

dr hab. inż. Adam Milik, prof. PŚ.  
Politechnika Śląska  
Wydział Automatyki Elektroniki i Informatyki  
Katedra Systemów Cyfrowych  
ul. Akademicka 16  
44-100 Gliwice

Gliwice, 12.12.2021

**Recenzja rozprawy doktorskiej mgr. inż. Macieja Czekaja  
dla Rady Dyscypliny Informatyka Techniczna i Telekomunikacja  
Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie**

(podstawą opracowania recenzji jest uchwała Rady Dyscypliny  
Informatyki Technicznej i Telekomunikacji z dnia 18.10.2021)

**Tytuł rozprawy:** Hardware acceleration of traffic classifiers for high throughput Ethernet  
(Sprzętowa akceleracja klasyfikatorów ruchu w sieci Ethernet o wysokiej przepustowości)

**Autor rozprawy:** mgr inż. Maciej Czekaj

**Promotor rozprawy:** dr hab. inż. Ernest Jamro, prof. AGH

**Promotor pomocniczy:** dr inż. Agnieszka Dąbrowska-Boruch

**Dziedzina:** nauki techniczne

**Dyscyplina:** Informatyka

## **1. Zagadnienia naukowe rozprawy – cel i teza pracy**

Tematem rozprawy jest akceleracja sprzętowa klasyfikatorów ruchu w sieci Ethernet o wysokiej przepustowości. Autor w rozdziale 1 stawia następującą tezę: *Optymalne wykorzystanie pamięci podręcznej w systemach klasyfikacji pakietów prowadzi do znaczącej poprawy wydajności oraz ograniczenia poboru energii.*

W celu wykazania postawionej tezy autor w pierwszej kolejności stawia sobie za cel dokonanie analizy ruchu sieciowego. Zbudowany model pozwala na weryfikację konstrukcji układów klasyfikacji pakietów z wykorzystaniem pamięci asocjacyjnej (ang. Ternary Content Addressable Memory - TCAM) oraz pamięci podręcznej (ang. cache memory). Następnie autor dokonuje weryfikacji efektywności przekazywania pakietów z wykorzystaniem pamięci podręcznej. Pozwala to na zbadanie wpływu na efektywność działania pamięci z różnym stopniem asocjacyjności. Czytelnik zostaje zapoznany również z projektem szerokopasmowej karty sieciowej a także z wyzwaniem jakie stawia projektowanie szerokopasmowych urządzeń

sieciowych. W szczególności istotnym wyzwaniem jest przetwarzanie ramek w czasie rzeczywistym. Następnie autor przedstawia wykorzystanie pamięci podręcznej do defragmentacji pakietów. Wyznaczenie parametrów pamięci podręcznej połączeń wykorzystywanej w procesie defragmentacji wymaga utworzenia modelu statystycznego a na jego podstawie odpowiedniego środowiska testowego pozwalającego na zamodelowanie transakcji. W wyniku przeprowadzonych eksperymentów autor wyznaczył rozmiar pamięci pozwalający na uzyskanie zadanego poziomu pakietów scalanych.

Praca mgr inż. Macieja Czekaja odpowiada na współczesne problemy związane z dynamicznym rozwojem społeczeństwa cyfrowego, które objawiają się rosnącym przepływem danych w sieciach komunikacyjnych. Jak autor wykazał w pracy, odpowiednia klasyfikacja danych ma bardzo istotne znaczenie pozwalając na efektywne przenoszenie ramek, scalanie pakietów czy też eliminację ataków. Szczególne wymagania powstają w przypadku szerokopasmowych łącz, w których proces klasyfikacji musi odbywać się na bieżąco, zapewniając zachowanie przepływności danych. Bardzo istotnym elementem pracy jest zastosowanie układów FPGA do konstrukcji układu klasyfikacji. Należy zwrócić tutaj uwagę na istotne aspekty implementacji układowej poruszane w pracy. Układy programowalne FPGA nie implementują pamięci asocjacyjnych wykorzystywanych do klasyfikowania ruchu sieciowego. Wymagało to od autora opracowania struktury pamięci podręcznej oraz odpowiedniego generatora skrótu umożliwiającego klasyfikację ramek. Wykazanie własności statystycznych ruchu sieciowego pozwoliło dobrać rozmiar pamięci podręcznej i uzyskać trafność przekraczającą 90% zapytań. Oznacza to, że poniżej 10% pakietów wymagało przetwarzania programowego. Opracowane rozwiązanie wykorzystano również do defragmentacji pakietów. Podobnie i w tym przypadku autor zamodelował proces transmisji pakietów w celu wyznaczenia właściwości układu defragmentacji.

Uwzględniając powyższe stwierdzam że, praca doktorska mgr. inż. Macieja Czekaja wpisuje się w aktualny nurt opracowania systemów klasyfikacji ruchu w sieciach Ethernet implementowanych w układach reprogramowalnych FPGA. Łączy ona analizę teoretyczną ruchu w sieciach Ethernet z praktyczną implementacją układową. Wzajemne powiązanie dogłębnej analizy teoretycznej problemu połączonej z modelowaniem pozwala na precyzyjne opracowanie struktur układu o pożądanych własnościach. Podsumowując, praca odpowiada na współczesne problemy wzrostu przepływności danych w sieciach komputerowych co wymaga opracowania odpowiednich akceleratorów sprzętowych dla zapewnienia przetwarzania szerokopasmowego strumienia danych w czasie rzeczywistym. Należy również podkreślić opracowanie struktury układowej klasyfikatora ruchu stosując systematyczne metody analizy statystycznej i modelowania transmisji w sieciach Ethernet.

## 2. Zawartość i ocena merytoryczna

Recenzowana rozprawa doktorska przedstawia w sposób systematyczny konstrukcję układu klasyfikacji ruchu w sieci Ethernet. W opracowaniu rozwiązania końcowego autor wykorzystał opis statystyczny ruchu w sieci co pozwoliło na precyzyjne dobranie własności poszczególnych elementów w projektowanym urządzeniu.

W rozdziale pierwszym autor przedstawia cel pracy jakim jest skonstruowanie sprzętowego klasyfikatora ruchu dla sieci szerokopasmowych Ethernet. Na podstawie nakreślonego celu formułuje tezę: *„Optymalne wykorzystanie pamięci podręcznej w systemach klasyfikacji pakietów prowadzi do znaczącej poprawy wydajności oraz ograniczenia poboru energii.”* W dalszej części rozdziału został przedstawiony sposób udowodnienia postawionej tezy. Autor w swojej pracy udowodnił znacznie szerszą tezę, która obejmuje również wyznaczenia za pomocą modelu statystycznego ruchu sieciowego rozmiaru pamięci podręcznej oraz jej konfiguracji spełniającej zadane własności.

Rozdział drugi wprowadza czytelnika w charakterystykę zadań realizowanych przez różne urządzenia sieciowe. Nakreślony szeroki obraz współczesnych sieci pozwala scharakteryzować oczekiwania funkcjonalne w stosunku do systemów klasyfikacji i przekazywania danych. Następnie autor przedstawia architektury wybranych akceleratorów komercyjnych. Ostatecznie pozwala to sformułować zadania postawione projektowanemu akceleratorowi, który będzie działał w ciągu przetwarzania (inline). Jego zadaniem jest przejęcie większości zadań klasyfikacji pakietów w celu istotnego ograniczenia obciążenia procesu klasyfikacji wykonywanej na drodze programowej. Do przetwarzania programowego zostają przekazane pakiety, które nie mogły zostać poprawnie sklasyfikowane przez jednostkę sprzętową.

Postawione zadanie zaprojektowanie akceleratora dokonującego klasyfikacji pakietów implementowanego w strukturach FPGA wprowadza istotne ograniczenia wynikające z dostępnych zasobów. Głównym elementem wykorzystywanym w klasyfikacji w czasie rzeczywistym jest pamięć asocjacyjna z trójwartościowym wyborem dopasowania (wprowadzenie stanu nieokreślonego w procesie dopasowania). Implementacja struktury pamięci asocjacyjnej w układzie FPGA jest możliwa do przeprowadzenia za pomocą zasobów logicznych ogólnego przeznaczenia (przerzutniki, bloki tablicowe LUT). Uzyskanie wymaganych rozmiarów pamięci asocjacyjnej, powoduje że implementacja jest niecelowa. Wymaga to opracowania metody alternatywnej szybkiego odszukania wzorca pakietu. W tym celu autor proponuje budowę pamięci podręcznej (ang. cache memory). Pamięć jest adresowana za pomocą skrótu wyznaczonego na podstawie zawartości pól wykorzystywanych do klasyfikacji pakietu.

Aby wykazać celowość zaproponowanego rozwiązania autor w rozdziale 4 konstruuje model statystyczny ruchu sieciowego. Istotą modelu jest wyznaczenie interwału czasu między

przybyciem kolejnych pakietów dla danego połączenia. W ruchu sieciowym występuje lokalność pakietów, co jest związane ze sposobem wykorzystania połączeń. Wykazanie tej własności odpowiada również sposobowi funkcjonowania pamięci podręcznej z zastępowaniem strony, która była wykorzystana najdawniej. Przeprowadzona analiza pozwala na dobranie rozmiaru pamięci oraz stopnia asocjacyjności pamięci w celu osiągnięcia zadanego współczynnika trafień. Uzyskane rezultaty można wykorzystać również w systemach programowych.

Dysponując przesłankami do konstrukcji karty sieciowej z układem klasyfikacji ruchu w rozdziale 5 pracy autor przedstawia trudności i wyzwania projektowe. Istotą projektowanego urządzenia jest zapewnienie szerokopasmowego przepływu danych. Powoduje to konieczność zaprojektowania układów przetwarzania potokowego, które będą zdolne do przetwarzania napływającego strumienia informacji w czasie rzeczywistym. Jak wykazuje autor, przypadkiem skrajnym są ramki o najmniejszej możliwej długości. Na ich podstawie należy określić wymaganą wydajność obliczeniową. W pracy zostaje przedstawiona implementacja układu wyznaczania skrótu zaprojektowanego w strukturze potokowej. Autor również porusza problem wyścigu danych oraz sposobu jego rozwiązania w obliczeniach realizowanych w sposób potokowy. Rozdział podsumowuje analiza poboru mocy, co pozwala wykazać, że jego głównym źródłem są bloki pamięci podręcznej. Pobór mocy układów pamięciowych jest porównywalny z poborem części logicznej dla struktury pamięci złożonej zaledwie z 4096 wpisów (podczas gdy wyznaczony rozmiar pamięci powinien wynosić 512k wpisów)

Opracowanie układu klasyfikacji ruchu sieciowego skłania również autora do zastosowania opracowanej metody w procesie defragmentacji pakietów. Istotnym elementem przemawiającym za defragmentacją pakietów jest możliwość ich późniejszego klasyfikowania. W celu opracowania efektywnej metody grupowania pakietów została wykorzystana pamięć podręczna połączeń. Następnie opracowano model pozwalający zweryfikować efektywność działania opracowanych rozwiązań. W tym celu autor posłużył się rzeczywistymi danymi zarejestrowanego ruchu sieciowego. W zarejestrowanych sekwencjach zaledwie 1% pakietów był zdefragmentowany. Na podstawie przeprowadzonych symulacji wykazano, że pamięć połączeń 2 lub 4drożna o 128 wpisach wykazuje wysoką efektywność grupowania pakietów.

Podsumowanie efektów prac autor dokonuje w rozdziale 7. **Na podstawie lektury pracy można stwierdzić, że postawiona w rozdziale 1 teza została udowodniona a cele osiągnięte przez autora.** W szczególności Autor wykazał skuteczne zastosowanie pamięci podręcznej do klasyfikacji pakietów oraz ich grupowania. W wyniku przeprowadzonych analiz statystycznych oraz symulacji autor dobrał rozmiar pamięci, tak aby uzyskać zadane parametry dla procesów klasyfikowania i scalania pakietów.

Wykaz literatur zawierający 133 pozycje należy uznać za wyczerpujący. W zestawieniu znajdujemy szeroki przekrój publikacji dobrze ilustrujący analizowaną tematykę uwzględniając

również publikacje ukazujące się w czasie tworzenia pracy. Wskazane jest aby źródła internetowe, do których odwołuje się praca zostały dołączone na nośniku elektronicznym ze względu na ich ulotność.

### **3. Uwagi krytyczne, wątpliwości, pytania**

Podczas obrony publicznej chciałbym poznać zdanie doktoranta na poniższe wątpliwości:

1. Jak autor rozwiązał problem analizy jednoczesnej dwóch ramek w przypadku konieczności wprowadzenia obu z nich do pamięci notatnikowej. Nie jest do końca jasne w jaki sposób zostały rozwiązane problemy związane z wyścigiem danych w przypadku jednoczesnej analizy danych. Identycznym problemem jest aktualizacja pola LRU pamięci podręcznej podczas jednoczesnego przetwarzania dwóch ramek danych.
2. Ze względu, że autor wybrał implementację układową w strukturach programowalnych celowym wydaje się przedstawienie pewnych szczegółów opisu w języku Verilog użytych w konstrukcji urządzenia. Przykładowo, autor pominął szczegóły odnoszące się do odwzorowania dwubramowych pamięci RAMB. Przedstawione w pracy schematy blokowe przepływu danych również zostały przedstawione w oderwaniu od opisu, na podstawie, którego powstały. W przypadku wymagających implementacji, celowe jest przedstawienie szczegółów, kładąc nacisk na sposób opisu oraz uzyskiwane w wyniku syntezy logicznej efekty.
3. W jaki sposób została przeprowadzona analiza poboru energii w systemie pamięci podręcznej? Autor nie przedstawił sposobu analizy. Istotą analizy poboru mocy jest wyznaczenie mocy dynamicznej, wynikającej z przełączeń układu. Jest ona zależna od sposobu funkcjonowania układu. W tym celu należy przeprowadzić proces symulacji struktury układowej z wykorzystaniem środowiska testowego możliwie wiernie oddającego sposób działania. W przypadku zastosowanej pamięci podręcznej jest ona adresowana skrótem wyznaczonym na podstawie zawartości wybranych pól przetwarzanej ramki. Z przedstawionych rozważań przez autora wynika, że wszystkie bloki pamięci RAMB pozostają aktywne, podczas gdy powinien odpowiadać blok pamięci, dla którego wyznaczony skrót należy do przestrzeni adresowej.
4. W celu określenia parametrów funkcjonalnych w procesie defragmentacji pakietów autor zaproponował przeprowadzenie symulacji z wykorzystaniem losowego rozmieszczenia pakietów fragmentowanych. Przedstawiony proces modelowania jest ciekawym ujęciem problemu, przy czym charakteryzuje się on znaczną złożonością obliczeniową. Powstaje zatem pytanie. Czy nie należało dokonać próby podsumowania w ujęciu wskaźników statystycznych umożliwiających projektowanie lub konfigurację układu defragmentacji pakietów na podstawie statystycznej analizy ruchu rzeczywistego. Warto zauważyć, że

układy FPGA stwarzają unikalną możliwość konstruowania dedykowanych rozwiązań. W odróżnieniu od implementacji programowej implementacja sprzętowa oferuje działanie równoległe nie wnosząc dodatkowego narzutu na czas przetwarzania. Oznacza to, że układ analizy statystycznej ruchu może działać całkowicie niezależnie, zbierając informacje niezbędne do wyznaczenia parametrów układu defragmentacji na podstawie rzeczywistego ruchu sieciowego.

5. Analizując zaproponowane metody weryfikacji symulacyjnej szczególnie w przypadku defragmentacji pakietów, nasuwa się pytanie czy nie jest celowym uproszczeniem modelowania przez wykorzystanie modeli transakcyjnych. Modele takie pozwalają uniknąć wielu szczegółów związanych z przekazywaniem pakietu a jednocześnie zachować dokładność modelowania wymaganą przy analizie wskazanego zagadnienia. Pozwala to zdecydowanie skrócić czas symulacji a uzyskane wyniki są równie wiarygodne jak te uzyskane w długotrwałej symulacji z uwzględnieniem szczegółów działania układu.

#### 4. Uwagi szczegółowe

Przedstawiona rozprawa ma również swoje słabe strony i pewne niedociągnięcia. Chciałbym podkreślić, że znaczna część uwag ma charakter polemiczny.

Niegrzeczny jest sformułowanie prędkości wyszukiwania wyrażonej przez częstotliwość. W tym przypadku należałoby wyrazić czas porównania, który wynosi około 500ps a nawet może zostać skrócony do 200ps. Pozwala to wykonać od 2 do 5 Gp/s (gdzie Gp oznacza  $10^9$  porównań)

Podsumowanie rozdziału 4 (str. 558) mówi o projektowaniu układu (hardware design) bardziej precyzyjnym ujęciem w tym przypadku jest wskazanie na architekturę układu (hardware architecture)

Schemat blokowy inteligentnej karty sieciowej (str. 59, rys. 5.1) przedstawia schemat blokowy złożony jedynie z bloków bez wzajemnego powiązania. Na schemacie występują bloki o identycznej nazwie (Checksum). Block Customer logic w kolorze żółtym nie został objaśniony w legendzie.

Pamięć blokowa RAMB oferuje dostęp do matrycy pamięciowej poprzez dwa pełne porty (odczyt-zapis). Sformułowanie przez autora zdanie o zwiększeniu liczby portów pamięci (str. 59) jest prawdziwe tylko w przypadku zwiększenia liczby portów przeznaczonych do odczytu poprzez równoległe połączenie pamięci, przy czym jeden z portów będzie portem wspólnym wszystkich modułów i będzie on przeznaczony do zapisu danych (za jego pomocą można również odczytać dane).

Schemat blokowy przepływu danych, który został pokazany przez autora na rys. 5.6 (str. 65) jest wynikiem specyficznego opisu w języku Verilog. Generalizacja wykorzystania przekształcenia iloczynu logicznego grupy sygnałów za pomocą multiplexera nie jest uzasadniona. W przypadku odwzorowania technologicznego należy zwrócić uwagę na możliwość łączenia bloków tablicowych za pomocą dodatkowych multiplexerów. Oznacza to że sposób odwzorowania będzie bazował na blokach tablicowych oraz elementach dodatkowych umożliwiających zwiększenie efektywności odwzorowania. Na schemacie przedstawiono port wejściowy `key[33:0]`. Sygnały `key[33:32]` nie są wykorzystywane. Autor nie uzasadnił w pracy pominięcia tych sygnałów.

Pamięci typu RAMB nie są jednym z dostępnych elementów pamięciowych (str. 65<sup>4</sup>). Obok pamięci RAMB występują pamięci powstające z bloku tablicowego LUT (ang. distributed RAM) w zależności od architektury występują w plastrach dedykowanych Slice-M. Historycznie poprzedzały one implementację pamięci blokowych RAMB, która pojawiała się w układach Virtex.

Koncepcja utworzenia pamięci RAMB na podstawie podzbioru/wyboru sygnałów nie została wyjaśniona poprawnie (str. 65<sup>4</sup>). Istotą odwzorowania pamięci RAMB jest wykorzystanie odpowiedniego wzorca w języku opisu sprzętu oraz przyłączenie odpowiednich sygnałów.

Przy konstrukcji układów o dużej wydajności obliczeniowej i przepustowości danych istotą jest spełnienie zadanych wymogów czasowych. Autor w podsumowaniu rozdziału 5 skoncentrował się na pobieranej mocy, podczas gdy ze wzrostem liczby wykorzystanych bloków pamięci należy zwrócić szczególną uwagę na wydłużenie się czasu propagacji sygnału poprzez rozbudowę struktury układów multipleksujących (wzrost liczby warstw logicznych układu).

Algorytm rekonstrukcji pakietów (str. 75) wskazuje, że pakiety mogą zostać odebrane w dowolnej kolejności natomiast odtworzenie podzielonej informacji następuje po odbiorze pakietu ze znacznikiem MF (More Fragments) = 0. Tak sformułowany algorytm będzie prowadził do odrzucenia pakietu danych w przypadku gdy ramka końcowa (MF = 0) zostanie odebrana przed przesłaniem wszystkich pakietów poprzedzających.

Algorytm 6.1 (str. 83) nie jest jasne dlaczego wymagany jest zbiór pakietów  $S'$  skoro zbiór  $S$  nie podlega modyfikacji. Sposób wyboru pakietu opisany w algorytmie oznacza wybór  $N$  początkowych sekwencji podczas gdy pozostałe sekwencje nie są wykorzystywane. Powyższe podejście nie pozwoli na wykorzystanie wszystkich przygotowanych wzorców pakietów w losowo tworzonych sekwencjach. W przedstawionym zapisie zastosowano duże litery do oznaczania zmiennych np.  $N$  (w kontekście algorytmu jest to stała). Przyjmuje się aby zmienne oznaczać małymi literami. W rozważanym przypadku zamiast  $N$  dla czytelności lepiej użyć  $n$ . W wyniku działania algorytmu otrzymuje się zbiór  $\{V_1, \dots, V_M\}$  przy czym  $V_i$  nie został

zdefiniowany uprzednio. Na podstawie analizy korku 7 można wnioskować że  $V_i$  jest zbiorem uporządkowanym pakietów (wektorem) powstałym w wyniku losowego wyboru ze zbioru  $S$  ( $S'$ ).

Autor powinien unikać odwołania w przód, co utrudnia lekturę i poznanie rozwiązania. W przypadku rozważanego algorytmu 6.1, algorytmy pomocnicze 6.2, 6.3 można było przedstawić w pierwszej kolejności, gdyż nie odwołują się do innych elementów. Przedstawienie algorytmu 6.1 powinno nastąpić w dalszej kolejności przez co można uniknąć rekurencyjnej analizy tekstu.

#### 4.1. Nieakceptowalne sformułowania

- str. 9<sup>16</sup> autor używa terminu cache'ujący, który jest używany potocznie, podczas gdy istnieje możliwość użycia jednoznacznego terminu ugruntowanego w języku polskim jak pamięć podręczna, pamięć notatnikowa
- Równanie 3.1  $R_k$  nie jest równoważny ( $\Leftrightarrow$ ) tylko równy ( $=$ )
- Równanie 3.2 nieokreślony czynnik  $i$ . Zapis  $\min$  jest niejednoznaczny bez określenia warunku:  $\min_i(R_i = 1)$ . Równanie może zostać zapisane, jako enkoder priorytetowy warunku
- str.35<sub>4</sub> Zakres dekodowania z maskowaniem – autor powinien zauważyć, że wartość dekodowana przez maskowanie jest w zakresie  $1 < p < 2^n$  a nie tylko strony lewej co sugeruje zakres od 0 jeżeli  $p \in \mathbb{N}$
- Przykład (str. 35<sub>1</sub>)  $p < 5$  nie jest równoważne  $((p < 4) \wedge (p = 5))$  a powinno być  $((p < 4) \wedge (p = 4))$
- Sformułowanie „is notoriously difficult to implement” powinno brzmieć “is difficult to implement”
- Sformułowanie “Li et al. [66] is an example of” Li nie jest przykładem a opracował lub przedstawił algorytm. Powyższe zdanie może zostać zapisane “An example of ... can be found in Li *et.al.* [66]”
- Sformułowanie żargonowe i potoczne: str. 53<sup>13</sup> the direct mapped-cache wins the contest
- Schemat blokowy pokazany na rys. 5.8 zawiera elementy z nieprzyłączonymi wejściami (np. bramka AND – EN\_RD0\_i, przerzutniki validKey\_reg\_reg, data\_valid\_reg\_reg, blok keyRam)
- module implemented in SystemVerilog – w tym przypadku powinno zostać użyte sformułowanie module described using SystemVerilog – język opisuje a nie jest odwzorowaniem
- Tabela 5.4. Zawiera liczbę wykorzystanych pamięci RAMB wyrażoną jako liczbę ułamkową (wiersze 1, 3, 5). Czy oznacza to, że 0.5 pamięci można wykorzystać do



innego celu? Wykorzystanie pamięci należy podać w liczbie całkowitej użytych bloków. W celu zilustrowania częściowego wykorzystania należy podać liczbę wykorzystanych bitów lub słów do całkowitej liczby bitów/słów dostępnych.

- Tytuł rozdziału 6.1.2 Rationale – brak dokończenia myśli o jakie przesłanki chodzi autorowi
- Str. 82<sub>10</sub> – nierówność nieostra powinna zostać opisana za pomocą odpowiedniego symbolu matematycznego (np.  $\geq$ ) a nie ze złożenia dwóch znaku równości i nierówności ( $>=$ )

#### 4.2. Niedociągnięcia edytorskie, drobne błędy

- rys. 3.1 (str. 33) Brak wskazania priorytetu analizy może być mylące dla czytelnika. Wymaga uważnego przeczytania wcześniejszego akapitu
- tabela 3.1 (str. 35) przecina kontekst wypowiedzi
- rys. 3.6b – niejednoznaczny sposób narysowania przerzutnika/komórki pamięci - symbol wejścia sygnału zegarowego został użyty w stosunku do wyjścia przerzutnika komórki pamięci
- str. 38<sup>5</sup> jest: can is –zbędne can
- str. 38<sup>13</sup> jest: machieve – powinno być: achieve
- str. 38<sub>13</sub> przypadkowe e na początku linii
- str. 39<sup>14</sup> brak wprowadzenia do listy zbierających własności
- str. 44<sup>2</sup> jest nun-uniformity – powinno być non-uniformity
- str. 45<sup>6</sup> jest e,g.,: - powinno być e.g.:
- str. 48<sup>7</sup> length 64 or – powinno być length of 64 or
- str. 53 rys. 4.10. – prawie nieczytelne podpisy osi, większość wykresów jest bardzo drobna – autor powinien zwrócić uwagę na dobór wielkości wykresu i legendy wielkości do tekstu podstawowego
- str. 58<sub>4</sub>, str. 60<sup>12</sup> – jednostka GB powinno być Gb
- str. 59<sub>2</sub> – ram powinno być RAM
- str. 63<sub>8</sub> – dwukrotnie The the
- str 63<sub>9</sub> – niekompletne zdanie: However, contrary to the CRC, the polynomial is an input variable and as long as the hashed message...
- str. 64<sup>9</sup> – jest 3-bt powinno być 3-bit
- str. 64<sup>11</sup> – przypadkowy znak: e
- str. 65 – rys. 5.6 i rys 5.7 – bardzo drobne elementy i podpisy – trudne do odczytania na wydruku
- str. 65<sub>3</sub> – jest figure 4.8 powinno być figure 5.8
- str. 72<sup>19</sup> – jest defined powinno być defines

- str. 74<sup>4</sup> – jest estension powinno być extension
- str 75<sub>2</sub> – jest it powinno być is
- str. 84 wzór 6.3 jest  $x_1$  powinno by  $x_1$  (indeks dolny)
- str. 92<sup>5</sup> – jest ration powinno być ratio
- str 94<sup>14</sup> – jest wort powinno być the worst

## **5. Ocena końcowa rozprawy**

Uważam, że przedstawione w rozprawie doktorskiej oryginalne pomysły, modele statystyczne, oraz efekty implementacji wnoszą istotny wkład w rozwój dyscypliny Informatyka Techniczna i Telekomunikacja. Oprócz przedstawionej rozprawy doktorant jest autorem dwóch publikacji naukowych prezentujących zagadnienia związane z konstruowaniem pamięci podręcznej w systemach przekazywania i defragmentacji pakietów danych.

Podsumowując stwierdzam, że pomimo pewnych niedociągnięć, rozprawa doktorska Pana mgr. inż. Macieja Czekaja spełnia wymagania stawiane w ustawie "Prawo o szkolnictwie wyższym i nauce", zatem wnoszę o dopuszczenie rozprawy doktorskiej do publicznej obrony i dalszych etapów przewodu doktorskiego w dyscyplinie Informatyka Techniczna i Telekomunikacja.