

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**Архітектура комп'ютерів**

**Лабораторна робота №4**

Виконав: студент групи ІО-92  
Зубко Павло

.

2011 р.

**Мета:** вивчити етапи виконання команд в ЕОМ. Навчитися розробляти мікроалгоритми вибірки, розпакування команд, виконання операцій і формування адреси наступної команди. Одержати навички розробки мікропрограм для ЕОМ з використанням мнемонічного мікроасемблера.

**Варіант №9209=1111111111111001**

Тип адресації – пряма

Код операції множення – 1101

link l1: ct

link l2: rdm

link ewh:16

link m:z,z,15,14,13,12,11,z,z,z,z

dw 0:6830h

dw 2:6830h

DW 4:6830h

dw 6:6830h

dw 8:6830h

dw 10:7fffh

dw 30h:4h

dw 31h:7h

accept r15:4h

accept r7:0

accept r6:10h

NEXT {ADD NIL, R7, Z; OEY; EWL;} \OEY видаёт r7 на шину вывода, EWL заносит данные из шины вывода в младшие разряды шины адреса

{R; CJP RDM, CP; ADD R8, BUS\_D, Z;} \R:читаем из памяти по адресу, который указан на шине адреса

{ADD NIL, R8, Z; CJP NO, end;}

{XOR R9, R9, R9;}

{ADD R9, R9, 0400h;}

{AND NIL, R9, R8; CJP not ZO, end;}

{XOR R9, R9, R9;} \очищаем R9

{ADD R9, R9, 03FFh;} \ставим маску на последние 10 бит (биты адреса)

{AND NIL, R9, R8; OEY; EWL;}

{R; CJP RDM, CP; ADD R13, BUS\_D, Z;}

{add r7, r7,1;}

```

{ADD NIL, R8, Z; OEY; JMAP;}

RETURN {ADD R7, R7, Z, NZ; CJP NZ, NEXT;}

org 1a0h

{load rm, z;}

{XOR R10, R10, R10;}

{XOR R11, R11, R11;}

{XOR R12, R12, R12;}

{xor r6,r6,r6;}

{add r6,r6,10h;}

{add r11,r11,r15;}

{or r13, r13, r13;load rm, flags; cjp not rm_n, chek;}

{add r12, r12, 1, z;}

{and r13, r13, 7fffh;}

chek{or r15, r15, r15;load rm, flags; cjp not rm_n, start;}

{xor r12, r12, 1;}

{and r15, r15, 7fffh;}

{and r11, r11, 7fffh;}

start {and nil, r13, 8000h; load rm, flags;}

{cjp rm_z, shift;}

{add r15, r15, r11, z; load rm, flags;}

{add r14, r14,z, rm_c;}

shift{or sl, r13,r13,z;}

{sub r6, r6, z, z; load rm, flags; cjp rm_z, setSign;}

{or sl.16, r15, r15, z;}

{or sl.25, r14, r14, z;cjp nz, start;}

setSign {or nil, r12, r12; load rm, flags; cjp rm_z, return;}

{xor r14, r14, 0ffffh;}

{xor r15, r15, 0ffffh;}

{add r15, r15, 1, z;}

{CJP NZ, RETURN;}

end {}

```