Early Video In Pre-OS Environment For Intel Servers

Prepared By :

Hiren Patel

11MCEC12

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD

Early Video In Pre-OS Environment For Intel Servers

Major Project

Submitted in fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

Prepared By :

Hiren Patel

11MCEC12

Internal Guide

Prof. Swati Jain

Nirma University

External Guide

Mr. Banuprakash Krishnappa

Intel Technology India LTD.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD

DECLARATION

This is to certify that,

I, **Hiren Patel**, **11MCEC12**, a student of semester IV Master of Technology in Computer Science Engineering, Nirma University, Ahmedabad , hereby declare that the project work **Early Video In Pre-OS Environment For Intel Servers** has been carried out by me under the guidance of Mr. Banuprakash Krishnappa, Intel Technology India Private Limited, Bangalore and Prof. Swati Jain, Department of Computer Science and Engineering, Nirma University, Ahmedabad. This Project has been submitted in the fulfillment of the requirements for the award of degree Master of Technology (M.Tech.) in Computer Science and Engineering, Nirma University, Ahmedabad during the year 2012 - 2013.

I have not submitted this work in full or part to any other University or Institution for the award of any other degree.

Hiren Patel (11MCEC12)

CERTIFICATE

This is to certify that the Major Project entitled **Early Video In Pre-OS Environment For Intel Servers** submitted by **Hiren Patel(11MCEC12)**, towards the fulfillment of the requirements for the degree of Master of Technology in Computer Science Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, have not been submitted to any other university or institution for award of any degree or diploma.

Mr. Banuprakash Krishnappa
External Guide,
Intel Technology India Ltd.

Prof. Swati Jain
Internal Guide,
Nirma University

Dr. Ketan Kotecha
Director,
Nirma University

Prof. Vijay Ukani
PG Coordinator - CSE,
Nirma University

Dr. Sanjay Garg
HOD-CSE,
Nirma University

ACKNOWLEDGEMENT

First and foremost, sincere thanks to **Mr. Banuprakash Krishnappa** Manager, Intel Technology India Private Limited, Bangalore. I enjoyed his vast knowledge and owe him lots of gratitude for having a profound impact on this report.

I would also like to thank my Internel guide **Prof. Swati Jain**, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance.

I would also like to thank my course cordinator **Prof. Vijay Ukani**, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance.

I would also like to thank **Dr. Sanjay Garg**, head of department-CSE, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance.

I would also like to thank **Dr. K Kotecha**, Director, Institute of Technology, Nirma University, Ahmedabad for providing me an opportunity to get an internship at Intel Technology India Private Limited, Bangalore.

I would like to thank my all faculty members for providing encouragement, exchanging knowledge during my post-graduate program.

I also owe my colleagues in the Intel, special thanks for helping me on this path and for making project at Intel more enjoyable.

Hiren Patel. (11MCEC12)

Contents

List of Figures

List of Tables

Abstract

In existing server systems, many tasks are required to be performed before the video subsystem can be initialized.Server systems have more complicated memory subsystem and thus, take longer for the BIOS to initialize the memory subsystem and ultimately to boot up.Thus, the user is deprived of video status of the boot process until the video subsystem has booted. On average, a server may take 30-55 seconds before video may be initialized. A user who is not familiar with the delayed video response time may think that the server has failed to boot and recycle the power button in error.User is not able to see any progress, debug, error message if any error comes during booting up of server.

Enabling the video in very early phase of booting process,in PEI(Pre EFI initialization-one phase of bios boot process) phase so that video comes up in 2-3 seconds.Enabling the Matrox VGA(Video graphics adaptor) card by writing to VGA card register and set text mode 80 x 25 and provide library for print messages.Video graphics adaptor initialization uses the CAR(cache as RAM) instead of system RAM to perform initialization.Display progress message and debug message to video console to diagnostic the system boot up errors.Using SOL(Serial over LAN) send all debug message to remote host and operate servers remotely for headless server system used in data centers.

Chapter 1

Introduction

A server is a physical computer dedicated to running one or more services (as a host), to serve the needs of the users of other computers on the network.Servers runs for long period without interruption and availability must often be very high, making hardware reliability and durability extremely important. Although servers can be built from commodity computer parts, mission-critical enterprise servers are highly fault tolerant and use specialized hardware with low failure rates in order to maximize uptime, for even a short-term failure can cost more than purchasing and installing the system. For example, it may take only a few minutes of down time at a national stock exchange to justify the expense of entirely replacing the system with something more reliable.
Major factors for Server are:

- **Storage Capacity**

  Servers may incorporate larger computer fans to help remove heat,faster and higher-capacity hard drives, and uninterrupted power supplies that ensure the servers continue to function in the event of a power failure. It gives higher performance and reliability at a correspondingly higher price. Hardware redundancy-installing more than one instance of modules such as power supplies and hard disks arranged so that if one fails another is automatically available-is widely used.

- **Amount of memory**

  Servers uses memory with error detection and correction, redundant disks, redundant power supplies and etc. to increase reliability. Such components are hot swappable,it allows to replace them on the running server without shutting it down. To

prevent overheating, servers have more powerful fans.

Servers take a long time for the hardware to start up and load the operating system. Servers do large amount of pre-boot memory testing and verification and startup of remote management services. The hard drive controllers then start up banks of drives sequentially, rather than all at once, so as not to overload the power supply with startup surges, and afterwards they initiate RAID system pre-checks for correct operation of redundancy. latest server technology is the balancing of increasing speeds of Intel Xeon processors with more memory and I/O capacity. The result is lower latency, higher throughput and continued outstanding performance.scalability lets you grow from a 4-socket to a 32-socket server, or significantly increase your memory - simply by adding additional server chassis.

Diffrent types of server:

Application Server

Also called an appserver. That program handles all application operations between users and an organization's backend business applications or databases. Application servers are used for complex transaction-based applications. To support high-end needs, an application server has to have built-in redundancy, monitors for high-availability, high-performance distributed application services and support for complex database access.

Database Server

A database server is an application which is based on the client/server architecture model. The application is divided into two parts: a front-end running on a workstation (where users collect and display the database information) and the back-end running on a server where the tasks such as data analysis and storage are performed.

Mail Server

Almost as ubiquitous and crucial as Web servers, mail servers move and store mail over corporate networks (via LANsandWANs) and across the Internet. Today, most people think of mail servers in terms of the Internet. Mail servers, however, were originally developed for corporate networks (LANs and WANs).

Web Server

At its core, a Web server serves static content to a Web browser by loading a file from a disk and serving it across the network to a user's Web browser.Any computer can be used as a Web server by installing server software and connecting the machine to the

Internet. There are many web server software applications, like public domain software from NCSA and Apache, and commercial packages from Microsoft, Netscape and others.

FTP Server

An FTP server is a software application running the File Transfer Protocol (FTP), the protocol for exchanging files over the Internet.FTP is most commonly used to download a file from a server using the Internet or to upload a file to a server

1.1 Server Platform

The central feature of the Intel Server boards is that they are designed around the Second Generation Intel Core Processor Family. This processor family is the next generation of 64-bit, multi-core processor, built on 32- nanometre process technology. Based on a new microarchitecture, the processor is designed for a two-chip platform consisting of an Second Generation Intel Core Processor and a Platform Controller Hub (PCH) as shown in figure1.1[1].

Platform Controller Hub (PCH):

It is a family of Intel microchips. I/O Functions have been reassigned between this new central hub and the CPU. Some north bridge functions, the memory controller and PCI-e lanes, were integrated into the CPU while the PCH took over the remaining functions in addition to the traditional roles of the south bridge.

Baseboard management controller:

It is a specialized micro controller embedded on the motherboard of a computer, generally a server and it uses IPMI architecture's intelligence. It is interface between system management software and platform hardware.

The BMC monitors the sensors continuously and send alert reports to remote host via the network if any of the parameters like temperature, cooling fan speeds, power status, if it cross certain limits which can be lead to failure of system. We can also remotely communicate with the BMC to take some action such as resetting or power cycling the system[1].

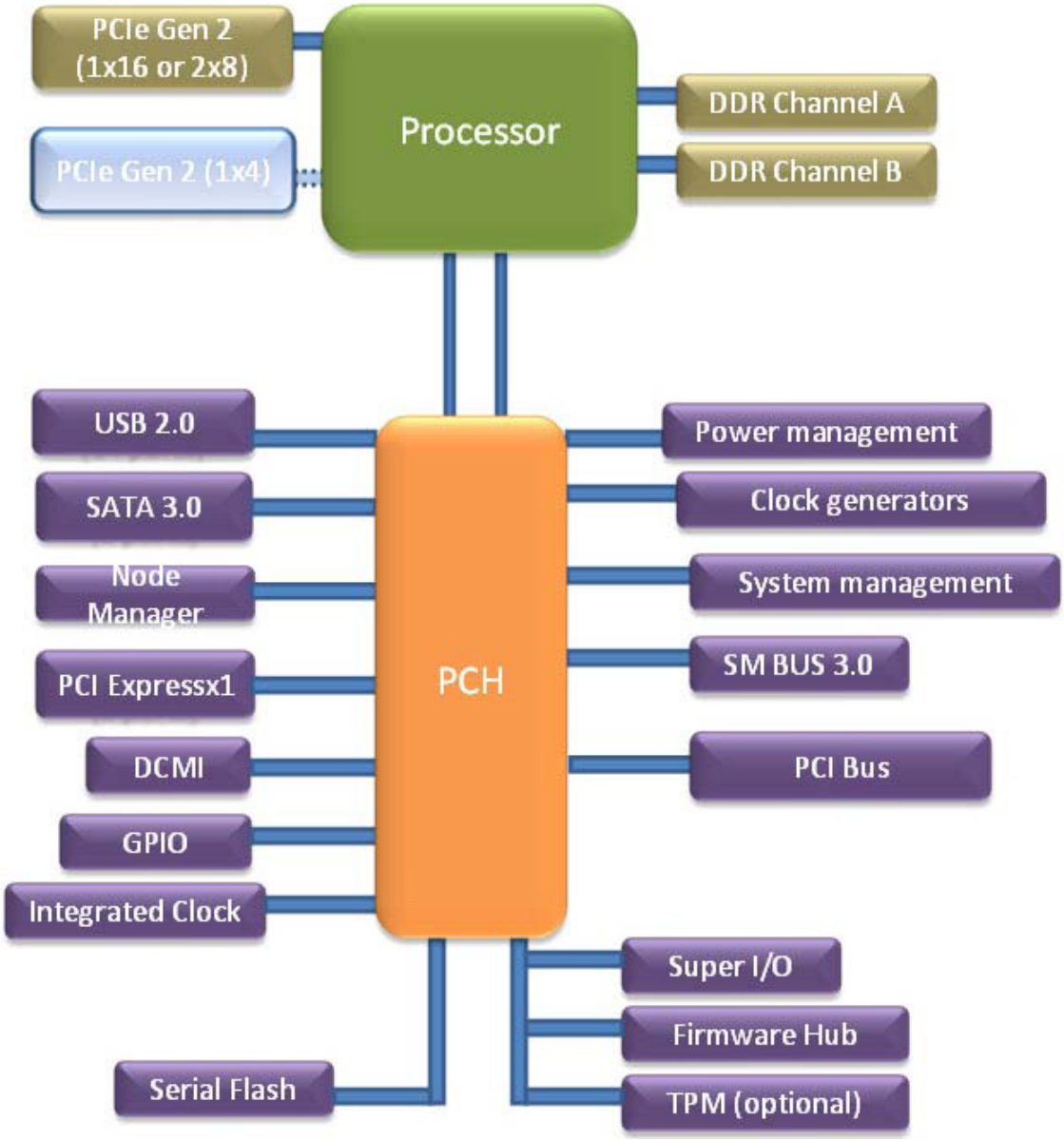

Figure 1.1: Server Platform Design

Chapter 2

BIOS Process Flow

BIOS stands for Basic Input Output System. It used to get the computer system started after press power on button. Set of software routines initialize and test hardware on start up.Power-On Self-Test (POST) refers to routines which run immediately after many digital electronic devices are powered on.

BIOS booting flow describe in below figure2.1[1][3].

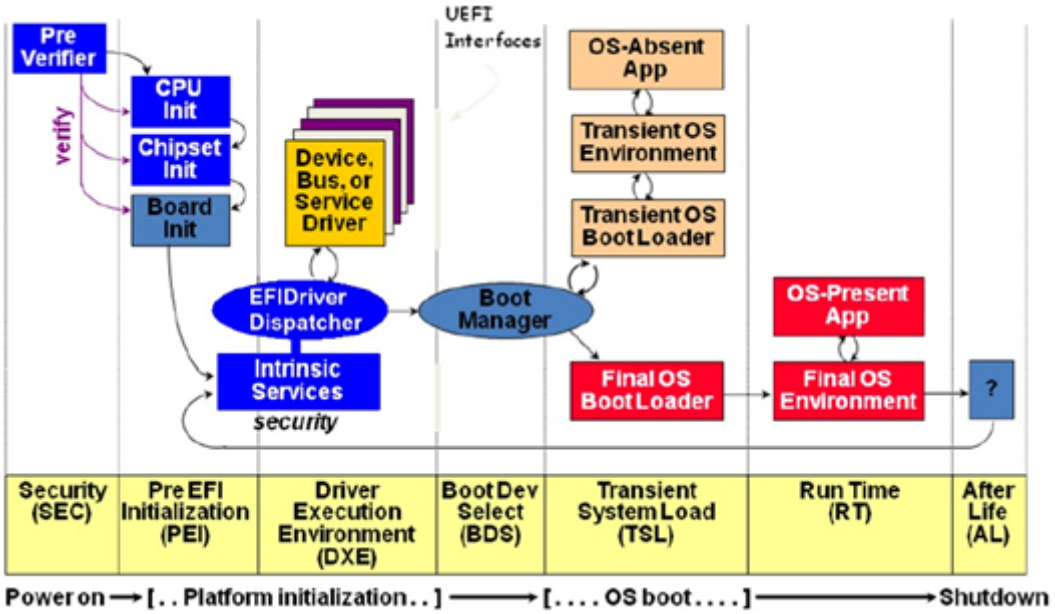

Figure 2.1: BIOS Flow Diagram

SEC Phase

This is the first phase of BIOS boot process it Flush cache and jump into main initialization routine in the ROM. Prior to the discovery of memory on the platform, a data area will be established within the CPU cache so that a stack-based programming language

can be used early in the initialization. Initializes temporary RAM using CPU cache. Also, provides optional security features[3]

PEI

In this phase, executes a series of early hardware initialization such as memory controller hub init, I/O controller hub, initialize built-in platform interfaces .Also determines what the boot mode is we are currently booting with (e.g Normal , Recovery,S3,etc.) [3][6]

DXE

The Driver execution Environment (DXE) is established based on the discovered resources described by the prior PEI phase of operations. It executes PCI enumeration and initialize keyboard, mouse, usb. It also initialize video by using OptionRom[1][3][6].

BDS

Boot Device Select (BDS) phase ultimately will attempt to connect the boot devices required to load and invoke the selected boot target (e.g. O/S)[3].

TSL

Transient System Load(TSL) load OS from selected boot media device[1].

RT

In Runtime(RT) phase OS is loaded and now applications can execute.

UEFI

This Unified Extensible Firmware Interface (UEFI) is an interface between the operating system (OS) and the platform firmware.The interface is in the form of data tables that contain platform-related information, and boot and runtime service calls that are available to the OS loader and the OS. Together, these provide a standard environment for booting an OS[1][3].

Chapter 3

Requirement of Early Video For Intel Server

3.1 Existing Server Scenario

In existing systems, desktop and mobile system have less complicated memory subsystems than servers, and thus take less time to boot the BIOS.Server systems have more complicated memory subsystem and thus, take longer for the BIOS to initialize the memory subsystem and ultimately to boot up.
Servers, specifically, may be slow to boot the video components, which results in a lack of early visual feedback to a user or operator. Systems having video feedback earlier in the boot cycle appear to boot faster, and provide additional boot status to a user. Desktop and mobile systems tend to boot faster than servers. The lack of visual feedback may give the appearance that the system is not booting at all.

In existing systems, many tasks are required to be performed before the video subsystem can be initialized. Thus, the user is deprived of video status of the boot process until the video subsystem has booted. On average, a server may take 30-55 seconds before video may be initialized[2]. A user who is not familiar with the delayed video response time may think that the server has failed to boot and recycle the power button in error.
Currently for video initialization third party video OptionRom is used so that it cannot be initialize before DXE phase. Also memory must be initialized before video initialize because memory required to load video OptionRom and for video memory and IO mem-

Post Code	Error
20-2F	Memory/Chipset
30-3F	Recovery
50-5F	I/O Buses
70-7F	Output Devices
90-9F	Input Devices
B0-BF	Boot Devices

Table 3.1: POST Codes Range

ory. So that User is not able to see any progress, debug, error message if any error comes during booting up of server.

In existing systems,during the Power-On Self Test (POST), the BIOS sends progress codes (POST codes) to I/O port 80h. If the POST fails, the last POST code generated is left at port 80h[10][11]. This code can be used to find out why the error occurred.Also, beep codes are used for diagnostic of system failure. The onboard speaker emits audible error codes (beep codes) during POST. This POST status code and beep code may not be understand by user .

In the tables 3.1, all POST codes and range values are listed in hexadecimal[11].

Port 80h code values typically increase during the boot process. The early codes are for subsystems closer to the processor and the later codes are for peripherals. Generally, the order of initialization is Processor ->Memory ->Busses ->Output/Input Devices ->Boot Devices.In table 3.2[11], some typical value of POST code and description is given.

Some beep codes and error discription is given in table 3.3[10].

3.2 Objective

A method for speeding video initialization in a platform, comprising: configuring a portion of cache memory as cache as RAM (CAR), during power on self test (POST) at boot process time; Video graphics adaptor (VGA) card initialization uses the processors cache memory instead of RAM to perform initialization[2][5]. Expected output of project is show in figure 3.1.

Identify required register setting to initialize VGA and set text mode 80x25 so that we don't need of OptionRom and legacy interrupt. Identify minimal path required to reach to VGA from CPU and initialize required PCI bridge and set PCI configuration space to

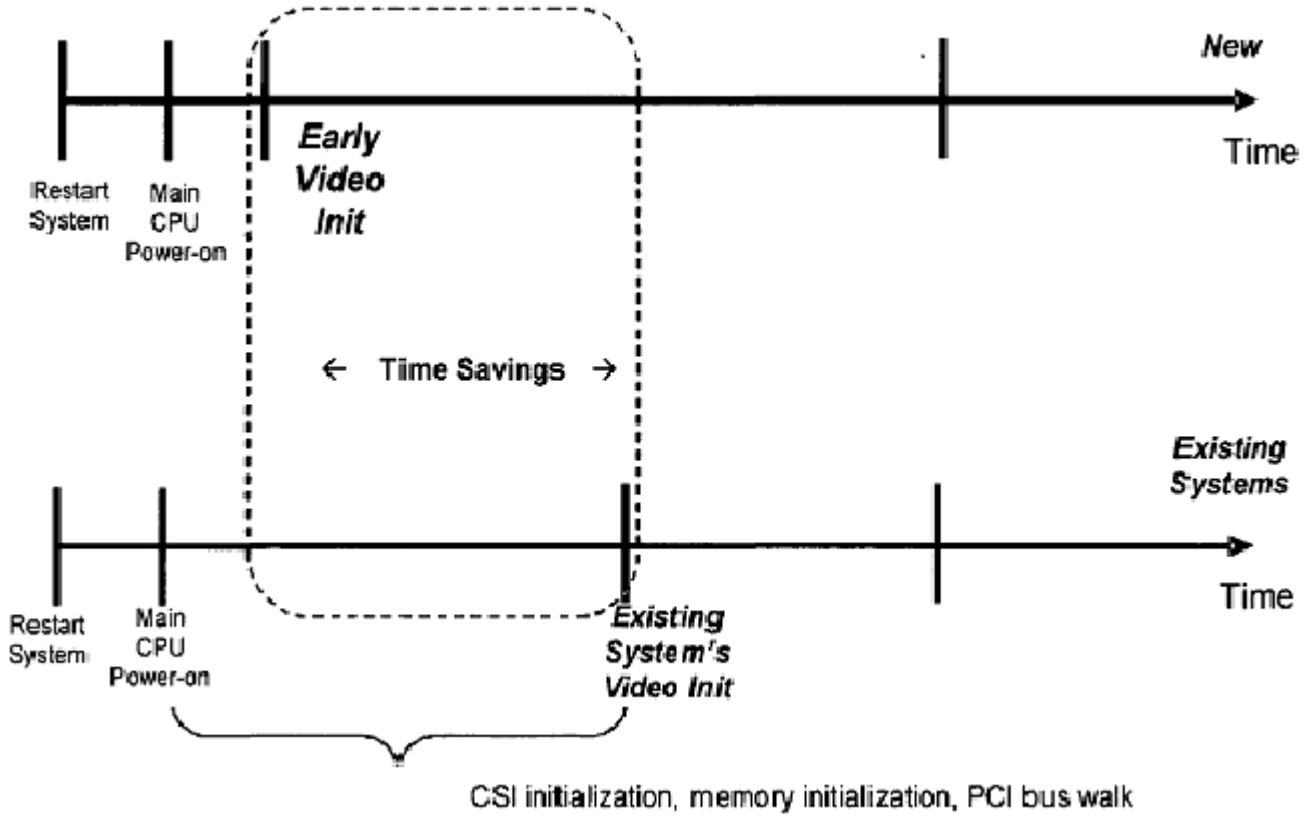

Figure 3.1: Final Outcome

initialize VGA. Video initialization code and PCI set code are mirrored in a CAR and execute. VGA initialization may occur before system RAM has initialized to enable early visual feedback to a user.

Video subsystem initialization is performed during a pre-EFI initialization (PEI) stage during boot up.Once video subsystem initialized, the video console is available to display status messages relating to the boot up process. We required video to come up in 2-3 second after power on server.

Data centre contains racks of servers and they are headless system. For that we need to forward error message to remote host.Serial over LAN (SOL) is a mechanism that enables redirection of serial character stream from UART, (Serial Port) to a LAN.With SOL console redirection system administrators can remotely view the text-based console on their remote servers from anywhere and perform any task that doesn't require a GUI[4]. We need to implement SOL into very early phase of booting so that we can redirect error, debug message to remote host.

Post Code	Description
21	Initializing a chipset component
22	Reading SPD from memory DIMMs
23	Detecting presence of memory DIMMs
25	Configuring memory
28	Testing memory
34	Loading recovery capsule
E4	Entered DXE phase
12	Starting Application processor initialization
13	SMM initialization
50	Enumerating PCI busses
51	Allocating resourced to PCI bus
92	Detecting the presence of the keyboard
90	Resetting keyboard
94	Clearing keyboard input buffe
95	Keyboard Self Test
EB	Calling Video BIOS
58	Resetting USB bus
5A	Resetting PATA/SATA bus and all devices
92	Detecting the presence of the keyboard
90	Resetting keyboard
94	Clearing keyboard input buffer
5A	Resetting PATA/SATA bus and all devices
28	Testing memory
90	Resetting keyboard
94	Clearing keyboard input buffer
01	INT 19
00	Ready to boot
20-2F	Memory/Chipset
30-3F	Recovery
50-5F	I/O Buses
70-7F	Output Devices
90-9F	Input Devices
B0-BF	Boot Devices

Table 3.2: POST Codes Description

Beep Pattern	Error
2 beeps	No Video Detected
3 beeps	Memory Error
high-low beeps	CPU thermal trip

Table 3.3: Beep Code

Chapter 4

Implementation

This feature enable video in very early boot process. Use cache as ram(CAR) to initialize video graphics card and set text mode so that video can comes up as quickly as power on the server. We can display progress message, debug message and error message to diagnostic the system boot up errors and no need to rely on POST status code and beep code .
VGA controller by writing to VGA register and set text mode without loading OpROM and without using legacy interrupt. Use frame buffer to write content on screen directly and make EFI protocol and services for VideoPrint function so that anyone can use it later as per requirement.

4.1 PCI Architecture

PCI architecture and device path for video card shown in below figure 4.1.

The PCI buses and PCI-PCI bridges are the glue connecting the system components together; the CPU is connected to PCI bus 0 is called the primary PCI bus. A special PCI device, a PCI-PCI bridge connects the primary bus to the secondary PCI bus, PCI bus 1. PCI bus 1 is described as being downstream of the PCI-PCI bridge and PCI bus 0 is up-stream of the bridge[9][12].
Peripheral devices have their own memory spaces. PCI device has three address space -

- PCI I/O
- PCI Memory
- PCI Configuration space

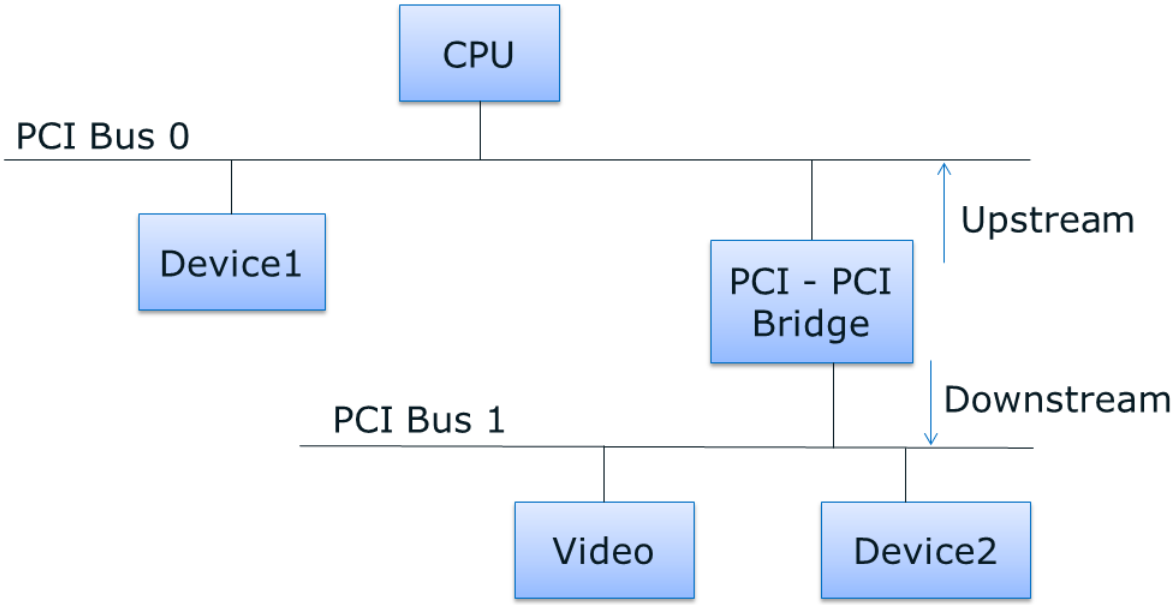

Figure 4.1: PCI Architecture

All of these address spaces can be accessed by the CPU. Device driver will use PCI I/O and PCI Memory address spaces.PCI Configuration space being for the PCI initialization code of device. We need to initialize dummy device path from root bus to Video card to enable video and set text mode by writing to registers of VGA controller.

The PCI Local Bus Specification requires all devices, including a PCI-to-PCI bridge, to implement a 256-byte configuration register address space. The first 64 bytes in each device's PCI Configuration Space must adhere to a standard configuration header format as described in figure 4.2 and 4.3[12][13]. The remaining 192 bytes of the Configuration Space may be used for additional capabilities as defined by the Capabilities Pointer or for device-specific purposes.

The first 16 bytes of the bridge header implement the common format for all devices as required by the PCI Local Bus Specification. The next 48 bytes of the device's Configuration Space are Header Type specific.Two type and it depends on bit 0 ;if its 0 than Standard PCI device and if its 1 than PCI-PCI bridge.

Vendor ID and Device ID: The vendor ID is assigned by the PCI-SIG. The 16-bit device ID is then assigned by the vendor.The Subsystem Vendor ID and the Subsystem Device ID identify the device model. Vendor ID indicates chip manufacturer, and the Subsystem Vendor ID indicates the card manufacturer[12].

Status Register: The Status register is indicate which features are supported and

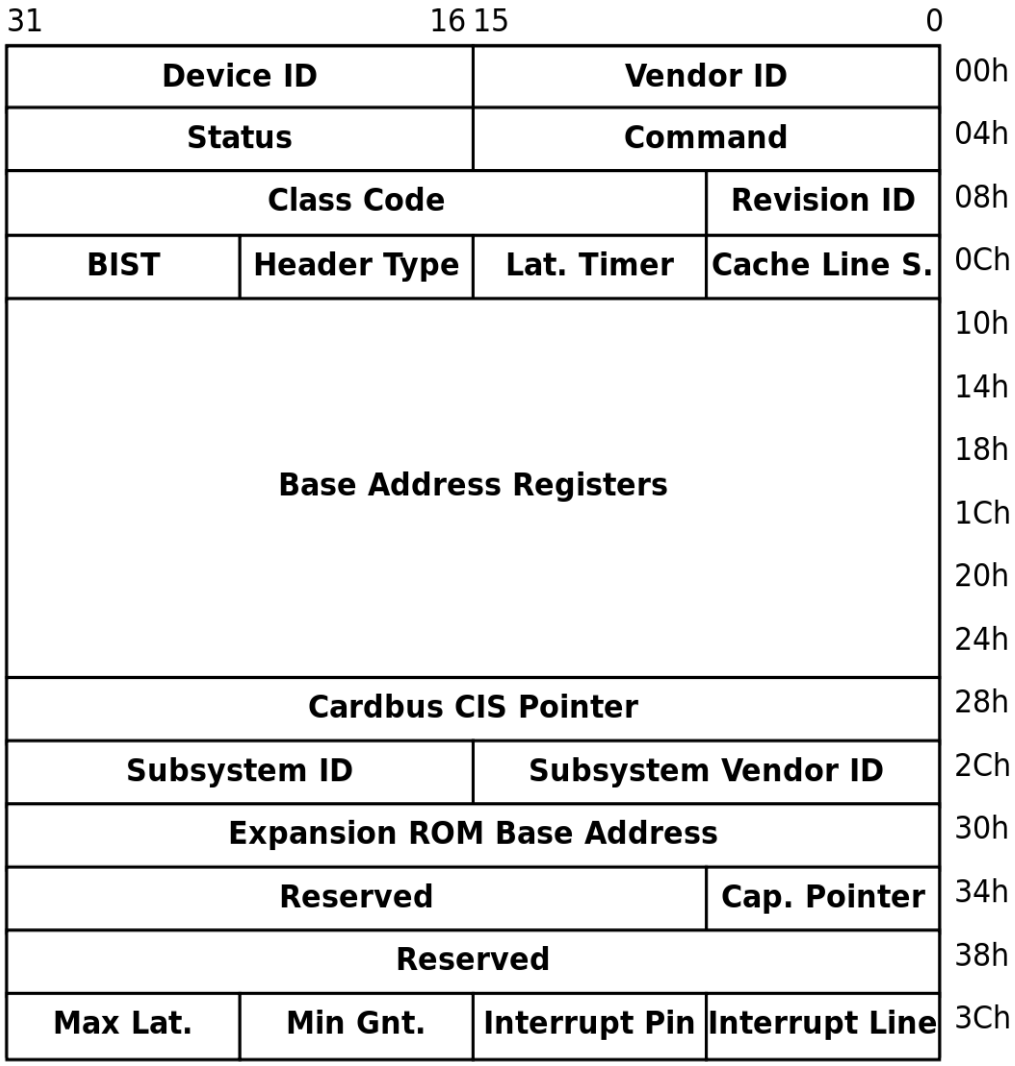

Figure 4.2: PCI Device Configuration Header

whether certain kinds of error have occurred[12].

Command Register: The Command register used to set certain bits like memory ,IO decode bit or VGA palette snoop enable or disable[13].

CacheLine Register: The Cache Line Size register is used to set memory-write-and-invalidate transaction for PCI device[13].

Class Code: It identifies the type of device that this is. There are standard classes for every sort of device; video, SCSI and so on[12][9]. The class code for SCSI is 0x0100.

Base Address Registers: These registers are used to allocate resources to device like PCI I/O and PCI memory space that the device can use and describe base address and limit of resources[9][12][13].

For PCI-PCI bridges to pass PCI I/O, PCI Memory or PCI Configuration address space reads and writes across them, they need to know the following:

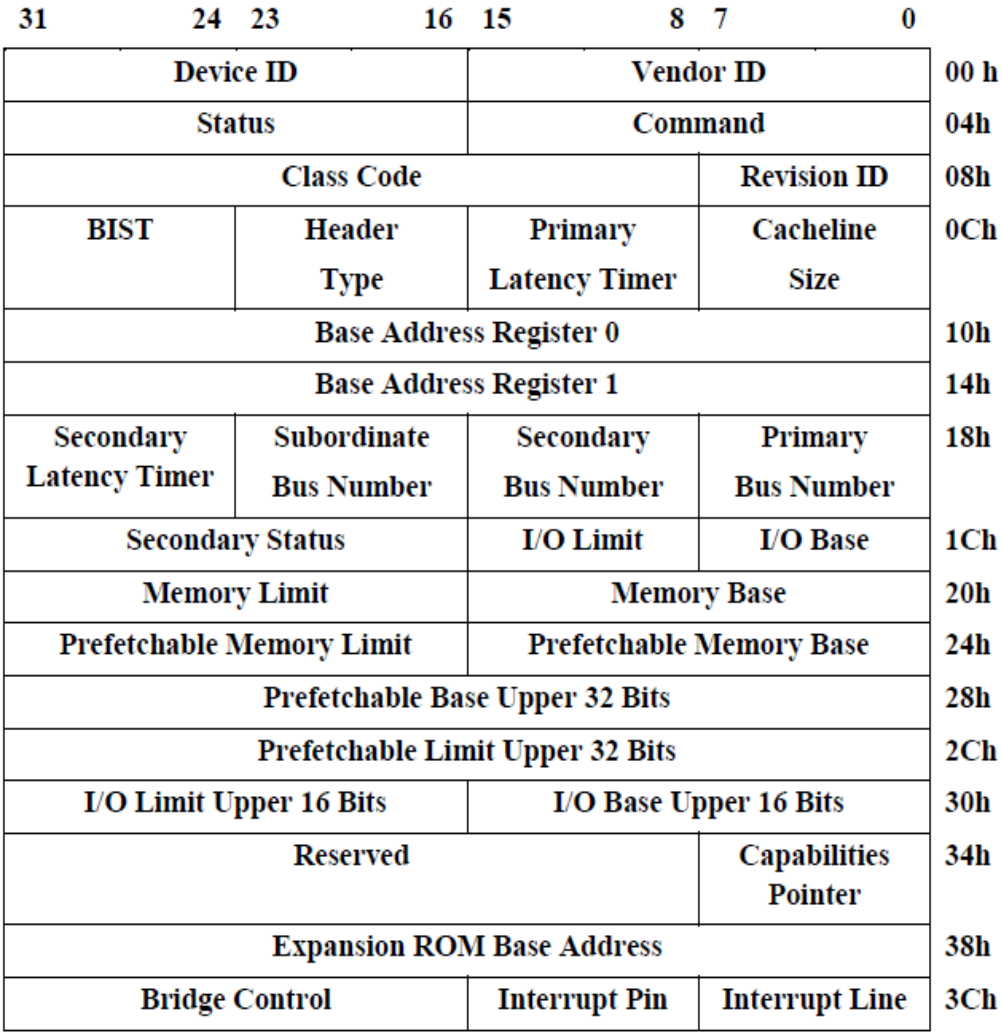

Figure 4.3: PCI Bridge Configuration Header

Primary Bus Number The bus number immediately upstream of the PCI-PCI Bridge

Secondary Bus Number The bus number immediately downstream of the PCI-PCI Bridge

Subordinate Bus Number The highest bus number of all of the buses that can be reached downstream of the bridge.

PCI I/O and PCI Memory Windows The window base and size for PCI I/O address space and PCI Memory address space for all addresses downstream of the PCI-PCI Bridge.

For writing to PCI configuration space we can use PciCfgBase address directly. This address can be vary according to bios implementation.

4.2 FlowChart

This are the function for implementing the EarlyVideo as described in figure 4.4 . We assume CAR is initialized and we used it directly for VGA memory and Io space.

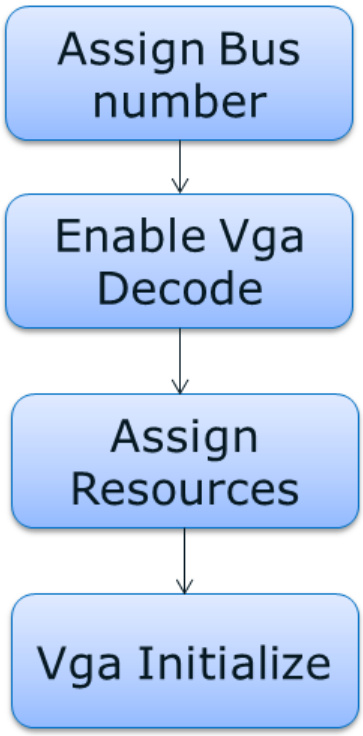

Figure 4.4: FlowChart

4.2.1 Assign Bus Number

PCI initialization code can address devices that are not on the main PCI bus, there has to be a mechanism that allows bridges to decide whether or not to pass Configuration cycles from their primary interface to their secondary interface. A cycle is just an address as it appears on the PCI bus. Here by reading type register's last bit we can decide that whether it's end device if it is 0 or PCI bridge if it is 1[12]. All of the PCI-PCI Bridges seeing configuration cycles may choose to pass them to the PCI buses downstream of themselves. Whether the PCI-PCI Bridge ignores the configuration cycle or passes it onto the downstream PCI bus depends on how the PCI-PCI Bridge has been configured[13]. Every PCI-PCI bridge has a primary bus interface number and a secondary bus interface number. The primary bus interface being the one nearest the CPU and the secondary bus interface being the one furthest away. Each PCI-PCI Bridge also has a subordinate bus

number and this is the highest numbered PCI bus downstream of the PCI-PCI bridge. When the PCI-PCI bridge sees a PCI configuration cycle than it will[12]:

- Ignore it if the bus number specified is not in between the bridge's secondary bus number and subordinate bus number (inclusive),
- Pass it onto secondary bus interface if the bus number specified matches the secondary bus number of the bridge,
- Pass it onto the secondary bus interface unchanged if the bus number specified is greater than the secondary bus number and less than or equal to the subordinate bus number.

To set device path from root bus to Video controller, write on bridge bus 00, device 1C, function 07 to assign a primary bus as 00 ,secondary bus as 01 and subordinate bus as 01.

4.2.2 Enable VGA Decode

15-10	9	8	7	6	5	4	3	2	1	0
Reserved	BBE	SSE	WC	PER	VPS	MWI	SC	BM	MA	IO

Figure 4.5: Command Register

Enable Vga decode, Memory and IO decode bit by writing to command register on bridge bus 00, device 1C, function 07 so that pci bridge knows that every memory and Io access cycle for this end device is for VGA and it transfer to Video controller ; if we not set this bits than it simply ignores the memory and Io access cycle
Also enable Memory and Io decode bit of end device by writing to command register of PCI end device on bus 01, device 00,function 00.

4.2.3 Assign Resources

There are two basic types of Base Address Register, the first indicates within which address space the devices registers must reside; either PCI I/O or PCI Memory space. This is indicated by Bit 0 of the register.

To find out just how much of each address space a given Base Address Register is requesting, you write all 1s into the register and then read it back. The device will specify zeros in the don't care address bits, effectively specifying the address space required.
Calculate resources like PCI memory and PCI Io space required for Video device and assign to it so that we can configure and initialize VGA. We need to program PCI IO and Memory bars and limits to PCI bridge.
PCI IO base address is 0xF000 and Memory base address is 0xF0000000.

4.3 VGA Initialization

This is the Matrox G200e VGA block diagram which is used in Romley servers shown in figure 4.6. We can configure bus controller as describe above. For text mode we don't need to set drawing engine because it is used for high resolution graphics mode.The VGA function includes different component as described below.

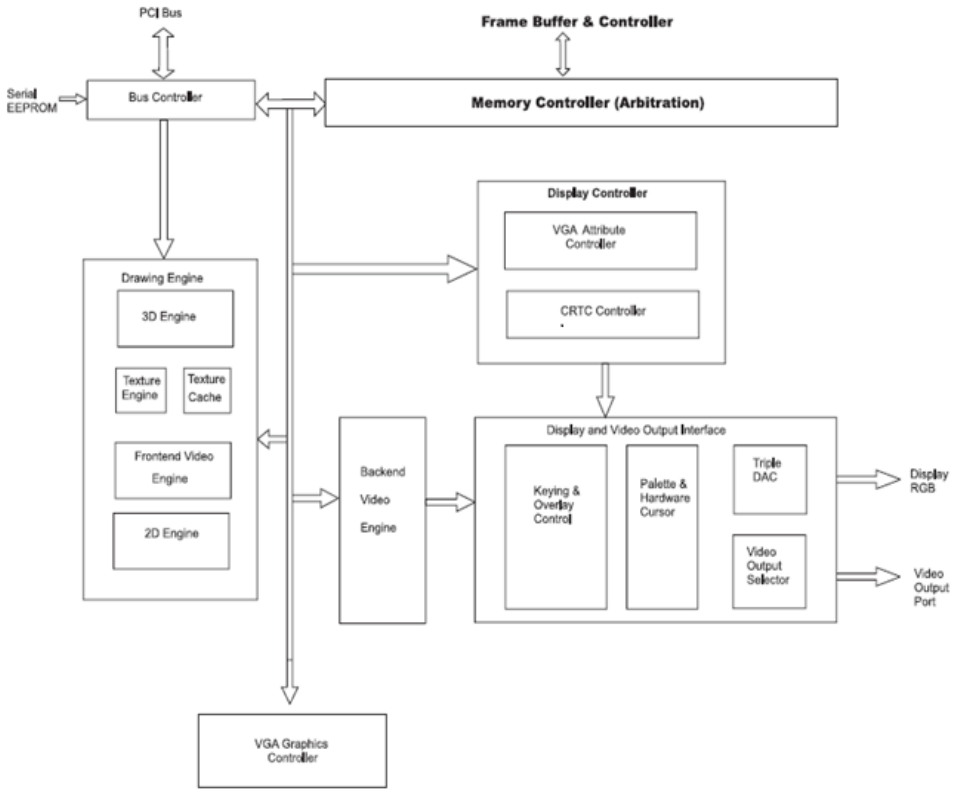

Figure 4.6: VGA Block Diagram

- **Video DAC :**

  Video DAC register is used to convert the video data into the video signal which

is sent to the display.It basically contains the colour palette.Red, green, and blue analog signals are output from the DAC[7].

- **CRT controller:**
  CRT controller used for setting different timing like horizontal and vertical synchronization signal timings, cursor and underline timings, refresh addressing for the video memory and addressing for the regenerative buffer[8].

- **Sequencer register:**
  Sequencer register used for
  - Setting basic memory timings for the video memory and character clock for controlling regenerative buffer fetches.
  - It allows the system to access memory during active display intervals by periodically inserting dedicated system microprocessor memory cycles between the display memory cycles.
  - Set map mask registers in the sequencer are available to protect entire memory maps from being changed[7].

- **Graphics controller:**
  This register is the interface
  - Between Video memory and the attribute controller during active display times
  - Between Video memory and the system microprocessor during memory accesses.

- **Attribute controller:**
  AC register takes in data from video memory through the graphics controller and formats it for display like set colour, etc[8][7].

Load font at location 0xA0000 and use 0xB8000 address as frame buffer base address to write a message on screen directly. Implement function like Clearscreen and VideoprintatXY.

Install videoprint function as pei service and dxe protocol so that anyone can use this function at boot process time to print any messages.

We put all this code in PEI module after NEM(non evict mode) is set so that cache can

be used as RAM.After CAR is enabled EarlyVideo code is called and it perform all step mentioned above to set VGA.

4.4 Console Redirection

For headless system in datacenter we need to send messages to remote host so for that we need to write messages to serial port A/serial port B.To display data to remote host we need console redirection enable if we are using null modem cable as connection between remote host and target and using on board serial peripheral ;other way is using SOL(Serial over LAN).

4.4.1 Using Null Modem Cable and On board serial peripheral

BIOS Configuration

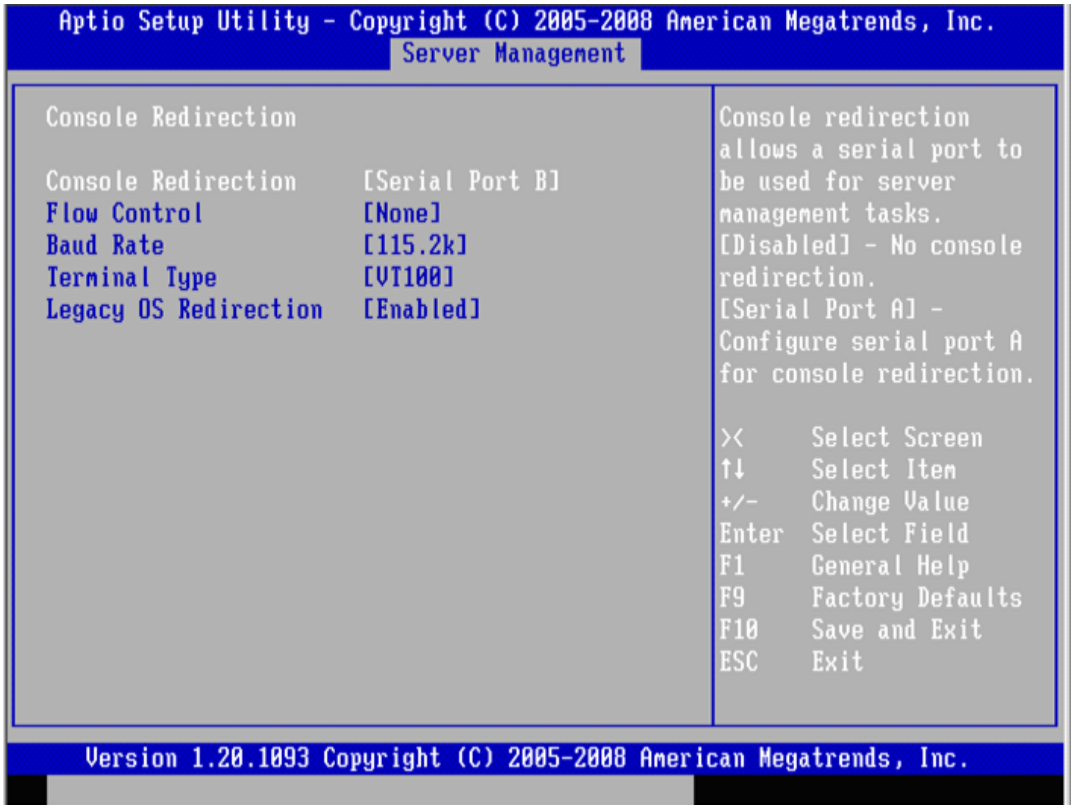

Figure 4.7: BIOS console screen

1. Restart the server and press F2 when prompted to start the BIOS Setup
2. Select Devices and I/O Ports; then, make sure that the Serial Ports value is set to Enabled

3. Go to the Server Management tab and select Remote Console Redirection;then, make sure that the values are set as follows.

   - BIOS Redirection Port: Serial
   - Baud Rate: 19.2K (Other baud rates which is supported)

4. Press Esc to exit the Remote Console Redirection and Devices and I/O Ports

5. Press F10 and make sure that OK is selected; then, press Enter.

4.4.2 Using SOL-Serial Over LAN

Serial over LAN (SOL) is a mechanism that enables redirection of serial character stream from UART (Universal Asynchronous Receiver Transmitter aka Serial Port) to a LAN using IPMI RMCP+ session on a managed server[4]. The advantages of SOL include the following:

- Remote administration without keyboard, video, or mouse (for example, headless servers)
- Reduced cabling by removing the need for a serial connector
- Standard Telnet interface that eliminates the need for special client software

With SOL console redirection system administrators can remotely view the text-based console on their remote managed servers from anywhere and perform any task that doesn't require a GUI. Thus SOL out-of-band remote console can be used from any location to diagnose and repair problems - eliminating the need to physically go to the system.

Overview on SOL

SOL is based on RMCP (Remote Management and Control Protocol) request-response protocol delivered using UDP datagrams to port 623[4] and it needs software running on the Baseboard Management Controller (BMC) and client software running on a management workstation and/or central network proxy. The BMC provides the intelligence behind Intelligent Platform Management. The BMC manages the interface between system management software and the platform management hardware, provides autonomous monitoring, event logging, and recovery control The BMC firmware is responsible for controlling the serial hardware MUX, the transformation of serial data to and from network

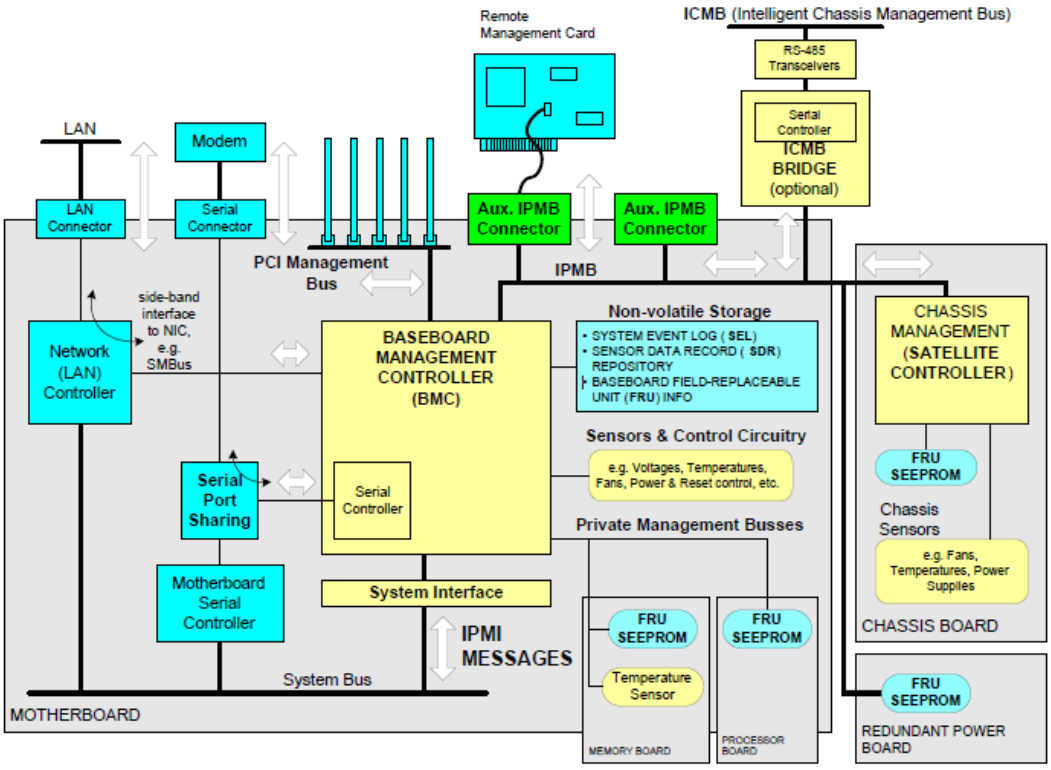

Figure 4.8: SOL Block diagram

packets, and the transmission and reception of SOL network packets through the NIC port. A remote SOL client is responsible for initiating the SOL session with the BMC and transformation of console input and output to and from network packets. To allow access from Telnet style programs, primary SOL services are implemented as a background network task. Throughout this document, this task will be referred to as the network proxy. The network proxy can run on individual management workstations or may be a centralized service that can be used by any management workstation.

System Configuration to Enable SOL Redirection

The following section provides step by step instructions on how to configure BIOS and BMC firmware to enable SOL using IDA [Intel Deployment Assistant] or Syscfg tool.BIOS configuration is same as explained above.

BMC Firmware Configuration Along with the BIOS configuration, the firmware in the Baseboard Management Controller (BMC) has to be configured for Out-of-band [OOB] communication over LAN Setting Users and Enabling SOL. The following are the BMC settings which can be done either through Syscfg utility or IDA.

- IP source (static or DHCP)

- IP Address
- Subnet mask
- Default gateway (only required if you will be connecting from client outside of subnet)
- Enable one user
- Enable users privilege level
- Set Users and passwords
- Enable text based console redirection (serial Over LAN - SOL)

Enabling and Configuring LAN Channel, User Settings and SOL Configuration Through IDA

- Select Configure a Server after boot from IDA CD
- Select Server Management Settings and click Next button
- Click LAN Channel 1 (onboard NICS1) if you want to configure BMC LAN channel 1, click Next button
- Select IP Address From a DHCP Server or Static IP Address for BMC LAN Channel IP and key in your IP address/Subnet Mask/Gateway depends on your network configuration

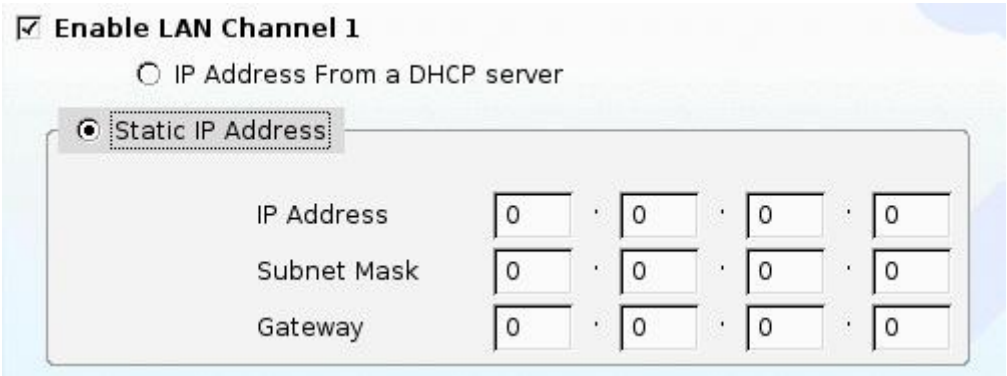

Figure 4.9: Lan Configuration

- Select Anonymous User line and click Edit button to configure BMC anonymous user.

- Under Edit User Data pop-up window, you can enable the user account you selected and assign user privileges to this user. Make sure to select Change Username and Password and enter password/confirm password, then click OK button:

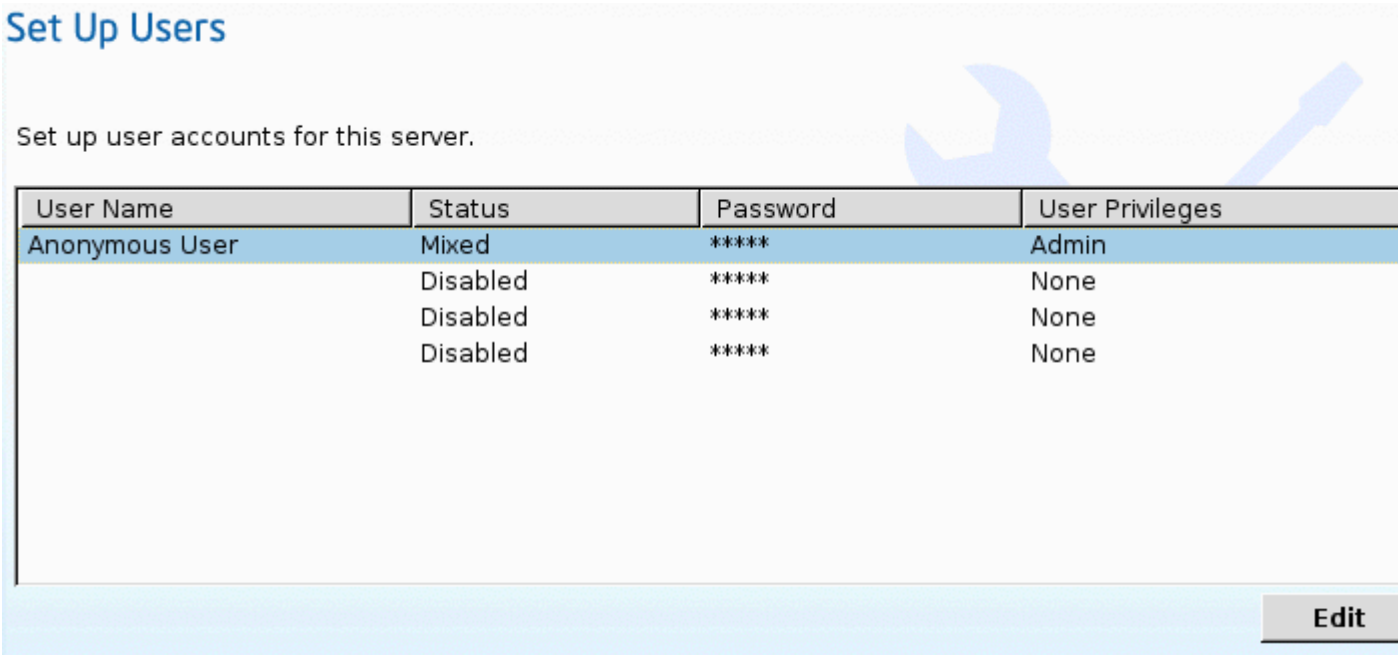

Figure 4.10: User Configuration

- Select the Enable Serial Over LAN option along with the LAN configuration option in the LAN Channel configuration page. This enabled the SOL configuration.

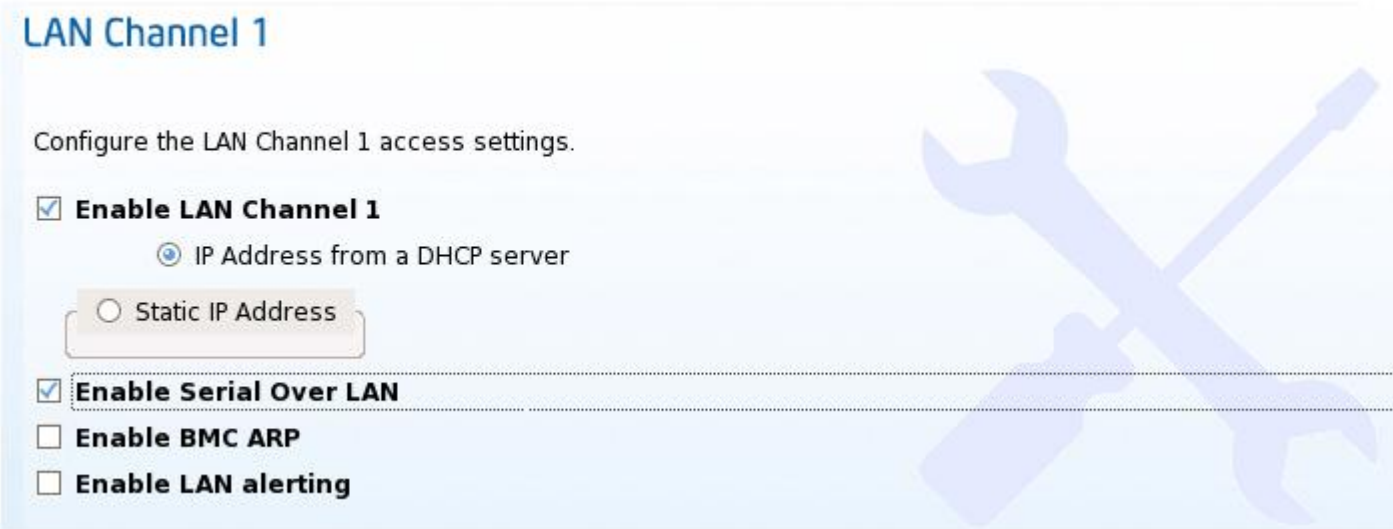

Figure 4.11: Enable SOL

4.4.3 Console Write Function

To write data to serial port A/B follow below mentioned steps

1. As mentioned above enable console redirection using bios setup.
2. On save and exit of setup option write the value: Serial port A/B, baud rate(115200,9600,19800,etc ,Serial port address(0x3f8,0x2f8,0x3e8,0x,2e8) to any cmos address using callback method.
3. Write a character one by one to serial port according to configuration settings.
   (a) Read configuration setting from cmos address.
   (b) Write baudrate decode bit and serial port A/B to pci configuration space of LPC com port.
   (c) Enable that serial ports by writing to Pilot3 activate register and write the address to register.
   (d) Send data character by character to serial port.

Chapter 5

Results and UseCase

Initializing the VGA early in the BIOS provides the ability to display the POST progress status for all phases of the boot (except SEC phase) and cover all POST errors that could occur in the system which could prevent the system to boot. This feature can also be used to display user configurable jumper settings in the system like the Recovery, Manufacturing, BMC force update, Password clear jumper etc. In brief, all error messages beyond the SEC phase of the boot can be displayed on the VGA thus providing better user experience through the POST

5.1 Post Progress

UseCase1: Memory Installed

- Power on/Reset system
- User can see progress message on screen

  "Memory Initialization complete. No errors found."

UseCase2: Chipset Initialized

- Power/Reset system
- User can see progress message on screen

  "Chipset Initialization complete. No errors found."

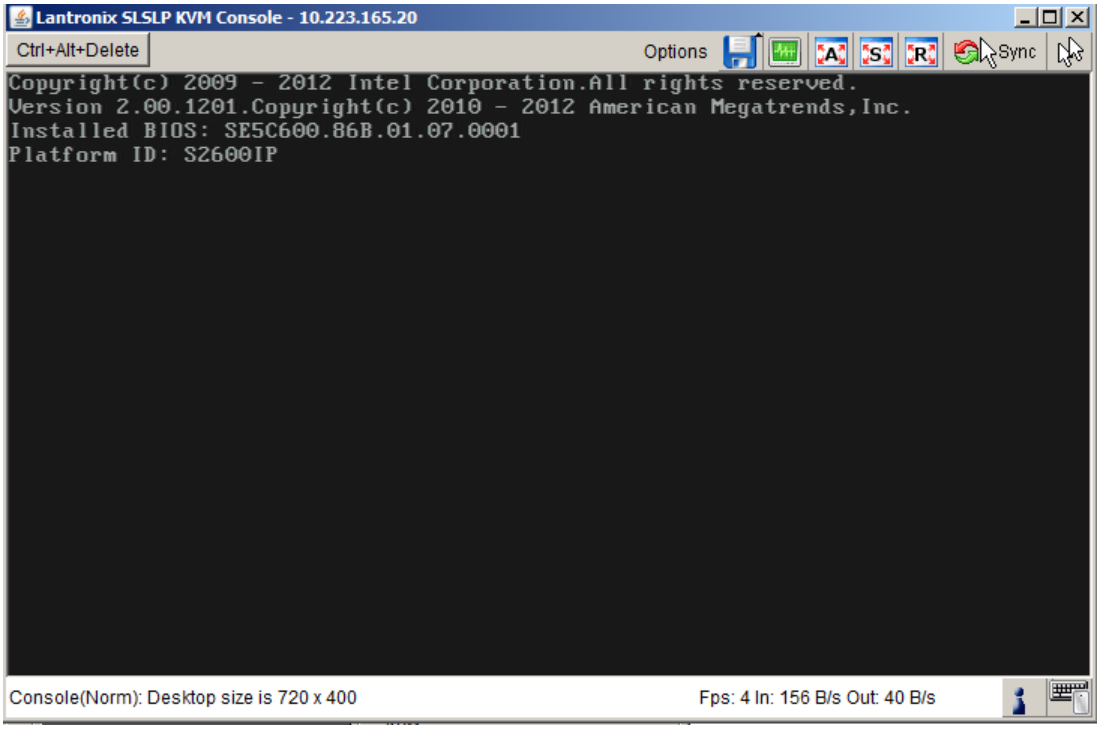

Figure 5.1: Normal Boot

5.2 Post Error

UseCase1: No RAM or Bad RAM is attached

- Remove all RAM from server board or insert bad RAM
- Reset system
- User can see error message on screen rather than read POST code or beep code.

  "No usable Memory found." or "Memory initialization failed."

UseCase2: Recovery Flash is not attached

- Recovery jumper
- Power/Reset system
- User can see error message on screen

  "Recovery capsule not found."

UseCase3: Mix UDIMM and RDIMM

- Insert one UDIMM and one RDIMM

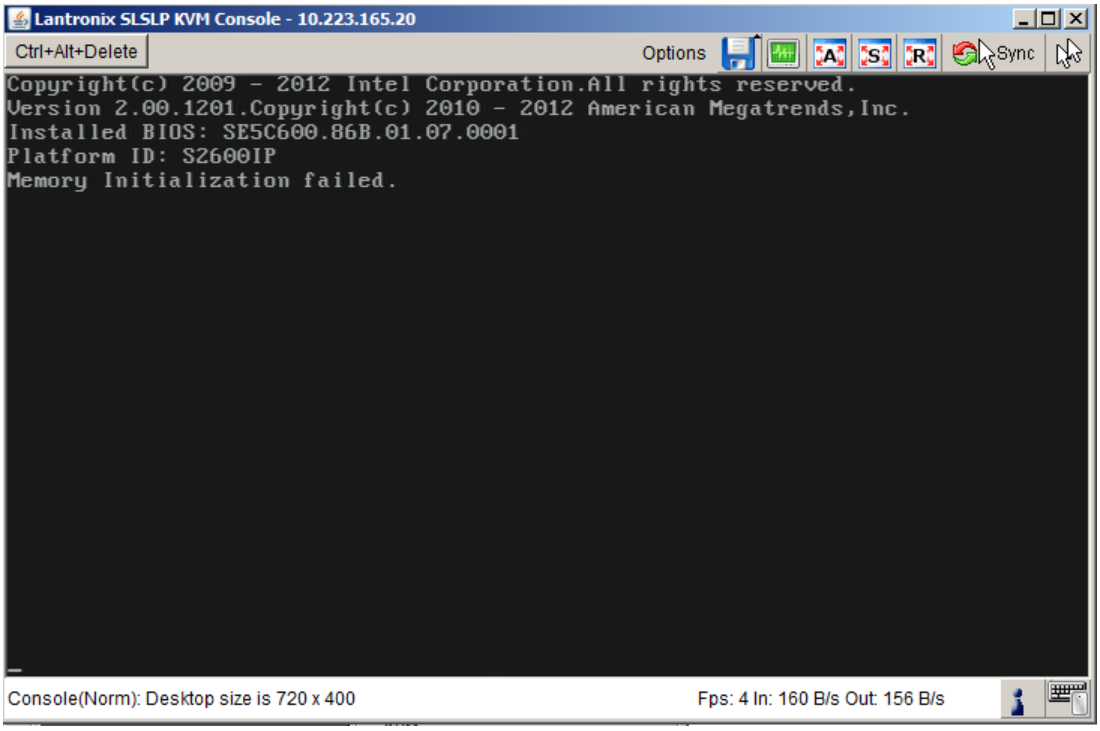

Figure 5.2: Boot Error-Bad RAM

- Power on system
- User can see error message on screen

  "Invalid DIMM population."

5.3 Jumper Status

UseCase1: Password Clear Jumper Set

- Set password clear jumper
- Power on system
- User can see message on screen

  "Password clear jumper is Set."

UseCase2: BMC in Update mode

- Set BMC update mode jumper
- Power on system

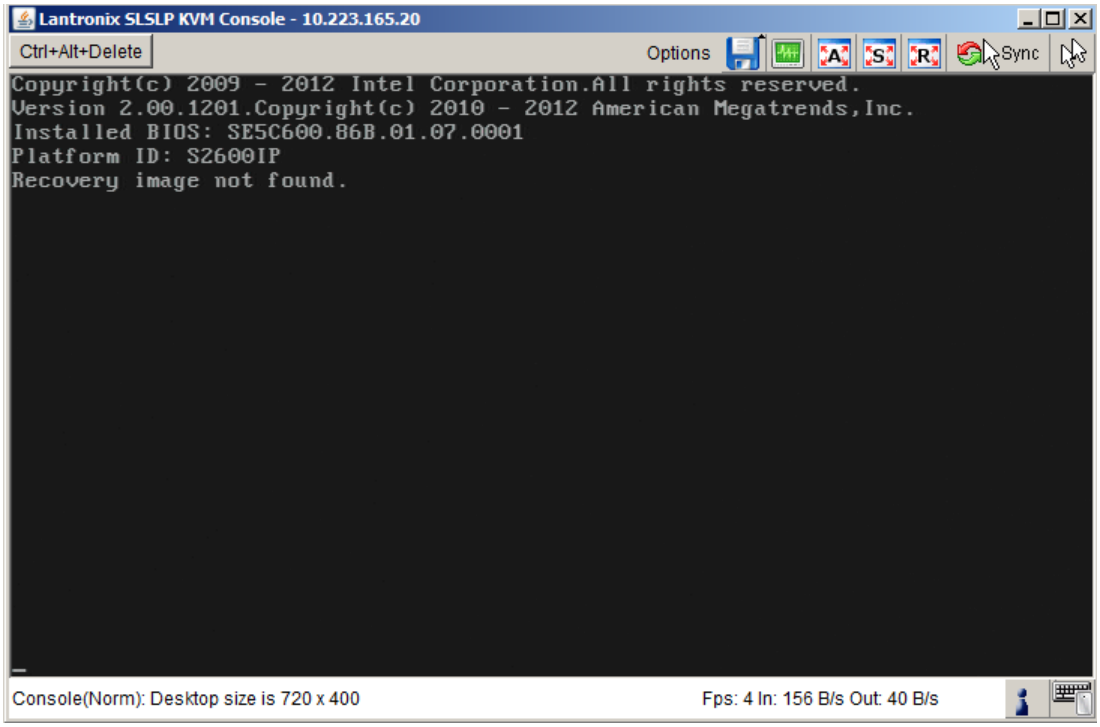

Figure 5.3: Boot Error-Recovery Flash is not attached

- User can see message on screen

  "Baseboard management controller in update mode."

UseCase3: Set BIOS to Default setting

- Set BIOS default setting jumper
- Power on system
- User can see message on screen

  "BIOS Settings reset to default settings."

5.4 Console Redirection Results

Using Null Modem Cable and On board serial peripheral

- Connect remote host to server using null modem cable and on board serial peripheral.
- Power on server and go to bios setup
- Go to server management tab and enable console redirection and select port serial A/B and set baudrate.

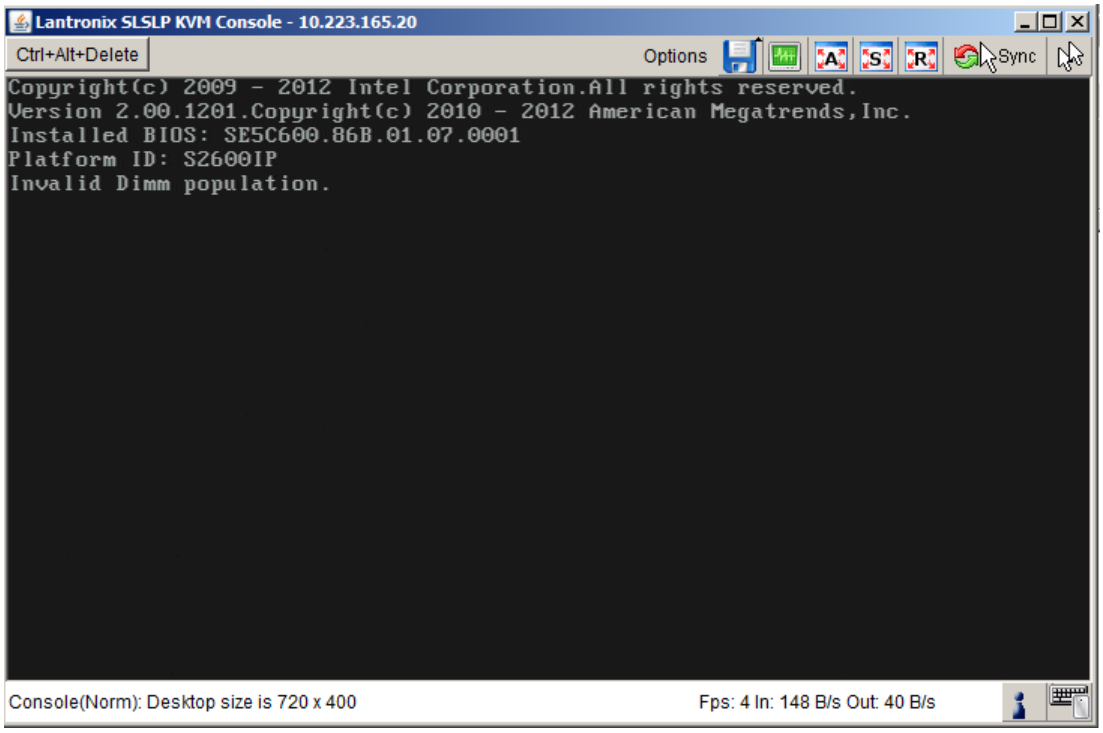

Figure 5.4: Boot Error-Mix UDIMM and RDIMM

- Restart the server
- Open hyper terminal or ace software on remote host select appropriate baudrate and com settings and start session.
- User can see messages on remote host

Using SOL-Serial Over LAN

- Do setting for sol as mentioned in previous chapter.
- Go to the terminal and write command

  ipmitool -H ipaddr -U user -P password -I lanplus sol set volatile-bit-rate 19.2 non-volatile-bit-rate 19.2 activate

- User can see messages on terminal.

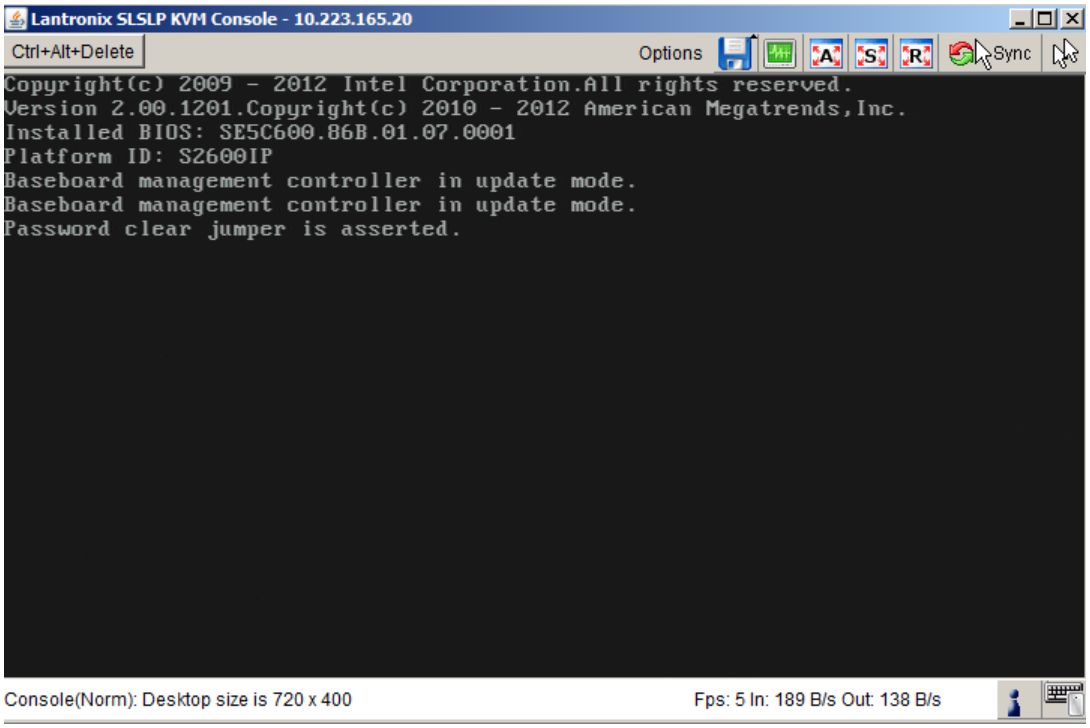

Figure 5.5: Jumper Status

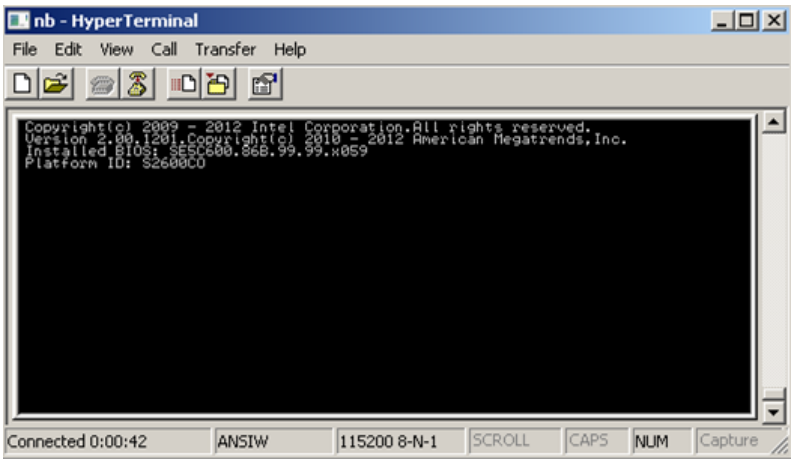

Figure 5.6: Hyper terminal

Figure 5.7: SOL output

Chapter 6

Future Enhancement

Initialize graphics mode of VGA instead of simple text mode 80x25 to enhanced more user experience and image or intel logo can be displayed at early phase of booting server

Chapter 7

Conclusion

- It enhanced the User experience as video comes up in 2-3 second on system power on.
- System Errors can be displayed on screen using early video e.g Memory errors, Recovery errors etc.
- No need to open chassis to know the status of certain jumper
- No need to monitor Beep codes and LED codes
- Message can be displayed on remote host for headless servers in datacenter

Bibliography

[1] Michael othman,Tim Lewis,Vincent Zimmer, Robert Hale, Harnessing the UEFI Shell, Intel press ,March 2010.

[2] Robert C. Swanson,Michael A. Rothman, Mallik Bulusu, Vincent J. Zimmer,Instant on video,Intel corporation ,July 26,2011.

[3] Unified Extensible firmware Interface Specifaction, version 2.3,May 2009

[4] Serial Over LAN (SOL) for EPSD Server Hardware, Intel Corporation,May 2010.

[5] Mike Kartoz, Pete Dice, and Gabe Hattaway,Fastboot BIOS,Intel Corporation Embedded and Communications Group,September 2008

[6] Michael A. Rothman,Reducing Platform Boot Times UEFI-based Performance Optimization,Intel Corporartion,2009.

[7] http://read.seas.harvard.edu/cs261/hwref/ibm-vga.txt

[8] www.osdever.net/FreeVGA/

[9] www.pcisig.com/specifications/pciexpress/specifications

[10] www.intel.com/support/motherboards/desktop/sb/cs-010249.html

[11] www.intel.com/support/motherboards/desktop/sb/CS-025434.html

[12] http://tldp.org/LDP/tlk/dd/pci.html

[13] PCI-to-PCI Bridge Architecture,PCI Special Interest Group,December 18,1998.