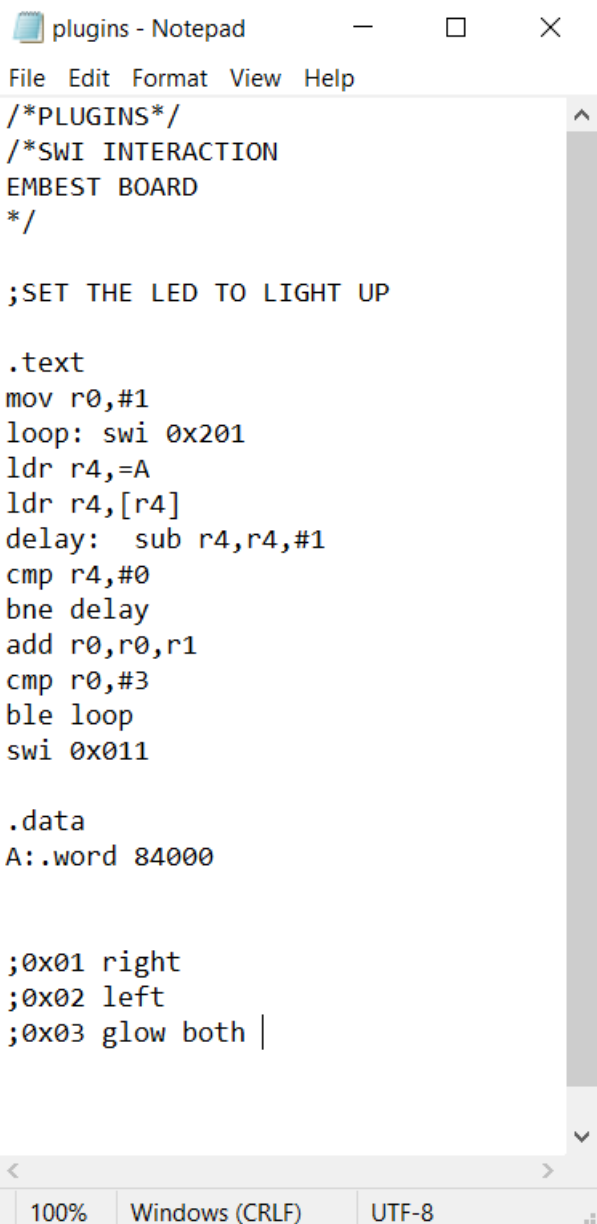




**Department of Computer Science & Engineering**  
**Microprocessor & Computer Architecture - UE20CS252**

No	Programs
Week No.7	<p>1. Demonstration of programs using plug-ins using ARMSIM.</p> <p>a. Set the LED to be light up.</p>  <p>The screenshot shows a Notepad window titled "plugins - Notepad" with a menu bar (File, Edit, Format, View, Help). The text area contains the following ARM assembly code:</p> <pre>/*PLUGINS*/ /*SWI INTERACTION EMBEST BOARD */  ;SET THE LED TO LIGHT UP  .text mov r0,#1 loop: swi 0x201 ldr r4,=A ldr r4,[r4] delay: sub r4,r4,#1 cmp r4,#0 bne delay add r0,r0,r1 cmp r0,#3 ble loop swi 0x011  .data A:.word 84000  ;0x01 right ;0x02 left ;0x03 glow both  </pre> <p>At the bottom of the window, there is a status bar showing "100%", "Windows (CRLF)", and "UTF-8".</p>

```

/*PLUGINS*/
/*SWI INTERACTION
EMBEST BOARD
*/

;SET THE LED TO LIGHT UP

.text
00001000:E3A00001 mov r0,#1
00001004:EF000201 loop: swi 0x201
00001008:E59F401C ldr r4,=A
0000100C:E5944000 ldr r4,[r4]
00001010:E2444001 delay: sub r4,r4,#1
00001014:E3540000 cmp r4,#0
00001018:1AFFFFFC bne delay
0000101C:E0800001 add r0,r0,r1
00001020:E3500003 cmp r0,#3
00001024:DAFFFFFF6 ble loop
00001028:EF000011 swi 0x011
    
```

## Student Exercises:

1. Execute the following programs on ARMSIM – PLUG-INS.

a. Display hexadecimal digits [0-9,A-F] on the 8 segment display.

;program to display 0 to F and f to 0 on the 8 segment display

.text

```
begin: mov r0,#0
mov r2,#0
again: swi 0x202 ; check whether black button pressed or not
cmp r0,#1; right button-upcounter
beq loop1
cmp r0,#2; left button- downcounter
beq loop2
b again
```

loop1:

```
mov r5,#16
ldr r1,=zero
```

back1:

```
ldrb r0,[r1]
swi 0x200 ;set 8 segment display to light up
bl delay
add r1,r1,#1
sub r5,r5,#1
cmp r5,#0
bne back1
b again
```

loop2:

```
mov r5,#16
ldr r1,=F
```

back2:

```
ldrb r0, [r1]
swi 0x200 ;set 8 set 8 segment display to light up
bl delay
add r1,r1,#1
sub r5,r5,#1
cmp r5,#0
bne back2
b again
```

delay:

```
mov r4, #64000
```

loop3:

```
sub r4,r4,#1
cmp r4, #0
bge loop3
mov pc, lr
```

.data

```
zero: .byte 0b11101101
one: .byte 0b01100000
two: .byte 0b01101110
three: .byte 0b11111010
four: .byte 0b00110011
five: .byte 0b10101011
six: .byte 0b10101111
seven: .byte 0b01110000
eight: .byte 0b11101111
nine: .byte 0b11100011
A: .byte 0b11100111
B: .byte 0b00101111
C: .byte 0b10001101
D: .byte 0b01101110
E: .byte 0b10001111
F: .byte 0b10000111
```

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView PluginUIView

General Purpose

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 : 000000  
R1 : 000000  
R2 : 000000  
R3 : 000000  
R4 : 000000  
R5 : 000000  
R6 : 000000  
R7 : 000000  
R8 : 000000  
R9 : 000000  
R10 (s1) : 000000  
R11 (fp) : 000000  
R12 (ip) : 000000  
R13 (sp) : 000054  
R14 (lr) : 000000  
R15 (pc) : 000010

CPSR Register  
Negative (N) : 0  
Zero (Z) : 0  
Carry (C) : 0  
Overflow (V) : 0  
IRQ Disable : 1  
FIQ Disable : 1  
Thumb (T) : 0  
CPU Mode : Sy

0x000000df

EmbestBoardPlugin

0.0 0.1 0.2 0.3  
1.0 1.1 1.2 1.3  
2.0 2.1 2.2 2.3  
3.0 3.1 3.2 3.3

8segment.s

```

:program to display 0 to F and f to 0 on the 8 segment display
.text
00001000:E3A00000    begin: mov r0,#0
00001004:E3A02000    mov r2,#0
00001008:EF000202    again: swi 0x202 ; check whether black button pressed or not
0000100C:E3500001    cmp r0,#1; right button-upcounter
00001010:0A000002    beq loop1
00001014:E3500002    cmp r0,#2; left button- downcounter
00001018:0A00000A    beq loop2
0000101C:EAF000F9    b again
00001020:          loop1:
00001020:E3A05010    mov r5,#16
00001024:E59F1058    ldr r1,=zero
00001028:          back1:
00001028:E5D10000    ldrb r0,[r1]
0000102C:EF000200    swi 0x200 ;set 8 segment display to light up
00001030:EB00000E    bl delay
00001034:E2811001    add r1,r1,#1

```

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:14651785 Elapsed Time:00:00:44.0545692  
Instructions per second:332582  
Execution starting ...

OutputView WatchView

b. Move a string from RIGHT to LEFT on the LCD display panel.

streaming - Notepad

File Edit Format View Help

|;streaming right to left

.text

```
mov r0,#30 ; r0=x
mov r1,#7 ; r1=y
mov r7,#0
ldr r8,=num
ldr r8,[r8]
ldr r2,=str
loop:
    swi 0x204
    bl sum
    cmp r0,#0
    subne r0,r0,#1
    swieq 0x11
    b loop
```

```
sum: cmp r7,r8
    addne r7,r7,#1
    bne sum
    swi 0x206
    mov r7,#0
    mov pc,lr
```

.data

```
str:.asciz "PESU"
num:.word 15000
```

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView PluginUI View

General Purpose f x

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 000000  
R1 : 000000  
R2 : 000000  
R3 : 000000  
R4 : 000000  
R5 : 000000  
R6 : 000000  
R7 : 000000  
R8 : 000000  
R9 : 000000  
R10 (s1): 000000  
R11 (fp): 000000  
R12 (ip): 000000  
R13 (sp): 000054  
R14 (lr): 000000  
R15 (pc): 000010

CPSR Register

Negative (N): 0  
Zero (Z): 0  
Carry (C): 0  
Overflow (V): 0  
IRQ Disable: 1  
FIQ Disable: 1  
Thumb (T): 0  
CPU Mode: Sy

0x000000df

streaming.s

```
;streaming right to left

.text

00001000:E3A0001E  mov r0,#30 ; r0=x
00001004:E3A01007  mov r1,#7 ; r1=y
00001008:E3A07000  mov r7,#0
0000100C:E59F8034  ldr r8,=num
00001010:E5980000  ldr r8,[r8]
00001014:E59F2030  ldr r2,=str
00001018:          loop:
00001018:EF000204      swi 0x204
0000101C:EB000003      bl sum
00001020:E3500000      cmp r0,#0
00001024:12400001      subne r0,r0,#1
00001028:0F000011      swieq 0x11
0000102C:EAF0FF9     b loop

00001030:E1570008  sum: cmp r7,r8
```

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:13467407 Elapsed Time:00:00:08.5808501  
Instructions per second:1569472  
Execution starting ...

24°C Partly sunny 08:25 26-03-2022

