*Project Report*
*On*

# A Microprocessor based Image Scanner

*For*
*The Design Assignment*
*Of*
**Microprocessor programming and interfacing**

*By*
Sankhya Chakravarty (2012A7PS014P)
Utkarsh Pathrabe (2012A7PS034P)
Shikhar Brajesh (2012A7PS114P)


BITS Pilani
25th April, 2014

# CONTENTS



## QUESTION:

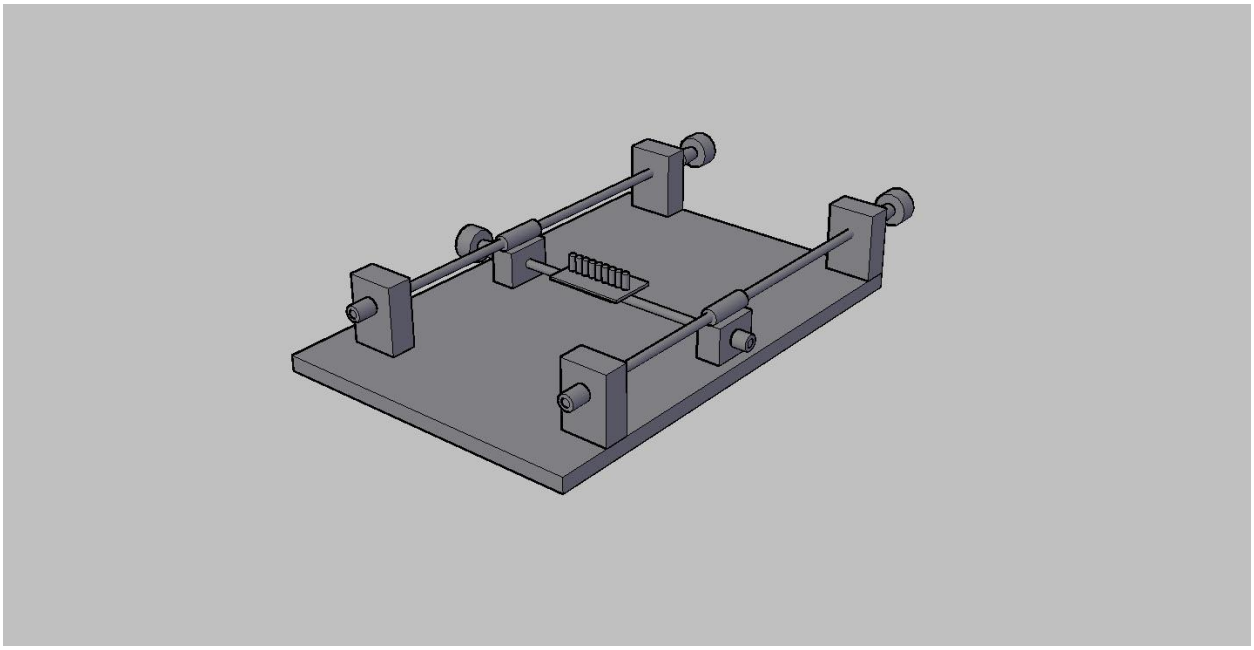
## P7

Design a microprocessor based scanner which will scan a black and white image and store it as binary data. The scanner has two stepper motors for motion along two orthogonal coordinates. The rotational motion is converted into transnational motion through a lead-screw mechanism. Five paired LED photodiodes intended for B&W image scanning are placed one centimeter apart. The maximum size scannable is 10cm X 10cm. The photodiode output is analog signal (between 0 to 5 Volts) which is to be digitized. Image information is stored sequentially in the RAM. The motor moves given direction signal and a pulse. (0 for clockwise and 1 for anticlockwise).

## DESIGN SPECIFICATIONS

The intel 8086 processor is the central processing unit of our image scanner system. Two Intel 8255 Programmable Peripheral Interfacing device are used to interface two stepper motors and a Analog-To-Digital Converter ADC0808. Each of the stepper motors are connected to the 8255 through the High Voltage High Current Darlington Transistor Array ULC2803A. The 5 paired photodiodes are interfaced to the system through the ADC0808. The ALP program which is used to run the device is stored in ROM chips. Two 2716 ROM chips are used for this purpose, one of which work as an even bank and the other working as an odd bank. For storing the image two RAM chips are used each of size 2 kilo-bytes. One of the RAM chips is working as an even bank while the other is woking as an odd bank. The RAM model used is 6116. The two ROM chips which count a total of 4 kilo-bytes of memory are interfaced to the address lines 00000H through 00FFFH whereas the ROM chips are addressed from 08000H to 08FFFH. Three 74LS373 octal latches are used to demultiplex the AD lines coming from the processor. Two 74LS245 bi-directional buffers are also used to demultiplex the AD lines coming from the processor. Some OR and NOT gates are used to generate the RD', WR' and M/IO' signal lines.

The rotational motion of the stepper motors are converted to translational motion by using the lead-screw mechanism. As shown in the diagram below, there are two stepper motors used, one for the horizontal motion and another for vertical motion.



## ASSUMPTIONS MADE WHEN IMPLEMENTING THE DEVICE

Certain assumptions are made while implementing the project. These assumptions are :

1. There is no power failure while the system is operating.
2. The 8086 Chip is already programmed with the specified code from an external source.
3. For one full rotation, both the stepper motors move 0.5 cm.
4. The pixels to be read from the paper are 0.125 cm apart. Therefore there are 8 gaps each of 0.125 cm in one centimeter. Therefore there are 81 points to be read in each row and 81 points to be moved along each column.
5. Size of the image is always 10 X 10 cm.
6. The starting position is from the top-left corner of the page. Motion of the photodiodes would be from left to right and top to bottom of a page in the read operation.
7. The stepper motor requires a software delay of 1 millisecond while it rotates into position.

## COMPONENTS USED

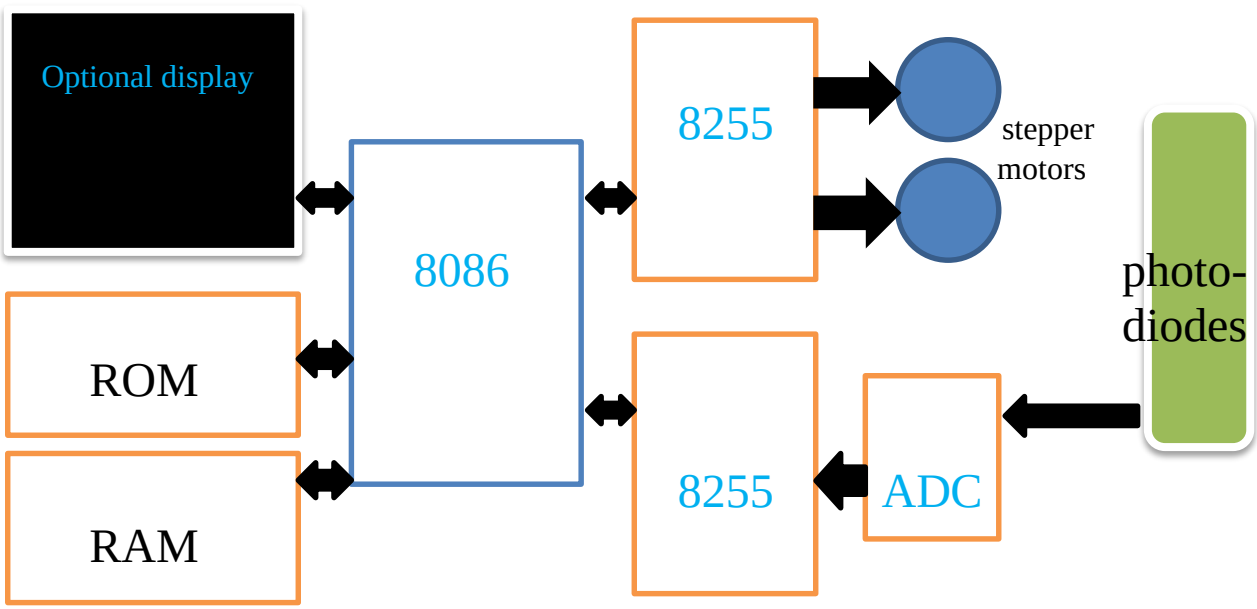
The following components are used in this project. The number of each component used and their purpose is depicted in the table given below.

| Name of the component | Number of the component used | Purpose |
|---|---|---|
| INTEL 8086 | 1 | It is the main processor of the image scanner. All the devices are controlled directly or indirectly through 8086. |
| INTEL 8255A | 2 | This device interfaces the ADC and Stepper motors to the main processor i.e. 8086. |
| ADC0808 | 1 | The main purpose of this device is to convert the analog signal that are coming from the photo-diodes to digital signals so that we can store the binary image in the memory. |
| ULN2003A | 2 | These devices are responsible for the rotation of the Stepper motors. One of the ULN2003A is connected to control the horizontal movement while the other is for vertical movement. |
| ROM2716 | 2 | For storing the ALP program which is used to run the scanner. |
| RAM6116 | 2 | These ram chips are used to store the binary data of the scanned image. |
| 74LS138 | 1 | This device is used to interface the ROM and RAM chips. |
| 74LS373 | 3 | Octal latches for address lines, BHE` |
| 74LS245 | 2 | Bidirectional buffer for data lines. |
| STEPPER MOTORS | 2 | The stepper motors are responsible for the movement along X and Y direction. |

Besides these devices, simple hardwares like AND, OR and NOT gates are used for various purposes which include the generation of IOR, IOW, MEMR, MEMW signals, connecting address lines to the peripheral devices and to control RAM and ROM through 74LS138.

The image sensor is made up of five photodiodes. Physically, these photodiodes would be connected side by side in a board, which would be connected to stepper motors enabling the structure to scan the entire image. But due to the limitations imposed on us by the prescribed Proteus software, we were unable to use photodiodes. Instead, we made the use of torch LDRs.

# BLOCK DIAGRAM

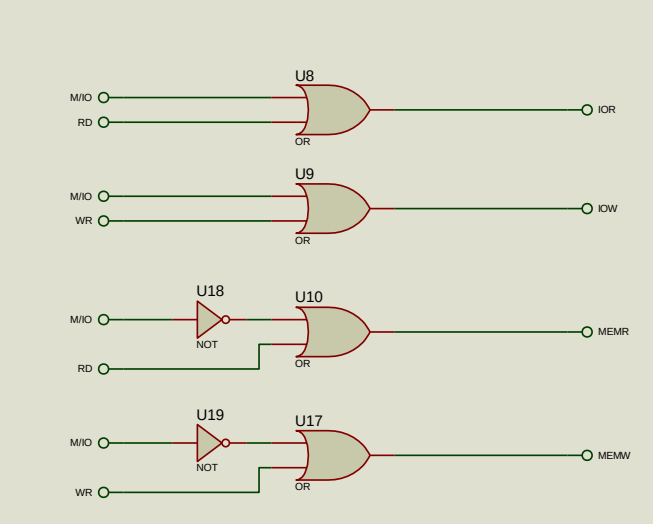



# THE COMPLETE HARDWARE CIRCUIT DIAGRAM

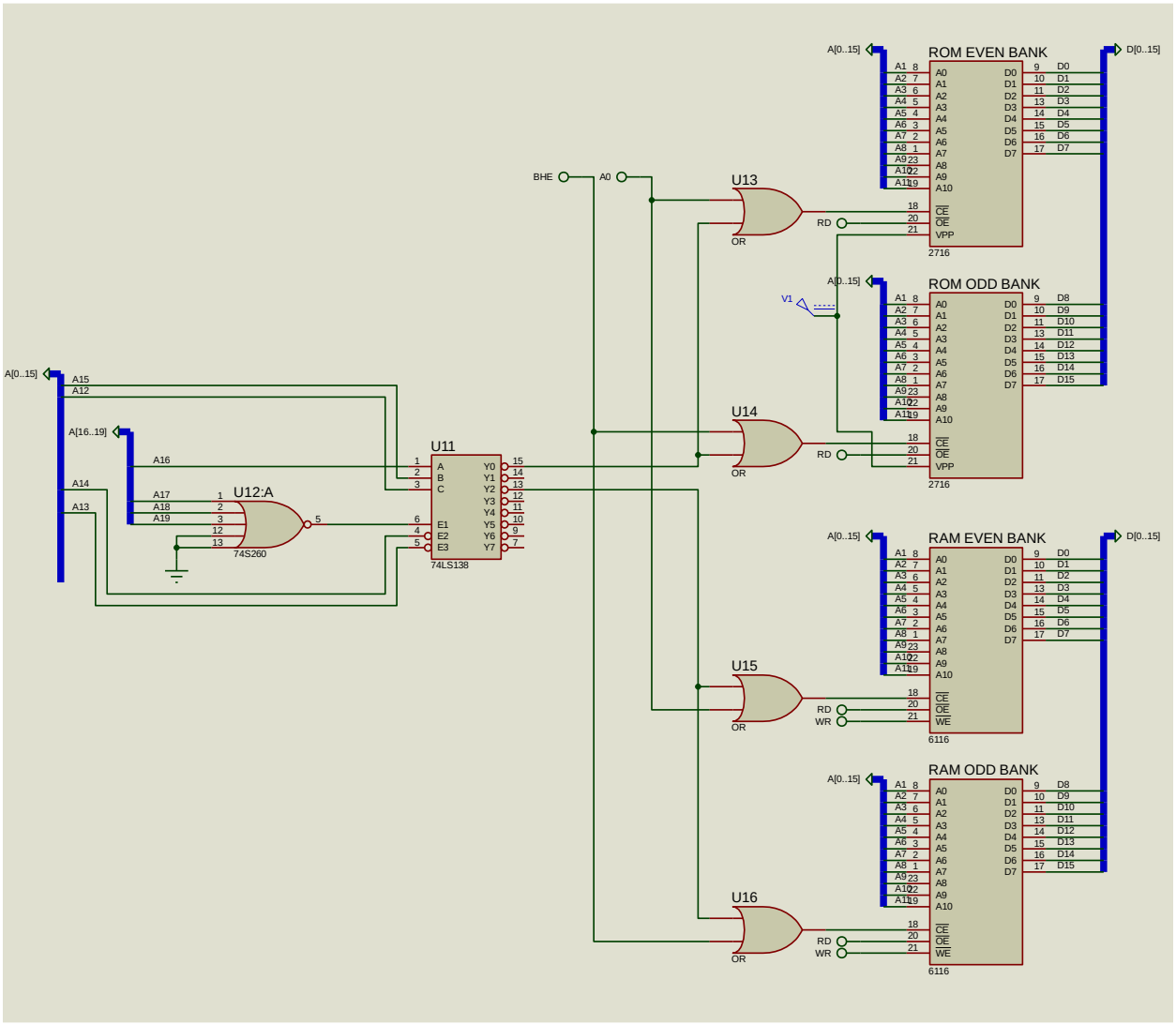
The complete circuit diagram is too big to fit in one page of this PDF document. So, we have shown the entire circuitory in parts. Besides, we have also included a BITMAP file containing the complete circuitory.

The various parts of the circuitory are illustrated below.

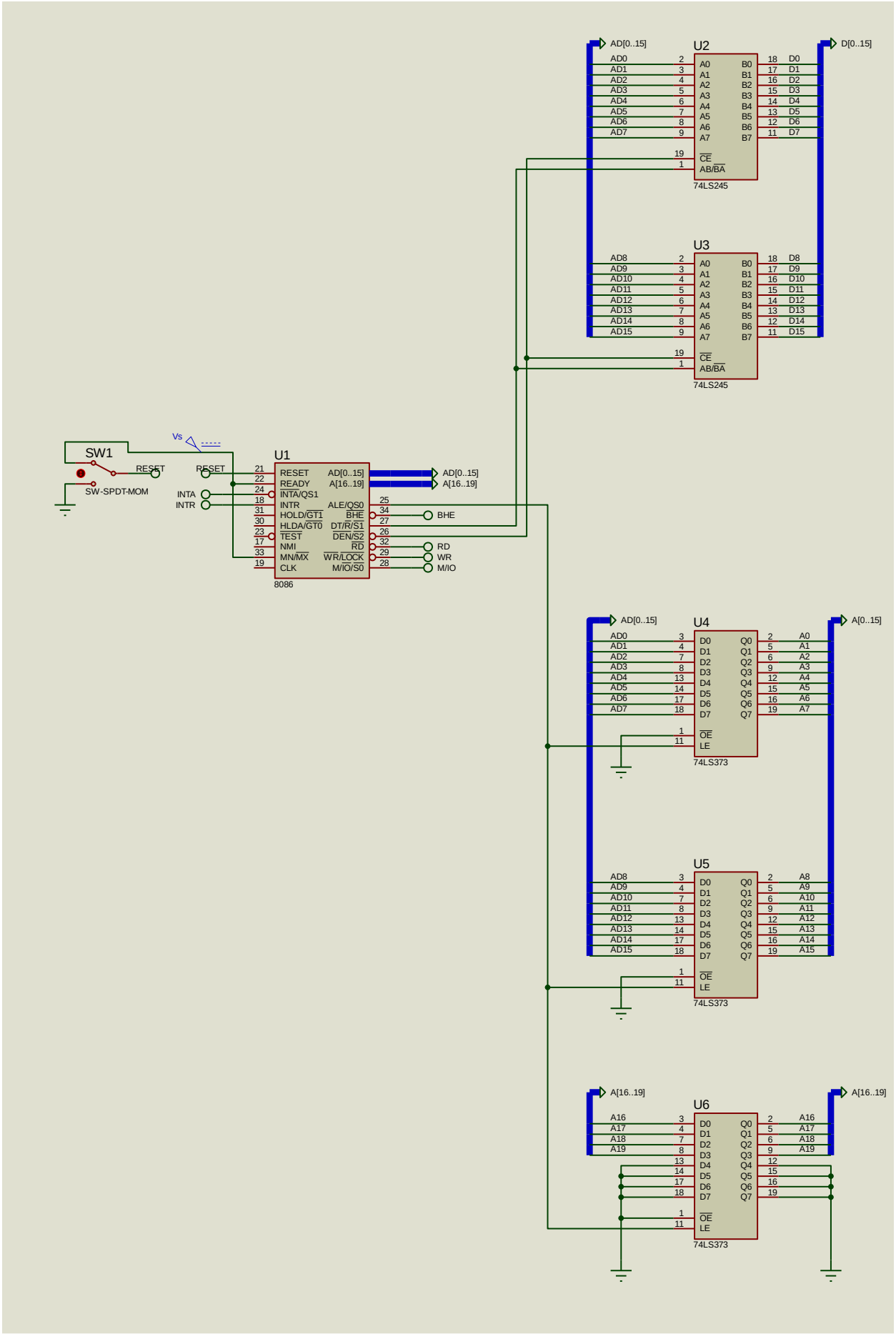
1. Creating IOR, IOW, MEMR, MEMW signals.



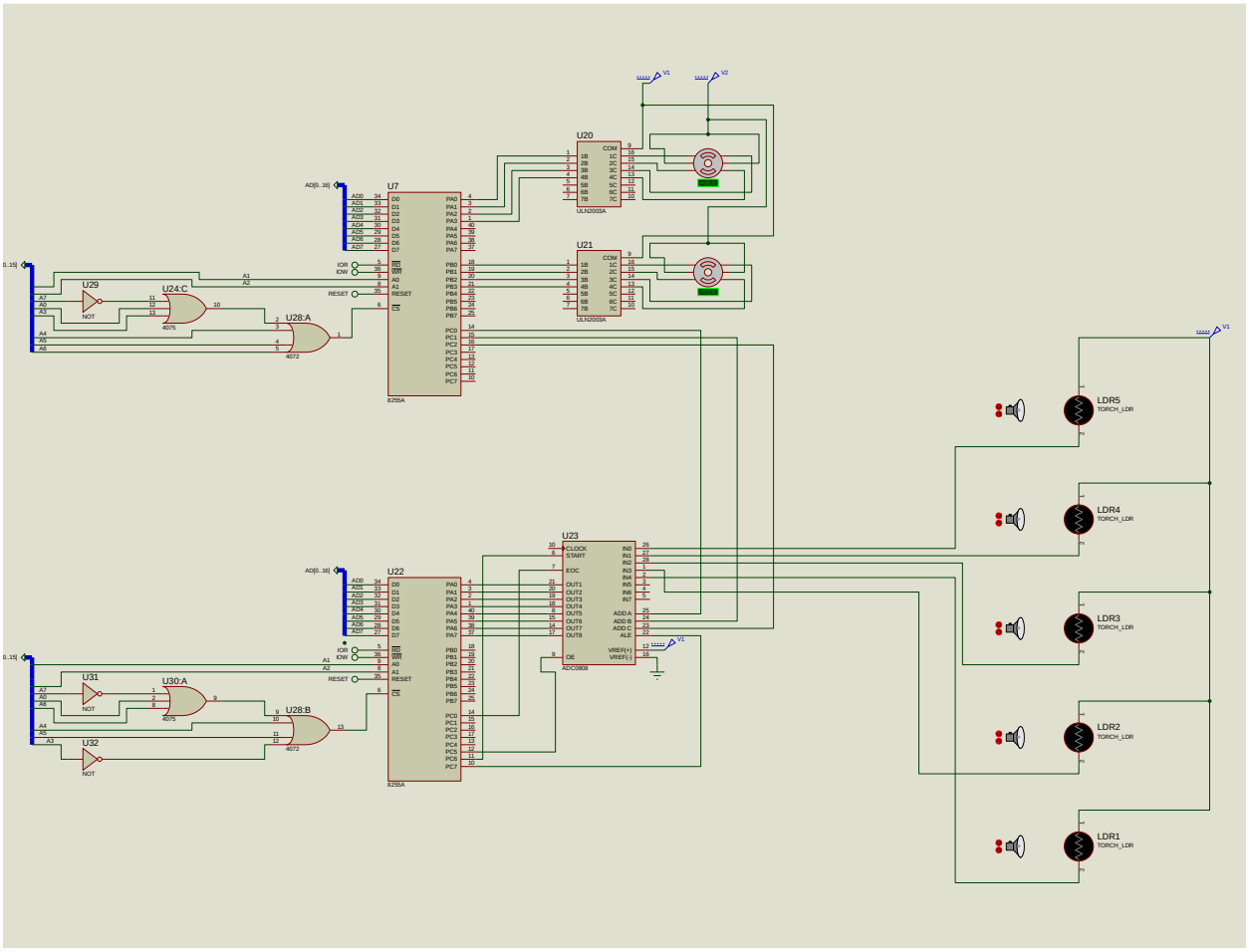
2. Interfacing ROM and RAM.




3. Demultiplexing Address and Data lines.


U1
8086
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
AD[0..15]
A[16..19]
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
SW1
SW-SPDT-MOM
Vs
INTA
INTR
BHE
RD
WR
M/IO
U2
U3
74LS245
U4
U5
U6
74LS373
D[0..15]
A[0..15]
A[16..19]

4. Connecting the Stepper Motors and Connecting the image sensors to ADC.



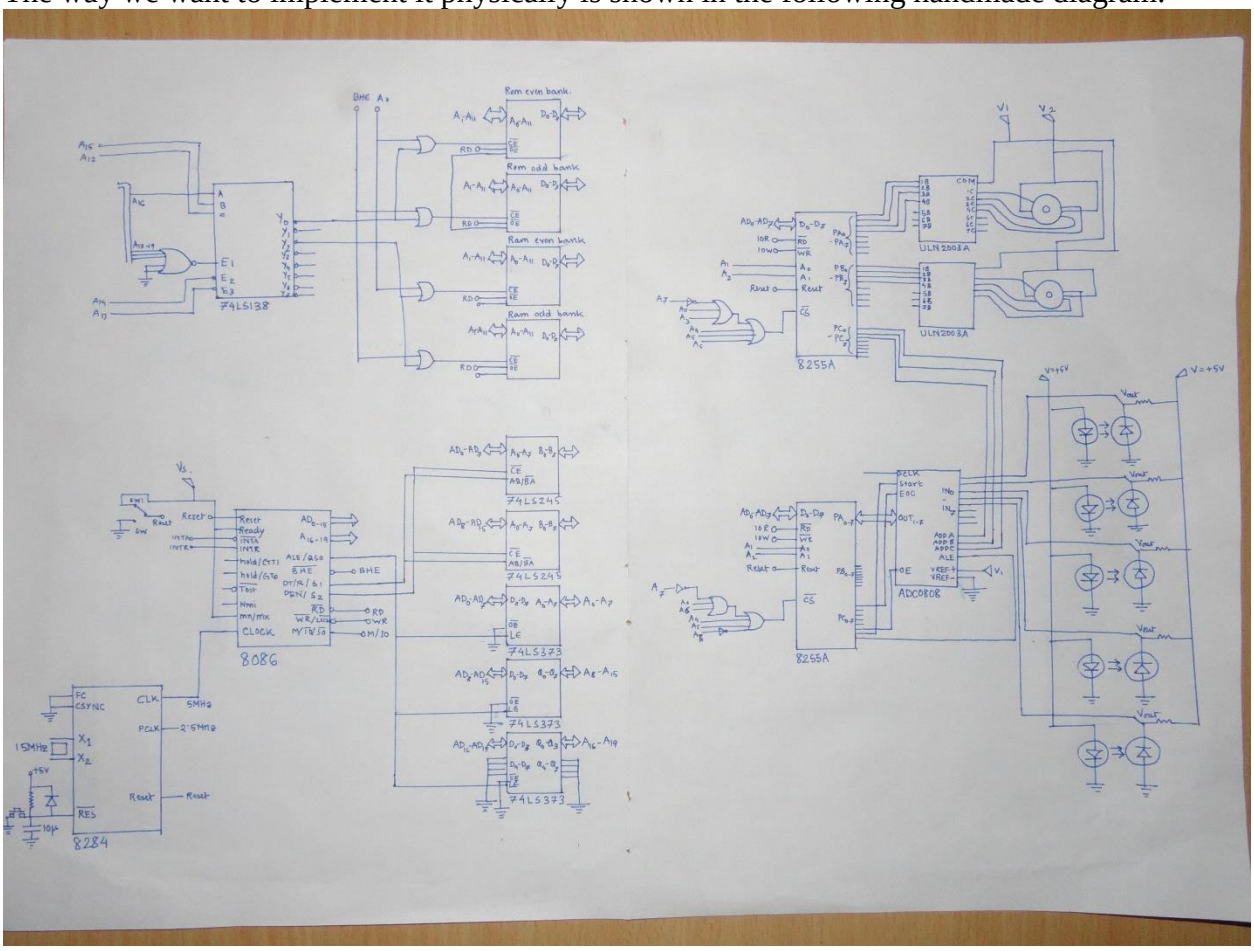

Please note that due to the software limitations in proteus, instead of using photodiodes we have used torch LDRs.

The memory related informations are provided below. These memory addresses are reflected in the ALP program.

| Ports in 1st 8255A | Address | Purpose |
|---|---|---|
| **A** | 80h | Horizontal stepper motor output port. |
| **B** | 82h | Vertical stepper motor output port. |
| **C** | 84h | Lower C ports select ADC lines. Upper ports are not used. |
| **Control register** | 86h | To initialize 8255 ports. |

| Ports in 2nd 8255A | Address | Purpose |
|---|---|---|
| **A** | 88h | Connecting ADC inpu port. |
| **B** | 8Ah | Not used. Initialized as output ports. |
| **C** | 8Ch | Lower ports are used as input ports while the upper ports are used as output ports. |
| **Control register** | 8Eh | To initialize 8255 ports. |

Though we used torch LDRs in proteus, physically we will implement this with photodiodes. The way we want to implement it physically is shown in the following handmade diagram.




This diagram may be a little difficult to see properly in the PDF file. So, we have also provided a .JPG image for this diagram.

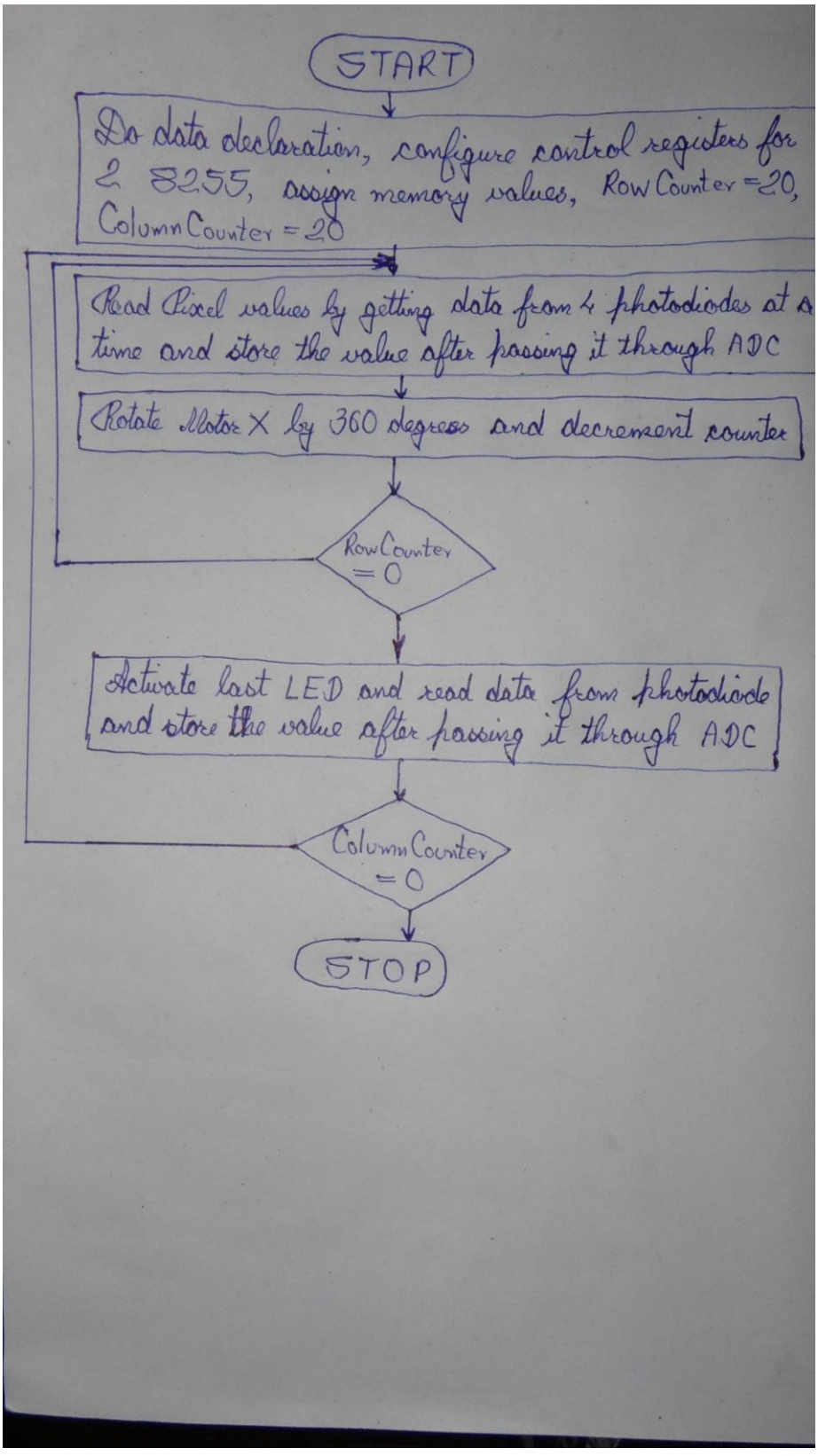
# FLOW CHART OF THE SOFTWARE

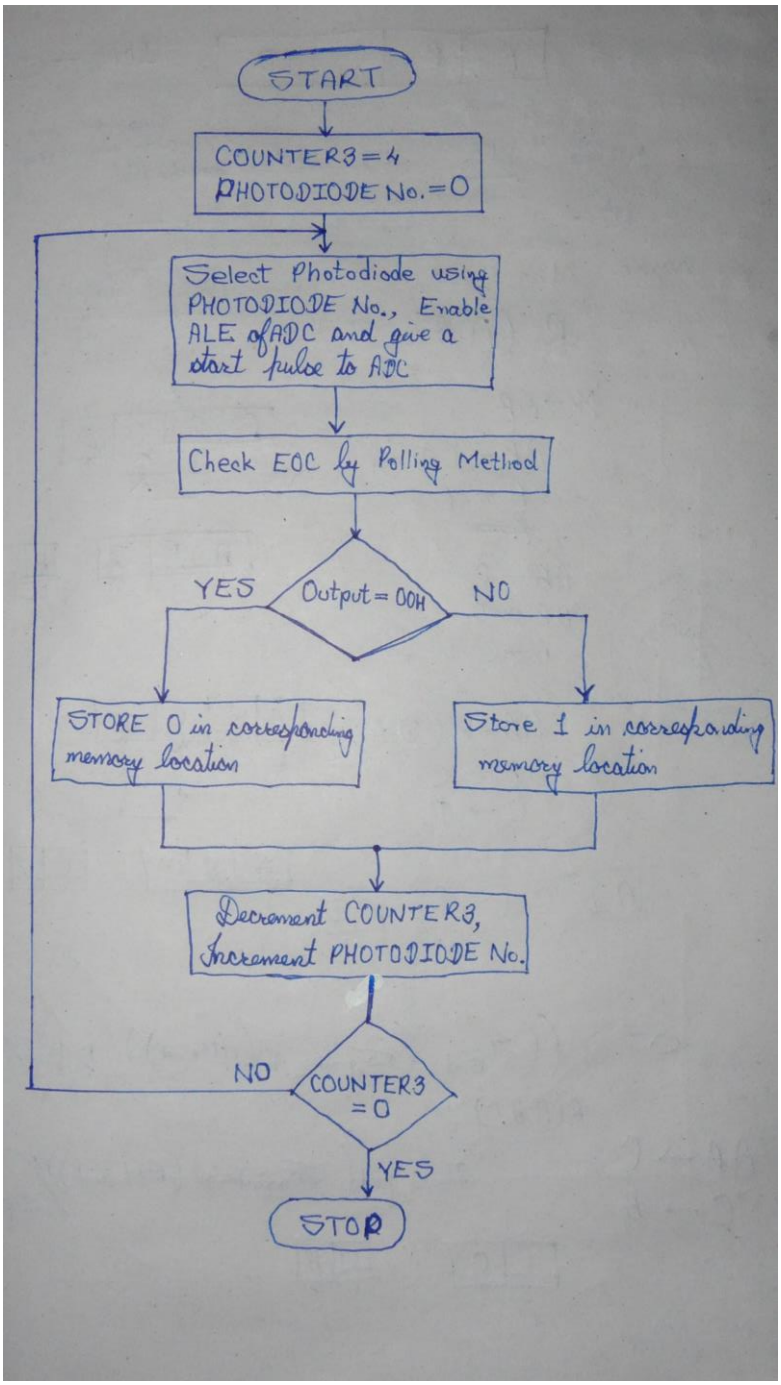
The process of scanning and storing the image into memory is described below step by step.

1. First, the image which is to be scanned is inserted. LED torches are used for lighting purpose. The photo-diodes capture the intensity of light which is passed through the image and generate corresponding analog current signal. For example, if the light is falling in that part of the image which is whitish ( white, cyan, light green etc. ) more light would transmit through the image and will cause a high analog signal. The opposite will happen when the light is incident on the dark part ( black, deep blue, deep green etc. ) which causes a low analog signal.
2. The array of photo-diodes are moved in vertical and horizontal directions with the help of two stepper motors to cover the entire image.
3. The analog signals thus produced is fed to ADC0808 which converts them to digital signals or more appropriately a string of binary digits.
4. The digital signal produced by ADC is sent to 8086 through the interfacing device 8255.
5. Finally, 8086 stores the image in the RAM.
6. The 10x10cm image is divided into pixels of size 0.125 cm each. Each centimeter is divided into 8 parts so that each part takes up one bit of a byte, the entire byte itself representing one centimeter. In this way, bit 0 would be the starting position and bit 7 would be the last position within a centimeter.
7. The memory has already been initialized as 0's. If a diode is read as 5V, then 1 would be written into the corresponding bit else no operation as already initialization as 0's.
8. In a row the last point is read separately by choosing only the required photodiode and this bit is stored separately in the next byte. In this way, one row consisting of 81 points would occupy 10+1 bytes.
9. The entire photodiode system is moved back to the start at the left and then vertically downwards by 0.125cm and then the read operation commences as from step 6.
10. The entire page therefore comprises of 81 columns and 81 rows. Each row taking 11 bytes would imply that the entire page would take 11x81 = 891 bytes of memory for storing the image read.

The flowchart is shown through the following diagrams.

1. The main flowchart.
2. Flowchart for readpix function.


START
Do data declaration, configure control registers for 2 8255, assign memory values, Row Counter = 20, Column Counter = 20
Read Pixel values by getting data from 4 photodiodes at a time and store the value after passing it through ADC
Rotate Motor X by 360 degrees and decrement counter
Row Counter = 0
Activate last LED and read data from photodiode and store the value after passing it through ADC
Column Counter = 0
STOP


START
COUNTER3 = 4
PHOTODIODE No. = 0
Select Photodiode using PHOTODIODE No., Enable ALE of ADC and give a start pulse to ADC
Check EOC by Polling Method
YES
Output = 00H
NO
STORE 0 in corresponding memory location
Store 1 in corresponding memory location
Decrement COUNTER3, Increment PHOTODIODE No.
NO
COUNTER3 = 0
YES
STOP

*Note that the following files are also included in the compressed zip file.*

1. *The DSN file of proteus implementation.*
2. *The ASM file that contains the program.*
3. *Two .JPEG images showing the complete hardware implementation, one hand-drawn while the other is a snapshot from proteus.*