



**Katedra Mikroelektroniki i Technik Informatycznych**  
**Politechnika Łódzka**



# Architektura komputerów

**dr inż. Bartosz Pękosławski**

Łódź, dn. 6.03.2022



# Dane kontaktowe

<b>Adres e-mail:</b>	<b><a href="mailto:bartoszp@dmcs.pl">bartoszp@dmcs.pl</a></b>
<b>Bieżące informacje:</b>	<b><a href="http://www.dmcs.pl/web/bartoszp/strona-pracownika">www.dmcs.pl/web/bartoszp/strona-pracownika</a></b>
<b>Materiały dla studentów:</b>	<b><a href="http://fiona.dmcs.pl/~bartoszp/ak.html">fiona.dmcs.pl/~bartoszp/ak.html</a> <a href="http://lux.dmcs.pl/ak">lux.dmcs.pl/ak</a></b>
<b>Godz. przyjęć:</b>	<b>wtorki 14:00-15:00, lab. ZK, MS Teams piątki 10:00-11:00, lab. ZK, MS Teams</b>



# Plan wykładów

1. Wstęp. Systemy zapisu liczb. Historia maszyn liczących i komputerów. Języki programowania.
2. Kodowanie liczb (NKB, U2, BCD, ZM, U1, IEEE 754)
3. Architektura typu single-cycle
4. Architektura typu multi-cycle
5. Mikroprogram
6. Architektura potokowa RISC. Konflikty zasobów i danych.
7. Rozwiązywanie konfliktów danych i sterowania w architekturze potokowej. Optymalizacja kodu. Forwarding. Wczesne wykrywanie skoków.
8. Pamięci cache, pamięć wirtualna
9. Przykład architektury RISC – mikrokontroler AVR ATmega
10. Programowanie mikrokontrolera AVR w języku C

# Literatura

## Literatura podstawowa:

1. J. L. Hennessy, D. A. Patterson, *Computer organization & Design*, Ed. 2...5, Morgan-Kaufmann Publishers, 1998...2013
2. B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, 2nd edition, 2009

## Literatura uzupełniająca:

3. J. L. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 5th edition, 2011
4. A. Tanenbaum, Todd Austin, *Structured Computer Organization*, Pearson, 6th edition, 2012
5. R. Baranowski, *Mikrokontrolery AVR ATmega w praktyce*. BTC 2005
6. W. Stallings, *Organizacja i architektura systemu komputerowego*, WNT, 2000
7. J. Biernat, *Arytmetyka Komputerów*, Wydawnictwo Naukowe PWN, 1996





# Ocena końcowa

70% ocena z egzaminu,

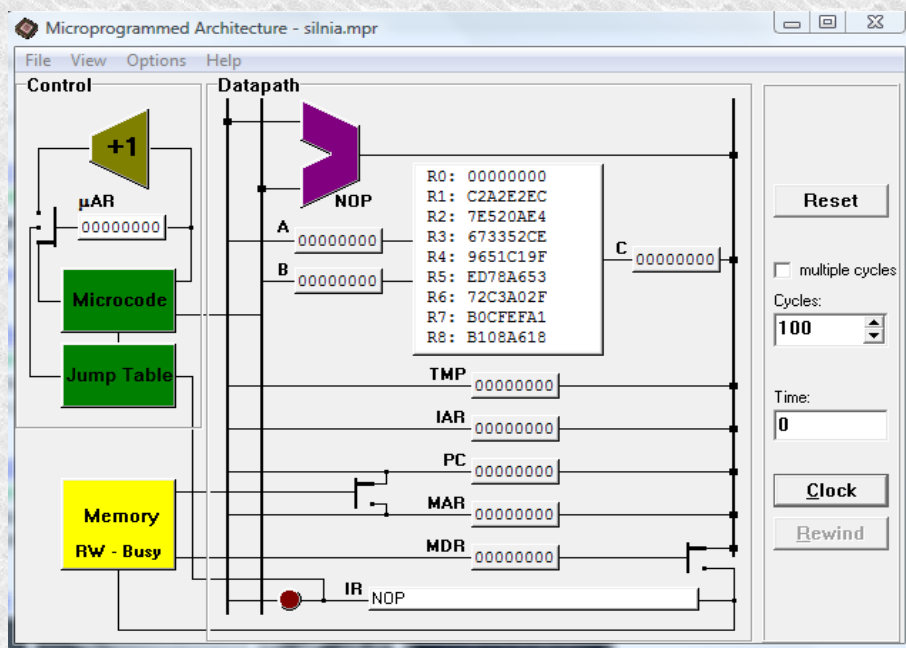
30% ocena z ćwiczeń

(konieczna ocena pozytywna z każdej części)

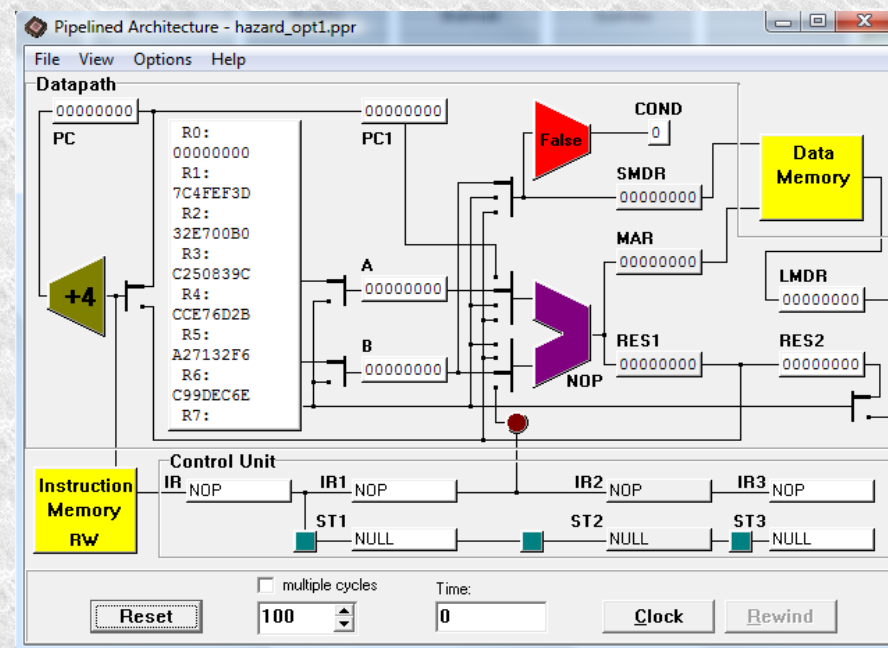
Zaliczenie ćwiczeń jest warunkiem koniecznym do dopuszczenia do egzaminu.

## Ćwiczenia – zadania egzaminacyjne

### Laboratorium (semestr 5) – symulator Escape



Architektura mikroprogramowa



Architektura potokowa



# Cele przedmiotu

- Omówienie kodowania liczb i układów arytmetycznych w systemach komputerowych
- Przedstawienie zasady działania architektur CISC i RISC: rejestrów, przepływu danych, jednostki sterującej oraz instrukcji maszynowych
- Zapoznanie z zaawansowanymi cechami współczesnych systemów komputerowych: pamięć cache, pamięć wirtualna
- Zapoznanie z procesem wykonywania instrukcji maszynowych i programów w języku asemblera



# Przedmiotowe efekty kształcenia

1. Student zna podstawy arytmetyki komputerowej: kodowanie liczb i sprzętowe realizacje układów arytmetycznych.
2. Student rozumie działanie elementów architektury procesora, sterowanie przetwarzaniem informacji, strukturę kodu maszynowego i instrukcje asemblera.
3. Student rozumie koncepcje sterowania mikroprogramowalnego oraz przetwarzania potokowego.
4. Student zna strukturę hierarchicznego systemu pamięci komputera oraz działanie cyfrowych protokołów komunikacyjnych.

**Egzamin pisemny: efekty 1 ÷ 4**





# Podstawy

## Podstawy:

**Podstawy elektrotechniki i elektroniki**

**Wstęp do techniki cyfrowej**



# Wykład 1 - plan

## Slajdy autorstwa dr-a Witolda Marańdy

1. Bardzo krótka historia systemów liczących i maszyn liczących
2. Zastosowania komputerów
3. Języki programowania
4. Przyszłość – interfejsy, konwergencja technologii