

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Reprogramowalne systemy wbudowane		Kod 1010515331010540118
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 3
Ścieżka obieralności/specjalność Aplikacje mobilne i wbudowane dla	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 12 Ćwiczenia: - Laboratoria: 20 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 4 100%
Odpowiedzialny za przedmiot / wykładowca:		
<p>dr inż. Marek Kropidłowski email: Marek.Kropidlowski@put.poznan.pl tel. 61 6652297 Katedra Inżynierii Komputerowej ul. Piotrowo 3a, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu elektroniki analogowej i cyfrowej oraz programowania strukturalnego.
2	Umiejętności:	Powinien posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
<ol style="list-style-type: none"> Przekazanie studentom wiedzy związanej z nowoczesnymi układami reprogramowalnymi oraz ich zastosowaniem w systemach cyfrowych i platformach sprzętowych dla systemów IoT. Zapoznanie studentów z nowoczesnymi metodami projektowania, testowania i prototypowania urządzeń elektronicznych bazujących na układach reprogramowalnych. Rozwijanie u studentów umiejętności rozwiązywania złożonych problemów projektowych w zakresie systemów cyfrowych. Kształtowanie u studentów umiejętności pracy zespołowej. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> Ma szczegółową wiedzę w zakresie wybranych działów elektroniki i telekomunikacji - [K_W3] Ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki - [K_W5] Ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych - [K_W6] Ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych i programowych - [K_W7] Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki - [K_W8] Ma podstawową wiedzę dotyczącą transferu technologii w odniesieniu do rozwiązań informatycznych - [K_W15] 		
Umiejętności:		

<p>1. Potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K_U1]</p> <p>2. Potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K_U9]</p> <p>3. Potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12]</p> <p>4. Potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]</p> <p>5. Potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych - [K_U21]</p> <p>6. Potrafi - zgodnie z zadaną specyfikacją, uwzględniając aspekty pozatechniczne - zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt - co najmniej w części - używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K_U27]</p>
Kompetencje społeczne:
<p>1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]</p> <p>2. Zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K_K4]</p> <p>3. Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]</p>

Sposoby sprawdzenia efektów kształcenia
<p>Ocena formująca:</p> <p>a) w zakresie wykładów:</p> <ul style="list-style-type: none">- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach, <p>b) w zakresie laboratoriów / ćwiczeń:</p> <ul style="list-style-type: none">- na podstawie oceny bieżącego postępu realizacji zadań, <p>Ocena podsumowująca:</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- z przedmiotu nie zaplanowano egzaminu. Sprawdzenie efektów kształcenia odbędzie się poprzez kolokwium przeprowadzone na przedostatnim wykładzie. <p>b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych poprzez sprawdzenie przygotowania zadanych projektów/ćwiczeń oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,- ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,- ocenę dokumentacji tworzonej systematycznie wraz z postępami prac projektowych; dokumentacja przygotowywana częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,- ocenę i obronę przez studenta sprawozdania z realizacji projektu, <p>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:</p> <ul style="list-style-type: none">- omówienia dodatkowych aspektów zagadnienia,- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,- uwagi związane z udoskonaleniem materiałów dydaktycznych,- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.
Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Specyfika dziedziny CAD/EDA, dostępne narzędzia programowe, metody projektowania w dziedzinie układów cyfrowych. Zakres zastosowań współczesnych języków opisu sprzętu - perspektywy rozwoju. Język VHDL wg normy IEEE 1076-2008. Podstawowe konstrukcje języka: jednostka projektowa, architektura, sygnały, atrybuty, procesy. Symulator języka VHDL - sposób działania, ?czas delta? a realizacja współbieżności w modelowaniu systemów.

Podzbiór syntezowalny języka VHDL wg normy IEEE 1076-2008. Zasady tworzenia opisu na poziomie RTL (tzw. HDL 'good practice'). Metody projektowania układów kombinacyjnych i sekwencyjnych z wykorzystaniem języków opisu sprzętu. Zasady projektowania i implementacji automatów. Zaawansowane elementy języka, pliki, procedury, funkcje, asercje. Budowa złożonego środowiska testowego (testbench), testbench z automatyczną weryfikacją. Parametryzacja kodu, konstrukcje ?generic? i ?generate?.

Synteza i implementacja urządzeń cyfrowych na platformach programowalnych FPGA i CPLD. Zależność opisu HDL od platformy implementacji (FPGA vs. ASIC). Konstrukcja nowoczesnych układów rekonfigurowalnych typu FPGA/CPLD/FPGA/FPOA/3D-PLD/PSoC?. Rola komponentów IP (Intellectual Property) w projektowaniu złożonych systemów cyfrowych. Wykorzystanie rekonfiguracji zdalnej (IRL), częściowej i dynamicznej (PDR) w cyfrowych systemach sterowania.

Systemy z rekonfiguracją dynamiczną: modele pamięci konfiguracji, metody projektowania, DRP, ICAP, wykorzystanie w systemach wbudowanych; studium przypadku: Driver assistance system.

Softprocesory i platformy rekonfigurowalne dla systemów wbudowanych: przegląd rozwiązań, metody i narzędzia implementacji [studium przypadku: procesor LEON3], systemy hybrydowe, układy AP SoC [studium przypadku: Xilinx Zynq].

Metody testowania i weryfikacji prototypów urządzeń cyfrowych na platformach FPGA: scenariusze weryfikacji prototypów, OCI (On-Chip Instrumentation), metody testowania zdalnego, symulacja z modelem funkcjonalnym (BFM), debugowanie w systemie (ISD), emulacja (QEMU, QBox), ko-symulacja (HIL).

Zajęcia laboratoryjne prowadzone są w formie dziewięciu 2-godzinnych spotkań, odbywających się w laboratorium, poprzedzonych sesją instruktorską na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów. Program laboratorium obejmuje następujące zagadnienia:

Opis syntezowalny układów kombinacyjnych i sekwencyjnych. Badanie wpływu stosowanego stylu opisu na wyniki syntezy. Konstruowanie zaawansowanych środowisk testowania (testbench). Wykorzystanie techniki OCI do weryfikacji prototypów urządzeń cyfrowych FPGA. Projektowanie i implementacja urządzeń cyfrowych z wykorzystaniem oprogramowania firm Mentor Graphics, Xilinx, Lattice, Aldec. Realizacja wybranych sterowników sprzętowych urządzeń peryferyjnych stosowanych we współczesnych systemach automatyki (BPI, SPI, I2C, UART, USB, w pełni cyfrowe realizacje przetworników AC/CA, sensory z interfejsem cyfrowym, LCD, PWM).

Część wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Metody dydaktyczne:

1. wykład:

prezentacja multimedialna uzupełniona przykładami podawanymi na tablicy, rozwiązywanie zadań, studium przypadków.

2. ćwiczenia laboratoryjne:

ćwiczenia praktyczne, dyskusja, praca w zespole, zawody projektowe, demonstracja projektowanych urządzeń.

Literatura podstawowa:

1. Piotr Zbysiński, Jerzy Pasierbiński, Układy programowalne pierwsze kroki, Wydanie II, Wydawnictwo BTC, Warszawa 2004, ISBN: 83-910067-0-0
2. Steve Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization, John Wiley & Sons, June 2007, ISBN: 0470054379.

Literatura uzupełniająca:

1. Peter J. Ashenden, The Designer's Guide to VHDL / Edition 3, Elsevier Science, June 2008, ISBN: 0120887851.
2. Mike Kuniavsky, Smart Things: Ubiquitous Computing User Experience Design, Elsevier Science & Technology, September 2010, ISBN: 0123748992.
3. Michael Keating, Pierre Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs / Edition 3, Springer-Verlag New York, August 2007, ISBN: 0387740988
4. Richard Munden, ASIC and FPGA Verification: A Guide to Component Modeling (Systems on Silicon), Elsevier Inc. 2005, ISBN: 0-12-510581-9. 4. Scott Hauck, Andre DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation / Edition 1, Elsevier Science, November 2007, ISBN: 0123705223

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
----------	--------------

1. udział w zajęciach laboratoryjnych:	20
2. przygotowanie do ćwiczeń	8
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych	10
4. udział w konsultacjach związanych z realizacją procesu kształcenia (mogą być realizowane drogą elektroniczną)	4
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	20
6. przygotowanie do testu (6godz)	6
7. udział w wykładach	12
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.)	20
Obciążenie pracą studenta	
forma aktywności	godzin
ECTS	
Łączny nakład pracy	100
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36
Zajęcia o charakterze praktycznym	58