

| | | |
|-------------------|---|--|
| Předmět | MLOS - LOS - Logické systémy | |
| Ústav | ÚAMT | |
| Úloha č. 6 | Sekvenční logika – návrh asynchronních a synchronních binárních čítačů, výhody a nevýhody, využití | |
| Student | | |

Cíle

- Funkce čítačů a použití v digitálních systémech.
- Rozdíly mezi asynchronními a synchronními čítači.
- Metodika návrhu binárních čítačů.
- Popis čítače v jazyce VHDL a jeho implementace do FPGA.

Teoretický úvod

Čítač je podle obecné definice zařízení, které slouží ke stanovení absolutní četnosti výskytu určitého jevu, nebo události. Jedná se o *počítadlo* - způsob realizace, funkční principy, nebo metoda detekce sledovaného jevu přitom nejsou podstatné. Čítače mají velké uplatnění v digitální technice např. čítání požadované sekvence, stavové automaty, děličky frekvence apod.

Z hlediska digitální techniky je čítač obvykle chápán jako sekvenční logický obvod, který zaznamenává počet změn vstupního hodinového signálu (nástupná, nebo sestupná hrana). S každou změnou hodinového signálu dochází ke změně hodnoty výstupu čítače. Výstup nabývá hodnoty v definovaném intervalu. V případě binárního čítače je interval definován šířkou výstupního slova. Hodnoty jsou reprezentovány použitým kódováním.

V technické praxi jsou často využívány binární čítače. Jejich výstupní hodnota je vyjádřena binárním kódem. Interval výstupních hodnot je $\langle 0 ; (2^n - 1) \rangle$, kde n je šířka výstupního slova čítače. Pravdivostní tabulka 3-bitového binárního čítače vpřed je v tabulce 1.

Tab. 1: Pravdivostní tabulka 3-bitového binárního čítače

| stav | současný stav | | | následující stav (buzení) | | |
|------|---------------|-------|-------|---------------------------|-------|-------|
| | Q_2 | Q_1 | Q_0 | D_2 | D_1 | D_0 |
| s0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s1 | 0 | 0 | 1 | 0 | 1 | 0 |
| s2 | 0 | 1 | 0 | 0 | 1 | 1 |
| s3 | 0 | 1 | 1 | 1 | 0 | 0 |
| s4 | 1 | 0 | 0 | 1 | 0 | 1 |
| s5 | 1 | 0 | 1 | 1 | 1 | 0 |
| s6 | 1 | 1 | 0 | 1 | 1 | 1 |
| s7 | 1 | 1 | 1 | 0 | 0 | 0 |

Stejný interval výstupních hodnot, ovšem odlišné kódování využívá Grayův čítač. Realizace Grayova čítače je náročnější než v realizace binárního čítače. Výhodou čítače v Grayově kódu je, že pro dvě následující hodnoty se výstupní slovo vždy mění pouze v jenom bitu. U binárního kódu dochází u dvou následujících hodnot ke změnám na jednom i více bitech. Proto u Grayova čítače nemůže nikdy nastat stav, že by na výstupu kvůli rozdílným zpožděním jednotlivých výstupů z KO D byla jiná hodnota než požadována. Tento typ čítače se často využívá např. při realizaci vyrovnávacích FIFO (First In, First Out) pamětí, které přenášejí data mezi různými hodinovými doménami. Pravdivostní tabulka 3-bitového čítače v Grayově kódu je v tabulce 2.

Tab. 2: Pravdivostní tabulka 3-bitového binárního čítače v grayově kódu

| stav | současný stav | | | následující stav (buzení) | | |
|------|---------------|-------|-------|---------------------------|-------|-------|
| | Q_2 | Q_1 | Q_0 | D_2 | D_1 | D_0 |
| s0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s1 | 0 | 0 | 1 | 0 | 1 | 1 |
| s2 | 0 | 1 | 1 | 0 | 1 | 0 |
| s3 | 0 | 1 | 0 | 1 | 1 | 0 |
| s4 | 1 | 1 | 0 | 1 | 1 | 1 |
| s5 | 1 | 1 | 1 | 1 | 0 | 1 |
| s6 | 1 | 0 | 1 | 1 | 0 | 0 |
| s7 | 1 | 0 | 0 | 0 | 0 | 0 |

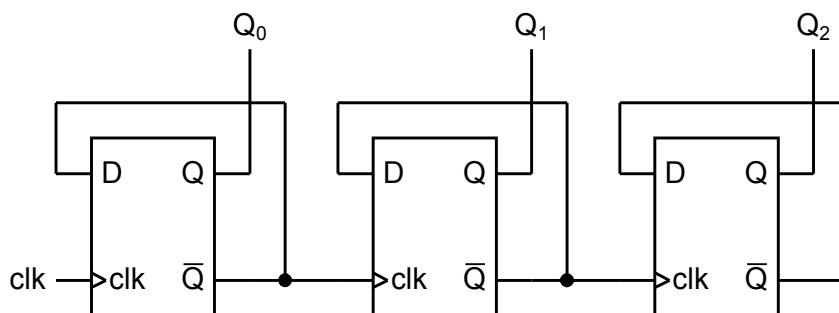
Pro implementaci čítání v desítkové soustavě je možné využít BCD čítače. Každá číslice desítkové soustavy je vyjádřena BCD kódem pomocí 4-bitového binárního čísla. Interval hodnot je omezen na $<0 \div 9>$.

Pro zkrácení intervalu čítání (oproti binárnímu čítači) se využívají čítače modulo M, které čítají v intervalu $<0 \div (M-1)>$. Typicky se tyto čítače používají pro dělení frekvence, kde M odpovídá dělicímu poměru.

Poslední skupinu čítačů tvoří speciální čítače, pro které je charakteristická absence určitých hodnot v čítaném intervalu. Takový čítač čítá posloupnost například 0-2-5-7.

Při volbě vhodného čítače hraje velmi významnou roli také otázka synchronnosti vůči řídicímu hodinovému signálu. Asynchronní struktury jsou zpravidla jednodušší, zároveň jsou však náchylnější ke vzniku hazardních stavů. Především u systémů pracujících s vysokou frekvencí hodinového signálu je vhodné využívat synchronní obvodové struktury.

Na obrázku 1 je příklad implementace 3-bitového asynchronního binárního čítače realizovaného pomocí klopných obvodů typu D. Čítač je tvořen kaskádním spojením klopných obvodů, které jsou zapojeny jako děličky frekvence dvěma.



Obr. 1: Tříbitový asynchronní čítač s klopnými obvody D

Na obrázku 2 jsou časové průběhy hodinového vstupu a výstupů 3-bitového asynchronního čítače z obrázku 1. Průběhy výstupů Q_0 , Q_1 a Q_2 mají různé zpoždění vůči hodinovému signálu clk. Toto zpoždění je způsobeno kaskádním zapojením KO D, protože změna na vstupním hodinovém signálu se postupně přenáší jednotlivými stupni čítače. Toto zpoždění roste s počtem použitých klopných obvodů.

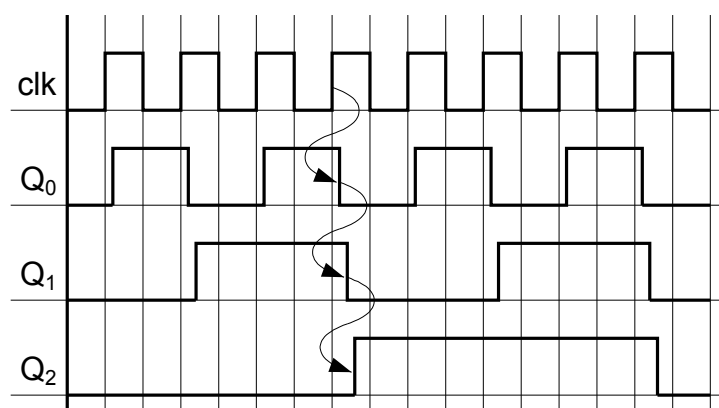
Změna hodnoty více než jednoho bitu binárního čítače se projeví postupně. Výstupní hodnota po dobu zpoždění neodpovídá skutečné hodnotě čítače. To se nejvíc projeví při přetečení čítače, kdy čítač přechází mezi stavy:

$$111 \rightarrow 000$$

Výstupní stav se ale změní postupně tři krát:

$$111 \rightarrow 110 \rightarrow 100 \rightarrow 000$$

Pokud je výstup čítače připojen ke vstupu kombinační logiky, jeho hodnota bude zpracována chybně.



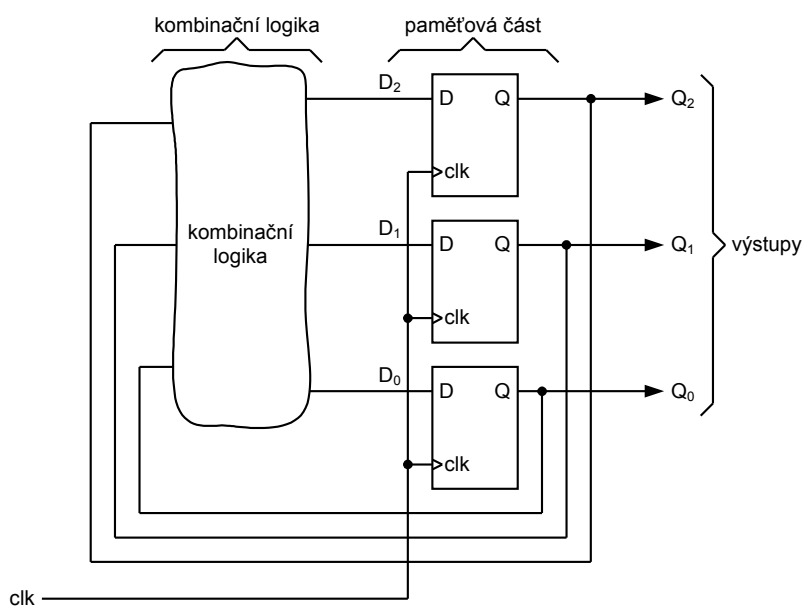
Obr. 2: Průběh výstupního signálu asynchronního čítače

Na obrázku 3 je koncepční schéma 3-bitového binárního synchronního čítače. Čítač je rozdělen na paměťovou část tvořenou KO D a kombinační logiku. Synchronní binární čítač obsahuje oproti asynchronnímu navíc kombinační logickou funkci, která dekóduje aktuální stav na stav následující.

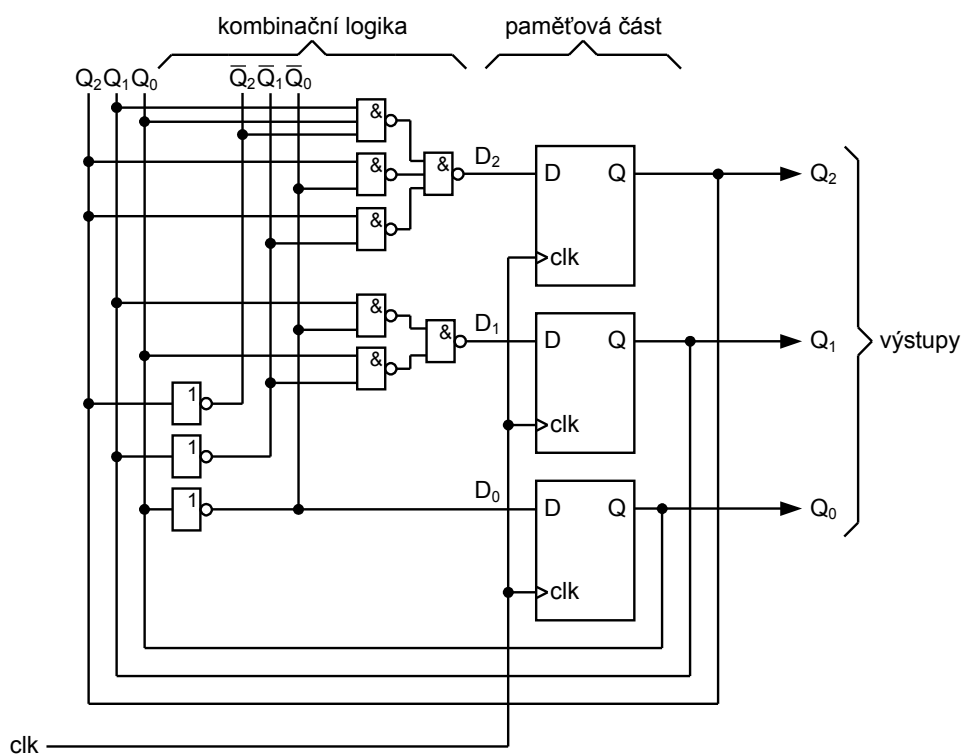
Hodnota následujícího stavu je připojena ke vstupům všech KO D před příchodem hrany hodinového signálu. Díky tomu je zabezpečeno, že hodnoty všech bitů jsou přeneseny ze vstupů na výstupy KO D synchronně s hranou hodinového signálu a se stejným spožděním.

Struktura synchronních čítačů je obvykle složitější než struktura asynchronních čítačů a jejich implementace je náročnější na plochu. Synchronní čítače jsou při návrhu digitálních obvodů využívány častěji, než asynchronní.

Na obrázku 4 je obvodové schéma 3-bitového synchronního binárního čítače navrženého dle koncepčního schématu z obrázku 3, vynačena je kombinační a paměťová část.



Obr. 3: Konceptní schéma 3-bitového binárního synchronního čítače



Obr. 4: Obvodové schéma 3-bitového binárního synchronního čítače

Na následující stránce je ukázka postupu návrhu synchronního 3-bitového binárního čítače. Tento návrh počítá se strukturou ze schématu z obrázku 3. Paměťová část je realizována pomocí KO D. Kombinační funkce je odvozena z pravdivostní tabulky.

Pravdivostní tabulka 3-bitového binárního čítače bude mít 2^n řádků, kde n je počet bitů čítače. Z pravdivostní tabulky jsou odvozeny funkce popisující výstupy D kombinační sítě. Vstupem kombinační sítě jsou výstupy Q čítače, jsou uvedeny na levé straně pravdivostní tabulky. Výstupy kombinační logiky D, které jsou zároveň vstupy paměťové části, jsou na pravé straně pravdivostní tabulky. Z pravdivostní tabulky jsou následně odvozeny funkce popisující kombinační logiku.

| Aktuální stav | | | Následující stav | | |
|----------------|----------------|----------------|------------------|----------------|----------------|
| Q ₂ | Q ₁ | Q ₀ | D ₂ | D ₁ | D ₀ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Kombinační funkce pro vstup D₀ je tvořena pouze invertorem:
 $D_0 = \overline{Q_0}$

Kombinační funkce pro vstup D₁ je tvořena hradlem XOR:
 $D_1 = Q_1 \text{ XOR } Q_0$

Kombinační funkce pro vstup D₂ je odvozena z Karnaughovy mapy:

| Q ₂ | Q ₁ Q ₀ | | | |
|----------------|-------------------------------|----|----|----|
| | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

$$D_2 = (Q_2 * \overline{Q_1}) + (Q_2 * \overline{Q_0}) + (Q_0 * Q_1 * \overline{Q_2})$$

Odvozené funkce popisující kombinační logiku 3-bitového mohou být použity například při popisu čítače v jazyce VHDL. V případě širších čítačů je odvození funkcí kombinační logiky náročné a je vhodnější v jazyce VHDL využít behaviorální popis.

Ukázka metodicky korektního behaviorálního VHDL popisu 3-bitového synchronního čítače, využívající oddělení kombinační části od sekvenční:

```
entity counter is
    Port ( clk : in  STD_LOGIC;
          Q   : out  STD_LOGIC_VECTOR (2 downto 0));
end counter;
architecture Behavioral of counter is
    signal d_int,q_reg : STD_LOGIC_VECTOR (2 downto 0);
begin
    -- sekvecni (pametova) cast
    process (clk) begin
        if (clk'event and clk='1') then
            q_reg <= d_int;
        end if;
    end process;

    -- kombinacni cast
    process (q_reg)
    begin
        d_int <= q_reg + 1;
    end process;
    Q <= q_reg;
end Behavioral;
```

Výše popsané čítače obsahují pouze jeden vstup, na základě kterého je inkrementována hodnota n-bitového výstupu. Takový čítač zaznamenává počet změn na vstupním řídicím signálu.

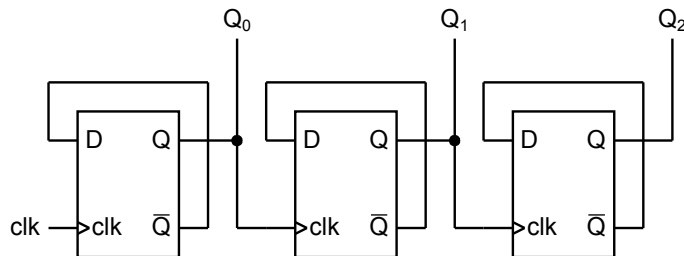
V případech, kdy není dostačující jednoduchý inkrementující čítač je možné jeho funkce rozšířit. Typickým případem rozšíření funkce čítače je možnost změny směru čítání. To je možné realizovat dvěma způsoby. V prvním případě čítač obsahuje dva vstupy, kdy jeden vyvolává inkrementaci a druhý dekrementaci. Další způsob využívá vstup, který nastavuje, zda čítač inkrementuje nebo dekrementuje a hodinový vstup.

Další typické rozšíření funkce čítačů je možnost nulování nebo přednastavení počáteční hodnoty čítače. To může být implementováno jako synchronní nebo asynchronní. V případě synchronní implementace dojde k provedení s hranou hodinového signálu a v případě asynchronní implmentace s úrovní příslušného vstupu.

Vypracování laboratorní úlohy

Úkol č. 1 (0,6 bodu)

Na obrázku je 3-bitový čítač realizovaný pomocí KO D. Doplňte pravdivostní tabulku a vyznačte typ čítače.



| CLK | Q ₂ | Q ₁ | Q ₀ |
|-----|----------------|----------------|----------------|
| 0→1 | 0 | 0 | 0 |
| 0→1 | | | |
| 0→1 | | | |
| 0→1 | | | |
| 0→1 | | | |
| 0→1 | | | |
| 0→1 | | | |
| 0→1 | | | |

- a) Synchronní čítač vpřed
- b) Synchronní čítač vzad
- c) Asynchronní čítač vpřed
- d) Asynchronní čítač vzad

BCD čítač je N-bitový synchronní binární čítač modulo M.

Doplňte N a M:

N =

M =

Úkol č. 2 (0,6 bodu)

Zjistěte výstupní frekvenci oscilátoru osazeného na vývojové desce Spartan-6. Navrhněte děličku frekvence tohoto hodinového signálu tak, aby výstupní signál měl periodu větší než 1 sekunda a menší než 2 sekundy. Uveďte pouze výpočet (schéma zapojení není nutné). Kolik bitů bude nutno použít v návrhu ?

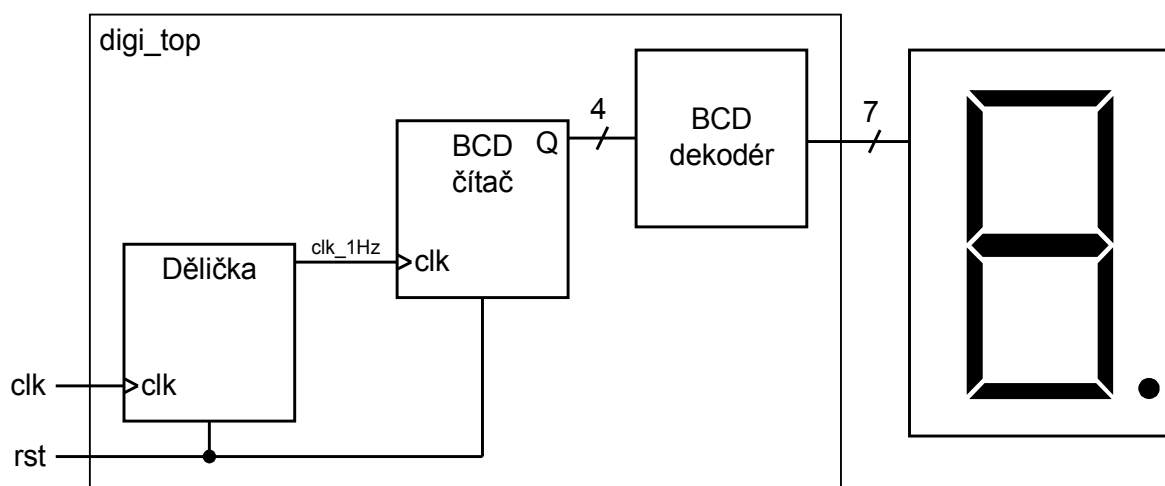
Úkol č. 3 (0,6 bodu)

Ve vývojovém prostředí Xilinx ISE založte nový projekt DEL_1HZ. Navrhněte VHDL popis **děličky frekvence (čítač modulo M)** tak, aby výsledná frekvence byla $\text{clk_1Hz} = 1\text{Hz}$. Základním hodinovým signálem clk bude oscilátor umístěný na vývojové desce. Výsledek zobrazte na jedné z LED diod. Výsledný návrh imlementujte a funkci otestujte na vývojové desce.

Úkol č. 4 (2x0,6 bodu = 1,2 bodu)

Ve vývojovém prostředí Xilinx ISE založte nový projekt BCD_C. Navrhněte BCD čítač čítající s frekvencí 1 Hz, kterého výstup se bude zobrazovat na 7-segmentovém displeji. Jako děličku frekvence (1 Hz) použijte blok z předcházejícího úkolu č. 3. **Systém navrhněte pomocí strukturálního popisu.**

Blokové zapojení celého systému by mělo vypadat následovně:



Výsledný návrh implementujte a funkci otestujte na vývojové desce.

Bonusový úkol č. 5 (0,6 bodu)

Upravte předcházející úkol tak, aby byl celý návrh plně synchronní a pracoval pouze se základní frekvencí 100 MHz, tj. všechny klopné obvody byly připojeny na stejný hodinový signál. Upravte děličku frekvence tak, aby generovala pouze povolovací impuls (enable) s frekvencí 1 Hz (po dobu jednoho taktu hlavních hodin). BCD čítač upravte tak, aby zápis do registru (v sekvenční části) probíhal pouze, pokud je povolovací (enable) signál aktivní.