Nazwa przedmiotu							
Architektura komputerów Computer architecture							
Dyscyplina							Oznaczenie przedmiotu
Informatyka							15_INFS1
Rodzaj przedmiotu	Stopień studiów	Tryb studiów		Język zajęć		Rok	Semestr
obowiązkowy	1	stacjonarne		polski / angielski		II	III
Rodzaj zajęć	Wyk.	Ćw.	Lab.	Sem.	Proj.	Liczba punktów ECTS	
Liczba godzin w semestrze	30	0	15	0	0	4	

Koordynator	Dr hab. inż. Sławomir Gryś, prof. PCz. grys@el.pcz.czest.pl
Prowadzący	Dr hab. inż. Sławomir Gryś, prof. PCz. grys@el.pcz.czest.pl Dr hab. inż. Stanisław Chudzik, prof. PCz. chudzik@el.pcz.czest.pl Asystent/Doktorant

I. KARTA PRZEDMIOTU

Cel przedmiotu	
C1.	Nabycie umiejętności przedstawiania danych w różnych formatach, ich konwersji, poznanie zasad arytmetyki komputerowej w powiązaniu z listą rozkazów procesora.
C2.	Zdobycie rozszerzonej wiedzy z zakresu budowy i organizacji komputera i jego poszczególnych elementów, tj. mikroprocesora, układów pamięciowych, układów wejścia-wyjścia.
C3.	Nabycie umiejętności klasyfikowania, porównywania i charakteryzowania podstawowych cech wybranej architektury komputera.
C4.	Nabycie podstawowych umiejętności programowania mikroprocesorów w języku niskiego poziomu.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji	
1.	Podstawowa wiedza z matematyki i rozszerzona z układów logicznych.
2.	Wiedza z podstaw programowania i struktur danych.
3.	Umiejętność korzystania ze źródeł literaturowych i zasobów internetowych, w tym w języku angielskim w stopniu wystarczającym do czytania literatury i specyfikacji technicznych, tzw. „datasheets”.

Efekty kształcenia	
EK1.	Student rozumie zasadę działania komputera jako całości i jego poszczególnych elementów, zna formaty liczb i zasady arytmetyki komputerowej oraz potrafi określić przeznaczenie, funkcjonalność i ograniczenia przykładowej, rzeczywistej architektury na podstawie jej parametrów i specyfikacji technicznej.
EK2.	Student wyjaśnia działanie oprogramowania demonstracyjnego oraz samodzielnie projektuje i dokumentuje proste oprogramowanie w języku niskiego poziomu.

Treści programowe: wykłady	Liczba godzin
W1 – Wprowadzenie - historia i ewolucja komputerów	1
W2 – Architektura von Neumanna i harwardzka. Elementy komputera	2
W3 – Jednostka centralna: elementy, cykl rozkazowy, lista rozkazów	2
W4 – Koprocesor arytmetyczny, format zmiennoprzecinkowy, norma IEEE-P754	2
W5 – Potokowe przetwarzanie rozkazów	2
W6 – Superskalarne przetwarzanie rozkazów	2
W7 – Przetwarzanie równoległe wg klasyfikacji Flynna: superkomputery, wieloprocesorowość, wielordzeniowość, rozszerzenia listy rozkazów	2
W8 – Pamięć: hierarchia, asocjacja, pamięć wirtualna, spójność pamięci	2
W9 – Układy otoczenia procesora: chipset, kontroler pamięci, kontroler DMA, kontroler układów wejścia-wyjścia, mechanizm przerwań	2
W10 – Przegląd architektur współczesnych komputerów oraz architektur alternatywnych	2
W11 – Reprezentacja danych w systemach komputerowych. Konwersje formatów	2
W12-14 – Operacje arytmetyczne w formacie stało- i zmiennopozycyjnym	7
W15 - Zaliczenie pisemne	2
SUMA	**30**

Treści programowe: laboratorium (ćwiczenia komputerowe)	Liczba godzin
L1 – Przedstawienie zasad odbywania zajęć, BHP oraz zasad zaliczenia laboratorium	0,5
L2 – Sterowanie liniami wejść/wyjść mikrokontrolera	1,5
L3 – Wewnętrzna pamięć danych RAM, tryby adresowania	1,5
L4 – Operacje arytmetyczne, stos, podprogramy	2

L5 – Sterowanie wyświetlaczem 7-segmentowym	1,5
L6 – Obsługa programowa klawiatury przeglądanej sekwencyjnie	2
L7 – Obsługa programowa klawiatury matrycowej	2
L8 – Sterowanie alfanumerycznym wyświetlaczem LCD	2
L9 – Układy czasowo-licznikowe	1,5
L10 – Zaliczenie zadań programistycznych / wpisy do indeksu	0,5
SUMA	**15**

Narzędzia dydaktyczne

1. Prezentacja multimedialna (wykład).
2. Tablica klasyczna lub interaktywna.
3. Oprogramowanie do wizualizacji wybranych zagadnień.
4. Autorski podręcznik akademicki do ćwiczeń z arytmetyki komputerowej (poz. 2 literatury obowiązkowej) dostępny w bibliotece uczelni i bibliotece wydziałowej.
5. Systemy mikroprocesorowe DSM-51 z mikroprocesorem MCU- 8051, dokumentacją i podręcznikiem.
6. Komputery PC.

Sposoby oceny efektów kształcenia (F – ocena Formująca, P – ocena Podsumowująca)

F1. Aktywność podczas wykładu (dyskusja, rozwiązywanie zadań z arytmetyki komputerowej przy tablicy).
F2. Ocena umiejętności analizy działania przykładowych programów.
P1. Wykład - test pisemny i zaliczenie zadań z arytmetyki komputerowej.
P2. Ocena umiejętności rozwiązywania postawionych zadań projektowych poprzez tworzenie odpowiedniego oprogramowania dla urządzeń mikroprocesorowych, prezentacji ich działania oraz wyciągania wniosków.

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	45
Zapoznanie się ze wskazaną literaturą	15
Przygotowanie się do zajęć laboratoryjnych	15
Przygotowanie się do testu i zadań	20
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**90 / 4**

Wykaz literatury podstawowej i uzupełniającej

1. Null L., Lobur J.: Struktura organizacyjna i architektura systemów komputerowych. Helion, Gliwice 2004.
2. Gryś S.: Arytmetyka komputerów w praktyce. Wyd. Naukowe PWN, Warszawa 2007 (dodruk 2013).
3. Baer J.L.: Microprocessor Architecture. From Simple pipelines to Chip Multiprocessors. Cambridge University Press, New York 2010.
4. Metzger P.: Anatomia PC, wyd. XI. Helion, Gliwice 2007.
5. Komorowski W.: Krótki kurs architektury i organizacji komputerów, Mikom, Warszawa 2004.
6. Patterson D., Hennessy J.: Computer Organisation and Design: The Hardware/Software Interface, Morgan Kaufmann, 2009.
7. Targowski A.: Historia – teraźniejszość – przyszłość informatyki, Politechnika Łódzka, Łódź 2013.
8. Czasopisma branżowe serii IEEE, ACM.

Macierz realizacji efektów kształcenia

Efekt kształcenia	Odniesienie efektu do efektów kształcenia dla dyscypliny naukowej Informatyka*	Cele przedmiotu	Forma zajęć	Narzędzia dydaktyczne	Sposób oceny
EK1	KINF1_W04, KINF1_U01, KINF1_U05	C1, C2, C3	W	1, 2, 3, 4	F1, P1
EK2	KINF1_U02, KINF1_U10	C1, C4	Lab	5, 6	F2, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student rozumie zasadę działania komputera jako całości i jego poszczególnych elementów, zna formaty liczb i zasady arytmetyki komputerowej oraz potrafi określić przeznaczenie, funkcjonalność i ograniczenia przykładowej, rzeczywistej architektury na podstawie jej parametrów i specyfikacji technicznej.**
2	Student nie rozumie zasady działania komputera jako całości ani jego poszczególnych elementów.
3	Student rozumie ogólną zasadę działania komputera jako całości i jego najważniejszych elementów, zna podstawowe formaty liczb i podstawowe zasady arytmetyki komputerowej.
3.5	Student rozumie ogólną zasadę działania komputera jako całości i jego najważniejszych elementów, zna formaty liczb i zasady arytmetyki komputerowej.
4	Student rozumie zasadę działania komputera jako całości i jego poszczególnych elementów, zna formaty liczb i zasady

	arytmetyki komputerowej.
4.5	Student rozumie zasadę działania komputera jako całości i jego poszczególnych elementów, zna formaty liczb i zasady arytmetyki komputerowej oraz potrafi określić podstawową funkcjonalność przykładowej, rzeczywistej architektury na podstawie jej parametrów i specyfikacji technicznej.
5	Student rozumie zasadę działania komputera jako całości i jego poszczególnych elementów, zna formaty liczb i zasady arytmetyki komputerowej oraz potrafi określić przeznaczenie, funkcjonalność i ograniczenia przykładowej, rzeczywistej architektury na podstawie jej parametrów i specyfikacji technicznej.
EK2	**Student wyjaśnia działanie oprogramowania demonstracyjnego oraz samodzielnie projektuje i dokumentuje proste oprogramowanie w języku niskiego poziomu.**
2	Student nie potrafi wyjaśnić działania prostych programów w języku niskiego poziomu.
3	Student zna listę rozkazów procesora i wyjaśnia działanie prostych programów w języku niskiego poziomu.
3.5	Student wyjaśnia działanie prostych programów w języku niskiego poziomu i potrafi zaproponować niewielkie jego modyfikacje.
4	Student wyjaśnia działanie oprogramowania demonstracyjnego oraz projektuje proste oprogramowanie w języku niskiego poziomu z pomocą prowadzącego zajęcia lub w zespole.
4.5	Student wyjaśnia działanie oprogramowania demonstracyjnego oraz projektuje proste oprogramowanie w języku niskiego poziomu.
5	Student wyjaśnia działanie oprogramowania demonstracyjnego oraz samodzielnie projektuje i dokumentuje proste oprogramowanie w języku niskiego poziomu.

III. INNE PRZYDATNE INFORMACJE O PRZEDMIOCIE

1. Wszelkie informacje dla studentów na temat planu zajęć dostępne są na tablicy ogłoszeń oraz na stronie el.pcz.pl.
2. Prowadzący udostępnia na pierwszych zajęciach treści wykładów.
3. Informacje na temat warunków zaliczania zajęć przekazywana jest studentom podczas pierwszych zajęć.