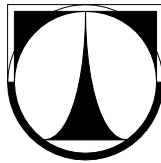


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2010

Jan Tůma

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 1802R022 – Informatika a logistika

**Inteligentní ovladač zpětných zrcátek
v automobilu**

**Smart device for the adjustment
of car side view mirrors**

Bakalářská práce

Autor: **Jan Tůma**
Vedoucí práce: Ing. Zbyněk Mader, Ph.D.

V Liberci 21. 5. 2010

Originál zadání

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 požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

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

21. 5. 2010

Jan Tůma

Abstrakt

V rámci bakalářské práce je navrženo a realizováno technické řešení pro elektronické ovládání zpětných zrcátek automobilu s možností uložení a opětovného nastavení zvolené polohy. Pro návrh bylo použito vývojové desky CoolRunnerII, osazené obvodem CPLD s programovatelnými hradlovými poli. Návrh a vlastní naprogramování funkčních schémat bylo realizováno v prostředí Xilinx ISE. V práci jsou podrobně popsány jednotlivé stavy systému včetně ošetření stavu dorazů při natáčení zrcátka vzhledem ke konstrukčnímu omezení jeho pohybu. K vlastní realizaci je pak použit servomechanismus pro ovládání zrcátka a externí paměť FLASH pro uložení a opětovné vyvolání požadovaného nastavení zrcátka. Ovládání servomechanismu je řešeno s využitím pulsně šířkové modulace, komunikaci s externí pamětí je realizována pomocí sériového rozhraní. Realizace navrženého řešení byla odzkoušena pro uložení dvou nezávislých poloh nastavení zpětného zrcátka.

Klíčová slova: Hradlové pole, sériová komunikace, pulsně šířková modulace, paměť FLASH, zpětné zrcátko automobilu.

Abstract

The problem addressed in this Bachelor's thesis is the design and realization of a device for electronic adjustment of car side view mirrors. An important feature of this device is the ability to store and restore individual settings of mirror position. CoolRunnerII CPLD Development Board was used for the design. The design and programming of the functional diagrams were done using XILINX ISE environment. The thesis gives a detailed description of the system states including the handling of mechanical limits on mirror position. The actuator used to change the mirror position is a PWM controlled servomechanism. Positions of the mirror are stored in an external FLASH memory. The communication with this memory is done through a serial interface. The device that was developed within the framework of this thesis was tested for storing two independent positions of the side view mirror.

Keywords: Complex programmable logic devices, serial communication, pulse width modulation, flash memory, side view mirror.

Obsah

Prohlášení	3
Abstrakt	4
Abstract	4
Obsah.....	5
Seznam obrázků	6
Seznam použitých zkratk.....	7
Úvod.....	8
1. Zdokonalení vlastností automobilu - tuning	9
1.1. Princip tuningu	9
1.2. Ovládání zpětných zrcátek s pamětí	10
2. Technické zajištění	12
2.1. Vývojová deska CoolRunnerII	12
2.2. Připojení servomechanismů	13
2.3. Připojení paměti FLASH.....	15
3. Návrh funkčního schéma v prostředí Xilinx ISE	21
3.1. Vstupní moduly pro nastavení polohy	22
3.2. Ovládání servomechanismů	28
3.3. Komunikace s FLASH pamětí - zápis a čtení	30
4. Realizace modulu pro ovládání zrcátek.....	36
Závěr	40
Literatura	41

Seznam obrázků

Obr. 1. Vývojová deska CoolRunnerII	12
Obr. 2. Obvod CPLD XC2C256	13
Obr. 3. Průběh PWM pro řízení úhlu natočení servomechanismu	14
Obr. 4. Vnitřní uspořádání paměti	15
Obr. 5. Zapojení master a slave obvodů	16
Obr. 6. Vztah mezi hodinovým signálem a přenášenými daty	17
Obr. 7. Časový sled signálů pro MODE3	18
Obr. 8 Princip přenosu dat pro zařízení master a slave	19
Obr. 9. Časový sled signálů pro zápis do bufferu paměti	19
Obr. 10. Časový sled signálů pro zápis z bufferu do hlavní paměti	20
Obr. 11. Časový sled signálů pro čtení z paměti	20
Obr. 12. Prostředí pro tvorbu projektů v ISE WebPack	21
Obr. 13. Blokové schéma čtyř základních modulů	22
Obr. 14. Blokové schéma modulu DORAZ	23
Obr. 15. Blokové schéma modulu OVLADANI	24
Obr. 16. Blokové schéma automatu READ_SAVE	26
Obr. 17. Blokové schéma automatu LOAD_DEF	27
Obr. 18. Blokové schéma ovládání servomechanismů	29
Obr. 19. Blokové schéma pro komunikaci s FLASH pamětí	31
Obr. 20. Blokové schéma pro automat BUFFER_WRITE	32
Obr. 21. Blokové schéma pro automat MEMORY_WRITE	34
Obr. 22. Blokové schéma pro automat MEMORY_READ	35
Obr. 23. Schéma úpravy napájecího napětí	37
Obr. 24. Schéma pro připojení servomechanismů	37
Obr. 25. Schéma připojení FLASH paměti	38
Obr. 26. Celkový pohled na modul inteligentního ovládání zpětného zrcátka	39

Seznam použitých zkratk

CLK	clock signal (hodinový signál)
CPLD	complex programmable logic device (komplexní programovatelné hradlové pole)
CS	chip select
MISO	master in slave out
MOSI	master out slave in
MSB	most significant bit (bit s nejvyšší hodnotou)
PWM	pulse width modulation (pulsně šířková modulace)
SPI	serial peripheral interface (sériové periferní rozhraní)
USB	universal serial bus (univerzální sériová sběrnice)
WP	write protect (ochrana proti zápisu)
log0	logická hodnota 0
log1	logická hodnota 1

Úvod

Snad všechna odvětví průmyslové výroby jsou dnes silně ovlivněna použitím moderních technologií (informačních), miniaturizací v technickém provedení (elektronické prvky), či novými, lehkými a pevnými materiály. Drobný příspěvek k tomuto způsobu řešení se snaží nabídnout i předložená bakalářská práce.

Problematika ovládání zpětných zrcátek automobilu s pamětí nastavení polohy, kterou se bakalářské práce zabývá, není sice nová, ale v běžné sériové výrobě není zatím obvyklá. Hlavním přínosem je uložení okamžitého nastavení polohy zrcátka do paměti a možnost pozdějšího opětovného nastavení do této polohy. Tato výhoda se ocení zejména v případech, kdy se v automobilu střídá více stále stejných řidičů (služební vozy, rodinné auto, ale také to může být i případ různé zátěže automobilu - jako např. se již dnes hodně používá přestavení světlometů, aby neoslňovaly protijedoucí vozidlo). Přitom správné a hlavně stálé nastavení polohy zpětných zrcátek je velmi důležité pro pohodlnou jízdu řidiče - aby se nemusel často nahýbat, případně klonit hlavu.

Pro vypracování bakalářské práce byly zvoleny poměrně běžně dostupné prvky, a to jak pro vlastní technickou realizaci (cenově dostupné), tak i z pohledu návrhu a realizace programového vybavení (byly použity prostředky, s kterými jsem se seznámil v rámci bakalářského studia).

Celá úloha pak byla řešena na konkrétní realizaci ovládání zrcátek pro automobil Renault 21 z roku 1991, což bylo na jednu stranu dobré pro ověření funkčnosti navržených řešení a úprav, na druhou stranu bylo nutné se držet stávajících prostorových a dalších technických dispozic (např. napájení).

Předložené řešení si nekladlo za cíl nabídnout automobilovým výrobcům alternativu pro sériovou výrobu, ale jistě by bylo možné výsledky práce úspěšně aplikovat pro tzv. auto-tuning, kdy se sériově vyrobený automobil individuálně upraví jak po vzhledové, tak i po technické stránce (což bylo nakonec i mojí motivací při výběru vhodného zadání bakalářské práce).

1. Zdokonalení vlastností automobilu - tuning

Význam slova „Tuning“ by se dal z angličtiny přeložit jako ladění či vyladování. V oblasti úprav na automobilech je to především snaha o atraktivnější vzhled a dynamičtější jízdní vlastnosti pro sportovní účely. Vzhledem k tomu, že jde o individuální záležitost, je konečný výsledek vždy závislý na představě a nakonec i finančních prostředcích majitele.

1.1. Princip tuningu

Vlastní „tuning“ se provádí na nejrůznějších prvcích automobilu. Tak např. na karoserii se jedná většinou o přidání různých dílů, většinou plastových (úpravy nárazníků, prahů, přidání přitlačného křídla aj.), či pomocí laminování může dojít až k celkové přestavbě karoserie. Tady záleží zcela na fantazii autorů. Nutno ale poznamenat, že některé z úpravy podléhají zejména z hlediska bezpečnosti schválení pro silniční provoz. Další úpravy se s oblibou věnují jednodílným či vícedílným ráfkům z lehkých slitin,



kdy se sice nejedná většinou o domácí výrobu, ale pouze výměnu za jiný typ ráfků, které jsou na trhu k dispozici v bohatém sortimentu. Řešit se může pak povrchová úprava,



většinou je to lesklý stříbrný lak nebo jiná kvalitní barva, která je odolná proti odření. Samostatnou kapitolou jsou pak úpravy na výfuku, kde jde nejen o elegantní vzhled, ale tyto úpravy také mohou přímo ovlivnit výkon motoru. Správně upravený výfuk může dát větší výkon motoru až o 15kW, což je už opravdu znát. Pokud se jedná o úpravy podvozku, je nutné, aby vůz stále dobře držel na silnici - a to zejména při rychlejších sportovních jízdách. To

většinou zajistí sportovní verze tlumičů a pružin, které také snižují celkovou výšku vozu a přitom jsou mnohem tvrdší, než standardní sériově dodávané tlumiče, což se projeví zejména menším nepříjemným nakláněním při ostrých průjezdech zatáčkami. Také se rozšiřují například rozteče kol použitím různě širokých podložek.

Samostatnou kapitolou je zvyšování výkonu motoru. Existuje mnoho nejrozumnějších způsobů, jako je ostřejší vačka, úprava sání, použití sportovního vzduchového filtru, atd. až k dokonalejším modifikacím samotné řídicí jednotky. Tady je však vhodné svěřit práci opravdovým odborníkům, protože uvedené úpravy mají zcela jistě vliv na spolehlivost a tedy i výdrž motoru.



Obrázky byly převzaty z veřejně dostupných webových stránek s tématem auto-tuningu.

1.2. Ovládání zpětných zrcátek s pamětí

Ovládání zpětných zrcátek s možností přímého nastavení žádané polohy není sice běžná úprava na automobilu, a to i z pohledu výše popsaného auto-tuningu, nicméně pro některé majitele aut to může být nejen zajímavá, ale i praktická věc. Jak jsem uvedl již v úvodu, pokud se např. v rodině střídá u jednoho automobilu více řidičů, bude jistě pohodlnější, než-li pokaždé nastavovat polohu zkusmo, využít uložených nastavení někde v paměti automobilu a pouhým stiskem tlačítka nastavit zpětná zrcátka do požadované (předem nastavené) polohy. Toto nastavení se zadá tak, že si každý řidič nastaví polohu zrcátek, jak mu nejlépe vyhovuje, a opět jen např. jedním stiskem patřičného ovladače dá pokyn k zapamatování polohy. Tento údaj je pak připraven k opětovnému použití. V podstatě se jedná o stejný princip, jako když si nastavíme u rádia právě naladěnou stanici do předem připravených předvoleb, což už dnes asi všichni považují za samozřejmé.

Důležitá je i vhodná volba ovládacích prvků v interiéru automobilu. Klasické mechanické ovládání, resp. mechanický převod pohybu ovládacího kolíku na natočení zrcátka, realizovaný většinou systémem pružin, je dnes již asi opravdu minulostí. Elektronické ovládání sice vyžaduje mnohem více prvků v systému, ale zase je mnohem citlivější. Klasický "prostorový pohyb" mechanického ovládání je možné nahradit joystickem, kdy přímo určíme směr změny nastavení polohy zrcátka, nebo dvěma dvousměrnými ovladači pro natáčení ve vertikální a horizontální ose zrcátka (častá je realizace kolébkového ovladače do čtyř protilehlých směrů). Takové nastavení je sice trochu pracnější, ale konstrukčně jednodušší.

Systém celého ovládání je možné rozdělit do dvou částí, a to na samotné ovládání natáčení zrcátka a na ukládání a načítání jeho okamžité polohy. K ovládání natáčení zrcátka je nejjednodušší princip čtyř spínačů. Každý z těchto spínačů slouží pro natočení zrcátka do jednoho ze čtyř směrů - tj. nahoru, dolů, doleva a doprava. Z pohledu přirozeného pohodlí řidiče je pak důležitá realizace vlastního pohybu zrcátka tak, aby řidič viděl změnu ve směru, který ovládá (a ne opačně), což představuje v podstatě zobrazení toho, co chce řidič za sebou vidět. Takže například když chceme zobrazit to, co se nachází za námi vpravo, stiskneme spínač pro natočení zrcátka doprava. Natáčení v horizontální a svislé ose jsou na sobě nezávislá, tudíž je možné zrcátko natáčet i v úhlopříčném směru. Toho se docílí stisknutím dvou tlačítek zároveň a to vždy jednoho tlačítka pro horizontální pohyb a jednoho pro vertikální pohyb.

Jelikož pohyb zrcátka je konstrukčně omezen, je zřejmé, že musí existovat nějaké omezení pro rozsah jeho natáčení. Tento rozsah může být dán buď mechanickou konstrukcí samotného mechanismu natáčení, nebo parametry servomechanismů starajících se o natočení zrcátka. V obou případech je ovšem potřeba stanovit krajní polohy. Hodnota krajní polohy se může buďto vypočítat, podle použitého servomechanismu, nebo se musí provést zkouška, ve které se servomechanismus nechá otočit za obě krajní polohy. V momentě, kdy servomechanismus překročí krajní polohu, je třeba ho vrátit minimálně o jeden krok zpátky a tuto polohu označit za krajní. Překročení krajní polohy se pozná podle náhlého skoku při otáčení servomechanismu, kdy se servomechanismus dostane mimo povolený rozsah a skočí na mechanickou zarážku. V této poloze se servomechanismus snaží stále otáčet, ovšem mechanická zarážka mu to nedovolí a tak při častém a delším setrvávání v této poloze může dojít k mechanickému poškození.

Pro ilustraci je uvedena fotografie zrcátka použitého při řešení bakalářské práce.

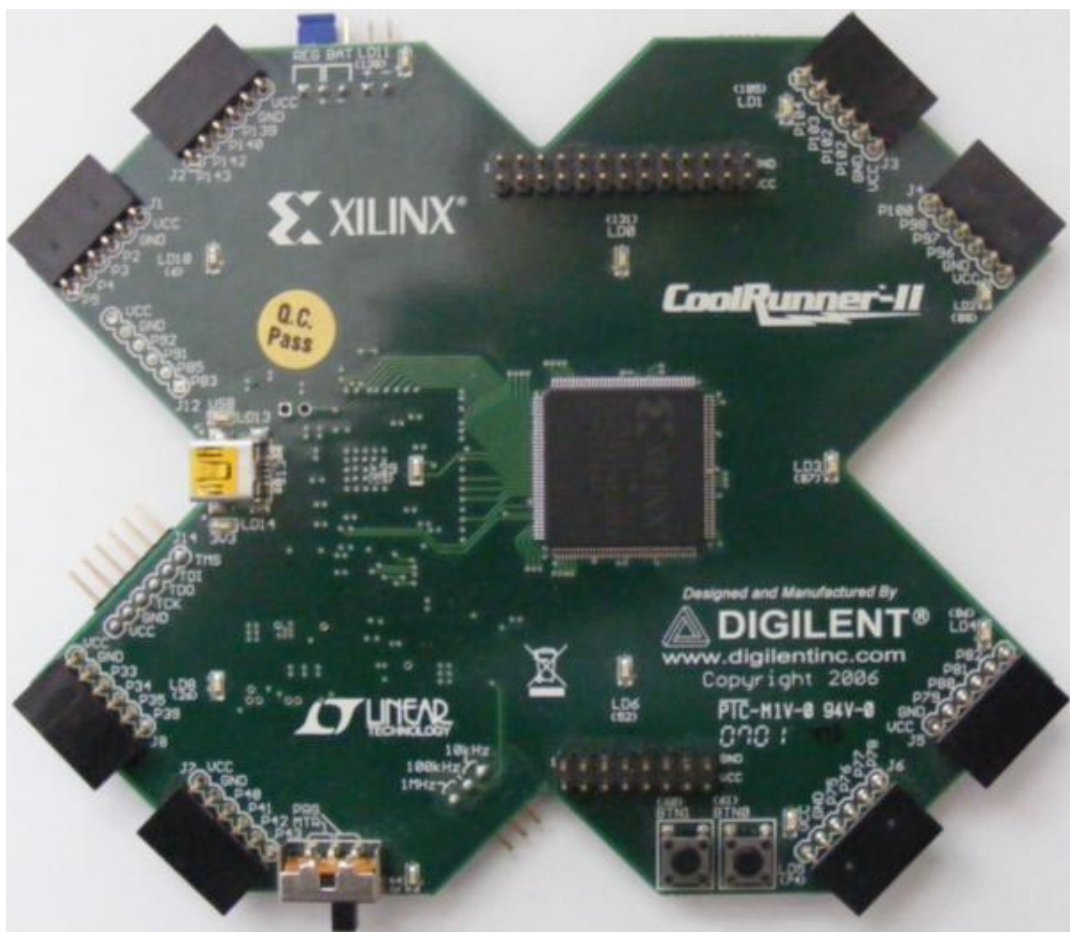


2. Technické zajištění

Pro vlastní technickou realizaci řešení bakalářské práce bylo kromě příslušného automobilu potřeba zajistit především vývojovou desku s osazeným hradlovým polem CPLD, servomechanismy pro ovládání natáčení zrcátka a externí FLASH paměť. Dále pak další drobný materiál, jako elektroinstalační materiál, elektronické prvky a ovládací tlačítka. Zařízení, která měla bezprostřední vliv na samotné řešení - návrh algoritmu, jsou popsána v následujících podkapitolách včetně příslušných parametrů. S těmito parametry se pak již dále v textu pracuje bez dalších odkazů.

2.1. Vývojová deska CoolRunnerII

V souladu se zadáním byla použita vývojová deska CollRunnurII Starter Kit od výrobce DIGILENT, viz obr. 1.



Obr. 1. Vývojová deska CoolRunnerII

Vývojová deska je osazena hradlovým polem CPLD XC2C256 firmy Xilinx s kapacitou 256 macrocell (místo pro program), viz obr. 2. V průběhu řešení se ukázalo, že tato kapacita byla největším technickým omezením pro řešení zadání a i pro ovládání jednoho zrcátka bylo v některých případech potřeba algoritmy "optimalizovat" z hlediska počtu potřebných prvků. Pro ovládání obou zrcátek v automobilu by pak byla potřeba větší kapacita hradlového pole (např. výrobce dodává 384 a 512 macrocell). Pole CPLD je programovatelné přes USB port na vývojové desce, napájení je možné realizovat také pomocí USB portu, nebo zdrojem napětí 3,6V-9V. Vnitřní frekvence je nastavena na 48MHz, případně je možné ji změnit pomocí vestavěného oscilátoru na hodnoty 1MHz, 100KHz nebo 10KHz.



Obr. 2. Obvod CPLD XC2C256

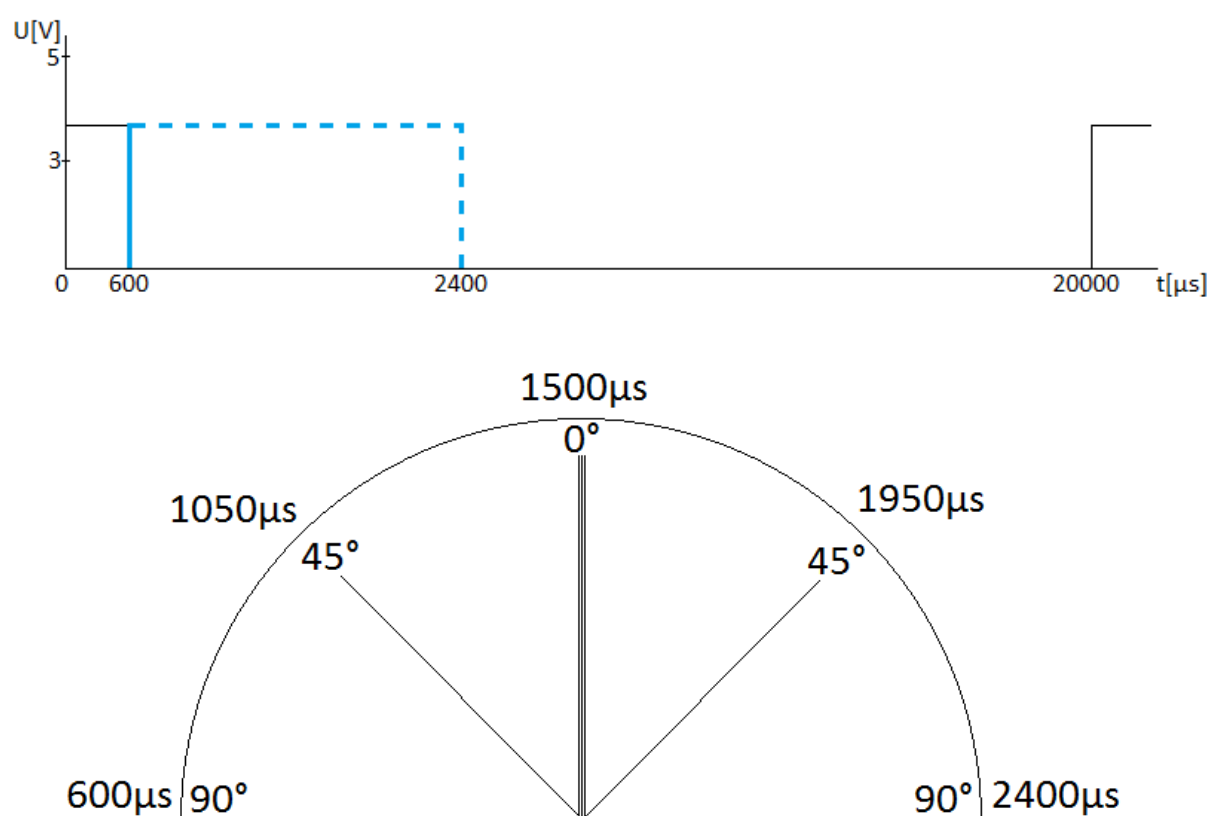
2.2. Připojení servomechanismů

Pro natáčení zrcátka byly použity dva servomechanismy, které jsou uloženy na místě původních analogových motorků a jsou napojeny na původní ovládací mechanismus. Díky tomu jsou úpravy a zásahy do konstrukce zrcátek minimální. Jeden servomechanismus ovládá natáčení ve vertikální ose, druhý v horizontální ose. Oba servomechanismy jsou na sobě nezávislé a díky tomu je možné zároveň natáčet zrcátko jak ve vertikální ose, tak v horizontální ose. Konkrétně byly použity modelářské servomechanismy HS-55 výrobce HITEC, jejichž rozměry jsou 22,8 x 11,6 x 24mm a



hmotnost činí 8g. Servomechanismy jsou připojeny pomocí třipinového konektoru, který obsahuje napájecí vodič, signálový vodič a společnou zem. Napájecí napětí je 4,8V-6 V, klidový odběr je 5,5mA a při zátěži může vystoupat až na 400mA.

Pro ovládání servomechanismu se používá pulsně šířková modulace (PWM). Doporučená frekvence pulsně šířkové modulace, kterou servomechanismy přijímají, je 50Hz. Úhel natočení servomechanismu je závislý na impulsech, které mají délku od 600 μ s do 2400 μ s. Přičemž tyto hodnoty zároveň udávají krajní polohy nastavení zrcátek, viz obr. 3. Střední poloha servomechanismu je tak dána impulsem o délce 1500 μ s. Délka pulsu PWM tak přímo udává polohu nastavení zrcátka, hodnota log1 je realizována napětím 3,3V. Pracovní rozsah, jakým servomechanismus disponuje, činí 180°.

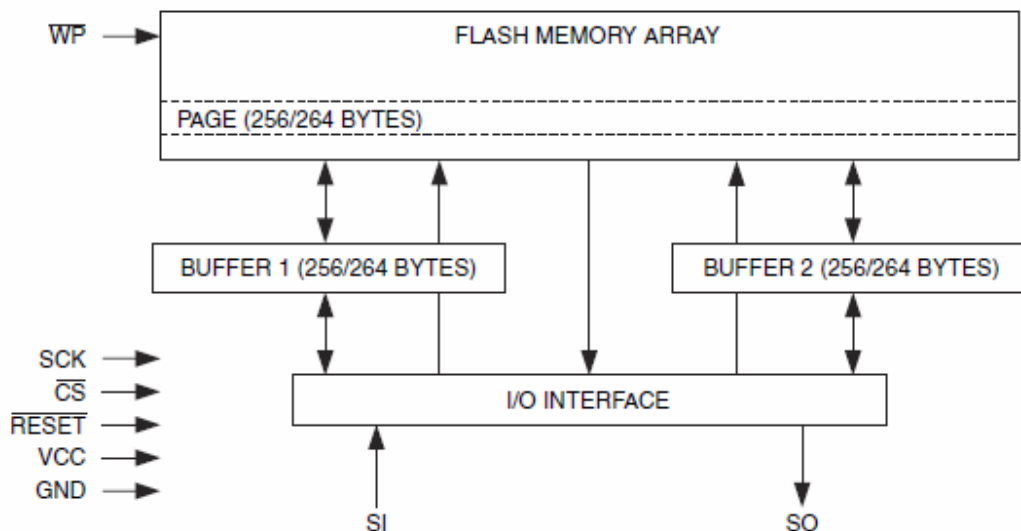


Obr. 3. Průběh PWM pro řízení úhlu natočení servomechanismu

2.3. Připojení paměti FLASH

Pro ukládání dat o požadovaném nastavení zrcátka byla použita paměť FLASH AT45DB081D firmy Atmel. Tato paměť disponuje velikostí 8 Mbit pro ukládání dat s maximální pracovní frekvencí 66 MHz. Ačkoliv tyto hodnoty daleko převyšují požadavky v rámci řešení bakalářské práce, byla paměť zakoupena z důvodu cenové a skladové dostupnosti. Pro napájení (a realizaci logického signálu) potřebuje paměť provozní napětí 2,7V-3,6V. Jako komunikační protokol s touto pamětí je možné použít sériovou komunikaci SPI Mode 0 a 3. Všechny tyto parametry splňují potřebné požadavky pro správné funkční připojení k desce CoolRunnerII. Mimo piny pro napájení a piny potřebné pro funkci SPI rozhraní paměť ještě disponuje pinem WP – Write Protect a pinem RESET. Pin Write Protect slouží pro uzamčení paměti proti jakémukoliv zápisu a pin RESET slouží pro ukončení jakékoliv aktuálně probíhané operace v paměti a přechodu do stavu nečinnosti.

Vnitřní uspořádání paměti je rozděleno do několika částí, viz obr. 4. Mezi základní části patří vstupně výstupní rozhraní, vyrovnávací paměti BUFFER1 a BUFFER2, každá o velikosti 264 bytu s možností nenávratného snížení na 256 bytu a vlastní paměť, která je ještě dále rozdělena na stránky, bloky a sektory.

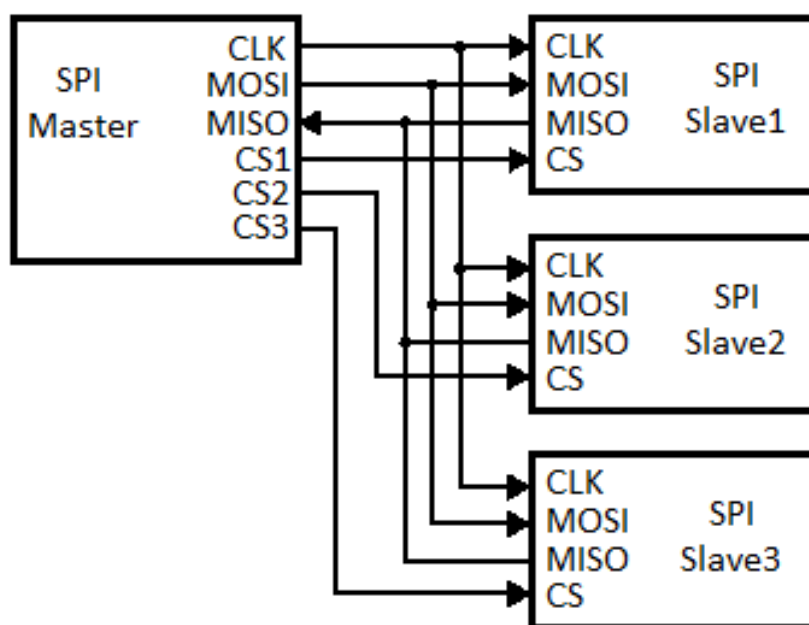


Obr. 4. Vnitřní uspořádání paměti, [1]

Pro komunikaci mezi vývojovou deskou a pamětí je použito sériového periferního rozhraní (SPI). Sériový typ komunikace na rozdíl od paralelního se vyznačuje nižším množstvím použitých vodičů pro komunikaci, jelikož bity jsou posílány po jednom za sebou na rozdíl od posílání více bitů v jednom okamžiku paralelně na více linkách. Sériová

komunikace se dá ještě rozdělit na half-duplex a full-duplex. V případě SPI, kdy posílání a příjem dat má každý svůj vodič, se jedná o full-duplex přenos dat, čili je možno data zároveň posílat i číst. V případě half-duplexu je pro posílání i příjem dat možno použít pouze jeden vodič a tak je tedy v jednom okamžiku možno data buď posílat, nebo přijímat.

Pro existenci SPI je zapotřebí jednoho obvodu master a jednoho nebo více obvodů zvaných slave -viz obr. 5. SPI pro svoji komunikaci potřebuje čtyři piny. Jedná se o pin CS - Chip-Select, CLK - Clock, MOSI - Master Out-Slave In a MISO – Master In-Slave Out.



Obr. 5. Zapojení master a slave obvodů

První zmíněný pin CS zajistí vybrání správného chipu pro komunikaci v případě, že na SPI je připojeno více než jedno slave zařízení. V případě, že k řídicímu obvodu je připojeno jen jedno slave zařízení, je pin CS držen v logice, jež slave zařízení bere jako zapnutou komunikaci. Ovšem některé zařízení, jako i v našem případě paměť FLASH, potřebují pro správnou komunikaci aktivaci pomocí změny logiky na pinu CS.

Pin CLK je použit pro hodinový signál, díky kterému je možnost synchronizovat přenos dat mezi master a slave zařízením.

Pinem MOSI jsou přenášena veškerá data z master zařízení do zařízení slave. To znamená jak příkazy, tak data.

Pin MISO slouží pro master zařízení k příjmu dat odeslaných slave zařízením.

Vedle těchto pinů se můžou vyskytnout ještě další, které závisí od použitého slave zařízení a které rozšiřují jeho funkce. V případě naší paměti se jedná již o výše zmíněné piny WP – Write Protect a pin RESET.

Komunikace SPI podporuje čtyři módy – 0,1,2 a 3. Tyto módy se liší ve využití hodinového signálu. Vztah mezi hodinovým signálem a daty se určuje dvěma konfiguračními bity CPOL a CPHA, kde platí pro nastavení:

CPOL = 0; klidová úroveň hodinového signálu je log0,

CPOL = 1; klidová úroveň hodinového signálu je log1,

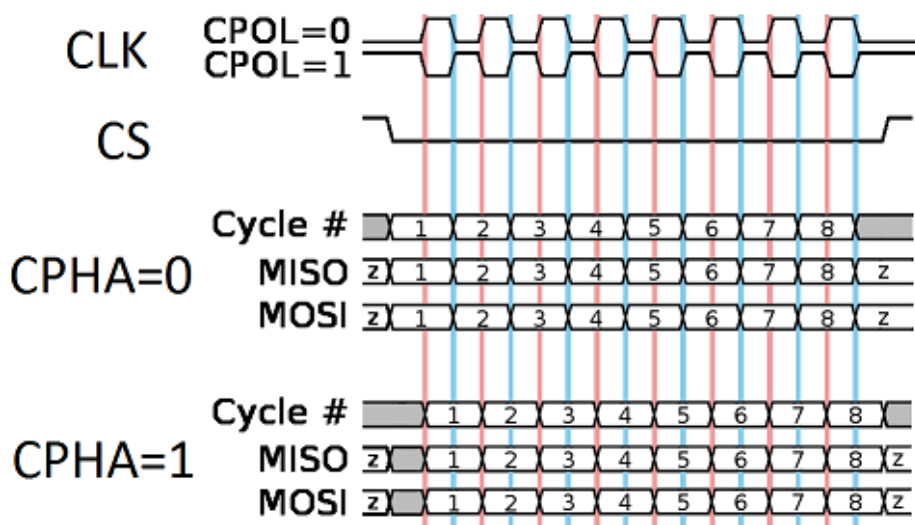
CPHA = 0; hodnota je čtena při vzestupné hraně,

CPHA = 1; hodnota je čtena při sestupné hraně,

což můžeme vyjádřit pomocí následující tabulky

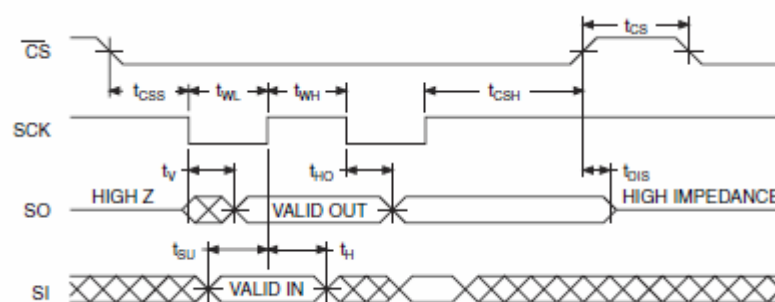
Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

případně znázornit na obr. 6.



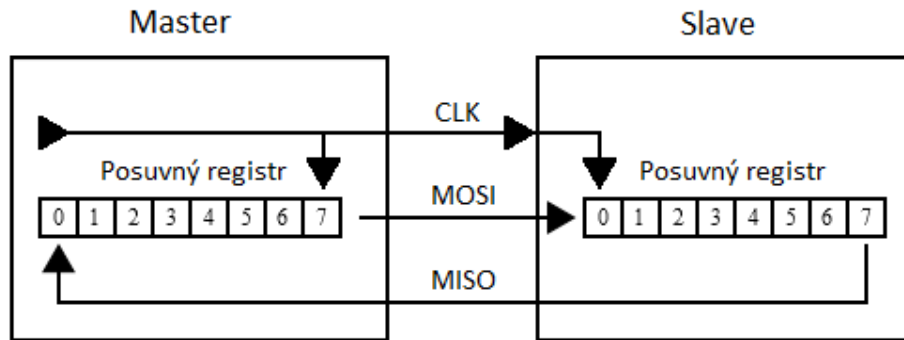
Obr. 6. Vztah mezi hodinovým signálem a přenášenými daty

Pro úspěšnou komunikaci s pamětí je zapotřebí dodržet přesně dané instrukce a jejich sled. Tato činnost je detailně popsána v datovém listu použité paměti. V našem případě je použit typ komunikace nazývaný MODE3, viz obr. 7. V klidovém stavu paměti, kdy paměť nepřijímá ani nevysílá data, je pin CS i pin hodinového signálu CLK nastaven na log1. V tomto případě se piny MOSI a MISO ignorují, tudíž nezáleží na tom, v jakém stavu se nacházejí (jestli se po nich něco vysílá).



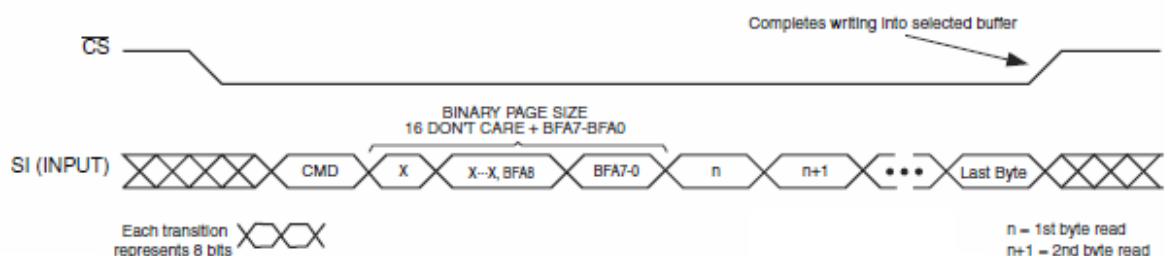
Obr. 7. Časový sled signálů pro MODE3, [1]

Pro navázání komunikace s pamětí je v první řadě zapotřebí změnit logickou hodnotu pinu CS z log1 na log0. Tím se paměť aktivuje a může začít přijímat instrukce a data. Zároveň je zapotřebí synchronizovat FLASH paměť s řídicím obvodem pomocí hodinového signálu. Tento signál je v našem případě v době nečinnosti nastaven stejně jako signál CS na log1. První hrana po aktivaci paměti tak bude sestupná. Před první sestupnou hranou je zapotřebí ještě dodržet minimální interval, během kterého se vnitřní program paměti připraví na přijímání hodinového signálu a dat. Tento čas je u použité paměti nastaven na hodnotu minimálně 5 ns. Při první sestupné hraně zároveň začíná přenos dat, jednotlivé požadavky přenosu jsou popsány níže (čtení, zápis). To probíhá synchronně s hodinovým signálem tak, že na sestupnou hranu se bity mění a na hranu vzestupnou se bity čtou. Pro správnou synchronizaci a odesílání dat je v programu použito několika automatů, pro příjem dat jsou použity posuvné registry s organizací přenosu MSB, obr. 8.



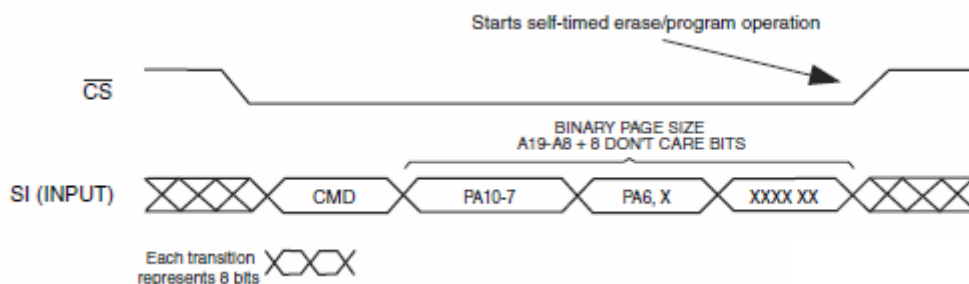
Obr. 8 Princip přenosu dat pro zařízení master a slave

Následuje popis jednotlivých instrukcí použitých v programu při komunikaci s pamětí. Aby bylo co číst, je potřeba data nejdříve zapsat. Pro zápis dat do paměti je zapotřebí požadovaná data nejdříve přesunout (zapsat) do jednoho ze dvou bufferů a až poté data nechat zapsat do hlavní paměti. Proces zápisu dat do bufferu je znázorněn na obrázku 9. Po aktivaci paměti pinem CS a zajištění hodinového signálu je zapotřebí paměti poslat osmibitový příkaz, označený jako CMD, podle kterého paměť bude nakládat s dalšími přijatými daty. Tento příkaz má pro naše potřeby vždy hodnotu 10000100. Po odeslání příkazu následuje odeslání 24 bitů, které určují počáteční adresu v bufferu, na kterou se následně odesílaná data začnou ukládat. Tato data začneme posílat ihned po odeslání adresy. Jelikož paměť je pro příjem těchto dat omezena a po naplnění bufferu začne dále přijímanými daty přepisovat buffer od začátku, je zapotřebí si množství těchto dat hlídat. Po přenosu posledního bitu našich dat se komunikace s pamětí ukončí návratem pinu CS do log1. Tím celý proces přenosu dat do bufferu končí.



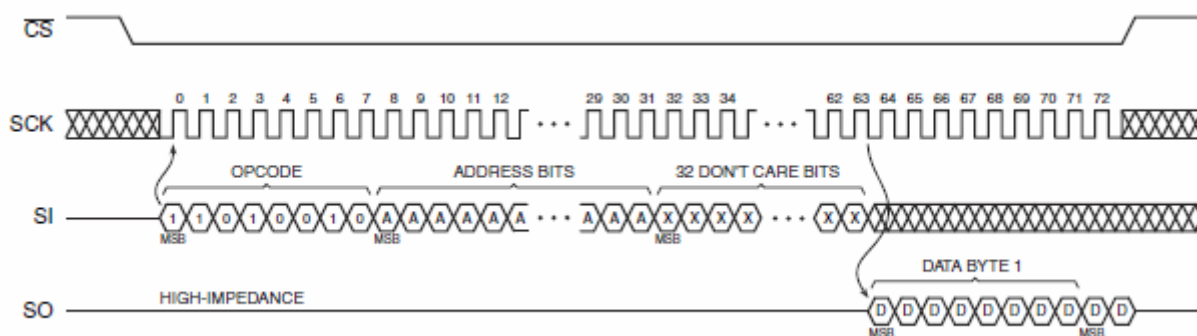
Obr. 9. Časový sled signálů pro zápis do bufferu paměti, [1]

Data z bufferu se do hlavní paměti zapíší podle následujícího obrázku 10. Pro tento proces je zapotřebí paměti předat pouze příkaz CMD, který je 10000011 a následně odeslat 24 bitů, které určí adresu pro uložení dat v paměti.



Obr. 10. Časový sled signálů pro zápis z bufferu do hlavní paměti, [1]

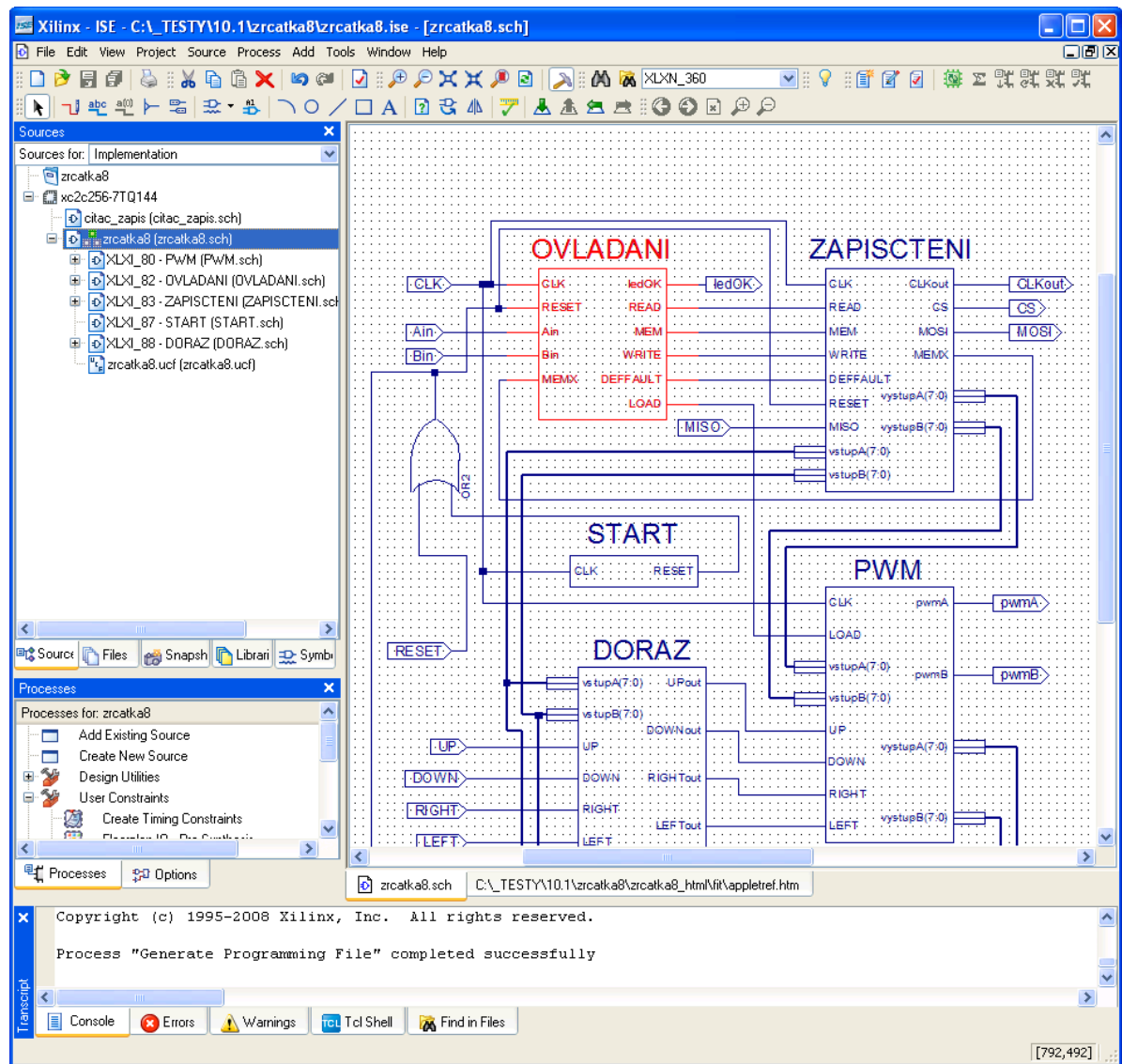
Proces načtení uložených dat je zobrazen na obrázku 11. Na tomto obrázku můžeme vidět detailně rozkreslené jednotlivé stavy všech komunikačních vodičů po jednotlivých bitech. Po aktivaci paměti, zapnutí hodinového signálu a odeslání příkazu pro čtení z paměti s požadovanou adresou následuje samotné čtení dat z paměti po vodiči MISO. Během tohoto přenosu musí zůstat pin CS v log0 a hodinový signál musí být stále vysílán. Po načtení všech dat bude paměť uvedena do stavu nečinnosti.



Obr. 11. Časový sled signálů pro čtení z paměti, [1]

3. Návrh funkčního schéma v prostředí Xilinx ISE

Důležitou částí podpory každého výrobce hradlových polí by měl být potřebný vývojový software. Firma Xilinx, jejíž výrobek je v bakalářské práci použit, nabízí pro potřeby vývojářů na svých internetových stránkách k volnému použití produkt ISE WebPack. Je to prostředí, které umožňuje celý vývoj projektu od jeho založení, přes tvorbu potřebných souborů, jejich simulaci až po tvorbu konfiguračního souboru. Dá se říci, že vzhled tohoto prostředí je standardní vzhledem k podobným produktům pracujícím s formuláři (viz obr. 12.). Při základních znalostech z oblasti hradlových polí je práce téměř intuitivní, případně je dostatečně k dispozici nápověda.

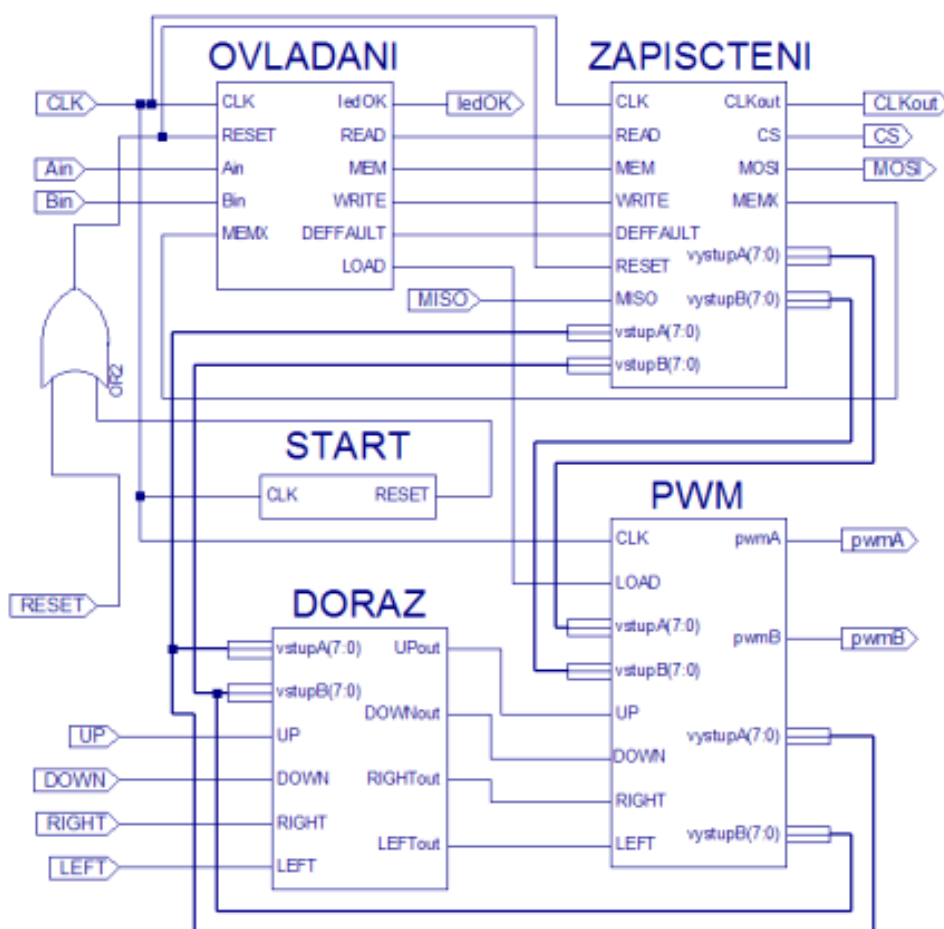


Obr. 12. Prostředí pro tvorbu projektů v ISE WebPack

Dále jsou popsány konkrétní realizace v tomto vývojovém prostředí, které tvoří podstatnou část řešení bakalářské práce. Celý program je modulově rozdělen do 3 částí, celkem čtyř modulů, které spolu komunikují:

- moduly pro zadání vstupních údajů od řidiče (DORAZ, OVLADANI)
- modul pro ovládání servomechanismů (PWM)
- modul pro komunikaci s externí pamětí (ZAPISCTENI)

Celé schéma je uvedeno obr. 13. Jednotlivé moduly (bloky schématu) budou postupně podrobněji popsány z hlediska funkčního zapojení a s popisem jednotlivých signálů.



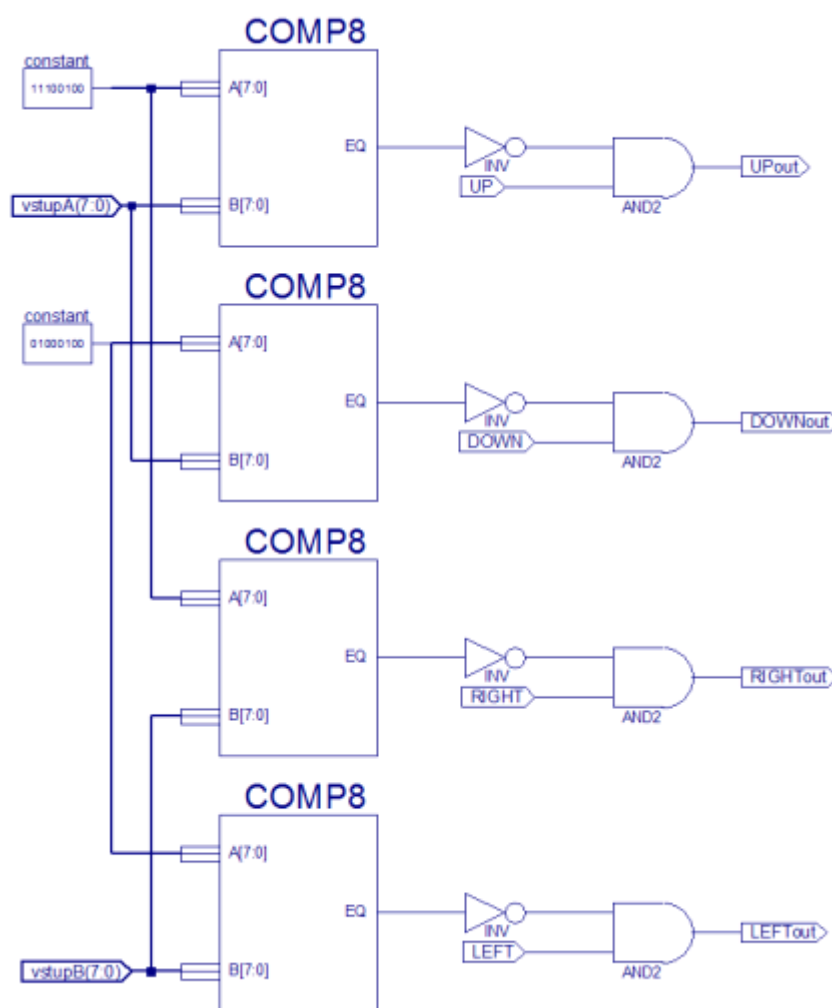
Obr. 13. Blokové schéma čtyř základních modulů

3.1. Vstupní moduly pro nastavení polohy

Z hlediska vstupních údajů jsou zpracovávány dvě informace - informace o požadavku řidiče pomocí ovládacích prvků (zadávání požadovaného směru pohybu zrcátka z kabiny automobilu) a informace k uložení resp. vyvolání požadované polohy. Informace z ovládacích prvků je zpracovávána v bloku DORAZ, kde je zároveň ošetřeno i

konstrukční omezení pohybu zrcátka. Blok OVLADANI pak zpracovává informaci o požadavku na uložení aktuální polohy zrcátka, nebo naopak vyvolání již uložené hodnoty k jejímu nastavení.

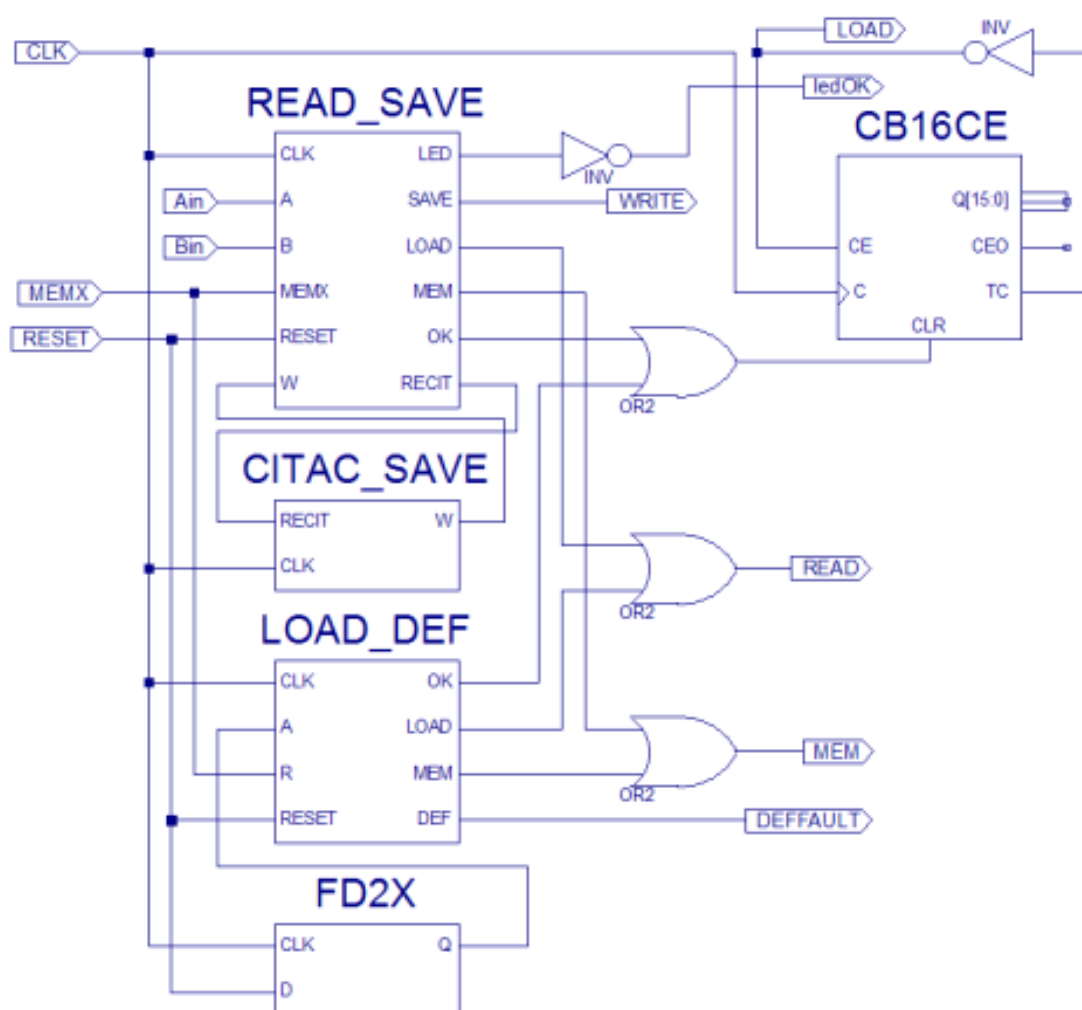
Na obr. 14. je uvedeno funkční schéma pro blok DORAZ. Výstupem tohoto bloku jsou čtyři piny, které představují směr natočení zrcátka, jedná se o piny UPout, DOWNout, RIGHTout a LEFTout. V případě, že některý z těchto pinů bude mít hodnotu log1, bude se zrcátko natáčet požadovaným směrem. Hodnota log1 bude na výstupu za podmínky, že



Obr. 14. Blokové schéma modulu DORAZ

aktuální hodnota polohy servomechanismu je v povoleném intervalu, tudíž negovaný výstup z komparátoru má hodnotu log1, a příslušný požadovaný vstup UP, DOWN, RIGHT nebo LEFT je také hodnoty log1. Tyto vstupy nastavuje uživatel pomocí ovladače v kabině auta. V případě, že zrcátko dosáhne krajní polohy, bude mít negovaný výstup z komparátoru hodnotu log0 a tudíž požadovaný výstup nebude moci nabývat hodnoty log1.

O nastavení a hlídání těchto krajních poloh se v programu stará několik komparátorů. Tato ochrana funguje na principu porovnávání požadované hodnoty pro natočení servomechanismu s krajními hodnotami poloh, které servomechanismus nesmí překročit. K porovnávání těchto hodnot je použito komparátorů COMP8, jejichž vstupy jsou hodnota aktuální polohy servomechanismů vstupA a vstupB a hodnoty krajních poloh constant. Hodnota aktuální polohy servomechanismu je získána z bloku PWM (viz kapitola 3.2.). O nastavení hodnoty krajní polohy je zmíněno v kapitole 2.1.



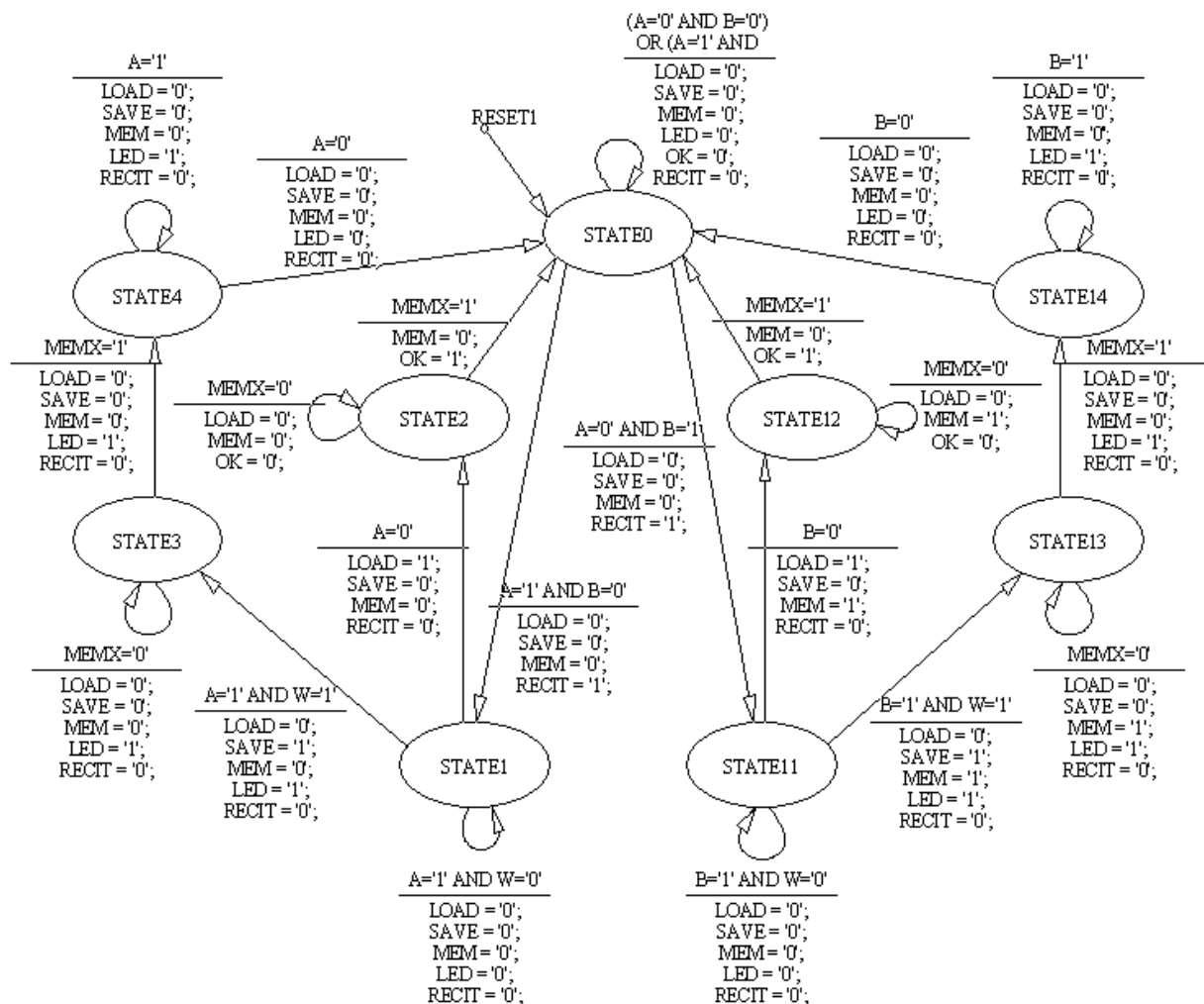
Obr. 15. Blokové schéma modulu OVLADANI

Struktura bloku OVLADANI je znázorněna na obr. 15. Uživatelským vstupem tohoto bloku jsou vstupy Ain a Bin, každý z těchto vstupů představuje fyzicky jedno tlačítko. Pomocí každého z tlačítek je možno polohu jak uložit, tak načíst. Hlavním konstrukčním prvkem tohoto bloku jsou dva automaty, READ_SAVE a LOAD_DEF. Dále je tu čítač CB16CE, který nastavuje dostatečně dlouhý impuls pro aktivaci načtené polohy.

První z automatů se stará o požadavky uživatele, druhý automat načítá z paměti výchozí (defaultní) polohu zrcátka. Tato poloha je důležitá při prvním spuštění programu, což zahrnuje i spuštění po přerušení napájení {načtení této hodnoty se provádí automaticky}. Při tomto prvním spuštění programu jsou totiž hodnoty čítačů nastaveny tak, že by jim odpovídající poloha zrcátka byla mimo povolený rozsah a tak by hrozilo zničení mechanismu natáčení zrcátka nebo samotného servomechanismu. Načtení této výchozí polohy tedy zajistí správnou funkci celého programu.

Automat **READ_SAVE**, jehož schéma jde vidět na obr. 16., obsluhuje tlačítka pro uložení a načtení polohy. Současný model je osazen dvěma tlačítky, uložení a načtení jedné a té samé polohy se provádí jedním tlačítkem a to tak, že dlouhé stisknutí tlačítka, které trvá déle než 1,8sec provede uložení aktuální polohy zrcátka a krátké stisknutí tlačítka, kratší než 1,8sec, provede načtení předtím uložené polohy. Automat pro zpracování vstupů z těchto dvou tlačítek má 9 stavů, stav STATE0, ve kterém se automat nachází v době „nečinnosti“, a po čtyřech stavech pro každé tlačítko. Stavy STATE1 až STATE4 reprezentují činnost jednoho tlačítka (proměnná A) a stavy STATE11 až STATE14 činnost druhého tlačítka (B).

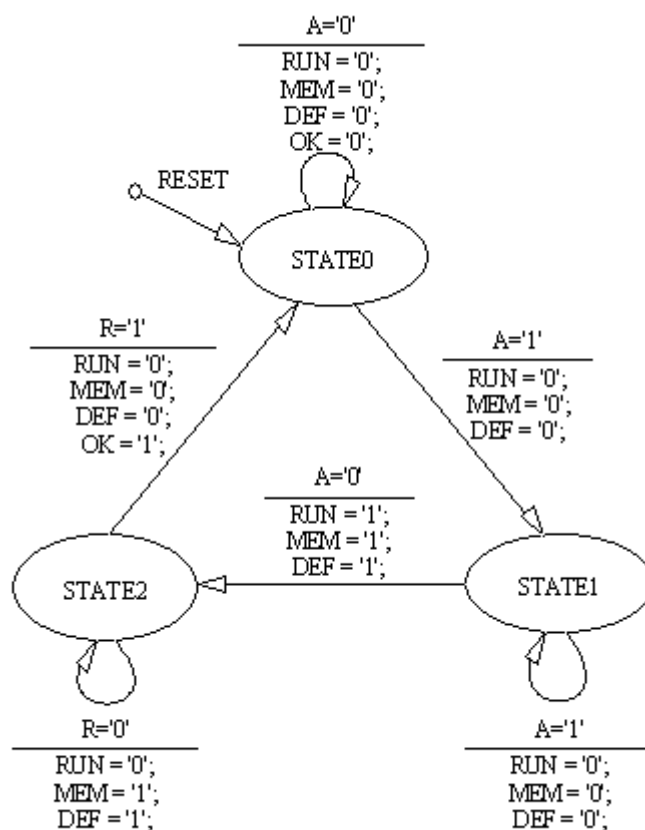
Po stisknutí jednoho ze dvou tlačítek přejde příslušná proměnná (A nebo B) ze stavu log0 na log1 a celý proces uložení nebo načtení polohy se aktivuje. Údaj o tom, o které tlačítko se jedná, je uložen na proměnné MEM. Jelikož se jedná o analogicky stejný proces pro obě tlačítka, bude už další popis uveden jen pro tlačítko A. Pro režim uložení polohy (tlačítko bylo stisknuto déle jak na 1,8sec), se nastaví hodnota proměnné A ze stavu log0 do stavu log1. Tím se automat nastaví ze stavu STATE0 do stavu STATE1, při tomhle kroku se výstup RECIT nastaví na log1 a dojde tak k restartování externího čítače CITAC_SAVE, viz obr. 15. Tento inkrementální čítač načítá do hodnoty 1,8sec a po uplynutí této doby se nastaví výstup čítače na log1. Tento výstup je přiveden na vstup W do automatu READ_SAVE. V případě, že po této době je tlačítko stále stisknuté, tudíž vstup A i W je log1, přejde automat do stavu STATE3. Při tomto přechodu se nastaví výstup SAVE na log1. Tento výstup aktivuje automat BUFFER_WRITE, viz obr. 19., který se stará o uložení aktuální polohy zrcátka. Navíc se výstup LED nastaví do log1, čímž může být uživateli oznámeno, že proces uložení začal a může tak uvolnit tlačítko. Toto oznámení může být provedeno například světelně nebo zvukově. V našem případě je to provedeno světelně a to rozsvícením diody P20 na desce CoolRunnerII.



Obr. 16. Blokové schéma automatu READ_SAVE

Ve stavu STATE3 automat setrvává tak dlouho, dokud nedostane od automatu MEMORY_WRITE, který ukládá polohu zrcátka, informaci o ukončení uložení. Tato informace se objeví na vstupu MEMX. Po uložení automat ještě přejde do stavu STATE4, kde čeká na eventuelní puštění tlačítka. Pokud je tlačítko puštěno a vstup A je tedy log0, přejde automat do stavu STATE0 a čeká na další příkaz. V případě, že je tlačítko stisknuté kratší dobu než 1,8sec, přejde automat do stavu STATE2. Automat toto pozná pomocí výše zmíněného čítače ve stavu STATE1, což znamená, že vstup A nabyl hodnoty log0 dříve než vstup z čítače W hodnoty log1. Při tomto přechodu se výstup LOAD změní na log1 a díky tomu se spustí automat MEMORY_READ, který načte požadovanou hodnotu z paměti. Po tomto načtení dostane automat informaci na vstupu MEMX. Pomocí výstupu OK aktivuje načtenou polohu a vrátí se do stavu STATE0.

Činnost druhého automatu **LOAD_DEF** (viz schéma na obr. 15.) popíšeme podle obr. 17. Automat má tři stavy, a vedle vstupu hodinového a resetovacího signálu má ještě dvě řídicí proměnné A a R a čtyři výstupy (RUN, MEM, DEF, OK). Ze stavu STATE0, kdy mají všechny výstupy hodnotu log0 a ve kterém automat nijak neovlivňuje chod programu, může přejít do stavu STATE1 v případě změny řídicí proměnné A z log0 na log1. To se může stát ve dvou případech, první případ je výše zmíněný při prvním spuštění programu a druhý případ je při stisknutí resetovacího tlačítka. Do stavu STATE2 se automat dostane opět buď automaticky (spuštění programu), nebo po uvolnění resetovacího tlačítka. V obou těchto případech se řídicí proměnná A vrátí do hodnoty log0. Při přechodu do tohoto stavu se nastaví všechny výstupy na hodnotu log1.



Obr. 17. Blokové schéma automatu LOAD_DEF

Výstup RUN spustí automat pro čtení z paměti a výstup MEM a DEF předává automatu pro čtení informací na jaké adrese je uložena defaultní hodnota. Setrvání ve stavu STATE2 a přechod do stavu STATE0 je od teď závislý pouze na vstupní proměnné R.

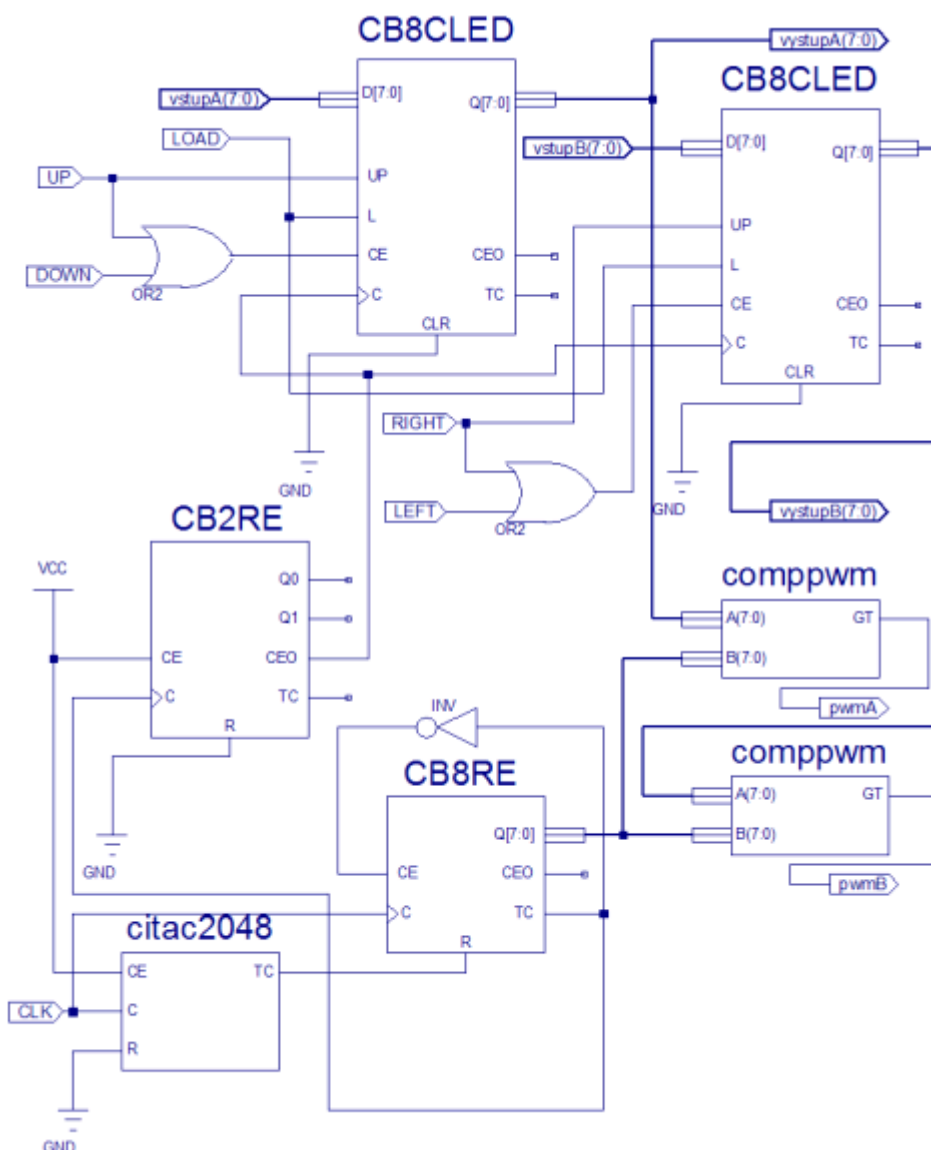
Ve stavu STATE2 automat setrvává, dokud je na vstupu R log0. V tomto stavu jsou také výstupní proměnné MEM a DEF stále v hodnotě log1. Výstup RUN je opět log0, protože pro spuštění čtecího automatu MEMORY_READ stačilo, aby hodnota log1 byla pouze po dobu jedné periody hodinového signálu. Do stavu STATE0 se automat vrátí ihned, jak dostane od čtecího automatu informaci o ukončení čtení z paměti, což signalizuje změna R na log1, při tomto přechodu se zároveň výstup OK změní na log1 a spustí tím přenos načtených hodnot do čítače v bloku PWM. Tím celý proces skončí a zrcátka se natočí do defaultní polohy.

3.2. Ovládání servomechanismů

Pro natáčení jednoho zrcátka jsou použity dva nezávislé servomechanismy, které jsou připojeny na původní ovládací mechanismus zrcátka. Jeden servomechanismus je použit pro natáčení ve vertikální ose, druhý pro natáčení v horizontální ose. Servomechanismy jsou ovládány pomocí signálu pulsně šířkové modulace PWM, viz kapitola 2.2. Frekvence PWM kterou použité servomechanismy přijímají je 50Hz. Tato frekvence nemusí být dodržena přesně, ale může být nastavena na hodnotu blízko této doporučené. Pro generování frekvence je použit 11bitový čítač citac2048. Tento inkrementální čítač počítá do hodnoty 2048, po dosažení maximálního čísla se automaticky restartuje a počítá znovu. Díky frekvenci 100Hz, s kterou čítač počítá, je dosaženo frekvence 48,83Hz, která je pro servomechanismy ještě použitelná. Úhel natočení servomechanismu je dán hodnotami impulsů, které mají délku od 600 μ s do 2400 μ s. Přičemž hodnoty 600 μ s a 2400 μ s udávají krajní polohy nastavení zrcátka. Střední polohu servomechanismu potom udává impuls o délce 1500 μ s. Délka řídicího impulsu je zajištěna pomocí osmibitového čítače, osmibitového čísla a komparátoru, který v našem případě neporovnává rovnost, ale používá relace větší, menší. V případě, že by byl použit obyčejný komparátor, bylo by ještě zapotřebí použít RS-klopný obvod. osmibitový čítač CB8RE pracuje také na 100Hz frekvenci, a jelikož počítá do hodnoty 256, trvá mu to 2560 μ s. Jeden přírůstek čítače trvá 10 μ s, což znamená, že rozsah potřebný pro plné využití chodu servomechanismu je rozdělen na 180 kroků. Ačkoliv je rozlišovací schopnost servomechanismu 8 μ s, což je 225kroků na dráze servomechanismu, a tudíž se při vzorkování po 10 μ s připravíme o 45 kroků, tak je i tak 180kroků plně dostačujících pro plynulé a přesné nastavení zrcátka. Osmibitové číslo získáváme z čítače CB8CLED, který

umožňuje čítat jak nahoru, tak dolu a navíc má vstup pro načtení osmibitového čísla. Těchto rozšířených funkcí se využívá při prodlužování nebo zkracování délky řídicího impulsu pulsně šířkové modulace a při načítání polohy z paměti.

Ovládání servomechanismů je realizováno v bloku PWM viz schéma na obr. 12. a následovně popsáno dle obr. 18. Jedenáctibitový čítač citac2048 generuje každou periodu jeden impuls, který restartuje čítač CB8RE. Tento čítač se sám nerestartuje, nýbrž se při dosažení svého maxima zastaví a čeká na další impuls k restartu. Tím je zajištěn v každé periodě jeden řídicí impuls pro daný servomechanismus. Tento průběh je pro všechny použité servomechanismy stejný, tudíž nám tento základní signál stačí generovat jednou a dále ho pak jen rozdělit pro každý servomechanismus.



Obr. 18. Blokové schéma ovládání servomechanismů

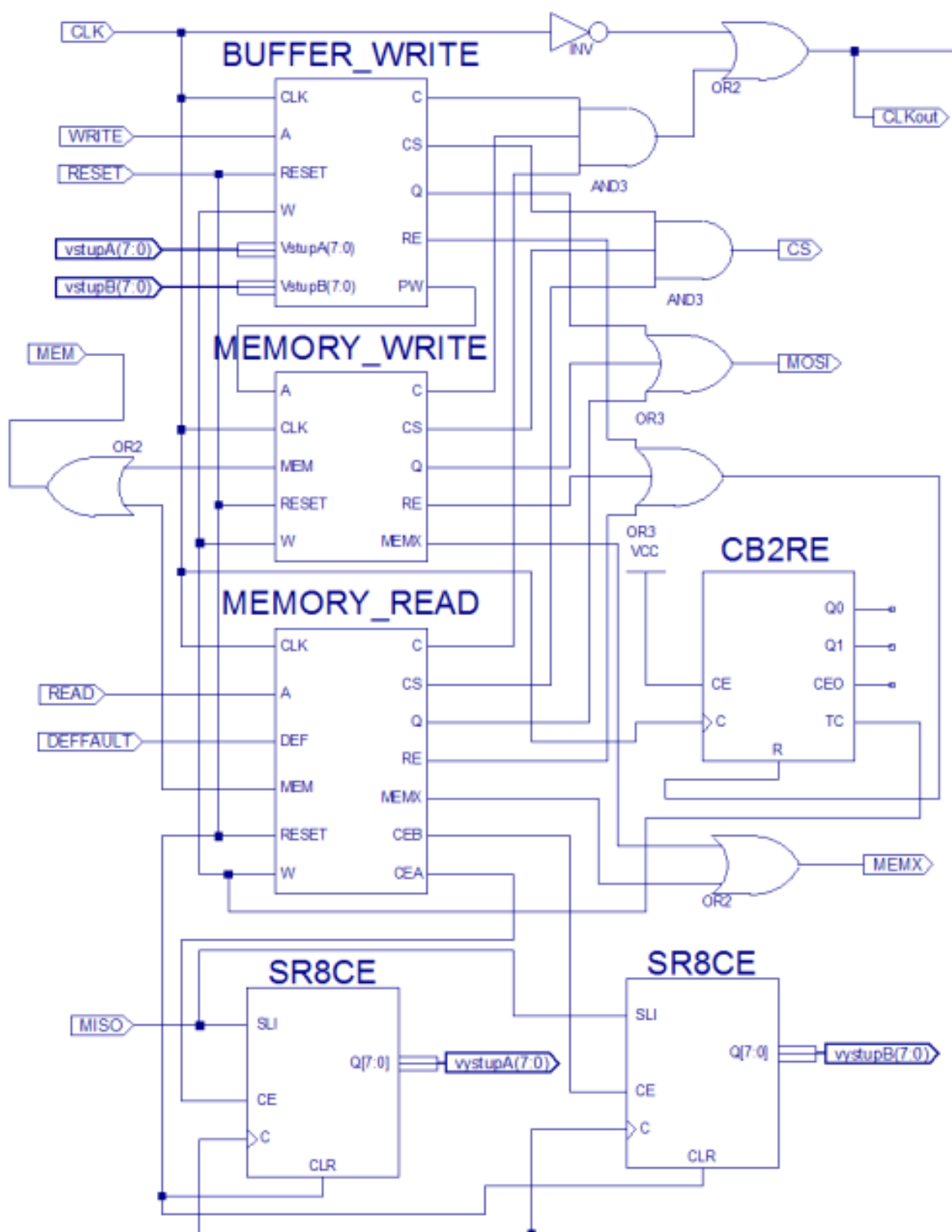
Ke generování výstupního impulsu požadované délky dochází v komparátoru, jehož výstupem je log1, dokud je hodnota ze společného čítače menší, než je hodnota z čítače který udává délku impulsu. Jakmile společný čítač CB8RE dosáhne větší hodnoty, výstup se přepne na log0 a zůstává v této hodnotě do konce periody. Na začátku další periody dojde opět k restartu 8bitového čítače CB8RE. Jelikož jeho hodnota bude nižší, než hodnota čítače CB8CLED, udávající délku impulsu, přepne komparátor compppwm výstup do log1 a vše se opakuje. Z důvodu plynulého nastavení zrcátka je 100Hz vstupní frekvence pro čítače udávající délku řídicího impulsu upravena pomocí čítače CB8RE a CB2RE. Tím se docílí dostatečně pomalé a tudíž přesné natáčení zrcátka.

Hodinový signál je přiveden vstupem CLK. Hodnota polohy načtená z paměti je přivedena vstupy vstupA a vstupB, jedná se o osmibitovou paralelní sběrnici. Tyto hodnoty se aktivují v případě, že vstup načtení bude mít hodnotu log1. Poslední čtveřice vstupů UP, DOWN, RIGHT a LEFT zajišťuje změnu délky řídicího impulsu pro servomechanismy. Pro každý servomechanismus jsou použity dva vstupy a to jeden pro prodloužení impulsu a druhý pro jeho zkrácení. Výstup tohoto bloku tvoří dva signály pulsně šířkové modulace, které jsou vyvedeny přímo na signálový pin servomechanismu a 2 výstupy, vystupA a vystupB, které slouží pro odesílání hodnoty polohy, která může být uložena do paměti.

3.3. Komunikace s FLASH pamětí - zápis a čtení

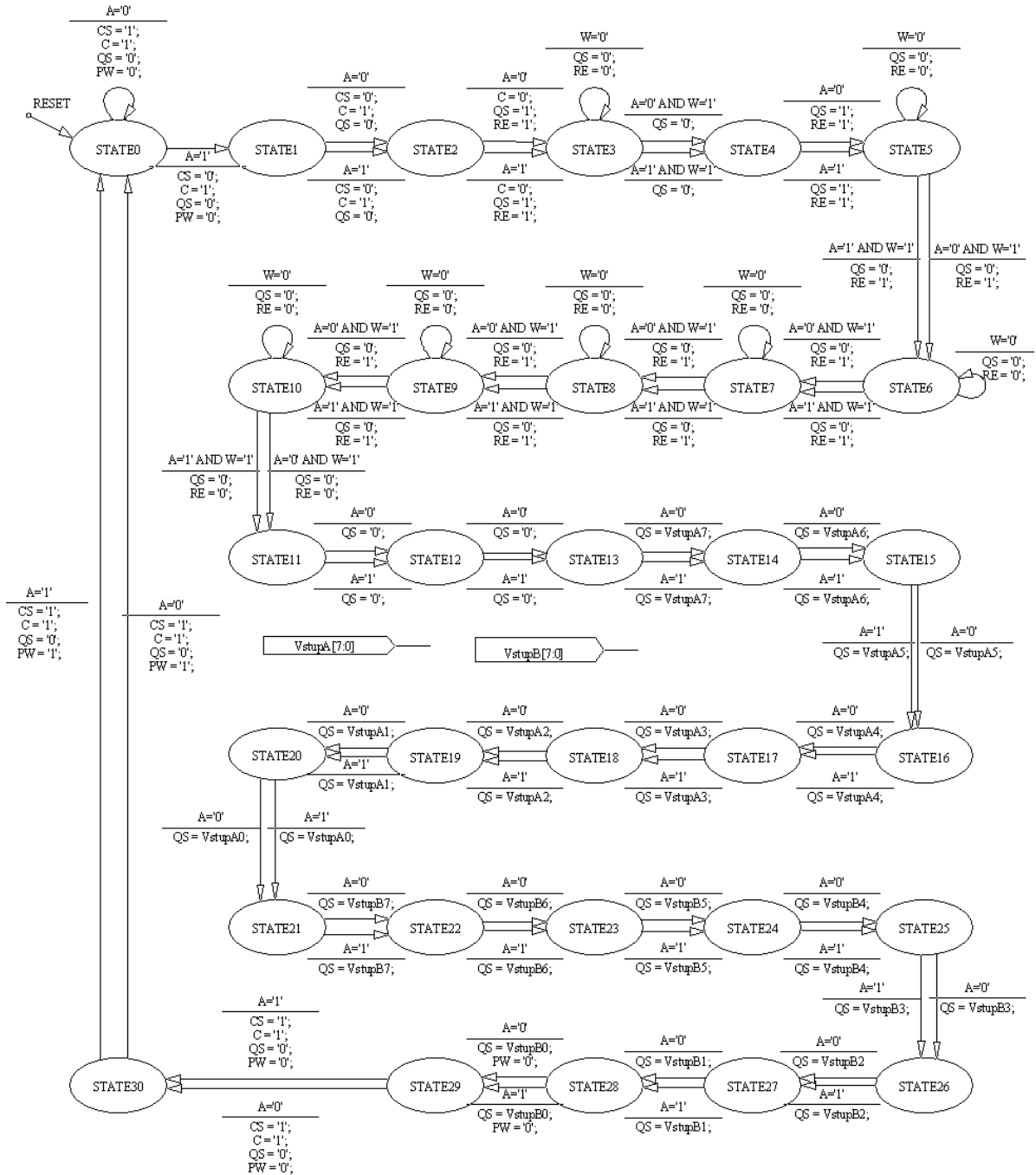
Ukládání do externí paměti probíhá ve dvou fázích. Nejprve je potřeba ukládaná data zapsat do bufferu a následně uložit data z bufferu do hlavní paměti, kde mohou být data uchována bez napájení paměti až 20let, zatímco v bufferu jsou data po přerušení napájení smazána. Blokové schéma zapojení je uvedeno na obr. 19. Mimo třech automatů jsou zde dva posuvné registry SR8CE a 2bitový čítač CB2RE. Posuvné registry se starají o načtení polohy z paměti. Tato poloha je z paměti načítána pomocí pinu MISO a správné načasování načítání je zajištěno pomocí hodinového signálu, který je shodný se signálem paměti a automatu MEMORY_READ, který mimo jiné aktivuje tyto posuvné registry ve správnou chvíli na správnou dobu. Pomocí čítače je zase možné zredukování počtů stavů jednotlivých automatů. To se provádí tak, že pokud se v automatu objeví tři stejné po sobě jdoucí stavy, automat resetuje čítač a setrvává v aktuálním stavu, dokud čítač neodpočítá tři hodinové cykly. Po této době čítač odešle automatu signál a automat přejde na další stav. Pro komunikaci s pamětí tu jsou mimo vstupu MISO tři výstupy, a to CS, MOSI a CLKout,

který posílá paměti hodinový signál, jenž je inverzní vůči vnitřnímu hodinovému signálu. Tato inverze je použita za účelem správné komunikace mezi automaty a pamětí.



Obr. 19. Blokové schéma pro komunikaci s FLASH pamětí

Automat **BUFFER_WRITE**, viz obr. 20., se stará o zapsání hodnot aktuální polohy zrcátka do bufferu paměti, aby pak tyto hodnoty mohly být dále zapsány do hlavní paměti. Tento automat má také jako jediný ze všech použitých automatů 2 paralelní vstupy VstupA a VstupB (umístěny uprostřed schéma), které slouží pro načtení hodnot polohy jednotlivých servomechanismů. Automat setrvává ve stavu STATE0 do doby, než na jeho proměnné A přijde log1. Od této chvíle se automat začne přesouvat postupně po jednotlivých stavech nezávisle na proměnné A, dokud neskončí zpátky ve stavu STATE0.



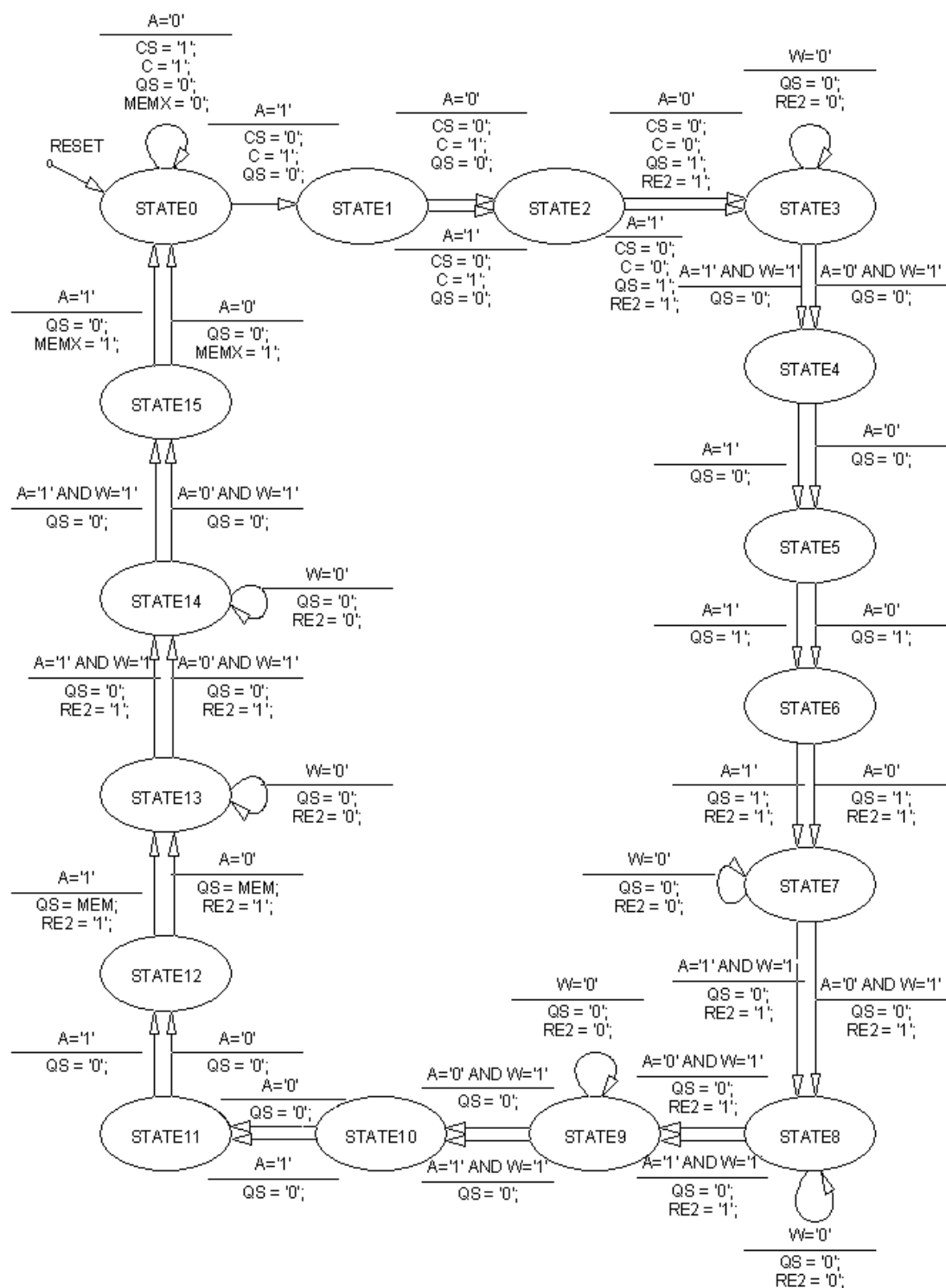
Obr. 20. Blokové schéma pro automat BUFFER_WRITE

V případě uložení dat je ovšem nejdříve potřeba aktivovat paměť a začít vysílat hodinový signál. To se provede ve stavech STATE1 a STATE2. Aby byla dodržena potřebná prodleva, je nejdříve změněna hodnota pinu CS z log1 na log0 a stejně tak i hodnota pinu C, který aktivuje hodinový signál. V následujících 8 stavech je pomocí výstupu Q poslán paměti osmibitový příkaz, v tomto případě se jedná o příkaz pro uložení dat do bufferu1. Zde je již vidět, že některé stavy můžou setrvávat ve smyčce. Tato smyčka je právě řízena čítačem CB2RE, který je pomocí výstupu RE restartován a pomocí vstupu W je možno pokračovat dál. Celkem je v tomto automatu 6 smyček. Protože každá trvá tři hodinové cykly, je ušetřeno 18 stavů. Po odeslání příkazu pak následuje 24bitů, které určují počáteční adresu v bufferu, na kterou se data začnou zapisovat. Jelikož je každá poloha v paměti uložena ve své adresní stránce, můžeme každou polohu začít ukládat od nulové adresy a proto je výstup Q po celou tuto dobu nastaven na hodnotu log0. Dále následuje 16bitů, které začínají přenášet aktuální hodnotu polohy zrcátka do paměti. První je přenášena poloha pro servomechanismus A. Jelikož přenos začíná bitem MSB, přenesení se jako první bit vstupA7, následovaný bitem vstupA6. Až se přenesou všech 8 bitů ze vstupA, začnou se stejným způsobem přenášet bity ze vstupB. Tento přenos probíhá tak, že se hodnoty jednotlivých bitů načítají do výstupu Q, který vše odesílá do paměti.

Až se přenesou poslední bit polohy, bude zapotřebí přehrát buffer1 do hlavní paměti. K tomu ovšem bude zapotřebí poslat paměti nový příkaz, což je možné provést po ukončení a opětovném otevření komunikace s pamětí. V dalším stavu automatu se tedy může komunikace s pamětí ukončit, což se provede změnou výstupu CS a C na log1 a navíc se ještě výstupem PW pošle signál automatu MEMORY_WRITE, který se postará o uložení nahraných dat do hlavní paměti.

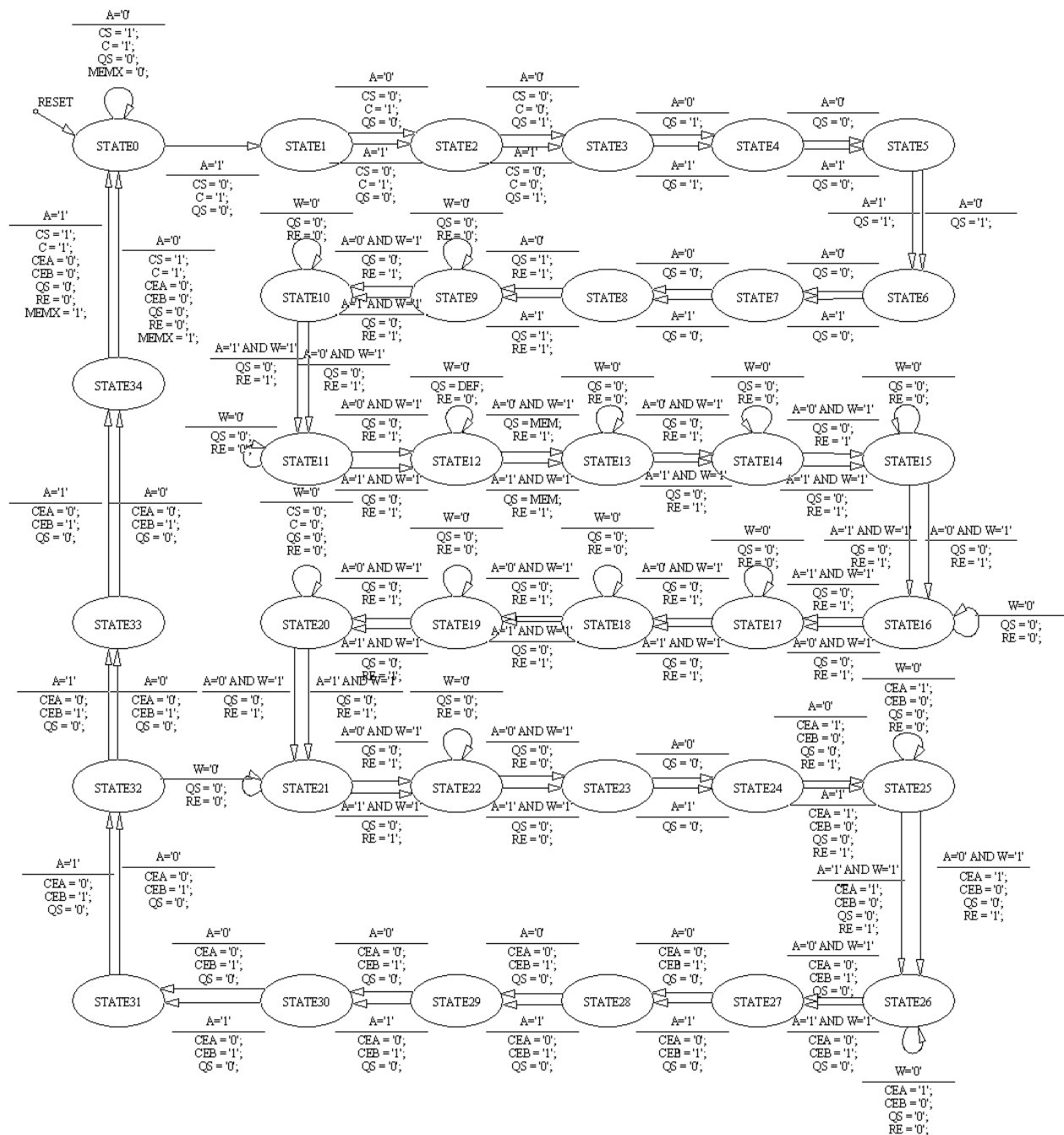
Zde tedy ke slovu přichází automat **MEMORY_WRITE**, jedná se o nejkratší ze zde použitých automatů pro ovládání paměti. Pro tuto operaci je tudíž zapotřebí také nejméně informací. Jelikož je základní posloupnost jako aktivace a ukončení komunikace s pamětí stejná, budou zde popsány jen hlavní rozdíly oproti předchozímu automatu. Po aktivaci paměti a odeslání osmibitového příkazu, který udává paměti, že bude zapisovat data z bufferu1 do hlavní paměti, následují tři byty, které udávají, na jakou adresu v paměti se data uloží. Tady je už zapotřebí data pro jednotlivé uložené polohy ukládat na různé adresy v paměti, aby se navzájem nepřepsala. To je zajištěno vstupní proměnnou MEM, která se pomocí výstupu Q odešle do paměti v okamžiku, kdy paměť přijímá informace o adrese

k zapsání dat. Na konci automatu je ještě výstup MEMX, který sdělí ovládacímu automatu RERAD_SAVE, že data byla zapsána do paměti. Jak je vidět na obr. 21, tak automat MEMORY_WRITE obsahuje 16 stavů.



Obr. 21. Blokové schéma pro automat MEMORY_WRITE

Poslední ze tří automatů **MEMORY_READ** zajistí opětovné načtení uložených dat z paměti. Tento automat je znázorněn na obr. 22. a je naopak od předchozího ze všech největší.



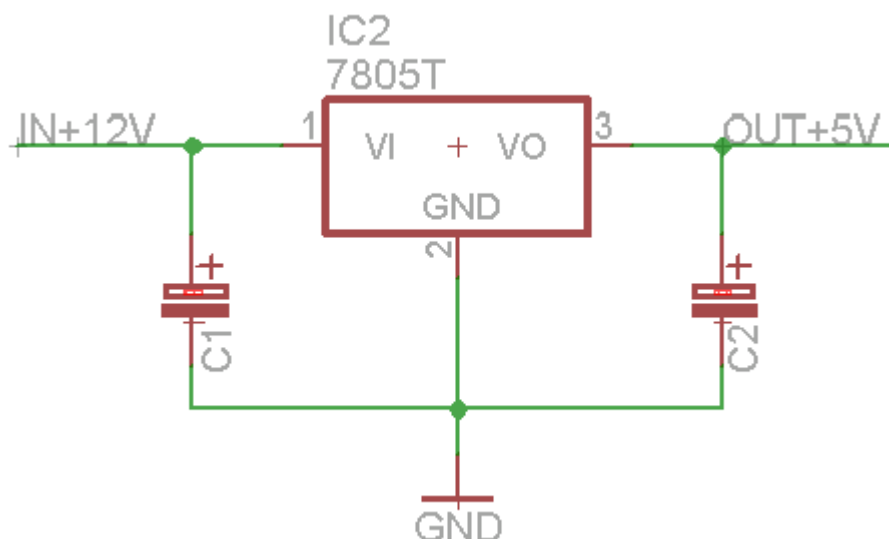
Obr. 22. Blokové schéma pro automat MEMORY_READ

I přes jeho složitost je ovšem princip fungování stejný jako u obou předchozích automatů. Po odeslání osmibitového příkazu čtení z paměti následuje 24bitů udávajících adresu, ze které se data budou číst. Adresa se opět určí pomocí vstupu MEM, který přes výstup Q odešle adresu do paměti. Po odeslání adresy následuje 32bitů, na kterých takzvaně nezáleží, je tedy jedno co budeme do paměti posílat. Jelikož nyní budeme data z paměti přijímat, tak ani na dalších odeslaných bitech nezáleží, ovšem nyní nám už paměť začíná přes pin MISO posílat požadovaná data. Tato data jsou načítána do výše zmíněných posuvných registrů SR8CE, viz kapitola 3.3. Tyto registry se pro příjem dat aktivují pomocí výstupů CEA a CEB. Nejdřív se aktivuje výstup CEA, který zůstává aktivní pro příjem 8 bitů prvního servomechanismu (natočení zrcátka v jedné ose). Poté se výstup deaktivuje a aktivuje se výstup CEB, který je opět aktivní tak dlouho, aby přijal dalších 8 bitů z paměti pro ovládání druhého servomechanismu (natočení v druhé ose). Po tom, co se informace z paměti načtou do posuvných registrů, je ještě potřeba pomocí výstupu MEMX poslat ovládacímu automatu READ_SAVE informaci, že data byla načtena. Ten už jen načtená data aktivuje a zrcátka se tak nastaví do požadované polohy.

4. Realizace modulu pro ovládání zrcátek

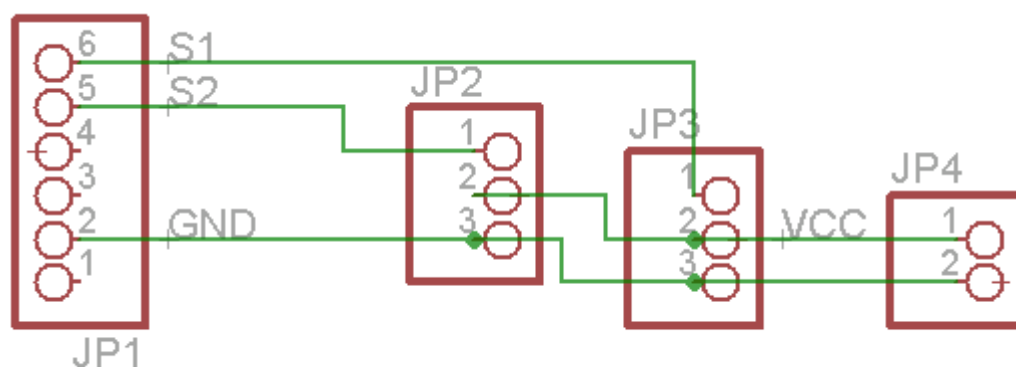
V této kapitole jsou popsány zbývající kroky k celkovému technickému řešení bakalářské práce, jako je napájení, připojení FLASH paměti a dalších ovládacích prvků k desce CoolRunnerII, či implementace servomechanismů do zrcátka automobilu.

Pro napájení desky CoolRunnerII je zapotřebí 3,6V-9V a servomechanismy pracují s napětím 4,8V-6V, tuto hodnotu ovšem 12ti voltová napájecí síť v automobilu nesplňuje. Proto je tedy zapotřebí použít vhodného regulátoru k dosažení použitelné hodnoty napětí. Za tímto účelem byl použit lineární regulátor napětí. Jedná se o regulátor MC7805CT v pouzdře TO220 výrobce ON Semiconductor. Tento regulátor poskytuje při vstupním napětí 7V-25V výstupní napětí 5V a maximální výstupní proud 1A, což našim potřebám plně vyhovuje. Jak je vidět na schématu obrázku 23, je pro zapojení tohoto regulátoru ještě zapotřebí dvou kondenzátorů proti rozkmitání obvodu. Kondenzátor C1 o kapacitě 100 μ F je použit pro vstupní napětí a druhý kondenzátor C2 o kapacitě 10 μ F je na výstupním napětí. To nám zajistí potřebné napájení všech použitých komponent z 12V napájecí sítě automobilu.



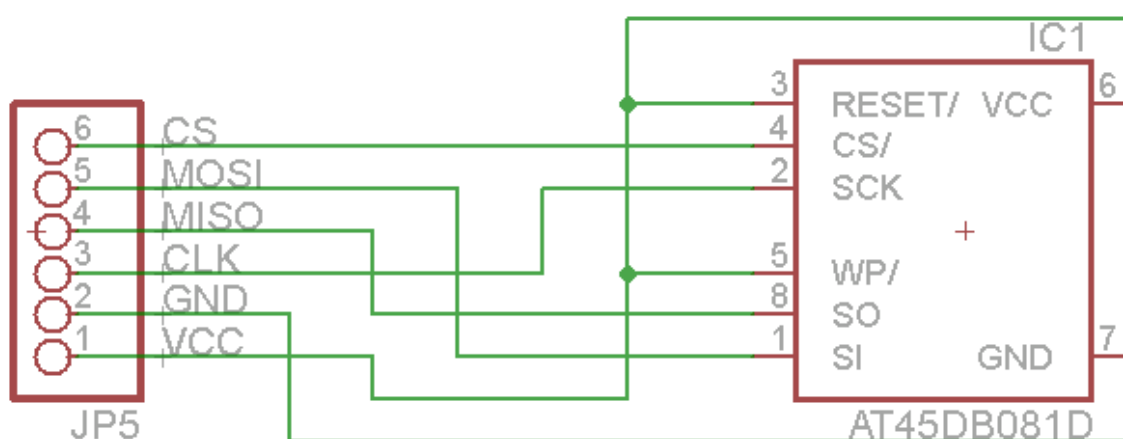
Obr. 23. Schéma úpravy napájecího napětí

FLASH paměť a servomechanismy jsou připojeny k desce CoolRunnerII pomocí šestipinových konektorů. Na jeden tento konektor je možné připojit buď jednu FLASH paměť nebo až čtyři servomechanismy. Schéma modulu pro připojení servomechanismů je vidět na obrázku 24. Nachází se zde jeden šestipinový konektor JP1, dva třípinové konektory JP2 a JP3 pro připojení jednotlivých servomechanismů a jeden dvoupinový konektor JP4 pro připojení externího napájení. Šestipinový konektor slouží pro připojení k vývojové desce. Na 5. a 6. pinu je přiveden přes vodiče S1 a S2 řídicí signál pro nastavení servomechanismů a na pinu č. 2 je pro tyto signály přivedena zem. Vedle řídicích signálů potřebují servomechanismy pro svoji funkci i napájení. To je přivedeno přes konektor JP4 a rozvedeno do konektorů pro jednotlivé servomechanismy.



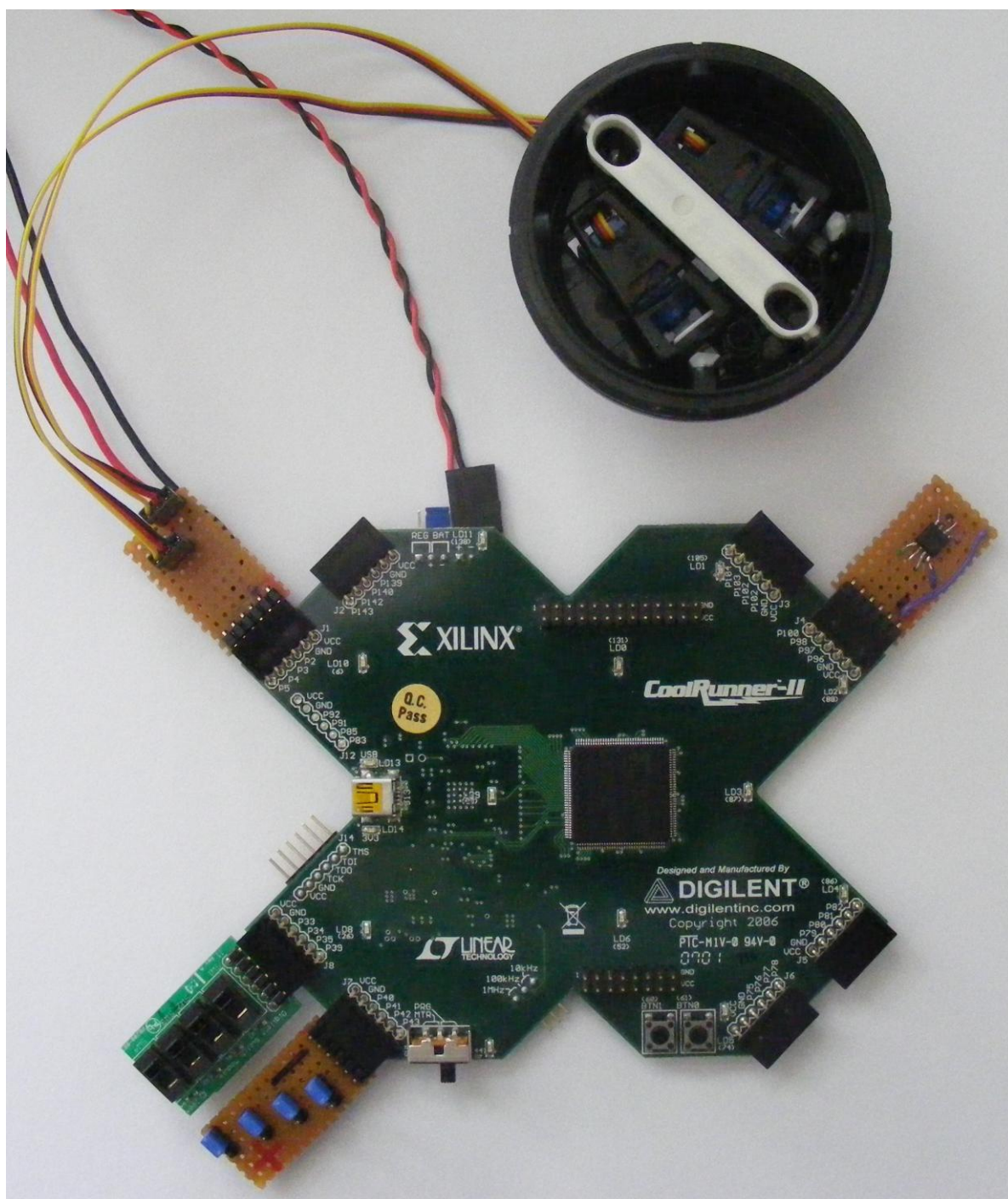
Obr. 24. Schéma pro připojení servomechanismů

Paměť FLASH pro své připojení vyžaduje minimálně 6 pinů. Proto stačí jeden šestipinový konektor. Vedle pinů GND a VCC pro napájení je zde na pozici číslo 3 vyveden hodinový signál, na pozici 4 a 5 jsou vstupy a výstupy pro přenos dat mezi deskou CoolRunnerII a pamětí a na pozici 6 je řídicí pin CS. Paměť obsahuje celkem 8 vývodů, ale jelikož nevyužíváme možnosti Write Protect a RESET, jsou tyto piny paměti na doporučení výrobce permanentně připojeny k napájení. Tím se nám potřebný počet pinů zredukuje z osmi na šest a na paměti se tyto piny zapojí podle schématu na obr. 25.



Obr. 25. Schéma připojení FLASH paměti

Celá realizace pro odzkoušení navrženého ovládání zpětných zrcátek automobilu je uvedena na obr. 26. K vývojové desce CoolRunnerII jsou připojeny oba servomechanismy, zabudované v korpusu ovládání skutečného zrcátka, dále je připojena paměť FLASH a ovládací prvky pro pohyb zrcátka a uchování hodnot nastavení do paměti.



Obr. 26. Celkový pohled na modul inteligentního ovládání zpětného zrcátka

Závěr

Tématem bakalářské práce je navržení a realizace ovládání zpětného zrcátka automobilu s možností zapamatování a pozdějšího opětovného nastavení konkrétní polohy. Návrh byl zpracován nejen po stránce programu k vlastnímu ovládání, ale včetně kompletního technického zajištění. Navíc komponenty, jako např. ovládací servomechanismy, byly voleny tak, aby vlastní realizace mohla být provedena na skutečném automobilu.

Pro realizaci byla použita vývojová deska CoolRunnerII, která je osazena hradlovým polem CPLD firmy Xilinx. Vlastní návrh algoritmu ovládání byl proveden ve vývojovém prostředí Xilinx ISE. Dále byly použity dva modelářské servomechanismy a paměť FLASH. Pro ladění celé úlohy byla vývojová deska CoolRunnerII doplněna ovládacími prvky (které nahradily skutečné ovládání v automobilu) a nezbytnými obvody pro zajištění potřebného zdroje napětí.

Navržené řešení je plně funkční a domnívám se, že je možné jej použít pro většinu běžně vyráběných automobilových zrcátek. Ošetřeny jsou i některé situace, důležité zejména z pohledu dlouhodobého provozu, jako např. konstrukční omezení pohybu zrcátka, tj. aby na krajních polohách nebyly zbytečně namáhány servomechanismy, nebo možnost externího resetování celého programu v případě výpadku napájení ap. Nutno přiznat, že předložená práce realizuje ovládání pouze jednoho zrcátka a to z důvodu kapacity hradlového pole. Realizace pro obě zpětná zrcátka neznamena ale žádný technický problém, pouze je zapotřebí vyšších finančních nákladů na zakoupení dalších komponent.

Literatura

- [1] Atmel Corporation. AT45DB081D [online] 4. 5. 2010
URL:< www.atmel.com/dyn/resources/prod_documents/doc3596.pdf>
- [2] Balada R. Řízení serva [online] 23. 10. 2009
URL:< <http://hw.cz/novinky/art3140-rizeni-serva-projekt-do-mmia.html>>
- [3] Digilent. X-Board Reference Manual [online] 2. 1. 2007
URL:< http://www.digilentinc.com/Data/Products/X-BOARD/X-Board_Reference_Manual.pdf>
- [4] ON Semiconductor. MC7800 - 1 A Positive Voltage Regulators [online] 1. 9. 2009
URL:< http://www.onsemi.com/pub_link/Collateral/MC7800-D.PDF>
- [5] Pinker J., Poupa M. Číslicové systémy a jazyk VHDL, Praha: BEN, 2006, 352 s., ISBN 80-7300-198-5
- [6] Xilinx, Inc. XC2C256 CoolRunner-II CPLD [online] 8. 3. 2007
URL:< http://www.xilinx.com/support/documentation/data_sheets/ds094.pdf>