**Nazwa przedmiotu:**

Podstawy techniki mikroprocesorowej

**Koordynator przedmiotu:**

Bogumił KONARZEWSKI

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Telekomunikacja

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

TMIK

**Semestr nominalny:**

7 / rok ak. 2018/2019

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

102

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 30h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

56

**Cel przedmiotu:**

\* ukształtowanie podstawowych umiejętności w zakresie projektowania układów z wykorzystaniem mikroprocesorów (mikrokontrolerów)
\* ukształtowanie podstawowych umiejętności w zakresie oprogramowania w języku asemblera systemów mikroprocesorowych
\* ukształtowanie podstawowych umiejętności w zakresie uruchamiania systemów mikroprocesorowych

**Treści kształcenia:**

Treść wykładu
1. Systemy liczenia – dziesiętny i binarny, zapis heksadecymalny, reprezentacja binarna liczb całkowitych (NKB, znak/moduł, zapis U1 i U2), operacje arytmetyczne i logiczne na liczbach binarnych, zapis stałoprzecinkowy i zmiennoprzecinkowy liczb rzeczywistych.
2. Układy cyfrowe: parametry (poziomy napięć, margines szumów, czas propagacji, moc pobierana, współczynnik dobroci, obciążalność), typowe rodziny (bipolarne, CMOS), rodzaje wejść (zwykłe, Schmitta) oraz wyjść (przeciwsobne, typu OC/OD, trójstanowe), zasady łączenia układów różnych rodzin.
3. Historia rozwoju mikroprocesorów, budowa CPU (układ sterowania, układ wykonawczy, generator sygnału takt.) i sposób działania (pamięć-kody rozkazów i dane, cykl rozkazowy, maszynowy, zegarowy), struktura systemu mikroprocesorowego (jednostka centralna, pamięć, układy wej./wyj, magistrale), mikroprocesor a mikrokontroler, parametry mikroprocesorów, architektury von Neumannna, Harvard oraz zmod. Harvard, mikroprocesory typu RISC i CISC, przetwarzanie potokowe i superpotokowe, mikroprocesory superskalarne i hiperskalarne, architektura ARM, systemy wieloprocesorowe i wielokomputerowe.
4. Mikroprocesor Z80 – architektura wew., cykle maszynowe, system przerwań, lista rozkazów i tryby adresowania, wykorzystanie stosu.
5. Podstawy programowania w asemblerze: rozkaz – mnemonik, dyrektywy asemblera, etapy tworzenia programu, format pliku typu HEX.
6. Mikroprocesor Z80 – generacja sygnału zegarowego oraz sygnału RESET, sprzęganie CPU z pozostałymi elementami systemu mikroprocesorowego, układ do pracy krokowej.
7. Pamięci półprzewodnikowe w systemach mikroprocesorowych – stałe (MROM, PROM, EPROM, EEPROM) i ulotne (SRAM, DRAM), parametry pamięci (pojemność, czas dostępu, czas cyklu), standaryzacja wyprowadzeń wg. JEDEC, przykłady pamięci: SRAM 6264, DRAM 4164, EPROM 2716, EEPROM 2816 i Flash EEPROM 29256, projektowanie podsystemów pamięci, dekodowanie pełne, niepełne oraz z liniową selekcją adresów, pamięci specjalizowane: dwuportowe, FIFO, z magistralą szeregową.
8. Pamięci kieszeniowe w systemach mikroprocesorowych – metody odwzorowanie: bezpośrednia, skojarzeniowa i zmodyfikowana skojarzeniowa, dostęp z przeplotem do pamięci w systemach mikroprocesorowych.
9. Przerwania w systemach mikroprocesorowych – jednopoziomowe i wielopoziomowe, z priorytetami i bez priorytetów, metody identyfikacji źródła zgłoszenia (przeglądanie, system łańcuchowy z wektoryzacją, koder priorytetowy), czułość na poziom oraz zbocze wejścia przerywającego mikroprocesora, przerwania NMI i INT w mikroprocesorze Z80 – procedury obsługi.
10. Budowa i tryby pracy programowalnego kontrolera przerwań 8259A (PIC) – sposób sprzęgania z mikroprocesorem Z80, dekodowanie PIC z wykorzystaniem scalonego komparatora oraz programowanie PIC.
11. Sprzęganie systemu mikroprocesorowego ze światem zewnętrznym (urządzeniami zewnętrznymi), metody transmisji (strobowna/z potwierdzeniem, programowa/ster. przerwaniami/ z bezp. dostępem do pamięci, szeregowa/równoległa, synchroniczna/asynchroniczna, symetryczna/asymetryczna), typy układów wej/wyj), sposoby adresowania układów wej/wyj, realizacja bezpośredniego wejścia i wyjścia w systemach mikroprocesorowych, wyświetlanie statyczne i dynamiczne, zjawisko „odbijania styków”.
12. Przykłady uniwersalnych układów wej/wyj – nieprogramowalnego 8212 i programowalnego 8255A (budowa i tryby pracy), wykorzystanie układu 8255A jako bezpośredniego wyjścia oraz wejścia (sterowanie diodami LED/wyświetalczem 7-segmentowym oraz odczyt stanu przełączników).
13.Przykłady sterujących układów wej/wyj – układ czasowo-licznikowy 8254 i układ bezpośredniego dostępu do pamięci 8237 (budowa i tryby pracy), wykorzystanie układu 8254 w systemach mikroprocesorowych do generacji skali czasu (generowania przerwania zegarowego), generacja skali czasu metodą pętli opóźniającej.
14. Sprzęganie systemu mikroprocesorowego ze światem zewnętrznym „analogowym” – podstawowe struktury przetworników A/C i C/A, sposoby dołączenia do systemu mikroprocesorowego i wymiany danych.
15. Mikrokontrolery 8051 i 68HC05K1 – przykłady układów o strukturze otwartej i zamkniętej, architektura wew., cykle maszynowe, system przerwań, lista rozkazów i tryby adresowania, układy portów wej/wyj. podsystem czasowo licznikowy.
16. Mikroprocesory rodziny 68000 – przykłady układów o 32-bitowej architekturze, architektura wew., cykle maszynowe, system przerwań, lista rozkazów i tryby adresowania.
17. Narzędzia uruchomieniowe dla systemów mikroprocesorowych – emulator pamięci EPROM, symulator programowy, monitor, symulator układowy, emulator.
18. Interfejsy w systemach mikroprocesorowych – RS232 (RS422/RS423/RS485), USB, SPI, I2C, Centronics.
Zakres laboratorium
Ćw. A. Skonstruowanie systemu mikroprocesorowego złożonego z procesora Z80 oraz pamięci RAM 6264 i pamięci EPROM 2716 (dekodowanych w sposób pełny), a następnie napisanie i uruchomienie prostego programu w języku asemblera Z80.
Ćw. B. Dołączenie do „standardowego systemu z mikroprocesorem Z80” (Z80 + pamięć RAM 6264 dekodowana w sposób niepełny) programowalnego kontrolera przerwań, a następnie napisanie w języku asemblera Z80 programu inicjującego pracę układu 8259A i obsługującego przerwania maskowalne INT zgłaszane przez urządzenia zewnętrzne. Studenci obserwują (w trybie pracy krokowej procesora) przyjęcie zgłoszenia przerwania oraz realizację procedury obsługi przerwania, również w przypadku zagnieżdżania procedur.
Ćw. C. Dołączenie do „standardowego systemu z mikroprocesorem Z80” (Z80 + pamięć RAM 6264 dekodowana w sposób niepełny) układu wejścia/wyjścia równoległego 8255 i układu czasowo-licznikowego 8254, a następnie napisanie w języku asemblera Z80 programu inicjującego pracę obu tych układów i realizującego pomiar czasu (skala czasu realizowana przez przerwania zegarowe zgłaszane na wejściu INT przez układ 8254) z wyświetlaniem na bieżąco wyniku na wyświetlaczu 7-segmentowym (dołączony do układu 8255, który pracuje jako bezpośrednie wyjście).
Ćw. D. Zrealizowanie układu do pomiaru czasu z wykorzystaniem systemu z mikrokontrolerem Intel 8031 z dołączoną pamięcią RAM 6264 (pracuje jako pamięć programu). Skala czasu realizowana przez przerwania zegarowe zgłaszane przy przepełnieniu wewnętrznego licznika mikrokontrolera, a wynik wyświetlany na bieżąco na wyświetlaczu 7-segmentowym dołączonym do portu P1 mikrokontrolera.
Ćw. E. Zrealizowanie układu do pomiaru czasu z wykorzystaniem systemu z mikrokontrolerem Motorola 68HC05K1 (w postaci symulatora sprzętowego). Skala czasu realizowana przez przerwania zegarowe zgłaszane przy przepełnieniu wewnętrznego licznika mikrokontrolera, a wynik wyświetlany na bieżąco na wyświetlaczu 7-segmentowym dołączonym do portu A mikrokontrolera.
Ćw. F. Napisanie w języku asemblera Motorola 68000 prostego programu do wyświetlania zadanej sekwencji na wyświetlaczu 7-segmentowym (istnieje możliwość sterowania indywidualnych segmentów) przy wykorzystaniem platformy uruchomieniowej IDP. Skala czasu realizowana jest przy pomocy pętli opóźniającej.
Przed każdym ćwiczeniem studenci rozwiązują zadnia projektowe (podane w instrukcjach laboratoryjnych) podobne do realizowanych w laboratorium podczas tego ćwiczenia. Każde zajęcia laboratoryjne trwają 225 min. bez przerwy. Ocena ćwiczeń odbywa się na podstawie prezentacji skonstruowanego systemu, sprawozdania i rozmowy z zespołem.

**Metody oceny:**

Wiedza i umiejętności studentów sprawdzane są podczas dwóch kolokwiów wykładowych i laboratorium. Ocena z kolokwiów wchodzi z wagą 0.4 do oceny z przedmiotu – należy jednak uzyskać co najmniej połowę maksymalnej liczby punktów z kolokwiów. Ocena z laboratorium wchodzi z wagą 0.6 do oceny z przedmiotu.

**Egzamin:**

nie

**Literatura:**

Literatura
[1] K. Badźmirowski i in. „Układy i systemy mikroprocesorowe” WNT, Warszawa 1990
[2]J. W. Coffron, W. E. Long „Technika sprzęgania układów w systemach mikroprocesorowych” WNT, Warszawa 1988
[3] W. Daca „Mikrokontrolery. Od układów 8-bitowych do 32-bitowych” Mikom,Warszawa 2000
[4] P. Hadam „Projektowanie systemów mikroprocesorowych” Wydawnictwo BTC, Warszawa 2004
[5] J. Karczmarczuk „Mikroprocesor Z80” WNT, Warszawa 1987
[6] P. Misiurewicz „Podstawy techniki mikroprocesorowej” WNT, Warszawa 1991
[7] pr. zb. „Modułowe systemy mikrokomputerowe” WNT, Warszawa 1990
[8] R. Pełka „Mikrokontrolery. Architektura, programowanie, zastosowania” WKiŁ, Warszawa 2000
[9] J. Pieńkos, S. Moszczyński, A. Pluta „Układy mikroprocesorowe 8080/8085 w modułowych systemach sterowania” WKiŁ, Warszawa 1988
[10] A. Rydzewski, K. Sacha „Mikrokomputer. Elementy, budowa, działanie” WCiKT NOT-SIGMA, Warszawa 1985
[11] K. Sacha, A. Rydzewski „Mikroprocesor w pytaniach i odpowiedziach” WNT, Warszawa 1985
[12] J. M. Sibigtroth „Zrozumieć małe mikrokontrolery” Wydawnictwo BTC, Warszawa 2003
[13] T. Starecki „Mikrokontrolery 8051 w praktyce” Wydawnictwo BTC, Warszawa 2002
[14] B. Zieliński „Układy mikroprocesorowe. Przykłady rozwiązań” Helion, Gliwice 2002

**Witryna www przedmiotu:**

brak

**Uwagi:**

brak

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W1:**

wiedza nt. struktur systemów mikroprocesorowych

Weryfikacja:

kolokwium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W2:**

wiedza nt. organizacji przerwań w systemach mikroprocesorowych

Weryfikacja:

kolokwium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W3:**

wiedza nt. metod wymiany danych systemu mikroprocesorowego z urządzeniami zewnętrznymi

Weryfikacja:

kolokwium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W4:**

wiedza nt. sprzęgania układów cyfrowych z różnych rodzin w ramach jednego systemu cyfrowego

Weryfikacja:

kolokwium/laboratorium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U1:**

przeprowadzenie diagnostyki systemu mikroprocesorowego z wykorzystaniem próbnika logicznego, programu monitora oraz symulatora

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U2:**

zaprojektowanie systemu mikroprocesorowego współpracującego z pamięcią oraz układami wejścia/wyjścia

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U3:**

oprogramowanie systemu mikroprocesorowego w języku asemblera łącznie z debuggowaniem w trybie pracy krokowej

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K1:**

umijętność pracy w zespole

Weryfikacja:

laboratorium

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**