

KARTA PRZEDMIOTU

Nazwa przedmiotu: Algorytmiczne metody syntezy układów cyfrowych (TIAu>SM2-AMSUC19)

Nazwa w języku polskim:

Nazwa w jęz. angielskim: Logic synthesis of digital circuits

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

EGZ

Język wykładowy:

polski

Strona WWW:

<https://platforma2.polsl.pl/rau3/course/view.php?id=342>

Skrócony opis:

Synteza logiczna, języki HDL, symulacja

Celem wykładów jest przedstawienie teoretycznych podstaw algorytmicznych metod syntezy układów cyfrowych oraz praktycznych aspektów procesu projektowania układów, realizowanych w strukturach programowalnych. Celem zajęć laboratoryjnych i projektowych jest zdobycie umiejętności korzystania z języków opisu sprzętu i posługiwania się narzędziami wspomagającymi proces projektowania.

Zajęcia mają charakter kontaktowy

Wymagane jest wstępne przygotowanie z zakresu techniki cyfrowej

Opis:

ECTS: 3

Suma godzin: 75h (kontaktowo 45h / praca własna 30h)

Wykład: 15

Laboratoria: 30

Praca własna: zapoznanie się z literaturą, przygotowanie do egzaminu

Wykład:

1. Różnorodne formy implementacji układów cyfrowych; układy ASIC: full-custom, Gate Array, Standard Cells, PLD; SoC, różnice, zakres zastosowań.
2. Klasyfikacja układów programowalnych, etapy syntezy, podstawowe problemy w powiązaniu z poszczególnymi architekturami układów programowalnych
3. Etapy i algorytmy syntezy logicznej:
 - problemy związane z minimalizacją:
 - minimalizacja dwupoziomowa, minimalizacja wielopoziomowa
 - strategię minimalizacji ukierunkowane na różne architektury układów,
 - wykorzystanie specyficznych zasobów układowych w powiązaniu z minimalizacją iloczynów
 - przykłady opisu i minimalizacji prostych układów kombinacyjnych przeprowadzonych dla różnych architektur układów programowalnych,
 - problemy związane z podziałem projektowanego układu na poszczególne bloki logiczne o zadanej architekturze:
 - metody ekspansji liczby iloczynów,
 - podstawowe modele dekompozycji funkcji logicznych: dekompozycja funkcji słabo określonych, dekompozycje złożone pojedynczych funkcji i zespołów funkcji logicznych, dekompozycja rozłączna/nierozłączna funkcji logicznych,
 - algorytmy poszukiwania efektywnej dekompozycji funkcji (elementy teorii grafów, macierze sąsiedztwa, kliki, algorytmy analizy grafu zgodności, szybkie algorytmy kolorowania wierzchołków grafu niekompatybilności kolumn matrycy podziałów, funkcja pokrycia - algorytmy rozwiązywania problemów pokrycia itp.),
 - binarne diagramy decyzyjne, algorytmy redukcji liczby wierzchołków, algorytmy dekompozycji wyrażeń opisanych za pomocą ROBDD
 - przykłady prostych projektów wskazujących na znaczenie dekompozycji w syntezie logicznej
 - podstawowe problemy związane z odwzorowaniem technologicznym projektowanego układu (technology mapping)
 - wykorzystanie bramki XOR, ekspanderów, buforów trójstanowych,
 - wybrane problemy syntezy automatów sekwencyjnych:
 - metody kodowania stanów wewnętrznych automatów sekwencyjnych,
 - aktywność wyjścia a proces kodowania stanów,
 - kodowanie stanów, a minimalizacja poboru mocy,
 - przykłady opisu i syntezy prostych układów sekwencyjnych,
4. Metodyka projektowania układów z wykorzystaniem narzędzi wspomagających projektowanie układów cyfrowych na bazie struktur programowalnych - przykłady prostych projektów.

Laboratorium

W ramach laboratorium studenci praktycznie poznają i ćwiczą metody projektowania układów cyfrowych implementowanych w strukturach programowalnych, ze szczególnym uwzględnieniem języków opisu sprzętu i z wykorzystaniem zaawansowanych narzędzi CAD. Całość zajęć prowadzona jest na specjalizowanych modelach uruchomieniowych, na których bezpośrednio można weryfikować wynik syntezy odpowiednio opisanego układu.

Literatura:

Wykład:

1. Materiały podstawowe - wykład + materiały na platformie zdalnej edukacji
2. Kania D. Układy Logiki Programowalnej, PWN 2012
3. T. Łuba, B. Zwierchowski, Komputerowe projektowanie układów cyfrowych, WKiŁ, 2000
4. T. Łuba, Synteza układów cyfrowych, WKiŁ, 2003

Laboratorium, projekt:

1. Materiały podstawowe - platforma zdalnej edukacji
2. Pasierbiński J., Zbysiński P., Układy programowalne w praktyce, WKŁ, 2002
3. Kalisz J. (red.), Język VHDL w praktyce, WKŁ, 2002
4. Skahill K., Język VHDL. Projektowanie programowalnych układów logicznych, WNT, 2001
5. Palnitkar S., Verilog HDL. A Guide to Digital Design and Synthesis, Prentice Hall, 2003
6. Lee W.F., Verilog Coding for Logic Synthesis. John Wiley & Sons Inc., 2003

Efekty uczenia się:

Wiedza: Zna i rozumie

K2A_W01 metody syntezy i weryfikacji systemów cyfrowych (kolokwium)

K2A_W03 języki opisu sprzętu (wykonanie ćwiczenia lab.)

Umiejętności: Potrafi

K2A_U02 tworzyć projekty systemów cyfrowych (wykonanie ćwiczenia lab.)

K2A_U16 zaprojektować układ cyfrowy, wykonując jego opis w języku opisu sprzętu, przeprowadzić syntezę i weryfikację poprzez opisanie testbench'a (wykonanie ćwiczenia lab.)

Metody i kryteria oceniania:

Ocena końcowa to ocena z egzaminu. Warunkiem dopuszczenia do egzaminu jest zaliczenie wszystkich zadań z lab.

Sylabus obowiązuje od roku akademickiego 2025/2026, a jego zawartość nie podlega zmianom w trakcie trwania semestru

Punkty przedmiotu w cyklach:

<bez przypisanego programu>			
Typ punktów		Liczba	Cykl pocz.
Europejski System Transferu Punktów (ECTS)		3	2020/2021-Z