# Modely systémov s diskrétnymi udalosťami v toolboxe Stateflow

## Úlohy

- 1. Zostavte matematické modely klopného obvodu RS a semaforu.
- 2. Navrhnite simulačné modely klopného obvodu RS a semaforu v toolboxe Stateflow.
- Vykonajte simulácie vytvorených simulačných modelov a prostredníctvom priebehov stavov overte ich správnosť.

### Zostavenie matematických modelov klopného obvodu RS a semaforu

#### Klopný obvod RS

Klopný obvod pozostáva z dvoch vstupov - *R* a *S*, a dvoch výstupov - *Q* a jeho negácie *Q non*. Stačí uvažovať 1 výstup *Q*. Vstup *S* (set) reprezentuje vstup klopného obvodu, ktorý ho nastavuje do stavu logickej jednotky a *R* (reset) reprezentuje vstup, ktorý ho nastavuje do stavu logickej nuly. Princíp činnosti klopného obvodu RS:

- ak je hodnota oboch vstupov R a S nulová, tak klopný obvod nemení svoj stav
- ak je hodnota nastavovacieho vstupu S=1 a hodnota nulovacieho vstupu R=0, klopný obvod prechádza do stavu Q=1 a v tomto stave ostáva aj po zmene hodnoty vstupu S,
- ak je hodnota nulovacieho vstupu R=1 a hodnota nastavovacieho vstupu S=0, klopný obvod zmení svoj stav na stav Q=0 a v tomto stave ostáva aj v po zmene vstupu R,
- ak R=1 a S=1, uvažujeme, že klopný obvod nemení svoj stav

Matematický model klopného obvodu sformulujeme vymenovaním množiny stavov, udalostí, určením počiatočného stavu systému a definovaním funkcií prechodov.

Množina stavov:	$X = \{0, 1\}$		
Množina udalostí:	$E = \{e_1, e_2, e_3, e_4,\}$ , kde $e_1 - R=0$ a S=1, $e_2 - R=1$ a S=0,		
	$e_3 - R=0$ a S=0, $e_4 - R=1$ a S=1.		
Počiatočný stav:	$x_0 = 0$		
Funkcie prechodov:	$f(0, e_1) = 1, f(0, e_2) = 0,$		
	$f(0, e_3) = 0, f(0, e_4) = 0,$		

$$f(1, e_1) = 1, f(1, e_2) = 0,$$
  
 $f(1, e_3) = 1, f(1, e_4) = 1.$ 

Následne si zakreslíme tento matematický model do stavového diagramu.



Obr. 1 Stavový diagram klopného obvodu RS

#### Semafor

Semafor zobrazuje svetelné signály troch farieb - červenej, oranžovej a zelenej. Podľa toho, akú farbu semafor zobrazuje rozlišujeme štyri stavy:

- Stoj ak svieti len červená, zakazuje prechod.
- Priprav svieti červená aj oranžová naraz, vodič sa môže pripraviť na jazdu.
- Choď svieti len zelené svetlo, vodič môže pokračovať v jazde.
- Pozor signál len s oranžovým svetlom, upozorňuje vodiča, že sa signál mení zo zelenej do červenej.

Okrem týchto štyroch stavov uvažujeme ešte dva ďalšie – vypnutý semafor a zapnutý. Medzi týmito dvoma stavmi prechádza semafor na základe stavu vypínača. Ak je semafor zapnutý, tak nadobúda vyššie vymenované stavy. Tie sa menia cyklicky v čase. V prípade, že je semafor vypnutý tak bliká iba oranžové svetlo.

Matematický model semaforu sformulujeme rovnako ako pri klopnom obvode RS.

Množina stavov:
$$X = \{vypnutý, zapnutý\}$$
Množina udalostí: $E = \{e_1, e_2\}$ , kde  $e_1 - vypínač = 1$ ,  $e_2 - vypínač = 0$ ,

Počiatočný stav:	x <sub>0</sub> = vypnutý,
Funkcie prechodov:	$f(vypnutý, e_1) = zapnutý,$
	$f$ (zapnutý, $e_2$ ) = vypnutý.

Tiež si tento model zakreslíme do stavového diagramu.



Obr. 2 Stavový diagram semaforu

V tomto príklade však uvažujeme ešte aj to, že ak je semafor zapnutý tak svetlá na ňom sa menia. Túto situáciu si tiež opíšeme stavovým diagramom.



Obr. 3 Stavový diagram zapnutého semaforu

Navrhnite simulačné modely klopného obvodu RS a semaforu v toolboxe Stateflow

#### Návod na vytvorenie simulačného modelu v toolboxe Stateflow

**Krok 1** - vytvoríme si nový Stateflow "chart" v Matlabe, buď ako je znázornené na Obr. 4, alebo príkazom **sfnew** zadaným do príkazového riadku Matlabu



Obr. 4 Otvorenie nového Stateflow "chartu"

Krok 2 – ak sa otvorí model v Simulinku otvoríme "Chart"





**Krok 3** – na základe stavového diagramu a matematického modelu systému vytvoríme toľko stavov koľko potrebujeme, stavy reprezentujú funkčné bloky na Obr. 6.



Obr. 6 Funkčné bloky reprezentujúce stavy v toolboxe Stateflow

FEI

**Krok 4** – stavy nazveme a určíme aké premenné sa zmenia pri vstupe do nich a výstupe z nich.



Obr. 7 Popis stavov

**Krok 5** – prepojíme stavy funkciou prechodu a určíme za akých podmienok sa má prechádzať medzi nimi. Určíme počiatočný stav. Na Obr. 8 je označená funkcia, ktorou označíme počiatočný stav. Podmienky musia byť zapísané v hranatých zátvorkách.



Obr. 8 Pridanie prechodov medzi stavmi

Krok 6 – v tomto kroku je potrebné definovať jednotlivé premenné. To urobíme kliknutím

na tlačidlo ma hornej lište otvoreného okna. Ďalší postup ako pridáme premenné je na Obr. 9.

Elle Edit Man Teolo A	Model Explorer
Kliknúť na "Add Dat	a" <u>&gt; ∮ ≨ fx ⊙ </u> → <b>*</b> ➡ ➡ ➡ = ➡
Search: by Name	V Name:
Model Hierarchy 🧷 🙉	Contents of:titled/Chart (only) Filter Contents Data premenna
<ul> <li>Simulink Root</li> <li>Base Workspace</li> <li>Nuntitled*</li> </ul>	Column View: Stateflow  Show Details 1 of 6 object(s)  Ame: premenna
Gonfiguration (A Configuration (A Configuration (A	Name Scope Port Resolve Signal DataType Size prem Local double Data mus Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Constant Const
Advice for untitle	Vyplniť údaje o premennej
	Type: double V >
	Initial value: Expression
	Minimum: Maximum:
	<
< >	Contents Search Results

Obr. 9 Definovanie premenných

Definované premenné potom vyzerajú ako na Obr. 10. To sa premietne aj do celého Stateflow "Chartu". Keďže sme si nadefinovali 1 vstupnú a 2 výstupné premenné tak Stateflow "Chart" bude vyzerať ako na Obr. 11.

	Name	Scope	Port	Resolve Signal	DataType	Size
010	premenna	Output	1		double	
101 010	vstup	Input	1		double	
010	premenna2	Output	2		double	

Obr. 10 Definované premenné



Obr. 11 Stateflow "Chart" s premennými

Krok 7 – Model systému v toolboxe Stateflow je vytvorený. Ďalej určíme, čo je vstupom systému a zobrazíme alebo uložíme výstupy. Vstupom môže byť napríklad konštanta

12

a výstupy môžeme zobraziť graficky. Tlačidlom si môžeme vyberať z knižnice funkčných blokov Simulinku. Vidíme ju na Obr. 12. Dokončený simulačný model môže vyzerať ako na Obr. 13.



Obr. 12 Knižnica funkčných blokov Simulinku



Obr. 13 Ilustračný príklad kompletného simulačného modelu

Krok 8 – v tomto kroku nakonfigurujeme parametre pre simuláciu modelu. Konfiguráciu otvoríme pomocou tlačidla , ktoré sa nachádza na hornej lište okna simulačného modelu. V okne, ktoré sa zobrazí zmeníme nastavenie v záložkách "Solver" a "Diagnostics". Správna konfigurácia je na Obr. 14 a Obr. 15.

G.	Configuration Parameters: krizovatka2_final/Co	onfiguration (Active)	×
Select: Solver Data Import/Export Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation HDL Code Generation	Simulation time Start time: 0.0 Solver options Type: Fixed-step Fixed-step size (fure sample time): Tasking and sample ons Periodic sample time constraint: Tasking mode for periodic sample times: Automatically handle rate transition for data transfer Hidder priority value indicates biologe task priority.	Stop time: inf Solver: discrete (no continuous states) auto Unconstrained Auto	

Obr. 14 Konfigurácia záložky "Solver"

\$	Configuration Parameters:	krizovatka2_final/Configuration (Active)	×
Select:	Solver		^
Solver Data Import/Export	Algebraic loop:	warning •	
Optimization     Discussion	Minimize algebraic loop:	warning •	
Hardware Implementation	Block priority violation:	warning •	
Model Referencing     Simulation Target	Min step size violation:	warning •	
<ul> <li>Code Generation</li> <li>HDL Code Generation</li> </ul>	Sample hit time adjusting:	none -	
	Consecutive zero crossings violation:	error 🔻	
	Unspecified inheritability of sample time:	warning -	
	Solver data inconsistency:	none 🔻	
	Automatic solver parameter selection:	none	
	Extraneous discrete derivative signals:	error	
	State name clash:	none	
	SimState interface checksum mismatch:	warning •	
	SimState object from earlier release:	error 🔹	

KKUI

Obr. 15 Konfigurácia záložky "Diagnostics"

#### Simulačný model klopného obvodu RS

Kompletný simulačný model klopného obvodu RS vytvorený v Simulinku je na Obr. 16. Riadenie systému zabezpečené subsystémom vytvoreným v toolboxe Stateflow uvádzame na Obr. 17.



Obr. 16 Simulačný model klopného obvodu RS

8



Obr. 17 Subsystém simulačného modelu klopného obvodu RS v toolboxe Stateflow Premenné simulačného modelu sú na Obr. 18.

	Name	Scope	Port	Resolve Signal	DataType	Si
101 010	r	Input	1		double	
101 010	stav	Output	1		double	
101 010	s	Input	2		double	

Obr. 18 Premenné simulačného modelu klopného obvodu RS



Simulácia a vyhodnotenie modelu

Obr. 19 Priebeh simulácie modelu klopného obvodu RS

Na základe priebehu hodnotíme, že simulačný model funguje správne. Vrchný graf zobrazuje priebeh vstupu S, stredný reprezentuje vstup R a spodný stav klopného obvodu. Výsledný stav klopného obvodu sa mení presne tak ako je definované v matematickom opise klopného obvodu RS, takže simulačný model je navrhnutý správne.

#### Simulačný model semaforu

Ako vyzerá celkový simulačný model semaforu navrhnutý v Simulinku zobrazujeme na Obr. 20. Subsystém, ktorý zabezpečuje riadenie semaforu na základe tlačidla vytvorený využitím funkcií toolboxu Stateflow ilustruje Obr. 21.



Obr. 20 Simulačný model semaforu



Obr. 21 Subsystém simulačného modelu semaforu v toolboxe Stateflow

Jednotlivé stavy sú tzv. "subcharty". Teda obsahujú ďalšie stavy, ktorými semafor prechádza ak je v jednom z týchto dvoch hlavných stavov. Ich štruktúra v Stateflowe je na Obr. 22 a Obr. 23.



Obr. 22 Štruktúra "subchartu" zapnuty v simulačnom modeli semaforu



Obr. 23 Štruktúra "subchartu" vypnuty v simulačnom modeli semaforu

**Pomôcka pre tento príklad**: v toolboxe Stateflow vytvoríme "subchart" kliknutím pravým tlačidlom myši na funkčný blok zobrazujúci stav (Obr. 24).

Stateflow (chart) untitlec	Group & Subchart
t Simulation Analysis Coc () () () () () () () () () () () () () (	Format Decomposit Execution C Library Link
	Set Breakpoints
	Requirements Traceability  Coverage
Stav1	C/C++ Code > HDL Code >
entry: premenna = 1	PLC Code
<pre>w exit: premenna = 0;</pre>	Properties
	Help
▶	
fe	

Obr. 24 Vytvorenie "subchartu"



#### Simulácia a vyhodnotenie modelu

#### Obr. 25 Priebeh simulácie modelu semaforu

Vrchný graf zobrazuje stav vypínača. Zvyšné tri sú priebehy jednotlivých farieb svetelných signálov zobrazovaných semaforom v poradí – červená, oranžová, zelená. . Je zrejmé, že ak je vypínač vypnutý bliká oranžové svetlo. Ihneď po zapnutí vypínača začne semafor zobrazovať červené svetlo. V závislosti od času postupne prechádza všetkými stavmi, ktoré má podľa sformulovaného opisu nadobúdať ak je zapnutý. Ak vypneme vypínač znovu bliká iba oranžová. Simulácia nám potvrdila, že navrhnutý model semaforu vyhovuje matematickému opisu.