



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowalne układy cyfrowe [S1EiT1E>PUC]

Przedmiot

Kierunek studiów

Elektronika i telekomunikacja/Electronics and Telecommunications

Rok/Semestr

3/5

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

5,00

Koordynatorzy

dr hab. inż. Olgierd Stankiewicz prof. PP
olgierd.stankiewicz@put.poznan.pl

Wykładowcy

Wymagania wstępne

Posiada podstawową wiedzę z algebry Boole'a. Posiada wiedzę z zakresu programowania w językach C/C++. Posiada ogólną wiedzę na temat cyfrowych układów kombinacyjnych i sekwencyjnych. Posiada ogólną wiedzę z zakresu arytmetyki binarnej i cyfrowej reprezentacji sygnałów. Umie wyszukiwać informacje potrzebne w procesie projektowania i w razie potrzeby korzystać z kursów edukacyjnych, zwłaszcza za pośrednictwem Internetu i kształcenia na odległość. Zna ograniczenia własnej wiedzy i umiejętności; potrafi precyzyjnie formułować pytania; rozumie potrzebę dalszego kształcenia i systematycznego czytania czasopism naukowych w danej dziedzinie. Potrafi pracować indywidualnie i w zespole; wie, że jest odpowiedzialny za zadania realizowane w zespole.

Cel przedmiotu

361 / 5 000 Głównym celem kursu jest przedstawienie różnych technik projektowania układów cyfrowych, które mogą być odpowiednie dla układów ASIC/FPGA. Jako język opisu sprzętu zostanie wykorzystany Verilog. Wiele przykładów pokaże, jak efektywnie wykorzystać wszystkie podstawowe i ogólne bloki FPGA (takie jak RAM, DSP itp.). Ćwiczenia laboratoryjne będą wykonywane z wykorzystaniem układów FPGA.

Przedmiotowe efekty uczenia się

Wiedza:

Student posiada podstawową umiejętność projektowania prostych urządzeń cyfrowych.

Student ma podstawową wiedzę na temat zasady działania szybkich interfejsów komunikacyjnych.

Student posiada podstawową wiedzę z zakresu projektowania maszyn stanowych.

Umiejętności:

Potrafi opisać złożony system cyfrowy w postaci hierarchii modułów przy użyciu języka Verilog.

Potrafi poprawnie wyznaczyć parametry interfejsu pomiędzy dwiema dziedzinami częstotliwości.

Potrafi pozyskiwać dane z literatury i innych źródeł, potrafi je integrować, dokonywać ich interpretacji oraz formułować i uzasadniać opinie.

Kompetencje społeczne:

Potrafi obserwować i analizować rozwój technik projektowania.

Możliwość samodzielnego uczenia się (podręczniki, programy komputerowe).

Znajomość odpowiedzialności za projektowane systemy elektroniczne i telekomunikacyjne.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Przedstawione powyżej efekty kształcenia weryfikowane są w następujący sposób:

Wykład: egzamin pisemny

Egzamin pisemny składa się z zestawu 6-10 pytań problemowych, na które oczekuje się odpowiedzi opisowej.

Każda odpowiedź punktowana jest od 0 do 1 punktu (możliwe są również punkty ułamkowe).

Egzamin jest zdany, jeśli liczba uzyskanych punktów przekroczy 50%. Powyżej 50% oznacza, że powyższa wiedza jest na poziomie zadowalającym.

Zagadnienia przedmiotowe, do których przygotowane są pytania, są przesyłane studentom drogą mailową za pośrednictwem

uczelnianego systemu poczty elektronicznej.

Laboratoria:

Aktywność na zajęciach, sprawozdania z poszczególnych zajęć. Projekt laboratoryjny realizowany indywidualnie/w małych grupach.

Treści programowe

Wykład:

Wprowadzenie do cyfrowych układów programowalnych. Ogólna struktura układów FPGA. Podstawowe bloki wbudowane (RAM, PLL, FIFO itp.). Porównanie różnych układów FPGA. Komunikacja międzydomenowa (interfejs synchroniczny ze źródłem). Szybkie interfejsy I/O – wykorzystanie modułów Gigabit GTP, GTX, GTH w HD-SDI, SATA, PCI-E i SerDes. Systemy w układzie (SoC). Języki programowania – Verilog, SystemC, SystemVerilog, migen, MyHDL. Zasady dobrego programowania, kod samoopisujący.

Metody i narzędzia do symulacji i syntezy projektów na układach FPGA – generowanie plików EDIF, partycjonowanie projektów, język skryptowy Python.

Przykłady efektywnej implementacji wybranych algorytmów (konwersja DCT, konwersja przestrzeni barw RGB-YUV, elementaryzacja, mnożenie złożone, operacje zmiennoprzecinkowe) dla układów FPGA.

Laboratoria:

Schemat i oprogramowanie do symulacji i syntezy. Podstawowe stanowiska testowe. Projektowanie różnych modułów:

generatorów liczb losowych, konwerterów liczb dziesiętnych kodowanych binarnie, modułów buforowych (np. stosu, FIFO),

jednostek arytmetycznych, maszyn stanowych. Projektowanie systemu zawierającego maszyny stanowe, np. demonstracja

sygnalizacji świetlnej lub kalkulatora równań w notacji infiksowej/postfiksowej.

Tematyka zajęć

Wykład:

Wprowadzenie do cyfrowych układów programowalnych. Ogólna struktura układów FPGA. Podstawowe

bloki wbudowane (RAM, PLL, FIFO itp.). Porównanie różnych układów FPGA. Komunikacja międzydomenowa (interfejs synchroniczny ze źródłem). Szybkie interfejsy I/O – wykorzystanie modułów Gigabit GTP, GTX, GTH w HD-SDI, SATA, PCI-E i SerDes. Systemy w układzie (SoC). Języki programowania – Verilog, SystemC, SystemVerilog, migen, MyHDL. Zasady dobrego programowania, kod samoopisujący.

Metody i narzędzia do symulacji i syntezy projektów na układach FPGA – generowanie plików EDIF, partycjonowanie projektów, język skryptowy Python.

Przykłady efektywnej implementacji wybranych algorytmów (konwersja DCT, konwersja przestrzeni barw RGB-YUV, elementaryzacja, mnożenie złożone, operacje zmiennoprzecinkowe) dla układów FPGA.

Laboratoria:

Schemat i oprogramowanie do symulacji i syntezy. Podstawowe stanowiska testowe. Projektowanie różnych modułów:

generatorów liczb losowych, konwerterów liczb dziesiętnych kodowanych binarnie, modułów buforowych (np. stosu, FIFO),

jednostek arytmetycznych, maszyn stanowych. Projektowanie systemu zawierającego maszyny stanowe, np. demonstracja

sygnalizacji świetlnej lub kalkulatora równań w notacji infiksowej/postfiksowej.

Metody dydaktyczne

Wykład: prezentacja multimedialna z przykładami prezentowanymi na tablicy.

Laboratoria: praca na komputerach z oprogramowaniem do symulacji i syntezy. Wykorzystanie układów FPGA.

Przykłady zilustrowane na ekranie/tablicy.

Literatura

Podstawowy

Skahill K., VHDL dla logiki programowalnej / Język VHDL , WNT, SBN-13: 978-0201895735, ISBN-10: 0201895730.

Giovanni De Micheli, Synteza i optymalizacja obwodów cyfrowych / Synteza i optymalizacja regulacji inny , WNT, ISBN-13: 978-0070163331 ISBN-10: 0070163332.

Dodatkowy

Łuba T., Rawski M., Tomaszewicz P., Zbierzchowski B., Synteza żywotna, Wydawnictwa Komunikacji i Łączności, Warszawa 2003.

Hajduk Z., Wprowadzenie do języka Verilog, BTC, Warszawa 2009.

Kamionka-Mikuła H., Małysiak H., Pochopień B., Synteza i analiza analityczna, WKŁ.Zbysiński P., Pasierbiński J.: Układy programowe pierwsze kroków, Wydawnictwo BTC, Warszawa 2004,

Łuba T.. : Synteza błędnych logicznych. Oficyna Wyd. PW, Warszawa, 2005.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	70	3,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	55	2,00