

**PES UNIVERSITY**

**Department of Computer Science & Engineering**

**Microprocessor & Computer Architecture Lab**

**UE23CS251B**

**WEEK 4 submission**

|  |  |
| --- | --- |
| **Name of the Student** | **Samyak Sankhla** |
| **SRN** | **PES1UG23CS511** |
| **Section** | **I** |
| **Department** | **CSE** |
| **Campus** | **RR** |

**Department of Computer Science & Engineering**

**Microprocessor & Computer Architecture Lab**

**UE23CS251B**

|  |  |
| --- | --- |
| 1 | Write an ALP using ARM7TDMI to add n numbers bytewise.  . .DATA    A: .byte 1,2,3,4,5,6,7,8,9,10  Program screen shot:  A:.byte 1,2,3,4,5,6,7,8,9,10  LDR R0,=A  MOV R1,#0  MOV R3,#10  LOOP: CMP R3,#0  BEQ DONE  SUB R3,R3,#1  LDRB R2,[R0],#1  ADD R1,R2,R1  B LOOP  DONE: SWI 0X11  Screen shot of Register set output: |
| 2 | Write an ALP using ARM7TDMI to generate a square given matrix with A  If (i==j) then A[i][j]=5  Otherwise A[i][j]=0  (Note:Any size of matrix can be given as input)  Considering 4X4 matrix    Example : 5 0 0 0  0 5 0 0  0 0 5 0  0 0 0 5  Before:  A:.word 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  After:  A:.word 5,0,0,0,0,5,0,0,0,0,5,0,0,0,0,5  Program screen shot:  LDR R0,=A  MOV R1,#16  MOV R3,#0  MOV R5,#5  FIVE: SUB R1,R1,#1  CMP R1,#0  BEQ DONE  STR R5,[R0]  ADD R0,R0,#4  MOV R2,#4  ZERO: SUB R1,R1,#1  SUB R2,R2,#1  STR R3,[R0]  ADD R0,R0,#4  CMP R2, #0  BNE ZERO  B FIVE  CMP R1,#0  BNE FIVE    DONE: SWI 0X11  A:.word 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  Screen shot of Register set output and memory location: |
| 3 | Write an ALP using ARM7TDMI to convert hexadecimal to decimal.  Input: **0x0124**(Hex)  Conversion Process:   * 0x124→ Extract **4**→ 4\* 16^0 + 0 = 4 * 0x124 → Extract **2** → 2 \* 16^1 + 4 = 36 * 0x124 → Extract **1** → 1\* 16^2 + 36 =292   Output: **292(**Decimal)  Program screen shot:  MOV R0,#0x0124  MOV R2,#0  MOV R3,#1  LOOP: CMP R0,#0  B EXIT  AND R4,R0,0xF  MUL R4,R3,R4  ADD R2,R2,R4  MOV R0,R0,LSR #4  MOV R4,#16  MUL R3,R4,R3  B LOOP  EXIT: SWI 0X11  Screen shot of Register set output and memory location: |
| 4 | Write an ALP using ARM7TDMI, for the given matrix arranged in Column major order, find the index of an element if coordinates of a matrix is given and also find the address of the indexed element. (Using MLA instruction)  . DATA    A:.WORD 1,2,3,4,5,6,7,8,9  .Index for the column major= y\*no of rows+x  Program screen shot:  LDR R0, =A  MOV R1, #3    MOV R2, #1  MOV R3, #2  MLA R4, R3, R1, R2  MOV R5, #4  MLA R6, R4, R5, R0  LDR R7, [R6]    SWI 0X11  .DATA  A:.WORD 1,2,3,4,5,6,7,8,9  Screen shot of Register set output and memory location: |
|  | Assignments Questions |
| 5 | Write an ALP using ARM7TDMI to reverse the elements stored in location A with location B  Before:  A:.word 1,2,3,4,5,6,7,8,9,10  After :  A:.word 10,9,8,7,6,5,4,3,2,1  Program screen shot:  LDR R0,=A  LDR R1,=A  ADD R1,R1,#36  MOV R2,#0  LOOP: LDR R5,[R0]  LDR R6,[R1]  STR R6,[R0],#4  STR R5,[R1],#-4  ADD R2,R2,#1  CMP R2,#5  BNE LOOP  SWI 0X11  A: .word 1,2,3,4,5,6,7,8,9,10  Screen shot of Register set output: |
| 6 | Write an ALP using ARM7TDMI to find the largest of all the BCD digits of a given 32bit number.  (hint:If R1=17845374 the largest digit is 8  Program screen shot:  LDR R0, =0x17845374  MOV R1, #0  MOV R2, #8  MOV R3, #0  LOOP:  AND R3, R0, #0xF  CMP R3, R1  MOVHI R1, R3    MOV R0, R0, LSR #4  SUB R2, R2, #1  CMP R2, #0  BNE LOOP  EXIT:  MOV R7, #1  SWI 0X11  Screen shot of Register set output: |

|  |
| --- |
| Extra |
| Write an ALP using ARM7TDMI to copy a block 400 bytes of data from location A to location B if the rate of data transfer rate is 40 bytes, LDM and STM instructions.  and  For the same transfer the block with auto-indexing. |

 .data

A:.WORD 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

B:.WORD  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,