8 bites CPU Szimulátor

A szimulátor itt érhető el: https://schweigi.github.io/assembler-simulator/

Ez egy **egyszerű oktatási célú 8 bites CPU szimulátor**, amely lehetővé teszi, hogy az assembly programozás alapjait biztonságosan, interaktív módon gyakoroljuk.

---

1. példa – Karakter kiírása a képernyőre

A legegyszerűbb program: egy betű megjelenítése.

```
    JMP start

start:
    MOV D, 232    ; a képernyő címe
    MOV [D], 'A'  ; írjunk ki egy 'A' betűt
    HLT           ; állítsuk le a programot
```

flowchart TD A[Start] --> B[Beállítjuk a D címet: 232] B --> C[Képernyő memóriába írjuk az A-t] C --> D[Program leáll: HLT]

---

2. példa – Egyszerű számolás

Töltsünk be egy számot a regiszterbe, adjunk hozzá és írjuk ki.

```
    JMP start

start:
    MOV A, 5      ; töltsünk be egy értéket A-ba
    ADD A, 3      ; adjunk hozzá 3-at (A = 8)
    MOV D, 232
    MOV [D], '0' + A   ; írjuk ki a szám ASCII kódját
    HLT
```

flowchart TD A[Start] --> B[Regiszter A = 5] B --> C[A = A + 3 → 8] C --> D[Képernyő cím: 232] D --> E[Képernyőre írjuk '8'-at] E --> F[HLT]

---

3. példa – Ciklus használata

Írjunk ki egymás után több karaktert (pl. 5 db 'X').

```
    JMP start
```

```
start:
    MOV D, 232    ; képernyő kezdőcím
    MOV B, 5      ; ismétlések száma

loop:
    MOV [D], 'X'  ; írjunk ki egy 'X'-et
    INC D         ; lépjünk a következő helyre
    DEC B         ; csökkentsük a számlálót
    JNZ loop      ; ha nem nulla, folytassuk
    HLT
```

flowchart TD A[Start] --> B[D = 232, B = 5] B --> C[Kiírunk 'X'-et] C --> D[D = D + 1] D --> E[B = B - 1] E --> F{B != 0?} F -->|Igen| C F -->|Nem| G[HLT]

---

4. példa – Szám bináris kiírása

Ez már egy összetettebb program: egy regiszterben tárolt számot írunk ki binárisan.

```
    JMP start

display_binary:
    MOV D, 232
    MOV B, 0
next:
    SHL A, 1
    JC print_1
    MOV [D], '0'
    JMP next2
print_1:
    MOV [D], '1'
next2:
    INC D
    INC B
    CMP B, 8
    JNZ next
    RET

start:
    MOV A, 54    ; írjuk ki a 54-et binárisan
    CALL display_binary
    HLT
```

flowchart TD A[Start] --> B[A = 54] B --> C[Hívjuk: display_binary] C --> D[B = 0, D = 232] D --> E[SHL A, 1 → Carry?] E -->|Carry=0| F[Kiírunk '0'] E -->|Carry=1| G[Kiírunk '1'] F --> H[D++, B++] G --> H[D++, B++] H --> I{B < 8?} I -->|Igen| D I -->|Nem| J[RET → vissza start-ba] J --> K[HLT]

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:8_bit_cpu_simulator

Last update: **2025/09/24 06:24**