

KARTA PRZEDMIOTU

Nazwa przedmiotu: **Microprocessors (EiTAu>SI5-Microp-17)**

Nazwa w języku polskim:

Nazwa w jęz. angielskim: **Microprocessors**

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

EGZ

Język wykładowy:

angielski

Strona WWW:

<https://platforma.polsl.pl/rau3/course/view.php?id=30>

Skrócony opis:

Architektura mikroprocesorów i jej działanie, wykonywanie instrukcji, architektura magistrali, system przerwań, architektura systemów mikroprocesorowych, interakcja z pamięcią i komponentami peryferyjnymi, koncepcja i implementacja systemu przerwań, programowanie z użyciem języka asemblera i języków wysokiego poziomu.

Podstawy interfejsów szeregowych, koncepcje i implementacja szeregowego przesyłania danych. Praktyczne aspekty projektowania i implementacji systemów wbudowanych, programowanie i debugowanie, systemy debugowania on-chip, projektowanie niestandardowych urządzeń peryferyjnych, integracja systemów, symulacja i modelowanie systemów.

Opis:

Wykład (30 godzin):

Wprowadzenie do mikroprocesorów

- Ewolucja od układu sekwencyjnego, przez układ mikroprogramowalny, do mikroprocesora
- Podstawowe bloki funkcjonalne mikroprocesora i ich właściwości
- Wpływ architektury na zachowanie i wydajność systemu obliczeniowego
- Cykl wykonywania instrukcji. Architektury Harvard i Princeton (von Neumanna).
- Koncepcje wykonywania sekwencyjnego i potokowego

System komputerowy - ogólne właściwości i wymagania

Arytmetyka systemów komputerowych

- Systemy liczbowe
- Liczby ze znakiem - efektywna reprezentacja
- Liczby stałoprzecinkowe
- Liczby zmiennoprzecinkowe
- Implementacja operacji arytmetycznych

Systemy wieloprocessorowe

- Klasyfikacja architektury
- Wydajność systemów wieloprocessorowych i obszary zastosowań
- Grafy przepływu danych - reprezentacja obliczeń
- Podstawy harmonogramowania operacji i zadań

Adresowanie zestawów instrukcji i argumentów

- Model pamięci i tryby adresowania
- Zarządzanie pamięcią, ochrona i wirtualizacja
- Operacje implementowane
- Implementacja warunkowego przepływu programu
- Podprogramy

Architektura magistrali systemu komputerowego

- Cykle interfejsu magistrali
- Sterowanie czasem cyklu interfejsu
- Podłączanie pamięci i urządzeń peryferyjnych do mikroprocesora
- Przejmowanie kontroli nad magistralą systemową - bezpośredni dostęp do pamięci

System przerwań

- Przechwytywanie zdarzeń - permanentne sprawdzanie stanu vs przerwania
- Koncepcja implementacji systemu przerwań
- Znajdowanie źródła przerwania. Koncepcja wektoryzowanego systemu przerwań. - Klasyfikacja systemów przerwań

Działanie wieloprogramowe

- Idea i korzyści
- Metody przełączania zadań – systemy kooperacyjne i z wywłaszczaniem
- Zarządzanie zasobami systemu
- Ochrona – tryby pracy procesora

Procesory zaawansowane

- Architektury RISC i CISC
- Architektura potokowa – właściwości, korzyści i ograniczenia

- System ochrony pamięci

Komunikacja szeregową i wybrane interfejsy

- Zasady asynchronicznej transmisji szeregowej
- Zasady synchronicznej transmisji danych
- Zasady nadawania i odbioru danych
- Synchronizacja odbiornika w trybie asynchronicznym i synchronicznym - Metody kontroli integralności danych
- Interfejsy i protokoły: RS232, 1-Wire, I2C, USB

Ćwiczenia (15 godzin):

- Wprowadzenie do programowania mikroprocesorów z wykorzystaniem języka assemblera i języków wysokiego poziomu
- Wprowadzenie do assemblera
- Implementacja podstawowych zadań obliczeniowych z wykorzystaniem języka assemblera
- Implementacja bloków wykonywanych warunkowo. Logika obliczeniowa i warunki arytmetyczne
- Projektowanie podprogramów
- Przekazywanie argumentów do podprogramów.
- Obsługa zmiennych lokalnych.
- Implementacja procedur obsługi przerwań
- Implementacja wielozadaniowości – przypadki kooperacyjne i wyłączone
- Projektowanie systemów pamięci
- Projektowanie i łączenie urządzeń peryferyjnych

Laboratorium (30 godzin):

- Programowanie systemów wbudowanych w C/C++
- Uruchamianie debugowania On Chip z wykorzystaniem narzędzi programistycznych CAD/EDA
- Wykorzystanie narzędzi inspekcyjnych, takich jak oprogramowanie DataVizualizer
- Kontrola zasobów mikrokontrolera
- Obsługa przerwań
- Łączenie urządzeń z rzeczywistym sprzętem. Obsługa: wyświetlaczy LED, LCD i graficznych LCD, klawiatur, czujników
- Współbieżne działanie programu
- Optymalizacja programu dla systemów wbudowanych
- Implementacja urządzeń peryferyjnych z wykorzystaniem technologii HDL i FPGA. - Powiązanie niestandardowego urządzenia peryferyjnego z językiem programowania wysokiego poziomu
- Tworzenie i wykorzystywanie modeli funkcjonalnych magistrali w weryfikacji

ECTS: 6

Suma godzin: 150h (75h zajęć kontaktowych + 75h pracy własnej)

Wykład: 30h

Ćwiczenia: 15h

Laboratorium: 30h

Praca własna studenta: 75h (przygotowanie do zajęć laboratoryjnych, zapoznanie się z dokumentacją techniczną systemów peryferyjnych wykorzystywanych w laboratorium)

Literatura:

1. M. Morris Mano: Computer System Architecture
2. M. Morris Mano, C.R. Kime: Logic and Computer Design Fundamentals.
3. W. Wolf: Computers as Components: Principles of Embedded Computing Systems Design
4. S. Furber ARM System-on-chip Architecture
5. N. Wirth: Algorithms + Data Structures = Programs
6. B.W. Kernighan, D.M. Ritchie: The C Programming Language

Efekty uczenia się:

Wiedza: zna i rozumie

1. Zagadnienia architektury komputerów, w szczególności warstwy sprzętowej (K1A_W06)
2. Zagadnienia z zakresu metodologii i technik programowania (K1A_W07)
3. Zagadnienia z zakresu architektury i oprogramowania systemów mikroprocesorowych (języki wysokiego i niskiego poziomu) (K1A_W08)

Umiejętności: potrafi

1. formułować specyfikację prostych systemów elektronicznych na poziomie realizowanych funkcji, również z wykorzystaniem języków opisu sprzętu (K1A_U14)
2. projektować proste układy i systemy elektroniczne przeznaczone do różnych zastosowań, w tym proste systemy cyfrowego przetwarzania sygnałów (K1A_U16)
3. wykorzystywać karty katalogowe i noty aplikacyjne w celu doboru odpowiednich komponentów projektowanego systemu lub układu elektronicznego (K1A_U17)
4. formułować algorytm, używać języków programowania wysokiego i niskiego poziomu oraz odpowiednich narzędzi informatycznych do tworzenia programów komputerowych sterujących systemem elektronicznym oraz do programowania mikrokontrolerów lub mikroprocesorów sterujących w systemie elektronicznym (K1A_U22)

Kompetencje społeczne: Gotowość do

1. krytycznej oceny wiedzy i postrzeganych treści oraz uznania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych, a także do konsultowania się z ekspertami w przypadku trudności w samodzielnym rozwiązaniu problemu (K1A_K01)
2. podporządkowania się zasadom pracy zespołowej i ponoszenia odpowiedzialności za wspólnie realizowane zadania (K1A_K04)

Metody i kryteria oceniania:

Wykład: Egzamin – 1/2 oceny końcowej

Laboratorium: Wykonanie wszystkich przypisanych zadań laboratoryjnych – 1/2 oceny końcowej

Ocena końcowa z przedmiotu jest obliczana w następujący sposób:

Ocena końcowa = $0,5 \cdot \text{ocena z wykładu} + 0,5 \cdot \text{ocena z laboratorium}$,
zaokrąglona do skali ocen obowiązującej na uczelni.

Sylabus obowiązuje od roku akademickiego 2024/25 i a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Punkty przedmiotu w cyklach:

Elektronika i Telekomunikacja, stacjonarne I stopnia inżynierskie 7 sem. (EiTAu-SI7)

Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	6	2020/2021-Z	