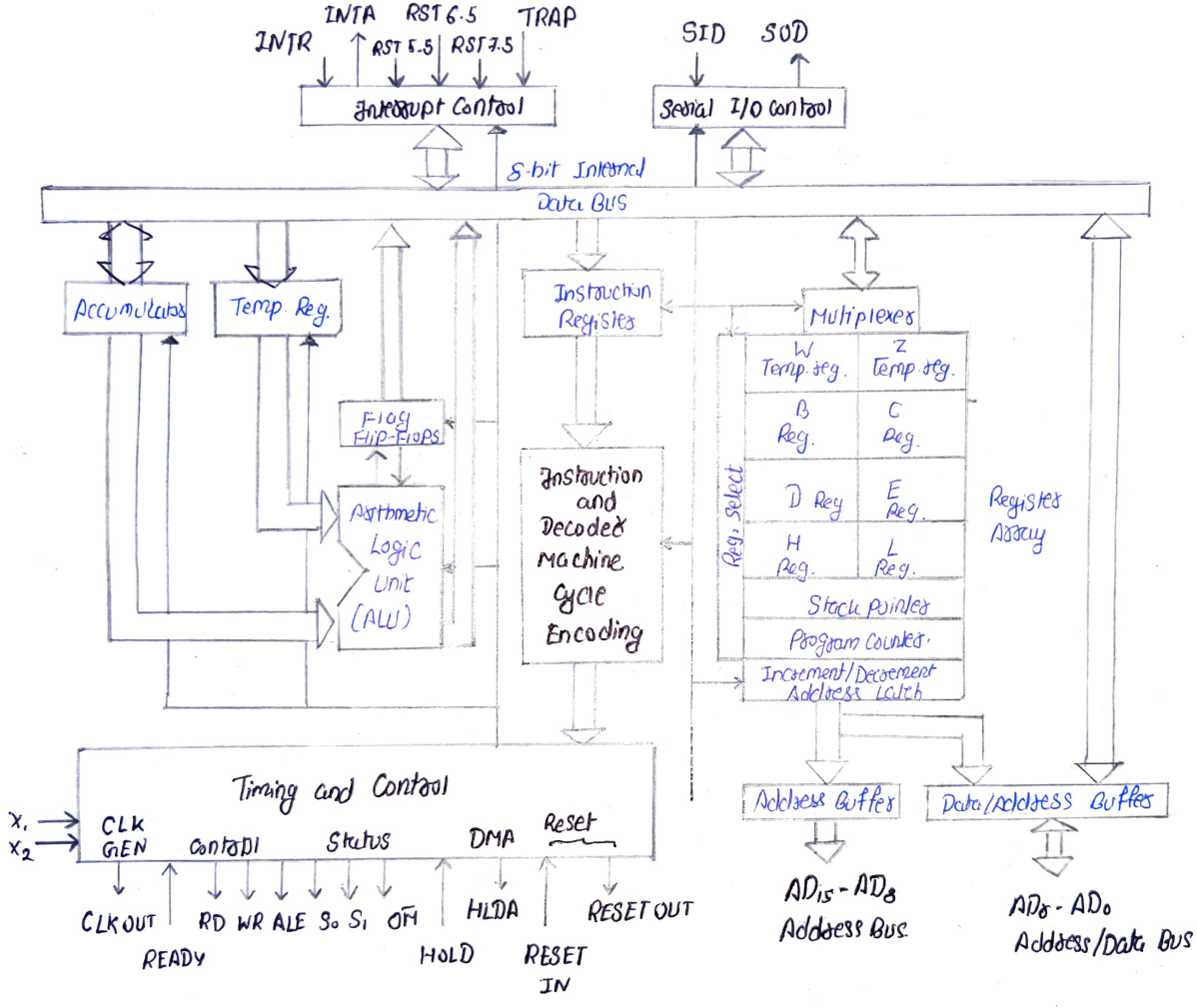
Name: Kumawat Ashish Ramniwas
Roll No: 1906055
Course: Microprocessor and Microarchitecture
Course Code: CS5469
Date: 20/12/21
Page No:

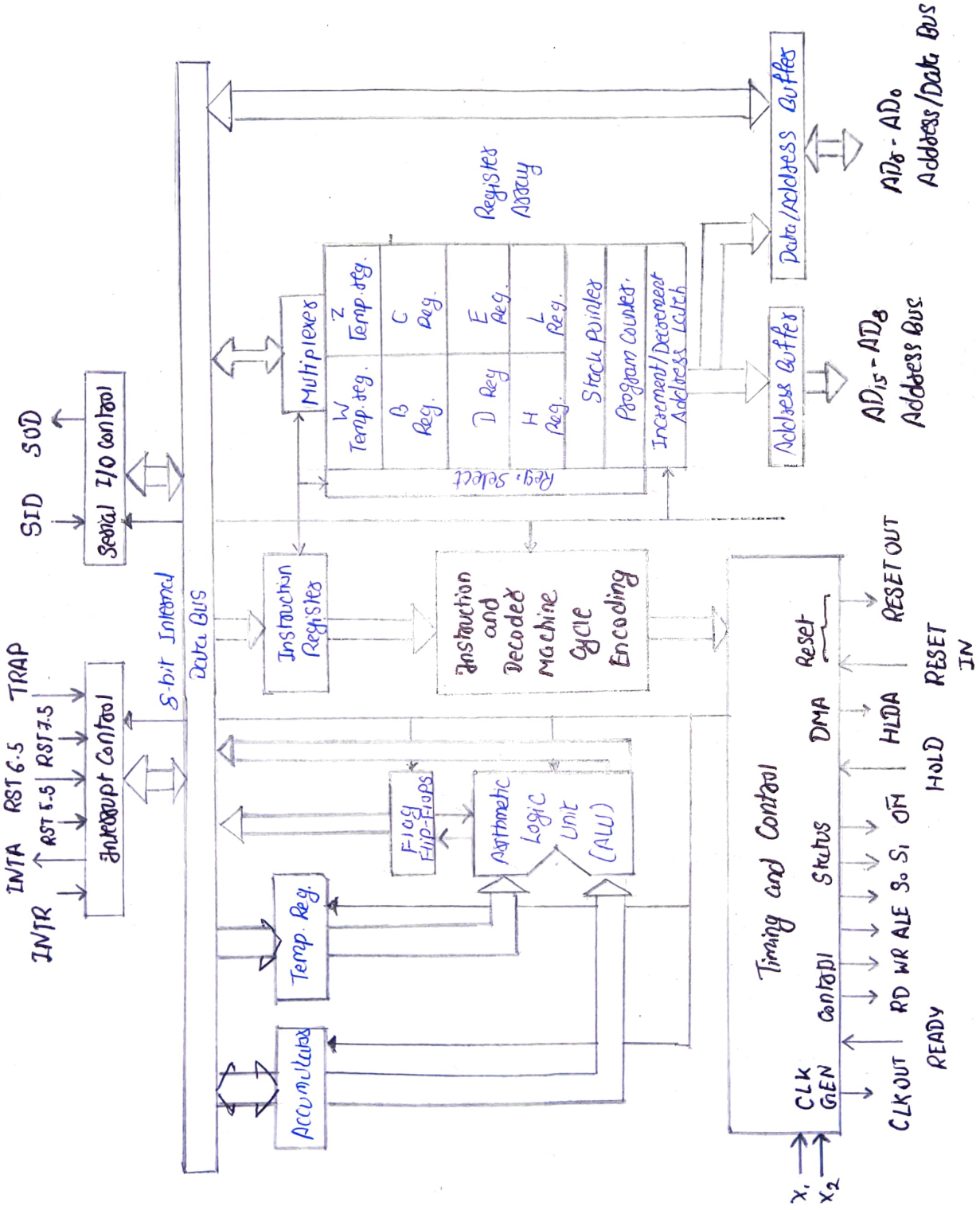
1. Solution 1.



Name: Kumawat Lakhan Makhanlal
Roll No: 1906055
Course: Microprocessor and Microarchitecture

Course code: CS5469
Date: 20/12/21
Page No:

1. Solution 1.



Name: Kumawat Lakhan Makhanlal.
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Microarchitecture.

Course code: CS5469
Date: 20/12/21
Page No:

8085 Microprocessor.

8085 is an 8-bit microprocessor as it operates on 8 bits and is created with N-MOS technology.

Architecture of 8085 Microprocessor.

The architecture of 8085 microprocessor provides the idea about what are the operations to be executed and how these are performed

1. Arithmetic and Logic unit (ALU):
   - It is used to perform mathematical operations like: addition, multiplication subtraction, division, decrement, increment etc.

2. Flag Register:
   - It is an 8-bit register that stores either 0 or 1 depending upon which value is stored in accumulator.

3. Accumulator:
   - Accumulator is used to perform I/O arithmetic and logical operations. It is connected to ALU and internal data bus.

4. General purpose Register:
   - There are 6 general purpose registers. These registers can hold 8-bit values.
   - These 8-bit registers are B, C, D, E, H, L.
   - These registers work as 16-bit registers when they work in pair like: B-C, D-E, H-L.

5. Program Counter:
   - program Counter holds the address value of the memory to the next instruction that is to be executed. It is a 16-bit register.

6. Stack pointer:
   - It works like stack. In stack, the content of register stored that is later used in the program.

Name: kumawat Lakhan Makhanlal.
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Microarchitecture.

Course code: CS5469
Date: 20/12/21
Page No:

7. Temporary Register:
- It is 8-bit register that holds data values during arithmetic and logical operations.

8. Instruction register and decoder:
- It is a 8-bit register that holds the instruction code that is being decoded. The instruction is fetched from the memory.

9. Timing and Control unit:
- Timing and control unit comes under CPU section and it controls the flow of data from CPU to other devices. It is also used to control the operations performed by the microprocessor and the devices connected to it.

10. Interrupt Control:
- Whenever a microprocessor is executing a main program and if suddenly an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After request is completed, the control goes back to the main program
- 5 interrupt in 8085 microprocessors: INTR, TRAP, RST 7.5/6.5/5.5.

11. Address bus and data bus:
- Data bus is bidirectional and carries the data which is to be stored.
- Address bus is unidirectional and carries location where data is to be stored.

12. Serial Input/output Control:
- It controls the serial data communication by using serial input data and serial output data.

Solution 2.7

1. Data Transfer Instructions:

1. MOV : This instruction copies the contents of source register into destination register without any alteration. MOV K,L

2. MVI: The 8-bit data is stored in the destination register or memory. MVI K,55L

3. LDA : The contents of a memory location, specified by a 16-bit address in the operand are copied to accumulator. LDA 2034K.

4. LXI : The instruction loads 16-bit data in the register pair designated in the register or the memory.

5. STA - The contents of the accumulator are copied into the memory location specified by the operand.

2. Arithmetic Instructions:

1. ADD : The contents of registers or memory are added to contents of the accumulator and result is stored in accumulator.

2. ADC : The contents of register or memory & M the carry flag are added to the contents of accumator and result is stored in accumulator.

3. ADI: The 8-bit data is added to contents of accumulator and result is stored in accumulator.

4. ACI: The 8-bit and carry flag are added to content of accumulator and result is stored in accumulator.

5. DAD: The 16-bit data of the specified register pair are added to the contents of HL register.

3. Logical Instruction.

	opcode	operand	Meaning	Example
1.	CMP	R M	Compare register / memory with accumulator	CMP M.

2.	STC	None	Set Carry	STC
3.	XRA	R M	Exclusive OR Register/memory with accumulator	XRA A
4.	ORA	R M	Logical OR Register/ memory with accumulator	ORA A
5.	ANA	R M	Logical AND register/ memory with accumulator	ANA M

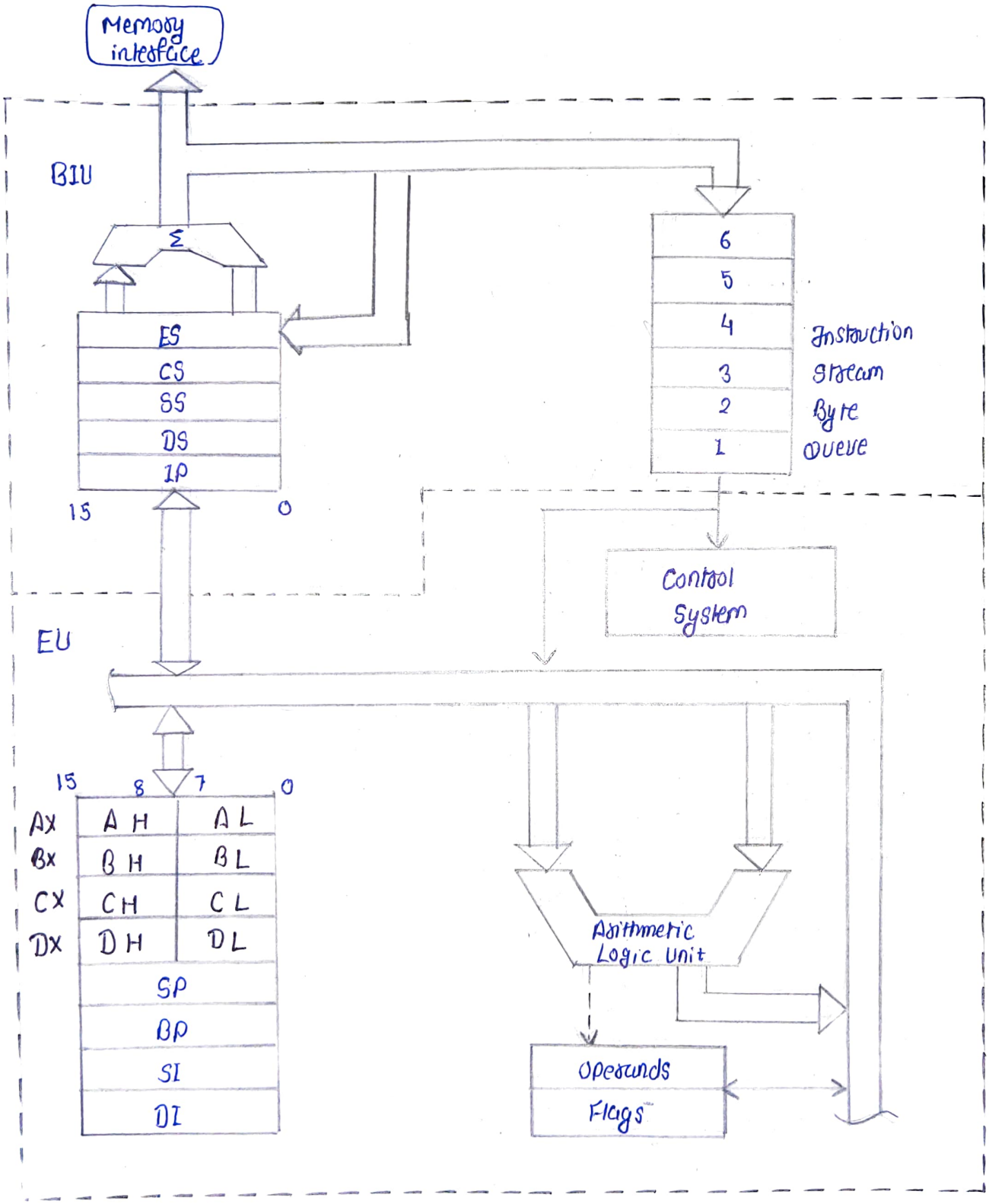
4. Branching Instructions:

	Opcode	operand	Meaning	Example.
1.	JMP	16-bit address	Jump unconditionally	JMP 2034H
2.	CALL	16-bit Address	Call unconditionally	CALL 2034H
3.	RET	None	Return unconditionally	RET
4.	RST	0-7	Restart (software Interrupt)	RST 3
5.	HLT	None	Halt the program	HLT

Name : Kumawat Lakhan Makhanlal
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Micro architecture

Course Code: CS5469
Date: 20/12/21
Page No:

3.



8086 Internal Architecture.

Name : Kumawat Lakhan Makhanlal
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Micro architecture

Course Code: CS5469
Date: 20/12/21
Page No:

Register Organization in 8086:

1. General 16-bit registers:

The registers AX, BX, CX and DX are general 16-bit registers.

AX Register: Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains high-order byte.

BX Register: This register is mainly used as a base register. It holds the starting base location of a memory region within a data segment. It is used as offset storage for forming physical address in case of certain addressing mode.

CX Register: It is used as default counter or count register in case of string and loop instructions.

DX Register: Data register can be used as a port number in I/O operations and implicit operand or destination in case of few instructions. In integer 32-bit multiply and divide instruction the DX register contains high-ordered word of the initial or resulting.

2. Segment Registers:

To complete 1 Mbyte memory is divided into 16 logical segments. The complete 1 Mbyte memory segmentation. There are four segment registers.

Code Segment (CS) is a 16 bit register containing address of 64 kB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly.

Stack Segment (SS) is a 16-bit register containing address of 64kB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer and base pointer registers is located in stack segment. SS register can be changed directly using pop instruction.

Name : Kumawat Lakhan Makhanlal
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Micro architecture

Course Code: CS5469
Date: 20/12/21
Page No:

Data Segment (DS) is a 16-bit register containing address of 64kB segment with program data. By default, the processor assumes that all data referenced by general registers and index registers is located in Data Segment.

Extra Segment (ES) is a 16-bit register address of 64kB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions.

3. Pointers and Index Registers:

The pointers IP, BP, SP usually contain offsets with in the code.

Stack pointer (SP) is a 16-bit register pointing to program stack in stack segment.

Base pointer (BP) is a 16 bit register pointing to data in Stack Segment.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing.

Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as destination data address.

4. Conditional Flags:

Carry Flag: This flag indicates overflow condition for unsigned integer arithmetic.

Auxiliary Flag: If an operation performed in ALU generates a carry/borrow from lower nibble to upper nibble, the AC flag is set.

Parity Flag: This flag is used to indicate parity of result.

Name : Kumawat Lakhan Makhanlal
Roll No: 1906055
Branch: CSE-1
Course: Microprocessor and Micro architecture

Course Code: CS5469
Date: 20/12/21
Page No:

Zero Flag: Set, if result of arithmetic or logical operation is zero.

Sign Flag. The sign of MSB bit, if result of operation is negative Sign flag is set

5. Control Flags:

Trap Flag: It is used for Single Step control, allows user to execute one instruction of program at a time.

Interrupt Flag: It is an interrupt enable / disable flag

Direction Flag. It is used in string operation. If it is set string bytes are accessed from higher memory address to lower memory address

Name: Kumawat Lakhan Makhanlal
Roll No: 1906055
Course: Microprocessor and Microarchitecture

Course code: CS5469
Date: 20/12/21
Page No:

Solution 4:

1. To Do: Assembly language program to transfer a character "H" serially at baude date of 9600.

2. ToDo: Assembly language program to receive bytes serially with baud rate of 9600 and simultaneously send the received bytes to port-1.

Program to transfer letter "H" serially at 9800 baud, continuously:

```
MOV   TMOD, #20H      : Timer 1, mode 2 (auto reload)
MOV   TH1, #-3 FDH    : 9600 baud rate
MOV   SCON, #50H      : 8-bit, 1 stop, REN enabled.
SETB  TR1             ; Start timer 1.
AGAIN : MOV SBUF, #"H"  ; letter "H" to transfer.
HERE : JNB TI, HERE     ; wait for last bit
       CLR TI           ; clear TI for next char
       SJMP AGAIN       ; keep sending A.
```

Name: Kumawat Lakhan Makhanlal
Roll No: 1906055
Course: Microprocessor and Microarchitecture

Course code: CS5469
Date: 20/12/21
Page No:

1. Steps to send data serially.

1. Set baud rate by loading TMOD register with the value 20H, this indicating timer 1 in 2 mode 2 to set baud rate.
2. The TH1 is loaded with proper values to set baud rate for serial data transfer. (#FDH)
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8-bit data is framed with start and stop bits.
4. TR1 is set to 1 to start timer 1.
5. TI is cleared by CLR TI instruction.
6. The character byte to be transferred serially is written in SBUF register.
7. The TI flag bit is monitored with the use of instruction JNB, TI, xx to see if the character has been transferred completely.
8. To transfer next byte, go to step 5.

Name: Kumawat Lakhan Makhanlal
Roll No: 1906055
Course: Microprocessor and Microarchitecture

Course code: CS5469
Date: 20/12/21
Page No:

2. Steps to receive data serially:

1. Set baud rate by loading registers with value 20H, indicating timer 1 in mode ~~dad~~ 2 to set baud rate.
2. The TH1 is loaded with proper values to set baud rate
3. The SCON registers is loaded with the value 50H, indicating serial mode 1 where an 8-bit data is framed with start and stop bits
4. TR1 is set to 1 to start timer 1.
5. RI is cleared by CLR RI instruction.
6. The RI flag is monitored with the use of instruction JNB RI, xx to see if an entire character has been received yet.
7. When RI is raised, SBUF has the byte; its contents are moved into a safe place.
8. To receive next character, go to step 5;

```
        MOV   TMOD,#20H     : time 1, mode 2 (auto reload)
        MOV   TH1,#FDH      : 9600 baud rate
        MOV   SCON,#50H     : 8-bit, 1 Stop, REN Enabled
        SETB  TR1           : Start timer 1
HERE:   JNB   RI,HERE       : wait for char to come in
        MOV   A,SBUF        : saving incoming byte in A
        MOV   P2,A          : send to port 1
        CLR   RI            : get ready to receive next byte
        SJMP  HERE          : keep getting data.
```