SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Department of Computer Science and Engineering

MC Assignment 1 – Theory Answers

Name: Roshni

USN: 4MW23CS122

Subject Code: BCS402

Semester: IV

Q1.

Question:

If the registers r1, r2 and r3 contain the values 0, 15 and 12 respectively, what will be the value in register r4 after execution of the following code segment? Assume that the registers are 32-bits in size.

MVN r0, r1

AND r4, r0, r2

EOR r4, r4, r3

Answer:

Given:

r1 = 0x00000000

r2 = 0x0000000F

r3 = 0x0000000C

1. MVN r0, r1 → r0 = ~0 = 0xFFFFFFFF

2. AND r4, r0, r2 → r4 = 0xFFFFFFFF & 0x0000000F = 0x0000000F

3. EOR r4, r4, r3 → r4 = 0x0000000F ⊕ 0x0000000C = 0x00000003

Final value of r4: 0x00000003 (decimal 3)

Q8.

Question:

Memory locations 0x40000000, 0x40000004, and 0x40000008 contain the data 10, 20, and 30 respectively. If the register r1 is initialized with the value 0x40000000, what will be the content of register r2 after execution of the following code segment?

LDR r5, [r1]

LDR r6, [r1, #4]

LDR r7, [r1, #8]

ADD r2, r5, r6

SUB r2, r2, r7

Answer:

Initial memory contents:

[0x40000000] = 10

[0x40000004] = 20

[0x40000008] = 30

Execution:

r5 = 10

r6 = 20

r7 = 30

r2 = r5 + r6 = 10 + 20 = 30

r2 = r2 - r7 = 30 - 30 = 0

Final value of r2: 0x00000000 (decimal 0)

Q9.

Question: Memory location 0x40000000 contains the data 0x12341234. The registers r0, r1, and r2 contain the values 0x00000000, 0x11112222, and 0x40000000 respectively. What will be the value in register r0, r1, and r2 after execution of the following instruction?

SWP r0, r1, [r2]

Answer:

Before execution:

Memory [0x40000000] = 0x12341234

r1 = 0x11112222

r2 = 0x40000000

Execution:

SWP exchanges the content of r1 with memory pointed by r2

So, r0 gets old memory value = 0x12341234

Memory at [r2] gets r1's value = 0x11112222

r1 remains unchanged

Final values:

r0 = 0x12341234

r1 = 0x11112222

Memory [0x40000000] = 0x11112222

Q11.

Question:

With the help of bit layout diagram, explain Current Program Status Register (CPSR) of ARM. Also write the layout of CPSR if zero flag is set, the rest NCVQ flags are all clear, the processor is in Thumb state, IRQ interrupts are enabled, and the processor is in SVC mode.

Answer:

CPSR (Current Program Status Register) contains:

Conditional flags (N, Z, C, V, Q)

Control bits (I, F, T)

Processor mode bits

CPSR Bit Layout:

31 30 29 28 27-8 7 6 5 4 3 2 1 0

N Z C V Q... I F T MODE BITS

Given conditions:

Z (Zero flag) = 1

N, C, V, Q = 0

T (Thumb state) = 1

I (IRQ enabled) = 0

Mode = SVC (0b10011 or 0x13)

Final CPSR layout (in binary):

N Z C V Q - - - - - - - - - - - - - - - - - I F T Mode

0 1 0 0 0 0 0 1 10011

CPSR value (in hex): 0x40000013

Q12.

Question:

Construct the ARM core data flow model and explain it with the help of a neat diagram.

Answer:

ARM Core Data Flow Model includes:

Instruction Fetch: PC fetches instructions from memory.

Instruction Decode: Decoding logic determines operation and operands.

Register File: General-purpose registers for input/output of ALU.

ALU: Performs arithmetic and logical operations.

Barrel Shifter: Efficiently performs shift operations.

CPSR: Stores flags and control status.

Data Memory: Accessed via Load/Store instructions.

Diagram Description:

[PC] → [Instruction Fetch] → [Instruction Decode]

↓ ↑

[Register File] → [ALU + Barrel Shifter] → [Result]

↓

[CPSR]

Each instruction goes through a data path of:

Fetch → Decode → Execute → Write Back

Q13.

Question:

What is pipeline in ARM? Illustrate with an example. Show the pipeline stages of ARM9 and ARM10 with example.

Answer:

Pipeline is a technique where multiple instructions are overlapped during execution to improve performance.

Basic 3-stage pipeline in ARM:

Fetch (F)

Decode (D)

Execute (E)

Example (ARM7/ARM9 3-stage):

Cycle 1: Fetch I1

Cycle 2: Decode I1, Fetch I2

Cycle 3: Execute I1, Decode I2, Fetch I3

ARM10 uses a 6-stage pipeline:

Fetch

Issue

Decode

Execute

Memory

Write

This allows higher clock speeds and better instruction throughput.

Q14.

Question:

Write a C program that prints the square of the integers between 0 to 9 using functions and explain how to convert this C function to an assembly function with command.

Answer:

C Program:

```
#include <stdio.h>

int square(int x) {
    return x * x;
}


int main() {
    for (int i = 0; i < 10; i++) {
        printf("%d ", square(i));
    }
    return 0;
}
```

Converting to Assembly: The square function can be converted into ARM assembly:

```
square:
    MUL r0, r0, r0  ; r0 = r0 * r0
    BX lr           ; return
```

*****