

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ И ДИЗАЙНА»

**Е. Г. Маежов, В. Ю. Иванов**

**Микропроцессорные устройства управления лифтов и эскалаторов**

Методические указания к выполнению контрольных работ

для студентов заочной формы обучения

направления подготовки 15.03.02 Технологические машины и оборудование

профиль подготовки Лифты и эскалаторы

Санкт-Петербург

2017

## Вопросы, подлежащие разработке

### I. Разработка МПСУ на ОМК

Составить структурную схему микропроцессорной системы автоматического управления (САУ).

Выбрать технические средства управления, определить их функции

Разработать принципиальную электрическую схему САУ.

Разработать обобщенный и подробный алгоритмы управления.

Разработать таблицу кодирования на основании таблицы истинности.

Разработать прикладную программу управления.

## Объем представляемого материала

Пояснительная записка содержит: задание, описание, обоснование, расчеты, выводы.

Графическая часть, содержит:

МПСУ на ОМК

- структурную схему микропроцессорной САУ;
- алгоритм;
- таблицу истинности;
- таблицу кодирования;
- текст прикладной программы управления;
- принципиальную электрическую схему микропроцессорной САУ;

## **ВВЕДЕНИЕ**

Однокристалльный микроконтроллер или однокристалльная микроЭВМ (ОМЭВМ) является эффективным средством автоматизации на основе применения методов цифровой обработки данных и управления разнообразными технологическими объектами и процессами.

Однокристалльный микроконтроллер конструктивно выполнен в виде одной большой интегральной схемы (БИС), включающей в себя устройства, необходимые для реализации цифровой системы контроля и управления минимальной конфигурации.

Специфическая организация ввода/вывода информации предопределяет область применения однокристалльных микроконтроллеров в качестве специализированных вычислительных устройств, включаемых в контур управления технологическим объектом или процессом. Структурная организация, набор команд и аппаратно-программные средства наиболее приспособлены для решения задач контроля и управления в системах автоматике.

Появление однокристалльных микроконтроллеров вызвало существенные изменения в подходах к разработке аппаратных и программных средств системы управления. В устройствах управления на основе однокристалльных микроконтроллеров аппаратные средства и программное обеспечение существуют в форме неделимого аппаратно-программного комплекса.

Результатом такой интеграции является уменьшение объема трудозатрат на разработку аппаратных средств по отношению к суммарным затратам на разработку и отладку прикладного программного обеспечения.

## Архитектура микроконтроллера

Микроконтроллер ATmega32 имеет следующие параметры:

- 120 инструкций с временем выполнения за 1 машинный такт;
- 32 Кбайт программируемой в системе FLASH-памяти программ (число циклов записи/стирания 1000);
- 2 кбайта энергонезависимой EEPROM-памяти данных (число циклов записи/стирания 100000);
- 512 байт статического ОЗУ (SRAM);
- 32x8 регистров общего назначения (РОН);
- 32 программируемых канала ввода/вывода;
- UART (полнодуплексный универсальный асинхронный приемопередатчик);
- SPI (последовательный синхронный интерфейс)
- полностью статическая архитектура, 0÷20 МГц;
- один 8-разрядный таймер/счетчик0 с предделителем;
- один 16-разрядный таймер/счетчик1 с предделителем, сравнением и захватом;
- двухканальный широтно-импульсный модулятор (ШИМ);
- система внутренних и внешних прерываний;
- программируемый сторожевой таймер (WDT);
- аналоговый компаратор.

Микроконтроллер имеет арифметико-логическое устройство АЛУ, которое выполняет операции над операндами, расположенными в регистровом файле (32 РОН). АЛУ выполняет три основные категории операций: арифметические, логические и битовые.

Система команд, включающая всю совокупность команд микроконтроллера, состоит из следующих групп: команды арифметических и логических операций; команды ветвления (передачи управления); команды пересылки данных; команды операций с битами.

## **Условное графическое обозначение (УГО) ОМК ATmega32**

### **Система команд микроконтроллера**

При создании программы разработчик оперирует программно доступными ресурсами микроконтроллера, к которым относятся программно доступные регистры (регистры общего назначения – РОН и регистры ввода/вывода - РВВ), внутренняя и внешняя память данных. Основными способами адресации являются: регистровая, непосредственная регистровая, прямая и косвенная. Микроконтроллер использует несколько разновидностей основных способов адресации.

Каждая команда сообщает процессору выполняемую операцию и методы доступа к операндам. Командная строка включает метку (символический адрес), мнемонику (символическое имя) команды, поле операндов и комментарий. Имя команды однозначно связано с выполняемой ею операцией.

### Арифметические и логические команды

Мнемоника	Описание	Операция	Операнды
ADD Rd,Rr	Сложение	$Rd \leftarrow Rd + Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
ADC Rd,Rr	Сложение с переносом	$Rd \leftarrow Rd + Rr + C$	$0 \leq d \leq 31, 0 \leq r \leq 31$
SUB Rd,Rr	Вычитание	$Rd \leftarrow Rd - Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
INC Rd	Инкремент	$Rd \leftarrow Rd + 1$	$0 \leq d \leq 31$
DEC Rd	Декремент	$Rd \leftarrow Rd - 1$	$0 \leq d \leq 31$
AND Rd,Rr	Конъюнкция	$Rd \leftarrow Rd \bullet Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
ANDI Rd,K	Конъюнкция с операндом	$Rd \leftarrow Rd \bullet K$	$0 \leq d \leq 31$
OR Rd,Rr	Дизъюнкция	$Rd \leftarrow Rd \vee Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
COM Rd	Инверсия	$Rd \leftarrow \$FF - Rd$	$0 \leq d \leq 31$
EOR Rd,Rr	Исключающее ИЛИ	$Rd \leftarrow Rd \oplus Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
CLR Rd	Очистить регистр	$Rd \leftarrow Rd \oplus Rd$	$0 \leq d \leq 31$
SER Rd	Установить регистр	$Rd \leftarrow \$FF$	$16 \leq d \leq 31$

### Команды ветвления (переходов, передачи управления)

Мнемоника	Описание	Операция	Операнды
RJMP k	Безусловный переход		$-2K \leq k \leq +2K$
RCALL k	Вызов подпрограммы		$-2K \leq k \leq +2K$
RET	Возврат из подпрограммы		
BREQ k	Переход, если РАВНО		$-64 \leq k \leq +63$
BRNE k	Переход, если НЕ РАВНО		$-64 \leq k \leq +63$
SBRC Rr,b	Пропустить, если бит в РОН сброшен		$0 \leq r \leq 31, 0 \leq b \leq 7$
SBRS Rr,b	Пропустить, если бит в РОН установлен		$0 \leq r \leq 31, 0 \leq b \leq 7$
SBIC A,b	Пропустить, если бит в РВВ сброшен		$0 \leq A \leq 31, 0 \leq b \leq 7$
SBIS A,b	Пропустить, если бит в РВВ установлен		$0 \leq A \leq 31, 0 \leq b \leq 7$

### Команды пересылки данных

Мнемоника	Описание	Операция	Операнды
LDI Rd,K	Загрузка константы (байта)	$Rd \leftarrow K$	$16 \leq d \leq 31$
MOV Rd,Rr	Пересылка данных (байта)	$Rd \leftarrow Rr$	$0 \leq d \leq 31, 0 \leq r \leq 31$
IN Rd,A	Пересылка из PVB в POH	$Rd \leftarrow A$	$0 \leq d \leq 31, 0 \leq A \leq 63$
OUT A,Rr	Пересылка из POH в PVB	$A \leftarrow Rd$	$0 \leq d \leq 31, 0 \leq A \leq 63$

### Команды операций с битами

Мнемоника	Описание	Операция	Операнды
CBR Rd,K	Сброс битов POH	$Rd \leftarrow Rd \bullet (\$FF-K)$	$16 \leq d \leq 31, 0 \leq K \leq 255$
SBR Rd,K	Установка битов POH	$Rd \leftarrow Rd \vee K$	$16 \leq d \leq 31, 0 \leq K \leq 255$
CBI A,b	Сброс битов PVB		$0 \leq A \leq 31, 0 \leq b \leq 7$
SBI A,b	Установка битов PVB		$0 \leq A \leq 31, 0 \leq b \leq 7$

Способы адресации представляют собой набор механизмов доступа к операндам. Одни из них просты и поэтому приводят к компактному формату команды и быстрому доступу к операнду, но объем доступных с их помощью ресурсов ограничен. Другие способы адресации позволяют оперировать со всеми имеющимися в системе ресурсами, но команда получается длинной, на ее ввод и выполнение тратится много времени.



## Порядок выполнения работы

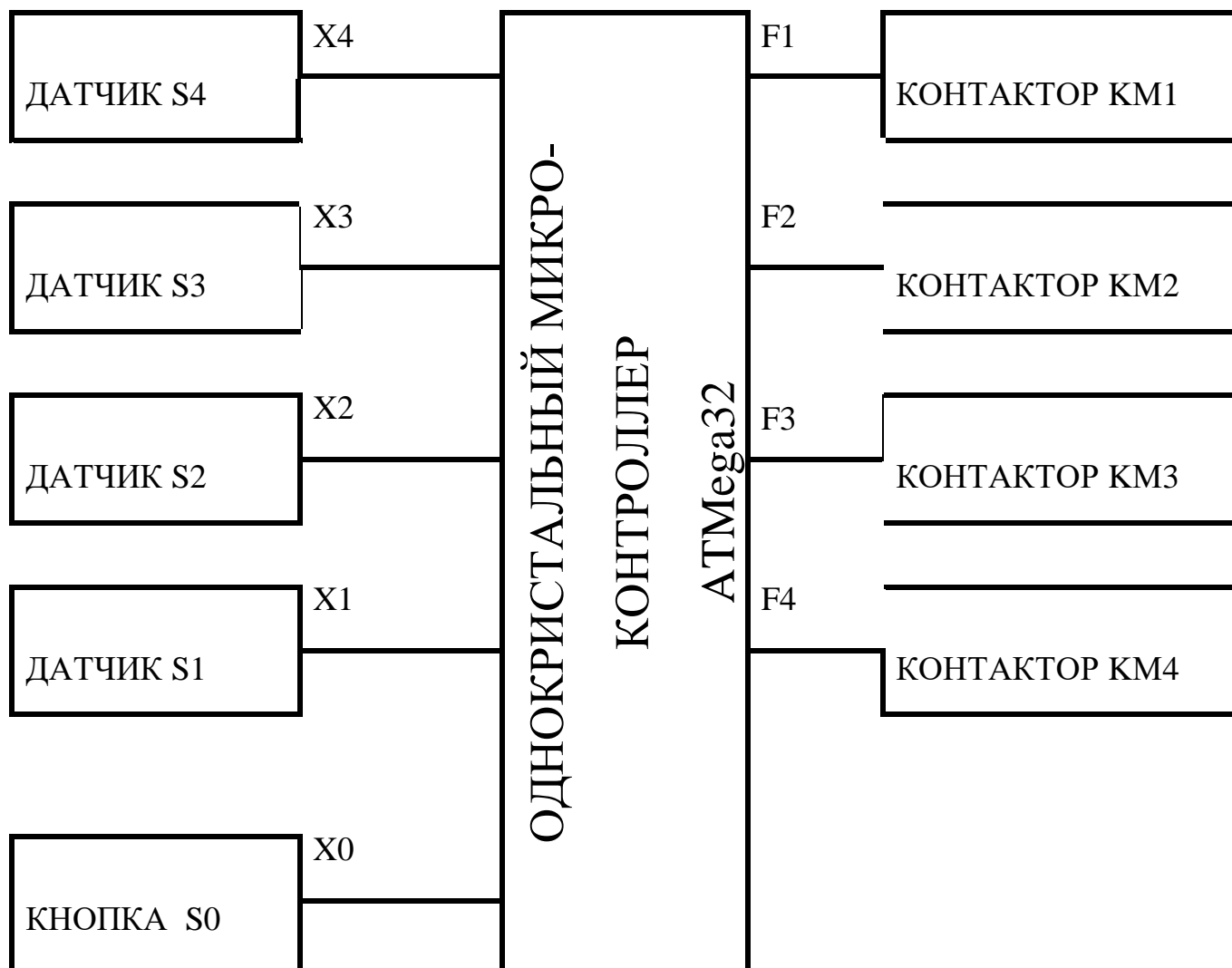
### 1. Разработка структурной схемы системы

Анализ задания на проектирование из таблицы 1 исходных данных

Таблица 2 Датчики, микроконтроллер и контакторы

Название	Кол-во	Обозначение
Дискретные датчики	4	S4 S3 S2 S1
Кнопка управления вводом	1	S0
Контакторы	4	W1 W2 W3 W4
Микроконтроллер	1	ATMega32

Составим структурную схему системы, в соответствии с заданием на проектирование:



### **Описание работы схемы:**

- однокристалльный микроконтроллер получает сигналы от датчиков S4, S3, S2, S1 по команде кнопки управления вводом S0;
- микроконтроллер обрабатывает полученную информацию в соответствии с таблицей истинности и формирует управляющие воздействия F1, F2, F3, F4;
- микроконтроллер выводит F1, F2, F3, F4 на контакторы К1, К2, К3, К4;
- контакторы КМ1, КМ2, КМ3, КМ4 управляют включением и выключением исполнительных электродвигателей М1, М2, М3, М4 (на схеме не показаны).

### **Обобщенный алгоритм:**

- I. Инициализация микроконтроллера.
- II. Проверка нажатия кнопки S0.
- III. Ввод данного от датчиков S4 S3 S2 S1.
- IV. Обработка входного данного.
- V. Формирование управляющего воздействия.
- VI. Обработка управляющего воздействия.
- VII. Вывод управляющего воздействия на исполнительное устройство.
- VIII. Переход к пункту 2 (защелкивание программы).

### **Разработка структурной схемы подключения датчиков, кнопки управления и контакторов**

Анализ задания на проектирование из таблицы 1 исходных данных

Порт ввода данных	D
Линии ввода данных от датчиков	PD0, PD1, PD2, PD3
Линия управления вводом	PD4
Порт вывода управляющего воздействия	PC1, PC2, PC3, PC4,
Линия вывода на исполнительное устройство	4321

Таблица 3

Ввод данных в микроконтроллер

Устройства ввода	Датчики				Кнопка
	S4	S3	S2	S1	S0
Сигналы	X4	X3	X2	X1	X0
Линии ввода	PD0	PD1	PD2	PD3	PD4

Таблица 4

Вывод данных в микроконтроллера

Линии вывода	PC4	PC3	PC2	PC1
Сигналы	F1	F2	F3	F4
Контакторы	K1	K2	K3	K4
Исполнительные устройства	W1	W2	W3	W4

Составим структурную схему подключения датчиков, кнопки управления и контакторов системы в соответствии с заданием:

		1	PB0	<b>Однокристалльный микроконтроллер ATMega32</b>	PA0	1			
		2	PB1		PA1	2			
		4	PB2		PA2	4			
		8	PB3		PA3	8			
		16	PB4		PA4	16			
		32	PB5		PA5	32			
		64	PB6		PA6	64			
		128	PB7		PA7	128			
Датчик S4	X4	1	PD0		PC7	128			
Датчик S3	X3	2	PD1		PC6	64			
Датчик S2	X2	4	PD2		PC5	32			
Датчик S1	X1	8	PD3		PC4	16	F1	KM1	M1
Кнопка S0	X0	16	PD4		PC3	8	F2	KM2	M2
		32	PD5		PC2	4	F3	KM3	M3
		64	PD6		PC1	2	F4	KM4	M4
		128	PD7		PC0	1			
Сигнал	Вес	Порт		Порт	Вес	Сигнал			

## **2. Разработка электрической принципиальной схемы**

Электрическая принципиальная схема  
микропроцессорной системы управления

Дискретная система автоматического управления обеспечивает управление электроприводами исполнительных механизмов и устройств. Управление осуществляется по сигналам четырех датчиков, имеющих по одному размыкающему контакту. Синхронизация ввода от датчиков обеспечивается кнопкой S0, с размыкающим контактом.

В качестве управляющего устройства используется однокристальный микроконтроллер ATmega32.

Дискретные датчики подключены к порту D. Исполнительные устройства подключены к порту C.

Для обеспечения сопряжения контакторов КМ1, КМ2, КМ3, КМ4 с портом C по нагрузочной способности используются промежуточные усилители, выполненные на базе оптосимисторов СИ1, СИ2, СИ3, СИ4.

Контакторы КМ1, КМ2, КМ3, КМ4 обеспечивают включение и выключение исполнительных трехфазных электроприводов М1, М2, М3, М4.

Подача трехфазного напряжения на силовую цепь обеспечивается автоматическими выключателями QF1, QF2, QF3, QF.

В схеме предусмотрена защита от перегрузок электродвигателя - с помощью тепловых реле КК1, КК2, КК3, КК4, КК5 и КК6, включенных в две фазы каждого двигателя. Размыкающие контакты КК1, КК2, КК3, КК4, КК5 и КК6 этих реле при перегрузках размыкают цепь питания контакторов КМ1, КМ2, КМ3, КМ4, тем самым отключая двигатели от сети.

При появлении логических нулей на линиях порта C открываются оптосимисторы СИ1, СИ2, СИ3, СИ4, срабатывают контакторы КМ1, КМ2, КМ3, КМ4 и своими контактами КМ1, КМ2, КМ3, КМ4 включают соответствующие двигатели М1, М2, М3, М4.

Оптосимистор состоит из светодиода и симистора, обеспечивающих гальваническую развязку низковольтных (микроконтроллер) и высоковольтных (электродвигатель) устройств и электрических цепей.

Для обеспечения рабочего тока светодиодов оптосимисторов СИ1, СИ2, СИ3, СИ4 используются резисторы R1, R2, R3, R4. Номинал сопротивления для активного выходного нулевого уровня управляющего воздействия рассчитывается по формуле:

$$R^0 = \frac{U_{\text{ПИТ}} - U_{\text{ВД}}^{\text{ПР}} - U_{\text{ВЫХ}}^0}{I_{\text{ВД}}^{\text{ПР}}},$$

где  $U_{\text{ПИТ}}$  – напряжение питания равное +5 В;  $U_{\text{ВД}}^{\text{ПР}}$  - прямое падение напряжения на светодиоде равное 2,8 В;  $U_{\text{ВЫХ}}^0$  - выходное напряжение логического нуля (от 0 В до 0,4 В);  $I_{\text{ВД}}^{\text{ПР}}$  - постоянный прямой ток светодиода равный 10 мА.

$U_{\text{ПИТ}}, U_{\text{ВД}}^{\text{ПР}}, U_{\text{ВЫХ}}^0, I_{\text{ВД}}^{\text{ПР}}$  - паспортные значения.

$$R^0 = \frac{5 - 2,8 - 0,1}{10 \cdot 10^{-3}} = 210 \text{ Ом.}$$

Проверяем наличие указанного номинала по шкале номинальных сопротивлений для резисторов общего применения по ряду Е6, Е12 и Е24, приведенных в таблице «Номинальные сопротивления по ряду Е6, Е12 и Е24».

Таблица Номинальные сопротивления по ряду Е6, Е12 и Е24

Индекс ряда	Числовые коэффициенты, Ом					
<i>Е6</i>	1,0	1,5	2,4	3,3	4,7	6,8
<i>Е12</i>	1,0	1,5	2,4	3,3	4,7	6,8
	1,2	1,8	2,7	3,9	5,6	8,1
<i>Е24</i>	1,0	1,5	2,2	3,3	4,7	6,8
	1,1	1,6	2,4	3,6	5,1	7,5
	1,2	1,8	2,7	3,9	5,6	8,1
	1,3	2,0	3,0	4,3	6,2	9,1

Примечание. Для получения номинала резистора каждый коэффициент может быть умножен на  $10^n$ , где индекс n изменяется от -2 до +6.

Расчетное значение  $R_0 = 555 \text{ Ом}$ . Из таблицы 1 выбираем ближайшее большее значение номинала для  $R_0$  равное  $2,4 \cdot 10^2 \text{ Ом}$ .

Получаем  $R_0 = 240 \text{ Ом}$ .

К параметрам резистора также относится номинальная мощность рассеяния  $P_n$  - это наибольшая мощность, которую резистор может рассеивать в течение га-

рантированного срока службы при сохранении параметров в установленных пределах. Значение  $P_n$  зависит от конструкции резистора, физических свойств материалов и температуры окружающей среды.

Рассчитаем мощность рассеяния  $P_o$  резистора  $R_o$  по формуле:

$$P_o = (I_o)^2 \cdot R_o = (10 \cdot 10^{-3})^2 \cdot 240 = 0,024 \text{ Вт.}$$

Номинальную мощность рассеяния  $P_{он}$  резистора (ближайшее большее значение) выбираем из таблицы «Значения номинальной мощности рассеяния».

Таблица      Значения номинальной мощности рассеяния

Номинальная мощность рассеяния, Вт											
0,01	0,025	0,05	0,062	0,125	0,25	0,5	1	2	3	4	5
8	10	16	25	40	63	80	100	160	250	500	-

Выбираем номинальную мощность  $P_n$  резистора. Расчетное значение мощности  $P_{он} = 0,024$  Вт.

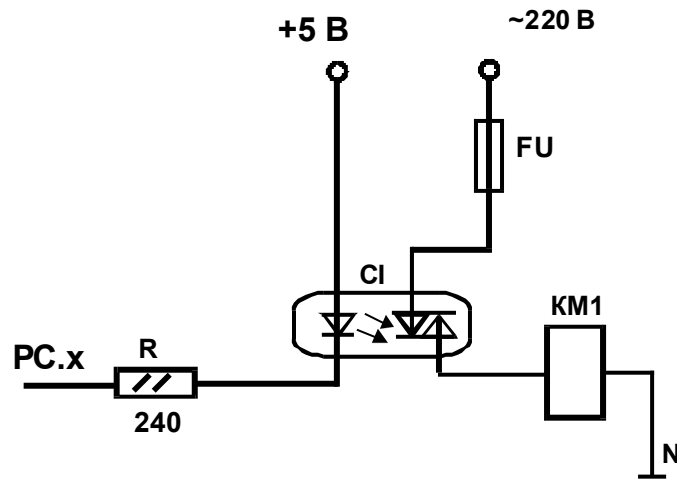
Выбираем из таблицы ближайшее большее значение  $P_n = 0,025$  Вт.

Выбираем резистор из справочника, расположенного в ПРИЛОЖЕНИИ 1 «Резисторы постоянные»: резистор типа С2 - 33Н (замена МЛТ).

В справочнике минимальная мощность резистора типа С2 - 33Н равна 0,125 Вт, поэтому корректируем значение мощности с 0,025 Вт на 0,125 Вт.

Выбираем резистор С2 - 33Н - 0,125 – 240 ± 5% - А - Д – ОЖО.467.093 ТУ.

На электрической принципиальной схеме вносим коррекцию изображения резистора с учетом выбранной мощности резистора как показано на рисунке.



Для питания микроконтроллера используется стабилизированный источник питания на +5 В, включающий в себя:

- понижающий трансформатор напряжения ТУ (~220/~9 В);
- однофазный мостовой двухполупериодный выпрямитель на выпрямительных диодах VD1, VD2, VD3, VD4;
- сглаживающие фильтры на конденсаторах C1, C2, C3, C4;
- интегральный стабилизатор напряжения КР142ЕН5А.

### 3. Разработка прикладной программы

#### Обработка входного данного

Обработка входных данных основана на структурной схеме подключения датчиков. Обработка входного данного включает следующие операции:

- маскирование входного данного;
- сортировка входного данного;
- сборка входного данного.

#### Маскирование входного данного

Операция маскирования используется для:

- выделения значащих разрядов данного (X1 X2 X3 X4), записанного с входов порта D в регистр микроконтроллера;



- обнуления не значащих разрядов (сигнала X0 кнопки S0 и остальных битов).

В таблице 5 показано содержимое регистра in\_data после ввода данных из порта.

Таблица 5 Содержимое регистра после ввода

Датчики				S0	S1	S2	S3	S4
Порт	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Вес	128	64	32	16	8	4	2	1
Регистр in_data	~	~	~	X0	X1	X2	X3	X4

Примечание. ~ - произвольное значение бита.

В таблице 6 показано, как должны располагаться биты введенного данного в регистре после маскирования.

Таблица 6 Содержимое регистра после маскирования

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	X1	X2	X3	X4

Маскирование (таблица 6) выполняется с помощью логической операции И между содержимым регистра и маской команда ANDI (andi in\_data,0b00001111).

Однако расположение битов таблицы 6 не соответствует весам 8-4-2-1 т.к. датчики расположены в произвольном порядке. Поэтому необходимо выполнить дополнительные маскирования и масштабирования для каждого разряда в отдельности. В таблице 7 показано, как должны располагаться биты введенного данного в регистре после сортировки и сборки, в соответствии заданными весами.

Таблица 7 Содержимое регистра после сортировки и сборки

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	X4	X3	X2	X1

Разработка маски для входного данного показана в таблице 8.

Таблица 8 Разработка маски

Вес	128	64	32	16	8	4	2	1
Регистр	~	~	~	X0	X1	X2	X3	X4
Маска	0	0	0	0	1	1	1	1

В таблице 9 показан результат маскирования.

Таблица 9 Результат маскирования

Вес	128	64	32	16	8	4	2	1
Регистр in_data	~	~	~	X0	X1	X2	X3	X4
Операция И	&	&	&	&	&	&	&	&
Маска	0	0	0	0	1	1	1	1
Регистр in_data	0	0	0	0	X1	X2	X3	X4

### Сортировка входного данного

Сортировка входного данного включает следующие операции:

- маскирование одного бита входного данного;
- масштабирование выделенного бита входного данного.

Операция маскирования бита входного данного используется для:

- выделения одного значащего разряда данного из данного X1 X2 X3 X4;
- обнуления остальных битов.

Операция масштабирования выделенного бита используется для перемещения выделенного бита в сторону своего веса. Масштабирование выполняется командами логического сдвига влево LSL или логического сдвига вправо LSR.

## Сортировка X1

Перед сортировкой необходимо скопировать содержимое исходного регистра IN\_DATA в регистр X1 с помощью команды MOV (mov X1, in\_data). А для замыкающего датчика перед сортировкой предварительно сделать инверсию регистра X1 командой COM (com X1).

В таблице 9 приведена сортировка разряда X1.

Таблица 10 Сортировка разряда X1

Вес	128	64	32	16	8	4	2	1
Регистр X1	0	0	0	0	X1	X2	X3	X4
Операция И	&	&	&	&	&	&	&	&
Маска	0	0	0	0	1	0	0	0
Результат	0	0	0	0	X1	0	0	0
Сдвиг вправо	0	0	0	0	0	X1	0	0
Сдвиг вправо	0	0	0	0	0	0	X1	0
Сдвиг вправо	0	0	0	0	0	0	0	X1
Регистр X1	0	0	0	0	0	0	0	X1

Примечание. Один сдвиг вправо выполняется командой LSR (lsr X1).

## Сортировка X2

Перед сортировкой необходимо скопировать содержимое исходного регистра IN\_DATA в регистр X2 с помощью команды MOV (mov X2, in\_data). А для замыкающего датчика перед сортировкой дополнительно сделать инверсию регистра X2 командой COM (com X2).

В таблице 11 приведена сортировка разряда X2.

Таблица 11 Сортировка разряда X2

Вес	128	64	32	16	8	4	2	1
Регистр X2	0	0	0	0	X1	X2	X3	X4
Операция И	&	&	&	&	&	&	&	&
Маска	0	0	0	0	0	1	0	0
Результат	0	0	0	0	0	X2	0	0
Сдвиг вправо	0	0	0	0	0	0	X2	0
Регистр X2	0	0	0	0	0	0	X2	0

Примечание. Один сдвиг вправо выполняется командой LSR (lsr X1).

### Сортировка X3

Перед сортировкой необходимо скопировать содержимое исходного регистра IN\_DATA в регистр X3 с помощью команды MOV (mov X3, in\_data). А для замыкающего датчика перед сортировкой дополнительно сделать инверсию регистра X3 командой COM (com X3).

Таблица 12 Сортировка разряда X3

Вес	128	64	32	16	8	4	2	1
Регистр X3	0	0	0	0	X1	X2	X3	X4
Операция И	&	&	&	&	&	&	&	&
Маска	0	0	0	0	0	0	1	0
Результат	0	0	0	0	0	0	X3	0
Сдвиг влево	0	0	0	0	0	X3	0	0
Регистр X3	0	0	0	0	0	X3	0	0

Примечание. Один сдвиг влево выполняется командой LSL (lsl X1).

### Сортировка X4

Перед сортировкой необходимо скопировать содержимое исходного регистра in\_data в регистр X4 с помощью команды MOV (mov X4, in\_data). А для замыкающего датчика перед сортировкой дополнительно сделать инверсию регистра X4 командой COM (com X4).

В таблице 13 приведена сортировка разряда X4.

Таблица 13 Сортировка разряда X4

Вес	128	64	32	16	8	4	2	1
Регистр X4	0	0	0	0	X1	X2	X3	X4
Операция И	&	&	&	&	&	&	&	&
Маска	0	0	0	0	0	0	0	1
Результат	0	0	0	0	0	0	0	X4
Сдвиг влево	0	0	0	0	0	0	X4	0
Сдвиг влево	0	0	0	0	0	X4	0	0
Сдвиг влево	0	0	0	0	X4	0	0	0
Регистр X4	0	0	0	0	X4	0	0	0

Примечание. Один сдвиг влево выполняется командой LSR (lsl X1).

### Сборка регистра входного данного

После сортировки каждый бит находится в отдельном регистре X1, X2, X3, X4. Поэтому необходимо выполнить сборку всех битов в один регистр (например обратно в регистр in\_data, т.к. его содержимое уже не имеет значения).

Перед сборкой необходимо обнулить регистр in\_data командой CLR (clr in\_data) или LDI (ldi in\_data,0).

Сборка бита X1 приведена в таблице 14.

Таблица 14 Сборка разряда X1

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	0	0	0	0
Операция ИЛИ	V	V	V	V	V	V	V	V
Регистр X1	0	0	0	0	0	0	0	X1
Регистр in_data	0	0	0	0	0	0	0	X1

Сборка выполняется командой операции ИЛИ OR (or in\_data,X1).

Сборка бита X2 приведена в таблице 15.

Таблица 15 Сборка разряда X2

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	0	0	0	X1
Операция ИЛИ	V	V	V	V	V	V	V	V
Регистр X2	0	0	0	0	0	0	X2	0
Регистр in_data	0	0	0	0	0	0	X2	X1

Сборка выполняется командой операции ИЛИ OR (or in\_data,X2).

Сборка бита X3 приведена в таблице 16.

Таблица 15 Сборка разряда X3

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	0	0	X2	X1
Операция ИЛИ	V	V	V	V	V	V	V	V
Регистр X3	0	0	0	0	0	X3	0	0
Регистр in_data	0	0	0	0	0	X3	X2	X1

Сборка выполняется командой операции ИЛИ OR (or in\_data,X3).

Сборка бита X4 приведена в таблице 17.

Таблица 17 Сборка разряда X4

Вес	128	64	32	16	8	4	2	1
Регистр in_data	0	0	0	0	0	X3	X2	X1
Операция ИЛИ	V	V	V	V	V	V	V	V
Регистр X4	0	0	0	0	X4	0	0	0
Регистр in_data	0	0	0	0	X4	X3	X2	X1

Сборка выполняется командой операции ИЛИ OR (or in\_data,X4).

После сборки всех разрядов содержимое регистра in\_data соответствует значению 0000X4X3X2X1, приведенному в таблице 7.

### Разработка таблицы кодирования

Анализ задания на проектирование из таблицы 1 исходных данных

Таблица истинности	Вариант 0
--------------------	-----------

Задание дано в виде сокращенной записи СДНФ:

$$F1 = \bigvee_i^4, \text{ при } i = 0, 1, 2, 7, 8, 9, 11, 13, 14$$

$$F2 = \bigvee_i^4, \text{ при } i = 0, 1, 2, 3, 8, 10, 12, 13, 15$$

$$F3 = \bigvee_i^4, \text{ при } i = 1, 2, 3, 4, 9, 11, 12, 14, 15$$

$$F4 = \bigvee_i^4, \text{ при } i = 0, 2, 4, 5, 10, 11, 13, 14, 15$$

Составляем таблицу истинности

Таблица 18 Задание функций F1, F2, F3, F4

№	АРГУМЕНТ				ФУНКЦИЯ			
	Вариант 0							
	8	4	2	1	8	4	2	1
	X4	X3	X2	X1	F1	F2	F3	F4
0	0	0	0	0	1	1	0	1
1	0	0	0	1	1	1	1	0
2	0	0	1	0	1	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	0	1	1
5	0	1	0	1	0	0	0	1
6	0	1	1	0	0	0	0	0
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	0	1	0
10	1	0	1	0	0	1	0	1
11	1	0	1	1	1	0	1	1
12	1	1	0	0	0	1	1	0
13	1	1	0	1	1	1	0	1
14	1	1	1	0	1	0	1	1
15	1	1	1	1	0	1	1	1

Определим веса функции F как сумму весов функций F1 F2 F3 F4 и запишем их в кодовую таблицу 19.

Таблица 19 Кодовая таблица

ФУНКЦИЯ				КОД	
8	4	2	1	F, dec	F, hex
F1	F2	F3	F4		
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F
0	1	1	0	6	6
0	0	1	1	3	3
0	0	0	1	1	1
0	0	0	0	0	0
1	0	0	0	8	8
1	1	0	0	12	C

1	0	1	0	10	A
0	1	0	1	5	5
1	0	1	1	11	B
0	1	1	0	6	6
1	1	0	1	13	D
1	0	1	1	11	B

Таблица 20

Таблица состояний TABL

№	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F	0D	0E	0F	06	03	01	00	08	0C	0A	05	0B	06	0D	0B	07

Составим программную таблицу TABL:

TABL:

.DB \$0D, \$0E, \$0F, \$06, \$03, \$01, \$00, \$08, \$0C, \$0A, \$05, \$0B, \$06, \$0D, \$0B, \$07

### Порядок нахождения табличного значения выходного данного

Таблица истинности TABL находится в программной памяти вместе с прикладной программой. При трансляции прикладной программы таблица распределяется в адресном пространстве. Пусть нулевому набору таблицы истинности присваивается базовый (нулевой) адрес ADDR0. Тогда вся таблица будет размещена с адреса ADDR0 по ADDR0F. После определения базового (нулевого) адреса таблицы истинности ADDR0 он записывается в шестнадцатиразрядный регистр Z (R31:R30).

Текущий адрес требуемого набора таблицы TABL определяется как сумма базового адреса ADDR0 и содержимого регистра in\_data. Таким образом искомый адрес ADDR<sub>x</sub> определяется:  $ADDR_x = ADDR0 + in\_data$  с помощью команды ADD (add ZL,in\_data).

В таблице показано распределение таблицы истинности в адресном пространстве программной памяти микроконтроллера.



Таблица 21

Распределение таблицы в памяти программ

Номер набора	Адрес	Содержимое таблицы	Определение адреса набора таблицы	Содержимое регистра in_data
0	ADDR0	0D	$ADDR0 = ADDR0 + 0000$	0D
1	ADDR1	0E	$ADDR1 = ADDR0 + 0001$	0E
2	ADDR2	0F	$ADDR2 = ADDR0 + 0010$	0F
3	ADDR3	06	$ADDR3 = ADDR0 + 0011$	06
4	ADDR4	03	$ADDR4 = ADDR0 + 0100$	03
5	ADDR5	01	$ADDR5 = ADDR0 + 0101$	01
6	ADDR6	00	$ADDR6 = ADDR0 + 0110$	00
7	ADDR7	08	$ADDR7 = ADDR0 + 0111$	08
8	ADDR8	0C	$ADDR8 = ADDR0 + 1000$	0C
9	ADDR9	0A	$ADDR9 = ADDR0 + 1001$	0A
10	ADDRA	05	$ADDRA = ADDR0 + 1010$	05
11	ADDRB	0B	$ADDRB = ADDR0 + 1011$	0B
12	ADDRC	06	$ADDRC = ADDR0 + 1100$	06
13	ADDRD	0D	$ADDRD = ADDR0 + 1101$	0D
14	ADDRE	0B	$ADDRE = ADDR0 + 1110$	0B
15	ADDRF	07	$ADDRF = ADDR0 + 1111$	07

Выходное данное определяется с помощью команды LPM (lpm out\_data,Z).

### Порядок обработки выходного данного

После взятия выходного данного из таблицы истинности необходимо произвести его обработку. Обработка выходных данных основана на структурной схеме подключения датчиков. В таблице 18 показан порядок обработки входных данных, включающий следующие операции:

- масштабирование;
- маскирование.



- II. Проверка нажатия кнопки S0, формирующей сигнал X0 разрешения ввода (квитирования), поступающий на линию порта ввода PIND.4:
  - II.1. Если кнопки S0 не нажата, то перейти к пункту II.
  - II.2. Если кнопки S0 нажата, то перейти к пункту III.
- III. Ввод данных от датчиков S4 S3 S2 S1.
- IV. Обработка входного данного.
  - IV.1. Маскирование входного данного.
  - IV.2. Сортировка X1.
  - IV.3. Сортировка X2.
  - IV.4. Сортировка X3.
  - IV.5. Сортировка X4.
  - IV.6. Сборка данного X4 X3 X2 X1.
- V. Формирование управляющего воздействия.
  - V.1. Определение базового адреса таблицы истинности.
  - V.2. Определение заданного адреса таблицы истинности.
  - V.2. Взятие из таблицы истинности выходного данного (управляющего воздействия).
- VI. Обработка выходного данного.
  - VI.1. Масштабирование выходного данного.
  - VI.2. Маскирование выходного данного.
- VII. Вывод управляющего воздействия на исполнительное устройство.
- VIII. Переход к пункту 2 (защелкивание программы).

;ПРИКЛАДНАЯ ПРОГРАММА "Вариант 0"

;Микропроцессорная система управления

;студент \_\_\_\_\_

;=====

.include "m32def.inc"

;=====

;Определение регистров данных

.def tmp=r16 ;Временный регистр

.def data=r17 ;Входные данные X3 X2 X1 X0

```

.def tmpL=r18 ;Временный регистр (младший байт)
.def tmpH=r19 ;Временный регистр (старший байт)
.def control=r20 ;Выходные данные F1 F2 F3 F4
;Входные данные X3 X2 X1 X0 получаем с датчиков.
;Выходные данные F1 F2 F3 F4 получаем из таблицы TABL
;по показаниям датчиков.
;Выходные данные это управляющие воздействия,
;поступающие на исполнительные устройства.
;=====
;ИНИЦИАЛИЗАЦИЯ Регистров Ввода-Вывода
;=====
;Указатель стека
ldi tmp,high(RAMEND)
out SPH,tmp
ldi tmp,low(RAMEND)
out SPL,tmp
;=====
;Настройка PORTD на вход
ldi tmp,$00
out DDRD,tmp
ldi tmp,$FF
out PORTD,tmp
;=====
;Настройка PORTC на выход
ldi tmp,$FF
out DDRC,tmp
;=====
;*****
;Главная программа
MAIN:
;-----
;Проверка сигнала X4 (разрешение ввода)

```

```

WAIT: sbis PIND,0
rjmp WAIT ;X4=0
nop ;X4=1
;-----
;Ввод данного от датчиков X3, X2, X1, X0
; Линии ввода данных порта C: 4321
in data,PIND
;-----
;Программный сброс X4
;Применяется только при отладке программы
cbi PIND,0
;-----
;Маскирование данного, полученного по
; Линии ввода данных порта C: 4321
andi data,0b00011110
;-----
;Масштабирование данного, полученного по
;линии ввода данных порта C: 4321
lsr data
;-----
;Определение адреса базового (нулевого) набора таблицы
ldi ZH, high(TABL*2)
ldi ZL, low(TABL*2)
;-----
;Расчет адреса набора таблицы по показаниям датчиков
ldi tmpH, $00
add ZL, data
adc ZH, tmpH
;-----
;Чтение выходной переменной из таблицы
lpm control, Z
;-----

```

;Масштабирование выходной переменной

lsl control

;-----

;Маскирование выходной переменной

andi control, 0b00011110 ;Бинарная маска

;-----

;Вывод управляющего воздействия на двигатели

out PORTC, control

;-----

;Зацикливание программы

rjmp MAIN

\*\*\*\*\*

#####

;ТАБЛИЦА ИСТИННОСТИ

TABL:

.DB \$05,\$06,\$0F,\$0A,\$0F,\$00,\$0B,\$01,\$05,\$0C,\$0A,\$0E,\$0F,\$0B,\$0D,\$00;

#####

**ЗАДАНИЕ № 01**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Размыкающий
- датчик S3		Размыкающий
- датчик S2		Размыкающий
- датчик S1		Замыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 4, 5, 7, 9, 10, 11, 12, 14, 15    F2= $\bigvee_i^4$ , при i = 3, 4, 6, 8, 9, 10, 11, 13, 14		
F3= $\bigvee_i^4$ , при i = 2, 3, 5, 7, 8, 9, 10, 12, 13    F4= $\bigvee_i^4$ , при i = 1, 2, 4, 6, 7, 8, 9, 11, 12		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PB2
- датчик S3	4	PB4
- датчик S2	2	PB7
- датчик S1	1	PB3
- кнопка S0	-	PB5
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PC7
- исполнительное устройство W2	4	PC6
- исполнительное устройство W3	2	PC5
- исполнительное устройство W4	1	PC4
Язык программирования		Ассемблер

**ЗАДАНИЕ № 02**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Размыкающий
- датчик S3		Замыкающий
- датчик S2		Замыкающий
- датчик S1		Замыкающий
- кнопка S0		Замыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 3, 4, 6, 8, 9, 10, 11, 13, 14      F2= $\bigvee_i^4$ , при i = 2, 3, 5, 7, 8, 9, 10, 12, 13		
F3= $\bigvee_i^4$ , при i = 1, 2, 4, 6, 7, 8, 9, 11, 12      F4= $\bigvee_i^4$ , при i = 0, 1, 3, 5, 6, 7, 8, 10, 11		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PD4
- датчик S3	4	PD7
- датчик S2	2	PD6
- датчик S1	1	PD3
- кнопка S0	-	PD2
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PA4
- исполнительное устройство W2	4	PA3
- исполнительное устройство W3	2	PA2
- исполнительное устройство W4	1	PA1
Язык программирования		Ассемблер



**ЗАДАНИЕ № 03**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Размыкающий
- датчик S2		Размыкающий
- датчик S1		Размыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 0, 1, 6, 7, 9, 11, 12, 13, 14    F2= $\bigvee_i^4$ , при i = 0, 5, 6, 8, 10, 11, 12, 13, 15		
F3= $\bigvee_i^4$ , при i = 4, 5, 7, 9, 10, 11, 12, 14, 15    F4= $\bigvee_i^4$ , при i = 3, 4, 5, 6, 7, 8, 9, 10, 11		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PC2
- датчик S3	4	PC6
- датчик S2	2	PC5
- датчик S1	1	PC3
- кнопка S0	-	PC7
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PD4
- исполнительное устройство W2	4	PD3
- исполнительное устройство W3	2	PD2
- исполнительное устройство W4	1	PD1
Язык программирования	Ассемблер	

**ЗАДАНИЕ № 04**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Размыкающий
- датчик S2		Размыкающий
- датчик S1		Размыкающий
- кнопка S0		Замыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 0, 5, 6, 8, 10, 11, 12, 13, 15    F2= $\bigvee_i^4$ , при i = 4, 5, 7, 9, 10, 11, 12, 14, 15		
F3= $\bigvee_i^4$ , при i = 3, 4, 6, 8, 9, 10, 11, 13, 14    F4= $\bigvee_i^4$ , при i = 2, 3, 5, 7, 8, 9, 10, 12, 13		

**ОМК ATMega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PB4
- датчик S3	4	PB7
- датчик S2	2	PB3
- датчик S1	1	PB2
- кнопка S0	-	PB6
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PD6
- исполнительное устройство W2	4	PD5
- исполнительное устройство W3	2	PD4
- исполнительное устройство W4	1	PD3
Язык программирования		Ассемблер

**ЗАДАНИЕ № 05**

на контрольную работу «Разработка МПУУ на ОМК»  
 по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Размыкающий
- датчик S3		Размыкающий
- датчик S2		Размыкающий
- датчик S1		Замыкающий
- кнопка S0		Замыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 1, 2, 3, 8, 9, 11, 13, 14, 15    F2= $\bigvee_i^4$ , при i = 0, 1, 2, 7, 8, 10, 12, 13, 14		
F3= $\bigvee_i^4$ , при i = 0, 1, 6, 7, 9, 11, 12, 13, 15    F4= $\bigvee_i^4$ , при i = 0, 5, 6, 8, 10, 11, 13, 14, 15		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PD6
- датчик S3	4	PD3
- датчик S2	2	PD4
- датчик S1	1	PD7
- кнопка S0	-	PD2
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PB6
- исполнительное устройство W2	4	PB5
- исполнительное устройство W3	2	PB4
- исполнительное устройство W4	1	PB3
Язык программирования		Ассемблер

**ЗАДАНИЕ № 06**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Замыкающий
- датчик S2		Замыкающий
- датчик S1		Замыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДФ		
F1= $\vee_i^4$ , при i = 0, 1, 2, 7, 8, 10, 12, 13, 14    F2= $\vee_i^4$ , при i = 0, 1, 6, 7, 9, 11, 12, 13, 15		
F3= $\vee_i^4$ , при i = 0, 5, 6, 8, 10, 11, 12, 14, 15    F4= $\vee_i^4$ , при i = 4, 5, 7, 9, 10, 12, 13, 14, 15		

**ОМК ATMega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PB5
- датчик S3	4	PB2
- датчик S2	2	PB3
- датчик S1	1	PB7
- кнопка S0	-	PB4
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PC6
- исполнительное устройство W2	4	PC5
- исполнительное устройство W3	2	PC4
- исполнительное устройство W4	1	PC3
Язык программирования		Ассемблер

**ЗАДАНИЕ № 07**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Замыкающий
- датчик S2		Размыкающий
- датчик S1		Замыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 0, 1, 2, 3, 4, 5, 10, 11, 13,      F2= $\bigvee_i^4$ , при i = 0, 1, 2, 3, 4, 9, 10, 12, 15		
F3= $\bigvee_i^4$ , при i = 0, 2, 3, 8, 9, 11, 13, 14, 15      F4= $\bigvee_i^4$ , при i = 1, 2, 7, 8, 10, 12, 13, 14, 15		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PD2
- датчик S3	4	PD5
- датчик S2	2	PD7
- датчик S1	1	PD4
- кнопка S0	-	PD3
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PB7
- исполнительное устройство W2	4	PB6
- исполнительное устройство W3	2	PB5
- исполнительное устройство W4	1	PB4
Язык программирования		Ассемблер

**ЗАДАНИЕ № 08**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:	Тип контакта
- датчик S4	Замыкающий
- датчик S3	Замыкающий
- датчик S2	Замыкающий
- датчик S1	Размыкающий
- кнопка S0	Размыкающий
Исполнительные устройства	Тип ИУ
- исполнительное устройство W1	3-фазный асинхронный двигатель
- исполнительное устройство W2	3-фазный асинхронный двигатель
- исполнительное устройство W3	3-фазный асинхронный двигатель
- исполнительное устройство W4	3-фазный асинхронный двигатель
Способ задания функции: сокращенные СДФ	
F1= $\bigvee_i^4$ , при i = 0, 1, 2, 3, 4, 9, 10, 12, 14      F2= $\bigvee_i^4$ , при i = 0, 1, 2, 3, 8, 9, 11, 13, 15	
F3= $\bigvee_i^4$ , при i = 1, 2, 7, 8, 10, 12, 13, 14, 15      F4= $\bigvee_i^4$ , при i = 0, 1, 7, 9, 11, 12, 13, 14, 15	

**ОМК ATMega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PC7
- датчик S3	4	PC6
- датчик S2	2	PC3
- датчик S1	1	PC4
- кнопка S0	-	PC5
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PB7
- исполнительное устройство W2	4	PB6
- исполнительное устройство W3	2	PB5
- исполнительное устройство W4	1	PB4
Язык программирования	Ассемблер	

**ЗАДАНИЕ № 09**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Размыкающий
- датчик S3		Размыкающий
- датчик S2		Замыкающий
- датчик S1		Замыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 0, 1, 2, 5, 7, 12, 13, 14, 15    F2= $\bigvee_i^4$ , при i = 0, 1, 3, 4, 6, 11, 12, 13, 14		
F3= $\bigvee_i^4$ , при i = 0, 2, 3, 4, 5, 11, 12, 13, 15    F4= $\bigvee_i^4$ , при i = 1, 2, 4, 5, 10, 11, 12, 14, 15		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PC2
- датчик S3	4	PC4
- датчик S2	2	PC6
- датчик S1	1	PC3
- кнопка S0	-	PC7
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PA5
- исполнительное устройство W2	4	PA4
- исполнительное устройство W3	2	PA3
- исполнительное устройство W4	1	PA2
Язык программирования		Ассемблер

**ЗАДАНИЕ № 10**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Замыкающий
- датчик S2		Замыкающий
- датчик S1		Размыкающий
- кнопка S0		Замыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 1, 2, 3, 4, 6, 11, 12, 13, 14,    F2= $\bigvee_i^4$ , при i = 0, 2, 3, 4, 5, 10, 11, 13, 15		
F3= $\bigvee_i^4$ , при i = 0, 1, 3, 4, 9, 10, 12, 14, 15    F4= $\bigvee_i^4$ , при i = 0, 1, 2, 3, 9, 11, 13, 14, 15		

**ОМК ATMega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PC7
- датчик S3	4	PC3
- датчик S2	2	PC2
- датчик S1	1	PC4
- кнопка S0	-	PC5
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PD5
- исполнительное устройство W2	4	PD4
- исполнительное устройство W3	2	PD3
- исполнительное устройство W4	1	PD2
Язык программирования		Ассемблер



**ЗАДАНИЕ № 11**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Размыкающий
- датчик S2		Размыкающий
- датчик S1		Замыкающий
- кнопка S0		Замыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 1, 3, 5, 6, 7, 8, 9, 14, 15      F2= $\bigvee_i^4$ , при i = 0, 2, 4, 6, 7, 8, 13, 14, 15		
F3= $\bigvee_i^4$ , при i = 0, 1, 3, 5, 7, 12, 13, 14, 15      F4= $\bigvee_i^4$ , при i = 0, 1, 2, 4, 6, 11, 12, 13, 14		

**ОМК ATmega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PA5
- датчик S3	4	PA7
- датчик S2	2	PA4
- датчик S1	1	PA2
- кнопка S0	-	PA6
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PB6
- исполнительное устройство W2	4	PB5
- исполнительное устройство W3	2	PB4
- исполнительное устройство W4	1	PB3
Язык программирования	Ассемблер	

**ЗАДАНИЕ № 12**

на контрольную работу «Разработка МПУУ на ОМК»  
по дисциплине: «Микропроцессорные системы управления лифтов и эскалаторов»

**Исходные данные для проектирования МПУУ на ОМК**

Дискретные датчики и кнопка управления вводом:		Тип контакта
- датчик S4		Замыкающий
- датчик S3		Замыкающий
- датчик S2		Размыкающий
- датчик S1		Размыкающий
- кнопка S0		Размыкающий
Исполнительные устройства	Тип ИУ	
- исполнительное устройство W1	3-фазный асинхронный двигатель	
- исполнительное устройство W2	3-фазный асинхронный двигатель	
- исполнительное устройство W3	3-фазный асинхронный двигатель	
- исполнительное устройство W4	3-фазный асинхронный двигатель	
Способ задания функции: сокращенные СДНФ		
F1= $\bigvee_i^4$ , при i = 1, 2, 4, 6, 7, 8, 13, 14, 15      F2= $\bigvee_i^4$ , при i = 0, 2, 3, 5, 7, 12, 13, 14, 15		
F3= $\bigvee_i^4$ , при i = 0, 1, 3, 4, 6, 11, 12, 13, 14      F4= $\bigvee_i^4$ , при i = 0, 1, 2, 4, 5, 10, 11, 13, 15		

**ОМК ATMega32**

Дискретные датчики и кнопка ввода	Вес	Порт ввода
- датчик S4	8	PB2
- датчик S3	4	PB3
- датчик S2	2	PB7
- датчик S1	1	PB5
- кнопка S0	-	PB6
Исполнительные устройства	Вес	Порт вывода
- исполнительное устройство W1	8	PD5
- исполнительное устройство W2	4	PD4
- исполнительное устройство W3	2	PD3
- исполнительное устройство W4	1	PD2
Язык программирования		Ассемблер

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Семикопенко И.А. Лифты, строительные подъемники и вышки [Электронный ресурс]: учебное пособие/ Семикопенко И.А., Вялых С.В., Герасименко В.Б.— Электрон. текстовые данные.— Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2012.— 88 с.— Режим доступа: <http://www.iprbookshop.ru/28353.html>.— ЭБС «IPRbooks», по паролю.
  2. Путинцев Н.Н. Автоматизированный электропривод [Электронный ресурс]: учебно-методическое пособие/ Путинцев Н.Н., Бородин А.М., Сысенко В.Т.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2014.— 80 с.— Режим доступа: <http://www.iprbookshop.ru/45355.html>.— ЭБС «IPRbooks», по паролю.
  3. Сипайлова Н.Ю. Вопросы проектирования электрических аппаратов [Электронный ресурс]: учебное пособие/ Сипайлова Н.Ю.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2014.— 168 с.— Режим доступа: <http://www.iprbookshop.ru/34657.html>.— ЭБС «IPRbooks», по паролю.
- б) дополнительная учебная литература
4. Проектирование автоматизированных систем управления [Электронный ресурс] : методические указания к курсовому и дипломному проектированию для студентов дневного и очно-заочного отделения, изучающих дисциплину "Проектирование автоматизированных систем управления по направлению подготовки 15.03.04 "Автоматизация технологически процессов и производств / СПГУТД ; сост.: В. В. Сигачева, Д. А. Шурыгин. - СПб., 2015. - 24 с. - Режим доступа: [http://publish.sutd.ru/tp\\_ext\\_inf\\_publish.php?id=2307](http://publish.sutd.ru/tp_ext_inf_publish.php?id=2307), по паролю.
  5. Трофимов В.Б. Интеллектуальные автоматизированные системы управления технологическими объектами [Электронный ресурс]/ Трофимов В.Б., Кулаков С.М.— Электрон. текстовые данные.— М.: Инфра-Инженерия, 2016.— 232 с.— Режим доступа: <http://www.iprbookshop.ru/51726.html>.— ЭБС «IPRbooks», по паролю.