LabVIEW

Dr Marko Dimitrijević

Programske strukture

Programske strukture

- Case strukture.
- Sequence struktura.
- Stacked Sequence struktura.
- Formula Node struktura.

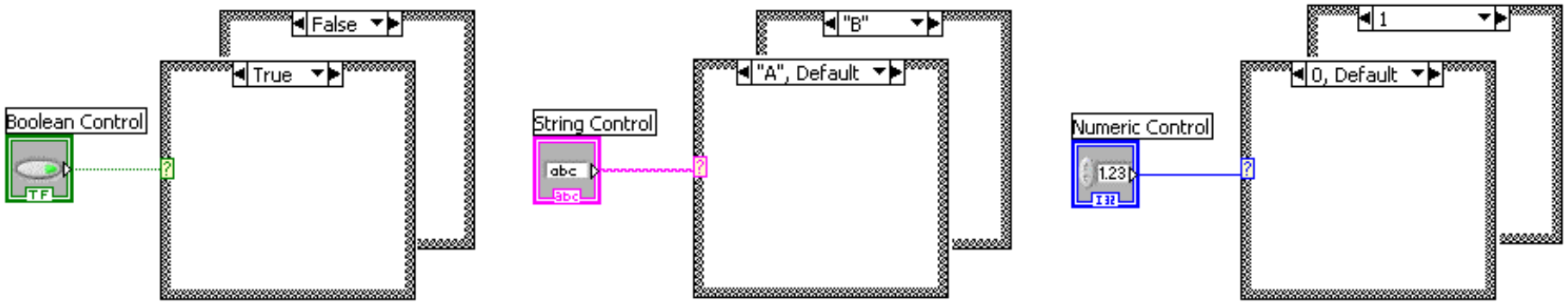
Case strukture

- **Case strukture** omogućuju izvršenje različitog kôda u zavisnosti od postavljenih uslova.
- U tekstualnim jezicima odgovaraju IF ili CASE iskazima.

Case strukture

- Nalazi se u Programming/Structures paleti
- Uokviriti postojeći kôd ili ga prevući u strukturu
- Kôd je naslagan kao špil karata, samo je jedan slučaj vidljiv

Programming /Structures

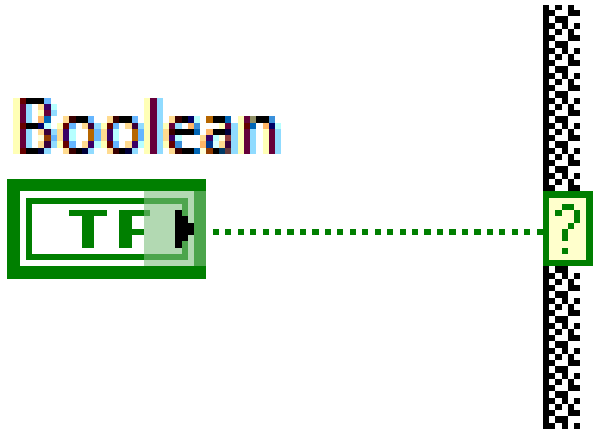


Subdijagrami

- Case struktura ima nekoliko subdijagrama, od kojih se samo jedan izvršava prilikom izvršavanja strukture.
- Samo je jedan subdijagram vidljiv, može se izabrati iz padajućeg menija sa vrha strukture.
- **Labela selektora** na vrhu strukture ukazuje na vidljiv subdijagram kao i vrednost selektorske promenljive.

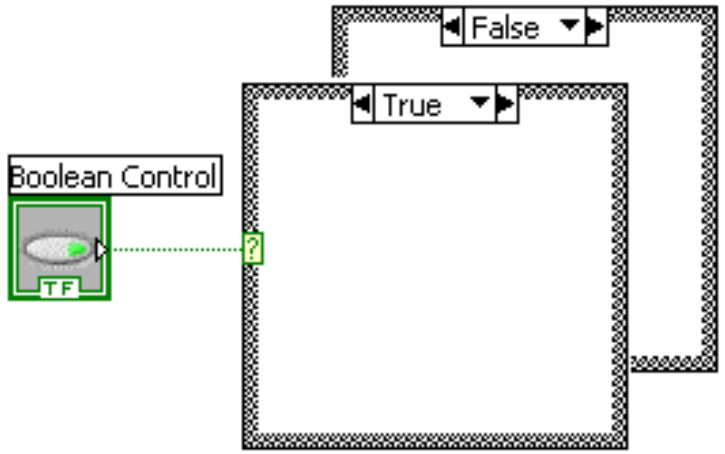
Terminal selekcije

- Case struktura ima jedan ulazni terminal, tzv. selektroski terminal (**selector terminal)**.
- Slektroski terminal je obeležen **?,** određuje koji subdijagram u Case strukturi će biti izvršen.
- Selektroski terminal se vezuije direktno za kontrolu ili izlaz funkcije.



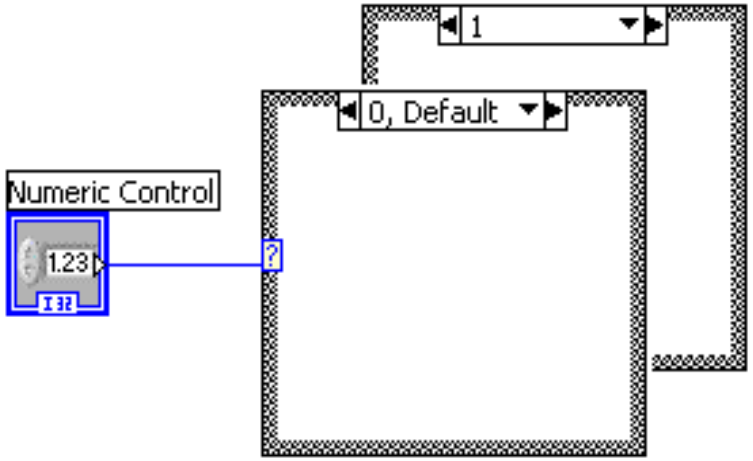
Primer Case strukture: logička promenljiva

- Case struktura sa logičkom promenljivom vezanom za selektroski terminal će imati samo dva subdiagrama: jedan koji će se izvršavati ukoliko je promenljiva **True**, drugi ukoliko je promenljiva **False**.



Primer Case strukture: celobrojna promenljiva

- Case struktura sa numeričkim celobrojnim selektorkim terminalom može imati proizvoljan broj subdijagrama: biće izvršen onaj koji ogovara promenljivoj vezanoj za selektorski terminal.



Labela selektora

- Numerički selektor može biti
  - Jedan celi broj, npr. 5
  - Lista celih brojeva, kao 1, 5, 11
  - Opseg brojeva, kao
    - 10..20 podrazumeva opseg od 10 do 20
    - ..10 podrazumeva brojeve manje ili jednake 10
    - 10.. podrazumeva brojeve veće ili jednake 10
- Selektorke labele moraju biti celi brojevi.

Dodavanje i brisanje subdijagrama

- Desnim klikom na strukturu može se izabrati:
  - **Add Case Before** ili **Add Case After** za dodavanje subdijagrama
  - **Duplicate Case** za kopranje subdijagrama u novi, sa novim uslovom.
  - **Delete This Case** za brisanje subdijagrama.

Dodavanje i brisanje subdijagrama

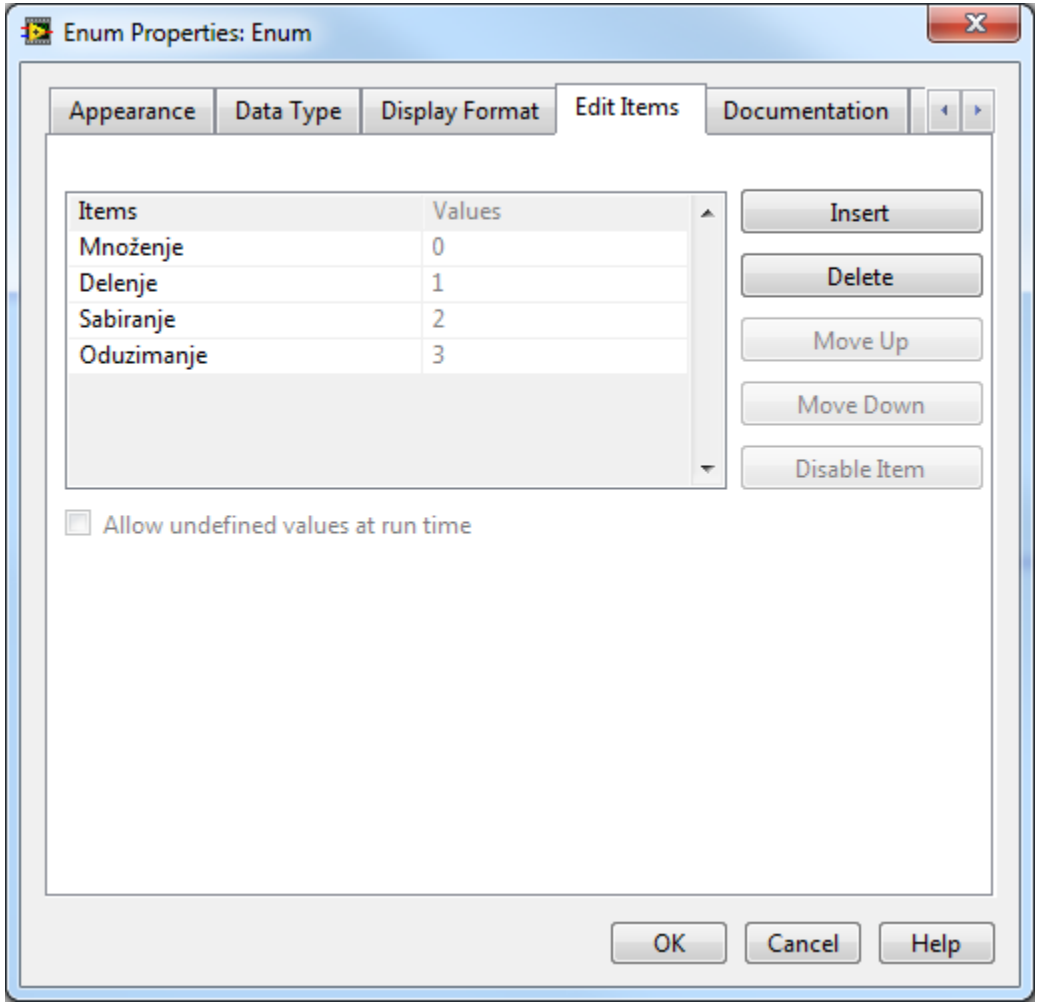
- Numeričke Case strukture imaju **default case**, subdijagram koji se izvršava ukoliko nijedan uslov nije ispunjen.
- Kako bi se subdijagram proglasio za default case, potrebno je desnim klikom na ivicu strukture izabrati opciju **Make This The Default Case.**

Enumeracije

- Enumeracije (Enum) su tip celobrojnih promenljivih u određenom, konačnom opsegu
- Mogu se dodati kao kontrole na front panel iz palete **Ring & Enum**, ili na blok dijagramu kao onstanta izborom **Programming>>Numeric>>Enum Constant**
- Svakoj vrednosti se dodeljuje celi broj sekvencijalno

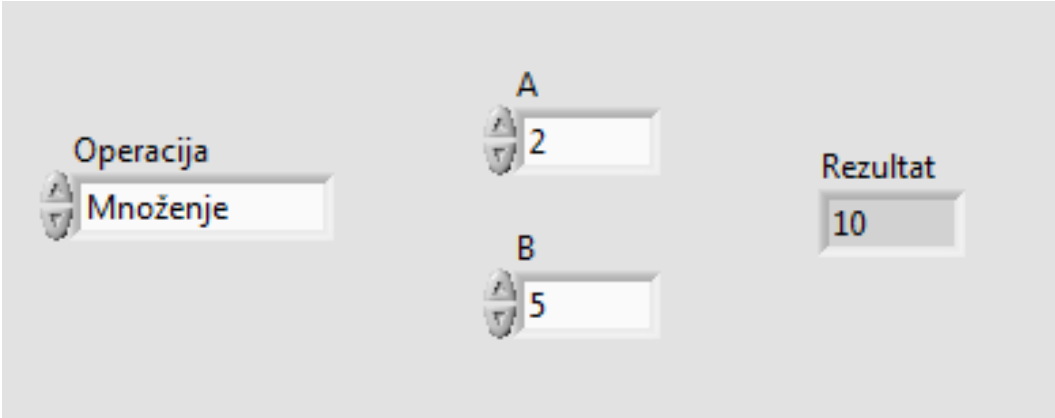
Enumeracije

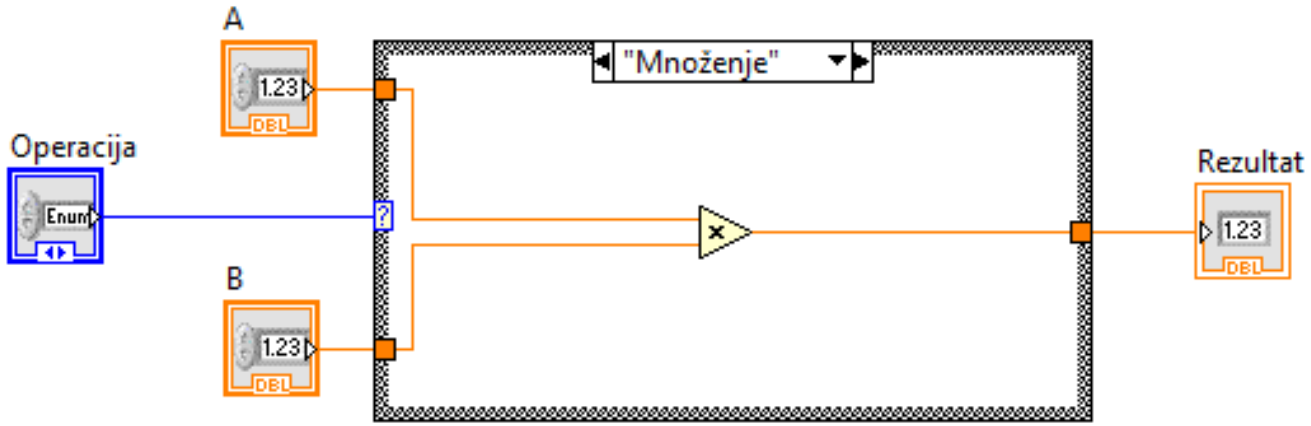
- Članovi enumeracije se mogu dodati opcijom Properties/Edit Items



Enumeracije

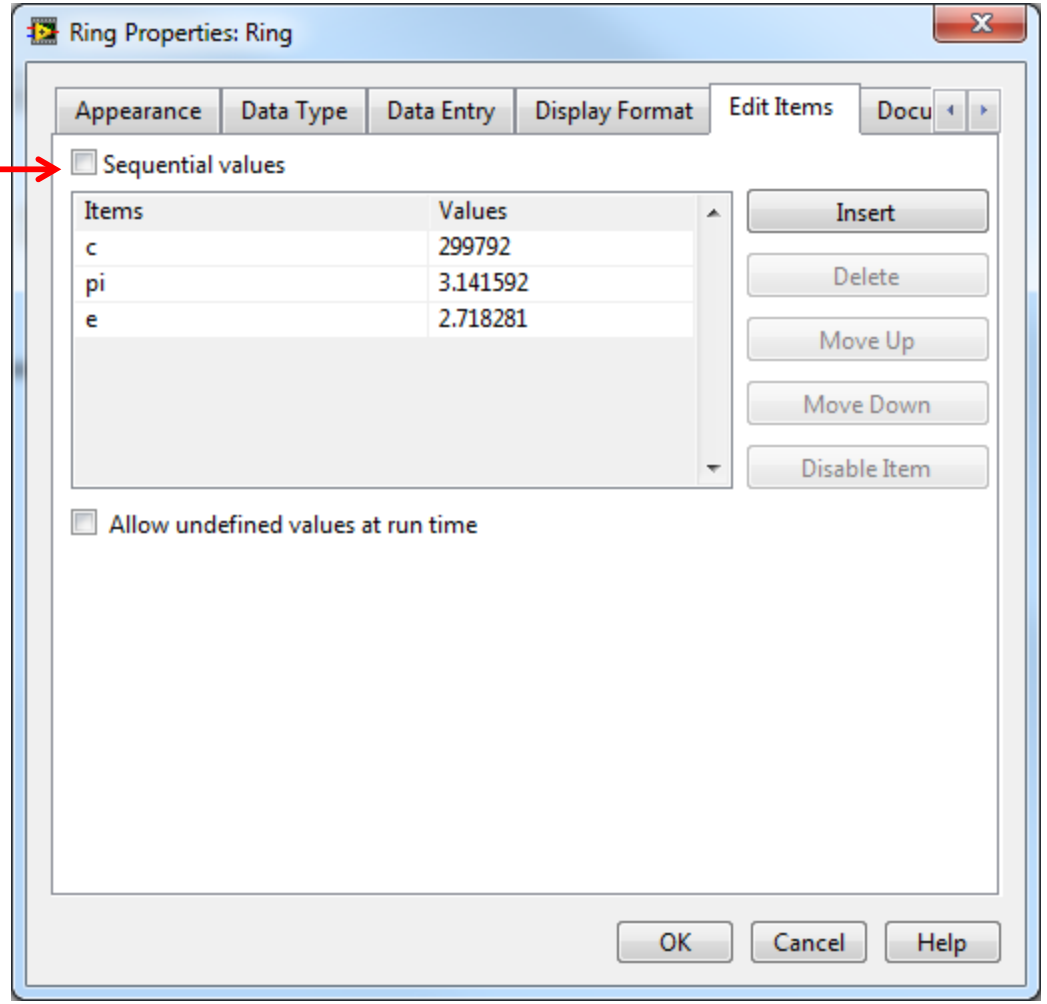
- Enumeracije se mogu povezati za Case strukturu, čime se dobija mogućnost selekcije većeg broja subdijagrama.





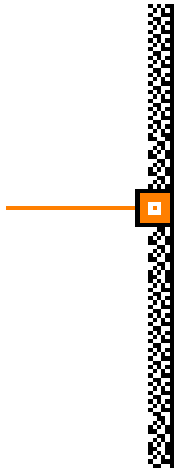
Prstenovi

- Prsten (ring) je sličan enumeraciji, s tom razlikom što je moguće imati različite (nesekvencijalne) vrednosti asocirane članovima
- Prsten može imati necelobrojnu vrednost
- **Ring & Enum >> Text Ring**



Tuneli

- Tuneli kod Case strukture se ponašaju slično kao kod petlji. Pojaviće se automatski ukoliko se dovede veza do ivice strukture.
- Ukoliko je tok podataka od spolja ka strukturi, tunel je ulaznog tipa.
- Ukoliko je tok podataka iz strukture ka spolja, tunel je izlaznog tipa.
- Ukoliko je izlazni tunel vezan za jedan subdijagram strukture, onda mora biti vezan i za ostale subdijagrame.
- Desnim klikom na tunel se može odrediti podrazumevana vrednost opciom **Use Default If Unwired**.

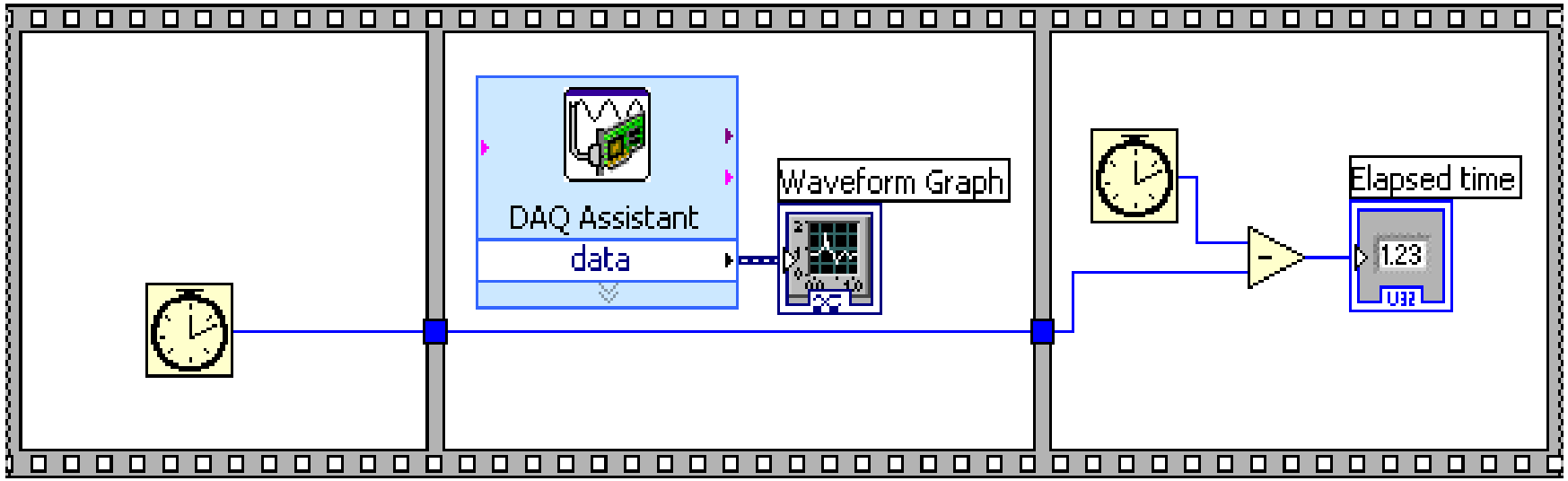


Flat Sequence struktura

- U tekstualnim programskim jezicima redosled dekleracija određuje redosled izvršavanja. U LabVIEW, redosled nije toliko trivijalan.
- **Flat Sequence strukture** se mogu iskoristiti za precizniju kontrolu toka programa. Kôd se može rasporediti u specificirane sekvence i time odrediti redosled izvršavanja.
- Flat Sequence struktura se može postaviti izborom iz **Programming/Structures**. Prilikom postavljanja, struktura ima jedan frejm.

Frejmovi u Sequence strukturi

- Kôd koji se izvršava sekvencijalno je raspoređen po subdijagramima koji se nazivaju **frejmovi**. Cela struktura liči na filmsku traku, gde se frejmovi prikazuju jedan za drugim.

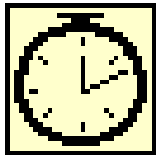


Dodavanje i brisanje frejmova

- Frejmovi se mogu dodavati Sequence strukturi desnim klikom na ivicu strukture i izborom opcije **Add Frame Before/After**.
- Frejm se može izbrisati desnim klikom na ivicu frejma i izborom opcije **Delete This Frame**.

Tick Count fukcija

- Prethodni primer koristi **Tick Count** funkciju, koja vraća broj milisekundi očitan sa internog tajmera.
- Interni tajmer se resetuje prilikom butovanja računara.
- Prethodni primer može poslužiti za određivanje potrebnog vremena za izvršavanje nekog kôda, u konkretnom primeru meri se vreme akvizicije neke veličine.
- Tick Count funkcija se nalazi u paleti **Porgramming/Timing**.

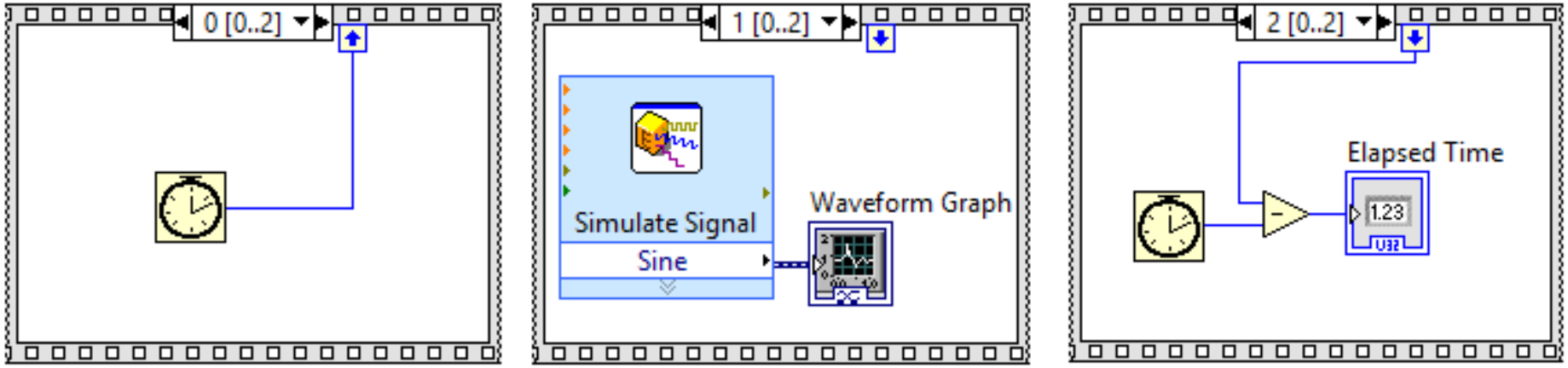


Stacked Sequence struktura

- **Stacked Sequence strukture** se mogu iskoristiti za precizniju kontrolu toka programa. Kôd se može rasporediti u specificirane subdijagrame i time odrediti redosled izvršavanja.
- Stacked Sequence struktura se može postaviti izborom iz **Programming/Structures**. Prilikom postavljanja, struktura ima jedan subdijagram.

Stacked Sequence struktura

- Kôd koji se izvršava sekvencijalno je raspoređen po subdijagramima. Cela struktura liči Case strukturu, pri čemu se izvršavaju svi subdijagrami prema redosledu.



Stacked Sequence struktura

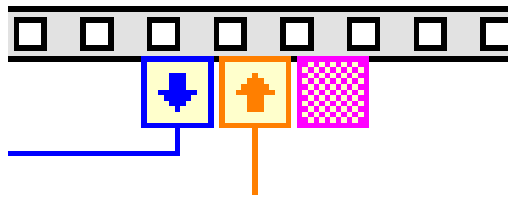
- **Stacked Sequence** struktura ima sličnu funkciju kao i Flat Sequence struktura, pri čemu su subdijagrami postavljeni kao kod Case strukture, jedan iznad drugog.
- Samo je jedan subdijagram vidljiv, može se izabrati iz padajućeg menija sa vrha strukture.
- **Labela subdijagrama** na vrhu strukture ukazuje na vidljiv subdijagram.

Dodavanje i brisanje subdijagrama

- Subdijagrami se mogu dodavati Sequence strukturi desnim klikom na ivicu strukture i izborom opcije **Add Frame Before/After**.
- Subdijagram se može izbrisati desnim klikom na ivicu strukture i izborom opcije **Delete This Frame.**
- Subdijagram se može iskopirati u novi desnim klikom na ivicu strukture i izborom opcije **Duplicate Frame**.

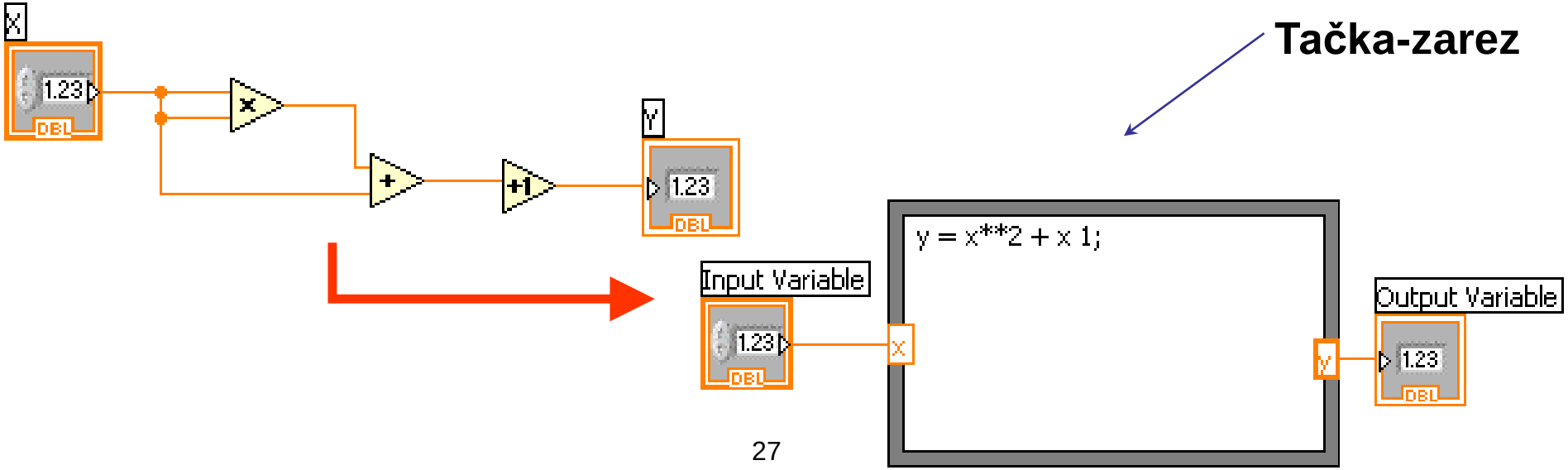
Stacked Sequence struktura - lokali

- Podaci se prolseđuju preko posebnih tunela, **lokala**, čija strelica ukazuje da li podaci ulaze ili izlaze iz subdijagrama.
- Lokal se može dodati desnim klikom na strukturu i izborom opcije **Add Sequence Local**.
- Strelica lokala prikazuje smer toka podataka, i određuje se vezivanjem za ulaz/izlaz funkcija, kontrola ili indikatora unutar subdijagrama.
- Lokal koji je vezan kao izlazni lokal nekog subdijagrama, dostupan je kao ulazni lokal samo subdijagramima koji se kasnije izvršavaju.



Formula node

- Formula Node struktura služi za iplementiranje komplikovanijih aritmetičkih formula, pri čemu se dobija na preglednosti.
- Struktura se nalazi u paleti Programming/Structures.
- Promenljive se kreiraju na ivicama strukture (x, y). Promenljive su case-sensitive.
- Svaka dekleracija se završava tačka-zarezom (;)
- U Help Window se mogu videti sve raspoložive funkcije.



Pregled

- Case strukture služe za izbor kôda koji se izvršava. Imaju funkciju sličnu IF ili CASE dekleracijama u tekstualnim programskim jezicima.
- Selektorksa promenljiva Case strukture može biti logička, numerička (celobrojna) ili string.
- Flat i Stacked Sequence strukture služe za određivanje redosleda izvršavanja kôda.
- Timed funkcija se koristi za očitavanje internog hronometra (u ms), čime je moguće precizno odrediti vreme izvršavanja kôda.
- Formula Node struktura služi preglednijoj formulaciji aritmetičkih operacija.

Vežba 7 – Jednostavan kalkulator

Jednostavan kalkulator

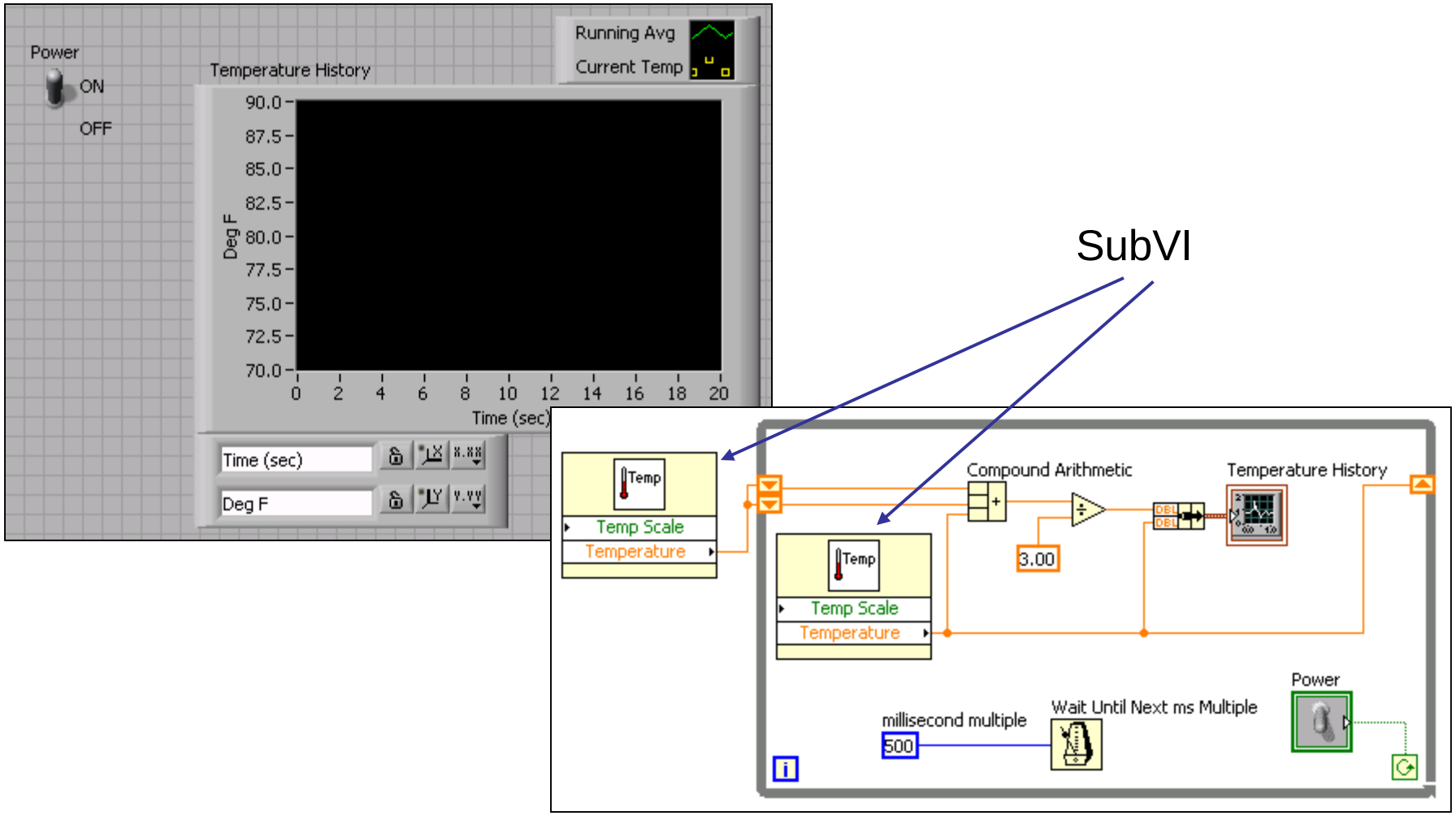
- Realizujte VI koji ima funkciju kalkulatora
- Kalkulator ima funkcije sabiranja, oduzimanja, množenja, deljenja i kvadratnog korena.
- Operandi A i B se unose u odgovarajuće kontrole na front panelu. Ukoliko je operacija unarna, uzima se operand A.
- VI se izvršava kontinualno, može se prekinuti klikom na dugme **Stop**
- Operacija se bira kontrolom **Operacija**
- Izračunavanje se obavlja nakon klika na dugme **Izračunaj**

Modularno programiranje

Modularno programiranje

- SubVI
- Ikone i konektorski panel
- Upotreba SubVI
- Kreiranje SubVI iz postojećeg kôda

Hijerarhija u LabVIEW



SubVI

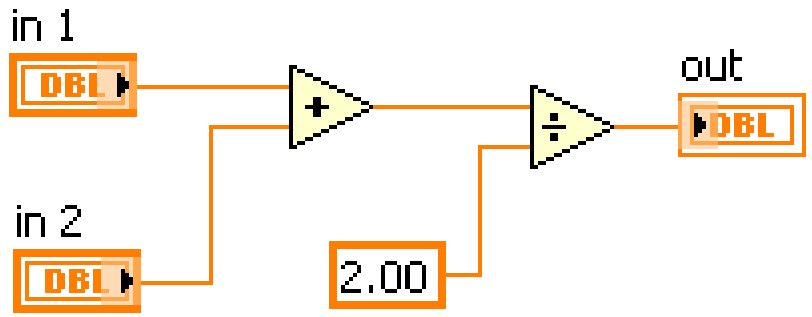
Pseudo-kôd funkcije

```
function average (in1,
  in2, out)
{
out = (in1 + in2)/2.0;
}
```

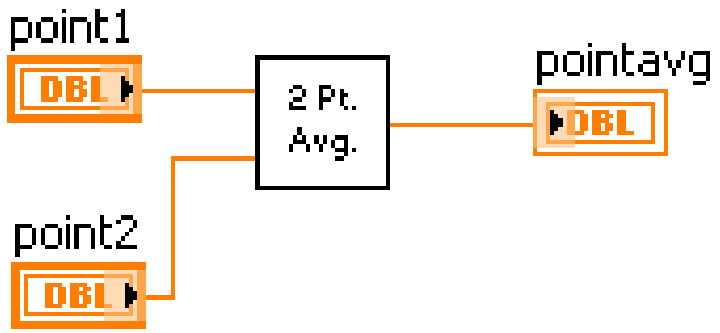
Poziv funkcije

```
main
{
average (point1, point2,
  pointavg)
}
```

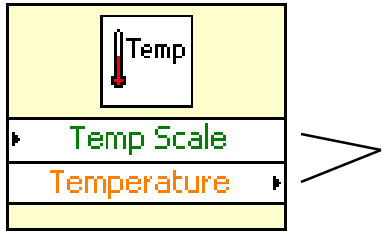
SubVI blok dijagram



Poziv SubVI



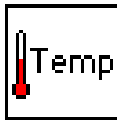
Ikona i konektor



terminali

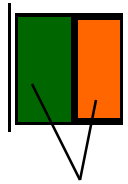
Ikona predstavlja SubVI u blok dijagramima drugih virtuelnih instrumenata

Ikona



Konektor je skup terminala preko kojih se prihvataju ka i prosleđuju podaci od SubVI

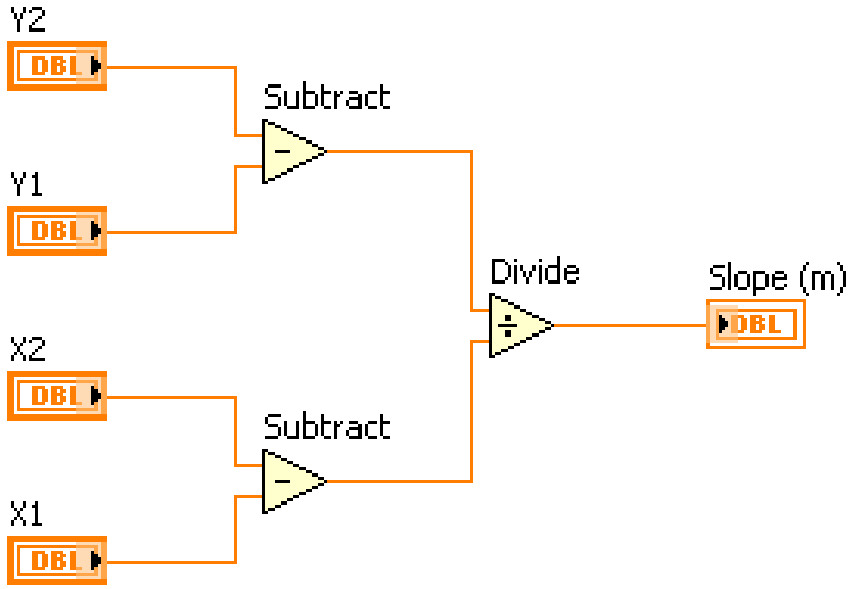
Konektor



terminali

Primer SubVI – izračunavanje strmine

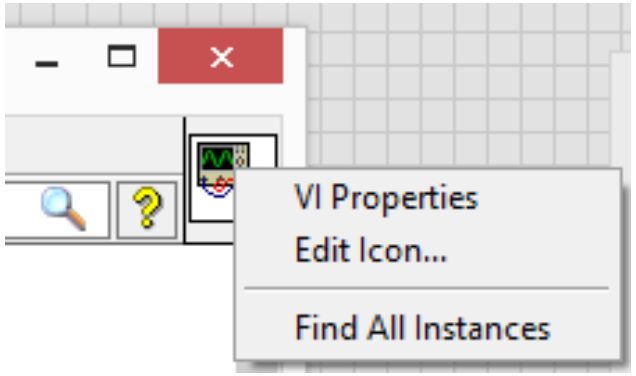
- Virtuelni instrument unutar drugog virtuelnog instrumenta je subVI
- Da bi se neki VI koristio kao subVI, neophodno je kreirati ikonu i konektor nakon kreiranja front panela i blok dijagrama



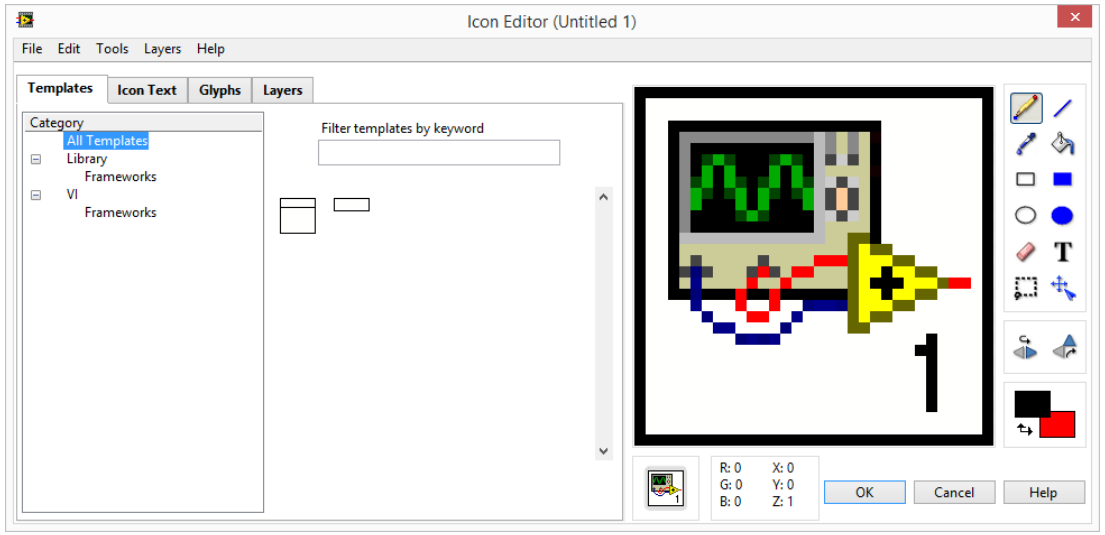
Kreiranje ikone

- Ikona je grafička reprezentacija VI
- Desni klik na Icon Pane (f. panel ili b. dijagram)

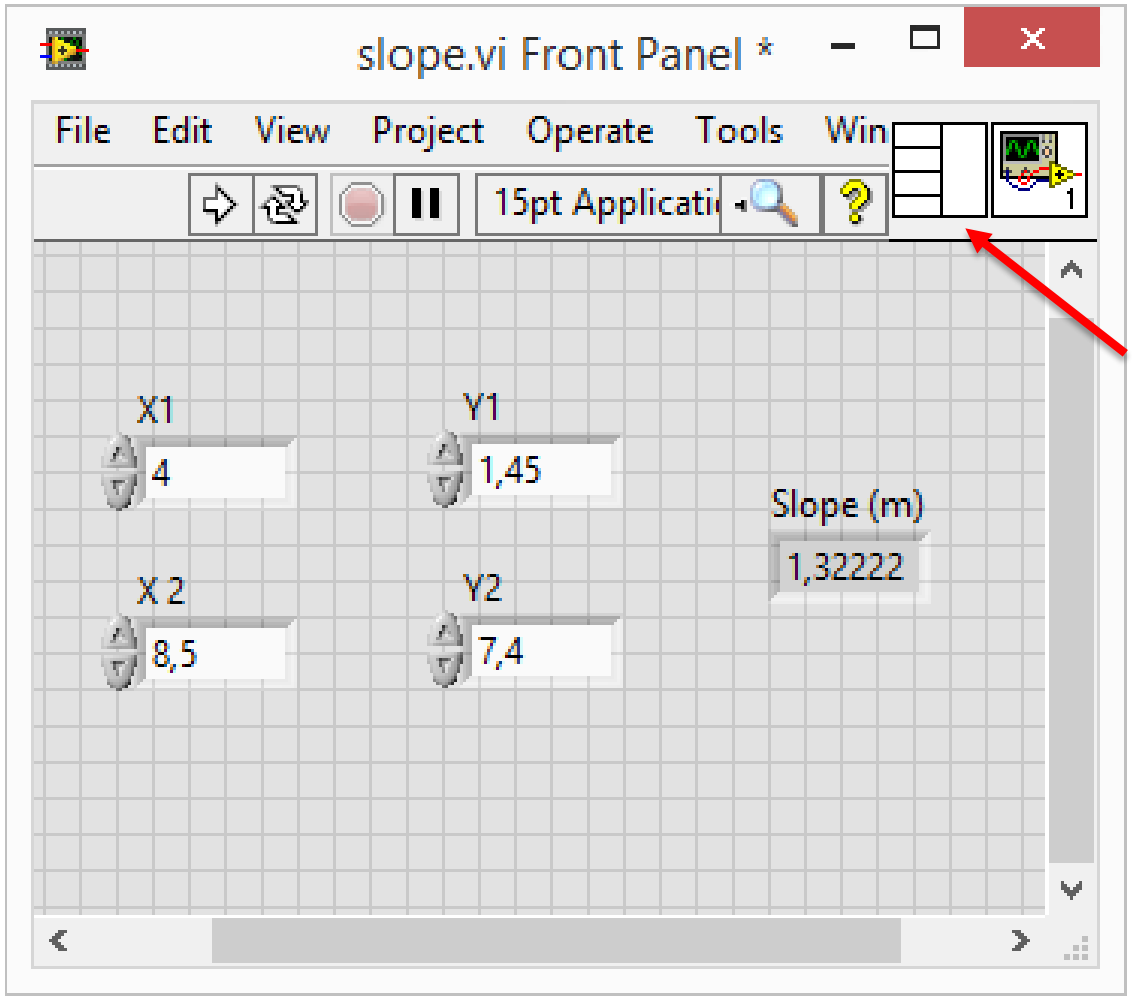
Osnovna ikona



Kreirajte ikonu

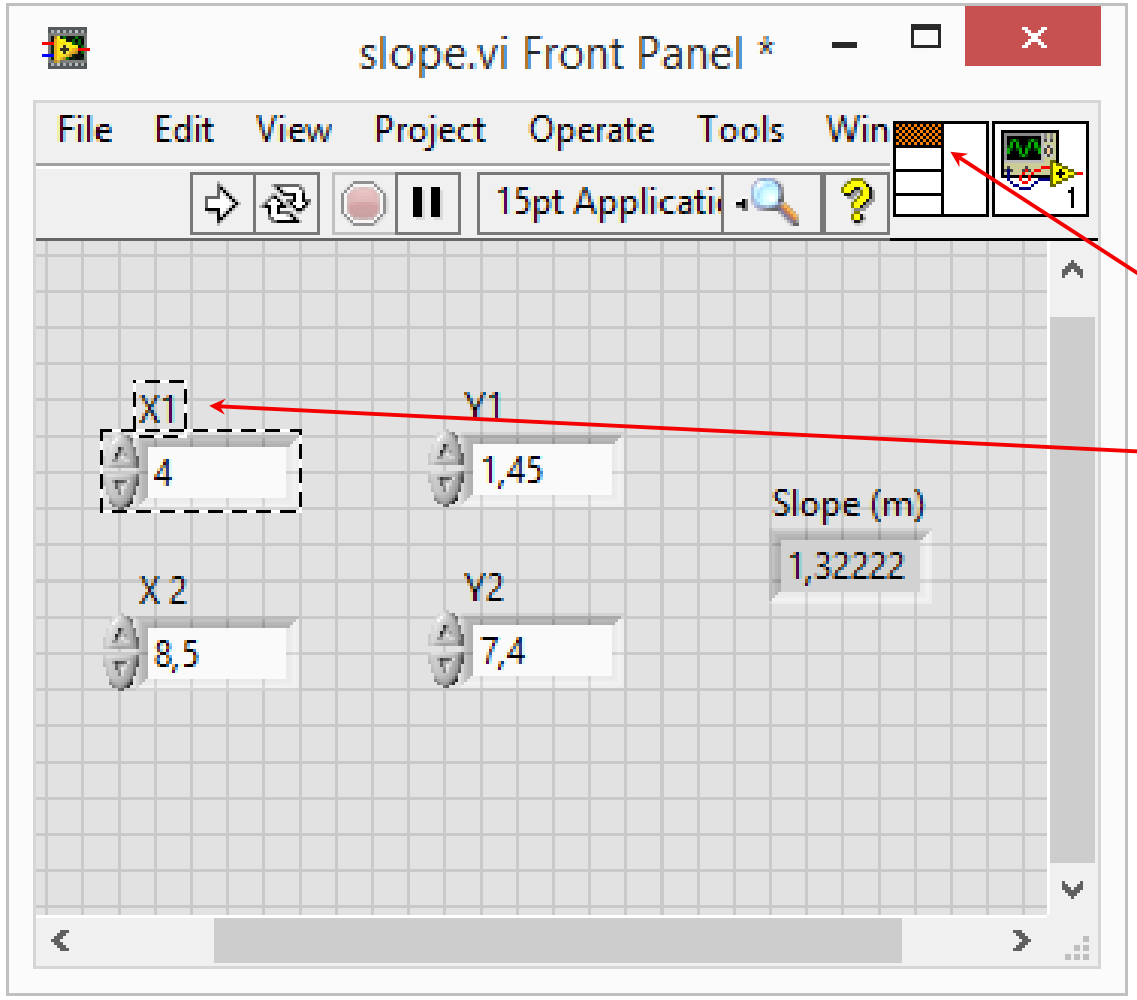


Povezivanje konektora



Konektorski panel
(samo na front panelu)

Kreiranje konektora

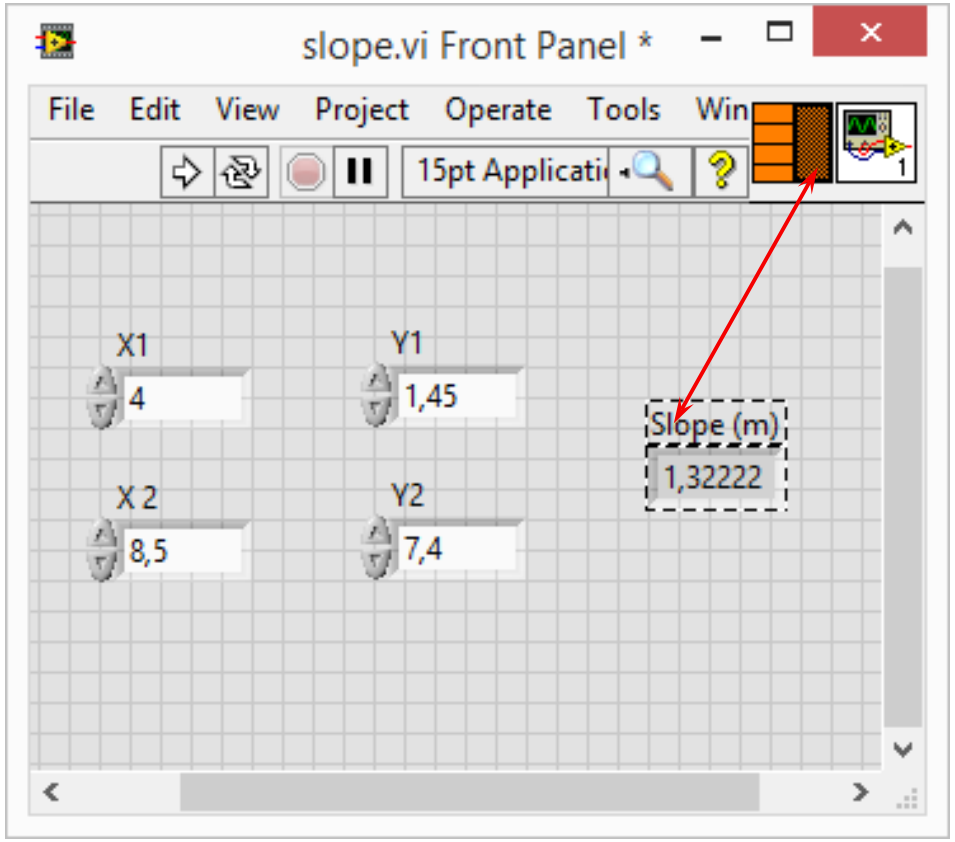


Povezati konektor i kontrolu/ indikator

Kreiranje konektora

Boje na terminalu konektora odgovaraju boji tipa kontrole/indikatora za koji su povezani

Klikom na terminal konektora selektuje se kontrola/indikator koji je za njega vezan

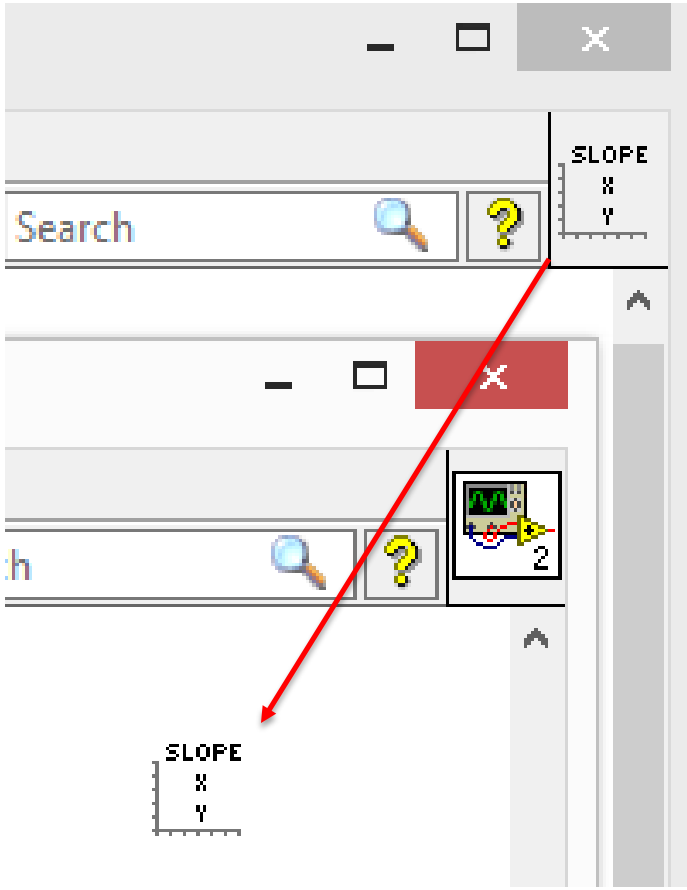


Umetanje subVI u virtuelni instrument

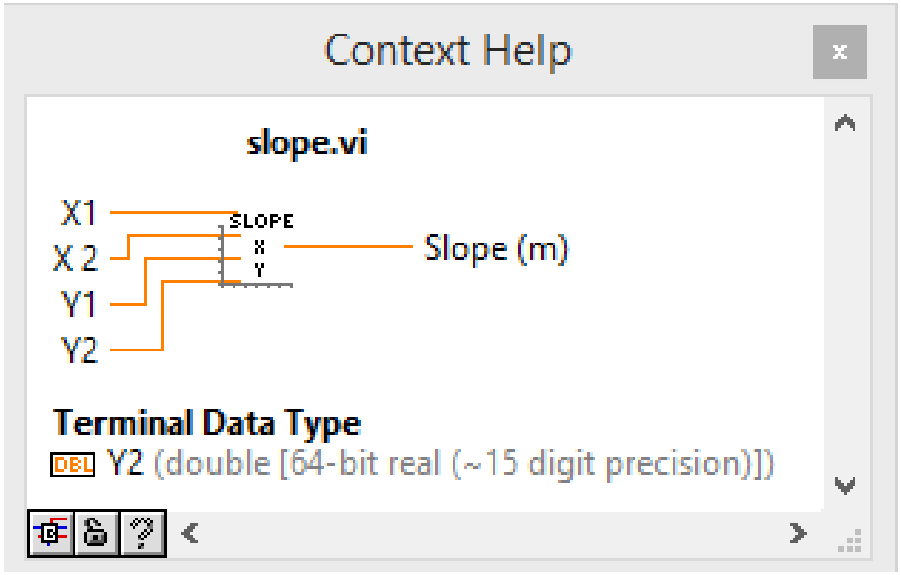
Izborom
All Functions/Select a VI…

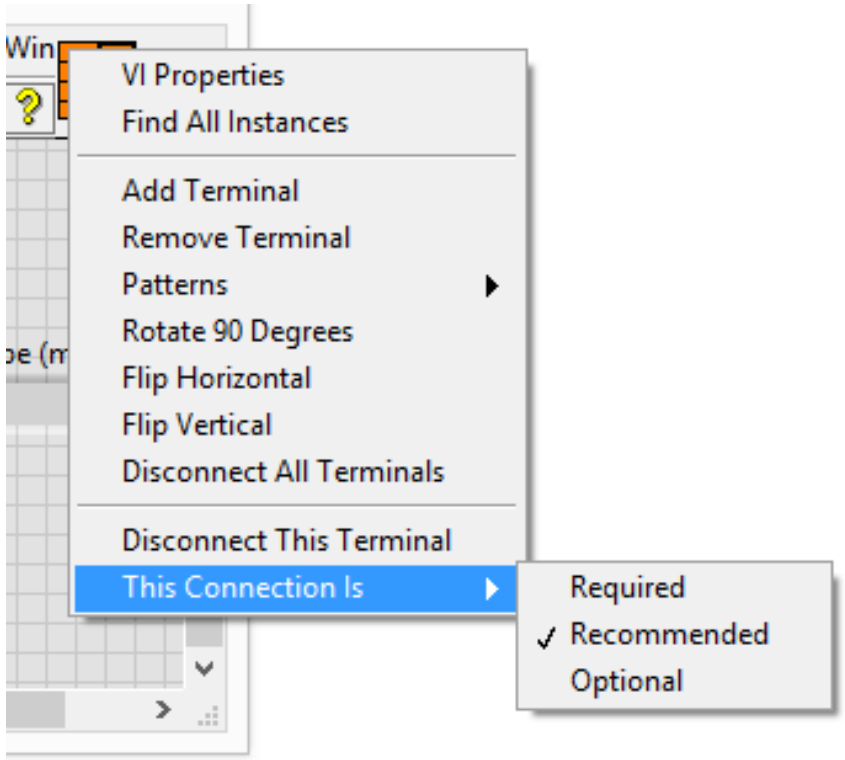
ili

Prevlačenjem ikone na blok dijagram



Help i klasifikovanje terminala



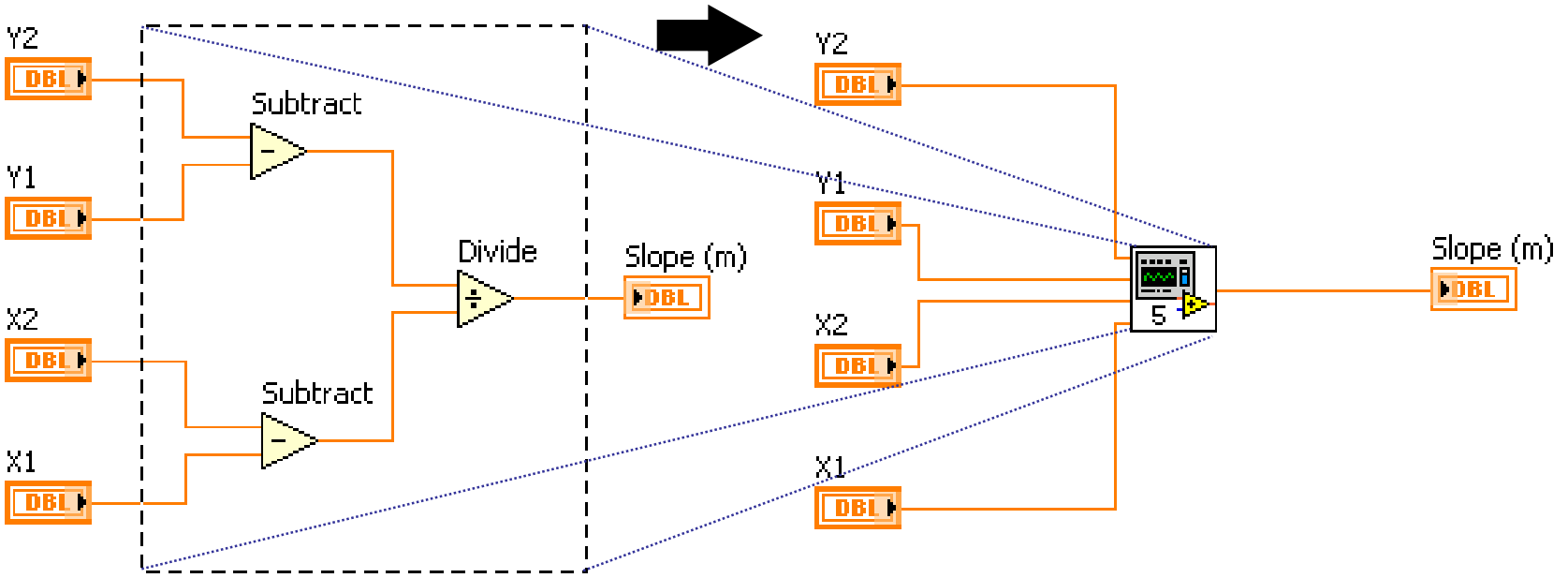


Ulazi i izlazi se mogu klasifikovati

- Required – obavezni
- Recommended – preporučeni
- Optional – opcioni

Opcija Create SubVI

- Uokviriti deo kôda koji želi da se konvertuje u subVI
- Izaberite opciju **Create SubVI** iz **Edit** menija



Pregled

- Virtuelni instrumenti mogu da se koriste kao subVI kada se definiše konektor i ikona
- Ikona se kreira pomoću alata Icon Editor
- Konektor se definiše izborom ulaznih i izlaznih terminala
- SubVIs se učitava opcijom Select a VI u All Functions paleti ili drag & drop tehnikom, prevlačenjem ikone na blok dijagram
- Svaki subVIs je podržan Context Help mehanizmom
- Opcija Edit/Create SubVI omogućava lako kreiranje subVI i modularnost blok dijagrama

Vežba 8 – SubVI

Zadatak - SubVI

- Pomoću realizovanih virtuelnih instrumenata za skalarni i vektorski proizvod vektora, realizovati VI koji izračunava vektor E, na osnovu formule:

$$E = A \times B + C \times D + (A \cdot C)(B \times D) + (B \cdot D)(A \times C)$$

- Vektori A, B, C, D se zadaju kontrolama na front panelu. Vektor E se prikazuje na indikatoru.