

Michał SAWICKI
Politechnika Śląska, Instytut Informatyki

ANALIZA ASYNCHRONICZNEGO TRANSFERU DANYCH W MODELACH SYSTEMÓW KOMUNIKACYJNYCH USB I FIREWIRE

Streszczenie. W niniejszym artykule przeprowadzono analizę transferu asynchronicznego w zaproponowanych modelach systemów komunikacyjnych USB i FireWire. W ramach analizy przybliżono szczegóły komunikacji pomiędzy komputerem a pamięcią masową oraz wyznaczono czasy realizacji transferu danych w analizowanych systemach komunikacyjnych. Oszacowane wartości czasów realizacji transferu danych umożliwiły wskazanie interfejsu komunikacyjnego dla pamięci masowej.

Słowa kluczowe: interfejs szeregowy, pamięć masowa, transfer danych

ANALYSIS OF ASYNCHRONOUS DATA TRANSFER IN COMMUNICATION SYSTEM MODELS FOR USB AND FIREWIRE

Summary. This paper presents an analysis of asynchronous data transfer in communication system models for USB and FireWire. The analysis shows the details of communication between the computer and mass storage and time of data transfer is estimated in these communication systems. Estimated value of execution time of data transfer allowed to specify a communication interface for mass storage.

Keywords: serial interface, mass storage, data transfer

1. Wprowadzenie

Jednym z głównych problemów, na jaki może natknąć się projektant systemu komputerowego, jest wybór odpowiedniego systemu komunikacyjnego. Przy wyborze systemu komunikacyjnego należy wziąć pod uwagę wiele aspektów, np. ilość wymienianych danych, liczbę

urządzeń komunikujących się ze sobą, przyszłą rozbudowę systemu komputerowego, programowy dostęp do portu.

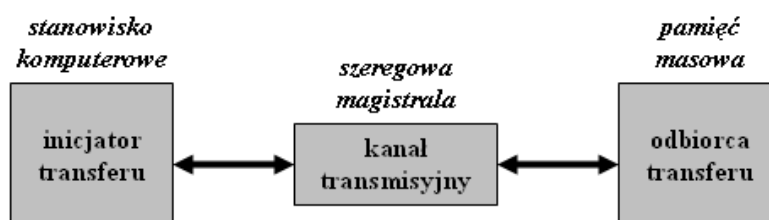
W tym artykule zostaną porównane porty USB i FireWire ze względu na czas realizacji transferu danych (zleconego przesyłu danych) w systemie komunikacyjnym, składającym się tylko z dwóch węzłów. Taka konfiguracja systemu komunikacyjnego jest często spotykana w systemach komputerowych domowego użytku. Jednym z węzłów jest najczęściej stanowisko komputerowe, a drugim zewnętrzna pamięć masowa pod postacią pamięci przenośnej (pendrive) lub zewnętrznego dysku twardego.

W celu porównania portów USB i FireWire zdefiniowano model idealnego systemu komunikacyjnego oraz oparte na nim „idealizowane” modele systemów USB i FireWire. Wykorzystując wymienione modele, wyznaczono analitycznie wartości czasu realizacji transferu danych. Pozwoliło to dokonać porównania portów USB i FireWire.

2. Model idealnego systemu komunikacyjnego

Model systemu komunikacyjnego (rys.1) składa się z trzech podstawowych elementów: inicjatora transferu danych, kanału komunikacyjnego i odbiorcy transferu. Inicjator transferu rozpoczyna transfer danych, wysyłając żądanie przesyłu danych, informujące odbiorcę o rozpoczęciu komunikacji (wymiany danych). Odbiorca transferu realizuje zlecone żądanie, zapisując odebrane dane lub odsyłając odczytane dane. Każda wymiana danych pomiędzy inicjatorem a odbiorcą jest realizowana za pośrednictwem kanału transmisyjnego.

W systemie komputerowym przeważnie występuje wiele aplikacji zlecających przesłanie danych za pomocą jednego interfejsu, dlatego w celu zapewnienia podziału pasma pomiędzy różnymi aplikacjami transfery danych są dzielone na mniejsze jednostki: transakcje.



Rys. 1. Model systemu komunikacyjnego
Fig. 1. Communication system model

W idealnym systemie komunikacyjnym zakłada się brak opóźnienia spowodowanego czasem propagacji sygnału w kanale, natychmiastową reakcją odbiorcy oraz bezbłędną pracą kanału.

Czas propagacji sygnału w idealnym kanale jest zerowy, co oznacza, że w tym samym momencie czasu dane wysyłane przez nadawcę są odbierane przez odbiorcę. Natychmiasto-

wa reakcja odbiorcy polega na tym, że bezzwłocznie po ostatnim bicie odebranego żądania jest wysyłany pierwszy bit odpowiedzi (potwierdzenia). Pozbawiona błędów praca kanału transmisyjnego polega na tym, że przesyłany sygnał nie ulega zmianie na skutek zewnętrznych zakłóceń, tzn. zdarzeniem pewnym jest odebranie bitu o wartości logicznej takiej, jaka została ustawiona przez nadawcę.

Huby i rozgałęźniki magistrali w idealnym systemie komunikacyjnym nie wprowadzają żadnego opóźnienia w transmisji pomiędzy inicjatorem a odbiorcą. Z tego względu nie są one uwzględniane przy obliczaniu czasu trwania transferów.

W kolejnych dwóch podpunktach zostaną zaprezentowane porty USB i FireWire w ujęciu idealnego systemu komunikacyjnego, w którym jest wykonywana asynchroniczna (masowa) operacja odczytu danych. Podobna sytuacja ma miejsce przy pobieraniu danych z zewnętrznego dysku twardego do komputera.

2.1. Model systemu komunikacyjnego USB

W modelu systemu komunikacyjnego USB inicjatorem transferu jest host (najczęściej rolę hosta pełni stanowisko komputerowe), a odbiorcą urządzenie podłączone do magistrali USB (zewnętrzna pamięć masowa). Kanałem transmisyjnym jest szeregową magistrala USB.

Każde urządzenie USB posiada zbiór konfiguracji, które z kolei posiadają zbiory interfejsów. Interfejs w tym przypadku jest rozumiany jako zbiór 16 punktów końcowych (buforów), z którymi może odbywać się wymiana danych. Kierunek operacji na punkcie końcowym może być tylko jeden: albo zapis albo odczyt (wyjątkiem jest zerowy punkt końcowy).

Inicjator transferu (host) rozpoczyna przesył danych wysłaniem do odbiorcy pakietu tokena (tabela 1). Pakiet tokena dla transakcji masowego odczytu zawiera PID, pole adresowe i 5-bitową sumę kontrolną CRC. Zawartość pola PID determinuje rodzaj pakietu. Przykładowo, pole PID o wartości 0x69 wskazuje, że host zainicjował transakcję odczytu, wysyłając pakiet tokena IN. Na pole adresowe składają się adres urządzenia i numer punktu końcowego. Rozmiar pola adresowego pozwala na zaadresowanie maksymalnie 127 urządzeń na magistrali (7 bitów), z czego w każdym urządzeniu może być zdefiniowanych maksymalnie 16 punktów końcowych (4 bity). Punkt końcowy o zerowym numerze służy do konfiguracji urządzenia.

Tabela 1

Format pakietu tokena IN

| Nazwa pola | Liczba bitów |
|--------------------------------|--------------|
| PID | 8 |
| Adres węzła | 7 |
| Numer punktu końcowego | 4 |
| Suma kontrolna CRC | 5 |
| Sumaryczna liczba bitów | 24 |

Odbiorca po odebraniu pakietu tokena IN odsyła bezzwłocznie pakiet danych (tabela 2), zawierający żądane dane. Przesłany pakiet danych składa się z PID, pola danych (bloku danych) i 16-bitowej sumy kontrolnej CRC. Nie występuje w nim pole adresowe, ponieważ przy operacji odczytu pakietu danych zawsze są kierowane do węzła będącego hostem, a na magistrali USB może występować tylko jeden taki węzeł.

Tabela 2

| Format pakietu danych | |
|--------------------------------|-------------------------------|
| Nazwa pola | Liczba bitów |
| PID | 8 |
| Pole danych | zależna od prędkości |
| Suma kontrolna CRC | 16 |
| Sumaryczna liczba bitów | 24+rozmiar pola danych |

Rozmiar bloku danych (MaxPacketSize) zależy od prędkości transmisji, z jaką pracuje masowy punkt końcowy (tabela 3). Przykładowo, jeżeli operacja odczytu na magistrali USB 2.0 jest realizowana z maksymalną możliwą szybkością (480 Mb/s), to w ramach jednego pakietu danych można przesłać maksymalnie 512 bajtów danych.

Tabela 3

| Wartości parametru MaxPacketSize | |
|----------------------------------|-------------------|
| Prędkość transmisji | MaxPacketSize |
| 1,5 Mb/s (LowSpeed) | niedozwolony tryb |
| 12 Mb/s (FullSpeed) | 64 bajtów |
| 480 Mb/s (HighSpeed) | 512 bajtów |

Po odebraniu pakietu danych przez inicjatora transferu (hosta) jest odsyłany jednobajtowy pakiet potwierdzenia (tabela 4). Ze względu na idealny charakter kanału transmisyjnego, wysłane potwierdzenie zawsze będzie informowało odbiorcę (pamięć masową) o poprawnym zakończeniu operacji odczytu, co oznacza, że nigdy nie wystąpią retransmisje uszkodzonych danych. Dlatego w przyjętym modelu masowa transakcja odczytu będzie zrealizowana tylko w ramach trzech pakietów.

Tabela 4

| Format pakietu potwierdzenia | |
|--------------------------------|--------------|
| Nazwa pola | Liczba bitów |
| PID | 8 |
| Sumaryczna liczba bitów | 8 |

Każdy pakiet transmitowany na magistrali USB musi być poprzedzony 32-bitową sekwencją synchronizacyjną, zapewniającą synchronizację na poziomie bitów w odbiorniku. Dodatkowo, każdy pakiet zakończony jest 2-bitowym znacznikiem końca pakietu (EOP).

Założono, że w modelu systemu USB komunikacja odbywa się z wysoką szybkością i nie występują żadne inne transfery, oprócz masowego. W rzeczywistym systemie komunikacyjnym USB, wyposażonym w zewnętrzną pamięć masową, oprócz transferu danych odbywa się

również transfer komend SCSI (Small Computer System Interface) oraz transfer statusów. Pamięć masowa udostępnia cztery punkty końcowe: kontrolny, przerwaniowy i dwa masowe. Kontrolny punkt końcowy służy do sterownia pamięcią masową (transfer komend). Masowe punkty końcowe muszą być jednokierunkowe, dlatego w celu zapewnienia dwukierunkowej wymiany danych z pamięcią masową są udostępniane dwa masowe punkty końcowe: jeden do odczytu, drugi do zapisu. Przerwaniowy punkt końcowy służy do informowania sterownika pamięci masowej o statusie zakończonej komendy sterującej.

Na początku każdej mikroramki host generuje pakiet początku mikroramki (SOF), który składa się z 11-bitowego numeru 1 ms ramki oraz z 5-bitowej sumy kontrolnej CRC. Pakiet SOF wnosi do transmisji danych narzut informacji nadmiarowych w wymiarze 50 bitów, występujący tylko raz dla każdej mikroramki.

Jeżeli inicjator (stanowisko komputerowe) zleci transfer danych o wielkości przekraczającej wartość parametru MaxPacketSize, to zlecony transfer jest dzielony na mniejsze jednostki: transakcje. W ramach masowej transakcji są przesyłane za każdym razem pakiety tokena, danych i potwierdzenia. Na rys. 2 została zaprezentowana mikroramka, w ramach której są wykonywane transakcje masowe.



Rys. 2. Mikroramka w systemie komunikacyjnym USB
 Fig. 2. The micro-frame in the USB communication system

Dana jest liczba D , będąca liczbą bajtów do odczytania z zewnętrznej pamięci masowej. Suma wszystkich informacji nadmiarowych wchodzących w skład jednej masowej transakcji odczytu wynosi:

$$l_{nad}^{USB} = \left(\underbrace{32}_{\text{sekwencja synchronizacyjna}} + \underbrace{24}_{\text{polopakietu}} + \underbrace{2}_{\text{znacznik EOP}} \right) + \underbrace{(32 + 24 + 2)}_{\text{pakiet danych}} + \underbrace{(32 + 8 + 2)}_{\text{pakiet potwierdzenia}} = 158 \quad (1)$$

Czas trwania jednej masowej transakcji odczytu bloku danych (z maksymalnym rozmiarem pola danych) przy założonych warunkach jest wyrażony wzorem (2).

$$t_{Tr} = \frac{l_{nad}^{USB} + \overbrace{4096}^{\text{rozmiar poladanych}}}{480} = 8,86[\mu s] \quad (2)$$

Czas transmisji samych nadmiarowych informacji jednej masowej transakcji odczytu wynosi:

$$t_{Tr}' = \frac{l_{nad}^{USB}}{480} = 0,33[\mu s] \quad (3)$$

Czas transmisji danych na magistrali USB został wyznaczony na podstawie następującego wzoru:

$$t_D^{USB} = \frac{8D}{480} \quad (4)$$

Czas transmisji pakietu początku mikroramki (SOF) wyraża się wzorem:

$$t_{SOF} = \frac{\left(\overbrace{32}^{\text{sekwencja synchronizacyjna}} + \overbrace{16}^{\text{połopakietu SOF}} + \overbrace{2}^{\text{znacznik EOP}} \right)}{480} = 0,10[\mu s] \quad (5)$$

W systemie komunikacyjnym USB jest wymagane, aby transakcja realizowana w ramach mikroramki zakończyła się przed końcem mikroramki. Z tego względu w mikroramkach może pojawić się wolne pasmo, co prowadzi do nieefektywnego wykorzystania czasu w mikroramce. Liczba wykonanych masowych transakcji odczytu (maksymalny rozmiar pola danych) w ramach jednej mikroramki wynosi:

$$m_{Tr} = \left\lfloor \frac{\overbrace{125}^{\text{czas trwania mikroramki}} - t_{SOF}}{t_{Tr}} \right\rfloor = 14 \quad (6)$$

Czas w ramach jednej mikroramki, przez który nie ma aktywności (niewykorzystane pasmo) na magistrali USB przy zleconym transferze odczytu wynosi:

$$t_{ba} = 125 - \underbrace{t_{SOF}}_{\text{pasmo SOF}} - \overbrace{m_{Tr} t_{Tr}}^{\text{pasmo transakcji masowych}} = 0,82[\mu s] \quad (7)$$

Zlecony transfer danych (operacja odczytu) zostanie podzielony na transakcje. Liczba potrzebnych transakcji do wykonania transferu wynosi:

$$n_{Tr} = \left\lceil \frac{D}{512} \right\rceil \quad (8)$$

Za pełne mikroramki uważa się mikroramki, w ramach których wykonano maksymalną możliwą liczbę transakcji masowego odczytu danych. Liczba pełnych mikroramek, w ramach których będzie zrealizowany zlecony transfer, wyraża się wzorem (9).

$$Tr_m = \left\lfloor \frac{n_{Tr}}{m_{Tr}} \right\rfloor \quad (9)$$

Powyższy wzór uwzględnia tylko mikroramki, w których całe możliwe pasmo zajęte jest przez transakcje zleconego transferu. W systemie USB jest wymagane, aby wszystkie pakiety danych oprócz ostatniego zawierały bloki danych o maksymalnym rozmiarze. Wzór (9) nie uwzględnia tego warunku, ponieważ ostatni pakiet danych transferu jest traktowany we wzorze (9) jako pakiet z blokiem danych o maksymalnym rozmiarze (MaxPacketSize).

Przez niepełną mikroramkę rozumie się mikroramkę, w ramach której wolne pasmo pozwala na realizację co najmniej jednej transakcji masowego odczytu. Wzór (10) wyraża liczbę

bę mikroramek potrzebnych do realizacji zleconego transferu (łącznie liczba pełnych i ostatniej ewentualnie niepełnej mikroramki).

$$Tr_m' = \left\lceil \frac{n_{Tr}}{m_{Tr}} \right\rceil \quad (10)$$

Na podstawie wzorów (3)-(5) i (7)-(10) można wyznaczyć całkowity czas realizacji zleconego transferu (operacji odczytu) w założonym modelu systemu komunikacyjnego USB. Czas ten jest wyrażony wzorem:

$$t_{\min}^{USB} = \overbrace{t_{SOE} Tr_m'}^{\text{łączny czas transmisji SOF}} + \underbrace{t_{ba} Tr_m}_{\text{łączne niewykorzystane pasmo}} + \overbrace{n_{Tr} t_{Tr}' + t_D^{USB}}^{\text{czas realizacji wszystkich transakcji}} \quad (11)$$

Podstawiając wcześniej wyliczone wartości i wyznaczone zależności, można określić ostateczną postać wzoru na czas realizacji zleconego transferu:

$$t_{\min}^{USB} = 0,1 \left\lceil \frac{\left\lceil \frac{D}{512} \right\rceil}{14} \right\rceil + 0,82 \left\lceil \frac{\left\lceil \frac{D}{512} \right\rceil}{14} \right\rceil + 0,33 \left\lceil \frac{D}{512} \right\rceil + \frac{8D}{480} \quad (12)$$

Jedynym parametrem wzoru (12) jest liczba bajtów danych do odczytania z zewnętrznej pamięci masowej. Wartość wyznaczona na podstawie wzoru (12) wyraża czas realizacji zleconego transferu w mikrosekundach.

Powyższe wzory nie uwzględniają bitów wstawianych (synchronizujących), dlatego wyprowadzone zależności opisują najlepszy przypadek transmisji danych w USB, tzn. sytuację gdy transmitowanych jest z rzędu co najwyżej pięć jedynek logicznych. Wzory (11)-(12) określają kres dolny (infimum) zbioru czasów realizacji transferu danych.

Dodatkowy bit wstawiany o wartości zero jest umieszczany po sześciu kolejnych jedynek logicznych transmitowanych w USB, dlatego najgorszym przypadkiem transmisji danych jest transmisja samych jedynek logicznych. Kolejne wyprowadzane zależności odnoszą się do najgorszego przypadku.

Liczba wstawionych bitów synchronizujących w ramach jednej transakcji masowej z maksymalnym rozmiarem pola danych wynosi:

$$l_{Tr}^w = \overbrace{\left\lceil \frac{24}{6} \right\rceil}^{\text{pakiet tołena}} + \underbrace{\left\lceil \frac{4120}{6} \right\rceil}_{\text{pakiet danych}} + \overbrace{\left\lceil \frac{8}{6} \right\rceil}^{\text{pakiet potwierdzenia}} = 691 \quad (13)$$

Czas trwania jednej masowej transakcji odczytu bloku danych (z maksymalnym rozmiarem pola danych) z uwzględnieniem bitów wstawianych wynosi:

$$t_{Tr}^w = \frac{l_{nad}^{USB} + \overbrace{4096}^{\text{pole danych}} + l_{Tr}^w}{480} = 10,30[\mu s] \quad (14)$$

Czas transmisji samych bitów synchronizujących, wstawianych w ramach jednej masowej transakcji, wynosi:

$$t_{bs}^w = \frac{l_{Tr}^w}{480} = 1,44[\mu s] \quad (15)$$

Podczas transmisji pakietu SOF są wstawiane dodatkowo dwa bity synchronizujące, dlatego czas transmisji pakietu SOF uwzględniający bity wstawiane wynosi:

$$t_{SOF}^w = \frac{\overbrace{50}^{\text{pakiet SOF}} + \overbrace{2}^{\text{bity wstawiane}}}{480} = 0,11[\mu s] \quad (16)$$

Liczba wykonanych masowych transakcji odczytu w ramach jednej mikroramki wynosi:

$$m_{Tr}^w = \left\lfloor \frac{125 - t_{SOF}^w}{t_{Tr}^w} \right\rfloor = 12 \quad (17)$$

Porównując wzory (6) i (17), można zauważyć, że wstawianie dodatkowych bitów synchronizujących zmniejsza liczbę wykonanych transakcji w ramach jednej mikroramki z 14 do 12 transakcji masowych.

Niewykorzystane pasmo w jednej mikroramce wynosi:

$$t_{ba}^w = 125 - \underbrace{t_{SOF}^w}_{\text{pasmo SOF}} - \overbrace{m_{Tr}^w t_{Tr}^w}_{\text{pasmo transakcji masowych}} = 1,27[\mu s] \quad (18)$$

Liczba pełnych mikroramek potrzebnych do zrealizowania zleconego transferu (z uwzględnieniem bitów wstawianych) wynosi:

$$Tr_m^w = \left\lceil \frac{n_{Tr}}{m_{Tr}^w} \right\rceil \quad (19)$$

Łączna liczba pełnych i ostatniej ewentualnie niepełnej mikroramki wyraża się wzorem:

$$Tr_m^{w'} = \left\lceil \frac{n_{Tr}}{m_{Tr}^w} \right\rceil \quad (20)$$

Na podstawie wzorów (3), (4), (8), (15), (16), (18) – (20) można wyznaczyć całkowity czas realizacji zleconego transferu z uwzględnieniem bitów wstawianych. Czas ten wynosi:

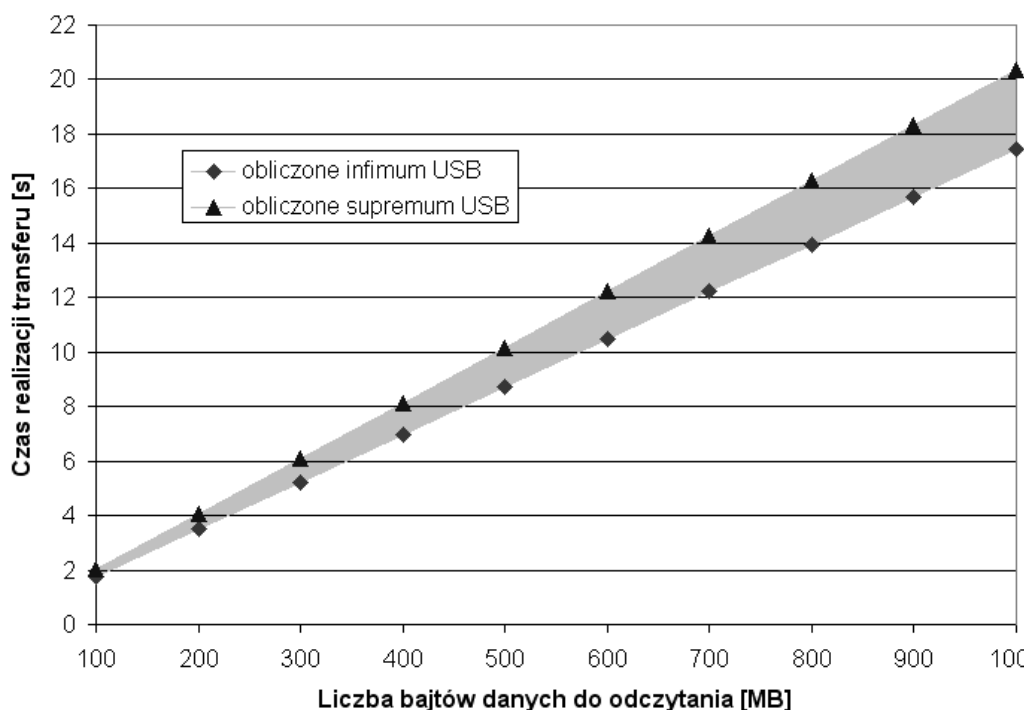
$$t_{max}^{USB} = \underbrace{t_{SOF}^w Tr_m^w}_{\text{łączny czas transmisji SOF}} + \underbrace{t_{ba}^w Tr_m^w}_{\text{łączne niewykorzystane pasmo}} + \overbrace{n_{Tr} (t_{Tr}^w + t_{bs}^w)}^{\text{czas realizacji wszystkich transakcji}} + t_D^{USB} \quad (21)$$

Podstawiając wcześniej wyliczone wartości i wyznaczone zależności, można wyznaczyć ostateczną postać wzoru na czas realizacji zleconego transferu w najgorszym przypadku transmisji na magistrali USB:

$$t_{\max}^{USB} = 0,11 \left\lceil \left\lceil \frac{D}{512} \right\rceil \right\rceil + 1,27 \left\lceil \frac{D}{512} \right\rceil + 1,77 \left\lceil \frac{D}{512} \right\rceil + \frac{8D}{480} \quad (22)$$

Jedynym parametrem wzoru (22) jest liczba bajtów danych do odczytania z zewnętrznej pamięci masowej. Wartość wyznaczona na podstawie wzoru (22) wyraża czas realizacji zleconego transferu w mikrosekundach. Wzory (21) i (22) określają kres górny (supremum) zbioru czasów realizacji transferu danych.

Przykładowo, dla przyjętych założeń operacja odczytu 1 GB danych z zewnętrznej pamięci masowej w najlepszym przypadku zajęłaby 17 s, a w najgorszym przypadku 20 s i byłaby zrealizowana w ramach dwóch milionów transakcji (dokładnie w ramach 1953125 transakcji). Na rys. 3 zostały zaprezentowane supremum i infimum zbioru czasów realizacji transferu danych dla różnych rozmiarów odczytywanego pliku (porcji danych). Czasy realizacji transferu danych o dowolnej wartości będą znajdowały się wewnątrz obszaru wyznaczonego przez kresy górny i dolny. Obszar ten został zacieniowany na rys. 3.



Rys. 3. Wykres zależności czasu realizacji operacji masowego odczytu od wielkości porcji danych w modelu systemu komunikacyjnego USB

Fig. 3. Graph of execution time of data transfer in the USB communication system model

2.1.1. Model systemu komunikacyjnego FireWire

W modelu systemu komunikacyjnego FireWire rolę inicjatora transferu danych może pełnić dowolny węzeł na magistrali, który chce wykonać operację odczytu z pamięci masowej, będącej odbiorcą transferu. Najczęściej jednak inicjatorem transferu jest stanowisko komputerowe wyposażone w kontroler FireWire. Kanalem transmisyjnym jest szeregowy magistrala FireWire.

Transakcja asynchroniczna w interfejsie FireWire składa się z dwóch faz. W pierwszej fazie są przesyłane pakiety żądania i potwierdzenia żądania. W drugiej fazie są przesyłane pakiety odpowiedzi i potwierdzenia odpowiedzi. Pakiety potwierdzeń żądania i odpowiedzi informują nadawcę odpowiednio żądania i odpowiedzi o poprawnym zakończeniu pierwszej lub drugiej fazy.

Inicjator transakcji rozpoczyna operację odczytu wysyłając do odbiorcy asynchroniczny pakiet żądania odczytu bloku danych (tabela 5). Pakiet ten zawiera adresy inicjatora i odbiorcy, a także m.in. kod transakcji, etykietę transakcji, adres w przestrzeni wymiany danych pamięci masowej oraz długość bloku danych. Nagłówek pakietu jest zabezpieczony 32-bitową sumą kontrolną CRC. Pole z kodem transakcji determinuje rodzaj zainicjowanej transakcji asynchronicznej. Przykładowo, pole kodu transakcji o wartości 0b0101 wskazuje, że stanowisko komputerowe żąda asynchronicznego odczytu bloku danych z pamięci masowej. W systemie FireWire jest możliwe realizowanie z jednym węzłem wielu transakcji asynchronicznych jednocześnie, dlatego pakiety poszczególnych transakcji są rozróżniane na podstawie wartości pola etykiety transakcji.

Tabela 5

Format pakietu żądania odczytu bloku danych

| Nazwa pola | Liczba bitów |
|--------------------------------|--------------|
| Adres docelowy | 16 |
| Adres źródłowy | 16 |
| Etykieta transakcji | 6 |
| Kod repetycji | 2 |
| Kod transakcji | 4 |
| Priorytet | 4 |
| Adres w węźle docelowym | 48 |
| Długość bloku danych | 16 |
| Rozszerzony kod transakcji | 16 |
| Suma kontrolna CRC nagłówka | 32 |
| Sumaryczna liczba bitów | 160 |

Odbiorca transakcji po odebraniu ostatniego bitu pakietu żądania odczytu rozpoczyna transmisję jednobajtowego pakietu potwierdzenia (tabela 6). Odebranie pakietu potwierdzenia przez inicjatora kończy pierwszą fazę asynchronicznej transakcji odczytu.

Tabela 6

Format pakietu potwierdzenia

| Nazwa pola | Liczba bitów |
|--------------------------------|--------------|
| Kod potwierdzenia | 4 |
| Parzystość potwierdzenia | 4 |
| Sumaryczna liczba bitów | 8 |

Po wysłaniu pakietu potwierdzenia odbiorca odsyła również asynchroniczny pakiet odpowiedzi (tabela 7) na żądanie odczytu bloku danych, zawierający żądane dane. Pakiet odpowiedzi zawiera adresy inicjatora i odbiorcy, ponieważ na magistrali może występować wiele węzłów mogących żądać odczytu danych z pamięci masowej. Analogicznie jak w przypadku pakietu żądania, pakiet danych zawiera m.in. pola kodu transakcji, etykiety transakcji i długości bloku danych, a w przypadku operacji odczytu również pole danych.

Tabela 7

Format pakietu odpowiedzi

| Nazwa pola | Liczba bitów |
|---------------------------------|---------------------------------|
| Adres docelowy | 16 |
| Adres źródłowy | 16 |
| Etykieta transakcji | 6 |
| Kod repetycji | 2 |
| Kod transakcji | 4 |
| Priorytet | 4 |
| Kod odpowiedzi | 4 |
| Zarezerwowane | 44 |
| Długość bloku danych | 16 |
| Suma kontrolna CRC nagłówka | 32 |
| Blok danych | uzależniona od prędkości |
| Suma kontrolna CRC bloku danych | 32 |
| Sumaryczna liczba bitów | 176+rozmiar bloku danych |

Maksymalny rozmiar pola danych (tabela 8) w pakiecie odpowiedzi jest zdeterminowany przez prędkość, z jaką komunikuje się inicjator i odbiorca. Przykładowo, dla transmisji S400 maksymalny rozmiar bloku danych w pakiecie odpowiedzi może wynieść 2048 bajtów.

Tabela 8

Rozmiary pola danych w asynchronicznym pakiecie odpowiedzi na żądanie odczytu

| Prędkość transmisji | Maksymalny rozmiar pola danych |
|---------------------|--------------------------------|
| S100 | 512 bajtów |
| S200 | 1024 bajty |
| S400 | 2048 bajtów |

Po odebraniu pakietu odpowiedzi przez inicjatora transferu transmitowany jest po raz drugi pakiet potwierdzenia (tabela 8), przy czym nadawcą w tym przypadku jest inicjator, a adresatem odbiorca transferu. Wraz z odebraniem ostatniego bitu potwierdzenia przez od-

biorec kończy się druga i ostatnia faza asynchronicznej transakcji odczytu bloku danych. W założonym modelu systemu komunikacyjnego FireWire nigdy nie występują retransmisje uszkodzonych danych, dlatego w przyjętym modelu asynchroniczna transakcja odczytu bloku danych będzie zrealizowana w ramach tylko czterech asynchronicznych pakietów.

Każdy pakiet transmitowany na magistrali FireWire musi być zakończony sekwencją synchronizującą (tzw. spływające bity), zapewniającą poprawny odbiór ostatnich bitów w pakiecie. Dodatkowo, do każdego pakietu są dodawane sprzętowo znaczniki początku i końca pakietu. Znaczniki te nie są uwzględnione w dalszych obliczeniach.

Założono, że w przyjętym modelu systemu FireWire komunikacja odbywa się z szybkością S400 i nie występują żadne inne transfery danych oprócz asynchronicznego. W rzeczywistym systemie komunikacyjnym FireWire, wyposażonym w zewnętrzną pamięć masową, transfer danych odbywa się za pośrednictwem protokołu transportowego SBP-2 (Serial Bus Protocol 2). Protokół SBP-2 jest odpowiedzialny za transmisję komend sterujących (enkapsulowanych komend SCSI) i za transmisję danych w systemie komunikacyjnym FireWire z pamięcią masową.

Na początku każdego cyklu Kontroler Cyklu generuje pakiet startu cyklu (CSP). Pakiet CSP wnosi do transmisji danych narzut informacji nadmiarowych o wielkości 150 bitów, występujący tylko raz dla każdego cyklu.

W założonym modelu Interwał Równych Szans (łączny czas wykonania jednej fazy transakcji asynchronicznej przez wszystkie węzły) będzie równy czasowi realizacji pojedynczej transakcji asynchronicznej, ponieważ na magistrali FireWire występują jedynie dwa węzły. Przed każdym interwałem równych szans występuje przerwa po resece arbitrażu.

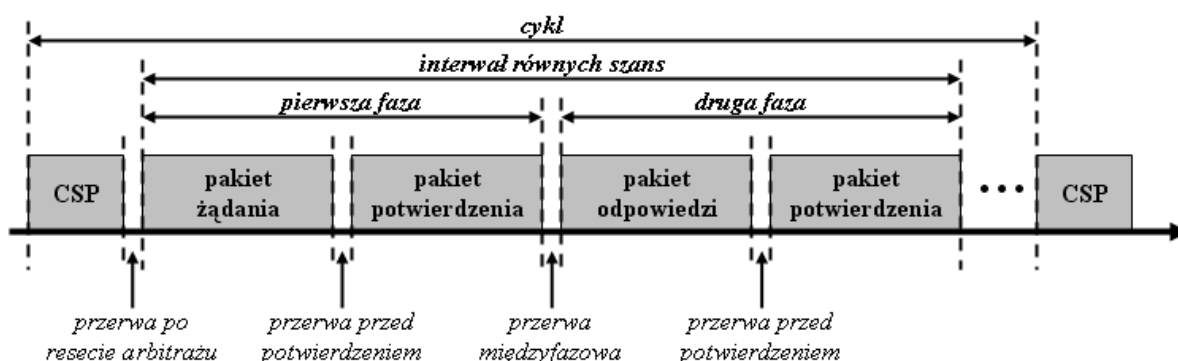
W systemie FireWire pomiędzy kolejnymi pakietami transakcji występują przerwy czasowe (tabela 9). W dalszych obliczeniach przyjęto maksymalne możliwe wartości przerw czasowych, a także zerowy czas trwania procedury arbitrażu asynchronicznego.

Tabela 9

Przerwy czasowe na magistrali FireWire

| Nazwa przerwy czasowej | Maksymalna możliwa wartość |
|--|----------------------------|
| Przerwa po resece arbitrażu (t_{ra}) | 10 μ s |
| Przerwa międzyfazowa (t_{mf}) | 10 μ s |
| Przerwa przed potwierdzeniem (t_p) | 0,05 μ s |

Na rys. 4 został zaprezentowany cykl, w ramach którego są realizowane asynchroniczne transakcje.



Rys. 4. Cykl w systemie FireWire

Fig. 4. Isochronous cycle in the FireWire communication system

Dana jest liczba D , będąca liczbą bajtów do odczytania z zewnętrznej pamięci masowej. Suma wszystkich informacji nadmiarowych wchodzących w skład jednej asynchronicznej transakcji odczytu wynosi:

$$l_{nad}^{FW} = \left(\underbrace{160}_{\text{pola pakietu}} + \underbrace{7}_{\text{sekwencja synchronizacyjna}} \right) + 2 \left(\underbrace{8+7}_{\text{pakiet potwierdzenia}} \right) + \left(\underbrace{176+7}_{\text{pakiet odpowiedzi}} \right) = 380 \quad (23)$$

W tabeli 10 zostały wyznaczone czasy transmisji poszczególnych pakietów wchodzących w skład jednej transakcji asynchronicznej (pakiet odpowiedzi zawiera maksymalny rozmiar pola danych).

Tabela 10

Czasy transmisji asynchronicznych pakietów

| Pakiet | Czas transmisji |
|--|----------------------|
| Asynchroniczny pakiet ządania odczytu bloku danych | $t_{pz}=0,42 \mu s$ |
| Pakiet potwierdzenia | $t_{pp}=0,04 \mu s$ |
| Asynchroniczny pakiet odpowiedzi na ządanie odczytu bloku danych | $t_{po}=41,42 \mu s$ |

Znając przerwy czasowe i czasy transmisji poszczególnych asynchronicznych pakietów, można wyznaczyć interwał równych szans (w tym przypadku również czas trwania jednej asynchronicznej transakcji odczytu bloku danych):

$$t_{int} = \underbrace{(t_{pz} + t_p + t_{pp})}_{\text{pierwsza faza}} + t_{mf} + \underbrace{(t_{po} + t_p + t_{pp})}_{\text{druga faza}} = 52,01[\mu s] \quad (24)$$

Uwzględniając przerwę po resecie arbitrażu, można wyznaczyć pasmo, jakie jest potrzebne do zrealizowania jednej asynchronicznej transakcji odczytu bloku danych:

$$t_{pint} = t_{int} + t_{ra} = 62,01[\mu s] \quad (25)$$

Interwał równych szans wraz z przerwą po resecie arbitrażu dla pakietu odpowiedzi z pustym polem danych wynosi:

$$t_{\text{pint}}' = \frac{l_{\text{nad}}^{FW}}{400} + 2t_p + t_{mf} + t_{ra} = 21,05[\mu\text{s}] \quad (26)$$

Czas transmisji danych na magistrali FireWire jest wyznaczony na podstawie następującego wzoru:

$$t_D^{FW} = \frac{8D}{400} \quad (27)$$

Czas transmisji jednego pakietu startu cyklu (CSP) wynosi:

$$t_{CSP} = \frac{150}{400} = 0,38[\mu\text{s}] \quad (28)$$

Przez pełny interwał równych szans rozumie się interwał równych szans w pełni mieszczący się w jednym cyklu. Na podstawie wzorów (25) i (28) można wyznaczyć liczbę pełnych interwałów równych szans w ramach jednego cyklu:

$$c_{\text{int}} = \left\lfloor \frac{\overbrace{\text{czas trwania cyklu}}{125} - t_{CSP}}{t_{\text{pint}}} \right\rfloor = 2 \quad (29)$$

Powyższy wzór nie odzwierciedla dokładnie sytuacji, jaka ma miejsce w rzeczywistym systemie FireWire, ponieważ w ramach pozostałego jałowego czasu w cyklu (braku aktywności na magistrali) jest wykonywana transmisja pakietu z kolejnego interwału równych szans. W tej sytuacji może dojść do drgań czasu trwania cyklu, co skutkuje zmianą czasu trwania kolejnego cyklu i zmniejszeniem pasma dla transakcji asynchronicznych. W celu uzyskania wartości odpowiadających przedstawionej sytuacji należałoby zamiast wzoru analitycznego (29) wykonać symulację komputerową zdarzeń dyskretnych i na jej podstawie wyznaczyć poszukiwane wartości. Pozostałe wolne pasmo (niezagospodarowane przez asynchroniczne transakcje) w cyklu wynosi:

$$t_{wp} = 125 - \underbrace{t_{CSP}}_{\text{pasma CSP}} - \underbrace{c_{\text{int}} t_{\text{pint}}}_{\text{pasma transakcji asynchronicznych}} = 0,61[\mu\text{s}] \quad (30)$$

Liczba potrzebnych transakcji do wykonania transferu odczytu danych wynosi:

$$n_{\text{int}} = \left\lfloor \frac{D}{\underbrace{2048}_{\text{rozmiar poleadanych}}} \right\rfloor \quad (31)$$

Przez pełny cykl rozumie się zgodnie z założeniem do wzoru (29) cykl, w ramach którego zostanie wykonana maksymalna możliwa liczba interwałów równych szans. Wzór (31) traktuje ostatnią transakcję transferu jako transakcję z pakietem odpowiedzi zawierającym pole danych o maksymalnym możliwym rozmiarze. Liczba pełnych cykli, w ramach których zostanie zrealizowany zlecony transfer danych:

$$Tr_c = \left\lceil \frac{n_{int}}{c_{int}} \right\rceil \quad (32)$$

Powyższy wzór uwzględnia tylko cykle, w których występują dwa pełne interwały równych szans. Następujący wzór uwzględnia już ewentualny niepełny ostatni cykl (np. tylko jeden interwał równych szans w ramach ostatniego cyklu):

$$Tr_c' = \left\lceil \frac{n_{int}}{c_{int}} \right\rceil \quad (33)$$

Na podstawie wzorów (26)-(27) i (30)-(33) można wyznaczyć całkowity czas realizacji zleconego transferu (operacja odczytu) w założonym modelu systemu FireWire. Czas ten wyrażony jest wzorem:

$$t^{FW} = \underbrace{t_{CSP} Tr_c'}_{\text{łączny czas transmisji CSP}} + \underbrace{t_{wp} Tr}_{\text{łączne niewykorzystane pasmo}} + \underbrace{n_{int} t_{pint}' + t_D^{FW}}_{\text{czas realizacji wszystkich transakcji}} \quad (34)$$

Podstawiając wcześniej wyliczone wartości i wyznaczone zależności, można wyznaczyć ostateczną postać wzoru na czas realizacji zleconego transferu:

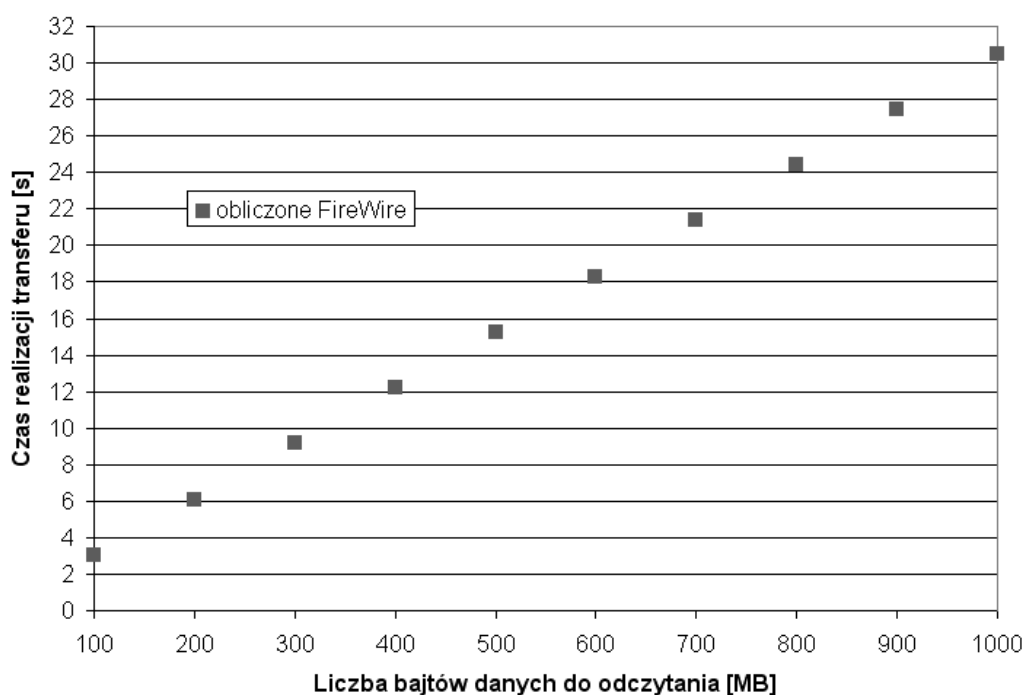
$$t^{FW} = 0,38 \left\lceil \frac{\left\lceil \frac{D}{2048} \right\rceil}{2} \right\rceil + 0,61 \left\lceil \frac{\left\lceil \frac{D}{2048} \right\rceil}{2} \right\rceil + 21,05 \left[\frac{D}{2048} \right] + \frac{8D}{400} \quad (35)$$

Jedynym parametrem wzoru (35) jest liczba bajtów danych do odczytania z zewnętrznej pamięci masowej. Wartość wyznaczona na podstawie wzoru (35) wyraża czas realizacji zleconego transferu w mikrosekundach.

Przykładowo, dla przyjętych założeń operacja odczytu 1 GB danych z zewnętrznej pamięci masowej zajęłaby 31 s i byłaby zrealizowana w ramach pół miliona transakcji (dokładnie w ramach 488282 transakcji). Na rys. 5 został zaprezentowany wykres funkcji ze wzoru (24) w zależności od wielkości odczytywanego pliku przy prędkości S400 w modelu systemu FireWire.

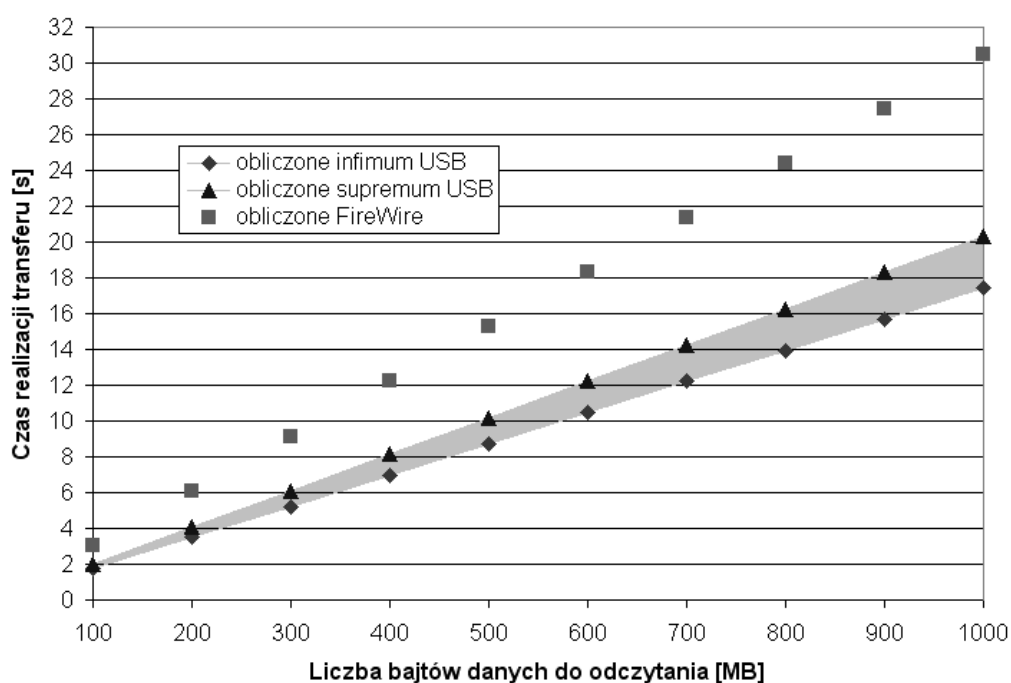
2.2. Porównanie systemów USB i FireWire

Mając wyznaczone zależności czasu realizacji transferu danych dla operacji odczytu, można dokonać porównania systemów USB i FireWire. Na rys. 6 zaprezentowano wykres funkcji czasu realizacji operacji odczytu w zależności od wielkości odczytywanego pliku (porcji danych) dla dwóch systemów USB i FireWire, przy maksymalnych możliwych prędkościach transmisji (USB 480 Mb/s, FireWire 400 Mb/s).



Rys. 5. Wykres zależności czasu realizacji transferu danych od wielkości odczytywanego pliku w modelu systemu FireWire

Fig. 5. Graph of execution time of data transfer in the FireWire communication system model



Rys. 6. Wykres zależności czasu realizacji operacji odczytu od wielkości odczytywanego pliku w modelach systemów USB i FireWire

Fig. 6. Comparison of execution time of data transfer in the USB and FireWire communication system models

Czas realizacji zleconego transferu danych w systemie USB jest krótszy od czasów wyznaczonych dla FireWire, ponieważ w USB, mimo że w ramach jednej transakcji jest przesyłana czterokrotnie mniejsza ilość danych (512 bajtów) niż w FireWire (2048 bajtów), każda transakcja wprowadza ponadczterokrotnie mniejszy narzut informacji nadmiarowych (86 bitów dla najlepszego przypadku) niż transakcja FireWire (380 bitów). Przerwy transmisji

w FireWire w ramach interwału równych szans powodują ograniczenie pasma w cyklu (125 μ s) do zaledwie dwóch pełnych transakcji asynchronicznych, gdzie w USB w ramach mikroramki trwającej tyle samo ile cykl FireWire jest możliwa realizacja 14 transakcji masowych (dla najlepszego przypadku). W rzeczywistym systemie FireWire, w ramach pozostałego wolnego pasma w cyklu, są transmitowane pakiety z kolejnej transakcji. W tej sytuacji transakcja FireWire nie musi zamykać się w ramach jednego cyklu, w przeciwieństwie do systemu USB, gdzie jest wymagane, aby transakcja zakończyła się przed końcem mikroramki, co prowadzi do nieefektywnego wykorzystania pasma.

3. Podsumowanie

Przeprowadzona analiza teoretyczna transferu danych wskazuje na interfejs USB jako ten, który należy wybrać do połączenia zewnętrznej pamięci masowej z komputerem. Należy mieć jednak na uwadze, że w analizie został jedynie uwzględniony transfer danych i to w idealnym systemie komunikacyjnym. W rzeczywistym systemie komunikacyjnym wiele czynników,

o których była mowa w punkcie 2, wpłynie znacząco na pogorszenie czasu realizacji transferu danych pomiędzy pamięcią masową a komputerem.

BIBLIOGRAFIA

1. Specyfikacja Universal Serial Bus 2.0
2. Specyfikacja Universal Serial Bus Mass Storage Class, Revision 1.3
3. IEEE Std 1394-1995: IEEE Standard for High Performance Serial Bus
4. Mielczarek W.: Szeregowy interfejs cyfrowy FireWire. Wydawnictwo Politechniki Śląskiej, Gliwice 2010.
5. Mielczarek W.: USB. Uniwersalny interfejs szeregowy. Helion, Gliwice 2005.
6. Anderson D.: Universal Serial Bus System Architecture. Mindshare, Inc., Addison-Wesley Developers Press, 1997.

7. Anderson D.: FireWire System Architecture, Mindshare. Inc., Addison-Wesley Developers Press, 2000.
8. Wesołowski K.: Podstawy cyfrowych systemów telekomunikacyjnych. Wydawnictwa Komunikacji i Łączności, Warszawa 2003.

Wpłynęło do Redakcji 13 marca 2012 r.

Abstract

This article shows the comparison of USB and FireWire ports due to time of data transfer in the communication system consisting of only two nodes. This configuration of the communication system is common in computer systems home use. One of the nodes is usually a computer, and the other is external storage such as removable media (pen drive) or external hard drive.

In order to compare the USB and FireWire two models of an „ideal” communication system were defined. Using these models the time of the data transfer was estimated. This allowed a comparison of USB and FireWire ports.

The analysis pointed to the USB interface as the one that should be used to connect external storage to your computer. However, please note that the analysis was performed on an ideal communication system and in the real communication system many factors, which were mentioned in the article, influences time of data transfer.

Adres

Michał SAWICKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, michal.sawicki@polsl.pl