Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroničke sustave i obradbu informacija

Napredne metode digitalne obrade signala
ak.godina 2008/2009

Korištenje mikrokontrolera za digitalnu obradu signala na primjeru Diskretne kosinusne transformacije

Seminarski rad

Juraić Marko, 0036415893

Orešković Lovro, 0036412176

Vranešević Branislav, 0036417898

Zagreb, siječanj 2009.

SADRŽAJ

1. Uvod

Diskretna kosinusna transformacija (DCT) služi za rastavljanje konačnog digitalnog signala na sumu kosinusnih funkcija različitih frekvencija. Primjenjuje se u mnogim područjima obrade signala, ponajviše za kompresiju slika i audio signala., a koristi se i pri numeričkom rješavanju parcijalnih diferencijalnih jednadžbi spektralnim metodama. Korištenje kosinusa umjesto sinusa je nužno za slijedeće primjene: kompresija; kosinusne funkcije su puno efikasnije (manje ih je potrebno za aproksimaciju tipičnih signala), dok kod diferencijalnih jednadžbi kosinusi daju točno određeni izbor rubnih uvjeta.

Općenito, DCT je vrlo slična diskretnoj Fourierovoj transformaciji, samo što koristi realne brojeve. DCT je ekvivalentna DFT-u dvostruke duljine uzorka, pri obradi realnih parnih signala (budući da je Fourierova transformacija realnih i parnih funkcija realna i parna), gdje su u nekim varijantama ulaz i/ili izlaz pomaknuti za pola uzorka.

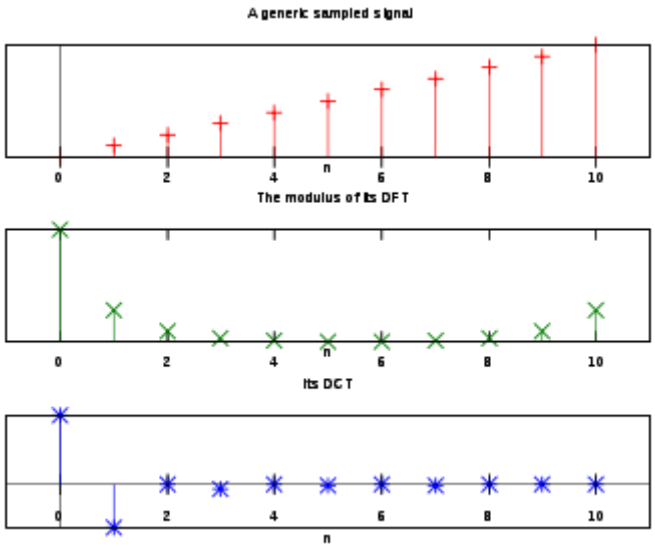
Ima osam standardnih DCT postupaka, od čega se četiri najčešće koriste.

Najčešće korištena DCT transformacija je tip-II DCT koji se često naziva samo DCT, a njen inverz je tip-III DCT koji se naziva inverzni DCT, odnosno IDCT.

Srodne transformacije DCT-u su diskretna sinusna transformacija (DST) koja je ekvivalentna DFT-u realnih i neparnih funkcija, te modificirana diskretna kosinusna transformacija (MDCT) koja se temelji na preklapanju podataka.

2. Primjene

DCT, posebice DCT-II se često koristi u obradi signala i slika, pogotovo kod kompresije s gubicima jer ima jako dobra svojstva u smislu čuvanja energije: većina informacije iz signala se nalazi na niskofrekvencijskim komponentama DCT-a.



Na slikama su prikazane DFT (srednja slika) i DCT (donja slika) ulaznog signala (prva slika). Primjećujemo da je većina informacije kod DCT-a u prvih par uzoraka, dok kod DFT to nije slučaj.

MDCT se koristi u AAC, Vorbis, WMA i MP3 kompresiji audio signala.

DCT je još široko zastupljen u rješavanju parcijalnih diferencijalnih jednadžbi spektralnim metodama te je usko vezana sa Chebyshevljevim polinomima.

Brzi DCT algoritmi se koriste u Chebyshevljevim aproksimacijama proizvoljnih funkcija nizom Chebyshevljevih polinoma.

3. DCT u 8-bitnim digitalnim sustavima

Floating point aritmetika je prespora za izvedbu u 8-bitnim procesorima. Ona se može koristiti u skaliranju nekih ulaznih podataka ili tome slično, ali gdje je potrebna brza obrada informacija primorani smo koristiti fixed point aritmetiku.

Mi smo koristili 8:8 fixed point aritmetiku. To znači da prvih 8 bitova predstavlja cijeli dio broja, a drugih 8 bitova decimalni dio.

Brojevi su zapisani kao 16-bitne cjelobrojne varijable. To nam omogućava dinamički raspon od -128 do +127 (-2^7 do (2^7)-1) te rezoluciju od 1/256 tj. 0.003906.

Pretvorbu iz cjelobrojnih varijabli u float te u drugu stranu smo vršili množenjem tj. djeljenjem s 256. Ova pretvorba zahtjeva samo 1 ciklus jer je 256 potencija broja 2 pa se množenje tj. djeljenje izvodi pomakom u lijevo tj. desno za 8 binarnih mjesta.

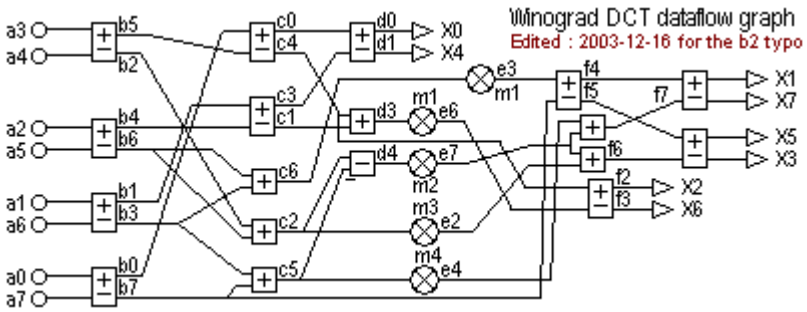
Mikrokontroler koji smo mi koristili u ovome seminaru je Atmel atMega32. Ta izvedba atmelovog mikrokontrolera ima hardverski izvedeno množilo koje nam omogućava množenje do dva 16-bitna broja, bez velikog broja ciklusa.

Nadalje, ono što utječe na krajnju upotrebljivost rješenja je realizacija koju smo odabrali. DCT je definiran kao $X_k = \sum_{N-1}^{n=0} x_n \cos\left[\frac{pi}{N}\left[n+\frac{1}{2}\right]k\right]$. Kada bi se odlučili za realizaciju koja bi bila direktno računanje formule dobili bi tzv. Butterfly oblik.

U tome obliku računa veliki je broj množenja koji uvelike otežava računanje tj. računalno je intenzivnije.

1992. Winograd i Feig su došli do algoritma za računanje DCT-a sa minimalnim brojem množenja. To je bio nastavak na Winograd-ov rad sa FFT algoritmima gdje je ranije uspio doći do sličnog rješenja. Brza Fourierova transformacija i Diskretna kosinusna transformacija su po izvedbi vrlo slične pa bilo koji napredak u jednom polju relativno brzo i lako se preslikava u drugo.

Butterfly realizacija za 1DN=8 DCT koristi 13 množenja i 22 zbrajanja. Za istu kosinusnu transformaciju Winograd-ov algoritam koristi 5 množenja i 29 zbrajanja.



Blok shema Winograd-ov algoritma

Ubrzanja u izvođenju koda su odmah vidljiva i iznose čak do 40 posto. Funkcija množenja koja je korištena u ovome programu poprilično je optimirana, i kao takvoj joj treba 34 ciklusa za izvršenje. Vidi se da smo korištenjem Winogradova algoritma uštedili 240 ciklusa.

Ono što Winogradov algoritam također postiže je veća točnost, tj. manji je "šum" uslijed množenja. Naime zbog zaokruživanja rezultata kod množenja uslijed podljeva i preljeva gubi se točnost. Što je manje množenja to je veća točnost.

Da bi izračun DCT-a bio što brži i zbog fixed aritmetike koja je korištena bilo je potrebno napisati funkciju množenja koja množi dva 8.8 fixed broja.

Asemblerski kod koji to radi ima ovakav oblik:

```
"mov  r22,%A1"    "\n\t"          //load a
"mov  r23,%B1"    "\n\t"
"mov  r20,%A2"    "\n\t"          //load b
"mov  r21,%B2"    "\n\t"
"muls       r23, r21"     "\n\t"      //(signed)ah * (signed)bh
"mov        r31, r0"      "\n\t
"mul        r22, r20"     "\n\t"      // al * bl
"mov        r30, r1"      "\n\t"
"mulsu      r23, r20"     "\n\t"      // (signed)ah * bl
"add        r30, r0"      "\n\t"
"adc        r31, r1"      "\n\t"
"mulsu      r21, r22"     "\n\t"      // (signed)bh * al
"add        r30, r0"      "\n\t"
"adc        r31, r1"      "\n\t"
"mov        %A0, r30"     "\n\t"
"mov        %B0, r31"     "\n\t"
```

Kao što je vidljivo iz gornjeg koda, ono što nam omogućuje relativno brzo množenje brojeva, od npr. float brojeva istoga raspona je dobar odabir zapisa brojeva, pošto je broj bitova ispred i iza decimalne točke jednak širini registra u mikrokontroleru. Takav odabir zapisa brojeva nam omogućuje da brojeve pomnožimo na ovaj način:

a0.a1*b0.b1= a0*b0+a1*b1+a0*b1+a1*b0

4. Zaključak

Za obradbu digitalnih signala (DSP) koriste se specifični DSP procesori. Takvi procesori imaju duboki cjevovod i registre velike širine, nekada 24 ili čak 32 bita. Nadalje, imaju hardverski izvedene operacije koje se koriste u DSP-u (multiply and accumulate MAC, konvolucije množenja djeljenja, rad sa brojevima sa pomičnom decimalnom točkom...). Oni jako dobro odrađuju svoj posao, jer su upravo za njega i namijenjeni.

Ono čega treba biti svjestan da i mikroprocesori opće namjene i manje širine riječi se, uz neke kompromise, mogu uspješno iskoristiti za obradbu digitalnih signala. Uz neke predostrožnosti lako je izvesti digitalnu filtraciju (FIR i IIR filtri) ili transformaciju (FFT, DCT), a upravo nam to daje veću fleksibilnost u izboru mikroprocesora jer DSP procesori mogu biti skupi ili teško nabavljivi.

5. Literatura

http://instruct1.cit.cornell.edu/courses/ee476/Math/index.html

http://instruct1.cit.cornell.edu/courses/ee476/Math/avrDSP.htm

http://elm-chan.org/works/akilcd/report_e.html