

MICROPROCESSORS AND COMPUTER ARCHITECTURE LAB

WEEK 1

PES1UG20CS224

Kumar Abhimanyu

1. Write an ALP using ARM instruction set to add and subtract two 32 bit numbers.
Both numbers are in registers.

Code :

.text

 ;Addition

 MOV r0, #50

 MOV r1, #25

 ADD r2, r0, r1

 ;Subtraction

 MOV r0, #50

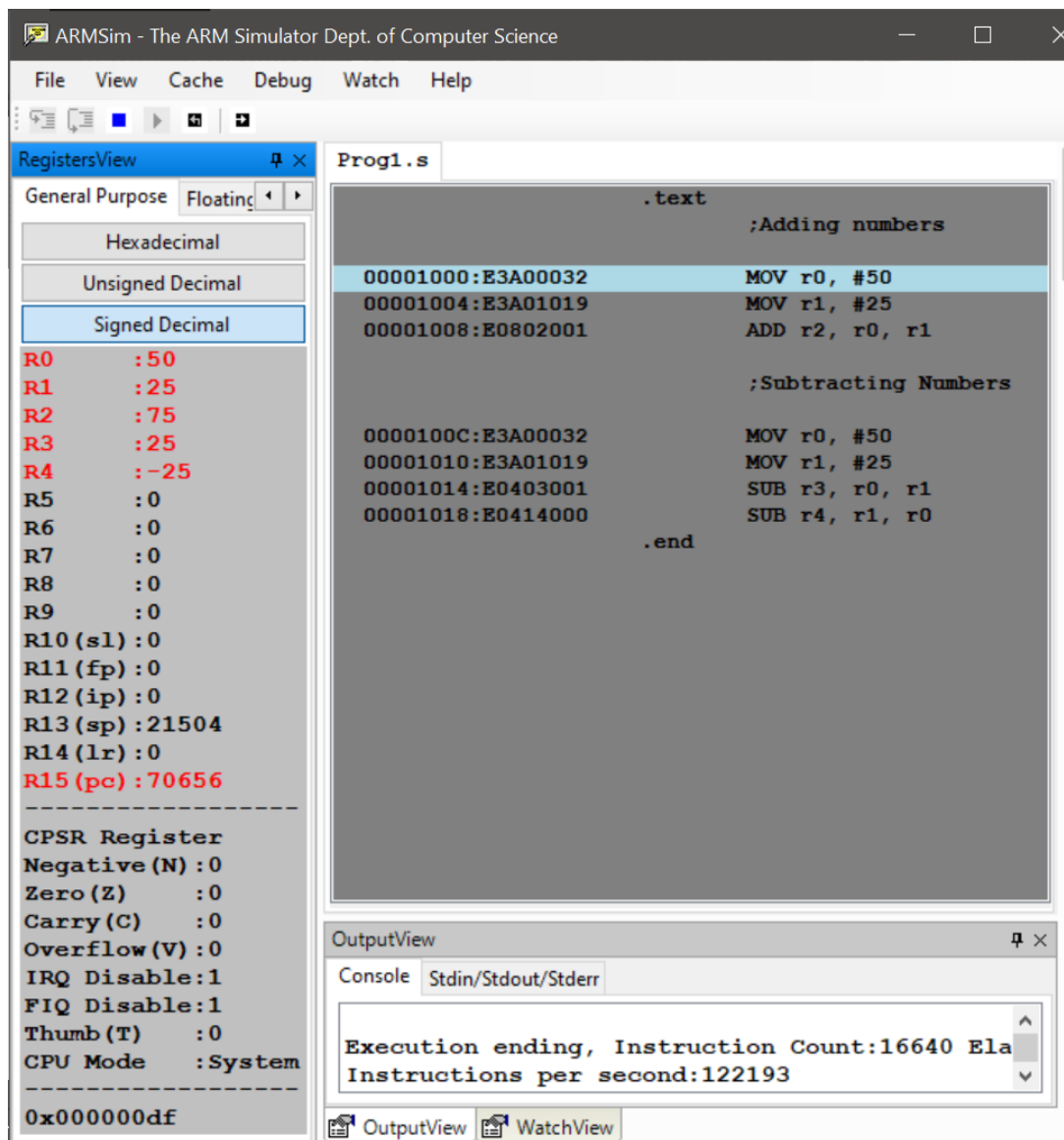
 MOV r1, #25

 SUB r3, r0, r1

 SUB r4, r1, r0

.end

Output :



- Write an ALP to demonstrate logical operations. All operands are in registers.

Code:

```
.text
```

```
    ;AND
```

```
    MOV r0, #1
```

```
    MOV r1, #1
```

```
    AND r2, r0, r1
```

```
    ;OR
```

```
    MOV r0, #1
```

```
    MOV r1, #0
```

```
    ORR r3, r0, r1
```

;NOT

MOV r0, #1

MOV r1, #1

MVN r4, r1

;XOR

MOV r0, #1

MOV r1, #0

EOR r5, r0, r1

.end

Output :

The screenshot displays the ARMSim - The ARM Simulator interface. The main window shows the assembly code for 'Prog2.s' in the 'RegistersView' tab. The code is as follows:

```
.text
;AND
00001000:E3A00001 MOV r0, #1
00001004:E3A01001 MOV r1, #1
00001008:E0002001 AND r2, r0, r1

;OR
0000100C:E3A00001 MOV r0, #1
00001010:E3A01000 MOV r1, #0
00001014:E1803001 ORR r3, r0, r1

;NOT
00001018:E3A00001 MOV r0, #1
0000101C:E3A01001 MOV r1, #1
00001020:E1E04001 MVN r4, r1

;XOR
00001024:E3A00001 MOV r0, #1
00001028:E3A01000 MOV r1, #0
0000102C:E0205001 EOR r5, r0, r1

.end
```

The left sidebar shows the 'RegistersView' with the following values:

Register	Value
R0	:1
R1	:0
R2	:1
R3	:1
R4	:-2
R5	:1
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

The bottom section shows the 'OutputView' with the following text:

```
Execution ending, Instruction Count:16640 Elapsed
Instructions per second:132171
```

3. Write an ALP to add 5 numbers where values are present in registers

Code :

```
.text
;Loading

MOV r0, #5
MOV r1, #5
MOV r2, #20
MOV r3, #30
MOV r4, #40

;Adding

ADD r5, r0, r1
ADD r6, r2, r3
ADD r7, r4, r5
ADD r8, r6, r7
.end
```

Output :

The screenshot displays the ARMSim ARM Simulator interface. The main window is titled "ARMSim - The ARM Simulator Dept. of Computer Science". It features a menu bar (File, View, Cache, Debug, Watch, Help) and a toolbar. The interface is divided into several panes:

- RegistersView**: Shows the state of ARM registers. The "Signed Decimal" tab is selected. Registers R0 through R15 are listed with their values. R15 (PC) is highlighted in red and shows the value 70656.
- Prog3.s**: Displays the assembly code being executed. The code includes comments for loading and adding numbers, and ends with ".end".
- OutputView**: Shows the execution output. The "Console" tab is selected, displaying the message: "Execution ending, Instruction Count:16640 Elapsed Instructions per second:136416".

At the bottom of the RegistersView pane, the memory address "0x000000df" is shown.

4. Write an ALP using ARM instruction set to check if a number stored in a register is odd or even

Code :

.text

```
MOV r0, #10
ANDs r0, r0, #1
```

BEQ condition

```
MOV r1, #1
```

B exit

condition:

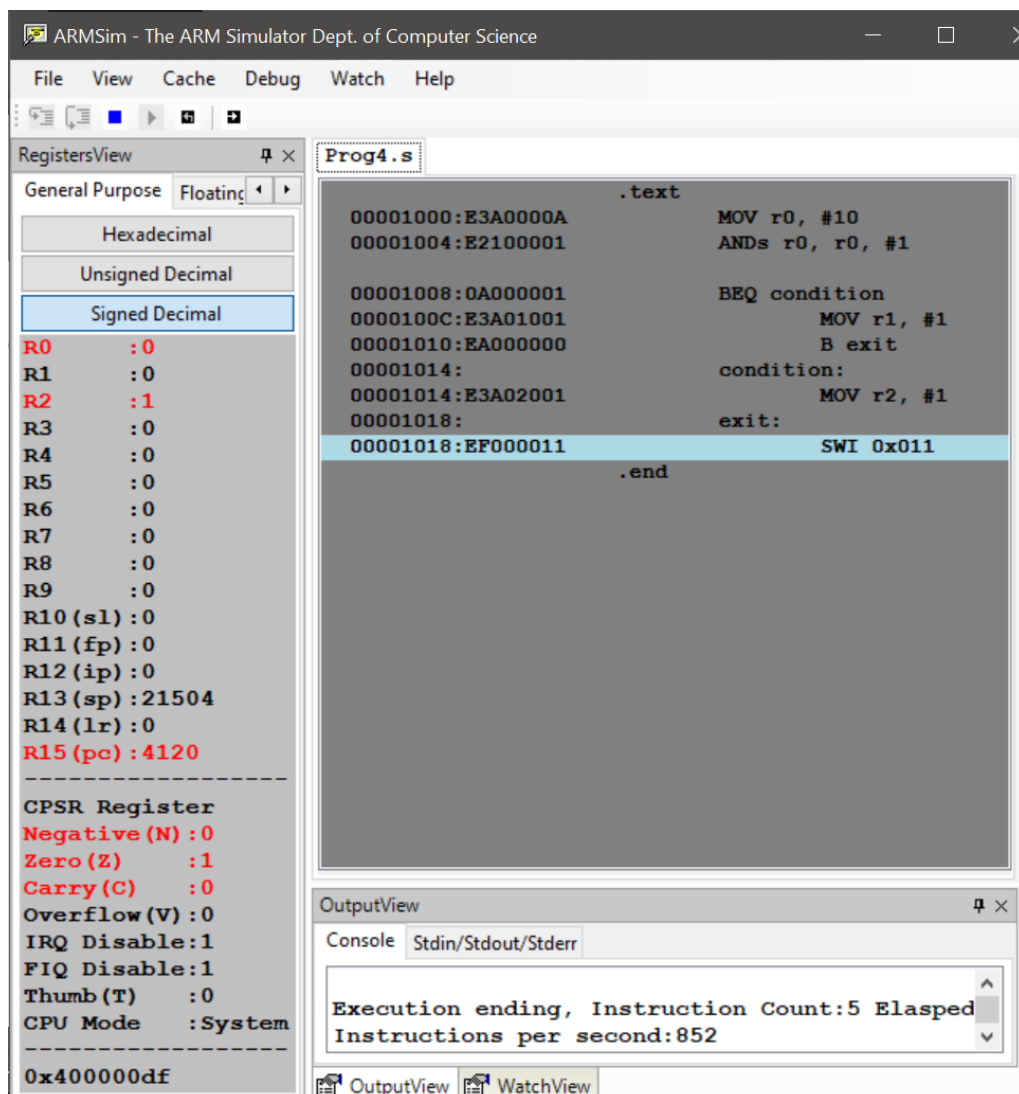
```
MOV r2, #1
```

exit:

```
SWI 0x011
```

.end

Output :



5. Write an ALP to compare the value of R0 and R1, add if R0 = R1, else subtract.

Code :

.text

;Loading numbers

MOV r0, #20

MOV r1, #20

;Setting Zero Flag

SUBS r2, r0, r1

;condition entered if Z=1 -> equal

BEQ condition

SUB r3, r0, r1

B exit

condition:

ADD r3, r0, r1

exit:

SWI 0x011

.end

Output :

