



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Projektowanie układów reprogramowalnych

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Mikrosystemy informatyczne

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

15

Laboratoria

30

Inne (np. online)

Ćwiczenia

Projekty/seminaria

15

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Marek Kropidłowski

email: marek.kropidlowski@put.poznan.pl

tel. tel. 616652297

Instytut Informatyki

ul. Piotrowo 3 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Paweł Śniatała

email: pawel.sniatala@put.poznan.pl

tel. tel. 616652297

Instytut Informatyki

ul. Piotrowo 3 60-965 Poznań

Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu elektroniki analogowej i cyfrowej oraz programowania strukturalnego. Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

- 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.



Przedmiotowe efekty uczenia się

Wiedza

1. Ma zaawansowaną i pogłębioną wiedzę z zakresu szeroko rozumianych systemów informatycznych oraz metod i narzędzi wykorzystywanych do ich implementacji, szczególnie dotyczących budowania warstwy sprzętowej systemów reprogramowalnych - [K2st_W1]
2. Ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu informatyki, szczególnie dotyczącą konstruowania systemów wbudowanych - [K2st_W3]
3. Ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów informatycznych, szczególnie warstwy sprzętowej systemów - [K2st_W5]

Umiejętności

1. Potrafi przy formułowaniu i rozwiązywaniu zadań inżynierskich integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]
2. Potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]
3. Potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8]
4. Potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K2st_U9]
5. Potrafi zgodnie z zadaną specyfikacją, zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]

Kompetencje społeczne

1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]
2. Rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [K2st_K2]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów: na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów: na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:



a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez egzamin ustny połączony z obroną projektu, w przypadku wątpliwości część pisemna (test w postaci elektronicznej na platformie Moodle);

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez sprawdzian projektowy i ocenę zadań realizowanych w ramach każdego spotkania laboratoryjnego;

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

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/FPAA/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 2-godzinnych spotkań, odbywających się w laboratorium, poprzedzonych sesją instruktażową na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów. Program laboratorium obejmuje następujące zagadnienia:

Opis syntezy 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.

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



Metody dydaktyczne

1. wykład: prezentacja multimedialna uzupełniona przykładami podawanymi na tablicy.
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, dyskusja, praca w zespole, zawody projektowe.
3. zajęcia projektowe: studium przypadków, omawianie i demonstracja działania projektowanych urządzeń

Literatura

Podstawowa

1. Mark Zwoliński, Projektowanie układów cyfrowych z wykorzystaniem języka VHDL, WKŁ2007, ISBN: 9788320616354.
2. Andrew Rushton, VHDL for Logic Synthesis, Third Edition, John Wiley & Sons, 2011, ISBN: 978-0-470-68847-2.
3. Scott Hauck, Andre DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation / Edition 1, Elsevier Science, November 2007, ISBN: 0123705223

Uzupełniająca

1. Michael Keating, Pierre Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs / Edition 3, Springer-Verlag New York, August 2007, ISBN: 0387740988.
2. Peter J. Ashenden, The Designer's Guide to VHDL / Edition 3, Elsevier Science, June 2008, ISBN: 0120887851.
3. Steve Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization, John Wiley & Sons, June 2007, ISBN: 0470054379.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
łączy nakład pracy	120	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do sprawdzianu/egzaminu, wykonanie projektu) ¹	58	2,0

¹ niepotrzebne skreślić lub dopisać inne czynności