Politechnika Białostocka

Wydział Elektryczny
Katedra Telekomunikacji i Aparatury Elektronicznej

Instrukcja do zajęć laboratoryjnych z przedmiotu:

Architektura i Programowanie Procesorów Sygnałowych

Kod: TS1C510210

Numer ćwiczenia: 2

Temat ćwiczenia:

Podstawowe procedury obliczeniowe procesora sygnałowego TMS320C67x. Arytmetyka zmiennoprzecinkowa.

Opracował:
dr inż. Dariusz Jańczak

Białystok 2017

Architektura i Programowanie Procesorów Sygnałowych

Temat: Podstawowe procedury obliczeniowe procesora sygnałowego TMS320C67x. Arytmetyka zmiennoprzecinkowa.

1. Cel ćwiczenia

Celem ćwiczenia jest ugruntowanie wiedzy studentów oraz nabycie przez nich umiejętności z zakresu podstaw programowania mikroprocesorów. W ramach zajęć studenci nabywają podstawową umiejętność programowania w języku niskiego poziomu oraz umiejętność posługiwania się zintegrowanym środowiskiem programistycznym. Studenci zdobywają umiejętność tworzenia, uruchamiania i testowania oprogramowania procesorów na przykładzie zmiennoprzecinkowego procesora sygnałowego TMS320C6713. Poznają zasady tworzenia asemblerowych programów obliczeniowych wykorzystujących zmiennoprzecinkową reprezentację liczb o pojedynczej i podwójnej precyzji. Realizowanymi zadaniami w bieżącym ćwiczeniu są podstawowe oraz specyficzne dla DSP zmiennoprzecinkowe działania arytmetyczne. Zadania obejmują obliczenia prowadzone na danych uzyskanych natychmiastowo oraz poprzez adresowanie pośrednie.

2. Zagadnienie do opracowania przed przystąpieniem do zajęć

Przed przystąpieniem do zajęć należy opracować następujące zagadnienia:

- reprezentacje liczb w formacie: zmiennoprzecinkowym pojedynczej i podwójnej precyzji (wg normy IEEE754),
- arytmetyka zmiennoprzecinkowa,
- zinterpretować ciąg bitów o długości 32 i 64 w różnych formatach liczbowych,
- zasady inicjalizacji zmiennych różnych typów (dyrektywy: .float, .double) (SPRU186 rozdz. 4.3),
- lista i składnia rozkazów zmiennoprzecinkowych procesora TMS320C6713 (SPRU189 rozdz. 4) pozwalających na realizację podstawowych działań (dodawanie, odejmowanie, mnożenie, porównywanie i konwersja formatów liczb),
- funkcje rejestrów stanu i konfiguracji dla operacji zmiennoprzecinkowych: FADCR, FAUCR, FMCR (SPRU189).

3. Przebieg ćwiczenia

Ćwiczenia laboratoryjne prowadzone są w oparciu o zestaw TMS320C6713 DSP Starter Kit. DSK składa się ze sprzętu i oprogramowania umożliwiającego realizację procedur przetwarzania sygnałów w oparciu o procesor TMS320C6713.

3.1 Podstawowe działania arytmetyczne na liczbach zmiennoprzecinkowych:
Napisać i uruchomić następujące programy:

- dodawanie dwóch liczby zmiennoprzecinkowych o pojedynczej i podwójnej precyzji (wynik działania zapisać do pamięci),
- mnożenie dwóch liczb zmiennoprzecinkowych o pojedynczej i podwójnej precyzji,
- konwersja formatów liczb:
  - z formatu zmiennoprzecinkowego o podwójnej precyzji na stałoprzecinkowy,
  - z formatu zmiennoprzecinkowego o podwójnej precyzji na zmiennoprzecinkowego o pojedynczej precyzji,
  - z formatu stałoprzecinkowego na zmiennoprzecinkowy o podwójnej precyzji,
- porównanie dwóch liczb zmiennoprzecinkowych o podwójnej precyzji,
- w oparciu o odpowiednie wartości liczbowe, stosując odpowiednie działania zobrazować funkcje rejestrów: FADCR, FAUCR, FMCR

3.2 Zadania złożone:
Wykonać zestaw zadań przydzielony grupie przez prowadzącego. Zakres zadań będzie obejmował:

- przetwarzanie lub sortowanie zbiorów danych,
- działania na wektorach i macierzach.

4. Sprawozdanie powinno zawierać:

Realizację wszystkich punktów ćwiczenia w tym:

- kody źródłowe wraz z opisem,
- wyniki działania procedur (stan i zmiany wartości rejestrów i komórek pamięci),
- uwagi i wnioski nasuwające się w trakcie wykonywania ćwiczenia.

5. Wymagania BHP

W trakcie realizacji programu ćwiczenia należy przestrzegać zasad omówionych we wstępie do ćwiczeń, zawartych w: „Regulaminie porządkowym w laboratorium” oraz w „Instrukcji obsługi urządzeń elektronicznych znajdujących się w laboratorium z uwzględnieniem przepisów BHP”. Regulamin i instrukcja są dostępne w pomieszczeniu laboratoryjnym w widocznym miejscu.

6. Literatura

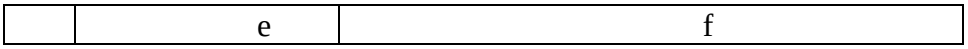
1. Kowalski H. A., *Procesory DSP dla praktyków,* BTC, Legionowo, 2013
2. Kowalski H. A., *Procesory DSP w przykładach,* BTC, Legionowo, 2012.
3. Dąbrowski A. (red.) *Przetwarzanie Sygnałów Przy użyciu Procesorów Sygnałowych*, Wyd. Politechniki Poznańskiej, Poznań 2000.
4. Gryś S., *Arytmetyka komputerów*, PWN, Warszawa, 2007
5. Texas Instruments, *TMS320C6000 Assembly Language Tools User's Guide*, 2008.
6. Texas Instruments, *TMS320C67x DSP CPU and Instruction Set Reference Guide*, 2006.
7. Texas Instruments, *TMS320C6000 Programmer's Guide*, 2006.
8. Texas Instruments, *TMS320C6000 Optimizing Compiler User's Guide*, 2017.
9. Welch T. B., Wright C.H.G., Morrow M.G., *Real-time Digital Signal Processing from Matlab to C with the TMS320C6x DSPs,* Taylor & Francis, 2012.

Dodatek

Liczby zmiennoprzecinkowe

wg: IEEE 754

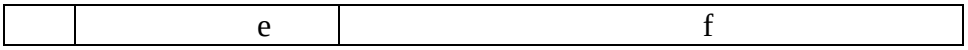
a) format pojedynczy: 32 bity



SP - single precision

.float

b) format podwójny: 64 bity



DP - double precision

.double

SP - Dokładne odwzorowanie liczb dziesiętnych co najmniej do 6 miejsca po przecinku
DP - Dokładne odwzorowanie liczb dziesiętnych co najmniej do 15 miejsca po przecinku

IEEE Floating-Point Notations

Symbol	Meaning
s	Sign bit
e	Exponent field
f	Fraction (mantissa) field
x	Can have value of 0 or 1 (don't care)
NaN	Not-a-Number (SNaN or QNaN)
SNaN	Signal NaN
QNaN	Quiet NaN
NaN_out	QNaN with all bits in the f field= 1
Inf	Infinity
LFPN	Largest floating-point number
SFPN	Smallest floating-point number
LDFPN	Largest denormalized floating-point number
SDFPN	Smallest denormalized floating-point number
signed Inf	+infinity or –infinity
signed NaN_out	NaN_out with s = 0 or 1

rozkazy arytmetyczne zmiennoprzecinkowe;

- działania arytmetyczne
  - add, (ADDDP, ADDSP, ADDAD)
  - sub, (SUBDP, SUBSP)
  - mpy, (MPYDP, MPYI, MPYID, MPYSP)
  - abs, (ABSDP, ABSSP)
  - rcp, (RCPDP, RCPSP)
  - rsq, (RSQDP, RSQSP)

- porównania
  - cmpeq, (CMPEQSP, CMPEQDP)
  - cmpgt, (CMPGTSP, CMPGTDP)
  - cmplt, (CMPLTSP, CMPLTDP)

rozkazy konwersji typu zmiennej.

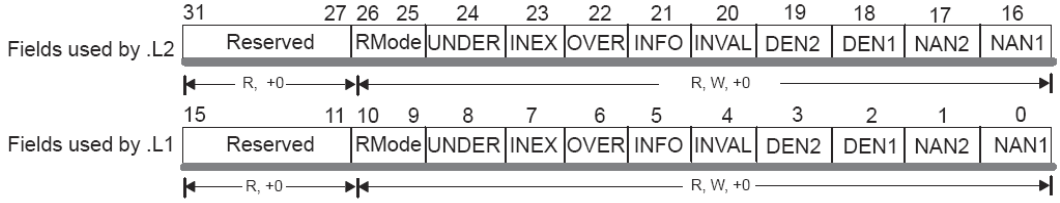
DPINT, DPTRUNNC
DPSP, SPDP
INTDP, INTSP, INTDPU, INTSPU
SPINT, SPTRUNNC

Jednostki wykonujące rozkazy zmiennoprzecinkowe

.L Unit	.M Unit	.S Unit	.D Unit
ADDDP	MPYDP	ABSDP	ADDAD
ADDSP	MPYI	ABSSP	LDDW
DPINT	MPYID	CMPEQDP	
DPSP	MPYSP	CMPEQSP	
DPTRUNC		CMPGTDP	
INTDP		CMPGTSP	
INTDPU		CMPLTDP	
INTSP		CMPLTSP	
INTSPU		RCPDP	
SPINT		RCPSP	
SPTRUNC		RSQRDP	
SUBDP		RSQRSP	
SUBSP		SPDP	

Rejestr stanu

2–8. Floating-Point Adder Configuration Register (FADCR)



Legend: R Readable by the **MVC** instruction
W Writeable by the **MVC** instruction
+0 Value is zero after reset

10–9	2	Rmode .L1	Value 00: Round toward nearest even representable floating-point number Value 01: Round toward 0 (truncate) Value 10: Round toward infinity (round up) Value 11: Round toward negative infinity (round down)
8	1	UNDER .L1	Set to 1 when result underflows
7	1	INEX .L1	Set to 1 when result differs from what would have been computed had the exponent range and precision been unbounded; never set with INVAL
6	1	OVER .L1	Set to 1 when result overflows
5	1	INFO .L1	Set to 1 when result is signed infinity
4	1	INVAL .L1	Set to 1 when a signed NaN is a source, NaN is a source in a floating-point to integer conversion, or when infinity is subtracted from infinity
3	1	DEN2 .L1	*src2* is a denormalized number
2	1	DEN1 .L1	*src1* is a denormalized number
1	1	NAN2 .L1	*src2* is NaN
0	1	NAN1 .L1	*src1* is NaN