Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 19

XIII Seminarium
ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE 2003
Oddział Gdański PTETiS

KOMUNIKACJA MIĘDZY KOMPUTEREM A PROCESOREM SYGNAŁOWYM Z ZASTOSOWANIEM PROTOKOŁU RTDX

Grzegorz GRABOWSKI[1], Wojciech ŚLESZYŃSKI[2], Artur CICHOWSKI[3]

Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk,
1. tel. 347-25-34, fax: 341-08-80 e-mail: ggrab@ely.pg.gda.pl
2. tel. 347-25-34, fax: 341-08-80 e-mail: wslesz@ely.pg.gda.pl
3. tel. 347-25-48, fax: 341-08-80 e-mail: arcich@ely.pg.gda.pl

W artykule opisano możliwości realizacji komunikacji między komputerem osobistym a procesorem sygnałowym TMS320C6711 firmy Texas Instruments. Scharakteryzowano własności karty uruchomieniowej ze zmiennoprzecinkowym procesorem sygnałowym TMS320C6711 pod kątem zastosowań dydaktyczno-naukowych. Przedstawiono przykłady praktycznych rozwiązań procedur komunikacyjnych, utworzonych w środowiskach programistycznych Visual C++ oraz LabView, w systemach cyfrowego przetwarzania sygnałów. Omówiono szerzej realizację komunikacji z wykorzystaniem protokołu RTDX (ang. *Real Time Data Exchange*). Opisano możliwości oprogramowania transmisji między komputerem PC a procesorem sygnałowym TMS320C6711 używając bibliotek dołączanych dynamicznie (DLL), klasy obiektów lub gotowych elementów środowiska SIMULINK.
Zaprezentowane rozwiązania są interesującym elementem, który mógłby zostać wykorzystany w ramach zajęć z cyfrowego przetwarzania sygnałów.

1. WPROWADZENIE

Rozwój współczesnej techniki opartej na systemach mikroprocesorowych przyczynił się do powstania wielu różnorodnych układów dedykowanych do przetwarzania cyfrowego. 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 dziedziny należą procesory sygnałowe (skrótowo oznaczane DSP) dedykowane do cyfrowego przetwarzania sygnałów. Procesory te, oprócz podstawowych parametrów technicznych, o których wspomniano na wstępie, różni stopień trudności oprogramowania, zależny od możliwości zastosowania konkretnego języka programistycznego i środowiska programowego dedykowanego dla danej grupy procesorów oraz wielkość nakładów pracy związanych z budową układów prototypowych. W celu uproszczenia programowania i budowy układu produkuje się karty z procesorami i innymi układami peryferyjnymi jak np. przetworniki analogowo-cyfrowe, realizującymi większość wymaganych przez projektanta funkcji. Jednym z takich rozwiązań jest karta DSK z procesorem zmiennoprzecinkowym TMS320C6711 firmy Texas Instruments. Karta

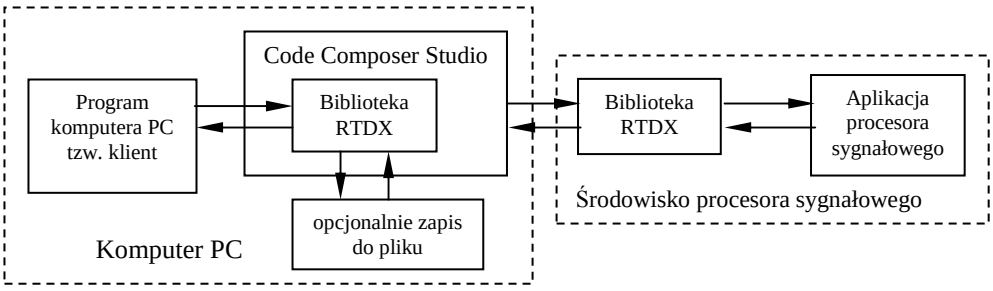
ta ze względu na swoje bogactwo możliwości jak również przyjazne dla użytkownika oprogramowanie doskonale nadaje się do zastosowań naukowo–dydaktycznych. Zawiera ona procesor TMS320C6711, którego najistotniejsze cechy przedstawiono poniżej:

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

Dodatkowo karta DSK zawiera układy peryferyjne: przełączniki do zadawania określonych stanów logicznych na wejścia procesora, diody elektroluminescencyjne LED, kodek 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, 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" (*z ang. „drag & drop"*). Kolejną zaletą opisywanego układu jest interfejs RTDX umożliwiający wymianę danych między procesorem sygnałowym a komputerem PC w czasie rzeczywistym. Dzięki temu programista ma możliwość obserwacji zmiennych programu, co w połączeniu ze stosunkowo prostym procesem tworzenia oprogramowania czyni opisywaną kartę wydajnym narzędziem w realizacji prototypu, bądź w zastosowaniach naukowych przy wszelkiego rodzaju badaniach, symulacjach itp.

2. INTERFEJS RTDX

Interfejs RTDX (*ang. "real time data exchange"*) dostarcza użytkownikowi możliwość wymiany danych i dzięki temu pozwala na ich obserwację w czasie działania programu. Transmisja między aplikacjami docelową uruchomioną na procesorze TMS i zewnętrzną komputera PC odbywa się w trybie „on-line" i nie powoduje przerw w działaniu programu. Sfera programowa interfejsu składa się z dwóch aplikacji: docelowej i sterującej uruchomionej na komputerze PC, działającej w połączeniu z Code Composer Studio.



Rys. 1. Schemat procesu komunikacji

Programem klientem może być aplikacja standardowych pakietów takich jak: National Instruments LabVIEW™, Quinn-Curtis Real-Time Graphics Tools, MATLAB, Microsoft

Excel. Dodatkowo można tworzyć własne aplikacje w środowiskach programistycznych Visual C++ oraz Visual Basic, dla których zostały przygotowane właściwe biblioteki. Dzięki temu pozostawia się programiście duże możliwości rozbudowy własnych aplikacji.
Interfejs RTDX jest doskonałym narzędziem do rozmaitych zastosowań związanych z kontrolą określonych parametrów, bądź ich zadawaniem. Może okazać się bardzo przydatny w układach sterowania, umożliwiając użytkownikowi podgląd bieżących wartości systemu, ich zadawanie lub rozdzielenie realizowanego programu na kilka części uruchomionych na oddzielnych platformach i komunikujących się przy pomocy interfejsu RTDX, co wpływa istotnie na wydajność systemu. Pozwala on także na budowę przyjaznych aplikacji – panelów operatorsko-pomiarowych uruchamianych na komputerze PC monitorujących pracę systemu.

Szybkość transmisji danych przy użyciu interfejsu RTDX dla procesora TMS320C6711 wynosi 20÷50 kilobajtów na sekundę. Dane transmitowane są poprzez tzw. kanały, które wymagają wcześniejszej deklaracji. W zależności od kierunku transmisji danych deklarowane są kanały wejściowe (do odbioru) bądź wyjściowe (do wysyłania).
Strumień danych wysyłany (bądź odbierany) z procesora sygnałowego jest odpowiednio formatowany przy użyciu biblioteki RTDX Target Library, w której umieszczone zostały wszelkie procedury związane z deklaracją kanałów transmisyjnych, monitoringiem i transmisją danych. Dane transmitowane są pomiędzy systemami przy użyciu interfejsu JTAG. Aplikacja kliencka uczestniczy w procesie wymiany danych z wykorzystaniem środowiska Code Composer Studio, umożliwiającego aktywację poszczególnych kanałów transmisyjnych, ich dodawanie, zapis transmitowanych danych do pliku i szereg innych działań konfiguracyjnych. Podstawowe funkcje obsługi RTDX dostarczone przez producenta w formie biblioteki DLL scharakteryzwano skrótowo poniżej:

- Enable/DisableChannel (“nazwa kanału RTDX”) – aktywacja kanału o podanej nazwie;
- channel.Open (“nazwa kanału RTDX”, “R/W”) – otworzenie kanału przepływu danych do odczytu bądź zapisu (w zależności od znacznika R/W);
- channel.ReadI1/I2/I4/F4/F8 (data) – odczyt pojedynczej danej i umieszczenie jej pod zmiennąo nazwie data;
- channel.ReadSAI1/I2/I4/F4/F8 (data_array) – odczyt tablicy danych i umieszczenie jej pod adresem zmiennej tablicowej o nazwie data_array
- channel.WriteI1/I2/I4/F4/F8 (datum, num_bytes_in_RTDX_DLL_buffer) – zapis zmiennej do bufora i wysłanie przez kanał RTDX;
- channel.Write (data_array, num_bytes_in_RTDX_DLL_buffer) – instrukcja służąca do transmisji zmiennej tablicowej;
- channel.Close () – zamykanie kanału RTDX.

Wyżej przedstawione funkcje nie wyczerpują wszystkich możliwości wykorzystania narzędzia programistycznego RTDX, a jedynie przedstawiają podstawowe operacje możliwe do zastosowania. Pominięto tu głównie funkcje inicjalizacji sterownika, których wywołanie jest niezbędne do prawidłowego działania.

3. OPIS KLASY RTDX

Proces oprogramowania komputera PC z użyciem biblioteki RTDX, mimo przygotowanych przez producenta funkcji, może sprawiać problemy przy większych projektach, gdyż wymaga zastosowania odpowiednich funkcji wywołanych we właściwej kolejności. Własność ta staje się przyczyną wielu błędów i powoduje wydłużenie procesu

programowania. Wymaga ono od programisty dokładnej znajomości każdej z funkcji biblioteki RTDX i umieszczania w wymaganej kolejności.
W celu eliminacji niedogodności związanej z programowaniem w sposób strukturalny w ramach badań utworzono klasę obiektu RTDX, której funkcje publiczne wywołują automatycznie odpowiednie procedury. Klasa ta obsługuje zarówno proces odbioru danych, jak i ich wysyłania.
Utworzona klasa RTDX składa się z konstruktora klasy, którego wywołanie wymaga podania typu tworzonego kanału, gdyż w zależności od tego uruchamiane są odpowiednie funkcje jego inicjalizacji. Dostępne rodzaje kanałów to:

- ArrayRead – kanał odczytu tablicy zmiennych;
- ArrayWrite - kanał zapisu tablicy zmiennych;
- ValueRead - kanał odczytu pojedynczej zmiennej;
- ValueWrite - kanał zapisu pojedynczej zmiennej.

Każde wywołanie konstruktora powoduje zwiększenie zmiennej statycznej klasy m_numChannel będącej licznikiem obiektów klasy. Zmienna ta jest wykorzystywana do kontroli ilości obiektów w celu uruchomienia funkcji inicjalizacji interfejsu RTDX, które powinno być dokonane jednorazowo przed wykorzystaniem kanałów komunikacyjnych.
Kolejną funkcją klasy o dostępie publicznym jest funkcja openChannel, która zawiera wszystkie potrzebne dla danego kanału procedury inicjalizacji. Przeciwieństwem tej funkcji jest procedura „zamykania" kanałów CloseChannel, w której zgrupowano funkcje zwalniające pamięć zajmowaną przez obiekt klasy. Wysyłania zmiennych bądź ich odbioru dokonuje się poprzez wywołanie odpowiednich funkcji składowych:

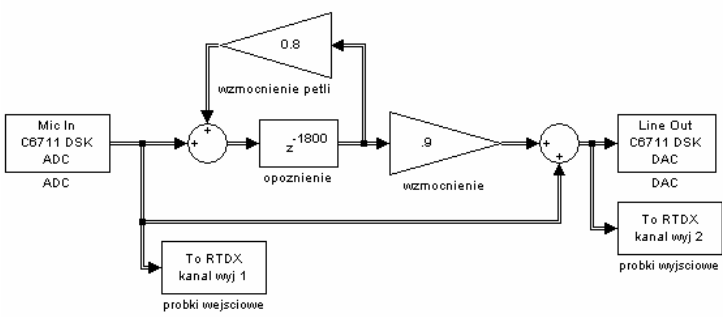
- writeArray – jej wywołanie wymaga podania tablicy i ilości wysyłanych elementów;
- readArray – argumentem jest tablica, w której zostaną umieszczone odczytane dane;
- readValue – funkcja odczytu pojedynczej danej, wymagająca podania zmiennej, w której umieszczona zostanie odczytana wartość;
- writeValue - funkcja wysyłania pojedynczej zmiennej, będącego argumentem.

Wywołania tych funkcji zostały zabezpieczone przed nieprawidłowym użyciem – żądaniem wysłania zmiennej określonego typu przez kanał, który nie jest do tego celu dedykowany. Wszystkie funkcje służące transmisji danych zostały przeciążone, dzięki czemu ich zastosowanie powoduje wywołanie procedur dopasowanych do formatu jej argumentów. Destruktor klasy RTDX (~RTDX()) składa się z funkcji zmniejszającej licznik obiektów klasy oraz funkcji powodującej deaktywację interfejsu RTDX, która wywołana jest tylko w przypadku „zamknięcia" wszystkich kanałów RTDX. Wywoływany jest on automatycznie i każdorazowo, gdy program wychodzi poza zasięg danego obiektu. Wykorzystanie klasy powoduje zmniejszenie nakładów pracy związanej z programowaniem, które dzięki niej staje się bardziej intuicyjne (ze względu na ilość funkcji i ich nazewnictwo). Utworzoną klasę RTDX wykorzystano do oprogramowania procesu komunikacji między procesorem sygnałowym zintegrowanym z kartą DSK a aplikacją kliencką komputera PC.

4. INTERFEJS RTDX W ŚRODOWISKU SIMULINK

Interfejs RTDX, przeznaczony jest do wielu środowisk między innymi również do pakietu SIMULINK środowiska MATLAB. SIMULINK jest podprogramem MATLAB'a dedykowanym do symulacji, w którym wszystkie realizowane funkcje zawarte są w odpowiednich blokach reprezentujących określoną funkcję. Tak jest także w przypadku procedur komunikacyjnych interfejsu RTDX, tzn. są dwa bloki, z których jeden służy do

wysyłania danych („To RTDX”), a drugi do ich odbioru („From RTDX”). Dodatkową zaletą wykorzystania środowiska SIMULINK jest możliwość generacji kodu źródłowego programu procesora sygnałowego TMS320C6711. Dzięki temu mamy możliwość tworzenia programów na procesory firmy Texas Instruments bez znajomości języka C++, a jedynie przy minimum wiedzy niezbędnym do obsługi środowiska SIMULINK. Dodatkowo SIMULINK pozwala na szybkie tworzenie wykresów i okien dialogowych umożliwiających obserwację zmiennych odbieranych z „kanałów” RTDX.



Rys. 2. Widok przykładowego schematu symulacyjnego SIMULINK – realizacja efektu echa i transmisja próbek we/wy przez interfejs RTDX.

Przedstawione możliwości pakietu SIMULINK świadczą o jego dużej użyteczności w zastosowaniach dydaktycznych szczególnie w ramach zajęć z zakresu cyfrowego przetwarzania sygnałów. Dodatkowo dzięki możliwości generacji kodu na procesor sygnałowy, działania z zakresu symulacji osiągają wymiar praktyczny, a sam proces tworzenia schematu symulacyjnego można uznać za pewnego rodzaju sposób programowania (podobnie jak w przypadku środowiska LabVIEW).

5. WYKORZYSTANIE INTERFEJSU RTDX W ŚRODOWISKU LABVIEW

Protokół RTDX wykorzystano do realizacji komunikacji pomiędzy komputerem, a procesorem sygnałowym w laboratoryjnym układzie napędowym. Składa się on z silnika indukcyjnego o mocy do 2,2kW obciążonego prądnicą prądu stałego oraz przemiennika częstotliwości sterowanego przy pomocy procesora sygnałowego TMS320C6711 i układu programowalnego ALTERA serii ACEX 1K. Do sterowania falownikiem opracowano dwa różne panele operatora, które korzystają z tych samych funkcji komunikacyjnych. Oba panele wykonano w środowisku programowania LabVIEW, ze względu na dużą bibliotekę gotowych przyrządów pomiarowych oraz łatwość tworzenia interfejsów użytkownika o wysokich walorach estetycznych. Panel operatorski wykorzystywany do diagnostyki uszkodzeń tranzystorów falownika napięcia umożliwia sterowanie pracą falownika napięcia, zadawanie podstawowych parametrów modulacji takich jak: współczynnik głębokości modulacji, szybkość jego zmian, częstotliwość modulacji, itp. Pozwala na przerywanie sygnałów bramkowych poszczególnych tranzystorów i wyświetla wyniki diagnostyki.W celu realizacji komunikacji w programie LabVIEW wykorzystano funkcję „*Call Library Function”* umożliwiającą wywoływanie funkcji ze standardowych bądź stworzonych przez użytkownika bibliotek [5]. W tym wypadku w środowisku Visual C++ utworzono własną bibliotekę dołączaną dynamicznie *rtdx_pr.dll*, wykorzystując wspomnianą wcześniej bibliotekę *rtdxint.dll*. Utworzona biblioteka zawiera następujące

funkcje: deklaracja kanału RTDX, otwarcie i zamknięcie kanału do odczytu bądź zapisu, odczyt i zapis zmiennej całkowitej lub rzeczywistej oraz zwolnienie kanału RTDX.

6. 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. Opracowane funkcje komunikacyjne umożliwiają szybkie tworzenie własnych aplikacji do sterowania procesorem sygnałowym w przypadku wykorzystania środowisk programowania Quinn-Curtis Real-Time Graphics Tools oraz LabVIEW.
Również możliwości pakietu SIMULINK świadczą o jego dużej użyteczności w zastosowaniach dydaktycznych, szczególnie w ramach zajęć z zakresu cyfrowego przetwarzania sygnałów. Dzięki możliwości generacji kodu procesora sygnałowego, symulacje komputerowe przeprowadzane w tym programie można w prosty sposób skonfrontować z rzeczywistością, a sam proces tworzenia schematu symulacyjnego można uznać za pewnego rodzaju programowanie.

7. BIBLIOGRAFIA

1. „TMS320C6000 Code Composer Studio Help”, Texas Instruments 2000;
2. „TMS320C6000 Peripherals Reference Guide”, Texas Instruments 2001;
3. „Visual C++ 6.0 dla każdego”, Davis Chapman, wyd. Helion, W-wa 1999;
4. „Visual C++ 6. Vademecum profesjonalisty”, Richard C. Leinecker, Tom Archer, wyd. Helion, W-wa 2000;
5. „Using external code in LabVIEW”, National Instruments 2000.

COMMUNICATION BETWEEN PERSONAL COMPUTER AND DIGITAL SIGNAL PROCESSOR USING RTDX PROTOCOL

In this paper the implementation of communication method between personal computer (PC) and digital signal processor (DSP) TMS320C6711 from Texas Instruments is presented. Properties of development starter kit (DSK) and on board floating point DSP TMS320C6711 are characterized regarding their applications for educational and scientific purposes. It contains examples of practical communication procedures solutions, created in programming environments such as MS Visual C++ and LabView in the digital signal processing systems. The use of the RTDX protocol for communication implementation is deeply covered, as well as the possibilities of implementing the transmission between the PC and the TMS320C6711 digital signal processor. The transmission is programmed using dynamic link libraries (DLL), custom classes or common SIMULINK components.
The solutions presented above can be very useful during lectures or other teaching activities on digital signal processing.