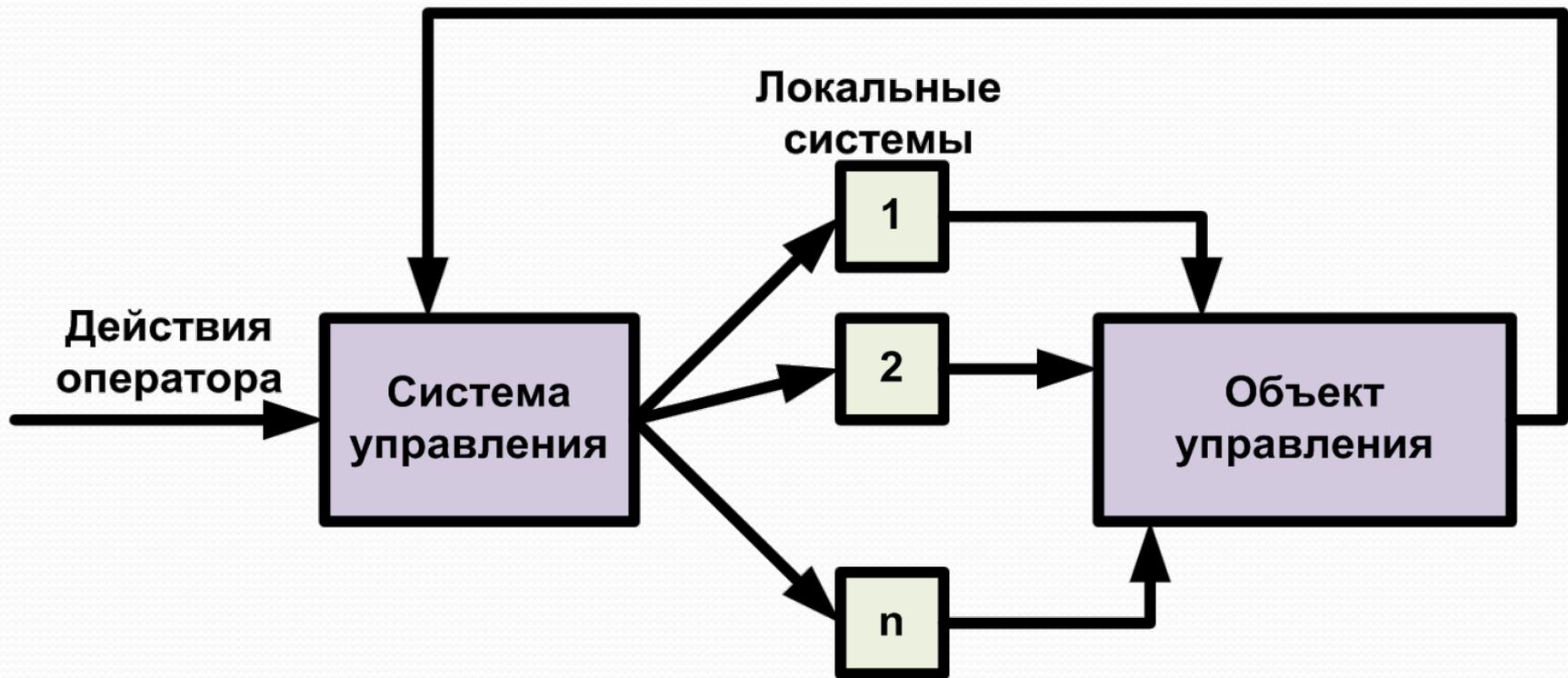


Метод создания программного обеспечения модулей, выполненных на основе программируемых логических интегральных схем

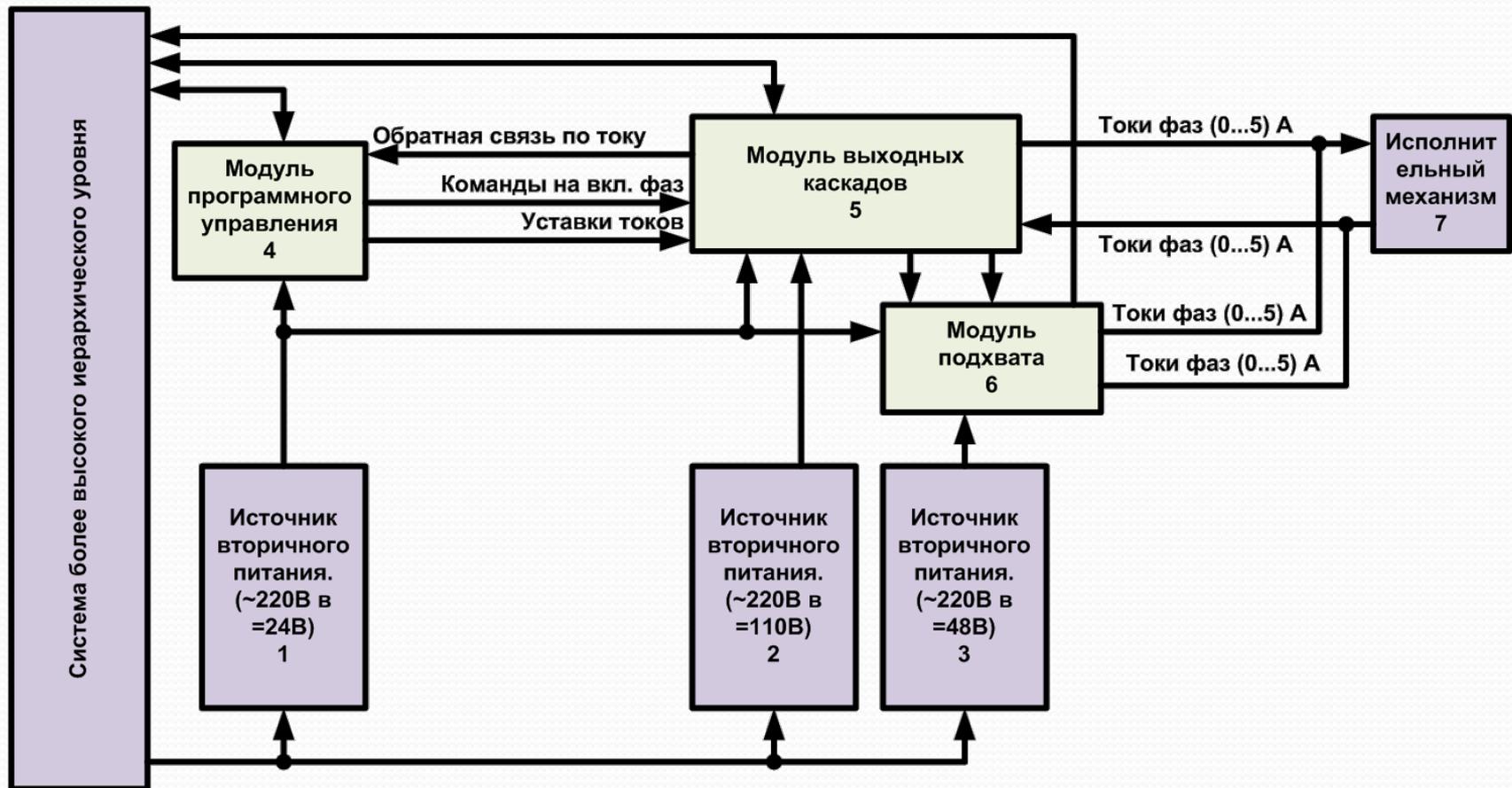
Юрий Юрьевич Янкин
Анатолий Абрамович Шальто

2012

Структура автоматизированной системы управления техническими средствами



Структура локальной системы регулирования положения исполнительного механизма



Опыт создания рассматриваемых модулей управления исполнительным механизмом

- Аппаратная реализация на жесткой логике:
 - Любое изменение алгоритма требует доработки печатных плат и монтажа модуля
 - Сложность разработки схемы по словесному описанию алгоритма работы
 - Сложность отладки
 - Отсутствие возможности формальной верификации
 - Сложность понимания Заказчиком
 - Габаритные размеры
 - Высокое энергопотребление

Опыт создания

- Программная реализация:
 - Программирование микроконтроллера С167 на языке С
 - Привлечение специалиста по программированию или освоение разработчиком модуля дополнительной специальности
 - Сложность разработки программы по словесному описанию алгоритма работы
 - Сложность понимания Заказчиком
 - Сложность отладки
 - Отсутствует возможность верификации

Путь решения

- Существуют средства, позволяющие совместить аппаратную реализацию с программируемостью: программируемые логические интегральные схемы (ПЛИС)

Традиционный подход к программированию

- Создание схемы из стандартных (библиотечных) функциональных блоков
- Описание на специализированных языках (VHDL, Verilog)

Предлагаемый подход

- Появились средства проектирования, не требующие привлечения программистов к программированию ПЛИС



Stateflow®

ALTERA



Цели работы

- Исключение трудоемкого процесса ручного программирования
- Привлечение специалистов по автоматике и силовым полупроводниковым преобразователям к разработке программного обеспечения
- Наглядность, простота восприятия, документируемость, модифицируемость программы
- Повышение степени автоматизации процесса разработки ПО
- Повышение качества ПО, в частности, за счет возможности верификации

Идея

- Распространить автоматное программирование на программирование ПЛИС
- Представить программное обеспечение модуля в виде системы автоматизированных объектов управления¹
- Разработана технология программирования ПЛИС, реализованная в модулях:

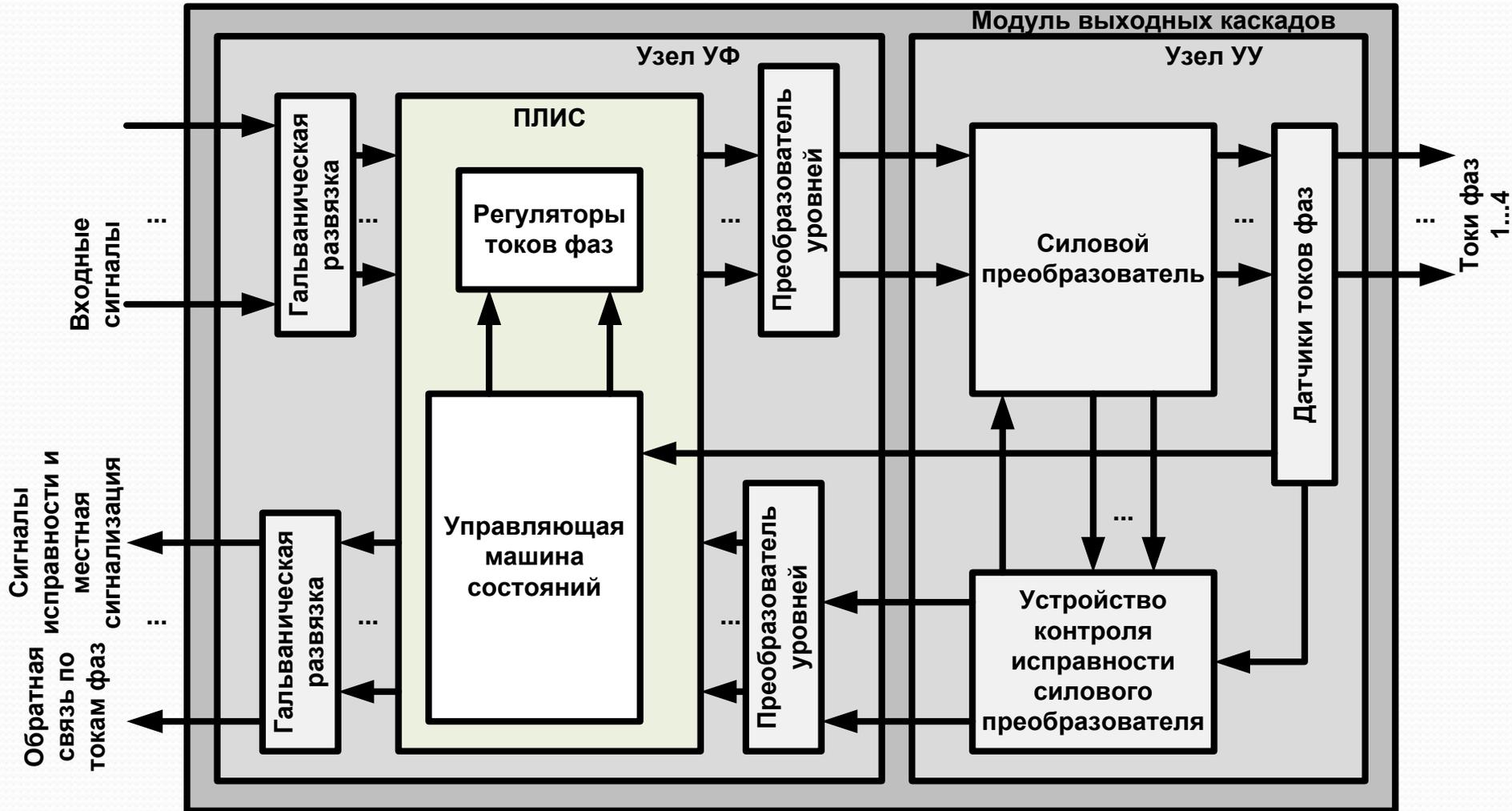
Название	Назначение
Модуль программного управления	Ввод дискретных сигналов от системы более высокого иерархического уровня, формирование уставок токов фаз шагового двигателя, контроль наличия токов фаз.
Модуль выходных каскадов	Формирование токов фаз по уставкам, поступающим из модуля программного управления.
Модуль подхвата	Удержание шагового двигателя в достигнутом положении при неисправности модуля выходных каскадов.

1. Поликарпова Н. И., Шалыто А. А. Автоматное программирование. СПб.: Питер, 2010. – 176 с.

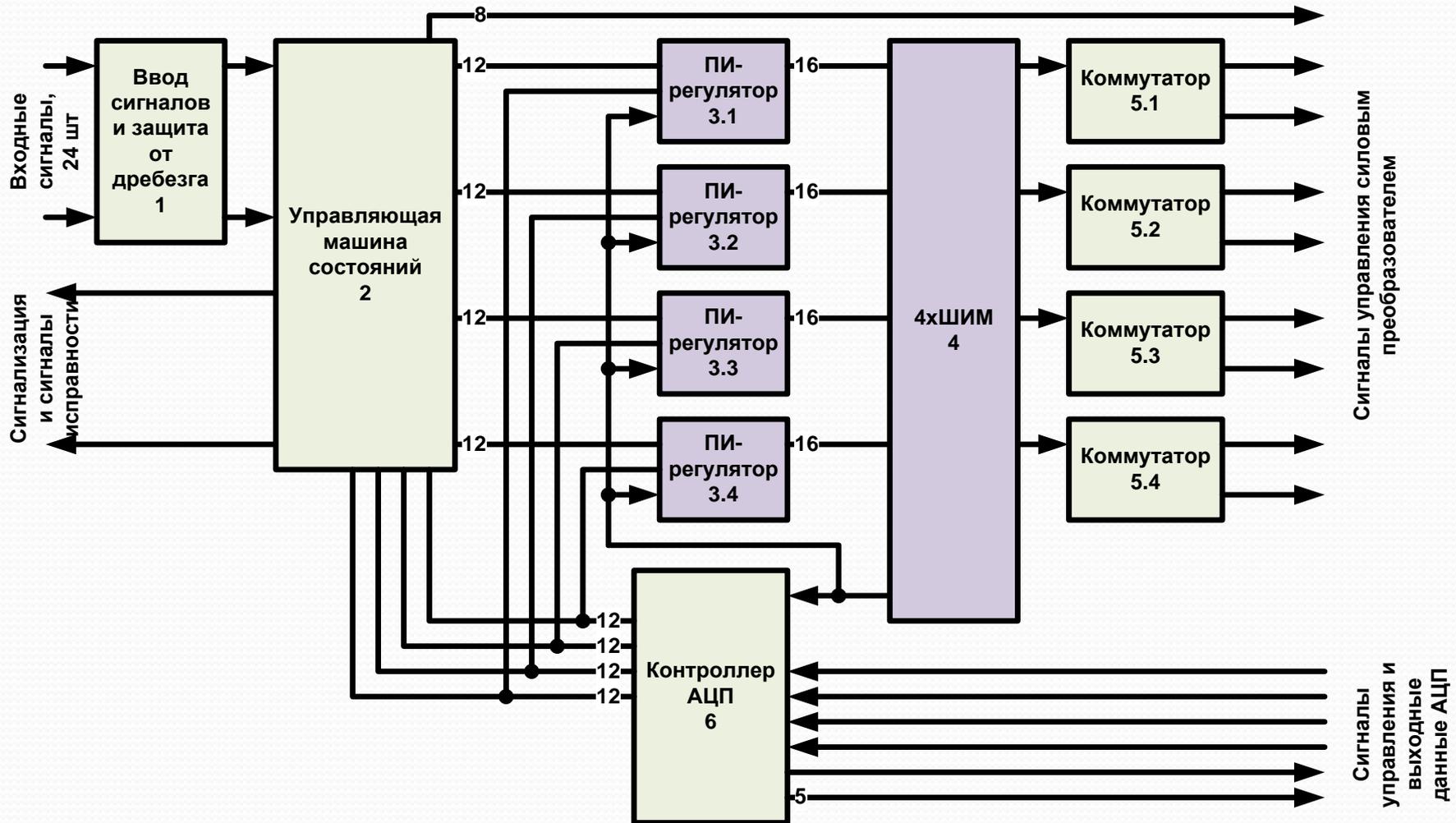
Требования к модулю выходных каскадов, который рассматривается в качестве примера

- выполнение алгоритмов работы системы при управлении шаговым электродвигателем
- возможность ввода 24 дискретных сигналов
- возможность вывода 20 дискретных сигналов
- поддержание тока в четырех обмотках шагового двигателя по уставкам, поступающим из модуля программного управления
- формирование четырех токов заданной формы
- защиту от ЭДС самоиндукции, короткого замыкания в нагрузке
- контроль обрыва нагрузки, величины напряжения питания

Структурная схема модуля выходных каскадов



Структура программного обеспечения модуля выходных каскадов



Функциональные блоки программы

Все функциональные блоки программы реализуются с помощью программного комплекса MATLAB. Создание программы разделяется на два этапа:

- Создание и моделирование управляющей машины состояний
- Создание и моделирование контура регулирования тока

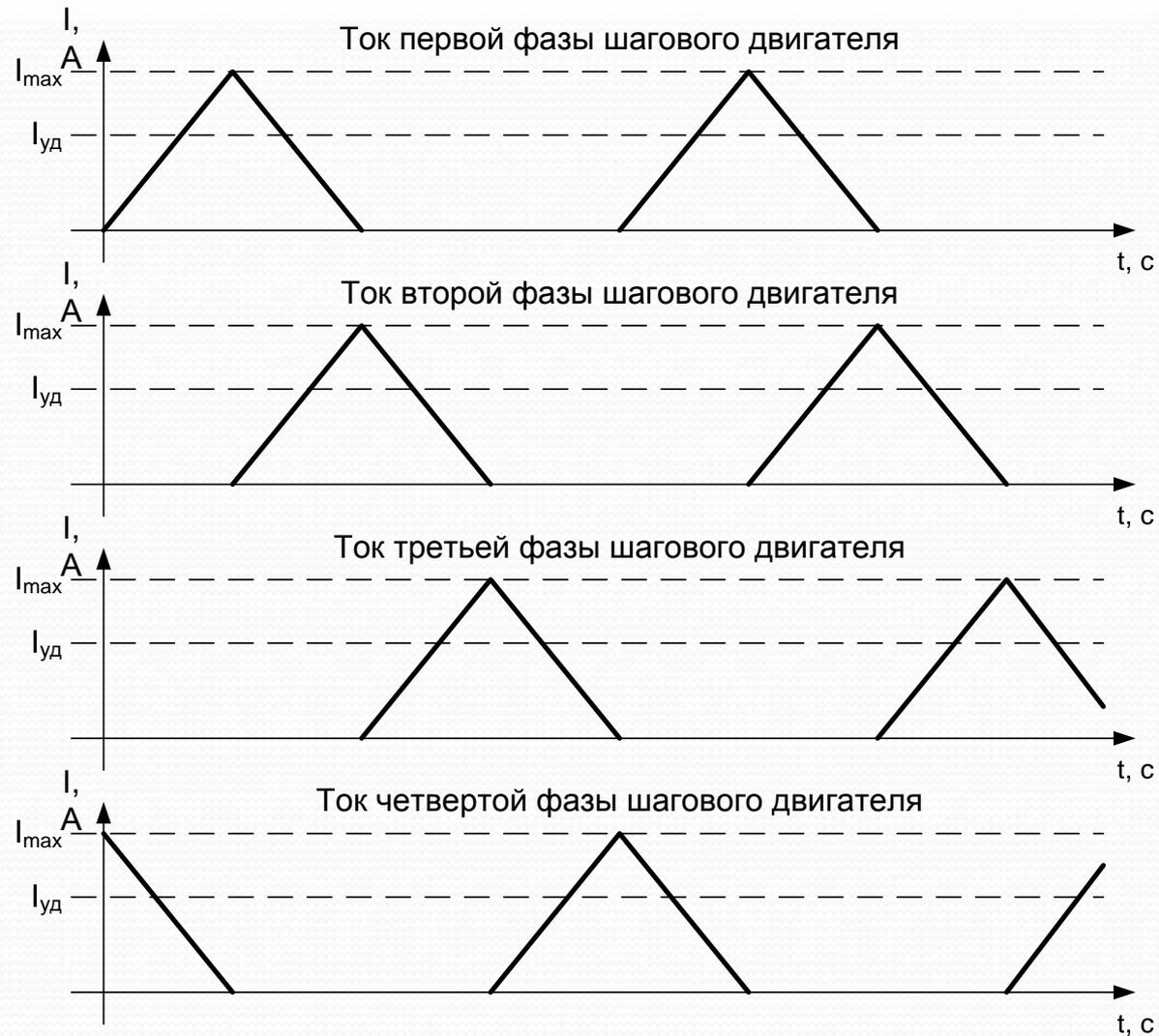
Номер	Наименование	Содержит автоматы?
1	Ввод сигналов и защита от дребезга	Да
2	Машина состояний (управление и контроль исправности)	Да
3	ПИ-регулятор тока	Нет
4	Широтно-импульсный модулятор (ШИМ)	Нет
5	Коммутатор	Да
6	Ввод сигналов датчиков токов	Да

Технология создания блоков, содержащих автоматы

1. Разработка перечня и описания входных и выходных переменных

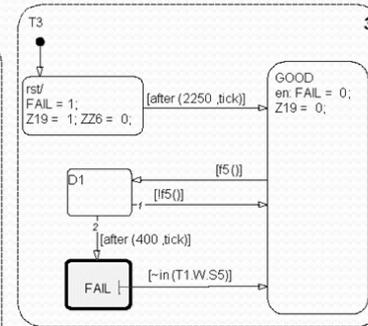
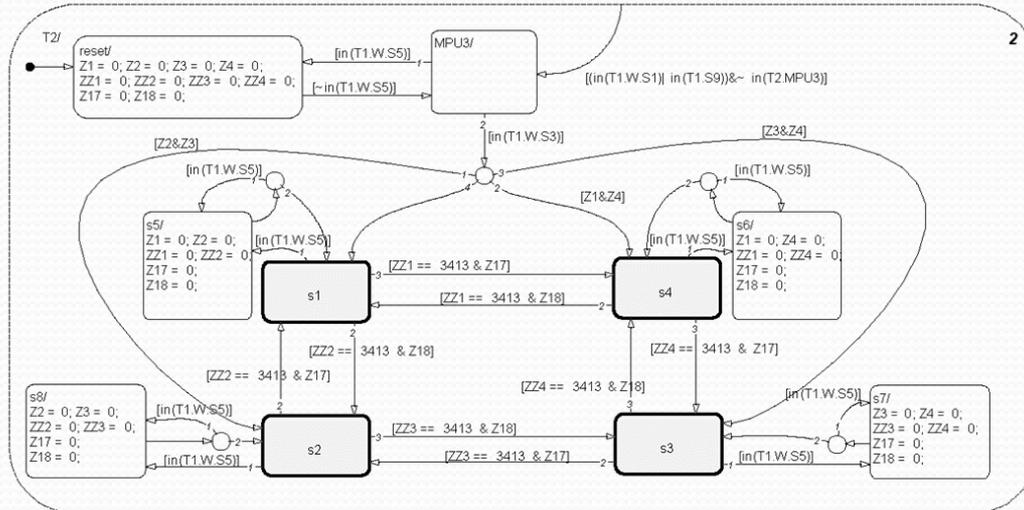
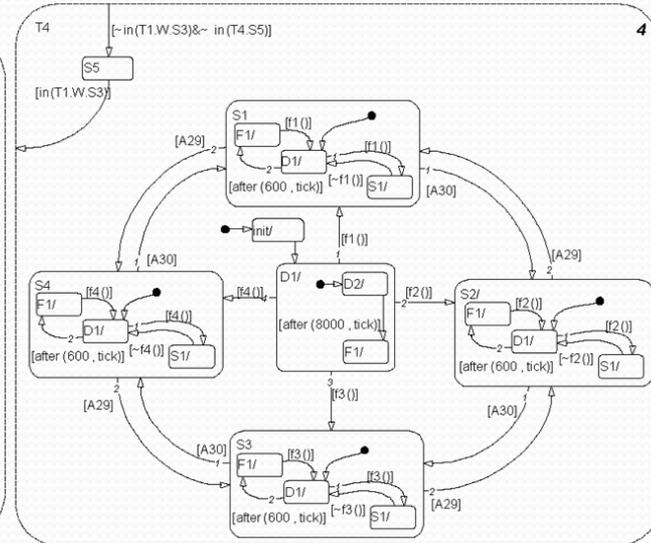
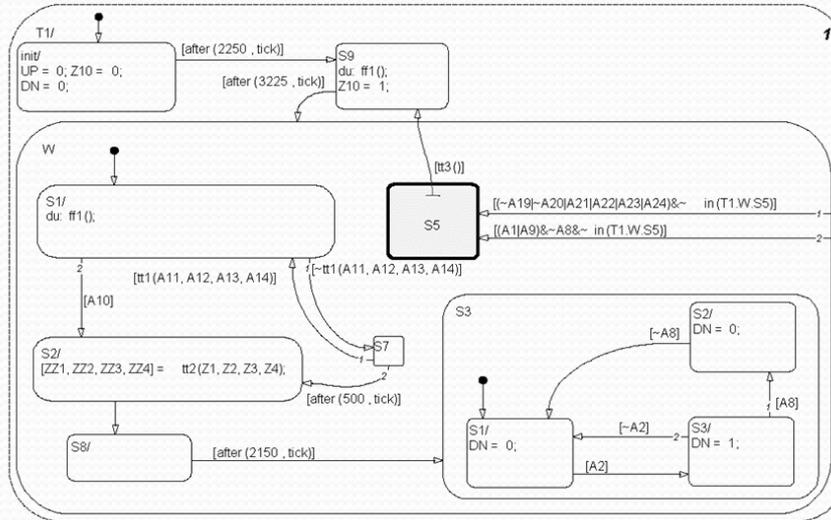
Имя	Направление	Тип данных	Описание
X ₁	Вход	boolean	Дистанционное управление
X ₂	Вход	boolean	Движение вниз
X _n	Вход
Z ₁	Выход	boolean	Включение фазы 1
Z ₂	Выход	uint16	Уставка тока фазы 1
Z _n	Выход

2. Получение алгоритма работы от Заказчика (в виде словесного описания и временных диаграмм)

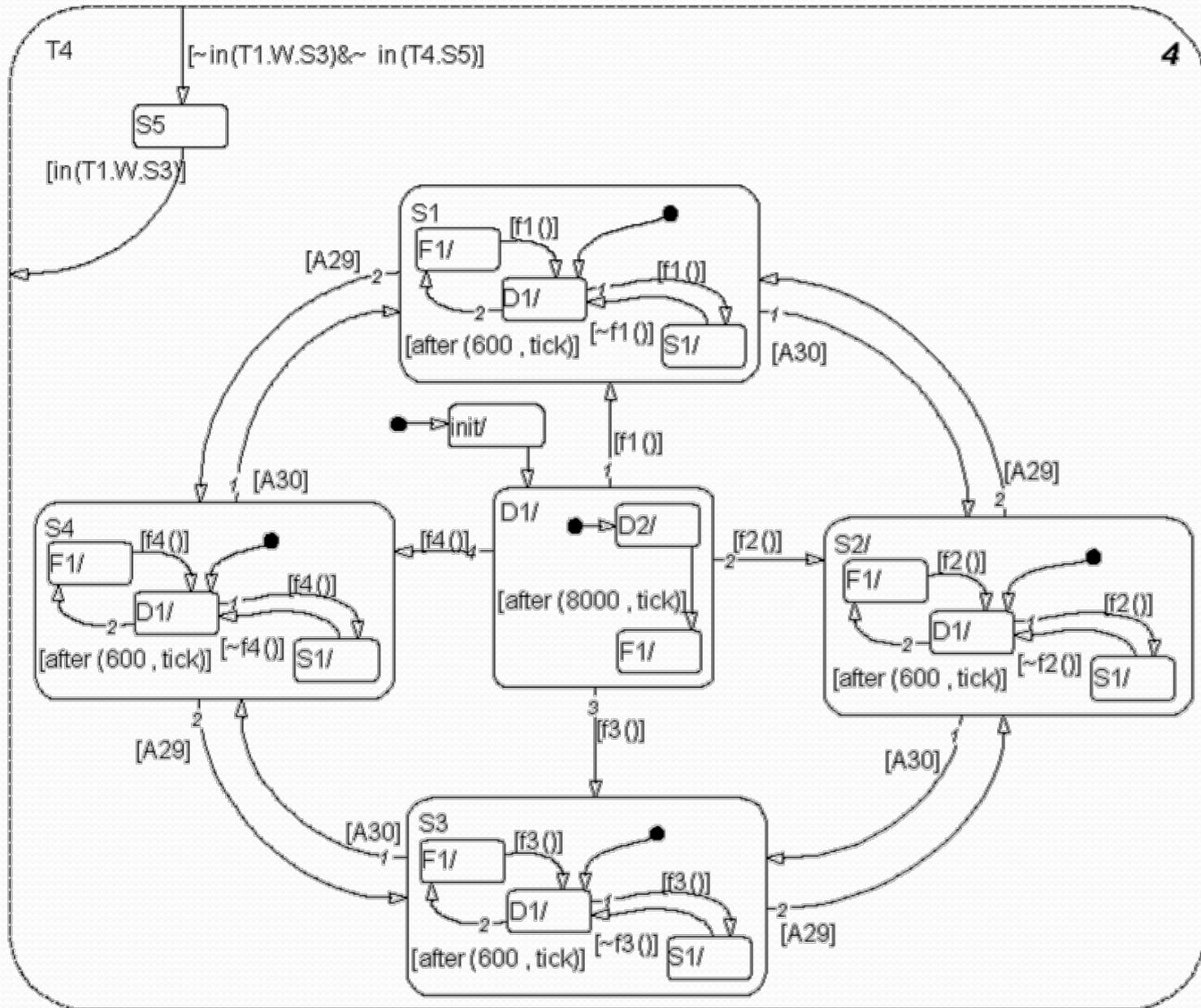


3. Проектирование системы графов переходов конечных автоматов с использованием *MATLAB*

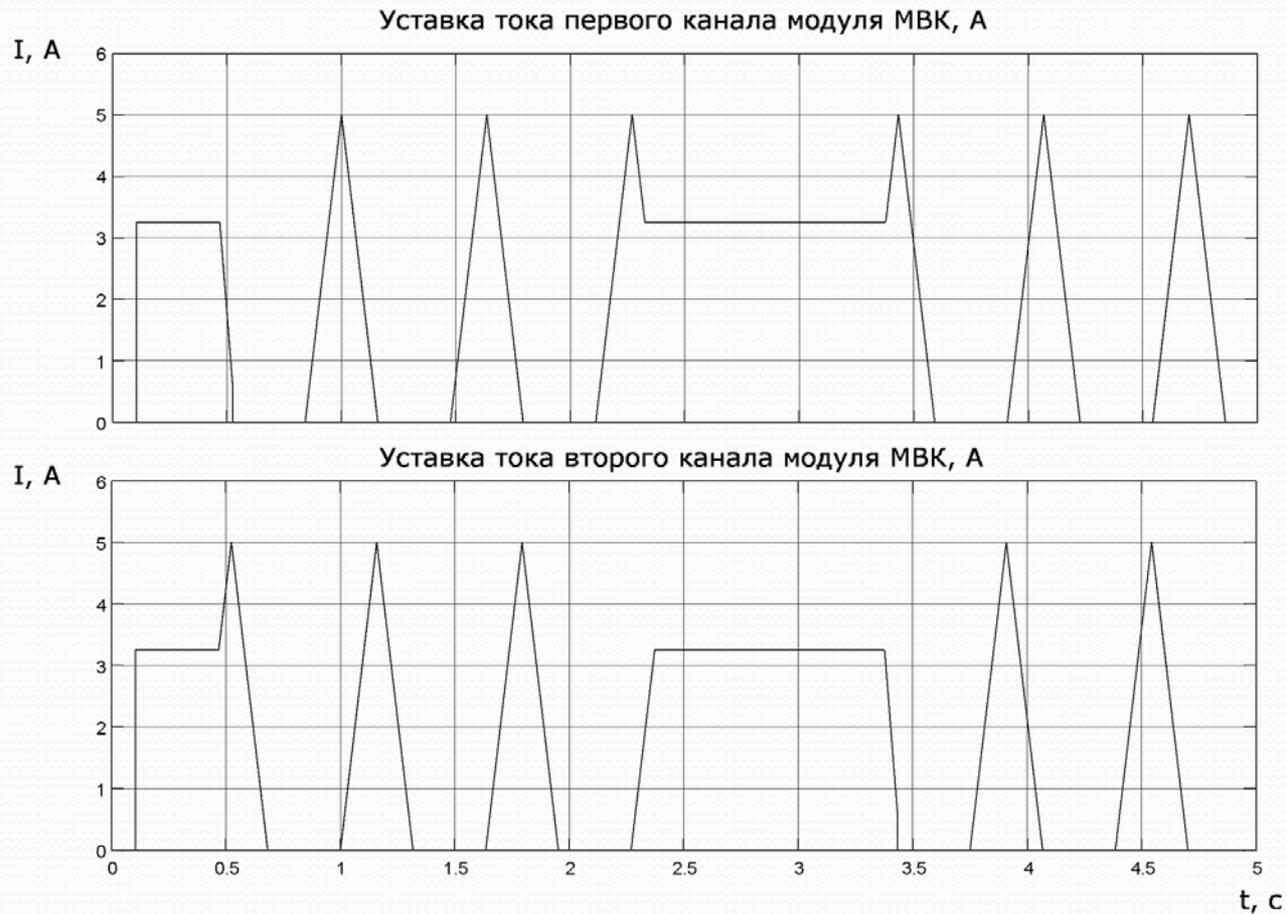
truthable t = tt(f,g,h,i)	function ff1	truthable [b, c, d, e] = tt2(w, x, y, z)	truthable t1 = tt3	truthable tl = tt4(x1)
function C1 = f1	function C2 = f2	function C3 = f3		
function C4 = f4	function C5 = f5	function C6 = f6		



Фрагмент графа переходов конечного автомата

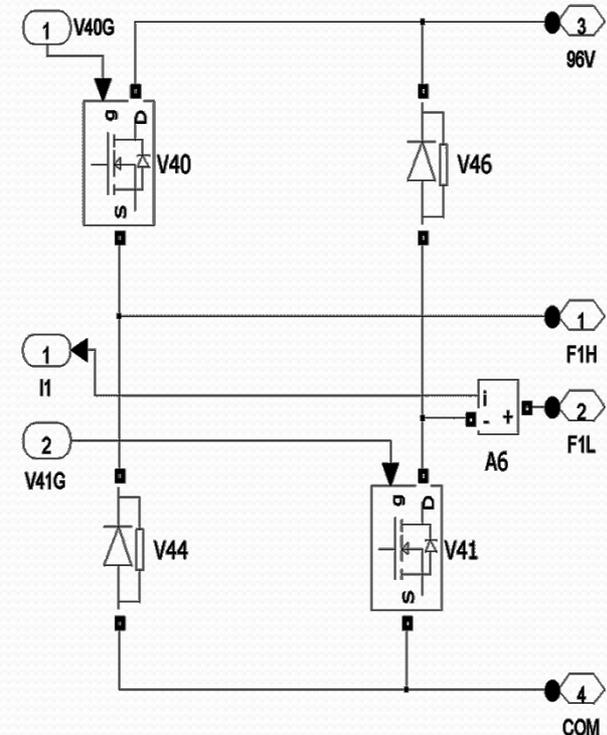
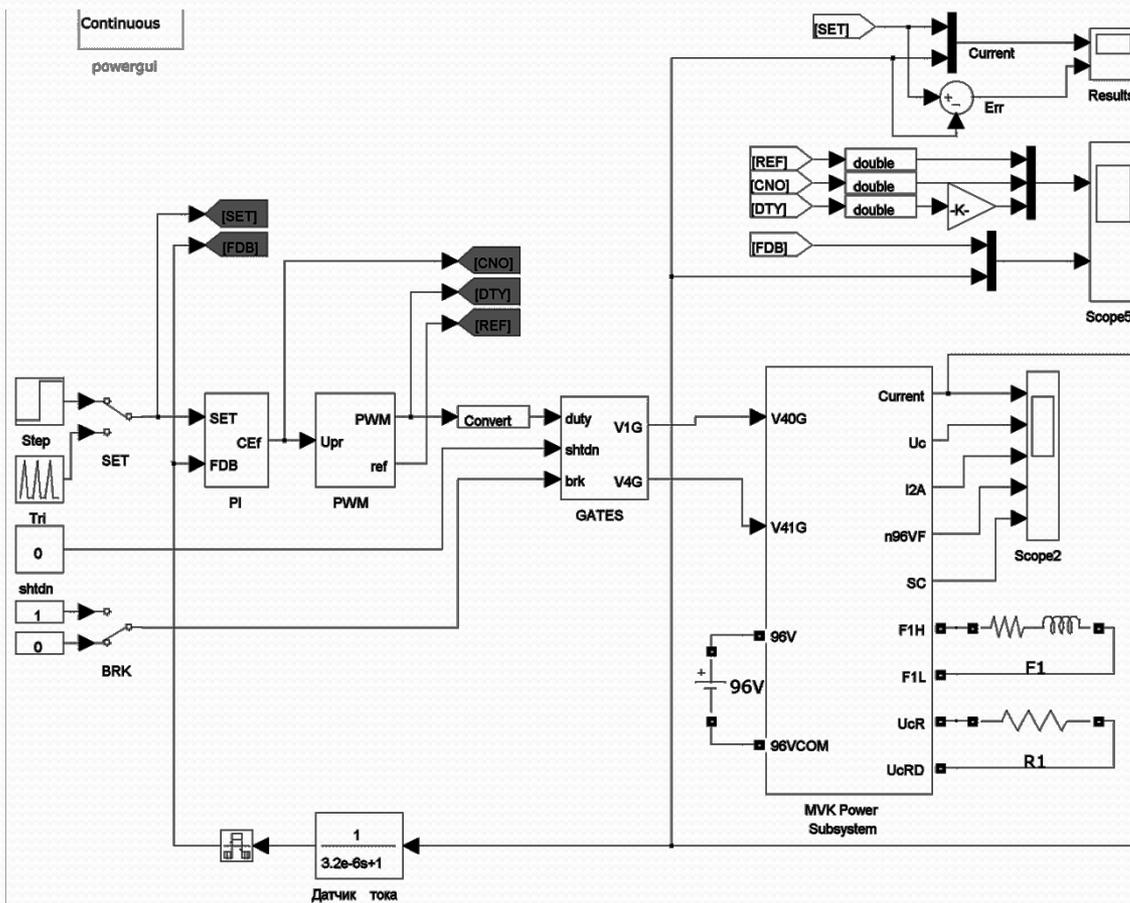


4. Моделирование системы графов переходов средствами *MATLAB-Simulink*

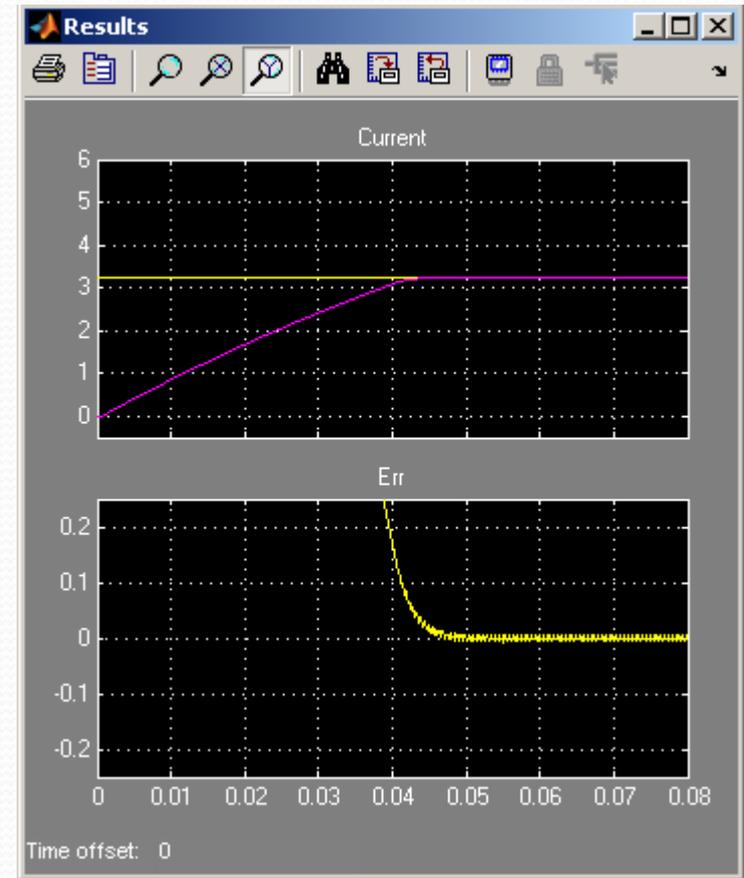
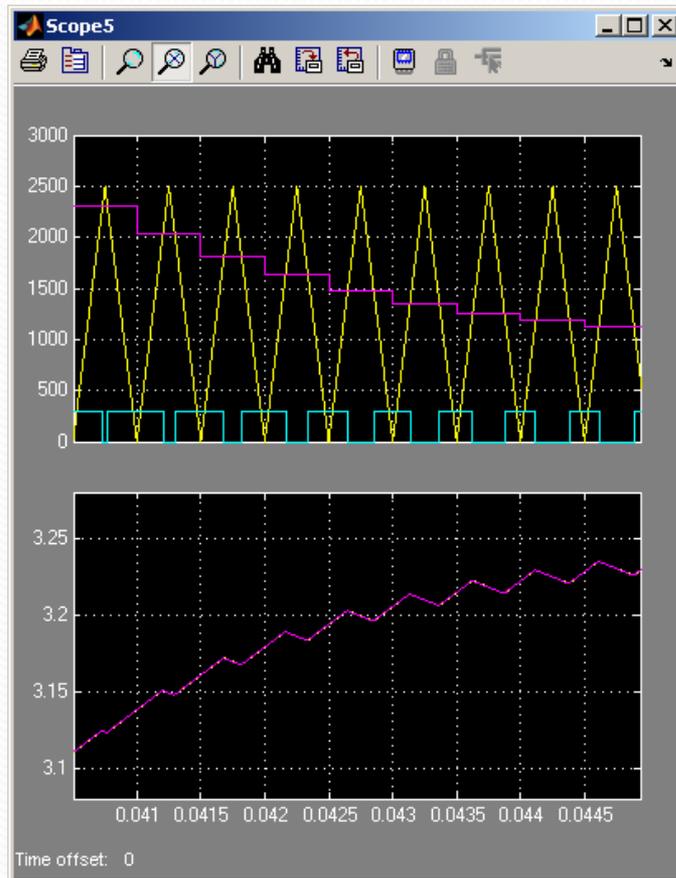


Технология создания контура регулирования тока

1. Разработка модели контура регулирования тока средствами *MATLAB-Simulink*



2. Синтез цифровых регуляторов и моделирование контура регулирования

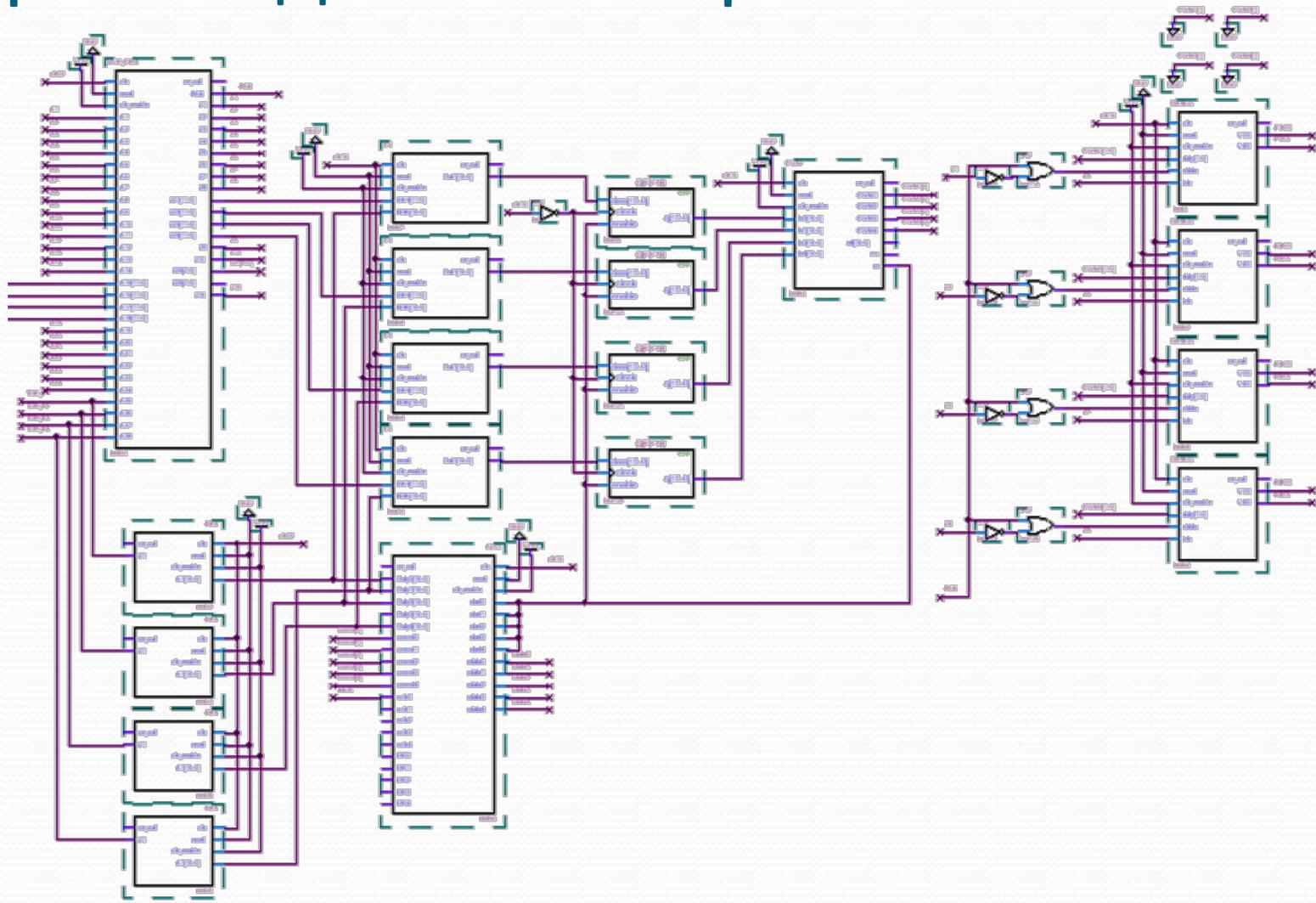


Генерация исходного кода из полученных функциональных блоков на языке VHDL средствами MATLAB

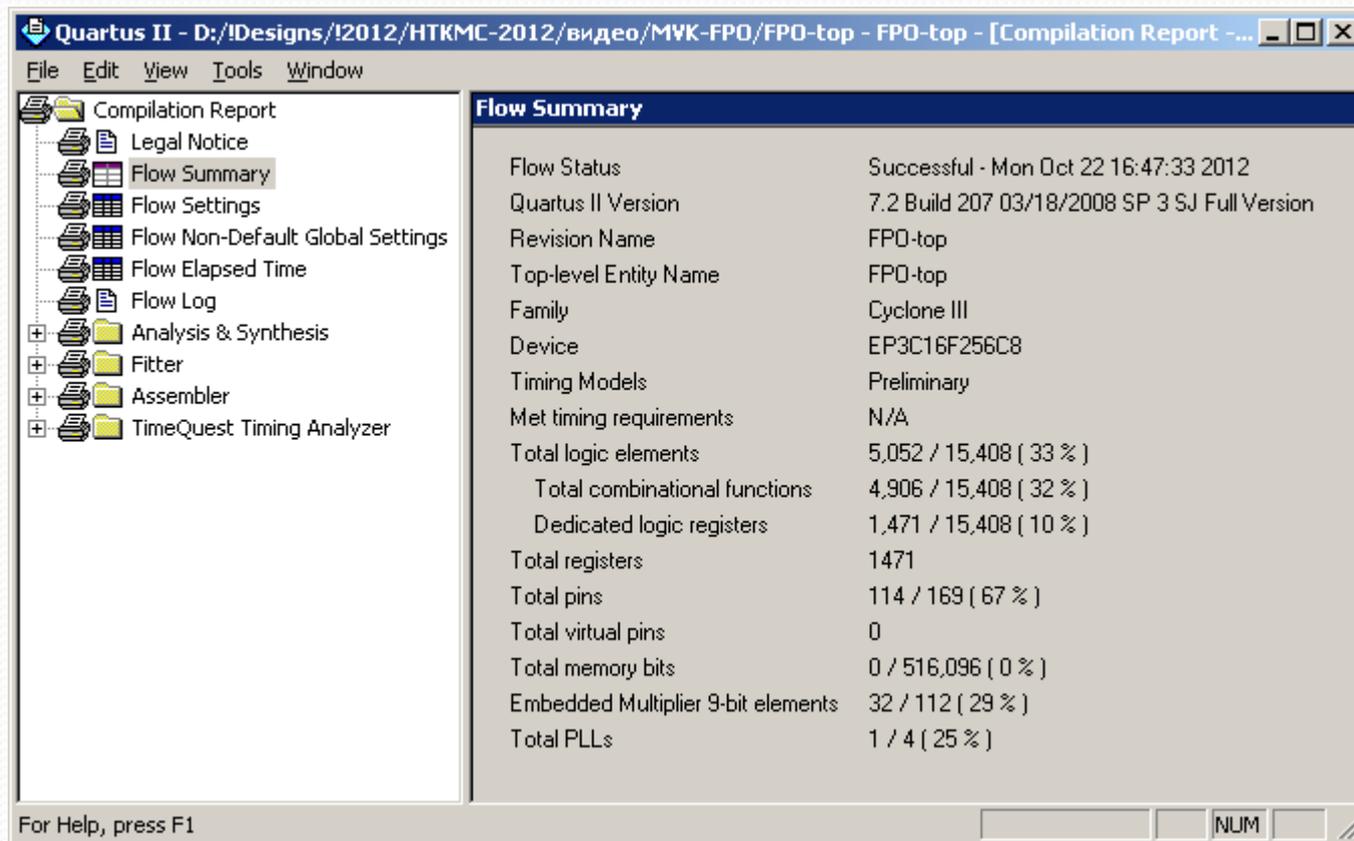
```
GATES_CH_1_output : PROCESS (is_S3,  
is_S4, is_T1, duty_signed, shtdn, brk,  
V1G_reg, V2G_reg, V3G_reg, V4G_reg)  
BEGIN  
    is_S3_next <= is_S3;  
    is_S4_next <= is_S4;  
    is_T1_next <= is_T1;  
    V1G_reg_next <= V1G_reg;  
    V2G_reg_next <= V2G_reg;  
    V3G_reg_next <= V3G_reg;  
    V4G_reg_next <= V4G_reg;  
    IF (shtdn AND ( NOT  
hdlcoder_to_stdlogic (is_T1 = IN_S5)))  
    = '1' THEN  
        is_S3_next <= IN_NO_ACTIVE_CHILD;  
        is_S4_next <= IN_NO_ACTIVE_CHILD;  
        is_T1_next <= IN_S5;  
        V1G_reg_next <= '0';  
        V2G_reg_next <= '0';  
        V3G_reg_next <= '0';
```

```
        V4G_reg_next <= '0';  
    ELSE  
        CASE is_T1 IS  
            WHEN IN_S1 =>  
                IF duty_signed = 0 THEN  
                    is_T1_next <= IN_S3;  
                    IF brk = '1' THEN  
                        is_S3_next <= b_IN_S2;  
                        V1G_reg_next <= '0';  
                        V3G_reg_next <= '0';  
                        V2G_reg_next <= '0';  
                        V4G_reg_next <= '0';  
                    ELSE  
                        is_S3_next <= IN_S1;  
                        V1G_reg_next <= '0';  
                        V3G_reg_next <= '0';  
                        V2G_reg_next <= '1';  
                        V4G_reg_next <= '1';  
                    END IF;  
                END IF;
```

Разработка схемы из полученных функциональных блоков средствами производителя микросхем



Компиляция и последующая загрузка в модуль



Quartus II - D:/!Designs/!2012/HTKMC-2012/видео/МVK-FPO/FPO-top - FPO-top - [Compilation Report ...]

File Edit View Tools Window

Compilation Report

- Legal Notice
- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- TimeQuest Timing Analyzer

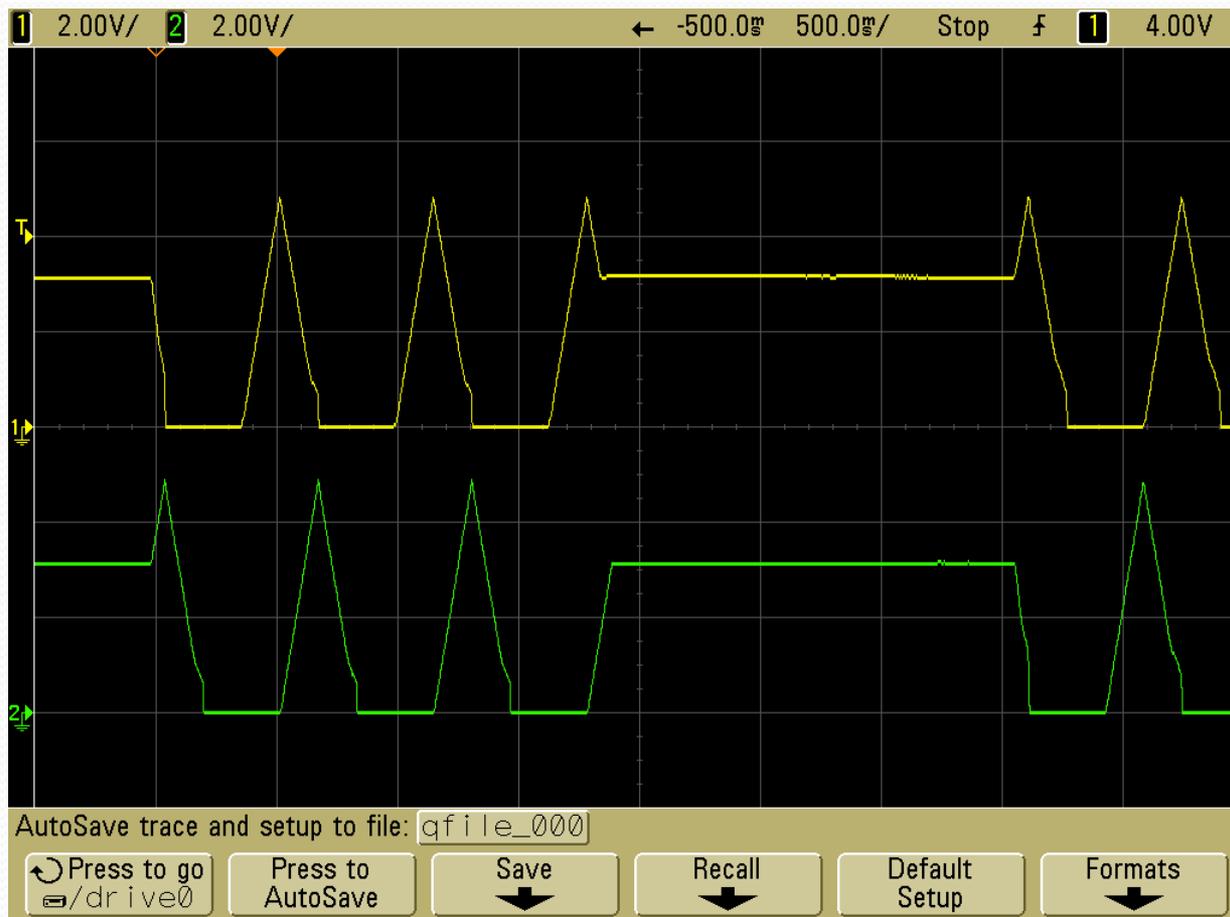
Flow Summary

Flow Status	Successful - Mon Oct 22 16:47:33 2012
Quartus II Version	7.2 Build 207 03/18/2008 SP 3 SJ Full Version
Revision Name	FPO-top
Top-level Entity Name	FPO-top
Family	Cyclone III
Device	EP3C16F256C8
Timing Models	Preliminary
Met timing requirements	N/A
Total logic elements	5,052 / 15,408 (33 %)
Total combinational functions	4,906 / 15,408 (32 %)
Dedicated logic registers	1,471 / 15,408 (10 %)
Total registers	1471
Total pins	114 / 169 (67 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	32 / 112 (29 %)
Total PLLs	1 / 4 (25 %)

For Help, press F1

NUM

Проверка функционирования модулей в составе прибора на стенде (комплексная отладка)



Результаты комплексной отладки и межведомственных испытаний

- Внесение изменений на этапе комплексной отладки происходило очень легко
- По результатам межведомственных испытаний двух образцов системы не возникло необходимости вносить изменения в ПО модулей
- Во время проведения МВИ представитель Заказчика отметил основные достоинства новых модулей по сравнению с предыдущими реализациями: «Простота изменения алгоритмов работы и выходных параметров модулей. Возможность проведения диагностики исполнительного механизма без дополнительного оборудования и извлечения модулей из прибора»

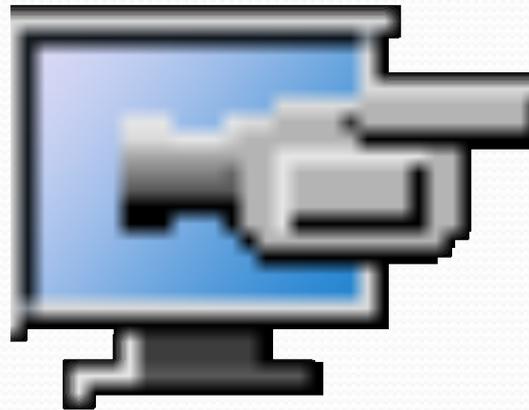
Выводы

- Применение приведенной технологии позволяет исключить процесс ручного программирования, привлечь специалистов по автоматике и силовым полупроводниковым преобразователям к разработке программного обеспечения.
- Автоматные программы по сравнению с традиционными наглядны, доступны для интуитивного понимания, хорошо документируются, легко модернизируются.
- Процесс программирования ПЛИС существенно формализован, что уменьшает число ошибок и облегчает его применение.
- Возможно использование методов формальной верификации.

Литература

1. *Harel D.* Statecharts: A Visual Formalism for Complex Systems // Science of Computer Programming. 1987. Vol 8, pp. 231 – 274.
2. *Поликарпова Н. И., Шалыто А. А.* Автоматное программирование. СПб.: Питер, 2010. – 176 с.
3. *Козаченко В. Ф.* Эффективный метод программной реализации дискретных управляющих автоматов во встроенных системах управления // Электропривод и автоматика.
URL: http://www.motorcontrol.ru/publications/state_machine.pdf (дата обращения 25.10.2010).
4. Basic Workflow for Building a Stateflow Chart / R2010b MathWorks Documentation.
URL: <http://www.mathworks.com/help/toolbox/stateflow/gs/bqnmvk8.html> (дата обращения 26.10.2010).
5. *Янкин Ю.Ю., Шалыто А.А.* Автоматное программирование ПЛИС в задачах управления электроприводом // Информационно-управляющие системы. 2011. №1. с. 50-56.
6. *Янкин Ю.Ю., Шалыто А.А.* Применение автоматного подхода при программировании модулей управления шаговыми двигателями, выполненных на основе ПЛИС // Системы управления и обработки информации: Научн.-техн. сб. / ОАО «Концерн «НПО «Аврора», СПб., 2011., Вып. 22 С. 92–103

Пример создания функционального блока ПО



Пример 2.exe

Метод создания программного обеспечения модулей, выполненных на основе программируемых логических интегральных схем

Юрий Юрьевич Янкин
Анатолий Абрамович Шальто

2012