Problem set: 1 Date: June 5, 2023

Introduction To Keil and Assembly Language

This set of experiments has the following objectives:

- Familiarization with the development environment for 89C5131A.
- Familiarization with the instruction set of 8051 family.

You should browse through the manual for Keil software and the data sheet for 89C5131A. 89C5131A has 256 bytes of RAM, of which the top 128 bytes can only be accessed using indirect addressing. Since the stack is always accessed through indirect addressing using the stack pointer, we shall locate the stack in this part of the memory. In our assembly programs, we shall initialize the stack pointer to 0CFH, which provides 32 bytes for the stack. It is common to place arrays also in this part of the memory, because arrays are also easily accessed using a pointer. The address range from 80H to 0CFH is available for this (since memory beyond this is reserved for the stack). Direct addressing in the address range 80H to 0FFH accesses special function registers, which control IO ports, timers, interrupts etc. In particular, the micro-controller board that we shall use has push button switches and LEDs connected to a port. The switches and the LEDs will be our primary medium for input/output during this set of experiments. Since we do not have a monitor program running on the board, most of the debugging should be carried out using simulation at this stage. To end the program, you put it in an endless loop.

1. Write an assembly language program to add two 16-bit numbers x and y.
   Use the following program as a starting point. Add your code in the ADD16 subroutine.

```
// – DO NOT CHANGE ANYTHING UNTIL THE **** LINE–//

ORG 00H
LJMP MAIN
ORG 100H
MAIN:
CALL ADD16
HERE:
SJMP HERE
ORG 130H

// *****************

ADD16:

// ADD YOUR CODE HERE

RET
END
```

   - The number x is stored at locations 60H and 61H, with its most significant byte (MSB) in 60H and the least significant byte in 61H.
   - The number y is stored at locations 62H (MSB) and 63H (LSB).
   - Since the result $z = x + y$ can be 17 bits long, store the result in memory locations 64H, 65H, 66H
   - For $z = z_{16}z_{15}z_{14}...z_3z_2z_1z_0$, where z_0 is the least significant bit and z_{16} is the most significant bit, the memory location 64H should have $0000000z_{16}$, the memory location 65H should have the bits $z_{15}z_{14}...z_8$, and the memory location 66H should have the bits $z_7z_6...z_0$.

2. Write an assembly language program to generate programmable software delay which is multiple of 250 microsecond. It should accept a 8 bit value as a count, stored at address 71H. The delay should be proportional to this count.In the sense, if the count value is 1 then delay will be 250us and if count value is 2 then delay will be 2 times of 250us and so on.The delay is generated by a countdown loop, which uses the instruction DJNZ (Decrement and Jump if NonZero).

   Write another version which uses a count up loop, using CJNE (compare and jump if not equal).In both cases, you will have to use a loop within a loop to get the delay required in order to manage a 8 bit count.

   You should assemble and debug these programs using Keil software on a PC or laptop. You should check that the program is operating correctly using single stepping and breakpoints provided by Keil Software.