

Лекция 5: “Функциональные компоненты СУ: преобразующее устройство”

Гончаров Олег Игоревич

Факультет вычислительной математики и кибернетики,
Московский государственный университет имени М.В. Ломоносова

2013

Физическая реализация УПУ

Усилительно-преобразующее устройство (или преобразующее устройство, корректирующее устройство, преобразователь информации) — элемент САУ, основной функцией которого является преобразование входных сигналов в управление по заданному алгоритму (закону управления).

Часто этот элемент САУ называют **контроллером** от *control*.

В основе могут лежать различные физические принципы:

- Механические регуляторы: вертушка Уатта, поплавковый регулятор и т.п.
- Пневматика и гидравлика.
- Электрические (электронные, электромагнитные, электромашинные).
- Цифровые системы (род электронных) — обработка информации происходит в цифровом виде.

Преобразование аналоговых и квантованных сигналов.

- **Пропорциональные звенья:** на основе операционных усилителей (ОУ).
- **Сумматоры:** на основе суммирующих и дифференциальных усилителей (строятся на основе ОУ).
- **Динамические звенья:** четырехполюсники (пассивные схемы на основе конденсаторов, резисторов и индуктивностей).
- **Обработка квантованных сигналов:** логические элементы, интегральные схемы (ИС), программируемые логические ИС.

Цифровые преобразующие устройства

Обработка цифровых сигналов (квантование по уровню и по времени).

История развития: специализированные устройства с жестким алгоритмом, программируемые логические контроллеры, компьютеры.

Элементарная база:

- Специализированные интегральные схемы (ИС).
- Программируемые логические интегральные схемы.
- ЭВМ (исполнимая программа)
 - ▶ Процессоры (микропроцессоры).
 - ▶ Микроконтроллеры (процессор, ОЗУ, ПЗУ и периферия на одном кристалле).
 - ▶ Сигнальные процессоры – DSP (специализированные процессоры, предназначенные для обработки сигналов в реальном времени).

- Работа в реальном времени: гарантированная длительность цикла выполнения операций, аппаратная защита от “зависаний”.
- Открытость систем: программная и физическая совместимость, заменимость компонент, масштабируемость.
- Надежность: наработка на отказ, дублирование систем.
- Защита интерфейсов в/в.
- Защита от внешних воздействий: пыль, вибрация, влага, отвод тепла.
- Конструктивные особенности: крепления.
- Надежность источника питания.

Архитектура

Обычно **гарвардская архитектура** — отдельная память программ и данных.

- ЦПУ (разрядность, частота, набор инструкций)
 - ▶ интерфейсы для отладки и записи программы (JTAG)
 - ▶ часы реального времени
 - ▶ сторожевой таймер
- Память
 - ▶ ОЗУ (обычно для данных),
 - ▶ энергонезависимая память с программами.
- Интерфейсы
 - ▶ сопряжение с другими устройствами (RS-485, CAN и т.п.).
 - ▶ подключение средств для взаимодействия с человеком (дисплей, мышь, клавиатура)
- Модули ввода/вывода
 - ▶ дискретные входы и выходы (логические сигналы)
 - ▶ аналоговый входы и выходы: АЦП и ЦАП
 - ▶ ввод частоты, времени события: счетчики и таймеры
 - ▶ ШИМ выходы
 - ▶ аппаратная реализация регуляторов: ПИД

- Промышленные контроллеры:
 - ▶ Одномодульные: единое устройство.
 - ▶ Многомодульные: несколько заменяемых модулей с общей шиной (процессорный модуль, DSP, интерфейсы связи, средства ввода-вывода)
 - ▶ Распределенные: несколько устройств, связанных промышленной сетью образуют единую систему.
- Промышленные компьютеры (архитектура ПЭВМ).

В случае систем на одном кристалле (микроконтроллеры), управляющие регистры периферии отображаются в память и/или соответствуют портам в/в.

В более сложных системах для объединения устройств используются шины (PCI, PC/104, CompactPCI), отдельные модули могут иметь свои процессоры, способ доступа зависит от конкретной реализации: порты в/в, протоколы (например ModBus, I2C), DMA.

Языки программирования и средства отладки:

- языки МЭК 61131-3 (восходят к старым ПЛК, 5 стандартных форм описания)
- стандартные языки программирования (C, C++)

При программировании обычно используется кросскомпиляция: разработка, компилирование, отладка ведется с помощью ПЭВМ, подключаемого к контроллеру через специальный интерфейс (JTAG, ISP).

Библиотеки:

- Стандартные библиотеки (libc, libc++ и т.п.)
- Библиотеки аппаратного уровня
- Операционные системы (в т.ч. и **реального времени**)

ОС реального времени

Основное отличие: **обеспечивают детерминированное время реакции на событие (прерывание) и детерминированное время выполнения задач**

Особенности ОС РВ:

- предсказуемость работы планировщика, отводит на выполнение задач детерминированное время,
- приоритеты задач,
- наследование приоритетов (исключает ситуацию инверсии приоритетов),
- возможность вытеснения задач ядром ОС,
- ограниченная латентность прерываний (в т.ч. запуск задач по таймеру) и их приоритеты,

Обычно нет виртуальной памяти, динамического загрузчика, список задач известен заранее.

Предоставляются стандартные средства взаимодействия процессов: критические секции, сигналы, сообщения и т.п.