OBJECTIVES: At the end of this section, the student will be able to do the following:

- Explain a picture of the 0-1 megabyte memory map and discuss the function of each section.
- Describe the organization of PC Base Memory.
- Explain the organization of PC Reserved Memory.
- Discuss the High Memory Area & Extended Memory and how they are accessed.
- Find the I/O addresses of specific I/O chips.
- Discuss a Generic Decode Logic for decoding I/O addresses.
PC MEMORY MAP

- The original PC used an 8088 microprocessor which limited the memory map to a size of 1M byte.
  - To maintain compatibility with the original PC's BIOS and DOS operating system, the memory map of the PC/AT is fundamentally limited to 1M byte.
  - Efforts have been made to overcome the 1M byte limitation
    - LIM Expanded Memory--permits access to up to 32M byte through a 64K byte window in the 1M byte address space.
    - Various DOS extenders and operating systems have been written to run the PC in protected mode, making all bytes directly addressable: OS/2, WINDOWS, DESQVIEW, as well as RAM-disk and disk caching programs.
# System Memory Map

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>DESCRIPTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>640K SYSTEM MEMORY</td>
<td>REAL MODE</td>
</tr>
<tr>
<td>09FFFF</td>
<td></td>
<td>ON BOARD DRAM</td>
</tr>
<tr>
<td>0A0000</td>
<td>128K VIDEO RAM</td>
<td>DISPLAY BUFFER</td>
</tr>
<tr>
<td>0BFFFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0C0000</td>
<td>128K I/O EXPANSION ROM</td>
<td>RESERVED FOR ROM</td>
</tr>
<tr>
<td>0DFFFF</td>
<td></td>
<td>ON I/O ADAPTERS</td>
</tr>
<tr>
<td>0E0000</td>
<td>64K ROM OR DRAM*</td>
<td>BIOS EXTENSION OR RAM</td>
</tr>
<tr>
<td>0EFFFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0F0000</td>
<td>64K ROM OR DRAM*</td>
<td>ROM BIOS OR RAM</td>
</tr>
<tr>
<td>0FFFFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100000</td>
<td>HIGH MEMORY AREA</td>
<td>ON BOARD DRAM</td>
</tr>
<tr>
<td>10FFEF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10FFFF0</td>
<td>EXTENDED SYSTEM MEMORY</td>
<td>ON BOARD DRAM</td>
</tr>
<tr>
<td>FFFE00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFFFE0000</td>
<td>128K ROM OR DRAM*</td>
<td>RESERVED FOR SYSTEM</td>
</tr>
<tr>
<td>FFFFFFFF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BASE
(Conventional)
MEMORY
BASE (Conventional) MEMORY

- **Interrupt Vector Table**
- **BIOS DATA**
- **DOS DATA**
- **I/O . SYS**
- **COMMAND.COM**
- **DEVICE DRIVERS**
- **MSDOS . SYS**
- **I/O . SYS**
- **DOS DATA**
- **BIOS DATA**
- **Interrupt Vector Table**

**Memory Segments:**
- A0000h: 640K
- 512K
- 256K
- 128K
- 64K

**Memory Addresses:**
- 0000h
- 0400h
- 0500h
- 256K
- 512K
- 128K
- 64K
- ~0700h
- ~12F0h
- ~2740h
- ~5DD0h / FB10h
- 0000h
- 0400h
- 0500h
- 0700h
- 12F0h
- 2740h
- 5DD0h / FB10h

**Memory Pages:**
- 64K
- 128K
- 256K
- 512K
- 640K

**Special Memory:**
- BOOTLOADER @7C00h
BASE MEMORY

The first 640K bytes of memory is used for the following:

- The interrupt vector table--1K bytes
- BIOS data area--starting at 400H
- DOS buffers, control areas, and installed device drivers
- DOS code--functions calls, interrupts handlers, etc.
- The DOS Shell, most often COMMAND.COM
BASE MEMORY

- Depending on the DOS version, the above may occupy anywhere from 25 to more than 40K bytes.
  - More installed drivers and DOS buffers means more memory is used.
- The remaining memory, up to the 640K byte boundary, is available for application programs.
- TSR (terminate-stay-resident) programs are installed after COMMAND.COM and also eat into the application memory space.
PC RESERVED MEMORY
RESERVED MEMORY

POST BIOS

EMPTY FOR USER SUPPLIED ROMS--OPTIONAL

VIDEO BIOS (ROM)

CGA

MDA

EGA, VGA

RAM ON ADAPTERS

OPTIONAL ADAPTER CARD ROMS

FFFFFH

F0000H

EFFFH

E0000H

D0000H

C8000H

C7FFFH

C0000H

B8000H

B0000H

A0000H
RESERVED MEMORY

- The memory from 640K to 1M byte is reserved for system use and is not typically available for applications.

- The memory reserved for the display adapter is typically on the display adapter itself. Writing into the memory reserved for the display adapter will directly change what appears on the screen.

- The video BIOS is also typically in an EPROM on the display card.
  - The video BIOS can often be copied (shadowed) into faster DRAM for improved execution speed. The code is copied, the EPROM disabled, and the DRAM enabled at the same address.
RESERVED MEMORY

- Provision has been made to add adapter cards with ROM, between the addresses 0C8000H and 0DFFFFFFH (General Expansion ROM).
  
  - NOTE: POST searches for Video ROM between C0000H-C8000H, & 64K of System Expansion ROM @ E0000H

- For POST to detect an Expansion ROM, the first 4 bytes of each Expansion ROM must be set up as follows:
  
  - Byte 0 = 55H; Byte 1 = AAH
  - Byte 2 = ROM Length in 512-byte blocks
  - Byte 3 = Entry point for ROM initialization (via FAR CALL)

- EXAMPLE: A POST card, located at 0D0000H, is given control before the machine loads DOS.
RESERVED MEMORY

ROM SCAN AREA

EGA / VGA Video BIOS

"55 AA" LENGTH....SUM 0

128 k

32 k

64 x 2k segments

PC Architecture For Technicians Level-1
Technical Excellence Development Series
RESERVED MEMORY

During POST, the BIOS searches the 0C0000H to E0000H (128K) region on 2K byte boundaries and does the following:

- Looks for the signature 55H AAH
- Finding it, does a Checksum on the number of 512 blocks indicated by the next byte.
  - e.g. 40H = 32K ( 64t * .5K = 32K)
- If the Checksum was Zero, transfers control to the third byte from 55H with a far subroutine call. (e.g. 0C0003H for Video)
- The code on the ROM can then initialize the adapter card and return to BIOS with a "RET" instruction.
RESERVED MEMORY

- The area from 0E0000H to 0FFFFFFFH is reserved for BIOS.
  - Typically only the upper 64K is used (F0000-FFFFFH).
  - The remaining 64K bytes can be occupied with a user supplied ROM (E0000-EFFFFF)
  - BIOS checks address 0E0000H for 55H 0AAH to determine whether an optional ROM is installed and, if so, calls the option ROM as described above.
- The upper 64K bytes contains the POST and BIOS code and a character set for CGA graphics.
- The upper 128K bytes can be shadowed for better performance.
RESERVED MEMORY

TOP OF MEMORY

- The addresses from \texttt{FFFE0000H} to \texttt{FFFFFFFFFFH} are also decoded in the range \texttt{000E0000H} to \texttt{000FFFFFH}.
- The 80386 and above processors hold the upper address lines high after reset until the first FAR transfer.
  - The first FAR transfer actually occurs on the first instruction in a PC compatible BIOS.
  - After the FAR JUMP to just below 1MB, the remainder of POST will execute in the compatible below 1MB region.
RESERVED MEMORY

<4GB Range

- FFFFFFFFH
- FFFF0000H

<1 MB Range

- 000FFFFFFH
- 000F0000H

BIOS ROM

64K BYTES
OTHER
PC
MEMORY
EXPANDED MEMORY

- Expanded memory is a bank-switching standard devised by Lotus, Intel, and Microsoft (hence the name LIM).
  - In the simplest form, the user must set aside 64K of memory (anywhere in the 1M memory map).
  - The region--0E0000H to 0EFFFFFH is often used because it is the region left empty for a user installed ROM.
- If the computer has extended memory, a driver can be installed to make it look like expanded memory.
- Because the 80386 & above processors can address Extended Memory in Protected Mode, Expanded Memory is less common today.
A20 GATE

- It is possible when operating real mode to address 1M byte plus 64K bytes. How is this possible?
- start a segment at 0FFFF:0000
  - the base address is FFFF0H
  - add the largest possible offset + 0FFFFH
  - the linear address is 10FFEFH
    - (0FFEFH is almost 64K above 1Mbyte)
- In the 8088 microprocessor, the 1 does not exist because there are only 20 address lines (A19:0).
- The address on the bus is then 0FFEFH
  - An address in the low part of memory (1st 64K).
A20 GATE

- To make the PC/AT compatible with XT software, address line A20 is basically grounded to force the wrap around to low memory.
- However, to use extended memory, A20 must be enabled, thus the A20 gate.
- The A20 gate is implemented in the Keyboard Controller.
- A “FAST A20” is often implemented in Port 92 (Port “A” Bit 1) or Port 78 or some other Vendor Specific I/O Port.
  - Writing to Port 92 is faster method of controlling A20 than sending commands through the Keyboard Microcontroller.
    - »NOTE: Port 92 is the PS/2 Compatibility Port.
  - The A20 Handler Software (e.g. HIMEM.SYS) must be told which I/O Port to use to control A20.
The High Memory Area

Starting with DOS 4.00, the Device Driver HIMEM.SYS was available to access the first 64K bytes (approx.) above the 1 MByte limit.

**FFFF0H** Segment

+ **0FFFFH** Offset

**10FFFEFH** Physical Address on Address Bus.

- This is 65,520 bytes above 1MB (64K - 16 bytes)
- By definition, all memory above 1MB is Extended Memory, which includes the HMA (High Memory Area).
- However, the **HMA** can be accessed in **REAL MODE** by using the Driver HIMEM.SYS.
EXTENDED MEMORY

Beginning with the PC/AT, it is possible to address >1MB of memory by putting the processor in protected mode.

- New PCs typically come with at least 8M bytes installed on the system board.

Since DOS (running in real mode) is not aware of extended memory, what is the extended memory good for?

- Windows uses extended memory.
- Various versions of UNIX use extended memory.
- OS/2 uses extended memory.
- Virtual disks and disk caching use extended memory.
I/O ADDRESS MAP
I/O ADDRESS MAP

- In addition to the memory address, Intel microprocessors have a separate I/O address map.
  - The size of the Intel I/O space is 64K addresses on all the 80X86 processors.

- IBM restricted the I/O address space on its PCs to 1K (400H), dividing it into two regions:
  - 0-511 (000H-1FFH)
    - Reserved for system board peripherals
  - 512-1023 (200H-3FFH)
    - Reserved for adapter cards
I/O ADDRESS MAP

- The system board physically decodes only the lower 10 address lines, so adapter cards that use higher addresses must be careful not to overlap the lower port addresses.

- EXAMPLE: The system board would decode 2000H as 0H, which is assigned to the DMA#1 chip.

  2000H - 0010 0000 0000 0000 in Binary
  A9---------A0

- Most of the I/O assignments on the figure are industry standard.
  - Some Ports such as 78H, 79H, 90H & 92H are Vendor specific & were not in the original industry standard.
I/O ADDRESS MAP (Cont.)

Many devices have several ports assigned to them.

- DEVICE PORTS
- DMA#1 0-0FH
- DMA#2 0C0-0DFH --even numbers only
- PIC#1 20H,21H
- PIC#2 0A0H,0A1H
- PIT 40H-43H
- RT CLOCK 70H-71H
- possibly COM1 and PRINTER
<table>
<thead>
<tr>
<th>Address (hex)</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 - 000F</td>
<td>16 bytes</td>
<td>DMA 1</td>
</tr>
<tr>
<td>0020 - 0021</td>
<td>2 bytes</td>
<td>Interrupt Controller 1</td>
</tr>
<tr>
<td>002E</td>
<td>1 byte</td>
<td>configuration Index Register</td>
</tr>
<tr>
<td>002F</td>
<td>1 byte</td>
<td>Configuration Data Register</td>
</tr>
<tr>
<td>0040 - 0043</td>
<td>4 bytes</td>
<td>Timer 1</td>
</tr>
<tr>
<td>0060</td>
<td>1 byte</td>
<td>Keyboard Controller Data Byte</td>
</tr>
<tr>
<td>0061</td>
<td>1 byte</td>
<td>NMI, speaker control</td>
</tr>
<tr>
<td>0064</td>
<td>1 byte</td>
<td>Kbd Controller, CMD/STAT Byte</td>
</tr>
<tr>
<td>0070, bit 7</td>
<td>1 bit</td>
<td>Enable NMI</td>
</tr>
<tr>
<td>0070, bits 6:0</td>
<td>7 bits</td>
<td>Real Time Clock, Address</td>
</tr>
<tr>
<td>0071</td>
<td>1 byte</td>
<td>Real Time Clock, Data</td>
</tr>
<tr>
<td>0078</td>
<td>1 byte</td>
<td>Reserved Brd. Config.</td>
</tr>
<tr>
<td>0079</td>
<td>1 byte</td>
<td>Reserved Brd. Config.</td>
</tr>
<tr>
<td>0080 - 008F</td>
<td>16 bytes</td>
<td>DMA Page Register</td>
</tr>
<tr>
<td>00A0 - 00A1</td>
<td>2 bytes</td>
<td>Interrupt Controller 2</td>
</tr>
<tr>
<td>00B2 - 00B3</td>
<td>2 bytes</td>
<td>APM control port</td>
</tr>
<tr>
<td>00C0 - 00DE</td>
<td>31 bytes</td>
<td>DMA 2</td>
</tr>
<tr>
<td>00F0</td>
<td>1 byte</td>
<td>Reset Numeric Error</td>
</tr>
<tr>
<td>0170 - 0177</td>
<td>8 bytes</td>
<td>Secondary IDE Channel</td>
</tr>
<tr>
<td>Address (hex)</td>
<td>Size</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>01F0 - 01F7</td>
<td>8 bytes</td>
<td>Primary IDE Channel</td>
</tr>
<tr>
<td>0278 - 027B</td>
<td>4 bytes</td>
<td>Parallel Port 2</td>
</tr>
<tr>
<td>02F8 - 02FF</td>
<td>8 bytes</td>
<td>On - Board Serial Port 2</td>
</tr>
<tr>
<td>0376</td>
<td>1 byte</td>
<td>Sec IDE Chan Cmd Port</td>
</tr>
<tr>
<td>0377</td>
<td>1 byte</td>
<td>Sec IDE Chan Stat Port</td>
</tr>
<tr>
<td>0378 - 037F</td>
<td>8 bytes</td>
<td>Parallel Port 1</td>
</tr>
<tr>
<td>03BC - 03BF</td>
<td>4 bytes</td>
<td>Parallel Port x</td>
</tr>
<tr>
<td>03E8 - 03EF</td>
<td>8 bytes</td>
<td>Serial Port 3</td>
</tr>
<tr>
<td>03F0 - 03F5</td>
<td>6 bytes</td>
<td>Floppy Channel 1</td>
</tr>
<tr>
<td>03F6</td>
<td>1 byte</td>
<td>Pri IDE Chan Cmd Port</td>
</tr>
<tr>
<td>03F7 (write)</td>
<td>1 byte</td>
<td>Floppy Chan 1 Cmd</td>
</tr>
<tr>
<td>03F7, bit 7</td>
<td>1 bit</td>
<td>Floppy Disk Chg Chan 1</td>
</tr>
<tr>
<td>03F7, bits 6:0</td>
<td>7 bits</td>
<td>Pri IDE Chan Status Port</td>
</tr>
<tr>
<td>03F8 - 03FF</td>
<td>8 bytes</td>
<td>On - Board Serial Port 1</td>
</tr>
<tr>
<td>4D0</td>
<td>1 byte</td>
<td>Int 1 edge/level control</td>
</tr>
<tr>
<td>4D1</td>
<td>1 byte</td>
<td>Int 2 edge/level control</td>
</tr>
<tr>
<td>LPT + 400h</td>
<td>8 bytes</td>
<td>ECP Port, LPT + 400</td>
</tr>
<tr>
<td>OCF8 - 0CFB</td>
<td>4 bytes</td>
<td>PCI Config Address Reg</td>
</tr>
<tr>
<td>0CF9</td>
<td>1 byte</td>
<td>Reset Control Register</td>
</tr>
<tr>
<td>0CFC - 0CFF</td>
<td>4 bytes</td>
<td>PCI Config Data Reg</td>
</tr>
<tr>
<td>FF00 - FF07</td>
<td>8 bytes</td>
<td>IDE Bus Master Reg.</td>
</tr>
</tbody>
</table>
SPECIAL I/O PORTS

- PORT 61H on the PC/AT is industry standard.

- PORT 61H functions follow:
  - Enable the speaker gate
  - Enable PIT CT2 gate (for sound to the speaker)
  - Enable and check parity errors

- Some Ports such as 78H, 79H, 90H & 92H are Vendor specific & were not in the original industry standard.
  - Board designers are free to use these ports to implement board specific features.
  - NOTE: These are NOT PC Compatible Ports.
## I/O Addr 61H Bit Assignments (Port B)

<table>
<thead>
<tr>
<th>BIT/VALUE</th>
<th>FUNCTION</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT 7</td>
<td>1 = ONBOARD PARITY ERROR</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>BIT 6</td>
<td>1 = ISA PARITY ERROR</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>BIT 5</td>
<td>1 = SPEAKER SIGNAL ON</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>BIT 4</td>
<td>Toggles with each refresh.</td>
<td>READ ONLY</td>
</tr>
<tr>
<td>BIT 3</td>
<td>1 = ISA PARITY ERROR DISABLED</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 2</td>
<td>1 = ONBOARD MEMORY PARITY ERROR DISABLED</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 1</td>
<td>1 = SPEAKER DATA ON</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 0</td>
<td>1 = SPEAKER ENABLED</td>
<td>READ / WRITE</td>
</tr>
</tbody>
</table>
## Example I/O Addr 78H Bit Assignments

<table>
<thead>
<tr>
<th>BIT/VALUE</th>
<th>FUNCTION</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT 7</td>
<td>1 = Force memory parity error</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 6</td>
<td>1 = Enable cache</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 5</td>
<td>1 = Shadow Video BIOS</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 4</td>
<td>1 = Shadow System BIOS</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 3</td>
<td>1 = Enable ROM Shadow</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 2</td>
<td>1 = Gate A20 enabled</td>
<td>READ / WRITE</td>
</tr>
<tr>
<td>BIT 1</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>BIT 0</td>
<td>1 = Enable Port 90</td>
<td>READ / WRITE</td>
</tr>
</tbody>
</table>
GENERIC

DECODE

LOGIC
GENERIC DECODE LOGIC

- The system board contain some logic to decode the BUS CYCLE DEFINITIONS of the microprocessor.
- The BUS CYCLE DEFINITIONS from the CPU are VALID when ADS# is asserted (Logic 0).
- The drawings show examples of logic that could be used to decode the BUS CYCLE DEFINITIONS.
- The signals generated by the GENERIC DECODE LOGIC would be used by the System Board to generate signals such as I/O chip selects and DRAM & PROM output enables.
**GENERIC DECODE LOGIC**

**e.g.**

I/O WR @ Addr 43H
43H = 0100 0011y
e.g.
I/O WR @ Addr 43H
43H = 0100 0011y
SUMMARY

WE HAVE DISCUSSED THE FOLLOWING:

- The 0-1 megabyte memory map and the functions of each section.
- The organization of PC Base Memory.
- The organization of PC Reserved Memory.
- The High Memory Area & Extended Memory and how they are accessed.
- The organization of the PC’s I/O Address Map.
- Generic Decode Logic for decoding I/O addresses.