

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ
KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

BAKALÁŘSKÁ PRÁCE
Návrh sčítačky typu CLA

zadání

Abstrakt

Bakalářská práce prezentuje návrh sčítačky typu CLA. Cílem bylo pro 4-bitovou sčítačku CLA provést analýzu na úrovni logických členů a porovnat ji s odlišnými druhy řešení sčítačky. Sčítačky byly porovnávány hlavně podle doby zpoždění. Porovnáním architektur a rozdílných implementací bylo zjištěno, že je vhodné sčítačku navrhovat pro danou aplikaci a tím dosáhneme nejlepších výsledků. Simulací jsme zjistili, jak sčítačka CLA řeší problém s propagací přenosu a zkracuje dobu zpoždění.

Klíčová slova

Sčítačka, CMOS, Tranzistor, Logický člen, Architektura

Abstract

The bachelor thesis deals with the CLA type of Adders. The main goal is to analyze the 4-bit Adder and compare it with other types of Adders. During the comparison we especially focused on delay of the carry. We confirmed that for the best results it is better to consider the specific application of use of the Adder. During the simulation it was found out how the CLA Adder solves the issue of propagation delay and shorts the delay of the carry. The comparison was primarily based on the delay of carry. By comparison of different structures and implementations it was found that the best solution when designing an Adder is to consider the specific requirements of each application. Using computer simulation we found out how the CLA Adder deals with propagation delay issues and how it shorts the delay of carry.

Key words

Adder, CMOS, Transistor, Logic gate, Architecture

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni. Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce. Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 7.6.2014

Sebastian Veselý

.....
podpis

Obsah

1	ÚVOD	10
2	TECHNOLOGIE CMOS	11
2.1	Struktura CMOS	12
3	LOGICKÉ ČLENY	13
4	SČÍTAČKY	17
4.1	Poloviční sčítačka (Half adder - HA)	19
4.2	Úplná sčítačka (Full adder - FA)	20
4.3	Sčítačka s propagací přenosu (Ripple Carry Adder - RCA) . . .	21
4.4	Sčítačka s predikací přenosu (Carry Look-Ahead Adder - CLA)	22
4.4.1	Generování signálu P_i a G_i	22
5	Návrh jednotlivých bloků sčítaček	24
5.1	Úplná sčítačka	24
5.2	Sčítačka s propagací přenosu (RCA)	26
5.3	Sčítačka s predikcí přenosu (CLA)	30
5.4	Zhodnocení návrhu pro 4-bit sčítačku	32
6	Layout a floorplan	33

SEZNAM ZKRATEK

CMOS.....	Dvojice MOS tranzistorů N a P umístěná na jednom substrátu (Complementary Metal–Oxide–Semiconductor)
NMOS.....	Tranzistor s kanálem typu P a oblastmi typu N
PMOS.....	Tranzistor s kanálem typu N a oblastmi typu P
VLSI.....	Míra integrace polovodičových prvků (Very-large-scale-integration)
CLA.....	Sčítačka s predikací přenosu (Carry Look-Ahead Adder)
RCA.....	Sčítačka s propagací přenosu (Ripple Carry Adder)
CSL.....	Carry Select Adder
CSA.....	Carry Save Adder
CSK.....	Carry Skip Adder
HA.....	Poloviční sčítačka (Half Adder)
FA.....	Úplná sčítačka (Full Adder)
TFA.....	Úplná sčítačka využívající hradla Transmission gate
DUT.....	Testované zařízení (Device under testing)
LSB.....	Nejnižší bitová pozice (Least signification bit)
MSB.....	Nejvyšší bitová pozice (Most signification bit)
FO4.....	Paralelní zapojení čtyř invertorů na výstupu (Fan-out-4)

Seznam obrázků

2.1	Znázornění logiky typu „static CMOS“	11
3.1	Logický člen NOT na tranzistorové úrovni a logické úrovni	14
3.2	Logický člen NOR na tranzistorové a logické úrovni	14
3.3	Logický člen NAND na tranzistorové úrovni a logické úrovni	15
3.4	Logický člen XOR na tranzistorové úrovni a logické úrovni	16
4.1	Hierarchie rozdělení sčítaček	18
4.2	Poloviční sčítačka	19
4.3	Úplná sčítačka složená ze dvou Polovičních sčítaček	21
4.4	Obecné znázornění architektury Sčítačky s predikací přenosu	23
5.1	Simulace testovaných zařízení	24
5.2	Konvenční statická Úplná sčítačka na tranzistorové úrovni složená z 32 tranzistorů [1]	25
5.3	Úplná sčítačka transmission gate složená z 16 tranzistorů (TFA) [1]	25
5.4	Blokové zapojení Sčítačky s propagací přenosu (RCA)	28
5.5	Kritické zpoždění 4-bitové Sčítačky s propagací přenosu složené z logických členů	28
5.6	Kritické zpoždění 4-bitové sčítačky s propagací přenosu složené z konvenč- ních statických Úplných sčítaček	29
5.7	Kritické zpoždění 4-bitové sčítačky s propagací přenosu z Úplných sčítaček typu TFA	29
5.8	4-bitová Sčítačka s predikací přenosu	31
5.9	Obvod realizující přenos Sčítačky s predikací přenosu	31
5.10	Kritické zpoždění 4-bitové sčítačky architektury typu CLA	32
6.1	Layout logického členu NOT	33
6.2	Layout logického členu AND	34
6.3	Floorplan 2-bitové Úplné sčítačky složené z logických členů	34

Seznam tabulek

3.1	Pravdivostní tabulka hradla NOT	14
3.2	Pravdivostní tabulka hradla NOR	15
3.3	Pravdivostní tabulka hradla NAND	15
3.4	Pravdivostní tabulka hradla XOR	16
3.5	Porovnání zpoždění jednotlivých hradel	16
4.1	Pravdivostní tabulka Poloviční sčítačky	19
4.2	Pravdivostní tabulka Úplné sčítačky	20
5.1	Zpoždění jednotlivých Úplných sčítaček	25
5.2	Srovnání jednotlivých druhů 4-bitových sčítaček	32
6.1	Porovnání zpoždění pro návrh layout	34

1 ÚVOD

Předkládaná práce se zabývá problematikou sčítaček. Sčítačky jsou nezbytným zařízením pro zpracovávání základních aritmetických operací. Své využití nacházejí především v procesorech, kde jsou důležitou součástí aritmetické jednotky.

Cílem práce je realizace 4-bitové sčítačky. Protože 4-bitovou sčítačku je možné realizovat mnoha způsoby, zabývali jsme se jednotlivými výhodami, nevýhodami a omezeními dílčích zapojení. Pro sestavení je možné užít několik architektur, některými z nich se budeme dále zabývat. Cílovou architekturou je pro nás Sčítačka s predikací přenosu, která by měla představovat určité zrychlení přenosu. Práce je rozdělena na část teoretickou a praktickou.

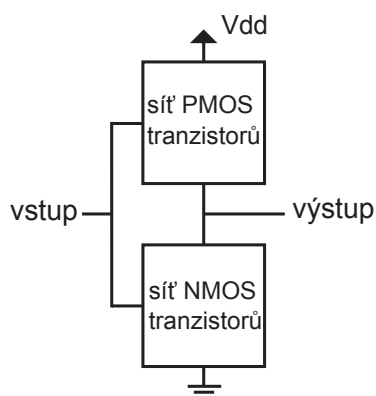
Teoretická část se soustředí obecně na technologii CMOS a konstrukci sčítaček. V kapitole 2 je stručně představena technologie CMOS a skupina „static CMOS“ a dále je popsána struktura tranzistoru CMOS. V kapitole 4 jsou vysvětleny základní principy funkce sčítaček a popsány rozdíly jejich jednotlivých řešení. Sčítačky jsou rozděleny podle druhů architektur a architektury jsou dále rozděleny na dílčí bloky, ze kterých se skládají. Pro popis funkce využíváme logických funkcí a pravdivostních tabulek.

Část praktická v kapitole 5 se zabývá konkrétní realizací určitých druhů sčítaček. Sestavili jsme logické členy potřebné pro návrhy některých sčítaček. Dále jsme realizovali sčítačku CLA, kterou jsme porovnávali s několika navrženými variantami implementací pro architekturu RCA. Z výsledků simulace jsme odečetli kritické doby zpoždění sčítaček a ty jsme dále porovnávali pro zhodnocení jednotlivých návrhů.

Po verifikaci schematického zapojení obvodu musí následovat realizace fyzického návrhu tímto způsobem návrhu se zabývá následující kapitola 6. Tento druh návrhu je bližším znázorněním reálné struktury tranzistorů CMOS. Provedení simulací v tomto návrhu by mělo být reálnějším přiblížením skutečné sčítačky. Pro tento druh návrhu provedeme simulace, které dále porovnáme s výsledky naměřenými v předešlé kapitole. Pro návrh sčítaček a logických členů byl využit software Electric VLSI Design a pro simulace LT Spice.

2 TECHNOLOGIE CMOS

Jde o kombinaci dvou typů MOS tranzistorů s kanály typu N a P. Provoz tranzistoru je řízený elektrickým polem. V logických obvodech, kde požadujeme binární výstup, se dnes používá nejčastěji skupina v angličtině nazývaná „static CMOS“. Obr. 2.1 naznačuje princip tohoto zapojení, kde ve spodní části sítě jsou umístěny tranzistory NMOS a v horní části sítě tranzistory PMOS. Důvodem tohoto umístění tranzistorů jsou jejich prahová napětí. Pokud je tranzistor zapojen jako spínač, prahová napětí tranzistoru NMOS/PMOS mohou způsobit degradovanou hodnotu logické úrovně. U tranzistoru PMOS je degradována logická 0 a pro tranzistor NMOS je to logická 1. Proto jsou tranzistory umístěny tak, aby každý z nich se využíval pouze pro hodnoty výstupu, které nejsou degradované. Tím zajistíme na výstupu silnou hodnotu logické úrovně. Toto řešení se také nazývá v angličtině „fully restored logic“ a značně zjednodušuje konstrukci obvodů. Druh této logiky je základem pro obvody s nízkým statickým ztrátovým výkonem v CMOS. V porovnání s ostatními druhy logiky, jako je „transmission gate“ nebo „pass logic“, vede použití logiky „static CMOS“ v některých případech k většímu počtu tranzistorů. [3]



Obrázek 2.1: Znázornění logiky typu „static CMOS“

2.1 Struktura CMOS

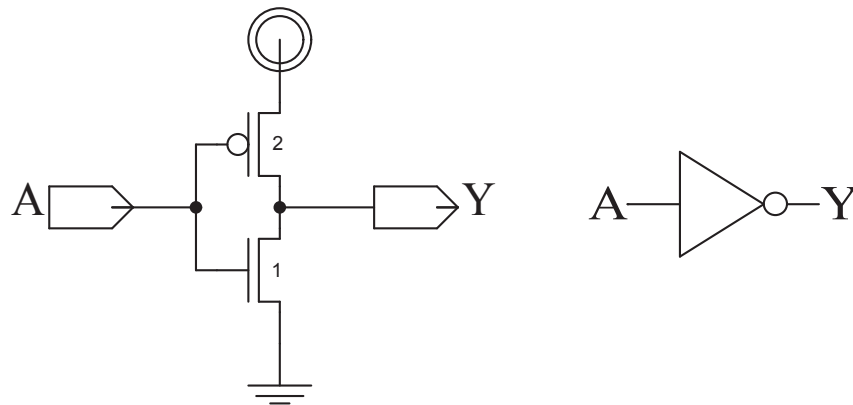
Pro konstrukce CMOS je základním materiálem křemík. Čistý křemík není příliš vodivý, takže vodivost se zvyšuje přidáním malého množství příměsí, nazývaných dopant. Dopant z V. skupiny periodické tabulky má o jeden valenční elektron více než křemík ze IV. skupiny a přebývající elektron se stává nosičem proudu. Protože nosičem proudu je záporně nabitý elektron, nazýváme polovodič typem N. Podobně dopant z III. skupiny má o jeden valenční elektron méně, chybějící elektron si přebírá z křemíku a vytváří díry. Díry jsou pozitivními nosiči, proto nazýváme tento polovodič typem P. Každý tranzistor se skládá z hradla (gate), což je vodivá vrstva odizolovaná velmi tenkou vrstvou oxidu křemíku a křemíkové desky, nazývané substrát. NMOS tranzistor má kanál typu P oblasti typu N sousedící s oblastí gate, nazývané source a drain. PMOS tranzistor má oblasti source a drain typu P a substrát typu N. V CMOS technologii, která obsahuje oba typy tranzistorů, je substrát pouze jednoho typu. Druhý tranzistor musí být umístěn ve speciální oblasti nazývané „well“. Tato oblast musí být dotována opačným dopantem atomu než substrát. Na gate je přiváděno napětí, které ovládá tok elektrického proudu mezi sourcem a drainem.

V případě NMOS tranzistoru se substrát obvykle uzemňuje, a přivedeme-li na gate nulový potenciál, tranzistor je vypnutý. Jestli na gate přivedeme napětí větší, než je napětí prahové, vytvoří se kolem něj elektrické pole a začne přitahovat volné elektrony. Ty vytvoří vodivý kanál a proud začne protékat mezi sourcem a drainem. V případě PMOS je situace opačná. Substrát je připojen na napájecí napětí a přivedením na gate napětí odpovídající napájecímu je tranzistor v nevodivém stavu, zatímco při přivedení nulového napětí má tranzistor nejmenší odpor kanálu [2] [3].

3 LOGICKÉ ČLENY

Pro sestavení sčítaček z logických členů byly zapotřebí určité logické členy. Tato kapitola zobrazuje jednotlivá zapojení logických členů na tranzistorové úrovni. Pro návrh logických členů byla užitá rodina „static CMOS”. Referenční šířku kanálu tranzistoru jsme zvolili $3\mu\text{m}$. V následujících schématech je referenční šířka kanálu označována jako číslo 1 a ostatní čísla jsou násobek referenční šířky. Délka kanálu se v jednotlivých schématech neliší a je stále $0,6\mu\text{m}$. Protože každý tranzistor má určitou velikost odporu, která je závislá na šířce kanálu, a majoritní nosiče tranzistorů PMOS mají horší mobilitu než NMOS, volíme šířku kanálu PMOS dvojnásobnou, aby jsme zajistili stejný odpor obou částí obvodu a tím i stejné velikosti vybíjecích i nabíjecích proudů. Jednotlivé logické členy jsme simulovali pro nejhorší možné vstupní kombinace, tak jsme změřili jejich maximální doby zpoždění, které mohou nastat. V Tab. 3.5 jsou uvedeny doby zpoždění jednotlivých logických členů. Doby zpoždění byly rozděleny na zpoždění doby nárůstu t_{pdr} a zpoždění doby poklesu t_{pdf} . Doby zpoždění jsme rozdělili, protože se pro jednotlivá zapojení mohou lišit. Rozdíly zpoždění mezi t_{pdr} a t_{pdf} jsou způsobeny sepnutým odlišným počtem tranzistorů. Doby zpoždění budou v následujících kapitolách využity pro výpočet předpokladu doby zpoždění sčítaček složených z logických členů. V následujících schématech A, B značí vstupní signál a Y značí signál výstupní.

Logický člen NOT



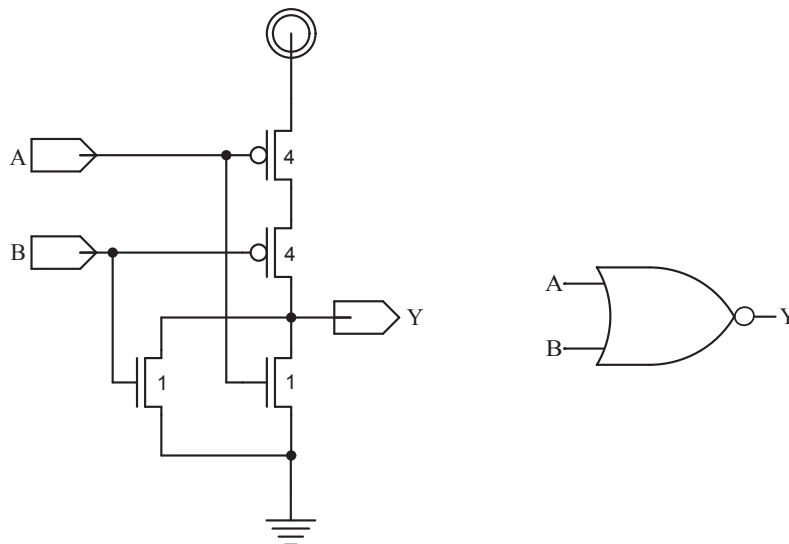
Obrázek 3.1: Logický člen NOT na tranzistorové úrovni a logické úrovni

A	Y
0	1
1	0

Tabulka 3.1: Pravdivostní tabulka hradla NOT

$$A = \bar{Y} \tag{3.1}$$

Logický člen NOR



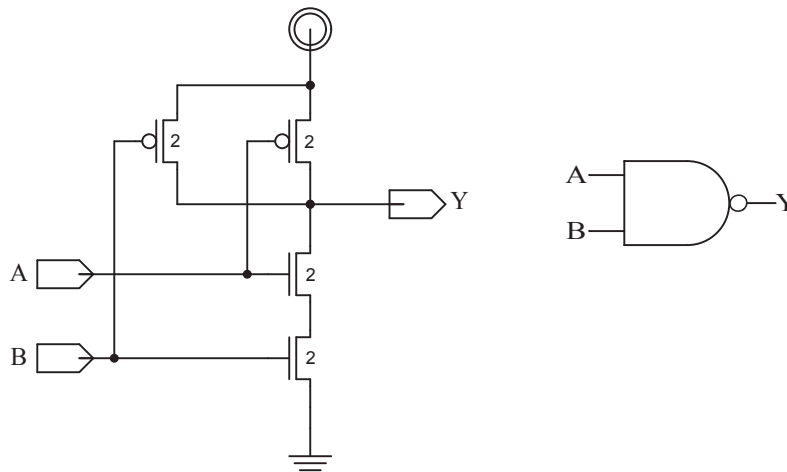
Obrázek 3.2: Logický člen NOR na tranzistorové a logické úrovni

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Tabulka 3.2: Pravdivostní tabulka hradla NOR

$$Y = \overline{A + B} = \overline{A} \cdot \overline{B} \quad (3.2)$$

Logický člen NAND



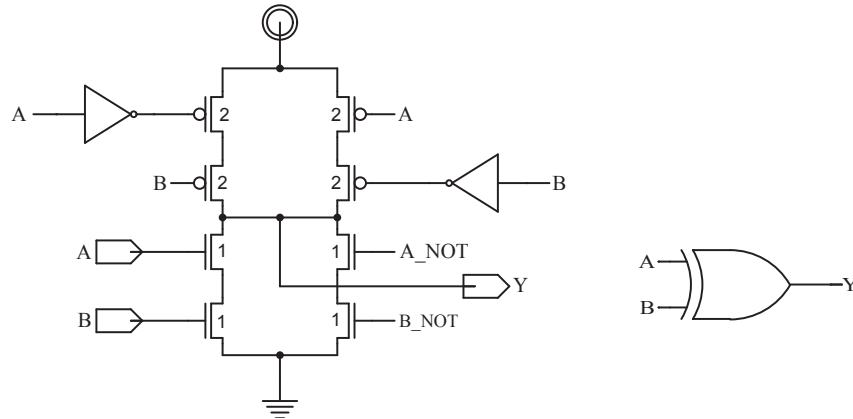
Obrázek 3.3: Logický člen NAND na tranzistorové úrovni a logické úrovni

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Tabulka 3.3: Pravdivostní tabulka hradla NAND

$$Y = \overline{A \cdot B} = \overline{A} + \overline{B} \quad (3.3)$$

Logický člen XOR



Obrázek 3.4: Logický člen XOR na tranzistorové úrovni a logické úrovni

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Tabulka 3.4: Pravdivostní tabulka hradla XOR

$$Y = A \oplus B \tag{3.4}$$

Logický člen	t_{pdr} [ns]	t_{pdf} [ns]	Počet tranzistorů
NOT	0,156	0,168	2
NOR2	0,104	0,253	4
NAND2	0,166	0,253	4
XOR2	0,303	0,241	12

Tabulka 3.5: Porovnání zpoždění jednotlivých hradel

4 SČÍTAČKY

Pro zpracování základních aritmetických operací je sčítání nezbytné. Používá se značně v mnoha VLSI návrzích a je zdaleka nejčastěji používanou operací v univerzálním systému a procesorech pro specifické použití. Také proto, že operace odčítání, násobení, dělení a výpočet adresy obvykle závisí na operaci sčítání, je sčítání vnímáno jako nezbytná součást aritmetické jednotky. Při sčítání je každý výsledný výstupní bit závislý na všech vstupních bitech, proto je důležitou vlastností sčítačky rychlost. Protichůdným požadavkem rychlosti je potřebná oblast na čipu, která se značně projevuje hlavně při zvyšování počtu bitů. I když některé typy mají lepší poměr rychlost/potřebná oblast na čipu, vždy je to do jisté míry kompromis mezi těmito požadavky a záleží na aplikaci, v níž chceme sčítačku použít [1]. V následující části bude uvedeno několik možností rozdělení sčítaček podle určitých hledisek a některými druhy se budeme zabývat.

Hierarchie sčítaček na Obr. 4.1 znázorňuje možnosti realizace n -bitové sčítačky. První rozdělení jsme zvolili podle typu architektury. Zvolené architektury se liší způsobem řešení zpracování přenosu z nižšího bitu na bity vyšší. Způsob řešení přenosu se právě odráží na již zmíněných důležitých kriteriích rychlosti a potřebné oblasti na čipu. Jednotlivé architektury je možné realizovat různými druhy implementací, proto další rozdělení je podle druhu implementace. Pro znázornění možnosti implementace jsme zvolili sčítačku RCA, která se skládá z Úplných sčítaček. Pro Úplnou sčítačku jsme dále zvolili rozdělení na realizaci ze dvou Polovičních sčítaček a logického členu OR, pouze z logických členů nebo pomocí tranzistorové implementace. Vytvořením hierarchie sčítaček jsme chtěli poukázat na různorodost sčítaček a možnost realizace sčítačky podle použití pro danou aplikaci.

Rozdělení sčítaček podle architektury:

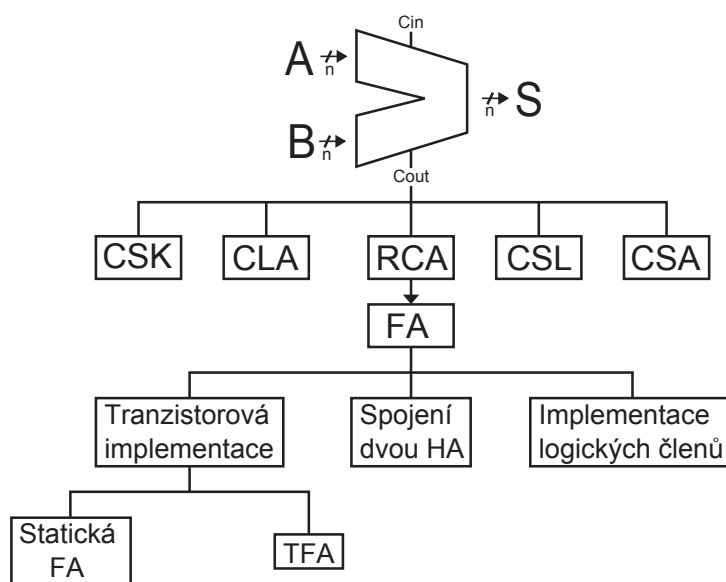
- Carry Look-Ahead Adder (CLA), Sčítačka s predikací přenosu
- Ripple Carry Adder (RCA), Sčítačka s propagací přenosu
- Carry Select Adder (CSL), Sčítačka s výběrem přenosu
- Carry Save Adder (CSA)
- Carry Skip Adder (CSK)

Rozdělení podle implementace:

- Úplná sčítačka (FA)
- Spojení dvou Polovičních sčítaček (HA)
- Implementace z logických členů

Implementace na tranzistorové úrovni:

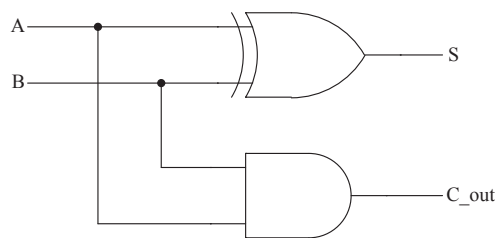
- Statická Úplná sčítačka (FA)
- Transmission Function Adder (TFA)



Obrázek 4.1: Hierarchie rozdělení sčítaček

4.1 Poloviční sčítačka (Half adder - HA)

Jedná se o nejjednodušší základní druh sčítačky. Lze ji vytvořit použitím dvou logických členů AND a XOR. Na vstup sčítačky přivádíme dvě jednobitová čísla. Výstupem sčítačky je dvoubitové číslo, kde nižší bit je součet jednotlivých bitů S a vyšší bit je přenos C_{out} do vyššího řádu. Protože sama nedokáže zpracovat přenos z předcházejícího řádu, lze ji použít pouze pro sčítání dvou jednobitových čísel. Na Obr. 4.2 je znázorněna Poloviční sčítačka složená z hradel XOR a AND. Funkci Poloviční sčítačky popisuje Tab. 4.1 a rovnice 4.1 a 4.2. Kde A , B představují vstupní operandy, S je výsledná suma operace a C_{out} je přenos aritmetické operace



Obrázek 4.2: Poloviční sčítačka

A	B	S	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabulka 4.1: Pravdivostní tabulka Poloviční sčítačky

$$S = A \oplus B \quad (4.1)$$

$$C_{out} = A \cdot B \quad (4.2)$$

4.2 Úplná sčítačka (Full adder - FA)

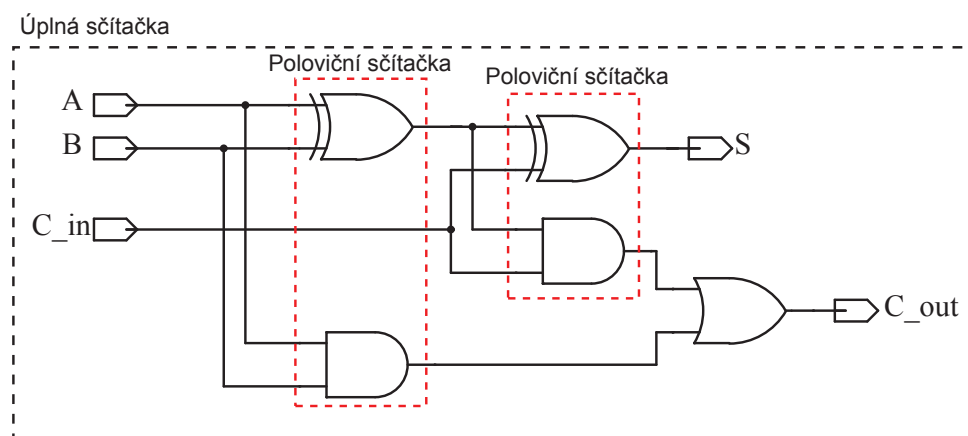
Sčítá dvě binární čísla a navíc je rozšířena o vstup C_{in} . Má tedy tři vstupy, dva pro jednotlivé vstupní bity a jeden pro přenos z předešlého součtu. Výstupy tvoří součet bitů a přenos stejně jako u Poloviční sčítačky. Pro úplnost je funkce popsána matematicky rovnicemi 4.3 a 4.4 a pravdivostní tabulkou Tab. 4.2. Zde stejně jako v předchozím případě A , B jsou vstupní operandy, S je výsledná suma operace, C_{out} je přenos aritmetické operace a C_{in} představuje vstup pro přenos z předchozího stupně. Na Obr. 4.3 je znázorněno, jak je možné Úplnou sčítačku konstruovat použitím dvou Polovičních sčítaček a hradla OR. Další možnosti, jak sestavit kompaktnější a rychlejší Úplnou sčítačku jsou implementace na tranzistorové úrovni. Optimalizace Úplné sčítačky na tranzistorové úrovni je popsána v kapitole 5.1.

C_{in}	A	B	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabulka 4.2: Pravdivostní tabulka Úplné sčítačky

$$S = A \oplus B \oplus C_{in} \quad (4.3)$$

$$C_{OUT} = (A \oplus B) \cdot C_{in} + A \cdot B \quad (4.4)$$



Obrázek 4.3: Úplná sčítačka složená ze dvou Polovičních sčítaček

4.3 Sčítačka s propagací přenosu (Ripple Carry Adder - RCA)

Základní částí RCA je Úplná sčítačka. Spojením C_{out} předešlé Full adder do C_{in} následující můžeme zhotovit vícebitovou sčítačku. Tímto zapojením do kaskády, lze zhotovit sčítačku pro libovolný počet bitů. Limitující pro RCA je zpoždění přenosu, to narůstá s počtem zpracovávaných bitů, protože je velmi závislé na délce cesty přenosu. V nejhorším případě je přenos od nejnižšího bitu až k nejvyššímu, tuto dobu nazýváme doba kritického zpoždění. V porovnání s ostatními architekturami sčítaček RCA zabírá nejmenší prostor a nabízí dobrý výkon pro náhodné vstupy. Velikost zabraného prostoru závisí stejně jako zpoždění na počtu zpracovávaných bitů. V případech, kdy potřebujeme vysokou rychlost a množství použitého hardwaru není rozhodující, je použití RCA pro aritmetické operace nevhodné [1].

Charakteristické zpoždění přenosového signálu podél řetězce kaskády FA způsobuje na výstupu falešný přechod. Přivedením na vstupy 4-bitové sčítačky RCA $A[0-3] = 0$, $B[0-3] = 1$, $C_{in} = 1$, by se ideálně měly výstupy $S[0-3] = 0$, nicméně kvůli zpoždění šíření přenosu jednotlivými bloky jsou součty nejprve 1 a postupně, jak přenos prochází jednotlivými Úplnými sčítačkami, se na výstupech objevuje 0. Tyto dynamické přenosy způsobují ztrátu energie, jež tvoří značnou část z celkové energie. Vhodným návrhem obvodu a optimalizací lze tuto chybu minimalizovat. Vzhledem k tomu, že se sčítačka RCA skládá z jednotlivých Úplných sčítaček, může být kritické zpoždění minimalizováno reimplementací Úplné sčítačky na tranzistorové úrovni [1].

4.4 Sčítačka s predikací přenosu (Carry Look-Ahead Adder - CLA)

Sčítačka tohoto typu snižuje dobu přenosu vytvořenou šířením přenosu zkrácením kritické cesty, která je nedostatkem sčítačky RCA. Hlavním cílem je zjistit, jestli v nižších stupních nastane přenos. Přenos je generován paralelně pro každý stupeň zvlášť, takže přenos CLA nezávisí na předchozím přenosu. Generování přenosu zpracovává samostatný obvod, který obsahuje velký počet logických členů. Obvod realizující přenos je zobrazen v kapitole 5.3 na Obr. 5.9, který prezentuje obvod generující přenos pro 4-bitové operandy. Při využití sčítačky CLA pro větší počty operandů se s narůstajícím počtem operandů doba zpoždění mezi vstupními operandy a výstupním součtem mění v logaritmické závislosti [1]. To umožňuje minimalizovat dobu zpoždění signálu přenosu pro větší počty operandů. V porovnání se sčítačkou RCA je sčítačka CLA rozměrnější a obsahuje větší počet tranzistorů kvůli realizaci obvodu pro přenos. Na Obr. 4.4 je blokově naznačena architektura sčítačky CLA. V tomto případě byla rozdělena na tři části. První část je Poloviční sčítačka, která ze vstupních operandů vytváří signály G_i a P_i . Další část je hlavní část sčítačky, která umožňuje paralelní přenos. Obsahuje dvoustupňovou implementaci logických členů a generuje signály C_{in} pro součet. Poslední částí je blok součtu, který generuje dílčí součty. Generování součtů lze realizovat logickým členem XOR. Vstupují do něj dva signály C_{in} z předešlého přenosu a P_i , který vzniká exkluzivním součtem vstupních operandů A_i a B_i . Podle kombinace C_{in} a P_i se generuje součet.

4.4.1 Generování signálu P_i a G_i

Podle určité kombinace vstupů A_i a B_i signál šíření P_i rozhoduje, zda se má carry-in šířit do i -tého bloku na výstup, zatímco signál G_i rozhoduje, jak má být nastaven carry-out nezávisle ze vstupů.

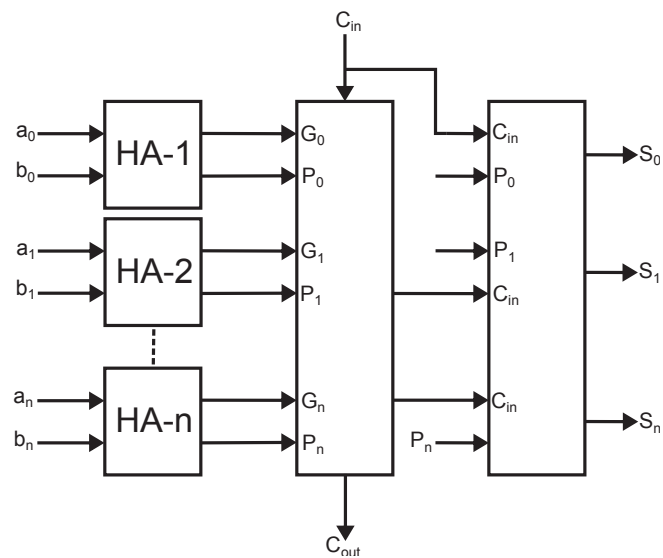
$$G_i = A_i \cdot B_i \quad (4.5)$$

$$P_i = A_i \oplus B_i \quad (4.6)$$

Z rovnic 4.5 a 4.6 je zřejmé, že v případě, kde $A_i = B_i = 1$, se generuje přenos 1 na i -té pozici, zatímco, když $A_i = B_i = 0$, generuje se místo toho přenos 0. Šíření přenosu na výstup i -tého bloku nastává, když $A_i \neq B_i$, proto když $C_{in} = 1$ a $A_i \neq B_i$ pro $i = 0, 1, 2, 3, 4, 5$, pak se bude C_{in} šířit až na pátou bitovou pozici. Rovnice 4.7 a 4.8 znázorňují součet a předpoklad přenosu pro i -tý stupeň [1].

$$S_i = P_i \oplus C_i \tag{4.7}$$

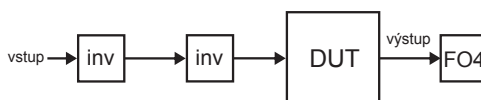
$$C_{i+1} = G_i + P_i \cdot C_i \tag{4.8}$$



Obrázek 4.4: Obecné znázornění architektury Sčítačky s predikací přenosu

5 Návrh jednotlivých bloků sčítaček

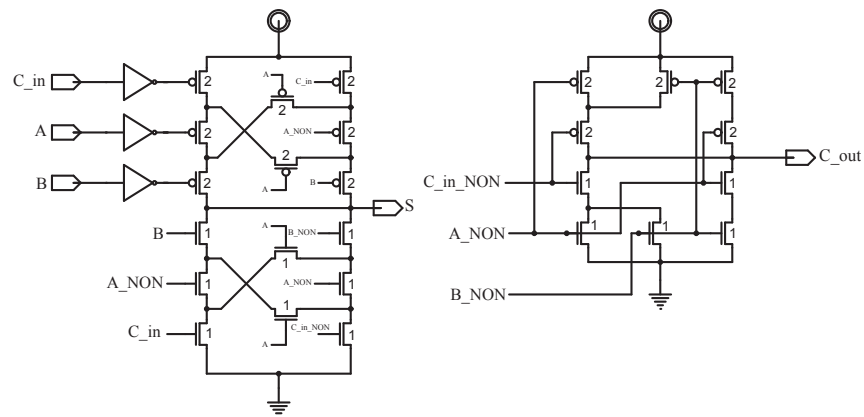
Pro srovnání parametrů sčítaček byly jednotlivé architektury simulovány. Pro méně kritické návrhy se užívá primárně dvou architektur, RCA a CLA. Proto pro porovnání Sčítačky s predikací přenosu byly sestaveny rozdílné realizace Sčítačky s propagací přenosu. Dále byla simulována jednotlivá zapojení a měřena jejich zpoždění při průchodu signálu kritickou cestou. Pro simulaci kritické doby zpoždění bylo potřeba nastavit vstupní operandy na nejproblematictější možnou kombinaci, která může nastat. Jednou z možností, jak už bylo zmíněno v teoretické části v kapitole 4.3, je současná změna $A[0-3]$, C_{in} z 0 do 1 při $B[0-3] = 0$. V reálném zařízení je sčítačka připojena na určitou vstupní a výstupní kapacitu. Nabíjení a vybíjení kapacity značně prodlužuje doby náběhu i doby doběhu. Proto pro reálnější simulaci byly sčítačky zatíženy vytvořenou kapacitou. Na vstupu představuje vstupní kapacitu sériové zapojení dvou invertorů, výstup byl zatížen vstupní kapacitou čtyř paralelně řazených invertorů. Zapojení čtyř paralelních invertorů je známo jako zapojení fan-out 4 (FO4). Zapojení fan-out 4 je běžně využíváno pro simulace zařízení. Blokové zapojení simulace je zobrazeno na Obr. 5.1.



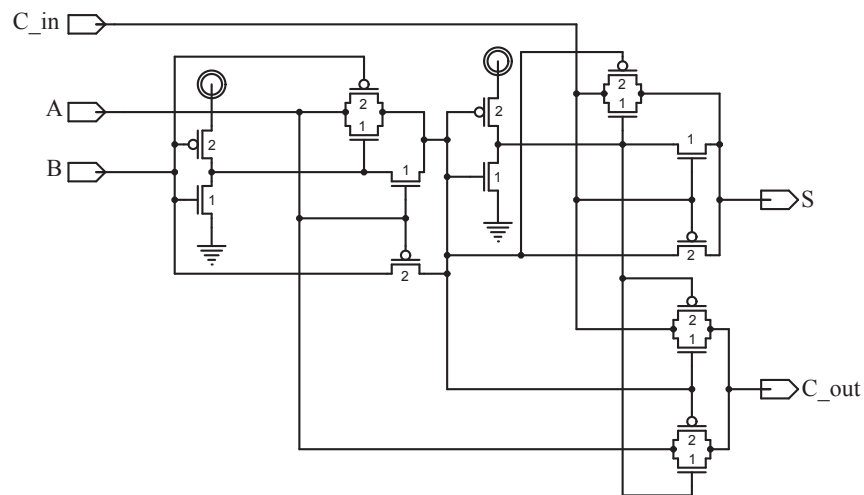
Obrázek 5.1: Simulace testovaných zařízení

5.1 Úplná sčítačka

Simulovány byly tři úplné sčítačky. První sčítačka je složena ze dvou Polovičních sčítaček a hradla OR. Zapojení sčítačky bylo už uváděno v předešlé kapitole 4.2 na Obr. 4.3. Následující dvě zapojení sčítaček jsou implementace na tranzistorové úrovni. První je konvenční statická sčítačka složená z 32 tranzistorů, převzato z [1], a je zobrazena na Obr. 5.2. Druhou úplnou sčítačkou složenou z 16 tranzistorů je Transmission Function Adder (TFA), její zapojení je uvedeno na Obr. 5.3. Schéma sčítačky TFA bylo převzato z [1] a je možno jej realizovat i pomocí jiných zapojení s odlišným počtem tranzistorů. Konvenční statická sčítačka využívá zapojení tranzistorů jako třívstupového hradla XOR a to kvůli velkému počtu kombinací vstupů vede k většímu počtu tranzistorů v porovnání s TFA, zatímco CMOS TFA je založená na hradlech transmission gate (TG), která implementují funkci XOR. Vzhledem k menšímu počtu tranzistorů obvod vykazuje menší statickou spotřebu. Tranzistorové implementace jsou kompaktnější, umožňují optimalizaci vhodným nastavením šířek kanálu pro určité aplikace. Tím je možné dosáhnout lepších parametrů. Zatímco sčítačka sestavená z logických členů není cílena na danou technologii, má širokou aplikovatelnost, ale nedosahuje takových parametrů.



Obrázek 5.2: Konvenční statická Úplná sčítačka na tranzistorové úrovni složená z 32 tranzistorů [1]



Obrázek 5.3: Úplná sčítačka transmission gate složená z 16 tranzistorů (TFA) [1]

Úplná sčítačka	simulace		odhad		počet tranzistorů
	$tp_{přenos}$	tpd_{sumy}	$tp_{přenos}$ (ns)	tpd_{sumy} (ns)	
TFA 16	0,246	0,538	-	-	16
FA z log. členů	0,982	0,758	1,263	0,606	36
Konvenční FA 32	0,533	0,688	-	-	32

Tabulka 5.1: Zpoždění jednotlivých Úplných sčítaček

5.2 Sčítačka s propagací přenosu (RCA)

Pro zapojení sčítačky RCA byla zkonstruována tři zapojení Úplných sčítaček, jež byla popsána v předešlých kapitolách. Simulované doby zpoždění jsou vyhodnoceny pro šířku operandu jednotlivých sčítaček. Pro Úplnou sčítačku složenou z logických členů byly provedeny předpoklady kritické doby zpoždění. Celková doba zpoždění se skládá ze dvou částí a každá z částí zahrnuje zpoždění jednotlivých hradel, přes která musí signál při kritické cestě projít. Rovnice 5.1 představuje dobu zpoždění pro součet a rovnice 5.2 pro přenos.

Kritická cesta sčítačky RCA se skládá ze dvou částí:

a) Kritická cesta pro výpočet součtu:

$$tp_{suma} = tp_{XOR} + tp_{XOR} \quad (5.1)$$

b) Kritická cesta pro přenos:

$$tp_{carry} = tp_{XOR} + tp_{AND} + tp_{OR} \quad (5.2)$$

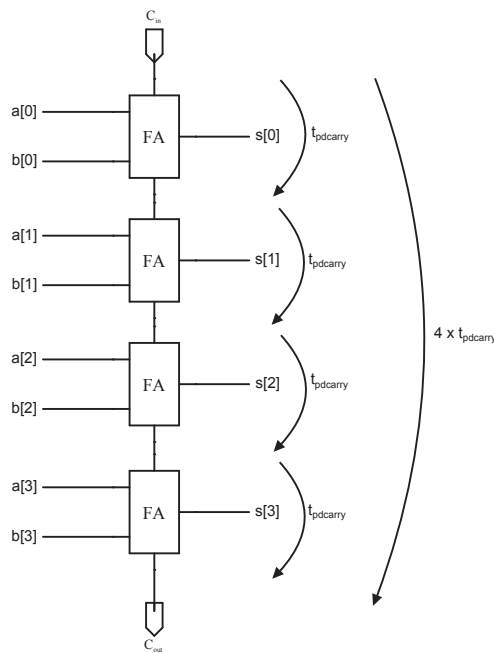
Zpoždění součtu je složeno ze zpoždění dvou logických členů. Doba zpoždění pro součet probíhá naráz ve všech čtyřech stupních, jednotlivé bity vstupují paralelně a stejně tak vychází jejich součty, zatímco signál carry musí projít celým přenosovým řetězcem. Doba přenosu je složena z průchodu třemi logickými členy. Pro 4-bitovou sčítačku při přenosu od LSB až MSB signál projde čtyřmi Úplnými sčítačkami, takže dohromady musí projít přes 12 logických členů. Z toho je zřejmé, že největší zpoždění sčítačky představuje cesta přenosu. Proto je pro sčítačku RCA rozhodující doba přenosu $tp_{přenos}$, kterou požadujeme co nejnižší. Na Obr. 5.4 je blokové zapojení sčítačky RCA s naznačenými dobami přenosu, z kterých se skládá celková doba přenosu. Pro sčítačky RCA je poslední ustálený průběh C_{out} , protože až po ustálení $S[3]$ může výstup C_{out} nabýt správné hodnoty. V případě sčítačky RCA se součty provedou velmi rychle a na výstupu pak dostáváme logickou hodnotu 1. To však není správná hodnota součtu, tato hodnota je způsobena šířením přenosu celým řetězcem. Vzhledem k napájecímu napětí 5V jsme dobu zpoždění měřili tak, jak je pro CMOS běžné v 50 % napájecího napětí, to je tedy 2,5V.

Největší zpoždění jsme naměřili pro sčítačku RCA složenou z logických členů. Na Obr. 5.5 graf ilustruje, jak se zpoždění šíří jednotlivými sčítačkami. Součet $S[0]$ se ustálí hned po provedení součtu vstupních operandů, protože signál C_{in} je přiveden na vstup. Součty

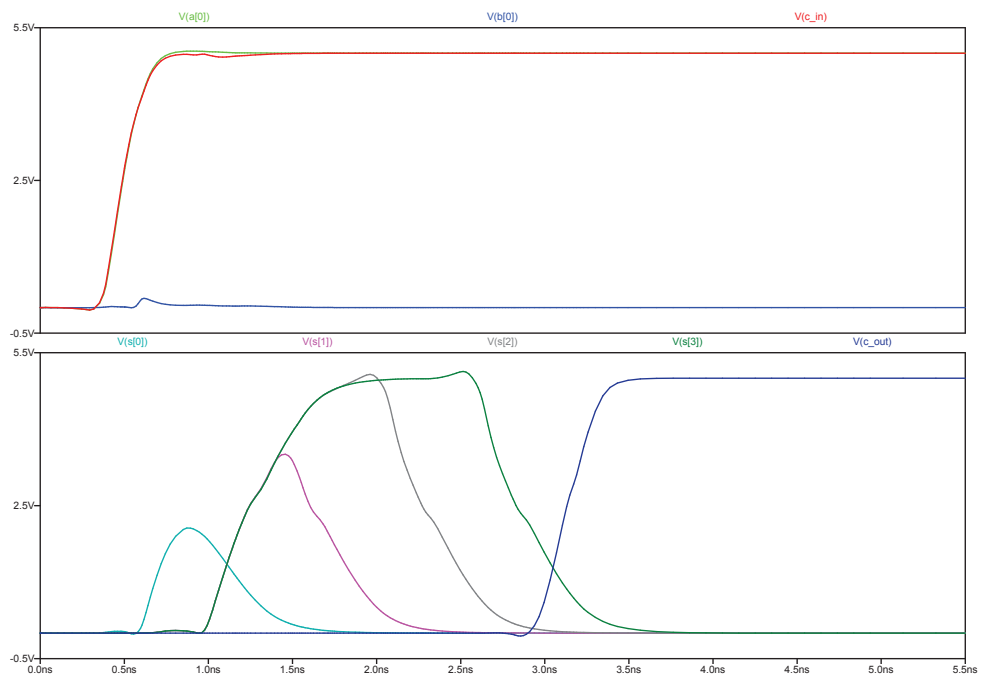
S[1-3] se postupně ustalují až po provedení předchozího součtu. Po dokončení posledního součtu se nastaví C_{out} .

Tranzistorové implementace se také skládají ze zpoždění součtu a přenosu a stejně jako u sčítačky z logických členů součet probíhá současně a přenos se šíří postupně, ale jednotlivé doby jsou minimalizovány s využitím tranzistorové implementace navržené přímo pro sčítání a přenos. Sčítačka RCA na Obr. 5.6 složená z konvenčních statických Úplných sčítaček má průběhy stejného charakteru, ale kritická doba zpoždění vychází o jednu nanosekundu menší. To je způsobeno zapojením sčítačky z logických členů, kde, jak vyplývá z grafu, už první součet, který není závislý na přenosu, trvá podstatně déle než u konvenční statické sčítačky. U tranzistorové implementace lze navíc ještě manipulovat s nastavením šířky kanálu jednotlivých tranzistorů. Vhodným nastavením šířek kanálu pro obvod realizující přenos by bylo možné dobu kritického zpoždění vylepšit. Vzhledem k tomu, že má i menší počet tranzistorů, bude vykazovat i menší spotřebu. Kompaktnější návrh má za následek menší obsaženou plochu.

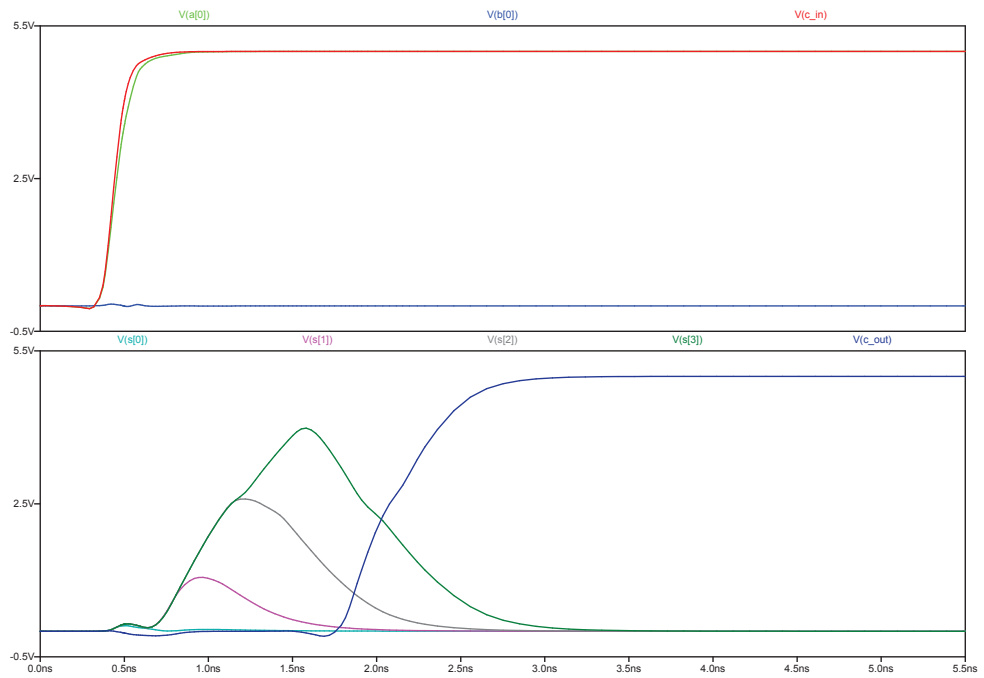
Nejlepšími parametry ze všech sčítaček disponuje sčítačka RCA s použitím logiky transmission gate neboli TFA. Graf kritického zpoždění sčítačky TFA je zobrazen na Obr. 5.7. Druh této logiky je naprosto odlišný oproti static CMOS a jeho řešení vyžaduje mnohem složitější návrh, ale dosahuje velmi dobrých parametrů. Kritická doba zpoždění je velmi malá v porovnání s ostatními uvedenými druhy sčítaček. Sčítačka TFA pracuje natolik rychle, že výstupy S[0-3] ani nepřesáhly rozhodovací úroveň, takže výstupy zůstaly po celou dobu 0. Výsledná doba kritického zpoždění závisela na výstupním bitu C_{out} . Ze všech uvedených sčítaček má nejmenší počet implementovaných tranzistorů na jednu Úplnou sčítačku.



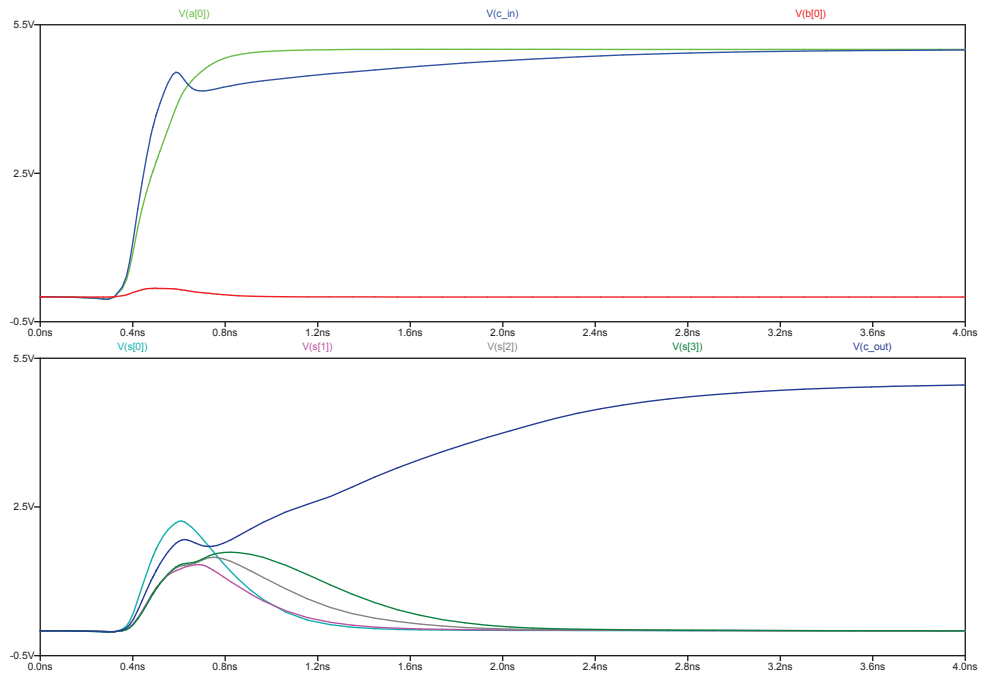
Obrázek 5.4: Blokové zapojení Sčítačky s propagací přenosu (RCA)



Obrázek 5.5: Kritické zpoždění 4-bitové Sčítačky s propagací přenosu složené z logických členů



Obrázek 5.6: Kritické zpoždění 4-bitové sčítačky s propagací přenosu složené z konvenčních statických Úplných sčítaček



Obrázek 5.7: Kritické zpoždění 4-bitové sčítačky s propagací přenosu z Úplných sčítaček typu TFA

5.3 Sčítačka s predikcí přenosu (CLA)

Sčítačka s predikací přenosu byla realizována pomocí logických členů. Zapojení logických členů je znázorněno na Obr. 5.8. Obvod realizující přenos je zobrazen na Obr. 5.9. Rovnice 5.3 a 5.4 představují předpoklad sčítačky CLA pro kritickou dobu zpoždění. Celková doba zpoždění se skládá stejně jako u RCA ze dvou částí.

Kritická cesta sčítačky CLA se skládá ze dvou částí:

a) *Kritická cesta pro výpočet součtu:*

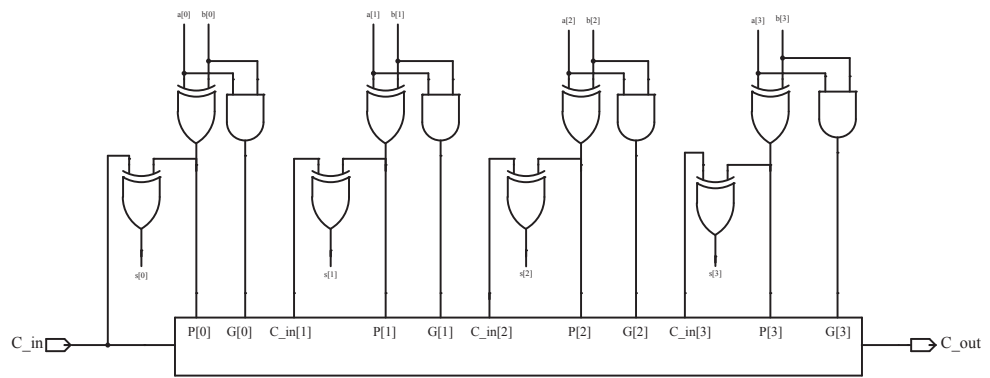
$$tp_{suma} = tp_{XOR} + tp_{AND} + tp_{OR} + tp_{XOR} \quad (5.3)$$

b) *Kritická cesta pro přenos:*

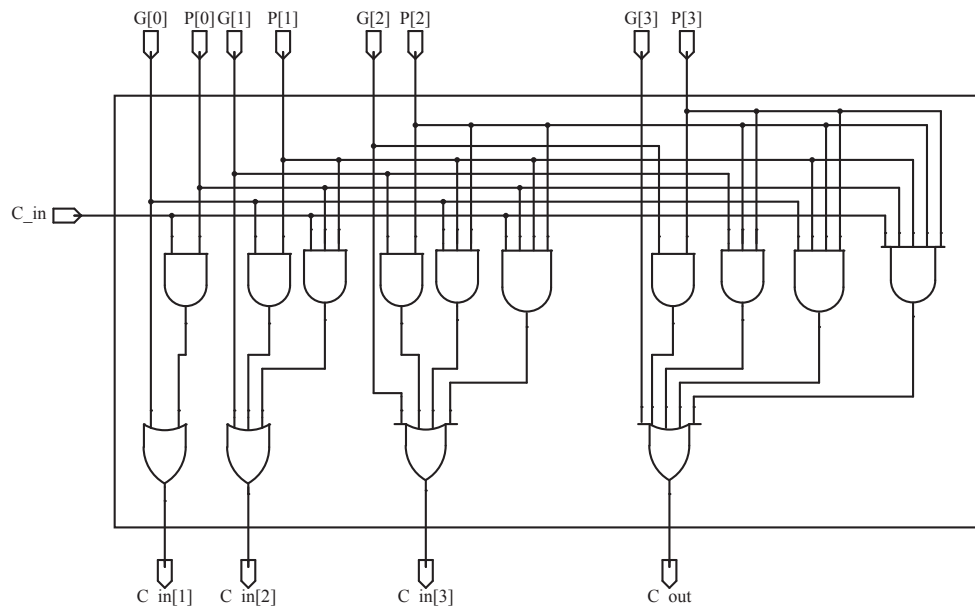
$$tp_{přenos} = tp_{XOR} + tp_{AND} + tp_{OR} \quad (5.4)$$

Zlepšení oproti RCA představuje doba přenosu, která probíhá paralelně, stejně jako součet, takže i při větším počtu bitů je cesta přenosu stále složena jen ze zpoždění tří logických členů. Z toho lze usuzovat, že při zvětšování počtu operandů by se doba zpoždění neměla příliš měnit. Na Obr. 5.9 je znázorněn obvod realizující přenos pro 4-bitovou sčítačku. Pro sestavení takového obvodu je zapotřebí použít vícevstupová hradla. S nárůstem počtu vstupů roste kapacita vstupů v důsledku přidávání tranzistorů při realizaci vícevstupových logických členů. Tím se samozřejmě zvyšuje latence. Se zvyšujícím se počtem bitů roste i zpoždění, ale nárůst zpoždění není v porovnání se sčítačkou RCA tak velký.

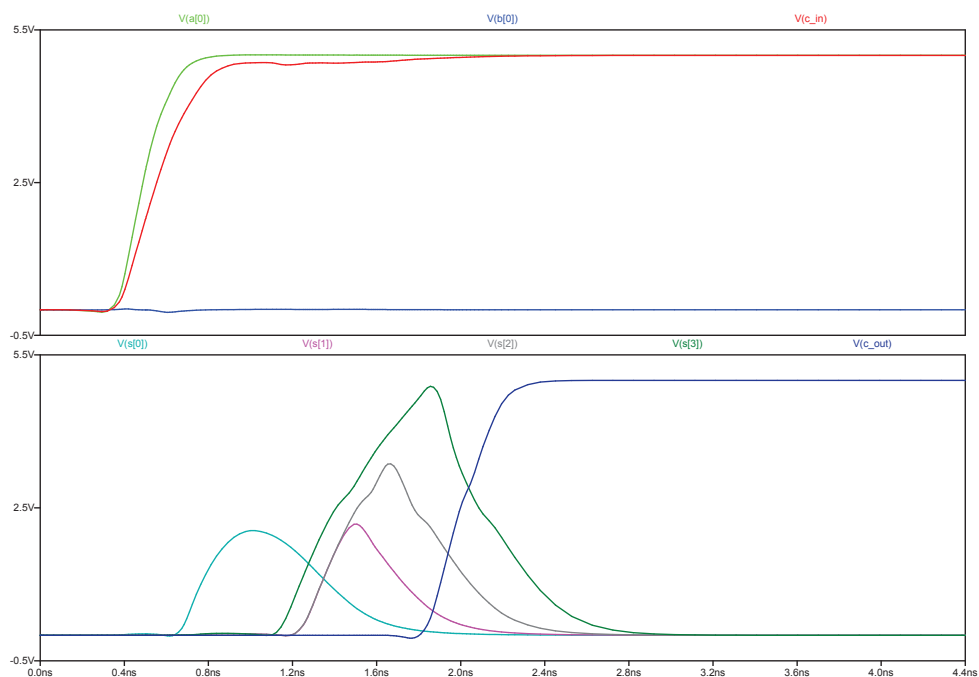
Při simulaci sčítačky CLA byl posledním ustáleným průběhem součet S[3]. Z tohoto výsledku je patrné, že sčítačka CLA je řešením toho, jak zkrátit velké zpoždění způsobené přenosem. Z Obr. 5.10 je patrné, jak jednotlivé součty probíhají nezávisle na sobě. Přestože jsme pro realizaci sčítačky CLA použili logické členy, což se v případě sčítačky RCA ukázalo jako nejhorší řešení, pro sčítačku CLA je kritická doba zpoždění poměrně malá. Ze všech simulovaných sčítaček v této práci vyšla kritická doba zpoždění sčítačky CLA druhá nejmenší. Protože simulace byly prováděny pouze pro čtyřbitové operandy, výhoda predikace přenosu není ve výsledku tolik patrná. Výhoda architektury CLA se spíše uplatní pro sčítání vícebitových operandů.



Obrázek 5.8: 4-bitová Sčítačka s predikací přenosu



Obrázek 5.9: Obvod realizující přenos Sčítačky s predikací přenosu



Obrázek 5.10: Kritické zpoždění 4-bitové sčítačky architektury typu CLA

5.4 Zhodnocení návrhu pro 4-bit sčítačku

Tab. 3.5 prezentuje výsledné odečtené doby ze simulace pro výše uvedené 4-bitové sčítačky. Pro sčítačky složené z logických členů jsme provedli odhad zpoždění. Odhady pro sčítačku RCA složenou z logických členů jsme vypočítali z rovnic 5.1 a 5.2, kde jsme za logické členy dosadili jejich doby zpoždění uvedené v tabulce 3.5. Stejným způsobem jsme provedli odhad pro sčítačku CLA, kde jsme dosadili změřené doby zpoždění do rovnic 5.3 a 5.4. Rozdíly mezi odhadnutými dobami zpoždění a dobami zpoždění odečtenými ze simulace jsou způsobeny tím, že pro odhad byly použity naměřené doby zpoždění logických členů, které byly odečteny taktéž ze simulace. Při odečtení doby zpoždění logických členů na výstup byla připojena vstupní kapacita fan-out 4, což představuje větší kapacitu než tu, na kterou jsou připojeny v obvodu sčítačky. V případě sčítačky RCA se tato chyba ještě zvětšila, protože jsme museli zároveň museli zvážit dobu přenosu, kterou jsme pro 4-bitovou sčítačku vynásobili čtyřmi. Z toho lze usoudit, že odhad byl skutečně velmi hrubý a příliš se neshoduje.

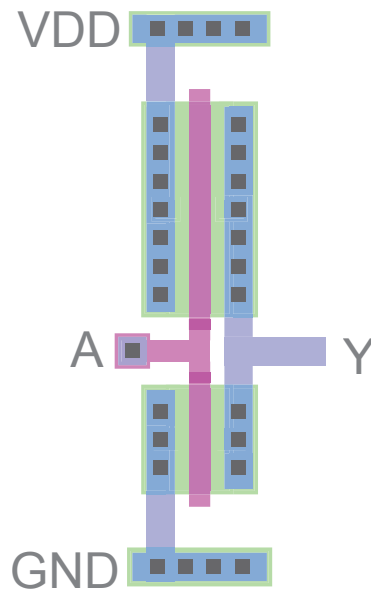
Sčítačka	Zpoždění (ns)	Odhad zpoždění (ns)	Počet tranzistorů
RCA z Logických členů	2,640	5,658	144
static CMOS 32 tr. RCA	1,622	-	128
TFA 16 tr. RCA	0,639	-	64
CLA	1,598	2,593	236

Tabulka 5.2: Srovnání jednotlivých druhů 4-bitových sčítaček

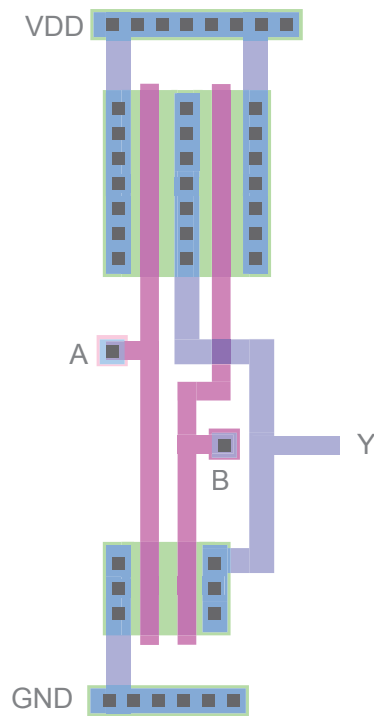
6 Layout a floorplan

Layout je zapojení tranzistorů CMOS, které odpovídá jejich fyzikální struktuře. Jak vypovídá název tohoto návrhu, jedná se o pohled shora. V layout návrhu jsou uvažovány kapacity vodičů, zatímco v schematicém návrhu se počítá pouze se součástkami. Z tohoto pohledu je layout dokonalejším přiblížením reálnému zapojení při simulaci. Pomocí layoutu můžeme sestavit jednotlivé funkční bloky, které lze spojovat stejně jako ve schematicém návrhu. Z důvodu časové náročnosti při seznamování s prostředím softwaru a následné realizaci jednotlivých zapojení nezbyl dostatek času na zrealizování sčítaček tohoto druhu návrhu. Proto jsme pro znázornění layoutu sestavili dva návrhy logických členů. První návrh představuje logický člen NOT, který je znázorněn na Obr. 6.1. Druhý návrh logického členu NAND je zobrazen na Obr. 6.2. Pro oba návrhy jsme provedli simulaci ve stejném zapojení jako předešlých kapitolách. Pro porovnání rozdílu doby zpoždění logických členů v návrhu layout jsme změřili doby zpoždění, které jsou uvedeny v Tab. 6.1.

Zapojením funkčních bloků vzniká floorplan. Pro znázornění floorplanu jsme realizovali návrh 2-bitové Úplné sčítačky složené z logických členů pouze v grafickém editoru. Návrh je znázorněn na Obr. 6.3. Ve skutečnosti jsou zapojení velice malá. Aby bylo možné realizovat propojení a zároveň dodržovat vzdálenosti vodičů, aby nedošlo k propojení nechtěných částí, realizuje se propojování vodičů v layoutu i ve floorplanu ve vrstvách. Jednotlivé barvy značí určitou vrstvu, kde se vodič nachází.



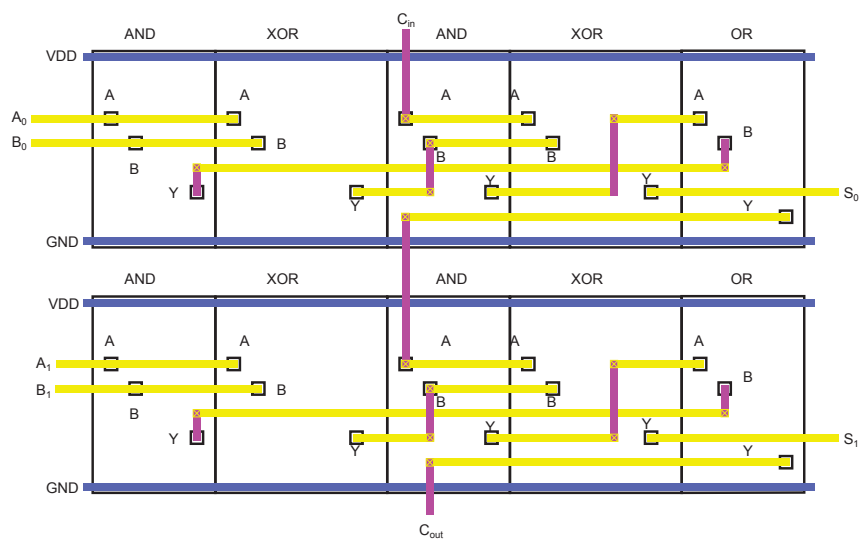
Obrázek 6.1: Layout logického členu NOT



Obrázek 6.2: Layout logického členu AND

Logický člen	t_{pdr}	t_{pdf}	Počet tranzistorů
NOT	0,220	0,256	2
NAND	0,244	0,412	4

Tabulka 6.1: Porovnání zpoždění pro návrh layout



Obrázek 6.3: Floorplan 2-bitové Úplně sčítačky složené z logických členu

Závěr

Práce byla zaměřena na návrh Sčítačky s predikací přenosu a jeho simulací. Návrh byl porovnáván s architekturou RCA odlišných implementací. Provedením simulací bylo zjištěno, že Sčítačka s predikací přenosu je řešením pro zrychlení přenosu. V porovnání s architekturou RCA, kdy jsme použili stejné logické členy jako pro realizaci Sčítačky s predikací přenosu, vyšla kritická doba zpoždění podstatně nižší. Na druhou stranu lepším řešením pro návrh sčítaček s menším počtem bitů by mohly být Sčítačky s propagací přenosu, které využívají Úplné sčítačky realizované na tranzistorové úrovni. Jejich kritická doba zpoždění vychází srovnatelná se Sčítačkou s predikací přenosu, v některých případech je dokonce i kratší. Musíme však zohlednit lineární nárůst doby zpoždění s počtem vstupních operandů pro architekturu RCA, takže pro vyšší počty bitů by architektura CLA byla výhodou.

V druhé praktické části jsme se zabývali layoutem a floorplanem. Bohužel kvůli časové náročnosti seznámení se s programem a realizací předešlých návrhů a jejich jednotlivých simulací jsme nestihli realizovat obdobná zapojení jako ve schematické části. Pro představu realizace v návrhu layout jsme sestavili dvojici logických členů NOT a NAND. Zaměřili jsme se na jejich doby zpoždění a porovnali jsme je s dobami zpoždění stejných členů realizovaných ve schematické části. Doby zpoždění změřené v návrhu layout vyšly větší. To je způsobeno tím, že ve schematickém návrhu jsou zanedbány kapacity jednotlivých vodičů. Dále jsme pro znázornění floorplanu navrhli v grafickém editoru zapojení Úplné sčítačky složené z logických členů.

Výsledkem této bakalářské práce je tedy schematický návrh 4-bitové Sčítačky s predikací přenosu a další návrhy sčítaček architektury RCA za použití rozdílných implementací, které slouží k porovnání. Všechny návrhy byly realizovány v softwaru ElectricVLSI design a simulovány v LT Spice.

Reference

- [1] YEO, Kiat Seng a Kaushik ROY. Low voltage, low power VLSI subsystems. New York: McGraw-Hill, c2005, xxi, 293 p. ISBN 00-714-3786-X.
- [2] ETIENNE SICARD, Sonia Delmas Bendhia a Kaushik ROY. Basics of CMOS cell design. New York: McGraw-Hill, 2007, xxi, 293 p. ISBN 00-715-0906-2.
- [3] WESTE, Neil H.E. a David Money HARRIS. CMOS VLSI Design: A Circuits and Systems Perspective. United State of America: Addison-Wesley, 2011, xxi, 293 p.