

SKLF	BICT				Datum měření:							2021		Příjmení a jméno:
					Den (vyznačte X):				Po	Út	St	Čt	Pá	
Hodina (vyznačte X):	7	8	9	10	11	12	13	14	15	16	17	18	19	

SYNTÉZA KOMBINAČNÍCH LOGICKÝCH FUNKCÍ POMOCÍ KOMBINAČNÍCH OBVODŮ

1 ZADÁNÍ

- Navrhněte pravdivostní tabulku pro dvoubitovou sčítačku s výstupním bitem pro přenos do vyššího řádu. Pomocí Karnaughovy mapy minimalizujte výsledné kombinačně logické funkce a zapojte je pomocí obvodů NAND.
- Pomocí dvojnásobného čtyřbitového multiplexoru realizujte funkci nižšího bitu sčítačky (y_0). Proveďte eliminaci pro bity b_0, b_1, a_0 . Zapojení navrhněte tak, aby multiplexor pracoval jako osmibitový.
- Ověřte správnost obou realizací srovnáním s integrovanou sčítačkou 74283 pomocí simulace v programu MicroCAP.

2 TEORETICKÉ POZNATKY

Realizací kombinační logické funkce rozumíme sestavení schématu zapojení pomocí příslušných obvodů, které ze vstupních proměnných vytvoří výstupní proměnnou s požadovanou logickou funkcí. V praxi se pro realizaci kombinačně logické funkce často používá pouze jeden integrovaný obvod. Buď se vyrábí sériově a najdeme ho v katalogu, nebo se sériově nevyrábí a k realizaci funkce využijeme paměti (PROM) nebo obvody PLD. Přesto se v základu vždy vychází ze zápisu logické funkce pomocí tvaru SOP (Sum Of Products) nebo POS (Product Of Sums).

Nejčastější způsoby realizace kombinačně logických funkcí v číslicové technice jsou:

- pomocí integrovaných obvodů typu NAND, INVERT, NOR
- pomocí multiplexorů a demultiplexorů
- pomocí speciálních obvodů – převodníky kódu, násobičky, sčítačky
- pomocí pamětí PROM nebo programovatelných obvodů typu PLD

V této úloze využijeme k realizaci obvodů především obvodů typu NAND a multiplexorů. Výhodou zapojení pomocí obvodů typu NAND je, že pokud nevyužijeme všechny členy NAND v jednom pouzdře při zapojení první logické funkce, můžeme tyto obvody využít i při zapojení následující logické funkce.

Realizace pomocí obvodů typu NAND

Pro realizaci obvodů pomocí této metody vycházíme z Karnaughovy mapy, do které zapíšeme hodnoty výstupní funkce. Poté tuto funkci zapíšeme v minimalizovaném tvaru (SOP). Na tuto funkci aplikujeme de Morganovo pravidlo, pomocí kterého převedeme tvar s logickými součiny na tvar s logickými součiny:

$$y = \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{c} \Rightarrow \text{de Morganovo pravidlo} \Rightarrow \overline{\overline{\bar{b} \cdot c} \cdot \overline{a \cdot \bar{b} \cdot \bar{c}}}$$

Výslednou logickou funkci můžeme po této úpravě zapojit jen pomocí obvodů typu NAND.

Realizace pomocí multiplexoru

U realizace obvodů pomocí tohoto způsobu je postup poněkud komplikovanější a jeho podrobný popis není předmětem této úlohy. Postup je uveden v přednáškách. U realizace pomocí této metody se vychází ze základu funkce multiplexoru. Každý multiplexor obsahuje adresové, datové vstupy a vstup S – select, který povoluje, popřípadě zakazuje přepis hodnoty na datovém vstupu na výstup. Tohoto se využívá při skládání multiplexorů do větších funkčních celků. Počet datových vstupů závisí na počtu adresových vstupů: počet datových vstupů = 2^N , kde N je počet adresových vstupů. Adresové vstupy adresují příslušný datový vstup, který má být převeden na výstup. Tohoto se právě využívá u realizace kombinační logické funkce, kdy na datové vstupy připojíme v závislosti na příslušné funkci buď +5V nebo GND. Tímto zajistíme, že při příslušných vstupních proměnných, které jsou přivedeny na adresové vstupy, dostaneme na výstupu požadovanou logickou hodnotu.

Mohlo by se na první pohled zdát, že u multiplexoru s N adresovými vstupy můžeme vytvořit obvod pouze pro N vstupních proměnných, ale v praxi se využívá zapojení, ve kterých realizujeme pomocí multiplexoru s N adresovými vstupy logickou funkci, která obsahuje $N+1$ vstupních proměnných. Postup návrhu takovýmto způsobem realizujeme následovně.

V prvním kroku výraz pro výslednou funkci doplníme tak, aby každý sčítanec obsahoval všechny eliminované proměnné. To znamená, že každý **součinový člen vynásobíme závorkou, která obsahuje přímou i negovanou hodnotu chybějící eliminované proměnné**. Ve druhém kroku závorky roznásobíme a vytkneme společné eliminované funkce. Ve třetím kroku všechny funkce doplníme do tabulky, jejíž vstupní hodnoty budou eliminované funkce a výstupní hodnoty budou určeny zbytkovou funkcí.

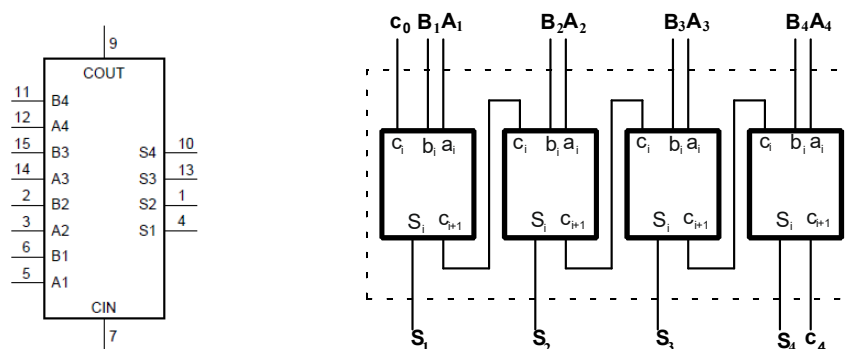
Integrovaná sčítačka 74283

Obvod 74283 je úplná čtyřbitová sčítačka. Tento obvod je vybaven vstupy A1 až A4, B1 až B4 a C0 (resp. CIN - Carry in). Výstupy jsou označeny S1 až S4 a C4 (COUT - Carry out).

Vstupy A1 až A4 a B1 až B4 slouží pro připojení dvou sčítaných čísel. Vstup C0 je určen pro přenos z nižšího řádu. Tento vstup slouží pro vzájemné propojení několika obvodů 74283 za účelem sčítání delších čísel. Podobný účel má výstup C4, který je přenosem do vyššího řádu. Výstupy S1 až S4 slouží pro zobrazení součtu. Chování obvodu lze popsat rovnicí:

$$(A_4, A_3, A_2, A_1) + (B_4, B_3, B_2, B_1) + C_0 = (C_4, S_4, S_3, S_2, S_1)$$

Z výše uvedené rovnice plyne, že výstup C4 lze chápat jako pátý bit výstupu. Pokud je výsledek součtu nižší nebo roven 1111 (15_{10}) je $C_4 = 0$, v opačném případě je C_4 nastaven a výstupy S1 až S4 obsahují spodní 4 bity výsledku.



Obr. 1. Značka obvodu 74283 a princip jeho činnosti

Vnitřní strukturu sčítačky si můžeme představit jako zřetězení čtyř jednobitových sčítaček. Každá sčítačka sečte jeden bit sčítanců a aplikuje přenos z nižšího řádu. Případný přenos pošle dál do vyššího bitu. Skutečná vnitřní struktura je však odlišná. Z důvodu rychlosti výpočtu se výstupní bity realizují přímo kombinační logikou.

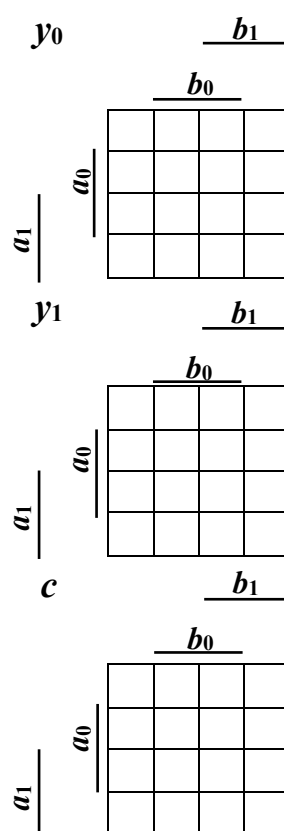
SKLF	BICT				Datum měření:							2021		Příjmení a jméno:
					Den (vyznačte X):				Po	Út	St	Čt	Pá	
Hodina (vyznačte X):	7	8	9	10	11	12	13	14	15	16	17	18	19	

Teoretická část (vyplnit ručně)

- A) Navrhnete pravdivostní tabulku pro dvoubitovou sčítačku s výstupním bitem pro přenos do vyššího řádu. Pomocí Karnaughovy mapy minimalizujte výsledné kombinačně logické funkce (SOP) a upravte je do struktury NAND-NAND pro zapojení z obvodů NAND, případně invertorů. Nakreslete schéma zapojení pro logické funkce y_0 , y_1 a přenosový bit c_1 do vyššího řádu (carry bit).

Tabulka 1

s	a_1	a_0	b_1	b_0	c_1	y_1	y_0
0	0	0	0	0			
1	0	0	0	1			
2	0	0	1	0			
3	0	0	1	1			
4	0	1	0	0			
5	0	1	0	1			
6	0	1	1	0			
7	0	1	1	1			
8	1	0	0	0			
9	1	0	0	1			
10	1	0	1	0			
11	1	0	1	1			
12	1	1	0	0			
13	1	1	0	1			
14	1	1	1	0			
15	1	1	1	1			

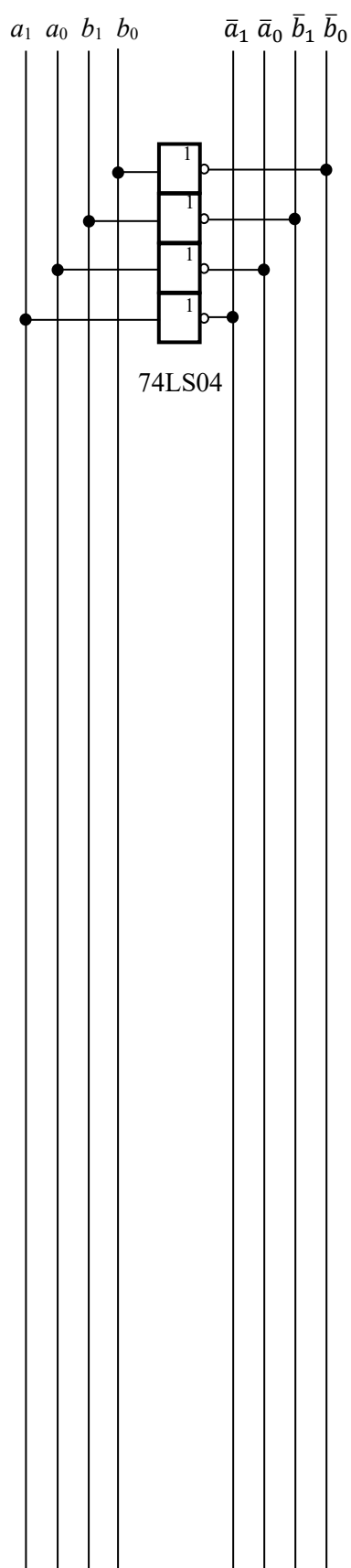


$y_0 =$

$y_1 =$

$c =$

Zapojení strukturou NAND-NAND (s využitím obvodů NAND, případně invertorů):

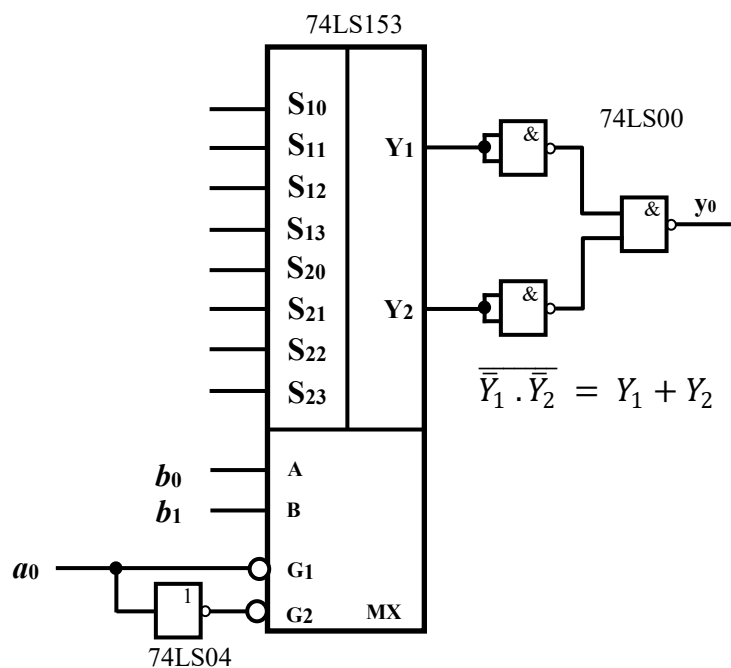


B) Pomocí dvojného čtyřbitového multiplexoru 74LS153 realizujte funkci nižšího bitu sčítačky y_0 . Proveďte eliminaci pro bity b_0, b_1, a_0 . Zapojení navrhnete tak, aby multiplexor pracoval jako osmibitový.
Nápověda: Vstup G_1 (resp. G_2) aktivuje čtyřbitový vstup S_1 (resp. S_2).

$y_0 =$

Tabulka 2

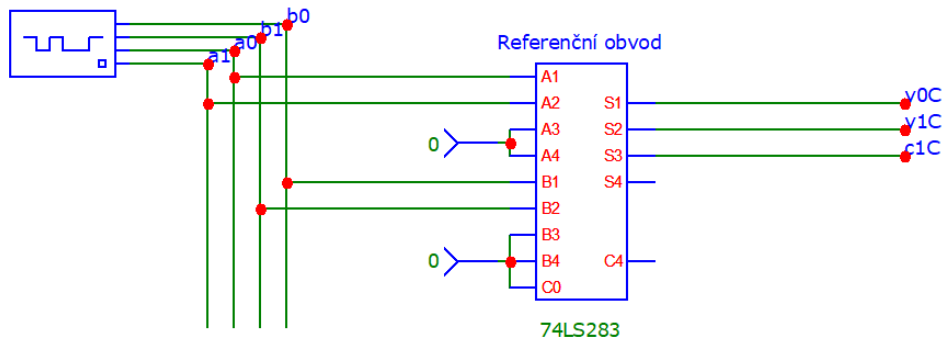
	Tabulka zbytkových funkcí			
s	a_0	b_1	b_0	S_{xx}
0	\bar{a}_0	\bar{b}_1	\bar{b}_0	$S_{10} =$
1	\bar{a}_0	\bar{b}_1	b_0	$S_{11} =$
2	\bar{a}_0	b_1	\bar{b}_0	$S_{12} =$
3	\bar{a}_0	b_1	b_0	$S_{13} =$
4	a_0	\bar{b}_1	\bar{b}_0	$S_{20} =$
5	a_0	\bar{b}_1	b_0	$S_{21} =$
6	a_0	b_1	\bar{b}_0	$S_{22} =$
7	a_0	b_1	b_0	$S_{23} =$



SKLF	BICT				Datum měření:				2021				Příjmení a jméno:
					Den (vyznačte X):				Po	Út	St	Čt	Pá
Hodina (vyznačte X):	7	8	9	10	11	12	13	14	15	16	17	18	19

Zadání simulace

C) Ověření funkce navržených obvodů bude provedeno v programu MicroCAP podobným způsobem jako u úlohy BCD-7, ke které je k dispozici videonávod. Všechny logické obvody použijte z řady LS.

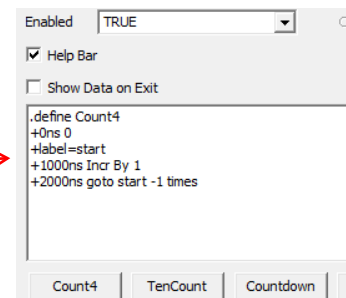


Obr. 2. Základ zapojení pro ověření syntézy sčítačky.

Vstupními signály do sčítaček jsou dvoubitová slova (a_1, a_0) a (b_1, b_0), viz tabulka 1 v teoretické části A). Jedná se tedy o celkem 4 bity, které vytvoří 16 kombinací značených stavovým indexem 0 až 15 (sloupec s v tabulce 1).

Pro generování těchto kombinací je možné využít čtyřbitový generátor Stim4. Po vložení do schématu je nutné generátor „naprogramovat“. Do textového pole v nastavení parametrů vložte následující příkazy:

```
.define Count4
+0ns 0
+label=start
+1000ns Incr By 1
+2000ns goto start -1 times
```



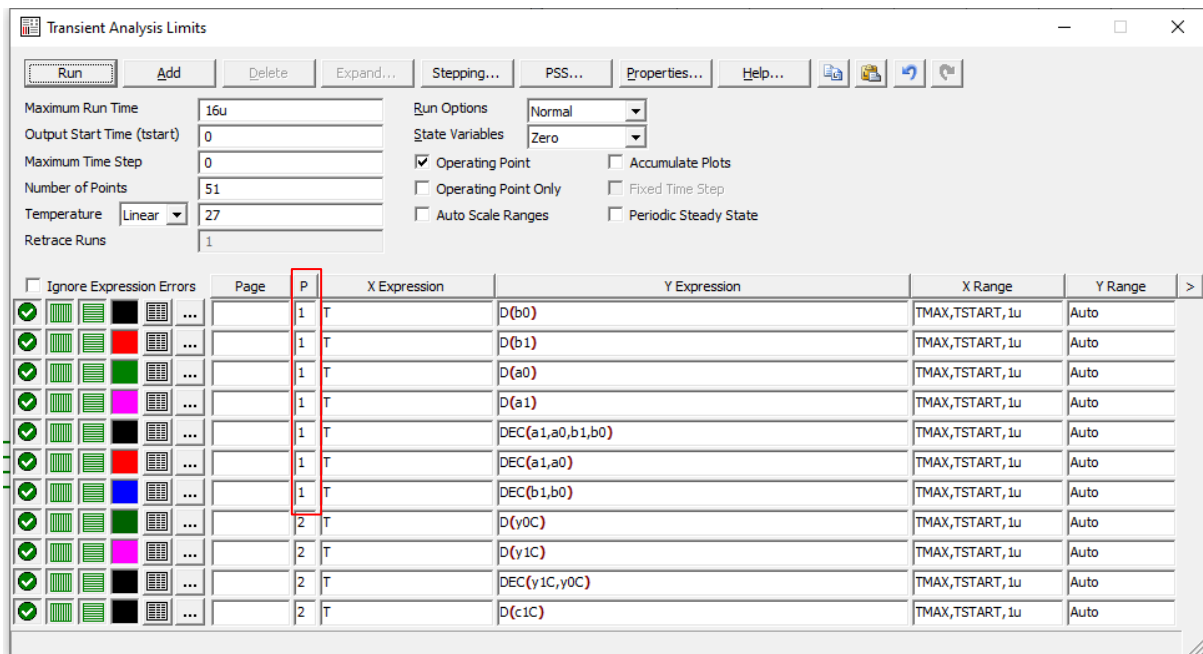
Generátor každou mikrosekundu inkrementuje výstupní kombinaci. Všechny 16 stavů tedy proběhne v intervalu 16 μ s.

Podle obr. 2 zapojte generátor a obvod 74LS283, jehož výstupy budou sloužit jako reference pro ověření správnosti syntézy podle bodů z teoretické části A) a B). Vstupní signály pojmenujte ve shodě s tabulkou 1 jako a_1, a_0, b_1, b_0 . Je nutné dodržet pořadí podle obr. 2 vzhledem ke značce generátoru, aby vstupní kombinace odpovídaly tabulce. Na značce je čtverečkem označen nejvyšší bit.

Obvod 74LS283 je logická čtyřbitová sčítačka se vstupem přenosu z nižšího řádu (C0). Pokud nastavíme na nulu vstupy A3, A4, B3, B4 a C0, tak se obvod bude chovat jako dvoubitová sčítačka podle tabulky 1. Pro výstupy platí

$y_0 = S1$
 $y_1 = S2$
 $c_1 = S3$ (přenos do vyššího řádu)

Při označování jmen uzlů (vodičů) použijte y0C, y1C a c1C pro odlišení výstupů pro bod C) zadání. Do stejného schématu doplňte realizaci obvodů podle zadání A) a B), jejichž výstupy označíte jako y0A, y1A a c1A, resp. y0B.



Obr. 3. Nastavení časové analýzy.

Analýza bude provedena v časové oblasti (Analysis/Transient). Délku simulace nastavte na 16μs. Pro zobrazení průběhů v grafu je možné s výhodou využít funkce

$$\text{DEC}(b_{\text{MSB}}, \dots, b_{\text{LSB}}),$$

kteřá zobrazí skupinu bitů jako číslo v desítkové soustavě (případně HEX a BIN v šestnáctkové nebo binární soustavě).

Vstupní signály pro testování sčítačky jsou generovány čtyřbitovým generátorem (obr. 2). V tabulce 1 je vidět, že (b_1, b_0) reprezentují nižší dva bity a (a_1, a_0) reprezentují vyšší dva bity vstupní kombinace. Funkce

$$\text{DEC}(a_1, a_0, b_1, b_0)$$

tedy zobrazí stavový index odpovídající tabulce 1 a funkce

$$\text{DEC}(a_1, a_0) \text{ a } \text{DEC}(b_1, b_0)$$

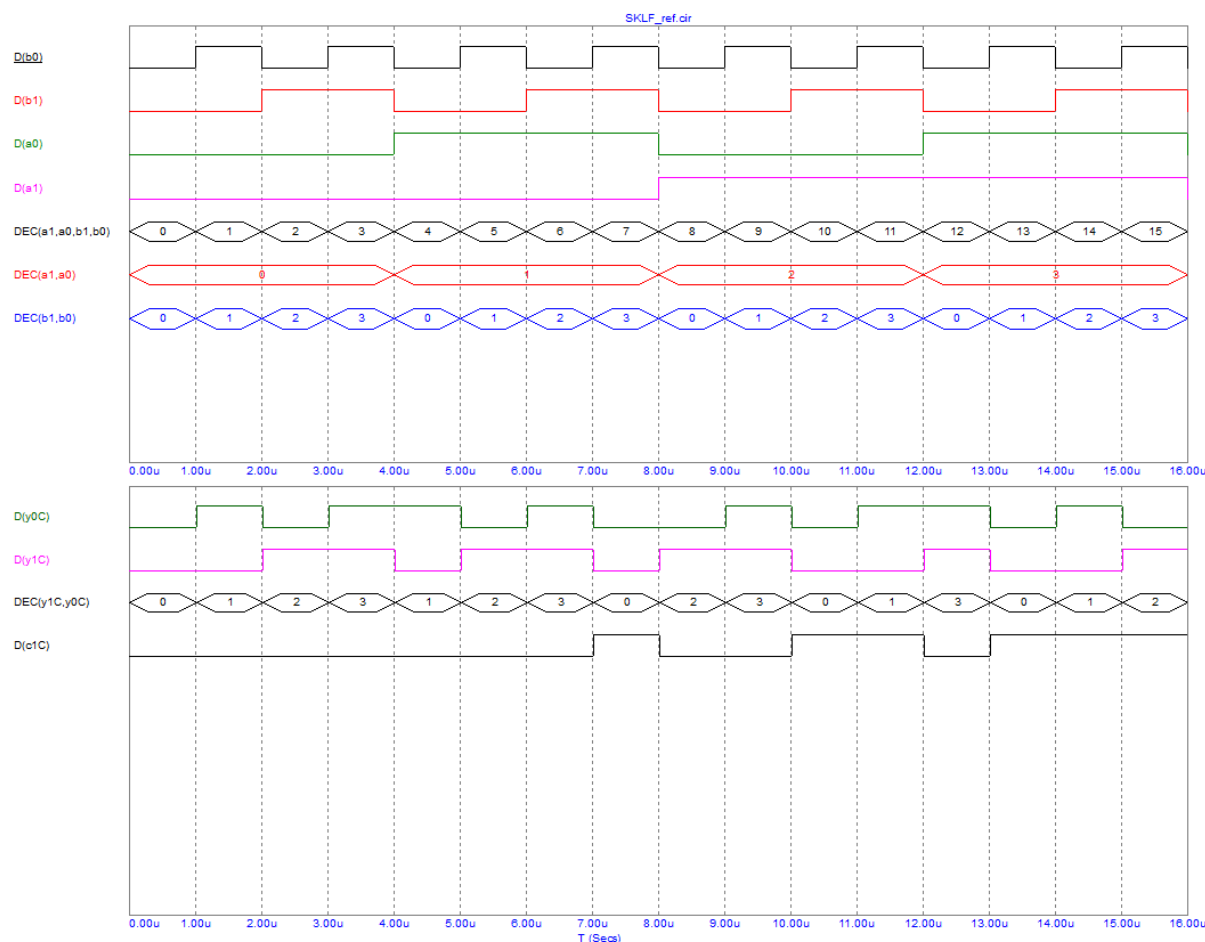
oba operandy v číselné podobě, viz obr. 3 a 4.

Je vhodné rozdělit zobrazení logických průběhů do dvou panelů (grafů) nad sebou. Prvních sedm řádků na obr. 3 má nastavené číslo 1 ve sloupci P. Zobrazí se tedy do panelu č. 1 (horní graf). **Všechny výstupy referenčního obvodu a obvodů podle bodů A) a B) zadání zobrazte do panelu č. 2.**

Ve shodě s nastavením na obr. 3 zvolte u všech výstupů rozsah osy X jako

$$\text{TMAX, TSTART, 1u}$$

aby se zobrazila mřížka s krokem 1μs pro snazší kontrolu výsledků.



Obr. 4. Zobrazení výsledků (ukázán jen referenční obvod).

Výstup referenčního obvodu zobrazte do spodního panelu přesně podle obr. 3 a 4. Kontrola funkce se provede srovnáním vstupních operandů a výstupů. Např. v intervalu mezi 11μs a 12μs jsou vstupní hodnoty 2_{DEC} a 3_{DEC}. Jejich součet je 5_{DEC}, což se zobrazí jako

$$(y1C = 0, y0C = 1) = 1_{DEC} + \text{přetečení do vyššího řádu } c1C = 1 \text{ (bitová váha 4)}$$

Pro obvod realizace **A)** zobrazte dále do panelu 2

DEC(y1A, y0A) – oba výstupní bity jako dekadické číslo

D(c1A) – přenos do vyššího řádu jako logický průběh

Pro obvod realizace **B)** zobrazte do panelu 2

D(y0B) – logický průběh nižšího bitu

Protokol ve formátu pdf obsahuje:

- ručně vyplněné a oskenované strany 3 až 5 (teoretická část),
- schéma v MicroCAPu,
- kopie okna s nastavením analýzy Transient,
- kopie grafu v čitelné podobě se všemi signály podle výše uvedených pokynů.

Současně vložte do e-Learningu soubor .cir s obvodem.