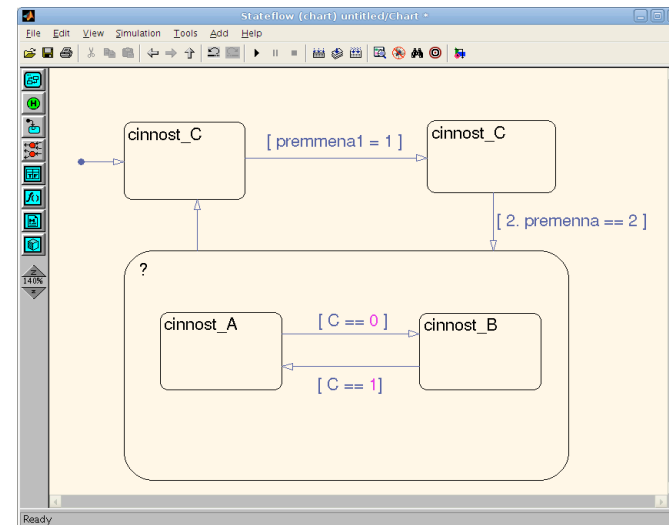


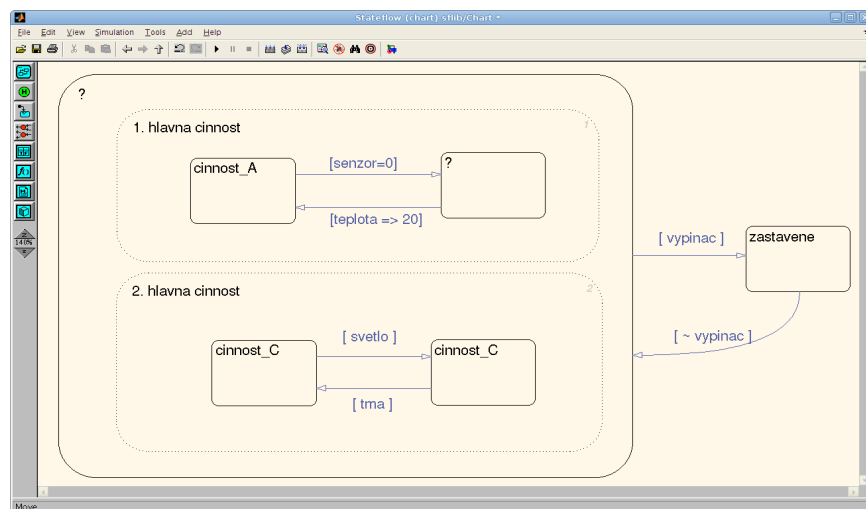
Prečo to nefunguje  
a  
ako to vlastne funguje?

Michal Kvasnica

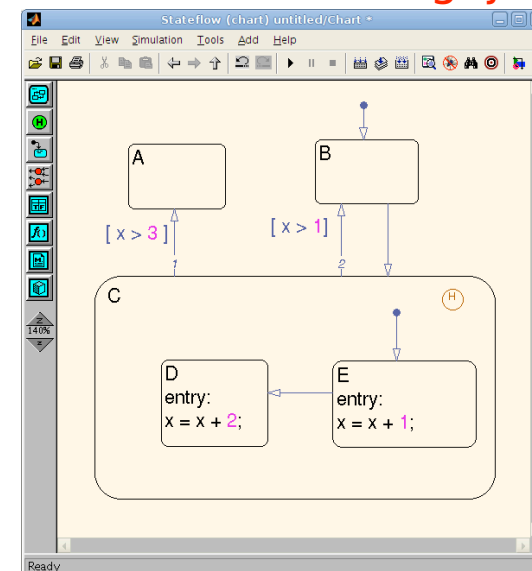
Prečo to nefunguje?



Prečo to nefunguje?



Ako to vlastne funguje?



## Postup návrhu logického riadenia na báze stavov

- Analyzovať riadený systém a pravidlá riadenia
- Vytvoriť:
  - zoznam stavov zariadenia
  - príkazy vykonávané v jednotlivých stavoch
  - prechody medzi stavmi
  - podmienky prechodov
- Analyzovať riadiaci systém a zistiť, či boli implementované všetky potrebné pravidlá a činnosti

## Prečo mnou navrhnutý riadiaci systém nefunguje?

- Čo znamená “nefunguje”?
  - vôbec nefunguje
  - nefunguje správne

## Postup návrhu logického riadenia na báze stavov

- Analyzovať riadený systém a pravidlá riadenia
- Vytvoriť:
  - zoznam stavov zariadenia
  - príkazy vykonávané v jednotlivých stavoch
  - prechody medzi stavmi
  - podmienky prechodov
- Analyzovať riadiaci systém a zistiť, či boli implementované všetky potrebné pravidlá a činnosti

## Prečo mnou navrhnutý riadiaci systém nefunguje?

- Čo znamená “nefunguje”?
  - vôbec nefunguje
  - nefunguje správne
- Vôbec nefunguje:
  - prečo? (zistiť príčiny)
  - ako ho opraviť? (navrhnuť opravu)

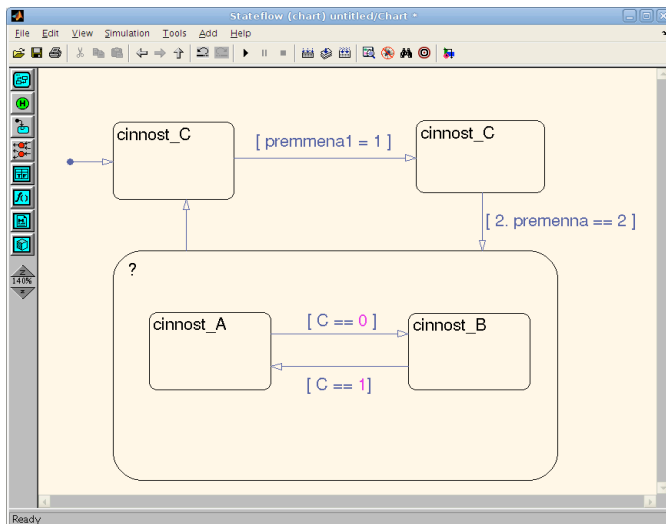
## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - všetky stavy musia byť pomenované
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

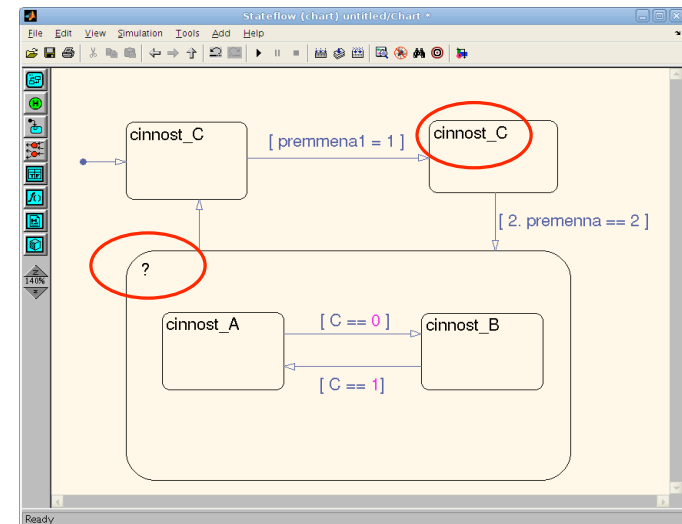
## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - **všetky stavy musia byť pomenované**
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

## Prečo systém vôbec nefunguje?



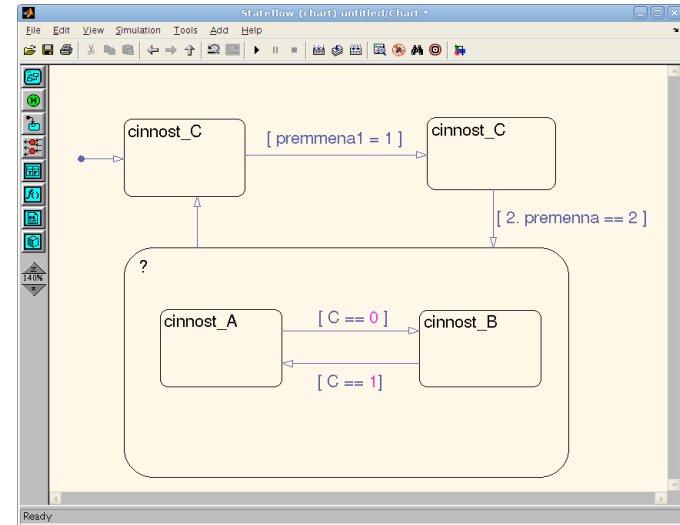
## Prečo systém vôbec nefunguje?



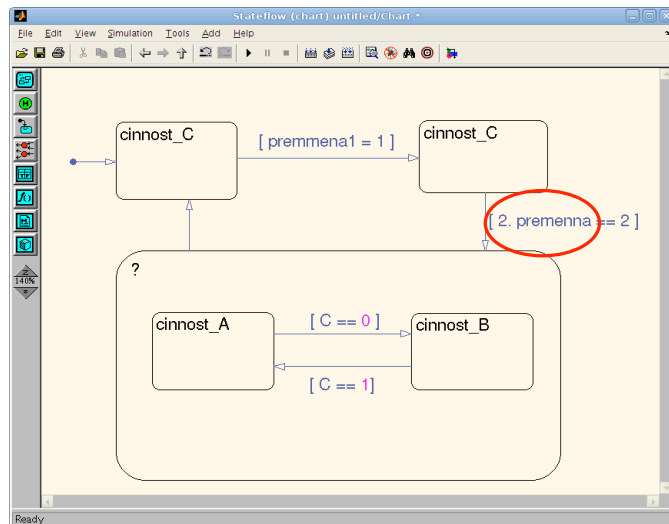
## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - všetky stavy musia byť pomenované
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

## Prečo systém vôbec nefunguje?



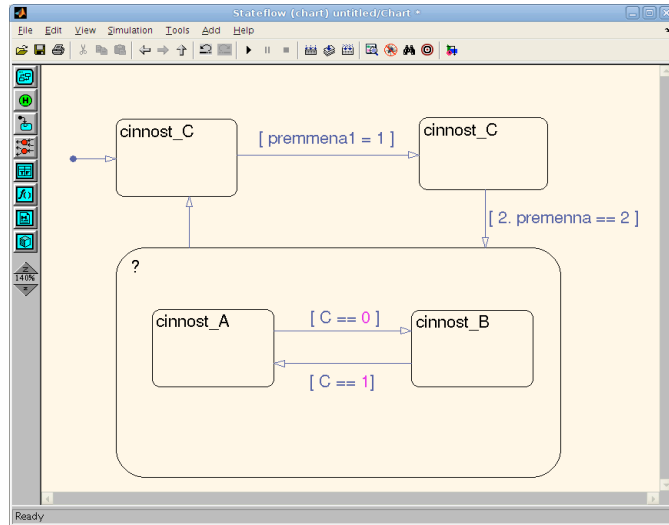
## Prečo systém vôbec nefunguje?



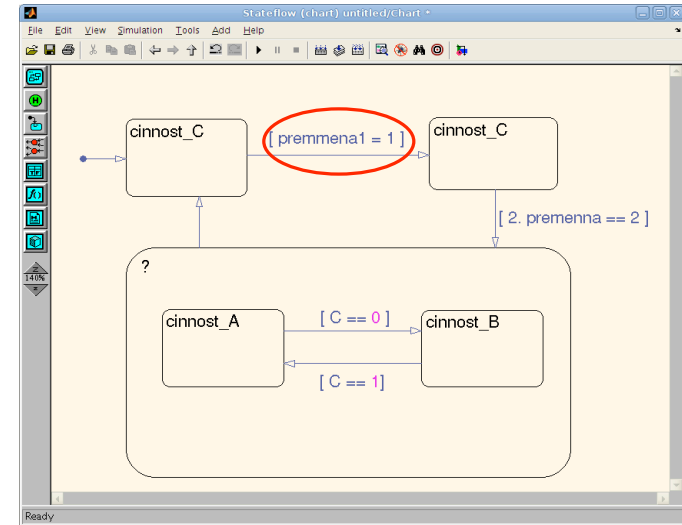
## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - všetky stavy musia byť pomenované
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

## Prečo systém vôbec nefunguje?



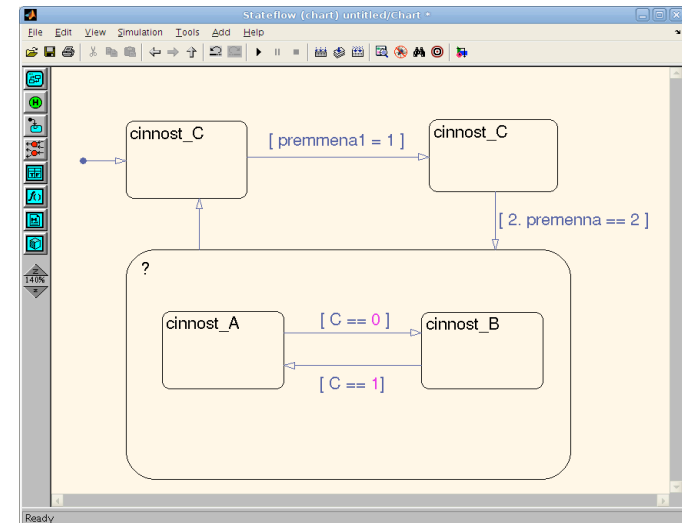
## Prečo systém vôbec nefunguje?



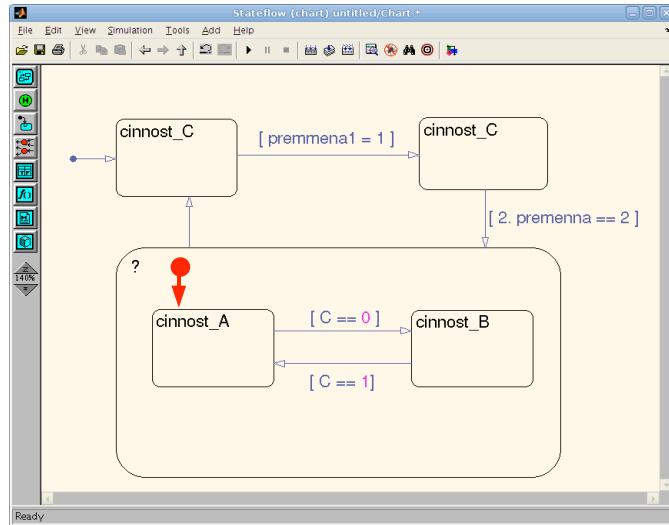
## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - všetky stavy musia byť pomenované
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

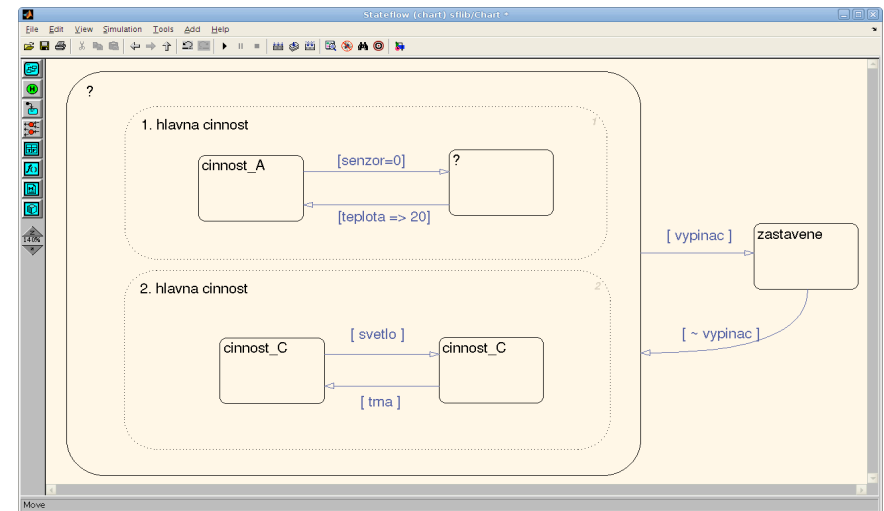
## Prečo systém vôbec nefunguje?



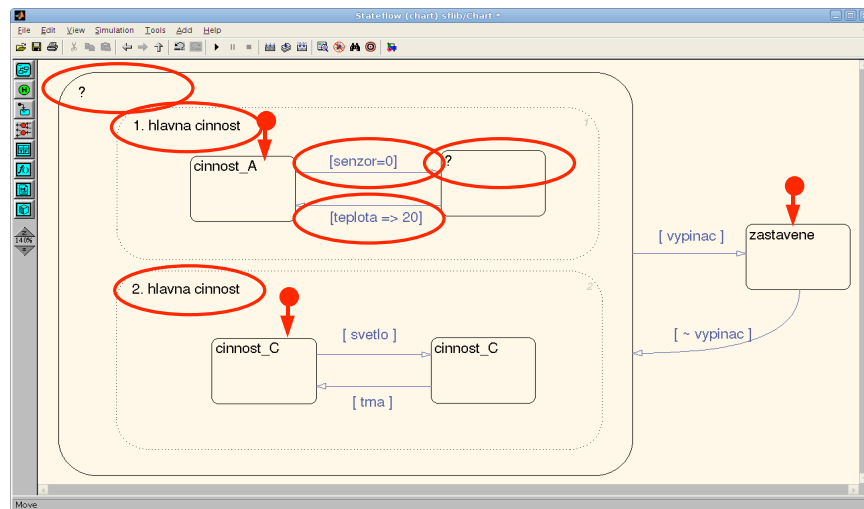
## Prečo systém vôbec nefunguje?



## Prečo systém vôbec nefunguje?



## Prečo systém vôbec nefunguje?



## Malá odbočka: podmienky

- Majme senzor **svetlo** ktorý má hodnotu 1 ak je v miestnosti dosť svetla, v opačnom prípade má hodnotu 0
- Podmienka "je dosť svetla" sa dá napísať dvojako:
  - [ svetlo == 1 ]
  - [ svetlo ]

## Malá odbočka: podmienky

- Takýto tvar podmienky v skutočnosti znamená **[ (svetlo == 1) ]**
- Podmienka v hranatých zátvorkách je splnená vtedy, keď je výraz v guľatých zátvorkách hodnotu **pravda**
- Preto **[ (svetlo == 1) ]** je v tomto prípade to isté ako **[ (svetlo) ]**
- Dôvod:
  - **[ (svetlo) ]** je pravda vtedy, keď je **svetlo** pravda.
  - to je pravda vtedy, keď má **svetlo** nenulovú hodnotu

## Malá odbočka: podmienky

- Majme senzor **hladina** ktorý vyjadruje výšku hladiny ako reálne číslo v rozsahu od 0 po 1 meter
- Podmienka “zásobník je napolovicu plný” sa dá napísať ako **[ hladina == 0.5]**
- To ale nie je to isté, ako **[ hladina ] !!!**
- Dôvod:
  - **[ hladina ]** je pravda pre **ľubovlnú** nenulovú hodnotu premennej **hladina**
  - **[ hladina == 0.5]** je pravda **iba pre jednu** jedinú hodnotu premennej **hladina**

## Malá odbočka: podmienky

- Majme senzor **hladina** ktorý vyjadruje výšku hladiny ako reálne číslo v rozsahu od 0 po 1 meter
- Podmienka “zásobník je napolovicu plný” sa dá napísať ako **[ hladina == 0.5]**
- To ale nie je to isté, ako **[ hladina ] !!!**

## Malá odbočka: podmienky

- Čo z toho plynie:
  - ak chápem podstate podmienok, za určitých okolností môžem používať skrátenejší zápis v tvare **[ svetlo ]**
  - ak sa nechcem trápiť, vždy použijem dlhší, ale bezpečnejší zápis, t.j. **[ senzor == 1 ]**
  - ak je premenná reálnu hodnotu (takmer) vždy je sa používa dlhý zápis, t.j. **[ x == 0.34 ]**

## Prečo systém vôbec nefunguje?

- Nie sú splnené formálne pravidlá softwaru, v ktorom riadiaci systém implementujeme
- Stateflow:
  - všetky stavy musia byť pomenované
  - názvy stavov a premenných majú isté pravidlá
  - musí byť dodržaná syntax jazyka
  - musia existovať štartovacie body

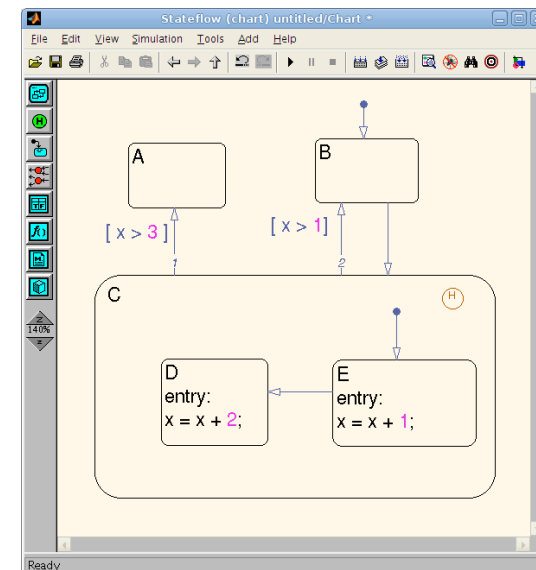
## Prečo mnou navrhnutý riadiaci systém nefunguje?

- Čo znamená “nefunguje”?
  - vôbec nefunguje
  - **nefunguje správne**
- Vôbec nefunguje:
  - prečo? (zistiť príčiny)
  - ako ho opraviť? (navrhnuť opravu)

## Prečo systém nefunguje správne?

- Nie sú správne implementované všetky pravidlá
- Ako zistiť ktoré pravidlá nie sú správne implementované?
  - musíme vedieť analyzovať beh riadiaceho systému

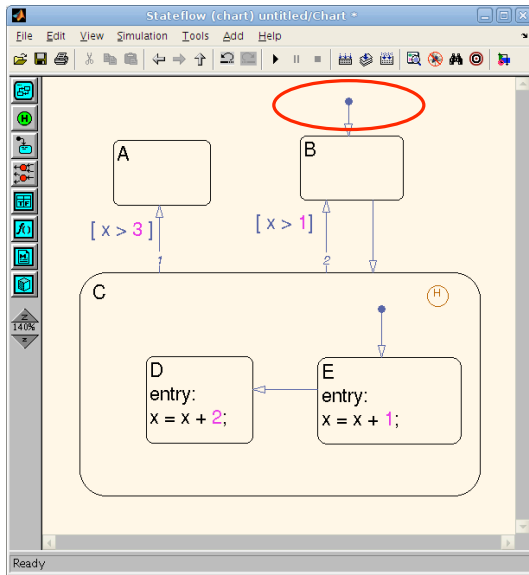
## Akú hodnotu má $x$ v stave A?





Akú hodnotu má  $x$  v stave **A**?

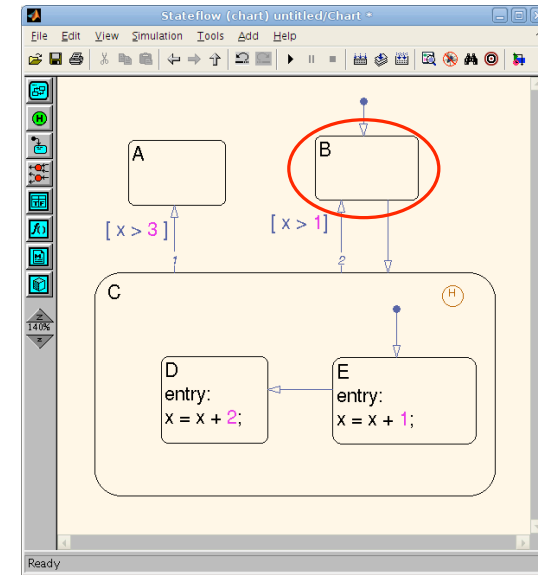
$t = 0$



$x = 0$   
Prečo?

Akú hodnotu má  $x$  v stave **A**?

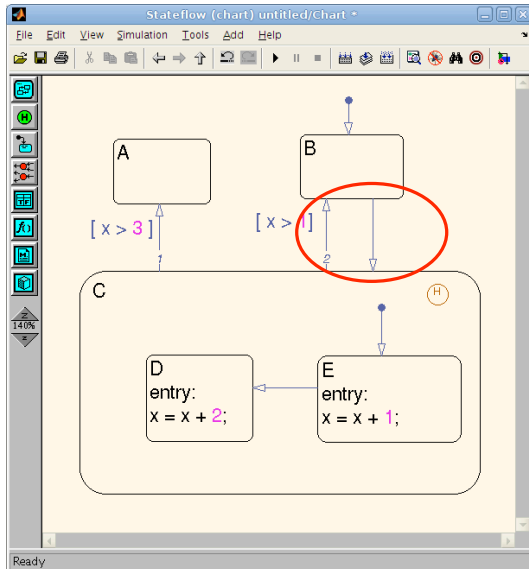
$t = 1$



$x = 0$

Akú hodnotu má  $x$  v stave **A**?

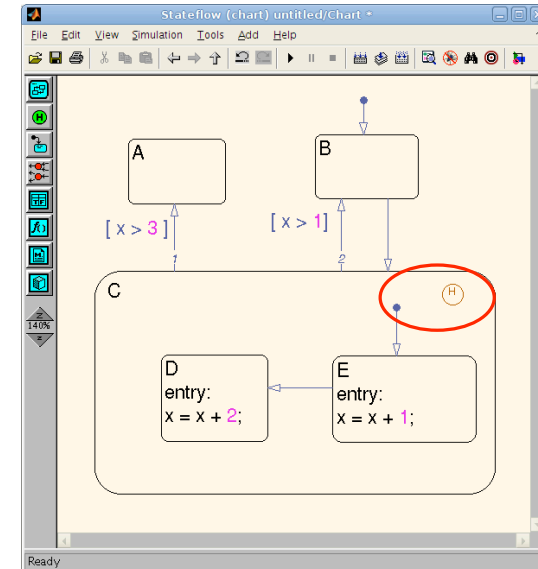
$t = 2$



$x = 0$

Akú hodnotu má  $x$  v stave **A**?

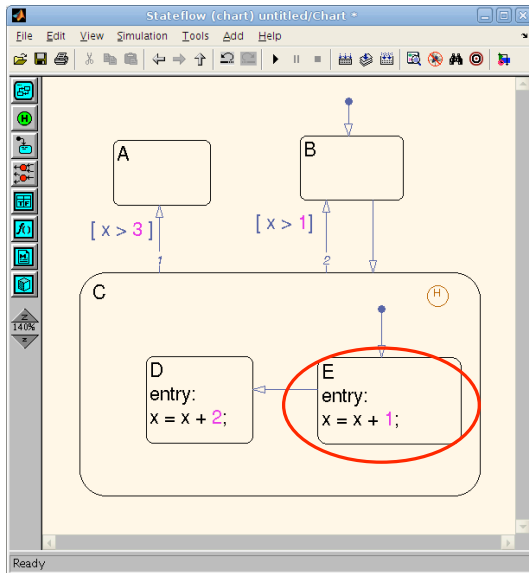
$t = 3$



$x = 0$

Akú hodnotu má  $x$  v stave **A**?

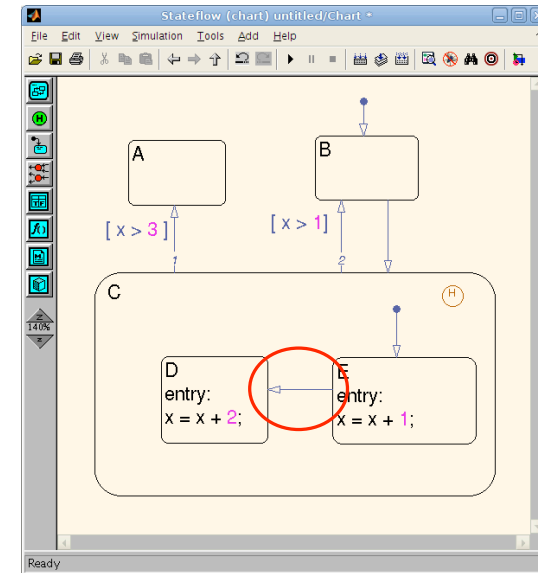
t = 4



x = 1

Akú hodnotu má  $x$  v stave **A**?

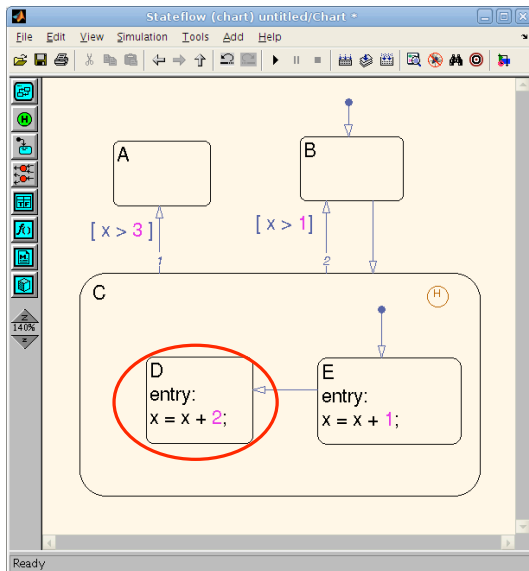
t = 5



x = 1

Akú hodnotu má  $x$  v stave **A**?

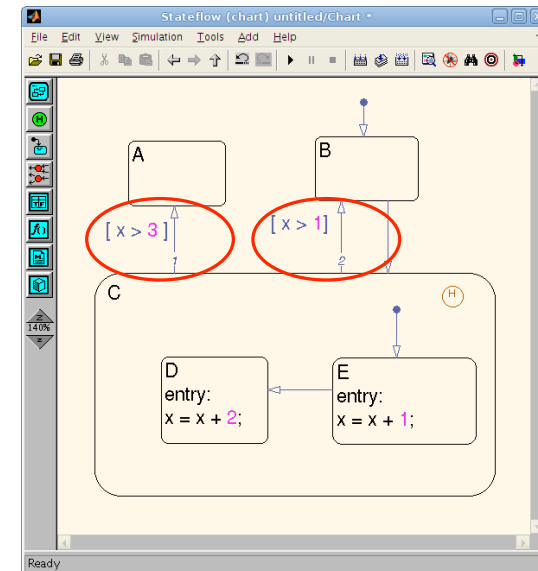
t = 6



x = 3

Akú hodnotu má  $x$  v stave **A**?

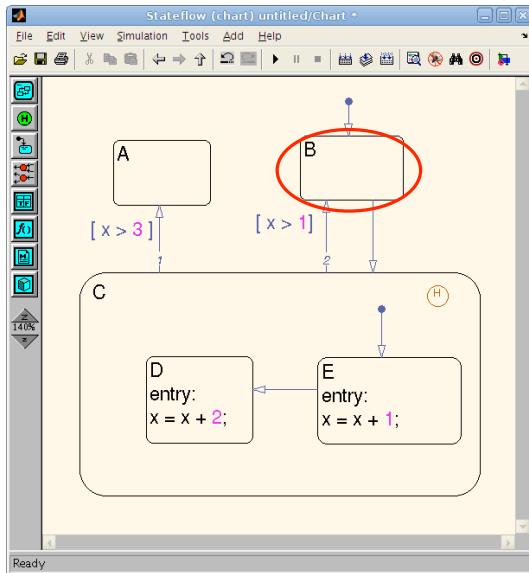
t = 7



x = 3

Akú hodnotu má  $x$  v stave **A**?

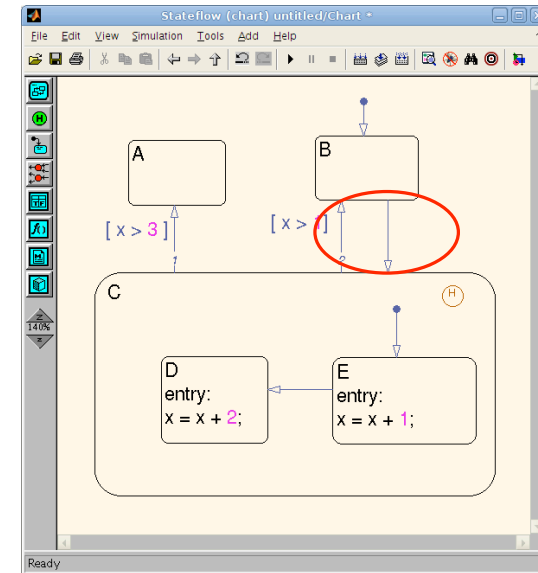
$t = 8$



$x = 3$

Akú hodnotu má  $x$  v stave **A**?

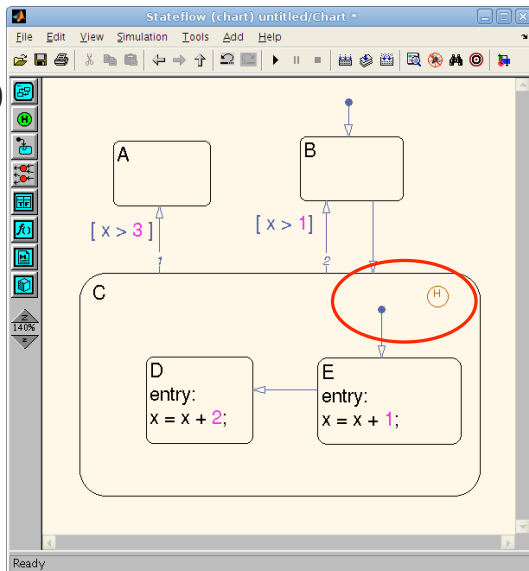
$t = 9$



$x = 3$

Akú hodnotu má  $x$  v stave **A**?

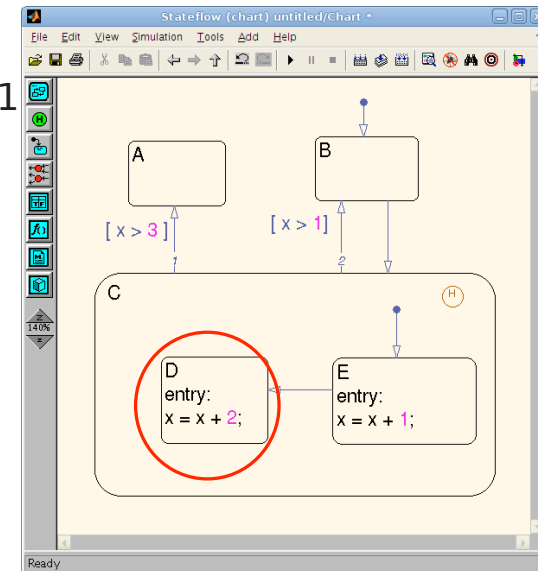
$t = 10$



$x = 3$

Akú hodnotu má  $x$  v stave **A**?

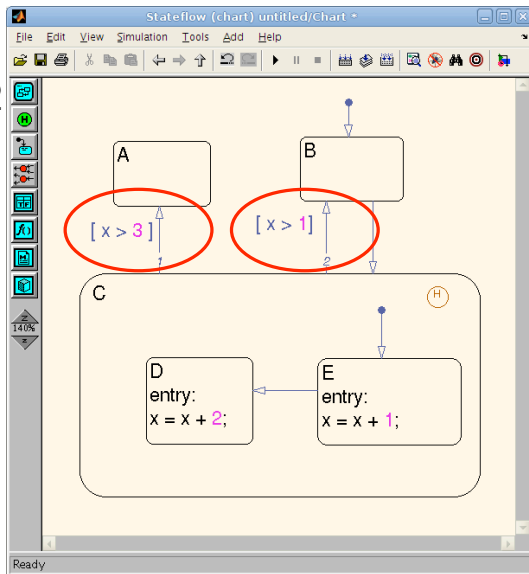
$t = 11$



$x = 5$

Akú hodnotu má  $x$  v stave **A**?

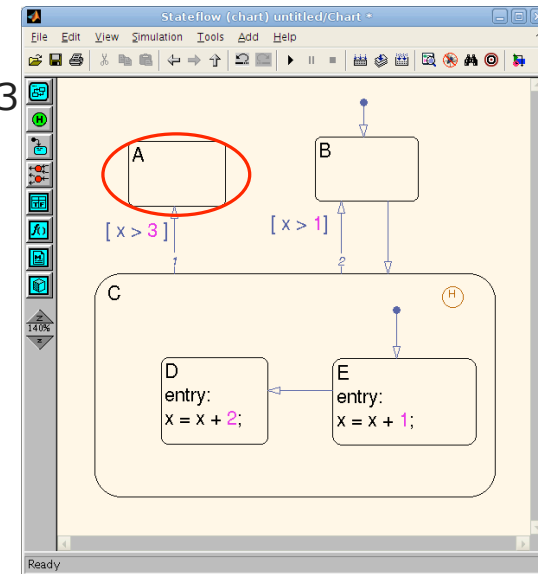
$t = 12$



$x = 5$

Akú hodnotu má  $x$  v stave **A**?

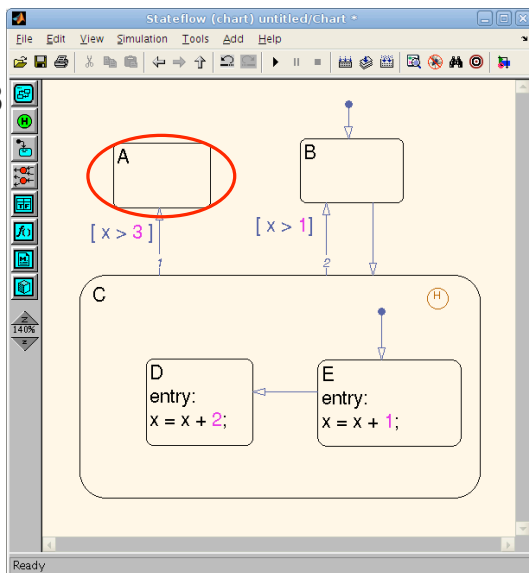
$t = 13$



$x = 5$

Akú hodnotu má  $x$  v stave **A**?

$t = 13$



$x = 5$

Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili

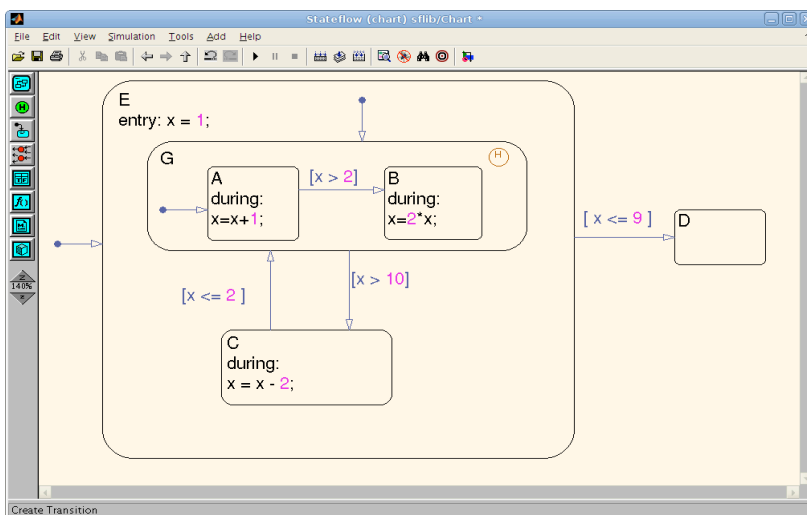
## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**

## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**
- Vyhodnotia sa výstupné podmienky
  - ak zo stavu odchádzame, vykoná sa **exit**
  - ak v stave ostávame, vykonajú sa príkazy **during**

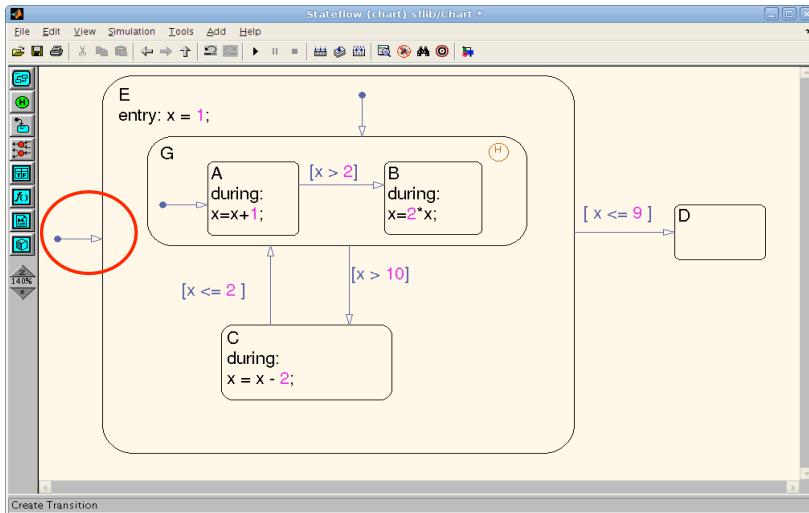
Akú hodnotu má **x** v stave **D**?



## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**
- Vyhodnotia sa výstupné podmienky
  - ak zo stavu odchádzame, vykoná sa **exit**
  - ak v stave ostávame, vykonajú sa príkazy **during**

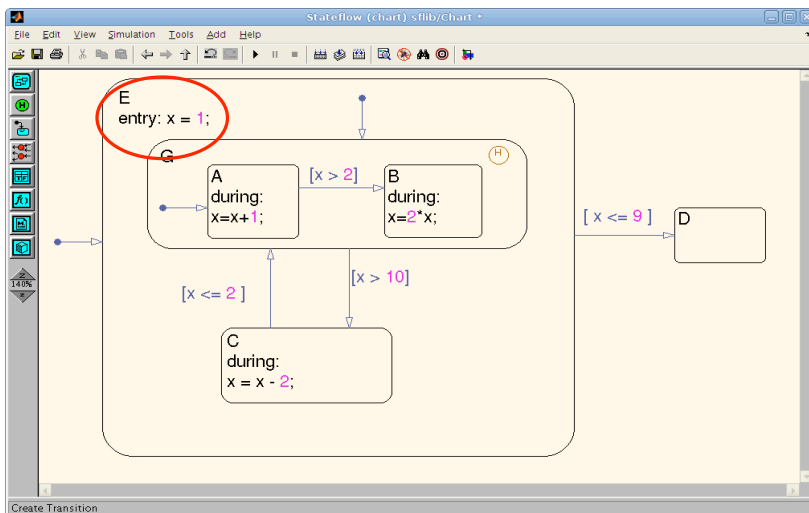
## Akú hodnotu má $x$ v stave **D**?



## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**
- Vyhodnotia sa výstupné podmienky
  - ak zo stavu odchádzame, vykoná sa **exit**
  - ak v stave ostávame, vykonajú sa príkazy **during**

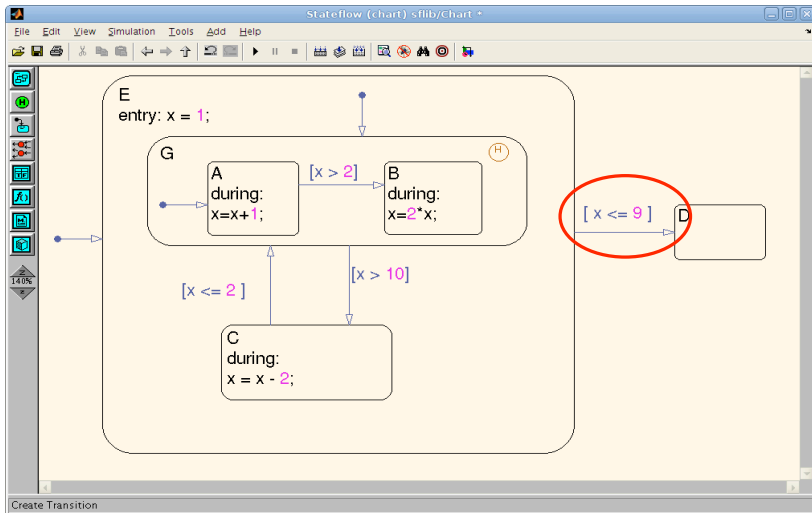
## Akú hodnotu má $x$ v stave **D**?



## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**
- Vyhodnotia sa výstupné podmienky
  - ak zo stavu odchádzame, vykoná sa **exit**
  - ak v stave ostávame, vykonajú sa príkazy **during**

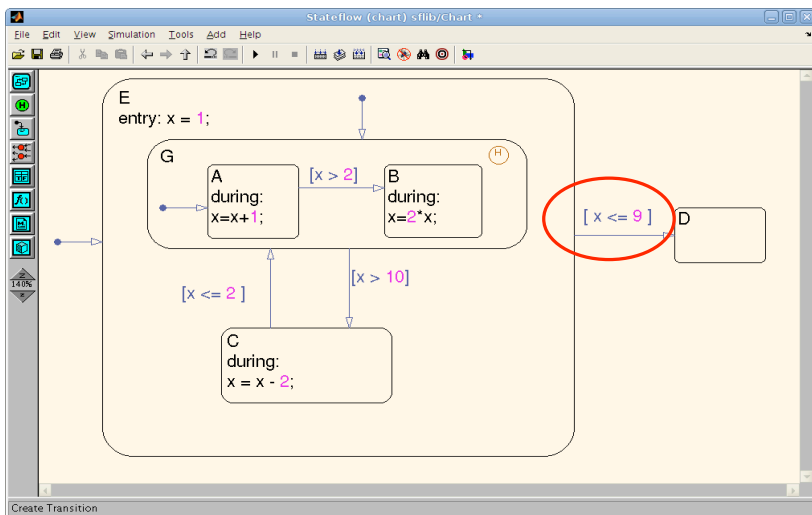
## Akú hodnotu má $x$ v stave **D**?



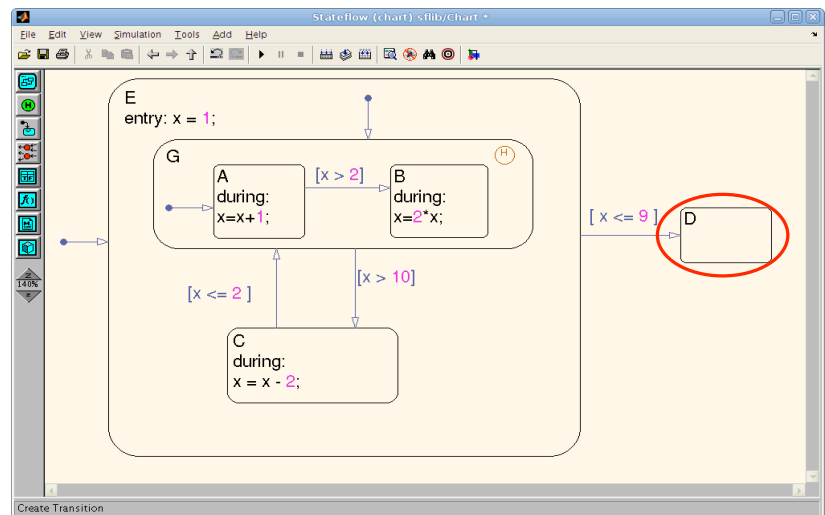
## Pravidlá behu riadiaceho systému založeného na stavoch

- Určí sa štartovací bod:
  - ak history neexistuje, navštívi sa štartovací bod
  - ak je história prázdna, navštívi sa štartovací bod
  - ak história nie je prázdna, navštívi sa stav, ktorý sme naposledy na danej úrovni opustili
- Po vstupe do stavu sa vykonajú príkazy **entry**
- Vyhodnotia sa výstupné podmienky
  - ak zo stavu odchádzame, vykoná sa **exit**
  - ak v stave ostávame, vykonajú sa príkazy **during**

## Akú hodnotu má $x$ v stave **D**?

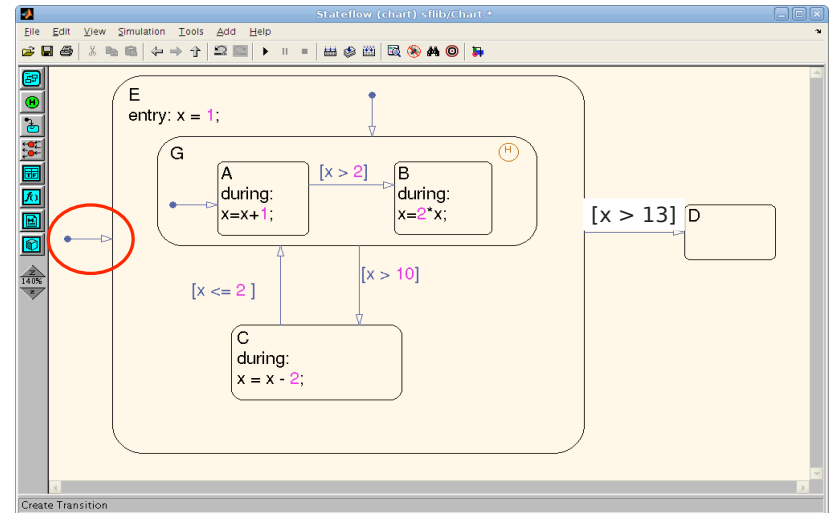
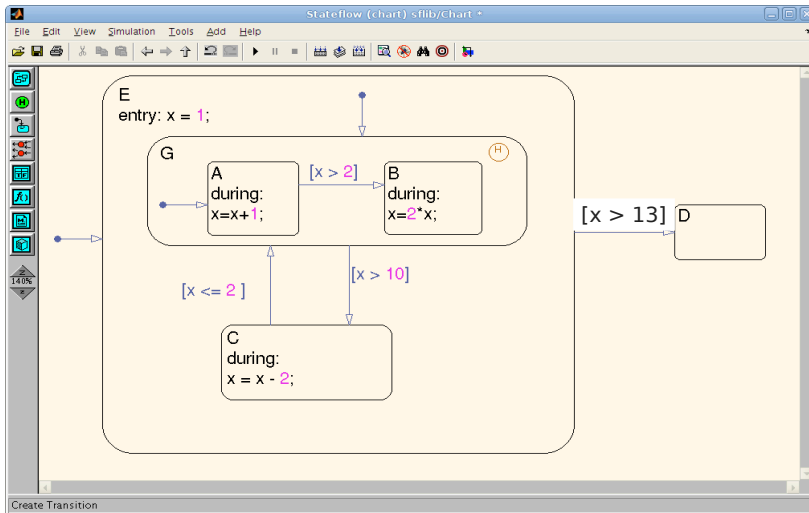


## Keď dosiahneme stav **D**, $x = 1$



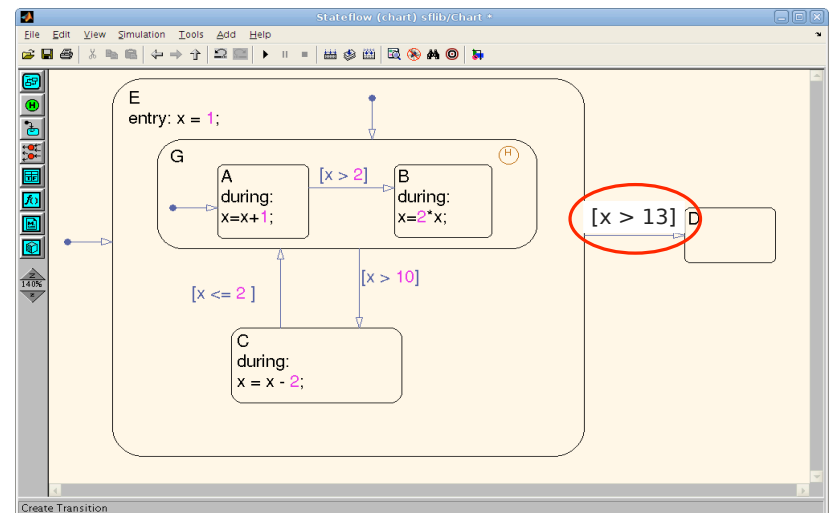
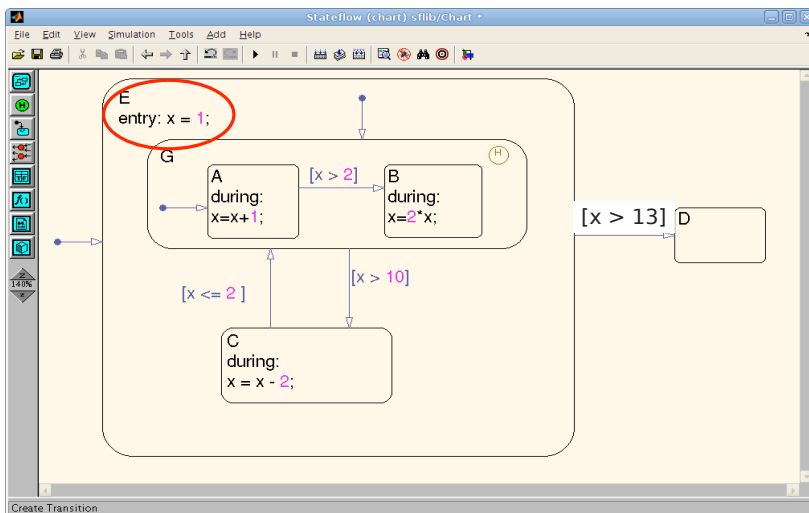
# Modifikovaný príklad

$x = 0$



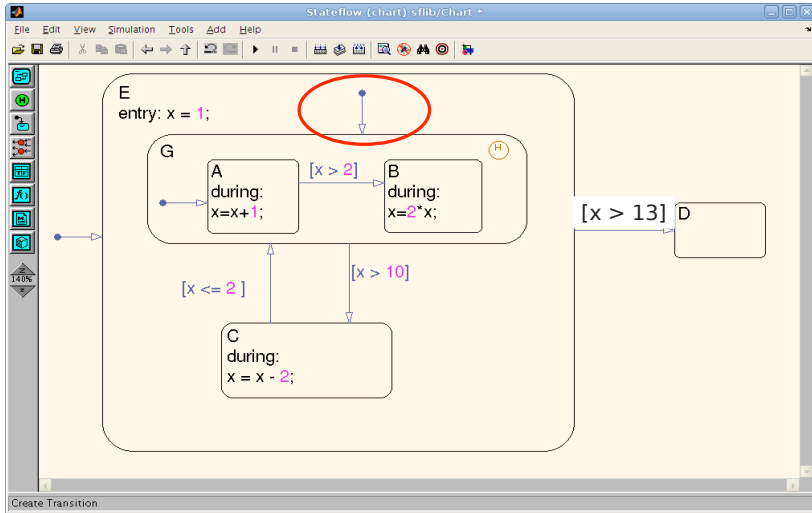
$x = 1$

$x = 1$

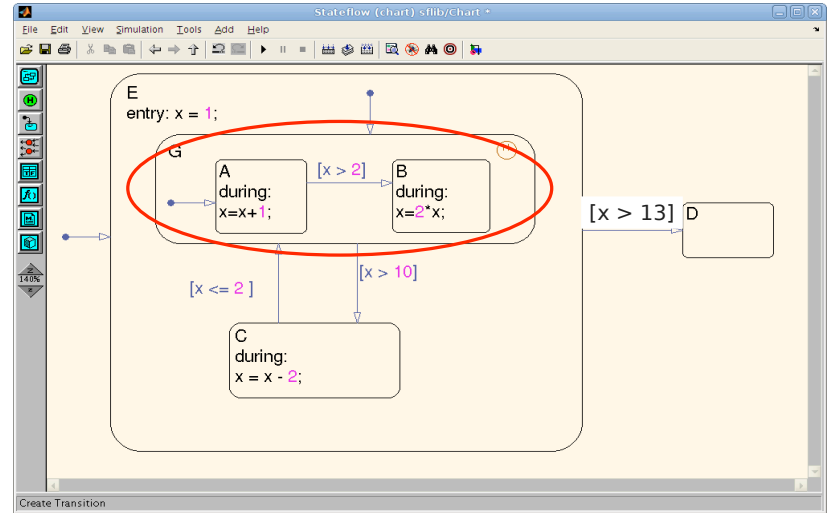




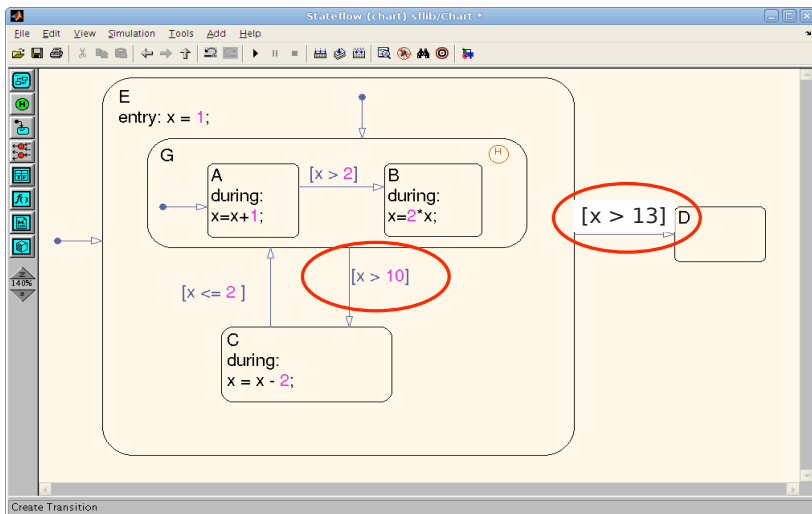
x = 1



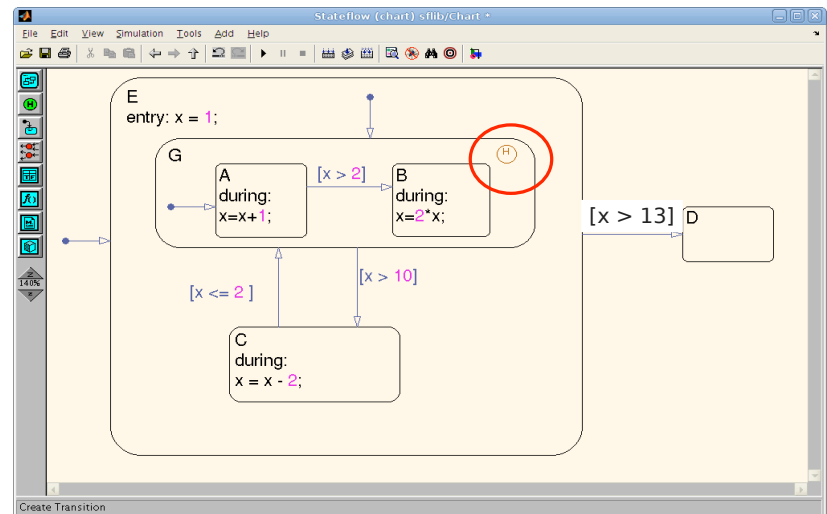
x = 1



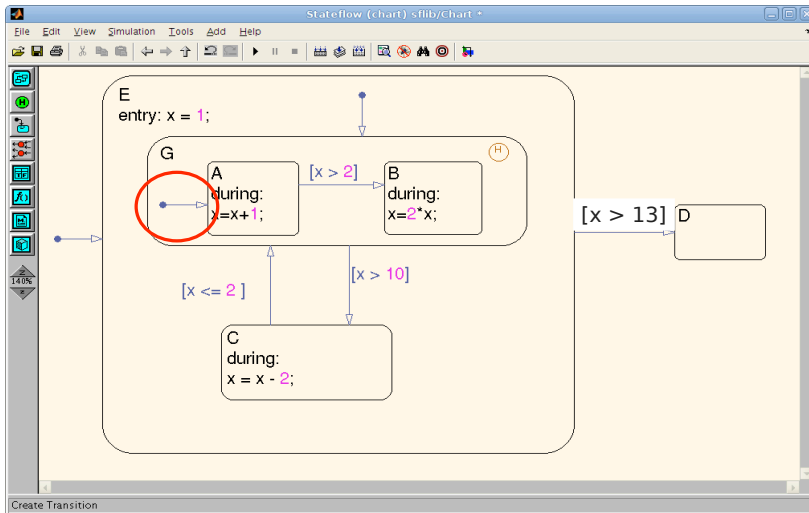
x = 1



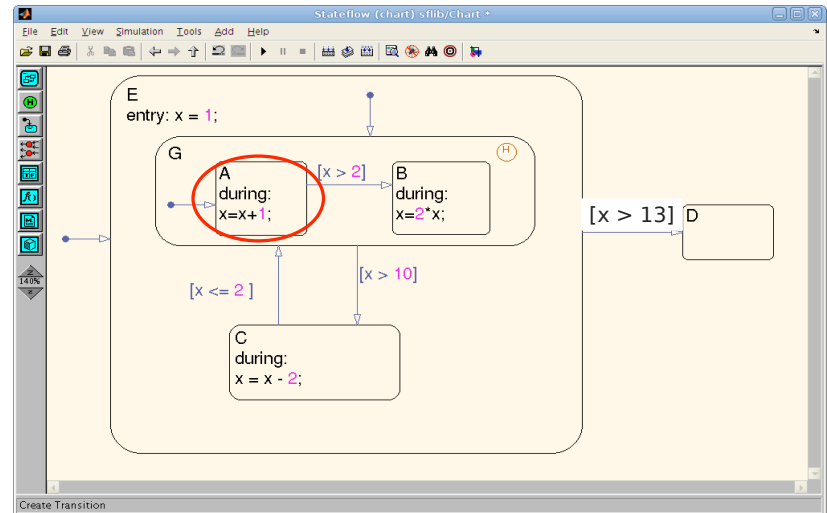
x = 1



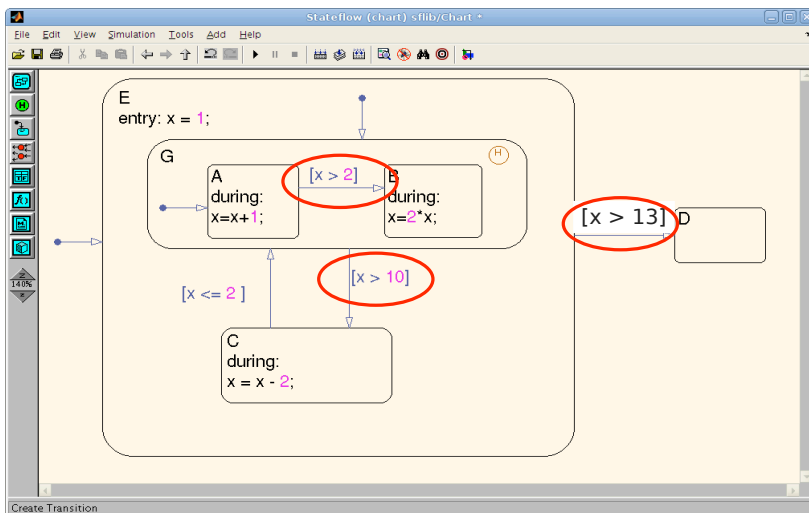
$x = 1$



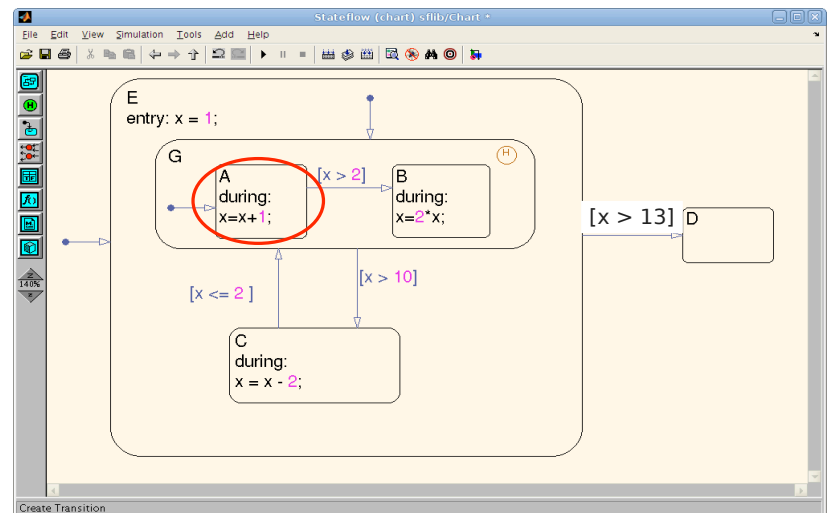
$x = 1$



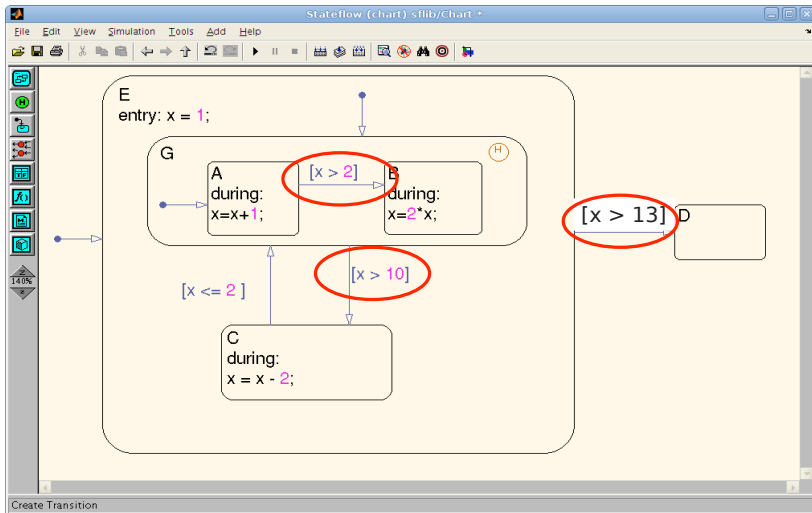
$x = 1$



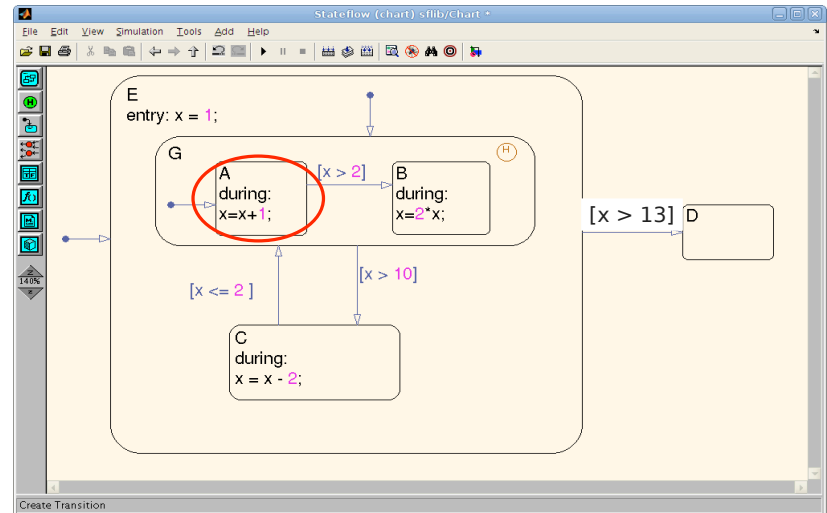
$x = 2$



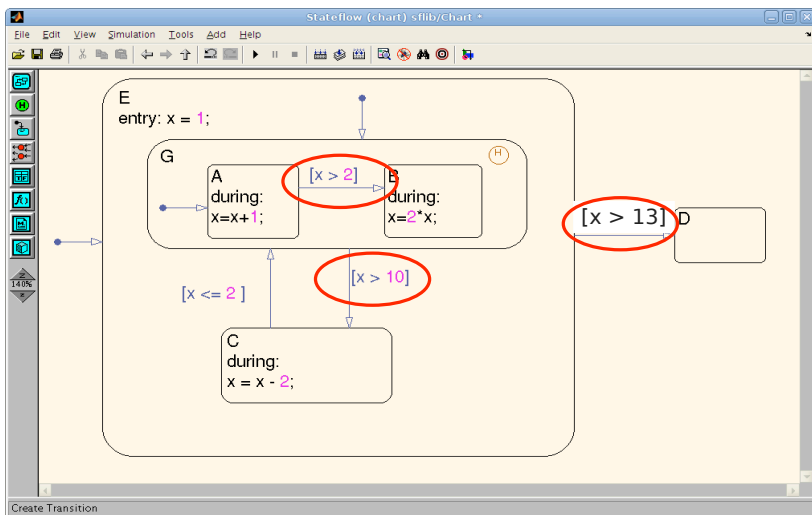
$x = 2$



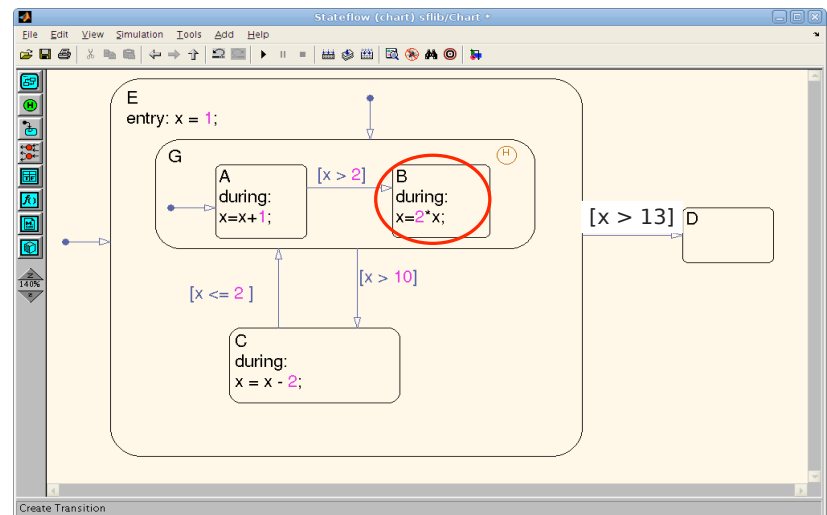
$x = 3$



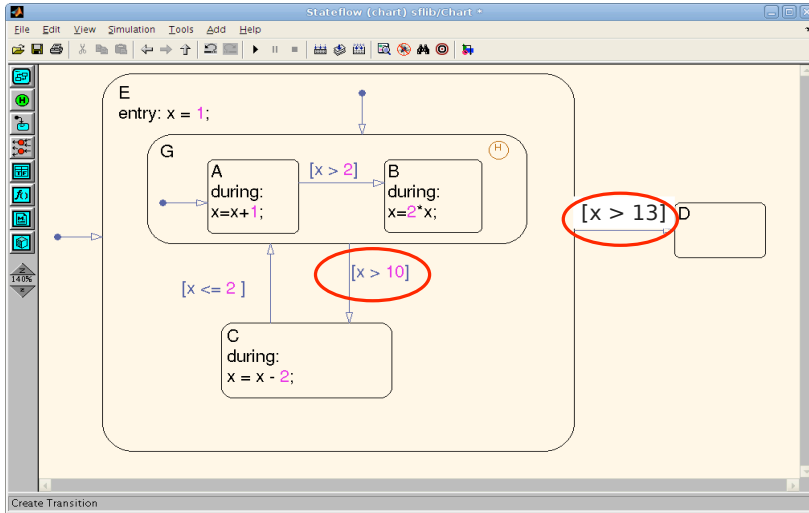
$x = 3$



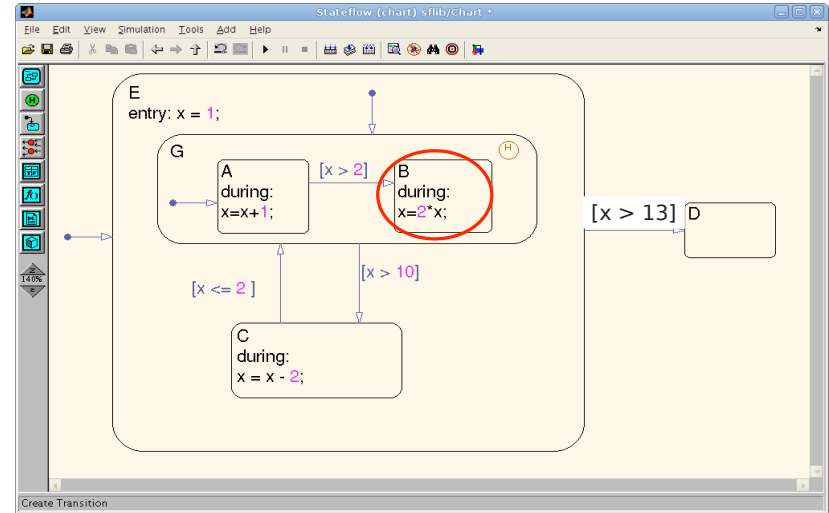
$x = 3$



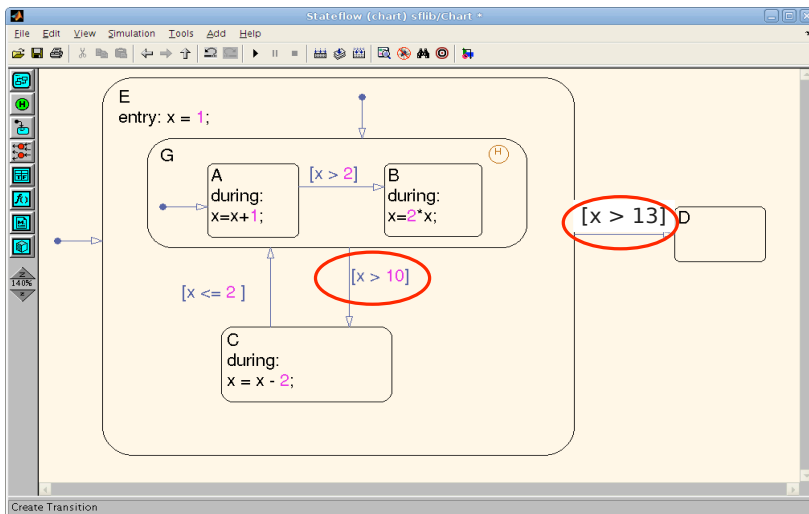
$x = 3$



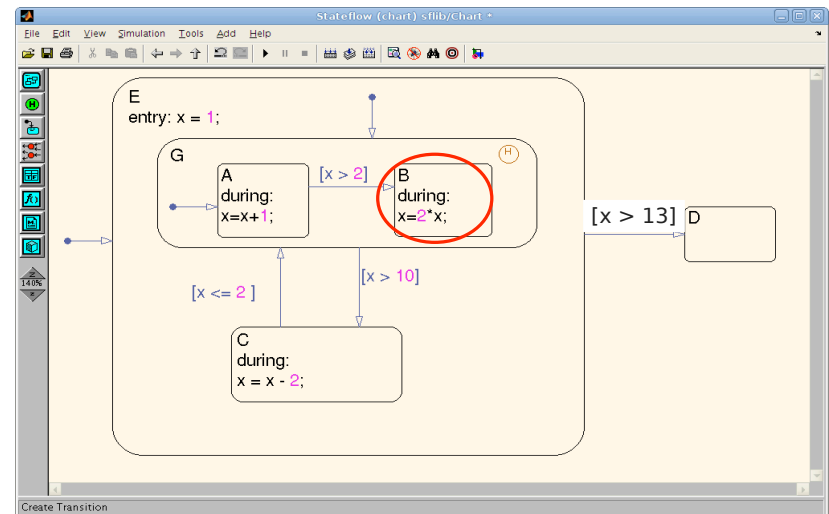
$x = 6$



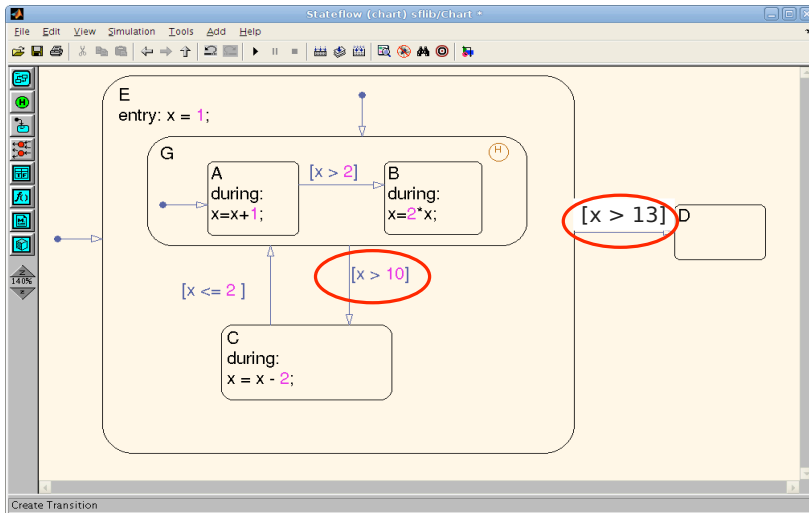
$x = 6$



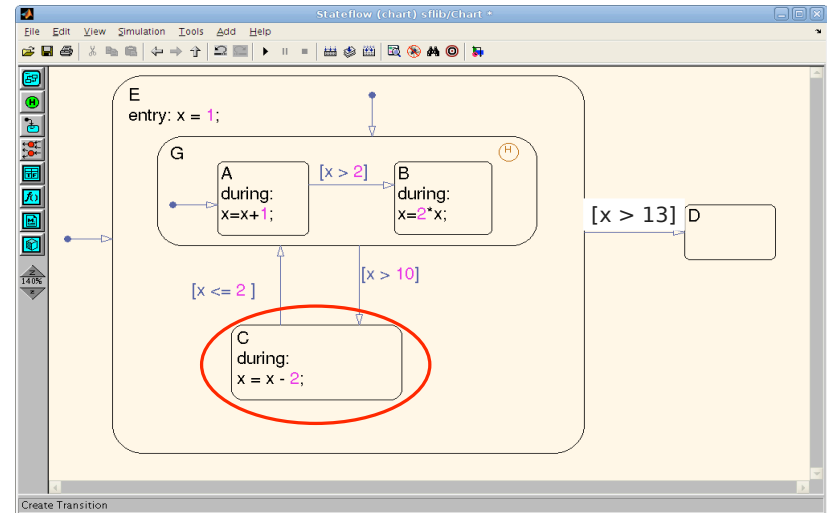
$x = 12$



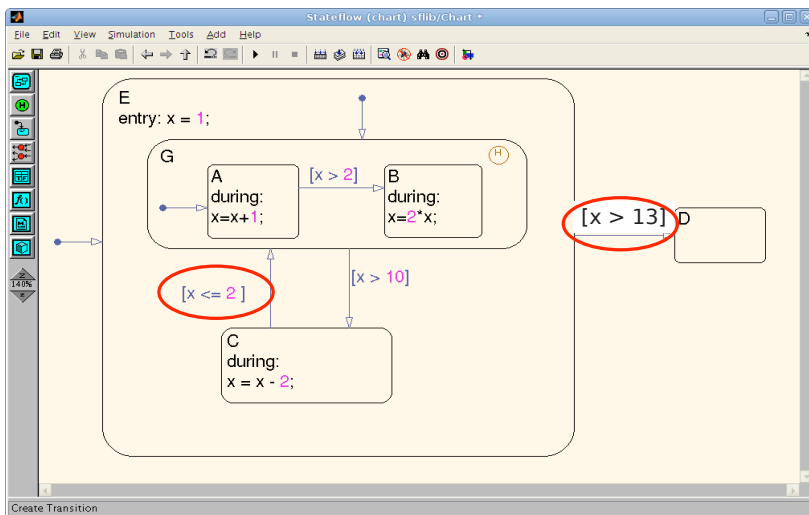
x = 12



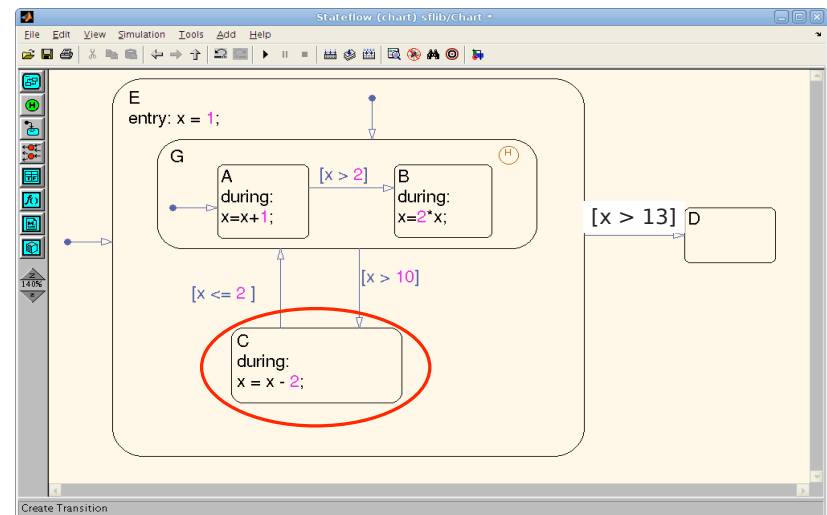
x = 12



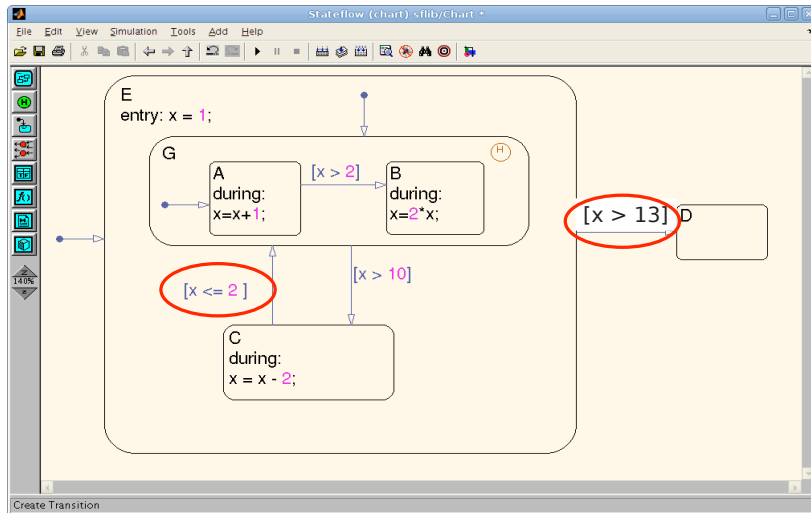
x = 12



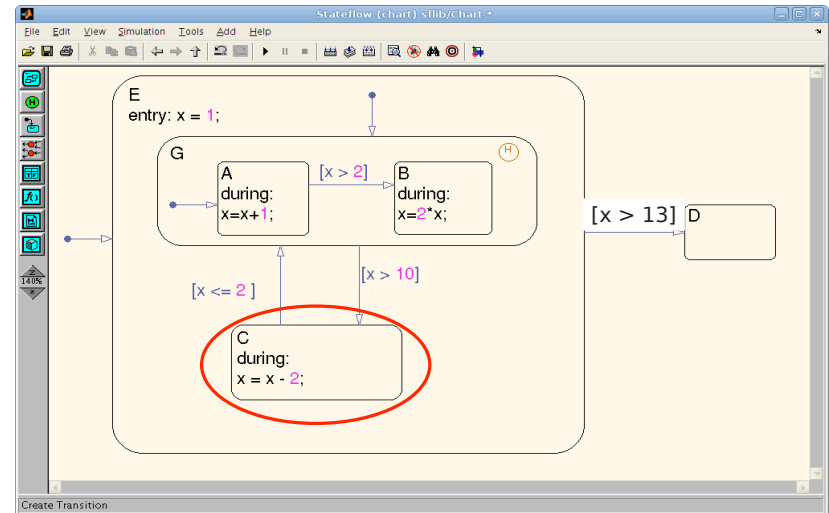
x = 10



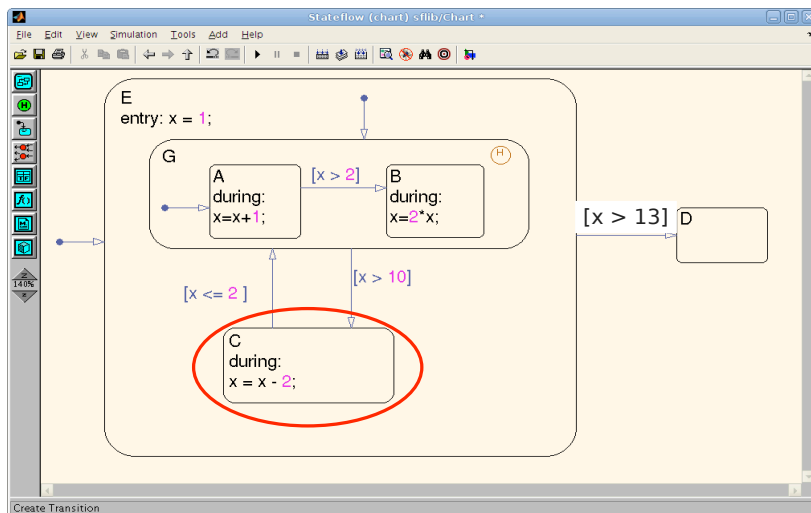
$x = 12$



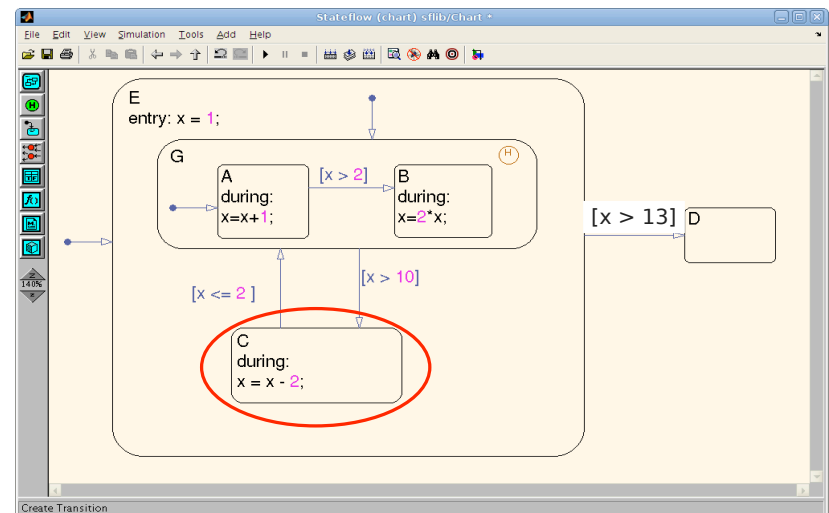
$x = 8$



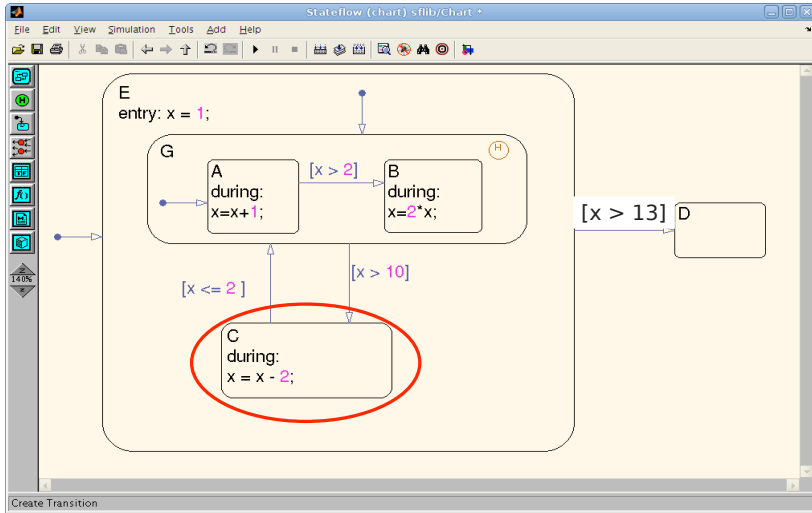
$x = 6$



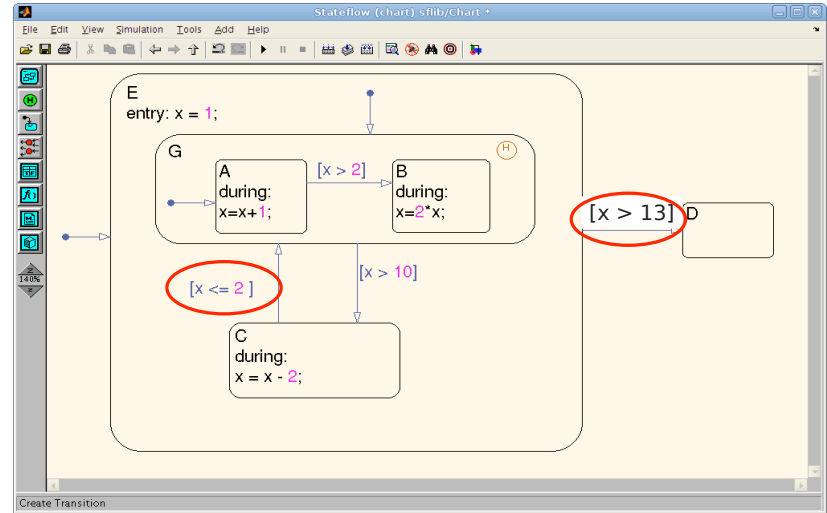
$x = 4$



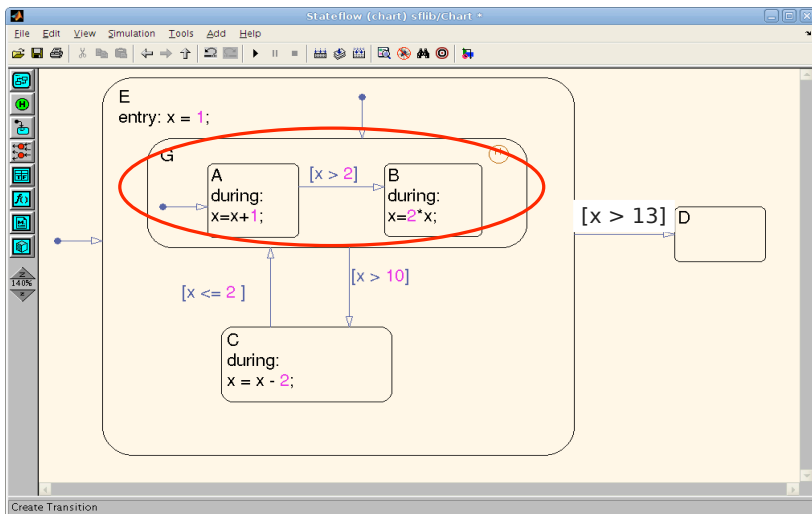
$x = 2$



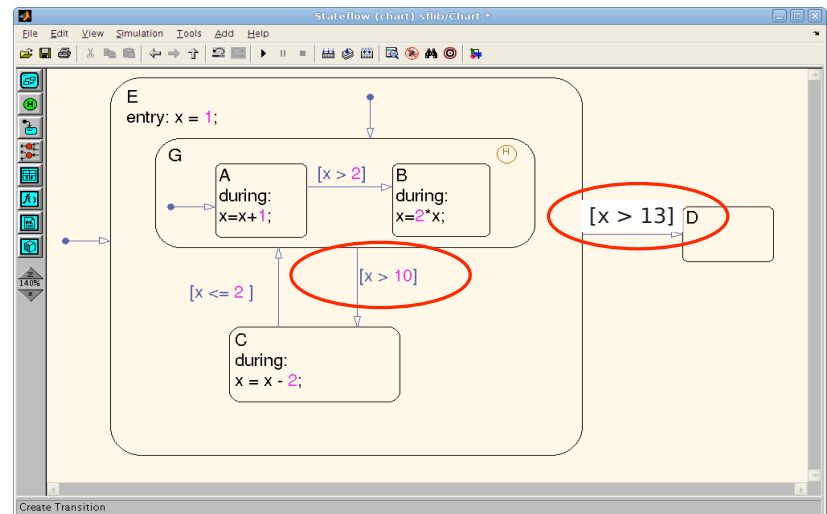
$x = 2$



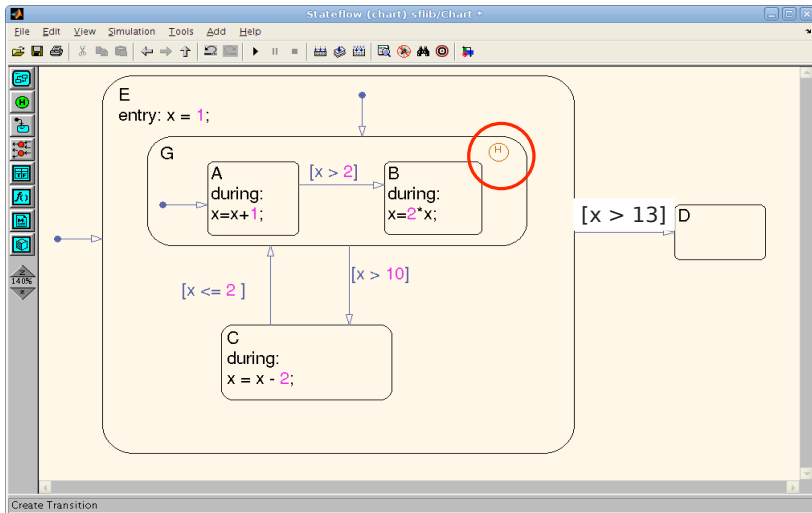
$x = 2$



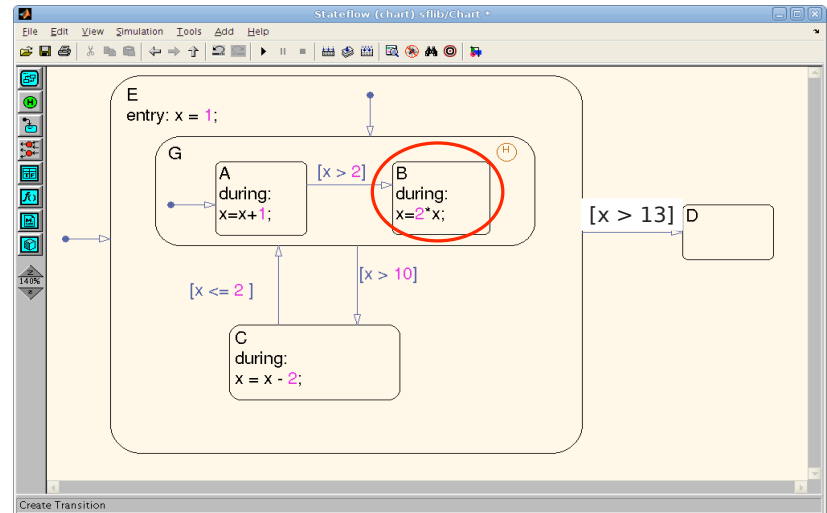
$x = 2$



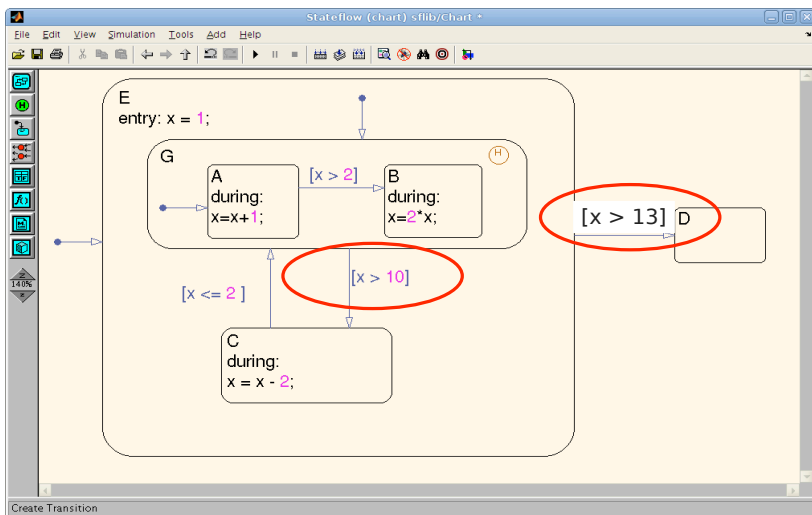
$x = 2$



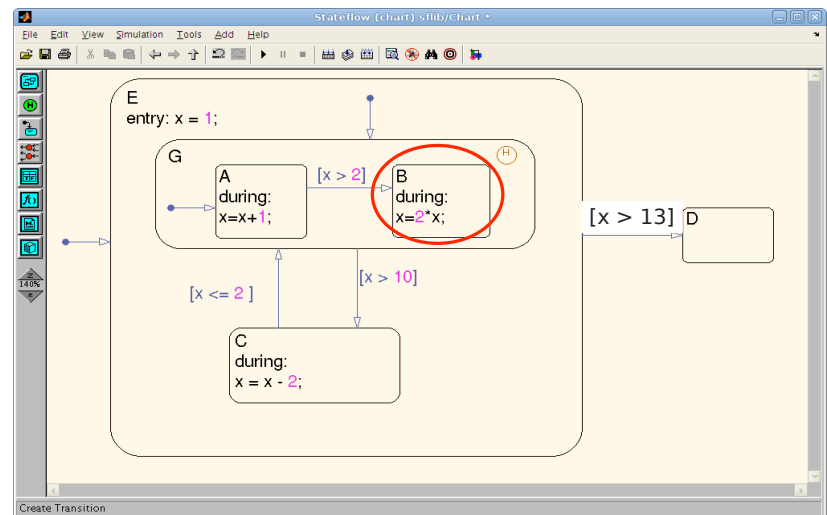
$x = 2$



$x = 2$

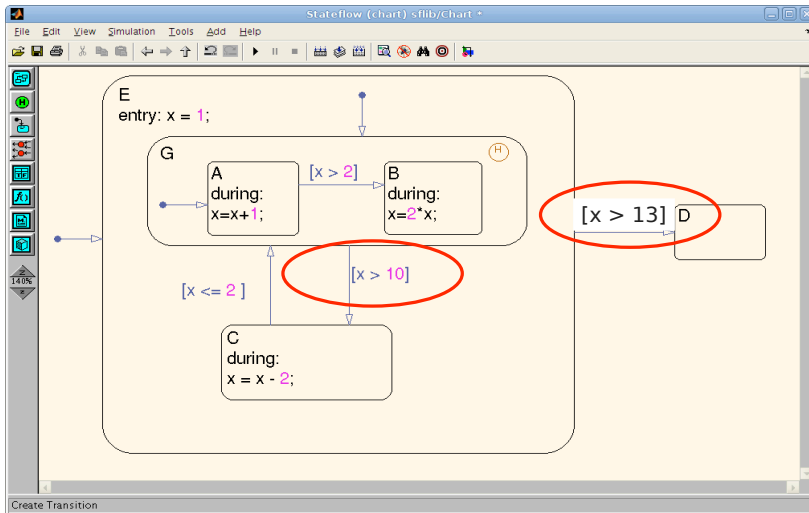


$x = 4$

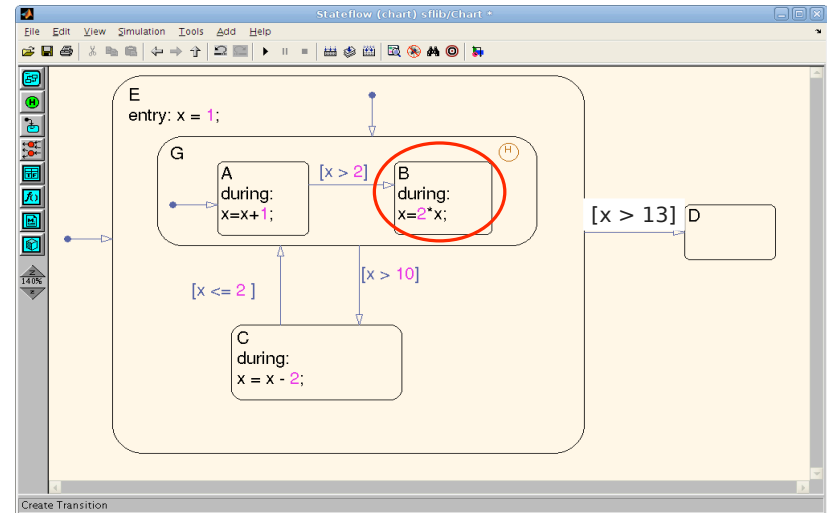




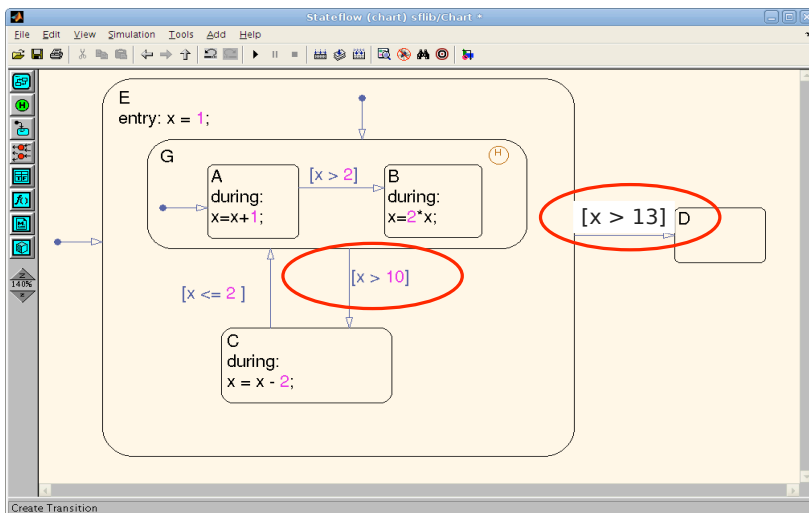
$x = 4$



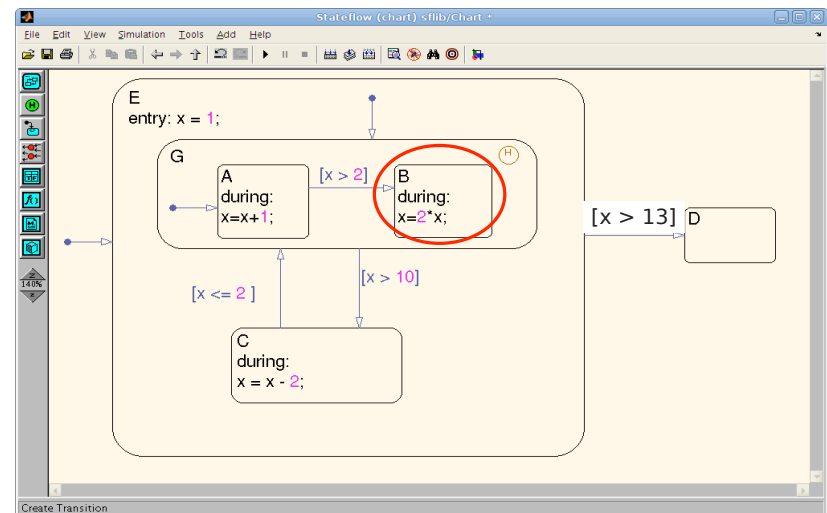
$x = 8$



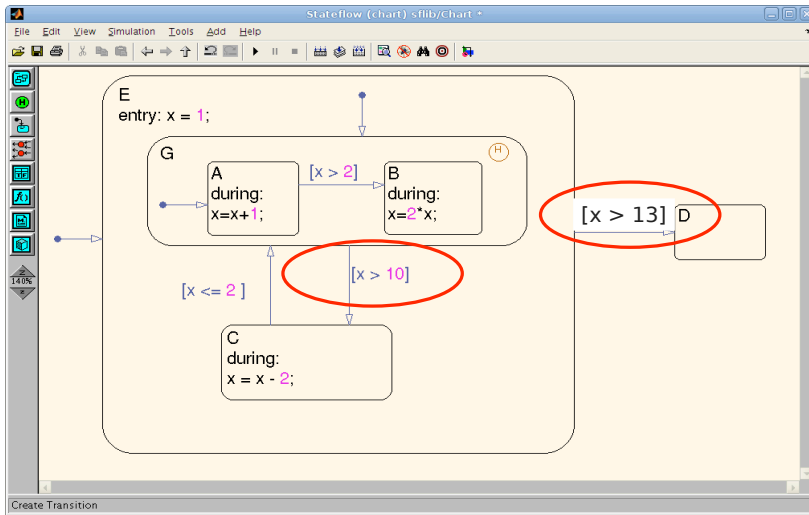
$x = 4$



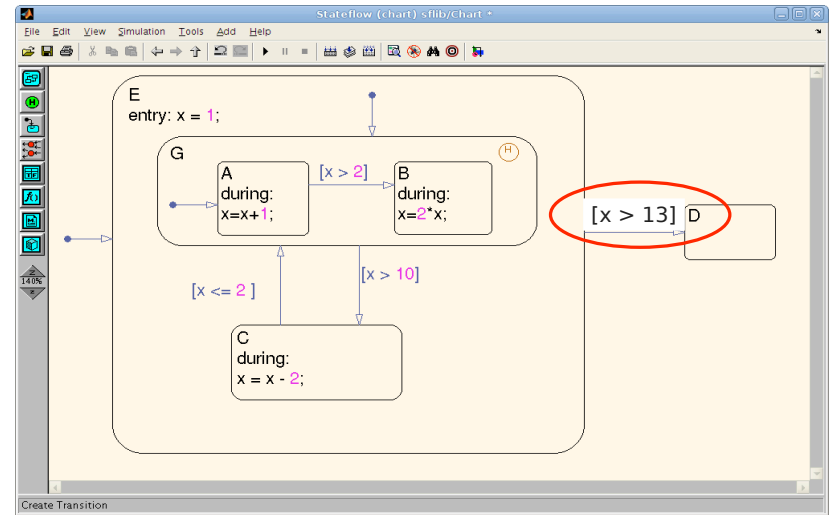
$x = 16$



x = 16



x = 16



x = 16

