

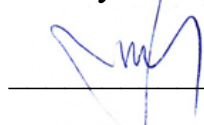
МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт высокоточных систем им. В.П. Грязева
Кафедра «Тренажерные системы и комплексы»

Утверждено на заседании кафедры
«Тренажерные системы и комплексы»
«__» _____ 20__ г., протокол №__

Заведующий кафедрой ТСК



Филиппов В.Н.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ по дисциплине (модулю)

Операционные системы реального времени

**основной профессиональной образовательной программы
высшего образования – программы бакалавриата**

по направлению подготовки
15.03.06 Мехатроника и робототехника

со профилем
Перспективные учебно-тренировочные средства

Форма(ы) обучения: *очная*

Идентификационный номер образовательной программы: 150306-03-22

Тула 2022 год

Лабораторная работа № 1

Разработка списочных расписаний для мультипроцессорных детерминированных вычислительных систем жесткого реального времени

Цель работы - приобретение практических навыков использования эвристического метода составления списочных расписаний для мультипроцессорных детерминированных вычислительных систем жесткого реального времени.

Методические указания.

1 Краткое теоретическое введение

Имеется однородная мультипроцессорная вычислительная система (ВС), состоящая из m идентичных процессоров ($m \geq 2$) и предназначенная для функционирования в реальном времени.

Вычисления, проводимые в реальном времени, представляют собой совокупность взаимосвязанных работ A_i ($i=1, \dots, n$), которые должны быть выполнены на процессорах ВС.

Отношения предшествования для совокупности работ задаются в форме ациклического связного ориентированного графа, i -м ($i=1, \dots, n$) вершинам которого соответствуют работы A_i ($i=1, \dots, n$).

Для каждой i -й работы известно целочисленное значение времени ее выполнения t_i ($i=1, \dots, n$). Вершины графа являются взвешенными, вес i -ой вершины - это t_i ($i=1, \dots, n$).

Требуется распределить работы из заданной совокупности с учетом их отношений предшествования по процессорам ВС, т.е. составить для нее расписание - план выполнения работ по каждому процессору ВС. Для ВС реального времени минимизируемой величиной при составлении расписания является длина расписания ω , т.е. общее время выполнения ВС всей совокупности работ.

В основе эвристического метода составления списочных расписаний лежит список $L=(A_{i_1}, A_{i_2}, \dots, A_{i_n})$, $i_k \in \{1, 2, \dots, n\}$, $k=1, \dots, n$. Работы в списке упорядочены слева направо по убыванию (не возрастанию) некоторого заданного приоритета.

Помимо списка L для указанного эвристического метода в качестве исходной информации используются следующие данные:

1. Матрица смежности $B = || b_{ij} ||$, $i, j = 1, \dots, n$, задающая структуру графа совокупности работ; n - число вершин графа.

Элементы матрицы определяются следующим образом:

$$b_{ij} = \begin{cases} 1, \text{ если } \Gamma(i) = \{j\}, \text{ т.е. из } i\text{-ой вершины есть} \\ \text{непосредственный переход в } j\text{-ую вершину;} \\ 0 - \text{ в противном случае.} \end{cases}$$

2. Вектор времен $T = || t_i ||$, $i = 1, \dots, n$, задающий целочисленные значения времен выполнения работ A_i ($i = 1, \dots, n$) на процессорах ВС.

Кроме того, эвристический метод использует матрицу достижимости $C = || c_{ij} ||$, $i, j = 1, \dots, n$, являющуюся производной матрицей от матрицы смежности.

Элементы матрицы достижимости определяются следующим образом:

$$c_{ij} = \begin{cases} 1, \text{ если } \Gamma^k(i) = \{j\}, k \in \{1, 2, \dots, n-1\}, \text{ т.е. из} \\ i\text{-ой вершины есть путь в } j\text{-ую вершину;} \\ 0 - \text{ в противном случае.} \end{cases}$$

Результатом составления расписания являются временные диаграммы Ганта для каждого процессора ВС реального времени, которые отображают на временной оси порядок выполнения соответствующих работ и простоев процессоров и определяют длину расписания ω .

Суть эвристического метода составления списочных расписаний для ВС реального времени заключается в следующем.

В нулевой момент времени список L просматривается слева направо в поисках открытых работ, т.е. работ, готовых к выполнению. Соответствующий столбец матрицы достижимости в этом случае является нулевым.

Первая найденная открытая работа назначается для выполнения без прерываний на первый процессор, и по завершении ее выполнения обнуляется соответствующая строка матрицы достижимости.

Вообще, в каждый целочисленный момент времени, если имеется свободный от выполнения работы процессор, список L просматривается снова. Если нет ни одной работы, готовой к выполнению, то процессор простаивает. Если же есть несколько свободных от выполнения работ процессоров, то первая найденная открытая работа назначается на процессор с меньшим номером.

Расписания, полученные таким способом, называются списочными расписаниями.

2 Пример

Рассмотрим случай для $m=2$, $n=6$. Граф для совокупности работ представлен на рис. 1.1

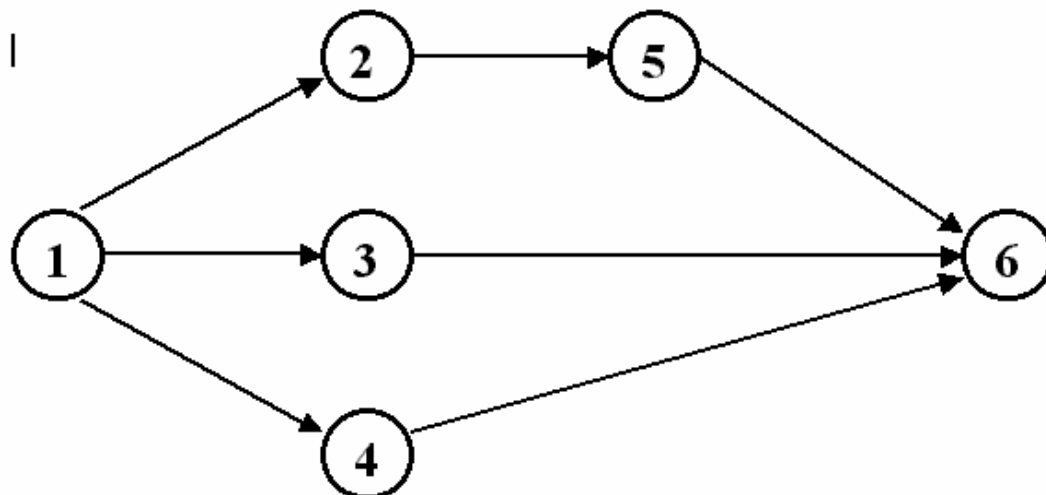


Рис. 1.1

Матрица смежности B , матрица достижимости C , вектор времен T и список L имеют следующий вид:

$$B = \begin{array}{c|c|c|c|c|c|c|} & j \rightarrow & 1 & 2 & 3 & 4 & 5 & 6 \\ i \downarrow & & & & & & & \\ 1 & & 0 & 1 & 1 & 1 & 0 & 0 \\ 2 & & 0 & 0 & 0 & 0 & 1 & 0 \\ 3 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 5 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 6 & & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$C = \begin{array}{c|c|c|c|c|c|c|} & j \rightarrow & 1 & 2 & 3 & 4 & 5 & 6 \\ i \downarrow & & & & & & & \\ 1 & & 0 & 1 & 1 & 1 & 1 & 1 \\ 2 & & 0 & 0 & 0 & 0 & 1 & 1 \\ 3 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 5 & & 0 & 0 & 0 & 0 & 0 & 1 \\ 6 & & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$T = \begin{array}{c} i \downarrow \\ \begin{array}{|c|c|c|} \hline 1 & 3 & \\ \hline 2 & 2 & \\ \hline 3 & 4 & \\ \hline 2 & 4 & \\ \hline 5 & 3 & \\ \hline 6 & 4 & \\ \hline \end{array} \end{array}$$

$$L = (A_1, A_2, A_3, A_4, A_5, A_6) .$$

Списочное расписание, составленное в соответствии с описанным выше эвристическим методом, представлено в виде временных диаграмм Ганта на рис. 1.2. Длина полученного расписания $\omega=14$.

P1	1	1	1	2	2	4	4	5	5	5	6	6	6	6
P2	■	■	■	3	3	3	3	■	■	■	■	■	■	■
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

единицы времени

P1, P2 - процессоры;
 ■ - простой процессора.

Рис. 1.2

3 Задание

Разработать на заданном языке программу построения списочного расписания для ВС реального времени в соответствии с выданным преподавателем вариантом задания и произвести тестирование разработанной программы.

Исходными данными для программы являются число процессоров m , число работ n , матрица смежности B и вектор времен T , результатами - временные диаграммы Ганта для каждого процессора ВС и длина полученного списочного расписания ω .

Для контроля следует предусмотреть вывод сформированных программой матрицы достижимости C и списка L .

4 Содержание отчета

1. Задание и его исходные данные.
2. Граф для совокупности работ.
3. Матрица достижимости C .
4. Список L с указанием для каждой его работы значений заданного приоритета.
5. Укрупненная схема программы.
6. Списочное расписание, построенное вручную по варианту задания (тест).
7. Текст программы.
8. Результаты выполнения (тестирования) программы.
9. Выводы по лабораторной работе.

5 Варианты заданий

Варианты заданий представлены в табл. 1.1.

Таблица 1.1.

№ ва ри ан та	M	n	Матрица смежности В												Вектор времён Т		Приоритет для списка L	
			i	j												i		T _i
				1	2	3	4	5	6	7	8	9	10	11	12			

1	4	12	1	0	1	1	1	1	0	0	0	0	0	0	0	1	2	Наибольшее
			2	0	0	0	0	0	0	1	0	0	0	0	0	2	2	значение крити-
			3	0	0	0	0	0	1	0	1	0	0	0	0	3	4	ческого по коли-
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	3	честву вершин
			5	0	0	0	0	0	0	0	0	0	1	0	0	5	2	пути от данной
			6	0	0	0	0	0	0	1	0	1	0	0	0	6	4	вершины до ко-
			7	0	0	0	0	0	0	0	0	0	1	1	0	7	4	нечной /
			8	0	0	0	0	0	0	0	0	0	1	0	0	8	2	Наименьшее
			9	0	0	0	0	0	0	0	0	0	1	1	0	9	3	значение крити-
			10	0	0	0	0	0	0	0	0	0	0	0	1	10	3	ческого по вре-
			11	0	0	0	0	0	0	0	0	0	0	0	1	11	4	мени пути от на-
			12	0	0	0	0	0	0	0	0	0	0	0	0	12	3	чальной верши-
																		ны до данной
2	3	11	1	0	1	1	1	0	0	0	0	0	0			1	2	Наименьшее
			2	0	0	0	0	1	0	0	0	0	0			2	3	значение крити-
			3	0	0	0	0	0	1	0	0	0	0			3	4	ческого по коли-
			4	0	0	0	0	0	0	1	0	0	0			4	3	честву вершин
			5	0	0	0	0	0	1	0	1	1	1	0		5	4	пути от началь-
			6	0	0	0	0	0	0	0	0	1	1	0		6	2	ной вершины до
			7	0	0	0	0	0	0	0	0	0	1	1		7	3	данной /
			8	0	0	0	0	0	0	0	0	1	1	0		8	3	Наибольшее
			9	0	0	0	0	0	0	0	0	0	0	1		9	4	значение крити-
			10	0	0	0	0	0	0	0	0	0	0	1		10	2	ческого по коли-
			11	0	0	0	0	0	0	0	0	0	0	0		11	4	честву вершин
																		пути от данной
																		вершины до ко-
																		нечной
3	3	10	1	0	1	1	1	0	0	0	0	0	0			1	2	Наименьшее
			2	0	0	0	0	1	0	0	0	0	0			2	3	значение крити-
			3	0	0	0	0	1	0	1	0	0	0			3	3	ческого по вре-
			4	0	0	0	0	0	0	0	0	0	1			4	4	мени пути от на-
			5	0	0	0	0	0	1	1	0	0	0			5	2	чальной верши-
			6	0	0	0	0	0	0	0	1	0	0			6	3	ны до данной
			7	0	0	0	0	0	0	0	0	1	1			7	4	/
			8	0	0	0	0	0	0	0	0	0	1			8	4	Наибольшее
			9	0	0	0	0	0	0	0	0	0	1			9	2	значение крити-
			10	0	0	0	0	0	0	0	0	0	0			10	3	ческого по вре-
																		мени пути от
																		данной вершины
																		до конечной

4	4	12	1	0	1	1	1	1	0	0	0	0	0	0	0	1	3	Наименьшее время выполнения работы, соответствующей данной вершине / Наибольшее значение критического по количеству вершин пути от данной вершины до конечной
			2	0	0	0	0	0	1	0	0	0	0	0	0	2	2	
			3	0	0	0	0	0	1	1	0	0	0	0	0	3	2	
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	3	
			5	0	0	0	0	0	0	0	1	1	1	0	0	5	4	
			6	0	0	0	0	0	0	0	0	0	0	1	0	6	4	
			7	0	0	0	0	0	0	0	0	1	0	1	0	7	3	
			8	0	0	0	0	0	0	0	0	0	0	0	1	8	2	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	0	1	10	2	
			11	0	0	0	0	0	0	0	0	0	0	0	1	11	3	
			12	0	0	0	0	0	0	0	0	0	0	0	0	12	4	
5	3	11	1	0	1	1	1	0	0	0	0	0	0	0	0	1	2	Наименьшее значение критического по количеству вершин пути от начальной вершины до данной / Наибольшая связность на глубину 1
			2	0	0	0	0	1	1	0	0	0	0	0	0	2	3	
			3	0	0	0	0	0	1	0	1	0	0	0	0	3	2	
			4	0	0	0	0	0	0	0	0	0	1	0	0	4	3	
			5	0	0	0	0	0	0	0	1	0	0	0	0	5	4	
			6	0	0	0	0	0	0	1	0	1	0	0	0	6	4	
			7	0	0	0	0	0	0	0	0	0	1	0	0	7	3	
			8	0	0	0	0	0	0	0	0	1	0	1	0	8	2	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	1	0	10	2	
			11	0	0	0	0	0	0	0	0	0	0	0	0	11	4	
6	3	10	1	0	1	1	1	0	0	0	0	0	0	0	0	1	3	Наименьшее время выполнения работы, соответствующей данной вершине / Наибольшее значение критического по времени пути от данной вершины до конечной
			2	0	0	0	0	1	0	0	0	0	0	0	0	2	2	
			3	0	0	0	0	1	0	1	0	0	0	0	0	3	3	
			4	0	0	0	0	0	1	0	0	1	0	0	0	4	2	
			5	0	0	0	0	0	0	0	0	1	0	0	0	5	4	
			6	0	0	0	0	0	0	1	1	0	0	0	0	6	3	
			7	0	0	0	0	0	0	0	0	1	0	0	0	7	2	
			8	0	0	0	0	0	0	0	0	0	1	0	0	8	4	
			9	0	0	0	0	0	0	0	0	0	1	0	0	9	2	
			10	0	0	0	0	0	0	0	0	0	0	0	0	10	4	

7	4	12	1	0	1	1	1	1	0	0	0	0	0	0	0	1	2	Наименьшее значение критического по времени пути от начальной вершины до данной / Наибольшее значение критического по количеству вершин пути от данной вершины до конечной
			2	0	0	0	0	0	0	0	1	0	0	0	0	2	3	
			3	0	0	0	0	0	0	0	0	1	0	0	0	3	3	
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	4	
			5	0	0	0	0	0	1	0	0	1	0	0	0	5	3	
			6	0	0	0	0	0	0	0	1	0	0	0	0	6	2	
			7	0	0	0	0	0	0	0	0	0	1	0	0	7	4	
			8	0	0	0	0	0	0	0	0	0	0	1	1	8	2	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	1	0	10	2	
			11	0	0	0	0	0	0	0	0	0	0	0	1	11	4	
			12	0	0	0	0	0	0	0	0	0	0	0	0	12	4	
8	3	11	1	0	1	1	1	0	0	0	0	0	0	0	0	1	3	Наименьшее значение критического по количеству вершин пути от начальной вершины до данной / Наибольшее значение критического по времени пути от данной вершины до конечной
			2	0	0	0	0	0	1	0	0	0	0	0	0	2	2	
			3	0	0	0	0	1	0	0	0	0	0	0	0	3	2	
			4	0	0	0	0	0	1	0	1	0	0	0	0	4	3	
			5	0	0	0	0	0	0	1	0	0	0	0	0	5	4	
			6	0	0	0	0	0	0	0	0	1	0	0	0	6	2	
			7	0	0	0	0	0	0	0	1	0	0	0	0	7	3	
			8	0	0	0	0	0	0	0	0	1	1	0	0	8	4	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	2	
			10	0	0	0	0	0	0	0	0	0	0	1	0	10	3	
			11	0	0	0	0	0	0	0	0	0	0	0	0	11	4	
9	3	10	1	0	1	1	1	0	0	0	0	0	0	0	0	1	2	Наименьшее значение критического по времени пути от начальной вершины до данной / Наибольшая связность на глубину 1
			2	0	0	0	0	1	0	0	0	0	0	0	0	2	4	
			3	0	0	0	0	0	1	0	0	1	0	0	0	3	3	
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	3	
			5	0	0	0	0	0	1	0	1	0	0	0	0	5	2	
			6	0	0	0	0	0	0	1	0	0	1	0	0	6	2	
			7	0	0	0	0	0	0	0	0	0	0	1	0	7	4	
			8	0	0	0	0	0	0	0	0	0	0	1	0	8	3	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	0	0	10	4	

10	4	12	1	0	1	1	1	1	0	0	0	0	0	0	0	1	3	Наибольшее значение критического по времени пути от данной вершины до конечной /
			2	0	0	0	0	0	0	0	1	0	0	0	0	2	2	Наименьшее время выполнения работы, соответствующей данной вершине
			3	0	0	0	0	0	1	0	0	0	0	1	0	3	2	
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	3	
			5	0	0	0	0	0	0	0	1	0	0	0	0	5	4	
			6	0	0	0	0	0	0	0	0	0	1	0	0	6	2	
			7	0	0	0	0	0	0	0	0	1	0	0	0	7	2	
			8	0	0	0	0	0	0	0	0	0	0	1	0	8	3	
			9	0	0	0	0	0	0	0	0	0	1	1	0	9	4	
			10	0	0	0	0	0	0	0	0	0	0	0	1	10	3	
			11	0	0	0	0	0	0	0	0	0	0	0	1	11	4	
			12	0	0	0	0	0	0	0	0	0	0	0	0	12	4	
11	3	11	1	0	1	1	1	0	0	0	0	0	0	0	0	1	3	Наибольшая связность на глубину 1 /
			2	0	0	0	0	1	0	1	0	0	0	0	0	2	2	Наименьшее значение критического по количеству вершин пути от начальной вершины до данной
			3	0	0	0	0	0	1	0	0	0	0	0	0	3	3	
			4	0	0	0	0	0	0	1	0	0	0	0	0	4	2	
			5	0	0	0	0	0	0	0	1	0	0	0	0	5	2	
			6	0	0	0	0	0	0	0	0	1	0	0	0	6	4	
			7	0	0	0	0	0	0	0	0	0	1	0	0	7	3	
			8	0	0	0	0	0	0	0	0	1	1	0	0	8	4	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	1	0	10	2	
			11	0	0	0	0	0	0	0	0	0	0	0	0	11	4	
12	3	10	1	0	1	1	1	0	0	0	0	0	0	0	0	1	2	Наименьшее время выполнения работы, соответствующей данной вершине /
			2	0	0	0	0	0	0	1	0	0	0	0	0	2	3	Наибольшая связность на глубину 1
			3	0	0	0	0	1	0	0	0	1	0	0	0	3	2	
			4	0	0	0	0	0	1	0	0	0	0	0	0	4	3	
			5	0	0	0	0	0	0	0	1	0	0	0	0	5	2	
			6	0	0	0	0	0	0	1	0	0	0	0	0	6	2	
			7	0	0	0	0	0	0	0	1	1	0	0	0	7	3	
			8	0	0	0	0	0	0	0	0	0	1	0	0	8	4	
			9	0	0	0	0	0	0	0	0	0	0	1	0	9	3	
			10	0	0	0	0	0	0	0	0	0	0	0	0	10	4	

Примечания.

1. Приоритет для формирования списка L имеет следующий вид:
Первый приоритет / Второй приоритет. Предпочтение отдается вершине-

работе по первому приоритету, а при наличии нескольких вершин-работ с одинаковым значением первого приоритета - по второму приоритету; в случае одинаковых значений второго приоритета предпочтение отдается вершине-работе с меньшим номером.

2. Связность на глубину 1 - это количество непосредственных приемников данной вершины в графе.

3. Критический путь - это путь наибольшей длины по количеству вершин, его составляющих, или по сумме времен выполнения работ для вершин, его составляющих.

Лабораторная работа № 2

Разработка прикладного программного обеспечения микропроцессорных систем жесткого реального времени на базе однокристальных микроконтроллеров.

Цель работы - приобретение практических навыков разработки прикладных программ на языке ассемблера для микропроцессорных систем управления объектом на базе микроконтроллера КМ1816ВЕ48 с использованием кросс-системы разработки.

Методические указания

1 Краткое теоретическое введение

1.1 Обобщенная структура микропроцессорной системы управления

Независимо от конкретного объекта управления и спектра задач, решаемых автоматизированной микропроцессорной системой, структура этой системы будет выглядеть примерно одинаково, как показано на рис. 2.1.



Рис. 2.1

Работа данной системы может быть описана следующим образом. Микроконтроллер, представляющий собой фактически однокристалльную микроЭВМ, реализует в соответствии с текущим состоянием системы заложенный в него алгоритм управления. Анализ текущего состояния системы выполняется с помощью осведомительных сигналов, поступающих с датчиков. Так как природа управляемого процесса может быть различной, вид и форма сигналов, формируемых датчиками, также произвольна. Поэтому может потребоваться преобразование исходного сигнала в цифровую форму для ввода в ЭВМ. Преобразование может быть выполнено с помощью аналого-цифровых преобразователей (АЦП) соответствующей разрядности и схем формирования сигналов, которые дополнительно выполняют функции гальванической развязки. Осведомительные сигналы могут также участвовать в формировании специальных признаков - флагов, необходимых микроконтроллеру для принятия альтернативных решений.

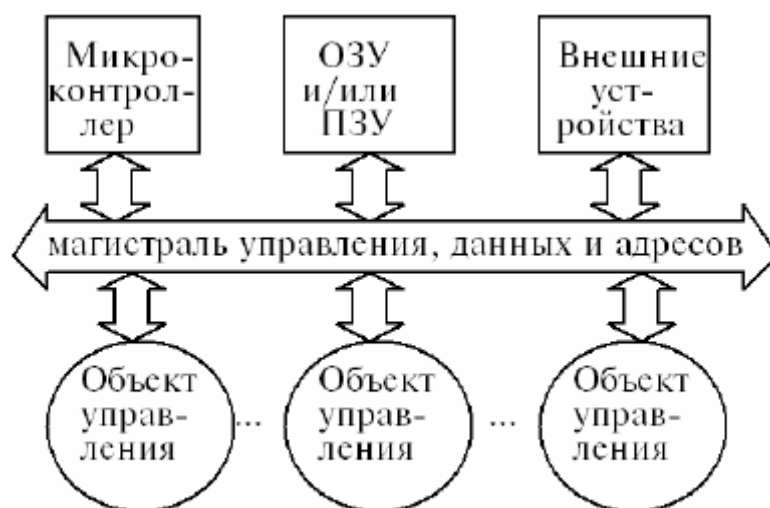
Алгоритм управления реализуется микроконтроллером путем оказания исполнительных воздействий на управляемый объект через посредство управляющих сигналов. Как было отмечено выше, для согласования сигналов различной формы может потребоваться преобразование, которое обычно реализуется с помощью цифро-аналоговых преобразователей (ЦАП) соответствующей разрядности и формирователей сигналов.

Таймеры в системе могут быть как встроенными в микроконтроллер, так и внешними, и необходимы для формирования и поддержания временных интервалов и соотношений в алгоритме управления.

Специфика структуры проектируемой системы проявляется при детализации проекта, связанной с выбором принципов взаимодействия с объектами, свойствами объектов управления и элементной базой.

Если объект управления один, то вопрос взаимодействия с объектом сводится к выбору типа интерфейса, обеспечивающего подключение микроконтроллера с целью приема осведомительных и выдачи управляющих сигналов для реализации алгоритма управления с необходимой точностью и быстродействием.

Если объектов управления несколько, то помимо указанного выше вопроса необходимо выбрать магистральную или радиальную конфигурацию системы управления, как показано на рис. 2.2а и рис. 2.2б.



а)



б)

Рис. 2.2

Свойства и физическая природа объектов управления определяют выбор преобразователей информации, их точность и быстродействие.

Выбор элементной базы зависит от класса решаемых задач, сложности алгоритмов обработки данных и управления, скорости управляемых процессов, потребностей в размерах памяти и т.д.

Большинство возникающих на практике задач управления может быть выполнено с помощью однокристалльных микроконтроллеров K1816BE48 и K1816BE51. На примере этих микроконтроллеров рассмотрим вопросы проектирования, возникающие при создании систем управления динамическим объектом, разработке подсистем сбора данных, создании цифровых измерительных приборов.

Тип микроконтроллера определен в задании, поэтому здесь следует отметить только его основные характеристики: разрядность арифметико-логического устройства - 8 бит; тактовая частота - от 1 до 6 МГц; емкость резидентной памяти программ - 1 кбайт; емкость встроенной памяти данных - 64 байта; три восьмиразрядных порта.

В зависимости от характера объекта управления ввод информации с датчиков может выполняться в режиме ожидания или последовательного опроса. В режиме ожидания микроконтроллер в цикле ждет появления на фиксированном входе какого-либо события, в качестве которого могут быть заданы перепад уровня сигнала, импульс, последовательность импульсов и др.

Выводимые из микроконтроллера сигналы могут иметь статическую и импульсную форму в зависимости от характера объекта управления.

Реализация временных соотношений при разработке систем на базе однокристалльных микроконтроллеров может выполняться путем соответствующего программирования встроенного таймера. Пусть, например, тактовая частота f генератора равна 6 МГц, тогда частота синхросигнала машинного цикла равна: $f_{\text{Ц}} = f/15 = 400 \text{ кГц}$. Следовательно, на вход таймера импульсы будут поступать с частотой $f_{\text{Т}} = f_{\text{Ц}}/32 = 12,5 \text{ кГц}$, что для одного такта составляет задержку $t_{\text{Т}} = 1/f_{\text{Т}} = 80 \text{ мкс}$. Таким образом, если на таймере необходимо сформировать задержку t_3 длительностью, например, в 4 мс, нужно в счетчик-таймер перед началом цикла ожидания загрузить код $N = 256 - t_3/t_{\text{Т}} = 256 - 50 = 206_{10} = \text{CE}_{16}$.

Временная задержка заданной длительности также может быть реализована методом вложенных программных циклов.

1.2 Средства отладки программного обеспечения

1.2.1 Состав и работа кросс-систем

Основными составными частями кросс-системы являются: кросс-ассемблер, кросс-редактор связей, эмулятор.

Работа с кросс-системой начинается с составления программы на языке ассемблера для микро-ЭВМ К1816ВЕ48. Подготовка программы выполняется с помощью любого текстового редактора.

Подготовленная программа транслируется кросс-ассемблером. Основными функциями кросс-ассемблера являются синтаксический и семантический контроль исходного текста программы, преобразование исходного текста в объектный код, формирование листинга ассемблирования.

Из объектного кода в машинный программа преобразуется с помощью кросс-редактора связей.

Полученный загрузочный модуль отлаживается с помощью соответствующего эмулятора.

1.2.2 Правила записи программ на языке ассемблера

Исходный текст программы на языке ассемблера имеет определенный формат. Каждая команда (и псевдокоманда) представляет собой строку четырехзвенной конструкции:

МЕТКА ОПЕРАЦИЯ ОПЕРАНД(Ы) КОММЕНТАРИЙ.

Звенья (поля) могут отделяться друг от друга произвольным числом пробелов.

Метка. В поле метки размещается символическое имя ячейки памяти, в которой хранится отмеченная команда или операнд. Метка представляет собой буквенно-цифровую комбинацию, начинающуюся с буквы. Используются только буквы латинского алфавита. Ассемблер МК51 допускает использование в метках символа подчеркивания (). Длина метки не должна превышать шесть символов для МК48 и 31 – для МК51. Метка всегда завершается двоеточием (:).

Псевдокоманды ассемблера не преобразуются в двоичные коды, а потому не могут иметь меток. Исключение составляют псевдокоманды резервирования памяти и определения данных (DS, DB, DW). У псевдокоманд, осуществляющих определение символических имен, в поле метки записывается определяемое символическое имя, после которого двоеточие не ставится.

В качестве символических имен и меток не могут быть использованы мнемокоды команд, псевдокоманд и операторов ассемблера, а также мнемонические обозначения регистров и других внутренних блоков МК.

Операция. В поле операции записывается мнемоническое обозначение команды МК или псевдокоманды ассемблера, которое является сокращением (аббревиатурой) полного английского наименования выполняемого действия. Например: MOV – move – переместить, JMP – jump – перейти, DB – define byte – определить байт.

Для МК48 и МК51 используется строго определенный и ограниченный набор мнемонических кодов. Любой другой набор символов, размещенный в поле операции, воспринимается ассемблером как ошибочный.

Операнды. В этом поле определяются операнды (или операнд), участвующие в операции. Команды ассемблера могут быть без-, одно- или двухоперандными. Операнды разделяются запятой (,).

Операнд может быть задан непосредственно или в виде его адреса (прямого или косвенного). Непосредственный операнд представляется числом (MOV A, #15) или символическим именем (ADDC A, #OPER2), с обязательным указанием префикса непосредственного операнда (#). Прямой адрес операнда может быть задан мнемоническим обозначением (IN A, P1), числом (INC 40), символическим именем (MOV A, MEMORY). Указанием на косвенную адресацию служит префикс @. В командах передачи управления операндом может являться число (LCALL 0135H), метка (JMP LABEL), косвенный адрес (JMPP @A) или выражение (JMP $\square - 2$, где \square – текущее содержимое счетчика команд).

Используемые в качестве операндов символические имена и метки должны быть определены, а числа представлены с указанием системы счисления, для чего используется суффикс (буква, стоящая после числа): В – для двоичной, Q – для восьмеричной, D – для десятичной и H – для шестнадцатеричной. Число без суффикса по умолчанию считается десятичным.

Обработка выражений в процессе трансляции. Ассемблеры МК48 и МК51 допускают использование выражений в поле операндов, значения которых вычисляются в процессе трансляции.

Выражение представляет собой совокупность символических имен и чисел, связанных операторами ассемблера. Операторы ассемблера обеспечивают выполнение арифметических ("+" – сложение, "-" – вычитание, * – умножение, / – целое деление, MOD – деление по модулю) и логических (OR – ИЛИ, AND – И, XOR – исключающее ИЛИ, NOT – отрицание) операций в формате 2-байтных слов.

Например, запись ADD A, #((NOT 13) + 1) эквивалентна записи ADD A, #0F3H и обеспечивает сложение содержимого аккумулятора с числом – 13, представленным в дополнительном коде.

Широко используются также операторы LOW и HIGH, позволяющие выделить младший и старший байты 2-байтного операнда.

Комментарий. Поле комментария может быть использовано программистом для текстового или символьного пояснения логической организации прикладной программы. Поле комментария полностью игнорируется ассемблером, а потому в нем допустимо использовать любые симво-

лы. По правилам языка ассемблера поле комментария начинается после точки с запятой (;).

Псевдокоманды ассемблера. Ассемблирующая программа транслирует исходную программу в объектные коды. Хотя транслирующая программа берет на себя многие из рутинных задач программиста, таких как присвоение действительных адресов, преобразование чисел, присвоение действительных значений символьным переменным и т.п., программист все же должен указать ей некоторые параметры: начальный адрес прикладной программы, конец ассемблируемой программы, форматы данных и т.п. Всю эту информацию программист вставляет в исходный текст своей прикладной программы в виде псевдокоманд (директив) ассемблера, которые только управляют процессом трансляции и не преобразуются в коды объектной программы.

Псевдокоманда **ORG 10H** задает ассемблеру адрес ячейки памяти (10H), в которой должна быть расположена следующая за ней команда прикладной программы.

Псевдокомандой **EQU** можно любому символическому имени, используемому в программе, поставить в соответствие определенный операнд.

Например, запись
PET EQU 13

приводит к тому, что в процессе ассемблирования всюду, где встретится символическое имя **PET**, оно будет заменено числом 13.

Символические имена операндов, переопределяемых в процессе исполнения программы, определяются псевдокомандой **SET**:

ALFA SET 3

...
...

...
ALFA SET ALFA+1

Ассемблер **MK51** позволяет определить символическое имя как адрес внутренних (псевдокоманда **DATA**), внешних (**XDATA**) данных или адрес бита (псевдокоманда **BIT**). Например, директива

ERROR_FLAG BIT 25H.3

определяет символическое имя **ERROR_FLAG** как третий бит ячейки ОЗУ с адресом 25H.

Псевдокоманда **DB** обеспечивает занесение в ПП константы, представляющей собой байт.

Псевдокомандой **END** программист дает ассемблеру указание об окончании трансляции.

В результате трансляции должна быть получена карта памяти программ, где каждой ячейке памяти поставлен в соответствие хранящийся в ней код.

В соответствии с форматом команд для представления их объектных кодов отводятся одна, две или три ячейки памяти программ. В первой ячейке всегда располагается код операции, во второй (а для МК51 и в третьей) – непосредственный операнд, адрес прямоадресуемого операнда, адрес перехода внутри страницы памяти программ (для команд передачи управления МК48) или смещение (для команд передачи управления МК51). Для команд LCALL. и LJMP во втором и третьем байтах объектного кода указывается адрес передачи управления (во втором – старшая часть, в третьем – младшая) .

Пример программы на языке ассемблера приведен на рисунке 2.3.

```
EN    I      ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЙ
JMP   START
NOP           ;АДРЕС 0003(ОБРАБОТКА ПЕРЕРЫВАНИЯ)
RETR        ;ЗАВЕРШЕНИЕ ОБРАБОТКИ ПЕРЕРЫВАНИЯ
START: MOV   A,#100B ;ПОДГОТОВКА ПОРТА P4
        ORLD  P4,A   ;К ВВОДУ (2-Й БИТ)
        MOVD  A,P4   ;ВВОД ИЗ ПОРТА P4
        JB2   START  ;ПЕРЕХОД, ЕСЛИ ВО 2-М БИТЕ ЕДИНИЦА
        MOV   A,#7
        MOVD  P5,A   ;ВЫВОД В ПОРТ P5
        ORL   P1,#0FFH ;ПОДГ. ПОРТА P1 К ВВОДУ(ВСЕ БИТЫ)
        IN    A,P1   ;ВВОД ИЗ ПОРТА P1
        CALL  PP
        JMP   MET
PP:     INS   A,BUS   ;ВВОД ИЗ ПОРТА P0
        OUTL  P1,A   ;ВЫВОД В ПОРТ P1
        RET
MET:    NOP
        JMP   START
```

Рис. 2.3

1.2.3 Кросс-ассемблер

Входным файлом для кросс-ассемблера является исходный текст программы на языке ассемблера однокристалльной микроЭВМ серии K1816BE48, записанный на диск в виде файла любым редактором текста

данной операционной системы. Расширение имени входного файла по умолчанию принимается равным **ASM**.

При разработке программ, реализуемых в виде единого модуля, можно использовать вариант кросс-ассемблера **x48.com**. После запуска этого ассемблера запрашивается имя исходного файла, условия и носитель (дисплей, принтер или файл) для вывода листинга откомпилированной программы. В последнем случае, листинг выводится в файл с расширением **PRN**.

В результате компиляции формируется сразу загрузочный модуль в формате **PGM**:

Маркер начала адреса записи	Адрес за- грузки	Маркер конца адреса записи	Коды программы
<	aaaa	>	www

где а и w - шестнадцатеричные значения в кодах ASCII адреса и кода программы.

Пример:

<0010> 26 10 36 12 35 B8 03 99 00 B9 00 85 95 25 A5 80

1.2.4 Эмулятор микроЭВМ K1816BE48

Отладка программа для ОМЭВМ K1816BE48 выполняется на ее программном эмуляторе **e48.exe**, входным файлом для которого является загрузочный модуль в формате **PGM**.

После запуска эмулятора выдается приглашение для ввода имени файла с отлаживаемой программой, который должен находиться в текущем каталоге. Затем на экране отображаются панель состояния микроЭВМ и командная строка эмулятора. Форма панели состояний может быть задана оконной (Рис. 2.3) или построчной (Рис. 2.4).

Команды эмулятора позволяют изменять состояние рабочих регистров и портов ОМЭВМ, содержимое ячеек внутренней и внешней памяти данных, а также корректировать команды в памяти программ в режиме построчного ассемблирования. Имеется возможность дизассемблирования части команд в текстовый файл на диске и ведения протокола отладки в специальном файле с расширением **LST**. Вид выводимой в протокол информации зависит от варианта вывода на экран. При оконном выводе фиксируется полное состояние ОМЭВМ, в том числе, содержимое внешней и внутренней памяти данных. При построчном выводе протоколируется только содержимое регистров.

Адр	Команда	Внутреннее ОЗУ															
Адр		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0060		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0070		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
		Внешнее ОЗУ															
Адр		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
		Регистры и переключатели															
		AC=00				CY=0				F0=0				R0=00			
		C/T=00				HC=0				F1=0				R1=00			
		SP=00				RB=0				T0=0				R2=00			
		PC=0010				IF=0				T1=0				R3=00			
		INT=DIS				C/T=DIS				R4=00				R5=00			
		CircleCounter=								R6=00				R7=00			
										INT=1				P1=FF			
														P2=FF			
														P3=00			
														P4=D			
														P5=B			
														P6=D			
														P7=0			
														C/TMode=STOP			
														4			

Space- Шаг ?- Помощь ^F- Переключение экранов ^C- Выход в DOS

Рис. 2.4

Адр	Команда	AC	CT	F0/1	T0/1	P1	P2	BS	PSW	TF	R0	R1	R2	R3	R4	R5	R6	R7	I
0010	26 JNT0 010	00	00	0	0	0	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0010	26 JNT0 010	00	00	0	0	0	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0010	26 JNT0 010	00	00	1	0	1	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0012	36 JT0 012	00	00	1	0	1	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0012	36 JT0 012	00	00	1	0	1	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0012	36 JT0 012	00	00	1	0	0	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0014	35 DIS ICNTI	00	00	1	0	0	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0015	B8 MOV R0,#03	00	00	1	0	0	0	FF	FF	00	00	0	00	00	00	00	00	00	1
0017	99 ANL P1,#00	00	00	1	0	0	0	FF	FF	00	00	0	03	00	00	00	00	00	1
0019	B9 MOV R1,#00	00	00	1	0	0	0	00	FF	00	00	0	03	00	00	00	00	00	1
001B	85 CLR F0	00	00	1	0	0	0	00	FF	00	00	0	03	00	00	00	00	00	1
001C	95 CPL F0	00	00	0	0	0	0	00	FF	00	00	0	03	00	00	00	00	00	1
001D	25 EN ICNTI	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
001E	A5 CLR F1	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
001F	80 MOVX A,@R0	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
0020	62 MOV T,A	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
0021	55 STRT T	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
0022	76 JF1 02A	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1
0024	B6 JF0 022	00	00	1	0	0	0	00	FF	00	28	0	03	00	00	00	00	00	1

Space- Шаг ?- Помощь ^F- Переключение экранов ^C- Выход в DOS

Рис. 2.5

Отладку программы можно вести в пошаговом режиме, когда каждая команда отлаживаемой программы выполняется после нажатия клавиши **<Space>**, или в автоматическом режиме, когда запущенная соответствующей командой эмулятора программа выполняется до точки останова или нажатия клавиши **<Space>**.

Ниже приведено описание команд эмулятора, вводимых нажатием определенных клавиш.

1. Команды модификации отдельных бит, регистров и ячеек памяти:

- <A>** - модификация содержимого аккумулятора;
- ** - смена текущего рабочего банка регистров;
- <C>** - модификация содержимого таймера / счетчика;
- <D>** - модификация содержимого ячейки внутренней памяти данных;
- <E>** - установка начального адреса области внешней памяти, отображаемой в окне экрана;
- <F>** - инвертирование флага F0 или F1;
- <I>** - инвертирование состояния входа INT;
- <M>** - установка начального адреса области внутренней памяти, отображаемой в окне экрана;
- <O>** - модификация содержимого счетчика циклов CircleCounter;
- <P>** - модификация содержимого одного из регистров P1, P2, P4, P5, P6, P7, PC, BUS;
- <R>** - модификация содержимого одного из регистров R0-R7;
- <S>** - инвертирование флага CY или HC;
- <T>** - инвертирование флага T0 или T1;
- <X>** - модификация значения ячейки внешней памяти данных;
- <?>** - вызов подсказки по командам эмулятора.

Ниже приведены несколько примеров использования команд модификации:

1) Установка значения аккумулятора, равного 3D :

<A>

Введите значение аккумулятора >3D<Enter>

2) Установка содержимого программного счетчика PC, равным 12F:

<P>

Установка P(C) P(1/2/4/5/6/7) (B)US ><C>

Введите значение регистра >'12F<Enter>.

3) Инвертирование флага F0:

<F>

Сменить F(0) F(1) >0 .

4) Модификации значения ячейки внешней памяти данных с адресом 32 , и отображение ее содержимого в окне:

<E>

Введите адрес начала внешней памяти >30<Enter>'

<X>

Введите адрес данных внешней памяти>32<Enter>'

Введите значение >15<Enter>'.

2. Команды редактирования программы:

<Ctrl-A> - ассемблирование. Данная команда позволяет осуществлять корректировку программы с построчным ассемблированием.

Пример занесения фрагмента программы с адреса 56:

<Ctrl-A>

Введите адрес > 56 <Enter>'

Ассемблер (Выход - Enter)/A:0056/>INC R0<Enter>'

Ассемблер (Выход - Enter)/A:0057/>ADD A,#3E<Enter>'

Ассемблер (Выход - Enter)/A:0059/>MOV @R0,A<Enter>'.

<Ctrl-D> - дизассемблирование участка памяти. Данная команда производит дизассемблирование участка памяти программ в текстовый файл .

Пример дизассемблирования участка памяти с 0-го по 55-й адрес и вывод результатов в файл 90v315.txt.

<Ctrl-D>

Введите имя файла >90V315<Enter>'.

Если при открытии файла возникает ошибка ввода-вывода, то выдается сообщение:

Файл не может быть открыт.

Если операция открытия файла выполняется успешно, то последовательно выдаются следующие сообщения:

Введите начальный адрес дизассемблирования > 0<Enter>'

Введите конечный адрес дизассемблирования > 55<Enter>'

<<<<<<<<<Ждите>>>>>>>>> .

3. Команды управления моделированием:

<Ctrl-B> - ввод или удаление точек (условий) останова.

Пример задания точки останова перед выполнением команды по адресу 56:

<Ctrl-B>

Останов по содержимому (A)C (R)X (P)C (T)aktCounter (C)lear P

Введите номер точки останова >1

Введите адрес останова > 56 <Enter>'.

<Ctrl-C> - прекращение работы и выход в DOS.

<Ctrl-E> - быстрое моделирование. При выполнении программы в этом режиме на экране меняется только РС и счетчик рабочих циклов. Возврат в командный режим происходит либо при нажатии любой клавиши, либо при выполнении одного из условий останова.

<Ctrl-F> - изменение формы отображения на экране.

<Ctrl-T> - обычный режим моделирования. При выполнении каждой команды программы происходит вывод на экран всех изменений состояния системы и дизассемблирование этой команды.

<Ctrl-R> - выход из подпрограммы или прерывания.

<Ctrl-L> - управление протоколированием отладки; можно открывать или закрывать файл протокола отладки программы.

Пример применения:

<Ctrl-L>

Если файл протокола ранее не был открыт, то выдается следующее сообщение:

Открыть файл листинга (Y/N) > Y

Введите имя файла > 90V315 <Enter>

Если при открытии файла возникает ошибка ввода-вывода, то выдается сообщение:

Файл не может быть открыт

Если файл протокола был открыт, то выдается сообщение:

Закрыть файл листинга (Y/N) > Y

Необходимо нажать **Y** для закрытия файла листинга или **N** для продолжения протоколирования.

<Ctrl-I>- инициализация ЦП. Выполнение этой команды аналогично подаче сигнала RESET на вход ОМЭВМ.

2 Задание

В соответствии с выданным преподавателем вариантом задания составить концептуальный алгоритм работы управляющей вычислительной системы (УВС) на базе микроконтроллера КМ1816ВЕ48 и после согласования данного алгоритма с преподавателем осуществить разработку, тестирование и отладку прикладной программы на языке ассемблера с использованием кросс-системы разработки.

Для всех вариантов задания необходимо предусмотреть возможность ввода внешнего прерывания по входу INT в случае какой-либо аварийной

ситуации (например, неисправность, сбой в работе аппаратуры и т. п.) . Подпрограмма обслуживания внешнего прерывания должна обеспечить подачу звукового и светового аварийных сигналов выводом нуля в биты 1 и 0 порта P7 соответственно, после чего перейти в режим ожидания нуля в бите 2 порта P7 . После прихода сигнала снятия аварийного режима (ноль в указанном бите) подпрограмма обслуживания внешнего прерывания прекращает подачу звукового и светового аварийных сигналов выводом единицы в биты 1 и 0 порта P7 соответственно и возобновляется работа прерванной программы , причем нужно предусмотреть возможность сохранения и восстановления содержимого аккумулятора .

Поскольку предусматривается использование в составе УВС стандартного расширителя ввода-вывода для получения дополнительных портов ввода-вывода P4-P7, младшие биты 3-0 порта P2, используемые для данного расширения, для других целей задействовать нельзя.

Тактовая частота микроконтроллера принимается равной 6 МГц и соответственно время выполнения машинного цикла составляет 2.5 Мксек .

Заданные программные временные задержки требуется рассчитать, оформить в виде подпрограмм и отладить отдельно с помощью кросс-системы, а при отладке основной программы для подпрограмм использовать «заглушки».

Примечания.

1. Биты в восьмиразрядных регистрах и восьмиразрядных (четырёхразрядных) портах нумеруются со старшего - седьмого (третьего) до младшего - нулевого.

2. По адресу 0003H должна располагаться подпрограмма обслуживания внешнего прерывания по входу INT.

3. При использовании портов P1, P2, P4-P7 для ввода данных эти порты должны быть предварительно подготовлены к вводу путем установки в них единиц по соответствующим командам вывода или командам логических операций.

4. Сравнение двух величин может быть реализовано путем вычитания и последующего перехода по флагу переноса. Например, может использоваться следующий вариант. Значение первой величины (уменьшаемое) заносится в аккумулятор и инвертируется; значение второй величины (вычитаемое) – в какой-либо регистр. Затем производится сложение содержимого аккумулятора и регистра, после чего реализуется переход по флагу переноса (JC) в случае отрицательной разности. В противном случае содержимое аккумулятора (после сложения) инвертируется и реализуется переход (JZ) по флагу нуля в случае равенства величин; в противном случае разность положительна.

5. Для организации вложенных циклов в подпрограммах временных задержек удобно использовать команду DJNZ.

6. Вводимые с клавиатуры данные при отладке предполагаются заданными в шестнадцатеричной форме.

3 Содержание отчёта

1. Задание и его исходные данные .
2. Укрупнённая схема программы для основной программы и подпрограмм, включая подпрограмму обработки прерывания .
3. Листинг программы с необходимыми комментариями
4. Протокол тестирования .
5. Выводы по выполненной работе .

4 Варианты заданий

1. Управление работой лифта

В системе различаются внешние кнопки для вызова лифта с этажа и внутренние - для вызова нужного этажа в лифте.

Имеются команды включения двигателя для работы на подъём, включения двигателя для работы на спуск и команда выключения двигателя, а также команды открытия и закрытия двери лифта.

Условие нажатия внешней кнопки (ноль) вводится в бит 0 порта P1, внутренней кнопки (ноль) - в бит 1 порта P1. Номер этажа внешней кнопки (от 1 до 12) вводится в порт P4, внутренней кнопки (от 1 до 12) - в порт P5.

Условие закрытия (ноль) и открытия (единица) двери вводится в бит 2 порта P1. Значение текущего номера этажа (от 1 до 12), на котором находится лифт, вводится в порт P0.

Команда включения двигателя на подъём (ноль) выводится в бит 3 порта P6, на спуск (ноль) - в бит 2 порта P6, выключения двигателя (единица) - в биты 3 или 2 порта P6 соответственно.

Команда открытия двери (ноль) и закрытия (единица) выводится в бит 7 порта P2.

Следует реализовать программную временную задержку длительностью 10 Сек после открытия двери.

В программной задержке нужно предусмотреть возможность досрочного выхода из нее в случае нажатия внутренней кнопки лифта.

2. Управление работой запорного устройства сейфа

Чтобы открыть сейф с секретным запорным устройством, нужно сначала повернуть диск запорного устройства, имеющий 255 делений по окружности относительно неподвижного фиксатора положения диска. Затем требуется набрать четыре последовательные цифры кода (от 0 до 9), после чего нужно нажать кнопку на запорном устройстве. Если положение диска и набранный код совпадают с истинным шифром, то запорное устройство открывается и включается сигнальная лампа.

При неправильно набранном шифре раздаётся звуковой сигнал в течение 5 сек, после чего можно повторить указанную процедуру.

После закрытия дверцы сейфа сигнальная лампа гаснет и запорное устройство закрывается.

Положение диска (значение от 1 до 255) вводится в порт P0.

Цифры кода вводятся в порты P2 (биты 7-4), P4, P5, P6 соответственно.

Условие нажатия кнопки (ноль) вводится в бит 7 порта P1.

Условие закрытия (единица) и открытия (ноль) дверцы сейфа вводится в бит 6 порта P1.

Команда включения (ноль) и выключения (единица) звукового сигнала выводится в бит 5 порта P1.

Команда включения (ноль) и выключения (единица) запорного устройства сейфа выводится в бит 4 порта P1.

Команда включения (ноль) и выключения (единица) сигнальной лампы выводится в бит 3 порта P1.

3. Управление работой холодильника

Имеются кнопка пуска, команды включения и выключения двигателя вентилятора, включения и выключения нагревателя размораживающего устройства на полную мощность и половину мощности соответственно и выключения нагревателя размораживающего устройства, включения и выключения компрессора.

При повышении температуры морозильного устройства выше нормы возникает угроза обмерзания и повышения температуры внутри камеры, поэтому в этом случае вентилятор охлаждения выключается, также выключается компрессор, а нагреватель размораживающего устройства включается сначала на полную мощность, а затем через 30 Сек выключается и включается на половину мощности и после еще 30 Сек выключается, после чего происходит включение компрессора и двигателя вентилятора.

Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7. Значение текущей температуры морозильного устройства вводится в порт P0 .

Включение и выключение вентилятора осуществляется выводом в бит 3 порта P6 нуля и единицы соответственно.

Включение и выключение компрессора осуществляется выводом в бит 3 порта P5 нуля и единицы соответственно.

Включение нагревателя размораживающего устройства на полную мощность осуществляется выводом нуля в бит 3 порта P4, на половину мощности - в бит 2 порта P4, а выключение - выводом единицы в указанные биты.

Предельная температура, которая не должна превышаться, составляет в машинном выражении значение 10Н .

4. Управление работой системы регулирования микроклимата

Имеются кнопка пуска системы, переключатель режимов работы: режим 1 и режим 2 для температуры - в машинном представлении диапазон от 10Н до 15Н и от 14Н до 19Н соответственно; режим 3 и режим 4 для влажности - в машинном представлении диапазон от 20Н до 30Н и от 40Н до 50Н соответственно. Возможны следующие комбинации режимов: 1-3, 1-4, 2-3, 2-4 .

В системе имеются нагревательный элемент, охлаждающее устройство, устройство поглощения влаги из воздуха, влагообразующее устройство.

Имеются команды включения и выключения указанных устройств.

Команда включения указанных устройств (ноль) выводится в бит 2 порта P4 для нагревательного элемента, в бит 1 порта P4 для охлаждающего устройства, в бит 3 порта P5 для устройства поглощения влаги, в бит 2 порта P5 для влагообразующего устройства. Команда выключения указанных устройств (единица) выводится в те же перечисленные биты .

Значение текущей температуры воздуха вводится в порт P0, а текущей влажности воздуха - в порт P1.

Режимы работы вводятся в биты 3 (режим 1 - ноль, режим 2 - единица) и 2 (режим 3 - ноль, режим 4 - единица) порта P6 .

При выходе текущих значений температуры и влажности воздуха за заданные пределы включаются или выключаются соответствующие устройства системы и реализуется временная задержка на 40 Сек .

Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7 .

5. Управление работой светофора

Имеются два транспортных светофора, работающих на пересекающиеся направления проезжей части и состоящих из набора цветных ламп : красной, жёлтой и зелёной, а также два пешеходных светофора, состоящих из набора цветных ламп : красной и зеленой,- и имеющих кнопку для требования форсированного перехода проезжей части.

Работа светофоров заключается в последовательной периодической смене следующих комбинаций :

ТС2-"кр", ТС1-"зел" ,	ПС2-"зел", ПС1-"кр";
ТС2-"кр", ТС1-"жел" ,	ПС2-"зел", ПС1-"кр";
ТС2-"кр", ТС2-"жел", ТС1-"кр" ,	ПС2-"кр", ПС1-"зел";
ТС2-"зел", ТС1-"кр" ,	ПС2-"кр", ПС1-"зел";
ТС2-"жел", ТС1-"кр" ,	ПС2-"кр", ПС1-"зел";
ТС2-"кр", ТС1-"жел", ТС1-"кр" ,	ПС2-"зел", ПС1-"кр" ;

где ТС1 и ТС2 - транспортные светофоры для 1-ого и 2-ого направлений соответственно, ПС1 и ПС2 - пешеходные светофоры для перехода проезжей части по 1-ому и 2-ому направлению соответственно.

Продолжительность каждой комбинации перед её сменой следующая : 1) и 4) - 10 Сек , 2) и 5) - 3 Сек , 3) и 6) - 2 Сек .

Помимо обычного режима "красный-желтый-зеленый" возможен режим "мигалки", когда включаются и выключаются желтые лампы обоих светофоров с периодом в 1 Сек и блокируется работа пешеходных светофоров.

Режим работы светофоров вводится в бит 3 порта Р6 : ноль - обычный режим, единица - режим "мигалки" .

Включение и выключение ламп осуществляется выводом в соответствующий бит нуля и единицы соответственно.

Для первого транспортного светофора это биты 2,1,0 порта Р4 для красной, желтой и зеленой ламп соответственно, а для аналогичных ламп второго транспортного светофора - это биты 2,1,0 порта Р5 . Для первого пешеходного светофора это биты 3 и 2 порта Р7 для красной и зелёной ламп соответственно, а для аналогичных ламп второго пешеходного светофора - это биты 1 и 0 порта Р7 .

В программной задержке для 1-ой и 4-ой комбинаций (10 сек) нужно предусмотреть возможность досрочного выхода из неё в случае нажатия кнопки соответствующего пешеходного светофора. Условие нажатия кнопки (0) вводится в бит 2 порта Р6 и бит 1 порта Р6 для 1-ого и 2-ого пешеходного светофора соответственно .

6. Управление работой стиральной машины

В системе имеется кнопка пуска машины, тумблер переключения режимов стирки белья разного типа : нормальный с интенсивной цирку-

ляцией и бережный с менее интенсивной циркуляцией. В первом режиме активатор вращается против часовой стрелки, во втором - по часовой стрелке .

Имеются команды включения двигателя для работы в первом или втором режиме, выключения двигателя, включения и выключения клапана подачи воды соответственно, включения и выключения нагревательного элемента соответственно .

Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7 . Значение текущего уровня воды в баке вводится в порт P0, значение текущей температуры воды в баке вводится в порт P1 .

Режим стирки вводится в бит 7 порта P2 : ноль - первый режим , единица - второй .

Команда включения двигателя для работы в первом режиме (ноль) выводится в бит 3 порта P6, во втором режиме (ноль) - в бит 2 порта P6, выключения двигателя (единица) - в указанные биты .

Команда открытия клапана подачи воды (ноль) выводится в бит 0 порта P5, закрытия (единица) - в тот же бит .

Команда включения нагревательного элемента (ноль) выводится в бит 2 порта P4, выключения (единица) - в тот же бит .

Уровень воды для стирки в любом режиме должен находиться в пределах от 70 См до 75 См, что соответствует в машинном представлении значениям 46Н и 4ВН .

Температура воды для стирки в первом режиме должна находиться в пределах от 70 град до 90 град (в машинном представлении - от 46Н до 5АН), во втором режиме - от 50 град до 60 град (от 32Н до 3СН) .

После включения двигателя следует реализовать программную временную задержку длительностью 1 мин для первого режима и 1.5 мин для второго режима .

7. Управление работой кодового замка двери

В нормальном состоянии замок закрыт. Чтобы открыть замок, необходимо сначала набрать три последовательные цифры кода (от 0 до 9), после чего нажать кнопку замка . Если набранный код совпадает с эталоном, то замок открывается и загорается сигнальная лампа. После входа в помещение и закрытия двери сигнальная лампа гаснет и замок закрывается .

При попытке открывания двери с неправильно набранным кодом, включается звуковой сигнал тревоги в течение 10 Сек , после чего можно повторить попытку набора кода.

Последовательные цифры кода вводятся в порты P4,P5,P6 соответственно .

Условие нажатия кнопки (ноль) вводится в бит 3 порта P7 .

Условие закрытия (единица) и открытия (ноль) двери вводится в бит 7 порта P2 .

Команда включения (ноль) и выключения (единица) звукового сигнала выводится в бит 6 порта P2 .

Команда открытия (ноль) и закрытия (единица) замка выводится в бит 5 порта P2 .

Команда включения (ноль) и выключения (единица) сигнальной лампы выводится в бит 4 порта P2 .

8. Управление работой надплитного воздухоочистителя

Имеются кнопка пуска системы, тумблер включения-выключения освещения плиты, тумблер включения-выключения вентиляционной системы : на первую скорость работы двигателя (нормальный режим) и на вторую скорость работы двигателя (форсированный режим) ; тумблер включения-выключения бактерицидной лампы; тумблер включения-выключения автоматического режима работы, который заключается в том, что сначала включается бактерицидная лампа на 45 Сек, затем бактерицидная лампа выключается и включается вентиляционная система на нормальный режим также на 45 Сек, после чего вентиляционная система выключается и вся описанная процедура повторяется.

Имеются команды включения и выключения освещения вентиляционной системы на нормальный или форсированный режим, бактерицидной лампы .

Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7 .

Условие включения освещения (ноль) вводится в бит 3 порта P6, условие включения бактерицидной лампы (ноль) - в бит 2 порта P6, условие автоматического режима (ноль) - в бит 1 порта P5 , режим работы вентиляционной системы - в бит 3 порта P5 : ноль - нормальный режим, единица - форсированный режим .

Включение и выключение освещения осуществляется выводом в бит 3 порта P4 нуля и единицы соответственно, включение и выключение бактерицидной лампы - выводом в бит 2 порта P4 нуля и единицы соответственно .

Команда включения вентиляционной системы в нормальном режиме (ноль) выводится в бит 7 порта P2, в форсированном режиме (ноль) - в бит 6 порта P2, выключения (единица) - в указанные биты .

Автоматический режим работы воздухоочистителя является приоритетным по отношению к другим режимам.

9. Управление работой кондиционера

Имеется кнопка пуска системы, переключатель режимов работы : режим 1 - одновременная непрерывная работа вентилятора и нагревательного элемента; режим 2 - одновременная непрерывная работа вентилятора и охлаждающего устройства; режим 3 - одновременная работа вентилятора и нагревательного элемента с периодическим (30 Сек) их выключением; режим 4 - одновременная работа вентилятора и нагревательного элемента с периодическим (45 Сек) выключением последнего; режим 5 - одновременная работа вентилятора и охлаждающего устройства с периодическим (30 Сек) их выключением; режим 6 - одновременная работа вентилятора и охлаждающего устройства с периодическим (45 Сек) выключением последнего .

Кроме того, имеется переключатель уровня заслонки для потока воздуха : 1 - высокий, 2 - средний , 3 - низкий .

Команда включения вентилятора, нагревательного элемента и охлаждающего устройства (ноль) выводится в бит 3 порта P6, бит 2 порта P6 и бит 1 порта P6 соответственно. Команда выключения указанных устройств (единица) выводится в те же перечисленные биты . Команда установки уровня заслонки (ноль) выводится в бит 7,6 или 5 порта P1 для высокого, среднего и низкого уровня соответственно.

Значение номера режима (от 1 до 6) вводится в порт P5, значение уровня заслонки (от 1 до 3) - в порт P4.

Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7 .

10. Управление работой СВЧ-электроплиты

Имеется кнопка пуска системы, переключатель уровня выходной мощности плиты : 1 – малый уровень; 2 - средний уровень ; 3 – большой уровень .

Команда включения плиты на малый уровень мощности (ноль) выводится в бит 3 порта P6, на средний уровень мощности (ноль) - в бит 2 порта P6, на большой уровень мощности (ноль) - в бит 1 порта P6, команда выключения плиты (единица) - в биты 3, 2 или 1 порта P6 соответственно.

Значение уровня выходной мощности плиты (от 1 до 3) вводится в порт P4, значение заданного времени нагревания, кратного 1 минуте (от 1 до 5 минут) – в порт P5, значение заданной температуры в пределах от 30 град до 120 град (в машинном представлении - от 1EH до 78H) – в порт P1, текущее значение температуры от сенсора температурного пробника пищи – в порт P0. Условие нажатия кнопки пуска (ноль) вводится в бит 3 порта P7 .

При достижении заданного значения температуры после включения плиты реализуется заданная временная задержка, по окончании которой должен раздаваться звуковой сигнал продолжительностью в 1 секунду.

Команда подачи и прекращения звукового сигнала реализуется выводом нуля и единицы соответственно в бит 1 порта P7.

Лабораторная работа № 3

Разработка прикладных программ для систем жесткого реального времени с использованием языков высокого уровня

Цель работы - приобретение практических навыков разработки прикладных программ на языках реального времени высокого уровня, таких как Модула, Ада, для систем жесткого реального времени.

Методические указания

1 Постановка задачи

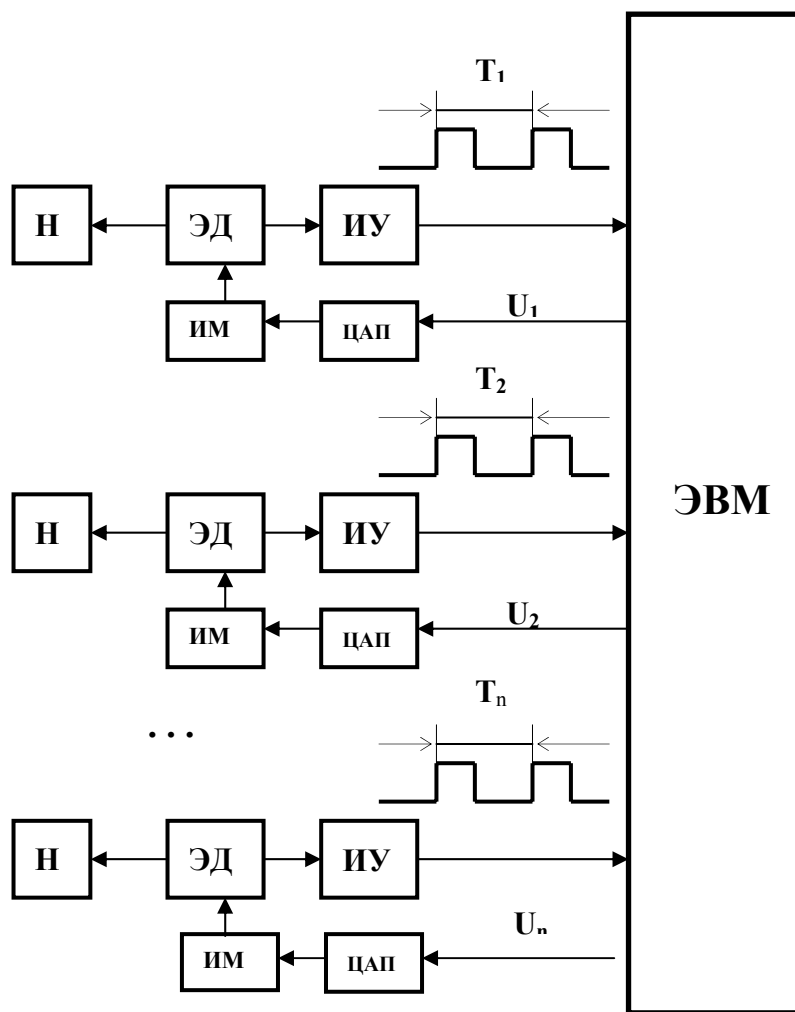
Имеется система жесткого реального времени на базе персонального компьютера, предназначенная для одновременного регулирования нескольких электродвигателей, которые в общем случае, могут быть разного типа.

Регулирование заключается в обеспечении независимой от нагрузки постоянной частоты вращения электродвигателей.

Укрупненная структура системы приведена на рис. 3.1.

Интервал времени T_i ($i=1, \dots, n$) в поступающей от измерительного устройства последовательности импульсов является мерой мгновенной частоты вращения данного электродвигателя. ЭВМ должна определить этот интервал, сравнить его с заданной величиной и, как только возникнет рассогласование, выдать соответствующее управляющее воздействие U_i ($i=1, \dots, n$), которое после цифро-аналогового преобразования передается на исполнительный механизм, вырабатывающий управляющее напряжение, которое служит для электродвигателя задающей величиной и корректирует его частоту вращения.

Каждый импульс измерительного устройства воспринимается ЭВМ как требование прерывания и вызывает запуск соответствующих программных компонентов, которые должны решить задачу регулирования.



ЭД - электродвигатель;
 Н - нагрузка;
 ИУ - измерительное устройство;
 ЦАП - цифро-аналоговый преобразователь;
 ИМ - исполнительный механизм

Рис. 3.1

Для определения рассогласования может быть использован, например, таймер-счетчик, который перебирает все состояния и начинает очередной перебор снова с нуля. В этом случае при каждом новом требовании прерывания определяется разность между текущим содержимым таймера-счетчика и его содержимым при предыдущем прерывании. Данная разность пропорциональна интервалу времени.

Другой, более предпочтительный вариант определения рассогласования заключается в использовании текущих значений системных аппаратных часов реального времени.

Рассмотрим, для примера, количественные характеристики, связанные с измерительным устройством.

Пусть заданная частота вращения электродвигателя должна составлять 300 оборотов в минуту. Измерительное устройство выдает за один оборот один импульс. Следовательно, требование прерывания возникает с интервалом в 200 мСек ($200 \cdot 10^{-3}$ секунд).

2 Задание

В соответствии с выданным преподавателем вариантом задания организовать многопроцессную (многозадачную) структуру прикладного программного обеспечения на указанном преподавателем языке (языках) высокого уровня для описанной системы жесткого реального времени.

Для внешних устройств требуется с использованием, в частности, датчика случайных чисел, часов реального времени и таймера разработать программный эмулятор, имитирующий их функционирование, включая выдачу сигнала прерывания, возникновение рассогласования и соответствующую реакцию на выдачу управляющего воздействия.

Выдача рассчитанного управляющего воздействия должна производиться в указанный порт вывода и одновременно отображаться на экране, причем, поскольку реальное подключение внешних устройств при отладке программного обеспечения отсутствует, следует ту часть программы, которая реализует выдачу управляющего воздействия в порт вывода, закомментировать.

Предусмотреть визуализацию процесса регулирования на экране в отдельном окне для каждого электродвигателя, в частности, заданное значение частоты вращения (интервала поступления требования прерывания), отклонение от него и корректировка рассогласования (управляющее воздействие), а также выдачу текстового сообщения в случае выключения (или аварии) электродвигателя.

Предполагается, что измерительные устройства в системе являются идентичными и выдают за пять оборотов электродвигателя один импульс.

Требуется рассчитать для каждого электродвигателя с учетом заданной частоты его вращения интервал поступления требований прерывания от соответствующего измерительного устройства.

Увеличение или уменьшение заданного значения интервала поступления требования прерывания в процессе регулирования частоты вращения электродвигателя должно приводить к выдаче в соответствующий для

данного электродвигателя порт вывода ЭВМ значения управляющего воздействия U для компенсации рассогласования: $U=128-K*\Delta t$ или $U=128+K*\Delta t$ соответственно, где Δt - время в миллисекундах, на которое увеличивается или уменьшается интервал поступления требования прерывания; K - заданная константа.

При $\Delta t=0$ выдача управляющего воздействия не производится. Пределы значений управляющих воздействий и соответственно пределы регулирования лежат в диапазоне 1-255. Если рассчитанное значение управляющего воздействия выходит за указанный диапазон, то это означает, что произошло фатальное рассогласование, регулирование для данного электродвигателя прекращается и на экран выдается соответствующее сообщение.

3 Содержание отчета

1. Задание и его исходные данные.
2. Укрупненная структура (схема программы) для прикладного программного обеспечения и его компонентов.
3. Листинг программы.
4. Экранная копия с визуализацией процесса регулирования.
5. Выводы по выполненной работе.

4 Варианты заданий

Варианты заданий представлены в табл. 3.1.

Таблица 3.1

№ вари- анта	Коли- чество ЭД	Частота враще- ния ЭД (об/мин)*10 ²				№ порта вывода				Значе- ние кон- станты К*10 ⁻²
		№ ЭД				№ ЭД				
		1	2	3	4	1	2	3	4	
1	4	1	1.2	1.5	2	32	33	34	35	10
2	3	2.5	3	1	-	13	14	15	-	12
3	4	1	1.5	2.5	3	28	29	30	31	14
4	3	1.5	2	3	-	10	11	12	-	11
5	4	1.5	1.2	2.5	2	24	25	26	27	13
6	3	1.2	1	3	-	7	8	9	-	11
7	4	2	3	1.2	2.5	20	21	22	23	13
8	3	3	1.5	1	-	4	5	6	-	10
9	4	3	2	2.5	1	16	17	18	19	12
10	3	2	1.2	1.5	-	1	2	3	-	14

Список литературы

1. Головкин Б.А. Расчет характеристик и планирование параллельных вычислительных процессов . - М.: Радио и связь, 1983 .
2. Вайрадян А.С. и др. Эффективное функционирование управляющих мультипроцессорных систем . - М.: Радио и связь, 1984 .
3. Теория расписаний и вычислительные машины . / Под ред. Э.Г. Коффмана. - М.: Наука, 1984 .
4. Теория расписаний и вычислительные машины / Под ред. Э.Г. Коффмана. – М.: Наука, 1984.
5. Сташин В.В. Проектирование цифровых устройств на однокристальных микроконтроллерах / В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. – М.: Энергоатомиздат, 1990.
6. Гурин Е.И., Коннов Н.Н., Кучин А.В. и др. Проектирование систем управления на базе однокристальных микроконтроллеров. Пенза, ПГТУ, 1996.
7. МикроЭВМ: В 8 кн.: Практ. пособие / Под ред. Л.Н.Преснухина . Кн. 3 Семейство ЭВМ "Электроника К1"/ А.В.Кобылинский и др. - М.: Высш. шк., 1988 .
8. Франке К.Введение в микро-ЭВМ. М.: Энергоатомиздат, 1988.
9. Белецкий Я. ТопСпид: Расширенная версия языка Модула-2 для персональных компьютеров IBM. -М.: Машиностроение, 1993.
10. Янг С. Алгоритмические языки реального времени: конструирование и разработка. М.: Мир, 1985.
11. JPI Modula - 2 . Руководство пользователя . / Перевод с англ. Барышев А.С. и др. - Минск, 1990 .
12. Василеску Ю. Прикладное программирование на языке Ада. - М.: Мир, 1990 .

Содержание

Лабораторная работа № 1. Разработка списочных расписаний для мультипроцессорных детерминированных вычислительных систем жесткого реального времени	3
Лабораторная работа № 2. Разработка прикладного программного обеспечения микропроцессорных систем жесткого реального времени на базе однокристальных микроконтроллеров	13
Лабораторная работа № 3. Разработка прикладных программ для систем жесткого реального времени с использованием языков высокого уровня	35
Список литературы	40

СИСТЕМЫ ЖЕСТКОГО РЕАЛЬНОГО ВРЕМЕНИ

Методические указания к выполнению лабораторных работ

Редактор В.В. Чувашова

Технический редактор Н.А. Вьялкова

Корректор Ж.А. Лубенцова

Компьютерная верстка Н.В. Иванова

ИД № 06494 от 26.12.01

Сдано в производство 19.02.03. Формат 60X841/16. Бумага газетная.

Печать офсетная. Усл. печ. л. 2,56. Уч.-изд. л. 3,05.

Заказ № 125. Тираж 75. "С" 24.

Издательство Пензенского государственного университета.

440026, Пенза, Красная, 40.

Разработчики методических указаний

Ломакин А.К., ст.преподаватель каф. САУ

(ФИО, должность, ученая степень, ученое звание)

(подпись)
