Automati stanja

State Machines

Šta je šema projektovanja?

- DEFINICIJA: Dobro projektovano i napravljeno rešenje poznatog problema, tako da to rešenje može lako da se ponovo koristi.

- Šeme projektovanje predstavljaju šablone (templates) za programere.

- Potrebno je dobro dizajnirati kod zbog toga što se na taj način štedi vreme i povećava čitljivost i dugotrajnost koda.

- Naročito je važno voditi računa o skalabilnosti, proširivosti, ponovnom korišćenju koda.

Osobine

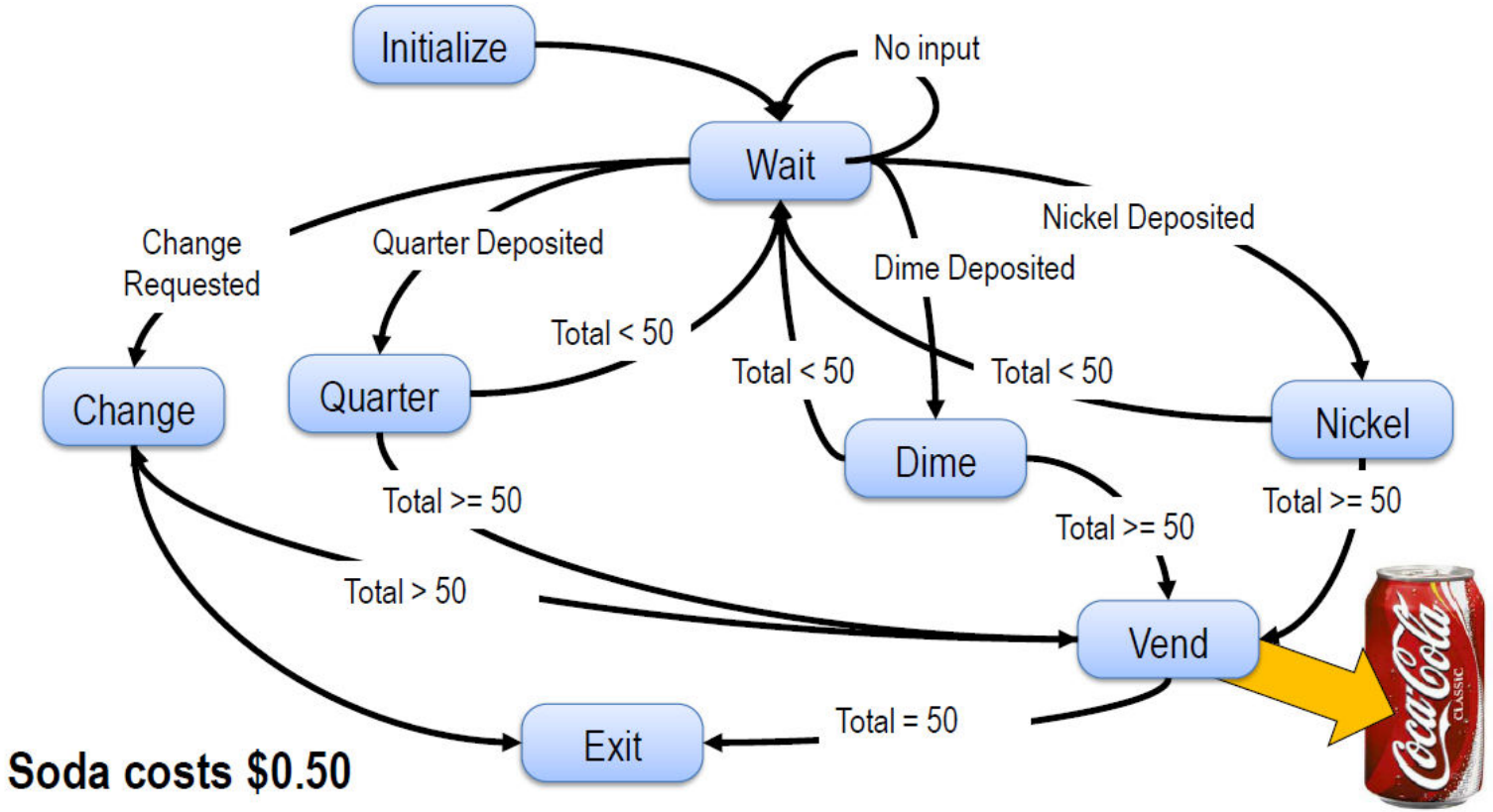
- Skalabilnost – dodavanje novih već postojećih komponenti
- Modularnost – rastavljanje komponenti na manje celine (module)
- Ponovno korišćenje – da li se isti kod može koristiti za druge i buduće projekte
- Proširivost – dodavanje nove funkcionalnosti
- Jednostavnost – najprostije rešenje koje zadovoljava već navedene kriterijume

Neki šabloni projektovanja u LabView-u

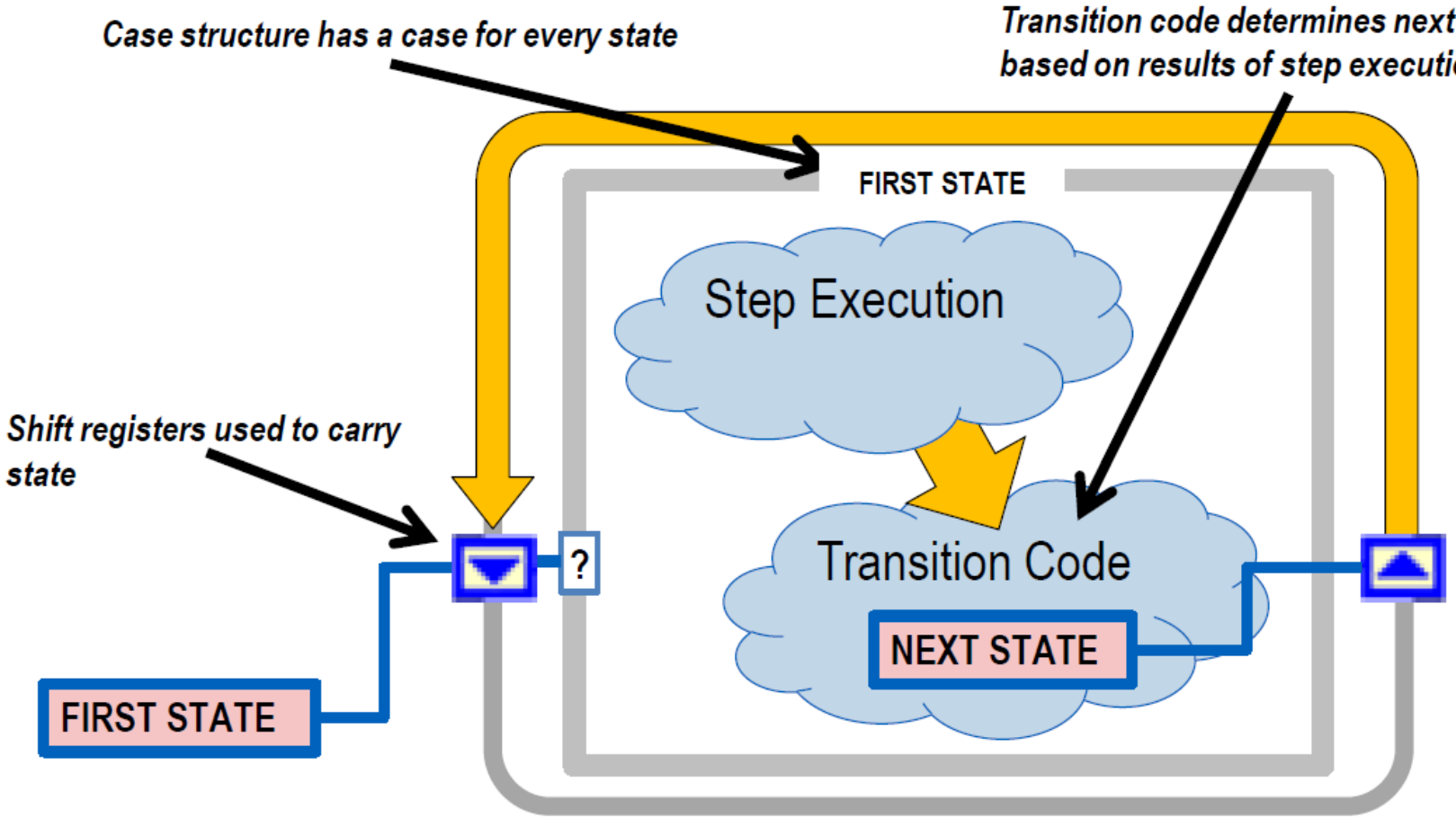
- State Machine
- Functional Global Variable
- Event Driven User Interface
- Producer / Consumer
- Queued State Machine – Producer / Consumer

State Machine – automat stanja

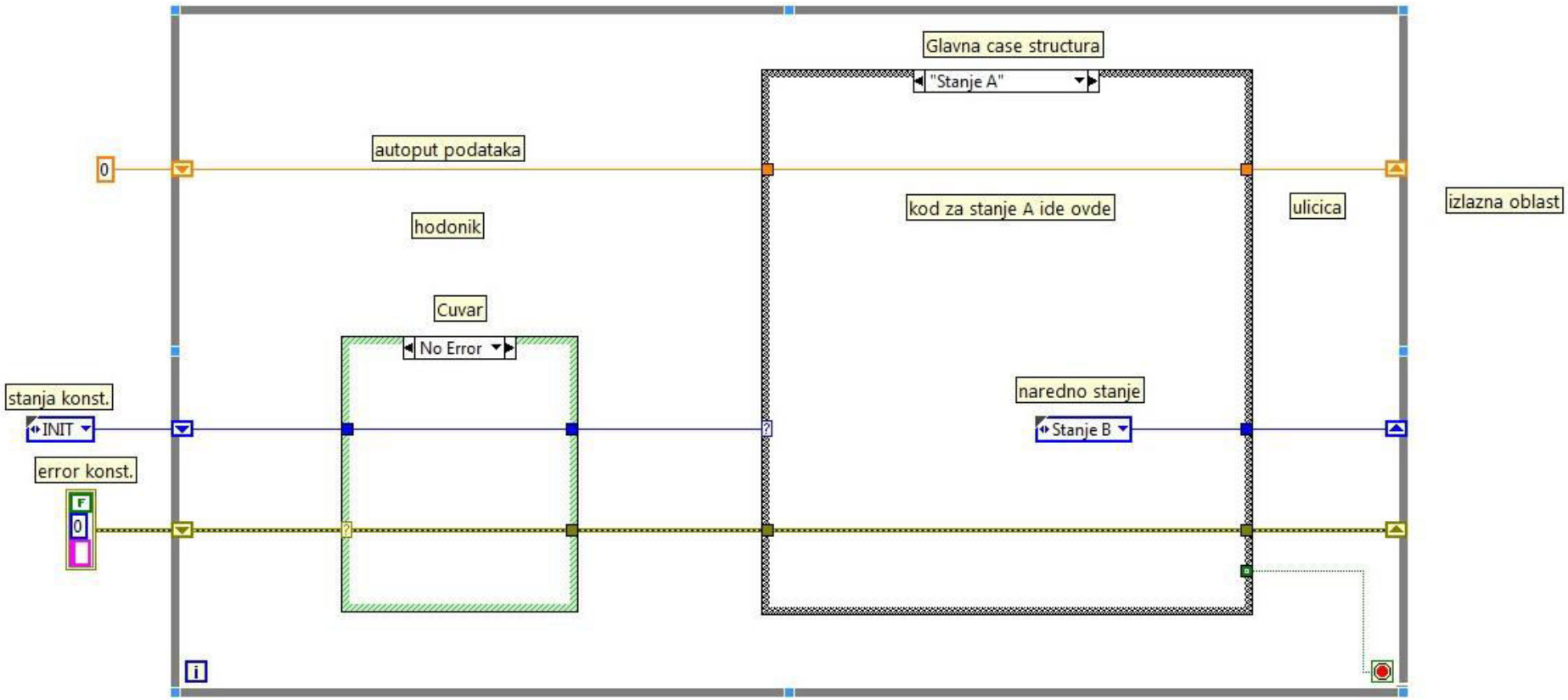
- Potrebno je izvršiti neku sekvencu događaja, čiji je redosled određen programom
- Osnovni elementi automata stanja su STANJA i PRELAZI.



Struktura automata stanja

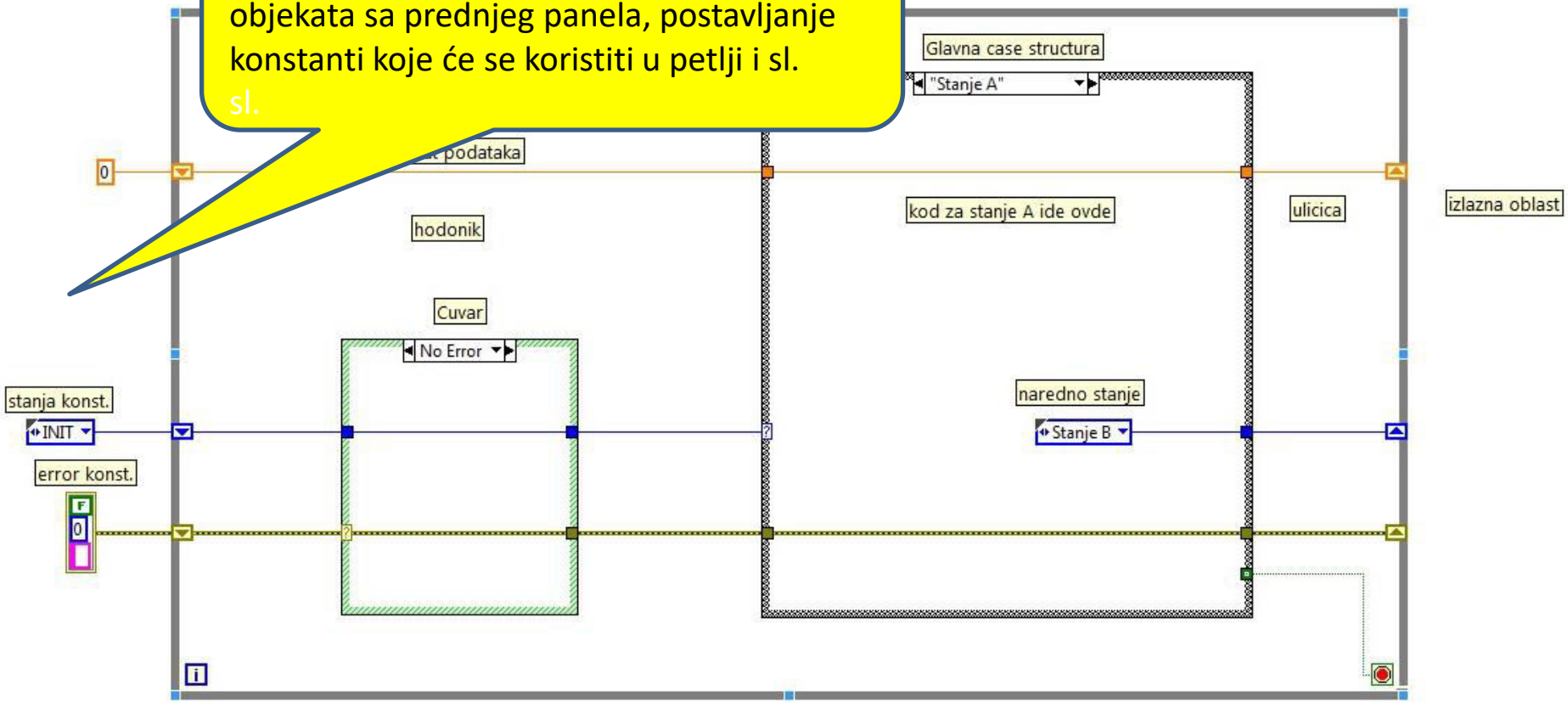


Struktura automata stanja



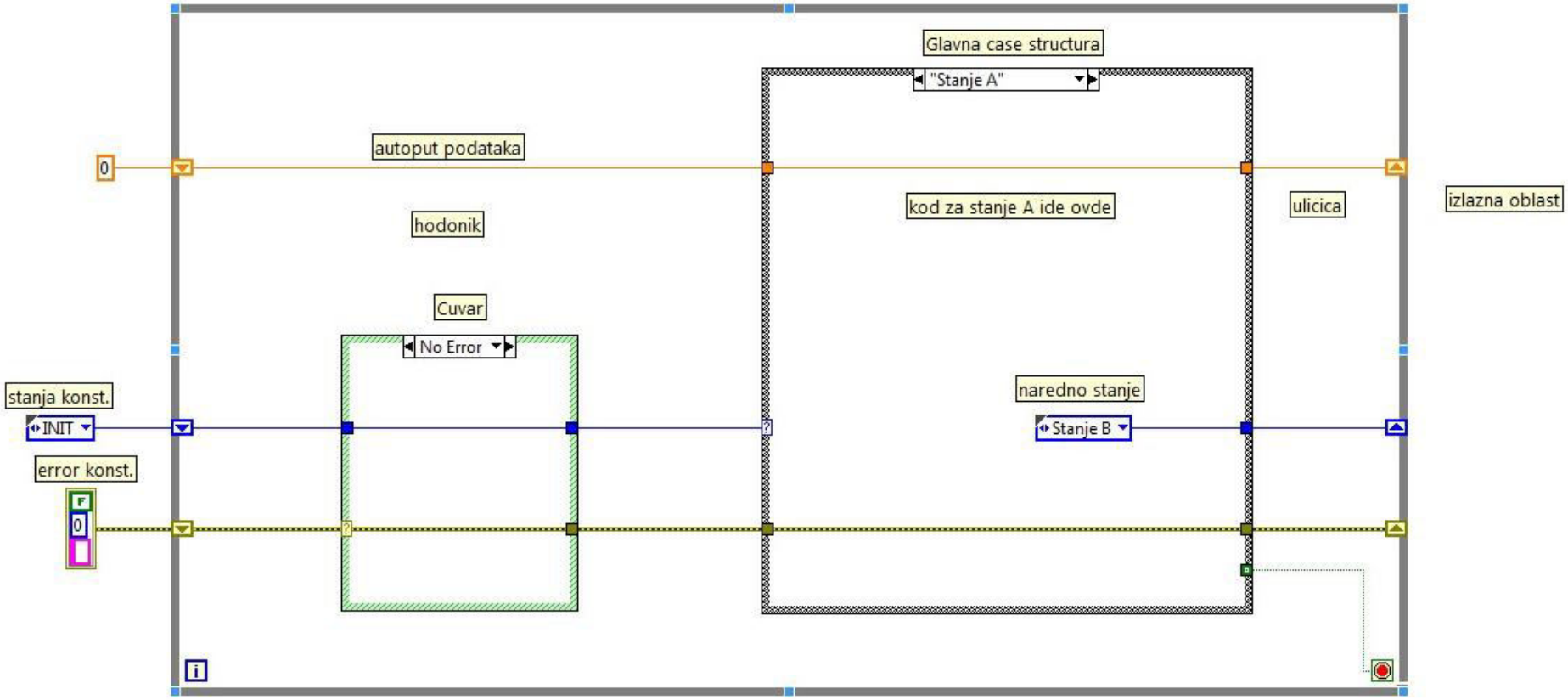
Struktura automata stanja

Ulazna oblast (Input area) – prva se izvršava; inicijalizacija shift registara i objekata sa prednjeg panela, postavljanje konstanti koje će se koristiti u petlji i sl.



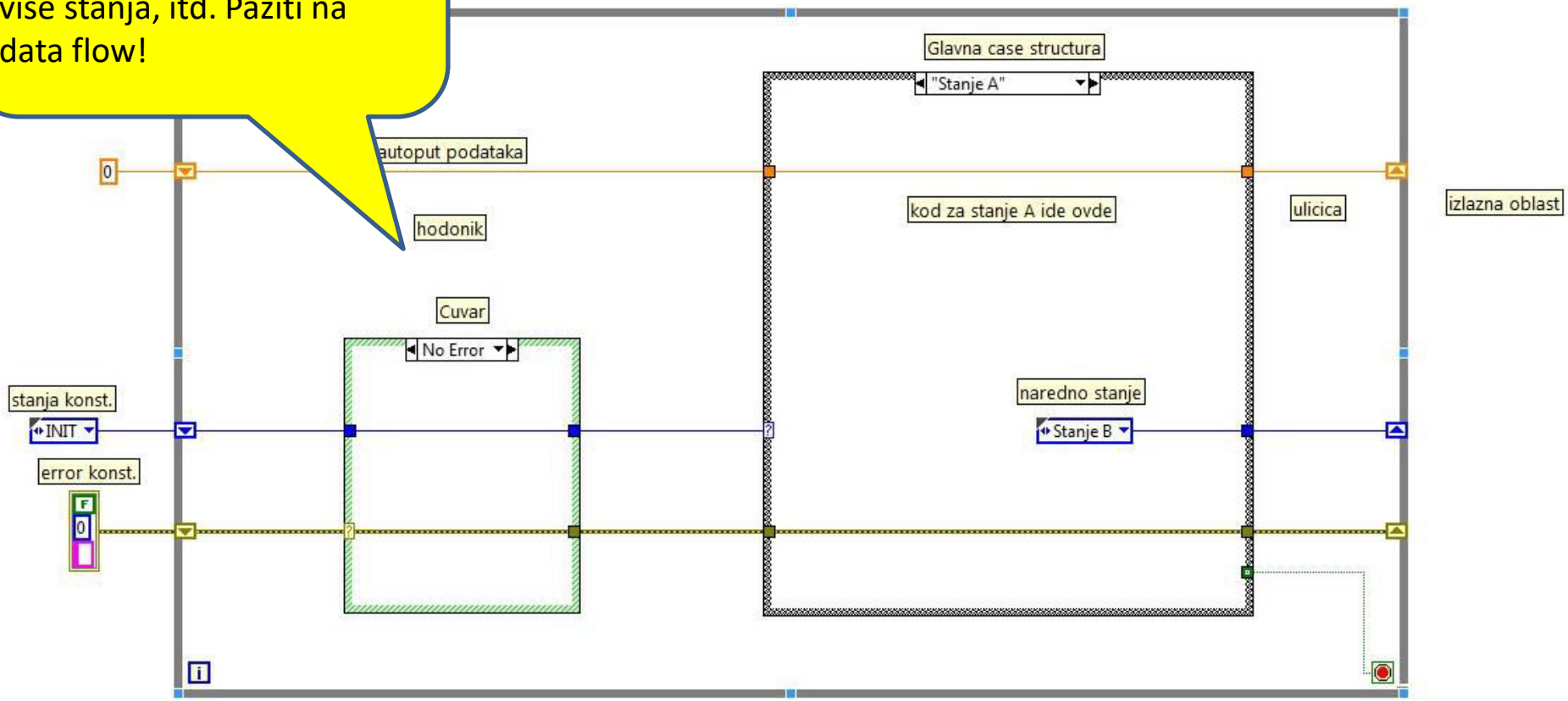
Struktura automata stanja

Glavna while petlja (Main Loop) - vodi automat stanja iz stanja u stanje. Shift registri – čuvanje podataka sa autoputa, informacija o stanju, kao i o postojanju i tipu greške



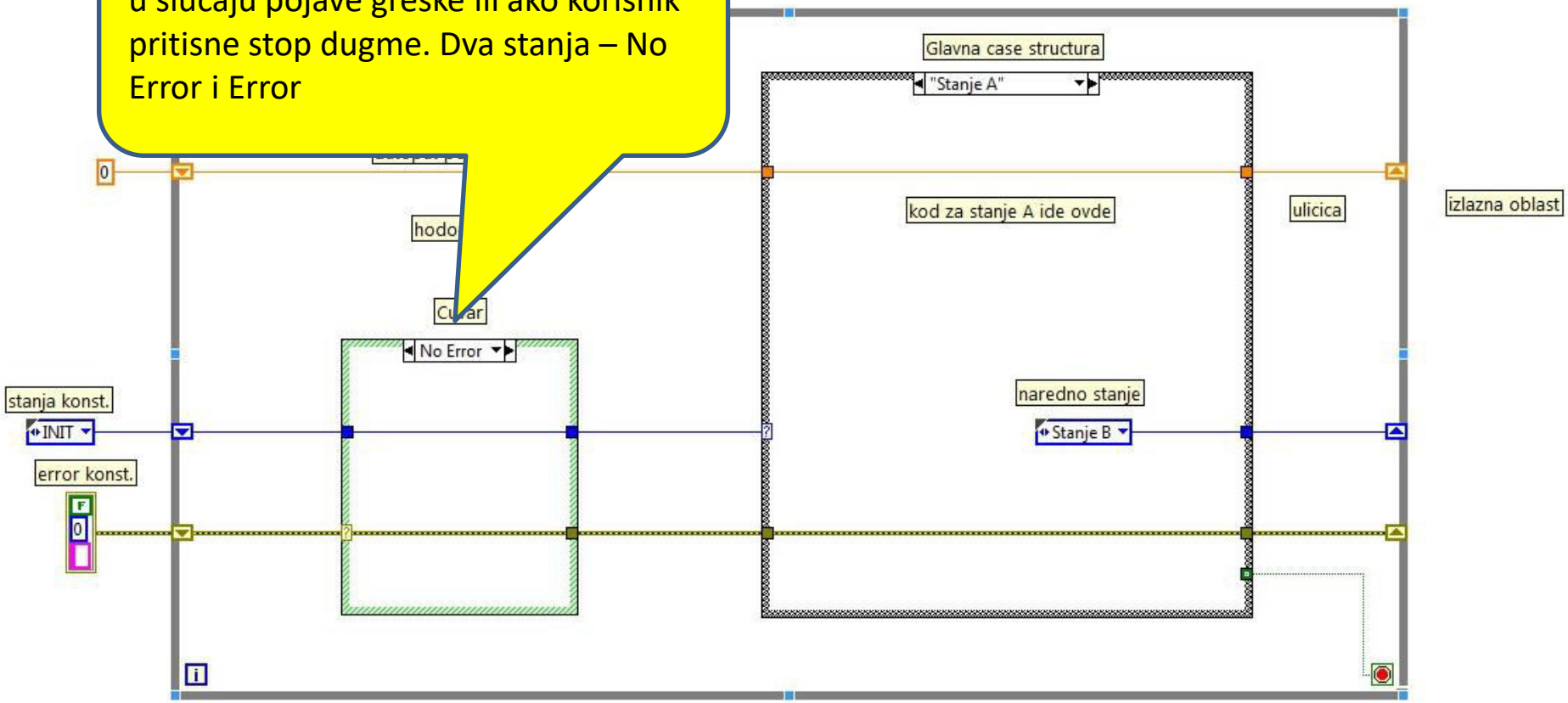
Struktura automata stanja

Hodnik (Lobby) – provera postojanja greške, lokacija za ulaze koji se koriste u svakoj iteraciji petlje ili u više stanja, itd. Paziti na data flow!

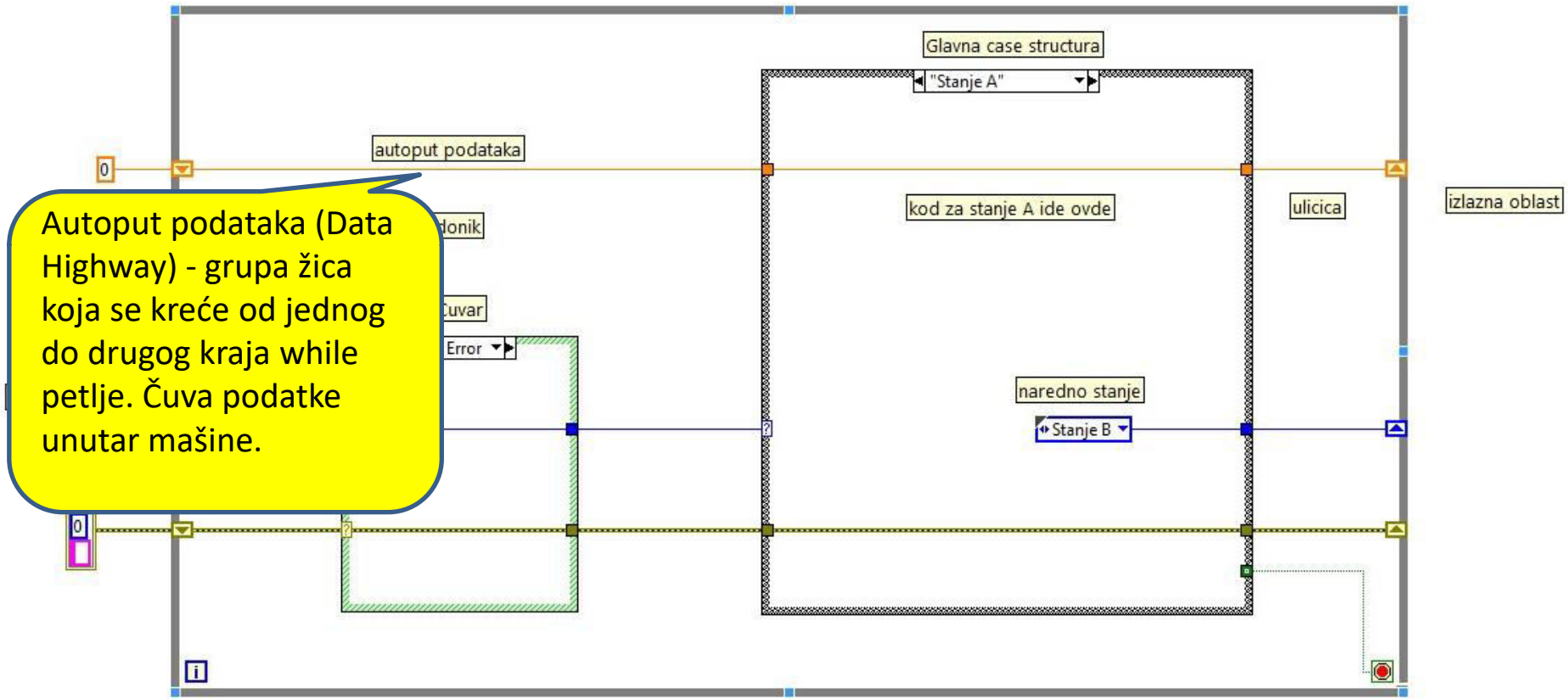


Struktura automata stanja

Čuvar (Guard Clause) - šalje automat stanja u stanje isključenja (shutdown) u slučaju pojave greške ili ako korisnik pritisne stop dugme. Dva stanja – No Error i Error



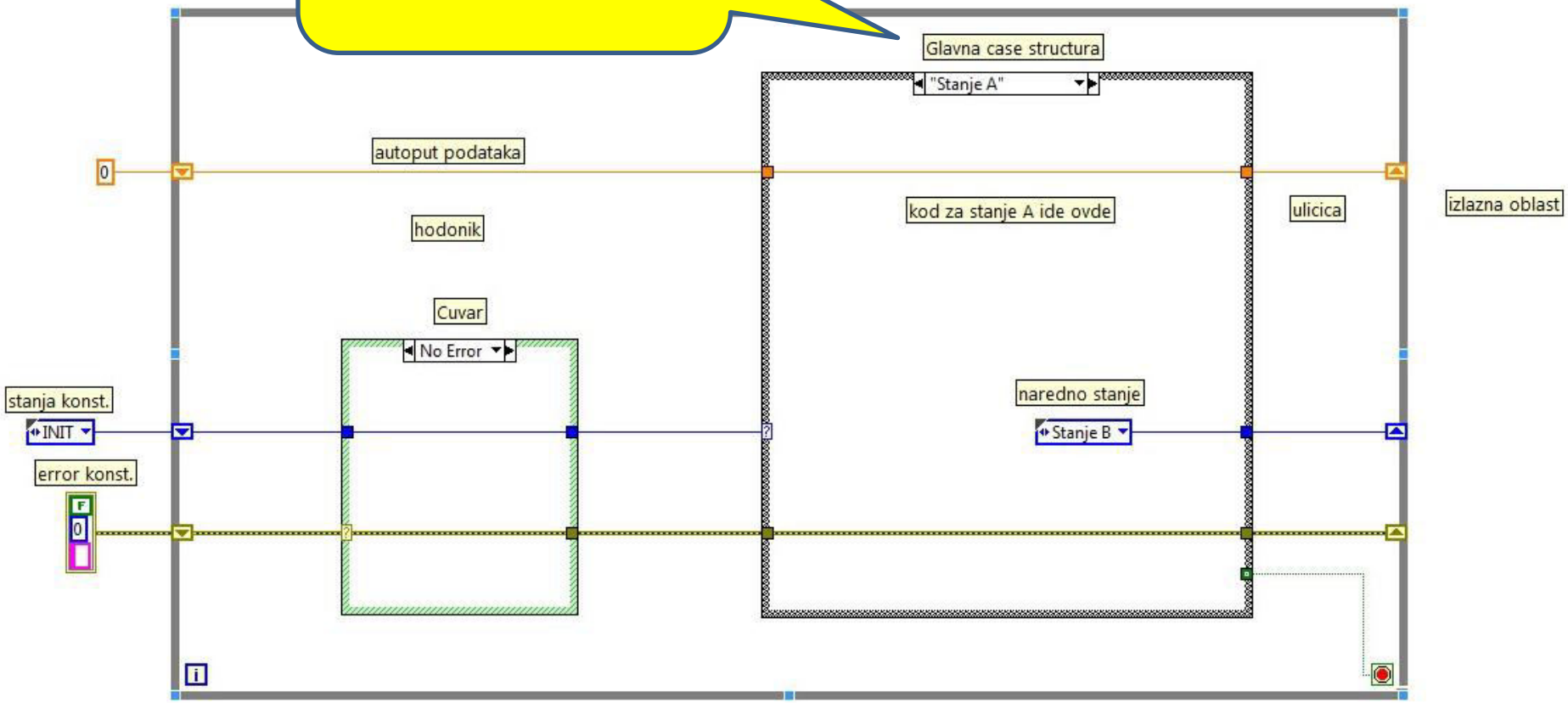
Struktura automata stanja



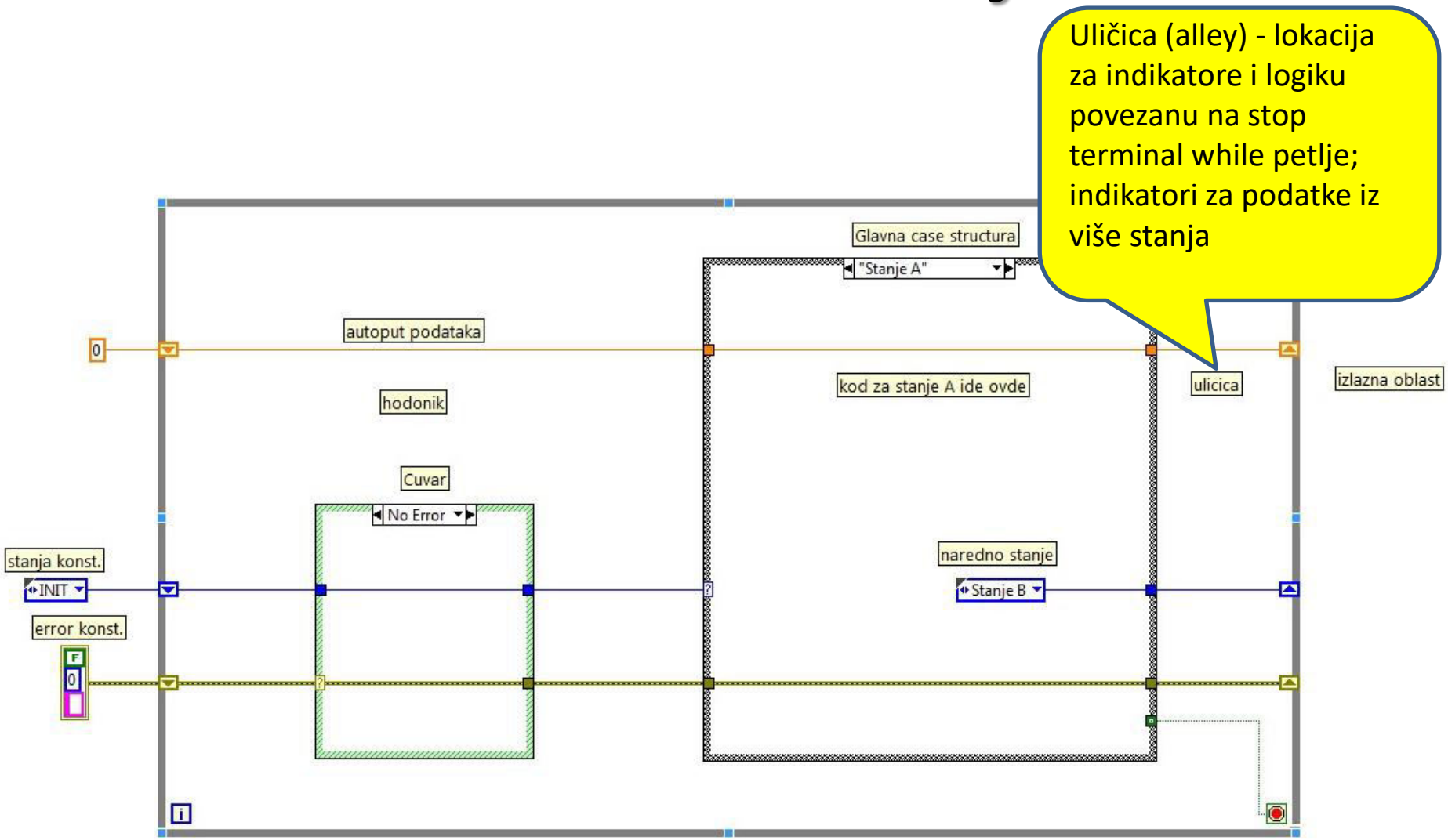
Autoput podataka (Data Highway) - grupa žica koja se kreće od jednog do drugog kraja while petlje. Čuva podatke unutar mašine.

Struktura automata stanja

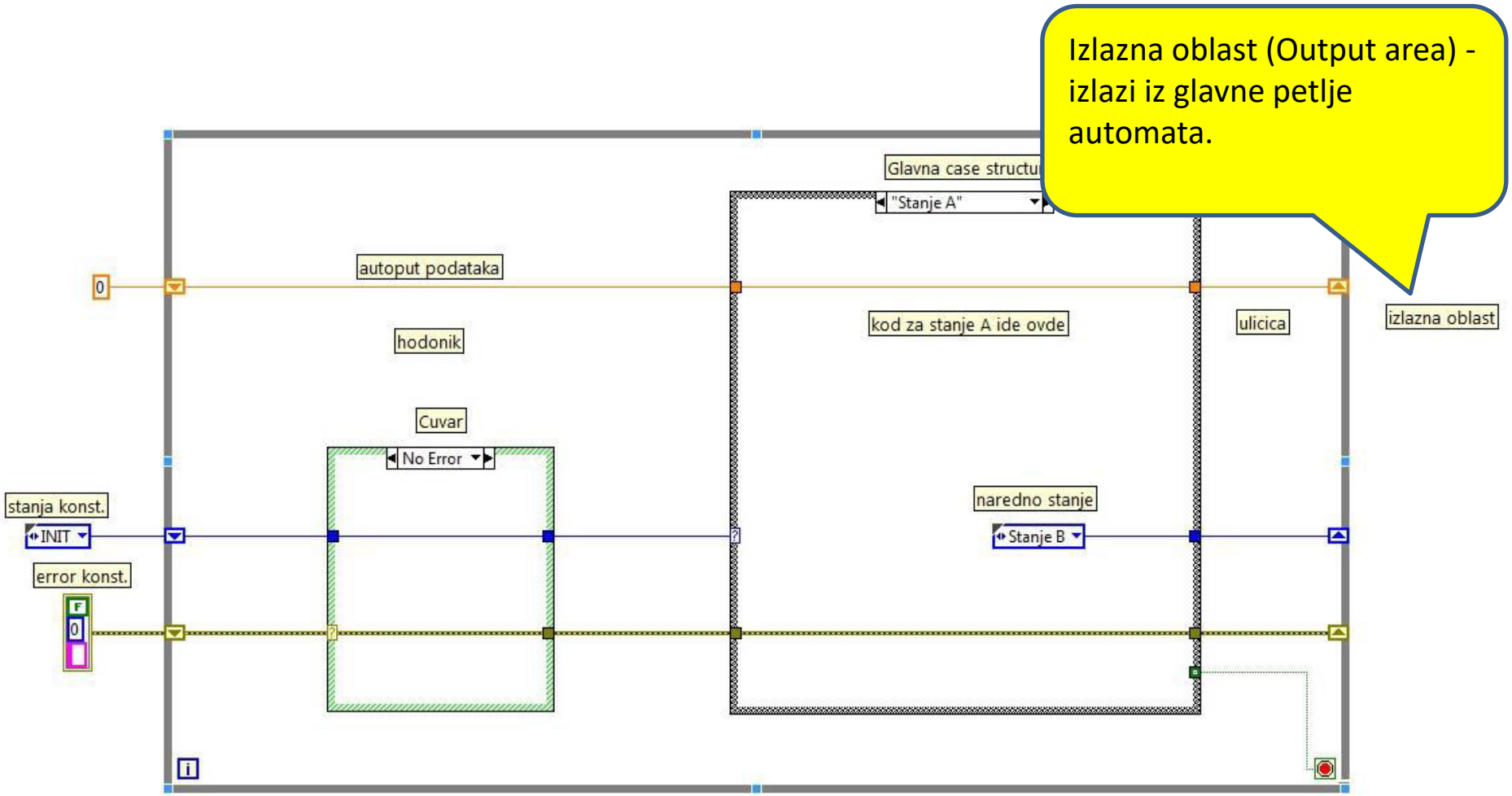
Glavna case struktura (Main Case Structure) - sadrži kod za svako od stanja automata.



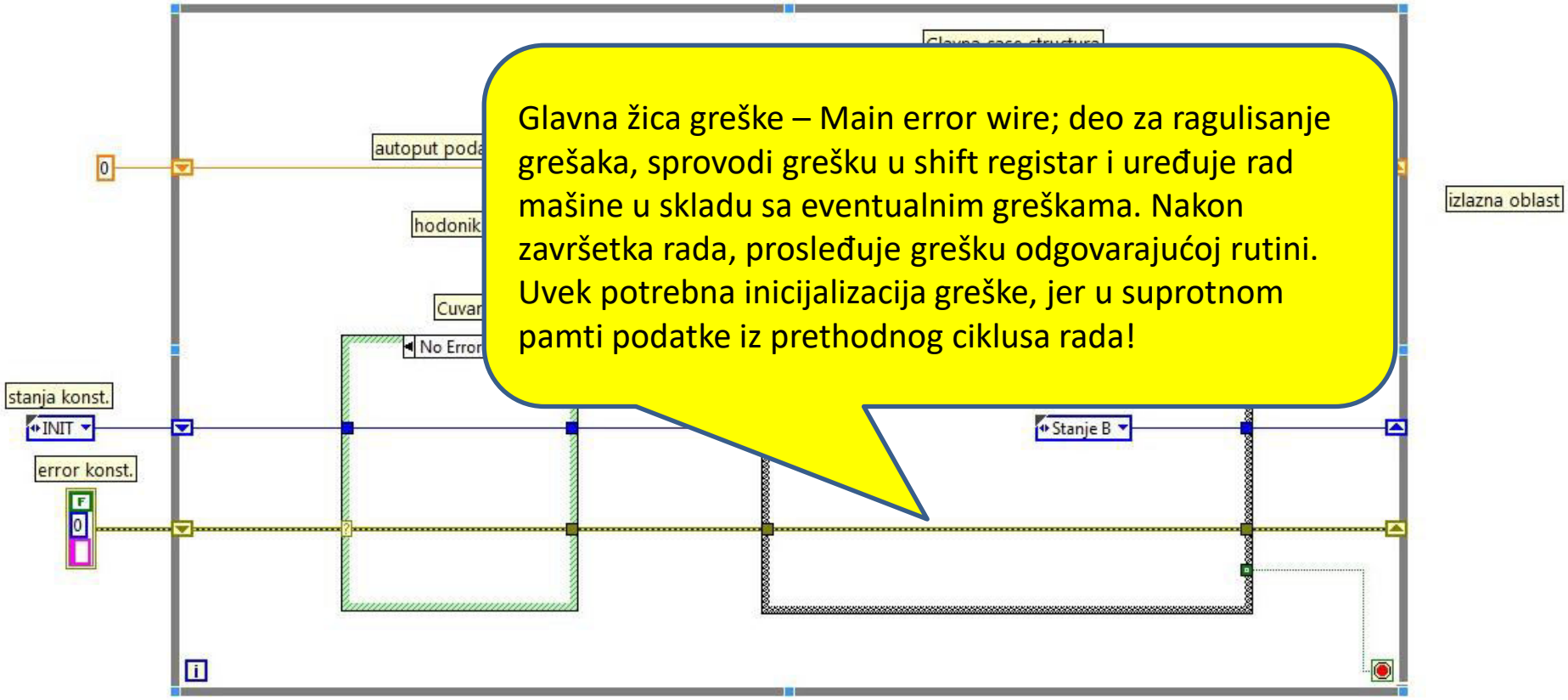
Struktura automata stanja



Struktura automata stanja



Struktura automata stanja



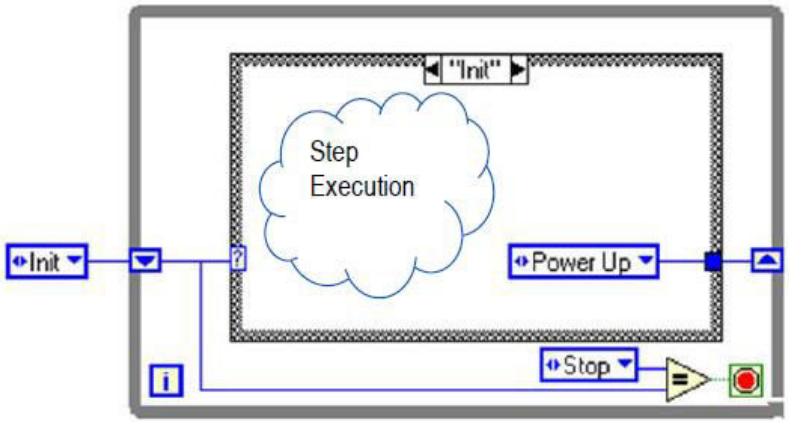
Kako rasporediti elemente?

1. Koliko stanja treba da ima pristup elementu?
2. Da li element treba da se sačuva?

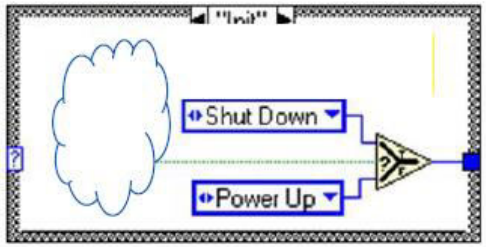
- Generalne sugestije:
  - Hodnik i uličica se izvršavaju u svakoj iteraciji glavne petlje; hodnik – kontrole koje se često menjaju i potrebne su svakom stanju (ili većini); uličica-indikatori koji se ažuriraju u svakom stanju
  - Ako se podatak mora sačuvati – autoput podataka
  - Delovi koji se izvršavaju na kraju, posle glavne petlje – izlazna oblast

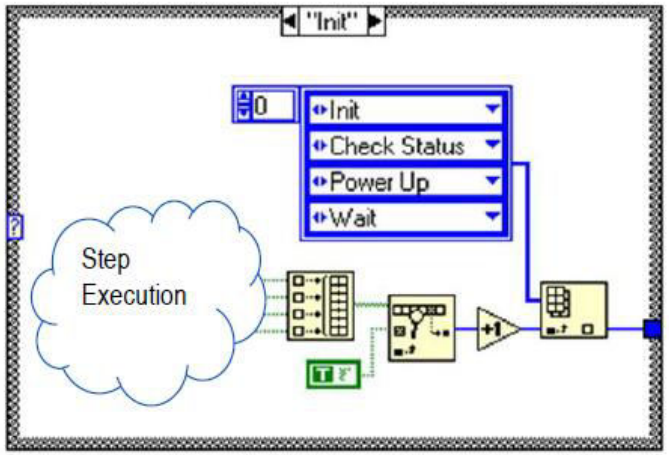
Različiti načini prelaza iz stanja

One-to-one

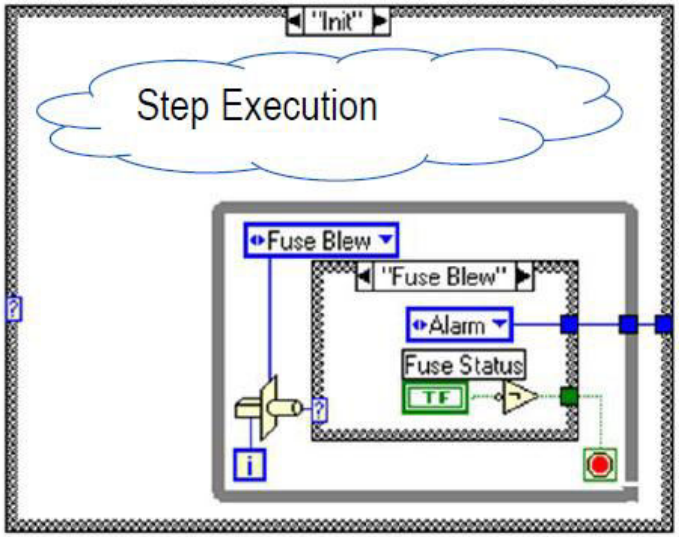


One-to-two





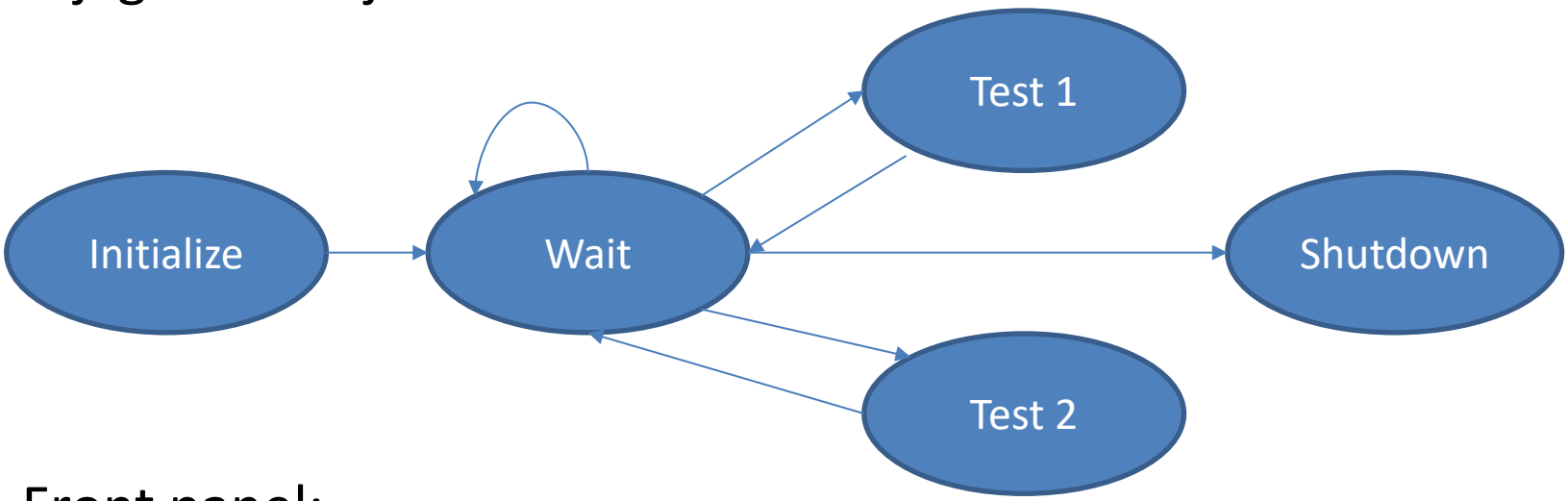
One-to-multiple using arrays



One-to-multiple using While loop

Primer realizacije automata stanja

- Izvršavanje dva testa po izboru
- Stanja: Initialize, Wait, Run Test 1, Run Test 2, Shutdown
- Dijagram stanja:



- Front panel:

Primer realizacije automata stanja

- Stanje **Wait**: