

# МЭС-2016

## **Оценка надежности при одиночных сбоях в кэш-памяти в маршруте проектирования системы на кристалле**

Ольга Мамутова

*Санкт-Петербургский политехнический  
университет Петра Великого*

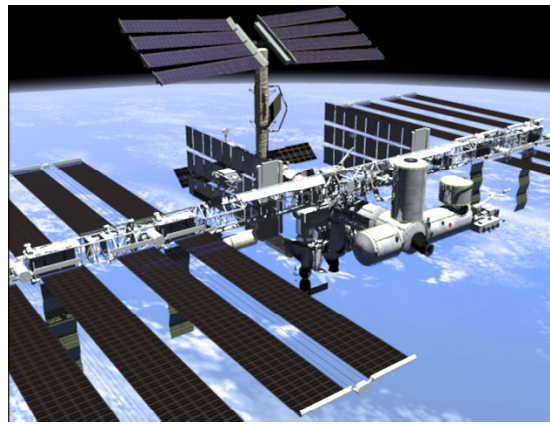
*Кафедра компьютерных систем и программных технологий*

# Актуальность проблемы одиночных сбоев в кэш-памяти

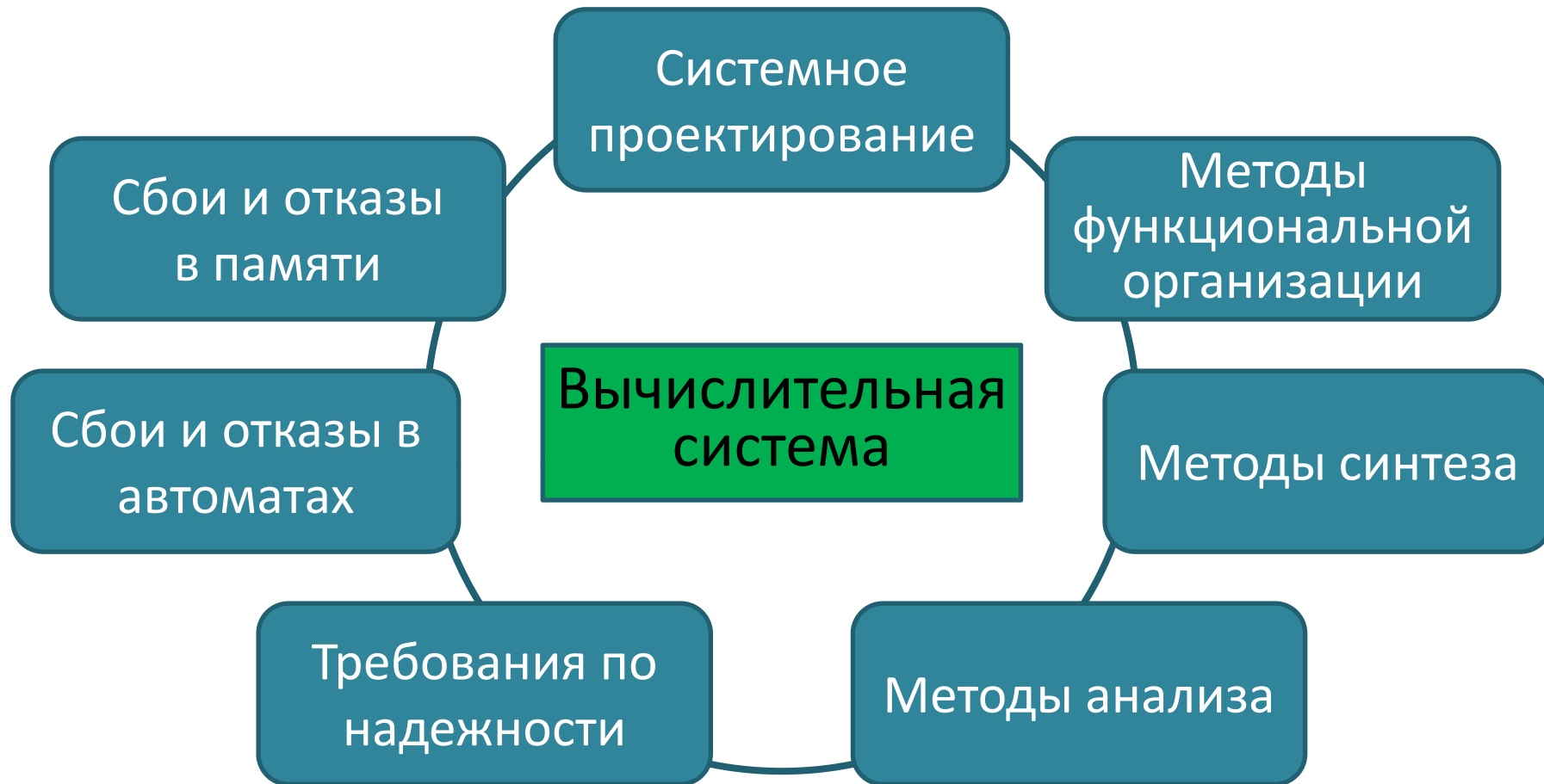
**Целевой класс систем:** вычислители в бортовых приборах авиакосмической техники.

**До 80% площади кристалла – память, уязвимая к одиночным сбоям.**

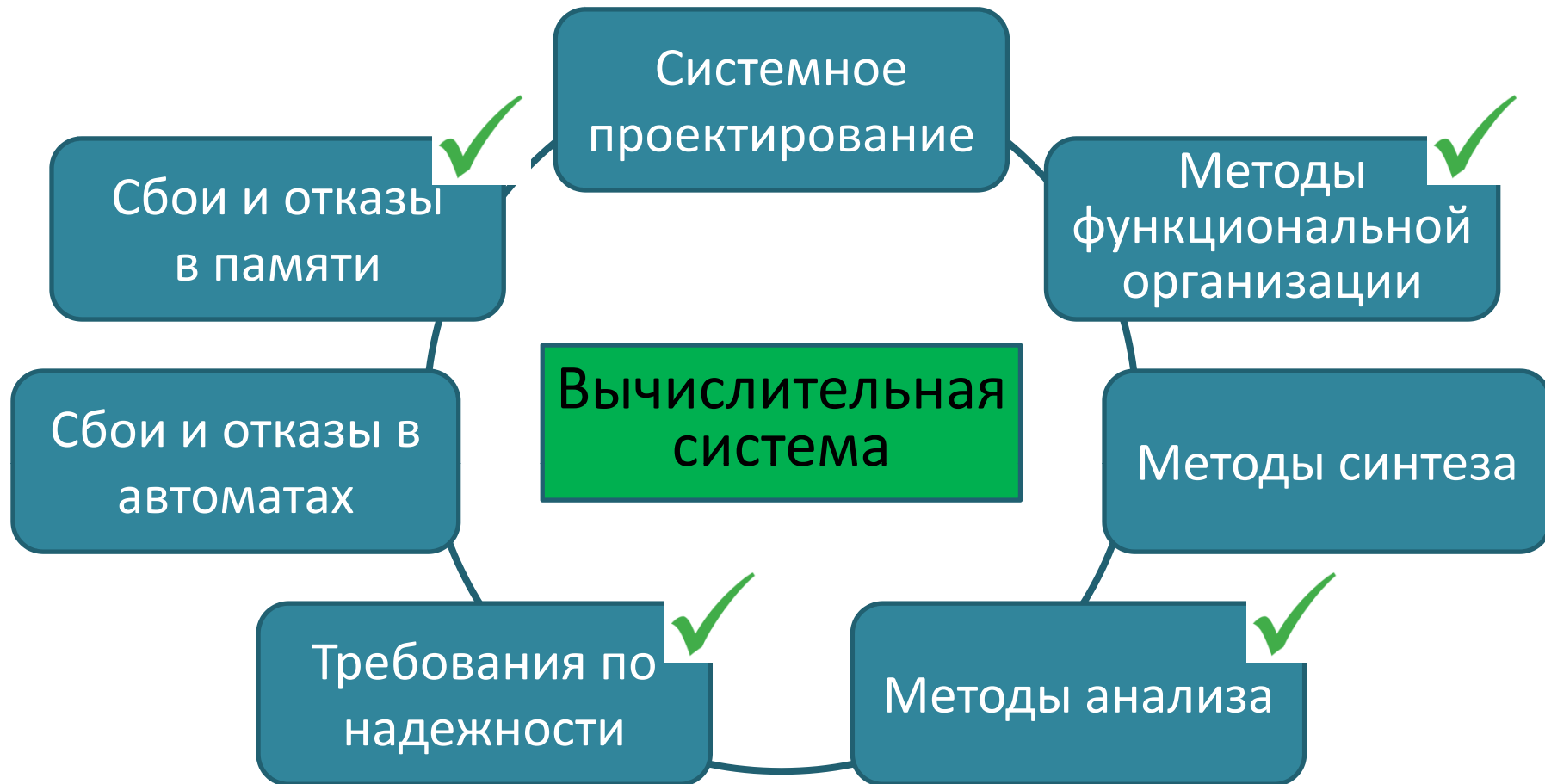
**Кэш-память хранит активно используемые процессором данные, поэтому велика вероятность информационного отказа.**



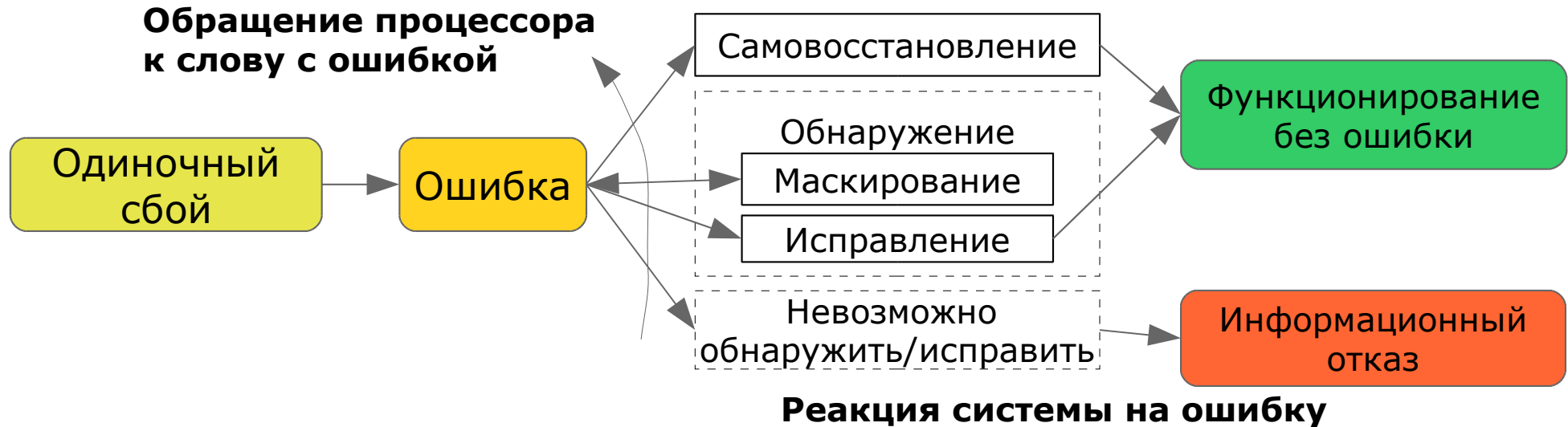
# Методология и теория проектирования отказоустойчивых восстанавливаемых вычислительных систем



# Методология и теория проектирования отказоустойчивых восстанавливаемых вычислительных систем



# Задача: анализ уязвимости процессора к одиночным сбоям в кэш-памяти



## Исходные данные:

интенсивность одиночных сбоев в элементе памяти

## Параметры:

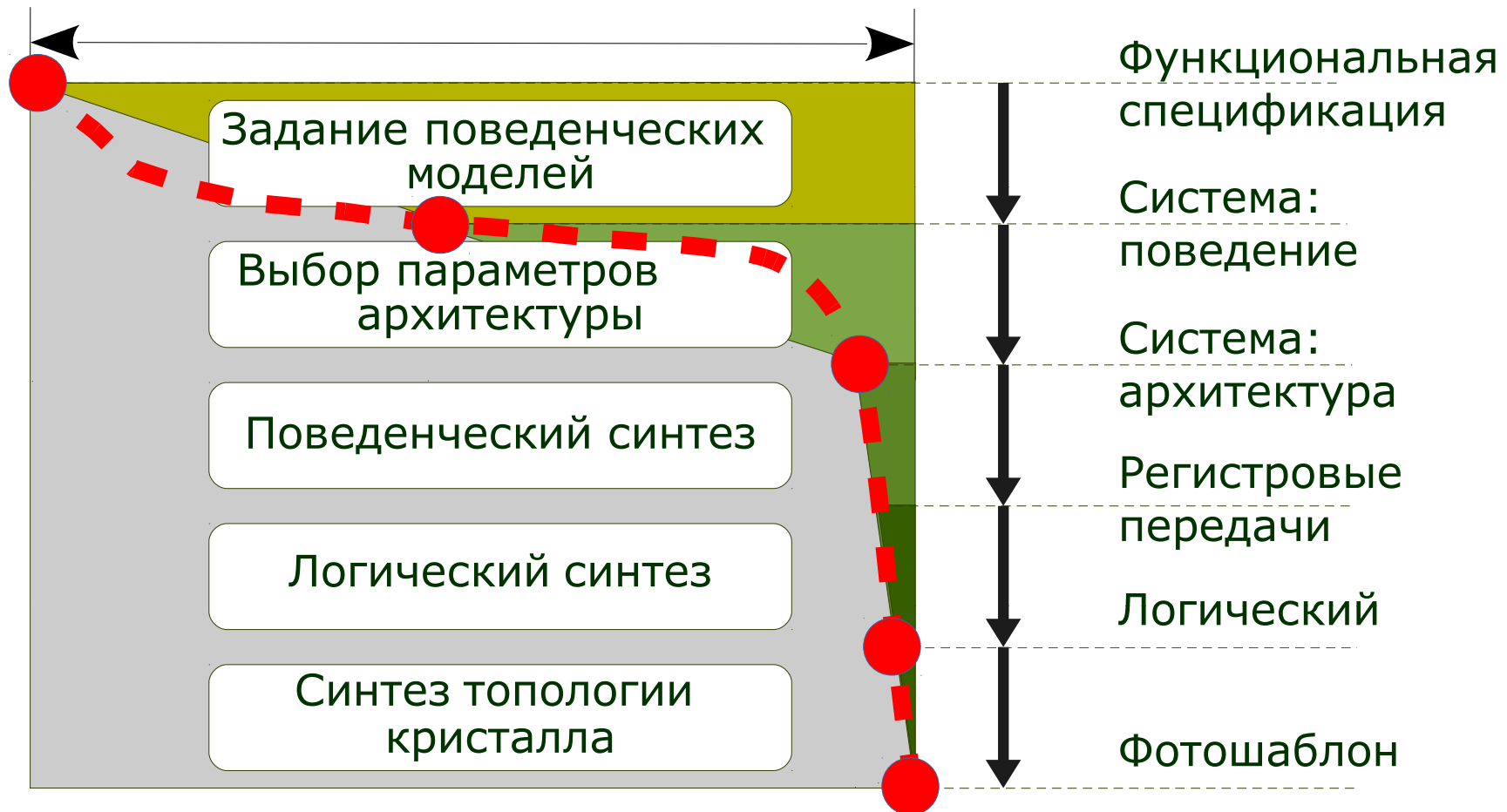
- архитектура кэш-памяти
- меры избыточности
- вычислительная нагрузка

## Показатель уязвимости:

вероятность распространения ошибки из кэш-памяти

# Маршрут проектирования системы

Пространство возможных решений    Уровень абстракции



**Платформенно-ориентированный** подход – стандарт де-факто.

**Требование к средствам анализа** – высокая скорость оценки.

# Методы анализа показателя уязвимости для кэш-памяти

## Аналитическое моделирование

- не учитывают характер вычислительной нагрузки
- или требуют анализа трассы исполнения программы

## Имитационное моделирование, Внесение неисправностей на базе симулятора процессора, Внесение неисправностей на базе симулятора в EDA

- требуют длительного моделирования для получения статистически значимых результатов

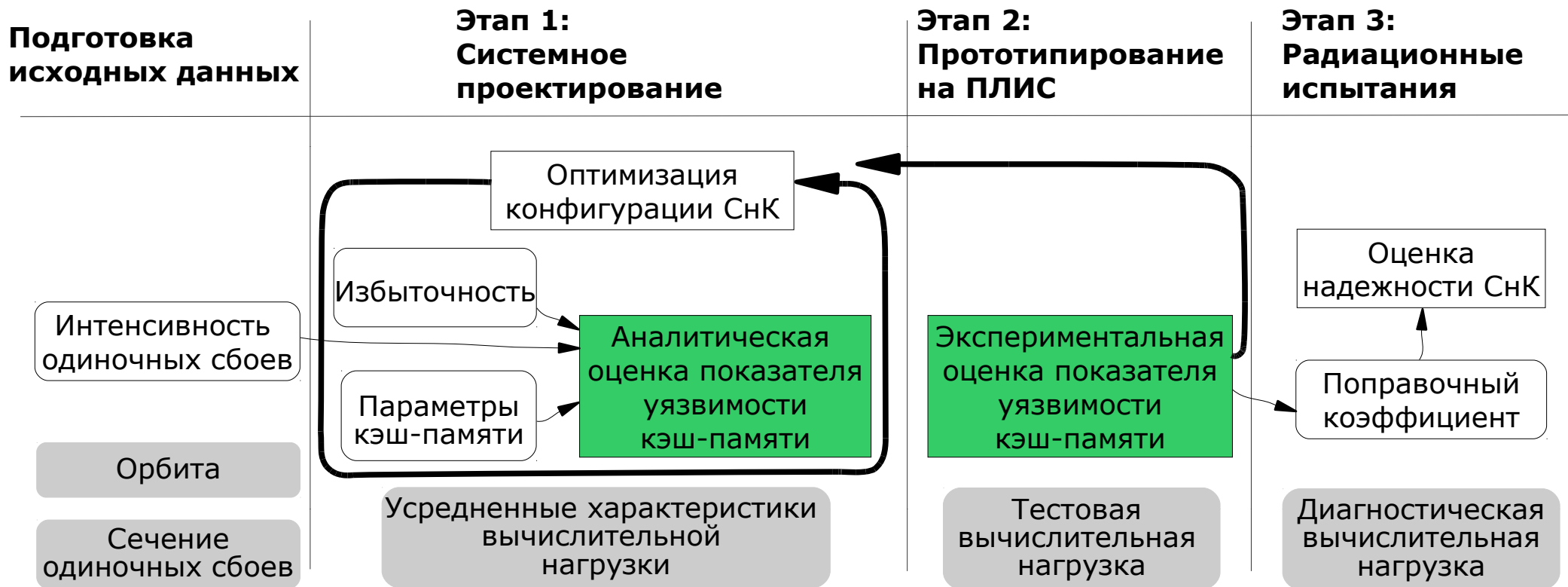
## Внесение неисправностей на базе ПЛИС

- сбой в кэш-памяти сложно эмулировать программно
- доступ к блокам памяти затруднен
- существенные затраты площади кристалла

## Физическое внесение неисправностей

- высока цена ошибки проектирования
- проблема интерпретации результатов для сложной системы

# Предлагаемая методика выбора параметров кэш-памяти



**1. Аналитическая модель** – при составлении бюджета и в ходе оптимизации на этапе системного проектирования.

**2. Внесение неисправностей на базе ПЛИС** – для валидации мер резервирования и уточнения оценок на этапе прототипирования.



# Математическая модель надежности кэш-памяти

**Этап проектирования:** системное проектирование

**Применение:**

- быстрая оценка для составления бюджета проекта СнК
- анализ закономерностей в функции показателя уязвимости

**Новизна:**

- учет характера вычислительной нагрузки
- без анализа трассы исполнения программы

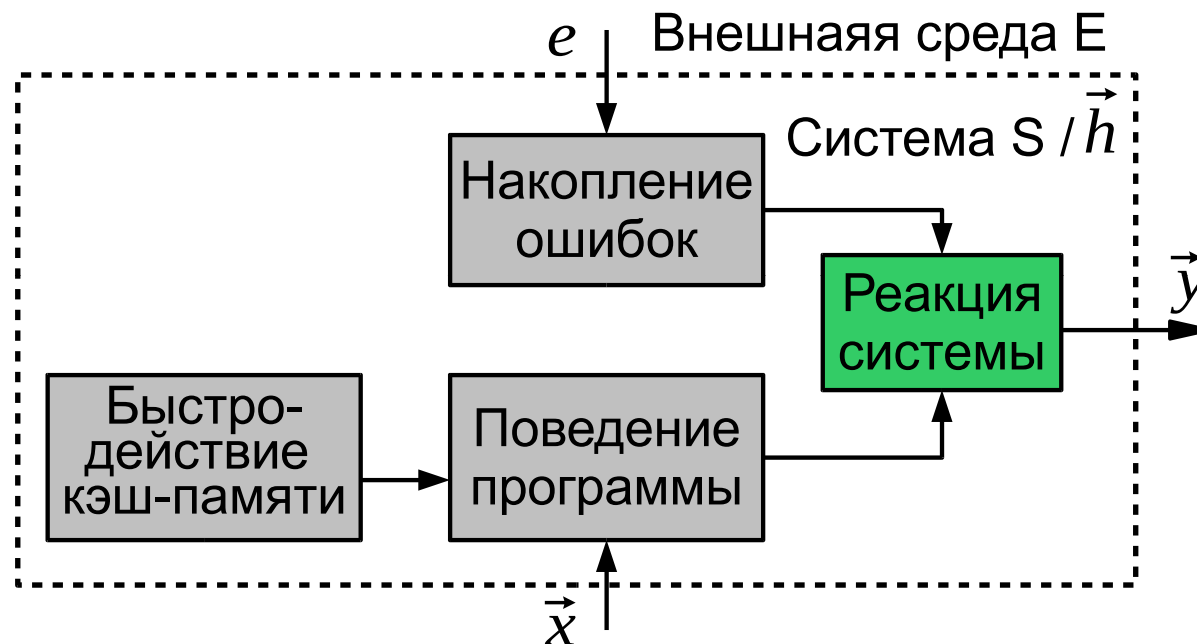
**Принятые допущения:**

- адреса обращений процессора:  
    модель независимых обращений
- вероятность промаха:  
    расширенный эмпирический закон  $\sqrt{2}$

# Структура математической модели

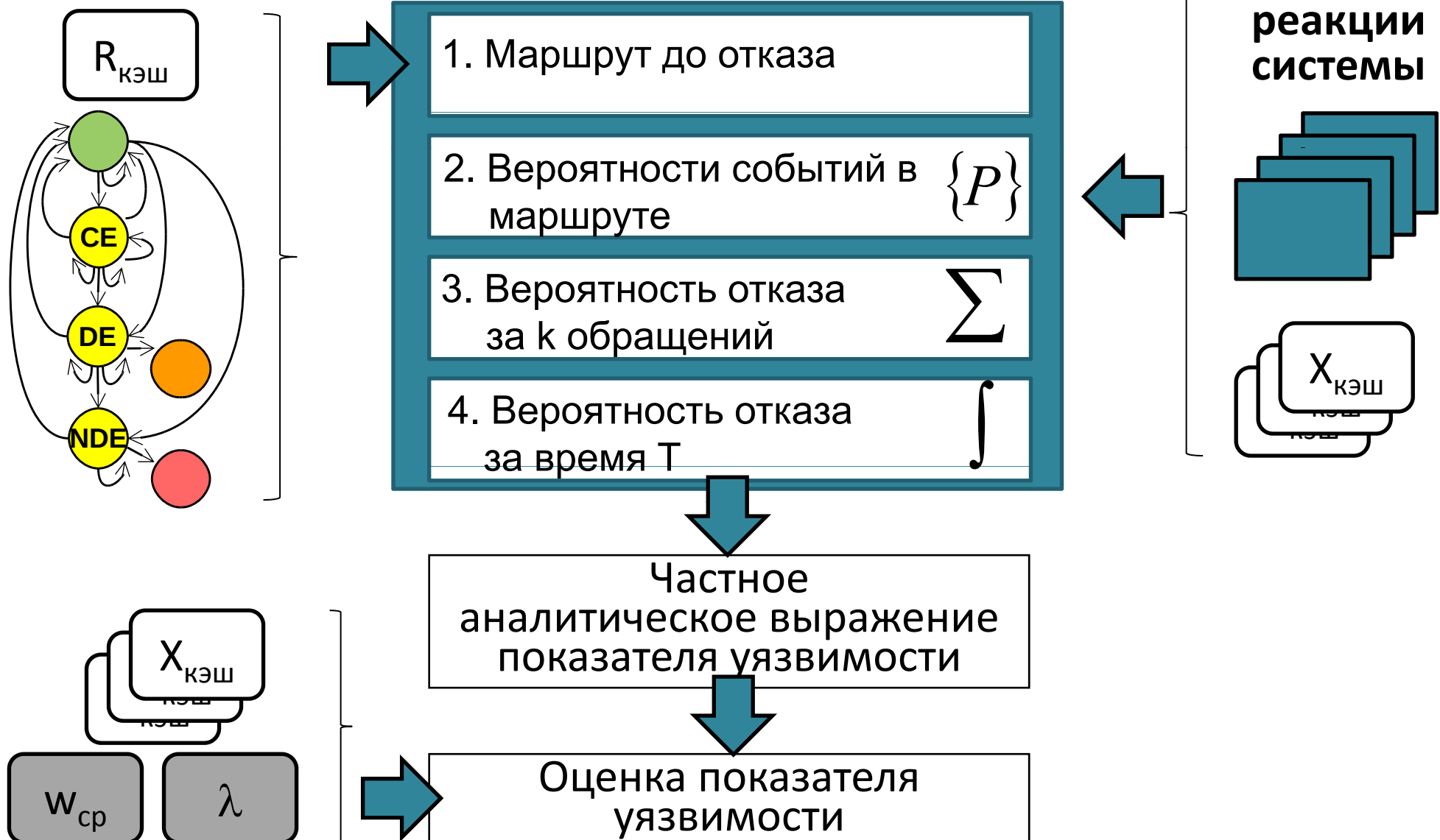
**Закон функционирования:**

$$\vec{y}(t) = F_S(\vec{x}, \vec{e}, \vec{h}, t)$$



- $\vec{y}(t)$  - показатели уязвимости массива строк/таблицы тэгов
- $\vec{h}$  - параметры резервирования, быстродействия стратегии записи/загрузки
- $e = \lambda$  - интенсивность одиночных сбояв
- $\vec{x}$  - распределение операций записи/чтения, промахов/попаданий, локальность обращений, штраф при промахе

# Алгоритм аналитической оценки показателя уязвимости



# Примеры аналитической оценки

	Бит четности	SEC-DED
Запись таблицы тэгов		
Строка массива данных		
Масштабирование		

# Внесение неисправностей на базе ПЛИС

**Этап проектирования:** прототипирование на базе ПЛИС

**Применение:**

- валидация мер повышения надежности в системе
- уточнение оценки показателя уязвимости

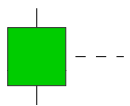
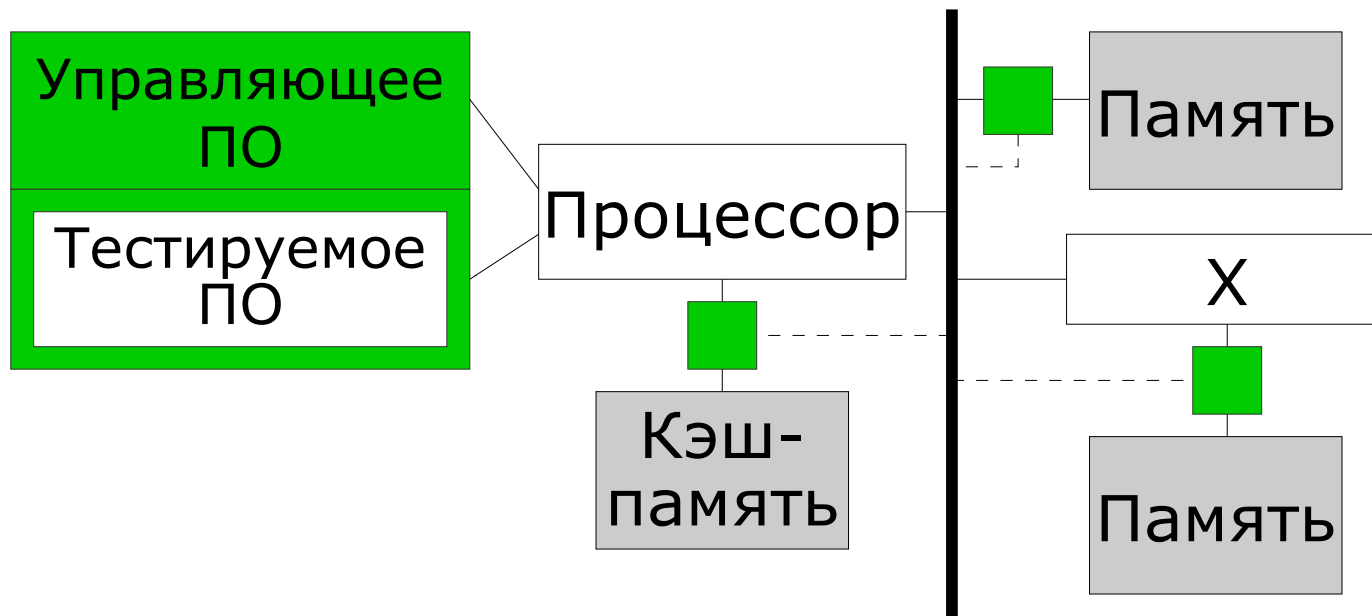
**Исходные данные:**

- синтезируемое описание устройства
- тестовые программы

**Суть подхода:**

- встроенное тестирование под управлением процессора с помощью сети саботажников

# Метод внесения неисправностей



сеть агентов внесения неисправностей для доступа к «теневой» памяти

Тестируемое ПО

тестовые, штатные или диагностические программы

Управляющее ПО

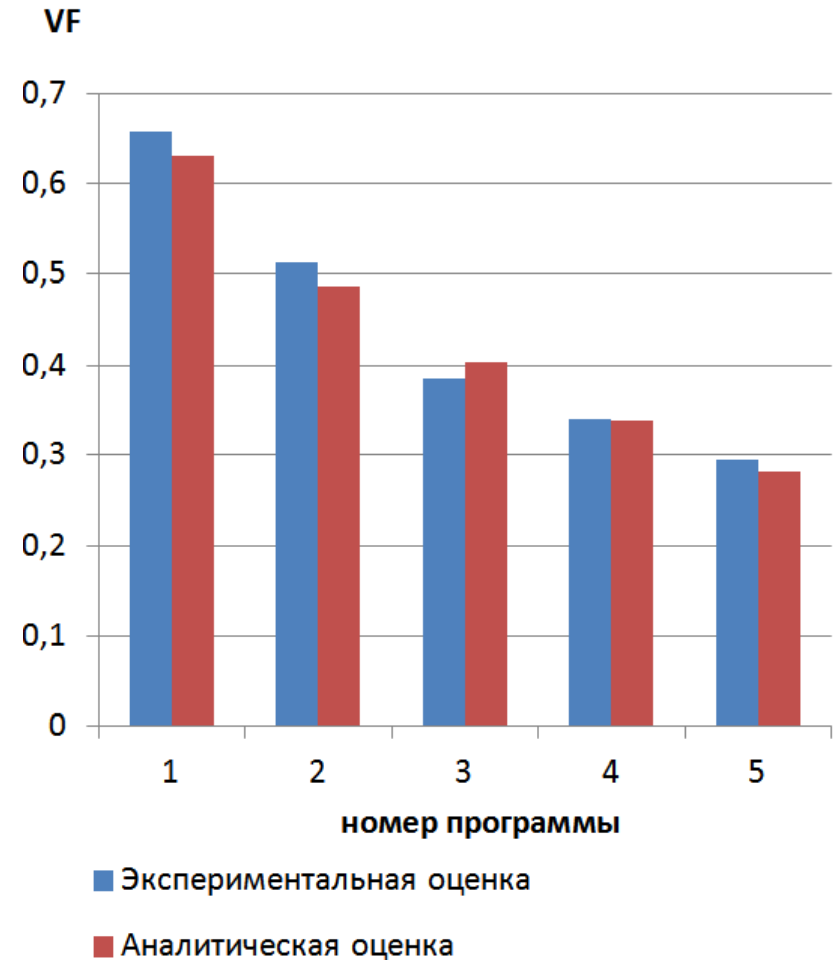
программа управления автономным внесением неисправностей

# Применение внесения неисправностей

1. Подтверждение аналитической оценки с помощью синтетических тестов.

2. Характеризация тестовых программ и уточнение оценок показателя уязвимости.

3. Определение поправочных коэффициентов для диагностических программ для физического внесения неисправностей.



# Заключение

- **Оценку уязвимости процессора к одиночным сбоям необходимо проводить на всех этапах проектирования, и такая оценка должна быть быстрой.**
- **Для кэш-памяти оценка усложняется сильной зависимостью от характера вычислительной нагрузки.**
- **Созданный подход к аналитической оценке показателя уязвимости кэш-памяти к одиночным сбоям решает проблему быстрой и грубой оценки на ранних этапах проектирования.**
- **Созданный подход к внесению неисправностей на базе ПЛИС решает проблему эффективного тестирования на этапе прототипирования.**