Laboratorium dydaktyczne programowania systemów mikroprocesorowych rodziny Motorola 680x0 w języku asemblera

Dr inż. Witold Marańda

12 października 2000

Spis treści

1 Wstęp

Katedra Mikroelektroniki i Technik Informatycznych posiada od 1995 roku laboratorium dydaktyczne służące do praktycznej nauki programowania systemów mikroprocesorowych w języku asemblera. Laboratorium powstało w ramach współpracy z firmą Motorola, która przekazała na rzecz Katedry mikroprocesorowe płyty prototypowe IDP M68EC0x0 stanowiące zasadnicze wyposażenie laboratorium. Ponadto dostępna jest szczegółowa dokumentacja tych płyt, procesorów rodziny 680x0 oraz języka asemblera.

Laboratorium jest wykorzystywane do nauczania przedmiotów prowadzonych na kierunku Elektronika („Architektura komputerów” oraz „Projektowanie i uruchamianie systemów mikroprocesorowych”), Informatyka („Metody i jezyki programowania”) oraz International Faculty of Engineering („Advanced Digital Systems”) na wydziale Elektroniki i Elektrotechniki Politechniki Łódzkiej. Zakres i tematyka ćwiczeń są dostosowane indywidualnie do programu różnych przedmiotów.

Ponadto sprzęt laboratorium jest również wykorzystywany do realizacji prac magisterskich w zakresie systemów wbudowanych oraz procesorów sygnałowych.

2 Zastosowania płyt prototypowych

2.1 Testowanie mikroprocesorów

Ze względu na umieszczenie mikroprocesora w osobnym module, możliwe jest testowanie poszczególnych egzemplarzy. Do wykonania zaawansowanych testów lub pomiarów możliwe jest wykorzystanie niebuforowanych sygnałów procesora dostępnych na złączu lokalnym.

Ponadto możliwe są do przeprowadzenia testy wydajności procesora w systemie poprzez konfigurację pamięci operacyjnej (relokacja obszarów, ustawienie wait-states) oraz dostęp do urządzeń zewnętrzych 8, 16 lub 32 bitowych.

Dostępny w systemie 24-bitowy timer pozwala na prezycjne pomiary wydajności.

2.2 Tworzenie i testowanie oprogramowania

Tworzenie i testowanie oprogramowania wspierane jest przez dwa programy uruchomieniowe, istniejące w pamieci ROM systemu IDP M68EC0x0.

Umożliwiają one użytkownikowi inspekcję i zapis do pamięci systemu oraz wewnętrznych rejestrów mikroprocesora, asemblację i disasemblację kodu oraz ładowanie programów do systemu przez port szeregowy. Dodatkowo możliwe jest uruchamiania programów w trybie pracy krokowej i ustawianie punktów przerwań.

2.3 Budowa i testowanie sprzętu

Złącza rozszerzeń płyty prototypowej pozwalają na dołączenie dodatkowego sprzętu i tworzenie współpracującego z nim oprogramowania. Zakres rozbudowy może być bardzo szeroki, od sterowania urządzeniami aż do rozbudowanego systemu komputerowego.

Program startujący płyty może być skonfigurowany w tryb automatycznego uruchamiania systemu operacyjnego z pamięci ROM (napisanego przez użytkownika lub dostępnych opcjonalnie XRAY+, pSOS+ firmy Motorola).

Magistrala płyty prototypowej zapewnia synchroniczny 32-bitowy transfer danych (magistrala adresowa ma 28 bitów) z predkością 12.5MHz lub 25.0MHz.

3 Wyposażenie laboratorium

3.1 Sprzęt

Wyposażenie laboratorium stanowią płyty prototypowe Motorola IDP M68EC0x0, wyposażone w wymienialne moduły procesorów rodziny 680x0. W laboratorium dostępne są moduły z procesorami 68030 oraz 68040. Płyty prototypowe są skonstrułowane pod kątem ułatwienia testowania orogramowania oraz rozbudowy sprzętowej systemu, ze względu na wbudowane programy uruchomieniowe oraz złącza rozszerzeń systemu.

Płyty IDP M68EC0x0 mają następujące parametry:

- złącze modułu procesora
- 5 złączy rozszerzeń systemu IDP M68EC0x0
- 2MB pamięci DRAM w podstawkach (MCM514256)
- dwie podstawki na pamięć EPROM do 1MB
- zegar czasu rzeczywistego z podtrzymywaniem bateryjnym (MK48T02)
- dwa porty szeregowe RS232C (MC68681)
- 24-bitowy timer (MC68230)
- port równoległy (MC68230)
- 7-segmentowy wyświetlacz LED
- przełącznik RESET i ABORT
- dwa złącza zasilania
- pamięć ROM z dwoma programami uruchomieniowymi

Moduł procesora posiada następujace cechy:

- procesor 68020/68030/68040
- procesor arytmetyczny 68882 (tylko dla 68020/68030)
- programowany układ translacji adresów, niezależny od procesora
- układ arbitracji magistrali systemowej
- układ arbitracji przerwań,
- lokalne złącze rozszerzenia modułu

Uzupełnieniem laboratorium są komputery PC pracujące pod kontrolą systemu MSDOS lub Windows 9x, wykorzystywane do edycji kodów źródłowych programów, kompilacji, asemblacji oraz transferu plików do systemu IDP M68EC0x0. Ponadto, komputery te służą jako terminale dla programów uruchomieniowych pracujących na płytach prototypowych.

3.2 Dokumentacja

W laboratorium dostępna jest literatura w języku angielskim dotycząca płyt prototypowych IDP M68EC0x0, języka asemblera mikroprocesorów rodziny Motorola 680x0 oraz specjalizowanych układów wejścia/wyjścia:

- M68000 Family Programmer's Reference Manual
- M68EC0x0 Integrated Development Platform User's Manual
- M68040 Microprocessor User's Manual

- M68030 32-bit Embedded Controller User's Manual
- M68020 Microprocessor User's Manual
- MC68681 Dual Asynchronous Receiver/Transmitter (DUART)
- MC68230 Parallel Interface/Timer (PI/T)

3.3 Oprogramowanie

Oprogramowanie dostepne w laboratorium pozwala na tworzenie, uruchamianie i testowanie programów dla płyt prototypowych. Programy dla systemu IDP M68EC0x0 są tworzone w wygodnym dla programisty środowisku MSDOS, Windows lub Linux. Programy mogą być pisane w języku C lub w asemblerze, a następnie po kompilacji, przesyłane łączem szeregowym do płyty. Następnie, programy mogą być uruchamiane i testowane pod kontrolą programów uruchomieniowych znajdujących się w pamięci ROM systemu IDP M68EC0x0.

System operacyjny płyty komunikuje się z użytkownikiem przez port szeregowy, tak więc do komunikacji z programami pracującymi w systemie IDP M68EC0x0 można wykorzystać dowolny system komputerowy z programem terminala szeregowego.

3.3.1 Kompilator języka C

Kompilator języka C, dostępny w laboratorium, jest kompilatorem skrośnym pracującym w środowisku MSDOS. Ograniczenia możliwości języka dotyczą zasadniczo funkcji bibliotecznych wejścia/wyjścia, które muszą odzwierciedlać specyfikę systemu IDP M68EC0x0.

3.3.2 Asembler

Asembler wykorzystywany w laboratorium jest asemblerem skrośnym pracującym w środowisku MSDOS, generującym kod dla mikroprocesorów Motorola 68000 i 68010 w formacie H68 przystosowanym do transmisji szeregowej. Dyrektywy asemblera oferują podstawową funkcjonalność, jednak bez bardziej zaawansowanych możliwości jak asemblacja warunkowa lub automatyczne dołączanie plików źródłowych.

3.3.3 Programy uruchomieniowe

Testowanie oprogramowania w systemie IDP M68EC0x0 wspierane jest przez dwa programy uruchomieniowe, istniejące w pamieci ROM systemu. Możliwe jest dowolne przełączanie pomiedzy tymi monitorami w czasie testowania oprogramowania.

Podstawowym monitorem jest program ROM68. Umożliwia on użytkownikowi inspekcję i zapis do pamięci systemu oraz wewnętrznych rejestrów mikroprocesora, asemblację i disasemblację kodu oraz ładowanie programów przez port szeregowy. Dodatkowo program umożliwia uruchamiania programów w trybie pracy krokowej i ustawianie punktów przerwań. Tabela 1 przedstawia skrótowo polecenia programu ROM68.

Drugi monitor MON68 dostępny na płycie posiada bardziej rozbudowane możliwości śledzenia programów i może współpracować bezposrednio z zaawansowanym programem uruchomieniowym XDB firmy Intermetrics, dostarczanym jako opcja przez Motorolę.

Tablica 1: Lista poleceń programu ROM68

Polecenia dotyczące pamięci systemu	
`AS`	Asembluj do pamięci
`CM`	Porównaj pamięć
`DI`	Disasembluj pamięć
`DM`	Wyświetl pamięć
`FM`	Wypełnij pamięć
`MM`	Kopiuj pamięć
`MP`	Mapuj pamięć
`PM`	Zapisz do pamięci
`SM`	Przeszukaj pamięć
Polecenia dotyczące rejestrów mikroprocesora	
`DO`	Wyświetl rejestry dodatkowe
`DR`	Wyświetl zawartość rejestrów
`PR`	Zapisz do rejestrów
Polecenia związane z uruchamianiem programów	
`CB`	Usuń punkt przerwań
`DB`	Ustaw punkt przerwań
`GO`	Uruchom program
`LB`	Wyświetl listę punktów przerwań
`ST`	Wykonaj instrukcję
Polecenia związane z przesyłaniem danych	
`BI`	Uruchom program monitora MON68
`DC`	Załaduj plik w postaci S-rekordów
Inne polecenia	
`EC`	Wyświetl dane szestnastkowo i dziesiętnie
`HC`	Konfiguracja sprzętu
`HE`	Wyświetl opis poleceń

4 Ćwiczenia dydaktyczne

Praca w laboratorium polega na pisaniu programów w języku asemblera, oraz ich kompilacji na komputerze PC. Następnie, pliki wynikowe przesyłane są do płyty prototypowej i uruchamiany jest program monitora. Od tego momentu komputer PC jest konsolą dla systemu IDP M68EC0x0. Użytkownik może teraz przeprowadzić sesję uruchomieniową korzystając z programu ROM68 lub MON68. W każdym momencie możliwe jest przełączanie pomiedzy pracą w systemie IDP M68EC0x0, a edycją i kompilacją programów.

Praca z płyta prototypową ma kilka istotnych zalet w porównaniu do wykorzytywania symulatora danego mikroprocesora. Przede wszystkim możliwe jest uruchamianie i testowania programów wykorzystujących bardziej zaawansowane mechanizmy (np. przerwania) lub zależności czasowe, oraz współpracujących z urządzeniami zewnętrznymi. Ponadto użycie rzeczywistego programu uruchomieniowego pozwala na lepsze poznanie specifiki tego rodzaju pracy.

4.1 Ćwiczenia podstawowe

Podstawowy zestaw ćwiczeń służy wyrobieniu umiejętności w posługiwaniu się narzędziami programistycznymi tj. asemblerem i programem uruchomieniowym. Studenci muszą opanować składnię języka asemblera, jego dyrektywy oraz sposób oznaczania trybów adresowania rozkazów mikroprocesora. Istotna jest również znajomość tworzenia i dostępu do struktur danych w języku asemblera.

Duże znaczenie ma opanowanie pracy z programem uruchomieniowym, tj. uruchomienie programu w trybie pracy krokowej, zakładanie pułapek oraz inspekcja rejestrów wewnętrznych mikroprocesora i pamięci.

Ćwiczenia wstępne dotyczą zwykle podstawowych operacji wejścia/wyjścia (zwłaszcza dla programów w języku C) oraz dostępu do rejestrów pamięci systemu IDP M68EC0x0. Większą atrakcyjność takich ćwiczeń można uzyskać wykorzystując układ zegara czasu rzeczywistego oraz 7-segmentowy wyświetlacz LED płyty.
Oto przykłady ćwiczeń (język C i asembler):

- pisanie i czytanie znaków na terminalu
- wyświetlanie cyfr na wyświetlaczu LED
- wyświetlanie prostych „animacji" na wyświetlaczu LED
- wypisywanie danych zegara czasu rzeczywistego na terminalu
- wyświetlanie stanu zegara na wyświetlaczu LED

Wykonywanie identycznych ćwieczeń w języku C i w języku asemblera jest cenne z dydaktycznego punktu widzenia, gdyż umożliwia porównanie efektywności i długości kodu, komfortu programowania oraz łatwości uruchamiania tworzonego oprogramowania. Pozwala to studentom na bardziej trafny dobór narzędzi programistycznych do określonych zadań.

4.2 Programowanie urządzeń we/wy

Ćwiczenia programowania urządzeń wejścia/wyjścia dotyczą portu szeregowego oraz równoległego. Ich zadaniem jest zapoznanie studentów ze specifiką dostępu do danych (np. polling) oraz z pracą ze specjalizowanymi układami MC68681 (port szeregowy), MC68230 (port równoległy) oraz MK48T02 (zegar czasu rzeczywistego).

Istotne znaczenie ma tutaj umiejętność korzystania z dokumentacji technicznej w celu poprawnego konfigurowania parametrów pracy układów wejścia/wyjścia.
Przykładowe ćwiczenia dotyczą następujących zagadnień:

- wyświetlanie i czytanie znaków z terminala
- wyświetlanie i formatowanie czasu i daty na terminalu
- ustawianie zegara czasu rzeczywistego z terminala
- buforowanie wprowadzanego i wyświetlanego tekstu
- konfiguracja trybów pracy portu szeregowego

- komunikacja pomiędzy dwoma płytami prototypowymi
- generowanie dźwięków poprzez wysyłanie na port równoległy danych (dołączony 8-bitowy przetwornik D/A oraz głośnik)

4.3 Ćwiczenia zaawansowane

Bardziej zaawansowane ćwiczenia dotyczą głównie operowania na strukturach danych, wykorzystania urządzeń we/wy razem z programowaniem systemu przerwań mikroprocesora.

Zasadniczego znaczenia nabiera tutaj możliwość uruchamiania oraz śledzenia programów za pomocą programów uruchomieniowych systemu IDP M68EC0x0.

Ćwiczenia te realizują często zadania praktyczne i są wstępem do indywidualnych projektów studenckich takich jak elementy systemów operacyjnych lub sterowanie urządzeniami zewnętrznymi.

Przykładowe ćwiczenia dotyczą następujących zagadnień:

- obsługa portu szeregowego (czytanie i pisanie z terminala) z wykorzystamiem przerwań
- wyświetlanie czasu i daty w określonych odstępach czasowych — programowanie timera
- generowanie dźwięków o określonej częstotliwości poprzez wysyłanie na port równoległy danych — programowanie przerwań i timera (dołączony 8-bitowy przetwornik D/A oraz głosnik)
- realizacja prostego instrumentu klawiszowego — generacja dźwięków i obsługa klawiatury
- sterowanie drukarką podłączoną do portu równoległego
- buforowanie danych napływających przez dwa porty szeregowe i drukowanie ich na drukarce podłączonej do portu równoleglego
- realizacja sterownika myszki typu Bus Mouse podłączonej do portu równoległego