Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Lab Report Cover Page

Course Number : 3613

Course Section : 001 / 002

Experiment # : Lab # 10

Student Name (print) : Von Kaukeano

TUid# : 915596703

Date : 11/7/19

Grade : _____ **/100**

TA Name : Sung Choi

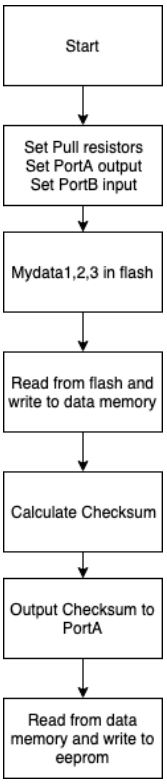
Introduction

The objective of this lab is learning how to write to EEPROM, how a checksum works, and how to read external data. In order to write to EEPROM you must wait until EEPE the enable bit becomes zero. Write to the address register EEAR, set the master enable write bit to one EEMPE, and set the slave enable bit to one EEPE after four clock cycles. We have already wrote data into data memory and flash memory and this lab is a combination off three different read and writes to different locations. Understanding how to write to EEPROM is important because if the steps are not properly done then you could have an error wrote into EEPROM. The Checksum is a way to check that the transfer of data is correct when written into memory. By adding the values, taking the twos, then adding it to the original sum should be an outcome of zero. It can be described as adding a positive number to its negative. For example, -3 _ 3 = 0. If the outcome is not zero then the data has been corrupted.

Procedure

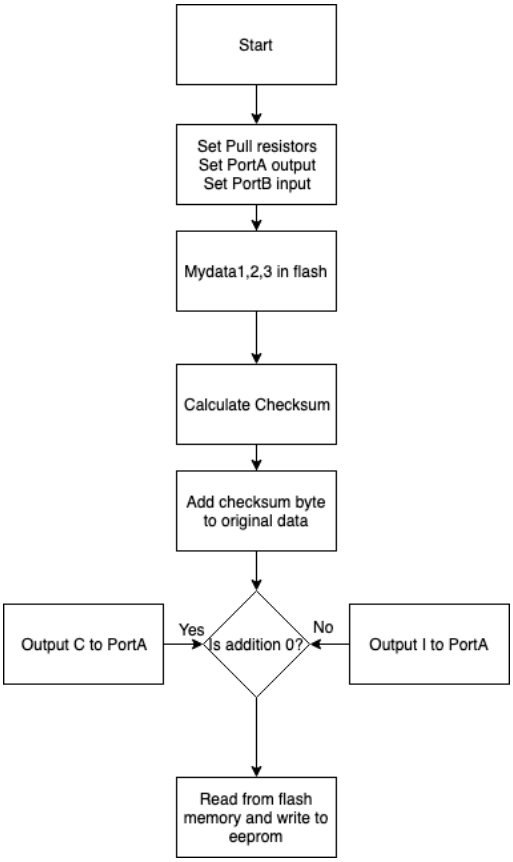
Activity 1

Beginning activity 1, I wrote the data sets into flash memory at their specific location beginning at 0x200 by .org 0x100. After loading the data, I began reading from flash and writing the values into data memory at location beginning at 0x200. As I read the values into data memory I had a register be the sum of the values for each data set then took the two's complement and output them to Port A which is the checksum byte. Finally I then read the values from data memory and wrote them to EEPROM beginning at location 0x200. After adding the values up, it matched with the output of the LEDS.



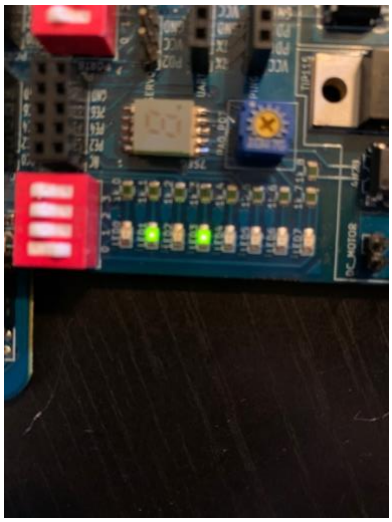
Activity 2

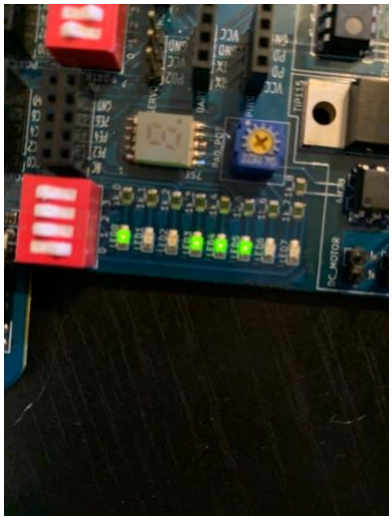
Beginning activity 2, I wrote the data sets into flash memory at their specific location beginning at 0x200 with the correct dataand 0x300 with the incorrect data . After loading the data, I started the program by checking for PinB0 input. Depending on the input it would ready from the flash memory and write to EEPROM at 0x100, 0x150, and 0x200. As it is writing to EEPROM I made a register be the addition of each data set, took the two's compliment, and then added them together to check if it was zero. If it was zero then it will output a 'C' onto the seven seg display, if incorrect and not zero it would output 'I' or 1 on the seven seg display. I received 3 'C's and their output of zero and three 'I's with outputs of nonzero.



Result and analysis

Activity 1





Above are the checksum bytes output to the LED's on Port A.

Memory: prog FLASH | Address: 0x0200,prog

```
prog 0x0200  12 36 83 a5 5d 29 54 45 4d 50 4c 45 45 6e 67 69 6e 65 65  .6ƒ¥])TEMPLEEnginee
prog 0x0213  72 69 6e 67 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ring.ÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Memory: data IRAM | Address: 0x0200,data

```
data 0x0200  12 36 83 a5 5d 29 54 45 4d 50 4c 45 45 6e 67 69 6e 65 65  .6ƒ¥])TEMPLEEnginee
data 0x0213  72 69 6e 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ring...............
data 0x0226  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ...................
```

Memory: eeprom EEPROM | Address: 0x0200,eeprom

```
eeprom 0x0200  12 36 83 a5 5d 29 54 45 4d 50 4c 45 45 6e 67 69 6e 65 65  .6ƒ¥])TEMPLEEnginee
eeprom 0x0213  72 69 6e 67 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ringÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Above is the data in all three memory different memory types.

https://drive.google.com/file/d/1Cvp0GPg8BoiagNw7m9g_xp7dpYbockSA/view?usp=sharing

Activity 2

Memory: prog FLASH | Address: 0x0200,prog

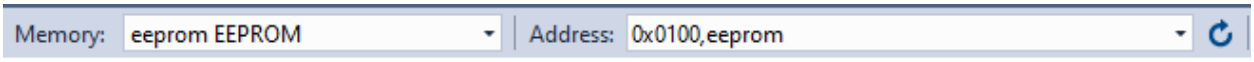
```
prog 0x0200  12 36 83 a5 5d 29 54 45 4d 50 4c 45 45 6e 67 69 6e 65 65  .6ƒ¥])TEMPLEEnginee
prog 0x0213  72 69 6e 67 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ring.ÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

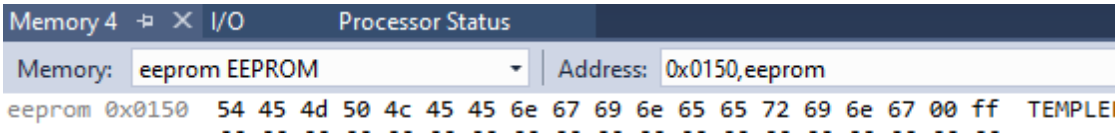
Memory 4 | I/O | Processor Status

Memory: prog FLASH | Address: 0x0300,prog

```
prog 0x0300  12 33 83 a5 5d 29 54 45 4d 42 4c 45 65 6e 67 69 6e 65 65  .3ƒ¥])TEMBLEenginee
prog 0x0313  72 69 6e 67 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ring.ÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Above are the correct and incorrect data written to flash.





Above are the data written in EEPROM. There was an error after adding the delay. Unsure if I am writing over registers used to write to EEPROM. It worked completely before adding it.

https://drive.google.com/file/d/1Zs8EgXUOFSxLLkARFb0DbGjSY8qR4VDv/view?usp=sharing

Discussion and Conclusion

This lab was interesting working with EEPROM. Writing to EEPROM is more difficult and confusing then working with flash or data memory. It can easily be effected constantly writing to it and creating bugs. It must be related to the wait time for the write enable to reset back to zero. I have also heard and calculated checksum in courses such as data and computer communications and it is valuable to see a byte by byte data comparison to see how the checksum is applicable in hardware rather than the theory. Everything matched my expectations besides the EEPROM in activity two which was correct before adding the delay for the Pin B input.

Code

1.

```
LDI R16,$00
LDI R17,$FF
OUT PORTB,R17
OUT DDRA,R17
OUT  DDRB,R16

RELOAD:
LDI ZL,00
LDI ZH,02
LDI R16,6
LDI R17,6
LDI R18,11
LDI R19,23
LDI XL,00
LDI XH,02
LDI R20,0
```

```
SRAM:
LPM R23,Z+
ST X+,R23
DEC R19
BRNE SRAM

LDI XL,00
LDI XH,02

LOOP1:
LD R23,X+
ADD R20,R23
DEC R16
BRNE LOOP1

NEG R20
OUT PORTA,R20
CALL DELAY
LDI R20,0
OUT PORTA,R20


LOOP2:
LD R23,X+
ADD R21,R23
DEC R17
BRNE LOOP2

NEG R21
OUT PORTA,R21
CALL DELAY
LDI R21,0
OUT PORTA,R21

LOOP3:
LD R23,X+
ADD R22,R23
DEC R17
BRNE LOOP3

NEG R22
OUT PORTA,R22
```

```
CALL DELAY
LDI R22,0
OUT PORTA,R22

LDI XL,00
LDI XH,02
LDI R19,23

WAIT:
SBIC EECR,EEPE
RJMP WAIT
LD R23,X
OUT EEARH,XH
OUT EEARL,XL
INC XL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R19
BRNE WAIT

DELAY:
   ldi  r24, 41
   ldi  r25, 150
   ldi  r28, 128
L1: dec  r28
   brne L1
   dec  r25
   brne L1
   dec  r24
   brne L1
         RET

RJMP RELOAD

.org 0x0100
DataSet1: .DB $12,$36,$83,$A5,$5D,$29
DataSet2: .DB 'T','E','M','P','L','E'
DataSet3: .DB "Engineering"
```

2.

```
LDI R16,$00
```

```
LDI R17,$FF
OUT PORTB,R17
OUT DDRA,R17
OUT DDRB,R16

Check:
SBIS PINB,0
RJMP Correct
RJMP Incorrect

Correct:
LDI ZL,0x00
LDI ZH,0x02
LDI R16,6
LDI R17,6
LDI R18,12
LDI R20,0
LDI R21,0
LDI R22,0

LDI YL,0x00
LDI YH,0x01


WAIT1:
SBIC EECR,EEPE
RJMP WAIT1
LPM R23,Z+
ADD R20,R23
OUT EEARH,YH
OUT EEARL,YL
INC YL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R16
BRNE WAIT1

MOV R0,R20
NEG R0
MOV R3,R0
ADD R0,R20
```

```
CALL Output1

LDI YL,0x50
LDI YH,0x01

WAIT2:
SBIC EECR,EEPE
RJMP WAIT2
LPM R23,Z+
ADD R21,R23
OUT EEARH,YH
OUT EEARL,YL
INC YL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R17
BRNE WAIT2

MOV R1,R21
NEG R1
MOV R4,R1
ADD R1,R21
CALL Output1

LDI YL,0x00
LDI YH,0x02

WAIT3:
SBIC EECR,EEPE
RJMP WAIT3
LPM R23,Z+
ADD R22,R23
OUT EEARH,YH
OUT EEARL,YL
INC YL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R18
BRNE WAIT3
```

```
MOV R2,R22
NEG R2
MOV R5,R2
ADD R2,R22
CALL Output1

RJMP Check

Incorrect:
LDI ZL,0x00
LDI ZH,0x03
LDI R16,6
LDI R17,6
LDI R18,12
LDI R20,0
LDI R21,0
LDI R22,0


LDI XL,0x00
LDI XH,0x01

WAIT4:
SBIC EECR,EEPE
RJMP WAIT4
LPM R23,Z+
ADD R20,R23
OUT EEARH,XH
OUT EEARL,XL
INC XL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R16
BRNE WAIT4


ADC R3,R20
CALL Output2

LDI XL,0x50
LDI XH,0x01
```

```
WAIT5:
SBIC EECR,EEPE
RJMP WAIT5
LPM R23,Z+
ADD R21,R23
OUT EEARH,XH
OUT EEARL,XL
INC XL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R17
BRNE WAIT5


ADC R4,R21

CALL Output2

LDI XL,0x00
LDI XH,0x02

WAIT6:
SBIC EECR,EEPE
RJMP WAIT6
LPM R23,Z+
ADD R22,R23
OUT EEARH,XH
OUT EEARL,XL
INC XL
OUT EEDR,R23
SBI EECR,EEMPE
SBI EECR,EEPE
DEC R18
BRNE WAIT6

ADC R5,R22

CALL Output2

RJMP Check
```

```
Output1:
LDI R19,0b00111001
OUT PORTA, R19
CALL DELAY
LDI R19,0
OUT PORTA, R19
RET

Output2:
LDI R19,0b00000110
OUT PORTA,R19
CALL DELAY
LDI R19,0
OUT PORTA, R19
RET

DELAY:
  ldi  r16, 41
   ldi  r17, 150
   ldi  r18, 128
L1: dec  r18
   brne L1
   dec  r17
   brne L1
   dec  r16
   brne L1
          RET

.org 0x0100
DataSet1: .DB $12,$36,$83,$A5,$5D,$29

DataSet2: .DB 'T','E','M','P','L','E'
DataSet3: .DB "Engineering"

.org 0x0180
DataSet4: .DB $12,$33,$83,$A5,$5D,$29
DataSet5: .DB 'T','E','M','B','L','E'
DataSet6: .DB "engineering"
```