

СЕКЦИЯ 2

НАУКОЕМКИЕ ТЕХНОЛОГИИ

**Председатель:
профессор Парфенов Е.М.**

**Руководитель экспертной комиссии:
профессор Белов Б.И.**

Ученый секретарь: Соловьев В.А.

**Экспертная комиссия: профессор ЧЕКАНОВ А.Н, к.т.н.
БИЛИБИН К.И., к.т.н. БУРАК Б.А., к.т.н. ГРИГОРЬЕВ В.П.,
к.т.н. ГРИДНЕВ В.Н., к.т.н. ИВАНОВ Ю.В., к.т.н. МАРКЕЛОВ В.
В., к.т.н. ПАРФЕНОВ О.Д., ГЛАЗУНОВА Г.П.**

РАЗРАБОТКА РАСПИСАНИЯ ЗАПУСКА В ПРОИЗВОДСТВО ЭЛЕКТРОННЫХ ЯЧЕЕК ПРИ ОБЪЕМНОМ ПЛАНИРОВАНИИ

Иванов Юрий Викторович, Курносенко Алексей Евгеньевич

каф. ИУ-4 МГТУ им. Н.Э. Баумана

Иу4, Московский государственный технический университет им. Н.Э. Баумана

WORKING OUT THE TOTAL PLAN OF ELECTRONIC BOARDS PRODUCTION LAUNCHING

Ivanov Jury V., Kurnosenko Alexey E.

IU-4 department of BMSTU Ivanov Jury V.

IU4, Bauman Moscow State Technical University.

mario7@online.ru

Аннотация

Рассматривается метод разработки расписания запуска электронных ячеек на сборку в условиях гибкого автоматизированного многономенклатурного производства. Метод включает этапы объемного и календарного планирования. Описывается первый этап, на котором решается задача распределения годовой программы запуска по плановым периодам – месяцам и декадам с учетом ограничений на годовой (месячный) объем запуска в единицах продукции, загрузку оборудования и объем запуска в денежном выражении.

Abstract

The method of working out the plan of launching the flexible automated production of electronic boards is discussed. The method contains of two main stages: total and calendar planning. On the first stage the problem of distribution of the year production program to months and decades is worked out. The limitations on the year (month) production program, equipment charging and on the total amount of production are taken into account.

Общая задача разработки расписания запуска электронных ячеек (ЭЯ) в условиях многономенклатурного производства представлена двумя взаимоувязанными этапами: объемного и календарного планирования.

Задачей первого этапа является разработка программ запуска ЭЯ в производство для гибкого автоматизированного сборочного комплекса (ГАСК) по плановым периодам (ПП) месяцам (ПП–М) и декадам (ПП–Д) с учетом следующих ограничений по каждому ПП:

- на максимальную и минимальную загрузку автоматизированного технологического оборудования (АТО);
- на максимальный и минимальный годовой (месячный) объем запуска в шт. ЭЯ;
- на минимальный объем запуска ЭЯ в денежном выражении (рублях).

Второй этап отвечает за разработку сменных и суточных программ запуска ЭЯ в производство с учетом следующих ограничений по каждому ПП:

- на декадный объем запуска ЭЯ;
- на максимальную продолжительность непрерывной сборки ЭЯ одного типа;
- на максимально допустимый объем переменных затрат производства в ГАСК.

Этап объемного планирования

Задача распределения годовой производственной программы по плановым периодам относится к классу задач объемного планирования и решается характерными для него методами линейного программирования (ЛП). Целесообразность применения методов ЛП для решения задачи объемного планирования показана в работах [2...5]. При этом существенным моментом является выбор конкретного метода решения задачи ЛП (ЗЛП).

Наиболее часто для решения ЗЛП в приведенной постановке задачи применяются различные версии метода последовательных улучшений [7,8]. По критерию обеспечения высокого быстродействия при решении задач с большим числом переменных и ограничений в сочетании с экономией ресурсов памяти ЭВМ на основе сравнения эффективности

процедур метода последовательных улучшений [5,7,8] для решения данной задачи был выбран метод обратной матрицы. За основу взята процедура «MSimplex» [5,8].

Чтобы сделать организационную структуру ГАСК более гибкой, указанные методы в условиях реального производства применялись с учетом дополнений [2]. Все ЭЯ условно подразделяются на две группы – регламентированную Y^{per} и нерегламентированную Y^{nper} . К первой относятся ЭЯ, по которым предприятие не может самостоятельно изменять объемы и сроки запуска по ПП–М (ПП–Д). Ко второй группе относятся остальные ЭЯ, по которым предприятию предоставлена возможность их распределения по срокам и объемам запуска. При этом решение задачи объемного планирования проводится в два этапа.

На этапе I–а осуществляется директивное распределение программы запуска ЭЯ регламентированной группы в производство, на этапе I–б применительно к ЭЯ нерегламентированной группы применен симплекс–метод.

Этап I–а.

Расчет ведется для каждого m -го ПП–М и d -го ПП–Д в составе каждого ПП–М по всем операциям $\{\tilde{O}_z\}$ автоматизированного технологического процесса (АТП) сборки ЭЯ (кроме подготовительных операций).

Требуемые для выполнения регламентированной части программы запуска временные ресурсы (фонд времени АТО в составе ГАСК) на операции \tilde{O}_z определяются по формуле:

$$T_{пzm}^{per} = \sum_{\substack{i=1 \\ i \in Y_m^{per}}}^{N_a} (N_{эяmi} \cdot T_{штiz}) \cdot K_{п-3} \quad (1),$$

где $K_{п-3} = \frac{T_{шт} + T_{п-3}}{T_{шт}}$ – коэффициент затрат подготовительно–заключительного времени на операцию.

Располагаемые ресурсы операции \tilde{O}_z при $K_3 = 1$ определяются по формуле [6]:

$$T_{pzm} = N_{об.z} \cdot F_{м.эф.zm} \cdot K_{тп.z} \cdot \rho_z \quad (2),$$

где $N_{об.z}$, $K_{тп.z}$, ρ_z – соответственно количество единиц оборудования, коэффициент выхода годных и коэффициент готовности для АТО, назначенного на операцию \tilde{O}_z .

Рассчитывается фактический коэффициент загрузки АТО на операции \tilde{O}_z при выполнении регламентированной части программы:

$$K_{з.ф.zm}^{per} = \frac{T_{пzm}^{per}}{T_{pzm}} \quad (3).$$

Ограничения на загрузку АТО в составе ГАСК для операции \tilde{O}_z можно записать в виде системы двух неравенств:

$$\begin{cases} K_{з.ф.zm}^{per} \leq K_{з.zm}^{max} \\ K_{з.ф.zm}^{per} \geq K_{з.zm}^{min} \end{cases} \quad (4).$$

Сначала производится проверка отсутствия чрезмерной загрузки АТО на операции \tilde{O}_z :

$$K_{з.ф.zm}^{per} \leq K_{з.zm}^{max} \quad (4a).$$

Если условие (4а) не выполнено, дальнейший расчет становится невозможным. В противном случае проводится проверка попадания фактического коэффициента загрузки АТО на операции \tilde{O}_z в допустимый диапазон:

$$K_{з.ф.zm}^{per} \geq K_{з.zm}^{min} \quad (4б).$$

Если условие (4б) выполнено, то максимальное и минимальное значения требуемых временных ресурсов на выполнение нерегламентированной части программы запуска для АТО на операции \tilde{O}_z приравняются к 0:

$$T_{pzm}^{nper,max} = 0; T_{pzm}^{nper,min} = 0 \quad (5),$$

в противном случае они подлежат определению по формулам:

$$T_{pzm}^{перг. max} = T_{pzm} \cdot K_{3, jm}^{max} - T_{пzm}^{перг.}; \quad T_{pzm}^{перг. min} = T_{pzm} \cdot K_{3, jm}^{min} - T_{пzm}^{перг.} \quad (6).$$

Определяется также минимальное значение объема запуска ЭЯ в рублях:

$$C_m^{перг. min} = C_m^{min} - \sum_{\substack{i=1 \\ i \in Y_m^{перг}}}^{N_a} C_i \cdot N_{эя mi} \quad (7).$$

Этап I–б.

Задачей этапа является корректировка разработанной на этапе I–а программы запуска ЭЯ в производство для каждого ПП–М (ПП–Д) с учетом ЭЯ нерегламентированной группы. Решение проводится симплекс–методом обратной матрицы. Формализованная запись для ПП–М приведена ниже.

$$\left\{ \begin{array}{l} F(N_{кэя mi}) = \sum_{m=1}^{12} \sum_{i=1}^{N_a} c_{mi} \cdot N_{кэя mi} \rightarrow \max; c_{mi} = \begin{cases} 1 & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.} \end{cases} \\ \sum_{i=1}^{N_a} a_{mi,k} \cdot N_{кэя mi} \leq b_k; a_{mi,k} = \begin{cases} T_{шт iz} \cdot K_{п-3} & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.}; \end{cases} \\ b_k = \left[\frac{T_{пzm}^{перг. max}}{N_{пп k}} \right]; k = 1, 12 \cdot |Z|. \\ \sum_{i=1}^{N_a} a_{mi,k} \cdot N_{кэя mi} \geq b_k; a_{mi,k} = \begin{cases} T_{шт iz} \cdot K_{п-3} & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.}; \end{cases} \\ b_k = \left[\frac{T_{пzm}^{перг. min}}{N_{пп k}} \right]; k = 12 \cdot |Z| + 1, 2 \cdot 12 \cdot |Z|. \\ \sum_{i=1}^{N_a} a_{mi,k} \cdot N_{кэя mi} \geq b_k; a_{mi,k} = \begin{cases} C_i & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.}; \end{cases} \\ b_k = \left[\frac{C_m^{перг. min}}{N_{пп k}} \right]; k = 2 \cdot 12 \cdot |Z| + 1, 2 \cdot 12 \cdot |Z| + 12. \\ \sum_{m=1}^{12} a_{mi,k} \cdot N_{кэя mi} \geq b_k; a_{mi,k} = \begin{cases} 1 & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.}; \end{cases} \\ b_k = \left[\frac{N_{эя i}^{min}}{N_{пп k}} \right]; k = 2 \cdot 12 \cdot |Z| + 12 + 1, 2 \cdot 12 \cdot |Z| + 12 + N_a. \\ \sum_{m=1}^{12} a_{mi,k} \cdot N_{кэя mi} \leq b_k; a_{mi,k} = \begin{cases} 1 & \text{при } i \in Y_m^{перг.}; \\ 0 & \text{при } i \in Y_m^{перг.}; \end{cases} \\ b_k = \left[\frac{N_{эя i}^{max}}{N_{пп k}} \right]; k = 2 \cdot 12 \cdot |Z| + 12 + N_a + 1, 2 \cdot 12 \cdot |Z| + 12 + 2 \cdot N_a. \end{array} \right. \quad (8),$$

где: $N_{эя mi}$ – искомые программы запуска ЭЯ в производство по ПП–М; $|Z|$ – количество технологических операций в составе АТП, за исключением подготовительных; $Y_m^{перг.}(Y_m^{перг.})$ – множество индексов типов ЭЯ, входящих в нерегламентированную (регламентированную) группу в данном ПП–М; $N_{эя i}^{min}(N_{эя i}^{max})$ – минимально (максимально) допустимый годовой объем запуска ЭЯ i –го типа за вычетом регламентированной части программы запуска; $N_{кэя mi}$ – программа запуска в производство ЭЯ i –го типа в m –ом ПП–М, выраженная в партиях ЭЯ (ориентация идет на сборку партии ЭЯ, равной транспортной партии); $N_{пп k}$ – количество ЭЯ в транспортной партии; $N_{эя mi} = N_{кэя mi} \cdot N_{пп k}$.

Пример

На сборку запускаются ЭЯ пяти различных типов на базе ИС в корпусе 201, а также ЭРЭ с радиальными и осевыми выводами. $N_{эя i}^{ном} = 10000$, $N_{эя i}^{max} = 10500$, $N_{эя i}^{min} = 9500$ [ЭЯ/год], $C_i = 300$ [руб.]. В качестве АТО для монтажа ЭК всех указанных типов используется сборочный центр. Для каждого ПП–М: $D = 24$, $C_m = 90000$ [руб.], $C_m^{min} = 80000$ [руб.], $F_{м.эф. zm} = 420$ [ч.], $K_{3, zm}^{треб.} = 0.8$; $\Delta K_{3, zm}^+ = \Delta K_{3, zm}^- = 0.05$. Для каждого ПП–Д в составе ПП–М: $D = 8$, $C_{md} = 30000$ [руб.], $C_{md}^{min} = 27000$ [руб.], $F_{м.эф. zmd} = 140$ [ч.]. Результаты этапа I сведены в таблицу 1.

Таблица 1. Результаты этапа I – программы запуска ЭЯ в производство для ПП–М и ПП–Д

Месяцы	Декада 1, №ЭЯ [шт.]	Декада 2 №ЭЯ, [шт.]	Декада 3 №ЭЯ, [шт.]	Месяц, №ЭЯ [шт.]
Январь	3[4860]	3[1440], 4[1840]	4[2220]	3[6000], 4[3860]
Февраль	4[2260]	4[2500]	4[2220]	4[6640]
Март	2[60], 3[4720]	2[2480]	2[2200]	2[4520], 3[4500]
Апрель	1[740], 2[1580]	2[2480]	2[2200]	1[700], 2[5980]
Май	1[2520]	1[2800]	1[2480]	1[7440]

Июнь	1[2480], 5[20]	5[2120]	5[1880]	1[2360], 5[3840]
Июль	5[1880]	5[1880]	5[1820]	5[5320]
Август	5[100]	5[80]	5[80]	5[260]
Сентябрь	5[100]	5[80]	5[80]	5[260]
Октябрь	5[100]	5[80]	5[80]	5[260]
Ноябрь	5[100]	5[80]	5[80]	5[260]
Декабрь	5[100]	5[80]	5[80]	5[260]

В ПП–М «Январь»÷ «Июнь» для монтажного АТО $K_3 = 0,88$; для остального АТО в составе ГАСК $K_3 = 0,1 \div 0,2$, при этом объем запуска ЭЯ в рублях составляет $C = 1860 \div 2958$ [тыс. руб./месяц]. Из системы ограничений (8) удалось выполнить все, кроме ограничения на загрузку АТО снизу для всех операций, кроме монтажной, т.е. при обеспечении необходимой загрузки монтажного АТО остальное технологическое оборудование вынуждено простаивать в большей степени, нежели это было допущено в исходных данных. Такая ситуация является типичной, так как лимитирующей по длительности операцией в ГАСК в подавляющем большинстве случаев является монтажная. Учитывая максимальную концентрацию разнотипных переходов на данном монтажном рабочем месте, эта ситуация в нашем случае проявляется наиболее выпукло. Для ПП–М «Июль»÷«Декабрь» заданной годовой программы запуска не хватило для обеспечения адекватной загрузки АТО, поэтому коэффициент загрузки для всех без исключения операций в ГАСК не превышает 4%. При этом объем запуска ЭЯ в рублях составляет $C = 80$ [тыс. руб./месяц], что соответствует заданному минимальному значению. Суммарная программа запуска по всем ПП–М находится в допустимых пределах относительно заданной годовой программы ($\pm 5\%$), то же самое относится к суммарной программе запуска по всем ПП–Д относительно месячной программы.

Таким образом, на основании результатов этапа I можно сделать следующие выводы:

- **выполнено деление ЭЯ на регламентированную и нерегламентированную группы;**
- разработаны месячные и декадные программы запуска ЭЯ в производство по всем ПП–М и ПП–Д, полностью удовлетворяющие ограничениям на максимальный и минимальный годовой (месячный) объем запуска;
- **полностью выполнены ограничения на максимальную загрузку АТО;**
- **ограничения на минимальную загрузку АТО выполнены только для монтажного оборудования в некоторых ПП;**
- полностью выполнены ограничения на минимальный объем запуска ЭЯ в денежном выражении;
- подготовлены данные для проведения этапа II – календарного планирования.

При увеличении задаваемой годовой номенклатуры и программы запуска ЭЯ можно ожидать полного удовлетворения всех ограничений, поставленных на рассматриваемом этапе.

Решить задачу данного этапа ручным директивным распределением годовой программы запуска не представляется возможным даже при небольшой номенклатуре и программе запуска ЭЯ из-за значительной трудоемкости проводимых расчетов.

Литература

1. Петров В. А., Масленников А. Н., Осипов Л. А. Планирование гибких производственных систем. – Л.: Машиностроение, 1985. – 182 с.: ил.
2. Организация, планирование приборостроительного производства и управление предприятием: Учебник для студентов приборостроительных специальностей вузов /В. А. Петров, Л. П. Беликова, Э. В. Минько и др.; под общ. ред. В. А. Петрова. – Л.: Машиностроение, 1987. – 424 с.: ил.
3. Шкурба В. В. и др. Планирование и управление в автоматизированном производстве. – Киев, Наукова думка, 1985.

4. Семенов А. И., Португал В. М. Задачи теории расписаний в календарном планировании мелкосерийного производства. – М.: Машиностроение, 1972.
5. Первозванский А. А. Математические модели в управлении производством. – М.: Наука, 1975 – 616 с.: ил.
6. Иванов Ю. В. Повышение эффективности сборки изделий. – М.: Вестник МГТУ, №4,1995, с. 66 – 75.
7. Романовский И. В. Алгоритмы решения экстремальных задач. – М.: Наука, 1975.
8. Грибов А. Б., Романовский И. В. Программирование симплекс–метода и его вариантов на Алголе. В сб. «Оптимальное планирование», №12. – Новосибирск: Наука, 1969.

РАЗРАБОТКА И ОПТИМИЗАЦИЯ РАСПИСАНИЯ ЗАПУСКА В ПРОИЗВОДСТВО ЭЛЕКТРОННЫХ ЯЧЕЕК ПРИ КАЛЕНДАРНОМ ПЛАНИРОВАНИИ

Иванов Юрий Викторович, Курносенко Алексей Евгеньевич
к.т.н., доц. каф. ИУ-4 МГТУ им. Н.Э. Баумана Иванов Юрий Викторович
Иу4, Московский государственный технический университет им. Н.Э. Баумана

WORKING OUT THE CALENDAR PLAN OF ELECTRONIC BOARDS PRODUCTION LAUNCHING

Ivanov Jury V., Kurnosenko Alexey E.
candidate of tech. science, assistant professor, IU-4 department of BMSTU Ivanov Jury V.
Iu4, Bauman Moscow State Technical University.
mario7@online.ru

Аннотация

В статье рассматривается метод разработки расписания запуска электронных ячеек на сборку в условиях гибкого автоматизированного многономенклатурного производства. Метод включает этапы объемного и календарного планирования. Описывается второй этап, на котором решается задача разработки сменных и суточных заданий. Определяется оптимальная последовательность сборки электронных ячеек. Учитываются ограничения на декадный объем запуска электронных ячеек, продолжительность непрерывной сборки электронных ячеек одного типа, а также на объем переменных затрат производства по каждому плановому периоду.

Abstract

The method of working out the plan of launching the flexible automated production of electronic boards is discussed. The method contains of two main stages: total and calendar planning. On the second stage the problem of making shift and day production programs is worked out. Optimum sequence of electronic board assembling is defined. Time limitations on decade production program, uninterrupted assembling of electronic boards of the same type and limitations on the production expenses in each planning period are taken into account.

На первом этапе [см. статью Иванова Ю.В. и Курносенко А.Е. «Разработка расписания запуска в производство электронных ячеек (этап объемного планирования)»] решалась задача разработки программ запуска электронных ячеек (ЭЯ) в производство для гибкого автоматизированного сборочного комплекса (ГАСК) по плановым периодам (ПП) месяцам (ПП-М) и декадам (ПП-Д) с учетом следующих ограничений по каждому ПП:

- на максимальную и минимальную загрузку автоматизированного технологического оборудования (АТО);
- на максимальный и минимальный годовой (месячный) объем запуска в шт. ЭЯ;
- на минимальный объем запуска ЭЯ в денежном выражении (рублях).

Этап календарного планирования

Задача этапа сводится к разработке сменно-суточных заданий, реализующих задания и сроки на основе результатов первого этапа планирования с учетом следующих ограничений по каждому ПП:

- на декадный объем запуска ЭЯ;
- на максимальную продолжительность непрерывной сборки ЭЯ одного типа;
- на максимально допустимый объем переменных затрат производства в ГАСК.

Особенности постановки задачи календарного планирования для ГАСК сборки ЭЯ в условиях многономенклатурного производства с учетом указанных ограничений не позволяют в полной мере использовать для ее решения методы, описанные в литературе по машиностроительному производству, поэтому потребовалась разработка метода,

позволяющего учесть данную специфику. Решение задачи календарного планирования разработанным методом проводится в два рассмотренных ниже этапа.

Этап II-а.

Задачей данного этапа является определение оптимальной последовательности сборки ЭЯ из заданной номенклатуры по всем ПП-Д в составе каждого ПП-М.

Оптимальная последовательность сборки ЭЯ соответствует минимуму суммарного времени, затрачиваемого на переналадку АТО при сборке ЭЯ в данной последовательности. Рассчитывается суммарное время переналадок монтажного АТО $T_{пер\Sigma}$ и решается задача оптимизации последовательности переналадок АТО при сборке ЭЯ декадной номенклатуры по критерию $T_{пер\Sigma} \rightarrow \min$, которую можно сформулировать в виде задачи о назначениях [2]:

$$\sum_{i=1}^{n_{md}} \sum_{j=1}^{n_{md}} t_{ij} x_{ij} \rightarrow \min \text{ при } \begin{cases} \sum_{i=1}^{n_{md}} x_{ij} = 1; \sum_{j=1}^{n_{md}} x_{ij} = 1; \sum_{i=1}^k x_{ij}^{(k)} = 1; \sum_{j=1}^k x_{ij}^{(k)} = 1; j = 1, 2, \dots, n_{md}; x_{ij} = 0, 1 \forall i, j. \end{cases} \quad (1),$$

где: n_{md} – количество типов ЭЯ в декадной программе d -го ПП-Д m -го ПП-М;

$T = \|t_{ij}\|_{n_{md} \times n_{md}}$ – матрица переналадок АТО со сборки ЭЯ типа i на сборку ЭЯ типа j ;

$t_{ij} = t_{пер ij} = T_{сnp ij} + T_{ссг ij} + T_{сбчан ij} + T_{снп ij}$ – время переналадки АТО [4] со сборки ЭЯ типа i на сборку ЭЯ типа j ; включает время смены/наладки приспособления для установки и точной фиксации ПП $T_{сnp ij}$, сборочной головки или их комплекта $T_{ссг ij}$, базовой части накопителя ЭК $T_{сбчан ij}$, носителя программ $T_{снп ij}$.

$X = \|x_{ij}\|_{n_{md} \times n_{md}}$ – матрица, в которой

$$x_{ij} = \begin{cases} 1, \text{ если в маршрут входит переналадка от ЭЯ типа } i \text{ к ЭЯ типа } j; \\ 0 \text{ в противном случае.} \end{cases} \quad (2).$$

Решение задачи заключается в преобразовании матрицы X от начального состояния, заданного неоптимальной последовательностью переналадок, в конечное состояние, определяемое системой (1). Применяемый метод решения – метод многоугольника [3].

В результате решения задачи получаем последовательность индексов собираемых ЭЯ $l_1, l_2, \dots, l_k, \dots, l_{n_{md}}$ ($l_k \in \{1, n_{md}\}$), $l_i \neq l_j \forall l_i, l_j \in \{1, n_{md}\}$ такую, что $t_{l_1 l_2} + t_{l_2 l_3} + \dots + t_{l_{n_{md}-1} l_{n_{md}}} + t_{l_{n_{md}} l_1} \rightarrow \min$ (3).

Этап II-б.

Задачей данного этапа является разработка собственно сменно-суточных заданий, т.е. определение программ запуска ЭЯ в производство на ПП – сутки (рабочие смены), входящие в состав каждого планируемого d -го ПП-Д в составе m -го ПП-М. Математическая формулировка задачи для ПП – рабочей смены:

определить множество сменных программ запуска ЭЯ в производство

$$\{N_{\text{эя } mdysi} \} \forall i = \overline{1, n_{md}} \quad (4)$$

при условии

$$\begin{cases} T_{к.сб. i} - T_{н.сб. i} \leq [T_{непр.сб. i}] \forall i = \overline{1, n_{md}}; \end{cases} \quad (5)$$

$$\begin{cases} C_{y \text{ on} \Sigma} \leq [C_{y \text{ on} \Sigma}] \forall y \in D \text{ либо } C_{s \text{ on} \Sigma} \leq [C_{s \text{ on} \Sigma}] \forall s \in y, y \in D; \end{cases} \quad (6)$$

$$\begin{cases} \sum_{s,y} N_{\text{эя } mdysi} = [N_{\text{эя } mdi}] \forall i = \overline{1, n_{md}}, s \in y, y \in D \end{cases} \quad (7).$$

где: $T_{н.сб. i}$ ($T_{к.сб. i}$) – время начала (окончания) сборки партии ЭЯ i -го типа;

$[T_{непр.сб. i}]$ – максимальная длительность непрерывной сборки партии ЭЯ i -го типа;

$N_{\text{эя } mdysi}$ – программа запуска в производство ЭЯ i -го типа на смену s рабочего дня y ;

$[N_{\text{эя } mdi}]$ – декадная программа запуска в производство ЭЯ i -го типа в d -ом ПП-Д в составе m -го ПП-М, разработанная на I-ом этапе; s – номер смены в составе рабочего дня; y – номер рабочего дня в составе декады; D – множество рабочих дней в составе декады;

$C_{s \text{ on} \Sigma}$ ($C_{y \text{ on} \Sigma}$) [переменные затраты на производство данного сменного (суточного) задания в ГАСК] вычисляются в соответствии с методикой [4,5], при этом

$$C_{s\text{on}\Sigma} = \sum_{z=1}^{|Z|} C_{s\text{on}}(\tilde{O}_z); C_{y\text{on}\Sigma} = \sum_{k=1}^{n_{\text{см}}} C_{s\text{on}\Sigma k} \quad (8),$$

где: $|Z|$ – количество технологических операций в составе автоматизированного технологического процесса (АТП), за исключением подготовительных; $n_{\text{см}}$ – количество рабочих смен в рабочем дне при производстве в ГАСК.

Для решения задачи в данной постановке был разработан метод, который можно условно назвать методом поэтапных расчетов и согласований.

Суть метода заключается в последовательном заполнении ПП – смен и суток, начиная с первого. При этом очередь ЭЯ, кандидатов на запуск в каждом ПП (смене), соответствует оптимальной последовательности переналадок (3), полученная на этапе II–а. При каждом запуске нового типа ЭЯ в производство определяется ее программа запуска на данный ПП (порядок определения см. ниже) при условии выполнения ограничения (5). Если ограничение выполнено, ЭЯ запускается в производство в данном ПП, если нет – временно отвергается, но остается в очереди. Производится переход к ЭЯ следующего типа в последовательности (3). Ограничение (5) может не выполняться, если количество ЭЯ в очереди недостаточно для осуществления чередования сборки ЭЯ различных типов с одновременным выполнением ограничения (7) для всех типов ЭЯ. ЭЯ удаляется из очереди в случае выполнения для нее ограничения (7). Ограничения (6) проверяются после заполнения каждого ПП, по которому осуществляется контроль объема переменных затрат. В случае невыполнения условий (6) программа запуска для данного ПП последовательно уменьшатся до момента выполнения условий. При этом возможен возврат в очередь некоторых типов ЭЯ, ранее удаленных из нее в данном ПП.

Выше было сказано, что при каждом запуске ЭЯ в ПП – рабочей смене определяется ее программа запуска $N_{\text{эя mdysi}}$. Под данной процедурой подразумевается определение оптимального размера партии запуска ЭЯ данного типа. Для решения данной задачи был выбран критерий минимальной суммы квадратов отклонений времени сборки партии ЭЯ от фонда времени АТО в выбранном ПП [1]. Так как размер данной партии должен быть кратен размеру транспортной партии (ориентация идет на сборку партии ЭЯ, равной транспортной партии), запишем искомый размер партии $N_{\text{эя mdysi}}$ в виде $N_{\text{эя mdysi}} = K_{\text{тр. эя mdysi}} \cdot N_{\text{ппк}}$, и в дальнейшем будем искать значение $K_{\text{тр. эя mdysi}}$ ($N_{\text{ппк}}$ – количество ЭЯ в транспортной партии).

Функция суммы квадратов отклонений запишется в этом случае в следующем виде:

$$F_i = \sum_{z=1}^{|Z|} (K_{\text{тр. эя mdysi}} \cdot T_{iz} - L_{z\text{ mdys}})^2 \quad (9),$$

где: z – индекс операций \tilde{O}_z в составе АТП сборки ЭЯ; $T_{iz} = T_{\text{шт } iz} + t_{\text{неп } l_i, l_{i-1}}$ – время сборки транспортной партии ЭЯ типа i с учетом времени переналадки АТО $t_{\text{неп } l_i, l_{i-1}}$;

$L_{z\text{ mdys}} = F_{\text{пп.эф. zmd}} \cdot N_{\text{об } z} \cdot K_{3.z\text{md}}^{\text{треб.}} \cdot K_{\text{тп } z} \cdot \rho_z$ – сменный фонд времени группы АТО с учетом требуемого декадного коэффициента загрузки $K_{3.z\text{md}}^{\text{треб.}}$ на операции \tilde{O}_z [4].

Дифференцируя F_i по $K_{\text{тр. эя isy}}$ и приравнявая производную к 0, получим:

$$\frac{dF_i}{dK_{\text{тр. эя mdysi}}} = 2 \sum_{z=1}^{|Z|} (K_{\text{тр. эя mdysi}} T_{iz} - L_{z\text{ mdys}}) \cdot T_{iz} = 0 \quad (10).$$

$$K_{\text{тр. эя mdysi}} = \left[\frac{\sum_{z=1}^{|Z|} L_{z\text{ mdys}} \cdot T_{iz}}{\sum_{z=1}^{|Z|} T_{iz}^2} \right] \quad (11).$$

Пример

На сборку запускаются ЭЯ пяти различных типов на базе ИС в корпусе 201, а также ЭРЭ с радиальными и осевыми выводами. $N_{\text{эя } i}^{\text{ном}} = 10000$, $N_{\text{эя } i}^{\text{max}} = 10500$, $N_{\text{эя } i}^{\text{min}} = 9500$ [ЭЯ/год], $C_i = 300$ [руб.]. В качестве АТО для монтажа ЭК всех указанных типов используется сборочный центр. Для каждого ПП–М: $D = 24$, $C_m = 90000$ [руб.], $C_m^{\text{min}} = 80000$ [руб.], $F_{\text{м.эф. zm}} = 420$ [ч.],

$K_{3.zm}^{треб.} = 0.8; \Delta K_{3.zm}^+ = \Delta K_{3.zm}^- = 0.05$. Для каждого ПП–Д в составе ПП–М: $D = 8$, $C_{md} = 30000$ [руб.], $C_{md}^{min} = 27000$ [руб.], $F_{м.эф. zmd} = 140$ [ч.]. Для каждого ПП–рабочего дня в составе ПП–Д $F_{м.эф. zmd} = 17.5$ [ч.]. Для каждого ПП–рабочей смены в составе ПП– рабочего дня $F_{м.эф. zmd} = 7$ [ч.] (для смен 1,2), $F_{м.эф. zmd} = 3.5$ [ч.] (для смены 3). Результаты этапа II сведены в таблицы 1 и 2. В связи с большим объемом результатов приведем наиболее характерные из них – для 2–ой декады января в таблице 1 и для 3–го рабочего дня этой декады по сменам в таблице 2).

**Таблица 1. Результаты этапа II (декада).
(рабочий день).**

Рабочие дни	№ЭЯ [шт.]	$C_{y\text{оп}\Sigma}$, [руб.]	В % отн. [$C_{y\text{оп}\Sigma}$]	$K_{нез. y\text{оп}\Sigma}$, [руб.]
1	3[700]	48717. 3	97,4	2211.1
2	3[700]	48717. 3	97,4	2211.1
3	3[40], 4[320]	43061. 5	86,1	1267.2
4	4[340]	42655. 8	85,3	1209.8
5	4[340]	42655. 8	85,3	1209.8
6	4[340]	42655. 8	85,3	1209.8
7	4[340]	42655. 8	85,3	1209.8
8	4[160]	18132. 3	36,2	540.0

Таблица 2. Результаты этапа II

Рабочие смены	№ЭЯ [шт.]	$C_{y\text{оп}\Sigma}$, [руб.]	$K_{нез. y\text{оп}\Sigma}$, [руб.]
1	3[40], 4[120]	16240. 8	529.3
2	4[140]	15835. 0	472.0
3	4[60]	10985. 8	265.8

По приведенным результатам можно сделать следующие выводы:

- декадная программа, определенная на этапе I, выполнена полностью;
- ограничения на допустимый объем переменных затрат сборки ЭЯ по каждому рабочему дню выполнены, при этом уровень затрат в 7 днях из 8 не опускается ниже 85% от максимально допустимого значения, что приемлемо. Это стало возможным из-за достаточной загрузки монтажного АТО в данном ПП–Д (88%);
- ограничение на минимальную допустимую продолжительность сборки ЭЯ одного типа выполнить не удалось: ЭЯ №3 собирается в течение 3–х рабочих дней, ЭЯ №4 – в течение 6–ти. Невыполнение данного условия связано с ограниченной номенклатурой ЭЯ в программе запуска данной декады (всего 2 типа ЭЯ), но его можно выполнить при увеличении задаваемой годовой номенклатуры и программы запуска ЭЯ.

Ручное директивное распределение декадной программы запуска, в отличие от автоматизированного, отличалось значительной трудоемкостью проводимых расчетов и, кроме того, не позволило выполнить ограничения на допустимый объем переменных затрат сборки ЭЯ по рабочим дням №1 и 2 для рассмотренной выше 2–ой декады января (см. таблицу 3).

**Таблица 3. Результаты этапа II – ручное распределение декадной программы запуска
(рабочие дни №1,2)**

Рабочие	№ЭЯ	$C_{y\text{оп}\Sigma}$	В %	$K_{нез. y}$
---------	-----	------------------------	-----	--------------

е дни	[шт.]	[руб.]	отн. [$C_{y_{опΣ}}$]	$опΣ$, [руб.]
1	3[800]	59325. 0	118,7	2576.0
2	3[640] 4[60]	53320. 0	106,6	2257.7

Литература

9. Основы автоматизации управления производством/ Макаров И. М., Евтихеев Н. Н., Дмитриева Н. Д. и др./ Под ред. Макарова И. М. – М.: Высшая школа, 1983 – 504 с.: ил.
10. Дендобренко Б. Н., Малика А. С. Автоматизация конструирования РЭА: Учебник для вузов. – М.: Высш. школа, 1980. – 384 с.: ил.
11. Сигал И. Х. Комбинированные алгоритмы решения задачи коммивояжера. – М.: Вычислительный центр АН СССР, 1985. – 48 с.
12. Иванов Ю. В. Повышение эффективности сборки изделий. – М.: Вестник МГТУ, №4,1995, с. 66 – 75.
13. Иванов Ю. В. Методика автоматизированного выбора структур ГАСК по обобщенному критерию. – М.: МГТУ, 1999. – 65 с.: ил.

ПРЕДОБРАБОТКА РУКОПИСНОГО ТЕКСТА, ВВЕДЕННОГО С БУМАЖНОГО НОСИТЕЛЯ, ПРИ РАСПОЗНАВАНИИ

Цыганков Ю.В.

Научный руководитель профессор, д.т.н., заведующий кафедрой ВТ ТПУ Марков Н.Г.
Томский политехнический университет, Томск, Россия

PRECONDITIONING OF HANDWRITING TEXT, ACQUIRED FROM PAPER, IN RECOGNITION PROCESS

Tsygankov Y.V.

Instructor Prof. Nikolay G. Markov, Dr. Tech. Sci. Head of Computer Engineering Department

Tomsk Polytechnic University, Tomsk, Russia

Mail: tyv@ce.cctpu.edu.ru

Аннотация

Предлагается конечное принятие решения о классификации рукописных символов, введенных с бумажного документа, производить по дорабатываемому, в процессе принятия решения, растровому оригиналу этого символа. Весь процесс распознавания разделяется на четыре этапа: сегментация, выделение образа, предварительная обработка образа символа и конечная классификация. Описанная предварительная обработка символа, не имеет деструктивного свойства, т.е. после преобразования все линии имеют толщину в один элемент; преобразование не нарушает топологию символа (линиям и узлам исходного изображения соответствуют линии и узлы в преобразованном изображении); преобразование не нарушает основных размеров символа.

Abstract

The final decisionmaking about a classification of hand-written character, which entered from the paper document, is offered to make on variable raster of this character, during decisionmaking. The process of recognition is divided into four stages: segmentation, selection of an image, preprocessing of character image and final classification. The preprocessing of character, has no destruction, i.e. after transformation all lines have a thickness in one element; the transformation does not disturb topology of character (lines and to knots of an initial image there meet lines and knots in a transformed image); the transformation does not disturb main sizes of character.

В последнее время уровень бумажного документооборота значительно снизился, все большее количество организаций предпочитает электронную обработку информации. В то же время существует множество учреждений, в которых поступление информации извне затруднено без участия бумажных документов. Эта проблема особо актуальна в банках, налоговых инспекциях и других подобных учреждениях. В связи с этим обострилась проблема автоматического ввода и распознавания рукописной информации с бумажных документов.

В настоящее время технически не сложно преобразовать бумажный документ в цифровой растровый вид. Для этого производится множество разнообразных сканеров, цифровых фото и видео камер. Дальнейшее развитие исследований в этой области направлено лишь на нахождение способов более качественной передачи цветов и точной передачи деталей. Правда, еще не до конца решена проблема интеграции этих устройств с программными продуктами.

Если ввод бумажных документов в цифровой растровый вид можно считать решенной проблемой, то распознавание введенного документа все еще остается актуальной задачей. Хотя и созданы программы распознавания печатных документов с приемлемым качеством распознавания текста высокого и среднего качества, большинство разработанных систем дают большие погрешности при обработке текста низкого качества или текста с разной степенью яркости. Если же говорить о распознавании рукописного текста, то можно сказать, что результатов, удовлетворяющих практику, нам не известно.

подавляющее большинство теорий распознавания символов направлены на выделение признаков из исходного образа, принимая любую предварительную обработку как нечто деструктивное. Те теории, которые используют предобработку, проводят лишь внешние, косметические преобразования. Мы же считаем, что конечное принятие решения о классификации символа эффективнее производить по дорабатываемому в процессе принятия решения растровому оригиналу этого символа.

Весь процесс распознавания разделим на четыре этапа: сегментация, выделение образа, предварительная обработка образа символа и конечная классификация.

Сегментация изображения документа это выделение на введенном документе той области, в которой представлен текст, отделение текстовой информации от графической. Многие системы распознавания печатного текста позволяют вручную сегментировать оригинал, т.к. автоматическая сегментация часто не дает положительных результатов. Например, рассматривают некоторые рисунки как текст, и текст, сильно подверженный искажению, как рисунок. Выделение образа представляет собой процесс разделения слитно написанного текста, на отдельные символы. А результатом предварительной обработки образа символа должен являться некоторый набор признаков, необходимый для дальнейшего принятия решения об идентификации символа.

Все три описанных выше процесса не могут быть разделены. Несмотря на видимое разделение, в каждый отдельный момент времени ведется обработка по всем трем направлениям в сочетании с процессом распознавания. Сегментация необходима, как правило, для минимизации вычислительного процесса, но эта задача до сих пор не решена. Введение критериев разделения текста и рисунка невозможно, а значит, необходимо каждый отдельный блок подвергать процессу распознавания и лишь затем, при неудовлетворительных результатах, делать выводы о наличии рисунка. После выделения образа необходимо произвести фильтрацию изображения. Аддитивный шум проявляется на изображении как разрозненные изменения изолированных элементов, не обладающие пространственной корреляцией. Искаженные элементы часто весьма заметно отличаются от соседних элементов. Это наблюдение послужило основой для многих алгоритмов, обеспечивающих подавление шума [1-5]. Поскольку шум пространственно декоррелирован, в его спектре, как правило, содержатся более высокие пространственные частоты, чем в спектре обычного изображения. Следовательно, низкочастотная фильтрация может служить эффективным средством сглаживания шумов. Нами были исследованы многие фильтрующие алгоритмы. В результате проделанных опытных работ получилось, что использование любого из методов в чистом виде невозможно, т.к. фильтрация приводит к размыванию границ символов, а часто к стиранию отдельных, значимых, элементов. Для устранения нежелательных последствий необходим постоянный контроль результатов фильтрации, а также предварительное исследование образа, на предмет предположительных размеров символов и толщины «стенки» символа. Исследования привели к тому, что наиболее удобным и простым в реализации оказался высокочастотный фильтр[5].

Массив Q размера $M \times M$ выходного изображения формируется путем дискретной свертки массива F размера $N \times N$ исходного изображения со сглаживающим массивом H размера $L \times L$ согласно формуле:

$$Q(m_1, m_2) = \sum_{n_1} \sum_{n_2} F(n_1, n_2) H(m_1 - n_1 + 1, m_2 - n_2 + 1).$$

Как уже нами отмечалось классификацию целесообразно производить по дорабатываемому образу, а значит далее необходимо произвести выделение образа, предварительную обработку образа и затем выделение признаков для последующей классификации. Используемое нами признаковое пространство и алгоритм классификации символов требует представление образа в несколько идеализированном виде. Для этого необходимо применить следующие преобразования.

Утоньшение «стенок» символа и заполнение пустот. Процесс утоньшения эффективен не всегда, а при использовании некоторых алгоритмов выделения признаков, таких как образное наложение, вообще невозможен. При использовании же сегментирующих алгоритмов необходим. Процедуру утоньшения необходимо проводить под строжайшим контролем.

Под утоньшением будем понимать преобразование изображения, удовлетворяющее следующим условиям: после преобразования все линии имеют толщину в один элемент; преобразование не нарушает топологию символа, т. е. линиям и узлам исходного изображения соответствуют линии и узлы в преобразованном изображении; преобразование не нарушает основных размеров символа. Предложенный метод утоньшения обладает тем преимуществом, что он ориентирован на построчную обработку растра, а это увеличивает его быстродействие [1].

Рассмотрим три последовательные строки растра (i-1)-ю, i-ю и (i+1)-ю. Выделим в i-ой строке произвольный элемент, который обозначим через α . Этот элемент имеет в выбранных строках восемь соседних элементов, которые обозначим как элементы двумерного массива:

j-1	J	j+1
$\alpha(i-1,j-1)$	$\alpha(i-1,j)$	$\alpha(i-1,j+1)$
$\alpha(i,j-1)$	α	$\alpha(i,j+1)$
$\alpha(i+1,j-1)$	$\alpha(i+1,j)$	$\alpha(i+1,j+1)$

i-1
i
i+1

Основная идея процедуры утоньшения заключается в том, чтобы отыскать на изображении крайние сверху, снизу, справа и слева элементы, а затем вынести решение о возможности их удаления с соблюдением перечисленных выше условий. Естественно, элемент α считать крайним сверху, если он и элемент $\alpha(i+1,j)$ зачернены, а элемент $\alpha(i-1,j)$ незачернен. Формально эту ситуацию можно выразить так: элемент является крайним сверху, если равна единице следующая булева функция, в которой символы переменных совпадают с номерами элементов:

$$f_{\text{в}} = \overline{\alpha(i-1,j)} \wedge \alpha(i+1,j) \wedge \alpha$$

Далее, если данный элемент α является крайним сверху, то будем придавать ему значение 0 (будем стирать этот элемент), если равна единице следующая функция:

$$f_{\text{в}} = \overline{\alpha(i-1,j+1)} \wedge \alpha(i,j-1) \vee \overline{\alpha(i-1,j-1)} \wedge \alpha(i,j+1) \vee \alpha(i,j+1) \wedge \alpha(i,j-1).$$

Таким образом, элементу α будет придаваться нулевое значение при равенстве единице следующей функции:

$$g_{\text{в}} = f_{\text{в}} \wedge f_{\text{в}} = \overline{\alpha(i-1,j)} \wedge \alpha(i+1,j) \wedge \alpha(\overline{\alpha(i-1,j+1)} \wedge \alpha(i,j-1) \vee \overline{\alpha(i-1,j-1)} \wedge \alpha(i,j+1) \vee \alpha(i,j+1) \wedge \alpha(i,j-1))$$

Элемент растра является крайним слева, если равна единице функция $f_{\text{л}} = \overline{\alpha(i,j-1)} \wedge \alpha(i,j+1) \wedge \alpha$. Этот элемент подлежит стиранию, если функция $f_{\text{л}} = \alpha(i-1,j-1) \wedge \alpha(i+1,j) \vee \alpha(i+1,j-1) \wedge \alpha(i-1,j) \vee \alpha(i-1,j) \wedge \alpha(i+1,j)$ равна единице.

Следовательно, $g_{\text{л}} = \alpha(i,j-1) \wedge \alpha(i,j+1) \wedge \alpha(\overline{\alpha(i-1,j-1)} \wedge \alpha(i+1,j) \vee \overline{\alpha(i+1,j-1)} \wedge \alpha(i-1,j) \vee \alpha(i-1,j) \wedge \alpha(i+1,j))$.

Рассуждая таким же образом, получим g_n и g_p , определяющие подлежащие стиранию крайние нижние и крайние правые элементы растра:

$$g_n = \alpha(i+1,j) \wedge \alpha(i-1,j) \wedge (\alpha(i+1,j-1) \wedge \alpha(i,j+1) \vee \alpha(i+1,j+1) \wedge \alpha(i,j-1) \vee \alpha(i,j-1) \wedge \alpha(i,j+1)).$$

$$g_p = \alpha(i,j+1) \wedge \alpha(i,j-1) \wedge (\alpha(i+1,j+1) \wedge \alpha(i-1,j) \vee \alpha(i-1,j+1) \wedge \alpha(i+1,j) \vee \alpha(i+1,j) \wedge \alpha(i-1,j)).$$

Процедура утоньшения состоит в последовательном преобразовании исходного символа на растре в новый символ путем стирания сначала крайних сверху, затем крайних слева, далее крайних снизу и справа элементов. Этот цикл можно повторять до тех пор, пока на некотором цикле ни одна из функций не обратится в единицу. Но ввиду не замкнутости большинства символов это приведет к нежелательному стиранию некоторых элементов. Реализация алгоритма на ЭВМ показала, что для утоньшения требуется не более $l_{max}/2$ циклов, где l_{max} — максимальная толщина линии в исходном символе. Отметим, что порядок стираний может быть и иным, например, можно начать со стирания крайних правых элементов, затем крайних левых и т. д. Если α — i -я строка исходного символа на растре α_v , α_l , α_n и α_p — i -я строка символа, полученного стиранием соответственно верхних, крайних левых, нижних и правых элементов, то можно записать

$$\alpha_v = \alpha \oplus g_v \quad \alpha_l = \alpha \oplus g_l \quad \alpha_n = \alpha \oplus g_n \quad \alpha_p = \alpha \oplus g_p$$

Здесь символ \oplus обозначает операцию сложения по модулю 2.

Как уже отмечалось, процедуры утоньшения могут привести к изменению топологии символа, если внутри линий будут пустоты. Поэтому процесс заполнения пустот целесообразно предпослать утоньшению и стиранию бахромы. Как показали эксперименты, хорошие результаты при заполнении пустот дает следующее преобразование символа: элемент α полагается равным единице, если равны единице пять из восьми соседних с ним элементов.

На рис. 1 показаны отдельные символы, введенные с помощью сканера, и результаты работы описанной предобработки. Как видно, результаты удовлетворяют поставленным требованиям.



Рис. 1 Результат описанной предобработки

В результате обработки описанными выше алгоритмами, к сожалению, не всегда получаем сегментированный символ, дальнейшая сегментация производится в процессе выделения признаков и распознавания.

Литература

1. Купчинаус С.В. «Алгоритм утоньшения изображений» межвузовский сб. дискретные системы обработки информации, вып 2 - Ижевск: Ижевск. Мех.институт. 1979, с 23-27
2. Губерман Ш.А. Розенцвейг В.В. «Алгоритм распознавания рукописных текстов» // Автоматика и телемеханика, 1976, №5
3. Сунь Ц.Ю., Берто М., Мори С. «Автоматическое распознавание рукописных символов» ТИИРЭ, т.68, №4, 1980, с. 44-65
4. Фор А. «Восприятие и распознавание образов» /Пер. с фр./-М.: Машиностроение, 1989, 272 с.
5. Виттих В.А., Соيفер В.А. «Обработка изображений в автоматизированных системах научных исследований» М.: Наука, 1982, 216с.

ИМПЕРИЯ СЧИСЛЕНИЙ
А. Ю. ПОТЁМКИН
МГТУ ИМ.Н.Э.БАУМАНА
КАФЕДРА ИУ4 «КОНСТРУИРОВАНИЕ И ТЕХНОЛОГИЯ ПРОИЗВОДСТВА ЭВ И
ТС»

EMPIRE of CALCULATIONS

A.Potemkin
BMSTU, IU4

Аннотация: В работе рассмотрены математические модели, алгоритмы и программная реализация системы проведения вычислений произвольной размерности.

The summary: In activity the mathematical models, algorithms and programmatic implementation of a system of realization of calculus of arbitrary dimension are reviewed.

ВЕДЕНИЕ

Вот уже на протяжении двух с половиной лет года я пишу свою программу, каждый день уделяя этому по несколько часов и отдавая всё своё свободное время. И вот, наконец, настало время, чтобы другие люди оценили мой труд.

Представляю проект "Империя Счислений", написанный под платформу Wintelx32 (8686) для операционной системы Windows 98. В своём далёком прошлом программа имеет своего предшественника – версию под DOS, которая уже презентовалась на районной конференции восточного округа Москвы и заняла место лауреата. В то время программа имела примитивный интерфейс, но сейчас проект презентуется с в несколько раз усовершенствованной внутренней и внешней структурой. Продукт занял 9 место на третьей межрегиональной всероссийской международной межотраслевой научной олимпиаде «Шаг в будущее, Москва», проходящей в МГТУ имени Баумана среди исследователей в области информационных технологий Москвы и Московской области.

С восьми лет я учился программировать на компьютере, а потом решил собрать все знания воедино и посвятить несколько лет своей жизни разработке программного проекта Империя Счислений. Но и на сегодняшний день далеко нельзя сказать, что я закончил свой проект. Настоящая же программа – логически завершённая бета версия будущего продукта. Я вложил в свою программу все силы и работал как днём, так и ночью.

2 Используемые Средства

Я использовал язык, который считаю самым лучшим, высокопрофессиональным и оптимальным – Visual C++ 6.0 (Microsoft Visual C++ Enterprise Edition 6.0 Service Pack 3). При разработке использовались новые возможности современных технологий AppWizard, ClassWizard, ActiveX, MFC, COM, OLE, API, ALT, SEH, MRU.

3 Поставленные цели

Изначально меня привлекали идеи программы не с красивыми картинками, а со сложной внутренней логической структурой. И я решил написать программный проект, дающий возможность выполнять такие вычислительные операции, которые очень нужны в науке и жизни с одной стороны и не выполняются никакой другой программой во всём мире (во всяком случае, никто не знает таких программ, MatLab на это не способна) с другой стороны. И вот Империя Счислений – это программа, способная выполнять вычислительные операции над числами неограниченной длины, произвольного экспоненциального показателя, как классические: сложение, вычитание, умножение, деление... так и очень даже изощрённые: поразрядное дополнение до единицы, точное дополнение до двух, противоположенное преобразование, экспоненциальное наложение и преобразование... а

главное – конвертация любого числа из любой системы счисления в любую другую. Здесь не накладывается никаких ограничений ни на размер числа, ни на его экспоненциальный показатель (позицию плавающей запятой), ни на фактический и формальный знак, ну разве только система счисления числа не может превышать $1d+5000$. Если назвать Империю Счислений калькулятором, то для неё не существует такого понятия "разрядность", потому что она бесконечна.

Также Империя Счислений может выполнять дополнительные операции над числами, упрощающими и ускоряющими задание исходных чисел, например, упрощение числа, ввод числа в экспоненциальной форме, обращение фактического знака числа...

4 Проблемы

У операционной системы Windows x32 максимальная разрядность вычислительных целочисленных операций – 64 бита, а с плавающей запятой – 80 бит, очевидно, что это накладывает большие ограничения на размеры чисел, следовательно, если речь идёт о числах, экспоненциальные показатели которых превышают 19, то здесь уже операционная система даёт приблизительный результат вычислений.

5 Проектирование Алгоритма

В отличие от операционной системы Империя Счислений действует по другому принципу: имитируя вычисления, она содержит в себе два драйвера – `drive_strings` и `drive_values`, позволяющих производить вычисления с числами неограниченных размеров. Драйвер `drive_strings` отвечает за хранение чисел в оперативной памяти, но ведь число может не уместиться в ней, в этом случае драйвер кэширует дисковое пространство для числа и управляет вводом – выводом, используя какую – то часть памяти, тогда число может быть гигабайтного размера. Логическое распознавание числа, хранимого вышеописанным драйвером, производит драйвер `drive_values`, способный имитировать все вычисления и представлять их через 3 элементарные операции – сложение, вычитание и умножение двух цифр. Таким образом, любой файл, также как любая последовательность символов может быть рассмотрен, как число, если конечно известна его система счисления и значение каждой цифры.

Никакие ранее созданные алгоритмы конвертации чисел не устроили меня, и мне пришлось создать новый алгоритм, универсальный для всех систем счисления, для любых экспоненциальных показателей чисел и знаков. Разработка его стоила мне большого труда. Старые бета – версии программы работали по другим, менее оптимальным, алгоритмам, также разработанным мною. Настоящий же алгоритм я назвал "Экспонентальным" (производится через экспонентальный логарифм). При разработке очень много времени было уделено оптимизации программного кода, в частности алгоритм программы специально рассчитан под процессоры корпорации Intel с технологией Pro, введена система буферизации ввода – вывода данных на диск...

Описать алгоритм Империи Счислений здесь совершенно не реально, ведь тысячи и тысячи строк исходного кода, составляющие её, не уместились бы даже в сотни листов печатного текста. Скажу лишь общий принцип "Экспонентального" алгоритма программы (он активизируется на этапе непосредственной конвертации, описанном ниже). Сначала вычисляется экспоненциальное приближение числа (дающее возможность оценить его порядок), далее просчитывается простое приближение, затем снова вычисляется экспоненциальное приближение, но уже с большей точностью, и таким образом, когда более точное приближение не отличается от последнего приближения, получается результат. В процессе вычислений работает очень большое количество функций, поочерёдно передающих друг другу управление, максимальная степень вложенности превышает десять вызовов функций. При разработке алгоритма программы широко использовались рекурсия, инвариантность, виртуальность, полиморфизм, абстрагирование, инкапсуляция,

контейнерность, дружелюбность, итераторность и другие возможности объектно – ориентированного программирования.

Империя Счислений интегрирует в себе 2 алгоритма конвертации – это описанный выше алгоритм, имитирующий все вычислительные операции с использованием драйверов `drive_strings` и `drive_values`, и алгоритм, с использованием возможностей операционной системы и с ограничением на размеры чисел. Алгоритм вычислений можно выбрать в опциях программы или указать в `ini` – файле.

6 Практическое Применение

С первого взгляда кажется, что конвертация чисел мало где может использоваться, но это далеко не так. В недалёкой перспективе я запланировал программный проект "Империя Архиваций", который, унаследовав возможности "Империи Счислений", будет архивировать файлы с помощью конвертации. Алгоритм прост – перекодировать заданный файл из его текущей системы счисления (определяющейся по количеству вариаций встречающихся в файле символов) в 256 – ричную систему счисления, и размер файла значительно сократится. Также с помощью конвертации чисел можно легко и совершенно нераспознаваемо шифровать информацию. Ведь любая последовательность символов, изменившая свою систему счисления хотя бы на единицу, становится неузнаваемой.

7 Интерфейс

В итоге я создал две вещи – это вычислительный алгоритм для конвертации чисел и полноценное Windows – приложение, способное обрабатывать все сообщения, имеющее удобный интерфейс, предоставляющее много сервиса.

Я не буду детально описывать интерфейс моей программы, подробную документацию Вы найдёте в справочной системе в комплекте продукта. Изложу лишь только основные положения. Империя Счислений имеет 21 диалоговых окон (без учёта стандартных окон), обеспечивающих сервис и многофункциональность программы, причём меню главного окна состоит из более 150 пунктов. Среди этих окон – специализированные редакторы конвертируемого числа и счислительных цифр (о них ниже), бинарный редактор, опциональные окна, резидентные контролеры, мастера...

В качестве исходных параметров Империя Счислений предлагает ввести число для конвертации (либо файл, содержащий его), систему счисления этого числа, систему счисления, в которую его следует сконвертировать и счислительные цифры (последовательность цифр, определяющая значение каждой используемой цифры (в качестве стандарта предлагается: "0123456789ABCDEFGHIJKLMNOR
QRSTUVWXYZabcdefghijklmnopqrstuvwxyzБГДЁЖЗИЙЛПУФЦШЩЪЫЬЭЮЯбвгдёжзийк
лмнпфцщъыьэюя")).

Счислительные цифры позволяет ввести специализированный редактор и бинарный редактор.

Процесс конвертации делится на четыре этапа: инициализация, генерация объектного кода, конвертация и финальные операции. На инициализации производится распознавание числа: определяется позиция запятой, считывается информация о знаке и создаётся объектный код, специфический для системы счисления конвертируемого числа, описывающий его, для дальнейших процессов вычислений. На этапе генерации временного объектного кода генерируется другой код на основе уже созданного кода, дающий исчерпывающую информацию о числе, универсальную для всех систем счисления и необходимую для непосредственной конвертации. На конвертации из объектного кода конвертируемого числа создаётся объектный код результата конвертации, этот процесс, как правило, самый долговременный и сложный. На этапе финальных операций производятся экспоненциальные счисления, если это указано в опциях (вычисляются дополнительные научные сведения о результате операций, например, научная нотация), вычитывается противоположенное преобразование (также в зависимости от опций) и из объектного кода

результата конвертации генерируется сам результат. Информацию в реальном времени о процессах конвертации и их процентах выдаёт специализированный "резидентный модуль", также показывающий время, затраченное на операции, и количество вычислительных тактов, проведённых для получения результата.

По окончании конвертационно – вычислительных операций выводится имя файла, содержащего результат, или сам результат (в зависимости от настроек), который можно просмотреть, как в текстовом, так и в бинарном формате (используя бинарный мастер). Бинарный мастер выводит результат операций, как последовательность кодов в 2, 8, 10, 16 и 256 – ричных системах счисления, где каждый код представляет собой цифру. Вы можете сохранить файл, связанный с результатом вычислений, сохранить его под выбранным именем или же определить этот файл, как временный, тогда он автоматически удалится по окончании работы программы. Система межмодульного обмена, интегрированная в Империю Счислений, позволяет просмотреть этот файл внешними редакторами. Также предоставляется возможность получить дополнительную научную информацию о результате конвертации, в частности научную экспоненциальную нотацию в системе счисления результата и противоположенное преобразование, которое так же можно сохранить.

8 Обработка Исключений

Империя Счислений интегрирует в себе "резидентный обработчик исключений", который в реальном времени контролирует корректность вычислительных операций, и в случае ошибки или при возникновении исключительной ситуации обработчик останавливает процесс вычислений, выдаёт соответствующее сообщение и предоставляет возможность пользователю выбрать выход из ситуации: обработка исключения, игнорирование, игнорирование всех ошибок, перезапуск приложения, выход из программы, завершение программного кода, установка настроек по умолчанию, остановка. Кроме того есть дополнительно 3 обработчика ошибок, которые вызываются в зависимости от фатальности исключений. Общаются ошибки по технологии SEH с использованием глобальной раскрутки стека вызовов.

9 SDI Концепции

Программа является SDI – приложением, при инсталляции мастер установки регистрирует тип файлов "eoc" (от английского названия "Empire Of Calculations") как "Числа Империи Счислений". Поддержаны почти все стандартные операции над файлами, такие как создание, открытие, загрузка и сохранение файлов. Во время работы программы открыто 15 файлов – это файл, связанный с исходным числом (в нём всегда содержится заданное число для конвертации), файл, связанный с вторичным аргументом (для бинарных операций), файл, связанный с результатом операций, файл противоположенного преобразования и файл объектного кода, а также 10 файлов – кэш буферов вычислительного драйвера. Империя Счислений предоставляет возможность межмодульного обмена между приложениями DOS и Windows, например, можно редактировать число для конвертации или просмотреть результаты операций внешними редакторами (Блокнот, WordPad, Word, Norton commander...). Впрочем, программа имеет встроенные редакторы конвертируемого числа, мастер технических подробностей, бинарный мастер просмотра результата... Открыть файл Империей Счислений можно также, щелкнув два раза мышкой по файлу "eoc", после инсталляции или выбрав Империю Счислений в окне "Открыть с помощью" операционной системы, или через командную строку, или через запрос на открытие от другой программы, либо воспользовавшись меню отправить.

Бинарный редактор, например, позволяет ввести каждую цифру конвертируемого числа, как код в 2, 8, 10, 16 и 256 – ричной системе счисления. Таким образом, Империя Счислений также является текстовым и бинарным редактором файлов.

10 Возможности Настройки

Империя Счислений имеет много возможностей настройки, которые записываются в системный реестр и ini – файл, в частности можно отключить поддержку плавающей запятой, отрицательных чисел, резидентный обработчик исключений, резидентный модуль, визуальное анимирование, задать размер оперативной памяти, буферизируемой для вычислений и путь дискового пространства для кэширования, можно отрегулировать приоритет вычислительного процесса, указать его скорость... Автоматически создаётся список последних открытых файлов по технологии MRU, позволяющий быстро перейти к недавно использованному файлу.

11 Справочная система

В комплекте продукта состоит справочная система со встроенным мастером поиска, содержанием, указателем, контекстной справкой, советами дня и справкой о пользовании справкой. Нажав правую кнопку мышки на любом элементе Империи Счислений, пользователь немедленно получает соответствующую краткую информацию, а при нажатии F1 выдаётся справка об активном элементе. Режим ознакомления программы помогает начинающим пользователям, выдавая информации обо всех элементах по которым был произведён щелчок мышью.

12 Инсталляционная утилита

Инсталлирует программу специальный мастер установки, позволяющий выбрать тип установки, комплект, директорию назначения или место в локальной сети и соответственно перенастроить системный реестр, рабочий стол и меню "Пуск" пользователя. Полный комплект установки составляет 7 мегабайт. Деинсталлировать продукт можно через приложение "Установка и удаление программ" на панели управления или ярлык в меню "Пуск".

13 Перспективы

В перспективе Империя Счислений будет уметь совершать арифметические операции с числами любых размеров и систем счисления, например, извлечение логарифма с основанием в любой системе счисления из числа любой системы счисления. В итоге я хочу сделать Империю Счислений уникальной вычислительной машиной.

14 Это Интересно...

Проект Империя Счислений разрабатывался два с половиной года и, по – видимому, будет разрабатываться ещё дольше. В это сложно поверить, но порой месяцами каждый день я в среднем уделяю программированию 4 – 5 часов, что практически составляет всё моё свободное время. Объектно – ориентированная структура программы содержит 27 классов (исключая классы MFC) и более 1000 объектов (очень сложно было бы сосчитать), включая функции (исключая библиотечные объекты и функции API). Директория с исходными кодами моего проекта занимает около 50 мегабайт памяти и содержит 816 файлов (без учёта предопределённых библиотек), среди них 28 файлов типа ".cpp" и 30 библиотек ".h", а самый главный ".cpp" файл программы инкапсулирует более 150 экранов текста (150000 знаков или 114 страниц А4 14 шрифтом). Исходные кода Империи Счислений лишь только в простом неформатированном текстовом формате занимают около 800 Кб. Причём все эти величины со временем динамически растут.

15 Заключение

...Но всё – таки все возможности Империи Счислений можно узнать лишь только методом полной практической индукции.

ВИРТУАЛЬНЫЙ МУЗЕЙ «ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»

Колосков Дмитрий Владимирович

Научный руководитель: профессор, д.т.н. Белов Б.И.

Физико-математического лицея №1580 при МГТУ им. Н.Э.Баумана

koloskov@iu4.bmstu.ru

VIRTUAL MUSEUM "A HISTORY OF DEVELOPMENT OF COMPUTER FACILITIES"

Koloskov D.V.

The scientific chief: the professor, Dr. Belov B.I.

Physical and mathematical Liceum №1580 at BMSTU

koloskov@iu4.bmstu.ru

Аннотация: В работе рассмотрены инструментальные средства, технологии и методика создания и администрирования информационно - справочной интернет системы по истории вычислительной техники, раздел "От Абака до компьютера".

The summary: In activity the tools are reviewed, technologies both the technique of creation and management by orders and decrees is informational - reference Internet of a system on a history of computer facilities, section "From an Abacus up to the computer".

Введение

За последние несколько лет в российском интернете появилось огромное количество школьных страничек, информационно-образовательных и развлекательных порталов, научных, технических проектов и просто домашних страничек, сделанных руками школьников.

Сегодня каждый из нас работает за высокопроизводительными компьютерами, часто объединенными в локальные сети и имеющие выход в интернет. Мы используем современные принтеры, сканеры, плоттеры, модемы и многие другие цифровые устройства – последние достижения науки и техники. Однако, мы не часто задумываемся об истории и развитии вычислительной техники, их создателях, в чьих умах рождались гениальные идеи построения вычислительных машин, облегчивших бы рутинный труд человека, помогшей бы многим ученым сделать следующий шаг в их деятельности.

Обзор проектов по виртуальным компьютерным музеям

Одно из направлений развития музейных экспозиций в Internet – появление виртуальных музеев информатики и вычислительной техники. Реально эти музеи могут и не существовать, однако их виртуальные экспозиции насчитывают многие экспонаты, фотографии и биографии известных людей.

Во многих странах Европы, США, Канады, Австралии существуют компьютерные музеи, как реальные, так и виртуальные. В них можно найти информацию, как и о первых вычислительных машинах, так и о современных технологиях. На их web сайтах можно подробно узнать о биографии ученых, выдающихся разработчиков и программистов, президентов крупнейших компьютерных компаний, сведения по архитектуре того или иного компьютера. Большинство музеев предлагают информацию на английском языке.

Интерес к виртуальным музеям постоянно растет и поэтому посещаемость сайтов увеличивается также быстро. Так, например, web сайт, содержащий каталог компьютерных музеев и выставок – The Virtual Museum of Computing (VMoC) (<http://nic.icom.org/vlmp/computing.html>) посещают ежедневно более трехсот человек.

Один из самых популярных сайтов, посвященных развитию вычислительной техники – это сайт калифорнийского колледжа (<http://www.computer-museum.org/>), который содержит богатую экспозицию как "настоящих", так и виртуальных экспонатов.

Также популярны музеи, посвященные выдающимся личностям, внесшие большой вклад в развитие вычислительных технологий.

На европейских серверах есть музеи, посвященные Чарльзу Бэббиджу (<http://www.comlab.ox.ac.uk/oucl/users/jonathan.bowen/babbage.html>), Аде Лавлейс (<http://www.cs.yale.edu/homes/tap/ada-lovelace.html>).

Среди виртуальных компьютерных музеев рунета крупнейшим является – “Компьютерный виртуальный музей” PC Week/RE (<http://www.computer-museum.ru>). Посещая его страницы создается впечатление, что здесь можно найти абсолютно все: история развития ВТ, программного обеспечения, вычисления в докомпьютерную эпоху.

Проект "История вычислительной техники"

Представленная мною работа носит название "История вычислительной техники" и размещена по адресу (<http://museum.iu4.bmstu.ru>) – на web сервере кафедры «Конструирование и производство электронной аппаратуры» МГТУ им. Н.Э.Баумана, занимающейся непосредственно разработкой и созданием ЭВМ с 1938 года. Цель создания виртуального музея – появление русскоязычного ресурса, в котором бы удалось сконцентрировать те необходимые сведения, которые нужны любому ученику, учителю информатики, желающим познакомиться с интересующей их информацией о развитии вычислительной техники в прошлом и современных информационных технологиях.

В проекте «История развития вычислительной техники» была сделана попытка по шагам проследить развитие вычислительной техники начиная от абака – дощечки покрытой слоем пыли, на которой острой палочкой проводились линии и какие-нибудь предметы, размещавшиеся в полученных колонках по позиционному принципу, и заканчивая современными микропроцессорами, компьютерами и серверными платформами.

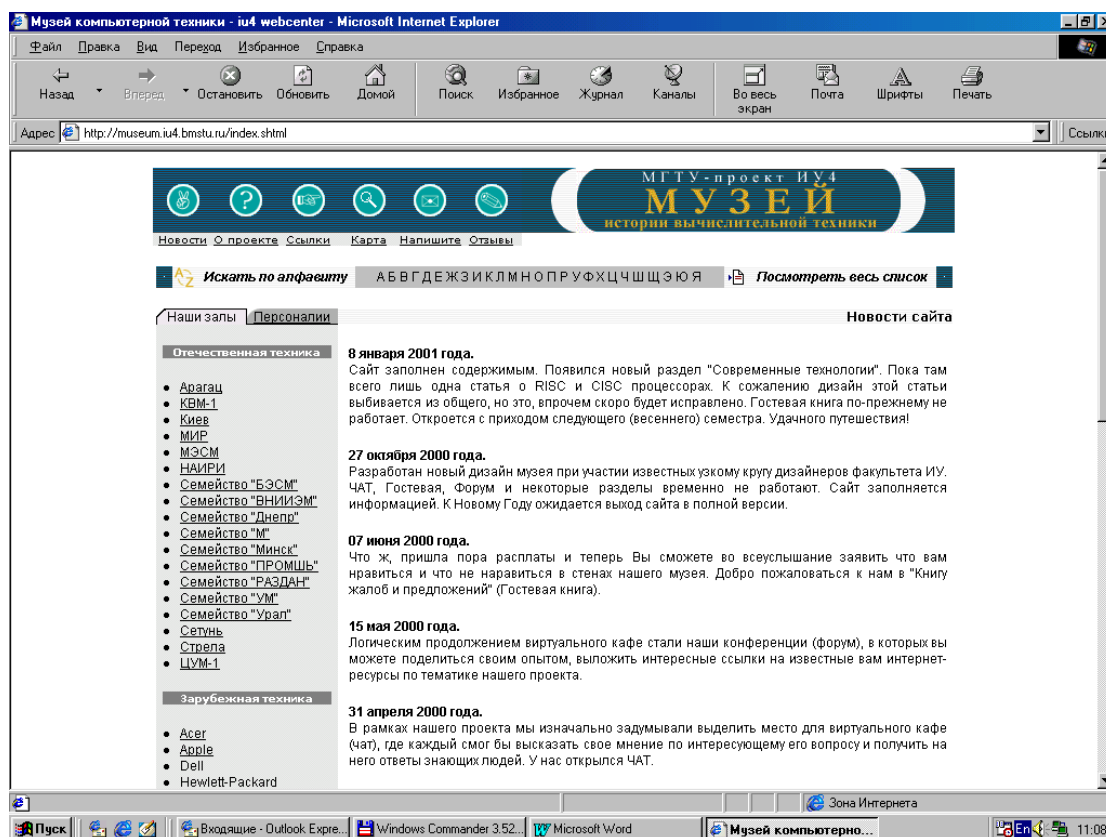


Рис.1 Web-сайт виртуального музея истории вычислительной техники

На страницах виртуального музея можно найти фотографии и описания арифметической машины Паскаля (Паскалево колесо), арифмометра Полени, машины

Бэббиджа, вычислителя Лейбница, созданных с небольшим промежутком времени с середины XVII века.

Очень интересен раздел, посвященный программируемой механической вычислительной машине Чарльза Бэббиджа. Начиная с 1812 года Ч. Бэббидж работал над изготовлением механической вычислительной машины более 30 лет. "Элементной базой" машины Бэббиджа служили "цифровые" колеса с различным числом зубьев. Созданная в 1834 году Чарльзом Бэббиджем аналитическая машина стала первым прообразом современных компьютеров.

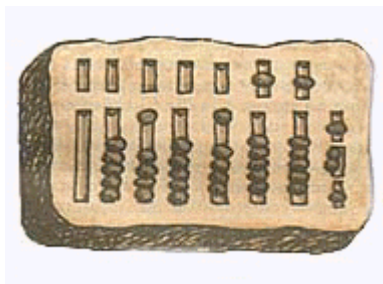


Рис.2. Абак

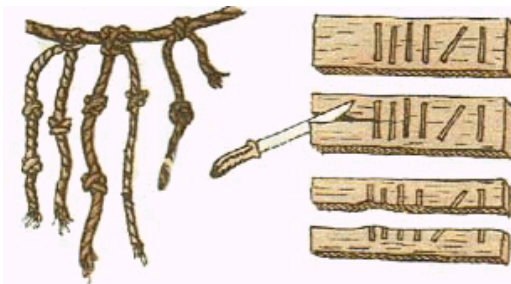


Рис.3. Узелки

Большое внимание уделено ЭВМ, созданным в середине XX века. Первая ЭВМ – "Эниак" была создана в США в 1948 году. Она была очень специализирована и выполняла ограниченное число задач. Первые отечественные ЭВМ данного поколения были созданы немного позднее. К ним относятся такие ЭВМ, как "БЭСМ-1", "Урал-1" и др.

Бурное развитие полупроводниковой технологии позволило в начале 70-х годов перейти к созданию больших и сверхбольших интегральных схем (БИС и СБИС), на основе которых были созданы ЭВМ, производящие десятки миллионов операций в секунду. Одними из самых мощных ЭВМ данного поколения являются ЭВМ семейства "Крейт" (США) и "Эльбрус" (Россия), которые являются многопроцессорными системами и содержат до нескольких миллиардов транзисторов.

Говоря о развитии вычислительной техники нельзя не сказать о программном обеспечении, без которого любой компьютер – просто кусок железа. Начиная с 60-ых годов особое внимание уделяется разработкам программного обеспечения. Появились языки программирования: Fortran, LISP, LOGO, Pascal. Разработана операционная система UNIX. Возникает потребность объединения отдельных компьютеров в сеть. В разделе "Персонали" отмечены ученые, внесшие наибольший вклад в создание и развитие вычислительной техники. Среди них Джон Буль, Чарльз Бэббидж, Норберт Винер, Джон Фон Нэйман, Лавлейс Ада Аугуста, Готфрид Вильгельм Лейбниц, наши соотечественники: Сергей Алексеевич Лебедев, Алексей Андреевич Ляпунов, наши современники: Питер Нортон, Билл Гейтс, Дмитрий Лозинский.

Последняя экспозиция виртуального музея – современные достижения компьютерных технологий – разработки конца 90-х годов, представлены разделы по CISC и RISC архитектурам микропроцессоров.

Сайт имеет простой, спокойный дизайн с удобной продуманной навигацией по структурной схеме типа «дерево». Имеющиеся виртуальные экспозиции разбиты на разделы: залы музея и персонали. На сегодняшний день разработаны четыре зала музея: отечественная техника, зарубежная техника, от абака до компьютера, CISC и RISC архитектуры микропроцессоров. Есть навигационное меню, помогающее ориентироваться по хронологической последовательности развития ЭВМ, приведены ссылки на другие виртуальные музеи мира по информационным технологиям.

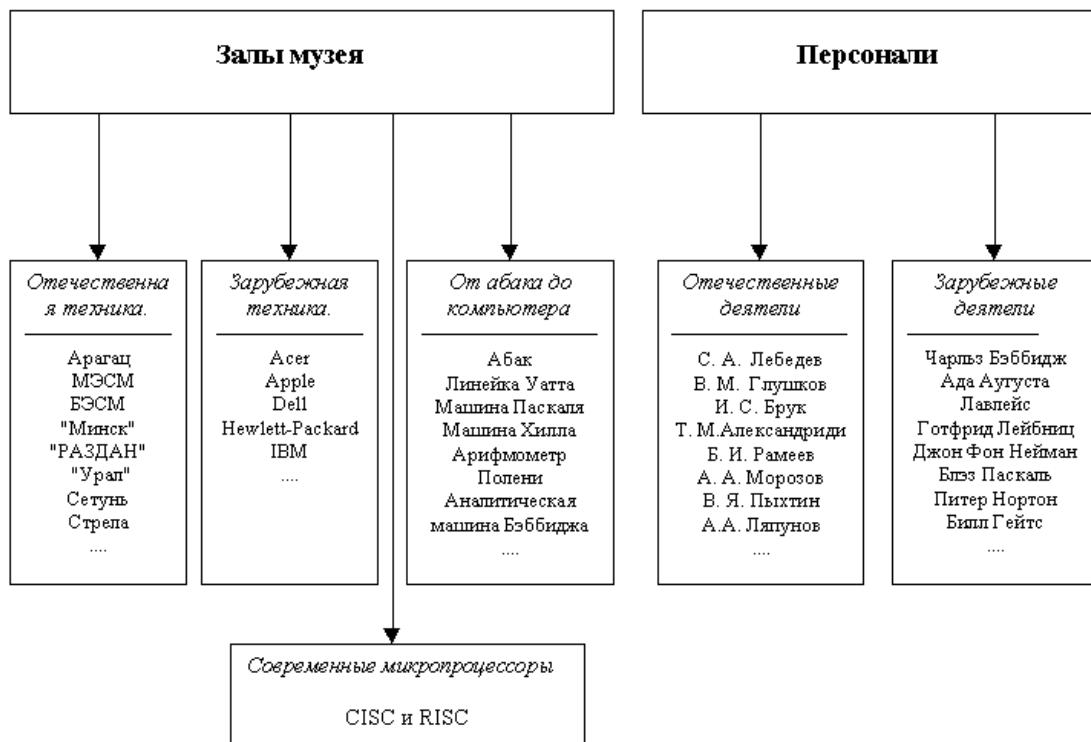


Рис.4. Структурная схема построения виртуального музея.

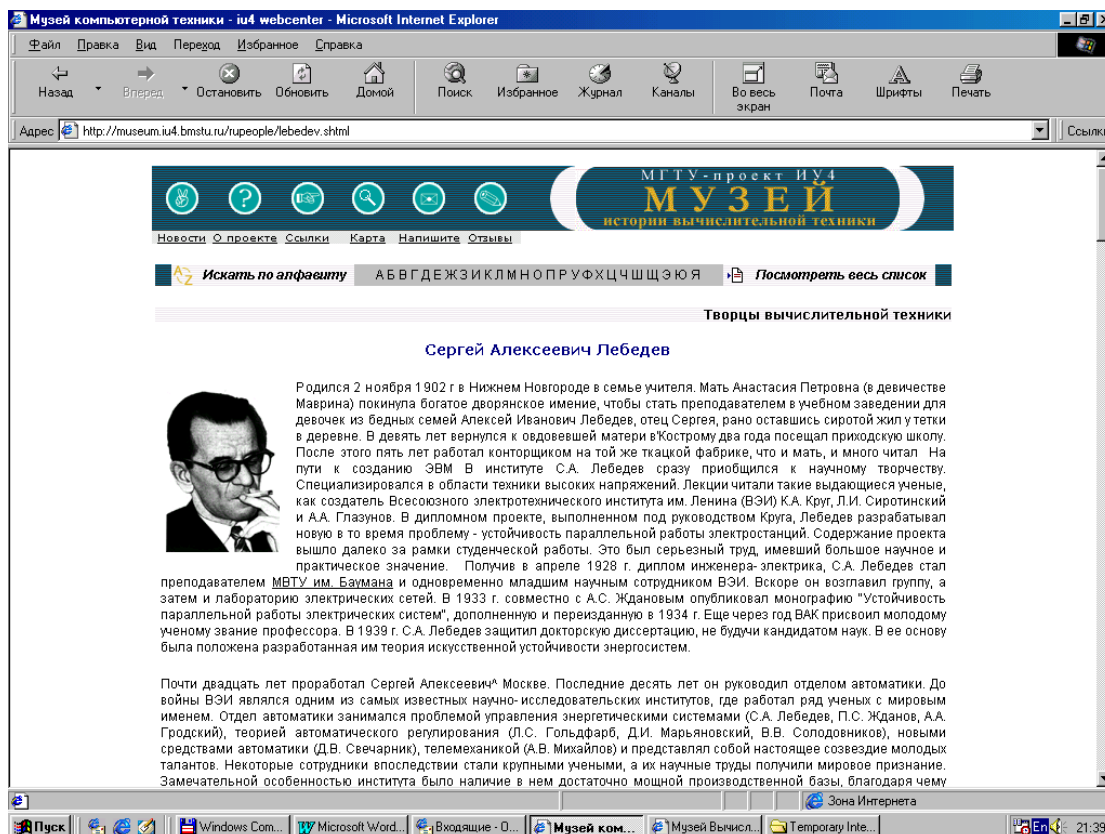


Рис. 5. Страница музея, посвященная С.А.Лебедеву

Технические средства реализации проекта

Для создания информационного сетевого ресурса, отвечающего требованиям стильного оформления и высоким качеством визуального восприятия, были использованы современные средства разработки и пакеты программного обеспечения ведущих мировых разработчиков, применяемые в профессиональном дизайне, обработке машинной графики и

разработке web ресурсов. Так весь проект реализован с помощью основного инструмента разработки – HTML версии 4.0 с применением элементов Dynamic HTML. Активные элементы, используемые при реализации проекта были написаны с использованием Java Script, удобным инструментарием создания единых стилистических подходов во всех страницах были таблицы стилей CSS. При разработке графических изображений основным инструментарием был Adobe Photoschop 5.5, также использована CorelXara. Анимационная часть проекта написана при помощи Ulead Gif Animator. Для создания большинства графических эффектов были использованы фильтры компании Ulead.

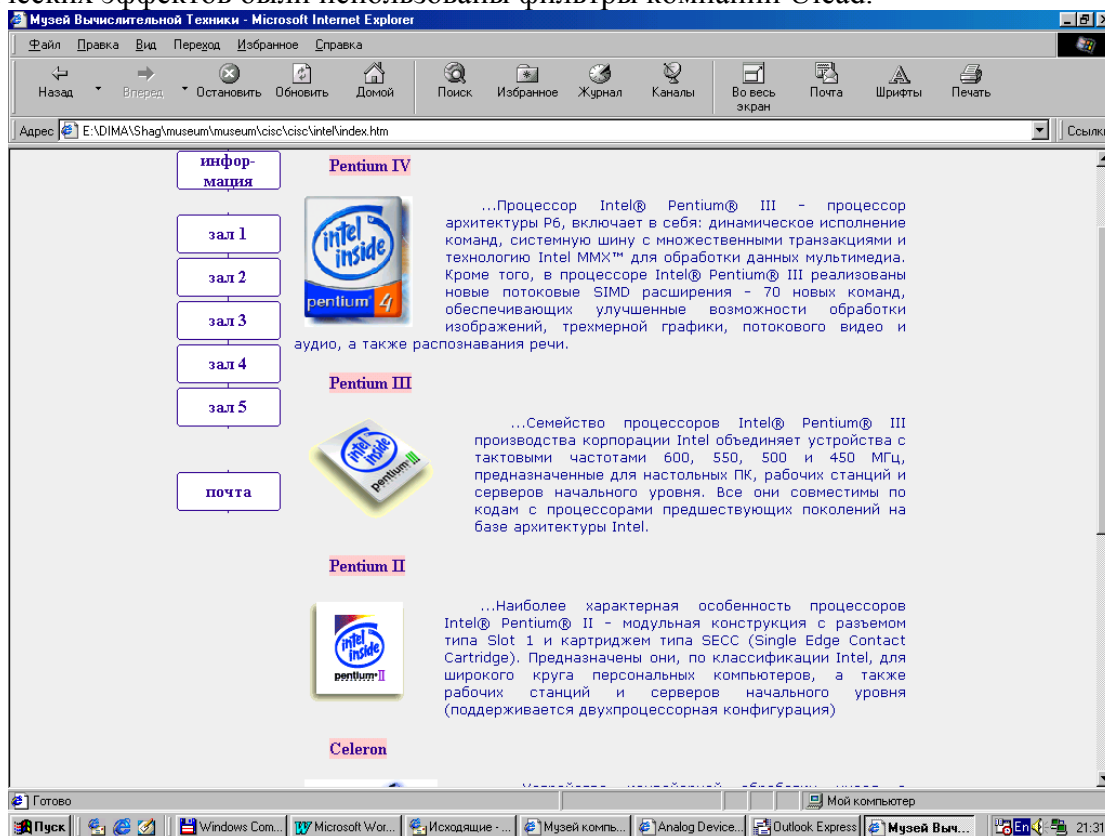


Рис. 6. Раздел по CISC и RISC архитектурам микропроцессоров

Заключение

Проект "История развития вычислительной техники" ещё достаточно молод. При создании этого web ресурса мною овладевало стремление и интерес заглянуть в прошлое, узнать как мыслили и рассуждали создатели всем известных машин МЭСМ, БЭСМ, Урал, ENIAC. Виртуальной экспозиции положено фундаментальное начало и с появлением новой техники, ростом информационных технологий его ждет, без сомнений, пополнение новых виртуальных экспозиций и интересное развитие.

ЛАБОРАТОРНАЯ УСТАНОВКА ПО КУРСУ ФИЗИКИ
РАЗДЕЛ: ТЕОРИЯ КОЛЕБАНИЙ
Купавцев Алексей Владимирович
Научный руководитель: к.т.н., доцент Власов А.И.
Средняя общеобразовательная школа №906
alexkup@newmail.ru .

**THE LABORATORY INSTALLATION AT THE RATE OF PHYSICS SECTION:
A VIBRATION THEORY**

Kupavcev A.V.
The scientific chief: Ph.D., senior lecturer Vlasov A.I.
Mean comprehensive school №906
alexkup@newmail.ru

Аннотация: В работе рассматривается разработка и эксплуатация лабораторной установки по курсу физика - раздел "Теория колебаний". В основу установки положены методы активного управления волновыми полями, в качестве средств цифровой обработки сигналов использована звуковая плата ПЭВМ, в качестве регистрирующей аппаратуры - микрофоны, излучателей колебаний - динамики.

The summary: In activity the mining and exploitation of the laboratory installation at the rate physics - section "Vibration theory" is esteemed. The methods of fissile control of wave fields are trusted to in the basis of the installation, as means of a digital signal processing the PC sound card utilised, as a recording equipment - microphones, radiators of oscillations - dynamics.

Введение

Активное управление волновыми полями – одно из бурно развивающихся сегодня направлений прикладной науки и техники. Методы изучения волновых процессов доведены физиками до совершенства, однако, задачи управления ходом этих процессов не имеют общего решения.

Среди наиболее наглядных примеров решения задач этого класса – созданные в последние годы средства защиты от вредного воздействия акустических шумов.

Большинство используемых сегодня методов защиты от шума реализованы пассивными средствами. При гашении низкочастотных колебаний такие средства малоэффективны, а их объемы и масса недопустимо возрастают. Приходится разрабатывать все новые и новые средства защиты людей, работающих с современной техникой. Все чаще требуются такие средства защиты, которые могут адаптироваться к изменяющимся условиям среды. Именно здесь и находят применение активные средства защиты, в которых используются дополнительные источники энергии.

Такие системы (САГ) формируют с помощью компенсирующих излучателей дополнительное гасящее поле с требуемыми пространственными и частотными характеристиками.

Но развитие САГ дальше невозможно без новых специалистов, а для этого надо их подготовить. Как показывает опыт, наибольший эффект в обучении дают программные системы, сочетающие в себе элементы тестирования и активного обучения. Методология применения таких систем заключается в разбиении учебного процесса на три этапа: экспериментальное исследование объекта в реальном масштабе времени, математическое моделирование и защита отчета с использованием контрольно-обучающей подсистемы. В ней пользователь получает возможность пройти обучение в сочетании с контролем полученных знаний по различным разделам теории проектирования САГ и смежных с ней дисциплин. Системы активного гашения как нельзя лучше подходят для того, чтобы на

конкретных примерах показать достаточно серьезные области приложения микропроцессорных систем и дать школьнику поэкспериментировать с ними.

1. Разработка проекта “Лабораторная установка по курсу физики, раздел: теория колебаний”.

Техническая реализация проекта.

Целью данной работы было создание программы для моделирования принципов активного управления акустическими полями, применительно для лабораторных работ по физики.

Предполагаемая лабораторная установка состоит из мультимедиа-компьютера (ПК + микрофон + динамики + звуковая плата) и программного обеспечения для генерации звуковых колебаний.(рис. 1)

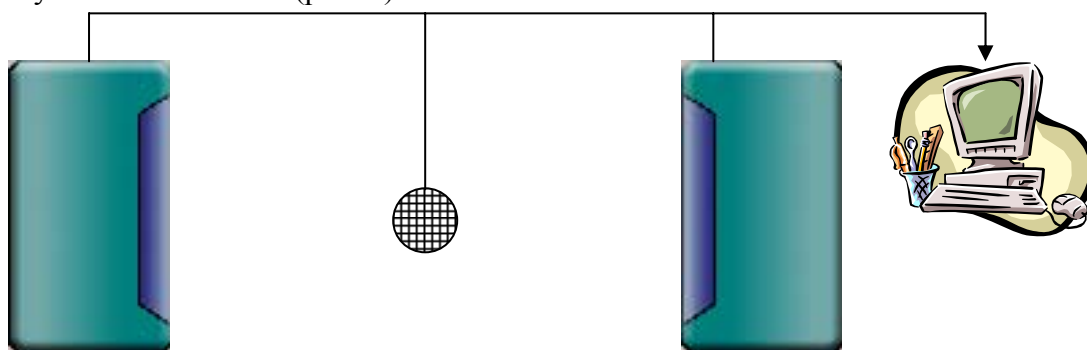


Рис.1. Схема лабораторной установки

Такой состав лабораторной установки позволяет использовать ее при достаточно минимальных денежных затратах, например в школах. Ведь давно не секрет что школы испытывают нехватку средств на закупку лабораторных принадлежностей – а вот персональный компьютер на данный момент есть почти в каждой школе, остается только сделать его мультимедийным и все.

Программная реализация проекта.

Программа выполнена на Borland Delphi 5 с использованием стандартных библиотек Microsoft Windows 9.x. В связи с тем что библиотеки Microsoft не могут преобразовывать звуковую волну в обратную ей и не могут воспроизводить две разных звуковых волны в различных колонках, одновременно, пришлось воспользоваться более низкоуровневыми библиотеками Windows – на уровне DLL библиотек. В связи с этим приходилось считывать звуковой файл ‘в ручную’ то есть на уровне работы с файлом и далее анализировать данные в файле, для последующего проигрывания и построения спектра звуковой волны. Процедура считывания файла с звуковыми данными и построения графика волны представлена в листинге 1.

Листинг 1

```
procedure TForm1.N9Click(Sender: TObject);  
var  
  f:file;  
  i,l,d:longint;  
  wavinfo:TPCMWaveFormat;  
begin  
  {}  
  DATA_ID:=mmioStringToFOURCC('data',0);  
if opendialog1.Execute then filename:=opendialog1.FileName
```

```

else exit;
label1.Caption:='Читаю звуковой файл...';
  disenb;
wav.series[0].Clear;
wout.series[0].Clear;
assignfile(f,filename);
reset(f,1);
  {--> проверки разные и смещение к звук. данным:}
  BlockRead(f,d,4);
blockread(f,wavinfo,SizeOf(TPCMWaveFormat));
repeat
  BlockRead(f,d,4);
  Seek(f,FilePos(f)-3);
until d=DATA_ID;
  Seek(f,FilePos(f)+3);
  BlockRead(f,d,4);

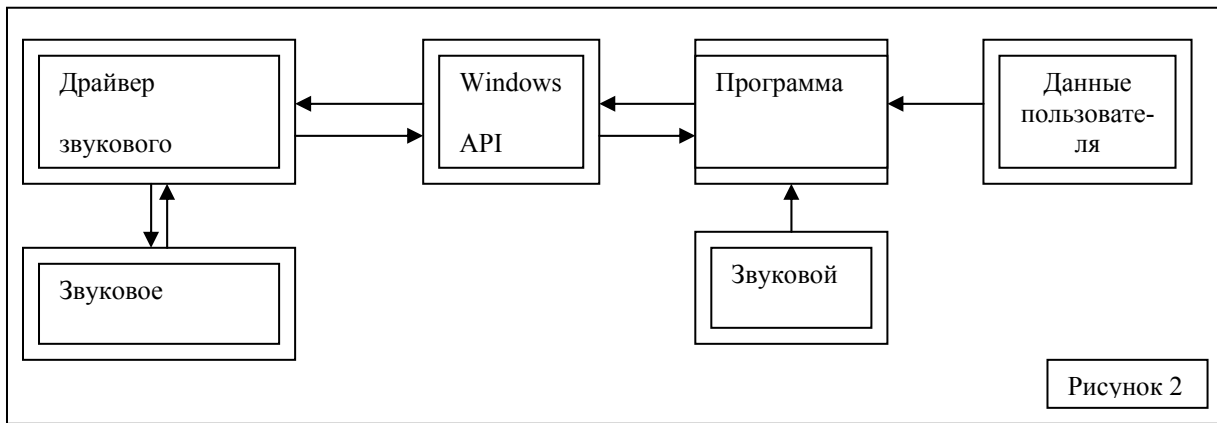
if d=0 then
begin
  ShowMessage('В файле отсутствуют звуковые данные');
  exit;
end;

  {--/ проверки разные и смещение к звук. данным:}
  i:=0;
label1.Caption:='Строю график...';
application.ProcessMessages;
while not eof(f) do begin
  if FileSize(f)-FilePos(f)<4 then Break;

  blockread(f,1,4);
  wav.Series[0].AddXY(i,1,"clgreen");
  wout.Series[0].AddXY(i,-1,"clblue");
  inc(i);
end;
closefile(f);
label1.Caption:='OK';
disenb;
end;

```

Более детальная схема работы программы представлена на рисунке 2.



1.3 Преобразование Фурье.

Для преобразования исходной волны из звукового файла в обратную ей волну используется быстрое преобразование Фурье. Быстрое преобразование Фурье (БПФ) – это название целого ряда эффективных алгоритмов, предназначенных для быстрого вычисления дискретно-временного ряда Фурье (ДВРФ). Основная идея БПФ – деление N – точечного ДВРФ на два и более меньших ДВРФ, каждый из которых можно вычислить отдельно, а затем линейно просуммировать с остальными, с тем чтобы получить ДВРФ исходной N – точечной последовательности. Эти ДВРФ меньшего размера можно в свою очередь поделить на еще меньшие ДВРФ соответственно меньших последовательностей. В общем случае вычисление N точечного ДВРФ требует выполнения $\log_2(N/2)$ шагов с операциями сложения и $N/2$ операциями умножения на каждом шаге. Таким образом, N точечное БПФ требует выполнения примерно $N \log_2(N)$ сложений и $N \log_2(N/2)$ умножений комплексных чисел, что значительно меньше тех N^2 операций, которые необходимы для раздельного вычисления N значений преобразования по N точечной последовательности данных. Если используется дополнение нулями, то за счет исключения, или удаления, вычислительных путей, содержащих одни лишь нулевые значения, можно достичь еще большего уменьшения объема вычислений. Программная реализация БПФ на языке Delphi приводится в листинге 2.

Листинг 2.

Процедура `ArrayCompExp` составляет таблицу комплексных экспонент. Входными данными является параметр `mode` (устанавливается в 0 для прямого ДВРФ и 1 для обратного ДВРФ); массив `W` – массив комплексных экспонент.

```

procedure ArrayCompExp(var W:array of complex; mode:Integer);
//вычисление комплексной синусоиды
var
  c1,c2:complex;
  o,n:integer;
  s:double;
begin
  n:=high(W);
  s:=2*pi/n;
  c1.re:=cos(s);
  c1.im:=mode*sin(s);
  c2:=compldg(1,0);
  for o:=0 to n do
    begin
      W[o]:=c2;
      c2:=MultComp(c2,c1);
    
```

end;

end;

Процедура DFT вычисляет дискретно-временной ряд Фурье по массиву отсчетов комплексных данных с помощью алгоритма БПФ с децимацией в частотной области. Здесь W – массив комплексных экспонент, полученный предыдущей процедурой, t – интервал отсчетов в секундах, параметр $mode$ – 0 для прямого ДВРФ, 1 – для обратного ДВРФ. X – комплексный массив исходных данных, замещающийся комплексными значениями преобразования.

procedure DFT(**const** W :array of complex; t :real; **var** X :array of complex;
const $mode$:integer);

var

$c1, c2, s$:complex;

$mm, ll, nn, jj, kk, nv2, nm1, i, j, k, v, n$:integer;

begin

$n := high(X) + 1$;

$v := round(\log_2(N))$;

$mm := 1$;

$ll := n$;

for $k := 1$ **to** v **do**

begin

$nn := round(ll/2)$;

$jj := mm + 1$;

$i := 1$;

while $i \leq N$ **do**

begin

$kk := i + nn$;

$c1 := addcomp(X[i-1], X[kk-1])$;

$X[kk-1] := rasncomp(X[i-1], X[kk-1])$;

$X[i-1] := c1$;

$inc(i, ll)$;

end;

if $nn = 1$ **then** continue;

for $j := 2$ **to** nn **do**

begin

$c2 := W[jj-1]$;

$i := j$;

while $i \leq n$ **do**

begin

$kk := i + nn$;

$c1 := addcomp(X[i-1], X[kk-1])$;

$X[kk-1] := multcomp(rasncomp(X[i-1], X[kk-1]), c2)$;

$X[i-1] := c1$;

$inc(i, ll)$;

end;

$jj := jj + mm$;

end;

$ll := nn$;

$mm := mm * 2$;

end;

$nv2 := round(n/2)$;

$nm1 := n - 1$;

$j := 1$;

for $i := 1$ **to** $nm1$ **do**

```

begin
if i<j then
begin
c1:=X[j-1];
X[j-1]:=X[i-1];
X[i-1]:=c1;
end;
k:=nv2;
while k<j do
begin
j:=j-k;
k:=round(k/2);
end;
j:=j+k;
end;
if mode=1 then
t:=1/(t*n);
s:=compldg(t,0);
for i:=1 to n do
begin
X[i-1]:=multcomp(X[i-1],s);
end;
end;
end;

```

2. Использование и тестирование проекта.

Ход лабораторной работы.

Вот каким может быть ход лабораторной работы. Звуковые колонки располагаются на ровной поверхности на расстоянии друг от друга (например 1м). Между ними располагается микрофон, который регистрирует сигнал в точке контроля. С помощью данного программного обеспечения учащийся подает синусоидальный сигнал на одну из колонок. Микрофон регистрирует сигнал в точке контроля. На основе этих данных школьник получает возможность рассчитать передаточную функцию среды от указанной колонки до контрольного микрофона и понять, от чего она зависит (например, для этого можно нагревать или охлаждать воздух в зоне проведения эксперимента или вносить дополнительные предметы на пути распространения волн).

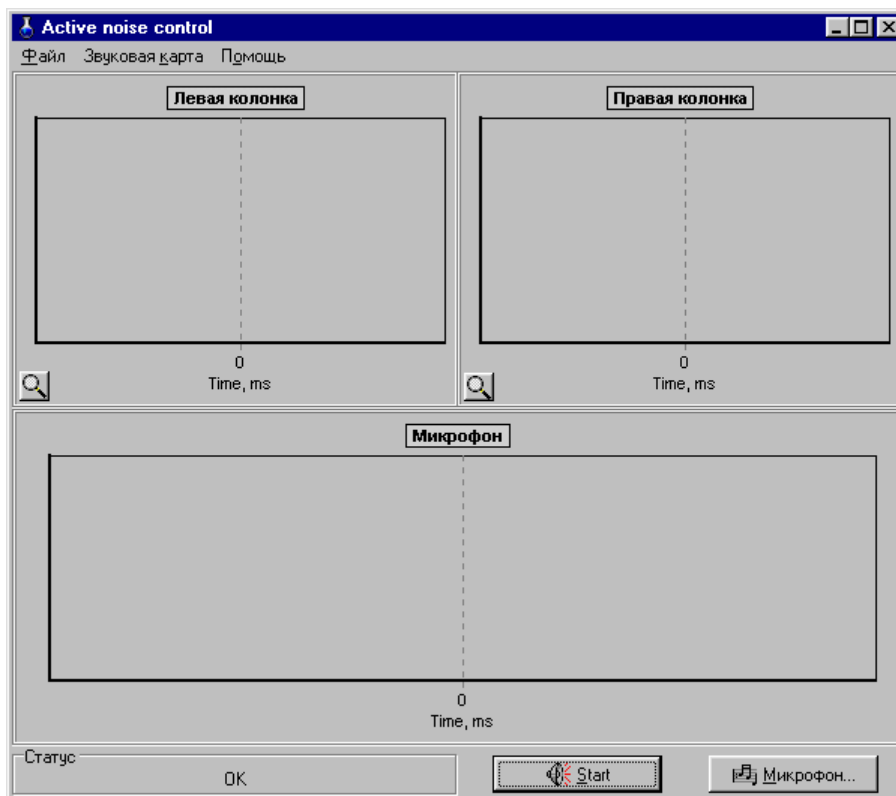
Исследовав среду и особенности распространения акустических волн в ней, учащийся переходит к моделированию принципов работы активных систем. Для этого на другую колонку подается синусоидальный сигнал в противофазе.

Вот задача для исследователя: найти те точки, в которых сигналы от двух колонок будут друг друга гасить, а также те, в которых эти сигналы будут друг друга усиливать. Действовать надо как расчетным, так и экспериментальным методом. После этого требуется подать на обе колонки одинаковый синусоидальный сигнал и посмотреть, как в этом случае изменится их взаимное влияние.

Возможных заданий может быть много, и все они позволяют на простом примере промоделировать работу целого класса серьезных прикладных систем.

Использование проекта

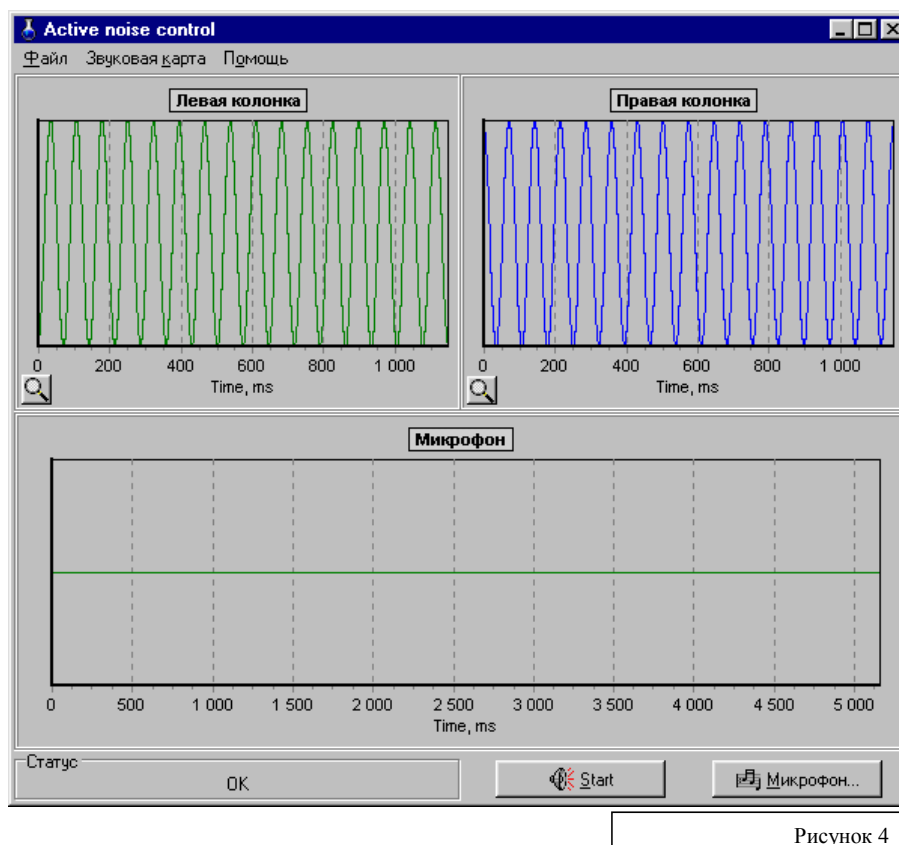
Программа снабжена достаточно дружелюбным и простым интерфейсом таким образом использование программы не только не требует специальных знаний, но и может быть доступно самому начинающему пользователю.



Риснок 3

На рисунке 3 показано окно программы – с этого окна пользователь начинает работу. Все кнопки и элементы меню снабжены подсказками и описаны в системе помощи.

Далее после открытия файла звуковых данных(WAV) программа автоматически строит график прямой и обратной звуковых волн записанных в файле. Далее после нажатие на кнопку Start происходит проигрывание и запись в микрофоне в зависимости от настроек указанных в опции Микрофон. Окно после нажатия на кнопку Start и открытия файла нарисовано на рисунке 4.



Рисунк 4

Выводы: Результаты опытной эксплуатации проекта в средней общеобразовательной школе №906.

1. возможность изучения теории поля экспериментальным путем
2. снижение, а точнее отсутствие утраты или повреждения лабораторного оборудования
3. динамичность и возможность непрерывного развития
4. удобство проведения лабораторной работы
5. дистанционное обучение по средствам сети INTERNET при помощи Active - X версии программы (в перспективе).

Список используемых источников

1. А.И. Власов, С.В. Колосков Профильные школы МГТУ им. Н.Э.Баумана и Интернет-технологии // Компьютер в школе. №8(№12). 1999. С.24-27.
2. С.В. Колосков Аппаратное и программное обеспечение для экспериментальных исследований систем активного гашения вибрационных и акустических полей // Сборник научных трудов студенческой научной конференции “Информатика и системы управления в XXI веке”. 1999. С.22-27
3. Власов А.И., Колосков С.В. Абитуриент on-line – новые технологические решения при профильной подготовке абитуриентов в МГТУ им. Н.Э.Баумана. //Международная конференция стран СНГ “Молодые ученые – науке, технологиям и профессиональному образованию для устойчивого развития: проблемы и новые решения” Сборник научных докладов и тезисов. Часть 2. - Москва. 1999. С.10-13.

НЕЙРОСЕТЕВОЙ КОМПЛЕКС НЕРАЗРУШАЮЩЕГО КОНТРОЛЯ ПЕЧАТНЫХ ПЛАТ.

Буянов А.А.

Научный руководитель: к.т.н., доц. Власов А.И.

Научный консультант: к.т.н., доцент Гриднев В.Н.

Кафедра "Проектирование и технология производства электронно-вычислительных и телекоммуникационных систем" МГТУ им.Н.Э.Баумана

<http://iu4.bmstu.ru>

NEURALCOMPLEX OF A NONDESTRUCTIVE EXAMINATION PRINTED CIRCUIT CARDS.

Bujanov A.A.

The scientific chief: Ph.D. Vlasov A.I.

The scientific adviser: Ph.D., senior lecturer Gridnev V.N..

Iu4, BMSTU

<http://iu4.bmstu.ru>

Аннотация: В работе рассмотрены методы и средства распознавания дефектов печатных плат на основе цифровой обработки изображений шлифов. Основное внимание уделено вопросам нейросетевых технологий распознавания.

Введение

Решение целого круга технических и технологических задач сопряжено с проведением обработки изображений, выделения отдельных элементов изображений и их идентификации и классификации. Примерами таких задач являются биометрические задачи распознавания образов (иридодиагностика, распознавание отпечатков пальцев и т.п.), задачи связанные с распознаванием текстов, выявлением различных неоднородностей и посторонних включений по фотоизображениям, выделение областей изображений по классификационным признакам (например, при обработке wavelet образов) и т.п.

Все вышеперечисленные и аналогичные задачи тесным образом связаны с понятием образ и распознавание образа. Образ, класс — классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку. Образное восприятие мира — одно из загадочных свойств живого мозга, позволяющее разобраться в бесконечном потоке воспринимаемой информации и сохранять ориентацию в океане разрозненных данных о внешнем мире. Воспринимаемая внешний мир, мы всегда производим классификацию воспринимаемых ощущений, т. е. разбиваем их на группы похожих, но не тождественных явлений.

Образы обладают характерным свойством, проявляющимся в том, что ознакомление с конечным числом явлений из одного и того же множества дает возможность узнавать сколь угодно большое число его представителей. В качестве образа можно рассматривать и некоторую совокупность состояний объекта управления, причем вся эта совокупность состояний характеризуется тем, что для достижения заданной цели требуется одинаковое воздействие на объект. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты. Именно эта объективность образов позволяет людям всего мира понимать друг друга.

Способность восприятия внешнего мира в форме образов позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов позволяет моделировать процесс их распознавания. Будучи отражением объективной реальности, понятие образа столь же объективно, как и сама реальность, а поэтому это понятие может быть само по себе объектом специального исследования.

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания. Обучение осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного образа и различными — на все объекты различных образов. Очень важно, что процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо других подсказок. В качестве объектов обучения могут быть либо картинки, либо другие визуальные изображения (буквы), либо различные явления внешнего мира, например звуки, состояния организма при медицинском диагнозе, состояние технического объекта в системах управления и др. Важно, что в процессе обучения указываются только сами объекты и их принадлежность образу. За обучением следует процесс распознавания новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему обучения распознаванию образов. В том случае, когда человек сам разгадывает или придумывает, а затем навязывает машине правило классификации, проблема распознавания решается частично, так как основную и главную часть проблемы (обучение) человек берет на себя.

Проведение дефектоскопии при производстве печатных плат также во многом сопряжено с задачами обработки изображений, выделения отдельных областей, их классификации и принятия решений по результатам такой классификации.

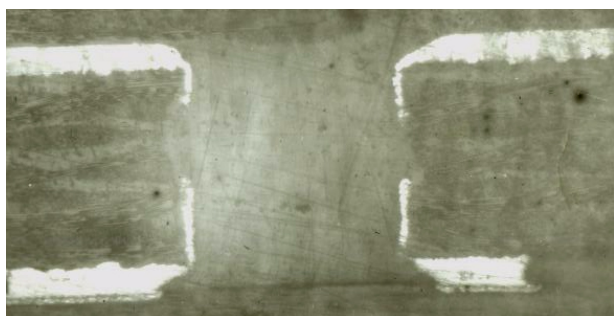
1. Дефектоскопия при производстве печатных плат - как объект применения нейросетевых методов

Традиционно контроль качества металлизированных отверстий печатных плат (ПП) осуществляется непосредственно на микрошлифах путем визуальной оценки геометрии и качества слоя металлизации. Структуру гальванической меди сравнивают с эталоном. Реально на производстве эту работу выполняют рабочий, в лучшем случае, высокой квалификации, но никак не технолог. Поскольку процесс выполняется визуальным методом, то время данной технологической операции велико, вероятность ошибок высока, кроме того, рабочий не может делать никакого анализа, кроме как «брак» или «не брак».

Автоматизация данного контроля дает ряд преимуществ:

1. Уменьшение время на данную операцию
2. Уменьшение вероятности ошибки, кроме того, в ответственном производстве технолог будет иметь возможность (по необходимости) контролировать правильность работы со своего рабочего места (при сетевой реализации программного комплекса) в режиме реального времени.
3. Если программный комплекс будет включать экспертную систему (она должна будет «знать» о структуре техпроцесса, его параметрах и режимах), то программный комплекс сможет оценить возможную причину брака и выработать рекомендации по методам его устранения (снижения). Экспертная система будет иметь возможность оценить процент брака при выборочных методах контроля.
4. Технологический отдел будет иметь возможность собирать различную статистическую информацию и, учитывая рекомендации программного комплекса, принимать решения о целесообразности использования тех или иных ТП.
5. При автоматизированном контроле качества металлизированных отверстий ПП необходимо распознавать образы микрошлифов на наличие различных видов дефектов.

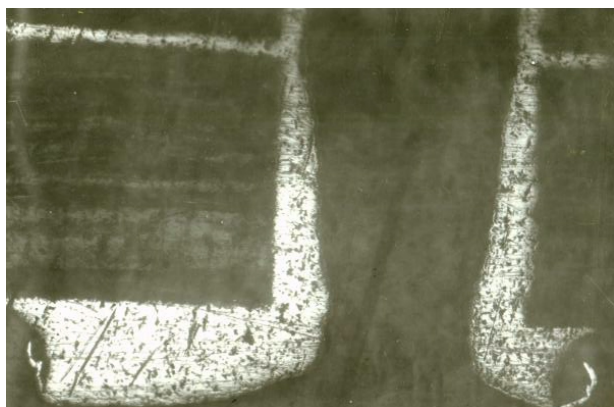
Примеры характерных дефектов металлизированных отверстий ПП:



а) Разрыв металлизации



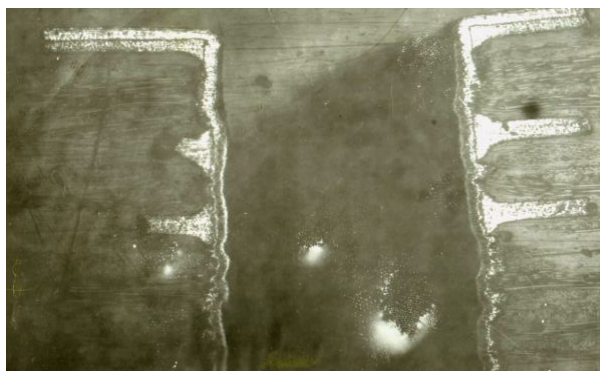
б) «Слезы»



в) Бочкообразная форма



г) Наросты меди



д) Увеличение площади торцов контактных площадок – «гвоздь»
Рис.1. Типовые дефекты металлизированных отверстий печатных плат.

Из примеров наглядно видно, что проблема анализа качества микрошлифов является непростой задачей даже для человека, т.к. человек, выполняющий данную работу, не является опытным технологом, а изготовить образцы микрошлифов на все разнообразие плат является не эффективным способом решения проблемы. Следовательно, система распознавания должна уметь не только грамотно классифицировать образы, но должна также быть обучена и проверена группой технологов. Для решения задачи автоматизации процесса дефектоскопии печатных плат по микрошлифам в МГТУ им.Н.Э.Баумана на кафедре ИУ4 "Проектирование и технология производства электронно-вычислительных и телекоммуникационных систем" ведется разработка аппаратно-программного комплекса дефектоскопии печатных плат.

Применение в создаваемом программном комплексе какого то определенного вида НС и алгоритма обучения не представляется возможным, п.э. необходимо использовать несколько различных видов НС с разными выходными параметрами, а окончательное решение должно формироваться методом «голосования» с различными весовыми коэффициентами. В общем случае искусственная нейросеть не является универсальной по своей структуре. Для ряда прикладных задач существует множество различных структур реализации, алгоритмов, методов обучения и т.п. Манипулируя этими различными видами можно получить, в конечном итоге, НС с оптимальными свойствами, параметрами, рассмотрим некоторые, из наиболее используемых методов [1-5].

1. Виды нейросетей, применяемых для распознавания образов.

Наибольшее использование для решения задач распознавания образов находят следующие нейросетевые методы: нейросетевой гауссов классификатор (Neural Gaussian Classifier), Сеть Хопфилда (HOPFIELD NET), Сеть Хемминга (HAMMING NET), Сеть Кохонена (KOHONEN'S NEURAL NETWORK), Сеть поиска максимума, Классификатор Карпендера/Гроссберга, Сеть встречного распространения (COUNTERPROPAGATION NETWORK), Сеть поиска максимума с прямыми связями (Feed-forward MAXNET), Входная звезда (INSTAR).

ТЕОРЕТИЧЕСКИ ЧИСЛО СЛОЕВ И ЧИСЛО НЕЙРОНОВ В КАЖДОМ СЛОЕ МОЖЕТ БЫТЬ ПРОИЗВОЛЬНЫМ, ОДНАКО ФАКТИЧЕСКИ ОНО ОГРАНИЧЕНО РЕСУРСАМИ КОМПЬЮТЕРА ИЛИ СПЕЦИАЛИЗИРОВАННОЙ МИКРОСХЕМЫ, НА КОТОРЫХ ОБЫЧНО РЕАЛИЗУЕТСЯ НС. ЧЕМ СЛОЖНЕЕ НС, ТЕМ МАСШТАБНЕЕ ЗАДАЧИ, ПОДВЛАСТНЫЕ ЕЙ.

ОБУЧЕНИЕ НС МОЖЕТ ВЕСТИ С УЧИТЕЛЕМ ИЛИ БЕЗ НЕГО. В ПЕРВОМ СЛУЧАЕ СЕТИ ПРЕДЪЯВЛЯЮТСЯ ЗНАЧЕНИЯ КАК ВХОДНЫХ, ТАК И ЖЕЛАТЕЛЬНЫХ ВЫХОДНЫХ СИГНАЛОВ, И ОНА ПО НЕКОТОРОМУ ВНУТРЕННЕМУ АЛГОРИТМУ ПОДСТРАИВАЕТ ВЕСА СВОИХ СИНАПТИЧЕСКИХ СВЯЗЕЙ. ВО ВТОРОМ СЛУЧАЕ ВЫХОДЫ НС ФОРМИРУЮТСЯ САМОСТОЯТЕЛЬНО, А ВЕСА ИЗМЕНЯЮТСЯ ПО АЛГОРИТМУ, УЧИТЫВАЮЩЕМУ ТОЛЬКО ВХОДНЫЕ И ПРОИЗВОДНЫЕ ОТ НИХ СИГНАЛЫ.

СУЩЕСТВУЕТ ВЕЛИКОЕ МНОЖЕСТВО РАЗЛИЧНЫХ АЛГОРИТМОВ ОБУЧЕНИЯ, КОТОРЫЕ, ОДНАКО, ДЕЛЯТСЯ НА ДВА БОЛЬШИХ КЛАССА:

ДЕТЕРМИНИСТСКИЕ И СТОХАСТИЧЕСКИЕ. В ПЕРВОМ ИЗ НИХ ПОДСТРОЙКА ВЕСОВ ПРЕДСТАВЛЯЕТ СОБОЙ ЖЕСТКУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ, ВО ВТОРОМ – ОНА ПРОИЗВОДИТСЯ НА ОСНОВЕ ДЕЙСТВИЙ, ПОДЧИНЯЮЩИХСЯ НЕКОТОРОМУ СЛУЧАЙНОМУ ПРОЦЕССУ.

ЕЩЕ ОДНА КЛАССИФИКАЦИЯ ДЕЛИТ НС НА СИНХРОННЫЕ И АСИНХРОННЫЕ. В ПЕРВОМ СЛУЧАЕ В КАЖДЫЙ МОМЕНТ ВРЕМЕНИ СВОЕ СОСТОЯНИЕ МЕНЯЕТ ЛИШЬ ОДИН НЕЙРОН. ВО ВТОРОМ – СОСТОЯНИЕ МЕНЯЕТСЯ СРАЗУ У ЦЕЛОЙ ГРУППЫ НЕЙРОНОВ, КАК ПРАВИЛО, У ВСЕГО СЛОЯ. АЛГОРИТМИЧЕСКИ ХОД ВРЕМЕНИ В НС ЗАДАЕТСЯ ИТЕРАЦИОННЫМ ВЫПОЛНЕНИЕМ ОДНОТИПНЫХ ДЕЙСТВИЙ НАД НЕЙРОНАМИ.

Нейросетевой гауссов классификатор (Neural Gaussian Classifier)

ПЕРСЕПТРОН МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ РЕАЛИЗАЦИИ ГАУССОВА КЛАССИФИКАТОРА ПО МАКСИМУМУ ВЕРОЯТНОСТИ (GAUSSIAN MAXIMUM LIKELIHOOD CLASSIFIER). В КЛАССИЧЕСКОМ АЛГОРИТМЕ ОБУЧЕНИЯ ПЕРСЕПТРОНА НЕ ИСПОЛЬЗУЮТСЯ ПРЕДПОЛОЖЕНИЯ ОТНОСИТЕЛЬНО РАСПРЕДЕЛЕНИЙ ПРИМЕРОВ ОБУЧАЮЩИХ ВЫБОРОК, А РАССМАТРИВАЕТСЯ ФУНКЦИЯ ОШИБКИ. ЭТОТ АЛГОРИТМ РАБОТАЕТ БОЛЕЕ УСТОЙЧИВО, ЕСЛИ ВХОДНЫЕ СИГНАЛЫ ФОРМИРУЮТСЯ В РЕЗУЛЬТАТЕ НЕЛИНЕЙНЫХ ПРОЦЕССОВ И РАСПРЕДЕЛЕННЫ НЕСИММЕТРИЧНО И НЕ ПО ГАУССОВУ ЗАКОНУ.

В ОСНОВЕ ПОСТРОЕНИЯ ГАУССОВА КЛАССИФИКАТОРА ЛЕЖАТ ПРЕДПОЛОЖЕНИЯ О РАСПРЕДЕЛЕНИЯХ ВХОДНЫХ СИГНАЛОВ. СЧИТАЕТСЯ, ЧТО ЭТИ РАСПРЕДЕЛЕНИЯ ИЗВЕСТНЫ И СООТВЕТСТВУЕТ ЗАКОНУ ГАУССА.

ФОРМУЛЫ ДЛЯ РАСЧЕТА ПАРАМЕТРОВ НЕЙРОСЕТЕВОГО ГАУССОВА КЛАССИФИКАТОРА ОПРЕДЕЛЯЮТСЯ СЛЕДУЮЩИМ ОБРАЗОМ. ПУСТЬ M_{A1} И S_{A1}^2 - СРЕДНЕЕ ЗНАЧЕНИЕ И ОТКЛОНЕНИЕ (МАТ. ОЖИДАНИЕ И ДИСПЕРСИЯ) ВХОДНОГО СИГНАЛА X_1 , КОГДА ВХОДНОЙ СИГНАЛ ПРИНАДЛЕЖИТ КЛАССУ А, M_{B1} И S_{B1}^2 - СРЕДНЕЕ ЗНАЧЕНИЕ И ОТКЛОНЕНИЕ ВХОДНОГО СИГНАЛА X_1 , КОГДА ВХОДНОЙ СИГНАЛ ПРИНАДЛЕЖИТ КЛАССУ В, И $S_{A1}^2 = S_{B1}^2 = S_1^2$. ТОГДА ЗНАЧЕНИЯ ВЕРОЯТНОСТИ, ИСПОЛЬЗУЕМЫЕ КЛАССИФИКАТОРОМ ПО МАКСИМУМУ ВЕРОЯТНОСТИ ПРОПОРЦИОНАЛЬНЫ СЛЕДУЮЩИМ ВЕЛИЧИНАМ:

$$L_A = - \sum_{i=1}^{n-1} \frac{(x_i - M_{A1})^2}{S_1^2} = - \sum \frac{x_i^2}{S_1^2} + 2 \sum \frac{M_{A1} x_i}{S_1^2} - \sum \frac{M_{A1}^2}{S_1^2}$$

$$L_B = - \sum_{i=1}^{n-1} \frac{(x_i - M_{B1})^2}{S_1^2} = - \sum \frac{x_i^2}{S_1^2} + 2 \sum \frac{M_{B1} x_i}{S_1^2} - \sum \frac{M_{B1}^2}{S_1^2}$$

КЛАССИФИКАТОР ПО МАКСИМУМУ ВЕРОЯТНОСТИ ДОЛЖЕН ВЫЧИСЛЯТЬ L_A И L_B И ВЫБИРАТЬ КЛАСС С НАИБОЛЬШЕЙ ВЕРОЯТНОСТЬЮ. ПЕРВЫЕ СЛАГАЕМЫЕ В ФОРМУЛАХ ИДЕНТИЧНЫ. ПОЭТОМУ ИХ МОЖНО ОПУСТИТЬ. ВТОРЫЕ СЛАГАЕМЫЕ МОГУТ БЫТЬ ВЫЧИСЛЕННЫ ПУТЕМ УМНОЖЕНИЯ ВХОДНЫХ СИГНАЛОВ НА СИНАПТИЧЕСКИЕ ВЕСА. ТРЕТЬИ СЛАГАЕМЫЕ - КОНСТАНТЫ, ЗНАЧЕНИЯ КОТОРЫХ МОЖНО ПРИСВОИТЬ СМЕЩЕНИЮ НЕЙРОНА.

Значения синаптических весов:

$$w_1 = \frac{2(M_{A1} - M_{B1})}{S_1^2}$$

$$b = \sum_{i=1}^{n-1} \frac{M_{A1}^2 - M_{B1}^2}{S_1^2}$$

Значение смещения:

Недостатки: примитивные разделяющие поверхности (гиперплоскости) дают возможность решать лишь самые простые задачи распознавания. Считаются априорно известными распределения входных сигналов, соответствующие разным классам.

Преимущества: программные или аппаратные реализации модели очень просты. Простой и быстрый алгоритм формирования синаптических весов и смещений.

Сеть Хопфилда (HOPFIELD NET).

В ОСНОВНОМ СЕТЬ ХОПФИЛДА ИСПОЛЬЗУЕТСЯ КАК АССОЦИАТИВНАЯ ПАМЯТЬ. ИСХОДНЫМИ ДАННЫМИ ДЛЯ РАСЧЕТА ЗНАЧЕНИЙ СИНАПТИЧЕСКИХ ВЕСОВ СЕТИ ЯВЛЯЮТСЯ ВЕКТОРЫ - ОБРАЗЦЫ КЛАССОВ. СЕТЬ ФУНКЦИОНИРУЕТ ЦИКЛИЧЕСКИ. ВЫХОД КАЖДОГО ИЗ НЕЙРОНОВ ПОДАЕТСЯ НА ВХОДЫ ВСЕХ ОСТАЛЬНЫХ НЕЙРОНОВ. НЕЙРОНЫ СЕТИ ИМЕЮТ ЖЕСТКИЕ ПОРОГОВЫЕ ФУНКЦИИ.

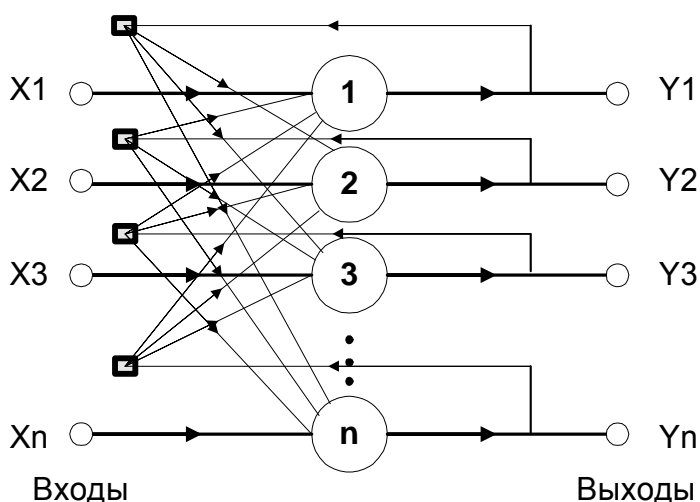


Рис.2. Структура сети Хопфилда.

*ИТЕРАЦИИ СЕТИ ЗАВЕРШАЮТСЯ ПОСЛЕ ТОГО, КАК ВЫХОДНЫЕ СИГНАЛЫ НЕЙРОНОВ ПЕРЕСТАЮТ МЕНЯТЬСЯ. РАЗМЕРНОСТИ ВХОДНЫХ И ВЫХОДНЫХ СИГНАЛОВ СОВПАДАЮТ. СЕТЬ, СОДЕРЖАЩАЯ N НЕЙРОНОВ МОЖЕТ ЗАПОМНИТЬ НЕ БОЛЕЕ $M=0.15*N$ ОБРАЗОВ. ПРИ ЭТОМ ЗАПОМИНАЕМЫЕ ОБРАЗЫ НЕ ДОЛЖНЫ БЫТЬ СИЛЬНО КОРРЕЛИРОВАННЫ.*

НЕДОСТАТКИ:

- Сеть обладает небольшой емкостью.
- Наряду с запомненными образами в сети хранятся и их негативы.
- Размерность и тип входных сигналов в точности совпадают с размерностью и типом выходных сигналов. Это существенно ограничивает применение сети в задаче распознавания образов.
- При использовании коррелированных векторов-образцов возможно заикливание сети в процессе функционирования.
- Квадратичный рост числа синапсов при увеличении размерности входного сигнала также можно считать недостатком сети.

Сеть Хемминга (HAMMING NET)

ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЕНИЯ РАССТОЯНИЯ ХЕММИНГА В ЗАДАЧЕ ПЕРЕДАЧИ ДВОИЧНЫХ СИГНАЛОВ ФИКСИРОВАННОЙ ДЛИНЫ ИССЛЕДОВАЛОСЬ В ТЕОРИИ ИНФОРМАЦИИ.

РАССТОЯНИЕ ХЕММИНГА МЕЖДУ ДВУМЯ БИНАРНЫМИ ВЕКТОРАМИ ОДИНАКОВОЙ ДЛИНЫ - ЭТО ЧИСЛО НЕСОВПАДАЮЩИХ БИТ В ЭТИХ ВЕКТОРАХ. НЕЙРОННАЯ СЕТЬ, КОТОРАЯ РЕАЛИЗУЕТ ПАРАЛЛЕЛЬНОЕ ВЫЧИСЛЕНИЕ РАССТОЯНИЙ ХЕММИНГА ОТ ВХОДНОГО ВЕКТОРА ДО НЕСКОЛЬКИХ ВЕКТОРОВ- ОБРАЗЦОВ, НОСИТ НАЗВАНИЕ СЕТИ ХЕММИНГА.

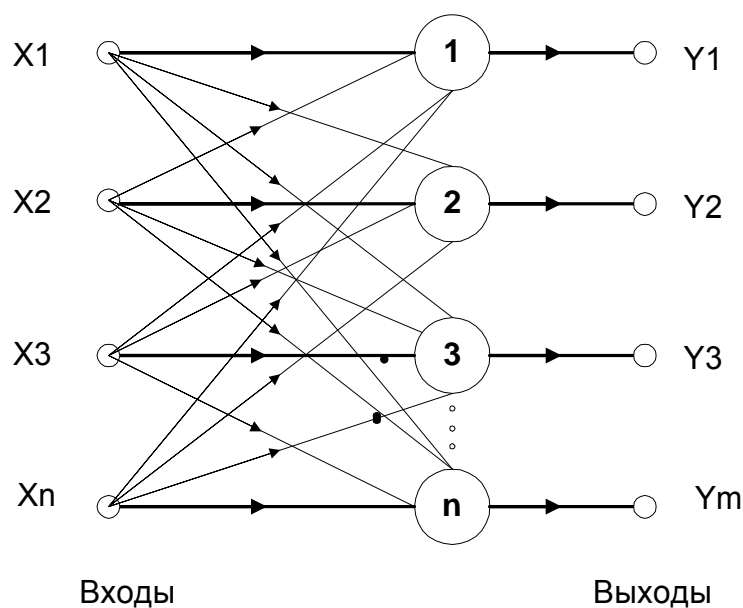


Рис.3. Структура сети Хемминга.

РАЗМЕРНОСТИ ВХОДНЫХ И ВЫХОДНЫХ СИГНАЛОВ МОГУТ НЕ СОВПАДАТЬ. ЧИСЛО СИНАПСОВ В СЕТИ: $(N \cdot M)$.

Недостатки :

- Сеть способна правильно распознавать (классифицировать) только слабо зашумленные входные сигналы.
- Возможность использования только бинарных входных сигналов существенно ограничивает область применения.

Преимущества:

- Сеть работает предельно просто и быстро.
- Выходной сигнал (решение задачи) формируется в результате прохода через всего лишь один слой нейронов. Для сравнения: в многослойных сетях сигнал проходит через несколько слоев. В сетях циклического функционирования сигнал многократно проходит через нейроны сети, причем число итераций, необходимое для получения решения, бывает заранее не известно.
- В модели использован один из самых простых алгоритмов формирования синаптических весов и смещений сети.
- В отличие от сети Хопфилда, емкость сети Хемминга не зависит от размерности входного сигнала, она в точности равна количеству нейронов (M). Сеть Хопфилда с входным сигналом размерностью 100 может запомнить 10 образцов, при этом у нее будет 10000 синапсов. У сети Хемминга с такой же емкостью будет всего лишь 1000 синапсов.

Сеть Кохонена (KOHONEN'S NEURAL NETWORK)

В МОЗГЕ НЕЙРОНЫ РАСПОЛАГАЮТСЯ В ОПРЕДЕЛЕННОМ ПОРЯДКЕ ТАК, ЧТО НЕКОТОРЫЕ ВНЕШНИЕ ФИЗИЧЕСКИЕ ВОЗДЕЙСТВИЯ ВЫЗЫВАЮТ ОТВЕТНУЮ РЕАКЦИЮ НЕЙРОНОВ ИЗ ОПРЕДЕЛЕННОЙ ОБЛАСТИ МОЗГА. НАПРИМЕР, В ТОЙ ЧАСТИ МОЗГА, КОТОРАЯ ОТВЕЧАЕТ ЗА ВОСПРИЯТИЕ ЗВУКОВЫХ СИГНАЛОВ, НЕЙРОНЫ ГРУППИРУЮТСЯ В СООТВЕТСТВИИ С ЧАСТОТАМИ ВХОДНОГО СИГНАЛА, НА КОТОРЫХ ОНИ РЕЗОНИРУЮТ. ХОТЯ СТРОЕНИЕ МОЗГА В ЗНАЧИТЕЛЬНОЙ СТЕПЕНИ ПРЕДОПРЕДЕЛЯЕТСЯ ГЕНЕТИЧЕСКИ, ОТДЕЛЬНЫЕ СТРУКТУРЫ МОЗГА ФОРМИРУЮТСЯ В ПРОЦЕССЕ САМООРГАНИЗАЦИИ. АЛГОРИТМ КОХОНЕНА В НЕКОТОРОЙ СТЕПЕНИ НАПОМИНАЕТ ПРОЦЕССЫ, ПРОИСХОДЯЩИЕ В МОЗГЕ. АЛГОРИТМ КОХОНЕНА ДАЕТ ВОЗМОЖНОСТЬ СТРОИТЬ НЕЙРОННУЮ СЕТЬ ДЛЯ РАЗДЕЛЕНИЯ ВЕКТОРОВ ВХОДНЫХ СИГНАЛОВ НА ПОДГРУППЫ. СЕТЬ СОСТОИТ ИЗ

М НЕЙРОНОВ, ОБРАЗУЮЩИХ ПРЯМОУГОЛЬНУЮ РЕШЕТКУ НА ПЛОСКОСТИ. ЭЛЕМЕНТЫ ВХОДНЫХ СИГНАЛОВ ПОДАЮТСЯ НА ВХОДЫ ВСЕХ НЕЙРОНОВ СЕТИ. В ПРОЦЕССЕ РАБОТЫ АЛГОРИТМА НАСТРАИВАЮТСЯ СИНАПТИЧЕСКИЕ ВЕСА НЕЙРОНОВ.

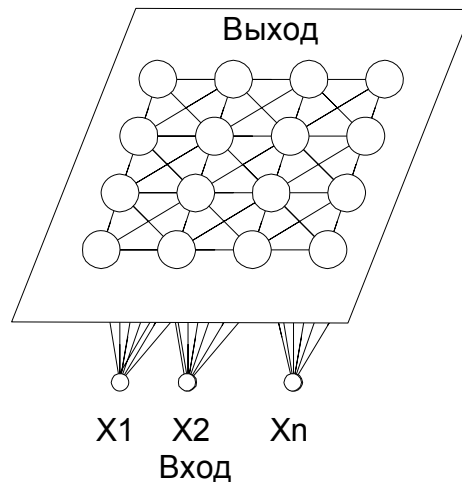


РИС.4. СТРУКТУРА СЕТИ КОХОНЕНА.

ВХОДНЫЕ СИГНАЛЫ - ВЕКТОРА ДЕЙСТВИТЕЛЬНЫХ ЧИСЕЛ - ПОСЛЕДОВАТЕЛЬНО ПРЕДЪЯВЛЯЮТСЯ СЕТИ. ЖЕЛАЕМЫЕ ВЫХОДНЫЕ СИГНАЛЫ НЕ ОПРЕДЕЛЯЮТСЯ. ПОСЛЕ ТОГО, КАК БЫЛО ПРЕДЪЯВЛЕНО ДОСТАТОЧНОЕ ЧИСЛО ВХОДНЫХ ВЕКТОРОВ, СИНАПТИЧЕСКИЕ ВЕСА СЕТИ ОПРЕДЕЛЯЮТ КЛАСТЕРЫ. КРОМЕ ТОГО, ВЕСА ОРГАНИЗУЮТСЯ ТАК, ЧТО ТОПОЛОГИЧЕСКИ БЛИЗКИЕ УЗЛЫ ЧУВСТВИТЕЛЬНЫ К ПОХОЖИМ ВНЕШНИМ ВОЗДЕЙСТВИЯМ (ВХОДНЫМ СИГНАЛАМ).

ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМА НЕОБХОДИМО ОПРЕДЕЛИТЬ МЕРУ СОСЕДСТВА НЕЙРОНОВ (МЕРУ БЛИЗОСТИ)

Недостатки:

- Сеть может быть использована для кластерного анализа только в том случае, если заранее известно число кластеров.

Преимущества:

- В отличие от сети ART Гроссберга, сеть Кохонена способна функционировать в условиях помех, так как число классов фиксировано, веса модифицируются медленно, настройка весов заканчивается после обучения (в сети ART настройка продолжается непрерывно).

Сеть поиска максимума (MAXNET)

СЕТЬ ЦИКЛИЧЕСКОГО ФУНКЦИОНИРОВАНИЯ. НА КАЖДОЙ ИТЕРАЦИИ БОЛЬШИЕ СИГНАЛЫ НА ВЫХОДАХ НЕЙРОНОВ ПОДАВЛЯЮТ СЛАБЫЕ СИГНАЛЫ НА ВЫХОДАХ ДРУГИХ НЕЙРОНОВ. ЕСЛИ В НАЧАЛЕ ФУНКЦИОНИРОВАНИЯ СЕТИ СИГНАЛ НА ВЫХОДЕ ОДНОГО ИЗ НЕЙРОНОВ ИМЕЛ МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ, ТО В КОНЦЕ ФУНКЦИОНИРОВАНИЯ ВСЕ ВЫХОДЫ НЕЙРОНОВ, КРОМЕ МАКСИМАЛЬНОГО, ИМЕЮТ ЗНАЧЕНИЯ, БЛИЗКИЕ К НУЛЮ. ТАКИМ ОБРАЗОМ, НЕЙРОН С НАИБОЛЬШИМ ВЫХОДНЫМ СИГНАЛОМ ПОБЕЖДАЕТ. ЭТО ДОСТИГАЕТСЯ ЗА СЧЕТ ИСПОЛЬЗОВАНИЯ ЛАТЕРАЛЬНЫХ СВЯЗЕЙ.

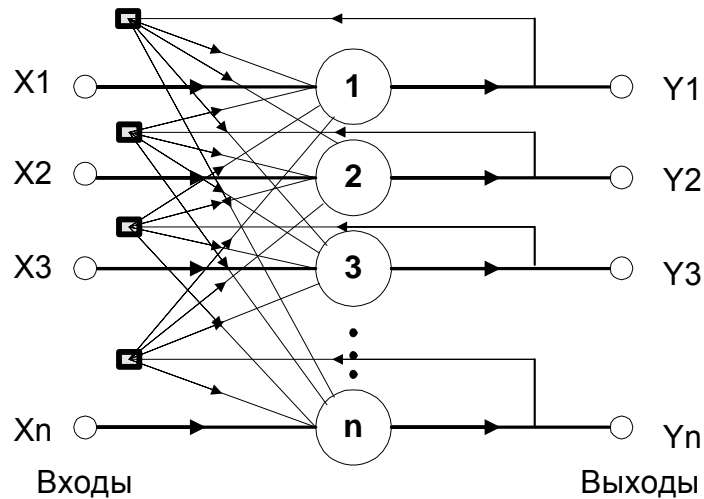


РИС.5. СТРУКТУРА СЕТИ ПОИСКА МАКСИМУМА.

ИТЕРАЦИИ СЕТИ ЗАВЕРШАЮТСЯ ПОСЛЕ ТОГО, КАК ВЫХОДНЫЕ СИГНАЛЫ НЕЙРОНОВ ПЕРЕСТАЮТ МЕНЯТЬСЯ.

Недостатки:

- Число итераций функционирования сети заранее не определено.
- Сеть определяет, какой из входных сигналов имеет максимальное значение, но в процессе функционирования теряет само значение максимального сигнала.
- Квадратичный рост числа синапсов при увеличении размерности входного сигнала также можно считать недостатком сети.

Преимущества: Сеть работает предельно просто.

Классификатор Карпендера/Гроссберга (сеть адаптивной резонансной теории - 1 (ART-1)).

СЕТЬ ART-1 ОБУЧАЕТСЯ БЕЗ УЧИТЕЛЯ, ОНА РЕАЛИЗУЕТ АЛГОРИТМ КЛАСТЕРИЗАЦИИ, ОЧЕНЬ ПОХОЖИЙ НА АЛГОРИТМ "ПОСЛЕДОВАТЕЛЬНОГО ЛИДЕРА" (SEQUENTIAL LEADER CLUSTERING ALGORITHM). В СООТВЕТСТВИИ С ЭТИМ АЛГОРИТМОМ ПЕРВЫЙ ВХОДНОЙ СИГНАЛ СЧИТАЕТСЯ ОБРАЗЦОМ ПЕРВОГО КЛАСТЕРА. СЛЕДУЮЩИЙ ВХОДНОЙ СИГНАЛ СРАВНИВАЕТСЯ С ОБРАЗЦОМ ПЕРВОГО КЛАСТЕРА. ГОВОРЯТ, ЧТО ВХОДНОЙ СИГНАЛ "СЛЕДУЕТ ЗА ЛИДЕРОМ" И ПРИНАДЛЕЖИТ ПЕРВОМУ КЛАСТЕРУ, ЕСЛИ РАССТОЯНИЕ ДО ОБРАЗЦА ПЕРВОГО КЛАСТЕРА МЕНЬШЕ ПОРОГА. В ПРОТИВНОМ СЛУЧАЕ ВТОРОЙ ВХОДНОЙ СИГНАЛ - ОБРАЗЕЦ ВТОРОГО КЛАСТЕРА. ЭТОТ ПРОЦЕСС ПОВТОРЯЕТСЯ ДЛЯ ВСЕХ СЛЕДУЮЩИХ ВХОДНЫХ СИГНАЛОВ. ТАКИМ ОБРАЗОМ, ЧИСЛО КЛАСТЕРОВ РАСТЕТ С ТЕЧЕНИЕМ ВРЕМЕНИ И ЗАВИСИТ КАК ОТ ЗНАЧЕНИЯ ПОРОГА, ТАК И ОТ МЕТРИКИ РАССТОЯНИЯ, ИСПОЛЬЗУЮЩЕЙСЯ ДЛЯ СРАВНЕНИЯ ВХОДНЫХ СИГНАЛОВ И ОБРАЗЦОВ КЛАССОВ.

ОСНОВНАЯ ЧАСТЬ СЕТИ ART-1 СХОЖА С СЕТЬЮ ХЕММИНГА, КОТОРАЯ ДОПОЛНЕНА ПОЛНОСВЯЗНОЙ СЕТЬЮ МАХNET. С ПОМОЩЬЮ ПОСЛЕДОВАТЕЛЬНЫХ СВЯЗЕЙ ВЫСЧИТЫВАЕТСЯ СООТВЕТСТВИЕ ВХОДНЫХ СИГНАЛОВ И ОБРАЗЦОВ КЛАСТЕРОВ. МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ СООТВЕТСТВИЯ УСИЛИВАЕТСЯ С ПОМОЩЬЮ ВЗАИМНОГО ЛАТЕРАЛЬНОГО ТОРМОЖЕНИЯ ВЫХОДНЫХ НЕЙРОНОВ. СЕТЬ ART-1 ОТЛИЧАЕТСЯ ОТ СЕТИ ХЕММИНГА ОБРАТНЫМИ СВЯЗЯМИ ОТ ВЫХОДНЫХ НЕЙРОНОВ К ВХОДНЫМ, КРОМЕ ТОГО ИМЕЕТСЯ ВОЗМОЖНОСТЬ ВЫКЛЮЧАТЬ ВЫХОДНОЙ НЕЙРОН С МАКСИМАЛЬНЫМ ЗНАЧЕНИЕМ СООТВЕТСТВИЯ И ПРОВОДИТЬ ПОРОГОВОЕ ТЕСТИРОВАНИЕ

СООТВЕТСТВИЯ ВХОДНОГО СИГНАЛА И ОБРАЗЦОВ КЛАСТЕРОВ, КАК ТОГО ТРЕБУЕТ АЛГОРИТМ "ПОСЛЕДОВАТЕЛЬНОГО ЛИДЕРА".

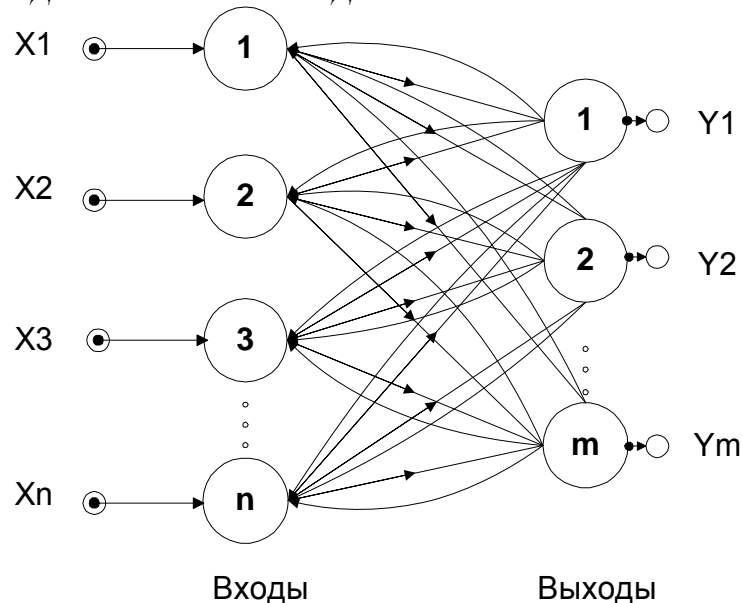


РИС.6. СТРУКТУРА КЛАССИФИКАТОРА КАРПЕНТОРА/ГРОССБЕРГА.

ЕМКОСТЬ СЕТИ СОВПАДАЕТ С ЧИСЛОМ НЕЙРОНОВ ВТОРОГО СЛОЯ И МОЖЕТ УВЕЛИЧИВАТЬСЯ В ПРОЦЕССЕ ФУНКЦИОНИРОВАНИЯ СЕТИ.

Недостатки:

- Неограниченное увеличение числа нейронов в процессе функционирования сети.
- В присутствии шума возникают значительные проблемы, связанные с неконтролируемым ростом числа образцов.

Преимущества: Обучение без учителя.

Сеть встречного распространения (COUNTER PROPAGATION NETWORK)

В СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ОБЪЕДИНЕНЫ ДВА АЛГОРИТМА: САМООРГАНИЗУЮЩАЯСЯ КАРТА КОХОНЕНА И ЗВЕЗДА ГРОССБЕРГА. ИХ ОБЪЕДИНЕНИЕ ПРИВОДИТ К ПОЯВЛЕНИЮ У МОДЕЛИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ СВОЙСТВ, КОТОРЫХ НЕТ У КАЖДОГО ИЗ ЭТИХ АЛГОРИТМОВ В ОТДЕЛЬНОСТИ.

НЕЙРОННЫЕ СЕТИ, КОТОРЫЕ ОБЪЕДИНЯЮТ РАЗЛИЧНЫЕ НЕЙРОПАРАДИГМЫ КАК СТРОИТЕЛЬНЫЕ БЛОКИ, БОЛЕЕ БЛИЗКИ К МОЗГУ ПО АРХИТЕКТУРЕ, ЧЕМ ОДНОРОДНЫЕ СТРУКТУРЫ. СЧИТАЕТСЯ, ЧТО В МОЗГЕ ИМЕННО КАСКАДНЫЕ СОЕДИНЕНИЯ МОДУЛЕЙ РАЗЛИЧНОЙ СПЕЦИАЛИЗАЦИИ ПОЗВОЛЯЮТ ВЫПОЛНЯТЬ ТРЕБУЕМЫЕ ВЫЧИСЛЕНИЯ.

В ПРОЦЕССЕ ОБУЧЕНИЯ СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ВХОДНЫЕ ВЕКТОРЫ АССОЦИИРУЮТСЯ С СООТВЕТСТВУЮЩИМИ ВЫХОДНЫМИ ВЕКТОРАМИ. ЭТИ ВЕКТОРЫ МОГУТ БЫТЬ ДВОИЧНЫМИ ИЛИ НЕПРЕРЫВНЫМИ. ПОСЛЕ ОБУЧЕНИЯ СЕТЬ ФОРМИРУЕТ ВЫХОДНЫЕ СИГНАЛЫ, СООТВЕТСТВУЮЩИЕ ВХОДНЫМ СИГНАЛАМ. ОБОБЩАЮЩАЯ СПОСОБНОСТЬ СЕТИ ДАЕТ ВОЗМОЖНОСТЬ ПОЛУЧАТЬ ПРАВИЛЬНЫЙ ВЫХОД, КОГДА ВХОДНОЙ ВЕКТОР НЕПОЛОН ИЛИ ИСКАЖЕН.

СЕТЬ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ИМЕЕТ ДВА СЛОЯ С ПОСЛЕДОВАТЕЛЬНЫМИ СВЯЗЯМИ. ПЕРВЫЙ СЛОЙ - СЛОЙ КОХОНЕНА, ВТОРОЙ - СЛОЙ ГРОССБЕРГА. КАЖДЫЙ ЭЛЕМЕНТ ВХОДНОГО СИГНАЛА ПОДАЕТСЯ НА ВСЕ НЕЙРОНЫ СЛОЯ КОХОНЕНА. ВЕСА СВЯЗЕЙ (W_{MN}) ОБРАЗУЮТ МАТРИЦУ W . КАЖДЫЙ НЕЙРОН СЛОЯ КОХОНЕНА СОЕДИНЕН СО ВСЕМИ НЕЙРОНАМИ СЛОЯ

ГРОССБЕРГА. ВЕСА СВЯЗЕЙ (VNP) ОБРАЗУЮТ МАТРИЦУ ВЕСОВ V. ОТЛИЧИЕ СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ОТ ДРУГИХ МНОГОСЛОЙНЫХ СЕТЕЙ С ПОСЛЕДОВАТЕЛЬНЫМИ СВЯЗЯМИ СОСТОИТ В ОПЕРАЦИЯХ, ВЫПОЛНЯЕМЫХ НЕЙРОНАМИ КОХОНЕНА И ГРОССБЕРГА.

В РЕЖИМЕ ФУНКЦИОНИРОВАНИЯ СЕТИ ПРЕДЪЯВЛЯЕТСЯ ВХОДНОЙ СИГНАЛ X И ФОРМИРУЕТСЯ ВЫХОДНОЙ СИГНАЛ Y. В РЕЖИМЕ ОБУЧЕНИЯ НА ВХОД СЕТИ ПОДАЕТСЯ ВХОДНОЙ СИГНАЛ И ВЕСА КОРРЕКТИРУЮТСЯ, ЧТОБЫ СЕТЬ ВЫДАВАЛА ТРЕБУЕМЫЙ ВЫХОДНОЙ СИГНАЛ.

Недостатки:

- Сеть не дает возможности строить точные аппроксимации (точные отображения). В этом сеть значительно уступает сетям с обратным распространением ошибки.
- К недостаткам модели также следует отнести слабую теоретическую проработку модификаций сети встречного распространения.

Преимущества:

- Сеть встречного распространения проста.
- Она дает возможность извлекать статистические свойства из множеств входных сигналов. Кохоненом показано, что для полностью обученной сети вероятность того, что случайно выбранный входной вектор (в соответствии с функцией плотности вероятности входного множества) будет ближайшим к любому заданному весовому вектору, равна $1/k$, k - число нейронов Кохонена.
- Сеть быстро обучается. Время обучения по сравнению с обратным распространением может быть в 100 раз меньше.
- По своим возможностям строить отображения сеть встречного распространения значительно превосходит однослойные перцептроны.
- Сеть полезна для приложений, в которых требуется быстрая начальная аппроксимация.
- Сеть дает возможность строить функцию и обратную к ней, что находит применение при решении практических задач

Сеть поиска максимума с прямыми связями -с управлением по возмущению (Feed-forward MAXNET)

МНОГОСЛОЙНАЯ СЕТЬ С ПРЯМЫМИ СВЯЗЯМИ. ВХОДНЫЕ СИГНАЛЫ ПОПАРНО СРАВНИВАЮТСЯ ДРУГ С ДРУГОМ. НАИБОЛЬШИЙ СИГНАЛ В КАЖДОЙ ПАРЕ ПЕРЕДАЕТСЯ НА СЛЕДУЮЩИЙ СЛОЙ ДЛЯ ДАЛЬНЕЙШИХ СРАВНЕНИЙ. НА ВЫХОДЕ СЕТИ ЛИШЬ ОДИН СИГНАЛ ИМЕЕТ НЕНУЛЕВОЕ ЗНАЧЕНИЕ. ОН СООТВЕТСТВУЕТ МАКСИМАЛЬНОМУ СИГНАЛУ НА ВХОДЕ СЕТИ.

В ОСНОВЕ ЛЕЖИТ НЕЙРОСЕТЕВОЙ КОМПАРАТОР, КОТОРЫЙ ВЫПОЛНЯЕТ СРАВНЕНИЕ ДВУХ АНАЛОГОВЫХ СИГНАЛОВ (X_1 И X_2). НА ВЫХОДЕ Z - ЗНАЧЕНИЕ МАКСИМАЛЬНОГО СИГНАЛА (X_1 ИЛИ X_2). ВЫХОДЫ Y_1 И Y_2 ПОКАЗЫВАЮТ, КАКОЙ ИМЕННО ВХОДНОЙ СИГНАЛ ИМЕЕТ МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ. НА РИСУНКЕ ПРОСТАВЛЕНЫ ЗНАЧЕНИЯ СИНАПТИЧЕСКИХ ВЕСОВ. СМЕЩЕНИЯ ВСЕХ НЕЙРОНОВ СЕТИ - НУЛЕВЫЕ. НЕЙРОНЫ, КОТОРЫЕ ПОМЕЧЕНЫ ЧЕРНЫМ ЦВЕТОМ, ИМЕЮТ ЖЕСТКИЕ ПОРОГОВЫЕ ФУНКЦИИ, ПЕРЕДАТОЧНЫЕ ФУНКЦИИ У ОСТАЛЬНЫХ НЕЙРОНОВ - ЛИНЕЙНЫЕ С НАСЫЩЕНИЕМ.

НА СЛЕДУЮЩЕМ РИСУНКЕ ПОКАЗАНА СЕТЬ ПОИСКА МАКСИМУМА С ПРЯМЫМИ СВЯЗЯМИ, КОТОРАЯ ДАЕТ ВОЗМОЖНОСТЬ ОПРЕДЕЛЯТЬ МАКСИМАЛЬНЫЙ СИГНАЛ ИЗ ВОСЬМИ ВХОДНЫХ СИГНАЛОВ. СЕТЬ СОСТОИТ ИЗ НЕСКОЛЬКИХ КОМПАРАТОРОВ И ДОПОЛНИТЕЛЬНЫХ НЕЙРОНОВ И СВЯЗЕЙ.

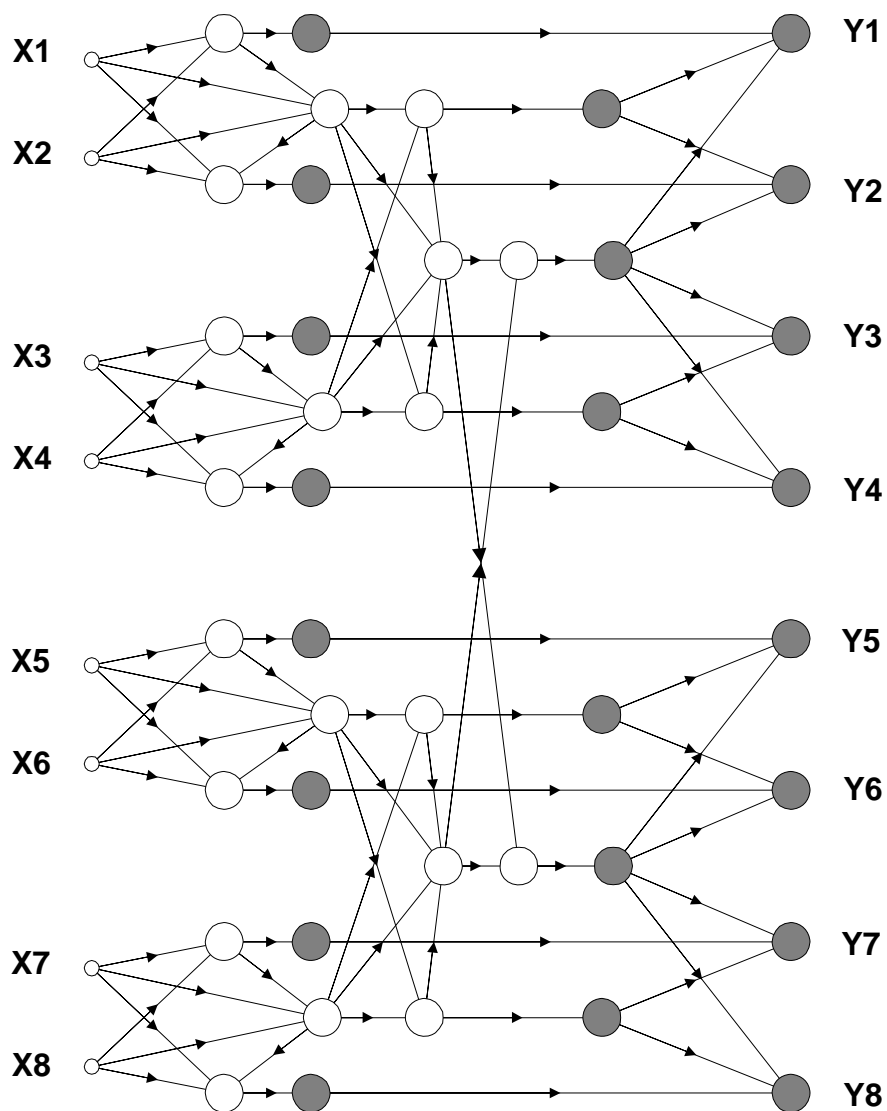


РИС.7. СЕТЬ ПОИСКА МАКСИМУМА С УПРАВЛЕНИЕМ ПО ВОЗМУЩЕНИЮ.

РАЗМЕРНОСТИ ВХОДНЫХ И ВЫХОДНЫХ СИГНАЛОВ СОВПАДАЮТ.

ЧИСЛО СЛОЕВ В СЕТИ: ПРИБЛИЗИТЕЛЬНО $\text{LOG}_2(M)$, ГДЕ M - РАЗМЕРНОСТЬ ВХОДНОГО СИГНАЛА.

СЕТЬ ПОИСКА МАКСИМУМА С УПРАВЛЕНИЕМ ПО ВОЗМУЩЕНИЮ ПРИМЕНЯЕТСЯ СОВМЕСТНО С СЕТЬЮ ХЕММИНГА, В СОСТАВЕ НЕЙРОСЕТЕВЫХ СИСТЕМ РАСПОЗНАВАНИЯ ОБРАЗОВ.

Недостатки: Число слоев сети растет с увеличением размерности входного сигнала.

Преимущества: В отличие от сети MAXNET циклического функционирования, в рассматриваемой модели заранее известен объем вычислений, который требуется для получения решения.

Входная звезда (INSTAR)

ВХОДНАЯ ЗВЕЗДА СОСТОИТ ИЗ НЕЙРОНА, НА КОТОРЫЙ ПОДАЕТСЯ ГРУППА ВХОДОВ ЧЕРЕЗ СИНАПТИЧЕСКИЕ ВЕСА. ВХОДНЫЕ И ВЫХОДНЫЕ ЗВЕЗДЫ МОГУТ БЫТЬ ВЗАИМНО СОЕДИНЕНЫ В СЕТИ ЛЮБОЙ СЛОЖНОСТИ. ГРОССБЕРГ РАССМАТРИВАЛ ВХОДНЫЕ ЗВЕЗДЫ КАК МОДЕЛИ ОТДЕЛЬНЫХ УЧАСТКОВ БИОЛОГИЧЕСКОГО МОЗГА.

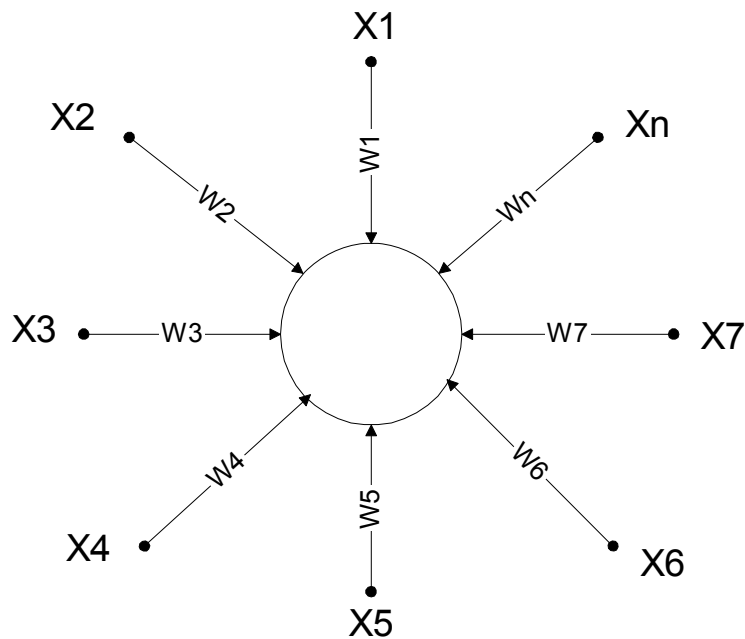


РИС.8. СТРУКТУРА ВХОДНОЙ ЗВЕЗДЫ (INSTAR).

ВХОДНАЯ ЗВЕЗДА ОБУЧАЕТСЯ РЕАГИРОВАТЬ НА ОПРЕДЕЛЕННЫЙ ВХОДНОЙ ВЕКТОР X И НИ НА КАКОЙ ДРУГОЙ. ЭТО ОБУЧЕНИЕ РЕАЛИЗУЕТСЯ ПУТЕМ НАСТРОЙКИ ВЕСОВ ТАКИМ ОБРАЗОМ, ЧТОБЫ ОНИ СООТВЕТСТВОВАЛИ ВХОДНОМУ ВЕКТОРУ. ВЫХОД ЗВЕЗДЫ ОПРЕДЕЛЯЕТСЯ КАК ВЗВЕШЕННАЯ СУММА ЕЕ ВХОДОВ. С ДРУГОЙ ТОЧКИ ЗРЕНИЯ, ВЫХОД МОЖНО РАССМАТРИВАТЬ КАК СВЕРТКУ ВХОДНОГО ВЕКТОРА С ВЕСОВЫМ ВЕКТОРОМ. СЛЕДОВАТЕЛЬНО, НЕЙРОН ДОЛЖЕН РЕАГИРОВАТЬ НАИБОЛЕЕ СИЛЬНО НА ВХОДНОЙ ОБРАЗ, КОТОРОМУ БЫЛ ОБУЧЕН.

Недостатки: Каждая звезда в отдельности реализует слишком простую функцию. Из таких звезд невозможно построить нейронную сеть, которая реализовала бы любое заданное отображение. Это ограничивает практическое применение входных звезд.

Преимущества: Входная звезда хорошо моделирует некоторые функции компонент живых (биологических) нейронных сетей. Сеть, включающая в себя входные звезды, может быть достаточно хорошей моделью отдельных участков мозга.

При решении практических задач входные звезды могут быть использованы для построения простых быстро обучаемых сетей.

2. Пример построения аппаратно-программного комплекса распознавания образов

Аппаратно-программный комплекс дефектоскопии печатных плат состоит из микроскопа, видеокамеры, средств ввода видеосигнала в ПК и специализированного прикладного программного обеспечения (ППО). Ядром ППО является подсистема распознавания образов, которая состоит из 6 модулей: модуль организации обмена с базами данных, модуль обработки обучающих пар, модуль обработки входной информации, нейроэмулятор, модуль алгоритмов обучения, модуль обработки выходных данных. На рис.9 представлена функциональная схема системы и связи ее модулей в двух режимах работы системы: режиме обучения и режиме реальной работы

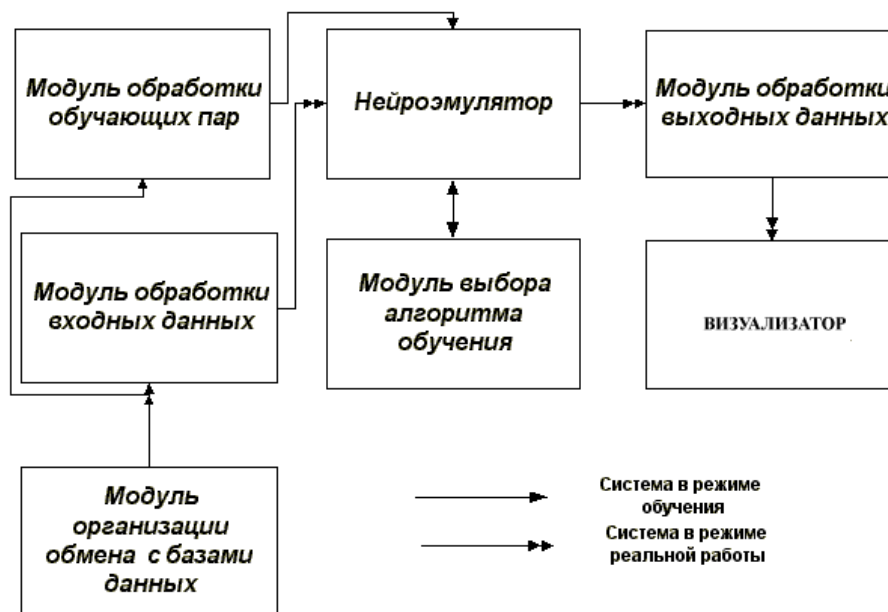


Рис.9. Структурная схема подсистемы распознавания образов.

10) Интерфейс системы представляет собой стандартный оконный интерфейс Windows-приложения. (рис

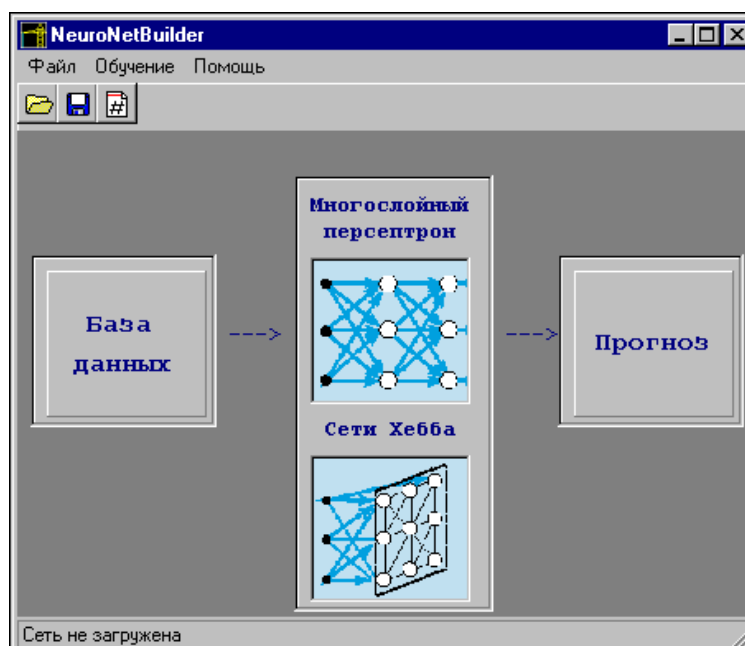


Рис. 10. Интерфейс программы.

Выводы

Совершенно ясно, что будущее методов неразрушающего контроля и дефектоскопии, связанных с обработкой видеообразов именно за использованием нейросетевых парадигм. Пока еще современные системы дефектоскопии и идентификации отнюдь не идеальны, осталось еще много нерешенных проблем. Например, система распознавания отпечатков пальцев плохо справляется с жирными пальцами или заляпанным стеклом у сканера и здесь на помощь разработчику могут прийти нейросетевых парадигмы. В дальнейшем рассмотренные методы могут найти применение (и уже находят) при решении задач дистанционного мониторинга различных объектов на таможне, распознавания рентгеновских снимков, обработки картографической информации и т.п.

Список использованной литературы

1. Ф.Уоссермен, Нейрокомпьютерная техника, М.,Мир, 1992.

2. Richard P. Lippmann, An Introduction to Computing with Neural Nets, IEEE Acoustics, Speech, and Signal Processing Magazine, April 1987.
3. Carpenter G.A, Grossberg S. Normal and amnesia learning, recognition, and memory by a neural model of cortico-hippocampal interactions. Trends in Neurosci., 16, pp.131-137, 1993.
4. А. Горбань, Д. Россиев Нейронные сети на персональном компьютере. //Новосибирск, Наука, 1996.
5. Охонин В.А. Вариационный принцип в теории адаптивных сетей. Препринт ИФ СО АН СССР, Красноярск, 1987, №61Б. 18 с. \
6. Власов А.И., Конькова А.Ф. Медико-диагностические экспертные системы для оценки адекватности адаптивной реакции организма на воздействие экстремальных факторов //Конверсия.-1995, №9-10.-С.18-21.
7. Макеев С.С. Биометрия, Биометрия, Биометрия! // Сборник научных трудов конференции "Научные технологии и интеллектуальные системы". - Москва, МГТУ им.Н.Э.Баумана, 16-17 апреля 2000 г.

ИССЛЕДОВАНИЕ МЕТОДОВ ОБЕСПЕЧЕНИЯ СТАБИЛЬНОГО ФУНКЦИОНИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ В ФОРСИРОВАННЫХ РЕЖИМАХ

Руденко Михаил

Научный руководитель: ст. преподаватель Соловьев В.А.

Лицей №1524, учащийся 11 "Г" класса

RESEARCH OF METHODS OF MAINTENANCE OF STABLE OPERATION OF MICROSYSTEMS IN AUGMENTED MODES

Rudenko Michael

The scientific chief: an item the teacher Solovjev V.A..

Liceum №1524, 11 "G" class

Аннотация: В работе исследованы вопросы обеспечения стабильного функционирования микропроцессорных систем в форсированных режимах. Основное внимание уделено вопросам стабильности работы и обеспечения допустимых тепловых режимов. По результатам исследований разработана методика обеспечения высоких функциональных показателей работы микропроцессорных систем в форсированных режимах и создана Информационно-Справочная Интернет Система MP-SYSTEMS, размещенная на выделенном домене сети Интернет <http://mpsystems.iu4.bmstu.ru>.

The summary: In activity the problems of maintenance of stable operation of microsystems in augmented modes are investigated. The main attention is given to problems of stability of activity and maintenance of permissible thermal modes. By results of researches the technique of maintenance of high functional parameters of activity of microsystems in augmented modes is designed and the System MP-SYSTEMS, located on the discharged domain of a network the Internet <http://mpsystems.iu4.bmstu.ru>. is built the Internet

Введение

В связи с все возрастающими темпами развития современной вычислительной техники перед рядовыми пользователями практически ежегодно встает проблема upgrade'a их компьютерной системы. С этим сопряжены немалые материальные затраты. Решение этой проблемы заключается в увеличении срока, в течение которого пользователь удовлетворен своей системой. Этого можно добиться путем применения форсированных режимов работы компьютерной техники и микропроцессоров в частности. Использование таких режимов позволяет вывести систему на современный технологический уровень, увеличивая тем самым ее предельную полезность для конечного пользователя.

Сначала я бы хотел привести мой личный опыт использования форсированных режимов микропроцессорной техники и основные принципы достижения максимального результата. Для того чтобы с результатами исследований могло ознакомиться как можно большее число людей, я решил использовать возможности глобальной сети Интернет, создав Информационно-Справочную Интернет Систему (сайт), посвященную теме доклада, и разместив ее в Сети по адресу: <http://www.mpsystems.iu4.bmstu.ru>. С ее структурой, содержанием и основными свойствами вы ознакомитесь во второй части работы.

1. Работа микропроцессорных систем в форсированных режимах

1.1. Техника разгона и личный опыт

1.1.1 Понятие разгона

Что такое разгон? Разгон (overclocking) – это функционирование некоторых или всех узлов компьютера в форсированном режиме. Самым распространенным видом overclocking'a является разгон процессора (или "камня"). Естественно падает надежность работы и немного сокращается срок службы, но с первой проблемой вполне можно справиться, а

насчет второй – я никогда не видел людей, которые использовали бы процессор 10 лет, а именно таков его приблизительный срок службы. Конечно, существует вероятность (около 0.01%) выхода процессора из строя, но это я знаю только понаслышке.

Из партии процессоров на конвейере выбираются несколько штук, которые, пройдя очень жесткое тестирование, маркируются на определенную номинальную частоту, по ним маркируются и остальные процессоры из всей партии. Так как другие процессоры не тестировались, среди них могло оказаться некоторое количество более качественных кристаллов, которые способны работать на повышенных частотах. При этом опытные образцы проходят тестирование на стабильность в экстремальных условиях, после чего с них снимают еще 10-15 % номинальной частоты. Все это увеличивает возможность появления процессоров с заниженной реальной тактовой частотой. Частота любого процессора есть произведение частоты материнской платы (Frontside Bus Clock) на коэффициент умножения (Multiplier Factor). Отсюда, как можно легко догадаться, имеется два варианта разгона:

1. Поднятие коэффициента умножения.
2. Повышение частоты материнской платы.

У современных процессоров компании Intel множитель заблокирован в связи с "борьбой" против перемаркировки, следовательно, изменять можно **только** FSB.

1.1.2 Краткая история разгона. Перемаркировка

Что такое перемаркировка? Некоторые темные личности, освоив разгон, поняли, что на этом можно зарабатывать деньги. Допустим, берется Pentium-120. Он без особых проблем может разогнаться до 133 МГц. С верхней поверхности процессора снимается тонкий слой и наносится новая маркировка, а в те времена, когда самым совершенным был Pentium-166 разница между Pentium-120 и Pentium-133 достигала 40-50 \$. Волна перемаркировки с США нахлынула на Европу, Intel решила положить конец этому беспределу.

Сначала нельзя было повысить коэффициент умножения выше определенного уровня – это давало полную защиту конечного пользователя от мошенников, так как он мог теперь поставить максимальный коэффициент умножения вместе со стандартной для данного типа процессора FSB и получить подлинный результат.

Но на этом ничего не закончилось – эпопея с блокированием коэффициента умножения продолжилась по одной простой причине: компания Intel не захотела терять прибыль. Действительно, почему бы ни заняться разгоном конечному пользователю? Допустим, он покупает процессор Celeron 300A, разгоняет его до 450 МГц и получает процессор за 300 \$, заплатив всего 80 \$ (цены приведены на начало 1999 г.), тогда зачем ему покупать дорогой процессор, если можно купить дешевый и разогнать его? Intel то же быстро это поняла, и новые процессоры стали выпускаться с 1-3, а потом и совсем с одним коэффициентом умножения. На данный момент все современные процессоры данной компании выпускаются с одним единственным коэффициентом умножения. Но пока 1997 год и Intel выпускает в массовую продажу свои первые PentiumII процессоры, рассчитанные на 66 МГц (ядро Klamath) и любые коэффициенты умножения. Эти камни были выпущены по 0.35 мкм технологии и разгонялись весьма посредственно, но у них был свой плюс: не заблокированный коэффициент умножения. Возьмем, например, PentiumII-266 (Klamath). В нормальных условиях этот процессор разгоняется до $4 \times 83 = 332$ МГц, но при этом возникают проблемы с другими устройствами компьютера (о которых я расскажу ниже). Следующая частота, на которой таких проблем не возникает, это 100 МГц, но при этом вряд ли заработает процессор: $4 \times 100 = 400$ МГц. Но есть одно но: так как коэффициент умножения не заблокирован, можно поставить $3.5 \times 100 = 350$ МГц – скорость процессора повышена, но не очень резко, при этом проблем с другими устройствами нет. Поэтому, если у пользователя были хорошая память, карты расширения и т.д., он мог поставить высокую внешнюю частоту (FSB) вместе с низким коэффициентом умножения, что часто давало отличный результат. Это естественно не понравилось Intel, и она заблокировала коэффициент умножения. Но дальше больше: выходят новые процессоры с технологией 0.25 мкм (Deschutes, а потом и Mendocino, построенный на ядре Deschutes), которые были рассчитаны

на гораздо более высокие частоты тактирования, и хотя коэффициент умножения у них был заблокирован, практически все из них «заходились» на 100, а иногда и на 112 МГц FSB, при паспортной частоте 66 МГц! В 1999 году, перед объявлением новой модели процессоров фирмы Intel - Pentium III - начали упорно ходить слухи, что «разгона не будет». Процессор якобы сделан так, что запустить его на FSB выше паспортной не удастся. Но, к счастью, этого не произошло. Также планировалось сделать специальный вывод у процессора, сигнал которого автоматически выставлял бы внешнюю частоту на материнской плате – 66 или 100 МГц. Эта задумка была воплощена в жизнь пин B21 (или AJ33 на сокетных моделях Celeron), но... не была поддержана большинством производителей материнских плат. Некоторые все же сконструировали свои платы с поддержкой этой опции, но стоило только наклеить скотчем пин B21 либо перерезать дорожку, соединяющую вывод AJ33 с контактом B21 – и проблема решена. Из несогласных с политикой Intel резко выделилась фирма ABIT, которая не только проигнорировала B21, но и дала возможность конечному пользователю самому выставлять напряжение на ядре процессора, а также опцию SEL100/66, позволяющую разблокировать коэффициент умножения у некоторых типов процессоров.

1.1.3 Выбор системной платы. Основные методы. Технические характеристики

Как видно, материнская плата очень важна для разгона, поэтому к выбору

материнской платы надо относиться очень

внимательно. На этом элементе компьютера

экономить не стоит, т.к. от него и зависят

возможности по разгону. На момент написания

этой работы существовала довольно сильная

проблема с выбором материнской платы, даже у

известных производителей. Основные характеристики были примерно одинаковыми, так



Рис. 1

как по своей сути их набор стандартен. Возможность разгона, как нестандартная опция,

требует таких же нестандартных решений и возможностей. Я расскажу об основных

характеристиках материнской платы, которые серьезно влияют на разгон.

Главный элемент материнской платы – чипсет (chipset). Эта микросхема определяет

стандартный набор характеристик, таких как типы

устанавливаемых процессоров, количество и типы

разъемов расширения, паспортные внешние

частоты и др. Все эти параметры влияют на очень



важные характеристики компьютера в целом: расширяемость и наращиваемость мощности. Поэтому первое, на что надо смотреть при покупке материнской платы, это установленный на ней чипсет.

Рис. 2

Помимо этого, чипсет иногда является причиной “плохого” разгона, так как болезненно реагирует на превышение паспортной внешней частоты. Платы известных производителей в основном строятся на зарекомендовавших себя чипсетах, отличающихся надежностью и производительностью. Между собой чипсеты разнятся по применимости в системах начального уровня (i810(e)), среднего (i440BX, Via Apollo Pro 133 (A), i815 {Solano}), профессионального уровня для high-end серверов (i820 (Camino), i840). Деление это довольно условное, так как, например, чипсеты среднего уровня спокойно могут работать в high-end серверах, но вот Camino не как не отнесешь к системе начального уровня: один 128 Мб модуль RIMM (тип памяти, работающий на i820) стоит сейчас порядка 500-700 \$. Наиболее перспективным чипсетом на сегодняшний день является i815, сочетающий в себе отличную производительность, расширяемость и надежность с относительно низкой ценой построенной на его основе материнской платы. Нельзя забывать и о «старичке» i440BX, а также о Via Apollo Pro 133 A, обладающего всеми новейшими характеристиками, но относительной невысокой производительностью (хотя есть информация, что на системах BeOS и Linux он показывает более высокую производительность, чем i440BX). Основные характеристики чипсетов приведены в таблице 1:

Основные характеристики	VIA Apollo Pro133A	Intel 440BX	Intel 820	Intel 810\810e
North Bridge	VT82C694X	82443BX	82820	82810
South Bridge	VT82C596B	82371EB	82801AA	82801
Шины				
PCI 2.1	+	+	+	+

Максимальное число PCI Master устройств	5	5	5	5 (4?)
AGP	1x/2x/4x	1x/2x	1x/2x/4x	Встроенный
ISA	+	+	-	-
Частота FSB	66/100/133 МГц	66/100 МГц	66/100/133 МГц	66/100\66/100/133 МГц
Асинхронная шина памяти	+	-	+	+
Память				
Поддерживаемые типы	FP, EDO, PC100 SDRAM, PC133 SDARM, VCM SDRAM, HSDRAM	EDO, PC100 SDRAM	PC100 SDRAM, Direct RDRAM	PC100 SDRAM\ PC100 SDRAM PC133 SDRAM
Максимальный объем	1,5 Гбайт	1 Гбайт	768 Мбайт*	1 Гбайт
Количество банков	8	8	6	8
ЕСС и контроль четности	+	+	+	+
SDRAM Timing	x-1-1-1-1-1-1-1	x-1-1-1-1-1-1-1-1-1	x-1-1-1-1-1-1-1	x-1-1-1-1-1-1-1
Интегрированные возможности				
UltraATA/66	+	-	+	+
Порты USB	4	2	2	2
Интегрированный мониторинг	+	-	-	-
AC 97 Link	+	-	+	+

Табл. 1

* реально на платах 512 Мбайт

Следующим по важности элементом в плане разгона является набор частот на материнской плате, который позволяет наиболее тонко подобрать оптимальное значение частоты тактирования процессора и FSB.

1.1.4 Алгоритм разгона

Здесь позволю себе привести опробованный мною на личном опыте алгоритм разгона, позволяющий безопасно разогнать процессор.

Каждый раз поднимаясь на одно значение, повышайте частоту материнской платы (также можно воспользоваться более удобным способом: посмотреть предварительно возможности вашего процессора в специальных таблицах (например, на разработанном мною [сайте mpystems.iu4.bmstu.ru](http://mpsystems.iu4.bmstu.ru)), после чего выставлять частоту намного увереннее и широкими

шагами). Если на каком-то значении процессор не запустился (черный экран), значит в таких условиях он работать не будет. Поставьте частоту на один шаг ниже и тестируйте систему на стабильность.

Бывают случаи, когда компьютер проходит процесс начальной загрузки (POST), но "повисает" после вывода таблицы с конфигурацией системы. Это поведение обуславливается невозможностью работы жесткого диска, PCI или AGP карт. Здесь можно помочь увеличением делителя соответствующих устройств, изменением в большую сторону режимов ожидания в SETUP'e, подачей на них более высокого напряжения.

Если система загрузилась, то надо протестировать ее на стабильность (очень глупо будет, только разогнав камень, сесть и писать доклад на научную конференцию, и после сбоя потерять все свои труды). Тестирование заключается в запуске многозадачной системы и большого количества приложений, активно работающих с процессором и памятью (здесь как никогда кстати подходят 3D игры). Если в течение часа никаких сбоев не появилось, то система полностью стабильна. Если же нет, то надо будет поставить частоту пониже, либо подумать о хорошем охлаждении. Иногда данная проблема связана с недостаточным напряжением (уровнем логического сигнала) и решается небольшим поднятием напряжения на ядре процессора (вот, где вступает в игру ваша хорошая материнская плата!).

1.1.5 Охлаждение

При увеличении частоты, а тем более при поднятии напряжения увеличивается мощность, потребляемая камнем. Он начинает намного сильнее греться. Надо отметить, что при повышении температуры ядра на 10 градусов Цельсия, срок службы процессора сокращается вдвое. Здесь в самую пору подумать об охлаждении. Для охлаждения процессора применяется ряд устройств:

1. Самым известным и распространенным из них является **кулер** (радиатор с вентилятором). Систему его работы я объяснять не буду – она предельно проста (чего не скажешь о теоретических расчетах для его изготовления).

2. Далее идет **термопаста**, обладающая высокой теплопроводностью. Ее наносят тонким слоем сверху на корпус процессора, а потом ставят кулер. Наиболее известная марка – КПТ-8.

3. Более совершенная система охлаждения – водяная. Особо "продвинутые" разгонщики делают ее самостоятельно в домашних условиях. Людей, отдающих предпочтение водяной системе охлаждения, на жаргоне оверклокеров (overclocker) называют "сантехниками", или "слесарями".

4. Довольно интересное устройство для охлаждения – полупроводниковый кулер **Пельтье** (рис. 3), названный так в честь своего изобретателя – французского часовщика Пельтье. Смысл явления был установлен и доказан в 1838 году Ленцем. Вот как объясняется это явление на физическом уровне: "В отличие от тепла Джоуля-Ленца, которое пропорционально квадрату силы тока ($Q=R \cdot I \cdot t$), тепло Пельтье пропорционально первой степени силы тока и меняет знак при изменении направления последнего.

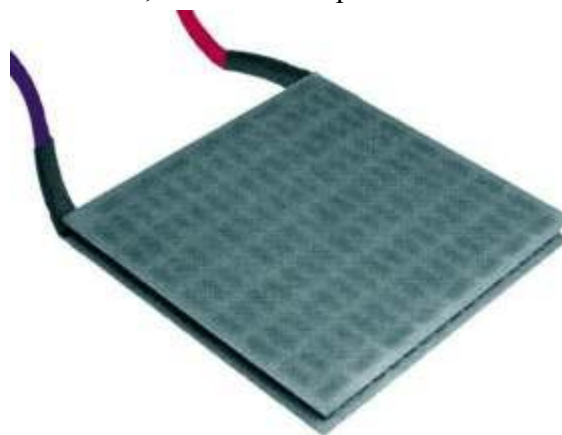


Рис. 3

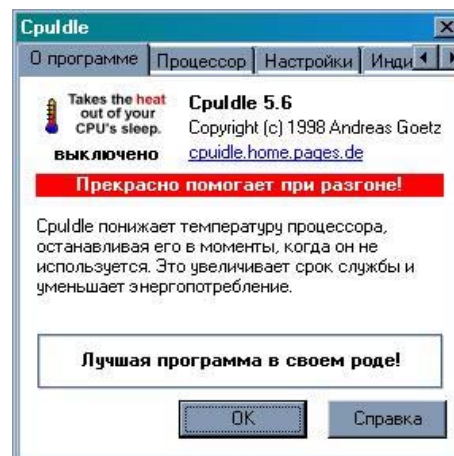
Тепло Пельтье, как показали экспериментальные исследования, можно выразить формулой: $Q_p = \Pi \cdot q$, где q - количество прошедшего электричества ($q=I \cdot t$), Π - так называемый

коэффициент Пельтье, величина которого зависит от природы контактирующих материалов и от их температуры... Модуль Пельтье, представляет собой термоэлектрический холодильник, состоящий из последовательно соединенных полупроводников р- и n-типа, образующих р-n- и n-p-переходы. Каждый из таких переходов имеет тепловой контакт с одним из двух радиаторов. В результате прохождения электрического тока определенной полярности образуется перепад температур между радиаторами модуля Пельтье: один радиатор работает как холодильник, другой радиатор нагревается и служит для отвода тепла".¹ Горячий радиатор охлаждают стандартным кулером. Этот тип охлаждающей системы довольно сложен в применении и используется профессиональными разгонщиками.

5. Охлаждать процессор можно также и программно. Программные кулеры запускаются уже после загрузки системы, поэтому сначала надо добиться устойчивого входа в активное состояние вашей ОС. Кулеры данного типа сделаны на одном простом принципе: остановке процессора (переходе его в спящий (suspend mode) режим) во время его простоя.

Рис. 4

Программные кулеры выполняют инструкцию HLT на правах ядра системы. Процессор выходит из спящего режима при получении инструкции NMI, при



поступлении прерывания, или при сигнале reset.² Примерный вид окна программного кулера представлен на рисунке 4.

6. Также важен в охлаждении устройств подбор корпуса, желательно, чтобы он имел два вентилятора высокой мощности и был довольно просторным.

Я бы хотел привести пример экзотического типа охлаждения. Кто-то из разгонщиков догадался, что верхний алюминиевый слой процессора проводит тепло хуже внутреннего медного. Вывод как всегда гениально прост: срезать этот "бесполезный" слой, отшлифовав нулевой - единичной наждачкой процессор, после чего желательно также отшлифовать нижнюю сторону радиатора. Как говорят опытные разгонщики, отвод тепла значительно улучшается.

Еще одна возможность понижения температуры процессора – понижение питания ядра. Этот способ подходит только для довольно качественных процессоров, так как при уменьшении питания снижается скорость переходных процессов, что негативно отражается на стабильности работы устройства.

1.1.6 Проблемы разгона

1.1.6.1 Краткая классификация

Теперь я бы хотел рассказать о других подводных камнях разгона, являющихся, может быть, даже более серьезными и трудно преодолимыми. Поднимая FSB, мы увеличиваем не только частоту процессора, но и других устройств, таких как видеокарта, контроллер жесткого диска, ISA устройства, оперативная память, кэш L2. С одной стороны это хорошо – поднимается общее быстродействие системы – к этому мы и стремимся, а с

¹ Ixbt.ru - iXBT: Полупроводниковые кулеры Пельтье.

² Современные ОС как Linux, Windows NT, Windows2000 автоматически выполняют инструкцию HLT, поэтому программные кулеры им не требуются – прим. автора.

другой - многие add-on карты могут просто не заработать и не дать процессору, какой бы он хороший ни был, разогнаться до требуемой характеристики.

1.1.6.2 Решения

Начну с ISA карт, как наиболее простых устройств. Решением проблемы их неработоспособности является установка максимального коэффициента деления частоты тактирования ISA шины (номинальная частота = 8 МГц) и задержек операций ввода/вывода (I/O Recovery) в сетаве.

PCI и AGP карты требуют аналогичных операций, с несколькими отличиями. Во-первых, современные PCI устройства довольно спокойно относятся к превышению частоты (41,5 МГц довольно реальная частота, при номинальной = 33 МГц). Во-вторых, PCI шина является очень важным элементом системы, так как на ней находится контроллер жесткого диска, и если не загружается операционная система, архивы создаются с ошибками и происходят аналогичные действия, то винить следует именно дисковую подсистему.

Чтобы избежать проблем с жестким диском:

- 1 Для подключения дисковых устройств используйте шлейфы минимальной длины.
- 2 Не устанавливайте жесткие диски, рассчитанные на разные режимы работы, на один шлейф (например, жесткие диски PIO3 и PIO4).
- 3 Устанавливайте CD-ROM привод на отдельный шлейф на второй (Secondary) IDE контролер.
- 4 Если это не помогает, попробуйте принудительно установить режим работы ваших HDD и CD-ROM приводов на ступень ниже. Например, если при установке в сетаве автоопределения режима работы HDD и CD-ROM приводов компьютер сообщает Вам, что ваш жесткий диск или CD-ROM привод работают в режиме PIO4, то отключите автоопределение и установите режим PIO3.

AGP видеокарты требуют увеличения делителя, лучшего охлаждения, или переконфигурирования в AGP 1x. Кстати, надо отметить, что именно эти видеокарты являются грозой разгонщиков. При покупке видеокарты надо внимательно изучить не только их "скоростные" характеристики, но и возможность стабильного функционирования на частотах выше 89 МГц.

1.1.7 Оперативная память

Для хорошего разгона потребуются приличная оперативная память. Приличная, значит такая, что сможет держать FSB, которую вы поставили. Память условно разбивается на спецификации, по которым можно узнать на какой номинальной частоте она может

работать. Номинальной эта частота называется потому, что очень часто "оперативка" может держать частоту тактирования значительно выше заявленной (например, у меня стоит РС66, которая работала без сбоев на 110 МГц!). Про память можно рассказать еще много интересного, но это уже отдельная тема. Вы сможете ознакомиться с ней более подробно, посетив разработанный мною [сайт](http://mpsystems.iu4.bmstu.ru) mpsystems.iu4.bmstu.ru.

1.1.8 Кэш L2

Довольно существенным подводным камнем является кэш второго уровня, поэтому на нем остановлюсь довольно подробно. Дело в том, что чаще всего он делается на пределе своих возможностей и очень болезненно реагирует на разгон. Это и понятно: кэш Level 2 (L2) построен на микросхемах статической памяти (SRAM), у которой возможна большая частота тактирования, чем у динамической памяти (DRAM), но все же заставлять кэш работать на 150-225 МГц, а иногда и на полной частоте процессора, когда SDRAM «тянет» не больше 133 МГц – это приличная нагрузка. Когда же конечный пользователь пытается разогнать свой процессор, он невольно разгоняет и кэш L2, результат нестабильной работы которого всем ясен: неправильная и несвоевременная загрузка страниц, бесконечные «промахи», а также перегрев, иногда приводящий к тому, что кэш сгорает, и из PentiumII процессор превращается в



Рис. 5

Celeron (конечно, если забыть о различиях в ширине шины адреса и др.). На рисунке 5 представлен кэш L2 с временем доступа 4.4 нс. (маркировка "-A44")

По соображениям прибыльности и эффективности производства процессоры одной марки производятся на одной конвейерной линии, при этом из-за перебоев в поставках, других непредвиденных проблем на процессоры с меньшей тактовой частотой ставят кэш, рассчитанный на старшие модели. Ниже будет описан один из таких случаев.

При переходе с ядра Klamath на ядро Deschutes появился дефицит на 266 и 300 МГц PentiumII. Эту «дыру» было поручено в спешном порядке закрыть заводу на Коста-Рике, который уже специализировался на самых дорогих процессорах PentiumII-450 МГц. На складе этого завода лежал кэш на 225 МГц, который и ставили на процессоры 300 МГц.

1.1.9 Практика разгона. Изобретение

Очень интересный момент подмечен мной из практики разгона. Если пытаться разогнать только что купленный процессор, то результат окажется ниже, чем если попытаться сделать то же самое через полгода – год. Примечательно, что если на процессор в течение длительного времени оказывается значительная нагрузка, он начинает разгоняться лучше. Такие нагрузки формируются искусственно, например, созданием 300-400 мегабайтных архивов, либо при обработке больших массивов векторной графики. Феномен такого поведения процессора тяжело объяснить полностью. Я могу лишь высказать предположение, что при большой нагрузке поднимается температура и, вследствие этого, на уровне кристаллической решетки происходит рекомбинация обратимых дефектов кристалла, что приводит к повышению его качества.

Разгон, из-за его "нагревающих" свойств, требует жесткого слежения за температурой. Мониторинг проводится с помощью специальных термодатчиков, которых обычно 2-3 штуки. Первый и основной находится под процессором, либо встраивается в сам камень. Второй отслеживает общую температуру системы, ну а последний (рис. 6) применяется либо для

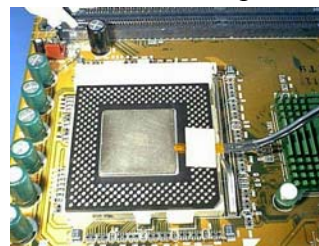
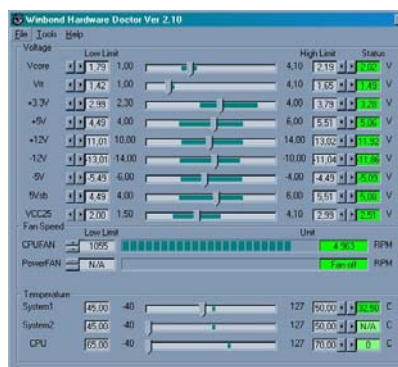


Рис. 6

мониторинга температуры второго процессора, либо любого другого места по выбору (например, видеокарты). Софтовую часть представляет либо BIOS, либо специальная программка-монитор, примерный вид которой представлен на рисунке 7. И вот тут то появляется проблема, которую пришлось решить самостоятельно, так как никто подсказать, даже на конференции, посвященной вопросам разгона, не мог. На моей материнской плате были предусмотрены все три термодатчика, но когда я включил компьютер, температура процессора не появлялась.

Проверив все заново, я включил питание опять – результат тот же. Все мои попытки узнать что-либо в технической поддержке не принесли результата. Сотрудник компании посоветовал обратиться на какую-нибудь конференцию, но и там, как вы уже знаете, я ничего выяснить не смог.



Пришлось подумать самостоятельно. Как мне удалось узнать, такая проблема появляется при использовании Slot1 системной платы с PPGA Celeron'ом и любым переходником. Логично предположить, что виновниками могли быть: материнская плата, процессор или переходник. На сайте производителя процессора Celeron, компании

Рис. 7

Intel, имеется огромный информационный файл о процессорах на ядре Mendocino. В этом документе есть маленькая страничка, посвященная встроенному термодиоду. На ней приводится несколько формул, краткое описание принципа работы и предостережение, что термодиод не протестирован на все сто процентов, поэтому Intel не может отвечать за его работоспособность. Методом исключения была отвергнута идея о виновности системной платы, так как обычные Slot1 процессоры прекрасно на ней работали и температуру показывали. Процессор отпал с 80% вероятностью, потому что его Slot1 собратья (Celeron) также не выявляли никаких проблем. Остался переходник, им я и решил заняться поплотнее. Здесь необходимо немного процитировать Intel:

NOTES:

1. Intel does not support or recommend operation of the thermal diode under reverse bias.
2. At room temperature with a forward bias of 630 mV.
3. n_ideality is the diode ideality factor parameter, as represented by the diode equation: $I-I_0(e^{(Vd*q)/(nkT)} - 1)$.
4. Not 100% tested. Specified by design characterization.

Table 30. Thermal Diode Interface

Pin Name SC242 Connector

Signal # 370-Pin Socket Pin # Pin Description

THERMDP B14 AL31 diode anode (p junction)

THERMDN B15 AL29 diode cathode (n junction)³

Очевидно, анод термодиода – контакт AL31 соединяется с контактом B14 Slot1, а катод AL29 – с B15. Вероятно, производители переходников либо по своей лени, либо, испугавшись грозного предупреждения Intel, просто не развели эти контакты. Вывод напрашивается сам собой – допаять эти 2 проводка.

Теперь техника. Берется переходник, переворачивается обратной стороной (там, где распаяны контакты сокета, предварительно, естественно процессор надо вынуть). После этого достается проводок, от него отрезается требуемая длина, потом, если жилки довольно толстые, вытягивают 1-3 штуки. Если жилок на один контакт больше одной, то они скручиваются. Далее облуживаем контакты и помещаем их в хлорвиниловые трубки, приготовленные заранее, на 2-3 см короче проводков. Это делается для того, чтобы жилки не "коротили" с корпусом переходника. После чего специальным паяльником проводки аккуратно припаиваются к соответствующим контактам переходника (B14 и B15). Потом противоположные концы паяют к сокетным контактам.

Хочу сказать, что подобная операция проводится на ваш страх и риск.

1.1.10 Предел разгона

Конечно, он есть. Помимо всех вышеперечисленных факторов, лимитирует разгон физическое строение кристалла (0.25 мкм, 0.18 мкм), поэтому разгонять процессор бесконечно нельзя – надо переходить на более совершенную логическую структуру.

1.1.11 Статистика. Выбор процессора для разгона

В Интернете есть сайты с огромными базами данных разогнанных процессоров разных марок (например, www.overclockers.com). Зная примерно модель, которую вы хотите купить, можно получить большую сводную таблицу о 100-150 процессорах этой марки, когда-либо разогнавшихся. Проанализировав эту информацию (ставя, например, "+" каждому процессору с определенной серией, разогнавшемуся до нужной вам величины, и "-" тем, которых разогнать не удалось), вы можете твердо, почти со 100% уверенностью выбирать себе облюбованный процессор.

Как я говорил ранее, пробные образцы процессоров проходят жесткое тестирование, после чего вся партия маркируется по их результатам. У каждого процессора есть серия. Она нанесена, в зависимости от типа процессора, сверху на чипе, либо сбоку на картридже и выглядит примерно так:

BER 056258 FV524 **RX366 128 SL36C MALAY**
L9401153-0341 M'C'98

Жирным шрифтом выделена самая важная информация в серийном номере.

1. **RX366** – частота процессора (366 МГц),
2. **128** – наличие кэша L2 в объеме 128 килобайт,

Рис. 8

3. **SL36C** – марка,

4. **MALAY** – страна производитель (в данном случае – Малайзия),

5. **L9401153-0341** – номер (40 – неделя изготовления 1153 – номер партии, -0341 – номер процессора в партии).



1.1.12 Разгон – один из основных критериев выбора процессора

Я думаю, что большинство пользователей, ознакомившись с данной работой, смогут более осмысленно подойти к проблеме выбора оптимальной конфигурации компьютерной

³ Intel – 24365811.pdf

системы. Но хочу отметить, что смысл разгона намного шире. Разгон – это новый инструмент для проектирования и тестирования архитектуры электронно-вычислительной аппаратуры и при приобретении ее конечным пользователем. Позвольте привести небольшой пример. Если спросить компьютерно-грамотного человека, который никогда не слышал о разгоне, какой процессор лучше купить: PentiumII-400 или Celeron-366 – он, безусловно, скажет, что PentiumII-400, если, конечно, нет определенных материальных трудностей. Исходя из его критериев выбора комплектующих – это единственно правильный ответ. Но любой, даже менее компьютерно-грамотный человек, знающий о разгоне, даст противоположенный ответ. Celeron-366 дешевле, разгоняется до 550 МГц (что в большинстве приложений эквивалентно PentiumIII-450!), а PentiumII-400 стабилен на частоте не выше 450 МГц. Эти выводы относятся и к другим типам комплектующих.

1.2. Выводы: зачем нужен и что дает нам разгон

1. Позволяет оптимизировать выбор процессора с учетом возможности увеличения его быстродействия.
2. Экономит приличные средства на модернизацию компьютера, помогает вдохнуть в старое "железо" новую жизнь.
3. Люди, увлекающиеся разгоном, узнают много нового и интересного об устройстве ПК.
4. Любой пользователь становится активным участником разработки и тестирования аппаратного обеспечения ПК.
5. Разгон является мощным профессиональным инструментом для тестирования микропроцессорных архитектур.

2. Разработка и эксплуатация Информационно-Справочной Интернет Системы MP-SYSTEMS

2.1 Структурно-функциональное построение информационно-справочной интернет системы MPSYSTEMS.

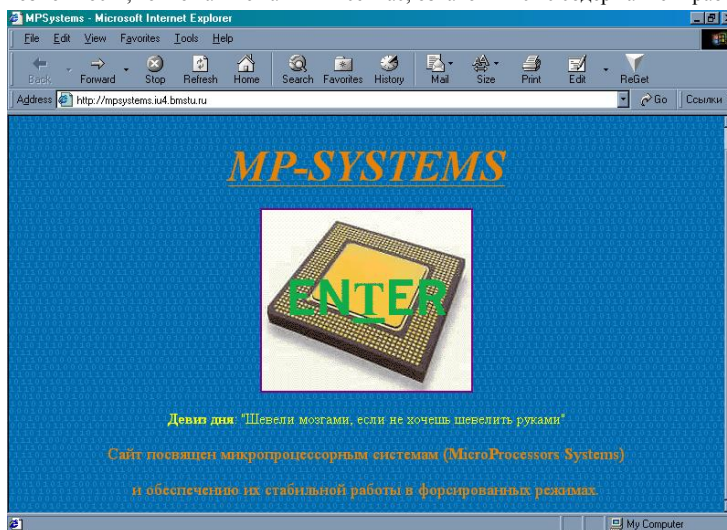
Я думаю, что общего описания принципов разгона достаточно для понимания его сути, поэтому перейду к той части работы, которая посвящена разработке и эксплуатации Информационно-Справочной Интернет Системы (ИСИС) MP-SYSTEMS.

Как уже говорилось выше, ИСИС была сделана для расширения аудитории пользователей, которым доступен доклад. Любой пользователь, будь он хоть из Магадана, имеет возможность, точно так же как и Вы сейчас, ознакомиться с содержанием работы.

Итак, он набирает в адресной строке браузера mpsystems.iu4.bmstu.ru и попадает на главную страницу:

Рис. 9

Далее с помощью встроенной системы навигации он переходит к основному разделу:



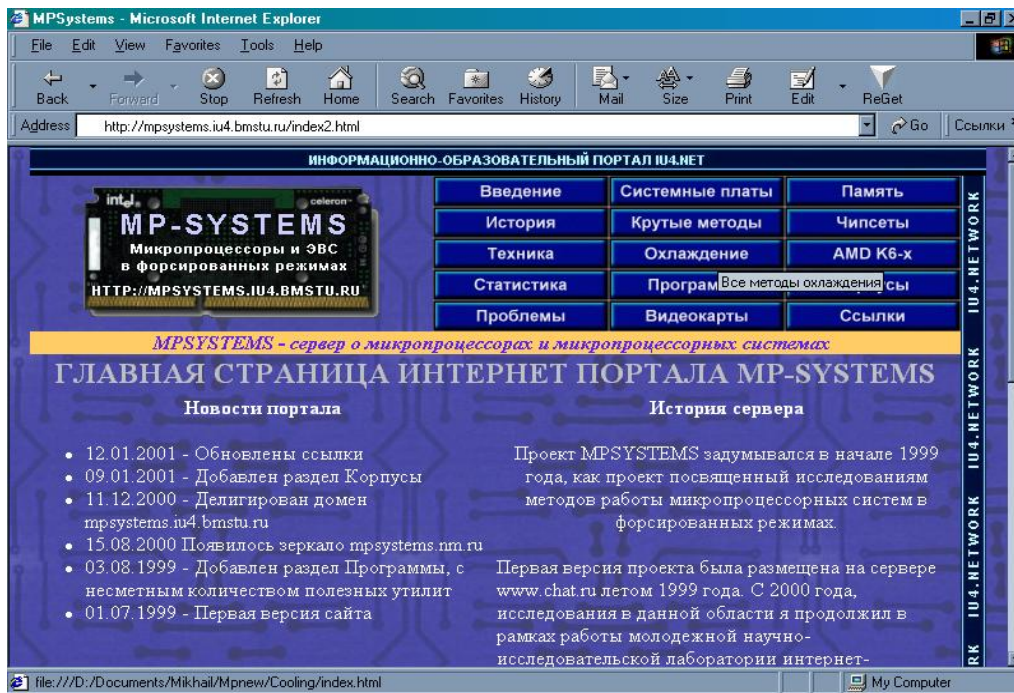


Рис.10.

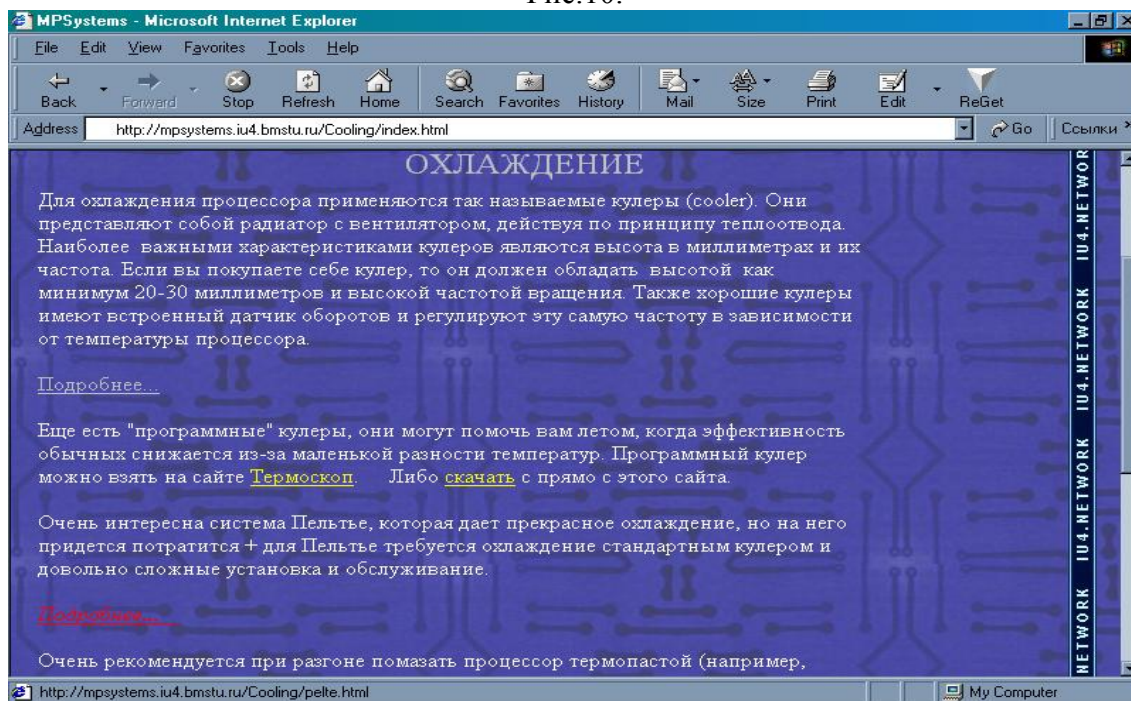


Рис.11.

Слева загружается логотип, а справа удобная встроенная в сайт система навигации. В ней представлены основные разделы ИСИС. ИСИС имеет разветвленную структуру. Например, при выборе раздела "Охлаждение" появляется страница с общим описанием всех возможностей охлаждения микропроцессорных систем и ссылками на конкретные статьи (рис. 11)

Для удобства пользователя статьи разделены на две части. Первая часть – краткое описание, достаточное для того, чтобы сразу произвести все операции, не вникая в их суть. Для тех же, кто хочет подробнее изучить конкретный раздел, существует специальная ссылка: "[Подробнее...](#)" Щелкнув на ней мышкой, пользователь переходит к подробной статье, с множеством таблиц, диаграмм и рисунков:

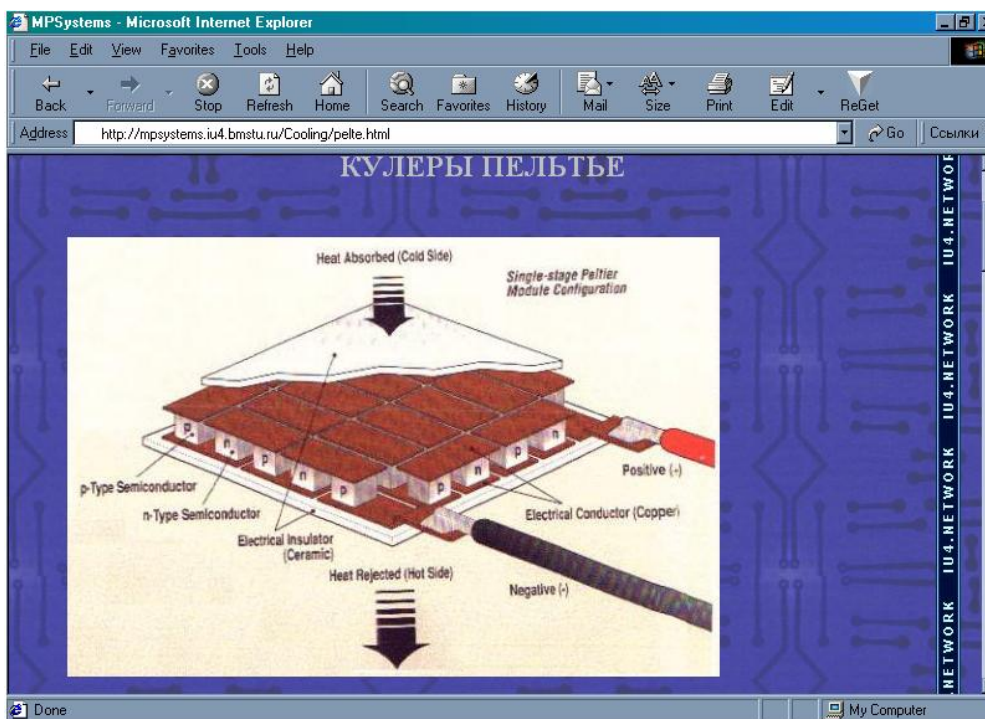


Рис.12.

Чтобы любой пользователь мог представить структуру ИСИС, на ней специально создан раздел: "Карта сервера", где в визуальной форме представлена вся конструкция ИСИС.

Для удобства пользователя с помощью языка интернет-программирования JAVA-SCRIPT созданы "живые" ссылки. Ссылки, которые еще не просматривались пользователем, имеют желтый цвет, контрастно выделяющийся на фиолетовом фоне. При наведении указателя мышки на какую-либо ссылку, последняя становится красной и изменяет шрифт, тем самым достигается эффект "оживления" статичной ссылки. Просмотренные ссылки меняют цвет на серебряный.

Теперь вернемся к основному разделу. Если немного прокрутить страничку с помощью полосы прокрутки, можно увидеть колонку новостей, содержащую последние наиболее значимые новости Интернет-портала MP-SYSTEMS и новости из мира разгона и микропроцессорной техники.

Этот сервер я администрирую лично, постоянно его изменяя. Поэтому вполне возможно, что структура сайта может немного отличаться от описанной в этой работе. Я бы хотел особенно подчеркнуть, что в отличие от простого "бумажного" доклада Информационно-Справочная Интернет Система обладает динамичностью и возможностью непрерывного развития, что, несомненно, указывает на ее перспективность и направленность в будущее.

2.2 Выводы:

Чем полезен Интернет-Проект и его основные преимущества:

1. Возможность ознакомления с докладом миллионов русскоязычных пользователей Интернета из любой точки мира.
2. Динамичность и способность к непрерывному развитию.
3. Актуальность и свежесть информации.
4. Удобство навигации.
5. Улучшенное восприятие информации пользователем (мультимедиа, гамма цветов, контрастные схемы, логичность построения).
6. Возможность внедрения тематических конференций, в которых пользователи могут обсуждать материалы сайта и обмениваться знаниями и опытом (в перспективе).
7. Количество копий научной работы практически неограниченно.
8. Быстрый и эффективный поиск необходимой информации.

Заключение

В дальнейших планах Интернет-портала MPSYSTEMS реализация средств интерактивного взаимодействия пользователя с Информационной Системой, предоставление ему возможностей контроля за режимами работы тестовых микропроцессорных систем в on-line режиме, автоматическое предложение вариантов разгона для конкретного микропроцессора.

Перспективы: Почему я хочу поступать на ИУ4

Ответ прост: чтобы продолжать заниматься любимым делом, в которое вложил уже много сил и интеллекта. В школьной научно-исследовательской лаборатории кафедры, где меня обучили многому: от HTML до создания LINUX – серверов. Мне очень понравилось на ИУ4, потому что я понял, что здесь смогу реализовать свой творческий потенциал в полной мере. Хочу обязательно поступить в аспирантуру и написать диссертацию (может быть, по специальности, связанной с вычислительной и микропроцессорной тематикой). Конечно, это будет уже в будущем. Но ведь будущее мы строим сегодня!

Выводы по научно-исследовательской работе:

Зачем нужен и что дает нам разгон:

1. Позволяет оптимизировать выбор процессора с учетом возможности увеличения его быстродействия.
2. Экономит приличные средства на модернизацию компьютера, помогает вдохнуть в старое "железо" новую жизнь.
3. Люди, увлекающиеся разгоном, узнают много нового и интересного об устройстве ПК.
4. Любой пользователь становится активным участником разработки и тестирования аппаратного обеспечения ПК.
5. Разгон является мощным профессиональным инструментом для тестирования микропроцессорных архитектур.

Чем полезен Интернет-Проект и его основные преимущества:

6. Возможность ознакомления с докладом миллионов русскоязычных пользователей Интернета из любой точки мира.
7. Динамичность и способность к непрерывному развитию.
8. Актуальность и свежесть информации.
9. Удобство навигации.
10. Улучшенное восприятие информации пользователем (мультимедиа, гамма цветов, контрастные схемы, логичность построения).
11. Возможность внедрения тематических конференций, в которых пользователи могут обсуждать материалы сайта и обмениваться знаниями и опытом (в перспективе).
12. Количество копий научной работы практически неограниченно.
13. Быстрый и эффективный поиск необходимой информации.

Список использованных источников

1. www.ixbt.ru
2. cdl.iu4.bmstu.ru
3. www.triptonics.com
4. www.conf.ru
5. forum.hardware.ru
6. www.tomshardware.com
7. www.overclockers.com
8. www.intel.com
9. www.amd.com
10. www.abit.com.tw
11. Михаил Гук Процессоры PentiumII Pentium Pro и просто Pentium. – Спб.: Издательство Питер, 1999
12. Л. Н. Преснухин В. А. Шахнов Конструирование Электронных Вычислительных Машин и Систем. – М.: Издательство Высш. Шк., 1986.

АНАЛИЗ КРИТЕРИЕВ ОПТИМАЛЬНОСТИ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ДЕЙСТВИЙ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

Добрецов С. В.

д.т.н., профессор Птицына Л. К.

**Санкт-Петербургский государственный технический университет, Санкт-Петербург,
Россия**

AN APPROACH TO OPTIMIZATION ANALYSIS OF INTELLIGENT AGENTS PLANNING ALGORITHMS

S. V. Dobretsov

Professor, Dr. L. K. Ptitsyna

St. Petersburg State Technical University, St. Petersburg, Russia

Dobretsov@mail.ru

Аннотация Структурированный подход к оценке и анализу алгоритмов планирования действий, используемых интеллектуальными агентами с выделением двух уровней показателей: универсальных, применимых ко всем алгоритмам планирования, и специфических, учитывающих характерные особенности конкретного класса алгоритмов. В качестве примера проводится анализ специфических показателей для класса нелинейных алгоритмов планирования с представлением среды в пространстве состояний.

Abstract Structural approach to optimization analysis of planning algorithms used by intelligent agents with evolving of two layers of evaluation parameters: universal, applicable to all planning algorithms, and algorithm-specific, peculiar to the particular group of algorithms. As an example, detailed analysis of specific parameters for the class of nonlinear algorithms with partial plan representation is performed.

Изучение проблемы построения планов действий являются важной составляющей теории решения задач и занимают одно из ведущих мест в области исследования интеллектуальных систем [1]. Однако подавляющее большинство опубликованных в рамках данной тематики работ посвящено разработке новых алгоритмов планирования, тогда как вопросы сравнения существующих алгоритмов и выбора среди них наиболее оптимального рассматриваются крайне редко [5]. Проведенное в рамках данной работы исследование предоставляет необходимые основы для подобного анализа, в частности набор базисных показателей для оценки и сравнения различных алгоритмов планирования. Выделенные показатели позволяют построить разнообразные критерии оптимальности для определения выбора наилучшего алгоритма планирования в зависимости от специфики прикладной системы.

Анализ существующих реализаций алгоритмов планирования [2], а также немногочисленных подходов к их сравнению [4], позволяет выделить две группы показателей, позволяющих проводить оценку алгоритмов планирования:

1. Универсальные характеристики, отражающие наиболее общие высокоуровневые свойства систем планирования.
2. Специфические показатели, использующие характерные особенности, присущие только системам определенного класса.

Показатели первой группы оценивают наиболее важные с прикладной точки зрения аспекты работы алгоритмов планирования (а также систем, их реализующих) и являются применимыми практически ко всем системам планирования независимо от используемого в них подхода. К подобным характеристикам относятся:

1. Функциональность: способность системы находить решения поставленных перед ней задач планирования. Численно может быть определена как отношение числа задач, правильно решенных системой, к общему числу рассмотренных задач.
2. Качество предлагаемого решения: группа характеристик, отражающих свойства найденного плана, среди которых наиболее часто используемыми являются:
 - Стоимость исполнения плана отождествляемая в простейшем случае с длиной плана.
 - Степень достижения целей, представленная с использованием некоторой метрики предметной области, отражающей близость состояний друг от друга (например, пропорционально количеству совпадающих или несовпадающих переменных).
3. Вычислительная сложность используемого алгоритма: теоретические оценки принципиальных временных (гарантированность нахождения решения за линейное, полиномиальное или экспоненциальное время) и емкостных (требования к вычислительной памяти) ограничений для данного алгоритма.
4. Быстродействие и ресурсоемкость, определяемые, в отличие от вычислительной сложности, экспериментальным путем посредством решения значительного количества тестовых задач. В рамках данной категории обычно выделяют такие параметры как: среднее, минимальное и максимальное время решения одной задачи данной системой;
5. Предсказуемость: возможность заранее прогнозировать поведение системы при решении конкретной задачи планирования с высокой степенью достоверности.

Рассмотрение специфических показателей имеет смысл только при анализе систем, принадлежащих к некоторому общему классу. Как правило, их практическая важность является величиной меньшего порядка по сравнению со значением универсальных показателей. Тем не менее, необходимость исследования специфических особенностей алгоритмов планирования определяется следующими причинами:

1. При анализе систем планирования, относящихся к одному классу, возникает необходимость в средствах более детального анализа.
2. Рассмотрение специфических особенностей каждого алгоритма планирования позволяет изучить влияние различных дополнительных факторов на значения высокоуровневых показателей системы планирования, таких как вычислительная сложность, быстродействие, предсказуемость.

В качестве примера рассмотрим класс алгоритмов нелинейного планирования с представлением среды в пространстве состояний [2]. Алгоритмы данного класса могут быть представлены в следующей обобщенной форме:

- 1: Проверка условия окончания.
- 2: Уточнение текущего частичного плана (выбор и достижение очередной подцели).
- 3: Оптимизация текущего частичного плана.
- 4: Проверка состоятельности текущего частичного плана.
- 5: Рекурсивный вызов алгоритма.

Для конкретизации анализа необходимо сделать ряд предположений, которые, как правило, являются естественными для большинства прикладных задач планирования и обеспечивают ограниченность пространства поиска:

1. Предметная область имеет конечное число возможных состояний.
2. Рассматриваемые задачи планирования имеют решения.
3. Целью функционирования системы является поиск минимального (в том смысле, что любое другое решение не может быть получено из данного решения путем удаления одного или нескольких шагов) решения задачи.

Легко видеть, что для рассматриваемого класса алгоритмов процесс поиска решения может быть представлен в виде дерева, каждой вершине которого соответствует с одной стороны некоторый набор потенциальных решений, а с другой стороны – рекурсивный вызов алгоритма планирования. На рис. 3.1 представлен фрагмент дерева поиска, в котором символами α , β , γ , δ , ϵ ,... обозначены потенциальные решения, отвечающие каждой вершине. Штрихами отмечены потенциальные решения, не являющиеся решениями исходной задачи

планирования, а символом \emptyset - вершины, в которых соответствующий вызов алгоритма обнаружил, что текущий частичный план не имеет решения. В зависимости от реализации

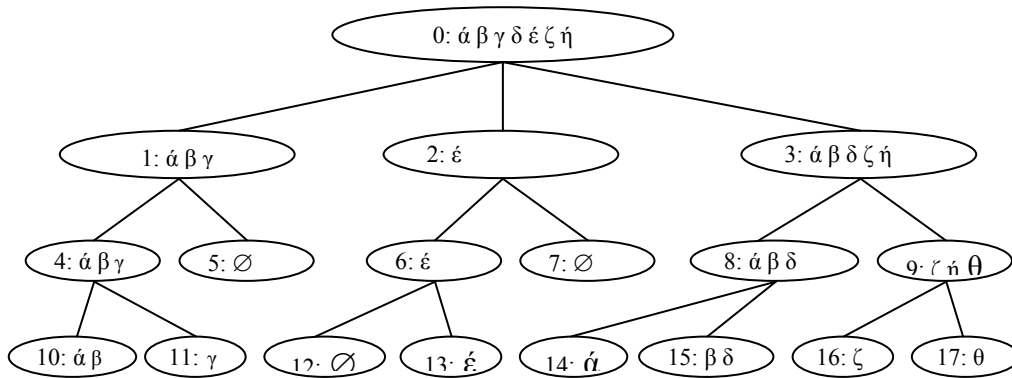


Рис. 1. Схема дерева поиска, выполняемого алгоритмом планирования.

алгоритма планирования, дерево поиска может являться избыточным, то есть одно и то же потенциальное решение может соответствовать нескольким ветвям поиска (в изображенном примере такими решениями являются α и β).

Для удобства дальнейшего анализа обозначим через F множество вершин-листьев (вершин, не имеющих ветвлений), а через K – множество всех потенциальных решений (в приведенном примере $K = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta\}$). Подобное представление позволяет выделить следующие специфические показатели, присущие данному классу алгоритмов планирования:

1. Степень ветвления дерева поиска b : среднее количество ветвлений в одной вершине. В общем случае, степень ветвления является произведением двух составляющих:

$$b = b_{\text{уточ}} b_{\text{опт}}$$

где $b_{\text{уточ}}$ – количество способов достижения выбранной подцели на этапе уточнения частичного плана, а $b_{\text{опт}}$ – количество вариантов, получаемых из уточненного частичного плана на этапе оптимизации. Если алгоритм не использует промежуточную оптимизацию, то $b_{\text{опт}} = 1$.

2. Глубина дерева поиска d , необходимая для нахождения решения, значение которой может быть оценено следующим образом:

$$d \geq R N_{\text{pre}} S$$

где S – количество шагов в решении задачи планирования, а N_{pre} – среднее количество предусловий одного оператора. Поскольку некоторые алгоритмы могут рассматривать одну и ту же подцель многократно в процессе решения задачи, необходимо также ввести корректирующий коэффициент R .

3. Степень избыточности дерева поиска ρ , определяемая как отношение суммы числа потенциальных решений, соответствующих каждой вершине в F к общему количеству потенциальных решений (мощности множества K). При $\rho = 1$ каждому потенциальному решению отвечает единственная вершина в F , значение $\rho > 1$ означает, что некоторое потенциальное решение может быть получено из нескольких вершин. Если алгоритм планирования обладает свойством систематичности, то для него всегда $\rho = 1$.
4. Показатель состоятельности вершин дерева поиска q , определяемый как отношение непустых вершин в F к общему количеству элементов в F . Появление пустых вершин на более ранних уровнях дерева поиска способствует значительному сокращению общего числа вершин и, следовательно, размеру пространства поиска.
5. Вычислительная стоимость $T_{\text{алг}}$ одного вызова алгоритма (отождествляемая в простейшем случае с временем выполнения) каждого шага алгоритма в отдельности.

С помощью выделенных показателей могут быть получены некоторые оценки производительности алгоритма планирования. Согласно [3] дерево поиска глубины d алгоритма описывается уравнением:

$$|F_d| = \frac{|K| \rho_d}{k_d} = O((bq)^d)$$

где k_d – среднее количество потенциальных решений, соответствующих одной вершине в F , а $|K|$ – количество элементов (мощность) множества K .

Тогда вычислительная стоимость перехода в дереве поиска с уровня $d-1$ на уровень d оценивается следующим образом:

$$T'_d = q |F_{d-1}| T_{\text{алг}} = q \frac{|K| \rho_{d-1}}{k_{d-1}} T_{\text{алг}} = O(b^{d-1} q^d T_{\text{алг}})$$

Общая стоимость поиска до глубины d определяется как:

$$T_d = \sum_{n=0}^d T'_n = q T_{\text{алг}} \sum_{n=0}^d |F_n| = \begin{cases} O\left(q \frac{(bq)^{d+1} - 1}{bq - 1} T_{\text{алг}}\right), \text{ при } bq > 1 \\ O(d T_{\text{алг}}), \text{ при } bq = 1 \end{cases}$$

На основании полученной оценки можно сделать предварительные заключения о степени влияния различных факторов на процесс решения задачи планирования. В частности, скорость решения задачи планирования зависит:

- Линейно от вычислительной стоимости одного вызова алгоритма;
- Полиномиально от степени ветвления b и показателя состоятельности вершин q ;
- Экспоненциально от глубины дерева поиска.

Представленные заключения позволяют сделать вывод о том, что увеличение вычислительной стоимости каждого вызова алгоритма из-за включения в него дополнительных шагов (таких как консервация полученных ранее результатов и дополнительная оптимизация) должно быть с избытком компенсировано уменьшением стоимости решения всей задачи планирования в первую очередь за счет уменьшения глубины дерева поиска. При этом может быть показано, что для рассматриваемого класса алгоритмов стоимость одного вызова алгоритма планирования находится в полиномиальной зависимости от размерности среды. Однако приведенные заключения являются общими и могут служить основой для сравнительного анализа алгоритмов планирования только при условии учета других важных факторов, таких как:

1. Стратегия поиска – набор правил, регулирующий порядок обхода дерева поиска. Обычно выделяют два типа поиска: “поиск в глубину” и “поиск в ширину”.
2. Дополнительные требования, определяемые спецификой задачи планирования, например, поиск хотя бы одного решения, поиск всех решений или поиск оптимального в некотором заданном смысле решения задачи планирования.

Литература

1. Искусственный интеллект. Кн. 2. Модели и методы // Под ред. Д.А.Поспелова. М: Радио и связь, 1990.
2. Добрецов С.В., Шестаков С.М. Планирование действий в искусственном интеллекте // ВАТТ СПбГТУ, №2 1998 г.
3. Kambhampati S., Knoblock C., Yang Q. Planning as Refinement Search: A Unified framework for evaluating design tradeoffs in partial order planning // Artificial Intelligence. Special issue on Planning and Scheduling. Vol. 76. No. 1-2, September 1995. pp. 167-238.
4. AIPS98 Planning Competition Results // Carnegie-Mellon University, 1998
5. Weld D. Recent Advances in AI Planning // AI Magazine, 1999

АНАЛИЗАТОР ШИНЫ CAN

Лебедев Е.В.

к.т.н. доц. каф. Вычислительной техники Салит В.В.
Томский политехнический университет, г. Томск, Россия

CAN ANALYZER

Lebedev E.V.

Assoc. Prof. Salit V.V.

Tomsk Polytechnic University, Tomsk city, Russia
e-mail: evgenl@ce.cctpu.edu.ru

Аннотация В результате проведенного анализа была установлена актуальность проблемы быстрой разработки, диагностики и технического обслуживания CAN-сетей. Выполнение данных функций возлагается на созданный на кафедре Вычислительной техники программно-аппаратный комплекс "CAN-анализатор". В статье кратко описана актуальность поставленной задачи, а также описаны основные функции разработанного комплекса.

Abstract As a result of analysis was specified actuality of problem of fast development, diagnostics and maintenance of CAN-networks. The execution of given functions are assigned on created on faculty of computer engineering the hardware-software complex called "CAN-analyzer". In paper is briefly circumscribed actuality of the set task, and also the trial functions of the designed complex are circumscribed.

Проникновение компьютеров и микропроцессоров в нашу жизнь становится всеобъемлющим. Само название "интеллектуальное устройство" однозначно говорит о присутствии в нем микропроцессора. Сегодня уже вполне реально строить системы автоматизации с участием некогда абсолютно пассивных, а ныне способных к "общению" разнообразных механических клапанов и задвижек. Открытие межгосударственных, и прежде всего экономических границ дало возможность отечественным специалистам не только познакомиться с самыми современными технологиями в области построения систем комплексной автоматизации, но и активно их использовать.

Основная цель построения распределенных систем автоматизации -- удешевление и упрощение технологий производства и эксплуатации системы автоматизации за счет, в частности, обеспечения сквозного сетевого доступа: от мощных супервизорных компьютеров и многофункциональных контроллеров до интеллектуальных пассивных элементов (датчики, регуляторы и т. п.). При этом такая связь должна удовлетворять всем современным требованиям по функциональности, надежности и открытости. На Западе коммуникационная технология построения единой информационной сети, объединяющей интеллектуальные контроллеры, датчики и исполнительные механизмы, определяется одним термином *fieldbus* – полевая шина, или промышленная сеть [1].

В связи со все увеличивающимся количеством распределенного управления в автомобилях и ростом стоимости проводки бортовой электроники, первостепенным фактором стало наличие мощной и надежной системы связи для обмена последовательными данными (сообщениями) между различными блоками управления. Это послужило отправной точкой для фирмы BOSCH, ведущего поставщика электронного оборудования для автомобилей, при разработке CAN-протокола и его стандартизации в международный стандарт ISO 11898. CAN (Control Area Network) – последовательный коммуникационный протокол, который эффективно поддерживает распределенное управление в реальном времени с очень высоким уровнем безопасности. В 1989 году фирмой INTEL был выпущен первый чип CAN-контроллера (82526). Сегодня 17 изготовителей, включая всех основных

производителей полупроводников, предлагают более 65 реализаций контроллеров CAN-протокола в форме автономных CAN-контроллеров или встроенных в микроконтроллер. Уже к 1999 году было инсталлировано более 70 миллионов CAN-узлов, причем в ближайшее время спрос резко увеличится. Уже сегодня CAN-интерфейс можно считать стандартным последовательным интерфейсом микроконтроллеров, используемым в любых типах распределенных встроенных приложений.

Благодаря выдающимся особенностям CAN-протокола, доступности недорогих сетевых контроллеров от многих изготовителей, а также простоте осуществления разработки сетевого решения, CAN-протокол сегодня применяется не только в практически любом типе мобильной системы (легковые автомобили, грузовики и автобусы, сельскохозяйственное оборудование, суда, самолеты, подъемники), но также в любых типах машин и механизмов - от ткацких, упаковочных, печатных машин до любых видов медицинского оборудования или систем управления роботами. CAN хорошо подходит для организации сети устройств с искусственным интеллектом. Благодаря доступности высокоуровневых коммуникационных стандартов и профилей, создание CAN-базирующихся сетей становится одним из самых многообещающих решений для открытых распределенных систем автоматизации, которые очень успешно конкурируют с другими стандартами в области индустриальной автоматизации [2].

Исходя из всего вышесказанного достаточно актуальной становится проблема быстрой разработки, диагностики и технического обслуживания CAN-сетей различной конфигурации и различного назначения. Выполнение данной функции возлагается на программно-аппаратный комплекс, называемый анализатором шины CAN или CAN-анализатором. На кафедре ВТ были проведены работы по созданию собственного варианта CAN-анализатора. Еще одним фактом в пользу собственной разработки анализатора являлось то, что стоимость зарубежных аналогов достаточно высока, а и их доступность на российском рынке и в настоящее время явно не является достаточной [3].

В качестве аппаратной части такого комплекса используется микроконтроллер обеспечивающий взаимодействие персонального компьютера с шиной CAN. Для достижения большей универсальности анализатор должен обеспечивать работу с несколькими моделями микроконтроллеров различных фирм производителей. Так как на рынке существует большое разнообразие контроллеров, различающихся по степени своей интеллектуальности, то в зависимости от применяемого аппаратного обеспечения может и изменяться алгоритм работы программной части. Отладка программной части может быть выполнена вообще без присутствия каких-либо аппаратных средств. В этом случае исходные данные для работы читаются из файла. На рисунке 1 изображено главное окно анализатора.

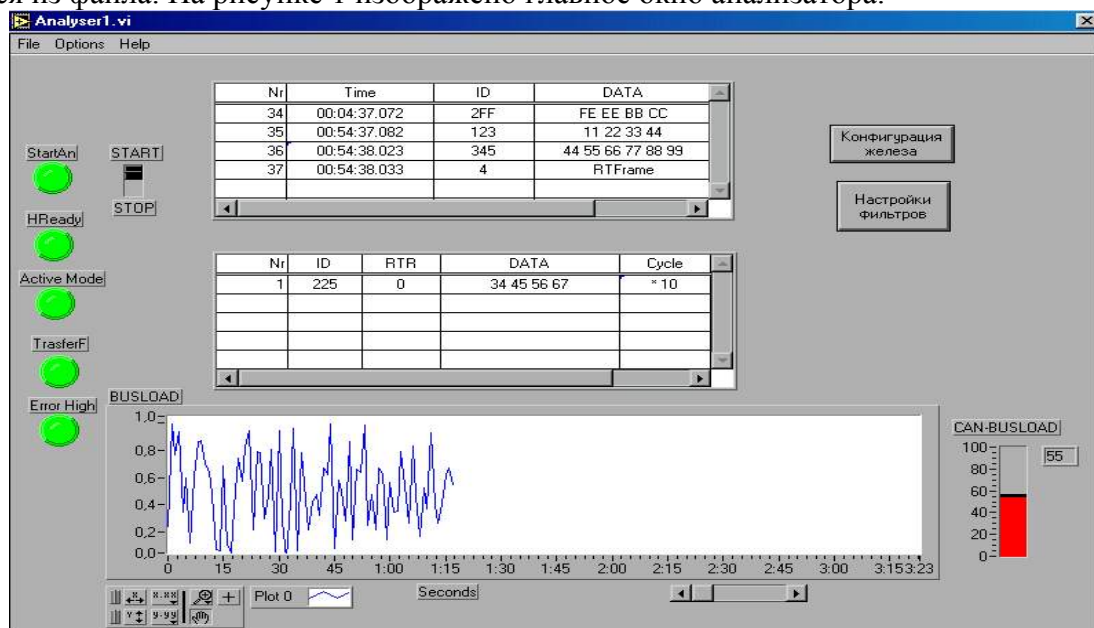


Рис.1. Главное окно анализатора

Выполнение всех основных аналитических функций комплекса возлагается на его программную часть. Под основными функциями понимаются такие функции:

1. поддержка различных версий спецификаций шины CAN;
2. обеспечение высокой производительности даже на максимальной скорости работы шины CAN (1Мбит/с);
3. активный и пассивный режимы работы;
4. обнаружение и мониторинг кадров ошибки;
5. прием и отображение принятых данных в последовательной форме со штампом времени;
6. наличие конфигурируемого фильтра объектов: указание получаемых идентификаторов по точному значению идентификатора или определению маски фильтра;
7. посылка сообщений в сеть, в том числе и циклическая;
8. отображение величины загрузки шины в данный момент и течении периода времени;
9. сохранение всей принятой информации на жесткий диск.

Поддержка различных версий спецификации шины CAN подразумевает поддержку двух спецификаций шины CAN2.0A и CAN2.0B отличающихся друг от друга форматами используемых кадров данных. Обеспечение поддержки обеих версий продиктовано расширением области использования разрабатываемого комплекса.

Обеспечение высокой производительности означает способность комплекса нормально функционировать в CAN-сетях с высокой скоростью обмена данными. Максимальная скорость передачи в сети CAN составляет 1Мбит/с, что требует от комплекса либо наличия высокой скорости обработки либо введения различных буферов.

Под активным режимом работы понимается такой режим работы при работе в котором анализатор становится полноправным узлом шины CAN и участвует в процессе передачи данных по шине. В пассивном режиме осуществляется только мониторинг сети без участия в процессе передачи. Используя анализатор, работающий в активном режиме можно использовать комплекс как вместо уже существующего узла, так и добавляя в сеть новый узел с легко изменяемыми настройками фильтров.

Обнаружение и мониторинг кадров ошибки позволяет судить о частоте возникновения ошибок в сети и на основе полученной статистики, накопленной в различных режимах работы анализатора и при различных настройках фильтров, сделать вывод о наиболее вероятном месте возникновения неисправностей.

Независимо от того в каком режиме работает комплекс он постоянно отображает данные передаваемые по шине другими узлами. Для упорядочивания этих данных, а также для облегчения ориентировки в них каждому кадру присваивается штамп времени, который соответствует относительному времени приема кадра. Все принятые кадры отображаются на экране последовательно в соответствии с присвоенными им штампом времени.

Наличие конфигурируемого фильтра объектов предоставляет пользователю анализатора богатые возможности фильтрации. Любой кадр данных в сети CAN содержит поле идентификатора (ID) и поле передаваемых данных (DATA). Поле идентификатора состоит из 11 бит по спецификации CAN2.0A и 29 бит по спецификации CAN2.0B. При настройке фильтра каждый из этих бит может принимать три состояния: 1, 0 или X. Состояние X означает что данный бит может принимать значение как единицы так и нуля. Таким образом, фильтр настраивает анализатор как на прием кадров с определенным идентификатором, так и на прием кадров из заданного диапазона идентификаторов. Количество передаваемых байт в одном кадре может быть от 0 до 8. Для каждого из байт можно применить аналогичную процедуру фильтрации как и для идентификатора, что позволяет анализатору принимать кадры с заданным содержанием независимо от их идентификатора. Оба способа можно комбинировать, т.е. принимать только кадры с заданным содержанием и идентификаторами лежащими в определенном диапазоне.

Работая в активном режиме анализатор может посылать в сеть кадры с различными идентификаторами и содержанием. В результате таких передач можно судить о способности различных узлов принимать предназначенные им данные. Так как скорость передачи по шине CAN достаточна высока, то иногда необходимо осуществлять циклический повтор определенного кадра или определенной группы кадров и соответственно, анализатор должен обеспечивать пользователю такую возможность.

Под отображением загрузки шины в данный момент понимается наличие индикатора загрузки шины за предыдущую секунду. Отображение величины загрузки шины за период времени нагляднее всего осуществлять с помощью графика у которого ось ординат представляет интервал времени, а ось абсцисс величину загрузки шины.

Сохранение результатов работы требуется для проведения комплексного анализа сети из нескольких точек. А также может понадобиться для более детального разбора передаваемой по сети информации.

Разработка верхнего уровня комплекса проведена на языке G в среде LabVIEW фирмы National Instrument.

Литература

1. Любашин А. Промышленные Fieldbus-сети.
<http://pcweek.ru/year1998/n48/cp1251/industrial%26built%2Din/chapt1.htm>
2. Пройдаков Э. Жданов А. Выставка Embedded systems'2000.
<http://kis.pcweek.ru/year2000/N9/CP1251/Industrial&built-in/chapt3.htm>
3. K. Etschberger. Controller Area Network (CAN) - мощное и высокоэффективное решение организации сети // перевод DATAMICRO <http://www.datamicro.ru:8101/can/tech/can-technology.shtml>

СЖАТИЕ ДАННЫХ ДЛЯ СИСТЕМЫ УДАЛЕННОГО ДОСТУПА VNC

Каплинский К. В.

Научный руководитель: Осокин А. Н., к.т.н., доцент

Томский политехнический университет

DATA COMPRESSION FOR VNC REMOTE DESKTOP SYSTEM

Constantin V. Kaplinsky

Supervisor: Alexander N. Osokin

Tomsk Polytechnic University

E-mail: const@ce.cctpu.edu.ru, WWW: <http://www.ce.cctpu.edu.ru/vnc/>

Аннотация Данная работа посвящена разработке и анализу эффективной системы сжатия информации для программного комплекса VNC, обеспечивающего удаленный сетевой доступ к графическим интерфейсам пользователя. Использование алгоритмов анализа и предобработки графической информации позволило достигнуть как увеличения степени сжатия, так и уменьшения времени кодирования по сравнению с наиболее эффективной из существующих схем сжатия VNC. Практическое применение новой схемы кодирования существенно повышает эффективность использования данной системы в сетях передачи данных с невысокой пропускной способностью (до 128 Kbps).

Abstract This report reflects development and analysis of an efficient data compression technique for VNC, a software suite intended for remote network access to graphical user interfaces. Special algorithms to analyse and pre-process image data used in front of actual compression stage have made it possible to achieve better overall compression ratios as well as higher coding speed, as compared to the most efficient existing VNC encoding scheme. Practical use of new compression technique shows notable performance improvement when modified VNC software is run in low-bandwidth (up to 128 Kbps) network environments.

VNC (Virtual Network Computing) [1] представляет собой комплекс программного обеспечения (ПО), реализующий полнофункциональный удаленный доступ к графическому экрану компьютера. Программный комплекс выполнен в технологии клиент-сервер и использует для обмена данными протокол RFB (Remote Frame Buffer). В качестве транспортного сетевого протокола используется стандартный протокол TCP. Клиентское ПО работает на компьютере, за которым находится пользователь, в то время как серверная часть выполняется на машине, к которой осуществляется удаленный доступ. В результате графический экран удаленного компьютера отображается на локальный монитор пользователя, при этом позволяя пользователю управлять удаленным компьютером с помощью локальной клавиатуры и манипулятора типа “мышь”. Важной особенностью программного комплекса VNC является открытость исходных текстов и используемых протоколов. VNC является свободным ПО и доступен в исходных текстах для множества популярных платформ, включая Unix, Win32 и MacOS.

Одним из существенных недостатков VNC и протокола RFB является неэкономичность представления данных, составляющих обновления экрана. В данной работе представлены результаты разработки автором эффективной системы кодирования для VNC — “VNC Tight Encoding”.

Протокол RFB основан на представлении экрана прямоугольной матрицей пикселей. Сервер хранит оригинальное содержимое экрана, отслеживает его изменения и передает эти изменения клиентам по их требованию. Обновления, передаваемые клиенту, содержат информацию о множестве прямоугольных областей экрана, при этом каждая отдельная область представлена матрицей пикселей, закодированной с помощью одной из нескольких схем кодирования.

Стандартный программный комплекс VNC поддерживает пять схем кодирования, которым в документации сопоставлены следующие обозначения: Raw (кодирование без сжатия), CopyRect (специальная схема кодирования для копирования областей экрана), RRE, CoRRE и Hextile. Другие реализации VNC могут включать реализации прочих схем кодирования. В частности, компания Tridia, Corp. реализовала в своем дистрибутиве TridiaVNC две дополнительные схемы, обеспечивающие лучшие степени сжатия по сравнению со стандартными: Zlib и ZlibHex. Обе эти схемы используют для сжатия распространенную библиотеку zlib, причем схема ZlibHex комбинирует ее использование с применением схемы кодирования Hextile.

В сетях с высокой пропускной способностью, как правило, используется Hextile-кодирование или кодирование без сжатия. Однако для медленных каналов со скоростями 128 Kbps и менее вопрос выбора оказывается достаточно сложным. Схема кодирования Zlib обеспечивает наилучшие степени сжатия по сравнению с другими существующими схемами, однако на практике даже такого сжатия оказывается недостаточно и, кроме того, Zlib-кодер работает достаточно медленно, потребляя значительные вычислительные мощности процессора. Таким образом, существует потребность в новых схемах кодирования, которые позволяли бы как увеличить степень сжатия, так и снизить нагрузку на процессор.

Для обеспечения существенного преимущества перед существующими схемами кодирования, реализованными в стандартной версии VNC, к новой схеме сжатия предъявлялись следующие требования.

1. Коэффициенты сжатия для типичных наборов данных должны превышать коэффициенты, которые достигаются использованием схемы кодирования Zlib. В наихудших случаях необходимо обеспечить качество сжатия, сравнимое с Zlib, в то же время обеспечив лучшие результаты при сжатии растровых данных, составляющих наиболее типичные графические пользовательские интерфейсы.
2. Производительность новой схемы сжатия в типичных случаях должна превышать скорость Zlib-кодера. Алгоритмы, существенно замедляющие процесс кодирования, допустимо использовать лишь в тех случаях, когда их применение позволяет достигнуть значительного увеличения степени сжатия.
3. Уровень сжатия должен быть регулируемым. При работе на высоких уровнях сжатия допустимо использовать медленные алгоритмы, потребляющие заметные вычислительные мощности, в то время как низкие уровни сжатия должны гарантировать высокую скорость обработки данных. Такой подход позволит использовать одну базовую схему кодирования на широком спектре вычислительного и коммуникационного оборудования, где как вычислительная мощность систем, так и пропускная способность линий передачи данных могут существенно варьироваться.
4. Для различных типов растровых данных должны использоваться различные алгоритмы предобработки. Содержимое растрового экрана, составляющее интерфейс пользователя, редко бывает однородным и состоит, как правило, из областей с разным статистическим характером растровых данных. Использование набора различных алгоритмов предобработки (фильтров) для данных с разными статистическими свойствами позволит увеличить адаптивность схемы кодирования и достигнуть лучших усредненных степеней сжатия.
5. Статистически разнородные данные должны формировать отдельные потоки информации на входе финального кодера. Данный подход позволяет исключить из массивов контекстуальной информации те наборы данных, которые с высокой вероятностью не будут содержать совпадений с фрагментами кодируемых сообщений [2, 3].
6. На финальном этапе сжатия должна использоваться стандартная библиотека zlib. Это позволит максимально упростить разработку и поддержку клиентского ПО для самых различных типов оборудования, от настольных персональных компьютеров до мобильных и встраиваемых систем.

7. Контекстуальная информация (словарь LZ-компрессора) должна сохраняться между отдельными обновлениями экрана. Как сервер, так и клиент должны использовать словарь, заполненный при сжатии предыдущих обновлений экрана, для кодирования последующих обновлений. Таким образом, должны обеспечиваться функции кэширования графической информации.

В соответствии с перечисленными требованиями была разработана эффективная схема кодирования с кодовым обозначением Tight. Пусть на вход системы кодирования поступает линейный массив данных, представляющий прямоугольную матрицу значений пикселей. Задача кодера — получить на выходе массив сжатых данных, который будет инкапсулирован в сообщение об обновлении экрана протокола RFB. Предлагаемая структура системы кодирования построена из следующих этапов обработки информации.

1. Анализатор растровых данных. Задача анализатора состоит в определении статистических свойств участка растровых данных с целью выбора подходящего алгоритма фильтрации (предобработки). В качестве критериев для выбора подходящего фильтра используются количество уникальных цветов в исследуемом фрагменте изображения, а также среднее квадратическое отклонение интенсивности цветовых составляющих соседних пикселей. Кроме того, на работу анализатора влияет набор входных параметров, зависящих от установленного уровня сжатия.
2. Процедуры фильтрации данных. Набор процедур-фильтров обеспечивает предобработку данных для рационального использования возможностей zlib-кодера. Выбор конкретной процедуры-фильтра определяется на предшествующем этапе при анализе свойств изображения. На этапе фильтрации генерируется набор данных, готовый для финальной обработки стандартными методами сжатия. Размер набора данных на выходе фильтра может отличаться от размера входного массива данных. В текущей реализации могут использоваться три алгоритма фильтрации, два из которых преобразуют цветовой формат растровых данных, а третий осуществляет один из видов разностного кодирования, позволяя увеличить степень сжатия для полноцветных фотореалистичных изображений.
3. Многопоточный zlib-кодер. Обеспечивает заключительный этап кодирования, на котором потоки данных на выходе фильтров подвергаются сжатию с помощью стандартной библиотеки zlib. В текущей реализации используется не более четырех независимых zlib-потоков. При этом номер потока определяется типом используемого фильтра. Порции информации малого объема (до 12 байтов) не подвергаются сжатию и выдаются на выход кодера без изменений. На параметры кодера влияет набор входных параметров, зависящих от установленного уровня сжатия.

Для сравнения новой схемы кодирования с существующими алгоритмами было спланировано и осуществлено сравнительное тестирование схем кодирования. Для его проведения были записаны и сохранены в файлах восемь тестовых сеансов, отражающих работу с разнообразными пользовательскими интерфейсами и изображениями различной сложности, с использованием различной глубины цвета (8, 16 и 24 бита на один пиксел). Каждому тестовому сеансу назначено кодовое имя, отражающее его содержание и цветовую глубину растровых данных: bugzilla-16, compilation-16, bars-16, kde-hearts-16, freshmeat-8, slashdot-24, photos-24, kde-hearts-24.

В качестве базовых критериев сравнения схем кодирования были выбраны два показателя: суммарный объем сжатых данных в байтах и суммарное время сжатия. Каждый из показателей определялся отдельно для каждого тестового сеанса и каждой схемы кодирования. Для измерения была разработана специализированная утилита сравнения, в которую были включены фрагменты исходного кода VNC, отвечающие за процесс кодирования растровых данных.

Тестирование производилось на компьютере с процессором Intel Pentium II 350 МГц, под управлением ОС KSI Linux 2.0 (ядро kernel-2.2.10_ac9.KSI2-1, системная библиотека glibc-2.0.7.KSI2-12). Утилита тестирования была собрана компилятором egcs-1.0.3a.KSI2-2 с использованием ключа оптимизации “-O2”.

В табл. 1 сведены результаты сравнительного тестирования для схем кодирования Hextile, ZlibHex, Zlib и разработанной схемы кодирования Tight. Данные для схемы Tight приведены для максимального предусмотренного уровня сжатия. Результаты тестирования схем RRE и CoRRE не приводятся, так как последние во всех случаях показывают худшие по сравнению с Hextile степени сжатия и при этом не дают существенного выигрыша во времени кодирования.

Таблица 1. Результаты сравнительного тестирования

Тестовый сеанс	Размер сжатых данных, байт (время сжатия, с)				
	без сжатия	Hextile	ZlibHex	Zlib	Tight
bugzilla-16	32729254	1590721 (1.3)	1096171 (5.0)	652572 (25.8)	499140 (14.9)
compilation-16	111178662	2788259 (5.2)	2643398 (5.6)	1836495 (231.5)	454748 (21.3)
bars-16	32880890	11812296 (2.6)	7814321 (76.4)	3632612 (74.7)	3191666 (52.3)
kde-hearts-16	22901336	13711535 (2.4)	4830374 (29.5)	2011507 (26.1)	1788952 (17.5)
freshmeat-8	107621635	4779238 (7.4)	3904280 (11.9)	2635469 (147.5)	2265923 (86.2)
slashdot-24	271717540	8716348 (5.7)	5748492 (44.7)	4487303 (341.8)	3297038 (134.2)
photos-24	25569676	16781057 (1.6)	10763572 (26.1)	10657746 (24.1)	6130479 (35.3)
kde-hearts-24	36152860	20052400 (2.4)	7908693 (60.6)	2066224 (46.0)	1790153 (20.6)

Как видно из табл. 1, предложенная схема сжатия Tight обеспечивает лучшие коэффициенты сжатия на всех тестовых сеансах. Кроме того, сжатие потребляет существенно меньше процессорного времени по сравнению с Zlib во всех случаях, кроме тестового сеанса photos-24. Увеличение времени сжатия для этого сеанса связано с использованием специализированного фильтра, который замедляет процесс кодирования, но позволяет значительно увеличить степень сжатия для фотореалистичных изображений.

Практические испытания новой схемы кодирования показывают справедливость выводов, полученных в результате сравнительного тестирования. В частности, при доступе к VNC-серверу через модемное PPP-соединение на скорости 56 Kbps (протокол V.90) использование схемы сжатия Tight заметно уменьшает визуальные задержки, вызванные передачей больших массивов данных, и значительно улучшает общую производительность удаленной работы.

В настоящее время разработанная схема кодирования внедрена в популярный дистрибутив TridiaVNC и широко используется его пользователями. Дополнительная информация, тестовые сеансы и полный исходный код для VNC Tight Encoding могут быть получены по следующему адресу в World-Wide Web: <http://www.ce.cctpu.edu.ru/vnc/>.

Литература

1. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood & Andy Hopper, "Virtual Network Computing", IEEE Internet Computing, Vol. 2 No. 1, Jan/Feb 1998, pp. 33–38.
2. К. В. Каплинский, А. Н. Осокин. Оптимизация алгоритмов сжатия информации для обработки электронной почты. // Новые информационные технологии в университетском образовании: Сборник трудов / Новосибирск: Изд. НИИ МИОО НГУ, 1998.
3. Russian-Korean International Symposium on Developing Data Compression Techniques for Structured Text Documents in Wide-Area Networks // Abstracts: The Second Science and Technology "KORUS-98", Tomsk, 1998.

ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА “МИКРОКОНТРОЛЛЕР” С УДАЛЕННЫМ ДОСТУПОМ ПО СЕТИ INTERNET

Ковынев А.Н.

Научный руководитель к.т.н. доц. каф ВТ Салит В.В.
Томский политехнический университет, Томск, Россия.

INFORMATION SEARCH SYSTEM "MICROCONTROLLER" WITH THE REMOVED ACCESS ON THE NETWORK INTERNET

Kovynev A.N.

Assoc. prof. Salit V.V.

Tomsk polytechnical university, Tomsk, Russia
alleksey@mail.ru, <http://www2.ce.cctpu.edu.ru/mcs/>

Аннотация. В настоящее время большое внимание уделяется автоматизации проектирования микропроцессорных систем. Одной из сложных задач для разработчика является выбор оптимального микроконтроллера с учетом системного подхода. Для решения задачи оптимального выбора хорошо зарекомендовали себя специализированные экспертные системы. В докладе рассматривается информационно-поисковая система “Микроконтроллер”, в которой совмещены параметрический поиск по базе данных с алгоритмом нечеткой логики. Система реализована с использованием технологии клиент-сервер и рассчитана на пользователей с различной квалификацией с возможностью доступа по сети Internet.

Abstract. Now large attention is given to automation of designing of microprocessor systems. One of difficult tasks for the developer is the choice of the optimum microcontroller of the system approach. For the decision of a task of an optimum choice well have recommended itself the specialized expert systems. In the report is considered the information search system "Microcontroller", in which are combined parametrical search on a database with algorithm of fuzzy logic. The system is realized with use of technology a client - server and is designed for the users with various qualification with an opportunity of access on a network Internet.

В процессе проектирования аппаратных средств встраиваемых систем одной из наиболее важных задач является выбор микроконтроллера (МК). Эта задача усложняется не только необходимостью анализа большого количества технических параметров микроконтроллера, (порядка несколько десятков, таких как: разрядность шины данных и адреса, производительность, объем оперативной памяти, номинальное напряжение питания и т.д.), но и необходимостью одновременного учета слабо формализуемых критериев, маркетинговых задач (таких как: отношение цена/качество, доступность на российском рынке, техническая и программная поддержка). Оптимальное решение проблемы выбора следует искать с учетом меняющихся требований к экономическим, техническим и программным характеристикам проектируемой системы. Кроме этого, с появлением новых технологий ситуация на рынке микроконтроллеров, в последнее время меняется достаточно динамично и в связи с этим появляется острая необходимость в быстром доступе к этой информации. Около десяти крупных компаний, выпускают сотни типов микроконтроллеров. В свою очередь множество представительств этих фирм в России предлагают потребителям ограниченный набор услуг по хранению на складе, цене и способам доставки микроконтроллеров.

Очень важно учитывать мнение экспертов в процессе выбора микроконтроллеров. Однако даже опытному разработчику встроенных систем трудно удержать в голове все множество критериев выбора микроконтроллеров. Поэтому разработчикам с различным уровнем квалификации, не говоря уже о студенте, полезно всегда иметь под рукой

систему экспертного поиска для ускорения и эффективного решения задачи выбора микроконтроллера.

Особую актуальность, данная задача приобрела с появлением глобальной сети Internet. С развитием Internet открылся огромный источник информации, а вместе с тем и глобальная проблема поиска необходимой информации. Причем чем специфичней информацию необходимо найти, тем более узкой должна быть область специализации информационно-поисковой системы (ИПС). Так как универсальные поисковые системы не позволяют сосредоточиться на одной решаемой задаче, в виду их широких возможностей.

На сегодняшний день, среди подобных специализированных поисковых систем в Internet, наибольшее распространение получили системы с параметрическим поиском [1,2,3]. Наряду с положительными качествами параметрический поиск обладает рядом существенных недостатков. Например, одна часть параметров может быть менее значима, чем другая, но исключать из запроса их не желательно, так как это серьезно повлияет на результат поиска. Возникают ситуации, в которых система не может найти информацию в базе данных (БД) строго соответствующую запросу. $M \cap F = \emptyset$, где M - множество микроконтроллеров соответствующих запросу пользователя, F - множество всех микроконтроллеров в БД.

Перспектива развития таких ИПС – это системы поиска с искусственным интеллектом или же экспертные системы (ЭС). В настоящее время уже есть аналогичные работающие системы на web сайтах Японии, США и России [4,5]. По мнению специалистов, в недалекой перспективе ЭС будут играть ведущую роль во всех фазах проектирования, разработки, производства, распределения, продажи, поддержки и в оказании каких-либо услуг. Специфика базы знаний (БЗ) таких систем, несомненно, будет накладывать некоторые ограничения на универсальность экспертных поисковых систем, но в тоже время такая организация позволяет максимально подробно приблизиться к решению этой задачи. Такая технология, получив коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуально взаимодействующих модулей.

Информационно-поисковая система “Микроконтроллер” с удаленным доступом по сети Internet предназначена для оптимального поиска микроконтроллеров для пользователей с различной квалификацией. ИПС “Микроконтроллер” представляет собой “клиент–серверную” структуру. В качестве сервера используется Web сервер Apache на платформе Linux, а в качестве клиента пользователь может использовать стандартные Web браузеры. Такая структура позволяет создать ИПС с удаленным доступом по сети Internet и снимает проблемы совместимости различных платформ между клиентом и сервером. Вся нагрузку по вычислению и поиску оптимального микроконтроллера берут на себя инструментальные средства сервера, что существенно облегчает клиентскую часть пользователей. Вся информация пользователи получают от сервера, в виде динамического HTML - документа.

Информационно-поисковая система для обеспечения более эффективного выбора микроконтроллеров, пользователям с различной квалификацией, предоставляет несколько вариантов поиска, таких как:

- ПОИСК ПО НАЗВАНИЮ МИКРОКОНТРОЛЛЕРА;
- ПАРАМЕТРИЧЕСКИЙ ПОИСК;
- ЭКСПЕРТНЫЙ ПОИСК.

Пусть M - множество типов микроконтроллеров в БД, тогда каждому типу $m_k^i \in M$ соответствует кортеж из множества параметров $p_{kj} \in P_k$, где $k=1,2,\dots,K$ - количество элементов множества M , $i=1,2,\dots,I$ - количество фирм производителей микроконтроллеров $j=1,2,\dots,J$ - количество параметров. Множество микроконтроллеров M состоит из

подмножеств $M_1 \cup M_2 \cup \dots \cup M_i \cup \dots \cup M_I = M$ при условии, что $(M_\alpha \cap M_\beta = \emptyset)$, для всех $\beta \neq \alpha$, где $(\beta, \alpha = 1, 2, \dots, I)$.

1. Поиск по названию микроконтроллера.

Под названием микроконтроллера будем понимать буквенно-цифровой идентификатор, который является одним из параметров $p_{k,1}$, микроконтроллера $m_k^i \in M$

В этом режиме существует два подрежима:

- Выбор из списка. Выбор названия микроконтроллера происходит в два этапа, в начале выбирается $P_{k,1}^c$ подмножества $M^{i=c}$, где $c = \text{const.}$, что соответствует выбору полного списка названий только одной фирмы-производителя микроконтроллеров, а затем выбирается интересующий пользователя микроконтроллер $n_{k=v}^{i=c} \in M$, где $v = \text{const.}$

- Поиск по ключевому слову. Выбор МК осуществляется по шаблону S заданному пользователем. Результатом поиска является множество микроконтроллеров n_k^i для которых выполняется условие $(p_{k,1} \cap S = 1)$. Это означает, что в качестве шаблона может быть введена строка символов содержащая не полное название МК. Данный режим позволяет выбирать семейство МК, которые изготавливаются различными фирмами-производителями. Например, для выбора МК из семейства MCS51 достаточно ввести "C51" или "80C5".

2. Параметрический поиск.

Параметрический поиск предполагает поиск микроконтроллеров n_k^i по параметрам, выбранным пользователем из предлагаемого списка $h_q \in P_q$, где $q \in J$. Список содержит как конкретные значения, так и интервальные. Результатом поиска является подмножество микроконтроллеров m_k^i удовлетворяющих условию $p_{k,j} \cap h_q, \forall j = q$.

3. Экспертный поиск.

Наиболее значимый интерес представляет конечно же режим экспертного поиска. В этом режиме система предоставляет максимальные возможности экспертной системы. Именно здесь система, выдает пользователю дерево вопросов, которое позволяет ИПС избавиться от не определяющих характеристик для поиска. В режиме экспертного поиска, так же как и в режиме параметрического поиска, пользователь ИПС определяет для себя множество параметров из списка $h_q \in P_k$, где $q \in J$, и кроме этого вводит множество нечетких элементов $Q \in P_k$, где $k \in J$. При каждом шаге по дереву вопросов с помощью поисковой системы определяется некоторое множество $n_k^i \in M$ подходящих микроконтроллеров, которые временно сохраняются для каждого запроса в экспертном поиске. После того, как ИПС пройдет по всему дереву поиска, будет окончательно сформировано множество M . Затем в зависимости от значений в множестве Q для каждого $n_k^i \in M$ вычисляется степень соответствия микроконтроллера запросам пользователя. Результирующий список предлагаемых микроконтроллеров ранжируется в соответствии с этим показателем и содержит полный перечень параметров с указанием степени соответствия запросу.

РЕАЛИЗОВАНА ИПС В ВИДЕ СЕРВЕРНОГО ПРИЛОЖЕНИЯ НЕСКОЛЬКИМИ МОДУЛЯМИ CGI (COMMON GATEWAY INTERFACE) ЗАПРОГРАММИРОВАННЫЕ НА ЯЗЫКЕ PERL, ТАКИЕ ПРИЛОЖЕНИЯ ЯВЛЯЕТСЯ СТАНДАРТОМ ИНТЕРФЕЙСА СВЯЗИ ВНЕШНЕЙ ПРИКЛАДНОЙ ПРОГРАММЫ С ИНФОРМАЦИОННЫМ WEB СЕРВЕРОМ. CGI-ПРОГРАММЫ НАЗЫВАЕМЫЕ ТАКЖЕ ШЛЮЗАМИ ЗАПУСКАЮТСЯ WEB СЕРВЕРОМ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ И ВЗАИМОДЕЙСТВУЮТ С СУБД MYSQL, В КОТОРОЙ НАХОДЯТСЯ БД И БЗ МИКРОКОНТРОЛЛЕРОВ ВСЕЙ ИПС.

КРОМЕ ЭТОГО ДЛЯ СОПРОВОЖДЕНИЯ ИПС СОЗДАНЫ УТИЛИТЫ ПО АДМИНИСТРИРОВАНИЮ БД И БЗ. УТИЛИТЫ ПО АДМИНИСТРИРОВАНИЮ ПОЗВОЛЯЮТ АДМИНИСТРАТОРАМ ПРОСМАТРИВАТЬ, ДОБАВЛЯТЬ, УДАЛЯТЬ И МОДИФИЦИРОВАТЬ ДАННЫЕ ВСЕЙ ИПС ЧЕРЕЗ INTERNET. ДОСТУП К ЭТОЙ

ИНФОРМАЦИИ С ПРАВАМИ АДМИНИСТРАТОРА КОНТРОЛИРУЕТСЯ СРЕДСТВАМИ СЕРВЕРА. В НАСТОЯЩЕЕ ВРЕМЯ ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА “МИКРОКОНТРОЛЛЕР” ПРОХОДИТ ТЕСТИРОВАНИЕ НА СЕРВЕРЕ КАФЕДРЫ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ ТОМСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА [6].

ЛИТЕРАТУРА

1. Atmel Microcontroller Search <http://www.atmel.com/atmel/products/select20.htm>
2. Intel product selector http://apps.intel.com/product_selector/nojava/tablem.asp
3. Philips. 80C51 Microcontroller Derivatives Searchable Database
<http://www-eu2.semiconductors.com/mcu/products/80c51/>
4. Яндекс Гуру. Система выбора товаров на основе искусственного интеллекта.
<http://www.guru.yandex.ru/>
5. CORVID Camcorder Selection Expert System
http://www.exsys.com/Demos/Camcorder/Camcorder_demo.html
6. Информационно-поисковая система “Микроконтроллер”
<http://www2.ce.cctpu.edu.ru/mcs/>

ФОРМИРОВАНИЕ БИБЛИОТЕК КОМПОНЕНТОВ В ИНТЕРАКТИВНОЙ САПР

Корконишко А.Н., Плиско Д.И.
Научный руководитель доцент Антропов А.Н.
ОмГТУ, Омск, Россия

CREATION OF COMPONENT LIBRARIES IN INTERACTIVE CAD

Korkonishko A.N., Plisko D.I.
The scientific leader docent Antropov A.N.
OmSTU, Omsk, Russia

адрес электронной почты: email@omgtu.omскеlecom.ru

Аннотация Рассматривается технология формирования библиотечных описаний компонентов РЭА на основе процедуры ассоциативного поиска в общем информационном поле и автоматического включения во вновь формируемое описание типовых элементов ранее сформированных библиотечных описаний. Рассматриваются функции подсистемы управления библиотеками компонентов прототипной интерактивной САПР.

Abstract The technology of creation of the library descriptions of components of the electronic equipment is considered on the basis of procedure of associative search in a general information field and automatic inclusion in the again formed description of typical elements before the generated library descriptions. The functions of a subsystem of management of libraries of components interactive CAD are considered.

Разработка технологии и инструментальных средств для формирования библиотечных описаний компонентов РЭА является самостоятельной и важной задачей в области автоматизации проектирования конструктивных модулей РЭА.

В связи с проблемой представления компонентов в структуре информационного обеспечения прототипной интерактивной САПР печатных плат поставлены следующие задачи:

- исследование и разработка концепции программы-менеджера библиотек;
- анализ содержания и разработка структуры интегрированной базы данных менеджера;
- исследование и разработка диалоговой процедуры формирования библиотечных образов компонентов;
- исследование и разработка процедур импорта и экспорта элементов библиотечных описаний в интегрированную базу данных;

В связи с необходимостью обеспечения межсистемного взаимодействия решается также задача конвертирования данных в форматы смежных систем. В частности, результатом конвертирования являются PRT- и SYM-файлы библиотечных описаний компонентов системы PCAD.

Для формирования и хранения образов компонентов разрабатываемое программное обеспечение должно содержать программу обработки графической и атрибутивной информации из состава библиотечных описаний компонентов (менеджер библиотек) и базу данных для хранения указанной информации. Для организации доступа к данным предлагается применение архитектуры «клиент-сервер», предусматривающей использование двух типов программ: а) программы, работающей совместно с приложением, использующим формируемые библиотечные описания (программа-клиент); б) программы, реализующей доступ к данным общего информационного поля менеджера библиотек (программа-сервер).

Менеджер библиотек САПР печатных плат (рисунок 1) позволяет проектировщику формировать библиотечные описания условных графических и конструкторско-технологических образов компонентов РЭА в интерактивном режиме. В процессе формирования описания компонента программа предлагает проектировщику импортировать

уже имеющиеся (вторичные) элементы библиотечных описаний из сопрягаемой с менеджером библиотек интегрированной базы данных. Программа также экспортирует вновь создаваемые (первичные) элементы библиотечного описания в общее информационное поле менеджера библиотек.

На основе импортированной информации обеспечивается формирование повторяющихся элементов библиотечных описаний в автоматическом режиме. Последующее доопределение, формируемого библиотечного описания до полного осуществляется в интерактивном режиме.

В процессе создания библиотечных описаний программа автоматически отслеживает ситуации, связанные со вторичным вводом повторяющихся элементов библиотечных описаний.

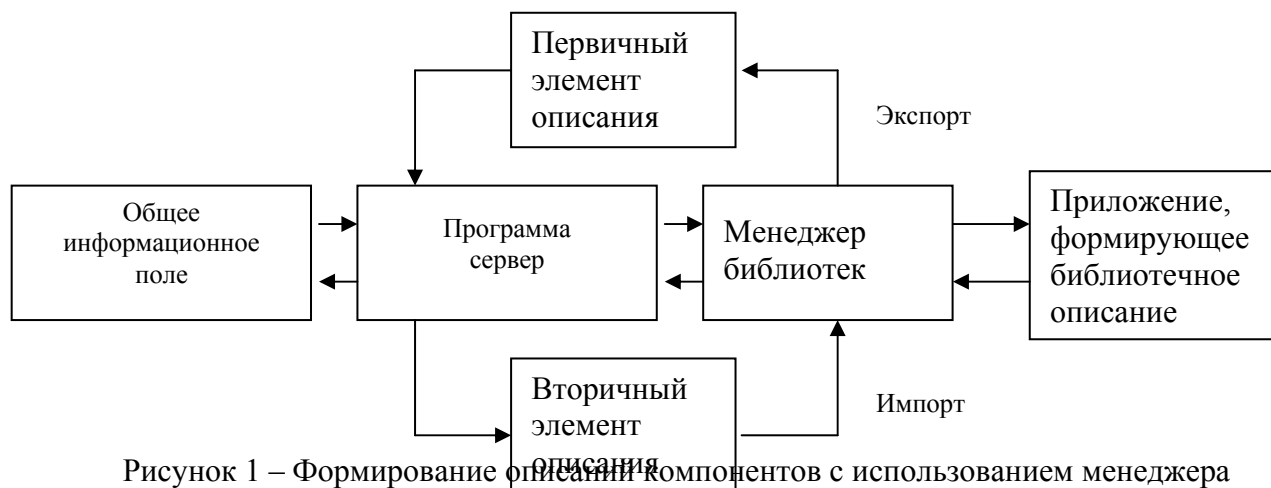


Рисунок 1 – Формирование описания компонентов с использованием менеджера

Доступ к базе данных, представляющей общее информационное поле менеджера библиотек, реализует программа-сервер, выполняющая следующие функции:

- мониторинг информационного поля менеджера библиотек по запросу;
- хранение, модификация и удаление типовых элементов описаний компонентов в базе данных по запросу;
- импорт элементов библиотечных описаний по запросу;
- автоматический поиск в общем информационном поле и экспорт элементов библиотечных описаний по запросу;
- проверка на достоверность и минимизация избыточности данных до включения в базу.

Общий алгоритм функционирования менеджера библиотек сводится к следующему.

После ввода идентификационного параметра типа компонента менеджер строит обращение к серверу для ассоциативного поиска в общем информационном поле и импорта требуемых элементов библиотечного описания. Обнаруженные элементы включаются в описание и предлагаются для редактирования. В случае, если элемент не обнаружен, выполняется процедура его формирования, после чего менеджер строит обращение к серверу для экспорта этого элемента описания в общее информационное поле.

Таким образом эффективность процедуры формирования библиотечных образов компонентов повышается за счет использования фактора учета предыстории ее применения.

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ ОПЫТНЫХ ИЗДЕЛИЙ С ПРИМЕНЕНИЕМ ПОВЕРХНОСТНОГО МОНТАЖА.

Ковалевский Ю. С.

Научный руководитель: д. т. н., проф. каф. ИУ4 Белов Б. И.

Московский Государственный Технический Университет им. Н. Э. Баумана,
г. Москва, Россия

FEATURES OF PCB DESIGN FOR EXPERIMENTAL DEVICES WITH SURFACE MOUNTING.

Kovalevsky Y. S.

Supervisor: doctor, professor of department ICS4 Belov B. I.

Moscow State Technological University named after N. E. Bauman,
Moscow, Russia.

E-mail: kys@mail.ru

Аннотация. Описываются особенности компоновки и трассировки печатных плат с использованием поверхностного монтажа при опытном и макетном производстве, упрощающие ручные операции по сборки узлов и замене элементов и улучшающие качество монтажа.

Abstract. There are described model and experimental surface mounting PC-boards arranging and routing features that simplify manual operations and improve mounting quality.

Введение.

Новым словом в развитии элементной базы изделий электронной промышленности явились компоненты, монтируемые на поверхность печатных плат. Технология монтажа на поверхность (ТМП) уже получила широкое распространение, несмотря на относительно недавнее появление. Хотя ТМП рассчитана в основном на серийное и массовое производство, часто встает вопрос о выпуске единичных экземпляров изделий при макетировании и опытном производстве. Поскольку многие компоненты для поверхностного монтажа не имеют прямых аналогов в корпусах, монтируемых в отверстия, полностью отказаться от ТМП на данном этапе не удастся. Однако технология монтажа на поверхность качественно различна при единичном и серийном производствах, что необходимо учитывать при проектировании макетных и опытных узлов. Если при этом изменение компоновки печатного узла и рисунка ППП повлияет на характеристики устройства незначительно, целесообразно разработать печатную плату с учетом особенностей поверхностного монтажа, выполняемого ручным или автоматизированным способом, а также обеспечивая простоту демонтажа и замены отдельных компонентов.

Компоновка и размещение компонентов.

При компоновке макетного печатного узла необходимо учитывать следующее:

1. Элементы должны располагаться на достаточном расстоянии друг от друга для упрощения ручного демонтажа. Минимальный зазор между элементами равен $w + \delta$, где w – толщина губки приспособления (при удалении элемента вакуумным приспособлением эта величина равна нулю), δ – зазор, предотвращающий случайный сдвиг элемента. Рекомендуемая величина δ составляет 2 мм.

2. При пайке ручной конвективной паяльной станцией в режиме фена сложно обеспечить равномерность потока горячего газа, что может привести к сдвигу элементов с контактных площадок. Силы поверхностного натяжения центруют элемент, если его сдвиг не привел к попаданию вывода на другой открытый элемент печатного рисунка. Поэтому контактные площадки должны располагаться на достаточном расстоянии друг от друга и от переходных отверстий.

3. Высокие компоненты приводят к появлению теневых участков при конвективной пайке. При этом различные области ППП нагреваются неравномерно, а следовательно, неравномерно изменяются во времени вязкие свойства припойной пасты в теневой зоне и вне ее. Если часть контактных площадок компонента оказываются в теневой зоне, возможно стягивание его в сторону площадки, на которой частицы припоя в припойной пасте уже расплавились, поскольку поверхностное натяжение на данной площадке выше. При значительном стягивании компонентов в корпусах 0805, 1206, SOT80 и им подобных может возникнуть эффект «надгробного камня». У корпусов типа SOIC, SOT23, SOT223 и подобных это может возникнуть некачественное паяное соединение или отсутствие соединения в теневой зоне.

Компоненты в высоких корпусах следует располагать на достаточном расстоянии друг от друга и от других компонентов. Если плотность компоновки некритична, расстояние между компонентами следует выдерживать не менее $1,5h$, где h – наибольшая высота соседних компонентов. Надежная пайка наблюдается при уменьшении расстояния до h .

В тех случаях, когда на расстояние между компонентами имеется ограничение сверху (например, если длина проводника оказывает существенное влияние на функционирование схемы), возможно уменьшение влияния теневых участков одним из следующих способов:

- Увеличение размера контактных площадок, находящихся вне теневой зоны. Это приводит к увеличению теплоемкости площадок и, как следствие, выравниванию скорости плавления припоя. Недостаток этого подхода – сложность расчета размеров площадок. Кроме того, компенсация недостатка нагрева за счет теплоемкости может привести к недопустимому увеличению площади.

- Изменение конфигурации контактных площадок, находящихся в теневой зоне. При этом создается площадка сложной формы, либо две площадки, соединенные коротким широким проводником (на практике длина проводника не более 1-2 мм, ширина не менее 1,5 мм). Область площадки, с которой непосредственно контактирует элемент, располагается в соответствии с конструктивными ограничениями (в том числе приведенными к теневой зоне), а остальная ее часть вне теневой зоны, но так, чтобы между областями обеспечивался хороший теплообмен. При этом часть площадки нагревается более интенсивно и за счет кондукции нагревает другую ее часть. Повышение теплоемкости площадки либо незначительно, либо компенсируется увеличением площадок, расположенных вне теневой зоны.

- Изменение направления потока горячего газа. Поскольку при ручной пайке не составляет труда изменить направление потока газа, теневые зоны можно уменьшить или полностью исключить. Однако нужно иметь в виду, что поток не должен сдвигать компоненты с контактных площадок, поэтому компоненты в корпусах 0805 и подобных, а в особенности цилиндрических корпусах типа SOD80, следует располагать вдоль предполагаемого направления потока газа.

При увеличении контактных площадок вне теневой зоны с целью увеличения их теплоемкости, дополнительные площади желательно располагать под корпусом компонента (в особенности для корпусов типа SOIC и подобных), поскольку в этом случае нагрев площадки увеличивается незначительно (см. рис. 1). Окно в паяльной маске (фоторезисте) следует выполнять как у обычной площадки для уменьшения расхода припойной пасты и ее растекания.

4. Паяные соединения элементов, монтируемых в отверстия, обладают значительно большей теплоемкостью по сравнению с поверхностно-монтируемыми компонентами за счет большего количества припоя и массивности выводов. При демонтаже вывода из отверстия и очистке отверстия от излишков припоя плате передается значительное количество теплоты, способное привести к плавлению припоя находящихся рядом соединений компонентов, монтируемых на поверхность. Поэтому монтажные отверстия следует располагать на достаточном расстоянии от контактных площадок для поверхностного монтажа. Это расстояние зависит от размеров выводов и отверстий и типа оборудования, применяемого для демонтажа. Обычно это расстояние составляет 2,5-3 мм. Проводник, соединяющий площадку с монтажным отверстием должен быть не шире 0,6 мм.

Принцип компоновки узла в целом выбирается из анализа сложности трассировки и производства ПП и влияния расположения элементов на электрические характеристики устройства. Если влияние этих факторов незначительно, желательно как можно большее число компонентов, монтируемых на поверхность, заменить на их прямые аналоги, монтируемые в отверстия, и сгруппировать компоненты по виду монтажа, разместив их в разных частях платы. В противном случае производятся выборочные изменения расположения компонентов на отдельных участках платы.

Топология печатной платы.

Основной особенностью топологии ПП для ТМП является наличие компенсаторов теплоемкости и ловушек припоя.

Компенсаторы вводятся для выравнивания теплоемкости контактных площадок. Если одна или несколько контактных площадок компонента подсоединены к широкому проводнику, либо к области металлизации, вблизи остальных площадок выполняются также зоны металлизации небольшого размера (см. рис. 2).

Получаемый рисунок должен быть приближен к симметричному. Тепловые барьеры должны иметь одинаковую конфигурацию.

Чтобы размер зоны металлизации не оказывал значительного влияние на симметричность температур

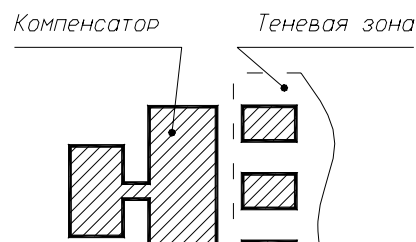
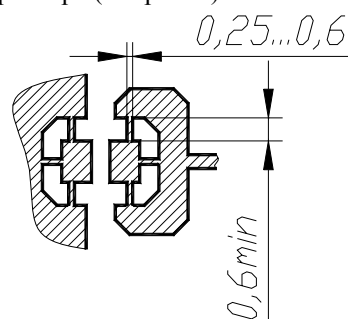


Рисунок 1 Изменение формы КП для корпуса SOT223.



контактных площадок, тепловые барьеры должны выполняться проводниками шириной 0,25-0,6 мм и длиной не менее 0,6 мм.

Ловушки припоя применяются как правило при пайке волной. Поскольку данный вид пайки не применяется в опытном производстве, ловушки можно не выполнять. За счет освободившейся площади можно увеличить плотность компоновки, либо выполнить дополнительные компенсаторы с учетом особенностей ручной конвективной пайки.

При выборе размеров контактных площадок необходимо также учитывать технологические нормы на топологию ПП. Так например для корпуса 0805 ширина контактной площадки варьируется в пределах $X=1,3 \dots 1,8$ мм. Однако при выполнении ПП по 3-ему классу плотности в дюймовой сетке при расстоянии между двумя элементами 2,54 мм зазор между их контактными площадками оказывается равным $2,54 - X = 1,24 \dots 0,74$ мм, а зазор, необходимый для прокладки одного проводника между этими площадками равен $2s+t=0,75$ мм, где $s = 0,25$ мм – минимальный зазор для 3-его класса плотности, $t = 0,25$ мм – минимальная ширина проводника. Таким образом, максимальная ширина контактной площадки вычисляется как $2,54 - 0,75 = 1,79$ мм, и выбирается равной 1,7 мм.

Контактные площадки высоких элементов, например электролитических конденсаторов в цилиндрическом корпусе, желательно увеличивать за пределами элемента для более эффективного нагрева. В этом случае конвективный поток нагревает дополнительную область площадки, и теплота передается пасте кондукцией. Для предотвращения растекания пасты в паяльной маске делается два окна с образованием перемычки, препятствующей распределению пасты по всей поверхности площадки (см. рис. 3).

Трассировка печатной платы.

Трассировку печатной платы целесообразно выполнять в сетке с шагом, в два раза меньшим шага при размещении компонентов. Для опытных плат шаг сетки размещения можно выбрать 1,25 (1,27) мм, а трассировки 0,625 (0,635) мм. В этом случае обеспечивается достаточная сложность разводки при отсутствии необходимости перерасположения компонентов.

Проводники желательно подводить к контактными площадкам так, чтобы они образовывали симметричную картину относительно центра элемента (см. рис. 4). Проводники, подключенные к одному элементу должны быть одинаковой ширины. Если это невозможно, более широкий проводник выполняется с сужением вблизи площадки. Ширина этого сужения должна быть равна w , длина $(1 \dots 2)w$, но не менее 0,6 мм, где w – наименьшая ширина проводников вблизи соединения с данным элементом.

Проводник, соединяющий контактную площадку с переходным отверстием, также должен иметь сужение вблизи площадки.

Покрытие печатных проводников и платы.

Для предотвращения растекания припоя печатную плату при применении ТМП покрывают паяльной маской: сухим пленочным или жидким фоторезистом.

Перед выполнением маски печатные проводники покрывают сплавом Розе, гальваническим оловянно-свинцовым сплавом, припоем ПОС-61 или другим металлическим покрытием, либо оставляют без покрытия («маска по меди»).

При выборе покрытия проводников нужно учитывать, что при конвективной пайке разогреваются не только контактные площадки, но остальная поверхность платы. Если покрытие более легкоплавкое, чем припой, содержащийся в припойной пасте, его плавление может вызвать отслоение маски.

Заключение.

Приведенные особенности проектирования печатных плат для макетного и опытного производства упрощают монтаж и демонтаж элементов вручную, а также внесение изменений в схему. Однако они могут значительно уменьшить плотность компоновки узла. Изменение технологии пайки приводит к коррекции топологии ПП. Например, пайка волной не требует компенсаторов, но требует ловушек припоя. С другой стороны полная переразводка платы при запуске устройства в серийное производство может оказаться сложной. По этим причинам макетный или опытный узел должен разрабатываться с учетом как упрощения ручных операций, так и перспективы групповой технологии при серийном (массовом) производстве.

Литература.

Рисунок 2 Выполнение компенсатора и теплового барьера при подсоединении корпуса 0805 к зоне металлизации.

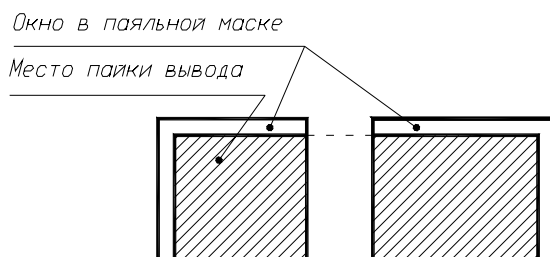


Рисунок 3 Выполнение окон в паяльной маске при увеличении КП.

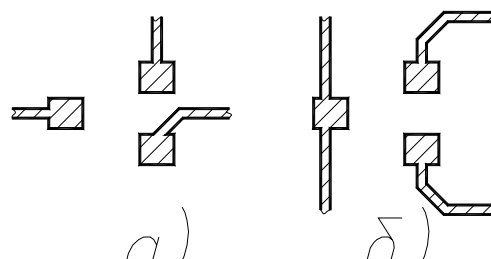


Рисунок 4 Выполнение трассировки проводников, подсоединенных к корпусу SOT23: а) неправильно; б) правильно.

1. **Большухин В. А.**, Сизов Ю. А. Технология поверхностного монтажа в производстве РЭА: Учеб. пособие. — М.: Московский Государственный Университет Леса, 1998. — 55с.: ил.
2. **Григорьев В. Н.**, Казаков А. А., Джинчарадзе А. К. и др. Монтаж на поверхность: Технология. Контроль качества./Под общ. ред. И. О. Шурчкова —М.: Изд-во стандартов, 1991. —180 [2] с.: ил.
3. **Казаков Н. И.**, Критский В. М. Технология пайки монтажных соединений в производстве РЭА: Учеб. пособие. — М.: М-во радиопромышленности СССР, Ин-т повышения квалификации руководящих работников и специалистов, 1989. — 129 с.: ил.
4. **Состояние** и развитие технологических процессов и оборудования для пайки электронных компонентов на поверхность печатных плат./Подгот. Грачевым А. А. —Киев.: О-во «Знание» УССР, 1991. — 14 с.

ПРИПОИ В ПРИПОЙНЫХ ПАСТАХ.

Ковалевский Ю. С.

Научный руководитель: д. т. н., проф. каф. ИУ4 Белов Б. И.

Московский Государственный Технический Университет им. Н. Э. Баумана,
г. Москва, Россия

SOLDERS IN SOLDER CREAMS.

Kovalevsky Y. S.

Supervisor: doctor, professor of department ICS4 Belov B. I.

Moscow State Technological University named after N. E. Bauman,
Moscow, Russia.

E-mail: kys@mail.ru

Аннотация. Приводятся свойства большинства припойных сплавов, применяемых для производства припойных паст для поверхностного монтажа. Рассматривается влияние характеристик частиц порошкообразного припоя и его содержания на свойства пасты.

Abstract. There are listed properties of most solder alloys used in surface mount solder creams manufacturing. There is examined the influence of solder fractions characteristics and its content upon cream properties.

Введение.

Припойные пасты находят широкое применение при технологии монтажа на поверхность печатных плат. Припойные пасты в своем составе имеют порошкообразный припой, флюс и органические наполнители. Содержащийся в пастах флюс обеспечивает растворение окислов, очищение контактных площадок, улучшает смачиваемость поверхности припоем. Клеящие свойства припойных паст позволяют фиксировать компоненты на печатных платах без применения клеев, чем повышается технологичность и ремонтпригодность узлов. Органические наполнители улучшают распределение порошка припоя в объеме пасты, регулируют ее тиксотропность, предотвращают разбрызгивание при лазерной пайке. Однако основным компонентом припойных паст является припой. В объеме пасты припой находится в виде взвешенных частиц, оплавляемых при пайке и образующих паяное соединение. На свойства пасты влияет как состав припойного сплава, так и размеры и форма частиц, их процентное содержание.

Состав и свойства припойных сплавов.

В припойных пастах наиболее широкое распространение получили оловянно-свинцовые припои. Среди них 63Sn/37Pb, 60Sn/40Pb, 50Sn/50Pb, 10Sn/90Pb, 5Sn/95Pb. Эти сплавы обладают низкой стоимостью, в особенности 5Sn/95Pb.

Однако сплавы с большим содержанием свинца обладают высокой температурой плавления. Такие сплавы применяются для покрытия выводов. При пайке компонентов на плату высокая температура плавления может привести к перегреву ИМС.

Сплавы 50Sn/50Pb, 10Sn/90Pb обладают значительной разницей температур ликвидус-солидус (соответственно, 33 и 16°C), что может привести к трещинам и пористости паяного соединения. Кроме того, сплавы, близкие к эвтектике, обеспечивают лучшее смачивание и большую скорость капиллярного подъема. Эвтектическим является сплав 63Sn/37Pb (температура эвтектики 183°C); сплав 60Sn/40Pb близок к эвтектике, его диапазон кристаллизации составляет 5°C.

Применение оловянно-свинцовых сплавов не рекомендуется для пайки выводов с золотым и серебряным покрытием из-за миграции материала покрытия в припой.

Для пайки выводов с серебряным покрытием применяются оловянно-свинцовые сплавы с 2% содержанием серебра. Сплав 62Sn/36Pb/2Ag широко используется как в нашей стране, так и за рубежом. Он является наиболее прочным среди оловянно-свинцовых сплавов. На его основе английской фирмой Мультикор Солдерс выпущен ряд паст, среди которых RM89 для нанесения дозатором, RM92 и WR10 для нанесения трафаретным способом. Кроме того, применяется сплав 10Sn/88Pb/2Ag, однако в отличие от эвтектического сплава 62Sn/36Pb/2Ag он обладает разницей температур ликвидус-солидус 22°C. Эти сплавы не рекомендуется для пайки золота.

Минимальное растворение серебра обеспечивается сплавами 95Sn/5Ag и 96,5Sn/3,5Ag. Они прочны, обладают хорошим смачиванием паяемых поверхностей, устойчивы к термоциклированию. Однако эти сплавы дороже, чем 62Sn/36Pb/2Ag, и имеют более высокую температуру плавления (62Sn/36Pb/2Ag имеет температуру эвтектики 179°C, а 96,5Sn/3,5Ag 221°C).

Свинцово-оловянные сплавы 75Pb/25Sn, 25Pb/75Sn обладают относительно высокой пластичностью, а растворение ими золота минимально относительно других сплавов олово-свинец. Между собой они

различаются температурой плавления. Наиболее легкоплавкий сплав 25Pb/75Sn (температура солидус 156°C, ликвидус 165°C) может применяться на платах с покрытием печатных проводников сплавом Розе, поскольку в этом случае уменьшается вероятность отслоения припойной маски при плавлении покрытия. Кроме того рекомендуется его применение, если на плате присутствуют компоненты со значимо различными теплоемкостями выводов. При этом обеспечивается гарантированное оплавление припоя без повреждения элементов. При пайке инфракрасным методом наблюдается значительный разогрев пластмассовых корпусов ИМС из-за высокой степени черноты, поэтому применение легкоплавких припоев также желательно. Наиболее легкоплавкий припой – эвтектический сплав 42Sn/58Bi (температура эвтектики 138°C). При повышенной ползучести применяется высокопрочный сплав с сурьмой 99Sn/1Sb (температура плавления 235°C).

Сплав с добавлением индия 3,5Sn/37,5Pb/25In (солидус 134°C, ликвидус 181°C) обеспечивает хорошее смачивание, что желательно для пайки цилиндрических корпусов. Однако при пайке компонентов с высокими выводами может наблюдаться подъем припоя по выводу с разрывом вблизи соединения. Кроме того, этот сплав не может применяться для пайки выводов с золотым покрытием.

В случае использования элементов с позолоченными выводами идеальным является сплав золота с оловом 80Au/20Sn. Однако применение данного сплава ограничено из-за его высокой стоимости.

Характеристики частиц припоя.

На свойства припойной пасты значительное влияние оказывают характеристики частиц припоя. Так вязкость припоя возрастает с уменьшением размера частиц, что является нежелательным. Частицы размером 10 мкм и менее при пайке могут выноситься флюсом за контактные площадки с образованием шариков припоя вне паяного соединения, что приводит к изменению свойств изоляции, а иногда к пробое. Однако увеличение размера частиц ухудшает однородность пасты, в особенности при нанесении через трафарет. Наиболее широко применяются пасты с размером частиц 25...75 мкм.

Для уменьшения абразивного действия на трафареты желательно, чтобы частицы имели форму близкую к сферической. Сферическая форма, обладая минимальной площадью, мало подвержена окислению. Так в пасте RM92 гарантируется не менее 97% частиц сферической формы при соотношении диаметров не более 1,5. Однако сферические частицы оседают в пасте более интенсивно, чем вытянутые и каплеобразные, снижая тем самым срок хранения пасты. Этого можно избежать, производя перемешивание пасты перед нанесением.

В настоящее время для производства порошкообразного припоя наиболее широко применяются методы распыления в инертной газовой среде, эмульгирования в формообразующей жидкости, а также эмульгирование с применением ультразвука.

При распылении наблюдается [1] больший процент частиц с вытянутой и каплеобразной формой. При применении эмульгирования форма частиц ближе к сферической, распределение по фракциям более узкое. Наилучший результат достигается эмульгированием с применением ультразвука. При этом микроструктура частиц более мелкозернистая, сферичность частиц наибольшая, при получении порошка эвтектических сплавов состав сплава ближе к эвтектике.

Содержание припоя в припойной пасте.

Содержание припоя в пасте, как и размеры частиц, влияет на ее вязкость: увеличение доли припоя приводит к росту вязкости пасты, затрудняется ее нанесение через трафарет, возрастают усилия, оказываемые на ракели. При уменьшении количества припоя ухудшается смачивание и растекание пасты при оплавлении.

Обычно в пастах содержится 70–90% частиц припоя по весу.

Влияние содержания припоя и размеров частиц на свойства припойной пасты RM92 приведены в таблице 1.

Таблица 1.

Свойства припойной пасты RM92 при различных размерах частиц и содержании припоя.

Содержание припоя, % по массе		88		90	
Размеры частиц, мкм		75...53	53...38	75...53	53...38 38...25
Вязкость					
По методу Брукфильда, сП		600 000	650 000	800 000	900 000 1 000 000
При скорости сдвига 6с ⁻¹ , П		145	187	208	235 240
Тиксотропный индекс		0,52		0,56	0,6
Растекание, мм					
1 ч, 20°C	площадка 0,7 мм	0,2			
	площадка 1,5 мм				
0,3 ч, 80 °C	площадка 0,7 мм	0,5	0,3	0,4	0,3
	площадка 1,5 мм	0,4	0,3		0,4
Сила отрыва при приклейке, г/мм ²		1,2	1,4	1,1	1,2 1,3

Заключение.

Отечественной и зарубежной промышленностью выпускается большое количество припойных паст на основе припоев широкого применения. Однако влияние порошкообразного припоя на свойства пасты и ее применение настолько велико, что при необходимости получения высококачественных паяных соединений должны использоваться специализированные пасты. Выбор припойного сплава диктуется как покрытием выводов компонентов и контактных площадок, так и технологией нанесения и оплавления пасты. Хотя большинству задач удовлетворяют пасты на основе сплава $62\text{Sn}/36\text{Pb}/2\text{Ag}$ с частицами сферической формы диаметра $38\dots75$ мкм, в настоящее время невозможно выделить какой-либо сплав как оптимальный для всех возможных случаев.

Литература.

5. **Глушкова Л. И.**, Угловой А. Б. Анализ физико-химических свойств порошков-компонентов паяльной пасты, применяемой в поверхностном монтаже: Автоматизация и внедрение прогрессивных процессов пайки в радиоэлектронике, приборостроении и машиностроении: Сборник статей./Под ред. Глушковой Л. И., Костина Д. Т. —Л.: Ленинградский Дом научно-технической пропаганды, 1990. — стр. 35-43.: ил.
6. **Григорьев В. Н.**, Казаков А. А., Джинчарадзе А. К. и др. Монтаж на поверхность: Технология. Контроль качества./Под общ. ред. И. О. Шурчкова —М.: Изд-во стандартов, 1991. —180 [2] с.: ил.
7. **Казаков Н. И.**, Критский В. М. Технология пайки монтажных соединений в производстве РЭА: Учеб. пособие. — М.: М-во радиопромышленности СССР, Ин-т повышения квалификации руководящих работников и специалистов, 1989. — 129 с.: ил.
8. **Материалы** технологические для производства и ремонта электронной аппаратуры: Краткий каталог. — М.: ЗАО Предприятие ОСТЕК, Февраль 1999. — 33с.: ил.
9. **Костин Д. Т.** Анализ тенденций развития технологии пайки паяльными пастами (По материалам зарубежной печати): Автоматизация и внедрение прогрессивных процессов пайки в радиоэлектронике, приборостроении и машиностроении: Сборник статей./Под ред. Глушковой Л. И., Костина Д. Т. —Л.: Ленинградский Дом научно-технической пропаганды, 1990. — стр. 3-11.
10. **Применение** поверхностного монтажа в радиоэлектронной аппаратуре. — Смоленск: Союз научных и инженерных обществ СССР, Смоленское областное правление, 1990. — 152 с.: ил.

РЕАЛИЗАЦИЯ ОПТИМИЗАЦИИ НЕСУЩИХ КОНСТРУКЦИЙ ВЕРОЯТНОСТНЫМ МЕТОДОМ В ППП «ОПТИМА»

Кяжкин А.В.

к.т.н., профессор Чеканов А.Н.

МГТУ им Баумана. г.Москва. Россия.

OPTIMIZATION OF LOAD-CARRYING STRUCTURES
WITH PROBABILISTIC METHOD IN PROJECT "OPTIMA"

Kjazhkin A.V.

Professor Chekanov A.N.

MSTU after Bauman. Moscow. Russia

zsb@atom.ru

Аннотация. В докладе отражены основные аспекты разработки пакета прикладных программ для автоматизированной вероятностной оптимизации в прочностных расчетах несущих конструкций. Кратко изложены теоретические основы оценки прочности конструкций вероятностными методами. А также проведен краткий анализ перспективности предложенного пакета в сравнении с существующими программами.

Abstract. The main aspects of developing the project for automatic probabilistic optimization in stress calculations of load-carrying structures are mentioned. Theoretical basics of stress analysis of load-carrying structures with probabilistic methods are written briefly. Also a short analysis of the perspectives of suggested program in comparison with existing one is done.

Радиоэлектронная аппаратура, работающая в условиях сложных внешних воздействий, требует определения несущей способности ее опорных конструкций. Однако в случае статически неопределимых систем несущую способность всей конструкции невозможно выразить однозначно, например, в виде оценки запаса прочности. Даже использование характеристики запаса прочности, в вероятностной трактовке, не позволяет оценить допустимые нагрузки всей несущей конструкции, поскольку и эта характеристика не обладает интегральными свойствами.

Разработанный нами пакет программного обеспечения позволяет обосновать распределение приложенных нагрузок с целью увеличения вероятности безотказной работы несущих конструкций в условиях ограниченности массогабаритных характеристик. В практике конструирования эта задача получила большее распространение, чем двойственная ей задача оптимизации, когда из условия заданного уровня вероятности безотказной работы добиваются минимальной массы конструкции. Основные функции рассматриваемой задачи - выбор направления поиска в сторону оптимального решения.

В основу построения вероятностной модели положен математический аппарат, основанный на том, что каждый элемент несущей конструкции обладает определенной несущей способностью по отношению к нагрузкам. Взаимное соотношение параметров прочности и нагрузки определяет отказ элементов, а к факторам, активно влияющим на отказ, можно отнести свойства материала, статистические данные о нагрузке, концентрации напряжений, геометрические разбросы - допуски на геометрию элементов и т.п. Вероятностная методика оценки элементов несущих конструкций позволяет наиболее полно учитывать многие случайные величины и законы их распределения.

Максимизация вероятности безотказной работы системы при заданном ограничении на массу конструкции производится с использованием метода неопределенных множителей Лагранжа. Однако в классической трактовке метод требует построения системы уравнений с учетом ограничения и решения ее с целью нахождения оптимальной точки. Воспользоваться методом в такой постановке весьма сложно, так как он требует составления большого числа уравнений, описывающих работоспособность многочисленных разнообразных элементов конструкции. Использование метода ограничено из-за того, что полученные сложные функции работоспособности необходимо продифференцировать, чтобы получить первую производную, как требует метод Лагранжа.

С целью решения задачи оптимизации несущей конструкции использован искусственный прием, который позволил разрешить вышеприведенные проблемы, но потребовал дополнительных итераций при оптимизации. Смысл адаптации метода множителей

Лагранжа заключается во введении коэффициента перераспределения массы. Между массой элемента и вероятностью безотказной работы существует зависимость, получить которую в математической форме достаточно сложно, однако очевидно, что увеличение массы элемента при статической оптимизации ведет к его более высокой надежности.

Вводя коэффициент вариации массы в метод множителей Лагранжа, можно добиться выполнения двух условий:

- 1) объединения уравнения работоспособности элемента и уравнений ограничения общей переменной, которая и определяется путем решения системы уравнений;
- 2) упрощения системы уравнений, что позволяет легко найти первую производную и, как следствие, быстро решить задачу на ЭВМ даже при большом числе элементов несущей конструкции.

Для того, чтобы в пределах одного пакета прикладных программ можно было одновременно проводить анализ и оптимизацию путем итерационного поиска, необходимо вызывать циклически ту часть пакета, которая выполняет задачу анализа. В качестве этой части была создана программа, позволяющая определять внутренние силовые факторы различных типовых конструкций с помощью метода конечных элементов.

Основной единицей представления конструкции в этом анализе является конечный элемент - геометрически упрощенное представление малой части физической конструкции. Модель конструкции создается из дискретных конечных элементов, которые образуют сетку (решетку). Созданная для расчета конечно-элементная модель - это лишь приближенное представление реальной конструкции, позволяющее анализировать ее реакцию на внешнее воздействие.

Предложенный пакет позволяет также проводить анализ конструкций, работающих в динамическом режиме нагружения, что дает возможность использовать эту программу для решения задач динамической оптимизации. Для любой модели может быть найден набор низших частот свободных колебаний и соответствующие им формы колебаний. Можно также вычислять напряжения, соответствующие определенным формам колебаний и получить изображение деформированной конструкции. Конструкция может быть подвергнута действию нагрузки, изменяющейся во времени по заданному закону (можно задавать линейные, кусочно-линейные, гармонические, случайные законы, а также реализации землетрясений и прикладывать такие воздействия независимо или в различных сочетаниях).

Среди современного программного обеспечения имеются аналоги разработанного пакета прикладных программ, позволяющие решать задачи по проектированию и оптимизации несущих конструкций. Однако следует заметить, что критериями оптимизации в этих программах являются, в основном, предельные значения физических параметров, такие как действующее механическое напряжение по критерию Мизеса, масса несущих конструкций при заданном пределе текучести и т.д. Среди конкурирующих пакетов были рассмотрены:

1. Cosmos. Разработчик – фирма Digital Solutions. Анализирует несущие конструкции на воздействие статических и динамических нагрузений. Позволяет проводить только однопараметрическую оптимизацию. Сложный, запутанный интерфейс. Для решения серьезных задач не пригоден. Операционная система MS-DOS, минимальные требования к аппаратному обеспечению – компьютер на базе процессора 486.

2. Ansys – В своем составе этот расчетный комплекс имеет средства анализа несущих конструкций. Оптимизация однопараметрическая. Нет возможности проводить вероятностную оптимизацию. Используется для решения широкого круга научных и практических задач анализа несущих конструкций. Операционная система MS-DOS, минимальные требования к аппаратному обеспечению – компьютер на базе процессора 486.

3. Nastran – Как и Ansys имеет средства анализа несущих конструкций. Основное достоинство этого пакета – универсальность программного интерфейса, что дало возможность использования его как расчетного блока таких САПР, как Unigraphics, Pro/Engineer и др. Ориентированность этого пакета прикладных программ на работу в составе других САПР привела к тому, что выполнение оптимизации занимает много времени. Связано это с необходимостью переключения между Nastran и другой САПР. Операционная система – Windows, минимальные требования к аппаратному обеспечению – компьютер на базе процессора Pentium.

4. Pro/Mechanica – Разработчик фирма Parametric Solutions. Является неотъемлемой частью системы Pro/Engineer. Оптимизация может быть многопараметрической, но нет возможности проводить вероятностную оптимизацию. Операционная система – Windows NT, Unix, Solaris. Интерфейс позаимствован из системы Unix, что несколько необычно для пользователей России, привыкших видеть стандартный интерфейс Windows. Минимальные требования к аппаратному обеспечению – компьютер на базе процессора Pentium.

Исходя из проведенного выше сравнения, можно сделать вывод о том, что отсутствие вероятностной оптимизации делает существующие программы малоприспособленными для оценки несущей способности сложной статически неопределимой конструкции.

Литература

1. Чеканов А.Н. Вероятностные расчеты и оптимизация несущих конструкций. Учебное пособие. – М.: ИЧП «Издательство Магистр», 1997.
2. Вентцель Е.С. Теория вероятностей. – М.: Физматгиз, 1962.
3. Капур К., Ламберсон Л. Надежность и проектирование систем. – М.: Мир., 1980.

4. Смотрицкий Х.Л., Чукреев П.А. К вопросу об оптимальном резервировании аппаратуры.
– Известия АН СССР, сер. Энергетика и автоматика, №4, 1959.

АНАЛИЗАТОР РАБОТОСПОСОБНОСТИ ЦИФРОВЫХ МИКРОСХЕМ ТТЛ-ТЕХНОЛОГИИ.

Померанцев К.В.

Научный руководитель: ст. преподаватель Соловьев В.А.

МГТУ им.Н.Э.Баумана, ИУ 4-13

ANALYZER OF FUNCTIONABILITY OF DIGITAL MICROCIRCUITS OF TTL-TECHNOLOGY.

Pomeranchev K.V.

The scientific chief: an item the teacher Solovjev V.A.

BMSTU, IU4

Аннотация. Анализатор микросхем предназначен для логического анализа микросхем без демонтажа. Данное устройство было мною спроектировано, описано в данном докладе и выполнено в макетном варианте. К анализатору микросхем была написана программа на языке программирования “VISUAL BASIC 3.0”. Это устройство было протестировано на предмет прохождения информации от компьютера к выходным каскадам усиления и обратно. Анализатор микросхем был проверен в работе. Все мои работы связанные с анализатором микросхем, я проводил на ОАО “МСЗ”, которое является режимным предприятием

The summary. The analyzer of microcircuits is intended for the logical analysis of microcircuits without dismantling. The given device was designed by me, is described in the given report and is executed in mock-up version. To an analyzer of microcircuits the program on the programming language « VISUAL BASIC 3.0 » The device was written was tested for transit of the information from the computer to output stages of strengthening and back. The analyzer of microcircuits was checked in activity. All my activities bound with an analyzer of microcircuits, I conducted on JSC «MSZ», which one is regime firm

ВВЕДЕНИЕ.

Цифровые вычислительные системы ставят уникальные проблемы поиска неисправностей, что объясняется их организацией и структурой. Как и в других электронных системах, компоненты компьютера по разным причинам отказывают, что приводит к частичному или полному выходу системы из строя.

Значительная часть компонентов компьютера реализована на больших интегральных схемах (БИС), благодаря которым законченные подсистемы компьютера поставляются в виде готовых блоков. Примером такого блока в виде БИС служит сам микропроцессор. Архитектура вычислительной системы часто превращает поиск неисправного компонента в сложную и длительную процедуру. По мере все большего уплотнения элементов в БИС проблемы их функционального тестирования резко усложняются. Любая БИС до поставки подвергается лишь ограниченному тестированию, а отказы могут проявиться при эксплуатации из-за возникновения комбинаций сигналов, которые не применялись в процессе тестирования, или таких изменений некоторого параметра компонента, которые вызывают нехарактерное поведение МС.

В процедуре отыскания неисправностей видное время занимает измерение уровня сигнала или наблюдения сигнала с помощью осциллографа в определенных точках его прохождения через исследуемое устройство. Если на входе какого-либо каскада сигнал нормальный, а на выходе отсутствует или его форма искажена, то каскад неисправен. Поиск неисправностей можно определить как логическую технику обнаружения и устранения причин нарушения работоспособности электронных систем. Процесс обнаружения неисправности является, в первую очередь, процессом логического мышления.

Поиск неисправностей представляет собой важную задачу. Экономика и образ жизни в развитых странах зависят от поддержания в рабочем состоянии электронных систем управления в промышленности, систем связи и вычислительной техники. Приблизительно треть бюджета Отдела оборонной электроники ассигнована на обслуживание аппаратуры, что характеризует важность отыскания неисправностей и их устранения для современного общества.

В начальный период частота отказов очень велика по сравнению со средним периодом работы. По этой причине многие компании используют испытания на принудительный отказ, т.е. работу компонентов или оборудования в целом при повышенных температурах для ускорения появления ранних отказов. Дефекты устраняются до того, как оборудование выйдет с завода-изготовителя, интенсивность его отказов у потребителя будет соответствовать нижней части кривой графика. По окончании периода приработки оборудования наступает период нормальной его работы, который длится примерно 5-25 лет, в зависимости от надежности, заложенной в конструкции и реализованной в аппаратуре.

Чем больше у нас информации о системе, тем уверенней мы можем отыскивать в ней неисправность. Наилучшим источником информации для быстрого и достаточно подробного ознакомления с устройством, оборудованием является инструкция по эксплуатации и техническому обслуживанию. Инструкция обычно содержит технические условия на оборудование, его описание и принцип действия, схемы, компоновку деталей и руководство по отысканию неисправностей. Это руководство включает перечень причин и признаков часто встречающихся повреждений

Для обнаружения неисправностей надо правильно выбирать контрольно-измерительную аппаратуру. Например: анализатор спектра полезен при отыскании неисправностей в системе связи, однако его бесполезно применять для проверки цифровой аппаратуры. Простой авометр незаменим при поиске обрывов цепи. При отыскании некоторых видов неисправностей в цифровой аппаратуре может понадобиться логический анализатор.

Установка в систему заведомо годных схем, печатных плат с целью устранения повреждения, называется заменой компонентов схем, плат. Как метод поиска неисправности это очень дорого обходится. Множество годных компонентов и печатных плат было испорчено при установке их в систему при включенном питании или в систему, не устраненная неисправность которой вызвала отказ заменяемой детали и соответственно вновь установленной детали. Когда аппаратуру нужно быстро ввести в действие, замена печатной платы или модуля в случае отказа оказывается оправданной. Это эффективный способ свести к минимуму время простоя.

Существует еще один способ отыскания неисправностей – сравнение с известными правильными результатами. Чтобы опознать неправильный выходной сигнал, мы обычно сравниваем его с выходным сигналом нормально функционирующей схемы. Мы должны или достаточно хорошо знать схему, чтобы определить каким должен быть выходной сигнал, или сравнить его с формой сигнала в инструкции по эксплуатации и обслуживанию. Если под рукой нет документации, в которой были бы приведены правильные выходные сигналы, их нужно сравнить с аналогичными сигналами такой же нормально работающей аппаратуры. Этот метод лучше всего подходит для цифровых устройств и аппаратуры с использованием микропроцессоров.

С такими же проблемами мы встречаемся и при проверке ИС, когда они выходят из строя. Для экономии времени и денег, для облегчения и ускорения нахождения неисправностей, и ремонта мною было разработано устройство для проверки интегральных схем (анализатор) без демонтажа и написана программа. Для подключения анализатора микросхем к персональному компьютеру я спроектировал плату цифрового ввода/вывода (плата ЦВВ). Кроме того, плата ЦВВ может работать с другими устройствами, необходимыми на производстве, например устройство имитации интерфейса.

1. Принцип работы

1.1 Принцип работы платы ЦВВ.

Плата ЦВВ предназначена для работы с:

1. устройством для имитации интерфейса (для проверки плат УСО (устройство связи с объектом))
2. устройством контроля и управления объектом
3. с анализатором (устройством) для проверки интегральных схем

с целью повысить производительность работы и упростить работу с другими внешними устройствами.

Плата ЦВВ состоит из четырех блоков, каждый из которых имеет свое определенное назначение.

Блок ввода/вывода данных.

Предназначен для передачи данных от внешнего устройства к компьютеру и обратно.

Блок задержки.

Блок задержки выдает сигнал готовности, который имеет определенную задержку, и служит для повышения надежности работы компьютера с внешним устройством. На вход блока поступает тактовая частота компьютера и сигнал, подтверждающий обращение к плате ЦВВ. На выходе блока формируется сигнал готовности, информирующий компьютер о продолжении работы.

Блок согласования управляющих сигналов и сигналов адреса.

Предназначен для усиления сигналов управления и четырех младших бит адреса, поступающих к внешнему устройству. Блок обеспечивает адресацию шестнадцати портов ввода/вывода и управляет направлением передачи данных.

Блок адресации.

Необходим для того, чтобы плата ЦВВ распознала, что именно к ней обращается компьютер. Блок запрещает работу платы ЦВВ, если имеется прямой доступ к памяти компьютера.

При любой операции, связанной с посылкой или приемом данных, на шину адреса выставляется адрес, по которому данные выдаются или записываются. Но существует различие в адресации памяти компьютера и портов ввода/вывода.

Плата ЦВВ подключается к порту ввода/вывода компьютера, следовательно, адрес платы ЦВВ должен быть выбран на основании изучения карты портов.

Блок адресации выдает сигнал разрешения работы если компьютер обращается по одному из этих адресов:

01E0 – 01EF

01F0 – 01FF

03E0 – 03EF

Выбрано три группы не занятых адресов (в соответствии с картой адресов) т.к. некоторые из них могут быть заняты другими устройствами, подключенными для расширения возможностей компьютера. Уровни напряжения логического нуля и единицы соответствуют стандартным уровням ТТЛ логики, как в компьютере, так и во внешних устройствах.

Проектирование платы ЦВВ.

При проектировании платы ЦВВ основная задача состоит в том, чтобы исходя из имеющихся на заводе возможностей добиться согласованной работы IBM PC с внешними устройствами.

При минимальной сложности схемы платы ЦВВ должен быть надежным и обеспечивать работу с шестнадцатью портами ввода/вывода, что необходимо для работы платы ЦВВ с устройствами.

Плата ЦВВ представляет собой стеклотекстолитовую основу с нанесенными с двух сторон медными проводниками (дорожками) и отверстиями для установки микросхем, радиоэлементов и разъема. Форма и размеры платы ЦВВ позволяют устанавливать ее в разъем компьютера. Конструктивно плата ЦВВ имеет два разъема: один для подключения к интерфейсу ISA, второй для подключения внешних устройств.

Схема платы ЦВВ выполнена на элементах:

К589АП16 - 2 шт.

КР580ВА86 - 1 шт.

КМ555ТМ2 - 2 шт.

КР531ЛА2 - 1 шт.

КР531ЛА3 - 1 шт.

К155ЛА7 - 1 шт.

Кроме того на плате установлены:

резистор МЛТ-0,125 - 1 шт.

конденсатор КМ5-М47 - 2 шт.

конденсатор К50-16 - 1 шт.

разъем для связи с ВУ - 1 шт.

Принципы системо-технической реализации платы ЦВВ.

В основу принципа работы платы ЦВВ положена одновременная передача данных и адреса, по которому эти данные передаются. Управление работой платы ЦВВ осуществляется с помощью персонального компьютера.

Для обеспечения работы платы ЦВВ на разъем X1A поступает обращение с компьютера (адрес). Чтобы исключить работу с памятью компьютера напрямую, в схеме адресации предусмотрен инвертор DD3.4, который при появлении высокого уровня на контакте 11 (сигнал AEN) разъема X1A запрещает работу блока. В результате этого на выходе микросхемы DD5 появляется высокий уровень, запрещающий и переводящий в третье состояние шинные формирователи DD1 и DD2. DD1 и DD2 обеспечивают обмен данными компьютера и внешнего устройства по 8-разрядной шине данных.

При наличии низкого уровня на входе EZ, что свидетельствует об адресации к плате ЦВВ, и "1" или "0" на входе SE формирователи откроются для передачи данных, соответственно от компьютера к внешнему устройству или от внешнего устройства к компьютеру.

Для согласования управляющих сигналов предусмотрен двунаправленный шинный формирователь DD6, передающий сигналы управления к ВУ.

Для расширения возможностей платы ЦВВ и работы его с внешними устройствами ("Анализатор для имитации интерфейса" и "Анализатор для проверки интегральных микросхем") на выходной разъем заводятся четыре младших разряда адреса с компьютера, также через шинный формирователь DD6. Это позволяет работать с шестнадцатью портами ввода / вывода. Принципиальная схема платы ЦВВ содержит элемент DD9.2, который служит для инверсии и передачи частоты внешнему устройству.

Кроме того, плата ЦВВ имеет схему задержки на триггерах DD7 и DD8 и инверторе DD9.1. Схема задержки позволяет обеспечить запас времени для обмена информацией между компьютером и внешним устройством. Этот запас времени необходим в связи с особенностями работы внешнего устройства и задержкой, возникающей в интерфейсе. Время запаса выбрано экспериментально, путем постепенного снижения до минимально допустимой.

В момент прихода сигнала “Работа” (низкий уровень на выходе микросхемы DD5), свидетельствующего об адресации платы ЦВВ, на вход D микросхемы DD7.1, по первому положительному фронту, на выходе \overline{Q} появляется высокий уровень. Через инвертор DD9.2 сигнал с выхода \overline{Q} микросхемы DD7.1 приходит на 10 контакт разъема X1A (сигнал I/O CH RDY), информируя компьютер о “снятии” готовности платы ЦВВ принимать новую информацию. Сигнал с выхода \overline{Q} микросхемы DD7.1 подается на вход D микросхемы DD7.2. По второму такту на выходе Q микросхемы DD7.2 появляется высокий уровень, который в свою очередь, приходя на вход D микросхемы DD8.1 приводит к появлению на выходе Q этой микросхемы высокого уровня по третьему такту. С выхода Q этой микросхемы сигнал приходит на вход D микросхемы DD8.2 и на четвертый такт на выходе \overline{Q} этой микросхемы появляется низкий уровень, который, приходит на вход -S микросхемы DD7.1, и входы -R элементов DD7.2 и DD8.1 и устанавливает DD7.1 в единичное состояние (на выходе \overline{Q} – низкий уровень) а элементы DD7.2 и DD8.1-в нулевое состояние.

За время нахождения на выходе \overline{Q} микросхемы DD7.1 высокого уровня, происходит обмен информацией между компьютером и внешним устройством.

После приведения микросхемы DD8.1 в нулевое состояние, на вход D микросхемы DD8.2 приходит низкий уровень, который по следующему – пятому такту приходит на выход Q и на выходе \overline{Q} микросхемы DD8.2 появляется высокий уровень, разрешающий работу микросхемы DD7.1 на шестой такт (при условии прихода сигнала “Работа”, свидетельствующего об адресации платы ЦВВ).

За время этих тактов состояние сигнала адресации безразлично.

Примечание:

Обмен информацией в том или ином направлении происходит по команде компьютера при наличии соответствующего сигнала чтения или записи. Шина данных компьютера отличается от шины данных внешнего устройства и является двунаправленной. У внешнего устройства шина данных состоит из шины выдачи данных и шины приема данных. По существу это является обоснованием применения шинных формирователей.

1.2 Функциональный состав анализатора микросхем.

Анализатор, в общем, состоит из компьютера, интерфейсной платы ЦВВ и самой платы анализатора.

Анализатор состоит из шести блоков, каждый из которых имеет свое определенное назначение:

Дешифратор адреса (микросхема DD7) – для выбора микросхем портов ввода/вывода.

Буфер данных (микросхема DD2) – для выдачи данных или для их приема в строго определенные моменты времени.

Программируемые порты ввода/вывода (микросхемы DD3, DD10) – через них осуществляется запись и чтение данных.

Блок регистров (микросхемы DD1, DD4, DD5, DD8) – для одновременной записи по всем шестнадцати каналам.

Блок усиления выходных сигналов (транзисторы, резисторы) – для выдачи нужных уровней сигналов с большим током.

Блок приемников (микросхемы DD9, DD11) – для организации согласования длинного кабеля (универсальных шин данных).

Проектирование анализатора микросхем.

Устройство для проверки интегральных схем, которое в дальнейшем будем называть анализатором, предназначено для проверки, анализа работоспособности микросхем без демонтажа с целью повысить производительность работы. Анализатор предназначен для работы с готовой платой ЦВВ. Анализатор представляет собой стеклотекстолитовую основу с нанесенными с двух сторон медными проводниками (дорожками) и отверстиями для установки микросхем, радиоэлементов и разъемов.

Схема анализатора выполнена на элементах:

КР580ВА86	- 1 шт.
КР580ВВ55А	- 2 шт.
КР580ИР82	- 2 шт.

KP580IP83	- 2 шт.
KP531АПЗ	- 2 шт.
KP531ЛА3	- 1 шт.
K155ИД4	- 1 шт.

Кроме того на плате установлены:

конденсатор KM50-16	- 4 шт.
конденсатор KM5-M47	- 16 шт.
резистор МЛТ-0,5	- 16 шт.
резистор МЛТ-0,125	- 80 шт.
транзистор КТ816А	- 16 шт.
транзистор КТ817А	- 16 шт.
разъем для подключения к плате ЦВВ	- 1 шт.

Управление работой анализатора осуществляется с помощью персонального компьютера.

Для связи между анализатором и платой ЦВВ служит кабель, изготовленный из провода марки МГШВ-0,14 и двух разъемов. Для связи анализатора микросхем с проверяемой микросхемой служит кабель, изготовленный из провода марки МГШВ-0,14, и клипсой на 16 контактов.

Принципиальная электрическая схема анализатора микросхем.

Для обеспечения работы анализатора с разъема X1 на шинный формирователь DD2 поступают данные с платы ЦВВ. На шину адреса выставляется адрес первого порта $A3=0$, $A2=0$ и через дешифратор DD7 выбирается микросхема DD3. На шину данных D выставляется управляющее слово и по сигналу записи происходит запись управляющего слова, при этом выбирается режим работы порта (нулевой режим – все три порта на вывод). $A0=1$, $A1=1$ – выбор режима программирования микросхемы. По адресу $A3=0$, $A2=1$ – выбирается микросхема DD10 и аналогично записывается управляющее слово. DD10 работает в нулевом режиме – порт А – вывод, порты В, С – чтение. В начальном состоянии все выходы микросхемы DD3=0 и выходы канала А микросхемы DD10=0. В связи с тем, что сигналы, подаваемые на верхнюю и нижнюю группы транзисторов, должны быть разные (во избежании короткого замыкания), микросхемы DD1 и DD5 стоят с инверторами, а значит сигналы с порта А и С микросхемы DD3 инвертируются и поступают на базу транзисторов (с порта В (DD3) и А (DD10)– без инверсии). Если на верхней группе транзисторов – единичные сигналы и на нижней – нулевые, все транзисторы закрыты, на выходе третье состояние.

Что бы подать на вывод проверяемой микросхемы единичный сигнал, нужно открыть соответствующий транзистор верхней группы (VT1-VT16), при этом соответствующий транзистор нижней группы (VT17-VT32) должен быть закрыт. Если требуется подать нулевой сигнал — нужно открыть транзистор нижней группы, при этом соответствующий транзистор нижней группы надо закрыть. Ну а если этот вывод микросхемы является выходом--то нужно закрыть соответствующие транзисторы верхней и нижней групп. Транзистор VT1 верхней группы соответствует транзистору VT17 нижней группы, VT2—VT18...и т.д. Но не должны быть открыты сразу транзистор верхней группы и ему соответствующий транзистор нижней группы. Для того чтобы открыть транзистор верхней группы нужно подать на базу этого транзистора нулевой уровень, для открытия транзистора нижней группы соответственно единичный уровень. Для управления выходами с блока усиления сигналов для каждого выхода надо подавать два бита (один занимает свое место в регистре DD1 (DD5), другой соответствующее место в регистре DD4 (DD8)). При записи единиц в порты А и С микросхемы DD3 на выходе регистров соответственно ноль. При записи единиц в порты В микросхемы DD3 и А микросхемы DD10 на выходе регистров – “1”.

По очереди в каждый порт записываются данные в соответствии с алгоритмом проверки. Причем, если нужно получить третье состояние (этот вывод микросхемы является выходом), то в эти биты мы записываем нули. Если необходимо подать единичные уровни на проверяемую микросхему, то записываем в порты А и С микросхемы DD3 единицы. Если нулевые, то единицу мы записываем в порт В микросхемы DD3 и порт А микросхемы DD10. В соответствующих битах в это время должен быть ноль.

Обязательное условие для проверки программы: не должно быть открыто двух транзисторов - нельзя открывать два транзистора, верхний и нижний сразу – короткое замыкание.

Микросхема DD6 после записи в порты переписывает информацию в регистры (микросхемы DD1, DD4, DD5, DD8) – информация появляется на всех шестнадцати выходах одновременно. Информация со входа микросхем DD1, DD4, DD5, DD8 записывается на выход, открываются соответствующие транзисторы и заданные уровни подаются на проверяемые микросхемы.

Микросхема DD6.1 – инвертор - идет на вход С микросхем DD1, DD4, DD5, DD8 – запись.

Микросхемы DD6.2 и DD6.4 – RS-триггер на включение регистров (сигнал Е – выборка микросхем).

После чего считываются все уровни с проверяемых микросхем через приемопередатчик (шинный приемник) микросхем DD9, DD11 с портов В и С микросхемы DD10. Информация

считывается в компьютер и ее обрабатывает программа (сравнивает с эталоном), после чего выдает результаты на экран.

Назначение сопротивлений.

R1-- R32 – для ограничения тока базы (не перегрузить микросхемы).

R33 -- R64 – для быстрого закрытия транзистора.

R65 -- R80 – для ограничения выходного тока (выходной ток через резистор от транзисторов).

R81 -- R96 – нагрузочное сопротивление (если проверяется микросхема с открытым коллектором, то они являются нагрузкой, для остальных – не влияют).

Результаты и выводы

На отдельных этапах производственного процесса анализатор микросхем может заменить целую группу приборов: осциллограф, вольтметр, пробник, генератор импульсов. Для более эффективного использования рабочего времени и сокращения времени простоя этих приборов целесообразно по возможности использовать анализатор микросхем, в то время как, например осциллограф можно использовать для других целей.

Анализатор микросхем предназначен для логического анализа микросхем без демонтажа. Использование в производственном процессе анализатора микросхем существенно сокращает затраты рабочего времени, необходимого для ремонта сложных электронных блоков, а это, естественно, снижает себестоимость ремонта, т. к. сокращается не только время ремонта, но и упрощается сам процесс поиска неисправности, а значит эту работу может выполнять рабочий более низкой квалификации.

При разработке проектируемого устройства, в частности, принципиальных электрических схем было учтено, что вместо транзисторов можно использовать интегральные микросхемы, которые гораздо меньше по весу и объему, более надежны и удобны при монтаже и дешевле. Поэтому транзисторы оставлены только в блоке усиления выходных сигналов для обеспечения мощного по току сигнала, т. к. выходной ток интегральных микросхем очень слабый.

Данное устройство было изготовлено в макетном варианте, а также было проверено в работе. Анализатор микросхем значительно облегчил работу оперативно-ремонтного персонала.