Introduction to 8086 Microprocessor Based Computer MDA-8086 Kit

Birzeit University
Information Technology Faculty
Computer Systems Engineering Department

August 23, 2010

Abstract

This experiment aims at introducing a general overview about Intel 8086 Microprocessor Based Trainer Kit "MDA-8086" and its WinComm software. This Experiment will be a brief manual for using the MDA-8086 Kit during other experiments in the Computer Design Lab.

Contents

Part I
Technical Introduction

1 Getting Started With MDA Kit

MDA-8086 Kit will be the main tool used in ENCS 511 Computer Design Lab. This kit is based on Intel 8086 Microprocessor. It contains many IO devices and components connected and gathered together. In the Technical Introduction PartI, Kit Basic operation and instructions will be introduced. It is an essential task of your In-Lab practices to go over this part before going to the Practices PartII. Also it is highly Recommended to keep this experiment as a reference during next experiments of the lab.

1.1 MDA-8086 Components Overview & Address Mapping

Figure1 shows the kit schematics. This kit has I/O facilities and built in interfacing devices such as 8255, 8251, 8259, etc. Main components build the kit are: (**As your first practice try to locate each part on the schematics figure, match that with the real Kit you have in the Lab)**

- **CPU**
  - Intel 8086 System Clock 4.9152 MHz
- **Main RAM**
  - SRAM 64 KB Input user's program & data.
  - Address of memory is 00000H ~ 0FFFFH, totally 64K Byte . (62256 x 2)
- **Monitor ROM**
  - It has program to control user's key input, LCD display, user's program.
  - 64K Byte, it has data communication program.
  - Range of ROM Address is F0000~FFFFFH.64 KB
- **Display Unit**
  - LCD (16 x 2 Line)
- I/O Port
  - 3 x 8255A (PROGRAMMABLE PERIPHERAL INTERFACE PPI)
- **Serial Port**
  - RS-232C (8251A x 1): It is ready to do data communication with IBM compatible personal computer.
- **Clock Generator**
  - 8284 Chip
- **Interrupt Controller**
  - 8259 Programmable Interrupt Controller
- **Level Meter**
  - 10 Steps
- **DOT Matrix**
  - 8 x 8 (3 Color)

- **Operation System Software**
  - 8086 Assembler
- **Keyboard**
  - 16 Key of Data, 10 Key of Function
- Expansion Connector System Bus 62pin x 1 External Interface 20pib x 1
- Step Motor Interface Driver T.R x 4 to control stepping motor, driver circuit of stepping motor is interfaced.
- **A/D Converter**
  - Convert analog signal to digital signal using ADC 0804
- **D/A Converter**
  - Convert digital signal to analog signal using with DAC 0800 and it is interfaced so as to more Level meter.
- **Power** 110V/220V

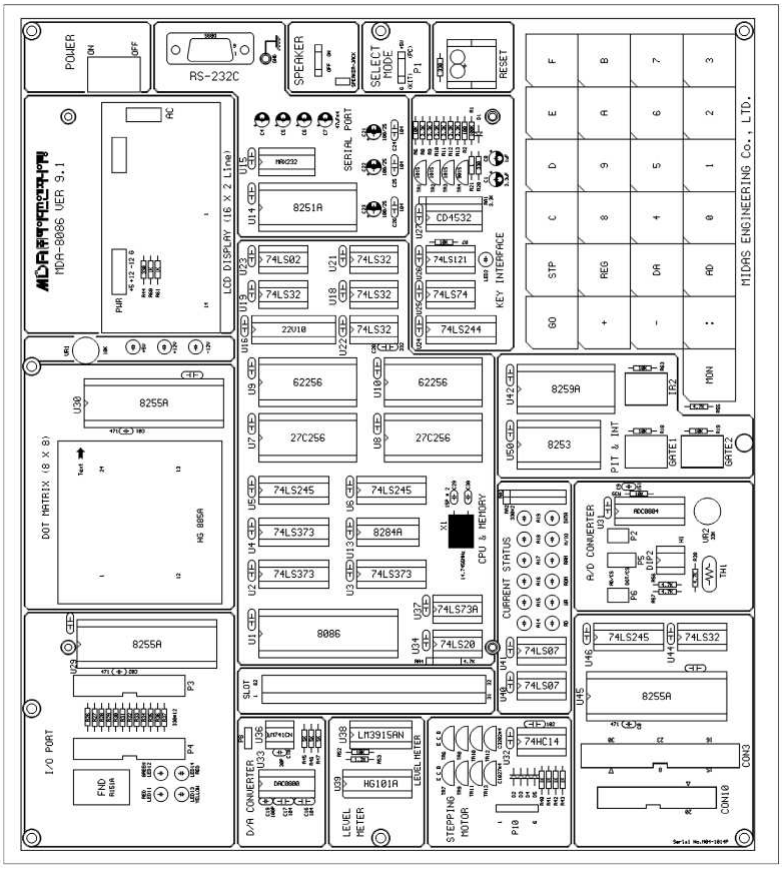

Figure 1: MDA-8086 Kit Components Schematics

1.2 Memory Map

Figure 2 shows the memory address mapping.

ADDRESS	MEMORY	DESCRIPTION
00000H ~ 0FFFFH	RAM	PROGRAM & DATA MEMORY
F0000H ~ FFFFFH	ROM	MONITOR ROM
10000H ~ EFFFFH	USER'S RANGE	

Figure 2: Memory Address Map

1.3 IO Map

Figure 3 shows the I/O address mapping. **(This Table will be useful for you to determine some IO devises addresses)**

ADDRESS	I/O PORT	DESCRIPTION
00H ~ 07H	LCD & KEYBOARD	LCD Display 00H : INSTRUCTION REGISTER 02H : STATUS REGISTER 04H : DATA REGISTER KEYBOARD 01H : KEYBOARD REGISTER (Only read) 01H : KEYBOARD FLAG (Only write)
08H ~ 0FH	8251 / 8253	8251(Using to data communication) 08H : DATA REGISTER 0AH : INSTRUCTION / STATUS REGISTER 8253(TIMER/COUNTER) 09H : TIMER 0 REGISTER 0BH : TIMER 1 REGISTER 0DH : TIMER 2 REGISTER 0FH : CONTROL REGISTER
10H ~ 17H	8259/SPEAKER	8259(Interrupt controller) 10H : COMMAND REGISTER 12H : DATA REGISTER SPEAKER → 11H : SPEAKER
18H ~ 1FH	8255A-CS1/ 8255A-CS2	8255A-CS1(DOT & ADC INTERFACE) 18H : A PORT DATA REGISTER 1AH : B PORT DATA REGISTER 1CH : C PORT CONTROL REGISTER 8255-CS2(LED & STEPPING MOTOR) 19H : A PORT DATA REGISTER 1BH : B PORT DATA REGISTER 1DH : C PORT CONTROL REGISTER 1FH : CONTROL REGISTER
20H ~ 2FH	I/O EXTEND CONNECTOR	
30H ~ FFH	USER'S RANGE	

Figure 3: I/O Address Map

1.4 Kit Operations Modes

On a power-up, One of the messages in figure 4 will be displayed on the Kit LCD.

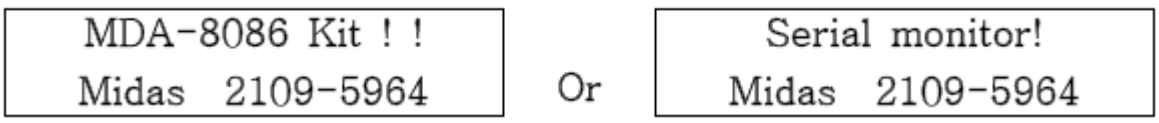

Figure 4: Message on Kit Display When Power Up

Each indicates a different operation mode for the kit. Kit operates either in **Serial Mode** or **Keypad Machine Code Mode**. Choosing between them can be done using jumper P1 located on kit (Figure 5).

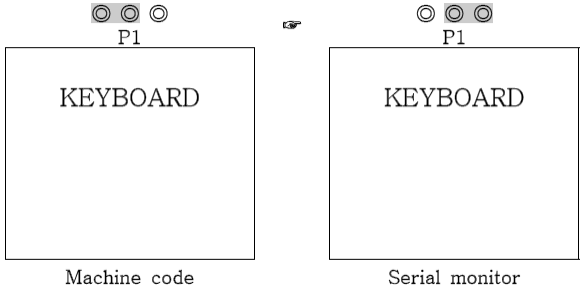

Figure 5: Switching Between Kit Operation Modes

When ever it is a **Kit Machine Code Mode,** kit keys-pad has the full control over the kit. On the other hand, when the kits operates in the **Serial Mode**, kit keys are functionless. WinComm software will be the window to the kit, where tools are provided to control the kit. The following sub-sections (1.4.1 & 1.4.2) will give you more insight of what can be done in each mode.

1.4.1 Machine Code Mode (Function Of Kit Keys)

MDA-8086 has 64K-byte monitor program. It is designed for easy function. After power is on, the monitor begins to work. In addition to all the key function the monitor has a memory checking routine. Below is the list of the keypads main functions.

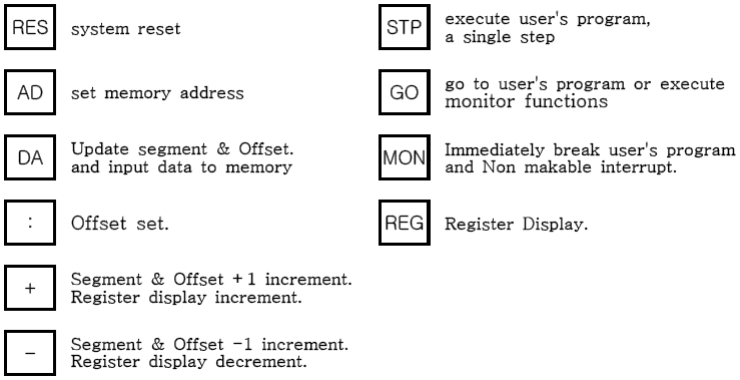

Figure 6: Kit Keys Functions

Some practices will be introduced in the **Kit Machine Code Mode Practices**2.

1.4.2 Serial Monitor Mode

Serial Monitor is the basic monitor program to do data communication between MDA-8086 and computer. As we mentioned before, to use serial monitor, switch jumper P1 to the Serial Monitor Mode (Figure 5). MDA-8086 is connected with PC's serial port using RS-232C cable. After pressing the **RESET** button, make sure that the LCD is giving the right message (Figure 4).

WinComm is an MS Windows application prepared by MEDAS, the company manufactured MDA Kit. It is used to debug the Kit Memory Content and CPU Execution.To start and setup WinComm:

1. Go to start All programs MIDAS ENG WinComm, or you can find it directly on the desktop.

```
** 8086 Monitor 1.0 **
** Midas 335-0964/5 **

8086 > ⊟ ← Carriage Return

8086 >?⊟
 HELP  COMMAND
 E segment : offset.....: Enter Data To Memory
 D segment : offset length.....: Dump Memory Contents
 R [register name].....: Register Display & Change
 M address1, length, address2.....: Move Memory From 1 to 2
 F address, length, data.....: Fill Memory With Any Data
 L Return key.....: Program Down Load
 G segment : offset.....: Execute Program
 T.....: Program 1 step execute
```

Figure 8: Help Command in WinComm

2. In WinComm, go to Option Set Serial Port, Set them as shown in Figure 7. (It is highly recommended to review some of the serial communications basics)

3. Now you can communicate with MDA-8086 kit from your PC

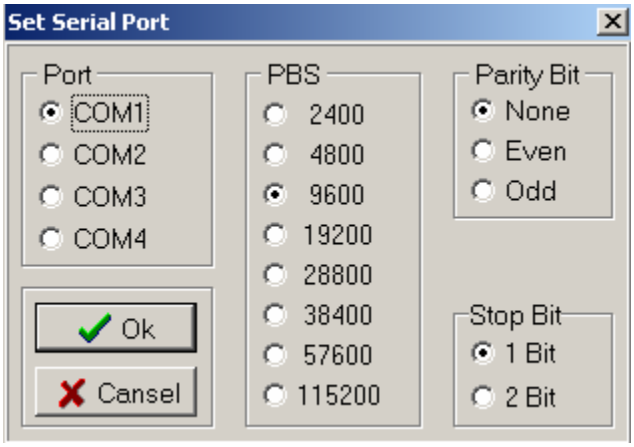

Figure 7: Set Serial Port Options For WinComm

1.5 Serial Monitor Mode Operations

User can only use command which stored at serial monitor. Serial monitor can execute the command when user types command and then CR (carriage return) key. If there is no any command at serial monitor, error message will be displayed with bell sound and serial monitor prompt will be displayed again. WinComm comes with different command, they are very similar to MS-DOS debug commands. You can show them by typing: ? [Press ENTER] (Figure 8) In the following sub-sections a brief is introduced about each command.

1.5.1 Memory Modify Command.

```
          Segment  Offset
             ↓      ↓
8086 >E 0000:1000↵
0000:1000 FF ? 11↵
0000:1001 FF ? 22↵
0000:1002 FF ? 33↵
0000:1003 FF ? 44↵
0000:1004 FF ? 55↵
0000:1005 FF ? / ↵ ← (Offset decrement)
0000:1004 55 ? / ↵
0000:1003 44 ? . ↵  ← (Escaping command)
```

1.5.2 Memory Display Command

```
          Segment  Offset
             ↓      ↓
8086 >D 0000:1000↵
0000:1000 11 22 33 44 55 FF FF FF - FF FF FF FF FF FF FF FF    ."3DU.....
0000:1010 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1020 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1030 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1040 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1050 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1060 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
0000:1070 FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF    ........
8086 >                                          Display the ASCII code to data
```

1.5.3 Memory Fill Command

```
       Segment  Length Data
          ↓      ↓     ↓
8086 >F 1000 FF 1234↵
```

Verify the result using D command!

1.5.4 Block Move Command

```
       Segment  Length  Data
          ↓      ↓      ↓
8086 >M 1000 100 2000↵
```

Verify the result using D command!

1.5.5 Display and Modify CPU Registers

```
8086 >R↵
       AX=0000  BX=0000  CX=0000  DX=0000
       SP=0540  BP=0000  SI=0000  DI=0000
       DS=0000  ES=0000  SS=0000  CS=0000
       IP=1000  FL=0000  = . . . . . .
```

1.5.6 Program Download &Program Execute Command

Sometimes, developers needs to fill a memory with a specific program instructions. What is really needed is to get the byte code of this program. Here, WinComm enables developers to load a HEX file of any program to the Kit memory using L Command.

```
8086 >L↵
   Down load start !!   ←
```

After pressing **L**, press **F3** and chose the HEX file to be loaded.

1.5.7 Execute a Program: You can use G or T command

☞ Execute program command

```
        Segment  Offset
           ↓      ↓
8086 >G 0000:1000↵
Execute Address = 0000:1000
        .
```

Part II
Practices

This part constitutes the crux of the experiment. The practices in this part are intended to be carried out sequentially, so follow them in the proper order. The following notes are worth taking into consideration before performing the practices.

- These Practices should be carried out after reading the practical introduction part. You may refer to that part several times while you are doing the practices.
- Check the kit operation mode every time you are doing an exercises.

2 Kit Machine Code Mode Practices

The aim of this practice is to verify the functionality of the kit keys-pad. Make sure that kit is operating using Kit Machine Code Mode.

2.1 Practice 1: Showing the memory content

Figure 9 shows how you can display the contents of memory

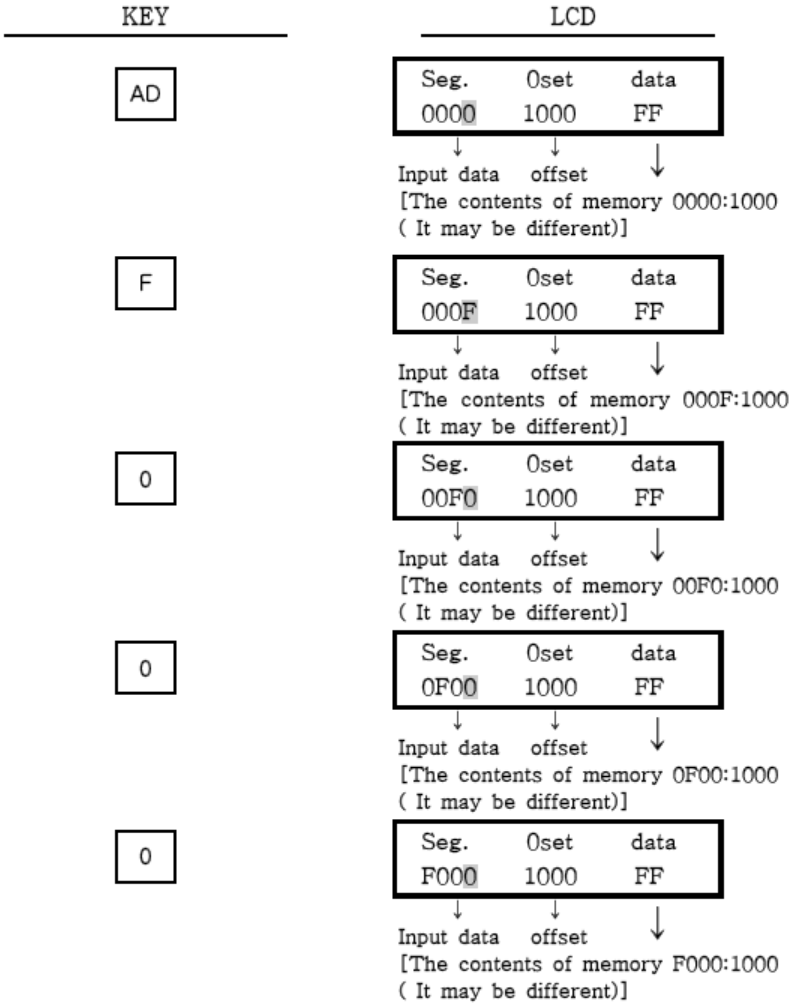

Figure 9: Showing Memory Content Using Kit Machine Code Mode

2.2 Practice 2: Modifying the memory content

Figure shows how to Change the content of the kit memory. The example will store the word AB at location 1000 of memory.

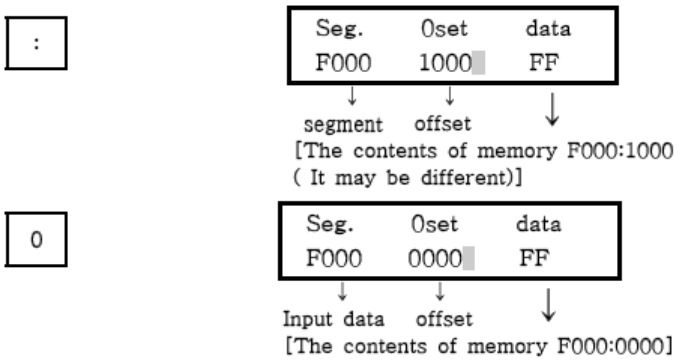

Figure 10: Changing the Memory Content Using Kit Machine Code Mode

2.3 TO-DO 1: Modifying the memory content

Store the data in Table 1 to addresses 01000H - 01004H:

Address	Content
1000	AB
1001	CD
1002	EF
1003	34
1004	E3

Table 1: Memory Content TO-DO 1

2.4 Practice 3: Display CPU Registers Contents

Use REG key to display register contents.

2.5 TO-DO 2: Example Program

In this TO-DO we need to fill manually though the keys-pad a specific program to the kit memory. You can store a program in memory using its machine code. Machine code can be input using keypad.

For example the machine code of MOV AX ,0 is B8 00 00.

Your exercise is to write an assembly code to add two numbers, and store it on MDA-8086 memory using machine code language. You can use MS-DOS debug program to know the machine code of your program. You can use STP key to execute your code in single steps.

3 Serial Monitor Mode Practices

3.1 Practice 4: Loading and Running a Program using MDA-Kit (ASM File)

In this Practice we need to write a program using Assembly language. Our goal is to get the hex file of this code and upload it to the kit.

As a first step: Write the following assembly code, name it led.asm:

```
CODE SEGMENT
    ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE ;
PPIC_C EQU 1FH
PPIC EQU 1DH
PPIB EQU 1BH
PPIA EQU 19H ;
    ;
    ORG 1000H
    MOV AL,10000000B
    OUT PPIC_C,AL
    ;
    MOV AL,11111111B
    OUT PPIA,AL
    ;
    MOV AL,00000000B
    OUT PPIC,AL
    ;
L1:
    MOV AL,11110001B
L2:
    OUT PPIB,AL
    CALL TIMER
    SHL AL,1
    TEST AL,00010000B
    JNZ L1
    OR AL,11110000B
    JMP L2
    INT 3
TIMER:
    MOV CX,1
TIMER2:
    PUSH CX
    MOV CX,0
TIMER1:
    NOP
    NOP
    NOP
    NOP
    LOOP TIMER1
    POP CX
    LOOP TIMER2
    RET
CODE ENDS
    END
```

When this code run on the kit processor, it will configure one of the PPIs connected on the base address **19H** to do a specific job. Your task is to:

1. Understand the code very well, every single line of it. (Refer to Figure 12)
2. Compile and build your program to get an **executable (EXE)** file. (Refer to A)
3. Change the file to **HEX** file that you can send to MDA-8086 Kit. (Refer to C)
4. Load the program to the Kit memory. (Refer to 1.5.6)
5. Execute the program. (Refer to 1.5.7)

```c
#include    "mde8086.h"

void    wait( long del )
{

    while( del-- );
}

void    main( void )
{
    unsigned char led;

    outportb( PPI1_CR, 0x80 );
    outportb( PPI1_B, 0xff );
    outportb( PPI1_A, 0xff );
    outportb( PPI1_C, 0X20 );

    led = 0xf1;
    do  {
       outportb( PPI1_B, led );
       led = led << 1;
       if( led & 0x10 )  led = 0xf1;
       wait( 10000 );
    } while( 1);

}
```

Figure 11: Led.c File

3.2 Practice 5: Loading and Running a Program using MDA-Kit (C File)

In this Practice we need to write a program using C language. Our goal is to get the hex (or ABS) file of this code and upload it to the kit.

Create a C file called led.c, type the following code in tha file.

This code does the same as what assembly code in practice 4 does. Notice the header file in the C code "mde8086.h" which exist under "C:\mda\8086\8086C" (refere to this file whenever you face a constant or definition).

You need to Compile and Build this file (Refer to B), then Convert it to Hex file (Refer to C), Load (Refer to 1.5.6) and Execute (Refer to 1.5.7) the program.

Modify the file you have created. Your task is to implement a binary counter counts from 0 to 7 and back again to 0. Use LED11,LED12, and LED13 (Figure 12).

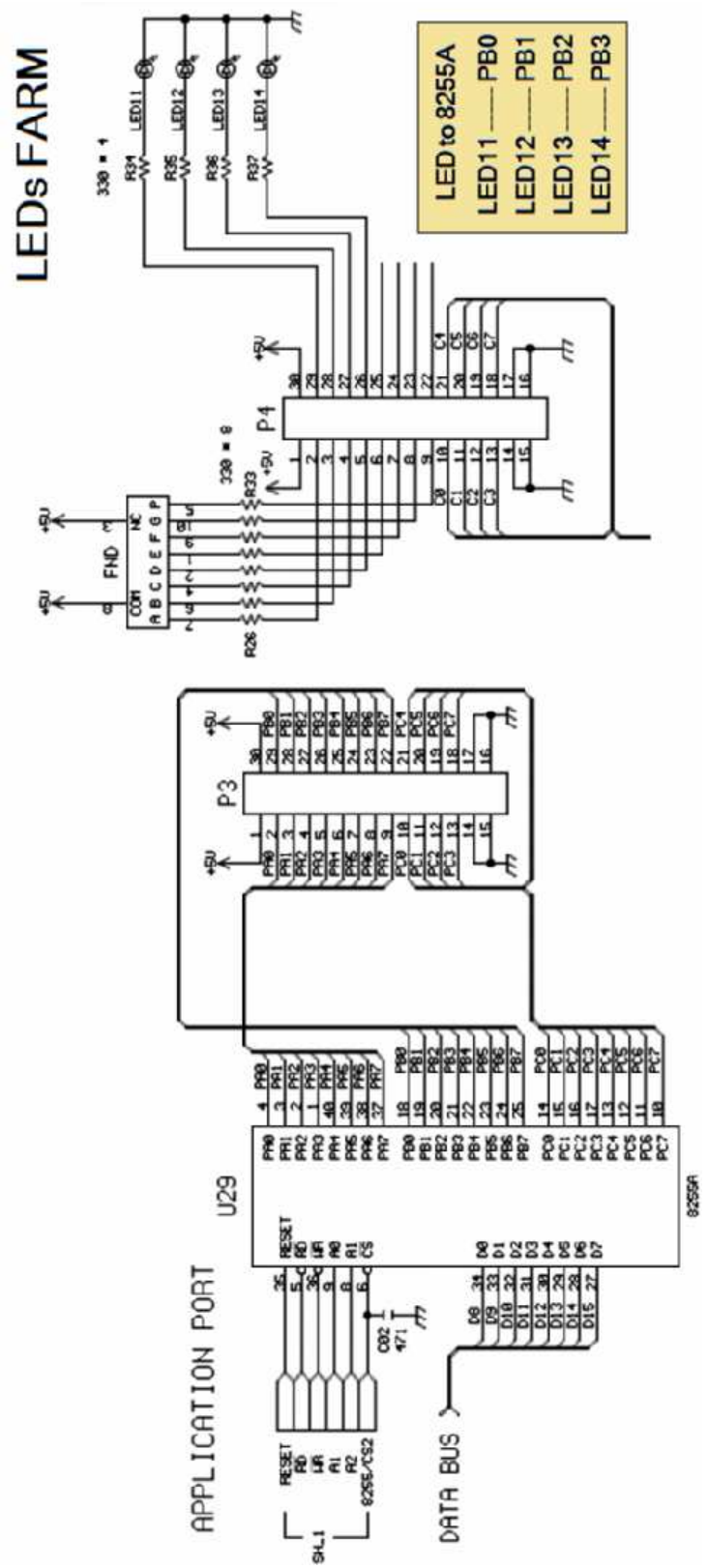

Figure 12: PPI Schematics

Part III
Appendix

A Compiling and Building ASM File

Here we give you one of the ways to compile and build Assembly files to produce executable file. Under Ms Windows OS, we use TASM program and TLINK program to compile and link the Assembly code respectively. Steps are as following:

1. TASM and TLINK programs are by default located on your Lab PC under "D:\Tasm" Folder.
2. Open DOS Command Terminal (CMD). Click Start -> Run -> Type "cmd" -> OK
3. Change Directory to the one where your ASM File located (CD followed by Directory path is used to change the current working directory).
4. To Compile the ASM file Type: **"D:\Tasm\Tasm file.asm"** (Supposed your file name is file.asm). After this step some lines will appear up on the command terminal showing the results of compilation. To have a successful compilation no errors should be found. An OBJ file will be generated after a successful compilation.
5. To Like the OBJ file Type: **"D:\Tasm\Tlink file.obj"**. After this step a new executable (EXE) file will be generated.

B Compiling and Building C File Using WinComm

Compiling C files can be done using so many well known compilers such as the ones produced by Borland. WinComm software contains Turbo Borland C Compiler. A prepared batch script is already exist in WinComm which enables developers to compile and build their C code. Steps to do so are:

1. Save your C file under "C:\mda\8086\8086C"
2. Using WinComm Window Go to File->Open work file. An open dialog shows up. Choose your C file.
3. Go to tool C8086, double click on C 8086 to compile the file.
4. Now find the generated ABS file in the same directory. Use this file to upload it to the kit.
5. Note: An EXE file in the same directory could be found. You can use this file to get the HEX file (refering to C). we prefer to use the ABS file.
6. Note: When facing any challenges in compiling the C file, you can go for more detailed steps using the DOS command line. The Turbo C compiler is found under "C:\mda\8086\8086C", follow this step to get detailed information about the file compilation:
   (a) C:\mda\8086\8086C> tcc -c file
   (b) Now, check the results of your file compilation, if no errors run the patch.

C Converting EXE File to HEX File.

HEX files are needed to be obtained as it is the only format accepted by the MDA Kit. EXE file can be converted to get HEX file following these steps:

1. All programs needed for conversion are by default located on your Lab PC under "D:\Tasm" Folder.
2. Open DOS Command Terminal (CMD). Click Start -> Run -> Type "cmd" -> Ok
3. Change Directory to the one where your asm File located (CD followed by Directory path is used to change the current working directory).

4. Type: **"D:\Tasm\EXE2BIN file.EXE"** (Supposed your executable file name is file.exe). After this step a **BIN** file will be generated.

5. Type: **"D:\Tasm\BIN2HEX file.BIN hexfile.HEX"**. (Supposed your Binary file name is file.bin). After this step a HEX file named hexfile.hex will be generated.