# *Національний технічний університет України*

# *«Київський політехнічний інститут»*

#### ***Факультет інформатики та обчислювальної техніки***

## Лабораторна робота №7

*з курсу "****Комп’ютерна архітектура****"*

***Виконав:***

*Бедь А.М.*

***Група*** *ІО-12,*

***Номер варіанту*** *1202*

***Київ - 2013р.***

**ВИКОНАННЯ КОМАНД В ЕОМ**

Ціль роботи – вивчити етапи виконання команд введення-виведення. Навчитися розробляти мікроалгоритми і мікропрограми реалізації кожного етапу зазначених команд. Вивчити способи взаємодії процесора з зовнішніми пристроями в програмному режимі опитування готовності пристроїв. Одержати навички розробки мікропрограм з використанням мнемонічного мікроассемблера.

**Завдання:**

1. Доробити мікропрограму, отриману при виконанні лабораторних робіт 4, 5 і 6, включивши до складу команд одноадресні команди введення і виведення. Формат команд зазначений на рис. 1. Код операції команди введення а5 а4 а2 а1 + 3 , а виведення - а5 а4 а2 а1 + 4.

2. Розробити програму в кодах команд для передачі двох слів із пристрою введення в пристрій виведення (адреси регістрів в табл. 1). Перед звертанням до РД зовнішнього пристрою варто перевіряти готовність пристрою до обміну. Для цього необхідно прочитати РС пристрою і перевірити біт готовності.

Варіант 120210 = 10010110010

|  |  |  |  |
| --- | --- | --- | --- |
| а3 | а2 | а1 | Адреса РС |
| ПВв | ПВив |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 22H | A2H |

6800h - Input

7000h - Output

**Лістинг коду:**

\Налагодження зв'язків

link ewh: 16

link l1: ct

link l2: rdm

link l3: rdd

link rb: z, 2, 1, 0

link m: z, 14, 13, 12, 11, 10, z, z, z, z, z, z

accept dev[2]:o, 00A2h, 00A4h, 3, 5

accept dev[1]:i, 0022h, 0024h, 3, 5

accept dev\_buf[1]: 7777h, 0DDDDh

accept rdm\_delay: 2

equ COD:0800h

equ TYPE\_ADDR:0210h

equ COD1:5800h \Умовний перехід

equ COD2:6000h \Безумовний перехід

equ COD3:6800h \Input

equ COD4:7000h \Output

\Завдання значень комірок пам'яті

dw 0111h:6800h

dw 0112h:7000h

dw 0113h:6800h

dw 0114h:7000h

\Адреса мікрокоманди

accept r6:0100h

accept r5:0110h

accept r7:0111h

accept r15:0001h

start

\Область програми

{xor nil, r11, r11; oey; ewh;}

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

\Завантеження мікрокоманди з ОП у регістр R8

load\_k

{cjp rdm, load\_k; r; or r8, bus\_d, z; load rm, flags;}

{cjp rm\_z, stop;}

\Перевірка формату команди

{cjp rm\_n, double\_a;}

\Перевірка операції

{or r12, r12, 7800h;}

{and nil, r8, r12; oey; jmap;}

org 0500h

\Перевірка адресації

{and nil, r8, 0400h; load rm, flags;}

{cjp not rm\_z, indirect;}

\Завантеження адреси операнда на шину даних

{xor nil, r11, r11; oey; ewh;}

\{and r11, r11, 03ffh;}

{or r11, 03ffh, z; }

{and nil, r8, r11; oey; ewl;}

\Завантаження операнда з ОП в R10

load\_o

{cjp rdm, load\_o; r; or r10, bus\_d, z;load rm, flags;}

\Множення 2-им способом

\Якщо операнд в r10 рівний нулю

\обнуляємо r15 і пропускаємо множення

{cjp not rm\_z, main\_mull;}

{xor r15, r15, r15;}

{cjp nz, end\_cmd;}

main\_mull

\Підготовка даних в регістрах

\r11,r10 - M

\r12 - X

{xor r11, r11, r11;}

{xor r12, r12, r12;}

{or r12, r12, r15;load rm, flags;}

{cjp rm\_z, end\_cmd;}

{xor r15, r15, r15;}

\Перевірка знаку для X

{or r9, r9, 8000h;}

{and r9, r9, r12; load rm, flags;}

{and r12, r12, 7fffh;}

\Перевірка знаку для M

{or r13, r13, 8000h;}

{and r13, r13, r10; load rm, flags;}

{cjp rm\_z, lable1;}

{sub r10, z, r10, nz;}

\Цикл множення

lable1

{or r4, r4, 0001h;}

{and r4, r4, r12;load rm, flags;}

{cjp rm\_z, lable2;}

{add r15, r15, r10, z; load rm, flags;}

{add r14, r14, r11, rm\_c;}

lable2

{or srl, r12, r12, z;}

{or sll, r10, r10, z;}

{or sl.25, r11, r11, z;}

{or r12, r12, z; load rm, flags;}

{cjp not rm\_z, lable1;}

\Встановлення знаку результату

{xor r9, r9, r13;}

{or r14, r14, r9;}

{cjp nz, end\_cmd;}

\Непряма адресація

indirect

{cjp nz, end\_cmd;}

\Не моя операція множення

org 0580h

\Реалізація умовного переходу

cond

\Перевірка типу адресації

{xor r12, r12, r12;}

{and r12, r8, 0400h; load rm, flags;}

{cjp not rm\_z, idcon;}

\Пряма адресація

{or r12, r15, z;}

{and r12, r12, 0080h; load rm, flags;}

{cjp rm\_z, end\_cmd;}

{or r12, 03ffh, z; }

{and nil, r8, r12; oey; ewl;}

laddrc

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

{xor r7, r7, r7;}

{or r7, r14, z;}

{cjp nz, naddr;}

\Непряма адресація

idcon

{cjp nz, end\_cmd;}

{cjp nz, naddr;}

org 0600h

\Реалізація безумовного переходу

uncond

\Перевірка типу адресації

{xor r12, r12, r12;}

{and r12, r8, 0400h; load rm, flags;}

{cjp not rm\_z, iducon;}

\Пряма адресація

{or r12, 03ffh, z; }

{and nil, r8, r12; oey; ewl;}

laddrunc

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

{xor r7, r7, r7;}

{or r7, r14, z;}

{cjp nz, naddr;}

\Непряма адресація

iducon

{cjp nz, end\_cmd;}

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\Input

org 0680h

{or r11, 0022h, z;}

{xor nil, r11, r11; oey; ewl;}

{or nil, r11, z; oey; ewl;}

read\_rsi

{cjp rdd, read\_rsi; i; or r11, bus\_d, z;}

\Перевірка біту готовності

{and r11, r11, 0080h; load rm, flags;}

{cjp rm\_z,read\_rsi;}

\Зчитування даних з РД

{or r11, 0024h, z;}

{xor nil, r11, r11; oey; ewl;}

{or nil, r11, z; oey; ewl;}

read\_rdi

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

{cjp nz, end\_cmd;}

\Output

org 0700h

{or r11, 00A2h, z;}

{xor nil, r11, r11; oey; ewl;}

{or nil, r11, z; oey; ewl;}

{xor r11, r11, r11;}

write\_rsi

{cjp rdd, write\_rsi; o; or r11, bus\_d, z;}

\Перевірка біту готовності

{and r11, r11, 0080h; load rm, flags;}

{cjp rm\_z, write\_rsi;}

\Запис даних в РД

{or r11, 00A4h, z;}

{xor nil, r11, r11; oey; ewl;}

{or nil, r11, z; oey; ewl;}

write\_rdi

{cjp rdd, write\_rdi; o; or nil, r15, z; oey;}

{cjp nz, end\_cmd;}

\\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\Двоадресна команда

double\_a

\Перевірка операції

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

org 0080h

\Перевірка типу адресації

{xor r12, r12, r12;}

{and r12, r8, 0318h;}

{xor nil, r12, TYPE\_ADDR; load rm, flags;}

{cjp not rm\_z, nAutoInc;}

\Отримуємо РЗП1 і РЗП2 в rb

{or nil, r8, z; oey; load rb;}

\Зчитуємо перший операнд з ОП

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

{or r3, r3, 0002h;}

{add rb, rb, r3;}

load\_op1

{cjp rdm, load\_op1; r; or r0, bus\_d, z;}

\Зчитуємо другий операнд з ОП зразу робимо оп. АБО і поміщаємо рез. в R14

{or srl, r10, r8, z;}

{or srl, r10, r10, z;}

{or srl, r10, r10, z;}

{or srl, r10, r10, z;}

{or srl, r10, r10, z;}

{or nil, r10, z; oey; load rb;}

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

load\_op2

{cjp rdm, load\_op2; r; or r14, bus\_d, r0;}

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

\результат в ОП

write\_rez

{cjp rdm, write\_rez; w; or nil, r14, z; oey;}

{add rb, rb, r3;}

{}

{cjp nz, end\_cmd;}

nAutoInc

{}

{cjp nz, end\_cmd;}

end\_cmd

\Формування адреси наступної команди

{add r7, r7, z, nz;}

naddr

{cjp nz, start;}

stop

{}