REPORT Lab :- 4

Introduction :- In this lab, I have interfaced LCD , I2c , DAC and I/O expander with 8051 Micro-Controller.

LCD:- The model of the LCD is HD44780 which has 14 pins. Out of that 7 are data pins which is connected to 8051 pins. The LCD can be interfaced with 8051 in 2 ways

1) With Memory Mapping
2) Without Memory Mapping

In this lab, I have interfaced with memory mapping. Therefore, SPLD , NVRAM comes into the picture. Rest 7 pins of LCD include, VSS, VCC, RS, R/W, E and Gnd. Here we use a potentiometer whose one end is connected to VCC , one end is connected to ground and the middle pin is connected to the VSS of the LCD. Thus by controlling the potentiometer we can increase or decrease the brightness of the LCD.

RS , R/W and E pins of LCD are connected to the SPLD . By controlling the logic of SPLD we can use the memory space of NVRAM and thus write or read to the LCD.

Basically, this LCD has 64 pixels in it, each row contains 16 pixels. There are around 10-15 in total hidden pixels which the company has provided if the user does not want to display it to the screen but wants it hidden in the LCD data.

So first problem and the very basic one was how to initialise the LCD, the understanding of memory mapping took a lot of time of me. One can not just power it up and can not expect the cursor to show up !! There is a whole line of commands which one needs to followed for the initialization.

The table for the initialisation is shown below :- (Reference Ben's LCD Guide)

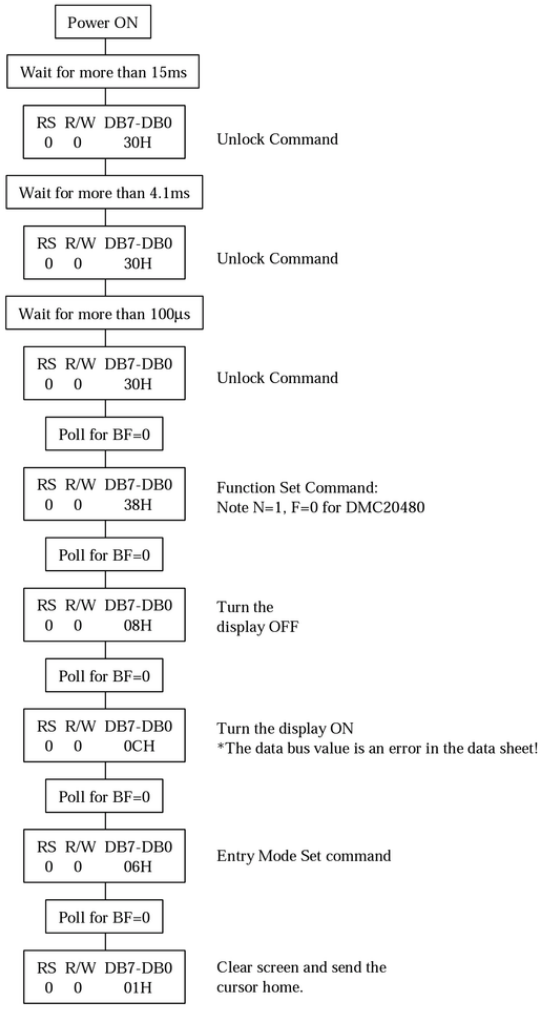
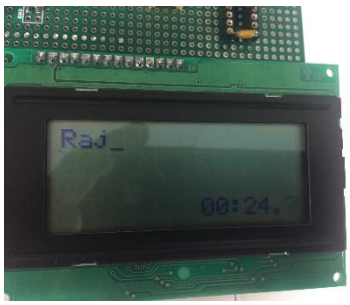


Figure 2: LCD Initialization Sequence (Corrected over old Optrex 20434 data sheet)

After the initialization we can perform several functions in the LCD

1) Print a string :- This function will print the string on the screen on the location where you have initialized your cursor while initialization of the LCD.



2) Print a string at a certain address :- Lets say one wants to print a string at a particular address , then this function can be used.

There are two types of memory location in the LCD

1) DDRAM
2) CGRAM

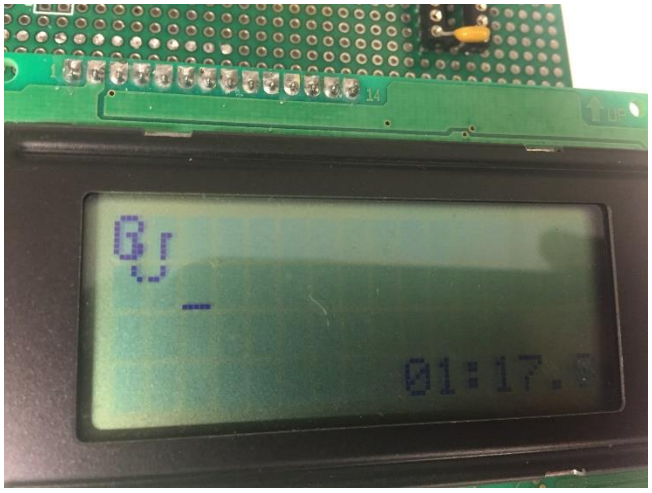
DDRAM has its addresses starting from 0X80 TO 0X8F , 0XCO TO 0XCF , 0X90 TO 0X9F , 0XD0 TO 0XDF in the 1st, 2nd ,3rd, and 4th line respectively.

So if one wants to print at address 0X8D he can move the cursor there and print the string.

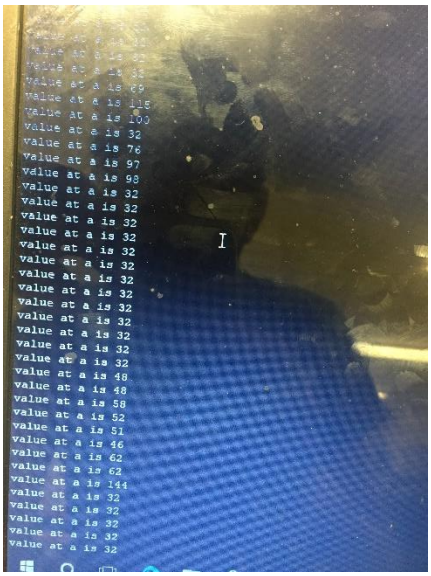
3) Go to location with the help of coordinates(x,y) :- If a person does not know the addresses of the LCD and wants to go at a certain location and print a string then he can use this function where he can input the (x,y) coordinate where “x” determines the rows and “y” determines the columns.

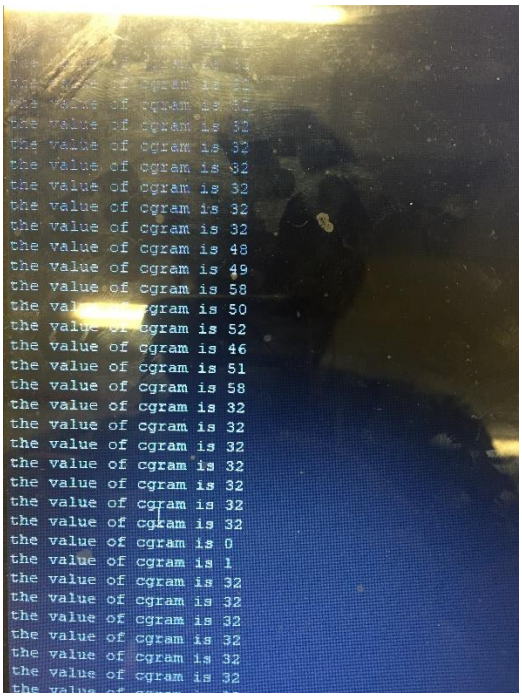
4) Clear the screen :- If one wants to clear the whole screen then he can use this command. By using this function, whatever is written on the screen will get erased.

5) Custom Character :- Custom characters can be created with the help of this LCD. We have to use CGRAM for storing this characters. One can make any design in a pixel and then combine many pixels to draw one custom character. In this lab, I have made this custom character generation in such a way that user can enter the values and the characters get printed on the LCD. So one pixel which consists of 8 rows and 5 columns is controlled by the user.



6) Dump DDRAM and CGRAM :- With the help of this function we can dump DDRAM and CGRAM locations on the UART and check where our characters are stored. This way we can keep a track of all the memory locations and see which is free which is filled up.





I2c:- (24LC16)

The second part of this lab is interfacing the 8051 with I2c. I2c has 3 address lines, A0, A1, A2. There are two other lines which are called SDA and SCL.
SCL:- Serial Clock
SDA:- Serial Data

There is also a write protection feature in this IC. When this pin is tied to VCC then it protects all the array from 000-7FF. There is a also noise protection system in this IC. SCL and SDA lines have Schmitt Trigger and filter circuits which supress noise spikes to assure proper device operation, even on a noisy bus.

I2c is a half duplex meaning it can communicate with master and slave once at a time. The master can be multiple and so does the slave. This particular IC can have 128 slaves at a same time. We also use pull up resistors for this IC because the rate at which this IC can communicate is much slower than the rate at which 8051 communicates. So to maintain the rate of data transfer and to avoid the data loss we use Pull up resistors. We also use decoupling capacitor to avoid noise and to encourage less loss of data.

We can transfer data to I2c and read data from I2c. For that first of all there is Start bit which is followed by a default address which is “1010” which is then followed by the bits A0, A1,A2 and then at the end there is R/W bit. If “0” then it will write and if “1” it will read.
The formation of this 8 bit is called collectively “ Control Byte”.

First of all, Master send the slave this address consisting of 8 bits, after receiving the address the slave sends an ACK bit to the master telling that I have received the address. After which the master sends the “Word Address” to the slave. Again the slave sends a ACK telling the master that it has received the “Word Address”. After receiving the Acknowledgement from the slave , master finally sends the Data to the slave and after receiving the data slave will send an ACK to the master. After ACK is received from the slave, master send a Stop Bit to the Slave indicating that the data transfer procedure has been completed.

Now, if the master wants to send a lot of data without the stop bit , he can do that also. This is called Page write. In this case, data is sent from the master on a continuous basis.

Similarly, there is also a sequential read procedure where the data is read one after the other on a continuous basis. Other function is called random read where the slave sends one ACK bit to the master after every read instruction happens.

So, we can write and read to a particular location starting from 000-7FF.

Reference Datasheet of 24LC16B

FIGURE 4-1: BYTE WRITE

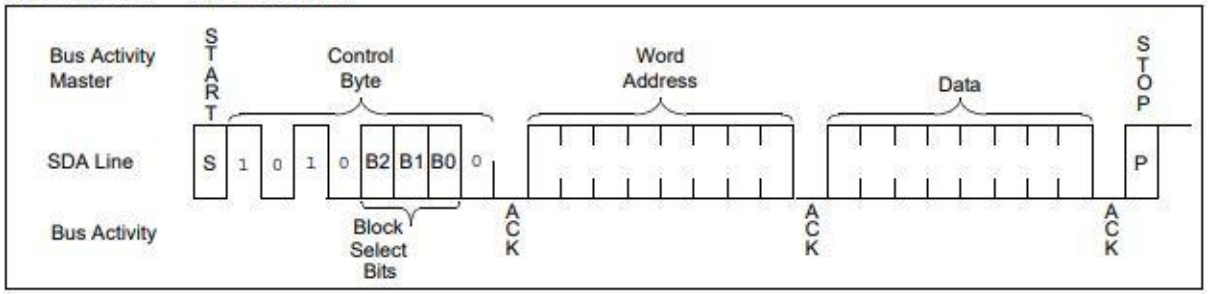


FIGURE 4-2: PAGE WRITE

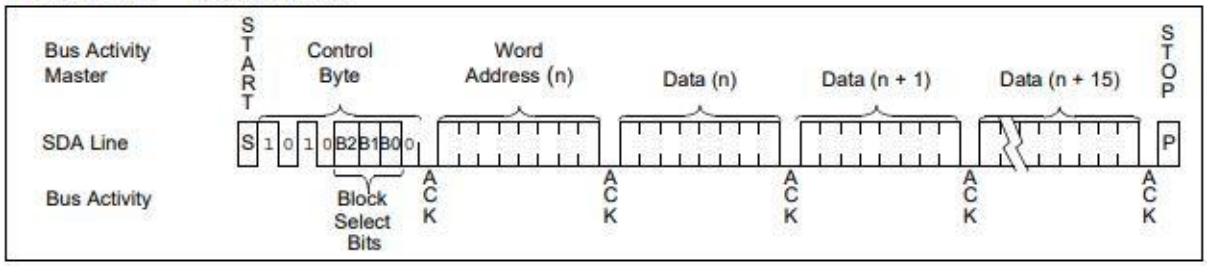


FIGURE 7-2: RANDOM READ

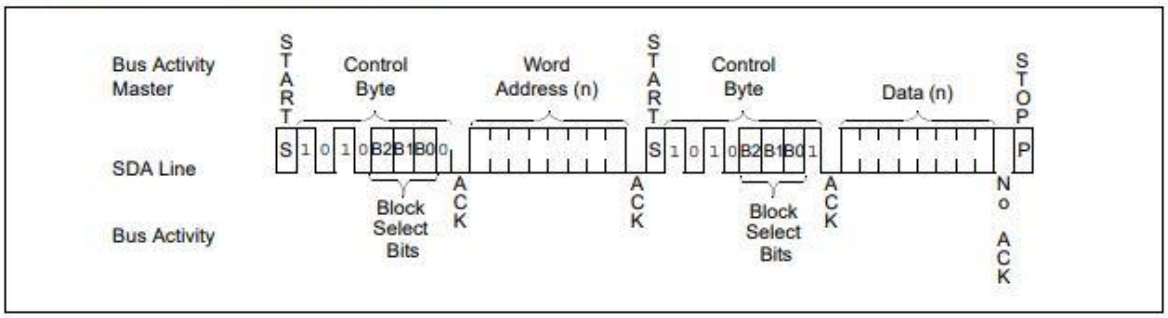
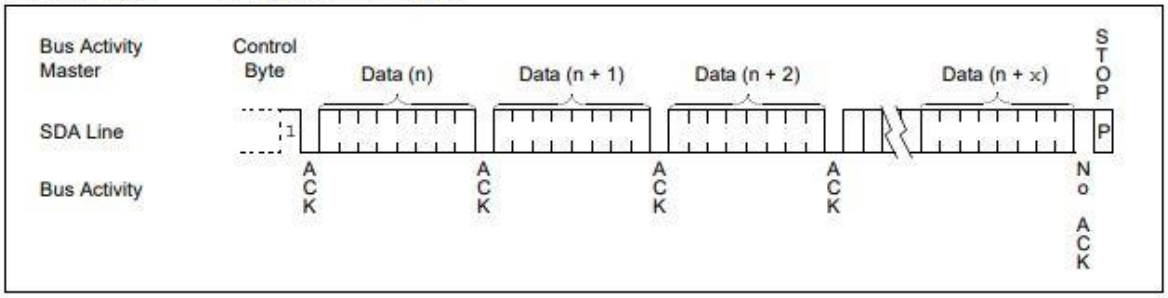


FIGURE 7-3: SEQUENTIAL READ



Clock Stretching :- This is a very particular feature of the slave where the slave has the power to stop the communication or slow it down if the master is sending the data at a very high speed to slave and slave is not able to acknowledge it at that speed. Using this , the slave can slow down the clock and tell the master to stop sending the data.

Cycling Power :- This means that even after power off there is no data loss, if the data sending procedure is stopped in the between , still after power up it will start from the same status.

Clock :- The clock runs from 00.00.0 where the last consists of mili Seconds, seconds and minutes. It starts as soon as the lcd is initialized. We can stop it, start it again and also restart it according to the user interface. Moreover, I have placed in such a way that while writing anything on the clock the clock timing is not overlapped. It is accurate upto 1sec.

Here, we can work simultaneously while the clock is running , for example we can print a string , print a custom character etc. while in the background the clock will still run.

Supplemental Element :-

1) Arm Development Board :-

   I have explained mainly two functionalities of ARM.

   1) LPM :- While, the board is in sleep mode, the voltage level reduces. Here on giving a interrupt we can set the sleep mode off of the ARM and the LED starts to blink. On giving the interrupt for the second time , the board again goes into sleep and the LED stops to blink. This saves power of the board.

   2) AES :- This function is about encrypting and decrypting of the data. Here , 2 blocks each of 8 bit is sent from one end to the other end, one contains data and other contains a cipher key which can be unlocked by the key on other end.
      It has 256 key length. The procedure contains first of all , encrypting the data by loading the cipher key of 256 key length. Then sending the data over another side and when the decrypted data is matched with the original data then the LED blinks. If the data is not matched due to some reason , then the LED wont blink.

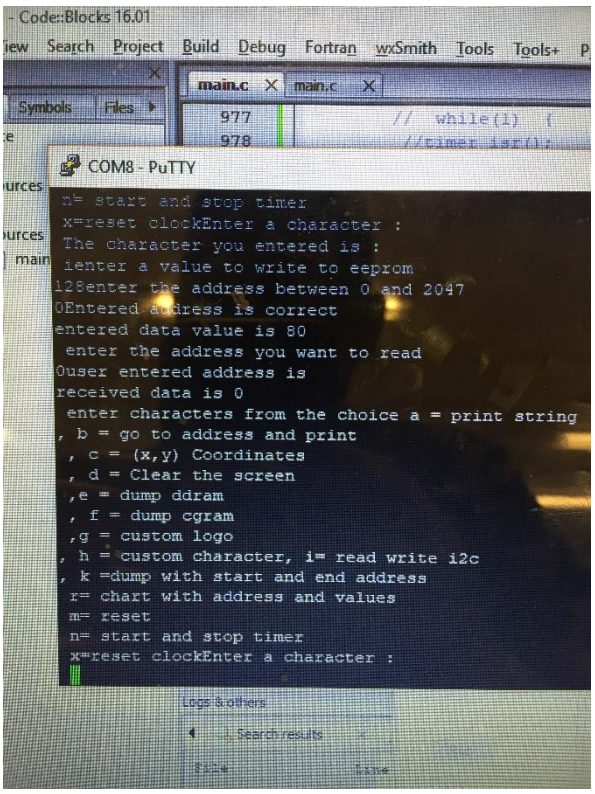
2)PCF8574A I/O expander :-

This expander has 7 Quasi Bi Directional Pins, P0-P7. It has one interrupt pin and it also has SDA and SCL pins which are connected to the I2c 's SCL and SDA pins. The pins can be defined as input or output. The control is in the hands of the user. We can assign data to those output pins.

Here , there are bits initially which are "0111" which identifies the IC PCF8574AN after that we send 3 bits which are A0, A1 and A2 pins which here I have connected to the ground. The next bit is of R/W which when set to 1 is read and when set to 0 will be able to write. Then we have an Ack bit, which has to be sent . After that I have to send the 8 bits of data which determines the i/p and o/p of each pin. For example, if the data is 01, lets say than it can be expressed as 0000 0001 , thus making p7-p1 bits as the output and p1 will be considered as a input.

Since this IC follows the concept of open drain (WIRED AND) the falling edge is determined to check the output. So "1" determines the input and "0" determines the output.

We can set the pins to input and output and then give data to them and can check the functionality by that. This is very useful because we get to create a lot of the extra input output pins by using this IC. There is also a decoupling capacitor which is attached to this IC for filteration of noise.



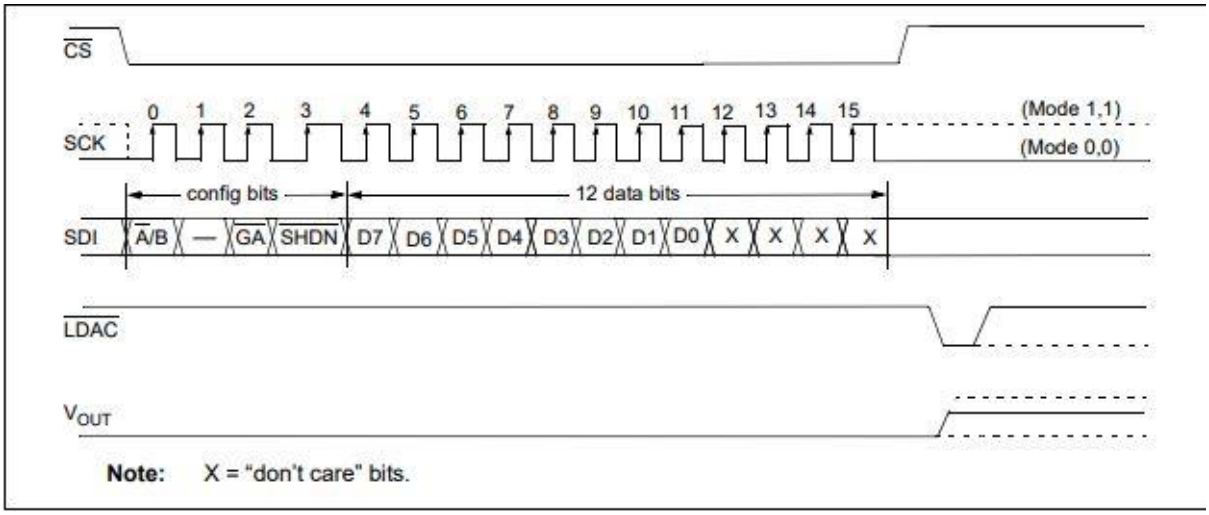
3) DAC:- Digital to Analog Converter :- Digital signal is given by the micro controller to the DAC(MCP4802) vide SPI interface , DAC gets the digital signal and converts that signal to the analog form. Thus we get different waveforms at the end on the oscilloscope. I am also able to change the shape of the waveform by changing the data values.

The structure of the DAC goes something like this. First of all we have to sent the CS bit to 0 since it is active low. After that we have to set the SCK high first and then after the data is sent which is of 12 bits the SCK pin is again set to low.

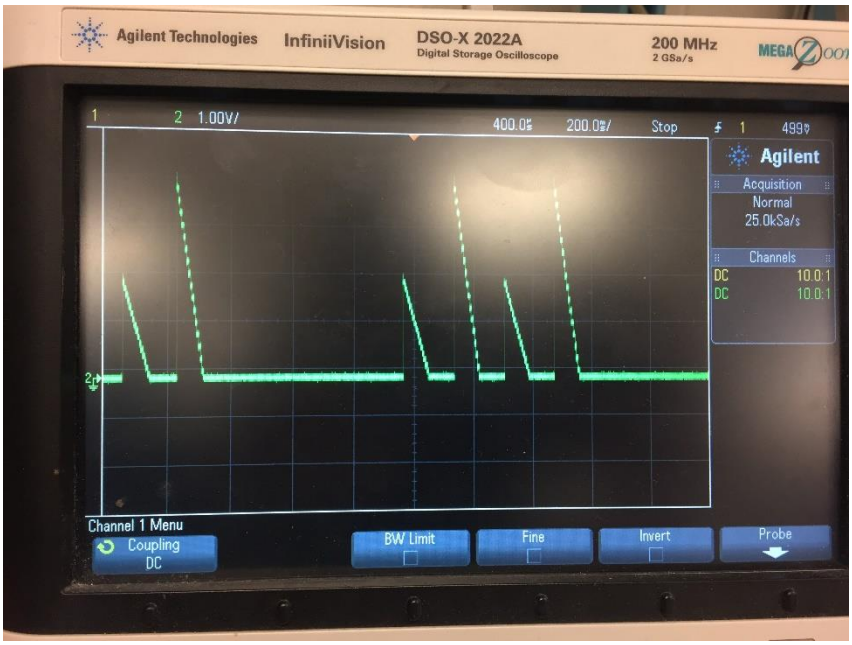
There are total 16 bits out of which first bit is for the output DAC(a) or DAC(b) , if its 1 then its DAC(b) and if its 0 then its DAC(a). The second pin is don't care. The third pin is for the gain. If you set that pin to 1, the gain will be 1x and if you set the pin to 0 the gain will be 2x. Fourth pin is output shutdown control bit , which when set to 1, we get the output in the analog wave in the end and if its set to 0 then it will shutdown the selected channel.

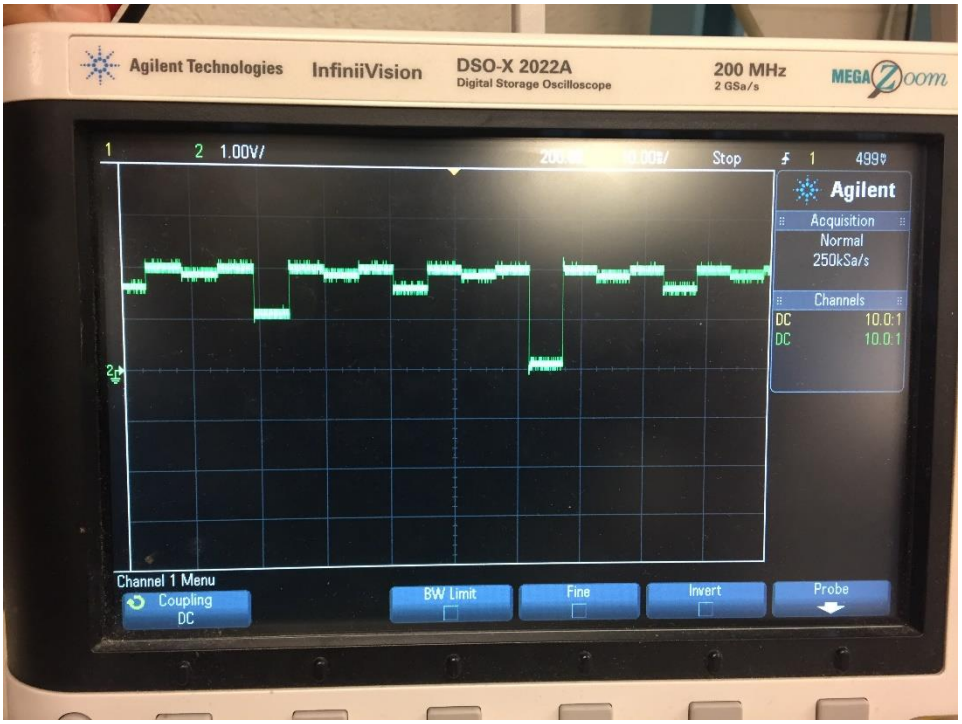
The next 8 bits are for the data which you need to select and the last 4 bits are again don't care.

The write procedure is something as follows :- (Reference DAC MCP4802 datasheet)



Note: X = "don't care" bits.





SPI interfacing :- Serial Peripheral interface is a full duplex communication , where transfer of data can be possible from the both the ends. SPI can not have multiple masters but can have multiple slaves where as I2c can have multiple masters and multiple slaves.

Both SPI and I2c have synchronous communication modes which means both require a clock to function. The SPI which we are using in this lab, is a 3 wire communication interface.

For SPI communication there is a master and slave. The SCLK , MOSI, MISO are common for all the multiple slaves but SS pin signal is different for all the slaves. MOSI means Master out slave input which means Master send the data to the slave and slave listens to it where as in MISO , i.e master in slave out , this means that master will listen to what slave will send it.

There are 2 transfer modes in SPI

CPOL = 0,1 (SCK low and high in idle state)

CPHA = 0,1 (Data samples at leading and trailing clock edge)

The main advantage of SPI is it has high data transfer and it consumes very low power as compared to I2c.