

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

IP funkce pro vyhodnocení otáčkového čidla

autor: Václav Kanta

2013

Anotace

V diplomové práci je popsán princip inkrementálních rotačních snímačů. Dále je zde probrána problematika měření periody a frekvence, jejich vzájemné porovnání a chyby těchto dvou metod. Je zde také vysvětlena sběrnice ISA a její sběrníkový cyklus pro čtení a zápis do paměťového prostoru. Práce se dále zabývá návrhem a implementací modulu pro vyhodnocení otáček a modulu sběrnice ISA, který zajišťuje komunikaci s PC104. Moduly jsou napsány ve VHDL a otestovány na simulátoru.

Klíčová slova

IRC, inkrementální rotační snímač, ISA, VHDL, FPGA

Abstract

IP function for evaluation of signals from incremental rotary encoder

Incremental rotary sensors are described in this diploma thesis. There is also an explanation of frequency measuring, period measuring, their errors and comparisons of these methods. Next part of this thesis deals with the ISA bus and its bus cycle for memory access. Finally, the thesis covers also a design and implementation of VHDL modules for an ISA communication and for evaluating the signals from incremental rotary sensor. The functionality of these modules was tested in a simulation.

Key words

IRC, incremental rotary encoder, ISA, VHDL, FPGA

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení diplomové práce, je legální.

V Plzni dne 9.5.2013

Václav Kanta

.....

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Doc. Ing. Martinu Poupovi, Ph.D. a konzultantovi Ing. Stanislavu Flíglvi, Ph.D. za odborné vedení práce, ochotný přístup a cenné rady.

Obsah

OBSAH	7
SEZNAM SYMBOLŮ A ZKRATEK	8
1. ÚVOD	10
2. ANALÝZA	11
2.1. INKREMENTÁLNÍ ROTAČNÍ SNÍMAČE IRC.....	11
2.1.1. Optické snímání polohy	11
2.1.2. Výstupní signály	13
2.1.3. Dekódování signálů.....	14
2.1.4. Připojení snímačů.....	15
2.2. ABSOLUTNÍ ROTAČNÍ SNÍMAČE	16
2.3. MĚŘENÍ PERIODY A FREKVENCE.....	17
2.3.1. Měření periody	17
2.3.2. Měření frekvence	18
2.3.3. Chyby měření periody a frekvence	18
2.4. ISA SBĚRNICE	21
2.4.1. Architektura ISA	21
2.4.2. Signály ISA	23
2.4.3. Paměťový 16 bitový sběrniceový cyklus ISA.....	29
3. NÁVRH A IMPLEMENTACE	32
3.1. MODUL VYHODNOCENÍ OTÁČEK IRC_TOP.....	32
3.1.1. Komponenta IRC	33
3.1.2. Komponenta MEM.....	36
3.2. MODUL ISA_TOP.....	37
3.3. VYHODNOCENÍ VÝSTUPNÍCH REGISTRŮ	38
3.4. SYNTÉZA.....	39
4. SIMULACE	40
4.1. SIMULACE ISA KOMUNIKACE	40
4.2. SIMULACE MODULU PRO VYHODNOCENÍ OTÁČEK	43
5. ZÁVĚR	44
6. SEZNAM POUŽITÉ LITERATURY	45
7. PŘÍLOHA 1 – SEZNAM ŘÍDÍCÍCH REGISTRŮ	46
8. PŘÍLOHA 2 – SEZNAM VÝSTUPNÍCH REGISTRŮ	52

Seznam symbolů a zkratek

LED	Light Emitting Diode (dioda emitující světlo)
MSB	Most Significant Bit (nejvýznamnější bit)
LSB	Least Significant Bit (nejméně významný bit)
IRC	Incremental Rotary Encoder (inkrementální rotační senzor)
VHDL	Hardware Description Language (jazyk sloužící pro popis hardware)
FPGA	Field Programmable Gate Array (programovatelné hradlové pole)
ISA	Industry Standard Architecture (sběrnice)
PC104	zabudovaný počítač
IP	Intellectual Property (duševní vlastnictví)
X1	jednonásobné rozlišení dekodéru IRC
X2	dvojnásobné rozlišení dekodéru IRC
X4	čtyřnásobné rozlišení dekodéru IRC
α	úhel mezi ryskami segmentu A a B pevného nepohyblivého kotouče
k	celé číslo
n	počet všech rysek na obvodu rotujícího kotouče
A, B	výstupní kvadrurní signály IRC
NOT A, NOT B	výstupní komplementární kvadrurní signály IRC
ZERO	výstupní nulový signál IRC
NOT ZERO	výstupní komplementární nulový signál IRC
CLK	hodinový signál
β	úhel natočení hřídele
cnt	hodnota čítače dekodéru IRC
N	počet vygenerovaných pulsů snímače za jednu otáčku
x	typ kódování
T_{IN}	perioda vstupního měřeného signálu
f_{IN}	frekvence vstupního měřeného signálu
+EV/2	komparační úroveň
n	počet pulsů
$t, t1, t2$	časové intervaly počítání pulsů
x	šum na vstupu čítače
e_n	efektivní hodnota šumu v měřeném signálu
$\Delta V / \Delta T$	rychlost přeběhu
$count_error$	chyba napočítaných pulsů ± 1
$time_base_error$	chyba frekvence oscilátoru
$trigger_error$	chyba spouštění
m	počet period měřeného signálu přes které proběhlo měření
IBM PC	IBM Personal Computer (osobní počítač IBM)
PC XT	IBM Personal Computer XT (Extended Technology)

PC AT	IBM Personal Computer AT (Advanced Technology)
PC/XT	ISA sběrnice pro 8 bitový počítač PC XT
PC/AT	ISA sběrnice pro 16 bitový počítač PC AT
Intel 8086	8bitový mikroprocesor firmy Intel architektury x86
Intel 80286	16bitový mikroprocesor firmy Intel architektury x86
I/O	vstupně/výstupní prostor
DMA	Direct Memory Access (přímý přístup do paměti)
T_s	adresovací interval sběrniceového cyklu ISA
T_c	datový (čekací) interval sběrniceového cyklu ISA
Avalon-MM	Avalon Memory Mapped Interface
FSM	Finite State Machine
DPMI	DOS Protected Mode Interface
LUT	Lookup Table (vyhledávací tabulka)
IRC_A	první inkrementální rotační snímač
IRC_B	druhý inkrementální rotační snímač

1. Úvod

Cíl diplomové práce je navrhnout modul v jazyce VHDL pro vyhodnocení snímačů otáček. K modulu budou připojeny dva inkrementální rotační snímače, uživatel tak získá možnost kontroly výstupních dat, jelikož bude mít výstupní informace dvakrát. Modul musí umožnit vyhodnocení otáček z libovolných inkrementálních snímačů, které mají dva základní výstupní signály posunuté o 90° elektrických a jeden nulový impulz pro určení absolutní polohy hřídele. Požadavek na počet impulzů snímače na otáčku je v rozsahu 200 až 2000. Požadavky na výstupní informace modulu jsou data pro zjištění rychlosti otáčení hřídele. Rychlost otáčení se bude zjišťovat měřením periody signálu a měřením frekvence signálu. Další požadavek na výstup modulu je informace o absolutní poloze hřídele. Poslední požadavek výstupu je počet impulzů od posledního vyčtení dat sloužících k určení rychlosti otáčení.

Modul také musí obsahovat řadič ISA sběrnice pro umožnění komunikace se zabudovaným počítačem PC104. IP funkce bude nahrána do programovatelného logického pole Xilinx Spartan-II na desce FPGA 6800, která obsahuje rozhraní 16 bitové ISA sběrnice a oscilátor 50 MHz.

V kapitole č. 2 je provedena analýza jednotlivých částí potřebných k realizaci modulu pro vyhodnocení čidla otáček. Je zde popsán princip inkrementálních rotačních snímačů a jejich výstupní signály. V kapitole je také probrána problematika měření periody a frekvence, jejich vzájemné porovnání a chyby těchto dvou metod. Na závěr kapitoly je vysvětlena sběrnice ISA (Industry Standard Architecture), význam jednotlivých signálů a popis sběrnicového cyklu pro čtení a zápis do paměťového prostoru.

V následující kapitole č. 3 je popis návrhu modulu pro vyhodnocení čidel otáček a modulu řadiče ISA sběrnice. V kapitole je popsán stavový automat pro vyhodnocení otáček a řešení detekce výpadku jednoho ze signálů. Seznam řídicích registrů je v příloze 1 a seznam výstupních registrů je v příloze 2. Dále kapitola pojednává o syntéze IP funkce a přístupu k registrům v paměťovém prostoru pomocí kompilátoru jazyka C (Open Watcom).

Kapitola č. 4 se zaměřuje na testování ISA komunikace a na testování modulu pro vyhodnocení inkrementálního rotačního snímače.

Závěrečná kapitola č. 5 shrnuje výsledky práce.

2. Analýza

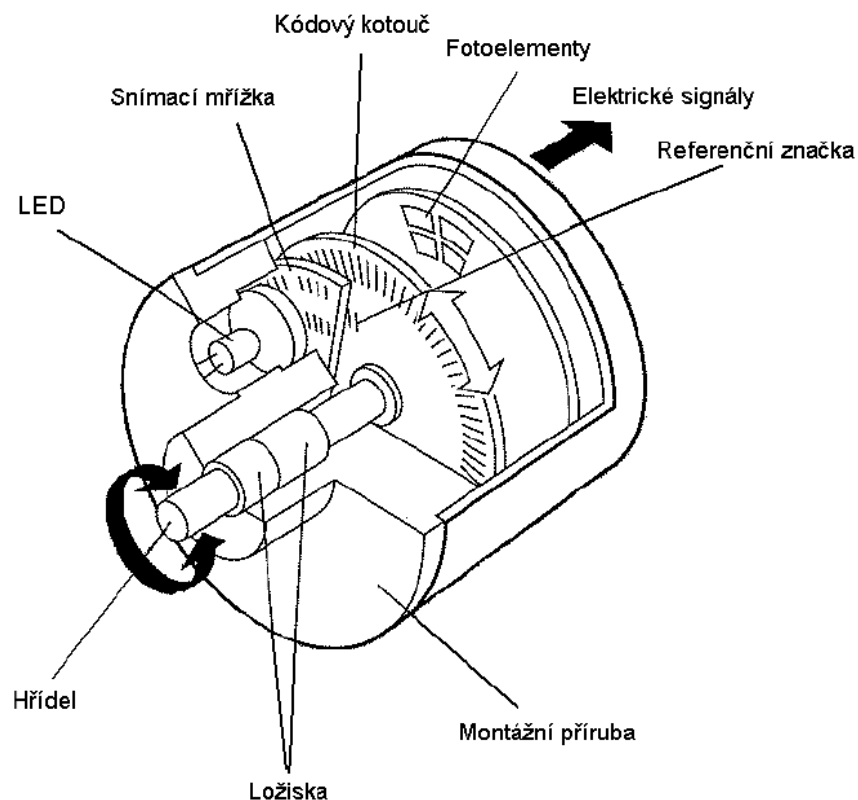
V této kapitole jsou popsány optické rotační snímače, problematika měření periody a frekvence a je zde vysvětlena komunikace přes sběrnici ISA.

2.1. Inkrementální rotační snímače IRC

Informace o inkrementálních rotačních snímačích jsem čerpal z [1], [2], [3] a [4].

2.1.1. Optické snímání polohy

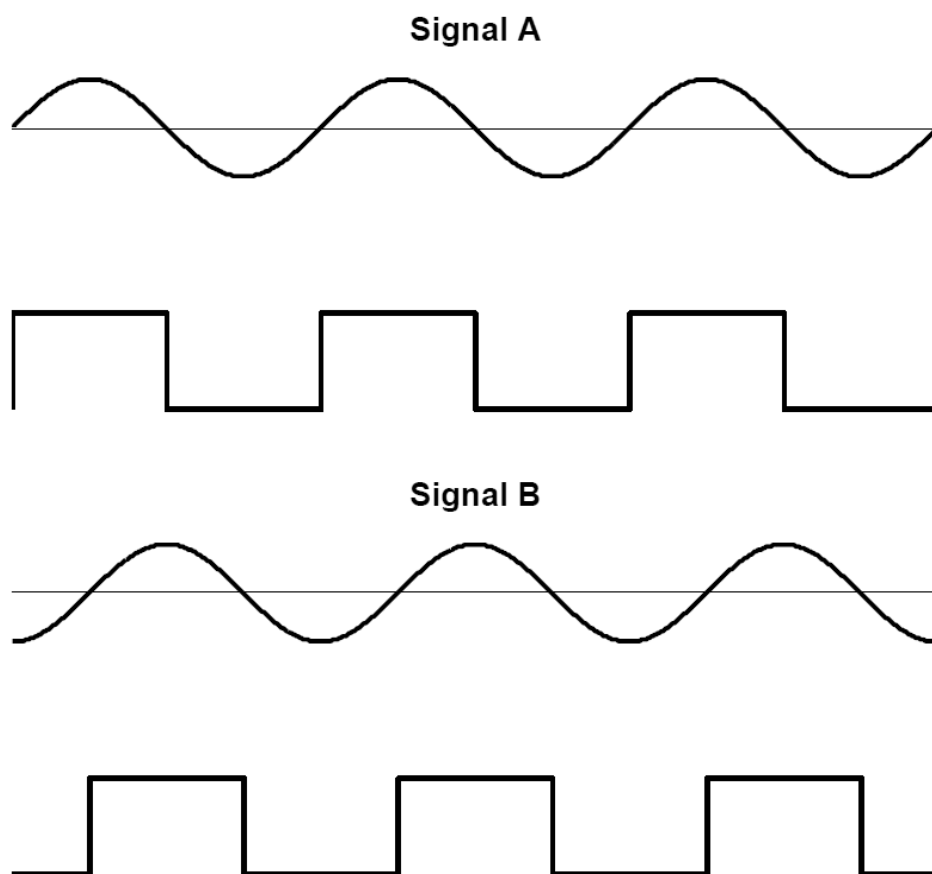
Rotační inkrementální optický snímač má čtyři základní části: pohyblivý kotouč s pravidelně se střídajícími průhlednými a neprůhlednými ryskami po celém obvodu, pevný nepohyblivý kotouč s ryskami o stejné rozteči jako rysky pohyblivého kotouče, svítivou LED diodu a snímač světla (fototranzistor nebo fotodioda). Pohyblivý kotouč je připevněn na hřídeli. Hned vedle pohyblivého kotouče je umístěn pevný nepohyblivý kotouč. Při otáčení hřídele, pohyblivý kotouč přerušuje procházející světlo z LED diody na snímač světla, který je připevněn na druhé straně naproti LED diodě. Uspořádání klasického optického inkrementálního rotačního snímače je na *obr. 2.1*.



Obr. 2.1. Inkrementální optický rotační snímač.[5]

Jsou-li v zákrytu průhledné rysky pohyblivého kotouče a pevného kotouče, množství dopadajícího světla na snímač světla je maximální. Pokud jsou v zákrytu průhledné rysky pohyblivého kotouče a neprůhledné rysky pevného kotouče, množství dopadajícího světla na snímač světla je minimální. Velikost světelného toku dopadající na snímač světla je přímo úměrná vzájemné poloze rysek pohyblivého kotouče a rysek pevného kotouče. Frekvence výstupního signálu snímače světla je přímo úměrná počtu rysek na otáčku.

Na aktivních plochách snímače světla se objeví teoreticky pilovitý průběh světelného toku, kde maximum je v momentě zákrytu průhledných rysek kotoučů. Ve skutečnosti, ale průběh světelného toku není pilovitý, ale sinusový. Zaoblení je způsobeno nedokonalou ostrotí rysek a také tím, že poloha a rozměry rysek nejsou úplně přesné. Výstupní sinusový signál snímače světla je komparátorem převeden na obdélníkový signál. Průběh signálů **A** a **B** je znázorněn na *obr. 2.2*. Inkrementální rotační snímače mohou být také kapacitní (využívají změnu kapacity kondenzátoru), magnetoelektrické (využívající Hallovu sondu), indukční (fungují na principu změny indukčnosti cívky). Optické rotační snímače patří mezi nejpřesnější.



Obr. 2.2 Signály A a B. Sinusové průběhy jsou výstupní signály senzorů světla (obdélníkové průběhy jsou získané komparátorem)

2.1.2. Výstupní signály

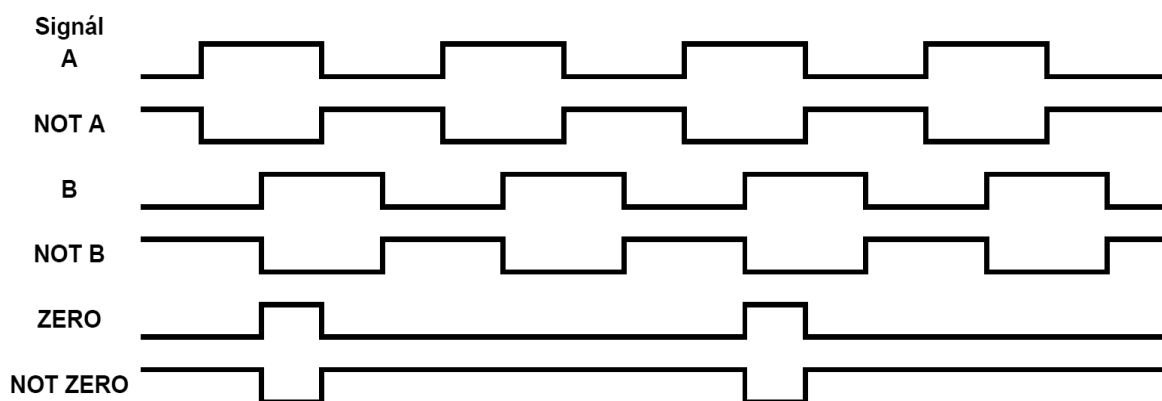
Abychom mohli určit i směr otáčení, pevný nepohyblivý kotouč musí mít druhý segment s ryskami posunutými (vůči prvnímu segmentu) o úhel

$$\alpha = \left(k + \frac{1}{2}\right) \cdot \frac{2\pi}{n} \quad (2.1)$$

kde n je počet průhledných i neprůhledných rysek dohromady na obvodu rotujícího kotouče a k je celé číslo. Tomuto segmentu přísluší další zdroj světelného záření a další snímač světelného záření. Signál z prvního snímače světelného záření se značí **A**, signál z druhého snímače světelného záření se značí **B**. Signály **A** a **B** se také nazývají kvadraturní.

Pro určení absolutní pozice hřídele má otáčivý kotouč ještě průhlednou rysku na menším poloměru než jsou rysky po celém obvodu. Pevný nepohyblivý kotouč má na odpovídajícím poloměru také průhlednou rysku. Těmto ryskám odpovídá třetí zdroj světelného záření a také snímač světelného záření. Snímač světelného záření v tomto případě vygeneruje pulz jednou za otáčku hřídele. Šířka takového pulzu je rovna čtvrtině periody rastru průhledných a neprůhledných rysek na obvodu kotouče. Tento signál se nazývá referenční, nulový (**ZERO**) nebo index-puls.

Některé inkrementální snímače mají kvadraturní signály a nulový signál doplněny komplementárními signály aby se zvýšila odolnosti vůči rušení. Průběhy kvadraturních signálů, nulového signálu a jejich komplementárních signálů jsou zobrazeny na *obr. 2.3*.



Obr. 2.3 Průběhy kvadraturních signálů a nulového signálu pro rotační snímač s 2 pulzy na otáčku (signály označené NOT jsou komplementární)

2.1.3. Dekódování signálů

Dekódování výstupních kvadraturních signálů **A** a **B** inkrementálního rotačního snímače se provádí čítáním pulzů těchto signálů. Podle toho, které hrany pulzů detekujeme, můžeme dekodování rozdělit na 3 druhy:

- s jednonásobnou přesností (X1) – detekujeme náběžnou nebo sestupnou hranu pouze jednoho kvadraturního signálu
- s dvojnásobnou přesností (X2) – detekujeme náběžnou a sestupnou hranu pouze jednoho kvadraturního signálu
- s čtyřnásobnou přesností (X4) – detekujeme náběžnou a sestupnou hranu obou kvadraturních signálů.

Směr otáčení je určen tím, který kvadraturní signál vede druhý. Objeví-li se první puls signálu **A**, jedná se o pravotočivý směr. Pokud se první objeví puls signálu **B**, směr otáčení je levotočivý. V závislosti na směru otáčení se hodnota čítače zvyšuje (pravotočivý směr) nebo snižuje (levotočivý směr) [4]. Jednotlivé způsoby dekodování jsou zobrazeny na *obr. 2.4*. Úhel natočení hřídele β se vypočítá podle vztahu:

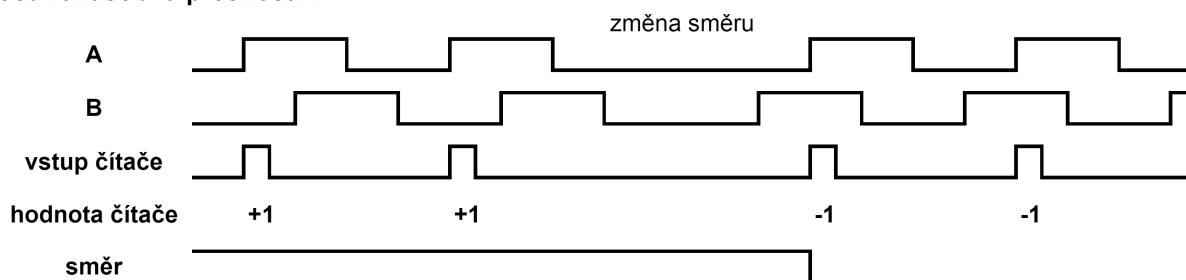
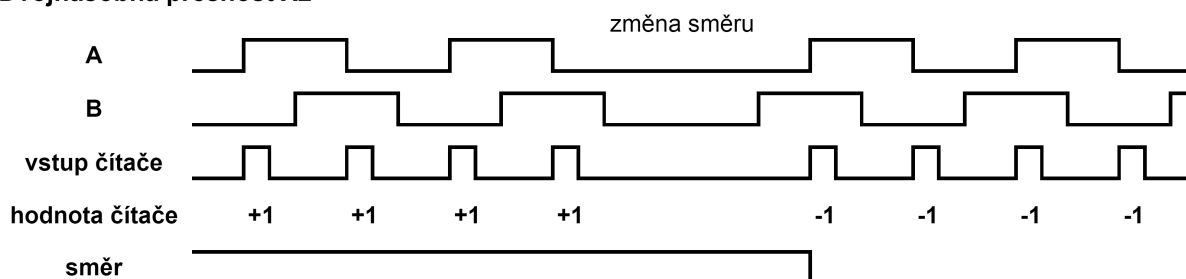
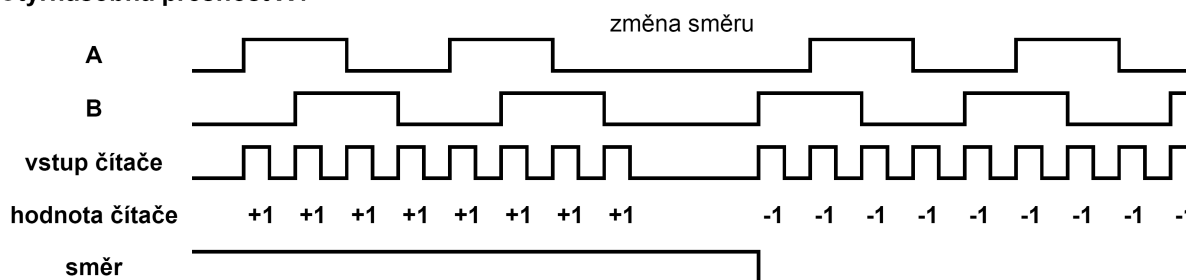
$$\beta = \frac{cnt}{x \cdot N} \cdot 360 \text{ [}^\circ\text{]} \quad (2.2)$$

kde cnt je hodnota čítače, N je počet vygenerovaných pulzů snímače za jednu otáčku a x je určeno typem kódování:

$$X1 = 1$$

$$X2 = 2$$

$$X4 = 4.$$

Jednonásobná přesnost X1**Dvojnásobná přesnost X2****Čtyřnásobná přesnost X4**

Obr. 2.4 Jednotlivé způsoby dekodování kvadrurních signálů (logická 1 signálu *směr* značí pravotočivý směr a logická 0 značí levotočivý směr)

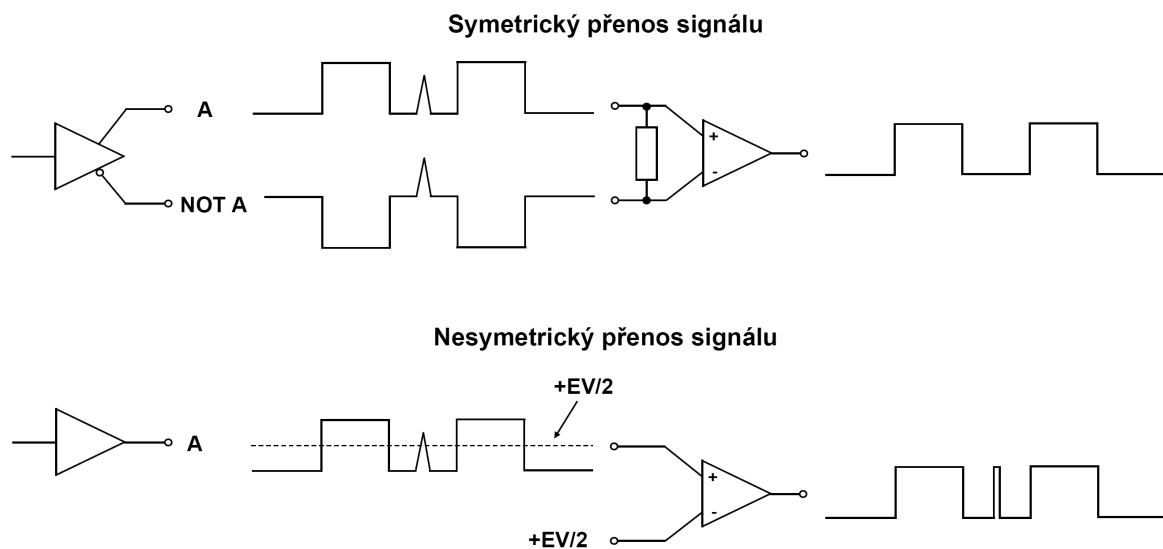
2.1.4. Připojení snímačů

Pro správné vyhodnocení signálů inkrementálního rotačního snímače, musíme snímač správně připojit k vyhodnocovací jednotce. Impedance kabelu, IRC snímače a vyhodnocovací jednotky musí být totožná. Snažíme se dosáhnout toho, aby se kabel jevil jako ohmická zátěž a nedocházelo v něm k odrazům.

Pokud je vstupní impedance přijímače vyhodnocovací jednotky větší než impedance kabelu, chová se kabel jako kapacitní zátěž. To způsobuje deformaci a prodloužení náběžných a sestupných hran signálů. Velikost deformace doby náběhu a doběhu určuje schopnost výstupního členu snímače vybit a nabít kapacitu kabelu. Kapacita kabelu je závislá na jeho délce a typu. Obvyklá kapacita kabelu je například 100 pF/m.

Výstupní signály IRC snímače jsou v kabelu vedeny rovnoběžně v žilách a dochází k vzájemnému rušení vlivem elektromagnetické indukce. Toto rušení lze snížit tím, že použijeme kabel, který má žíly spletené po párech.

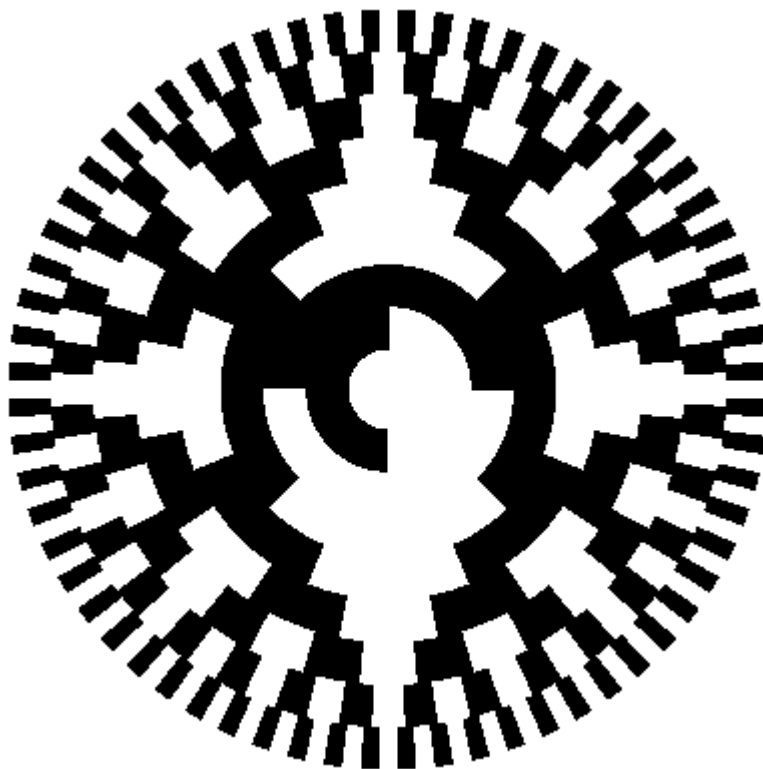
V praxi se setkáváme se dvěma způsoby přenosu signálu, symetrickým a nesymetrickým. U symetrického přenosu se přenáší přímý signál a jeho negace. V přijímači se potom vyhodnocuje rozdíl těchto signálů. Takovýmto způsobem přenosu výrazně potlačíme vliv rušení. Nesymetrický přenos využívá pouze přenos přímého signálu. Stačí nám tedy kabel s menším počtem žil a přijímač s menším počtem vstupů, ale odolnost vůči rušení je mnohem horší [2]. Symetrický a nesymetrický přenos je znázorněn na *obr. 2.5*.



Obr. 2.5 Způsoby přenosu signálu

2.2. Absolutní rotační snímače

Optický rotační absolutní snímač má polohu přímo zakódovanou v uspořádání rysek na pohyblivém kotouči. Nejvýznamnější bit (MSB) je zakódován u středu kotouče a nejméně významný bit (LSB) je zakódován na obvodu kotouče. Poloha je kódovaná v Greyově kódu, kdy se vždy mění pouze jeden bit. Absolutní snímač umožňuje okamžité vyhodnocení absolutní pozice hřídele po připojení napájení. To je výhoda oproti inkrementálnímu snímači, u kterého musíme čekat na nulový puls, abychom určili absolutní pozici. Příklad pohyblivého kotouče absolutního rotačního snímače je na *obr. 2.6*.



Obr. 2.6 Pohyblivý kotouč absolutního optického rotačního snímače [6]

2.3. Měření periody a frekvence

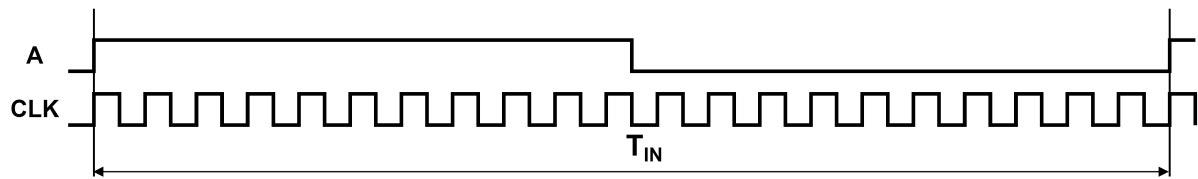
Rychlost otáčení hřídele zjistíme měřením periody nebo frekvence pulzů. Informace do této kapitoly jsem získal z [7].

2.3.1. Měření periody

Perioda T_{IN} vstupního signálu je rovna obrácené hodnotě frekvence signálu f_{IN} :

$$T_{IN} = \frac{1}{f_{IN}} \quad (2.3)$$

Měření periody neznámého signálu se provede čítáním hodinových pulsů po dobu periody měřeného signálu. Měření můžeme provést pro několik period za sebou, výsledná perioda se potom určí jako průměrná hodnota naměřených period. Měření periody umožňuje přesnější měření signálů o nízké frekvenci než měření frekvence. Měření periody je znázorněno na obr. 2.7.



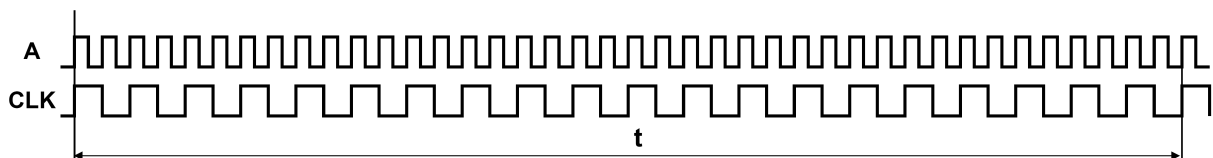
Obr. 2.7 Měření periody

2.3.2. Měření frekvence

Frekvence signálu může být definovaná počtem pulsů měřeného signálu za daný časový interval:

$$f = \frac{n}{t} \quad (2.4)$$

kde n je počet pulsů a t je časový interval v kterém jsme pulsy počítali. Měření frekvence je přesnější pro měření signálů o vysoké frekvenci než měření periody. Měření frekvence je znázorněno na obr. 2.8.



Obr. 2.8 Měření frekvence

2.3.3. Chyby měření periody a frekvence

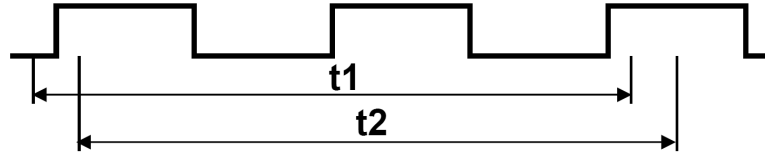
Hlavní zdroje chyb měření frekvence a periody jsou:

- kvantizační chyba (*count_error*),
- chyba frekvence oscilátoru (*time_base_error*),
- chyba spouštění (*trigger_error*).

Kvantizační chyba (*count_error*) vzniká na nejméně významném bitu (LSB) a je způsobená tím, že vstupní měřený signál je asynchronní s hodinovým kmitočtem. Vznik kvantizační chyby je naznačen na obr. 2.9. Časové intervaly $t1$ a $t2$ představují intervaly měření. Interval měření je při měření periody určen periodou měřeného signálu a při měření

frekvence je odvozen od daného počtu hodinových pulsů. V časovém intervalu $t1$ čítač napočítá 3 pulsy a v časovém intervalu $t2$ čítač napočítá pouze 2 pulsy.

vstup čítače pulsů



Obr. 2.9 Kvantizační chyba (při měření periody je na vstup čítače přiveden hodinový signál a při měření frekvence je na vstup čítače připojen měřený signál)

Chyba frekvence oscilátoru (*time_base_error*) je způsobena odchylkou skutečné frekvence od nominální frekvence.

Chyba spouštění (*trigger_error*) se projevuje pouze u měření periody a je to náhodná chyba, která je zapříčiněna šumem na měřeném signálu a šumem na vstupu čítače. Při měření periody je interval čítání pulsů řízen měřeným signálem. Šum způsobí nepřesný začátek a konec měřicího intervalu. To má za následek náhodnou časovou chybu.

Chybu měření periody můžeme definovat jako součet kvantizační chyby, chyby frekvence oscilátoru a chyby spouštění. Relativní kvantizační chyba při měření periody je:

$$count_error = \pm \frac{t_c}{T_{IN}} \quad (2.5)$$

kde t_c je perioda hodinového signálu a T_{IN} je perioda měřeného signálu.

Relativní chyba spouštění při měření periody je:

$$trigger_error = \frac{1.4 \cdot \sqrt{x^2 + e_n^2}}{\Delta V / \Delta T} \quad (2.6)$$

kde x je šum na vstupu čítače, e_n je efektivní hodnota šumu v měřeném signálu (měřeno přes šířku pásma čítače) a $\Delta V / \Delta T$ je rychlost přeběhu.

Při měření periody vzniklou relativní kvantizační chybu a relativní chybu spouštění můžeme snížit měřením přes několik period signálu a následným vypočítáním aritmetického průměru.

Chyba způsobená měřením přes několik period měřeného signálu je následující:

$$\pm \frac{\text{count_error}}{m} \pm \frac{\text{trigger_error}}{m} \pm \text{time_base_error} \quad (2.7)$$

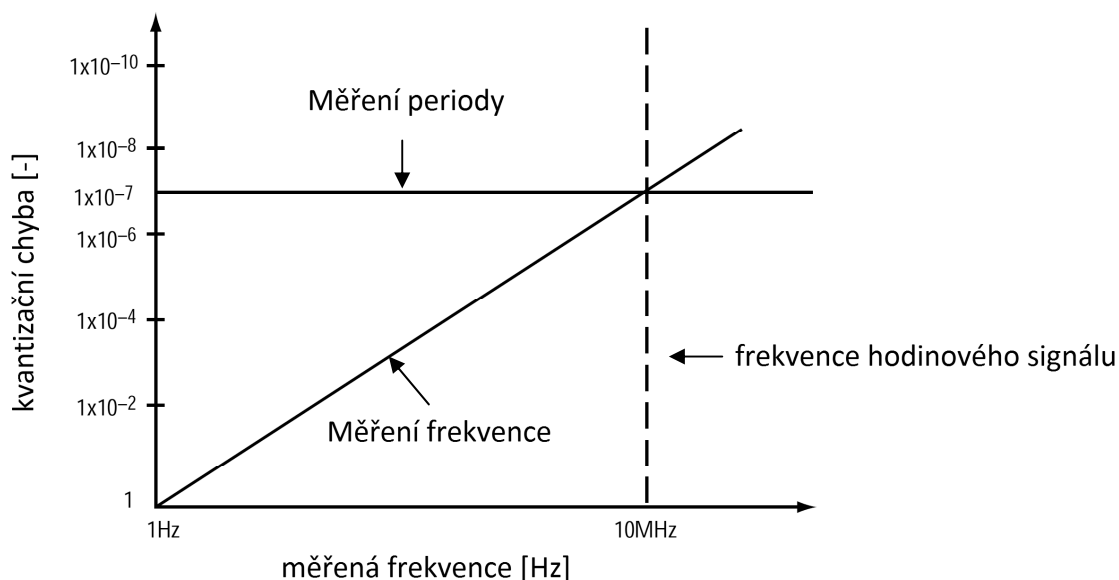
kde m je počet period měřeného signálu přes které proběhlo měření.

Chybu měření frekvence můžeme definovat jako součet kvantizační chyby a chyby frekvence oscilátoru. Relativní kvantizační chyba při měření frekvence je:

$$\text{count_error} = \pm \frac{1}{f_{IN}} \quad (2.8)$$

kde f_{IN} je měřená frekvence.

Velikost relativní kvantizační chyby při měření frekvence je nepřímo úměrná vstupní měřené frekvenci f_{IN} . Při měření periody je relativní kvantizační chyba konstantní a je určena periodou hodinového signálu t_C . Grafické porovnání velikostí relativních kvantizačních chyb metody měření periody a metody měření frekvence je znázorněno na obr. 2.10.



Obr. 2.10 Grafické porovnání velikostí relativních kvantizačních chyb metody měření periody a metody měření frekvence (frekvence hodinového signálu je 10 MHz) [7]

Pro měřené frekvence nižší než frekvence hodinového signálu je relativní kvantizační chyba při měření periody vždy menší než relativní kvantizační chyba vzniklá při měření frekvence pro daný interval měření. Zanedbáme-li chybu spouštění (*trigger_error*) a chybu

frekvence oscilátoru (*time_base_error*), můžeme tvrdit, že měření periody má vždy vyšší rozlišení než odpovídající měření frekvence pro všechny měřené frekvence nižší než frekvence hodinového oscilátoru.

Pro měřené frekvence vyšší než frekvence hodinového signálu výše uvedené zlepšení rozlišení neplatí. Relativní kvantizační chyba měření periody je větší než relativní kvantizační chyba měření frekvence pro měřené frekvence vyšší než frekvence hodinového oscilátoru.

Měříme-li například kmitočty o frekvenci $f_{IN} = 10$ Hz, relativní kvantizační chyba při měření frekvence je:

$$count_error = \pm \frac{1}{f_{IN}} = \pm \frac{1}{10} = \pm 10^{-1} \quad (2.9)$$

Relativní kvantizační chyba měření periody při hodinovém kmitočtu 10 MHz (perioda $t_C = 10^{-7}$ s) a intervalem měření $T_{IN} = 0,1$ s (frekvence měřeného signálu $f_{IN} = 10$ Hz) je:

$$count_error = \pm \frac{t_C}{T_{IN}} = \pm \frac{10^{-7}}{0,1} = \pm 10^{-6} \quad (2.10)$$

V uvedeném příkladu je relativní kvantizační chyba při měření periody o pět řádů nižší.

2.4. ISA sběrnice

Tato kapitola popisuje ISA sběrnici, její signály a sběrníkový cyklus pro 16 bitový přístup do paměťového prostoru.

2.4.1. Architektura ISA

Sběrnice ISA (Industry Standard Architecture) je standard počítačové sběrnice pro počítače, které jsou kompatibilní s původními počítači IBM PC, PC XT a PC AT. Sběrnice byla původně určena pro počítače s 8 bitovou vnější sběrnici s procesorem Intel 8086 (PC XT) a později rozšířena pro počítače s 16 bitovým procesorem Intel 80286 (PC AT). Typ ISA sběrnice pro 8 bitový počítač PC XT nazveme jako PC/XT sběrnice a typ ISA sběrnice pro 16 bitový počítač PC AT nazveme jako PC/AT sběrnice.

PC/XT sběrnice obsahuje celkem 62 vodičů, mezi které patří 8 datových, 20 adresových, napájecí $\pm 5V$ a $\pm 12V$, hodinové, řídicí vodiče zápisu a čtení, přerušení, atd. Konektor PC/XT sběrnice je znázorněn na *obr. 2.11*.

GND	B01	A01	CHCHK#
RESET DRV	B02	A02	SD7
+5 V	B03	A03	SD6
IRQ9	B04	A04	SD5
-5 V	B05	A05	SD4
DRQ2	B06	A06	SD3
-12 V	B07	A07	SD2
NOWS#	B08	A08	SD1
+12 V	B09	A09	SD0
GND	B10	A10	CHRDY
SMWTC#	B11	A11	AEN
SMRDC#	B12	A12	SA19
IOWC#	B13	A13	SA18
IORC#	B14	A14	SA17
DAK3#	B15	A15	SA16
DRQ3	B16	A16	SA15
DAK1#	B17	A17	SA14
DRQ1	B18	A18	SA13
REFRESH#	B19	A19	SA12
BCLK	B20	A20	SA11
IRQ7	B21	A21	SA10
IRQ6	B22	A22	SA9
IRQ5	B23	A23	SA8
IRQ4	B24	A24	SA7
IRQ3	B25	A25	SA6
DAK2#	B26	A26	SA5
TC	B27	A27	SA4
BALE	B28	A28	SA3
+5 V	B29	A29	SA2
OSC	B30	A30	SA1
GND	B31	A31	SA0

Obr. 2.11 Pohled shora na konektor 8 bitové PC/XT sběrnice [10]

PC/AT sběrnice rozšiřuje sběrnici typu PC/XT tak, že se vedle konektoru 8 bitové sběrnice PC/XT přidá druhý kratší konektor s 36 vodiči, který je znázorněn na *obr. 2.12*. Konektor 8 bitové sběrnice PC/XT je tedy nezměněn a je tak umožněna zpětná kompatibilita s 8 bitovými kartami. Druhý konektor přidá 4 adresové vodiče a 8 datových vodičů. Celkový počet adresových vodičů se tak navýší na 24 a celkový počet datových vodičů na 16. Také se přidají vstupy žádostí o přerušení, čtyři 16 bitové DMA kanály a řídicí signály pro výběr 8bitového nebo 16bitového přenosu. PC/AT sběrnice má dohromady 98 vodičů.

M16#	D01	C01	SBHE#
IO16#	D02	C02	LA23
IRQ10	D03	C03	LA22
IRQ11	D04	C04	LA21
IRQ12	D05	C05	LA20
IRQ15	D06	C06	LA19
IRQ14	D07	C07	LA18
DAK0#	D08	C08	LA17
DRQ0	D09	C09	MRDC#
DAK5#	D10	C10	MWTC#
DRQ5	D11	C11	SD8
DAK6#	D12	C12	SD9
DRQ6	D13	C13	SD10
DAK7#	D14	C14	SD11
DRQ7	D15	C15	SD12
+5 V	D16	C16	SD13
MASTER16#	D17	C17	SD14
GND	D18	C18	SD15

Obr. 2.12 Pohled shora na přidaný druhý konektor PC/AT sběrnice [10]

ISA sběrnice byla dále vylepšena pro použití s 32 bitovými procesory a dostala označení EISA (Extended Industry Standard Architecture). Zabudovaný počítač PC104 využívá ke komunikaci s ostatními moduly odvozenou strukturu PC/AT sběrnice [8].

2.4.2. Signály ISA

V této kapitole jsou popsány signály 16 bitové PC/AT sběrnice, která zahrnuje všechny signály 8 bitové PC/XT sběrnice. Poznatky jsem získal z [9] a [10].

Adresa je vystavena při zahájení sběrnicového cyklu během adresovacího intervalu T_s , je 24 bitová a je viditelná všem nainstalovaným ISA kartám a logice na systémové desce. Současný master na sběrnici tak může komunikovat s jakýmkoli paměťovým nebo I/O slavem. Popis signálů pro adresování je v *tab. 2.1*. Znak # za signálem znamená, že signál je aktivní v logické nule.

Tab. 2.1. Signály pro adresování ISA.

Signál	Funkce
SA[19:0]	System Address Bus. SA obsahuje spodních 20 bitů adresy, které jsou zachycené ve vyrovnávací paměti systémové desky. Adresa se pro sběrniceový cyklus na sběrnici SA objeví při náběžné hraně signálu BALE a je zachycena při sestupné hraně signálu BALE.
LA[23:17]	Latchable Address Bus. Nezachycované horní bity adresy. Horní 4 bity sběrnice LA [23:20] jsou horní 4 bity adresy. Spodní tři bity LA[19:17] jsou totožné s horními třemi bity sběrnice SA[19:17]. Adresa LA se na sběrnici objeví před začátkem sběrniceového cyklu (během předchozího sběrniceového cyklu). To umožňuje dřívější dekódování adresy, adresované zařízení tak bude připravené dokončit sběrniceový cyklus dříve než kdyby se signál chipselect dekoval ze sběrnice SA.
SBHE#	System Bus High Enable. Signalizuje použití horních 8 bitů datové sběrnice SD[15:8]. Signál je aktivní v logické nule.

Datová sběrnice se skládá ze dvou 8bitových sběrnic: SD[7:0] a SD[15:8]. Data jsou přenášena během intervalu T_c . Datové sběrnice jsou popsány v tab. 2.2.

Tab. 2.2. Datové sběrnice ISA.

Signál	Funkce
SD[7:0]	System Data Bus. Spodních 8 bitů datové sběrnice. Při přenosu dat s 16 bitovým zařízením je spodních 8 bitů dat přenášeno mezi současným masterem a sudou adresou. Při přenosu dat s 8 bitovým zařízením je spodních 8 bitů přenášeno mezi současným masterem a sudou nebo lichou adresou. Sběrnice spodních 8 bitů datové sběrnice je dostupná pro všechny 8 bitové a 16 bitové ISA karty.
SD[15:8]	System Data Bus. Horních 8 bitů datové sběrnice. Při přenosu dat s 16 bitovým zařízením je horních 8 bitů dat přenášeno mezi současným masterem a lichou adresou. Při přenosu dat s 8 bitovým zařízením horních 8 bitů není přenášeno. Sběrnice horních 8 bitů je dostupná pouze u 16 bitových ISA karet.

Systémová deska s mikroprocesorem musí při zahájení sběrnicevého cyklu definovat typ sběrnicevého cyklu. Typ sběrnicevého cyklu se určí pomocí řídicích signálů během intervalu T_s . Popis řídicích signálů je v *tab. 2.3*.

Tab. 2.3. Řídící signály ISA.

Signál	Funkce
SMRDC#	System Memory Read Command. Signál se nastaví logikou systémové desky během sběrnicevého cyklu pro čtení z paměťového prostoru, který adresuje první MB paměti (000000h až 0FFFFFFh). Tento signál je připojen k 8 bitové části všech 8 bitových a 16 bitových ISA karet. 8 bitové paměťové karty by měly tento řídicí signál použít jako signál pro povolení dekódování adresy na adresové sběrnici. Tím zabráníme vybrání 8 bitové paměťové karty, když je na adresové sběrnici adresa větší než 1 MB.
SMWTC#	System Memory Write Command. Signál se nastaví logikou systémové desky během sběrnicevého cyklu pro zápis do paměťového prostoru, který adresuje první MB paměti (000000h až 0FFFFFFh). Tento signál je připojen k 8 bitové části všech 8 bitových a 16 bitových ISA karet. 8 bitové paměťové karty by měly tento řídicí signál použít jako signál pro povolení dekódování adresy na adresové sběrnici. Tím zabráníme vybrání 8 bitové paměťové karty, když je na adresové sběrnici adresa větší než 1 MB.
IOWC#	Channel Ready. Tento signál se nastaví logikou systémové desky během sběrnicevého cyklu pro zápis do I/O prostoru.
IORC#	I/O Read Command. Tento signál se nastaví logikou systémové desky během sběrnicevého cyklu pro čtení z I/O prostoru.
MRDC#	Memory Read Command. Signál se nastaví logikou systémové desky během sběrnicevého cyklu pro čtení z paměťového prostoru. 16 bitové paměťové karty by měly tento řídicí signál použít jako signál pro povolení dekódování adresy na adresové sběrnici.
MWTC#	Memory Write Command. Signál se nastaví logikou systémové desky během sběrnicevého cyklu pro zápis do paměťového prostoru. 16 bitové paměťové karty by měly tento řídicí signál použít jako signál pro povolení dekódování adresy na adresové sběrnici.

Pro zápis do I/O prostoru musí systémová deska s mikroprocesorem vygenerovat signál IOWC# a pro čtení z I/O prostoru musí vygenerovat signál IORC#. Pro zápis do paměťového prostoru, systémová deska s mikroprocesorem nastaví signál MWTC# nebo SMWTC# a pro čtení z paměťového prostoru nastaví signál MRDC# nebo SMRDC#.

Časování ISA sběrniceového cyklu je odvozeno od BCLK. Popis signálů pro časování je uveden v tab. 2.4.

Tab. 2.4. Časovací signály ISA.

Signál	Funkce
BCLK	Bus Clock. Hodinový signál BCLK je odvozen od hodinového kmitočtu systémové desky. IBM počítač PC AT byl založen na 16 bitovém mikroprocesoru 80286 pracujícím na kmitočtu 8 MHz. V pozdějších ISA zařízeních je frekvence BCLK obvykle 8,33 MHz. Tato frekvence začala být používána od uvedení prvních 25 MHz mikroprocesorů 80386. Střída signálu je 1:1.
BALE	Buffered Address Latch Enabled. Při náběžné hraně signálu BALE se na sběrnici SA vystaví spodních 20 bitů adresy. Při sestupné hraně signálu BALE se adresa SA zachytí ve vyrovnávací paměti systémové desky. 16 bitová ISA paměťová karta by měla při sestupné hraně BALE zachytit dekodovanou adresu LA[23:17] a podržet ji po celou dobu sběrniceového cyklu.
CHRDY	Channel Ready. Rozšiřující ISA karta nastaví signál CHRDY na log. 0, pokud vyžaduje více čekacích intervalů, než kolik je obvykle dodáváno standardním časovačem na systémové desce. Signál CHRDY se nastaví, pokud adresový dekodér rozpozná, že je karta adresována a současně obdrží jeden z řídicích signálů SMRDC#, SMWTC#, MRDC# nebo MWTC#. Čekací takty T_c nesmí trvat déle, než 2,5 μ s. Je-li CHRDY v log. 1, ISA karta je připravená.
NOWS#	No Wait State. Rozšiřující paměťová ISA karta nastaví signál NOWS# na log. 0, pokud nevyžaduje žádný čekací interval T_c . Signál NOWS# se nastaví, pokud adresový dekodér rozpozná, že je karta adresována a současně obdrží jeden z řídicích signálů SMRDC#, SMWTC#, MRDC# nebo MWTC#. Signál NOWS# můžeme použít pro zkrácení doby přístupu k 8 bitové kartě, u které základní 4 čekací takty T_c snížíme na jeden. Přístup k 16 bitové paměťové kartě můžeme zkrátit z jednoho čekacího intervalu T_c na žádný. Přístup k 16 bitové I/O kartě nemůžeme zkrátit, vždy musí být minimálně jeden čekací interval T_c .

Jestliže adresový dekodér 16 bitové ISA karty rozpozná, že je daná karta adresována, musí karta nastavit signál M16# nebo IO16# do log. 0. Těmito signály karta signalizuje, že master komunikuje s 16 bitovým slavem a data se budou přenášet po obou datových sběrnících SD[7:0] i SD[15:8]. Pokud se signál M16# ani IO16# nenastaví do log. 0, master komunikuje s 8 bitovým zařízením a data se budou přenášet po SD[7:0] sběrnici. Signály určující druh přenosu jsou popsány v tab. 2.5.

Tab. 2.5. Řídící signály 16bitové komunikace ISA.

Signál	Funkce
M16#	Memory Size 16. Tento signál generovaný 16 bitovou paměťovou ISA kartou je nastaven na log. 0 pokud karta rozpozná, že je adresována. Signál M16# značí, že se data budou přenášet přes obě datové sběrnice SD[7:0] a SD[15:8]. Pokud není nastaven signál N0WS#, časovač na systémové desce vloží jeden čekací interval T_c do sběrnicevého cyklu. Je-li karta 8 bitová, M16# se nenastaví a komunikace bude probíhat pouze po sběrnici SD[7:0].
IO16#	I/O Size 16. Tento signál generovaný 16 bitovou ISA I/O kartou je nastaven na log. 0 pokud karta rozpozná, že je adresována. Signál IO16# poskytuje stejné funkce pro 16 bitové I/O rozšiřující karty jako M16# poskytuje pro 16 bitové paměťové ISA karty. IO16# také signalizuje časovači na systémové desce, aby vložil jeden čekací interval T_c do sběrnicevého cyklu.

Popis signálů spojených s přímým přístupem do paměti (DMA) je v tab. 2.6.

Tab. 2.6. DMA signály ISA.

Signál	Funkce
DRQ[3:0] DRQ[7:5]	DMA Request. Žádosti o DMA. DRQ[3:1] jsou připojeny ke všem 8 bitovým a 16 bitovým ISA kartám, zatímco DRQ[7:5] a DRQ0 jsou připojeny pouze k 16 bitovým ISA kartám.
DAK#[3:0] DAK#[7:5]	DMA Acknowledge. Potvrzení žádosti o DMA.
TC	Transfer Complete. Ukončení přenosu kanálem DMA.
AEN	Address Enable. Signalizace platnosti adresy generované DMA. Zabrání I/O zařízením odpovídat na I/O řídicí signály.
MASTER16#	Potvrzení převzetí sběrnice 16 bitovým masterem.

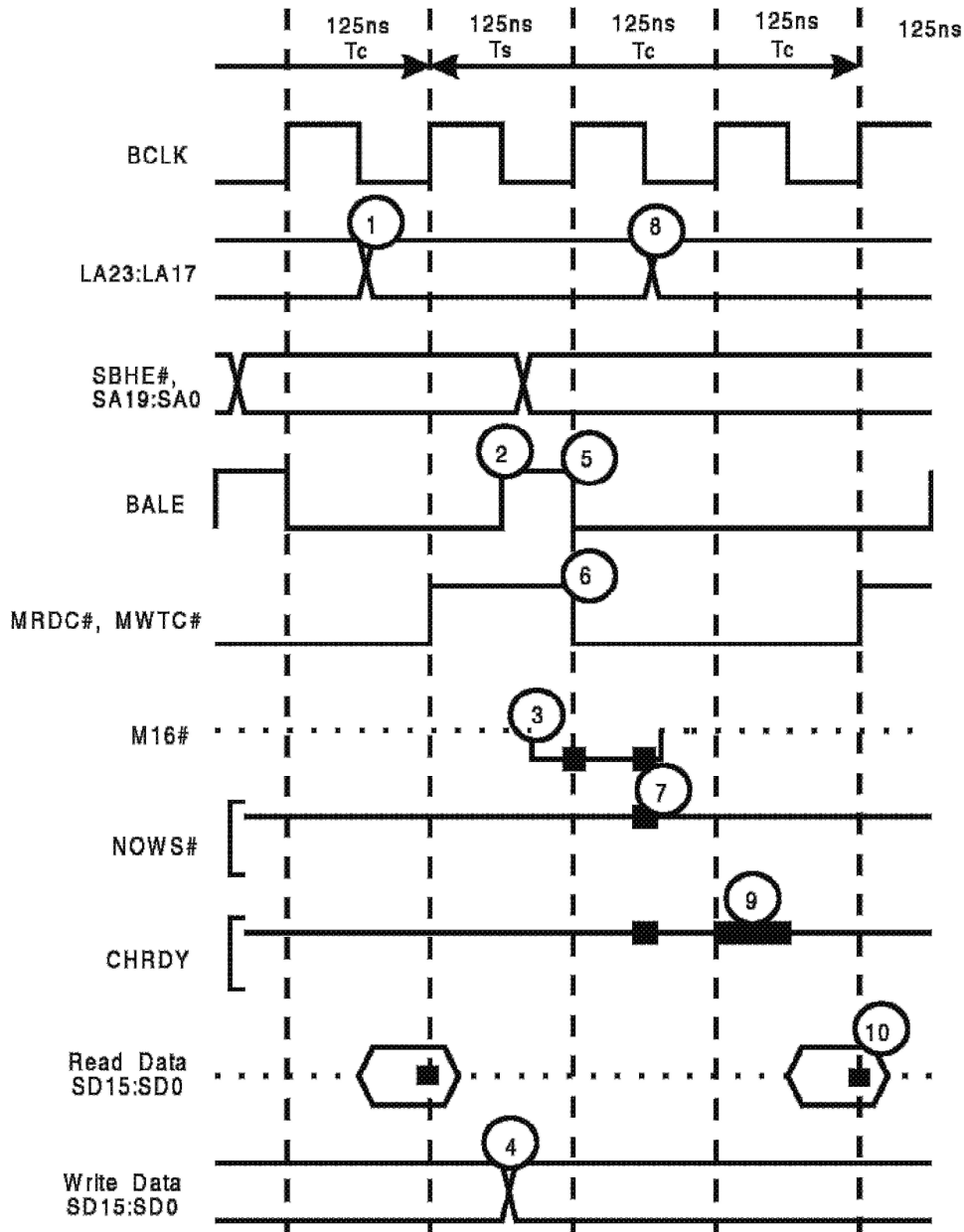
Zbývající signály ISA sběrnice jsou popsány v *tab. 2.7*

Tab. 2.7. Zbývající signály ISA.

Signál	Funkce
IRQ[15:14] IRQ[12:9] IRQ[7:3]	Interrupt Request. Žádosti o přerušení.
CHCHK#	Channel Check. Signalizace chyby na ISA kartě.
RESDRV	Reset Drive. Signál RESDRV je nastaven do log. 1 dokud se napájecí napětí po zapnutí nestabilizuje. RESDRV se také nastaví, když napájecí napětí klesne pod přípustnou mez. Nastavený signál RESDRV zakáže ISA kartám jakoukoli činnost, dokud se napětí nestabilizuje a zaručí, že všechny nainstalované karty po zapnutí začnou pracovat v definovaném stavu.
OSC	Oscillator. Hodinový signál o kmitočtu 14,31818 MHz. Střída signálu 1:1.
REFRESH#	Systémová deska signál nastaví do log. 0, když právě obnovuje dynamické paměti.

2.4.3. Paměťový 16 bitový sběrniceový cyklus ISA

Standardní paměťový 16 bitový sběrniceový cyklus se skládá z jednoho adresového intervalu T_s a dvou datových intervalů T_c , z nichž první interval je čekací. Celkem je tedy dlouhý tři periody hodinového kmitočtu BCLK. V případě, že frekvence BCLK je 8 MHz, doba trvání tohoto cyklu je 375 ns. Sběrniceový cyklus je nakreslen na *obr. 2.13* a jednotlivé kroky jsou vysvětleny v *tab. 2.8*.



Obr. 2.13 Standardní přístup do 16 bitového paměťového prostoru [10]

Tab. 2.8. Popis jednotlivých kroků standardního 16 bitového paměťového sběrnicevého cyklu.

krok	Popis
1	Horních 7 bitů adresy se objeví na LA sběrnici před začátkem sběrnicevého cyklu. To umožní dřívejší dekodování adresy.
2	Signál BALE se nastaví do log. 1 v polovině adresového intervalu T_s . Při náběžné hraně signálu BALE se na sběrnici SA vystaví spodních 20 bitů adresy. Některá 16 bitová paměťová zařízení začnou dekodovat adresu na LA sběrnici až po náběžné hraně signálu BALE.
3	Detekuje-li adresový dekodér 16 bitové paměťové ISA karty, že je karta adresována, nastaví signál M16# do log. 0. Tímto signálem systémová deska zjistí, že komunikuje s 16 bitovým zřízením.
4	Pokud se jedná o zápisový sběrnicevý cyklus, data se sběrnici SD přiřadí v polovině adresového intervalu T_s . Data na sběrnici zůstanou do poloviny adresového intervalu T_s následujícího sběrnicevého cyklu.
5	Při sestupné hraně BALE se adresa SA zachytí ve vyrovnávací paměti systémové desky. Adresa je tak adresovému dekodéru k dispozici do konce sběrnicevého cyklu. Při sestupné hraně BALE 16 bitová paměťová ISA karta zachytí adresu na LA sběrnici, protože LA adresa není k dispozici po celou dobu sběrnicevého cyklu.
6	Logika sběrnice systémové desky vzorkuje signál M16# na konci adresového intervalu T_s . Je-li signál M16# aktivní, nastaví se při začátku prvního datového intervalu T_c příslušný řídicí signál SMRDC# pro čtení nebo SMWTC# pro zápis. Tento řídicí signál zůstane aktivní až do skončení sběrnicevého cyklu.
7	Logika sběrnice systémové desky podruhé vzorkuje signál M16# v polovině datového intervalu T_c . Druhé vzorkování tak dává dodatečný čas 16 bitovému paměťovému zařízení nastavit signál M16#, pokud to nestihlo včas pro první vzorkování. Tato zařízení buď včas neobdržela LA adresu, aby nastavila M16# před prvním vzorkováním, anebo čekají na nastavení BALE signálu než začnou dekodovat LA adresu. V polovině prvního datového intervalu T_c také standardní časovač systémové desky vzorkuje signály Nows# a CHRDY. Pokud tyto signály jsou nastaveny, sběrnicevý cyklus je ukončen po prvním datovém intervalu T_c . V takovém případě, 16 bitová paměťová deska může provést sběrnicevý cyklus během dvou BCLK taktů. Pokud CHRDY není nastaveno, standardní časovač ignoruje Nows# a vkládá jeden nebo více čekacích intervalů.

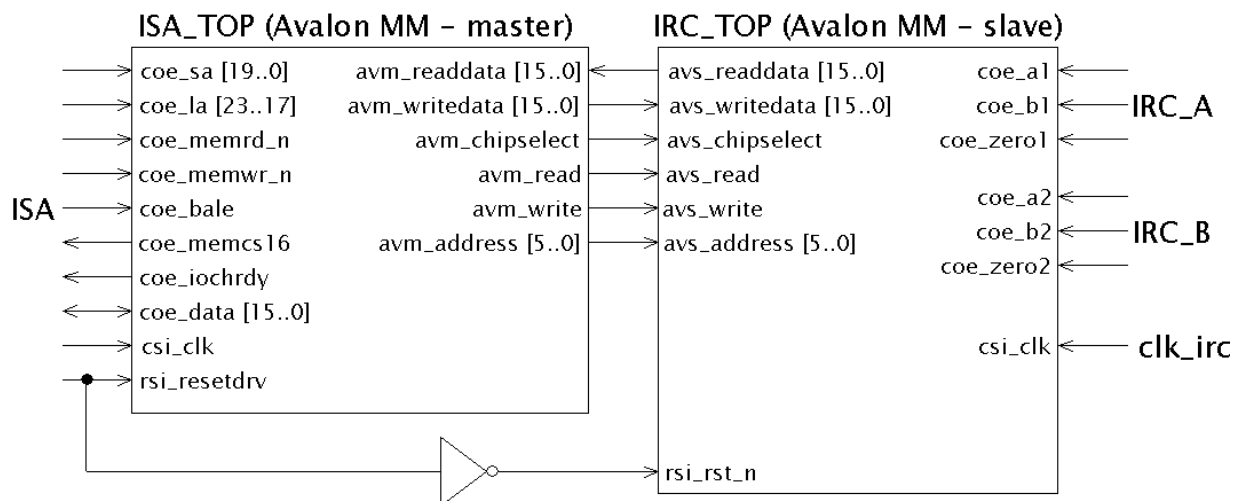
Tab. 2.8. Popis jednotlivých kroků standardního 16 bitového paměťového sběrnicevého cyklu.

Dokončení.

krok	Popis
8	Horních 7 bitů adresy pro následující sběrnicevý cyklus se může objevit na LA sběrnici už během prvního datového intervalu současného sběrnicevého cyklu. Zbytek adresy se na SA sběrnici neobjeví dříve než v polovině adresového intervalu následujícího sběrnicevého cyklu.
9	Signál CHRDY je vzorkován během druhého datového intervalu T_c . Pokud zařízení nemůže dokončit sběrnicevý cyklus v tomto hodinovém taktu, musí nastavit CHRDY do log. 0. Pokud je signál CHRDY navzorkován v log. 0, sběrnicevý cyklus se prodlouží přidáním dalšího datového (čekacího) intervalu T_c . Signál CHRDY je potom kontrolován na začátku každého přidaného datového (čekacího) intervalu T_c , jestli sběrnicevý cyklus může být dokončen.
10	Standardní sběrnicevý cyklus pro přístup do 16 bitového paměťového prostoru trvá tři hodinové takty BCLK, pokud není předčasně ukončen nastavením Nows# do log. 0, a pokud CHRDY je v log. 1. Logika sběrnice nastaví řídicí signál (MRDC#, MWTC#, SMRDC#, SMWTC#) do log. 1. Pokud se jedná o paměťový zápisový cyklus, data na datové sběrnici SD zůstanou do první poloviny adresového intervalu T_s následujícího sběrnicevého cyklu. Zařízení má tak dostatek času k zapsání dat a zároveň zařízení adresováno v následujícím sběrnicevém cyklu není ovlivněno, protože řídicí signál pro čtení/zápis není ještě aktivní.

3. Návrh a implementace

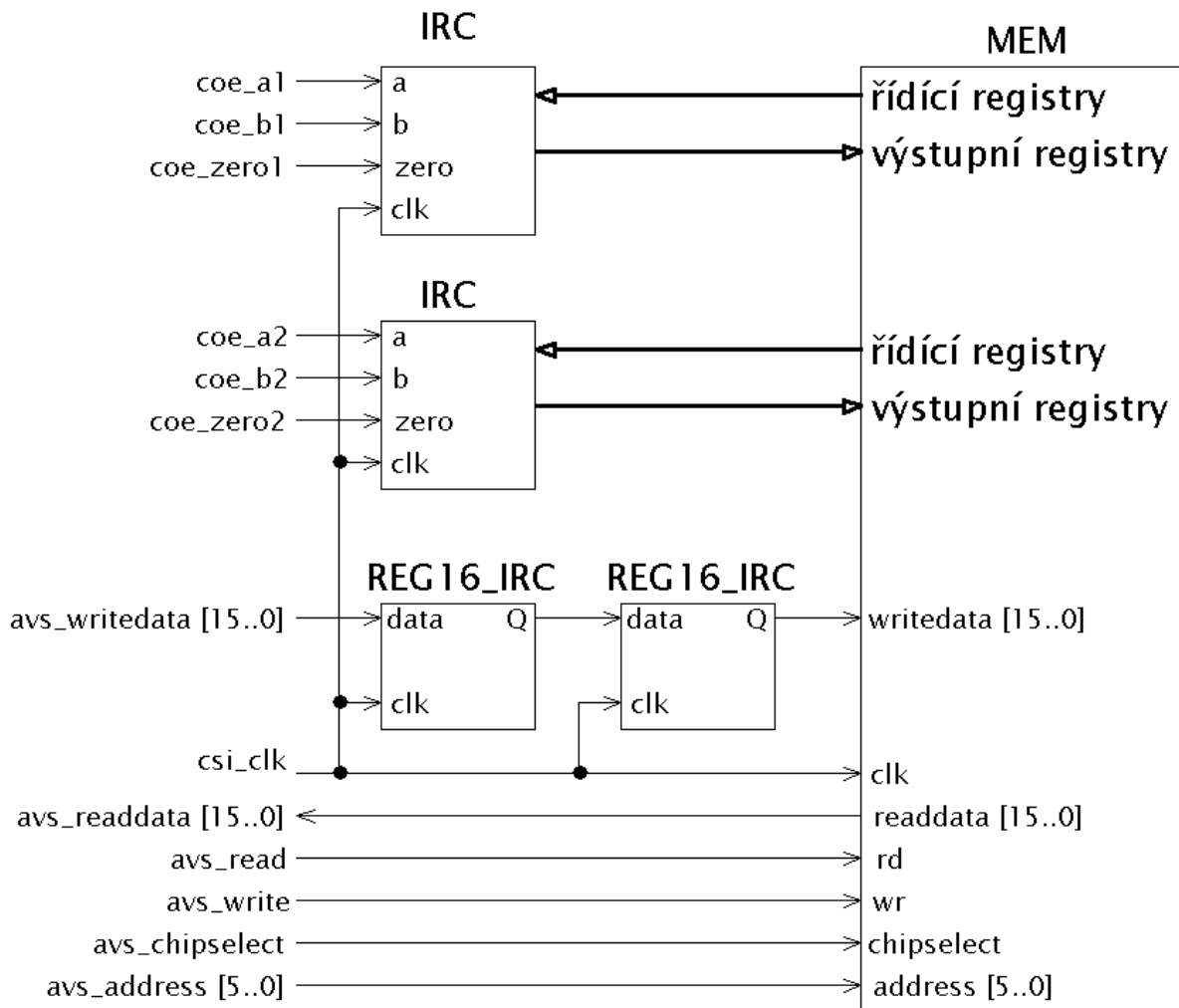
Výsledná IP funkce obsahuje top level entitu s označením *top* a obsahuje dva moduly spojené sběrnici Avalon Memory Mapped Interface (Avalon-MM): *isa_top* a *irc_top*. Modul *isa_top* je Avalon MM-master a umožňuje 16 bitový přístup do paměťového prostoru po ISA sběrnici. Modul *irc_top* je Avalon MM-slave a slouží k vyhodnocení otáček. Blokové schéma top level entity je na obr. 3.1. K modulu *irc_top* jsou připojeny 2 inkrementální rotační snímače (**IRC_A**, **IRC_B**). Vstup *csi_clk* modulu *irc_top* je připojen k oscilátoru **clk_irc**. Všechny zdrojové kódy jsou k dispozici na příloženém CD.



Obr. 3.1 Blokové schéma top level entity

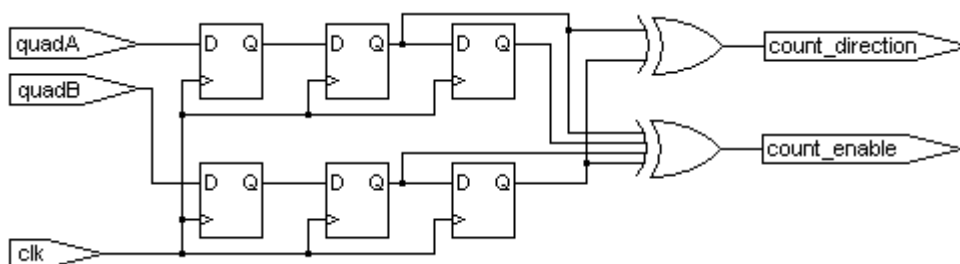
3.1. Modul vyhodnocení otáček IRC_TOP

Modul *irc_top* se skládá ze tří nejdůležitějších komponent: dvou komponent *irc*, které vyhodnocují signály ze dvou inkrementálních rotačních snímačů a komponenty *mem*, která obsahuje řídicí a výstupní 16 bitové registry pro oba snímače. Synchronizace hodinových signálů je řešena pomocí dvojitého vzorkování. Zjednodušené blokové schéma modulu *irc_top* je na obr. 3.2. Nulovací signál *rsi_rst_n* přivedený na všechny bloky je z důvodu přehlednosti na obrázku vynechán.

Obr. 3.2 Zjednodušené blokové schéma modulu *irc_top*

3.1.1. Komponenta IRC

Komponenta *irc* se skládá z bloků: *irc_decoder* a *met_rem*. Blok *irc_decoder* dekóduje kvadrurní signály s čtyřnásobnou přesností (X4), tzn. detekuje náběžnou a sestupnou hranu obou kvadrurních signálů. Obvod pro vyhodnocení otáček s čtyřnásobnou přesností je na obr. 3.3.



Obr. 3.3 Obvod pro vyhodnocení otáček s čtyřnásobnou přesností (X4) [11]

Protože kvadrurní vstupní signály jsou vůči hodinovému signálu asynchronní, v obvodu jsou na každém vstupu přidány za sebou dva D klopné obvody, abychom zabránili metastabilitě. Tyto D klopné obvody jsou v bloku *met_rem*. Výstupní signál *count_enable* slouží jako hodinový vstup čítače. Čítač podle hodnoty signálu *count_direction* čítá směrem nahoru nebo dolů. Je-li *count_direction* v log. 0, jedná se o levotočivý směr a čítač čítá dolů. Pokud je *count_direction* v log. 1, směr je pravotočivý a čítač čítá nahoru. Pokud máme k dispozici nulový signál, můžeme z hodnoty čítače zjistit absolutní polohu hřídele. Rychlost otáčení zjistíme počítáním hodinových impulsů po dobu periody kvadrurních impulsů (měření periody) a nebo počítáním kvadrurních impulsů za daný časový interval (měření frekvence). VHDL kód popisující obvod pro vyhodnocení otáček vypadá takto:

```

process(clk) -- D signal a
begin
  if(clk'event and clk='1') then
    sqa<=a;
  end if;
end process;

process(clk) -- D signal b
begin
  if(clk'event and clk='1') then
    sqb<=b;
  end if;
end process;

process(a,sqb) -- xor
begin
  scout_direction<=a xor sqb;
end process;

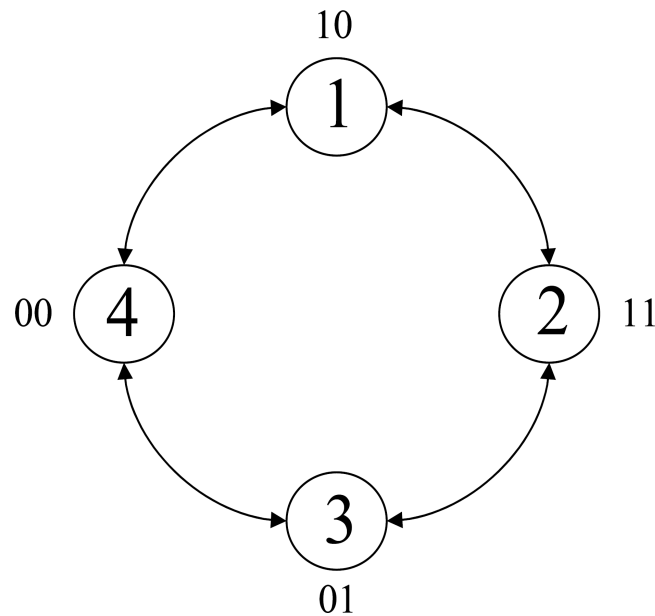
process(a,sqa,b,sqb,sa,sb) -- xor
begin
  if(sb='1' and sa='1')then
    scout_enable<=a xor sqa xor b xor sqb;
  elsif(sb='0' and sa='1')then
    scout_enable<=a xor sqa;
  elsif(sb='1' and sa='0')then
    scout_enable<=b xor sqb;
  end if;
end process;

```

Je-li kvadrurní signál **A** k dispozici, signál *sa* je roven log. 1. Pokud je kvadrurní signál **B** k dispozici, signál *sb* je roven log. 1. Kvadrurní signály **A**, **B** jsou ve VHDL kódu označeny *a*, *b*. Signál *count_enable* je v kódu značen *scount_enable*, signál *count_direction* je v kódu značen *scount_direction*.

Výše uvedený obvod po připojení čítače můžeme popsat konečným automatem (Finite State Machine – FSM), jehož stavový diagram je na *obr. 3.4*. Stavů 00, 01, 11, 10 jsou

hodnoty čítače. Podle směru otáčení hřídele čítač inkrementuje nebo dekrementuje svoji hodnotu.



Obr. 3.4 Stavový diagram dekodéru otáček

Detekce výpadku signálu je řešena pomocí maximální dovolené derivace. Změní-li se množství napočítaných pulsů měření periody (registry: **RPMR1HA**, **RPMR1LA** respektive **RPMR1HB**, **RPMR1LB**) nebo měření frekvence (registry: **RPMR2HA**, **RPMR2LA** respektive **RPMR2HB**, **RPMR2LB**) o hodnotu větší než uvedenou v řídicím registru **DER1HA**, **DER1LA** respektive **DER2HA**, **DER2LA**, nastaví se příslušný příznak v chybovém registru **ER**.

Výpis VHDL kódu, který převádí proměnnou *cnt2* (čítač hodinových pulsů) na signál *rpmr1* (měření periody) a který detekuje výpadek signálu je následující:

```

if(cnt1=0) then
  cnt2:=0;
  cnt5:=0;
  cnt6<=X"00000001";
  pos:=0;
else
  if(cnt1=c1) then -- 1 rotation (4x decoding mode)
    if dir=1 then
      rpmr1<=std_logic_vector(to_signed(cnt2-1,32));
    else
      rpmr1<=std_logic_vector(to_signed(-(cnt2-1),32));
    end if;
    if(((cnt2-old_cnt2)>to_integer(unsigned(der1))) or ((old_cnt2-cnt2)>to_integer(unsigned(der1))))then
      err1<='1';
    end if;
    old_cnt2:=cnt2;
    cnt1:=1;
    cnt2:=1;
  end if;
end if;
  
```

(3.2)

Tento kód je v procesu, který se spouští při každé náběžné hraně hodinového signálu. Proměnná *cnt1* je hodnota čítače. Čítač čítá při náběžné a sestupné hraně obou kvadraturních signálů. Proměnná *c1* je rovna 3 nebo 5, podle toho, jestli jsou k dispozici jeden nebo dva kvadraturní signály. To nám zaručí, že se do signálu *rpmr1* uloží data po každé periodě kvadraturního signálu. Proměnná *dir* značí směr otáčení. Signál *rpmr1* poté rozdělíme na dva signály *rpmr1h* a *rpmr1l*, které se zapíšou do výstupních registrů **RPMR1HA** a **RPMR1LA**, respektive **RPMR1HB** a **RPMR1LB**:

```
rpmr1h<=rpmr1(31 downto 16);
rpmr1l<=rpmr1(15 downto 0);
```

(3.3)

Proměnná *cnt2* obsahuje počet hodinových pulsů napočítaných během jedné periody kvadraturního signálu a v proměnné *old_cnt2* je uložena hodnota *cnt2* napočítaná v předchozí periodě kvadraturního signálu. Absolutní hodnota rozdílu *cnt2* a *old_cnt2* je porovnávána se signálem *der1*, který obsahuje maximální povolenou hodnotu, o kterou se *cnt2* za periodu může změnit. Signál *der1* získáme ze signálů *der1h* a *der1l*. Tyto signály obsahují data řídicích registrů **DER1HA**, **DER1LA** respektive **DER1HB**, **DER1LB**:

```
der1(31 downto 16)<=der1h;
der1(15 downto 0)<=der1l;
```

(3.4)

Signál *err1* značí výpadek signálu a nastaví příslušný příznak v chybovém registru **ER**. Stejný postup řešení detekce výpadku kvadraturního signálu je použit při měření frekvence.

3.1.2. Komponenta MEM

Komponenta *mem* obsahuje 16 bitové registry. Řídicích registrů je 15 a výstupních registrů je 19. Do řídicích registrů lze zapisovat i číst, výstupní registry jsou pouze pro čtení. Protože frekvence hodinového signálu ISA sběrnice je odlišná od frekvence hodinového signálu modulu *irc_top*, data určená pro zápis do řídicích registrů jsou dvojitě vzorkována z důvodu odstranění metastability. Poslední písmeno (A nebo B) v názvu registru určuje příslušnost snímači. První snímač je označen písmenem A, druhý písmenem B. U některých registrů předposlední písmeno (H nebo L) značí horních 16 bitů (H) a dolních 16 bitů (L). Seznam řídicích registrů je v příloze 1 a seznam výstupních registrů je v příloze 2.

3.2. Modul ISA_TOP

Pokud je aktivní signál *coe_memrd_n* nebo *coe_memwr_n*, adresy na vstupních datových sběrnicích *coe_sa* a *coe_la* se při sestupné hraně signálu *coe_bale* zapíší do registrů. Adresa *coe_sa* se zapíše do *reg20* a adresa *coe_la* se zapíše do *reg7*. Adresový dekodér *la_sa_test* je použit dvakrát. V prvním případě má adresové vstupy připojeny na výstupy z registrů *reg20* a *reg7* a generuje signál *avm_chipselect*. V druhém případě má adresové vstupy připojeny přímo na vstupní datové sběrnice *coe_sa* a *coe_la* a generuje přes třístavový budič *out3s* signál *coe_memcs16*. Adresa *avm_address* pro modul *irc_top* je dekodována dekodérem *addr*, jehož vstup je přiveden na výstup *reg20*.

Signál pro čtení *avm_read* je připojen na negovaný vstupní signál *coe_memrd_n* a signál pro zápis *avm_write* je připojen na negovaný vstupní signál *coe_memwr_n*. Signál *coe_iochrdy* je vygenerován přes třístavový budič *out3s*, pokud je aktivní signál *coe_memrd_n* nebo *coe_memwr_n* a pokud je zároveň vygenerován *avm_chipselect*.

Sběrnice *avm_writedata* je připojena na sběrnici *coe_data* a slouží pro přenos dat určených k zapsání do řídicích registrů modulu *irc_top*. Sběrnice *avm_readdata* obsahuje vyčtená data z řídicích a výstupních registrů modulu *irc_top* a přes dvojitý vzorkovač tvořený registry *reg16* je připojena na vstup třístavového budiče *out3s16*. Tento budič má třístavový výstup připojen na obousměrnou sběrnici *coe_data*, která slouží pro 16 bitový přenos dat po ISA sběrnici. Dvojitý vzorkovač je použit pro odstranění metastability, protože frekvence hodinového signálu ISA sběrnice je odlišná od frekvence oscilátoru modulu *irc_top*.

Všechny registry (*reg20*, *reg7* a *reg16*) mají nulovací vstup připojen na negovaný signál *rsi_resetdrv* a hodinový vstup na signál *csi_clk*.

3.3. Vyhodnocení výstupních registrů

Abychom se dostali do paměti nad 1 MB, musíme kód v C přeložit jako 32-bit .EXE za pomoci 32 bitového extenderu založeného na DPMI. K tomu můžeme použít například kompilátor Watcom, ve kterém zvolíme Targets→New Target→DOS - 32-bit→PMODE/W. Zde je uveden ukázkový kód pro čtení a zápis do paměti:

```
#include <stdio.h>
#define MEM_ADR 0xf00000u // bázová adresa, Memory hole 15MB

int main(int argc, char** argv){
    unsigned short *pamet;
    pamet=(unsigned short *)MEM_ADR;1

    printf("CREG= %04X\n",pamet[0]); // čtení z paměti
    printf("NOPRA= %04X\n",pamet[1]);

    pamet[0]=0x004d; // zápis do paměti
    pamet[1]=0x00c8;

    printf("\nCREG= %04X\n",pamet[0]); // čtení z paměti
    printf("NOPRA= %04X\n",pamet[1]);

    return(0);
}
```

Níže je příklad algoritmu pro zjištění absolutní pozice z registrů **POSRA** (resp. **POSRB**). Data z registru **POSRA** (resp. **POSRB**) jsou uloženy do proměnné *POSR*. Absolutní pozici můžeme zjistit, pouze pokud máme k dispozici nulový signál **ZERO** a oba dva kvadraturní signály **A** a **B**. Proměnná *NOPR* obsahuje počet pulsů na otáčku uložených v registru **NOPRA** (resp. **NOPRB**). Výsledný úhel natočení hřídele je uložen v proměnné *angle3*.

```
angle = ((360e3) / ((4) * (NOPR))) * POSR;
angle2 = angle / 1e3;
if angle2 > 360
    angle3 = angle2 - 360;
else
    angle3 = angle2;
end;
```

Rychlost otáčení měřením periody zjistíme z registrů **RPMR1HA** a **RPMR1LA** (resp. **RPMR1HB** a **RPMR1LB**). Proměnná *NOPR* obsahuje počet pulsů na otáčku a proměnná *RPMR1* je složena z registrů **RPMR1HA** a **RPMR1LA**. Proměnná *Tclk* [ns] je perioda hodinového signálu modulu *irc_top*. Výsledný počet otáček za minutu je uložen v proměnné *rpm1*.

```
rpm1 = (60 / Tclk) / (RPMR1 * NOPR);
```

Rychlost otáčení měřením frekvence zjistíme z registrů **RPMR2HA** a **RPMR2LA** (resp. **PMR2HB** a **RPMR2LB**). Proměnná *NOPR* obsahuje počet pulsů na otáčku a proměnná *RPMR2* je složena z registrů **RPMR2HA** a **RPMR2LA** (resp. **RPMR2HB** a **RPMR2LB**). Proměnná *Tclk* [ns] je perioda hodinového signálu modulu *irc_top* a proměnná *PERIODR* je složena z registrů **PERIODRHA** a **PERIODRLA** (resp. **PERIODRHB** a **PERIODRLB**). Proměnná *sigavail* závisí na počtu kvadraturních signálů, které jsou k dispozici. Je-li k dispozici pouze jeden kvadraturní signál, *sigavail* je rovno 2. Pokud jsou k dispozici oba dva kvadraturní signály, *sigavail* je rovno 1. Výsledný počet otáček za minutu je uložen v proměnné *rpm2*.

$$\text{rpm2} = ((\text{RPMR2} / \text{NOPR}) / (\text{PERIODR} * \text{Tclk})) * \text{sigavail} * 15; \quad (3.8)$$

Nulové otáčky zjistíme z registrů **ZEROROTHA** a **ZEROROTLA** (resp. **ZEROROTHB** a **ZEROROTLB**). Z těchto registrů složíme proměnnou *ZEROROT*. Proměnná *NOPR* obsahuje počet pulsů na otáčku a proměnná *Tclk* [ns] je perioda hodinového signálu. Počet kvadraturních signálů, které jsou k dispozici určují proměnnou *sigavail*. Je-li k dispozici pouze jeden kvadraturní signál, *sigavail* je rovno 2. Pokud jsou k dispozici oba dva kvadraturní signály, *sigavail* je rovno 1. Výsledná proměnná *rpmx* značí nulové otáčky.

```
if (ZEROROT > ((15 / (NOPR * Tclk)) * sigavail + 1))
    rpmx=0;
end; \quad (3.9)
```

3.4. Syntéza

Výsledná IP funkce byla syntetizována pro FPGA Xilinx řady Spartan-II. Výsledek syntézy v Xilinx ISE (Integrated Software Environment) je v *tab. 3.1*.

Tab. 3.1. Výsledek syntézy.

Device utilization summary:				
Selected Device : 2s200fg456-5				
Number of Slices:	1880	out of	2352	79%
Number of Slice Flip Flops:	1529	out of	4704	32%
Number of 4 input LUTs:	2969	out of	4704	63%
Number of IOs:	80			
Number of bonded IOBs:	73	out of	284	25%
Number of GCLKs:	2	out of	4	50%

4. Simulace

Simulace IP funkce jsem provedl v ModelSimu. Testbench má název *toptb.vhdl*. Testbench umožňuje simulaci 16 bitového paměťového standardního sběrnicevého cyklu ISA sběrnice pro čtení a zápis. Můžeme přistupovat k libovolnému řídicímu nebo výstupnímu 16 bitovému registru. Testbench umožňuje nastavit rychlost a směr otáčení, počet pulsů na otáčku a frekvenci oscilátoru.

4.1. Simulace ISA komunikace

Část VHDL kódu pro simulaci paměťových sběrnicevých cyklů:

```
wait until csi_clk='1';
coe_la<=B"1111_000";
wait for 50 ns;
coe_bale<='1';
wait for 20 ns;
coe_sa<=B"0000_0000_0000_0000" & X"8";
wait for 40 ns;
coe_memrd_n<='0';
coe_bale<='0';
wait for 40 ns;
coe_la<=(others=>'X');
wait for 180 ns;
coe_memrd_n<='1';
wait for 20 ns;
coe_sa<=(others=>'X');

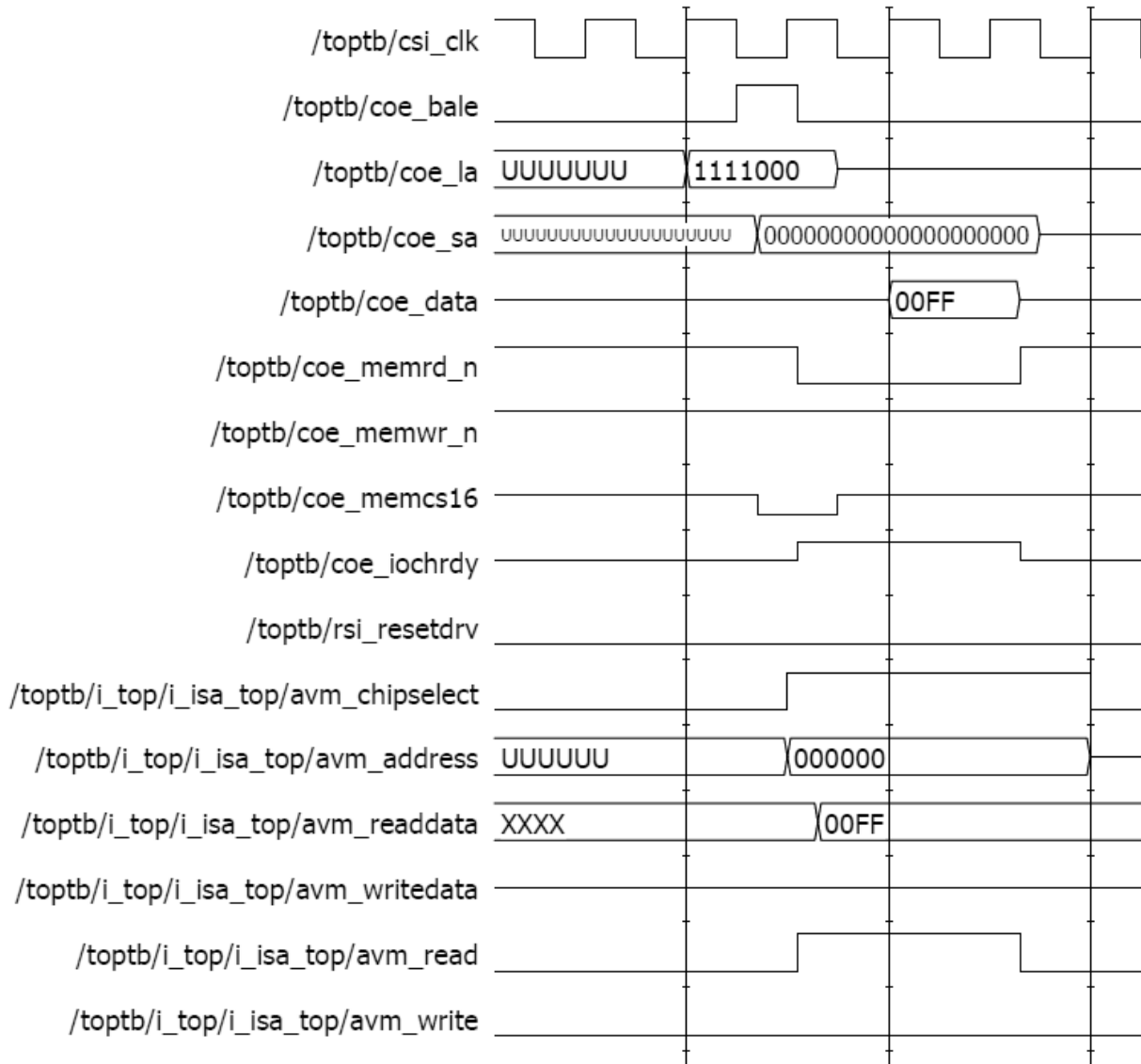
wait for 5 ms;

wait until csi_clk='1';
coe_la<=B"1111_000";
wait for 50 ns;
coe_bale<='1';
wait for 20 ns;
coe_sa<=B"0000_0000_0000_0000" & X"1";
wait for 40 ns;
coe_data<=X"00C8";
coe_memwr_n<='0';
coe_bale<='0';
wait for 40 ns;
coe_la<=(others=>'X');
wait for 180 ns;
coe_memwr_n<='1';
wait for 20 ns;
coe_sa<=(others=>'X');
wait for 10 ns;
coe_data<=(others=>'Z');
```

(4.1)

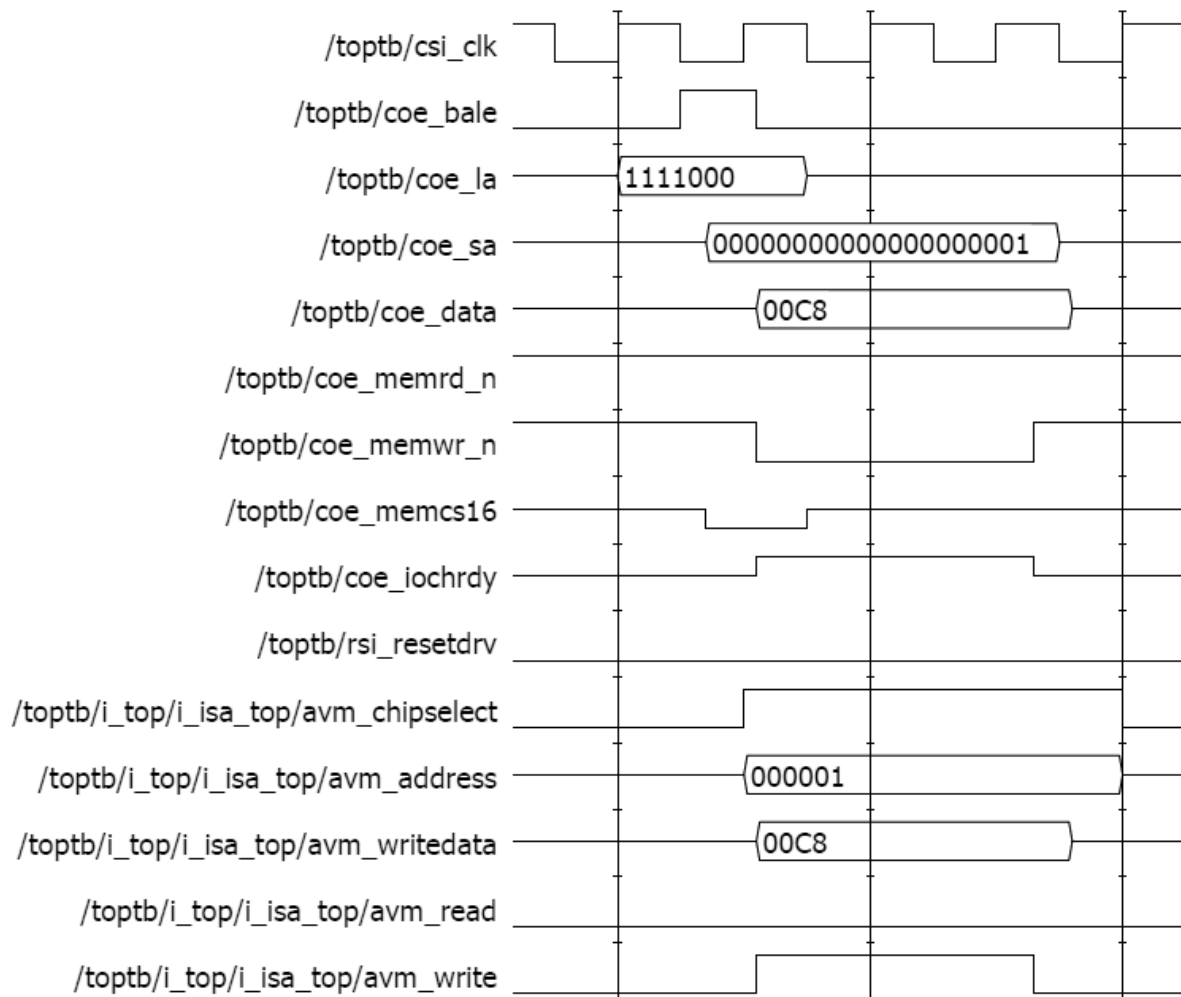
Kód je součástí simulačního procesu. Nejprve čteme z registru **DER1LA** na adrese 0xf00008 a poté zapisujeme data 0xC8 do registru **NOPRA** na adrese 0xf00001.

Průběh simulace sběrniceového cyklu pro čtení z 16 bitové paměťové ISA karty je na obr. 4.1. Čteme řídicí registr **CREG** na adrese 0xf00000. Na 16 bitové datové sběrnici se objeví 0x0FF, což je obsah **CREG**.



Obr. 4.1 Simulace sběrniceového cyklu pro čtení z 16 bitové paměťové ISA karty

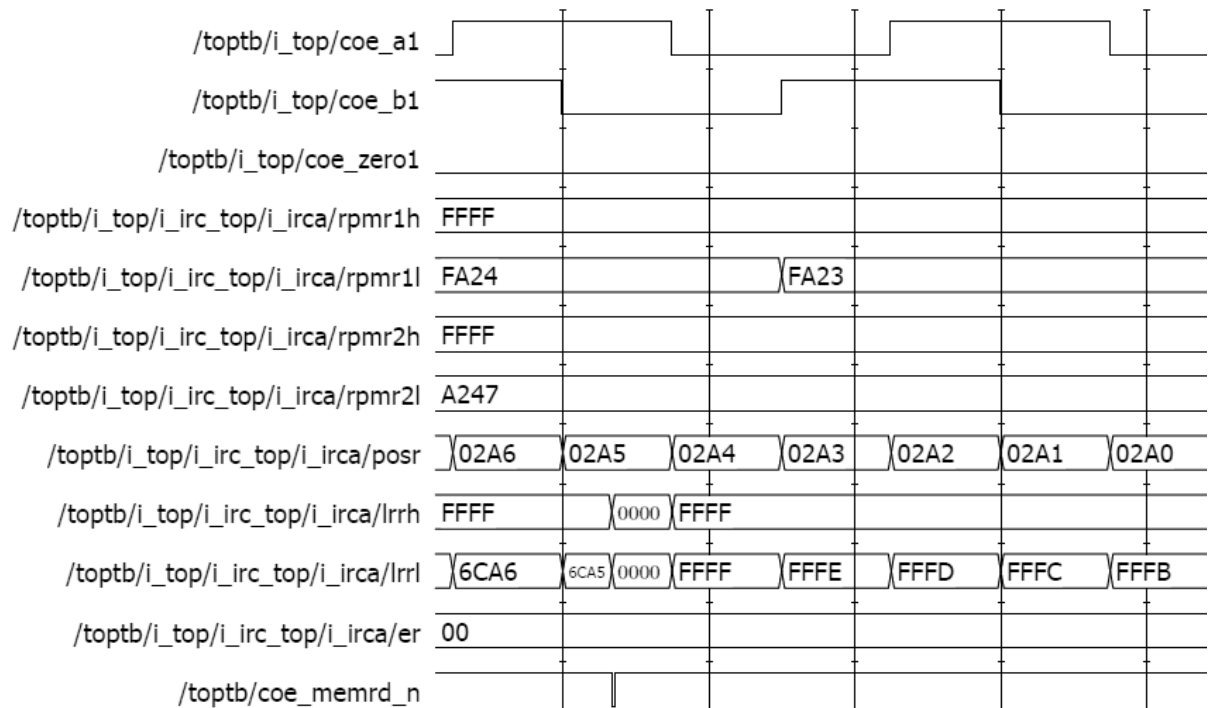
Průběh simulace sběrniceového cyklu pro zápis z 16 bitové paměťové ISA karty je na obr. 4.2. Zapisujeme data 0xC8 do řídicího registru **NOPRA** na adrese 0xf00001.



Obr. 4.2 Simulace sběrniceového cyklu pro zápis do 16bitové paměťové ISA karty

4.2. Simulace modulu pro vyhodnocení otáček

Průběh simulace modulu pro vyhodnocení otáček je na *obr. 4.3*. Na obrázku jsou signály a registry pouze pro jeden snímač. Vyčtení registru **RPMR1LA** způsobí vynulování čítače pulsů od posledního vyčtení **LRRHA** a **LRRLA**. Směr otáčení je levotočivý, číslo v registru je tedy záporné a je kódované dvojkovým doplňkem.



Obr. 4.3 Simulace modulu pro vyhodnocení otáček

5. Závěr

Cílem této práce bylo navrhnout modul pro vyhodnocení otáček a modul umožňující komunikaci po ISA sběrnici. Modul pro vyhodnocení otáček jsem nejprve otestoval na vývojové desce DE2-115 od firmy Altera. Protože jsem inkrementální rotační snímač neměl k dispozici, kvadratické signály a nulový signál jsem simuloval pomocí čítačů řízených oscilátorem. Data z registrů jsem zobrazoval na 7 segmentových LED displejích.

Původně modul pro vyhodnocení otáček vypočítával rychlost otáčení přímo v jednotkách ot/min a absolutní pozici přímo ve stupních natočení. To však vyžadovalo velký počet násobiček a děliček. Obvod FPGA Altera Cyclone IV na desce DE2-115 byl pro realizaci této funkce dostačující, protože disponuje celkem 266 hardwarovými násobičkami 18 x 18 bit. Výsledná IP funkce je však primárně určena pro realizaci v obvodu FPGA Xilinx řady Spartan-II osazeném na desce FPGA 6800, která je nasazená na základní desku počítače PC104. Obvody řady Spartan-II nedisponují hardwarovými násobičkami, proto při syntéze bylo nutné realizovat násobičky a děličky pomocí velkého množství LUT tabulek. Z tohoto důvodu se však výsledná IP funkce nevešla do cílového obvodu na desce FPGA 6800. Aby se výsledná IP funkce vešla do FPGA obvodu Xilinx Spartan-II, musel jsem tedy násobičky a děličky odstranit. Konečné provedení výstupních registrů rychlostí otáček a absolutní pozice tedy potom obsahuje pouze údaje o počtu pulsů, z kterých lze rychlost otáčení hřídele a absolutní pozici následně vypočítat programem běžícím na CPU počítače PC104.

Z důvodu časové tísně a potížemi s hardwarem jsem stačil v době před odevzdáním práce otestovat na desce FPGA 6800 pouze čtení z jednoho 16 bitového registru umístěného v paměťovém prostoru. Kompletní přístup do 16 bitového paměťového prostoru IP funkce z ISA sběrnice jsem otestoval simulacemi na simulátoru v ModelSimu.

6. Seznam použité literatury

- [1] NOVÁK, Petr. *Rotační inkrementální senzory* [online]. 2002, [citováno 15.4.2013]. Dostupné z WWW: <<http://www.odbornecasopisy.cz/download/au100232.pdf>>.
- [2] BARTOŠÍK, Petr. *Snímače polohy a otáček v praxi* [online]. 2003, [citováno 15.4.2013]. Dostupné z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=28744>.
- [3] NOVOTNÝ, Marek. *The design and the control of linear motor drive for micropositioning* [online]. 2003, [citováno 15.4.2013]. Dostupné z WWW: <https://noppa.aalto.fi/noppa/kurssi/as-74.3135/materiaali/control_of_linear_motors.pdf>
- [4] NATIONAL INSTRUMENTS. *Encoder measurements: how to guide* [online]. 2012, [citováno 1.5.2013]. Dostupné z WWW: <<http://www.ni.com/white-paper/7109/en>>.
- [5] Ústav mechatroniky a technické informatiky [online]. [citováno 1.5.2013]. Dostupné z WWW: <http://www.mti.tul.cz/files/svm/Snimace_polohy.pdf>.
- [6] HUISMAN, Sander. *Gray Code* [online]. 2010, [citováno 8.5.2013]. Dostupné z WWW: <<http://shuisman.com/?p=569>>.
- [7] HEWLET PACKARD. *Fundamentals of the electronics counters* [online]. 1997, [citováno 5.5.2013]. Dostupné z WWW: <<http://www.leapsecond.com/pdf/an200.pdf>>.
- [8] Industry Standard Architecture. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001- , last modif. on 21 February 2013 [citováno 5.5.2013]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Industry_Standard_Architecture>.
- [9] ŠNOREK MIROSLAV, RICHTA KAREL: *Připojování periférií k PC*. Praha, GRADA, 1996. ISBN 80-7169-146-1.
- [10] SHANLEY TOM, ANDERSON DON: *ISA System Architecture*. Third edition. Reading, Massachusetts, Addison-Wesley, 1995. ISBN 0-201-40996-8.
- [11] Fpga4fun. *Quadrature Decoder* [online]. 2008, [citováno 29.4.2013]. Dostupné z WWW: <<http://www.fpga4fun.com/QuadratureDecoder.html>>.

7. Příloha 1 – Seznam řídicích registrů

CREG - Řídicí registr

adresa: 0x0

Bit	15	14	13	12	11	10	9	8
	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZSB	ZSA	SBB	SAB	SBA	SAA	SCB	SCA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bit 7 - ZSB: Přítomnost nulového signálu čidla B

0	nulový signál čidla B není k dispozici
1	nulový signál čidla B je k dispozici

bit 6 - ZSA: Přítomnost nulového signálu čidla A

0	nulový signál čidla A není k dispozici
1	nulový signál čidla A je k dispozici

bit 5 - SBB: Přítomnost signálu b čidla B

0	signál b není k dispozici
1	signál b je k dispozici

bit 4 - SAB: Přítomnost signálu a čidla B

0	signál a není k dispozici
1	signál a je k dispozici

bit 3 - SBA: Přítomnost signálu b čidla A

0	signál b není k dispozici
1	signál b je k dispozici

bit 2 - SAA: Přítomnost signálu a čidla A

0	signál a není k dispozici
1	signál a je k dispozici

bit 1 - SCB: Přítomnost čidla B

0	nepřipojeno
1	připojeno

bit 0 - SCA: Přítomnost čidla A

0	nepřipojeno
1	připojeno

NOPRA - Počet pulzů na otáčku - čidlo A

adresa: 0x1

Bit	15	14	13	12	11	10	9	8
	-	-	-	-	-	NPA10	NPA9	NPA8
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NPA7	NPA6	NPA5	NPA4	NPA3	NPA2	NPA1	NPA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	1

bity 10..0 - NPA10..0: Počet pulzů na otáčku

Maximální počet pulzů na otáčku je 2047

NOPRB - Počet pulzů na otáčku - čidlo B

adresa: 0x2

Bit	15	14	13	12	11	10	9	8
	-	-	-	-	-	NPB10	NPB9	NPB8
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NPB7	NPB6	NPB5	NPB4	NPB3	NPB2	NPB1	NPB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	1

bity 10..0 - NPB10..0: Počet pulzů na otáčku

Maximální počet pulzů na otáčku je 2047

PERIODRHA - Interval měření frekvence (horních 16 bitů) - čidlo A

adresa: 0x3

Bit	15	14	13	12	11	10	9	8
	PHA15	PHA14	PHA13	PHA12	PHA11	PHA10	PHA9	PHA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PHA7	PHA6	PHA5	PHA4	PHA3	PHA2	PHA1	PHA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - PHA15..0: Interval měření frekvence (horních 16 bitů: 31..16)

PERIODRLA - Interval měření frekvence (dolních 16 bitů) - čidlo A
adresa: 0x4

Bit	15	14	13	12	11	10	9	8
	PLA15	PLA14	PLA13	PLA12	PLA11	PLA10	PLA9	PLA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PLA7	PLA6	PLA5	PLA4	PLA3	PLA2	PLA1	PLA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	1

bity 15..0 - PLA15..0: Interval měření frekvence (dolních 16 bitů: 15..0)

PERIODRHB - Interval měření frekvence (horních 16 bitů) - čidlo B
adresa: 0x5

Bit	15	14	13	12	11	10	9	8
	PHB15	PHB14	PHB13	PHB12	PHB11	PHB10	PHB9	PHB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PHB7	PHB6	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - PHB15..0: Interval měření frekvence (horních 16 bitů: 31..16)

PERIODRLB - Interval měření frekvence (dolních 16 bitů) - čidlo B
adresa: 0x6

Bit	15	14	13	12	11	10	9	8
	PLB15	PLB14	PLB13	PLB12	PLB11	PLB10	PLB9	PLB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PLB7	PLB6	PLB5	PLB4	PLB3	PLB2	PLB1	PLB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	1

bity 15..0 - PLB15..0: Interval měření frekvence (dolních 16 bitů: 15..0)

DER1HA - Maximální derivace RPM1 (horních 16 bitů) - čidlo A
adresa: 0x7

Bit	15	14	13	12	11	10	9	8
	D1HA15	D1HA14	D1HA13	D1HA12	D1HA11	D1HA10	D1HA9	D1HA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D1HA7	D1HA6	D1HA5	D1HA4	D1HA3	D1HA2	D1HA1	D1HA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D1HA15..0: maximální derivace RPM1 (horních 16 bitů: 31..16)

DER1LA - Maximální derivace RPM1 (dolních 16 bitů) - čidlo A
adresa: 0x8

Bit	15	14	13	12	11	10	9	8
	D1LA15	D1LA14	D1LA13	D1LA12	D1LA11	D1LA10	D1LA9	D1LA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D1LA7	D1LA6	D1LA5	D1LA4	D1LA3	D1LA2	D1LA1	D1LA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D1LA15..0: maximální derivace RPM1 (dolních 16 bitů: 15..0)

DER1HB - Maximální derivace RPM1 (horních 16 bitů) - čidlo B
adresa: 0x9

Bit	15	14	13	12	11	10	9	8
	D1HB15	D1HB14	D1HB13	D1HB12	D1HB11	D1HB10	D1HB9	D1HB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D1HB7	D1HB6	D1HB5	D1HB4	D1HB3	D1HB2	D1HB1	D1HB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D1HB15..0: maximální derivace RPM1 (horních 16 bitů: 31..16)

DER1LB - Maximální derivace RPM1 (dolních 16 bitů) - čidlo B

adresa: 0xA

Bit	15	14	13	12	11	10	9	8
	D1LB15	D1LB14	D1LB13	D1LB12	D1LB11	D1LB10	D1LB9	D1LB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D1LB7	D1LB6	D1LB5	D1LB4	D1LB3	D1LB2	D1LB1	D1LB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D1LB15..0: maximální derivace RPM1 (dolních 16 bitů: 15..0)**DER2HA - Maximální derivace RPM2 (horních 16 bitů) - čidlo A**

adresa: 0xB

Bit	15	14	13	12	11	10	9	8
	D2HA15	D2HA14	D2HA13	D2HA12	D2HA11	D2HA10	D2HA9	D2HA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D2HA7	D2HA6	D2HA5	D2HA4	D2HA3	D2HA2	D2HA1	D2HA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D2HA15..0: maximální derivace RPM2 (horních 16 bitů: 31..16)**DER2LA - Maximální derivace RPM2 (dolních 16 bitů) - čidlo A**

adresa: 0xC

Bit	15	14	13	12	11	10	9	8
	D2LA15	D2LA14	D2LA13	D2LA12	D2LA11	D2LA10	D2LA9	D2LA8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D2LA7	D2LA6	D2LA5	D2LA4	D2LA3	D2LA2	D2LA1	D2LA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D2LA15..0: maximální derivace RPM2 (dolních 16 bitů: 15..0)

DER2HB - Maximální derivace RPM2 (horních 16 bitů) - čidlo B
adresa: 0xD

Bit	15	14	13	12	11	10	9	8
	D2HB15	D2HB14	D2HB13	D2HB12	D2HB11	D2HB10	D2HB9	D2HB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D2HB7	D2HB6	D2HB5	D2HB4	D2HB3	D2HB2	D2HB1	D2HB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D2HB15..0: maximální derivace RPM2 (horních 16 bitů: 31..16)

DER2LB - Maximální derivace RPM2 (dolních 16 bitů) - čidlo B
adresa: 0xE

Bit	15	14	13	12	11	10	9	8
	D2LB15	D2LB14	D2LB13	D2LB12	D2LB11	D2LB10	D2LB9	D2LB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	D2LB7	D2LB6	D2LB5	D2LB4	D2LB3	D2LB2	D2LB1	D2LB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	1	1	1	1	1	1	1	1

bity 15..0 - D2LB15..0: maximální derivace RPM2 (dolních 16 bitů: 15..0)

8. Příloha 2 – Seznam výstupních registrů

RPMR1HA - Výstupní registr měření periody (horních 16 bitů) - čidlo A adresa: 0xF

Bit	15	14	13	12	11	10	9	8
	R1HA15	R1HA14	R1HA13	R1HA12	R1HA11	R1HA10	R1HA9	R1HA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R1HA7	R1HA6	R1HA5	R1HA4	R1HA3	R1HA2	R1HA1	R1HA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R1HA15..0: měření periody kvadraturních pulsů (horních 16 bitů)
Záporné číslo je kódováno v dvojkovém doplňku.

RPMR1LA - Výstupní registr měření periody (dolních 16 bitů) - čidlo A adresa: 0x10

Bit	15	14	13	12	11	10	9	8
	R1LA15	R1LA14	R1LA13	R1LA12	R1LA11	R1LA10	R1LA9	R1LA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R1LA7	R1LA6	R1LA5	R1LA4	R1LA3	R1LA2	R1LA1	R1LA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R1LA15..0: měření periody kvadraturních pulsů (dolních 16 bitů)
Záporné číslo je kódováno v dvojkovém doplňku.

RPMR1HB - Výstupní registr měření periody (horních 16 bitů) - čidlo B adresa: 0x11

Bit	15	14	13	12	11	10	9	8
	R1HB15	R1HB14	R1HB13	R1HB12	R1HB11	R1HB10	R1HB9	R1HB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R1HB7	R1HB6	R1HB5	R1HB4	R1HB3	R1HB2	R1HB1	R1HB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R1HB15..0: měření periody kvadraturních pulsů (horních 16 bitů)
Záporné číslo je kódováno v dvojkovém doplňku.

RPMR1LB - Výstupní registr měření periody (dolních 16 bitů) - čidlo B

adresa: 0x12

Bit	15	14	13	12	11	10	9	8
	R1LB15	R1LB14	R1LB13	R1LB12	R1LB11	R1LB10	R1LB9	R1LB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R1LB7	R1LB6	R1LB5	R1LB4	R1LB3	R1LB2	R1LB1	R1LB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R1LB15..0: měření periody kvadraturních pulsů (dolních 16 bitů)

Záporné číslo je kódováno v dvojkovém doplňku.

RPMR2HA - Výstupní registr měření frekvence (horních 16 bitů) - čidlo A

adresa: 0x13

Bit	15	14	13	12	11	10	9	8
	R2HA15	R2HA14	R2HA13	R2HA12	R2HA11	R2HA10	R2HA9	R2HA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R2HA7	R2HA6	R2HA5	R2HA4	R2HA3	R2HA2	R2HA1	R2HA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R2HA15..0: měření frekvence (horních 16 bitů)

Záporné číslo je kódováno v dvojkovém doplňku.

RPMR2LA - Výstupní registr měření frekvence (dolních 16 bitů) - čidlo A

adresa: 0x14

Bit	15	14	13	12	11	10	9	8
	R2LA15	R2LA14	R2LA13	R2LA12	R2LA11	R2LA10	R2LA9	R2LA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	R2LA7	R2LA6	R2LA5	R2LA4	R2LA3	R2LA2	R2LA1	R2LA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R2LA15..0: měření frekvence (dolních 16 bitů)

Záporné číslo je kódováno v dvojkovém doplňku.

RPMR2HB - Výstupní registr měření frekvence (horních 16 bitů) - čidlo B

adresa: 0x15

Bit	15	14	13	12	11	10	9	8
	R2HB15	R2HB14	R2HB13	R2HB12	R2HB11	R2HB10	R2HB9	R2HB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	R2HB7	R2HB6	R2HB5	R2HB4	R2HB3	R2HB2	R2HB1	R2HB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R2HB15..0: měření frekvence (horních 16 bitů)

Záporné číslo je kódováno v dvojkovém doplňku.

RPMR2LB - Výstupní registr měření frekvence (dolních 16 bitů) - čidlo B

adresa: 0x16

Bit	15	14	13	12	11	10	9	8
	R2LB15	R2LB14	R2LB13	R2LB12	R2LB11	R2LB10	R2LB9	R2LB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	R2LB7	R2LB6	R2LB5	R2LB4	R2LB3	R2LB2	R2LB1	R2LB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - R2LB15..0: měření frekvence (dolních 16 bitů)

Záporné číslo je kódováno v dvojkovém doplňku.

ZEROROTHA - Výstupní registr měření nulových otáček (horních 16 bitů) - čidlo A

adresa: 0x17

Bit	15	14	13	12	11	10	9	8
	ZHA15	ZHA14	ZHA13	ZHA12	ZHA11	ZHA10	ZHA9	ZHA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ZHA7	ZHA6	ZHA5	ZHA4	ZHA3	ZHA2	ZHA1	ZHA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - ZHA15..0: měření nulových otáček (horních 16 bitů)

ZEROROTLA - Výstupní registr měření nulových otáček (dolních 16 bitů) - čidlo A
adresa: 0x18

Bit	15	14	13	12	11	10	9	8
	ZLA15	ZLA14	ZLA13	ZLA12	ZLA11	ZLA10	ZLA9	ZLA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZLA7	ZLA6	ZLA5	ZLA4	ZLA3	ZLA2	ZLA1	ZLA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - ZLA15..0: měření nulových otáček (dolních 16 bitů)

ZEROROTHB - Výstupní registr měření nulových otáček (horních 16 bitů) - čidlo B
adresa: 0x19

Bit	15	14	13	12	11	10	9	8
	ZHB15	ZHB14	ZHB13	ZHB12	ZHB11	ZHB10	ZHB9	ZHB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZHB7	ZHB6	ZHB5	ZHB4	ZHB3	ZHB2	ZHB1	ZHB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - ZHB15..0: měření nulových otáček (horních 16 bitů)

ZEROROTLB - Výstupní registr měření nulových otáček (dolních 16 bitů) - čidlo B
adresa: 0x1A

Bit	15	14	13	12	11	10	9	8
	ZLB15	ZLB14	ZLB13	ZLB12	ZLB11	ZLB10	ZLB9	ZLB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZLB7	ZLB6	ZLB5	ZLB4	ZLB3	ZLB2	ZLB1	ZLB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - ZLB15..0: měření nulových otáček (dolních 16 bitů)

POSRA - Výstupní registr absolutní pozice - čidlo A

adresa: 0x1B

Bit	15	14	13	12	11	10	9	8
	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - PA15..0: absolutní pozice

Musí být k dispozici nulový signál.

POSRB - Výstupní registr absolutní pozice - čidlo B

adresa: 0x1C

Bit	15	14	13	12	11	10	9	8
	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - PB15..0: absolutní pozice

Musí být k dispozici nulový signál.

LRRHA - Výstupní registr počtu pulzů od posledního vyčtení (horních 16 bitů) - čidlo A

adresa: 0x1D

Bit	15	14	13	12	11	10	9	8
	RHA15	RHA14	RHA13	RHA12	RHA11	RHA10	RHA9	RHA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RHA7	RHA6	RHA5	RHA4	RHA3	RHA2	RHA1	RHA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - RHA15..0: počet pulzů od posledního vyčtení (horních 16 bitů: 31..16)

Záporné číslo je kódováno v dvojkovém doplňku.

LRRLA - Výstupní registr počtu pulzů od posledního vyčtení (dolních 16 bitů) - čidlo A
adresa: 0x1E

Bit	15	14	13	12	11	10	9	8
	RLA15	RLA14	RLA13	RLA12	RLA11	RLA10	RLA9	RLA8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RLA7	RLA6	RLA5	RLA4	RLA3	RLA2	RLA1	RLA0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - RLA15..0: počet pulzů od posledního vyčtení (dolních 16 bitů: 15..0)
Záporné číslo je kódováno v dvojkovém doplňku.

LRRHB - Výstupní registr počtu pulzů od posledního vyčtení (horních 16 bitů) - čidlo B
adresa: 0x1F

Bit	15	14	13	12	11	10	9	8
	RHB15	RHB14	RHB13	RHB12	RHB11	RHB10	RHB9	RHB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RHB7	RHB6	RHB5	RHB4	RHB3	RHB2	RHB1	RHB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - RHB15..0: počet pulzů od posledního vyčtení (horních 16 bitů: 31..16)
Záporné číslo je kódováno v dvojkovém doplňku.

LRRLB - Výstupní registr počtu pulzů od posledního vyčtení (dolních 16 bitů) - čidlo B
adresa: 0x20

Bit	15	14	13	12	11	10	9	8
	RLB15	RLB14	RLB13	RLB12	RLB11	RLB10	RLB9	RLB8
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RLB7	RLB6	RLB5	RLB4	RLB3	RLB2	RLB1	RLB0
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bity 15..0 - RLB15..0: počet pulzů od posledního vyčtení (dolních 16 bitů: 15..0)
Záporné číslo je kódováno v dvojkovém doplňku.

ER - Výstupní chybový registr

adresa: 0x21

Bit	15	14	13	12	11	10	9	8
	-	-	CO6B	CO5B	CO4B	CO3B	CO2B	R2B
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	R1B	CO6A	CO5A	CO4A	CO3A	CO2A	R2A	R1A
Read/Write	R	R	R	R	R	R	R	R
Počáteční hodnota	0	0	0	0	0	0	0	0

bit 0 - R1A: Překročena derivace rpm1 - čidlo A

0	v pořádku
1	překročena derivace rpm1 - čidlo A

bit 1 - R2A: Překročena derivace rpm2 - čidlo A

0	v pořádku
1	překročena derivace rpm2 - čidlo A

bit 2 - CO2A: Přetečení čítače cnt2 - čidlo A

0	v pořádku
1	výpadek signálu ZERO

bit 3 - CO3A: Přetečení čítače cnt3 - čidlo A

0	v pořádku
1	přetečení čítače

bit 4 - CO4A: Přetečení čítače cnt4 - čidlo A

0	v pořádku
1	přetečení čítače

bit 5 - CO5A: Přetečení čítače cnt5 - čidlo A

0	v pořádku
1	přetečení čítače

bit 6 - CO6A: Přetečení čítače cnt6 - čidlo A

0	v pořádku
1	přetečení čítače

bit 7 - R1B: Překročena derivace rpm1 - čidlo B

0	v pořádku
1	překročena derivace rpm1 - čidlo B

bit 8 - R2B: Překročena derivace rpm2 - čidlo B

0	v pořádku
1	překročena derivace rpm2 - čidlo B

bit 9 - CO2B: Přetečení čítače cnt2 - čidlo B

0	v pořádku
1	přetečení čítače

bit 10 - CO3B: Přetečení čítače cnt3 - čidlo B

0	v pořádku
1	přetečení čítače

bit 11 - CO4B: Přetečení čítače cnt4 - čidlo B

0	v pořádku
1	přetečení čítače

bit 12 - CO5B: Přetečení čítače cnt5 - čidlo B

0	v pořádku
1	přetečení čítače

bit 13 - CO6B: Přetečení čítače cnt6 - čidlo B

0	v pořádku
1	přetečení čítače