

Nazwa modułu: **Projektowanie Systemów Cyfrowych**

Rocznik: **2012/2013** Kod: **JIS-2-205-s** Punkty ECTS: **5**

Wydział: **Fizyki i Informatyki Stosowanej** Poziom studiów: **Studia II stopnia**

Specjalność: **Systemy wbudowane i rekonfigurowalne** Kierunek: **Informatyka Stosowana**

Semestr: **2** Profil kształcenia: **Ogólnoakademicki (A)** Język wykładowy: **Polski**

Forma i tryb studiów: **Stacjonarne** Strona www: **<http://fatcat.agh.edu.pl/~skoczen/psc>**

Osoba odpowiedzialna: **dr inż. Skoczeń Andrzej (skoczen@fis.agh.edu.pl)**

Osoby prowadzące: **dr inż. Skoczeń Andrzej (skoczen@fis.agh.edu.pl)
dr Świątek Krzysztof (swientek@agh.edu.pl)
dr inż. Mindur Bartosz (mindur@fis.agh.edu.pl)**

Opisy efektów kształcenia dla modułu

Kod EKM	Student, który zaliczył moduł wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W001	Student biegle zna syntaktykę języka Verilog i zadowalająco języka VHDL. Rozumie znaczenie wymienionych i innych języków opisu sprzętu w przebiegu projektowania urządzeń.	IS1A_W10, IS1A_W11, IS2A_W12	Aktywność na zajęciach, Udział w dyskusji, Kolokwium
M_W002	Student zna i rozumie podstawowe pojęcia omawiane w ramach wykładu dotyczące analizy i syntezy kombinacyjnych i sekwencyjnych układów cyfrowych. Zna również podstawowe sprzętowe architektury układów sterowania i ścieżki przetwarzania danych.	IS2A_W10, IS2A_W11, IS2A_W12	Aktywność na zajęciach, Udział w dyskusji
M_W003	Student dysponuje aktualną wiedzą na temat technologii produkcji cyfrowych układów scalonych wielkiej skali integracji ASIC i FPGA (CPLD). Zna również strukturę bibliotek komórek standardowych stosowanych w projektowaniu urządzeń cyfrowych.	IS2A_W10, IS2A_W11, IS2A_W12	Aktywność na zajęciach, Wykonanie ćwiczeń laboratoryjnych, Kolokwium
Umiejętności			
M_U001	Student potrafi zbudować zarówno funkcjonalny jak i syntezywalny model sprzętu. Potrafi również przeprowadzić symulację zbudowanego modelu i analizować jej wyniki.	IS2A_U09, IS2A_U15	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Kolokwium, Projekt
M_U002	Student potrafi stworzyć prostą aplikację sprzętową sterowania i/lub przetwarzania danych.	IS2A_U09, IS2A_U15	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Kolokwium, Sprawozdanie
Kompetencje społeczne			
M_K001	Student potrafi pracować w zespole projektowym. Potrafi samodzielnie zdobyć odpowiednią wiedzę i umiejętności niezbędne do realizacji jego części zadania zespołowego.	IS2A_K02	Wykonanie projektu, Sprawozdanie

M_K002	Student umie przedstawić wykonany projekt w sposób komunikatywnej prezentacji. Potrafi także wskazać obszary zastosowań tworzonych aplikacji i ekonomiczne aspekty zastosowanych rozwiązań.	IS2A_K02	Wykonanie projektu, Sprawozdanie
--------	---	----------	-------------------------------------

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł wie/umie/potrafi	Forma zajęć								
		Wykład	Ćwic. audyt.	Ćwic. lab.	Ćwic. proj.	Konw.	Zaj. sem.	Zaj. prakt.	Inne	E-learning
Wiedza										
M_W001	Student biegle zna syntaktykę języka Verilog i zadowalająco języka VHDL. Rozumie znaczenie wymienionych i innych języków opisu sprzętu w przebiegu projektowania urządzeń.	+	-	-	-	-	-	-	-	-
M_W002	Student zna i rozumie podstawowe pojęcia omawiane w ramach wykładu dotyczące analizy i syntezy kombinacyjnych i sekwencyjnych układów cyfrowych. Zna również podstawowe sprzętowe architektury układów sterowania i ścieżki przetwarzania danych.	+	-	-	-	-	-	-	-	-
M_W003	Student dysponuje aktualną wiedzą na temat technologii produkcji cyfrowych układów scalonych wielkiej skali integracji ASIC i FPGA (CPLD). Zna również strukturę bibliotek komórek standardowych stosowanych w projektowaniu urządzeń cyfrowych.	-	-	-	-	-	-	-	-	-
Umiejętności										
M_U001	Student potrafi zbudować zarówno funkcjonalny jak i synteżowalny model sprzętu. Potrafi również przeprowadzić symulację zbudowanego modelu i analizować jej wyniki.	+	-	+	+	-	-	-	-	-
M_U002	Student potrafi stworzyć prostą aplikację sprzętową sterowania i/lub przetwarzania danych.	+	-	+	+	-	-	-	-	-
Kompetencje społeczne										
M_K001	Student potrafi pracować w zespole projektowym. Potrafi samodzielnie zdobyć odpowiednią wiedzę i umiejętności niezbędne do realizacji jego części zadania zespołowego.	-	-	+	+	-	-	-	-	-
M_K002	Student umie przedstawić wykonany projekt w sposób komunikatywnej prezentacji. Potrafi także wskazać obszary zastosowań tworzonych aplikacji i ekonomiczne aspekty zastosowanych rozwiązań.	-	-	+	+	-	-	-	-	-

Treść modułu kształcenia (program wykładów i pozostałych zajęć)

Wykład

- 1. Język opisu sprzętu - narzędzie projektowania w mikroelektronice.**

Różne języki opisu i weryfikacji sprzętu oraz obszary ich zastosowań: HDL: Verilog, VHDL, Abel, AHDL; HDVL: SystemVerilog; SLDL: SystemC, Gezel; HVL: e, OpenVera, Sugar, ForSpec, PSL. Specyfika i porównanie z językami programowania. Automatyzacja procesu projektowania sprzętu elektronicznego. Obecne trendy.
- 2. Mikroelektronika i układy programowalne.**

Mikroelektronika: historia i przyszłość. Prawo Moore'a. Sukces technologii CMOS. Pojęcie maski i warstw technologicznych. Obecny stan rozwoju i wyzwania przed technologią CMOS. Technologia układów programowalnych PLD.
- 3. Przebieg procesu projektowania układu cyfrowego.**

Opceje projektowe. Poziomy abstrakcji opisu sprzętu i procesy przekształceń pomiędzy nimi. Projektowanie układu do specjalnych zastosowań (ASIC) i układu programowalnego (FPGA, CPLD). Przebieg projektu w środowisku ISE firmy Xilinx.
- 4. Teoretyczne podstawy układów logicznych.**

Algebra Boole'a, funkcje logiczne i ich reprezentacje, metody minimalizacji (metoda Quine'a-McKluskeya, metoda ESSPRESSO). Synteza logiczna a synteza wysokiego poziomu.
- 5. Podstawy Verilog-a.**

Konwencje językowe, typy danych, moduły, porty, konkretyzacja modułu, zadania, funkcje, dyrektywy kompilatora, typy bramek. Zadania i funkcje systemowe.
- 6. Modelowanie behavioralne w Verilog-u.**

Blok sekwencyjny i równoległy, przypisanie proceduralne blokujące i nieblokujące, sterowanie przebiegów czasowych, instrukcja warunkowa, rozgałęzienia, pętle. Przygotowanie modułu testującego.
- 7. Modelowanie RTL w Verilog-u.**

Modelowanie przepływu danych w Verilogu: przypisania ciągłe, opóźnienia, operatory. Modele symulacyjne a modele syntezywalne. Styl kodowania prowadzący do poprawnych wyników syntezy.
- 8. Automat skończony i mikroprogramowanie.**

Rodzaje automatu (FSM): Moore'a i Mealy'ego. Style zapisu automatu w Verilogu. Kodowanie stanów automatu. Eliminacja stanów równoważnych. Automat ze ścieżką danych FSM-D. Architektura mikroprogramowana. Lista rozkazów i assembler.
- 9. Inne konstrukcje Veriloga.**

Modelowanie na poziomie kluczy. Dynamiczna analiza czasowa. Symulacyjna biblioteka komórek standardowych.
- 10. Synteza logiczna i statyczna analiza czasowa.**

Lista konstrukcji syntetyzowalnych RTL. Biblioteki technologiczne komórek standardowych dla syntezy. Czasowe ograniczenia projektowe. Generacja i stosowanie przypisów czasowych SDF.
- 11. Wprowadzenie do VHDL-a.**

Jednostka projektowa: interfejs, architektura. Porty lokalne, formalne i rzeczywiste, Definiowanie typów danych. Pakiety i ich standaryzacja. Biblioteki. Funkcje i procedury.

12. Podstawy VHDL-a

Operatory, przypisania transportowe i inercyjne, instrukcje warunkowe, rozgałęziona i pętle. Przykłady syntezy kodu RTL w VHDL-u.

13. Symulacja analogowa i cyfrowo-analogowa.

Symulatory analogowe: SPICE, Verilog-A i cyfrowo-analogowe Verilog-AMS. Behawioralne modelowanie modułów analogowych.

14. Układy programowalne.

Struktury PLD: ROM, PLA, PAL. Modelowanie PLD w Verilogu. Technologia połączeń programowalnych. Porównanie CPLD i FPGA. Architektura układów rodziny Spartan-3x firmy Xilinx. Podstawowe bloki funkcjonalne, ich budowa i parametry.

15. Przegląd bloków funkcjonalnych elektroniki cyfrowej.

Rekomendacje projektowe. Przykłady kodów w Verilog-u i VHDL-u przydatne przy syntezie i testowaniu modeli urządzeń cyfrowych.

Ćwiczenia laboratoryjne**1. Symulacja behawioralna w Verilog-u**

Efekty kształcenia:

- student potrafi zapisać w Verilogu funkcjonalność urządzenia cyfrowego
- student potrafi posługiwać się środowiskiem projektowym ISE w stopniu pozwalającym na symulację modelu prostego cyfrowego bloku funkcjonalnego,
- student potrafi przygotować moduł testujący w Verilogu i interpretować symulacyjnie uzyskane przebiegi czasowe.

2. Synteza w oparciu o kod RTL w Verilogu i symulacja po syntezie

Efekty kształcenia:

- student potrafi zapisać w Verilogu syntezywalny model urządzenia cyfrowego.
- student potrafi w środowisku ISE przeprowadzić syntezę i symulację uzyskanego modelu strukturalnego.

3. Symulacja i synteza za pomocą języka VHDL

Efekty kształcenia:

- student potrafi zapisać w VHDL-u syntezywalny model urządzenia cyfrowego,
- student potrafi w środowisku ISE przeprowadzić syntezę i symulację uzyskanego modelu strukturalnego.

4. Implementacja automatu skończonego w układzie Spartan-3AN

Efekty kształcenia:

- student potrafi stworzyć model RTL automatu skończonego, przeprowadzić jego symulacyjną weryfikację, syntezę i implementację w układzie Spartan-3AN, tak by poprawność pracy automatu dała się sprawdzić za pomocą pokrętła, przycisków, przełączników, diod LED dostępnych na module ćwiczeniowym.

5. Implementacja sterownika wyświetlacza LCD

Efekty kształcenia:

- student potrafi zapisać syntezywalny model sterownika wyświetlacza LCD według zadanych diagramów,
- student potrafi przeprowadzić syntezę i implementację tego urządzenia.

6. Implementacja modułu asynchronicznej transmisji szeregowej UART

Efekty kształcenia:

- student potrafi zapisać syntezywalny model modułu asynchronicznej transmisji szeregowej

UART według zadanych diagramów,

- student potrafi przeprowadzić syntezę i implementację tego urządzenia.

Ćwiczenia projektowe

1. Projektowanie modułów arytmetycznych

Studenci w dwuosobowych zespołach realizują

Efekty kształcenia:

- student potrafi zinterpretować specyfikację modułu,
- student potrafi przeprowadzić podział potrzebnego sprzętu na część sterującą i ścieżkę danych,
- student potrafi zbudować syntezywalny model i przeprowadzić jego weryfikację i syntezę.

2. Projekt z układem Spartan-3AN

Studenci w dwuosobowych zespołach realizują projekty. Każdy zespół otrzymuje do wykonania

inny, przydzielony losowo projekt. Punktem wyjścia dla studentów są dostarczone założenia i wskazówki projektowe. W ramach projektu należy w oparciu o zasoby sprzętowe dostępne na module ćwiczeniowym z układem Spartan-3AN stworzyć działające urządzenie oraz dokumentację wykonania projektu.

Efekty kształcenia:

- student potrafi zinterpretować specyfikację urządzenia w celu zbudowania go za pomocą dostępnych narzędzi programowych,
- student potrafi efektywnie wykorzystać podstawowe narzędzia projektowe dostępne w środowisku ISE Xilinx do stworzenia prostej aplikacji sprzętowej na module ćwiczeniowym z układem Spartan-3AN,
- student potrafi współpracować w grupie realizując swoją część zadania,
- student potrafi zademonstrować funkcjonalność zbudowanego urządzenia zgodną z zadaną specyfikacją i przygotować dokumentację.

Sposób obliczania oceny końcowej

Ocena końcowa O z modułu obliczana jest jako średnia ważona oceny z kartkówki w laboratorium OL, kolokwium zaliczeniowego OK, i projektu OP:

$$O = 0.3 \cdot OL + 0.35 \cdot OK + 0.35 \cdot OP$$

Wymagania wstępne i dodatkowe

- Znajomość podstaw algebry Boole'a
- Znajomość podstaw elektroniki cyfrowej
- Podstawowa wiedza o strukturze języków programowania
- Przydatna jest znajomość składni języka C

Zalecana literatura i pomoce naukowe

- Materiały z wykładu dostępne na stronie przedmiotu: <http://fatcat.agh.edu.pl/~skoczen/hdl>
- Z. Hajduk, "Wprowadzenie do języka Verilog", Wydawnictwo BTC, 2009
- J. Majewski, P. Zbysiński, "Układy FPGA w przykładach", Wydawnictwo BTC, 2007
- IEEE Standard Verilog Hardware Description Language, IEEE Std 1364-2001, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00954909>
- IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-2008, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4772740>

Uwagi

Sposób i tryb wyrównania zaległości powstałych wskutek nieobecności studenta na zajęciach:

Pod koniec semestru przewidziany jest dodatkowy termin ćwiczeń (ogłaszany 2 tygodnie wcześniej na stronie internetowej przedmiotu i przez prowadzących), w którym można wykonać pomiary, których student z przyczyn losowych nie mógł wykonać w pierwotnym terminie. Studenci mogą wówczas odrabiać ćwiczenia po uprzednim uzyskaniu zgody prowadzącego zajęcia w jego grupie oraz odpowiedzi z części teoretycznej, potwierdzonej wpisem do protokołu. Obecność na wykładzie: zgodnie z Regulaminem Studiów AGH.

Zasady zaliczania zajęć:

W ramach laboratorium komputerowo-elektronicznego studenci pracując w dwuosobowych zespołach wykonują szereg ćwiczeń, które traktowane są jako niezbędne przygotowanie do wykonania projektu. Każde spotkanie w laboratorium rozpoczyna się od krótkiej kartkówki. Ocena, z tych kartkówki OL stanowi pierwszy składnik oceny końcowej O. Ostatnie spotkanie w laboratorium przeznaczone jest na komputerowe kolokwium zaliczeniowe polegające na samodzielnym zrealizowaniu prostego zadania projektowego. Ocena OK stanowi drugi składnik oceny końcowej O.

Projekt oceniany jest w oparciu o procent zrealizowanych założeń projektowych i/lub ocenę niezawodności działania stworzonego urządzenia. Dodatkowo oceniany jest sposób zaprezentowania informacji technicznych zawartych w opracowanej dokumentacji. Ocena OP stanowi trzeci składnik oceny końcowej O.

Zaliczenie laboratorium komputerowo-elektronicznego wymaga zaliczenia:

- wszystkich kartkówki,
- wszystkich projektów,
- kolokwium zaliczeniowego.

Warunkiem uzyskania zaliczenia z pojedynczego projektu jest:

- pomyślna prezentacja działania układu,
- poprawnie wykonane pomiary,
- zaliczone sprawozdanie - dokumentacja projektu.

Nakład pracy studenta (bilans punktów ECTS)

--	--

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	30 godz
Samodzielne studiowanie tematyki wykładów	20 godz
Udział w laboratoriach	28 godz
Udział w ćwiczeniach projektowych	15 godz
Kolokwium zaliczeniowe	2 godz
Przygotowanie do laboratoriów	15 godz
Samodzielna realizacja projektu	15 godz
Przygotowanie do kolokwium zaliczeniowego	20 godz
Sumaryczne obciążenie pracą studenta	145 godz
Punkty ECTS za moduł	5 ECTS