

1. Omogućiti da se program **Convert C to F (2_1).vi** može koristiti i kao potprogram, tj. formirati **subvi Convert C to F (2_1).vi**.

2. Napisati program u LabVIEW-u koji računa zbir dva broja.

- Startovati LabVIEW
- Birati **New VI**
- U okviru **Controls Palette** birati numerički kontroler tipa **Digital Control**, i postaviti ga na Front panel. Kontroler nazvati *broj 1*
- Ponoviti prethodnu tačku i drugi kontroler nazvati *broj 2*
- U okviru **Controls Palette** birati numerički indikator tipa **Digital Indicator**, i postaviti ga na Front panel. Indikator nazvati *zbir*
- Preći na Block Diagram
- U okviru **Functions Palette** birati funkciju **Numeric**, tipa **Add** i postaviti je u blok dijagram
- U okviru **Tools Palette** birati **Wiring Tool** i izvršiti odgovarajuće povezivanje terminala i ikone funkcije u blok dijagramu
- Vratiti se na Front Panel, upisati željene vrednosti sabiraka i pokrenuti program.

3. Napisati program u LabVIEW-u koji proširuje prethodni primer, pa na osnovu izbora korisnika, računa zbir, razliku, proizvod ili deljenje dva broja.

- U front panel prethodnog primera dodati numerički kontroler tipa **Vertical Pointer Slide**, i nazvati ga *operacija*
- Promeniti opseg skale kontrolera tako da se opseg kreće od 0 do 3.
- Promeniti reprezentaciju brojeva na skali kontrolera *operacija* tako što treba birati opciju Byte (I8). Reprezentacija objekata na front panelu nalazi se u okviru menija klikom na desni taster nakon pozicioniranja na željeni objekat.
- pored brojeva na skali (0, 1, 2 i 3) uneti oznake koji položaj predstavlja koju operaciju. Birati:

0: zbir

1: razlika

2: proizvod

3: deljenje

- Promeniti naziv postojećeg indikatora *zbir* u *rezultat*
- Preći na Block Diagram
- Izbrisati žice postavljene u primeru 1
- U okviru **Functions Palette** birati strukturu **Case** i postaviti je u blok dijagram
- Povezati kontroler *operacija* na ulaz **Case** strukture tj. na uslovni terminal iste. Obratiti pažnju na promenu boje uslovnog terminala usled prilagođavanja vrsti podataka vezanim za njega.
- U okviru iste strukture promeniti i nazive okvira, tj. pored dva postojeća (0 i 1) dodati još dva slučaja (2 i 3). To se postiže:
 - strelicom doći do oznake slučaja 1

- desnim klikom na strelicu otvara se meni u kojem treba birati opciju **Add Case After** nakon čega se pojavljuje i slučaj 2 u datoj **case** strukturi. Ponoviti još jednom ceo postupak kako bi se pojavio i slučaj 3.

- Na taj način dobijena je case struktura sa 4 uslova:
 - 0 - zbir
 - 1 - razlika
 - 2 - proizvod
 - 3 - deljenje
- Strelicama se zaustaviti na prozoru uslova "0". Obzirom da ovaj uslov znači "zbir", u okvir "0" treba postaviti numeričku funkciju **Add** koja se može iskoristiti iz prethodnog primera
- Strelicom preći na prozor uslova "1". Obzirom da ovaj uslov znači "razlika", u okvir "1" treba postaviti numeričku funkciju **Subtract**
- Strelicom preći na prozor uslova "2". Obzirom da ovaj uslov znači "proizvod", u okvir "2" treba postaviti numeričku funkciju **Multiply**
- Strelicom preći na prozor uslova "3". Obzirom da ovaj uslov znači "deljenje", u okvir "3" treba postaviti numeričku funkciju **Divide**
- Povezati na odgovarajući način sve terminale i ikone funkcija u blok dijagramu
- Preći na front panel, birati operaciju i vrednosti ulaznih promenljivih, pokrenuti program i posmatrati rezultat.

4. Napisati program u LabVIEW-u koji od 2 string kontrolera i jednog numeričkog kontrolera sa ulaza:

prvi string: "rezultat merenja iznosi"

broj: 5.00

drugi string: "volti"

formira jedinstveni string:

"rezultat merenja iznosi 5.00 volti"

i prikazuje dužinu konačnog stringa.

5. Napisati program u LabVIEW-u koji vrši generisanje 2 niza slučajnih brojeva koji su prethodno skalirani (pomnoženi) faktorom 100 koristeći :

- **For** petlju (broj elemenata niza birati definisanjem preko numeričkog kontrolera)

- **While** petlju (obezbediti zaustavljanje **while** petlje kada broj elemenata niza koji se formira ovom petljom postane isti kao broj elemenata niza koji se formira **for** petljom)

Obe realizacije raditi u okviru istog blok dijagrama odnosno u okviru istog **vi** programa. Prvi niz nazvati "niz for petlje" a drugi "niz while petlje".

Koja od ove dve petlje se izvršava prva a koja druga? Šta bi trebalo uraditi da bi se redosled izvršavanja petlji regulisao, tako da se **for** petlja npr. vrši prva a **while** druga? Koju strukturu bi trebalo primeniti u tom cilju?

6. Napisati program u LabVIEW-u koji traži maksimalnu vrednost u seriji od 100 slučajnih brojeva. Maksimalnu vrednost prikazati na front panelu, a takođe obezbediti i grafički prikaz serije slučajnih brojeva. Pomoć: koristiti **for** petlju, funkciju za traženje maksimalne vrednosti **Max & Min** u okviru menija **Comparison**, unutar **Functions Palette**, kao i **shift** registar.

7. Napisati program u LabVIEW-u koji vrši usrednjavanje niza slučajnih brojeva pri čemu je prozor usrednjavanja 3 odbirka (tj. 3 slučajna broja). Dakle, treba omogućiti pamćenje vrednosti 3 poslednja slučajna broja, najpre prvog, drugog i trećeg, naći im srednju vrednost i ucrtati na grafik, zatim sledeća tri broja, odnosno, drugog, trećeg i četvrtog i ponoviti postupak traženja srednje vrednosti, pa preći na treći, četvrti i peti slučajni broj i ponavljati postupak dok god korisnik ne zaustavi proces. Rezultat prikazati u grafičkom obliku na indikatoru tipa **waveform chart**, a kontrolu procesa regulisati preko **while** petlje omogućavajući da korisnik sam bira kada će se proces zaustaviti (korišćenjem logičkog kontrolera **Stop**). Pomoć: za pamćenje uzastopnih vrednosti slučajnih brojeva koristiti **shift** registre. Vreme izvršavanja **while** petlje vremenski regulisati tako da se obezbedi pauza između uzastopnih iteracija od pola sekunde (500ms). Takođe, obezbediti prikazivanje trenutnog slučajnog broja na front panelu.