Міністерство освіти та науки України

Національний технічний університет України

**«Київський політехнічний інститут»**

Факультет прикладної математики

Кафедра спеціалізованих комп’ютерних систем

##### РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

з дисципліни

«Архітектура комп’ютера»

**Виконав:**

студент групи: КВ-92

Степанюк М.Ф.

**Київ 2012**

***Завдання***

Розробити програму обчислення функції:

,

де  - операція, що задана в табл.1. Результат цієї операції умножається на . Логічні операції є порозрядними.

Операнди  є 16-розрядними додатними цілими числами (старший знаковий розряд дорівнює нулю). Операнди вводяться з пристрою введення (ПВв). Послідовність введення визначається самостійно. Результат (в 32-розрядному форматі) виводиться в пристрій виведення (ПВив), а також записується в дві сусідні комірки пам’яті . Адреси регістрів стану (РС) і даних (РД) зовнішніх пристроїв подані в табл. 2, адреси пам’яті задати самостійно. Послідовність даних, що вводяться та виводяться, визначається розробником. Система команд містить лише одноадресні команди. Структура одно адресної команди подана на рис.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Ф | Операція | ТА | Адреса в ОП |
| 15 | 14……...11 | 10 | 9……………..0 |

***Рис. 1.1.***

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.1.2.



***Рис. 1.2. Формат регістру стану зовнішнього пристрою***

**Початкові умови згідно з варіантом**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| а7 | а6 | | а5 | а4 | а3 | а2 | а1 | |
| 1 | 1 | | 0 | 0 | 0 | 1 | 1 | |
| Адреси РС | | | | Тип адресації | Операція | | | |
| ПВв | | ПВив | | Пряма | Спосіб множення | | | - |
| 52H | | D2H | |  | 2-й | | | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Структура НОЗП | | | Структура ПМК | |
| R0 |  | РЗП (регістри загального призначення) | 0 | Вектори команд |
| … | … |
| R6 | 15H |
| R7 | ЛК | Робочі регістри | … |  |
| R8 | РК | 20H | Мікропрограма |
| R9 | ВС | … |
| R10 | Допоміжні | 66H |
| R11 |
| R12 |
| R13 | Операнд |
| R14 | адреса операнда |
| R15 | Аккумулятор |
| RQ |  |

***Архітектура та алгоритм обчислення***

***Рисунок 2.1. Модель програміста***

|  |  |  |  |
| --- | --- | --- | --- |
| Структура ОП | | Структура ЗП | |
| 00000H |  | 0000H |  |
| … | … |  |
| 00020H | Програма обчислення | 0012H | РС (пристрій вводу) |
| … | 0013H |
| 0004BH | 0014H | РД (пристрій вводу) |
| … |  | 0015H |
| 00050H | Дані | … |  |
| … | 0092H | РС (пристрій виводу) |
| 00069H | 0093H |
| … |  | 0094H | РД (пристрій виводу) |
| FFFFFH | 0095H |
|  | | … |  |
| FFFFH |

***Рис 2.2. Розподіл загальної пам’яті***



***Рисунок 2.3. Алгоритм обчислень***

***Програма обчислень***

Для реалізації програми обчислень потрібні такі команди:

|  |  |  |
| --- | --- | --- |
| Команда | Код | Операція |
| in ax,[mem] | 0 0 0 1 | Зчитування з ЗП |
| out [mem],ax | 0 0 1 0 | Запис у ЗП |
| mov [mem],ax | 0 0 1 1 | Пересилання з пам’яті до акумулятора |
| mov ax,[mem] | 0 1 0 0 | Пересилання з акмулятора до пам’яті |
| Swap ax,reg | 0 1 0 1 | Обмін регістрів |
| exit | 0 1 1 0 | Вихід |
| or [mem],ax | 0 1 1 1 | Діз’юнкція |
| test ax,[mem] | 1 0 0 0 | Перевірка готовності ЗП |
| jz [mem] | 1 0 0 1 | Умовний перехід |
| mul [mem],ax | 1 0 1 0 | Множення |

***Таблиця 3.1. Програма обчислень***

***Структура комп’ютера та мікропрограма обчислень***



***Рисунок 4.1. Структурна схема ЕОМ***

***Програмний код***

\Initialization

LINK l1:ct

LINK l2:rdm

LINK l3:rdd

LINK ewh:16

LINK M:z,z,z,z,z,z,z,14,13,12,11,z

ACCEPT dev[1]:I,052h,054h,12,2

ACCEPT dev[2]:O,0D2h,0D4h,12,2

\ My values

ACCEPT dev\_buf[1]:0004h,0007h,0003h

ACCEPT R7:0020h

ACCEPT R10:03ffh

DW 0020h:0001110001010110% \test [0056h]

DW 0022h:0001010001011010% \jz [005Ah]

DW 0024h:0010110001010110% \in r15,[0056h]

DW 0026h:0011110001100100% \mov [0064h],r15

DW 0028h:0001110001010110% \test [0056h]

DW 002Ah:0001010001011100% \jz [005Ch]

DW 002Ch:0010110001010110% \in r15,[0056h]

DW 002Eh:0011110001100110% \mov [0066h],r15

DW 0030h:0001110001010110% \test [0056h]

DW 0032h:0001010001011110% \jz [005Eh]

DW 0034h:0010110001010110% \in r15,[0056h]

DW 0036h:0011110001110110% \mov [0068h],r15

DW 0038h:0100010001100110% \or r15,[0066h]

DW 003Ah:0100110001100100% \mul [0064h]

DW 003Ch:0001110001011000% \test [0058h]

DW 003Eh:0001010001100000% \jz [0060h]

DW 0040h:0010010001011000% \out [0058h]

DW 0042h:0000110000000000% \swap r15,r13

DW 0044h:0001110001011000% \test [0058h]

DW 0046h:0001010001100010% \jz [0062h]

DW 0048h:0010010001011000% \out [0058h]

DW 004Ah:0101010000000000% \exit

\Data

DW 0056h:006Ah \(IN)

DW 0058h:006Ch \(OUT)

DW 005Ah:0020h \Return adresses

DW 005Ch:0028h

DW 005Eh:0030h

DW 0060h:003Eh

DW 0062h:0046h

DW 0064h:006Eh \(X)

DW 0066h:0070h \(Y)

DW 0068h:0052h \(Z)

DW 006Ah:0052h \IN

DW 006Ch:00D2h \OUT

DW 006Eh:0000h \X

DW 0070h:0000h \Y

DW 0052h:0000h \Z

\Microcommands vector init

ORG 0h \Program start vector

{cjp nz,go;}

ORG 2h \swap r15,r13

{cjp nz,swap;}

ORG 4h \jz [mem] vector

{cjp nz,zjump;}

ORG 6h \test [mem] vector

{cjp nz,testdevice;}

ORG 8h \out [mem],r15 vector

{cjp nz,output;}

ORG 0Ah \in r15,[mem] vector

{cjp nz,input;}

ORG 0Ch \mov r15,[mem] vector

{cjp nz,movToReg;}

ORG 0Eh \mov [mem],r15 vector

{cjp nz,movToMem;}

ORG 10h \orr r15,[mem] vector

{cjp nz,orr;}

ORG 12h \mul vector

{cjp nz,mul;}

ORG 14h \exit vector

{cjp nz,exit;}

ORG 20h

go

\From OP command loading

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

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

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

\Command format analysis

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

{cjp not rm\_z,exit;}

\Adressation type analysis

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

{cjp rm\_z,exit;}

\Operand adress selection

{and r14,r8,03ffh;}

\Microcommands selector

{or nil,r8,z;oey;JMAP;}

\From Memory loading microprogram

FromMem

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

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

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

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

{crtn nz;}

\Swap r15,r13

swap

{or r11,r15,z;}

{or r15,r13,z;}

{or r13,r11,z;}

{cjp nz,next;}

mul

{cjs nz,FromMem;}

{or r11,z,17;}

{xor nil,r14,r15; load rm,flags;}

{or nil,r14,r14; cjp not no ,ll1;}

{sub r14, 0,r14,nz;}

ll1 {or nil,r15,r15; cjp not no ,ll2;}

{xor r15,r15,8000h;}

ll2 {cjp not rm\_c,ll3;}

{add r13,r13,r14,z;}

ll3 {or srl,r13,r13,z;}

{or sr.9,r15,r15,z;}

{sub r11,r11,z,z; load rm, flags;cem\_c;cem\_n;}

{cjp not rm\_z, ll2;}

{cjp not rm\_n,fin;}

{xor r15,r15,0FFFFh;}

{xor r13,r13,0FFFFh;}

{add r15,r15,1,z;}

{add r13,r13,0,rm\_c;}

fin {cjp nz,next;}

\mov r15,[mem]

movToReg

{cjs nz,FromMem;}

{or r15,r14,z;}

{cjp nz,next;}

\mov [mem],15

movToMem

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

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

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

{w;cjp rdm,cp;or nil,r15,z;oey;}

{cjp nz,next;}

\and r15,[mem]

orr

{cjs nz,FromMem;}

{and r15,r15,r14;}

{cjp nz,next;}

\jz [mem]

zjump

{cjp not rn\_z,no\_j;}

{cjs nz,FromMem;}

{sub r7,r14,1;}

no\_j {cjp nz,next;}

\in r15,[mem]

input

{cjs nz,FromMem;}

{add r14,r14,2,z;}

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

{i;cjp rdd,cp;or r15,bus\_d,z;}

{cjp nz,next;}

\out [mem],r15

output

{cjs nz,FromMem;}

{add r14,r14,2,z;}

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

{o;cjp rdd,cp;or nil,r15,z;oey;}

{cjp nz,next;}

\test [mem]

testdevice

{cjs nz,FromMem;}

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

{i;cjp rdd,cp;or r14,bus\_d,z;}

{or nil,r14,z;load rn,flags;}

{cjp nz,next;}

Next

{add r7,r7,2,z;}

{cjp nz,go;}

exit {}

***Висновок***

В процесі виконання роботи була розроблена система команд для обчислення заданої функції з введенням вхідних даних та виведенням результату через зовнішні пристрої. Був розроблений інструментарій для забезпечення необхідних перетворень інформації, які необхідні для обчислень за даним алгебраїчним виразом. Робота базована на принципі мікропрограмного керування, алгоритм обчислення арифметичної функції був розбитий на окремі мікропідпрограми (система команд високого рівня), складені з команд системи команд низького рівня (мікрокоманди Міколасемблеру), яка зберігається в мікропрограмній пам’яті.