Barrys SS-65 Ver 1.4 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 along with a new file and hashing scheme to promote 6144 block data schemes used for IP frames and packet exchanges . I will be using in this work a direct Interface converting Solar Energy to Mechanical Bits and going directly to the Motherboard itself. I am still using Solar Energy to a Interface than to the Motherboard. This is the final version of this model.

I have updated this Motherboard Design to include the following:

1). Video Card one slot with three areas of space Private, Public, Shared resources including Data Strings and Node Points.

2). Hashing and Algorithm scheme for second layer OSI Data link layer processing with 6144 Character block data with 8 rows by 768 Characters matrix.

3). 6144 Character Data block tied into the {256 bit address scheme from last work}.

4). 24 layer frame with address entanglement and security

5). Frames at the OSI layer have 4 IP packet burst of 1536 notice 36 above the standard 1500.

Table of Contents

Chapter 1

Visual Design

Solar to Mechanical Energy 1-A

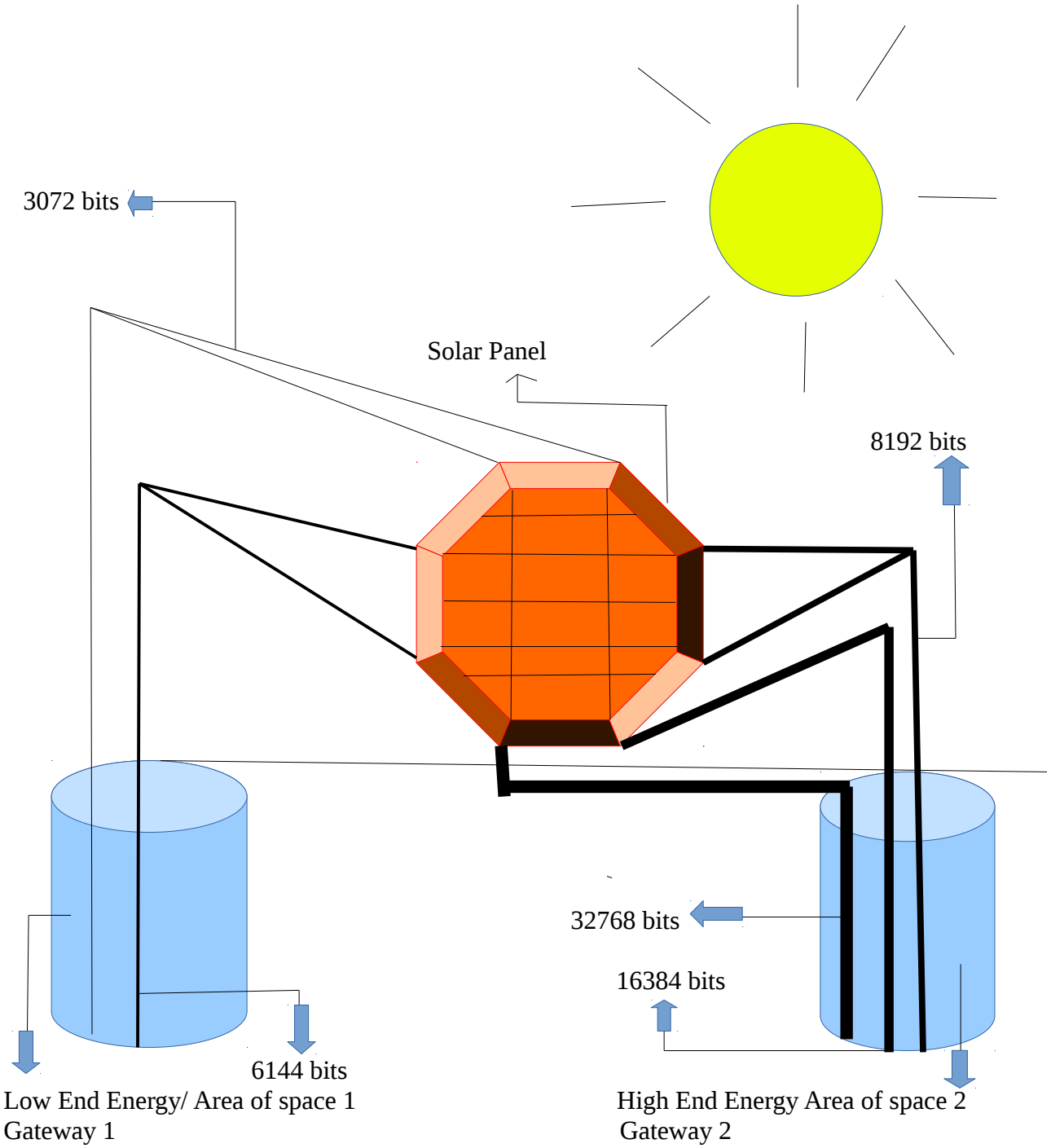
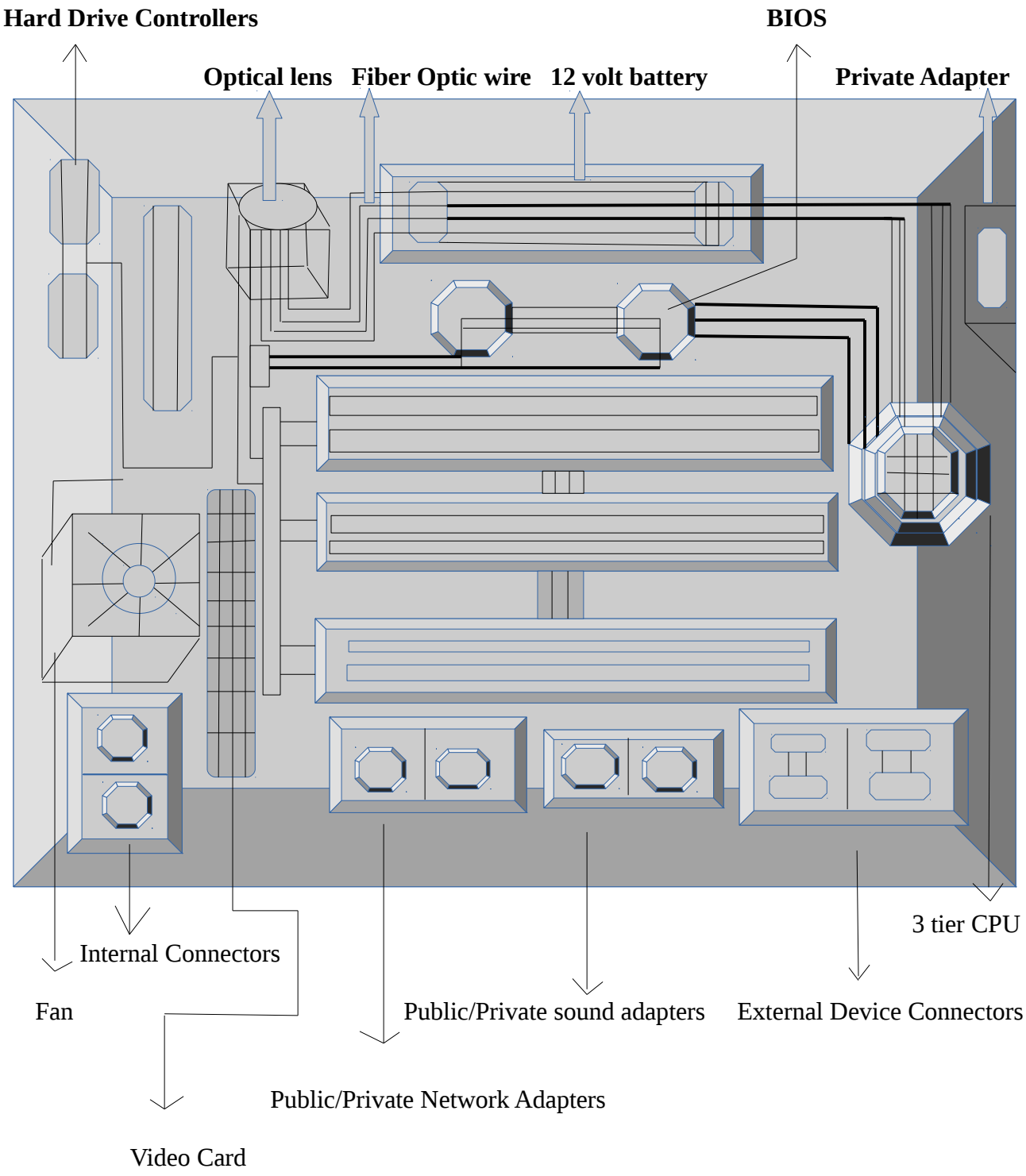


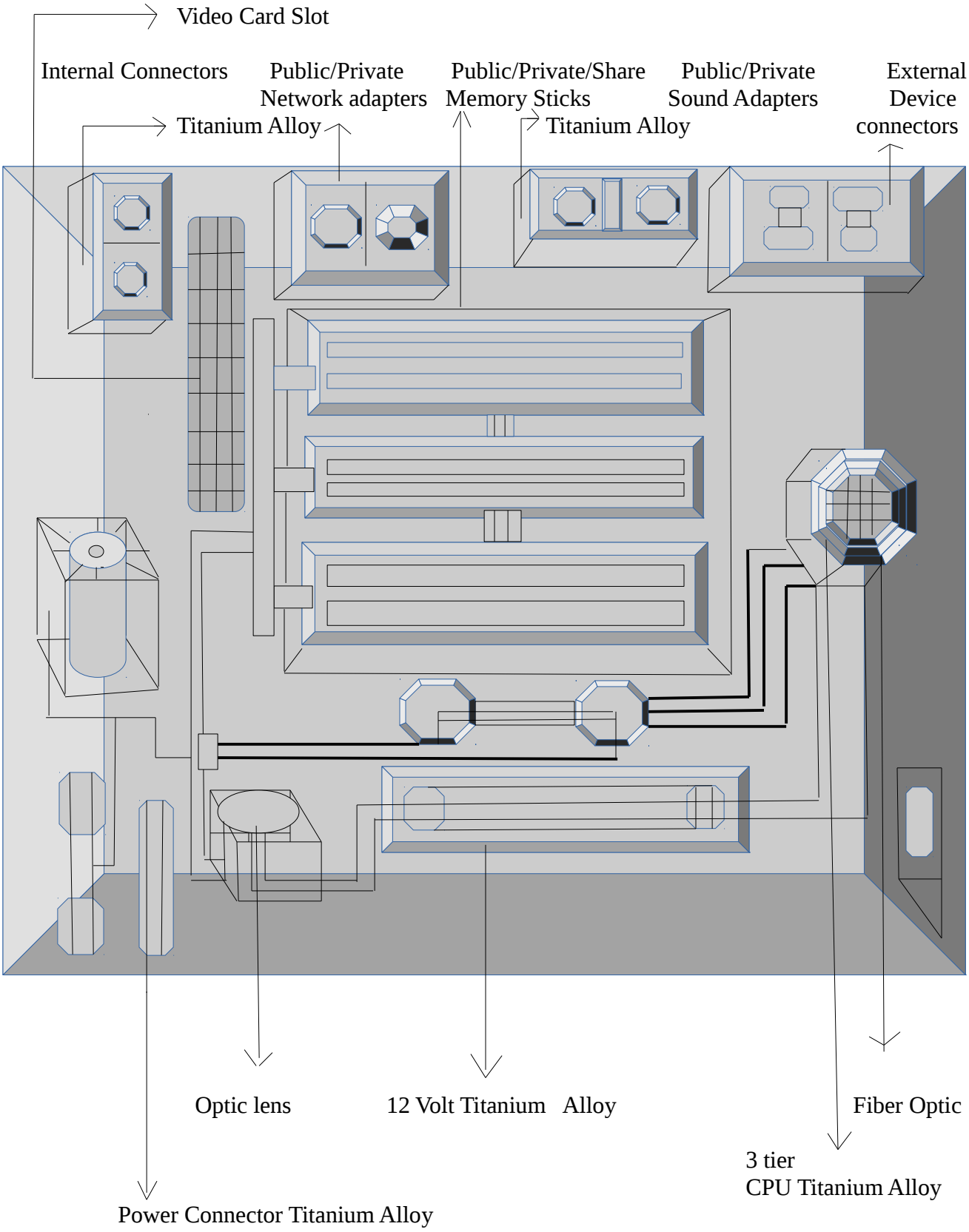
Table of Light to Mechanical Energy Conversion 2-A

# of wires	Total Bits	Material
2	3072	Copper
2	6144	Silver/Copper plated
1	8192	Thin Fiber Optic
1	16384	Thick Fiber Optic
1	32768	2* Thick Fiber Optic

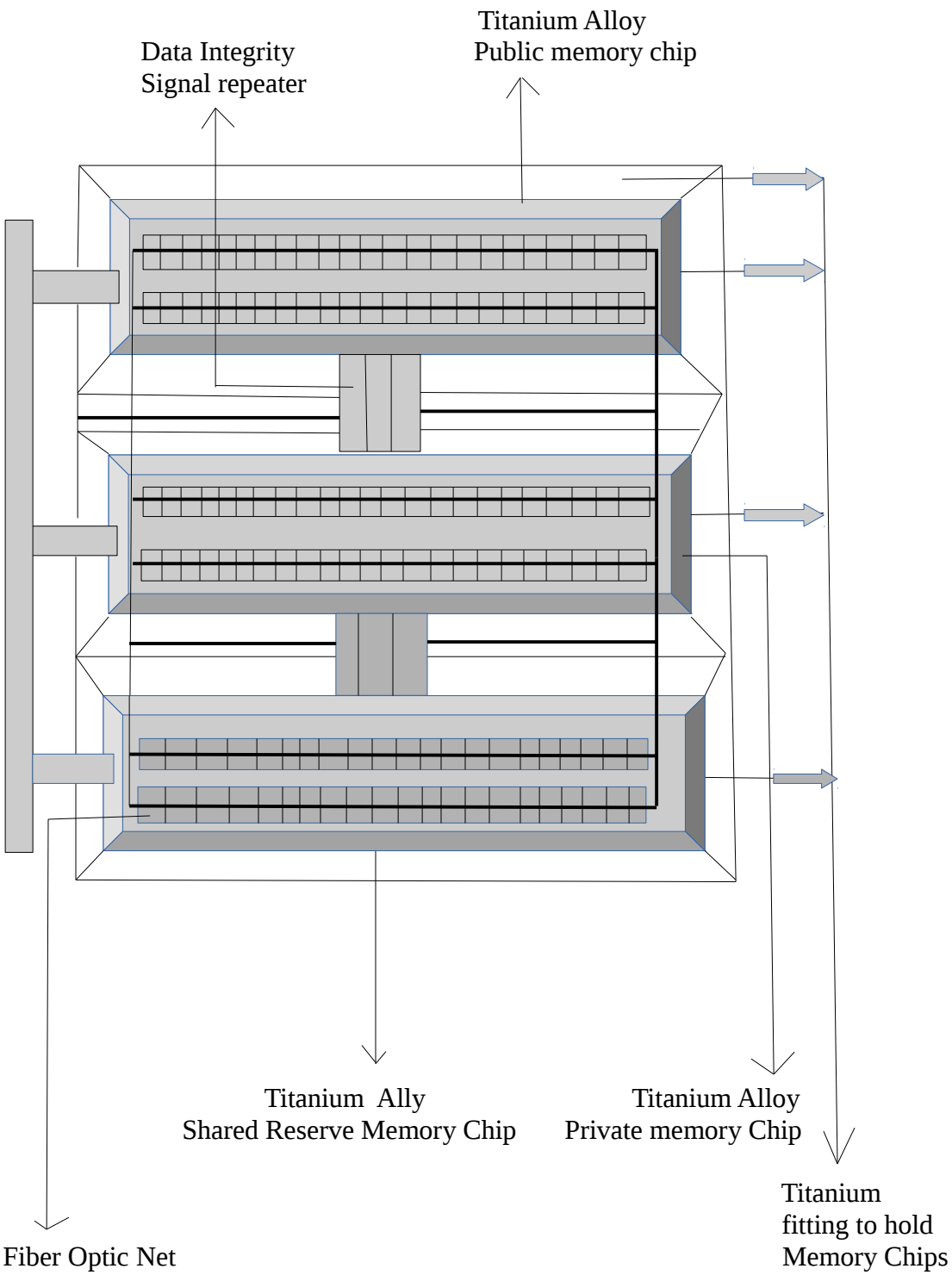
Model Super Sonic 65 Motherboard- Design Rev 1.4 3-A



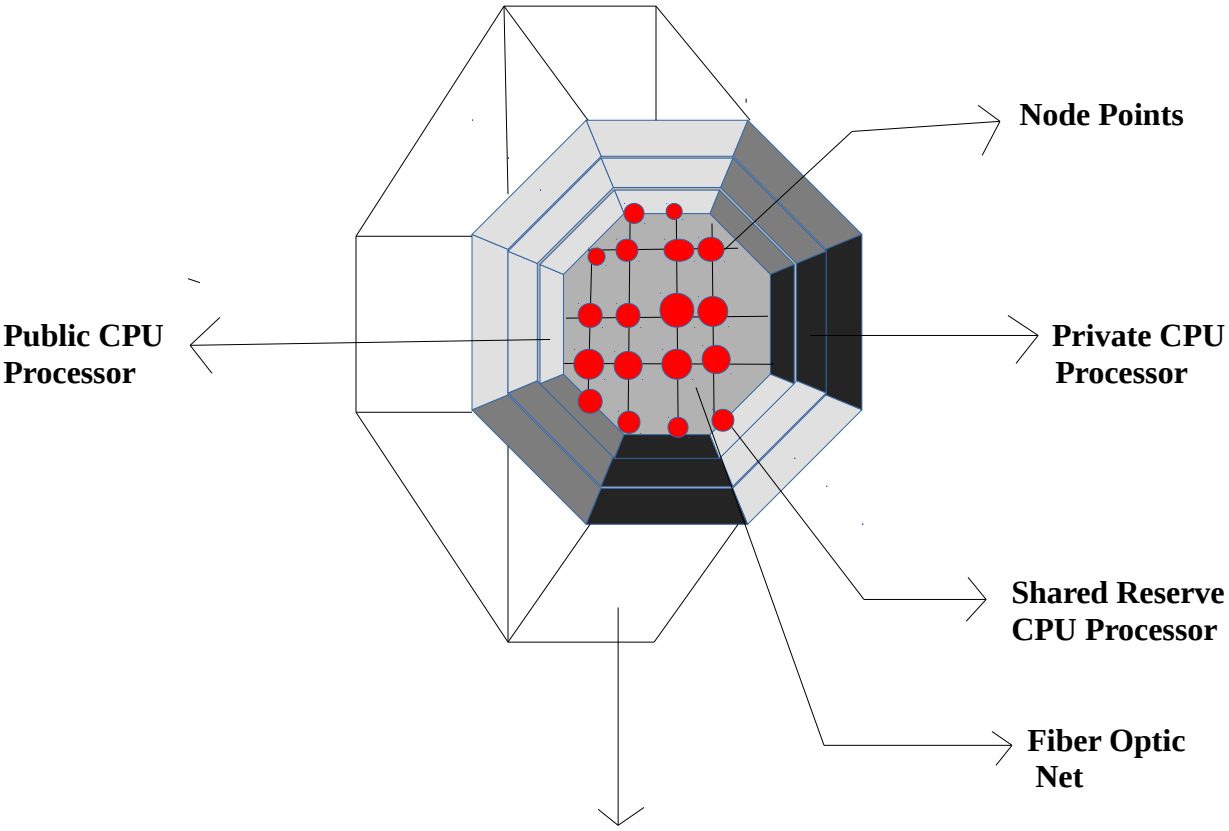
Model Super Sonic 65 Motherboard- Design Rev 1.4 4-A



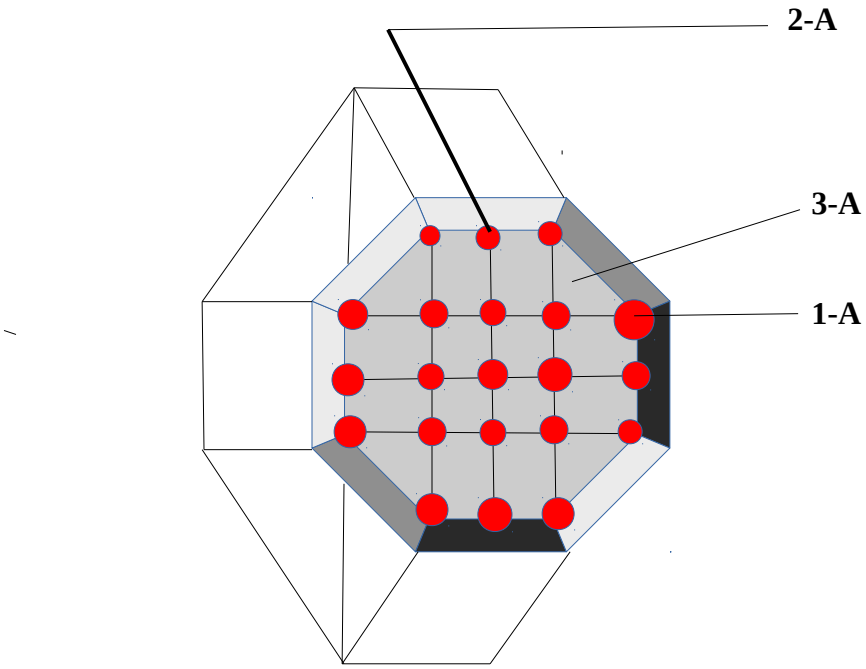
Model Super Sonic 65 Motherboard- Design Rev 1.4 5-A



Model Super Sonic 65 Motherboard- Design Rev 1.4 6-A General View

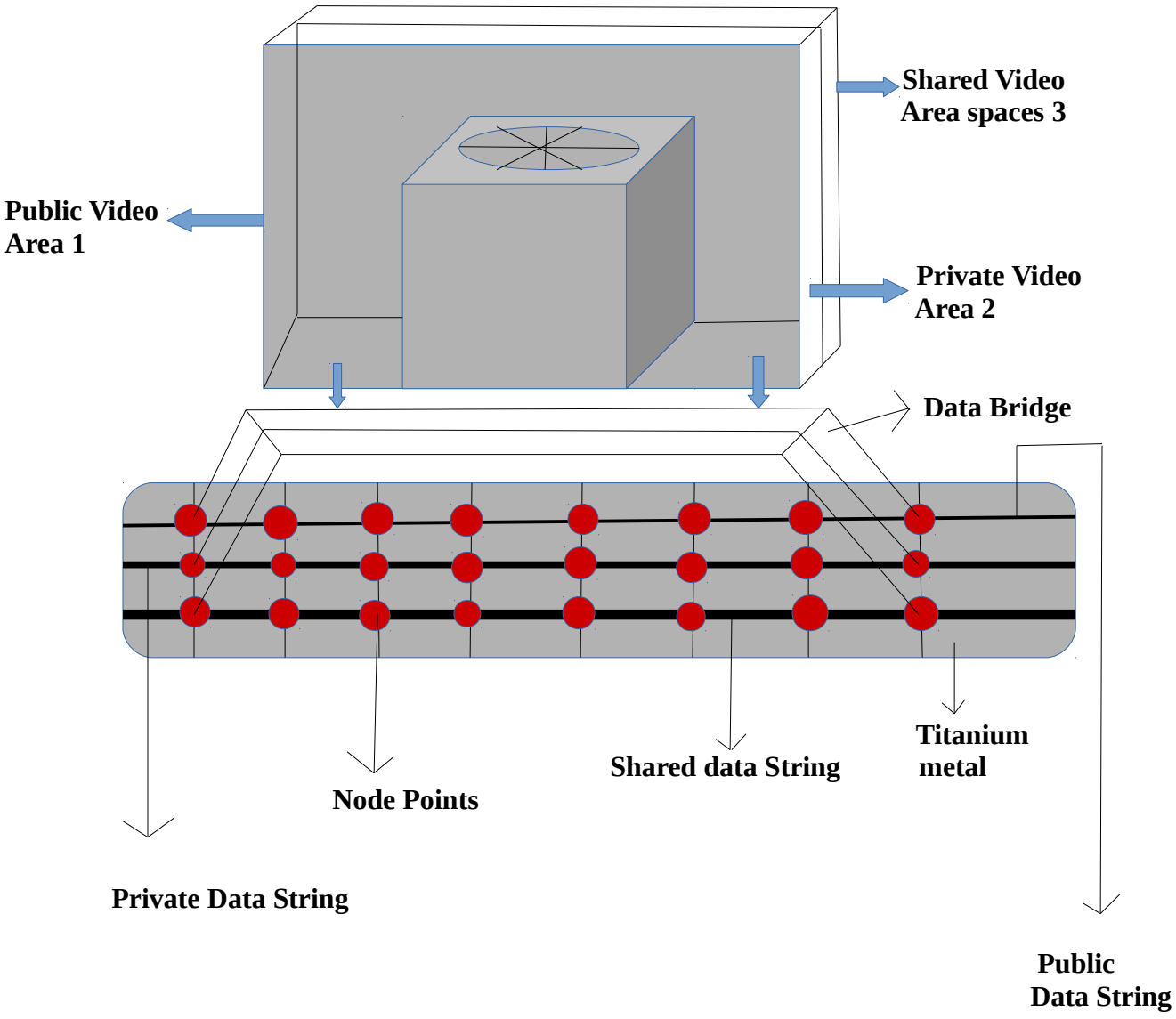


Model Super Sonic 65 Motherboard- Design Rev 1.4 6-B Front View

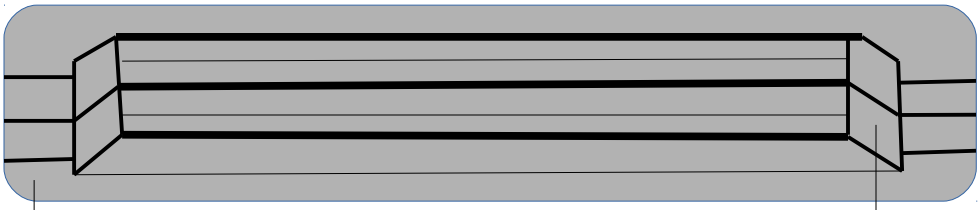


1-A Data Points
2-A Data Strings
3-A Fiber Optic Net

Model Super Sonic 65 Motherboard- Design Rev 1.4 7-A General View



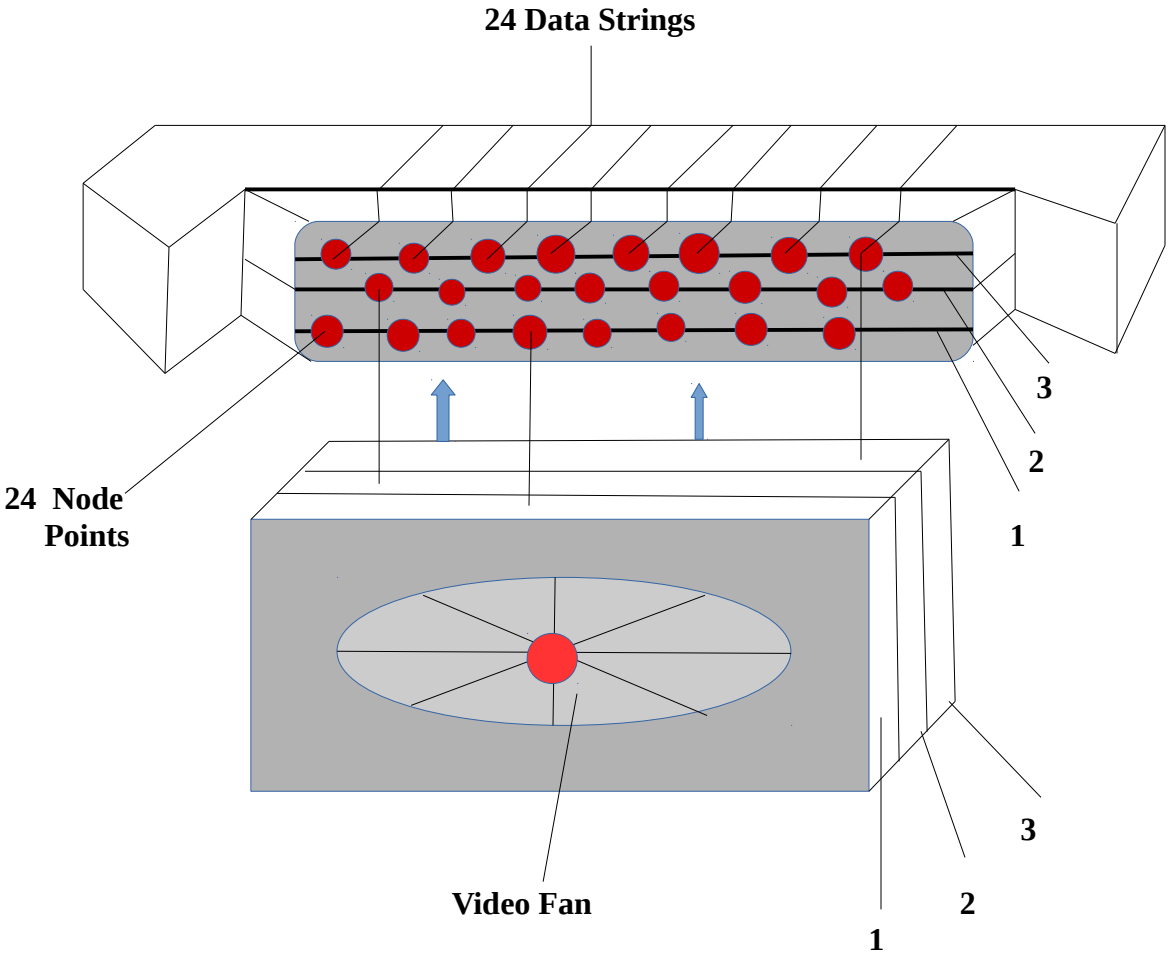
Model Super Sonic 65 Motherboard- Design Rev 1.4 8-A General View



Titanium Metal

Video Card Slot

Model Super Sonic 65 Motherboard- Design Rev 1.4 9-A General View



Area of Space #	Defined
1	Public
2	Private
3	Shared

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 6144 bits on the low end. I than use the table and it is determined to use copper with silver 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 because of the usage of Titanium on critical components such as the CPU ,Memory Chips and video slots . This allows for more heat tolerance equating to more bit processing. 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 along with shared resource such as the CPU ,Memory sticks and video slots with three areas of space coupled 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 a 6144 characters with data link layer processing bytes into frames with packet bursts of 1536.

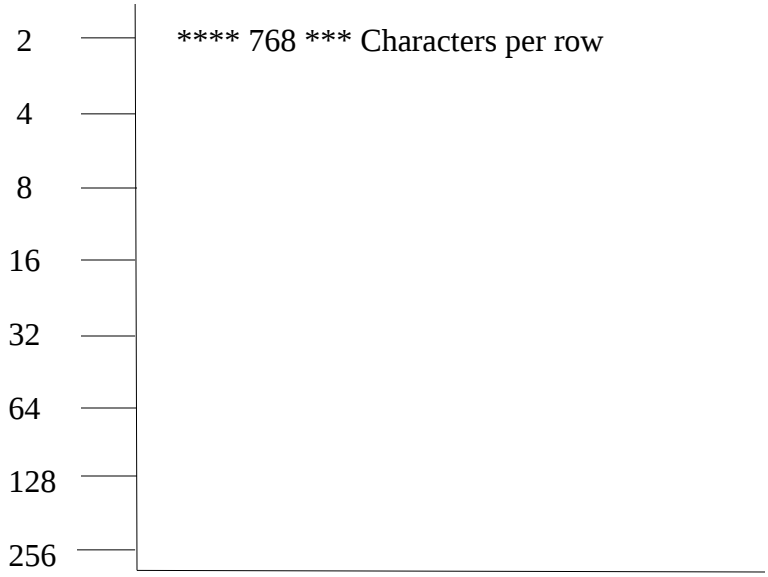
I have created a video slot that uses 24 node and data points but each node is not equal size and spacing wise it does not follow a matrix with neat rows and columns visually see visual displays it is not evenly aligned and spaced. The video slot has 3 areas of space Public, Private, and shared on one card. The video card uses a dependency of CPU resources example if I choose Private CPU processing than I use Private Video processing. This is defined in the Logic gateway processing.

I will be presenting my Single 6144 Data Block Processing in the next chapter..

Chapter 2

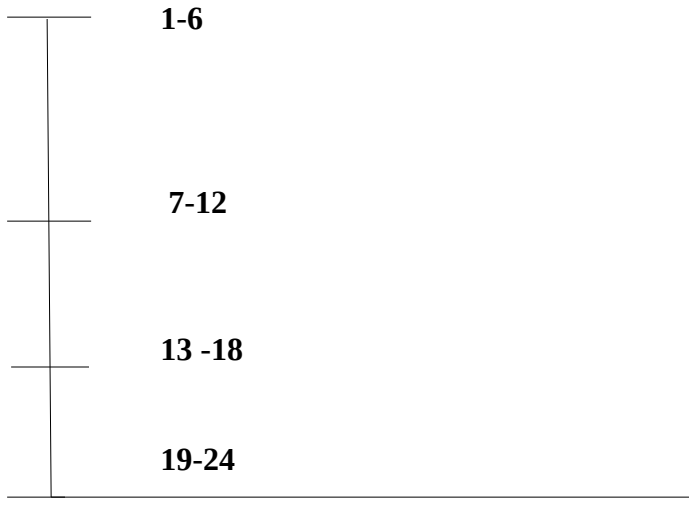
Single 6144 Data Block processing

I will begin by defining the block data of 6144 characters with the 256 bit addressing scheme from here I will create a matrix of 8 rows each with 768 characters matrix. See chart below.



Since I have defined my address scheme as 256 bits see last work written in version 1.3, I divide 6144 characters by 256 bits and it comes up with 24 frames. I divide 24 frames by 6 to get the 4 packet of 1536 frames in bursts. To secure the data when sending outbound to the Internet or Intranet, I use a frame entanglement swapping frame 1 and 2. This is reassembled at the final destination or hop for old timers. I create four packets of 1536 frames equal to 6144 Characters. To create a even more secure environment, The user chooses which packets order is to be sent see chart below. I could even swap the last frame's of 23 and 24. The process is simple to follow take a 6144 block of data and create a matrix of 8 rows by 768 characters this equals 6144 characters. The next step is to take 6144 Characters divide by the address length 256 equals 24 frames swap frames one and two along with 23 and 24. create the IP Packet burst of 1536 with boundary's set for frame sequence processing and assembled at final destination. Client chooses which order is to be followed and assembles at final destination.

Packet #	Frame sequence
1	**1-6**
2	**7-12**
3	**13 -18**
4	**19-24**



4 packets of 1536 burst with frame sequences of 6.

Chapter 3

Group Frame to Packet Processing

Please remember the following OSI layer Stack Protocol Frame and Packet Processing

Physical	Bits to Bytes
Data Link layer	Bytes to Frames
a). MAC address Layer/Data Link Layer	Bytes to Frames
IP layer	Frames to Packets

The conversion and process is the following:

6144 Bits * 256 Bits = 1572864 bits

The next step is to convert the bits into bytes this is dependent on how you represent characters in fields for example in the 80's you could use 2 bytes or 16 bits to represent 1 character in my presentation is 256 bytes since I created a 6144 Character block of Data.

1572864 bits divided by 256 bytes = 6144 characters / 6 frames used for sequencing. This equals 1536 and than I select 4 because 1536 * 4 = 6144. The frames can be broken up because on the lower level second layer you create a sub-level within the second layer defining the mac address and how it is going to be used for frames into packets. This is a basic review. If you do not understand this, You may wish to study the OSI 7 layer stack and TCP/IP protocols.

Chapter 4

Logic Gateways and Area of Space Processing

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.

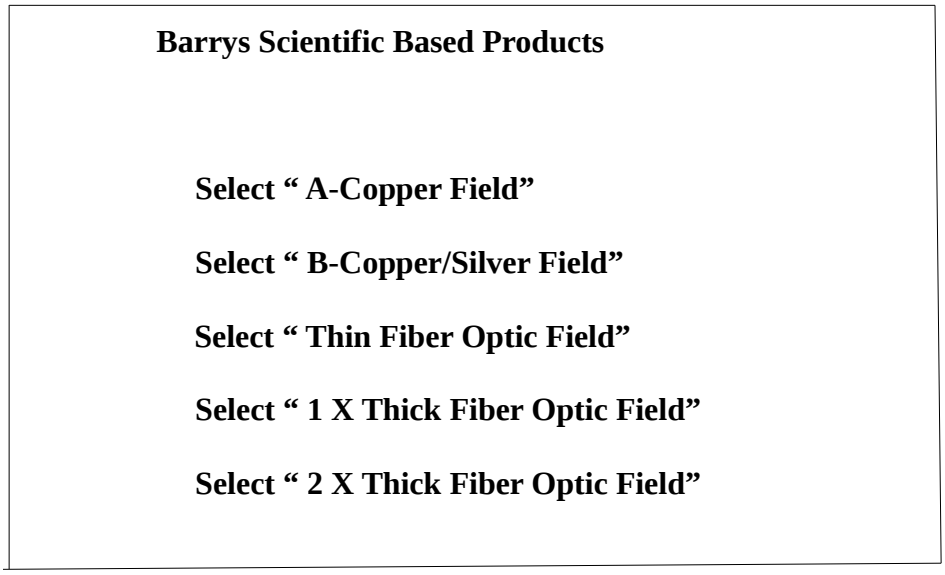
Load Encryption-table-module-table

Variable	**Encryption strength # bits**
aCopperField	3072
bCopper/silverfield	6144
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 = bCopper/silverfield
rem set switches to on or off and check conditions
if h = “on”
set 3072-bits
move “3072” h
else
if i = “on”
set 6144-bits
move “6144” 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.

Barrys Scientific Based Products

1). Public Processing

2). Private Processing

3). Shared Processing

L**oad Area of Space processing**

Variable

2aPublic-processing
2bPrivate-processing
3cReserve-shared-processing

Area of Space Processing

2aPublic-processing =J
2bPrivate-processing =k
3cReserve-shared-processing = l

```
0 = “off”
1 = “on”
 m=  Array-1 {1,2,3}
 Rem Arrays are defined as CPU, Memory, Video slot
 rem testing areas of space
if j = “on”
set “256” m
rem field m is set to 256 bits
move “256” Array-1

 else
 n = Array-2 {4,5,6}
 if k = “on”
 set “256” n
rem field n is set to 256 bits
move “256” Array-2
  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”  Array-2
 else if
 “no”
 set “256” k
 move “256”  Array-1
 else
 exit
```

To access the Menu the user chooses what Area of Space is selected, The user has to choose what they want to select. 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. Notice the usages of Arrays in the logic programming this is a little bit different than before previous versions.

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

Chapter 5

Final Thoughts

Final Thoughts

I have updated and have fully developed the SS 65 Motherboard Design known as Super Sonic with a revised Version 4. The updates include Video Slots with Area of Space processing and 6144 Single and Group Character processing. I have created One video card slot with 3 Areas of space, 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, Areas of space and usage of arrays.

I have updated the Visual Design, and Logic control processing {Gateways and now Areas of space processing with arrays}. This work is the final revised version of the SS 65 Motherboard Design all critical components have been fully updated and developed.

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 that uses data node Points and strings. The node points uses uneven symmetry as far as size and spacing are concerned.

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.

Barry L. Crouse

01/03/2018

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.