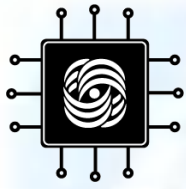


ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Лекция 10: *Обеспечение отказоустойчивости ИУС РВ*

Кафедра АСВК,
Лаборатория Вычислительных Комплексов
Балашов В.В.



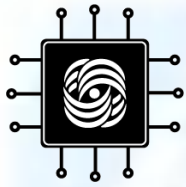
Цена отказа: Ariane-5

(причина: неисправность ПО)

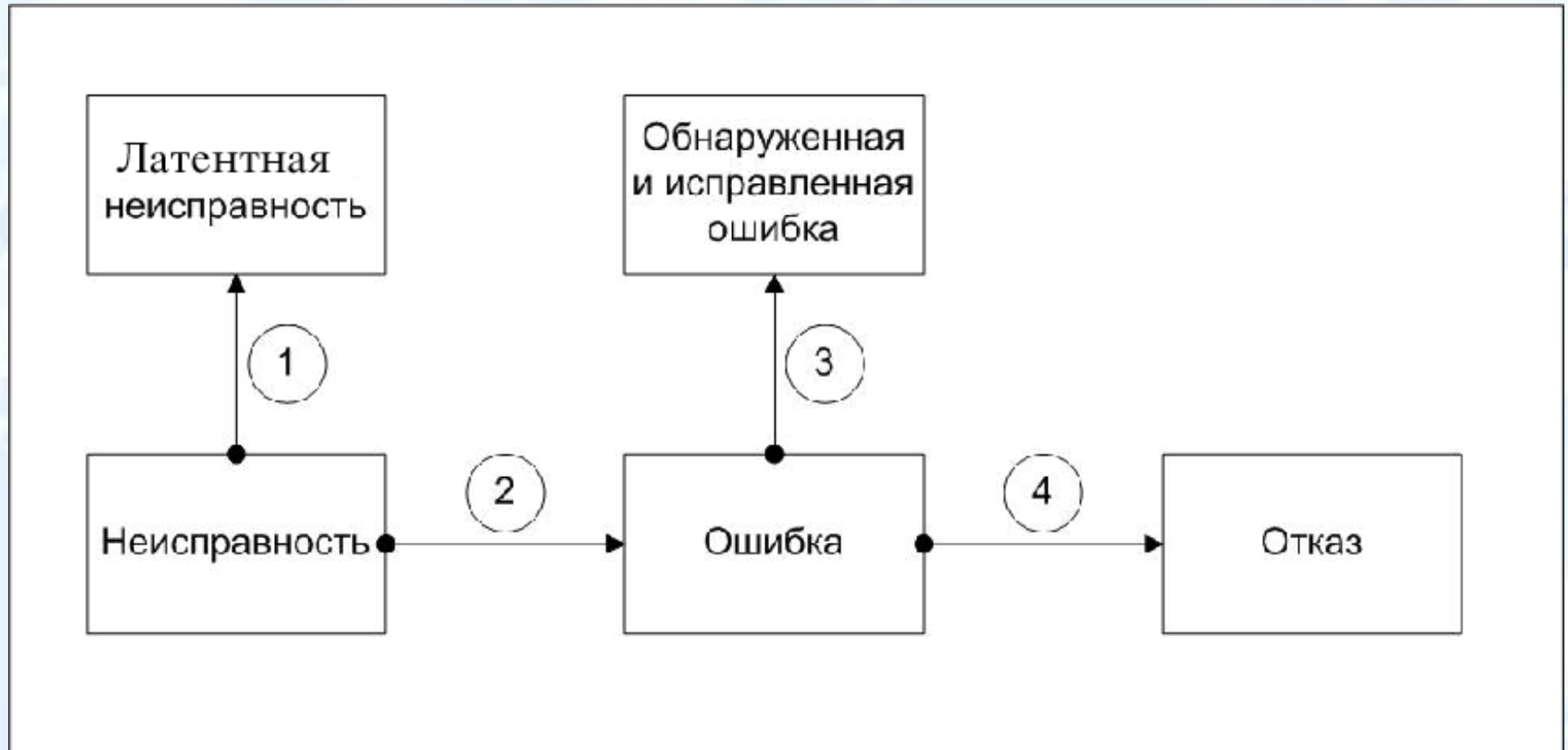
- Июнь 1996 года, взрыв ракеты спустя 40 сек. после старта,
- Ущерб – \$500млн (разработка – \$7 млрд.),
- Причина – 64bit float -> 16bit int.



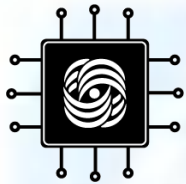
*Кажется,
что-то
пошло не
так...*



Неисправность, ошибка, отказ



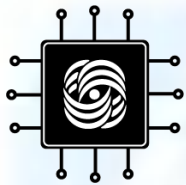
- Полностью избежать неисправностей невозможно
- Цель: минимизировать вероятность отказа



Классификация неисправностей

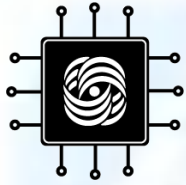
- Активность: латентные / активные
- Постоянство: проходящие / постоянные
- Источник: внешнее воздействие / ошибка разработки
- Распространение последствий: обнаруживаются и локализуются / проникают в другие подсистемы
- Одиночность: одиночные / групповые
- Взаимосвязанность: независимые / связанные

(источник: Software Fault Tolerance: A Tutorial, NASA, 2014, pp.28-29, <http://www.iet.unipi.it/c.bernardeschi/didattica/ANNO2014-15/DEP/SoftwFT.pdf>)



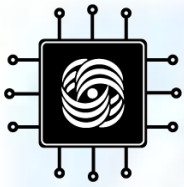
Шаги противодействия неисправности

- Обнаружение
- Ограничение распространения
- Маскировка
- Диагностика
- Восстановление
- Возобновление штатного функционирования



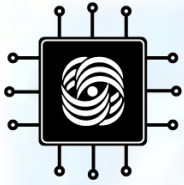
Классификация неисправностей (2)

- По локализации
 - Программные
 - Аппаратные
- По этапу возникновения
 - Проектирование/разработка
 - => все изделия, созданные по проекту
 - Производство
 - Дефект серии => все изделия серии
 - Дефект при производстве конкретного изделия
 - Эксплуатация
 - => одиночные изделия, с учётом особенностей эксплуатации



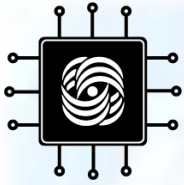
Борьба с серийными неисправностями

- Возникают на этапе проектирования, разработки или серийного производства
 - На ранних этапах их и следует обнаруживать...
- Затрагивают всю серию компонентов
 - Отзыв серии и всех использующих её систем...
- Борьба: использование проектного или реализационного разнообразия
 - Аппаратура: различные архитектуры, производители и элементная база
 - ПО: различные языки, алгоритмы/подходы, команды разработчиков



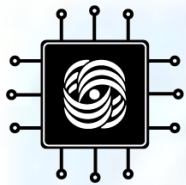
Специфика ИУС РВ

- Жесткие условия эксплуатации
 - Внешние воздействия вызывают неисправности оборудования
- Недопустимость прекращения функционирования при возникновении ошибки
 - Ошибка = реализация неисправности
- Невозможность оперативного ремонта (или ремонта вообще)
 - Самолет, спутник
- Реакция на ошибки должна укладываться в директивные сроки



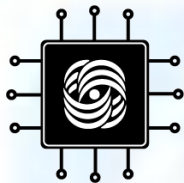
Принципы построения отказоустойчивых систем

- Недопустимость единственной точки отказа
- Поддержка локализации отказа (обнаружения отказавшего компонента)
- Нераспространение последствий отказа далее по системе
 - Защита аппаратуры
 - Блокировка распространения некорректных результатов вычислений
- Постепенная деградация
 - По мере отказа подсистем, вначале отключаются второстепенные функции
 - Функции, критические для выживания/восстановления системы поддерживаются «до последнего»
 - Защитный режим: поддержка существования + обеспечение удалённого доступа для диагностики и обслуживания

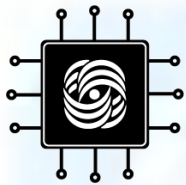


Механизмы обеспечения отказоустойчивости

- Аппаратные
- Программные
- Аппаратно-программные

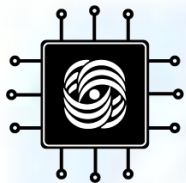


Аппаратные МОО



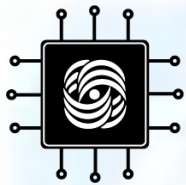
Аппаратное резервирование

- По разнообразию компонентов
 - Использование идентичных компонентов
 - Борьба с дефектами изделия, в т.ч. возникающими в ходе эксплуатации
 - Использование различных компонентов
 - Функционально идентичны
 - Различная элементная база, производитель, проект и т.п.
 - Борьба с дефектами проекта, серии (не только изделия)
- По уровню
 - Система/подсистема
 - Отдельные компоненты

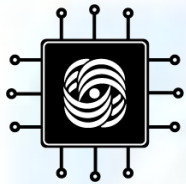


Аппаратное резервирование

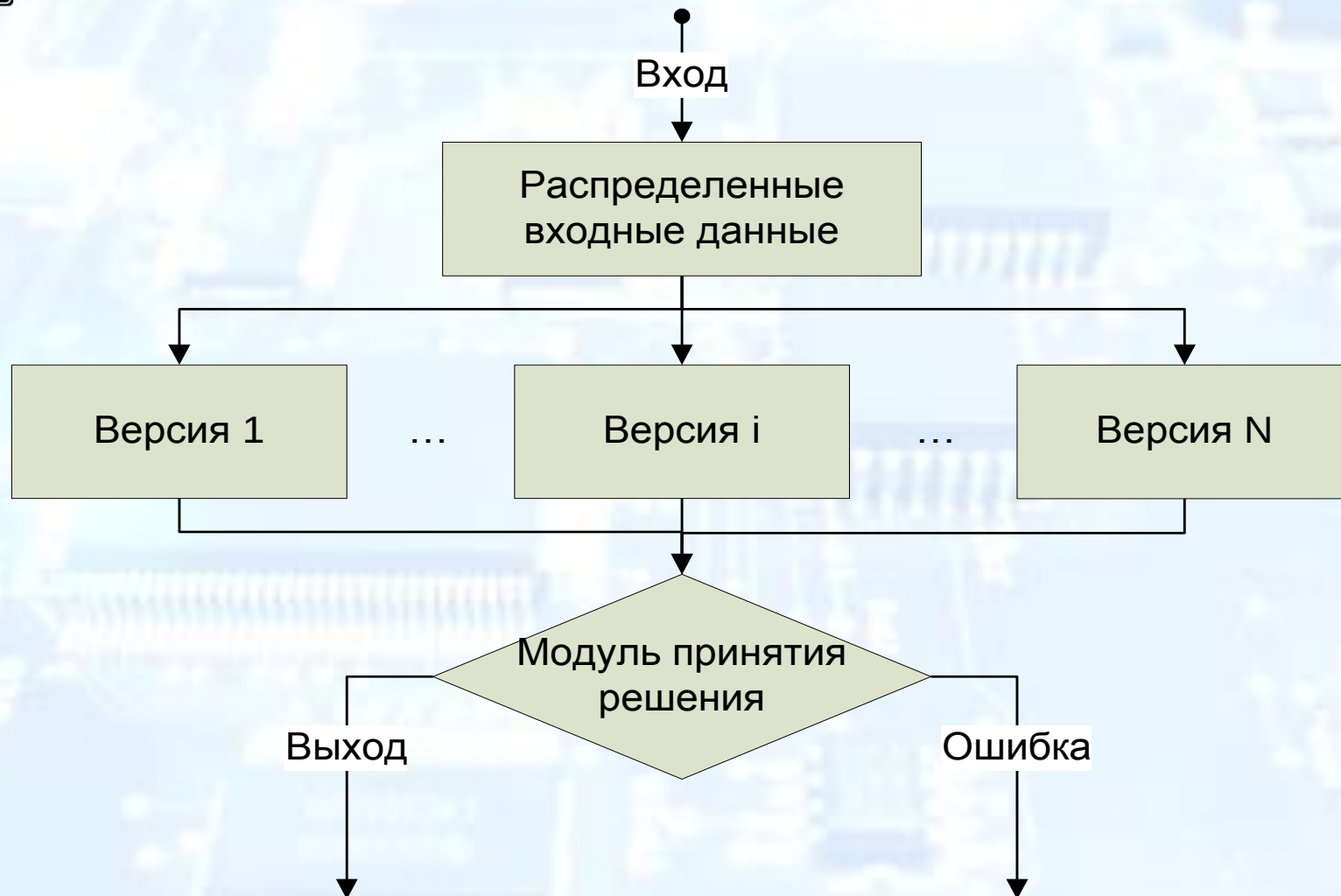
- Активное: основные и резервные компоненты функционируют одновременно
 - Синхронизация данных
 - Минимальное время переключения на резервный компонент
 - Использование результатов:
 - Игнорирование результатов резервных компонентов
 - Голосование (нет выделенного основного компонента)
 - Повышенное энергопотребление
- Пассивное: резервный компонент включается при выходе основного из строя
 - Затраты времени на инициализацию
 - Проблема записи состояния основного компонента (нужно внешнее запоминающее устройство)
 - Экономия энергии
- Жаргон: «горячий» и «холодный» резерв



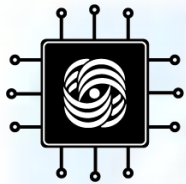
Программные МОО



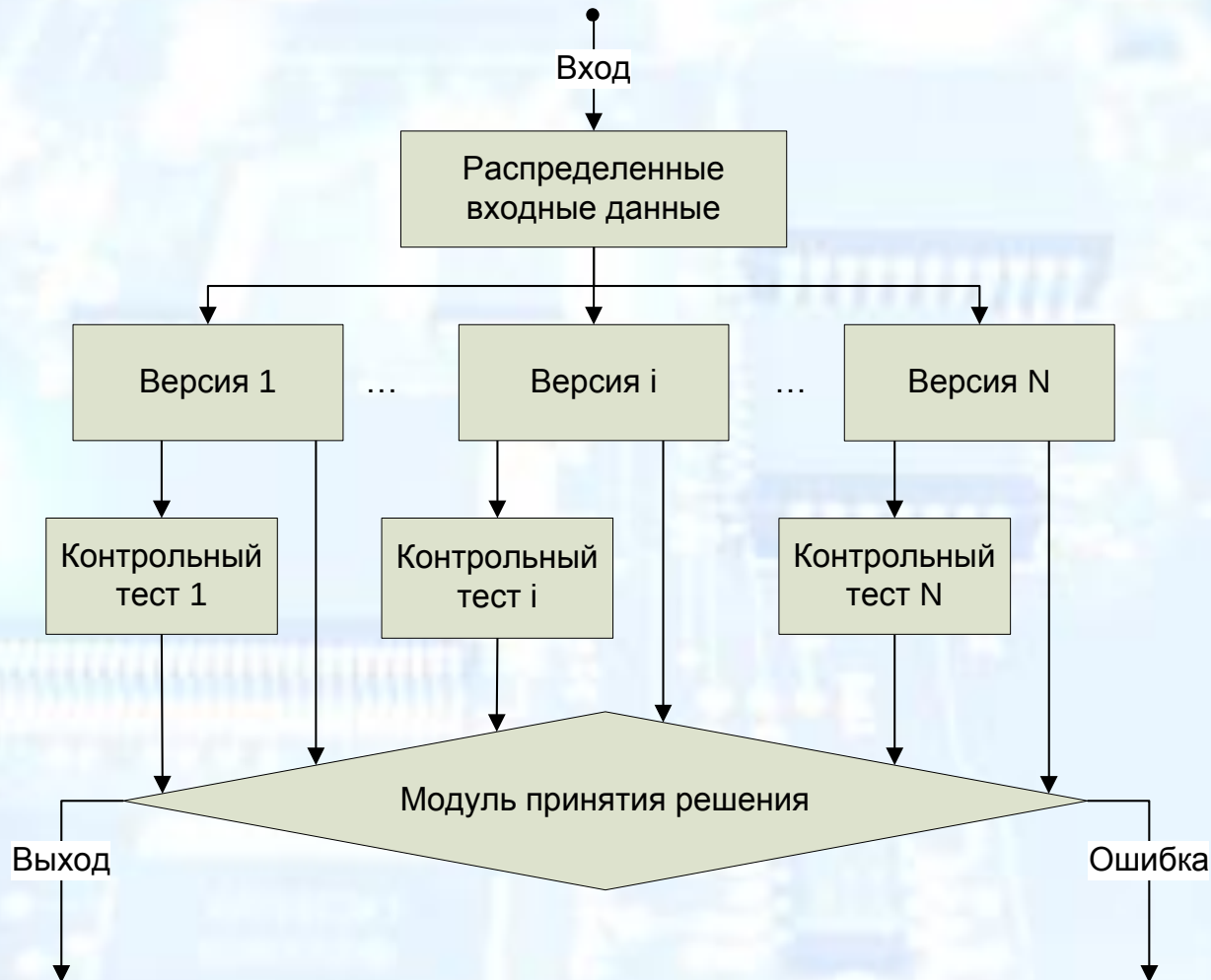
N-версионное программирование



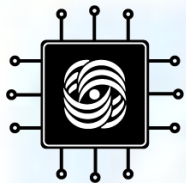
- Версии ПО функционально эквивалентны
- Различаются: алгоритмы, методы разработки, языки программирования, группы разработчиков



N-самотестируемое программирование

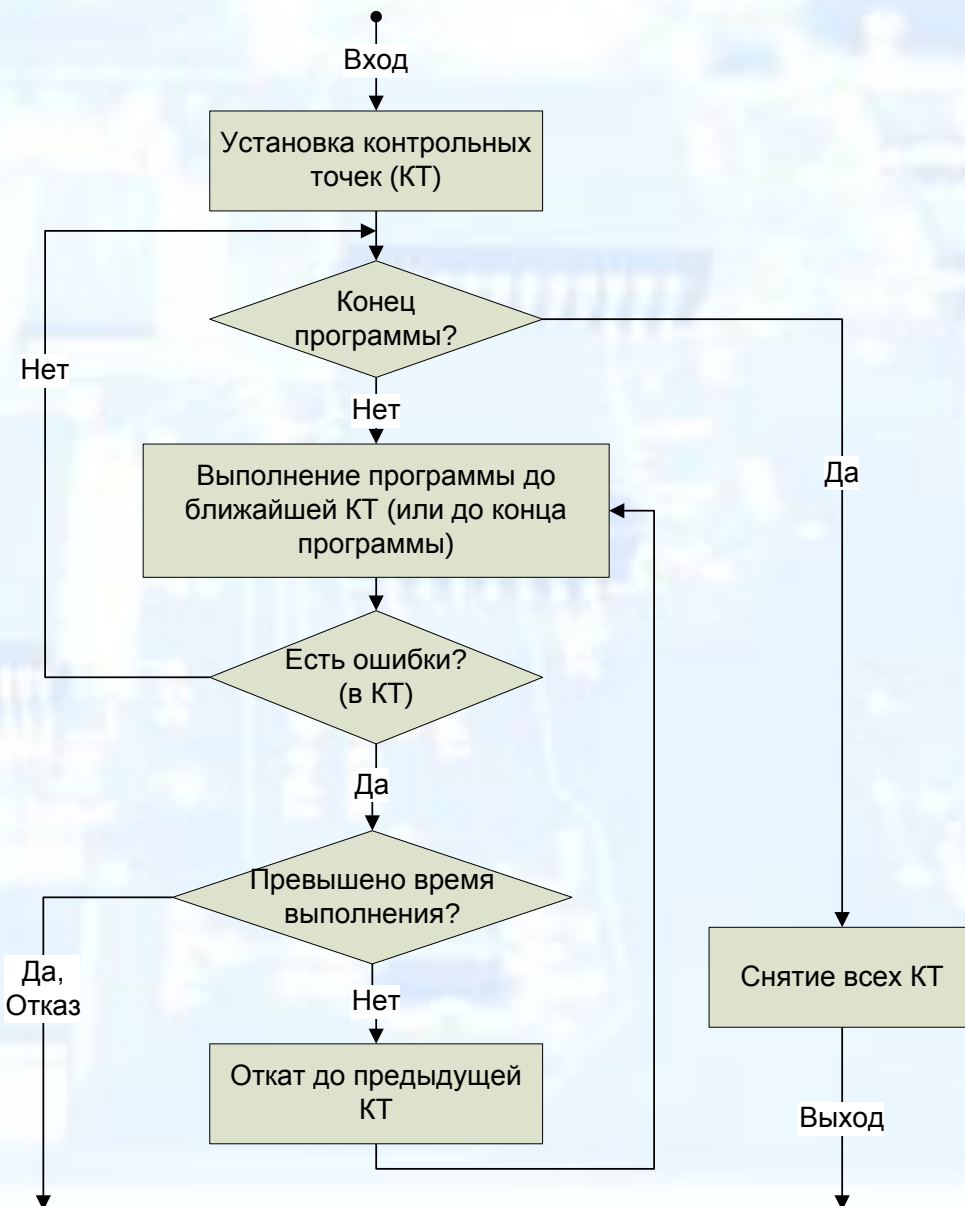


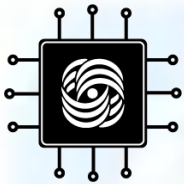
- Версии ПО запускаются последовательно (до первого результата, прошедшего контрольный тест) или параллельно (голосование среди успешных результатов)



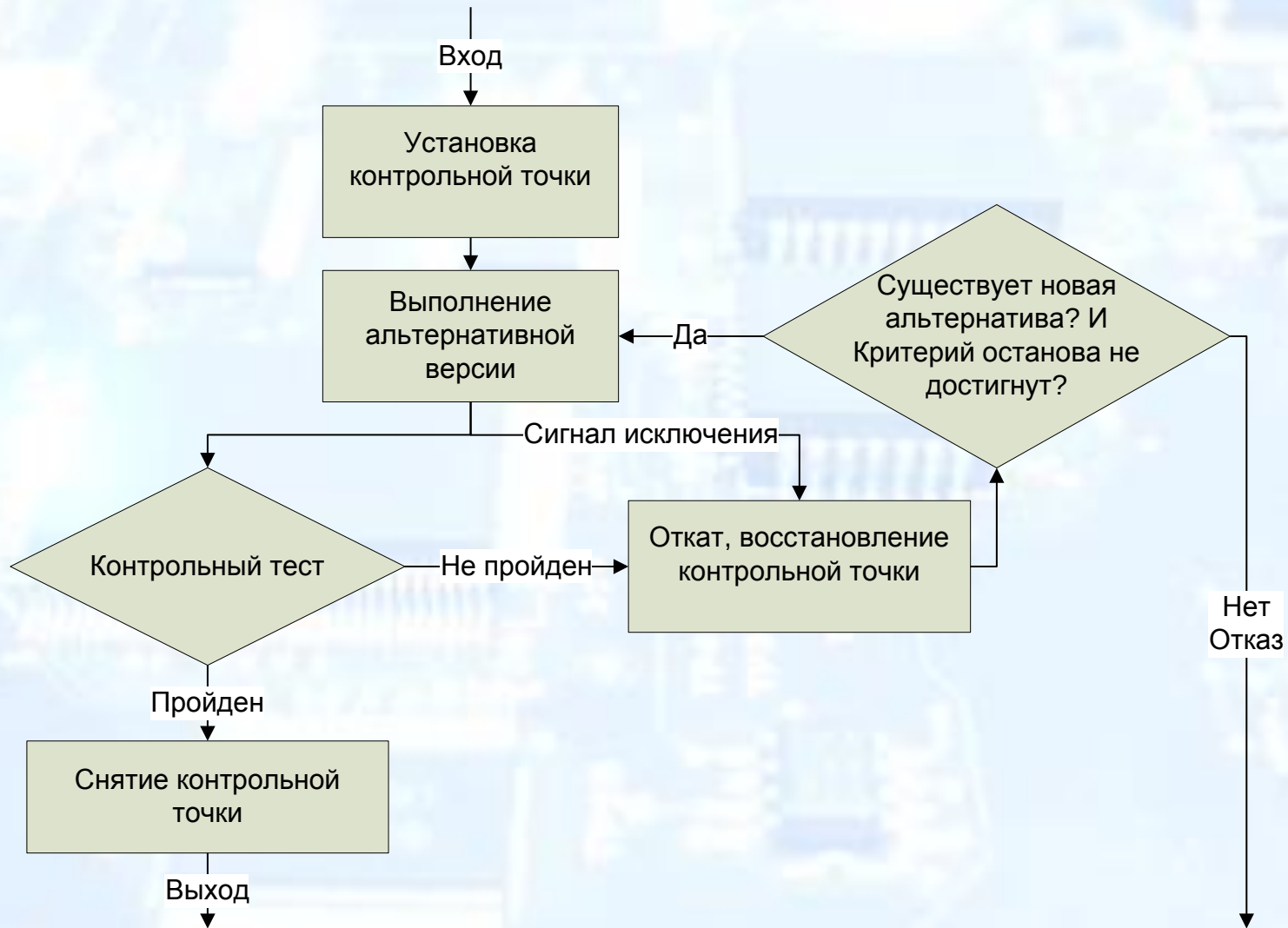
Контрольные точки

- Борьба только со случайными «однократными» неисправностями (инверсия бита памяти и т.п.)
- Не помогает от «постоянных» неисправностей, в т.ч. в ПО
- Затраты ресурсов на сохранение состояния в КТ, сложность механизмов поддержки сохранения

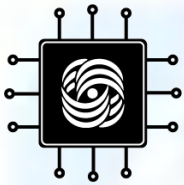




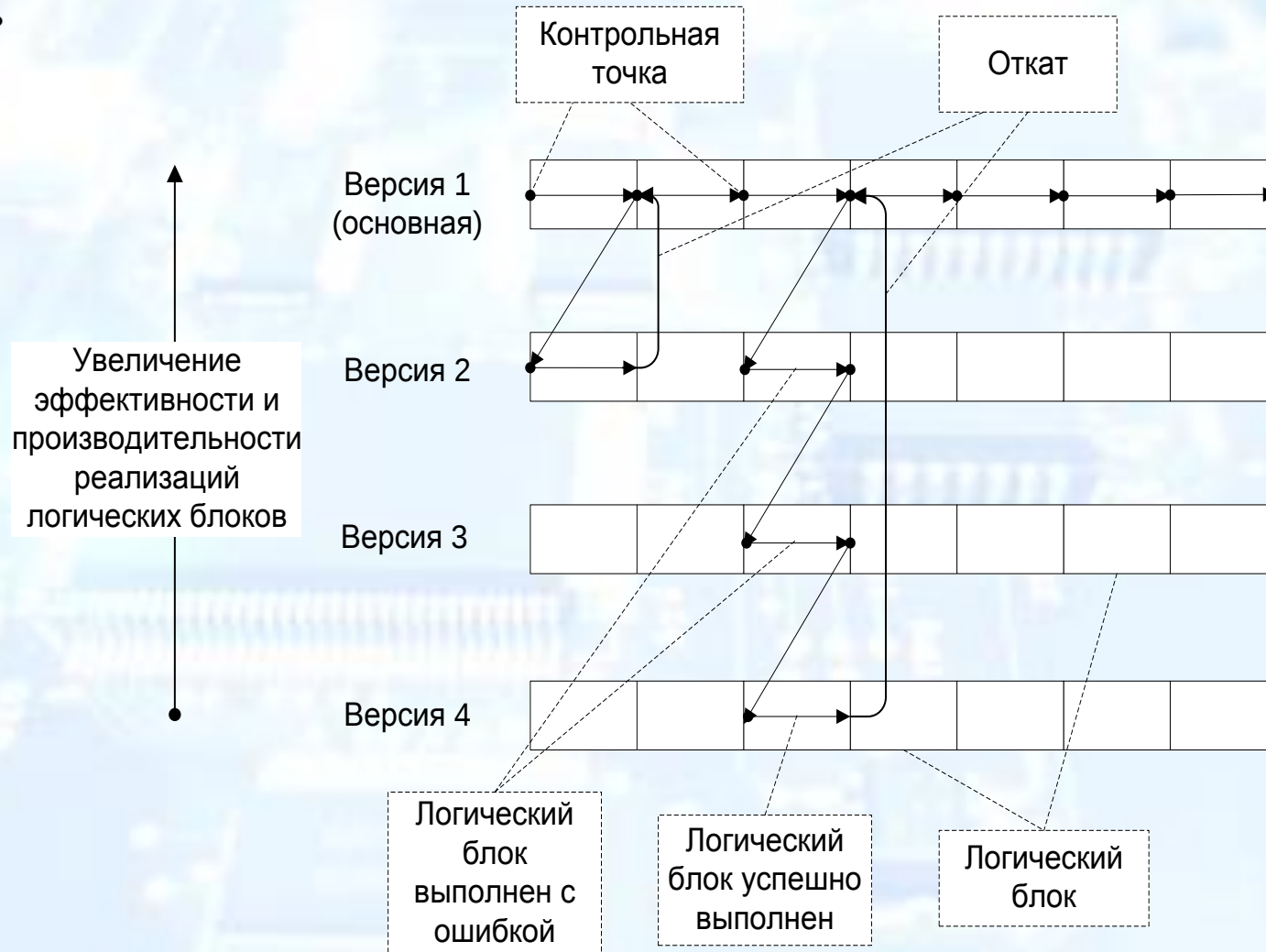
Восстановление блоками



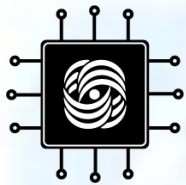
- Сочетает N-версионное программирование и контрольные точки



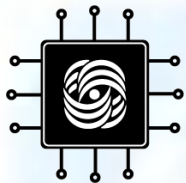
Восстановление блоками



- Перебор блоков по убыванию «скорости»
- WCET оценивается по наихудшему пути...

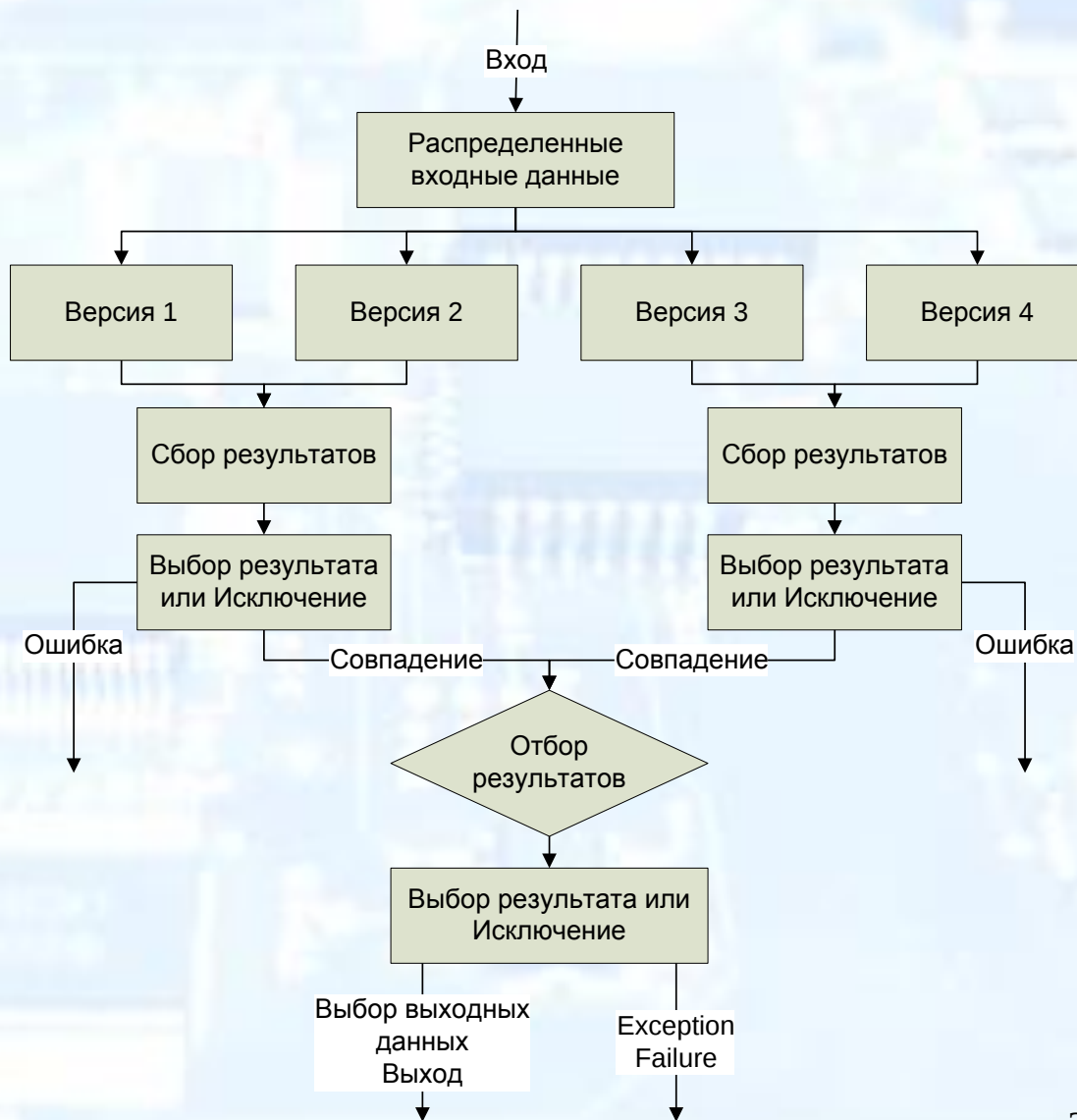


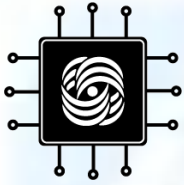
Программно-аппаратные МОО



N-самоконтролируемое программирование

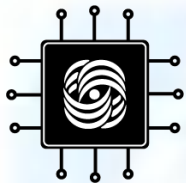
- Борьба с аппаратными и программными ошибками
- ПО в каждой паре: одна версия + контрольный тест, или более одной версии + алгоритм выбора



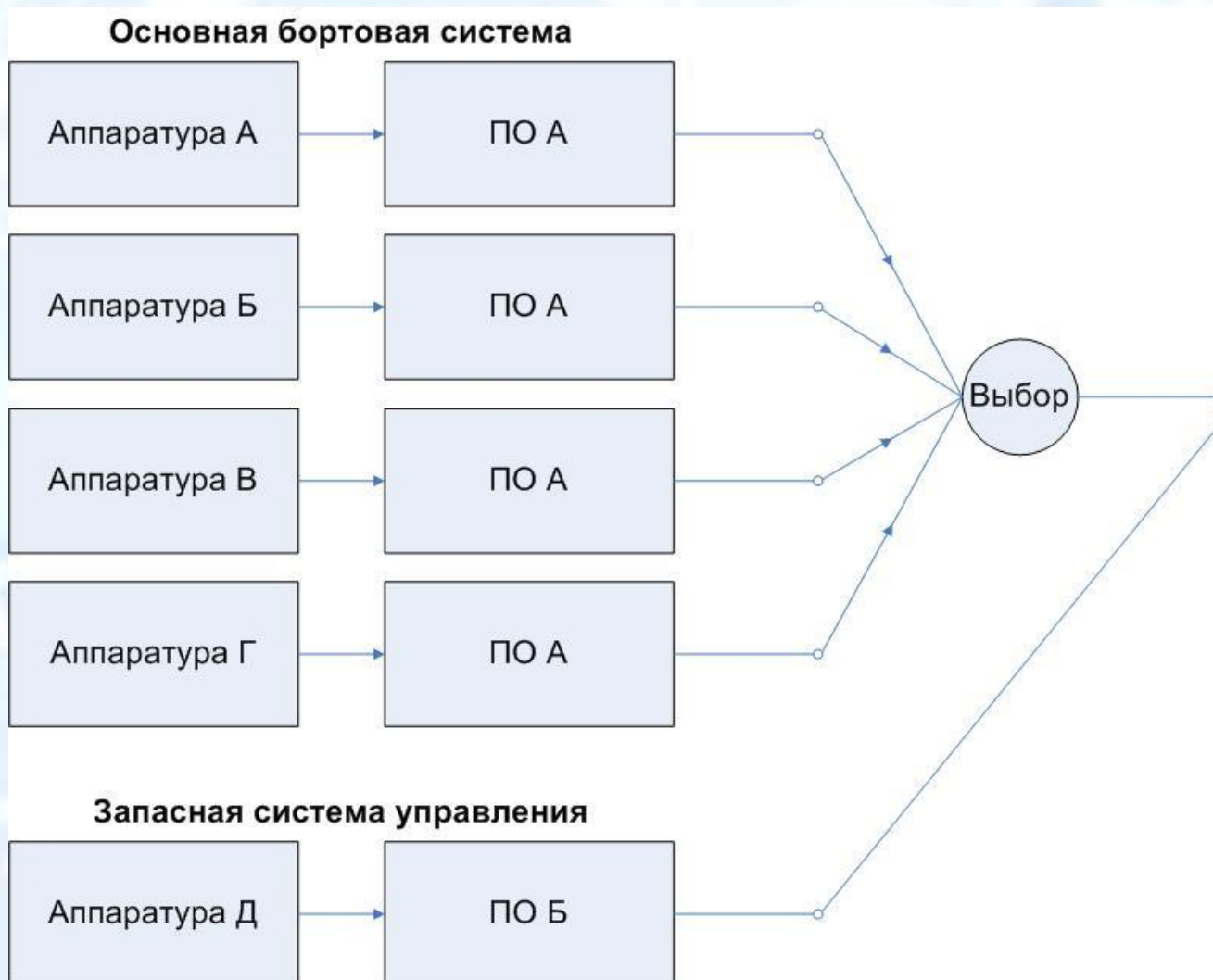


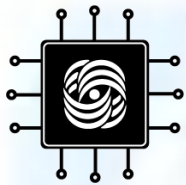
Активное резервирование + N-версионное программирование

- На каждом аппаратном компоненте выполняется своя версия программы
- Все компоненты активны
- Борьба с программными и аппаратными неисправностями

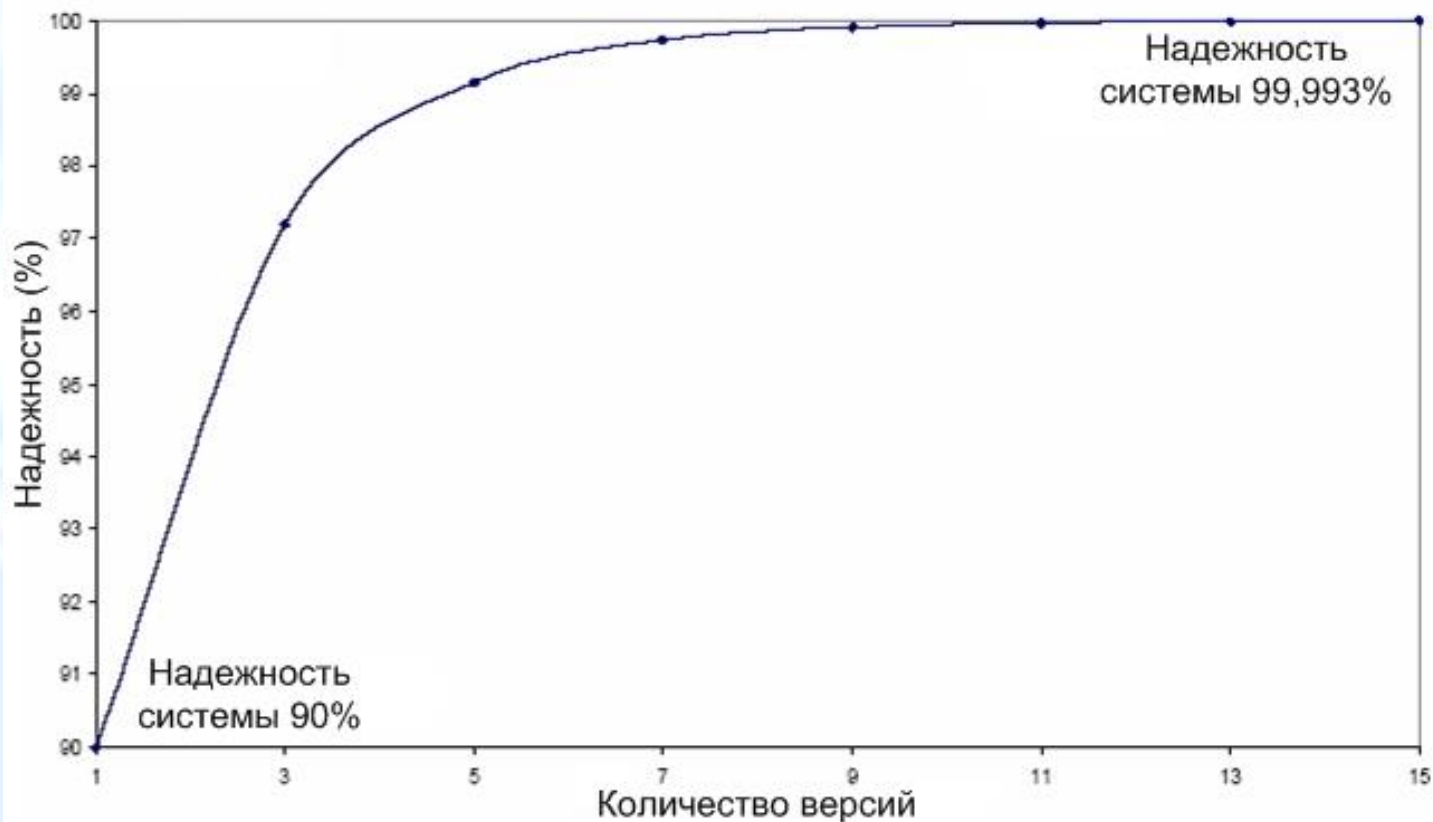


Космический челнок





Зависимость надежности от количества версий



Принципы проектирования безопасных и отказоустойчивых систем

1. Требования отказоустойчивости и безопасности должны быть неотъемлемой частью спецификации системы, а для некоторых систем – основой для процесса проектирования.
2. Ожидаемые виды отказов и частоты их возникновения должны быть определены в начале процесса проектирования.
3. Должны быть определены области локализации неисправностей в системе (fault containment regions). Последствия неисправности в одной из таких областей не должны распространяться на другие области.

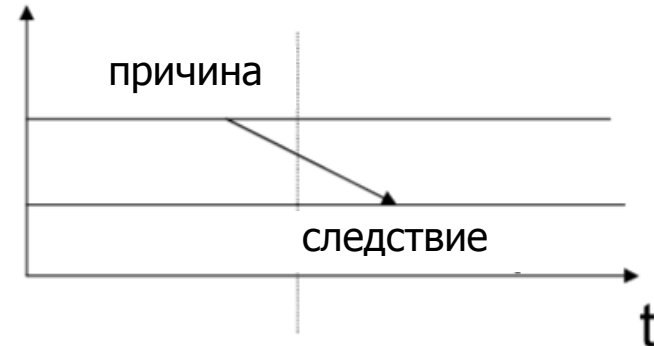


Салон
цел

Двигатель и
электронику
не жалко

Принципы проектирования безопасных и отказоустойчивых систем

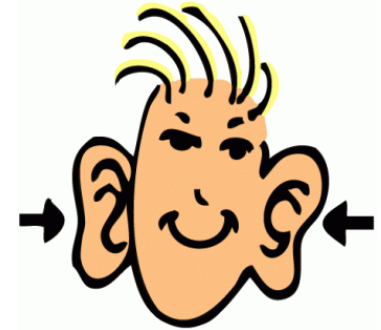
4. Понятие времени и состояния системы должны быть строго определены. В противном случае затруднительно отличить первичную неисправность от ее последствий.
5. Необходимо четко определить интерфейсы, чтобы скрыть внутреннее устройство компонентов системы.
6. Необходимо обеспечить независимость возникновения неисправностей в различных компонентах.



Две независимые
тормозные
системы

Принципы проектирования безопасных и отказоустойчивых систем

7. Компонент системы должен считать себя корректно функционирующим, пока два или более других компонента не примут противоположную точку зрения.
8. Механизмы обеспечения отказоустойчивости должны быть устроены так, чтобы не усложнять анализ поведения системы. МОО должны быть отделены от основной функциональности системы.
9. Возможность диагностирования должна быть заложена в систему на этапе проектирования. В частности, должна обеспечиваться возможность выявления скрытых неисправностей, не проявляющихся в поведении системы.



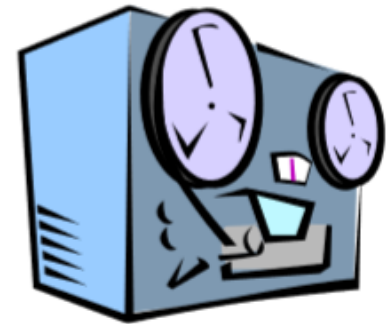
Принципы проектирования безопасных и отказоустойчивых систем

10. Интерфейс с оператором должен быть интуитивно понятным и устойчивым к ошибкам. Безопасность должна обеспечиваться несмотря на ошибки человека-оператора.



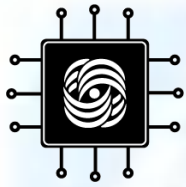
Подушка безопасности

11. Любая аномалия функционирования системы должна быть зарегистрирована. Некоторые аномалии не обнаруживаются на уровне интерфейсов между компонентами. Регистрация должна фиксировать в т.ч. внутренние аномалии, в противном случае они могут быть замаскированы в результате работы МОО.



12. Система должна реализовывать стратегию «никогда не сдавайся», гарантируя заданный минимальный уровень обслуживания. Полный выход из строя крайне нежелателен.



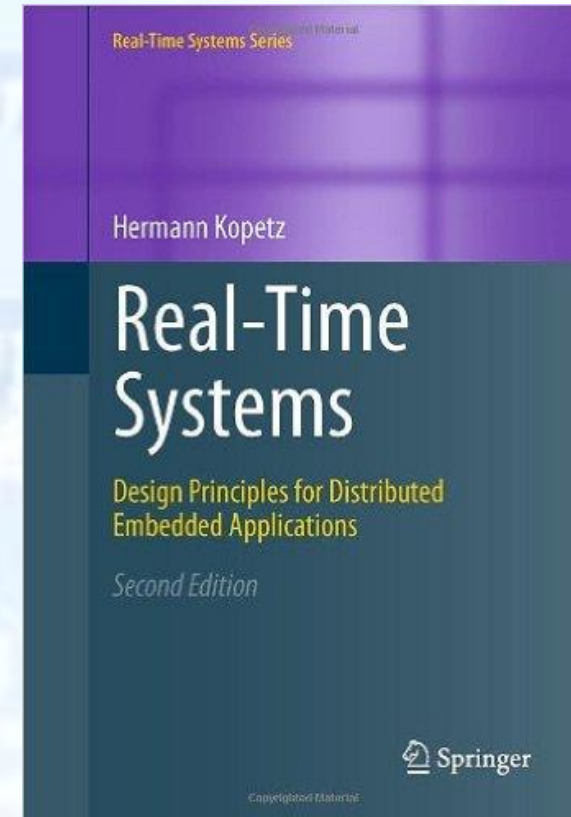


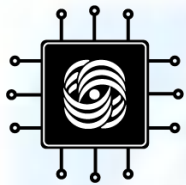
Принципы проектирования безопасных и отказоустойчивых систем

Источник:

“Real-Time Systems: Design Principles for Distributed Embedded Applications”
(Real-Time Systems Series),
H. Kopetz, Springer, 2011

https://vowi.fsinf.at/images/temp/2/2c/20110606133809!TU_Wien-Echtzeitsysteme_VO_%28Kopetz%29_-_TU_Wien-Echtzeitsysteme_VO_%28Kopetz%29_-_TU_Wien-Echtzeitsysteme_VO_%28Kopetz%29_-_Real_Time_Systems_-_Design_Principles_for_Distributed_Embedded_Applications_--_Hermann_Kopetz_--_2._Edition.pdf





Спасибо за внимание!