

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: BA2612 Elektrotechnika a informatika

Studijní obor: 2612R011 Elektronické informační a řídicí systémy

Ovladač řízení a rychlosti modelu autíčka

Speed and Revolving Controller of Car Model

Bakalářská práce

Autor: **Antonín Puncman**

Vedoucí práce: Ing. Zbyněk Mader, Ph.D.

V Turnově 16.5. 2010

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje Zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne úhradu nákladů, které vynaložila na vytvoření díla, až do její skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce.

Datum:

Podpis:

Abstrakt

Cílem této bakalářské práce je navržení a sestrojení vzdáleně proporcionálně ovládaného modelu autíčka včetně návrhu, konstrukce a realizace všech elektronických systémů a ovládacích prvků a jejich reálné odzkoušení. Klíčovou podmínkou k realizaci samotných číslcových výpočtů je využití pouze hradlové logiky CPLD (complex programmable logic device) a nikoliv mikroprocesoru. V kapitole „Ovladač“ je popsáno řešení samotného ovládacího prvku a převodu analogové veličiny otáčení potenciometrem na číslcovou podobu srozumitelnou pro CPLD a to včetně kompletního návrhu zapojení. V kapitole „Servo“ pak práce popisuje princip a možnosti nejdostupnějšího, výstupem CPLD ovládaného, servomechanismu. V této kapitole je taky obsažena problematika a řešení komunikace mezi CPLD a servomechanismem včetně jejího grafického vyobrazení. Kapitola „Motor“ pojednává o pohonném ústrojí modelu a jeho zapojení. Popisuje sestavení hradlové logiky, která prostřednictvím pulzně šířkového modulování výstupního signálu roztáčí, po zesílení v kapitole navrženým budičem, stejnosměrný elektromotor. Poslední kapitola o řídicí vazbě rozvíjí a následně uvádí v praxi možnosti aktivního elektronického systému upravující jízdní vlastnosti podvozku prostřednictvím korekce ovládacích prvků. V závěru práce pojednává o realizaci fyzického modelu a zmiňuje negativní zkušenosti s použitými vývojovými prostředky.

Klíčová slova

- PWM
- CPLD
- servo
- A/D převodník
- budič

Abstract

The aim of this bachelor thesis is to design and construct proportionally remote-controlled model of car, including design and implementation of all electronic systems and controls and realistic testing. Key condition for the realization digital calculations is to use only gate-logic of CPLD (Complex Programmable Logic Device) not a microprocessor. Chapter "Ovladač" describes the solution of control and the transformation of the analogue variable rotation of potentiometer on the digital form comprehensible to the CPLD including the complete design of integration. In the section "Servo", the work describes the principle and options of CPLD controlled, most available servomechanism. The chapter also covered the question and solution of communication between CPLD and servomechanism including and graphical representation. Chapter "Motor" is about the powertrain of model and its integration. Describes the compilation of gate-logic, which induce by pulse width modulating of output signal DC motor to spin. Design of DC motor exciter is contained to. The last chapter "Řídící vazba" develops and then states in real possibilities of active electronic system governing the handling chassis trough correction of control elements. In "Závěr" the thesis treat of realization and construction of real physical model and mention negative experiences with used developmental devices.

Key words

- PWM
- CPLD
- servo
- A/D convertor
- exciter

Obsah

Titulní strana.....	1
Originální zadání práce.....	2
Prohlášení.....	3
Abstrakt	4
1. Úvod	8
2. Ovladač.....	9
2.1 A/D.....	9
2.2 Citlivost.....	9
2.3 Zapojení.....	9
2.4 Realizace ovladače.....	10
3. Servo.....	12
3.1 Výběr.....	12
3.2 Impulz.....	12
3.3 Generování Impulzu.....	13
3.4 Funkčnost a zapojení.....	15
4. Motor.....	16
4.1 Ovladač motorku.....	16
4.2 PWM.....	17
4.3 Budič.....	18
4.4 Typ motorku.....	19
4.5 Zapojení.....	19
5. Řídicí vazba.....	20
5.1 Teorie.....	20
5.2 Verze 1.....	21
5.3 Verze 2.....	21
5.4 Zapojení vazeb.....	22
6. Závěr.....	23

Seznam použité literatury.....	25
Příloha A – Schéma zapojení a návrh layoutu DPS ovladače.....	26
Příloha B – Blokové schéma řízení serva.....	27
Příloha C – Blokové schéma řízení PWM.....	28
Příloha D – Schéma zapojení a návrh layoutu DPS budiče.....	29
Příloha E – Blokové schéma vazba4evo (Verze 1).....	30
Příloha F – Blokové schéma vazba2 (Verze 2).....	31
Příloha G – Blokové schéma řízení modelu autíčka s vazbou Verze 1.....	32
Příloha CD s elektronickou podobou práce.....	CD

1. Úvod

Jak již napovídá název tématu práce, na následujících stranách bude popsána jedna z možností, jak realizovat proporcionální ovládání akcelerace a řízení směru pohybu podvozku modelového vozidla. V oblasti modelářského hobby lze čerpat mnoho inspirací jak tohoto dosáhnout, od upoutaných, plně mechanických modelů, přes čistě analogové soupravy, až po dnešní plně číslicové přenosy a servomechanizmy.

Mým cílem je dosáhnout proporcionálního ovládání, jehož vstupem bude pohyb potenciometry, na výstupu interpretován změnou úhlu natočení kol servem řízené nápravy a regulace otáček motoru realizované změnou šířky pulzů pulzně šířkové modulace (PWM). Zadání též ukládá využití obvodu CPLD, konkrétně vývojový přípravek CoolRunnerII Starter Kit a třeshičkou na dortu je inteligentní vazba úhlu natočení kol na rychlost pohybujícího se modelu pro zlepšení ovladatelnosti a omezení přetáčivosti podvozku, toliko charakteristické pro vozidla se zadním náhonem.

K řešení tedy vyvstává hned několik problematik. Jak převést změnu odporu, realizovanou potenciometrem, do číslicové, pro programovatelné hradlové pole srozumitelné podoby. Volba, pochopení a ovládnutí proporcionálního servomechanismu, který bude mechanicky měnit úhel natočení kol. Konstrukce proud-zesilujícího členu pulzně šířkového modulování (budiče) elektromotorku pohánějící autíčko. Dále, přinutit programovatelné hradlové pole, aby číslicové podobě změny odporu doopravdy správně rozumělo, převedlo ji na PWM motorku a řídicí impuls serva, algoritmem přepočítávalo „dávku plynu“ a pro zlepšení jízdních vlastností rozšířilo poloměr otáčení autíčka. Nakonec nesmím zapomenout na zkonstruování samotného experimentálního modelu.

Závěrem úvodu stojí za zmínku, že jako modelářský a automobilistický nadšenec neskrývám sympatie k těmto úkolům. Informace, znalosti a schopnosti potřebné ke splnění cílů a náplně tématu zadání tudíž čerpám nejen z literatury a internetu, ale z vlastní kutilské zkušenosti a především z výuky na TUL, neb jako absolvent gymnázia jsem středoškolské vzdělání technického charakteru neabsolvoval.

Sledem zásad pro vypracování, podrobnější analýza, návrhy řešení a řešení samotné následuje v dalších kapitolách.

2. Ovladač

2.1 A/D

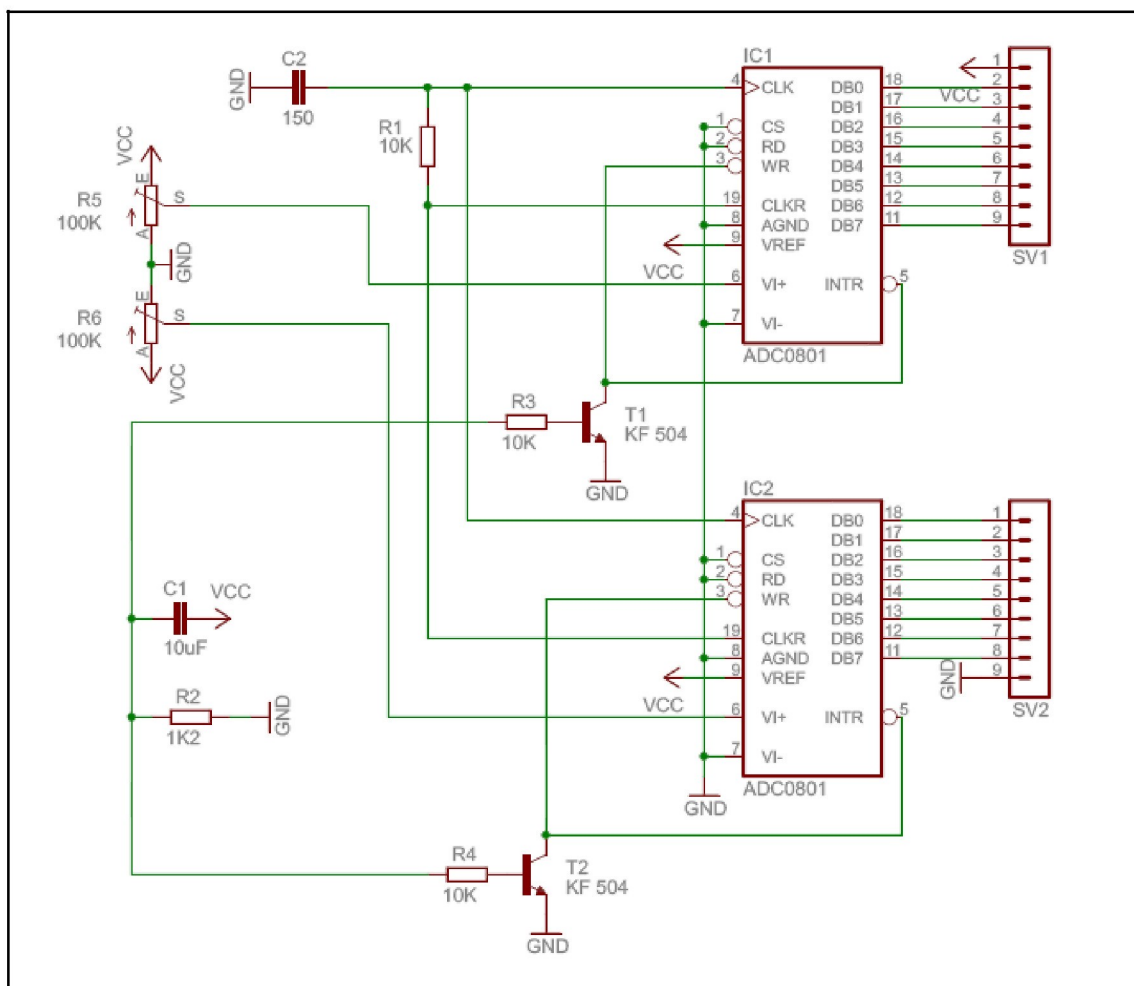
Je hned několik cest jak převádět, v našem případě pootočení potenciometrem, na číslo číslicových vstupů CPLD. Změnu odporu ale nejlépe indikovat analogově, sledováním změny napětí regulovaného potenciometrem. Tuto změnu však číslicový obvod zpracovat neumí. Logika věci tedy velí užití analogově-digitálního převodníku a těch máme na výběr ze dvou druhů. Můj los padl na převodníky paralelní, neb se pro můj účel zdály vhodné z hlediska snadno sledovatelné a následně dobře odladitelné číslicové komunikace oproti A/D převodníkům sériovým.

2.2 Citlivost

Již při řešení vstupů je nutné mít na paměti citlivost, respektive počet kroků proporcionality řízení. Z pozorování a zkušenosti s modelářskými ovladači je mi známo, že na úhel pootočení modelářského serva o 90° , je proporcionalita 34 kroků dostačující, z hlediska citlivosti serva pak 127 poloh na rozsah úhlu 90° dokonce největší realizovatelná. Výběr konkrétního paralelního A/D převodníku tedy padl na, mě v danou chvíli nejdostupnější, osmibitový reverzní ADC0801. Díky jeho reverzitě převádí, v mnou navrženém zapojení, napětí regulované potenciometrem v rozsahu pouze sedmi bitů tj. v rozsahu čísel 0 až 127.

2.3 Zapojení

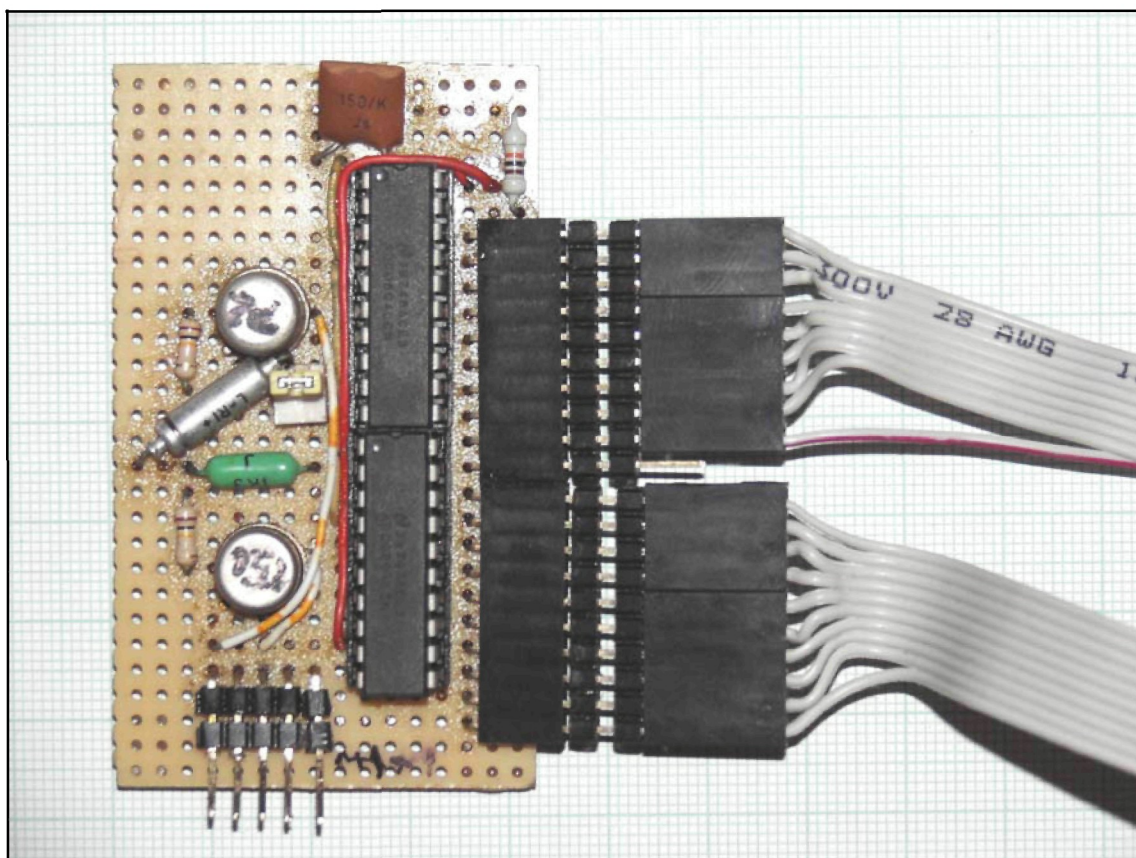
Po prozkoumání datasheetu [4], možností zapojení převodníků řady ADC080x a několika pokusech a omylech bylo vyvinuto zapojení (obr 2.1) komparující kladné napětí, regulované potenciometry, s nulou, záporný reverzní vstup je uzemněn, rozsah výstupu převodníku 0 až 255 se tím omezil na polovinu. Takt čítání převodníku je udáván RC obvodem složeným z kondenzátoru C2 a rezistoru R1. Napájení je pak realizováno vývojový přípravkem CoolRunnerII. Schéma zapojení na obr. 2.1 je navrženo zároveň pro dva potenciometry R5 a R6 ovládající zatačení a akceleraci.



Obr. 2.1 : schéma zapojení A/D převodníků

2.4 Realizace ovladače

Samotný průběh vstupu ovladače CPLD je tedy realizován dvěma potenciometry, jejichž otáčení je převáděno na dva osmi bitové údaje (Obr.2.1 ADC0801, ve vzestupném pořadí významnosti bitů DB0 až DB7), které jsou dále každý zvlášť zpracovávány vývojovým přípravkem a převáděny na PWM a řídicí impuls serva, jehož generování je popsáno v následující kapitole. Návrh možného layoutu plošného spoje s převodníky a potenciometry integrovanými na DPS (deska plošných spojů) a čitelné schéma je vyobrazeno v příloze A. Laboratorní pokusný plošný spoj a vzdálený ovládací modul jsou pak fotodokumentovány na obrázcích 2.2 a 2.3. Modul se sestává pouze z potenciometrů a je připojen k DPS pětivodičovým cca 3 metry dlouhým kabelem.



Obr. 2.2 : experimentální DPS s převodníky

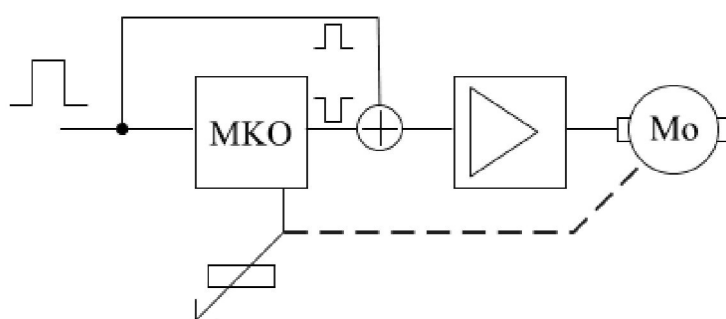


Obr. 2.3 : vzdálený ovládací modul s potenciometry

3. Servo

3.1 Výběr

Zatáčení modelu má být realizováno prostřednictvím servomotorku. Vývoj či výroba není předmětem této práce. Tedy, k pohonu zatáčení je zvolen a užit ozkoušený a masově vyráběný typ servopohonu se standardizovaným impulzem. Tato serva jsou díky masové produkci pro modelářské účely snadno dostupná, a to nejen po finanční stránce. Významná přednost spočívá v nepřeberném množství velikostí, tvarů, silových charakteristik v kterých jsou k dostání. Dle zdroje [5] abych přesně citoval „dnešní modelářská serva obsahují elektromotor s převodovkou a řídicí elektroniku. Zjednodušené zapojení elektroniky je na blokovém schematu (Obr. 3.1). Do vstupu přichází řídicí impuls, který spustí monostabilní klopný obvod, ten vygeneruje impuls o délce odpovídající momentální poloze serva a opačné polarity než je vstupní řídicí impuls. Tyto dva impulzy se porovnají a výsledkem je rozdílový impuls, který po zesílení přes můstkový spínač způsobí roztočení elektromotoru jedním nebo druhým směrem. Elektromotor přes převodovku otáčí výstupní hřídeli a současně i potenciometrem, který působí jako zpětná vazba do monostabilního klopného obvodu. Směr otáčení je takový, že impuls generovaný monostabilním klopným obvodem se

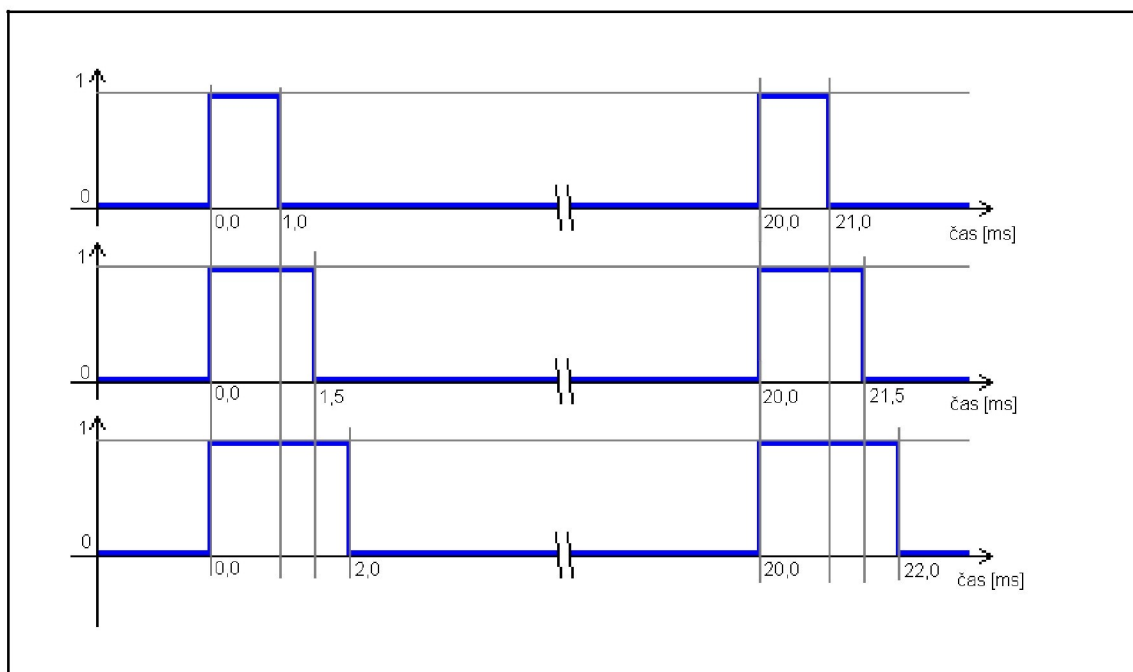


svojí délkou přibližuje délce vstupního řídicího impulsu a až jsou oba impulzy stejně dlouhé, elektromotor se zastaví. Servo dosáhlo polohu, která odpovídá momentálně přijímanému řídicímu impulsu.“

Obr. 3.1 : Zjednodušené zapojení elektroniky serva

3.2 Impulz

Vstupem serva jsou 3 vodiče, impulzní, napájecí (VCC) a zemnicí (GND). Napájet je lze napětím od 3 do 6 V. Řízeno je pak pulzy dlouhými 1 až 2 milisekundy s 50 Hz frekvencí a to v rozsahu otáčení osy serva 90°. I přes standardizaci některá serva reagují v až dvojnásobném rozsahu 0,5 ms až 2,5 ms. Šířka pulzu přímo úměrně odpovídá otočení osy, taková serva dosahují otáčení osy až o 180°. Středová poloha serva je dána 50 Hz pulzem o délce 1,5 ms, viz Obr. 3.2.



Obr. 3.2 : Řídicí signál v polohách 1, 1,5 a 2 ms

3.3 Generování impulsu

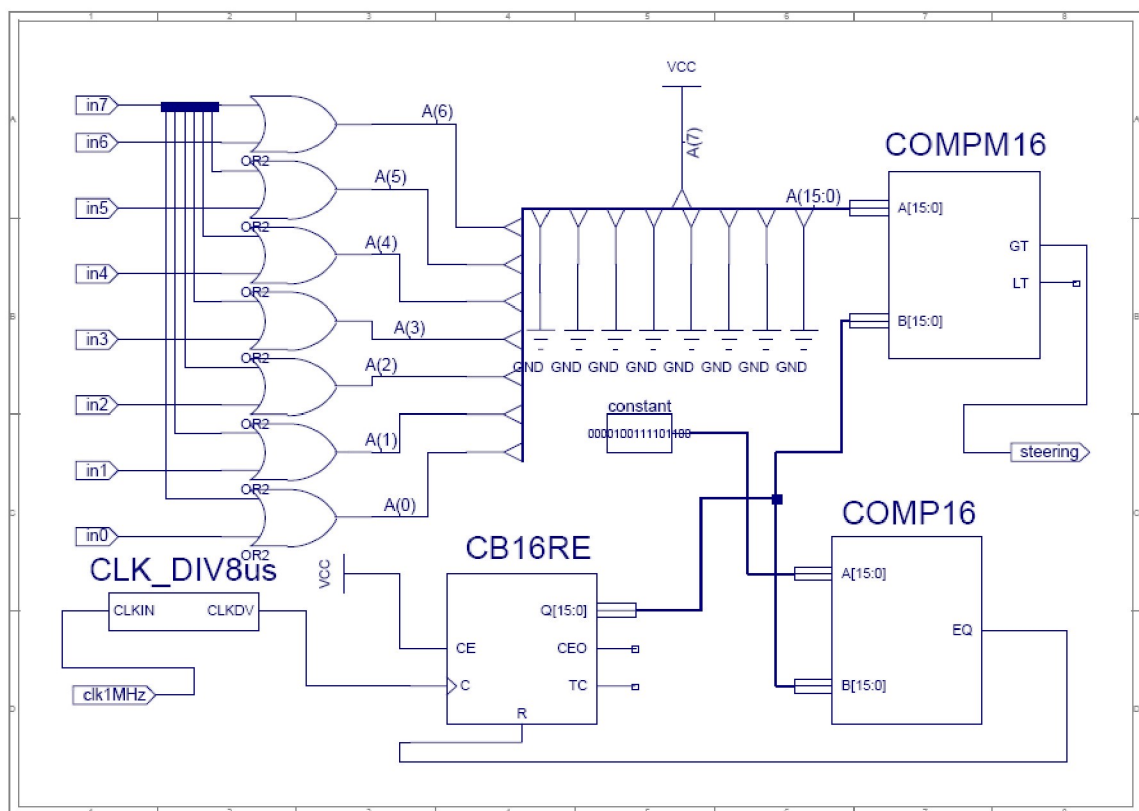
V tomto bodě se dostáváme k otázce, jak programovatelným hradlovým polem generovat řídicí impuls a plynule měnit jeho délku číslem vydaným jedním z převodníků. Vývojový přípravek CoolRunnerII je programován ve vývojovém prostředí Xilinx ISE 10.1 a to prostřednictvím blokových schémat za použití implicitních bloků. Poměrně hodně času spolykal v důsledku neúspěšný vývoj sčítání podělených frekvenčních pulzů integrovaného oscilátoru přípravku. Sedm děliček, sedm časových délek v poměru k sedmi hodnotám bitů. Bohužel, po zkrocení prostředí ISE syntéza ukázala, že CoolRunnerII má pro tuto verzi generování nedostatek paměti. Vývoj tedy pokračoval dále. Nestandardní blok děličky byl použit pouze jednou a to v rámci generování nejmenšího kroku impulsu, generuje pulz po 8mi kmitech 1MHz oscilátoru.

Zdrojový kód děličky:

```

1 `timescale 100 ps / 10 ps
2 module CLK_DIV8us (CLKDV,CLKIN);
3 parameter integer DIVIDE_BY = 8;
4 input CLKIN;
5 output CLKDV;
6 reg CLKDV;
7 integer CLOCK_DIVIDER;
8 initial begin
9   CLOCK_DIVIDER = 0;
10  CLKDV = 1'b0;
11 end
12 always @(posedge CLKIN)
13 begin
14   CLOCK_DIVIDER= CLOCK_DIVIDER+1;
15   if (CLOCK_DIVIDER ==
16       (DIVIDE_BY/2 + 1))
17   begin
18     CLOCK_DIVIDER = 1;
19     CLKDV = ~CLKDV;
20   end
21 endmodule

```



Obr. 3.3 : Blokové schéma komunikace hradlového pole se servem

Na obr. 3.3 je znázorněno konečné komunikační propojení vstupní informace interpretované vstupy in0 až in7, hradel a výstupu ovládání impulzu serva (výstup „steering“). Tato poslední, dle pozorování bez chyb fungující vývojová verze, vyhodnocuje výsledky asi takto. Oscilátor CoolRunnerII dodá impuls s frekvencí 1 MHz, tomu dělička prodlouží periodu z 1us na pulz s periodou 8 us, dále je zpracováván 16 bitový čítačem CB16RE, který počítá tyto impulzy až do čísla komparovaného v COMP16, to je dáno blokem „constant“, kde je binárně v 16 bitech zapsána hodnota 2540. Po dosažení tohoto čísla vyhodnotí komparátor rovnost a resetuje čítač. Vynásobením 0,008 ms a konstantou 2540 dostaneme požadovanou nosnou periodu 20 ms neboli frekvenci 50Hz. Tato část udává kdy klopit výstup na hadinu jedna. Číslo generované čítačem je dále porovnáváno v komparátoru COMP16 s 16 bitovým číslem jehož prvních 7 bitů je (A(0) až A(7)) je dáno vstupní informací vstupů in0 až in7. Osmý bit trvale nastaven na hodnotu 1. Ostatní mají pevnou hodnotu 0. COMP16 je komparátor porovnávací a vyhodnocuje nerovnost vstupu A a B. Dokud je vstup A větší než vstup B, je na výstupu GT hodnota jedna, přesáhne-li však hodnota B, v našem případě čítání do 2540, hodnotu tj. číslo v rozsahu 128 až 255, překloupí se výstup

do nulové hladiny. Výsledkem tak vznikne sedmi bity říditelný 50 Hz impulz o šířce 1 až 2 ms neb minimální hodnota pro sklopení do nulového stavu je dána 8. bitem A(7), po vynásobení hodnotí bitu 127 a 0,008 ms vyjde požadovaná krajní hodnota 1 ms. Nejdelšího impulzu 2 ms je pak dosaženo naplněním nižších bitů A(0) až A(6) jedničkami, pootočením potenciometru do krajní polohy.

Ve schématu je dále řešen drobný nedostatek převodníku, a to že v krajní poloze využívá nám neúčinný 8. bit a generuje hodnotu 10000000b, tu sérií hradel OR2 převádím na hodnotu předcházející a to 01111111b. Obr.3.3 je pro lepší čitelnost ve větším rozlišení též zařazen jako příloha B.

3.4 Funkčnost a zapojení

Funkčnost tohoto schématu byla testována a ověřena na vývojovém kitu hned s několika servy, od čínské neznámkové produkce až po značkové HITEC HS-303. Ve všech případech úspěšně s maximální citlivostí kroku.

Dle dokumentace Xboardu [6] byl sled pinů zapojen následovně, 8 výstupních pinů ovladače serva DB7 až DB0 bylo připojeno k vývojovému přípravku na pinech v pořadí P5, P4, P3, P2, P143, P142, P140 a P139, ty pak jsou programem přiděleny vstupům schématu v pořadí od in7 až po in0. Výstup serva „steering” je programem zvolen na pinu P79, servo i ovladač jsou oba napájeny zdrojem přípravku z pinů VCC a GND.

4. Motor

4.1 Ovladač motorku

Zpět ke kapitole 2. Ovladač. Dle zásad pro vypracování má být rychlost modelu plynule nastavitelná posuvným potenciometrem, což bohužel v případě mého modelu dodrženo není a to z důvodu následujícího. Po opakovaném nákupu různých modelů posuvných potenciometrů bylo užití posuvného typu zavrženo. Žádný mě dostupný potenciometr nedokázal opakovaně vykazat použitelnou plynulost natož linearitu změny odporu která by byla úměrná pohybu jezdce. A/D převodník pak generoval čísla více či méně náhodná, což působilo nemalé potíže při vývoji a odlaďování programové části této práce. Rychlost pohybu je tedy též řízena otočným potenciometrem se stejným zapojením jako řízení směru, jen s tím rozdílem, že v rámci uživatelsky příjemného pojetí bylo ovládání rozšířeno o jakýsi “kohoutek” pistolového ovladače se zubovým převodem a vratnou pružinou jak je patrné na obr. 4.1.

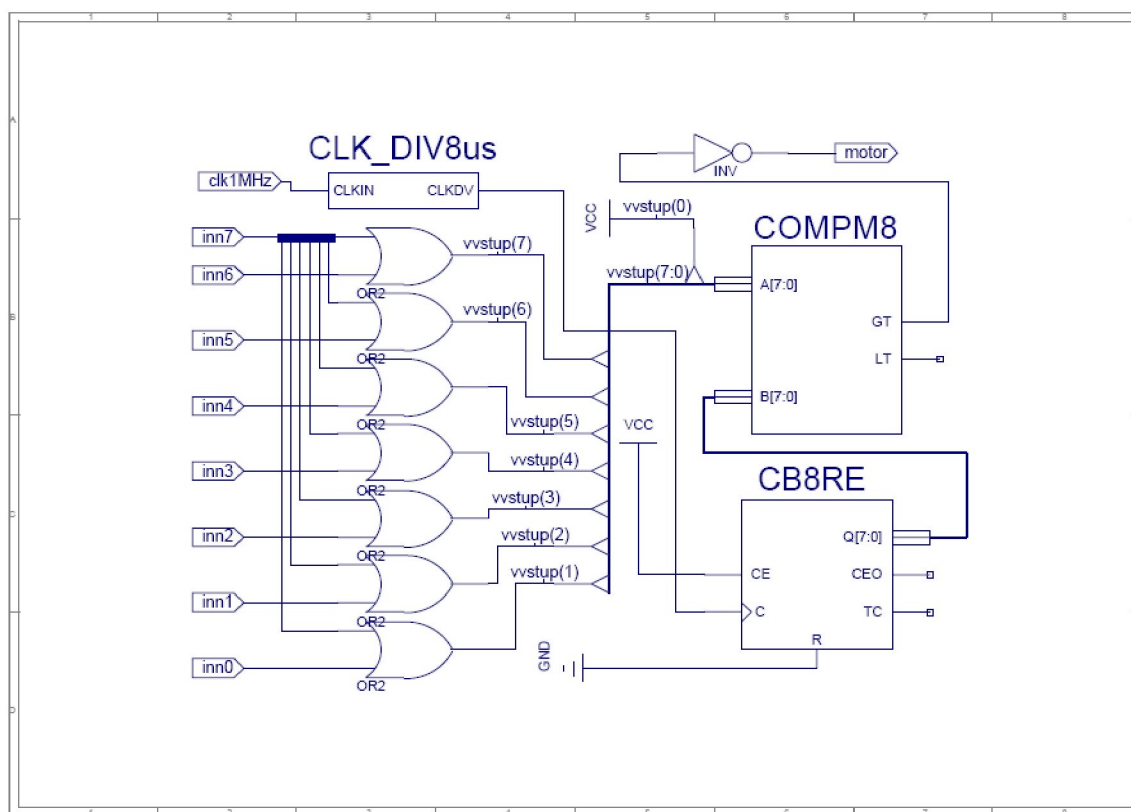
Citlivost, zapojení převodníku a celková realizace je totožná jako u zatáčení, vstup přípravku je též realizován osmi respektive sedmi bity.



Obr. 4.1: Vzdálený řídicí modul – pistolový ovladač

4.2 PWM

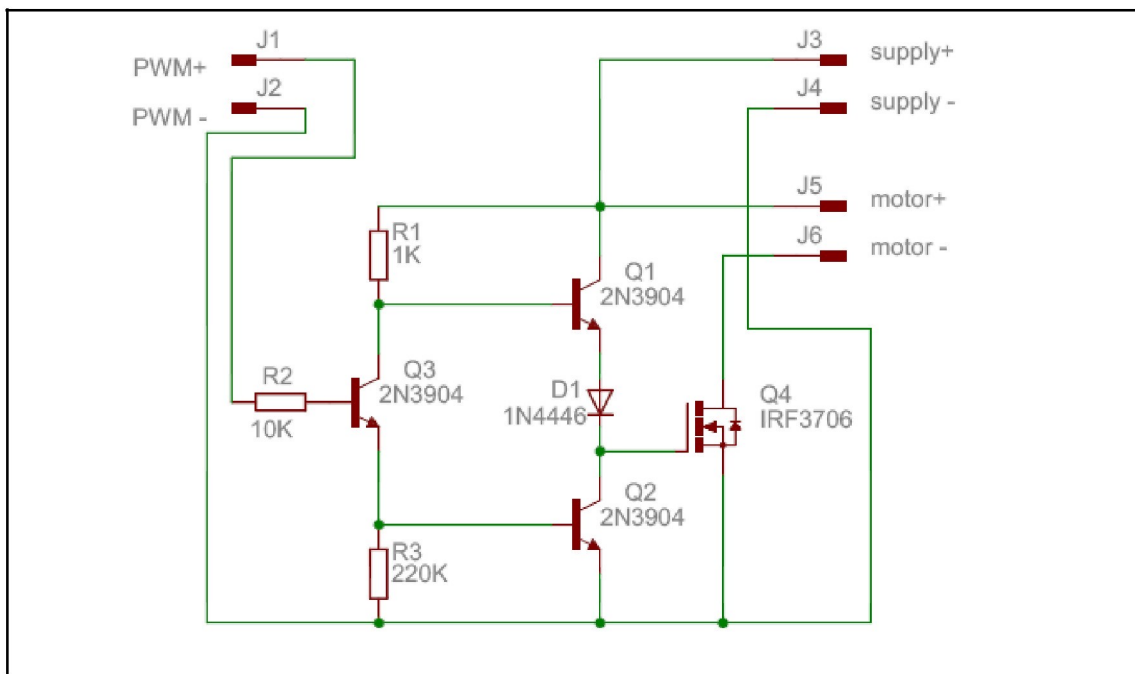
Pulzně šířková modulace, neboli PWM (Pulse Width Modulation). Jedná se o obdobu střídavy, která je řešena již v kapitole ovládání serva. Leč u PWM se mění šířka pulsu, a ne jen v rozsahu druhé desetiny šířky nosné frekvence. Generování PWM je tedy v zásadě o něco jednodušší a méně náročnější na výpočetní prostředky. Pro úsporu hardwarových prostředků použijeme tu samou děličku 8 us jako u zatáčení a ta nám zároveň poskytne délku poloviny kroku o který se bude šířka pulzů s periodou cca 2 ms zkracovat. Generování PWM probíhá dle schématu následovně. Osmibitový čítač CB8RE periodicky napočítá 255 pulsů o délce 8 us, celkově tedy periodu 2 ms a s tou klopí výstup na logickou hladinu 1 (po invertaci, kvůli použitému zapojení budiče, 0). Číslo udávané výstupem čítače je porovnáváno komparátorem COMPM8 s osmibitovou vstupní ovládací informací, je-li menší udržuje výstup logickou hladinu 1 (inv 0). Přeroste-li však porovnávané číslo, výstup se překlopí na hladinu 0 (po invertaci 1). První bit řídicího čísla je nastaven pevně na hodnotu 1. Zbýlých sedm je určováno úhlem otočení potenciometru ovladače. Perioda 2 ms je tedy rozdělena na 127 kroků dlouhých 0,016 us. Opět, stejně jako u zatáčení je osmý vstupní bit inn7 sérií hradel OR2 využít k převodu z krajního údaje 10000000b na 01111111b.



Obr.4.2 : Blokové schéma PWM řízené 8 bitovým vstupním číslem

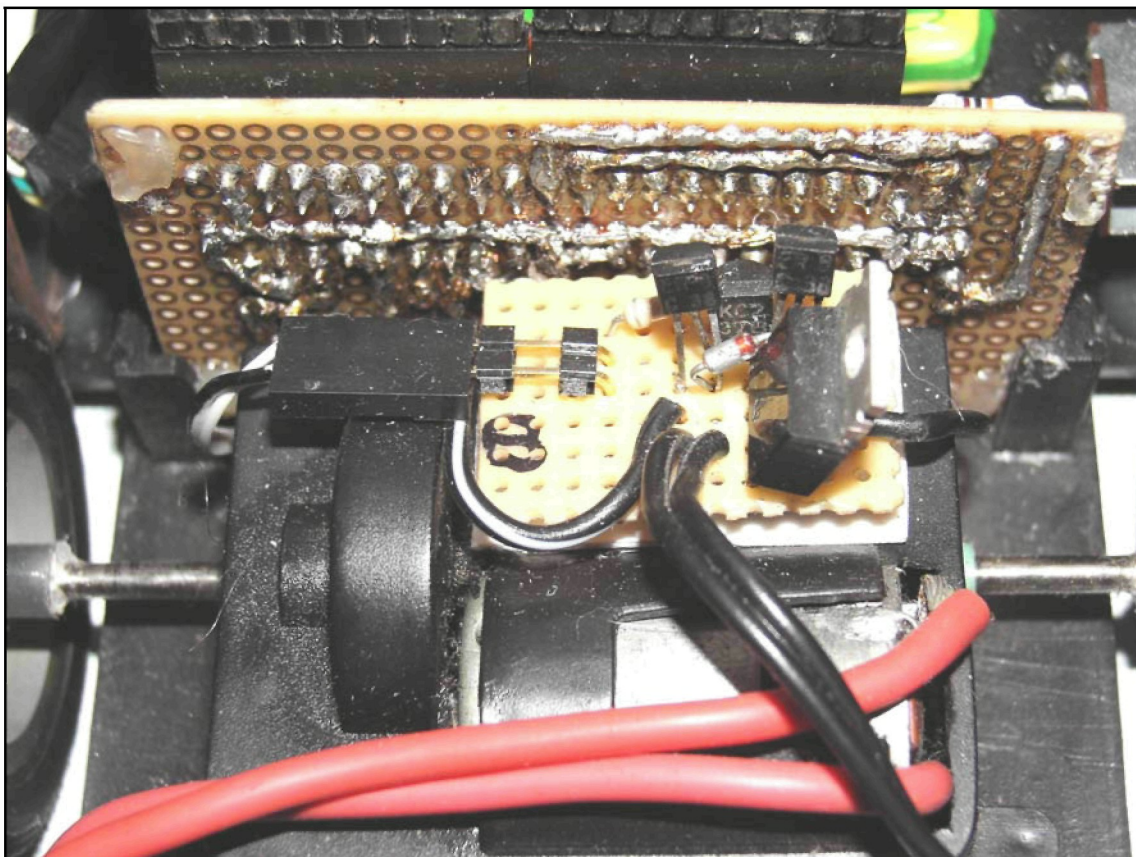
4.3 Budič

Protože samotný PWM výstup s 3,3 V a maximálním proudem 0,5 A, které je vývojový přípravek schopen dodat ani zdaleka nedostačuje pro napájení pohonného elektromotoru, je potřeba vyvinout pro obě napájecí veličiny zesilovací člen. Dále ho budu nazývat budič. Teorie nabízí několik možností. Spínací relé, bohužel pro PWM příliš pomalé a ztrátové. Výkonné DC-DC měniče si též s PWM neporadí. Zaměřil jsem se tedy na návrh budiče využívajícího k zesílení PWM tranzistoru. Cesta se ukázala schůdná, leč trnitá. Tranzistory sice bez zadýchání zvládají kmitočty PWM a dnešní snadno dostupné výkonné N-mosfety (například v budiči využitý IRF3706 [7]) bez problému zesilují napětí 12 V při proudech v řádu desítek A, při spínacím napětí od 4,5 V. Bohužel tyto součástky jsou citlivé na experimentování ne příliš znalých a zkušených uživatelů. Počtu prototypů moc nechybělo a málem by byl nabyt dvojmístného čísla. Poslední odzkoušená fungující verze pak se pak vyznačuje zapojením vyobrazeným na obr. 4.3 a reálnou formou obr. 4.4.



Obr. 4.3 : Schéma budiče

Piny PWM (+) a (-) označují vstup PWM generované CPLD, pozor tento budič je navržen jako invertní. To znamená, že nepřerušovaný logická signál na hladině 1 dává motoru povel aby se netočil a jeho přerušování teprve začne motor roztáčet. Piny “supply (+) a (-)” jsou určeny pro napájení 6 až 9,6 V. Piny motor (+) a (-) pak slouží k připojení stejnosměrného elektromotoru. Schéma a návrh layoutu DPS v příloze D.



Obr. 4.4 : Laboratorní vývojová verze budiče instalovaná na podvozku

4.4 Typ motorku

K pohonu autíčka slouží stejnosměrný elektromotor řady 300 6V s doporučeným napětím 7,2 - 9,6 V, při napájení 9,6 V dosahuje cca 17200 otáček za minutu. Modelu pro napájení tedy postačí 6-8 článků s ohledem na zprůvodování pak je doporučeno spíše 6 maximálně 7.

4.5 Zapojení

Dle [6], 8 výstupních pinů ovladače motoru DB7 až DB0 bylo připojeno k vývojovému přípravku na pinech v pořadí P43, P42, P41, P40, P39, P35, P34 a P33, ty pak jsou programem přiděleny vstupům schématu v pořadí od inn7 až po inn0. Výstup serva “steering” je programem zvolen na pinu P96, PWM ovladač je napájen zdrojem přípravku z pinů VCC a GND, zem impulsu PWM je zprostředkována pinem GND.

5. Řídicí vazba

5.1 Teorie

Jak chápat 4. zásadu pro vypracování “Vytvořte a zrealizujte elektronický systém, který je schopen plynule natáčet kola modelu autíčka podle toho, do jaké pozice je natočen model volantu v závislosti na rychlosti pohybujícího se modelu”. Logika věci a moje dosavadní modelářská a automobilistická praxe nabízí probádání možností aktivního vylepšení jízdních vlastností modelu. Oním fatálním nedostatkem je v našem případě tzv. přetáčivost podvozků se zadním náhonem a těžištěm vozidla umístěným v přední polovině šasi. K přetáčivému chování dochází tehdy, když má zadní náprava při změně směru jízdy vozidla přebytek výkonu a nedostatek boční trakce. Zadní náprava má potenciál “ujíždět bokem” a “přebíhat” nápravu přední, jinak řečeno dochází ke smyku. Možností jak toto chování omezit, redukovat nebo mu dokonce úplně zamezit se dnes v praxi používá mnoho. Nepřeberné množství aktivních systémů řídicích světlou výšku vozidel, aktivní řízení brzd na jednotlivých kolech, systémy omezující výkon motoru/dávku plynu při ztrátě trakce a systémy to vše kombinující, aby podvozek seděl na zemi “jako přibitý”, nejevil známky ať už nedotáčivosti, či přetáčivosti a vedl stopu zvolenou řidičem. Vycházím tedy z toho, že u částečně mnou a částečně čínským dělníkem sestrojeného modelu autíčka mohou ovlivnit pouze 2 řídicí prvky, a to bez zpětné vazby na to, jestli se model skutečně pohybuje rychlostí přikázanou ovladačem, nebo zde působí setrvačnost modelu či nedostatek trakce. Po analýze všech možností se nabízejí 3 řešení, jak dosáhnout vazbou mezi plynem (řízení otáček motoru) a zatačením, alespoň omezení přetáčivosti modelu. První možnost, naplňující zadání, je plynulé snížení výchylky řízení směru kol s rostoucím množstvím plynu. Druhá možnost, o poznání sofistikovanější, už z důvodu, že varianta jedna nás prostě jen nenechá zatočit a pošle autíčko rovněji jen aby se nedostalo do smyku, je plynulé omezení plynu v závislosti na aktuální výchylce zatačení. V praxi by to vypadalo asi tak, že ať přidáte plyn sebevíc, pokud budete mít kola zatočena v maximálním úhlu, nenabídne vám PWM více než polovinu možné šířky impulzu s tím, že vám bude plynule omezení redukovat v závislosti na zmenšování úhlu výchylky kol. Třetí verze je pak kombinace obojího a vyhodnocování priority zatačení a plynu dle aktuálního stavu obou veličin. Z těchto tří možností se povedlo realizovat dvě, první pro naplnění zásad pro vypracování a druhá ze zvědavosti co to bude dělat ☺.

5.2 Verze 1

Jak již bylo nastíněno v předchozí podkapitole, tato vazba, kde je množstvím plynu omezeno zatáčení vozidla, není z hlediska ovládání úplně ideální, sic plní zadání. Analogie takového řízení nás nutí nepřidávat plyn proto, aby jsme mohli zatočit.

Kostrbaté, ale v důsledku funkční.

A nyní jak na to. Máme k dispozici dva vstupy. Dvě čísla od 0 do 127 interpretující množství plynu a polohu, potažmo výchylku serva. Plyn je řízen lineárně postupným “plněním” PWM sto dvaceti sedmi kroky. Zatáčení pak mění lineárně výchylku se středovou polohou udanou číslem rovným polovině kroků tj. 64. Mnou navržený systém binárních výpočtů, jehož obsáhlé a neskladné blokové schéma je obsaženo v příloze E (pracovní název vazby “vazba4evo”), postupuje následovně.

Od osmibitové informace plynu (vstupy pp) odečtu odčítačkou ADSU8 číslo 64, je li výsledek záporný, přeteče a výstup nemění, je li výsledek kladný, pošle ho dál. Binárním posunutím je číslo poděleno dvěma a dále je odečteno od hodnoty 32 (tj. čtvrtina kroků). Prozatím tento výsledek nazvu C. Výchylka byla spočtena dle toho, jestli je číslo udávající otočení serva větší než 96 nebo menší 32, odečtením čísla 96 nebo odečtením od čísla 32. Za podmínky, že je poloha serva (vstupy sp) vně intervalu $\langle 32; 96 \rangle$ a v případě porovnání výchylky a čísla C je výchylka větší, podle “směru zatočení” pod 64 nebo nad 64 je od 64 hodnota $(32+C)$ buď odečtena nebo přičtena a bitovou výhybkou dána na výstup (výstupy spo). V konečném pozorování pak růst dávky plynu nad polovinu rozsahu snižuje plynule rozsah výchylek na každou stranu až o 25%.

Při napájení pěti až sedmi články model s takto implementovaným algoritmem skutečně vykazoval na hladkém prášném povrchu omezení, při nižším napájecím napětí dokonce naprostou eliminaci přetáčivosti.

5.3 Verze 2

Aneb verze pro kutily a zvědavce. Algoritmus omezující plyn v závislosti na úhlu natočení kol se zdá z hlediska příznivějších operací s čísli jednodušší a snáze odvoditelnější. Ano je to tak. Jeho vývoj a odladění ale stál nedokonalostí vývojového prostředí ISE 10.1 mnohem více času než zapeklité omezení výchylky se závislostí na plynu. Vazba navržená a odladěná pro tuto efektnější teorii počítá jedničky a nuly následovně.

Nejdříve, podobně jako předchozí systém, porovná hodnotu polohy serva (vstupy sp) se středovou hodnotou 63 (blok constant 00111111b) a to prostřednictvím osmibitového komparátoru COMPM8. Tento údaj pak poslouží k přepnutí bitové výhybky mezi dvěma odčítačkami ADSU8, jedna odečítá od vstupní hodnoty číslo 63, druhá odečítá od 63 vstupní hodnotu ovládání serva. Tím dostaneme hodnotu odchylky zatačení od přímého směru (mezi 0 a 64). Tu opět odečteme odčítačkou od 63 čímž dostaneme doplněk odchylky, která nám v součtu s 63 dá nejvyšší možnou omezenou hodnotu čísla ovládající plyn. Komparátorem COMPM8 tedy program porovná spočtenou algoritmickou hodnotu s hodnotou přiváděnou na vstup pohybem ovladače (vstupy pp), bude-li hodnota vstupu větší než číslo vypočtené, program jí přepnutím bitové výhybky na výstupu (výstupy ppo) vypočtenou hodnotou nahradí. Dále jsou čísla zpracovávány totožně na PWM a impuls serva, jako tomu bylo v předchozích kapitolách. Takto číselně řešený algoritmus pak omezuje maximum plynu v rozsahu od hodnoty 64 do hodnoty 127 o velikost aktuální výchylky zatočení. Rozsáhlé blokové schéma této vazby je zaznamenáno v příloze F (pracovní název “vazba2”).

5.4 Zapojení vazeb

O zapojení vazby, vstupů, výstupů, generování PWM a řídicího impulsu serva a vůbec celé komunikaci CPLD vypovídá poslední příloha G. Schéma v tomto případě využívá vazbu verze 1 (ve schématu označena pracovním názvem jako “vazba4evo”) která je zastoupena pouze blokovým symbolem a stejnojmennými vstupními a výstupními piny. Další příloha zapojení vazby verze 2 není nutná, neb se schematicky jedná pouze o zaměnění různojmenných bloků stejného zobrazení. Zapojení vstupních a výstupních pinů desky je pak převzato z bodů 3.4 a 4.5.

6. Závěr

Co dosud ve zprávě nebylo zmíněno. Vývoj / konstrukce / výběr / modifikace samotného podvozku. Probíhal v intervalu několika dní a to především metodou pokus - omyl, selekcí domácího inventáře torz autíček a podvozků, sestrojených, ať už vlastní konstrukcí, nebo modifikací podvozků hraček převážně čínské produkce. Ve výběrovém řízení nakonec zvítězil podvozek čínské produkce o rozměrech cca 26 x 12 cm, šířkou pokrývající šířku desky („Xboardu”) vývojového přípravku. Podvozek má zadní náhon, o pohon se stará původní převodovka, s mnou, za původní “slabý” motor zaměněným motorem řady 300. Model je dále vybaven modelářským servem popsáním v kapitole 3. Šasi je upraveno a modifikováno pro umístění DPS vývojového přípravku, DPS ovladače a DPS budiče. Výsledný vzhled viz obr.6.1.



Obr. 6.1 : Experimentální model autíčka s ovladačem

K napájení motoru je v případě obr. 6.1 použit osmi článkový nikl-metal hydridový modelářský akumulátor o kapacitě 1100 mAh, samotný Xboard je pak napájen nezávisle jedním lithium-iontovým článkem o napětí 3,7 V.

Cíle vymezené v úvodu této práce byly dosaženy a definované problémy vzali za své. Práce pomohla rozvinout schopnosti a v praxi plnohodnotně vyzkoušet do této doby pouze teoretické znalosti nabyté při výuce na TUL. Kombinací návrhů hardwaru a softwaru bylo dosaženo nejen teoretických, ale i v reálném čase pozorovatelných a hmatatelných výsledků, především díky implementaci navržených elektronických systému na reálný model autíčka. Bylo úspěšně vyvinuto a odzkoušeno jak proporcionální číslicové ovládání generující ovládací impulzy serva a motoru, tak aktivní algoritmická vazba mezi oběma ovládacími kanály. K vývoji bylo použito několikero informačních především elektronických zdrojů.

Programování probíhalo v programu Project Navigator ISE 10.1, jenž na jednu stranu svými vlastnostmi a možnostmi vytvářet program při výuce osvojenou cestou a to prostřednictvím tvorby blokového schématu práci ulehčoval, na druhou stranu chybovostí převodu blokových schémat do HDL (Hardware Description Language) a následné špatné syntézy programu přidělal mnoho hodin práce s hledáním neexistujících chyb v zapojení. I přes to se povedlo ovládání programu zkrotit a byly nalezeny způsoby jak chybovost obejít, případně nedat prostředí ISE prostor k produkci překladových a syntézních chyb.

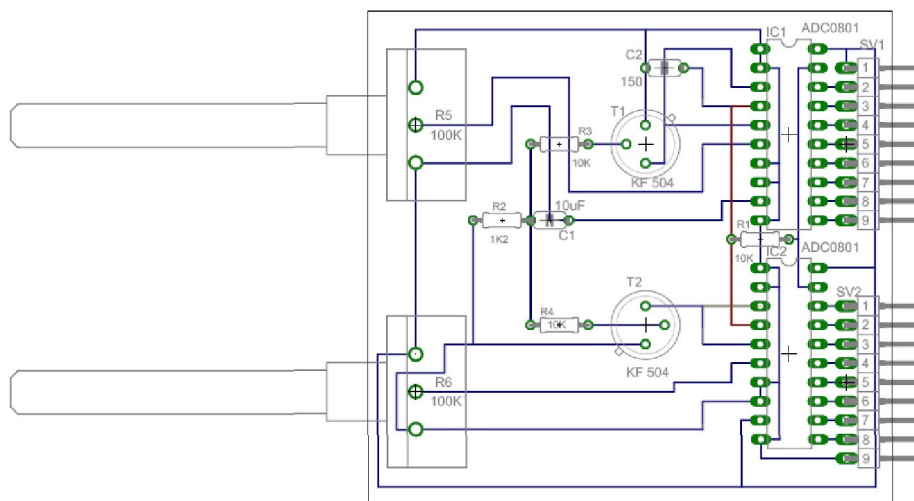
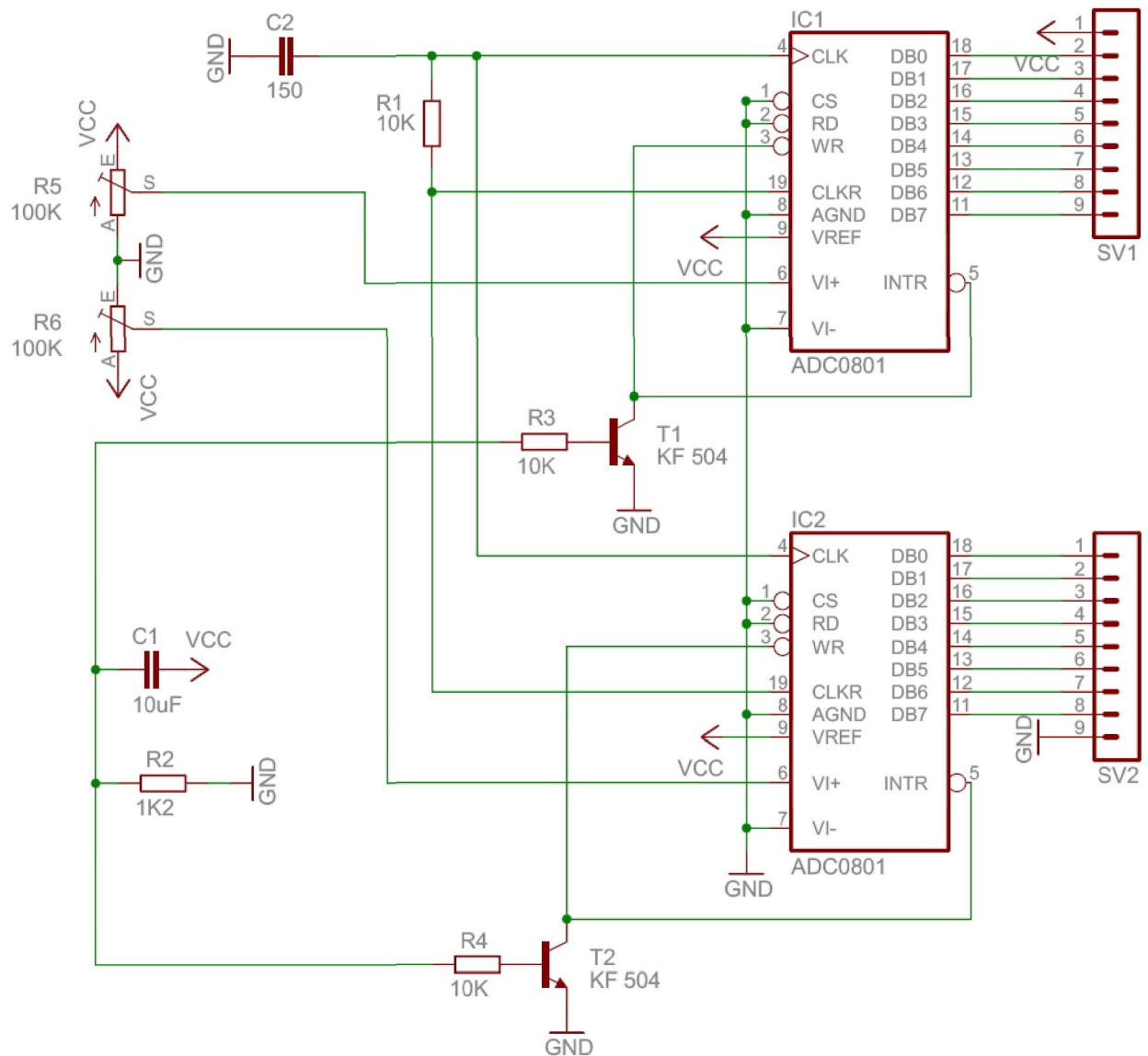
Další lehce záporná zkušenost získaná při tvorbě tohoto projektu byla lehká nespolehlivost desek CoolRunnerII Xboard. Tyto dnes již nevyráběné vývojové přípravy jsou naprosto jedinečné pro svoji minimální spotřebu, integrovaný zdroj schopný převést napájení v rozsahu 3,3 až 9 V na napájení hradlové 3,3 voltové logiky, integrovaný oscilátor s volitelnou frekvencí, procesor Picoblaze a hlavně obrovské množstvím pinů dle potřeby nastavitelných jako vstupy či výstupy. Během práce jsem vystřídal defakto 3 desky, u první selhal obvod řídící napájení, druhá vykazovala nechuť připojit se přes USB rozhraní a až třetí prozatím funguje spolehlivě a bez zaváhání. Toliko negativní zkušenosti a doporučení na co si dávat pozor.

Jak jsem již před pár odstavci zmínil, hlavních bodů práce, tj. návrh hardwaru a softwaru a jejich praktická realizace a odzkoušení, se i přes zmíněné nesnáze podařilo úspěšně dosáhnout. Závěrem bych tedy už jen podotkl, že díky této práci jsem pronikl do světa možností tvorby realizace a aplikace číslicových elektronických systémů nejen pro modelové využití. Rád bych pracoval na dalších podobných projektech návrhu a realizace.

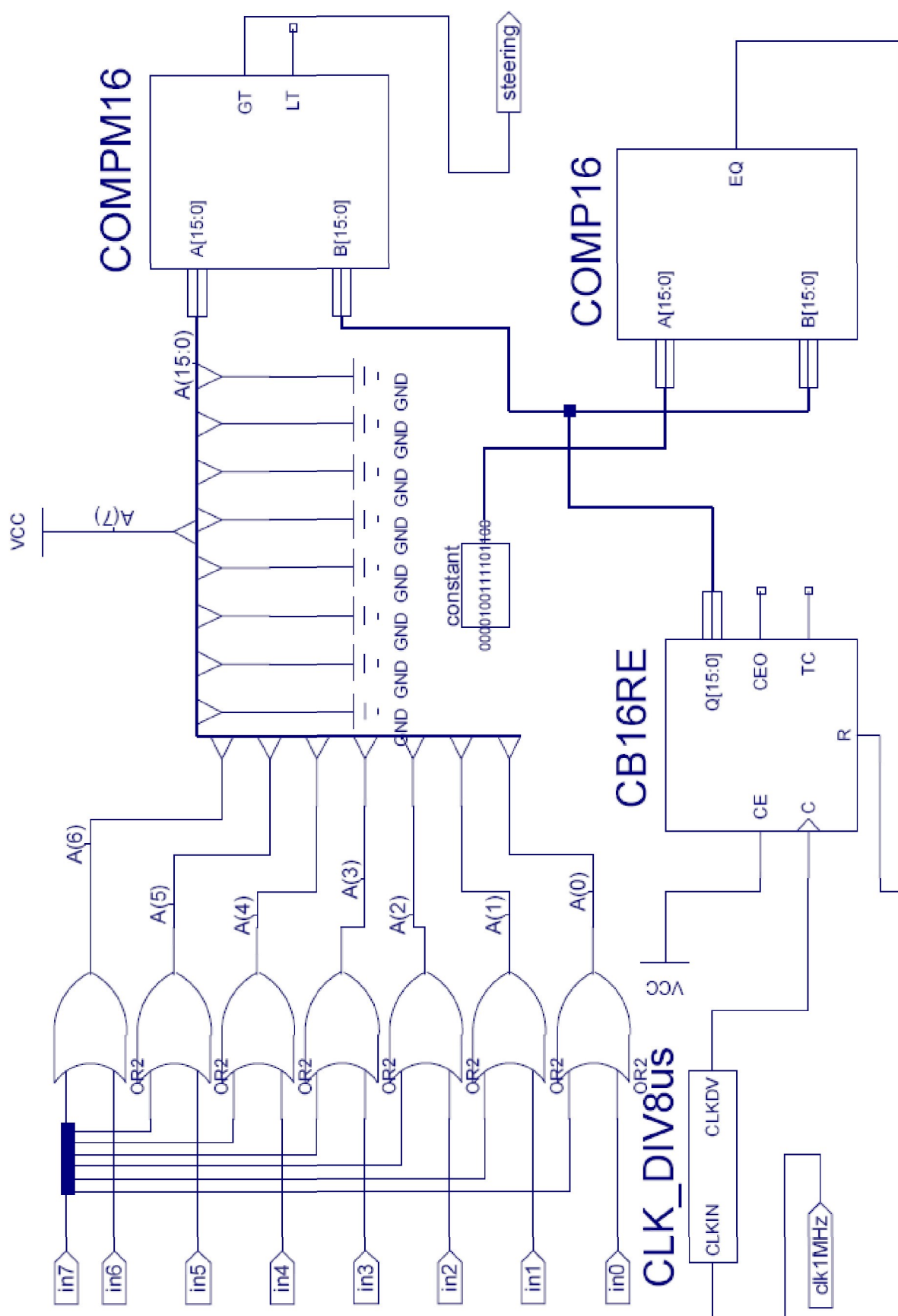
Seznam použité literatury

- [1] Pinker Jiří, Poupa Martin: Číslicové systémy a jazyk VHDL, BEN 2006
ISBN 80-7300-198-5
- [2] www.xilinx.com
- [3] www.hw.cz
- [4] www.gme.cz/_dokumentace/dokumenty/320/320-011/dsh.320-011.1.pdf
- [5] www.vlastikd.webz.cz/bastl/serva.htm
- [6] www.xilinx.com/support/documentation/data_sheets/ds094.pdf
- [7] www.gme.cz/_dokumentace/dokumenty/213/213-276/dsh.213-276.1.pdf
- [8] www.robotika.cz/guide/servo/en
- [9] www.cs.wikipedia.org

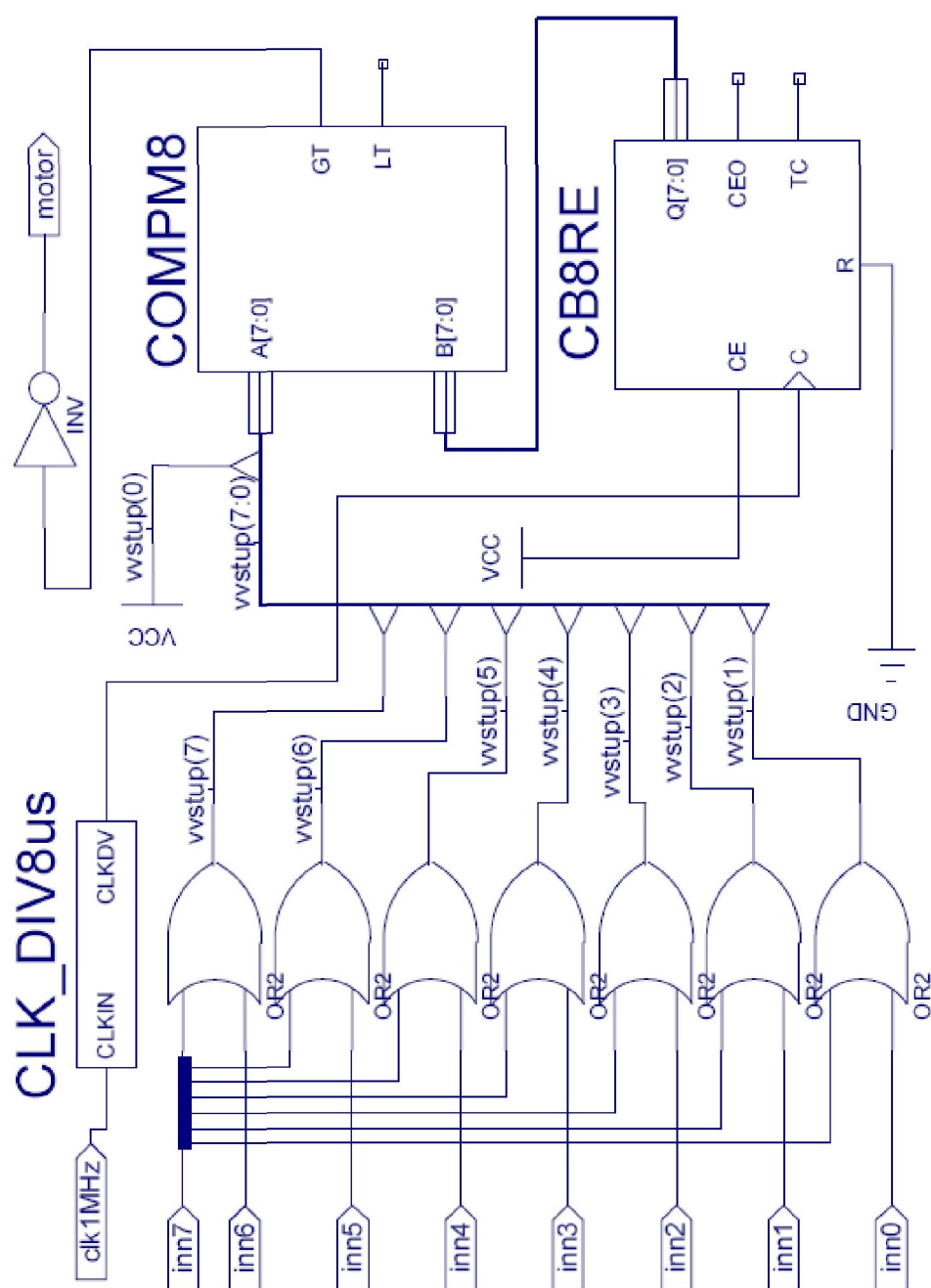
Příloha A – Schéma zapojení a návrh layoutu DPS ovladače



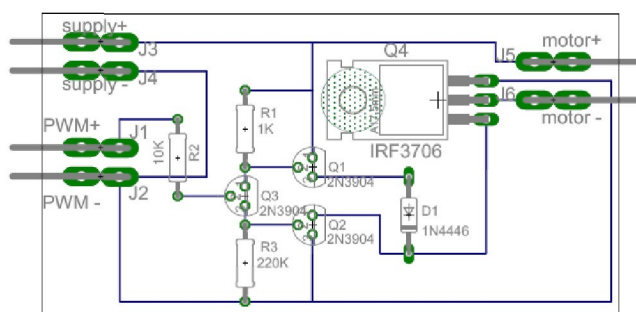
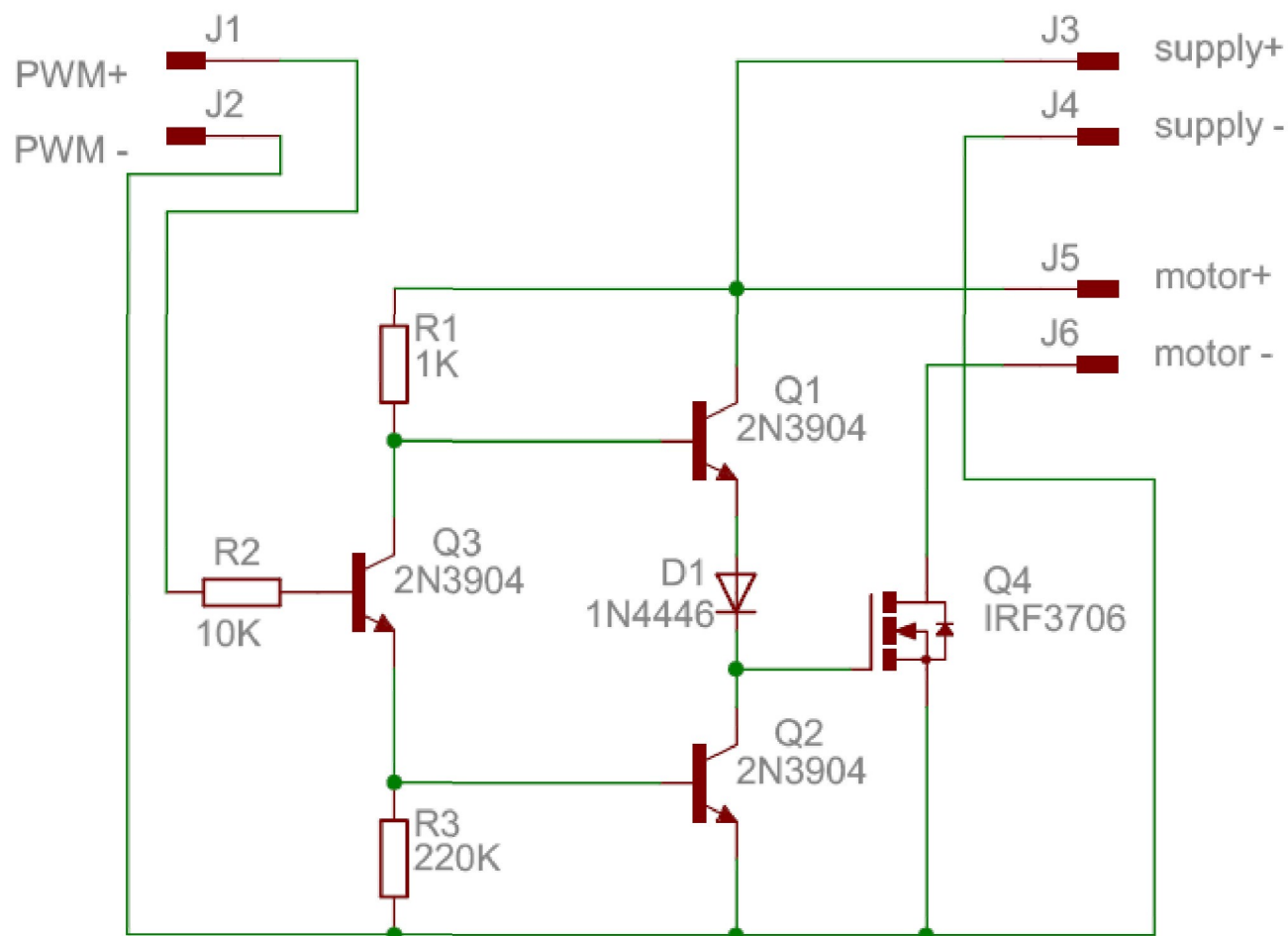
Příloha B – Blokové schéma řízení serva



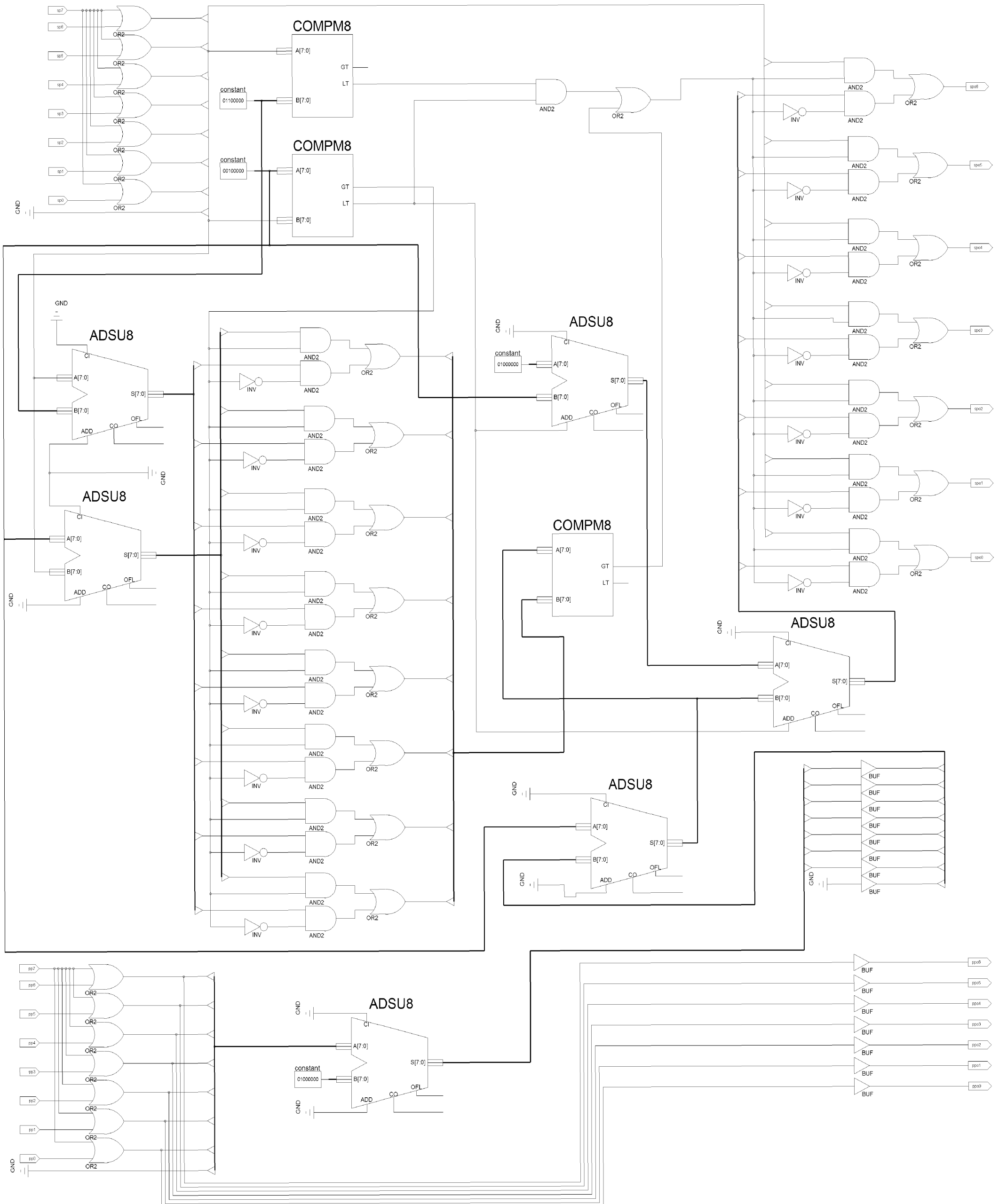
Příloha C – Blokové schéma řízení PWM



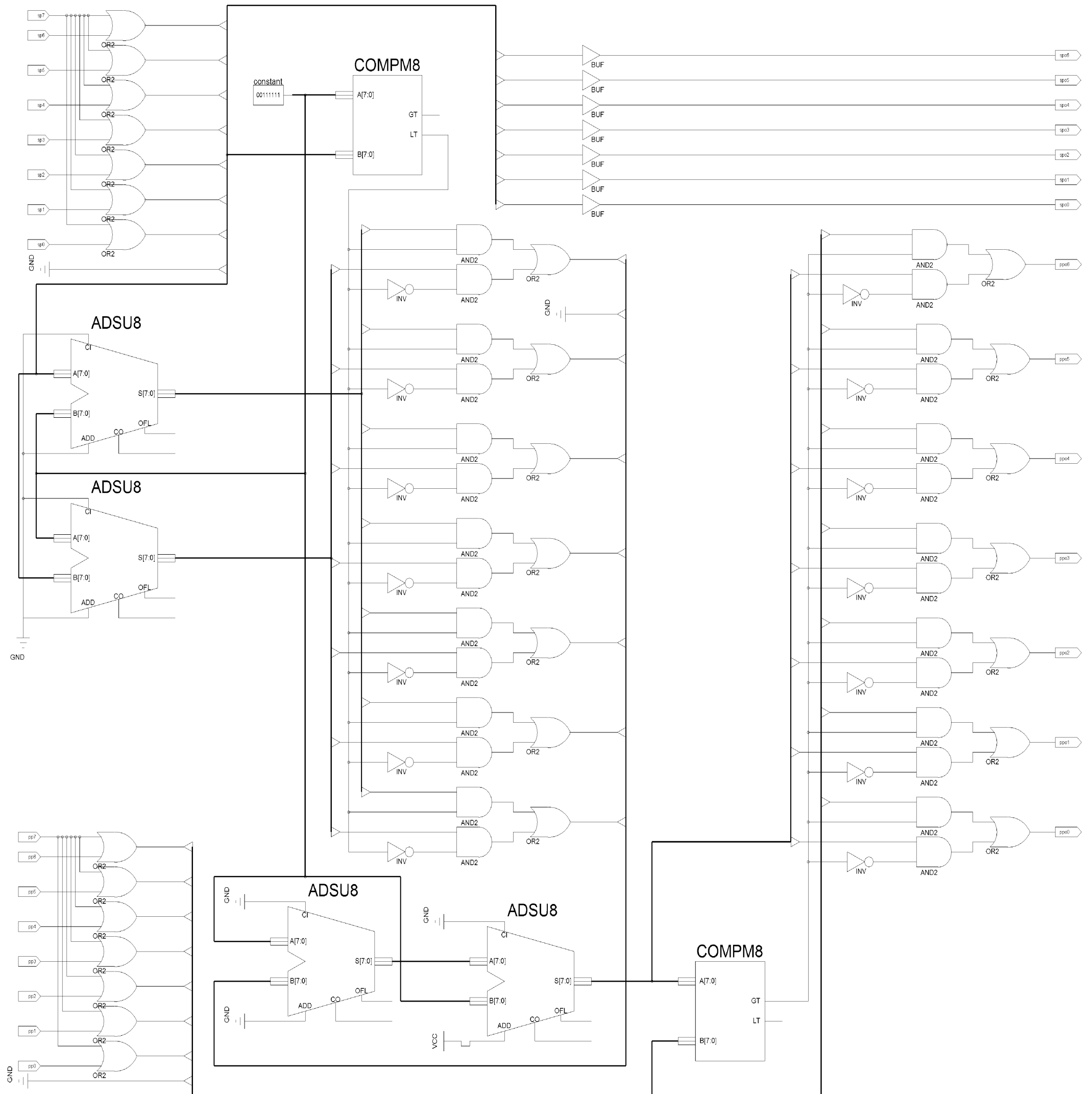
Příloha D – Schéma zapojení a návrh layoutu DPS budiče



Příloha E - Blokové schéma vazba4evo (Verze 1)



Příloha F - Blokové schéma vazba2 (Verze 2)



Příloha G - Blokové schéma proporcionálního řízení modelu autíčka s vazbou Verze 1

