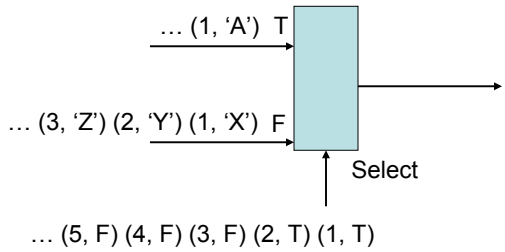
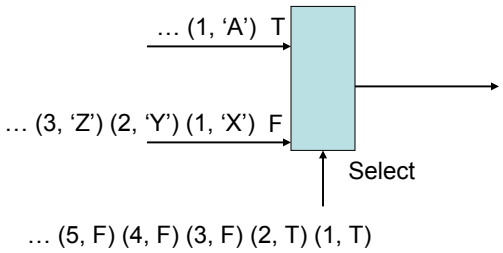
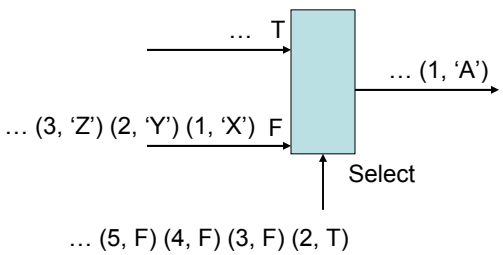


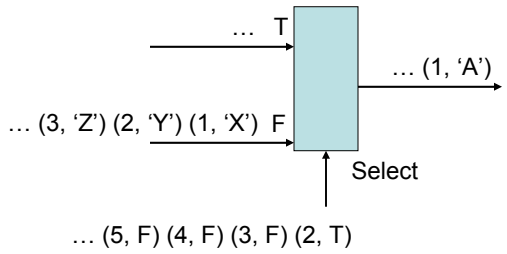
Without out-of-order execution



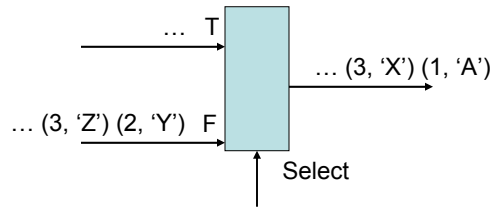


Blocked on reading from the T input.

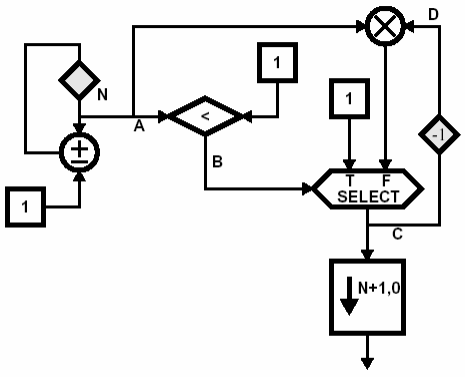
With out-of-order execution



… (5, F) (4, F) (3, F) (2, T)



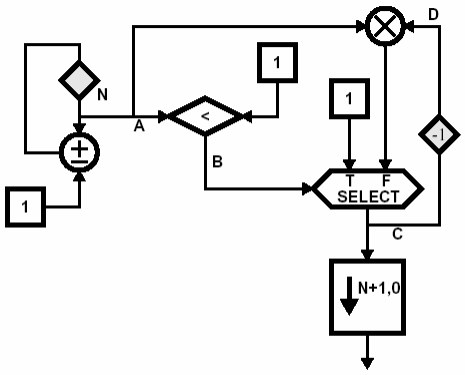
… (5, F) (4, F) (2, T)



A	B	C	D
N	F		
N-1	F		
N-2	F		
…	…		
2	F		
1	F		
0	T	1	

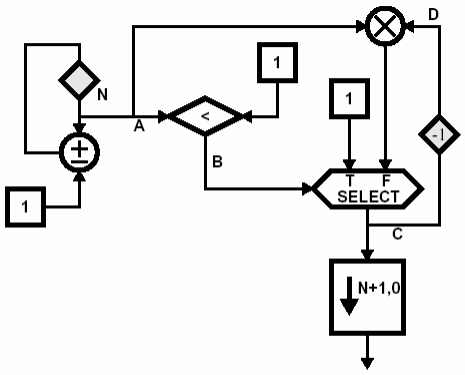
Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.

From: DATAFLOW PROCESS NETWORKS, Edward A. Lee and Thomas M. Parks
Proceedings of the IEEE, vol. 83, no. 5, pp. 773-801, May, 1995
http://ptolemy.eecs.berkeley.edu/publications/papers/95/processNets/



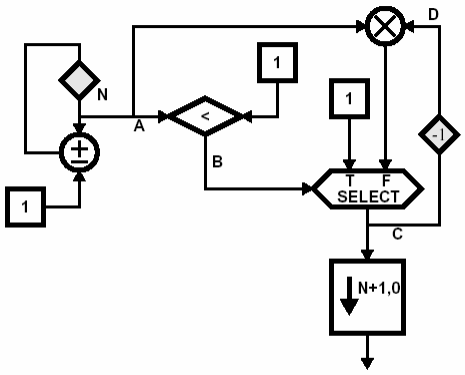
A	B	C	D
N	F		
N-1	F		
N-2	F		
...	...		
2	F		
1	F		1
0	T	1	

Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.



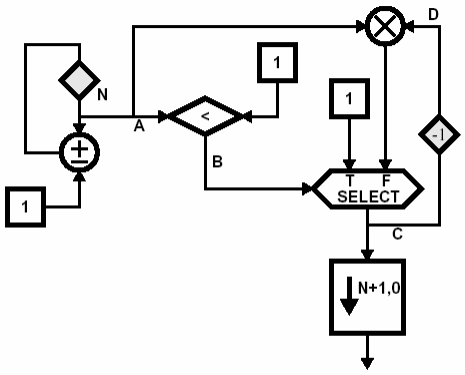
A	B	C	D
N	F		
N-1	F		
N-2	F		
...	...		
2	F		
1	F	1	1
0	T	1	

Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.



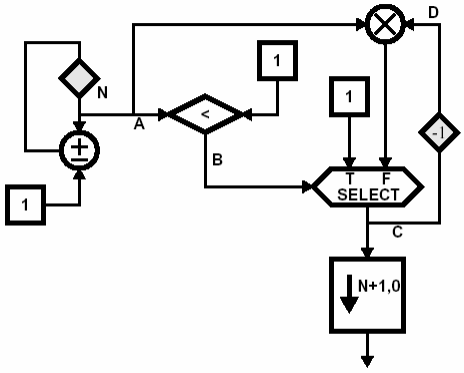
A	B	C	D
N	F		
N-1	F		
N-2	F		
...	...		
2	F		1
1	F	1	1
0	T	1	

Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.



A	B	C	D
N	F		
N-1	F		
N-2	F		
...	...		
2	F	2	1
1	F	1	1
0	T	1	

Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.



A	B	C	D
N	F	N!	(N-1)!
N-1	F	(N-1)!	(N-2)!
N-2	F	(N-2)!	(N-3)!
...	...	...	...
2	F	2	1
1	F	1	1
0	T	1	

Figure 25. This factorial program deadlocks without out-of-order execution, as provided for example by the tagged token model.