



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Specjalizowane układy obliczeniowe [S2Inf1-PB>SUOBL]

### Przedmiot

Kierunek studiów  
Informatyka

Rok/Semestr  
1/1

Studia w zakresie (specjalność)  
Przetwarzanie brzegowe

Profil studiów  
ogólnoakademicki

Poziom studiów  
drugiego stopnia

Język oferowanego przedmiotu  
polski

Forma studiów  
stacjonarne

Wymagalność  
obligatoryjny

### Liczba godzin

Wykład  
20

Laboratorium  
30

Inne  
0

Ćwiczenia  
0

Projekty/seminaria  
0

### Liczba punktów ECTS

4,00

### Koordynatorzy

dr inż. Marek Kropidłowski  
marek.kropidlowski@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu elektroniki 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 urządzeń brzegowych. - 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: test przeprowadzony na przedostatnim wykładzie lub w godzinach laboratoryjnych; test w postaci elektronicznej na platformie Moodle, w przypadku wątpliwości realizowana jest część ustna zaliczenia;

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 obejmuje projektowanie układów reprogramowalnych z wykorzystaniem języków opisu sprzętu.

## Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

Synteza i implementacja urządzeń cyfrowych na platformach programowalnych FPGA. Zależność opisu HDL od platformy implementacji (FPGA vs. ASIC). Konstrukcja nowoczesnych układów rekonfigurowalnych typu FPGA/CPLD/FPGA/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 brzegowych.

Systemy z rekonfiguracją dynamiczną: modele pamięci konfiguracji, metody projektowania. Softprocesory i platformy rekonfigurowalne dla systemów wbudowanych: przegląd rozwiązań, metody i narzędzia implementacji, systemy hybrydowe, układy AP SoC.

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:

Badanie wpływu stosowanego stylu opisu na wyniki syntezy. Konstruowanie zaawansowanych środowisk testowania (testbench). Projektowanie i implementacja urządzeń cyfrowych z wykorzystaniem oprogramowania firm Mentor Graphics, AMD-Xilinx, Altera-Intel.

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, 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 |
|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------|
| Łączny nakład pracy                                                                                                                                  | 100    | 4,00 |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem                                                                                            | 50     | 2,00 |
| Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu) | 50     | 2,00 |