

**Khalifa University of Science, Technology and Research**

**Electronic Engineering Department**

**ELCE332 Microprocessor Systems laboratory**

**Laboratory Experiment 3**

**HCS12 Input and Output Ports**

**Lab Partners**

Hamad Eisa Alhazami 100036985

Jasim Alhammadi 100035510

**Date Experiment Performed**: 11-2-2015

**Date Lab Report Submitted**: 18-2-2015

**Lab Instructor: Mr. Mohammed Alzaabi/ Mr. Mahmoud Khounji**

**Spring 2015**

# Summary

In this report, the objectives achieved after conducting the four main tasks in the laboratory experiment are discussed. The first task explained how to use LEDs and the 8 bits in accumulator A to derive output lines. The second task was about reading input lines, this showcases the use of DIP switches. The third task was a mix up of task 1 and 2 where the objective was to read DIP switches and write them to LEDs. The fourth and final task was mainly about using subroutines to cause a time delay in the output.

Contents

[**Summary 2**](#_Toc412029669)

[**1. Introduction 4**](#_Toc412029670)

[**1.1 Aim 4**](#_Toc412029671)

[**1.2 Objectives 4**](#_Toc412029672)

[**2. Design and Results 5**](#_Toc412029673)

[**2.2 Task 1: Deriving output lines 5**](#_Toc412029674)

[**2.2 Task 2: Reading input lines 6**](#_Toc412029675)

[**2.3 Task 3: Reading DIP switches and writing them to LEDs 7**](#_Toc412029676)

[**2.4 Task 4: Using subroutine to implement delay 8**](#_Toc412029677)

[**3. Assignment Questions 10**](#_Toc412029678)

[**4. Analysis and Interpretation 13**](#_Toc412029679)

[**5. Conclusions and Recommendations 14**](#_Toc412029680)

# Introduction

The HCS12 microcontroller consists of various components; in addition to that, it includes a fully integrated suite of I/O capabilities. These capabilities are a parallel and serial I/O, analog input and timer functions. Dual purpose functions are shared by the different I/O pins in the microcontroller; those functions are done by using a set of 1024 control registers which are located at memory locations $0000-$03FF.

The tasks in this experiment mainly focused on specific components in the microcontroller which are parallel ports, Light Emitting Diodes (LED), DIP switches and push buttons.

Parallel ports serve as general purpose I/O ports; they function as either input or output ports. There are nine parallel ports with the names A, B, E, AD, H, J, K, P and T, out of these nine parallel ports, this experiment concentrated on the LEDs controlled by ports B, J and P in addition to the DIP switches and the port H controlled push buttons.

## **1.1 Aim**

To introduce the students to the read and write data from the input and output ports and how delays can be implemented using loops.

## **1.2 Objectives**

1- Understand the microcontroller IO ports.

2- Configure the ports as inputs or output.

3- Read and write data from input and output ports.

4- To examine the DIP switches of PTH for I/O programming on Dragon12+ Board.

5- To do I/O bit programming in HCS12 Assembly language.

6- To create binary counter on Dragon12+ Board.

7- Download, run, and test code on a Dragon12+ Board.

# 2. Design and Results

## **2.2 Task 1: Deriving output lines**

In the first task, it is required to write a program that activates the LEDs to be switched on/off for a certain value being set and then executed on Dragon 12 board. The board is connected to the PC and the code for the program is downloaded on the microcontroller. The code for the program is shown below.

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**; export symbols**

**XDEF Entry,**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$FF**

**STAA DDRB ;Make PORTB output**

**;PTJ1 controls the LEDs connected to PORTB**

**LDAA #$FF**

**STAA DDRJ ;Make PORTJ output**

**STAA DDRP ;Make PORTP output**

**LDAA #$00**

**STAA PTJ ;Turn off PTJ1 to allow the LEDs to show data**

**LDAA #$0F**

**STAA PTP ; Disable the 7-segment display**

**;-------Switch on LEDs connected to PORTB based on the Acc A value**

**LDAA #$33**

**STAA PORTB ;Store A into PORTB**

**BRA Entry**

The value #$55 (85 in decimal) is chosen to test the program, register PORTB is added to the data window on the program to trace the value and compare it with the value on the board. The program is executed and run on the board. As it is observed on the board, the correct LEDs are lit (LEDs: 1, 3, 5, 7) showing the value #$55 in binary (01010101) where the 1s are the lit LEDs and 0s are the turned-off LEDs. Also the same values were shown in PORTB in the program as 85 in decimal. The hexadecimal value is changed to $#33 (51 in decimal), and LEDs showed the same value in binary (00110011). When the line STAA DDRP is commented out, the 7-segment output is activated on the board.

## **2.2 Task 2: Reading input lines**

In the second task, it is required to write a program that shows the value in the data window after setting the DIP switches on the Dragon 12 board. The board is connected to the PC and the code for the program is downloaded on the microcontroller. The program’s code is shown below.

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**; export symbols**

**XDEF Entry,**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$0**

**STAA DDRH**

**LDAA PTH**

**BRA Entry**

Register PTH is added to the data window in order to trace its value and the value of Acc. A. The program is executed and run on the board. As observed in the data window, the value representing the binary input is shown in the PTH label (in decimal) in the data window and in Accumulator A. The code is modified in order to read bit 4 from PTH by adding the line:

**ANDA #%00000100**

As observed on the board, only bit 4 (LED 4) is turned on when the switches are switched on and off.

## **2.3 Task 3: Reading DIP switches and writing them to LEDs**

In the third task, it is required to write a program that activate the LEDs immediately by switching the DIP switches on the Dragon 12 board. The program will be a combination of the program from tasks 1 and 2. The board is connected to the PC and the code for the program is downloaded on the microcontroller. The code for the program is shown below.

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**; export symbols**

**XDEF Entry,**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$FF**

**STAA DDRB ;Make PORTB output**

**;PTJ1 controls the LEDs connected to PORTB**

**LDAA #$FF**

**STAA DDRJ ;Make PORTJ output**

**STAA DDRP ;Make PORTP output**

**LDAA #$00**

**STAA PTJ ;Turn off PTJ1 to allow the LEDs to show data**

**LDAA #$0F**

**STAA PTP ; Disable the 7-segment display**

**;-------Switch on LEDs connected to PORTB based on the Acc A value**

**LDAA #$0**

**STAA DDRH**

**LDAA PTH**

**STAA PORTB ;Store A into PORTB**

**BRA Entry**

The operation here is done by taking the value of register PTH for DIP switches and storing it in register PORTB for the LEDS. The branching (BRA) is used after the previous operation in order to continue executing the read/write from the switches to the LEDS. The program is executed and run on the board. As observed on the board, the switched on LEDs represent the binary 1 while the switched off LEDs represent binary 0.

## **2.4 Task 4: Using subroutine to implement delay**

In the fourth task, it is required to implement a time delay by modifying the program from task 3, the time delay will be induced after switching the switches (PTH) on Dragon 12 board. The board is connected to the PC and the code for the program is downloaded on the microcontroller. The code for the program is shown below.

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**; export symbols**

**XDEF Entry,**

**R1 EQU $1000**

**R2 EQU $1001**

**R3 EQU $1002**

**R4 EQU $1003**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$FF**

**STAA DDRB ;Make PORTB output**

**;PTJ1 controls the LEDs connected to PORTB**

**LDAA #$FF**

**STAA DDRJ ;Make PORTJ output**

**STAA DDRP ;Make PORTP output**

**LDAA #$00**

**STAA PTJ ;Turn off PTJ1 to allow the LEDs to show data**

**LDAA #$0F**

**STAA PTP ; Disable the 7-segment display**

**;-------Switch on LEDs connected to PORTB based on the Acc A value**

**LDAA #$0**

**STAA DDRH**

**LDAA PTH**

**JSR DELAY**

**STAA PORTB ;Store A into PORTB**

**BRA Entry**

**DELAY**

**PSHA ;Save Reg A on Stack**

**LDAA #100 ;Change this value to see**

**STAA R4 ;how fast LEDs shows data coming from PTH DIP switches**

**;--10 msec delay. The Serial Monitor works at speed of 48MHz with XTAL=8MHz on Dragon12+ board**

**;Freq. for Instruction Clock Cycle is 24MHz (1/2 of 48Mhz).**

**;(1/24MHz) x 10 Clk x240x100=10 msec. Overheads are excluded in this calculation.**

**L4 LDAA #100**

**STAA R3**

**L3 LDAA #10**

**STAA R2**

**L2 LDAA #240**

**STAA R1**

**L1 NOP ;1 Intruction Clk Cycle**

**NOP ;1**

**NOP ;1**

**DEC R1 ;4**

**BNE L1 ;3**

**DEC R2 ;Total Instr.Clk=10**

**BNE L2**

**DEC R3**

**BNE L3**

**DEC R4**

**BNE L4**

**;--------------**

**PULA ;Restore Reg A**

**RTS**

The delay is calculated by the equation:

Here, the branching inside the delay subroutine is used as a loop to consume cycles. i.e. each cycle is about 1 µs. The delay consists of three loops, each value of R registers shows the number of loop iterations. The program is executed and run on the board. As observed on the board, the effect of the delay can be noticed by modifying the value of R registers to large values.

# 3. Assignment Questions

**1) Create a program that implements a binary counter. Use the delay subroutine given in Task-4 to display the counting sequence on the LEDs connected to PORTB.**

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**; export symbols**

**XDEF Entry,**

**R1 EQU $1000**

**R2 EQU $1001**

**R3 EQU $1002**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$FF**

**STAA DDRB ;Make PORTB output**

**STAA DDRJ ;Make PORTJ output**

**LDAA #$0**

**STAA DDRH ;PTH as Input**

**STAA PTJ ;Turn off PTJ1 to allow the LEDs on PORTB**

**CLRB**

**;-------Get data from DIP switches connected to PORTH**

**LOOP:**

**INCB**

**BSR DELAY**

**STAB PORTB**

**BRA LOOP**

**DELAY:**

**PSHA ;**

**LDAA #120**

**STAA R3**

**L3 LDAA #100**

**STAA R2**

**L2 LDAA #240**

**STAA R1**

**L1 NOP**

**NOP**

**NOP**

**DEC R1**

**BNE L1**

**DEC R2 ;Total Instr.Clk=10**

**BNE L2**

**DEC R3**

**BNE L3**

**PULA ; Restore Reg A**

**RTS**

**2) Create a program that flash all the LED’s with a delay of 0.1 sec in between.**

**; Include derivative-specific definitions**

**INCLUDE 'derivative.inc'**

**;export symbols**

**XDEF Entry**

**DDR: EQU $0002**

**PTA: EQU $0000**

**R1: EQU $1000**

**R2: EQU $1001**

**R3: EQU $1002**

**Entry:**

**bset DDR,%11111111**

**bset PTA,%11111111**

**BSR DELAY**

**bclr PTA,%00000001**

**BSR DELAY**

**bclr PTA,%00000010**

**BSR DELAY**

**bclr PTA,%00000100**

**BSR DELAY**

**bclr PTA,%00001000**

**BSR DELAY**

**bclr PTA,%00010000**

**BSR DELAY**

**bclr PTA,%00100000**

**BSR DELAY**

**bclr PTA,%01000000**

**BSR DELAY**

**bclr PTA,%10000000**

**DELAY**

**PSHA ;**

**LDAA #100 ;**

**STAA R3 ;**

**L3 LDAA #1000**

**STAA R2**

**L2 LDAA #240**

**STAA R1**

**L1 NOP ;**

**NOP ;**

**NOP ;**

**DEC R1 ;**

**BNE L1 ;**

**DEC R2 ;Total Instr.Clk=10**

**BNE L2**

**DEC R3**

**BNE L3**

**PULA ;Restore Reg A**

**RTS**

**3) Re-examine the toggle program in assignment question 2 which flashes the LEDs of PORTB with 0.1 sec delay. Now, modify that program to get the byte of data from PTH switches and give it to R3 register of the DELAY loop. Run the program to show how you can set the time delay size using the PTH switches.**

**R1 EQU $1000**

**R2 EQU $1001**

**R3 EQU $1002**

**ORG $4000 ;Flash ROM address for Dragon12+**

**Entry:**

**LDAA #$FF**

**STAA DDRB ;Make PORTB output**

**STAA DDRJ ;Make PORTJ output, (Needed by Dragon12+)**

**LDAA #$0**

**STAA DDRH ;PTH as Input**

**STAA PTJ ;Turn off PTJ1 to allow the LEDs on PORTB ;to show data (Needed by Dragon12+)**

**;PTJ1 controls the LEDs connected to PORTB (For Dragon12+ ONLY)**

**;-------Get data from DIP switches connected to PORTH**

**LOOP1 BSR DELAY**

**LDAA #$55**

**STAA PORTB**

**BSR DELAY**

**LDAA #$AA ;toggling the sitched ON LEDs.**

**STAA PORTB**

**BRA LOOP1**

**DELAY:**

**PSHA ;Save Reg A on Stack**

**LDAA #100 ;Change this value to see**

**LDAB PTH**

**STAB R3 ;how fast LEDs shows data coming from PTH DIP switches**

**L3 LDAA #100**

**STAA R2**

**L2 LDAA #240**

**STAA R1**

**L1 NOP ;1 Intruction Clk Cycle**

**NOP ;1**

**NOP ;1**

**DEC R1 ;4**

**BNE L1 ;3**

**DEC R2 ;Total Instr.Clk=10**

**BNE L2**

**DEC R3**

**BNE L3**

**;--------------**

**PULA ;Restore Reg A**

**RTS**

**ORG $FFFE**

**DC.**

# 4. Analysis and Interpretation

In the first task, the value #$FF is stored in registers DDRB, DDRJ and DDRP to make PORTB, PORTJ and PORTP as outputs. To disable the 7-segment display, the HEX value $30F is loaded to register PTP. The value of PORTB is set by Accumulator A. In the second task, the value of the PTH is taken from register DDRH of DIP switches using accumulator A as an intermediate. In the third task, the value of the PORTB is taken from register DDRH of DIP switches using accumulator A as an intermediate. Thus, it combines the second task by reading from the board, and the first task by storing the value in PORTB (LEDs register). In the fourth task, the DELAY is branched before storing the value of accumulator A in the PORTB register. The delay consists of three nested loops in order to apply the multiplication of R1 \* R2 \* R3. The nested loop is used because the register cannot store a value greater than #$FF (255 in decimal), otherwise only one loop will do the job. Also, NOP instructions are used to consume cycles without performing any operation.

# 5. Conclusions and Recommendations

To conclude, the main purpose behind the tasks in this experiment was to show the uses of the numerous ports of the microcontroller. To be more specific, the tasks focused on introducing the basic techniques of using DIP switches as inputs and LEDs as outputs. Furthermore, a new concept was also illustrated in the final task; the idea was to use subroutines to induce a time delay when presenting the output.