Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

*Виконав:*

Студент групи ІО-32

Білецький М.В.

м. Київ

2015 р.

**ОБРОБКА ІНФОРМАЦІЇ В ЕОМ НА ПРОГРАМНОМУ,   
МІКРОПРОГРАМНОМУ І АПАРАТНОМУ РІВНЯХ**

***Мета роботи*** – вивчити етапи формування системи команд процесорів. Навчитися розробляти мікроалгоритми і мікропрограми виконання етапів команд.

Варіант завдання:

Номер залікової книжки: 322410= 110010011000

A = (X + Y) \* Z;

1 спосіб множення;

Пряма адресація.

Алгоритм обчислень

Лістинг програми

link l1:ct

link l2:rdm

link ewh:16

link M:z,z,z,z,z,z,z,z,14,13,12,11 \bufer, 4-1 of it link to 14-11 of local\_bus

accept dev[1]:I, 12h, 13h, 12, 2 \input device

accept dev[2]:O, 92h, 93h,12, 2 \output device

accept dev\_buf[1]:0fff4h, 24ffh, 31ach

accept r7: 20h \reg of current adr

accept r11:17 \counter register for mul

dw 06h:000eh \x

dw 08h:0007h \y

dw 0Ah:0013h \z

dw 20h: 0001000000000110% \ save x to r15

dw 22h: 0010000000001000% \ r15 (+) y

dw 24h: 0100000000001010% \ mult

dw 26h: 0011000000001010% \ result to z

dw 28h: 0101000000000000% \ END

org 0h {cjp nz, loadFromOP;}

org 2h {cjp nz, saveToReg;}

org 4h {cjp nz, AddXY;}

org 6h {cjp nz, saveToMem;}

org 8h {cjp nz, mult;}

org 0Ah {cjp nz, end;}

loadFromOP

{or nil, r7, r7; ewl; oey;}

{xor nil, r7, r7; ewh; oey;}

{r;cjp rdm, cp; or r8, z, bus\_d;} \write command to r8

{and nil, r8, 8000h; load rm, flags;}

{cjp not rm\_z, end;}

\rasshifrovka

{and nil, r8, 400h;load rm, flags;}

{cjp not rm\_z, end;}

{and r14, r8, 03ffh;} \adr of operand

{or nil, r8, z; oey; JMAP;} \jump to action

ADDXY \popravit

{cjs nz, fromMem;}

{add r15, r15, r14,z;} \x + y

{cjp nz, next;}

saveToReg

{cjs nz, fromMem;}

{or r15, 0, r14;} \r15=operand

{cjp nz, next;}

saveToMem

{or nil, r14,z;ewl;oey;}

{w; cjp rdm, cp; or nil, r15, z;oey;} \mem[i]=operand

{cjp nz, next;}

fromMem

{or nil, r14,z; ewl; oey;}

{r; or r14, bus\_d, z; cjp rdm, cp;} \r14=operand

{crtn nz;}

next

{add r7, r7, 2, z;}

{cjp nz, loadFromOP;} \next command

mult \multiplication

{cjs nz, fromMem;}

{xor r0, r0, r0;}

{cont;load rm, z;} \Обнулення RM

label1 {cjp not rm\_c,label2;} \Аналіз цифри множника

{add r0,r0,r14,z;} \Додавання множеного до R12

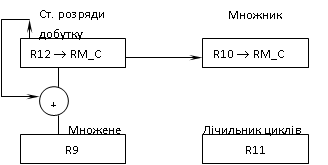
label2 {or srl,r0,r0,z;} \Зсув в R12 і R10 суми

{or sr.9,r15,r15,z;} \часткових добутків

{sub r11,r11,z,z;load rm,flags;cem\_c;} \Декремент і перевірка

{cjp not rm\_z, label1;} \на нуль лічильника R11

Операційна схема множення (1-ий спосіб)



**Висновок:**

У данній лабараторній роботі ми вивчили етапи формування системи команд процесорів та навчилися розробляти мікроалгоритми і мікропрограми виконання етапів команд.