

# H M Mythreya

## PES2UG20CS130

### MPCA-Lab Week-3

Task 1: Write a program in ARM7TDMI-ISA to find GCD of two numbers.

a) Operands in CPU registers

```
.TEXT
    MOV R0,#50
    MOV R1,#20

LOOP1:
    CMP R0,R1
    BEQ RES
    BLT LOOP2
    SUB R0,R0,R1
    B LOOP1

LOOP2:
    SUB R1,R1,R0
    B LOOP1

RES:
    MOV R2,R0
    SWI 0X011

.END
```

The screenshot displays two windows from an ARM7TDMI-ISA development environment. The **RegistersView** window on the left shows the state of the 16 general-purpose registers (R0-R15). R0 and R1 contain the values 10 and 20, respectively, which are the inputs for the GCD program. R15 (PC) is at address 4152. Below the registers, the CPSR register is shown with various flags: Negative (N) is 0, Zero (Z) is 1, Carry (C) is 1, Overflow (V) is 0, IRQ Disable is 1, FIQ Disable is 1, Thumb (T) is 0, and CPU Mode is System. The **CodeView** window on the right shows the assembly code for the program, with the instruction `SWI 0X011` at address 00001028 highlighted in blue. The code is organized into sections: `.TEXT`, `LOOP1:`, `LOOP2:`, `RES:`, and `.END...`.

Register	Value
R0	10
R1	20
R2	10
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10 (s1)	0
R11 (fp)	0
R12 (ip)	0
R13 (sp)	70656
R14 (lr)	0
R15 (pc)	4152

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

CodeView  
p1.o  
-----  
00001000:E3A00032 MOV R0,#50  
00001004:E3A01014 MOV R1,#20  
-----  
00001008:E1500001 LOOP1: CMP R0,R1  
0000100C:0A000004 BEQ RES  
00001010:BA000001 BLT LOOP2  
00001014:E0400001 SUB R0,R0,R1  
00001018:EAF00000 B LOOP1  
-----  
0000101C:E0411000 LOOP2: SUB R1,R1,R0  
00001020:EAF00000 B LOOP1  
-----  
00001024:E1A02000 RES: MOV R2,R0  
00001028:EF000011 SWI 0X011  
-----  
0000102C:00000000 .END...

## b) Operands in memory locations.

.DATA

A: .WORD 50  
B: .WORD 20

.TEXT

LDR R1,=A  
LDR R2,=B

LDR R3,[R1]

LDR R4,[R2]

LOOP1:

CMP R3,R4  
BEQ RES  
BLT LOOP2  
SUB R3,R3,R4  
B LOOP1

LOOP2:

SUB R4,R4,R3  
B LOOP1

RES:

MOV R3,R4  
SWI 0X011

.END

The screenshot displays a debugger interface with two main windows: **RegistersView** and **CodeView**.

**RegistersView:** This window shows the state of the processor registers. The **General Purpose** tab is selected, and the **Unsigned Decimal** view is chosen. The registers are listed as follows:

Register	Value
R0	0
R1	0
R2	4160
R3	10
R4	10
R5	0
R6	0
R7	0
R8	0
R9	0
R10 (s1)	0
R11 (fp)	0
R12 (ip)	0
R13 (sp)	70656
R14 (lr)	0
R15 (pc)	4152

Below the registers, the **CPSR Register** is shown with the following fields:

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

The **CodeView** window shows the assembly code for the program, with the **p1\_B.o** file loaded. The code is organized into sections: **.DATA**, **.TEXT**, and **.END**. The **.TEXT** section contains the main logic of the program, including the **LOOP1** and **LOOP2** loops, and the **RES** label. The **.END** section marks the end of the program.

The assembly code is as follows:

```
.DATA
0000103C:00000032      A: .WORD 50
00001040:00000014      B: .WORD 20

.TEXT
00001000:E59F102C      LDR R1,=A
00001004:E59F202C      LDR R2,=B

00001008:E5913000      LDR R3,[R1]
0000100C:E5924000      LDR R4,[R2]
00001010:E1530004      LOOP1:
00001014:0A000004      CMP R3,R4
00001018:BA000001      BEQ RES
0000101C:E0433004      BLT LOOP2
00001020:EAF00000      SUB R3,R3,R4
00001024:E0444003      B LOOP1

00001028:EAF00000      LOOP2:
0000102C:E1A03004      SUB R4,R4,R3
00001030:EF000011      B LOOP1

00001034:E1A03004      RES:
00001038:EF000011      MOV R3,R4
0000103C:EF000011      SWI 0X011

.END
00001040:00000000
```

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

a) Post-indexing addressing mode.

```
.DATA
    A: .WORD 23,12,16,17,5
```

```
.TEXT
    LDR R0,=A
    MOV R1,#0
    B LOOP1
```

```
LOOP1:
    LDR R2,[R0],#4
    ADD R3,R3,R2
    ADD R1,R1,#1
    CMP R1,#5
    BEQ LOOP2
    B LOOP1
```

```
LOOP2:
    SWI 0X011
```

The screenshot displays a debugger interface with two main panes: RegistersView on the left and CodeView on the right.

**RegistersView:**

- General Purpose registers: R0: 4160, R1: 0, R2: 5, R3: 73, R4: 0, R5: 0, R6: 0, R7: 0, R8: 0, R9: 0, R10 (s1): 0, R11 (fp): 0, R12 (ip): 0, R13 (sp): 70656, R14 (lr): 0, R15 (pc): 4140.
- 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.

**CodeView:**

The CodeView pane shows the program code and data. The data section (.DATA) contains the array A: .WORD 23,12,16,17,5. The text section (.TEXT) contains the assembly code:

```
00001000:E59F0020    LDR R0,=A
00001004:E3A01000    MOV R1,#0
00001008:EAF00000    B LOOP1

LOOP1:
0000100C:E4902004    LDR R2,[R0],#4
00001010:E0833002    ADD R3,R3,R2
00001014:E2811001    ADD R1,R1,#1
00001018:E3510005    CMP R1,#5
0000101C:0A000000    BEQ LOOP2
00001020:EAF00000    B LOOP1

LOOP2:
00001024:EF000011    SWI 0X011...
```

b) Pre-indexing addressing mode.

```
.DATA
A: .WORD 23,12,16,17,5
B: .WORD 0

.TEXT
LDR R0,=A
LDR R5,=B
MOV R1,#0
MOV R4,#0
B LOOP1

LOOP1:
LDR R2,[R0,R4]
ADD R3,R3,R2
ADD R1,R1,#1
CMP R1,#5
BEQ LOOP2
ADD R4,R4,#4
B LOOP1

LOOP2:
STR R5,[R3]
SWI 0X011
```

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0	: 4160
R1	: 5
R2	: 5
R3	: 73
R4	: 16
R5	: 4180
R6	: 0
R7	: 0
R8	: 0
R9	: 0
R10 (s1)	: 0
R11 (fp)	: 0
R12 (ip)	: 0
R13 (sp)	: 70656
R14 (lr)	: 0
R15 (pc)	: 4144

-----

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

CodeView

p2 B.o

```
.DATA
00001040:00000017      A: .WORD 23,12,16,17,5
:0000000C
:00000010
:00000011
:00000005
00001054:00000000      B: .WORD 0

.TEXT
00001000:E59F0030      LDR R0,=A
00001004:E59F5030      LDR R5,=B
00001008:E3A01000      MOV R1,#0
0000100C:E3A04000      MOV R4,#0
00001010:EAF0FFFF      B LOOP1

LOOP1:
00001014:E7902004      LDR R2,[R0,R4]
00001018:E0833002      ADD R3,R3,R2
0000101C:E2811001      ADD R1,R1,#1
00001020:E3510005      CMP R1,#5
00001024:0A000001      BEQ LOOP2
00001028:E2844004      ADD R4,R4,#4
0000102C:EAF0FFF8      B LOOP1

LOOP2:
00001030:E5835000      STR R5,[R3]
00001034:EF000011      SWI 0X011...
:00000000
:00000014
```

3) 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

a) Post-indexing addressing mode.

```
.DATA
    A: .WORD 23,12,16,17,5
```

```
.TEXT
    LDR R0,=A
    MOV R1,#0
    B LOOP1
```

```
LOOP1:
    LDR R2,[R0],#8
    ADD R3,R3,R2
    ADD R1,R1,#1
    CMP R1,#3
    BEQ LOOP2
    B LOOP1
```

```
LOOP2:
    SWI 0X011
```

The screenshot displays an ARM7TDMI-ISA development environment with two main windows: RegistersView and CodeView.

**RegistersView:** This window shows the state of the ARM7TDMI-ISA registers. The 'General Purpose' tab is selected, and the 'Unsigned Decimal' view is chosen. The registers are listed as follows:

Register	Value
R0	4164
R1	0
R2	5
R3	44
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10 (s1)	0
R11 (fp)	0
R12 (ip)	0
R13 (sp)	70656
R14 (lr)	0
R15 (pc)	4140

Below the registers, the CPSR Register is shown with the following status:

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

The bottom of the window shows the value 0x600000df.

**CodeView:** This window shows the assembly code for the program. The file 'p3\_A.o' is open, and the code is displayed in hexadecimal and assembly format. The code is organized into sections: .DATA, .TEXT, and .LOOP1/.LOOP2.

```
.DATA
0000102C:00000017      A: .WORD 23,12,16,17,5
:0000000C
:00000010
:00000011
:00000005

.TEXT
00001000:E59F0020      LDR R0,=A
00001004:E3A01000      MOV R1,#0
00001008:EAF00000      B LOOP1

LOOP1:
0000100C:E4902008      LDR R2,[R0],#8
00001010:E0833002      ADD R3,R3,R2
00001014:E2811001      ADD R1,R1,#1
00001018:E3510003      CMP R1,#3
0000101C:0A000000      BEQ LOOP2
00001020:EAF00000      B LOOP1

LOOP2:
00001024:EF000011      SWI 0X011...
:00000000
```

## b) Pre-indexing addressing mode.

.DATA

A: .WORD 23,12,16,17,5

B: .WORD 0

.TEXT

LDR R0,=A

LDR R5,=B

MOV R1,#0

MOV R4,#0

B LOOP1

LOOP1:

LDR R2,[R0,R4]

ADD R3,R3,R2

ADD R1,R1,#1

CMP R1,#3

BEQ LOOP2

ADD R4,R4,#8

B LOOP1

LOOP2:

STR R5,[R3]

SWI 0X011

RegistersView

General Purpose
Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 0  
R1 : 0  
R2 : 0  
R3 : 0  
R4 : 0  
R5 : 0  
R6 : 0  
R7 : 0  
R8 : 0  
R9 : 0  
R10 (s1) : 0  
R11 (fp) : 0  
R12 (ip) : 0  
R13 (sp) : 70656  
R14 (lr) : 0  
R15 (pc) : 4096  
-----  
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 : System  
-----  
0x000000df

CodeView

p3\_B.o

.DATA  
00001040:00000017 A: .WORD 23,12,16,17,5  
:0000000C  
:00000010  
:00000011  
:00000005  
00001054:00000000 B: .WORD 0  
  
.TEXT  
00001000:E59F0030 LDR R0,=A  
00001004:E59F5030 LDR R5,=B  
00001008:E3A01000 MOV R1,#0  
0000100C:E3A04000 MOV R4,#0  
00001010:EAF00000 B LOOP1  
  
LOOP1:  
00001014:E7902004 LDR R2,[R0,R4]  
00001018:E0833002 ADD R3,R3,R2  
0000101C:E2811001 ADD R1,R1,#1  
00001020:E3510003 CMP R1,#3  
00001024:0A000001 BEQ LOOP2  
00001028:E2844008 ADD R4,R4,#8  
0000102C:EAF00000 B LOOP1  
  
LOOP2:  
00001030:E5835000 STR R5,[R3]  
00001034:EF000011 SWI 0X011...  
:00000000  
:00000014

4) 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) Linear-Search (Found condition)

.DATA

```
ARRAY: .WORD 17,20,6,25,1
SEARCH: .WORD 20
RESULT: .WORD 0
```

.TEXT

```
LDR R0,=ARRAY
LDR R1,=SEARCH
LDR R5,=RESULT
MOV R2,#0
```

LDR R4,[R1]

LOOP1:

```
    CMP R2,#5
    BEQ FINISH
    LDR R3,[R0],#4
    CMP R3,R4
    BEQ FOUND
    ADD R2,R2,#1
    B LOOP1
```

FOUND:

```
    MOV R6,#0
    STR R6,[R5]
    B END
```

FINISH:

```
    MOV R6,#1
    STR R6,[R5]
    B END
```

END:

```
    SWI 0X011
```

The screenshot displays an ARM7TDMI-ISA assembly simulator with three main panels: RegistersView, CodeView, and OutputView.

**RegistersView:** Shows the state of registers R0 through R15. R0 is 4192, R1 is 4192, R2 is 1, R3 is 20, R4 is 20, R5 is 4208, 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 70656, R14 (lr) is 0, and R15 (pc) is 4184. 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 memory address 0x600000df is also shown.

**CodeView:** Displays the assembly code with memory addresses. The .DATA section includes ARRAY (00001058:00000011), SEARCH (0000106C:00000014), and RESULT (00001070:00000000). The .TEXT section includes the main program logic, starting with LDR R0,=ARRAY (00001000:E59F0044) and ending with SWI 0X011 (00001048:EF000011).

**OutputView:** Shows the output of the program, with the label 'result' having a value of 0.

## Not found Condition:

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 4204

R1 : 4204

R2 : 5

R3 : 1

R4 : 5

R5 : 4208

R6 : 1

R7 : 0

R8 : 0

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 70656

R14 (lr) : 0

R15 (pc) : 4172

-----

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

CodeView

p4\_A.o

.DATA

00001058:00000011 ARRAY: .WORD 17,20,6,25,1

:00000014

:00000006

:00000019

:00000001

0000106C:00000005 SEARCH: .WORD 5

00001070:00000000 RESULT: .WORD 0

.TEXT

00001000:E59F0044 LDR R0,=ARRAY

00001004:E59F1044 LDR R1,=SEARCH

00001008:E59F5044 LDR R5,=RESULT

0000100C:E3A02000 MOV R2,#0

00001010:E5914000 LDR R4,[R1]

LOOP1:

00001014:E3520005 CMP R2,#5

00001018:0A000007 BEQ FINISH

0000101C:E4903004 LDR R3,[R0],#4

00001020:E1530004 CMP R3,R4

00001024:0A000001 BEQ FOUND

00001028:E2822001 ADD R2,R2,#1

0000102C:EAF000F8 B LOOP1

FOUND:

00001030:E3A06000 MOV R6,#0

00001034:E5856000 STR R6,[R5]

00001038:EA000002 B END

FINISH:

0000103C:E3A06001 MOV R6,#1

00001040:E5856000 STR R6,[R5]

00001044:EAF000F8 B END

END:

00001048:EF000011 SWI 0X011

0000104C:00000000

:00000014

:00000018

OutputView WatchView

Label Value

result 1



## a) Binary-Search (Found condition)

.DATA

ARRAY: .WORD 1,6,17,20,25

SEARCH: .WORD 6

RESULT: .WORD 0

.TEXT

LDR R0,=ARRAY

LDR R1,=SEARCH

LDR R7,=RESULT

MOV R2,#1

MOV R3,#4

LDR R4,[R1]

LOOP1:

CMP R2,R3

BEQ FINISH

ADD R5,R2,R3

MOV R5,R5,LSR #1

MOV R5,R5,LSL #2

LDR R6,[R0,R5]

CMP R4,R6

BEQ FOUND

BLT LESSTHAN

BGT GREATERTHAN

B LOOP1

FOUND:

MOV R8,#0

STR R8,[R7]

B END

LESSTHAN:

MOV R3,R5

MOV R3,R3,LSR #2

B LOOP1

GREATERTHAN:

MOV R2,R5

MOV R2,R2,LSR #2

B LOOP1

FINISH:

MOV R8,#1

STR R8,[R7]

B END

END:

SWI 0X011

The screenshot displays a debugger interface with three main panels: RegistersView, CodeView, and WatchView.

**RegistersView:** Shows the state of ARM registers. R0 is 4228, R1 is 4248, R2 is 1, R3 is 2, R4 is 6, R5 is 4, R6 is 6, R7 is 4252, R8 is 0, R9 is 0, R10 (s1) is 0, R11 (fp) is 0, R12 (ip) is 0, R13 (sp) is 70656, R14 (lr) is 0, and R15 (pc) is 4216. 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.

**CodeView:** Displays the assembly code for the program. The code includes labels for .TEXT, LOOP1, FOUND, LESSTHAN, GREATERTHAN, FINISH, and END, with corresponding ARM instructions. The address 0x600000df is highlighted.

**WatchView:** Shows a single watchpoint for the variable 'result', which has a value of 0.

## Not Found Condition:

RegistersView

General Purpose Floating Point

Hexadecimal  
Unsigned Decimal  
Signed Decimal

R0 : 4228

R1 : 4248

R2 : 1

R3 : 1

R4 : 5

R5 : 4

R6 : 6

R7 : 4252

R8 : 1

R9 : 0

R10 (s1) : 0

R11 (fp) : 0

R12 (ip) : 0

R13 (sp) : 70656

R14 (lr) : 0

R15 (pc) : 4212

-----

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

CodeView

p4\_B.o

0000109C:00000000 RESULT: .WORD 0

.TEXT

00001000:E59F0070 LDR R0,=ARRAY

00001004:E59F1070 LDR R1,=SEARCH

00001008:E59F7070 LDR R7,=RESULT

0000100C:E3A02001 MOV R2,#1

00001010:E3A03004 MOV R3,#4

00001014:E5914000 LDR R4,[R1]

LOOP1:

00001018:E1520003 CMP R2,R3

0000101C:0A000011 BEQ FINISH

00001020:E0825003 ADD R5,R2,R3

00001024:E1A050A5 MOV R5,R5,LSR #1

00001028:E1A05105 MOV R5,R5,LSL #2

0000102C:E7906005 LDR R6,[R0,R5]

00001030:E1540006 CMP R4,R6

00001034:0A000002 BEQ FOUND

00001038:BA000004 BLT LESSTHAN

0000103C:CA000006 BGT GREATERTHAN

00001040:EAffFFFF4 B LOOP1

FOUND:

00001044:E3A08000 MOV R8,#0

00001048:E5878000 STR R8,[R7]

0000104C:EA000008 B END

LESSTHAN:

00001050:E1A03005 MOV R3,R5

00001054:E1A03123 MOV R3,R3,LSR #2

00001058:EAffFFFEE B LOOP1

GREATERTHAN:

0000105C:E1A02005 MOV R2,R5

00001060:E1A02122 MOV R2,R2,LSR #2

00001064:EAffFFFEB B LOOP1

FINISH:

00001068:E3A08001 MOV R8,#1

0000106C:E5878000 STR R8,[R7]

00001070:EAffFFFFF B END

END:

00001074:EF000011 SWI 0X011...

:00000000

:00000014

:00000018

OutputView WatchView

Label Value

result 1