TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Elektronická řídicí jednotka pro účely samořízeného modelu auta

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Ing. Petr Hons**
Vedoucí práce: Ing. Petr Mrázek, Ph.D.

Liberec 2016

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2015/2016**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ing. Petr Hons**
Osobní číslo:	**M13000186**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Elektronická řídicí jednotka pro účely samořízeného modelu auta**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Proveďte rozbor současného stavu elektronické řídicí jednotky automaticky řízeného vozu na autodráhu.
2. Na základě rozboru navrhněte vlastní HW řídicí jednotky.
3. Naprogramujte základní SW rutiny pro přímou komunikaci s HW a navrhněte řešení systémové části, tzn. vytvořte funkce umožňující jednoduché programování zařízení.
4. Na laboratorní dráze ověřte funkčnost celého zařízení zvoleným algoritmem řízení.

Rozsah grafických prací:	**dle potřeby dokumentace**
Rozsah pracovní zprávy:	**cca 40–50 stran**
Forma zpracování diplomové práce:	**tištěná/elektronická**

Seznam odborné literatury:

[1] **Hrbáček, Jiří: Komunikace mikrokontroléru s okolím – 1.díl, BEN – technická literatura, 2002, 160 s. ISBN 80-86056-42-2.**

[2] **Herout, Pavel: Učebnice jazyka C, KOPP, 2009, 280s., ISBN 978-80-7232-383-8.**

[3] **Daďo, S.; Kreidel, M.: Senzory a měřicí obvody. Praha: ČVUT, druhé vydání, 1999, ISBN 80-01-02057-6.**

Vedoucí diplomové práce:	**Ing. Petr Mrázek, Ph.D.** Ústav mechatroniky a technické informatiky
Konzultant diplomové práce:	**Ing. Jan Koprnický, Ph.D.** Ústav mechatroniky a technické informatiky
Datum zadání diplomové práce:	**10. října 2015**
Termín odevzdání diplomové práce:	**16. května 2016**

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2015

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou 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 diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou 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.

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

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 12.5.2016

Podpis:

Na tomto místě chci poděkovat panu Ing. Petru Mrázkovi, Ph.D. za skvělé vedení diplomové práce a za cenné připomínky v průběhu řešení.

Dále chci poděkovat panu Ing. Janu Koprnickému, Ph.D. za cenné konzultace a obětavou spolupráci.

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Abstrakt

Tato diplomová práce se zabývá návrhem elektronické řídicí jednotky pro účely samořízeného modelu auta. První část práce se věnuje rozboru problematiky různých mikrokontrolérů od několika výrobců. V další části se tato práce věnuje rozboru elektronického obvodu a výběru vhodných elektronických komponent. (mikrokontrolér, zdroj, obvod řízení motorku, akcelerometr a další) Součástí diplomové práce je také návrh elektronického obvodu a desky plošných spojů (DPS). V poslední části se tato práce věnuje realizaci řídícího programu a jeho ověření v praxi.

Klíčová slova

Samořízený model auta, Mikrokontrolér, Zdroj, Akcelerometr, Pulsně šířková modulace (PWM), deska plošných spojů (DPS).

Abstract

This master thesis deals with the design of the electronic control unit for self-controlled car model. The first section of the thesis deals with the analysis of the different microcontrollers from several manufacturers. In the next section of the thesis deals with the analysis of electronic circuit and selecting the appropriate electronic components. (microcontroller, power supply, motor control circuit, accelerometer and others) The thesis also includes the design of electronic circuits and printed circuit boards (PCB) In the last section of the thesis deals with the implementation of the control program and its verification in practice.

Key words

Self-controlled car model, Microcontroller, Power supply, Accelerometer, Pulse-width modulation (PWM), Printed circuit board (PCB).

Obsah

Seznam obrázků a tabulek....8
Seznam použitých zkratek....9
Úvod....11

1 Teoretický rozbor problematiky mikroprocesorů....12

1.1 Popis procesoru....12
1.2 Stručná historie procesoru....15
1.3 Dělení procesorů podle jejich vlastností....20
1.4 Obvody a technologie zvyšující výpočetní výkon....28
1.5 Rodiny mikrokontrolérů a obvodů SoC....33
1.5.1 Mikrokontroléry 8051....33
1.5.2 Mikrokontroléry PIC....34
1.5.3 Mikrokontroléry MSP430....35
1.5.4 Mikrokontroléry AVR....35
1.5.5 Rodina SoC x86....36
1.6 Architektura procesorů ARM....37

2 Teoretický rozbor řízení modelu auta....41

2.1 Související fyzikální vztahy....41
2.2 Regulace....44
2.3 Filtrování signálu....45
2.3.1 Integrační článek....46
2.3.2 Klouzavý průměr....47
2.3.3 FIR filtr....48
2.3.4 IIR filtr....49

3 Rozbor stavu současné řídící jednotky vozu na autodráhu....50

4 Kritéria pro volbu součástek pro novou řídící jednotku....51

4.1 Mikrokontrolér....51
4.2 Zdrojová část....52
4.3 Akcelerometr....53
4.4 H-můstek....54
4.5 Paměť FLASH....54
4.6 Ostatní součástky....55

5 Rozbor částí elektronického zapojení....56

5.1 Zdrojová část....56
5.2 Řídící část....56

5.3 Akcelerometr....57
5.4 H-můstek....57
5.5 Paměť FLASH....57
5.6 Konektory....57
6 Deska plošných spojů....58
7 Software....59
7.1 Vývojové nástroje....59
7.2 Popis řídícího programu....61
7.2.1 Časovače a volání procesů....63
7.2.2 Sériová komunikace....63
7.2.3 Zápis dat to paměti FLASH....64
7.2.4 Měření a zpracování signálů....65
7.2.5 Regulátor....68
8 Ověření funkčnosti a praktické testy....69
Závěr....71
Literatura a informační zdroje....72
Příloha A: Schéma obvodu plošného spoje....74
Příloha B: Deska plošných spojů – horní strana (TOP)....75
Příloha C: Deska plošných spojů – spodní strana (BOTTOM)....76
Příloha D: Seznam součástek....77

Seznam obrázků a tabulek

Obr.1 Vnitřní uspořádání procesoru Intel 8080
Obr.2 Vnitřní uspořádání architektury MIPS
Obr.3 Elektronkový počítač ENIAC
Obr.4 Mikroprocesor Intel 4004 (rok 1971)
Obr.5 Mikroprocesor Intel 8080 (rok 1974)
Obr.6 Registry mikroprocesoru MOS 6502
Obr.7 Registry mikroprocesoru Intel 8080
Obr.8 Rozdělení vybraných procesorů dle architektury CISC a RISC
Obr.9 Von Neumannova architektura
Obr.10 Harvardská architektura
Obr.11 Zřetězení instrukcí
Obr.12 Superskalární vykonávaní instrukcí
Obr.13 Dvoubitový prediktor programových skoků
Obr.14 Paměti HBM
Obr.15 Pohyb po kružnici
Obr.16 Schéma regulátoru
Obr.17 Integrační článek
Obr.18 Schématické znázornění FIR filtru
Obr.19 Schématické znázornění IIR filtru
Obr.20 Schématické znázornění funkce H- můstku MC33926
Obr.21 Osazená deska plošných spojů
Obr.22 Vývojové prostředí Keil µVision 5
Obr.23 Vývojový kitu LPC-Link 2 od firmy NXP
Obr.24 Aplikace Slot Car Tool
Obr.25 Závislost magnitudy na frekvenci navrženého filtru FIR
Obr.26 Filtrování hodnot z akcelerometru navrženým filtrem FIR
Obr.27 Průběh příkonu motorku v závislosti na odstředivém zrychlení

Tabulka 1 Vývojové generace počítačů
Tabulka 2 Registry v čipech PA-RISC
Tabulka 3 Porovnání výpočetního výkonu mikroprocesorů
Tabulka 4 Pracovní registry procesoru ARM
Tabulka 5 Časy potřebné na projetí tří kol dráhy dle použité metody řízení

Seznam použitých zkratek

ADC	analogově-digitální převodník
ARM	procesorová architektura (Advanced RISC Machine / Acorn RISC M.)
AVR	skupina mikročipů od firmy Atmel
BOR	Brownout Reset, způsob generování resetovacího signálu (podle napětí)
CAN	Controller Area Network, komunikační sběrnice
CISC	Complex Instruction Set Computer, procesory s rozsáhlou (komplexní) instrukční sadou
CNT	označení pro blok čítačů
DAC	digitálně-analogový převodník
DMA	Direct Memory Access, přímý přístup do paměti (periferie)
DC	Direct Current, stejnosměrný proud
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná (nevolatilní) paměť
EMC	elektromagnetická kompatibilita
FIR	finite impuls response - filtry s konečnou impulsní odezvou
FLASH	nevolatilní elektricky programovatelná paměť
FMC	Flash Memory Controller, kontrolér správy paměti FLASH
FPGA	Field Programmable Gate Array - Programovatelná hradlová pole
FPU	Floating-Point Unit, matematický koprocesor
GPIO	General-Purpose Input/Output, vstupně-výstupní registry (porty)
GPR	General Purpose Register (Obecně účelné registry – univerzální)
GPU	grafický procesor
I2C	Inter-Integrated Circuit, sériová sběrnice
I2S	Integrated Interchip Sound, synchronní sériové rozhraní (pro DAC atd.)
IDE	Integrated Development Environment (vývojové prostředí)
IIR	infinite impuls response - filtry s nekonečnou impulsní odezvou
IRQ	Interrupt ReQuest, správa žádostí o přerušení programu
ISA	Instruction Set Architecture (Architektura souboru instrukcí)
JTAG	Joint Test Action Group, standard pro programování a ladění CPU
JVM	Java Virtual Machine
KiB	Kibibajt, 1 KiB = 1024 bajtů (přesně)
LAN	Local Area Network, počítačová síť
LCD	Liquid Crystal Display, displej z tekutých krystalů
LED	Light-Emitting Diode, dioda emitující světlo
MIPS	a) Microprocessor without Interlocked Pipeline Stages, architektura CPU b) Million Instruction Per Second, milion instrukcí za sekundu
MMU	Memory management unit, jednotka správy paměti

MMX	MultiMedia eXtensions, multimediální technologie (instrukce)
MPU	Memory Protection Unit, ochrana dat v paměti
MNI	Non-Maskable Interrupt, nemaskovatelné přerušení
OLED	Organic Light-Emitting Diode, technologie organických elektroluminiscenčních diod
OSC	oscilátor, obvod oscilátoru
PATA	(Paralel) Advanced Technology Attachment, počítačová paralelní sběrnice pro připojení velkokapacitních paměťových zařízen
PI	Proporcionálně Integrační regulátor
PID	Proporcionálně Integračně Derivační regulátor
PIC	jednočipový mikropočítač firmy Microchip Technology
POR	Power On Reset, způsob generování resetovacího signálu
PWM	Pulse Width Modulation, pulzně šířková modulace
RISC	Reduced Instruction Set Computing, procesory s redukovanou instrukční sadou
RAM	Random-Access Memory, paměť s libovolným výběrem (adresací)
RTC	Real-time clock, hodiny reálného času
SATA	Serial ATA, počítačová sériová sběrnice pro připojení velkokapacitních paměťových zařízení
SD	Secure Digital, paměťová karta (digitální záznam)
SIMD	Single Instruction, Multiple Data
SoC	System On Chip (Systém na čipu - integrovaný obvod)
SPI	Serial Peripheral Interface, sériové periferní rozhraní
TMR	Timer, časovač (časovací obvody)
UART	Universal Synchronous / Asynchronous Receiver and Transmitter, Synchronní / asynchronní sériové rozhraní
USB	Universal Serial Bus, univerzální sériová sběrnice
VLIW	Very Long Instruction Word
WDT	Watchdog Timer, periferie, která resetuje systém při jeho zaseknutí

Úvod

Tato diplomová práce vznikla na základě podmětu od Technické univerzity v Liberci na inovaci elektronické řídící jednotky samořízeného modelu auta pro autodráhu, který je využívám pro výuku studentů v oblasti řízení a informatiky. Hlavním důvodem inovace byla nedostupnost a ukončení výroby některých klíčových součástek pro výrobu dosud používané jednotky. Dalším objektivním důvodem inovace bylo využití novějšího typu mikrokontroléru, který svými parametry umožní i složitější metody řízení.

Tato práce se ve své teoretické části zabývá problematikou mikrokontrolérů včetně historického vývoje, popisem základních funkčních částí mikrokontrolérů, rozborem procesorových architektur a moderních trendů v této rozsáhlé problematice. Pro ilustraci jsou uvedeny některé vybrané rodiny mikrokontrolérů a popsány jejich charakteristické vlastnosti.

Dále se zabývá problematikou návrhu elektronického zapojení, včetně teoretického rozboru výběru vhodných součástek a mikrokontroléru. V další své části se práce věnuje analýze současného stavu elektronického zapojení pro samořízený model auta. Velký důraz je zde kladen nejen na samotnou funkčnost, ale i na vhodnost použitého řešení.

Po tomto rozboru následuje samotná realizace osazené desky plošných spojů, její oživení, proměření a test funkčnosti. Následující část se věnuje tvorbě a popisu řídícího programu v mikrokontroléru včetně jeho otestování. V této části jsou navrženy a otestovány různé varianty řízení tohoto modelu.

1 Teoretický rozbor problematiky mikroprocesorů

Vývoj v oblasti návrhu a technologie výroby integrovaných obvodů umožnil široké použití mikroprocesorů a mikrokontrolérů. Vlivem konkurence nebo naopak i spolupráce několika různých výrobců čipů vzniklo postupně několik typů mikrokontrolérů pro obecná i úzce specifikovaná použití. Výběr vhodného čipu je tedy závislý nejen na jeho ceně a parametrech, ale také na dané platformě a dalších souvislostech.

1.1 Popis procesoru

Procesorem se rozumí základní jednotka počítače. Je to logický automat zpracovávající informace. Jeho základní částí je aritmetickologická jednotka a řadič. Chování procesoru je definováno programem.

Procesorové řezy jsou sestaveny z více čipů, které jsou specializované na konkrétní úkoly a funkce. Například části vykonávající aritmetické operace, řadič mikroprogramu (mikrokódu), řadič přerušení, vstupně-výstupní operace a podobně. Výhodou tohoto řešení je možnost pomocí obvodů definovat vlastní složitost, funkce i šířku zpracovávaných dat. Vzhledem k cenové náročnosti jednotlivých integrovaných obvodů, faktoru spolehlivosti, komplexní složitosti a fyzikálních omezení šíření signálu je dnes toto řešení odsunuto do pozadí.

Mikroprocesor je procesor vyráběný technologií velké integrace, který obsahuje logické, aritmetické a řídící jednotky v jednom pouzdře. Dnes s rozvojem integrace je integrován na jednom fyzickém médiu, tedy na jedné destičce.

Mikrokontrolér je z elektronického hlediska integrace mikroprocesorového jádra spolu s pamětí programu, pamětí dat, podpůrných obvodů a periferií do jednoho pouzdra, nebo dnes častěji na jedno fyzické medium (křemíkovou destičku). Lze konstatovat, že dnes se vyrábějí především mikrokontroléry integrované na jednu křemíkovou destičku, na které je celý mikrokontrolér včetně všech podpůrných obvodů a periferií.

Základní části procesoru jsou:

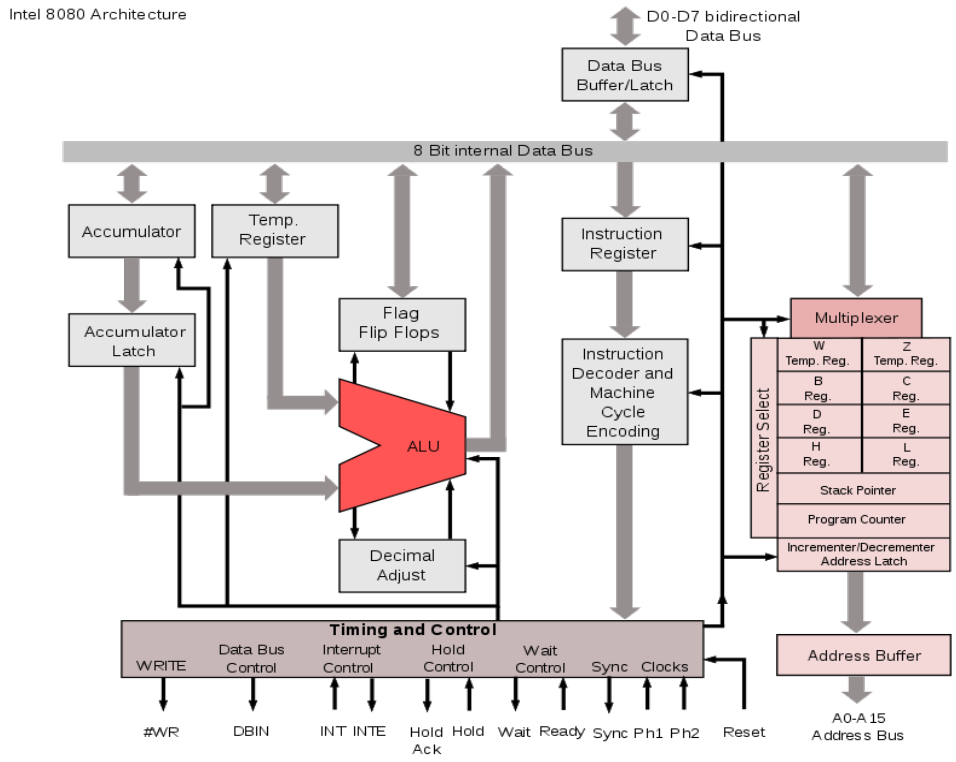
- registry
- aritmetickologická jednotka
- řadič
- vnitřní sběrnice

Registry, což jsou paměťové obvody uchovávající nejčastěji používaná data. Některé procesory rozlišují jejich specializaci. Např. procesory rodiny x86 rozlišují registry vhledem k jejich použití na akumulátor, čítač, registry pro data, bázové registry, stavový registr, ukazatele instrukcí a zásobníku nebo indexové registry. Vzhledem k problematice kompilace byla postupem času snaha modernizovat architekturu tak, aby registry byly univerzální (obecné). Výjimkou jsou čistě specifické registry, jako je stavový registr a ukazatel instrukcí.

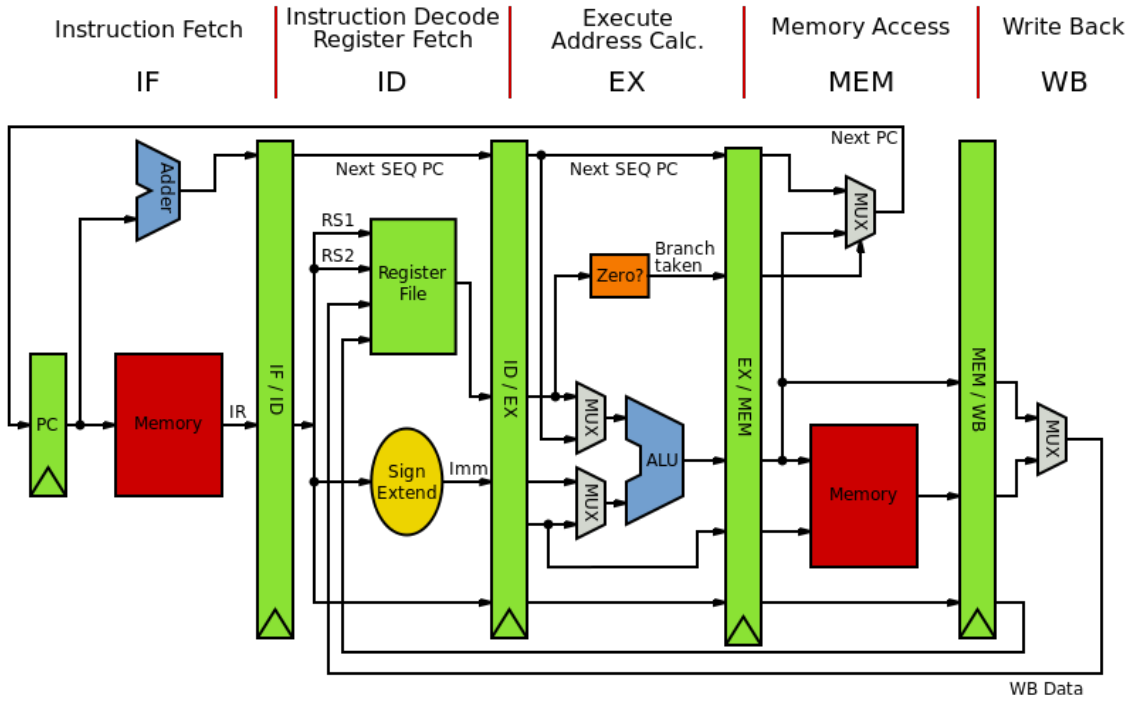
Aritmetickologická jednotka je souhrn obvodů vykonávající aritmetické i logické operace. Mezi aritmetické operace patří například operace sčítání, operace sčítání s přenosem do vyššího řádu, operace odečítání, operace odečítání s přenosem, porovnávání a podobně. Mezi operace logické patří bitový součet (značený OR), bitový součin (značený AND), nonekvivalence (značená XOR), bitové posuny, rotace a podobně. Tyto operace se provádějí nad registry, ale některé procesory je mohou provádět i nad oblastí v paměti. Aritmetickologická jednotka může být v některých procesorech obsažena vícekrát. V takovém případě můžou být některé jednotky specializované pro konkrétní sadu operací.

Řadič je obvod, který zajišťuje sled logických signálů pro provedení dané operace. Programový řadič obsahuje též dekodér instrukcí. Například operace sčítání vyžaduje načtení instrukčního slova, jeho dekódování, načtení operandů z registrů nebo paměti, provedení dané operace a uložení výsledku. Řadič může být proveden jako speciální sekvenční automat (čítač, dekodér, kombinační logické obvody), nebo jako mikroprogramový řadič využívající svou paměť pro uložení mikroprogramu. Tyto obě řešení se liší ve složitosti, ceně a rychlosti. Funkce řadiče může být také rozdělena do několika spolupracujících částí, které daný sled signálů vykovávají paralelně. Příkladem může být obvod generující signály pro operace s registry a zároveň jiný obvod může generovat signály pro operace s pamětí. Komplikovanější řadiče mohou řešit i paralelismus zpracování instrukcí. Řadiče nemusejí být určeny pouze pro vykonávaní programu, ale mohou být specializované i na podpůrné operace (řadič přerušení, řadič operací s pamětí či vstupně výstupních obvodů, řadiče jednotlivých periferií).

Vnitřní sběrnice zajišťuje spojení mezi jednotlivými bloky procesoru. Sběrnice lze rozdělit podle jejich typu na datovou, adresovou a řídicí. Jejich šířka je většinou pro každý typ procesoru odlišná. Stejně tak i jejich počet, protože mohou být například zdvojené. K dalším částem jádra procesoru mohu patřit i další podpůrné obvody, jejichž cílem je minimalizovat dobu potřebnou na vykonání operace. Mezi ně patří obvody zajišťující rychlejší přísun dat a instrukcí (cache), obvody pro rychlejší zpracování dat (např. hardwarové násobičky, koprocesory a akcelerátory), obvody pro paralelismus operací s pamětí a registry, či paralelismus vykonávání jednotlivých operací. K tomuto tématu bude věnována samostatná kapitola.



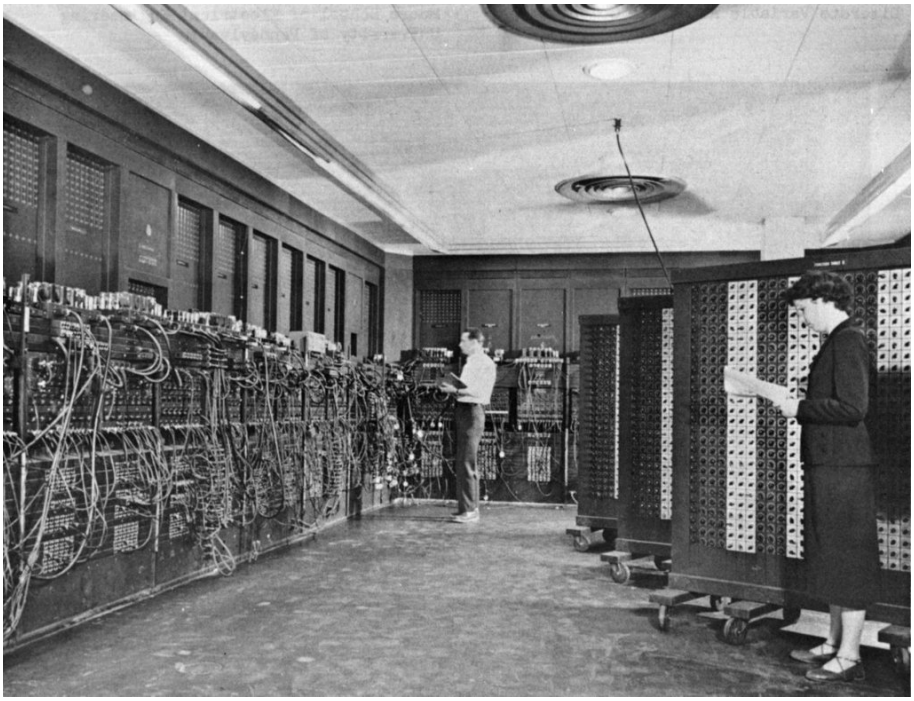
Obr.1 Vnitřní uspořádání procesoru Intel 8080



Obr.2 Vnitřní uspořádání architektury MIPS

1.2 Stručná historie procesoru

Původně mechanické automaty byly úspěšně nahrazeny elektrickými obvody s relátky a později elektronkami. Velké nároky na energii a prostor pomohl vyřešit objev polovodiče, který zároveň umožnil zvýšení operačního výkonu.



Obr.3 Elektronkový počítač ENIAC

Parametry počítače ENIAC:

Zahájení vývoje	1943
Rok dokončení	1946
Počet elektronek	17 468
Počet krystalových diod	7200
Počet relé	1500
Počet rezistorů	70 000
Počet kondenzátorů	10 000
Počet ručně pájených spojů	přibližně 5 miliónů
Hmotnost	30 tun
Prostorová náročnost	63 m^3
Energetická náročnost	150 kW
Pracovní frekvence	100 kHz

Vývoj počítačů lze rozdělit do generací, které popisuje následující tabulka:

Generace	0	1	2	3	4
Rok	1940	1951	1957	1964	1981
Prvky	relé	elektronky	tranzistory	SSI	LSI
Hlavní paměť		buben	ferity	ferity	LSI
Kapacita paměti		1 KB	10 KB	1 MB	10 MB
MIPS	0,001	0,01	0,1	1	10
Příklad	Mark I	Univac 1	IBM 7090	IBM 360	Intel 4004

Tabulka 1 Vývojové generace počítačů

Vývoj technologie výroby polovodičů vedl k postupné integraci více tranzistorů na jedné křemíkové destičce. To umožnilo vznik integrovaných obvodů. Zprvu se jednalo o jednoduché kombinační a sekvenční logické obvody (logická hradla, klopné obvody, multiplexory).

Později vznikaly složitější obvody se specifickou funkcí. Mezi takové obvody patřily **procesorové řezy**, které tvořily sadu několika integrovaných obvodů, z nichž bylo možné sestavit plnohodnotný procesor. Sadu obvodů tvořil například mikroprogramový řadič, aritmeticko-logická jednotka, obvod pro výpočet příznaků přenosu, obvod pro řízení přerušení, řídicí obvod pro sběrnici, registry, paměť ROM a později i RAM. Bylo tedy možné vytvořit více či méně složité počítače s libovolnou datovou šířkou. Mezi zástupce procesorových řezů patřily řady Intel 3000 a AMD Am2900.

V roce 1971 vytvořila firma Intel první mikroprocesor a později si jej nechala patentovat. Mikroprocesor Intel 4004 byl vyvinut pro kalkulačky a tedy jeho 4- bitová šířka slova byla pro tento účel dostačující. Pro svoji funkci využíval až 20 podpůrných obvodů.

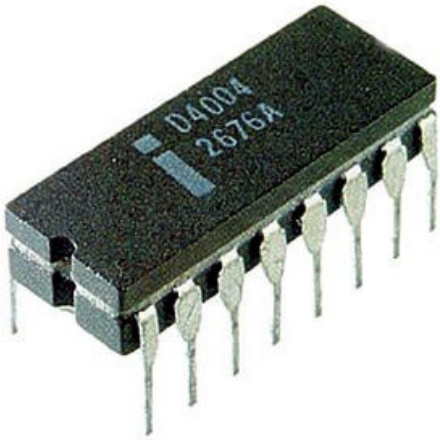
V roce 1974 byl vyvinul dnes legendární 8- bitový mikroprocesor Intel 8080 (slangově nazýván "BOBO"). Jeho adresní prostor 64 KiB (společně pro data i instrukce) byl na svou dobu dostačující. Nebyl však úplně monolitický, k jeho činnosti byly nutné alespoň dva pomocné obvody (generátor hodin a řadič sběrnice). Pro vytvoření mikropočítače byla potřeba celá sada nejméně 6 obvodů. I přes nutnost použít podpůrné obvody a 3 napájecí napětí (-5V, +5V a +12V) byl tento mikroprocesor velmi populární. Vyráběl se v licenci v mnoha jiných firmách, například u nás v podniku Tesla pod označením MHB 8080.

Nedlouho po té vzniklo několik konkurenčních 8- bitový procesorů od jiných firem, z nichž nejznámější jsou Motorola MC6800, MOS 6502 a Zilog Z80. Posledně

jmenovaný je zdola kompatibilní s Intel 8080, rozšířený o nové registry, instrukce a pokročilý mechanismus přerušení, kde každá periferie z jeho rodiny obvodů vysílá svůj vektor přerušení. Tyto procesory již byly napájeny jediným napětím +5V, nevyžadovaly podpůrné obvody a proto se velmi rozšířily. Několik jiných firem vyrábělo jejich licencované klony nebo použily vlastní rozšíření jejich jádra. Na jejich základě též vzniklo několik 8- bitový počítačů, například Apple II, Atari, Commodore 64 nebo Sinclair ZX Spectrum.

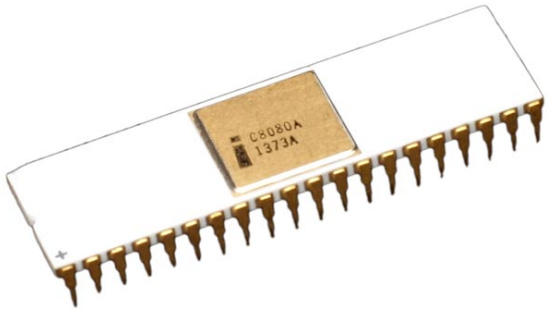
Poznámka:

Jedním z několika nástupců Intel 8080 byl 16- bitový procesor Intel 8086, který je znám z oblasti domácích počítačů a s nímž jsou dnešní osobní počítače skupiny značené PC zpětně kompatibilní.



Počet tranzistorů	2300
Maximální frekvence	750 kHz
Napájecí napětí	15 V
Technologie	PMOSFET
Výrobní proces	10 µm
Paměť programu	4 KiB
Paměť dat:	640 B
Výpočetní výkon	0,07 MIP

Obr.4 Mikroprocesor Intel 4004 (rok 1971)



Počet tranzistorů	4500
Maximální frekvence	2 MHz
Napájecí napětí	5 V
Technologie	NMOS
Výrobní proces	4,5 µm
Adresní prostor	64 KiB
Výpočetní výkon	0,64 MIPS

Obr.5 Mikroprocesor Intel 8080 (rok 1974)

Ukázka kódu v assembleru funkce kopírování dat pro procesor **Intel 8080:**

```
                ;       BC - Number of bytes to copy
                ;       DE - Address of source data block
                ;       HL - Address of target data block

memcpy      public
loop        mov     a,b         ;Test BC,
            ora     c           ;If BC = 0,
            rz                  ;Return
            ldax    d           ;Load A from (DE)
            mov     m,a         ;Store A into (HL)
            inx     d           ;Increment DE
            inx     h           ;Increment HL
            dcx     b           ;Decrement BC
            jmp     loop        ;Repeat the loop
```

Ukázka kódu v assembleru funkce kopírování dat pro procesor **MC6800:**

```
;       cnt - Number of bytes to copy
;       src - Address of source data block
;       dst - Address of target data block

memcpy      public
            ldab    cnt+1       ;Set B = cnt.L
            beq     check       ;If cnt.L=0, goto check
loop        ldx     src         ;Set IX = src
            ldaa    ix          ;Load A from (src)
            inx                 ;Set src = src+1
            stx     src
            ldx     dst         ;Set IX = dst
            staa    ix          ;Store A to (dst)
            inx                 ;Set dst = dst+1
            stx     dst
            decb                ;Decr B
            bne     loop        ;Repeat the loop
            stab    cnt+1       ;Set cnt.L = 0
check       tst     cnt+0       ;If cnt.H=0,
            beq     done        ;Then quit
            dec     cnt+0       ;Decr cnt.H
            decb                ;Decr B
            bra     loop        ;Repeat the loop
done        rts                 ;Return
```

Procesor **Z80** řeší stejnou funkci kopírování dat jednou instrukcí **LDIR**.

S možností integrovat stále více tranzistorů na jeden čip bylo možno postupně integrovat procesorové jádro spolu s periferiemi a pamětí do jednoho pouzdra integrovaného obvodu. Takový integrovaný obvod nazýváme mikrokontrolér. Uplatnění našel především v jednoúčelových zařízeních. Vzhledem k jeho určení došlo ke specializaci pro různá konkrétní využití a zároveň o aplikační univerzálnost.

Mezi dnes známé rodiny mikrokontrolérů patří:

- 8051 (Intel, Philips, Atmel atd...)
- PIC (Microchip)
- AVR (Atmel)
- Z8 (Zilog)
- MSP430 (Texas Instruments)
- HC(S)12 (Freescale dříve Motorola)
- ARM (vývoj Acorn, výroba NXP, Freescale atd...)

1.3 Dělení procesorů podle jejich vlastností

Procesory lze rozčlenit podle několika faktorů do skupin, které se mohou vzájemně prolínat a sdílet některé společné vlastnosti.

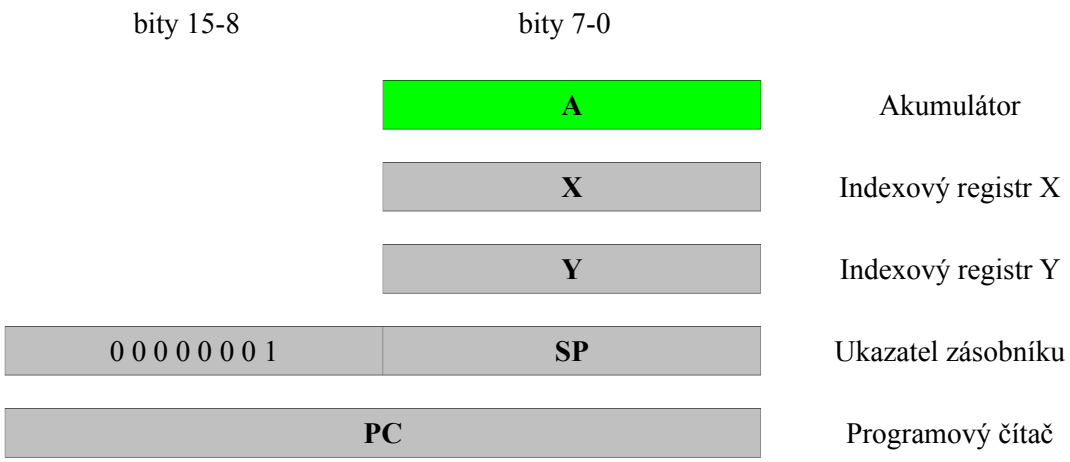
Dělení podle platformy a oblasti použití:

- Osobní počítače. Zde je kladen důraz na výpočetní výkon a na kompatibilitu.
- Pracovní stanice. Zde je také kladen důraz výpočetní výkon. Některé pracovní stanice obsahují také akcelerátory výpočtů či další specializované obvody.
- Servery (datové, komunikační, tiskové, databázové). U serverů je vyžadována schopnost dobrého paralelismu zpracovávaných procesů. V poslední době z důvodu chlazení celého systému je také kladen důraz na úsporu energie.
- Mobilní počítače (notebooky, palmtopy, handheldy, pocket PC, PDA) V této oblasti je kladen důraz spíše na poměr mezi spotřebou a výkonem, tedy na době provozu z akumulátoru.
- Grafické procesory a akcelerátory lze rozdělit na integrované do procesoru a na dedikované v podobě grafické karty. Podle druhu použití je lze dále rozdělit na profesionální použití (např. pro CAD/CAM aplikace, fyzikální výpočty), herní zaměření, nebo pro obecné použití v akceleraci moderních aplikací.
- Embedded aplikace (vestavěné). Zde není vyžadována vzájemná kompatibilita, protože daná aplikace má své konkrétní využití v konkrétním zařízení.
- Automobilová, průmyslová, telekomunikační, vojenská a zdravotnická technika. Zde je kladen důraz především na spolehlivost, odolnost vůči rušivým vlivům (EMC), na teplotní rozsahy a určené prostředí.

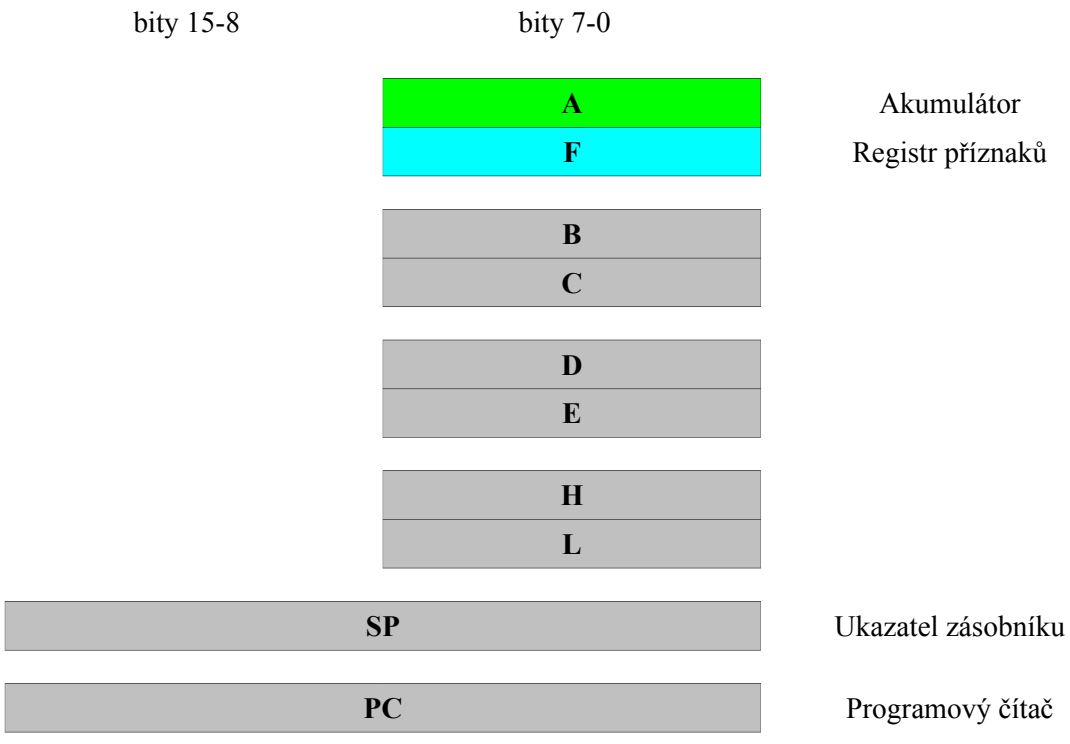
Dělení podle architektury souboru instrukcí (ISA):

- Střadačové (akumulátorové) procesory
- Zásobníkové procesory
- Procesory s univerzálními registry

Střadačové (akumulátorové) procesory provádějí většinu aritmetickologických operací s jedním implicitním registrem (operandem), který uchovává hodnotu předchozí operace. Tento registr se proto nazývá střadač (akumulátor). Toto řešení umožňuje jednodušší a tedy levnější implementaci i zkrácení instrukčního slova, protože není nutno do instrukčního slova zakódovat index tohoto implicitního registru. Toto řešení ale naopak zhoršuje možnost paralelního zpracování instrukcí. Mezi typické představitele patří procesory Intel 8080, Zilog Z80, Motorola MC6800, MOS 6502, 8051, procesory rodiny PIC.



Obr.6 Registry mikroprocesoru MOS 6502



Ze dvou 8-bitových registrů lze vytvořit 16-bitový (AF, BC, DE, HL)
Složené registry BC, DE a HL lze použít pro adresaci v paměti.

Obr.7 Registry mikroprocesoru Intel 8080

Zásobníkové procesory (stack processors) provádějí veškeré operace s operandy uloženými na vrcholu zásobníku. Například součet dvou operandů vyžaduje uložení obou operandů na vrchol zásobníku, kde následně operace součtu oba operandy z vrcholu zásobníku odebere, sečte a výsledek je opět uložen na vrchol zásobníku. Výsledek může sloužit jako operand pro následující operaci, nebo je ze zásobníku odebrán. Procesor má několik prvků vrcholu zásobníku umístěné v lokálních registrech. Využívá většinou pouze dvoufázovou instrukční pipeline. V první fázi načte instrukční slovo a současně provede operaci u předešlé instrukce. Většina zásobníkových procesorů používá horizontální instrukční formát. U zásobníkových procesorů nedochází k hazardům při vykonávání instrukcí. Jednoduchost a rychlost těchto procesorů je vykoupena horší možností implementace jazyka C, protože struktura tohoto jazyka vyžaduje náhodný přístup k operandům ve funkcích. Zásobníkové procesory nejsou příliš rozšířené, ale pro některé aplikace má tento procesor své výhody. Mezi představitele této skupiny patří procesory HARRIS RTX 2000, MuP21 a F21. Oba posledně jmenované procesory mají šířku instrukčního slova 20 bitů, kde do jednoho dvacetibitového slova je možné uložit až čtyři instrukce.

Procesory s univerzálními registry jsou dnes nejpoužívanější procesory. Pro operace je použit soubor velmi rychlých univerzálních registrů (GPR), které mohou být v instrukci jak zdrojem tak i cílem. Základní myšlenka se opírá o fakt, že program většinou pracuje v jeden čas pouze s omezeným počtem operandů a nejrychleji přístupná data jsou právě v registrech. Proto těchto registrů obvykle bývá větší počet (většinou od 8 až do 128). Ovšem tento počet je vykoupený delším instrukčním slovem, do kterého je nutno indexy použitých registrů zaznamenat. Jistým kompromisem může být použití registrového okna, kdy je v jeden čas možné použít jen část registrů současně (např 32 registrů ze 128), nebo pro aktuální proces možnost adresovat pouze nižší počet registrů než je celkem. Možnost náhodného přístupu do registrů a jejich větší počet umožňuje lepší využití paralelního zpracování instrukcí a také usnadňuje kompilaci zdrojového kódu. Taktéž je možné v jedné instrukci číst dva operandy a výsledek uložit do třetího, kde se některé registry mohou shodovat. Mezi nejznámější zástupce těchto procesorů patří procesory skupiny ARM, mikroprocesory řady MSP430, PIC32, řady MIPS, řady SPARC, řady PA-RISC, Itanium, PowerPC a procesory PC v režimu x86-64 (EM64T / AMD64). Pro ilustraci jsou registry procesoru postaveného na čipech PA-RISC v následující tabulce.

Registry	Význam
GR 0 až GR 31	univerzální pracovní registry
SHR 0 až SHR 6	stínové registry používané při přerušení
SR 0 až SR 7	registry používané při skocích (space registers)
PSW	Registr stavových příznaků (Processor Status Word)
CR 0 až CR 31	řídicí registry
FPR 0 až FPR 31	registry používané matematickým koprocesorem

Tabulka 2 Registry v čipech PA-RISC

Z hlediska instrukční sady lze dělit procesory:

- CISC
- RISC
- Post-RISC

CISC (Complex Instruction Set Computer) procesory jsou charakteristické velmi „košatou“ instrukční sadou strojových instrukcí (mnohdy do značné míry ortogonální) a poměrně velkým počtem způsobů adresace operandu. Instrukce mají proměnlivou délku (často od jednoho bajtu až jednotky bajtů). Proměnlivý je i počet taktů potřebných na vykonání jedné operace (i desítky taktů). Toto vyžaduje použití složitého řadiče a celková složitost procesoru roste. Díky vysoké složitosti byl řadič navržen na principu paměti s mikroprogramy. Řídicí obvody mohou zabírat na čipu přibližně 60% místa. Snaha některých návrhářů procesorů dokonce byla, aby procesor měl instrukce a módy adresace podobné až shodné s instrukcemi některých programovacích jazyků. Například assembler procesoru Motorola 68000 se do značné míry přibližuje jazyku C. Procesor většinou obsahuje relativně nízký počet registrů. Pro některé registry bývají vyhrazeny některé specifické funkce (např. čítač, data, indexovací registr). Mezi zástupce těchto procesorů patří Intel 8080, Zilog Z80, Motorola MC6800, MOS 6502, procesory rodiny x86 (Intel / AMD / VIA Technologies / Cyrix) 8086, 80286, 80386 atd.

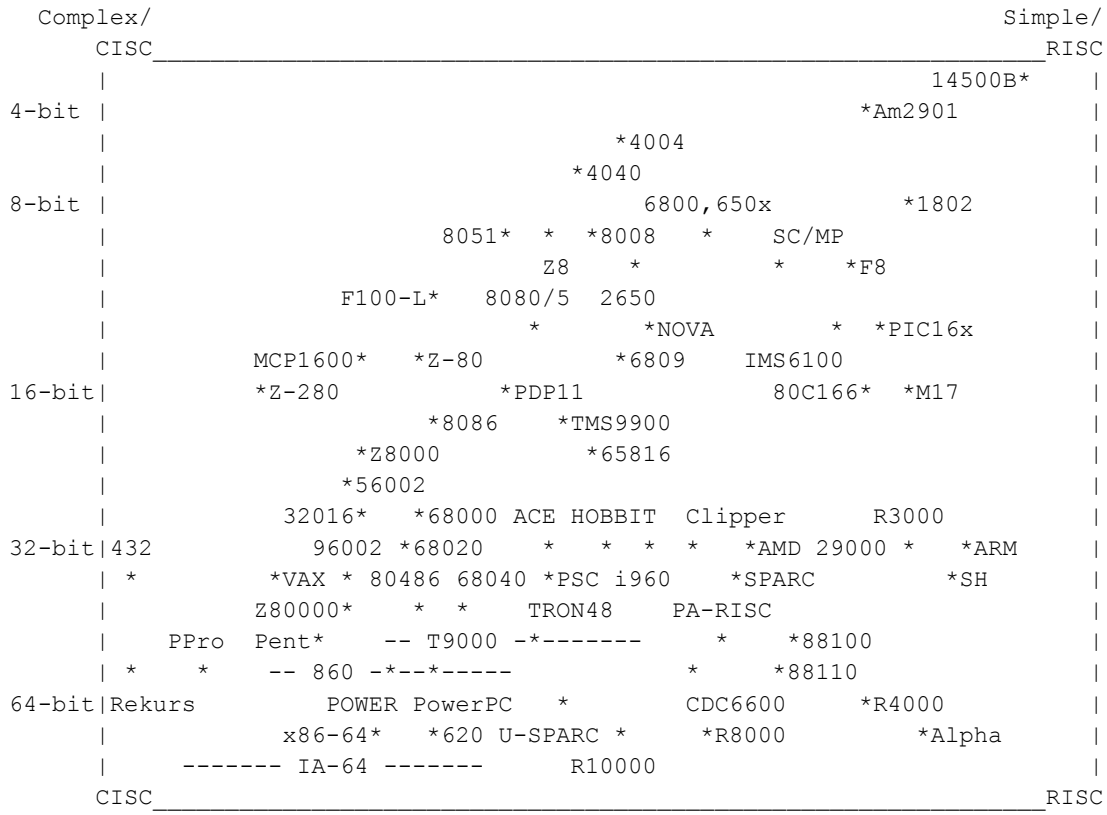
Ukázka kódu CISC procesoru rodiny x86**:**

```
81 EC 88 00 00 00              sub     esp,88h
56                             push    esi
8B B4 24 90 00 00 00           mov     esi,dword ptr [esp+90h]
8D 44 24 0C                    lea     eax,dword ptr [esp+0Ch]
56                             push    esi
E8 77 08 00 00                 call    4018F0
83 C4 08                       add     esp,8
8D 44 24 04                    lea     eax,dword ptr [esp+4]
8D 8C 24 94 00 00 00           lea     ecx,dword ptr [esp+94h]
```

RISC (Reduced Instruction Set Computer) procesory vycházejí ze zjištění, že větší část programu využívá jen určitou část instrukčního souboru a zbylé méně používané instrukce lze často nahradit jednoduššími instrukcemi. (V některých případech se využívalo jen 30 % všech možných kombinací.) V roce 1980 firma IBM a lidé z University of Berkeley přišli s revoluční myšlenkou vytvořit procesor s jednoduchými instrukcemi s výhodnější ortogonalitou. (Řídicí obvody zabírají pouze 6–10 % místa) Prostor, jenž u procesorů CISC zabíral složitý řadič, byl využit pro větší počet univerzálních registrů (většinou 32-bitových). To umožnilo snížit počet operací čtení a zápisu operandů do paměti. Zjednodušil se též dekodér instrukcí a bylo implementováno proudové zpracování instrukcí. Jednodušší instrukce (většinou stejné délky) je možné lépe zpracovat paralelně. Přístup do paměti je pouze pomocí instrukcí přesunu (LOAD / STRORE). Nejčastěji se zde uplatňuje harvardská architektura. Mezi velmi úspěšné RISC procesory patří celá rodina mikroprocesorů označovaná MIPS. Mezi zástupci RISC procesorů jsou procesory skupiny ARM, mikroprocesory řady MSP430, procesory řady MIPS, řady SPARC, řady PA-RISC atd.

Post-RISC procesory v sobě zahrnují kombinaci instrukčních sad CISC a RISC. Někdy jsou označovány jako CRISC (Complex RISC). Problém obtížného paralelního vykonávání instrukcí je u moderních procesorů řešen převodem CISC instrukce na mikroinstrukce, které lze zpracovat paralelně a jednodušším způsobem. To ovšem vyžaduje komplikovanější dekodéry instrukčního toku, což se může projevit i většími nároky na spotřebu energie. Proto některé moderní procesory ukládají přeložené instrukce do speciálně vyhrazené dočasné paměti (cache L0). Pro zvýšení výkonu je použito proudové zpracování mikroinstrukcí a spekulativní provádění instrukcí. Do této kategorie lze zařadit většinu moderních PC procesorů, kde se většinou zachovává zpětná kompatibilita.

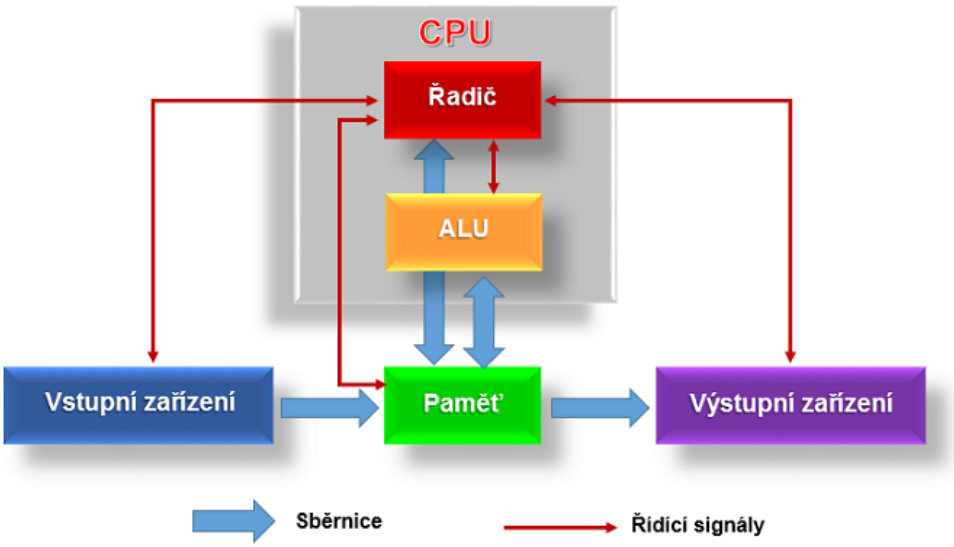
Vhledem k mnoha druhům procesorů a jejich vlastnímu řešení zpracování instrukčního toku však mnohdy nelze jednoznačně určit do jaké skupiny daný procesor patří. Tuto skutečnost ukazuje následují graf.



Obr.8 Rozdělení vybraných procesorů dle architektury CISC a RISC

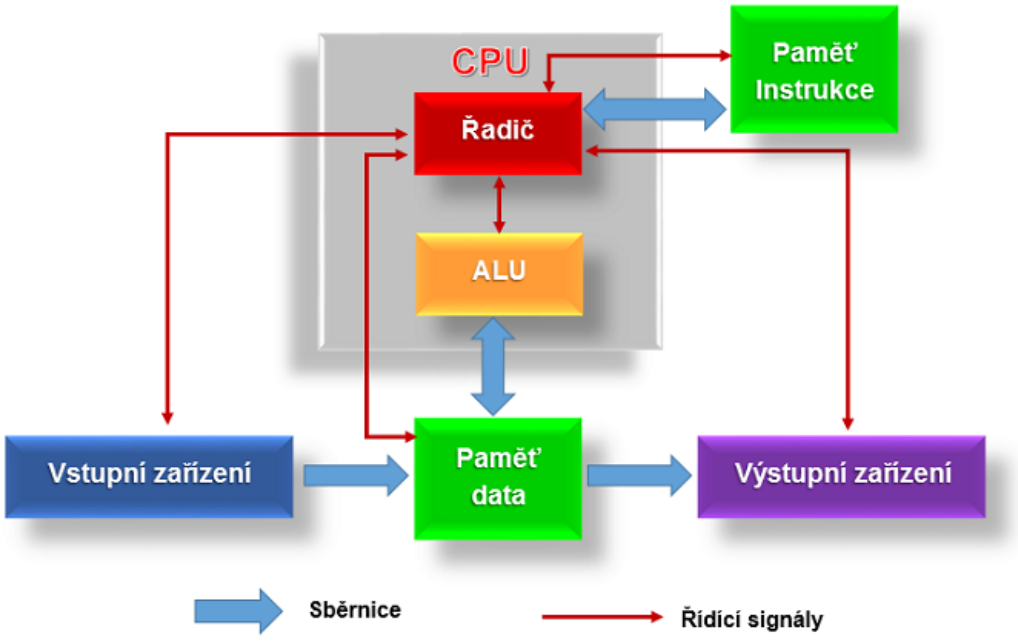
Z hlediska přístupu do paměti lze dělit procesory podle architektury:

- Von Neumannova architektura byla představena v roce 1946. Popisuje počítač se společnou pamětí pro data a instrukce. Zpracování operací je sekvenční. Procesor počítače je složen z řídící a výkonné aritmetickologické jednotky. Je programován obsahem paměti, která je dělena do buněk stejné velikosti. Každá buňka je adresována svým pořadím. Řídící jednotka zpracovává postupně jednotlivé instrukce uložené v paměti. Vstup a výstup dat zajišťují vstupní a výstupní jednotky. Jako nevýhoda se uvádí možnost mylně interpretovat data jako program.



Obr.9 Von Neumannova architektura

- Harvardská architektura vznikla v roce 1943 (koncepce IBM Harvard MARK1). Fyzicky odděluje paměť programu a paměť dat (oddělené sběrnice). Je odděleno řízení procesoru od řízení vstupních a výstupních jednotek. Tato architektura umožňuje rozdílnou bitovou šířku paměti pro program a paměti pro data (rozdílná organizace pamětí). Dvojí paměť umožňuje paralelní přístup k oběma pamětem současně. Oddělení paměti pro program a data nejen zvyšuje rychlost zpracování, ale přispívá i k bezpečnosti systému. Do této skupiny patří například rodina PIC 10 - PIC 18.



Obr.10 Harvardská architektura

- Modifikovaná Harvardská architektura v sobě kombinuje funkce architektury Harvardské a von Neumannovy. Paměť dat a paměť programu je i nadále oddělena, ale architektura umožňuje přenos dat mezi rozdělenými paměťovými oblastmi. Tuto architekturu využívají některé signálové procesory.

Z hlediska šířky zpracovávaných dat lze dělit procesory:

- 4-bitové procesory byly mezi prvními procesory (Intel 4004). Uplatnění nacházely v jednoduchých systémech jako například jednoduché kalkulačky nebo hodinky.

- 8-bitové procesory se i dnes objevují v různých aplikacích, kde jednoduchost těchto procesorů určitým aplikacím zcela vyhovuje. Patří sem procesory řady PIC 10 - PIC 18, klony 8051 a dříve rozšířené procesory Intel 8080, Zilog Z80, Motorola MC6800, MOS 6502, atd.

- 16-bitové procesory se dnes používají v aplikacích, kde 16 bitů často stačí na adresaci paměti. Mezi tyto procesory patří rodina MSP430, která patří k procesorům s nízkou spotřebou. Zástupcem procesorů z řady PC jsou procesory 8086, 80186 a 80286.

- 32-bitové procesory jsou dnes značně rozšířené. Důvodem jejich použití je výhodnost zpracování operandů o velikosti 32 bitů a snadnější možnost adresovat více paměti než 64 KiB. U mnohých mikrokontrolérů je možné adresovat integrované periferie podobným způsobem jako paměť RAM i ROM. Mezi zástupce těchto procesorů lze uvést 32-bitové procesory AMR Cortex-M, PIC32, Intel 80386/80486 a následníky procesoru Pentium.

- 64-bitové procesory vznikly také na základě nutnosti snadno adresovat více jak 4GiB paměti. U počítačů PC se tak nahradil dřívější komplikovanější způsob adresace nad tuto hranici omezenou bitovou šířkou registrů. Mezi zástupce těchto procesorů lze uvést 64-bitové procesory AMR Cortex-A57, AMD Athlon 64, Intel Core, VIA Nano a novější procesory.

Matematické koprocesory mohou používat i větší bitovou šířku, např. 80 nebo 128 bitů. Bitová šířka dat nad 128 bitů se dnes používá spíše pro zpracování vektorového počtu, kde daný počet bitů se rozpadá na vektor, který je zpracován paralelně (např. 4x32 bitů). Příkladem mohou být operace s vektory a operace využívající paralelního zpracování dat jednou instrukcí (SIMD).

1.4 Obvody a technologie zvyšující výpočetní výkon

Hardwarová násobička (resp. dělička) urychluje aritmetické operace násobení (resp. dělení). Tato operace lze samozřejmě vyřešit posloupností instrukcí, ale často znatelně pomaleji. Některé procesory pro násobení využívají specializované instrukce, kde se během této instrukce provede zároveň podmíněný součet a bitový posun. Počet těchto instrukcí za sebou pak určuje velikost výsledku (například 8-bitové násobení vyžaduje 8 těchto instrukcí). Násobení může výrazně urychlit specializovaný obvod. Výpočet může být postupný, kde se provádějí součty postupně, zcela řešen kombinační logikou, nebo může být řešen kombinací obou postupů. Výhodou hardwarových násobiček řešených čistě kombinační logikou je jejich rychlost, protože mohou provést výpočet i během jednoho taktu, ale vyžadují výrazně větší množství použitých tranzistorů.

Koprocesor je specializovaný procesor, který urychluje některé specifické výpočty a který může rozšířit funkce primárního procesoru, kterým je řízen. Nejznámější koprocesor je matematický koprocesor (FPU), který provádí aritmetické operace s čísly s pohyblivou desetinnou čárkou (např. sčítání, odečítání, násobení, dělení, mocna, odmocnina, goniometrické funkce, logaritmy atd.). Tyto výpočty mohou být vykonány paralelně s činností procesoru. Dnes je často používán i grafický koprocesor (GPU), který urychluje grafické operace. Může být součástí procesoru, nebo jako samostatný čip spolu s pamětí na grafické kartě. Dříve jeho funkcí bylo především provádět operace s 2D objekty, později komplikovanější operace při vykreslování 3D objektů, jako například vykreslení textur podle parametrů osvětlení objektu. Moderní grafické akcelerátory obsahují i velké množství výpočetních jednotek, které znatelně urychlují výpočty 3D prostoru. Schopnost moderních akcelerátorů paralelně počítat i s čísly s pohyblivou desetinnou čárkou je dnes využívána i pro náročné výpočty (například technické a fyzikální výpočty). Jedním z parametrů určující výpočetní výkon může být i množství samostatných výpočetních jednotek, které se dnes pohybuje i do několik tisíců v jenom čipu. Grafické akcelerátory dnes umožňují i akceleraci přehrávání videa a zpracování zvukových signálů. Do kategorie koprocesorů patří i koprocesory určené pro kryptografické operace.

Dalším způsobem jak zvýšit výpočetní výkon může být převedení některých operací na periferie a podpůrné obvody. DMA řadič paměti může určitou operaci provést rychleji a jeho činnost může být prováděna paralelně s činností procesoru. Těchto nezávislých řadičů může být více a je též možné jejich činnost spojit s konkrétními periferiemi. Například obvody zajišťující komunikaci (USB, UART, SPI, LAN, CAN) mohou data číst nebo ukládat přímo do paměti k pozdějšímu zpracování

bez nutnosti přímé obsluhy procesorem. Samotné periferní obvody mohou obsahovat i vlastní vyrovnávací paměť nebo jejich činnost může být naprogramována. Příkladem může být periferie komunikace CAN, kde je možné definovat hardwarovou filtraci příchozích zpráv, které se automaticky ukládají na konkrétní místo v paměti. Obdobným příkladem mohou být vícekanálové AD a DA převodníky, které zpracovaná data mohou automaticky číst a ukládat do paměti, nebo předat jiné periferii ke zpracování. Akceleraci výpočtů a různých operací mohou též zajistit i Programovatelná hradlová pole (FPGA - Field Programmable Gate Array) nebo zakázkové obvody.

Taktéž samotné procesory (či jejich výpočetní jádra) používají pro zvýšení výpočetního výkonu několik strategií a technologií. Subskalární (sekvenční) procesory vykonávají každou instrukci v několika hodinových taktech (např. Intel 8080, Zilog Z80). Skalární procesory využívají zřetězení vykonávání instrukcí. To umožnilo vykonání jedné instrukce za dobu jednoho hodinového taktu. Jedna samotná instrukce sice stále vyžaduje na své zpracování více fází, ale během každé fáze se též vykonávají fáze následujících instrukcí a tedy je možné s každým hodinovým taktem dokončit jednu instrukci.

1	**IF**	**ID**	**EX**	**MEM**	**WB**		
2		**IF**	**ID**	**EX**	**MEM**	**WB**	
3			**IF**	**ID**	**EX**	**MEM**	**WB**
4				**IF**	**ID**	**EX**	**MEM**
5					**IF**	**ID**	**EX**

IF (Instruction Fetch) – načtení instrukce
ID (Instruction Decode) – dekódování instrukce
EX (Execute) - vykonání
MEM (Memory access) - přístup do paměti
WB (Register write back) – zápis výsledku

Obr.11 Zřetězení instrukcí

Každou fázi vykonávají určité obvody a tedy je plné využití těchto obvodů. Rozdělení do více fází je též výhodné z hlediska hodinového kmitočtu a tedy takové procesory mohou využívat větší frekvence hodinového taktu. Toto řešení ovšem musí ošetřit hazardy, kdy kterákoliv rozpracovaná instrukce nemůže v některé své fázi ihned pokračovat. To může způsobit například čekání na výsledek předchozí operace nebo čtení z paměti. V takovém případě musí procesor zpracování uzpůsobit a například ke zpracování vložit prázdné instrukce. Do jisté míry lze těmto některým případů předejít již při kompilaci vhodným seřazením instrukcí. Superskalární procesory využívají

zřetězení a paralelní zpracovávání instrukcí. Mohou tedy za jeden takt dokončit více než jednu instrukci.

1	IF 1	IF 2	ID 1	ID 2	EX 1	EX 2	M 1	M 2	WB 1	WB 2
2			IF 1	IF 2	ID 1	ID 2	EX 1	EX 2	M 1	M 2
3					IF 1	IF 2	ID 1	ID 2	EX 1	EX 2
4							IF 1	IF 2	ID 1	ID 2
5									IF 1	IF 2

IF (Instruction Fetch) – načtení instrukce
ID (Instruction Decode) – dekódování instrukce
EX (Execute) - vykonání
M (Memory access) - přístup do paměti
WB (Register write back) – zápis výsledku

Obr.12 Superskalární vykonávaní instrukcí

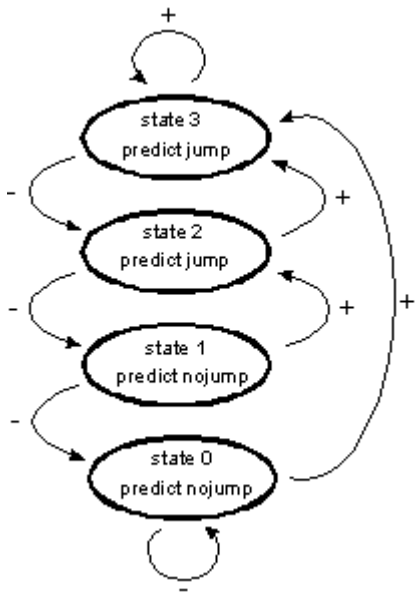
U těchto procesorů mají podstatný vliv na výpočetní výkon instrukce skoku, protože je nutné nedokončené instrukce „zahodit“ a načíst nové instrukce. U procesorů, které z důvodu vyšších frekvencí zpracovávají instrukce ve větším počtu fází (třeba i 20) je výpadek zpracování o to znatelnější.

Jedna z metod jak toto částečně řešit je vložení podmínky vykonání do samotné instrukce. Míra větvení programu se tak zmenšuje. Tuto vlastnost využívají například procesory ARM.

Další možností je existence takzvaných „branch delay slotů“. Ta zajišťuje, že se dokončí a provede i instrukce bezprostředně následující za skokem. Z provedeného měření vyplývá, že až 70% „branch delay slotů“ bylo možné zaplnit plnohodnotnou instrukcí. Tuto vlastnost využívají některé procesory RISC.

Velmi úspěšnou metodou jsou prediktory skoků. Mnohdy lze dopředu odhadnout, zda se skok provede a na základě tohoto odhadu se začnou do instrukční pipeline vkládat buď instrukce z jedné nebo druhé větve programu. Odhad se provádí na základě stavů bitových ukazatelů k příslušné instrukci, kde se stavy mění podle toho zda byl skok proveden nebo nikoliv. Úspěšnost odhadu velmi ovlivňuje celkový výkon procesoru.

Na softwarové úrovní lze výkon ovlivnit především vhodným uspořádáním instrukcí. Velmi používanou metodou je tak zvané rozbalování cyklů, kde se tělo cyklu několikrát ve výsledném kódu zopakuje a tedy negativní vliv skoků se sníží.



Obr.13 Dvoubitový prediktor programových skoků

Architektura VLIW (Very Long Instruction Word) seskupuje několik instrukcí do jednoho dlouhého instrukčního slova. Toto řešení zjednodušuje paralelní dekódování instrukcí, ale klade větší důraz na kompilátor, který musí řešit vzájemné hazardy a posloupnost instrukcí. Tuto architekturu mají procesory Itanium a většina dnešních grafických čipů.

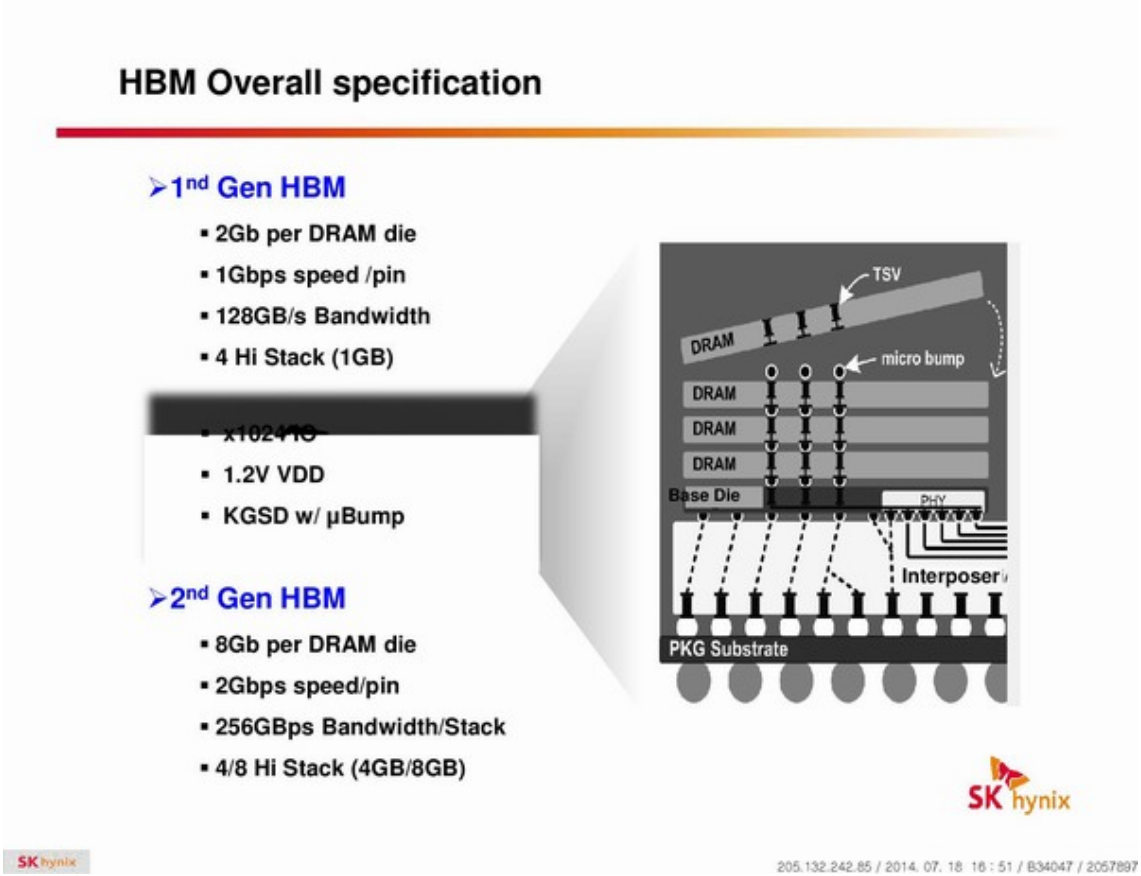
Instrukce SIMD (Single Instruction Multiple Data) provádějí operace s daty paralelně. Například ve 128 bitech můžou být uloženy 4 slova o velikosti 32 bitů, které se zpracují paralelně. Tyto instrukce jsou obsaženy například v instrukční sadě SSE v moderních PC procesorech (od Pentium MMX výše). Procesory ARM tuto technologii nazývají NEON. Tyto instrukce se využívají ke zpracování videa a signálů.

Celkový výpočetní výkon lze zvýšit i paralelním zpracováním procesů. Jednotlivé procesy (vlákna) může zpracovávat více procesorů najednou. Technologie Hyper-threading umožňuje zpracovávat souběžně v jeden čas dvě vlákna na jednom procesoru. Je toho docíleno zdvojením některých částí procesoru (dekodéry, registry) tak, aby procesor zpracovával dva instrukční toky zároveň. Zvyšuje se tak využití částí procesoru, které nemusejí instrukce v jednom vláknu v daný čas používat.

Podmínkou pro efektivní zpracování instrukcí je též souvislý instrukční tok a nízká přístupová doba operací s pamětí. Z důvodu ceny a fyzikálních limitů je latence pamětí limitována. Vyrovnávací paměti výrazně snižují dobu potřebnou pro vybavení dat z paměti. Tato vyrovnávací paměť (cache), může být určena pro data, instrukce, nebo instrukce a data současně. Protože průměrná přístupová doba je závislá na

složitosti a velikosti vyrovnávací paměti, může být vyrovnávací paměť rozdělena na několik úrovní.

Propustnost pamětí také hraje v aplikačním výkonu významnou roli. V mikrokontrolérech je většinou paměť integrována, ale může být použita i paměť diskrétní v podobě integrovaných obvodů. V oblasti PC se v posledních letech pomalu začíná prosazovat integrovaná paměť HBM. Jde o metodu začlenění dynamické paměti do pouzdra procesoru nebo čipu grafického akcelerátoru Na samotný čip procesoru jsou sendvičově umístěny čipy pamětí a vzájemně elektricky propojeny. Tato nová technologie by měla výrazně zvýšit propustnost sběrnice pamětí, ale zhoršuje odvod tepla ze samotného čipu procesoru. Pro lepší odvod tepla se dnes experimentuje s tak zvanými nanotrubičkami.



Obr.14 Paměti HBM

1.5 Rodiny mikrokontrolérů a obvodů SoC

Mikrokontrolér obsahuje na jednom čipu mikroprocesorové jádro, periferie a většinou i statickou paměť RAM. Jsou určeny především do vestavěných systémů (Embedded systems). Tyto čipy jsou řízeny jednodušším programem, nebo výkonnější mikrokontroléry mohou být řízeny variantou jednoduššího operačního systému.

Obvody SoC (System On Chip) sdružují do jednoho pouzdra mikroprocesorové jádro, periferie a další podpůrné obvody a akcelerátory. Například SoC Apple A4 v sobě sdružuje jádro procesoru ARM (CPU), grafický akcelerátor (GPU), podpůrné periferie a kontrolér paměti DDR. Obvody SoC mohou obsahovat i periferie pro bezdrátový přenos, řadiče a různé specializované obvody. Nacházejí využití jak ve vestavěných systémech tak například i mobilních telefonech a počítačích. Tyto obvody jsou většinou řízeny specializovaným operačním systémem například Android, Linux, Windows Embedded Compact, RTOS-32, FreeRTOS, Nucleus RTOS, uC3, µC/OS-III™ a další.

Architektura procesorového jádra a podpůrných obvodů se liší nejen podle jejich určení a použití, ale i podle vývojové generace a především výrobce. V poslední době se stále více setkáváme se snahou jakési unifikace procesorového jádra napříč rozmanitým spektrem výrobců čipů. Tento trend je zapříčiněn i vývojáři a požadavkem na portování již vytvořeného programu (aplikace) na jiný typ mikrokontroléru resp. SoC.

Mezi další rozdělení těchto čipů patří i oblast použití a pracovních podmínek. Příkladem může být řízení procesů, řízení motorů, měření a zpracování signálů, zabezpečovací systémy, mobilní telefony, přenosné počítače, specializované aplikace. Pracovní podmínky lze rozdělit na nenáročné prostředí (normal), průmyslové prostředí (industrial), automobilový průmysl (automotive), medicínské přístroje (medical), letecký průmysl (aerospace), vojenské využití (military) a kosmický průmysl. Jedním z hlavních kritérií je například rozsah pracovních teplot.

1.5.1 Mikrokontroléry 8051

Rodina mikrokontrolérů 8051 patří mezi jedny z prvních a zároveň i nejznámější mikrokontroléry. Navazuje na svého předchůdce Intel 8048. Mikrokontrolér 8051 vyvinula firma Intel v roce 1980. Dodnes se vyrábějí funkčně posílené 8051-kompatibilní mikrokontroléry. Počet nezávislých výrobců se blíží číslu 20. Mezi ně patří například Atmel, Infineon Technologies, NXP, ST Microelectronics a Texas Instruments. Mikrokontrolér 8051 našel uplatnění v průmyslu i jako součást periferií počítače PC. V bývalém Československu byl vyráběn pod označením MHB 8051 i včetně jeho předchůdce MHB 8048.

Charakteristické vlastnosti:

- Harvardská architektura
- 8 bitový střadač + pomocné registry
- 8 bitová ALU
- 8 bitová datová sběrnice , 16 bitová adresová sběrnice
- RAM 128 bajtů, ROM 4 KiB
- Vstupně výstupní port o velikosti 4 bajty.
- Čítače / časovače o velikosti 16 bitů (dva nebo tři)
- UART (jeden nebo dva)
- Dvouúrovňová priorita přerušení
- Možnost část paměti adresovat bitově (velmi výhodné pro některé aplikace)
- Původní jádro vyžadovalo 12 hodinových cyklů na strojový cykl
- Moderní jádro vyžaduje 1 až 6 hodinových cyklů při frekvenci až 100 MHz
- Instrukce vyžaduje jeden až dva strojové cykly
- Programovací jazyk assembler 8051 nebo i jazyk C
- Existuje mnoho emulátorů a vývojových prostředí

1.5.2 Mikrokontroléry PIC

Rodina mikrokontrolérů PIC patří mezi nejznámější mikrokontroléry. Firma Microchip Technology je vyrábí v 8 bitové verzi (PIC10, PIC12, PIC14, PIC16, PIC17, PIC18), 16 bitové verzi (PIC24) a 32 bitové verzi (PIC32). Uplatnění nacházejí v běžném i v automobilovém průmyslu.

Charakteristické vlastnosti:

- Harvardská architektura
- Šířka datové sběrnice je odlišná od sběrnice instrukční
- Šířka instrukčního slova se liší od verze (např. 10, 12, 14, 16 bitů)
- Jeden střadač označený W
- Paměť RAM je z důvodu přístupu často označována jako registry
- Konfigurační registry a registry periferií jsou mapované do datové paměti
- Využívá se stránkování paměti
- Přísně ortogonální instrukční sada
- Malé množství strojových instrukcí pevné délky (RISC)

- Některé instrukce obsahují podmínku pro vykonání následující instrukce
- Nižší verze neobsahují datový zásobník a mají omezený počet zanoření do podprogramu (což omezuje použití jazyka C)
- strojový cyklus má 4 takty
- instrukce vyžaduje většinou jeden strojový cyklus
- Konstantní odezva na přerušení
- Programovací jazyk assembler PIC, od verze PIC18 i podpora pro jazyk C
- Volně stažitelné vývojové prostředí IDE MPLAB

1.5.3 Mikrokontroléry MSP430

Tyto mikrokontroléry vyrábí firma Texas Instruments. Jedná se o 16 bitové mikrokontroléry určené především pro nízkopříkonová zařízení. Dle typu obsahují periferní obvody SPI, I2C, USB, ADC, DAC, WDT, BOR, DMA, PWM, časovače, čítače, analogové komparátory, analogové operační zesilovače (OpAmp), hardwarovou násobičku a podobně.

Charakteristické vlastnosti:

- Architektura 16-bit RISC, von-Neumann
- ortogonální instrukční sada
- 16 registrů 16-bitových (zde je zahrnut i programový čítač, ukazatel zásobníku, status registr a generátor konstant, tedy pro univerzální použití je určeno 12 registrů)
- Zaměřeno na nízkou spotřebu (např. typ MSP430F550x má uváděnou spotřebu 115 µA/MHz, režim spánku 1.9 µA / 1.1 µA / 0.18 µA při napájení 3 V)
- Většina instrukcí se vykoná během jednoho taktu. Více taktů potřebují operace s pamětí, zásobníkem a skoky.
- Maximální frekvence 25 MHz (typ MSP430F5xx)
- Několik větví pro šíření kmitočtu (např. CPU může běžet na jiné frekvenci než na kterých běží periferie)

1.5.4 Mikrokontroléry AVR

Mikrokontroléry AVR vyrábí firma Atmel. Jedná se především o 8-bitové a 32 - bitové mikrokontroléry. Do této rodiny patří 8-bitové mikrokontroléry ATtiny, ATmega a ATxmega (mají podporu kryptografie) . 32-bitové mikrokontroléry nesou

označení AVR32. Zajímavou alternativou jsou mikrokontroléry FPSLIC™, které obsahují paměť RAM pro programový kód a FPGA o počtu 5 000 až 40 000 hradel.

Charakteristické vlastnosti 8-bitových mikrokontrolérů AVR:

- Harvardská architektura
- RISC (počtem instrukcí se blíží CISC)
- 16-bitová délka instrukcí
- 32 pracovních registrů (u rodiny ATtiny jen 16)
- Spojování registrů umožňuje přímé adresování
- Přístup do paměti operacemi Load / Strore
- Program může využít podmíněné skoky nebo také instrukci SKIP podmíněného vykonání následující instrukce (obdoba jako u PIC)
- Používá například Arduino

1.5.5 Rodina SoC x86

Do této rodiny patří například Vortex86, který v devadesátých letech vyvinula firma Rise Technology a později převzala firma SiS. Procesorové jádro je 32-bitové architektury x86, instrukčně kompatibilní s procesorem Intel Pentium MMX. Uplatnění nachází v ve vestavěných systémech nebo na počítačových deskách jako samostatný počítač.

Charakteristické vlastnosti:

- SoC (System On Chip)
- Architektura CISC x86 32-bit
- Podpora MMX instrukcí
- Obsahuje FPU a hardwarový dekodér videa H.264
- Jedno nebo dvě procesorová jádra (VortexDX3)
- Podpora pamětí DDR3 do kapacity 2 GB
- Cache L1 instrukční 16 KiB + datová 16 KiB, L2 256 KiB
- Ethernet 100Mb/s , SATA, PATA
- ISA, SPI, COM, LPT, I2C, GPIO, JTAG, USB 2.0,PCI Express 1.1

1.6 Architektura procesorů ARM

Procesory ARM jsou intelektuální vlastnictví firmy ARM Holdings (dříve ARM Limited). Tato firma již procesory nevyrábí, ale věnuje se především vývoji této architektury, kterou licencuje jiným výrobcům mikroprocesorů. Výrobci k těmto licencovaným jádrům procesoru přidávají své periferní obvody. Tato strategie umožnila větší objem výroby procesorů ARM a zároveň financování jejich nového vývoje.

Například v roce 2005 bylo licencováno 1,6 miliardy procesorů, v roce 2011 celkem 15 miliard kusů vydaných licencí. (jak uvádí zdroj: http://www.zive.cz/clanky/procesory-arm-zaklad-nove-ery/sc-3-a-164061/default.aspx)

Procesory ARM jsou výkonné moderní procesory s nízkou spotřebou na jeden MHz.

Počátky architektury ARM

První mikroprocesor s architekturou ARM byl navržen v roce 1984. Jednalo se o ARM1, který byl inspirovaný architekturou RISC I. Oproti soudobým procesorům architektury CISC, které obsahovaly menší počet pracovních registrů ale velké množství složitějších instrukcí, se vývoj procesoru ARM opíral o koncept většího počtu univerzálních pracovních registrů a jednoduché instrukční sady inspirované architekturou RISC I. Znatelný rozdíl oproti architektuře procesorů RISC I však byl v implementaci podmíněných skoků a dále místo použití registrových oken byly registry rozděleny do skupin podle aktuálního stavu procesoru. Pracovní registry a interní i externí datová sběrnice měly šířku 32 bitů.

V roce 1986 byl vyvinut procesor ARM 2, který byl oproti svému předchůdci drobně vylepšen a získal hardwarovou násobičku zcela nezávislou na ALU. Procesor na kmitočtu 8 MHz dosahoval výpočetního výkonu 4 MIPS.

Pro porovnání je zde uvedena tabulka soudobých mikroprocesorů

Mikroprocesor	Rok výroby	Počet tranzistorů	Frekvence [MHz]	Výpočetní výkon 16-bit [MIPS]	Výpočetní výkon 32-bit [MIPS]
Motorola 68000	1980	68 000	8	1	0,5
Intel 80286	1982	134 000	8	1,2	-
ARM 2	1986	30 000	8	4	4

Tabulka 3 Porovnání výpočetního výkonu mikroprocesorů

Pracovní registry mikroprocesoru ARM

Pracovní registry jsou členěny do skupin podle aktuálního stavu procesoru. Pro všechny režimy jsou společné univerzální registry r0 až r7. Mezi speciální registry patří:

r15 Program counter (PC) – ukazatel instrukcí

r14 Link register – obsahuje návratovou adresu skoku do podprogramu

r13 Stack poiter – ukazatel vrcholu zásobníku

Příznaky jsou uloženy v registru označeném CPSR (Current Program Status Register). Každý další režim má navíc zvláštní registr SPSR (Saved Program Status Register).

Uživatelský režim	Rychlé přerušení (FIQ)	Normální přerušení (IRQ)	Privilegovaný režim (Supervisor)	Ochrana paměti (Abort)	Čekání na koprocesor (Undefined)
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13	r13_fiq	r13_irq	r13_svc	r13_abt	r13_undef
r14	r14_fiq	r14_irq	r14_svc	r14_abt	r14_undef
r15/PC	r15/PC	r15/PC	r15/PC	r15/PC	r15/PC

Tabulka 4 Pracovní registry procesoru ARM

Instrukční sady mikroprocesoru ARM

Mezi instrukční sady mikroprocesoru ARM patří Jazelle (RCT a DBX), VFP, DSP instrukce, NEON, Thumb a Thumb-2.

Jazelle - instrukční sada pro multi-tasking Java Virtual Machine (JVM)
VFP - instrukční sada pro výpočty s plovoucí desetinnou čárkou
DSP - instrukční sada zaměřená na digitální signálové výpočty
NEON - instrukční sada pro paralelní vektorové výpočty (SIMD)

Instrukční sada Thumb a Thumb-2 byla inspirována analýzou strojových kódů, které vytvářejí překladače jazyka C (resp. C++). Vznikla zkrácením původních 32-bitových instrukčních slov na 16-bitová instrukční slova tak, aby výsledný instrukční tok byl menší při zachování přibližně stejného výpočetního výkonu. Důsledkem je vyšší hustota instrukčního kódu, menší binární obraz programu, menší nároky na vyrovnávací paměť a sběrnici.

Zkrácení instrukcí si vynutilo snížení počtu používaných univerzálních registrů na 8. Instrukční sada Thumb-2 umožňuje vykonávání instrukcí o délce 16-bitů i délce 32-bitů v jednom programu. Výrobce uvádí zmenšení délky kódu při použití instrukční sady Thumb-2 o 32% při poklesu výpočetního výkonu o 2 %.

Využití podmíněného vykonávání instrukcí ilustruje následující příklad. Jedná se o algoritmus výpočtu největšího společného dělitele v jazyce C.

```
int gcd(int a, int b)
{
    while (a != b) do
    {
        if (a > b)
        {
            a = a - b;
        }
        else
        {
            b = b - a;
        }
    }
    return a;
}
```

Níže je uveden stejný algoritmus v assembleru mikroprocesoru ARM.

```
gcd
     CMP      r0, r1
     SUBGT    r0, r0, r1
     SUBLT    r1, r1, r0
     BNE      gcd
```

Jak je z příkladu patrné, jsou podmíněné skoky nahrazeny instrukcemi s podmíněným vykonáním. To zkracuje délku kódu i časovou režii potřebnou pro vykonání skoků (například 3 takty na jeden skok). Pro úplnost lze poznamenat, že v případě použití instrukční sady Thumb-2 by se zkrátil výsledný programový kód na 14 bajtů.

2 Teoretický rozbor řízení modelu auta

Program řídicí jednotky pro účely samořízeného modelu auta musí respektovat dané zákonitosti z oblasti matematiky, fyziky a techniky. Detailní rozbor této problematiky je daleko nad rámec této práce, proto následující kapitoly obsahují stručnější souhrn.

2.1 Související fyzikální vztahy

Zákon zachování energie lze slovy popsat tak, že energii nelze vyrobit ani zničit, ale pouze přeměnit na jiný druh energie. Konkrétně pro uvažovaný model auta platí, že dodaná elektrická energie se přemění na kinetickou energii modelu auta, na ztrátové energie a spotřebu energie elektronickou částí. Ztrátové energie zohledňují energetickou účinnost motorku, tření rotující částí, valivý odpor kol, aerodynamický odpor prostředí, elektromagnetické vyzařování a ztráty způsobené vibracemi.

Celkovou dodanou energii lze spočítat

$$W = \int_0^t p\,dt \tag{2.1}$$

kde W… celková dodaná energie
p… okamžitá hodnota výkonu (příkonu)
t … čas

Okamžitý elektrický příkon modelu auta je dána vztahem

$$p = u \cdot i \tag{2.2}$$

kde p… okamžitá hodnota výkonu
u… okamžité napětí
i … okamžitý proud

Kinetická energie modelu auta je dána vztahem

$$E_k = \frac{1}{2} m v^2 \tag{2.3}$$

kde E_k … kinetická energie modelu auta
m … hmotnost modelu auta
v … rychlost modelu auta

Rychlost je dána vztahem

$$v = \frac{ds}{dt} \tag{2.4}$$

kde v … rychlost
s … dráha
t … čas

Zrychlení je dáno vztahem

$$a = \frac{dv}{dt} \tag{2.5}$$

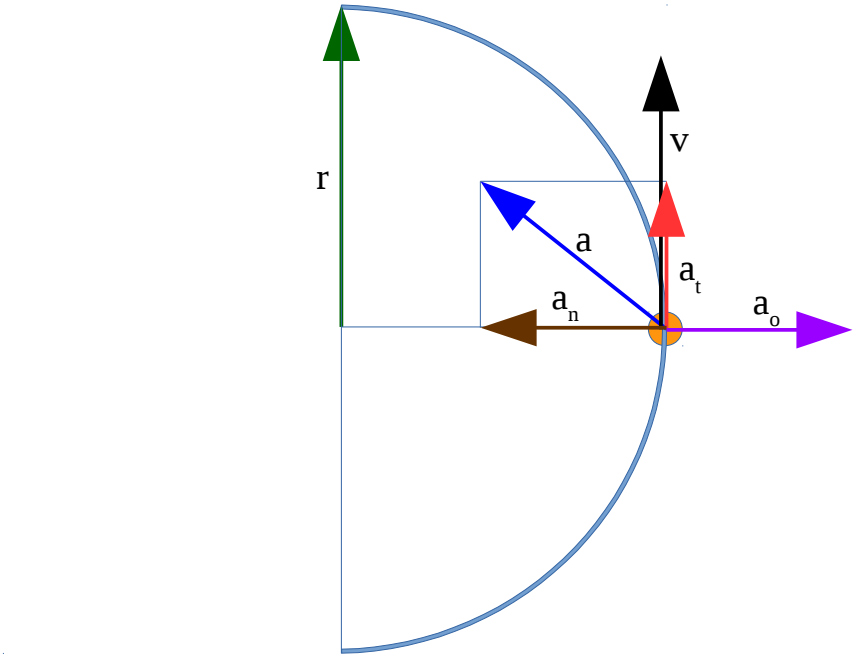
kde a … zrychlení
v … rychlost
t … čas

Síla působící na těleso vlivem zrychlení

$$F = ma \tag{2.6}$$

kde F … síla
m … hmotnost tělesa
a … zrychlení

Trajektorie modelu auta jedoucího v zatáčce s konstantním poloměrem je část kružnice. **Pohyb po kružnici** je znázorněn na obrázku.



Obr.15 Pohyb po kružnici

Tečné zrychlení je dáno vztahem

$$a_t = \frac{dv}{dt} \tag{2.7}$$

kde a_t … tečné zrychlení
v … rychlost
t … čas

Normálové zrychlení je dáno vztahem

$$a_n = r\,\omega^2 = \frac{v^2}{r} \tag{2.8}$$

kde a_n … normálové zrychlení
r … poloměr kružnice
ω … úhlová rychlost
v … rychlost

Výsledné zrychlení je dáno vztahem

$$a = \sqrt{a_n^2 + a_t^2} \tag{2.9}$$

kde a … výsledné zrychlení
a_n … normálové zrychlení
a_t … tečné zrychlení

Odstředivé zrychlení a_o vyvolává odstředivou sílu. Má stejnou velikost jako normálové zrychlení a_n , ale opačný směr.

2.2 Regulace

Na model auta působí v zatáčce odstředivá síla, která vytlačuje zadní část modelu mimo dráhu. Z tohoto důvodu je rychlost modelu v zatáčce limitována. Model auta tedy musí upravit svoji rychlost tak, aby nepřekročil limitní rychlost v zatáčce, ale zároveň mohl využít vyšší rychlost na rovných úsecích. Jednou z možností jak tohoto docílit je regulace.

Regulátor může být spojitý nebo nespojitý. Nespojitý je mnohem jednodušší na realizaci, protože přepíná mezi omezeným množstvím stavů. Nejčastěji používaná varianta nespojitého regulátoru používá dva stavy (např. zapnuto / vypnuto). Nežádoucímu kmitání mezi stavy může zabránit hystereze.

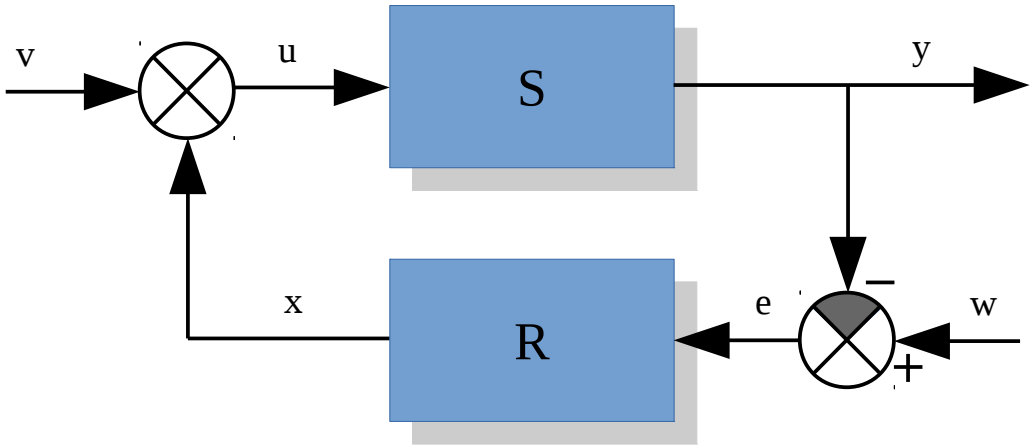
Spojitá regulace je složitější a v mnoha případech vyžaduje nákladnější realizaci, ale regulovaná veličina má mnohem výhodnější časový průběh.

Proporcionálně integračně derivační regulátor (PID) je popsán vztahem

$$x(t) = r_0 e(t) + r_d \frac{de(t)}{dt} + r_i \int_0^t e(t)\,dt + x(0) \tag{2.10}$$

kde x … akční veličina
e … regulační odchylka
t … čas
r_0 … proporcionální konstanta
r_d … derivační konstanta
r_i … integrační konstanta

Pokud je $r_d = r_i = 0$ jedná se o regulátor P, který regulační odchylku pouze zesílí.



S … regulovaná soustava
R … regulátor
v … poruchová veličina
u … řídící veličina
y … regulovaná veličina
w … žádaná hodnota
e … regulační odchylka
x … akční veličina

Obr.16 Schéma regulátoru

2.3 Filtrování signálu

Měřený signál obsahuje krom užitečného signálu také nežádoucí šum. Zdrojem šumu může být například nepřesnost měření (kvantizační šum), elektromagnetické rušení, ovlivnění senzoru okolním prostředím, zvlnění napájecího nebo i referenčního napětí převodníku, či samotný proces převodu analogové veličiny do číslicové podoby. Částečně lze tento šum odfiltrovat vhodným kapacitorem ve funkci dolní propusti na vstupu převodníku. V mnoha případech je nutné z měřeného signálu získat jen signál v určitém frekvenčním pásmu. Tuto funkci zajišťují filtry.

Filtry lze rozdělit podle jejich charakteru:

- elektrické / mechanické / elektromechanické
- analogový / číslicový
- lineární / nelineární
- dolní propust / horní propust / pásmová propust / pásmová zádrž

- s lineární fází / s nelineární fází
- úzkopásmové / širokopásmové

Vlastnosti analogových filtrů:

- spojité signály
- rychlé
- nepřesné
- neflexibilní
- citlivé na přesnost součástek
- citlivé na šum
- velký dynamický rozsah
- nevhodné pro nízké kmitočty

Vlastnosti číslicových filtrů:

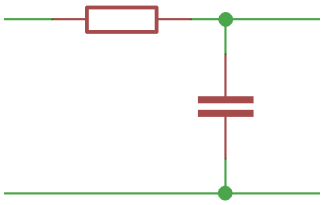
- diskrétní signály
- přesné
- vysoce lineární
- univerzální
- flexibilní
- vhodné pro nízké frekvence
- konečné frekvenční pásmo je rovné polovině vzorkovací frekvence

Číslicové filtry lze dále rozdělit:

- číslicový integrační článek
- klouzavý průměr
- FIR
- IIR

2.3.1 Integrační článek

Číslicový integrační článek je jednoduchý filtr napodobující funkci analogového filtračního článku tvořeného rezistorem a kapacitorem.



Obr.17 Integrační článek

Softwarová implementace integračního článku jako dolní propust:

```
s32 filter (s16 x)  //s32 .. signed long, s16..signed short
{
      static a = 0;

      a += x - (a >> 5);
      return(a >> 5);
}
```

Poznámka: dělení je zde nahrazeno posunem pro ještě rychlejší výpočet

Vlastnosti:

- velmi jednoduchá implementace
- velmi rychlý výpočet
- velmi malé nároky na paměť dat i programu
- filtr 1.řádu, vyšší řády je nutné realizovat řazením za sebe
- strmost -20dB/dek

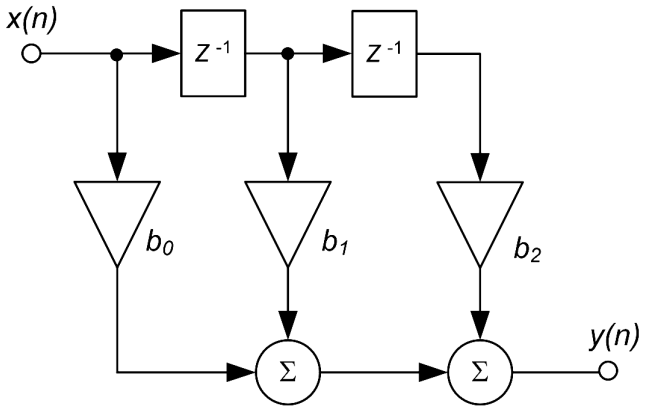
2.3.2 Klouzavý průměr

Jeho princip spočívá ve výpočtu průměru několika po sobě jsoucích hodnot. Jeho výpočet lze urychlit za pomocí kruhového bufferu a akumulátoru, ke kterému se přičítají nové hodnoty a odečítají se od něho hodnoty na konci kruhového bufferu. Výsledek je roven obsahu akumulátoru poděleného počtem prvků kruhového bufferu.

Vlastnosti:

- velmi jednoduchá implementace
- impulsní charakteristika má konečný počet hodnot
- je vždy stabilní

2.3.3 FIR filtr



Obr.18 Schématické znázornění FIR filtru

Filtr je popsán diferenční rovnicí:

$$y[n] = h[0]\,x[n] + h[1]\,x[n-1] + \ldots + h[N-1]\,x[n-N-1] \tag{2.11}$$

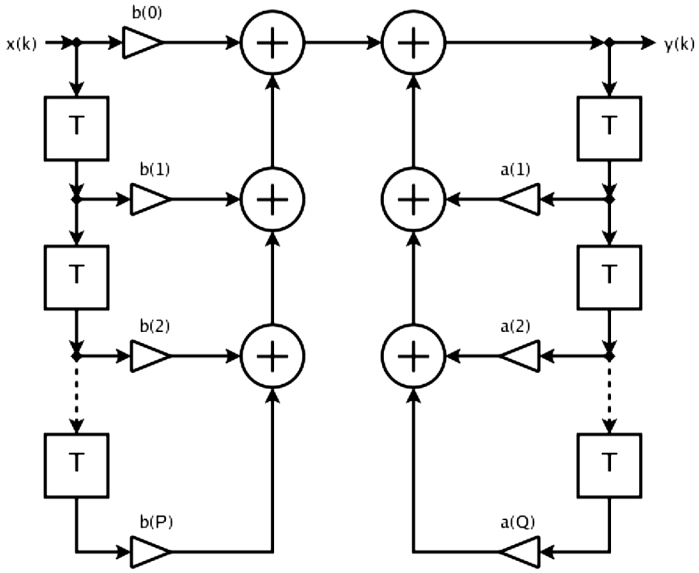
Vlastnosti:

- impulsní charakteristika má konečný počet hodnot
- vždy stabilní (dáno n-násobným pólem v nule)
- může mít lineární fázi
- zpoždění vstupního signálu o (N-1)/2 vzorků
- jednodušší návrh než filtry IIR
- vhodné i pro výpočty v celých číslech

Obecný přenos filtru je dán vztahem

$$h(z)=\frac{B(z)}{A[z]}=\frac{b_0+b_1\cdot z^{-1}+b_2\cdot z^{-2}+\ldots+b_N\cdot z^{-N}}{1-a_1\cdot z^{-1}-a_2\cdot z^{-2}-\ldots-a_M\cdot z^{-M}} \tag{2.12}$$

2.3.4 IIR filtr



Obr.19 Schématické znázornění IIR filtru

Filtr je popsán diferenční rovnicí:

$$y[n] = (1 / a0) * (b0*x[n] + b1*x[n-1] + ... + bP*x[n-P] - a1*y[n-1] - a2*y[n-2] - ... - aQ*y[n-Q] \quad (2.13)$$

Vlastnosti:

- impulsní charakteristika nemá konečný počet hodnot
- není zaručena stabilita
- nelineární fáze
- kratší zpoždění než FIR filtr
- výrazně nižší řád než FIR filtr s podobnými vlastnostmi
- složitější návrh než filtru FIR
- jsou citlivé na přesnost výpočtů, někdy je nutné provádět výpočet v číslech s plovoucí desetinnou čárkou, což má vyšší nároky na výkon CPU nebo numerický koprocesor

3 Rozbor stavu současné řídící jednotky vozu na autodráhu

Současná řídící elektronická jednotka automaticky řízeného vozu na autodráhu (konkrétně SlotCar v. 2.1) byla navržena v roce 2009 za účelem soutěže Freescale Race Challenge.

Od té doby uběhlo více jak 7 let, což by se nemuselo na první pohled jevit jako dlouhá doba, ale v oblasti vývoje mikroelektroniky za tuto dobu došlo k mnoha změnám. Na trhu se objevily nové typy součástek a některé typy se postupně doprodávají ze skladových zásob. Toto se týká především integrovaných obvodů. Jedním z důvodů je neustálý vývoj technologií a jejich lepší ekonomická dostupnost pro výrobce mikročipů. Velký vliv má také vývoj nových typů mikrokontrolérů a jejich architektur.

Výrobci také reagují na poptávku po daných typech součástek. Souběžně s tímto procesem inovují výrobci elektronických zařízení své výrobky, čímž ovlivňují již zmíněnou poptávku po různých typech součástek.

Neméně důležité pro realizaci elektronických zařízení v „kusové“ výrobě je dostupnost „kusového“ množství součástek. Může tak nastat situace, kdy je možné určitou součástku zakoupit pouze ve velkém množství (např. 1000 ks).

Současná řídící elektronická jednotka automaticky řízeného vozu na autodráhu obsahuje součástky, které mají pro „kusovou výrobu“ špatnou dostupnost, nebo se daný typ již nevyrábí.

Konkrétně:

- H-můstek MC33931 je v současné době u prodejců ve **velmi malém množství**. (může to naznačovat doprodej zásob) Náhrada nebyla nalezena.
- Akcelerometr MMA7361 **se již nevyrábí**. Náhrada nebyla nalezena.
- Mikroprocesor MCF51JM128 je v současné době obtížné zakoupit v kusovém množství. Jemu podobné typy ze stejné rodiny mikrokontrolérů většinou již u prodejce nejsou ani skladem.

Nedostupnost některých klíčových součástek si vyžádala provést inovaci řídící jednotky. Inovace zároveň umožnila použití jiného typu mikrokontroléru s větším výpočetním výkonem, větší pamětí pro program a větší pamětí dat.

4 Kritéria pro volbu součástek pro novou řídící jednotku

4.1 Mikrokontrolér

Postupný vývoj technologií má vliv na úroveň integrace, rozsah parametrů a cenu mikrokontrolérů. Na trhu je mnoho typů mikrokontrolérů od několika výrobců.

Parametry ovlivňující výběr vhodného mikrokontroléru:

- velikost paměti programu a dat
- výpočetní výkon procesoru (počet instrukcí za sekundu, aplikační výkon)
- počet vstupů a výstupů. (GPIO)
- komunikační periferie (SPI, I2C, CAN, UART, USB, LAN)
- analogové převodníky, jejich vlastnosti a režimy (ADC, DAC, 8/10/12/16 bit)
- přítomnost a vlastnosti napěťové reference.
- možnost připojení záložní baterie.
- počet a vlastnosti čítačů / časovačů (TMR, CNT)
- počet a vlastnosti generátorů pulsně šířkové modulace (PWM)
- využitelnost přerušení (IRQ, MNI)
- řadiče a správce paměti (DMA, MMU, MPU)
- kontrolér správy paměti FLASH, resp EEPROM a jejich členění (FMC)
- vlastnosti generátorů hodinové frekvence (OSC, RTC, Periodic Interrupt Timer)
- šíření hodinové frekvence do periferií a možnost nastavení.
- úsporné režimy a způsoby probuzení (SLEEP, IDLE, POWER-DOWN)
- vlastnosti hlídacích obvodů (WDT, POR, BOR)
- přítomnost pomocných akcelerátorů (FPU, GPU)
- počet univerzálních registrů a jejich šířka (8, 16, 32, 64-bit)
- přítomnost a členění registrových bank
- podpora operačního systému
- architektura jádra (CISC, RISC, MIPS, ARM)
- maximální, minimální a průměrná spotřeba (ztrátový výkon)
- rozsah pracovních teplot (Commercial, Industrial, Automotive, Extended, Military)
- odolnost proti rušení (EMC)
- možnost ladění (debugování)
- rozměry pouzdra
- cena, dostupnost a technologické nároky

- cena vývojové aplikace (kompilátoru) a dalšího nutného hardware (JTAG)

Na základě prostudování několika katalogových listů mikrokontrolérů od různých výrobců byl zvolen mikrokontrolér LPC1759 od společnosti NXP.

Vlastnosti mikrokontroléru LPC1759:

- architektura: ARM Cortex-M3
- velikost paměti programu: 512 KiB (FLASH)
- velikost paměti dat: 64 KiB
- max. hodinová frekvence: 120 MHz
- pouzdro: LQFP80 (80 pinů)
- komunikační periferie: CAN, I2C, I2S, SPI, UART, USB
- další periferie: ADC, DAC, DMA, PWM, MPU, časovače
- rozsah napájecího napětí 2,4 V až 3,6 V
- rozsah pracovních teplot: -40 °C až +85 °C
- ladící rozhraní JTAG

4.2 Zdrojová část

Zdrojová část každého elektronického zapojení je jedním z klíčových prvků.

Parametry ovlivňující návrh zdrojové části:

- zda je vstupní napětí stejnosměrné či střídavé
- minimální a maximální úrovně vstupního napětí
- výstupní napětí, jeho přesnost a počet výstupních napětí (např. často 3,3V + 5V)
- minimální úbytek napětí (Dropout voltage)
- zvlnění výstupního napětí (vlivem obvodu, změny odběru a vstupního napětí)
- účinnost a vlastní spotřeba
- maximální výkonové ztráty
- tepelný odpor pouzdra (jádro-pouzdro, pouzdro-okolí resp. pouzdro-chladič)
- rozsah pracovních teplot
- složitost celého obvodu zdrojové části
- rozměry pouzdra a celé zdrojové části
- cena ,dostupnost a technologické nároky

Varianta se spínaným zdrojem by přinesla výhodu větší energetické účinnosti. Z toho vyplývají menší nároky na chlazení součástek zdrojové části. Také v případě výpadku dodávky energie by bylo větší využití akumulované energie v kapacitorech. Bohužel spínané zdroje vyžadují větší množství součástek pro jejich realizaci, což v tomto projektu není možné splnit.. Také úbytek napětí by nemusel vyhovovat i pro napájení z USB napětím 5V. Proto varianta se spínaným zdrojem byla zavržena.

Pro zdrojovou část byl vybrán lineární LDO stabilizátor LM2937IMP-3.3. Jako náhradu lze použít MIC5200-3.3YS.

Parametry LP2951CSD-3.3:

- max. vstupní napětí 26 V
- výstupní napětí 3.3 V
- max. výstupní proud 400 mA
- úbytek napětí 400 mV pro proud 250 mA při teplotě 80°C
- pouzdro SOT-223
- pracovní teplota −40°C až 125°C
- ochrana proti zkratu a přehřátí

Parametry MIC5200-3.3YS:

- max. vstupní napětí 26 V
- výstupní napětí 3.3 V
- max. výstupní proud 250 mA
- pouzdro SOT-223
- pracovní teplota −40°C až 125°C
- ochrana proti přehřátí
- levnější než LP2951CSD-3.3

4.3 Akcelerometr

Parametry akcelerometru LIS331DLH:

- rozsah napájecího napětí: 2,16 V až 3,6 V
- rozsah měření zrychlení: možno zvolit ± 2g, ±4g, ±8g
- rozlišení měřeného zrychlení: 12 bit

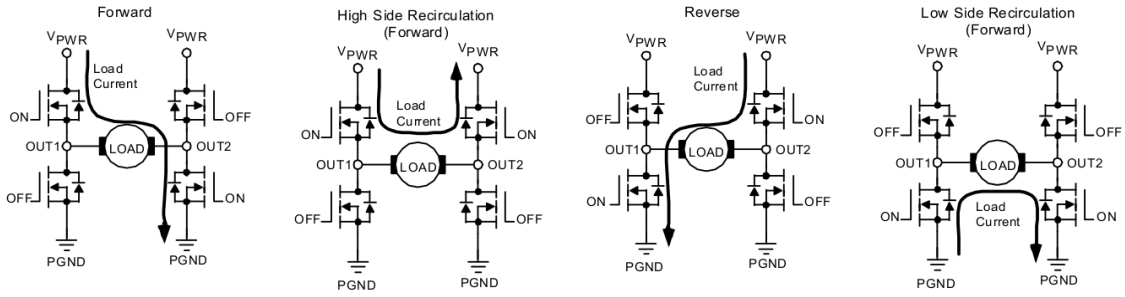
- teplotní vliv: ±0,1 m g /°C
- komunikační rozhraní: SPI
- frekvence vzorkování 0.5; 1; 2; 5; 10; 50; 100; 400; 1000 Hz
- rozsah pracovních teplot: -40 °C až +85 °C

4.4 H-můstek

Pro funkci H-můstku byl zvolen obvod MC33926. Jeho pouzdro má sice mezeru mezi pájecími ploškami pouze 150µm, ale je to jediný vhodný obvod pro tuto řídící jednotku.

Parametry H-můstku MC33926:

- rozsah napájecího napětí: 5 V až 28 V
- napěťové úrovně řídících signálů: kompatibilní s logikou 3,3V a 5V
- max. trvalý výstupní proud: 5A
- typická impedance sepnutého kanálu 120 mΩ
- max. frekvence PWM: 20 kHz
- pouzdro: 32-PIN PQFN
- rozsah pracovních teplot: -40 °C až +125 °C
- ochrana proti zkratu a přehřátí



Obr.20 Schématické znázornění funkce H- můstku MC33926

4.5 Paměť FLASH

Pro uchování změřených hodnot během jízdy byla zvolena paměť FLASH SST25VF010A. Její alternativou je LE25U40CMD, která nedávno zahájila svoji výrobu. Liší se v kapacitě paměti. SST25VF010A má kapacitu 128 KiB. LE25U40CMD má kapacitu 512 KiB, proto byla osazena.

4.6 Ostatní součástky

Výběr ostatních součástek byl proveden s ohledem na jejich parametry, cenu a dostupnost. **Rezistory** zde mají jako hlavní požadavky především požadované napětí. Rezistory použité v děliči napětí mají nižší povolenou toleranci. Výkonové nároky odpovídají použitým pouzdrům.

Použité **kapacitory** mají důraz kladen na volbu dle potřebné kapacity a napětí. U některých filtračních kapacitorů může jistou úlohu sehrát i jejich frekvenční charakteristika a parazitní hodnoty. Tyto speciální požadavky budou řešeny na základě prvních měření osazené desky. Nejedná se zde však o funkcionalitu závislou na vysokých frekvencích a proto se úpravy nepředpokládají.

Použité **diody** vyhovují požadavkům na jejich napětí i dovolené proudové zatížení. Přechodové jevy, jako například připojení na napájení, jsou v toleranci mezních podmínek.

Použitý **varistor** (R20) zde má funkci ochrany obvodů proti napěťovým špičkám (tedy i statické elektřině). U ostatních částí se nepředpokládá přímé ohrožení statickou elektřinou.

5 Rozbor částí elektronického zapojení

Elektronické zapojení je zobrazeno v příloze A. Bylo vytvořeno v aplikaci Cadsoft Eagle 6.3.0.
Jednotlivé části lze rozdělit do následující skupin.

5.1 Zdrojová část

Napájení je přivedeno přes konektor CON1. Varistor R20 chrání další části obvodu před účinky přepěťových pulsů. Vstupní napětí je usměrněno přes usměrňovací blok diod D1 a filtrováno kapacitory C18 a C6, které zároveň filtrují napětí pro H - můstek U3. Vzhledem k možným napěťovým špičkám na napájení je vhodné aby kapacitory měly max. provozní napětí alespoň 25V (nejlépe 50V vzhledem k charakteristice varistoru R20).

Diody D2 a D3 slouží k oddělení konektoru CON2, kde je zapojen externí kapacitor s výrazně vyšší kapacitou. Tento externí kapacitor dodává energii řídící části pokud je krátkodobě přerušen přívod energie. Na tento konektor je možné připojit i rozměrově malou baterii (např. 2x CR2032). Dioda D4 umožňuje napájení z USB konektoru. Lineární stabilizátor IC1 reguluje napětí 3,3V , které napájí obvody U1 (mikrokontrolér), U2 (akcelerometr) , U4 (paměť FLASH) a konektor CON7 pro přípojení SD karty.

5.2 Řídící část

Řídící část tvoří mikrokontrolér LPC1759 (U1). Jeho napájení je filtrováno kapacitory C11 a C12, které jsou k němu připojeny co nejblíže. Mikrokontrolér je možné programovat pomocí rozhraní JTAG přivedeného přes konektor CON4. Vstup $\overline{\text{RESET}}$ je připojen na pullup R18 a kapacitor C21 zajišťující inicializační signál a zároveň filtruje nežádoucí rušení. Vstup $\overline{\text{TRST}}$ je připojen na pullup R8. Volbou osazení R19 nebo R22 se vybírá zda bude JTAG signálem RST ovládat vstupy $\overline{\text{RESET}}$ nebo $\overline{\text{TRST}}$.

Základní hodinovou frekvenci generuje krystal X1, kterou může mikrokontrolér použít pro generování vyšší frekvence pomocí PLL frekvenční násobičky.

Mikrokontrolér ovládá ostatní obvody pomocí signálů na pinech a také přes dvě oddělené SPI sběrnice. Jedna SPI sběrnice je vyhrazena pouze pro komunikaci s akcelerometrem U2. Druhá oddělená SPI sběrnice je sdílena mezi paměť FLASH U4 a konektor pro připojení SD karty. K mikrokontroléru je též připojen konektor USB.

5.3 Akcelerometr

Akcelerometr LIS331DLH (U2) má napájecí napětí filtrované kapacitory C14 a C17. Tato filtrace by měla přispět k odrušení parazitních signálů v napájení. S mikrokontrolérem komunikuje přes SPI sběrnici.

5.4 H-můstek

H-můstek MC33926PNB (IC2) má napájecí napětí filtrované kapacitory C6 a C18. Tato filtrace by měla také odrušit větší část rušení vzniklého při spínání H - můstku vlivem PWM. Časový průběh spínání je vhodné zvolit tak, aby proud motorku procházel vždy jen směrem k zemi.

Motorek je buzen z výstupů OUT1 a OUT2 přes filtr L1. Kapacitor C22 částečně odrušuje vysoké frekvence generované spínáním tranzistorů a motorkem. Pokud nebude osazen filtr L1, neměl by pak být osazen ani kapacitor C22.

Signál MD_SF informuje mikroprocesor o případném zkratu nebo přehřátí H - můstku. Analogový signál MD_SF poskytuje informaci o proudu protékajícím motorkem (je-li připojen). Kapacitor C8 je součást nábojové pumpy pro spínací napětí tranzistorů. Použité hodnoty souvisejících součástek jsou dle dokumentace.

Signály MD_IN1, MD_IN2, MD_D1, MD_D2, MD_EN a MD_SLEW slouží k řízení H - můstku. Jejich bližší význam je popsán v dokumentaci od výrobce.

5.5 Paměť FLASH

Paměť FLASH komunikuje s procesorem přes sběrnici SPI, která je sdílena s konektorem pro SD kartu. Paměť je ovládána signálem $\overline{CE}$ (Chip select) a $\overline{HOLD}$ (odpojení od sběrnice). Pin $\overline{WP}$ je trvale připojen k napájecímu napětí.

5.6 Konektory

Význam konektorů:

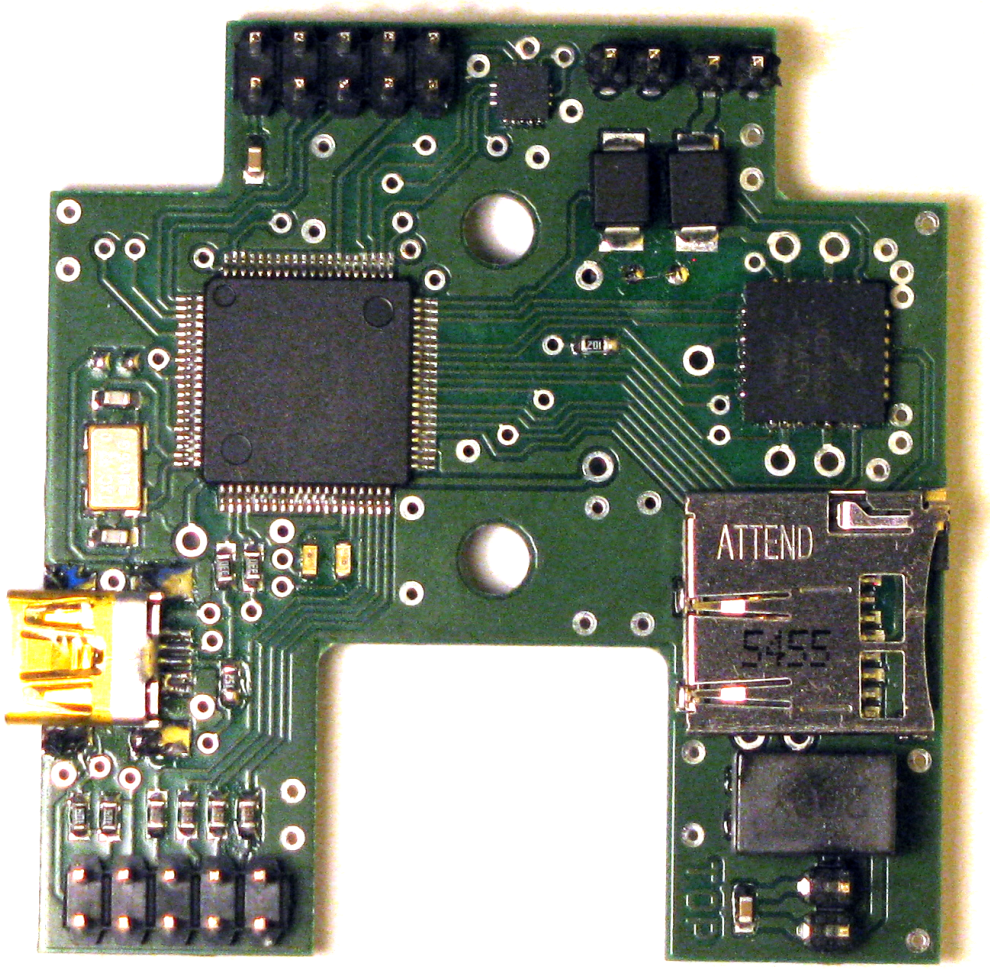
CON1: přívod napájení
CON2: externí kapacitor, nebo externí baterie
CON3: připojení motorku
CON4: JTAG
CON5: USB
CON6: napájení, zem, komunikační signály a signály pro pozdější využití

6 Deska plošných spojů.

Deska plošných spojů je zobrazena v příloze B a C. Jedná se o první prototyp. V přílohách je zobrazena horní i dolní část plošného spoje včetně součástek. Návrh byl vytvořen v aplikaci Cadsoft Eagle 6.3.0.

Při návrhu bylo zohledněno prostorové uspořádání jednotlivých částí. Pevně danou polohu má USB konektor, konektor pro SD kartu (micro-SD) a díry pro uchycení. Součástky byly umístěny tak, aby prostorově nebyly v kolizi s jinou hmotou. Komplikaci při návrhu způsobily díry pro uchycení, protože jejich umístění „rozbíjí“ plochu na několik částí.

Zohledněno bylo také tepelné vyzařování výkonově namáhaných součástek (v rámci prostorových možností). Návrh desky plošných spojů akceptuje nároky na omezení šíření rušení a obecně platná doporučení.



Obr.21 Osazená deska plošných spojů

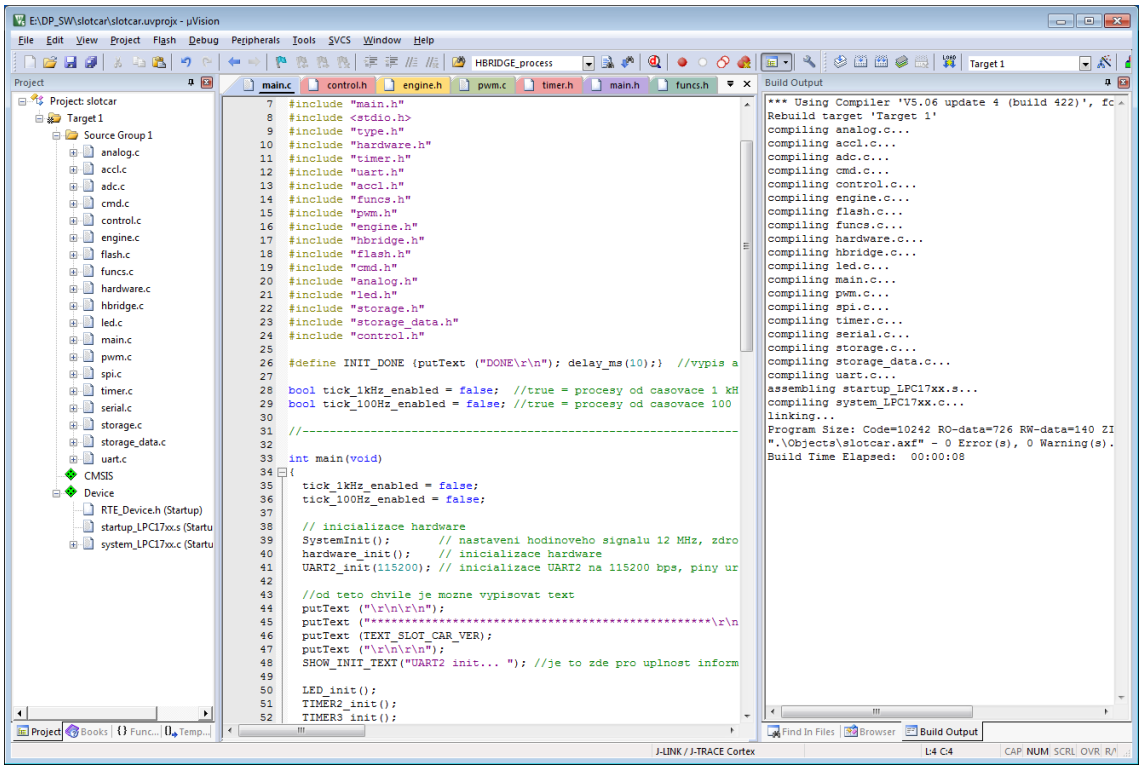
7 Software

Vlastnosti každého zařízení určuje jeho hardwarová část. Například rozměry, vstupní a výstupní tok energie, rozsah teplot a pracovních podmínek. Je-li hardware ovládán programem, určuje vlastnosti i tento program. Často se pro tuto vlastnost užívá pojem chování zařízení.

Součástí realizovaného zařízení je mikrokontrolér, jehož program určuje chování tohoto celého zařízení. Program byl vytvořen v jazyce C a do strojového kódu byl přeložen překladačem.

7.1 Vývojové nástroje

Pro překlad programu v jazyce C byl použit překladač, který je součástí vývojového prostředí Keil µVision 5 (MDK-ARM).

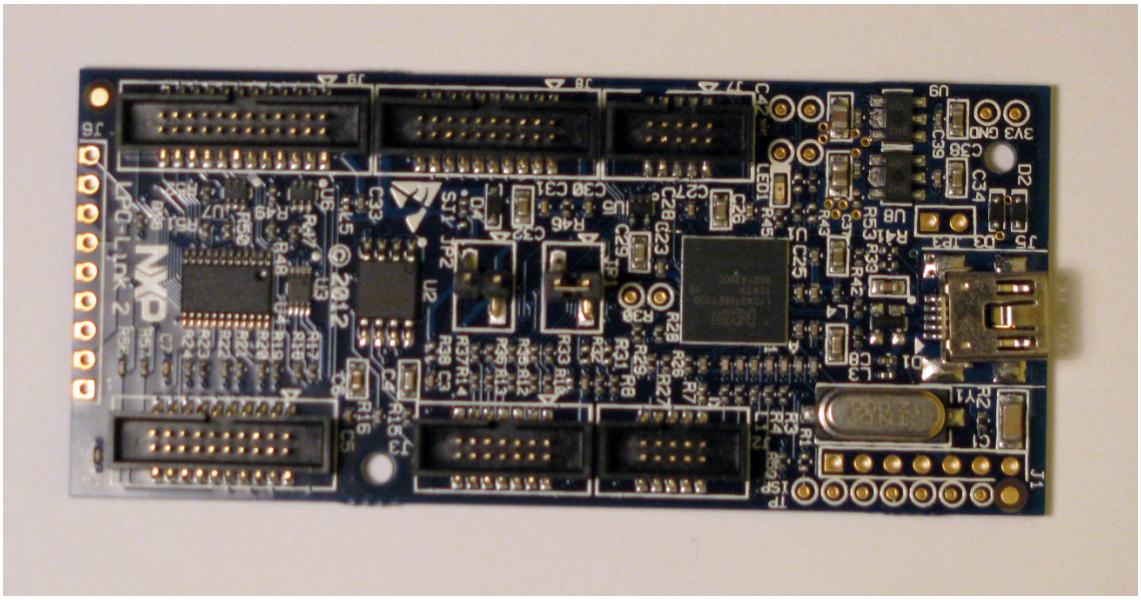


Obr.22 Vývojové prostředí Keil µVision 5

Toto vývojové prostředí je možno po registraci stáhnout ze stránek výrobce procesoru https://www.keil.com/demo/eval/arm.htm

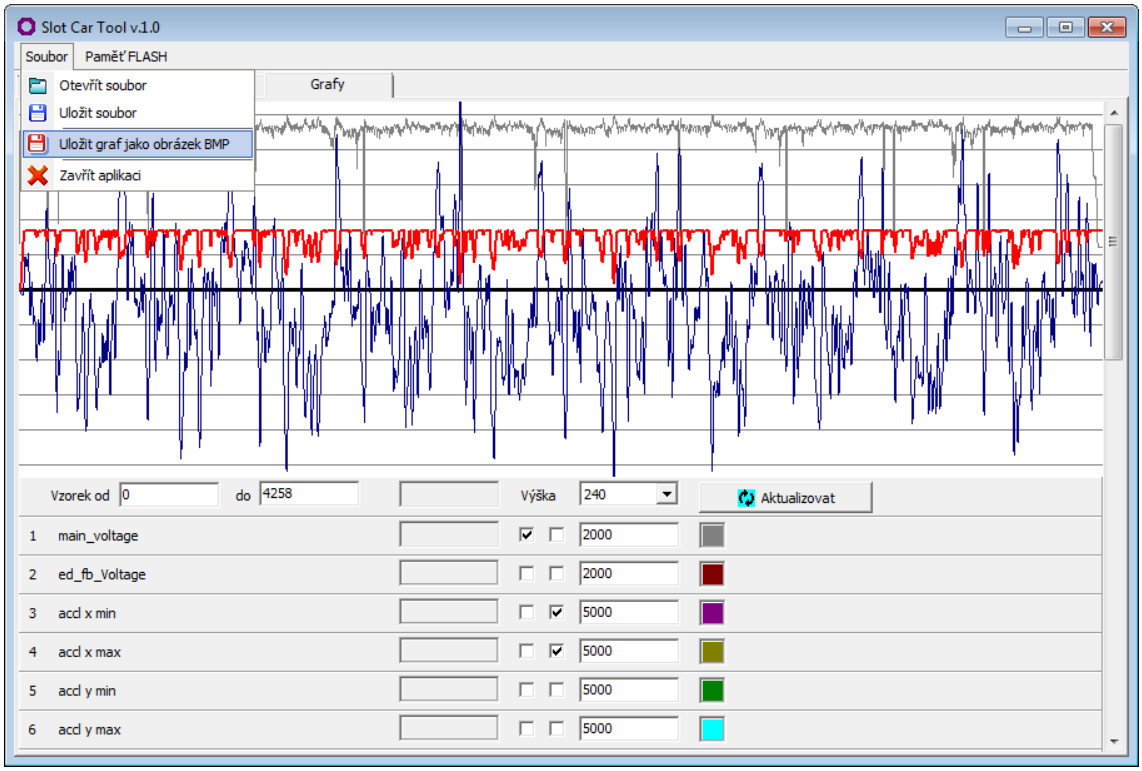
Volně dostupná verze (Free Edition) má dle licenčního ujednání omezení na délku strojového kódu do 32 KiB. Vývojové prostředí umožňuje překlad z jazyka C do strojového kódu, zavedení programu do mikroprocesoru pomocí programovacího nástroje (JTAG) a jeho ladění (debug). Podrobný popis tohoto vývojového prostředí není předmětem této práce.

Pro zápis programu do flash mikrokontroléru a ladění byl použit vývojový kit LPC-Link 2 od firmy NXP, do kterého byl předem zaveden program vykonávající funkci JTAG programátor-debuger. Ve vývojovém prostředí se „hlásí“ pod označením J-LINK.



Obr.23 Vývojový kitu LPC-Link 2 od firmy NXP

Pro snadné a rychlé vyčítání dat z řídící jednotky, včetně následného zobrazení dat v grafu byla autorem této práce vytvořena aplikace Slot Car Tool. Tato aplikace s řídící jednotkou komunikuje prostřednictvím sériové linky. Načtená data lze uložit do souboru, načíst ze souboru a v podobě obrázku uložit i graf. V grafu lze zvolit která data budou zobrazena, v jakém měřítku a v jaké barvě. Lze zvolit začátek i konec zobrazených dat a také odečítat hodnoty z grafu pro zvolený vzorek. Informace o datových kanálech a číselných formátech včetně popisných textů přebírá z vyčtených dat. Tuto strukturu definuje řídící jednotka. Součástí je i jednoduchý terminál pro sledování komunikace a zadávání příkazů jednotce.



Obr.24 Aplikace Slot Car Tool

7.2 Popis řídícího programu

Řídící program určuje chování celého zařízení. Respektuje hierarchické uspořádání do několika vrstev, které mezi sebou komunikují.

Uspořádání vrstev:

➔ **Programová vrstva**
Tato vrstva zajišťuje řízení celého zařízení pomocí komunikace se systémovou vrstvou, které zadává požadavky na vykonání potřebných operací.

➔ **Systémová vrstva**
Tato vrstva vykonává požadavky programové vrstvy a obousměrně komunikuje s podprogramy periferií mikroprocesoru. Jako příklad lze uvést řízení otáček motorku, zpracování změřených veličin, ovládání výstupů, vykonání operací s paměťovým médiem.

➔ **Podprogramy periferií mikroprocesoru**
Tato vrstva zprostředkovává komunikaci mezi nadřazenou vrstvou a periferními obvody mikroprocesoru. Zahrnuje části zdrojového programu, které přímo ovládají periferní obvody mikroprocesoru nebo obsluhují přerušení od jednotlivých periferií.

➔ **Periferní obvody mikroprocesoru**
Tato vrstva zahrnuje vstupně-výstupní obvody mikroprocesoru, komunikační periferie (UART, SPI, PWM), časovače a generátory přerušení

➔ **Fyzická vrstva**
Tato vrstva zahrnuje výkonné členy a detektory, například driver motorku, akcelerometr, indikační LED, napěťové děliče atd.

Zdrojový kód je členěn do souborů dle svého zaměření:

main.c	- hlavní proces (volání podřízených procesů)
control.c	- proces řízení modelu auta (algoritmy + regulátor)
engine.c	- proces řízení výkonu motorku
analog.c	- proces měření analogových veličin
cmd.c	- proces zpracování příkazů ze sériové linky
storage_data.c	- proces generování dat pro zápis do paměťového média
storage.c	- vrstva pro správu zápisu dat do paměti FLASH
flash.c	- vrstva pro správu dat paměti FLASH (čtení, zápis, mazání)
accl.c	- inicializace a komunikace s akcelerometrem
adc.c	- inicializace a komunikace s ADC převodníkem
hardware.c	- inicializace hardware (nastavení funkcí pinů, atd.)
hbridge.c	- inicializace a komunikace s H-můstkem
led.c	- inicializace a ovládání LED
pwm.c	- inicializace a komunikace s generátorem PWM
spi.c	- inicializace a komunikace po sběrnici SPI
timer.c	- inicializace a správa hardwarových časovačů
uart.c	- inicializace a komunikace po sériové lince
funcs.c	- obecné funkce (např. převod čísla do HEX a opačně)
filter.c	- filtry

7.2.1 Časovače a volání procesů

Každý proces, vrstva a ovladač mají svoji inicializační funkci, která je volána po spuštění programu (např. **storage_init**()). Každý proces je volán z příslušného časovače, nebo z nekonečné smyčky (např. **storage_process**()). Z nekonečné smyčky jsou volány procesy, jejichž vykonání může být delší než je doba mezi přerušeními od časovačů.

Jsou to procesy:

```
UART2_process()
storage_process()
```

Hardwarový časovač TIMER2 každou 1 ms volá obsluhu časovače **tick_1kHz**().
Tato funkce se tedy provádí pravidelně s frekvencí 1 kHz a volá následující procesy :

```
analog_process()
engine_power_process()
ACCL_process()
HBRIDGE_process()
LED_proces()
```

Hardwarový časovač TIMER3 každých 10 ms volá obsluhu časovače **tick_100Hz**().
Tato funkce se tedy provádí pravidelně s frekvencí 100 Hz a volá následující procesy :

```
control_process()
storage_data_process()
```

7.2.2 Sériová komunikace

Sériová komunikace je obousměrná. Její obslužný kód je v souboru **uart.c**. Příchozí znak na UART vyvolá přerušení, ve kterém se uloží tento znak do kruhové bufferu a přerušení se ukončí. Pravidelně volaný proces **UART2_process**() přijaté znaky předá k dalšímu zpracování (např. vykonání příkazu)

Znaky odesílané (např. po zpracování příkazu) jsou uloženy do kruhového bufferu. Z něho jsou předány na UART v přerušení, které nastane je-li možné odeslat další znak. Protože toto přerušení nastane až po odeslání znaku, zajišťuje funkce zápisu do kruhové bufferu i přímé odeslání na UART v případě, že vysílací fronta je prázdná.

Pro odesílání znaků po sériové lince jsou určeny tyto funkce:

```
putChar(char c)
putText(char* text)
putBytes(u8* buffer, u32 count)
printf(…)
```

Poznámka: Funkce **printf()** byla přesměrována tak, aby ji bylo možné použít pro odesílání na použitou sériovou linku

Příkazy zpracovávané ze sérové linky:

ver	- výpis informací o verzi sofwaru
flash erase	- vymazání paměti FLASH
analog	- výpis aktuální hodnoty analogových veličin
monitor	- výpis max. počtu taktů CPU pro vykonání přerušení časovače
pwm	- nastavení PWM motorku, pokud neběží regulátor
test	- skupina příkazů pro ladící operace
@xxxxxxxx	- výpis 128 bajtů z adresy xxxxxxxx ve FLASH v hexadecimální podobě

7.2.3 Zápis dat to paměti FLASH

Po startu aplikace je detekována přítomnost dat v paměti FLASH. Pokud tato paměť již obsahuje změřená data, další zápis je zakázán a během vykonávání programu s nic neukládá. Pokud je paměť FLASH prázdná, je zápis dat povolen a během vykonávání programu se ukládají změřená data. Pro zápis dat do této paměti je také použit kruhový buffer. Důvodem je potřebná doba pro fyzický zápis na médium.

Celý řetězec zápisu dat:

storage_data_process()	převezme změřené hodnoty a vytvoří z nich blok dat
storage_write_bytes(…)	zapíše blok dat do kruhového bufferu
storage_process()	nezávisle na probíhajících procesech vyčítá obsah bufferu a předává data do bufferu určeného pro přímý zápis (do FLASH se tedy zapíše blok dat)
storage_flush_device_buf()	tento buffer je předán funkci pro přímý zápis do FLASH
FLASH_page_program(…)	provede přímý zápis do FLASH

7.2.4 Měření a zpracování signálů

Během vykonávání programu jsou měřeny hodnoty akcelerace a napětí na vstupu do H-můstku. Vzorkovací frekvence je 1 kHz. Změřené hodnoty jsou filtrovány.

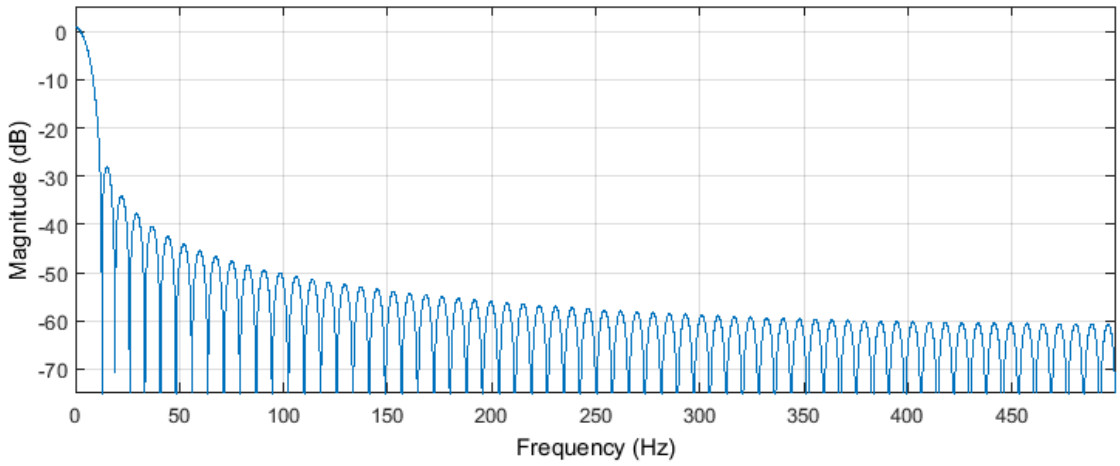
Byly otestovány tyto filtry:

- Integrační článek
- Klouzavý průměr
- FIR filtr
- IIR filtr

Samotný integrační článek neposkytoval dostatečné filtrační vlastnosti, proto bylo otestováno sériové řazení tří integračních článků. Tato varianta již poskytovala dostatečnou filtrační schopnost, ale odezva byla o trochu pomalejší než u filtru FIR.

Klouzavý průměr byl počítán ze 128 po sobě jdoucích vzorků. Jeho filtrační schopnost byla horší než u trojice integračních článků.

FIR filtr byl navržen nástrojem FDATOOL v aplikaci MATLAB. Vykazoval lepší vlastnosti než filtrace pomocí integračních článků a filtrace klouzavým průměrem. Jeho vlastnosti vyhovují požadavkům pro regulaci v řídící jednotce.



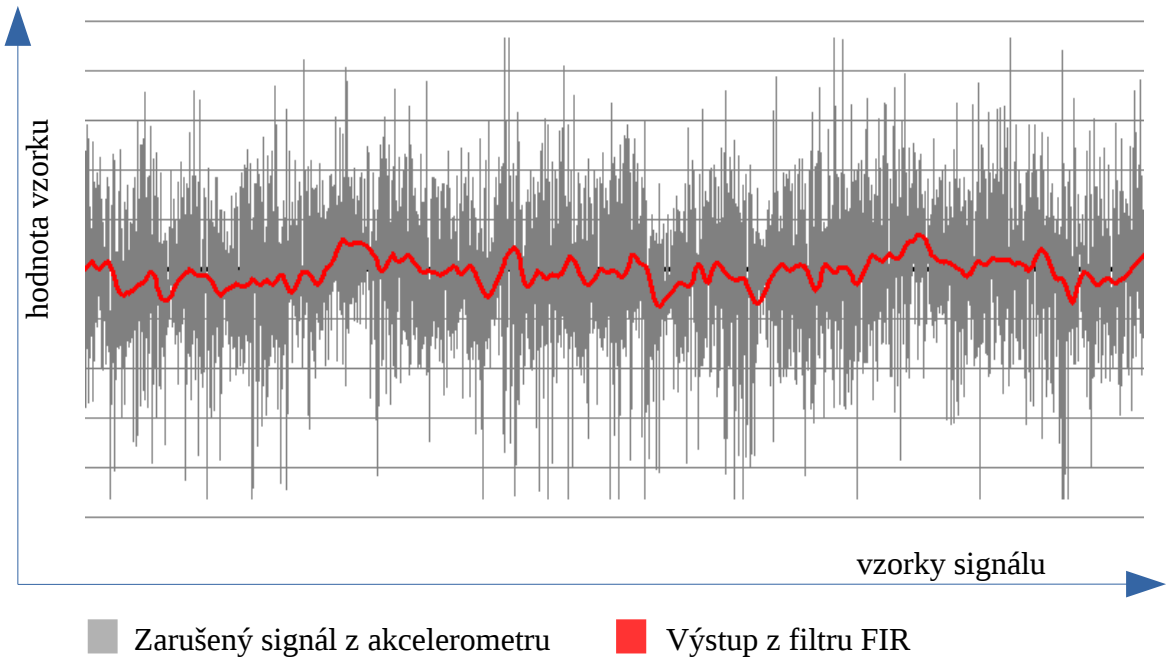
Obr.25 Závislost magnitudy na frekvenci navrženého filtru FIR

Realizace FIR filtru v řídící jednotce:

```
u16 FIR_filter(s16 sample)
{
      const u16 N = 128;
      const s16 B[N] = {
        33, 40, 46, ... tabulka je zde zkrácena ...
      };

      static s16 x[N] = {0};
      static s32 yn;
      u16 i;

      x[0] = sample;
      yn = 0;
      for (i=0 ; i<N ; i++)
      {
            yn += (s32)B[i] * (s32)x[i];
      }
      for (i=(N-1) ; i>0 ; i--)
      {
            x[i] = x[i-1];
      }
      return( yn >> 15 );
}
```



Obr.26 Filtrování hodnot z akcelerometru navrženým filtrem FIR

IIR filtr byl navržen také v aplikaci MATLAB. Protože je IIR filtr velmi citlivý na přesnost prováděných výpočtů, bylo nutné výpočty provádět v číslech s plovoucí desetinnou čárkou. Použitý mikrokontrolér nemá matematický koprocesor, proto prováděné výpočty vyžadují výrazně více strojových cyklů mikrokontroléru, než násobení celých čísel. Z tohoto důvodu nebyl ve finální verzi programu řídící jednotky použit.

IIR filter je realizován takto:

```
s16 IIR_filter(s16 value)
{
      const u8 N = 6;
      const float c = 1.0e-04;

      static float x[N] = {0.0};
      static float y[N] = {0.0};
      static float b[N] = {0.0234 *c, 0.1170 *c, 0.2341 *c,
                           0.2341 *c, 0.1170 *c, 0.0234 *c};
      static float a[N] = {1, -4.4918, 8.0941, -7.3121,
                           3.3110, -0.6011};
      float yn;
      u8 i;

      x[0] = (float)value;
      yn = 0;
      for (i = 1; i < N ; i++)
      {
            yn += - y[i]*a[i];
      }
      for (i = 0; i < N ; i++)
      {
            yn += + x[i]*b[i];
      }
      y[0]= yn;

      for (i=(N-1) ; i>0 ; i--)
      {
           x[i] = x[i-1];
           y[i] = y[i-1];
      }

      return( (s16) yn );
}
```

7.2.5 Regulátor

V řídící jednotce byl otestován proporcionální regulátor (P) i proporcionálně derivační (PD). Výstup z použitého filtru byl vhodný pro obě varianty regulátorů.

Proporcionální regulátor (P) byl realizován vztahem

$$y = 36 + \frac{(1500 - a)}{150} \tag{7.1}$$

kde y … výstupní hodnota
a … kladná hodnota odstředivého zrychlení čtená z akcelerometru

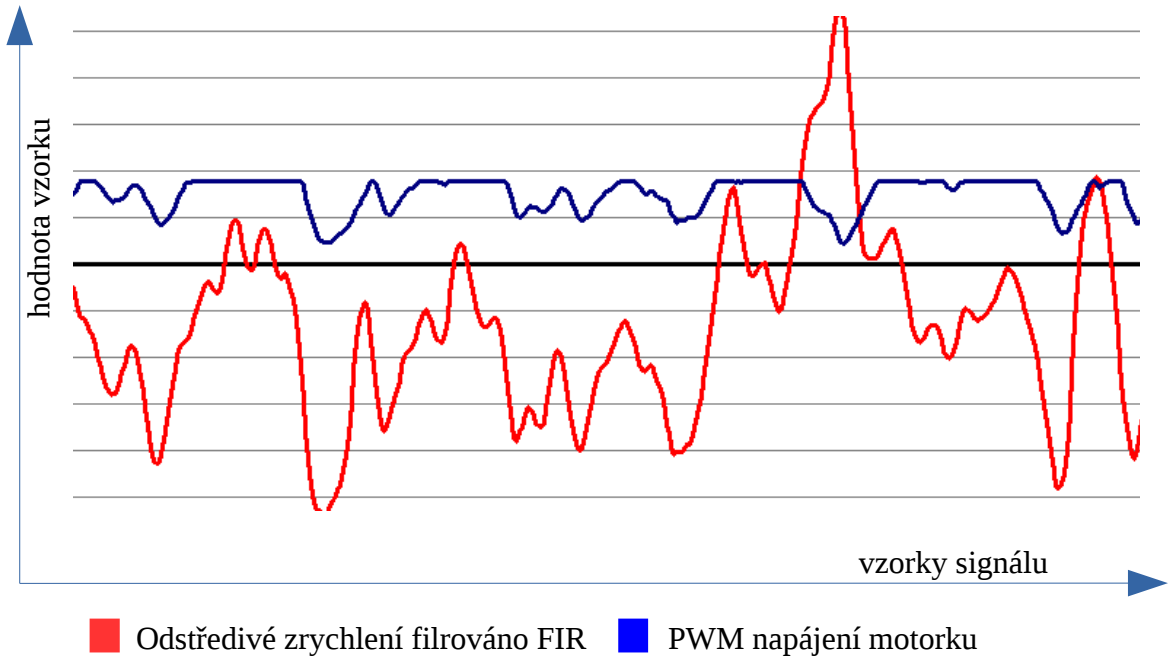
Hodnota 36 je offset regulátoru, která byla též použita pro omezení horního výkonu motorku (36 odpovídá PWM 36 %). Hodnota 1500 koresponduje s požadovanou hodnotou maximálního odstředivého zrychlení. Hodnota 150 určuje zesílení regulátoru. Výstupní výkon je následně ovlivněn rampovým náběhem.

Proporcionálně integrační regulátor (PI) byl realizován vztahem

$$y = 38 + \frac{(1500 - a_n)}{150} + \frac{a_n - a_{n-1}}{200} \tag{7.2}$$

kde y … výstupní hodnota
a_n … kladná hodnota odstředivého zrychlení čtená z akcelerometru
a_{n-1} … předchozí hodnota a_n

Hodnota 38 je offset regulátoru, která byla též použita pro omezení horního výkonu motorku (38 odpovídá PWM 38 %). Hodnota 1500 koresponduje s požadovanou hodnotou maximálního odstředivého zrychlení. Hodnota 150 určuje zesílení regulátoru a hodnota 200 koresponduje s parametrem integrační části regulátoru. Výstupní výkon je následně ovlivněn rampovým náběhem.



Obr.27 Průběh příkonu motorku v závislosti na odstředivém zrychlení

8 Ověření funkčnosti a praktické testy

Řídící část včetně vytvořeného programu byla vyzkoušena v praxi. Nejprve bylo provedeno oživení hardwarové části. Tento krok částečně ověřil základní požadované vlastnosti zařízení. Dalším krokem bylo naprogramování mikroprocesoru základním programem, aby bylo možné jednotlivé části odzkoušet. V následných krocích byl program rozvíjen, testován a laděn. Pro komunikaci s jednotkou a flexibilní zobrazení změřených hodnot byla vytvořena a použita aplikace Slot Car Tool.

Byly otestovány tyto varianty řízení:

a) Projetí celého úseku s konstantní PWM příkonu motorku.
b) Projetí celého úseku s využitím proporcionálního regulátoru (P)
c) Projetí celého úseku s využitím proporcionálně integračního regulátoru (PI)

Projetí celého úseku s konstantní PWM příkonu motorku je velmi závislé na charakteru trasy. Je zřejmé, že prudké zatáčky vyžadují nižší rychlost, která následně v rovných úsecích způsobuje časovou ztrátu. Tato varianta byla použita pro otestování základních funkcí zařízení.

Projetí celého úseku s využitím regulátoru P nebo PI vyžadovalo vhodné nastavení parametrů regulátoru. V praktických testech se projevil velký vliv charakteru trasy a prokluzování kol na jeho povrchu. Nalezení nejoptimálnější metody nebylo účelem této práce.

Srovnání použitých metod řízení:

Metoda řízení	Čas potřebný na projetí tří kol dráhy [s]
a) konstantní PWM	28,2
b) regulátor P	24,1
c) regulátor PI	23,0

Tabulka 5 Časy potřebné na projetí tří kol dráhy pro použité metody řízení

Závěr

Tato diplomová práce vznikla na základě podmětu od Technické univerzity v Liberci na inovaci elektronické řídící jednotky samořízeného modelu auta pro autodráhu, který je využívám pro výuku studentů v oblasti řízení a informatiky. Hlavním důvodem inovace byla nedostupnost a ukončení výroby některých klíčových součástek pro výrobu dosud používané jednotky, jejíž návrh není dílem Technické univerzity v Liberci. Dalším objektivním důvodem inovace bylo využití novějšího typu mikrokontroléru, který svými parametry umožní i složitější metody řízení.

Na základě provedené analýzy současného stavu elektronické řídící jednotky a analýzy problematiky řízení byla navržena a fyzicky vytvořena nová řídící jednotka. Návrh zohlednil dostupnost použitých součástek, dostupnost vývojových a ladících nástrojů, požadavky na složitější metody řízení, snadnou komunikaci s řídící jednotku, vizualizaci měřených hodnot včetně informací souvisejících s během programu a možnosti dalších možných dodatečných úprav i rozšíření.

V teoretické části se tato diplomová práce zabývá problematikou mikrokontrolérů včetně historického vývoje, popisem základních funkčních částí mikrokontrolérů, rozborem procesorových architektur a moderních trendů v této rozsáhlé problematice. Dále se z části zabývá problematikou řízení, měřením a filtrováním signálů, návrhem elektronického zapojení a rozborem řídícího programu.

Nově vytvořená řídící jednotka byla oživena včetně měření charakteristických parametrů a otestována řídícím programem. Dále byla pro řídící jednotku vytvořena aplikace pro vyčítání a vizualizaci dat. Program řídící jednotky byl otestován ve třech metodách řízení příkonu motorku na základě měřených veličin.

V praktických testech byly potvrzeny předpoklady vlivu použitých řídících metod. Nejpomalejší bylo projetí celé dráhy konstantní příkonem motorku. Mnohem výhodnější byly varianty projetí celého úseku na základě řízení použitými regulátory.

Při realizaci této práce byly využity teoretické i praktické znalosti získané studiem. Podrobné otestování funkčního modelu potvrdilo splnění celého zadání této diplomové práce.

Literatura a informační zdroje

DAĎO, S. a M. KREIDL, 1996. *Senzory a měřicí obvody.* Praha: Vydavatelství ČVUT. 315 s. ISBN 80-01-02057-6.

HEROUT, Pavel, 2009. *Učebnice jazyka C*. 6. vyd. České Budějovice: Kopp. 271 s. ISBN 978-80-7232-383-8

HRBÁČEK, Jiří, 2002. *Komunikace mikrokontroléru s okolím 1*. Praha: BEN technická literatura. 160 s. ISBN 80-86056-42-2.

MICROCHIP, *MIC5200* [online]. Micrel Inc., 2005 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/57506.pdf

MICROCHIP, *SST25VF010A* [online]. Rev. A. Silicon Storage Technology, 2011 [cit. 2016-04-01]. ISBN:978-1-61341-695-2. Dostupné z: http://www.farnell.com/datasheets/1669409.pdf

NXP, *LPC1759/58/56/54/52/51* [online]. Rev. 8.6. NXP B.V., 2015 [cit. 2016-04-01]. Dostupné z: http://www.nxp.com/documents/data_sheet/LPC1759_58_56_54_52_51.pdf

NXP, *LPC176x/5x User manual* [online]. Rev. 3. NXP B.V., 2015 [cit. 2016-04-01]. Dostupné z: http://www.nxp.com/documents/user_manual/UM10360.pdf

NXP, *MC33926* [online]. Rev. 10.0. Freescale Semiconductor, 2014 [cit. 2016-04-01]. Dostupné z: http://www.nxp.com/assets/documents/data/en/data-sheets/MC33926.pdf

ON SEMICONDUCTOR, *MBRS130LT3G* [online]. Rev. 9. ON Semiconductor, 2012 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/1708275.pdf

ON SEMICONDUCTOR, *LE25U40CMD* [online]. Rev. 1. Semiconductor Components Industries, 2015 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/2118264.pdf

PINKER, Jiří, 2004. *Mikroprocesory a mikropočítače*. Praha: BEN technická literatura. ISBN 80-7300-110-1

PLÍVA, Zdeněk, 2010. *EAGLE prakticky*. Praha: BEN technická literatura. ISBN 978-80-7300-252-7

ROOT.CZ, 2016. Architektura mikrořadičů s jádry ARM Cortex-M3 [online]. [cit. 2016-04-01]. Dostupné z: https://www.root.cz/clanky/architektura-mikroradicu-s-jadry-arm-cortex-m3

ROOT.CZ, 2016. Mikroprocesory s architekturou ARM [online]. [cit. 2016-04-01]. Dostupné z: http://www.root.cz/clanky/mikroprocesory-s-architekturou-arm

ROOT.CZ, 2016. Mikroprocesory s architekturou RISC - čipy Motorola řady 88000 [online]. [cit. 2016-04-01]. Dostupné z: https://www.root.cz/clanky/mikroprocesory-s-architekturou-risc-cipy-motorola-rady-88000-88k

ROOT.CZ, 2016. Osmibitové mikrořadiče AVR [online]. [cit. 2016-04-01]. Dostupné z: https://www.root.cz/clanky/osmibitove-mikroradice-avr

ROOT.CZ, 2016. Osmibitové mikrořadiče PIC (1) [online]. [cit. 2016-04-01]. Dostupné z: https://www.root.cz/clanky/osmibitove-mikroradice-pic-1

ROOT.CZ, 2016. Příchod hackerů: narodil se mainframe [online]. [cit. 2016-04-01]. Dostupné z: https://www.root.cz/clanky/prichod-hackeru-narodil-se-mainframe

STC, *DBL201G* [online]. Rev. 3. STC, 2005 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/51626.pdf

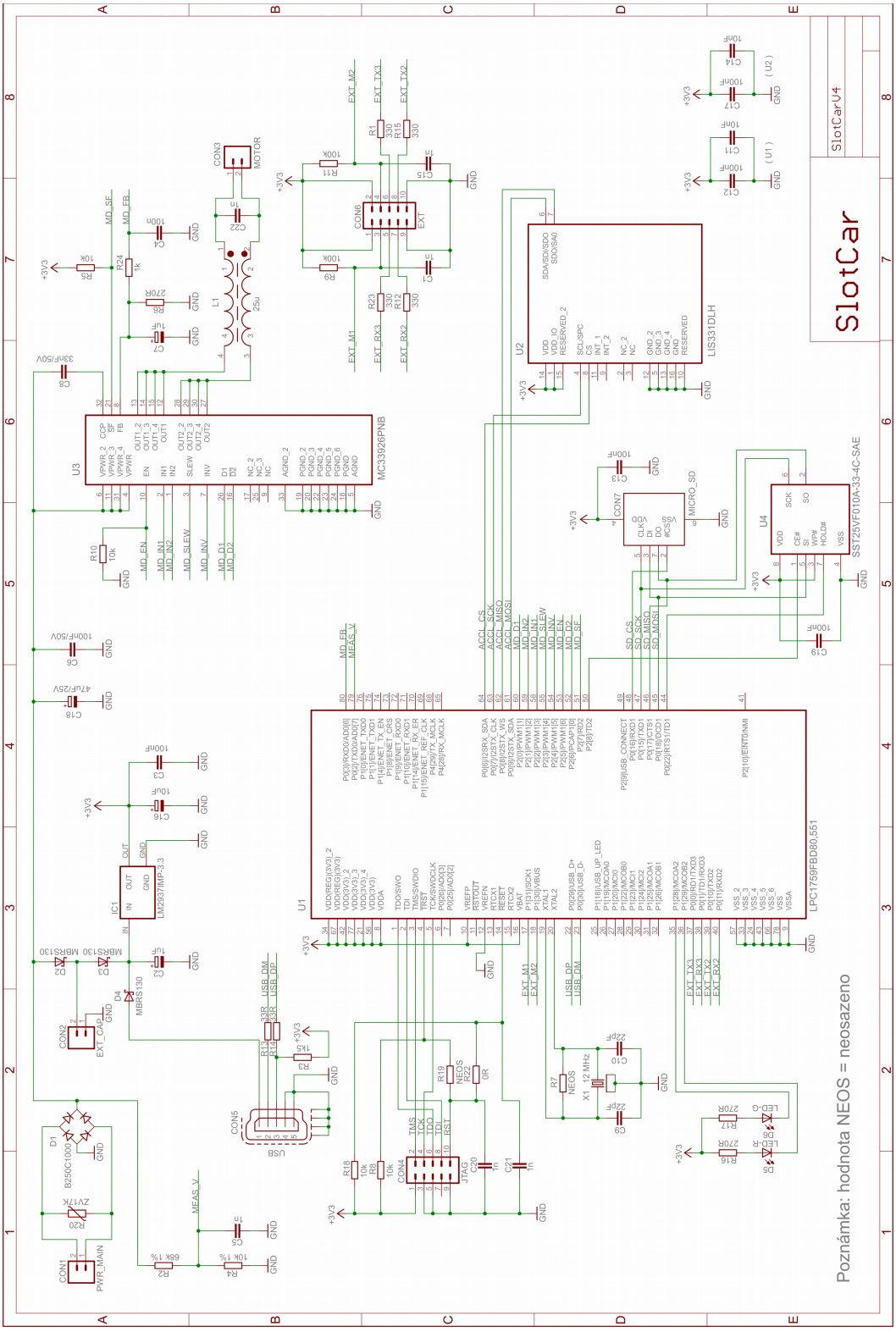
STMICROELECTRONICS, *LIS331DLH* [online]. Rev. 3. STMicroelectronics, 2009 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/1690400.pdf

TEXAS INSTRUMENTS, *LM2937* [online]. Texas Instruments, 2014 [cit. 2016-04-01]. Dostupné z: http://www.farnell.com/datasheets/1997654.pdf

VÁŇA, Vladimír, 2009. *ARM pro začátečníky*. Praha: BEN technická literatura. ISBN 978-80-7300-246-6

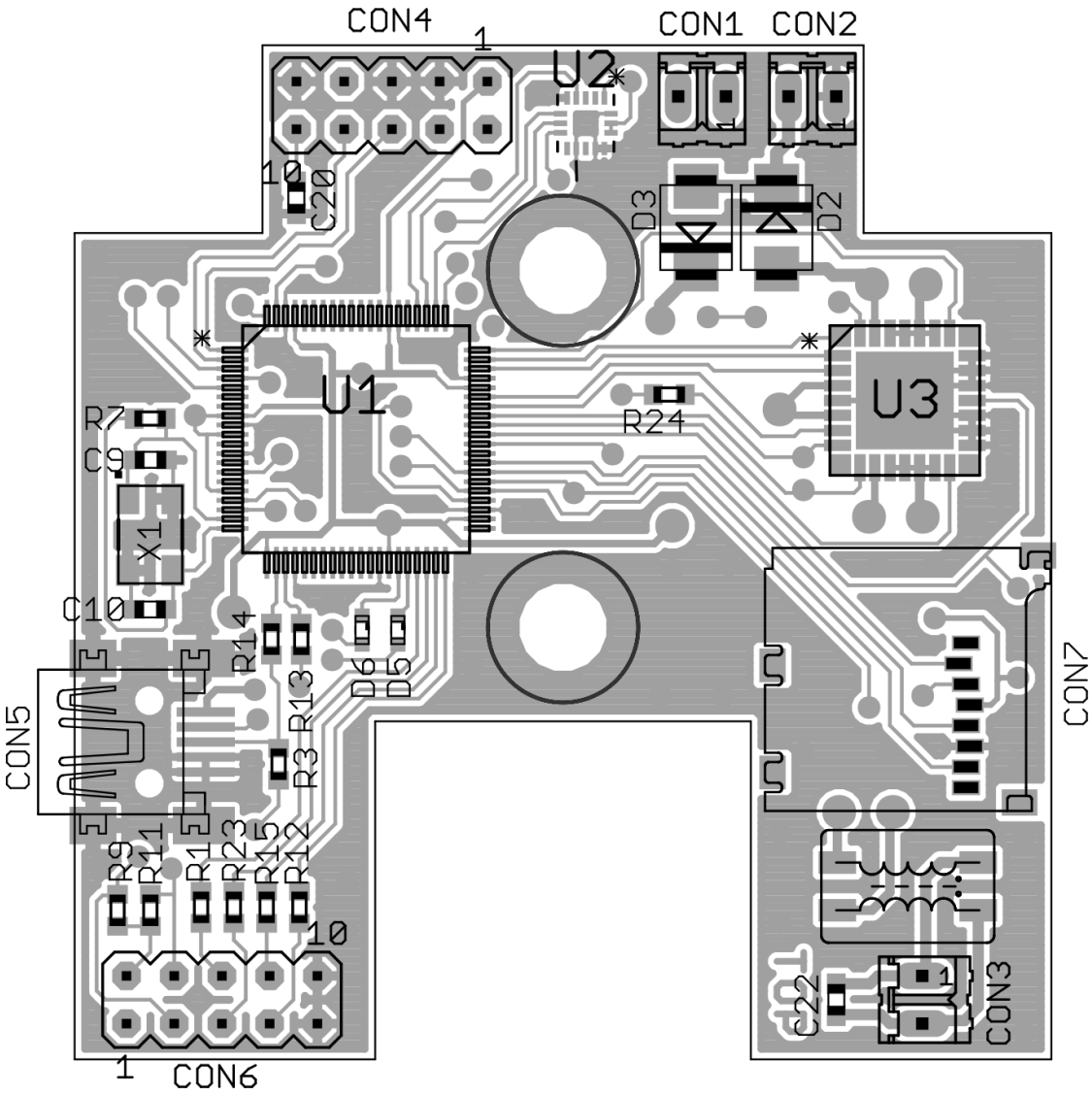
ZÁHLAVA, Vít, 2010. *Návrh a konstrukce desek plošných spojů*. Praha: BEN technická literatura, 123 s. ISBN 978-80-7300-266-4

Příloha A: Schéma obvodu plošného spoje



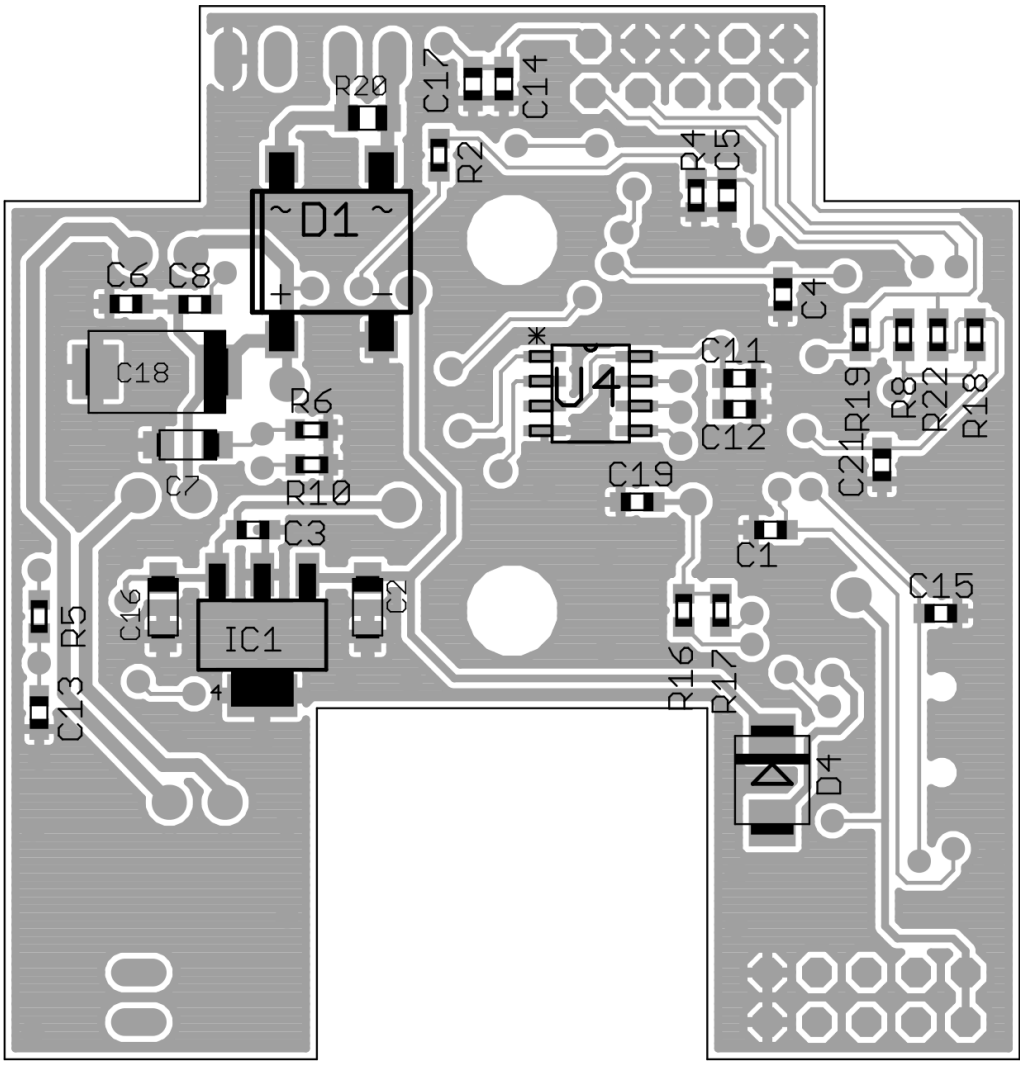
obr.16 Schéma obvodu plošného spoje

Příloha B: Deska plošných spojů – horní strana (TOP)



obr.17 Deska plošných spojů – horní strana (TOP)

Příloha C: Deska plošných spojů – spodní strana (BOTTOM)



obr.18 Deska plošných spojů – spodní strana (BOTTOM)

Příloha D: Seznam součástek

ozn.	hodnota	pouzdro
C1	1n	C0603
C2	1u	SMC_A
C3	100n	C0603
C4	100n	C0603
C5	1n	C0603
C6	100n/50V	C0603
C7	1u	SMC_A
C8	33n/50V	C0603
C9	22p	C0603
C10	22p	C0603
C11	10n	C0603
C12	100n	C0603
C13	100n	C0603
C14	10n	C0603
C15	1n	C0603
C16	10u	SMC_A
C17	100n	C0603
C18	47u/25V	SMC_D
C19	100n	C0603
C20	1n	C0603
C21	1n	C0603
C22	1n	C0603
CON1	PWR_MAIN	02P
CON2	EXT_CAP	02P
CON3	MOTOR	02P
CON4	JTAG	MA05-2
CON5	MINI-USB	MINI-USB
CON6	EXT	MA05-2
CON7	MICRO_SD	MICRO_SD
D1	B250C1000SMD	DBS
D2	MBRS130LT3G	SMB
D3	MBRS130LT3G	SMB
D4	MBRS130LT3G	SMB
D5	LED-R	SML0603
D6	LED-G	SML0603
IC1	LM2937IMP-3.3	SOT223
L1	SRF0905-250Y	9,2mm x 6mm x 5mm

R1	330	R0603
R2	68k 1%	R0603
R3	1k5	R0603
R4	10k 1%	R0603
R5	10k	R0603
R6	270R	R0603
R7	neosazeno	R0603
R8	10k	R0603
R9	100k	R0603
R10	10k	R0603
R11	100k	R0603
R12	330	R0603
R13	33R	R0603
R14	33R	R0603
R15	330	R0603
R16	270R	R0603
R17	270R	R0603
R18	10k	R0603
R19	neosazeno	R0603
R20	ZV17K	CN0805
R22	0R	R0603
R23	330	R0603
R24	1k	R0603
U1	LPC1759FBD80	QFP50P1400X1400X160-80N
U2	LIS331DLH	LGA16R50P3X5_300X300X100
U3	MC33926PNB	QFN80P800X800X220-33N
U4	SST25VF010A	SOIC127P600X175-8N
X1	12 MHz	GSX-531