Department: CSED Program: B.Tech (CSE)

Subject: Microprocessor lab (CSN14402) Semester: 4th Session: 2023-2024(Even Semester)

|  |  |  |  |
| --- | --- | --- | --- |
| **SL. No.** | **Existing Experiments** | **Modifications (if any)** | **Remarks (if any)** |
| 1. | ALP for addition of two-8bit numbers without carry   * Where address is stored in registers. * Where address is taken direct from memory. |  |  |
| 2. | ALP for addition of two-8bit numbers with carry. |  |  |
| 3. | ALP Subtraction of two-8bit numbers with borrow. |  |  |
| 4. | ALP for addition of two-16-bit numbers. | 4(a). Add two 16-bit numbers without using memory pointers   * for 8-bit Operation * for 16-bit Operation   4(b). Subtract two 16-bit numberswithout using memory pointers |  |
| 5. | ALP to add 5 numbers stored in consecutive memory location. | ALP to add a series of ten 8-bit numbers stored at memory location  starting from 2000H. Store the lower byte of the result in lower order address and higher byte in the higher order address. |  |
| 6. | ALP to take the input of N numbers from memory location 2201 and store sum of even number at 2210 and sum of odd number at 2211. | 6(a). ALP to separate odd and even numbers from the given list of 10 numbers. Store odd numbers in another list starting from memory location 2100H. Store even numbers in another list starting from memory location 2200H. Starting address of the list is 2000H.  6(b). ALP to find the maximum among a series of 10 numbers stored at memory location starting from 2000H. Store the maximum number after the last stored number. |  |
| 7. | ALP to transfer of data from one block of memory location to any other block of memory location (non-Overlapping) |  |  |
| 8. | ALP to transfer of data from one block of memory location to any other block of memory location (Overlapping) |  |  |
| 9. | ALP to perform division of two-8-bit numbers | ALP to multiply two 8-bit numbers stored at memory locations 2000H and 2001H respectively   * Using DAD instruction * Without Using DAD   Store the multiplied result at the subsequent memory locations. Also check the cases whether any of the number is zero or not. If so, store 0000H at the respective locations. |  |
| 10. | ALP to check whether the given number is prime or not. |  |  |
| 11. | ALP to print the Fibonacci series between 1 to 100. |  |  |
| 12. | ALP to find the Factorial of the given number. |  |  |
| 13. | ALP to find the given number using Linear Search. |  |  |
| 14. | ALP to find the given number using Binary Search. |  |  |
| 15. | ALP to arrange the given set of numbers using Bubble Sort. |  |  |
| 16. | ALP to arrange the given set of numbers using Selection Sort. |  |  |
| 17. | ALP to arrange the given set of numbers using Insertion Sort. |  |  |
| 18. | Write an assembly language program to check whether an 8-bit number stored at a specific memory location is positive or negative. Store the result accordingly. |  |  |
| 19. | Write an assembly language program to separate odd and even numbers from a list of 10 numbers, storing them in different lists starting from specified memory locations. |  |  |
| 20. | Add two hexadecimal numbers in a high-level language and then convert that program into assembly language. |  | Additional |
| 21. | Write an assembly language program to complement the lower nibble of an 8-bit number stored at a specific memory location without affecting the higher nibble |  | Additional |
| 22 | Write an assembly language program to check whether an 8-bit number stored at a specific memory location is positive or negative. Store the result accordingly. Also find the sum of positive and negative numbers separately and frequency of those numbers. |  | Additional |
| 23. | Write an assembly language program to separate odd and even numbers from a list of 10 numbers, storing them in different lists starting from specified memory locations. |  | Additional |