DESIGN REPORT

The basic idea behind the program is to divide the final computation of prefix sum among the various processors. For example, let's say we have 80 numbers. So, we have to compute the prefix sum 80 times, i.e. we'll have S_0 to S_{79}.

Now, consider having 8 processors, from P_0 to P_7. We divide and send the data to each processor as follows.

- P0 will have all the data. But, it will compute S0 to S9.
- P1 will have the data from X0 to X19. It will compute from S10 to S19.
- P2 will have the data from X0 to X29. It will compute from S20 to S29.

Similarly, it continues like that and finally,

- P7 will have the data from X0 to X79. It will compute from S70 to S79.

Instead of calculating the PrefixSum directly on the array variable "Sum", I have used a temporary variable subSum, which at each iteration, is used to calculate the PrefixSum up to the previous level and then this subSum is added with the corresponding value of X to get the PrefixSum at that iteration or level.

This will yield the correct solution because I haven't changed the basic logic behind the PrefixSum computation. So, as said above, each processor will have access to different amount of input data X, but will compute equal amount of PrefixSum values in parallel time. This will reduce the overall execution time of the program.

NOTE :

1. This program works best when the number of elements or input variables is a multiple of the number of processors.
2. Another important limitation is that the number of input variables should be greater than the number of processors.
3. I have used only integer variables as input to calculate the PrefixSum. However, the program can be modified to accommodate other data types too.