

PROJEKTOWANIE UKŁADÓW CYFROWYCH W JĘZYKU PROGRAMOWANIA SPRZĘTU VHDL - PREZENTACJA WITRYNY INTERNETOWEJ

Krystyna Maria NOGA¹, Patryk Seweryn MOTYKA²

¹Uniwersytet Morski Gdynia, Katedra Automatyki Okrętowej
tel.: 58 55 86 471, e-mail: k.noga@we.umg.edu.pl

²absolwent Wydziału Elektrycznego Akademii Morskiej w Gdyni

Streszczenie: W artykule przedstawiono nową witrynę internetową zrealizowaną w Katedrze Automatyki Okrętowej Akademii Morskiej w Gdyni. Na stronie zostały umieszczone informacje niezbędne do przygotowania własnych projektów układów cyfrowych opracowywanych w edytorze tekstowym i graficznym języka VHDL w środowisku Quartus. Strona zawiera przykłady rozwiązań począwszy od układów prostszych do bardziej zaawansowanych. W każdym z tych projektów przedstawiono przykładowe rozwiązania. Ponadto do każdego projektu, opracowanego przy użyciu obu wymienionych edytorów, zostały dołączone komentarze ułatwiające zrozumienie rozpatrywanych zagadnień.

Słowa kluczowe: technika cyfrowa, dydaktyka, układy programowalne, język VHDL, Quartus, edytor tekstowy, edytor graficzny.

1. WSTĘP

Obserwowany od dłuższego czasu szybki rozwój technologii spowodował, że nowe urządzenia, aplikacje oraz coraz doskonalsze oprogramowanie w istotnym stopniu zmieniły zasady pracy w zakładzie pracy, domu, a także sposób i metody przekazywania wiedzy oraz podejście do wypoczynku. Nakłada to obowiązek, przede wszystkim na szkoły średnie i wyższe, przekazywania wiedzy uczniom i studentom w odpowiedni sposób, zachęcania do poszukiwań nowych rozwiązań technologicznych, nauczania sprawnego myślenia, a także zapewnienia wysokiej jakości kształcenia. Realizacja tych celów nie jest łatwa. Współcześnie studenci mają sporo obowiązków i zainteresowań, część z nich pracuje zawodowo, nie zawsze mają czas na właściwe wypełnienie swoich obowiązków. Często przyczyną braku czasu jest nadmiar nowinek. Studenci potrafią wykorzystać nowe technologie, ale z inicjowaniem nowych pomysłów i wykonywaniem zadań przekraczających poziom standardowy jest już znacząco gorzej. Również sposób korzystania z wiedzy przekazywanej na wykładach czy ćwiczeniach w ostatnim okresie uległ sporej zmianie. Studenci dosyć szybko czują się znużeni, nawet wówczas, gdy prowadzący zajęcia korzysta z nowatorskich rozwiązań technologicznych. Materiały edukacyjne dostępne przez Internet powinny pomagać studentom w przyswajaniu i uzupełnianiu wiedzy. Pozwalają one na dowolne dostosowanie czasu i tempa nauki. Dobrze zaprojektowana strona powinna zapewnić łatwy dostęp do informacji oraz szybką nawigację w sieci, co znacznie przyspiesza proces kształcenia. Dodatkowo materiały dostępne przez Internet są dobrym sposobem reklamowania uczelni.

Jedną z dziedzin nauki i techniki rozwijającą się od dłuższego czasu w szybkim tempie jest technika cyfrowa, a szczególnie technologia i aplikacje z wykorzystaniem układów programowalnych *PLD* (ang. Programmable Logic Device). W Internecie dostępnych jest sporo materiałów dotyczących teorii, zasad projektowania oraz przykładów zastosowania układów programowalnych. Materiały te charakteryzują się różnym stopniem złożoności i uporządkowania, jednak nie zawsze przedstawiają wybrane zagadnienie w zadawalający sposób. Z analizy dostępnych w Internecie materiałów wynika, że nie jest dostępna strona, która wyjaśnia sposób przygotowania algorytmu cyfrowego sterowania zaczynając od podstaw a kończąc na przedstawieniu gotowego projektu przygotowanego z wykorzystaniem języka programowania sprzętu *VHDL* (ang. Very High Speed Integrated Circuits Hardware Description Language), zarówno przy użyciu edytora tekstowego, jak i graficznego. Propozycją uzupełnienia tego braku jest strona internetowa opracowana w Katedrze Automatyki Okrętowej (KAO) Akademii Morskiej w Gdyni w ramach inżynierskiej pracy dyplomowej [1].

W artykule przedstawiono nową witrynę internetową, na której zostały umieszczone informacje niezbędne do przygotowania własnych projektów układów cyfrowych opracowanych w edytorze tekstowym i graficznym języka *VHDL* w środowisku programowania *Quartus*. Strona zawiera przykłady rozwiązań dla około 20 zagadnień począwszy od najprostszych, takich jak realizacja sumatora, wykorzystanie sumatora do mnożenia, układ wybranego translatora i projekt licznika sekwencyjnego pracującego zgodnie z określonym grafem. Do układów bardziej zaawansowanych można zaliczyć sterowanie pracą plotera, układ wykrywający określoną kombinację bitów wejściowych oraz układ sterowania windą ładunkową. Każdy projekt zawiera opis przykładowego rozwiązania. Ponadto do każdego projektu opracowanego przy użyciu obu edytorów zostały dołączone komentarze ułatwiające zrozumienie rozpatrywanych zagadnień. Na stronie internetowej zostały umieszczone pliki źródłowe wszystkich projektów, co umożliwi uruchomienie ich na własnym komputerze.

2. PRZEGLĄD STRON INTERNETOWYCH Z ZAKRESU TECHNIKI CYFROWEJ

W dalszej części artykułu krótko scharakteryzowano wybrane strony internetowe, które prezentują zagadnienia zwią-

zane z projektowaniem układów cyfrowych z wykorzystaniem języka *VHDL*. Opis zawiera przedstawienie ogólnego zarysu treści strony, sposobu korzystania z niej oraz dodatkowych elementów. Wybrane strony przedstawiają nieco inne podejście do kursów internetowych niż prezentowana witryna opracowana w Katedrze.

Strona Uniwersytetu w Pensylwanii (rys. 1), dostępna pod adresem https://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html [2], ma dość skromną oprawę graficzną i skupia się jedynie na przekazie treści, dzięki czemu jest bardzo czytelna. Dzięki małej ilości grafiki szybsze jest także działanie strony, dzięki czemu, dysponując wolnym łączem, można skrócić czas transmisji, a także ilość przesyłanych danych. Witryna zawiera opis języka *VHDL*, przy czym zasady projektowania w tym języku są przedstawione w sposób bardzo szczegółowy. Stronę można wykorzystać jako kurs umożliwiający zapoznanie się z budową programów oraz ogólną strukturą języka.

VHDL Tutorial
1. Introduction
2. Levels of representation and abstraction
3. Basic Structure of a VHDL file
Behavioral model
Concurrency
Structural description
4. Lexical Elements of VHDL
5. Data Objects: Signals, Variables and Constants
Constant
Variable
Signal
6. Data types

Rys. 1. Strona na Uniwersytecie w Pensylwanii [2]

Kolejna strona <http://roboblog.eu> [3] zawiera ciekawostki, nowości, projekty ze świata technologii i kursy programowania. Jeden z kursów zawiera poradnik dotyczący układów *PLD*. Zawarte w nim porady są pomocne przy budowie rzeczywistych układów z zastosowaniem języka *VHDL*. Informacje te zawierają treści podstawowe i polegają na prezentacji sprzętu niezbędnego do realizacji konkretnych projektów. Są mniej przydatne do nauki i zrozumienia zasad projektowania układów cyfrowych. Autorzy korzystają z oprogramowania oraz układów programowalnych firmy *Altera*, oferują także sprzedaż przygotowanych pakietów startowych. Strona jest słabo czytelna, a znalezienie konkretnej informacji wymaga czasu, gdyż kurs jest podzielony na kilka dość długich części.

Inna wybrana strona <http://www.fpga.agh.edu.pl/tc/> [4] została stworzona w Akademii Górniczo-Hutniczej (AGH). Strona ta zawiera materiały pomocnicze oraz zadania przygotowane na potrzeby laboratorium Techniki Cyfrowej. Na stronie znajdują się również przykładowe zadania z rozwiązaniami. Sposób przedstawienia tych zadań jest podobny do zadań umieszczonych na stronie internetowej Katedry Automatyki Okrętowej.

Kolejną witryną przedstawiającą kody opisu sprzętu w języku *VHDL* oraz symulacje dla wielu układów cyfrowych firmy *Xilinx* jest strona <http://esd.cs.ucr.edu/labs/tutorial> stworzona przez Weijun Zhang [5]. Jednak zawarte w niej treści są dosyć ubogie, gdyż nie zawierają wielu podstawowych wiadomości z zakresu Techniki Cyfrowej.

Ostatnią analizowaną stroną jest witryna znajdująca się pod adresem <https://forbot.pl/blog/> [6], która jest nadal aktualizowana i rozbudowywana. Autorzy skupili się nie tylko na projektowaniu, ale również na implementacji programów dla rzeczywistych układów. Do uruchomienia algorytmów wykorzystano zestaw Elbert v2 – Spartan 3A, (wyposażony jest on w układ *PLD* Spartan 1C3S50A firmy *Xilinx* z maksy-

malnie 108 wyprowadzeniami I/O), który w sierpniu 2018 r. można było kupić za 179,00 zł. Kursy dostępne na stronie dotyczą tego zestawu i zawierają informacje dotyczące podstaw programowania w języku *VHDL* oraz sporo praktycznych rozwiązań. Przykładowo dostępne są informacje dotyczące: podstaw *VHDL* w praktyce, podstawowych pojęć, porównania z mikrokontrolerami, zasad instalacji środowiska *ISE Xilinx*, bramek logicznych, multiplekserów, rejestrów przesuwanych, przygotowania większego projektu, edytora graficznego, a także symulacji działania układu oraz podsumowanie kursu. Warto zaznaczyć, że strona zawiera nie tylko kursy, które dotyczą zestawu Elbert v2, ale także zestawów Arduino, Raspberry Pi, podstaw elektroniki, techniki cyfrowej oraz budowy robotów i zasad lutowania.

3. OPIS OPROGRAMOWANIA WYKORZYSTANEGO DO BUDOWY STRONY INTERNETOWEJ

Opracowana w Katedrze strona została zbudowana za pomocą programu *WebSite X5 Evolution 13* firmy Income-dia [7]. Natomiast do obróbki zrzutów ekranów został wykorzystany program *Paint*, a do wykonania oprawy graficznej środowisko *GIMP* (ang. GNU Image Manipulation Program) [8]. *WebSite* pozwala na szybkie stworzenie szablonu strony oraz ułatwia dostęp do wszystkich treści oraz zarządzanie stronami. Program ten jest prosty w obsłudze, jednak nie zawiera zaawansowanych funkcji i nie ma opcji dodawania własnego kodu skryptowego, do czego niezbędny jest oddzielny program. Program jest płatny, dostępny na witrynie <http://www.websitex5.com> [7]. Najtańszy pakiet w sierpniu 2018 r. kosztował 79 zł, zawierał 50 modyfikowalnych wzorców, natomiast wersja *Evolution* kosztowała 259 zł i zawierała 500 wzorców. Program *WebSite* w trakcie tworzenia wzorca graficznego pozwala na skorzystanie z już wcześniej przygotowanych szablonów, które można dowolnie modyfikować. Przykładowo możliwa jest zmiana typu menu, zmiana tła poszczególnych elementów wzorca oraz marginesów i wyrównania. Zmiany te można uzyskać za pomocą kaskadowych arkuszy stylu *CSS* (ang. Cascading Style Sheets), przy czym nie jest wymagana znajomość konfiguracji. W bardzo łatwy sposób możliwa jest również zmiana stylu menu głównego. Odbywa się to za pomocą poleceń dostępnych w zakładce *Styl* menu głównego. Program pozwala również na umieszczenie własnego obrazu tła dla przycisków, co znacznie ułatwia projektowanie menu. W programie *WebSite* znajduje się również mapa witryny, co znacznie przyspiesza proces modyfikacji. Ponadto dostępnych jest kilka opcji, dzięki którym w łatwy sposób można dodać nową podstronę. Mimo że dostępne szablony najczęściej nie są najlepszej jakości, to jednak możliwość modyfikacji usprawnia ich tworzenie.

Program *GIMP* został wykorzystany do bardziej zaawansowanej obróbki graficznej oprawy witryny. Pozwolił on na zastosowanie filtrów i warstw, które były niezbędne do uzyskania określonych efektów, np. do stworzenia przycisków i zastosowania przezroczystości. Program jest bezpłatny i dostępny na witrynie <http://www.gimp.org> [8].

4. APLIKACJA WWW

Zadaniem opracowanej w KAO strony internetowej było przedstawienie praktycznych przykładów zastosowania wiedzy teoretycznej z zakresu projektowania i analizy układów cyfrowych. Zostały przedstawione projekty w formie zagadnień - zadań wraz z ich rozwiązaniami, które przygo-

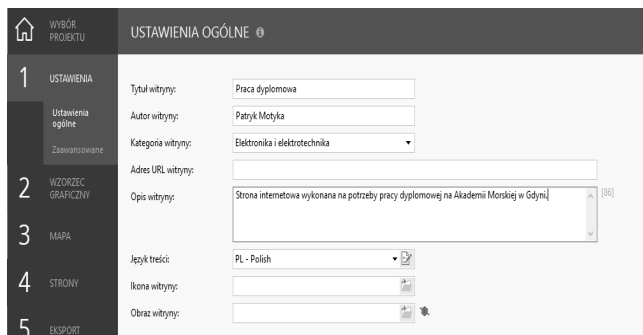
towano w edytorze graficznym i tekstowym środowiska *Quartus*. Strona zawiera również przykładowe przebiegi czasowe, uzyskane w przypadkach użycia edytora graficznego i tekstowego, które umożliwiają sprawdzenie poprawności działania zaprojektowanych układów. Zadania zostały podzielone na następujące grupy tematyczne [9–14]:

- układy kombinacyjne:
 - bloki arytmetyczne: sumatory, subtraktory, komparatory,
 - układy komutacyjne: multiplexery, demultiplexery, transkodery,
- układy sekwencyjne:
 - liczniki synchroniczne, asynchroniczne,
 - rejestry,
 - liczniki scalone.

Opracowana witryna ma przejrzystą szatę graficzną, łatwy też jest dostęp do każdego zadania. Prosta grafika umożliwiła wyróżnienie określonej treści za pomocą półprzezroczystego tła oraz intuicyjnego pionowego i poziomego menu. Informacje zawarte w zadaniach są zwięzłe i zrozumiałe zarówno dla osób rozpoczynających naukę, jak i dysponujących podstawową wiedzą teoretyczną. Na stronie zostały udostępnione wszystkie pliki źródłowe, które można pobrać na własny dysk i uruchomić w środowisku *Quartus II* w wersji 13.

4.1. Realizacja aplikacji WWW

Szablon zbudowanej strony został w całości wykonany za pomocą programu *WebSite*. Umożliwia on łatwe wykonanie witryny oraz jej dalsze zmiany. Ustawienia ogólne (rys. 2) umożliwiły umieszczenie informacji dotyczącej autora, tytułu witryny oraz opisu strony.



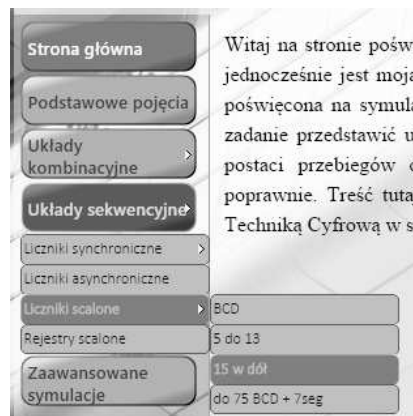
Rys. 2. Ustawienia ogólne witryny

Witrynę do Internetu przekazano za pomocą programu *WebSite*, przy czym możliwe jest również jej bezpośrednie zgranie na dysk. Strona główna witryny została przedstawiona na rysunku 3. Została ona umieszczona na serwerze Akademii Morskiej w Gdyni i jest dostępna pod adresem http://atol.am.gdynia.pl/tc/dyplom_vhdl [15].



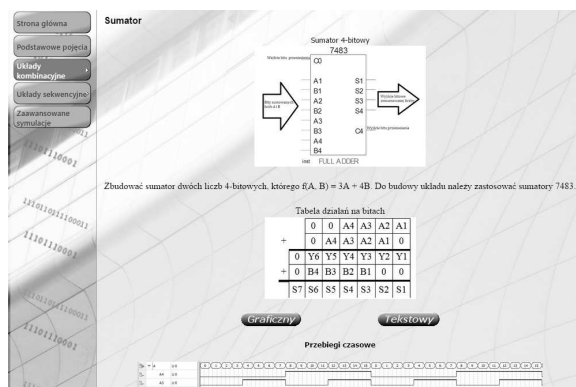
Rys. 3. Strona główna witryny

Przykładowe menu opracowanej witryny zostało przedstawione na rysunku 4. Menu poziome w znaczny sposób ułatwia kategoryzację i wyszukiwanie określonej informacji, zmniejsza również rozmiar menu i czas potrzebny do otwarcia kolejnej strony. Ostatni poziom menu zawiera szczegółowe nazwy dostępnych zadań.

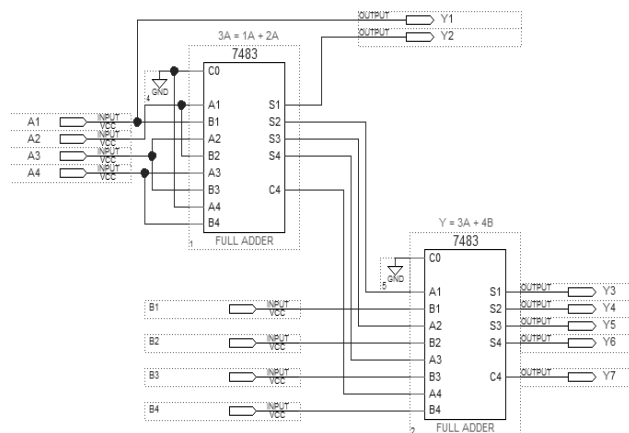


Rys. 4. Przedstawienie poziomów menu

Za pomocą przycisków *Graficzny* i *Tekstowy* możliwe jest zapoznanie się ze schematem logicznym zaprojektowanego układu (zrzut ekranu) dla trybu graficznego, a dla trybu tekstowego możliwe jest uzyskanie programu źródłowego wraz z licznymi komentarzami. Możliwe jest również pobranie plików źródłowych. Przykładowe zadanie dla układu realizującego funkcję $S(A, B) = 3A + 4B$ z wykorzystaniem 4-bitowego sumatora, gdzie A i B są symbolami 4-bitowych wejść, przedstawiono na rysunku 5, natomiast na rysunku 6 znajduje się schemat logiczny zaprojektowanego układu.



Rys. 5. Zadanie dla układu realizującego funkcję $S = 3A + 4B$

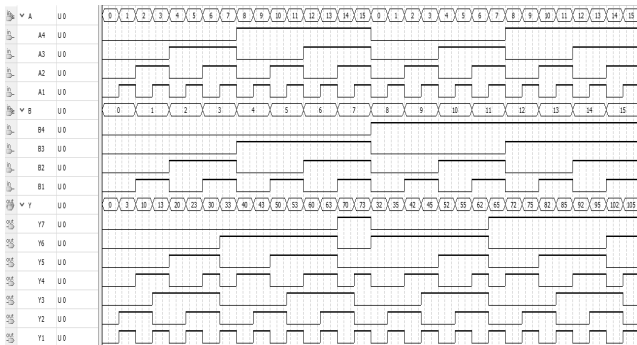


Rys. 6. Schemat układu realizującego funkcję $S = 3A + 4B$

4.2. Wybrane zagadnienia z techniki cyfrowej, przykłady symulacji

Omawiana strona zawiera zarówno przykłady do zaprojektowania, dla których nie jest wymagana szczegółowa wiedza z zakresu techniki cyfrowej oraz zadania bardziej zaawansowane. Zagadnienia podstawowe zostały tak dobrane, aby ułatwiły zrozumienie układów zaawansowanych.

W ramach zagadnień dotyczących układów kombinacyjnych na omawianej stronie internetowej przedstawiono między innymi przykłady opisujące sposób realizacji różnych operacji arytmetycznych z wykorzystaniem sumatora dwóch liczb 4-bitowych, jakim jest dla przykładu opisany wcześniej układ realizujący funkcję $S(A, B) = 3A + 4B$. Uzyskane przebiegi czasowe dla tego układu opracowanego przy użyciu edytora graficznego przedstawiono na rysunku 7. Oczywiście przebiegi czasowe dla układu opracowanego przy użyciu edytora tekstowego są takie same, co potwierdza poprawność projektów.



Rys. 7. Przebiegi dla układu realizującego funkcję $S = 3A + 4B$

Na stronie przedstawiono także działanie układu sumatora/subtraktora dwóch liczb 4-bitowych A i B , przy czym dla wejścia sterującego $C0 = 0$ układ realizuje sumę liczb $A+B$, natomiast dla $C0 = 1$ – ich różnicę $A-B$. Zadanie umożliwia także zapoznanie się z zasadami zapisu liczb w kodach: znak-moduł, uzupełnieniowym do 1 oraz uzupełnieniowym do 2. Strona internetowa zawiera również opis scalonego komparatora, który ma wiele zastosowań nie tylko w układach porównujących liczby dwójkowe.

Opisując bloki komutacyjne, na stronie omówiono zasady budowy transkodera kodu Aikena na BCD z wyprowadzeniami do wyświetlacza 7-segmentowego. Do budowy układu wykorzystano demultiplexer o 4 wejściach adresowych (wyjścia są aktywne w logice dodatniej), bramki logiczne oraz układ dekodera kodu BCD na kod wskaźnika 7-segmentowego. Na opracowanej stronie przedstawiono także budowę układu wykorzystującego multiplexery i realizującego funkcję przedstawioną przy pomocy podanego wykresu (w układzie współrzędnych X i Y).

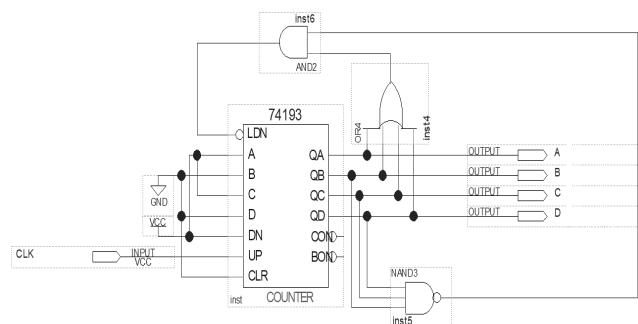
W dziale dotyczącym układów sekwencyjnych przedstawiono projekt licznika synchronicznego modulo 10 pracującego w kodzie BCD, który został zbudowany z przerzutników JK oraz bramek. Projekt przeznaczony do edycji graficznej zawiera między innymi tabelę stanów, wzbudzeń, tablice Karnauga ułatwiające minimalizację funkcji, schemat logiczny i uzyskane przebiegi. Stworzony za pomocą edytora tekstowego projekt tego samego licznika został podzielony na 2 programy. Program $JK10$ został wykorzystany w programie głównym dla celów symulacji działania przerzutnika JK. Dzięki zastosowaniu instrukcji *port map* możliwe było przypisanie wejść programu głównego do komponentu $JK10$, co przyspieszyło programowanie oraz

zwiększyło przejrzystość kodu [12]. W tej części witryny internetowej został także zaprezentowany projekt licznika synchronicznego, pracującego według grafu $2 \rightarrow 4 \rightarrow 9 \rightarrow 12 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 2$, do budowy którego wykorzystano przerzutniki D oraz bramki logiczne, przy czym stany układu zakodowano binarnie. W tym projekcie szczególną uwagę zwrócono na konieczność ustawienia określonego stanu początkowego układu różnego od zera, o czym studenci niestety często zapominają. Projekt tego samego licznika został także opracowany przy użyciu edytora tekstowego, w sposób odmienny niż zrobiono to dla wcześniej wspomnianego licznika modulo 10. W tym przypadku wykorzystano instrukcję *case*, której użycie dodatkowo zwiększyło czytelność programu. Deklaracja zmiennej *Stan* z przypisaniem wartości początkowej $0010_2 = 2_{10}$ zabezpiecza układ przed pojawieniem się stanu $0000_2 = 0_{10}$, który w grafie nie występuje.

Na stronie omówiono także zasady budowy algorytmu sterowania pracą plotera, którego zadaniem w omawianym projekcie jest wykreślenie sześciangu, przy czym dokładniejszy opis budowy plotera i zasad jego sterowania są dostępne na stronie [14]. W edycji graficznej do budowy układu sterowania wykorzystano demultiplexer z zanegowanymi wyjściami, licznik scalony oraz bramki logiczne. Użycie licznika pozwoliło na cykliczną zmianę wejść adresowych demultiplexera, co skutkuje wykonaniem kolejnych ruchów przez pisak plotera. Inne przykłady rozwiązań sterowania pracą plotera można znaleźć na stronie [14], przykładowo z wykorzystaniem multiplexerów i bramek, samych multiplexerów oraz demultiplexerów i bramek. Do budowy algorytmu sterowania przy użyciu edytora tekstowego wykorzystano, jak w poprzednio omawianym projekcie, instrukcję *case*. W tym przypadku kolejne kroki pracy pisaka odpowiadają również stanowi licznika.

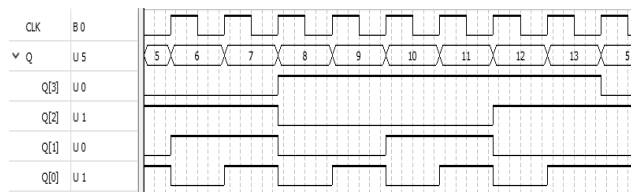
Na witrynie opracowanej w KAO można znaleźć także projekt licznika modulo 10 pracującego w kodzie BCD, do budowy którego wykorzystano licznik scalony oraz bramki. Korzystając z edytora tekstowego możliwe było zastosowanie inkrementacji, która wymagała dołączenia do projektu biblioteki *std_logic_unsigned.all*. Zastosowanie tej biblioteki znacznie zmniejszyło rozmiary programu oraz jego złożoność.

Do budowy kolejnego projektu, tj. licznika pracującego w górę (w systemie dodawania) zgodnie z grafem $5 \rightarrow 6 \rightarrow \dots \rightarrow 13 \rightarrow 5$ w kodzie binarnym, w edycji graficznej wykorzystano licznik scalony oraz bramki (rys. 8). W układzie użyto wejścia *LDN (LOAD)*, które umożliwia przepisanie stanów $0101_2 = 5_{10}$ z wejść buforowych na wyjścia układu licznika w przypadku, gdy na wyjściu pojawi się stan początkowy po włączeniu zasilania, czyli 0000_2 lub $1110_2 = 14_{10}$. Uzyskane przebiegi (rys. 9), oczywiście takie same, gdy stosuje się obydwie wymienione edytory, potwierdzają prawidłowość projektów.



Rys. 8. Schemat układu licznika od 5 do 13

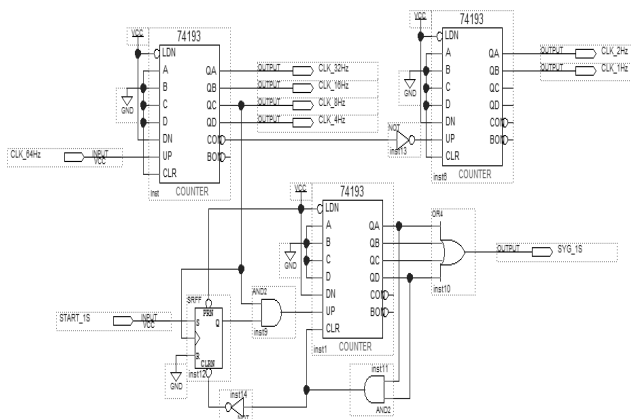
Na omawianej stronie dostępne są ponadto projekty licznika zliczającego w dół, modulo 75 w kodzie BCD oraz licznika zliczającego od 0 do 300 w kodzie BCD z dodatkowymi wejściami *Start*, *Sto* i *Zeruj*. Inne przykłady projektów liczników opracowane jedynie przy użyciu edytora tekstowego zostały także przedstawione na stronie [14].



Rys. 9. Przebiegi licznika sygnalizującego stany od 5 do 13

Właściwości rejestrów omówiono na stronie na przykładzie projektu 4-ro bitowego rejestru przesuwonego z szeregowym wejściem i równoległym wyjściem wykorzystującego przerzutniki typu D. Bardziej szczegółowy opis rejestrów przesuwanych znajduje się w [12, 13]. W tej części opracowanej witryny można również zapoznać się z budową rejestru, który ma możliwość przesuwania informacji w prawo, lewo oraz zerowania zawartości rejestru. Użycie edytora graficznego umożliwia wykorzystanie programu bibliotecznego układu scalonego, natomiast edytor tekstowy pozwala na użycie instrukcji *shift_left* lub *shift_right*, które służą do przesuwania zawartości rejestru odpowiednio w lewo lub prawo.

Zaawansowane projekty, przedstawione na zrealizowanej w KAO witrynie, wykorzystują algorytmy sterowania z zastosowaniem większej liczby układów, do których budowy niezbędna jest większa wiedza z zakresu techniki cyfrowej i zasad projektowania w języku *VHDL*. Na stronie zaprezentowano projekt dzielnika częstotliwości dla generatora fali prostokątnej o częstotliwości 64 Hz. Układ ma dodatkowo możliwość wygenerowania pojedynczego 1-sekundowego sygnału przy użyciu wejścia *START_1S*. Do budowy układu wykorzystano liczniki scalone, bramki logiczne oraz przerzutnik SR. W układach cyfrowych prostym sposobem budowy układów czasowych jest zastosowanie liczników do odmierzenia czasu (rys. 10) (w środowisku *Quartus* brak jest oprogramowania dla czasomierzy, przerzutników astabilnych i monostabilnych). Sygnał zegara *CLK_64Hz* jest podawany na wejście zegarowe licznika, a wówczas przykładowo na wyjściu *CLK_32Hz* uzyskuje się dwukrotnie mniejszą częstotliwość. Przerzutnik RS pozwala na wygenerowanie tylko jednego impulsu 1-sekundowego, po pojawieniu się stanu wysokiego na wejściu *START_1S*.



Rys. 10. Schemat logiczny układu dzielnika częstotliwości

Kolejnym bardziej zaawansowanym projektem jest układ do gry w kostkę dla dwóch osób, który po rzucie porównuje liczbę oczek obydwu graczy, przy czym gracz z większą liczbą oczek wygrywa rundę. Gra po 3 wygranych wyłania zwycięzcę oraz blokuje dalszą rozgrywkę, aż do momentu wyzerowania wyniku. Zaprojektowanie gry wymagało wygenerowania liczb losowych, w tym celu wykorzystano liczniki modulo 6 zliczające impulsy z dużą częstotliwością. Liczniki dla gracza pierwszego i drugiego działają z różną częstotliwością zegara. Zliczanie impulsów wstrzymuje się w chwili rzucenia kostką, po czym następuje porównanie liczby oczek obydwu graczy za pomocą komparatora. Dla każdego gracza został stworzony oddzielny układ zliczający liczbę wygranych rund. Należy zaznaczyć, że projekt gry w kostkę opracowany przy użyciu edytora tekstowego pozwolił na większą swobodę w projektowaniu.

Kolejny projekt omówiony na stronie umożliwia wykrucie określonej kombinacji (hasła) składającego się z 3 liczb 4 bitowych. Praktycznie podobne zagadnienie występuje podczas budowy układu dekodującego stosowanego w sejfach. Zaprojektowany układ ma możliwość zmiany hasła oraz wyłączenia blokady przy użyciu panelu serwisowego. Trzykrotne niepowodzenie podczas wprowadzenia hasła skutkuje zablokowaniem układu.

Na opisywanej stronie został przedstawiony również układ sterowania pracą windy ładunkowej, z której symulatorem studenci kierunku Elektrotechnika Okrętowa mają możliwość zapoznania się w czasie zajęć laboratoryjnych z techniki cyfrowej [13, 14]. Zaprojektowany układ ma zabezpieczenia termiczne uzwojeń silnika dla pierwszego i drugiego biegu, natomiast dla biegu trzeciego zabezpieczenie nadprądowe oraz termiczne z przełączeniem na bieg drugi przy zadziałaniu zabezpieczenia. Zrealizowano sekwencyjne przełączanie biegów, a styczniki kierunkowe działają symetrycznie zarówno w przypadku opuszczania, jak i podnoszenia windy. Układ wykonuje również funkcję *Stop* z hamowaniem silnika, przy czym inne styczniki są w tym czasie odłączone. Zabezpieczenie nadprądowe jest uruchamiane po określonej zwłoce czasowej. Ponieważ środowisko *Quartus* nie zawiera układów czasowych, o czym już wspomniano wcześniej, dlatego układ zaprojektowany przy użyciu edytora graficznego nieco różni się od układu sterowania wykonanego w technologii TTL [13]. Dla tego samego algorytmu sterowania opracowanego z wykorzystaniem edytora tekstowego dane lokalne (sygnały) bezpośrednio wskazują na ich przeznaczenie. Program został rozpisany na procesy, co znacznie przyspieszyło działanie układu, przy czym w przypadku, gdy nie stosowano procesów występowały stany nieustalone.

5. PODSUMOWANIE

Internet umożliwił dostęp do dużej liczby kursów z różnych dziedzin, jednak nie zawsze są to kursy, które w dostępny sposób wyjaśniają zasadę działania układów oraz przedstawiają określone rozwiązania. Wiele stron dotyczących techniki cyfrowej zawiera sporo zbytecznej teoretycznej wiedzy, która może zniechęcić do dalszej lektury z powodu nadmiaru informacji. Natomiast strony zawierające opis rzeczywistych układów zazwyczaj zawierają niewiele wiadomości merytorycznych, gdyż głównie przedstawiają określony produkt, do zakupu którego zachęcają czytelnika strony.

Głównym celem przygotowanej witryny było przedstawienie przykładów zastosowania wiedzy teoretycznej z

zakresu techniki cyfrowej do projektowania i analizy układów cyfrowych w środowisku *Quartus*, zarówno z wykorzystaniem edytora tekstowego, jak i graficznego. Wszystkie projekty zawierają liczne komentarze ułatwiające zrozumienie omawianego zagadnienia oraz przykładowe przebiegi, które potwierdzają prawidłowość rozwiązania. Każdy projekt zawiera szczegółowy opis rozwiązania. Przedstawione przykładowe rozwiązania inspirować do użycia innych możliwości zależnie od wiedzy i pomysłowości programisty. Witryna umożliwia praktyczne poznanie układów programowalnych. Zgodnie z opinią przedstawioną na stronie [6] układy te w wielu zastosowaniach sprawdzają się znacznie lepiej od mikrokontrolerów.

Strona miała zawierać elementy interaktywne w postaci możliwości bezpośredniego uruchomienia określonego pliku opracowanego i dostępnego na witrynie. Okazało się jednak, że na komputerze użytkownika nie jest to możliwe, gdyż otwarcie plików opracowanych w środowisku *Quartus* wymaga podania całej ścieżki dostępu do uruchamianego projektu. Dlatego też na opracowanej stronie internetowej zostały udostępnione wszystkie pliki źródłowe, które można pobrać i uruchomić na własnym komputerze w środowisku *Quartus II* w wersji 13.

Już pobieżna analiza układów zaprojektowanych przy użyciu obu wymienionych edytorów wymusza konieczność oszacowania czasu pracy niezbędnego do przygotowania projektu. Z całą pewnością czas opracowania algorytmu dla edytora tekstowego jest krótszy, ale pod warunkiem, że znane są ogólne zasady programowania w określonym środowisku oraz zasady budowy programu w języku *VHDL*. Jednocześnie bez znajomości budowy i działania wielu układów, np. liczników, przerzutników, bramek, multiplexerów, demultiplexerów, sumatorów, komparatorów, implementacja algorytmu sterowania za pomocą edytora tekstowego może być bardzo utrudniona, a może nawet niemożliwa. Dlatego też najpierw należy poznać działanie różnych układów używając środowiska graficznego. Zaletą edytora graficznego jest ułatwienie zrozumienia działania układów,

łatwiejsza jest także korekta błędów, a projektowanie nie wymaga znajomości składni języka *VHDL*, z czym początkujący projektant może mieć problem. Natomiast z całą pewnością zastosowanie edytora tekstowego do projektowania znacznie upraszcza to działanie i pozwala na zastosowanie programów bibliotecznych, co rozszerza możliwości programowania oraz znacznie zmniejsza ilość kodu wynikowego. Z tego powodu edytor tekstowy jest zalecany do stosowania w bardziej rozbudowanych projektach.

6. BIBLIOGRAFIA

1. Motyka P. S.: Technika Cyfrowa w środowisku Quartus (tryb tekstowy i graficzny) – interaktywna strona internetowa, inżynierska praca dyplomowa, AM, 2017
2. www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
3. <http://www.roboblog.eu>
4. <http://www.fpga.agh.edu.pl/tc/>
5. <http://esd.cs.ucr.edu/labs/tutorial/>
6. <https://forbot.pl/blog/>
7. www.websitex5.com
8. <http://www.gimp.org>
9. Noga K. M., Radwański M.: Multisim Technika Cyfrowa w przykładach, BTC, Wydanie I, Legionowo 2009
10. Tertulien Ndjountche, Digital Electronics 3 Finite State Machines, ISTE Ltd and John Wiley & Sons, Inc., Wielka Brytania 2016
11. Kalisz J.: Kurs Języka VHDL, Wojskowa Akademia Techniczna, Warszawa 2008
12. Zwoliński M.: Projektowanie układów cyfrowych z wykorzystaniem języka VHDL, Wydawnictwo Komunikacji i Łączności, Warszawa 2002
13. Noga K. M.: Laboratorium podstaw techniki cyfrowej, Akademia Morska w Gdyni, wydanie czwarte, 2005
14. <http://atol.am.gdynia.pl/tc>
15. http://atol.am.gdynia.pl/tc/dyplom_vhdl

DESIGNING OF DIGITAL SYSTEMS IN THE VHDL HARDWARE DESCRIPTION LANGUAGE - PRESENTATION OF THE WEBSITE

The article presents a new website, realized at Department of Ship Automation in Gdynia Maritime University, based on an engineering diploma thesis. The website contains information necessary to prepare own digital circuit designs developed in the text and graphic editor of the VHDL hardware description language in the Quartus environment. There are examples of solutions from simpler to more advanced ones on the website, e.g. an adder implementation, a design of a sequential counter working in accordance with a specific graph, control plotters work, a system detecting a specific combination of input bits, a control system of a cargo lift, and so on. Each project contains a description of an example solution. Furthermore, for each project prepared in both editors, a commentary was added to facilitate understanding of the issues under consideration. The source files of all projects have been placed on the website, which allows students to be run on their own computers.

Keywords: digital technique, didactics, programmable logic devices, language VHDL, Quartus, text editor, graphic editor.