**ISSUES AND SOLUTIONS:**

**LAB 1 PART1**

* FACED ISSUES TO INPUT CHARACTERS IN THE FORM OF HEXADECIMAL STRING TO AN HEXADECIMAL NUMBER WHERE THE CHARACTER CAN HAVE ALPHABETS FROM A TO F AND DIGITS. USED STRTOL TO CONVERT THE HEX STRING TO HEX
* HEX DUMP FACED ISSUE WHEN PAGE ADDRESS OF HEX START ADDRESS AND HEX END ADDRESS ARE DIFFERENT. LATER ON CHANGED THE PAGE ADDRESS, WHICH IS PART OF THE SLAVE ADDRESS AND STARTIED PRINTING CORRECT VALUE OF ADDRESS.
* COULD NOT GET WRITE BYTE WORKING WHEN PRINTED ONE AFTER THE OTHER. THEN LEARNT ABOUT ACKNOWLEDGE POLLING AND SOLVED THE ISSUE. AFTER SOLVING, THE ACKNOWLEGEMENT WAS RECEIVED AND COULD WRITE DATA IN SEQUENTIAL MANNER.
* PROBLEM WITH ENTER BEING TAKEN AS A CHARACTER. TOOK ASCII VALUE OF IT,13 AND MADE USER REENTER CHARACTER.
* DURING HEX DUMP, SEQUENTIAL READ WAS PERFORMED FROM START HEX ADDRESS TO END HEX ADDRESS. SO, IF HEX ADDRESS IS NOT A MULTIPLE OF SIXTEEN, COULD NOT FIGURE HOW TO PRINT THE FIRST LINE 16 TIMES. TOOK A DIFFERENCE OF END AND START HEX ADDRESS AND SEND THE WORD ADDRESS BEFORE FOR LOOP TO ENTER THE VALUE OF DATA AND ADDRESS IN THE CORRECT MANNER.
* PROBLEM OF RECEIVING ACKNOWLEDGE AT END OF READ DATA. SEND DELAY BETWEEN WRITE AND READ OPERATIONS TO SOLVE THE ISSUE.
* USED ARRAYS EARLIER FOR SEQUENTIAL READ WHERE ARAY SIZE NEEDED TO BE MENTIONED. FOR LARGE VALUE OF ARRAY ADDRESS, THE PROGRAM PRINTED GARBAGE VALUES AT THE CODE LOCATION. ISSUE SOLVED BY TAKING VALUE OF READ OPERATION IN A SINGLE VARIABLE AND PRINTING IT CONTINUOUSLY IN A FOR LOOP.
* COULD NOT GET PROPER COMMUNICATION SIGNALS ON THE LOGIC ANALYSER. NEEDED TO CHANGE SETTINGS WITH POST FILL TIME LIMIT OF 10 SEC, TRIGGER IMMEDIATELY WHEN ACQUISITION STARTS, SAMPLE FREQUNCY TO 500KHZ. ALSO, FOR SOME OPERATIONS, DID NOT GET THE START SIGNAL. MADE SDA HIGH INITIALLY SO THAT START SIGNAL CAN PROCEED.
* DID NOT KNOW THE DIFFERENCE BETWEEN MASTER SENDING ACK AND EEPROM SENDING ACK. WHEN MASTER SENDS ACK MADE SDA LOW AND SENT A CLOCK PULSE DURING THAT TIME. FOR EEPROM TO SEND ACK, MADE SDA HIGH AND SENT A CLOCK PUSLE TO WIAT FOR EEPROM TO MAKE SEND ACK BY MAKING SDA LOW.
* USED FOR LOOP TO PRINT SAME VALUE OF DATA WITH AN ACKNOWLEDGEMENT TO WRITE TO THE PAGE ADDRESS. WHEN USED PRINT STATEMENTS IN BETWEEN, THE VALUE OF WRITE DATA COUL NOT BE WRITTEN PROPERLY.

HAD ISSUES WITH GETTING READ AFTER HEX DUMP. OBSERVED THAT SDA SIGNAL WAS REMAINING LOW BEFIRE THE READ CYCLE AND HENCE ONLY READING DURING THE SECOND READ OPERATION. MADE SDA HIGH AND GOT THE READ OUTPUT CORRECT

* TOOK REFERENCE OF A DOUBT OF I2C CODE ON A FORUM AND UNDERSTOOD THE I2C IMPLEMENTATION AND MADE CHANGES TO THE CODE BY LEARNING ABOUT READ, WRITE, SEQUENTIAL READ, PAGE WRITE CYCLES, WHICH SPECIFIED WHERE ACK, NACK NEEDS TO BE THERE, WHERE SDA, SCL WILL BE LOW OR HIGH, AND WHO IS THE RECEIVER, WHICH WILL DECIDE WHO WILL SEND THE SIGNALS.

**LAB 2 PART2**

* FACED A LOT OF ISSUES WITH LCD DISPLAY. INCLUDING FROM HARDWARE TO SOFTWARE.
* WHEN LCD WAS CONNECTED WITH THE WIRE WRAPPINGS, IT DID NOT SEEM TO TURN ON . USED A POTENTIOMETER TO TURN IT ON. BUT COULD NOT GET IT WORKING AFTER REPEATED CONNECTION CHECKS . FOUND OUT THAT THE LCD SOMEHOW WAS NOT GETTING PROPER CONNECTION EVEN AFTER THE SOLDERING WAS DONE PROPERLY.
* AFTER SWITCHING ON THE LCD COULD NOT BE INITIALSED WITH THE LCD INITIALISATION ROUTINE. FOUND OUT THAT THE NVRAM PLD WAS REPLACED WITH THE LCD SPLD CODE AND THERE WAS NO MEMORY WHERE LCD COULD HAVE ITS DATA STORED. REPLACED THE SPLD CODE WITH THE ANOTHER SPLD CODE AND INITIALISATION WORKED.
* THE LCD SEEMED TO GIVE RANDOM VALUES ON PRINTING CHARACTERS. USED A DECOUPLING CAPACTOR AND USED DELAYS BETWEEN FUNCTIONS TO GET THE LCD TO WORK
* THE INTERRUPT TIMER REGISTERS WERE INITIALISED CORRECTLY BUT THE TIMER WOULD NOT DISPLAY THE CLOCK. FOUND OUT THAT IN THE INITIALISATION TMOD REGISTER WAS TMOD=1 AND NOT EQUAL TO TMOD|=1 WHICH IS WHY TIMER LOOP WOULD NOT RUN
* FOUND OUT THAT EVEN AFTER GOING INTO THE TIMING CLOCK LOOP, THE CURSOR WOULD SWITCH POSITIONS WHERE CLOCK NEEDS TO BE PRINTED BUT WOULD NOT WRITE THE VALUE OF THE CLOCK. SAME THING WHEN DONE WITH TOP PART OF THE LCD DISPLAY GAVE CORRECT RESULTS.
* THE UI SEEMED TO MESS UP THE LCD WORKING. THE LCD PRINTED VALUES AT RANDOM LOCATION. FIGURES OUT THAT THE TIMER WHERE GETTING END ON THE LAST COLUMN, THE VALUE GOT INCREMENTED TO THIRD A COLUMN ONE MORE THAN THE WHERE THE CLOCK ENDED BECAUSE OF AUTO INCREMENT DONE WHEN PRINTING CHARACTERS IN LCD. MADE USER DEFINED CHARACTER AND LOCATION WHERE CHARACTER WAS DEFINED GLOBAL AND PRINTED THE VALUE FOR THAT IN THE INFINITE CLOCK LOOP SO WHEN CHARACTER ‘W’ OR ‘P ’ IS USED.
* WHEN THE TIMER STAYED OFF THE LCD WORKED ACCURATELY BUT WHEN CLOCK WAS USED THE UART SEEMED TO WORK VERY SLOWLY . FOUND THAT THE TIMER HAD SOME DELAY. WHEN IT WAS REMOVED FROM PUTCHAR FUNCTION WHILE AND SOME DELAY FROM THE TIMER FUNCTION WAS REMOVED, IT WORKED WELL.
* PUTTY’S VECTOR ADDRESS WAS CHANGED TO CODE LOCATION FOR THE INTEERUPT TO WORK.
* FIGURING OUT HOW TO WRITE TO THE LCD DISPLAY AND MAKING THE CONNECTIONS WAS A TOUGH PART. FOUND OUT THAT THE RS R/W AND ENABLE SIGNALS NEED TO BE MAPPED WITH MEMORY MAPPING WHERE THE ENABLE PIN ONLY TURNS ON WHEN UPPER PART OF ADDRESS SIGNALS ARE USED AND ALL ARE 1. USED UPPER PART OF ADDRESSES AS LOWER PART OF ADDRESSES HAVE COULD HAVE SOME OPERATION GOING ON. THE ENABLE WORKS ONLY WHEN EITHE ROF READ OR WRITE IS USED AND ALL THE ADDRESS PINS ARE 1.
* USED LED TOGGLE TO KNOW IF FUNCTION GOES INSIDE THE ISR. THE LED TOGGLED AFTER INTERRUPT INITILAISATION BUT SEEMED TO STOP AFTER A PRINTF STATEMENT WAS USED INSIDE IT. ALSO PRINTF STATEMENT OUTSIDE ANY FUNCTIONS OTHER THAN LCD INITIALISATION WOULD NOT PRINT INSIDE THE UART.
* FOUND OUT THAT AFTER THE CLOCK PRINTED IT WOULD ROLL OVER AND PRINT ON INREMENTAL LOCATIONS. USED AND ARRAY TO LOAD THE TIMER VALUES AND PRINTED ONLY THAT ARRAY TO PRINT TO THE SAME ADDRESSES FOR CLOCK.
* NEED TO TRANSER INSTRUCTIONS AND VALUES THROUGH MOVX INSTRUCTION.
* NEED TO WAIT FOR LCD BUSY WAIT INSTEAD OF JUST GIVIGN DELAY FOR LCD INSTRUCTION AND DATA COMMANDS TO BE EXECUTED OTHEWISE AN ERROR IS THROWN IN FORM OF INVALID ADDRESSING OR DATA PRINTING.
* USED UINT32 AS SIZE OF MILI WHEN ISR IS EXECUTED OTHERWISE THE VALUE OF COUNT WOULD OVERFLOW AND RESET BACK TO ZERO.
* USED EXTERN AND VOLATILE FOR DEFINING VARIABLES SO THAT THEY CAN BE USED IN DIFFERENT FILES AND COMPILER DOES NOT OPTIMIZE IT TO ANY VALUE OTHER THAN DONE IN THE PROGRAM.
* USED TIMER MODE 1 INTERRUPT AND USED 50 USED LED TOGGLE TO KNOW IF FUNCTION GOES INSIDE THE ISR. THE LED TOGGLED AFTER INTERRUPT INITILAISATION BUT SEEMED TO STOP AFTER A PRINTF STATEMENT WAS USED INSIDE IT. ALSO PRINTF STATEMENT OUTSIDE ANY OTHER FUNCTIONS
* FOUNF OUT THAT USING DECOUPLING CAPACITOR REDUCES NOISE BY A BIG MARGIN WHICH WAS VISIBLE BY THE OUTPUT AVAIABLE.

<https://github.com/HARSH1911RATHORE/ESD_LAB4_PART2>

<https://github.com/HARSH1911RATHORE/ESD_LAB4_PART1>