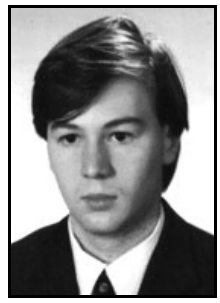
Grzegorz SUŁKOWSKI[2], Maciej TWARDY[2], Kazimierz WIATR[1,2]
[1] AKADEMIA GÓRNICZO-HUTNICZA
[2] ACK CYFRONET AGH

Implementacja standardu sieci Ethernet IEEE 802.3 w układach FPGA na potrzeby systemu bezpieczeństwa typu Firewall

Mgr inż. Grzegorz SUŁKOWSKI



Ukończył elektronikę na Wydziale EAIE Akademii Górniczo-Hutniczej w Krakowie. Od 2006 roku jest Konstruktorem Systemów Obliczeniowych w dziale Archiwizacji i Bezpieczeństwa Danych w ACK CYFRONET AGH. Jego zainteresowania skupiają się wokół algorytmów obliczeniowych oraz ich realizacji w logice reprogramowalnej.

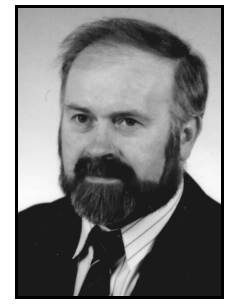
e-mail: Grzegorz.Sulkowski@cyfronet.pl

Mgr inż. Maciej TWARDY

Ukończył elektronikę na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Od 2005 roku kieruje Działem Archiwizacji i Bezpieczeństwa Danych w Akademickim Centrum Komputerowym CYFRONET AGH. Jego zainteresowania związane są z szeroko pojętą informatyką oraz projektowaniem układów cyfrowych w oparciu o logikę reprogramowalną.

e-mail: Maciej.Twardy@cyfronet.pl

Prof. dr hab. inż. Kazimierz WIATR



Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na Akademii Górniczo-Hutniczej oraz Dyrektor Akademickiego Centrum Komputerowego Cyfronet AGH. Prowadzone prace badawcze dotyczą komputerowego sterowania procesami, systemów wizyjnych, systemów wieloprocesorowych, układów programowalnych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.

e-mail: wiatr@agh.edu.pl

Streszczenie

W artykule omówiono wyniki implementacji standardu sieci Ethernet IEEE 802.3 w układach reprogramowalnych FPGA. Autorzy prezentują przyjętą formułę dekompozycji kontrolera sieciowego dokonując równocześnie charakterystyki poszczególnych modułów opisanych za pomocą języka VHDL w odniesieniu do wymogów stawianych przez standard. Przeprowadzone prace stanowią pierwszy etap realizacji projektu badawczego zmierzającego do opracowania w pełni sprzętowego systemu bezpieczeństwa typu Firewall. To nowatorskie podejście ma na celu stworzenie rozwiązania o wysokiej odporności na włamania oraz o dużej elastyczności wewnętrznej architektury, pozwalającej wykorzystać oferowane przez technologię FPGA możliwości rekonfiguracji zasobów sprzętowych.

Słowa kluczowe: systemy bezpieczeństwa informatycznego, układy programowalne, języki opisu sprzętu, Ethernet, firewall.

IEEE 802.3 Ethernet standard implementation in FPGA logic to the needs of the Firewall security system

Abstract

The article describes results of the Ethernet IEEE802.3 implementation in FPGA chip. Authors present applicated decomposition model of the Ethernet controller and characterize each of the sub-module created in VHDL language. Executed work is the first stage of the research project being intended to full hardware implementation of the firewall security system using FPGA technology. The goal of this innovatory approach is to prepare high security system with high inter-module flexibility with opportunities involved by FPGA recon-figuration functionality.

Keywords: information security systems, programmable logic, hardware description language, Ethernet, firewall.

1. Wstęp

Spośród wielu istniejących obecnie technologii realizacji sieci lokalnych (ang. *Local Area Networks*) zdecydowanie najbardziej popularną jest Ethernet. Historycznie termin ten pojawił się już w latach 70-tych ubiegłego wieku w rozwiązaniu sieci LAN opracowanym przez laboratorium badawcze firmy Xerox. To właśnie on stał się pierwowzorem dla pierwszej wersji specyfikacji IEEE 802.3, wydanej w roku 1983. Od tego czasu standard przeszedł dynamiczny rozwój, opracowano szereg jego modyfikacji, wprowadzających coraz większe prędkości transmisji, jak również obsługę nowych mediów komunikacyjnych.

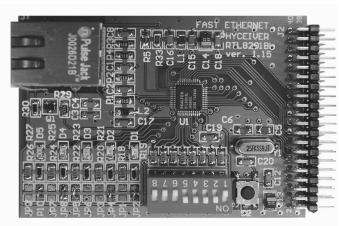
Obecnie Ethernet obejmuje swym zasięgiem całą rodzinę technologii opisywanych przez specyfikację 802.3, wykorzystujących metody dostępu typu CSMA/CD (ang. *Carrier Sence Multiple Access Collision Detect*) [3]. Powszechne wykorzystywanie Ethernetu doprowadziło do sytuacji, w której nie sposób już praktycznie znaleźć wśród organizacji naukowych, jak również publicznych i komercyjnych takiej, która nie wykorzystuje systemów komputerowych komunikujących się w tym standardzie. Informacja elektroniczna, transmitowana kanałami sieciowymi, ma w wielu przypadkach charakter poufny. Jej utrata bądź nieuprawnione wykorzystanie może spowodować nieodwracalne szkody wspomnianym wyżej instytucjom. Zagrożenia te, dodatkowo spotęgowane przez gwałtowny rozwój globalnych technologii komunikacyjnych (np. Internet), wymusiły konieczność szybkiego opracowania odpowiednich środków ochrony, pozwalających na ograniczenie dostępu do krytycznych zasobów (zgodnie z założeniami polityki bezpieczeństwa organizacji) [5]. Systemy bezpieczeństwa, realizujące te zadania, od początku swego istnienia do dziś, opierają się w zdecydowanej większości na rozwiązaniach programowych. Takie podejście posiada jednak wiele wad, do których należy przede wszystkim duża podatność na próby naruszenia bezpieczeństwa, związana m. in. z wykorzystywaniem błędów systemów operacyjnych, stanowiących platformę dla właściwego mechanizmu zabezpieczającego.

Autorzy, w ramach prowadzonego projektu badawczego, chcą wykazać istnienie odmiennej drogi budowy wydajnego i stabilnego systemu bezpieczeństwa typu Firewall, polegającej na jego implementacji w układach logiki reprogramowalnej FPGA. Pozwoli to wyeliminować wszelkie składniki oparte na oprogramowaniu, wykluczając tym sposobem możliwość włamań poprzez uruchamianie obcego kodu, przejmowanie uprawnień, itp. Reguły bezpieczeństwa (polityka bezpieczeństwa) pozostaną jedynym elementem systemu istniejącym poza strukturą FPGA. Analiza ruchu sieciowego na poziomie sprzętowym pozwala oczekiwać znacznego zwiększenia poziomu wydajności w stosunku do rozwiązań konwencjonalnych.

2. Platforma badawcza

Wszelkie prace implementacyjne prowadzono z wykorzystaniem płyty firmy Digilent z układem FPGA Xilinx Spartan2E. Niezbędne było jej doposażenie o minimum dwie karty sieciowe w standardzie FastEthernet. Zostały one zaprojektowane i wykonane w ramach realizacji projektu w oparciu o układ PHY (ang.

Physical Control Layer) RTL8201BL produkcji firmy Realtek. Odpowiada on za kodowanie, dekodowanie i synchronizację na poziomie nośnika danych, pozwalając na realizowanie transmisji z prędkościami 10 lub 100Mb/s przy zachowaniu pełnej zgodności ze standardami IEEE 802.3/802.u. Od strony funkcjonalnej istotna jest możliwość pełnej parametryzacji trybu pracy, zarówno w sposób manualny, jak również sprzętowo poprzez szeregowy interfejs MDC/MDIO. Karta wyposażona została w zestaw diod LED sygnalizujących kluczowe parametry, m.in.: istnienie poprawnego stanu łącza (ang. link), aktualną prędkość transmisji 10 lub 100Mb/s, wystąpienie kolizji na łączu. Karta interfejsu sieciowego została przedstawiona na rys. 1.



Rys. 1. Karta interfejsu sieciowego Ethernet 10/100 Mb/s
Fig. 1. Ethernet 10/100 Mb/s Network Interface Card

3. Realizacja sprzętowa

3.1. Tor odbiorczy

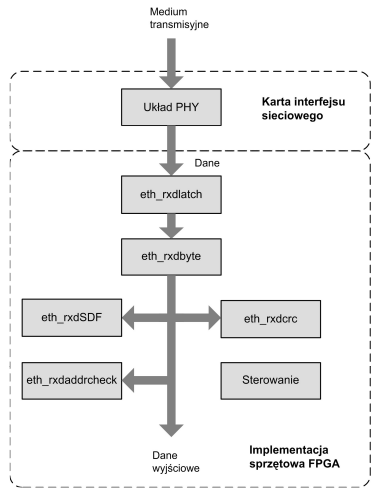
Zadaniem toru odbiorczego (RX) jest właściwe sformatowanie odebranych od warstwy fizycznej danych, weryfikacja poprawności ramki oraz jej przekazanie do warstw wyższych modelu sieciowego ISO/OSI. Funkcjonalność toru odbiorczego została zaprojektowany w oparciu o proponowany przez standard IEEE 802.3 [2] model proceduralny. Tor odbiorczy na najwyższym poziomie hierarchii (ang. *top level*) opisuje moduł eth_rxd, agregujący logikę zarządzającą, komunikacyjną oraz funkcjonalną, złożoną z czterech podstawowych elementów: eth_rxdlatch, eth_rxdbyte, eth_rxdcrc oraz eth_rxdaddrcheck. Dodatkowo zaimplementowano moduły wspomagające główną logikę toru odbiorczego: eth_rxdnible, eth_rxdnible2byte, eth_rxdSDF. Proces odbioru danych od warstwy fizycznej PHY w torze RX rozpoczyna się w module eth_rxdlatch, który jest bezpośrednim łącznikiem pomiędzy warstwą fizyczną a sieciową. Głównym zadaniem eth_rxdlatch jest zatrzaśnięcie linii sygnałowych układu PHY i przekazanie ich do następnego modułu – eth_rxdbyte.

Zastosowanie zatrzasku bezpośrednio na wejściu układu ma na celu wyeliminowanie błędów wynikających z niestabilnych wartości logicznych na sygnałów wyjściowych PHY.

Moduł eth_rxdbyte przetwarza odebrane dane wykorzystując dwa dodatkowe elementy: eth_rxdnible oraz eth_rxdnible2byte. Pierwszy z nich - eth_rxdnible - odpowiada za detekcje narastającego zbocza na linii zegarowej PHY (zatrzaśniętej w eth_rxdlatch) i wygenerowanie sygnału ważności danych dla modułu eth_rxdnible2byte. Z kolei eth_rxdnible2byte, reagując na narastające zbocze sygnału zegarowego PHY, przeprowadza następujące operacje:

- konwertuje odebrane nible do bajtów,
- zlicza liczbę odebranych bajtów,
- generuje znaczniki pola dla bieżącego pola ramki,
- sprawdza czy odbierana ramka jest ramką 802.3Q,
- sprawdza, czy liczba odebranych bajtów nie przekracza maksymalnego rozmiaru ramki,
- generuje znacznik początku ramki.

Sygnały wyjściowe z eth_rxdnible2byte przekazywane są do bloków eth_rxdSDF, eth_rxdaddrcheck oraz eth_rxdcrc. Pierwszy z wymienionych odpowiada za weryfikacje czy odbierane dane i stan toru RX są zsynchronizowane. Podejście takie ma na celu wyeliminowanie problemów powstałych w wyniku nieoczekiwanego wystąpienia stanu niskiego (logiczne 0) na linii RxDV w trakcie nadawania ramki. Po rozpoznaniu poprawnego początku ramki i właściwej synchronizacji toru RX, moduł eth_rxdcrc zaczyna wyliczać sumę kontrolną, która jest weryfikowana z zawartością pola FCS, przesyłanego wraz z ramką. Równocześnie z rozpoczęciem pracy przez eth_rxdcrc uruchamiany jest blok eth_rxdaddrcheck, weryfikujący MAC adres odbieranej ramki. Proces odbierania ramki kończy się w momencie, gdy na linii RxDV wystąpi stan niski. Odbierane dane są zapisywane w buforze toru RX skąd, po poprawnie zakończonym odbiorze, są odczytywane przez moduły warstw wyższych. Schemat blokowy toru odbiorczego przedstawiono na rys. 2.



Rys. 2. Schemat blokowy toru odbiorczego.
Fig. 2. Receiving channel block diagram.

Proces odbioru ramki przez zaprojektowany tor odbiorczy został sprawdzony modułem testującym (ang. *testbench*), symulującym zachowanie warstwy fizycznej PHY z użyciem rzeczywistych ramek sieciowych przechwyconych z produkcyjnej sieci Ethernet.

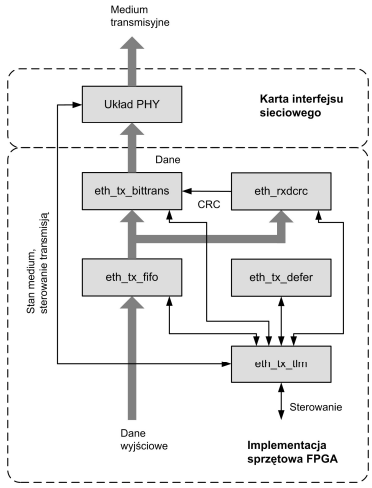
3.2. Tor nadawczy

Zadaniem toru nadawczego jest odpowiednie przygotowanie ramki Ethernetowej na podstawie danych przekazanych przez klienta warstwy MAC. Obejmuje ono dołożenie niezbędnych pól, takich jak: preambuła, pole startu ramki (ang. *Start Frame Delimiter*), w razie konieczności rozszerzenie ramki do minimalnej dopuszczalnej długości (ang. *padding*), a na koniec, w celu zabezpieczenia danych przed przekłamaniem, obliczenie sumy kontrolnej CRC. Tryb pracy toru nadawczego jest ściśle uzależniony od aktualnych parametrów medium transmisyjnego. Przekazywane są one do warstwy MAC z kart sieciowych, a konkretnie z układów PHY.

Specyfikacja IEEE 802.3 [2] podaje przykładowy model proceduralny toru nadawania, który stał się punktem wyjścia do opracowania implementacji sprzętowej przedstawionej na rys. 3.

Modelowi proceduralnemu odpowiada zestaw modułów sprzętowych, opisanych za pomocą języka VHDL. Funkcjonują one zgodnie z algorytmem CSMA/CD, realizując dwie podstawowe grupy zadań, zdefiniowanych w standardzie IEEE 802.3 [2]:

a) enkapsulację danych klienta MAC, obejmującą dołożenie niezbędnych pól, obliczenie sumy kontrolnej CRC – moduły eth_tx_bittrans oraz eth_tx_crc,
b) zarządzanie transmisją, obejmujące zachowywanie niezbędnych opóźnień czasowych, wykrywanie i obsługiwanie kolizji, ponawianie transmisji z wykorzystaniem mechanizmów backoff, rozszerzenia ramek oraz obsługę trybu burst mode - moduły eth_tx_tlm oraz eth_tx_defer.



Rys. 3. Tor transmisyjny: schemat blokowy
Fig. 3. Transmission channel: block diagram

Poniżej zaprezentowana została szczegółowa charakterystyka poszczególnych modułów, wchodzących w skład toru transmisyjnego:

- **moduł eth_tx_crc** – jest odpowiedzialny za generowanie 32 bitowej sumy kontrolnej CRC, transmitowanej jako ostatnie pole ramki Ethernet FCS (ang. *Frame Check Sequence*). Służy ona weryfikowaniu integralności transmitowanych danych w stacji odbiorczej. Do obliczenia ciągu kontrolnego wykorzystywany jest wielomian generacyjny CRC-32,
- **moduł eth_tx_bittrans** – odpowiada procesowi *BitTransmitter* w modelu IEEE 802.3 [2]. Jest to skończony automat stanów sterujący generowaniem oraz transmisją poszczególnych pól ramki Ethernet. Dodatkowo moduł zawiera główny licznik wysłanych nibli (połówek bajtów), pozwalający na właściwe pozycjonowanie elementów ramki,
- **moduł eth_tx_defer** – odpowiada procesowi *Deference* w modelu IEEE 802.3 [2], zarządzającemu opóźnianiem ramki, według poniższych reguł
  a) **tryb *half duplex*** - moduł nieustająco monitoruje stan medium (nawet wtedy, gdy ramki nie są transmitowane), śledząc stan sygnału obecności nośnej (ang. *carrierSense)* pochodzącego z układu PHY. Kiedy tylko stwierdzi, że medium jest zajęte, rozpoczyna opóźnienie rozpoczęcia ewentualnej transmisji. W momencie, kiedy medium staje się wolne (sygnał *carrierSense* zmienia wartość na 0 logiczne), moduł kontynuuje opóźnienie przez wymagany standardem okres równy wartości minimalnej szczeliny czasowej oddzielającej kolejno transmitowane ramki (ang. *interFrameSpacing)*. **Po tym czasie transmisja jest rozpoczynana niezależnie od stanu sygnału *carrierSense***, a po jej zakończeniu (lub w wypadku, gdy nie ma żadnych ramek do wysłania) moduł rozpoczyna ponowne monitorowanie stanu medium.
  b) **tryb *full duplex*** - moduł nie monitoruje stanu sygnału *carrierSense*. Po zakończeniu transmisji ramki odmierza jedynie wymagane opóźnienie równe *interFrameSpacing,*
- **moduł eth_tx_tlm** – kluczowy moduł toru transmisyjnego, zawierający mechanizmy zarządzania dostępem do medium transmisyjnego, zgodnie z funkcjonalnością protokołu rywalizacyjnego CSMA/CD, w szczególności obsługę kolizji. W trybie *half duplex,* w obrębie okna kolizyjnego, moduł monitoruje stan sygnału wystąpienia kolizji (ang. *Collision*), pochodzącego z warstwy fizycznej. W momencie jej wystąpienia wysyłany jest sygnał zakłócający (ang. *Jam*), przedłużający transmisję w celu zapewnienie właściwej propagacji informacji o wystąpieniu kolizji do wszystkich stacji współdzielących medium. Moduł eth_tx_tlm realizuje również algorytm z binarno-wykładniczym rozszerzaniem czasu (ang. *binary-expotential backoff*), randomizującym czas rozpoczęcia retransmisji ramki. Ponowienie transmisji opóźnia się o całkowitą wielokrotność *r* szczeliny czasowej równej oknu kolizyjnemu (ang. *collision window*), definiowanej jako parametr *slotTime*. Wartość zwielokrotnienia *r* stanowi liczbę losową z przedziału:

$$0 \le r \le 2^k, \quad \text{gdzie} \quad k = \min(n,10) \tag{1}$$

3.3. Zarządzanie

Moduł MII, zapewnia mechanizmy komunikacji pomiędzy warstwami 1 oraz 2 modelu ISO/OSI. Dostarcza informacji o aktualnych parametrach warstwy fizycznej medium transmisyjnego pochodzących z PHY do modułów RX i TX. Pozwala również zmieniać konfigurację wewnętrznych rejestrów PHY'a, a tym samym modyfikować konfigurację trybu pracy. Ponieważ omawiany moduł nie posiada dużej złożoności funkcjonalnej, przyjęto by implementacja była możliwie zwarta i jednocześnie nie zajmowała dużej liczby zasobów układowych. Pozwoliło to na wykorzystanie pojedynczej maszyny stanów uzupełnionej o kilka układów pomocniczych, jak np. dzielniki częstotliwości.

4. Wyniki implementacji - podsumowanie

Przetestowane tory nadawania i odbioru zostały zaimplementowane za pomocą oprogramowania ISE6.3i firmy Xilinx w układzie FPGA na platformie testowej opisanej w rozdziale 2. Uzyskane wykorzystanie zasobów sprzętowych zebrano w tabeli 1.

Tab. 1. Wykorzystanie zasobów sprzętowych
Tab. 1. Device utilization Sumary

Zajętość zasobów układowych Spartan XC2S200E–6				
	RX	**Utylizacja procentowa**	**TX**	**Utylizacja procentowa**
Number of Slice Flip Flops:	277 z 4,704	5%	179 z 4,704	3%
Number of 4 input LUTs:	393 z 4,704	8%	473 z 4,704	10%
Number of GCLKs:	1 z 4	25%	2 z 4	50%

Maksymalna wyliczona przez narzędzia do syntezy częstotliwość zegara systemowego bez optymalizacji wydajnościowej wynosiła około 75 MHz, co pozwala na pracę z szybkościami transmisji 10 i 100Mb/s. Kolejnym etapem rozwoju sprzętowego kontrolera sieci Ethernet jest obsługa standardu 1Gb/s. Wiąże się to nie tylko z niezbędnym uzupełnieniem funkcjonalnym opisu poszczególnych modułów w języku VHDL, ale również z zaprojektowaniem i wykonaniem dedykowanych kart interfejsów sieciowych.

Niezależnie dalszej rozbudowy sprzętowego kontrolera sieci Ethernet, obecna jego wersja pozwala na prowadzenie prac badawczych związanych z realizacją kluczowych elementów filtrujących systemu bezpieczeństwa typu Firewall.

Praca naukowa finansowana ze środków na naukę w latach 2006-2008 jako projekt badawczy.

5. Literatura

[1] K. Wiatr: Akceleracja obliczeń w systemach wizyjnych, Wydawnictwa Naukowo-Techniczne, Warszawa 2003.
[2] 802.3 IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks, 2002.
[3] K. Nowicki: Ethernet – sieci, mechanizmy. Infotech, Gdańsk 2006.
[4] K. Skahill: Język VHDL. Projektowanie programowalnych układów logicznych. Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
[5] W. Stallings: Ochrona danych w sieci i intersieci. W teorii i praktyce. Wydawnictwa Komunikacji i Łączności, Wydawnictwa Naukowo-Techniczne, Warszawa 1997.

Artykuł recenzowany