Online Data Acquisition and Offline Download

Group No: D6
Rahul Dalia (04D07032) <rahuldalia@ee.iitb.ac.in>
Krishnendu Saha (04D07033) <krishnendu@ee.iitb.ac.in>
Amol Thuley (04D07035) <amol@ee.iitb.ac.in>

Supervisor: Prof. H. Narayanan
Guide: Prof. S.V. Kulkarni

Abstract

Partial Discharge is one of the main reasons in the failure of high voltage equipments. Partial discharge is name given to electrical discharge involving only a portion of dielectric between two electrodes which does not bridge gap. Therefore it is required in high voltage transmission lines to measure the partial discharge. Partial Discharge occurs whenever the local stress exceeds the breakdown stress which may eventually lead to failure of the material. The values of partial discharge measured need be stored in the computer for the offline analysis. Our project is part of the experiment (whose aim is to measure the partial discharge), where we need to acquire the sampled data coming from an ADC into a memory stack. Once the complete data acquisition is done it is transferred to the computer through USB port.

Problem Statement

To design a microcontroller based system which takes input from a very fast ADC. This data is then stored in a large memory stack. Once the data acquisition is done the data from the memory is downloaded in a computer for later analysis. The transmission of the data from the system to the computer is done through USB.

1) Introduction

The Electronics Design Project involves the acquisition of 11-bit data which is coming at a high speed (1-10 MIPS) from an ADC. The data from the ADC is stored in Memory. The ADC is interfaced to Memory through a 16-bit counter. This counter provides the address to the Memory unit for the acquisition of each data provided by ADC. Once the complete data acquisition is done,

the data in the Memory is available for offline downloading. Another microcontroller is used to read the data from Memory unit and then it transfers the data serially to next unit which converts this serial data to required RS232 (DB 9) format. This RS232 format data is then converted to USB data using serial to USB converter from which it is downloaded to the computer using the HyperTerminal.

2) Design Approach

The implementation of the hardware for the given problem statement is divided into a number of modules. The main hardware modules into which this whole system is divided are:

- Simulation of high speed ADC using a Microcontroller and its interface to Memory unit.
- Memory interfaced to another microcontroller for reading the data
- Microcontroller interfaced to Serial to USB Converter through RS232 circuit.

The block diagram for the complete system is:

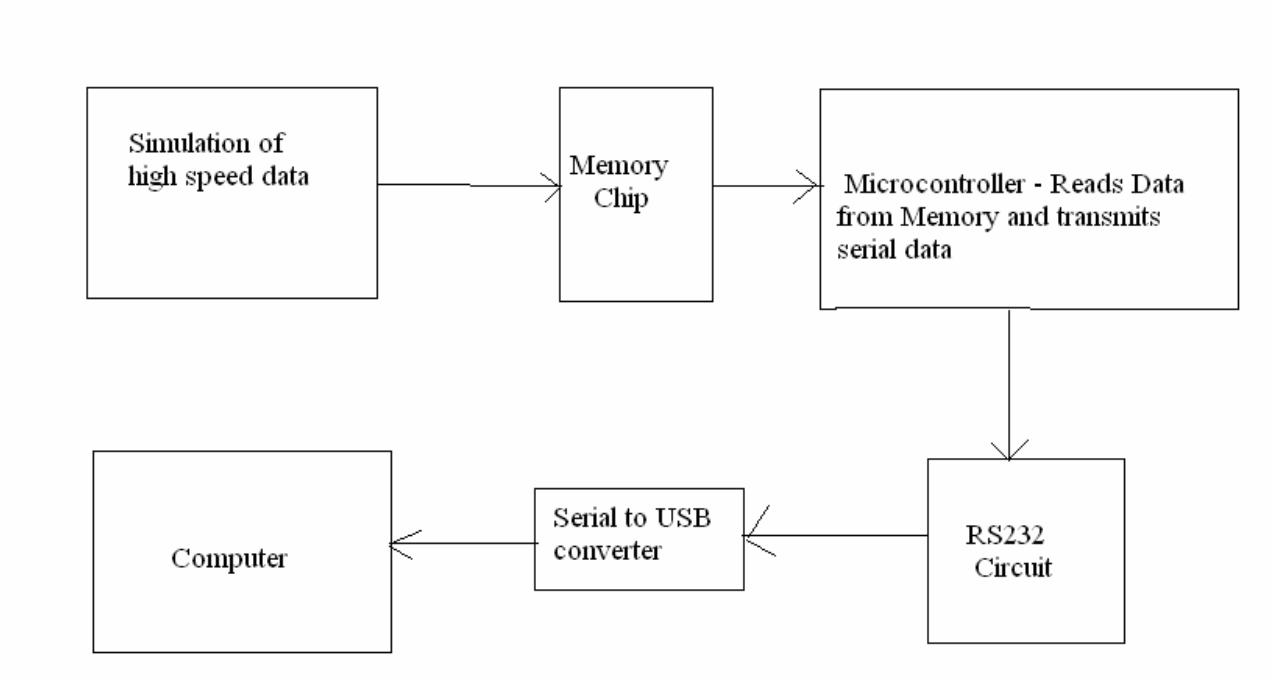

Fig1. Block Diagram of the project

3) Module Description

3.1 ADC to memory interface using microcontroller

The **ADC** provides an 11-bit parallel data and the required control signals for writing the data in Memory Unit through a data buffer and a control signal buffer respectively. Using one of these Control signals the 16-bit counter is incremented by 1 whenever a new data has to be written in the

Memory. This latches the address and then the data is also latched to Memory unit. After this write control signal is given to Memory unit which writes the data in the Memory unit.

Since the ADC provides the data at a very high rate so the memory that has to be used must have very low access time and must be able to write data at a very high speed. This constrained us to use any kind of flash ROM available in the market and hence we have used SRAM. The memory used is 8-bit SRAM (628128). Since the input is an 11- bit parallel data, we have used two RAMs in the memory unit which share the same address bus and control signals. Hence the data is written simultaneously into these two RAMs, the first 8 bits in the one RAM and the last 3 bits in the other one. Since at present we are simulating the ADC part, so the data and control signals are provided by a microcontroller. To simulate such a high speed part using a microcontroller we need a high speed microcontroller. However because of unavailability of high speed microcontroller to us, a common microcontroller AT89C52 is used.

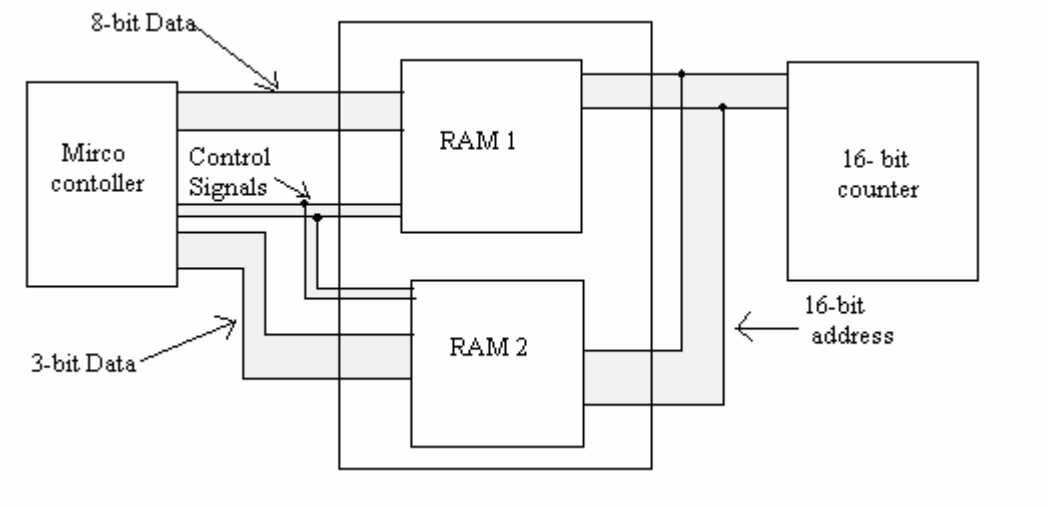

Fig.2 Block Diagram for the simulation of 11-bit data (Data writing Board)

3.2 Memory interfaced to another microcontroller for reading the data

The Reading section consists of a microcontroller (AT89C52), a latch (74573) and two bidirectional buffers 74245. This unit reads each data from RAM and then sends this data serially to RS232 circuit. The reading part is done by internal Intel architecture of 8051 for reading data from external RAM which uses the MOVX instruction. Port P0 is multiplexed to provide the lower address as well as to read the data from RAM. For this purpose latch 74LS573 is used. The latch enable signal is provided by the ALE pin of the 89C52 microcontroller. The higher address is provided by the Port 2 of the microcontroller. Since this design allows to read data from only one RAM at a time, we have multiplexed the data bus as well. Two bidirectional data buffers are used for this purpose. The reading is carried out from one RAM at a time and is switched to the other

RAM once all the data stored in the first RAM is read completely. The Microcontroller reads each byte of data from RAM and then transfers it serially to the RS232 circuit. Since the data is downloaded offline so there is no speed constraint and a standard clock frequency of 11.059 MHz is used for the compatibility of serial port output. The baud rate is set to 9600 b/sec.

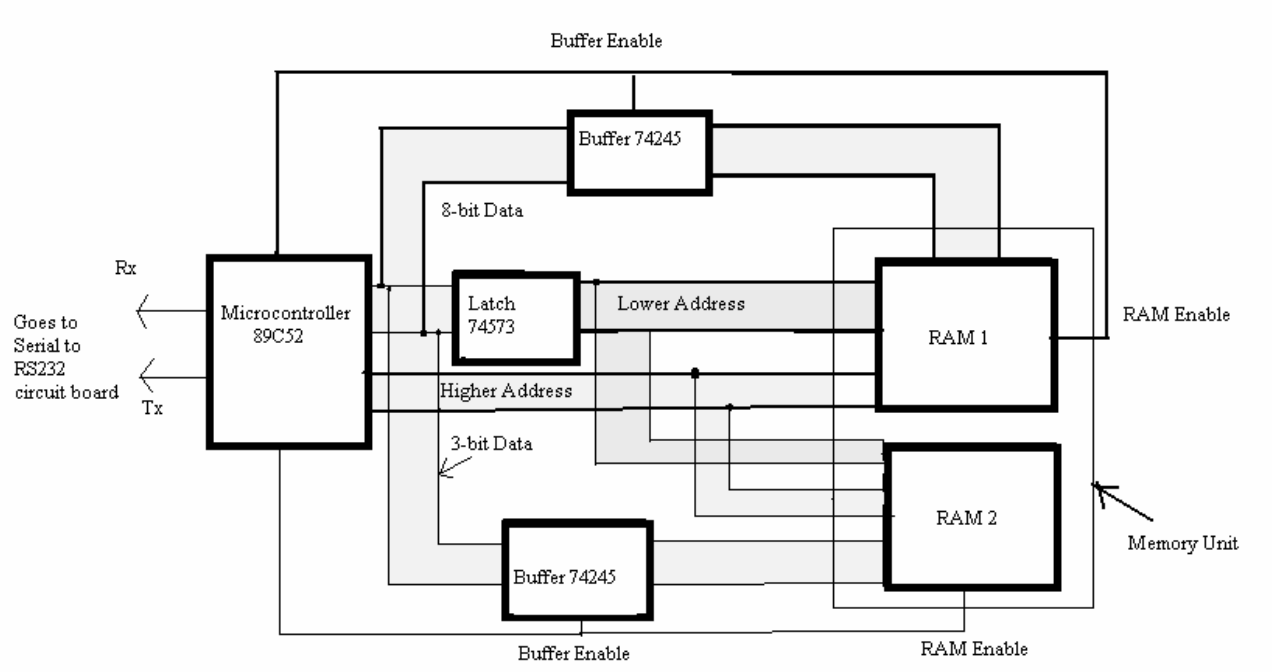

Fig.3. The diagram for memory interfacing with the Microcontroller and transmission of serial data by the Microcontroller (Reading Board)

5.3 Microcontroller interfaced to Serial to USB Converter through RS232 circuit

This module changes the serial data from the microcontroller into the standard RS232 format. After getting the data in RS232 format it is required to change the data in the USB form. This is achieved by using a serial to USB converter. The USB B-type connector is connected to the one side of the serial to USB converter and on the other side its serial port is connected to the RS232 chip. The A-type USB connector is plugged into the computer. The data is received in the computer by using the HyperTerminal.

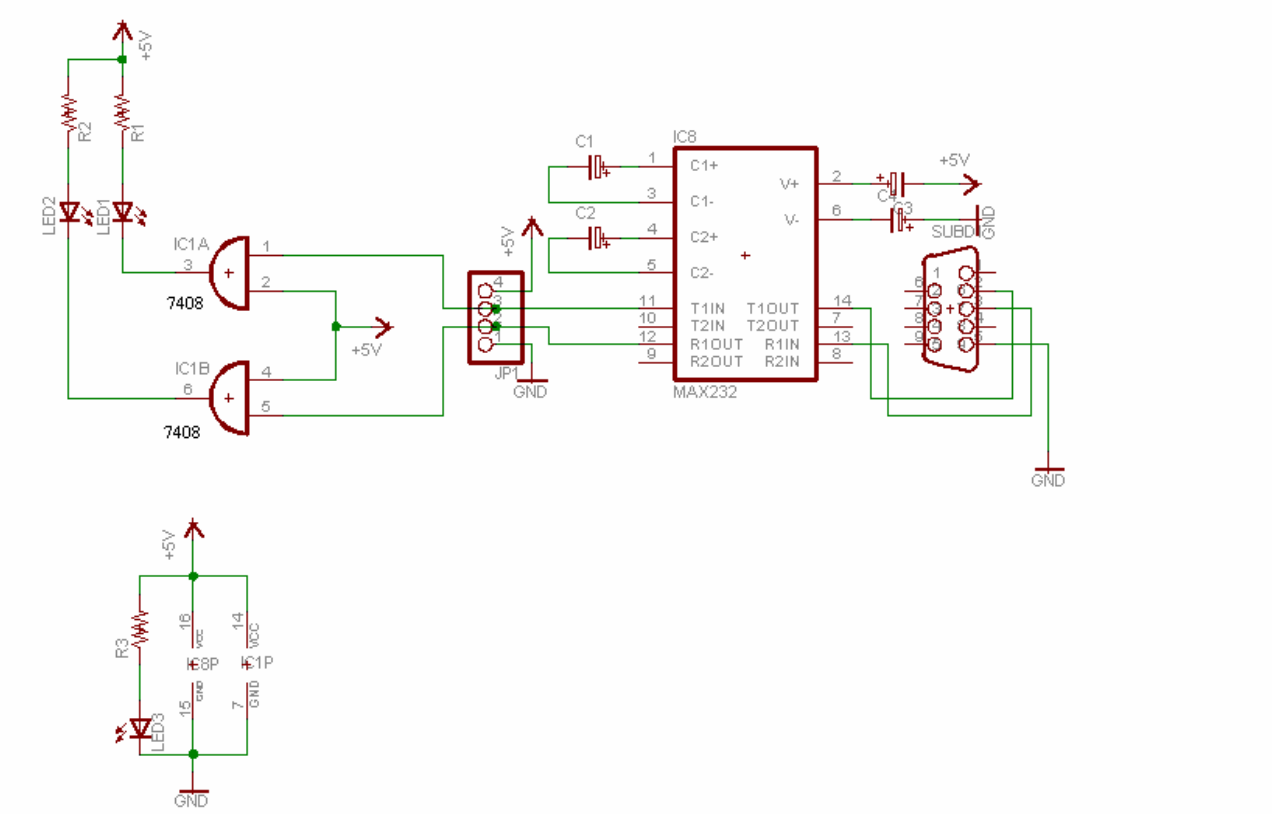

Fig 4. Schematic for RS232 chip

4) **Integration of all the parts into one system**.

Since the problem statement aims at online data acquisition and offline download, we have designed the acquisition and download part on two different PCBs, as only one of the two is going to be used at a time. While acquiring data, the memory board is mounted on the data writing board. The data writing board takes 16-bit input from another system which is being designed by a Research Associate under Prof S.V. Kulkarni. This 16-bit input consists of 11-bit data, a clock signal (for driving the counter), read and write signal (for the memory) and two power supply signals. Once the data is stored in the memory, the memory board is unmounted from the data writing board and is available for offline downloading. For offline downloading, the memory board is mounted on to the 'reading' board and from the reading board data is transferred to the computer through serial to USB converter. The RS232 circuit board is connected to T_x and R_x pins of microcontroller on one side and serial port of serial to USB converter on the other side. Since SRAM is used in the project, so keeping in mind the volatile nature of the SRAM data, the memory board has been powered by an on-board battery.

5) PCB Layout

We have used four circuit boards in our system.

- **Data Writing Board:** This board has four 4-bit counters to provide the address to the memory. It takes 16-bits input. This board is provided with the two 18-bit female connectors to mount the memory board.
- **Reading Board:** This board has a microcontroller 89C52, a latch 74573 and two bidirectional buffers 74245. It also has two 18-pin female connectors to enable mounting of the memory board. In addition to this, a 4-pin connector has been provided to connect to RS232 circuit.
- **Memory Board:** This board has two SRAM chips 628128. This board is designed in such a way that while writing data, both chips can be enabled simultaneously but while reading only one is selected at a time. This has been achieved by providing switches. Two 18-pin male connectors are provided for mounting it on the data writing and reading boards. The board has been powered by on-board batteries.
- **RS232 Circuit Board:** This board has an RS232 circuit with MAX232 chip on it. It contains a 4-pin connector to join to the reading board and a RS232 port to connect to the serial port of the serial to USB converter.

6) Software

Since the reading of two RAMs is done separately, there will be two different data Files. A code in C++ is written to merge these two data files into a single data file. The 8-bit data is read from one file and the remaining 3- bit data is read from the other. For accuracy the byte from the second RAM is ANDed with 0000 0111, to get the 3-bit data. After this the two data are merged into an 11-bit data and stored in a FILE.

7) Problems faced

The main problems faced during the project were:

- A high speed microcontroller was required to simulate the ADC part but such a high speed microcontroller in DIP package was not available in the market.
- The memory which was initially planned to use was the SD Card. But SD card are not fast enough to be used. In place of that, volatile SRAM is used which needs to have power supply all the time unless the data is read.

- Since the writing and reading parts use the same address and data lines of the memory, the complete circuit became too complex and big to be implemented on bread board and single PCB. So the whole project was divided into smaller parts and each subpart has been implemented separately on different PCBs.

8) Conclusion

In this project we have successfully designed following part of the project

1. Reading Data from RAM.
2. Sending data serially to computer using hyper terminal.
3. Writing data to RAM at a speed of

However the complete system is not completely functional at present.

9) Further Developments: Since the SRAM is volatile in nature, so it is somewhat difficult to physically disconnect and reconnect the SRAM. One thing which can be implemented is using both SRAM and Flash memory to carry out the whole procedure. Since SRAM is faster, so the data is acquired in SRAM. Once the data is acquisition is done, data from the SRAM can be transferred to flash memory (S.D. Card) which is non-volatile in nature. Then we can physically disconnect the S D Card and can use it for reading purposes later on. In this way we can make use of high speed of SRAM and non-volatile nature of SD Cards.

9) References:

- Datasheets of 89C52, RAM - 628128, Latch – 74573, Buffer - 74245, 4-bit counter – 74191, and MAX232.
- “The 8051 Microcontroller and Embedded Systems” by Muhammad Ali Mazidi and Janice Gillespie Mazidi.