

1/ Semestrální práce – aproximace fce

1a/

Pomocí vhodných metod realizujte fce aproximující elementární fce:

Fci $y_f = \sin_f(x_f)$ aproximující fci sinus

kde x_f a y_f je reálné číslo v jednoduché přesnosti (float, 4B),

poznámky:

Pro vlastní aproximaci používejte pouze float aritmetiku s jednoduchou přesností (float, 4B) v libovolném prog.jazyce.

Cílem je aproximovat fci s co největší přesností, případně minimalizace výp.náročnosti.

K ověření přesnosti výsledků srovnajte s vestavěnými funkcemi pro float-aritmetiku (float, 4B i float 8B) např. $\sin_f(x_f) - \sin(x_f)$, případně $(\text{double})\sin_f(x_f) - \sin((\text{double})x_f)$

Práce by měla obsahovat: výběr metody aproximace a redukce argumentu, způsob získání případných parametrů a koeficientů aprox.metody, realizaci vlastní aproximační fce, stanovení výpočetní náročnosti, stanovení přesnosti fce (+ speciální případy viz níže), analýza přesnosti vs počet koeficientů aprox.fce, analýzu výsledků a závěr.

Pro hodnocení přesnosti (mimo jiné) zkuste bod $x = \pi/2$ a zjistěte pro jak velké okolí x je $\sin(x) = 1$. Dále se podívejte na body $x = 1000000 * \pi$, $x = 1000000 * \pi + \pi/2$.

1b/

Navrhněte funkci `distance(lo0,lo1,la0,la1)` pro výpočet vzdálenosti dvou gps bodů. Body jsou reprezentovány souřadnicemi jako int32 kde 1bit = $1e-5$. (tj. např. souřadnice 49.72670, 13.35246 je uložena jako čísla 4972670, 1335246).

Poznámky:

Algorismus navrhněte s ohledem na efektivitu (rychlost, paměť). Výpočet by měl probíhat v int32 (operace +,*) tak aby celkový výsledek měl rozlišení 1bit = 1m a měl přesnost +/-0.1% +/-0.5m. (tj. max. chyba +/-1.1m pro 100m vzdálenost, +/-100.5m pro 100km rozdíl atp.)

Pro výpočet vzdálenosti předpokládejte že body jsou blízko sebe (do 1000km), tj. nemusíte uvažovat zakřivení země = plane projection). https://en.wikipedia.org/wiki/Geographical_distance

Výsledky můžete ověřit přesnějším výpočtem a/nebo externími kalkulátory, jako např. <https://gps-coordinates.org/distance-between-coordinates.php>

(hint: Jedná se o „problémově závislou) (tj. neuniverzální) aproimaci fce `cos()` a `sqrt()`

Práce by měla obsahovat: výběr metody aproximace, diskusi omezení, realizaci vlastní aproximační fce, stanovení výpočetní náročnosti, stanovení přesnosti fce, analýzu výsledků a závěr.

Semestrální práce – VHDL FSM

FSM =Finite state machine = stavový automat

Prozkoumejte možnosti kódování stavů FSM. Pro daný N-stavový automat zjistěte vliv kódování na složitost (počet CLB a D) a rychlost obvodu.

https://www.xilinx.com/support/documentation/sw_manuals/xilinx10/isehelp/pp_db_xst_hdl_synthesis_options.htm

<https://www.allaboutcircuits.com/technical-articles/encoding-the-states-of-a-finite-state-machine-vhdl/>

https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/hdl/vhdl/vhdl_file_dir_enum_encoding.htm

Semestrální práce – VHDL sin

Implementujte logiku microstepování (sinus generator) s pwm výstupem pro obvod typu CPLD.

Vstupem jsou signály plus,minus (hranou dojde k posunu na generatoru o jeden krok do plusu/minusu). výstupem je pwm signal a polarita. Přepokládejte existenci signálu hodin.

Otestujete dve topologie:

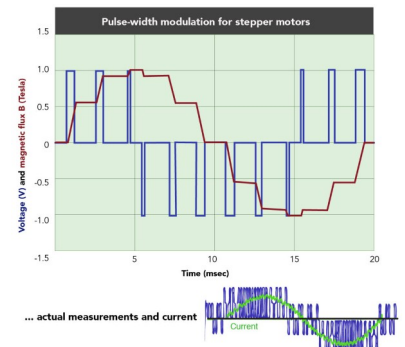
1/ x-sin(x)-pwm

2/ x-aprox_sin(x)-pwm

Analyzujte složitost/rychlost jednotlivých topologií pro 4, 16 a 64 mikrokroků (mikrokroky rozdělují 90st na n-částí)

Xilinx ISE pro CPLD:

<https://uart.cz/577/cpld-a-prvni-aplikace/>



Semestrální práce – zadané téma

Zpracování daného tématu dle domluvy – řešerše tématu, praktické testy (je-li aplikovatelné), výstup ve formě referátu + odpřednášení (cca 30+ minut netriviálních informací). Možná témata:

- Souborové systémy systém pro (NAND) flash (= SSD type flash) pro microcontrolery
- Realtime OS – implementace pro micronrolery. srovnání
- Srovnání (RISC) mikrokontrolerů – MIPS (PIC32) vs ARM (STM32)
- Debugování v microcontrolerech, debugování v multiprocesorových systémech
- architektura RISC-V (vs ARM)
- ? (vaše téma)

