Aplikativna uporaba DSP kartice eZdsp^{TM} LF2407

Dejan KOS, Martin Terbuc (mentor)
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Smetanova 17, 2000 Maribor, Slovenija
dejan.kos@email.si

Applications of eZdsp^{TM} LF2407 board.

Abstract: When the development of DSPs started their use was limited to few fields as space researching, medicine, etc.. Because of technological advancement they are now cheap and applicable in general use.

As the demonstration of possible applications of DSP board, four experiments are presented. Matching hardware is added to the realized software. Software was written in C programming language using Code Composer.

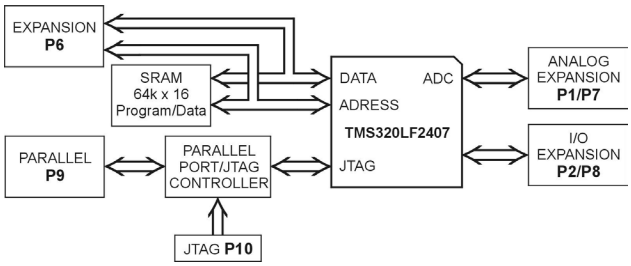
1 Uvod

Digitalni signalni procesorji (DSP), so v začetku razvoja (v 60-ih letih prejšnjega stoletja) zaradi cene bili namenjeni zgolj uporabo na določenih rezerviranih področjih, ki jim še danes pripisujemo velik pomen (raziskovanje vesolja, medicina, ipd.). Tehnološki razvoj je omogočil, da so DSP-ji danes namenjeni tudi širši uporabi in če je verjeti velikim proizvajalcem DSP-jev bodo le ti v prihodnosti, na nekaterih področjih povsem izpodrinili operacijske ojačevalnike.

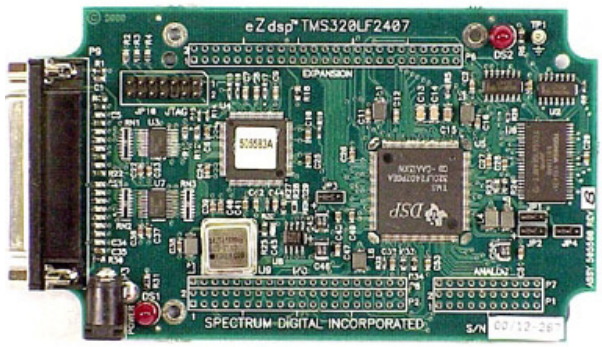
Za demonstracijo nekaterih možnih aplikacij DSP kartice so izdelani posamezni eksperimenti, ki so zasnovani za uporabo v pedagoškem procesu. Namen eksperimentov je prikazati možne načine programiranja DSP-ja in predstaviti pomembnejše funkcije DSP kartice.

2 Opis kartice eZdsp^{TM} LF2407

DSP kartica *eZdsp^{TM} LF2407* je kit komplet proizvajalca *Spectrum Digital*, ki vsebuje digitalni signalni procesor s celoštevilsko aritmetiko (angl. fixed-point) proizvajalca *Texas Instruments* [2]. Na trgu dosega prodajno ceno 299 $ [1]. Kartica je primarno zasnovana za regulacijo trifaznih asinhronskih motorjev, enosmernih (DC) motorjev s trajnimi magneti in reluktančnih motorjev. Blokovno shemo DSP kartice prikazuje slika 1.



Slika 1: Blokovna shema DSP kartice *eZdsp^{TM} LF2407*



Slika 2: *eZdsp^{TM} LF2407*

DSP kartica prikazana na sliki 2, ima naslednje splošne lastnosti:

- 40 MIPS (pri 40 MHz),
- 2 kW SARAM , 544 W DARAM,
- 64 kW SRAM na kartici (),
- 32 kW FLASH,
- 16 x 10-bitnih multipleksiranih analognih vhodov (čas pretvorbe 375 ns po kanalu),
- 4 x 16-bitnih časovnikov,
- 4 x 16-bitnih neodvisnih pulzno širinsko

moduliranih (PŠM) generatorjev,
- 2 x 6 16-bitnih odvisnih PŠM (angl. PWM) generatorjev in
- 6 x 8 binarnih vhodno/izhodnih enot.

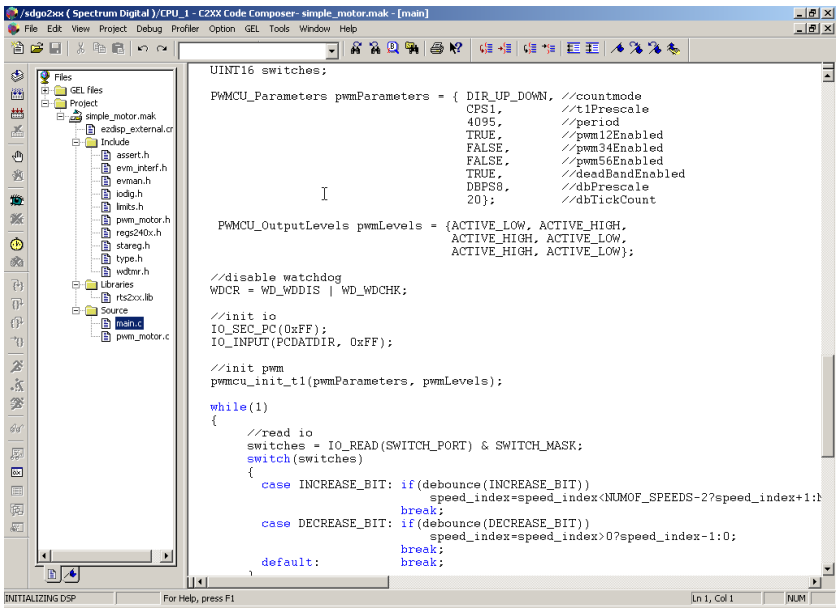
Arhitektura procesorja je narejena za procesiranje v sprotnem času (angl. real-time). Kartica nudi bogat nabor perifernih enot in pri številnih aplikacijah, poleg že omenjenih, predstavlja ekonomsko in tehnološko upravičeno rešitev:

- neomejena prilagodljivost hitrosti brezkrtačnih (angl. brushless) motorjev, s čimer lahko zmanjšamo porabo električne energije tudi do 25 %,
- zmanjšanje števila senzorjev pri regulaciji motorjev z uporabo naprednejših programskih algoritmov ipd..

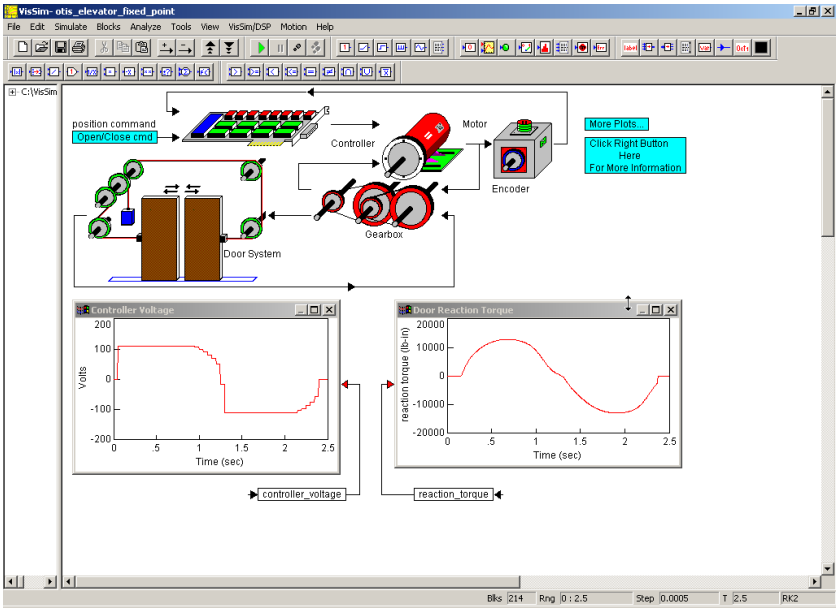
Zaradi fixed-point procesorja ima kartica omejene možnosti v aritmetiki, vendar tudi številne prednosti pred arhitekturo s plavajočo vejico. DSP-ji s celoštevilsko aritmetiko imajo bistveno nižjo porabo energije. Zaradi manjše osnove procesorja (silicijevo jedro) so enostavnejši za izdelavo, posledično pa tudi cenejši. Na račun preproste zgradbe je veliko časovno kritičnih podatkovnih povezav hitrejših.

3 Programiranje DSP kartice

Za programiranje, nalaganje programa in izvajanje programa po korakih se uporablja orodje *Code Composer* podjetja *Texas Instruments*. Omogoča programiranje s pomočjo ANSI C programskega jezika in avtomatsko generiranje izvršne kode. V času izvajanja lahko opazujemo vsebino oz. vrednosti spremenljivk, vrednosti v registrih DSP kartice pa lahko tudi izrišemo v obliki grafa. Program *Code Composer* je popolno programsko orodje, v katerem lahko uporabljamo funkcijo razhroščevanja (angl. debug), ki se uporablja za iskanje napak v programu. S pomočjo te funkcije vstavljamo prekinitve (angl. breakpoints) in medtem, ko program izvršujemo po korakih (angl. single step ali trace mode) opazujemo spreminjanje registrov oz. pomnilniškega prostora. Program omogoča vključevanje knjižnic (angl. header - *.h), izvornih datotek (angl. source - *.c) in datotek v zbirnem jeziku (angl. assembler - *.asm). Programsko okolje je grafično predstavljeno tako, da so barvno ločene ukazne vrstice (črna), komentar (zelena), makroji in funkcije (modra) in imena vključenih datotek (rdeča).



Slika 3: Programsko okolje *Code Composer*



Slika 4: Okolje programa *VisSim*

Druga možnost, ki jo imamo pri razvoju in izdelavi programske opreme je program *VisSim* podjetja *Visual Solutions, Inc*. [3].

VisSim je grafično programsko orodje, ki omogoča načrtovanje algoritmov izključno v grafični obliki, podobno kot v programu *Simulink®*, vendar ima integrirano podporo za bloke, lastne ciljnem sistemu, *eZdsp™* kitu. Z razliko od programa *Simulink®* tukaj blokom ne moremo spreminjati velikosti, ampak so fiksni. *VisSim* se lahko uporablja kot simulator, ali pa kot orodje ki omogoča zagon algoritma na ciljni kartici *eZdsp™* v realnem času. Kadar se *VisSim* uporablja zgolj kot simulator, ciljni sistem (DSP kartica) ni potreben. Uporablja se lahko za hiter razvoj in testiranje programske opreme (angl. quick prototyper). Program *VisSim* generira programsko kodo v programskem jeziku C za celoten model, ali za izbran segment (npr. regulator v regulacijski progi). V tem primeru se regulacijski algoritem izvaja na DSP kartici, medtem, ko se ostali algoritmi izvajajo v simulacijskem načinu. Posamezne bloke lahko predstavimo tudi v obliki bitne slike, kar prikazuje slika 4. S tem bistveno povečamo preglednost programa.

4 Opis eksperimentov

Izvedeni eksperimenti so sestavljeni iz programske opreme napisane v programskem jeziku C in pripadajoče strojne opreme. Namen je bil izdelati takšne eksperimentalne sisteme, ki bodo vplivali na okolje in s tem zagotavljali pomembno motivacijsko komponento učnega programa. Programi so napisani v programu *Code Composer* v jeziku C, program *VisSim* pa je uporabljen zgolj kot orodje za verifikacijo rezultatov in kot simulacijski pripomoček.

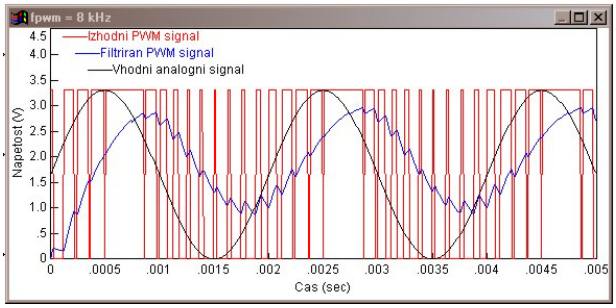
4.1 Generiranje PŠM signala glede na vhodni analogni signal

Namen tega eksperimenta je predstavitev uporabe A/D pretvornika PŠM signalov v kombinaciji s prekinitvami (angl. interrupts). Na začetku vsake fiksno nastavljena periode PŠM signala, se izvede branje trenutne vrednosti na analogno digitalnem (A/D) pretvorniku. Skladno z amplitudo analognega signala se nastavi širina pulza PŠM signala (angl. duty cycle). V programu se poslužujemo prekinitev, ki nam dajejo takt za branje analognega vhoda in nastavljanje širine pulza PŠM signala (slika 5). Generiramo t.i. asimetričen PŠM signal. Njegova značilnost je, da je signal aktiven na eni od skrajnih mej nastavljene periode.

Da lahko signal opazujemo, PŠM signal filtriramo preko nizko pasovnega sita. Sito dimenzioniramo tako, da filtriramo želeno pasovno širino signala in izločimo čimveč

komponent šuma. Slika 5 prikazuje primer simulacije generiranja PŠM signala frekvence 8 kHz s sinusnim signalom frekvence 500 Hz in filtriranje PŠM signala z nizko pasovnim sitom. Frekvenca PŠM signala mora v tem primeru biti znatno višja od vhodnega analognega signala (po Shannonovem teoremu vsaj dva krat). Za frekvenco signala 500 Hz določimo časovno konstanto RC sita [5]:

$$RC = \frac{1}{2\pi f} = 3{,}2 \cdot 10^{-4}\, s \qquad (1)$$

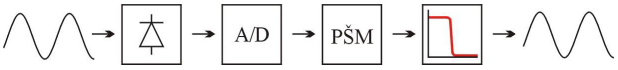


Slika 5: PŠM signal filtriran z nizko pasovnim sitom prvega reda

Ker signal kvalitetneje moduliramo pri višjih frekvencah izberemo PŠM signal frekvence 8 kHz. Dušenje pri tej frekvenci PŠM signala izračunamo po enačbi:

$$(dB) = -10 \cdot \log[1 + (2\pi f \cdot RC)^2] = -24\, dB \quad (2)$$

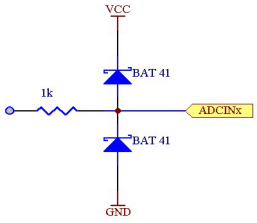
Principa delovanja opisanega eksperimenta prikazuje slika 6.



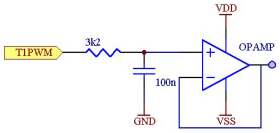
Slika 6: Blokovna shema za generiranje PŠM signala z analognim signalom

Da zaščitimo A/D pretvornik pred preveliko vhodno napetostjo uporabimo zaščitno vezje z dvema Schottky diodama in uporom (slika 7).

Spremenljivo breme bi vplivalo na izhodni signal, zato RC sito priključimo na preko napetostnega sledilnika. Zaradi visoke vhodne upornosti in majhne izhodne upornosti operacijskega ojačevalnika, izničimo vpliv priključenega bremena na RC sito (slika 8).



Slika 7: Zaščita A/D pretvornika



Slika 8: Nizko pasovno RC sito

4.2 *Digitalni merilnik napetosti*

Eksperiment je namenjen demonstraciji pretvorbe analogne vrednosti napetosti v obliko, ki omogoča prikazovanje vrednosti na LED zaslonu. Program izvaja tipanje napetosti analognega vhoda in skladno z napetostjo prikazuje vrednosti na dvo znakovnem LED zaslonu. Zaslon krmilimo s pomočjo petnajstih binarnih izhodov (2 x 7 segmentov in decimalna pika). Merilno območje A/D pretvornika (od 0 do 3,3 V) razširimo z vhodnim prilagoditvenim vezjem. Izdelano je tako, da omogoča merjenje napetosti od 0 do 4 V ali od 0 do 40 V, kar izbiramo preko stikala s katerim v tokokrog vključimo ustrezne napetostne delilnike, ter vklopimo ali izklopimo decimalno piko. Za zaščito A/D pretvornika uporabimo zaščitno vezje s slike 7. Blokovna shema eksperimenta je prikazana na sliki 9.

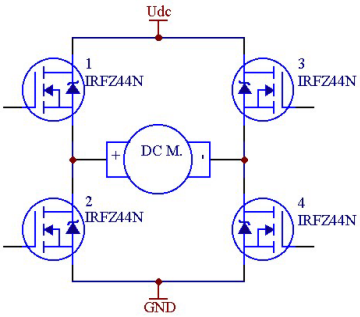
Slika 9: Blokovna shema pri digitalnem merjenju napetosti

4.3 *Krmiljenje enosmernega motorja*

Za demonstracijo osnovnega pristopa pri krmiljenju motorjev smo izdelali eksperiment. Za spreminjanje smeri in hitrosti vrtenja enosmernega motorja, potrebujemo dva PŠM signala in ustrezno strojno opremo, ki odvisno od signalov dovaja napetost na sponke motorja.

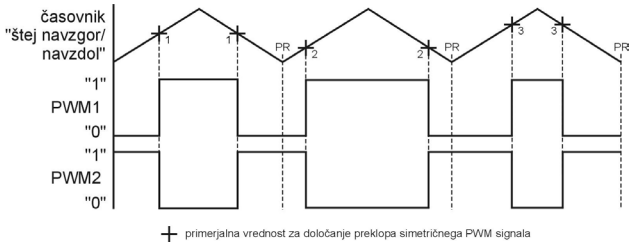
Programsko generiramo dva PŠM signala, ki sta medsebojno invertirana in z njima krmilimo

tranzistorje v H mostiču. Frekvenco PŠM signala nastavimo zunaj slišnega področja človeka (npr. 20 kHz), da ne povzročamo motečega hrupa pri delovanju. Z enim PŠM signalom krmilimo tranzistorja 1 in 4, z drugim pa tranzistorja 2 in 3 (slika 10).



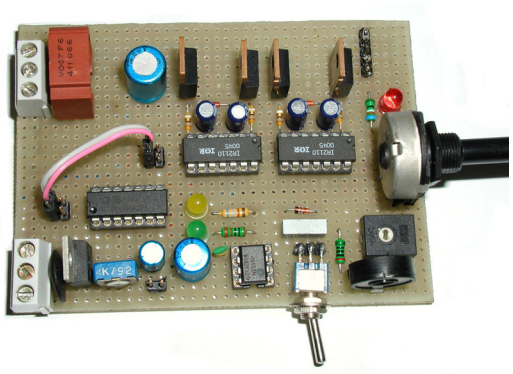
Slika 10: Tranzistorji v H mostiču

V tej nalogi so uporabljeni t.i. simetrični PŠM signali. Simetrični PŠM signali se od asimetričnih PŠM signalov razlikujejo v tem, da imajo področje aktivnosti ali neaktivnosti na sredini periode signala, z enakim odmikom od obeh skrajnih mej (slika 11) [4]. Kot vidimo na sliki, je za oblikovanje simetričnega PŠM signala potrebno števec nastaviti na štetje "navzgor/navzdol". Signal PWM1 je glede na primerjalno vrednost časovnika definiran kot "aktiven zgoraj", medtem ko je signal PWM2 definiran kot "aktiven spodaj". Uporaba simetričnih PŠM signalov je nujna, kadar z njimi krmilimo tri fazni mostič.



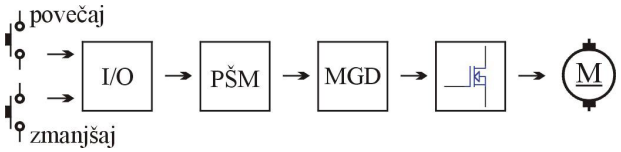
Slika 11: Simetrična PŠM signala (PWM1 in PWM2)

Ukaz za spreminjanje širine pulza PŠM signala se izvaja preko dveh binarnih vhodov (povečanje, zmanjšanje). Aktivnost določenega binarnega vhoda rezultira v inkrementalni spremembi širine pulza PŠM signala, na v programu prednastavljene vrednosti. S postavljanjem binarnih vhodov se pomikamo navzgor ali navzdol po polju definiranih vrednosti, s katerimi nastavljamo širino pulza PŠM signala. Ker generiramo dva invertirana PŠM signala povečanje širine aktivnega pulza prvega signala, posledično pomeni zmanjšanje širine aktivnega pulza drugega PŠM signala. Za krmiljenje tranzistorjev v H mostiču (slika 12) uporabimo dve prožilni vezji (angl. gate driver - MGD).



Slika 12: H mostič z napajanjem in testnim vezjem

Princip krmiljenja enosmernega motorja nam prikazuje slika 13.



Slika 13: Blokovna shema krmiljenja enosmernega motorja

4.4 *Predvajanje tonskega zapisa s PŠM signalom*

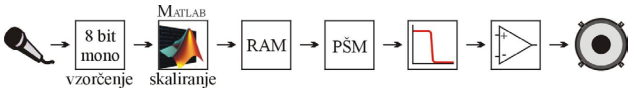
Za demonstracijo uporabe zunanjega RAM pomnilnika (SRAM na kartici), vanj naložimo vzorčene vrednosti tonskega zapisa. Snemalnik zvokov uporabimo za vzorčenje tonskega zapisa. S pomočjo dobljenih vzorčenih vrednosti, ki jih preberemo preko funkcije za branje tonskega zapisa programa *Matlab®* ("wavread"), nastavljamo širino pulza PŠM signala. V programu *Matlab®* vzorčen tonski zapis v bipolarni obliki (-1 do 1) pred tem pretvorimo v unipolarno obliko (0 do 32767), kar omogoča nastavljanje širine pulza PŠM signala v 16-bitnem registru DSP kartice. Pomembno je, da frekvenco PŠM signala nastavimo na enako vrednost pri kateri je

potekalo vzorčenje tonskega zapisa. Če tonski zapis vzorčimo npr. v 16-bitnem mono formatu pri 8 kHz, mora biti frekvenca PŠM signala tudi 8 kHz. Tako s pomočjo PŠM signala opišemo izvorni signal pri enaki frekvenci kot je vzorčen. Da tonski zapis lahko predvajamo je potrebno signal ojačiti, še prej pa ga pretvoriti v analogno obliko.

Ker DSP kartica nima integriranega D/A pretvornika ga izdelamo v obliki RC sita (nizko pasovno sito). Digitalni PŠM signal pretvorimo v analognega, ter ga ojačimo s preprosto avdio ojačevalno stopnjo. Zaradi omejenega pomnilniškega prostora SRAM pomnilnika, ki je rezerviran za shranjevanje podatkov (32 kW), imamo na voljo kratek čas za predvajanje tonskega zapisa in verifikacijo delovanja programske in strojne opreme. To lahko odpravimo, če program izvajamo v zanki. Izdelano avdio ojačevalno stopnjo prikazuje slika 14.

Slika 14: Avdio mono ojačevalna stopnja

Za predstavitev principa delovanja vzorčenja in predvajanja tonskega zapisa si oglejmo sliko 15.



Slika 15: Blokovna shema vzorčenja in predvajanja tonskega zapisa

5 Zaključek

Glede na realizirano programsko in strojno opremo lahko trdimo, da so eksperimenti smiselno zaokroženi in omogočajo uporabo v pedagoškem procesu ali v demonstracijske namene. V eksperimentih so predstavljene glavne funkcije DSP kartice in sicer:

- pulzno širinsko modulirani (PŠM oz. PWM) signali,
- analogno digitalni (A/D) pretvornik in
- binarne vhodno/izhodne (I/O) enote.

Za lažje pisanje programov je bil izdelan nabor knjižnic, kot pripomoček kasnejšim uporabnikom izdelane opreme pa tehniška dokumentacija z naslovom "*eZdsp LF2407 (programiranje DSP kartice)*".

Na podlagi izdelanih eksperimentov sedaj pripravljamo učila, za širšo uporabo v pedagoškem procesu.

6 Zahvala

Zahvaljujem se dr. Martinu Terbuc za mentorstvo pri izvajanju dela in dr. Miranu Rodič za številne nasvete. Podjetju *Texas Instruments* (oddelku za izobraževanje) se zahvaljujem za donacijo DSP kartic *eZdsp*^{TM}.

7 Literatura

[1] http://www.spectrumdigital.com/cgi/catalog.cgi (eZdsp, product number 761119)
[2] http://focus.ti.com/docs/prod/folders/print/tms320lf2407a.html
[3] http://www.vissim.com/index.htm
[4] Texas Instruments SPRU357b: *TMS320LF /LC240xA DSP Controllers Reference Guide, System and Peripherals*, Revised December 2001
[5] Microchip AN-538: *Using PWM to Generate Analog Output*, September 1995