PES University, Bengaluru

(Established under Karnataka Act No. 16 of 2013)

UE19CS252

March 2021: IN SEMESTER ASSESSMENT B Tech 4th SEMESTER

ISA – 1

UE19CS252 - Microprocessor and Computer Architecture

Time: 2 Hr	Answer All Questions	Max Marks: 60

Q1

a. Write the equivalent ARM code snippet for the following C–language statement. Use conditional instructions.

```
IF ( R0 == R1) || (R1 ==  R2)  R5=R1-R2;
ELSE  R5 = R1+R2;
```

3 M
3M
4 M
10M

Solution:

```
.text
          MOV R0, #5
          MOV R1, #10
     MOV R2, #1
          CMP R0, R1
          BEQ L1
       CMP R1, R2
       BEQ L1
          ADD R5, R1, R2
          B L2
          L1: SUB R5, R1,R2
          L2: SWI 0x011
.end
```

b. You are expected to perform Y * 253. Let register R1 holds the initial value of Y and to hold the final result of the operation (Multiplication). Write ARM assembly program to perform the operation, without using the MUL or MLA instruction. Try to use as few instructions as possible.

One Possible Solution:

.text MOV R1,#4 MOV R2,#3 RSB R1,R2,R1,LSL #8 .end	Solution 2: MOV R1,#4 MOV R0,R1,LSL #8 (R0← R1*256) ADD R2,R1,R1,LSL #1 (R2←R1*3) SUB R3,R0,R2
.text MOV r1, #4 MOV R2,R1,LSL #8 SUB R1,R2,#3 .end	.text MOV R1, #4 MOV R2,#0 LOOP: ADD R2,R2,#253 SUB R1,R1,#1 CMP R1,#0 BNE LOOP MOV R1,R2 .end

c. Let register R0=0XFFFFFFFF and initial value of R13 for all the instruction be 0x00008010. Show the content of stack and update on stack pointer when following instructions are executed.

iv. LDR R0,[SP],#4i.STR R0,[SP,#-8]----→**0XFFFFFFFF is stored in the address 0x008008**

ii. STR R0,[SP,#-8]! → **0XFFFFFFFF is stored in the address 0x008008 and SP changes to 0x00008008**

iii. ADD SP,SP,#4→ **R13 will have 0x008014**

iv. LDR R0,[SP],#4 → **R0 get the value in the address 0x0088010 and SP will point to 0x00008014**

Q2

a. If register R1 contains the value 0x5C and register R2 contains the value 0x6A, what are the results of the following ARM instructions:

i. EOR R0, R1, R2, ii. AND R0, R1, R2

Solution: (i) R0 = 0x36 (ii) R0 = 0x48

Consider a “Full Descending (FD)” stack organization. How is the growth of stack observed? Give equivalent instruction for PUSH and POP operations on a FD stack.

Solution:

Stack pointer points to topmost element in the stack.

Stack grow towards lower address of the memory

LDMFD is PoP and STMFD is PUSH.

b. Encode the instruction SUB R12,R14,#15.

c. Encode the instruction : LDMIA R2! , {R3,R5,R6,R7}

Solution c: 1110 00 1 0010 0 1110 1100 0000 0000 1111 (E24EC00F)

Solution d: 1110 100 **0**1**011**0010 00000 00**0** **111** 01000 (E8B200E8)

2 M
2 M
3 M
3 M
10 M

Q3

b. Show how the instructions below are executed on a 5 stage (IF, ID, EXE, MEM, WB) pipeline processor.

- Branch is Predicted as ***Not Taken***.
- The result of BEQ is ***Taken*** i.e R1== R2.
- Data Forwarding MEM to MEM, EXE to EXE, EXE to MEM is enabled.
- PC update and Comparison happen in EXE stage.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Beq R1, R2,X	IF	ID	E											
LDR R10,[R11]		IF	ID											
SUB R14,R10,R10			IF											
X:ADD R4,R1,R2				IF	ID	EXE	MEM	WB						
LDR R1,[R4]					IF	ID	EXE	MEM	WB					
SUB R1,R1,R1						IF	ID	STALL	EXE	MEM	WB			
ADD R1,R1,R1								IF	ID	EXE	MEM	WB		

a. Consider a non-pipelined processor using the 5-stage datapath with clock period 0.5 ns. Assume that due to clock skew and pipeline registers, pipelining the processor lengthens the clock cycle period by 10%. Also, assume that the processor uses a unified single-ported cache for data and instruction accesses, resulting in a structural hazard between IF and MEM stages. Suppose that data references represent 30% of the instructions executed and that the ideal CPI of the pipelined

5 M
5 M
10 M

processor, ignoring the structural hazard is 1. How much speedup can we gain from pipelining? Assume a balanced pipeline and ignore the pipeline fill and drain overheads.

Solution:
Without pipelining: Cock period = 0.5 ns
CPI = 5
With pipelining: Clock period = 0.5 + (10% * 0.5) = 0.55 ns
30% of instructions access memory during the MEM stage. Each of these instructions results in a structural hazard with a stall penalty of 1 cycle.
Therefore: CPI = Ideal CPI + Stall CPI = 1 + (0.3)(1) = 1.3
Speedup from pipelining = (0.5 ns * 5) / (0.55 ns * 1.3) = 3.5

Q4

a. It is believed that Speedup in a pipeline processor is equal to K in an ideal execution. Give two reasons why we do not design 100 stage or 1000 stage pipeline processor.
b. Choosing efficient data structure for Branch History Table is inevitable to predict the outcome of branch in 1st stage of pipeline processor. Explain

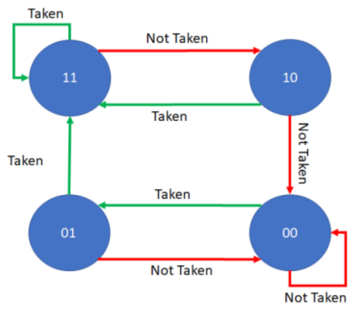
2 M
2 M
3 M
3 M
10 M

Solution a: CPI increases due to

i. Pipeline register overhead.
ii. Miss Prediction during the execution of branch instructions require flushing.
iii. Latency Increase as delay for each stage is fixed to be the delay of the slowest stage.

Solution b: Efficient search on BHT will enable prediction in IF stage which lead to no stall, and eliminate the requirement of filling delay slot.

c. Consider the 2 bit branch predictor given below



A Snapshot of the taken or not taken behavior of the branch is as given.
T, T, T, NT,T,NT,NT,NT,NT,T,T,T,T,T,NT
Suggest and Justify the starting state which will lead to less miss prediction.

Solution: **With ST: 5 Miss Prediction (Suggested)**
WT: 6 Miss Prediction
SNT: 8 Miss Prediction
WNT: 7 Miss Prediction

d. List and explain 3 possible solution which you can depend on compiler to overcome CPU stalls due to hazards.
Solutions:
Reordering the Instructions to avoid dependency.
Inserting NOOP to avoid hardware stalls.
Identifying instructions for delay slot.
Static Branch Prediction.

Q5

a.
i. Do you agree that *Memory is a bottleneck for performance?* Give 3 comments to justify the same. What was the solution proposed by designers?
Solution: Memory Hierarchy (Width of the Bus, Access time, Miss Rate, Cost).
ii. Justify block transfer instead of word transfer between cache and Memory. Will Block transfer affect the performance?
Solution: Principle of Locality (Temporal and Spatial Locality). If bus width is not sufficient to carry entire block, CPU may need to wait.

1+1 M
5 M
3 M
10

b. Consider a system which uses 8 bits address. The cache is organized in a direct mapped manner. Each block hold 4 word (Assume word is 1 Byte). The cache has 16 Lines.

i. Identify number of bits for TAG, Lines/ Cache Blocks and Word.

ii. Compute the miss rate if the data in the following address is accessed in the given sequence. 106,76,107,171,106,79,107,106,170,76,107

Solution:

Tag= 2 bits, Line = 4 bits and word = 2 bits

106= 01 1010 10
107=01 1010 11
170= 10 1010 10
171= 10 1010 11
106,107,170,171 will be mapped to same Cache Block.
76= 01 0011 00
79= 01 0011 11
76,79 will be mapped to same Line.
Access: 106,76,107,171,106,79,107,106,170,76,107

106 is Miss as it is 1st access. Placed in 10th Line.
76 is Miss as it is 1st access. Placed in 3rd Line.
107 is Hit as it is already in the10th Line.
171 is Miss, Placed in 10th Line replacing 106.
106 is Miss, Placed in 10th Line replacing 171.
79 is Hit.
107 is Hit as it is already in the 10th Line.
106 is Hit
170 is Miss, Placed in 10th Line replacing 106.
76 is Hit
107 is Miss, placed in 10th Line replacing 171
Miss rate =6 Miss/ 11

c. Help me in choosing the best cache design between Version1 and Version2.

Version 1: Split Cache (D-Cache & I-Cache) with 35% data access. The average miss rate in the L1 instruction cache was 2%. The average miss rate in the L1 data cache was 10% , the miss penalty is 9 CCs.

Version 2: Unified Cache (Common for Data and Instruction access), with average miss rate 3% for the entire cache and the miss penalty is again 9 CCs.

Which design is better and by how much?

Solution:

Memory Stalls (V1) = 1 x 0.02 x9 + 0.35 x 0.1 x 9 = 0.495
Memory Stalls (V2) = 0.03 x 9= 0.27
Suggestion: Version 2 is the right choice

Q6 Consider the program fragment to copy a zero terminated character string (1 Byte Character) from one location to another. Assume random replacement, write through and write allocate policy is used. The cache is a split cache. The Line size =8 bytes. (1+2+1 +1) 5 M 10 M

```
Strcopy: LDRB R3,[R4]
         STRB R3,[R5]
         BEQ R3,#0, Exit
         ADD R4,R4,#1
         ADD R5,R5,#1
Exit:
```

i. How many memory accesses occurs per iteration? **5 Instruction + 2 Data Access**

ii. If the length of string is 6, discuss miss rate and hit rate on D-Cache while copying string from one location to another? **Note: Termination character is inclusive in length of the string.** **1 Miss and 5 Hit.**

iii. I suggest you to prefer, Write No Allocate instead of Write Allocate Policy. What motivates you to accept my suggestion?

Since each location is accessed only once, it is better to apply Write no allocate and write directly in the memory.

iv. I suggest you to prefer, Write Back instead of Write Through. What motivates you to accept my suggestion?

Saving extra time to write in RAM.

b. The cache is organized in a 2-way set associative manner. Each block hold 1 word (Assume word is 1 Byte). The cache has 16 Lines.

i. Give the list of data which will remain in cache after accessing the last element if the data are accessed in the sequence given below and LRU replacement policy is followed.

76,107,171,106,140,79,76,171,76, 143,107,212,147

ii. What is the Hit Rate and Miss Rate.

iii. Compute the Memory stall considering only the data access and the total data access is 13% of the total instruction executed and Miss Penalty is 25.

Solution:

	Set 0	Hit Rate= 5/13 Miss Rate= 9/13	Memory Stall= Memory Access x Miss rate x Miss Penalty Memory Stall = 0.13 x (9/13) x 25 = 2.25
	Set 1		
106	Set 2		
107	Set 3		
~~171,~~147			
76	Set 4		
~~140, 212~~			
	Set 5		
	Set 6		
79	Set 7		
143			