

**XIV Seminarium**  
**ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICIE' 2004**  
Oddział Gdański PTETiS

**EFEKTYWNE PROGRAMOWANIE PROCESORA TMS320C6711  
PRZY UŻYCIU PAKIETU SIMULINK**

**Grzegorz GRABOWSKI<sup>1</sup>, Janusz NIEZNAŃSKI<sup>2</sup>,  
Paweł SZCZEPANKOWSKI<sup>3</sup>, Juliusz FRĄCKOWIAK<sup>4</sup>**

Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk,

- |                    |                |                               |
|--------------------|----------------|-------------------------------|
| 1. tel. 348-60-76, | fax: 341-08-80 | e-mail: ggrab@ely.pg.gda.pl   |
| 2. tel. 347-16-75, | fax: 341-08-80 | e-mail: jniez@ely.pg.gda.pl   |
| 3. tel. 348-60-75, | fax: 341-08-80 | e-mail: pszczep@ely.pg.gda.pl |
| 4. tel. 348-60-76, | fax: 341-08-80 | e-mail: julio@ely.pg.gda.pl   |

W artykule opisano alternatywny, w stosunku do standardowych metod kodowania, sposób programowania procesora TMS320C6711 z wykorzystaniem bibliotek pakietu SIMULINK. Scharakteryzowano własności karty uruchomieniowej ze zmiennoprzecinkowym procesorem sygnałowym TMS320C6711 firmy Texas Instruments, pod kątem tworzenia programów do układów prototypowych. Opisano także cechy pakietu SIMULINK jako wydajnego narzędzia pozwalającego na stosunkowo szybkie sprawdzenie wybranych algorytmów i ich praktyczną realizację. Przedstawiono podstawowe bloki możliwe do wykorzystania w środowisku symulacyjnym SIMULINK wraz z prezentacją metodyki tworzenia projektów procesora TMS320C6711. Opisano także możliwości realizacji komunikacji w czasie rzeczywistym między komputerem osobistym a procesorem sygnałowym TMS320C6711. Szczegółowo omówiono tworzenie programu cyfrowego przetwarzania sygnałów realizującego efekt echa umożliwiającego również komunikację z urządzeniem zewnętrznym - komputerem PC. Zaprezentowany sposób programowania może być wykorzystany w sytuacjach, gdy istnieje potrzeba szybkiego tworzenia oprogramowania np. w przypadku prototypów, bez głębszej wiedzy na temat programowania w języku C++.

## 1. WPROWADZENIE

Rozwój współczesnej techniki opartej na systemach mikroprocesorowych przyczynił się do powstania wielu różnorodnych układów dedykowanych do cyfrowego przetwarzania sygnałów. Układy te charakteryzują się wieloma parametrami technicznymi takimi jak: szybkość przetwarzania, ilość dostępnej pamięci itd., jak również funkcjami dodatkowymi np. liczbą portów wejścia/wyjścia, dostępnymi funkcjami komunikacyjnymi. Do najwydajniejszych układów z tej rodziny należą procesory sygnałowe DSP (ang. *Digital Signal Processing*) dedykowane do cyfrowego przetwarzania sygnałów. Procesory te, oprócz podstawowych parametrów technicznych, różnią się stopniem trudności oprogramowania zależnym od możliwości zastosowania konkretnego języka programowania i środowiska programowego dedykowanego dla danej grupy procesorów oraz wielkością nakładów pracy związanych z budową układów prototypowych. W celu

uproszczenia programowania i budowy układu produkuje się karty z procesorami i układami peryferyjnymi jak np. przetworniki analogowo-cyfrowe, realizującymi większość wymaganych przez projektanta funkcji. Jednym z takich rozwiązań jest karta EVM z procesorem zmiennoprzecinkowym TMS320C6711 firmy Texas Instruments. Karta ta, ze względu na bogactwo oferowanych funkcji jak również przyjazne dla użytkownika oprogramowanie doskonale nadaje się do zastosowań naukowo-dydaktycznych. Zawiera ona procesor TMS320C6711, którego najistotniejsze cechy przedstawiono poniżej:

- 32 – bitowe rejestry;
- możliwość wykonania 8 instrukcji w jednym cyklu zegarowym;
- częstotliwość taktowania 150MHz;
- dwa 32 – bitowe zegary taktujące;
- dwa wielokanałowe buforowane porty szeregowo;
- 32 – bitowy zewnętrzny interfejs pamięci;
- kontroler bezpośredniego dostępu do pamięci EDMA;
- 16 – bitowy port HPI do komunikacji z komputerem sterującym.

Dodatkowo karta EVM zawiera następujące układy peryferyjne: przełączniki do zadawania określonych stanów logicznych na wejściach procesora, diody elektroluminescencyjne LED, przetwornik TLC320AD535 realizujący operacje przetwarzania A/D i D/A oraz interfejs JTAG. Produkt ten w stosunku do innych dodatkowo wyróżnia przyjazne środowisko programistyczne Code Composer Studio IDE, w którym, oprócz pisania kodu źródłowego i jego kompilacji, istnieje możliwość konfiguracji procesora i jego peryferii w sposób interaktywny, z użyciem okien dialogowych i wykorzystaniem techniki „przeciągnij i upuść” (ang. „*drag & drop*”).

Dodatkowo na rynku istnieje wiele innych narzędzi programistycznych umożliwiających programowanie procesora TMS320C6711, na tle których można wyróżnić grupę środowisk zorientowanych na programowanie graficzne (np. produkt firmy Hyperception). Tworzenie kodu przy użyciu tego rodzaju narzędzi nie wymaga od projektanta znajomości języka programistycznego, jak to ma miejsce w przypadku wykorzystania kompilatorów C++ bądź komend asemblera. Polega ono na utworzeniu z dostępnych bloków struktury odpowiadającej konkretnemu algorytmowi. Jednym z takich narzędzi jest pakiet SIMULINK wykorzystujący różnego rodzaju biblioteki, a między innymi te które dedykowane są dla procesora TMS320C6711 i cyfrowych algorytmów przetwarzania sygnałów. Ponadto umożliwia on wstępne oszacowanie poprawności wykonywania konkretnego algorytmu na podstawie symulacji.

## 2. BIBLIOTEKI PAKIETU SIMULINK

SIMULINK jest zintegrowanym środowiskiem symulacyjnym MATLABa zawierającym wiele dodatkowych modułów (ang. *toolboxes*). Do najważniejszych, z punktu widzenia opisanego zagadnienia, należą:

- Real-Time Workshop;
- Embedded Target for TI C6000DSP;
- Filter Design & Analysis Tool;
- DSP Blockset.

Pierwsza z wymienionych powyżej bibliotek służy do generacji kodu odpowiadającego utworzonemu schematowi blokowemu, umożliwiając tym samym przeniesienie aplikacji na inną architekturę sprzętową. Generowany kod wynikowy jest zgodny ze standardem ANSI

C, dzięki czemu jest on niezależny od systemu docelowego. Zawiera on zarówno zmienne formatu stałoprzecinkowego (maks. 32-bitowego) jak również zmiennoprzecinkowego, dzięki czemu zapewniona jest wysoka precyzja obliczeń. Zmienne te lokowane są w programie statycznie jak i dynamicznie, dzięki czemu oszczędza się na wykorzystaniu pamięci operacyjnej systemu docelowego. Utworzony program może obsługiwać zarówno procesy jednowątkowe jak i wielowątkowe wykonywane z różną częstotliwością. Dodatkowo Real-Time Workshop umożliwia kompilowanie kodu S-funkcji, które dają szerokie możliwości tworzenia programów w środowisku MATLABa.

Biblioteka SIMULINKa DSP Blockset zawiera szereg bloków zorientowanych na zastosowanie w cyfrowym przetwarzaniu sygnałów. Bloki te są graficznymi reprezentantami efektywnych algorytmów, które mogą zostać poddane kompilacji dzięki Real-Time Workshop. Poniżej wymieniono wybrane bloki opisywanej biblioteki: źródła sygnału; bloki filtrów cyfrowych (adaptacyjne, IIR, FIR, w tym *Filter Design & Analysis Tool* – narzędzie do interaktywnego projektowania filtrów cyfrowych); bloki transformacji (FFT, IFFT, i wiele innych); bloki dedykowane do estymacji;

- reprezentacje operacji matematycznych (algebra liniowa, operacje na macierzach, wielomiany, statystyka);
- bloki zarządzania danymi w postaci sygnału i ramki-zbioru danych (buforowanie, indeksowanie, itp.).

Na uwagę zasługuje bardzo ciekawe i przyjazne użytkownikowi narzędzie *Filter Design & Analysis Tool* służące do interaktywnego projektowania filtrów cyfrowych o skończonej i nieskończonej odpowiedzi. Projektowanie filtru z użyciem wspomnianego narzędzia polega na zdefiniowaniu szeregu parametrów w tym: jego struktury, charakterystyki amplitudowej, rzędu, częstotliwości próbkowania i innych. *Filter Design & Analysis Tool* po wybraniu parametrów filtru udostępnia użytkownikowi wiele ciekawych funkcji, między innymi:

- wykreślenie charakterystyki amplitudowej i fazowej;
- wykreślenie odpowiedzi filtru na pobudzenie skokowe i impulsowe;
- obserwację położenia zer i biegunów filtru;
- możliwość podglądu i eksportu współczynników filtru cyfrowego do pliku.

Kolejną ważną pod względem szybkiego tworzenia oprogramowania jest biblioteka Embedded Target for TI C6000DSP, będąca w pewnym sensie rozszerzeniem do biblioteki DSP Blockset. Zawiera ona zbiór bloków ściśle związanych z peryferiami karty prototypowej TMS320C6711 DSK i funkcjami procesora DSP. Są to :

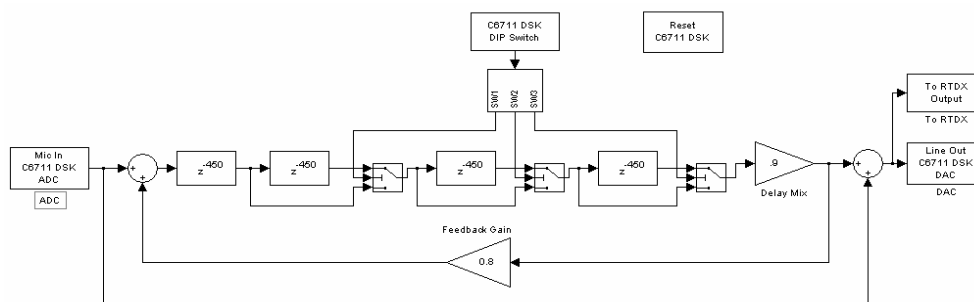
- kodek AD535 (zintegrowany przetwornik A/D i D/A);
- diody sygnalizacyjne LED; przełączniki DIP SWITCH (z narożnym, „fortepianowym” układem dźwigni przełączających) funkcja RESET;
- wymiana danych z wykorzystaniem kanałów RTDX.

Wraz z Real Time Workshop umożliwia tworzenie projektu środowiska programistycznego Code Composer Studio zawierającego kod w języku C++ odpowiadający schematowi SIMULINKa oraz ładowanie skompilowanego programu do karty prototypowej. Dzięki temu możliwe jest tworzenie kodu dla konkretnego procesora sygnałowego bez znajomości języka C++ i sposobu działania układów peryferyjnych karty prototypowej. Projektant unika w ten sposób długotrwałego procesu zapoznawania się z obsługą konkretnych elementów, które wymagają odpowiedniej obsługi, inicjalizacji itd. Przykładem może być sposób wymiany danych między procesorem sygnałowym TMS320C6711 a komputerem osobistym PC z użyciem tzw. kanałów RTDX. Wymaga on w przypadku standardowego kodowania w języku C++ umieszczenia w kodzie programu odpowiednich instrukcji

inicjalizujących te kanały, wysyłających i odbierających dane, zamykających kanały. Dodatkowo funkcje te muszą być w odpowiedniej kolejności, która zapewni prawidłową funkcjonalność. Natomiast opisywana metoda z użyciem generatora kodu Real-Time Workshop w połączeniu z biblioteką Embedded Target for TI C6000DSP wymaga jedynie umieszczenia w schemacie SIMULINKa odpowiedniego bloku („From RTDX” – do odbioru danych lub „To RTDX” – do transmitowania danych) i nadaniu właściwych parametrów w odpowiednim oknie konfiguracyjnym.

### 3. REALIZACJA PROGRAMU W ŚRODOWISKU SIMULINK NA PRZYKŁADZIE APLIKACJI PRZETWARZANIA DŹWIĘKU

Pakiet SIMULINK z wykorzystaniem bibliotek dedykowanych do kompilacji i generacji kodu dla procesora sygnałowego TMS320C6711 firmy Texas Instruments jest bardzo wydajnym narzędziem wspomagającym projektowanie. Sposób programowania z użyciem SIMULINKa opisany skrótowo poniżej przedstawiono w celu zobrazowania możliwości efektywnego, pod względem poświęconego na programowanie czasu, procesu kodowania. Posłużono się przykładem programu realizującego efekt echa o parametrach zadawanych przełącznikami.



Rys. 1. Widok przykładowego schematu pakietu SIMULINK – realizacja efektu echa i transmisja próbek wyjściowych przez interfejs RTDX.

Sygnal wyjściowy z bloku reprezentującego kodek AD535 sumowany jest z sygnałem wyjściowym opóźnionym względem aktualnego sygnału wejściowego o określoną ilość próbek zadawaną przełącznikami typu DIP SWITCH. Do odczytu stanu tych przełączników służy blok o nazwie „C6711 DSK DIP Switch”, którego wyjście steruje przełącznikami „Switch” z biblioteki DSP Blockset powodującymi włączenie do toru głównego bloku opóźniającego o zadaną ilość próbek. Sygnal wyjściowy podawany jest do bloku reprezentującego wyjście kodeka AD535 znajdującego się na płycie prototypowej zestawu DSK. Dodatkowo jest on przesyłany do komputera PC przez tzw. kanał RTDX poprzez blok o nazwie „To RTDX”. Na schemacie blokowym umieszczono również blok o nazwie „Reset C6711 DSK”, dzięki czemu istnieje możliwość wymuszenia restartu programu z poziomu SIMULINKa, poprzez kliknięcie myszką w jego obszar. Tak utworzony schemat po niewielkiej zmianie, mianowicie po zastąpieniu bloków dedykowanych dla procesora serii TMS innymi blokami pakietu SIMULINK, może zostać poddany wstępnej symulacji dla oszacowania poprawności wykonywania algorytmu.

Tworzenie kodu wynikowego po utworzeniu schematu blokowego polega na wybraniu w menu głównym SIMULINKa opcji „Simulation Parameters” i wskazania na zakładce „Real-Time Workshop” na rozwijanej liście o nazwie „Category” opcji „TI C6000 code generation”. Po wybraniu przycisku „Build & Run” zostanie utworzony projekt środowiska Code Composer Studio, który zostanie automatycznie skompilowany i wczytany do procesora sygnałowego. Zakłada się, iż podczas wykonywanego procesu zainstalowane są wszystkie niezbędne do tego składniki, gdyż tylko wtedy opisana operacja zakończy się powodzeniem.

#### 4. WNIOSKI KOŃCOWE

Zmiennoprzecinkowy procesor sygnałowy TMS320C6711 oferuje duże możliwości obliczeniowe oraz jest programowalny przy użyciu języków wysokiego poziomu. Dzięki temu w stosunkowo prosty sposób i szybko można zaimplementować dość złożone algorytmy cyfrowe.

Dzięki wykorzystaniu możliwości pakietu SIMULINK w procesie generacji kodu procesora sygnałowego, symulacje komputerowe przeprowadzane w tym programie można w prosty sposób skonfrontować z rzeczywistością. Pozwala to na tworzenie oprogramowania metodą graficzną poprzez „budowanie” schematu blokowego techniką „*drag&drop*”. Zaletami tej metody są przede wszystkim prostota tworzenia oprogramowania, a w konsekwencji oszczędność czasu poświęconego na tworzenie kodu, jak również czytelność samego schematu dla osób trzecich. Do wad można by zaliczyć możliwość dalszej rozbudowy wygenerowanego automatycznie kodu, która wymaga od programisty pewnej znajomości sposobu przydzielania etykiet w programie oraz brak bloków obsługujących inne elementy systemu procesora TMS320C6711 takie jak np. port McBSP czy interfejs dostępu do pamięci zewnętrznej EMIF.

W dzisiejszych czasach coraz częściej potrzebne są środowiska do szybkiego prototypowania dzięki czemu we wczesnej fazie można decydować o powodzeniu projektu. Stąd też coraz bardziej upowszechnia się metoda graficznego programowania o czym świadczy dostępność na rynku coraz szerszej gamy tego typu narzędzi. Pakiet SIMULINK ciągle zostaje uzupełniany o nowe biblioteki dedykowane do innych procesorów jak również istniejące biblioteki uzupełniane są o dodatkowe funkcje. Na tle środowisk do graficznego programowania procesora TMS320C6711 SIMULINK, oprócz możliwości efektywnego tworzenia kodu, wyróżnia się przyjaznym interfejsem udostępniającym projektantowi opcję symulacji tworzonych algorytmów.

#### 5. BIBLIOGRAFIA

1. „TMS320C6000 Code Composer Studio Help”, Texas Instruments 2000;
2. „TMS320C6000 Peripherals Reference Guide”, Texas Instruments 2001;
3. „Embedded Target for the TI TMS320C6000 DSP Platform User’s Guide”, version 2, The MathWorks;
4. „Embedded Target for TI C6000 DSP 2.0 Release Notes”, The MathWorks;
5. „Real-Time Workshop User’s Guide”, version 6, The MathWorks;
6. “Filter Design Toolbox User’s Guide”, version 3, The MathWorks.

### **EFFECTIVE PROGRAMMING OF TMS320C6711 DIGITAL SIGNAL PROCESSOR USING SIMULINK TOOLBOXES**

The use of SIMULINK toolboxes for the rapid code development for digital signal processors (DSPs) is discussed. The target hardware platform is a development starter kit (DSK) from Texas Instruments, featuring a TMS320C6711 floating point device. The properties of the platform are outlined regarding its applications for rapid prototyping and scientific research. The functionalities of relevant SIMULINK toolboxes are reviewed briefly. A detailed example of code development is provided, the application used for the example being an audio reverberation algorithm. The code provides an interface to DIP switches and real-time communication with the host PC. The advantages and disadvantages of SIMULINK-based graphical programming are also discussed. This kind of programming approach can be particularly useful when development time is at a premium. It does not require extensive knowledge of C/C++ programming.