**3. Код програми:**

link l1: ct

link l2: rdm

link ewh: 16

dw 0h: 0fff9h \ X

dw 1h: 0005h \Y

dw 2h: 0

dw 3h: 0

dw 4h: 0

accept r7:0 \ pochatkiviy adres danyh v OP

accept r8:10h\kinceviy adres danyh v OP

accept r11:0

accept rdm\_delay:3

{xor nil, r0, r0; oey; ewh;} \ obnulyaem starshie RAD

{or nil, r7, z; oey; ewl;} \ pishem iz r7 v mladshie RAD

{cjp rdm, cp; r; or r0, bus\_d, z;} \ pishem v r7 iz OP

{add r7, r7, 1, z;}

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

{cjp rdm, cp; r; or r1, bus\_d, z;}

{add r7, r7, 1, z;}

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

{cjp rdm, cp; r; or r2, bus\_d, z;}

{add r7, r7, 1, z;}

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

{cjp rdm, cp; r; or r3, bus\_d, z;}

{add r7, r7, 1, z;}

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

{cjp rdm, cp; r; or r4, bus\_d, z;}

{LOAD RM, Z;} \Obnulujemo registr flagiv RM

{AND NIL, R0, 8000h; LOAD RM, FLAGS;}

\Perevirka registru na najyvnist ostannogo bitu, rezul-t u RM

{CJP RM\_Z, NOT2;} \Strubajemo na NOT2, yakshcho nemaje bitu u RM\_Z

{ADD R11, R11, 1, Z;} \(Flag, yakshco choslo vidjemne)

{XOR R0, R0, 0FFFFh;}\Peretvorennya u dodatne chislo

{ADD R0, R0, 1, Z;}

NOT2 {AND NIL, R1, 8000h; LOAD RM, FLAGS;} \Te same z inshim registrom

{CJP RM\_Z, label1;}

{AND R1, R1, 07FFFh;}

{SUB R11, R11, Z, Z;} \(Takozh flag vid'jemnosti, yakshso (-)\*(-), otrimujemo (+)

label1{OR SRL, r0, z;}

{OR SR.9, r4, r4, z;}

{OR SRL, r0 , z;}

{OR SR.9, r4, r4, z;}

{LOAD RM, NZ; CEM\_Z; CEM\_v; cem\_n;}

{OR SL.25, r1, z;}

cicle {OR SLL, r1,z;}

{CJP NOT RM\_C, noadd;}

{ADD r2, r2, r4, z; LOAD RM, FLAGS;}

{ADD r3, r3, r0, RM\_C;}

noadd{OR SRL, r0, z;}

{OR SR.9, r4, z;}

{OR SLA, r9,r1, z;}

{OR NIL, R9, R9; CJP NOT ZO, cicle;} \Jakshco mnojnik ne 0 - stribajemo dali

{AND R11, R11, R11; CJP ZO, goon;} \Perevirka, chi <0

{ADD R3, R3, 8000h, Z;} \jakshco tak - zapis 1 u pershij bit

goon {or nil, r8,z; oey; ewl;} \zapis v RAD

ll3 {cjp rdm, ll3; w; or nil, r3, z; oey;} \zapis

{ADD r8, r8, 1, z;}

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

ll4 {cjp rdm, ll4; w; or nil, r2, z; oey;} \ zapis

end {}

1. **4. Висновки:**
2. *У даній роботі побудований алгоритм множення двох 16-розрядних операндів четвертим способом.*

*Початкові значення регістрів завантажуються з памяті. (регістр R0 відповідає X, R1 відповідає Y). Отримані результати моделювання співпадають з прогнозованими. В результаті виконання цієї роботи, я пригадав й закріпив теоретичні аспекти цієї теми.*