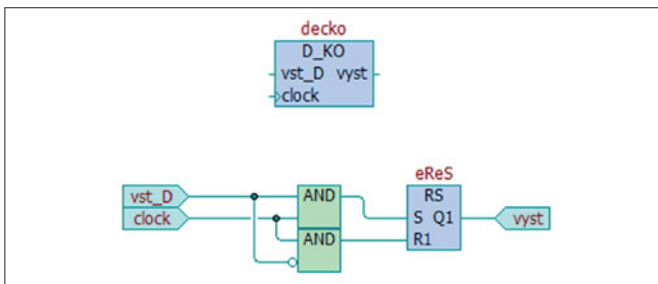


Inspiromat pro výuku a Tecomat: logika (nejenom) pro programátory – Díl třetí: sekvenční logické úlohy, intuitivní postup (třetí část)

Řešení třetí: s čítačem CTU

V dolní části obr. 13 je (opět v jazyce CFC) uvedeno řešení, které využívá opačný přístup. Vychází ze skutečnosti, že funkční blok CTU (*Counter Up*) realizuje kompletní binární dopředný čítač v délce 16 bitů (čítá modulo 2^{16}). Požadovaná funkce je řešena na jeho nejnižších čtyřech stupních. Pro úplnost uvedme význam jeho proměnných: CU – čítací vstup reaguje na náběžnou hranu, při které se zvýší stav čítače o jednu, R – resetovací vstup, jedničková hodnota nuluje obsah čítače, PV – hodnota předvolby, při jejím dosažení je nastaven výstup Q, CV – hodnota čítače. Zde je hodnota předvolby nastavena na 16. Při jejím dosažení nebo při stisku rudého tlačítka je čítač vynulován. Signálky jsou pak ovládány hodnotami dolní čtveřice bitů číselné hodnoty. Řešení by fungovalo rovnocenně i pro hodnoty předvolby 32, 64 nebo vyšší mocniny dvou až do 2^{15} .



Obr. 14. K příkladu 7: impulzní vzorkovací obvod (dynamický obvod D) – značka FB a jádro jeho deklarace (statický vzorkovací obvod se liší jen charakterem proměnné *clock* – reaguje na její jedničkovou hladinu)

Úloha 9: zpětný čítač z binárních stupňů
Obdobně jako v řešení 1 předchozího příkladu řešte zpětný čítač, který při stisku modrého tlačítka přejde z nulové hodnoty na hodnotu 1111 a tu pak postupně snižuje (vazby jsou vedeny od náběžných hran jednotlivých stupňů *s_modra*, *s_zelena*...).

Úloha 10: zpětný čítač s funkčním blokem CTD
Obdobně jako v řešení 2 řešte zpětný čítač s využitím funkčního bloku CTD.

Úloha 11: Desítkový čítač
Předchozí tři řešení (dopředný čítač) upravte tak, aby dopředný čítač pracoval jako desítkový – po dosažení hodnoty 1001 (9 desítkově) se dalším stiskem *s_modra* vynuloval a pak opět pokračoval v čítání.

Příklad 6: statický vzorkovací obvod (D-latch)

Jestliže je tisknuto tlačítko *tl_modre*, kopíruje signálka *s_ruda* stav tlačítka *tl_rude*, po uvolnění

tlačítka *tl_modre* si pamatuje svůj poslední stav. Při řešení s pevnou logikou realizuje tuto funkci klopný obvod typu statický D-latch (zádrž či vzorkovací obvod). Tato funkce se využívá v situacích, kdy je třeba řídit propustnost dat, je-li splněna podmínka přenosu. V okamžiku konce podmínky je zaznamenán (zmrazen) poslední stav sledovaných proměnných.

Řešení první: s podmíněným příkazem

```
IF tl_modre = 1 THEN s_ruda := tl_rude;
```

Řešení druhé: s logickým výrazem

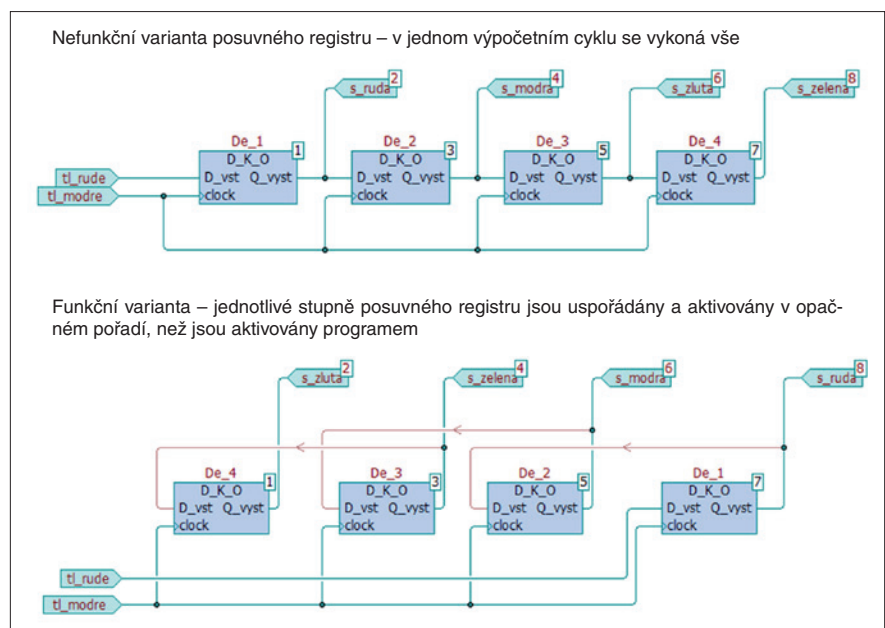
```
s_ruda := (tl_modre AND tl_rude) OR (NOT tl_modre AND s_ruda);
```

Řešení třetí: s pamětí RS

V horní části obr. 14 je uvedeno tělo uživatelského bloku D_KO v jazyce CFC.

Příklad 7: impulzní vzorkovací obvod

Signálka *s_ruda* kopíruje stav tlačítka *tl_rude* jen v okamžiku stisku tlačítka *tl_modre* (s jeho náběžnou hranou), jinak si pamatuje svůj poslední stav. Při řešení s pevnou logikou řeší tuto funkci klopný obvod typu impulzní D-latch (impulzní vzorkovací obvod). Tato funkce se využívá v situacích, kdy je zapotřebí zaznamenat stav proměnné na začátku významné události (např. při vyhodnocení poruchového stavu) nebo na začátku vzorkovacího intervalu.



Obr. 15. K příkladu 8: posuvný registr vpravo – sousledné (nefunkční) a protisměrné (funkční) řazení FB

Řešení

Lze použít kterékoliv řešení předchozího příkladu, pouze s tím rozdílem, že místo statické hodnoty proměnné *tl_modre* použijeme impuls vygenerovaný od náběžné hrany. Opět je účelné toto řešení použít pro deklaraci uživatelského funkčního bloku. Řešení program jádra obou funkčních bloků může být na pohled shodné (obr. 14), liší se pouze v deklaraci proměnné *clock*:

```
clock : BOOL; // statický obvod
clock : BOOL R_EDGE; // impulzní obvod
```

U vnitřního okraje značky dynamického FB je pak před jménem *clock* symbol špičky (obr. 14 nahoře), který vyjadřuje impulzní charakter vstupu, zatímco u statického FB je jen okraj značky.

Příklad 8: posuvný registr vpravo

Stiskem (s náběžnou hranou) tlačítka *tl_modre* uložte stav tlačítka *tl_rude*, rozsviňte signálku *s_ruda* (set) a současně přesuňte její dosavadní stav o pozici vpravo. Dalším stiskem *tl_modre* proveďte další posun vpravo – až k signálce *s_modrá*, jejíž stav se s dalším stiskem z registru vysune.

Řešení

Na obr. 15 jsou uvedeny dvě varianty řešení s použitím funkčního bloku *D_K_O* (dynamický obvod D). Varianta v horní části obrázku se jeví jako logická, funkční bloky jsou uspořádány v pořadí jejich očekávané aktivace. Překvapivě je ale nefunkční, protože během jednoho výpočetního cyklu se uskuteční všechny aktivace jednotlivých stupňů posuvného registru, takže výsledkem je prázdný posuvný registr. Funkční je program v dolní části obrázku, kde jsou jednotlivé stupně uspořádány v protisměrném pořadí oproti běhu programu. S podobným problémem se při programování sekvenčních úloh lze setkat častěji.

Poznámka

V pravém rohu značek funkčních bloků na obr. 15 (i jinde) jsou uvedeny čtvercové značky s vepsanými čísly. Generuje je editor CFC podle pořadí, v jakém jsou značky umístěny na ploše (zleva doprava a shora dolů). V tomto pořadí jsou pak funkční bloky programem aktivovány. Pořadí jejich aktivace lze změnit jen jiným rozmístěním na ploše.

Úloha 12

Vytvořte různé varianty posuvných registrů, např. tak, aby se stav posledního stupně vrátil na začátek do pozice *s_ruda*. Jinou alternativou může být požadavek, aby se stav signálky *s_ruda* postupně rozšiřoval na další pozice vpravo apod.

Příklad 9: zpožděný náběh, TON

Signálka *s_ruda* kopíruje stisk tlačítka *tl_rude* se zpožděním 2 s, při jeho uvolnění zhasne okamžitě. Krátkodobý stisk kratší než 2 s stav signálky neovlivní.

Řešení

Požadavek realizuje funkční blok časovače TON. Jeho použití v CFC a LD je uvedeno na obr. 16.

Tento typ zpoždění je účelný např. pro odložený start významné události, aby se zabránilo její nechtěné aktivaci

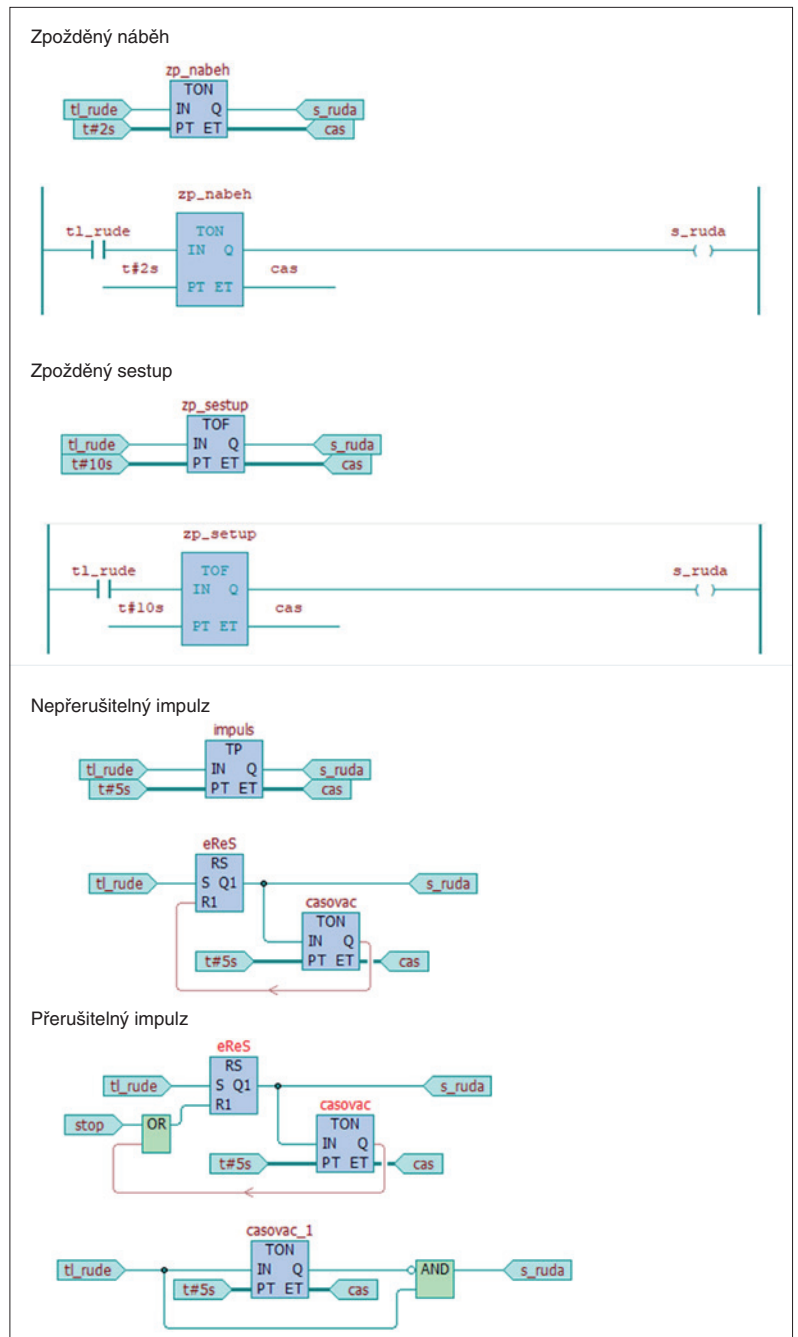
při krátkodobém nebo náhodném požadavku, např. při náhodném stisku tlačítka. V mechatronických systémech je žádoucí vyčkat po dosažení žádané polohy na ustálení mechanismu, dříve než je aktivován další pohyb. Funkčním blokem TON lze řešit i „časovou filtraci“, při které se potlačí krátkodobé impulzy, např. záškrtů kontaktů nebo rušení.

Příklad 9: zpožděný sestup, TOF

Signálka *s_ruda* kopíruje stisk tlačítka *tl_rude* současně se stiskem, při jeho uvolnění svítí dál a zhasne až se zpožděním 10 s. Krátkodobé uvolnění s opětovným stiskem (nulový impuls) kratší než 10 s neovlivní stav signálky.

Řešení

Požadavek realizuje funkční blok časovače TOF. Jeho použití v CFC a LD je uvedeno na obr. 16.



Obr. 16. K příkladu 9 až 11: použití časovačů

Popsaným způsobem lze řídit např. osvětlení, ventilátor na WC a další úlohy, kde je požadován časový přesah řízené aktivity po skončení požadavku na její aktivaci. Obdobně lze takto opět řešit časovou filtraci, při které se potlačí krátkodobé poklesy řídicí proměnné (nulové impulzy).

Poznámka 1

Pro upřesnění uvádíme význam vstupů a výstupů funkčních bloků časovačů: IN – řídicí logická proměnná časovače, PT – předvolený čas, Q – výstupní logická proměnná, ET – aktuální hodnota doby časování. Podrobný popis chování použitých funkcí a funkčních bloků je popsán v příručce Knihovny pro programování PLC Tecomat podle normy IEC 61131-3 nebo Esperanto programátorů PLC.

Poznámka 2

Pro funkční bloky TON a TOF platí obdoba De Morganových pravidel:

$$\text{TON}(x) = \text{NOT TOF}(\text{NOT } x),$$

$$\text{TOF}(x) = \text{NOT TON}(\text{NOT } x).$$

Příklad 10: nepřerušitelný impuls zadané délky

Po stisku tlačítka tl_rude se rozsvítí signálka s_ruda a bude svítit nezávisle na stavu tlačítka 5 s – nezhasne po uvolnění tlačítka ani po jeho opakovaném stisku, ale až po uplynutí nastavené doby 5 s, po které lze signálku novým stiskem opět rozsvítit na dalších 5 s.

Řešení první

Požadavek realizuje funkční blok časovače TP. Jeho použití je formálně obdobné jako použití TON a TOF na obr. 16. Popsaným způsobem lze řídit zejména procesy, které mají trvat stálou dobu zadané délky, např. osvětlení, ohřev, ventilace, průtok média apod. Výhodně lze tento postup použít u dávkových procesů, kde dávka energie nebo objem materiálu jsou úměrné době aktivace procesu, např. při řízení elektrického ohřevu, vytápění přímotopnými topidly, dávkování tekutin (dávkovací čerpadla), dávkování sypkých materiálů apod. Začátek impulsu generovaného časovačem TP je odvozen od náběžné hrany řídicí proměnné

Seriál krátkých filmů o programovacím prostředí Mosaic na YouTube

V kanálu YouTube stránek www.academy.cz je uvedena kolekce krátkých instruktážních videosouborů, které postupně a názornou formou popisují práci s vývojovým systémem Mosaic, jenž je určen k programování systému Tecomat. Každý videosoubor je zaměřen na určité téma, a uživatel si tak může vybrat to své. Seriál je postupně rozšiřován o nové soubory. Stačí se přihlásit k odběru kanálu a zájemcům neunikne žádná část.

ne a lze jej považovat za zobecnění impulsu od náběžné hrany, který generuje funkční blok R_TRIG , jenž ale trvá nastavenou dobu, nikoliv jen jeden výpočetní cyklus. Po tuto dobu lze řešit např. inicializaci zahajovaného procesu, signalizovat změnu stavu apod.

Řešení druhé

Jiná možnost řešení je uvedena na obr. 16 (opět v CFC). Řešení je sice komplikovanější, ale připouští možnost úpravy, která by umožňovala zkrácení generovaného impulsu.

Příklad 11: přerušitelný impuls zadané délky

Po stisku tlačítka tl_rude se rozsvítí signálka s_ruda a bude svítit 5 s nezávisle na stavu tlačítka – nezhasne po uvolnění tlačítka ani po jeho opakovaném stisku, ale až po uplynutí nastavené doby 5 s, po které lze signálku novým stiskem opět rozsvítit na dalších 5 s. Signálku je ale možné zhasnout i stiskem zvoleného tlačítka ve významu stop (např. tl_modre).

Řešení

Zadání vyhovuje řešení z obr. 16 (opět v CFC). Je zde uvedena ještě jiná alternativa, používající časovač TON.

Ing. Ladislav Šmejkal, CSc., *Teco a. s.*,
Ing. Josef Kovář a Ing. Zuzana Prokopová,
učitelé automatizace na SPŠ Zlín



Tecomat Foxtrot

Platforma pro automatizaci a komunikaci strojů, procesů, budov a dopravy



www.tecomat.cz

IEC-61131 | IoT

| Smart House

| Smart City

| Industry 4.0

www.tecoacademy.cz

► Pokročilé technologie v řízení a rozvoji českých měst a obcí

Seminář *Pokročilé technologie v řízení a rozvoji českých měst a obcí* se uskuteční ve čtvrtek 5. prosince 2019 (8:45 až 16:00) v prostorách budovy G Technické univerzity

v Liberci (Studentská 2). Vstup je zdarma. Navazuje na seminář *Chytré technologie*, který se letos v lednu rovněž konal na půdě Technické univerzity v Liberci (TUL). Organizuje jej Národní klastrová asociace ve spolupráci s Fakultou mechatroniky, informatiky a mezioborových studií, Fakultou strojní a Fakultou umění a architektury TUL. Seminář je určen především pro zastupitele a vedoucí představitele

měst a obcí, odborů strategického rozvoje měst a firem, výzkumných institucí, škol a pracovníků IT. Podrobný program semináře a možnost přihlásit se je na adrese: <http://nca.cz/cs/pokrocile-technologie-v-rizeni-ceskym-mesta-obci>. Účastníci semináře budou seznámeni s nově připravovanou akreditací studijní specializace Chytré technologie, na které se podílejí všechny tři uvedené fakulty TUL. (šm)

krátké zprávy



Předplatné časopisu AUTOMA lze pohodlně sjednat na stránkách www.automa.cz