

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Reprogramowalne systemy wbudowane</b>		Kod <b>1010515331010510118</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>2 / 3</b>
Ścieżka obieralności/specjalność <b>Aplikacje mobilne i wbudowane dla</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>12</b> Ćwiczenia: - Laboratoria: <b>20</b> Projekty/seminaria: -		Liczba punktów <b>3</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) <b>kierunkowy z danego kierunku</b>		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>3 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Marek Kropidłowski email: marek.kropidlowski@put.poznan.pl tel. 616652297 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań		dr hab. inż. Paweł Śniatała email: pawel.sniatala@put.poznan.pl tel. 616652297 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu elektroniki analogowej i cyfrowej oraz programowania strukturalnego.
2	<b>Umiejętności:</b>	Student powinien posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu.
3	<b>Kompetencje społeczne</b>	W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
<ul style="list-style-type: none"> <li>- Przekazanie studentom wiedzy związanej z nowoczesnymi układami reprogramowalnymi oraz ich zastosowaniem w systemach cyfrowych i platformach sprzętowych dla systemów IoT.</li> <li>- Zapoznanie studentów z nowoczesnymi metodami projektowania, testowania i prototypowania urządzeń elektronicznych bazujących na układach reprogramowalnych.</li> <li>- Rozwijanie u studentów umiejętności rozwiązywania złożonych problemów projektowych w zakresie systemów cyfrowych.</li> <li>- Kształtowanie u studentów umiejętności pracy zespołowej.</li> </ul>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<ol style="list-style-type: none"> <li>1. Ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu informatyki szczególnie projektowania systemów wbudowanych i oprogramowania dla systemów z ograniczeniami sprzętowymi - [K2st_W2]</li> <li>2. Ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu informatyki, szczególnie dotyczącą konstruowania systemów wbudowanych - [K2st_W3]</li> <li>3. Ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki i pokrewnych dyscyplin naukowych, szczególnie dotyczącą nowoczesnych systemów cyfrowych - [K2st_W4]</li> <li>4. Zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich i prowadzeniu prac badawczych związanych z uruchamianiem i testowaniem dedykowanych systemów cyfrowych - [K2st_W6]</li> </ol>		
<b>Umiejętności:</b>		

1. Potrafi planować i przeprowadzać eksperymenty, w tym pomiary analizatorami stanów logicznych i symulacje systemów cyfrowych opisanych językami HDL, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy związane ze złożonymi problemami inżynierskimi i prostymi problemami badawczymi - [K2st\_U3]
2. Potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K2st\_U4]
3. 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]
4. Potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st\_U6]
5. Potrafi - stosując m.in. koncepcyjnie nowe metody - rozwiązywać złożone zadania informatyczne związane z rekonfiguracją dynamiczną warstwy sprzętowej urządzeń. - [K2st\_U10]
6. 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]

#### Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

a) w zakresie wykładów:

na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach.

b) w zakresie laboratoriów / ćwiczeń:

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 końcowego z zajęć laboratoryjnych, w przypadku wątpliwości realizowana część pisemna - test w formie elektronicznej na platformie Moodle;

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

sprawdzian projektowy i ocena zadań realizowanych w ramach spotkań laboratoryjnych;

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

- i. omówienia dodatkowych aspektów zagadnienia,
- ii. efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- iii. umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- iv. uwagi związane z udoskonaleniem materiałów dydaktycznych,
- v. wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

#### Treści programowe

Program wykładu obejmuje następujące zagadnienia:

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. Budowa złożonego środowiska testowego (testbench), testbench z automatyczną weryfikacją. Parametryzacja kodu, pakietów i bibliotek VHDL.

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 warstwie sprzętowej systemów wbudowanych.

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 instruktazową na początku semestru. Zadania realizowane są przez 2-osobowe zespoły studentów. Program laboratorium obejmuje następujące zagadnienia:

Badanie wpływu stosowanego stylu opisu na wyniki syntezy. Konstruowanie zaawansowanych środowisk testowania (testbench). Wykorzystanie techniki OCI do weryfikacji prototypów systemów cyfrowych FPGA. Projektowanie i implementacja urządzeń cyfrowych z wykorzystaniem oprogramowania firm Mentor Graphics, Xilinx. Realizacja wybranych sterowników sprzętowych dla systemów z interfejsem AXI, AXI-Lite, AXI-Stream.

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

**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

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

**Bilans nakładu pracy przeciętnego studenta**

<b>Czynność</b>	<b>Czas (godz.)</b>
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)	2 10
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	6
6. przygotowanie do testu (6godz)	12
7. udział w wykładach	10
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.)	

**Obciążenie pracą studenta**

<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	78	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1
Zajęcia o charakterze praktycznym	48	2