

Barrys SS 75 with 65536 bit Architecture Motherboard Design/Special Project # 1

by

Barry L. Crouse

Introduction

Thank you for taking the time in reading this Scientific Work. I was recently given some nice presents such as Registries that became corrupted, Different Characters in the Bios mainly controllers, 256 AES Encrypted Operating Systems on my PCIE 1 Gig Video Card. I will have to provide a solution to these special problems that came about in the end of July 2015.

The Three main problems I hope to address is the following:

- 1). Introduction to the 65536 bit Architecture.
- 2). Creating Dynamic BIOS Instruction sets that overcomes the New and improved flavor ??
- 3). Creating a New Algorithm Scheme.

I will also introduce a new Mathematical equation as a extra bonus to the special problems that were created for my benefit a little humor. As a additional note, **I will classify this as a special project since my copyright career was suppose to be ended after reaching 61.**

Thank you for reading this work!

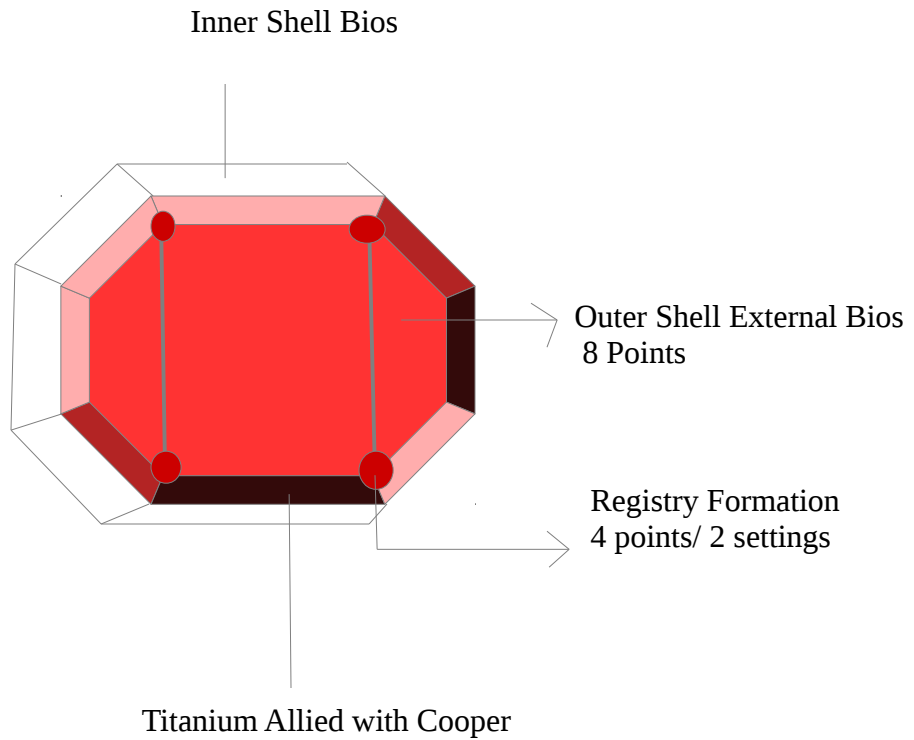
Table of Contents/Special Projects # 1

Chapter 1	Visual Design
Chapter 2	Chart and Spec Comparisons
Chapter 3	Bios Processing
Chapter 4	Mathematical equations
Chapter 5	Final Thoughts

Chapter 1

Visual Design

Model Super Sonic 75 Motherboard- Design 1-A



Bios Configuration Specs	# of Points on Chip	# of points per Registry	# of setting
External	8	4	2
Internal	8	4	2
Total Number of Points	16	8	4

BIOS Spec Charts

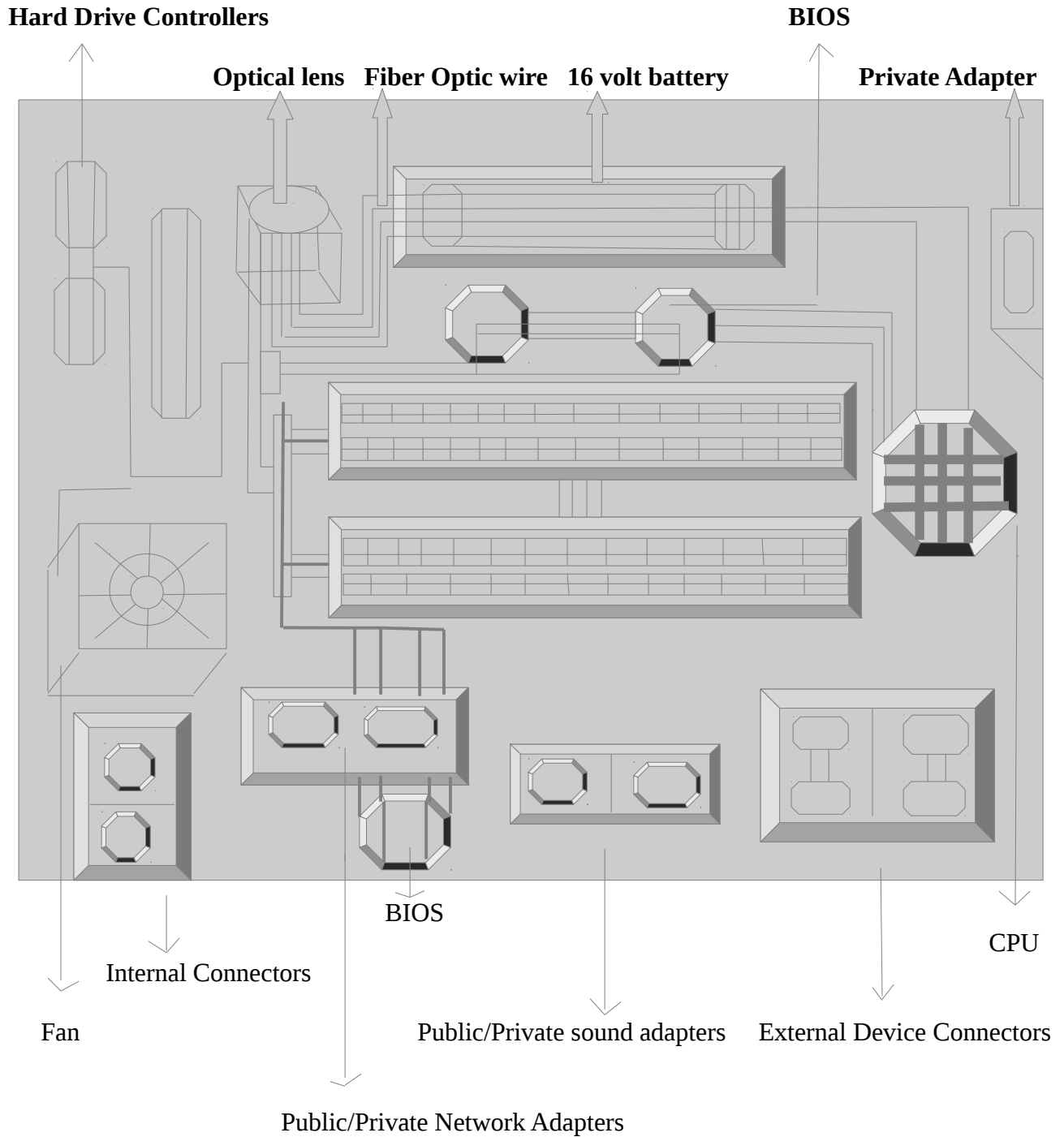
Outer shell Bios Dynamic Instruction Sets

Setting #	# of bit Instruction Set
1	16
2	256

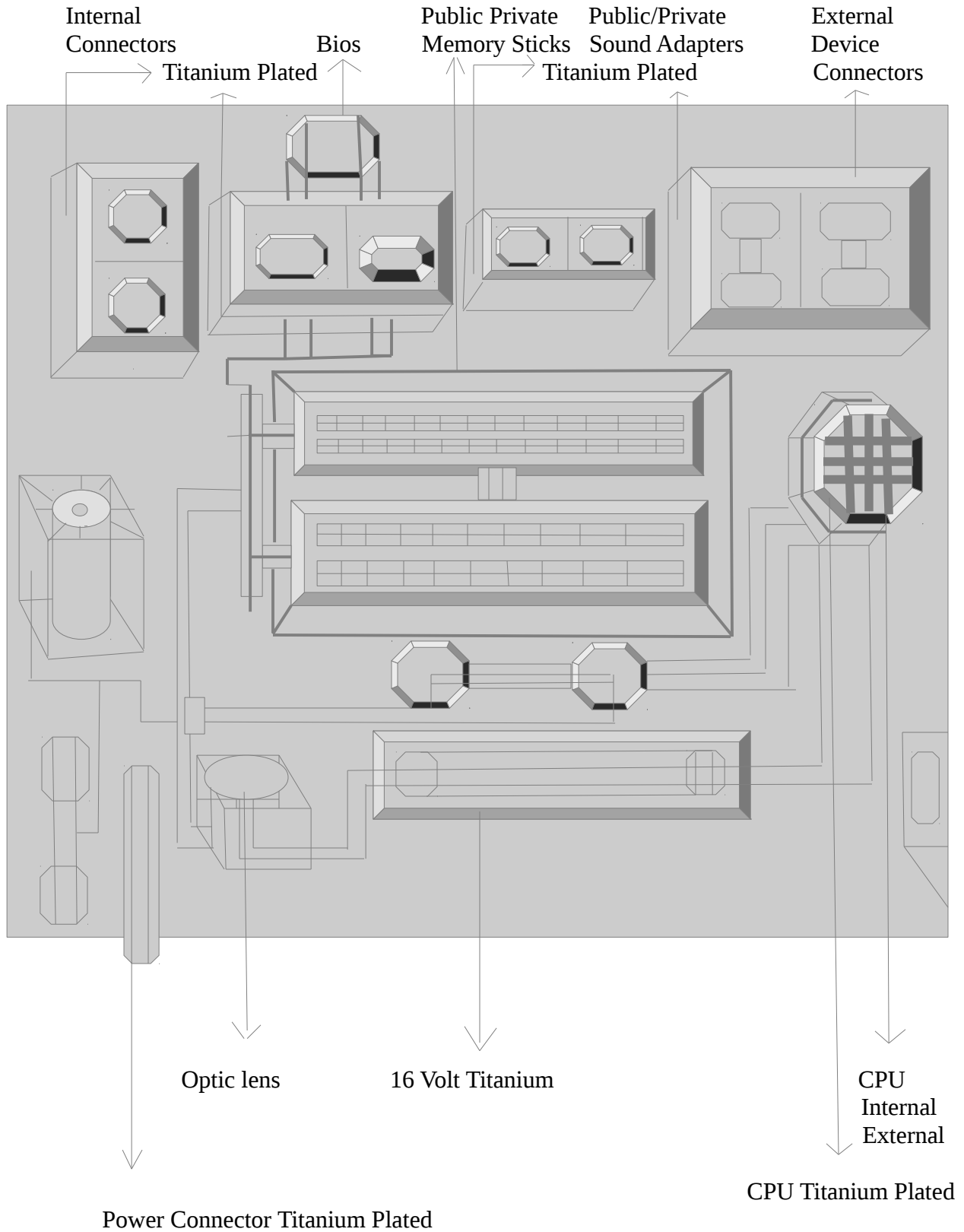
Inner Shell Bios Dynamic Instruction Sets

Setting #	# of bit Instruction Set
3	4096
4	$4096/16 = 256$ Encrypt
5	65536
6	$65536/16 = 4096$ Encrypt

Model Super Sonic 75 Motherboard- Design 2-A

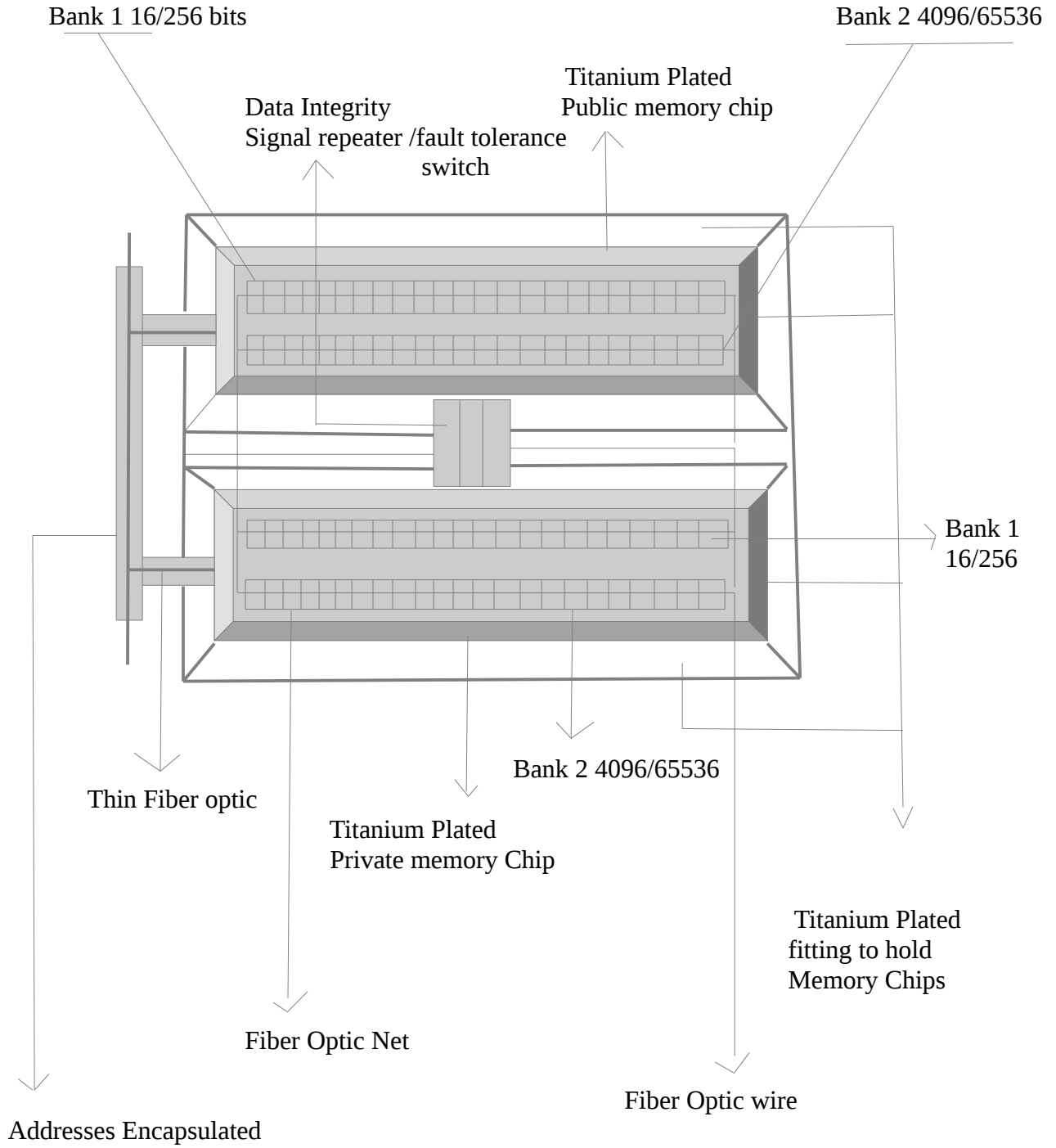


Model Super Sonic 75 Motherboard- Design 3-A

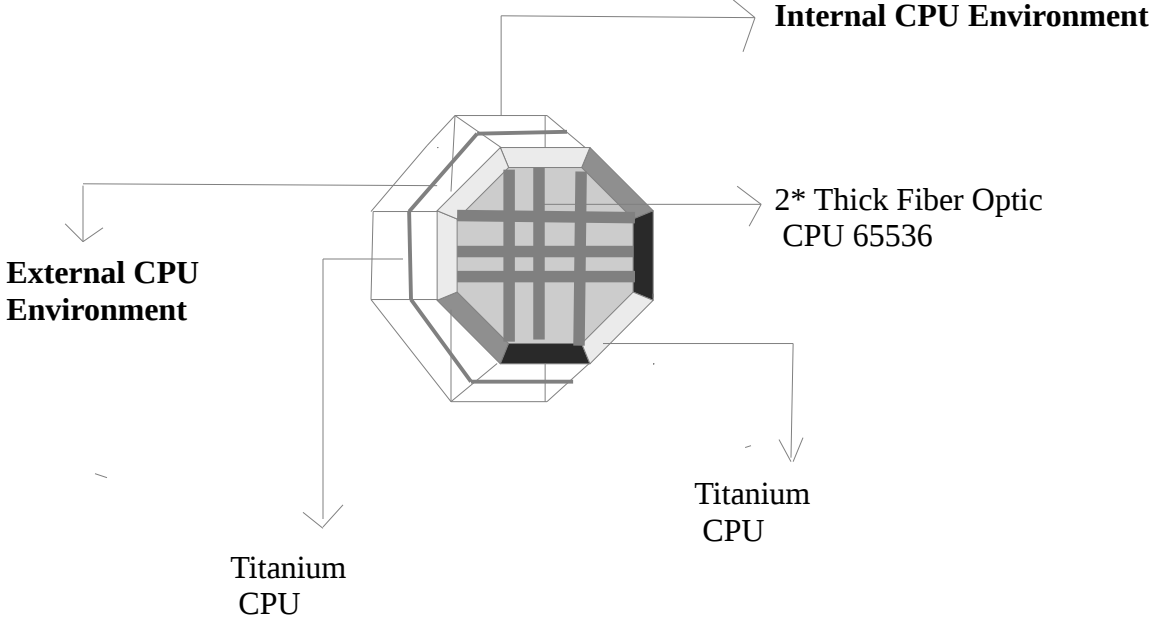


Model Super Sonic 75 Motherboard- Design 4-A

16/256/4096/65536 Bit Instruction Sets



Model Super Sonic 75 Motherboard- Design 5-A



Overview of Design

I would like to provide a brief overview of this design. I created a BIOS with a different Algorithm that has the following advantages

- 1). Bios Instruction sets are not Static but Dynamic.
- 2). The Bios a 65536 Register and loads dynamic Instruction Sets.
- 3). Instruction Sets can be set dynamically 16,256,4096,65536, and Encrypted
- 4). Created a Internal and External Bios employing a outer and Inner Shell.
- 5). The Inner Shell Bios may in the future utilize 4096 and or 65536 Instruction sets.
- 6). Inner Shell Bios Instruction sets can have encryption deployed.

I have completed Research and found no 256 bit Register so this would work well creating a 65536 bit register with 256 bit instruction sets as a example this creates a level of security encrypting commands within the outer shell bios and in the future using and employing the Inner shell Bios. The model is based on the principle of time formation taking points and forming them and creating a Register as a space and than using Dynamic Instruction sets that is based on Intelligent Design Mechanisms.

The Algorithm employed in the past created the following”

8 Bits = 1 Byte
2 bytes = 1 Character

The first Operating Systems employed the following Architecture

8 Bit
16 Bit
32 Bit
64 Bit

This is Multiplied by 2 to extend the Register. **My proposal is the following:**

1 * 16	= 16 Bits
16 * 16	= 256 Bits
16 * 16 * 16	= 4096 Bits
16 * 16 * 16 * 16	= 65536 Bits

I am basically exponentiating by 16 up to the 4th power this creates a Algorithm that creates a different method so thus “swapping”, “borrowing” “updating” a Registry would not work because the way or method utilized would not accept the traditional types of activities. The method presently used allows for this. The Registry is static not dynamic making it predictable. Please find the Specification Chart below with a comparison.

Old Method/bits	Proposed New Method/Bits
8	16
16	256
32	4096
64	65536

Chapter 2

Chart and Spec Comparisons

Bit Strengths 2-B

Proposed New Algorithm Specifications

# of bits	# of Bytes	# of characters/field
16	1	1
256	16	4
4096	256	16
4096	256	16/4 = 4 Encrypted
65536	4096	256/4= 64 Encrypted

Presently most operating software utilizes 64 bit operating systems Linux, Windows, etc. This would break down to the following:

64 Bits = 8 Bytes
8 Bytes = 2 Characters

Please remember this is a static Registry not Dynamic. Please also note the new method allows for Character Encryption and it is available at 4096/65536 which allows for compression of fields. Check-sums can now be run in encrypted mode or character extensions thus probability of hacking into a hard disk and accessing the cache decreases by 50 percent just by changing the Algorithm and Character formatting and presentation.

Chapter 3

Bios Processing

Rem This is a system level program that is not visible and is determined by the user making the choice. I am checking to either load the External or Internal Bios.

Bios-processing

Bios-External =h

Bios-Internal =i

```
0 = "off"
1 = "on"
if h = "on"
goto Bios-space-1
else
if i = "on"
goto Bios-space-2
else
if h and i = "off"
exit
```

rem Area-spaces checking conditions on or off

Bios-space-1

```
0 = "off"
1 = "on"
j = A-bios
k = B-bios
```

```
rem set switches to on or off and check conditions
if j = "on"
set 16-bits abios
move "16" j
else
if k = "on"
set 256-bits abios
move "256" k
else
if j and k = "off"
```

goto **Barrys Scientific Products Bios Screen**

Bios-space-2

0 = "off"
1 = "on"
w = C-bios
x = D-Bios-Encrypted256
y = E-Bios
z = F-Bios-Encrypted4096

rem set switches to on or off and check conditions
if w = "on"
set 4096-bits
move "4096" w

else
if x = "on"
perform x=4096/16
set 256enc-bits
move "256" x
else
if y = "on"
set 65536-bits
move "65536" y
else
if z = "on"
perform z=65536/16
set 4096enc-bits
move 4096enc-bits z
if w,x, y, z = "off"
exit

Barrys Scientific Products Bios Screen

Barrys Scientific Based Products Bios Screen

Digital Time Clock Setting

Security and System Level Encryption

Hardware Configuration

Hardware Advanced Settings

Advanced Settings

Default Settings

Exit and Settings Save

I will now present some practical Mathematical Equations in the next chapter.

Chapter 4

Mathematical equations

I would now like to present a method of using bit compression by taking the Internal and External Bios creating a shelled equation.

$$\begin{aligned} \textcircled{a} &= \sqrt{(a * b) / q1} \\ &\quad / q2 \\ a &= \text{Internal Bios} \\ b &= \text{External Bios} \\ q1 &= \text{1}^{\text{st}} \text{ area of space} \\ q2 &= \text{2}^{\text{nd}} \text{ area of space} \end{aligned}$$

$$\textcircled{a} = \sqrt{\textcircled{a}}$$

The Equation takes the number of bits Internal and External and Multiplied and is than square rooted it is than divided by the area of space this is the 1st level it than goes to the next level and is square rooted example would be the 1st level 256 squared to 16 multiplied by 1 area of space and than is taken to the 2nd level and is squared again to 4. To simplify this going from 256 to 4 if I reverse the process I go from 4 * 4 = 16 = 16* 16 = 256.

I will now present my final thoughts in the next chapter.

Chapter 5

Final Thoughts

Final Thoughts

I should point out the major differences between the SS65 and the SS75 Motherboard Design. The differences are the following:

- 1). SS75 Motherboard Design needs 16 volts instead of 12 for the SS65.
- 2). The Registries are 65536 SS75 instead of 32768 SS65.
- 3). Instruction Sets for the SS 75 are Dynamic and based on Pre Bios settings.
- 4). SS75 offers a different Algorithm and Bit Representation
- 5). SS75 offers a different Compression that uses Shells.
- 6). SS75 Introduces a New Character through the Mathematical Equation.
- 7). SS75 Introduces New Internal and External Components

The process of the Internal and External Bios uses points within the Bios. These points are 8 per environment Internal and External total 16. The next step is The registry or space of 65536 is created and Instruction sets are loaded Dynamically 4 points per Instruction set and 2 instruction settings are loaded per Environment Internal is 2 External is 2.

This is similar to Sub-Atomic particles that either choose to remain within the Atomic Atom or begin the decay process forming the External space.

I hope that those who have decided to engage in these types of actions will understand that when you wish ill intent on a person such as malicious System level virus attacks it may be turned against you into something that is used for good.

I would like to finally state it is possible to load a Dynamic Register and creating 65536 and contract and expand but this is a different subject area because this deals with spatial areas and would lose sight of the main theme in this Scientific work.

If you wish to see other works they are published and copyrighted on www.PublishResearch.com.

Please use Google the following keywords:

Barry Crouse Network Topology Design

Barry Crouse Intelligent Design

Barry Crouse Physics

These works are down loadable directly from Google

Barry L. Crouse Ph.d Computer Information Systems

09/17/2015

Email barry.crouse@yandex.com

If you enjoy this work, I would like to invite you to www.PublishResearch.com to read other Scientific works !

If you Google the following keywords, You will find these works listed

Barry Crouse Network Topology Design

Barry Crouse Physics

Barry Crouse Intelligent Design

Thank you for reading this work.

Barrys Scientific Based products is a State Registered Trademark of the State of Washington

