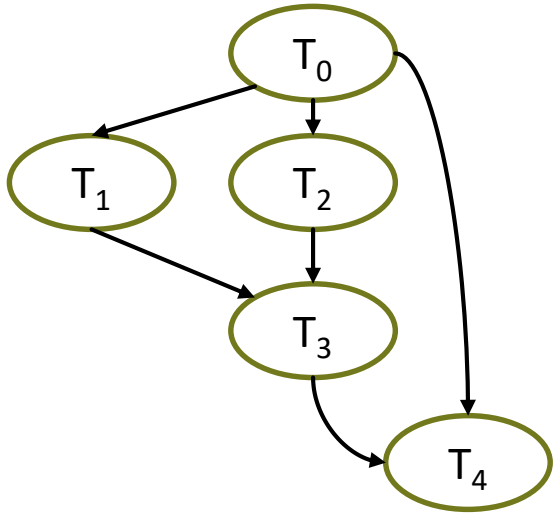
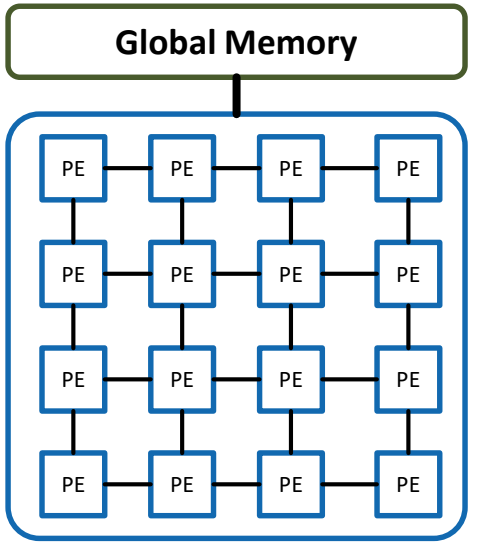
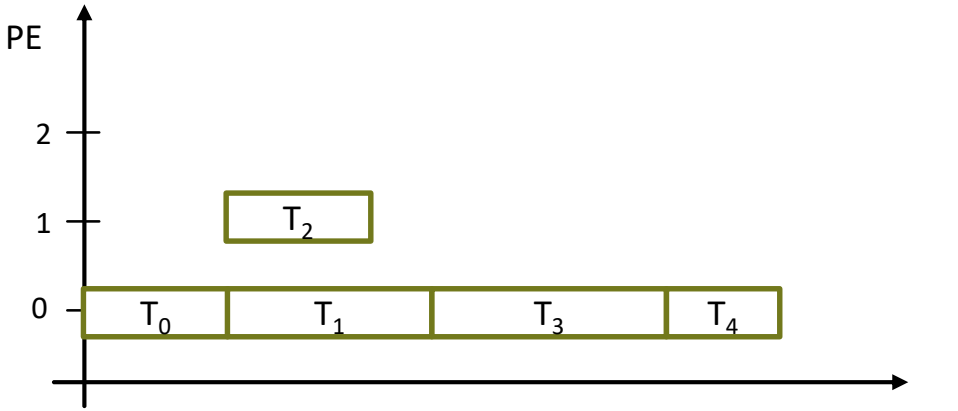
ASA: Scheduling

Scheduling





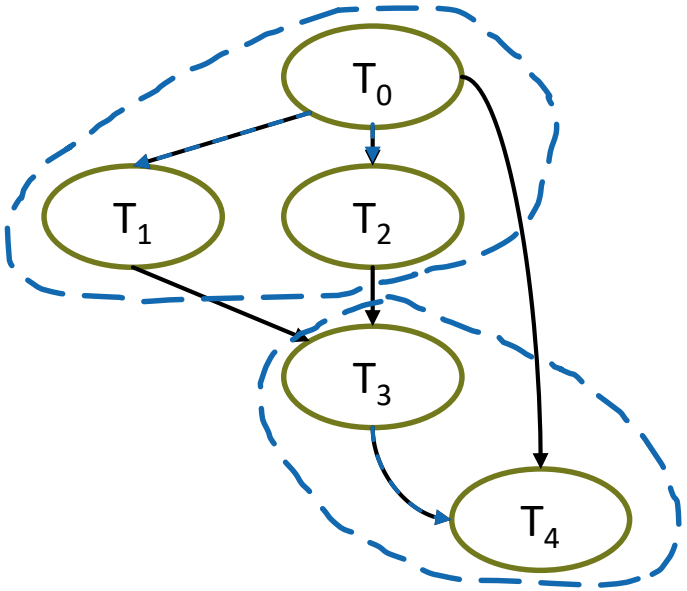
In traditional task scheduling, a task can start only when all the predecessors completed (compute-then-communicate)

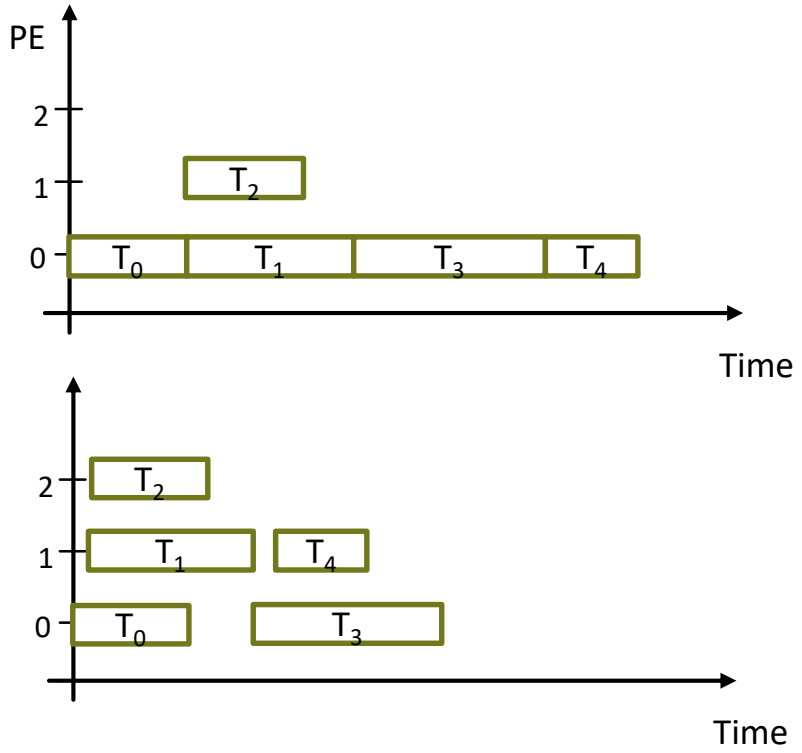


Execution trace on 3 PEs

Streaming Scheduling

We want to enable streaming communications between tasks





In this way we exploit spatial (pipeline) parallelism and reduce off-chip memory accesses

Solving this adds complexity to an already complicated problem:

- We need to understand whether it is better to stream or not, by building streaming blocks
- We need to understand how to schedule these streaming blocks