**Национальный исследовательский ядерный университет «МИФИ»**

|  |  |
| --- | --- |
|  | **Факультет Кибернетики и информационной безопасности**  **Кафедра «Компьютерные системы и технологии»** |

**ОТЧЕТ**

**по выполнению лабораторного практикума**

**по куру «Микропроцессорные устройства и системы»**

Студент гр. Б18-503 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Григорьев М.П. /

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ёхин М.Н. /

**Лабораторная работа №1, 2**

Вариант 4

F (X3, X2, X1, X0) = 2,4,5,6,7,8,15

Вариант 4;

Режим Т0 - 3

2 сек – 2, 9

3 сек – 13, 8

4 сек – 5

5 сек – 14, 7

Для всех остальных наборов – 1 сек

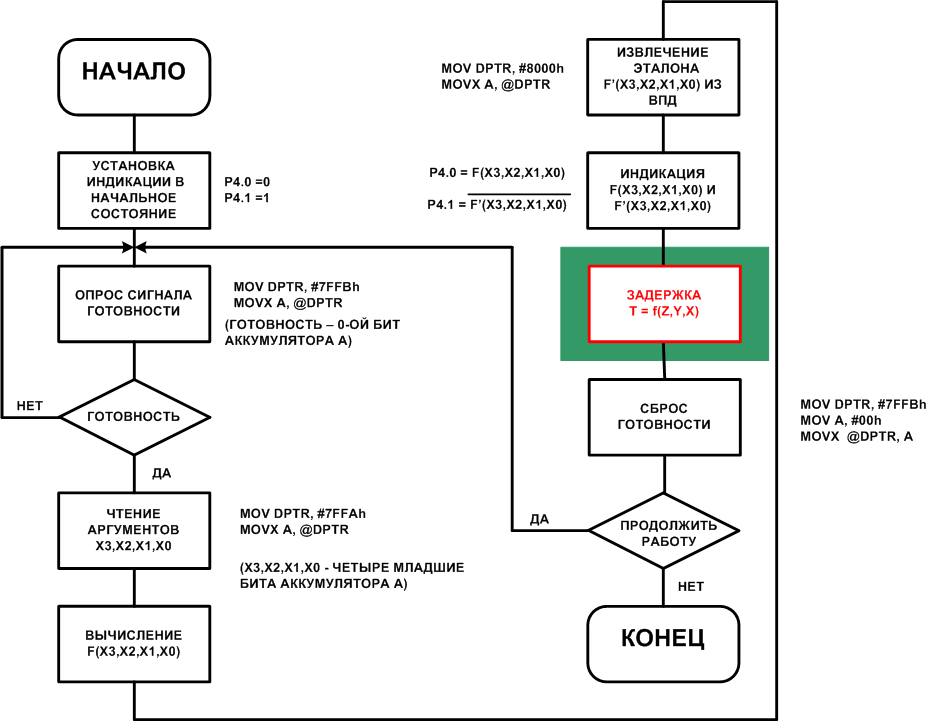


Рисунок 1 – Алгоритм работы программы

**Исходный код программы:**

ORG 8000h

; Загрузка значений таймеров

MOV 40H, #1; 0

MOV 41H, #1; 1

MOV 42H, #2; 2

MOV 43H, #1; 3

MOV 44H, #1; 4

MOV 45H, #4; 5

MOV 46H, #1; 6

MOV 47H, #5; 7

MOV 48H, #3; 8

MOV 49H, #2; 9

MOV 4AH, #1; 10

MOV 4BH, #1; 11

MOV 4CH, #1; 12

MOV 4DH, #3; 13

MOV 4EH, #5; 14

MOV 4FH, #1; 15

P4 EQU 0C0h

MOV P4, #02h

POLL:

MOV DPTR, #7FFBh

MOVX A, @DPTR

JNB ACC.0, POLL; Проверка бита готовности

MOV DPTR, #7FFAh

MOVX A, @DPTR; Загрузка Х3-Х0

MOV R0, A

; not x3 x2

MOV C, ACC.3 ; C = X3

CPL C ; C = not X3

ANL C, ACC.2 ; C = (not X3) & X2

MOV 01, C ; bit memory(01) = C

; x2 x1 x0

MOV C, ACC.2 ; C = X2

ANL C, ACC.1 ; C = X2 & X1

ANL C, ACC.0 ; C = X2 & X1 $ X0

ORL C, 01

MOV 01, C ; bit memory(01) = C

; not x3 x1 not x0

MOV C, ACC.3 ; C = X3

CPL C ; C = not X3

ANL C, ACC.1 ; C = not X3 & X1

MOV 00, C ; bit memory(00) = C = not X3 & X1

MOV C, ACC.0 ; C = X0

CPL C ; C = not X0

ANL C, 00 ; C = not X3 & X1 & not X0

ORL C, 01

MOV 01, C ; bit memory(01) = C

; X3 not x2 not x1 not x0

MOV C, ACC.2 ; C = X2

CPL C ; C = not X2

ANL C, ACC.3 ; C = X3 & not X2

MOV 00, C ; bit memory(00) = C = X3 & not X2

MOV C, ACC.1 ; C = X1

CPL C ; C = not X1

ANL C, 00 ; C = X3 & not X2 & not X1

MOV 00, C ; bit memory(00) = C = X3 & not X2 & not X1

MOV C, ACC.0 ; C = X0

CPL C ; C = not X0

ANL C, 00 ; C = X3 & not X2 & not X1 & not X1

ORL C, 01

MOV P4.0, C ; Загрузка в Р4.0 F

JB ACC.3, HIGHB; Прочитать старшие или младшие биты эталонна

MOV DPTR, #8000h ; Загрузка эталона для наборов 0-7

AJMP SHIFT

HIGHB:

MOV DPTR, #8001h ; Загрузка эталона для наборов 8-15

SHIFT:

MOVX A, @DPTR

MOV R1, A ; Запись эталона в R1

MOV A, R0 ; Записать ZYX в A

MOV R3, A ; Сохранить X

JZ GETST ; A = 0

JNB ACC.3, CONT

CPL ACC.3 ; Сброс старшего бита

MOV R0, A

JZ GETST

CONT:

MOV A, R1

DECCYCLE:

DEC R0 ; R0 = R0 - 1

RRC A ; Сдвиг эталона

CJNE R0, #00h, DECCYCLE

AJMP RESULT

GETST:

MOV A, R1

RESULT:

MOV C, ACC.0

MOV P4.1, C ; Запись эталонного значения

; Time

MOV A, R3

ANL A, #00001111B;

ADD A, #40H

MOV R1, A

MOV A, @R1

MOV B, #17

MUL AB

MOV R3, #200;

MOV TMOD, #00000011B ; Задание режима таймера Т3

PreTimer3:

MOV R2, A

Timer:

CLR TR0;

MOV TL0, #00AH;

SETB TR0;

WORK:

JBC TF0, WAIT ; Ожидание завершения подсчета таймера

SJMP WORK

WAIT:

DJNZ R2, TIMER; Цикл для 1 сек

DJNZ R3, PreTimer3; Внешний цикл для 2-5 сек

MOV DPTR, #7FFbh

MOV A, #00h

MOVX @DPTR, A ; Сброс бита готовности

AJMP POLL ; Возвращение на опрос бита готовности

END

**Результаты тестирования программы:**

Для тестирования программы необходимо загрузить инвертированные эталонные значения функции и сравнить получившиеся в ходе выполнения программы значения с эталонными. Они будут противоположны по значению.

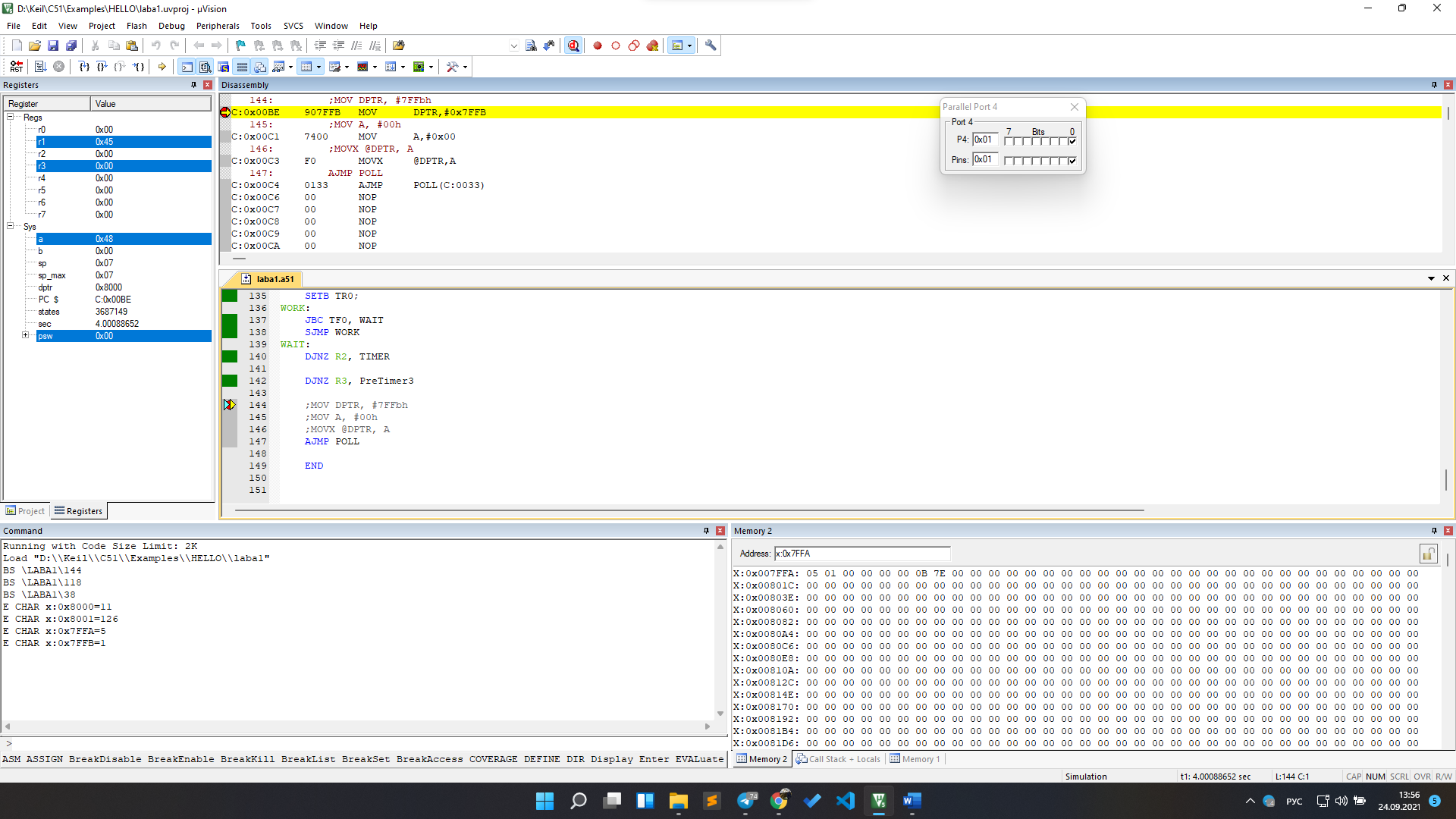


Рисунок 2 ­­– Проверка пятого набора

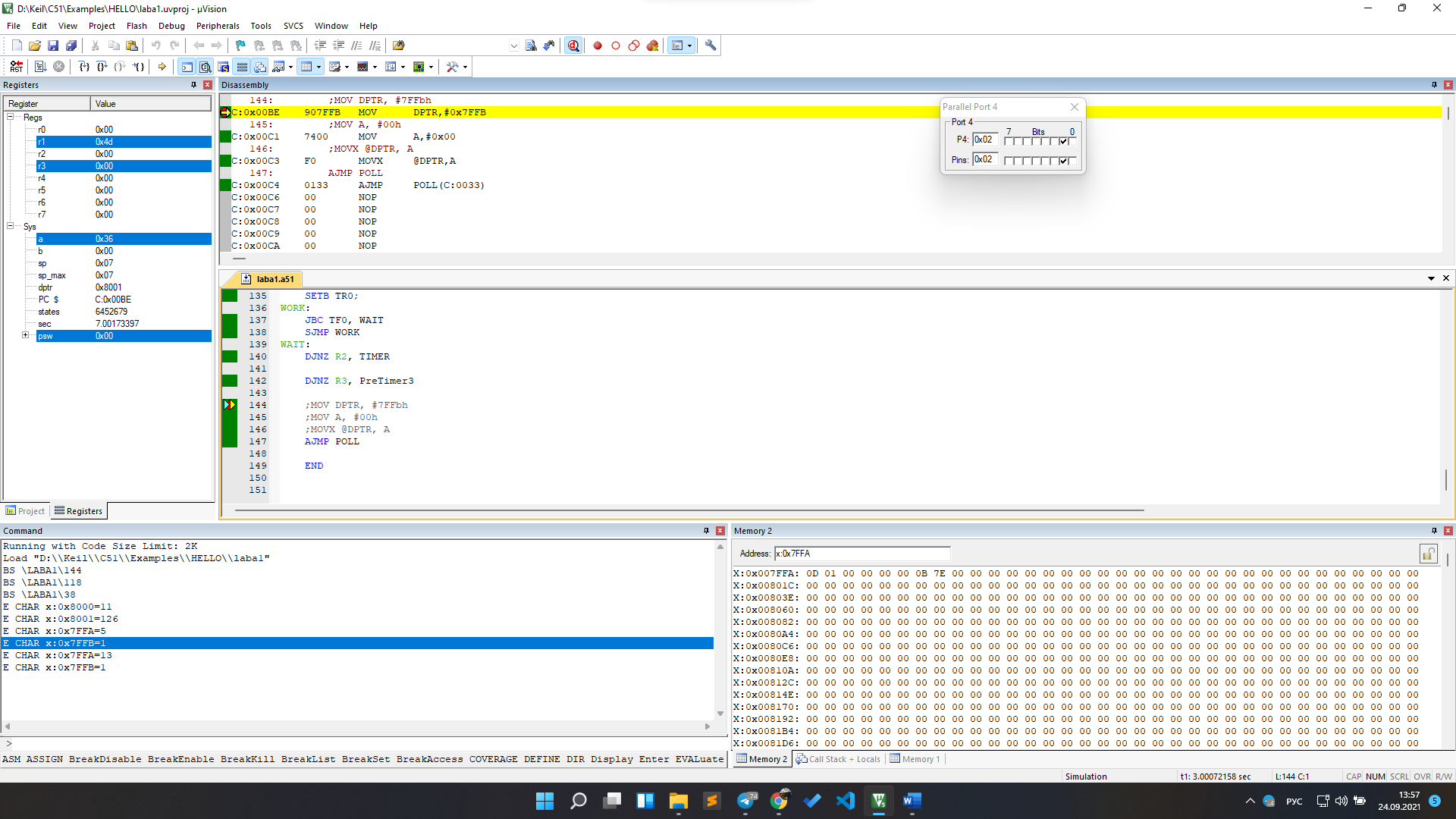


Рисунок 3 – Проверка 13го набора

Итог: тестирование программы выполнено успешно, все требования выполнены, так как полученные значения совпали с эталонными.

**Результаты тестирования программы:**

Для тестирования программы необходимо задать необходимый режим работы таймера, а также проверить время выполнения программы в зависимости от заданных наборов. Соответствие времени выполнения наборам указано в условии.

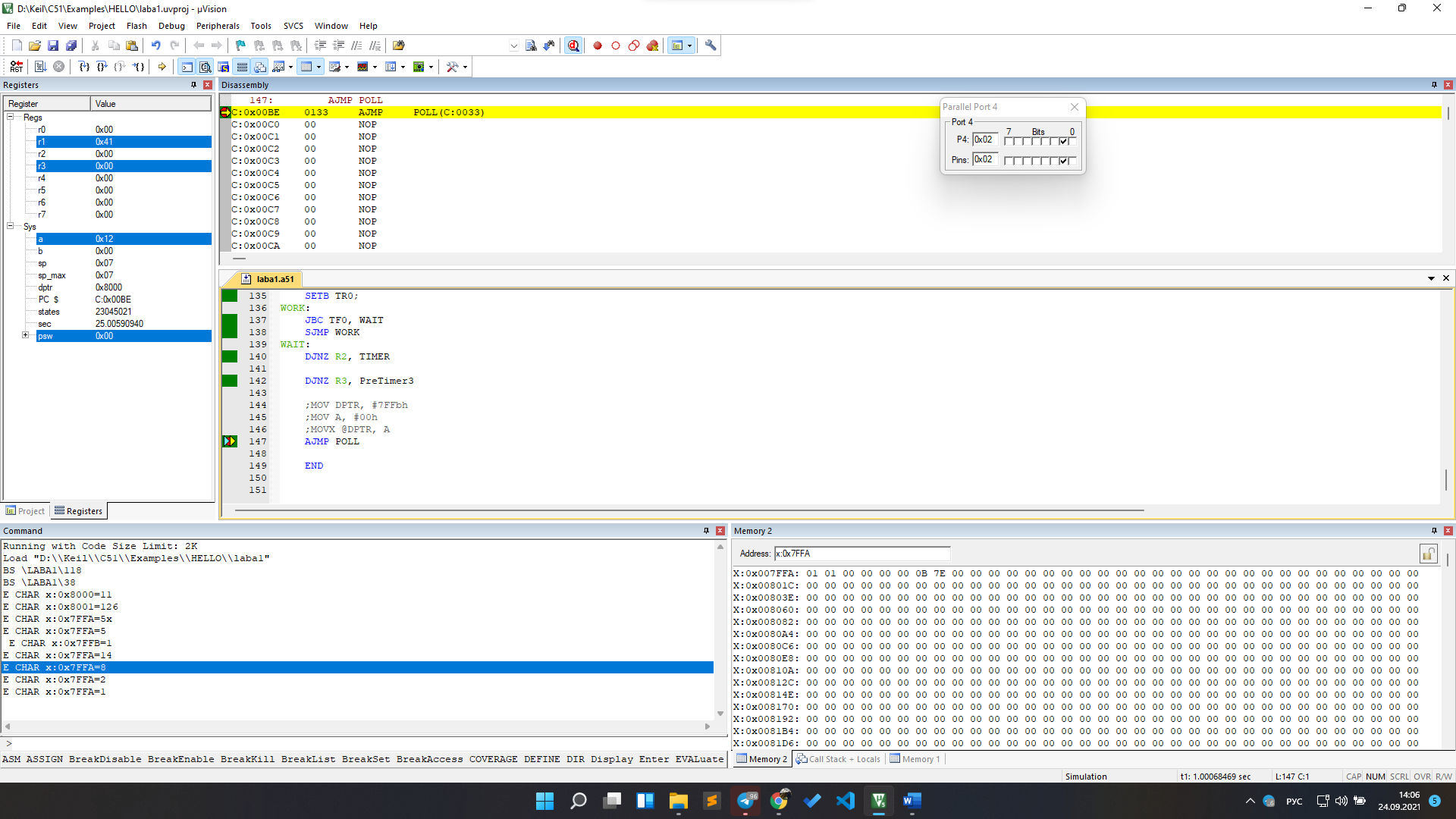


Рисунок 5 – Проверка времени выполнения для 1 набора, равного 1 сек

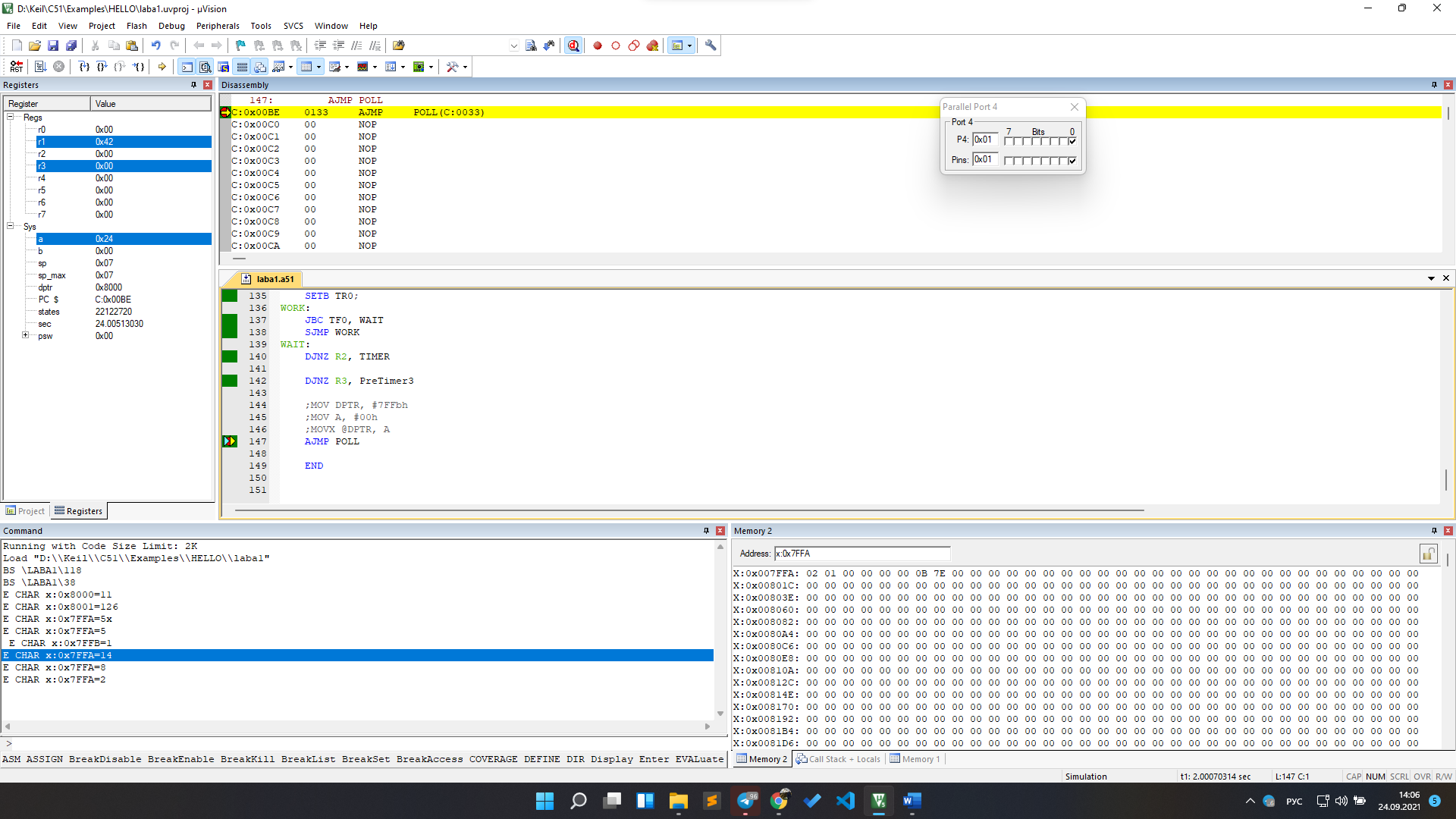


Рисунок 6 – Проверка времени выполнения для 2 набора, равного 2 сек

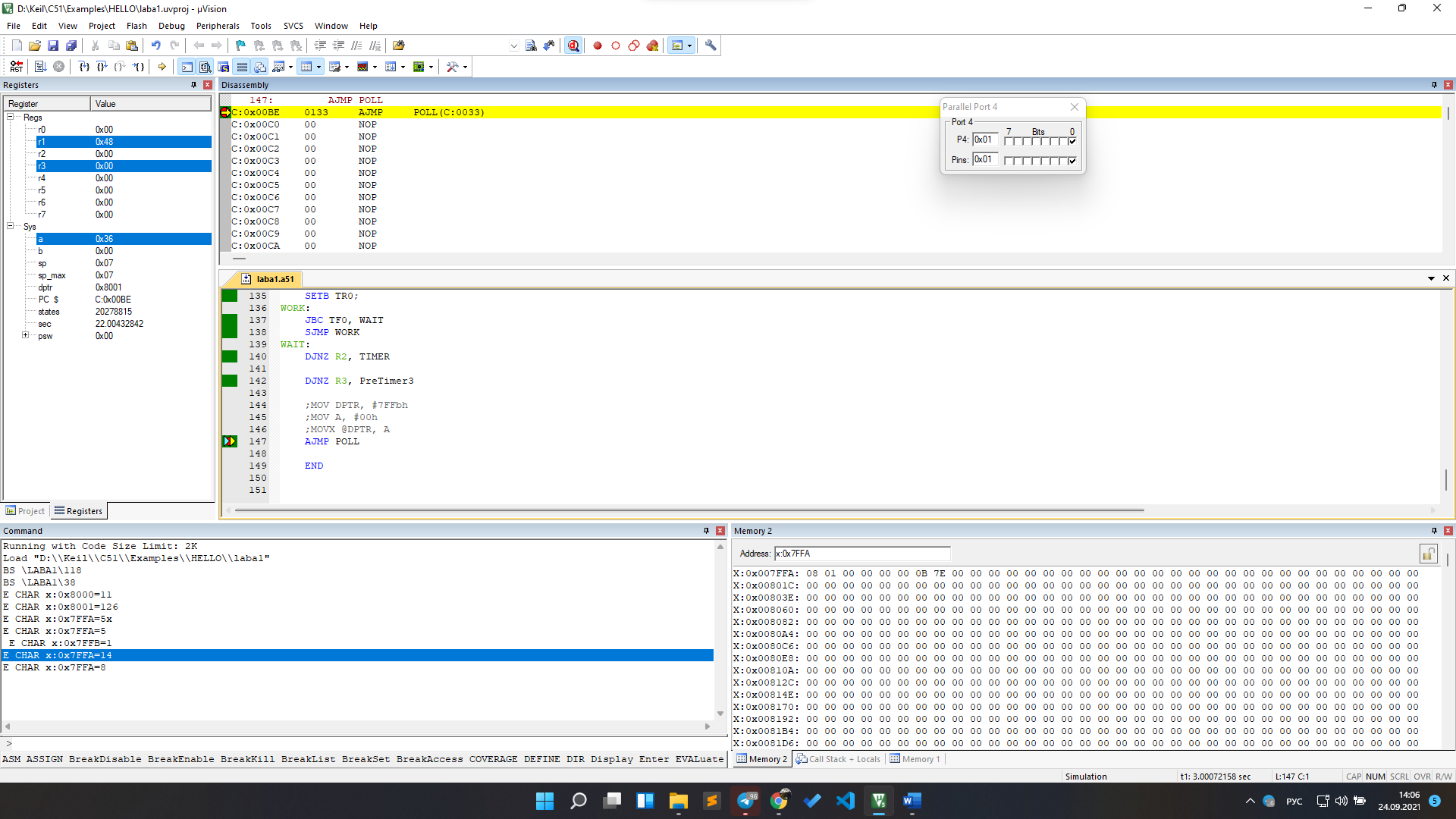


Рисунок 7 – Проверка времени выполнения для 8 набора, равного 3 сек

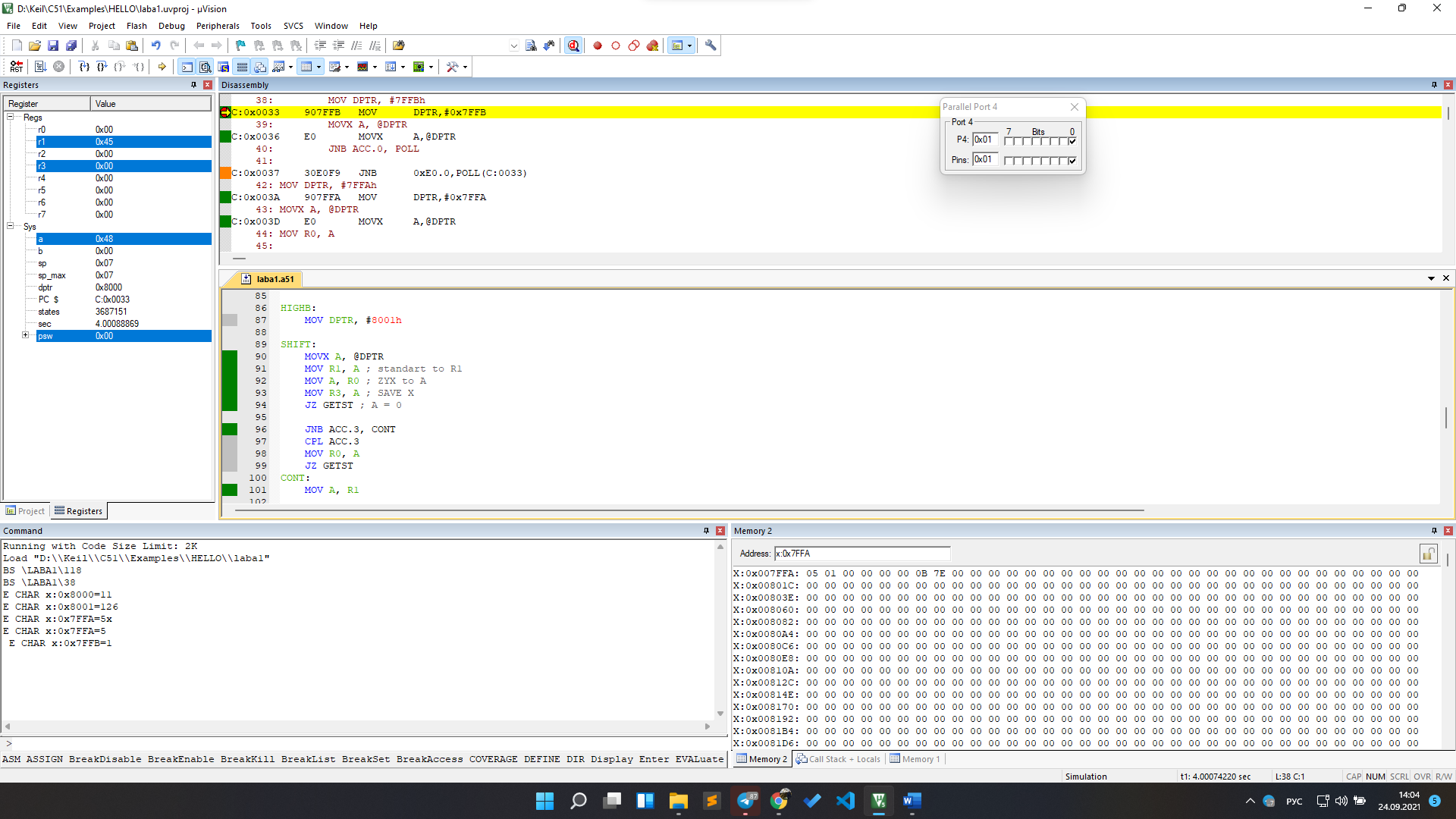


Рисунок 8 – Проверка времени выполнения для 5 набора, равного 4 сек

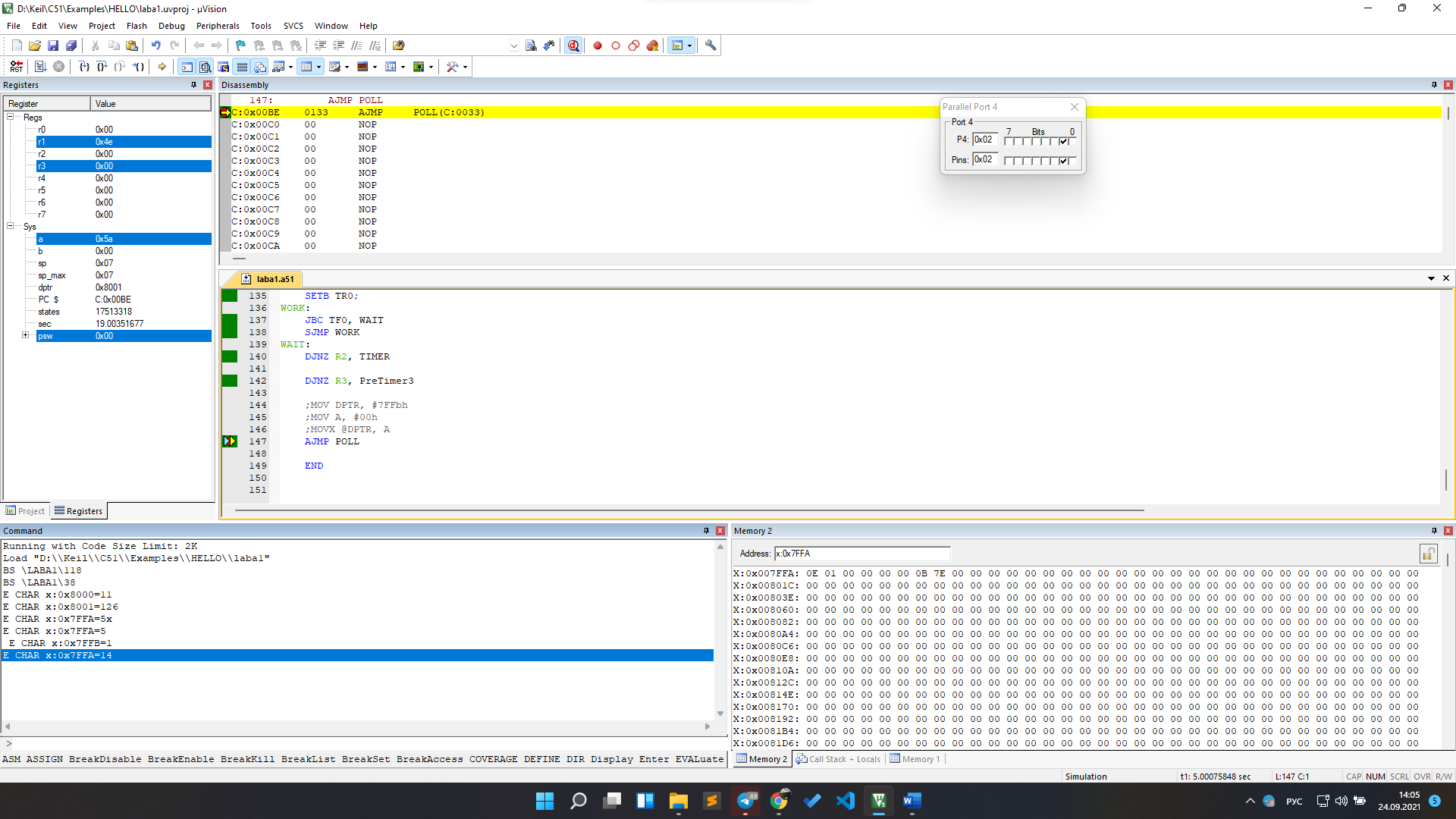


Рисунок 9 – Проверка времени выполнения для 14 набора, равного 5 сек

Итог: тестирование программы выполнено успешно, все требования выполнены, так как время выполнения совпадает с заданным в условии.