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*»

Виконала:

студентка групи ІВ-71

Молчанова В.С.

Перевірив:

доц. Верба О. А.

Київ - 2019 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

**1 Початкові умови згідно з варіантом**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер залікової книжки: 1110011110010 | | | | | | |
| a7 | a6 | a5 | a4 | a3 | a2 | a1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Адреса РС | | | Адресація | Операція | | |
| Порт вводу | | |
| 72h | | | (X & Y) \* Z | | |
| Адреса РД | | | Спосіб множення | | |
| Пряма |
| 74h | | | 2ий | | |

**2 Алгоритм обчислення**



**3 Програма обчислень**

Для реалізації програми обчислень потрібні такі команди

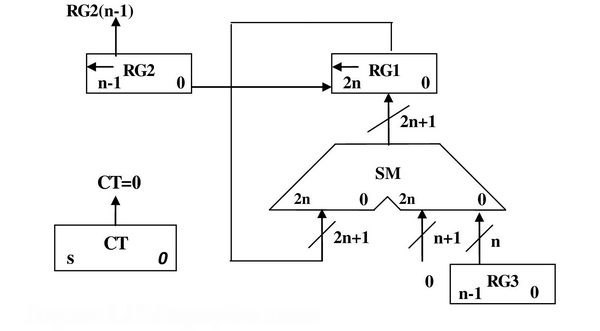
|  |  |  |
| --- | --- | --- |
| Команда | Код | Операція |
| mov r8, [mem] | 0 0 0 0 | Зчитування команди з ЗП |
| mov r15, x | 0 0 1 0 | Запис операнда у регістр |
| and r15, y | 0 1 0 0 | Логічне множення операндів |
| mov [mem], r15 | 0 1 1 0 | Запис операнду у ЗП |
| mul r14, r15 | 1 0 0 0 | Множення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

RG1:=0;

RG2:=X;

RG3:=Y;

CT:=n;

RG2[n-1]

RG1:=RG1+RG3;

RG1:=l(RG1).0;

RG2:=l (RG2).0;

CT:=CT-1;

CT=0

Початок

Кінець

1

1

0

0

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 r3: 0 \sign of x&y

accept r4: 0 \sign of z

accept r7: 20h \reg of current adr

dw 06h: 0000000000001110% \x

dw 08h: 0000000000000111% \y

dw 0Ah: 1100000000010011% \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, 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 r8

{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

\r14=z, r15=x&y

mult \multiplication

{cjs nz, fromMem;}

{xor r0, r0, r0;}

checkr14

{ and nil, r14, 8000h; load rm, flags; }

{ cjp not rm\_z, save\_r14; }

checkr15

{ and nil, r15, 8000h; load rm, flags; }

{ cjp not rm\_z, save\_r15; }

start

{and nil, r14, 8000h; load rm, flags;} \mnozhnyk[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

\signs from lab3

{ xor r3, r3, r4; }

{ add r15, r15, r3, z; }

{ cjp rm\_z, next; }

save\_r14

{ sub r14, r14, 49152, nz; }

{ add r3, r3, 49152, z; }

{ cjp nz, checkr15; }

save\_r15

{ sub r15, r15, 49152, nz; }

{ add r4, r4, 49152, z; }

{ cjp nz, start; }

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.