



**School of Electronics Engineering ,VIT, Vellore**

<b>Reg.No</b>	19BEC0358		
<b>Student Name</b>	ARPIT PATAWAT		
<b>Course Code</b>	ECE3003	<b>Slot &amp; Semester</b>	L35+L36 FALL -- 2021-22
<b>Course Name</b>	Microcontroller and its applications		
<b>Program Title</b>	SERIAL COMMUNICATION		
<b>Date of Exp.</b>	11-10-2021	<b>Date of Submission</b>	23-10-2021
<b>Faculty</b>	A.Karthikeyan		

### Question

---

1. Write an 8051-assembly program to transfer data serially at baud rate 9600 with 8 bit data, one stop bit and observe the transmitted data in the serial window of the simulator.
2. Write an 8051 Assembly Language program to get data from the PC and display it on P1. Assume 8051 is connected to PC and observe the incoming characters. As you press a key on the PC's keyboard, the character is sent to the 8051 serially at 4800 baud rate and is displayed on LEDs. The characters displayed on LEDs are in ASCII (binary).
3. Assume that the 8051 serial port is connected to the COM port of IBM PC, P1 and P2 of the 8051 are connected to LEDs and switches, respectively.  
Assume that XTAL-11.0592Mhz.

Write an 8051 assembly program to

(a) send to PC the message "NAMEANDREGNO", OUTPUT- WILL BE IN UART

(b) receive any data send by PC and put it on LEDs connected to P1,  
and INPUT - UART -- OUTPUT IN P1

(c) get data on switches connected to P2 and send it to PC serially.

4800 baud rate P2 INPUT AND OUTPUT - UART

**TASK 1) -**

**Aim:** To write an 8051 ALP to transfer data serially and observe the transmitted data using keil software and to verify the result manually.

**Tools Required:** Keil Micro vision Software

**Algorithm:**

1. assign address of string data to data pointer
2. select timer 1 mode 2
3. baud rate = 9600 → set TH1 = -3 / FD
4. set SCON = 50 (8 bit data transfer with receive enable)
5. start timer 1 and take the first byte of data using DPTR and load it to A
6. copy data of A to SBUF and monitor Transmit interrupt flag
7. if flag becomes 1 then clear flag and repeat the same process for the next byte

**Program:**

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000H				Assembler directive	Defining origin of the program	NONE
XX	MOV	DPTR, #MYDATA	Immediate	2	3	Data transfer	Assign DPTR to string data	None
	MOV	TMOD, #20H	Immediate	1	2	Data transfer	Timer 1 mode is selected	None
	MOV	TH1, #-3	Immediate	1	2	Data transfer	Bode rate 9600 is selected	None
	MOV	SCON, #50H	Immediate	1	2	Data transfer	8 bit data transfer mode with REN	None

							enabled	
	SETB	TR1	Direct	1	1	Boolean	Start timer 1	NONE
	MOV	R1, #15	Immediate	1	2	Data transfer	Running the loop for 15 times	NONE
AGAIN:	CLR	A	Direct	1	1	Logical	Clear	NONE
	MOVC	A, @A+DPTR	Indexed	1	2	Data transfer	Load the data from ROM to A	PARITY
	MOV	SBUF, A	Register	1	1	Data transfer	Data moved from SBUF accumulator	NONE
HERE:	JNB	TI, HERE	Direct	2	3	Boolean	Jump to here label if TI = 0	NONE
	CLR	TI	Direct	1	1	Logical	Clear TI flag	NONE
	INC	DPTR	Direct	2	1	Arithmetic	Increment DPTR	NONE
	DJNZ	R1, AGAIN	Direct	2	2	Program branching	Decrement R1 till it becomes 0	NONE
	SJMP	XX		2	2	Program branching	Short jump to XX label	NONE
MYDATA:	DB	'ARPIT 19BEC0358'					Define string data	NONE
	END					Assembler	End of	NONE

						directive	program	
--	--	--	--	--	--	-----------	---------	--

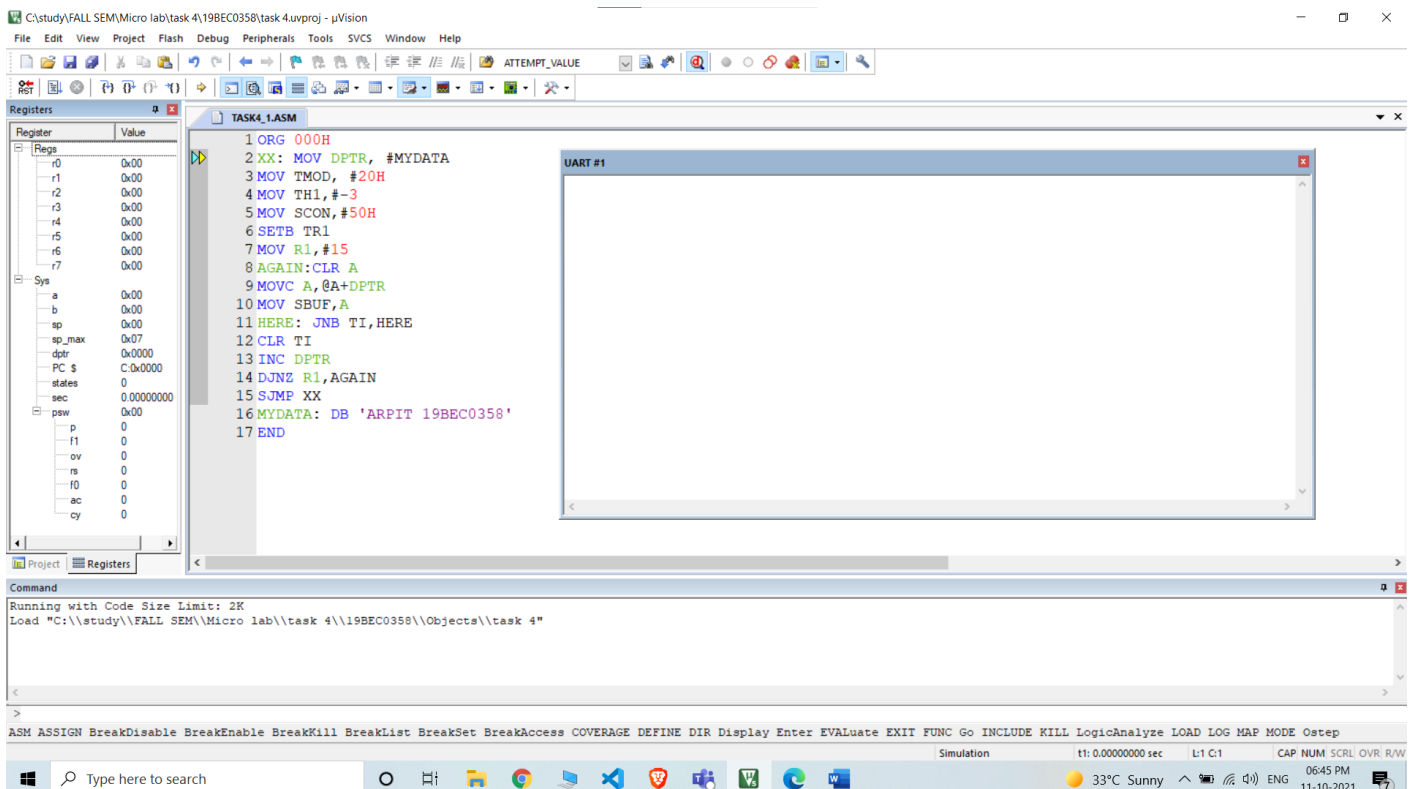
**Output:** serial monitor containing the Result: ARPIT 19BEC0358

**Manual Calculation:** BODE RATE = 9600

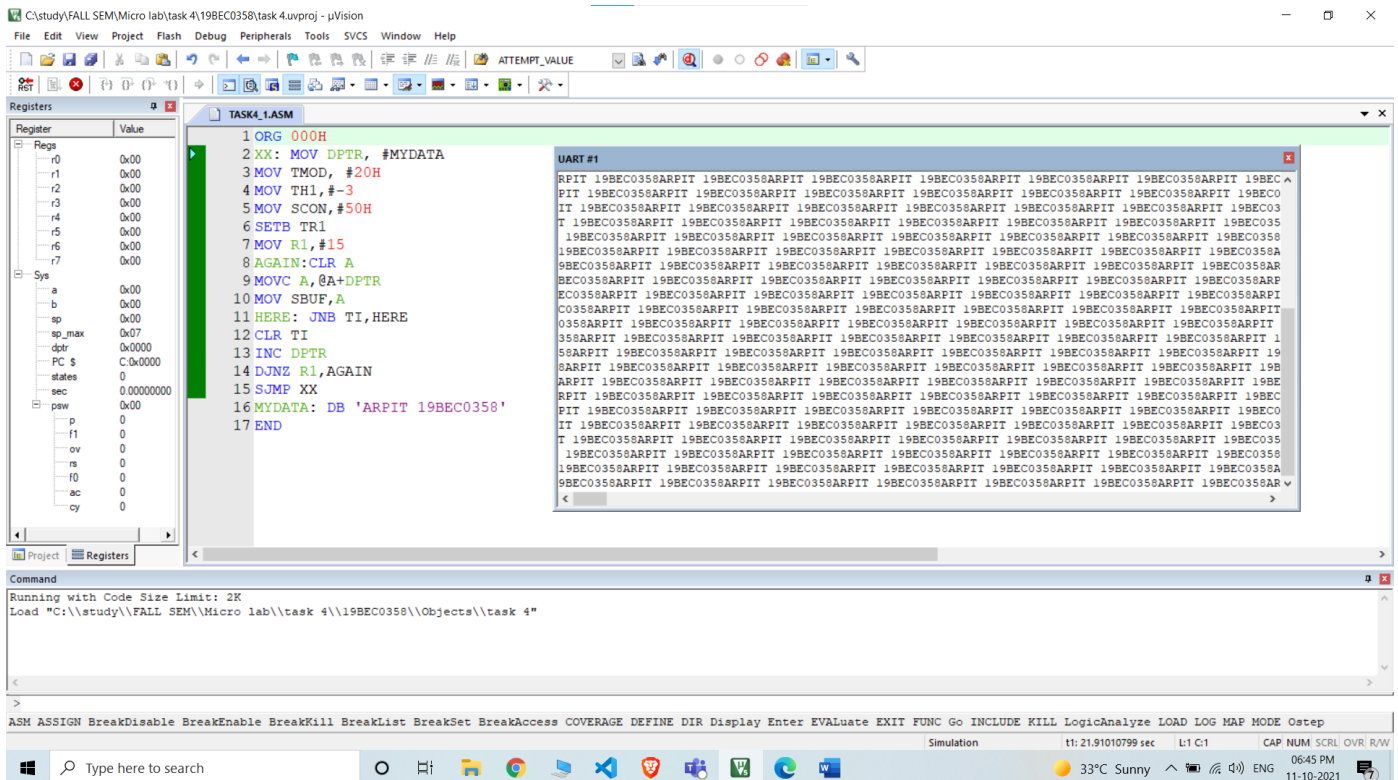
$$28,800 / 9600 = 3$$

## Results and Observations

### Program and registers before execution:



### Program and registers after execution: FINAL STEP



## Inferences:

1. About PSW VALUES – they remain unchanged except parity
2. ABOUT THE OUTPUT VALUES IN REGISTERS –no change, the final result is in serial monitor where name and register number is displayed.

**Result:** the 8051 ALP to transfer data serially and observe the transmitted data using Keil software is executed and the results are verified Manually.

**TASK 2) -**

**Aim:** To write an 8051 ALP get data from PC and display on Port1 using keil software and to verify the result manually.

**Tools Required:** Keil Micro vision Software

**Algorithm:**

1. select timer 1 mode 2
2. baud rate = 4800 → set TH1 = -6 / FA
3. set SCON = 50 (8 bit data transfer with receive enable)
4. start timer 1
5. monitor receive interrupt flag
6. if flag becomes 1 then load the data from SBUF to A then A to port 1
7. clear flag and repeat the same process for the next byte

**Program:**

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000H				Assembler directive	Defining origin of the program	NONE
	MOV	TMOD, #20H	Immediate	1	2	Data transfer	Timer 1 mode is selected	None
	MOV	TH1, #-6	Immediate	1	2	Data transfer	Bode rate 4800 is selected	None
	MOV	SCON, #50H	Immediate	1	2	Data transfer	8 bit data transfer mode with REN enabled	None
	SETB	TR1	Direct	1	1	Boolean	Start timer 1	NONE

HERE :	JNB	RI, HERE	Direct	2	3	Boolean	Jump to HERE label if RI = 0	NONE
	MOV	A, SBUF	Register	1	1	Data transfer	Data moved from SBUF to accumulat or	NONE
	MOV	P1, A	Register	1	1	Data transfer	Data is moved from A to port 1	None
	CLR	RI	Direct	1	1	Logical	Clear RI flag	NONE
	SJMP	HERE		2	2	Program branchin g	Short jump to HERE label	NONE
	END					Assembl er directive	End of program	NONE

**Output:** Port 1 containing the Result: 41 which is when 'A' is pressed.

**Manual Calculation:** BAUD RATE = 4800

$$28,800 / 4800 = 6$$

### Results and Observations

**Program and registers before execution:**



Registers window (Initial State):

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC	0x0000
states	0
sec	0.00000000
psw	0x00

Code window (TASK4\_2.ASM):

```

1 ORG 0000H
2 MOV TMOD, #20H
3 MOV TH1, #-6
4 MOV SCON, #50H
5 SETB TR1
6 HERE: JNB RI, HERE
7 MOV A, SBUF
8 MOV P1, A
9 CLR RI
10 SJMP HERE
11 END
12

```

Parallel Port 1 window:

Port 1	7 Bits	0
P1: 0xFF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Pins: 0xFF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Command window:

```

Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 4\\19BEC0358\\Objects\\task 4"

```

## Program and registers after execution: FINAL STEP

Registers window (Final State):

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC	0x0000
states	0
sec	0.00000000
psw	0x00

Code window (TASK4\_2.ASM):

```

1 ORG 0000H
2 MOV TMOD, #20H
3 MOV TH1, #-6
4 MOV SCON, #50H
5 SETB TR1
6 HERE: JNB RI, HERE
7 MOV A, SBUF
8 MOV P1, A
9 CLR RI
10 SJMP HERE
11 END
12

```

Parallel Port 1 window:

Port 1	7 Bits	0
P1: 0x41	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pins: 0x41	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Command window:

```

Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 4\\19BEC0358\\Objects\\task 4"

```

### Inferences:

1. About PSW VALUES – they remain unchanged
2. ABOUT THE OUTPUT VALUES IN REGISTERS –no change, the final result is in port 1 where ascii value of input letter is displayed ( A/41 in this case)

**Result:** the 8051 ALP to ALP get data from PC and display on Port1 using Keil software is executed and the results are verified Manually.

### TASK 3) -

**Aim:** To write an 8051 ALP to get transfer data serially then to receive data from keyboard and display it on port 1 then finally take data from port 2 and display it on serial window using keil software and to verify the result manually.

**Tools Required:** Keil Micro vision Software

**Algorithm:**

1. select timer 1 mode 2
2. baud rate = 4800 → set TH1 = -6 / FA
3. set SCON = 50 (8 bit data transfer with receive enable)
4. start timer 1
5. Make port 1 as input
6. Take the string data using DPTR and display on serial monitor
7. Receive data from UART and display on port 1
8. Take the data from port 2 and display on serial window

**Program:**

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000H				Assembler directive	Defining origin of the program	NONE
	MOV	TMOD, #20H	Immediate	1	2	Data transfer	Timer 1 mode is selected	None
	MOV	TH1, #-6	Immediate	1	2	Data transfer	Baud rate 4800 is selected	None

	MOV	SCON, #50H	Immediate	1	2	Data transfer	8 bit data transfer mode with REN enabled	None
	MOV	P2, #11111111B	Immediate			Data transfer	Port 2 is made input	NONE
	SETB	TR1	Direct	1	1	Boolean	Start timer 1	NONE
	MOV	R1, #15	Immediate	1	2	Data transfer	Running the loop for 15 times	NONE
	MOV	DPTR, #MYDATA	Immediate	2	3	Data transfer	Assign DPTR to string data	None
LOOP :	CLR	A	Direct	1	1	Logical	Clear	NONE
	MOVC	A, @A+DPTR	Indexed	1	2	Data transfer	Load the data from ROM to A	PARITY
	MOV	SBUF, A	Register	1	1	Data transfer	Data moved from A to SBUF	NONE
HERE :	JNB	TI, HERE	Direct	2	3	Boolean	Jump to here label if TI = 0	NONE
	CLR	TI	Direct	1	1	Logical	Clear TI flag	NONE
	INC	DPTR	Direct	2	1	Arithmetic	Increment DPTR	NONE
	DJNZ	R1, LOOP	Direct	2	2	Program branching	Decrement R1 till it becomes 0	NONE

AGAI N:	JNB	RI, AGAIN	Direct	2	3	Boolean	Jump to AGAIN label if RI = 0	NONE
	MOV	A, SBUF	Register	1	1	Data transfer	Data moved from SBUF to accumulat or	NONE
	MOV	P1, A	Register	1	1	Data transfer	Data is moved from A to port 1	None
	CLR	RI	Direct	1	1	Logical	Clear RI flag	NONE
	MOV	A, P2	Register	1	1	Data transfer	Data is moved from port 2 to A	None
	MOV	SBUF, A	Register	1	1	Data transfer	Data moved from A to SBUF	NONE
THIS:	JNB	TI, THIS	Direct	2	3	Boolean	Jump to this label if TI = 0	NONE
	CLR	TI	Direct	1	1	Logical	Clear TI flag	NONE
	SJMP	AGAIN		2	2	Program branchin g	Short jump to again label	NONE
MYD ATA:	DB	'ARPIT 19BEC0 358'					Define string data	NONE
	END							

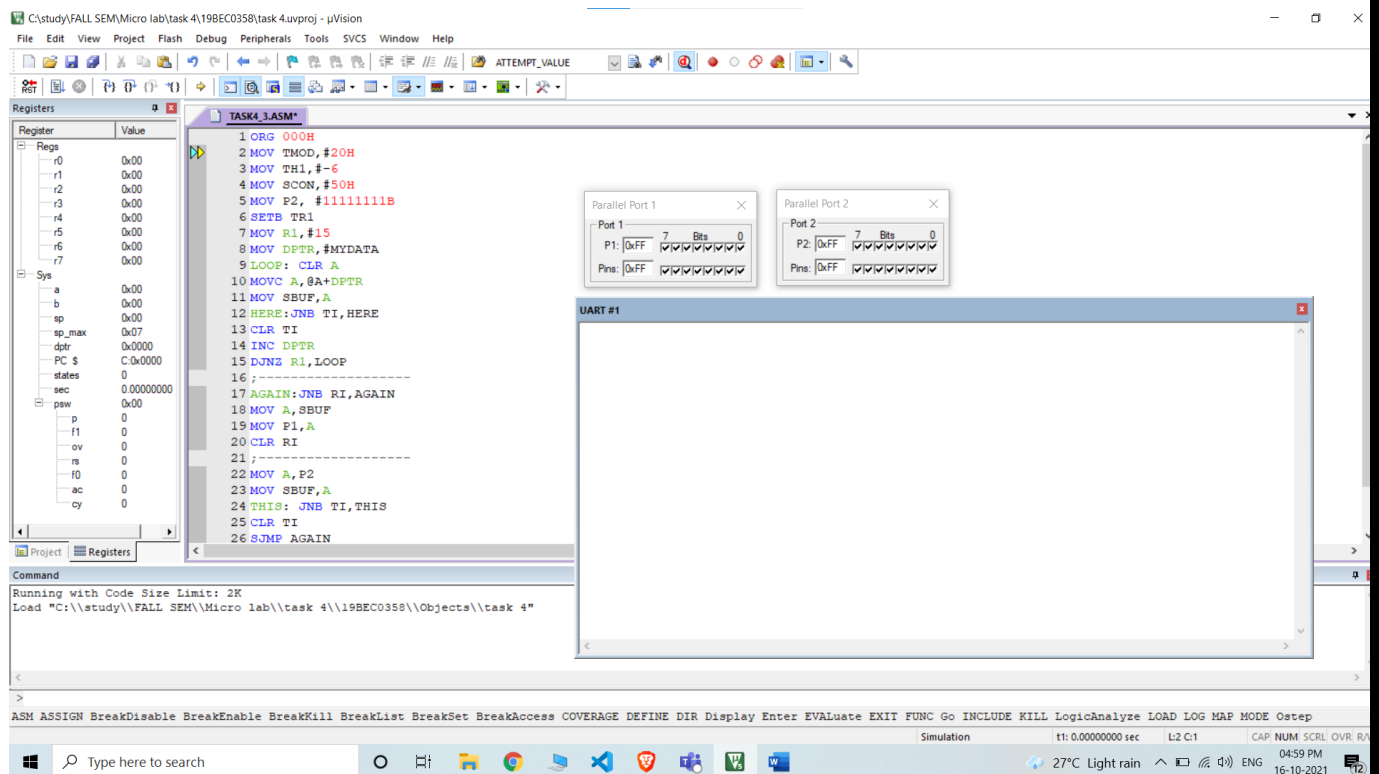
**Output:** Serial window: string data and input from port 2, port 1: input from serial window.

**Manual Calculation: BAUD RATE = 4800**

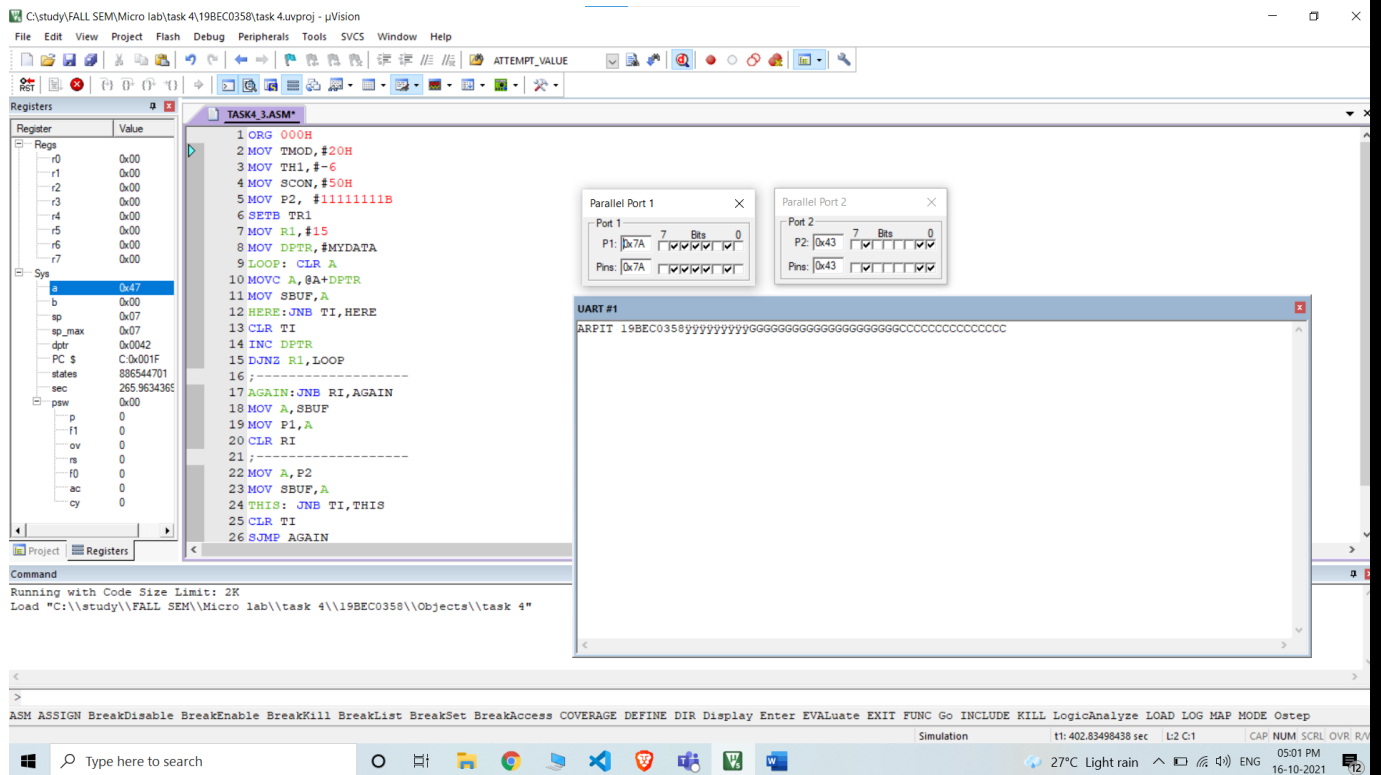
$$28,800 / 4800 = 6$$

## Results and Observations

### Program and registers before execution:



### Program and registers after execution: FINAL STEP



## Inferences:

1. About PSW VALUES – they remain unchanged
2. ABOUT THE OUTPUT VALUES IN REGISTERS –no change, the final result is in port 1 and serial window

**Result:** the 8051 ALP to display string data to serial window & output to port 2 and take input from port 1 using Keil software is executed and the results are verified Manually.

-----XXXXXXXX-----