Documentation for z80 Mainframe

Colton Beck

April 10, 2019

Legal Information

Disclaimer of Warranty

Limitation of Liability

Fonts

This was typeset in Computer Modern using pdfLaTeX and bibTeX. The timing diagrams were made with the tikz-timing package. The code listings were made with the listings package

Dedication

Dedicated to caffeine for giving me the energy to write this and sleep deprivation for making me think this was a good idea.

Contents

List of Figures

List of Tables

Chapter 1

Main Board

1.1 Power Supply

The system is designed to use one power supply to provide the low voltage for the peripherals and main board. For the system as shown here, the supply should be rated at least seven amperes at five volts and fifteen amperes at twelve volts for a combined two hundred fifteen watts of total power. If additional peripherals are expected to be added, then the power supply should be of a higher wattage so that the whole system can be powered from a unified supply to avoid the risk of ground loops that could induce excessive noise.

1.2 Physical Connections

1.2.1 Power

For the rear panel power connections, any connectors may be used as long as they have sufficient current capacity for the load expected. For my design, I elected to use 3-pin XLR connectors as listed in Appendix B.7. If XLR connectors are used for power, they should be marked as such to avoid any accidental damage. The system is designed such that the power is daisy-chained from one preipheral to the next. This does mean that the total power draw needs to be taken into consideration when selecting a conector type to use.

1.2.2 Data

For the data connections, it is recommended to use 8p8c modular connectors along with cat 5 cable to match the impedance for the differential connection. If a different cable is used, the termination resistors on the peripherals need to be changed to be of a suitable value to prevent ringing. The system is designed such that the data is daisy-chained from one preipheral to the next. This has

Pin Number	Function	Wire Color
1	Data-	White/Orange
2	Data+	Orange
3	NC	White/Green
4	NC	Blue
5	NC	White/Blue
6	NC	Green
7	Clock-	White/Brown
8	Clock+	Brown

Table 1.1: Interface Pin Functions

the advantage that termination resistors only need to be used on the first and last device in the chain as specified on pg. 12 of the datasheet[2].

1.3 Interface

The inter-peripheral communication is done via differental I^2C running at 400kbit/s. This link is designed to be wired as illustrated in figure 1.1 and in table 1.1. Using a parallel connection would allow for a higher throughput and would simplify some of the design however this comes at a higher cost due to the connectors and cabling. A circuit, shown in figure **??**, is used to identify when a peripheral is being addressed and triggers a wait for the z80 and notifies a PIC that there is data ready to transmit. This portion of the circuit is designed to be modular since one copy is needed for each peripheral that is to be addressed and is to be expandable through stackable modules, shown in figure **??**.

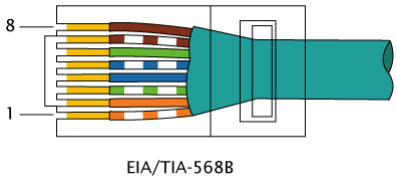


Figure 1.1: Example wiring using EIA-568B[1]

The z80 mainframe was designed to be modular and expandable. It accomplishes this by having a simple interface that either gets data from memory using direct memory access(DMA) or writes data using DMA. The system is limited to 128 connected peripherals because of the limitation of I^2C's 7-bit addresses.

The interface PIC(U?? in appendix C.3) is designed to look for specific addresses after which the it pulls the BUSREQ line low and reads from 0x0800 to 0x084F or writes to 0x850 to 0x089F. The peripherals listed here are designed to be configurable as to what address they respond to and the OS is configurable for where it is trying to address these devices at. Both are configured in hardware rather than in software to simplify configuration so the OS can determine settings without the ROM needing to be modified to be installation dependent.

1.4 Configuration

The system is configured through DIP switches for the addresses that the peripherals described here are located at.

1.5 ROM

The ROM, listed in appendix A.1, is intended to funtion as a bootstrap script to prepare the system to load a program from another medium. It also serves to provide callable funtions for the default peripherals to ease implimentation.

Chapter 2

Front Panel

The front panel connects directly to the main board through a 36 pin header that caries the 16-bit address bus, the 8-bit data bus, power, and the 10 control lines; BUSREQ, BUSACK, WAIT, MREQ, IORQ, RD, WR, M1, HALT, and RESET. It is designed with status LEDs for the control lines and buses as well as hexadecimal readouts for the buses. The front panel also includes the circuitry for an instruction single stepping circuit along with halt and reset controls. There also are four 8-bit inputs and four 8-bit outputs.

2.1 Outputs

2.2 Inputs

Chapter 3

VGA Terminal

Chapter 4

Dot Matrix Printer

The main output for the z80 mainframe is the printer. This particular setup is designed to use an Epson LX-810 printer interfacing over a parallel port as shown in figure (null pointer).

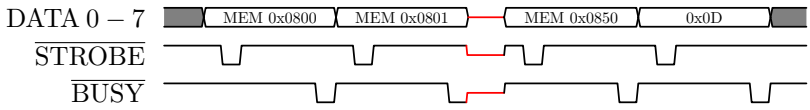


Figure 4.1: Driver Board to Printer Timing

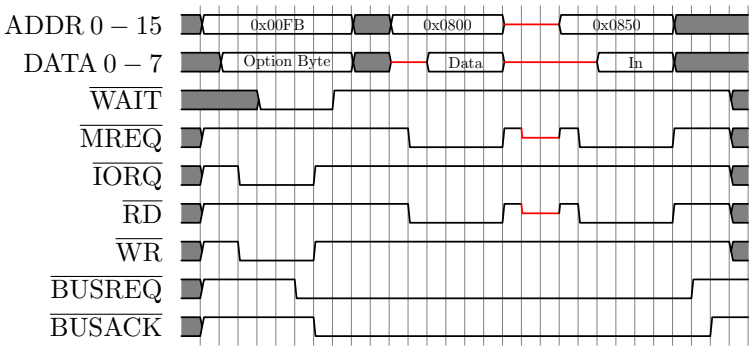


Figure 4.2: Main Bus to Printer Driver Timing

Chapter 5

Card Punch & Reader

Chapter 6

Paper Tape Punch & Reader

Appendix A

Code Listings

A.1 ROM Listing

A.2 Main Board Interface Listing

A.3 VGA Terminal Listing

A.4 Line Printer Driver Listing

```
1  char header[]={0x00,0x1B,0x40,0x1B,0x52,0x00,0x1B,
2                 0x74,0x01,0x1B,0x36,0x12,0x1B,0x50};
3  void setup() {
4    // put your setup code here, to run once:
5    Serial.begin(2400);
6    pinMode(2,OUTPUT);
7    pinMode(3,OUTPUT);
8    pinMode(4,OUTPUT);
9    pinMode(5,OUTPUT);
10   pinMode(6,OUTPUT);
11   pinMode(7,OUTPUT);
12   pinMode(8,OUTPUT);
13   pinMode(9,OUTPUT);
14   pinMode(10,OUTPUT);
15   pinMode(11,INPUT);
16   digitalWrite(10,HIGH);
17   for(int i=0; i<sizeof(header);i++)
18   {
19     for(int j=0;j<8;j++)
20     {
21       if(((header[i]>>j)&1)==1)
22       {
23         digitalWrite(j+2,HIGH);
24       }
25       else
26       {
27         digitalWrite(j+2,LOW);
28       }
29     }
30     delayMicroseconds(10);
31     digitalWrite(10,LOW);
32     delayMicroseconds(10);
33     digitalWrite(10,HIGH);
34     while(digitalRead(11)==HIGH){};
35   }
36   Serial.println("Ready...");
37 }
38 int feed = 0;
39 void serialEvent()
40 {
41   char data=Serial.read();
42   for(int j=0;j<8;j++)
43     {
```

```
44        if(((data>>j)&1)==1)
45        {
46          digitalWrite(j+2,HIGH);
47        }
48        else
49        {
50          digitalWrite(j+2,LOW);
51        }
52      }
53      delayMicroseconds(10);
54      digitalWrite(10,LOW);
55      delayMicroseconds(10);
56      digitalWrite(10,HIGH);
57      Serial.print(data);
58      if(data == 0x0d){feed++;Serial.println();};
59      if(data != 0x0d){feed=0;};
60      if(feed >= 3){
61        feed=0;
62        data = 0x0c;
63        for(int j=0;j<8;j++)
64        {
65          if(((data>>j)&1)==1)
66          {
67            digitalWrite(j+2,HIGH);
68          }
69          else
70          {
71            digitalWrite(j+2,LOW);
72          }
73        }
74        delayMicroseconds(10);
75        digitalWrite(10,LOW);
76        delayMicroseconds(10);
77        digitalWrite(10,HIGH);
78      }
79      while(digitalRead(11)==HIGH){};
80  }
81
82  void loop() {
83    // put your main code here, to run repeatedly:
84
85  }
```

A.5 Card Punch Driver Listing

A.6 Card Reader Driver Listing

A.7 Paper Tape Punch Driver Listing

A.8 Paper Tape Reader Driver Listing

Appendix B

Part List

B.1 Main Board

B.2 Front Panel

B.3 VGA Terminal

B.4 Line Printer Driver Board

Ref	Value	Desc	PN	Quantity	Price	Notes
R1	4k7	1/4 watt	CF14JT4K70CT-ND	1	$0.10	
R2	10k	1/4 watt	CF14JT10K0CT-ND	1	$0.10	
R3,R4	27R	1/4 watt	CF14JT27R0CT-ND	2	$0.10	
R5,R6	270R	1/4 watt	CF14JT270RCT-ND	2	$0.10	
R7,R8	2k	1/4 watt	CF14JT2K00CT-ND	2	$0.10	
R9,R11,R12,R14	600R	1/4 watt		4		
R10,R13	120R	1/4 watt	CF14JT120RCT-ND	2	$0.10	
C1,C2	47p	Ceramic	BC1009CT-ND	2	$0.27	
C3	100n	Ceramic	BC5229CT-ND	1	$0.23	
C4	470u	Polarized electrolytic	P5141-ND	1	$0.10	
C5,C6,C7,C8	10n	Ceramic	BC5136-ND	4	$0.21	
D1,D2		5mm red led		2		
U1	FT230XS	FTDI USB to Basic UART	768-1135-1-ND	1	$2.04	
U2	PIC16F1503-IP		PIC16F1503-I/P-ND	1	$0.93	
U3	74HC595	8-bit Serial-in Parallel-out Shift Register	296-1600-5-ND	1	$0.53	
U4	PCA9615	Differential I2C Bus buffer	568-11484-1-ND	1	$2.95	
J1		USB type B Female		1		
J2		1x6 .1" male header		1		
J3		1x3 Power connector		1		
J4		8p8c female		1		
J5		Dsub-25 female w/ mounting holes		1		

B.5 Card Punch & Reader Driver Board

B.6 Paper Tape Punch & Reader Driver Board

B.7 Miscellaneous Parts

Desc	PN	Quantity	Price	Notes
Male XLR Receptacle	SC2465-ND	5	$5.26	
Female XLR Receptacle	SC1992-ND	5	$5.80	
Male XLR Plug	889-2138-ND	5	$4.43	
Female XLR Plug	SC2465-ND	5	$4.75	

Appendix C

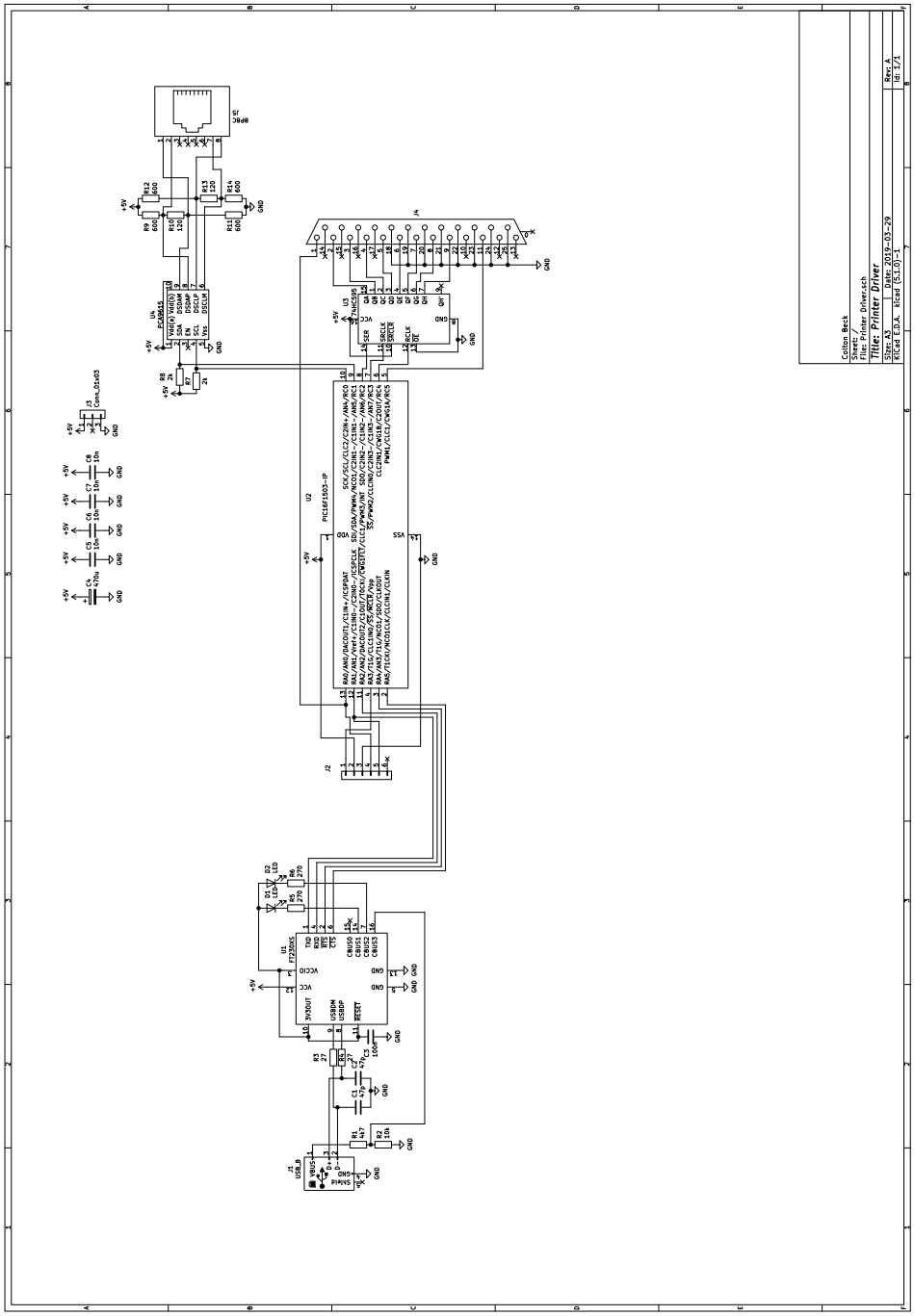
Circuit Diagrams

C.1 Main Board

C.2 Front Panel

C.3 VGA Terminal

C.4 Line Printer Driver Board



C.5 Card Punch & Reader Driver Board

C.6 Paper Tape Punch & Reader Driver Board

Appendix D

PCB Masks

D.1 Main Board

D.2 Front Panel

D.3 VGA Terminal

D.4 Line Printer Driver Board

D.5 Card Punch & Reader Driver Board

D.6 Paper Tape Punch & Reader Driver Board

Appendix E

Part Drawings

Bibliography

[1] Wikimedia Commons. File:rj-45 tia-568b left.png — wikimedia commons, the free media repository, 2015. [Online; accessed 3-March-2019].

[2] NXP Semiconductor. *2-channel multipoint Fast-mode Plus differential I2C buffer with hot swap logic*, 5 2016. Rev. 1.1.