

# Parallel Processing

## Assignment 3

**This assignment is individual assignment, every student should submit by himself.**

**Due: Next Section**

### 1. True or False:

- There are two criteria to measure a parallel program efficiency, Steps and Total work.
- If Total work of a parallel program equals the Total work of a sequential program for the same problem then we have a Work Efficient program.
- Reduce operator must be unary operator.
- Serial Reduce program has a work complexity =  $O(n \log n)$ .
- Parallel Reduce program has a work complexity =  $O(\log n)$ .
- Reduce pattern requires an Identity element.
- The division operation is a reduction operator.

**2. The following program to add two matrices. Compute the number of steps and the total work complexity for the for addition operation of this program.**

```
#include <stdio.h>
int main()
{
    int m, n, c, d, first[10][10], second[10][10], sum[10][10];

    printf("Enter the number of rows and columns of matrix\n");
    scanf("%d%d", &m, &n);
    printf("Enter the elements of first matrix\n");

    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            scanf("%d", &first[c][d]);

    printf("Enter the elements of second matrix\n");

    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            scanf("%d", &second[c][d]);

    printf("Sum of entered matrices:-\n");

    for (c = 0; c < m; c++) {
        for (d = 0; d < n; d++) {
            sum[c][d] = first[c][d] + second[c][d];
            printf("%d\t", sum[c][d]);
        }
        printf("\n");
    }

    return 0;
}
```

**3. Write a CUDA program to solve the matrices addition in a parallel fashion. Then compute the number of steps and the total work complexity for addition operation of this program.**

**4. Compare the complexity in questions 3 and 4.**