

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота № 4
з дисципліни «Архітектура комп'ютерів»
на тему «*ОБРОБКА ІНФОРМАЦІЇ В ЕОМ НА ПРОГРАМНОМУ,
МІКРОПРОГРАМНОМУ І АПАРАТНОМУ РІВНЯХ*»

Виконав:
студент 2 курсу
ФІОТ гр. ІО-32
Попенко Р.Л.

Київ, 2015 р.

1 Завдання

Розробити архітектуру комп'ютера, систему команд, структурну схему та мікропрограму реалізації етапів виконання команд.

Система команд повинна забезпечувати обчислювання заданого алгебраїчного виразу на програмному рівні. Операнди є цілими 16-розрядними числами (один знаковий та 15 основних розрядів). Операнди вводяться у режимі програмного опиту готовності зовнішніх пристроїв. Результат виводиться у пристрій виводу. Послідовність даних, що вводяться та виводяться, визначається розробником. Система команд містить лише одноадресні команди (рис.1).



Рисунок 1.1 – Структура команди.

Кількість типів команд обирається розробником, виходячи з необхідності реалізації заданих обчислень з урахуванням вводу та виводу даних, а також перевірки готовності зовнішніх пристроїв. Введення кожного типу команд у систему команд повинно бути обґрунтованим. Зокрема, система команд повинна містити команди виконання заданих операцій, вводу та виводу даних, команди умовних та безумовних переходів, пересилання даних.

Обробка інформації на мікро програмному рівні повинна містити вибірку команди, розпакування команди, виконання операції та формування адреси наступної команди. Однакові блоки мікрокоманд бажано оформляти у вигляді мікропідпрограм. При програмному опитуванні дані регістру стану зовнішнього пристрою мають формат, зображений на рис.2.



Рисунок 1.2 – Формат регістру стану зовнішнього пристрою

Початкові умови згідно з варіантом

Номер залікової книжки: 110010010010							
a ₈	a ₇	a ₆	a ₅	a ₄	a ₃	a ₂	a ₁
1	0	0	1	0	0	1	0
	Адреса РС			Адресація	Операція		
	ПВв	ПВив			(X & Y) * Z		
	12h	92h			Спосіб множення		
	Адреса РД			Пряма	Зій		
	13h	93h					

2 Архітектура та алгоритм обчислення

Структура НОЗП			Структура ПМК	
R0	Результат	РЗП (регістри загального призначення)	0	Вектори команд
⋮			⋮	
			15H	
R7	ЛК	Робочі регістри	⋮	Мікропрограма
R8	РК		20H	
⋮			⋮	
			66H	
R14	адреса операнда			
R15	Операнд			
RQ				

Рисунок 2.1 Модель програміста

Структура ОП		Структура ЗП	
00000H	Дані	0000H	
⋮		⋮	
0006H		0012H	РС (пристрій вводу)
0008H		0013H	РД (пристрій вводу)
00AH			
⋮		⋮	
0020H	Програма обчислення	⋮	
0022H		0092H	РС (пристрій виводу)
0024H			
0026H			
0028H			
⋮		0093H	РД (пристрій виводу)
FFFFFFH			
		⋮	
		FFFFFFH	

Рисунок 2.2 Розподіл загальної пам'яті

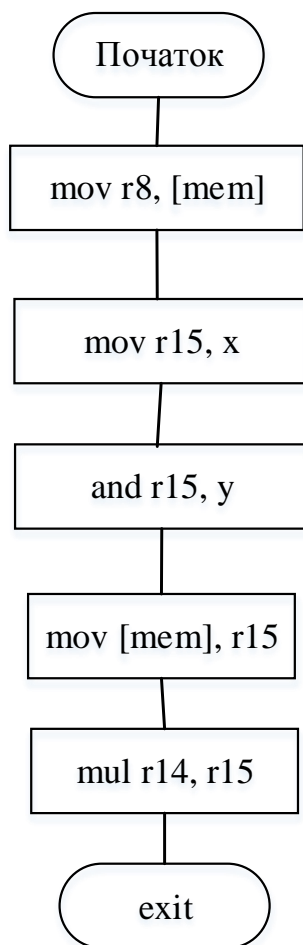


Рисунок 2.3 Алгоритм обчислень

3 Програма обчислень

Для реалізації програми обчислень потрібні такі команди

Команда	Код	Операція
mov r8, [mem]	0 0 0 0	Зчитування команди з ЗП
mov r15, x	0 0 1 0	Запис операнда у регістр
and r15, y	0 1 0 0	Логічне множення операндів
mov [mem], r15	0 1 1 0	Запис операнду у ЗП
mul r14, r15	1 0 0 0	Множення операндів
exit	1 1 0 0	Вихід з програми

Таблиця 3.1 Програма обчислень

4 Структура комп'ютера та мікропрограма обчислень

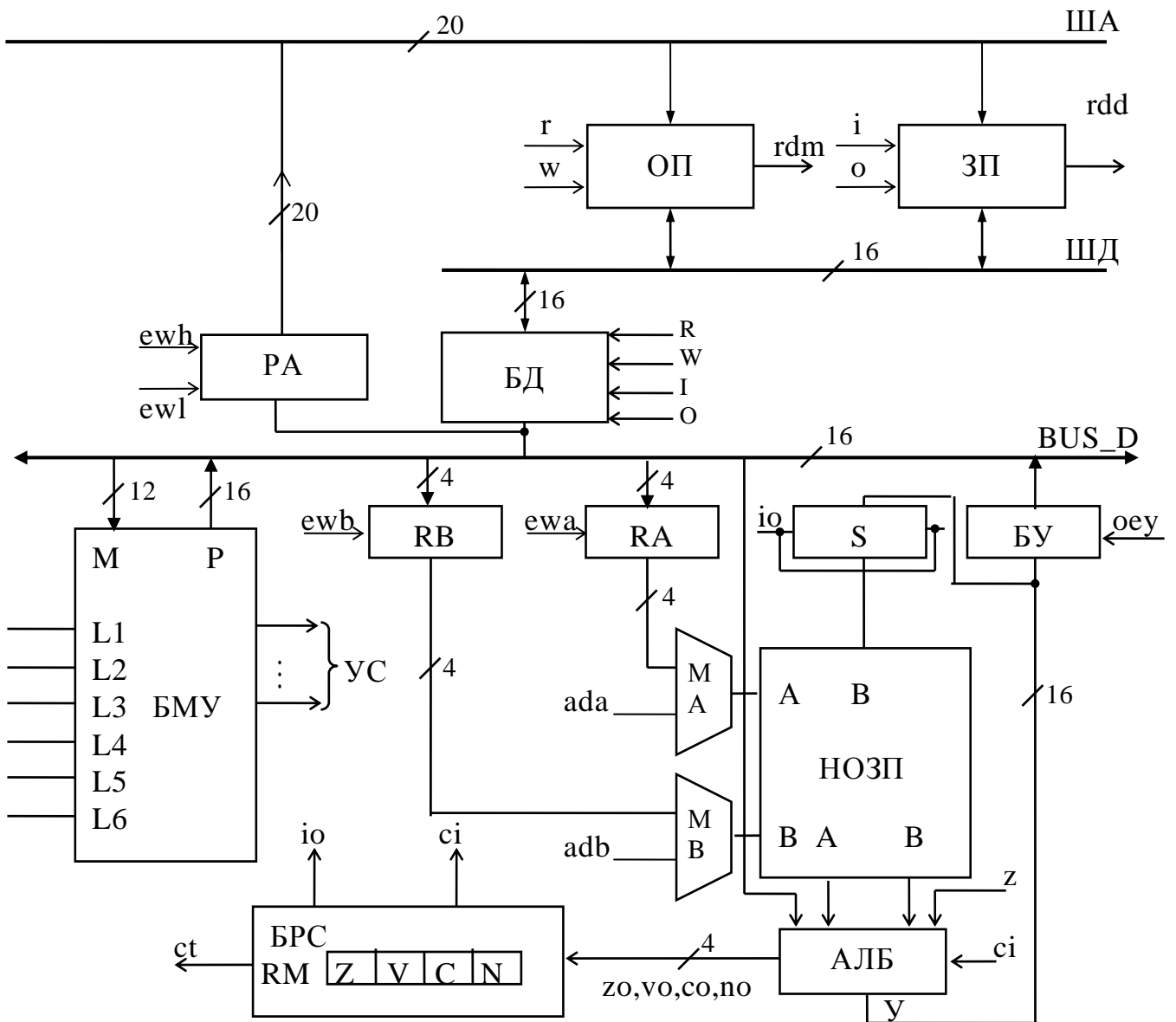
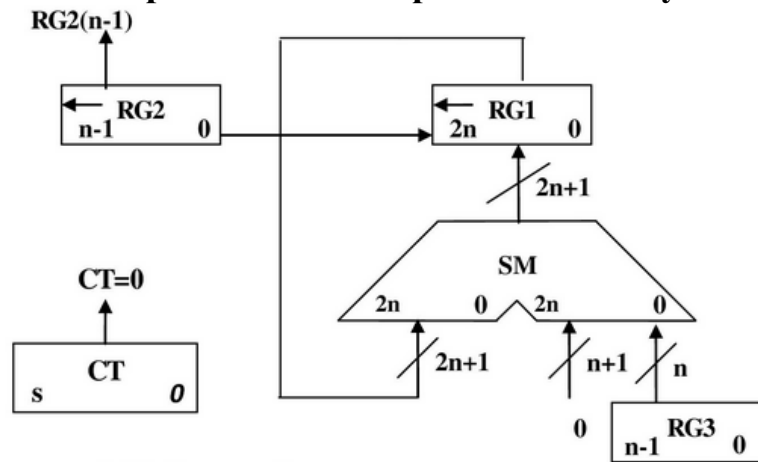
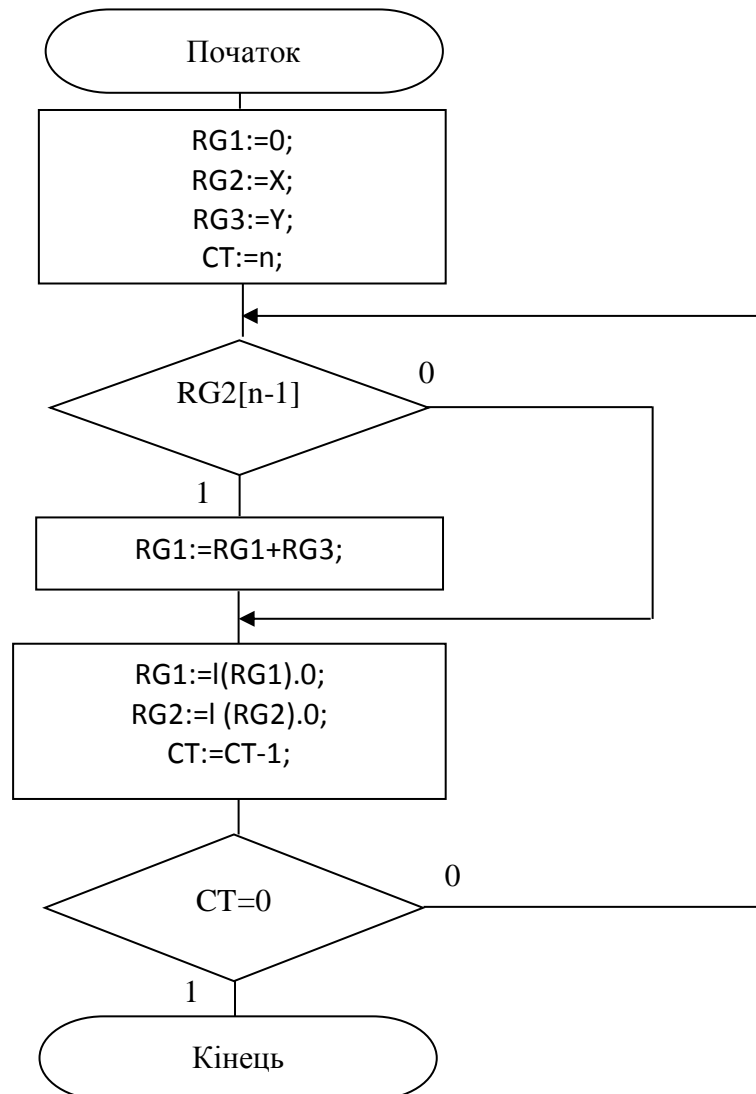


Рисунок 4.1 Структурна схема ЕОМ

Операційна схема третього способу:



Змістовний мікроалгоритм третього способу:



```

link l1:ct
link l2:rdm
link ewh:l6
link M:z,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, 3lach
accept r7: 20h \reg of current adr

dw 06h:000eh \x
dw 08h:0007h \y
dw 0Ah:0013h \z

dw 20h: 00010000000000110% \ save x to r15
dw 22h: 00100000000001000% \ r15 & y
dw 24h: 01000000000001010% \ mult
dw 26h: 00110000000001010% \ result to z
dw 28h: 0101000000000000% \ END

org 0h {cjp nz, loadFromOP;}
org 2h {cjp nz, saveToReg;}
org 4h {cjp nz, andXY;}
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 r6

{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

andXY
{cjs nz, fromMem;}
{and r15, r15, r14;} \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;}

start
{and nil, r14, 8000h; load rm, flags;} \mnozheny[1]=?
{cjp rm_z, shift;}
{add r0, r0, r15, z;}

shift
{or sll, r14, r14, z;} \z=r14
{or sll, r0, r0, z;} \res=r0

{or nil, r14, r14; load rm, flags;}
{cjp not rm_z, start;} \if z!=0

{or srl, r0, r0, z;} \usuvaemo zsuv
{or r15, 0, r0;} \result
{cjp nz, next;}

\{and nil, r1, 8000h; load rm, flags;}
\{cjp rm_z, end;}
\{xor r1, r1, 0ffffh;}
\{xor r0, r0, 0ffffh;}
\{add r0, r0, 1, z; load rm, flags;} proverka znaka i perevod v PK
\{add r1, r1, 0, rm_c;}
\{or r1, r1, 8000h;}

end {}

```

5 Висновки

В процесі розробки був розроблений інструментарій для забезпечення необхідних перетворень інформації, які необхідні для обчислень за даним алгебраїчним виразом з введенням та виведенням інформації на зовнішні пристрої. На мікропрограмному рівні була розроблена система команд та реалізований алгоритм організації процесу обчислень.