

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физический факультет

Кафедра радиофизики

## СИСТЕМЫ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТАЛЬНЫХ УСТАНОВОК

Методические указания к вводной лабораторной работе  
практикума ТСАНИ

Новосибирск  
2015

**УДК 53:004.9(075.8)**  
**ББК В185.505**  
**Б28**

*Батраков А. М. Системы автоматизации экспериментальных установок* : Метод. указ. ; Новосиб. гос. ун-т. – Новосибирск : РИЦ НГУ, 2015. – 16 с.

Работа является введением в практикум и даёт общее представление об автоматизации экспериментов, формулирует и описывает основные понятия, применяемые в данной области.

Рецензент *Г. И. Кузин*

Ответственная за выпуск *О. А. Тенкеджи*

© Новосибирский государственный университет, 2015  
© А. М. Батраков, 2015

## 1. Почему и зачем нужны системы автоматизации экспериментальных установок

Современные экспериментальные исследования немислимы без использования ЭВМ. Хорошо известны и понятны функции компьютера как средства автоматизации при подготовке публикаций, презентаций и представления результатов работы. Можно представить себе функции компьютера и при обработке экспериментальных данных. И в том и в другом случае его использование заметно увеличивает эффективность деятельности исследователя. Однако нет никаких принципиальных ограничений на то, что подобные работы не могут быть выполнены «вручную», без использования вычислительной техники. Статьи можно писать – и неплохо – ручкой на бумаге, графики строить «по клеточкам» на миллиметровке, а обчислять экспериментальные данные с помощью простого калькулятора.

В отличие от этих финальных стадий научной работы, получение экспериментальных данных в необходимом количестве и соответствующего качества в современных исследованиях принципиально невозможно без систем автоматизации экспериментальных установок. Эти системы включают в себя как средства автоматизированного управления и контроля, позволяющие реализовать сложные алгоритмы управления установкой, так и автоматизированные системы сбора данных, предоставляющие экспериментатору необходимые данные.

Действительно, если для нормальной работы вашей установки необходимо взаимосогласованным образом изменять всего несколько параметров (токи – напряжения – температуры – положение – время – мощность), причём погрешность согласования (точность перестройки) должна быть не хуже 1 %, или 0,1 %, а для особо тонких экспериментов – 0,01 % – 0,001 %, то выполнить это «вручную», без средств автоматизированного управления, невозможно. Можно представить, насколько усложняется задача, если таких «каналов управления» не десяток, а несколько сотен, как в установках по исследованию плазмы, или тысяч, как в современных ускорителях.

А как быть, если необходимо сутками измерять амплитуды импульсов в нескольких десятках тысяч каналов, да ещё и статистически обрабатывать эти данные, что является типовой задачей систем регистрации в ядерно-физических экспериментах?

Подобных примеров можно привести немало, и вывод из них следует очевидный – современные исследования нереальны без адекватных решаемым задачам систем автоматизации экспериментальных установок. Что же включает в себя понятие «системы и средства автоматизации экспериментальных установок»?

## 2. Системы и средства автоматизации экспериментальных установок

Под системой автоматизации подразумевается набор электронных устройств, объединённых и подключённых с помощью специализированных линий связи к управляющему компьютеру, снабжённому соответствующими программами. Задача системы автоматизации – выполнение значительной части операций по управлению установкой и сбору данных без участия оператора. Структура простейшей системы автоматизации показана на рис. 1.

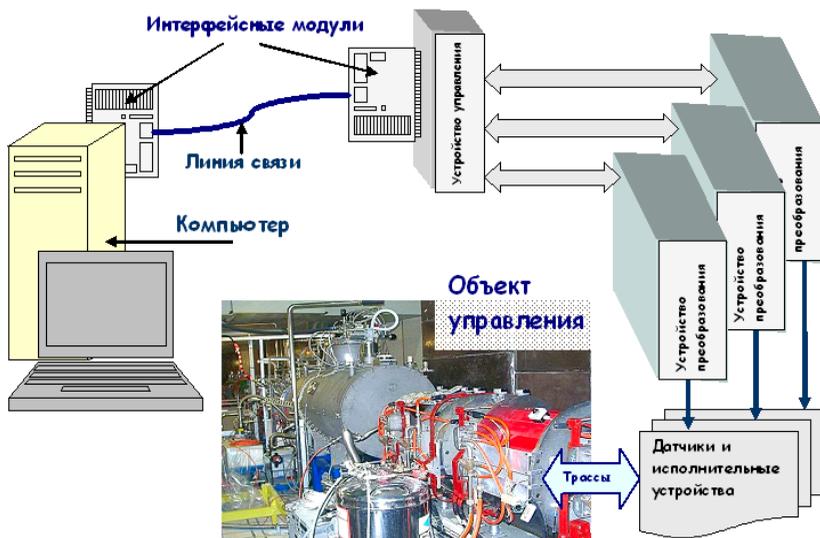


Рис. 1. Структура системы автоматизации установки

Система автоматизации содержит следующие технические средства:

1. Управляющий компьютер.
2. Интерфейсный модуль, обеспечивающий подключение управляющего компьютера к линии связи.
3. Линия связи для соединения интерфейсного модуля управляющего компьютера и устройств управления системой автоматизации.
4. Устройства управления / интерфейсы, подключенные с одной стороны к линии связи, а с другой – к периферийным устройствам преобразования. Способы подключения периферийных устройств к устройствам управления могут быть различными.
5. Устройства преобразования, соединённые с датчиками или исполнительными элементами.

Остановимся на некоторых деталях технических (или аппаратных) средств автоматизированных систем.

*Линия связи* – совокупность проводов, объединённых по функциональному признаку и служащая целям передачи/приёма электрических сигналов между устройствами в соответствии с определёнными правилами. В автоматизированных системах вместо понятия «линия связи» чаще используется термин «Магистраль» или «Шина». Правила, по которому сигналы передаются/принимаются, называются «Протокол шины». Более строгие определения этих терминов принципиальный читатель может найти в [1].

*Интерфейсные модули* осуществляют преобразование внутренних сигналов шины компьютера и устройств управления системой в сигналы линии связи, а также обеспечивают выполнение протокола при передаче сигналов.

Назначение *устройств преобразования* – обеспечение перевода с «цифрового языка» компьютера и интерфейсов на «аналоговый язык» оконечных исполнительных устройств, являющихся частью установки. Хорошо известно, что компьютеры работают с числами, т. е. величинами дискретными, принимающими конечное множество значений. Разница двух дискретных величин не может быть меньше некоторого минимального значения. С другой стороны, весь реальный мир является миром величин аналоговых, принимающих бесконечное количество сколь угодно близких значений из непрерывного множества и описываемых непрерывной функцией времени. Аналоговые величины (или сигналы) используются для представления непрерывно изменяющихся тока, температуры, давления, напряжения, угла и т. п.

Естественно, что для преобразования этих сигналов в понятный для компьютера цифровой вид необходимы специальные устройства. Такие устройства получили название аналого-цифровые преобразователи (АЦП). Существуют устройства, выполняющие обратное действие: преобразование цифровых посылок компьютера в аналоговый сигнал. Они называются цифро-аналоговыми преобразователями (ЦАП). АЦП и ЦАПы образуют два самых больших класса электронных устройств, входящих в понятие «технические средства автоматизации».

Кроме цифро-аналоговых и аналого-цифровых преобразователей широко применяются так называемые устройства дискретного ввода/вывода (часто называемые регистрами ввода/вывода цифровых данных или I/O-регистрами). Типичное применение I/O-регистров – определение (измерение) состояний «включено/выключено» и управление дистанционными выключателями.

В системах автоматизации ядерно-физического назначения одними из массовых устройств являются счётчики импульсов, предназначенные для подсчёта событий, прошедших предварительный отбор по определённой совокупности критериев.

Для измерения интервалов времени и генерации временных последовательностей служат разнообразные времязадающие и времяизмеряющие устройства, составляющие группу «таймеров».

Как правило, для автоматизации даже небольшой установки необходимо объединить в систему различные электронные устройства. Функцию управления такой комбинацией возлагают на устройство управления, называемое *контроллером системы*. Часто контроллер содержит интерфейс для подключения к линии связи. Если система требует непрерывного взаимодействия с управляющим компьютером и пропускной способности линии оказывается недостаточно, применяют «интеллектуальный» контроллер со встроенным процессором, который берёт на себя часть операций и разгружает линию связи. Одно из распространённых названий интеллектуального контроллера системы – CPU (Central Processing Unit).

### 3. Стандарты систем автоматизации и протоколы объединительных шин

Для того, чтобы различные электронные устройства можно было собрать в систему, эти устройства и способы их взаимодействия должны удовлетворять единым правилам. Совокупность таких правил в зависимости от индивидуальных свойств системы называется либо стандартом системы, либо протоколом шины. Система удовлетворяет требованиям того или иного стандарта, если входящие в неё элементы конструктивно, электрически и логически совместимы. Если же единым правилам удовлетворяют интерфейсы и линии связи, а исполнение элементов системы произвольно, то в системе выполняется лишь протокол шины.

Ниже приведены некоторые подробности *стандартов систем автоматизации и протоколов шин*.



Рис. 2. Модуль в конструктиве «Евромеханика» одинарной ширины высотой 3U

Конструктивная совместимость – единообразие механических узлов и габаритов элементов системы: блоков, каркасов, разъёмов и т. п. Конструктивная совместимость достигается при модульном принципе построения механических элементов системы, основой которого является понятие «модуль» и «каркас». Под модулем (module, unit) понимается унифицированная «механическая единица», вставляемая в унифицированный каркас, называемый в русском написании «крейтом» (от англ. crate).

В научном приборостроении в последние годы наибольшее распространение получили унифицированные конструкции, называемые «Евромеха-

ника» [2]. Модуль в конструктиве «Евромеханики» (рис. 2) состоит из печатной платы с электронными компонентами, передней панели с разъёмами ввода/вывода входных/выходных сигналов, а также многоконтактного разъёма для подключения к системной магистрали. Модули бывают одинарной, двойной и т.д. ширины, а также высотой 3U (100 мм по печатной плате), 6U (233 мм), 9U (367 мм), 12U (500 мм).

Крейты «Евромеханики» представляют собой каркас, собираемый из унифицированного алюминиевого проката, в котором подготовлено место для установки магистрали и направляющие для модулей (рис. 3).



Рис. 3. Крейт «Евромеханики» 6 U с нестандартной магистралью, в котором будет собираться силовое оборудование

Электрическая совместимость – согласованность параметров электрических сигналов магистрали: напряжений питания, логических уровней, токов потребления, времён переключения. Электрическая совместимость позволяет устанавливать модули в произвольную позицию на магистрали, за исключением контроллера, который в большинстве систем устанавливается в выделенную позицию.

Логическая (или информационная) совместимость базируется на унификации структуры системной магистрали, набора команд и процедур, а также способов взаимодействия модулей. Это наиболее сложная часть стандарта, определяющая функциональные возможности системы в целом.

Необходимость разработки стандартов для научного приборостроения и построения автоматизированных систем на их базе была осознана достаточно давно. Первым (и простейшим) стандартом для научно-исследова-

тельских систем был опубликованный в 1968 г. **стандарт NIM** (Nuclear Instrumentation Module), ориентированный на построение систем сбора данных в ядерной физике. С сегодняшних позиций это был неполный стандарт, так как в нём специфицировались только электрические и механические параметры. Отсутствие в NIM'e информационно-логических спецификаций объяснялась отсутствием цифровых интегральных схем, которые позволили бы создавать компактные интерфейсные узлы модулей. Поэтому управление режимами, выбор диапазонов, конфигурирование модуля выполнялось вручную. Представление о модуле NIM можно получить из рис. 4.



Рис. 4. Модуль NIM

Вскоре после того, как разработчикам стали доступны цифровые логические схемы, для ядерной электроники был создан законченный стандарт, получивший название **SAMAC** (Computer Automated Measurement And Control). Первая версия появилась в 1969 г., а окончательная – в 1975 г. [3]. Система SAMAC (в русском написании – КАМАК) является типичной магистрально-модульной системой. В ней используется модульный принцип построения, обеспечиваемый унифицированными конструкциями и магистралью для информационных связей между блоками. Основным конструктивным элементом является крейт, в котором могут устанавливаться до 25 функциональных блоков – модулей, подключаемых к общей магистрали с помощью разъёма, расположенного сзади и являющегося частью платы (рис. 5).

Изначально создававшийся для построения ядерно-физических систем сбора данных, SAMAC достаточно быстро распространился как удобный стандарт для научного приборостроения. В 1980 г. SAMAC стал стандартом СССР (ГОСТ 26.201-80) и был принят в качестве базового для научного приборостроения в Академии наук СССР.

Вследствие этого во многих научно-исследовательских институтах страны по-прежнему широко используются средства автоматизации, выполненные в этом стандарте [4]. Впоследствии переработан и сейчас является действующим ГОСТом 27080-93 [5].

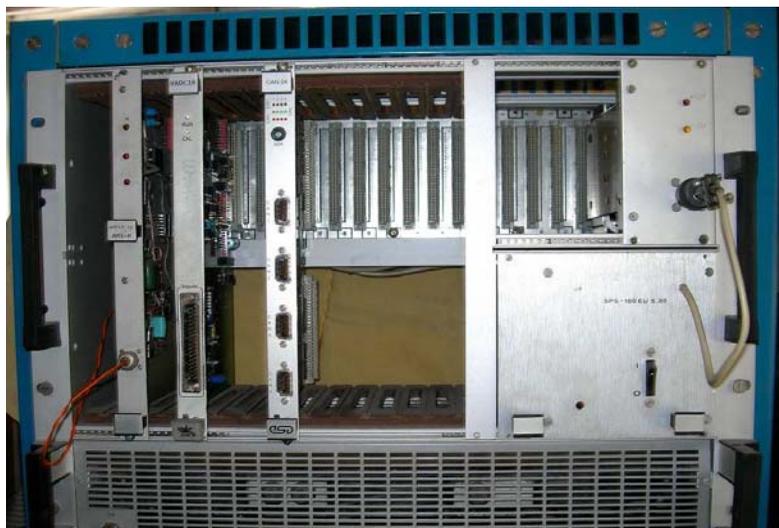


Рис. 5. Крейт КАМАК с модулями

С появлением в конце 1970-х годов достаточно мощных процессоров, разработанных компаниями Intel и Motorola, появилась возможность изготовления магистрально-модульных систем, в которых контроллер содержал бы такой процессор, т. е. был бы интеллектуальным. Подобные возможности открыли бы путь к созданию автономных, не связанных с внешним компьютером, законченных систем, предназначенных для широкого рынка промышленных и военных применений. С этой целью обе компании создали рабочие группы для подготовки стандартов, ориентированных на архитектуры новых процессоров. Специалисты Intel разработали стандарт Multibus-II, а специалисты Motorola – стандарт VME. Оба этих стандарта после ряда доработок стали международными (IEC-821, IEEE 1014-1987, ГОСТ Р МЭК 821-2000), однако в научных организациях стандарт VME применяется гораздо шире. Достаточно часто он применяется в военных системах.

Этот стандарт ориентирован на процессоры семейства Motorola 68000. В простом варианте (только один системный разъем – верхний) магистраль содержит 24 шины адреса и 16 шин данных. При установке второго, нижнего, разъема магистраль может оперировать с 32-разрядными адресом и данными.

В модулях VME используются конструктивы «Евромеханики». VME-крейт и модули показаны на рис. 6. Подробности относительно стандарта VME можно найти в [6].



*Рис. 6.* Комбинированный крейт VME, предназначенный как для модулей 6U, так и для модулей 3U. Конструктивы «Евромеханики» допускают и такой вариант

Появление первых процессоров семейства Pentium инициировало очередной виток разработки стандартов, адекватных возможностям процессора. В результате в 1992 г. появляется первая версия шины PCI (Peripheral Component Interconnect), подготовленная специалистами Intel. Документ стандартизовал конструктивы, напряжения питания и протокол шины для периферийных плат, устанавливаемых внутри компьютера.

Для модульных систем, выполненных в конструктивах «Евромеханики», в 1995 г. была подготовлена версия этой же шины (PCI), использующая другой тип разъёма, но полностью сохранившая протокол и напряжения питания. Реализация шины PCI в конструктивах «Евромеханики» получила название CompactPCI (cPCI). В первую очередь стандарт cPCI предназначался для организации телекоммуникационных систем и автоматизации в энергетике, промышленности, обороне [7].



*Рис. 7.* VME-модуль с 32-разрядной шиной данных

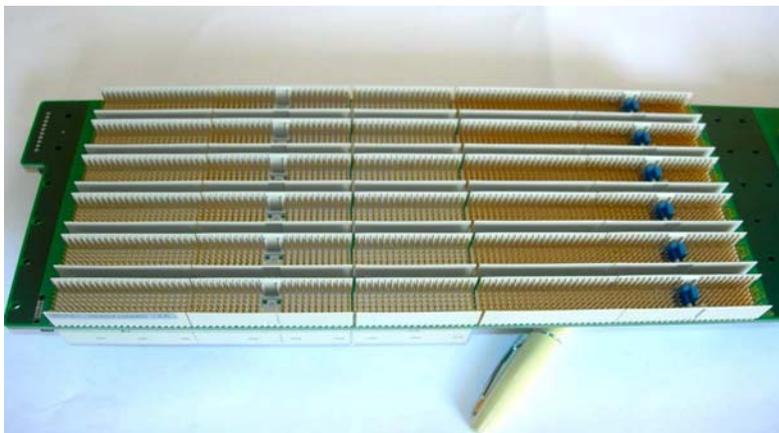


Рис. 8. Фотография магистрали CompactPCI крейта 6U.

Правые два разъёма – системные, остальные три сделаны «на проход»

Магистраль сPCI крейта 6U показана на рис. 8. В отличие от внутри-компьютерной PCI магистрали, модульная версия для установки устройств содержит 8 слотов, из которых один, контроллерный, является выделенным. Два разъёма (J1, J2) являются системными, предназначенными для 64-разрядных систем, а три других (J3, J4 J5), сделанные «на проход» на обратную сторону, предназначены для ввода/вывода различных сигналов. Такое решение позволяет разгрузить переднюю панель от многоконтактных разъёмов.

Использование CompactPCI для построения многофункциональных измерительных и управляющих систем достаточно быстро выявило их некоторую ограниченность, вызванную отсутствием средств для жёсткой, с точностью до единиц наносекунд синхронизации работы отдельных модулей. Такие задачи достаточно распространены в измерительной и управляющей технике. Например, вызывает проблему выполняемый от системного контроллера строго синхронный старт сразу нескольких устройств. Для преодоления этого недостатка компания National Instruments, один из мировых лидеров в области измерительной техники, в 1997 г. предложила новый стандарт – **PXI (PCIeXtensions for Instrumentation)** [8]. Стандарт PXI базируется на сPCI и сохранил все качества последнего. Вместе с тем к стандартной сPCI-магистральной добавлены средства, обеспечивающие межмодульную синхронизацию. Структуру модифицированной магистрали (т. е. PXI-магистрали) можно найти в [9].

Самыми современными (в 2009 г.) являются стандарты **PCIexpress** и **PXIexpress**. Последний применяется в практикуме ТСАНИ и будет представлен в Работе 2.

Выше были кратко описаны наиболее известные *стандарты*, применяемые в научном приборостроении. Стандарты, требуя конструктивной, электрической и логической совместимости элементов, применимы при создании магистрально-модульных систем. Рассмотрим теперь *стандартные шины*, служащие для объединения в систему элементов, имеющих различные напряжения питания и конструктивное исполнение, и обязанные поддерживать лишь стандартный *протокол* используемой шины. Такой способ применяется для организации систем из обычных измерительных приборов, для подключения крупногабаритного оборудования, имеющего соответствующий интерфейс, для построения распределённых систем автоматизации. Иногда стоимость системы, построенной с помощью стандартных шин, оказывается меньше, чем при использовании полных стандартов.

По-видимому, первой параллельной шиной, предназначенной для объединения измерительных приборов в систему, была шина HPIB, созданная в начале 60-х годов в фирме Hewlett-Packard (**HP Interface Bus**). Вскоре предложенный фирмой протокол был доработан, стандартизован и получил название IEEE-488, или GPIB (General Purpose Interface Bus). Шина GPIB является типичным примером параллельной объединительной магистрали, работающей на расстояниях до нескольких метров. Подробности можно найти в [10]. И до настоящего времени эта шина активно применяется для организации локальных автоматизированных систем. В России (в СССР) стандартизована и известна под названием КОП (Канал Общего Пользования).

Наиболее широкое применение среди стандартных находят последовательные шины. Данный тип шин позволяет объединять устройства, отстоящие друг от друга на десятки или сотни метров с помощью линии связи, состоящей всего из нескольких проводов. Хорошо известным всем примером является шина USB. Способам построения систем с помощью последовательных шин посвящена Работа 3.

#### **4. Программное обеспечение систем автоматизации**

Неотъемлемым элементом системы автоматизации являются программное обеспечение. Оно решает 3 задачи:

1. Взаимодействие с аппаратными средствами.
2. Взаимодействие с пользователем.
3. Реализация логики работы, присущей данной конкретной установке.

Для небольших, «настольных» установок все три функции обычно выполняются единой программой, которая и реализует графический интерфейс пользователя, взаимодействует с аппаратурой и содержит некую спе-

цифичную для конкретной установки логику (управление рабочим циклом установки, обратные связи, слежение за различными параметрами, и т. п.).

Существует большое количество готовых программных пакетов для решения подобных задач. Самыми известными из них являются LabVIEW и LabWindows/CVI от National Instruments. Эти пакеты отличаются используемым языком программирования: в LabVIEW применяется визуальное конструирование программы из «кубиков» (язык G), а в CVI используется более традиционный подвид языка СИ. Часто управляющее ПО для небольших установок создаётся на основе пакетов численного анализа, таких как MatLab, поскольку эти пакеты содержат мощный математический аппарат, средства для организации пользовательского интерфейса, а главное – они знакомы многим физикам.

Для больших же установок подход с единой программой, совмещающей в себе все три функции, неприемлем по ряду причин:

1. Большое количество аппаратуры и размеры установки делают невозможным обслуживание всей электроники одним компьютером и одной программой.
2. Объем кода становится столь велик, что совместить всё в одной программе едва ли возможно.
3. Разным программам (управляющим, расчетным, журналирующим) может требоваться доступ к одной и той же электронике, причем одновременно. Также зачастую программе управления одной подсистемой требуются данные от электроники, относящейся к другой подсистеме. Например, системе измерения параметров пучка в ускорителе надо знать настройки магнитной системы, управляющей ускоряющими секциями.

Поэтому на всех крупных установках используются системы управления, построенные по так называемой «трехуровневой архитектуре» (рис. 9).

Самый верхний уровень представлен прикладными программами (в англоязычной литературе он обычно именуется “human interface layer”, “presentation layer” или “supervisory layer”).

Нижний уровень – уровень аппаратуры (“device interface layer”, “field layer”) – состоит из устройств и их драйверов.

И наконец средний уровень (“control layer”) является связующим, обеспечивая передачу данных между прикладными программами и аппаратурой, решая также большое количество служебных задач, таких как мультиплексирование данных, разрешение конфликтов, управление доступом и т. д. Фактически, средний уровень – это «программная шина данных».

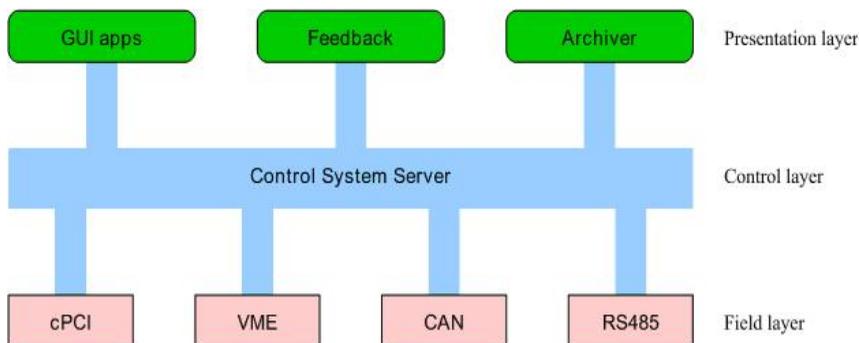


Рис. 9. Трёхуровневая архитектура программного обеспечения крупной экспериментальной установки

Основными достоинствами трёхуровневой архитектуры являются:

- разграничение функций – каждый уровень решает четко определенную задачу;
- распределенность – подобные системы всегда реализуются по модели клиент-сервер (прикладные программы являются клиентами, а средний уровень – сервером); так что к одному серверу (т. е. к одной и той же аппаратуре) может иметь доступ несколько клиентов одновременно, и один клиент может общаться сразу с несколькими серверами (в том числе на разных ЭВМ и даже в разных частях света);
- разрешение конфликтов – при одновременном обращении к аппаратуре нескольких клиентов запрос будет исполнен аппаратурой один раз, и результат его исполнения будет отправлен всем клиентам.

Следствием этих достоинств является масштабируемость – одно из основных требований, предъявляемых к крупным системам управления.

Для разработки систем управления, имеющих трёхуровневую архитектуру, наибольшее распространение в экспериментальной физике получил комплекс программных пакетов EPICS [11], созданный в Аргоннской лаборатории и поддерживаемый сейчас коллаборацией институтов. Родственный ему Vsystem больше используется в коммерческих проектах. Также в научной среде используются системы TANGO [12], TINE [13] и DOOCS [14]. Аналогичная трёхуровневая архитектура используется и в большинстве промышленных средств автоматизации.

## 5. Задачи, структура практикума и оценки

В несколько упрощённом виде работу исследователя можно представить следующим образом: «теория → проектирование и изготовление экспериментальной установки → измерения → анализ данных и новая теория». На физическом факультете читается большое количество курсов, призванных дать студентам теоретическую подготовку, что позволяет при добросовестной учёбе успешно справиться с первым и последним этапами, т. е. с теоретическими обоснованиями и анализом данных. Курс по проектированию экспериментальных установок, а также научные руководители в институтах способствуют преодолению и второго этапа. И лишь несколько лабораторных практикумов учат работе, необходимой на третьем, очень ответственном этапе – организации автоматизированных измерений и получении экспериментальных данных. *Практикум ТСАНИ является одним из них, поэтому относиться к нему нужно со всей серьёзностью.*

Главная задача практикума по Техническим Средствам Автоматизации Научных Исследований – получение студентами теоретических знаний и практических навыков в работе с автоматизированными системами управления и измерения. С этой целью предлагается изучить и получить практические навыки в следующем:

1. Способы построения и структуры автоматизированных систем.
2. Принципы работы системных средств: контроллеров, объединительных магистралей, линий передачи данных.
3. Принципы работы, схемы построения и параметры измерительных и управляющих устройств, применяемых в системах автоматизации.
4. Программирование систем реального времени.

Отметим, что для быстрого выполнения заданий крайне желательно хорошее знание языка СИ, которое должно было быть получено на младших курсах. Необходимым условием успешной работы является подготовка к лабораторным работам дома, включающая изучение описаний, подготовка текстов программ и т. п.

Практикум состоит из двух частей. Первая, «обязательная» часть, включает пять работ, которые должны быть выполнены всеми студентами.

**Работа 0.** «Системы автоматизации экспериментальных установок». Эта работа является введением в практикум и даёт общее представление об автоматизации экспериментов, формулирует и описывает основные понятия, применяемые в данной области.

**Работа 1.** «Изучение среды программирования LabWindows/CVI». В начале работы студентам предлагается вспомнить и продемонстрировать умение писать программы на языке СИ. Далее описывается, как использовать LabWindows/CVI, приводятся примеры и формулируются практические задания. К работе прилагаются справочные материалы по языку СИ и LabWindows/CVI.

**Работа 2.** «Магистрально-модульные системы автоматизации». Описываются общие принципы организации магистрально-модульных систем и вводятся основные понятия. Далее описывается основной комплект оборудования и очень кратко – шина PCIexpress, применяемая в крейтах практикума. Выполняя задания, необходимо научиться работать с основным комплектом оборудования, а также на практике понять, как работают магистрально-модульные системы, самостоятельно управляя с помощью регистра ввода\вывода, имитирующего системный контроллер, одной из простейших параллельных шин, реализованной в выносном стенде. Одно из заданий направлено на получение начальных сведений об аналого-цифровых и цифроаналоговых преобразователях. Кроме того, в работе предполагается дальнейшее освоение LabWindows/CVI.

**Работа 3.** «Аналого-цифровые и цифро-аналоговые преобразователи». Объясняются принципы построения аналого-цифровых и цифро-аналоговых преобразователей. Рассматриваются основные характеристики и погрешности преобразования аналог → код и код → аналог. Практические задания предоставляют возможность изучить свойства реальных АЦП и ЦАПов, измерить их характеристики преобразования и определить погрешности.

Вторая часть включает несколько работ, каждая из которых представляет собой небольшой экспериментальный стенд и простейшую систему автоматизации этого стенда. Выполняя «экспериментальные работы», необходимо понять возможности и особенности применяемой аппаратуры и автоматизировать предлагаемый эксперимент.

В большинстве работ присутствуют задания, отмеченные звёздочкой. Эти задания не являются обязательными и ориентированы на студентов, желающих получить углублённые представления по изучаемому вопросу.

**Работа 4.** «Распределённые системы управления и последовательные шины передачи данных». Данная работа служит для ознакомления с последовательными шинами передачи данных, реализацией протокола шины программным путём, дающей представление о принципах работы контроллера шины и интерфейсов. Кроме изучения последовательных шин, выполняя практические задания, можно получить навыки в построении распределённых систем измерения температурных полей.

## Оценка за практикум

Зачёт по практикуму ТСАНИ является дифференцированным. Студенты, выполнившие и сдавшие преподавателю работы 0, 1, 2, 3, 4, могут рассчитывать на оценку *«удовлетворительно»*. Выполнившие и сдавшие 5 основных + 1 «экспериментальную» заслуживают оценки *«хорошо»*. Оценка *«отлично»* может быть поставлена за 5 + 2 + \*), т. е. за пять основных + 2 «экспериментальных», в одной из которых должно быть выполнено задание с \*). *Следует отметить, что приведённые выше требования носят рекомендательный характер, и окончательное решение об оценке принимает преподаватель.*

Предполагаемый график выполнения работ «на отлично»:

Занятие 1: ТБ + Работа 0 + начало Работы 1.

Занятие 2: Работа 1.

Занятие 3: Работа 1 + начало Работы 2.

Занятия 4, 5: Работа 2.

Занятия 6, 7: Работа 3.

Занятия 8, 9: Работа 4.

Занятия 10, 11, 12, 13, 14, 15: Три «экспериментальных» работы.

Нетрудно увидеть, что выполнить необходимый объём возможно лишь при посещении всех занятий в семестре.

## 6. Контрольные вопросы

Подготовьте в письменном виде (как File.doc) короткие ответы на следующие вопросы и покажите преподавателю:

1. Сколько и какие работы необходимо сделать для получения за практикум оценки «хорошо»?
2. Какие элементы образуют системы автоматизации экспериментальных установок? Каково назначение этих элементов?
3. Аналого-цифровой преобразователь, имеющий шкалу 1000 мВ, выдаёт 8-разрядный двоичный код. Какова, по-вашему, разрешающая способность этого прибора? Сколько разрядов десятичного числа необходимо для представления отсчётов преобразователя?
4. Что такое магистрально-модульные системы? Что входит в понятие «Стандарт магистрально-модульной системы»?
5. Какие задачи решает программное обеспечение систем автоматизации?

6. Переведите число в двоичном формате 11000101 в десятичный, а затем полученное десятичное число переведите в шестнадцатеричное. Покажите преподавателю, как вы это сделали.
7. Переведите число, предложенное преподавателем, в двоичный и шестнадцатеричный форматы.
8. Каков результат следующих побитовых операций: FF&11, F0|0B?

**Внимание! Если вы не ответили на вопросы 6, 7, 8 за 5 минут, то вам необходимо изучить «Приложение» к данной работе и выполнить приведённые там упражнения.**

## 7. Список литературы

1. Мячев А. А. Интерфейсы средств вычислительной техники. Энциклопедический справочник. М.: Радио и связь, 1993.
2. Standard for Mechanical Core Specifications for Microcomputers. IEEE Std. 1101.2-1992.
3. Report EUR 4100e, IEEE Standard 583-1975.
4. <http://web.inp.nsk.su/activity/automation/device/devicenew/camac/devcam.html>
5. <http://www.dbases.ru/cgi-bin/catalog/catalog.cgi?i=9758>
6. [http://www.interfacebus.com/Design\\_Connector\\_VME.html](http://www.interfacebus.com/Design_Connector_VME.html)
7. <http://en.wikipedia.org/wiki/CompactPCI>
8. <http://en.wikipedia.org/wiki/PXI>
9. <http://zone.ni.com/devzone/cda/tut/p/id/4811>
10. [http://www.itt-ltd.com/reference/ref\\_ieee488.html](http://www.itt-ltd.com/reference/ref_ieee488.html)
11. <http://www.aps.anl.gov/epics/>
12. <http://www.esrf.eu/computing/cs/tango/>
13. <http://adweb.desy.de/mst/tine/>
14. <http://doocs.desy.de/>

## Биты, байты, системы счисления, двоичные операции

Студенты третьего курса физического факультета, проходящие практикум ТСАНИ, демонстрируют поразительную неосведомлённость в казалось бы несложных вопросах, касающихся способов кодирования чисел в ЭВМ, систем счисления, двоичных операций и т. д. По-видимому, это связано с тем, что курс информатики в школах и в университете обходит стороной или недостаточно подробно освещает эти вопросы, однако для успешного прохождения практикума их знание необходимо. Простейшим выходом из сложившейся ситуации видится это краткое введение.

### Двоичное представление чисел и основные логические операции

Информация в ЭВМ представляется в двоичном виде, и электронные устройства компьютера «понимают» лишь два значения сигналов, которые обозначаются «0» и «1». За «0» обычно принимается сигнал, меньший определённого уровня, а за «1» — больший (возможно, уже несколько другого) уровня. К примеру, в ТТЛ (транзисторно-транзисторной логике) за «0» принимается сигнал от 0 до 0,8 В, а за «1» — сигнал от 2 В до 5 В. Такие уровни позволяют обеспечить устойчивую работу при наличии шумов. Каждая такая единица информации называется «битом».

Работа ЭВМ основана на логических операциях с двоичными сигналами. Обычно сигнал «0» называют «ложью», а сигнал «1» — «истиной». Основными логическими операциями являются: **НЕ**, **И**, **ИЛИ**, **Исключающее ИЛИ**. Рассмотрим каждую из них.

Операция **НЕ** (*not*) является операцией от одного операнда и инвертирует его, т.е.  $not(1) = 0$ ,  $not(0) = 1$ . Выражение  $Y = not(A)$  можно представить следующей таблицей истинности:

<i>A</i>	$Y=not A$
0	1
1	0

Операция **И** (*and* — логическое умножение) является операцией от двух операндов — *A* и *B*. Для операции **and** ( $Y = A \text{ and } B$ ) справедлива следующая таблица.

<i>A</i>	<i>B</i>	$Y=A \text{ and } B$
0	0	0
0	1	0
1	0	0
1	1	1

Операция **ИЛИ** (*or* – логическое сложение) – также операция от двух операндов.  $A \text{ or } B$  истинно, когда или  $A$ , или  $B$  истинны. По-другому,  $A \text{ or } B$  ложно, лишь когда и  $A$  и  $B$  ложны. Это представляется следующей таблицей:

$A$	$B$	$Y=A \text{ or } B$
0	0	0
0	1	1
1	0	1
1	1	1

Можно заметить, что:

$$(\text{not } A) \text{ and } (\text{not } B) = \text{not } (A \text{ or } B), \text{ а также}$$

$$(\text{not } A) \text{ or } (\text{not } B) = \text{not } (A \text{ and } B),$$

эти правила называются законами де Моргана.

И последняя операция, которая нам понадобится, – это операция **«исключающего ИЛИ»** (*xor*).  $A \text{ xor } B$  истинно тогда, когда  $A$  и  $B$  отличны друг от друга.

$A$	$B$	$Y=A \text{ xor } B$
0	0	0
0	1	1
1	0	1
1	1	0

### Упражнения

1. Приведите результат следующих логических операций:

$$1 \text{ or } (1 \text{ and } \text{not}(1)) = ?$$

$$0 \text{ xor } \text{not}(1) = ?$$

$$(1 \text{ and } \text{not}(0)) \text{ xor } \text{not}(1) = ?$$

2. Каким образом с помощью логических операций можно реализовать сложение с переносом двух однобитных чисел? Как определить результат сложения и результат переноса?

### Системы счисления

Разберёмся, каким образом числа можно представить в двоичном коде. Рассмотрим привычный для нас способ записывать десятичные числа. Возьмём для примера число 932 (состоящее из трёх десятичных разрядов). На самом деле, записывая 932, мы имеем в виду 9 сотен, 3 десятка и 2 единицы, т. е.

$$923_{10} = 9 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0.$$

Основанием такой системы счисления является 10. Для записи мы применяем десять знаков: «0», «1», «2»...«8», «9».

ЭВМ может оперировать всего двумя знаками: «0» и «1». В таком представлении число выглядит как строка из нулей и единиц. Возьмём, например, число 10111 (пятиразрядное число). Для того, чтобы перевести его в привычную нам десятичную систему, достаточно провести аналогичную описанной ранее операцию:

$$10111_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 23.$$

В общем случае в системе счисления с основанием  $n$  применяется  $n$  знаков и  $k$  разрядные числа записываются:

$$a_k \cdot n^k + a_{k-1} \cdot n^{k-1} + \dots + a_0 \cdot n^0.$$

Числа в двоичной системе счисления однако весьма неудобны, так как состоят из большого числа разрядов. Даже такое небольшое в десятичной записи число, как  $1586_{10}$ , превратится в двоичное  $11000110010_2$ . Однако переводить каждый раз из двоичной в десятичную запись и обратно проблематично для человека.

HEX	Десятичное	Двоичное
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Для уменьшения числа разрядов и вместе с тем достаточно простого перевода применяются системы счисления с основанием, являющимся степенью двойки: восьмеричная и шестнадцатеричная. Рассмотрим только шестнадцатеричную систему счисления (Hexadecimal, или просто HEX). Для записи чисел в шестнадцатеричной системе потребуется 16 знаков, и десяти цифр уже не хватает. Поэтому применяются цифры от 0 до 9 и буквы A, B, C, D, E, F.

Для перевода двоичного числа в шестнадцатеричное достаточно разделить первое на группы по четыре бита, и каждую группу записать одним знаком в шестнадцатеричной системе:

$$11\ 0000\ 0110\ 1101_2 = 302D_{16}.$$

Несложно произвести и обратный перевод:

$$ABCD_{16} = 1010101111001101_2.$$

### Упражнения

1. Представьте, не пользуясь калькулятором, в двоичном виде следующие шестнадцатеричные числа: **12, FF, AF, DE, 10**.
2. Представьте, не пользуясь калькулятором, в шестнадцатеричном виде следующие двоичные числа: **1010, 10001, 110011, 11100011, 11111, 11**.
3. Представьте, не пользуясь калькулятором в десятичном виде следующие шестнадцатеричные числа: **100, 10, FF, AA, 1A, 20**.

### Представление чисел в ЭВМ

В вычислительной технике часто применяется такое понятие, как байт. Байт — это 8-битовое двоичное число, соответственно одним байтом можно представить 256 различных чисел (комбинаций): от 0 до 255. В байте биты нумеруются справа налево от нуля до 7, соответственно номеру разряда. Также существует понятие «машинное слово». Размер машинного слова зависит от конкретной машины, а иногда и языка программирования и может равняться байту, двум байтам, четырём байтам и даже восьми байтам.

Двоичные числа легко умножать и делить на  $2^n$ . Делается это так же, как и в случае умножения/деления на  $10^n$  десятичных чисел. Например, чтобы умножить на 2 число  $1101_2$  ( $13_{10}$ ), достаточно приписать справа 0 и получить  $11010_2$  ( $26_{10}$ ). А для того, чтобы поделить на 4 число  $1100_2$  ( $12_{10}$ ), достаточно стереть два крайних правых знака и получить  $11_2$  ( $3_{10}$ ). Умножение / деление на  $2^n$  на языке логических операций называются сдвигом влево / вправо на  $n$ . Обозначим сдвиг числа  $x$  влево на  $n$  как  $x \ll n$ , а сдвиг вправо соответственно  $x \gg n$ . При этом все лишние биты (вышедшие за границу байта) выкидываются, а недостающие обнуляются. Например:

$$1100\ 1011 \ll 1 = 1001\ 0110$$

$$1100\ 1001 \gg 2 = 0011\ 0010$$

Теперь рассмотрим операции «побитовое И» (&), «побитовое ИЛИ» (|) и «побитовое **исключающее или**» (^). Они выполняются по очереди для

всех битов, и результирующий байт состоит из битов, являющихся результатом операции:

$$1110\ 0000 \ \& \ 1010\ 1100 = 1010\ 0000$$

$$1100\ 1010 \ | \ 1101\ 0101 = 1101\ 1111$$

$$1010\ 1100 \ \wedge \ 0111\ 0101 = 1101\ 1001$$

Легко заметить, что побитовое **&** удобно применять для нуления некоторых битов в байте:  $11110011 \& 10011111 = 10010011$ . Побитовое **|** удобно для выставления некоторых битов в единицу. Операцию «исключающее или» можно использовать для инверсии всех битов, если в качестве одного операнда использовать  $11111111$ :  $A \wedge 1111\ 1111 = \bar{A}$ .

0111 1111	127
0111 1110	126
	...
0000 0001	1
0000 0000	0
1111 1111	-1
1111 1110	-2
	...
1000 0001	-127
1000 0000	-128

Отрицательные числа в ЭВМ представляются в двоичном дополнительном коде (см. таблицу). Обратите внимание, что 7-й разряд (восьмой по счёту) указывает на знак числа. Для получения двоичного дополнения к числу (его отрицания) достаточно инвертировать все биты и добавить единицу.

### Упражнения

1. С помощью какой операции можно в 16-битном слове занулить старшие 8 бит (биты 8-15)?
2. Каков результат следующих побитовых операций:  $FF \ \& \ 11$ ,  $F0 \ | \ 0B$ ,  $10C \ \wedge \ FFF$ ?
3. Найдите двоичное дополнение к следующим числам:  $10001000$ ,  $11110000$ ,  $00011000$ ,  $01011010$ .

Учебное издание

**Батраков Александр Матвеевич**

## **Системы автоматизации экспериментальных установок**

Методические указания к вводной лабораторной работе

Подписано в печать 02.09.2015 г.  
Формат 60×84 1/16. Уч.-изд. л. 1,4.  
Уч.-изд. л. 1,5. Тираж 30 экз.  
Заказ № 230

Редакционно-издательский центр НГУ  
630090, Новосибирск-90, ул. Пирогова, 2