Algorithm: Simple synchronizer ALPHA with unbounded space:
code for p_i, $0 \leq i \leq n-1$.

Initially round=0 and
buffer[*r*], *safe*[*r*] and *ack-missing*[*r*] are empty, for all $r \geq 1$

1: when Synch-send$_i$ (*S*) occurs:
2: *round* := *round* + 1
3: *ack-missing*[*round*] := {*j* : p_j is a recipient of a message in *S*}
4: enable Asynch-send$_i$(<*m*,round>) to p_j, for each *m* ε *S* with recipient p_j

5: when Asynch-recv$_i$(<m,r>) from *p*j occurs:
6: add (*m*,j) to *buffer*[*r*]
7: enable Asynch-send$_i$(<*ack*,*r*>) to p_j // acknowledge

8: when Asynch-recv$_i$(<*ack*,*r*>) from p_j occurs:
9: remove *j* from *ack-missing*[*r*]
10: if *ack-missing*[*r*] = Ø then // all neighbours have acknowledged
11: enable Asynch-send$_i$(<*safe*,*r*>) to all neighbors //p_i is safe

12: when Asynch-recv$_i$(<*safe*,*r*>) from p_j occurs:
13: add *j* to *safe*[*r*]
14: if *safe*[*r*] includes all neighbors then // all neighbors are safe
15: enable Synch-recv$_i$(*buffer*[*r*]) // for round *r*