



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-EM-06-01-S6
Nazwa przedmiotu	Sterowniki wbudowane w systemach elektromobilnych
Nazwa przedmiotu w języku angielskim	Embedded Systems in Electromobility
Obowiązuje od roku akademickiego	2020/21

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	
Poziom kształcenia	
Profil studiów	
Forma i tryb prowadzenia studiów	
Zakres	
Jednostka prowadząca przedmiot	
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	
Status przedmiotu	
Język prowadzenia zajęć	
Usytuowanie modułu w planie studiów - semestr	
Wymagania wstępne	
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	30	0	30	0	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.	EM1_W11 EM1_W14
	W02	Zna budowę/ struktury mikroprocesorów i układów logiki programowalnej.	EM1_W11 EM1_W14
	W03	Zna zagadnienia programowania/konfiguracji mikroprocesorów i układów logiki programowalnej.	EM1_W11 EM1_W14
	W04	Zna różne formy opisu układów cyfrowych.	EM1_W11 EM1_W14
	W05	Zna języki programowania systemów cyfrowych.	EM1_W11 EM1_W14
	W06	Zna techniki sprzęgania układów logiki programowalnej z systemami mikroprocesorowymi i układami pomiarowymi oraz wykonawczymi.	EM1_W11 EM1_W14
Umiejętności	U01	Potrafi dokonać dekompozycji funkcji systemu cyfrowego na podukłady.	EM1_U02
	U02	Potrafi programować złożone systemy cyfrowe.	EM1_U03
	U03	Potrafi łączyć w funkcjonalną całość różne formy opisu działania podukładów systemów cyfrowych.	EM1_U03 EM1_U06
	U04	Potrafi zarządzać projektem systemu cyfrowego z użyciem środowiska do projektowania systemów logiki programowalnej.	EM1_U04 EM1_U06
	U05	Potrafi dokonać weryfikacji poprawności projektu systemu cyfrowego z użyciem narzędzi symulacyjnych.	EM1_U10
	U06	Potrafi uruchamiać systemy cyfrowe mikroprocesorowe i logiki programowalnej.	EM1_U03 EM1_U04
	U07	Potrafi posługiwać się sprzętem pomiarowym w celu uruchamiania układów mikroprocesorowych i logiki programowalnej.	EM1_U09
	U08	Potrafi czytać dokumentację, schematy systemów cyfrowych.	EM1_U08
	U09	Potrafi opracowywać dokumentację projektu systemu cyfrowego z użyciem narzędzi dostarczanych przez środowisko projektowe.	EM1_U08 EM1_U15
	U10	Potrafi pracować w zespole projektowym ale również samodzielnie realizować przydzielone zadania projektowe.	EM1_U14
	U11	Potrafi stosować normy bezpieczeństwa w projektach oraz w praktyce zawodowej.	EM1_U12 EM1_U16
	U12	Potrafi samodzielnie poszerzać i uaktualniać swoją wiedzę zarówno z zakresu systemów cyfrowych jak i dziedzin, w których je stosuje.	EM1_U13
Kompetencje społeczne	K01	Rozumie odpowiedzialność projektanta systemu cyfrowego za bezpieczeństwo jego użytkowników.	EM1_K02
	K02	Rozumie i uwzględnia w swojej działalności projektowej bezpieczeństwo systemów cyfrowych stosując zabezpieczenia przed nieautoryzowanym dostępem do funkcji systemu.	EM1_K04
	K03	Rozumie konieczność ciągłego poszerzania wiedzy i umiejętności w związku z koniecznością rozumienia działania obiektu sterowania oraz bardzo szybkim rozwojem dziedziny systemów cyfrowych.	EM1_K01
	K04	Rozumie związki i wpływ systemów cyfrowych na środowisko i inne dziedziny techniki.	EM1_K02

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	Czym są systemy wbudowane i 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. 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.
	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 koprocessorów 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.
	Analiza przykładowych projektów (do wyboru w konsultacji ze studentami oraz innymi prowadzącymi – tworzenie narzędzi do innych projektów): • 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 warunkowego pod adres bezwzględny, 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, • Sterowniki różnych typów przetwornic impulsowych, • Programowalny 3-fazowy modulator wektora przestrzennego napięcia SVPWM, • Sterowniki przekształtników z miękką komutacją.
laboratorium	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 laboratoryjnych 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.

*) 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		X	
U02			X		X	
U03			X		X	
U04			X		X	
U05					X	
U06					X	
U07					X	
U08			X		X	
U09			X		X	
U10					X	
U11			X		X	
U12			X		X	
K01			X		X	
K02			X		X	
K03			X		X	
K04			X		X	

A.

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład		Zaliczenie kolokwium
laboratorium		Poprawna realizacja zadań laboratoryjnych.

*) zostawić tylko realizowane formy zajęć

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	h
		30	0	30			
3.	Inne (konsultacje, egzamin)*	4		2			h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	66					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,64					ECTS
6.	Liczba godzin samodzielnej pracy studenta	34					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,36					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	100					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	4					

* 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