**Midterm Mega Quiz File**

**DMA stands for\_\_\_\_\_\_\_\_\_**

**►Direct Memory Access (Page 4)**

►Distinct Memory Access

►Direct Module Access

►Direct Memory Allocation

**Following is not a method of I/O**

►Programmed I/O

► Interrupt driven I/O

**►Hardware Based I/O (Page 4)**

►None of given

**The Function of I/O controller is to provide \_\_\_\_\_\_\_\_\_\_\_\_.**

►I/O control signals

►Buffering

►Error Correction and Detection

**►All of given (Page 5)**

**Standard PC operates in two modes in terms of memory which are**

►Real mode and Extended Mode

►Base mode and Memory Mode

►None of the given

**►Real mode and protected mode (Page 6)**

**There are two main types of interrupts namely \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.**

►PC based and Window based

►Hardware based and Kernal based

**►Hardware interrupts and Software interrupts (Page 10)**

►None of the given

**Interrupt Vector Table (IVT) in short is a \_\_\_\_\_\_\_ bytes sized table.**

**►1024 (Page 10)**

►2048

►3072

►4096

**Maximum number of interrupts in a standard PC is**

► 64

► 128

**► 256 (Page 10)**

► 512

**Which of the following are types of ISR \_\_\_\_\_\_\_\_\_\_.**

►BIOS (Basic I/O service ) ISR

►DOS ISR

►ISR provided by third party device drivers

**►All of the given (Page 13)**

**IVT is a table containing \_\_\_\_\_\_ byte entries each of which is a far address of an interrupt service routine.**

►2

**►4 (Page 20)**

►8

►16

**To set the interrupt vector means is to change the double word sized interrupt vector within the IVT.**

**►True (Page 22)**

►False

**Set the Interrupt vector means to change the double word sized interrupt vector within IVT.**

**►True (Page 22)**

►False

**Each paragraph in keep function is \_\_\_\_ bytes in size.**

►4

►8

**►16 (Page 24)**

►32

**Interrupt service number is usually placed in the \_\_\_\_\_\_\_\_ register.**

►AL

►CL

**►AH (Page 26)**

►AX

**Timer interrupt is a \_\_\_\_\_\_\_\_\_.**

**►Hardware Interrupt (Page 28)**

►Software Interrupt

►Both of these

►None of these

**Timer interrupt occurs \_\_\_\_\_\_\_ times every second by means of hardware.**

**►18.2 (Page 28)**

►16.2

►15.2

►14.2

**Keyboard Status Byte is located at the address**

► 0040:0000H

► 0040:0013H

► 0040:0015H

**► 0040:0017H (Page 29)**

**The output on the monitor is controlled by a controller called \_\_\_\_\_\_\_\_\_\_within the PC.**

**►Video controller (Page 30)**

►Bus controller

►Ram controller

►None of the given

**Display device (Monitor) performs \_\_\_\_\_\_\_\_\_ I/O.**

►**memory mapped (Page 30)**

►Isolated

►Both of above

►None of these

**The keyboard makes use of interrupt number \_\_\_\_\_\_\_ for its input operations.**

**►9 (Page 34)**

►10

►11

►12

**Interrupt 9 usually reads the \_\_\_\_\_\_\_\_\_ from keyboard.**

►ASCII code

**►Scan code (Page 34)**

►Both ASCII and Scan code

►None of the above

**Usually interrupt procedures are reentrant procedures especially those interrupt procedure compiled using C language compiler are reentrant.**

**►True (Page 38)**

►False

**\_\_\_\_\_\_ is Disk interrupt.**

►10H

►11H

**►13H (Page 42)**

►14H

**The service \_\_\_\_\_\_\_\_\_ is called the keyboard hook service.**

►15H/2FH

**►15H/4FH (Page 44)**

►15H/FFH

**NMI Stand for**

**►Non Maskable Interrupt (Page 46)**

►Non Multitude Interrupt

►Non Maskable Instruction

►None of Given

**The microprocessor package has many signals for data. Below are some in Correct priority order (Higher to Lower).**

**►Reset,Hold,NMI,INTR (Page 46)**

►NMI, INTR,Hold,Reset

►INTR,NMI,Reset,Hold

►None of the Given

**A single interrupt controller can arbitrate among \_\_\_\_ different devices.**

►4

►6

**►8 (Page 47)**

►10

**Hardware Interrupts are \_\_\_\_\_\_\_\_\_\_.**

►Preemptive

**►Non-Preemptive (Page 48)**

►Both Preemptive and Non-Preemptive

►None of given

**A software interrupt does not require EOI (End of interrupt).**

**►True (Page 49)**

►False

**In keyboard status byte bit no. 2 and 3 are used for ctrl and alt keys respectively. which of the following condition is used to check that Ctrl + Alt keys are pressed. Where: unsigned char far \* scr = (unsigned char far \*)(0x00400017);**

**►if (((\*scr)&12)==12) (Page 52)**

►if (((\*scr)&8)==8)

►if (((\*scr)&4)==4)

►if (((\*scr)&2)==2)

**To store each character in keyboard buffer \_\_\_\_\_\_\_\_\_\_\_\_\_\_bytes are required**

**►2 (Page 54)**

►4

►6

►8

**\_\_\_\_ No. of bytes are used to store the character in the keyboard buffer.**

► 1

**►2 (Page 54)**

►4

► 8

**Total No. of bytes that can be stored in Keyboard Buffer is\_\_\_\_.**

►16

**►32 (Page 54)**

►64

►128

**If keyboard buffer is empty the head and tail points at the same location.**

**►True (Page 55)**

►False

**Tail of keyboard should get to get the start of buffer.**

**►True (Page 55)**

►False

**Interrupt \_\_\_\_\_ is empty; we can use its vector as a flag.**

►9H

►13H

►15H

**►65H (Page 65)**

**Register can be used to divide frequency is \_\_\_\_\_\_\_\_\_**

**►Counter Register (Page 69)**

►Accumulator Register

►None of these

**Counter register can be used to divide clock signal.**

**►True (Page 69)**

►False

**In counter register bit no. 3 changes its value between 0 and 1 with in \_\_\_\_clock cycles**

►1

►2

►4

**► 16 (Page 69)**

**Command register is an \_\_\_\_\_ bit register**

►4

**►8 (Page 71)**

►16

►32

**The interval timer can operate in \_\_\_\_\_\_\_ modes.**

►Five

►Seven

►Four

**►Six (Page 72)**

**The following command “outportb (0x61,inportb(0x61) & 0xFC);” will**

**►Turn on the speaker (Page 74)**

►Turn off the speaker

►Toggle the speaker

►None of the given

**PPI stands for**

►Parallel Programmable interface

**►Peripheral Programmable interface (Page 76)**

►Port Programmable interface

►None of the given

**PPI is used to perform parallel communication**

**►True (Page 81)**

►False

**The PPI acts as an interface between the CPU and a parallel \_\_\_\_\_\_\_\_ .**

**►I/O device (Page 83)**

►CPU

►BUS

►None of given

**An I/O device cannot be directly connected to the busses so controller is placed between CPU and I/O.**

**►True (Page 83)**

►False

**\_\_\_\_\_\_\_\_\_\_is used to control the printer via the BIOS**

►Int 16H

**►Int 17H (Page 84)**

►Int 18H

►Int 19H

**Standard PC can have \_\_\_\_\_ PPI.**

►1

**►4 (Page 84)**

►8

►16

**BIOS DO NOT support \_\_\_\_\_\_.**

►LPT1

►LPT2

►LPT3

**►LPT4 (Page 91)**

**LPTs can be swapped.**

**►True (Page 92)**

►False

**\_\_\_\_\_\_\_\_ store the base address for LPT1.**

►40:00H

►40:02H

**►40:08H (Page 92)**

►40:1AH

**At IRQ 7 Interrupt # \_\_\_ is used.**

► 0x0A

►0x0B

► 0x0C

**►0x0F (Page 95)**

**If printer is \_\_\_\_\_ then printer sends back the ACK signal to the printer interface**

**►idle (Page 97)**

►busy

►Out of paper

►None of the given

**PPI interconnection \_\_\_\_\_\_\_ bits is cleared to indicate low nibble is being sent.**

►D1

►D2

► D3

**► D4 (Page 101)**

**\_\_\_\_\_ bit is cleared to indicate the low nibble is being sent.**

►D1

►D2

►D3

**►D4 (Page 104)**

**There are \_\_\_\_\_\_\_\_\_\_ kinds of serial communication.**

**►2 (Page 105)**

►3

►4

►5

**In case of synchronous communication a timing signal is required to identify the start and end of a bit.**

**►True (Page 105)**

►False

**In \_\_\_\_\_\_\_\_\_\_\_\_each byte is needed to be encapsulated in start and end.**

►Synchronous communication

**►Asynchronous communication (Page 106)**

►Both

►None of given

**UART stands for\_\_\_\_\_\_\_**

**►Universal Asynchronous Receiver Transmitter (Page 107)**

►Universal Adjustment and Realigning Tool

►Unconventional Assisted Recovery Team

►None of these

**DCE stands for \_\_\_\_\_\_\_\_\_\_.**

**►Data communication equipment (Page 109)**

►Distributed Computing Environment

►Data Communications Equipment

►Data Carrier Equipment

**DTE is \_\_\_\_\_\_\_\_\_\_\_\_.**

**►Data terminal equipment (Page 109)**

►Data transmitting equipment

►Dual terminal equipment

►None of the given.

**DSR stands for \_\_\_\_\_\_\_\_\_\_ .**

**►Data set ready (Page 111)**

►Data service ready

►Data stock ready

►None of the given

**BIOS supports \_\_\_\_\_\_\_\_\_\_\_\_\_UARTS as COM ports.**

►6

**►4 (Page 113)**

►3

►2

**The bit \_\_\_\_\_\_ of Line control register in UART, if cleared will indicate that DLL is the data register.**

►1

►3

►5

**►7 (Page 114)**

**The baud rate is set in accordance with the divisor value loaded within the UART internal registers base +0 and base +1.**

**►TRUE (Page 114)**

►FALSE

**\_\_\_\_\_\_\_\_\_ is used to identify the cause of interrupt.**

**►Interrupt ID Register (Page 116)**

►PC Register

►AC Register

►None of All These

**In self test mode the output of the UART is routed to its input.**

**►True (Page 117)**

►False

**Int 14H \_\_\_\_\_\_\_\_\_\_ can be used to set the line parameter of the UART or COM port.**

**►Service # 0 (Page 119)**

► Service # 1

►Service # 2

►None of the given options

**Int 14H\_\_\_\_\_\_\_\_\_\_\_\_ can be used to send a byte**

►Service#0

**►Service#1 (Page 121)**

►Service#2

►None of the given option

**Int 14H \_\_\_\_\_\_\_\_\_\_ can be used to receive a byte.**

►Service # 0

►Service # 1

**► Service # 2 (Page 121)**

►None of the given options

**The -------- function receive a byte and COM port number is passed as parameter using BIOS service**

►Receivebyte ();

►Receive ();

**►Receivechar (); (Page 125)**

►None of the given option

**The\_\_\_\_\_\_\_\_ function uses the COM port number to receive a byte from the COM port using BIOS services.**

►recievebyte()

►initialize ()

►receive()

►recievechar() (Page 125)

**The \_\_\_\_\_\_\_\_function initialize the COM port whose number is passed as parameter using BIOS services.**

►Initializecom()

**►Initialize() (Page 125)**

►Recievechar()

►None of these option

**The \_\_\_\_\_\_\_\_\_\_\_\_ function simply enables the self test facility within the modem control register**

►STOn()

►SelfTest()

**►SelfTestOn() (Page 127)**

►None of these

**\_\_\_\_whenever received indicates the start of communication and \_\_\_\_\_\_ whenever received indicates a temporary pause in the communication.**

**►XON & XOFF (Page 135)**

►XOFF & XON

►XON & YOFF

►YON & XOFF

**------------ whenever receive indicates the start of communication ........... whenever receive indicates the end of communication**

**►XON\XOFF (Page 135)**

►XOFF\XON

►XON\YOFF

►YON\XOFF

**Software based flow control make use of -------- control characters**

►Xon

►XOFF

**►Both (Page 135)**

►None

**The BIOS interrupt 0x1AH can be used to configure real time clock**

**►True (Page 136)**

►False

**\_\_\_\_\_\_\_\_ is a device incorporated into the PC to update time even if the computer is off.**

►Clock counter

►ROM

►Clock

**►Real time clock (Page 136)**

**The BIOS interrupt \_\_\_\_\_\_\_\_ can be used to configure RTC.**

**►1AH (Page 136)**

►2AH

►3AH

►4AH

**Interrupt \_\_\_\_\_\_ is used to get or set the time.**

►0AH

**►1AH (Page 136)**

►2AH

►3AH

**------------ is used to read time from RTC**

**►1A\02H (Page 137)**

►1A\03H

►1A\04H

►1A\05H

**------------ is used to set time from RTC**

►1A\02H

**►1A\03H (Page 138)**

►1A\04H

►1A\05H

**------------ is used to read date from RTC**

►1A\02H

►1A\03H

**►1A\04H (Page 138)**

►1A\05H

**Only \_\_\_\_\_\_\_\_ ports are important from programming point of view.**

**► 70 and 71H (Page 141)**

►71 and 72H

►70 and 72H

►72 and 73H

**The amount of memory above conventional memory (extended memory) can be determined using the service \_\_\_\_\_\_\_.**

**►15H/88H (Page 162)**

►16H/88H

►17H/88H

►21H/88H

**\_\_\_\_\_\_\_\_\_ used to determine the amount of conventional memory interfaced with the processor in kilobytes.**

►INT 10 H

►INT 11 H

**►INT 12 H (Page 162)**

►INT 13 H

**To distinguish 486 with Pentium CPUID Test is used.**

►True (Page 166)

►False

**If CPUID instruction is not present then the processor can be a**

**► 486 processor (Page 166)**

► 386 processor

► 286 processor

► All of the above

**Bit number \_\_\_\_\_\_\_ of coprocessor control word is the Interrupt Enable Flag.**

**►7 (Page 168)**

►8

►9

►10

**The bit number \_\_\_\_\_\_\_ of the coprocessor control word is the interrupt enable flag.**

**►7 (Page 168)**

►8

►9

►6

**Keyboard uses port \_\_\_\_ as status port.**

**►64H (Page 177)**

►66H

►67H

►69H

**Which port is known as Data Port\_\_\_\_\_\_**

**►60H (Page 177)**

►61H

►64H

►69H

**The keyboard input character scan code is received at \_\_\_ port.**

**►60H (Page 179)**

►61H

►62H

►63H

**\_\_\_\_\_\_\_\_means typematic rate will be sent in next byte.**

**►0xF3 (Page 180)**

►0xF4

►0xF5

►0xF6

**\_\_\_\_\_\_\_ is LED control byte.**

►0xFD

**►0xED (Page 181)**

►0xFF

►0xEE

**By cascading two DMAs \_\_\_\_ bits can be transferred.**

►4

►8

**►16 (Page 186)**

►32