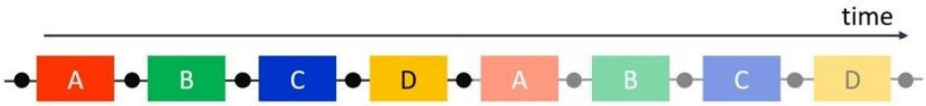
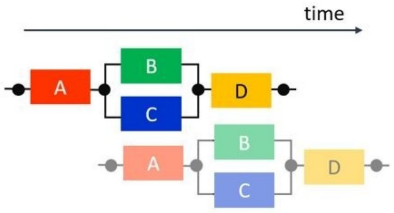
1. Is FPGA the von Neumann architecture?
2. What are the three paradigms for programming FPGAs?
3. Why the Producer-Consumer Paradigm can improve the performance on FPGA? Why Producer-Consumer Paradigm has limited impact on a CPU?
4. The communication pattern in Producer-Consumer Paradigm can be mapped to what types of data buffers?
5. What is the memory access pattern of the streaming data paradigm?
6. What are the differences between FIFO and Ping-pong? Briefly explain it.
7. Check the following code block and diagrams:
    a. In each Steps, what kinds of parallelism is exploited?

```
void diamond(data_t vecIn[N], data_t vecOut[N])
{
   data_t c1[N], c2[N], c3[N], c4[N];
   #pragma HLS dataflow
   A(vecIn, c1, c2);
   B(c1, c3);
   C(c2, c4);
   D(c3, c4, vecOut);
}
```

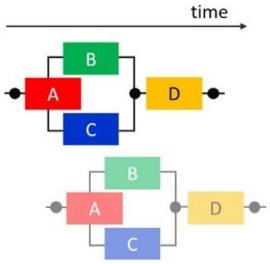
Step 1:



Step 2:



Step 3:



8. Read the Blocking/Non-Blocking Semantics.
9. What is the data-driven task-level parallelism?

10. What is the control-driven task-level parallelism?
11. What are the two main canonical forms for the dataflow optimization?
12. How to specify a PIPO using HLS pragma?
13. How to specify a FIFO using HLS pragma?
14. Run the example "**simple_data_driven**" and execute its simulation. Remove all the "hls_thread_local" In the "test.cpp" and see what will happen? Explain why it needs "hls_thread_local" for data_driven and how "hls_thread_local" affects the simulation.
15. Run the example "**using_fifos**" and "**using_pipos**". Please explain the coding styles for generating FIFO and PIPO. Compare the performance (clock cycles and hardware resources) of the two designs. Explain what the observations you obtain.