CPE301 – SPRING 2022

Design Assignment 1

Student Name: Angelo Nolasco
Student #: 5005497011
Student Email: Nolasco@unlv.nevada.edu
Primary Github address: angelon3121@gmail.com
Directory:

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

N/A

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

Assembly Code for Task 1

```
. include <m328pbdef.inc>


.def num1_H = R18    ; define upper byte of number 1 as r18
.def num1_L = R19    ; define lower byte of number 1 as r19
.def num2_H = R20    ; define upper byte of number 2 as r20
.def num2_L = R21    ; define lower byte of number 2 as r21

.ORG 0          ; ROM starting at 0
; task 1, store 16-bit number 0x1234 into the SRAM location 0x402
        LDI R16, 0x34               ; R16 = 0x34
        LDI XL, LOW(0x402)          ; load the low byte of X with value 0x02
        LDI XH, HIGH(0x402)         ; load the high byte of X with value 0x4
        ST X+, R16                  ; copy R16 to memory location X
        LDI R16, 0x12               ; R16 = 0x12
        ST X+, R16                  ; copy R16 to memory location X

; task 2, store 16-bit number 0x5678 into the SRAM location 0x410
LDI R17, 0x56               ; R17 = 0x56
LDI XL, LOW(0x410)          ; load the low byte of X with value 0x10
LDI XH, HIGH(0x410)         ; load the high byte of X with value 0x4
ST X+, R17                  ; copy R17 to memory location X
LDI R17, 0x78               ; R17 = 0x78
ST X+, R17                  ; copy R17 to memory location X

; task 3, need to add 0x1234 and 0x5678, then store into the EEPROM
```

```
LDI num1_L, 0x34          ; num1_L = 0x34
LDI num2_L, 0x56          ; num1_L = 0x56
LDI num1_H, 0x12          ; num1_H = 0x12
LDI num2_H, 0x78          ; num1_H = 0x78
ADD num1_L, num2_L        ; adding the lower bytes of number
ADC num1_H, num2_H        ; adding the upper bytes of number

LDI XH, HIGH(0x20)        ; load the high byte of X with value 0x0
LDI XL, LOW(0x20)         ; load the low byte of X with value 0x2
mov R22, num1_L
CALL STORE_IN_EEPROM          ; call the label
LDI XH, HIGH(0x21)            ; load the high byte of X with value 0x1
LDI XL, LOW(0x21)         ; load the low byte of X with value 0x2
mov R22, num1_H
CALL STORE_IN_EEPROM          ; call the label

; task 4, store 16-bit numbers into 0x0910 then retrieve to 0x500 SRAM and add the 10    ;
number, store it to 0x406
LDI R25,0x00                              ; R25 =0x00
LDI num10,0xA                             ; num10 = 0xA
LDI XL, LOW(0x500)                        ; load the low byte of X with value 0x00
LDI XH, HIGH(0x500)                       ; load the high byte of X with value 0x5
LDI ZH, HIGH(XSQR_TABLE<<1)                       ; look-up table high-byte adder
LDI ZL, LOW(XSQR_TABLE<<1)            ; look-up table low-byte adder
L1:



; part of task 3
STORE_IN_EEPROM:



; part of task 4
.ORG 0x0910
XSQR_TABLE:
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Assembly Code for Task 1

```
. include <m328pbdef.inc>
```

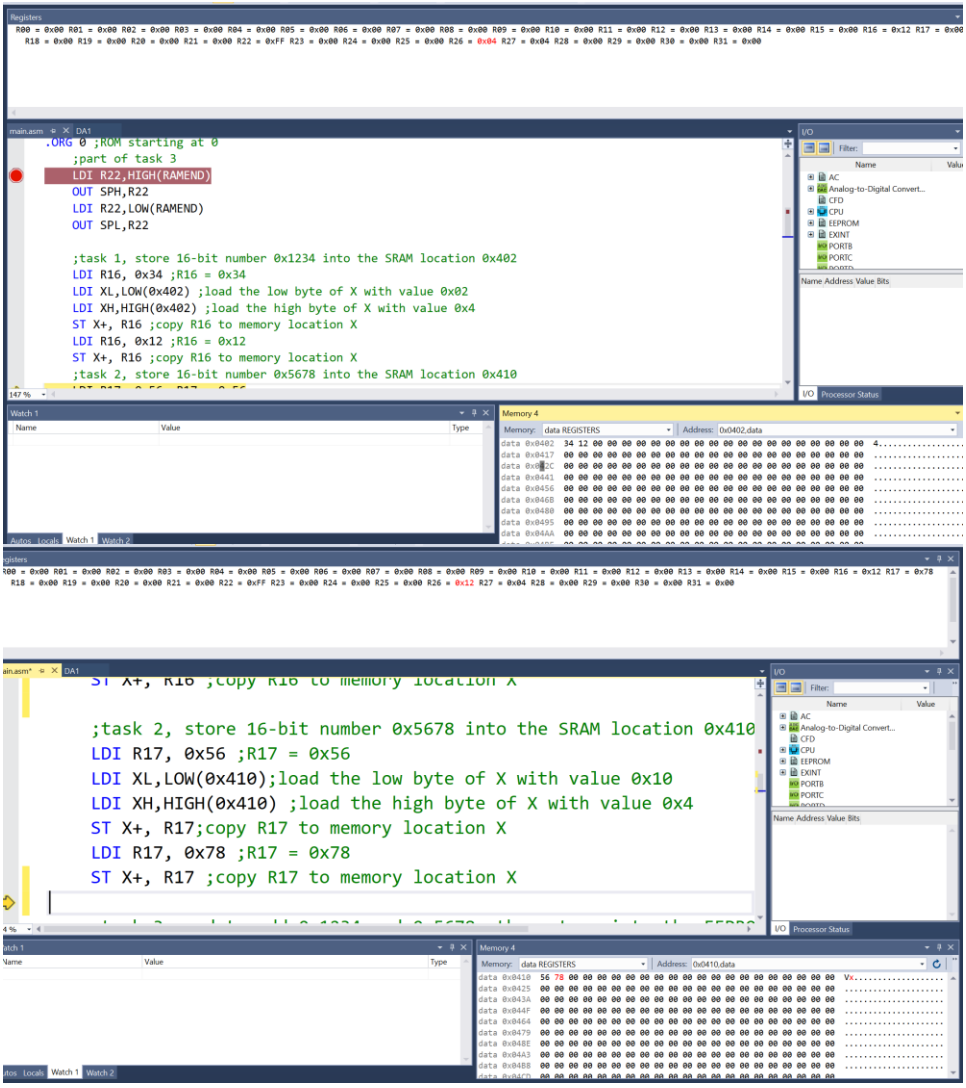
```
. set EEMWE = EEMPE
. set EEWE = EEPE
. EQU SUM = 0x406

.def num1_H = R18    ; define upper byte of number 1 as r18
.def num1_L = R19    ; define lower byte of number 1 as r19
.def num2_H = R20    ; define upper byte of number 2 as r20
.def num2_L = R21    ; define lower byte of number 2 as r21
.def num10 = R23     ; define number of times the number will be add as r23

.ORG 0          ; ROM starting at 0

; part of task 3
LDI R22, HIGH(RAMEND)
OUT SPH, R22
LDI R22, LOW(RAMEND)
OUT SPL, R22

; task 1, store 16-bit number 0x1234 into the SRAM location 0x402
        LDI R16, 0x34               ; R16 = 0x34
        LDI XL, LOW(0x402)          ; load the low byte of X with value 0x02
        LDI XH, HIGH(0x402)         ; load the high byte of X with value 0x4
        ST X+, R16                  ; copy R16 to memory location X
        LDI R16, 0x12               ; R16 = 0x12
        ST X+, R16                  ; copy R16 to memory location X

; task 2, store 16-bit number 0x5678 into the SRAM location 0x410
LDI R17, 0x56               ; R17 = 0x56
LDI XL, LOW(0x410)          ; load the low byte of X with value 0x10
LDI XH, HIGH(0x410)         ; load the high byte of X with value 0x4
ST X+, R17                  ; copy R17 to memory location X
LDI R17, 0x78               ; R17 = 0x78
ST X+, R17                  ; copy R17 to memory location X

; task 3, need to add 0x1234 and 0x5678, then store into the EEPROM
LDI num1_L, 0x34            ; num1_L = 0x34
LDI num2_L, 0x56            ; num1_L = 0x56
LDI num1_H, 0x12            ; num1_H = 0x12
LDI num2_H, 0x78            ; num1_H = 0x78
ADD num1_L, num2_L          ; adding the lower bytes of number
ADC num1_H, num2_H          ; adding the upper bytes of number

LDI XH, HIGH(0x20)          ; load the high byte of X with value 0x0
LDI XL, LOW(0x20)           ; load the low byte of X with value 0x2
mov R22, num1_L
CALL STORE_IN_EEPROM        ; call the label
```

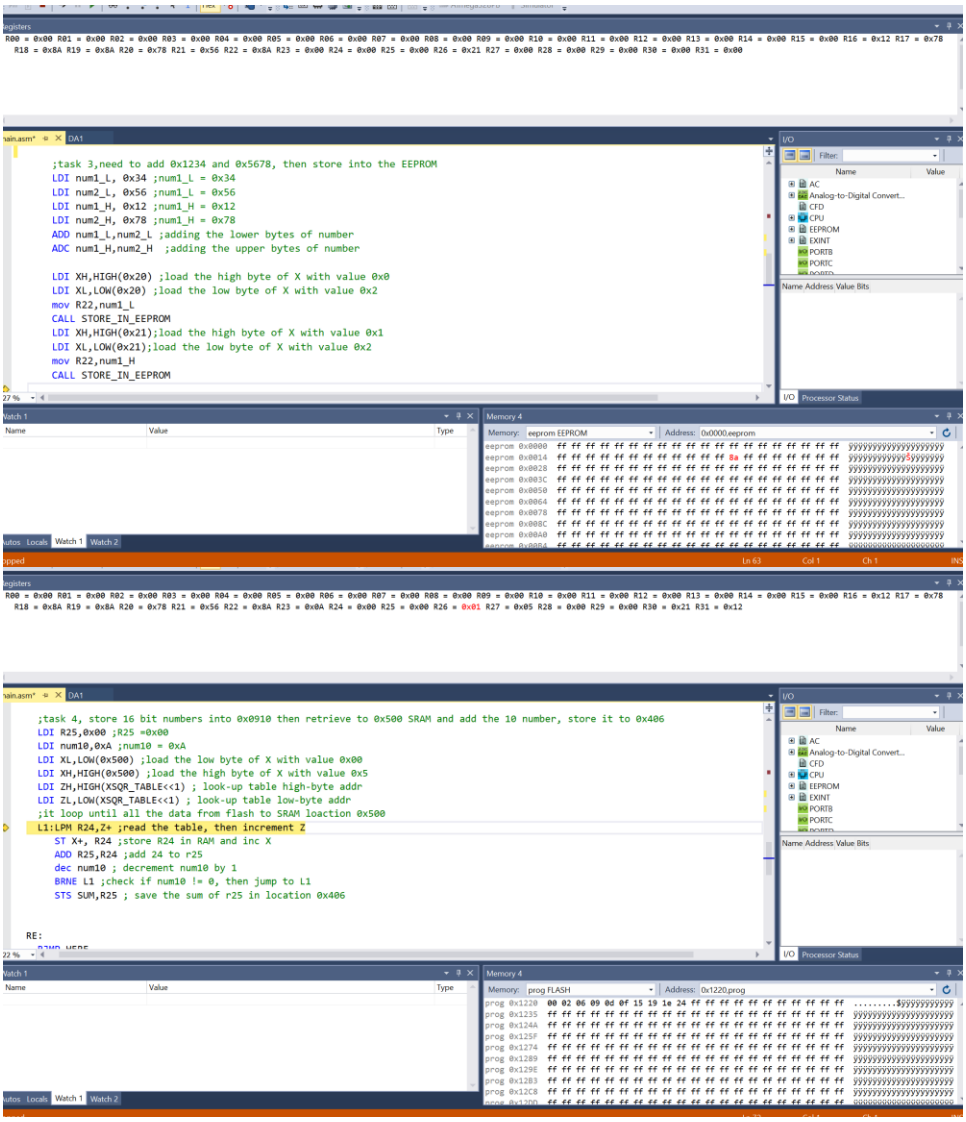
```
LDI XH, HIGH(0x21)              ; load the high byte of X with value 0x1
LDI XL, LOW(0x21)          ; load the low byte of X with value 0x2
mov R22, num1_H
CALL STORE_IN_EEPROM       ; call the label

; task 4, store 16-bit numbers into 0x0910 then retrieve to 0x500 SRAM and add the 10    ;
number, store it to 0x406
LDI R25,0x00                        ; R25 =0x00
LDI num10,0xA                       ; num10 = 0xA
LDI XL, LOW(0x500)                  ; load the low byte of X with value 0x00
LDI XH, HIGH(0x500)                 ; load the high byte of X with value 0x5
LDI ZH, HIGH(XSQR_TABLE<<1)                 ; look-up table high-byte adder
LDI ZL, LOW(XSQR_TABLE<<1)          ; look-up table low-byte adder
L1: LPM R24, Z+                     ; read the table, then increment Z
  ST X+, R24                   ; store R24 in RAM and increment X
  ADD R25, R24                 ; add 24 to r25
  dec num10                    ; decrement num10 by 1
  BRNE L1                      ; check if num10! = 0, then jump to L1
  STS SUM, R25                 ; save the sum of r25 in location 0x406


HERE:
RJMP HERE

; part of task 3
STORE_IN_EEPROM:
SBIC EECR, EEPE
RJMP STORE_IN_EEPROM
OUT EEARH, XH
OUT EEARL, XL
OUT EEDR, R22
SBI EECR, EEMPE
SBI EECR, EEPE
RET



; part of task 4
.ORG 0x0910
XSQR_TABLE:
.DB 0,2,6,9,13,15,21,25,30,36
```

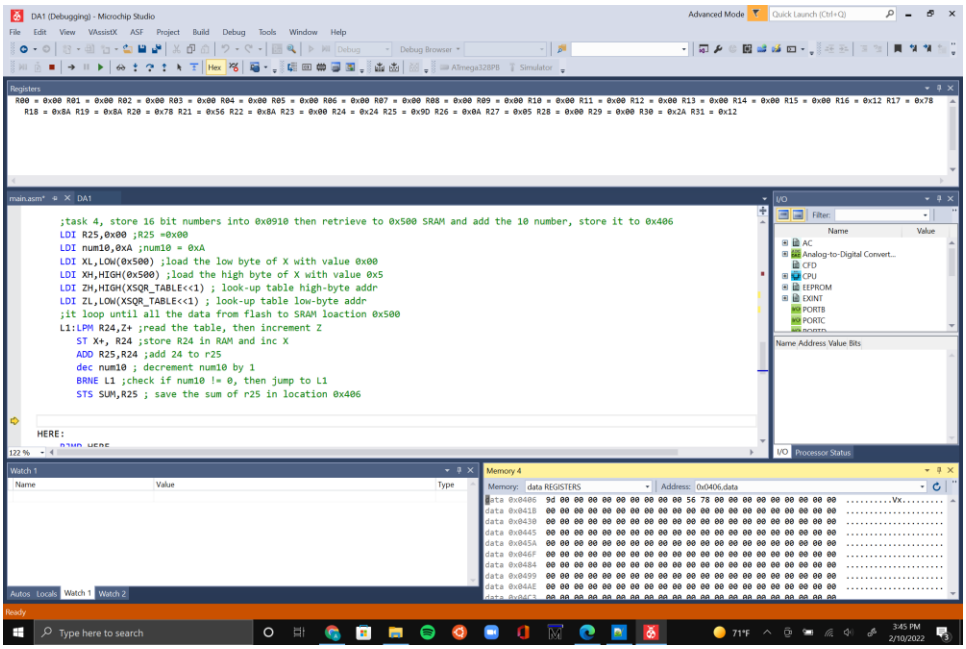
4. SCHEMATICS

N/A

5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)




R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00 R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0x12 R17 = 0x78
R18 = 0x8A R19 = 0x8A R20 = 0x78 R21 = 0x56 R22 = 0x8A R23 = 0x00 R24 = 0x00 R25 = 0x00 R26 = 0x21 R27 = 0x00 R28 = 0x00 R29 = 0x00 R30 = 0x00 R31 = 0x00
;task 3,need to add 0x1234 and 0x5678, then store into the EEPROM
LDI num1_L, 0x34 ;num1_L = 0x34
LDI num2_L, 0x56 ;num1_L = 0x56
LDI num1_H, 0x12 ;num1_H = 0x12
LDI num2_H, 0x78 ;num1_H = 0x78
ADD num1_L,num2_L ;adding the lower bytes of number
ADC num1_H,num2_H ;adding the upper bytes of number
LDI XH,HIGH(0x20) ;load the high byte of X with value 0x0
LDI XL,LOW(0x20) ;load the low byte of X with value 0x2
mov R22,num1_L
CALL STORE_IN_EEPROM
LDI XH,HIGH(0x21);load the high byte of X with value 0x1
LDI XL,LOW(0x21);load the low byte of X with value 0x2
mov R22,num1_H
CALL STORE_IN_EEPROM
I/O
Filter:
Name
Value
AC
Analog-to-Digital Convert...
CFD
CPU
EEPROM
EXINT
PORTB
PORTC
Name Address Value Bits
I/O
Processor Status
Watch 1
Name
Value
Type
Memory 4
Memory:
eeprom EEPROM
Address:
0x0000.eeprom
eeprom 0x0000 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x0014 ff ff ff ff ff ff ff ff ff ff ff ff 8a ff ff ff ff ff yyyyyyyyyyy5yyyyyy
eeprom 0x0028 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x003C ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x0050 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x0064 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x0078 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x008C ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x00A0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
eeprom 0x00B4 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
Autos
Locals
Watch 1
Watch 2
Ln 63
Col 1
Ch 1
INS
R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00 R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0x12 R17 = 0x78
R18 = 0x8A R19 = 0x8A R20 = 0x78 R21 = 0x56 R22 = 0x8A R23 = 0x0A R24 = 0x00 R25 = 0x00 R26 = 0x01 R27 = 0x05 R28 = 0x00 R29 = 0x00 R30 = 0x21 R31 = 0x12
;task 4, store 16 bit numbers into 0x0910 then retrieve to 0x500 SRAM and add the 10 number, store it to 0x406
LDI R25,0x00 ;R25 =0x00
LDI num10,0xA ;num10 = 0xA
LDI XL,LOW(0x500) ;load the low byte of X with value 0x00
LDI XH,HIGH(0x500) ;load the high byte of X with value 0x5
LDI ZH,HIGH(XSQR_TABLE<<1) ; look-up table high-byte addr
LDI ZL,LOW(XSQR_TABLE<<1) ; look-up table low-byte addr
;it loop until all the data from flash to SRAM loaction 0x500
L1:LPM R24,Z+ ;read the table, then increment Z
ST X+, R24 ;store R24 in RAM and inc X
ADD R25,R24 ;add 24 to r25
dec num10 ; decrement num10 by 1
BRNE L1 ;check if num10 != 0, then jump to L1
STS SUM,R25 ; save the sum of r25 in location 0x406
RE:
I/O
Filter:
Name
Value
AC
Analog-to-Digital Convert...
CFD
CPU
EEPROM
EXINT
PORTB
PORTC
Name Address Value Bits
I/O
Processor Status
Watch 1
Name
Value
Type
Memory 4
Memory:
prog FLASH
Address:
0x1220.prog
prog 0x1220 00 02 06 09 0d 0f 15 19 1e 24 ff ff ff ff ff ff ff ff ff ff ....$yyyyyyyyyy
prog 0x1235 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x124A ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x125F ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x1274 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x1289 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x129E ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x12B3 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
prog 0x12C8 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff yyyyyyyyyyyyyyyyyy
Autos
Locals
Watch 1
Watch 2



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

N/A

7. VIDEO LINKS OF EACH DEMO

N/A

8. GITHUB LINK OF THIS DA

https://github.com/AngeloNol/DA_submission.git