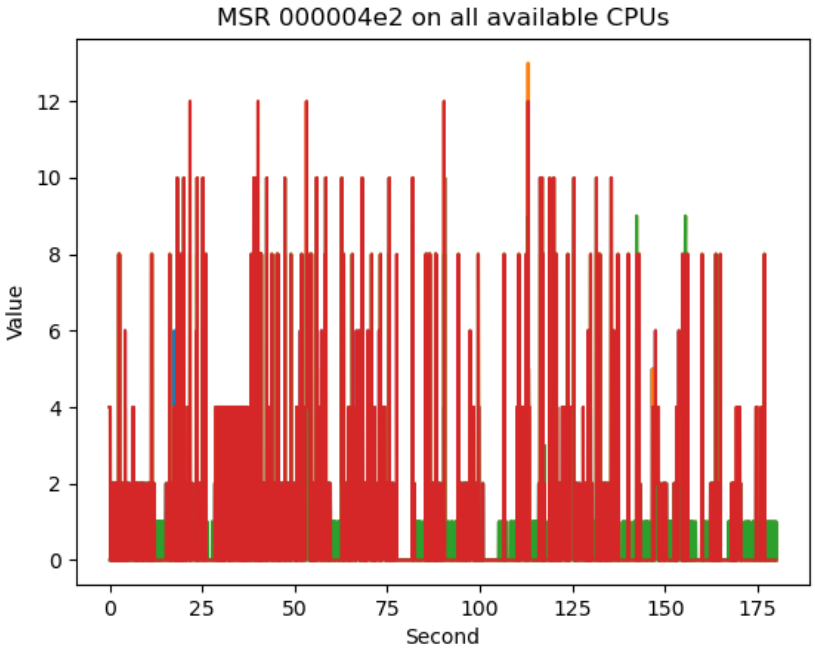



**Figure 7.6:** Values of MSR 0x4e2 on the Intel Core i7-6700K. Please note that at the time of sampling hyper-threading was deactivated in BIOS and thus this experiment was conducted only on the 4 physical cores of this CPU.

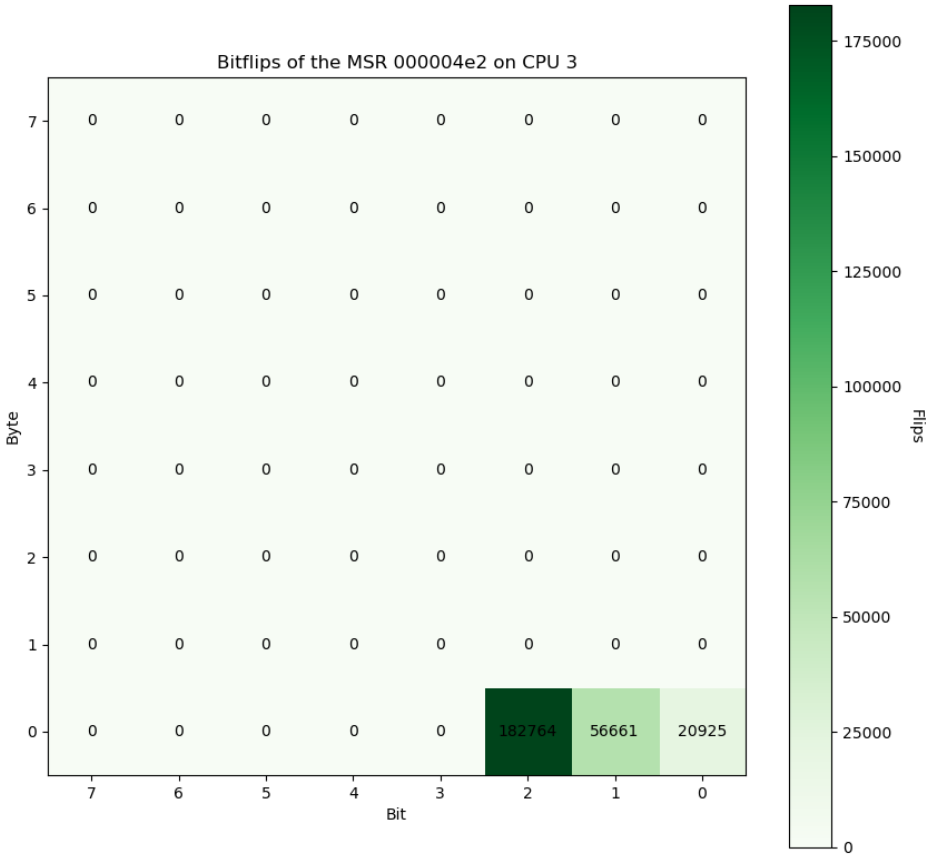



**Figure 7.7:** Bit-flip map of MSR 0x64e2 on the Intel Core i7-6700K on CPU 3. Please note that at the time of sampling hyper-threading was deactivated in BIOS and thus this experiment was conducted only on the 4 physical cores of this CPU.

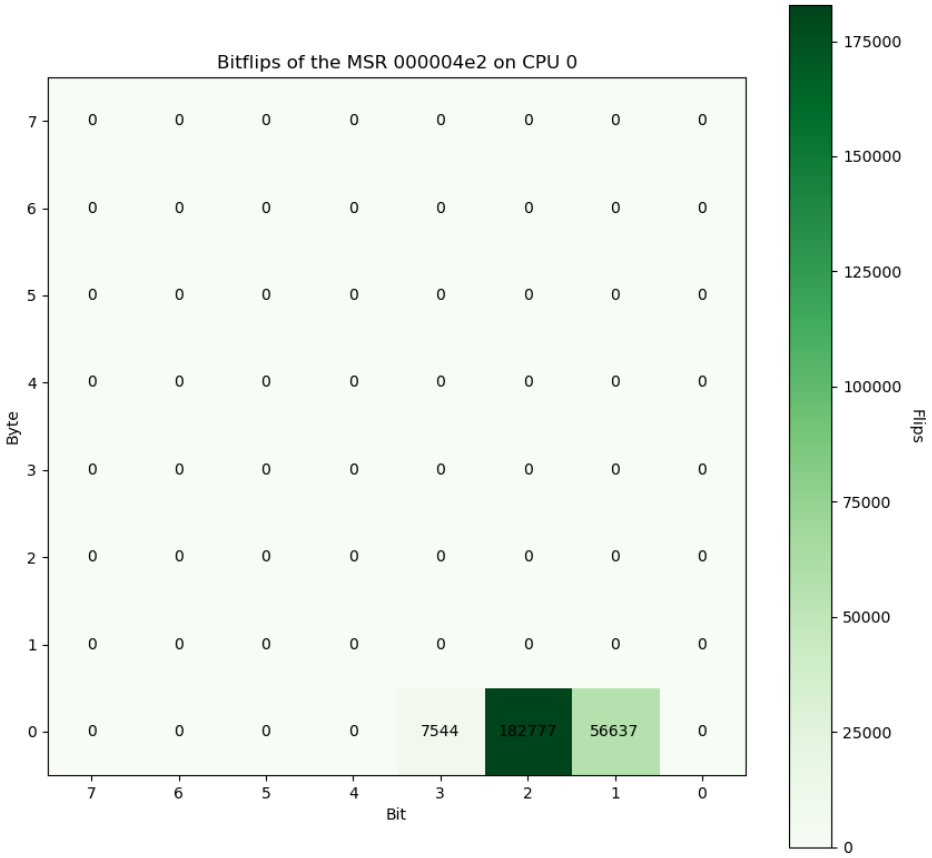



**Figure 7.8:** Bit-flip map of MSR 0x64e2 on the Intel Core i7-6700K on CPU 0. Please note that at the time of sampling hyper-threading was deactivated in BIOS and thus this experiment was conducted only on the 4 physical cores of this CPU.

## 7.4 Intel Xeon Silver 4208

As can be seen in Table 7.1, our Xeon CPU, the Xeon Silver 4208, has 42 undocumented dynamic MSRs which is, compared to the other tested CPUs, by far the most. When looking at its registers, we can see that it shares some undocumented dynamic MSRs with the Intel Core CPUs. Specifically 0x1a1, and 0x637. These MSRs share the same properties, which means the assumed classification of these MSRs from Section 7.3 holds for the Intel Xeon Silver 4208 as well. In this section, we will examine these registers in more detail.

### 7.4.1 Another Thermally Related MSR (0x1a3)

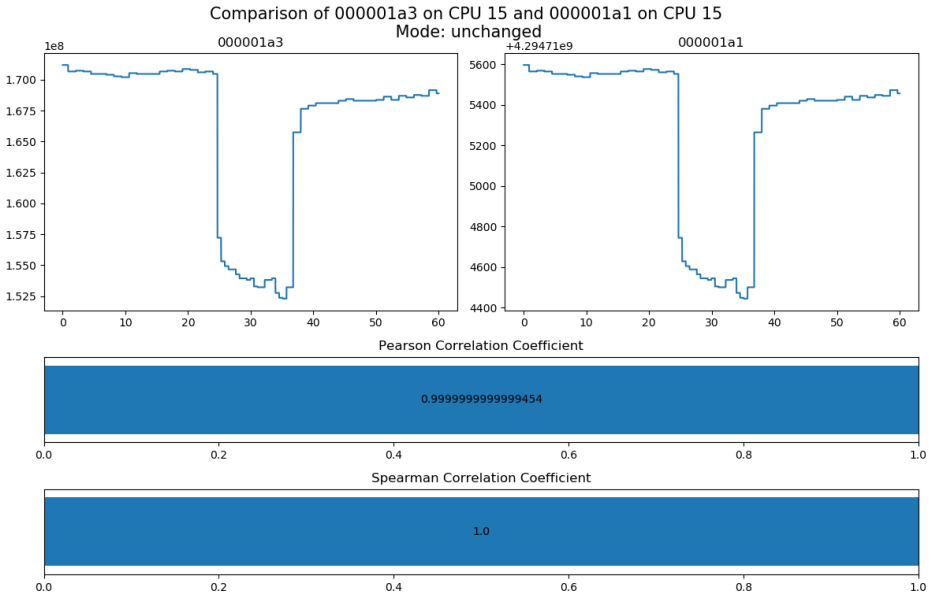



**Figure 7.9:** Correlation of MSR 0x1a3 and 0x1a1.

Interestingly, we found an undocumented dynamic MSR, 0x1a3 to be specific, that yielded the exact same output curve as the register 0x1a1, see Figure 7.9. The difference only being the offset as well as the scaling. From the plot, we can further deduce that MSR 0x1a3 reads the data from the same sensor as MSR 0x1a1 does but transforms them in a specific way.

### 7.4.2 40 RAPL or Performance Counter MSRs

The remaining 40 dynamic undocumented MSRs, including 0x637, all correlate with each other. Except for 0xb05, the correlations are nearly complete for both Spearman as well as Pearson, see Figure 7.10 for an example, which means that the conclusions we draw for 0x637 in Section 7.3 still hold for these 40 registers.

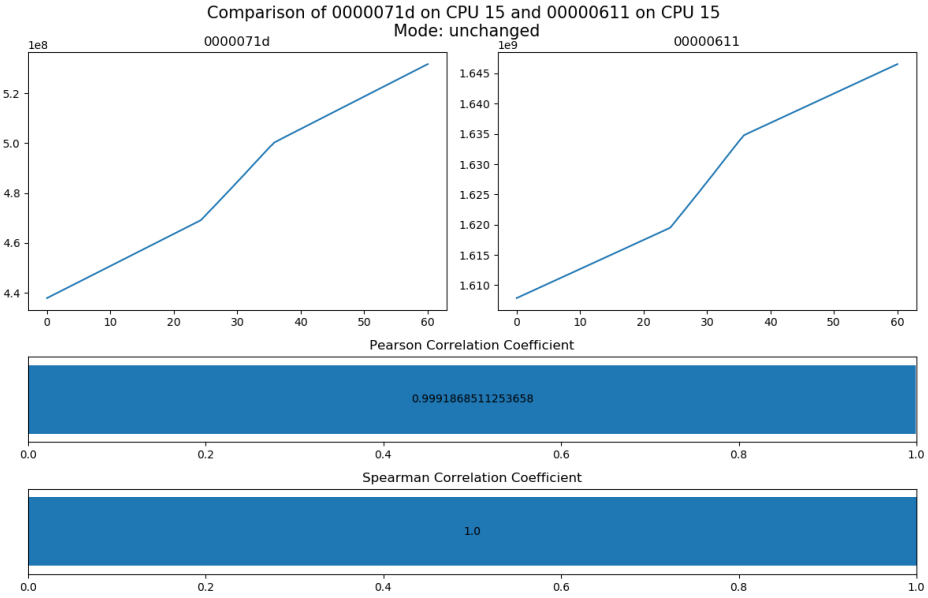



**Figure 7.10:** Correlation of MSR 0x71d and 0x611.

As the correlation between 0xb05 and 0x637 is still very strong, see Figure 7.11, with $r > 0.977$ for Pearson and $r > 0.999$ for Spearman, we conclude that 0xb05 is also a RAPL or Performance Counter MSR as 0x637.

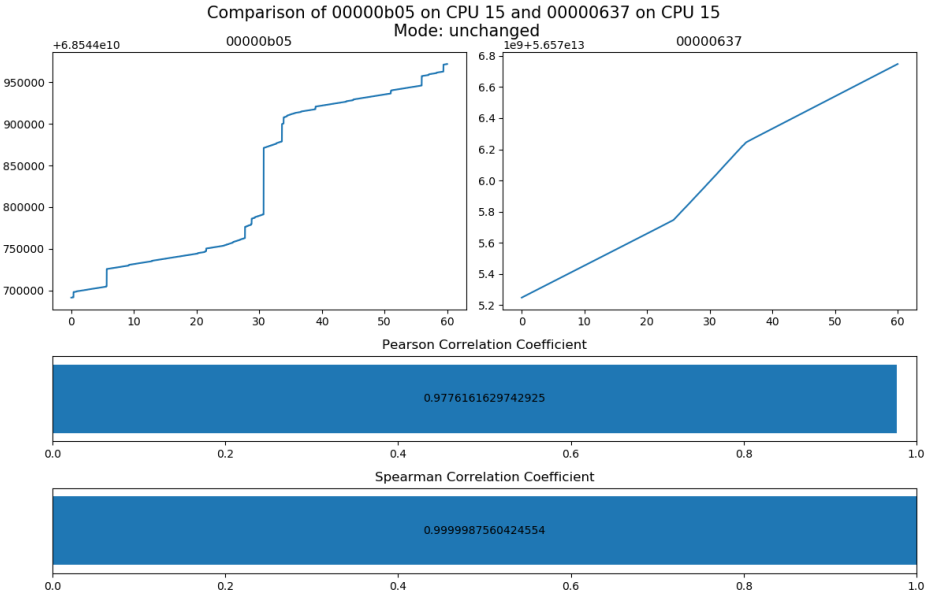



**Figure 7.11:** Correlation of MSR 0xb05 and 0x637.

A full set of relevant, i.e. ($r \geq 0.75$), correlations of MSR 0x637, which includes all other 39 MSRs discussed in this subsection, can be found in the appendix in Section 10.3.1.

## 7.5 AMD Ryzen 7 CPU

On our Ryzen 7 CPU, we found only two undocumented dynamic MSRs that do not correlate with any documented MSR, namely 0xc0010293 and 0xc0011083. For more information on these MSRs see Figures 7.12, 7.13, 7.14, and 7.15 respectively.

Further research, that would be beyond the scope of this thesis, is needed to reverse engineer these registers.

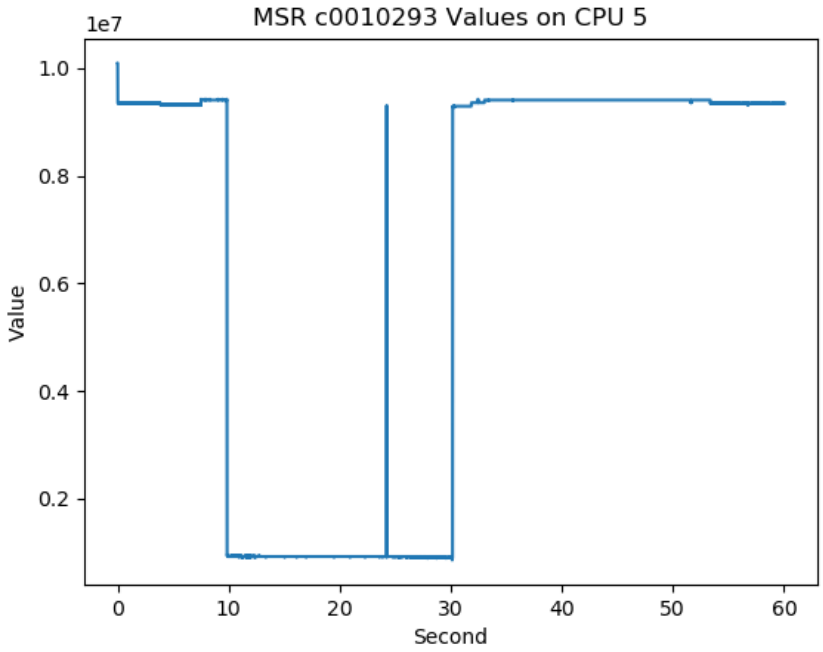



**Figure 7.12:** Value curve of MSR 0xc0010293 on the AMD Ryzen Threadripper 1920X on CPU 5.

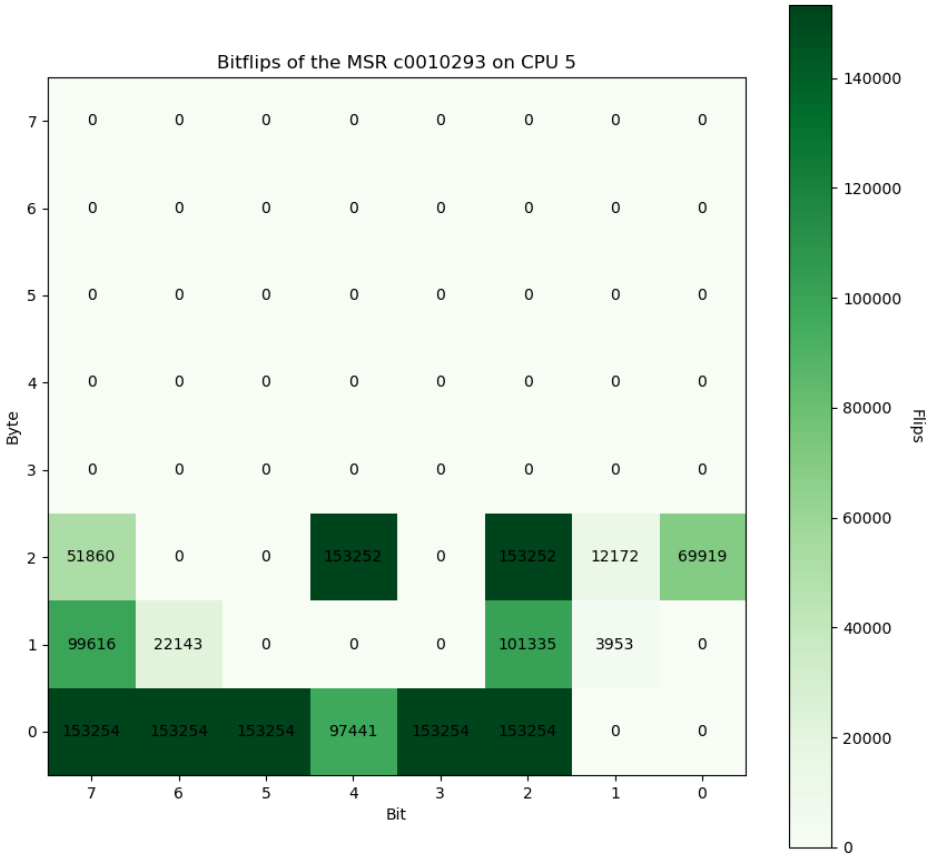



**Figure 7.13:** Bit-flip map of MSR 0xc0010293 on the AMD Ryzen Threadripper 1920X on CPU 5.

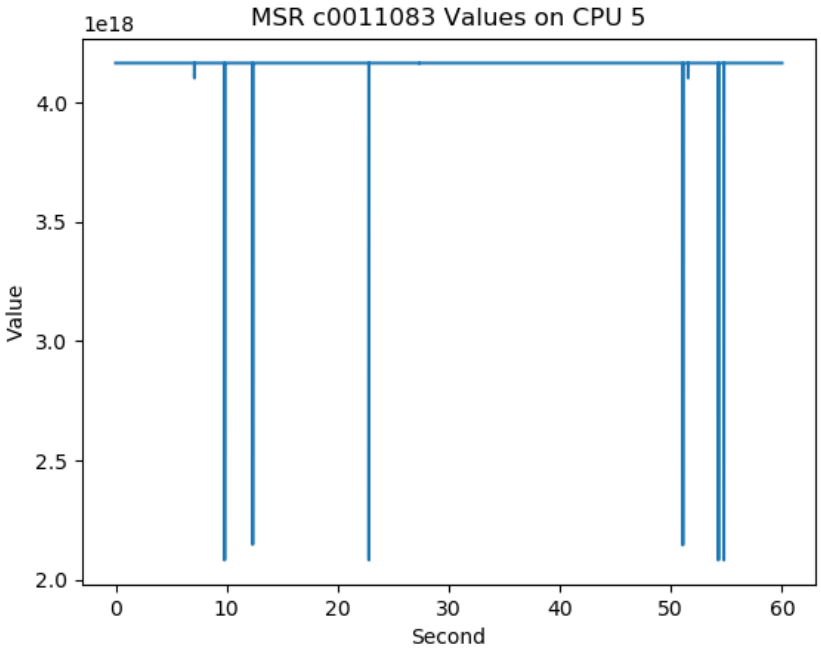



**Figure 7.14:** Value curve of MSR 0xc0011083 on the AMD Ryzen Threadripper 1920X on CPU 5.

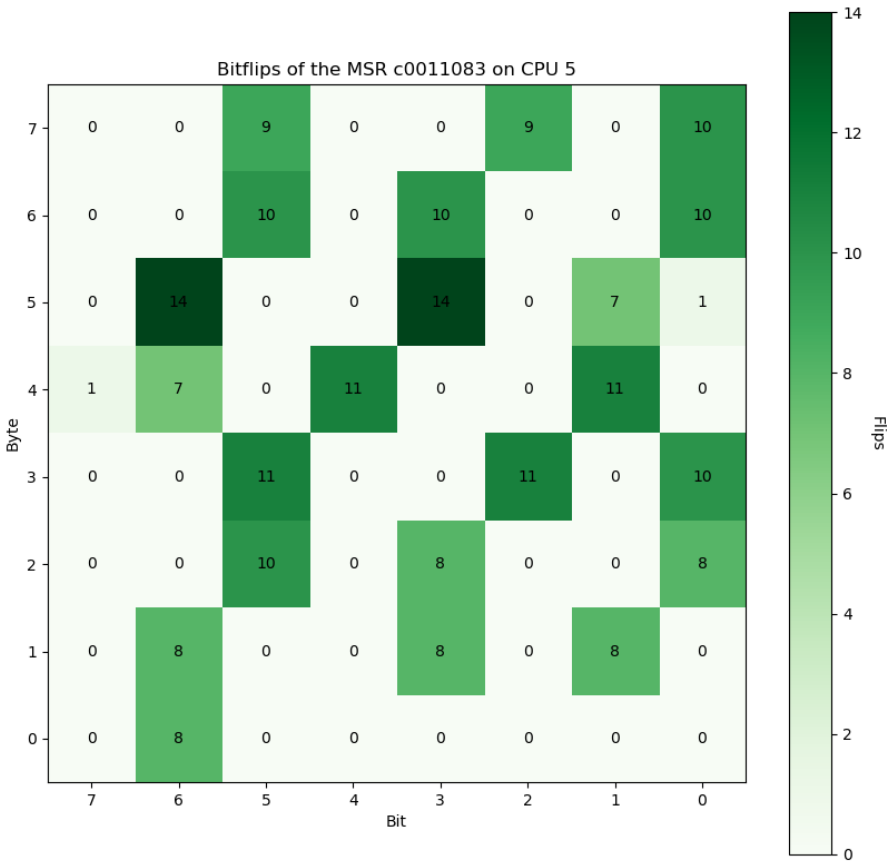



**Figure 7.15:** Bit-flip map of MSR 0xc0011083 on the AMD Ryzen Threadripper 1920X on CPU 5.

# Chapter 8

# Future Work

Simplified, MSRs can be used to read values, for example, of a sensor or states, enable or disable CPU features via individual bits, or configure CPU features. MSRs may even combine multiple categories. For example, they can have a 63 bit counter of some sort and one bit that can be used to reset the counter. In this thesis, we covered a part of the **read values** category, specifically the part of undocumented dynamic MSRs, which means that there is a wide range of unresearched categories of MSRs that we discuss in detail below.

## 8.1 Nomenclature

To make things easier, we decided to name the found categories described above.

- **Value-MSRs** these are MSRs that hold values. These values can either be dynamic or static. They can describe various things, such as readings or initialization values.
- **Flag-MSRs** these are MSRs that contain one or more bits that enable or disable CPU features.
- **Hybrid-MSRs** these are MSRs that combine Value-MSRs and Flag-MSRs

## 8.2 Value-MSRs

We found 53 undocumented dynamic MSRs on the CPUs we conducted our experiments on and determined their correlation and thus possible purpose. However, further research is needed to reverse-engineer what exactly these registers are there for.

Research also has to be conducted in the direction of static Value-MSRs.

## 8.3 Flag-MSRs

Flag-MSRs can be quite interesting for researchers working with microarchitectures as they might enable or disable certain CPU features that allow for faster, more efficient, more accurate, or even new research. As an example, there may be bits that disable certain pre-fetchers or security checks. Such bits may be hidden within the undocumented/reserved bits of documented MSRs or in entirely undocumented MSRs. A documented example is bit 9 of MSR 0x1a0 that can be used to disable the hardware prefetcher. This MSR is documented in Table 2-2 in Chapter 2 of Volume 4 of the Intel manual [9].

## 8.4 Hybrid-MSRs

Future research should note that MSRs may not strictly be Flag-MSRs or Value-MSRs but may very well be Hybrid-MSRs, which are a mixture of the two.

# Chapter 9

# Conclusion

In this thesis, we successfully developed a software toolset laying the foundation for future undocumented MSR based research. Furthermore, we conducted first experiments, foundational data gathering, and present techniques on how to categorize undocumented dynamic MSRs via cross-correlation. To summarize, we found 5880 undocumented MSRs with a share of 53 dynamic MSRs. We were able to categorize 43 dynamic MSRs and draw conclusions for them as to what their data tells us. On the AMD Ryzen Threadripper 1920X and the Intel Xeon Silver 4208 we found irregularities in the examined statistical parameters of the undocumented static MSRs that are the base for our hypothesis that there may be physical differences between the individual CPUs. As a last point, we discussed possible future work based on the findings and results of this thesis.

# Bibliography


[1] AMD. Open-Source Register Reference For AMD Family 17h Processors Models 00h-2Fh. https://www.amd.com/system/files/TechDocs/56255_OSRR.pdf, July 2018.

[2] AMD. AMD Ryzen™ Threadripper™ 1920X Prozessor. https://www.amd.com/de/products/cpu/amd-ryzen-threadripper-1920x, visited 2020-02-24.

[3] Chok, N. S. Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data. Master's thesis, University of Pittsburgh, 2010.

[4] Corbet, J. The Big Kernel Lock lives on. https://lwn.net/Articles/86859/, 2004-05-26.

[5] Corbet, J. The new way of ioctl(). https://lwn.net/Articles/119652/, 2005-01-18.

[6] Domas, C. Hardware backdoors in x86 cpus. *At Black Hat USA* (2018).

[7] Durand, P. Open-Source Register Reference For AMD Family 17h Processors Models 00h-2Fh. http://www.cs.kent.edu/~durand/CS0/Notes/Chapter05/isa.html, visited 2020-05-13.

[8] Heo, T., and Mickler, F. Concurrency Managed Workqueue *cmwq*. https://www.kernel.org/doc/html/v4.15/core-api/workqueue.html, September 2010.

[9] Intel. Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. https://software.intel.com/sites/default/files/managed/39/c5/325462-sdm-vol-1-2abcd-3abcd.pdf, May 2019.

[10] Intel. Intel® Core™ i5-8265U Prozessor. https://ark.intel.com/content/www/de/de/ark/products/149088/intel-core-i5-8265u-processor-6m-cache-up-to-3-90-ghz.html, visited 2020-02-20.

[11] Intel. Intel® Core™ i7-6700K Prozessor. https://ark.intel.com/content/www/de/de/ark/products/88195/

intel-core-i7-6700k-processor-8m-cache-up-to-4-20-ghz.html, visited 2020-02-20.

[12] Intel. Intel® Core™ i7-8700K Prozessor. https://ark.intel.com/content/www/de/de/ark/products/126684/intel-core-i7-8700k-processor-12m-cache-up-to-4-70-ghz.html, visited 2020-02-20.

[13] Intel. Intel® Core™ i9-9900K Prozessor. https://ark.intel.com/content/www/de/de/ark/products/186605/intel-core-i9-9900k-processor-16m-cache-up-to-5-00-ghz.html, visited 2020-02-20.

[14] Intel. Intel® Xeon® Prozessor E5-1630 v4. https://ark.intel.com/content/www/de/de/ark/products/92987/intel-xeon-processor-e5-1630-v4-10m-cache-3-70-ghz.html, visited 2020-02-20.

[15] Intel. Intel® Xeon® Silver Prozessor 4208. https://ark.intel.com/content/www/de/de/ark/products/193390/intel-xeon-silver-4208-processor-11m-cache-2-10-ghz.html, visited 2020-02-20.

[16] Khawaja, O. try/catch in Linux Kernel. https://binarydebt.wordpress.com/2018/11/16/try-catch-in-linux-kernel/.

[17] Kroah-Hartman, G. The Linux Kernel Driver Interface. https://elixir.bootlin.com/linux/v5.3/source/Documentation/process/stable-api-nonsense.rst, 2019-10-24.

[18] man-pages project, T. L. IOCTL(2). http://man7.org/linux/man-pages/man2/ioctl.2.html, 2017-05-03.

[19] NumPy developers. The Official NumPy Website. https://numpy.org/, visited 2020-01-22.

[20] Python Software Foundation. The Python Language Reference: Data model. https://docs.python.org/3/reference/datamodel.html#objects-values-and-types, visited 2020-01-22.

[21] Rubini, A. Miscellaneous Character Drivers. https://www.linuxjournal.com/article/2920, 1998-06-30.

[22] Russell, R. Unreliable Guide To Hacking The Linux Kernel. https://www.kernel.org/doc/html/v5.3/kernel-hacking/hacking.html.

[23] S.A, S. Delivering code in production with debug features activated is security-sensitive. https://rules.sonarsource.com/java/tag/owasp/RSPEC-4507, visited 2020-07-23.

[24] SciPy developers. The Official SciPy Website. https://www.scipy.org/, visited 2020-01-22.

[25] The Linux man-pages project. MSR(4). https://man7.org/linux/man-pages/man4/msr.4.html, 2009-03-31.

[26] The Linux man-pages project. PREAD(2). http://man7.org/linux/man-pages/man2/pread.2.html, 2017-09-15.

[27] The Linux man-pages project. OPEN(2). http://man7.org/linux/man-pages/man2/open.2.html, 2018-04-30.

[28] The Linux man-pages project. CLOSE(2). http://man7.org/linux/man-pages/man2/close.2.html, 2019-10-10.

[29] The Matplotlib development team. The Official Matplotlib Website. https://matplotlib.org/, visited 2020-01-22.

[30] Torvalds, L. The Linux Kernel errno-base.h. https://elixir.bootlin.com/linux/v5.3/source/include/uapi/asm-generic/errno-base.h.

[31] Torvalds, L. The Linux Kernel Source Version 5.3. https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tag/?h=v5.3.

[32] Torvalds, L. The Linux Kernel Syscall Table. https://elixir.bootlin.com/linux/v5.3/source/arch/x86/entry/syscalls/syscall_64.tbl.

[33] Xu, H., and Deng, Y. Dependent evidence combination based on shearman coefficient and pearson coefficient. *IEEE Access 6* (2017), 11634–11640.

[34] Z, K. How to determine cause of processor frequency scale down to 200 MHz due to ThermStatus. https://software.intel.com/en-us/forums/software-tuning-performance-optimization-platform-monitoring/topic/815194, 2019-03-09T07:39.

# Chapter 10

# Appendix

## 10.1 Found Undocumented MSRs

### 10.1.1 AMD Ryzen Threadripper 1920X

| | | | | | |
|---|---|---|---|---|---|
| 401 | 402 | 410 | 411 | 424 | 42b |
| 420 | 421 | 422 | 423 | 42a | 431 |
| 426 | 427 | 428 | 429 | 430 | 437 |
| 42c | 42d | 42e | 42f | 436 | 441 |
| 432 | 433 | 434 | 435 | 440 | 447 |
| 438 | 439 | 43a | 43b | 446 | 455 |
| 442 | 443 | 444 | 445 | 454 | 45f |
| 448 | 449 | 44a | 44b | 45e | 465 |
| 456 | 457 | 45c | 45d | 464 | 46b |
| 460 | 461 | 462 | 463 | 46a | 471 |
| 466 | 467 | 468 | 469 | 470 | 477 |
| 46c | 46d | 46e | 46f | 476 | 47d |
| 472 | 473 | 474 | 475 | 47c | c000040a |
| 478 | 479 | 47a | 47b | c0000409 | c0002007 |
| 47e | 47f | da0 | c0000408 | c000040f | c000200f |
| c000040b | c000040c | c000040d | c000040e | c000200e | c000201c |
| c000200a | c000200b | c000200c | c000200d | c000201b | c000202b |
| c0002017 | c0002018 | c0002019 | c000201a | c000202a | c0002038 |
| c000201d | c000201e | c000201f | c0002027 | c0002037 | c000203e |
| c000202c | c000202d | c000202e | c000202f | c000203d | c0002044 |
| c0002039 | c000203a | c000203b | c000203c | c0002043 | c000204a |
| c000203f | c0002040 | c0002041 | c0002042 | c0002049 | c0002057 |
| c0002045 | c0002046 | c0002047 | c0002048 | c000204f | c000205d |
| c000204b | c000204c | c000204d | c000204e | c000205c | c000206a |
| c0002058 | c0002059 | c000205a | c000205b | c0002069 | c0002070 |

| | | | | | |
|---|---|---|---|---|---|
| c000205e | c000205f | c0002067 | c0002068 | c000206f | c000207b |
| c000206b | c000206c | c000206d | c000206e | c000207a | c0002081 |
| c0002071 | c0002072 | c0002073 | c0002077 | c0002080 | c000208c |
| c000207c | c000207d | c000207e | c000207f | c000208b | c0002092 |
| c0002082 | c0002083 | c0002087 | c000208a | c0002091 | c000209d |
| c000208d | c000208e | c000208f | c0002090 | c000209c | c00020a3 |
| c0002093 | c0002097 | c000209a | c000209b | c00020a2 | c00020ae |
| c000209e | c000209f | c00020a0 | c00020a1 | c00020ad | c00020b7 |
| c00020a7 | c00020aa | c00020ab | c00020ac | c00020b3 | c00020bf |
| c00020af | c00020b0 | c00020b1 | c00020b2 | c00020be | c00020ca |
| c00020ba | c00020bb | c00020bc | c00020bd | c00020c7 | c00020d0 |
| c00020c0 | c00020c1 | c00020c2 | c00020c3 | c00020cf | c00020db |
| c00020cb | c00020cc | c00020cd | c00020ce | c00020da | c00020ea |
| c00020d1 | c00020d2 | c00020d3 | c00020d7 | c00020e7 | c00020f0 |
| c00020dc | c00020dd | c00020de | c00020df | c00020ef | c00020fc |
| c00020eb | c00020ec | c00020ed | c00020ee | c00020fb | c000210c |
| c00020f1 | c00020f2 | c00020f3 | c00020f7 | c000210b | c0002112 |
| c00020fd | c00020fe | c00020ff | c0002107 | c0002111 | c0002118 |
| c000210d | c000210e | c000210f | c0002110 | c0002117 | c000211e |
| c0002113 | c0002114 | c0002115 | c0002116 | c000211d | c0002124 |
| c0002119 | c000211a | c000211b | c000211c | c0002123 | c000212a |
| c000211f | c0002120 | c0002121 | c0002122 | c0002129 | c0002137 |
| c0002125 | c0002126 | c0002127 | c0002128 | c000212f | c000213d |
| c000212b | c000212c | c000212d | c000212e | c000213c | c0002143 |
| c0002138 | c0002139 | c000213a | c000213b | c0002142 | c000214e |
| c000213e | c000213f | c0002140 | c0002141 | c000214d | c000215d |
| c0002147 | c000214a | c000214b | c000214c | c000215c | c000216c |
| c000214f | c0002157 | c000215a | c000215b | c000216b | c0002172 |
| c000215e | c000215f | c0002167 | c000216a | c0002171 | c0002178 |
| c000216d | c000216e | c000216f | c0002170 | c0002177 | c000217e |
| c0002173 | c0002174 | c0002175 | c0002176 | c000217d | c0002184 |
| c0002179 | c000217a | c000217b | c000217c | c0002183 | c000218a |
| c000217f | c0002180 | c0002181 | c0002182 | c0002189 | c0002190 |
| c0002185 | c0002186 | c0002187 | c0002188 | c000218f | c0002196 |
| c000218b | c000218c | c000218d | c000218e | c0002195 | c000219c |
| c0002191 | c0002192 | c0002193 | c0002194 | c000219b | c00021a2 |
| c0002197 | c0002198 | c0002199 | c000219a | c00021a1 | c00021a8 |
| c000219d | c000219e | c000219f | c00021a0 | c00021a7 | c00021ae |
| c00021a3 | c00021a4 | c00021a5 | c00021a6 | c00021ad | c00021b4 |
| c00021a9 | c00021aa | c00021ab | c00021ac | c00021b3 | c00021ba |
| c00021af | c00021b0 | c00021b1 | c00021b2 | c00021b9 | c00021c0 |
| c00021b5 | c00021b6 | c00021b7 | c00021b8 | c00021bf | c00021c6 |
| c00021bb | c00021bc | c00021bd | c00021be | c00021c5 | c00021cc |

| | | | | | |
|---|---|---|---|---|---|
| c00021c1 | c00021c2 | c00021c3 | c00021c4 | c00021cb | c00021d2 |
| c00021c7 | c00021c8 | c00021c9 | c00021ca | c00021d1 | c00021d8 |
| c00021cd | c00021ce | c00021cf | c00021d0 | c00021d7 | c00021de |
| c00021d3 | c00021d4 | c00021d5 | c00021d6 | c00021dd | c00021e4 |
| c00021d9 | c00021da | c00021db | c00021dc | c00021e3 | c00021ea |
| c00021df | c00021e0 | c00021e1 | c00021e2 | c00021e9 | c00021f0 |
| c00021e5 | c00021e6 | c00021e7 | c00021e8 | c00021ef | c00021f6 |
| c00021eb | c00021ec | c00021ed | c00021ee | c00021f5 | c00021fc |
| c00021f1 | c00021f2 | c00021f3 | c00021f4 | c00021fb | c0002202 |
| c00021f7 | c00021f8 | c00021f9 | c00021fa | c0002201 | c0002208 |
| c00021fd | c00021fe | c00021ff | c0002200 | c0002207 | c000220e |
| c0002203 | c0002204 | c0002205 | c0002206 | c000220d | c0002214 |
| c0002209 | c000220a | c000220b | c000220c | c0002213 | c000221a |
| c000220f | c0002210 | c0002211 | c0002212 | c0002219 | c0002220 |
| c0002215 | c0002216 | c0002217 | c0002218 | c000221f | c0002226 |
| c000221b | c000221c | c000221d | c000221e | c0002225 | c000222c |
| c0002221 | c0002222 | c0002223 | c0002224 | c000222b | c0002232 |
| c0002227 | c0002228 | c0002229 | c000222a | c0002231 | c0002238 |
| c000222d | c000222e | c000222f | c0002230 | c0002237 | c000223e |
| c0002233 | c0002234 | c0002235 | c0002236 | c000223d | c0002244 |
| c0002239 | c000223a | c000223b | c000223c | c0002243 | c000224a |
| c000223f | c0002240 | c0002241 | c0002242 | c0002249 | c0002250 |
| c0002245 | c0002246 | c0002247 | c0002248 | c000224f | c0002256 |
| c000224b | c000224c | c000224d | c000224e | c0002255 | c000225c |
| c0002251 | c0002252 | c0002253 | c0002254 | c000225b | c0002262 |
| c0002257 | c0002258 | c0002259 | c000225a | c0002261 | c0002268 |
| c000225d | c000225e | c000225f | c0002260 | c0002267 | c000226e |
| c0002263 | c0002264 | c0002265 | c0002266 | c000226d | c0002274 |
| c0002269 | c000226a | c000226b | c000226c | c0002273 | c000227a |
| c000226f | c0002270 | c0002271 | c0002272 | c0002279 | c0002280 |
| c0002275 | c0002276 | c0002277 | c0002278 | c000227f | c0002286 |
| c000227b | c000227c | c000227d | c000227e | c0002285 | c000228c |
| c0002281 | c0002282 | c0002283 | c0002284 | c000228b | c0002292 |
| c0002287 | c0002288 | c0002289 | c000228a | c0002291 | c0002298 |
| c000228d | c000228e | c000228f | c0002290 | c0002297 | c000229e |
| c0002293 | c0002294 | c0002295 | c0002296 | c000229d | c00022a4 |
| c0002299 | c000229a | c000229b | c000229c | c00022a3 | c00022aa |
| c000229f | c00022a0 | c00022a1 | c00022a2 | c00022a9 | c00022b0 |
| c00022a5 | c00022a6 | c00022a7 | c00022a8 | c00022af | c00022b6 |
| c00022ab | c00022ac | c00022ad | c00022ae | c00022b5 | c00022bc |
| c00022b1 | c00022b2 | c00022b3 | c00022b4 | c00022bb | c00022c2 |
| c00022b7 | c00022b8 | c00022b9 | c00022ba | c00022c1 | c00022c8 |
| c00022bd | c00022be | c00022bf | c00022c0 | c00022c7 | c00022ce |

| | | | | | |
|---|---|---|---|---|---|
| c00022c3 | c00022c4 | c00022c5 | c00022c6 | c00022cd | c00022d4 |
| c00022c9 | c00022ca | c00022cb | c00022cc | c00022d3 | c00022da |
| c00022cf | c00022d0 | c00022d1 | c00022d2 | c00022d9 | c00022e0 |
| c00022d5 | c00022d6 | c00022d7 | c00022d8 | c00022df | c00022e6 |
| c00022db | c00022dc | c00022dd | c00022de | c00022e5 | c00022ec |
| c00022e1 | c00022e2 | c00022e3 | c00022e4 | c00022eb | c00022f2 |
| c00022e7 | c00022e8 | c00022e9 | c00022ea | c00022f1 | c00022f8 |
| c00022ed | c00022ee | c00022ef | c00022f0 | c00022f7 | c00022fe |
| c00022f3 | c00022f4 | c00022f5 | c00022f6 | c00022fd | c0002304 |
| c00022f9 | c00022fa | c00022fb | c00022fc | c0002303 | c000230a |
| c00022ff | c0002300 | c0002301 | c0002302 | c0002309 | c0002310 |
| c0002305 | c0002306 | c0002307 | c0002308 | c000230f | c0002316 |
| c000230b | c000230c | c000230d | c000230e | c0002315 | c000231c |
| c0002311 | c0002312 | c0002313 | c0002314 | c000231b | c0002322 |
| c0002317 | c0002318 | c0002319 | c000231a | c0002321 | c0002328 |
| c000231d | c000231e | c000231f | c0002320 | c0002327 | c000232e |
| c0002323 | c0002324 | c0002325 | c0002326 | c000232d | c0002334 |
| c0002329 | c000232a | c000232b | c000232c | c0002333 | c000233a |
| c000232f | c0002330 | c0002331 | c0002332 | c0002339 | c0002340 |
| c0002335 | c0002336 | c0002337 | c0002338 | c000233f | c0002346 |
| c000233b | c000233c | c000233d | c000233e | c0002345 | c000234c |
| c0002341 | c0002342 | c0002343 | c0002344 | c000234b | c0002352 |
| c0002347 | c0002348 | c0002349 | c000234a | c0002351 | c0002358 |
| c000234d | c000234e | c000234f | c0002350 | c0002357 | c000235e |
| c0002353 | c0002354 | c0002355 | c0002356 | c000235d | c0002364 |
| c0002359 | c000235a | c000235b | c000235c | c0002363 | c000236a |
| c000235f | c0002360 | c0002361 | c0002362 | c0002369 | c0002370 |
| c0002365 | c0002366 | c0002367 | c0002368 | c000236f | c0002376 |
| c000236b | c000236c | c000236d | c000236e | c0002375 | c000237c |
| c0002371 | c0002372 | c0002373 | c0002374 | c000237b | c0002382 |
| c0002377 | c0002378 | c0002379 | c000237a | c0002381 | c0002388 |
| c000237d | c000237e | c000237f | c0002380 | c0002387 | c000238e |
| c0002383 | c0002384 | c0002385 | c0002386 | c000238d | c0002394 |
| c0002389 | c000238a | c000238b | c000238c | c0002393 | c000239a |
| c000238f | c0002390 | c0002391 | c0002392 | c0002399 | c00023a0 |
| c0002395 | c0002396 | c0002397 | c0002398 | c000239f | c00023a6 |
| c000239b | c000239c | c000239d | c000239e | c00023a5 | c00023ac |
| c00023a1 | c00023a2 | c00023a3 | c00023a4 | c00023ab | c00023b2 |
| c00023a7 | c00023a8 | c00023a9 | c00023aa | c00023b1 | c00023b8 |
| c00023ad | c00023ae | c00023af | c00023b0 | c00023b7 | c00023be |
| c00023b3 | c00023b4 | c00023b5 | c00023b6 | c00023bd | c00023c4 |
| c00023b9 | c00023ba | c00023bb | c00023bc | c00023c3 | c00023ca |
| c00023bf | c00023c0 | c00023c1 | c00023c2 | c00023c9 | c00023d0 |

| | | | | | |
|---|---|---|---|---|---|
| c00023c5 | c00023c6 | c00023c7 | c00023c8 | c00023cf | c00023d6 |
| c00023cb | c00023cc | c00023cd | c00023ce | c00023d5 | c00023dc |
| c00023d1 | c00023d2 | c00023d3 | c00023d4 | c00023db | c00023e2 |
| c00023d7 | c00023d8 | c00023d9 | c00023da | c00023e1 | c00023e8 |
| c00023dd | c00023de | c00023df | c00023e0 | c00023e7 | c00023ee |
| c00023e3 | c00023e4 | c00023e5 | c00023e6 | c00023ed | c00023f4 |
| c00023e9 | c00023ea | c00023eb | c00023ec | c00023f3 | c00023fa |
| c00023ef | c00023f0 | c00023f1 | c00023f2 | c00023f9 | c0002400 |
| c00023f5 | c00023f6 | c00023f7 | c00023f8 | c00023ff | c0002406 |
| c00023fb | c00023fc | c00023fd | c00023fe | c0002405 | c000240c |
| c0002401 | c0002402 | c0002403 | c0002404 | c000240b | c0002412 |
| c0002407 | c0002408 | c0002409 | c000240a | c0002411 | c0002418 |
| c000240d | c000240e | c000240f | c0002410 | c0002417 | c000241e |
| c0002413 | c0002414 | c0002415 | c0002416 | c000241d | c0002424 |
| c0002419 | c000241a | c000241b | c000241c | c0002423 | c000242a |
| c000241f | c0002420 | c0002421 | c0002422 | c0002429 | c0002430 |
| c0002425 | c0002426 | c0002427 | c0002428 | c000242f | c0002436 |
| c000242b | c000242c | c000242d | c000242e | c0002435 | c000243c |
| c0002431 | c0002432 | c0002433 | c0002434 | c000243b | c0002442 |
| c0002437 | c0002438 | c0002439 | c000243a | c0002441 | c0002448 |
| c000243d | c000243e | c000243f | c0002440 | c0002447 | c000244e |
| c0002443 | c0002444 | c0002445 | c0002446 | c000244d | c0002454 |
| c0002449 | c000244a | c000244b | c000244c | c0002453 | c000245a |
| c000244f | c0002450 | c0002451 | c0002452 | c0002459 | c0002460 |
| c0002455 | c0002456 | c0002457 | c0002458 | c000245f | c0002466 |
| c000245b | c000245c | c000245d | c000245e | c0002465 | c000246c |
| c0002461 | c0002462 | c0002463 | c0002464 | c000246b | c0002472 |
| c0002467 | c0002468 | c0002469 | c000246a | c0002471 | c0002478 |
| c000246d | c000246e | c000246f | c0002470 | c0002477 | c000247e |
| c0002473 | c0002474 | c0002475 | c0002476 | c000247d | c0002484 |
| c0002479 | c000247a | c000247b | c000247c | c0002483 | c000248a |
| c000247f | c0002480 | c0002481 | c0002482 | c0002489 | c0002490 |
| c0002485 | c0002486 | c0002487 | c0002488 | c000248f | c0002496 |
| c000248b | c000248c | c000248d | c000248e | c0002495 | c000249c |
| c0002491 | c0002492 | c0002493 | c0002494 | c000249b | c00024a2 |
| c0002497 | c0002498 | c0002499 | c000249a | c00024a1 | c00024a8 |
| c000249d | c000249e | c000249f | c00024a0 | c00024a7 | c00024ae |
| c00024a3 | c00024a4 | c00024a5 | c00024a6 | c00024ad | c00024b4 |
| c00024a9 | c00024aa | c00024ab | c00024ac | c00024b3 | c00024ba |
| c00024af | c00024b0 | c00024b1 | c00024b2 | c00024b9 | c00024c0 |
| c00024b5 | c00024b6 | c00024b7 | c00024b8 | c00024bf | c00024c6 |
| c00024bb | c00024bc | c00024bd | c00024be | c00024c5 | c00024cc |
| c00024c1 | c00024c2 | c00024c3 | c00024c4 | c00024cb | c00024d2 |

| | | | | | |
|---|---|---|---|---|---|
| c00024c7 | c00024c8 | c00024c9 | c00024ca | c00024d1 | c00024d8 |
| c00024cd | c00024ce | c00024cf | c00024d0 | c00024d7 | c00024de |
| c00024d3 | c00024d4 | c00024d5 | c00024d6 | c00024dd | c00024e4 |
| c00024d9 | c00024da | c00024db | c00024dc | c00024e3 | c00024ea |
| c00024df | c00024e0 | c00024e1 | c00024e2 | c00024e9 | c00024f0 |
| c00024e5 | c00024e6 | c00024e7 | c00024e8 | c00024ef | c00024f6 |
| c00024eb | c00024ec | c00024ed | c00024ee | c00024f5 | c00024fc |
| c00024f1 | c00024f2 | c00024f3 | c00024f4 | c00024fb | c0002502 |
| c00024f7 | c00024f8 | c00024f9 | c00024fa | c0002501 | c0002508 |
| c00024fd | c00024fe | c00024ff | c0002500 | c0002507 | c000250e |
| c0002503 | c0002504 | c0002505 | c0002506 | c000250d | c0002514 |
| c0002509 | c000250a | c000250b | c000250c | c0002513 | c000251a |
| c000250f | c0002510 | c0002511 | c0002512 | c0002519 | c0002520 |
| c0002515 | c0002516 | c0002517 | c0002518 | c000251f | c0002526 |
| c000251b | c000251c | c000251d | c000251e | c0002525 | c000252c |
| c0002521 | c0002522 | c0002523 | c0002524 | c000252b | c0002532 |
| c0002527 | c0002528 | c0002529 | c000252a | c0002531 | c0002538 |
| c000252d | c000252e | c000252f | c0002530 | c0002537 | c000253e |
| c0002533 | c0002534 | c0002535 | c0002536 | c000253d | c0002544 |
| c0002539 | c000253a | c000253b | c000253c | c0002543 | c000254a |
| c000253f | c0002540 | c0002541 | c0002542 | c0002549 | c0002550 |
| c0002545 | c0002546 | c0002547 | c0002548 | c000254f | c0002556 |
| c000254b | c000254c | c000254d | c000254e | c0002555 | c000255c |
| c0002551 | c0002552 | c0002553 | c0002554 | c000255b | c0002562 |
| c0002557 | c0002558 | c0002559 | c000255a | c0002561 | c0002568 |
| c000255d | c000255e | c000255f | c0002560 | c0002567 | c000256e |
| c0002563 | c0002564 | c0002565 | c0002566 | c000256d | c0002574 |
| c0002569 | c000256a | c000256b | c000256c | c0002573 | c000257a |
| c000256f | c0002570 | c0002571 | c0002572 | c0002579 | c0002580 |
| c0002575 | c0002576 | c0002577 | c0002578 | c000257f | c0002586 |
| c000257b | c000257c | c000257d | c000257e | c0002585 | c000258c |
| c0002581 | c0002582 | c0002583 | c0002584 | c000258b | c0002592 |
| c0002587 | c0002588 | c0002589 | c000258a | c0002591 | c0002598 |
| c000258d | c000258e | c000258f | c0002590 | c0002597 | c000259e |
| c0002593 | c0002594 | c0002595 | c0002596 | c000259d | c00025a4 |
| c0002599 | c000259a | c000259b | c000259c | c00025a3 | c00025aa |
| c000259f | c00025a0 | c00025a1 | c00025a2 | c00025a9 | c00025b0 |
| c00025a5 | c00025a6 | c00025a7 | c00025a8 | c00025af | c00025b6 |
| c00025ab | c00025ac | c00025ad | c00025ae | c00025b5 | c00025bc |
| c00025b1 | c00025b2 | c00025b3 | c00025b4 | c00025bb | c00025c2 |
| c00025b7 | c00025b8 | c00025b9 | c00025ba | c00025c1 | c00025c8 |
| c00025bd | c00025be | c00025bf | c00025c0 | c00025c7 | c00025ce |
| c00025c3 | c00025c4 | c00025c5 | c00025c6 | c00025cd | c00025d4 |

| | | | | | |
|---|---|---|---|---|---|
| c00025c9 | c00025ca | c00025cb | c00025cc | c00025d3 | c00025da |
| c00025cf | c00025d0 | c00025d1 | c00025d2 | c00025d9 | c00025e0 |
| c00025d5 | c00025d6 | c00025d7 | c00025d8 | c00025df | c00025e6 |
| c00025db | c00025dc | c00025dd | c00025de | c00025e5 | c00025ec |
| c00025e1 | c00025e2 | c00025e3 | c00025e4 | c00025eb | c00025f2 |
| c00025e7 | c00025e8 | c00025e9 | c00025ea | c00025f1 | c00025f8 |
| c00025ed | c00025ee | c00025ef | c00025f0 | c00025f7 | c00025fe |
| c00025f3 | c00025f4 | c00025f5 | c00025f6 | c00025fd | c0002604 |
| c00025f9 | c00025fa | c00025fb | c00025fc | c0002603 | c000260a |
| c00025ff | c0002600 | c0002601 | c0002602 | c0002609 | c0002610 |
| c0002605 | c0002606 | c0002607 | c0002608 | c000260f | c0002616 |
| c000260b | c000260c | c000260d | c000260e | c0002615 | c000261c |
| c0002611 | c0002612 | c0002613 | c0002614 | c000261b | c0002622 |
| c0002617 | c0002618 | c0002619 | c000261a | c0002621 | c0002628 |
| c000261d | c000261e | c000261f | c0002620 | c0002627 | c000262e |
| c0002623 | c0002624 | c0002625 | c0002626 | c000262d | c0002634 |
| c0002629 | c000262a | c000262b | c000262c | c0002633 | c000263a |
| c000262f | c0002630 | c0002631 | c0002632 | c0002639 | c0002640 |
| c0002635 | c0002636 | c0002637 | c0002638 | c000263f | c0002646 |
| c000263b | c000263c | c000263d | c000263e | c0002645 | c000264c |
| c0002641 | c0002642 | c0002643 | c0002644 | c000264b | c0002652 |
| c0002647 | c0002648 | c0002649 | c000264a | c0002651 | c0002658 |
| c000264d | c000264e | c000264f | c0002650 | c0002657 | c000265e |
| c0002653 | c0002654 | c0002655 | c0002656 | c000265d | c0002664 |
| c0002659 | c000265a | c000265b | c000265c | c0002663 | c000266a |
| c000265f | c0002660 | c0002661 | c0002662 | c0002669 | c0002670 |
| c0002665 | c0002666 | c0002667 | c0002668 | c000266f | c0002676 |
| c000266b | c000266c | c000266d | c000266e | c0002675 | c000267c |
| c0002671 | c0002672 | c0002673 | c0002674 | c000267b | c0002682 |
| c0002677 | c0002678 | c0002679 | c000267a | c0002681 | c0002688 |
| c000267d | c000267e | c000267f | c0002680 | c0002687 | c000268e |
| c0002683 | c0002684 | c0002685 | c0002686 | c000268d | c0002694 |
| c0002689 | c000268a | c000268b | c000268c | c0002693 | c000269a |
| c000268f | c0002690 | c0002691 | c0002692 | c0002699 | c00026a0 |
| c0002695 | c0002696 | c0002697 | c0002698 | c000269f | c00026a6 |
| c000269b | c000269c | c000269d | c000269e | c00026a5 | c00026ac |
| c00026a1 | c00026a2 | c00026a3 | c00026a4 | c00026ab | c00026b2 |
| c00026a7 | c00026a8 | c00026a9 | c00026aa | c00026b1 | c00026b8 |
| c00026ad | c00026ae | c00026af | c00026b0 | c00026b7 | c00026be |
| c00026b3 | c00026b4 | c00026b5 | c00026b6 | c00026bd | c00026c4 |
| c00026b9 | c00026ba | c00026bb | c00026bc | c00026c3 | c00026ca |
| c00026bf | c00026c0 | c00026c1 | c00026c2 | c00026c9 | c00026d0 |
| c00026c5 | c00026c6 | c00026c7 | c00026c8 | c00026cf | c00026d6 |

| | | | | | |
|---|---|---|---|---|---|
| c00026cb | c00026cc | c00026cd | c00026ce | c00026d5 | c00026dc |
| c00026d1 | c00026d2 | c00026d3 | c00026d4 | c00026db | c00026e2 |
| c00026d7 | c00026d8 | c00026d9 | c00026da | c00026e1 | c00026e8 |
| c00026dd | c00026de | c00026df | c00026e0 | c00026e7 | c00026ee |
| c00026e3 | c00026e4 | c00026e5 | c00026e6 | c00026ed | c00026f4 |
| c00026e9 | c00026ea | c00026eb | c00026ec | c00026f3 | c00026fa |
| c00026ef | c00026f0 | c00026f1 | c00026f2 | c00026f9 | c0002700 |
| c00026f5 | c00026f6 | c00026f7 | c00026f8 | c00026ff | c0002706 |
| c00026fb | c00026fc | c00026fd | c00026fe | c0002705 | c000270c |
| c0002701 | c0002702 | c0002703 | c0002704 | c000270b | c0002712 |
| c0002707 | c0002708 | c0002709 | c000270a | c0002711 | c0002718 |
| c000270d | c000270e | c000270f | c0002710 | c0002717 | c000271e |
| c0002713 | c0002714 | c0002715 | c0002716 | c000271d | c0002724 |
| c0002719 | c000271a | c000271b | c000271c | c0002723 | c000272a |
| c000271f | c0002720 | c0002721 | c0002722 | c0002729 | c0002730 |
| c0002725 | c0002726 | c0002727 | c0002728 | c000272f | c0002736 |
| c000272b | c000272c | c000272d | c000272e | c0002735 | c000273c |
| c0002731 | c0002732 | c0002733 | c0002734 | c000273b | c0002742 |
| c0002737 | c0002738 | c0002739 | c000273a | c0002741 | c0002748 |
| c000273d | c000273e | c000273f | c0002740 | c0002747 | c000274e |
| c0002743 | c0002744 | c0002745 | c0002746 | c000274d | c0002754 |
| c0002749 | c000274a | c000274b | c000274c | c0002753 | c000275a |
| c000274f | c0002750 | c0002751 | c0002752 | c0002759 | c0002760 |
| c0002755 | c0002756 | c0002757 | c0002758 | c000275f | c0002766 |
| c000275b | c000275c | c000275d | c000275e | c0002765 | c000276c |
| c0002761 | c0002762 | c0002763 | c0002764 | c000276b | c0002772 |
| c0002767 | c0002768 | c0002769 | c000276a | c0002771 | c0002778 |
| c000276d | c000276e | c000276f | c0002770 | c0002777 | c000277e |
| c0002773 | c0002774 | c0002775 | c0002776 | c000277d | c0002784 |
| c0002779 | c000277a | c000277b | c000277c | c0002783 | c000278a |
| c000277f | c0002780 | c0002781 | c0002782 | c0002789 | c0002790 |
| c0002785 | c0002786 | c0002787 | c0002788 | c000278f | c0002796 |
| c000278b | c000278c | c000278d | c000278e | c0002795 | c000279c |
| c0002791 | c0002792 | c0002793 | c0002794 | c000279b | c00027a2 |
| c0002797 | c0002798 | c0002799 | c000279a | c00027a1 | c00027a8 |
| c000279d | c000279e | c000279f | c00027a0 | c00027a7 | c00027ae |
| c00027a3 | c00027a4 | c00027a5 | c00027a6 | c00027ad | c00027b4 |
| c00027a9 | c00027aa | c00027ab | c00027ac | c00027b3 | c00027ba |
| c00027af | c00027b0 | c00027b1 | c00027b2 | c00027b9 | c00027c0 |
| c00027b5 | c00027b6 | c00027b7 | c00027b8 | c00027bf | c00027c6 |
| c00027bb | c00027bc | c00027bd | c00027be | c00027c5 | c00027cc |
| c00027c1 | c00027c2 | c00027c3 | c00027c4 | c00027cb | c00027d2 |
| c00027c7 | c00027c8 | c00027c9 | c00027ca | c00027d1 | c00027d8 |

| | | | | | |
|---|---|---|---|---|---|
| c00027cd | c00027ce | c00027cf | c00027d0 | c00027d7 | c00027de |
| c00027d3 | c00027d4 | c00027d5 | c00027d6 | c00027dd | c00027e4 |
| c00027d9 | c00027da | c00027db | c00027dc | c00027e3 | c00027ea |
| c00027df | c00027e0 | c00027e1 | c00027e2 | c00027e9 | c00027f0 |
| c00027e5 | c00027e6 | c00027e7 | c00027e8 | c00027ef | c00027f6 |
| c00027eb | c00027ec | c00027ed | c00027ee | c00027f5 | c00027fc |
| c00027f1 | c00027f2 | c00027f3 | c00027f4 | c00027fb | c0002802 |
| c00027f7 | c00027f8 | c00027f9 | c00027fa | c0002801 | c0002808 |
| c00027fd | c00027fe | c00027ff | c0002800 | c0002807 | c000280e |
| c0002803 | c0002804 | c0002805 | c0002806 | c000280d | c0002814 |
| c0002809 | c000280a | c000280b | c000280c | c0002813 | c000281a |
| c000280f | c0002810 | c0002811 | c0002812 | c0002819 | c0002820 |
| c0002815 | c0002816 | c0002817 | c0002818 | c000281f | c0002826 |
| c000281b | c000281c | c000281d | c000281e | c0002825 | c000282c |
| c0002821 | c0002822 | c0002823 | c0002824 | c000282b | c0002832 |
| c0002827 | c0002828 | c0002829 | c000282a | c0002831 | c0002838 |
| c000282d | c000282e | c000282f | c0002830 | c0002837 | c000283e |
| c0002833 | c0002834 | c0002835 | c0002836 | c000283d | c0002844 |
| c0002839 | c000283a | c000283b | c000283c | c0002843 | c000284a |
| c000283f | c0002840 | c0002841 | c0002842 | c0002849 | c0002850 |
| c0002845 | c0002846 | c0002847 | c0002848 | c000284f | c0002856 |
| c000284b | c000284c | c000284d | c000284e | c0002855 | c000285c |
| c0002851 | c0002852 | c0002853 | c0002854 | c000285b | c0002862 |
| c0002857 | c0002858 | c0002859 | c000285a | c0002861 | c0002868 |
| c000285d | c000285e | c000285f | c0002860 | c0002867 | c000286e |
| c0002863 | c0002864 | c0002865 | c0002866 | c000286d | c0002874 |
| c0002869 | c000286a | c000286b | c000286c | c0002873 | c000287a |
| c000286f | c0002870 | c0002871 | c0002872 | c0002879 | c0002880 |
| c0002875 | c0002876 | c0002877 | c0002878 | c000287f | c0002886 |
| c000287b | c000287c | c000287d | c000287e | c0002885 | c000288c |
| c0002881 | c0002882 | c0002883 | c0002884 | c000288b | c0002892 |
| c0002887 | c0002888 | c0002889 | c000288a | c0002891 | c0002898 |
| c000288d | c000288e | c000288f | c0002890 | c0002897 | c000289e |
| c0002893 | c0002894 | c0002895 | c0002896 | c000289d | c00028a4 |
| c0002899 | c000289a | c000289b | c000289c | c00028a3 | c00028aa |
| c000289f | c00028a0 | c00028a1 | c00028a2 | c00028a9 | c00028b0 |
| c00028a5 | c00028a6 | c00028a7 | c00028a8 | c00028af | c00028b6 |
| c00028ab | c00028ac | c00028ad | c00028ae | c00028b5 | c00028bc |
| c00028b1 | c00028b2 | c00028b3 | c00028b4 | c00028bb | c00028c2 |
| c00028b7 | c00028b8 | c00028b9 | c00028ba | c00028c1 | c00028c8 |
| c00028bd | c00028be | c00028bf | c00028c0 | c00028c7 | c00028ce |
| c00028c3 | c00028c4 | c00028c5 | c00028c6 | c00028cd | c00028d4 |
| c00028c9 | c00028ca | c00028cb | c00028cc | c00028d3 | c00028da |

| | | | | | |
|---|---|---|---|---|---|
| c00028cf | c00028d0 | c00028d1 | c00028d2 | c00028d9 | c00028e0 |
| c00028d5 | c00028d6 | c00028d7 | c00028d8 | c00028df | c00028e6 |
| c00028db | c00028dc | c00028dd | c00028de | c00028e5 | c00028ec |
| c00028e1 | c00028e2 | c00028e3 | c00028e4 | c00028eb | c00028f2 |
| c00028e7 | c00028e8 | c00028e9 | c00028ea | c00028f1 | c00028f8 |
| c00028ed | c00028ee | c00028ef | c00028f0 | c00028f7 | c00028fe |
| c00028f3 | c00028f4 | c00028f5 | c00028f6 | c00028fd | c0002904 |
| c00028f9 | c00028fa | c00028fb | c00028fc | c0002903 | c000290a |
| c00028ff | c0002900 | c0002901 | c0002902 | c0002909 | c0002910 |
| c0002905 | c0002906 | c0002907 | c0002908 | c000290f | c0002916 |
| c000290b | c000290c | c000290d | c000290e | c0002915 | c000291c |
| c0002911 | c0002912 | c0002913 | c0002914 | c000291b | c0002922 |
| c0002917 | c0002918 | c0002919 | c000291a | c0002921 | c0002928 |
| c000291d | c000291e | c000291f | c0002920 | c0002927 | c000292e |
| c0002923 | c0002924 | c0002925 | c0002926 | c000292d | c0002934 |
| c0002929 | c000292a | c000292b | c000292c | c0002933 | c000293a |
| c000292f | c0002930 | c0002931 | c0002932 | c0002939 | c0002940 |
| c0002935 | c0002936 | c0002937 | c0002938 | c000293f | c0002946 |
| c000293b | c000293c | c000293d | c000293e | c0002945 | c000294c |
| c0002941 | c0002942 | c0002943 | c0002944 | c000294b | c0002952 |
| c0002947 | c0002948 | c0002949 | c000294a | c0002951 | c0002958 |
| c000294d | c000294e | c000294f | c0002950 | c0002957 | c000295e |
| c0002953 | c0002954 | c0002955 | c0002956 | c000295d | c0002964 |
| c0002959 | c000295a | c000295b | c000295c | c0002963 | c000296a |
| c000295f | c0002960 | c0002961 | c0002962 | c0002969 | c0002970 |
| c0002965 | c0002966 | c0002967 | c0002968 | c000296f | c0002976 |
| c000296b | c000296c | c000296d | c000296e | c0002975 | c000297c |
| c0002971 | c0002972 | c0002973 | c0002974 | c000297b | c0002982 |
| c0002977 | c0002978 | c0002979 | c000297a | c0002981 | c0002988 |
| c000297d | c000297e | c000297f | c0002980 | c0002987 | c000298e |
| c0002983 | c0002984 | c0002985 | c0002986 | c000298d | c0002994 |
| c0002989 | c000298a | c000298b | c000298c | c0002993 | c000299a |
| c000298f | c0002990 | c0002991 | c0002992 | c0002999 | c00029a0 |
| c0002995 | c0002996 | c0002997 | c0002998 | c000299f | c00029a6 |
| c000299b | c000299c | c000299d | c000299e | c00029a5 | c00029ac |
| c00029a1 | c00029a2 | c00029a3 | c00029a4 | c00029ab | c00029b2 |
| c00029a7 | c00029a8 | c00029a9 | c00029aa | c00029b1 | c00029b8 |
| c00029ad | c00029ae | c00029af | c00029b0 | c00029b7 | c00029be |
| c00029b3 | c00029b4 | c00029b5 | c00029b6 | c00029bd | c00029c4 |
| c00029b9 | c00029ba | c00029bb | c00029bc | c00029c3 | c00029ca |
| c00029bf | c00029c0 | c00029c1 | c00029c2 | c00029c9 | c00029d0 |
| c00029c5 | c00029c6 | c00029c7 | c00029c8 | c00029cf | c00029d6 |
| c00029cb | c00029cc | c00029cd | c00029ce | c00029d5 | c00029dc |

| | | | | | |
|---|---|---|---|---|---|
| c00029d1 | c00029d2 | c00029d3 | c00029d4 | c00029db | c00029e2 |
| c00029d7 | c00029d8 | c00029d9 | c00029da | c00029e1 | c00029e8 |
| c00029dd | c00029de | c00029df | c00029e0 | c00029e7 | c00029ee |
| c00029e3 | c00029e4 | c00029e5 | c00029e6 | c00029ed | c00029f4 |
| c00029e9 | c00029ea | c00029eb | c00029ec | c00029f3 | c00029fa |
| c00029ef | c00029f0 | c00029f1 | c00029f2 | c00029f9 | c0002a00 |
| c00029f5 | c00029f6 | c00029f7 | c00029f8 | c00029ff | c0002a06 |
| c00029fb | c00029fc | c00029fd | c00029fe | c0002a05 | c0002a0c |
| c0002a01 | c0002a02 | c0002a03 | c0002a04 | c0002a0b | c0002a12 |
| c0002a07 | c0002a08 | c0002a09 | c0002a0a | c0002a11 | c0002a18 |
| c0002a0d | c0002a0e | c0002a0f | c0002a10 | c0002a17 | c0002a1e |
| c0002a13 | c0002a14 | c0002a15 | c0002a16 | c0002a1d | c0002a24 |
| c0002a19 | c0002a1a | c0002a1b | c0002a1c | c0002a23 | c0002a2a |
| c0002a1f | c0002a20 | c0002a21 | c0002a22 | c0002a29 | c0002a30 |
| c0002a25 | c0002a26 | c0002a27 | c0002a28 | c0002a2f | c0002a36 |
| c0002a2b | c0002a2c | c0002a2d | c0002a2e | c0002a35 | c0002a3c |
| c0002a31 | c0002a32 | c0002a33 | c0002a34 | c0002a3b | c0002a42 |
| c0002a37 | c0002a38 | c0002a39 | c0002a3a | c0002a41 | c0002a48 |
| c0002a3d | c0002a3e | c0002a3f | c0002a40 | c0002a47 | c0002a4e |
| c0002a43 | c0002a44 | c0002a45 | c0002a46 | c0002a4d | c0002a54 |
| c0002a49 | c0002a4a | c0002a4b | c0002a4c | c0002a53 | c0002a5a |
| c0002a4f | c0002a50 | c0002a51 | c0002a52 | c0002a59 | c0002a60 |
| c0002a55 | c0002a56 | c0002a57 | c0002a58 | c0002a5f | c0002a66 |
| c0002a5b | c0002a5c | c0002a5d | c0002a5e | c0002a65 | c0002a6c |
| c0002a61 | c0002a62 | c0002a63 | c0002a64 | c0002a6b | c0002a72 |
| c0002a67 | c0002a68 | c0002a69 | c0002a6a | c0002a71 | c0002a78 |
| c0002a6d | c0002a6e | c0002a6f | c0002a70 | c0002a77 | c0002a7e |
| c0002a73 | c0002a74 | c0002a75 | c0002a76 | c0002a7d | c0002a84 |
| c0002a79 | c0002a7a | c0002a7b | c0002a7c | c0002a83 | c0002a8a |
| c0002a7f | c0002a80 | c0002a81 | c0002a82 | c0002a89 | c0002a90 |
| c0002a85 | c0002a86 | c0002a87 | c0002a88 | c0002a8f | c0002a96 |
| c0002a8b | c0002a8c | c0002a8d | c0002a8e | c0002a95 | c0002a9c |
| c0002a91 | c0002a92 | c0002a93 | c0002a94 | c0002a9b | c0002aa2 |
| c0002a97 | c0002a98 | c0002a99 | c0002a9a | c0002aa1 | c0002aa8 |
| c0002a9d | c0002a9e | c0002a9f | c0002aa0 | c0002aa7 | c0002aae |
| c0002aa3 | c0002aa4 | c0002aa5 | c0002aa6 | c0002aad | c0002ab4 |
| c0002aa9 | c0002aaa | c0002aab | c0002aac | c0002ab3 | c0002aba |
| c0002aaf | c0002ab0 | c0002ab1 | c0002ab2 | c0002ab9 | c0002ac0 |
| c0002ab5 | c0002ab6 | c0002ab7 | c0002ab8 | c0002abf | c0002ac6 |
| c0002abb | c0002abc | c0002abd | c0002abe | c0002ac5 | c0002acc |
| c0002ac1 | c0002ac2 | c0002ac3 | c0002ac4 | c0002acb | c0002ad2 |
| c0002ac7 | c0002ac8 | c0002ac9 | c0002aca | c0002ad1 | c0002ad8 |
| c0002acd | c0002ace | c0002acf | c0002ad0 | c0002ad7 | c0002ade |

| | | | | | |
|---|---|---|---|---|---|
| c0002ad3 | c0002ad4 | c0002ad5 | c0002ad6 | c0002add | c0002ae4 |
| c0002ad9 | c0002ada | c0002adb | c0002adc | c0002ae3 | c0002aea |
| c0002adf | c0002ae0 | c0002ae1 | c0002ae2 | c0002ae9 | c0002af0 |
| c0002ae5 | c0002ae6 | c0002ae7 | c0002ae8 | c0002aef | c0002af6 |
| c0002aeb | c0002aec | c0002aed | c0002aee | c0002af5 | c0002afc |
| c0002af1 | c0002af2 | c0002af3 | c0002af4 | c0002afb | c0002b02 |
| c0002af7 | c0002af8 | c0002af9 | c0002afa | c0002b01 | c0002b08 |
| c0002afd | c0002afe | c0002aff | c0002b00 | c0002b07 | c0002b0e |
| c0002b03 | c0002b04 | c0002b05 | c0002b06 | c0002b0d | c0002b14 |
| c0002b09 | c0002b0a | c0002b0b | c0002b0c | c0002b13 | c0002b1a |
| c0002b0f | c0002b10 | c0002b11 | c0002b12 | c0002b19 | c0002b20 |
| c0002b15 | c0002b16 | c0002b17 | c0002b18 | c0002b1f | c0002b26 |
| c0002b1b | c0002b1c | c0002b1d | c0002b1e | c0002b25 | c0002b2c |
| c0002b21 | c0002b22 | c0002b23 | c0002b24 | c0002b2b | c0002b32 |
| c0002b27 | c0002b28 | c0002b29 | c0002b2a | c0002b31 | c0002b38 |
| c0002b2d | c0002b2e | c0002b2f | c0002b30 | c0002b37 | c0002b3e |
| c0002b33 | c0002b34 | c0002b35 | c0002b36 | c0002b3d | c0002b44 |
| c0002b39 | c0002b3a | c0002b3b | c0002b3c | c0002b43 | c0002b4a |
| c0002b3f | c0002b40 | c0002b41 | c0002b42 | c0002b49 | c0002b50 |
| c0002b45 | c0002b46 | c0002b47 | c0002b48 | c0002b4f | c0002b56 |
| c0002b4b | c0002b4c | c0002b4d | c0002b4e | c0002b55 | c0002b5c |
| c0002b51 | c0002b52 | c0002b53 | c0002b54 | c0002b5b | c0002b62 |
| c0002b57 | c0002b58 | c0002b59 | c0002b5a | c0002b61 | c0002b68 |
| c0002b5d | c0002b5e | c0002b5f | c0002b60 | c0002b67 | c0002b6e |
| c0002b63 | c0002b64 | c0002b65 | c0002b66 | c0002b6d | c0002b74 |
| c0002b69 | c0002b6a | c0002b6b | c0002b6c | c0002b73 | c0002b7a |
| c0002b6f | c0002b70 | c0002b71 | c0002b72 | c0002b79 | c0002b80 |
| c0002b75 | c0002b76 | c0002b77 | c0002b78 | c0002b7f | c0002b86 |
| c0002b7b | c0002b7c | c0002b7d | c0002b7e | c0002b85 | c0002b8c |
| c0002b81 | c0002b82 | c0002b83 | c0002b84 | c0002b8b | c0002b92 |
| c0002b87 | c0002b88 | c0002b89 | c0002b8a | c0002b91 | c0002b98 |
| c0002b8d | c0002b8e | c0002b8f | c0002b90 | c0002b97 | c0002b9e |
| c0002b93 | c0002b94 | c0002b95 | c0002b96 | c0002b9d | c0002ba4 |
| c0002b99 | c0002b9a | c0002b9b | c0002b9c | c0002ba3 | c0002baa |
| c0002b9f | c0002ba0 | c0002ba1 | c0002ba2 | c0002ba9 | c0002bb0 |
| c0002ba5 | c0002ba6 | c0002ba7 | c0002ba8 | c0002baf | c0002bb6 |
| c0002bab | c0002bac | c0002bad | c0002bae | c0002bb5 | c0002bbc |
| c0002bb1 | c0002bb2 | c0002bb3 | c0002bb4 | c0002bbb | c0002bc2 |
| c0002bb7 | c0002bb8 | c0002bb9 | c0002bba | c0002bc1 | c0002bc8 |
| c0002bbd | c0002bbe | c0002bbf | c0002bc0 | c0002bc7 | c0002bce |
| c0002bc3 | c0002bc4 | c0002bc5 | c0002bc6 | c0002bcd | c0002bd4 |
| c0002bc9 | c0002bca | c0002bcb | c0002bcc | c0002bd3 | c0002bda |
| c0002bcf | c0002bd0 | c0002bd1 | c0002bd2 | c0002bd9 | c0002be0 |

| | | | | | |
|---|---|---|---|---|---|
| c0002bd5 | c0002bd6 | c0002bd7 | c0002bd8 | c0002bdf | c0002be6 |
| c0002bdb | c0002bdc | c0002bdd | c0002bde | c0002be5 | c0002bec |
| c0002be1 | c0002be2 | c0002be3 | c0002be4 | c0002beb | c0002bf2 |
| c0002be7 | c0002be8 | c0002be9 | c0002bea | c0002bf1 | c0002bf8 |
| c0002bed | c0002bee | c0002bef | c0002bf0 | c0002bf7 | c0002bfe |
| c0002bf3 | c0002bf4 | c0002bf5 | c0002bf6 | c0002bfd | c0002c04 |
| c0002bf9 | c0002bfa | c0002bfb | c0002bfc | c0002c03 | c0002c0a |
| c0002bff | c0002c00 | c0002c01 | c0002c02 | c0002c09 | c0002c10 |
| c0002c05 | c0002c06 | c0002c07 | c0002c08 | c0002c0f | c0002c16 |
| c0002c0b | c0002c0c | c0002c0d | c0002c0e | c0002c15 | c0002c1c |
| c0002c11 | c0002c12 | c0002c13 | c0002c14 | c0002c1b | c0002c22 |
| c0002c17 | c0002c18 | c0002c19 | c0002c1a | c0002c21 | c0002c28 |
| c0002c1d | c0002c1e | c0002c1f | c0002c20 | c0002c27 | c0002c2e |
| c0002c23 | c0002c24 | c0002c25 | c0002c26 | c0002c2d | c0002c34 |
| c0002c29 | c0002c2a | c0002c2b | c0002c2c | c0002c33 | c0002c3a |
| c0002c2f | c0002c30 | c0002c31 | c0002c32 | c0002c39 | c0002c40 |
| c0002c35 | c0002c36 | c0002c37 | c0002c38 | c0002c3f | c0002c46 |
| c0002c3b | c0002c3c | c0002c3d | c0002c3e | c0002c45 | c0002c4c |
| c0002c41 | c0002c42 | c0002c43 | c0002c44 | c0002c4b | c0002c52 |
| c0002c47 | c0002c48 | c0002c49 | c0002c4a | c0002c51 | c0002c58 |
| c0002c4d | c0002c4e | c0002c4f | c0002c50 | c0002c57 | c0002c5e |
| c0002c53 | c0002c54 | c0002c55 | c0002c56 | c0002c5d | c0002c64 |
| c0002c59 | c0002c5a | c0002c5b | c0002c5c | c0002c63 | c0002c6a |
| c0002c5f | c0002c60 | c0002c61 | c0002c62 | c0002c69 | c0002c70 |
| c0002c65 | c0002c66 | c0002c67 | c0002c68 | c0002c6f | c0002c76 |
| c0002c6b | c0002c6c | c0002c6d | c0002c6e | c0002c75 | c0002c7c |
| c0002c71 | c0002c72 | c0002c73 | c0002c74 | c0002c7b | c0002c82 |
| c0002c77 | c0002c78 | c0002c79 | c0002c7a | c0002c81 | c0002c88 |
| c0002c7d | c0002c7e | c0002c7f | c0002c80 | c0002c87 | c0002c8e |
| c0002c83 | c0002c84 | c0002c85 | c0002c86 | c0002c8d | c0002c94 |
| c0002c89 | c0002c8a | c0002c8b | c0002c8c | c0002c93 | c0002c9a |
| c0002c8f | c0002c90 | c0002c91 | c0002c92 | c0002c99 | c0002ca0 |
| c0002c95 | c0002c96 | c0002c97 | c0002c98 | c0002c9f | c0002ca6 |
| c0002c9b | c0002c9c | c0002c9d | c0002c9e | c0002ca5 | c0002cac |
| c0002ca1 | c0002ca2 | c0002ca3 | c0002ca4 | c0002cab | c0002cb2 |
| c0002ca7 | c0002ca8 | c0002ca9 | c0002caa | c0002cb1 | c0002cb8 |
| c0002cad | c0002cae | c0002caf | c0002cb0 | c0002cb7 | c0002cbe |
| c0002cb3 | c0002cb4 | c0002cb5 | c0002cb6 | c0002cbd | c0002cc4 |
| c0002cb9 | c0002cba | c0002cbb | c0002cbc | c0002cc3 | c0002cca |
| c0002cbf | c0002cc0 | c0002cc1 | c0002cc2 | c0002cc9 | c0002cd0 |
| c0002cc5 | c0002cc6 | c0002cc7 | c0002cc8 | c0002ccf | c0002cd6 |
| c0002ccb | c0002ccc | c0002ccd | c0002cce | c0002cd5 | c0002cdc |
| c0002cd1 | c0002cd2 | c0002cd3 | c0002cd4 | c0002cdb | c0002ce2 |

| | | | | | |
|---|---|---|---|---|---|
| c0002cd7 | c0002cd8 | c0002cd9 | c0002cda | c0002ce1 | c0002ce8 |
| c0002cdd | c0002cde | c0002cdf | c0002ce0 | c0002ce7 | c0002cee |
| c0002ce3 | c0002ce4 | c0002ce5 | c0002ce6 | c0002ced | c0002cf4 |
| c0002ce9 | c0002cea | c0002ceb | c0002cec | c0002cf3 | c0002cfa |
| c0002cef | c0002cf0 | c0002cf1 | c0002cf2 | c0002cf9 | c0002d00 |
| c0002cf5 | c0002cf6 | c0002cf7 | c0002cf8 | c0002cff | c0002d06 |
| c0002cfb | c0002cfc | c0002cfd | c0002cfe | c0002d05 | c0002d0c |
| c0002d01 | c0002d02 | c0002d03 | c0002d04 | c0002d0b | c0002d12 |
| c0002d07 | c0002d08 | c0002d09 | c0002d0a | c0002d11 | c0002d18 |
| c0002d0d | c0002d0e | c0002d0f | c0002d10 | c0002d17 | c0002d1e |
| c0002d13 | c0002d14 | c0002d15 | c0002d16 | c0002d1d | c0002d24 |
| c0002d19 | c0002d1a | c0002d1b | c0002d1c | c0002d23 | c0002d2a |
| c0002d1f | c0002d20 | c0002d21 | c0002d22 | c0002d29 | c0002d30 |
| c0002d25 | c0002d26 | c0002d27 | c0002d28 | c0002d2f | c0002d36 |
| c0002d2b | c0002d2c | c0002d2d | c0002d2e | c0002d35 | c0002d3c |
| c0002d31 | c0002d32 | c0002d33 | c0002d34 | c0002d3b | c0002d42 |
| c0002d37 | c0002d38 | c0002d39 | c0002d3a | c0002d41 | c0002d48 |
| c0002d3d | c0002d3e | c0002d3f | c0002d40 | c0002d47 | c0002d4e |
| c0002d43 | c0002d44 | c0002d45 | c0002d46 | c0002d4d | c0002d54 |
| c0002d49 | c0002d4a | c0002d4b | c0002d4c | c0002d53 | c0002d5a |
| c0002d4f | c0002d50 | c0002d51 | c0002d52 | c0002d59 | c0002d60 |
| c0002d55 | c0002d56 | c0002d57 | c0002d58 | c0002d5f | c0002d66 |
| c0002d5b | c0002d5c | c0002d5d | c0002d5e | c0002d65 | c0002d6c |
| c0002d61 | c0002d62 | c0002d63 | c0002d64 | c0002d6b | c0002d72 |
| c0002d67 | c0002d68 | c0002d69 | c0002d6a | c0002d71 | c0002d78 |
| c0002d6d | c0002d6e | c0002d6f | c0002d70 | c0002d77 | c0002d7e |
| c0002d73 | c0002d74 | c0002d75 | c0002d76 | c0002d7d | c0002d84 |
| c0002d79 | c0002d7a | c0002d7b | c0002d7c | c0002d83 | c0002d8a |
| c0002d7f | c0002d80 | c0002d81 | c0002d82 | c0002d89 | c0002d90 |
| c0002d85 | c0002d86 | c0002d87 | c0002d88 | c0002d8f | c0002d96 |
| c0002d8b | c0002d8c | c0002d8d | c0002d8e | c0002d95 | c0002d9c |
| c0002d91 | c0002d92 | c0002d93 | c0002d94 | c0002d9b | c0002da2 |
| c0002d97 | c0002d98 | c0002d99 | c0002d9a | c0002da1 | c0002da8 |
| c0002d9d | c0002d9e | c0002d9f | c0002da0 | c0002da7 | c0002dae |
| c0002da3 | c0002da4 | c0002da5 | c0002da6 | c0002dad | c0002db4 |
| c0002da9 | c0002daa | c0002dab | c0002dac | c0002db3 | c0002dba |
| c0002daf | c0002db0 | c0002db1 | c0002db2 | c0002db9 | c0002dc0 |
| c0002db5 | c0002db6 | c0002db7 | c0002db8 | c0002dbf | c0002dc6 |
| c0002dbb | c0002dbc | c0002dbd | c0002dbe | c0002dc5 | c0002dcc |
| c0002dc1 | c0002dc2 | c0002dc3 | c0002dc4 | c0002dcb | c0002dd2 |
| c0002dc7 | c0002dc8 | c0002dc9 | c0002dca | c0002dd1 | c0002dd8 |
| c0002dcd | c0002dce | c0002dcf | c0002dd0 | c0002dd7 | c0002dde |
| c0002dd3 | c0002dd4 | c0002dd5 | c0002dd6 | c0002ddd | c0002de4 |

| | | | | | |
|---|---|---|---|---|---|
| c0002dd9 | c0002dda | c0002ddb | c0002ddc | c0002de3 | c0002dea |
| c0002ddf | c0002de0 | c0002de1 | c0002de2 | c0002de9 | c0002df0 |
| c0002de5 | c0002de6 | c0002de7 | c0002de8 | c0002def | c0002df6 |
| c0002deb | c0002dec | c0002ded | c0002dee | c0002df5 | c0002dfc |
| c0002df1 | c0002df2 | c0002df3 | c0002df4 | c0002dfb | c0002e02 |
| c0002df7 | c0002df8 | c0002df9 | c0002dfa | c0002e01 | c0002e08 |
| c0002dfd | c0002dfe | c0002dff | c0002e00 | c0002e07 | c0002e0e |
| c0002e03 | c0002e04 | c0002e05 | c0002e06 | c0002e0d | c0002e14 |
| c0002e09 | c0002e0a | c0002e0b | c0002e0c | c0002e13 | c0002e1a |
| c0002e0f | c0002e10 | c0002e11 | c0002e12 | c0002e19 | c0002e20 |
| c0002e15 | c0002e16 | c0002e17 | c0002e18 | c0002e1f | c0002e26 |
| c0002e1b | c0002e1c | c0002e1d | c0002e1e | c0002e25 | c0002e2c |
| c0002e21 | c0002e22 | c0002e23 | c0002e24 | c0002e2b | c0002e32 |
| c0002e27 | c0002e28 | c0002e29 | c0002e2a | c0002e31 | c0002e38 |
| c0002e2d | c0002e2e | c0002e2f | c0002e30 | c0002e37 | c0002e3e |
| c0002e33 | c0002e34 | c0002e35 | c0002e36 | c0002e3d | c0002e44 |
| c0002e39 | c0002e3a | c0002e3b | c0002e3c | c0002e43 | c0002e4a |
| c0002e3f | c0002e40 | c0002e41 | c0002e42 | c0002e49 | c0002e50 |
| c0002e45 | c0002e46 | c0002e47 | c0002e48 | c0002e4f | c0002e56 |
| c0002e4b | c0002e4c | c0002e4d | c0002e4e | c0002e55 | c0002e5c |
| c0002e51 | c0002e52 | c0002e53 | c0002e54 | c0002e5b | c0002e62 |
| c0002e57 | c0002e58 | c0002e59 | c0002e5a | c0002e61 | c0002e68 |
| c0002e5d | c0002e5e | c0002e5f | c0002e60 | c0002e67 | c0002e6e |
| c0002e63 | c0002e64 | c0002e65 | c0002e66 | c0002e6d | c0002e74 |
| c0002e69 | c0002e6a | c0002e6b | c0002e6c | c0002e73 | c0002e7a |
| c0002e6f | c0002e70 | c0002e71 | c0002e72 | c0002e79 | c0002e80 |
| c0002e75 | c0002e76 | c0002e77 | c0002e78 | c0002e7f | c0002e86 |
| c0002e7b | c0002e7c | c0002e7d | c0002e7e | c0002e85 | c0002e8c |
| c0002e81 | c0002e82 | c0002e83 | c0002e84 | c0002e8b | c0002e92 |
| c0002e87 | c0002e88 | c0002e89 | c0002e8a | c0002e91 | c0002e98 |
| c0002e8d | c0002e8e | c0002e8f | c0002e90 | c0002e97 | c0002e9e |
| c0002e93 | c0002e94 | c0002e95 | c0002e96 | c0002e9d | c0002ea4 |
| c0002e99 | c0002e9a | c0002e9b | c0002e9c | c0002ea3 | c0002eaa |
| c0002e9f | c0002ea0 | c0002ea1 | c0002ea2 | c0002ea9 | c0002eb0 |
| c0002ea5 | c0002ea6 | c0002ea7 | c0002ea8 | c0002eaf | c0002eb6 |
| c0002eab | c0002eac | c0002ead | c0002eae | c0002eb5 | c0002ebc |
| c0002eb1 | c0002eb2 | c0002eb3 | c0002eb4 | c0002ebb | c0002ec2 |
| c0002eb7 | c0002eb8 | c0002eb9 | c0002eba | c0002ec1 | c0002ec8 |
| c0002ebd | c0002ebe | c0002ebf | c0002ec0 | c0002ec7 | c0002ece |
| c0002ec3 | c0002ec4 | c0002ec5 | c0002ec6 | c0002ecd | c0002ed4 |
| c0002ec9 | c0002eca | c0002ecb | c0002ecc | c0002ed3 | c0002eda |
| c0002ecf | c0002ed0 | c0002ed1 | c0002ed2 | c0002ed9 | c0002ee0 |
| c0002ed5 | c0002ed6 | c0002ed7 | c0002ed8 | c0002edf | c0002ee6 |

| | | | | | |
|---|---|---|---|---|---|
| c0002edb | c0002edc | c0002edd | c0002ede | c0002ee5 | c0002eec |
| c0002ee1 | c0002ee2 | c0002ee3 | c0002ee4 | c0002eeb | c0002ef2 |
| c0002ee7 | c0002ee8 | c0002ee9 | c0002eea | c0002ef1 | c0002ef8 |
| c0002eed | c0002eee | c0002eef | c0002ef0 | c0002ef7 | c0002efe |
| c0002ef3 | c0002ef4 | c0002ef5 | c0002ef6 | c0002efd | c0002f04 |
| c0002ef9 | c0002efa | c0002efb | c0002efc | c0002f03 | c0002f0a |
| c0002eff | c0002f00 | c0002f01 | c0002f02 | c0002f09 | c0002f10 |
| c0002f05 | c0002f06 | c0002f07 | c0002f08 | c0002f0f | c0002f16 |
| c0002f0b | c0002f0c | c0002f0d | c0002f0e | c0002f15 | c0002f1c |
| c0002f11 | c0002f12 | c0002f13 | c0002f14 | c0002f1b | c0002f22 |
| c0002f17 | c0002f18 | c0002f19 | c0002f1a | c0002f21 | c0002f28 |
| c0002f1d | c0002f1e | c0002f1f | c0002f20 | c0002f27 | c0002f2e |
| c0002f23 | c0002f24 | c0002f25 | c0002f26 | c0002f2d | c0002f34 |
| c0002f29 | c0002f2a | c0002f2b | c0002f2c | c0002f33 | c0002f3a |
| c0002f2f | c0002f30 | c0002f31 | c0002f32 | c0002f39 | c0002f40 |
| c0002f35 | c0002f36 | c0002f37 | c0002f38 | c0002f3f | c0002f46 |
| c0002f3b | c0002f3c | c0002f3d | c0002f3e | c0002f45 | c0002f4c |
| c0002f41 | c0002f42 | c0002f43 | c0002f44 | c0002f4b | c0002f52 |
| c0002f47 | c0002f48 | c0002f49 | c0002f4a | c0002f51 | c0002f58 |
| c0002f4d | c0002f4e | c0002f4f | c0002f50 | c0002f57 | c0002f5e |
| c0002f53 | c0002f54 | c0002f55 | c0002f56 | c0002f5d | c0002f64 |
| c0002f59 | c0002f5a | c0002f5b | c0002f5c | c0002f63 | c0002f6a |
| c0002f5f | c0002f60 | c0002f61 | c0002f62 | c0002f69 | c0002f70 |
| c0002f65 | c0002f66 | c0002f67 | c0002f68 | c0002f6f | c0002f76 |
| c0002f6b | c0002f6c | c0002f6d | c0002f6e | c0002f75 | c0002f7c |
| c0002f71 | c0002f72 | c0002f73 | c0002f74 | c0002f7b | c0002f82 |
| c0002f77 | c0002f78 | c0002f79 | c0002f7a | c0002f81 | c0002f88 |
| c0002f7d | c0002f7e | c0002f7f | c0002f80 | c0002f87 | c0002f8e |
| c0002f83 | c0002f84 | c0002f85 | c0002f86 | c0002f8d | c0002f94 |
| c0002f89 | c0002f8a | c0002f8b | c0002f8c | c0002f93 | c0002f9a |
| c0002f8f | c0002f90 | c0002f91 | c0002f92 | c0002f99 | c0002fa0 |
| c0002f95 | c0002f96 | c0002f97 | c0002f98 | c0002f9f | c0002fa6 |
| c0002f9b | c0002f9c | c0002f9d | c0002f9e | c0002fa5 | c0002fac |
| c0002fa1 | c0002fa2 | c0002fa3 | c0002fa4 | c0002fab | c0002fb2 |
| c0002fa7 | c0002fa8 | c0002fa9 | c0002faa | c0002fb1 | c0002fb8 |
| c0002fad | c0002fae | c0002faf | c0002fb0 | c0002fb7 | c0002fbe |
| c0002fb3 | c0002fb4 | c0002fb5 | c0002fb6 | c0002fbd | c0002fc4 |
| c0002fb9 | c0002fba | c0002fbb | c0002fbc | c0002fc3 | c0002fca |
| c0002fbf | c0002fc0 | c0002fc1 | c0002fc2 | c0002fc9 | c0002fd0 |
| c0002fc5 | c0002fc6 | c0002fc7 | c0002fc8 | c0002fcf | c0002fd6 |
| c0002fcb | c0002fcc | c0002fcd | c0002fce | c0002fd5 | c0002fdc |
| c0002fd1 | c0002fd2 | c0002fd3 | c0002fd4 | c0002fdb | c0002fe2 |
| c0002fd7 | c0002fd8 | c0002fd9 | c0002fda | c0002fe1 | c0002fe8 |

| | | | | | |
|---|---|---|---|---|---|
| c0002fdd | c0002fde | c0002fdf | c0002fe0 | c0002fe7 | c0002fee |
| c0002fe3 | c0002fe4 | c0002fe5 | c0002fe6 | c0002fed | c0002ff4 |
| c0002fe9 | c0002fea | c0002feb | c0002fec | c0002ff3 | c0002ffa |
| c0002fef | c0002ff0 | c0002ff1 | c0002ff2 | c0002ff9 | c001001f |
| c0002ff5 | c0002ff6 | c0002ff7 | c0002ff8 | c0002fff | c0010122 |
| c0002ffb | c0002ffc | c0002ffd | c0002ffe | c0010121 | c0010241 |
| c001003e | c0010060 | c0010119 | c0010120 | c0010240 | c0010247 |
| c0010188 | c0010189 | c001018a | c001018b | c0010246 | c0010297 |
| c0010242 | c0010243 | c0010244 | c0010245 | c0010296 | c0010419 |
| c0010290 | c0010292 | c0010293 | c0010294 | c0010418 | c001041f |
| c0010404 | c0010411 | c0010412 | c0010417 | c001041e | c0010425 |
| c001041a | c001041b | c001041c | c001041d | c0010424 | c001042b |
| c0010420 | c0010421 | c0010422 | c0010423 | c001042a | c0010431 |
| c0010426 | c0010427 | c0010428 | c0010429 | c0010430 | c0010437 |
| c001042c | c001042d | c001042e | c001042f | c0010436 | c001043d |
| c0010432 | c0010433 | c0010434 | c0010435 | c001043c | c0010443 |
| c0010438 | c0010439 | c001043a | c001043b | c0010442 | c0010449 |
| c001043e | c001043f | c0010440 | c0010441 | c0010448 | c001044f |
| c0010444 | c0010445 | c0010446 | c0010447 | c001044e | c0010455 |
| c001044a | c001044b | c001044c | c001044d | c0010454 | c001045b |
| c0010450 | c0010451 | c0010452 | c0010453 | c001045a | c0010461 |
| c0010456 | c0010457 | c0010458 | c0010459 | c0010460 | c0010467 |
| c001045c | c001045d | c001045e | c001045f | c0010466 | c001046d |
| c0010462 | c0010463 | c0010464 | c0010465 | c001046c | c0010473 |
| c0010468 | c0010469 | c001046a | c001046b | c0010472 | c0010479 |
| c001046e | c001046f | c0010470 | c0010471 | c0010478 | c001047f |
| c0010474 | c0010475 | c0010476 | c0010477 | c001047e | c0010485 |
| c001047a | c001047b | c001047c | c001047d | c0010484 | c001048b |
| c0010480 | c0010481 | c0010482 | c0010483 | c001048a | c0010491 |
| c0010486 | c0010487 | c0010488 | c0010489 | c0010490 | c0010497 |
| c001048c | c001048d | c001048e | c001048f | c0010496 | c001049d |
| c0010492 | c0010493 | c0010494 | c0010495 | c001049c | c00104a3 |
| c0010498 | c0010499 | c001049a | c001049b | c00104a2 | c00104a9 |
| c001049e | c001049f | c00104a0 | c00104a1 | c00104a8 | c00104af |
| c00104a4 | c00104a5 | c00104a6 | c00104a7 | c00104ae | c00104b5 |
| c00104aa | c00104ab | c00104ac | c00104ad | c00104b4 | c00104bb |
| c00104b0 | c00104b1 | c00104b2 | c00104b3 | c00104ba | c00104c1 |
| c00104b6 | c00104b7 | c00104b8 | c00104b9 | c00104c0 | c00104c7 |
| c00104bc | c00104bd | c00104be | c00104bf | c00104c6 | c00104cd |
| c00104c2 | c00104c3 | c00104c4 | c00104c5 | c00104cc | c00104d3 |
| c00104c8 | c00104c9 | c00104ca | c00104cb | c00104d2 | c00104d9 |
| c00104ce | c00104cf | c00104d0 | c00104d1 | c00104d8 | c00104df |
| c00104d4 | c00104d5 | c00104d6 | c00104d7 | c00104de | c00104e5 |

| | | | | | |
|---|---|---|---|---|---|
| c00104da | c00104db | c00104dc | c00104dd | c00104e4 | c00104eb |
| c00104e0 | c00104e1 | c00104e2 | c00104e3 | c00104ea | c00104f1 |
| c00104e6 | c00104e7 | c00104e8 | c00104e9 | c00104f0 | c00104f7 |
| c00104ec | c00104ed | c00104ee | c00104ef | c00104f6 | c00104fd |
| c00104f2 | c00104f3 | c00104f4 | c00104f5 | c00104fc | c0010503 |
| c00104f8 | c00104f9 | c00104fa | c00104fb | c0010502 | c0010509 |
| c00104fe | c00104ff | c0010500 | c0010501 | c0010508 | c001050f |
| c0010504 | c0010505 | c0010506 | c0010507 | c001050e | c0010515 |
| c001050a | c001050b | c001050c | c001050d | c0010514 | c001051b |
| c0010510 | c0010511 | c0010512 | c0010513 | c001051a | c0010521 |
| c0010516 | c0010517 | c0010518 | c0010519 | c0010520 | c0010527 |
| c001051c | c001051d | c001051e | c001051f | c0010526 | c001052d |
| c0010522 | c0010523 | c0010524 | c0010525 | c001052c | c0010533 |
| c0010528 | c0010529 | c001052a | c001052b | c0010532 | c0010539 |
| c001052e | c001052f | c0010530 | c0010531 | c0010538 | c001053f |
| c0010534 | c0010535 | c0010536 | c0010537 | c001053e | c0010545 |
| c001053a | c001053b | c001053c | c001053d | c0010544 | c001054b |
| c0010540 | c0010541 | c0010542 | c0010543 | c001054a | c0010551 |
| c0010546 | c0010547 | c0010548 | c0010549 | c0010550 | c0010557 |
| c001054c | c001054d | c001054e | c001054f | c0010556 | c001055d |
| c0010552 | c0010553 | c0010554 | c0010555 | c001055c | c0010563 |
| c0010558 | c0010559 | c001055a | c001055b | c0010562 | c0010569 |
| c001055e | c001055f | c0010560 | c0010561 | c0010568 | c001056f |
| c0010564 | c0010565 | c0010566 | c0010567 | c001056e | c0010575 |
| c001056a | c001056b | c001056c | c001056d | c0010574 | c001057b |
| c0010570 | c0010571 | c0010572 | c0010573 | c001057a | c0010581 |
| c0010576 | c0010577 | c0010578 | c0010579 | c0010580 | c0010587 |
| c001057c | c001057d | c001057e | c001057f | c0010586 | c001058d |
| c0010582 | c0010583 | c0010584 | c0010585 | c001058c | c0010593 |
| c0010588 | c0010589 | c001058a | c001058b | c0010592 | c0010599 |
| c001058e | c001058f | c0010590 | c0010591 | c0010598 | c001059f |
| c0010594 | c0010595 | c0010596 | c0010597 | c001059e | c00105a5 |
| c001059a | c001059b | c001059c | c001059d | c00105a4 | c00105ab |
| c00105a0 | c00105a1 | c00105a2 | c00105a3 | c00105aa | c00105b1 |
| c00105a6 | c00105a7 | c00105a8 | c00105a9 | c00105b0 | c00105b7 |
| c00105ac | c00105ad | c00105ae | c00105af | c00105b6 | c00105bd |
| c00105b2 | c00105b3 | c00105b4 | c00105b5 | c00105bc | c00105c3 |
| c00105b8 | c00105b9 | c00105ba | c00105bb | c00105c2 | c00105c9 |
| c00105be | c00105bf | c00105c0 | c00105c1 | c00105c8 | c00105cf |
| c00105c4 | c00105c5 | c00105c6 | c00105c7 | c00105ce | c00105d5 |
| c00105ca | c00105cb | c00105cc | c00105cd | c00105d4 | c00105db |
| c00105d0 | c00105d1 | c00105d2 | c00105d3 | c00105da | c00105e1 |
| c00105d6 | c00105d7 | c00105d8 | c00105d9 | c00105e0 | c00105e7 |

| | | | | | |
|---|---|---|---|---|---|
| c00105dc | c00105dd | c00105de | c00105df | c00105e6 | c00105ed |
| c00105e2 | c00105e3 | c00105e4 | c00105e5 | c00105ec | c00105f3 |
| c00105e8 | c00105e9 | c00105ea | c00105eb | c00105f2 | c00105f9 |
| c00105ee | c00105ef | c00105f0 | c00105f1 | c00105f8 | c00105ff |
| c00105f4 | c00105f5 | c00105f6 | c00105f7 | c00105fe | c0010605 |
| c00105fa | c00105fb | c00105fc | c00105fd | c0010604 | c001060b |
| c0010600 | c0010601 | c0010602 | c0010603 | c001060a | c0010611 |
| c0010606 | c0010607 | c0010608 | c0010609 | c0010610 | c0010617 |
| c001060c | c001060d | c001060e | c001060f | c0010616 | c001061d |
| c0010612 | c0010613 | c0010614 | c0010615 | c001061c | c0010623 |
| c0010618 | c0010619 | c001061a | c001061b | c0010622 | c0010629 |
| c001061e | c001061f | c0010620 | c0010621 | c0010628 | c001062f |
| c0010624 | c0010625 | c0010626 | c0010627 | c001062e | c0010635 |
| c001062a | c001062b | c001062c | c001062d | c0010634 | c001063b |
| c0010630 | c0010631 | c0010632 | c0010633 | c001063a | c0010641 |
| c0010636 | c0010637 | c0010638 | c0010639 | c0010640 | c0010647 |
| c001063c | c001063d | c001063e | c001063f | c0010646 | c001064d |
| c0010642 | c0010643 | c0010644 | c0010645 | c001064c | c0010653 |
| c0010648 | c0010649 | c001064a | c001064b | c0010652 | c0010659 |
| c001064e | c001064f | c0010650 | c0010651 | c0010658 | c001065f |
| c0010654 | c0010655 | c0010656 | c0010657 | c001065e | c0010665 |
| c001065a | c001065b | c001065c | c001065d | c0010664 | c001066b |
| c0010660 | c0010661 | c0010662 | c0010663 | c001066a | c0010671 |
| c0010666 | c0010667 | c0010668 | c0010669 | c0010670 | c0010677 |
| c001066c | c001066d | c001066e | c001066f | c0010676 | c001067d |
| c0010672 | c0010673 | c0010674 | c0010675 | c001067c | c0010683 |
| c0010678 | c0010679 | c001067a | c001067b | c0010682 | c0010689 |
| c001067e | c001067f | c0010680 | c0010681 | c0010688 | c001068f |
| c0010684 | c0010685 | c0010686 | c0010687 | c001068e | c0010695 |
| c001068a | c001068b | c001068c | c001068d | c0010694 | c001069b |
| c0010690 | c0010691 | c0010692 | c0010693 | c001069a | c00106a1 |
| c0010696 | c0010697 | c0010698 | c0010699 | c00106a0 | c00106a7 |
| c001069c | c001069d | c001069e | c001069f | c00106a6 | c00106ad |
| c00106a2 | c00106a3 | c00106a4 | c00106a5 | c00106ac | c00106b3 |
| c00106a8 | c00106a9 | c00106aa | c00106ab | c00106b2 | c00106b9 |
| c00106ae | c00106af | c00106b0 | c00106b1 | c00106b8 | c00106bf |
| c00106b4 | c00106b5 | c00106b6 | c00106b7 | c00106be | c00106c5 |
| c00106ba | c00106bb | c00106bc | c00106bd | c00106c4 | c00106cb |
| c00106c0 | c00106c1 | c00106c2 | c00106c3 | c00106ca | c00106d1 |
| c00106c6 | c00106c7 | c00106c8 | c00106c9 | c00106d0 | c00106d7 |
| c00106cc | c00106cd | c00106ce | c00106cf | c00106d6 | c00106dd |
| c00106d2 | c00106d3 | c00106d4 | c00106d5 | c00106dc | c00106e3 |
| c00106d8 | c00106d9 | c00106da | c00106db | c00106e2 | c00106e9 |

| | | | | | |
|---|---|---|---|---|---|
| c00106de | c00106df | c00106e0 | c00106e1 | c00106e8 | c00106ef |
| c00106e4 | c00106e5 | c00106e6 | c00106e7 | c00106ee | c00106f5 |
| c00106ea | c00106eb | c00106ec | c00106ed | c00106f4 | c00106fb |
| c00106f0 | c00106f1 | c00106f2 | c00106f3 | c00106fa | c0011001 |
| c00106f6 | c00106f7 | c00106f8 | c00106f9 | c0011000 | c001100b |
| c00106fc | c00106fd | c00106fe | c00106ff | c001100a | c0011014 |
| c0011006 | c0011007 | c0011008 | c0011009 | c0011011 | c0011021 |
| c001100c | c001100e | c001100f | c0011010 | c0011020 | c0011029 |
| c0011015 | c0011016 | c0011017 | c0011018 | c0011028 | c001102f |
| c0011022 | c0011024 | c0011025 | c0011026 | c001102e | c0011077 |
| c001102a | c001102b | c001102c | c001102d | c0011076 | c0011095 |
| c0011041 | c0011042 | c0011074 | c0011075 | c0011094 | |
| c0011078 | c0011083 | c0011092 | c0011093 | 413 | |
| c0011096 | c0011097 | c00110a2 | 412 | 425 | |

### 10.1.2 Intel Core i7-6700K

| | | | | | |
|---|---|---|---|---|---|
| 20 | 21 | 3e | 104 | 140 | 1da |
| 33 | 35 | 103 | 13d | 1c6 | 1f5 |
| 95 | 102 | 13a | 1aa | 1f4 | 302 |
| 11f | 121 | 1a8 | 1f0 | 2f5 | 4e3 |
| 19d | 1a1 | 1e0 | 2f4 | 4e2 | 609 |
| 1db | 1dc | 2e7 | 4e0 | 608 | 622 |
| 1fb | 2e0 | 397 | 607 | 621 | 634 |
| 305 | 393 | 603 | 620 | 633 | 704 |
| 503 | 601 | 61d | 632 | 703 | 714 |
| 615 | 618 | 631 | 702 | 713 | 724 |
| 623 | 630 | 637 | 712 | 723 | 734 |
| 635 | 636 | 709 | 722 | 733 | 742 |
| 705 | 708 | 719 | 732 | 741 | 748 |
| 715 | 718 | 729 | 740 | 747 | |
| 725 | 728 | 739 | 746 | 2e | |
| 735 | 738 | 745 | 2a | 80 | |
| 743 | 744 | 23 | 7a | 118 | |
| 749 | 22 | 59 | 110 | 150 | |

### 10.1.3 Intel Core i7-8700K

| | | | | | |
|---|---|---|---|---|---|
| 20 | 21 | 22 | 23 | 2a | 80 |
| 33 | 35 | 3e | 59 | 7a | 110 |
| 95 | 102 | 103 | 104 | 10f | 140 |
| 118 | 11f | 121 | 13a | 13d | 1c6 |
| 150 | 19d | 1a1 | 1a8 | 1aa | 1f4 |
| 1da | 1db | 1dc | 1e0 | 1f0 | 2f5 |
| 1f5 | 1fb | 2e0 | 2e7 | 2f4 | 4e2 |

| | | | | | |
|---|---|---|---|---|---|
| 302 | 305 | 393 | 397 | 4e0 | 608 |
| 4e3 | 503 | 601 | 603 | 607 | 621 |
| 609 | 615 | 618 | 61d | 620 | 633 |
| 622 | 623 | 630 | 631 | 632 | 703 |
| 634 | 635 | 636 | 637 | 702 | 713 |
| 704 | 705 | 708 | 709 | 712 | 723 |
| 714 | 715 | 718 | 719 | 722 | 733 |
| 724 | 725 | 728 | 729 | 732 | 741 |
| 734 | 735 | 738 | 739 | 740 | 747 |
| 742 | 743 | 744 | 745 | 746 | 753 |
| 748 | 749 | 750 | 751 | 752 | 759 |
| 754 | 755 | 756 | 757 | 758 | 765 |
| 760 | 761 | 762 | 763 | 764 | |
| 766 | 767 | 768 | 769 | 2e | |

### 10.1.4 Intel Core i9-9900K

| | | | | | |
|---|---|---|---|---|---|
| 20 | 21 | 22 | 59 | 10f | 140 |
| 33 | 35 | 3e | 104 | 13d | 1c6 |
| 95 | 102 | 103 | 13a | 1aa | 1f4 |
| 118 | 11f | 121 | 1a8 | 1f0 | 2f5 |
| 150 | 19d | 1a1 | 1e0 | 2f4 | 4e2 |
| 1da | 1db | 1dc | 2e7 | 4e0 | 608 |
| 1f5 | 1fb | 2e0 | 397 | 607 | 621 |
| 302 | 305 | 393 | 603 | 620 | 633 |
| 4e3 | 503 | 601 | 61d | 632 | 703 |
| 609 | 615 | 618 | 631 | 702 | 713 |
| 622 | 623 | 630 | 637 | 712 | 723 |
| 634 | 635 | 636 | 709 | 722 | 733 |
| 704 | 705 | 708 | 719 | 732 | 741 |
| 714 | 715 | 718 | 729 | 740 | 747 |
| 724 | 725 | 728 | 739 | 746 | 753 |
| 734 | 735 | 738 | 745 | 752 | 759 |
| 742 | 743 | 744 | 751 | 758 | 765 |
| 748 | 749 | 750 | 757 | 764 | f71 |
| 754 | 755 | 756 | 763 | f70 | f77 |
| 760 | 761 | 762 | 769 | f76 | |
| 766 | 767 | 768 | f75 | 2e | |
| f72 | f73 | f74 | 2a | 80 | |
| f78 | f79 | 23 | 7a | 110 | |

### 10.1.5 Intel Xeon Silver 4208

| | | | | | |
|---|---|---|---|---|---|
| 20 | 21 | 35 | 56 | 6b | 118 |
| 31 | 33 | 55 | 67 | 110 | 13a |

| | | | | | |
|---|---|---|---|---|---|
| 53 | 54 | 64 | 108 | 12a | 154 |
| 5f | 63 | 102 | 129 | 153 | 19d |
| af | e3 | 128 | 152 | 17f | 1ae |
| 11f | 122 | 150 | 178 | 1aa | 1e1 |
| 13d | 140 | 157 | 1a8 | 1e0 | 2e0 |
| 155 | 156 | 1a5 | 1dc | 1fb | 603 |
| 1a1 | 1a3 | 1db | 1f7 | 601 | 61c |
| 1c6 | 1da | 1f6 | 503 | 618 | 637 |
| 1e2 | 1f0 | 305 | 615 | 623 | 709 |
| 2e7 | 302 | 609 | 622 | 708 | 713 |
| 607 | 608 | 621 | 705 | 712 | 71b |
| 61d | 620 | 704 | 70d | 71a | a01 |
| 702 | 703 | 70c | 719 | a00 | a44 |
| 70a | 70b | 718 | 793 | a43 | a4b |
| 714 | 715 | 790 | a42 | a4a | a51 |
| 71c | 71d | a41 | a49 | a50 | a5b |
| a02 | a40 | a48 | a4f | a5a | a63 |
| a45 | a47 | a4e | a59 | a62 | a6a |
| a4c | a4d | a58 | a61 | a69 | a70 |
| a52 | a53 | a60 | a68 | a6f | a7a |
| a5c | a5f | a67 | a6e | a79 | a82 |
| a64 | a65 | a6d | a78 | a81 | a89 |
| a6b | a6c | a73 | a80 | a88 | a8f |
| a71 | a72 | a7f | a87 | a8e | a99 |
| a7b | a7c | a85 | a8d | a98 | aa1 |
| a83 | a84 | a8c | a93 | aa0 | aa8 |
| a8a | a8b | a92 | a9f | aa7 | aae |
| a90 | a91 | a9c | aa5 | aad | ab8 |
| a9a | a9b | aa4 | aac | ab3 | ac0 |
| aa2 | aa3 | aab | ab2 | abf | ac7 |
| aa9 | aaa | ab1 | abc | ac5 | acd |
| aaf | ab0 | abb | ac4 | acc | ad3 |
| ab9 | aba | ac3 | acb | ad2 | adf |
| ac1 | ac2 | aca | ad1 | adc | ae5 |
| ac8 | ac9 | ad0 | adb | ae4 | aec |
| ace | acf | ada | ae3 | aeb | af2 |
| ad8 | ad9 | ae2 | aea | af1 | afc |
| ae0 | ae1 | ae9 | af0 | afb | b04 |
| ae7 | ae8 | aef | afa | b03 | b0a |
| aed | aee | af9 | b02 | b09 | b10 |
| af3 | af8 | b01 | b08 | b0f | b16 |
| aff | b00 | b07 | b0e | b15 | b1c |
| b05 | b06 | b0d | b14 | b1b | b22 |

| | | | | | |
|---|---|---|---|---|---|
| b0b | b0c | b13 | b1a | b21 | b28 |
| b11 | b12 | b19 | b20 | b27 | b2e |
| b17 | b18 | b1f | b26 | b2d | b34 |
| b1d | b1e | b25 | b2c | b33 | b3a |
| b23 | b24 | b2b | b32 | b39 | b40 |
| b29 | b2a | b31 | b38 | b3f | b46 |
| b2f | b30 | b37 | b3e | b45 | b4c |
| b35 | b36 | b3d | b44 | b4b | b52 |
| b3b | b3c | b43 | b4a | b51 | b58 |
| b41 | b42 | b49 | b50 | b57 | b5e |
| b47 | b48 | b4f | b56 | b5d | e04 |
| b4d | b4e | b55 | b5c | e03 | e0a |
| b53 | b54 | b5b | e00 | e09 | e14 |
| b59 | b5a | c8c | e08 | e13 | e1a |
| b5f | c8b | e07 | e12 | e19 | e24 |
| e05 | e06 | e11 | e18 | e23 | e2a |
| e0b | e10 | e17 | e22 | e29 | e34 |
| e15 | e16 | e21 | e28 | e33 | e3a |
| e1b | e20 | e27 | e32 | e39 | e44 |
| e25 | e26 | e31 | e38 | e43 | e4a |
| e2b | e30 | e37 | e42 | e49 | e54 |
| e35 | e36 | e41 | e48 | e53 | e5a |
| e3b | e40 | e47 | e52 | e59 | e64 |
| e45 | e46 | e51 | e58 | e63 | e6a |
| e4b | e50 | e57 | e62 | e69 | e74 |
| e55 | e56 | e61 | e68 | e73 | e7a |
| e5b | e60 | e67 | e72 | e79 | e84 |
| e65 | e66 | e71 | e78 | e83 | e8a |
| e6b | e70 | e77 | e82 | e89 | e94 |
| e75 | e76 | e81 | e88 | e93 | e9a |
| e7b | e80 | e87 | e92 | e99 | ea4 |
| e85 | e86 | e91 | e98 | ea3 | eaa |
| e8b | e90 | e97 | ea2 | ea9 | eb4 |
| e95 | e96 | ea1 | ea8 | eb3 | eba |
| e9b | ea0 | ea7 | eb2 | eb9 | ec4 |
| ea5 | ea6 | eb1 | eb8 | ec3 | eca |
| eab | eb0 | eb7 | ec2 | ec9 | ed4 |
| eb5 | eb6 | ec1 | ec8 | ed3 | eda |
| ebb | ec0 | ec7 | ed2 | ed9 | ee4 |
| ec5 | ec6 | ed1 | ed8 | ee3 | eea |
| ecb | ed0 | ed7 | ee2 | ee9 | ef4 |
| ed5 | ed6 | ee1 | ee8 | ef3 | efa |
| edb | ee0 | ee7 | ef2 | ef9 | f04 |

| | | | | | |
|---|---|---|---|---|---|
| ee5 | ee6 | ef1 | ef8 | f03 | f0a |
| eeb | ef0 | ef7 | f02 | f09 | f14 |
| ef5 | ef6 | f01 | f08 | f13 | f1a |
| efb | f00 | f07 | f12 | f19 | f24 |
| f05 | f06 | f11 | f18 | f23 | f2a |
| f0b | f10 | f17 | f22 | f29 | f34 |
| f15 | f16 | f21 | f28 | f33 | f3a |
| f1b | f20 | f27 | f32 | f39 | f44 |
| f25 | f26 | f31 | f38 | f43 | f4a |
| f2b | f30 | f37 | f42 | f49 | f54 |
| f35 | f36 | f41 | f48 | f53 | f5a |
| f3b | f40 | f47 | f52 | f59 | f64 |
| f45 | f46 | f51 | f58 | f63 | f6a |
| f4b | f50 | f57 | f62 | f69 | f74 |
| f55 | f56 | f61 | f68 | f73 | f7a |
| f5b | f60 | f67 | f72 | f79 | |
| f65 | f66 | f71 | f78 | 2e | |
| f6b | f70 | f77 | 2a | 4f | |
| f75 | f76 | 23 | 4e | 5e | |
| f7b | 22 | 3e | 59 | 80 | |

## 10.2 Found Undocumented Dynamic MSRs on Intel Xeon Silver 4208

| | | | | | |
|---|---|---|---|---|---|
| 1a1 | 1a3 | 637 | 71d | a45 | a65 |
| a85 | aa5 | b01 | b05 | b08 | b09 |
| b0a | b0b | b0c | b0d | b0e | b0f |
| b18 | b19 | b1a | b1b | b1c | b1d |
| b1e | b1f | b28 | b29 | b2a | b2b |
| b2c | b2d | b2e | b2f | b38 | b39 |
| b3a | b3b | b3c | b3d | b3e | b3f |

## 10.3 Correlations

### 10.3.1 Undocumented Dynamic MSR Correlations on Intel Xeon Silver 4208

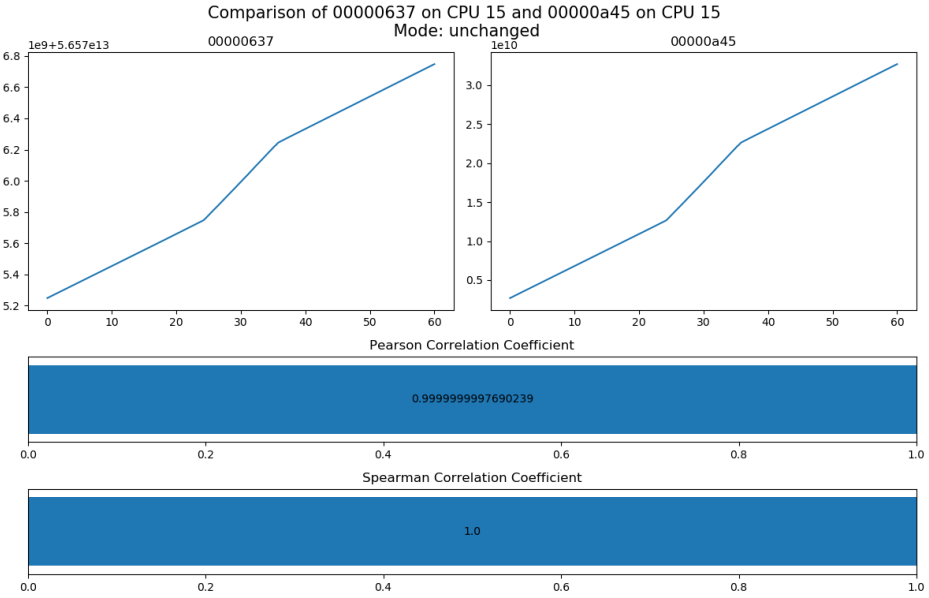



**Figure 10.1:** Correlation of MSR 0x637 and 0xa45.

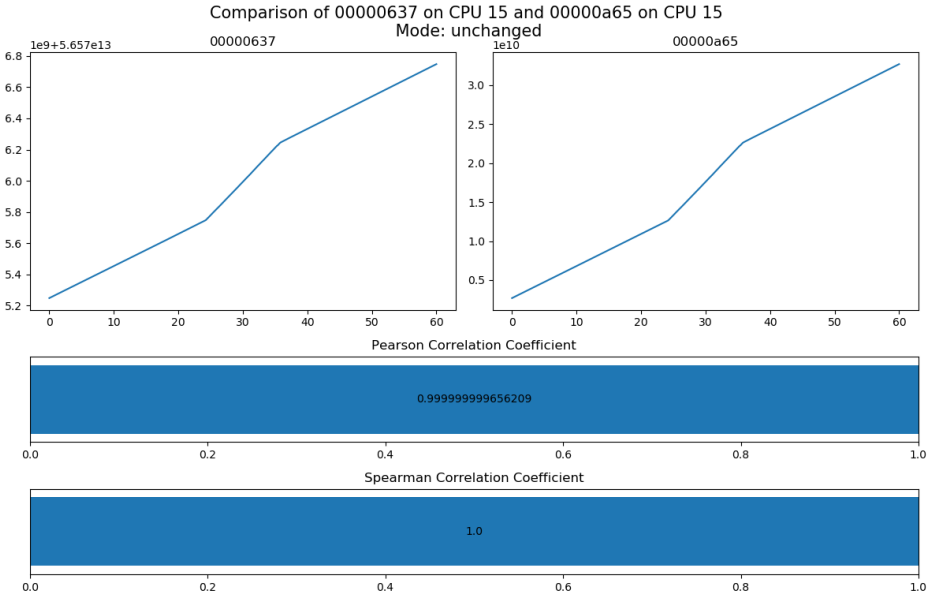



**Figure 10.2:** Correlation of MSR 0x637 and 0xa65.

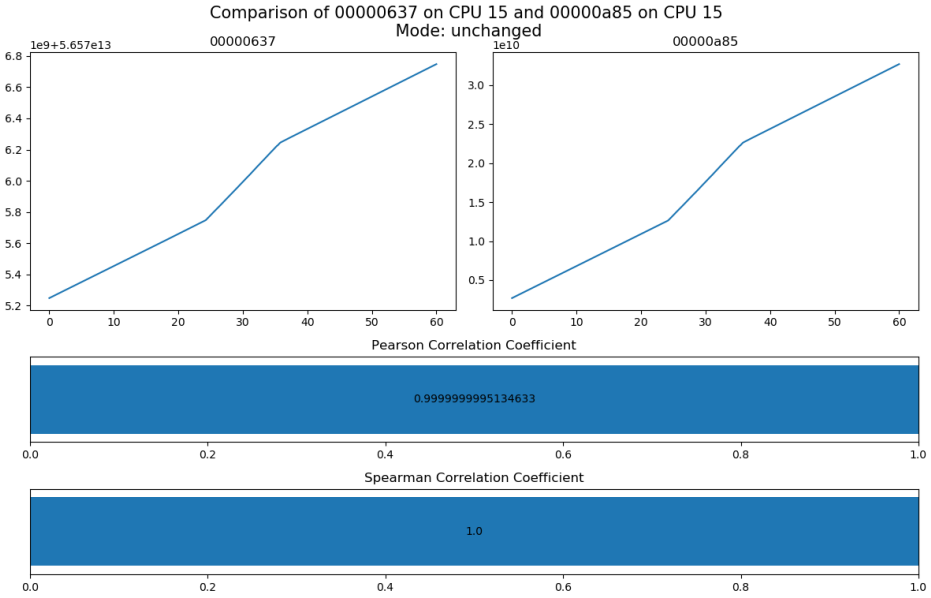



**Figure 10.3:** Correlation of MSR 0x637 and 0xa85.

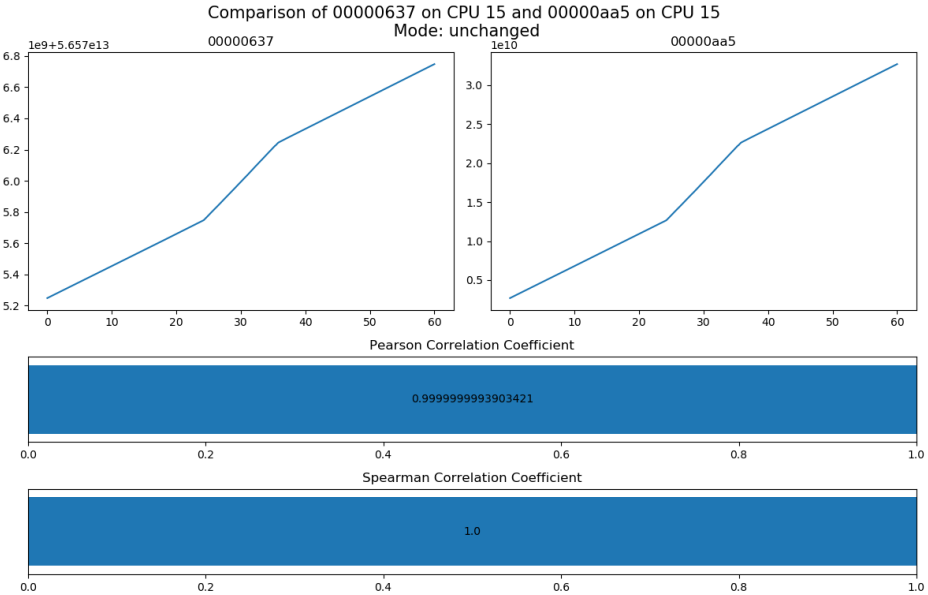



**Figure 10.4:** Correlation of MSR 0x637 and 0xaa5.

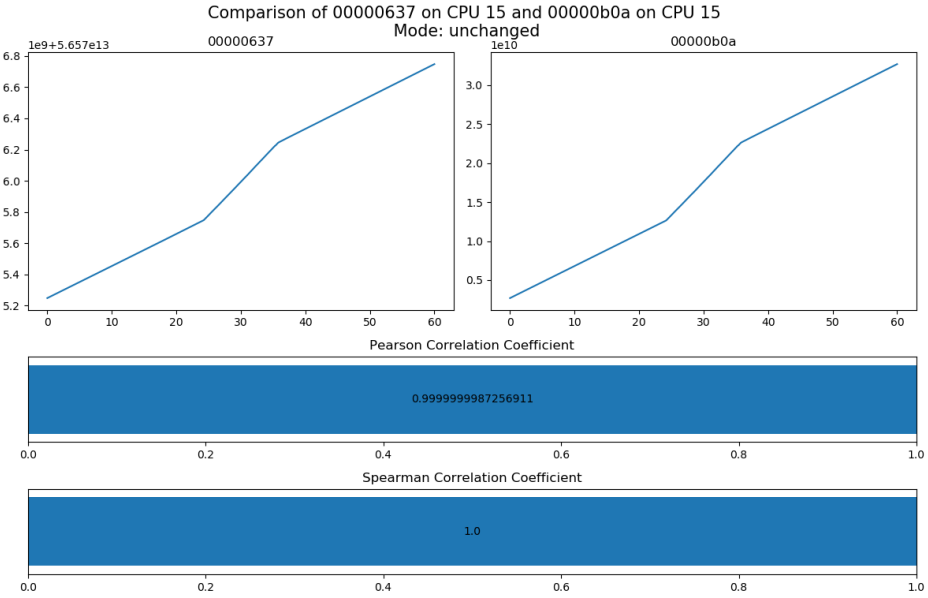



**Figure 10.5:** Correlation of MSR 0x637 and 0xb0a.

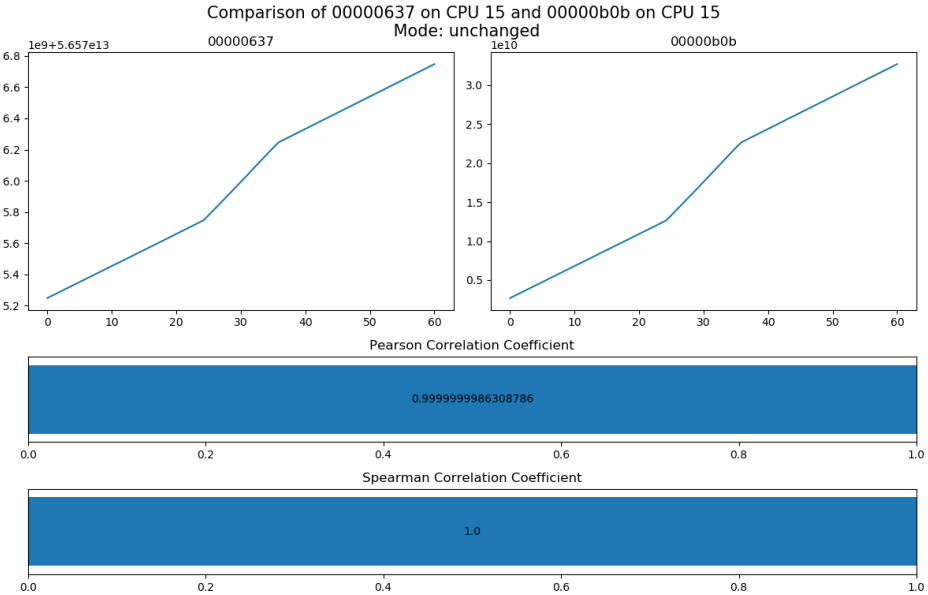



**Figure 10.6:** Correlation of MSR 0x637 and 0xb0b.

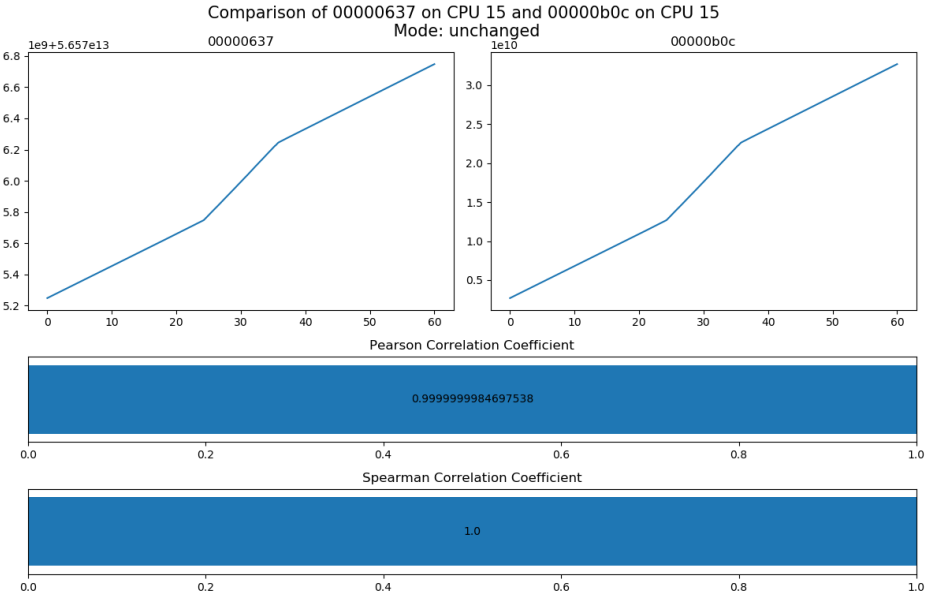



**Figure 10.7:** Correlation of MSR 0x637 and 0xb0c.

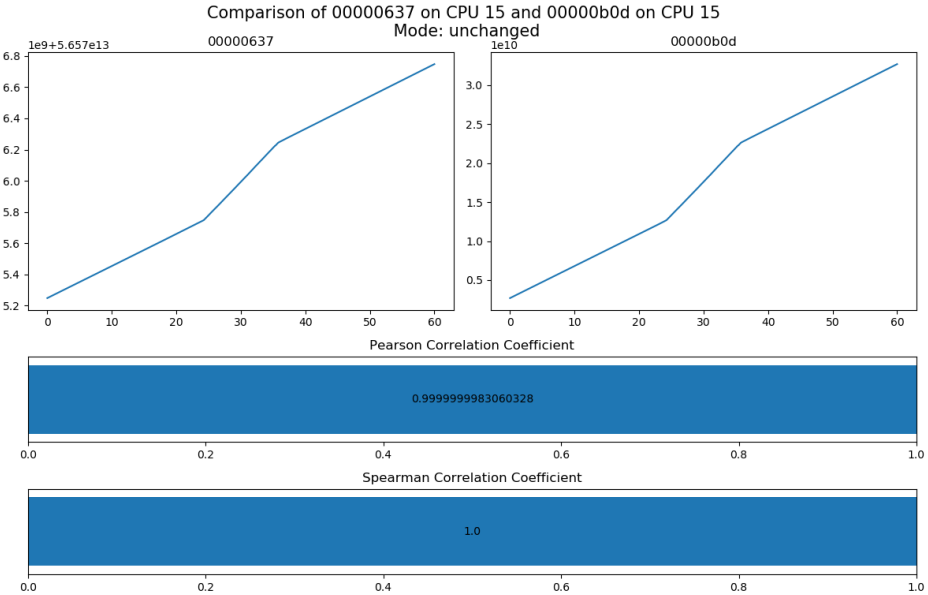



**Figure 10.8:** Correlation of MSR 0x637 and 0xb0d.

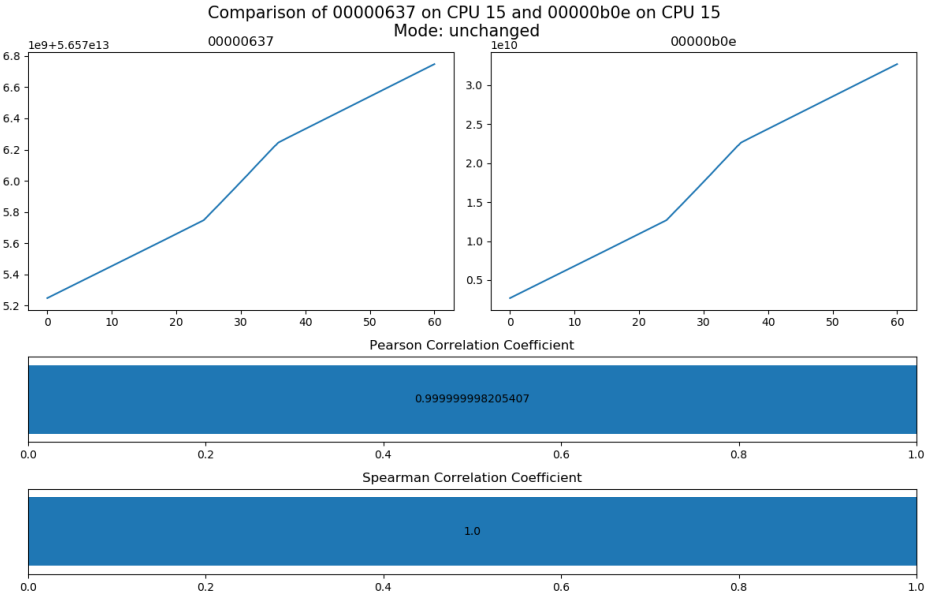



**Figure 10.9:** Correlation of MSR 0x637 and 0xb0e.

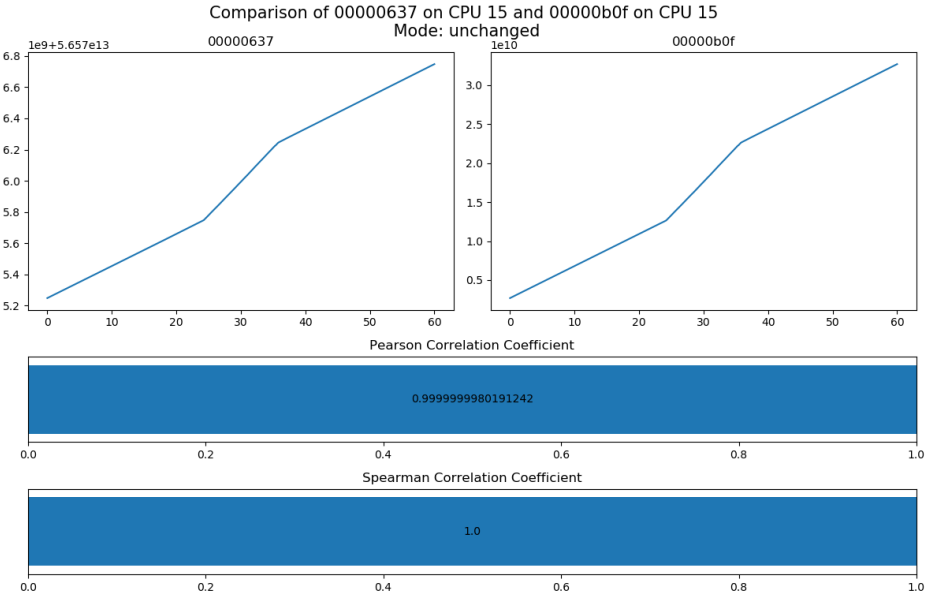



**Figure 10.10:** Correlation of MSR 0x637 and 0xb0f.

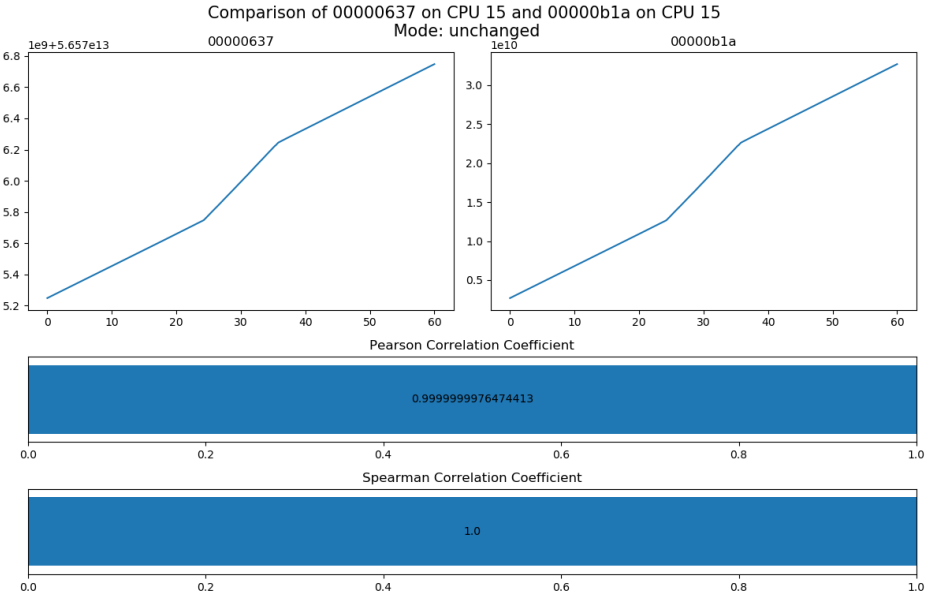



**Figure 10.11:** Correlation of MSR 0x637 and 0xb1a.

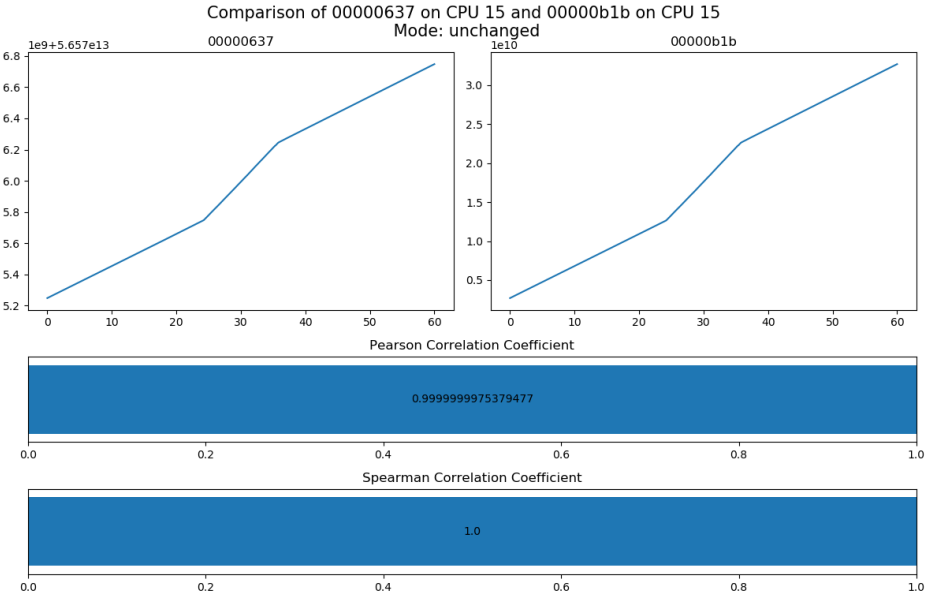



**Figure 10.12:** Correlation of MSR 0x637 and 0xb1b.

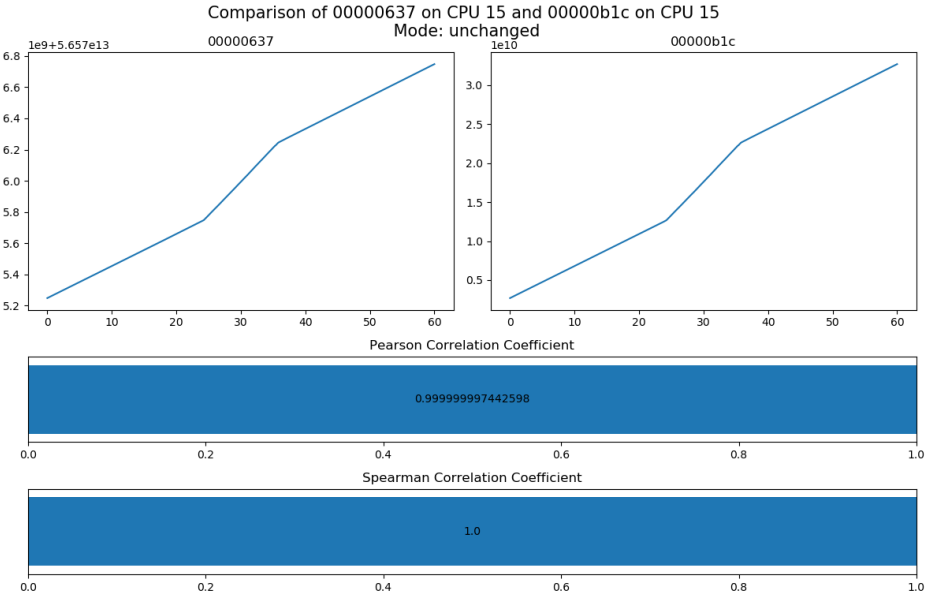



**Figure 10.13:** Correlation of MSR 0x637 and 0xb1c.

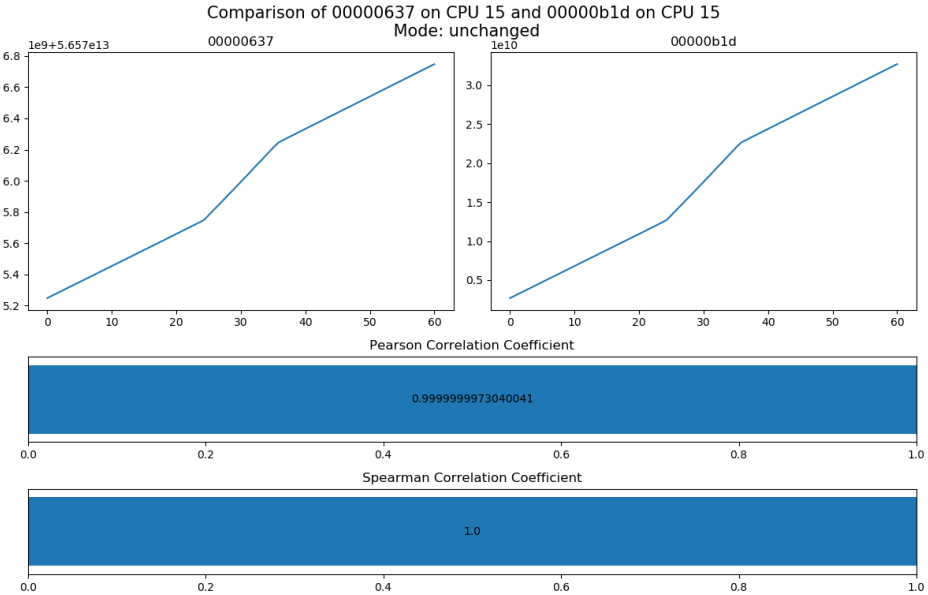



**Figure 10.14:** Correlation of MSR 0x637 and 0xb1d.

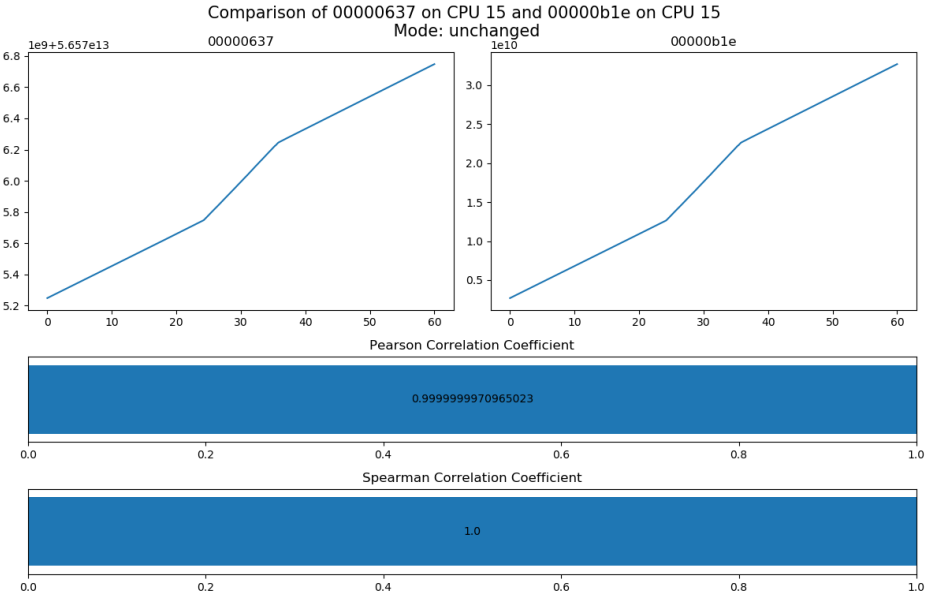



**Figure 10.15:** Correlation of MSR 0x637 and 0xb1e.

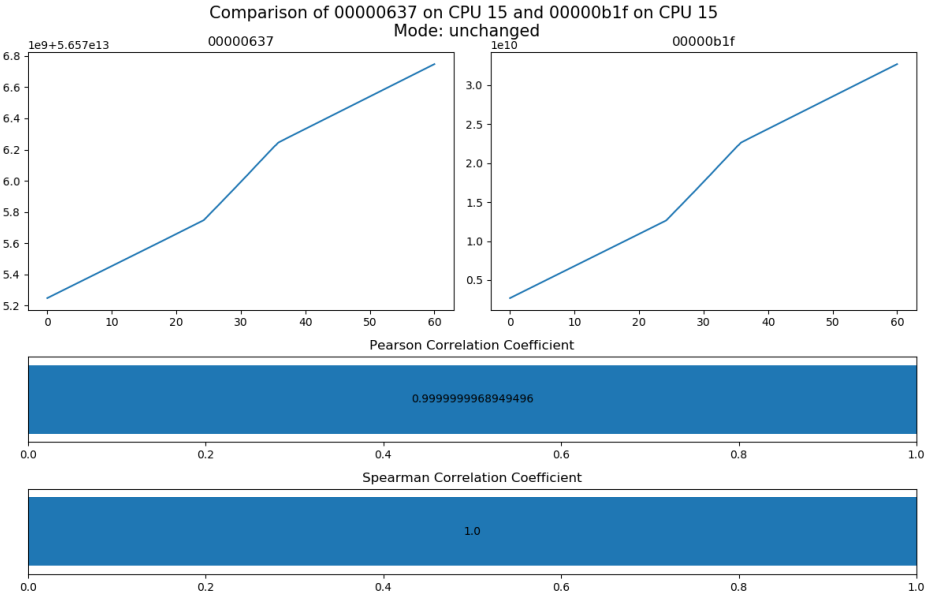



**Figure 10.16:** Correlation of MSR 0x637 and 0xb1f.

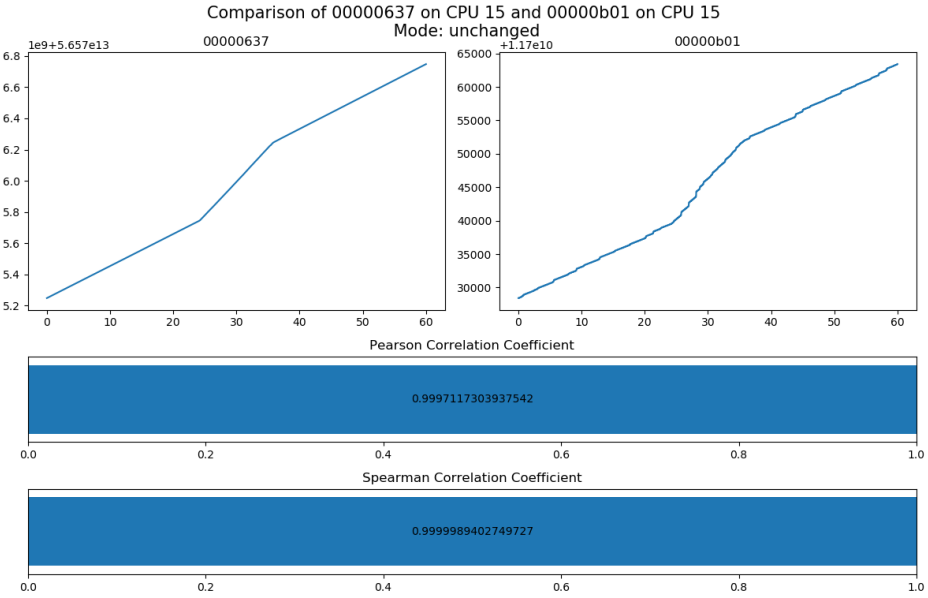



**Figure 10.17:** Correlation of MSR 0x637 and 0xb01.

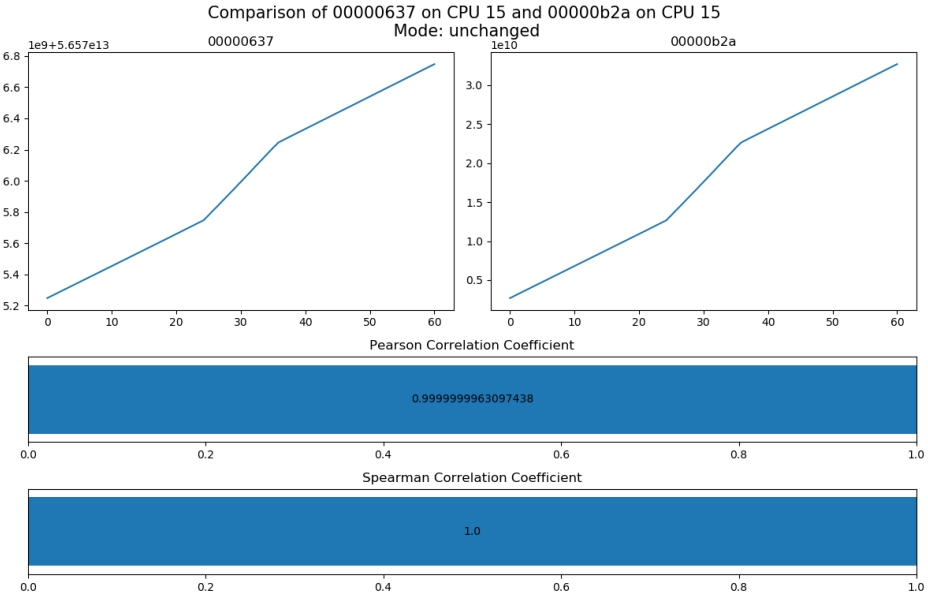



**Figure 10.18:** Correlation of MSR 0x637 and 0xb2a.

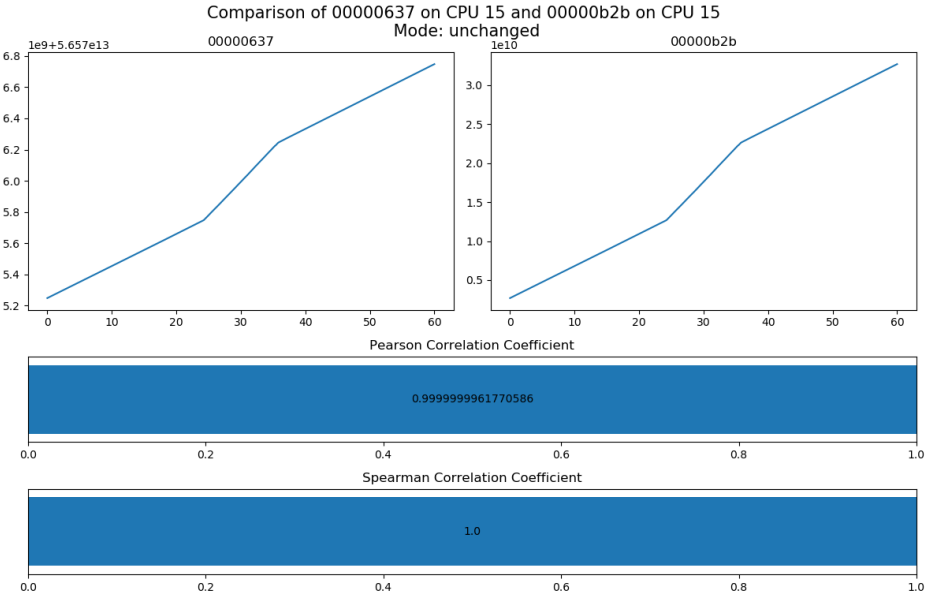



**Figure 10.19:** Correlation of MSR 0x637 and 0xb2b.

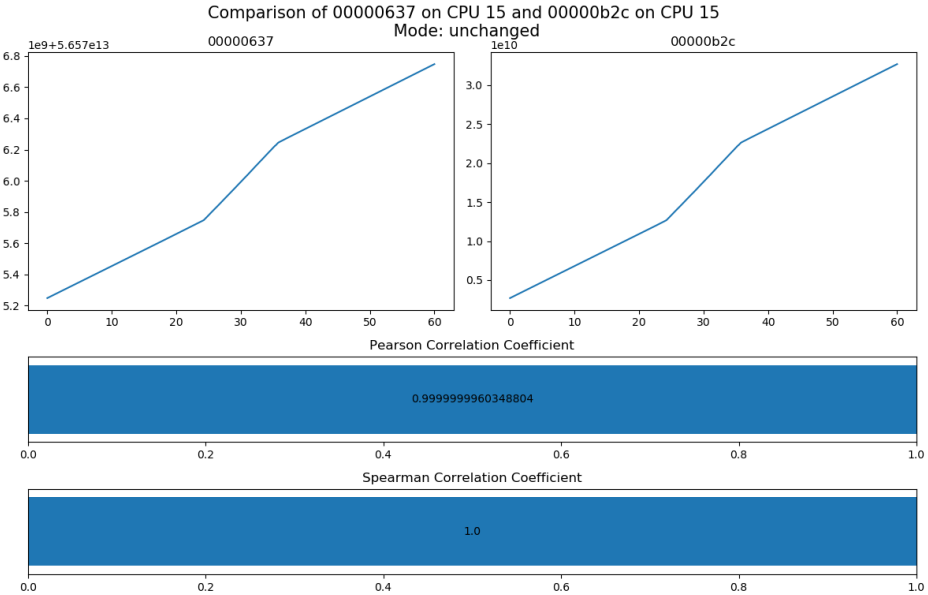



**Figure 10.20:** Correlation of MSR 0x637 and 0xb2c.

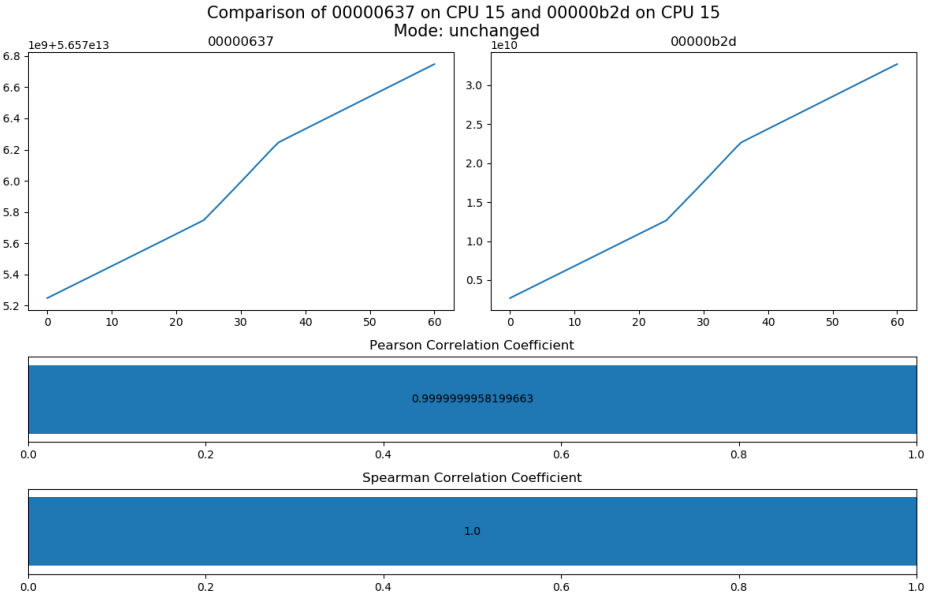



**Figure 10.21:** Correlation of MSR 0x637 and 0xb2d.

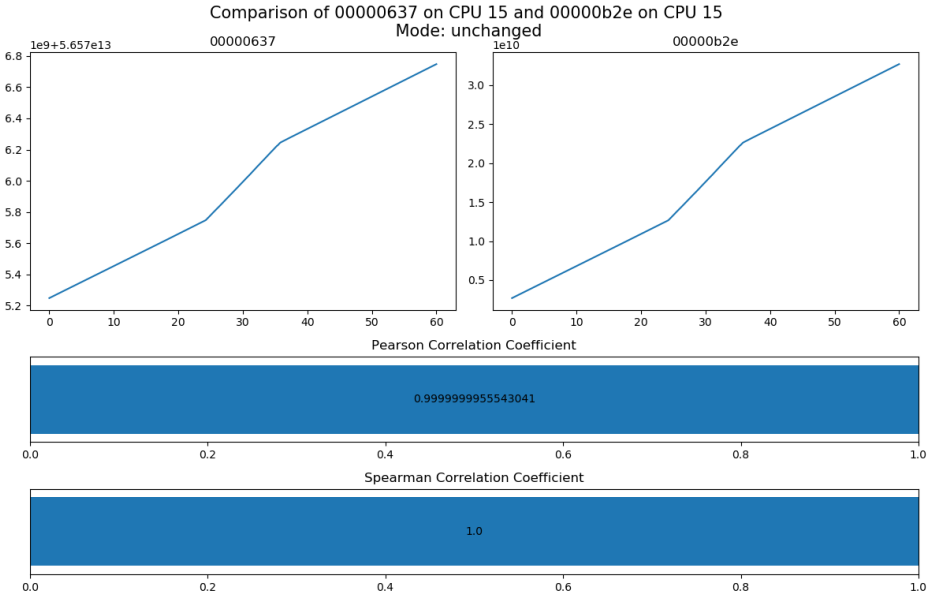



**Figure 10.22:** Correlation of MSR 0x637 and 0xb2e.

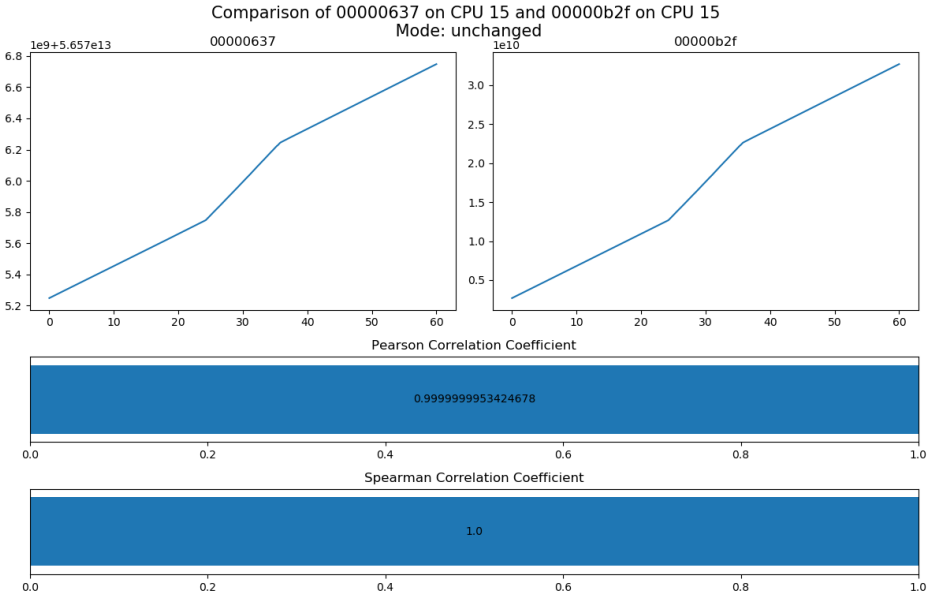



**Figure 10.23:** Correlation of MSR 0x637 and 0xb2f.

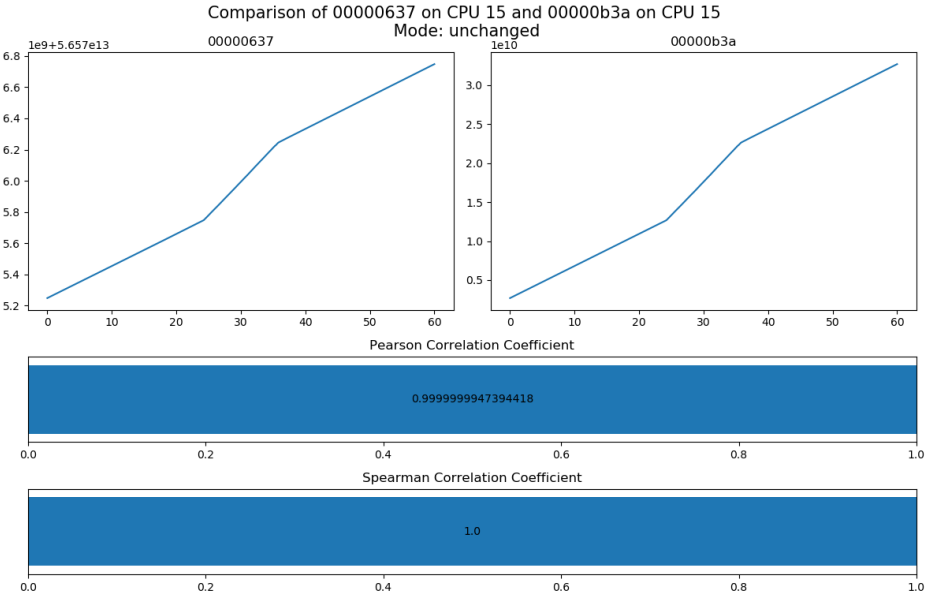



**Figure 10.24:** Correlation of MSR 0x637 and 0xb3a.

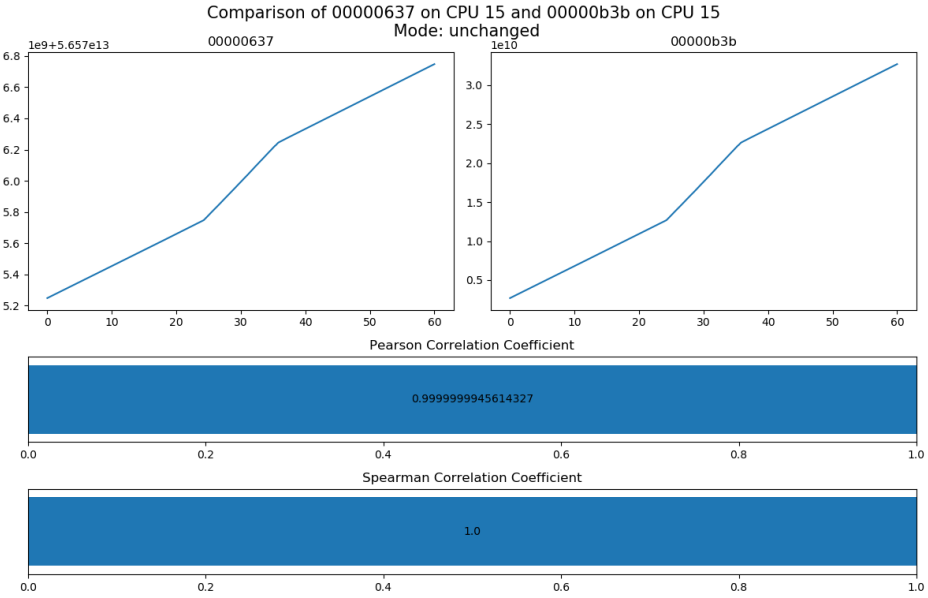



**Figure 10.25:** Correlation of MSR 0x637 and 0xb3b.

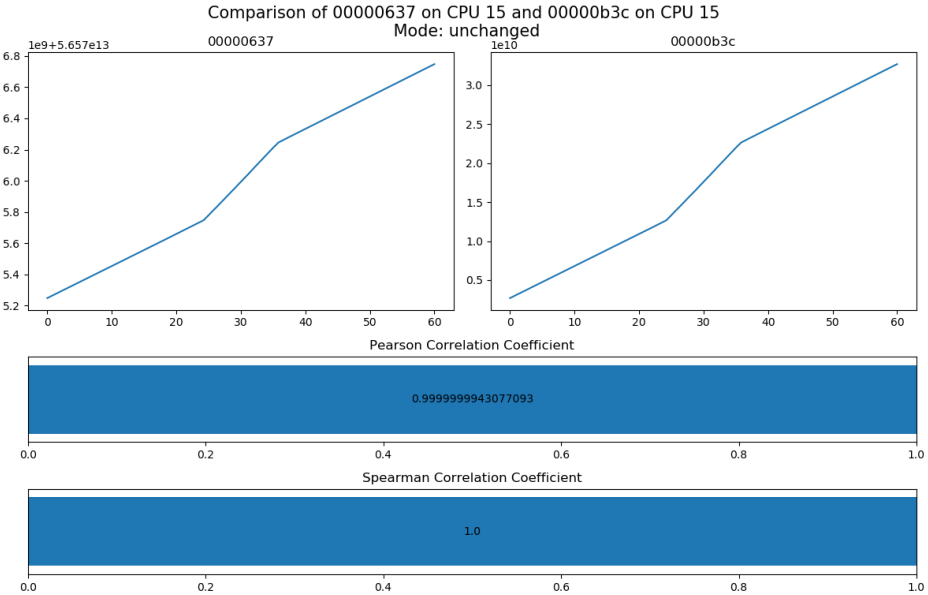



**Figure 10.26:** Correlation of MSR 0x637 and 0xb3c.

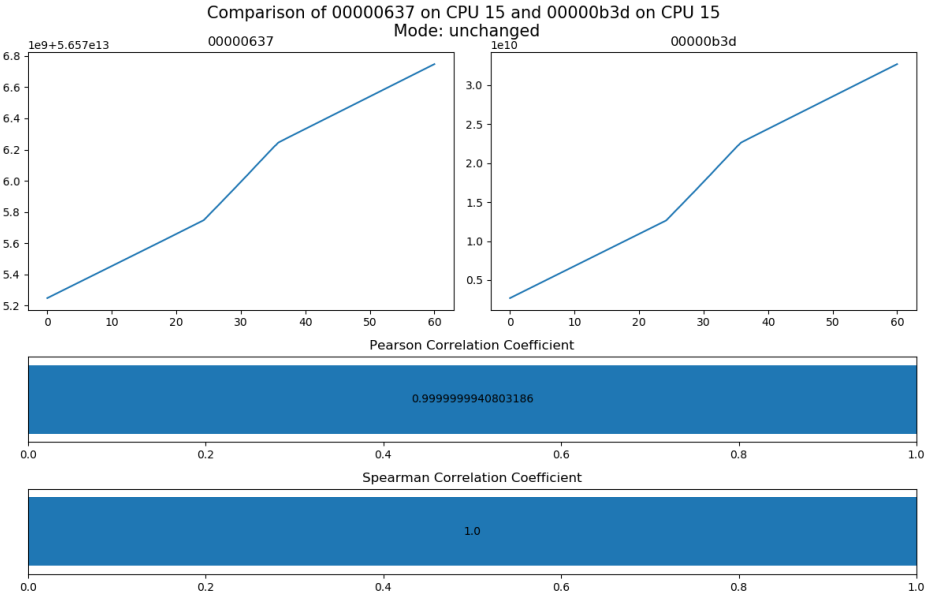



**Figure 10.27:** Correlation of MSR 0x637 and 0xb3d.

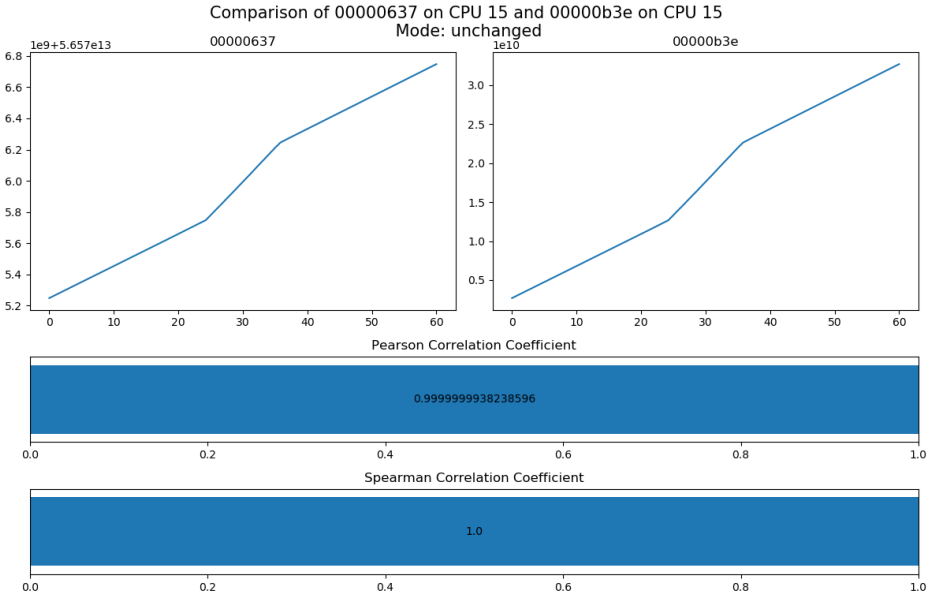



**Figure 10.28:** Correlation of MSR 0x637 and 0xb3e.

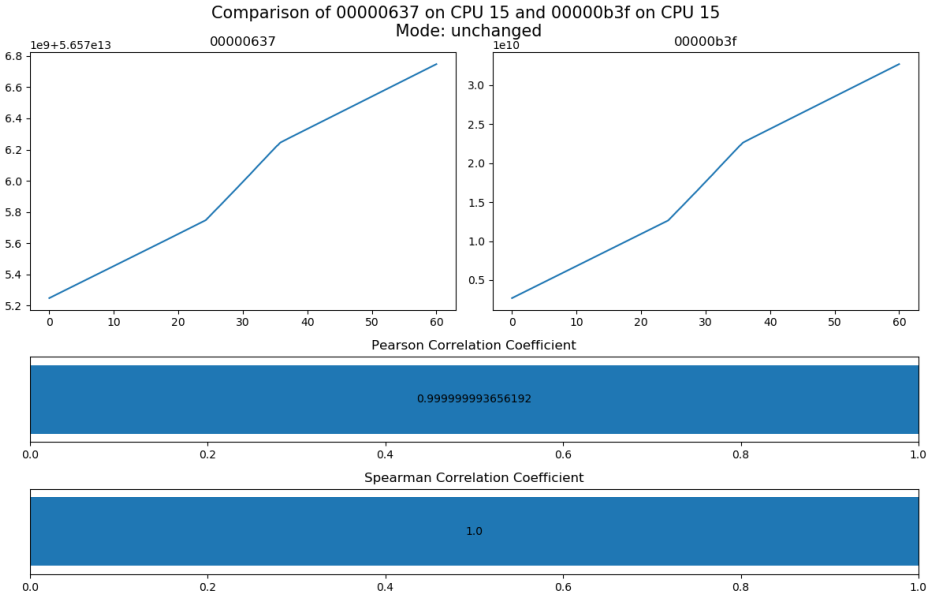



**Figure 10.29:** Correlation of MSR 0x637 and 0xb3f.

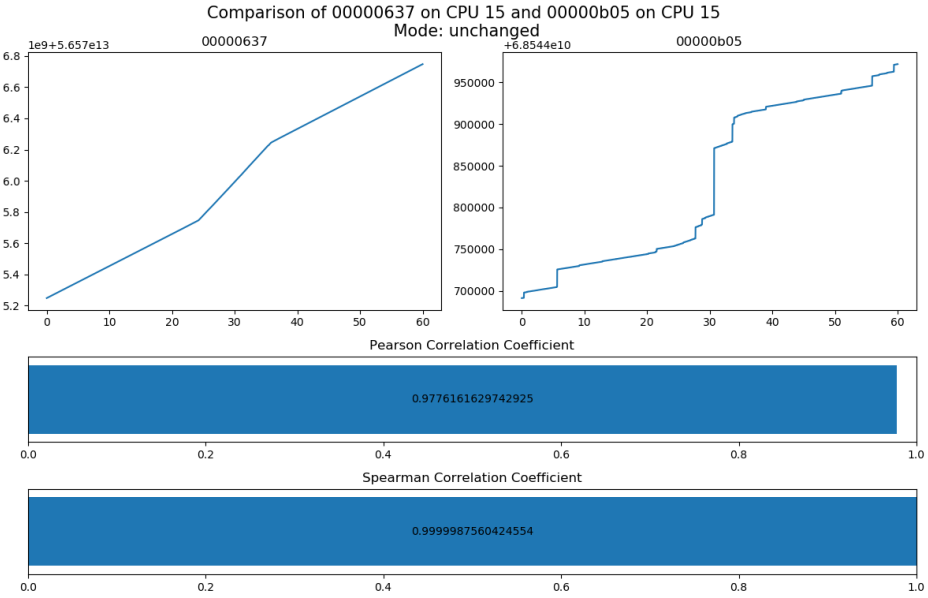



**Figure 10.30:** Correlation of MSR 0x637 and 0xb05.

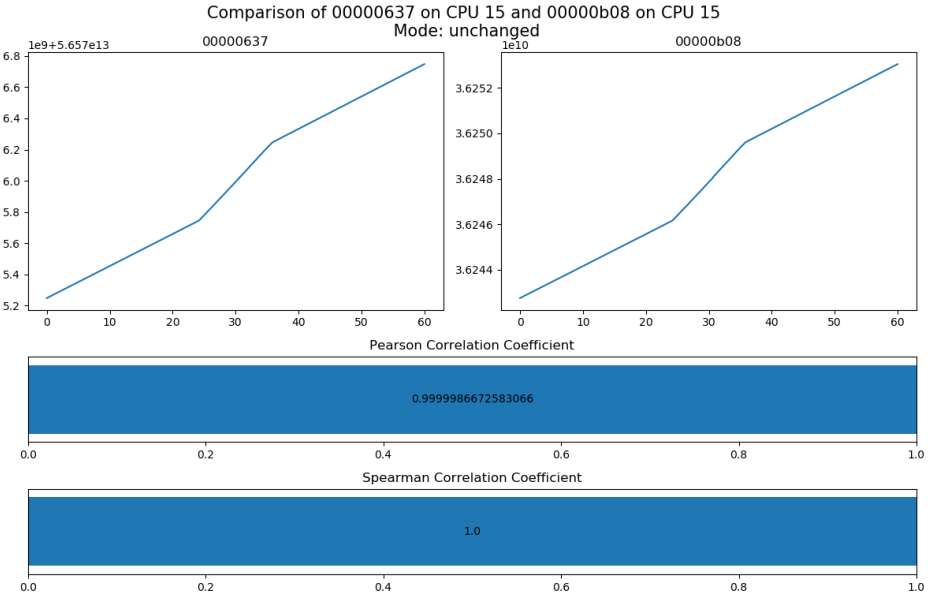



**Figure 10.31:** Correlation of MSR 0x637 and 0xb08.

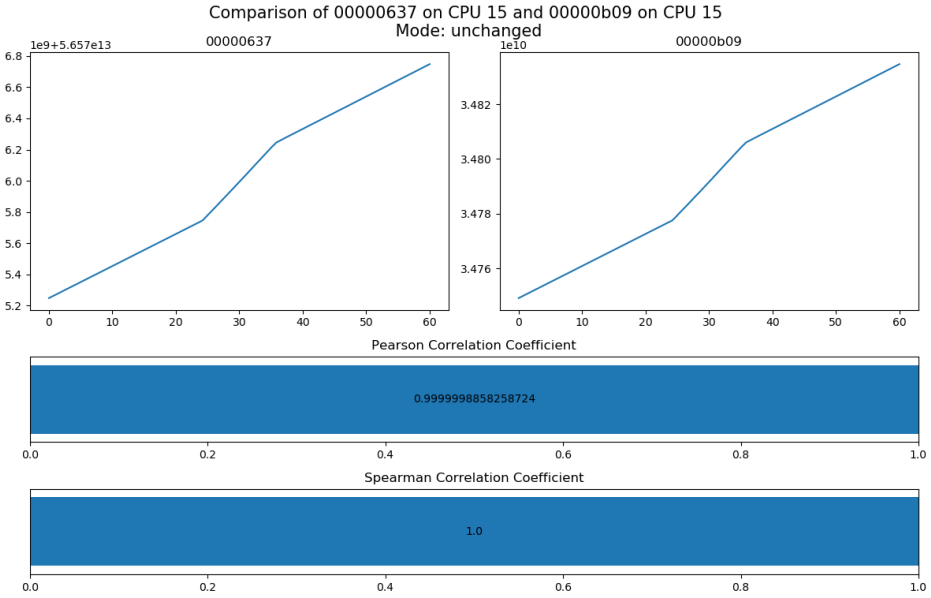



**Figure 10.32:** Correlation of MSR 0x637 and 0xb09.

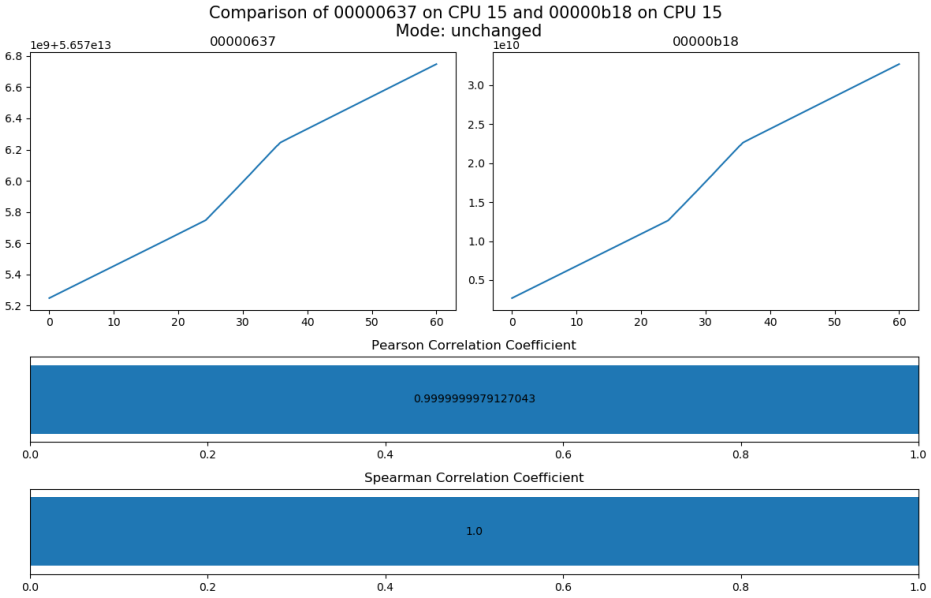



**Figure 10.33:** Correlation of MSR 0x637 and 0xb18.

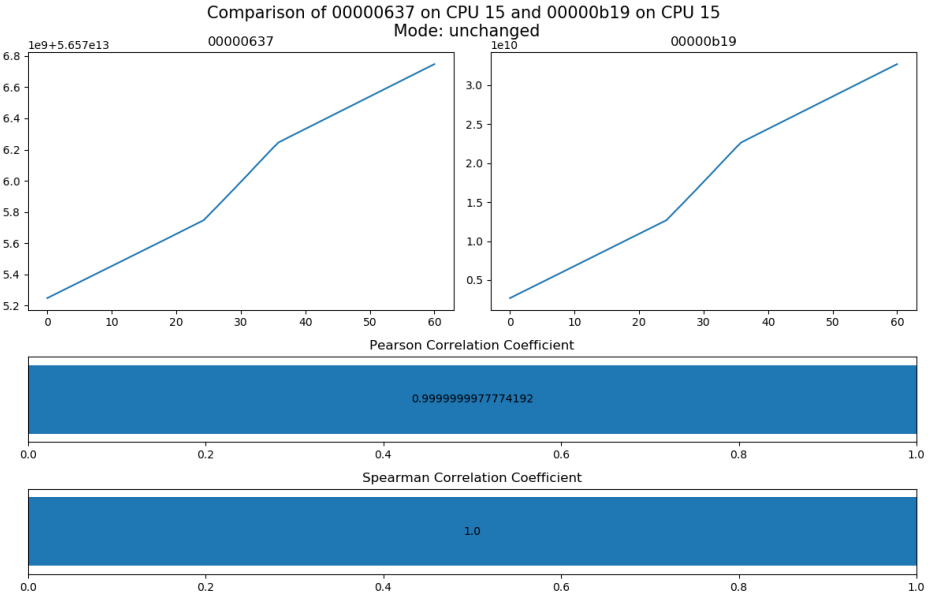



**Figure 10.34:** Correlation of MSR 0x637 and 0xb19.

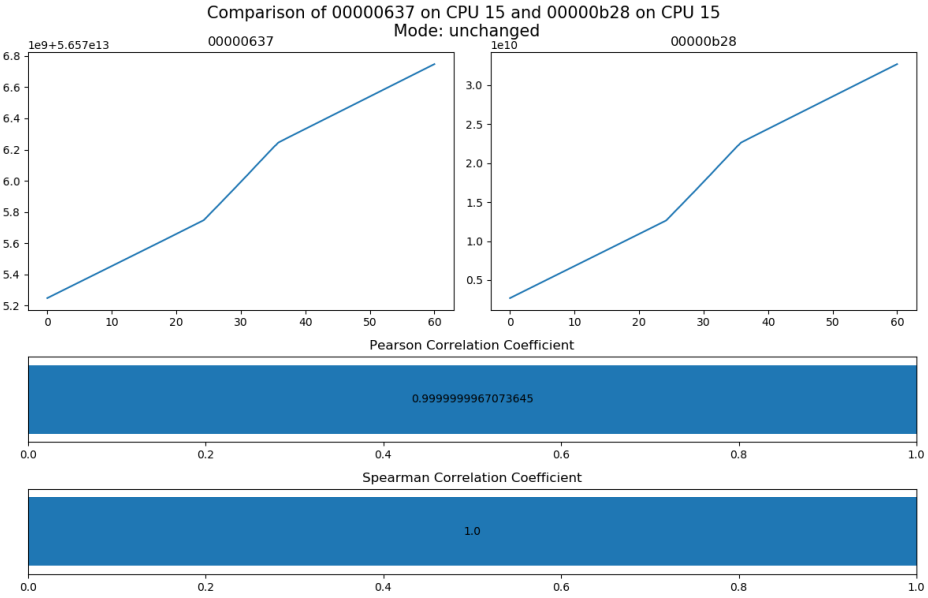



**Figure 10.35:** Correlation of MSR 0x637 and 0xb28.

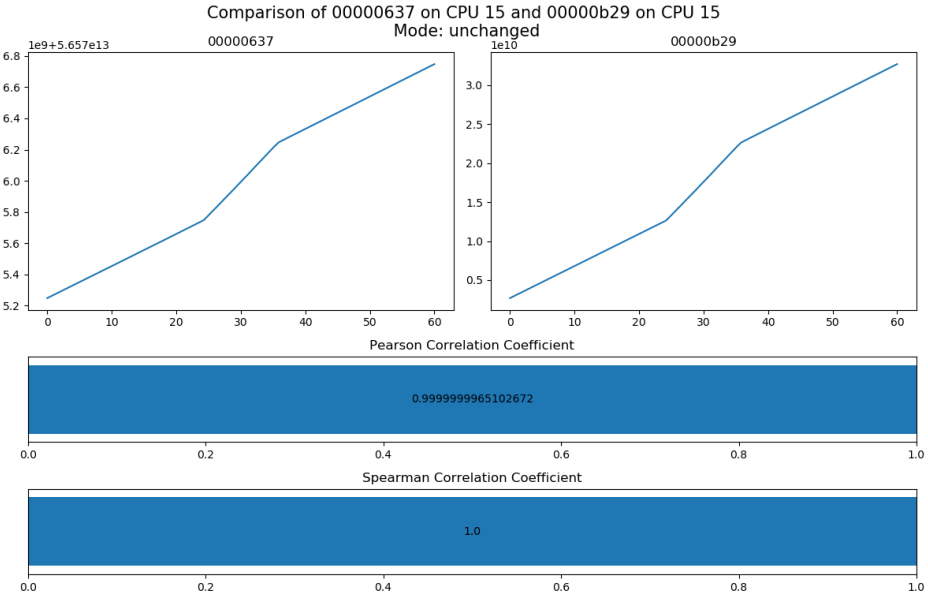



**Figure 10.36:** Correlation of MSR 0x637 and 0xb29.

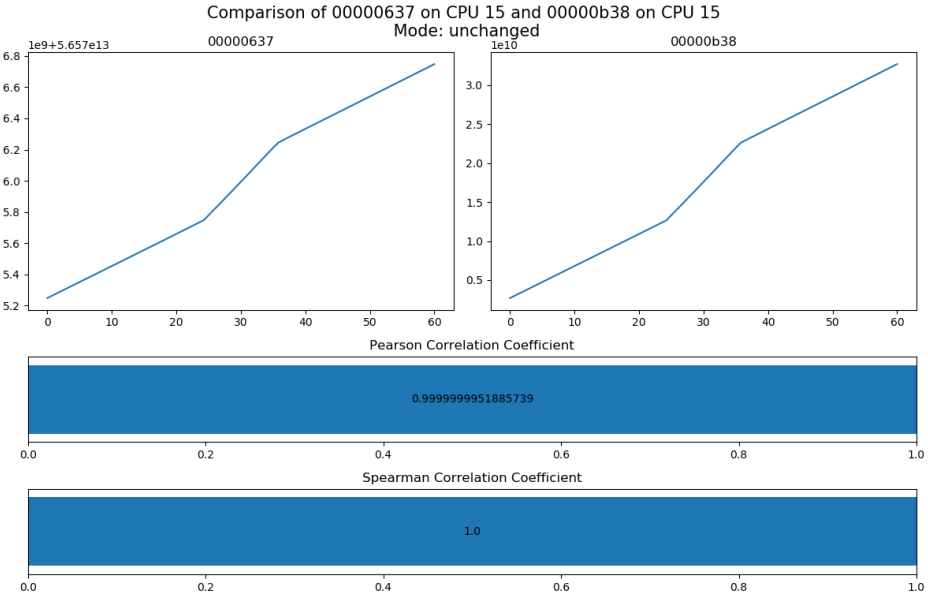



**Figure 10.37:** Correlation of MSR 0x637 and 0xb38.

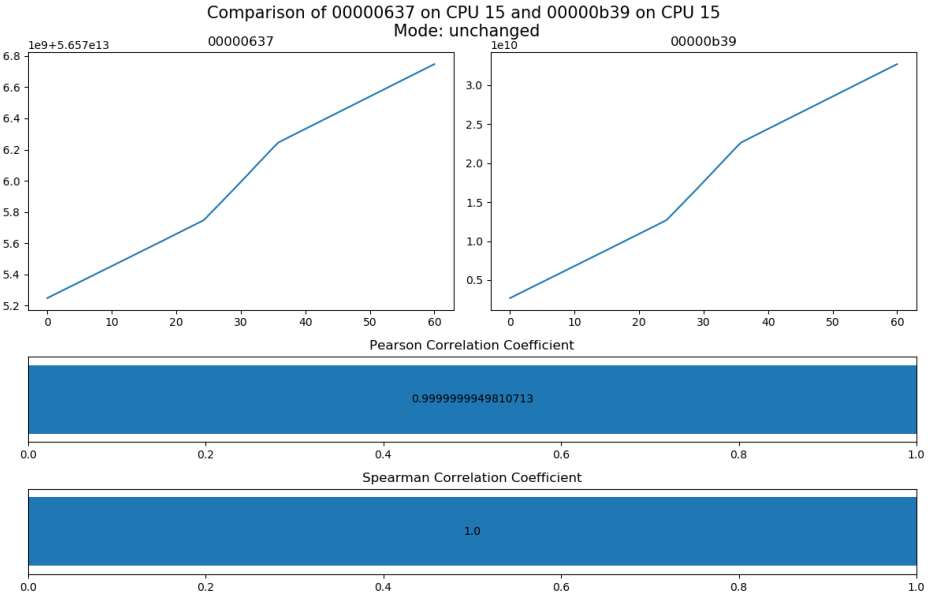



**Figure 10.38:** Correlation of MSR 0x637 and 0xb39.

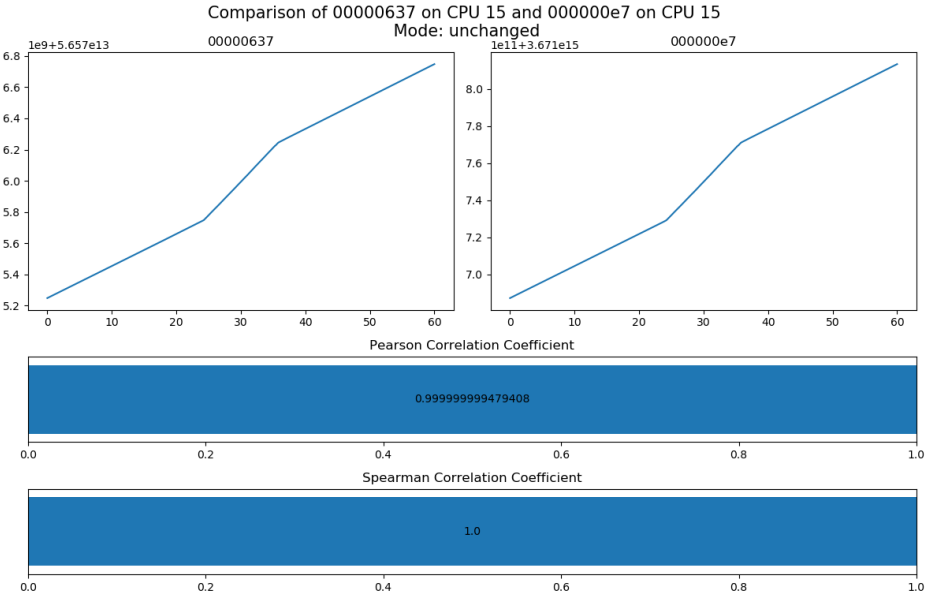



**Figure 10.39:** Correlation of MSR 0x637 and 0xe7.

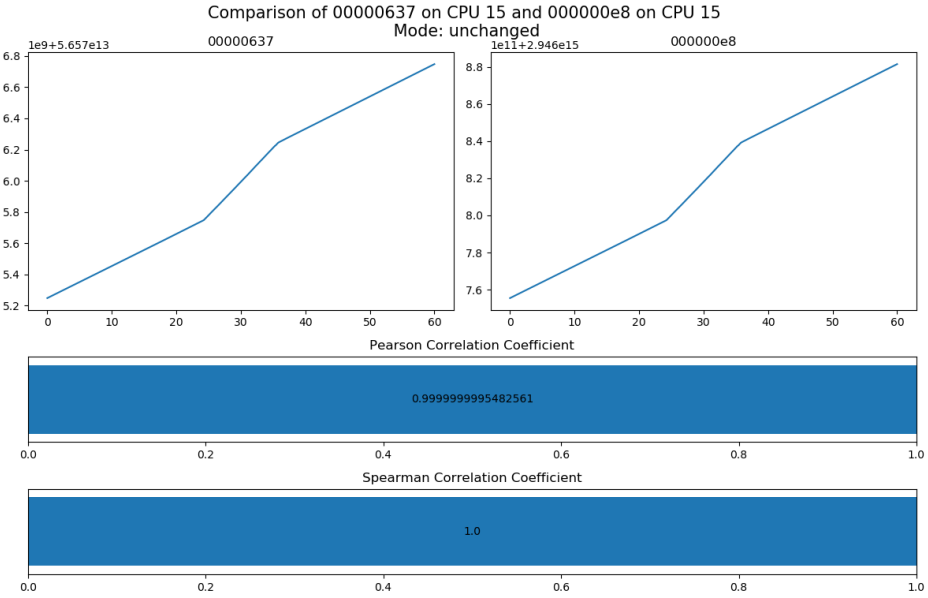



**Figure 10.40:** Correlation of MSR 0x637 and 0xe8.

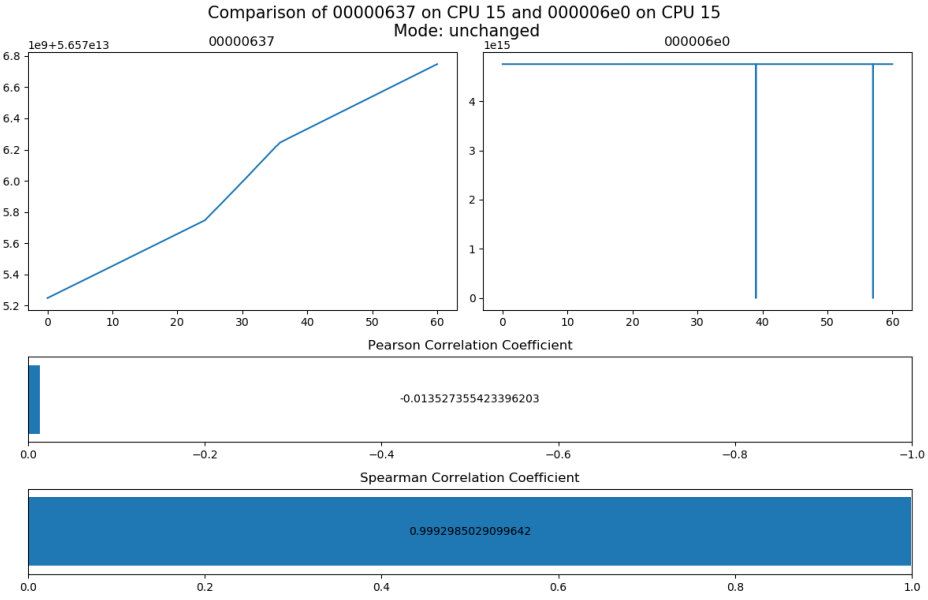



**Figure 10.41:** Correlation of MSR 0x637 and 0x6e0.

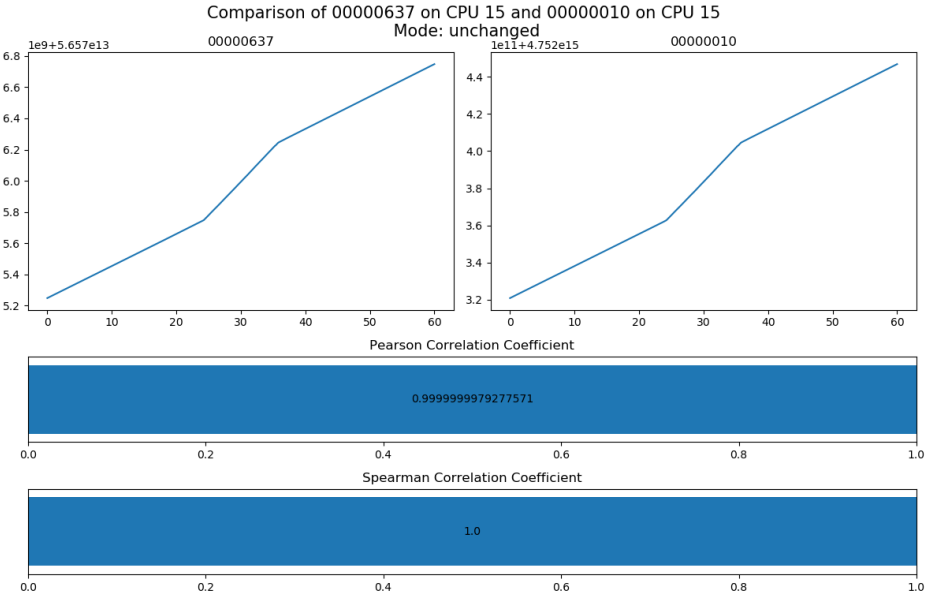



**Figure 10.42:** Correlation of MSR 0x637 and 0x10.

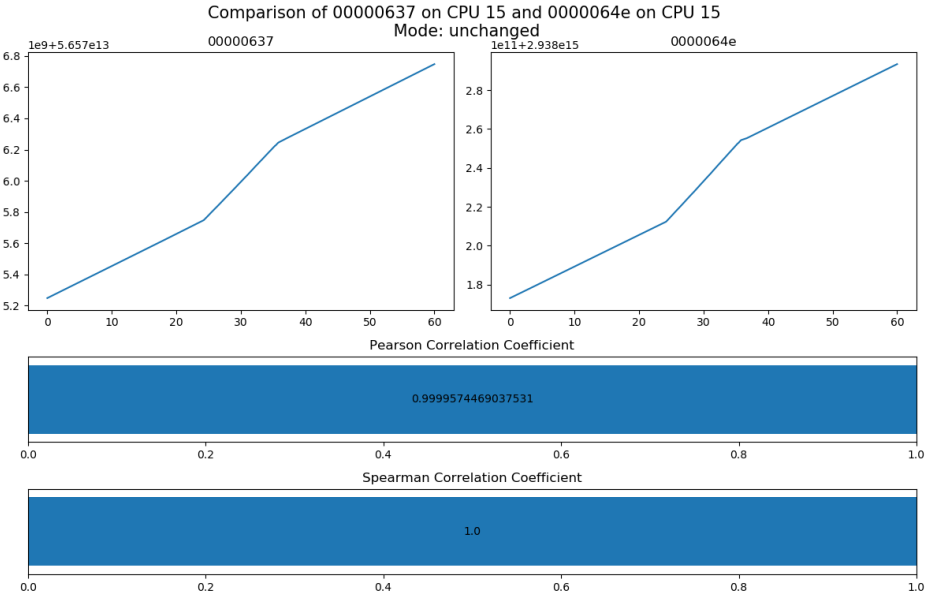



**Figure 10.43:** Correlation of MSR 0x637 and 0x64e.

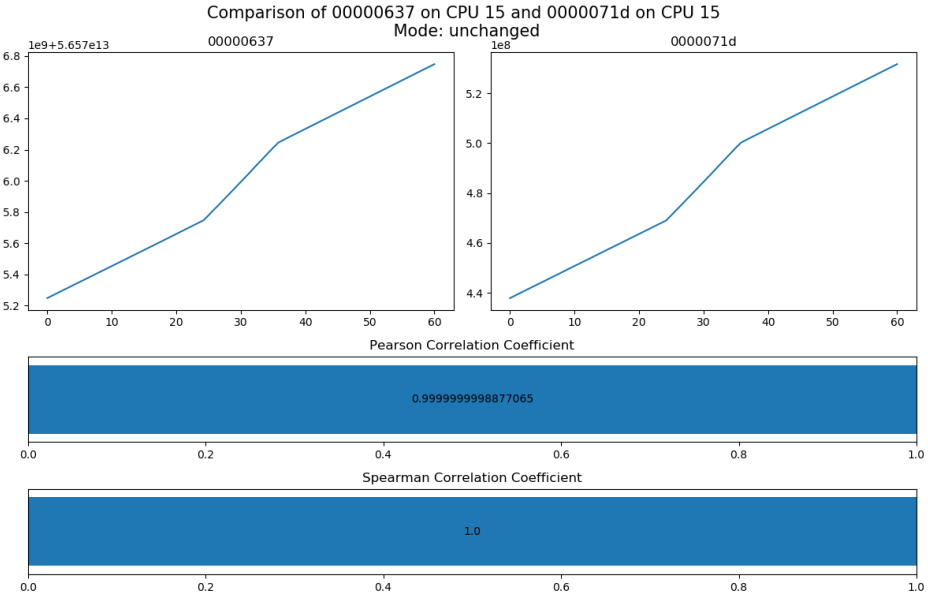



**Figure 10.44:** Correlation of MSR 0x637 and 0x71d.

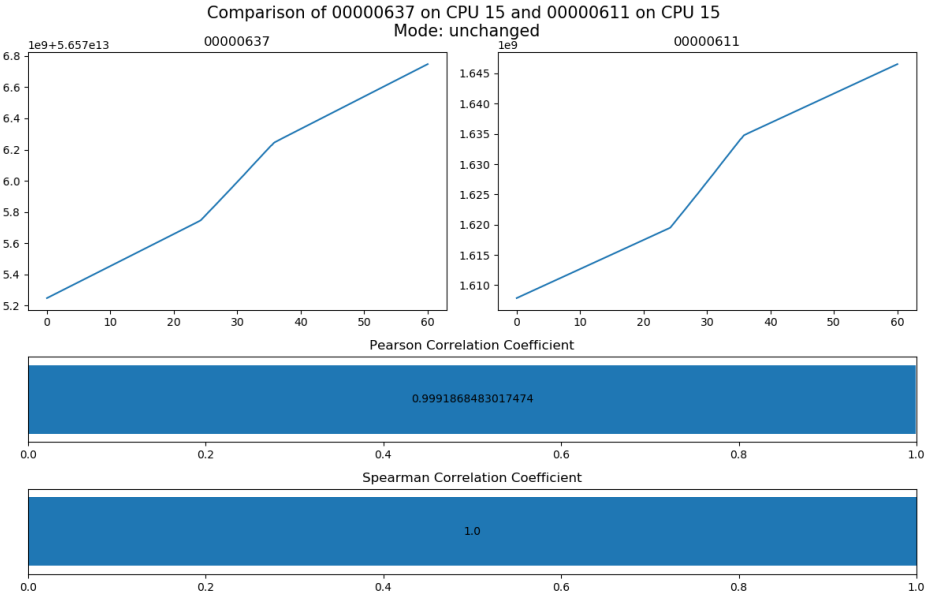



**Figure 10.45:** Correlation of MSR 0x637 and 0x611.

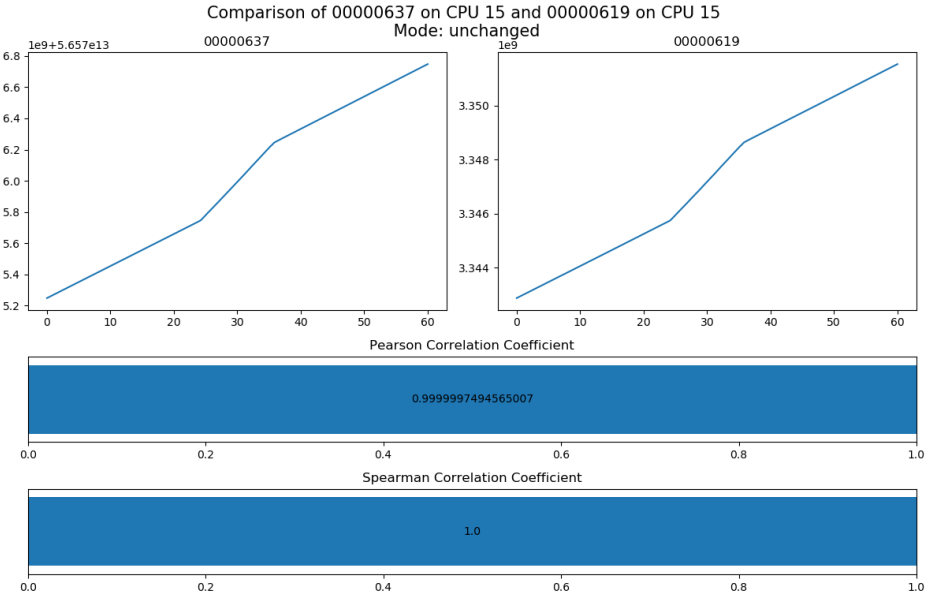



**Figure 10.46:** Correlation of MSR 0x637 and 0x619.

### 10.3.2 Correlations of MSR 0x1a1 with MSR IA32_PACKAGE_THERM_STATUS (0x1b1) on Intel Core CPUs

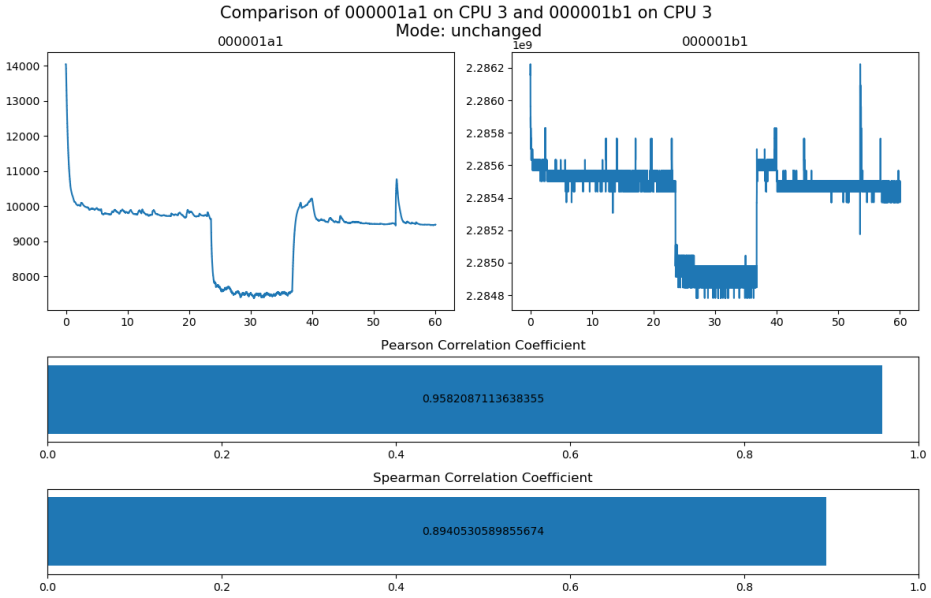



**Figure 10.47:** Correlation of MSR 0x1a1 and the IA32_PACKAGE_THERM_STATUS MSR (0x1b1) on the Intel Core i7-6700K

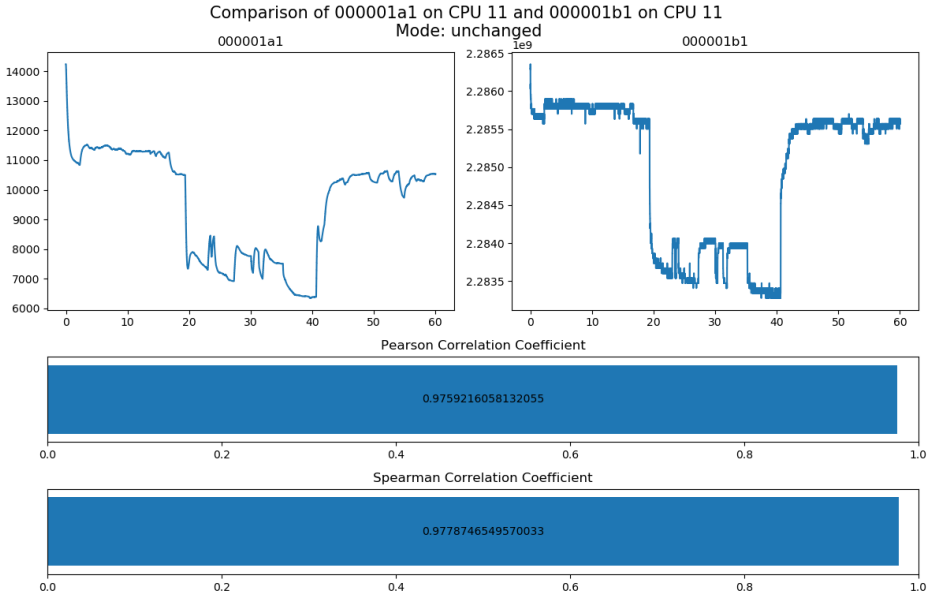



**Figure 10.48:** Correlation of MSR 0x1a1 and the IA32_PACKAGE_THERM_STATUS MSR (0x1b1) on the Intel Core i7-8700K

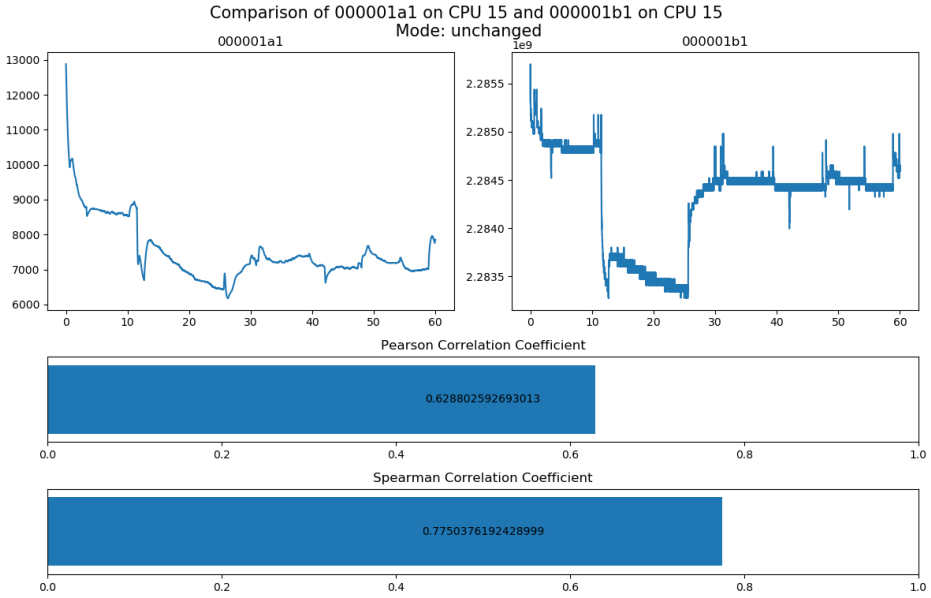



**Figure 10.49:** Correlation of MSR 0x1a1 and the IA32_PACKAGE_THERM_STATUS MSR (0x1b1) on the Intel Core i9-9900K

### 10.3.3 Strong correlations of 0x637 on all tested Intel Core CPUs

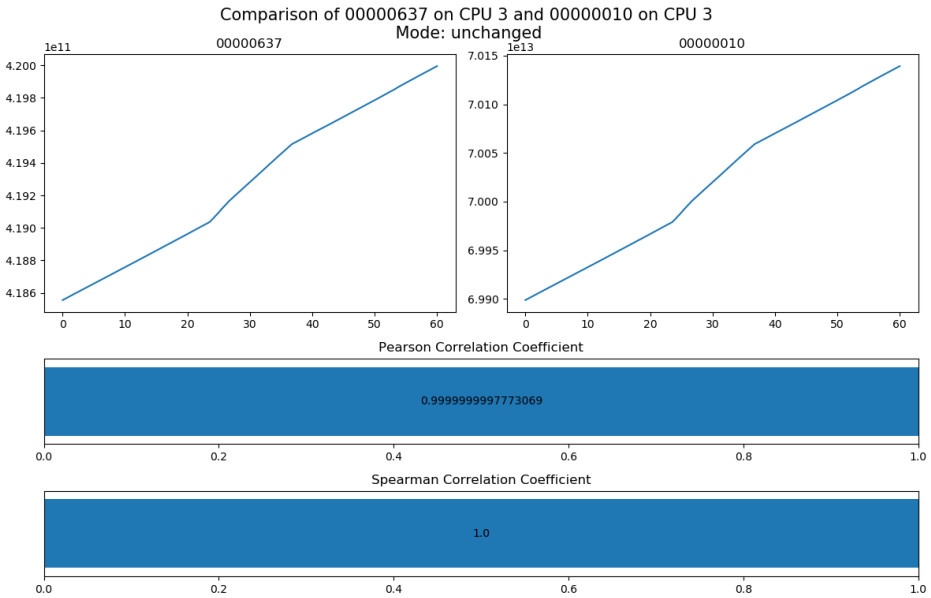



**Figure 10.50:** Correlation of MSR 0x637 and MSR 0x10 on the Intel Core i7-6700K.

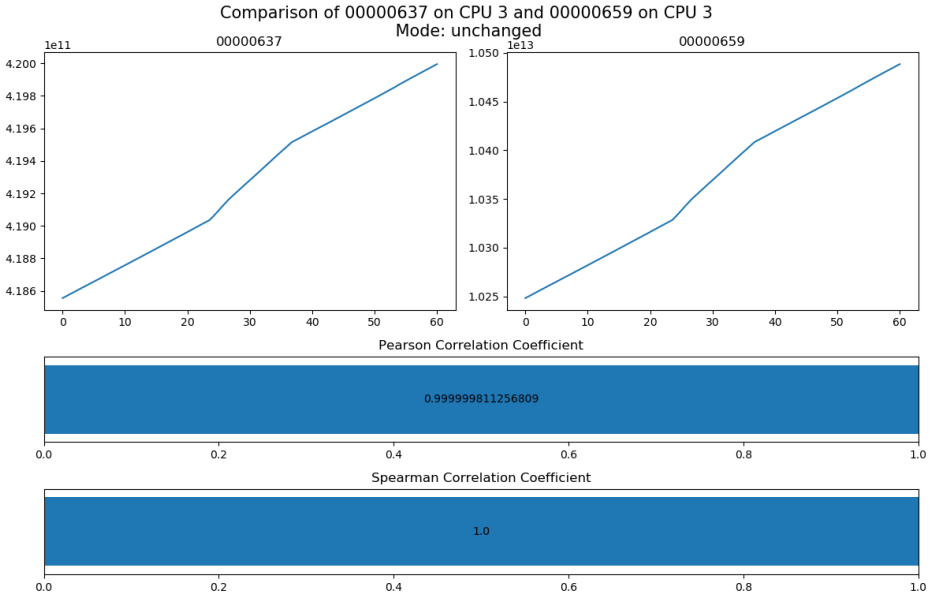



**Figure 10.51:** Correlation of MSR 0x637 and MSR 0x659 on the Intel Core i7-6700K.

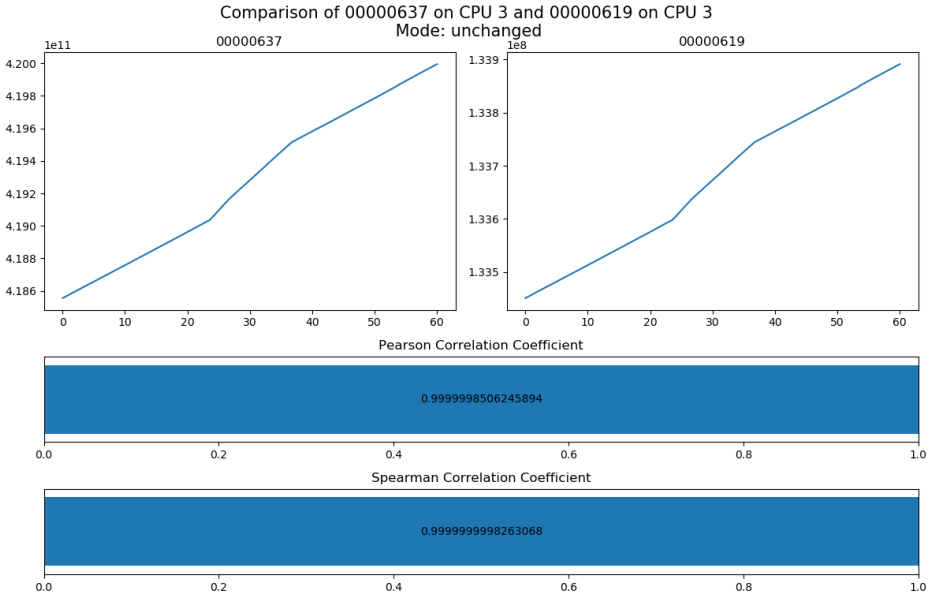



**Figure 10.52:** Correlation of MSR 0x637 and MSR 0x619 on the Intel Core i7-6700K.

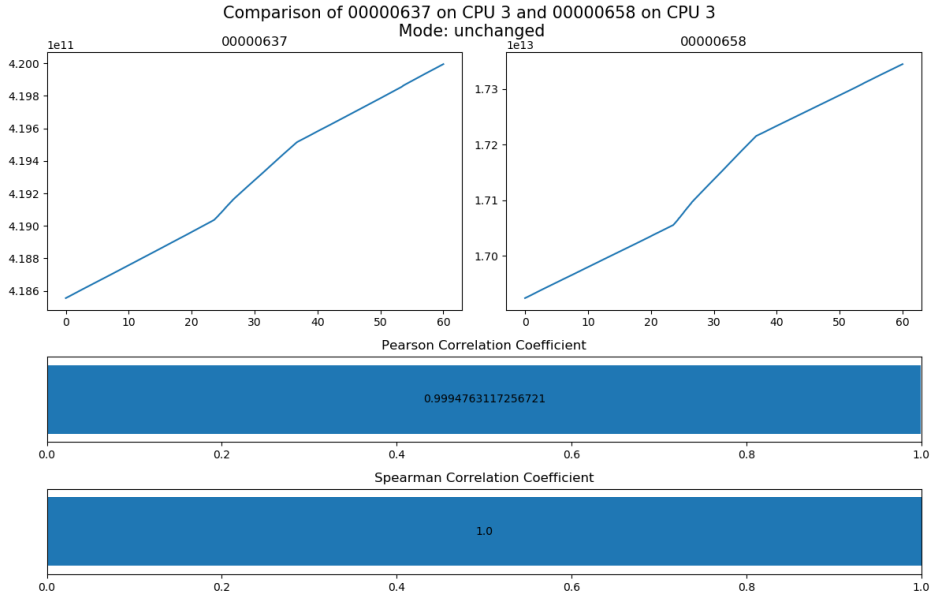



**Figure 10.53:** Correlation of MSR 0x637 and MSR 0x658 on the Intel Core i7-6700K.

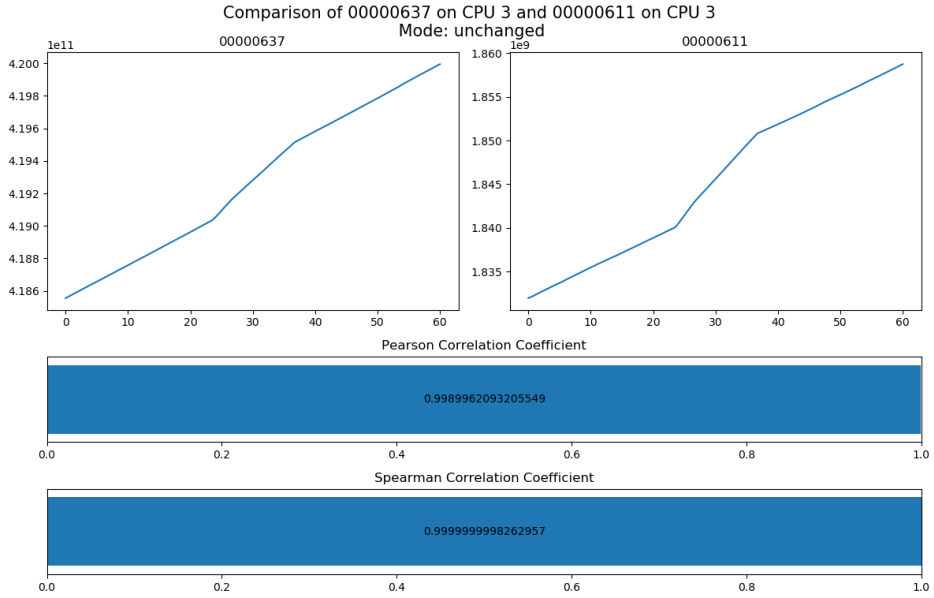



**Figure 10.54:** Correlation of MSR 0x637 and MSR 0x611 on the Intel Core i7-6700K.

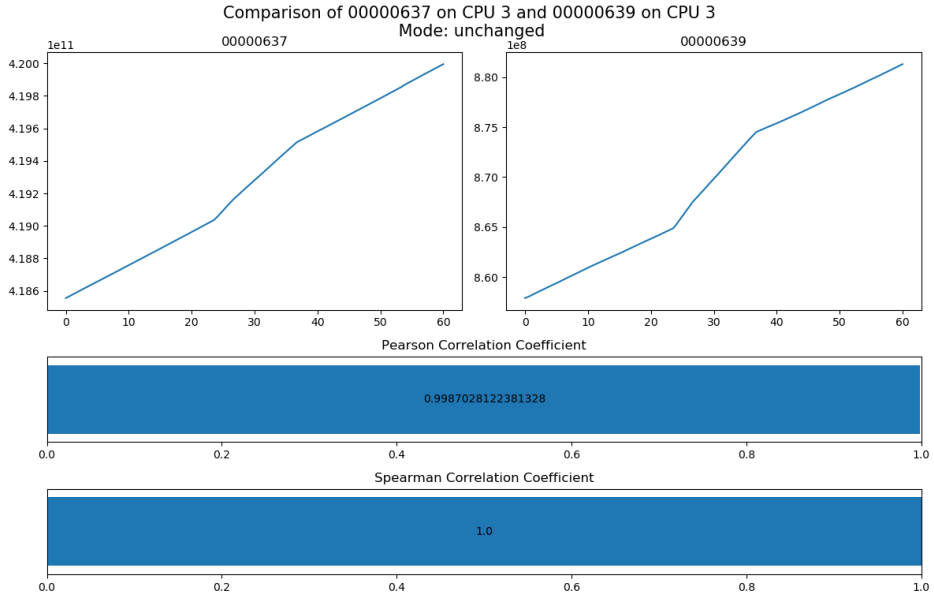



**Figure 10.55:** Correlation of MSR 0x637 and MSR 0x639 on the Intel Core i7-6700K.

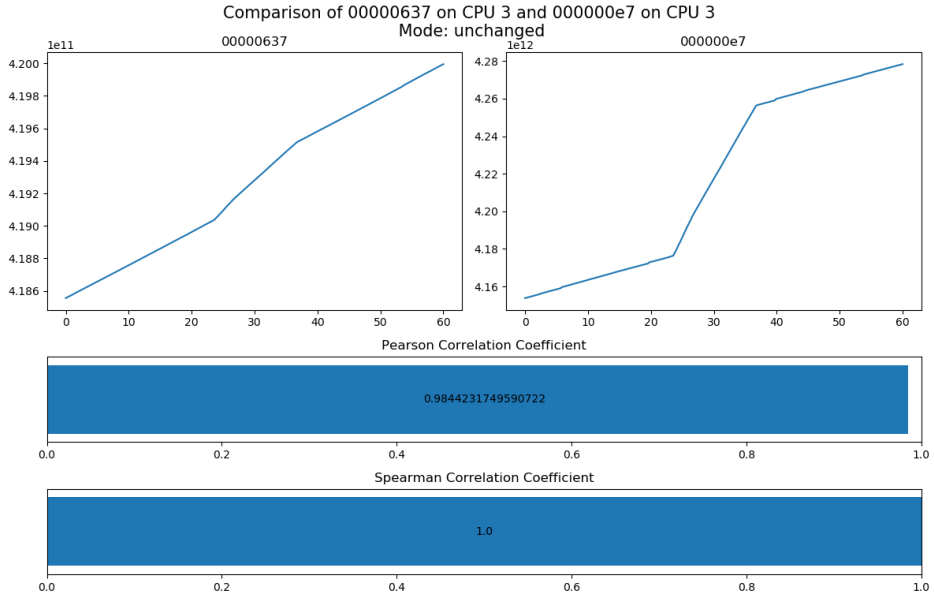



**Figure 10.56:** Correlation of MSR 0x637 and MSR 0xe7 on the Intel Core i7-6700K.

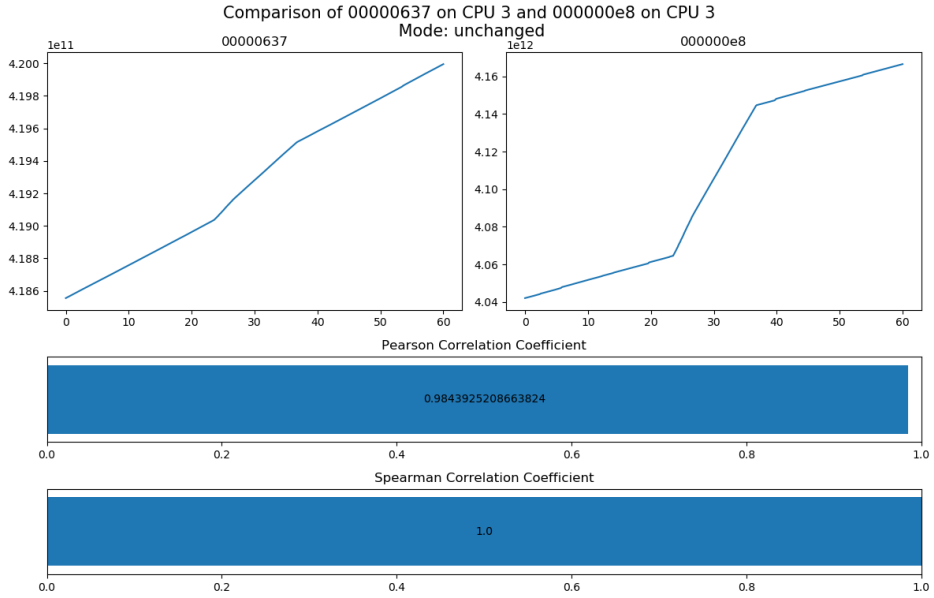



**Figure 10.57:** Correlation of MSR 0x637 and MSR 0xe8 on the Intel Core i7-6700K.

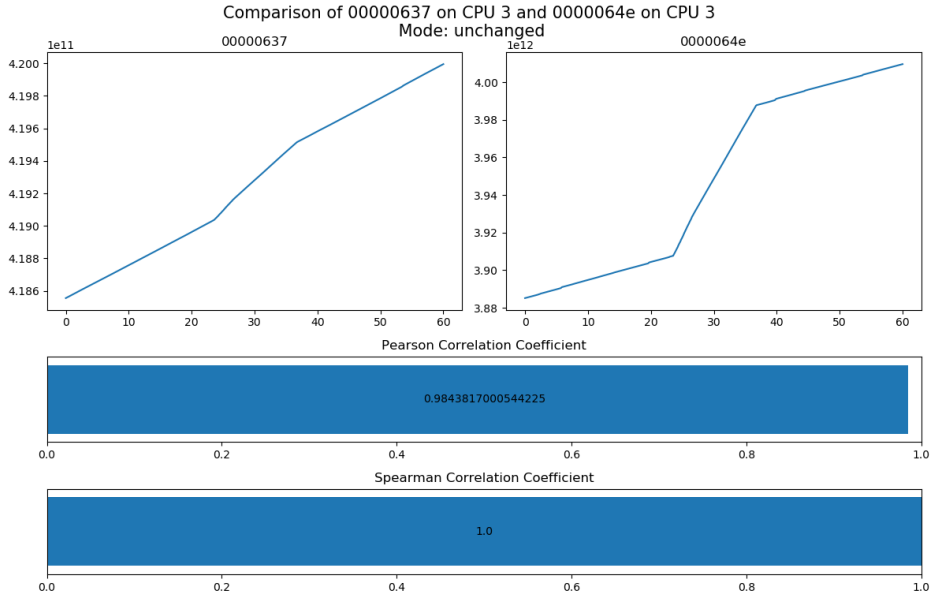



**Figure 10.58:** Correlation of MSR 0x637 and MSR 0x64e on the Intel Core i7-6700K.

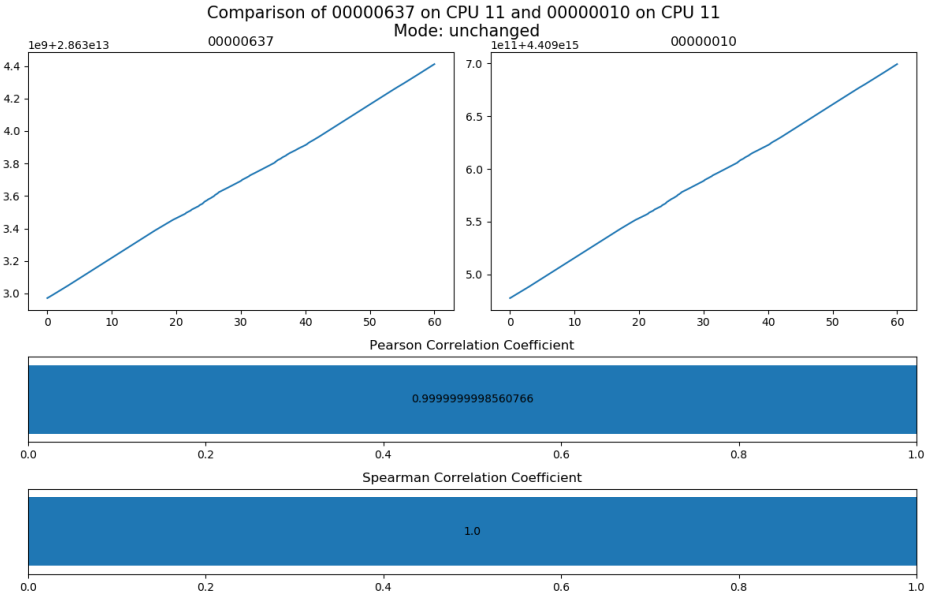



**Figure 10.59:** Correlation of MSR 0x637 and MSR 0x10 on the Intel Core i7-8700K.

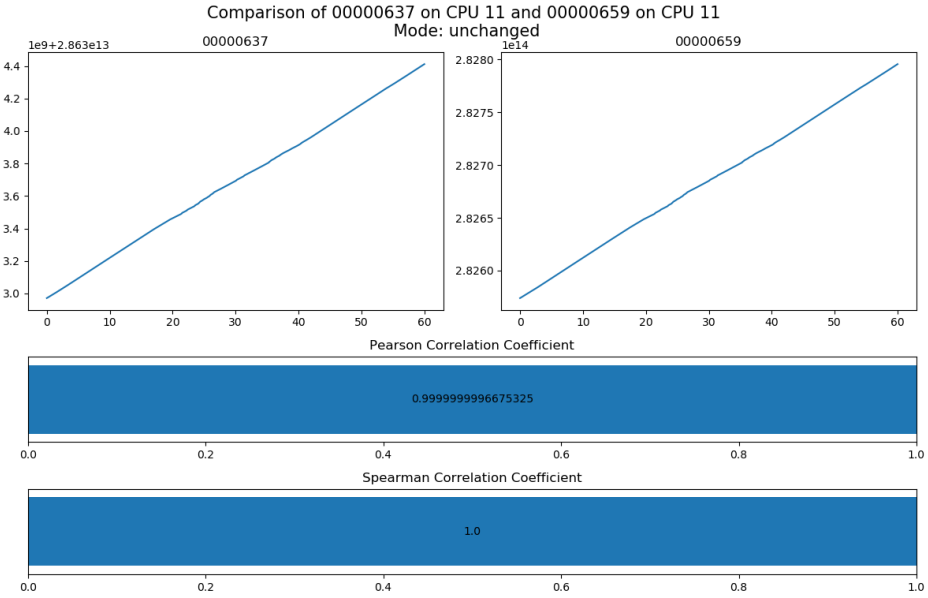



**Figure 10.60:** Correlation of MSR 0x637 and MSR 0x659 on the Intel Core i7-8700K.

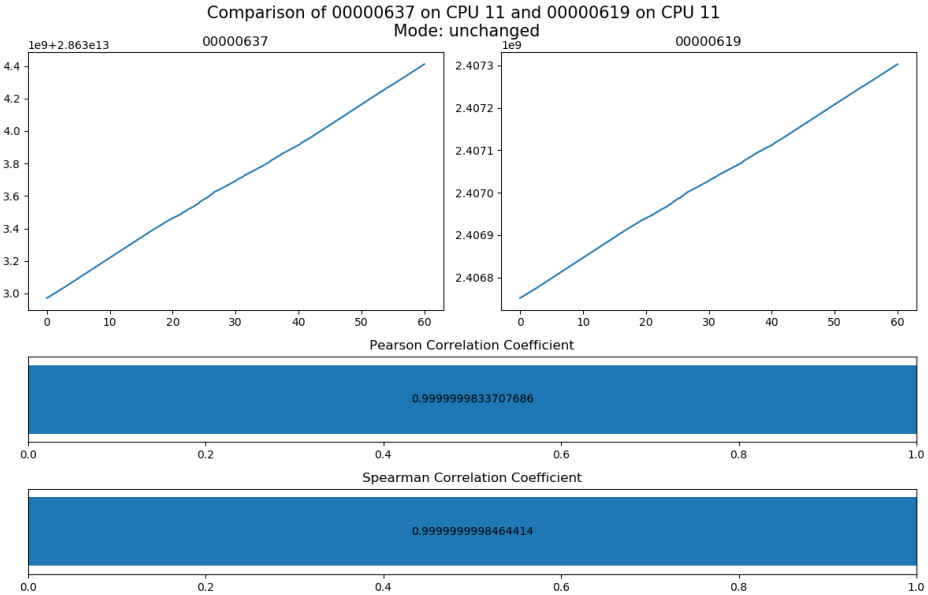



**Figure 10.61:** Correlation of MSR 0x637 and MSR 0x619 on the Intel Core i7-8700K.

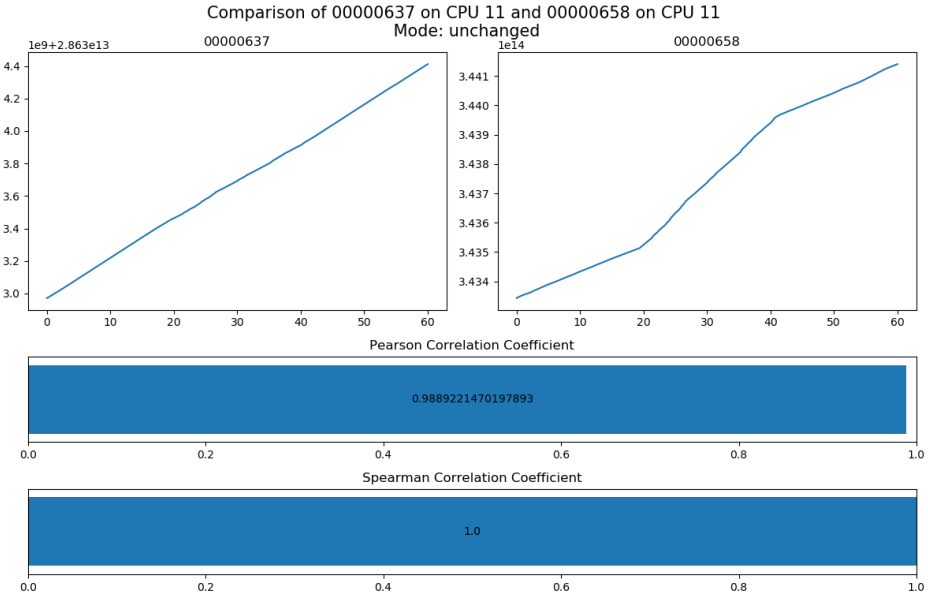



**Figure 10.62:** Correlation of MSR 0x637 and MSR 0x658 on the Intel Core i7-8700K.

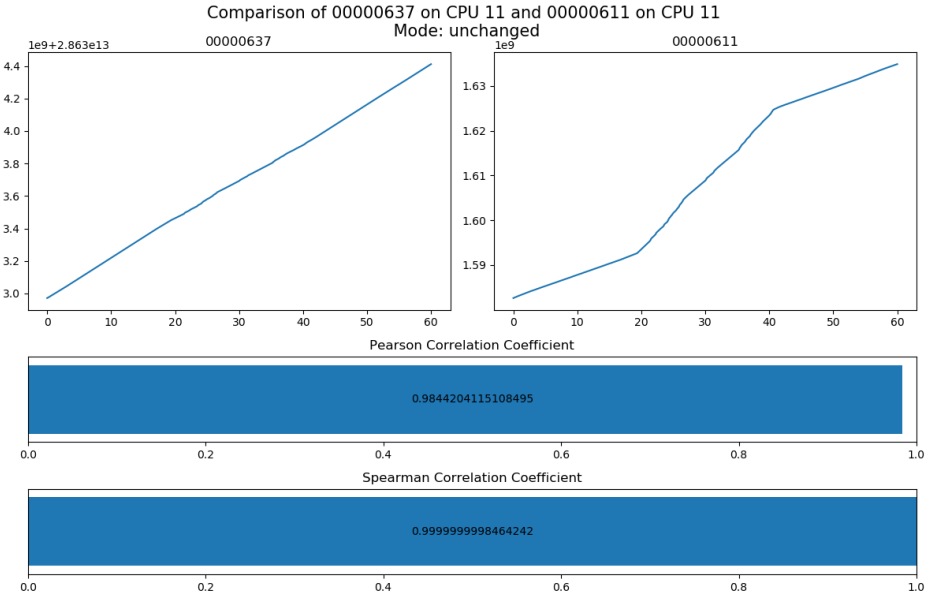



**Figure 10.63:** Correlation of MSR 0x637 and MSR 0x611 on the Intel Core i7-8700K.

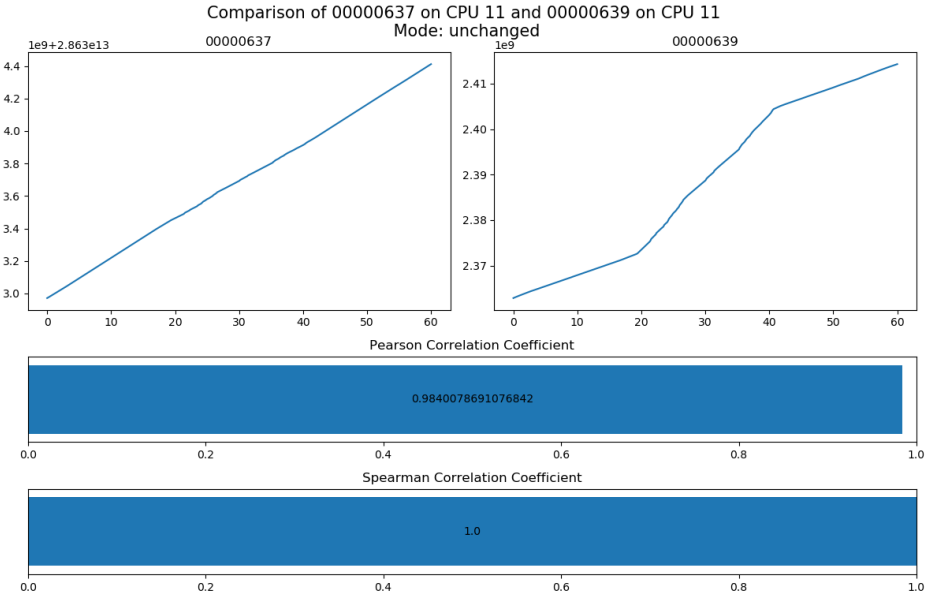



**Figure 10.64:** Correlation of MSR 0x637 and MSR 0x639 on the Intel Core i7-8700K.

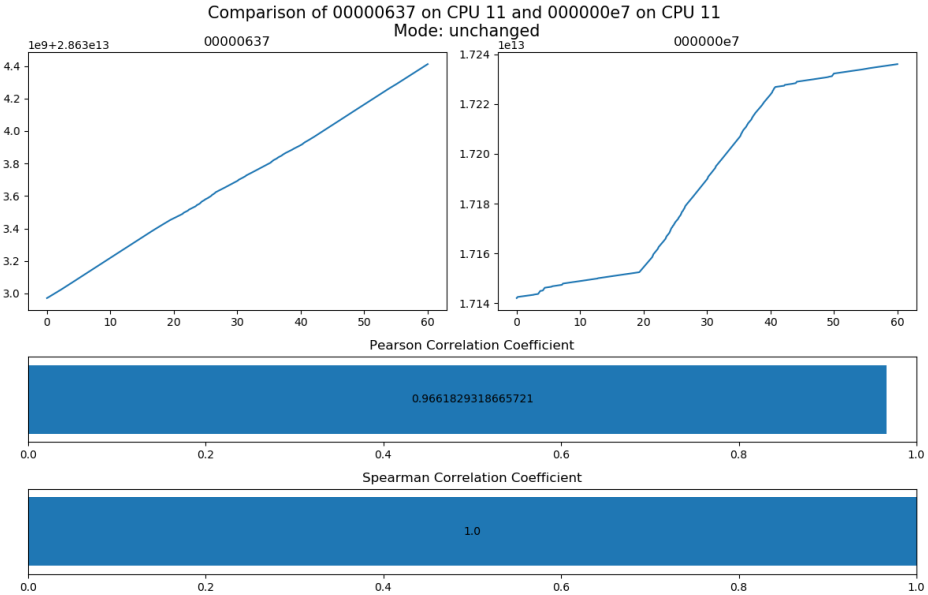



**Figure 10.65:** Correlation of MSR 0x637 and MSR 0xe7 on the Intel Core i7-8700K.

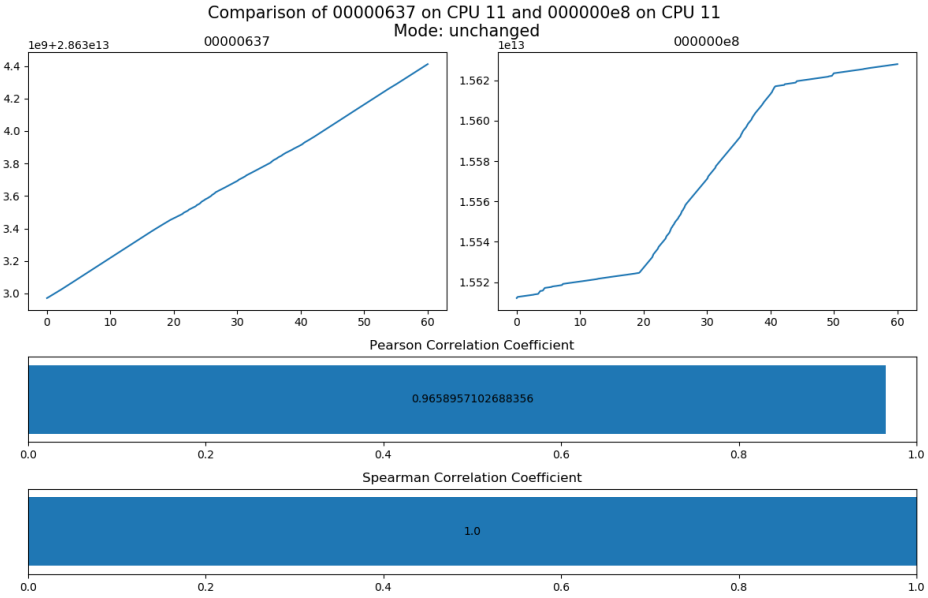



**Figure 10.66:** Correlation of MSR 0x637 and MSR 0xe8 on the Intel Core i7-8700K.

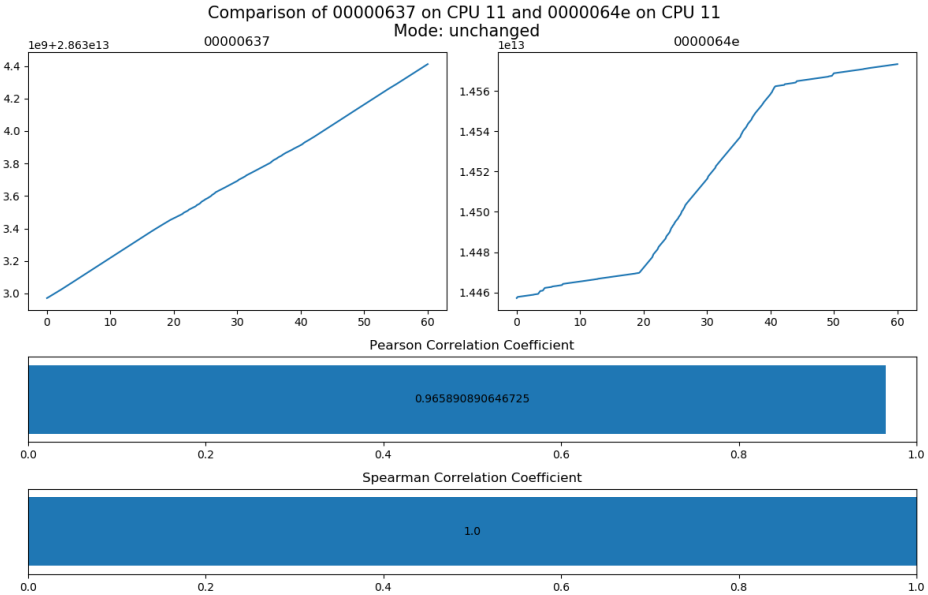



**Figure 10.67:** Correlation of MSR 0x637 and MSR 0x64e on the Intel Core i7-8700K.

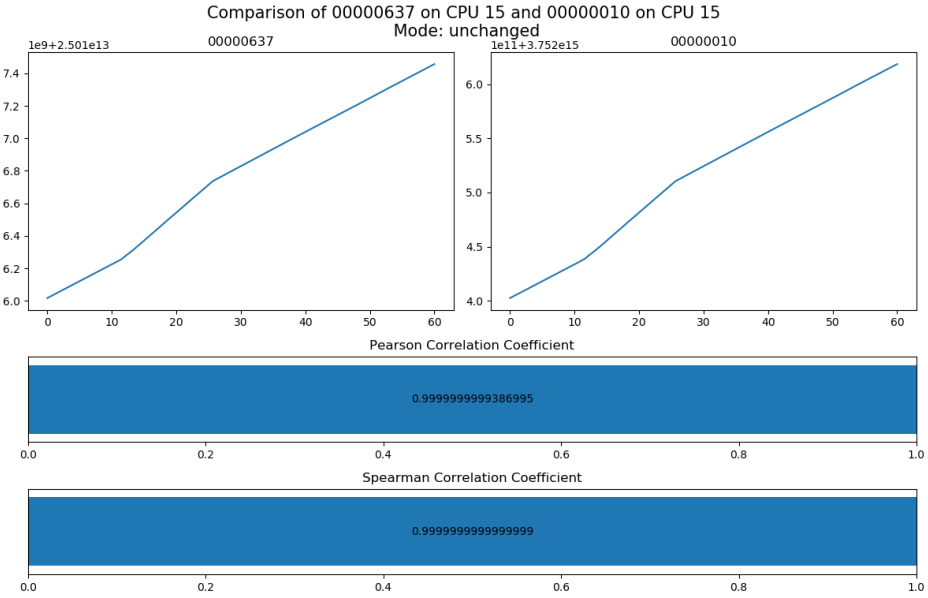



**Figure 10.68:** Correlation of MSR 0x637 and MSR 0x10 on the Intel Core i9-9900K.

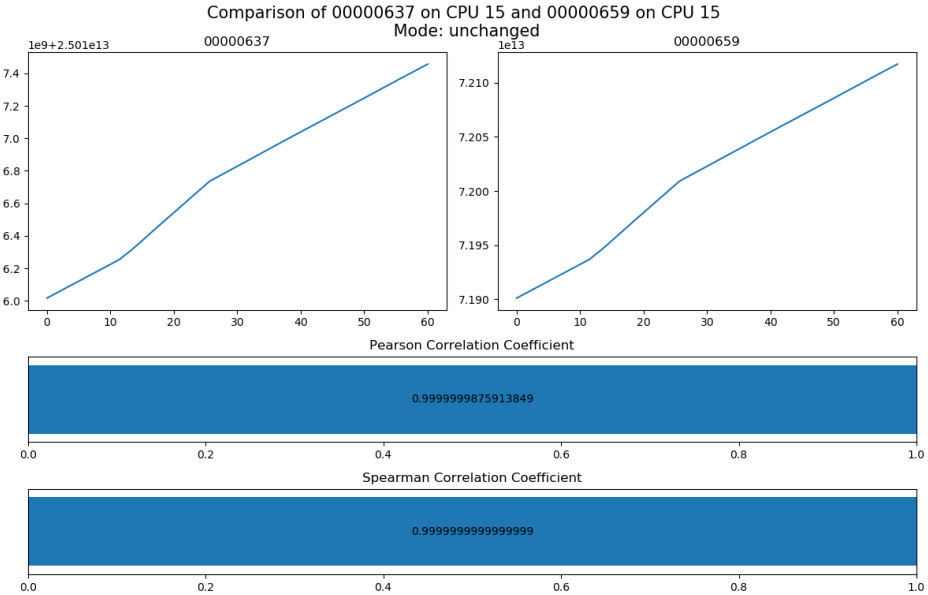



**Figure 10.69:** Correlation of MSR 0x637 and MSR 0x659 on the Intel Core i9-9900K.

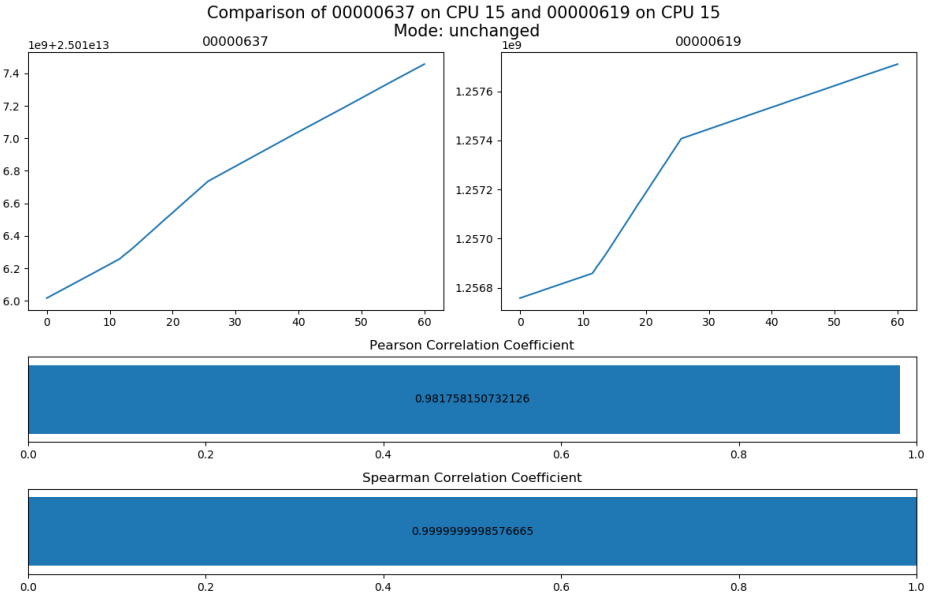



**Figure 10.70:** Correlation of MSR 0x637 and MSR 0x619 on the Intel Core i9-9900K.

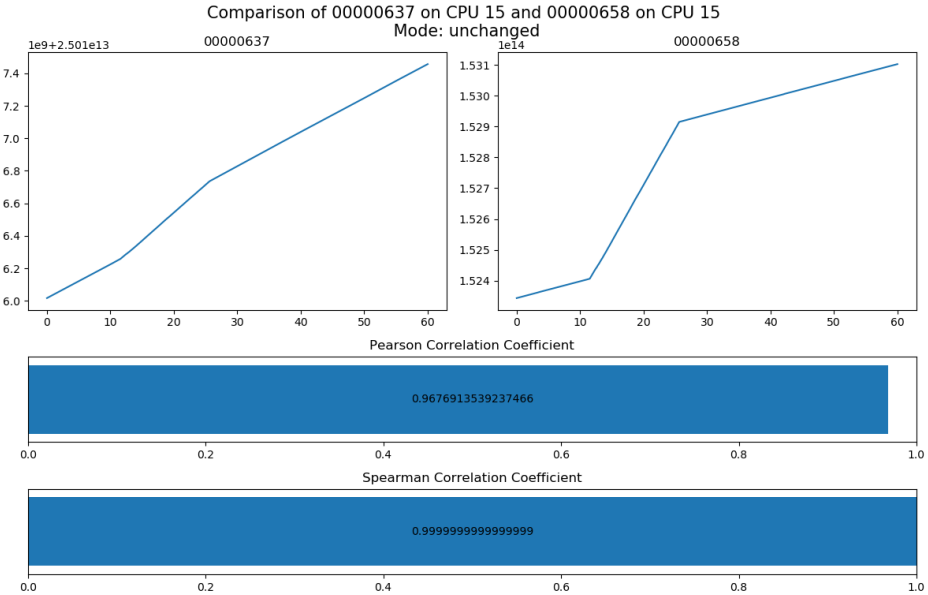



**Figure 10.71:** Correlation of MSR 0x637 and MSR 0x658 on the Intel Core i9-9900K.

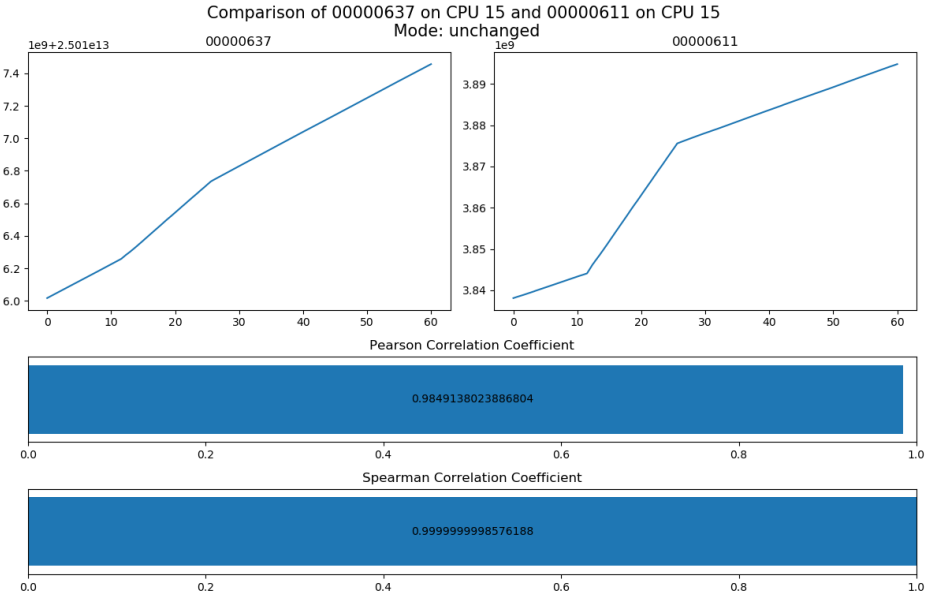



**Figure 10.72:** Correlation of MSR 0x637 and MSR 0x611 on the Intel Core i9-9900K.

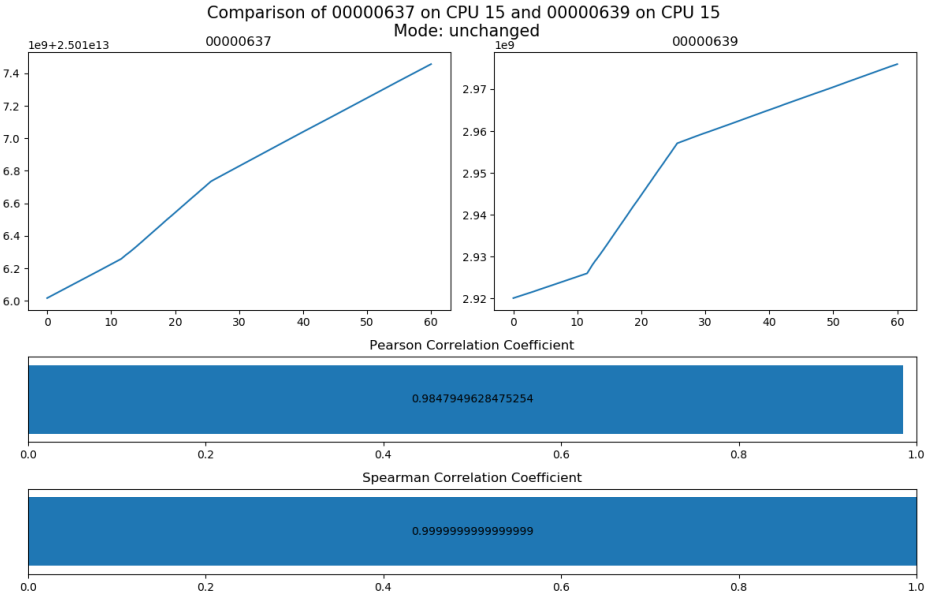



**Figure 10.73:** Correlation of MSR 0x637 and MSR 0x639 on the Intel Core i9-9900K.

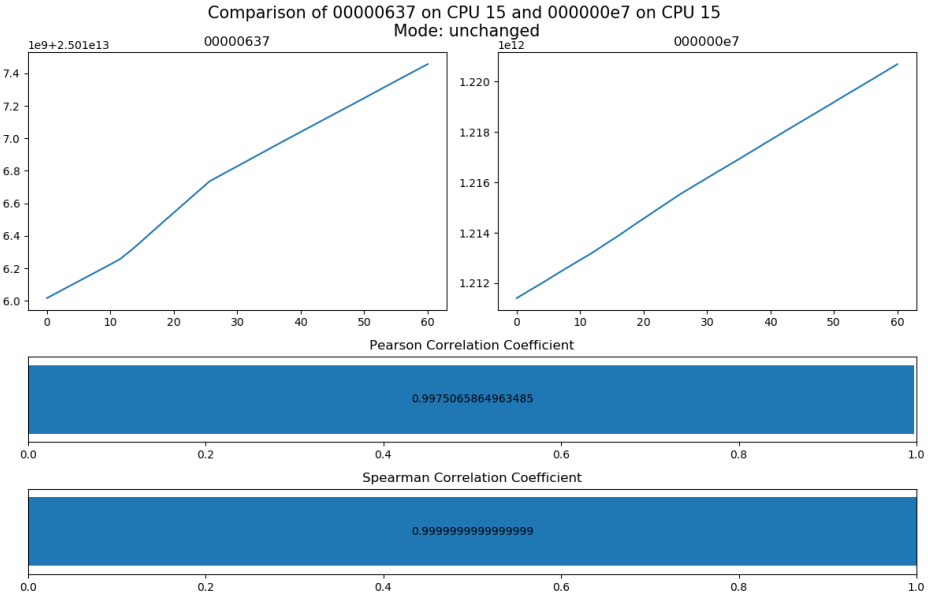



**Figure 10.74:** Correlation of MSR 0x637 and MSR 0xe7 on the Intel Core i9-9900K.

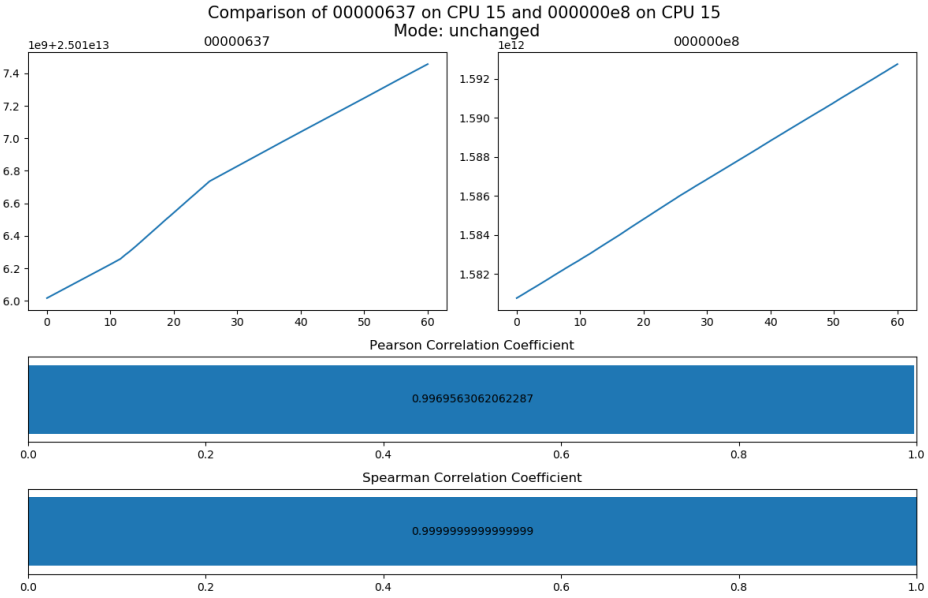



**Figure 10.75:** Correlation of MSR 0x637 and MSR 0xe8 on the Intel Core i9-9900K.

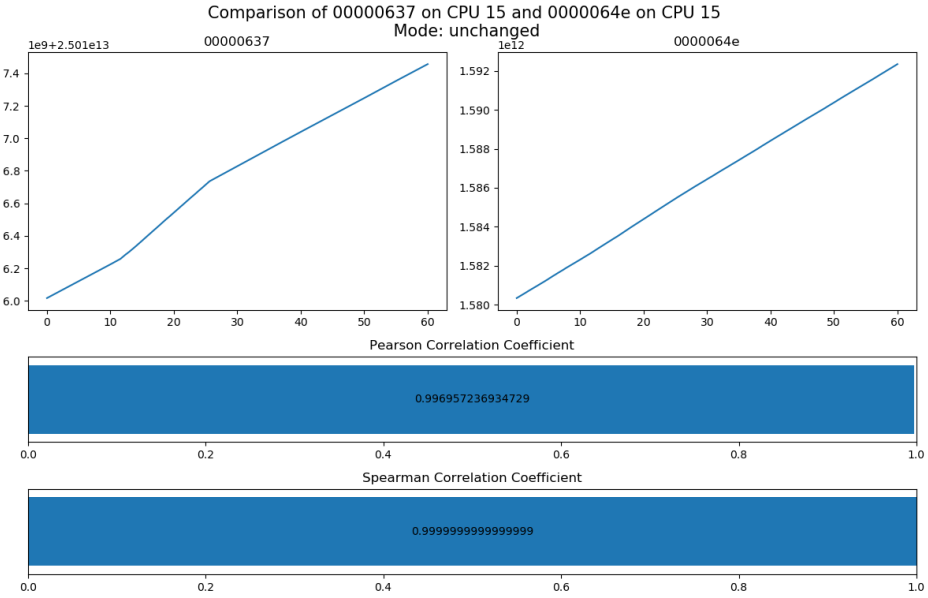



**Figure 10.76:** Correlation of MSR 0x637 and MSR 0x64e on the Intel Core i9-9900K.