Name Surname: Date: 06.03.2025

Id:

Sign:

BLG 322E - COMPUTER ARCHITECTURE

Quiz 1

QUESTION-1:

A task **T** that will be executed on integers consists of seven suboperations X_i (i =1,2,3,4,5) and Y_i (i=1,2). These suboperations are implemented using combinational digital circuits with the following propagation delays:

X_1=**20ns**, X_2=**15ns**, X_3=**20ns**, X_4=**20ns**, X_5=**5ns**, Y_1=**25ns**, Y_2=**10ns.**

Fig. 1 on the right shows the block diagram of the circuit.

- The arrows between the suboperations indicate the dependencies between them. Therefore, the suboperations should be executed in the given order.
- Suboperation Y_1 can execute in parallel with X_2.
- Similarly, suboperation Y_2 can also execute in X_4.

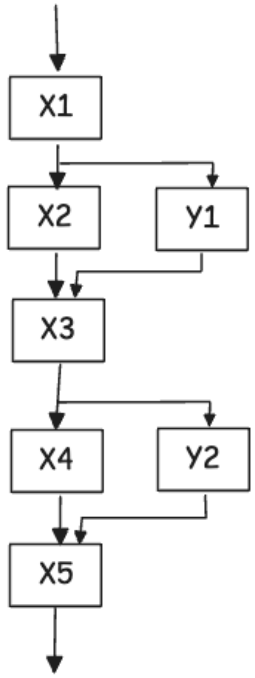

Fig. 1: Dependency diagram of the circuit.

a) (50 pts) Design a pipeline P_A that executes task **T** and meets the following constraints:

- The pipeline achieves the highest possible speedup if it is executed on an array with an infinite number of elements (maximize the speedup).
- Use a minimum number of registers with a propagation delay of 5 ns.
- Use the units given in the original circuit (X_i(i=1,2,3,4,5) and Y_i (i=1,2)).

b) (10 pts) How long does it take to execute the task only on the first element of array using the pipeline P_A? (**T_1** = ?)

c) (10 pts) Calculate the highest possible speedup pipeline P_A can achieve when it executes a task **T** on an array with an ***infinite*** number of elements.

Name Surname: Date: 06.03.2025

Id:

Sign:

Name Surname: Date: 06.03.2025

Id:

Sign:

QUESTION-2:

A compiler writer is evaluating two different machine-language code sequences, S1 and S2, which implement the same high-level operation. The hardware design team has provided the Cycles Per Instruction (CPI) values for each relevant instruction type, as follows:

Instruction Type	**A**	**B**	**C**
CPI_i	1	2	4

For a particular high-level statement (e.g., a loop, conditional, or arithmetic expression), the compiler can generate **two** alternative code sequences with the following **instruction counts**:

Code Sequence	**# of A Instructions**	**# of B Instructions**	**# of C Instructions**	**Total Instructions**
S1	3	2	1	6
S2	2	3	2	7

a) **(15 pts)** Compute the total clock cycles required by each code sequence. Determine which sequence finishes in fewer cycles (i.e., is faster).

b) **(15 pts)** Compute the effective CPI of each code sequence. State which sequence has the lower effective CPI.