Krystyna Maria NOGA

Uniwersytet Morski w Gdyni, Wydział Elektryczny, Katedra Automatyki Okrętowej

doi:10.15199/48.2019.12.62

Nauczanie Techniki Cyfrowej na kierunku Elektrotechnika w Uniwersytecie Morskim w Gdyni

Streszczenie. *W artykule przedstawiono zagadnienia omawiane w ramach zajęć z Techniki Cyfrowej na kierunku Elektrotechnika w Uniwersytecie Morskim w Gdyni. Nauczanie techniki cyfrowej polega na przedstawieniu podstawowych pojęć, poznaniu sposobów opisu i zasad projektowania układów. Podczas wykładu i ćwiczeń studenci poznają podstawy teoretyczne, analizują pracę różnych układów cyfrowych, projektują własne układy, przygotowują algorytmy cyfrowego sterowania. Natomiast w laboratorium w Katedrze Automatyki Okrętowej istnieje możliwość praktycznego zaprojektowania cyfrowych układów kombinacyjnych oraz sekwencyjnych o różnej skali trudności i zaawansowania.*

Abstract*. The article presents issues discussed in the course of classes in Digital Technology at the faculty of Electrical Engineering at the Gdynia Maritime University. Teaching digital technology involves presenting basic concepts, understanding the ways of describing and designing circuits. Students also learn about arithmetic and commutative systems, counters, registers and time dependencies circuits. During the lecture students learn theoretical basics, while in the classroom they design specific digital circuits. In the laboratory at the Department of Ship's Automation it is possible to get to know practical digital combinatorial and sequential circuits of varying difficulty and advancement levels. (**Teaching Digital Technique in the faculty of Electrical Engineering at the Gdynia Maritime University).*

Słowa kluczowe: technika cyfrowa, układy cyfrowe, układy programowalne, nauczanie, dydaktyka, symulacje, język programowania sprzętu VHDL.
Keywords: digital technology, digital circuits, programmable devices, teaching, didactics, simulations, VHDL programming language of equipment.

Wstęp

Dynamicznie zmieniające się kierunki rozwoju nauk technicznych, a szczególnie elektroniki, informatyki, techniki cyfrowej zmuszają do dyskusji nad programami nauczania na poziomie akademickim, metodami i technikami nauczania. Wymienione dziedziny nauki są powiązane ze sobą, występują w zaawansowanych rozwiązaniach technologicznych. Również rozwój Internetu spowodował zmiany w sposobie nauczania przede wszystkim na studiach technicznych. Studenci poprzez sieć mają dostęp do wielu pomocy dydaktycznych. Obecnie w wielu dziedzinach pracy inżynierskiej, przede wszystkim w zakresie projektowania, coraz większą rolę odgrywają badania symulacyjne. Dzisiaj na rynku dostępnych jest sporo programów umożliwiających symulację układów z dziedziny elektroniki, elektrotechniki oraz techniki cyfrowej. Jednym z częściej wykorzystywanych do prac badawczych, projektowych oraz dydaktyki jest środowisko SPICE oraz Multisim [1, 2, 3]. Programy te posiadają bogate biblioteki gotowych elementów oraz narzędzi symulacyjnych. Zagadnienia związane z dydaktyką w zakresie techniki cyfrowej są tematem wielu prac [1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Do prezentacji zagadnień i realizacji ćwiczeń są wykorzystywane różne platformy edukacyjne, np. DE2, DE2_115 firmy Terasic [6, 11, 12, 13]. Wiadomości z Techniki Cyfrowej (TC) są również przydatne podczas programowania sterowników PLC, które są wykorzystywane przede wszystkim jako elementy automatyki przemysłowej [14]. Nauczanie zasad logiki rozmytej także wymaga znajomości podstaw z TC, przykłady zagadnień omawianych w Kettering University w USA przedstawiono między innymi w pracy [15].

W artykule przedstawiono zagadnienia omawiane w ramach zajęć z TC na kierunku Elektrotechnika w Uniwersytecie Morskim w Gdyni (UMG). Program nauczania obejmuje przedstawienie podstawowych pojęć, np. symbole i tabele stanów bramek, przerzutników, oraz zasad minimalizacji funkcji logicznych. Studenci poznają układy arytmetyczne, komutacyjne, liczniki, rejestry, zasady projektowania układów cyfrowych. Podczas wykładu studenci poznają podstawy teoretyczne. Natomiast na zajęciach ćwiczeniowych projektują określone układy cyfrowe, najczęściej na tzw. papierze lub wirtualnie jako zadania kontrolne. W laboratorium z przedmiotu TC w Katedrze Automatyki Okrętowej (KAO) istnieje możliwość praktycznego poznania cyfrowych układów kombinacyjnych oraz sekwencyjnych o różnej skali zaawansowania. Ze względu na specyfikę uczelni studenci mają także możliwość zapoznania się z układami pracującymi na statkach, np. z symulatorem windy cumowniczej, przygotowują także algorytmy sterowania pracą np. dźwigu, silników krokowych. Student weryfikuje wiedzę z wykładu i ćwiczeń, poznaje metody projektowania i testowania układów cyfrowych. Zajęcia laboratoryjne są istotnym elementem w kształceniu na poziomie technicznym. Umożliwiają one przyszłym inżynierom przygotowanie i praktyczne sprawdzenie różnych projektów. Istotna jest także możliwość poznania własnych możliwości motorycznych.

Specjalnie na potrzeby dydaktyki kursu TC w laboratorium została stworzona sieć lokalna LAN oparta na Microsoft Network i podłączona do sieci Internet, studencki serwer plików sFTP oraz witryna internetowa dostępna pod adresem atol.am.gdynia.pl\tc [16].

Wykład i ćwiczenia

Na Wydziale Elektrycznym UMG na kierunku Elektrotechnika zajęcia z TC dla studiów dziennych realizowane są w następującym wymiarze: na pierwszym roku studiów inżynierskich 30 godzin wykład, 15 godzin ćwiczenia tablicowe, na drugim roku 30 godzin zajęć laboratoryjnych. Natomiast na drugim roku magisterskich studiów dziennych zajęcia obejmują 15 godzin wykładu i 15 godzin laboratorium, przy czym zajęcia dotyczą zasad wykorzystania języka programowania sprzętu VHDL. Na studiach niestacjonarnych zagadnienia z zakresu TC są omawiane na drugim roku w czasie 15 godzin wykładu, 8 godzin ćwiczeń i 15 laboratorium. Na studiach magisterskich niestacjonarnych studenci powinni uczestniczyć w 8 godzinach wykładowych i 10 laboratoryjnych.

Na studiach inżynierskich wykład obejmuje następujące zagadnienia:

elementy teorii układów cyfrowych, podstawowe układy, symbole,

analiza i synteza układów kombinacyjnych, sekwencyjnych,

- scalone bloki funkcjonalne – bloki arytmetyczne, komutacyjne, liczniki, rejestry, pamięci,
- układy uzależnień czasowych,
- podstawowe charakterystyki układów cyfrowych, wykrywanie i eliminacja hazardów.

Studenci przed wykładem otrzymują od prowadzącego zajęcia materiały dydaktyczne zawierające podstawowe wiadomości teoretyczne, schematy omawianych układów oraz linki do zbudowanych na potrzeby przedmiotu TC wirtualnych układów. Układy te zostały przygotowane w środowisku Multisim, są one omówione w pracy [3] i dostępne na stronie internetowej [16]. Ze względu na ograniczenia czasowe, nie zawsze na wykładzie występuje możliwość prezentacji działającego wirtualnego układu. Taka forma współpracy jest przez większość studentów oceniana pozytywnie. W trakcie semestru każdy student powinien oddać pracę kontrolną, w wersji papierowej i elektronicznej (poprzez umieszczenia jej na serwerze sFTP). Przykładowo należy zaprojektować układ realizujący określoną funkcję, przy czym należy go zrealizować w kilku wersjach, np. z wykorzystaniem tylko multipleksera i bramek logicznych lub multipleksera, demultipleksera i bramek. Sprawdzenie tych prac zajmuje niestety sporo czasu, czasami jest to szukanie źle zapisanych zer lub jedynek logicznych. Jednak, zdaniem autorki artykułu, czas poświęcony przynosi efekty. Studenci, którzy są aktywni i na bieżąco przygotowują się do zajęć raczej nie mają kłopotów z zaliczeniem wykładów. Niestety, w ostatnim okresie takie podejście studentów do zdobywania wiedzy nie jest standardem.

Na ćwiczeniach tablicowych studenci wykorzystują wiadomości teoretyczne, analizują i projektują określone układy. Również w przypadku ćwiczeń student powinien zaliczyć pracę kontrolną. Przykładowo należy zaprojektować rewersyjny licznik synchroniczny pracujący według określonego grafu, a do budowy należy wykorzystać poznane przerzutniki, multipleksery, demultipleksery i bramki logiczne. Jest to zadanie wymagające nieco więcej czasu niż w przypadku zadania kontrolnego z wykładu. Dlatego też na serwerze są dostępne tzw. szablony zawierające stałą część schematu przygotowanego w środowisku Multisim, który zawiera między innymi przerzutniki, bramki, wyświetlacze. Większość studentów oddaje prace w terminie, poprawia je i uzupełnia również w terminie. Jednak jak zwykle zdarzają się „maruderzy”, i to oni czasami zakłócają proces nauczania.

Na studiach magisterskich wykład obejmuje następujące zagadnienia:
- podstawowe pojęcia z zakresu układów programowalnych PLD, krótka historia rozwoju,
- podstawowe architektury, budowa, parametry układów CPLD, FPGA,
- układy firmy Altera, Xilinx, Lattice Semiconductor, sposoby konfiguracji układów PLD,
- edytor tekstowy języka programowania sprzętu VHDL, struktura programu, podstawowe zasady projektowania, instrukcje, przykłady rozwiązań, środowisko Quartus.

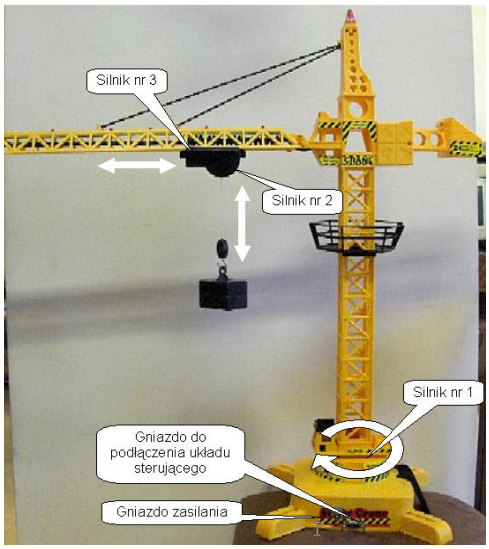
Prowadzenie zajęć na studiach magisterskich sprawia studentom i nauczycielom zdecydowanie mniej problemów. Studenci są bardziej dojrzali, odbyli już praktyki, bardzo często na statkach, więc wiedzą czego oczekuje „życie”. Są bardziej zaangażowani w zdobywanie wiedzy, w czasie zajęć laboratoryjnych chętniej angażują się w realizację bardziej ambitnych projektów.

Laboratoria

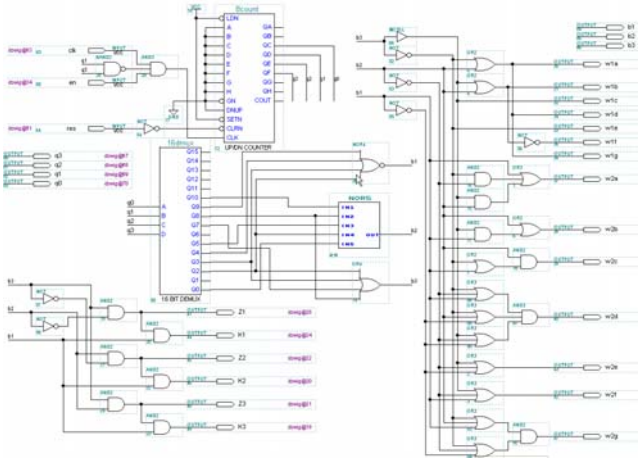
W laboratorium z przedmiotu Technika Cyfrowa na studiach inżynierskich istnieje możliwość sprawdzenia cyfrowych układów kombinacyjnych oraz sekwencyjnych, podstawowych, złożonych lub bardziej zaawansowanych. Na zajęciach jest wykorzystywany pakiet Multisim, Max Plus Baseline lub Quartus firmy Altera (obecnie Intel). Studenci projektują i budują nie tylko układy wirtualne. Również implementują swoje projekty w strukturach układów programowalnych CPLD oraz FPGA z wykorzystaniem edytora graficznego języka programowania sprzętu VHDL. W laboratorium dostępne są liczne modele obiektów sterowania, np. model koparki, sztaplarki, dźwigu, windy cumowniczej, robota-człowieka ROBOWISDOM, domu mieszkalnego, przenośnika taśmowego, modele różnych aut [2, 8, 9, 10, 11, 12, 13, 17, 18, 19, 20]. Na stronie internetowej [16] dostępnych jest wiele wzorcowych przykładów, projektów sterowań wybranymi modelami. Na podstawie dostępnych materiałów, zgodnie z instrukcją laboratoryjną, student powinien przygotować własny projekt, własny algorytm sterowania określonym modelem. Studenci na studiach inżynierskich wykonują ćwiczenia laboratoryjne obejmujące:
- badanie charakterystyk bramek logicznych i synteza układów kombinacyjnych,
- bloki komutacyjne - kodery, dekodery, translatory, multipleksery, demultipleksery,
- bloki arytmetyczne - sumator, układ mnożący, odejmujący, komparator,
- sterowanie pracą plotera, unipolarnym silnikiem krokowym,
- synteza układów sekwencyjnych – studenci wykorzystują również projekty liczników przygotowanych w ramach ćwiczeń tablicowych,
- projekt własny - tematyka do uzgodnienia, np. cyfrowe życzenia noworoczne, cyfrowa reklama miejsca urodzenia, moje hobby, reklama UMG,
- liczniki i rejestry scalone,
- układy uzależnień czasowych,
- układ sterowania pracą windy cumowniczej, diagnostyka, symulacja awarii, przy czy dostępne są dwa modele: jeden wykonany w technologii TTL, drugi - CPLD,
- sterowanie modelem dźwigu.

Na podkreślenie zasługuje fakt wykorzystania w laboratorium wielu modeli obiektów sterowania, studenci w czasie zajęć laboratoryjnych powinni uruchomić określony obiekt, a algorytm sterowania powinien spełniać przedstawione w instrukcji laboratoryjnej wymagania. Istotne jest także poznanie przez studentów zasad sterowania obiektami i urządzeniami z którymi mogą się spotkać w czasie praktyki lub pracy zawodowej.

Przykładem wykorzystywanego obiektu jest prosty model dźwigu, który został wykonany na bazie zabawki dla dzieci i przystosowany do sterowania za pomocą sygnałów logicznych. Model składa się z podstawy w której umieszczono sterownik, obrotowej wieży i wysięgnika, po którym porusza się wózek z wyciągarką haka. Do poruszania elementami modelu zastosowano trzy silniki prądu stałego: silnik nr 1 zapewniający obrót wieży w podstawie, nr 2 napędzający wyciągarkę haka, nr 3 zapewniający ruch wózka po wysięgniku. Każdy silnik jest sterowany dwoma sygnałami logicznymi: Zx załącza zasilanie silnika, Kx określa kierunek ruchu, gdzie x to numer silnika. Na rysunku 1 został przedstawiony model wraz z zaznaczonymi elementami wykonawczymi. Przykładowy algorytm sterowania sekwencyjnego, dostępny na laboratoryjnym serwerze, realizuje następujące ruchy: stop, hak w górę, wieża obrót w lewo, karetka do przodu, hak w dół, stop, hak w górę, karetka do tyłu, wieża obrót w prawo, hak w dół, stop. Do budowy układu sterującego wykorzystano 8–bitowy licznik (wyjścia g3, g2, g1, g0), demultiplekser posiadający 4 wejścia adresowe z aktywnymi jedynkami na wyjściu oraz bramki logiczne (rys. 2).



Rys. 1. Model dźwigu



Rys. 2. Układ logiczny sterowania pracą dźwigu

W celu spowolnienia pracy układu dokonano przesunięcia wyjść licznika o dwa bity w lewo. Układ posiada ponadto wejście EN, które umożliwia zatrzymanie pracy dźwigu oraz wejście RES przywracające stan zerowy licznika. W tabeli 1 przedstawiono stany sygnałów analizowanego układu.

Tabela 1. Stany układu sterowania pracą dźwigu

stan licznika	b1	b2	b3	Z1	K1	Z2	K2	Z3	K3
0	X	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	0	0
2	0	0	1	1	0	0	0	0	0
3	0	1	1	0	0	0	0	1	0
4	0	1	0	0	0	1	0	0	0
5	X	0	0	0	0	0	0	0	0
6	1	1	0	0	0	1	1	0	0
7	1	1	1	0	0	0	0	1	1
8	1	0	1	1	1	0	0	0	0
9	0	1	0	0	0	1	0	0	0
10	X	0	0	0	0	0	0	0	0

Sygnały sterujące silnikami są określone następującymi zależnościami

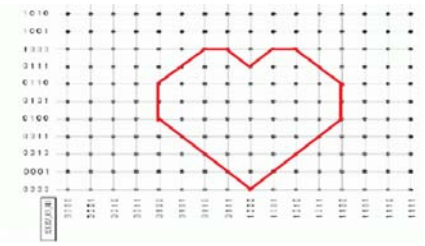
$$Z1 = b3 * \overline{b2}; \quad Z2 = b2 * \overline{b3}; \quad Z3 = b2 * b3; \tag{1}$$
$$K1 = b1 * Z1; \quad K2 = b1 * Z2; \quad K3 = b1 * Z3.$$

gdzie:

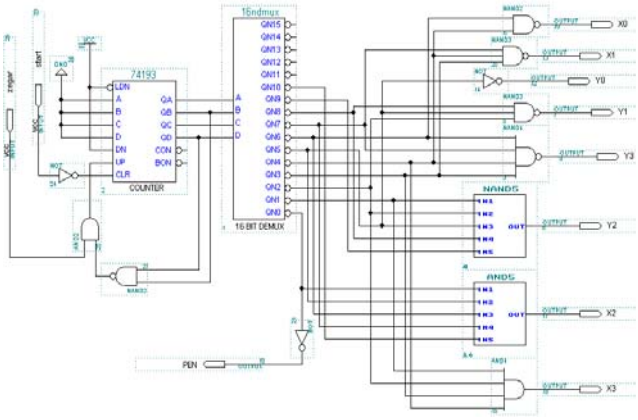
$$b1(g0, g1, g2, g3) = \overline{\sum(2, 3, 4, 9)};$$
$$b2g0, g1, g2, g3) = \overline{\sum(0, 2, 5, 8, 10)}; \tag{2}$$
$$b3(g0, g1, g2, g3) = \sum(2, 3, 7, 8).$$

Algorytm sterowania współpracuje z platformą edukacyjną zawierającą układ programowalny CPLD, wyświetlacze 7-segmentowe oraz przełączniki. Na wyświetlaczach 7-segmentowych pojawia się informacja o numerze pracującego silnika oraz kierunku ruchu, przy czym przykładowo D oznacza ruch haka w dół, L – obrót wieży w lewo, E – stop. W celu poprawy czytelności schematu niektórym sygnałom wejściowym i wyjściowym przyporządkowano nazwę połączenia, co wyeliminowało konieczność rysowanie wielu linii. Innym sposobem zwiększenia przejrzystości byłoby zastosowanie wirtualnej szyny połączeniowej.

Innym przykładem wykorzystywanego modelu jest ploter. Zadaniem studenta jest opracowanie algorytmu sterowania pracą plotera, przy pomocy którego w układzie współrzędnych X, Y należy wykreślić określony element graficzny, przy czym obie osie opisane są 4 bitami. Przykładowy element graficzny został przedstawiony na rysunku 3. Natomiast rysunek 4 prezentuje schemat logiczny układu sterowania, przy czym wykorzystano licznik 4-bitowy, demultiplekser oraz bramki logiczne. Uzyskane przebiegi czasowe (rys. 5) potwierdzają prawidłowość algorytmu sterowania. Ćwiczenie umożliwia poznanie zasad działania urządzeń pracujących w płaszczyźnie ruchu XY oraz zasad sterowania pracą plotera z wykorzystaniem układów logiki programowalnej.

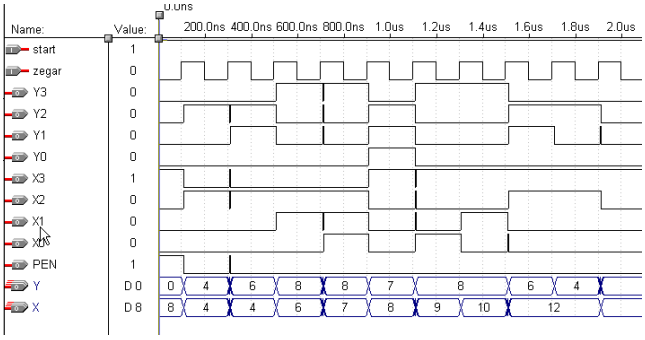


Rys. 3. Element graficzny



Rys. 4. Schemat logiczny układu sterowania pracą plotera

Zakres ćwiczeń laboratoryjnych jest uzgadniany indywidua-lnie z każdą grupą, częściowo jest to związane ze stopniem przygotowania do zajęć. Niestety zdarza się, że student nie jest przygotowany do zajęć. Podobne wnioski dotyczą także sprawozdań z zrealizowanych zajęć, które są czasami oddawane ze znacznym opóźnieniem, nie zawierają wniosków.

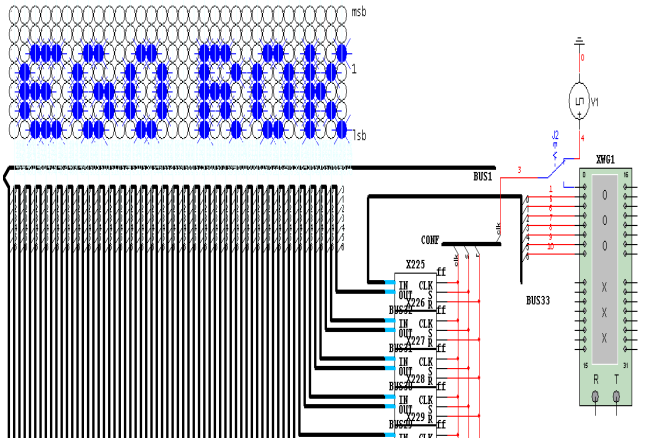


Rys. 5. Przykładowe przebiegi czasowe sygnałów sterujących pracą plotera

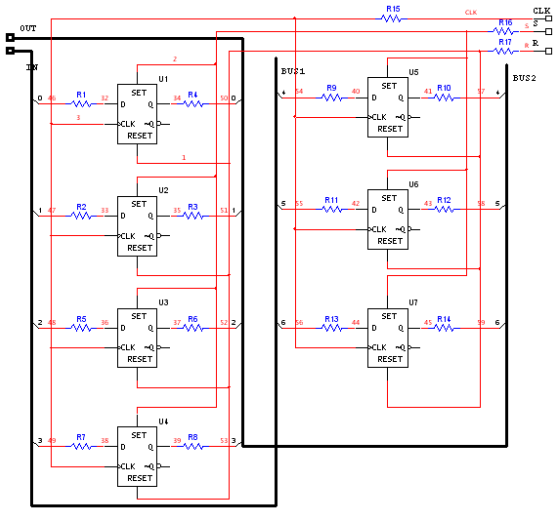
Niektórzy studenci wykazują się również mniejszym zainteresowaniem w przygotowaniu układu bardziej ambitnego, wymagającego samodzielnego pomysłu, nieco większego nakładu pracy. Przykładem są zajęcia, w czasie których studenci realizują projekt będący podsumowaniem wiadomości zdobytych w czasie pierwszej serii zajęć. Studentom pozostawiono dużą dowolność, zakładając konieczność wykorzystania środowiska Multisim, wirtualnej szyny, elementów generujących dźwięk oraz symulacji ruchu, np. z wykorzystaniem układu z krążącym zerem lub jedynką. Środowisko Multisim posiada bardzo dużo elementów typu kolorowe diody, monitor LCD, buzzery, możliwe jest także zastosowanie wirtualnej szyny, wstawianie dowolnych plików graficznych lub stworzenie własnego podobwodu. Tematyka projektu jest narzucona. Przykładem ciekawego projektu jest układ, w którym student przedstawił kontur mapy swojej gminy, z zaznaczeniem przepływającej rzeki oraz miejsca urodzenia. Do zaznaczenia granic gminy i rzeki zostały wykorzystane kolorowe diody, które są zapalane cyklicznie, zgodnie z pracą zaprojektowanego rozdzielacza impulsów, przy czym układ zawiera licznik, demultiplekser, bramki logiczne, generator fali prostokątnej. W przypadku zapalenia się diod wskazujących miejscowość dodatkowo jest uruchamiany buzzer. Projekt wymagał wykorzystania sporej ilości diod, co zajęło studentowi trochę tak cennego czasu, jednak koncepcja sterowania nie jest skomplikowana. W tym przypadku istotny był także pomysł, wykonanie i jak zwykle dobre chęci.

Innym ciekawym projektem opracowanym w czasie zajęć są życzenia noworoczne opracowane w środowisku Multisim. Na rysunku 6 przedstawiono część opracowanego schematu, ze względu na powtarzalność pominięto dolną część. Z 224 diod, umieszczonych w 32 kolumnach i 7 wierszach, zbudowano wyświetlacz pozwalający na przesuwanie tekstu *Wesołych świąt Bożego Narodzenia oraz wystrzałowego Nowego Roku*. Jest to wyświetlacz pozwalający na przesuwanie tekstu (obrazu) z prawej do lewej strony. Do budowy wykorzystano przerzutniki typu D. Pracują one jako rejestr przesuwny, są odpowiedzialne za zmianę stanu diody, tj. po pojawieniu się kolejnego zbocza narastającego na wejściu zegarowym podświetlony zostaje odpowiedni piksel wyświetlanego tekstu. Każda kolumna wyświetlacza stanowi osobny rejestr. Ze względu na taką samą budowę każdego rejestru zbudowano go jako podobwód (rys. 7) i wykorzystano 32 razy (na rys. 6 zostały przedstawione tylko 4 rejestry). Układ można w znaczny sposób uprościć, przykładowo po zastosowaniu oktalnych przerzutników typu D, np. 74VHC273, 74HC574.

W celu uzyskania odpowiednich stanów na wejściach szeregowych rejestru zastosowano generator słów WG (ang. Word Generator) taktowany z tego samego źródła co rejestr przesuwny. Przykładowe stany generowane przez WG zostały przedstawione na rysunku 8. Word Generator służy do generacji słów binarnych. Kolejne słowa są umieszczane w oknie w zapisie heksadecymalnym, dziesiętnym, binarnym lub ASCII, zależnie od wybranej opcji w polu *Display*.

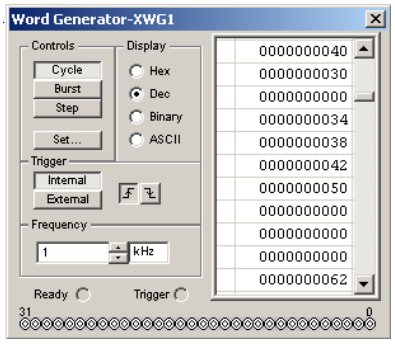


Rys. 6. Fragment układu wyświetlającego tekst



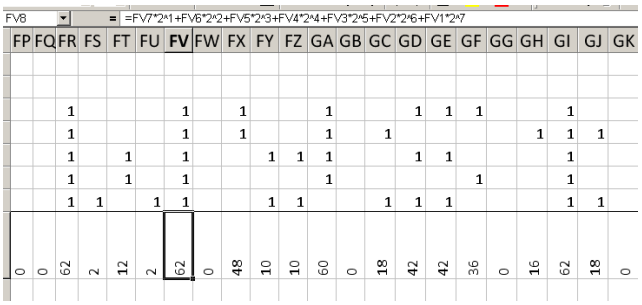
Rys. 7. Rejestr przesuwny

Generowane słowa można wprowadzić ręcznie lub, po naciśnięciu przycisku *Set,* można skorzystać z gotowych wzorców, takich jak liczniki zliczające w górę i w dół, rejestry przesuwne. Można również wprowadzoną sekwencję zapisać lub odczytać z osobnego pliku. Wpisana sekwencja słów jest zapisywana w pliku projektu razem ze schematem. W czasie pracy generatora wartość kolejnych słów pojawia się w dolnej części okna.



Rys. 8. Generator słów

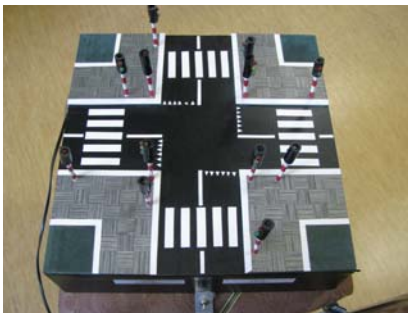
W omawianym przykładowym projekcie do określenia wprowadzanych stanów wykorzystano środowisko Excel, w którym w odpowiednich wierszach i kolumnach zakodowano, przy pomocy zer i jedynek, wyświetlany tekst. Na rysunku 9 zaprezentowano widok części arkusza, tj. napis *Wyst*. Następnie do każdej komórki w wybranym wierszu arkusza Excel wpisano zależność, która umożliwiła automatyczne określenie stanów wpisywanych do Word Generatora. Przy określeniu tych zależności, założono, że dolna linijka zbudowanego wyświetlacza posiada najmniejszą wagę LSB (ang. Least Significant Bit), natomiast linijka górna posiada wagę najwyższą MSB (ang. Most Significant Bit), przy czym nie wykorzystano bitu 2^0. Uzyskane w arkuszu Excel wielkości wprowadzono ręcznie do WG.



Rys. 9. Fragment zapisu kodu dla generowanego tekstu

Z przedstawionych projektów wynika, że stopień złożoności realizowanych algorytmów i projektów jest różny. Zadania przygotowane i uruchamiane przez studentów prezentują różny poziom skali trudności, nakładu pracy i pomysłowości. Również sterowanie silnikiem krokowym, modelem windy cumowniczej wymaga od studentów nieco większej samodzielności i pomysłowości, także w tym przypadku poziom przygotowanych projektów jest skrajnie różny.

Nieco mniej czasochłonne jest przygotowanie algorytmów sterowania w języku VHDL w edytorze tekstowym środowiska Max Plus Baseline lub Quartus. W laboratorium TC w Katedrze Automatyki Okrętowej został zbudowany miedzy innymi model skrzyżowania ulicznego, który umożliwia opracowanie algorytmu sterowania światłami ulicznymi. Schemat makiety został przedstawiony na rysunku 10. Opracowany przykładowy algorytm sterowania umożliwia sterowanie światłami ulicznymi w trybie dziennym i nocnym. W trybie nocnym, w celu polepszenia dynamiki jazdy, sygnalizacja świetlna jest wyłączona, obowiązują wówczas znaki drogowe. Dla pojazdów miga światło żółte, ostrzegające, że należy zwracać uwagę na znaki drogowe i zachować szczególną ostrożność. Wówczas też światło na przejściu dla pieszych jest nieczynne. W trybie dziennym zasada działania sygnalizatorów świetlnych na skrzyżowaniu prostopadłym została opracowana również zgodnie z obowiązującymi przepisami ruchu drogowego.



Rys. 10. Widok makiety skrzyżowania ulicznego

Do sterowania światłami wykorzystano ośmiobitową zmienną *WY*. Jej poszczególne bity są odpowiedzialne za konfigurację świateł, zgodnie z zapisem: *Wy0* - P2 zielone, *Wy1* - S2 zielone, *Wy2* - S2 żółte, *Wy3* - S2 i P2 czerwone, *Wy4* - P1 zielone, *Wy5* - S1 zielone, *Wy6* - S1 żółte, *Wy7* - S1 i P1 czerwone, przy czym litera P oznacza sygnalizację dla pieszych, S dla samochodów, natomiast 1 oznacza kierunek ruchu poziomy, a 2 pionowy. Sterowanie odbywa się za pomocą czterech wejść:

- noc - jeżeli jest w stanie wysokim, to układ działa w trybie nocnym,
- clk - zegar taktujący licznik programu,
- test - zapala wszystkie światła, w celu testowania,
- reset - umożliwia zerowanie licznika programu i ponowny start.

Przykładowe uzyskane przebiegi czasowe oraz prawidłowa kolejność zapalania się świateł ulicznych potwierdza prawidłową konstrukcję algorytmu sterowania.

Analiza przygotowywanych samodzielnie przez studentów projektów laboratoryjnych w okresie ostatnich 20 lat (autorka artykułu posiada spore archiwum prac studentów) spowodowała sformułowanie dosyć pesymistycznych wniosków. W ostatnim okresie projekty te są mniej ambitne. Wynika to między innymi z dużej liczby godzin zajęć dydaktycznych, dużej ilości zagadnień omawianych w całym toku nauczania, konieczności pracy zarobkowej nawet przez studentów studiów dziennych, możliwości i chęci rozwijania innych zainteresowań. W tym przypadku istotna jest także motywacja studenta do systematycznej nauki, jego możliwości i indywidualne predyspozycje. Na szczęście są również i optymistyczne wnioski. W roku akademickim 2017/2018, w 40-lecie pracy dydaktycznej autorki artykułu, po raz pierwszy grupa laboratoryjna złożona z dwóch studentów była zainteresowana szybszym wykonaniem ćwiczeń, które wykonała w dodatkowym, uzgodnionym terminie. Studenci wykonali wszystkie ćwiczenia z wyprzedzeniem miesiąca, byli zawsze perfekcyjnie przygotowani do zajęć. W roku akademickim 2018/2019, na początku zajęć studentów poinformowałam o zdarzeniu, wzbudziło to pozytywne zainteresowanie i tym razem już sześciu studentów uporało się z ćwiczeniami szybciej. Stąd wniosek, że dobrze zorganizowany czas, chęci i motywacja do systematycznej nauki mogłaby rozwiązać wiele problemów z którymi borykają się studenci i nauczyciele akademiccy.

Na studiach magisterskich studenci w czasie zajęć laboratoryjnych opracowują algorytmy sterowania z wykorzystaniem edytora tekstowego języka VHDL i środowiska Quartus. Również ich zadaniem jest implementacja algorytmu w strukturze układu programowalnego FPGA, który jest dostępny na platformie edukacyjnej DE2_115. Od kilku lat studenci pracują w grupach najczęściej dwu osobowych, od początku zajęć zajmują się jednym zagadnieniem, np. sterowaniem modelem koparki, elektrowni okrętowej, windy cumowniczej, modelami aut wyposażonymi dodatkowo w ultradźwiękowe czujniki odległości. Opracowane algorytmy muszą zawierać także informacje dla użytkownika, czyli przykładowo na wyświetlaczach 7-segmentowych oraz monitorze LCD powinny pojawiać się informacje o stanie sterowanego obiektu. Można korzystać z opracowanych przez innych studentów bibliotek, które są dostępne na serwerze laboratoryjnym. Na kolejnych zajęciach studenci przedstawiają efekty swoich prac, wprowadzają kolejne elementy, poprawiają błędy. Są oni zadowoleni z dyskusji na forum całej grupy, możliwości pracy zespołowej, twierdzą, że lepiej uczą się w grupie. Podkreślają, że jest to bardzo istotne, a wręcz niezbędne, w przyszłej pracy zawodowej. Wcale nie tak rzadko spotykają się dodatkowo w warunkach domowych w celu rozbudowy lub ulepszenia projektu. Warunkiem zaliczenia laboratorium jest prezenta-

cja działania zaprojektowanego algorytmu oraz pisemne przedstawienie sprawozdania, czyli dokumentacji technicznej zgodnej z obowiązującymi standardami.

Podsumowanie

Wieloletnie doświadczenie dydaktyczne autorki artykułu w zakresie nauczania TC spowodowało, że w celu poprawy poziomu wiedzy reprezentowanej przez studentów zostały wprowadzone dodatkowe projekty kontrolne, które student powinien rozwiązać po wykładach i zajęciach ćwiczeniowych, co jest również zgodne z sugestiami przedstawionymi w pracy [21]. Rozwiązanie tych zadań pozwala na systematyczną pracę studenta, ułatwia zrozumienie zagadnień omawianych na kolejnych zajęciach. Jednym z warunków powodzenia tej metody jest postawa i zaangażowanie studenta. Brak odpowiedniej wiedzy w znacznym stopniu utrudnia, a czasami nawet uniemożliwia właściwe przygotowanie do zajęć laboratoryjnych. Studenci prawidłowo przygotowani do tych zajęć nie mają problemów z zaprojektowaniem i połączeniem układu. Według studentów zajęcia laboratoryjne ułatwiły im zdobycie wiedzy praktycznej, zrozumienie wielu zagadnień z zakresu TC. Niestety, studenci UMG inaczej oceniają wejściówki niż studenci autora artykułu [21]. Uważają, że są one zbędne, skracają o około 15 minut czas poświęcony na zajęcia praktyczne. Nie zawsze brak zaliczenia wejściówki powodował brak wykonania ćwiczenia. Stąd wniosek, że czasami wystarczy kilkanaście minut poświęconych na zrozumienie zagadnienia i wówczas jest możliwe prawidłowe wykonanie ćwiczenia. Dobrze przygotowany student nie powinien mieć kłopotów z zaliczeniem zajęć, wykonaniem ćwiczeń laboratoryjnych, z wykorzystaniem układów i urządzeń rzeczywistych. Nie powinien ponadto mieć problemów w przyszłej pracy zawodowej.

Proces nauczania wymaga nie tylko znacznej aktywności ze strony nauczyciela. Również student powinien być odpowiedzialny za proces uczenia się, co zostało słusznie podkreślone w pracy [4]. Wymaga to sporego wysiłku z obu stron, co w dzisiejszych czasach nie jest łatwe w realizacji. Zmiany w organizacji planu zajęć w UMG, ze względu na znaczne tygodniowe obciążenie zajęciami, jest raczej niemożliwe. Na dzień dzisiejszy nie jest także możliwe przeprowadzenie w czasie wykładów krótkich quizów z zastosowaniem tabletu graficznego, czego powodem jest brak takiego urządzenia. Być może sytuacja ulegnie zmianie w związku z obowiązywaniem od 2018 roku nowej Ustawy o Szkolnictwie Wyższym, w której duży nacisk położono na doskonałość i jakość kształcenia.

Rozwijająca się elektronika, technika cyfrowa, wymaga ciągłej korekty zakresu materiału prezentowanego na zajęciach dydaktycznych, bieżącej analizy dostępnego sprzętu, środowisk wspomagających projektowanie układów cyfrowych. Istotne jest również odpowiednie wyposażenie laboratorium, co wymaga znacznych nakładów finansowych [5]. Z tym problemem zmagają się pracownicy dydaktyczni, mając nadzieję, że nowa ustawa chociaż w drobnym zakresie rozwiąże problem.

***Autor**: dr inż. Krystyna Maria Noga, Uniwersytet Morski w Gdyni, Katedra Automatyki Okrętowej, ul. Morska 81-87, 81-225 Gdynia, E-mail: k.noga@we.umg.edu.pl.*

LITERATURA

[1] M.Górecka, K. Górecki, Porównanie wybranych narzędzi do komputerowej analizy układów cyfrowych, *Przegląd Elektrotechniczny*, 94 (2018), nr. 11, 72-75

[2] K. M. Noga, M. Radwański, The teaching of digital techniques with Multisim 2001, *9th Baltic Region Seminar on Engineering Education, Monash Engineering Educations Series*, Gdynia Maritime University, 17 – 20 June 2005, Seminar Proceedings, 45-48

[3] K. M. Noga, M. Radwański, Multisim. Technika cyfrowa w przykładach, *Wydawnictwo BTC*, Legionowo 2009

[4] P. Dębiec, Zorientowane na studenta nauczanie podstaw techniki cyfrowej", *Przegląd Elektrotechniczny*, 94 (2018), nr. 9, 117–120

[5] P. Dębiec, Effective Learner-Centered Approach for Teaching an Introductory Digital Systems Course, *IEEE Transactions on Education*, 61 (2018), n. 1, 38-45

[6] P. Dębiec, M. Byczuk, Teaching discrete and programmable logic design techniques using a single laboratory board, *IEEE Transactions on Education,* 54 (2011), n. 4, 652-656

[7] M. F. Brejza, J. Hooker, S. D. Oakley, R. G. Maunder, Design of digital for undergraduate microelectronic teaching, *11 European Workshop on Microelectronics Education (EWME)*, 2011, 1-4

[8] K. M. Noga, Nowe stanowiska dydaktyczne do nauki cyfrowego sterowania z wykorzystaniem układów programowalnych, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 30 (2011), 93–96

[9] K. M. Noga, Zajęcia laboratoryjne z Techniki Cyfrowej w Akademii Morskiej, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 46 (2015), 85-88

[10] K. M. Noga, Wykorzystanie robota humanoidalnego ROBOWISDOM w dydaktyce techniki cyfrowej, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 51 (2016), 127-130

[11] K. M. Noga, M. Radwański, Our Stations for Teaching Programmable Devices, *Innovations in E-learning, Instruction Technology, Assessment and Engineering Education, Polytechnic University*, USA, Springer 2008, 268-273

[12] K. M. Noga, M. Radwański, Modern improvements in the digital logic laboratory, *Technological Developments in Networking, Education and Automation*, USA, Springer 2010, 109-114

[13] K. M. Noga, M. Radwański, Our experiences in teaching of digital logic, *Innovations in E-learning, Instruction Technology, Assessment and Engineering Education*, Polytechnic University, USA, Springer 2007, 237-242

[14] D, Cetnarowicz, P. Kardyś, A. Dąbrowski, P. Pawłowski, PLC – elektroniczny element automatyki przemysłowej, *Przegląd Elektrotechniczny*, 94 (2018), n. 9, 121-124

[15] D. L. Foster, An undergraduate survey course on asynchronous sequential logic, ladder logic and fuzzy logic, *IEEE Transactions on Education*, 55 (2012), n. 4,. 459-465

[16] http://atol.am.gdynia.pl\tc

[17] K. M. Noga, Cyfrowe sterowanie z zastosowaniem układów programowalnych. *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 40 (2014), 73-76

[18] K. M. Noga, Modele sterowane cyfrowo, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 46 (2015), 81-84

[19] K. M. Noga, Wykorzystanie platform Arduino UNO oraz DE2_115 do sterowania modelami aut, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, 60 (2018), 85–90

[20] K. M. Noga, Zastosowanie układów programowalnych i języka VHDL w nauczaniu sterowania cyfrowego, *Zeszyty Naukowe Akademii Morskiej w Gdyni*, 98 (2017), 210-216

[21] P. Pawłowski, A. Dąbrowski, A. Stankiewicz, K. Piniarski, Kurs podstaw elektroniki dla studentów kierunku Automatyka i Robotyka, *Przegląd Elektrotechniczny*, 94 (2018), n. 8, 67-70