

Total No. of Questions—8]

[Total No. of Printed Pages—2

Seat No.	
-------------	--

[5352]-569

S.E. (Computer) (II Sem.) EXAMINATION, 2018

MICROPROCESSOR

(2015 COURSE)

Time : Two Hours

Maximum Marks : 50

- N.B. — (i) Answer Q. No. 1 or Q. No. 2, Q. No. 3 or Q. No. 4, Q. No. 5 or Q. No. 6, Q. No. 7 or Q. No. 8.
(ii) Neat diagram must be drawn whenever necessary.
(iii) Figures to the right indicate full marks.
(iv) Assume suitable data, if necessary.

1. (a) Explain immediate and register addressing mode with an example. [2]
(b) Draw and explain the flag register of 80386. [4]
(c) Draw and explain segment descriptor. [6]

Or

2. (a) What is the use of Interrupt Flag ? [2]
(b) Explain paging mechanism. [4]
(c) Draw and explain the 80386 address translation mechanism considering PG bit in CR0 is set. [6]

3. (a) What is CPL and RPL ? [2]
(b) Explain Interrupt no. 0 and 4. [4]
(c) Explain the role of Task Register in multitasking and the instructions used to modify and read TR. [6]

P.T.O.

Or

4. (a) List five aspects of protection in the 80386. [2]
(b) Write a short note on 'I/O permission Bit Map'. [3]
(c) Draw and explain TSS. [7]

5. (a) Write short note on Virtual 8086 Mode. [3]
(b) Explain software initializations required for protected mode. [4]
(c) Draw and explain structure of the TLB. [6]

Or

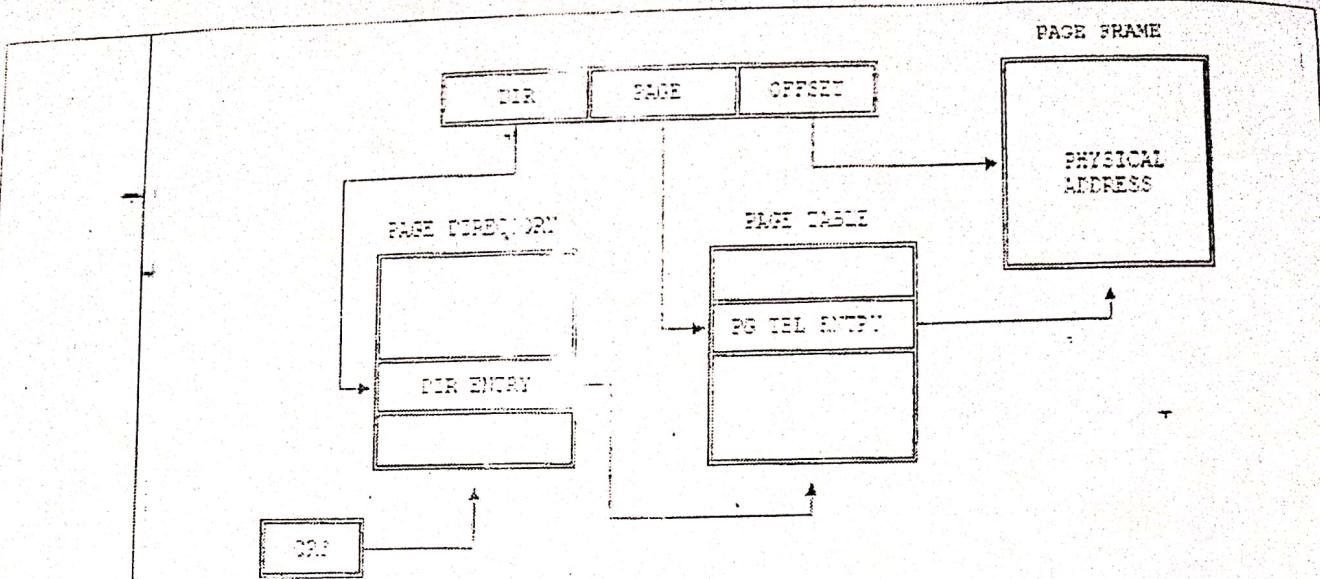
6. (a) What are the contents of various registers of processor 80386 after reset ? [3]
(b) Explain entering and leaving V86 mode. [4]
(c) Draw and explain debug registers of the 80386. [6]

7. (a) Explain the following signals : [3]
 (i) W/R#
 (ii) D/C#
 (iii) M/IO#
(b) Explain any four 80387 constant instructions. [4]
(c) Draw read cycle with non-pipelined address timing. [6]

Or

8. (a) Explain the following signals : [3]
 (i) INTR#
 (ii) NMI#
 (iii) RESET#
(b) Draw and explain 80387 register stack [4]
(c) Explain any six 80387 data transfer instructions. [6]

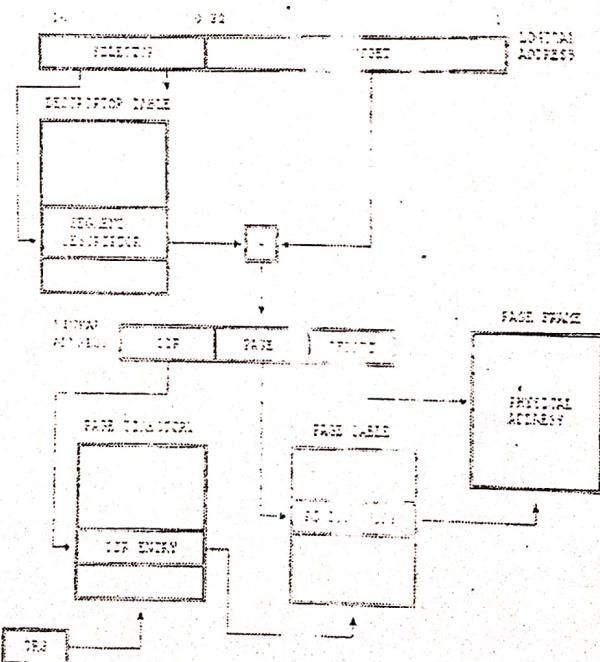
Q1)	a)	Explain immediate and register addressing mode with an example.	[02]
Answer		<p>(i) Immediate:</p> <p>Certain instructions use data from the instruction itself as one of the operands. Such an operand is called an immediate operand. The operand may be 32-, 16-, or 8-bits long.</p> <p>For example: MOV EAX,0000005h</p> <p>(ii) Register: Operands may be located in one of the 32-bit/ 16-bit/8-bit general registers. Eg: MOV ECX,EDX</p>	
Q1)	b)	Draw and explain the flag register of 80386.	[04]
Answer		Diagram 2 and description 2 marks	
		<p>VIRTUAL MODE RESUME FLAG NESTED TASK FLAG I/O PRIVILEGE LEVEL INTERRUPT ENABLE</p>	
Q1)	c)	Draw and explain segment descriptor	[06]
Answer		<p>SEGMENT BASE is... SEGMENT LIMIT is...</p>	
		Explanation 3 marks , diagram 3 marks	
Q2)	a)	What is the use of Interrupt Flag?	[02]
Answer		The IF flag, when set, allows recognition of external interrupts signalled on the INTR pin. When IF is reset, external interrupts signalled on the INTR are not recognized. IOPL indicates the maximum CPL value allowing alteration of the IF bit when new values are popped into EFLAGS or FLAGS.	
Q2)	b)	Explain paging mechanism.	[04]
Answer			



Explanation of conversion process is expected, diagram 2 marks, explanation: 2 marks

Q2)	c)	Draw and explain the 80386 address translation mechanism considering PG bit in CR0 is set.	[06]
-----	----	--	------

Auswer Diagram 3 marks and explanation 3



Q3)	a)	What is CPL and RPL?
-----	----	----------------------

CPL: An internal processor register records the current privilege level (CPL). Normally the CPL is equal to the DPL of the segment that the processor is currently executing. CPL changes as control is transferred to segments with differing DPLs.

RPL: Selectors contain a field called the requestor's privilege level (RPL). The RPL is intended to represent the privilege level of the procedure that originates a selector.

b)	Explain Interrupt no. 0 and 4.
----	--------------------------------

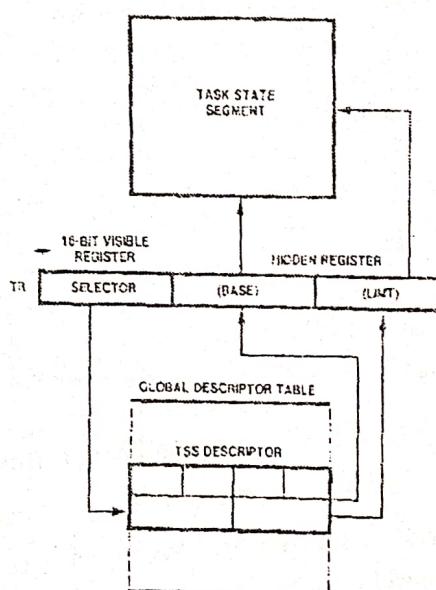
Auswer 2 mark each

[04]

Q3)	c)	Explain the role of Task Register in multitasking and the instructions used to modify and read TR. [06]
-----	----	---

Answer 2 marks and brief explanation 4 marks

The task register (TR) identifies the currently executing task by pointing to the TSS. The task register has both a "visible" portion (i.e., can be read and changed by instructions) and an "invisible" portion (maintained by the processor to correspond to the visible portion; cannot be read by any instruction). The selector in the visible portion selects a TSS descriptor in the GDT. The processor uses the invisible portion to cache the base and limit values from the TSS descriptor. Holding the base and limit in a register makes execution of the task more efficient, because the processor does not need to repeatedly fetch these values from memory when it references the TSS of the current task.



The instructions L TR and STR are used to modify and read the visible portion of the task register. Both instructions take one operand, a 16-bit selector located in memory or in a general register.

LTR (Load task register) loads the visible portion of the task register with the selector operand, which must select a TSS descriptor in the GDT. LTR also loads the invisible portion with information from the TSS descriptor selected by the operand. LTR is a privileged instruction; it may be executed only when CPL is zero. LTR is generally used during system initialization to give an initial value to the task register; thereafter, the contents of TR are changed by task switch operations.

STR (Store task register) stores the visible portion of the task register in a general register or memory word. STR is not privileged.

OR

Q3)	a)	List five aspects of protection in the 80386. [02]
-----	----	--

Answer

1. Type checking
2. Limit checking
3. Restriction of addressable domain
4. Restriction of procedure entry points
5. Restriction of instruction set

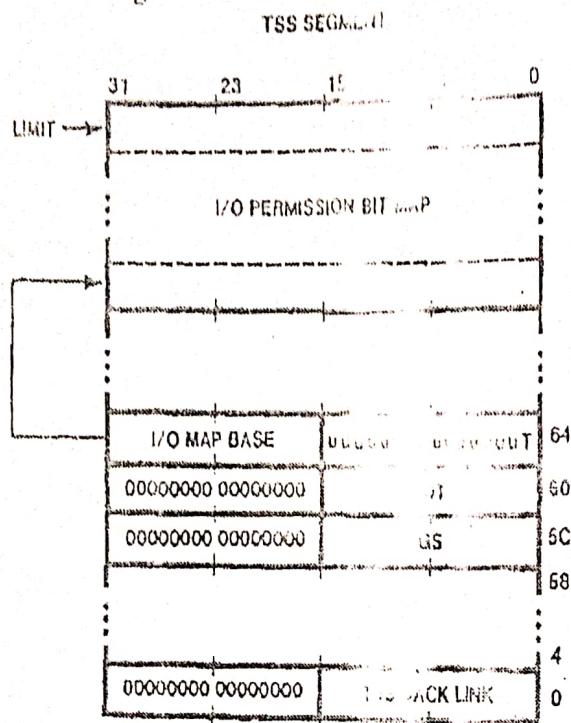
(Q) b)

Write a short note on 'I/O Permission Bit Map'.

[03]

Answer

The I/O instructions that directly refer to addresses in the processor's I/O space are IN, INS, OUT, OUTS. The 80386 has the ability to selectively trap references to specific I/O addresses. The structure that enables selective trapping is the *I/O Permission Bit Map* in the TSS segment.



The I/O permission map is a bit vector. The size of the map and its location in the TSS segment are variable. The processor locates the I/O permission map by means of the I/O map base field in the fixed portion of the TSS. The I/O map base field is 16 bits wide and contains the offset of the beginning of the I/O permission map. The upper limit of the I/O permission map is the same as the limit of the TSS segment.

(Q) c)

Draw and explain TSS.

[07]

Answer

Diagram 4 marks and explanation 3 marks

Q5)	a)	Write short note on Virtual 8086 Mode.	[03]
Answer		<p>The 80386 supports execution of one or more 8086, 8088, 80186, or 80188 programs in an 80386 protected-mode environment. An 8086 program runs in this environment as part of a V86 (virtual 8086) task. V86 tasks take advantage of the hardware support of multitasking offered by the protected mode.</p> <p>The purpose of a V86 task is to form a "virtual machine" with which to execute an 8086 program. A complete virtual machine consists not only of 80386 hardware but also of systems software.</p> <p>Thus, the emulation of an 8086 is the result of cooperation between hardware and software:</p> <ul style="list-style-type: none"> • The hardware provides a virtual set of registers (via the TSS), a virtual memory space (the first megabyte of the linear address space of the task), and directly executes all instructions that deal with these registers and with this address space. • The software controls the external interfaces of the virtual machine (I/O, interrupts, and exceptions) in a manner consistent with the larger environment in which it executes. In the case of I/O, software can choose either to emulate I/O instructions or to let the hardware execute them directly without software intervention. <p>Software that helps implement virtual 8086 machines is called a <i>V86 monitor</i>.</p>	
Q5)	b)	Explain software initializations required for protected mode.	[04]
Answer		Most of the initialization needed for protected mode can be done either before or after	

switching to protected mode. Once in protected mode, however, the initialization procedures must not use protected-mode features that are not yet initialized.

- 1) IDT
- 2) Stack
- 3) Global descriptor table
- 4) Page tables
- 5) First task

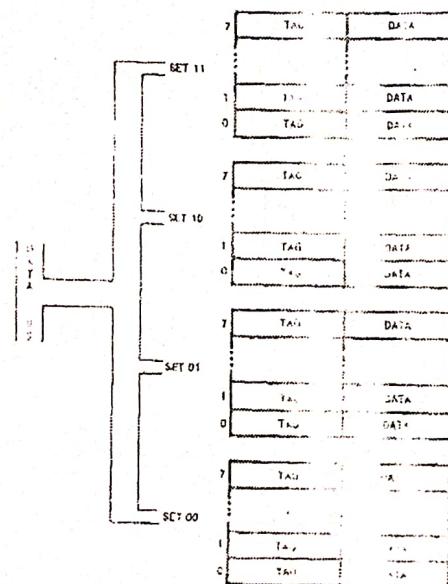
(Q5) c)

Draw and explain structure of the TLB.

[06]

Answer

Diagram 3 marks and explanation 3 marks



OR

(a) a)

What are the contents of various registers of processor 80386 after reset?

[03]

Answer

The contents of EAX depends on the result of the power-up self-test.

DX holds component identifier and revision number.

CR0 all bits are zero except bit 1 (base status depends on presence of 80387 NDP).

EFLAGS = 0000000H

IP = 000010F0H

CS selector = 0 01

DS selector = 0 00H

ES selector = 0 00H

SS selector = 0 00H

FS selector = 0 00H

GS selector = 0 00H

IDTR:

base = 0

limit = 0FFFFH

b) Explain entering and leaving protected mode.

[04]

Answer

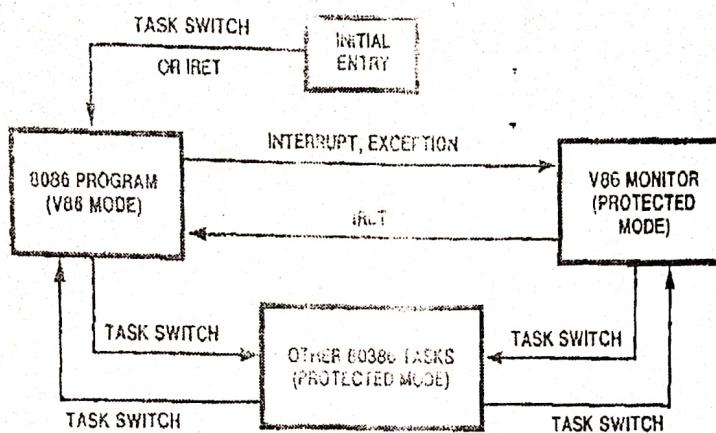
Real-address mode is in effect until a signal on the RESET pin. Even if the system is going to be used in protected mode, the start-up program will execute in real-address mode temporarily while it enters protected mode.

Switching to Protected Mode.

The only way to leave real-address mode is to switch to protected mode. The processor enters protected mode when a MOV to CRO instruction sets the PE (protection enable) bit in CRO.

(For compatibility with the 80286, the LMSW instruction may also be used to set the PE bit.)

MOOL TRANSITION DIAGRAM



(Q6) **Q6) $\neg c$** Draw and explain debug registers of the 80386.

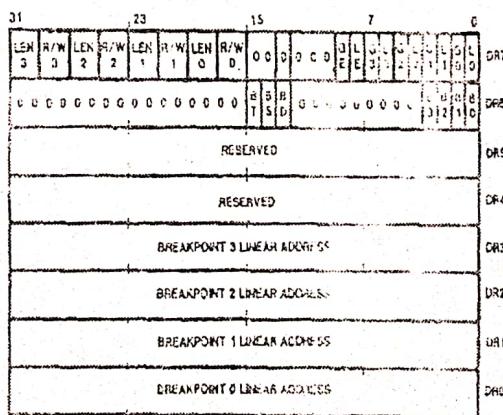
[06]

Answer

Diagram 3 marks, explanation 1 mark.

Total 8 debug registers are present out of them 2 are reserved and six are used by 80386 to control debug feature.

List: DR0-DR7



Q7)	<p>a) Explain following signals</p> <ul style="list-style-type: none"> i. W/R## ii. D/C# iii. RST# 	[03]
-----	--	------

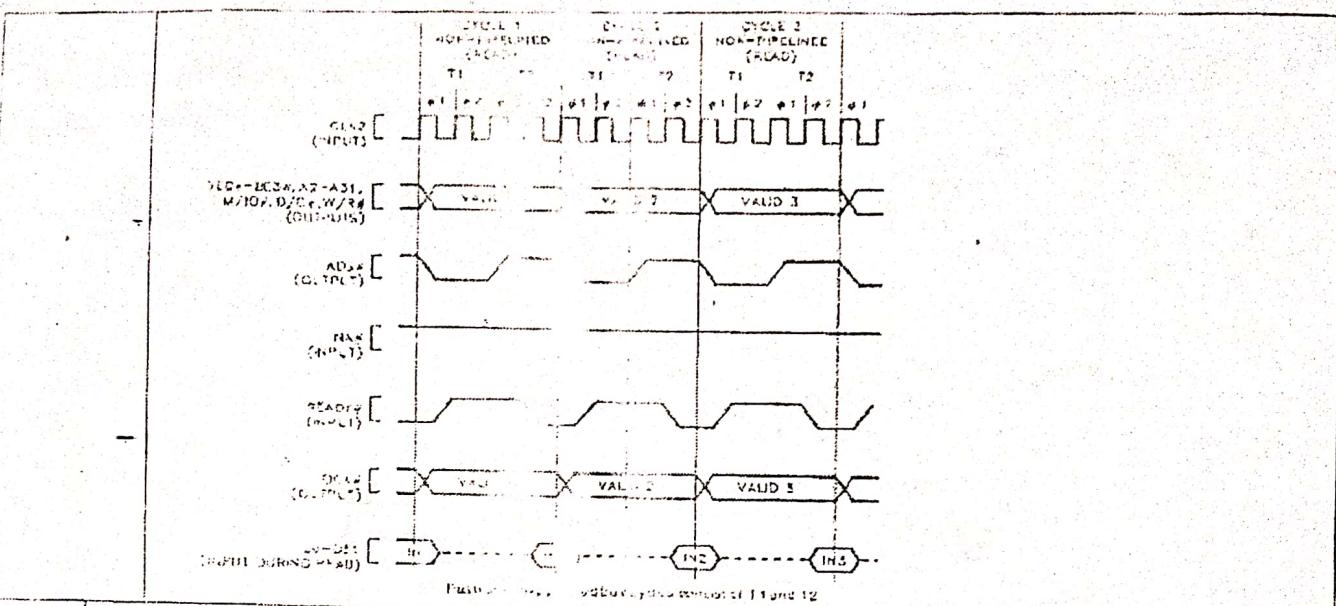
Answer 1 mark each

Q7) b) Explain any four 80387 constant instructions. [04]

Answer 1 mark each

(7) e) Draw 'read cycle with non-pipelined address timing'. [06]

Answer [10]



OR

- Q4) a) Explain following signals
 i. INT#
 ii. NMI#
 iii. RESET#

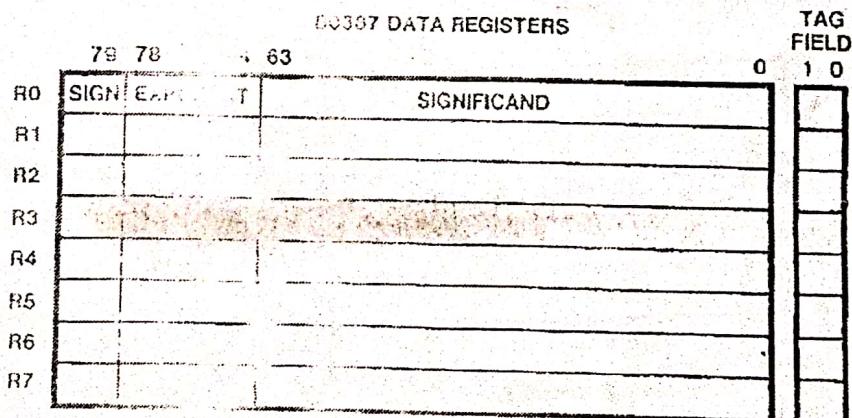
[03]

Answer 1 mark each

[04]

- b) Draw and explain 80387 floating point stack.

Answer Diagram 3 marks, explanation 6 marks



- c) Explain any six 80387 data transfer instructions.

[06]

Answer 1 mark each