**Лабораторная работа 4**

**Программы с использованием ветвлений и циклов на языке ассемблера. Утилиты Absolute Lister и Hex Conversion Utility**

**Цель работы:** изучить команды группы программного управления и команды повторения. С использованием этих команд научиться реализовывать ветвления и циклы в ассемблерных программах. Ознакомиться с утилитами Absolute Lister и Hex Conversion Utility.

В данной лабораторной работе была создана программа, подсчитывающее количество отрицательных элементов в массиве, состоящем из n целых чисел. Для этого написал следующий код:

\_myvect .sect"myvect";Созданем секцию для векторов прерываний

reset .word \_c\_int00 ;RESET вектор

.space 1;Резервируем место для NMI прерывания

;Создаем таблицу векторпов программных прерываний

\_mytrap .sect"mytrap";Секция для векторов программных прерываний

;Создаем стек

\_mystack .usect"mystack",100;Резервируем 100 слова под стек

.text

stacka .word \_mystack;адрес mystrack секции

ivta .word \_myvect;адрес myvect секции

tvta .word \_mytrap;адрес mytrap секции

ieval .word 1;значения регистра разрешения прерываний

gctrl .word 1EF78000h;значение регистра управления памятью

lctrl .word 1EF78000h;значение регистра управления памятью

mctrla .word 100000h;адрес регистра управления глобольной памятью

\_c\_int00: ;точка входа в программу, при reset начинаем отсюда

; Инициализация регистра DP

ldp stacka;

;Устанавливаем указатель на таблицу прерываний

ldi@ivta, AR0

ldpe AR0, IVTP;

;Устанавливаем укзатель на таблицу программных прерываний

ldi@tvta, AR0

ldpe AR0,TVTP

;Иницилизирум трегистр управления глобольной паматью

ldi@mctrla, AR0

ldi @gctrl,R0

sti R0, \*AR0

;Инициализируем регистр управления локальной памятью

ldi@lctrl, R0

sti R0, \*+AR0(4)

;Инициализируем указатель стека

ldi@stacka, SP

;Инициализируем регистр разрешения прерываний

;Запись 1 в IIE разрешает прерывание таймера

ldi @ieval,IIE

;Глобально разрешаем прерывания, очищаем и размещаем кэш

or 3800h, ST

Br begin;переход к началу приложения

nop

nop

nop

.data

N .word 5; длина массива

M .word 13h, -8h, 2h, 6h, 10h

a .word 1; счетчик отрицательных чисел

b .word 2

.text

begin;Начало программы

ldp a

ldi N;

ldi @N, AR1; загружаем длину массива

ldi @M, AR0; загружаем первое слово и сравниваем с нулем

ldi @b, AR2; число с которым сравниваем

ldi @a, AR3

ldi 0, AR2

rpts 4;счетчик

cmpi AR0,AR2

bgt thenlbl

br elselbl

thenlbl

addi 1,AR3

addi 1, AR1

sti AR3, @a

br endiflbl

elselbl

addi 0,AR3

addi 1, AR1

sti AR3, @a

endiflbl

ldhi 8000h,AR5

or 0050h,AR5;AR5=80001000h

sti AR3,\*AR3

stop br stop;

nop

nop

.end

Для проверки закомпилировал проект и получил результат, показанный на рисунке 1.

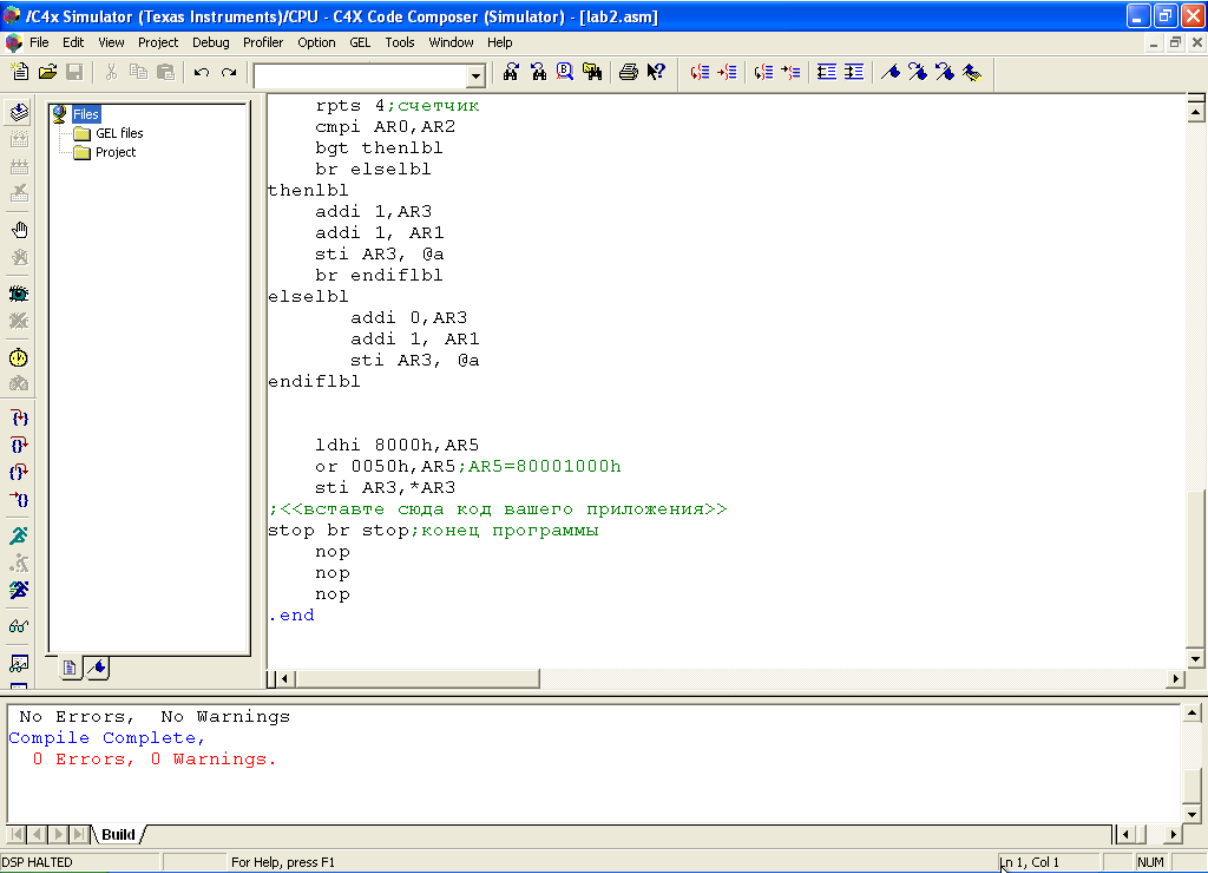


Рисунок 1 – Компиляция проекта