STUDIA INFORMATICA 2002
Volume 23 Number 3 (50)

Bartłomiej ZIELIŃSKI, Krzysztof TOKARZ
Politechnika Śląska, Instytut Informatyki

EFEKTYWNOŚĆ MIKROSTEROWNIKÓW RODZINY 8051 W ZASTOSOWANIACH SIECIOWYCH

Streszczenie. Zbudowano model konwertera protokołów z zastosowaniem mikrosterownika rodziny 8051. Zbadano zależność wydajności konwertera od wersji mikrosterownika i częstotliwości jego taktowania.

EFFICIENCY OF 8051 FAMILY MICROCONTROLLERS IN NETWORK APPLICATIONS

Summary. A protocol converter has been built using 8051 microcontroller family. It has been tested how converter's efficiecy depends upon microcontroller version and its clock frequency.

1. Wprowadzenie

Efektywność pracy sieci zawierającej segment przewodowy i bezprzewodowy zależy między innymi od prędkości transmisji na obu łączach oraz od rodzaju i parametrów stosowanych protokołów transmisyjnych. Nie są to jednak jedyne czynniki określające efektywność całej sieci. Często bowiem integracja segmentu przewodowego i bezprzewodowego wymaga zastosowania urządzeń mikroprocesorowych, tzw. konwerterów protokołów [1, 2]. Oczywiście parametry konstrukcyjne takiego konwertera mają również wpływ na jego wydajność – chodzi tu m. in. o takie parametry, jak rodzaj mikroprocesora, częstotliwość jego taktowania, ilość pamięci dostępnej dla buforowanych danych [3], a także sposób napisania oprogramowania sterującego.

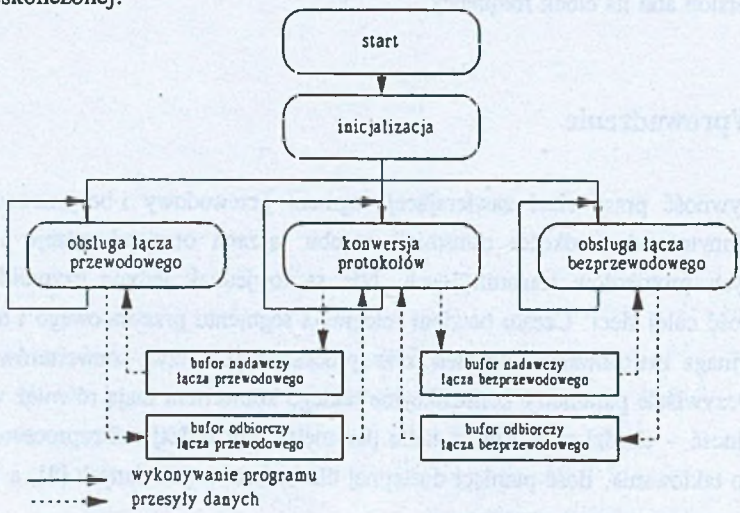
W niniejszej pracy zbadano wpływ częstotliwości taktowania mikroprocesora na wydajność transmisji danych na przykładzie modelu konwertera zbudowanego w oparciu o mikrosterownik jednoukładowy rodziny 8051 [4, 5, 6].

2. Konstrukcja modelu konwertera

Konwerter protokołów powinien być wyposażony w dwa porty szeregowe - jedno obsługujące łącze przewodowe, drugie - bezprzewodowe [1]. Dla uproszczenia konstrukcji w modelu zastosowano tylko jedno łącze. Wobec tego przesył danych przebiega w następujących etapach:

- odbiór znaku z linii wejściowej portu szeregowego i umieszczenie go w buforze odbiorczym,
- kopiowanie znaków z bufora odbiorczego do bufora podręcznego w celu ich obróbki, zmiany formatu itp.,
- kopiowanie danych z bufora podręcznego do bufora nadawczego,
- nadawanie znaków z bufora nadawczego przez linię wyjściową portu szeregowego.

Opisany tok postępowania jest w przybliżeniu zgodny z programem obsługi pełnego konwertera, pokazanym na rys. 1. Program taki można przedstawić jako trzy współbieżnie wykonywane procesy, z których dwa (procedury obsługi łączy transmisyjnych) wykorzystują mechanizm przerwań, trzeci natomiast (procedura konwersji protokołów) przebiega w pętli nieskończonej.



Rys. 1. Struktura programu konwertera protokołów
Fig. 1. The structure of protocol converter software

Bufory nadawczy i odbiorczy umieszczono w przestrzeni adresowej zewnętrznej pamięci danych, natomiast bufor podręczny – w wewnętrznej pamięci danych. W związku z tym rozmiar bufora podręcznego jest ograniczony do około 128 B, o ile zastosowana wersja mikrosterownika zawiera 256 B tej pamięci. Natomiast rozmiary buforów nadawczego i odbiorczego określone są przez rozmiar zastosowanego układu pamięci – w tym przypadku zastosowano pamięć o pojemności 8 KB, co pozwala na zastosowanie buforów nie przekraczających 2 KB. Zastosowanie bufora roboczego zostało podyktowane łatwością operowania na wewnętrznej pamięci danych, na której można wykonywać nie tylko operacje odczytu i zapisu, lecz także operacje logiczne i arytmetyczne. Natomiast pamięć zewnętrzna umożliwia tylko realizację operacji odczytu i zapisu.

Jako jednostkę centralną zastosowano układ 80C32 oraz – w celach porównawczych – układ Dallas 80C320 [6, 7] o zmodyfikowanej architekturze, pozwalającej na uzyskanie około trzykrotnie większej szybkości wykonywania programu. Układ ten zawiera ponadto dodatkowy port szeregowy, dzięki czemu można z łatwością zastosować go do realizacji pełnego konwertera protokołów.

3. Oprogramowanie modelu konwertera

Zadaniem programu sterującego układem jest [1]:

- odpowiednie ustawienie parametrów pracy układu, m. in. trybów pracy poszczególnych elementów mikrosterownika,
- obsługa portu szeregowego,
- realizacja algorytmu, stanowiącego model algorytmu konwersji protokołów.

Inicjalizacja parametrów konwertera obejmuje ustawienie:

- trybu pracy portu szeregowego,
- trybu pracy licznika T2, taktującego port szeregowy,
- wartości początkowej licznika T2, zależnej od przyjętej prędkości transmisji (stanowiącej parametr programu),
- trybu pracy licznika T1, służącego do odmierzania czasu,
- wartości początkowej licznika T1, zależnej od częstotliwości taktowania mikrosterownika (stanowiącej parametr programu),
- trybu pracy układu przerwań.

Procedura obsługi przerwania portu szeregowego rozpoznaje, czy powodem przerwania jest gotowość nadajnika, czy odbiornika.

Jeżeli przerwanie pochodzi od odbiornika, odebrany znak jest umieszczany w cyklicznym buforze odbiorczym. Jednocześnie zerowana jest zmienna, określająca, ile czasu upły-

nęło od nadejścia ostatniego znaku. Ponadto, dla celów badawczych, inkrementuje się zmienną określającą całkowitą liczbę odebranych znaków (liczba ta jest zerowana przy rozpoczęciu pracy programu).

Jeżeli przerwanie pochodzi od nadajnika, wykonywane jest sprawdzenie, czy w buforze nadawczym są przygotowane znaki do wysłania. Jeżeli tak, kolejny znak jest wpisywany do rejestru nadajnika. Ponadto, dla celów badawczych, inkrementuje się zmienną określającą całkowitą liczbę wysłanych znaków (liczba ta jest zerowana przy rozpoczęciu pracy programu).

Algorytm będący modelem algorytmu konwertera protokołów działa w pętli nieskończonej. Jeżeli w buforze odbiorczym znajduje się liczba znaków nie mniejsza niż pewna zadana wielkość stała (stanowiąca parametr programu), lub gdy od nadejścia ostatniego znaku upłynął zadany czas (także stanowiący parametr programu), następuje skopiowanie odpowiedniej liczby znaków z bufora odbiorczego do bufora roboczego. Następnie znaki te są kopiowane z bufora roboczego do bufora nadawczego, po czym następuje włączenie nadajnika. Podczas nadawania możliwe jest odbieranie kolejnych znaków.

W pełnym konwerterze między tymi dwoma etapami kopiowania znaków powinno nastąpić przekształcenie postaci przesyłanej informacji.

Prócz wymienionych wartości, sparametryzowane są także wielkości bufora roboczego oraz buforów nadawczego i odbiorczego.

Program napisano w języku C dla mikrosterowników rodziny 8051, korzystając z kompilatora IAR C-51 w wersji 5.21 i środowiska roboczego dla Windows w wersji 2.01. Do uruchamiania oprogramowania użyto także emulatora układowego USP-51A firmy Signum Systems. Emulator był także używany podczas badań, dzięki czemu zmiana parametrów programu nie pociągała za sobą konieczności zmiany zawartości pamięci stałej, gdyż program wykonywany był z pamięci emulatora.

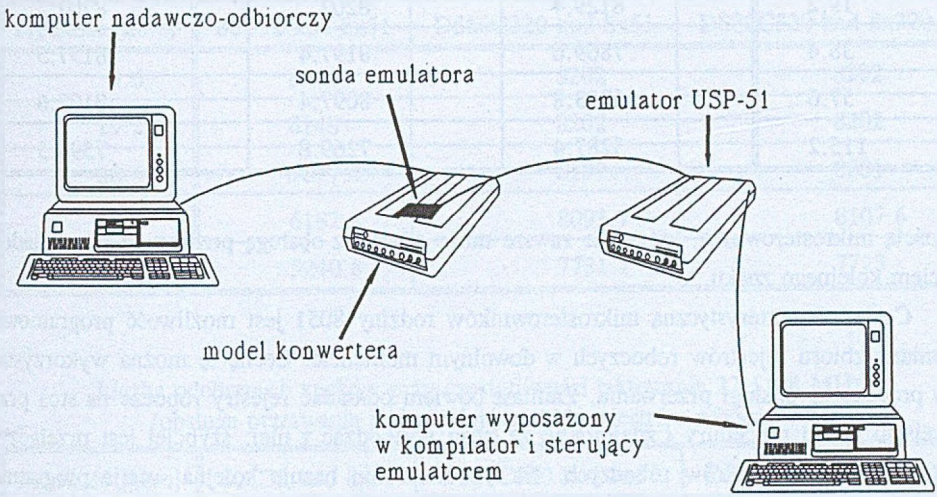
4. Badania wydajności układu

Wykonany model konwertera poddano badaniom wydajnościowym. W tym celu skonfigurowano stanowisko laboratoryjne, w skład którego wchodziły:

- komputer typu PC, służący do wysyłania znaków łączem szeregowym do konwertera i odbierania znaków z konwertera,
- układ konwertera, zawierający podstawkę pod mikroprocesor, 8 KB pamięci RAM oraz bufory portu szeregowego,
- emulator układowy, pełniący funkcję mikroprocesora w układzie konwertera,

– komputer typu PC, służący do zmian parametrów w programie źródłowym i sterowania emulatorem układowym oraz śledzenia wykonywania programu.

Do przesyłania znaków między komputerem a konwerterem wykorzystano program HyperTerminal, dostępny w systemie operacyjnym Windows 2000. Konfiguracja stanowiska badawczego pokazana jest na rys. 2.



Rys. 2. Konfiguracja stanowiska badawczego
Fig. 2. Test stand configuration

Pierwsze badania miały na celu określenie, z jaką maksymalną prędkością transmisji może pracować mikrosterownik rodziny 8051 przy transmisji ciągłej. Znaki przesyłano w seriach po 8 KB (dokładnie 8202 znaki) z następującymi prędkościami transmisji: 115.2, 57.6, 38.4, 19.2, 9.6 kb/s (są to prędkości dostępne w programie HyperTerminal). Częstotliwość zegara taktującego mikroprocesor wynosiła 11.0592 MHz – umożliwia ona uzyskanie dokładnej prędkości transmisji w żądanym zakresie. Dla każdej prędkości wykonano po 5 przesyłów (wyniki podane w tabeli stanowią średnią z wykonanych pomiarów). Testy wykonano na mikrosterownikach 80C32 i DS80C320 oraz dodatkowo DS80C320 z kodem generowanym specjalnie dla tego typu mikrosterownika. Tabela 1 zawiera średnią liczbę znaków, jaka została odebrana przez mikrosterownik i umieszczona w buforze przy określonej prędkości transmisji.

Jak wynika z danych zamieszczonych w tabeli 1, najwyższa prędkość transmisji możliwa do wykorzystania dla mikrosterownika 80C32 wynosi zaledwie 9.6 kb/s, natomiast dla wersji DS80C320 jest ona dwukrotnie wyższa. Program skompilowany specjalnie dla tej wersji jest tylko nieznacznie szybszy. Stosowanie prędkości wyższych niż podane jako maksymalne powoduje utratę części informacji. Efekt ten spowodowany jest zbyt małą pręd-

Tabela 1

Liczba odebranych znaków przy częstotliwości taktowania 11.0592 MHz
(obsługa przerwania korzystająca ze stosu)

Prędkość [kb/s]	80C32 kod 8x51	DS80C320 kod 8x51	DS80C320 kod 8x320
9.6	8202	8202	8202
19.2	8129.4	8202	8202
38.4	7869.6	8197.4	8197.5
57.6	6783.8	8097.4	8102.6
115.2	5287.4	7269.8	7596.5

kością mikrosterownika, który nie zawsze może zdążyć z obsługą przerwania przed nadejściem kolejnego znaku.

Cechą charakterystyczną mikrosterowników rodziny 8051 jest możliwość programowej zmiany zbioru rejestrów roboczych w dowolnym momencie. Cechę tę można wykorzystać w procedurze obsługi przerwania. Zamiast bowiem odkładać rejestry robocze na stos przy wejściu do tej procedury i zdejmować ze stosu wychodząc z niej, szybciej jest przełączyć aktywny zbiór rejestrów roboczych. Na tym założeniu bazuje kolejna wersja programu. Wyniki uzyskane z jej użyciem - uzyskane w identycznych warunkach jak poprzednie - zestawiono w tabeli 2.

Tabela 2

Liczba odebranych znaków przy częstotliwości taktowania 11.0592 MHz
(obsługa przerwania podmieniająca zbiór rejestrów roboczych)

Prędkość [kb/s]	80C32 kod 8x51	DS80C320 kod 8x51	DS80C320 kod 8x320
9.6	8202	8202	8202
19.2	8148	8202	8202
38.4	7877.4	8199	8201
57.6	6812.8	8093.7	8107.6
115.2	5240.8	7731.7	7723

Porównując wyniki umieszczone w tabelach, można stwierdzić, że technika zamiany rejestrów roboczych powoduje generalnie wzrost efektywności mikrosterownika, nie na tyle duży jednak, aby możliwe było osiągnięcie wyższych prędkości transmisji niż dla wersji korzystającej ze stosu. Pomimo to stanowi ona podstawę do dalszych badań.

Kolejne dwie grupy badań wykonano dla układu taktowanego zegarem 18.4320 MHz oraz 22.1184 MHz – te częstotliwości także pozwalają na uzyskanie dokładnej prędkości transmisji. Wyniki pomiarów zebrano w tabelach odpowiednio 3 i 4.

Tabela 3

Liczba odebranych znaków przy częstotliwości taktowania 18.4320 MHz (obsługa przerwania podmieniająca zbiór rejestrów roboczych)

Prędkość [kb/s]	80C32 kod 8x51	DS80C320 kod 8x51	DS80C320 kod 8x320
9.6	8202	8202	8202
19.2	8148	8202	8202
38.4	7877.4	8199	8201
57.6	6182.8	8093.7	8107.6
115.2	5240.8	7731.7	7723

Tabela 4

Liczba odebranych znaków przy częstotliwości taktowania 22.1184 MHz (obsługa przerwania podmieniająca zbiór rejestrów roboczych)

Prędkość [kb/s]	80C32 kod 8x51	DS80C320 kod 8x51	DS80C320 kod 8x320
9.6	8202	8202	8202
19.2	8202	8202	8202
38.4	8166.3	8202	8202
57.6	8059	8202	8202
115.2	5482	8088.3	8089

Jak wynika z zamieszczonych wyników, zwiększanie częstotliwości zegara taktującego powoduje – zgodnie z przypuszczeniami – wzrost mocy obliczeniowej mikrosterownika. Dwukrotne zwiększenie tej częstotliwości umożliwia około dwukrotne zwiększenie maksymalnej prędkości transmisji. Niestety, nie jest możliwe ustalenie dokładnych zależności, ponieważ użyty program HyperTerminal oferuje stosunkowo wąski wybór prędkości transmisji. Ponadto często trudno jest uzyskać dokładnie taką samą prędkość transmisji w komputerze PC i w układzie z mikrosterownikiem 8051 – wynika to z różnic w konstrukcji portu szeregowego.

Interesujący wynik uzyskano dla prędkości 57.6 kb/s dla mikrosterownika DS80C320 (tabela 3 i 4). Otóż zegar 22.1184 MHz jest wówczas wystarczający dla uzyskania tej prędkości, natomiast zegar 18.432 Hz – tylko nieznacznie wolniejszy – okazuje się zbyt wolny.

Układ gubi jednak wówczas tylko kilka znaków z ponad 8000 wysłanych. Dlatego też konieczna była dokładna analiza kodu wynikowego generowanego przez kompilator. Okazało się, że inkrementacja zmiennej typu *long unsigned* jest wykonywana z użyciem procedury dodawania 32-bitowego. Po zmianie typu tej zmiennej na *unsigned* kod nie zawierał już takiego wywołania. W tej wersji zegar 18.432 MHz okazał się już wystarczający dla prędkości 57.6 kb/s. Zmienna, której typ zmieniono, jest zmienną diagnostyczną, dzięki której można określić liczbę wysłanych i odebranych znaków, tak więc nie jest niezbędna w ostatecznej wersji programu.

5. Kierunki dalszych badań

Uzyskane wyniki mogą stanowić podstawę do dalszych badań wydajności mikrosterowników rodziny 8051 w układach transmisji danych. W szczególności można zbadać:

- uzyskiwane prędkości transmisji dla innych częstotliwości zegara taktującego mikrosterownik,
- wpływ sposobu optymalizacji kodu wynikowego przez kompilator IAR C-51 na wydajność systemu,
- wpływ złożoności użytego protokołu transmisyjnego i jego parametrów na efektywność systemu.

Można także podjąć próbę napisania programu obsługi przerwania portu szeregowego w asemblerze - porównanie wyników dla programu napisanego w języku C i wasemblerze może stanowić interesujący test wydajności zastosowanego kompilatora.

LITERATURA

1. Zieliński B.: Bezprzewodowe sieci komputerowe wykorzystujące konwersję protokołów. Rozprawa doktorska, Instytut Informatyki Politechniki Śląskiej, Gliwice 1997.
2. Zieliński B.: Bezprzewodowe sieci komputerowe. Helion, Gliwice 2000.
3. Zieliński B., Tokarz K.: Wpływ mocy obliczeniowej konwertera protokołów na wydajność transmisji. Studia Informatica, Vol. 21, Number 1 (39).
4. Rydzewski A.: Mikrokomputery jednoukładowe rodziny MCS-51. WNT, Warszawa 1992.
5. Małysiak H.: Mikrokomputery jednoukładowe serii MCS48, MCS51, MCS96. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 1992.
6. Starecki T.: Mikrokontrolery jednoukładowe rodziny 51. NOZOMI, Warszawa 1996.

7. DS80C320/DS80C323 High Speed/Low Power Micro. Dallas Semiconductor, 1996. http://pdfserv.maxim-ic.com/arpdf/DS80C320-DS80C323.pdf.

Recenzent: Dr hab. inż. Edward Hrynkiewicz, prof. Pol. Śląskiej

Wpłynęło do Redakcji 18 marca 2002 r.

Abstract

The network's efficiency depends upon several parameters, ie. transmission speed and protocols used. If the network contains microprocessor-based elements – for example protocol converters – their efficiency may also have influence upon overall network characteristic.

The protocol converter has, among others, two transmission links (wired and wireless ones). Proper cooperation between all the elements is ensured by the software, which may have similar structure to the one shown on fig. 1.

The microprocessor chosen for the tests was standard 80C32 single-chip microcontroller and Dallas DS80C320 version. The control program was written in C language for 8051 family of microcontrollers usign IAR C-51 environment.

For the tests, the stand has been configured. It consisted of two PC computers, protocol converter and in-circuit emulator as shown on fig. 2. The transmission between PC and protocol converter was controlled by HyperTerminal available in Windows 2000. The transmission speeds were: 115.2, 57.6, 38.4, 19.2, 9.6 kb/s and the clock frequencies were: 11.0592 MHz, 18.4320 MHz and 22.1184 MHz. The efficiency measure was number of characters received, when the number of characters sent was constant and equal to 8202. Therefore, if these numbers are not equal, the converter's efficiency is not sufficient for the transmission speed used. The test results are collected in tables 1 to 4.

As an overall result, it can be said that the higher the clock frequency is, the faster transmission speed can be used. The DS80C320 version can always work at speeds twice as high as standard 80C32.