

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ»**



**Институт Интеллектуальных Кибернетических Систем
Кафедра «Компьютерные системы и технологии»**

**Отчёт о лабораторной работе №1-2
«Реализация переключательной функции с формированием задержек
при переключении»**

Студент группы Б20-503 Коломенский В. Г. / _____ /
Руководитель _____ / _____ /

Москва 2023

Оглавление

1. Введение.....	1
2. Построение МДНФ	2
3. Расчёт задержки.....	3
4. Реализация программы	4
5. Тестирование программы.....	8
6. Список литературы и ссылки.....	14

1. Введение

По заданию лабораторной работы было необходимо реализовать переключательную функцию, которая будет принимать истинное значение на наборах, указанных на рисунке 1. Номер набора определяется 4 битами, считываемыми из внешней памяти.

В качестве результата необходимо вывести на индикаторы само рассчитанное в процессе выполнения программы значение функции и эталонное значение (инвертированное значение функции на заданном наборе).

В зависимости от номера набора в процессе формирования значения функции, в программе должно быть реализовано извлечение эталонного значения из памяти и формирование задержки перед сбросом бита готовности. Величина задержки на каждом наборе формировалась исходя из задания, которое приведено на рисунке 1.

№ ВАРИАНТА	ПЕРЕКЛЮЧАТЕЛЬНАЯ ФУНКЦИЯ $F(X_3, X_2, X_1, X_0)$	КОС. АДРЕС	БАЗ. АДРЕС	2 СЕК	3 СЕК	4 СЕК	5 СЕК	РЕЖИМ T0
8	2,3,4,9,11,12,13	08	15	0,13	4,9	2,11	12	3

Рисунок 1. – Условие задания

2. Построение МДНФ

Перед началом написания кода программы необходимо построить МДНФ. Эталонная диаграмма Вейча и МДНФ приведены на рисунке 2.

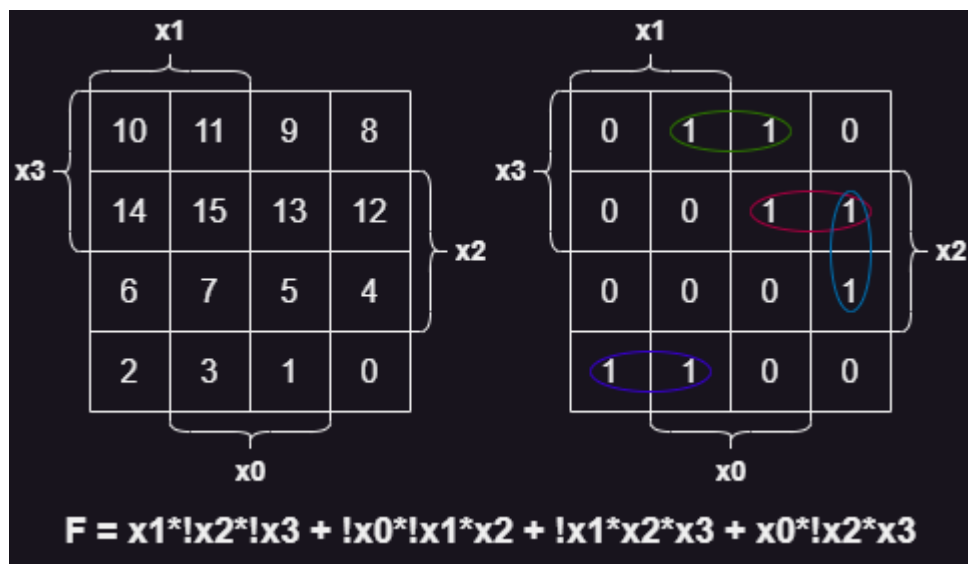


Рисунок 2. – Минимизация функции

3. Расчёт задержки

В режиме работы T0 = 3 разрядность таймерного регистра составляет 8 бит. При частоте 11,059 МГц за 1 секунду счётчик с разрядностью в 8 бит переполнится $(2^8 - 16) * 15$ раз. Поэтому для задания нужной задержки будет использоваться два вложенных цикла. Один из них работает $(2^8 - 16)$ раз, другой работает $15 * n$ раз, где n – количество секунд задержки. Переход на следующую итерацию вложенного цикла происходит при переполнении регистра таймера.

4. Реализация программы

Реализация программы представлена в приложении 1.

Приложение 1. – реализация программы

```
1 ; F(x3, x2, x1, x0) = 2, 3, 4, 9, 11, 12, 13
2 ; R0 - address of the reference function
3 ; R1 - shift counter
4
5 ; Start
6     ORG     0000h           ; Starting address
7     P4     EQU 0C0h        ; Define P4
8
9 ; Preparing the environment for work
10    MOV     DPTR, #8000h    ; \;
11    MOV     A, #08h         ; | Loading an indirect address
12    MOVX    @DPTR, A        ; /
13
14    MOV     DPTR, #8008h    ; \;
15    MOV     A, #15h         ; | Loading the base address
16    MOVX    @DPTR, A        ; /
17
18    MOV     DPTR, #8000h    ; \;
19    MOVX    A, @DPTR        ; | Calculating the address of the reference
20    MOV     DPL, A          ; | function in external memory and writing
21    MOVX    A, @DPTR        ; | this address to the register R0
22    MOV     DPL, A          ; |
23    MOV     R0, DPL         ; /
24
25    MOV     A, #01Ch        ; \;
26    MOVX    @DPTR, A        ; | Writing a reference function
27    INC     DPTR            ; | to external memory
28    MOV     A, #03Ah        ; |
29    MOVX    @DPTR, A        ; /
30
31    MOV     DPTR, #8020h
32    MOV     A, #2
33    MOVX    @DPTR, A
34
35    MOV     DPTR, #8021h
36    MOV     A, #1
37    MOVX    @DPTR, A
38
39    MOV     DPTR, #8022h
40    MOV     A, #4
41    MOVX    @DPTR, A
42
43    MOV     DPTR, #8023h
44    MOV     A, #1
45    MOVX    @DPTR, A
46
47    MOV     DPTR, #8024h
48    MOV     A, #3
```

```

49     MOVX     @DPTR, A
50
51     MOV      DPTR, #8025h
52     MOV      A, #1
53     MOVX     @DPTR, A
54
55     MOV      DPTR, #8026h
56     MOV      A, #1
57     MOVX     @DPTR, A
58
59     MOV      DPTR, #8027h
60     MOV      A, #1
61     MOVX     @DPTR, A
62
63     MOV      DPTR, #8028h
64     MOV      A, #1
65     MOVX     @DPTR, A
66
67     MOV      DPTR, #8029h
68     MOV      A, #3
69     MOVX     @DPTR, A
70
71     MOV      DPTR, #802Ah
72     MOV      A, #1
73     MOVX     @DPTR, A
74
75     MOV      DPTR, #802Bh
76     MOV      A, #4
77     MOVX     @DPTR, A
78
79     MOV      DPTR, #802Ch
80     MOV      A, #5
81     MOVX     @DPTR, A
82
83     MOV      DPTR, #802Dh
84     MOV      A, #2
85     MOVX     @DPTR, A
86
87     MOV      DPTR, #802Eh
88     MOV      A, #1
89     MOVX     @DPTR, A
90
91     MOV      DPTR, #802Fh
92     MOV      A, #1
93     MOVX     @DPTR, A
94
95 ; Preparing input values
96 PREPARING:
97     MOV      DPTR, #7FFBh ; \;
98     MOVX     A, @DPTR     ; | Cyclic polling of the readiness bit
99     JZ       PREPARING    ; /
100    MOV      DPTR, #7FFAh  ; \;
101    MOVX     A, @DPTR     ; | Writing the input value
102    MOV      20h, A        ; /
103
104 ; Calculation of a logical function
105    MOV      C, 1          ;
106    ANL      C, /2         ;

```

```

107     ANL     C, /3           ;
108     MOV     8, C           ;
109     MOV     C, 2           ;
110     ANL     C, /0           ;
111     ANL     C, /1           ;
112     MOV     9, C           ;
113     MOV     C, 2           ;
114     ANL     C, /1           ;
115     ANL     C, 3           ;
116     MOV     10, C          ;
117     MOV     C, 0           ;
118     ANL     C, /2           ;
119     ANL     C, 3           ;
120     ORL     C, 8           ;
121     ORL     C, 9           ;
122     ORL     C, 10          ;
123     MOV     8, C           ;
124
125 ; Comparison of the obtained result with the reference function
126     MOV     DPTR, #8000h    ; \;Preparing the address of
127     MOV     DPL, R0         ; / the reference function
128     JB      3, PTR_1        ; \;Checking which half
129     AJMP    PTR_2          ; / of the reference function to use
130 PTR_1:
131     INC     DPTR           ; \;Preparation for working with the
132     CLR     3             ; / second half of the reference function
133 PTR_2:
134     MOVX    A, @DPTR       ;
135     MOV     R1, 20h        ; Preparing the shift counter
136 SHIFT:
137     CJNE    R1, #00h, PTR_3 ; \;
138     AJMP    PTR_4          ; |
139 PTR_3:      ; | Shift the desired bit to
140     RR      A              ; | the first (index=0) position
141     DEC     R1             ; |
142     AJMP    SHIFT         ; /
143 PTR_4:
144     MOV     C, 8           ; \;
145     MOV     P4.0, C        ; | Output of the received result
146     MOV     C, ACC.0       ; | and the reference to the P4 channel
147     CPL     C              ; |
148     MOV     P4.1, C        ; /
149     MOV     DPTR, #7FFAh   ;
150     MOVX    A, @DPTR       ;
151     MOV     DPTR, #8000h   ;
152     ANL     A, #00001111b  ;
153     ADD     A, #20h        ;
154     MOV     DPL, A         ;
155     MOVX    A, @DPTR       ;
156     MOV     B, A           ;
157     MOV     A, #15d        ;
158     MUL     AB             ;
159     MOV     R2, A          ;
160     MOV     TMOD, #00000011b ;
161     CLR     TR0            ;
162     MOV     TL0, #00h      ;
163     SETB    TR0           ;
164 PTR_5:

```



```

165     MOV     R1, #0FFh
166 PTR_6:
167     CLR     TR0
168     MOV     TL0, #16d
169     SETB    TR0
170 PTR_7:
171     JBC     TF0, PTR_8
172     AJMP    PTR_7
173 PTR_8:
174     DJNZ    R1, PTR_6
175     DJNZ    R2, PTR_5
176     MOV     DPTR, #7FFBh ; \;
177     MOV     A, #00h ; | Resetting the ready bit to 0
178     MOVX    @DPTR, A ; /
179     MOV     DPTR, #7FFAh ; \;
180     MOVX    A, @DPTR ; | The following set
181     INC     A ; | of input data
182     MOVX    @DPTR, A ; /
183
184     AJMP    PREPARING
185
186 END

```

5. Тестирование программы

В рамках тестирования разработанной программы были проверены все 16 возможных входных комбинаций. Результат тестирования представлен на рисунках 3.1 – 3.16 ниже.

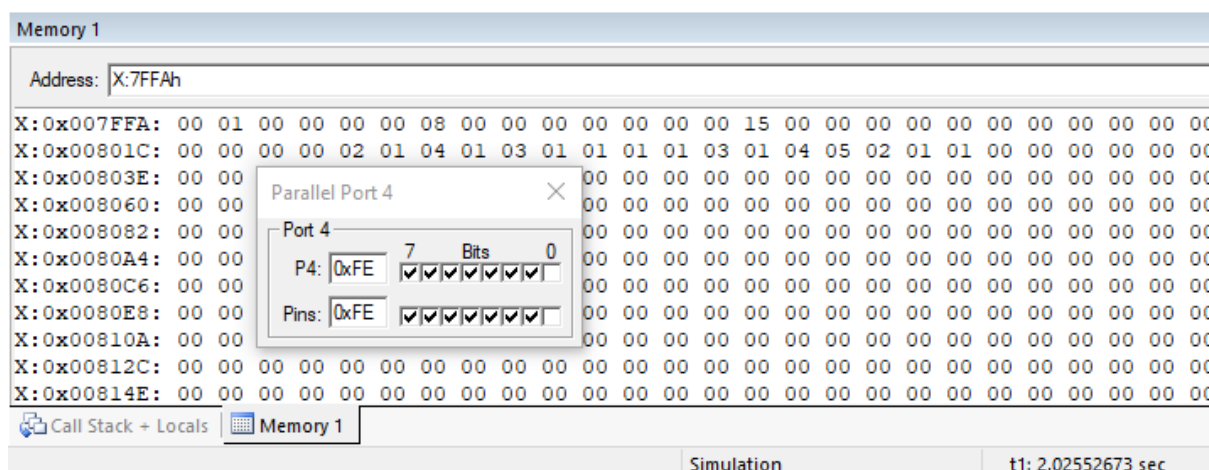


Рисунок 3.1 – Результат тестирования на наборе «0».

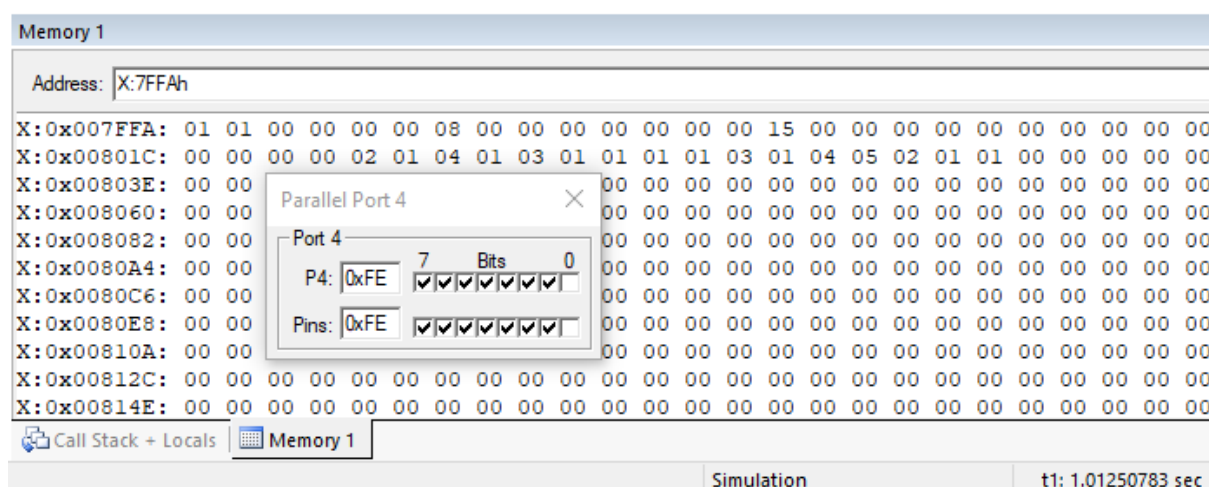


Рисунок 3.2 – Результат тестирования на наборе «1».

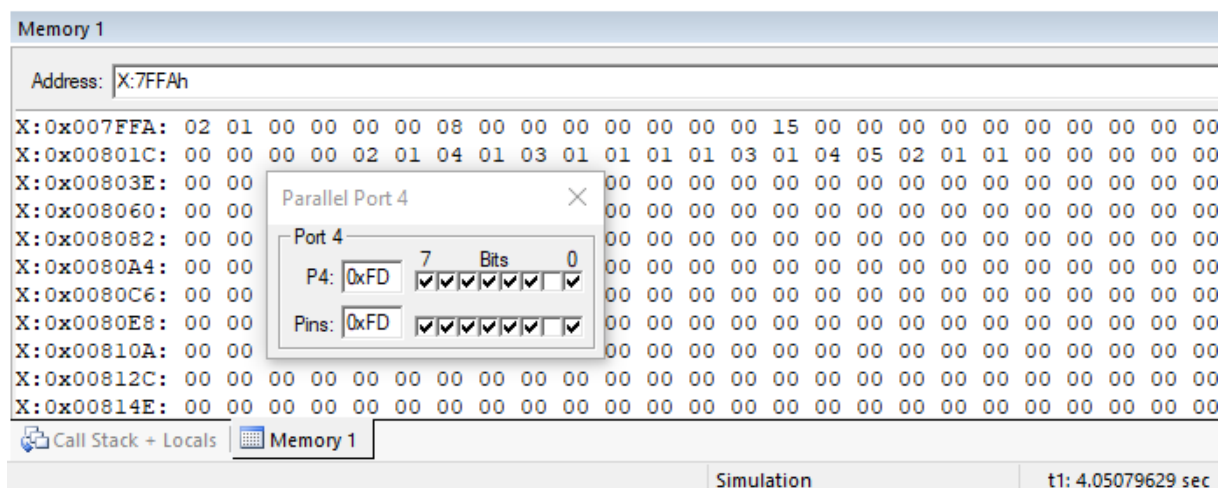


Рисунок 3.3 – Результат тестирования на наборе «2».

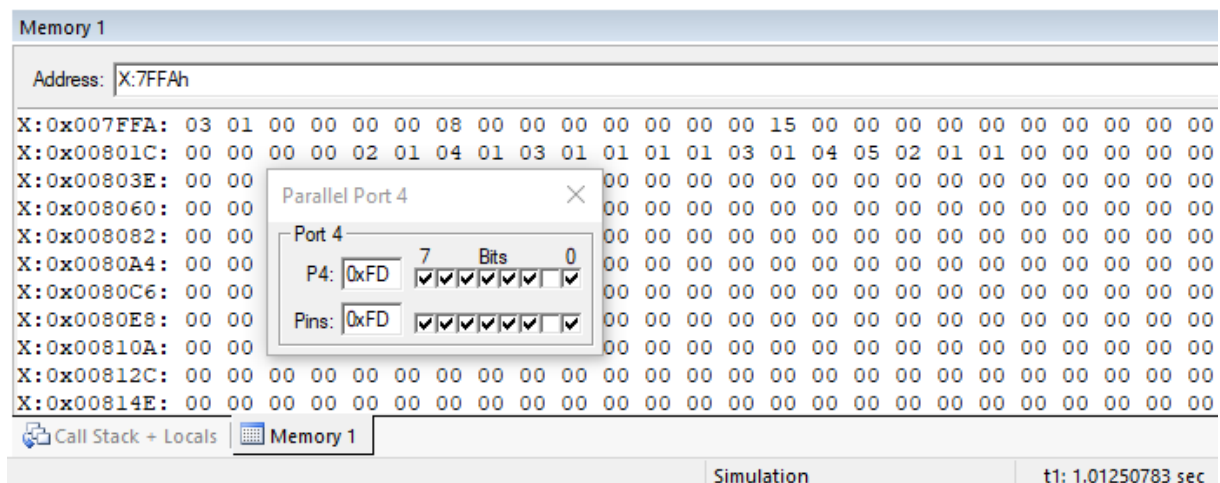


Рисунок 3.4 – Результат тестирования на наборе «3».

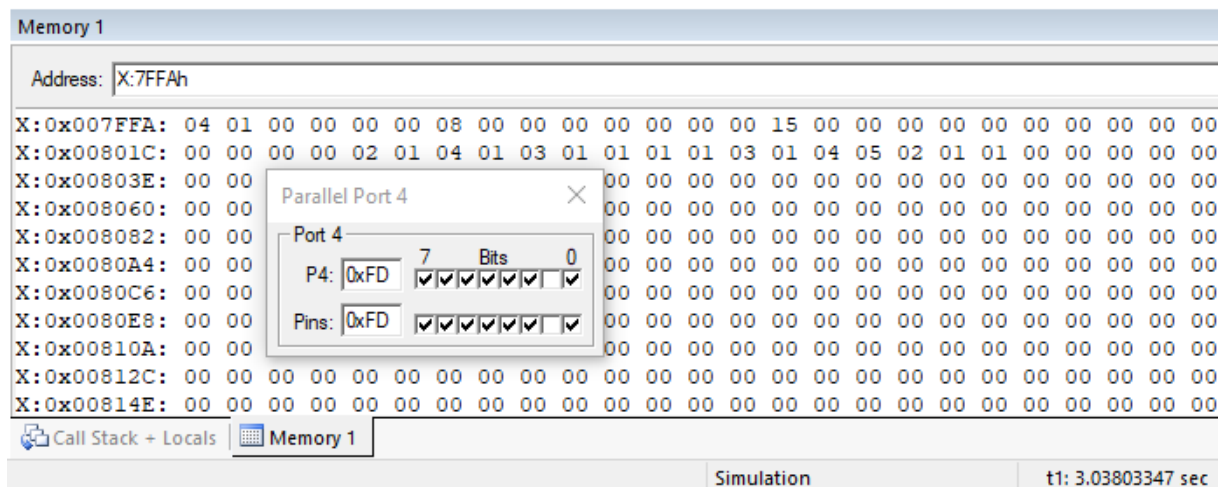


Рисунок 3.5 – Результат тестирования на наборе «4».

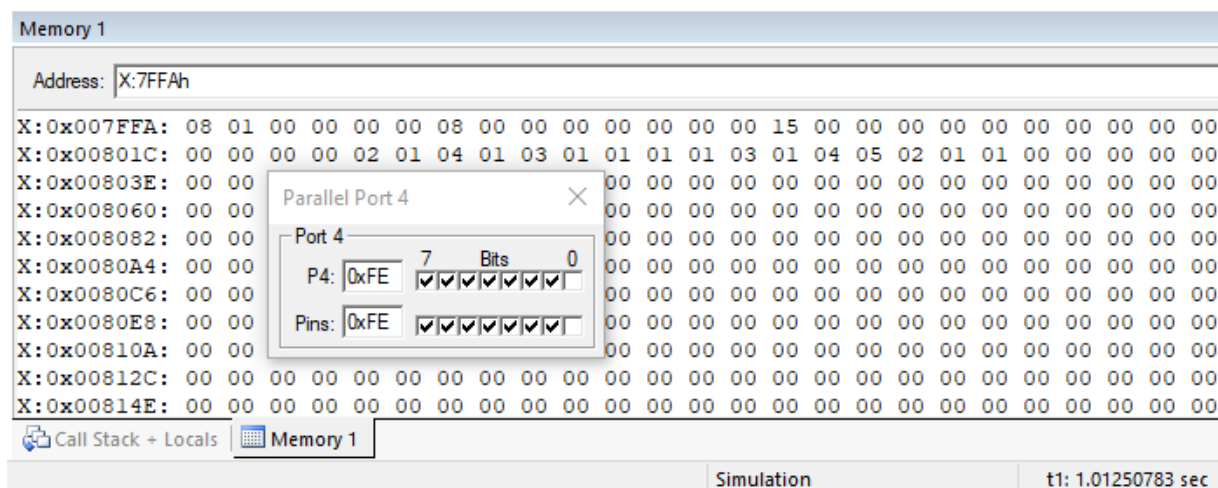


Рисунок 3.9 – Результат тестирования на наборе «8».

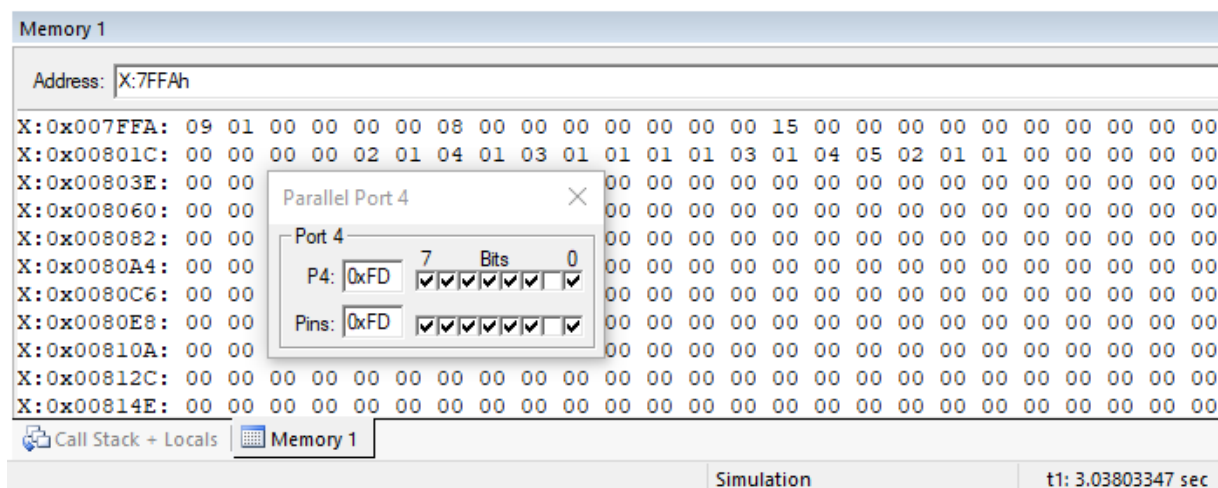


Рисунок 3.10 – Результат тестирования на наборе «9».

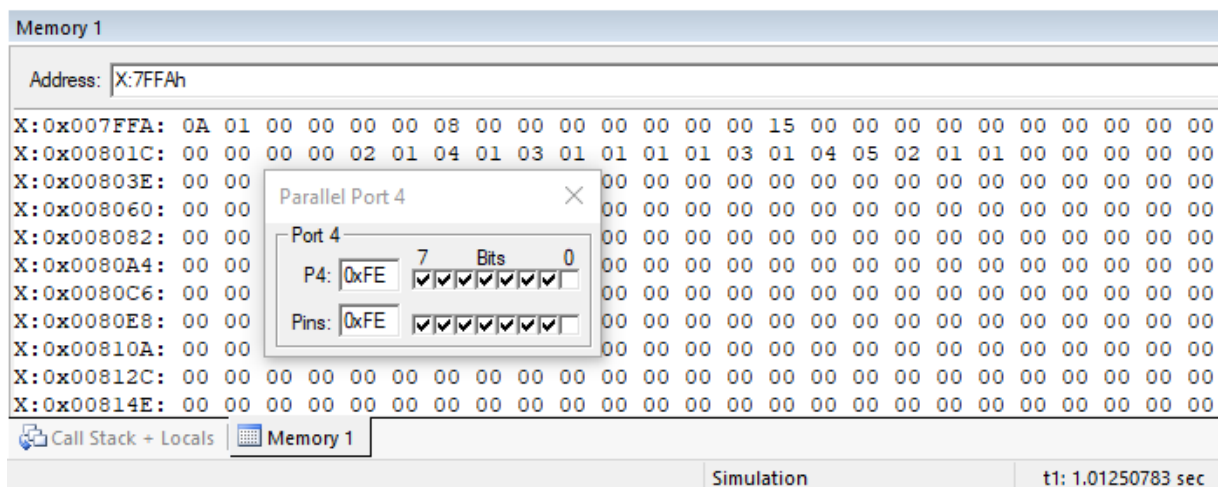


Рисунок 3.11 – Результат тестирования на наборе «A».

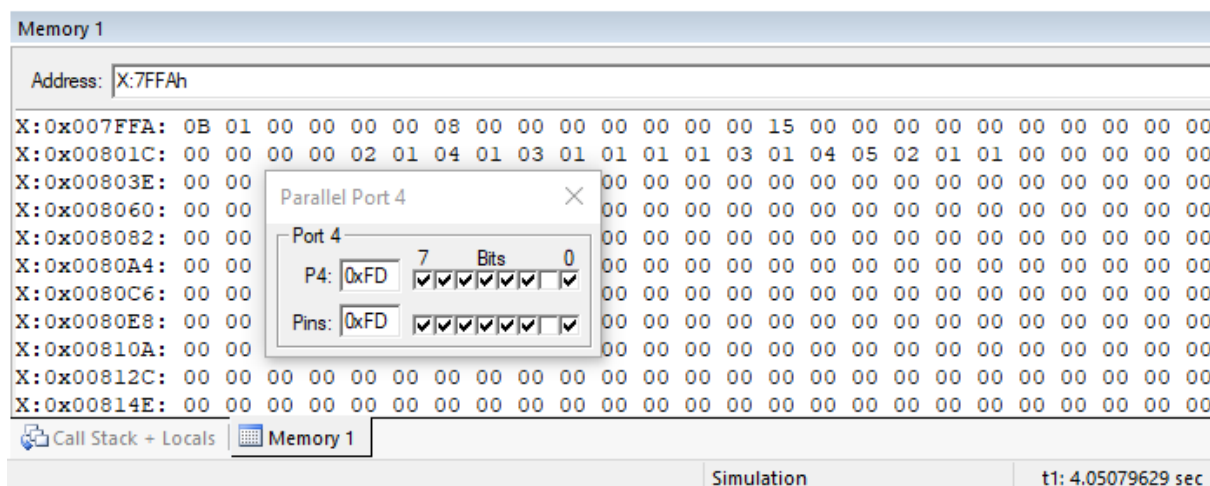


Рисунок 3.12 – Результат тестирования на наборе «В».

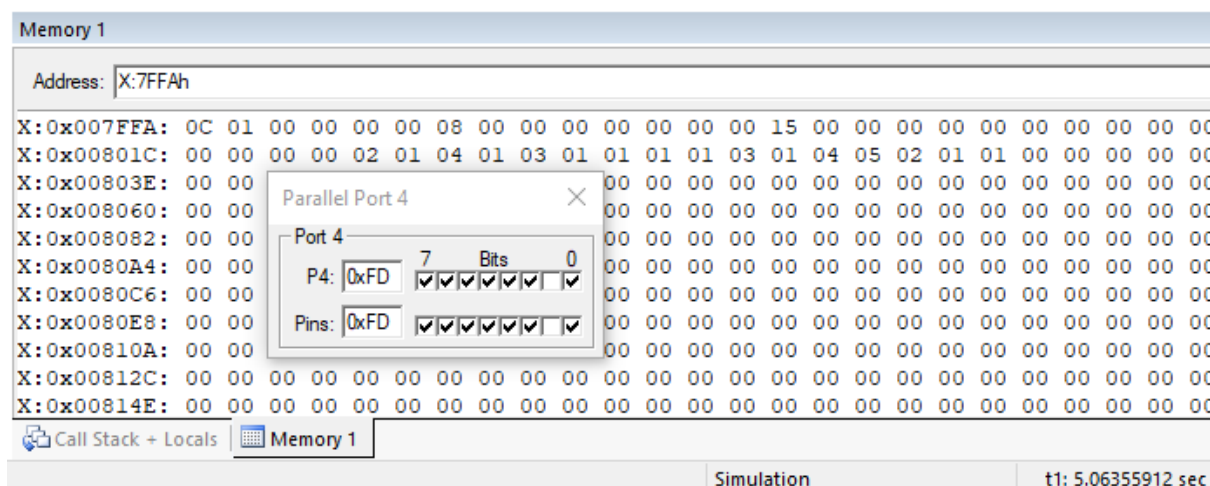


Рисунок 3.13 – Результат тестирования на наборе «С».

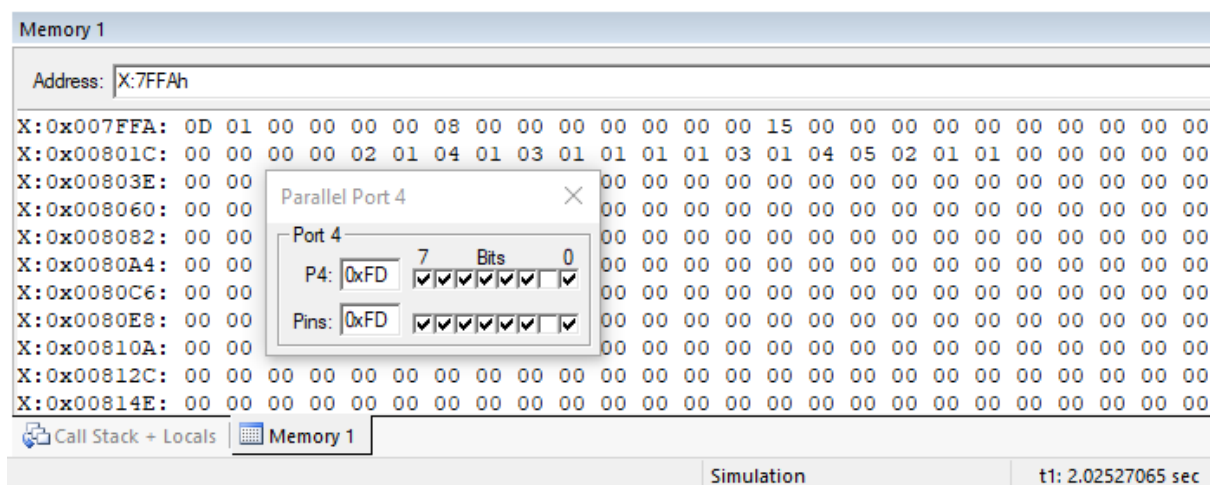


Рисунок 3.14 – Результат тестирования на наборе «D».

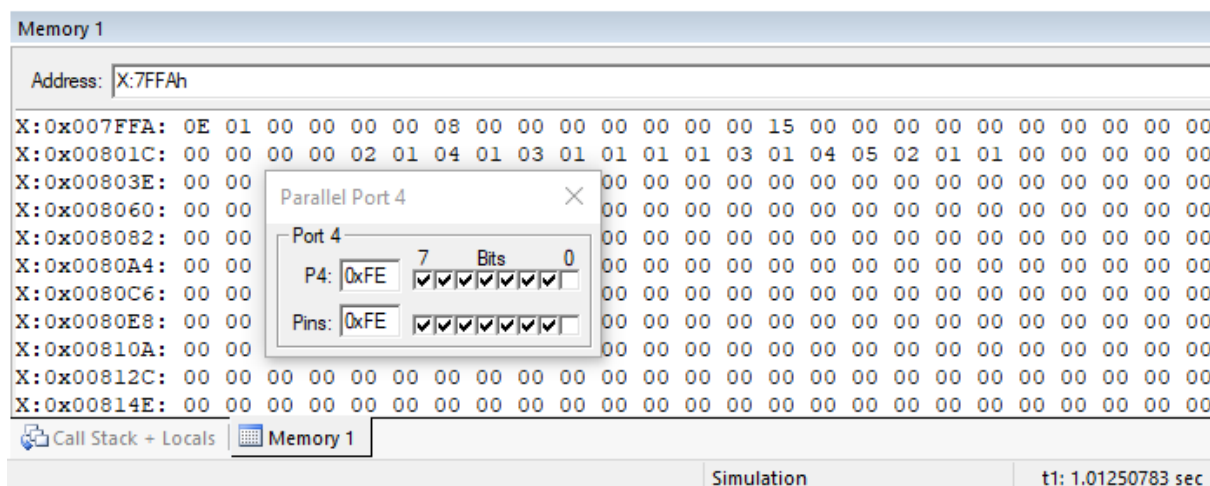


Рисунок 3.15 – Результат тестирования на наборе «Е».

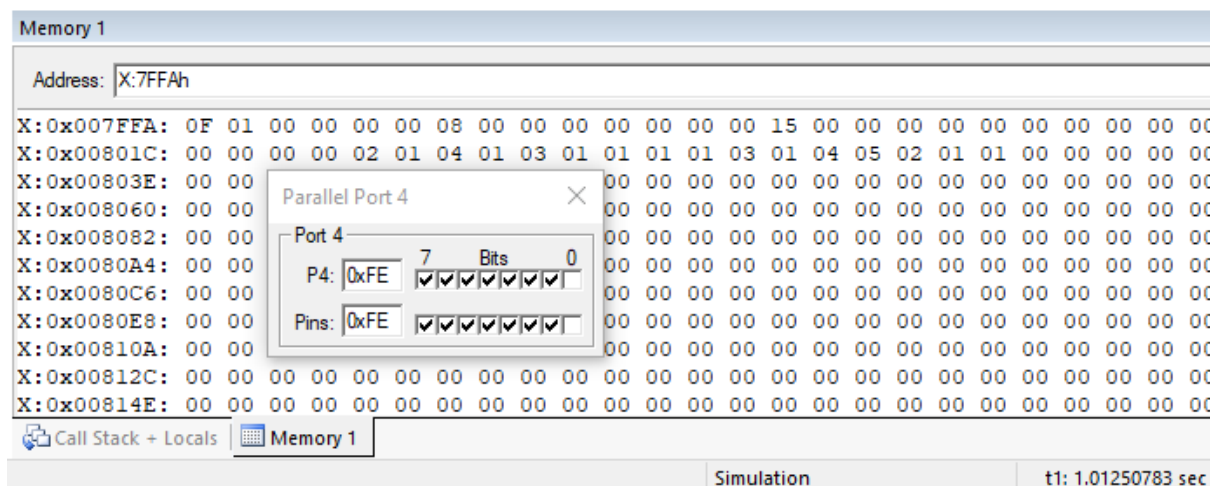


Рисунок 3.16 – Результат тестирования на наборе «F».

6. Список литературы и ссылки

1. Стрелец А. И., Иванников В. С., Ёхин М.Н. Методические указания для выполнения лабораторной работы “Битовый процессор” по курсу “Микропроцессорные устройства и системы” с использованием виртуального стенда. Москва 2018.
2. Е. В. Моисейкин. Микроконтроллеры семейства MCS-51 Теория и практика. Учебно-методическое пособие. Екатеринбург Издательство Уральского университета 2017.
3. Исходный код программы [Электронный ресурс] // <https://github.com/Hypex146/MDnS-Lab-2>.