Algorithm 1: Proposed parallel prefix sum algorithm.

Input: An array $\vec{X} = (x_0, x_1, \ldots, x_{n-1})$ of n numbers; number p of processors.
Output: Partial sums $x_0, \Sigma_{x_0}^{x_1}, \Sigma_{x_0}^{x_2}, \ldots, \Sigma_{x_0}^{x_{n-1}}$ of all the numbers in $\vec{X}$.

begin
  Divide $\vec{X}$ into p equal-lengthed segments, $\vec{S}_0, \vec{S}_1, \ldots, \vec{S}_{p-1}$. Assume the number of elements in $\vec{S}_i$ is m, and $m = \lfloor n/p \rfloor$ if $i \neq p-1$, or $m = n - \lfloor n(n-1)/p \rfloor$ if $i = p-1$.
  Assign $\vec{S}_i$ to processor p_i, $i \in [0, p-1]$.
  forall $\vec{S}_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,m-1})$ **in parallel do**
    for $j \rightarrow 1$ **to** $m-1$ **do**
      $x_{i,j} = x_{i,j} + x_{i,j-1}$
  for $i \rightarrow 1$ **to** $p-1$ **do**
    Divide numbers in $\vec{S}_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,m-1})$ into p equal-lengthed segments, $\vec{S}_{i,0}, \vec{S}_{i,1}, \ldots, \vec{S}_{i,p-1}$. Assume the number of elements in $\vec{S}_{i,j}$ is w, $j \in [0, p-1]$.
    Assign $\vec{S}_{i,j}$ to processor p_j, $j \in [0, p-1]$.
    forall $\vec{S}_{i,j} = (x_{i,j,0}, x_{i,j,1}, \ldots, x_{i,j,w-1})$ **in parallel do**
      for $k \rightarrow 0$ **to** $w-1$ **do**
        $x_{i,j,k} = x_{i,j,k} + x_{i-1,m-1}$ /* $x_{i-1,m-1}$ is the last number in segment $\vec{S}_{i-1}$ */
end