CPE301 – SPRING 2023

Design Assignment 1

Student Name: David Lenzin
Student #: 2001654470
Student Email: lenzin@unlv.nevada.edu
Primary Github address: https://github.com/dlenzin15/submissions
Directory: submission/DA1

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

No hardware components were used. Only Microchip Studio 7 was used to write code that was simulated.

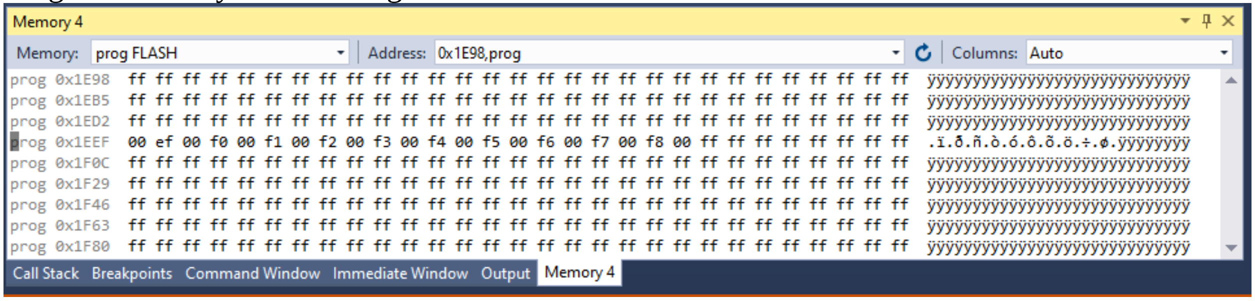
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
;
; AssemblerApplication1.asm
;
; Created: 2/19/2023 7:13:40 PM
; Author : lenzin
;

.include "m328pbdef.inc"

.org 0x0F77          //Loads to address 2*(0x0F77) = 0x1EEE
data2: .db 0xFF      //Padding byte to bring us to address 0x1EEF
data1: .dw 0x00EF, 0x00F0, 0x00F1, 0x00F2, 0x00F3, 0x00F4, 0x00F5, 0x00F6, 0x00F7,
0x00F8 //Load the 10 16-bit numbers into program memory at address 0x1EEF

.org 0x0             //Return to address 0x0000

//Clear registers to be used for calculation. 32-bit answer will be stored in
R20:R21:R22:R23
ldi r20, 0x00
ldi r21, 0x00
ldi r22, 0x00
ldi r23, 0x00

//Reset Z-pointer
ldi zl, 0xEF
ldi zh, 0x1E  //Initialize Z-pointer to memory address 0x1EEF

//Reset Counter
ldi r24, 0x0A

l2:
       //Load the number to add to the sum with the Z-pointer
       ldi r16, 0x00
       ldi r17, 0x00 //Add 2 bytes to make this a 32-bit number
       lpm r18, z+          //Load upper byte to r18
       lpm r19, z+         //Load lower byte to r19

       //Reset X and Y pointers. X-pointer points to the SRAM, Y-pointer points to the
EEPROM
       ldi xl, 0x00
       ldi xh, 0x05 //Note: SRAM starts at address 0x0100 and ends at address 0x08FF.
The middle of the SRAM is at address 0x0500
       ldi yl, 0x00
       ldi yh, 0x02 //Middle of EEPROM is at 0x0200

       //Load running sum into registers
       ld r20, x+
       ld r21, x+
       ld r22, x+
       ld r23, x

       //Add to the running sum
       add r20, r19
       adc r21, r18
       adc r22, r17
       adc r23, r16

       //Reset X-Pointer
       ldi xl, 0x00
       ldi xh, 0x05

       //Save new sum to the SRAM and EEPROM
       st x, r20                         //Save byte to register
       call store_in_eeprom //Save to EEPROM
       inc xl                            //Increment x-pointer for next instruction
       inc yl                            //Increment y-pointer for next instruction

       st x, r21                         //Save byte to register
       call store_in_eeprom //Save to EEPROM
       inc xl                            //Increment x-pointer for next instruction
       inc yl                            //Increment y-pointer for next instruction

       st x, r22                         //Save byte to register
       call store_in_eeprom //Save to EEPROM
       inc xl                            //Increment x-pointer for next instruction
       inc yl                            //Increment y-pointer for next instruction

       st x, r23                         //Save byte to register
       call store_in_eeprom //Save to EEPROM
       inc xl                            //Increment x-pointer for next instruction
       inc yl                            //Increment y-pointer for next instruction

       dec r24             //Decrement loop counter
       brne l2             //Loop if counter does not equal 0
       jmp end             //If loop is finished, jump to end

store_in_eeprom:
       SBIC EECR, EEPE                   //Wait until EEPE becomes 0
       RJMP store_in_eeprom              //Loop until EEPE becomes 0
       ld r25, x                         //Load running sum value into r25
       OUT EEARH, YH                     //Write new EEPROM address (high)
       OUT EEARL, YL                     //Write new EEPROM address (low)
       OUT EEDR, r25                     //Write sum to EEPROM data register
```

```
        SBI EECR, EEMPE                     //Set EEMPE to 1
        SBI EECR, EEPE                      //Set EEPE to 0
        ret                                         //Return from function

end: jmp end                        //End of program
```

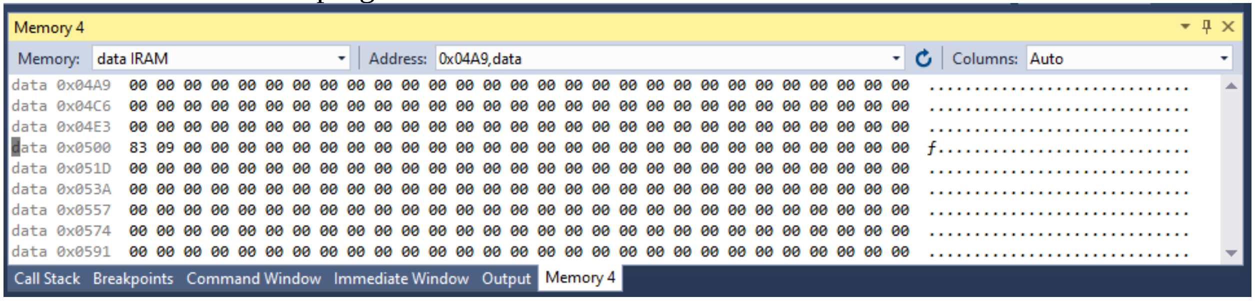
3. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Program Memory after loading the values:



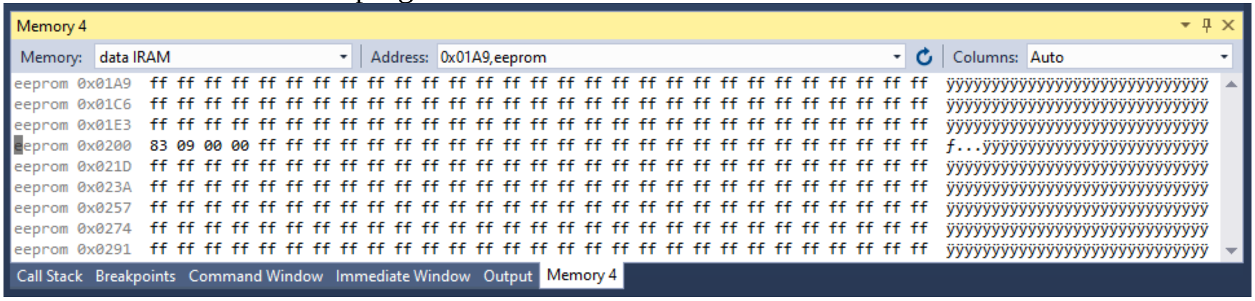
```
Memory 4
Memory: prog FLASH          Address: 0x1E98,prog          Columns: Auto
prog 0x1E98  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1EB5  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1ED2  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1EEF  00 ef 00 f0 00 f1 00 f2 00 f3 00 f4 00 f5 00 f6 00 f7 00 f8 00 ff ff ff ff ff ff ff ff  .ï.ð.ñ.ò.ó.ô.õ.ö.÷.ø.ÿÿÿÿÿÿÿÿ
prog 0x1F0C  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1F29  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1F46  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1F63  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x1F80  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

SRAM at the end of the program:



```
Memory 4
Memory: data IRAM          Address: 0x04A9,data          Columns: Auto
data 0x04A9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x04C6  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x04E3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x0500  83 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ƒ............................
data 0x051D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x053A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x0557  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x0574  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
data 0x0591  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .............................
```

EEPROM at the end of the program:



```
Memory 4
Memory: data IRAM          Address: 0x01A9,eeprom          Columns: Auto
eeprom 0x01A9  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x01C6  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x01E3  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x0200  83 09 00 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ƒ...ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x021D  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x023A  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x0257  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x0274  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
eeprom 0x0291  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

4. GITHUB LINK OF THIS DA

https://github.com/dlenzin15/submissions/tree/main/DA1

Student Academic Misconduct Policy

http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".

David Lenzin