

## Tekst Projektnog Zadatka

Realizovati (u proizvoljnom jeziku) simulator petofazne protočne obrade (IF, ID, EX, MEM, WB). Podržati sljedeće instrukcije: ADD, SUB, MUL, DIV, LOAD i STORE (sa njihovim standardnim značenjem). ADD, SUB, LOAD i STORE zahtijevaju 1 ciklus za izvršavanje (EX faza). MUL i DIV zahtijevaju 2 ciklusa za izvršavanje (EX faza). LOAD i STORE zahtijevaju 3 ciklusa za pristup memoriji. Smatrati da se IF, ID i WB faze uvijek izvrše u jednom ciklusu. Svaka instrukcija raspolaže sa maksimalno tri registarska operanda (odredišni registar i izvorišni registar kod unarnih operacija, odnosno odredišni registar i dva izvorišna registra kod binarnih operacija). Kao ulaz prosljeđuje se niz instrukcija. U tabelarnoj formi prikazati izlaz koji za svaki ciklus (kolonu) prikazuje faze (ili zastoje) u kojima se nalaze ulazne instrukcije (vrste). Omogućiti da se specificira da li će se vršiti prosljeđivanje podataka između protočnih stepeni ili neće. Omogućiti ispis slučajeva u kojima dođe do prosljeđivanja podatka, sa svim relevantnim informacijama (redni brojevi instrukcija i njihove faze izvršavanja pri prosljeđivanju). Realizovati detekciju hazarda (WAR, RAW i WAW zavisnosti) između instrukcija u ulaznom nizu.

## Text of the Project Task

Realize (in an arbitrary language) a simulator of five-phase flow processing (IF, ID, EX, MEM, WB). Support the following instructions: ADD, SUB, MUL, DIV, LOAD and STORE (with their standard meaning). ADD, SUB, LOAD and STORE require 1 cycle to execute (EX phase). MUL and DIV require 2 cycles to execute (EX phase). LOAD and STORE require 3 cycles to access memory. Consider that the IF, ID and WB phases are always performed in one cycle. Each instruction has a maximum of three register operands (destination register and source register for unary operations, or destination register and two source registers for binary operations). A string of instructions is passed as input. Display the output in tabular form, which for each cycle (column) shows the phases (or stops) in which there are input instructions (types). Allow to specify whether or not to pass data between flow stages. Enable the printing of cases in which data is forwarded, with all relevant information (serial numbers of instructions and their stages of execution during forwarding). Realize hazard detection (WAR, RAW and WAW dependencies) between instructions in the input sequence.

## Text der Projektaufgabe

Realisiere (in einer beliebigen Sprache) einen Simulator der fünfphasigen Flussverarbeitung (IF, ID, EX, MEM, WB). Unterstützt die folgenden Anweisungen: ADD, SUB, MUL, DIV, LOAD und STORE (mit ihrer Standardbedeutung). ADD, SUB, LOAD und STORE erfordern 1 Zyklus zur Ausführung (EX-Phase). MUL und DIV benötigen 2 Zyklen zur Ausführung (EX-Phase). LOAD und STORE benötigen 3 Zyklen, um auf den Speicher zuzugreifen. Bedenken Sie, dass die IF-, ID- und WB-Phasen immer in einem Zyklus durchgeführt werden. Jeder Befehl hat maximal drei Registeroperanden (Zielregister und Quellregister für unäre Operationen oder Zielregister und zwei Quellregister für binäre Operationen). Als Eingabe wird eine Reihe von Anweisungen übergeben. Zeigen Sie die Ausgabe in tabellarischer Form an, die für jeden Zyklus (Spalte) die Phasen (oder Stopps) anzeigt, in denen Eingabeanweisungen (Typen) vorhanden sind. Ermöglicht die Angabe, ob Daten zwischen Ablaufphasen weitergegeben werden sollen oder nicht. Ermöglichen Sie das Drucken von Fällen, in denen Daten weitergeleitet werden, mit allen relevanten Informationen (Seriennummern von Anweisungen und deren Ausführungsstadien während der Weiterleitung). Realisieren Sie die Gefahrenerkennung (WAR-, RAW- und WAW-Abhängigkeiten) zwischen Anweisungen in der Eingabesequenz.