

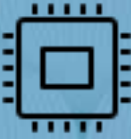


**Uniwersytet
w Siedlcach**

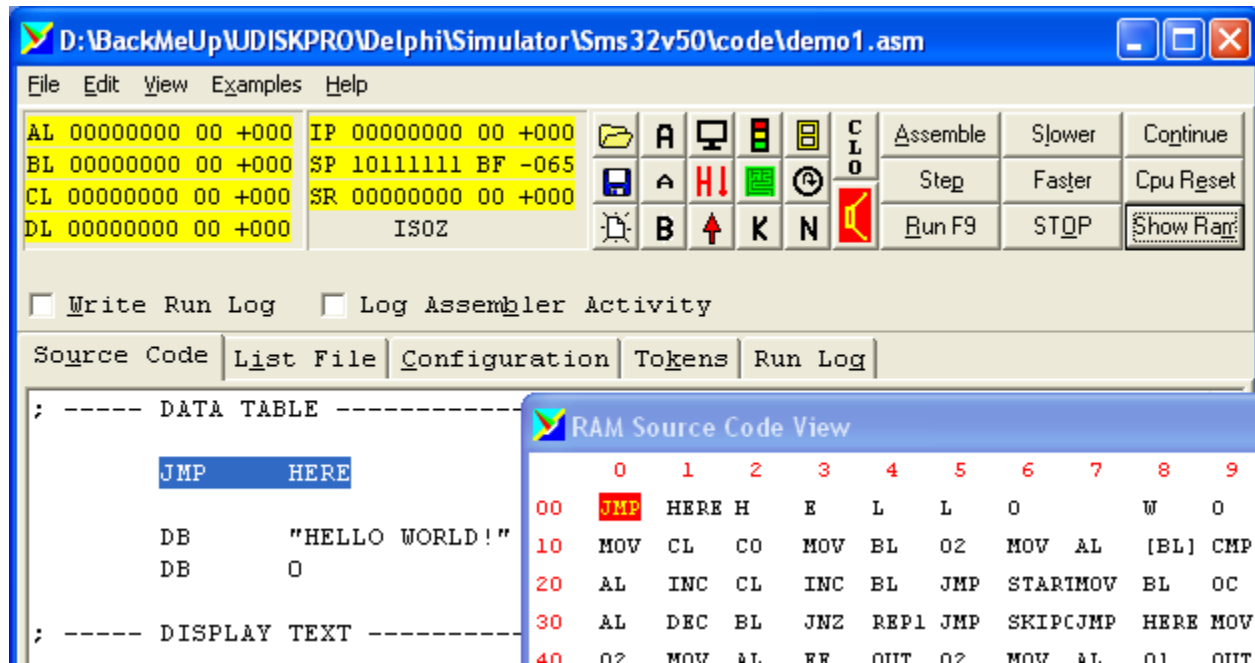
Architektura Systemów Komputerowych

**dr Marcin
Stępnia**

CEL: poznać architekturę komputera wg. von Neumanna

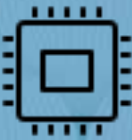


- Nie tylko poznać ale i programować w prostym assemblerze na prostej maszynie, tj. symulatorze **Microprocessor Simulator V5.0**
- <https://nbest.co.uk/Softwareforeducation/sms32v50/index.php>





Microprocessor Simulator



D:\BackMeUp\UDISKPRO\Delphi\Simulator\Sms32v50\code\demo1.asm

File Edit View Examples Help

Assembly Slower Continue Step Faster CPU Reset RunFS STOP Show Regs

Write Run Log Log Assembly Activity

Source Code List File Configuration Tokens Run Log

```
AL 00010001 11 +017 IF 10001011 SE -117
BL 00010100 14 +020 SP 10111111 BF -065
CL 11001100 CC +052 SR 00000000 DO +000
DL 00000000 00 +000 IS0Z
```

OUT 02 ; LSB used for multiplexing

MOV AL,01 ; 0000 0001b

OUT 02 ;

MOV AL,0 ; 0000 0000b

OUT 02 ;

DEC BL ; Count down

JNZ REP2 ; Jump out of loop on zero

----- HEATER AND THERMOSTAT -----

IN 03 ; Input from thermostat on port 3

CMP AL,01 ; Is it too warm

JZ OFF ; If no then jump to OFF

MOV AL,80 ; Use MSB to turn heater on.

OUT 03 ; Send 10000000 to port 3

JMP Skip2 ; Jump past heater-off code

OFF:

MOV AL,0 ; Turn off

OUT 03 ; Send 0

Skip2:

MOV BL,20 ; Time D

ON:

DEC BL ; BL cou

JNZ ON ; Jump o

----- SNAKE IN THE MAZE -----

MOV AL,FF ; Maze r

OUT 04 ; Snake

MOV BL,0A ; Count

MOV AL,4F ; 4 mesh

REP5:

OUT 04 ; Send d

DEC BL ; Count

JNZ REP5 ; Jump o

----- SPIN THE STEPPER MOTOR -----

MOV BL,20 ; Count

MOV AL,11 ; 0001 0001b

REP6:

OUT 05 ; Steppe

ROL AL ; Rotate

DEC BL ; Count

JNZ REP6 ; Jump o

Key Pad

INT 04 / Port 08

HELLO WORLD!

Traffic Lights on Port One

MSB 01010101 LSB

Stepper Motor - Part 5

Seven Segment Displays

PORT 2

MSB 00000000 LSB

Snake and Maze

Port 04

Thermometer

Heater

Thermostat

ON

MSB 10000000 LSB

13.99 C

Port 04

FF to Reset

Up Down Left Right Distance

0 1 0 0 1 1 1 1

88 46 20 16 6 4 2 1

Stopped

Motor

up

down

MSB 00001000 LSB

Repair Crushed Life

RAM Source Code View

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|-----------|-------|-----|------|------|-----|--------|-----|------|-----|-----|----|-----|------|-----|------|
| 00 | JMP | HERE | H | E | L | L | 0 | U | O | R | L | D | I | O | CLO | |
| 10 | MOV | CL | CO | MOV | BL | 02 | MOV | AL | [BL] | CMP | AL | 0 | JZ | END1 | MOV | [CL] |
| 20 | AL | INC | CL | INC | DL | JMP | START | MOV | BL | 0C | MOV | AL | AA | OUT | 01 | NOT |
| 30 | AL | DEC | BL | JNZ | REP1 | JMP | SKIP01 | JMP | HERE | MOV | BL | 0C | MOV | AL | FF | OUT |
| 40 | 02 | MOV | AL | FE | OUT | | | | | | | | | | | |
| 50 | DEC | BL | JNZ | REP2 | IN | | | | | | | | | | | |
| 60 | JMP | SKIP2 | MOV | AL | 0 | | | | | | | | | | | |
| 70 | FF | OUT | 04 | MOV | BL | | | | | | | | | | | |
| 80 | BL | 20 | MOV | AL | 11 | | | | | | | | | | | |
| 90 | MOV | BL | 0C | MOV | AL | | | | | | | | | | | |
| A0 | MIDDLEEND | END | END | END | | | | | | | | | | | | |
| B0 | END | END | END | END | END | | | | | | | | | | | |
| C0 | 48 | 45 | 4C | 4C | 4F | | | | | | | | | | | |
| D0 | | | | | | | | | | | | | | | | |
| E0 | | | | | | | | | | | | | | | | |
| F0 | | | | | | | | | | | | | | | | |

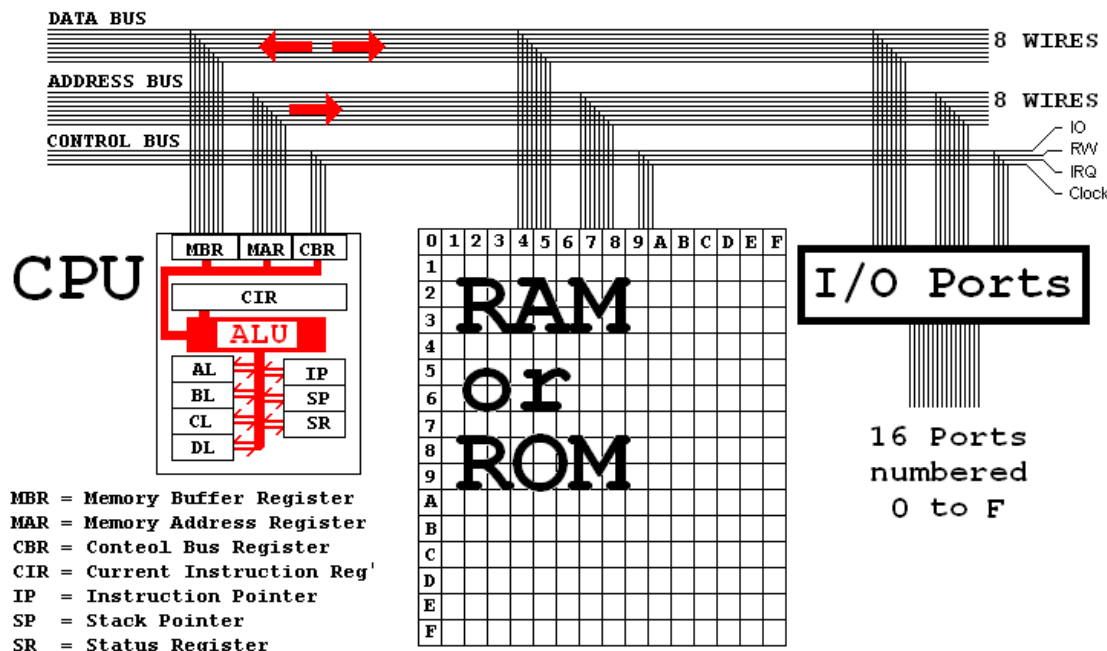
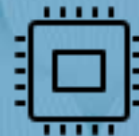
Key press generates INT 03 - Read key from port 07

INT 03/Port 07

Hexadecimal ASCII Source



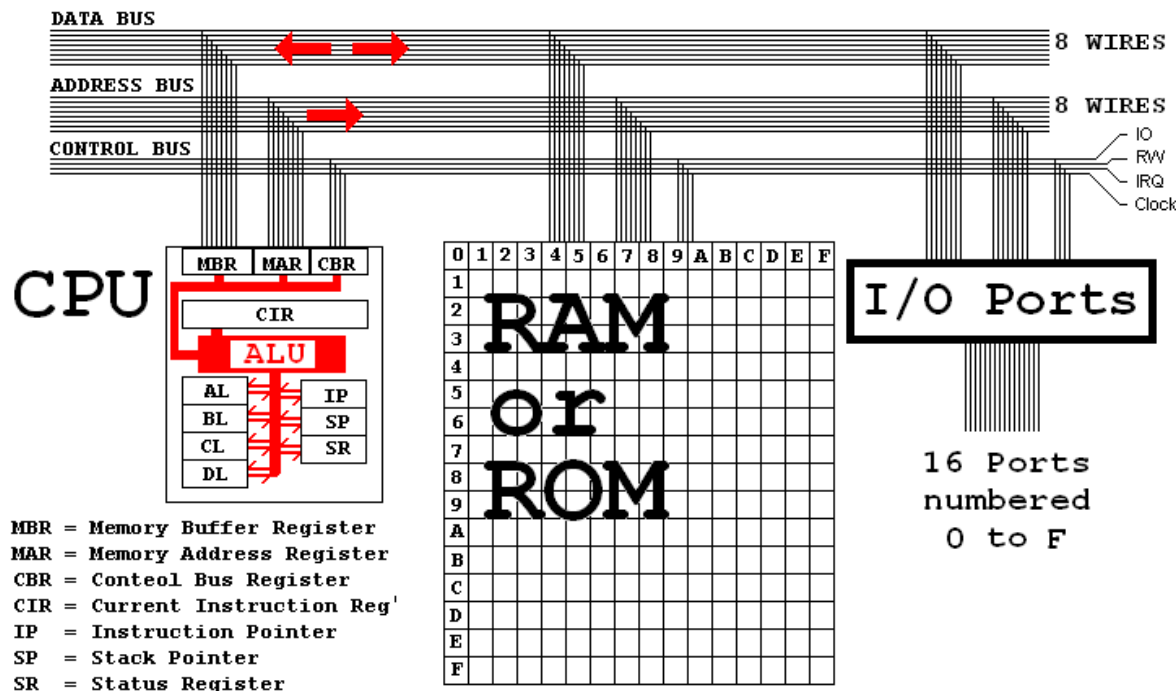
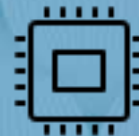
Microprocessor Simulator



- procesor - (CPU) central processing unit
- 256 bajtów pamięci (RAM)
- 16 portów input oraz output (IO) każdy adresowany od 0 do F. Tylko 6 jest używanych
- Sprzętowy zegar uruchamiany przerwaniem numer 02. Można oprogramować kolejne takty zegara. Długość taktów można konfigurować
- Klawiatura (rzeczywista) może być obsługiwana przerwaniem numer 03.
- Urządzenia zewnętrzne (symulowane) podłączone za pomocą portów input-output



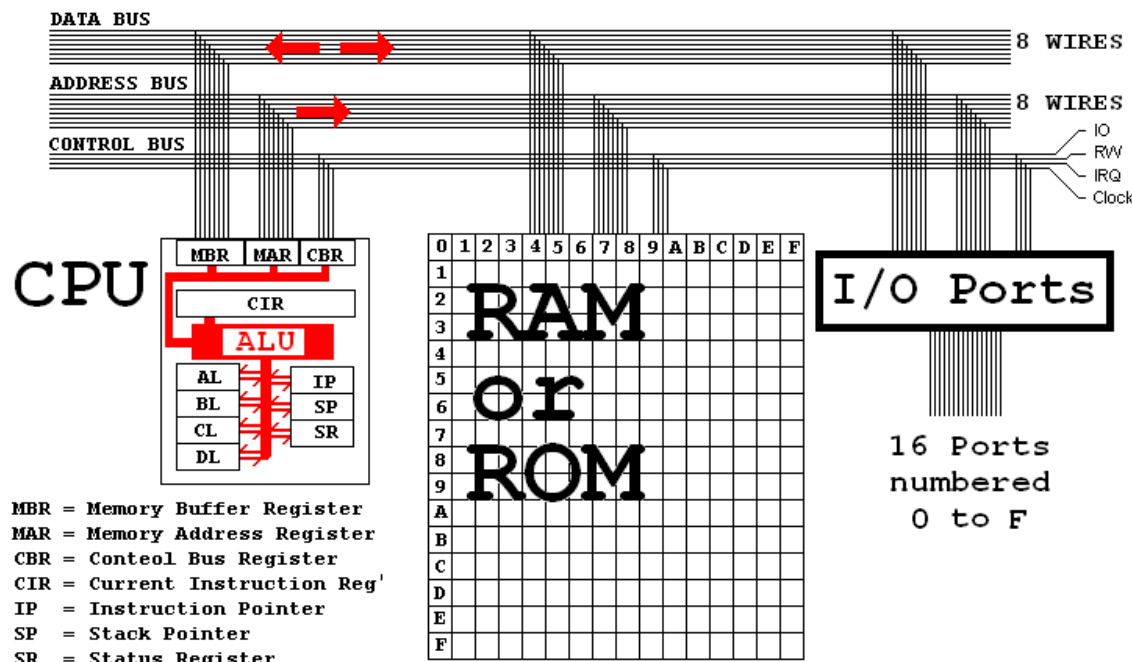
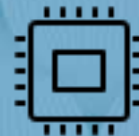
Microprocessor Simulator



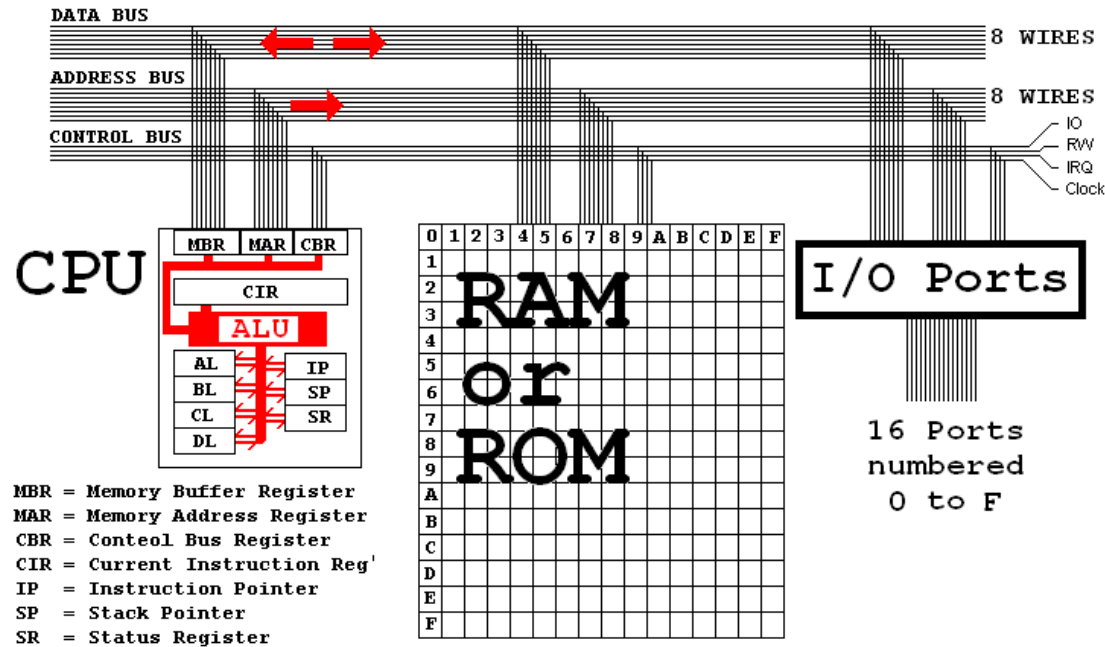
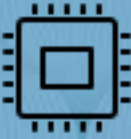
- Jest to też typowa architektura mikrokontrolerów gdzie zamiast RAM jest zazwyczaj ROM - read only memory.
- **CPU - Central Processing Unit.** Jest to "mózg" komputera. Tam są dokonywane wszystkie obliczenia (przetwarzanie bajtów), pobierani z pamięci i z portów wejściowych (input), oraz zapisywanie do pamięci lub wysyłanie danych do portów wyjściowych (output). CPU ma lokalną pamięć w postaci rejestrów. ALU (arithmetic and logic unit) służy do przetwarzania danych bajtów. Dane są brane z rejestrów, przetwarzane i umieszczane w rejestrach. Instrukcja MOV służy do przesyłania danych pomiędzy rejestrami a komórkami pamięci RAM.



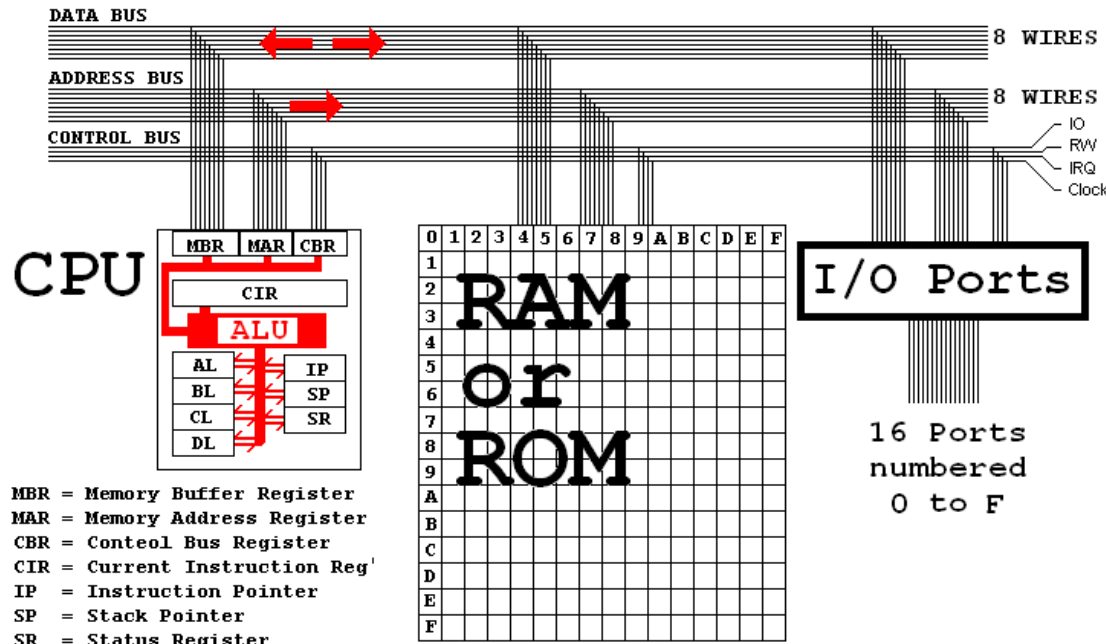
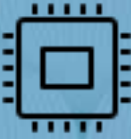
Microprocessor Simulator



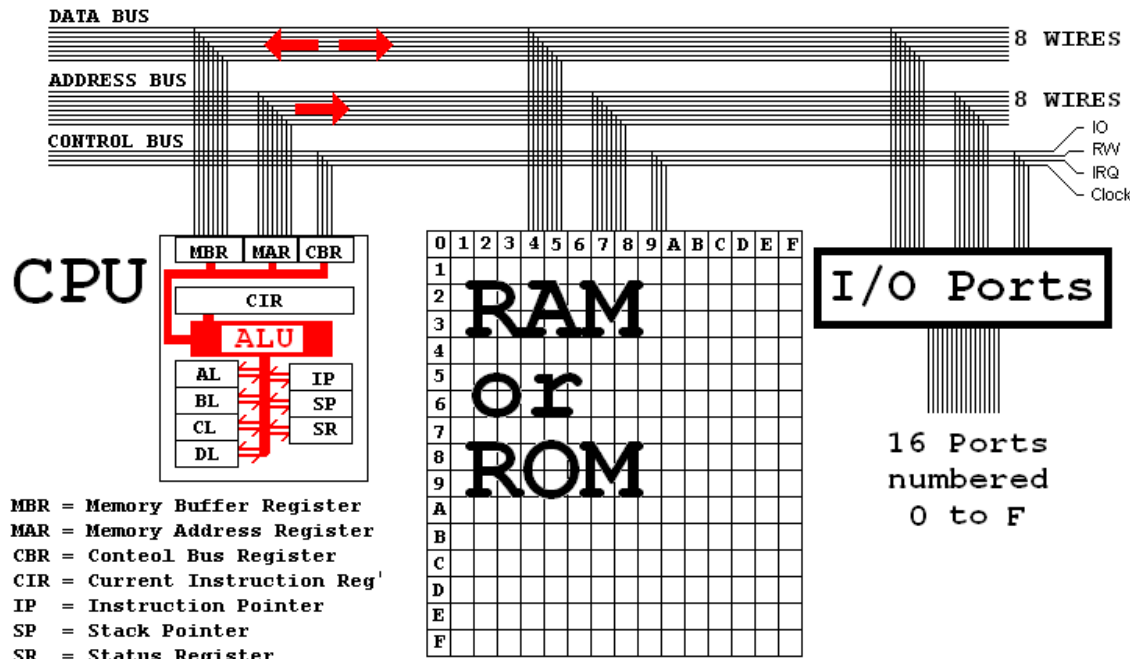
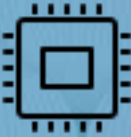
- **Rejestry ogólnego przeznaczenia - General Purpose Registers**
- Cztery rejestry AL, BL, CL oraz DL. Każdy ma po 8 bitów, czyli 1 bajt. można w nim umieścić liczbę bez znaku od 0 do 255, oraz liczbę ze znakiem w przedziale od -128 do +127. Jest to pamięć szybka, podręczna, tymczasowa. Szybkie procesory rzeczywiste posiadają znacznie więcej takich rejestrów. Te cztery rejestry w symulatorze mają swoje odpowiedniki w rzeczywistych 16 bitowych procesorach. Są to rejestry A, B, C, oraz D. Przy czym AL to dolna (ang. Low) część rejestru A, zaś AH to jego górna (ang. High) część. Podobnie dla pozostałych rejestrów B, C oraz D.



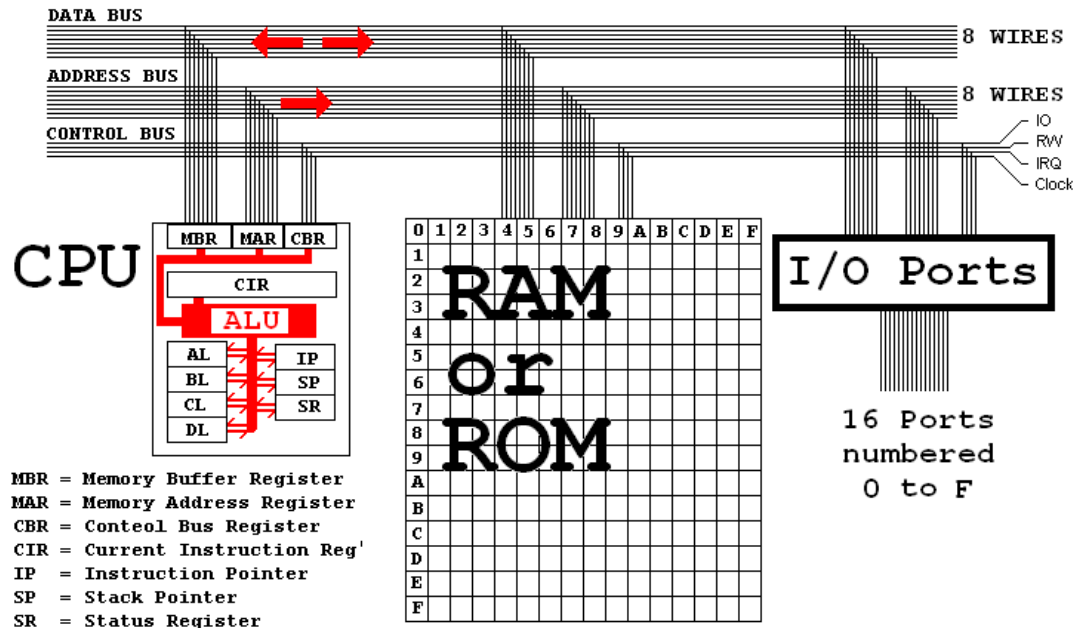
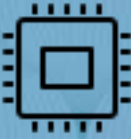
- **Rejestry specjalnego przeznaczenia - Special Purpose Registers**
 - IP, SR oraz SP
- **IP - Instruction pointer.** Zawiera adres komórki w RAM gdzie jest instrukcja aktualnie wykonywana. Po wykonaniu instrukcji zazwyczaj ten adres wzrasta o jedną „instrukcję” chyba, że jest wykonywany skok, wywoływana jest procedura (CALL), lub następuje przerwanie programowe (INT) lub sprzętowe. W pamięci RAM ta instrukcja jest podświetlana na czerwono z żółtym tekstem.



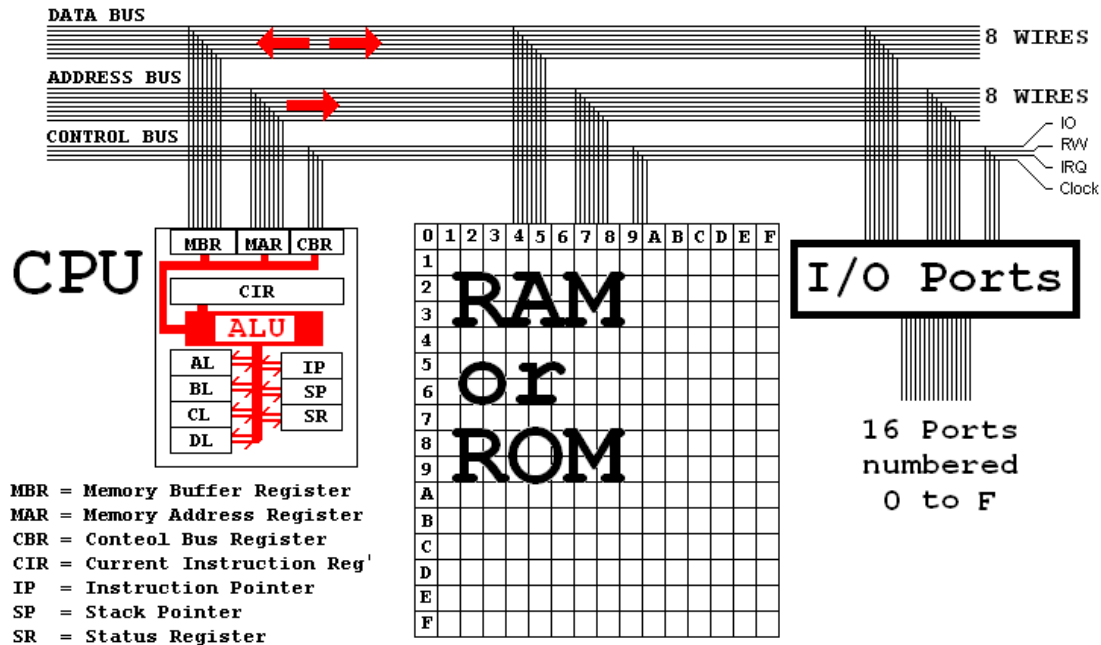
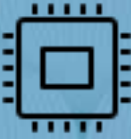
- **Rejestry specjalnego przeznaczenia - Special Purpose Registers**
- **SR - Status Register.** Ten rejestr zawiera flagi (pojedyncze bity) wskazujące na aktualny stan CPU
- Flaga 'Z' zero jest ustawiona (bit 1), jeśli wynik ostatniego obliczenia jest 0
Flaga 'S' sign - znaku jest ustawiona (bit 1), jeśli wynik ostatniego obliczenia jest ujemny
Flaga 'O' overflow – przepełnienia jest ustawiona (bit 1), jeśli wynik ostatniego obliczenia wykracza poza rejestr
Flaga 'I' interrupt – przerwania jest ustawiona, jeśli możliwe są przerwania sprzętowe.



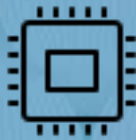
- **Rejestry specjalnego przeznaczenia - Special Purpose Registers**
- **SP - Stack Pointer**
- Stos to specjalnie wydzielony obszar pamięci RAM, do którego dostęp jest poprzez kolejkę LIFO (last in first out). Wartość w rejestrze SP wskazuje na następną wolną komórkę stosu. Pierwsza komórka stosu ma adres BF; następne są na lewo od niej. Można wpisywać wartość na stos, oraz zdejmować to co zostało ostatnio zapisane. W RAM aktualny wskaźnik stosu (komórka o tym adresie) jest zaznaczony na niebiesko z żółtym tekstem.



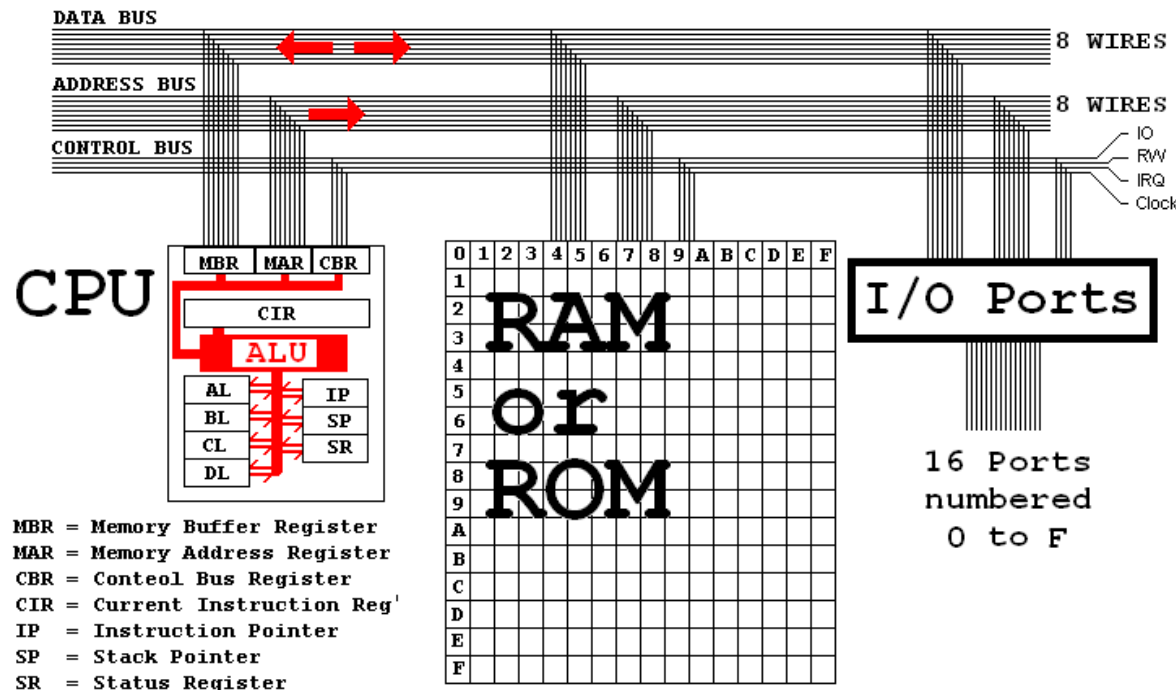
- **CIR – Current Instruction register** – rejestr, w którym znajduje się aktualnie wykonywana instrukcja programu
- **MBR** – rejestr (jako bufor) do przechowywania bajta danych do wpisania do RAM lub portu IO, albo czytanego z RAM lub portów IO
- **MAR** - rejestr (jako bufor) do przechowywania adresu do RAM lub portów IO
- **CBR** - rejestr dla magistrali kontrolnej, przechowujący jej aktualny tryb działania



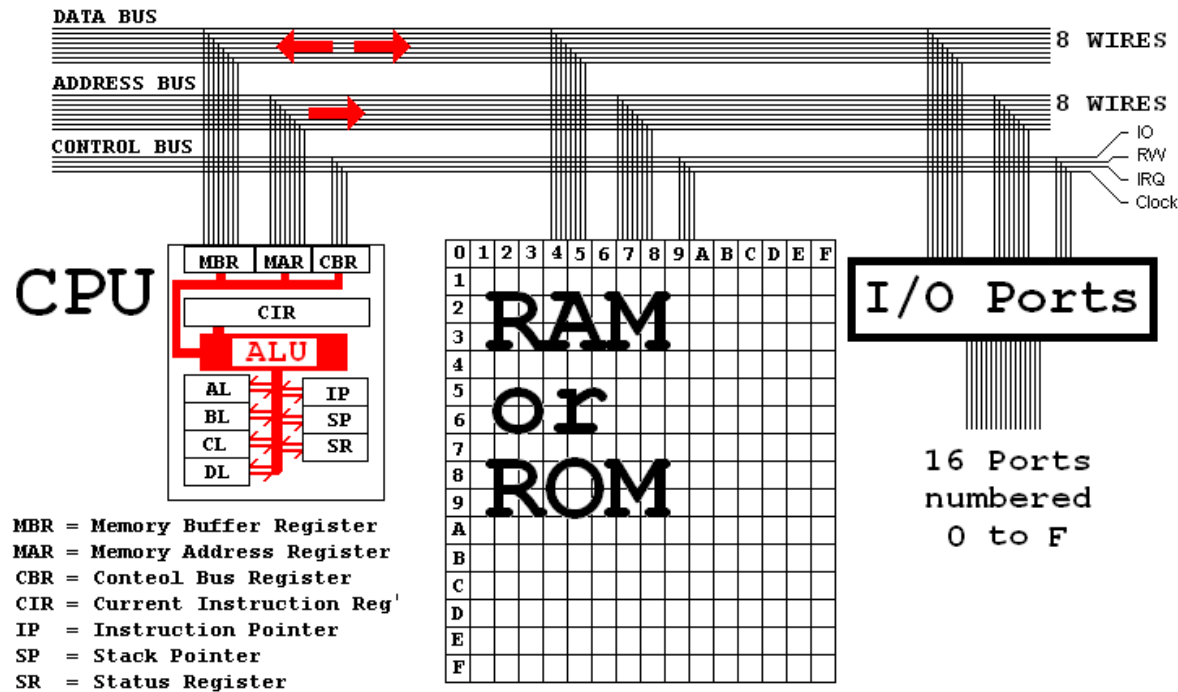
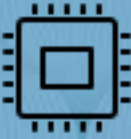
- **RAM - Random Access Memory**
- Symulator posiada 256 bajtowych komórek pamięci RAM. Komórki te są adresowane od 0 do 255, ale zapisywane heksadecymalnie w nawiasach kwadratowych od [00] do [FF].
- Program jest umieszczany na początku pamięci RAM.



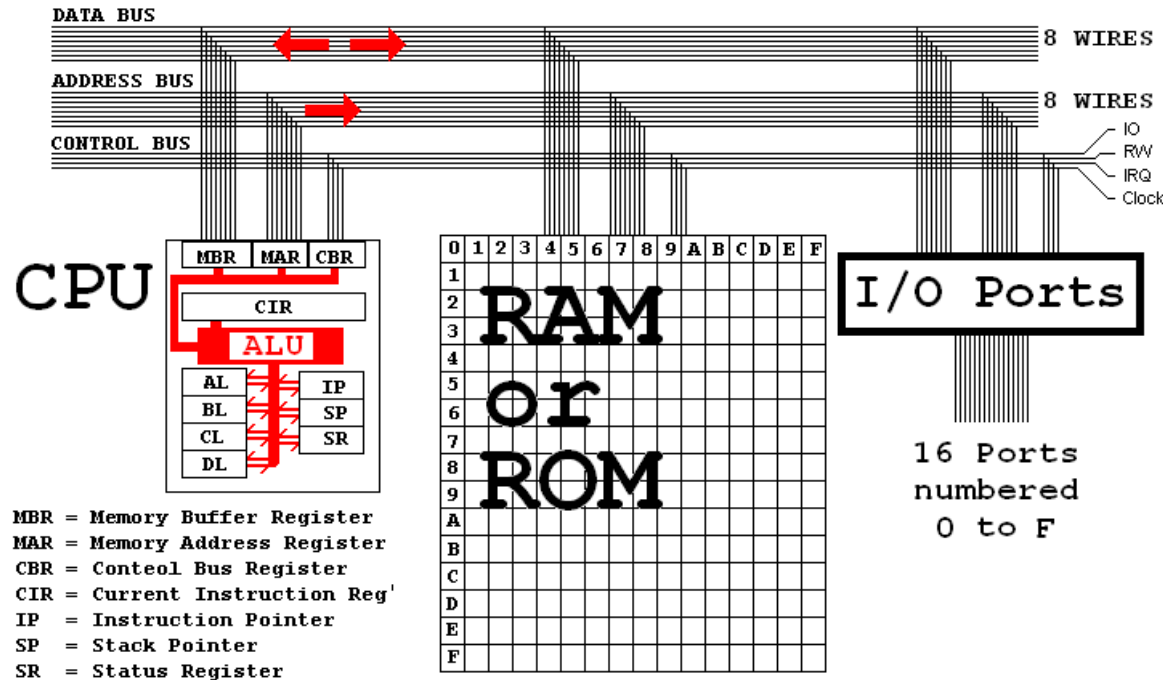
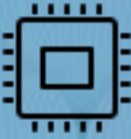
Microprocessor Simulator – magistrale



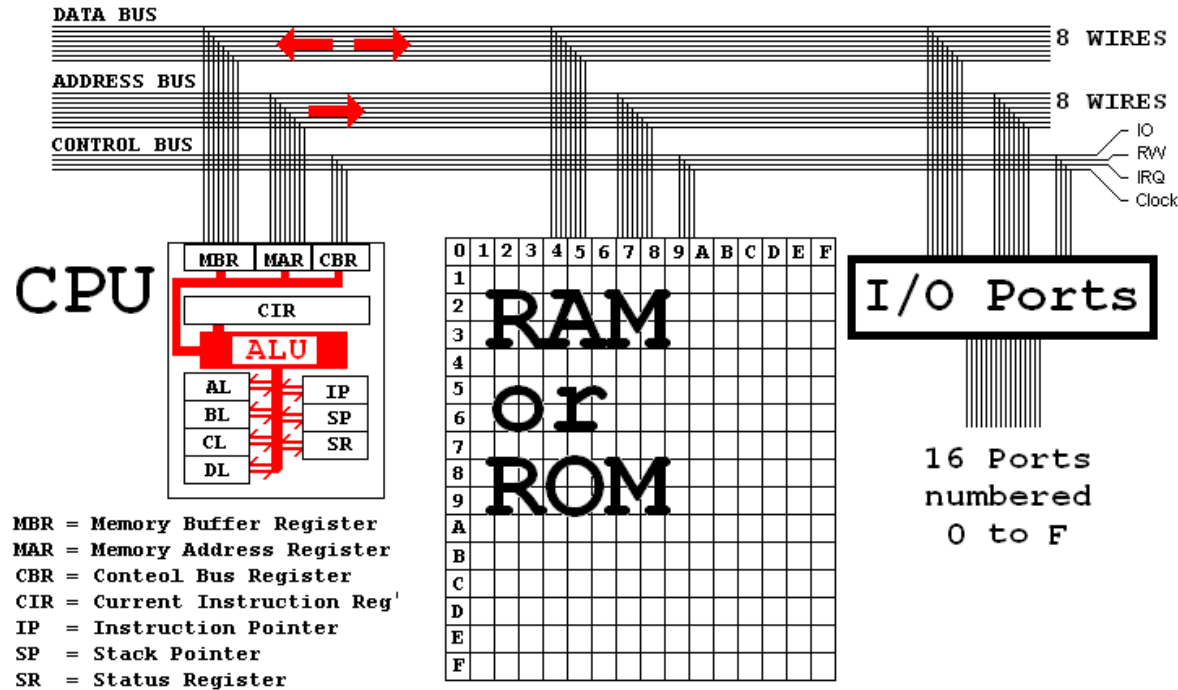
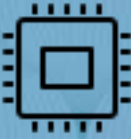
- **Magistrale - Busses**
- Magistrale (inaczej szyny) to wiązki przewodów do przekazywania impulsów elektrycznych kodujących bity. W chipach (układach scalonych) są to równoległe ścieżki miedziane. W 8-bitowych procesorach magistrale składają się z 8 przewodów. Ale 64-bitowe procesory mają magistrale złożone z 64 przewodów.



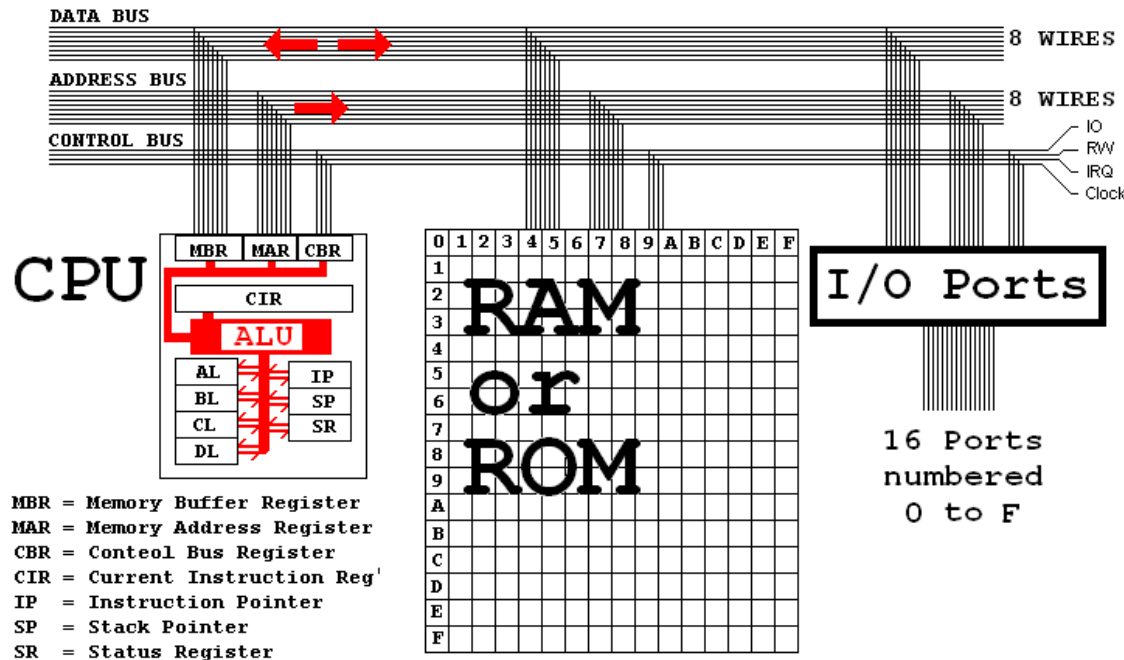
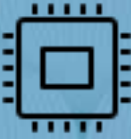
- **Magistrale – Busses.** Magistrale (inaczej szyny) to wiązki przewodów do przekazywania impulsów elektrycznych kodujących bity.
- **Magistrala danych -Data Bus**
- Służy do przekazywania danych (w postaci pojedynczych bajtów) pomiędzy CPU, RAM oraz portami IO. Symulator ma 8-bitową magistralę danych



- **Magistrale – Busses.** Magistrale (inaczej szyny) to wiązki przewodów do przekazywania impulsów elektrycznych kodujących bity.
- **Magistrala adresowa - Address Bus**
- Służy do lokalizacji komórki w RAM (lub portu IO) na podstawie adresu (lub numeru portu). Symulator ma 8-bitową magistralę adresową



- **Magistrale – Busses.** Magistrale (inaczej szyny) to wiązki przewodów do przekazywania impulsów elektrycznych kodujących bity.
- **Magistrala kontrolna - Control Bus.** Służy do wyznaczania dostępu albo do RAM albo do portów IO, oraz do wyznaczania trybu: wpisywanie do RAM (albo portów IO) albo czytanie z RAM albo z portów IO. IRQ do przerwań. RW do read-write
- Zegar (System Clock) poprzez odpowiedni przewód przekazuje impulsy służące do taktowania pracy CPU.



- **Magistrale – Busses.** Magistrale (inaczej szyny) to wiązki przewodów do przekazywania impulsów elektrycznych kodujących bity.
- **Przerwania sprzętowe - Hardware Interrupts.** Wymagają co najmniej jednego przewodu do powiadamiania CPU, że nastąpiło przerwanie, tj. urządzenie zewnętrzne podłączone do portu wejściowego (input) żąda dostępu do CPU, poprzez wykonanie odpowiedniego kodu (procedury) związanej z tym urządzeniem poprzez numer przerwania przypisany temu urządzeniu. Rzeczywiste procesory mają znacznie więcej przewodów oraz więcej numerów przerwań.

Dziękuję za uwagę!

Slajdy na podstawie wykładów prof. Stanisława Ambroszkewicza

Tło obrazka autorstwa rawpixel.com – pobrane z serwisu [Freepik](#)
[Memory Slot](#) icon by [Icons8](#)
[Electronics](#) icon by [Icons8](#)