



# Politechnika Świętokrzyska

## WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI

Załącznik nr 9  
do Zarządzenia Rektora Nr 35/19  
z dnia 12 czerwca 2019 r.

### IV. Opis programu studiów

#### 3. KARTA PRZEDMIOTU

Kod przedmiotu	<b>E-AiEP-08-s6</b>
Nazwa przedmiotu	<b>Układy programowalne w automatyce</b>
Nazwa przedmiotu w języku angielskim	<b>Programmable Logic Devices in Control Systems</b>
Obowiązuje od roku akademickiego	<b>2019/20</b>

#### USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	<b>Automatyka i Elektrotechnika Przemysłowa</b>
Poziom kształcenia	<b>I stopień</b>
Profil studiów	<b>Ogólnoakademicki</b>
Forma i tryb prowadzenia studiów	<b>Studia stacjonarne</b>
Zakres	
Jednostka prowadząca przedmiot	<b>Katedra Elektrotechniki Przemysłowej i Automatyki</b>
Koordynator przedmiotu	<b>dr hab. inż. Grzegorz Radomski</b>
Zatwierdził	<b>Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk</b>

#### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<b>Przedmiot kierunkowy</b>
Status przedmiotu	<b>Wybieralny</b>
Język prowadzenia zajęć	<b>Polski</b>
Usytuowanie modułu w planie studiów - semestr	<b>Semestr VI</b>
Wymagania wstępne	<b>Układy cyfrowe</b>
Egzamin (TAK/NIE)	<b>Nie</b>
Liczba punktów ECTS	<b>4</b>

<b>Forma prowadzenia zajęć</b>	<b>wykład</b>	<b>ćwiczenia</b>	<b>laboratorium</b>	<b>projekt</b>	<b>Inne</b>
<b>Liczba godzin w semestrze</b>	<b>30</b>	<b>0</b>	<b>30</b>	<b>0</b>	<b>0</b>

## EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Zna i rozumie w zaawansowanym stopniu zagadnienia projektowania złożonych systemów cyfrowych.	AiEP1_W10
	W02	Zna budowę/ struktury układów logiki programowalnej.	AiEP1_W10
	W03	Zna zagadnienia programowania/konfiguracji układów logiki programowalnej.	AiEP1_W10
	W04	Zna różne formy opisu układów cyfrowych.	AiEP1_W10
	W05	Zna języki programowania systemów cyfrowych.	AiEP1_W10
	W06	Zna techniki sprzęgania układów logiki programowalnej w systemami mikroprocesorowymi i układami pomiarowymi oraz wykonawczymi.	AiEP1_W09, AiEP1_W10
Umiejętności	U01	Potrafi dokonać dekompozycji funkcji systemu cyfrowego na podukłady.	AiEP1_U01, AiEP1_U08
	U02	Potrafi programować złożone systemy cyfrowe.	AiEP1_U03, AiEP1_U04
	U03	Potrafi łączyć w funkcjonalną całość różne formy opisu działania podukładów systemów cyfrowych.	AiEP1_U03, AiEP1_U04
	U04	Potrafi zarządzać projektem systemu cyfrowego z użyciem środowiska do projektowania systemów logiki programowalnej.	AiEP1_U04
	U05	Potrafi dokonać weryfikacji poprawności projektu systemu cyfrowego z użyciem narzędzi symulacyjnych.	AiEP1_U10
	U06	Potrafi uruchamiać systemy cyfrowe logiki programowalnej.	AiEP1_U09
	U07	Potrafi posługiwać się sprzętem pomiarowym w celu uruchamiania układów logiki programowalnej.	AiEP1_U09
	U08	Potrafi czytać dokumentację, schematy systemów cyfrowych.	AiEP1_U08
	U09	Potrafi opracowywać dokumentację projektu systemu cyfrowego z użyciem narzędzi dostarczanych przez środowisko projektowe.	AiEP1_U08, AiEP1_U15
	U10	Potrafi pracować w zespole projektowym ale również samodzielnie realizować przydzielone zadania projektowe.	AiEP1_U14
	U11	Potrafi stosować normy bezpieczeństwa w projektach oraz w praktyce zawodowej.	AiEP1_U12
	U12	Potrafi samodzielnie poszerzać i uaktualniać swoją wiedzę zarówno z zakresu systemów cyfrowych jak i dziedzin, w których je stosuje.	AiEP1_U11, AiEP1_U13
Kompetencje społeczne	K01	Rozumie odpowiedzialność projektanta systemu cyfrowego za bezpieczeństwo jego użytkowników.	AiEP1_K3, AiEP1_K4, AiEP1_K5
	K02	Rozumie i uwzględnia w swojej działalności projektowej bezpieczeństwo systemów cyfrowych stosując zabezpieczenia przed nieautoryzowanym dostępem do funkcji systemu.	AiEP1_K3, AiEP1_K4, AiEP1_K5
	K03	Rozumie konieczność ciągłego poszerzania wiedzy i umiejętności w związku z koniecznością rozumienia działania obiektu sterowania oraz bardzo szybkim rozwojem dziedziny systemów cyfrowych.	AiEP1_K1, AiEP1_K5

	K04	Rozumie związki i wpływ systemów cyfrowych na środowisko i inne dziedziny techniki.	AiEP1_K2, AiEP1_K4, AiEP1_K5
--	-----	---	------------------------------------

## TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1-2. Czym są układy logiki programowalnej? Przegląd układów logiki programowalnej. Układy CPLD, matryce FPGA – ich programowanie i konfiguracja. Interfejsy stosowane do programowania i konfiguracji układów logiki programowalnej. Interfejs JTAG. Wykorzystanie wyprowadzeń układów w fazie programowania, konfiguracji i działania zaprogramowanego systemu cyfrowego.
	3-7. Środowisko Quartus II++ firmy Altera. Definicja projektu. Tworzenie komponentów. Użycie komponentów. Kompilacja projektu. Definiowanie pliku wymuszeń wektorów testowych. Symulacja funkcjonalna i czasowa projektu. Definiowanie fizycznych wyprowadzeń wejść i wyjść układu. Definicja stanu nieużywanych w projekcie wyprowadzeń układu. Generacja plików dla programatora. Programowanie układów fizycznych.
	8-10. Formy opisu projektu układu logicznego. Opis schematyczny. Opis z użyciem języka HDL. Język VHDL. Jednostki projektowe języka VHDL (Entity, Architecture, Process). Deklaracja interfejsu komponentu (Port). Sygnały i zmienne pojęcia rozróżnienie. Instrukcje podstawienia dla sygnałów i dla zmiennych. Instrukcje stosowane w części Architecture: instrukcje strukturalne, instrukcje procesowe – działanie tych dwóch rodzajów instrukcji sposób ich użycia, efekty ich działania. Instrukcja synchronizująca. Deklaracje typów, podtypów, zmiennych i stałych w języku VHDL. Instrukcje sterujące: warunkowe, pętle, skoki. Podstawowe typy stosowane w opisie układów logicznych: std_logic, std_logic_vect, integer. Operatory. Metodologia projektowania i programowania złożonych systemów cyfrowych. Złożoność układowa, wybór fizycznych wejść wyjść systemu, ścieżki krytyczne, maksymalna częstotliwość zegarowa, szybkość działania systemu.
	11-13. Metodologia programowania złożonych systemów cyfrowych w strukturach logiki programowalnej. Sprzęganie realizowanych systemów w procesorami w spójny system sterowania. Aspekty dekompozycji funkcji sterujących pomiędzy mikroprocesor i układy logiki programowalnej. Realizacja specjalizowanych, autonomicznych układów sterowania i koprocesorów wspierających system mikroprocesorowy w realizacji algorytmu sterującego. Realizacja podukładów: dekodery adresowych, rejestrów konfiguracyjnych, rejestrów stanu, układów przerwań, resetów fizycznych i programowych. Wykorzystanie gotowych komponentów: pętli PLL, wielofazowych przebiegów zegarowych, układów arytmetycznych.

	<p>14-15. Analiza przykładowych projektów (do wyboru w konsultacji ze studentami oraz innymi prowadzącymi – tworzenie narzędzi do innych projektów):</p> <ul style="list-style-type: none"> <li>• Opis prostych układów logicznych,</li> <li>• Licznik rewersyjny,</li> <li>• Realizacja modułu pamięci RAM, dwukierunkowa szyna danych, dostęp do magistrali z użyciem buforów trójstanowych,</li> <li>• Moduł klawiatury z eliminacją drgań zestyków,</li> <li>• Układ nadawczo-odbiorczy synchronicznej transmisji szeregowej,</li> <li>• Moduł sterownika wyświetlacza LCD,</li> <li>• Realizacja sterownika mikroprogramowalnego o zadanej liście mikroinstrukcji (np.: wystawienia wektora wyjściowego, skoku warunkowego pod adres względny, skoku bezwarunkowego pod adres względny, skoku warunkowego pod adres bezwzględny, skoku bezwarunkowego pod adres bezwzględny, wywołania podprogramu, powrotu z podprogramu),</li> <li>• Moduł komórki sztucznego neuronu i z jego użyciem sztucznej sieci neuronowej,</li> <li>• Generatory przebiegów: prostokąt, trójkąt, sinus-cosinus,</li> <li>• Modulatory: amplitudy, częstotliwości, fazy, próbkowanie sygnałów,</li> <li>• Układy filtrów cyfrowych,</li> <li>• Sterowniki różnych typów przetwornic impulsowych,</li> <li>• Programowalny 3-fazowy modulator wektora przestrzennego napięcia SVPWM,</li> </ul> <p>Sterowniki przekształtników z miękką komutacją.</p>
laboratorium	1. Zapoznanie z systemem Quartus II++. Uruchamianie przykładowych projektów na płytach układów logiki programowalnej, użycie oscyloskopu MSO (Mixed Signal Oscilloscope) do uruchamiania i weryfikacji poprawności działania projektów złożonych układów cyfrowych.
	2. Programowanie prostych systemów cyfrowych. Proces programowania i symulacji systemu cyfrowego. Uruchamianie projektów na platformie sprzętowej.
	3. Programowanie prostych systemów cyfrowych. Proces programowania i symulacji systemu cyfrowego. Uruchamianie projektów na platformie sprzętowej.
	4. Rozdanie laboratoryjnych zadań projektowych i ich dyskusja. Projektowanie i programowanie złożonych systemów cyfrowych i ich symulacja funkcjonalna.
	5. Projektowanie i programowanie złożonych systemów cyfrowych i ich symulacja funkcjonalna.
	6. Symulacja czasowa projektów. Poprawianie błędów. Doskonalenie projektu.
	7. Poprawianie błędów projektowych. Doskonalenie projektu.
	8. Uruchamianie projektów na platformie sprzętowej. Weryfikacja układowa.

\*) zostawić tylko realizowane formy zajęć

## METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X	X		
W02			X	X		
W03			X	X		
W04			X	X		
W05			X	X		
W06			X	X		
U01			X	X		
U02			X	X		
U03			X	X		

U04			X	X		
U05			X	X		
U06			X	X		
U07			X	X		
U08			X	X		
U09			X	X		
U10			X	X		
U11			X	X		
U12			X	X		
K01			X	X		
K02			X	X		
K03			X	X		
K04			X	X		

### FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	<i>Dokumentacja projektu laboratoryjnego. Sprawdzian praktyczny i ustny przy oddaniu projektu laboratoryjnego.</i>
laboratorium	Wybierz element.	<i>Dokumentacja projektu laboratoryjnego. Sprawdzian praktyczny i ustny przy oddaniu projektu laboratoryjnego.</i>

\*) zostawić tylko realizowane formy zajęć

### NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	30		30			h
2.	Inne (konsultacje, egzamin)*	2		2			h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>64</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>2,56</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>36</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>1,44</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>30</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>1,88</b>					ECTS
9.	<b>Sumaryczne godzinowe obciążenie pracą studenta</b>	<b>100</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>4</b>					

\* wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć

## **LITERATURA**

1. [Kania D.](#): „Układy logiki programowalnej”, Wydawnictwo Naukowe PWN, Warszawa 2012.
2. [Pawłowski M.](#), [Skorupski A.](#): „Projektowanie złożonych układów cyfrowych”, [Wydawnictwa Komunikacji i Łączności](#), Warszawa 2017.
3. Skahill K.: „Język VHDL Projektowanie programowalnych układów logicznych”, Wydawnictwa Naukowo-Techniczne, Warszawa 2010.
4. Ashenden P. J.: „The VHDL Cookbook”, Dept. Computer Science University of Adelaide South Australia 1990.
5. „VHDL Handbook”, ARDI AB Electronics.

*Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje*