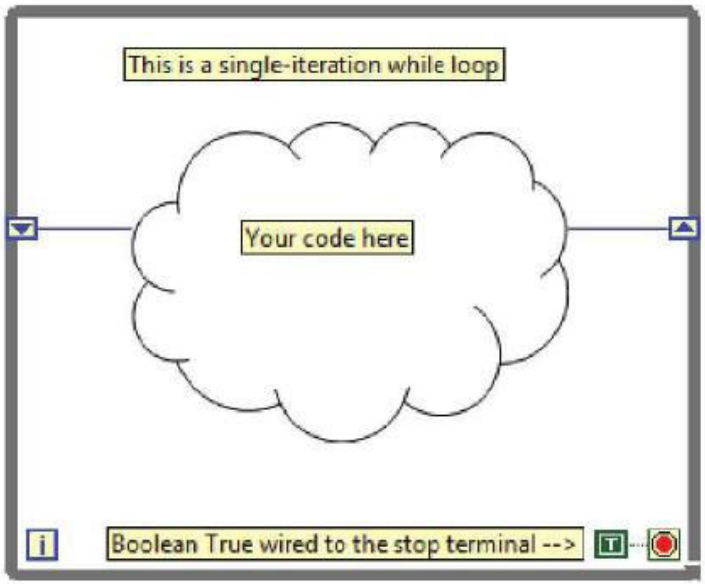
Action engine
šablon projektovanja

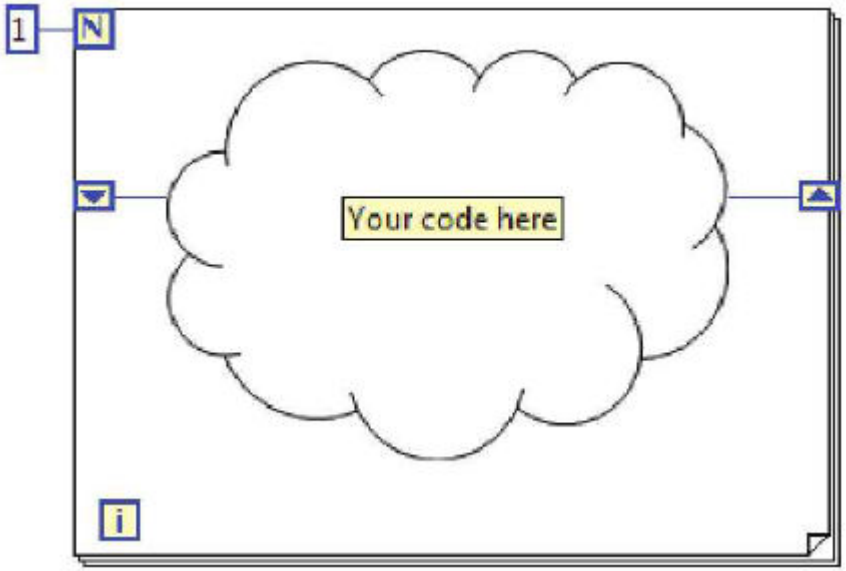
Šta je Action engine šema projektovanja?

- Sub-VI koji skladišti podatke od jednog poziva do drugog i omogućava manipulaciju njima
- Naziva se i funkcionalna globalna promenljiva
- Često se koristi uz automate stanja
- Primeri: brojač i tajmer
- Strukturno slična automatu stanja
- Osnovne osobine:
  - Sadrži petlju sa neinicijalizovanim shift registrima
  - Petlja se izvršava SAMO JEDNOM!

Struktura

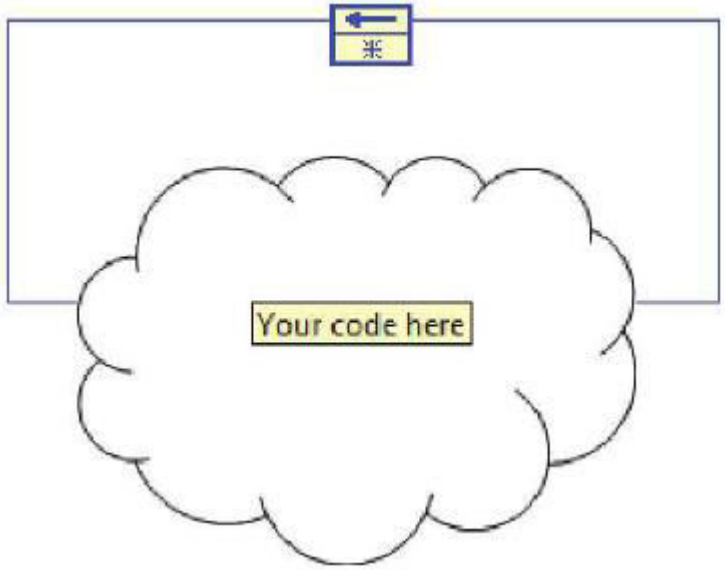
- Neinicijalizovani shift registri čuvaju vrednosti trajno
- Kod procesira vrednost sa ulaza, smešta je u registar i završava petlju
- Implementacija pomoću **while** ili **for** petlje

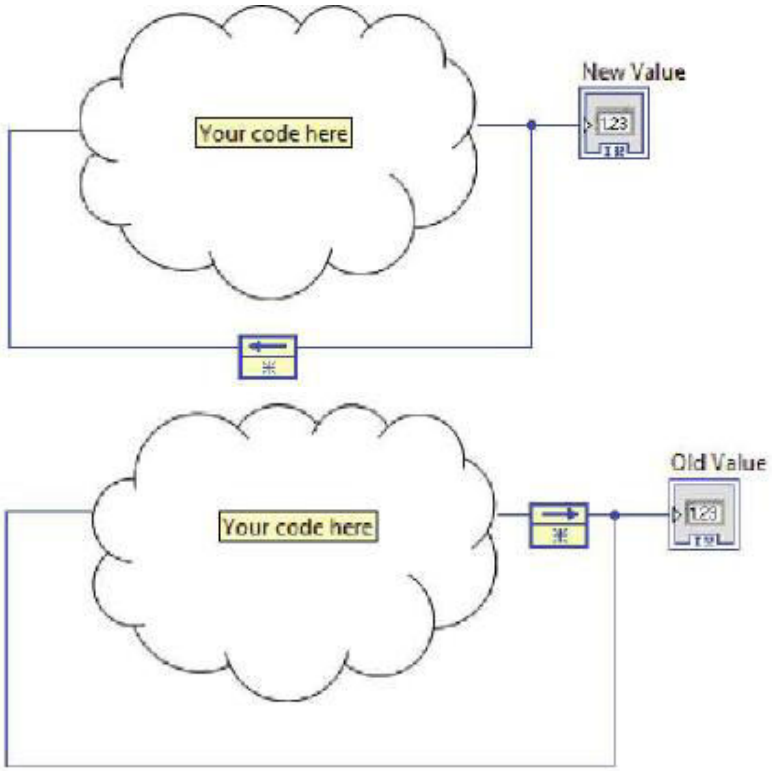




Struktura – feedback node

- Feedback node – čvor koji prenosi (čuva) vrednosti podatka iz jedne iteracije u drugu
- Mogu se inicijalizovati; za razliku od shift registra, nije neophodna petlja!
- Pogodan za action engine šemu



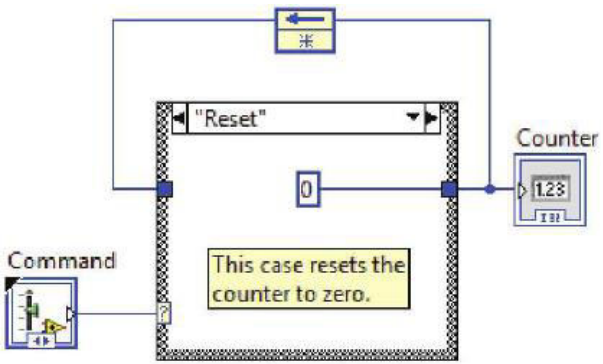


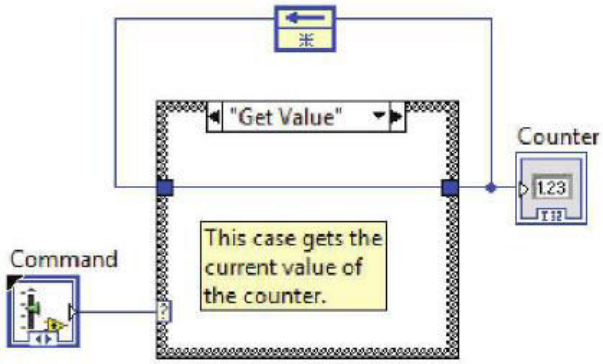
Struktura – feedback node

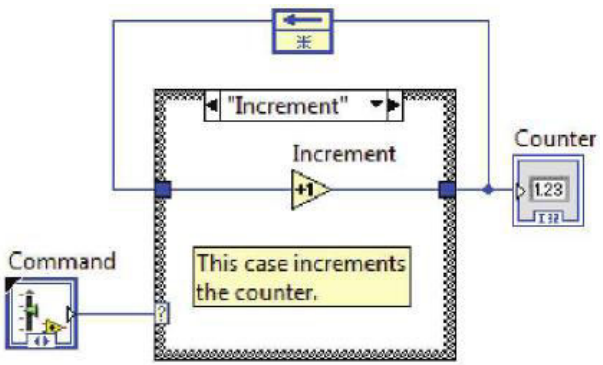
- Prednosti:
  - Dizajniran za primenu u action engine šemama
  - Blok dijagram je pregledniji
  - Lakše se kreira
  - Zauzima manje mesta na blok dijagramu
- Nedostaci:
  - Nije kompatibilan sa starijim verzijama LabView-a
  - Nije često u upotrebi

Primer - brojač

- Namena: čuva vrednosti promenljive od poziva do poziva
- Tri funkcije (najmanje):
  - Reset
  - Preuzmi vrednost
  - Povećaj
  - (Smanji)
- Implementacija

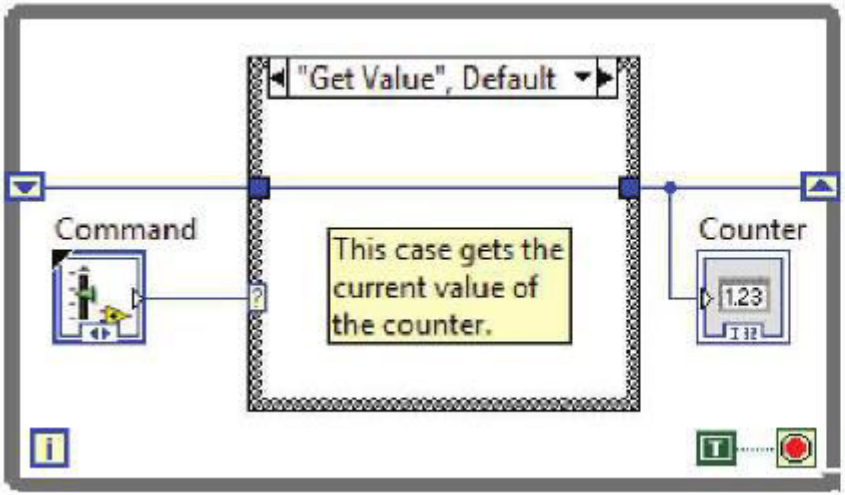


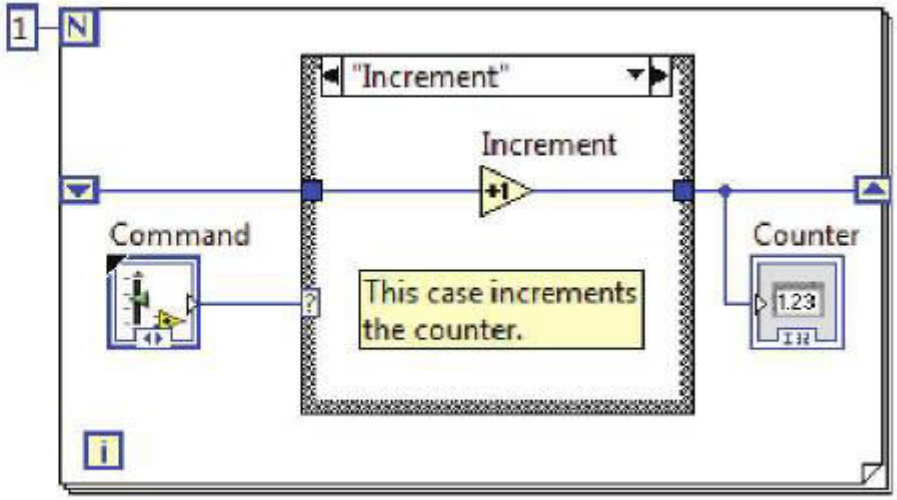




Primer - brojač

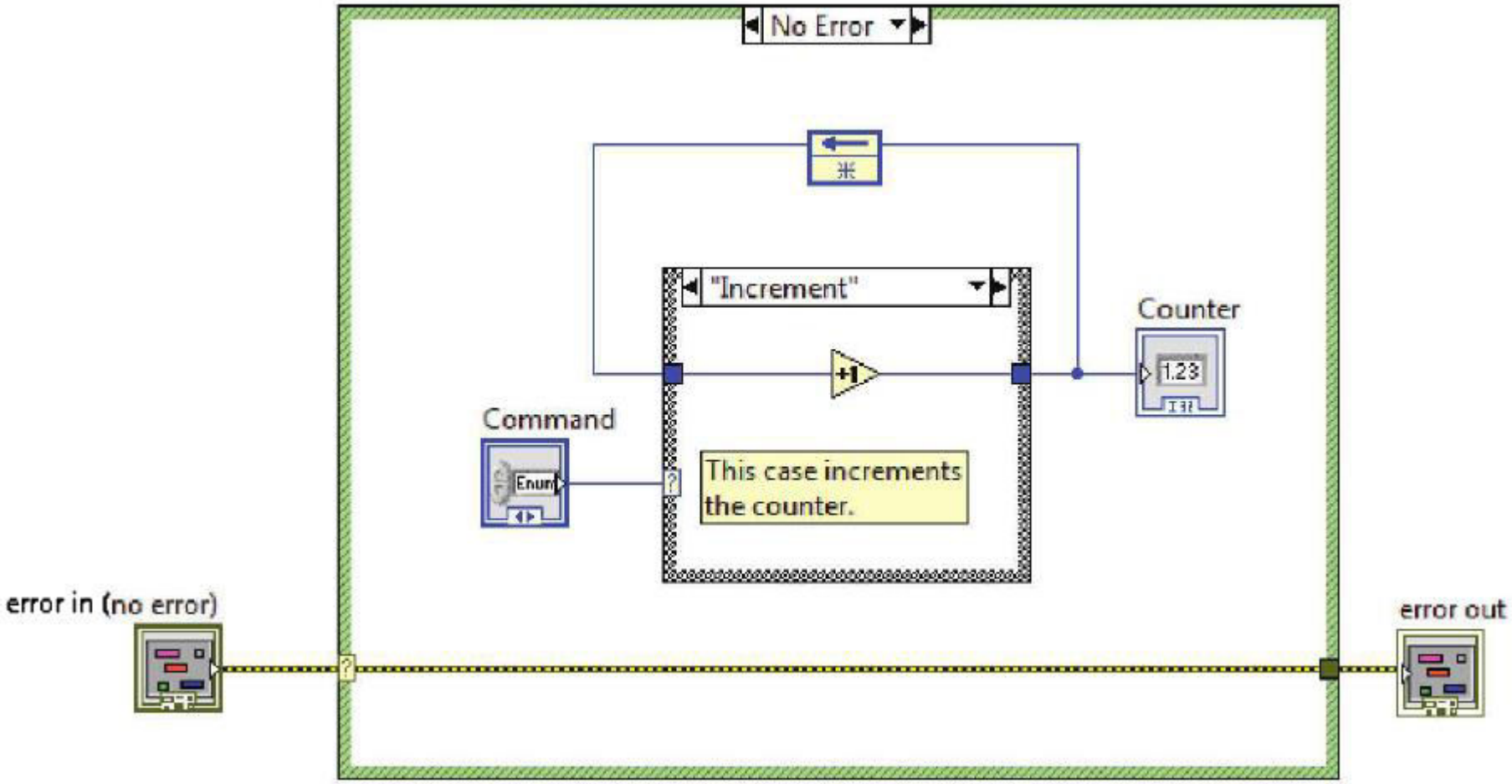
- Implementacija preko petlji





Primer - brojač

- Implementacija sa obradom greške (finalno)



Struktura – promenljive stanja

- Do sada – čuvanje i promena podataka objedinjeni
- Mogu se i razdvojiti – moguća promena načina čuvanja podataka nezavisno od mehanizma promene
- Promenljiva stanja – promenljive koje opisuju “istoriju” sistema; action engine – podaci koji se čuvaju
- Metode se kreiraju unutar kernel Sub-VI-ja; ima funkciju samo promene podataka u skladu sa namenom
- Kernel nema mehanizam čuvanja podataka – on se kreira posebno
- Mehanizam:
  - Action engine prosleđuje promenljive kernelu kao ulaze
  - Kernel izvršava metode, menja podatke i prosleđuje ih na izlaz
  - Action engine čuva podatke do sledećeg poziva

Struktura – promenljive stanja

- Procedura kreiranja action engine-a baziranog na promenljivim stanja (kernel varijanta)

1. Definisati skup funkcija (metode) koje engine treba da izvede

2. Kreirati algoritam za svaku od metoda. Identifikovati vrednosti koje treba da se čuvaju (promenljive stanja)

3. Kreirati kernel – sub-VI koji se sastoji od kontrole (enum) i case structure, koja definiše svaku metodu kao case; svaka promenljiva stanja mora imati ulaz i izlaz

4. Kreirati novi sub-VI koji omogućava čuvanje promenljivih stanja i prosleđuje ih kernelu na obradu. Ovaj novi sub-VI je u stvari action engine

Primer - štoperica

1. Metode:
   - Reset
   - Proteklo vreme
2. Algoritmi:

   Reset:

   1. Preuzmi vrednost tekućeg vremena
   2. Ažuriraj staru vrednost vremena tekućom
   3. Prosledi 0 kao vrednost proteklog vremena

   Proteklo vreme:

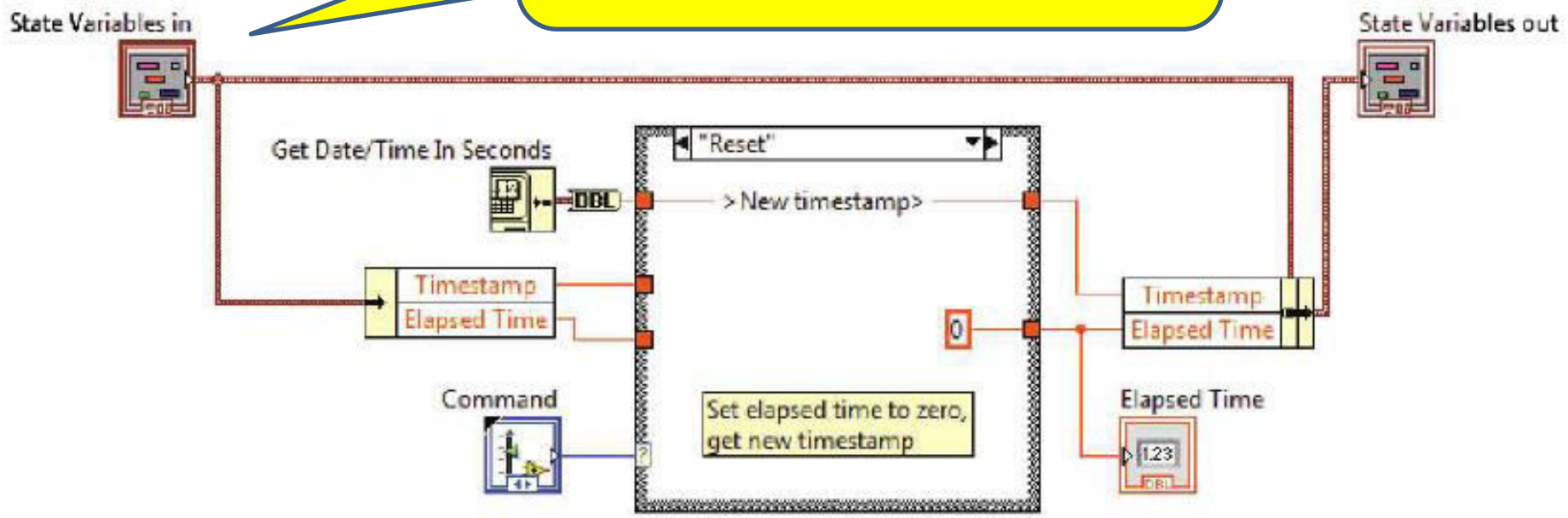
   1. Preuzmi vrednost tekućeg vremena
   2. Oduzmi prethodnu vrednost vremena od tekuće (Δt)
   3. Dodaj Δt na prethodnu vrednost proteklog vremena
   4. Ažuriraj staru vrednost vremena tekućom
   5. Prosledi novu vrednost proteklog vremena

Primer - štoperica

3. Kernel

Reset

Ulazi i izlazi (promenljive stanja) najčešće se grupišu u klastere
sl.

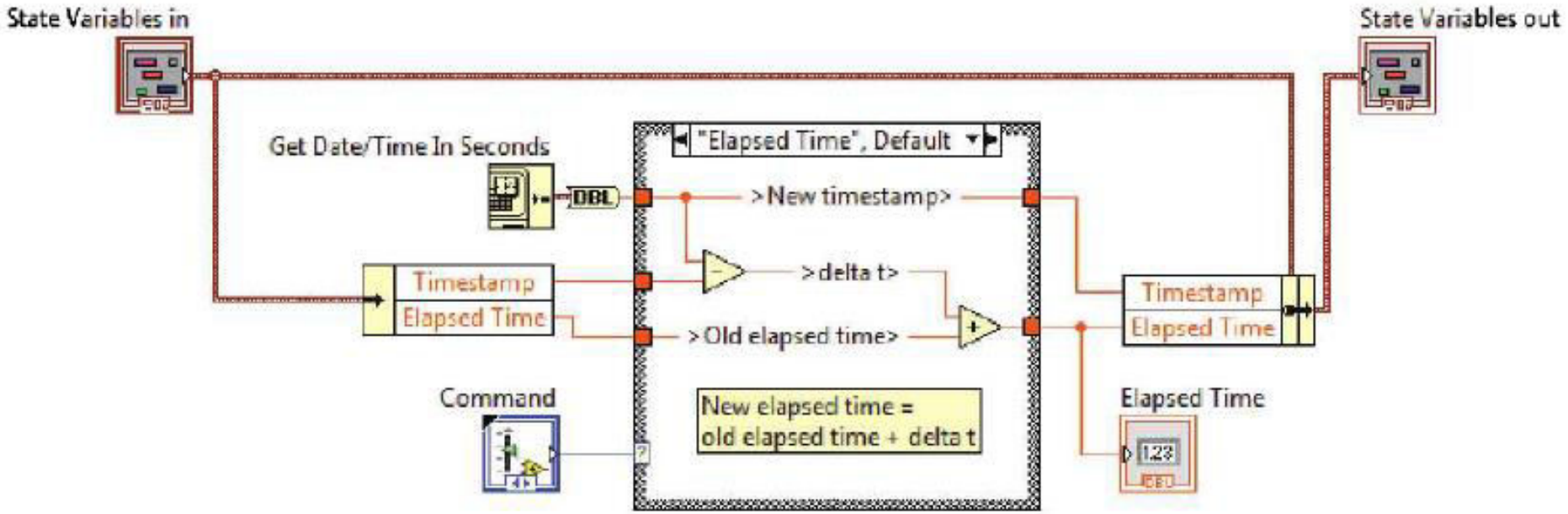


Komanda je case selektor

Primer - štoperica

3. Kernel

Proteklo vreme



Primer - štoperica

4. Action engine (čuvanje+kernel) – sve varijante