

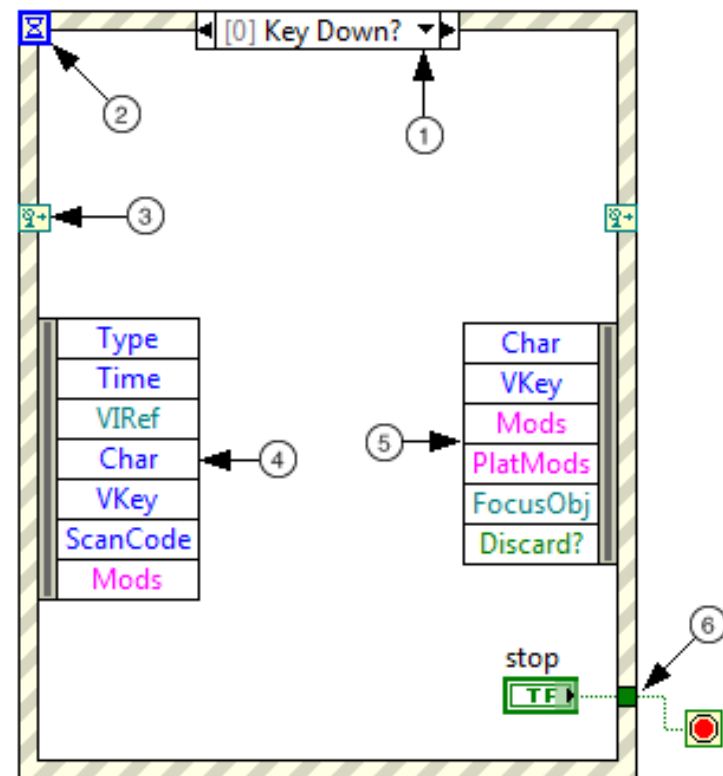
**Automati stanja aktivirani
dogadajima
(Event-driven)**

Pojam

- Klasični automati stanja su stalno u pripravnosti (polling) – nije optimalno iskorišćenje resursa
- Event-driven – red izvršavanja zavisi od događaja
- Događaji:
 - Korisnički (User Events)
 - Vremenski (Time Events)
 - Podaci (Data Events)
- Prelazi iz stanja u stanje su fleksibilni i zavise od događaja
- Događaji aktiviraju scenarije izvršavanja (nema “praznog hoda”)
- Mnogo efikasniji od klasičnih, pogotovo za veliki broj elemenata na prednjem panelu

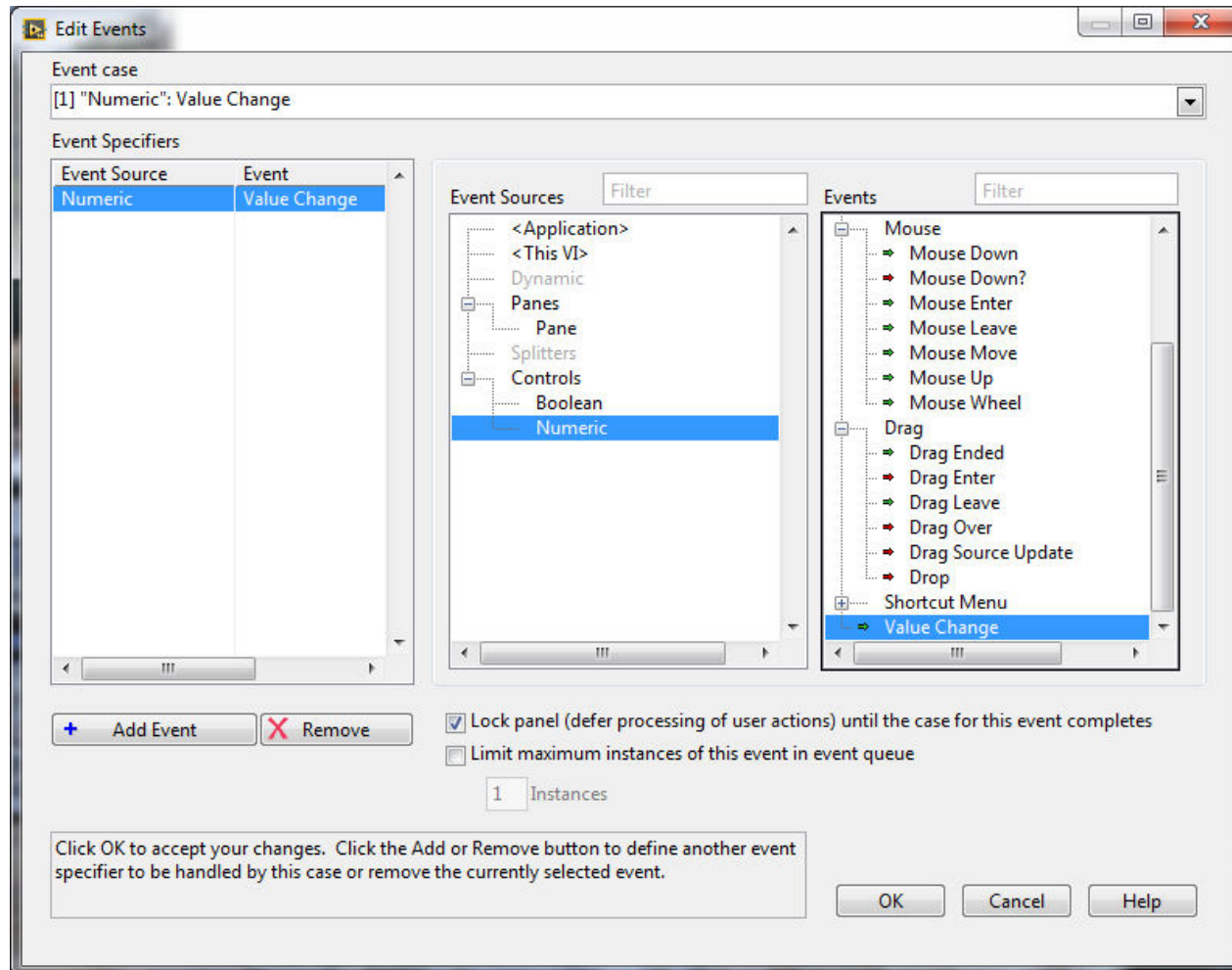
Event struktura

- Odgovara na dva tipa događaja:
 - Notify events – obrađuju se tek nakon što se izvršila promena na prednjem panelu
 - Filter events – mogu da “preduprede” promene na prednjem panelu
- Elementi:
 1. Event selector
 2. Timeout terminal
 3. Terminal dinamičkih događaja
 4. Čvor podataka događaja
 5. Čvor filtera događaja
 6. Izlazni tunel



Event struktura

- Konfiguracija



Event struktura

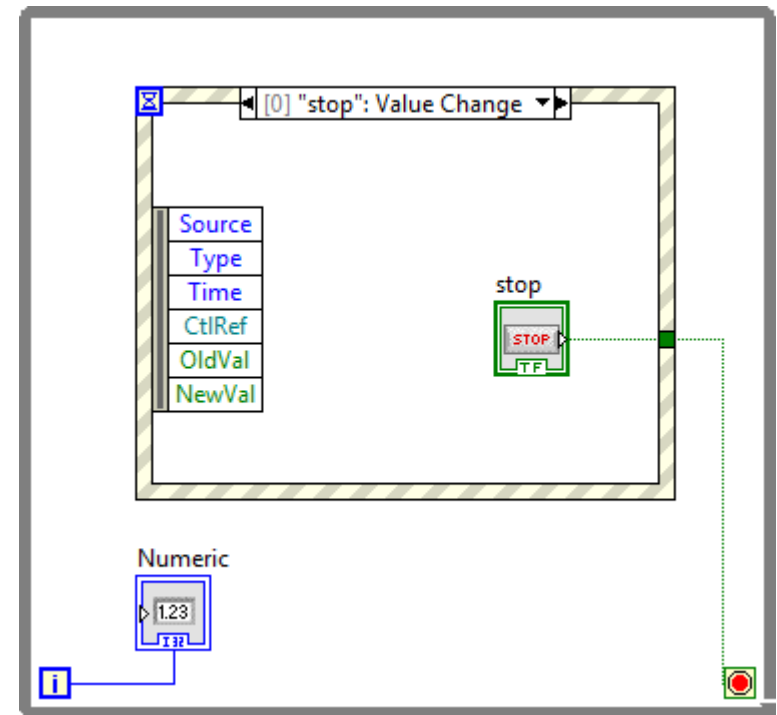
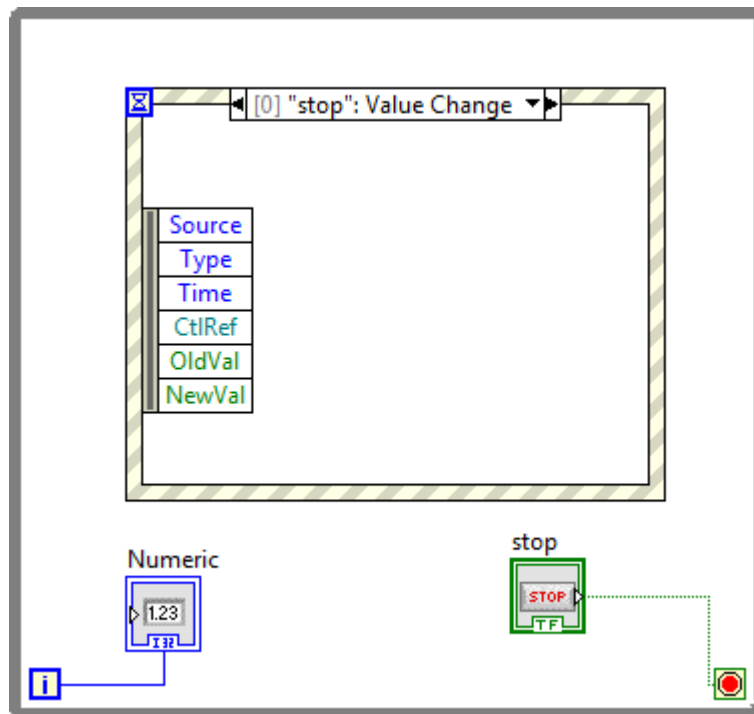
Obratiti pažnju!

- Event strukture imaju “redove događaja”
- Ukoliko struktura ne može trenutno da odgovori na događaj, on se smešta u red i aktivira čim to *data flow* omogući; Ovaj red nije dostupan programeru!
- Uglavnom unutar petlji, kao i Case strukture – najbolje po jedna struktura u petlji
- Zaključavanje prednjeg panela – onemogućeno izvršavanje dok se ne izvrši kod vezan za događaj
 - Nakon izvršavanja, sve je ponovo dostupno, a događaji u međuvremenu stavljeni su u red- VAŽNO!
 - Može doći do “zamrzavanja” ekrana
 - Rešenje – omogućiti događaje samo onda kada struktura čeka na njih (korišćenjem Property node-a Disable)

Event struktura

Obratiti pažnju!

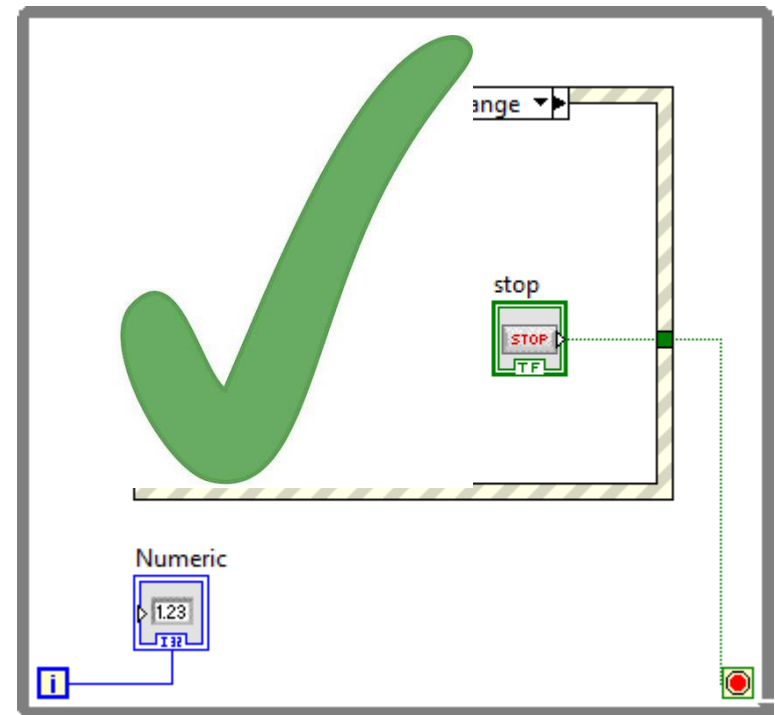
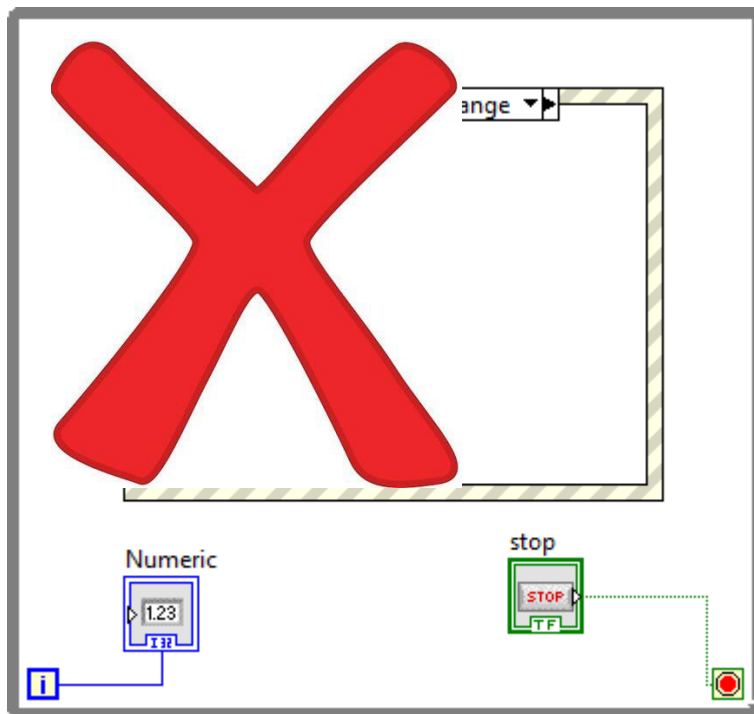
- Bool kontrole (latch) se ne resetuju dok se ne “pročitaju”
- Treba ih postaviti u strukturu koja ih koristi (ako je moguće)



Event struktura

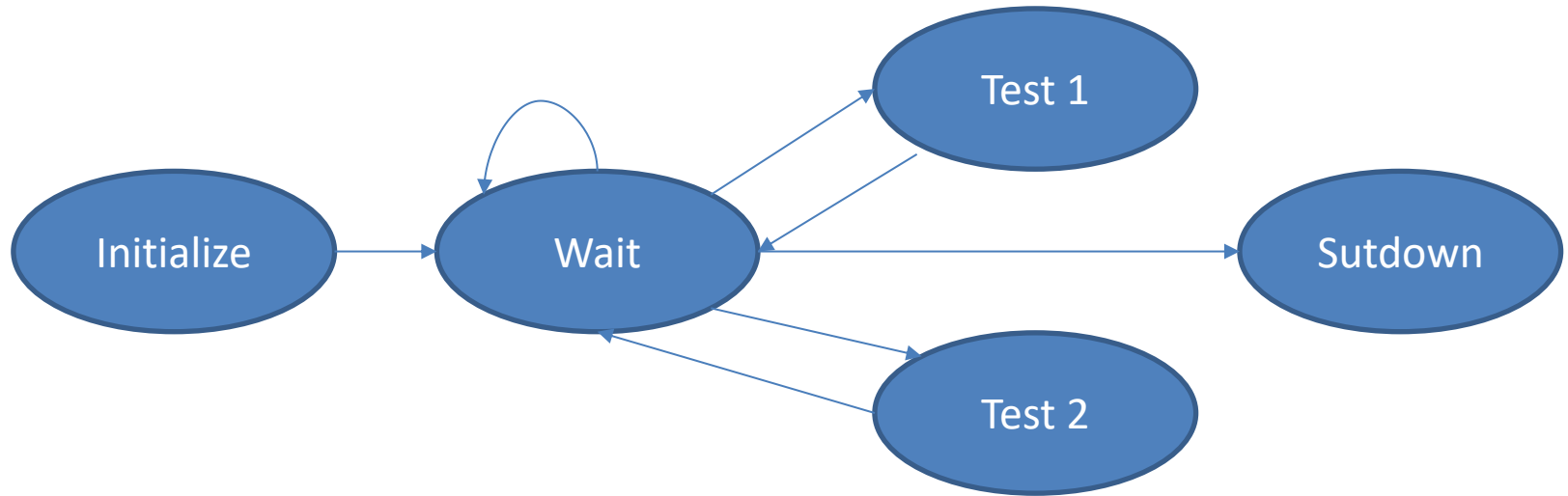
Obratiti pažnju!

- Bool kontrole (latch) se ne resetuju dok se ne “pročitaju”
- Treba ih postaviti u strukturu koja ih koristi (ako je moguće)



Primer

Test 1 i Test 2



Run Test 1

Run Test 2

STOP

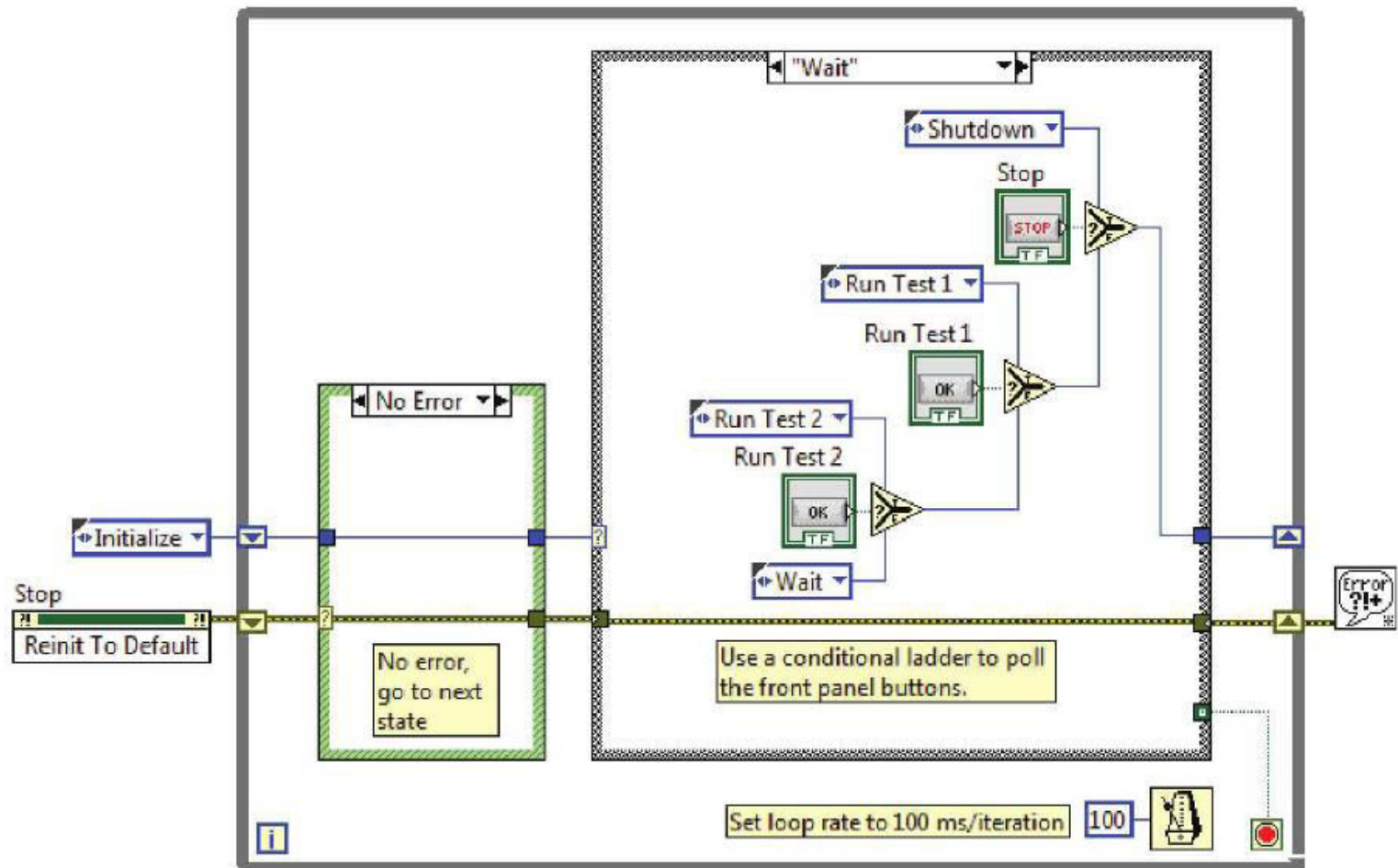
error out

status	code
	0

source

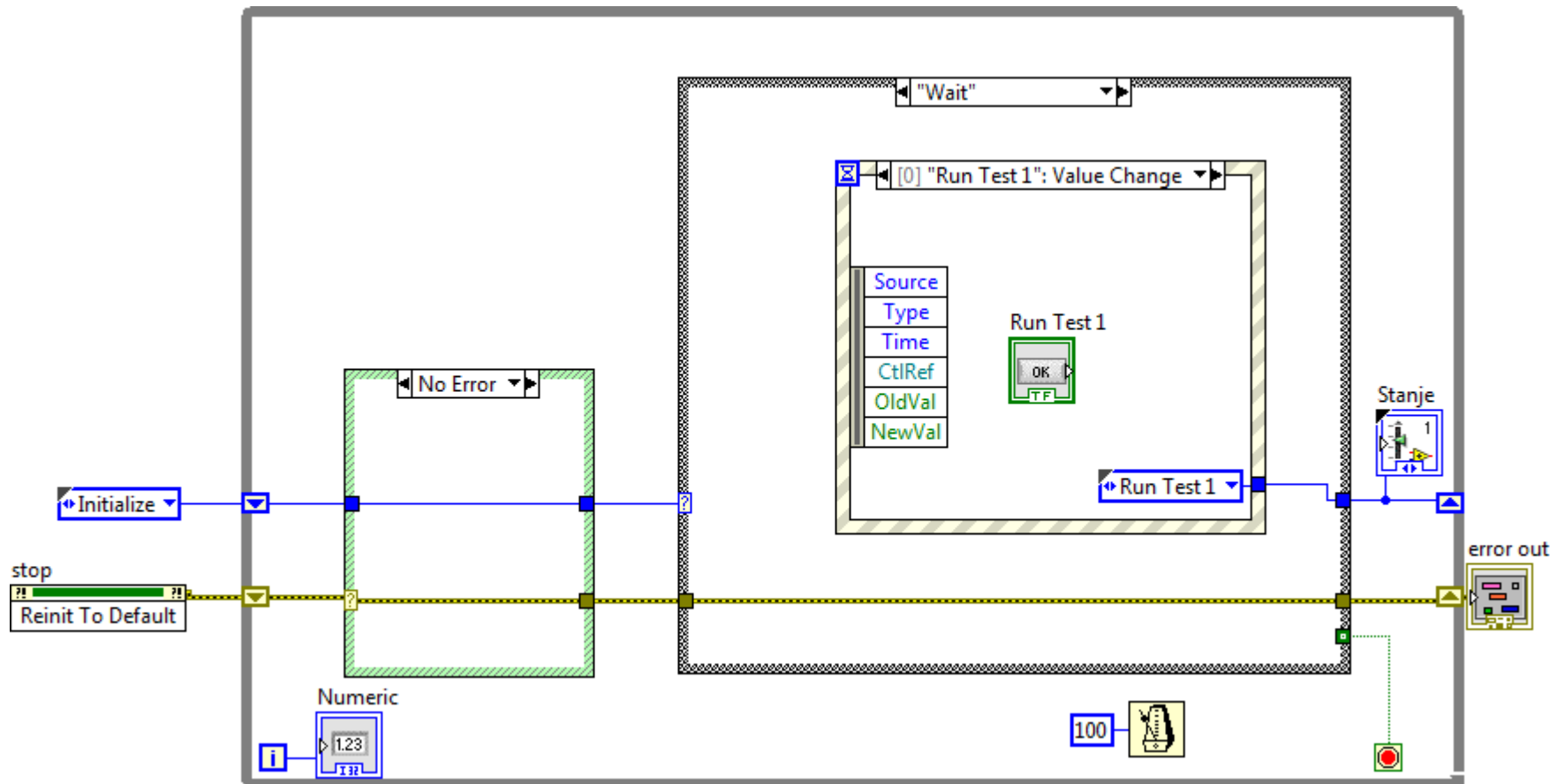
Primer

Tets 1 i Test 2 - klasični automat – stanje Wait



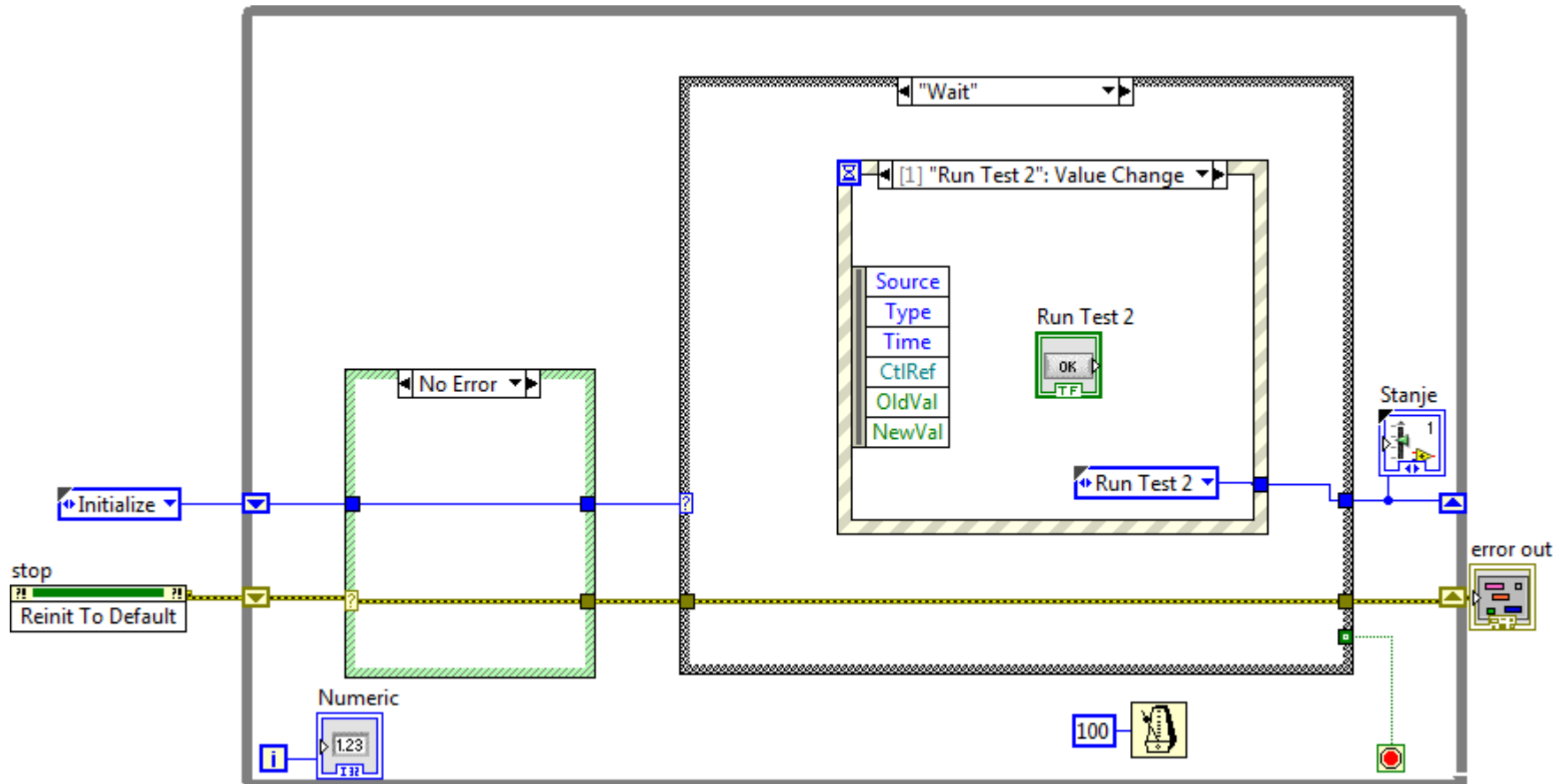
Primer

Test 1 i Test 2 – event based automat



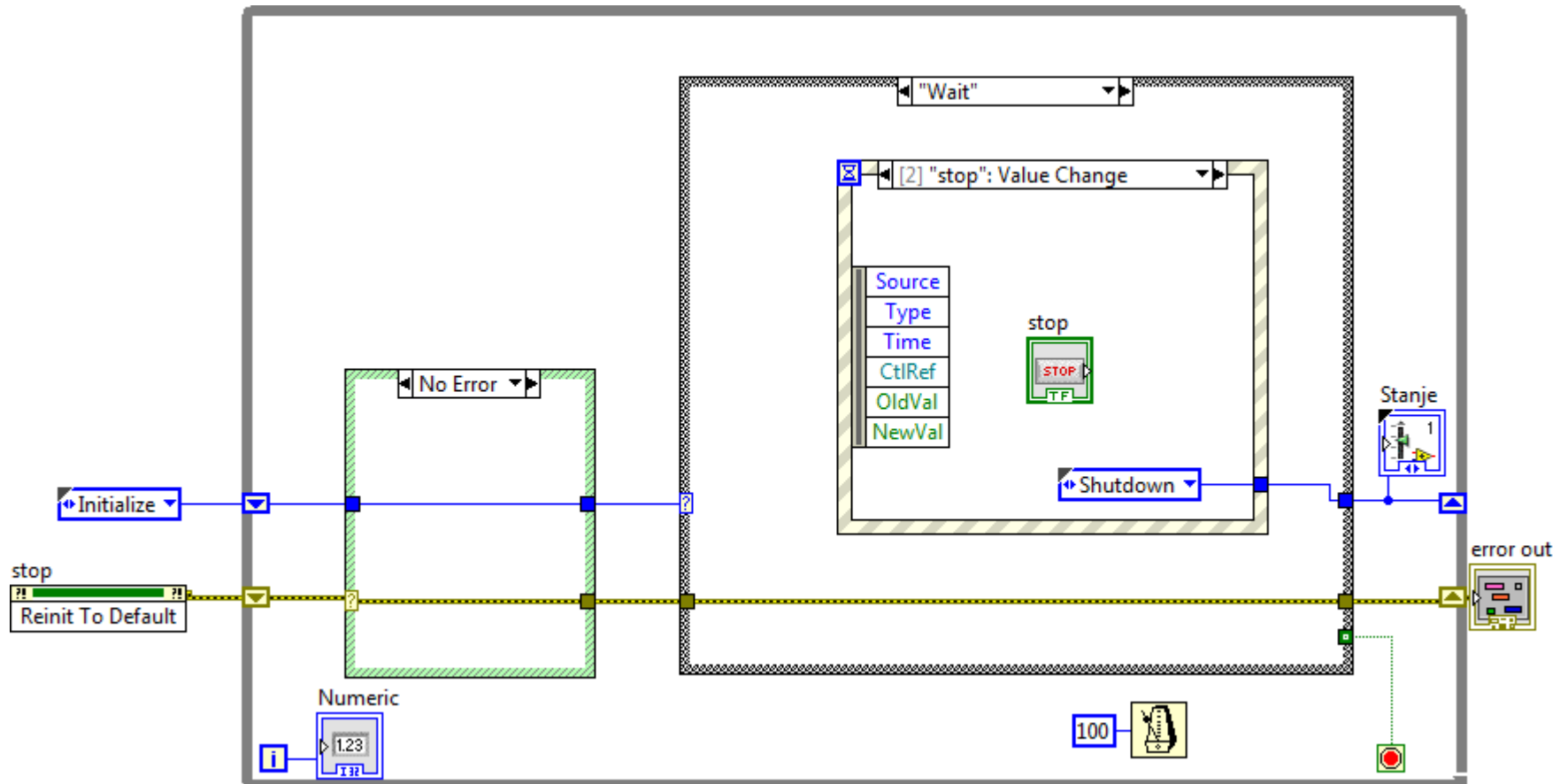
Primer

Test 1 i Test 2 – event based automat



Primer

Test 1 i Test 2 – event based automat

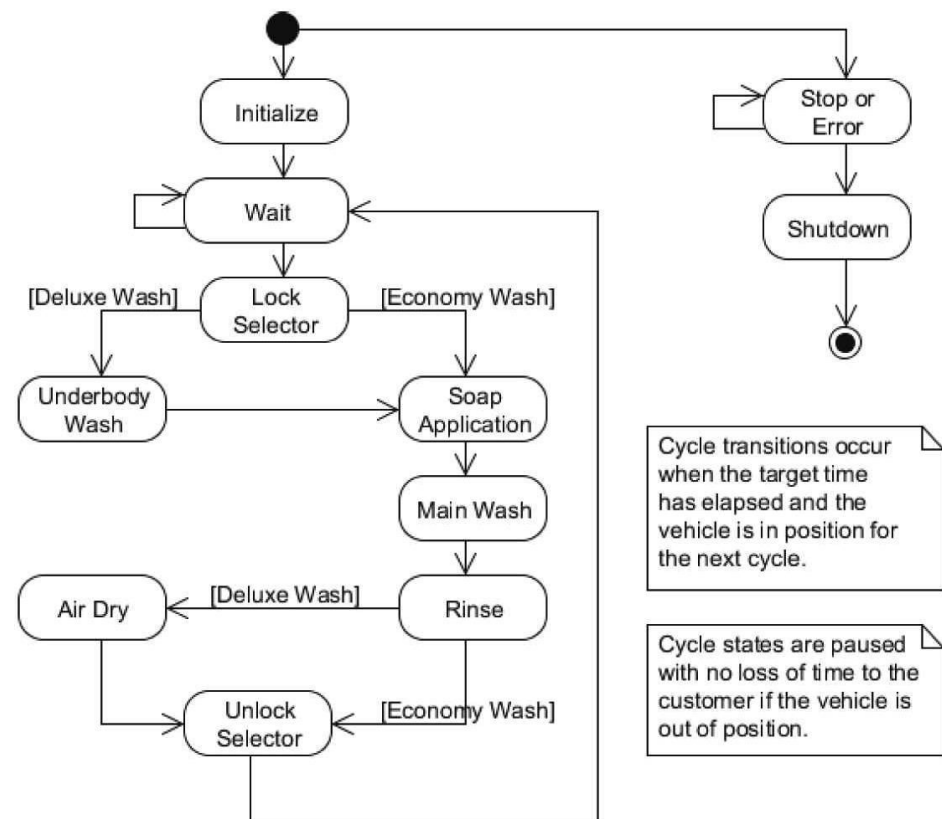


Event-driven automat stanja sa redovima

- Automat stanja sa događajima eliminiše čekanje (polling)
- I dalje svako stanje određuje sledeće
- Nekada više stanja mora deliti istu informaciju (sa data highway-a, npr.)

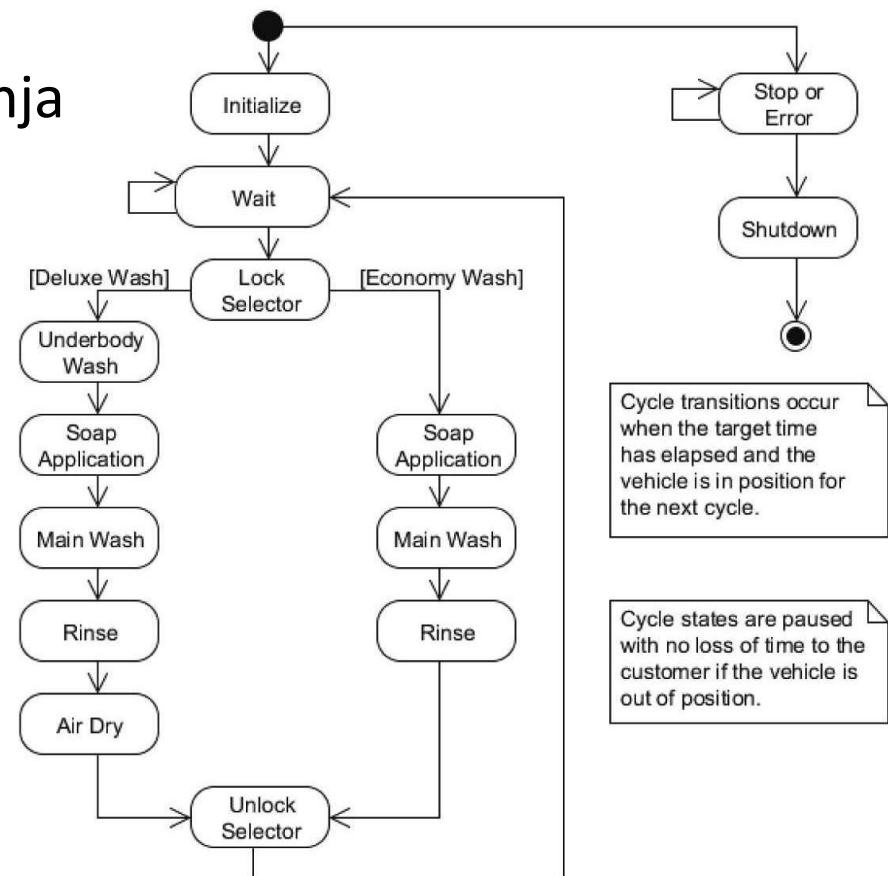
- Primer

- Grananje zavisi od opcije pranja (Deluxe ili Economy)
- Dva mesta grananja



Event-driven automat stanja sa redovima

- Isti problem, sa drugačijim pristupom
- Postoje dve moguće sekvence stanja
- Sekvenca zavisi od stanja
Lock Selector, tj. od opcije pranja
- Stanja se mogu grupisati
u liste – redove (ques)
- Liste će upravljati
automatom stanja
- Nema odlučivanja u
svakom stanju
- Preglednije i pogodnije
za modifikacije



Redovi

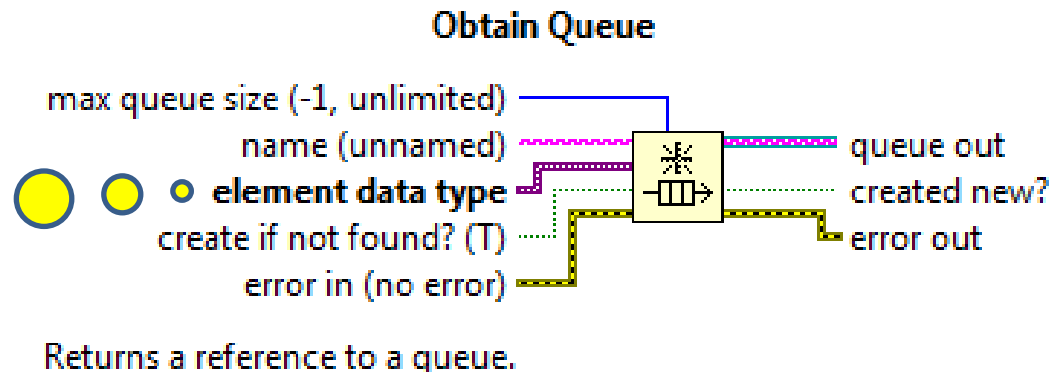
- Red u LabView-u predstavlja uređeni bafer namenjen za smeštanje podataka i čuvanje za kasniju upotrebu
- First in – first out (FIFO) princip
- Posebna vrsta niza
 - Podaci istog tipa
 - Elementi mogu biti dodavani samo na kraju reda
 - Elementi se mogu preuzimati samo na kraju reda
 - Nizovi – random access
 - Redovi – sequential access

Funkcije za rad sa redovima

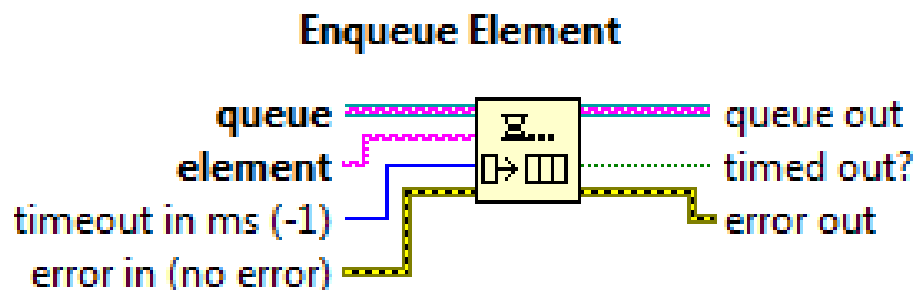
Programming... Synchronization... Queue Operation

- Obtain queue

Element nije dodat u red, samo definiše tip podataka!

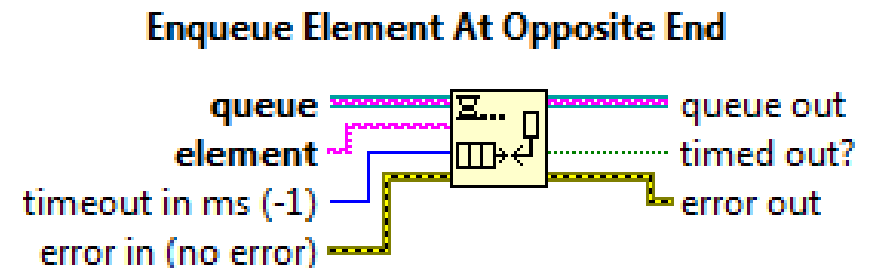


- Enqueue element



Adds an element to the back of a queue.

- Enqueue element at opposite end

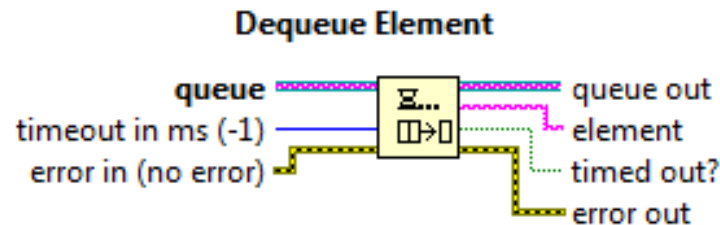


Adds an element to the front of a queue.

Funkcije za rad sa redovima

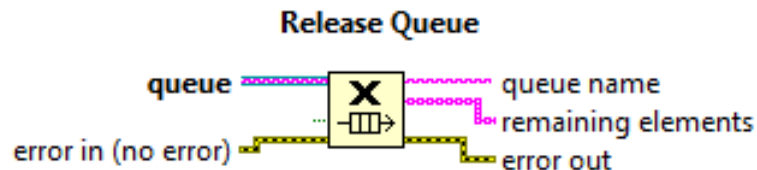
Programming... Synchronization... Queue Operation

- Dequeue element



Removes an element from the front of a queue and returns the element.

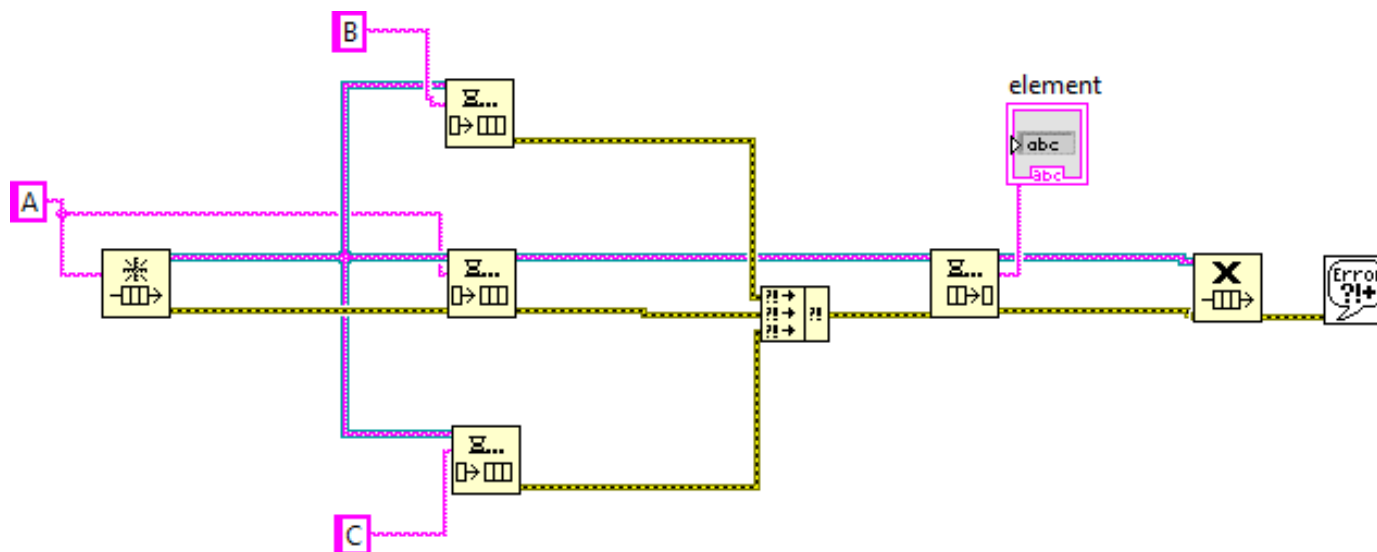
- Release Queue



Releases a reference to a queue.

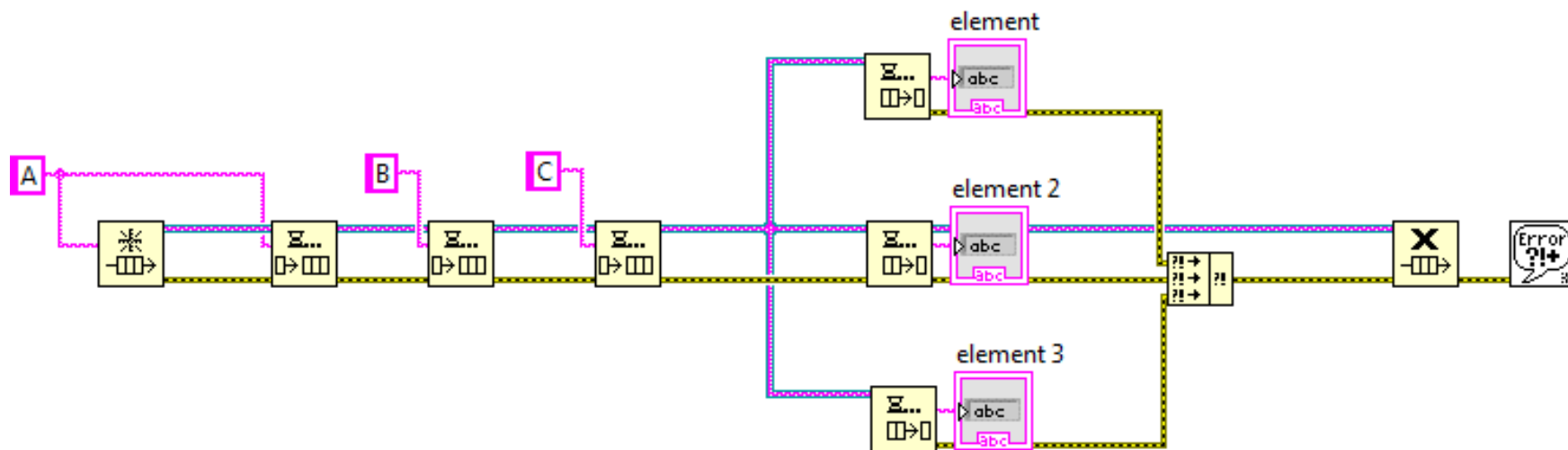
Prenos informacija pomoću redova

- By-reference umesto By-value (kod podataka)
- Žicom se prenosi samo referenca na red, a ne sam red
- Primer – plaćanje u gotovini i pomoću čekova
- Zahvalni za rad sa jednom petljom
- Kod više paralelnih petlji može biti problema
 - Upis elemenata u isti niz na više različitih mesta u isto vreme

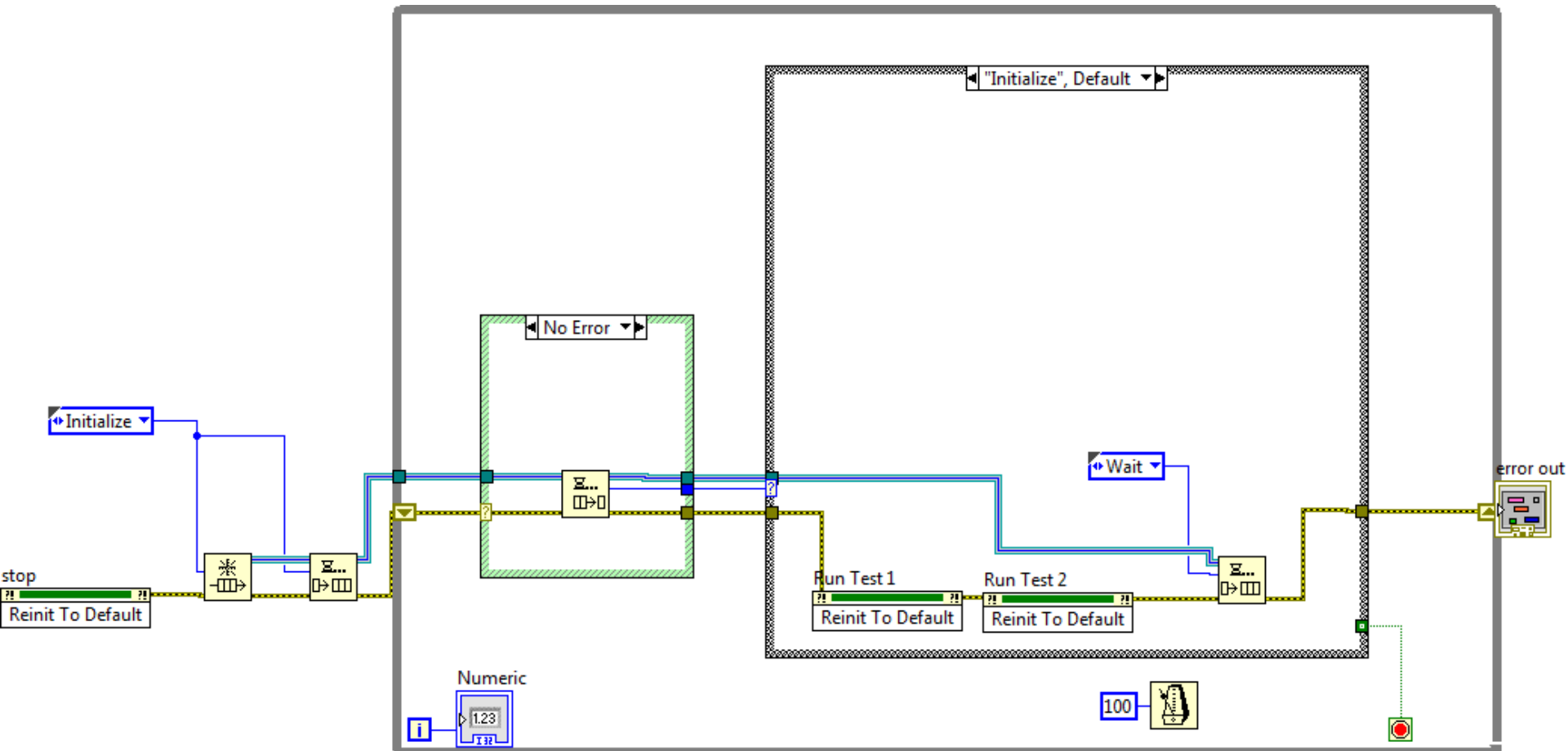


Prenos informacija pomoću redova

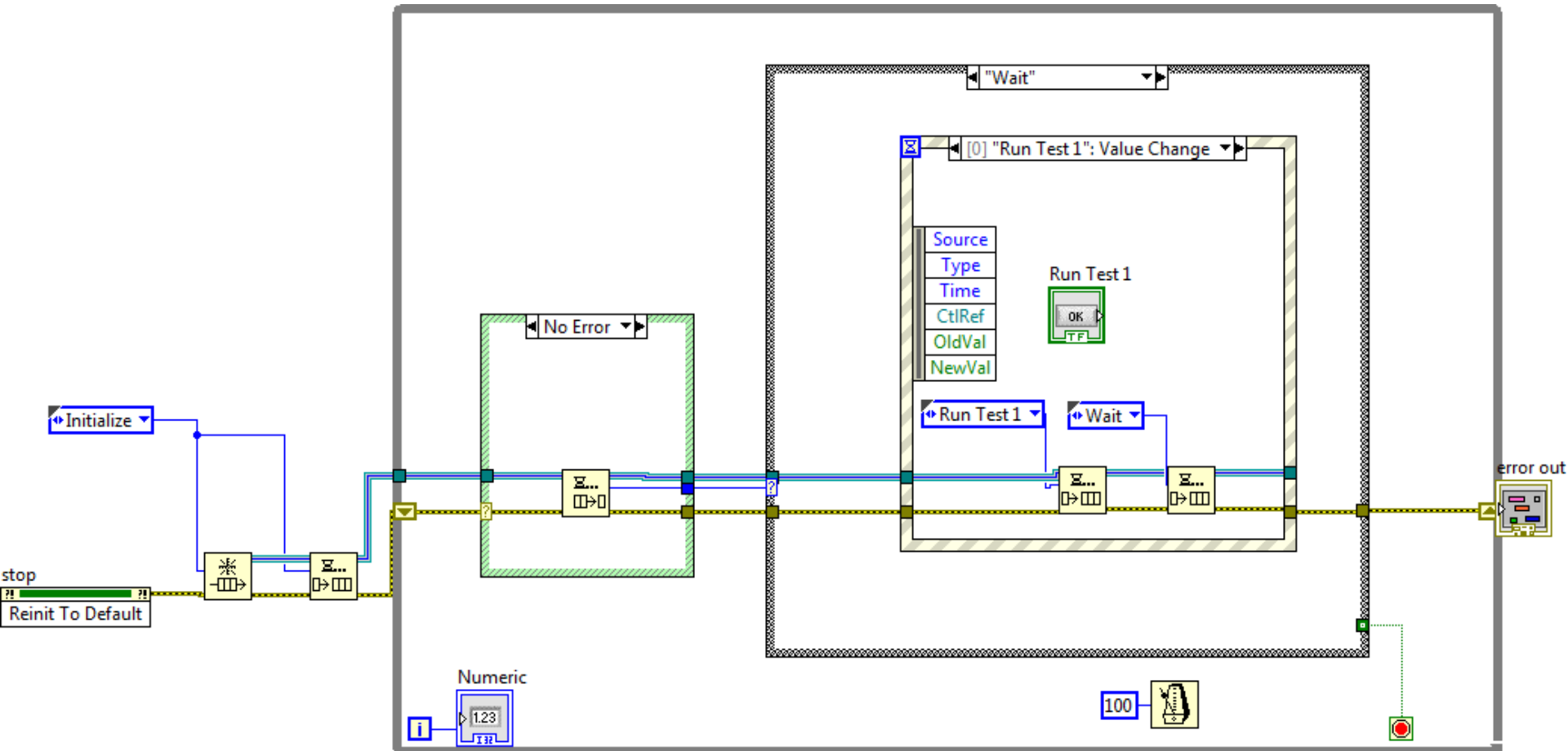
- By-reference umesto By-value (kod podataka)
- Žicom se prenosi samo referenca na red, a ne sam red
- Primer – plaćanje u gotovini i pomoću čekova
- Zahvalni za rad sa jednom petljom
- Kod više paralelnih petlji može biti problema
 - Čitanje elemenata iz istog niza na više različitih mesta u isto vreme



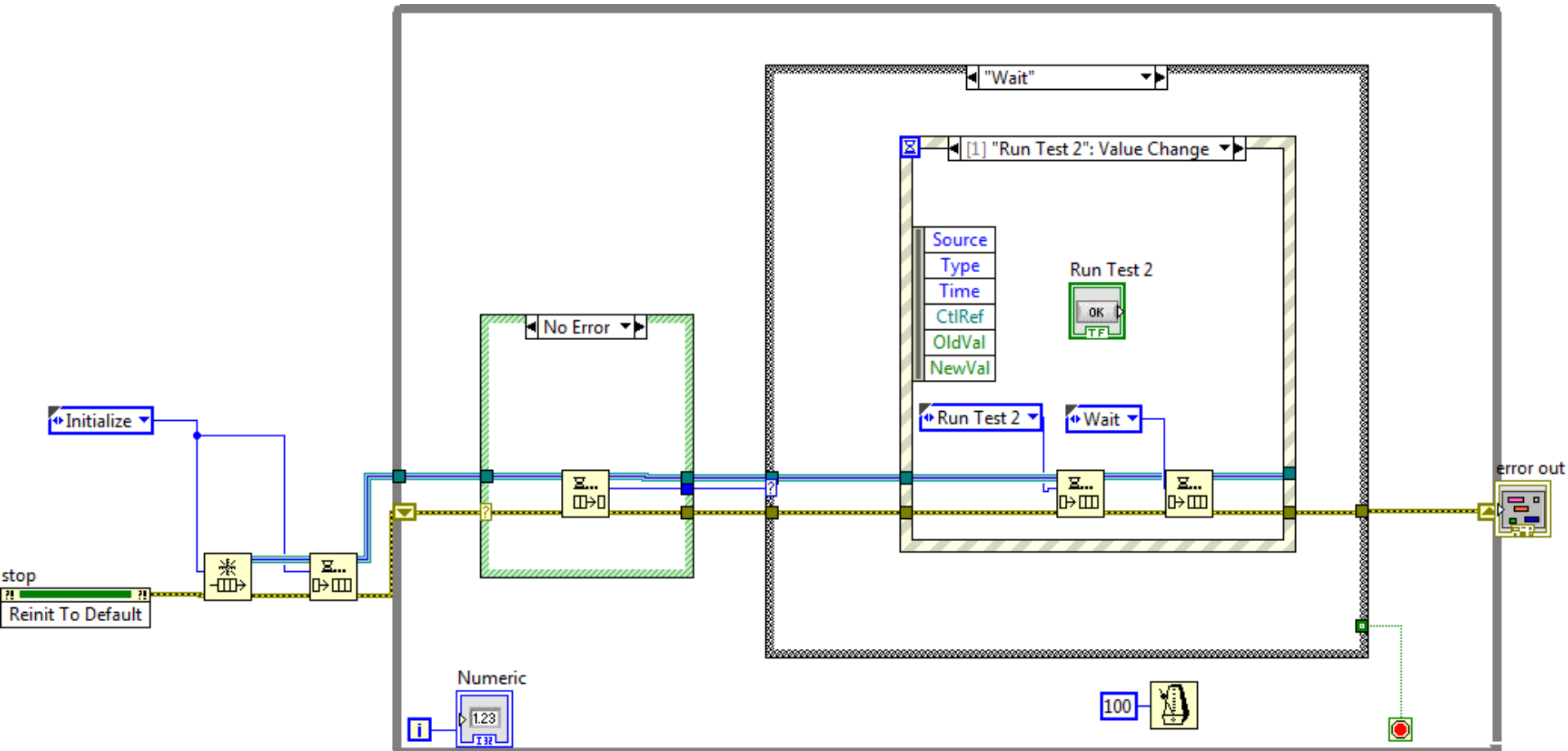
Primer



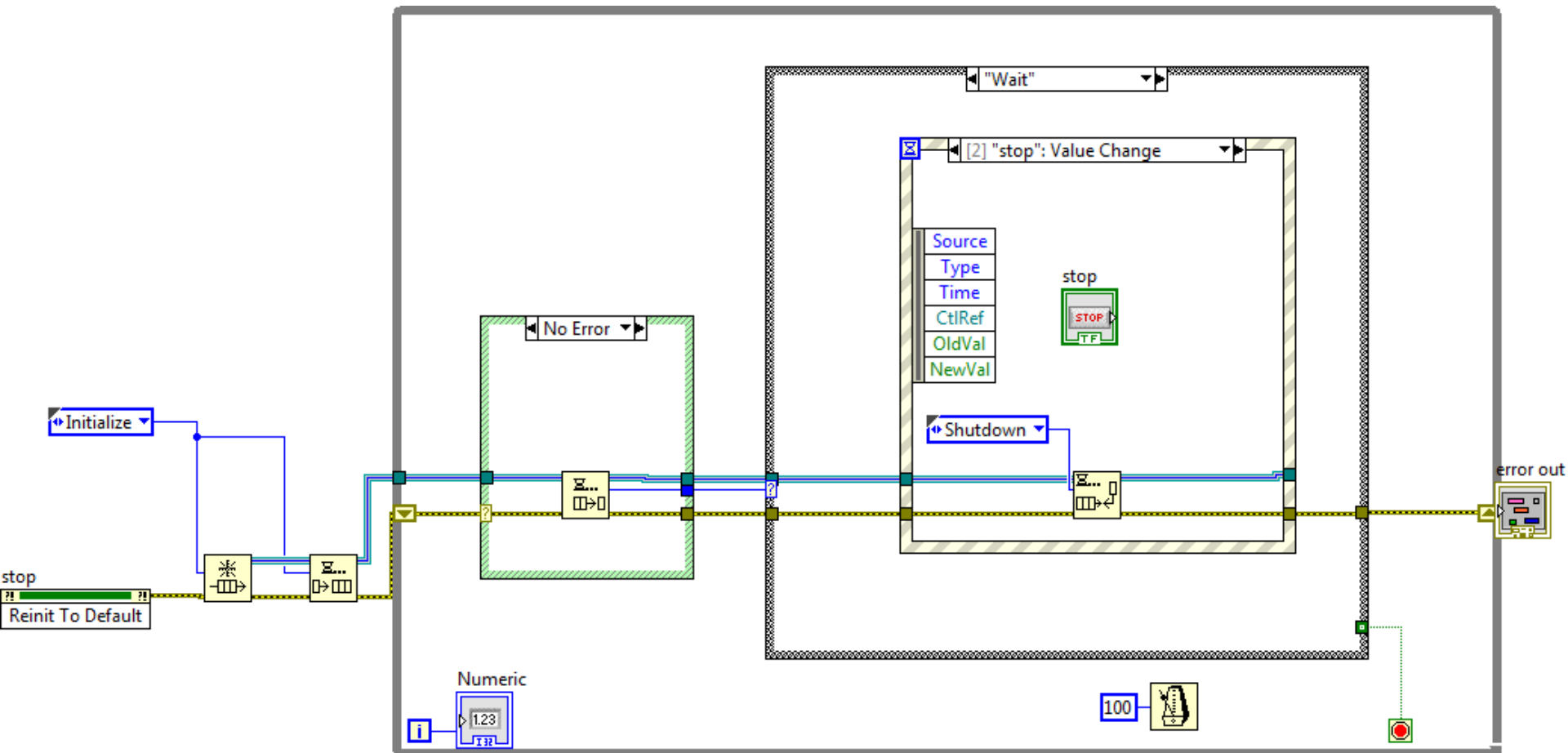
Primer



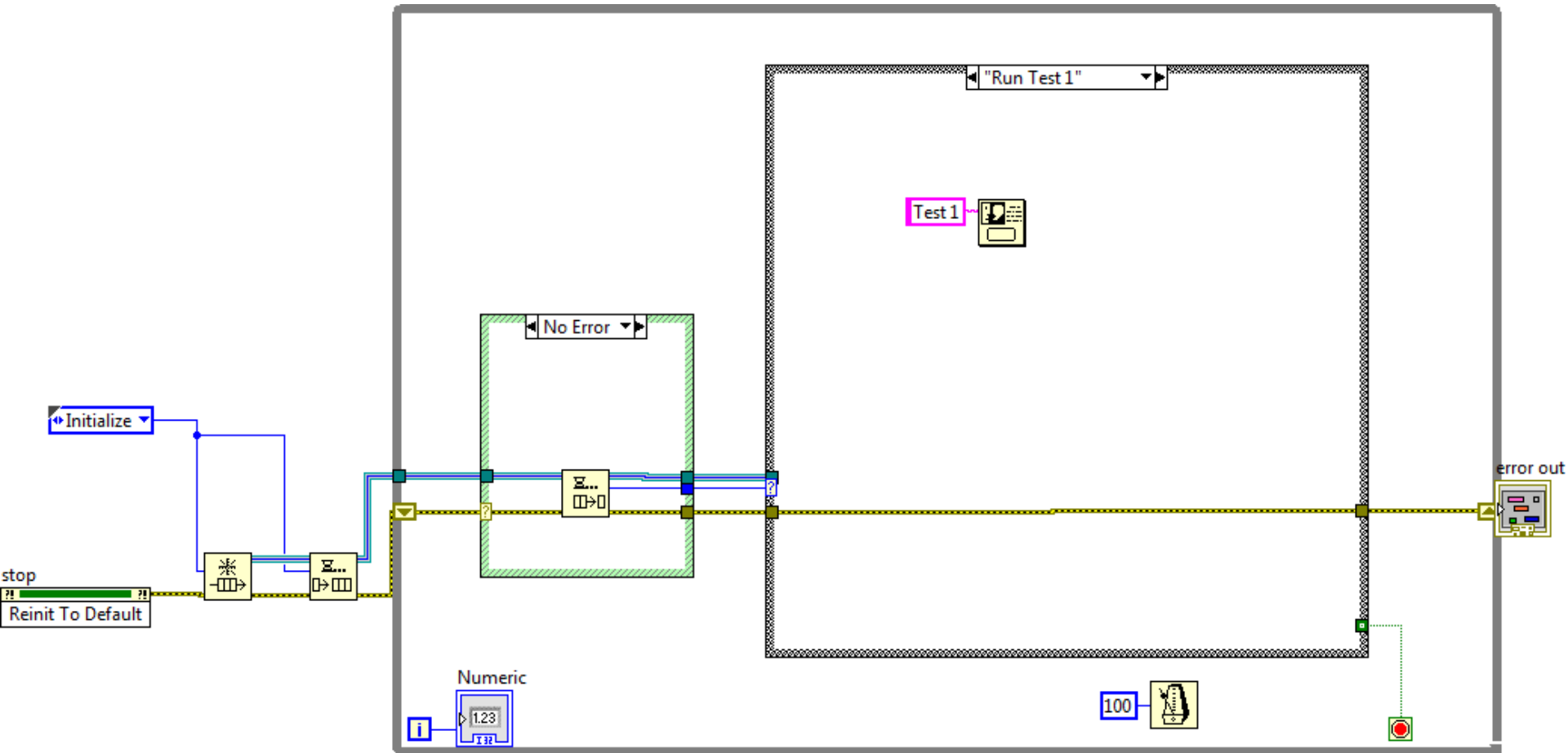
Primer



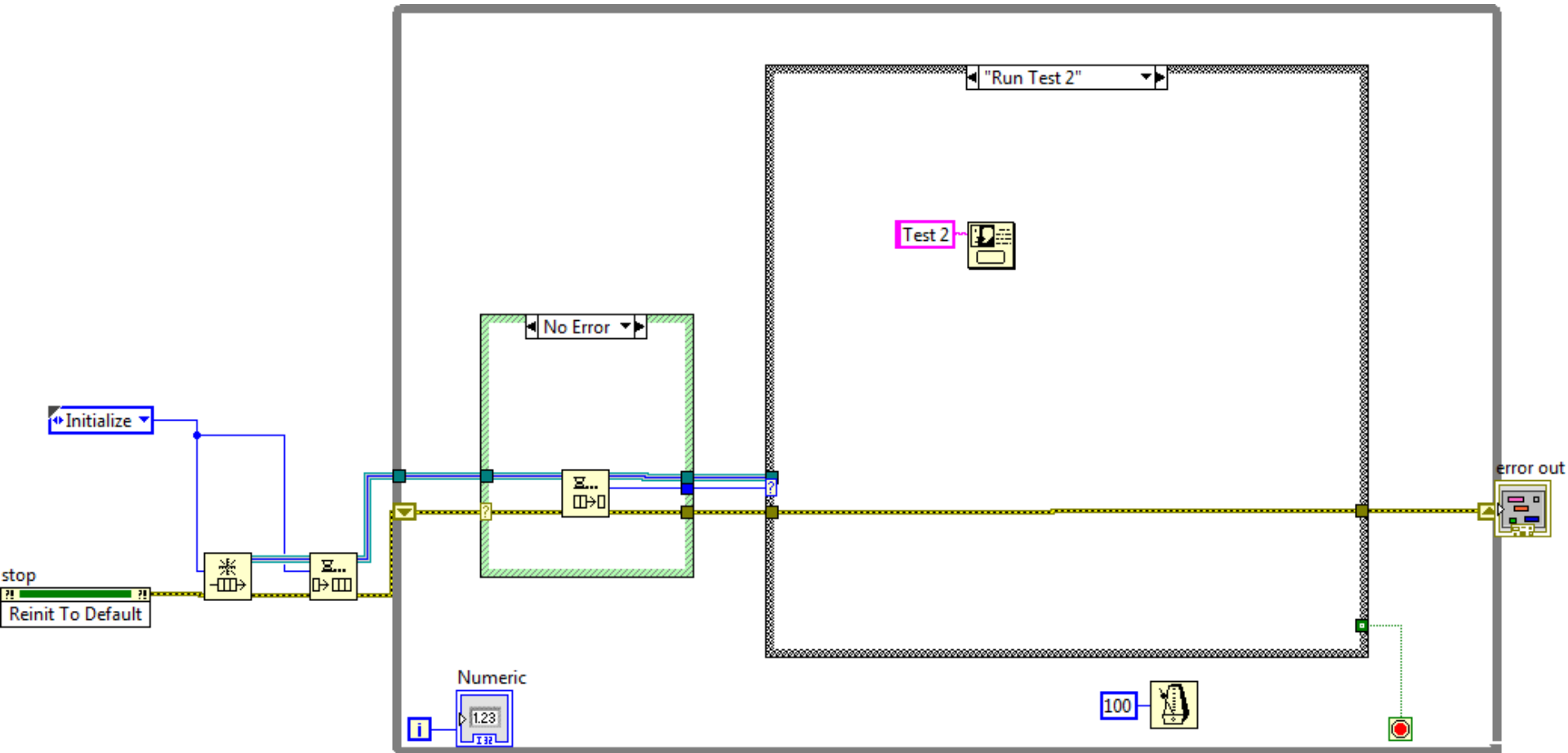
Primer



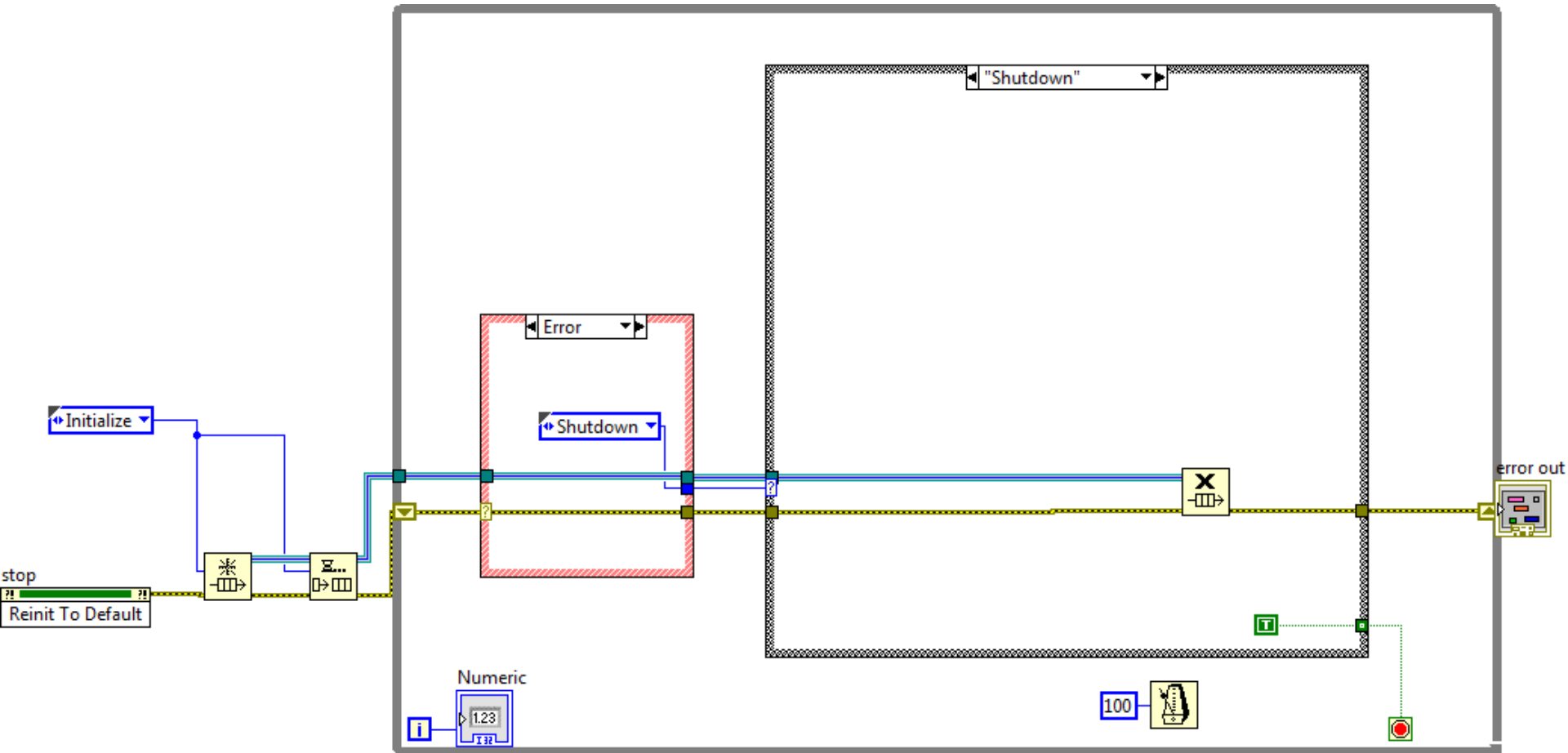
Primer



Primer



Primer



Slučaj sa većim brojem elemenata reda

