

Biblioteka IQMath dla rdzenia C64x+

operacje zmiennoprzecinkowe w stałoprzecinkowych procesorach DSP

Wiele algorytmów implementowanych w procesorach sygnałowych DSP wywodzi się z opracowań pierwotnie wykonanych dla platformy PC. Środowisko zmiennoprzecinkowe PC często wykorzystuje się po to, aby dysponować wystarczającą elastycznością zakresu liczbowego operacji matematycznych. Kod daje się wtedy stosunkowo łatwo przenieść na zmiennoprzecinkowe procesory DSP. Przepisanie kodu w taki sposób, aby działał na stałoprzecinkowych procesorach sygnałowych, może się jednak okazać nie lada wyzwaniem.

Powodów, aby w docelowej implementacji sprzętowej stosować stałoprzecinkowy, a nie zmiennoprzecinkowy procesor DSP, jest wiele. Może się na przykład okazać, że nie ma potrzeby użycia zintegrowanych urządzeń peryferyjnych. Przykładem może być procesor DSP DM6437 firmy Texas Instruments. Układ ten oferuje cyfrowe wejścia i wyjścia wideo. Daje on również możliwość bezpośredniego wprowadzenia analogowego sygnału wideo poprzez zintegrowany przetwornik wideo DAC. W serii układów zmiennoprzecinkowych C67x nie ma natomiast procesora DSP, który dawałby takie możliwości. Wybór stałoprzecinkowego procesora DSP może być też podyktowany wydajnością. Na przykład, układ C6455 może pracować z taktowaniem do 1,2 GHz, udostępniając projektantowi ponad 2 MB wewnętrznej pamięci RAM. Najsilniejszy zmiennoprzecinko-

wy procesor sygnałowy, układ C6727, pracuje z częstotliwością do 350 MHz i ma 288 KB wewnętrznej pamięci RAM. Jak zatem widać, różnice możliwości są znaczne.

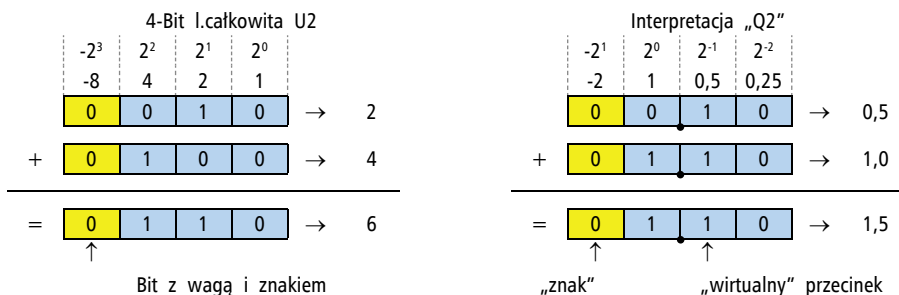
Często zdarza się jednak, że na stałoprzecinkowych procesorach DSP konieczne jest wykonanie operacji na liczbach zmiennoprzecinkowych. Łatwym rozwiązaniem byłoby oparcie się na bibliotecę wspierającą wykonywanie programu (*Runtime Support Library*), emulującej odpowiednie funkcje. Wydłuża to jednak wyraźnie czas wykonywania operacji. Texas Instruments oferuje obecnie projektantom szybszą, dającą wielkie możliwości alternatywę – bibliotekę IQMath. Biblioteki tej z powodzeniem używa się od lat w różnych zastosowaniach bazujących na procesorach DSP z serii C28x. Począwszy od końca 2007 r. jest ona również dostępna dla procesorów DSP z rdzeniem C64x+.

Zasada działania biblioteki IQMath

Liczby zmiennoprzecinkowe zazwyczaj przedstawia się za pomocą mantysy i wykładnika. Implementacja takiej procedury w stałoprzecinkowym procesorze sygnałowym jest bardzo skomplikowana. Emulacja operacji zmiennoprzecinkowych jest w konsekwencji stosunkowo powolna. Podejście zaproponowane w bibliotece IQMath jest zupełnie inne. Wykorzystuje się tu zwykle, stałoprzecinkowe liczby 32-bitowe, w których stawiany jest „wirtualny przecinek”. Procesor DSP wykonuje obliczenia w zwyczajny sposób, tj. na liczbach stałoprzecinkowych, z tą jedyną różnicą, że użytkownik inaczej interpretuje wartości. Na rys. 1 pokazano prosty przykład. Dla ułatwienia zastosowano tylko 4 bity.

Po lewej stronie pokazano dobrze znaną reprezentację binarną U2. Bit LSB (najmniej znaczący) ma wagę $2^0=1$, następny bit ma wagę $2^1=2$ itd. Bit MSB (najbardziej znaczący bit) zawiera również informację o znaku. Tę samą wartość binarną pokazano po stronie prawej, jednak z „wirtualnym przecinkiem” za drugą cyfrą. Bity po lewej stronie przecinka obliczane są tak jak zawsze. Po prawej stronie przecinka na część ułamkową przeznaczono dwa bity (Q2). Prawa strona oznacza teraz ujemne wartości potęgi 2, tj. 2^{-1} i 2^{-2} . Daje to wagi od 0,5 do 0,25. Operacja dodawania pokazana w przykładzie wykonywana jest w sposób standardowy: stosowane są zwykle rozkazy dodawania. W bibliotece IQMath tego typu wartości nazywane są wartościami IQ. Mają one składową całkowitą (I) oraz składową ułamkową (Q), odpowiednio przed i po przecinku.

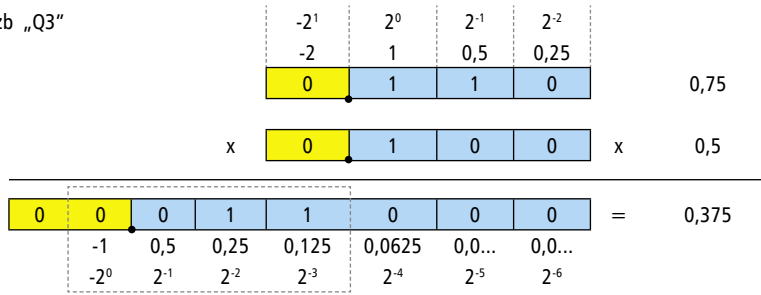
W przypadku operacji mnożenia podejście jest inne. Jeżeli mnożone są przez siebie dwie liczby 4-bitowe, pełna reprezentacja wyniku wymaga użycia 8 bitów. Jaki jest format wartości IQ? Także w tej sytuacji wynik wymaga 8 bitów. Niemniej jednak, w wyniku mnożenia zmieni-



Rys. 1.

Przykład:

Mnożenie liczb „Q3”



Te 4 bity zapisujemy do pamięci => liczba Q3

Rys. 2.

się również miejsce postawienia przecinka. Aby zachować dokładność, z iloczynu wystarczy zatem wydzielić odpowiednie 4 bity. Na rys. 2 przedstawiono przykład mnożenia.

Mnożna i mnożnik są liczbami IQ3 (stosowane jest również oznaczenie I1Q3), co oznacza, że w każdej jest jedno miejsce przed przecinkiem i trzy miejsca po przecinku. To samo dotyczy iloczynu.

W celu wykonania mnożenia liczb IQ procesor musi więc pracować z iloczynem o dwa razy większej dokładności jako wynikiem pośrednim. I tak właśnie jest w procesorach sygnałowych z rdzeniem C64x+. W jednym cyklu zegara mogą one wykonać nawet dwie operacje mnożenia 32 x 32 bity, w każdej otrzymując wynik 64-bitowy. Ponieważ rejestry w rdzeniu mają szerokość tylko 32 bitów, do uzyskania wyniku pośredniego iloczynu wykorzystuje się dwa połączone ze sobą rejestry, które tworzą parę 64-bitową. Następnie za pomocą przesunięć wybieranie i zapamiętywanie odpowiednie 32 bity. Wytworzony kod assemblera dla pojedynczej operacji mnożenia liczb IQ przedstawiono na list. 1 (czynniki w rejestrach A4 i B4, iloczyn w rejestrze A6, we wszystkich przypadkach przykładu liczby IQ8, czyli inaczej I24Q8):

W celu lepszego zrozumienia kodu na rys. 3 pokazano dwa czynniki IQ8 w rejestrach A4 i B4. Rejestry B7 i B6 służą do zapisania 64-bitowego wyniku pośredniego jako pary. Następnie zostają wydzielenie odpowiednie 32 bity i ostatecznie uzyskiwany jest wynik, ponownie w postaci liczby IQ8.

Do tego typu operacji właściwie nie jest potrzebna specjalna biblioteka; można je odtworzyć używając zwykłego kodu C. Jednak w efekcie znacznie ucierpi na tym czytelność kodu źródłowego. Zastosowanie biblioteki IQMath pozwala tego uniknąć. Na rys. 4 pokazano w pierwszym bloku zmiennoprzecinkową operację matematyczną. W drugim bloku ten sam kod został przeniesiony na procesor stałoprzecinkowy. Stosowany jest 64-bitowy wynik po-

Zmienny przecinek	float	Y, M, X, B;
		Y = M * X + B;
Q- wariant z użyciem języka C	int	Y, M, X, B;
		Y = ((long long) M * (long long) X) >> Q + B;
Q- wariant z użyciem IQMath	_iq	Y, M, X, B;
		Y = _IQmpy(M, X) + B;

Rys. 4.

średni. Zwróćmy uwagę na to, że w procesorze DSP C64x+ wartości 32-bitowe reprezentowane są w C jako liczby typu „int”, a wartości 64-bitowe jako liczby typu „long long”. Trzeci blok pokazuje ten sam kod w wersji IQMath. Jedynie mnożenie zostało zastąpione specjalną funkcją. Dodawanie można implementować w zwykły sposób.

Odpowiednia pozycja przecinka dziesiętnego

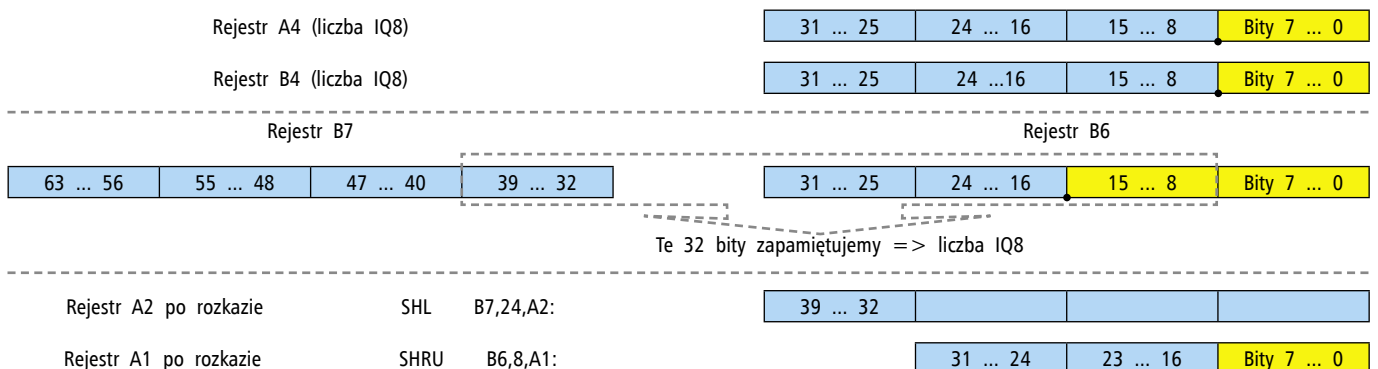
W bibliotece IQMath, w 32-bitowej liczbie stawiany jest „wirtualny” przecinek. Miejsce postawienia przecinka jest wybierane dowolnie. Dla wszystkich liczb IQ istnieje ustawienie globalne. Wybór jest dokonywany za pomocą instrukcji #define GLOBAL_Q. Jeżeli wartość GLOBAL_Q zostanie ustawiona na 16 (IQ16), wówczas każda liczba _iq będzie interpretowana jako liczba U2 z 16 bitami przed przecinkiem (wraz z bitem znaku) i 16 bitami po przecinku. Ustawienie wartości GLOBAL_Q na 24 (IQ24) da natomiast liczbę z 8 bitami przed przecinkiem i 24 bitami po przecinku. Oznacza to, że dla potrzeb reprezentacji można uzyskać całkowicie nowe zakresy wartości. W tab. 1 podano przykłady minimalnych i maksymalnych wartości, które dają się reprezentować oraz maksymalną rozdzielczość dla różnych ustawień przecinka.

Istnieje jednak możliwość wyboru różnych pozycji przecinka w określonych miejscach kodu. Rozkaz IQmpy20(a,b) zakłada na przykład, że

```

List. 1.
MPY32 A4, B4, B7:B6 ; Multiply A4 by B4 (64-bit intermediate result)
NOP 3
SHL B7, 24, A2 ; Of the upper 32 bits, only the 8 LSB are needed,
; the 24 MSBs are "shifted up and out"
SHRU B6, 8, A1 ; Of the lower 32 bits, only the 24 MSBs are needed,
; the 8 LSBs are "shifted down and out"
OR A1, A2, A6 ; Combine both 32-bit values in one
    
```

oba czynniki a i b oraz ich iloczyn są wszystkie liczbami w formacie IQ20. Istnieje jeszcze jedna alternatywna możliwość, w której położenie przecinka w obu czynnikach i iloczynie może być różne. Daje to dużą elastyczność i pozwala projektantowi wybrać rozdzielczość i zakres w zależności od zastosowania.



Rys. 3.

Tab. 1. Minimalne i maksymalne wartości dla różnych ustawień przecinka

GLOBAL_Q	Wartość min.	Wartość max.	Dokładność
31	-1	+0.999 999 999	0.000 000 000 5
30	-2	+1.999 999 999	0.000 000 001
24	-128	+127.999 999 940	0.000 000 060
16	-32 768	+32 767.999 984 741	0.000 015 259
8	-8 388 608	+8 388 607.996 187 500	0.003 906 250
1	-1 073 741 824	+1 073 741 823.5	0.500 000 000
0	-2 147 483 648	+2 147 483 648.0	1.000 000 000

Miejsce postawienia przecinka wymaga uważnego wyboru. Zbyt duża liczba miejsc po przecinku może doprowadzić do sytuacji, w której liczba bitów przed przecinkiem będzie zbyt mała do reprezentowania dużych liczb. Może to spowodować błąd przekroczenia zakresu. Problemy mogą się też pojawić w przypadku zbyt małej dokładności po przecinku. Jeżeli wartość GLOBAL_Q nie jest optymalna dla wszystkich fragmentów programu, inne wartości Q można stosować lokalnie.

Funkcje IQ i wydajność

Obok mnożenia, biblioteka IQMath zawiera również inne operacje matematyczne, takie jak dzielenie, sinus, pierwiastek i funkcje wykładnicze. Wiele funkcji biblioteki występuje nie tylko jako w pełni skompilowane moduły, ale jest również dostępnych jako kod źródłowy w języku C w pliku nagłówkowym. Ma to tę zaletę, że kompilator C nie zawsze musi implementować odwołanie do funkcji, ale zamiast tego potrzebuje tylko wzajemnie zagnieździć poszczególne operacje i zastosować technikę przetwarzania potokowego oprogramowania (*software pipelining*). Jest to jedyny sposób, aby zoptymalizować wydajność dla rdzenia C64x+.

W tab. 2 podano listę czasów wykonania poszczególnych operacji. W kolumnie cykli wykonania w trybie niezależnym podano liczbę cykli potrzebnych do wykonania określonej operacji. Kolumna cykli wykonania w trybie potokowym dotyczy natomiast sytuacji, w której stosowana jest metoda wpłatania w pętlę. Jak widać, w przypadku mnożenia, liczba potrzebnych cykli zostaje zmniejszona z 20 do 1. Oznacza to w efekcie, że w każdym cyklu daje się wykonać pełną operację mnożenia liczb IQ.

IQMath – opcja przełączania

Jeżeli kod zmiennoprzecinkowy zostaje ręcznie przeniesiony na procesor stałoprzecinkowy, zazwyczaj jest to proces jednokierunkowy, w którym nie ma prostej drogi powrotu. Krótki przykład z operacją przesunięcia napisany w C pokazano na rys. 4. Gdyby w algorytmie został później znaleziony błąd i konieczne było sprawdzenie wszystkich obliczeń zmiennoprzecinkowych, kod trzeba by przepisać od nowa. Byłoby to bardzo czasochłonne.

W przypadku IQMath jest jednak inaczej. Prosta zmiana opcji #define w pliku nagłówkowym pozwala wybrać, czy program ma wykony-

wać obliczenia z zastosowaniem liczb IQ, czy na prawdziwych liczbach zmiennoprzecinkowych. Krótki fragment odpowiedniego pliku nagłówkowego (*iqmath.h*) jasno ilustruje sposób działania tego mechanizmu (list. 2).

Jeżeli opcja #define MATH_TYPE jest ustawiona na IQ_MATH, każda zmienna _iq w kodzie jest interpretowana jako całkowita liczba 32-bitowa (I32_IQ odpowiada typowi int) i wywołane zostają odpowiednie operacje IQ (np. IQ25sin dla liczb IQ25). Jeżeli natomiast opcja MATH_TYPE zostanie zmieniona na FLOAT_MATH, zmienną _iq wykorzystuje się dla liczb zmiennoprzecinkowych i używane są zwykłe operacje zmiennoprzecinkowe zamiast ich odpowiedników IQ (np. sin).

Problemy pojawiające się w wynikach algorytmów IQ mogą być spowodowane przekroczeniem zakresu lub niedokładnością. Może się zdarzyć, że niewłaściwie zostało wybrane miejsce postawienia przecinka. Przez szybkie przedstawienie kodu na obliczenia zmiennoprzecinkowe można łatwo sprawdzić algorytm. Wystarczy zmienić opcję MATH_TYPE na FLOAT_MATH i wykonać kod ponownie. Ponieważ zostanie zastosowana emulacja obliczeń zmiennoprzecinkowych, algorytm wykonana się oczywiście dużo wolniej niż poprzednio. Pozwala to jednak szybko rozstrzygnąć, czy w algorytmie jest jakiś problem ogólny, czy też jedynie niewłaściwie dobrano miejsce postawienia przecinka.

IQMath – również na PC

Ponieważ wiele algorytmów najpierw opracowywanych jest na komputerach klasy PC, byłoby korzystne, gdyby biblioteka IQMath mogła być również wykorzystywana bezpośrednio na platformie PC. Umożliwia to biblioteka PC *IQmath_pc.lib*. Pozwala ona wykonywać w środowisku PC te same, 32-bitowe operacje IQ, które działają na procesorach DSP. W konsekwencji użytkownik może korzystać z biblioteki IQMath już od pierwszych faz tworzenia kodu. Biblioteka

```

List. 2.
// FILE: IQmath.h
//#####
#if MATH_TYPE == IQ_MATH
typedef I32_IQ _iq;
typedef I32_IQ _iq31;
typedef I32_IQ _iq30;
...
#define _IQ31(A) ((I32_IQ) ((A) * 2147483648u))
#define _IQ30(A) ((I32_IQ) ((A) * 1073741824u))
...
#if GLOBAL_Q == 25
#define _IQsin(A) (_IQ25sin(A))
#endif
...
//#####
#else // MATH_TYPE == FLOAT_MATH
typedef float _iq;
typedef float _iq31;
typedef float _iq30;
...
#define _IQ(A) (A)
#define _IQ31(A) (A)
...
#define _IQsin(A) sin(A)
...
#endif // No more.
    
```

Tab. 2. Czasy wykonania operacji

Nazwa Funkcji	Format -IQ	Liczba cykli (tryb niezależny)	Liczba cykli (tryb potokowy)
IQNsin	N=1 do 29	56	4
IQNcos	N=1 do 29	54	4
IQatan	N=1 do 29	134	32,1
IQNsqrt	N=0 do 30	79	8,6
IQNisqrt	N=0 do 30	83	11,1
IQNmag	N=0 do 30	89	12,6
IQpow	N=1 do 30	550	n.a.
IQmpy	N=1 do 31	20	1
IQNdiv	N=0 do 31	73	11,1

IQMath jest idealnym rozwiązaniem, nawet jeżeli punktem wyjścia jest kod zmiennoprzecinkowy. Wystarczy po prostu włączyć opcję `MATH_TYPE FLOAT_MATH`, tak aby wszystkie operacje matematyczne były wykonywane w trybie zmiennoprzecinkowym. Zaoszczędza to później pracy przy konwersji kodu z wersji zmiennoprzecinkowej na IQ. Biblioteka IQMath stanowi zatem wsparcie całego procesu rozwojowego oprogramowania, od pierwszych algorytmów prototypowych na platformie PC, aż do późniejszej, docelowej implementacji sprzętowej.

Podsumowanie

Biblioteka IQMath dla C64x+ bardzo ułatwia przenoszenie kodu zmiennoprzecinkowego na platformę DSP. Nawet jeśli algorytm został pierwotnie napisany na platformie PC, dzięki bibliotece PC, IQMath można stosować już od samego początku. Opcja ponownego kompilowania programu dla operacji zmiennoprzecinkowych po dokonaniu konwersji na IQ, pozwala ponadto wykrywać ogólne błędy w algorytmie. Jeżeli zaś zostanie podjęta decyzja na korzyść użycia procesora zmiennoprzecinkowego, zastosowanie kodu IQMath jest nadal możliwe. W tym celu opcję `MATH_TYPE` zmienia się na `FLOAT_MATH`, co skutkuje autentycznym kodem zmiennoprzecinkowym, bez potrzeby dodatkowych zmian.

Istnieją oczywiście przypadki zastosowań, w których używane są autentyczne procesory zmiennoprzecinkowe. Jeżeli większość czasu wykonywania operacji zajmują obliczenia zmiennoprzecinkowe, być może wykonywane nawet z podwójną dokładnością (64 bity), procesor zmiennoprzecinkowy DSP C67x będzie zazwyczaj lepszy niż C64x+. Jednak w wielu zastosowaniach tylko niektóre części kodu wymagają operacji zmiennoprzecinkowych. Większość kodu składa się z kodu sterującego i 8-, 16- lub 32-bitowych operacji stałoprzecinkowych. W takich przypadkach narzuca się wręcz wybór serii C64x+, która dzięki bibliotece IQMath umożliwia skuteczne wykonywanie także operacji zmiennoprzecinkowych.

Łącze do pobrania biblioteki IQMath dla serii C64x+:

<http://focus.ti.com/docs/toolsw/folders/print/sprc542.html>

Bloki IQMath są również dostępne dla środowiska MATLAB Simulink:

<http://focus.ti.com/docs/toolsw/folders/print/sprc590.html>

Robert Finger
Texas Instruments, Niemcy

korekta i uzupełnienia
Krzysztof Kardach

R E K L A M A



**Wzmacniacz słuchawkowy
AVT1024**

AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11
tel. 022 257 84 50, fax 022 257 84 55,
e-mail: handlowy@avt.pl

Dostępne wersje:
A - płytką drukowaną: 5,5zł
B - komplet elementów: 25zł
C - układ zmontowany: 85zł

www.sklep.avt.pl

ELSINCO®

Electronic Measurement Technology

Szanowni Państwo!

*Serdecznie dziękujemy
wszystkim Osobom i Firmom
za współpracę w zbliżającym
się już ku końcowi Roku 2008*

*Życzymy Państwu
Radosnych Świąt Bożego Narodzenia
i
Szczęśliwego Nowego Roku 2009*

Zespół ELSINCO Polska

*Twój kompetentny partner
na rynku zaawansowanych rozwiązań pomiarowych*

ELSINCO Polska Sp. z o.o.

ul. Gdańska 50, 01-691 Warszawa

Tel.: + 22 832 40 42 , Fax: +22 832 22 38

e-mail: office@elsinco.pl, www.elsinco.pl

