

# UNIVERZITET U BEOGRADU ELEKTROTEHNIČKI FAKULTET

*Katedra za elektroniku*

*Predmet: Uvod u projektovanje integrisanih kola*



**Projekat: Cyclic redundancy check (CRC)**

***Faza 5: CRC sistem za detekciju grešaka u prenosu***

Rok za predaju:

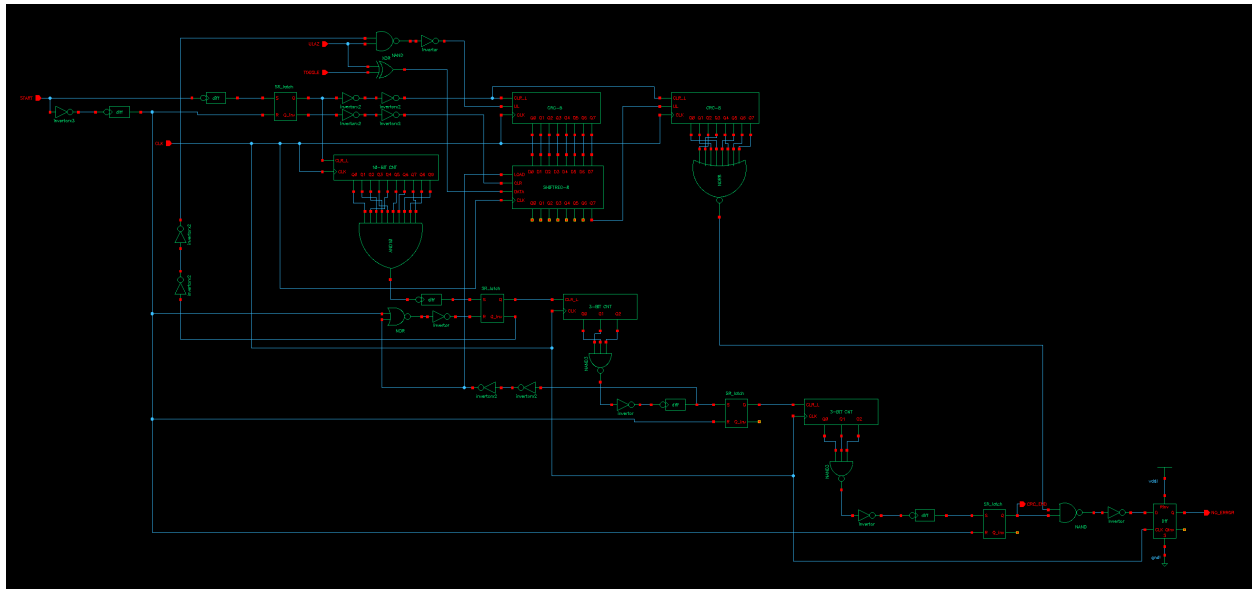
Projekat radili:

Ime	Prezime	broj indeksa
Dejan	Petković	77/2012
Predrag	Kuzmanović	49/2012

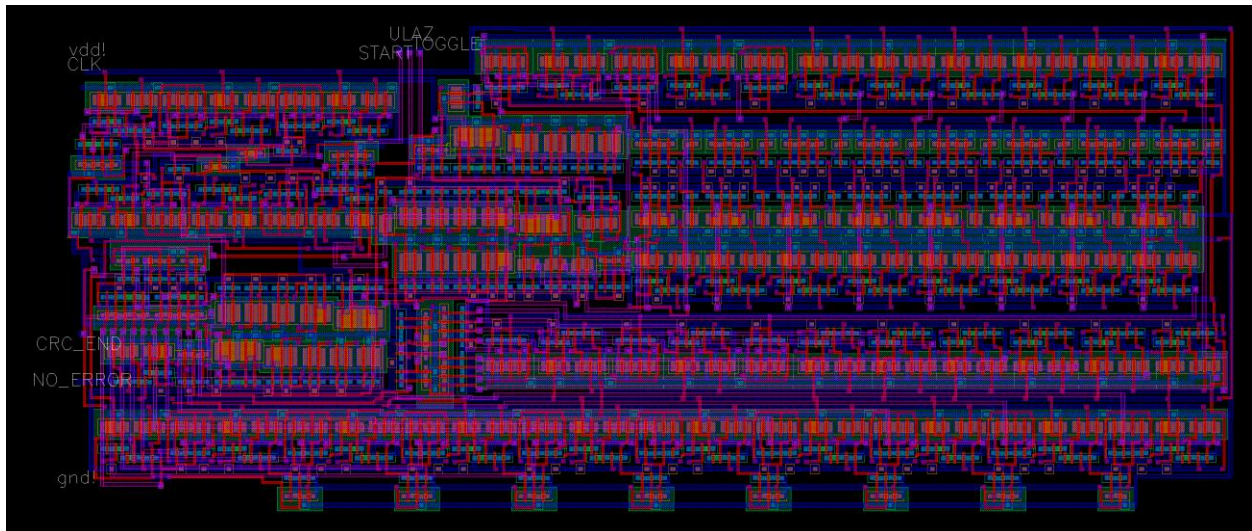
Izveštaj slati na e-mail: [elmezeni@el.etf.rs](mailto:elmezeni@el.etf.rs), [petrovicv@etf.rs](mailto:petrovicv@etf.rs)

Staviti u cc i predmetnog profesora: [jelena@etf.rs](mailto:jelena@etf.rs)

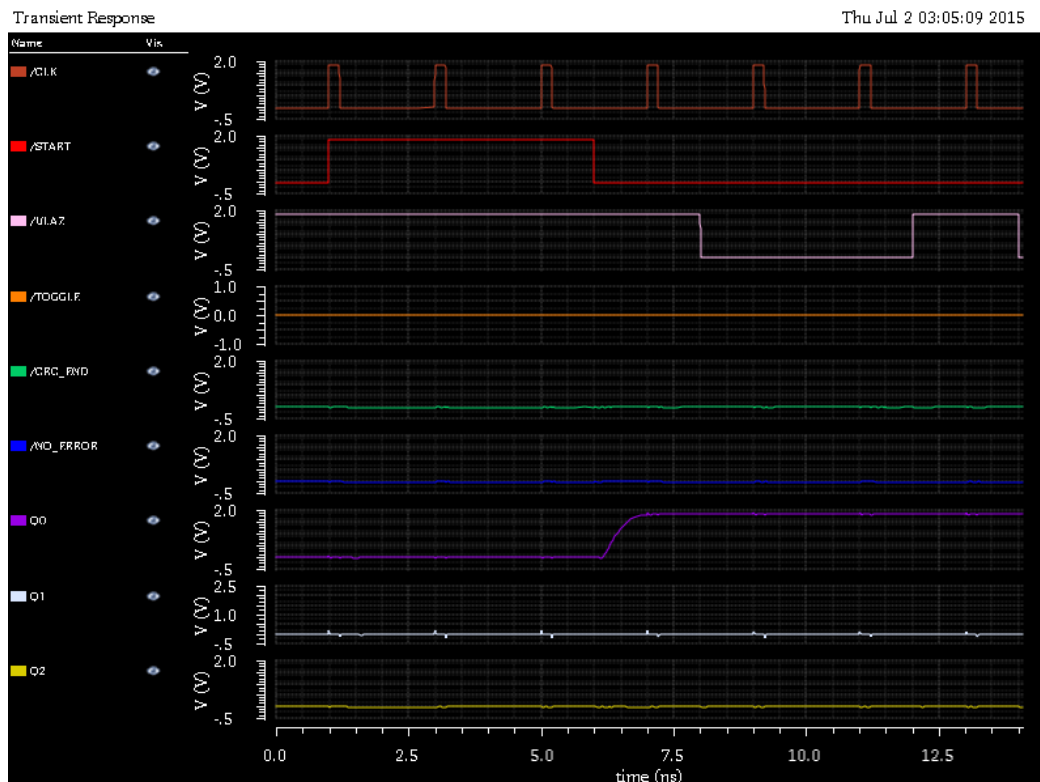
## Zadatak – Projektovanje celokupnog sistema



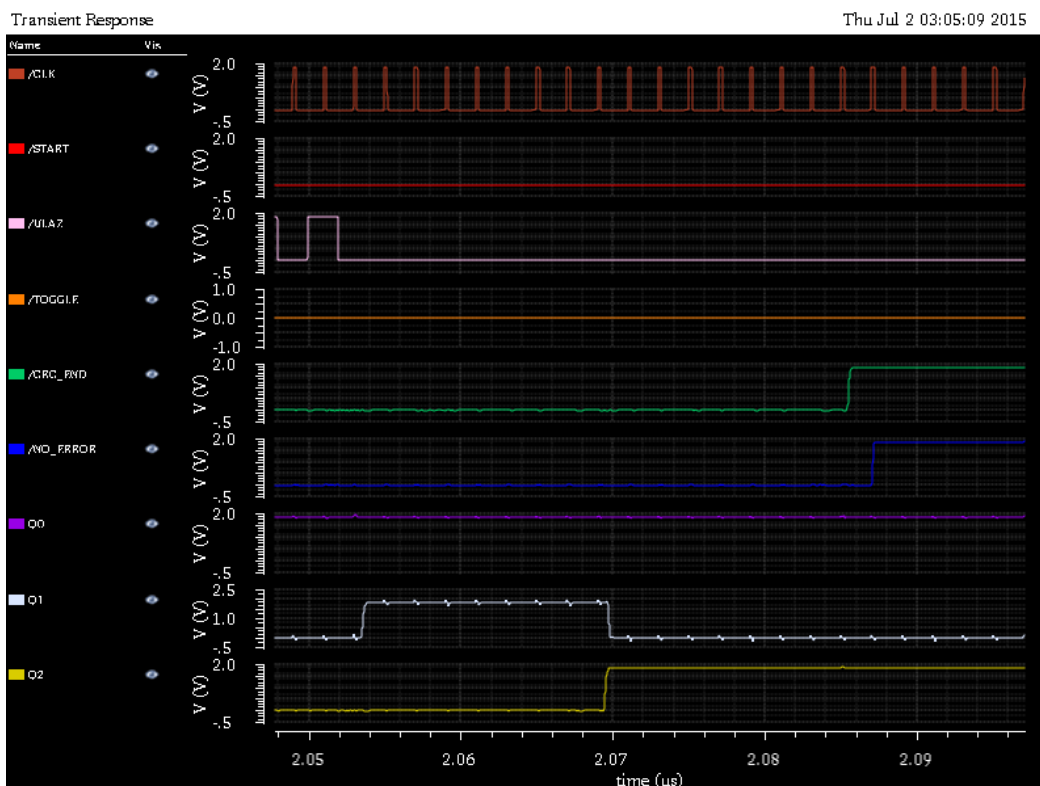
Slika 1.1. Schematic celokupnog sistema nacrtan u Virtuoso Schematic Editor-u



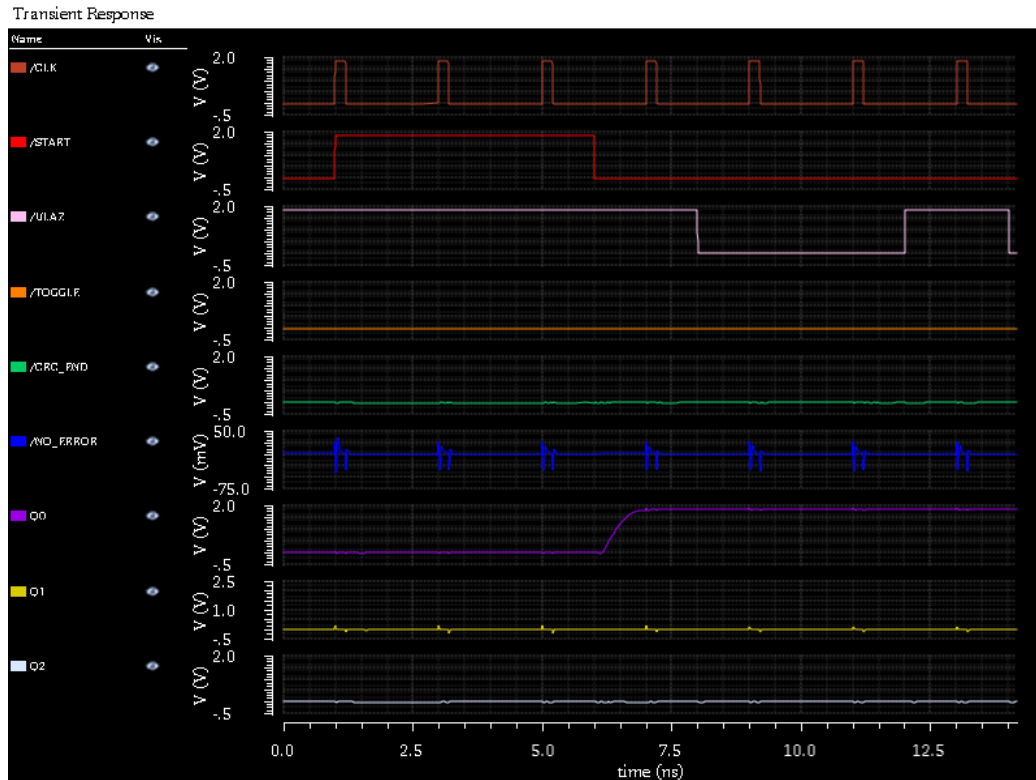
Slika 1.2. Layout celokupnog sistema izrađen u Virtuoso Layout Suite editoru (sa uključenim ruler-om)



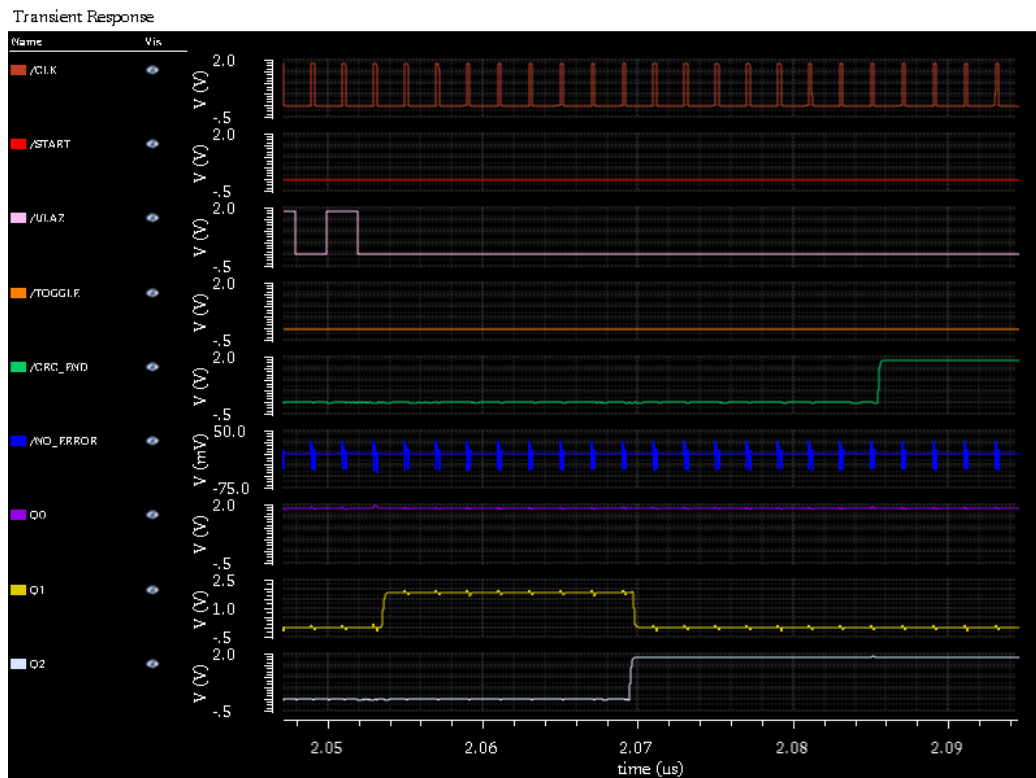
Slika 1.3. Dijagrami relevantnih signala dobijeni post-layout simulacijom, bez grešaka u prenosu, 1. deo



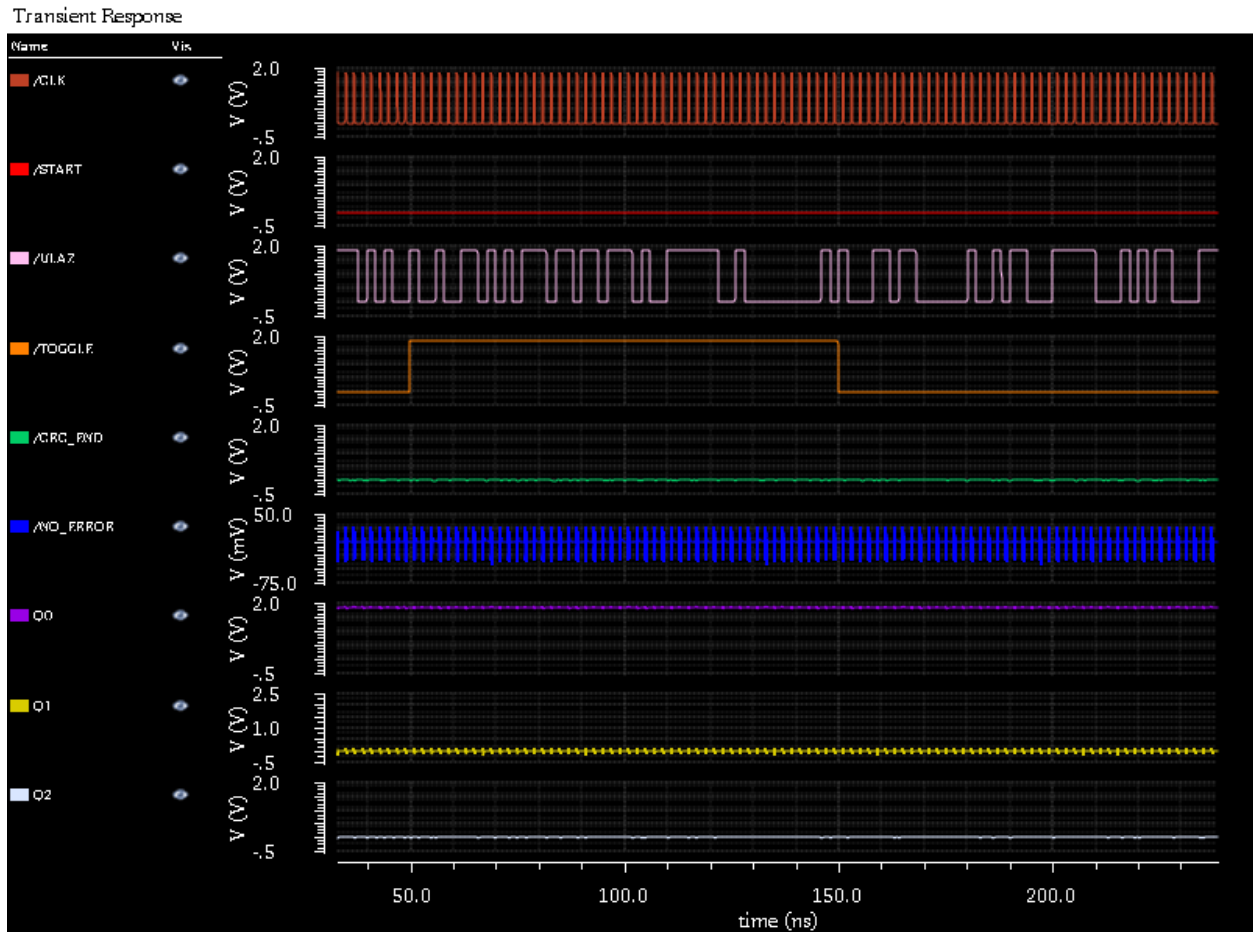
Slika 1.4. Dijagrami relevantnih signala dobijeni post-layout simulacijom, bez grešaka u prenosu, 2. deo



Slika 1.5. Dijagrami relevantnih signala dobijeni post-layout simulacijom, sa greškom u prenosu, 1. deo



Slika 1.6. Dijagrami relevantnih signala dobijeni post-layout simulacijom, sa greškom u prenosu, 2. deo



Slika 1.7. Prikaz aktiviranja *TOGGLE* signala radi generisanja greške u prenosu, post-layout simulacija

Dimenzije layout-a celokupnog sistema (širina i visina u lambda dužinama i mikrometrima):

$93.015 \mu\text{m} (1033.5 \lambda) \times 39.015 \mu\text{m} (433.5 \lambda)$

Minimalna perioda taktnog signala (odrediti sa korakom 100ps): 2ns

## Zaključak:

Zbog velikih opetretnih kapacitivnosti, bilo je potrebno baferisati signal  $V_{k2}$  kojim se vrši reset 1. SR leča (odnosno signala Q1) i paralelan upis u pomerački registar. Takođe, bilo je potrebno baferisati i invertovani izlaz 1. SR leča Q1\_inv koji se preko AND kola vraća na ulaz predajnika.

Što se tiče izrade leajuta za ovu fazu, najpre su izmerene dimenzije svih komponenti i napravljen je vrlo značajni, veliki generalni uvid u pozicije svih signala koje je potrebno povezati, kao i u to sa kakvim se već leajutima raspolaže. Zatim je šema podeljena u nekoliko blokova koji će se posebno realizovati, pa zajedno sklopiti u jedan leajut, korišćenjem svog sakupljenog znanja i poštovanjem svih znanih pravila. Ovde je primećen značaj dosadašnjeg „razmišljanja unapred“, mada je opet izvedeno nekoliko modifikacija delova nekih kola. Na primer, napravljeno je nekoliko verzija osnovnih logičkih kola (različitih dimenzija kanala PMOS tranzistora) koji će se bolje uklapati sa ostalim komponentama u bloku u kome se nalaze, ali i uticati na bolji i brži rad sistema. Na primer, primećeno je da „slabiji“ baferi remete rad kola zbog male brzine prenosa signala sa ulaza na izlaz, ili da će ponekad nekompatibilne komponente manje površine rušiti estetiku leajuta i otežavati rad. Dalje, može se primetiti da u projektu nigde nije korišćen metal 3, što je prednost. Pošto je izbegavano korišćenje mnogo metala 2 u komponentama niže hijerarhije, sada je omogućeno raspolaganje njime za povezivanje svih unutrašnjih signala koji se ne mogu ostvariti u metalu 1 ili možda polisilicijumu. Veliki dobitak ne bi bio ostvaren korišćenjem metala 3 jer su duže linije u metalu 2 relativno udaljene, pa bi se praktično vodile tim istim putem i u metalu 3. Inače, te duge linije su uglavnom potrebne jer se na primer, široko razmaknuti izlazi 10-bitnog brojača vode na uske ulaze AND10 kola koji su na minimalnim razmacima jedni od drugih. Isto važi i za ulaze NOR8 kola. Što se tiče globalnih signala, poštovano je pravilo da se napajanje i masa vode kroz kolo kroz putanju u obliku nadovezanog latiničnog slova S. Sličnim rezonom je ostvareno povezivanje taktnog signala CLK. Konačan oblik leajuta je pravougaonik, iz već poznatog razloga - velike dužine 10-bitnog brojača.

Za testiranje ispravnosti rada sistema bilo je potrebno generisati stimulus fajlove sa ulazima velikog trajanja (više od 1024 perioda takta). U tu svrhu, napisan je C++ program koji generiše te stimulus fajlove i na taj način je proces pisanja testova automatizovan. Program je interaktivne prirode, tj. korisnik ima slobodu u izboru parametara ulaznih signala. U dodatku je priložen kompletan kod ovog programa. Moguće je izabrati željeno trajanje periode taktnog signala, kao i osobine signala START kojim započinje rad sistema. Ovo je omogućilo da se na jednostavan način izračuna minimalna perioda taktnog signala, koja iznosi 2ns (što odgovara frekvenciji od 500MHz). Pri računanju minimalne periode taktnog signala, uzeto je u obzir ograničenje po pitanju kašnjenja koje unosi 10-bitni brojač.

Dodatak:

*Kod za generisanje stimulus fajlova napisan u programskom jeziku C++:*

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <iomanip>
using namespace std;

int main() {
    double delay = 0.005; // kasnjenje prelaza, 5ps
    unsigned inputNumber = 1024; // broj bita u 1 bloku koji se stiti CRC8 kodom
    double Vdd = 1.8; // napon logicke "1"
    ofstream outputFile;
    outputFile.open("PROJEKAT_test2"); // 2: sa TOGGLE - om

    // zaglavlje fajla
    outputFile << "simulator lang=spectre"; outputFile.put('\n');
    outputFile << "Vdd (vdd! 0) vsource dc=1.8"; outputFile.put('\n');
    outputFile << "Gnd (gnd! 0) vsource dc=0"; outputFile.put('\n');

    // ulazni signali: CLK, START, TOGGLE, ULAZ
    double T, delay_START, T_START; // parametri za generisanje vremenskih oblika signala
    cout << "Unesite zeljenu vrednost periode takta [ns]: "; cin >> T;
    // cout << "T: " << T << endl;
    cout << "Unesite kasnjenje impulsa signala START [ns]: "; cin >> delay_START;
    cout << "Unesite trajanje impulsa signala START [ns]: "; cin >> T_START;

    unsigned delay_CLK; // ovo racunamo
    delay_CLK = delay_START + T_START + T/2;
    while (delay_CLK >= T) delay_CLK -= T; // "po modulu" T

    // za CLK se uzima periodicni takt sa Dclk = 10%, pomeren za 1ns, i periodom T
    outputFile<<"v1 (CLK 0) vsource delay="<<delay_CLK<<"n type=pulse val0=0 val1="<<Vdd;
    outputFile<<" rise="<<delay<<"n fall="<<delay<<"n width="<<0.1*T<<"n period="<<T<<"n";
    outputFile.put('\n');

    // za START je uzet pozitivan impuls trajanja T_START, koji pocinje u delay_START
    outputFile<<"v2 (START 0) vsource type=pwl wave=[[0 0 "<<delay_START<<"n 0 "<<delay_START+delay<<"n "<<Vdd<<" ";
    outputFile<<delay_START+T_START<<"n "<<Vdd<<" "<<delay_START+T_START+delay<<"n 0]";
    outputFile.put('\n');

    // TOGGLE neka bude neaktivno za sada, posle cemo testirati i tu funkcionalnost, da bi se detektovala greska
    //outputFile << "v3 (TOGGLE 0) vsource type=pwl wave=[[0 0]"; outputFile.put('\n');
    outputFile<<"v3 (TOGGLE 0) vsource type=pwl wave=[[0 0 50n 0 50.005n 1.8 150n 1.8 150.005n 0]";
    outputFile.put('\n');

    // za ULAZ se uzima da se menja "usred" periode takta; svaki put kad treba ULAZ da se menja generise sa 0 ili 1
    unsigned x = rand() % 2; // 0 or 1
    long double curTime; // dokle smo dosli, ns
    outputFile << fixed;
    outputFile.precision(3);
    // ULAZ je u "protivfazi" sa CLK-om
    outputFile<<"v4 (ULAZ 0) vsource type=pwl wave=[[0 "<<x*Vdd<<" "<<delay_START+T_START<<"n "<<x*Vdd;
    curTime = delay_START + T_START;
    for (int i = 0; i < inputNumber; i++) {
        unsigned x = rand() % 2; // 0 or 1
        // prelaz
        curTime += delay;
        outputFile << " " << curTime << "n " << x*Vdd;
        curTime += T - delay;
        outputFile << " " << curTime << "n " << x*Vdd;
    }
    // zavrsetak, ]
    outputFile << "]"> outputFile.put('\n');
    // kapacitivna opterecenja za CRC_END i NO_ERROR
    unsigned CL = 3; // u fF
    outputFile<<"c0 (0 NO_ERROR) capacitor c="<<CL<<"f"; outputFile.put('\n');
    //outputFile << "c10 (0 NO_ERROR_inv) capacitor c=" << CL << "f"; outputFile.put('\n');

    // kraj
    outputFile.close();
    system("PAUSE");
    return 1;
}
```