

Microprocessor and Computer Architecture

UE20CS252

4th Semester, Academic Year 2021-22

Date: 27/03/2022

Name: Vishwa Mehul Mehta	SRN: PES2UG20CS389	Section: F
-----------------------------	-----------------------	---------------

Week# ____7____

Program Number: ____1____

Title of the Program

1. Write a program in ARM7TDMI-ISA to find the sum of all the digits in an 32bit number.

1. ARM Assembly Code(1)

```
1  .data
2  A: .word 1234
3  X: .word 0xCCCCCCCC
4  RES: .word 0
5
6  .text
7
8  ldr r0,=A
9  ldr r0,[r0] ; 1234
10 ldr r1,=X
11 ldr r1,[r1] ; magic num
12 ldr r2,=RES
13 mov r3,#0 ; result
14 loop: ; we check if no. less than 10 if yes then that must be added to the sum of digits
15     cmp r0,#10
16     blt add
17     mov r4,r0
18     b mod
19
20 mod: ; we divide by 10 if the number isn't less than 10 and find the mod
21     cmp r4,#10
22     blt div ; to divide by 10
23     sub r4,r4,#10
24     b mod
25
26 div: ; divide by 10
27     add r3,r3,r4 ; add the mod value to the result
"single_digit.s" 35L, 620C
1,1 Top
27
28     umull r5,r0,r0,r1 ; next 2 lines are for dividing the number to 10
29     mov r0,r0,lsr #3
30     b loop
31
32 add:
33     add r3,r3,r0
34     str r3,[r2]
35     .end
```

2. Output Screen Shot (1)

The screenshot displays the ARMSim ARM simulator interface. The main window is titled "single_digit.s" and contains assembly code. The left sidebar shows the "RegistersView" with a list of registers (R0-R15) and their values. The bottom section shows the "MemoryView" and "OutputView".

RegistersView:

Register	Value
R0	1
R1	3435973837
R2	4196
R3	-10
R4	2
R5	2576980380
R6	0
R7	0
R8	0
R9	0
R10 (s1)	0
R11 (fp)	0
R12 (ip)	0
R13 (sp)	21504
R14 (lr)	0
R15 (pc)	70656

CPSR Register:

Flag	Value
Negative (N)	1
Zero (Z)	0
Carry (C)	0
Overflow (V)	0
IRQ Disable	1
FIQ Disable	1
Thumb (T)	0
CPU Mode	System

Assembly Code (single_digit.s):

```
.data
0000105C: A: .word 1234
00001060: X: .word 0xCCCCCD
00001064: RES: .word 0

.text
00001000:E59F0048 ldr r0,=A
00001004:E5900000 ldr r0,[r0] ; 1234
00001008:E59F1044 ldr r1,=X
0000100C:E5911000 ldr r1,[r1] ; magic num
00001010:E59F2040 ldr r2,=RES
00001014:E3A03000 mov r3,#0 ; result
00001018: loop: ; we check if no. less than 10 if yes then that must be added to the sum of digits
00001018:E350000A cmp r0,#10
0000101C:BA000009 blt add
00001020:E1A04000 mov r4,r0
00001024:EAF0FFFF b mod
00001028: mod: ; we divide by 10 if the number isn't less than 10 and find the mod
```

MemoryView:

Address	Value
00001064	0000000A
00001068	81818181
0000106C	81818181
00001070	81818181
00001074	81818181
00001078	81818181
0000107C	81818181
00001080	81818181
00001084	81818181
00001088	81818181
0000108C	81818181
00001090	81818181
00001094	81818181
00001098	81818181
0000109C	81818181
000010A0	81818181
000010A4	81818181
000010A8	81818181
000010AC	81818181
000010B0	81818181
000010B4	81818181
000010B8	81818181
000010BC	81818181
000010C0	81818181
000010C4	81818181
000010C8	81818181
000010CC	81818181
000010D0	81818181
000010D4	81818181
000010D8	81818181
000010DC	81818181
000010E0	81818181
000010E4	81818181
000010E8	81818181
000010EC	81818181
000010F0	81818181
000010F4	81818181
000010F8	81818181
000010FC	81818181
00001100	81818181
00001104	81818181
00001108	81818181
0000110C	81818181
00001110	81818181
00001114	81818181
00001118	81818181
0000111C	81818181
00001120	81818181
00001124	81818181
00001128	81818181
0000112C	81818181
00001130	81818181
00001134	81818181
00001138	81818181
0000113C	81818181
00001140	81818181
00001144	81818181
00001148	81818181
0000114C	81818181
00001150	81818181
00001154	81818181
00001158	81818181
0000115C	81818181
00001160	81818181
00001164	81818181
00001168	81818181
0000116C	81818181
00001170	81818181
00001174	81818181
00001178	81818181
0000117C	81818181
00001180	81818181
00001184	81818181
00001188	81818181
0000118C	81818181
00001190	81818181
00001194	81818181
00001198	81818181
0000119C	81818181
000011A0	81818181
000011A4	81818181
000011A8	81818181
000011AC	81818181
000011B0	81818181
000011B4	81818181
000011B8	81818181
000011BC	81818181
000011C0	81818181
000011C4	81818181
000011C8	81818181
000011CC	81818181
000011D0	81818181
000011D4	81818181
000011D8	81818181
000011DC	81818181
000011E0	81818181
000011E4	81818181
000011E8	81818181
000011EC	81818181
000011F0	81818181
000011F4	81818181
000011F8	81818181
000011FC	81818181
00001200	81818181
00001204	81818181
00001208	81818181
0000120C	81818181
00001210	81818181
00001214	81818181
00001218	81818181
0000121C	81818181
00001220	81818181
00001224	81818181
00001228	81818181
0000122C	81818181
00001230	81818181
00001234	81818181
00001238	81818181
0000123C	81818181
00001240	81818181
00001244	81818181
00001248	81818181
0000124C	81818181
00001250	81818181
00001254	81818181
00001258	81818181
0000125C	81818181
00001260	81818181
00001264	81818181
00001268	81818181
0000126C	81818181
00001270	81818181
00001274	81818181
00001278	81818181
0000127C	81818181
00001280	81818181
00001284	81818181
00001288	81818181
0000128C	81818181
00001290	81818181
00001294	81818181
00001298	81818181
0000129C	81818181
000012A0	81818181
000012A4	81818181
000012A8	81818181
000012AC	81818181
000012B0	81818181
000012B4	81818181
000012B8	81818181
000012BC	81818181
000012C0	81818181
000012C4	81818181
000012C8	81818181
000012CC	81818181
000012D0	81818181
000012D4	81818181
000012D8	81818181
000012DC	81818181
000012E0	81818181
000012E4	81818181
000012E8	81818181
000012EC	81818181
000012F0	81818181
000012F4	81818181
000012F8	81818181
000012FC	81818181
00001300	81818181
00001304	81818181
00001308	81818181
0000130C	81818181
00001310	81818181
00001314	81818181
00001318	81818181
0000131C	81818181
00001320	81818181
00001324	81818181
00001328	81818181
0000132C	81818181
00001330	81818181
00001334	81818181
00001338	81818181
0000133C	81818181
00001340	81818181
00001344	81818181
00001348	81818181
0000134C	81818181
00001350	81818181
00001354	81818181
00001358	81818181
0000135C	81818181
00001360	81818181
00001364	81818181
00001368	81818181
0000136C	81818181
00001370	81818181
00001374	81818181
00001378	81818181
0000137C	81818181
00001380	81818181
00001384	81818181
00001388	81818181
0000138C	81818181
00001390	81818181
00001394	81818181
00001398	81818181
0000139C	81818181
000013A0	81818181
000013A4	81818181
000013A8	81818181
000013AC	81818181
000013B0	81818181
000013B4	81818181
000013B8	81818181
000013BC	81818181
000013C0	81818181
000013C4	81818181
000013C8	81818181
000013CC	81818181
000013D0	81818181
000013D4	81818181
000013D8	81818181
000013DC	81818181
000013E0	81818181
000013E4	81818181
000013E8	81818181
000013EC	81818181
000013F0	81818181
000013F4	81818181
000013F8	81818181
000013FC	81818181
00001400	81818181
00001404	81818181
00001408	81818181
0000140C	81818181
00001410	81818181
00001414	81818181
00001418	81818181
0000141C	81818181
00001420	81818181
00001424	81818181
00001428	81818181
0000142C	81818181
00001430	81818181
00001434	81818181
00001438	81818181
0000143C	81818181
00001440	81818181
00001444	81818181
00001448	81818181
0000144C	81818181
00001450	81818181
00001454	81818181
00001458	81818181
0000145C	81818181
00001460	81818181
00001464	81818181
00001468	81818181
0000146C	81818181
00001470	81818181
00001474	81818181
00001478	81818181
0000147C	81818181
00001480	81818181
00001484	81818181
00001488	81818181
0000148C	81818181
00001490	81818181
00001494	81818181
00001498	81818181
0000149C	81818181
000014A0	81818181
000014A4	81818181
000014A8	81818181
000014AC	81818181
000014B0	81818181
000014B4	81818181
000014B8	81818181
000014BC	81818181
000014C0	81818181
000014C4	81818181
000014C8	81818181
000014CC	81818181
000014D0	81818181
000014D4	81818181
000014D8	81818181
000014DC	81818181
000014E0	81818181
000014E4	81818181
000014E8	81818181
000014EC	81818181
000014F0	81818181
000014F4	81818181
000014F8	81818181
000014FC	81818181
00001500	81818181
00001504	81818181
00001508	81818181
0000150C	81818181
00001510	81818181
00001514	81818181
00001518	81818181
0000151C	81818181
00001520	81818181
00001524	81818181
00001528	81818181
0000152C	81818181
00001530	81818181
00001534	81818181
00001538	81818181
0000153C	81818181
00001540	81818181
00001544	81818181
00001548	81818181
0000154C	81818181
00001550	81818181
00001554	81818181
00001558	81818181
0000155C	81818181
00001560	81818181
00001564	81818181
00001568	81818181
0000156C	81818181
00001570	81818181
00001574	81818181
00001578	81818181
0000157C	81818181
00001580	81818181
00001584	81818181
00001588	81818181
0000158C	81818181
00001590	81818181
00001594	81818181
00001598	81818181
0000159C	81818181
000015A0	81818181
000015A4	81818181
000015A8	81818181
000015AC	81818181
000015B0	81818181
000015B4	81818181
000015B8	81818181
000015BC	81818181
000015C0	81818181
000015C4	81818181
000015C8	81818181
000015CC	81818181
000015D0	81818181
000015D4	81818181
000015D8	81818181
000015DC	81818181
000015E0	81818181
000015E4	81818181
000015E8	81818181
000015EC	81818181
000015F0	81818181
000015F4	81818181
000015F8	81818181
000015FC	81818181
00001600	81818181
00001604	81818181
00001608	81818181
0000160C	81818181
00001610	81818181
00001614	81818181
00001618	81818181
0000161C	81818181
00001620	81818181
00001624	81818181
00001628	81818181
0000162C	81818181
00001630	81818181
00001634	81818181
00001638	81818181
0000163C	81818181
00001640	81818181
00001644	818181

Microprocessor and Computer Architecture

UE20CS252

4th Semester, Academic Year 2021-22

Date: 27/03/2022

Name: Vishwa Mehul Mehta	SRN: PES2UG20CS389	Section: F
-----------------------------	-----------------------	---------------

Week# 7

Program Number: 2

Title of the Program

2. Write a program in ARM7TDMI-ISA to find the number of occurrences of a given character in a string.

Example:

Given string: My name is Bond.

Character: 'n'.

Expected output: Display 2 in a register.

1. ARM Assembly Code(1)

```
vishwa@Acer-Vishwa: /mnt/c/U  X  vishwa@Acer-Vishwa: /mnt/c/U  >
1  .data
2  S: .asciz "This is MPCA lab week 6."
3  C: .asciz "i"
4  RES: .word 0
5  .text
6  ldr r0,=S
7  ldr r1,=C
8  mov r2,#0 ; result
9  ldr r5,=RES
10 ldrb r4,[r1]
11
12 loop:
13     ldrb r3,[r0],#1
14     cmp r4,r3
15     addeq r2,r2,#1
16     cmp r3,#0
17     bne loop
18 str r2,[r5]
19 .end
```

2. Output Screen Shot (1)

The screenshot displays the ARMSim ARM Simulator interface. The main window is titled "ARMSim - The ARM Simulator Dept. of Computer Science". The interface is divided into several panes:

- RegistersView:** Shows the current state of the ARM registers. R0 is 0, R1 is 0, R2 is 2 (highlighted with a red box), R3 is 0, R4 is 105, R5 is 4180, R6 is 0, R7 is 0, R8 is 0, R9 is 0, R10 (s1) is 0, R11 (fp) is 0, R12 (ip) is 0, R13 (sp) is 21504, R14 (lr) is 0, and R15 (pc) is 70656. The CPSR Register shows Negative (N): 0, Zero (Z): 1, Carry (C): 1, Overflow (V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb (T): 0, and CPU Mode: System.
- Assembly Code:** Displays the assembly code for the program. The code includes a data section with a string "This is MPCA lab week 6." and a text section with instructions: `ldr r0,=S`, `ldr r1,=C`, `mov r2,#0 ; result`, `ldr r5,=RES`, `ldrb r4,[r1]`, a loop with `ldrb r3,[r0],#1`, `cmp r4,r3`, `addeq r2,r2,#1`, `cmp r3,#0`, `bne loop`, and `str r2,[r5]`.
- MemoryView:** Shows the memory contents at address 00001054. The memory is filled with the value 81818181.
- OutputView:** Shows the output of the program, which is currently empty.

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: Vishwa Mehul Mehta

Name: Vishwa Mehul Mehta

SRN: PES2UG20CS389

Section: F

Date: 27/03/2022