

**Maciej ŚLIWIŃSKI, Witold ŻYWIEC, Wojciech CIESIELKA, Andrzej GOŁAŚ**

AGH AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA,  
Al. Mickiewicza 30, 30-059 Kraków

## System wbudowany dedykowany dla „inteligentnego budynku” wykorzystujący układ FPGA

**Mgr inż. Maciej ŚLIWIŃSKI**

Absolwent Wydziału Inżynierii Mechanicznej i Robotyki Akademii Górniczo-Hutniczej. W 2013 roku ukończył studia na kierunku Mechatronika o specjalności Systemy Inteligentne. Obecnie jest studentem studiów doktoranckich w Katedrze Systemów Energetycznych i Urzędzeń Ochrony Środowiska. Interesuje się projektowaniem oraz programowaniem układów cyfrowych, szczególnie w kontekście systemów wbudowanych.



e-mail: maciejs@agh.edu.pl

**Mgr inż. Witold ŻYWIEC**

Absolwent Akademii Górniczo-Hutniczej w Krakowie. Ukończył Mechatronikę na Wydziale Inżynierii Mechanicznej i Robotyki w 2013r., obecnie słuchacz studiów doktoranckich w dyscyplinie Automatyki i Robotyki w Katedrze Systemów Energetycznych i Urzędzeń Ochrony Środowiska. Interesuje się zagadnieniami związanymi z przetwarzaniem sygnałów oraz projektowaniem systemów pomiarowych oraz monitoringu.



e-mail: wzywiec@agh.edu.pl

**Dr inż. Wojciech CIESIELKA**

Adiunkt w Katedrze Systemów Energetycznych i Urzędzeń Ochrony Środowiska na Wydziale Inżynierii Mechanicznej i Robotyki AGH. Autor ponad 80 publikacji naukowych. Opiekun i wykładowca na specjalności Systemy Inteligentne. Jego zainteresowania naukowe obejmują zagadnienia związane z systemami inteligentnymi, cyfrowym przetwarzaniem sygnałów, informatyką oraz ochroną środowiska i energetyką.



e-mail: ghciesie@cyf-kr.edu.pl

**Prof. dr hab. inż. Andrzej GOŁAŚ**

Profesor na Wydziale Inżynierii Mechanicznej i Robotyki AGH. W przeszłości Prorektor AGH i kierownik Katedry Systemów Energetycznych i urzędzeń Ochrony Środowiska. Autor ponad 120 publikacji naukowych. Członek Komitetu Polityki Naukowej. Jego szerokie zainteresowania naukowe obejmują m.in. takie dyscypliny jak systemy inteligentne, informatykę oraz ochronę środowiska i energetykę.



e-mail: ghgolas@cyf-kr.edu.pl

### Streszczenie

W artykule zaprezentowano system wbudowany dedykowany dla „inteligentnego budynku” wykorzystujący układ FPGA. W założeniach systemu skoncentrowano się na umieszczeniu w jednym układzie procesora programowalnego wspomagane dedykowanymi układami logicznymi. Zapewniono w ten sposób równoległą i niezależną pracę urzędzeń pomiarowych, wykonawczych oraz obsługujących komunikację bezprzewodową. Artykuł zawiera krytyczną analizę przedstawionego rozwiązania i porównanie z komercyjnie dostępnymi.

**Słowa kluczowe:** FPGA, układy reprogramowalne, SoPC, zdalne zarządzanie, akwizycja danych, automatyzacja budynku, inteligentny budynek.

### The embedded system as a smart building control unit using FPGA

#### Abstract

The purpose of this paper is to present System-on-a-Programmable-Chip approach to design of a smart building control unit. It focuses on observed and potential profits as well as problems possible and encountered during realization of such a system. The discussed project utilizes an FPGA device as the basis of the embedded system consisting of both memory and logic elements along with intellectual property (IP) processor core and custom hardware. The main emphasis of the system concept was to incorporate wireless communication and to free processor resources by moving more demanding tasks to dedicated logic structures. Modules implemented in Hardware Description Language were designed to provide parallel and independent control, communications, measurements and data acquisition. This paper presents the critical analysis of SoPC [7] control units for building management systems in comparison to other possible solutions [2-5] and accepted standards [1].

**Keywords:** FPGA, reprogrammable devices, smart building, remote control, data acquisition, SoPC.

### 1. Wprowadzenie

Rozwój technologiczny postępujący na przestrzeni ostatnich lat nierozdzielnie związany jest z wkraczaniem nowoczesnych rozwiązań do obszarów życia, w których wcześniej był on zupełnie

nieobecny. Dlatego przy coraz łatwiejszym dostępie do układów programowalnych i różnego rodzaju sterowników, powstają rozwiązania mające na celu poprawę komfortu oraz zwiększenia bezpieczeństwa w środowiskach, w których człowiek przebywa najczęściej, czyli w miejscu pracy i w domu.

Wraz z wprowadzaniem nowoczesnych technologii i systemów automatyzacji do budynków już w latach 70. ubiegłego stulecia, objekty te zaczęto nazywać nieco mylnie „inteligentnymi budynkami”. Również obecnie, termin ten jest nadużywany, ponieważ o inteligencji można mówić wyłącznie w stosunku do systemów, które byłyby w stanie same sobie stawiać problem i go rozwiązywać. Pomimo to, różne organizacje zajmujące się standaryzacją w tym zakresie, nazywają inteligentnymi te budynki, które poprzez integrację odpowiednich podsystemów znajdujących się w nim, pozwalają na efektywne zarządzanie jego zasobami jednocześnie minimalizując koszty eksploatacji [1].

Obecnie popularyzowany jest trend tworzenia systemów wbudowanych dla urzędzeń sterujących wszelkimi obiektami. Tak również jest w przypadku sterowników automatyzacji budynków, dlatego wdrażane są coraz nowsze układy dedykowane i programowalne, jednakże ciągle brakuje kompletnych rozwiązań wykorzystujących reprogramowalne układy FPGA, które idealnie nadają się do tych celów. Niniejszy artykuł przedstawia koncepcję i praktyczną realizację układu sterowania inteligentnym budynkiem zaimplementowaną wewnątrz struktury układu FPGA.

### 2. Istniejące realizacje głównych sterowników

Analizując systemy wykorzystywane w „inteligentnych budynkach” i klasyfikując je pod względem głównej jednostki możemy podzielić je na cztery podstawowe grupy: historycznie pierwsze komputery PC, sterowniki PLC, mikrokontrolery i najnowsze układy FPGA. Obecnie w dalszym ciągu spotyka się zwłaszcza w zastosowaniach komercyjnych systemy wykorzystujące komputery klasy PC z uwagi na ich duże moce obliczeniowe oraz łatwość tworzenia interfejsu użytkownika. Z czasem jednak zaczęto rezygnować z takich jednostek centralnych, na rzecz coraz tańszych i łatwo dostępnych układów mikroprocesorowych oraz układów programowalnych, zmniejszając w ten sposób rozmiary

sterownika i wykorzystywane zasoby. Był to pierwszy krok w stronę systemów wbudowanych dedykowanych do takich budynków, które charakteryzują się tym, że posiadają platformę sprzętową zaprojektowaną stricte do współpracy z urządzeniami zewnętrznymi danego systemu, oraz poprzez zaimplementowanie odpowiedniego programu sterującego, realizują wyłącznie zadania związane ze sterowaniem podsystemami całego obiektu. Najczęściej spotykanymi rozwiązaniami, szczególnie w obszarze amatorskich konstrukcji, są jednostki centralne wykorzystujące sterowniki PLC [2, 3], które z uwagi na przystosowanie do pracy w środowisku automatyki przemysłowej, wydają się być idealne do współpracy z elementami wykonawczymi oraz pomiarowymi. Jednakże mogą one stanowić problem w przypadku tworzenia atrakcyjnych interfejsów dla użytkowników systemu, oraz integracji z podzespołami komunikującymi się poprzez zaawansowane interfejsy komunikacyjne.

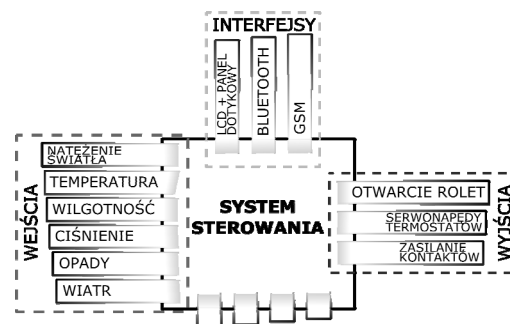
Alternatywą dla sterowników PLC stały się ostatnio platformy sprzętowe wykorzystujące tanie mikroprocesory. Przykładem może być coraz popularniejsza platforma Arduino [4, 5], za pomocą której można obecnie zbudować stosunkowo szybko i przy niskim nakładzie pracy, funkcjonalny system sterowania podstawowymi elementami automatyki domowej. Jednakże wraz ze wzrostem skomplikowania założeń projektowych czy ilości różnych podzespołów, z którymi musi komunikować się centralny sterownik, rośnie nieproporcjonalnie stopień złożoności programu sterującego. Wynika to z faktu, że mikroprocesory w takich układach wykonują instrukcje liniowo i nawet przy zastosowaniu technik pseudo-wielowątkowości, cały system działa w sposób szeregowy, gdzie poszczególne podzespoły są obsługiwane kolejno i jednostkowo w danej chwili czasowej. Stąd wymagana jest konieczność umiejętnego napisania głównego programu sterującego, aby nie wprowadzać do systemu niepotrzebnych opóźnień lub nie pominać ważnych komunikatów z zewnętrznych urządzeń.

Biorąc pod uwagę powyższy problem, projektanci systemów sterowania inteligentnymi budynkami, coraz chętniej wykorzystują w swoich rozwiązaniach układy reprogramowalne FPGA. Dają one możliwość implementacji skomplikowanych lub czasowo krytycznych operacji na tzw. sterowniki sprzętowe, które działają jak wyspecjalizowane układy ASIC. Co więcej, jeden układ FPGA może zajmować się równolegle kilkoma takimi wątkami, które na siebie nie wpływają i nie opóźniają się wzajemnie. I tak właśnie wykorzystywane są obecnie te układy, tzn. jako element wyłącznie wspierający pracę innych układów nadrzędnych, np. mikroprocesorów [6]. Brakuje realizacji kompletnych systemów, w których przy wykorzystaniu układów FPGA, dokonuje się w pełni równoległego przetwarzania informacji oraz komunikacji z elementami pomiarowymi i wykonawczymi. System wbudowany zaprezentowany w niniejszym artykule umożliwia zarządzanie pracą wszystkimi podsystemami inteligentnego budynku w sposób równoległy.

### 3. Struktura i funkcje systemu

Koncepcję systemu opracowano mając na uwadze możliwość jego adaptacji do zastanej infrastruktury budynku. Dlatego zdecydowano się na zastosowanie na szeroką skalę komunikacji radiowej umożliwiającej przesyłanie danych pomiarowych, poleceń oraz interakcję użytkownika. Prezentowane rozwiązanie posiada architekturę scentralizowaną, której sercem jest jednostka wykorzystująca układ FPGA wyposażona w szereg wejść i wyjść, a także ekran LCD z panelem dotykowym. System prowadzi stały monitoring warunków atmosferycznych panujący na zewnątrz budynku oraz wybranych parametrów przekładających się na komfort w pomieszczeniach. Dane te stanowią wejście dla algorytmu sterującego elementami wykonawczymi mającymi za zadanie utrzymanie tych parametrów na zadanych poziomach. Zebrane informacje prezentowane są na bieżąco za pomocą graficznego interfejsu użytkownika (GUI) umożliwiającego również zmianę ustawień oraz sterowanie wybranymi wyjściami. Schemat wejść i wyjść układu pokazano na rys. 1.

Stacja pogodowa jest jedynym elementem wykorzystującym połączenie przewodowe za pomocą interfejsu RS-232. Moduł odpowiedzialny za komunikację z nią odpytuje ją w zadanych odstępach czasu o temperaturę, ciśnienie, wiatr oraz opady. Dane o warunkach panujących w pomieszczeniach obejmują temperaturę, wilgotność i natężenie światła. Zmierzone wartości transmitowane są drogą radiową z sieci rozproszonych elementów pomiarowych. Poza interfejsem graficznym wyświetlanym na wbudowanym ekranie istnieje możliwość wydawania poleceń oraz otrzymywania raportów za pośrednictwem technologii Bluetooth i GSM.

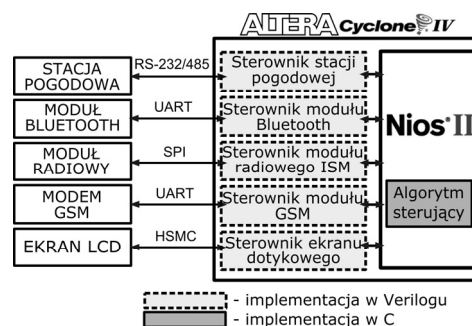


Rys. 1. Schemat wejść i wyjść systemu sterowania „inteligentnego budynku”  
Fig. 1. Inputs and outputs of the „smart building” system

System posiada szereg elementów wykonawczych, których kontrola również odbywa się drogą bezprzewodową. Sterownik modułu radiowego przystosowano do równoczesnej pracy z trzema protokołami transmisji oraz dwoma rodzajami modulacji sygnału obsługiwany przez jeden nadajnik. Dwa z nich służą do komunikacji z urządzeniami wykorzystującymi protokoły FS20 i FHT. Pozwalają one na sterowanie napędami rolet, stopniem otwarcia zaworów kaloryferów oraz włączanie i wyłączanie gniazdek sieciowych. Trzeci protokół stworzono na potrzeby transmisji danych pomiędzy jednostką centralną, a siecią modułów pomiarowych rozmieszczonych wewnątrz budynku.

### 4. Sterownik systemu w układzie FPGA

Do roli centralnego sterownika w zaprezentowanym powyżej układzie sterowania inteligentnym budynkiem, wykorzystano zestaw uruchomieniowy Terasic VEEK, który jest wyposażony w układ FPGA Cyclone IV firmy Altera. Zdecydowano się na to rozwiązanie z uwagi na dużą pojemność tego układu (liczoną w tzw. „elementach logicznych”) oraz liczne peryferia płyty głównej z 7-calowym ekranem LCD oraz panelem dotykowym na czele. Ideą dla tego projektu było opracowanie i wykonanie kompletnego sterownika systemu, który zawierałby całą część cyfrową wewnątrz jednego układu reprogramowalnego, czyli tzw. SoPC (ang. System on a Programmable Chip) [7]. Schemat blokowy zaproponowanego rozwiązania przedstawiono na rys. 2.



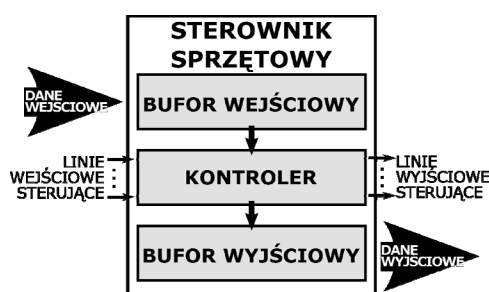
Rys. 2. Realizacja sterownika „inteligentnego budynku” wewnątrz układu FPGA  
Fig. 2. The „smart building” controller on the FPGA device

Struktura wewnętrzna sterownika składa się z 5 sterowników sprzętowych, napisanych w języku opisu sprzętowego Verilog [8], a także procesora programowego Nios II, który został zaprogramowany przy użyciu języka C. Zadaniem procesora jest integracja pracy wspomnianych sterowników sprzętowych oraz podejmowanie decyzji co do sterowania elementami wykonawczymi znajdującymi się wewnątrz budynku.

W postaci sterowników sprzętowych zaimplementowano moduły do obsługi:

- stacji pogodowej,
- modułu Bluetooth,
- modemu GSM,
- transceivera radiowego pasma ISM,
- ekranu LCD z panelem dotykowym.

Ich działanie polega na wymianie informacji z zewnętrznymi urządzeniami poprzez różnego rodzaju interfejsy komunikacyjne, buforowanie danych wejściowych, kodowanie lub dekodowanie poleceń oraz wstawianie do bufora wyjściowego danych wyników. Ogólny schemat blokowy każdego sterownika przedstawiono na rys. 3.



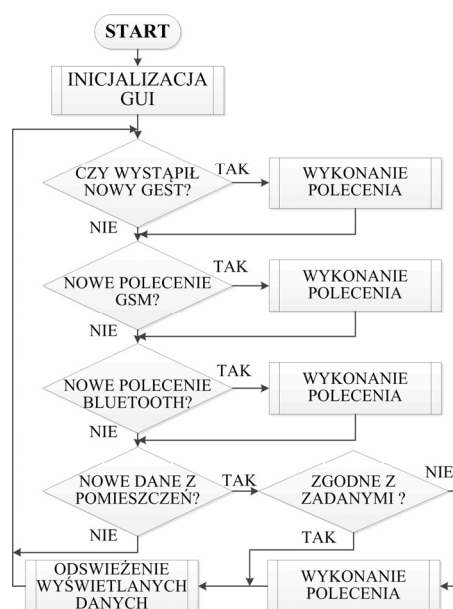
Rys. 3. Ogólny schemat blokowy sterowników sprzętowych  
Fig. 3. The overall block diagram of hardware drivers

Wynika z niego, że funkcjonalnie, wszystkie sterowniki są do siebie podobne, różnią się jednakże rodzajem standardu komunikacji, jaki wykorzystują przy wymianie danych z poszczególnymi urządzeniami, a także konkretnymi operacjami, jakie są wykonywane przez dany kontroler. Należy podkreślić, że zastosowanie oddzielnych sterowników dla każdego zewnętrznego elementu systemu pozwala na w pełni równoległą współpracę z każdym z nich. Oznacza to, że np. jeżeli dokładnie w tym samym czasie zajdzie konieczność odczytania nowej wiadomości SMS z modemu GSM oraz wysłania polecenia do sterownika rolet okiennych, to oba te zdarzenia będą wykonane w tym samym czasie i bez wzajemnego wpływu na siebie. Jest to możliwe, dzięki buforom sprzętowym, znajdującym się wewnątrz każdego sterownika sprzętowego, których zadaniem jest przetrzymywanie kolejnych danych, czekających na obsłużenie. W ten sposób, procesor Nios II odczytuje kolejne polecenia, które zostały wstępnie przetworzone i zdekodowane, w momencie, kiedy tylko jest to możliwe. Istotne jest to, że sam procesor, którego program jest wykonywany w sposób liniowy, nie wprowadza opóźnień do systemu, ponieważ operacje, które on wykonuje, zostały sprowadzone do minimum, dzięki kontrolerom poszczególnych sterowników. Służy on jedynie jako element decyzyjny w systemie, tzn. przekazuje polecenia sterujące do modułów, na podstawie danych wejściowych. Działanie to jest analogiczne przy komunikacji w drugą stronę (tj. od procesora do sterowników), ponieważ po stronie procesora nie są formułowane skomplikowane ramki wyjściowe, a jedynie krótkie zakodowane polecenia, które później interpretuje wybrany kontroler i dopiero on tworzy ramkę danych zgodną z danym standardem komunikacyjnym i dokumentacją urządzenia. Wszystkie dane wymieniane między sterownikami sprzętowymi a procesorem, są przekazywane przy użyciu równoległych linii danych, co również skraca czas obsługi pojedynczego zdarzenia. Dodatkowym atutem takiej architektury sterownika jest to, że cały system może działać bez przerwy, nawet w trakcie aktualizacji programu głównego procesora Nios II. Wszystkie

polecenia są przetwarzane i przetrzymywane przez sterowniki sprzętowe, niezależnie od działania samego procesora. Takie podejście jest dużym atutem, ponieważ raz zaprojektowany system w warstwie logicznej, może współpracować z dowolnym procesorem, którego rola sprowadza się jedynie do podejmowania decyzji co do sterowania. Możliwa jest nawet bezinwazyjna zmiana algorytmu sterującego w trakcie użytkowania systemu, jeżeli zajdzie taka potrzeba.

## 5. Algorytm sterujący

Dzięki przeniesieniu większości wymagających czynności do dedykowanych sterowników sprzętowych, główny algorytm sterujący pracą systemu mógł zostać znacznie uproszczony do postaci przedstawionej na rys. 4. Rolę programu zaimplementowanego w procesorze programowym ograniczono do cyklicznego odpytywania czujników i sprawdzania zawartości buforów poszczególnych modułów. W przypadku pojawienia się poleceń wydanych przez użytkownika następuje ich interpretacja i podejmowane są stosowne działania (np. przesłanie raportu bądź uruchomienie odpowiedniego aktuatora). Gdy bufor sterownika modułu radiowego informuje o oczekujących nowych wynikach pomiarów, zmierzone wartości przypisywane są do zmiennych przyporządkowanych poszczególnym pomieszczeniom identyfikowanym za pomocą unikatowych adresów.



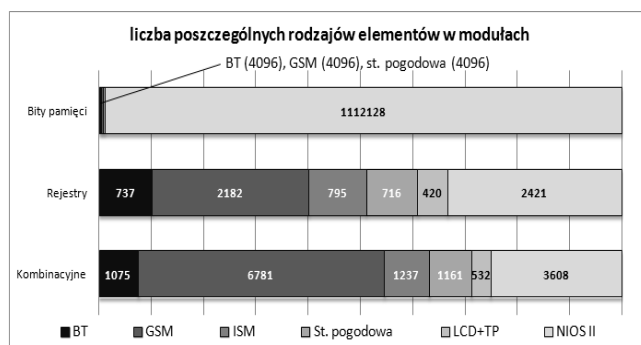
Rys. 4. Algorytm głównego programu sterującego  
Fig. 4. The main control algorithm

Algorytm porównuje dane o mikroklimacie otrzymane z sieci czujników z wartościami zadanymi przez użytkownika. Na tej podstawie, jeśli zaistnieje taka potrzeba oraz możliwość, decyduje o podjęciu czynności mających na celu osiągnięcie pożądanych warunków w pomieszczeniu. Dla przykładu, gdy wystąpi różnica temperatur pomiędzy zadaną a zmierzoną sterowanie ogrzewaniem pozwala na regulację stopnia otwarcia zaworów głowic termostatycznych w krokach 20% (od 0% oznaczającego całkowite zamknięcie do pełnego otwarcia przy 100%). W przypadku wolnozmiennych parametrów takich jak temperatura powietrza w pomieszczeniu jest to wystarczające rozwiązanie. Polecenia przesłane do rolet i gniazdek sieciowych pozwalają na odpowiednio zamykanie/otwieranie i włączanie/wyłączanie.

Najbardziej wymagającym obliczeniowo zadaniem procesora jest zmiana wyświetlanych treści GUI. Jednak dzięki zastosowaniu pamięci SRAM pośredniczącej w procesie odświeżania zawartości ekranu procesor obciążony jest wyłącznie gdy zachodzi potrzeba wprowadzenia zmian.

## 6. Testy

Projektowanie systemu równoległe na wielu poziomach abstrakcji wymagało przeprowadzenia szczegółowych testów na każdym etapie realizacji. Każdy sterownik sprawdzano indywidualnie w celu eliminacji błędów we współpracy z dedykowanymi urządzeniami. W następnym kroku dokonano integracji wszystkich elementów systemu wbudowanego i symulacji jego działania. Na tym etapie pojawiły się problemy związane z sumowaniem czasów propagacji sygnału między kolejnymi elementami logicznymi. Powstałe w ten sposób opóźnienia uniemożliwiały spełnienie wymogów odnośnie zależności czasowych układu przy wysokiej częstotliwości taktowania. Opóźnienia w dedykowanych układach nie dotyczyły wszystkich operacji. Problem ten napotkano w strukturach wykonujących złożone operacje o charakterze liniowym takich jak interpretacja długich ciągów znaków ASCII. Implementacja tych algorytmów w strukturze FPGA okazała się mało efektywna z uwagi na rozmiar zajętych zasobów i wydajność porównywalną do prostych procesorów.



Rys. 5. Zestawienie liczby wykorzystywanych elementów w modułach  
Fig. 5. Summary of the number of elements used in the modules

Na rys. 5 przedstawiono porównanie liczby elementów kombinacyjnych, rejestrów oraz bitów pamięci, wykorzystywanych przez poszczególne moduły w układzie FPGA. Procesor programowy Nios II zdecydowanie dominuje w kategorii wykorzystania bloków pamięci, z uwagi na rozmiar ustawionej pamięci podręcznej dla niego. Wartość ta została ustalona z dużym nadmiarem, aby zapewnić możliwość łatwej implementacji dużo bardziej skomplikowanych algorytmów sterowania w przyszłości. Pozostałe bity pamięci przypisano do modułów, które wykorzystują moduł UART i zajmują po równo 4096 bitów. W przypadku pozostałych kategorii, najobszerniejszym modułem jest kontroler modemu GSM. Używa on więcej elementów kombinacyjnych niż procesor Nios II i prawie tyle samo rejestrów, stąd wniosek, że niektóre funkcje lepiej zaimplementować w algorytmie sterowania procesora lub opracować bardziej uniwersalny sterownik sprzętowy. Cały system zajął stosunkowo niewielką część zasobów układu Cyclone IV. Struktura obejmuje 15% wszystkich dostępnych elementów logicznych i 28% komórek pamięci. Umożliwia to dalszą rozbudowę kontrolera, dodawanie kolejnych sterowników sprzętowych i rozszerzenie funkcjonalności systemu bez ryzyka wystąpienia problemów z rozmieszczeniem i połączeniem komponentów na powierzchni układu scalonego.

## 7. Podsumowanie

Przedstawiony projekt stanowi kompletne rozwiązanie systemu wbudowanego przeznaczonego do sterowania elementami należącymi do „Inteligentnego Budynku”. W roli centralnego sterownika wykorzystano układ reprogramowalny FPGA, który pozwolił na stworzenie w pełni równoległej platformy zarządzającej pracą

poszczególnymi urządzeniami wykonawczymi, pomiarowymi oraz komunikacyjnymi. Podejście to pozwoliło na bardziej elastyczne wykorzystanie gotowych podzespołów automatyki domowej, ponieważ nawet najbardziej złożone sposoby komunikacji, wymyślone przez producentów tych urządzeń, są skutecznie obsługiwane przez jednostkę centralną systemu. Jest to możliwe dzięki zastosowaniu sterowników sprzętowych zaimplementowanych wewnątrz struktury układu FPGA. Każdy z nich skupia się wyłącznie na obsłudze pojedynczego urządzenia zewnętrznego, przez co nie ma potrzeby tworzenia skomplikowanego algorytmu sterującego dla pojedynczego procesora. Co więcej, uzyskano w ten sposób również pewną niezależność pracy poszczególnych podzespołów, gdyż wymiana danych i skomplikowane obliczenia dla każdego z nich wykonywane są bez wzajemnego wpływu na siebie. Dodatkowym atutem jest zwiększenie efektywności systemu, rozumianej w sensie programistycznym, czyli braku wprowadzania opóźnień w całym systemie wynikających z czasu obsługi najwolniejszego ogniwa.

Pomimo licznych zalet jakie posiada przedstawiony system sterowania, przeprowadzone testy nakazują zwrócić uwagę na pewne kwestie zastosowanego rozwiązania. Mianowicie okazuje się, że pisanie bardzo złożonych sterowników w warstwie sprzętowej może być nieefektywne, ponieważ swoją złożonością zajmują duży obszar dostępnych zasobów układu FPGA, gdzie te same operacje mogłyby być wykonane równie szybko i skutecznie przez procesor programowy, przy użyciu np. odpowiednich bibliotek do obsługi ciągów znakowych. Ewentualnie w niektórych przypadkach należałoby zastosować bardziej uniwersalny rodzaj kontrolera, który swoją strukturą byłby zbliżony do bardzo prostego mikroprocesora.

Powyższe rozwiązanie jest dobrą podstawą do rozbudowania przedstawionej koncepcji tworzenia systemów wbudowanych dla automatyki budynków. Szczególnie przy analizie finansowej projektu, a także możliwości zaadaptowania sterownika do współpracy z już istniejącymi podzespołami, system ten staje się bardzo atrakcyjny dla odbiorców indywidualnych.

## 8. Literatura

- [1] Niezabitowska E.: Budynek inteligentny. Tom I. Potrzeby użytkownika, a standard budynku inteligentnego, Gliwice 2005.
- [2] Kwaśniewski J.: Inteligentny dom i inne systemy sterowania w 100 przykładach, BTC, 2011.
- [3] Yunwei Dong, Bo Zhang, Kang Dong: An Integrated PLC Smart Home System in Pervasive Computing. 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing.
- [4] Riley M.: Inteligentny dom. Automatyzacja mieszkania za pomocą platformy Arduino, systemu Android i zwykłego komputera, Helion, 2013.
- [5] Baraka K., Ghobril M., Malek S., Kanj R., Kayssi, A.: Low Cost Arduino/Android-Based Energy-Efficient Home Automation System with Smart Task Scheduling. 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks.
- [6] Stefanowicz Ł., Wiśniewski R., Węgrzyn M.: Zastosowanie układów reprogramowalnych FPGA w projekcie „Inteligentny Dom”. PAK, vol. 58, s. 614-616, 2012.
- [7] Hamble J. O., Hall T. S.: Using System-on-a-Programmable-Chip Technology to Design Embedded Systems. IJCA, Vol. 13, No. 3, Sept. 2006.
- [8] Hajduk Z.: Wprowadzenie do języka Verilog, BTC, 2009.

otrzymano / received: 17.01.2014

przyjęto do druku / accepted: 03.03.2014

artykuł recenzowany / revised paper