CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test –III

Sub:	MICROPROCESSOR							Code:	10EC62
Date:	30 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE/TCE

Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1	With a neat block diagram, explain the maximum mode operation of 8086.	[10]	CO2	L2
2	Draw a timing diagram and explain to execute a memory read & write operation in maximum mode of 8086 processor.	[10]	CO5	L2
3	Explain NRZI encoding along with flowchart used to generate the USB data.	[10]	CO5	L3
4 (a)	Represent 20.59375_{10} into short real form, 178.625_{10} using 80 bit temporary real from and -29.563_{10} using long real format. Use hex format for expressing the answer.	[06]	CO2	L2
(b)	Write a Program using 8086 instruction to check whether PCI bus extension is available using BIOS?	[04]	CO2	L2
5	Explain data types of 8087 co-processor.	[10]	CO5	L3
6	Write a program to find the area of a circle using arithmetic co-processor instruction set.	[10]	CO2	L3
7	Explain data transfer instructions of 8087 NDP.	[10]	CO2	L3

CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test –III

Sub:	MICROPROCESSOR							Code:	10EC62
Date:	30 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE/TCE

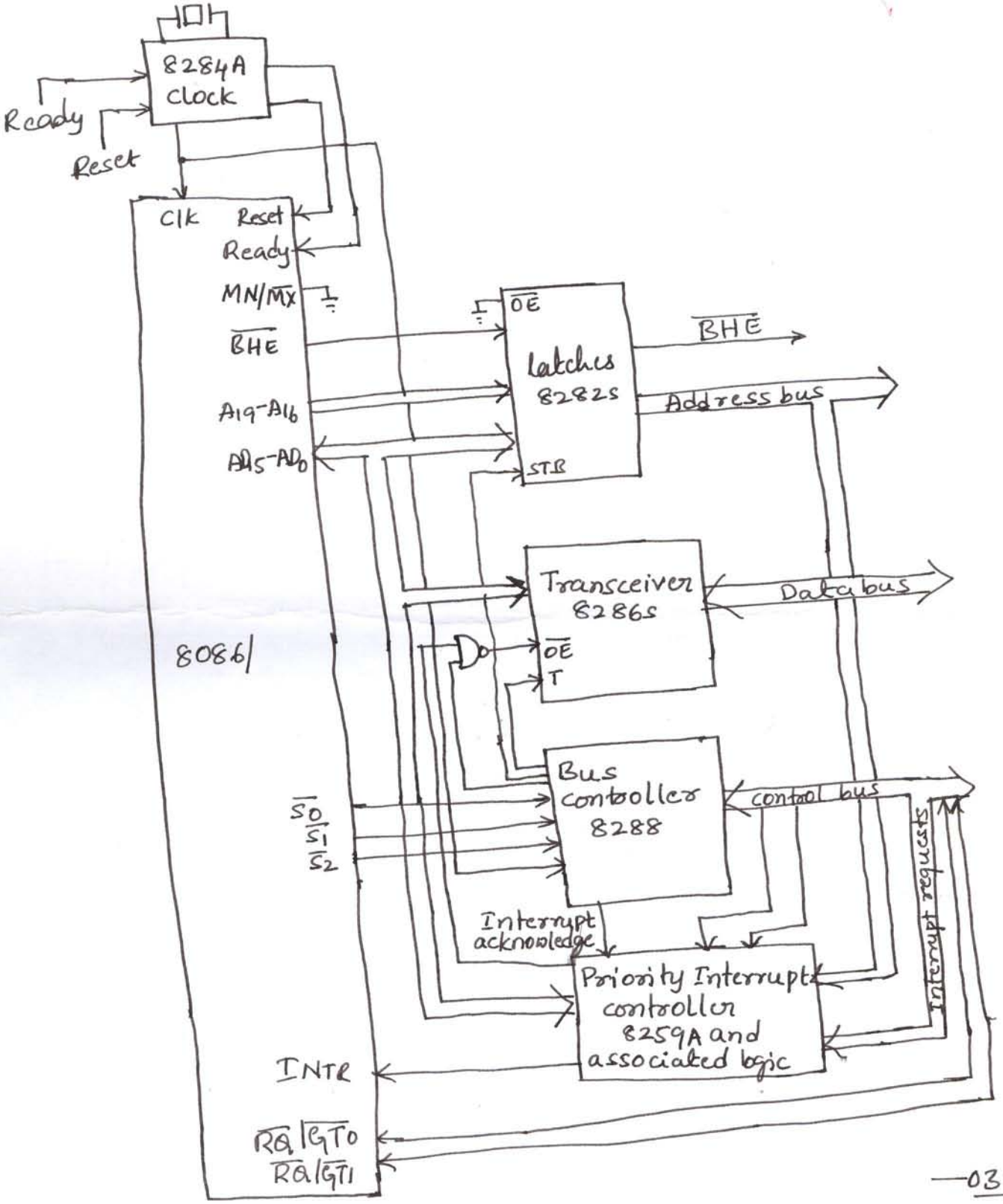
Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1	With a neat block diagram, explain the maximum mode operation of 8086.	[10]	CO2	L2
2	Draw a timing diagram and explain to execute a memory read & write operation in maximum mode of 8086 processor.	[10]	CO5	L2
3	Explain NRZI encoding along with flowchart used to generate the USB data.	[10]	CO5	L3
4 (a)	Represent 20.59375_{10} into short real form, 178.625_{10} using 80 bit temporary real from and -29.563_{10} using long real format. Use hex format for expressing the answer.	[06]	CO2	L2
(b)	Write a Program using 8086 instruction to check whether PCI bus extension is available using BIOS?	[04]	CO2	L2
5	Explain data types of 8087 co-processor.	[10]	CO5	L3
6	Write a program to find the area of a circle using arithmetic co-processor instruction set.	[10]	CO2	L3
7	Explain data transfer instructions of 8087 NDP.	[10]	CO2	L3

① With a neat block diagram, explain the maximum mode operation of 8086.

10 marks

→



—03 marks

Block diagram Maximum mode configuration

Maximum mode is for medium size to large systems, includes two or more processor.

Additional circuitry needed to translate control signals.

$\overline{S_0}$	$\overline{S_1}$	$\overline{S_2}$	Description	control signal generation
0	0	0	Interrupt acknowledge.	$\overline{INTA}$
0	0	1	Read I/o port	$\overline{IORC}$
0	1	0	write I/o port	$\overline{IOWC}$
0	1	1	Halt	-
1	0	0	Instruction fetch	-
1	0	1	Read memory	$\overline{MRDC}$
1	1	0	write memory	$\overline{MWTC}$
1	1	1	Inactive - passive.	-

$QS1, QS0$ → status of the instruction queue.

$\overline{LOCK}$ → Indicates bus is not to be relinquished to other potential masters.

$\overline{RQ}/\overline{GT0}$ → Inputting bus requests and outputting bus grant. ~~that high priority~~.

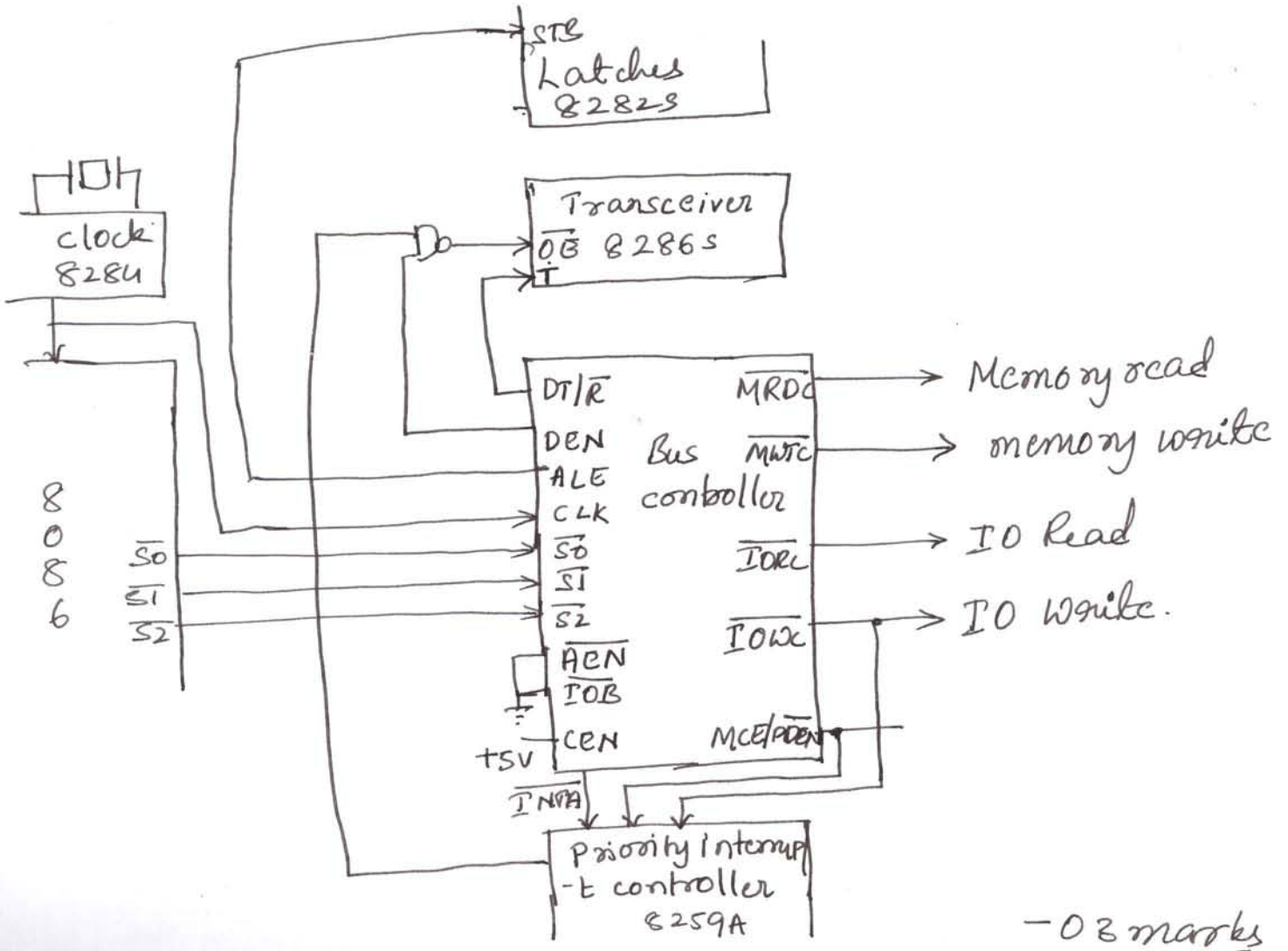
$\overline{RQ}/\overline{GT0}$ → Same as $\overline{RQ}/\overline{GT1}$, ~~but~~ but $\overline{RQ}/\overline{GT0}$ has high priority.

HOLD and $\overline{HLDA}$ pins become the $\overline{RQ}/\overline{GT0}$ and $\overline{RQ}/\overline{GT1}$

Bus request consist of negative pulse arriving before the the start of the current bus cycle. Grant is negative pulse is sued at the beginning of the current bus cycle. provided previous bus transfer was not low byte of a word or interrupt acknowledgement did not occur or instruction with a lock prefix is not executed.

The following diagram shows the derivation of different control signals from status signals $\overline{S0}, \overline{S1}, \overline{S2}$.

– 04 marks

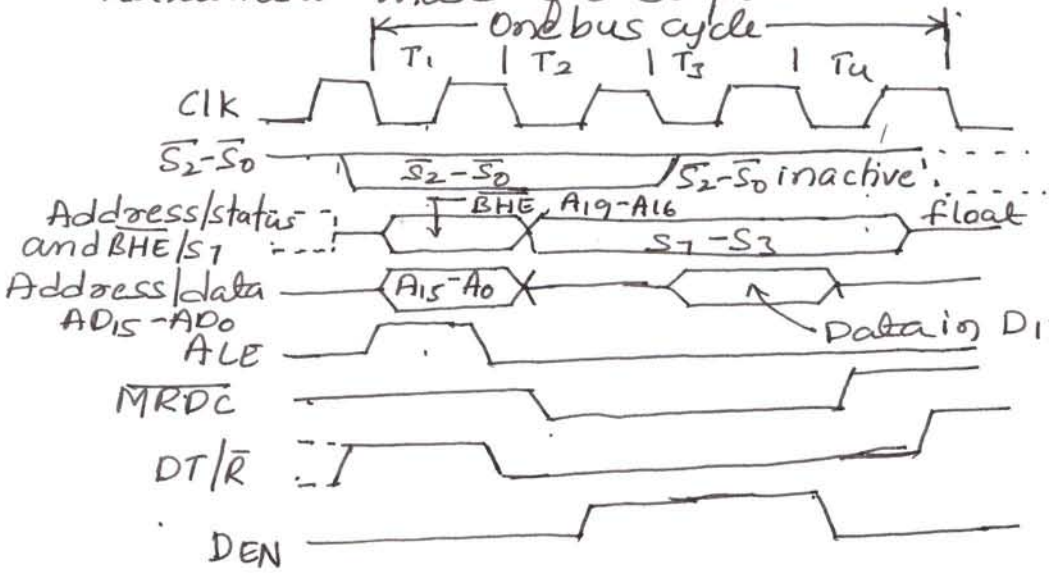


-03 marks

---

② Draw a timing diagram and explain to execute a memory read and write operation in maximum mode of 8086 processor. -10 marks

→ Timing diagram for memory read operation in maximum mode of 8086 processor.

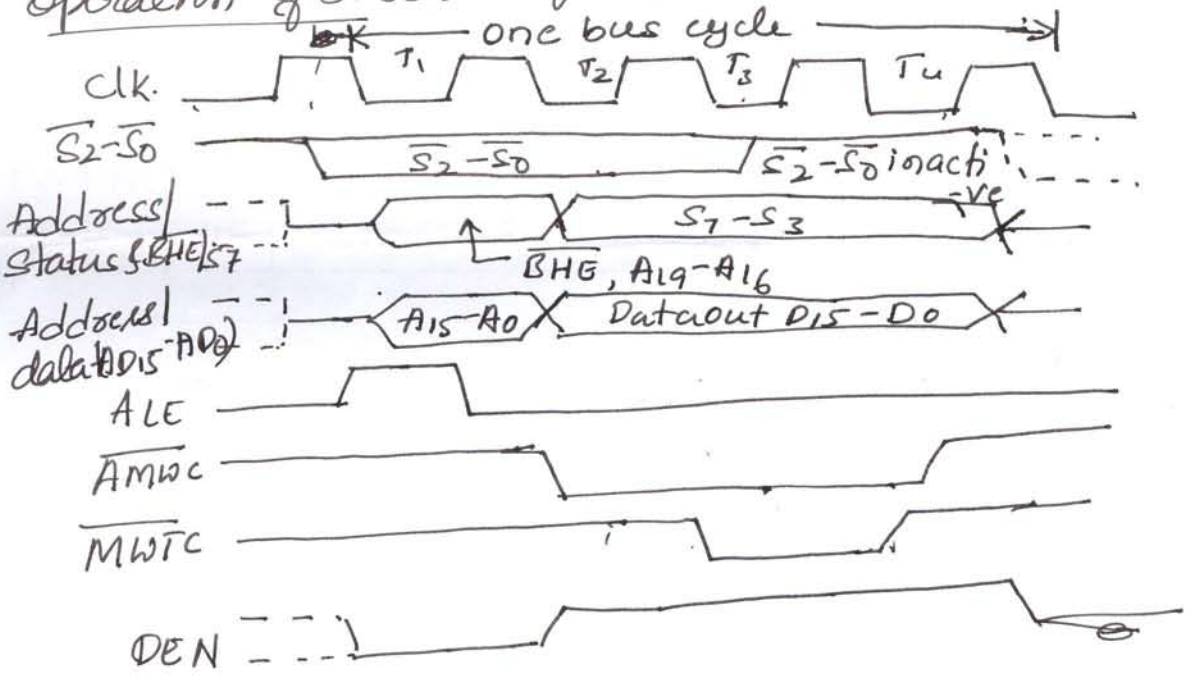


$\overline{S0}, \overline{S1}, \overline{S2}$ – Set prior to the beginning of bus cycle. Upon detecting a change from passive state, 8288 bus controller will output a pulse on its ALE pin.

$DT/\overline{R}=0$ during T_1, $DEN=1$ during T_2 enables transceiver. Activates $\overline{MRDC}$ by placing '0' and maintained till T_4. When ALE is high address is latched after that $A_{19}-A_{16}$ carries status signals. once ALE is dropped to zero waits for the external device to send the data. once data is received by 8086, $DT/\overline{R}$ goes to logic '1', & DEN drops to zero, $\overline{MRDC}$ rises again.

– 05 marks

Memory write timing diagram in maximum mode operation of 8086:

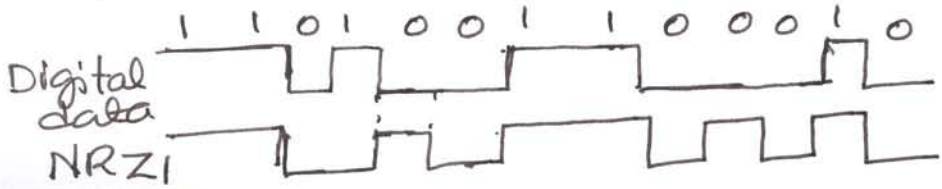


$\overline{S2}, \overline{S1}, \overline{S0}$ Set prior to bus cycle, upon detecting a change from passive state. ALE is set before T_1, once address is latched goes to low. $DT/\overline{R}$ is set to '1'. DEN also set to 1 during T_2 in 8288 $\overline{MWTC}$ is activated from T_3 to T_u and $\overline{AMWC}$ is activated from T_2 to T_u. Once data is received by external device, all the signals are deactivated.

– 05 marks

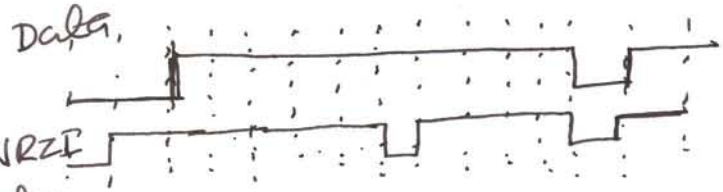
③ Explain NRZI encoding along with flowchart used to generate the USB data. - 10 marks

→ NRZI stands for Non-return to zero Inverted, USB uses NRZI data encoding for transmitting packets. This method doesnot change the signal level for the transmission of a logic '1'; but the signal level is inverted for each change to a logic '0'. The following waveform shows the NRZI encoding scheme. The top waveform shows the data to be transmitted, the second waveform shows the NRZI encoded data

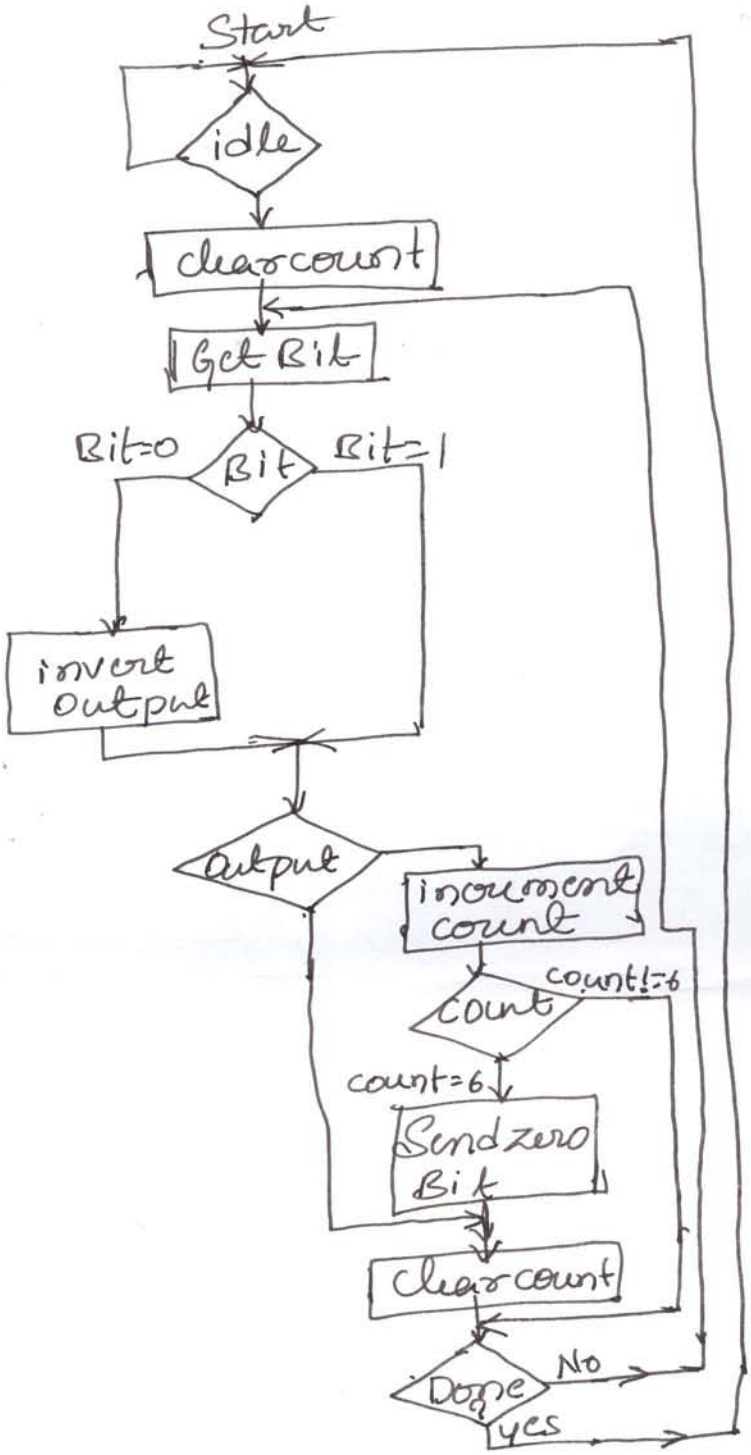


— 05 marks

The actual data transmitted includes sync bits using a method called bit stuffing. If a logic '1' is transmitted for more than 6 bits in a row, the bit stuffing technique adds an extra bit after six continuous 1's in a row. Bit stuffing ensures that the receiver can maintain synchronization for long string of 1's. Data are always transmitted beginning with the least significant bit first, followed by subsequent bits.



Data stream after bit stuffing



Flow chart used to generate USB data

4a) Represent 20.5935_{10} in short real, 8.625_{10} using 80-bit temporary real and -29.563_{10} using long real format. Use hex format for expressing the answer. — 06 marks

(i) 20.5935_{10}

$(20)_{10} \rightarrow (10100)_2$

$(0.5935)_{10} \rightarrow (1001011\ldots)_2$.

$(20.5935)_{10} \rightarrow (10100.1001011\ldots)_2$.

$1.F \times 2^{Exp}$ format $\rightarrow 1.01001001011\ldots \times 2^{4}$.

Normalization:

Bias exponent for short real $= 2^7 - 1 + Exp$

$= 127 + 4 = (131)_{10} = (10000011)_2$

∴ In short real format.

S	Exponent	Fraction
0	10000011	01001001011 - - - - - - -

31 30 23 22 0

Since it is positive real number

In Hex form: $(41A4B\ldots)_{16}$. — 02 marks

(ii) $(178.625)_{10}$: $(178)_{10} \rightarrow (10110010)_2$, $(0.625)_{10} \rightarrow (0.10100\ldots)_2$

$(178.625)_{10} \rightarrow (10110010.101000\ldots)_2$.

$= 1.0110010101000\ldots \times 2^{7}$

Normalization: Biased exponent for temporary real $= 2^{16} - 1 + EXP$

$= 16384 - 1 + 7 = 16390_{10}$

$(10000000000110)_2$.

In temporary real format

positive real number →	0	10000000000110	0110010101000- - -

63 62 52 51 0

In hexa: $(4006655\ldots)_H$. — 2 marks

(iii) $(-29.563)_{10} \rightarrow$

$(29)_{10} \rightarrow (11101)_2$.

$(0.563)_{10} \rightarrow (1001000000\,10\text{---})_2$.

$(29.563)_{10} \rightarrow (11101.1001000000010\ldots)_2$.

$= 1.1101100100000010\ldots \times 2^4$

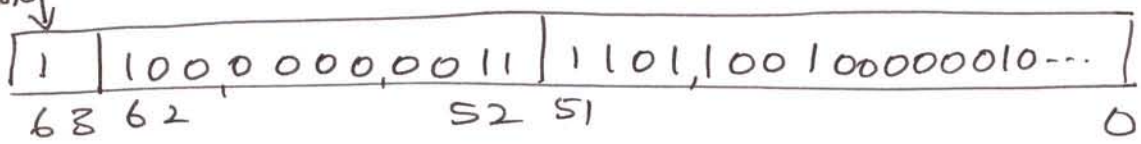
Biased exponent for long real $= 2^{10} - 1 + \text{Exp}$

$= 1024 - 1 + 4$.

$= (1027)_{10}$

$= (10000000011)_2$.

Since negative number



In Hex $(C\,0\cdot3\,D\,9\,0\,2\ldots)_H$ — 2 marks from

4(b)

(5) Explain data types of 8087 co proce & 8086?

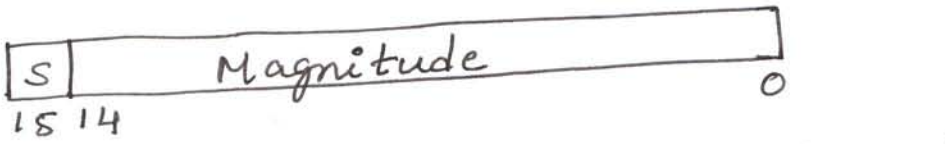
→ Data types supported by 8087 are.

(a) Word integer:

Size of the data type: 2 bytes.

Range of values -32768 to 32767

format:



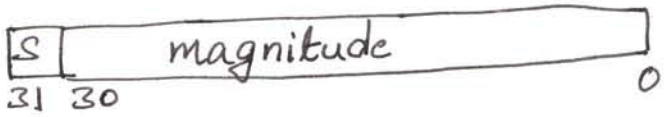
Negative integers are stored in 2's complement.

(b) Short integer:

Size of the data type: 4 bytes.

Range of values: -2×10^9 to $2 \times 10^9 - 1$

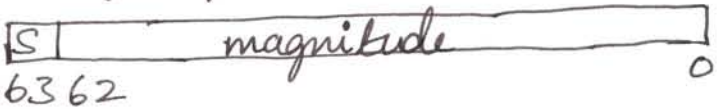
format:



(c) Long Integer:

Size of the data type: 8 bytes

Range of values: -9×10^{18} to 9×10^{18}



All the above integer types store negative integer in twos complement form.

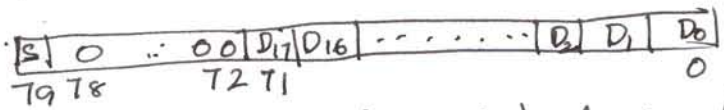
-04 marks

(d) Packed BCD:

Length of the data type : 10 bytes.

Range of values :- -10^{18} to $-10^{0}-1$

format:



Entire most significant byte is dedicated to sign. Most significant bit of this byte dedicated to sign rest of the bits are assigned with '0'.
Remaining 9-bytes represent the magnitude with two BCD digits packed into each byte.

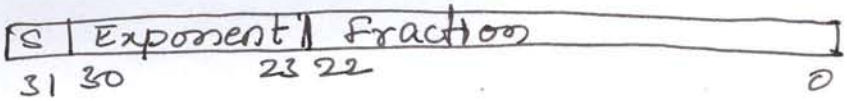
— 02 marks

(e) Short real :

Length of the data type: 4 bytes

Range of values :- $\pm 1 \times 10^{-38}$ to $\pm 1 \times 10^{38}$

format:

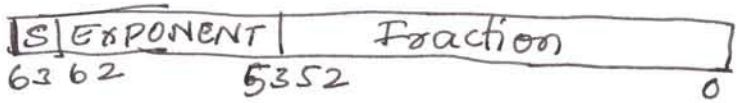


(f) Long real:

Length of the data type : 8 bytes

Range of values: $\pm 10^{-308}$ to $\pm 10^{308}$

format:

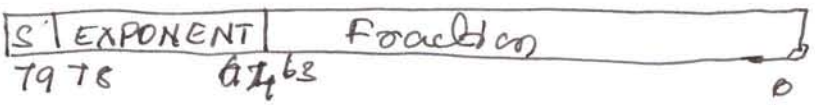


(g) Temporary real:

Length of the data type : 10 bytes

Range of values: $\pm 10^{-4932}$ to $\pm 10^{4932}$

format:



04 marks

⑥ Write a program to find the area of a circle using arithmetic co-processor instruction set. – 10 marks

→
```
.model small
.data
mes1 db "PCI BUS IS PRESENT$"
MCS2 db "PCI BUS IS NOT PRESENT$"
.code
~~MOV AX,~~ '
.startup
  mov AH,0B1H
   mov AL,1
   INT 1AH
   mov DX,offset mcs2
   .If carry?
     mov DX, offset mes1
   .endif
   mov AH, 9
    INT21H
    .Exit.
```
– 10 Marks

⑦ Explain the data transfer instructions of 8087 NDP. 10 marks

The data transfer instructions of 8087 NDP are

(i) FLD SRC – Load real
SRC can be ST(i) or memory operand of short real, long real or temporary real type
Decrement ST, convert the contents of SRC to temporary real and put the result in (ST).
– 1 Mark

(ii) FILD SRC – Load integer.
SRC:- a word integer, long integer, short integer memory operand.
Decrement ST, convert the contents of SRC to temporary real & put the result in (ST).
1 mark

(iii) FBLD SRC – Load BCD

SRC: packed decimal memory operand. 1-mark

(iv) FST DST – Store real

DST: ST(i) or memory operand of long or short real type. 1 mark

(v) FIST DST – Store integer.

DST: word integer or short integer memory operand. 1 mark

(vi) FBSTP DST:- Store BCD and pop.

DST: packed decimal memory operand. 1 mark

(vii) FSTP DST: Store real and pop.

DST: ST(i) or memory operand of short real, or long real or temporary real type. 1 mark

(viii) FISTP DST: Store integer and pop

DST: word integer, short integer or long integer memory operand 1 mark

(ix) FXCH // DST

DST: ST(i), if not specified, ST(1) is assumed. Inter change the contents of the destination register with (ST). 1 mark

For load instructions ST will be decremented by 1 and source contents are converted to temporary real and put the result in (ST).

For store instructions (ST) contents will be converted to destination real format or integers or packed decimal format.

For store and pop instructions after storing in destination ST is incremented. 1 mark