SRN | | | | | | | | | | | | |

PES UNIVERSITY	**PES University, Bengaluru** (Established under Karnataka Act No. 16 of 2013)	**UE20CS252**

MAY 2022: END SEMESTER ASSESSMENT (ESA) B TECH 4th SEMESTER

UE20CS252 – MICROPROCESSOR AND COMPUTER ARCHITECTURE

Time: 3 Hrs	**Answer All Questions**	**Max Marks: 100**

Note: ARM7TDMI – ISA stands for ARM Instruction Set Architecture,
ALP – Assembly Language Program

1 a) **Convert the following pseudo code into an ALP using ARM7TDMI – ISA.** **06**

IF([A]==[B]) then C=[A]*[B];
ELSE IF ([B]==[C]) D=[A]-[B];
ELSE E=[A]+[B].

[Note: Use Conditional Execution Instructions].

Assume that the registers R1, R2, R3 contain the addresses of memory locations A, B and C respectively.
Also, Let registers R6, R7, to carry the contents of memory locations A and B respectively.

Answer:

```
.DATA
    A: .WORD 10
    B:  .WORD 20
    C:  .WORD 0
    D:  .WORD 0
    E:  .WORD 0
    ; initialization 1 mark
.TEXT
    LDR R1,=A
    LDR R2,=B
    LDR R3,=C
    LDR R4,=D
    LDR R5,=E

    LDR R6,[R1]
    LDR R7,[R2]
    LDR R11,[R3]
  ; first IF  - 2marks
    CMP R6,R7
    MULEQ R8,R6,R7
    STREQ R8,[R3]
    ; second IF – 2 marks
    CMPNE R7,R8
    SUBEQ R9,R6,R7
    ; ELSE  - 1 mark
    ADDNE R10,R6,R7

    SWI 0X011
.END
```

SRN | | | | | | | | | | | | |

	b)	**The code snippet given below performs matrix multiplication. Complete the code by writing the missing instructions.** **Answer:**	06

```
.DATA
A: .WORD 1,2,3,4,5,6,7,8,9
B: .WORD 1,2,3,4,5,6,7,8,9
C: .WORD 0,0,0,0,0,0,0,0,0
.TEXT
        LDR R0,=A
        LDR R1,=B
        LDR R2,=C
        MOV R3,#0
        MOV R4,#0
          MOV R10,#3
        MOV R8,#0

LOOP1:MLA R11,R3,R10,R8      ; 2 marks
        MOV R11,R11,LSL #2
        LDR R5,[R0,R11]

        MLA R12,R8,R10,R4    ; 2 marks
        MOV R12,R12,LSL #2
        LDR R6,[R1,R12]

        MUL R11,R5,R6
        ADD R9,R9,R11
        ADD R8,R8,#1
        CMP R8,#3
        BNE LOOP1
        MLA R12,R3,R10,R4      ; 2 marks
        MOV R12,R12,LSL #2
        STR R9,[R2,R12]

        MOV R8,#0
        MOV R9,#0
        ADD R4,R4,#1
        CMP R4,#3
          BNE LOOP1
        MOV R4,#0
        ADD R3,R3,#1
        CMP R3,#3
        BNE LOOP1
        SWI 0X011
.END

// Students might have used other //registers. Please check the logic //accordingly.
```

SRN | | | | | | | | | | | | |

c) **How are the parameters passed to the functions in ARM7TDMI-ISA using stack? Explain with an example.** **04**

Answer:

Step1: Parameters are stored on the stack.
Step2: Store link (return address)
Step3: Call function
Step4: First instruction will be to retrieve parameters from the stack.

Example:

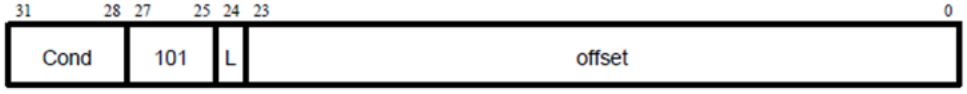
```
        LDR   R6, =A
       MOV  R1, #25                ; parameter1
       MOV  R2, #25                ; parameter2
       STMFD R13!, { R1, R2}       ; parameters are pushed on stack.      ; 2 marks
            BL  LINK
            STR R0, [R6]                  ; return value in Reg. R0.
       SWI 0x11
LINK:  LDMFD R13!, { R4, R5}       ; parameters are poped from the  stack   ;2 marks
        ADD R0, R4, R5             ;  Result is in register R0.
        MOV PC, LR
  A:   .WORD  0
.END
```

d) **Consider the following code snippet. Assume ARM7TDMI - 3 stage pipeline processor. The execution of the branch instruction happens in the execute stage.** **04**

```
.TEXT                          LOOP: SUB R1, R1, R0
LDR R3,=A                            B GCD
LDR R4,=B                      RES: MOV R2,R0
LDR R0,[R3]                          SWI 0X011
LDR R1,[R4]                    .END
GCD: CMP R0, R1
     BEQ RES
     BLT LOOP
     SUB R0,R0,R1
     B GCD
```

Determine the hexadecimal equivalent instruction encoding for the branch instruction BEQ and B instructions.

Note: The encoding pattern for a branch instruction is as given below with the condition code for EQ,LT and AL being 0000, 1011 and 1110 respectively.



i. BEQ RES instruction: ; 2 marks

(0000 101 0 0000 0000 0000 0000 0001 0000)$_2$

(0 A 0 0 0 0 1 0)$_{16}$

ii. B GCD : ; 2 marks

(1110 101 0 1111 1111 1111 1111 1111 1000)$_2$

(E A F F F F F 8)$_{16}$

SRN | | | | | | | | | | | |

2	a)	Consider five stage pipeline architecture of a processor. Assume that the individual stages of the data path have the following latencies. IF – 300ps, ID – 400ps, EX – 350ps, MEM – 500ps and WB – 100ps. What is the clock cycle time in a pipelined and non-pipelined processor? If there are 50 instructions compute the ratio of non pipelined to pipelined processors. **Answer:** **Non-pipelined Processor latency : ; 2 marks** **Time taken** = (IF + ID + EX + MEM + WB) = 300+400+350+500+100 = **1650ps.** **For 50 instructions: Kstage* Tc* n Instructions** = 5 x 500 x 50 **= 125000 ps** **Pipelined Processor Latency: ; 2 marks** Time taken = MAX(300,400,350,500,100) x No. of stages **= 500 x 5 = 2500ps.** **For 50 instructions : Kstage * 1 instruction *Tc + (n-1)* Tc** **=** 5 x 1 x 500 + (49) 500 = **27000 ps.** **Ratio of Non pipelined to pipelined processors = 125000 / 27000 = 4.63 times.**	04

SRN | | | | | | | | | | | | |

b) Examine how data dependencies affect execution in the basic 5-stage pipeline processor. The execution of the branch instruction happens in the execute stage. **06**

LW $1, 40($6)
ADD $6, $2, $2
SW $6, 50($1)

i. Assuming there is no forwarding in this pipelined processor,
Indicate the hazards and add NOP instructions to eliminate the same.

ii. Assuming there is full forwarding,
Indicate the hazards and add NOP instructions to eliminate the same.

Show the same using pipeline execution diagrams.

Answer: **4 Marks**

i. case 1:

LW $1, 40($6) IF ID EX MEM WB

ADD $6, $2, $2 IF ID EX MEM WB

SW $6, 50($1) IF ID EX MEM WB

Using NOPs the instruction sequence is as follows:

LW $1, 40($6)
ADD $6, $2, $2
NOP
NOP
SW $6, 50($1)

ii. case2 (Forwarding results in no bubble states). **2 Marks**

LW $1, 40($6) IF ID EX MEM WB

ADD $6, $2, $2 IF ID EX MEM WB

SW $6, 50($1) IF ID EX MEM WB

c) Consider the following sequence of instructions. **06**

```
LABEL1:  LW    $1, 40 ($6)
         BEQ   $2, $3, LABEL2  ; branch taken
         ADD   $1, $6, $4
LABEL2:  BEQ   $1, $2, LABEL1  ; branch not taken
         SW    $2, 20 ($4)
         AND   $1, $1, $4
```

i. Draw the pipeline execution diagram for the code assuming there are no delay slots and the branches execute in the EX stage.

ii. Draw the pipeline execution diagram for the code assuming delay slots are used and branch instructions are executed in the ID stage.

NOTE: $1,$2, $3, indicates register1, register2, register3 respectively and so on.

Answer:

Case i. **2 marks**

```
LABEL1:  LW    $1, 40 ($6)              IF  ID  EX  MEM   WB
         BEQ   $2, $3, LABEL2  ; BT        IF  ID  EX   MEM   WB
         ADD   $1, $6, $4                      IF  ID   FLUSHES…….
        ( dependency is seen)
LABEL2:  BEQ   $1, $2, LABEL1  ; BNT                  IF  ID  EX  MEM  WB
         SW    $2, 20 ($4)                              IF  ID  EX   MEM  WB
         AND   $1, $1, $4                                  IF  ID  EX   MEM  WB
```

Case ii. **2 marks**

```
LABEL1:  LW    $1, 40 ($6)              IF  ID  EX  MEM   WB
         BEQ   $2, $3, LABEL2  ; BT        IF  ID  EX   MEM   WB
         AND   $1, $1, $4                      IF  ID  EX   MEM  WB
         ADD   $1, $6, $4                          IF  ID    EX   MEM  WB
        ( safe instruction is written after branch taken avoids flushing)
LABEL2:  BEQ   $1, $2, LABEL1  ; BNT                  IF  ID  EX  MEM  WB
         SW    $2, 20 ($4)                              IF  ID  EX   MEM  WB
```

Explanation : 2 marks

d) Consider a program with the following behavior. **04**

T, T, T, NT, NT, T, T, T, T, NT, NT, T (where NT –branch Not Taken & T – branch Taken).

How many miss predictions are seen if the initial state is strongly taken for a 2 bit prediction?

Answer:

Given behavior :

T, T, T, NT, NT, T, T, T, T, NT, NT, T

Initial State: Strongly Taken (T).

Outcome:

T T T T T NT NT T T T T NT **2 marks**

No of miss predictions are: 7 **2 marks**

SRN | | | | | | | | | | | | |

3	a)	A computer system with a word length of 32 bits has a 64MB byte-addressable main memory and a 128KB; 4-way set associative cache memory with a block size of 2048 bytes. i. Determine the block number of the address $(546888)_{10}$ and find the set number in the cache memory and the tag bits. ii. Compute the total number of bits required for a direct mapped cache? **Answer:** **i. Word Length = 32 bits = 4 bytes.** **Block Size = 2048 bytes = 512 words.** **Main Memory Capacity = 64MB requires – 26bits as address bits** **Cache size = 128KB = 32K words.** **Total number of blocks in cache = 32 x 2^10 / 512 = 64 blocks** **Number of sets in Cache = 64/4 = 16 - 1 mark.** i. Block Number in main memory = **546888 / 2048 = 267. - 1 mark.** **SET number in cache memory = 267 % 16 = 11 - 1 mark.** **Number of TAG Bits = 13 - 1 mark.** **←----------26----------→** **TAG SET WORD** **13 4 9** ii. The total number of bits required for a direct mapped cache: **←----------26----------→ - 1 mark** **TAG BLOCK WORD** **13 6 9** Total number of bits required for Cache is (dirty bit + valid bit = 2 x 64 =128bits) **-> 2048 x 8+ 64 x 13 + 128 = 17344 bits - 1 mark**	06
	b)	Assume cache memory of 16KB with a block size of 512 bytes. If the hit time is 1.08ns and miss rate is 3.4%, what is the average memory access time if the system has 90 clock cycles overhead and takes 4CC to transfer every 64 bytes from main memory? **Answer:** Cache Memory Capacity = 16KB. Block size is 512bytes. Hit time = 1.08ns, Miss rate = 3.4%, Miss penalty = Overhead + timetaken for 64bytes transfer x (512 bytes/64) = 90 cc + 4 cc x 8 = **122cc.** **2 marks** AMAT = Hit time + Miss Rate x Miss Penalty. = 1.08ns + 3.4% x 122cc **AMAT= 5.228cc** **1 marks** **Assume 1 cc = 1ns. - 1 mark if attempted**	04

	c)	Consider a scenario of multilevel cache memory system. It is given that the write policies of L1 and L2 cache memory are write-back policy and write through policy respectively. Describe the procedure of handling L1 write miss considering the component involved and the possibility of replacing a dirty block. **Answer:** Step1 . L1 generates a miss that goes to L2 cache. Step2. Since L1 has write back policy, dirty bit is set as per given data. Step3. Assume L2 has the block that needs to be written to L1 cache. But, L1 block dirty bit is set, it has to be copied to the L2 cache, but buffer is not available in L1 cache. Hence it has to write to L2 cache. On writing to L2 cache, it also updates in the main memory because of write through policy of L2 cache. Step4. Once, it is updated, the requested block from L2 cache is copied on L1 cache. **Each step1,2,4 carries 1 mark +step3 carries 2 marks.**	**05**

SRN | | | | | | | | | | | | |

d) Consider a memory system that has 512 byte cache with 64 byte blocks. Assume that the main memory is 2KB large. Main memory has 64 byte blocks: M0, M1, M2, M3,….M31. The figure below sketches the memory blocks that can reside in different cache blocks if the cache was fully associative. **05**

Cache block	Set #	Associativity	Memory Block
0	0	0	M0, M1, M2, ..M31
1	0	1	M0, M1, M2,…M31
….	0	…	…..
7	0	7	M0, M1, M2, ..M31

Show the contents of the table if cache is organized as

i. Direct Mapped Cache **2 marks**

Cache block	Set #	Associativity	Memory Block
0	0	0	M0, M8, M16, M24
1	1	0	M1, M9, M17, M25
2	2	0	M2, M10, M18, M26
3	3	0	M3, M11, M19, M27
4	4	0	M4, M12, M20, M28
5	5	0	M5, M13, M21, M29
6	6	0	M6, M14, M22, M30
7	7	0	M7, M15, M23, M31

ii. 4 –Way Set Associative Cache. **3 marks**

Cache block	Set #	Associativity	Memory Block
0	0	4	M0, M8, M16, M24
1		4	M2, M10, M18, M26
2		4	M4, M12, M20, M28
3		4	M6, M14, M22, M30
4	1	4	M1, M9, M17, M25
5		4	M3, M11, M19, M27
6		4	M5, M13, M21, M29
7		4	M7, M15, M23, M31

SRN | | | | | | | | | | | |

4	a)	Consider the following data What are the observations with respect to cache size and block size based on miss rate? Justify. Discuss with respect to AMAT as well. **Answer:** **1.5 x 4 =6.0 marks** 1. As cache size increases, miss rate reduces. 2. As cache block size increases, miss rate reduces but increases later on due to capacity misses. 3. Computing AMAT: Smaller block size has larger AMAT while larger block size has lesser AMAT. 4. Smaller Cache size with larger block size has larger AMAT and larger block size and cache size has lesser AMAT.	**06**
	b)	Consider a system that has 64 bit logical address and 41 bit physical address. If the page size is 16KB and the TLB with direct mapped cache can store 512 entries, determine the number of tag bits required for i. 16KB - L1 cache with a block size of 128 words. ii. 16MB L2 – cache with a block size of 128 words **Answer:** **2 marks** **i.** **←----------41 bits - Physical Address----------→** **TAG - 27 bits** | **BLOCK - 7 bits** | **WORD - 7 bits** **ii.** **2 marks** **←----------41 bits - Physical Address----------→** **TAG - 17 bits** | **BLOCK - 17 bits** | **WORD - 7 bits**	**04**
	c)	When an interrupt is detected, the status register is saved and all but the highest priority interrupt is disabled. Why low priority interrupts disabled? Why is the status register saved prior to disabling interrupts? **Answer:** **2 marks** Devices connected to the processor capable to generate interrupts can interrupt the processor. To ensure good functionality of the devices and to resolve multiple and simultaneous interrupts are used. To give priority for high priority devices, low priority interrupts are disabled . **2 marks** To resume the environment after the interrupt service is completed, status register is saved.	**04**

SRN | | | | | | | | | | | | |

	d)	Direct memory access allows devices to access memory directly rather than working through the CPU. This improves the performance of the peripherals, but adds to complexity to memory system implementations. i. Does CPU relinquish (giveup) control of the memory when DMA is active? ii. For example, can a peripheral simply communicate with the memory directly, avoiding the CPU completely? iii. How will graphics card benefit from DMA? **Answer:** **3 x 2 marks = 6 marks.** i. No. when memory is busy, if CPU is also contending for memory, it will be delayed during the DMA transfer. ii. No. peripherals cannot communicate with memory directly. CPU has to initiate the process, perform the operations by DMA and DMA need to interrupt CPU after completion of the task. iii. Data required by the graphics card will be done by DMA controller based on the instruction of CPU. Thus instead of graphics processor in the graphics card waiting for the data, DMA controller performs the transfer of data without the intervention of the CPU. However, data transfer will be done under the control of CPU.	**06**
5	**a)**	Consider the operation of finding the dot product of 2 vectors as mentioned below: Let a=[a1, a2, a3,..an] and b=[b1,b2,b3…bn] is defined as $\Sigma a_i b_i$ = a1b1+a2b2+a3b3+…+anbn. Realize the problem to be solved on parallel computing system. Write the loops using pseudo code. The value of n is 100K. You may need 2 levels of computation. **Answer:** a. Compute the product of the corresponding values of ai and bi in parallel. **2 marks** Let's say - > a1b1 = C1 , a2b2 = C2 , a3b3 =C3,……anbn = Cn. b. Compute the sum of the partial products in parallel. **2 marks** C1+C2+C3+…..+Cn.	**04**
	b)	Consider a parallel computing system with 128 processors. i. Compute the speedup if the code has 15% serial processing. ii. Compute the scaled speedup if the code can be 95% parallelized. { Hint: Use Amdahl's and Gustafson's Laws formulae appropriately}. **Speedup = 1 / (1 – F) + (F / N) .** Where F is proportion of parallel computation = 1 / (1-0.85) + (0.85/128) = **6.34**. - **2 marks** If parallelization is increased to 95%, then, **Scaled Speedup = N-(N-1)*S** where N - the number of processors and S- proportion of serial computing. = 128-(128-1)*0.05 = **121.65** **- 2 marks**	**04**

SRN | | | | | | | | | | | |

	c)	Mention the design constraints or issues in a parallel computing system. **Answer:** **- 4 x 1 mark** 1. Decomposition or Partitioning 2. Mapping 3. Communication 4. Synchronization	04
	d)	What are the 3-ways of accelerating applications in GPU system? Actual computations are done on the streaming multiprocessors [SM]. What are the components of each SM? How many warp schedulers would be in GPU that has 1536 cores? "GPU has better throughput than the speed of a Multicore processor" – Justify. **Answer:** i. 3 ways of accelerating: 1. Libraries 2. Programming Languages 3. Directives. **– 3 marks** ii. SM components are: Control units, registers, execution pipelines, caches **- 2 marks** iii. Warp Schedulers : 2 per SM x (1536 / 32) = 96 **- 2 marks** iv. Large number of cores in GPU results in better performance than CPU. **– 1 mark.**	08