



Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 4

З дисципліни «Архітектура комп'ютерів-1»

На тему «ОБРОБКА ІНФОРМАЦІЇ В ЕОМ НА ПРОГРАМНОМУ  
І МІКРОПРОГРАМНОМУ РІВНЯХ»

Виконав:  
студент 2 курсу ФІОТ  
групи ІВ-71  
Мазан Я. В.  
Залікова – 7109

ПЕРЕВІРИВ:  
доц. Верба О. А.

### Теоретичні відомості:

Системою команд процесора називають сукупність всіх команд, які може виконувати процесор. Залежно від кількості реалізованих команд комп'ютери прийнято ділити на два основних типи:

RISC – з скороченою системою команд;

CISC – з комплексною системою команд.

Для RISC-систем кількість команд визначається десятками, а для CISC – сотнями.

Команда – інформаційне слово, що в загальному випадку містить такі дані:

- операція, що виконується з операндами;
- місце розташування операндів в пам'яті;
- місце запису результату операції;
- спосіб визначення адреси наступної команди.

### Мета:

вивчити етапи формування системи команд процесорів, різновиди форматів команд та способів адресації операндів. Навчитися розробляти мікроалгоритми і мікропрограми виконання етапів команд з використанням мнемонічного мікроасемблера.

### Завдання:

$7409_{10} \rightarrow 1110011110001_2$

$a_3$	$a_2$	$a_1$	Операція
0	0	1	X - Y

$a_7$	$a_6$	$a_5$	Адреси портів ЗП	
			РС	РД
1	1	1	72H	74H

a <sub>4</sub>	Тип адресації
0	Пряма

a <sub>2</sub>	a <sub>1</sub>	Спосіб множення
0	1	2-й

$F = (X - Y) \cdot Z;$

**Система команд для моєї мікропрограми:**

Мікроманда	Опис	Код мікрокоманди
test	Перевірка на готовність зоанішнього пристрою	0001
throw	Повернення з мікропрограми при неготовності ЗП	0010
input	Введення даних із ЗП	0011
subb	Віднімання	0100
mul	Множення 2-м способом	0101
res	Запис результату обчислень	0110
end	Вихід	0111

**Код програми:**

```
\FORMULA F = (x - y) * z

\nastroiyka shemy
link l1:ct
link l2:rdm
LINK l3:rrd
link ewh:10 \mladshiy razriad starshego adresa
link M: z,z,z,z,z,z,z,z,14,13,12,11
ACCEPT dev[1]:l,072h,074h,12,2 \i - type of devide - input device
ACCEPT dev_buf[1]:0008h \x

\dannye
dw 0001h: 0h\res1
dw 0002h: 0h\res2
dw 0004h: 0072h\input--x address
dw 0006h: 0005h\y
dw 0008h: 0001h\z

DW 0020h:000Ah \ Return addr

\programma
dw 000Ah: 00001000000000100% \test
dw 000Ch: 00010000000100000% \jz
dw 000Eh: 00011000000000100% \input
```

```
dw 0010h: 0010000000000110% \sub
dw 0012h: 0010100000001000% \mul
dw 0014h: 0011000000000001% \res
dw 0016h: 0011100000000000% \end
```

\mikroprogram in microcommands memory

```
ORG 0h {cjp nz,start;}
ORG 1h {cjp nz,test;}
ORG 2h {cjp nz,throw;}
ORG 3h {cjp nz,input;}
ORG 4h {cjp nz,subb;}
ORG 5h {cjp nz,mul;}
ORG 6h {cjp nz,res;}
ORG 7h {cjp nz,end;}

```

\ustanovka nachalnogo adresa

```
start {and r13, r13, z;}
      {or r13, r13, 000Ah;}

```

\chtenie comandy

```
begin {ewh; oey; xor nil, r13, r13;}
      {ewl; oey; or nil, r13, z;}
      {cjp rdm, cp; r; or r14, bus_d, z;}

```

\raspakovka comandy

\proverka formata

```
{and nil, r14, 8000h; load rm, flags;}
{cjp not rm_z, end;}

```

\proverka tipa adresazyi

```
{and nil, r14, 0400h; load rm, flags;}
{cjp not rm_z, end;}

```

\zagruzka adresa registra v RB

```
{oey;or nil,r14,z;load rb;load ra;}

```

\perekhod po kodu operacyi

```
{oey;or nil, r14,z;jmap;}

```

\mikroprogramy komand

FromMem

```
{or nil,r14,z;ewl;oey;}
{r;or r12,bus_d,z;cjp rdm,cp;}
{crtm nz;}

```

\ Command test

```
test
{cjs nz,FromMem;}
{or nil,r12,z;oey;ewl;}
{i;cjp rdd,cp;or r12,bus_d,z;}
{or nil,r12,z;load rm,flags;}
{cjp nz,formadd;}

```

\ Command jz [mem]

```
throw
      {cjp not m_z,no_j;}
      {cjs nz,FromMem;}

```

```

    {or r13,r12,z;}
    {cjp nz,begin;}
no_j {cjp nz,formadd;}

```

```

\ Command in r2,[mem]
input
{cjs nz,FromMem;}
{add r12,r12,2,z;}
{or nil,r12,z;ewl;oeY;}
{i;cjp rdd,cp;or r2,bus_d,z;}
{cjp nz,formadd;}

```

\-----mul-----

```

mul {cjs nz,FromMem;}
{or r5, r12, z;}
    {or R5, 8000h;}          \ set marker bit in r1(X)

    {and nil, R2, 8000h; load RM, FLAGS;} \ set all bit 1 in R1 if -Y
    {cjp RM_Z, MMUL;}          \ or do nothing
    {or R1, 0FFFFh;}          \

```

```

MMUL {and nil, R5, 0001h; load RM, FLAGS;} \ Check if 0 bit X == 0
    {cjp RM_Z, link1;}          \ and jump link1 if TRUE

    {add R4, R4, R2, RM_C; load RM, FLAGS;} \ SUM (R3-R4)=(R3-R4)+Y
    {add R3, R3, R1, RM_C;}      \

```

```

link1 {add SRL, R5, Z;}          \ shift registers
    {add SLL, R2, Z;}          \ RY< RX>

```

```

    {add SL.25, R1, 0h;}        \

```

```

    {xor R6, R6;}              \ Check if x = 0
    {or R6, R5;}              \ (without sing bits)
    {and R6, 0FFFCh; load RM, FLAGS;} \

```

```

    {cjp not RM_Z, MMUL;}      \ if not repeat sum

```

```

    {and nil, R5, 0001; load RM, FLAGS;} \ if X.0 = 1
    {cjp RM_Z, formadd;}          \ then add X
    {xor R2, 0FFFFh;}            \ first need convert it
    {xor R1, 0FFFFh;}            \ from -X to +X
    {add R2, R2, 0001h; load RM, FLAGS;} \ then simple ADD
    {add R1, R1, Z, RM_C;}
    {add R4, R4, R2, Z; load RM, FLAGS;}
    {add R3, R3, R1, RM_C;}
    {cjp nz,formadd;}

```

\-----sub-----

```

subb
{cjs nz,FromMem;}
{xor R12, R12, 0FFFFh;}      \ Y = -Y
    {add r2,r2,r12,nz;}      \ X - Y + 1
    {cjp nz,formadd;}

```

\-----res-----

```

res
{cjs nz,FromMem;}
{xor nil, r14, Z; oey; ewl;}
{cjp rdm, CP; W; or nil, Z, r4; oey;}

{add r14,r14,1h,z;}

{xor nil, r14, Z; oey; ewl;}
{cjp rdm, CP; W; or nil, Z, r3; oey;}

{cjp nz,formadd;}

\-----

\formir sled adresa i na vyborku komandy

formadd{add r13,r13,2,z;}
        {cjp nz,begin;}

\-----end-----

\konec programmy

end {}

```

## **Висновок:**

У даній роботі побудований алгоритм множення двох 16-розрядних операндів другим способом. Операнди вводяться з пристрою виведення та виводяться на пристрій виведення. Обробка інформації виконується на програмному, мікропрограмному і апаратному рівнях. Отримані результати моделювання співпадають з прогнозованими. В результаті виконання цієї роботи, я пригадав та закріпив теоретичні аспекти цієї теми.