

AKADEMIA GÓRNICZO-HUTNICZA



Wydział Informatyki, Elektroniki i Telekomunikacji

Układy logiki programowalnej w serwisach zgodnych z paradygmatem SOA wspomagających nadzór wizyjny

Autoreferat rozprawy doktorskiej

Autor:

mgr inż. Robert Brzoza-Woch

Promotor:

prof. dr hab. inż. Krzysztof Zieliński

Kraków, 2013

Streszczenie

W pracy zaprezentowano nowy sposób konstrukcji cyfrowych systemów nadzoru wizyjnego. Opracowane rozwiązanie zakłada pracę każdej kamery jako usługi sieciowej zgodnej z paradygmatem SOA (Service Oriented Architecture) oraz użycie do implementacji jej funkcjonalności układów logiki programowalnej. Zastosowanie układów logiki programowalnej zwiększa możliwości obliczeniowe poszczególnych kamer, a zgodność z paradygmatem SOA znacząco ułatwia integrację infrastruktury nadzoru wizyjnego z innymi systemami informatycznymi.

Na przykładzie opracowanej przez autora i przetestowanej w praktyce platformy sprzętowo-programowej została objaśniona metodologia tworzenia systemów nadzoru wizyjnego zgodnych z paradygmatem SOA. Platforma ta została zaprojektowana głównie do rozwoju usług realizujących nadzór wizyjny i bazuje na nowoczesnym układzie FPGA (Field Programmable Gate Array) oraz dodatkowych modułach sprzętowych.

W zaprezentowanym rozwiązaniu obraz jest przetwarzany bezpośrednio w każdej kamerze realizującej nadzór wizyjny, dzięki czemu transmisja danych może zostać ograniczona do przesyłania informacji na temat wykrytych zdarzeń. Takie podejście eliminuje konieczność przesyłania dużych ilości danych graficznych przez sieć oraz poprawia skalowalność systemu.

1 Wprowadzenie

Na wstępie zaprezentowano przede wszystkim cel i założenia pracy doktorskiej, omówiono najważniejsze osiągnięte rezultaty oraz przedstawiono uzyskany przez autora wkład w rozwój dziedziny systemów automatycznego nadzoru wizyjnego.

1.1 Motywacja do podjęcia tematu

Architektura klasycznych systemów nadzoru wizyjnego opiera się na scentralizowanym przetwarzaniu danych uzyskanych z wielu kamer. Niezależnie, czy są to systemy analogowe, czy cyfrowe, z architektonicznego punktu widzenia działają one podobnie. Centralna jednostka nadzoru wizyjnego w takich przypadkach przetwarza bardzo duże ilości danych i – najczęściej – są to dane redundantne. Typowo zachodzi także konieczność stosowania osobnej infrastruktury sieciowej dla systemu nadzoru wizyjnego, co podwyższa całkowite koszty takiego systemu oraz wpływa negatywnie na jego skalowalność pod kątem rozbudowy istniejącej konfiguracji. Aby umożliwić wykorzystanie istniejącej infrastruktury sieciowej budynków do implementacji systemów nadzoru wizyjnego, należałoby znacząco ograniczyć ilość przesyłanych danych. Wtedy także zarządzanie potencjalnie elastycznym systemem nadzoru wizyjnego może zostać uproszczone przez zastosowanie prostych i otwartych standardów komunikacyjnych, a zmniejszenie ruchu sieciowego stanie się kolejnym czynnikiem poprawiającym skalowalność pod względem ilości węzłów przesyłających dane.

1.2 Tematyka i zakres rozważań

W niniejszej dysertacji autor koncentruje się na następujących zagadnieniach związanych z tematyką nadzoru wizyjnego:

- ogólna architektura systemów nadzoru wizyjnego oraz sposoby ich integracji z innymi systemami informatycznymi,
- umiejscowienie logiki przetwarzania danych w różnych architekturach systemów nadzoru wizyjnego,
- wybrane aspekty implementacji jednego węzła (jednej kamery) systemu nadzoru wizyjnego pracującego zgodnie z zaproponowanymi założeniami.

Algorytmy przetwarzania obrazu są nieodzownym elementem nowoczesnych, cyfrowych systemów nadzoru wizyjnego. W omawianej pracy autor skoncentrował się na dwóch wybranych algorytmach przetwarzania obrazów. Zostały one omówione w zakresie koniecznym do ogólnego zrozumienia ich działania wraz z podaniem źródeł literaturowych zapewniających większy poziom szczegółowości (np. [40, 16]).

1.3 Teza pracy

Brzmienie tezy pracy jest następujące:

Użycie rozwiązań sprzętowych zbudowanych z zastosowaniem układów programowalnych pozwala uzyskać serwisy nadzoru wizyjnego o podwyższonej ergonomii i zwiększonej łatwości integracji z systemami klasy enterprise przy zachowaniu aspektów ekonomicznych i wybranych parametrów QoS.

Teza pracy dotyczy wybranych aspektów implementacji i wydajności systemów nadzoru wizyjnego. Zakłada się w niej wykorzystanie układów logiki programowalnej (PLD – *Programmable Logic Devices*) do implementacji tzw. inteligentnej kamery (ang. *smart camera*) przeznaczonej budowy elastycznych i łatwych w instalacji systemów nadzoru wizyjnego zgodnych z paradygmatem architektury zorientowanej na usługi SOA (*Service Oriented Architecture*). Zgodność z SOA pozwala na łatwą integrację nadzoru wizyjnego z innymi systemami informatycznymi klasy biznesowej. Wybrane aspekty ekonomiczne i jakościowe QoS (*Quality of Service*) w proponowanym rozwiązaniu muszą pozostać zachowane na akceptowalnym poziomie dla celów nadzoru wizyjnego.

1.4 Wkład w rozwój dziedziny

Główne osiągnięcia autora omawiane w pracy doktorskiej są następujące.

1. W ramach prowadzonych prac autor zaprojektował i wykonał w praktyce platformę sprzętowo-programową do rozwoju usług sieciowych (ang. *Web Service*) wspomagających nadzór wizyjny, w szczególności mogącą pracować jako inteligentna kamera. W skład platformy wchodzi część zrealizowana jako fizyczny sprzęt oraz elementy realizowane programowo, m.in. konfiguracja logiki programowalnej układu FPGA (*Field Programmable Gate Array*), program sterujący mikrokontrolerem implementowanym wewnątrz FPGA.

2. Bazując na wyżej wymienionej platformie sprzętowej, autor pracy wraz z współpracownikiem utworzyli usługę (serwis) detekcji ruchu zgodną z paradygmatem SOA. Usługa ta ma funkcjonalność wykrywania poruszających się obiektów w wybranym przez klienta obszarze zainteresowania. Wyniki detekcji są umieszczane w wewnętrznej kolejce zdarzeń, która może zostać odczytana przez klienta usługi. Klient usługi ma także możliwość zadania parametrów pracy modułu detekcji ruchu takich jak: interwał czasowy zapisu do kolejki, czułość, częstotliwość uaktualniania wzorca tła. Usługa detekcji ruchu działa w oparciu o specjalnie opracowany, sprzętowy moduł akceleracji obliczeń zaimplementowany w logice programowalnej układu FPGA.
3. Na opracowanej platformie sprzętowej autor pracy wraz z współpracownikiem utworzyli także usługę klasyfikacji obiektów. Usługa ta może służyć do rozróżniania obiektów podobnych, w szczególności: marek samochodów, typów pojazdów, ludzkich twarzy. Klient usługi, po wysłaniu zapytania, może otrzymać informację o nazwie ostatnio wykrytego obiektu z uprzednio wyuczonego zbioru wykrywanych obiektów. Przy implementacji tej usługi udało się uzyskać wysoki stopień sprzętowego przyspieszenia obliczeń. Mimo, że algorytmy przetwarzania obrazu nie stanowią głównego elementu rozprawy doktorskiej, jej autor wniósł duży wkład w implementację tych algorytmów w układzie FPGA.
4. Autor pracy opracował mechanizmy zdalnej rekonfiguracji dla usług działających na układach programowalnych. Mechanizmy te zostały z powodzeniem zaimplementowane w praktyce na opracowanej platformie sprzętowej. Zdalna rekonfiguracja pozwala na zmianę funkcjonalności usługi realizowanej sprzętowo przez to samo fizyczne łącze, którego używa usługa dzięki czemu nie ma konieczności uzyskiwania fizycznego dostępu do urządzenia realizującego usługę.

1.5 Własny wkład autora oraz wykorzystanie gotowych rozwiązań

W tej części wymieniono elementy zrealizowane samodzielnie przez autora oraz zaznaczono, przy realizacji których współpracował. Wymieniono także gotowe moduły, z których autor korzystał w realizacji platformy sprzętowej do nadzoru wizyjnego.

Autor pracy doktorskiej:

- zaproponował nowe podejście do architektury systemów nadzoru wizyjnego zgodnych z paradygmatem SOA,
- wykonał projekt sprzętu uniwersalnej platformy dla usług realizujących nadzór wizyjny,
- wykonał w praktyce i samodzielnie uruchomił egzemplarze testowe platformy,
- opracował wewnętrzną, implementowaną w układzie FPGA architekturę platformy,
- opracował aplikację dla modułów sieciowych EM1206,

- zaproponował własne, praktyczne rozwiązanie zdalnej rekonfiguracji układu FPGA platformy wraz z dedykowanym protokołem sieciowym i aplikacją sterującą,
- jest współautorem implementacji modułów realizujących przetwarzanie obrazu dla usług detekcji ruchu i klasyfikacji obiektów wg odpowiednich, powszechnie dostępnych algorytmów opisanych w literaturze, np. [40].

W zaprojektowanej i wykonanej platformie sprzętowej do realizacji nadzoru wizyjnego wykorzystano (nie licząc seryjnie produkowanych układów scalonych) dwa gotowe moduły sprzętowe:

1. moduł sprzętowy TREX-S2 firmy Terasic zawierający układ FPGA, nieulotną pamięć konfiguracji oraz złącza, na których dostępne są praktycznie wszystkie wyprocedowania FPGA,
2. programowalny moduł komunikacji sieciowej EM1206 firmy Tibbo, na którym działają opracowane przez autora pracy aplikacje dedykowane dla platformy sprzętowej (są one odpowiedzialne za komunikację sieciową do warstwy transportowej).

Zastosowanie gotowego modułu z układem FPGA umożliwiło znacznie szybszą realizację prototypu platformy sprzętowej, ponieważ zwolniło z konieczności montażu układów w obudowach BGA (Ball Grid Array), co jest trudne do wykonania bez specjalizowanego sprzętu. Z kolei zastosowanie modułów sieciowych było podyktowane koniecznością zapewnienia działania podsystemu rekonfiguracji nawet wtedy, gdy układ FPGA nie zawiera poprawnej konfiguracji (podsystem rekonfiguracji byłby znacznie bardziej skomplikowany, gdyby zastosowano klasyczne rozwiązanie przy wprowadzonych założeniach). Zarówno moduł z układem FPGA jak i moduł komunikacji sieciowej mogą być zastąpione przez autorskie, dedykowane rozwiązania ¹.

2 Wyzwania i technologie w dziedzinie nadzoru wizyjnego

W pracy doktorskiej zawarto informacje na temat klasycznych oraz współczesnych rozwiązań z dziedziny nadzoru wizyjnego, a także podjęto próby zakreślenia tendencji w tym obszarze. Przedyskutowano wybrane zagadnienia związane elastycznością, skalowalnością oraz wymaganiami na przepustowość sieci dla różnych architektur systemów nadzoru wizyjnego. Szczególny nacisk położono na integrację systemów nadzoru wizyjnego z systemami informatycznymi zgodnymi z paradygmatem SOA. Tego typu podejście wymaga jednak zastosowania nowoczesnych technologii do implementacji poszczególnych elementów składowych systemów nadzoru wizyjnego. Dlatego przedstawiono, omówiono oraz oceniono przydatność wielu dostępnych rozwiązań w dziedzinie nadzoru wizyjnego oraz pokrewnych obszarach, takich jak przetwarzanie obrazu, czy komunikacja sieciowa.

¹Aktualnie autor prowadzi prace nad własnym, bardziej wydajnym modułem sprzętowym zastępującym EM1206

2.1 Zróźnicowanie implementacji systemów nadzoru wizyjnego

Założono, że docelowy, najbliższy optymalnemu system powinien:

- pozyskiwać z otoczenia informację na temat zdefiniowanych zdarzeń bądź anomalii, jak np. ruch w monitorowanej strefie, do której jest ograniczony dostęp,
- działać jako usługa sieciowa lub zbiór usług oferujących klientom rezultaty swoich obliczeń (np. wyników przetwarzania obrazu),
- móc autonomicznie sterować zbiorem układów wykonawczych w reakcji na zdefiniowane zdarzenia,
- być bez trudu i dużych kosztów rozbudowywany pod kątem ilości kamer.

W związku z tym poddano ocenie trzy typy architektur nadzoru wizyjnego:

1. klasyczne i obecnie najpopularniejsze systemy, które przesyłają analogowe sygnały wideo do centralnego punktu, gdzie sygnały te są przetwarzane na postać cyfrową i następnie zapisywane w pamięci masowej bądź przetwarzane,
2. scentralizowane systemy cyfrowe opierające swoje działanie na kamerach z interfejsami sieciowymi, transmitujące (najczęściej skompresowany) sygnał wideo do centralnej jednostki, a dalej działające podobnie jak systemy opisane w punkcie 1,
3. systemy opierające się o przetwarzanie danych wykonywane przed wysłaniem z kamer, w których każda “inteligentna” kamera może autonomicznie wykonywać operacje detekcji zdefiniowanych zdarzeń i wysyłać zewnątrz jedynie meta-informacje (np. tekstowe) opisujące wykryte zdarzenie (np. “wystąpił ruch na pewnych współrzędnych”).

Pod kątem wymienionych wcześniej wymagań, rozwiązanie z punktu 3 okazuje się najlepiej pasować do założonych scenariuszy głównie dlatego, że dane w tym przypadku są przetwarzane niezależnie w każdej kamerze i dodanie do systemu nowej kamery powoduje proporcjonalny przyrost mocy obliczeniowej całego systemu oraz tylko nieznacznie zwiększa obciążenie istniejącej infrastruktury sieciowej. Dodatkowo, małe ilości transmitowanych danych mogą być bez trudu zawarte w pakietach wysokopoziomowych protokołów (np. SOAP [12], *Simple Object Access Protocol*) wykorzystywanych przez usługi sieciowe.

2.2 SOA – architektura zorientowana na usługi

Architekturze SOA poświęcono osobny punkt pracy doktorskiej. Architektura ta jest jednym z elementów koniecznych do zbudowania elastycznego systemu nadzoru wizyjnego, który mógłby być łatwo integrowalny z innymi systemami informatycznymi klasy biznesowej. W pracy zastosowano pragmatyczne podejście do architektury SOA, przytaczając i omawiając jej osiem podstawowych zasad, pierwotnie zdefiniowanych przez T. Erla w [1]. Optymalnie byłoby, gdyby usługa sieciowa spełniała wszystkie wymienione zasady.

2.3 Układy logiki programowalnej

Technologia układów logiki programowalnej (PLD) jest jednym z istotnych elementów pozwalających realizować szybkie obliczenia w bardzo wydajny sposób. Obecnie najpowszechniej stosowanymi przedstawicielami technologii PLD są układy FPGA – *Field Programmable Gate Array* ². Cechują się one dużą ilością zasobów, co pozwala na realizację nawet bardzo złożonych algorytmów przetwarzania informacji w sposób równoległy. Praca z układami programowalnymi (implementacja ich docelowej funkcjonalności) jest znacznie bardziej pracochłonna niż opracowywanie np. programów dla systemów mikroprocesorowych. W związku z tym pojawiły się na rynku liczne narzędzia do tzw. wysokopoziomowej syntezy kodu (*high-level synthesis*) wspomagające pracę programisty (np. [7, 13, 14]). Jednak, podobnie jak w przypadku pierwszych kompilatorów języków programowania wysokiego poziomu, nie są to zawsze doskonałe narzędzia.

2.4 Przegląd dostępnych rozwiązań w dziedzinie nadzoru wizyjnego

Dostępne rozwiązania zostały omówione i ocenione pod kątem przydatności w realizacji systemów nadzoru wizyjnego. Autor dysertacji rozpatruje zarówno rozwiązania przydatne jako elementy pojedynczej “inteligentnej kamery” (np. interesujące realizacje algorytmów przetwarzania obrazów), jak i opracowane przez innych badaczy rozwiązania platform sprzętowych, a także koncepcje całych systemów nadzoru wideo. Szczególny nacisk położony został na rozwiązania z dziedziny systemów elastycznych i skalowalnych, choć omówiono także wiele innych, niemniej ciekawych rozwiązań z dziedziny systemów scentralizowanych. Przegląd dostępnych rozwiązań podzielono na kategorie omówione poniżej.

1. **Podsystemy przetwarzania obrazu oparte na układach FPGA.** W tej części omówiono zarówno rozwiązania bardziej ogólne, jak i specjalizowane dla zastosowań w dziedzinie nadzoru wizyjnego. Są to rozwiązania nieautonomiczne, czyli takie, które mogą stanowić pewien element (lub nawet rdzeń funkcjonalności) większego systemu, lecz bez odpowiedniego otoczenia zapewniającego i odbierającego dane nie są w stanie samodzielnie funkcjonować.
2. **Platformy sprzętowe i rozwiązania niskopoziomowe.** Ta część koncentruje się na niskopoziomowych rozwiązaniach, które, po odpowiedniej adaptacji, mogą mieć zastosowanie w platformach sprzętowych stanowiących bazę dla systemów nadzoru wizyjnego.
3. **Kompletne rozwiązania systemów nadzoru wizyjnego.** Omówiono i oceniono tutaj wiele rozwiązań całych architektur systemów nadzoru wizyjnego. Dostępne rozwiązania bazują na różnych mediach transmisji danych, różnych sposobach ograniczania ilości przesyłanych danych, czy wreszcie różnych organizacjach struktury tworzonej sieci. Oceny dokonano pod kątem skalowalności, elastyczności systemów oraz rozwiązań komunikacji sieciowej.

²W pracy doktorskiej omówiono także inne, również popularne układy programowalne CPLD (*Complex Programmable Logic Devices*), choć mają one mniejsze zastosowanie w realizacji złożonych obliczeń.

4. **Usługi sprzętowe realizowane w urządzeniach wbudowanych i na układach FPGA.** Jest to osobna kategoria rozwiązań dotyczących tworzenia serwerów usług sieciowych realizowanych w nietypowy sposób, bo przy wykorzystaniu mikrokontrolerów i układów FPGA zamiast tradycyjnej infrastruktury serwerowej. Z punktu widzenia autora projektującego platformę sprzętową dla inteligentnej kamery, jest to bardzo interesująca klasa rozwiązań, ponieważ kamera również posiada funkcjonalność serwera usługi sieciowej.

3 Zarys projektu sprzętowej usługi nadzoru wizyjnego

W pracy przedstawiono ogólne informacje na temat projektu sprzętowej usługi sieciowej realizującej funkcjonalność “inteligentnej kamery” do rozproszonych systemów nadzoru wizyjnego zgodnych z paradygmatem SOA. Autor kładzie nacisk na ogólne zagadnienia dotyczące konstrukcji urządzeń tej klasy, począwszy od założeń konstrukcyjnych, jakie powinno spełniać takie urządzenie, aż do prezentacji ogólnej koncepcji działania przykładowej platformy, która spełnia postawione wymagania.

Najpierw zdefiniowano założenia, które miała spełniać opracowywana platforma sprzętowa. Założenia podzielono na następujące kategorie wyszczególnione poniżej.

- **Ogólna charakterystyka platformy sprzętowej inteligentnej kamery.** Założenia dotyczące klasy projektowanej platformy jako urządzenia wbudowanego (ang. *embedded device*) opartego na układzie FPGA, z pasywnym chłodzeniem i o niewielkich wymiarach zewnętrznych,
- **Wymagania odnośnie możliwości przechowywania danych.** Zdefiniowano tutaj zapotrzebowanie kamery na różne rodzaje pamięci: operacyjną, masową, konfiguracji itd.
- **Wymagania energetyczne.** Postawiono tutaj wymagania, jaką moc powinna maksymalnie pobierać sprzętowa usługa sieciowa. Im mniejsza będzie pobierana moc (lepiej spełnione założenie), tym tańsza może być późniejsza eksploatacja takiej usługi oraz łatwiejsze zasilanie z odnawialnych źródeł energii.
- **Komunikacja sieciowa i jakość usług (QoS).** W tym punkcie wyszczególniono założenia dotyczące dostępności usługi (związane z jej niezawodnością) oraz czasu jej odpowiedzi na zapytania klientów. Dodatkowo przeanalizowano i teoretycznie wyizolowano czynniki mające największy wpływ na czas odpowiedzi, by w późniejszych etapach zminimalizować je.
- **Kontrola układów wykonawczych.** Jest to dodatkowa funkcjonalność, którą powinna mieć inteligentna kamera. Zdefiniowano zatem klasy urządzeń, które powinny być poprawnie sterowane z kamery (m.in. zamek do drzwi, syrena alarmowa, oświetlenie, silnik serwo z wbudowanym sterownikiem) oraz wymagania (głównie czasowe) interfejsu elektrycznego dla tych urządzeń.
- **Aktualizacja konfiguracji, naprawy i utrzymanie sprawności działania.** W tym punkcie omówiono założenia dotyczące zagadnień związanych z czynnościami

serwisowymi sprzętowej usługi sieciowej i związanymi ze zdarzeniami zaplanowanymi (np. aktualizacja oprogramowania) oraz niezaplanowanymi (np. awarie sprzętu). Spełnienie tych założeń ma na celu ułatwienie późniejszych czynności serwisowych.

Mimo, że opisane wcześniej założenia projektowe nie zmieniły się, to jednak sama platforma sprzętowa do realizacji usług nadzoru wizyjnego może być zaimplementowana na wiele sposobów. Jest ona efektem wieloletnich badań popartych weryfikacją praktyczną. W punkcie dotyczącym ewolucji platformy sprzętowej przedstawiono wybrane poprzednie wersje implementacji modułów sprzętowych oraz architektury wewnętrznej inteligentnej kamery. Mogą one być postrzegane jako interesujący wkład w ostateczną wersję inteligentnej kamery.

Następnie została w skrócie omówiona ogólna architektura oraz rozwiązania techniczne najistotniejszych elementów ostatecznej wersji platformy sprzętowej. Nieco bardziej szczegółowo omówiono aspekty związane z zagadnieniami komunikacji sieciowej, podsystemem zdalnej rekonfiguracji oraz najważniejszymi modułami przetwarzającymi i przechowującymi dane.

W pracy omówiono także alternatywne rozwiązania wybranych aspektów implementacji platformy sprzętowej do realizacji usług nadzoru wizyjnego. Rozważania dotyczą zastosowania mikrokontrolera ogólnego przeznaczenia o dużej mocy obliczeniowej zamiast układu FPGA, co aktualnie jest bardzo popularne głównie ze względu na większą łatwość implementacji algorytmów przy użyciu języków wysokiego poziomu. Takie rozwiązanie jest jednak znacznie mniej elastyczne jak chodzi o dostępność specjalizowanych układów peryferyjnych. Drugi aspekt rozważań dotyczy alternatywnej implementacji komunikacji sieciowej, opartej o układ warstwy fizycznej, nazywany PHY (od ang. *physical layer*) zamiast zastosowanych modułów komunikacyjnych EM1206 [10] firmy Tibbo. Rozwiązanie takie mogłoby podnieść wydajność komunikacji sieciowej z usługą, jednak znacznie by utrudniło implementację mechanizmów zdalnej rekonfiguracji przy wykorzystaniu tego samego fizycznego medium ze względu na konieczność multipleksacji pakietów w niskich warstwach modelu OSI (*Open Systems Interconnection*).

4 Szczegóły implementacji i osiągnięte rezultaty

Kolejną częścią pracy dotyczy szczegółów technicznych, z którymi musi się zmierzyć konstruktor opracowujący sprzętową usługę sieciową dla systemów nadzoru wizyjnego. Oprócz szczegółów implementacji, przy omawianiu poszczególnych elementów systemu uwzględniono także testy jakości oraz szybkości ich działania w praktycznej implementacji. W tym rozdziale autor koncentruje się osobno na działaniu poszczególnych modułów systemu, natomiast ewaluacja działania całości platformy sprzętowo-programowej została opisana w kolejnym rozdziale.

Rozdział rozpoczyna się od szczegółowego omówienia platformy sprzętowo-programowej oraz modyfikacji, które należy wprowadzić w konfiguracji układu FPGA w celu implementacji na niej konkretnych usług sieciowych mających zastosowanie w systemach nadzoru wizyjnego.

Pierwszą taką usługą jest system detekcji ruchu w obrazie rejestrowanym przez wbudowany w platformę sprzętową moduł kamery. Wymaga on jedynie niewielkich zmian

wewnętrznej architektury systemu w FPGA, innego programu sterującego dla mikrokontrolera sterującego Nios-II [8] oraz innego tzw. akceleratora sprzętowego, tj. dedykowanego modułu przyspieszającego obliczenia. W dalszej części rozdziału omówiono także najważniejsze informacje na temat implementacji samego akceleratora sprzętowego.

Drugą przykładową usługą opracowaną dla platformie sprzętowej jest klasyfikacja obiektów podobnych. Tę usługę zaimplementowano na dwa sposoby:

1. wykorzystując jeden *akcelerator gruboziarnisty*,
2. wykorzystując wiele *akceleratorów drobnoziarnistych*.

Akcelerator gruboziarnisty jest złożonym modułem sprzętowym wewnętrznym dla układu FPGA i odpowiedzialnym za wykonywanie prawie wszystkich najcięższych obliczeń algorytmu klasyfikacji obiektów – pozostałe moduły systemu tylko przygotowują dane dla tego akceleratora, by następnie odebrać od niego wyniki obliczeń i udostępnić je dla klientów usługi. Akceleratory drobnoziarniste są również modułami sprzętowymi implementowanymi w FPGA, lecz mają znacznie prostszą funkcjonalność, np. obliczanie funkcji arcus tangens lub pierwiastek kwadratowy z liczby naturalnej. Jedynie wspomagają one wykonywanie algorytmu klasyfikacji obiektów, a sam algorytm wykonywany jest głównie jako sekwencyjny kod mikrokontrolera Nios-II. Podejście z akceleratorem gruboziarnistym oferuje trochę lepsze wyniki przetwarzania danych niż podejście z akceleratorami drobnoziarnistymi, lecz wymagało od programisty znacznie więcej pracy przy implementacji oraz zużywa znacznie więcej zasobów układu FPGA (niektóre bloki funkcjonalne zostały wykorzystane w stu procentach, co nie pozostawia możliwości dalszej rozbudowy usługi sprzętowej bez zmiany układu FPGA na większy, więc także droższy).

W dalszej części rozdziału autor koncentruje się na implementacji komunikacji sieciowej przy użyciu dedykowanych modułów EM1206 firmy Tibbo. Przedstawiono także uzyskane szybkości działania dla dwóch alternatywnych wersji podsystemu komunikacji sieciowej:

- optymalizowanej pod kątem minimalizacji czasu odpowiedzi urządzenia przy małej liczbie klientów,
- optymalizowanej pod kątem obsługi zapytań od wielu klientów kosztem dłuższego czasu obsługi pojedynczego żądania.

Wyższe warstwy komunikacji sieciowej zostały opisane w punkcie dotyczącym logiki działania serwera usługi. Jest to moduł programowy służący w opracowanej platformie do zarządzania kolejką zapytań od klientów, wyzwania procesu przetwarzania danych oraz formatowania odpowiedzi dla klientów usługi.

W dalszej części rozdziału omówiono szczegóły implementacji innych, równie istotnych i ciekawych elementów opracowywanej platformy sprzętowej, tj. modułów odpowiedzialnych za wydajne pobieranie obrazu z kamery do pamięci operacyjnej, podsystemu zdalnej rekonfiguracji, modułów sterowania układami wykonawczymi oraz modułu generowania obrazu podglądu, mającego zastosowanie przy rozwijaniu nowych funkcjonalności platformy sprzętowo-programowej.

5 Weryfikacja działania systemu

Ważny rozdział pracy stanowi ocena działania sprzętowej usługi realizującej nadzór wizyjny jako całości. Zawiera on informacje na temat pełnej funkcjonalności oraz ogólnej szybkości działania mierzonej z punktu widzenia klienta dla dwóch przykładowych usług: detekcji ruchu oraz rozpoznawania obiektów.

Bardzo ważnym elementem pracy są rozważania na temat zgodności utworzonych usług z paradygmatem SOA. Temu zagadnieniu poświęcony jest osobny punkt, w którym sprawdzana jest zgodność z każdym założeniem zdefiniowanym przez T. Erla w [1]. Rozważania te można skonkludować, iż praktycznie wszystkie założenia są spełnione.

W kolejnej części pracy przedstawiono zastosowania opracowanej platformy sprzętowej lub innej usługi tworzonej zgodnie z metodologią przedstawioną w pracy doktorskiej. Jako przykłady zastosowania wymieniono:

- zaawansowane systemy dostępu oraz detekcji wtargnięcia do strzeżonego obszaru,
- system rozpoznawania typów pojazdów do automatycznego naliczania opłat i obliczania statystyk użytkowania dróg,
- system bezpieczeństwa na drodze monitorujący rodzaje nadjeżdżających pojazdów i kierujący je w odpowiednim kierunku zależnie od wykrytego typu pojazdu i warunków atmosferycznych,

6 Podsumowanie

W podsumowaniu pracy dokonano analizy stopnia zgodności z tezą postawioną na początku rozprawy:

Użycie rozwiązań sprzętowych zbudowanych z zastosowaniem układów programowalnych pozwala uzyskać serwisy nadzoru wizyjnego o podwyższonej ergonomii i zwiększonej łatwości integracji z systemami klasy enterprise przy zachowaniu aspektów ekonomicznych i wybranych parametrów QoS.

- Modułowa konstrukcja oraz możliwość zdalnej rekonfiguracji ułatwia utrzymanie działania oraz serwisowanie opracowanej usługi sprzętowej, co polepsza jej ergonomię.
- Zgodność z paradygmatem SOA pozwala twórcom konfiguracji i oprogramowania na kompozycję złożonych usług nadzoru wizyjnego z rozproszoną logiką obliczeniową. Tak utworzony system nadzoru wizyjnego można łatwiej integrować z innymi systemami informatycznymi klasy biznesowej (np. wspomniane automatyczne naliczanie opłat za przejazd).
- Pomimo wyższych (choć malejących z rozwojem technologii) kosztów pojedynczej kamery w porównaniu z klasycznymi systemami nadzoru wizyjnego, koszty fizycznej instalacji i utrzymania inteligentnej kamery są porównywalne z klasycznymi rozwiązaniami, a dodatkowo nie ma konieczności stosowania centralnego serwera gromadzącego dane wymagającego nieporównywalnie więcej energii elektrycznej.

- Możliwość potencjalnego wykorzystania już istniejącej infrastruktury sieciowej do celów nadzoru wizyjnego dodatkowo obniża koszt całego systemu.
- Czasy odpowiedzi opracowanych, przykładowych usług są na akceptowalnym poziomie oraz opracowane urządzenie działa stabilnie, co zapewnia utrzymanie odpowiednich parametrów QoS.

Dalej omówiono wkład w rozwój dziedziny ponownie cytując odpowiednie źródła z literatury podanej w pracy i konfrontując je ze wskazanymi odpowiednimi fragmentami dysertacji. Wreszcie, na koniec, przedstawiono możliwości i plany dalszej rozbudowy opracowanej platformy sprzętowej w celu przyspieszenia jej działania (lepsza komunikacja sieciowa) oraz podniesienia jej walorów funkcjonalnych (implementacja automatycznego ustawiania obszaru zainteresowania dla usługi klasyfikacji obiektów).

Literatura

- [1] T. Erl, SOA Principles of Service Design, Prentice Hall 2008
- [2] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall 2005
- [3] N. M. Josuttis, SOA in Practice, O'Reilly 2007
- [4] MAX 3000A Programmable Logic Device Family Data Sheet, <http://www.altera.com/literature/lit-m3k.jsp>, last access 2012.08.19.
- [5] CoolRunner-II CPLD Family Data Sheet, <http://www.xilinx.com/products/silicon-devices/cpld/coolrunner-ii/index.htm>, last access 2012.08.19.
- [6] Stratix II Device Handbook, <http://www.altera.com/literature/lit-stx2.jsp>, last access: 2012.08.19.
- [7] Impulse Accelerated Technologies, CoDeveloper Manual, Version 3.70, 2007 www.impulseaccelerated.com/ReleaseFiles/Help/iAppMan.pdf, [last accessed: July 25, 2012].
- [8] Nios II Processor Reference Handbook <http://www.altera.com/literature/lit-nio2.jsp> [last accessed: August 15, 2012]
- [9] Stratix II FPGA Prototyping System, <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=44&No=66>, [last accessed: March 2, 2011].
- [10] EM1206 BASIC-programmable Ethernet Module, <http://www.tibbo.com/products/modules/x20x/em1206.html>, [last accessed: March 2, 2011].
- [11] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language - W3C Recommendation 26 June 2007, <http://www.w3.org/TR/wsd120/>, [last accessed: March 2, 2011].

- [12] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) - W3C Recommendation 27 April 2007, <http://www.w3.org/TR/soap12-part1/>, [last accessed: March 2, 2011]
- [13] Altera Corp., SOPC Builder User Guide, http://www.altera.com/literature/ug/ug_sopc_builder.pdf [last accessed: 2013.01.13]
- [14] Altera Corp., About Qsys, http://quartushelp.altera.com/current/mergedProjects/system/qsys/qsys_about_qsys.htm [last accessed: 2013.01.13]
- [15] I. Kuon, J. Rose, Measuring the Gap Between FPGAs and ASICs, IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 26, no. 2, 2007, pp.203-215.
- [16] J. Canny, *A computational Approach to Edge Detection*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.8, no.6, 1986, pp.679-698
- [17] Altera DSP Builder Handbook, <http://www.altera.com/literature/technology/dsp/lit-dsp.jsp>, last access: 2012.08.19.
- [18] Altera Nios II Processor Documentation, <http://www.altera.com/literature/lit-nio2.jsp>, last access: 2012.08.19.
- [19] S. Hengstler, H. Aghajan, A. Goldsmith, Application-oriented Design of Smart Camera Networks, in: Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras, IEEE, Vienna, 2007, pp.12-19.
- [20] L. Longfei, Y. Songyu, Real-Time Duplex Digital Video Surveillance System and its Implementation with FPGA, in: Proceeding of the International Conference on ASIC, IEEE, Shanghai, 2001, pp.471-473.
- [21] C. Desmouliers, E. Oruklu, J. Saniie, FPGA-Based Design Of a High-Performance and Modular Video Processing Platform, in: Proceedings of the International Conference on Electro/Information Technology, IEEE, Windsor, ON, 2009, pp.393-398.
- [22] P. NirmalKumar, E. MuraliKrishnan, E. Gangadharan, Enhanced, Performance of H.264 using FPGA Coprocessors In video surveillance, in: Proceedings of the Internal Conference on Signal Acquisition and Processing, IEEE, Bangalore, 2010, pp.157-161.
- [23] G. Benet, J. E. Simo, G. Andreu-Garcia, J. Rosell, J. Sanchez, Embedded Low-Level Video Processing for Surveillance Purposes, in: Proceedings of the Conference on Human System Interactions (HSI), IEEE, Rzeszow, 2010, pp.779-786.
- [24] S. Cuenca-Asensi, H. Ramos-Morillo, H. Llorens-Martinez, F. Macia-Perez, Reconfigurable Architecture for Embedding Web Services, in: Proceedings of the Southern Conference on Programmable Logic, IEEE, San Carlos de Bariloche, 2008, pp.119-124.

- [25] A. Ben Atitallah, P. Kadionik, N. Masmoudi, H. Levi, FPGA implementation of a HW/SW platform for multimedia embedded systems, *Design Automation for Embedded Systems*, Springer, 2008, pp.293-311.
- [26] H. T. Ngo, R. N. Rakvic, R. P. Broussard, R. W. Ives, Rapid Prototyping of an Automated Video Surveillance System: A Hardware-Software Co-Desin Approach, *Mobile Multimedia/Image Processing, Security, and Applications*, Proceedings of the SPIE, Volume 8063, 2011.
- [27] P. Latha, M. A. Bhagyaveni, Reconfigurable FPGA Based Architecture for Surveillance Systems in WSN, in: *The international Conference on Proceeding of the Wireless Communication and Sensor Computing*, IEEE, Chennai, 2010, pp.1-6.
- [28] H. Jian-sheng, The Research into the Key Problem of Multi-Channel Video Acquisition and Display System Based on FPGA, *Advanced Materials Research*, vols. 433-440, 2012, pp.5611-5615.
- [29] X. Xue, J. XU, Y. Li, Z. Wan, The Design of 4-Channel Video Multiplexer Based on FPGA, in *proc. of The International Conference on Information Management and Engineering*, IEEE, Chengdu, 2010, pp.377-380.
- [30] A. Ben Atitallah, P. Kadionik, N. Masmoudi and H. Levi, FPGA implementation of a HW/SW platform for multimedia embedded systems, *Design Automation for Embedded Systems*, vol.12, no.4, 2008, pp.293-311.
- [31] L. N. Elkhatib, F. A. Hussin, P. Sebastian, Camera Sensor Network Localization Using FPGA, in *proc. of International Conference on Intelligent and Advanced Systems*, IEEE, Kuala Lumpur, 2010, pp.1-6.
- [32] J. Tang, X. Tian, A Distributed Video Monitoring Terminal Based on DaVinci DSP, in *proc. of International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*, IEEE, Hunan, 2012, pp.13-16.
- [33] A. Kandhalu, A. Rowe, R. Rajkumar, DSPcam: A Camera Sensor System for Surveillance Networks, in *proc. of International Conference on Distributed Smart Cameras*, IEEE, Como, 2009.
- [34] Z. Shelby, Embedded Web Services, *Wireless Communications*, IEEE, vol. 17, issue 6, pp.52-57.
- [35] C. Groba, S. Clarke, Web services on embedded systems - A performance study, in *proc. of International Conference on Pervasive Computing and Communications Workshops*, IEEE, Mannheim, 2010, pp.726-731.
- [36] D. Lioupis, M. Stefanidakis, A Web Service for Embedded Distributed Computation, in *proc. of the Euromicro Conference on Parallel, Distributed and Network-Based Processing*, IEEE, 2005, pp. 20-25.

- [37] G. B. Machado, F. Siqueira, R. Mittmann, C. A. V. Vieira, Embedded Systems Integration Using Web Services, in proc. of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, IEEE, 2006, pp.18-23.
- [38] C. E. Chang, F. Mohd-Yasin, A. K. Mustapha, An Implementation of Embedded RESTful Web Services, in proc. of the Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, IEEE, Malaysia, 2009, pp.45-50.
- [39] D. Pellerin, S. Thibault, Practical FPGA Programming in C, 2005, Prentice Hall.
- [40] A. Ruta, Y. Li, Learning Pairwise Image Similarities for Multi-classification Using Kernel Regression Trees, Pattern Recognition, 2012, Vol. 45, No. 4, pp. 1396-1408.
- [41] Jack E. Volder, The CORDIC Trigonometric Computing Technique,IRE Transactions on Electronic Computers, Vol. EC-8, pp.330-334, 1959.
- [42] OpenCores, VHDL Implementation of CORDIC, <http://opencores.org/project, cordic> [last access: 2013.01.16].