Barrys SS-65 Ver 1.3 Motherboard Design

by

Barry L. Crouse

Introduction

Thank you for taking the time in reading this Technological Work. I will be doing things a little differently in this paper.

This work will use a lot of Visual Designs and Mathematical Equations to promote previous ideas and concepts along with creating Logic Gateways and Memory addressing schemes describing the method and process to be used in this technological work. I will be using in this work a direct Interface converting Solar Energy to Mechanical Bits and going directly to the Motherboard itself. In previous works, I used Solar Energy to a Interface than to the Motherboard. The direct Interface allows for 32768 bits to be processed coupled with my equation **Barrys Mechanical Space** demonstrating a practical application combinning Computer Siences and Mechanical Engineering.

I have updated this Motherboard Design to include the following :

1). 3 Layer CPU including shared resources, Node Points, and Data Strings.

2). Added Memory sticks to now have a total of three on the board with Public, Private, and Shared memory addresses.

3). Created a 256 Asymmetrical Bit scheme for memory addresses.

4). Testing Memory address switches with some simple logic program including Address switch entanglement.

5). Low End Energy for Gateway processing is 3072 bits instead of 2048.

Table of Contents

Chapter 1

Visual Design

Solar to Mechanical Energy 1-A

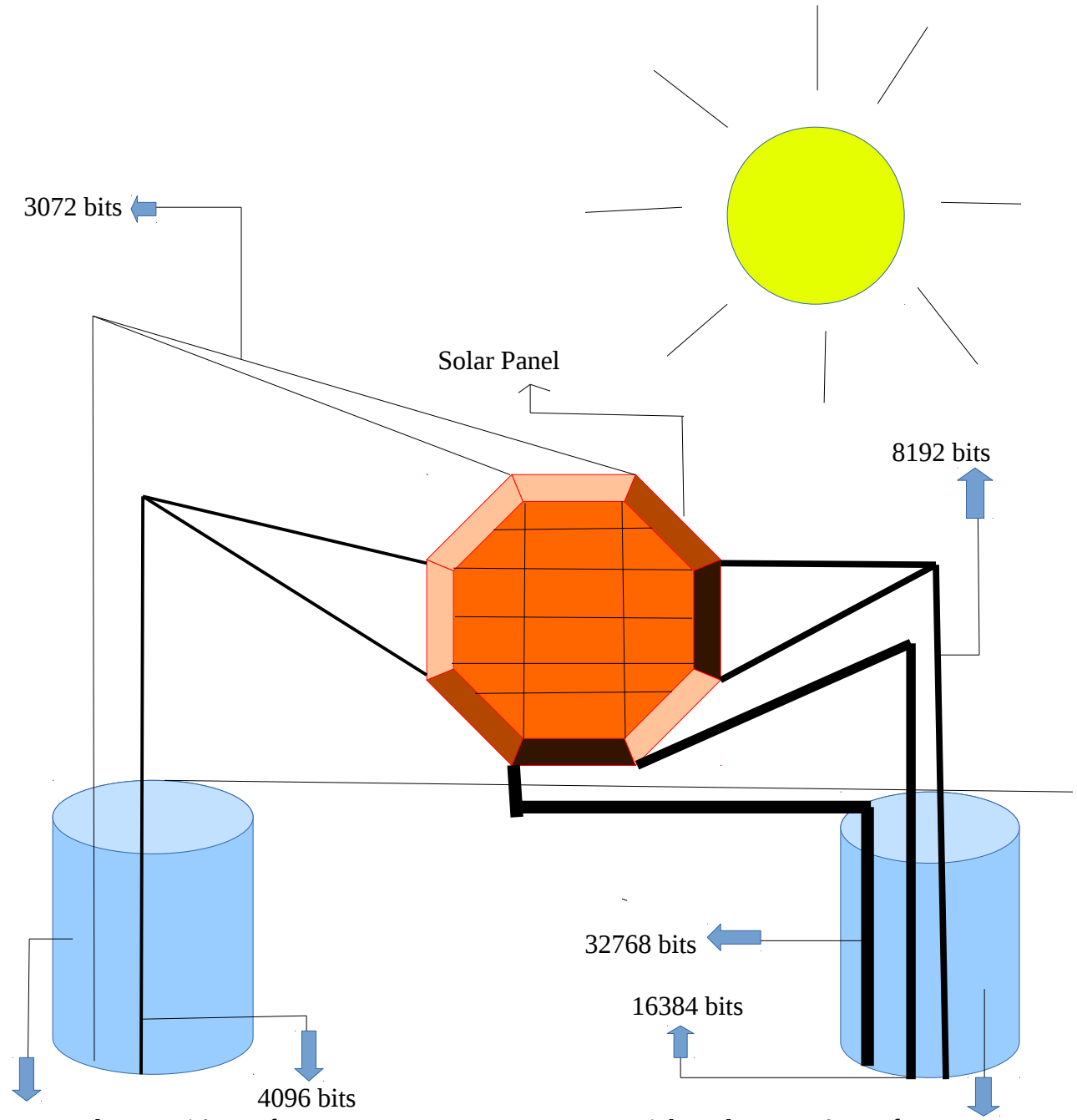
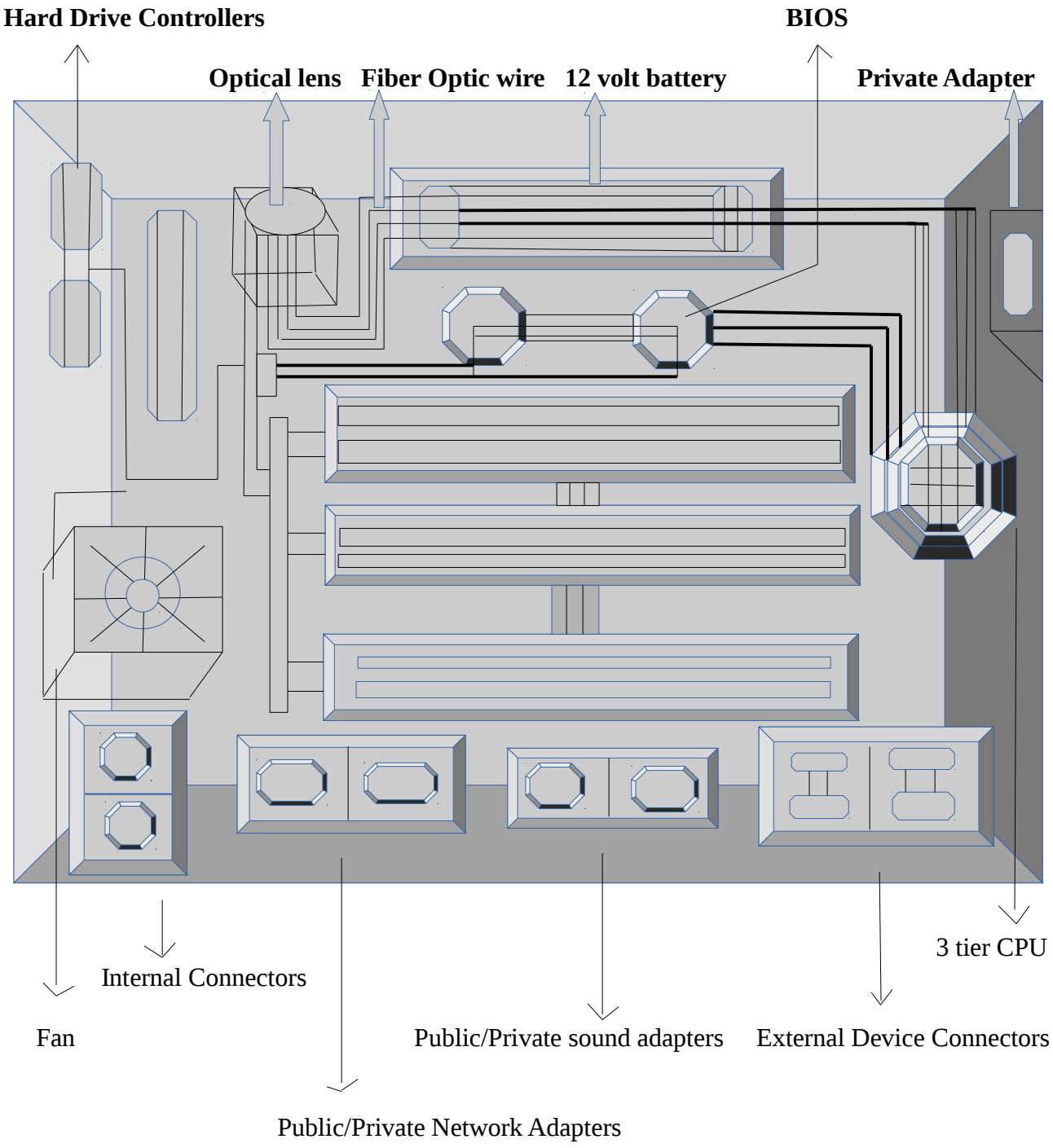


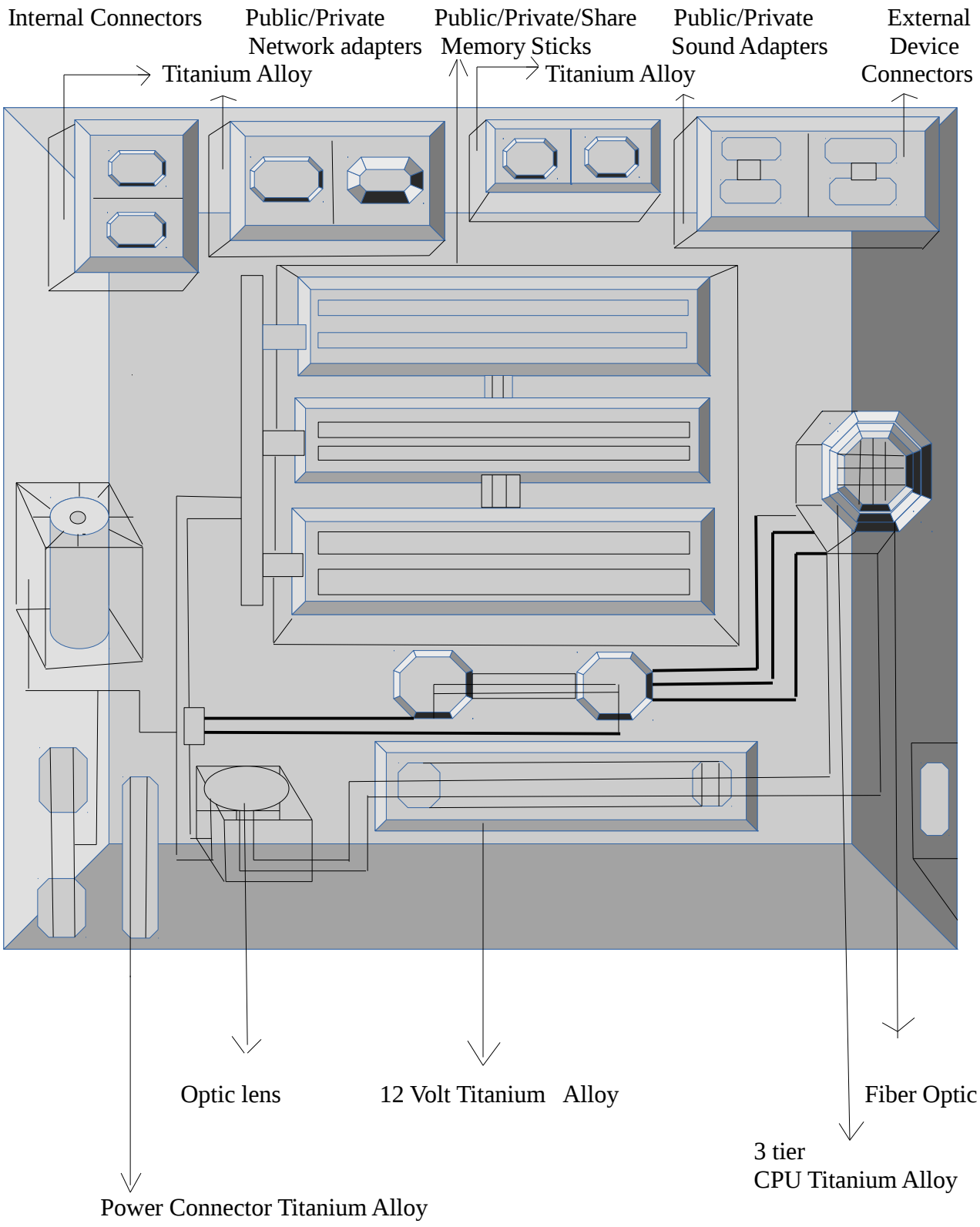
Table of Light to Mechanical Energy Conversion 2-A

# of wires	Total Bits	Material
2	3072	Copper
2	4096	Copper
1	8192	Thin Fiber Optic
1	16384	Thick Fiber Optic
1	32768	2* Thick Fiber Optic

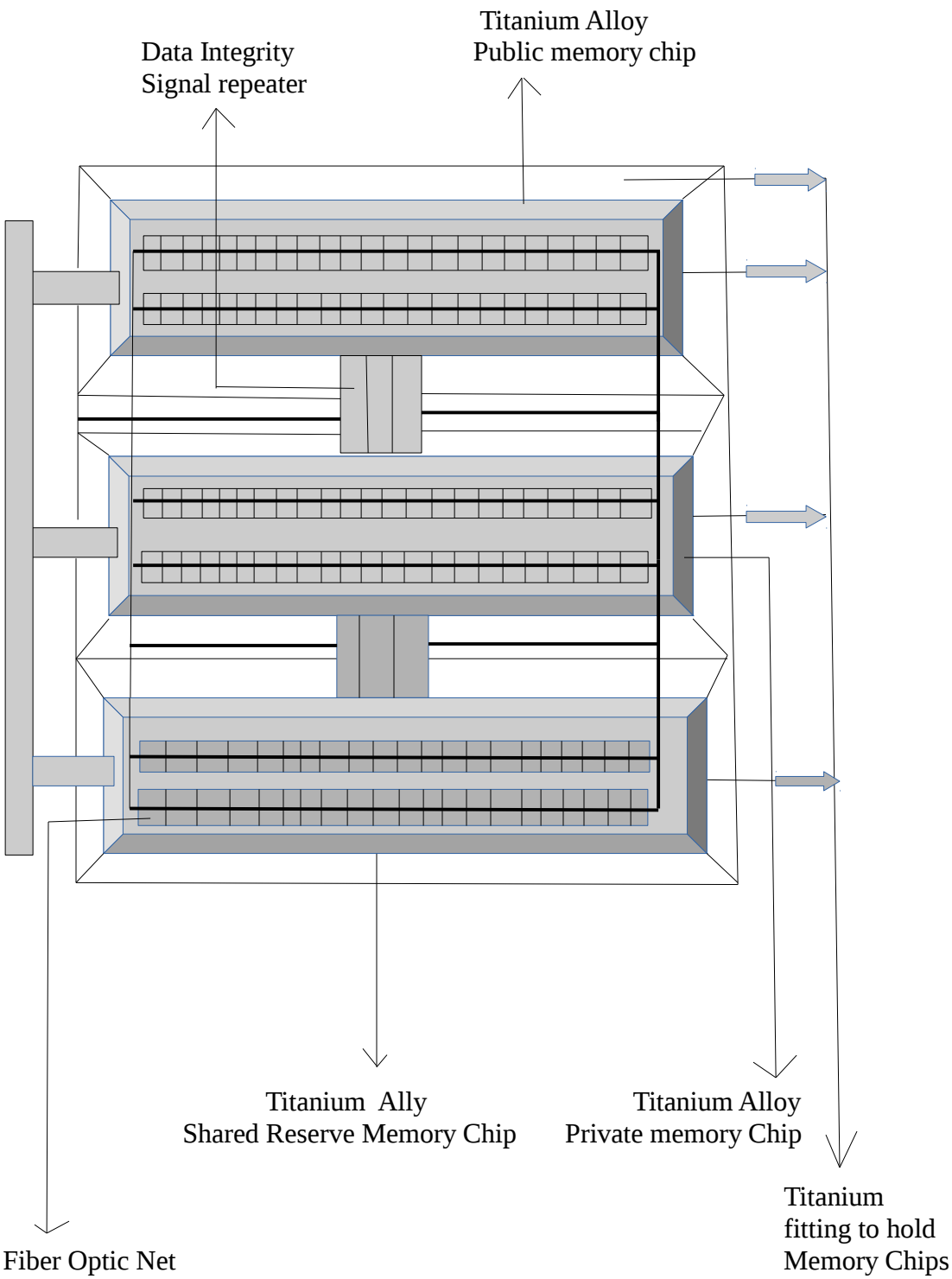
Model Super Sonic 65 Motherboard- Design Rev 1.3 3-A



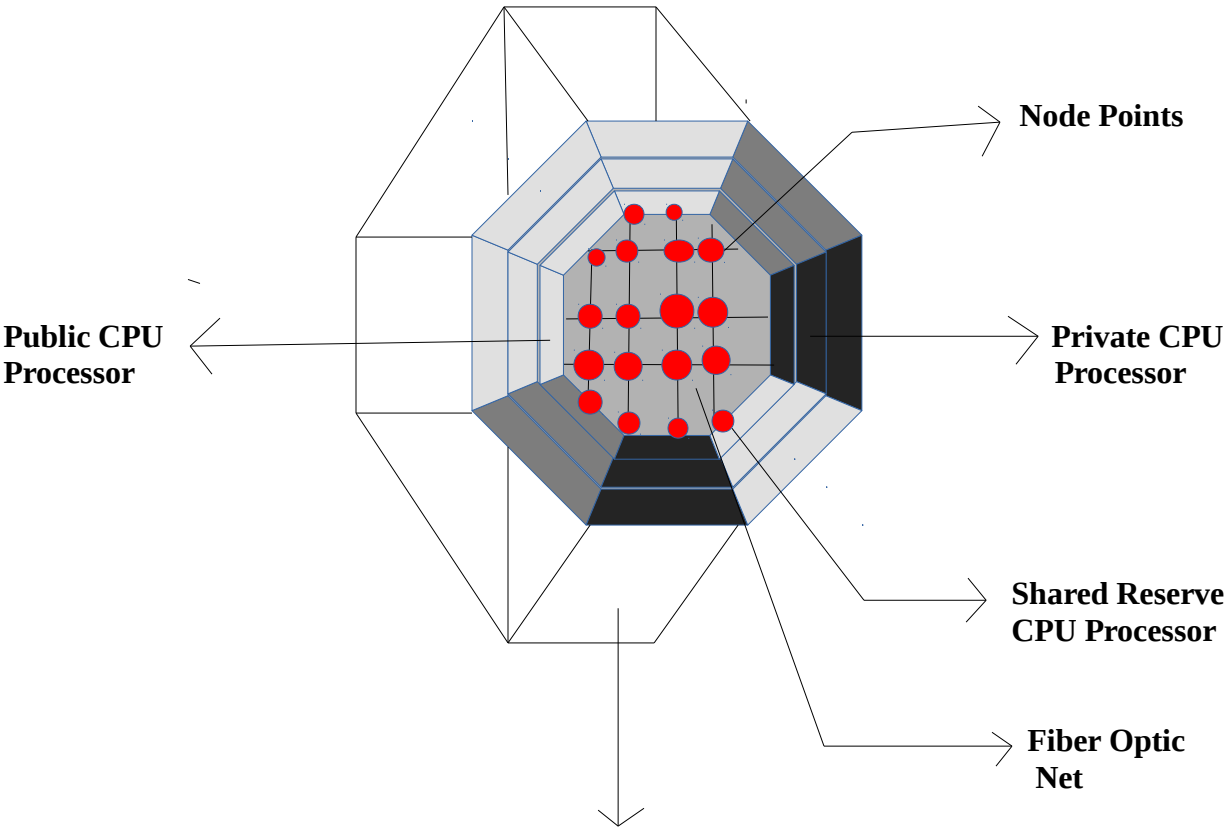
Model Super Sonic 65 Motherboard- Design Rev 1.3 4-A



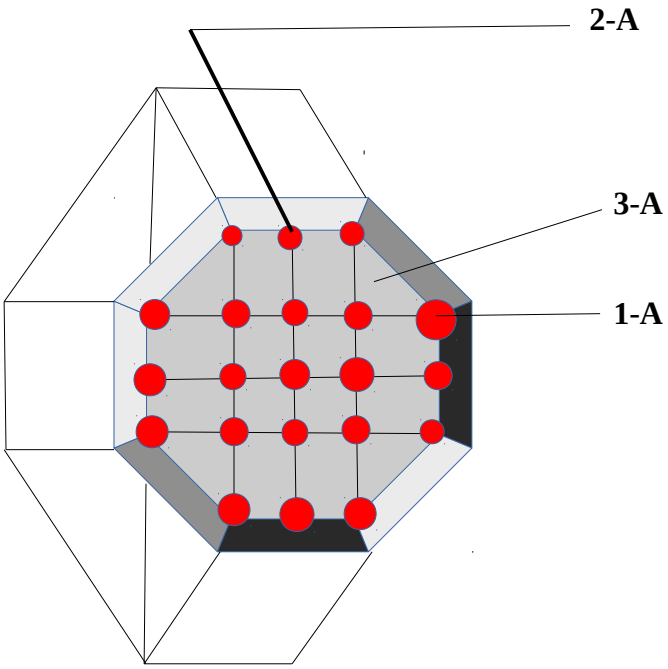
Model Super Sonic 65 Motherboard- Design Rev 1.3 5-A



Model Super Sonic 65 Motherboard- Design Rev 1.3 6-A General View



Model Super Sonic 65 Motherboard- Design Rev 1.3 6-B Front View



1-A Data Points
2-A Data Strings
3-A Fiber Optic Net

Overview of Design

I would like to provide a brief overview of this design. I am taking a solar panel and using the energy to convert this to Mechanical. This is determined by a metric based system or a snapshot of how much energy is being utilized and it than routes this to the appropriate Area of Space which is in relations to the type of material that is being used. Example is I have just polled and took a snapshot of the energy and it is determined to be 4096 bits I than use the table and it is determined to use copper wire to transport the bits see chart 1-A and 2-A.

The second part is the Motherboard itself. As you can see the material being used has a higher heat tolerance than the standard motherboard do to the usage of Titanium on critical components such as the CPU and Memory Chips. Please note there is a limited amount of Optical lens for allowance of more Energy or bits to be processed this allows for better Privacy and Security methods such as Encryption and Authentication methods. This Design allows for Public or Private shared resource such as the CPU and Memory sticks along with 256 bit addressing schemes. The Standard CMOS battery allows for 3 volts at 4096 bits this would require 24 volts instead of the 12 volts. The next chapter shows bit compression so this allows for more bit processing coupled with better grade metal for higher heat tolerances.

The Motherboard as shown previously allows for shared resources on the CPU and Memory sticks along with address entanglement which is shown in Chapter three. The CPU uses a 3 tier processor based on IP Configuration example IP address 126.x.x.x would go to the Public CPU area of processing while IP address 192.168.x.x would go to the Private area of Processing. The CPU employs a 3 tier Fiber Optic nets that has Node points with Data Stings for the best path to be taken .The memory sticks on the CPU now has three slots and also employs a Private, Public, and Shared Space this also uses Fiber Optic Nets inside the memory stick.

I will also employ **Barrys Mechanical Space Equation** in the next chapter.

Chapter 2

Mathematical Equations

a). Barry Equality Field Equation

b). Barrys Mechanical Space

Mathematical Equations

a). Barry Equality Field Equation

I will first begin by showing the Barry Equality Field Equation it is written below

$$\& = (m2 - m1) * (c2 - c1) / q1 \; q2 \; q3 \; q4$$

As stated in previous works q1 – q4 represent areas of space. In this scenario, I will take the low end energy of space with 4096/3072 with the speed of 100mbs and will represent area number 1. Please note the material copper

The 2nd area of space will be 16384/8192 with the speed of 1000 mbs this is shown as area number 2. Please note the material Fiber Optic

$$\& = (4096\ 2^{nd}\ \text{power} - 3072) * (\ 100\ 2^{nd}\ \text{power} - 100) / 1$$

& = (16777216 -3072) * (10000 – 100) /1

& = (16774144) * (9900)/1

& = 166064025600

The second area of space will now be measured and shown.

& = (16384 2nd power – 8192) * (1000 2nd power – 1000)/ 2

& = (268435456 – 8192) * (1000000 – 1000)/ 2

& = (268427264) * (999000) /2

& = (268158836736000) /2

& = 134079418368000

I will now add the 2 areas of space to arrive at my energy totals.

134079418368000 + 166064025600

$\overline{\&}$ = $1.342454824 \times 10^{14}$

The Barry Equality Field equation just took and shown that Energy can be Dynamic and Asymmetrical by showing the different levels of metals and materials along with the differences of Areas of space-energy. I will now show the next equation called **Barrys Mechanical Space below.**

Mathematical Equations

b). Barrys Mechanical Space below

The Equation is written as follows below

The Equation is written as follows

Barrys Mechanical Space = { $\sqrt{2}$ Space } * { $\sqrt{3}$ Object}

Area space 1	=	Mechanical Energy	= 100 mbs
object	=	Bits	= 4096
Area Space 2	=	Mechanical Energy	= 1000 mbs
object	=	Bits	= 8192

I will be compressing the Area space and object's but at different levels of energy the space represents in this dimension and the object itself represents bits. In theory, If I had an object within the bits such as a picture-example I can process the same picture within the Area spaces but at different speeds and bits; thereby, demonstrating some fundamental Quantum Mechanics because I provided alternate paths or tunnels for the picture or object itself but at different levels of Energy.

$\sqrt{100}$ * $\sqrt{4096}$ Area Space 1

Compression		**Dimension**	**object**
1st	compression	10	64
2nd	compression	3.16227766	8
3rd	compression		2.828427125

Barrys Mechanical Space for Area 1 = 3.16227766 * 2.828427125

Barrys Mechanical Space for Area 1 = 8.94427191

$\sqrt{1000}$ * $\sqrt{8192}$ Area Space 2

Compression	**Energy**	**object**
1st Compression	31.622776602	90.509667992
2nd Compression	5.623413252	9.51365692
3rd Compression		3.084421651

Barrys Mechanical Space for Area 2 = 5.623413252 * 3.084421651

Barrys Mechanical Space for Area 2 = 17.344977587

Barrys Mechanical Space = Area space 1 + Area Space 2

Barrys Mechanical Energy = 26.289249497

As you can see, The energy was not constant in different areas of spaces if you compare the Mechanical Energy and areas of spaces in fact supports Energy is Dynamic and Asymmetrical or unevenly distributed. The idea of Energy being constant cannot be supported and further supports the Barry Equality Field Equation. The equation Barrys Mechanical Energy shows how I can compress bits to allow for 32768 bits on the board and maybe even more but the specs can push 32768 bits nevertheless. The areas of space compress the speed to allow for alternate dimensions which equates to in the programming world frame processing first frame example 1st dimension Area space 1 2nd frame 2nd dimension Area space 2 but the 2nd dimension is processed more quickly than the 1st demonstrating a concept of dimensional processing the event has already taken place before the 1st event or 1st dimension.

In the next chapter, I will go over the logic gateways to interface with the Motherboard itself.

Chapter 3

Logic Gateways and Memory Addressing

In this chapter, I will process logic gateways specifically polling the Area of Spaces and taking snapshots and than interfacing with the motherboard itself. I will also provide a simple logic testing program for the memory addressing that shows address entanglement on Public and Private areas of space.

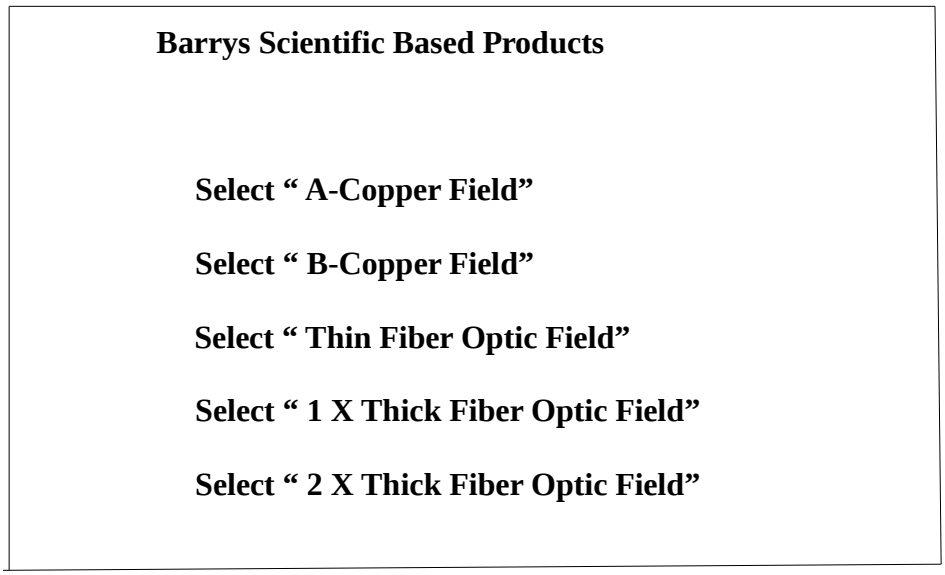
Load Encryption-table-module-table

Variable	**Encryption strength # bits**
aCopperField	3072
bCopperfield	4096
c1xThinFiberopticfield	8192
d1xThickfiberopticfield	16384
e 2xThickfiberopticfield	32768

The next step is to load the menu and Logic Gateways.

{

Load Read-Only-memory-Table



Rem This is a system level program that is not visible and is polled
Rem before running test conditions

Gateway-processing

Gateway-1 =f
Gateway-2 =g
poll f
poll g

0 = “off”
1 = “on”
if f = “on”
goto Area-space-1
else
if g = “on”
goto Area-Space-2
else
if f and g = “off
poll f and G
exit

rem Area-spaces checking conditions on or off

Area-space-1
0 = “off”
1 = “on”
h = aCopperField
i = bCopperfield
rem set switches to on or off and check conditions
if h = “on”
set 3072-bits
move “3072” h
else
if i = “on”
set 4096-bits
move “4096” i
else
if h and i = “off”

goto **Gateway-processing**

Area-Space-2

0 = “off”
1 = “on”
x = c1xThinFiberopticfield
y = d1xThickfiberopticfield
z = e 2xThickfiberopticfiel

rem set switches to on or off and check conditions
if x = “on”
set 8192-bits
move “8192” x

else

if y = “on”
set 16384-bits
move “16384” i
else
if z = “on”
set 32768-bits
move “32768” x
else
if x, y, z = “off”

goto Gateway-processing
poll Read-Only-memory-table
rem proceed to Super sonic 65 motherboard
set 0
clear tables
exit }

This was a simple logic gate program to demonstrate the following

1). Solar Energy is used for the Solar Panel
2). Solar Panel than is converted to Mechanical Energy -Bits
3). The solar panel is than polled and a snapshot is taken
4). The Gateway processing logic control is than initiated as outlined above- **Gateway-processing**
5). conditions are tested and than goes directly to the SS 65 Motherboard itself.

I will now set variables on how to process bits for memory addresses. This is achieved by first going back to the basics 1 byte = 8 bits 2 bytes = 16 bits = 1 character. I can extend this by simply padding the beginning of the byte with 72 bits + Actual address bit 128 + 56 bits end or trailer thus I have a 256 bit addressing scheme. I have formed an Asymmetrical bit addressing scheme with Dynamic Energy within the constant of 256 bits shelled energy that is dynamic.

128 Bits = 1 Byte	Sub-Physical layer
256 Bits = 1 Character	Physical Layer
32768 / 256 = 128 Character Frame	Mac Address Layer

IP Packet = Public Address = Frame + Header + Trailer
IP Packet = Private Address = Frame + Header + Trailer
IP Packet = Shared Reserve = Private or Public with address Entanglement.

Barrys Scientific Based Products

1). **Public Address Processing**

2). **Private Address Processing**

3). **Shared Memory Address Processing**

L**oad Memory Address Module**

Variable

2aPublic-memory
2bPrivate memory
3cReservesharedmemory

Memory Address Processing

2aPublic-memory =J
2bPrivate-memory =k
3cReservesharedmemory = l

0 = “off”
1 = “on”
rem testing address switches
if j = “on”
set “256” m
rem field m is set to 256 bits
move “256” m

else
if k = “on”
set “256” n
rem field n is set to 256 bits
move “256” n
else
if l= “on”
set “256” o
rem field o is set to 256 bits
Rem user selects condition for address entanglement
Print “ Copy o to J Yes/No”
if “yes”
set “256” j
move “256” j
else if
“no”
set “256” k
move “256” k
else
exit

To access the memory addresses, The user has to choose what type of Memory to process Public or Private. If you wish to mix the two, You select the Address Entanglement feature Public to Private or Private to Public this also places the choices on the user. The module tests address switches on and off along with shared states on and off.

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

Chapter 4

Final Thoughts

Final Thoughts

I have updated the SS 65 Motherboard Design known as Super Sonic with a revised Version 3. The updates include Memory Sticks with slot expansion, 3 level CPU, Public, Private, Shared Areas of space including Node Points and Data Strings that uses Fiber Optic Nets and Titanium. I have also updated the Logic Programming with New Screens and Memory addressing schemes that are Asymmetrical in nature Constant equals 256 bits with 72 bits padding 128 bits address 56 bits trailer shelled within the constant.

I have updated the Visual Design, and Logic control processing {Gateways and memory addressing schemes}. This work was a little different this time because it took bits and pieces of different ideas and concepts I have previously written and created a cohesive work that enforces the ideas I have attempted to convey in the past along in addition to some new concepts and or ideas.

I wanted to create a work that took past ideas and create a work that presents some new ideas worthy of consideration. I wish to thank you for taking the time in reading this work. In this work, If you notice, I did not write the CPU logic program because this would have been overwhelming all at one so I attempted to break this down into chunks because of the many ideas and concepts being presented. I may create a Revised version in subsequent works.

Barry L. Crouse

10/08/2017

Email crouseb395@gmail.com

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

Thank you for reading this work.