

**Barrys SS 75 with Dynamic Registries/Special Project # 2**

**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. This is the 2<sup>nd</sup> solution step in creating a overall secured computer environment. A registry that Contracts and expands in accordance with Intelligent Design Principles that are dynamic in nature.

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

- 1). Creating a Dynamic Registry that prevents borrowing, stealing, and or data theft.
- 2). Creating Registry that is Dynamic and adjusts to it's environment accordingly.
- 3). Creating a Registry or Area of space that provides the object or Instruction sets that are shelled within the registry.

I have provided a Algorithm for the Dynamic Registry along with some coding examples to make the objects-Instructions set and Registry Area of space work together in harmony. Please also note I have provided a way to encrypt the Registry in Chapter 2 also, I have provided a special Chapter 3 regarding Time Origins that is applicable to this special project. I have attempted to incorporate theory and practical application in this work.

Thank you for reading this work!

## **Table of Contents/Special Projects # 2**

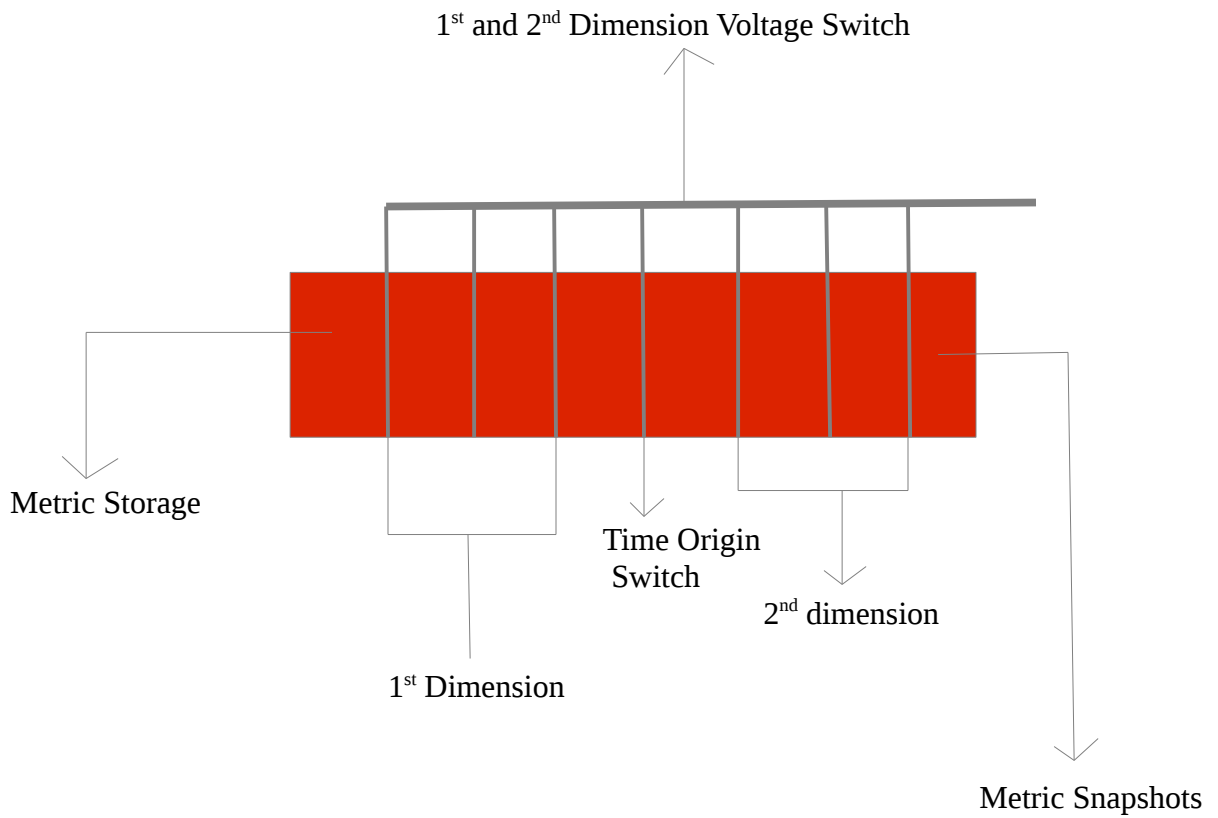
<b>Chapter 1</b>	<b>Visual Design</b>
<b>Chapter 2</b>	<b>Registry Processing</b>
<b>Chapter 3</b>	<b>Time origin switch</b>
<b>Chapter 4</b>	<b>Final Thoughts</b>

## **Chapter 1**

### **Visual Design**

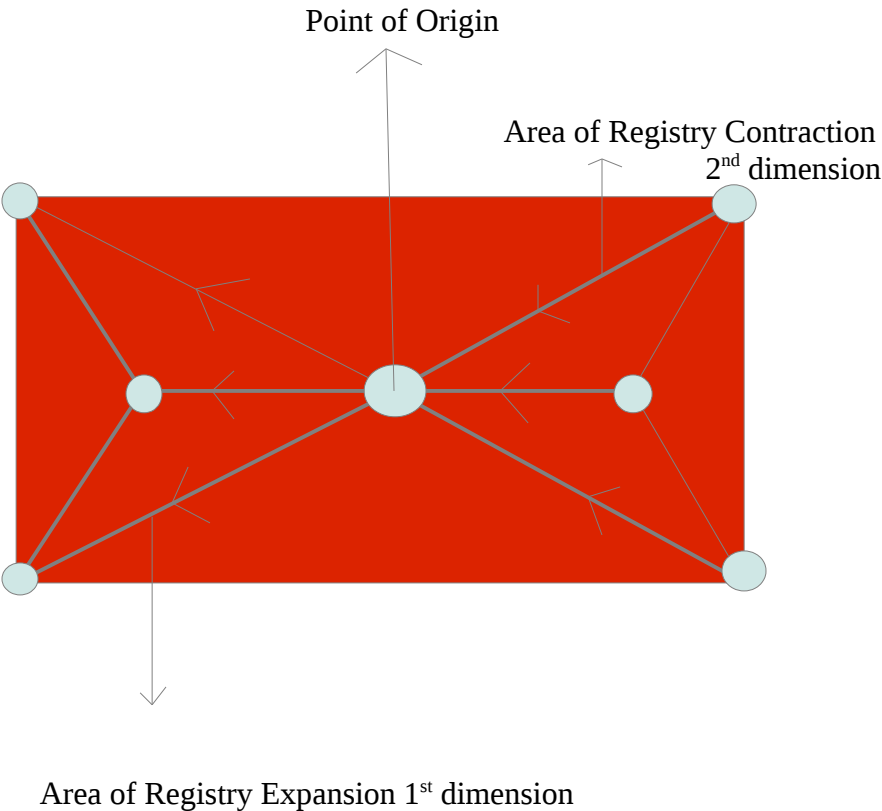
# Model Super Sonic 75 Motherboard- Design 1-A

## Time Origin Voltage Switch



**Model Super Sonic 75 Motherboard- Design 2-A**

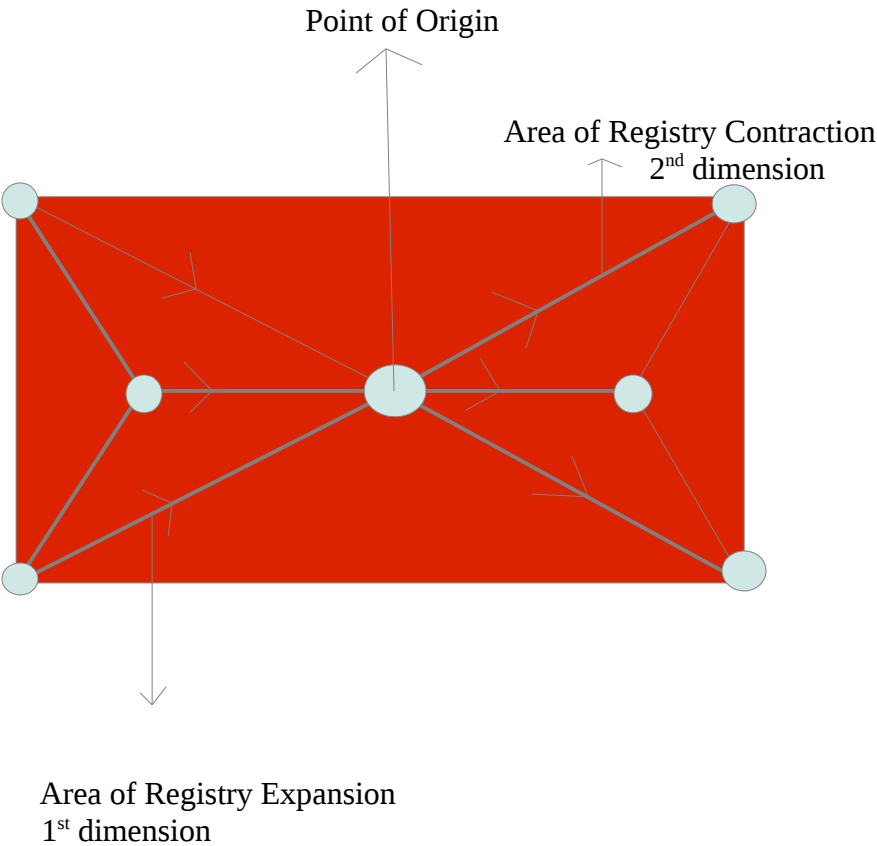
**Registry Storage Chip Method 1**



<b>Registry Area-spaces</b>	<b>Registry Size</b>	<b>Point of origin dimension</b>
Expansion	16	1st
Expansion	256	1st
Expansion	65536	1st
Contraction	256	2nd
Contraction	16	2nd
Contraction	4	2nd

**Model Super Sonic 75 Motherboard- Design 3-A**

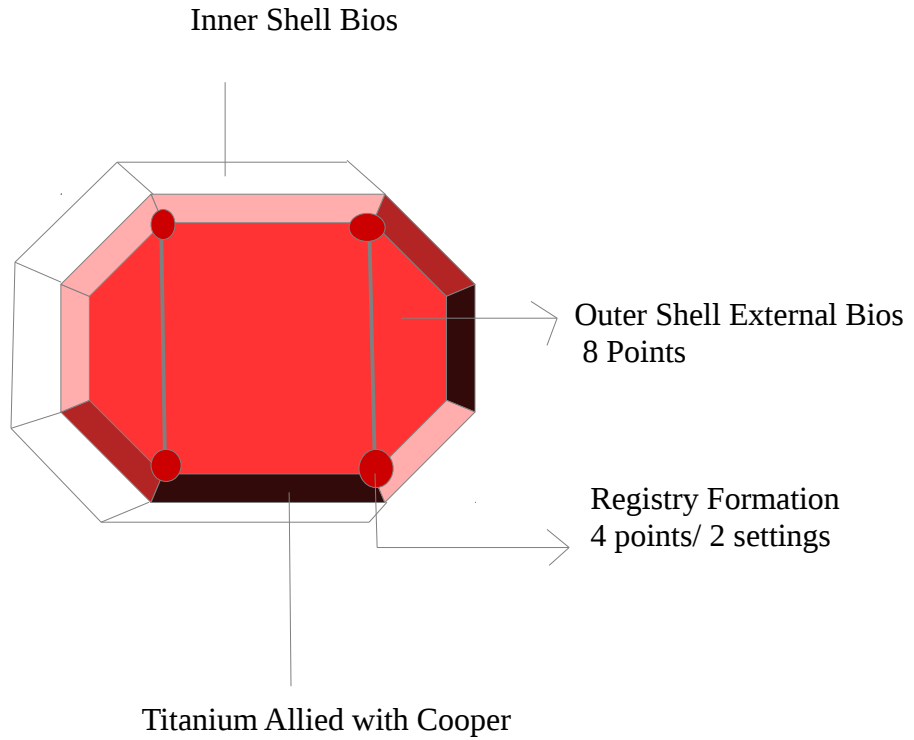
**Registry Storage Chip Method 2**



<b>Registry Area-spaces</b>	<b>Registry Size</b>	<b>Point of origin dimension</b>
Contraction	256	1st
Contraction	16	1st
Contraction	4	1st
Expansion	16	2nd
Expansion	256	2nd
Expansion	65536	2nd

## Model Super Sonic 75 Motherboard- Design 4-A

### Bios Chip



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



## BIOS Spec Charts

### Outer shell Bios Dynamic Instruction Sets

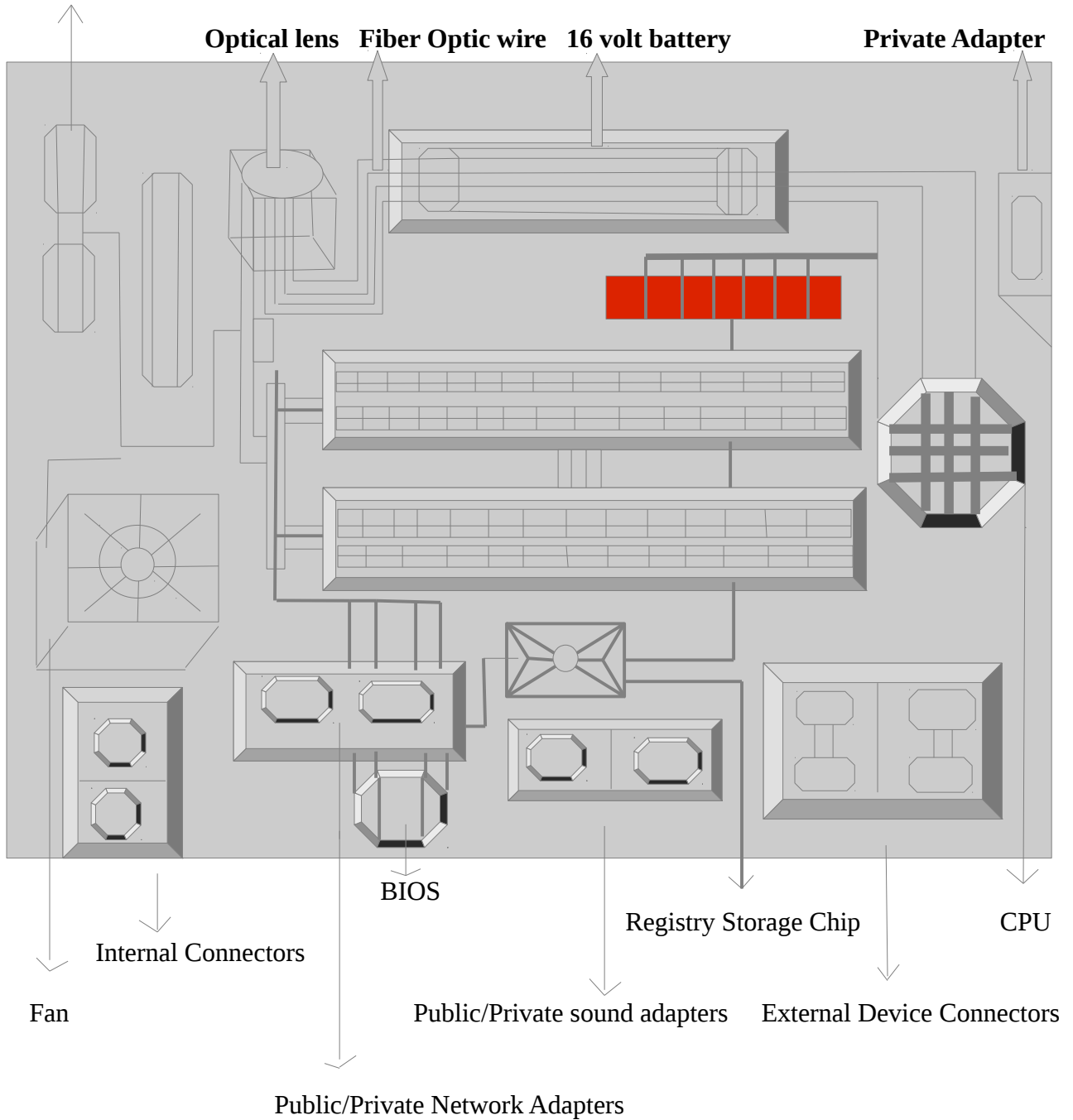
Setting #	# of bit Instruction Set
1	16
2	256

### Inner Shell Bios Dynamic Instruction Sets

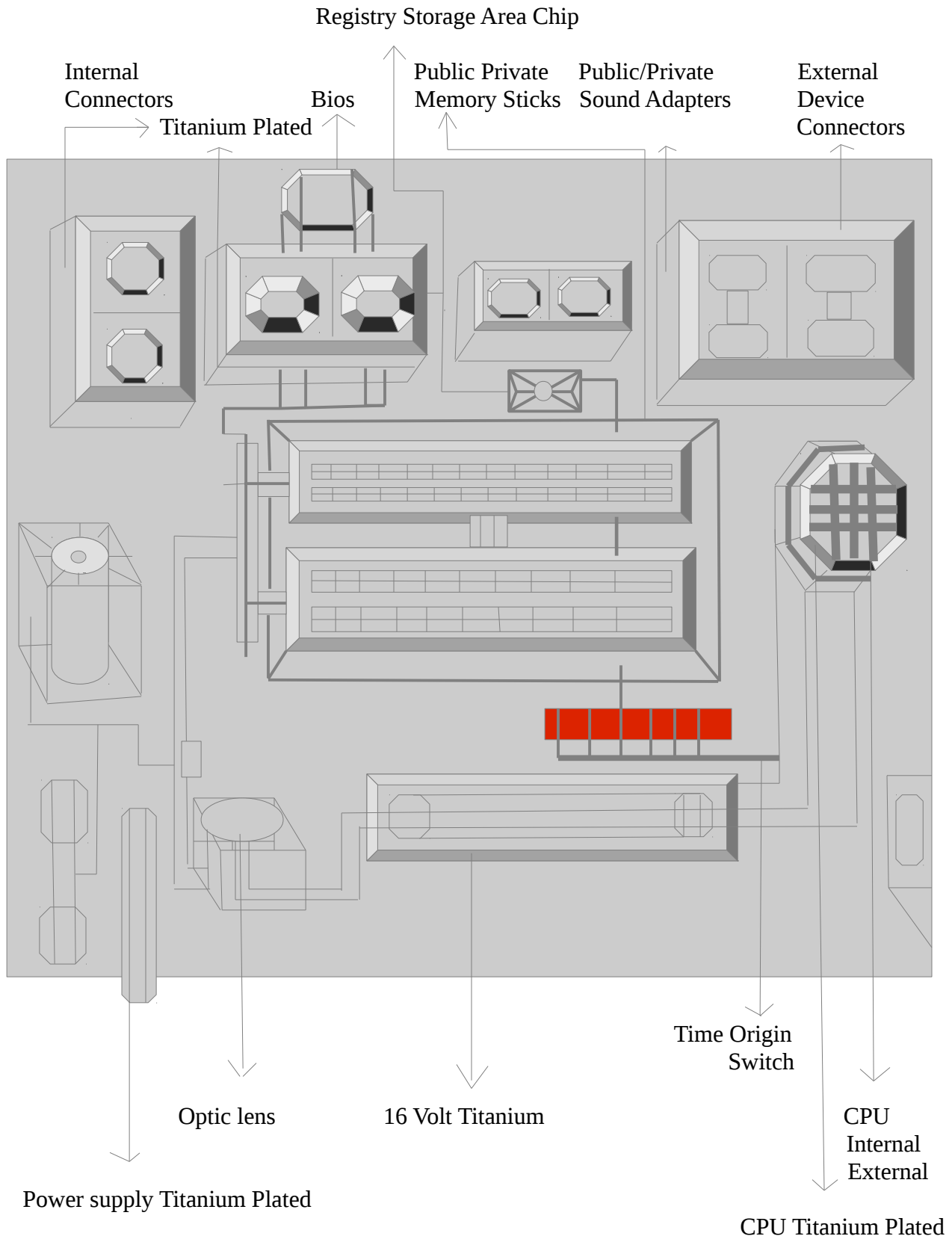
Setting #	# of bit Instruction Set
3	4096
4	$4096/16 = 256$ Encrypt
5	65536
6	$65536/16 = 4096$ Encrypt

# Model Super Sonic 75 Motherboard- Design 5-A

## Hard Drive Controllers

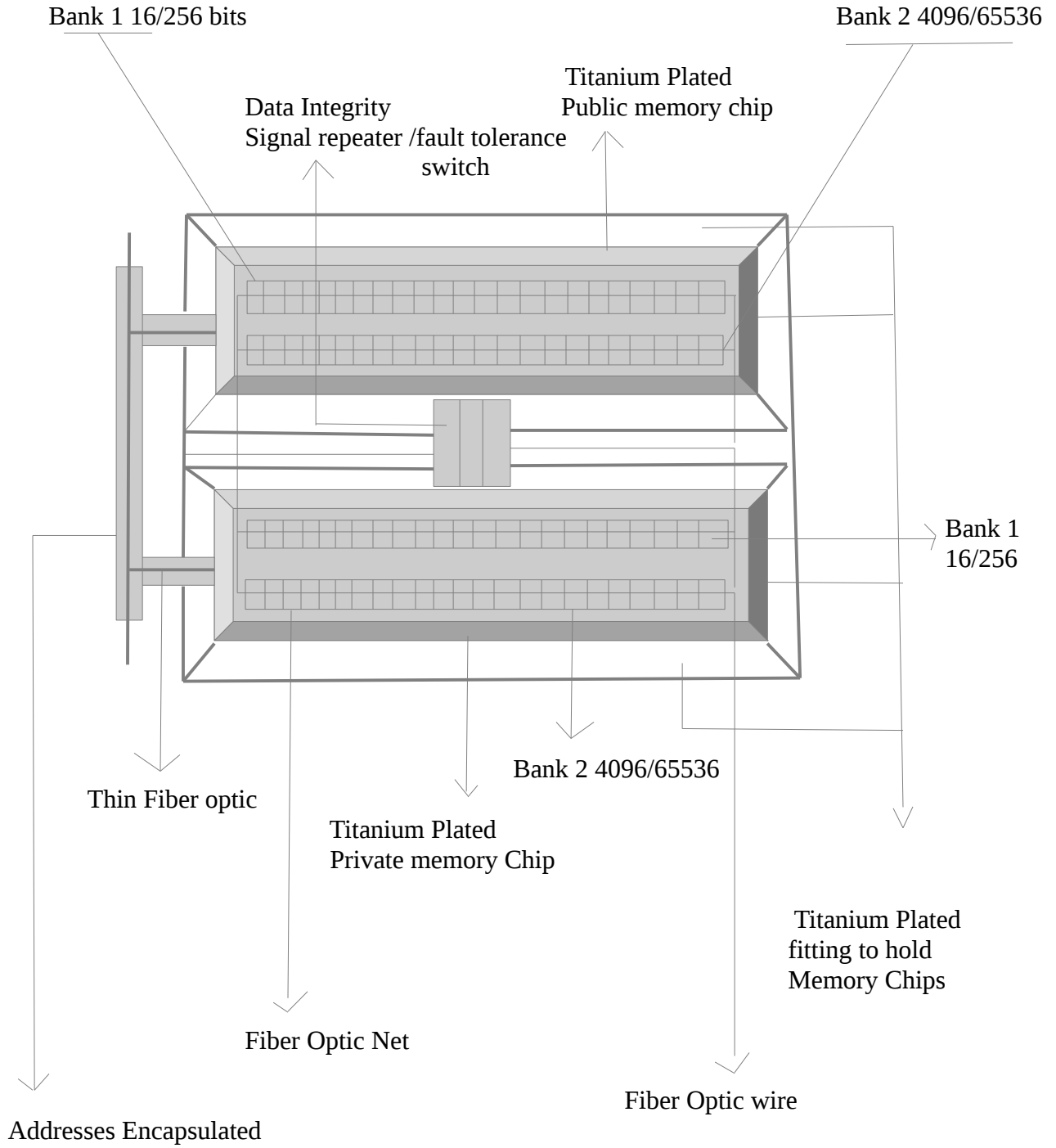


# Model Super Sonic 75 Motherboard- Design 6-A

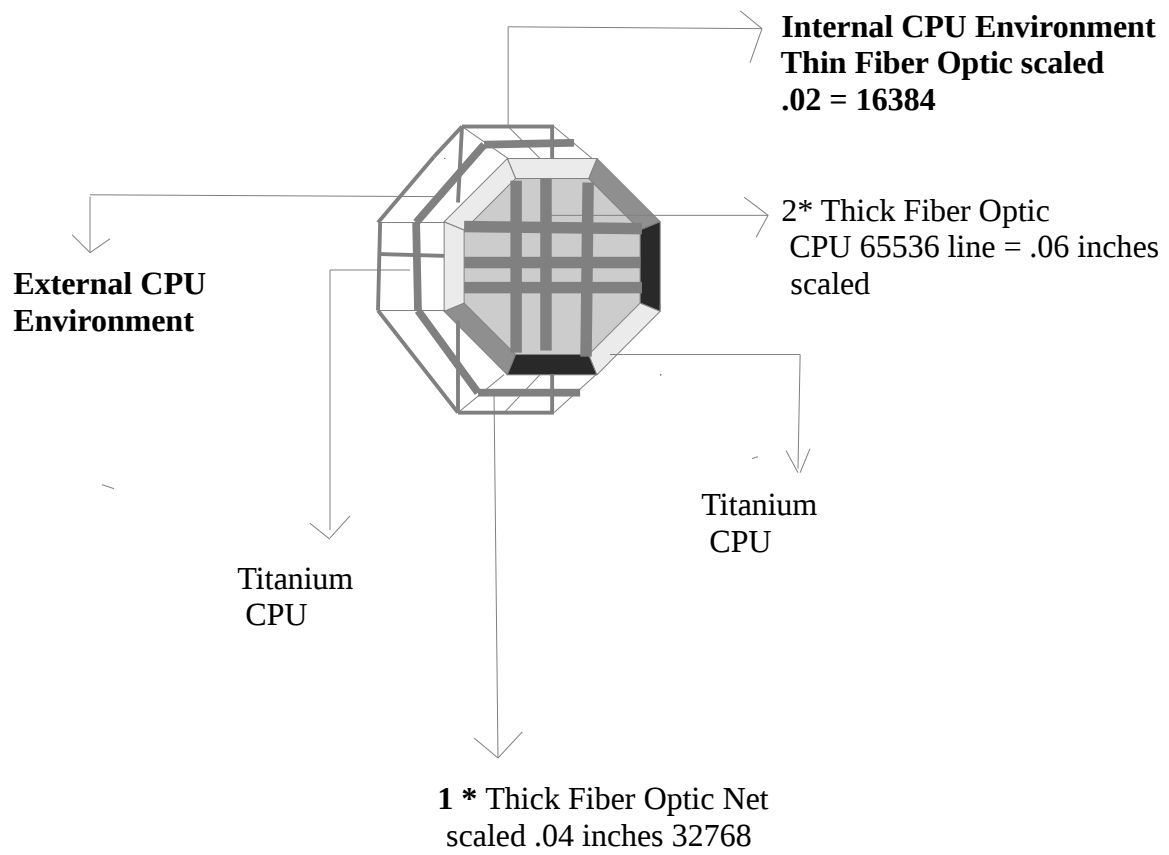


# Model Super Sonic 75 Motherboard- Design 7-A

## 16/256/4096/65536 Bit Instruction Sets



## Model Super Sonic 75 Motherboard- Design 8-A



## Overview of Design

I would like to provide a brief overview of this design process regarding dynamic Registries.

- 1). Registries are Dynamic not static this helps prevent data theft.
- 2). Registries use a point of origin to determine dimension along with Expansion and contraction.
- 3). Registries use a base 16 instead of 8 with expansion and contraction.
- 4). The point of origin is similar to a time line but does emphasizes Intelligent Design principles energy in this Scientific Work goes from 2<sup>nd</sup> to 1<sup>st</sup> decay but expands in the 1<sup>st</sup> dimension. See chart 1-A
- 5). The Registry is flexible to Dynamic Instruction sets so long as the spatial area is greater than the object-Instruction set **in this work**.
- 6). The 1<sup>st</sup> dimension to reach the 2<sup>nd</sup> must decay and than expand from a energy point of view see chart 2-A
- 7). The Time Origin Switch uses a metric system and snapshot to determine what information to pass to the Registry Storage chip.
- 8). Instruction sets represent objects and Registries represent spatial areas in this work the Registries or Area of space will be greater than the Instruction sets in size. Please note it maybe possible to make the object greater than the area but this would defeat the purpose or **main idea of this work**.

In this Scientific work, I will utilize the process of going from the 2<sup>nd</sup> dimension to 1<sup>st</sup> decay than expand see chart 1-A. Please note the amount of programming code in the next chapters as to the reason for not coding the 1<sup>st</sup> to 2<sup>nd</sup> dimension contract and than expand 1<sup>st</sup> to 2<sup>nd</sup>.

Please note the object represents the Instruction sets similar to particles and the Registries represent areas of space. I would also suggest it is possible to take areas of space and divide them into Quadrants and applying different color spectrum's or different levels of heat or energy to create a more dynamic and secured environment. This can be achieved that others may write.

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 expand the Register. **My proposal is the following:**

**Method 1 Contract 2<sup>nd</sup> Dimension and Expand 1<sup>st</sup> dimension**

$\sqrt{65536}$	= 256
$\sqrt{256}$	= 16
$\sqrt{16}$	= 4
$1 * 16$	= 16 Bits
$\sqrt{(16 * 16 * 16)}$	= 256 Bits
$16 * 16 * 16 * 16$	= 65536

Please note the algorithm applied is for both objects instruction sets and Areas of space or registries. This shows that energy levels are dynamic and does not subscribe to the one size fits all mentality energy is unevenly distributed and does not subscribe to uniformity principles that have led to many problems in Computer Sciences and Mechanical Engineering issues example data theft, borrowing, stealing and other forms of data piracy. In this project, I have provided a Encryption technique for characters format and presentation expanding the characters 16 to the 3<sup>rd</sup> power and contracting it in one event. Please note the spatial area now has the ability to be encrypted as well.

## **Chapter 2**

### **Registry Processing**



In this chapter, I will provide a pseudo program that demonstrates contraction 2<sup>nd</sup> dimension to 1<sup>st</sup> Expansion along with a Pre-Bios Screen and then a Bios Screen itself. The process used is first the area of space or registry is tested and must be greater than the object or Instruction set. The point of origin is the Switch it determines the dimension whether to decay or expand as illustrated in the charts. The switch tests the methods used and acts as a gateway either allow or deny the object to pass through dimensions.

### Load Pre-Bios Registry table

Variable	Character/ # bits	Dimension	Original representation
A-Contraction	4	2	16
B-Contraction	16	2	256
C-Contraction	256	2	65536
D-Expansion	16	1	16
E-Expansion	256	1	4096
F-Expansion	65536	1	65536

### Load Pre-Bios object Table

Variable	Character/ # bits
A-Bios	4
B-Bios	16
C-Bios	256
D-Bios	16
E-Bios-encrypt	256
F-Bios	65536

Both the Areas and object tables are loaded. I wanted to highlight something the E-bios object calculation is as follows  $\sqrt{16 * 16 * 16} = 4096$  if you square root this it is 256 this allows for the encryption mechanisms to take place in the Spatial Area. In my example, I created a Registry or area of space at 65536 than I took the object E-bios 256 instruction set that is encrypted creating a 65536 area and object encrypted 256. This is only a example there are other combination you could use to create other dynamic size instruction or object sets along with spatial areas that are dynamic.

Voltage-switch-1 = vs1

Voltage-switch-2 = "on"

Voltage-switch-3 = "off"

Voltage-method-a-1 = "A-Contraction"

Voltage-method-b-1 = "B-Contraction"

Voltage-method-c-1 = "C- Contraction"

Voltage-method-d-2 = "D-Expansion"

Voltage-method-e-2 = "E-Expansion"

Voltage-method-f-2 = "F-Expansion"

Rem test switch for on or off

```
If Voltage-switch-2 = "on" and "1"  
goto method-testing-1  
If Voltage-switch=3 = "off" goto exit  
else  
exit
```

**method-testing-1**

If vs1 = A-Contraction goto Pre-Bios Object setting  
else  
If vs1 = B-Contraction goto Pre-Bios Object setting  
else  
if vs1 = C-Contraction goto Pre-Bios Object setting  
else  
if vs1 = D-Expansion goto Pre-Bios object setting  
else  
if vs1 = E-Expansion goto Pre-Bios object setting  
else  
if vs1 = F-Expansion goto Pre-Bios object setting

{

**Load Pre-Bios Object Menu**

```
*****  
**                                                                 **  
**      Barrys Scientific Based Products                        **  
**                                                                 **  
**      Select “ A-Bios 4”                                       **  
**                                                                 **  
**      Select “ B-Bios 16”                                       **  
**                                                                 **  
**      Select “ C-Bios 256”                                       **  
**                                                                 **  
**      Select “ D-Bios 16”                                       **  
**                                                                 **  
**      Select “ E-Bios-encrypt-256                                       **  
**                                                                 **  
**      Select “ F-Bios 65536 ”                                       **  
**                                                                 **  
*****
```

**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.**

## **Error-handling-routine**

```
If A-Bios less than or equal A-Contraction  
goto Bios-processing  
else  
if B-Bios less than or equal B-Contraction  
goto Bios-processing  
else  
if C-Bios less than or equal C-Contraction  
goto Bios-processing  
else  
if D-Bios less than or equal D-Expansion  
goto Bios-processing  
if E-Bios-encrypt-256 less than or equal E-Expansion  
goto Bios-processing  
else  
if F-Bios less than or equal F-Expansion  
goto Bios-processing  
else  
select F-Expansion  
select E-bios-encrypt-256  
rem automatic default area set to 65536 bios instructions set to 256
```

## **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  
w =C-bios

rem set switches to on or off and check conditions  
if j = "on"  
perform j =  $\sqrt{16}$   
set j="4"  
move "4" j  
else  
if k = "on"  
perform k=  $\sqrt{256}$   
set k="16"  
move "16" k  
else  
if w= "on"  
perform  $\sqrt{65536}$   
set w="256"  
move "256" w  
if j = "off"  
else  
if k = "off"  
else  
if w = "off:  
goto Bios-space-2  
exit

### **Bios-space-2**

0 = "off"  
1 = "on"  
x = D-Bios  
y = E-Bios-encrypt-256  
z = F-Bios  
rem set switches to on or off and check conditions  
if x = "on"  
perform w = 16 \* 1  
set x= 16  
move "16" x

```
else
  if y = "on"
  perform y= 16*16*16
  perform y =  $\sqrt{4096}$ 
  set y= 256
  move "256" y
  else
  if z = "on"
  perform z= 16*16*16*16
  set z=65536
  move "65536" z
  else
  if x="off"
  else
  if y="off"
  else
  if z="off"
  goto Barrys Scientific Products Bios Screen
  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**

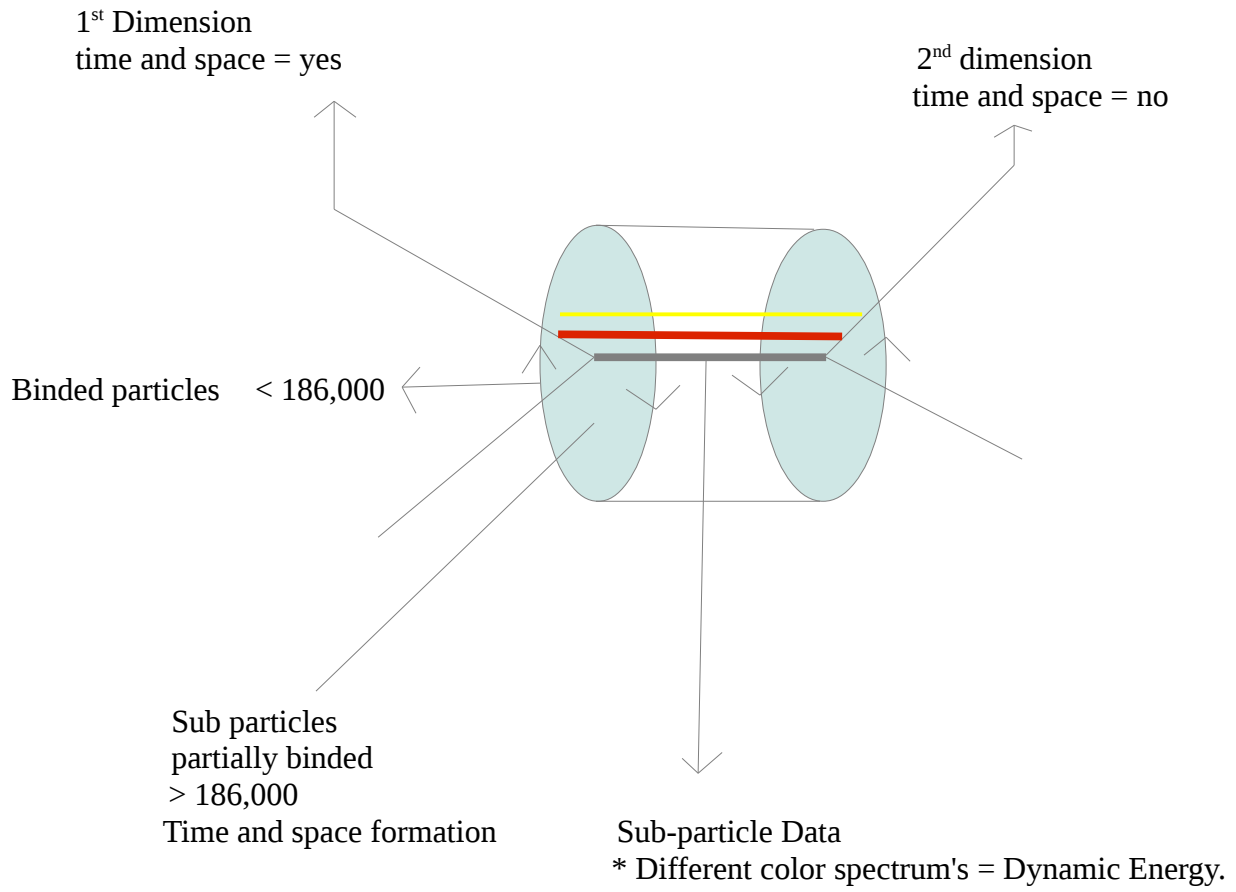
**Default Settings**

**Exit and Settings Save**

## **Chapter 3**

### **Time origin switch**

# Time and Space Visual Chart





## Time and Space Origins

This is a topic that will play a 21<sup>st</sup> Century role in the development of Physics, Computer Sciences, Engineering, and Mathematics.

In our present day understanding of time and space, We are taught that time and space suddenly appeared commonly called the Big bang Theory. This violates the laws of Sir Issac Newtons 2<sup>nd</sup> law of Therm o-Dynamics Energy must come from somewhere. Time and Space are formed from partially binded particles that exceed the speed of light these sub-atomic particles have Intelligent Design Metrics that decide to expand or decay to allow our Universe as we know it to exist. These Particles whether they are Sub or nano are the basis and building blocks of our Universe they do not suddenly appear. If they were to appear in the 1<sup>st</sup> They must go through a decay process to be binded to our Universe much like a switch in a Computer Network Environment Test have been shown in various labs that demonstrate these particles exceed the speed of light and have multiple tunnel paths. The color spectrum shows the different levels of energy that shows how far along they are if you are astute enough.

These Nano and Sub-Atomic particles in their present forms have no concern for time or space only when they start the decay process than it is binded to time and space thus Energy uses a Metric switch that allows for the formation of time and space and allows our Universe to exist. The switch is smart it checks the energy of our Universe because to many of these particles could cause a accelerated decay of our Universe this is similar to taking a snapshot in Computer Sciences to measure levels of packets and throughput a self check and correction process.

Thus this paper supports Sir Issac Newton's 2<sup>nd</sup> law of Thermo Dynamic because Energy comes from a alternative dimension and allows for decay and or expansion it does not suddenly appear out of nowhere. I would like to emphasize a main point if you cannot accept this line of reasoning than to attempt to deploy this application will prove to be futile and pointless and above all counter productive. **This is based on God's Design application and or principles.**

Many Secular Scientist have been presented factual data demonstrating nano and Sub-particles are present it is there but we cannot see it because it exceeds the speed of light and in fact is measured by Nano's billionths of seconds hint  $> 186,000$  mph. Thus the argument of Time and Space appear out of no where cannot be viewed as serious anymore but a Secular Religious viewpoint to keep one's job along with position, power, and one's hefty salary.

Please note I hope that those who are of the Secular Nature understand that Science is based on evidence and factual data and your theory has failed that test and should not be valid anymore by the same token I understand why they fight so hard to keep it intact so that the "System" is in harmony and agreement for the purposes of retaining power, position, along with money. I have been asked in various ways why I wish not to continue anymore because of the reasons listed above in essence the system area space is larger than me hint (object) a Internal lesson that is applicable it is okay to have a little chuckle and a good laugh so go ahead.

I have recently been shown a Unified theory of Wave Particles and Quantum Atoms. The problems with this are that wave Particles have a limited constraint called time and space. Quantum Theory uses Sub-scales that overcome these limitations where time and space are irrelevant thus we would have a rapid decay of particle or in this sense q-bits. Old Midwest proverb 1 bad apple Wave particle can spoil the rest Quantum Atoms.

## **Chapter 4**

### **Final Thoughts**

## Final Thoughts

I should point out the major differences between the SS75 and the SS75 version 1.2 Motherboard Design. The differences are the following:

- 1). The SS 75 version 1.2 utilizes a Time origin Switch the regular SS 75 does not have this feature.
- 2). The SS 75 Version 1.2 utilizes a Registry Storage Chip regular SS 75 does not.
- 3). The SS 75 Version 1.2 uses a Dynamic Registry and Instruction Sets.
- 4). SS 75 1.2 has two tables Objects and Spatial area. Regular SS 75 uses 1 table.
- 5). The Error Handling Routine has been added to harmonize Objects and Spatial Areas check for conditions and capable of loading system defaults.
- 6). Bios Instruction sets objects and Registries Spatial Areas are not in a constant but Dynamic States ;therefore, limitations of time and space have been overcome. Time and Space do not have to equal each other only if it is chosen as a metric choice.
- 7). The Spatial or Registry Area of Space can now be encrypted in SS75 Version 1.2.

The process of object and Spatial Reconciliation has been achieved in this work. The Time Origin Switch takes snapshots and Stores them. The switch has 6 settings 1<sup>st</sup> and 2<sup>nd</sup> Dimension and is considered a smart switch that makes determinations of what wire and or dimension to utilize. The switch is where areas of space become known and is sent to the Bios Storage Chip-object based. I have also provided a way to encrypt the Registry or spatial area to help prevent data theft, stealing, and or borrowing of one's computer and is applicable to physics and or Mathematical processes.

The best way to understand this is the Time Origin switch is the Area of Space and the Bios Storage Chip is the object where the Instruction Sets are housed or shelled.

This paper utilizes a method of Contraction from the 2<sup>nd</sup> and than Expansion in the 1<sup>st</sup> . This method demonstrates applicable principles that can be deployed through the usage of Dynamic settings objects and Registries areas of space. I would also like to state that there are other methods that can be deployed such as the 1<sup>st</sup> dimension contracting and the 2<sup>nd</sup> dimension expanding also it is possible to deploy a object larger than the space itself but this would detract from the main idea in this paper to provide a method to harmonize Dynamic objects and Spatial areas in this case Instruction sets and Registries.

If you wish to see other works they are published and copyrighted on [www.PublishResearch.com](http://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

10/13/2015

Email [barry.crouse@yandex.com](mailto:barry.crouse@yandex.com)

If you enjoy this work, I would like to invite you to [www.PublishResearch.com](http://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