National Sun Yat-Sen University
ASSEMBLY LANGUAGE AND MICROCOMPUTER
Program Assignment #3
Due 11:59 PM Dec 28 2020

<**Programming Problem III**> Write an ARM assembly code to implement a ***deasm*** program which can partially deassembly the instruction contents of your program. Your program should identify every ***data processing and branch instruction*** written in a given program ***test.s***, and show its condition filed, instruction name, and the target operand. You also have to show the relative PC for each instruction.

For example, if you execute the program as follows:

deasm

Then the screen should display the following results：

PC	***condition***	***instruction***	***destination***
0	AL	ADD	r1
4	EQ	SUB	r2
8	AL	BL	320
12	EQ	MOV	r3
16	AL		
20	LT	CMP	r0

……

Here the instruction for PC=16 is not an ARM data processing nor branch instruction, so you just need to show ***PC*** and ***condition*** fields only. ARM compare instructions do not use the destination register field; however, you don't need to take care of these special cases. For the branch instructions (***B*** and ***BL***), the ***destination*** field shown will be the target PC value.

The program ***test.s*** will be given by using ***.include*** gcc assembly directive. In your assembly program, you should write something like:

```
                ……
                BL start_deasm
                .include 'test.s'
start_deasm:    …….
                ……..
```

The program ***test.s*** will be embedded, and complied along with your other part of the program. This test file has to be put along in the same directory with you ***deasm*** program.