

Symulator poleceń Intel 8086

Autor: Bruno Talaga

Spis treści

I. Wstęp.....	3
II. Intel 8086.....	4
III. Funkcjonalność symulatora	5
III.I. Instrukcja MOV	5
III.II. Instrukcja XCHG.....	9
III.III. Instrukcja PUSH	11
III.IV. Instrukcja POP.....	12
III.V. Adresowanie pamięci.....	13
III.VI. Dodatkowe funkcje	16
IV. Informacje dodatkowe na temat symulatora	20

I. Wstęp

Program został opracowany jako projekt zaliczeniowy z przedmiotu Architektura Systemów Komputerowych. Ma on na celu zasymulowanie niektórych poleceń dostępnych dla procesora Intel 8086 (polecenia MOV, XCHG, PUSH, POP), interakcji pomiędzy rejestrami, pamięcią oraz stosem. Symulator został napisany jako aplikacja sieciowa w języku JavaScript, jest hostowany pod adresem brunotalaga.com/CPUsimProject/ a repozytorium jest dostępne pod adresem: github.com/w4ty/CPUsimProject/ gdzie zawarty jest cały kod projektu.

Autorem projektu jest Bruno Talaga.

II. Intel 8086

Intel 8086 jest 16-bitowym mikroprocesorem wyprodukowanym przez firmę Intel w roku 1978, został zaprojektowany jako rozszerzenie 8-bitowego Intel 8080/8085.

Został on wykorzystany przez firmę IBM w komputerze IBM PC w 1981 roku, co silnie wpłynęło na popularność mikroprocesora, czego konsekwencją było rozwinięcie architektury x86, najpopularniejszej architektury procesorów w komputerach personalnych.

Prędkość zegara pierwszego modelu 8086 wynosiła 5MHz, późniejsze modele jednak miały taktowania sięgające 10MHz. Modeli 8086 było siedem, 8086, 8086-1, 8086-2, 8086-4, I8086, M8086, 80C86.

III. Funkcjonalność symulatora

III.I. Instrukcja MOV

Program pozwala na wpisanie wartości do rejestru w systemie heksadecymalnym za pomocą instrukcji MOV:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
01F3	0000	0000	0000

BP: 0000 DI: 0000 SI: 0000 DISP: 0000

Słot (wskaźnik FFEE):

Pole komend

AX

1 2

MOV AX 01F3

Tryb adresowania pamięci:

☐ Indeksowy ☐ Bazowy ☒ Indeksowo-bazowy

Wybierz tryb adresacji:

1 2

Log operacji:

MOV AX, 01F3

Pamięć:

Wartość musi zawsze być liczbą heksadecymalną o długości czterech znaków, w przypadku próby wpisania wartości niepoprawnej, program zignoruje komendę i wypisze błąd wskazujący na nieprawidłowy argument:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP: 0000 DI: 0000 SI: 0000 DISP: 0000

Słot (wskaźnik FFEE):

Pole komend

AX

1 2

MOV AX 333

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

Wybierz tryb adresacji:

DI

1 2

Log operacji:

MOV AX, 333
INVALID ARGUMENT "333"

Pamięć:

Możliwe jest wypełnienie rejestrów losowymi wartościami oraz ich wyzerowanie:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
E7EF	0048	C61E	2F45

BP	DI	SI	DISP
0000	0000	0000	0000

Stos (wskaźnik FFEE):

Pole komend

AX

1 ↙ ↘ 2

MOV

DI

1 ⊕ 2

Tryb adresowania pamięci:

☒ Indeksowy
 ☐ Bazowy
 ☐ Indeksowo-bazowy

Log operacji:

MOV AX, E7EF
 MOV BX, 0048
 MOV CX, C61E
 MOV DX, 2F45

Pamięć:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP	DI	SI	DISP
0000	0000	0000	0000

Stos (wskaźnik FFEE):

Pole komend

AX

1 ↙ ↘ 2

MOV

DI

1 ⊕ 2

Tryb adresowania pamięci:

☒ Indeksowy
 ☐ Bazowy
 ☐ Indeksowo-bazowy

Log operacji:

MOV AX, E7EF
 MOV BX, 0048
 MOV CX, C61E
 MOV DX, 2F45
 MOV AX, 0000
 MOV BX, 0000
 MOV CX, 0000
 MOV DX, 0000

Pamięć:

Możliwe jest wpisywanie wartości jednego rejestru do drugiego:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
F3E9	8ABE	EA61	A888

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

DX

1 2

MOV AX DX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

Pamięć:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
A888	8ABE	EA61	A888

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

DX

1 2

MOV AX DX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

MOV AX, DX

Pamięć:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
A888	8ABE	EA61	A888

BP	DI	SI	DISP
EA61	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

DX

1 2

MOV BP CX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

MOV AX, DX
MOV BP, CX

Pamięć:

Dozwolone jest także wpisywanie do pamięci wartości heksadecymalnej lub wartości rejestru i wypisywanie z pamięci do rejestru:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
2971	2C50	9FDD	A490

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

AX ▾

1 ↙
2 ↘

MOV ▾

[DI+DISP]

AX

⌂
⌂

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI ▾

1 ⌂
2 ⌂

Log operacji:

MOV [DI+DISP], AX

Pamięć:

0000:71
0001:29

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
2971	2971	9FDD	A490

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

BX ▾

1 ↙
2 ↘

MOV ▾

BX

[DI+DISP]

⌂
⌂

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI ▾

1 ⌂
2 ⌂

Log operacji:

MOV [DI+DISP], AX
MOV BX, [DI+DISP]

Pamięć:

0000:71
0001:29

III.II. Instrukcja XCHG

Zaimplementowana jest instrukcja XCHG, pozwalająca na zamianę wartości rejestrów i pamięci:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	2971	9FDD	AA90

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

CX

1 2

XCHG CX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

Pamięć:

0000:71
0001:29

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	9FDD	2971	AA90

BP	DI	SI	DISP
0000	0000	0000	0000

Słot (wskaźnik FFEE):

Pole komend

CX

1 2

XCHG CX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

XCHG CX, BX

Pamięć:

0000:71
0001:29

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	2971	2971	AA90



BP	DI	SI	DISP
0000	0000	0000	0000



Stos (wskaźnik FFEE):



Pole komend

CX

1 2

XCHG [DI+DISP], BX

1 2

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

XCHG CX, BX
XCHG [DI+DISP], BX



Pamięć:

0000 DD
0001 9F



Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	0000	2971	AA90



BP	DI	SI	DISP
2971	0000	0000	0000



Stos (wskaźnik FFEE):



Pole komend

CX

1 2

XCHG BP, BX

1 2

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

XCHG CX, BX
XCHG [DI+DISP], BX
XCHG BP, BX



Pamięć:

0000 DD
0001 9F



III.III. Instrukcja PUSH

Program symuluje działanie stosu, co pozwala użyć instrukcji PUSH do wpisania danych z rejestru do stosu:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	0000	2971	AA90

BP: 2971, DI: 0000, SI: 0000, DISP: 0000

Stos (wskaźnik: FFEE):

Pole komend

1 2

PUSH AX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

1 2

Log operacji:

Pamięć:

0000:DD
0001:9F

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	0000	2971	AA90

BP: 2971, DI: 0000, SI: 0000, DISP: 0000

Stos (wskaźnik: FFEC):

FFEC: 71
FFED: 29

Pole komend

1 2

PUSH AX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

1 2

Log operacji:

PUSH AX

Pamięć:

0000:DD
0001:9F

III.IV. Instrukcja POP

Dzięki implementacji stosu możliwe jest także użycie instrukcji POP do wypisania wartości ze stosu (wskazywanej przez wskaźnik stosu) do dowolnego rejestru:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	0000	2971	AA90

BP	DI	SI	DISP
2971	0000	0000	0000

Słot (wskaźnik FFEC):

FFEC: 71
FFFD: 29

Pole komend

BX

1 2

POP BX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

PUSH AX

Pamięć:

0000: DD
0001: 9F

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
2971	2971	2971	AA90

BP	DI	SI	DISP
2971	0000	0000	0000

Słot (wskaźnik FFEC):

FFEC: 71
FFFD: 29

Pole komend

BX

1 2

POP BX BX

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

PUSH AX
POP BX

Pamięć:

0000: DD
0001: 9F

III.V. Adresowanie pamięci

Program symuluje i oblicza adresy pamięci korzystając z trzech trybów adresowania pamięci: indeksowego, bazowego oraz indeksowo-bazowego:

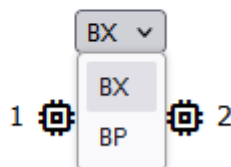
Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy



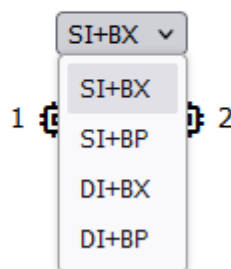
Tryb adresowania pamięci:

☐ Indeksowy ☒ Bazowy ☐ Indeksowo-bazowy



Tryb adresowania pamięci:

☐ Indeksowy ☐ Bazowy ☒ Indeksowo-bazowy



Przykładowo chcąc wpisać wartość FFCE dla różnych trybów adresowania pamięci:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP	DI	SI	DISP
0001	0002	0003	0010

Słot (wskaźnik FFCE):

Pole komend

AX ▼

1 ↙
↘ 2

MOV ▼

[DI+DISP]

FFCE

⌂
⌂

Tryb adresowania pamięci:

☒ Indeksowy
 ☐ Bazowy
 ☐ Indeksowo-bazowy

DI ▼

1 ⌂
⌂ 2

Log operacji:

MOV [DI+DISP], FFCE

Pamięć:

0012 CE
0013 FF

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP	DI	SI	DISP
0001	0002	0003	0010

Słot (wskaźnik FFCE):

Pole komend

AX ▼

1 ↙
↘ 2

MOV ▼

[BX+DISP]

FFCE

⌂
⌂

Tryb adresowania pamięci:

☐ Indeksowy
 ☒ Bazowy
 ☐ Indeksowo-bazowy

BX ▼

1 ⌂
⌂ 2

Log operacji:

MOV [DI+DISP], FFCE
MOV [BX+DISP], FFCE

Pamięć:

0012 CE
0013 FF
0010 CE
0011 FF

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP	DI	SI	DISP
0001	0002	0003	0010

Słot (wskaźnik FFCE):

Pole komend

AX ▼

1 ↙
↘ 2

MOV ▼

[SI+BP+DISP]

FFCE

⌂
⌂

Tryb adresowania pamięci:

☐ Indeksowy
 ☐ Bazowy
 ☒ Indeksowo-bazowy

SI+BP ▼

1 ⌂
⌂ 2

Log operacji:

MOV [DI+DISP], FFCE
MOV [BX+DISP], FFCE
MOV [SI+BP+DISP], FFCE

Pamięć:

0012 CE
0013 FF
0010 CE
0011 FF
0014 CE
0015 FF

Do wyliczenia adresu pamięci można także stosować dowolne wartości heksadecymalne o długości czterech znaków dodawane operatorem „+”:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Instrukcja

Tabela wartości

AX	BX	CX	DX
0000	0000	0000	0000

BP	DI	SI	DISP
0001	0002	0003	0010

Stos (wskaznik: FFFE):

Pole komend

AX

1 2

MOV [005A+0F21] FFCE

Tryb adresowania pamięci:

☐ Indeksowy ☐ Bazowy ☒ Indeksowo-bazowy

SI+BP

1 2

Log operacji:

```
MOV [DI+DISP], FFCE
MOV [BX+DISP], FFCE
MOV [SI+BP+DISP], FFCE
MOV [005A+0F21], FFCE
```

Pamięć:

```
0012 CE
0013 FF
0010 CE
0011 FF
0014 CE
0015 FF
007b CE
007c FF
```

III.VI. Dodatkowe funkcje

Program pozwala szybko zmienić wartości kilku rejestrów poprzez wpisanie do tabeli oczekiwanych wartości:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP **DI** **SI** **DISP**

0000	0000	0000	0000
------	------	------	------

Słot (wskaźnik FFFE):

Pole komend

AX

1 2

MOV

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

MOV AX, 00FF
MOV BX, 3421
MOV CX, ABCD
MOV DX, 5679

Pamięć:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP **DI** **SI** **DISP**

0001	0002	0003	0004
------	------	------	------

Słot (wskaźnik FFFE):

Pole komend

AX

1 2

MOV

Tryb adresowania pamięci:

☒ Indeksowy ☐ Bazowy ☐ Indeksowo-bazowy

DI

1 2

Log operacji:

MOV AX, 00FF
MOV BX, 3421
MOV CX, ABCD
MOV DX, 5679
MOV BP, 0001
MOV DI, 0002
MOV SI, 0003
MOV DISP, 0004

Pamięć:

Pamięć może zostać wyczyszczona klikając odpowiedni przycisk:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP	DI	SI	DISP
0001	0002	0003	0004

Słot (wskaźnik: FFFE):

Pole komend

BX ▾

1 ✓
2 ✗

MOV ▾

↺
↻

Tryb adresowania pamięci:

☐ Indeksowy
 ☐ Bazowy
 ☒ Indeksowo-bazowy

SHBX ▾

1 Ⓜ
2 Ⓜ

Log operacji:

Pamięć:

3428 FF
 3429 00
 0006 FF
 0007 00

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP	DI	SI	DISP
0001	0002	0003	0004

Słot (wskaźnik: FFFE):

Pole komend

BX ▾

1 ✓
2 ✗

MOV ▾

↺
↻

Tryb adresowania pamięci:

☐ Indeksowy
 ☐ Bazowy
 ☒ Indeksowo-bazowy

SHBX ▾

1 Ⓜ
2 Ⓜ

Log operacji:

CLEARED MEMORY

Pamięć:

Tak samo stos, resetuje to także wskaźnik stosu:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP	DI	SI	DISP
0001	0002	0003	0004

Stos (wskaźnik: FFF6):

```

FFF6: 79
FFF7: 56
FFF8: CD
FFF9: AB
FFFA: 21
FFFB: 34
FFFC: FF
FFFD: 00
        
```

Pole komend

DX

1 ↙ ↘ 2

PUSH ▾ DX

⌂ ⌕

Tryb adresowania pamięci:

☐ Indeksowy ☐ Bazowy ☒ Indeksowo-bazowy

SHBX ▾

1 ⌂ ⌕ 2

Log operacji:

```

CLEARED MEMORY
PUSH AX
PUSH AX
POP AX
PUSH BX
PUSH CX
PUSH DX
        
```

Pamięć:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP	DI	SI	DISP
0001	0002	0003	0004

Stos (wskaźnik: FFFE):

Pole komend

DX

1 ↙ ↘ 2

PUSH ▾ DX

⌂ ⌕

Tryb adresowania pamięci:

☐ Indeksowy ☐ Bazowy ☒ Indeksowo-bazowy

SHBX ▾

1 ⌂ ⌕ 2

Log operacji:

```

CLEARED MEMORY
PUSH AX
PUSH AX
POP AX
PUSH BX
PUSH CX
PUSH DX
CLEARED STACK
        
```

Pamięć:

Log operacji też można wyczyścić:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX	DX
00FF	3421	ABCD	5679

BP	DI	SI	DISP
0001	0002	0003	0004

Słot (wskaźnik: FFFE):

Pole komend

DX ▾

1 ✓
2 ✗

PUSH ▾ DX

⌂
⌂

Tryb adresowania pamięci:

☐ Indeksowy
 ☐ Bazowy
 ☒ Indeksowo-bazowy

SHBX ▾

1 ⌂
2 ⌂

Log operacji:

Pamięć:

Program ma wbudowaną instrukcję tłumaczącą działanie instrukcji MOV, XCHG, PUSH i POP, oraz jak można budować argumenty:

Symulator podstawowych instrukcji x86 - Bruno Talaga

Tabela wartości

AX	BX	CX
00FF	3421	ABCD

BP	DI	SI
0001	0002	0003

Słot (wskaźnik: FFFE):

Instrukcja

ARG1 - zmienna, z której przenosimy wartość.
Dozwolone typy: REJESTR, WARTOŚĆ (HEXADECYMALNA), ADRES PAMIĘCI

- XCHG - zamienia wartości dwóch zmiennych**
 Przyjmuje dwa argumenty: XCHG ARG1, ARG2
 ARG1 - zmienna, do której przenosimy wartość
 Dozwolone typy: REJESTR, ADRES PAMIĘCI
 ARG2 - zmienna, z której przenosimy wartość
 Dozwolone typy: REJESTR, ADRES PAMIĘCI
- PUSH - wrzuca wartość zmiennej na stos**
 Przyjmuje jeden argument: PUSH ARG1
 ARG1 - zmienna, którą wrzucamy na stos
 Dozwolone typy: REJESTR
- POP - zdejmuje wartość ze stosu i przypisuje ją do zmiennej**
 Przyjmuje jeden argument: POP ARG1
 ARG1 - zmienna, do której przypisujemy wartość ze stosu
 Dozwolone typy: REJESTR

Oznaczenia typów w polu argumentów:

- REJESTR** - AX, BX, CX, DX, BP, DI, SI, DISP
Wielkość liter ma znaczenie
- WARTOŚĆ (HEXADECYMALNA)** - 0000 - FFFF
Wartość musi być zapisana w formacie heksadecymalnym oraz zawierać musi składnik: sia z 4 znaków, rozdzieloną: 0000

IV. Informacje dodatkowe na temat symulatora

Symulator został napisany w języku JavaScript, repozytorium zawierające cały kod jest dostępne na platformie Github.

Całość programu jest oparta na interpretowaniu argumentów wpisanych w pola tekstowe w zakładce „Pole komend”, pomimo tego, że symulator zawiera przyciski pozwalające na automatyczne wpisanie wybranych rejestrów i adresów, wszystko użytkownik może wpisać samemu w odpowiednie pola tekstowe, co program po uruchomieniu komendy zinterpretuje.

Nazwy rejestrów muszą zawsze zostać wpisane dokładnie tak samo jak nazwane są w rzeczywistości, co oznacza, że przykładowo wpisanie „ax” zamiast „AX” spowoduje błąd niepoprawnego argumentu. Liczby muszą być podawane jako heksadecymalne o długości czterech znaków, w ich przypadku wielkość liter nie ma znaczenia, więc zarówno „ffff” jak i „FFFF” będą poprawne. Liczenie adresu pamięci zawsze odbywa się kwadratowych nawiasach „[]”, dozwolone są nazwy rejestrów (aby pobrać z nich wartość do obliczeń), liczby heksadecymalne oraz operator „+”. Wszystkie argumenty obliczeń adresu pamięci trzymają się takich samych zasad jak zwykłe argumenty.

Cały program jest lekki, pobierając trochę ponad 150kB w wersji hostowanej na stronie internetowej, z czego około 130kB to ikony używane w przyciskach (w momencie pisania sprawozdania).

Wersja hostowana na stronie brunotalaga.com/CPUsimProject/ korzysta z serwerów Github Pages i jest automatycznie aktualizowana wraz z branch'em master, co oznacza, że rozwijanie projektu jest łatwe i szybkie.