



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-TD-11-s4
Nazwa przedmiotu	Układy FPGA
Nazwa przedmiotu w języku angielskim	FPGA systems
Obowiązuje od roku akademickiego	2020/21

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	<input type="text"/>
Poziom kształcenia	<input type="text"/>
Profil studiów	<input type="text"/>
Forma i tryb prowadzenia studiów	<input type="text"/>
Zakres	<input type="text"/>
Jednostka prowadząca przedmiot	<input type="text"/>
Koordynator przedmiotu	Dr hab. inż. Grzegorz Radomski
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<input type="text"/>
Status przedmiotu	<input type="text"/>
Język prowadzenia zajęć	<input type="text"/>
Usytuowanie modułu w planie studiów - semestr	<input type="text"/>
Wymagania wstępne	<input type="text"/>
Egzamin (TAK/NIE)	<input type="text"/>
Liczba punktów ECTS	<input type="text"/>

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	15		0	30	0

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.	T11_W03, T11_W04
	W02	Zna budowę/ struktury układów logiki programowalnej.	T11_W03, T11_W04
	W03	Zna zagadnienia programowania/konfiguracji układów logiki programowalnej.	T11_W03
	W04	Zna różne formy opisu układów cyfrowych.	T11_W06
	W05	Zna języki programowania systemów cyfrowych.	T11_W06
	W06	Zna techniki sprzęgania układów logiki programowalnej w systemami mikroprocesorowymi i układami pomiarowymi oraz wykonawczymi.	T11_W03
Umiejętności	U01	Potrąfi definiować funkcje systemu cyfrowego konieczne do realizacji zadania projektowego i dokonać dekompozycji funkcji systemu cyfrowego na podukłady.	TI_U5, T11_U13
	U02	Potrąfi programować złożone systemy cyfrowe.	TI_U06, TI_U07, T11_U17
	U03	Potrąfi łączyć w funkcjonalną całość różne formy opisu działania podukładów systemów cyfrowych.	TI_U06, T11_U17
	U04	Potrąfi zarządzać projektem systemu cyfrowego z użyciem środowiska do projektowania systemów logiki programowalnej.	T11_U11
	U05	Potrąfi dokonać weryfikacji poprawności projektu systemu cyfrowego z użyciem narzędzi symulacyjnych.	T11_U11
	U06	Potrąfi uruchamiać systemy cyfrowe logiki programowalnej.	TI_U10, T11_U11, T11_U20
	U07	Potrąfi posługiwać się sprzętem pomiarowym w celu uruchamiania układów logiki programowalnej.	T11_U21
	U08	Potrąfi czytać dokumentację, schematy systemów cyfrowych.	
	U09	Potrąfi opracowywać dokumentację projektu systemu cyfrowego z użyciem narzędzi dostarczanych przez środowisko projektowe.	T11_U14, T11_U11
	U10	Potrąfi pracować w zespole projektowym ale również samodzielnie realizować przydzielone zadania projektowe.	T11_U15
	U11	Potrąfi stosować normy bezpieczeństwa w projektach oraz w praktyce zawodowej.	T11_U15
	U12	Potrąfi samodzielnie poszerzać i uaktualniać swoją wiedzę zarówno z zakresu systemów cyfrowych jak i dziedzin, w których je stosuje.	T11_U01
	U13	Rozumie i uwzględni w swojej działalności projektowej bezpieczeństwo systemów cyfrowych stosując zabezpieczenia przed nieautoryzowanym dostępem do funkcji systemu.	T11_U18
Kompetencje społeczne	K01	Rozumie odpowiedzialność projektanta systemu cyfrowego za bezpieczeństwo jego użytkowników.	T11_K01
	K02	Rozumie konieczność ciągłego poszerzania wiedzy i umiejętności w związku z bardzo szybkim rozwojem dziedziny systemów cyfrowych.	T11_K03
	K03	Rozumie związki i wpływ systemów cyfrowych na środowisko i inne dziedziny techniki.	T11_K01

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	Zagadnienia organizacyjne, podanie literatury przedmiotu, wymagań i zasad zaliczenia i oceniania. Przedstawienie tematyki przedmiotu.
	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.
	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.
	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. Instrukcje 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.
	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: dekoderów adresowych, rejestrów konfiguracyjnych, rejestrów stanu, układów przerwań, resetów fizycznych i programowych. Wykorzystanie gotowych komponentów: pętli PLL, wielofazowych przebiegów zegarowych, układów arytmetycznych.
	Analiza przykładowych projektów (do wyboru w konsultacji ze studentami oraz innymi prowadzącymi – tworzenie narzędzi do innych projektów): <ul style="list-style-type: none"> • Opis prostych układów logicznych, • Licznik rewersyjny, • Realizacja modułu pamięci RAM, dwukierunkowa szyna danych, dostęp do magistrali z użyciem buforów trójstanowych, • Moduł klawiatury z eliminacją drgań zestyków, • Układ nadawczo-odbiorczy synchronicznej transmisji szeregowej, • Moduł sterownika wyświetlacza LCD, • Realizacja sterownika mikroprogramowalnego o zadanej liście mikroinstrukcji (np.: wystawienia wektora wyjściowego, skoku warunkowego pod adres względny, skoku bezwarunkowego pod adres względny, skoku bezwarunkowego pod adres bezwzględny, wywołania podprogramu, powrotu z podprogramu), • Moduł komórki sztucznego neuronu i z jego użyciem sztucznej sieci neuronowej, • Generatory przebiegów: prostokąt, trójkąt, sinus-cosinus, • Modulatory: amplitudy, częstotliwości, fazy, próbkowanie sygnałów, • Układy filtrów cyfrowych.
projekt	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.

	Programowanie prostych systemów cyfrowych. Proces programowania i symulacji systemu cyfrowego. Uruchamianie projektów na platformie sprzętowej.
	Programowanie prostych systemów cyfrowych. Proces programowania i symulacji systemu cyfrowego. Uruchamianie projektów na platformie sprzętowej.
	Rozdanie zadań projektowych i ich dyskusja. Projektowanie i programowanie złożonych systemów cyfrowych i ich symulacja funkcjonalna.
	Projektowanie i programowanie złożonych systemów cyfrowych i ich symulacja funkcjonalna.
	Symulacja czasowa projektów. Poprawianie błędów. Doskonalenie projektu.
	Poprawianie błędów projektowych. Doskonalenie projektu.
	Uruchamianie projektów na platformie sprzętowej. Weryfikacja układowa. Ocena projektów.

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów kształcenia					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			
W02			X			
W03			X			
W04			X			
W05			X			
W06			X			
U01				X		
U02				X		
U03				X		
U04				X		
U05				X		
U06				X		
U07				X		
U08				X		
U09				X		
U10				X		
U11				X		
U12				X		
U13				X		
K01			X	X		
K02			X	X		
K03			X	X		

A.

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład		Zaliczenie kolokwium.
projekt		Poprawne zrealizowanie zadania projektowego.

*) 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	15			30		h
3.	Inne (konsultacje, egzamin)*	2			4		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	51					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,04					ECTS
6.	Liczba godzin samodzielnej pracy studenta	24					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,96					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	30					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,20					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	75					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3					

* 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.
6. Pasierbiński J., Zbysiński P.: „Układy programowalne w praktyce”, Wydawnictwa Komunikacji i Łączności, Warszawa 2001.

Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje