

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики

институт

Базовая кафедра вычислительных и информационных технологий

кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8

Преподаватель

Студент ИМ22-05Б

номер группы, зачетной книжкой

подпись, дата

подпись, дата

Середкин В. Г.

инициалы, фамилия

Попов Д. Д.

инициалы, фамилия

Красноярск 2025

Содержание

1	Цель работы	3
2	Задание	3
3	Ход выполнения работы	4
3.1	План работы	4
3.2	Принцип работы	5
3.3	Используемые компоненты	5
3.4	Информационная схема	6
3.5	Электрическая блок-схема.....	6
3.6	Функциональная схема	7
3.7	Формирование адресов	8
3.8	Алгоритм функционирования	9
3.9	Листинг программного кода	

1 Цель работы

Проектирование системы управления объектом.

2 Задание

Разработать информационную модель и развернутую структурную схему (функциональную схему) для системы управления объектом, предложенной преподавателем. Информационную модель реализовать с учетом технического задания для проектирования выбранной системы.

Вариант:

Микропроцессорная система управления антиблокировкой колес в автомобиле.

3 Ход выполнения работы

3.1 План работы

Для упрощения проектирования системы управления объектом, разделим её на 4 задачи:

1. Принцип работы;
2. Перечислим используемые компоненты;
3. Реализация информационной структурной модели управления антиблокировкой колес в автомобиле.;
4. Реализуем функциональную схему управления антиблокировкой колес в автомобиле.

3.2 Принцип работы

ABS (антиблокировочная система тормозов) — это электронная система безопасности автомобиля, которая предотвращает блокировку колёс во время торможения. Это позволяет сохранить управляемость автомобилем и сократить тормозной путь на скользких поверхностях.

Система ABS состоит из датчиков скорости вращения колёс, гидравлического модулятора давления, электронного блока управления (ЭБУ) и исполнительных механизмов.

Датчики скорости установлены на каждом колесе и передают информацию о частоте вращения в электронный блок управления, подключённый к порту А параллельно программируемого интерфейса (PPI). ЭБУ анализирует данные от датчиков в реальном времени по шине данных.

Если при торможении один или несколько датчиков сообщают о резком снижении скорости вращения колеса (что может привести к его блокировке), ЭБУ отправляет сигнал через порт В PPI на гидромодулятор, который регулирует давление тормозной жидкости в системе.

Гидромодулятор осуществляет быстрое циклическое изменение давления, имитируя "прокачку" тормозов, что предотвращает полную блокировку колеса. Этот процесс происходит многократно в секунду и ощущается водителем как вибрация педали тормоза.

После завершения торможения система возвращается в исходное состояние и продолжает мониторинг параметров движения автомобиля, ожидая следующего сигнала от датчиков.

3.3 Используемые компоненты

Элементы:

1. Микропроцессор (CPU) – I8085A;
2. Параллельно программируемый адаптер (PPI) – I8255A;
3. Оперативная память (RAM);
4. Мультиплексор (MUX);
5. Буферный регистр (RG);
6. Шина данных (DB);
7. Шина адреса (AB);
8. Аналогово-цифровой преобразователь (ADC).

Приемники и исполнители:

1. Датчик скорости колес – аналоговый;
2. Гидравлический модулятор – цифровой;
3. Электронный блок управления – цифровой;
4. Тормозные цилиндры – аналоговый;

3.4 Информационная схема

Информационная схема представлена на рисунке 1.

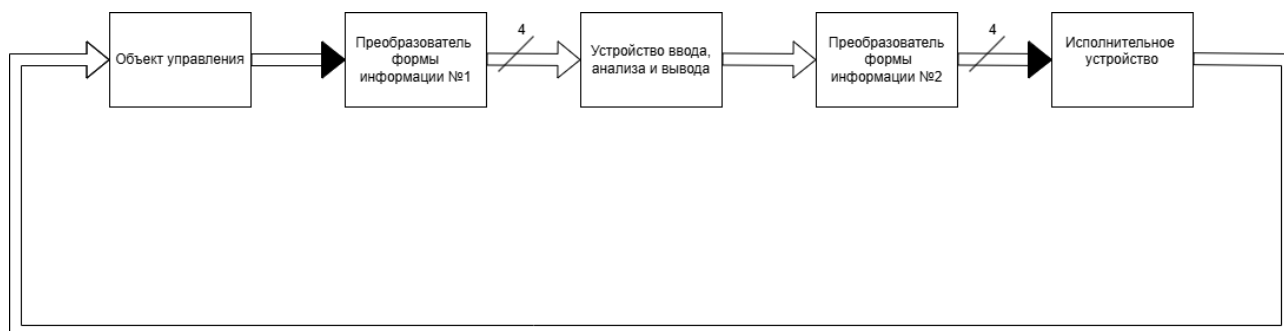


Рисунок 1 — Информационная схема

На рисунке 1 изображены:

1. Объект управления – автомобиль;
2. ПФИ №1 – Датчик скорости колес;
3. Устройство ввода анализа и вывода – Электронный блок управления;
4. ПФИ №2 – Гидравлический модулятор;
5. Блок исполнительных устройств – Рабочие тормозные цилиндры;

3.5 Электрическая блок-схема

Электрическая блок-схема представлена на рисунке 2.

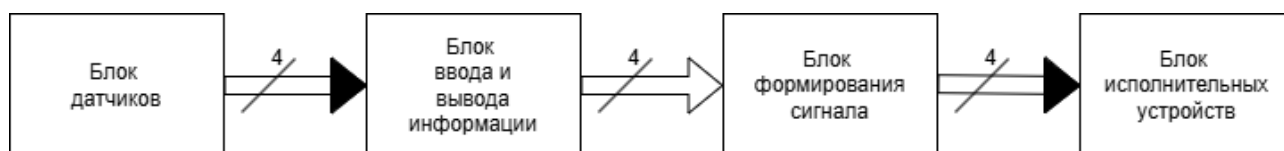


Рисунок 2 — Электрическая блок-схема

3.6 Функциональная схема

Функциональная схема представлена на рисунке 3.

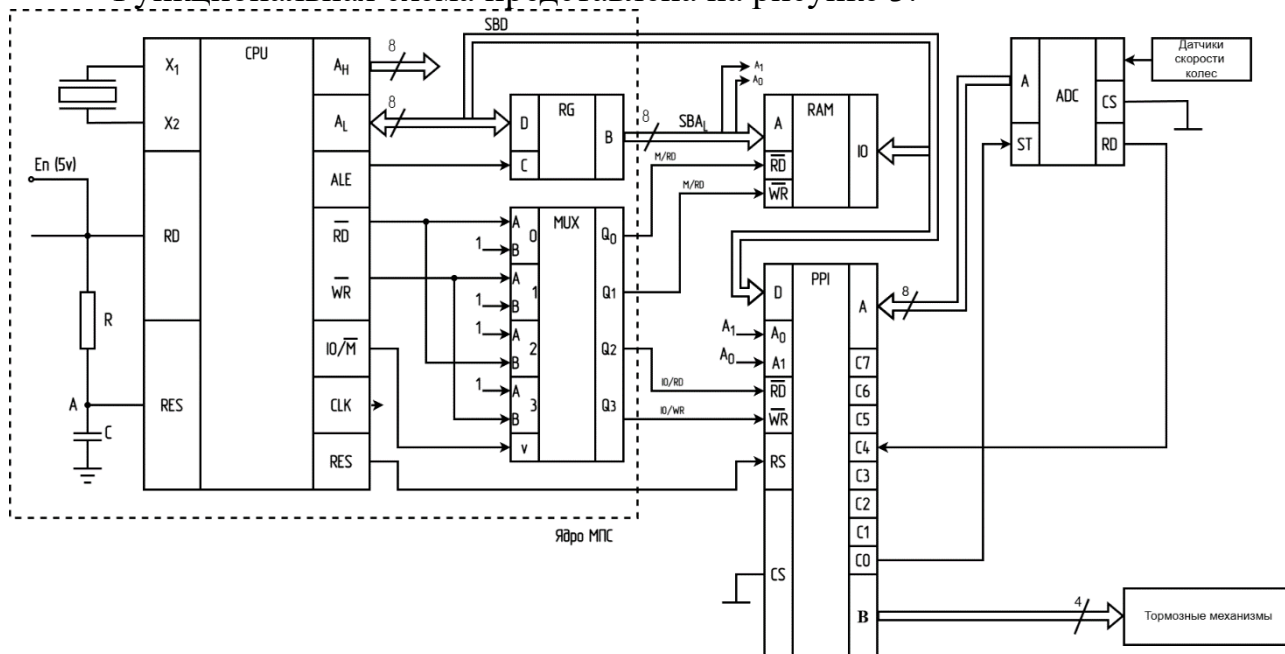


Рисунок 3 — Функциональная схема

3.7 Формирование адресов

Формат управляющего слова: «1», следовательно, старший бит УС = 1. Обе группы портов будут работать в режиме «однонаправленный ввод/вывод без квитирования», следовательно, биты 7, 6 и 3 = 0. Так как группа А запрограммирована на ввод, бит 5 = 1. Группа В запрограммирована на вывод, следовательно, биты 1 и 0 = 0. Таким образом, управляющее слово: 10010000 в двоичной системе счисления.

Адреса и выбранные порты представлены в таблице 1.

Таблица 1 – Адреса и выбранные порты

Регистр	Адрес	16-ый код	Устройство / Сигнал
A0-A7	01	-	Датчик скорость / In
C	02	02	Силовой двигатель / Out
B	03	01	Тормозные механизмы / Out

3.8 Алгоритм функционирования

Алгоритм основной программы представлен на рисунке 1.

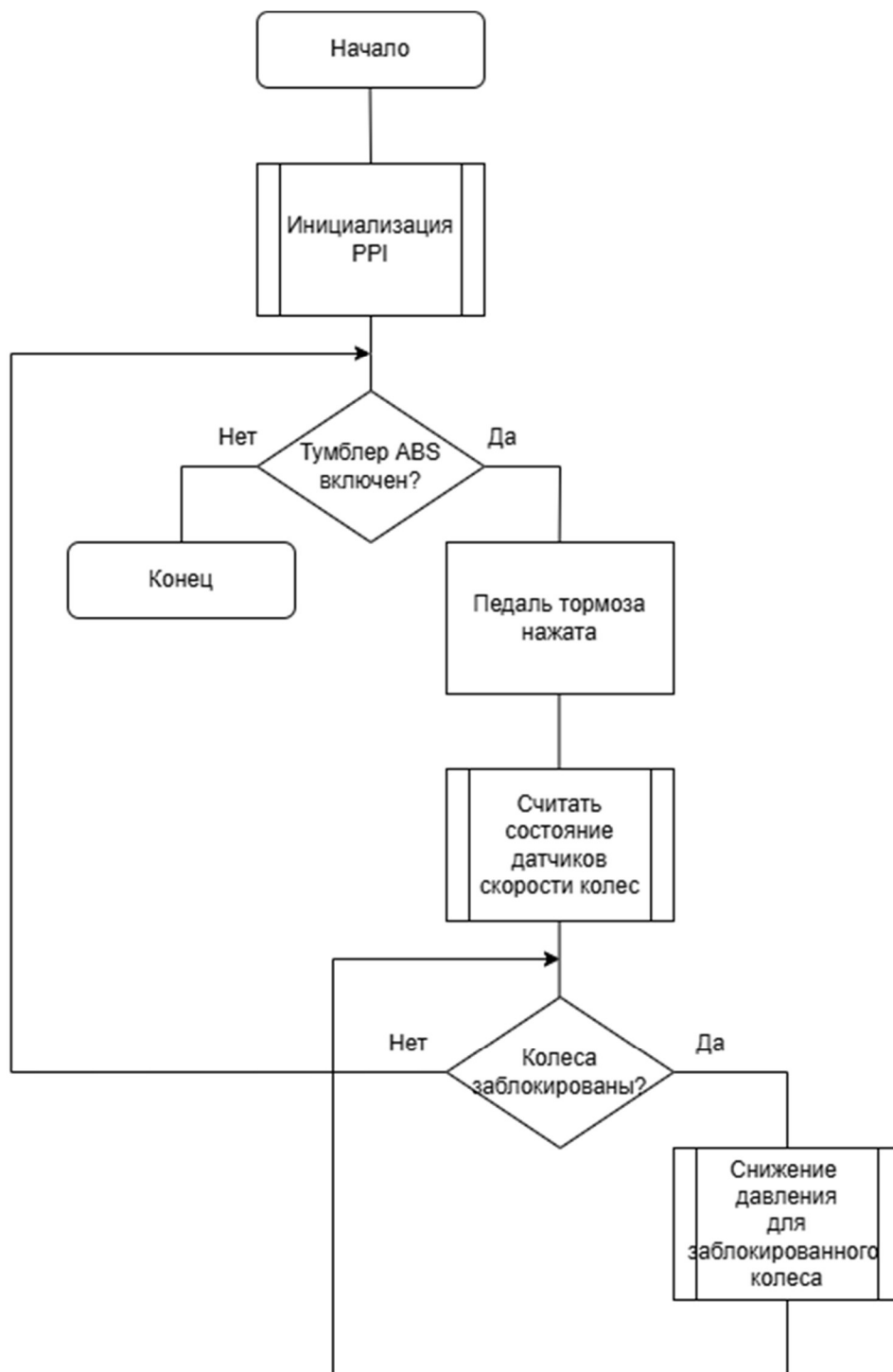


Рисунок 1 — Основной алгоритм

3.9 Листинг программного кода

Листинг программного кода представлен на рисунке.

1	; Адрес начала программы
2	.org 0x0000
3	
4	; Инициализация PPI (I8255A)
5	Init_PPI:
6	MVI A, 0b10010011 ; Control Word: Port A - input, Port B - output
7	OUT 0x03 ; Отправляем в регистр управления PPI
8	
9	Start:
10	; Ожидаем включения тумблера ABS
11	Wait_ABS_Tumbler: MVI A, 0x04 ; Выбираем Port C (вход)
12	OUT 0x03 ; Не обязательно, но может использоваться для
13	выборки IN 0x04 ; Считываем данные с Port C
14	CPI 0x01 ; Если тумблер ABS включен (предположим, что бит 0
15	= 1)
16	JNZ Wait_ABS_Tumbler ; Ждём, пока тумблер не будет включён
17	; Ожидаем нажатие педали тормоза
18	Wait_Brake:
19	MVI A, 0x04 ; Выбираем Port C (вход)
20	OUT 0x03 ; Не обязательно, но может использоваться для
21	выборки IN 0x04
22	; Считываем данные с Port C
23	CPI 0x01 ; Если педаль нажата (предположим, что бит 0 = 1)
24	JNZ Wait_Brake ; Ждём, пока педаль не будет нажата
25	
26	
27	
28	

```

29 Read_Sensors:
30     ; Читаем состояние датчиков скорости с Port A
31     IN 0x00                ; Считываем данные с Port A
32     MOV B, A               ; Сохраняем в регистре B
33
34     ; Предположим, что C содержит эталонное значение скорости
35     MVI C, 0x7F            ; Эталонная скорость (например, среднее значение)
36
37 Check_Lock:
38     MOV A, B
39     CMP C                  ; Сравниваем текущую скорость с эталонной
40     JC No_Block            ; Если текущая скорость больше эталона — нет
41     блокировки
42
43     ; Блокировка обнаружена — начинаем регулировку давления
44     JMP Regulate_Pressure
45
46 No_Block:
47     ; Нет блокировки — продолжаем обычное торможение
48     MVI A, 0x00            ; Отключаем регулировку давления
49     OUT 0x01               ; Отправляем на Port B
50     JMP Start              ; Возвращаемся к началу
51
52 Regulate_Pressure:
53     ; Цикл регулирования давления
54     ; Режимы:
55     ; 0x01 — снижение давления
56     ; 0x02 — удержание
57     ; 0x04 — повышение
58
59 Reduce_Pressure:
60     MVI A, 0x01            ; Команда на снижение давления
61     OUT 0x01               ; На Port B
62     CALL Delay             ; Задержка
63     CALL Read_Sensors      ; Проверяем состояние колёс снова
64     MOV B, A
65     CMP C
66     JC Exit_Regulation    ; Если скорость восстановилась — выходим
67
68 Hold_Pressure:
69     MVI A, 0x02            ; Команда на удержание давления
70     OUT 0x01
71     CALL Delay
72     JMP Check_Lock        ; Проверяем снова
73
74 Increase_Pressure:
75     MVI A, 0x04            ; Команда на увеличение давления
76     OUT 0x01
77     CALL Delay
78     JMP Reduce_Pressure   ; Продолжаем цикл
79
80 Exit_Regulation:
81     MVI A, 0x00            ; Отключаем регулировку
82     OUT 0x01
83     JMP Start
84
85 ; Подпрограмма чтения датчиков скорости
86 Read_Sensors:
87     IN 0x00                ; Считываем данные с Port A
88     RET
89
90 ; Простая подпрограмма задержки

```

91	Delay:
92	MVI D, 0xFF
93	Wait_D:
94	MVI E, 0xFF
95	Wait_E:
96	DCR E
97	JNZ Wait_E
98	DCR D
99	JNZ Wait_D
100	RET
101	
102	
103	
104	
105	
106	
107	
108	
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	