НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНИХ КОМП’ЮТЕРНИХ СИСТЕМ

**Лабораторна робота №3  
з дисципліни «Комп’ютерна схемотехніка»**

Виконав  
студент 3-го курсу  
групи КВ-41  
Горпинич-Радуженко Іван

Київ – 2017

**Постановка задачі**

Розробити мікроалгоритм перетворення числа з формату INT у формат REAL. Виконати операцію для кількох чисел.

**Код мікропрограми**

macro mov reg1, reg2: { or reg1, z, reg2; }

accept rq: 0fe1fh \ argument

accept r0: 0 \ result

accept r1: 0 \ argument in direct code

accept r2: 0 \ order

accept r3: 0 \ buffer

accept r4: 0 \ count

link l1: ct

{ or nil, rq, 0; load rn, flags; }

{ cjp rn\_z, final; } \ if argument = 0, go to end

{ cjs nz, dircode; } \ transforms argument into direct code and writes it to r1

{ cjs nz, ressign; } \ writes result's sign to r0[15]

{ cjs nz, order; } \ writes result's order to r2 and r0[13-8]

{ cjs nz, mantissa; } \ writes mantissa to r0[7-0]

{ cjp nz, end; }

org 100h

dircode {}

{ or nil, rq, 0; load rn, flags; }

{ cjp not rn\_n, positive; }

{ mov r1, 8000h; } \ r1 := 1000 0000 0000 0000

{ mov r3, rq; } \ r3 := argument

{ xor r3, 0ffffh; } \ r3 := not r3

{ add r3, 1; } \ r3 := r3 + 1

{ or r1, r3; } \ r1 := - r3

{ crtn nz; }

positive {}

{ mov r1, rq; }

{ crtn nz; }

org 140

ressign {}

{ mov r0, r1; }

{ and r0, 8000h; } \ r0[15] := result's sign

{ crtn nz; }

org 150h

order {}

{ mov r3, r1; }

{ and r3, 7fffh; } \ r3 := (r1 < 0) ? -r1 : r1

{ mov r4, 10h; }

{ push; } \ sla buffer, until 1 in senior level detected

{ sub r4, 0; }

{ loop no; or sla, r3, 0; }

{ add r2, r4, nz; } \ r2 := result's order

{ mov r3, r2; }

{ push nz, 7; }

{ or sla, r3, 0; } \ buffer := sla order (8 times)

{ rfct; }

{ or r0, r3; } \ r0[13-8] := buffer

{ crtn nz; }

org 200h

mantissa {}

{ mov r3, r1; }

{ and r3, 7fffh; } \ r3 := (r1 < 0) ? -r1 : r1

{ mov r4, r2; }

{ and nil, r4, 0fff8h; load rn, flags; }

{ cjp rn\_z, onebyte; }

{ sub r4, 8, nz; load rn, flags; }

{ cjp rn\_z, eightlev; }

{ push; }

{ or srl, r3, 0; } \ if order > 8, srl buffer (order-8) times

{ loop nz; sub r4, 0; }

{ cjp nz, final; }

eightlev {}

{ and r3, 0ffh; } \ if order = 8, r3 := 0000 0000 [mantissa]

onebyte {}

{ sub r4, 8, r4, nz; }

{ push; }

{ or sll, r3, 0; } \ if order < 8, sll buffer (8-order) times

{ loop zo; sub r4, 0; }

final {}

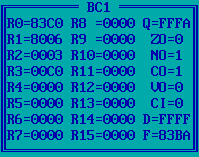
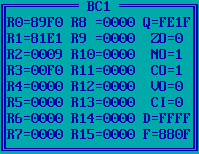
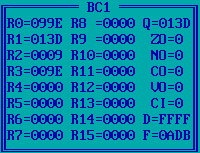
{ or r0, r3; } \ r0[7-0] := buffer

{ crtn nz; }

end {}

**Перевірка**

Після виконання мікропрограми в регістрі R0 зберігається результат у форматі з плаваючою комою, в регістрі R1 – результат у прямому коді, в регістрі R2 – порядок мантиси. Початкове значення задається в регістрі Q.

**