CT Lab: ALU and Branch Instructions

1 Introduction

An **A**rithmetic **L**ogic **U**nit (ALU) can perform add, multiply, shift, rotate and compare operations. You have already used some of these operations in previous labs. In this lab you will learn to use additional ALU operations as well as branch instructions.

Implement a program for the conversion of BCD values into binary values. The numbers will be displayed in various ways.

2 Learning Objectives

- You can apply logical, arithmetic and shift instructions.
- You can implement a multiplication with the **`MULS`** instruction as well as with a combination of **`shift`** and **`add`** instructions.
- You can compare values with each other and apply corresponding branch instructions.

3 Tasks

3.1 Task 1 –BCD to Binary

Use the project frame *bcd.* Read a BCD input from the DIP-switches. The switches S3 to S0 shall contain the ones and the switches S11 to S8 shall contain the tens. The two digits shall be combined, so that the BCD value can be displayed on LED7 to LED0 and the corresponding binary value on LED15 to LED8. Additionally the BCD value shall be displayed on the 7-segment display DS1 / DS0 and the HEX value on DS3 / DS2 (See **Figure 1**).

You shall only enter valid BCD values. The program does not have to validate the input.

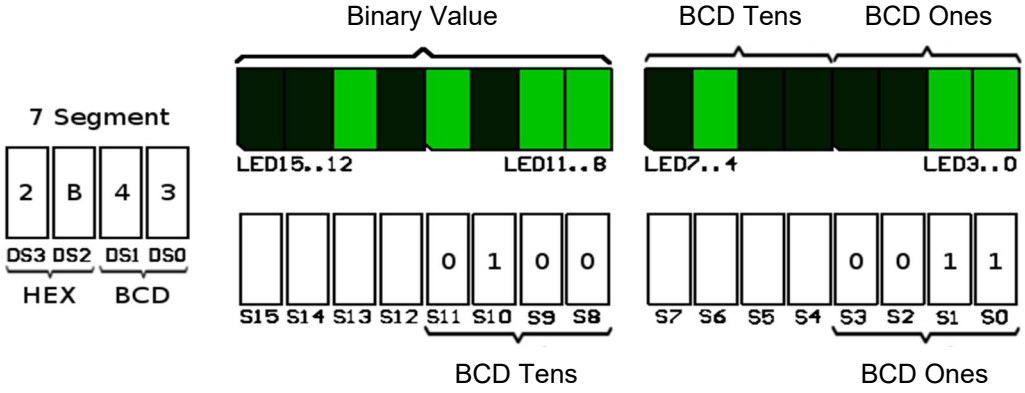

Figure 1: Function Example with *Tens = 4* and *Ones = 3*

Solve the task as follows:

a) Use the **`MULS`** instruction for the multiplication by 10.

b) Expand your implementation of the multiplication. If button T0 is pressed, use only **`shift`** and **`add`** instructions for the multiplication and set the background of the LCD to **red**. If button T0 is not pressed use the **`MULS`** instruction from a) and set the LCD background to **blue**.

- There are special addresses for accessing the display with binary values. Check the CT Board Wiki for the binary interface of the 7-segment displays.

3.2 Task 2 – Rotating „Disco Lights“

Expand your program, so that the total number of ‘1’ bits in the binary value is represented as a bar on the LEDs:

a) LED31 to LED 16 shall display a bar. The width of the bar from the right shall correspond to the number of LEDs that are turned on for LED15 to LED8. I.e. the width of the LED bar corresponds to the number of ones in the binary value.

b) Let this LED bar fully rotate once per main loop. The direction of the rotation shall be from **left** to **right**. Ones that “fall out” on the right side shall re-enter on the left side. Pause after every shift operation, so that the rotation is visible.

Figure 2 shows various stages of the rotating LED bar.

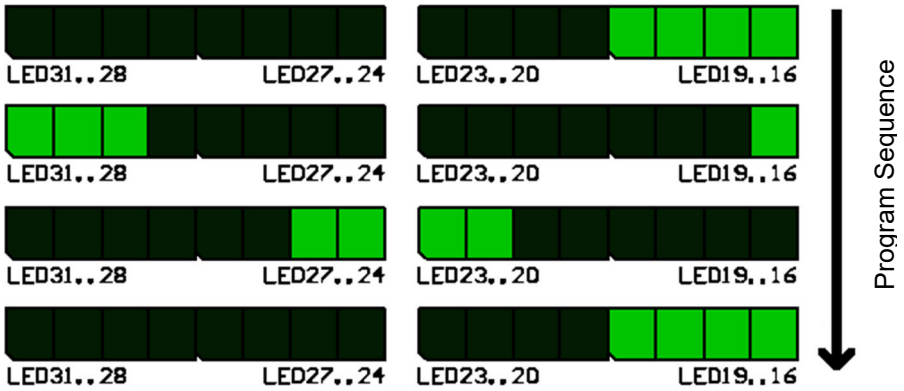

Figure 2: Rotating „Disco Lights“

- Use the predefined subroutine for your pause. It can be called using **`BL pause`**. With this subroutine no relevant registers get changed.
- LED31 to LED16 are aligned to a half word address. Use **`STRH`** for updating to avoid overwriting any other registers on the CT Board!
- How do you seamlessly rotate a half word (16 bit)? The processor only has a rotate instruction for a word (32 bit).
  Load the half word into the lower two bytes of a register. Duplicate the two bytes into the upper two bytes of the same register (using LSLS and ORRS). Then go ahead and use the 32 bit rotation.

4 Optional Tasks

4.1 Task 3 – Verifying if BCD Input Value is valid (optional)

Verify the validity of the value entered on the DIP switches. If the value is invalid all LEDs shall be turned off.

4.2 Task 4 – Observing Instruction Cycles (optional)

Assemble, link and load the project *optional*. Observe what happens at the end of each instruction in the registers and with the flags. This optional task is recommended because, amongst other things, it shows the implementation of a C switch-case instruction in assembly.

5 Grading

The working programs have to be presented to the lecturer. The student has to understand the solution / source code and has to be able to explain it to the lecturer.

Criteria	**Weight**
The program meets the requirements Task 1.	2/4
The program meets the requirements Task 2.	2/4