

Microprocessor and Computer Architecture (MPCA) Laboratory

UE20CS252 4th Semester,
Academic Year 2021-22

Date: 04/02/2022

Date: 28/01/2022

Name : SUNDEEP A	SRN: PES1UG20CS445
SEC: H	ROLL NO : 48

Week # 2

Program Number: 1

Title of the Program

1. Write a program in ARM7TDMI-ISA to find GCD of two numbers.
 - a. Assume operands to be in the CPU registers
 - b. Assume operands in the memory locations.

Program Code

1.a.

```
.TEXT
    MOV    R0,#6
    MOV    R1,#4
GCD: CMP   R0,R1
    BEQ    RES
    BLT    LOOP
    SUB    R0,R0,R1
    B      GCD

LOOP: SUB  R1,R1,R0
    B      GCD

RES: MOV   R2,R0
    SWI    0X011

.END
```

1.b

```
.DATA
    A: .WORD 6
    B: .WORD 4
.TEXT
    LDR R3,=A
    LDR R4,=B
    LDR R0,[R3]
    LDR R1,[R4]
GCD: CMP R0,R1
    BEQ RES
    BLT LOOP
    SUB R0,R0,R1
    B GCD

LOOP: SUB R1,R1,R0
    B GCD

RES: MOV R2,R0
    SWI 0X011

.END
```

Screenshot of ArmSimulator of the Program Executed

1.a

The screenshot displays the ArmSimulator interface. On the left, the 'General Purpose' register window shows the state of registers R0 through R15 and the CPSR. R0 and R1 are highlighted in red, indicating they are the current registers of interest. The CPSR shows flags like Negative (N), Zero (Z), Carry (C), and Overflow (V). The right panel shows the assembly code with the current instruction highlighted in blue. The bottom panel shows the 'OutputView' with the 'Console' tab selected, displaying execution statistics.

Register	Value
R0	00000002
R1	00000002
R2	00000000
R3	00000000
R4	00000000
R5	00000000
R6	00000000
R7	00000000
R8	00000000
R9	00000000
R10 (s1)	00000000
R11 (fp)	00000000
R12 (ip)	00000000
R13 (sp)	00005400
R14 (lr)	00000000
R15 (pc)	00001028

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

0x600000df

Assembly Code:

```
.TEXT
00001000:E3A00006    MOV R0,#6
00001004:E3A01004    MOV R1,#4
00001008:E1500001    GCD: CMP R0,R1
0000100C:0A000004    BEQ RES
00001010:BA000001    BLT LOOP
00001014:E0400001    SUB R0,R0,R1
00001018:EAF0FFFA    B GCD

0000101C:E0411000    LOOP: SUB R1,R1,R0
00001020:EAF0FFFA    B GCD

00001024:E1A02000    RES: MOV R2,R0
00001028:EF000011    SWI 0X011

.END
```

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:16 Elapsed Time:00:00:00.0109694
Instructions per second:1458

1.b

The screenshot displays an ARM7TDMI emulator interface. On the left, the 'General Purpose' register window shows R0 through R15 with their current values in hexadecimal. Below this is the 'CPSR Register' window showing status flags like Negative (N), Zero (Z), Carry (C), and Overflow (V). The main area shows the assembly code with memory addresses and instructions. The instruction at 00001030, 'SWI 0x011', is highlighted in blue. The 'OutputView' at the bottom shows the execution ending with an instruction count of 18 and an elapsed time of 00:00:00.0088966.

```
.DATA
0000103C:      A: .WORD 6
00001040:      B: .WORD 4

.TEXT
00001000:E59F302C      LDR R3,=A
00001004:E3A04D41      LDR R4,=B
00001008:E5930000      LDR R0,[R3]
0000100C:E5941000      LDR R1,[R4]
00001010:E1500001      GCD: CMP R0,R1
00001014:0A000004      BEQ RES
00001018:BA000001      BLT LOOP
0000101C:E0400001      SUB R0,R0,R1
00001020:EAffFFFA      B GCD

00001024:E0411000      LOOP: SUB R1,R1,R0
00001028:EAffFFF8      B GCD

0000102C:E1A02000      RES: MOV R2,R0
00001030:EF000011      SWI 0x011

00001034:0000103C      .END
00001038:00000000
```

CPSR Register
Negative (N) : 0
Zero (Z) : 1
Carry (C) : 1
Overflow (V) : 0
IRQ Disable: 1
FIQ Disable: 1
Thumb (T) : 0
CPU Mode : System
0x600000df

OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:18 Elapsed Time:00:00:00.0088966
Instructions per second:2023

Program Number: 2

Title of the Program

2. Write a program in ARM7TDMI-ISA to find the sum of N data items in the memory. Store the result in the memory location.

Program Code

- a. Use Pre-indexing addressing mode

```
.DATA
A: .WORD 10,20,30,40
SUM: .WORD 0

.TEXT
MOV R2,#0
LDR R1,=A
LDR R3,=SUM
MOV R6,#0
SUB R1,R1,#4
LOOP:
LDR R4,[R1,#4]
ADD R1,R1,#4
```

```

    ADD R2,R2,R4
    ADD R6,R6,#1
    CMP R6,#4
    BNE LOOP
STR R2,[R3]

.END

```

b. Use Post- Indexing addressing mode

```

.DATA
    A: .WORD 10,20,30,40
    SUM: .WORD 0
.TEXT
    MOV R2,#0
    LDR R1,=A
    LDR R3,=SUM
    MOV R6,#0

LOOP:
    LDR R4,[R1],#4
    ADD R2,R2,R4
    ADD R6,R6,#1
    CMP R6,#4
    BNE LOOP
STR R2,[R3]

.END

```

c. Use Auto-indexing addressing mode

```

d. .DATA
e.     A: .WORD 10,20,30,40
f.     SUM: .WORD 0
g. .TEXT
h.     MOV R2,#0
i.     LDR R1,=A
j.     LDR R3,=SUM
k.     MOV R6,#0
l.     SUB R1,R1,#4
m. LOOP:
n.     LDR R4,[R1,#4]!
o.     ADD R2,R2,R4

```

```

p.    ADD R6,R6,#1
q.    CMP R6,#4
r.    BNE LOOP
s.    STR R2,[R3]
t.
u.    .END
v.

```

Screenshot of ArmSimulator of the Program Executed

2.a

The screenshot displays the ArmSimulator interface for a program named `2_pre.s`. The **RegistersView** panel on the left shows the state of various registers. The **MemoryView5** panel shows a memory dump starting at address 00001048. The **OutputView** panel at the bottom shows the execution summary.

RegistersView:

- General Purpose: Floating Point
- Hexadecimal
- Unsigned Decimal
- Signed Decimal
- R0 : 0
- R1 : 0
- R2 : 100
- R3 : 4168
- R4 : 40
- R5 : 0
- R6 : 4
- R7 : 0
- R8 : 0
- R9 : 0
- R10 (sl) : 0
- R11 (fp) : 0
- R12 (ip) : 0
- R13 (sp) : 21504
- R14 (lr) : 0
- R15 (pc) : 70656

CPSR Register:

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

MemoryView5:

Address	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
00001048	00000064	100	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001070	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001098	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010C0	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181

OutputView:

```

Execution ending, Instruction Count:16658 Elapsed Time:00:00:00.2036326
Instructions per second:81804

```

2.b

The screenshot displays the ArmSimulator interface for a program named `2_pre.s`. The **RegistersView** panel on the left shows the state of various registers. The **MemoryView5** panel shows a memory dump starting at address 00001040. The **OutputView** panel at the bottom shows the execution summary.

RegistersView:

- General Purpose: Floating Point
- Hexadecimal
- Unsigned Decimal
- Signed Decimal
- R0 : 0
- R1 : 0
- R2 : 100
- R3 : 4160
- R4 : 40
- R5 : 0
- R6 : 4
- R7 : 0
- R8 : 0
- R9 : 0
- R10 (sl) : 0
- R11 (fp) : 0
- R12 (ip) : 0
- R13 (sp) : 21504
- R14 (lr) : 0
- R15 (pc) : 70656

CPSR Register:

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

MemoryView5:

Address	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
00001040	00000064	100	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001068	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
00001090	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181
000010B8	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181	81818181

OutputView:

```

Execution ending, Instruction Count:16655 Elapsed Time:00:00:00.1936708
Instructions per second:85996

```

2.c

The screenshot displays an ARM7TDMI emulator interface. The **RegistersView** window on the left shows the state of various registers: R0-R15, CPSR, and CPU Mode. R15 (PC) is highlighted at 70656. The **MemoryView** window shows a memory dump starting at address 00001044, with a search filter of 00001044. The **OutputView** window at the bottom shows the execution ending with an instruction count of 16655 and an elapsed time of 00:00:00.2034554.

RegistersView

General Purpose Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 0
R1 : 0
R2 : 100
R3 : 4164
R4 : 40
R5 : 0
R6 : 4
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
Negative (N) : 0
Zero (Z) : 1
Carry (C) : 1
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : System
0x600000df

2_auto.s

.DATA
00001034: A: .WORD 10,20,30,40
00001044: SUM: .WORD 0

.TEXT
00001000:E3A02000 MOV R2,#0
00001004:E59F1020 LDR R1,=A
00001008:E59F3020 LDR R3,=SUM
0000100C:E3A06000 MOV R6,#0
00001010:E2411004 SUB R1,R1,#4
00001014: LOOP:
00001014:E5B14004 LDR R4,[R1,#4]!
00001018:E0822004 ADD R2,R2,R4
0000101C:E2866001 ADD R6,R6,#1
00001020:E3560004 CMP R6,#4
00001024:1AFFFFFFFA BNE LOOP
00001028:E5832000 STR R2,[R3]
0000102C:00001034 .END
00001030:00001044

MemoryView5

Word Size
8Bit 16Bit 32Bit

00001044 00000064 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
0000106C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001094 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010BC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:16655 Elapsed Time:00:00:00.2034554
Instructions per second:81860

Program Number: 3

Title of the Program

Write a program in ARM7TDMI-ISA to find the sum of N data items at alternate [odd or **even** positions] locations in the memory. Store the result in the memory location.

Program Code

a. Use Pre-indexing addressing mode

```
;here we are finding the sum of n numbers at odd locations
.DATA
    A: .WORD 10,20,30,40,50,60
    SUM: .WORD 0
.TEXT
    MOV R2,#0
    LDR R1,=A
    LDR R3,=SUM
    MOV R6,#0
    SUB R1,R1,#8 ;change #8 to #4 to find the sum of numbers at even positions
LOOP:
```

```

    LDR R4,[R1,#8]
    ADD R1,R1,#8
    ADD R2,R2,R4
    ADD R6,R6,#2
    CMP R6,#6
    BNE LOOP
STR R2,[R3]

.END

```

b. Use Post- Indexing addressing mode

```

;here we are finding the sum of n numbers at odd locations
.DATA
    A: .WORD 10,20,30,40,50,60
    SUM: .WORD 0
.TEXT
    MOV R2,#0
    LDR R1,=A
    LDR R3,=SUM
    MOV R6,#0

LOOP:
    LDR R4,[R1],#8
    ADD R2,R2,R4
    ADD R6,R6,#2
    CMP R6,#6
    BNE LOOP
STR R2,[R3]

.END

```

c. Use Auto-indexing addressing mode

```

;here we are finding the sum of n numbers at odd locations
.DATA
    A: .WORD 10,20,30,40,50,60
    SUM: .WORD 0
.TEXT
    MOV R2,#0
    LDR R1,=A
    LDR R3,=SUM
    MOV R6,#0
    SUB R1,R1,#8;change the #8 to #4 to get the sum of n numbers at even
positions
LOOP:

```

```

LDR R4,[R1,#8]!
ADD R2,R2,R4
ADD R6,R6,#2
CMP R6,#6
BNE LOOP
STR R2,[R3]

.END

```

Screenshot of ArmSimulator of the Program Executed

3.a.

RegistersView

Register	Value
R0	0
R1	0
R2	90
R3	4176
R4	50
R5	0
R6	6
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

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

3_pre.s

```

.DATA
00001038:      A: .WORD 10,20,30,40,50,60
00001050:      SUM: .WORD 0
.TEXT
00001000:E3A02000      MOV R2,#0
00001004:E59F1024      LDR R1,=A
00001008:E59F3024      LDR R3,=SUM
0000100C:E3A06000      MOV R6,#0
00001010:E2411008      SUB R1,R1,#8;change #8 to #4 to find the sum of numbers at even positions
00001014:      LOOP:
00001014:E5914008      LDR R4,[R1,#8]
00001018:E2811008      ADD R1,R1,#8
0000101C:E0822004      ADD R2,R2,R4
00001020:E2866002      ADD R6,R6,#2
00001024:E3560006      CMP R6,#6
00001028:1AFFFFF9      BNE LOOP
0000102C:E5832000      STR R2,[R3]

00001030:00001038      .END
00001034:00001050

```

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:16652 Elapsed Time:00:00:00.1845053
Instructions per second:90252

3.b.

RegistersView

Register	Value
R0	0
R1	0
R2	90
R3	4168
R4	50
R5	0
R6	6
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

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

3_post.s

```

.DATA
00001030:      A: .WORD 10,20,30,40,50,60
00001048:      SUM: .WORD 0
.TEXT
00001000:E3A02000      MOV R2,#0
00001004:E59F101C      LDR R1,=A
00001008:E59F301C      LDR R3,=SUM
0000100C:E3A06000      MOV R6,#0

00001010:      LOOP:
00001010:E4914008      LDR R4,[R1],#8
00001014:E0822004      ADD R2,R2,R4
00001018:E2866002      ADD R6,R6,#2
0000101C:E3560006      CMP R6,#6
00001020:1AFFFFFFFA      BNE LOOP
00001024:E5832000      STR R2,[R3]

00001028:00001030      .END
0000102C:00001048

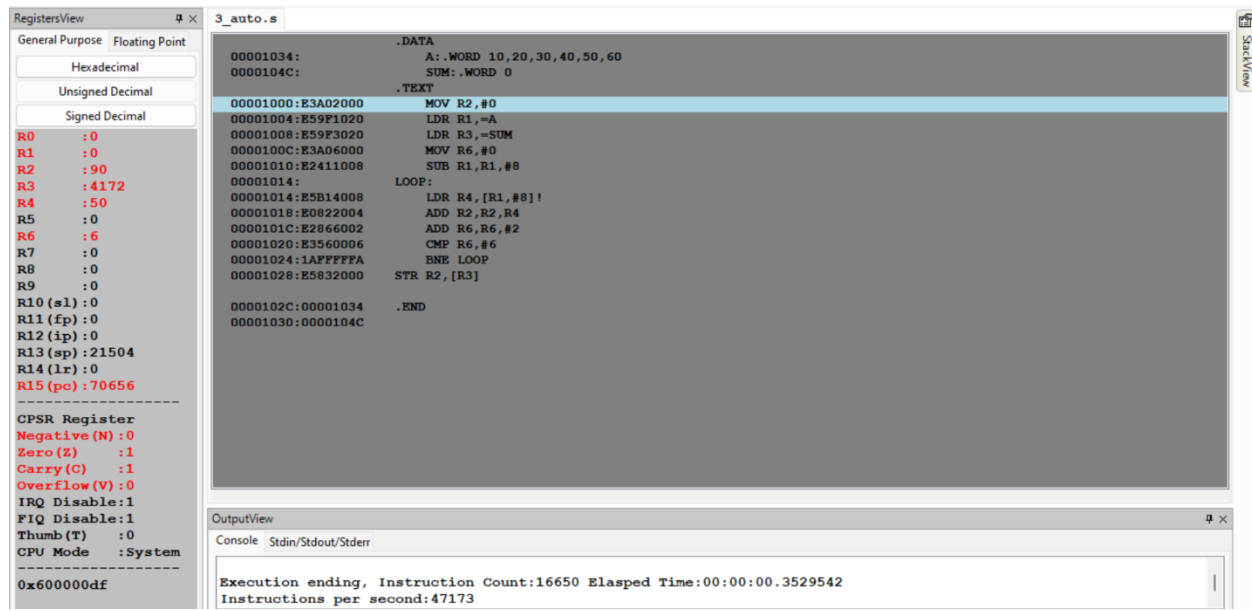
```

OutputView

Console Stdin/Stdout/Stderr

Execution ending, Instruction Count:16650 Elapsed Time:00:00:00.4766986
Instructions per second:34927

3.c.



Program Number: 4

Title of the Program

Write a program in ARM7TDMI-ISA to search for an element in an array.
Store 00 if the search is unsuccessful and 01 if the search is successful in the register.

a. Use Linear Search Technique

Program Code

```
.DATA
    A: .WORD 10,20,30,40,50
    KEY: .WORD 40
.TEXT
    LDR R0,=A
    LDR R1,=KEY
    LDR R5,[R1]
    MOV R4,#1
LOOP: LDR R2,[R0],#4;POST INDEXING
    CMP R5,R2
    BEQ FOUND
    ADD R4,R4,#1
    CMP R4,#5
    BNE LOOP
    MOV R3,#0;If key is not found then r3 contains 0
```

B EXIT

FOUND:MOV R3,#1;If key is found then r3 contains 1

EXIT:SWI 0X011

Screenshot of ArmSimulator of the Program Executed

The screenshot displays the ArmSimulator interface with three main panels:

- RegistersView:** Shows the state of 16 general-purpose registers (R0-R15) and the CPSR register. R0-R15 are in hexadecimal format. R0 is 4176, R1 is 4180, R2 is 40, R3 is 1, R4 is 4, R5 is 40, 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 4148. The CPSR register shows Negative (N) as 0, Zero (Z) as 1, Carry (C) as 1, Overflow (V) as 0, IRQ Disable as 1, FIQ Disable as 1, Thumb (T) as 0, and CPU Mode as System. The address 0x600000df is shown at the bottom.
- Linear.s:** Displays the assembly code for the program. The code includes data definitions for 'A' and 'KEY', and a loop that searches for 'KEY' in an array 'A'. If found, it sets R3 to 1 and branches to 'FOUND'. If not found, it branches to 'EXIT'. The 'FOUND' label sets R3 to 1, and the 'EXIT' label performs a software interrupt (SWI 0X011).

```
.DATA
00001040:      A: .WORD 10,20,30,40,50
00001054:      KEY: .WORD 40
.TEXT
00001000:E3A00D41      LDR R0,=A
00001004:E59F102C      LDR R1,=KEY
00001008:E5915000      LDR R5,[R1]
0000100C:E3A04001      MOV R4,#1
00001010:E4902004      LOOP: LDR R2,[R0],#4:POST INDEXING
00001014:E1550002      CMP R5,R2
00001018:0A000004      BEQ FOUND
0000101C:E2844001      ADD R4,R4,#1
00001020:E3540005      CMP R4,#5
00001024:1AFFFFF9      BNE LOOP
00001028:E3A03000      MOV R3,#0;If key is not found then r3 contains 0
0000102C:EAD00000      B EXIT
00001030:E3A03001      FOUND:MOV R3,#1;If key is found then r3 contains 1
00001034:      EXIT:SWI 0X011
```
- OutputView:** Shows the execution output. The console output indicates the execution ending with an instruction count of 27, an elapsed time of 00:00:00.0078192, and instructions per second of 3453.