



**MICROCHIP 2010**

*MASTERS Conference*

# Проектирование отказоустойчивых устройств.

## Аппаратные и программные методы повышения надежности

# Цель

Получить представление о безопасных устройствах и обсудить основные аспекты необходимости разработки приложений относительно вопросов безопасности.

- | **Определение ключевых элементов в безопасном устройстве**
- | **Реализация ключевых элементов в приборах, которые должны быть безопасными**

# План

- | **Откуда возникают требования по безопасности?**
- | **Базовые понятия об отказах**
- | **Программные задачи с примером реализации «class B»**
- | **Соображения по аппаратной части**
- | **Соображения по программной реализации**

# КОМИТЕТЫ... КОМИССИИ...

**Существует масса Комитетов. Все они специализируются на определенных приложениях, задачах и регионах.**

**Некоторые из них:**

- **RTCA (US Radio Technical Commission for Aeronautics)**
- **DoD (Department of Defense)**
- **DGQ (Deutsche Gesellschaft für Qualität)**
- **Nasa (National Aeronautic and Space Association)**
- **VDE (Verband der Elektrotechnik Elektronik Informationstechnik e.V.)**
- **Misra (Motor Industry Software Reliability Association)**
- **IEC (International Electrotechnical Commission)**
- **DIN (Deutsche Industrie Norm)**

**... и каждые создают специфические стандарты и положения.**

# Стандарты, Регламенты, Положения

## Некоторые из стандартов:

- IEC61508
- DIN V 19250
- DIN V VDE 0801
- IEC61511
- IEC61132
- ISA TR84.02
- RTCA DO 178B
- IEC60335
- IEC60730

**...и много больше.**

# Почему так много стандартов по безопасности?

Возможно что вы хотите развлечься на «тарзанке» или сесть за руль автомобиля или лететь на самолете. При возникновении аварии возможна опасность для жизни, окружающей среды или экономики.

Вопрос в том, в какой степени будет нанесен ущерб другим людям или окружающей среде при пользовании тем или иным оборудованием? При прыжках на «тарзанке» может пострадать только один человек. При вождении автомобиля - другие автомобили или другие люди. Падение самолета может иметь катастрофические последствия.

Мы видим, что существуют разные ситуации - так что имеет смысл иметь различные конкретные правила.

Пример:

RTCA DO 178B для авионики

IEC 60730 для домашней автоматики

EN 60335-1 для бытовых приборов

ISO 26262 для автомобильных приложений



# MICROCHIP 2010

---

## MASTERS Conference

# Общие вопросы проектирования.

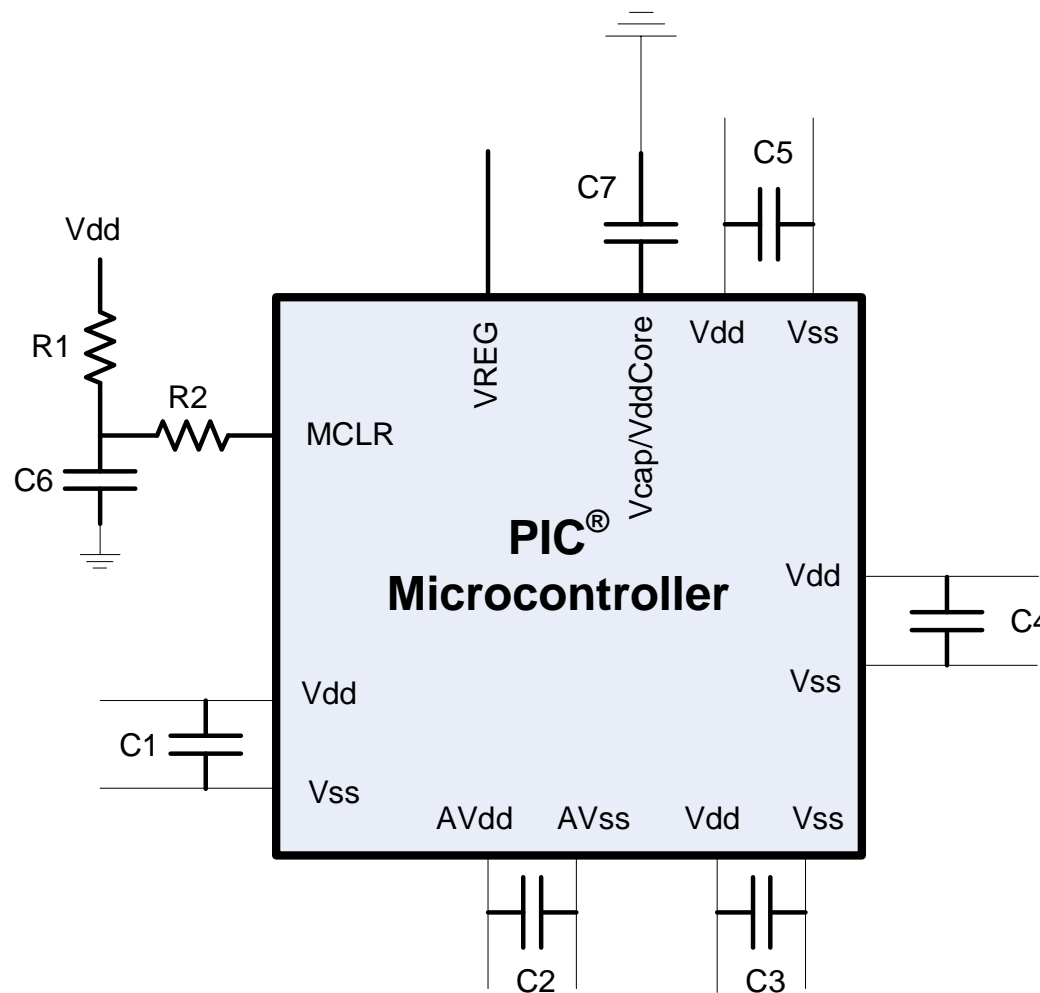
# EMS/ESD

# Общие вопросы проектирования. EMS/ESD

- | **Порты В/В**
- | **Входы прерываний**
- | **Вход сброса**
- | **Цепи питания**
- | **Генератор**
- | **Brown Out Reset (BOR)**
- | **Watch Dog Timer (WDT)**



# Начальные сведения

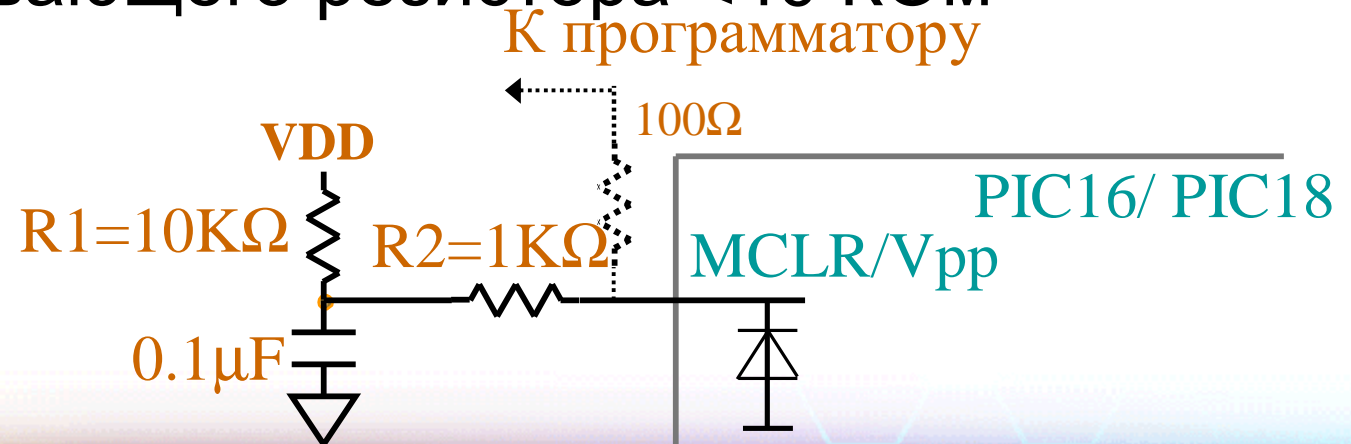


- | Конденсаторы на каждом выводе питания
  - | 0.1uF || 0.01uF керамика
  - | Располагать как можно ближе к выводам
- | Если разрешен внутренний Vreg
  - | Конденсатор регулятора (Vcap)
  - | 10uF 16V керамика или тантал
- | R-C-R схема сброса

# Микроконтроллерная схема

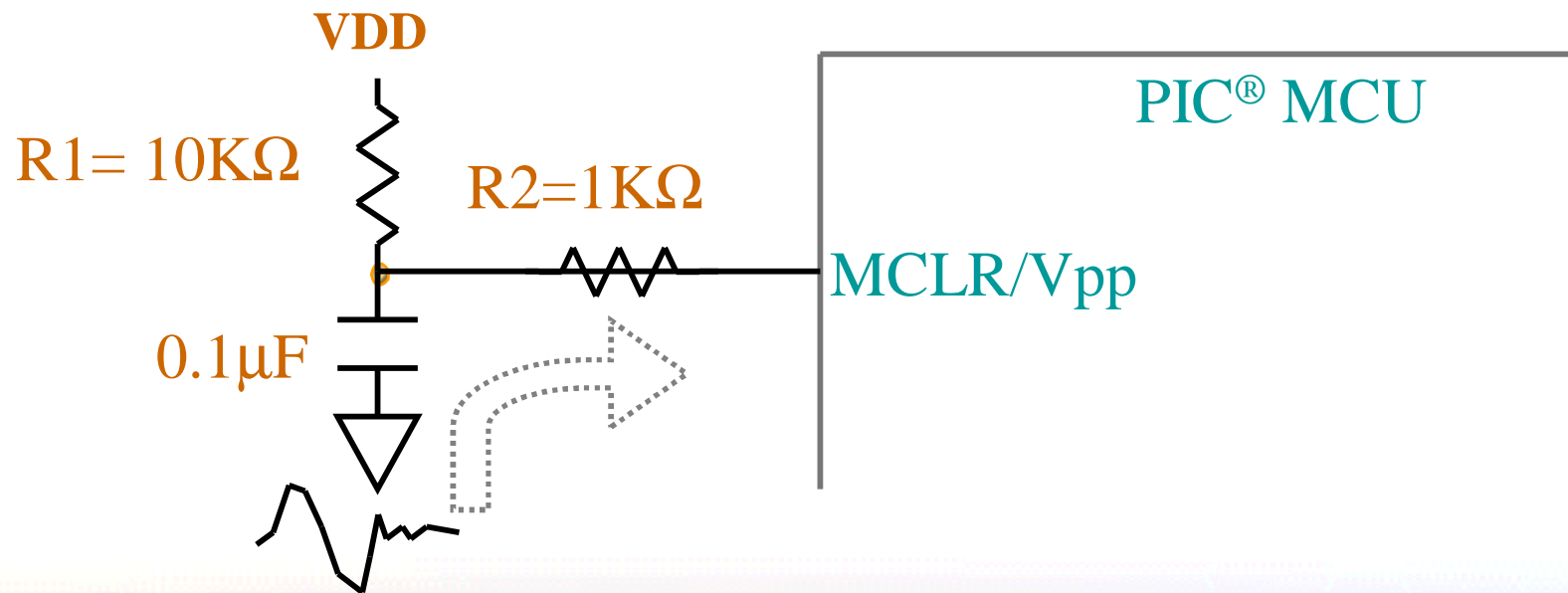
## Вход сброса

- Последовательный резистор ограничивает ток через вход MCLR при воздействии ESD или EOS
- Конденсатор уменьшает ВЧ шум
- Рекомендуемое значение сопротивления подтягивающего резистора  $< 40 \text{ КОм}$



# Микроконтроллерная схема

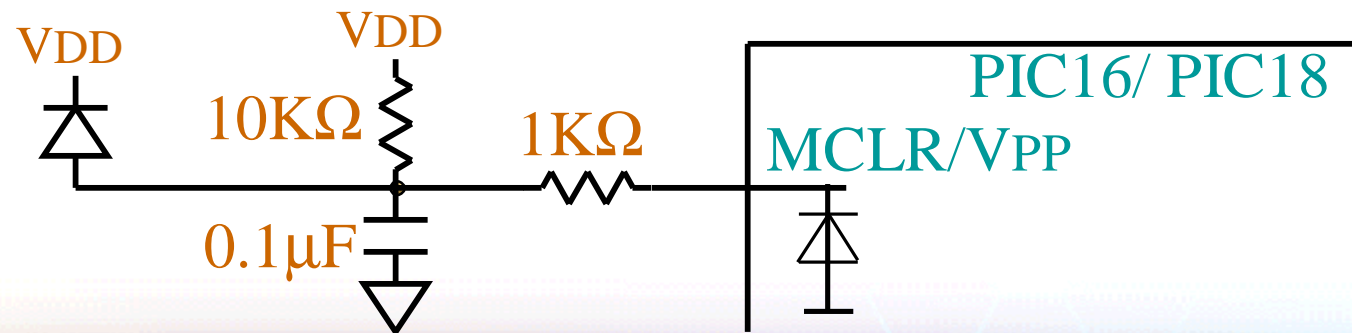
- | Вход сброса
  - | Зачем нужен R2?



# Микроконтроллерная схема

## Вход сброса

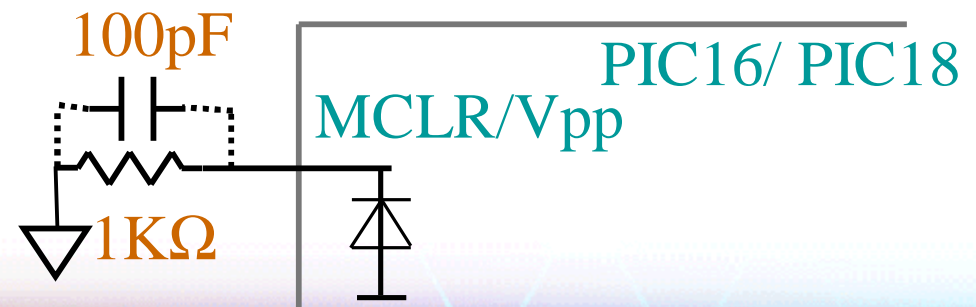
- Вход MCLR нужен также для программирования (VPP)
- Если не требуется внутрисхемное программирование, то нужно добавить диод для дополнительной ESD защиты



# Микроконтроллерная схема

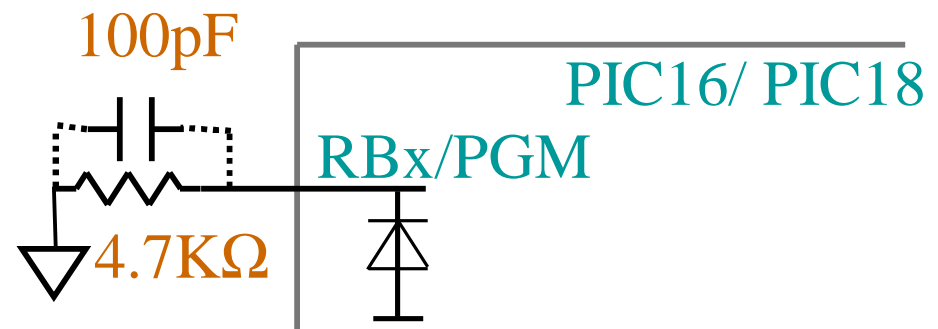
## Вход сброса

- В некоторых контроллерах можно использовать внутреннюю цепь MCLR
- Если функционально вход MCLR не нужен – отключите его.
- Если MCLR отключен, тогда...

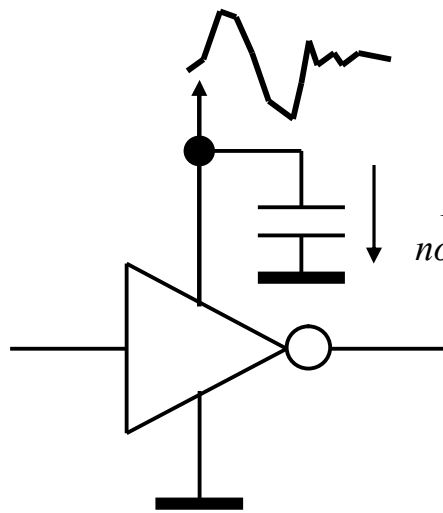


# Микроконтроллерная схема

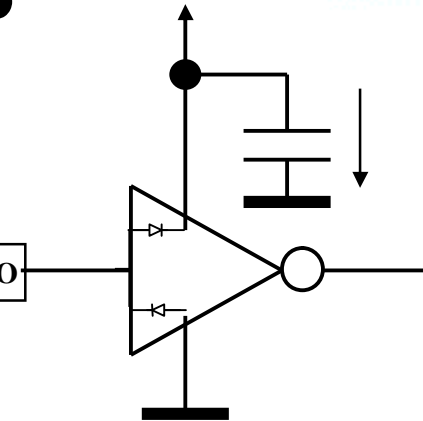
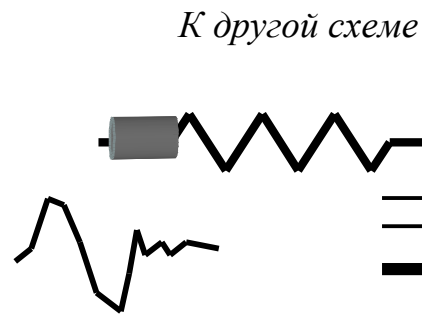
- | Низковольтное программирование (PGM pin)
  - | Если используется низковольтное программирование (LVP) то...



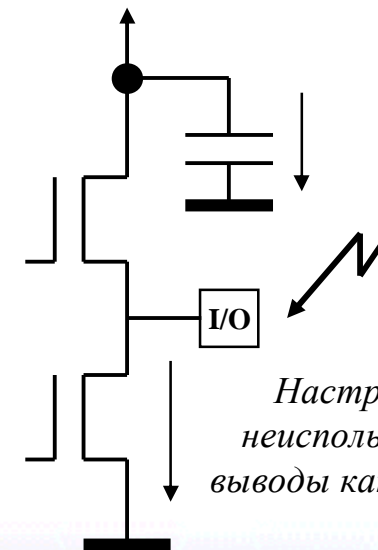
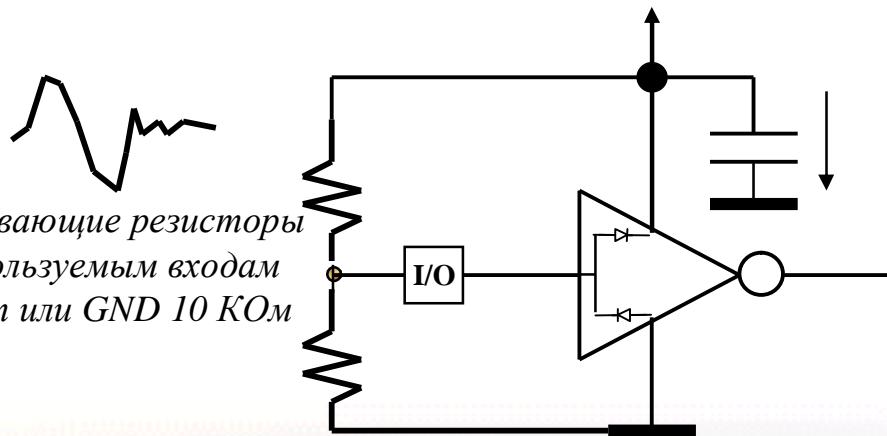
# Микроконтроллерная схема порты В/В



*Развязка  
по питанию*



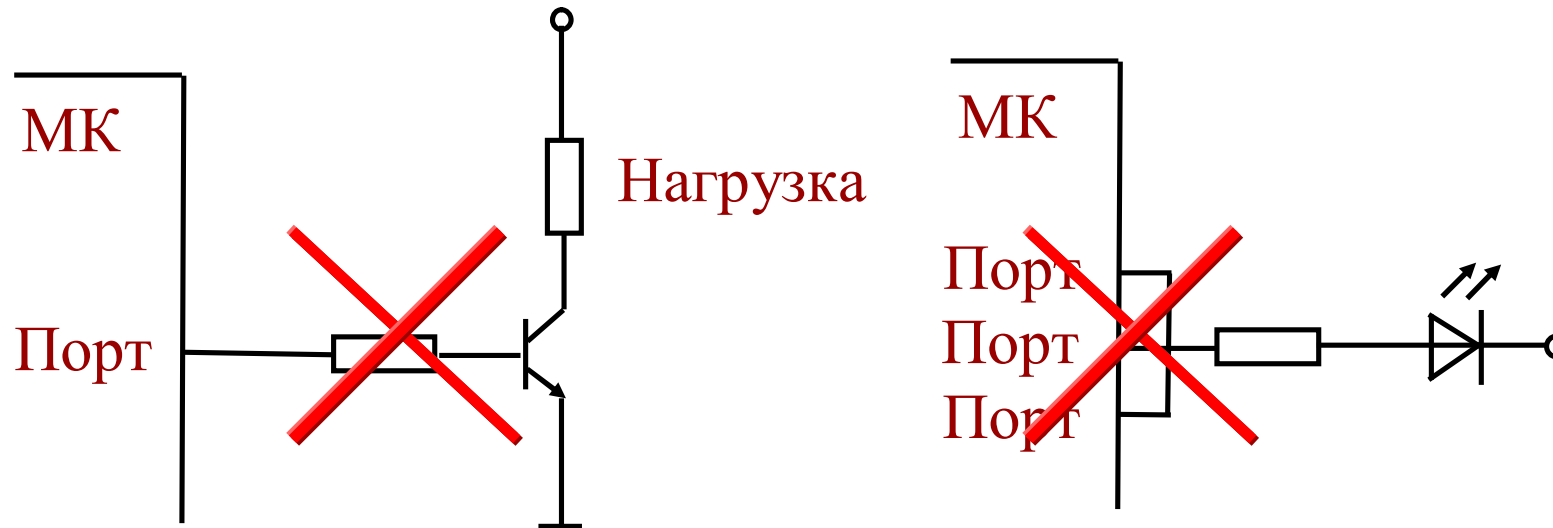
*Подтягивающие резисторы  
к неиспользуемым входам  
к +Uпит или GND 10 КОм*



*Настроить  
неиспользуемые  
выводы как выход 0*

# Микроконтроллерная схема порты В/В

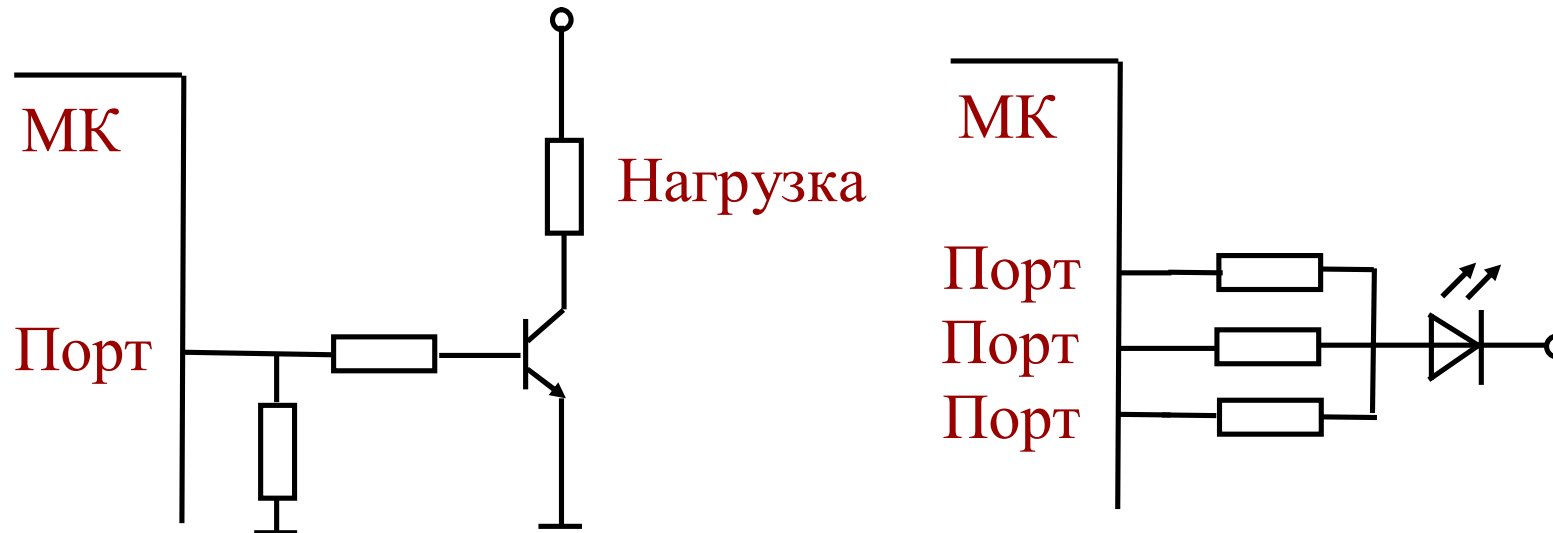
## Остерегайтесь «непредсказуемых» ситуаций





# Микроконтроллерная схема порты В/В

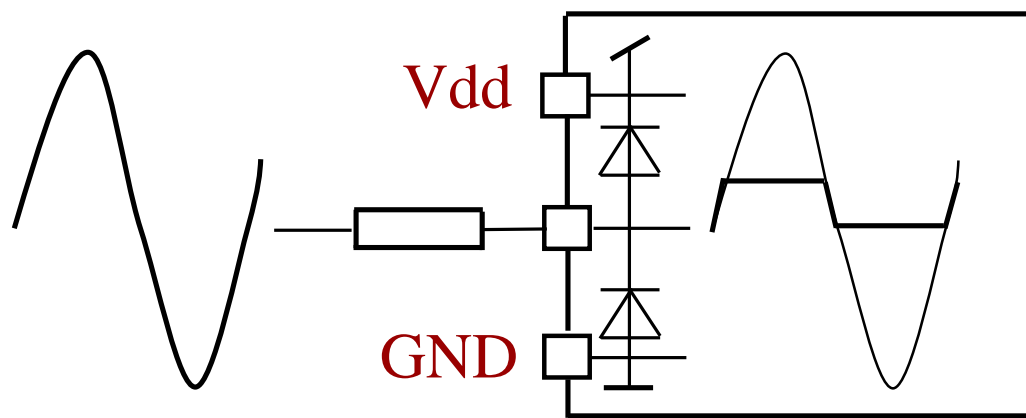
## Остерегайтесь «непредсказуемых» ситуаций



# Микроконтроллерная схема порты В/В

## Цифровые порты В/В

- Входные ESD диоды
- Максимальное напряжение на входе  $V_{dd} + 0.6V$ . Ток защитных диодов 20мА



AN521

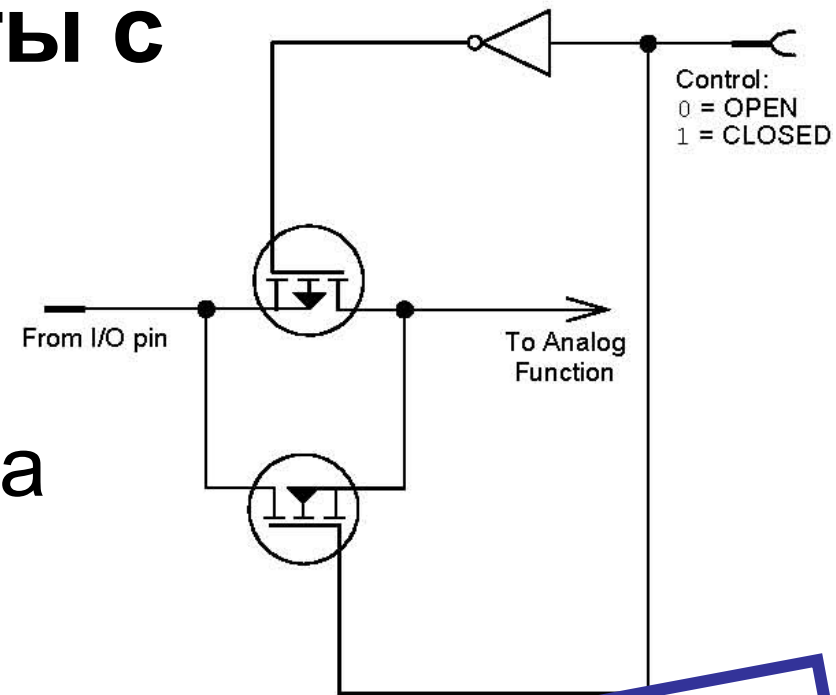
$R=5M$

$I_{peak} = 64\mu A$

# Микроконтроллерная схема порты с аналоговым буфером

## Аналоговые порты с входным аналоговым буфером

- При напряжении на входе больше  $V_{dd}+0.4V$  верхний транзистор открывается

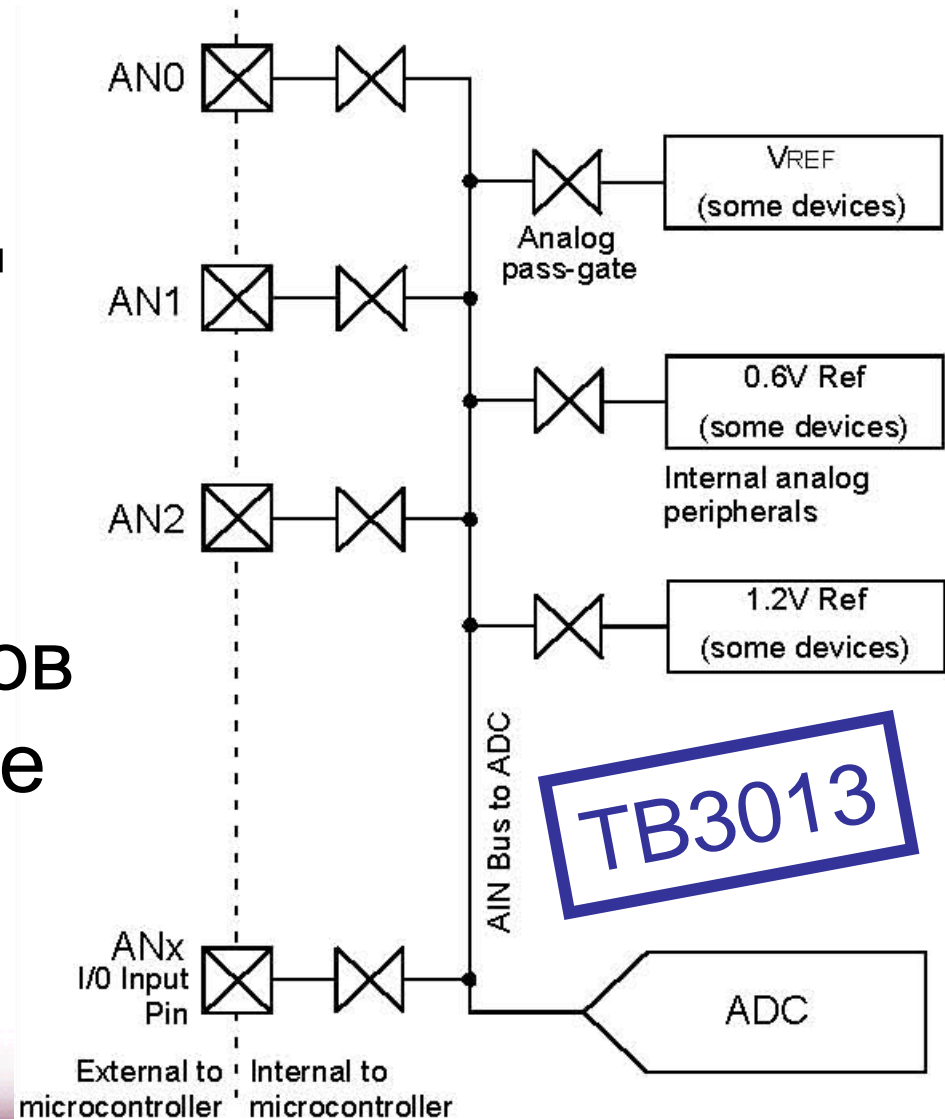


Using the ESD Parasitic Diodes  
on Mixed Signal Microcontrollers  
**TB3013**

# Микроконтроллерная схема порты с аналоговым буфером

## Физически АЦП имеет один вход

- | Высокое напряжение на любом из аналоговых входов создаст смещение для АЦП



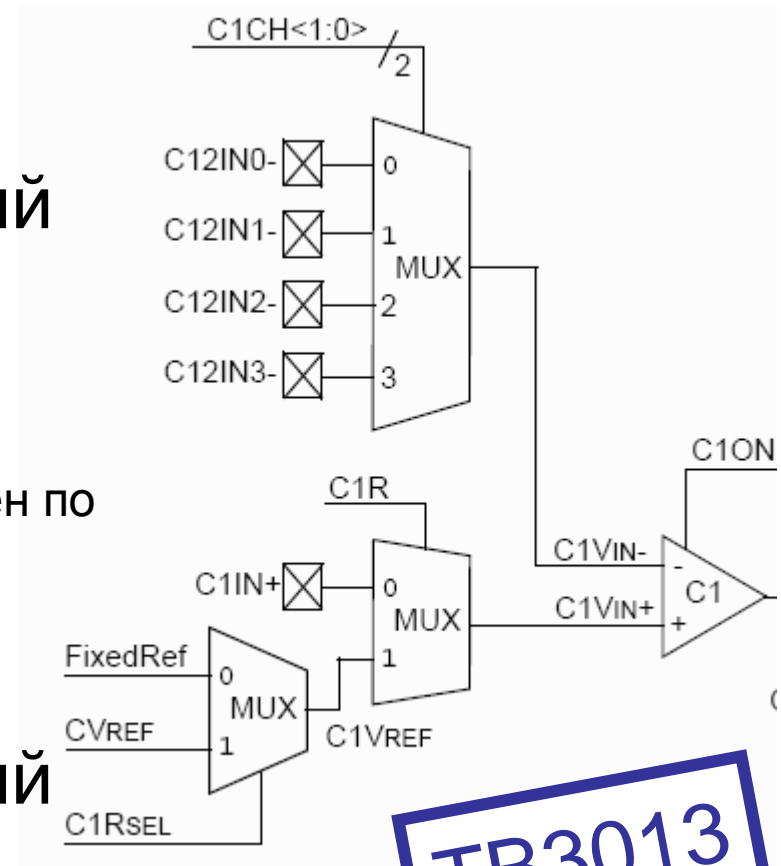
# Микроконтроллерная схема порты с аналоговым буфером

## Компаратор

- | Так же имеет аналоговый буфер
- | Источник опорного напряжения 0.6В (подключен по умолчанию, если есть)

## Драйвер ЖКИ

- | Так же имеет аналоговый буфер для соединения VLCD и сегментов/общих выводов



# Микроконтроллерная схема порты с аналоговым буфером

## Внутренний генератор

- Стабилизируется источником опорного напряжения 0.6В
- Для некоторых контроллеров этот же источник 0.6В (или 1.2В) может быть подключен к АЦП
- Если 0.6В подключен к входу АЦП и есть перенапряжение на других каналах  $\hat{e}$  генератор остановится.

ТВ3013

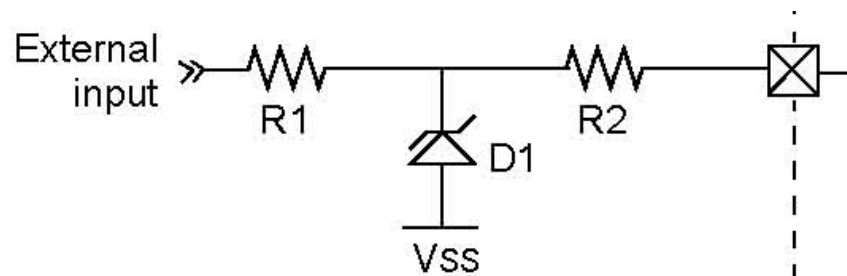
# Микроконтроллерная схема порты с аналоговым буфером

## Защита. Выбор портов В/В.

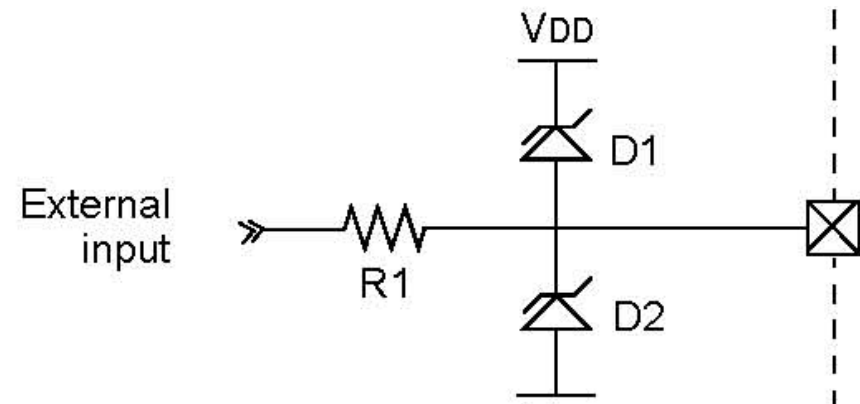
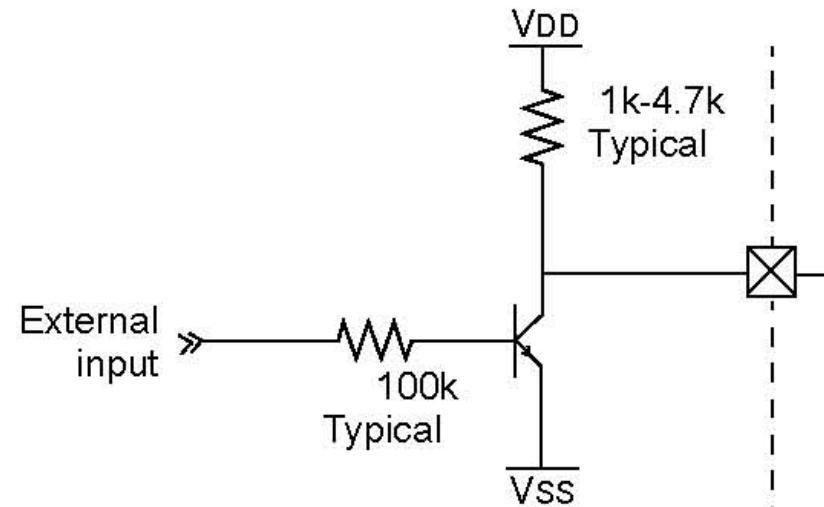
- Цифровые порты не имеют аналогового буфера
  - Допускают перенапряжение
  - Но будущие продукты на тех же выводах могут иметь аналоговые функции
- Никогда не допускайте перенапряжения на входе MCLR/Vpp

# Микроконтроллерная схема порты с аналоговым буфером

## Защита портов В/В.



ТВ3013





# Микроконтроллерная схема порты с аналоговым буфером

## Защита. Программный метод.

- Если есть вывод с перенапряжением и нельзя выбрать чисто цифровой вывод
  - На время измерения АЦП по одному из каналов «перенапряженный» вывод настроить как цифровой выход
  - После измерения настроить на аналоговый вход для измерения
  - используйте последовательный резистор для ограничения тока.

# Микроконтроллерная схема отрицательное напряжение

- | **Напряжение на входе ниже  $V_{ss}$** 
  - | Ток через подложку, отрицательное смещение.
  - | Диоды на кристалле становятся транзисторами и шунтируют ток на подложку.
  - | Вероятность сдвига частоты и POR
  - | При достаточном токе есть вероятность защелкивания è резкое увеличение тока è локальный перегрев è деградация.


# Микроконтроллерная схема отрицательное напряжение

- | **Напряжение на входе ниже  $V_{SS}$** 
  - | Чувствительность к отрицательным напряжениям увеличивается с увеличением температуры
  - | Защитить диодами или пересмотреть схему источника питания

# Микроконтроллерная схема

## Вход прерываний

- Прерывания по фронтам чувствительны к шумам

- Используйте прерывания по уровню или опрос входа внутри ISR 

- Используйте терминаторы линии для уменьшения переотражения сигналов

- Внимательно разводите проводники линий прерываний для уменьшения восприимчивости к помехам

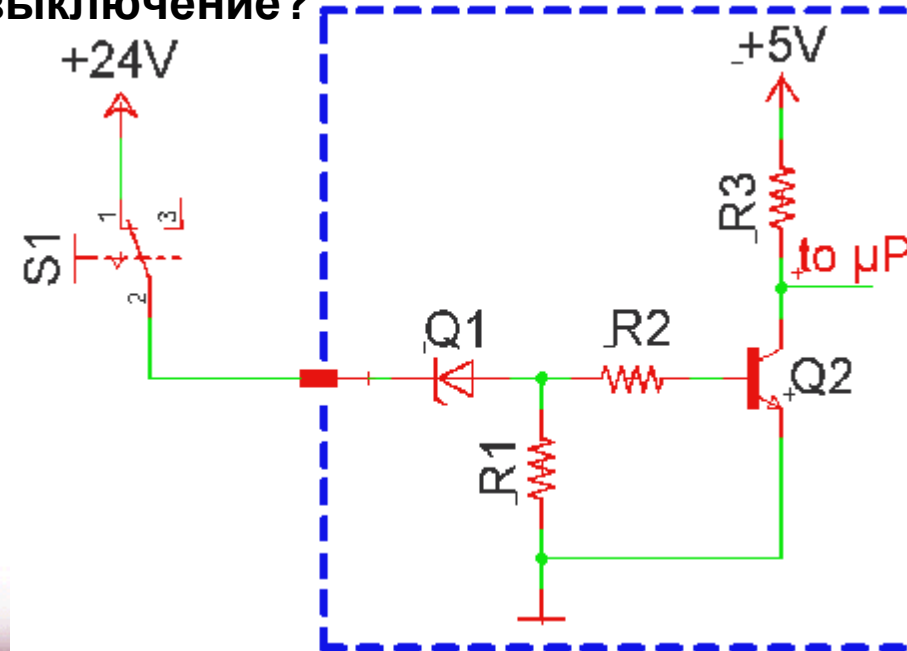
# Аппаратный анализ FMEA

- | FMEA означает Failure Modes and Effects Analysis
- | Бывают различные FMEA, например Дизайна, Системы, Схемы, Программы ...
- | FMEA это путь нахождения и описания проблемы – и ошибок!

Пример: Кнопка аварийного останова – сигнал для микроконтроллера.

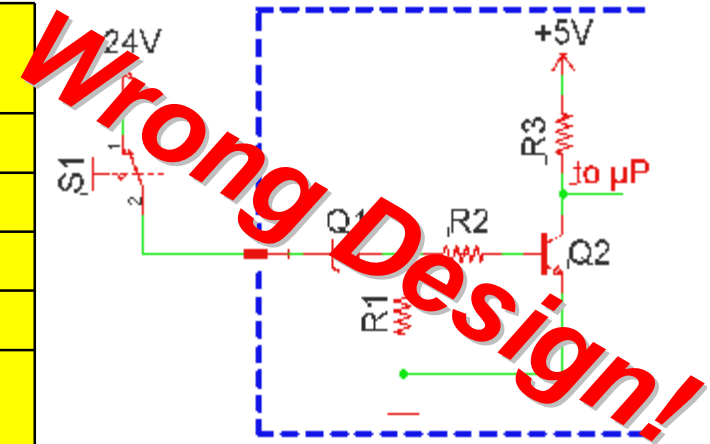
В опасной ситуации пользователь может немедленно остановить оборудование.

Эта схема обеспечит надежное выключение?



# Таблица FMEA схемы

Comp	pin	mode	Description
R2	1 or 2	open	safe mode, Application switch OFF
R2	1 & 2	short	disturb Q1 or Q2, Application switch OFF
R3	1 or 2	open	safe mode, Application switch OFF
R3	1 & 2	short	Application Always ON
R1	1 or 2	open	not detect, critical EMI noise sensity and Offset Voltage sensitivity depending on temperature and value of R3.
R1	1 & 2	short	safe mode, Application switch OFF
Q1	K or A	open	safe mode, Application switch OFF
Q1	K & A	short	not detect, Offset Voltage sensitivity
Q2	E	open	safe mode, Application switch OFF
Q2	B	open	safe mode, Application switch OFF
Q2	C	open	safe mode, Application switch OFF
Q2	E & B	short	safe mode, Application switch OFF
Q2	E & C	short	Application Always ON
Q2	B & C	short	Application Always ON



Этот пример с углеродными резисторами – только демонстрирует какой может быть FMEA.

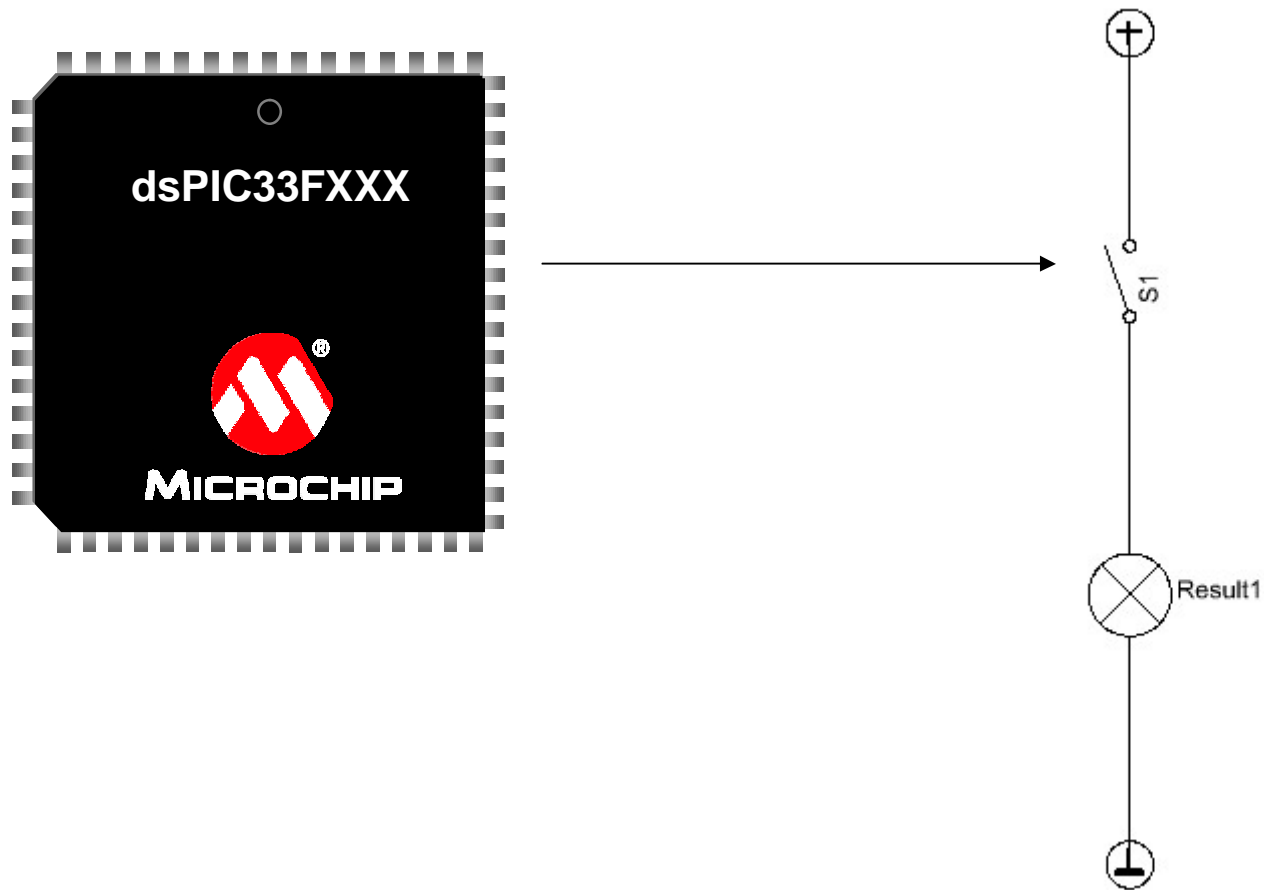
Для проволочных резисторов функция “КЗ” не тестируется.



**MICROCHIP** 2010  
*MASTERS Conference*

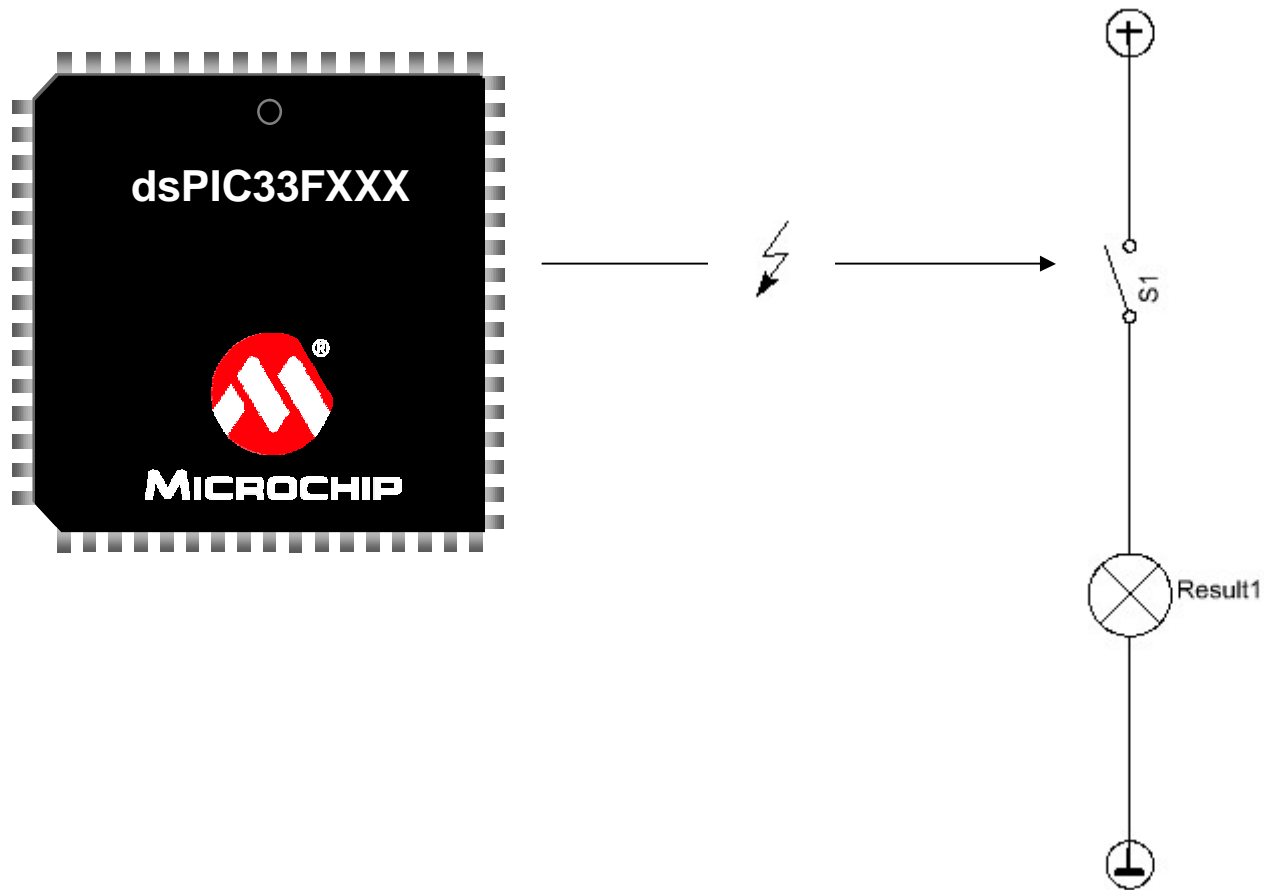
# Соображения по аппаратной реализации

# Как сделать это безопасным?

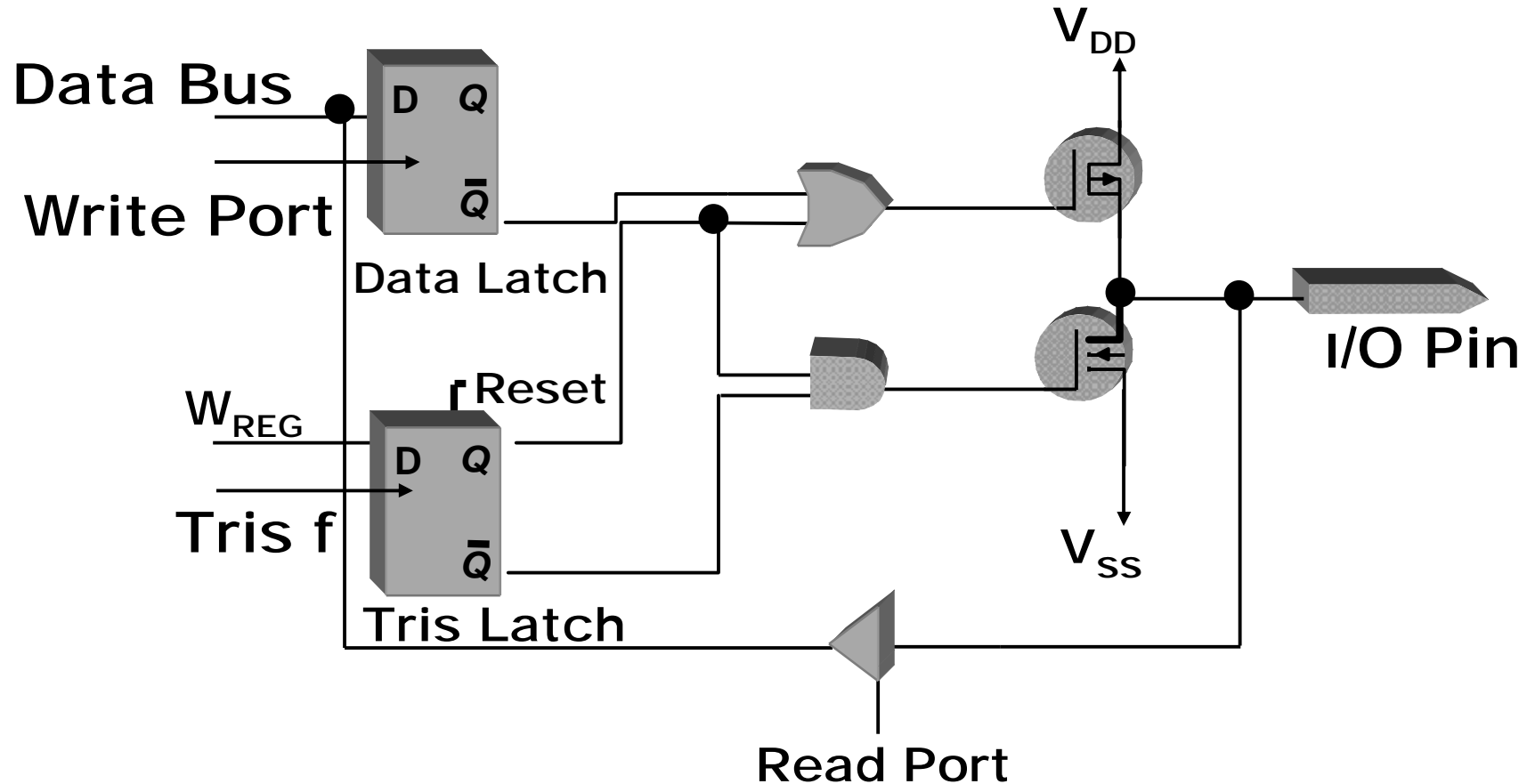




# Что если вывод “зависнет”?

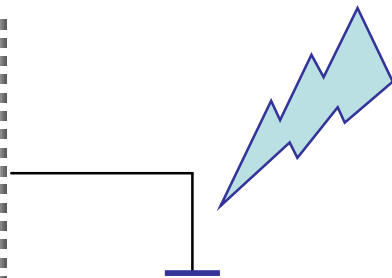
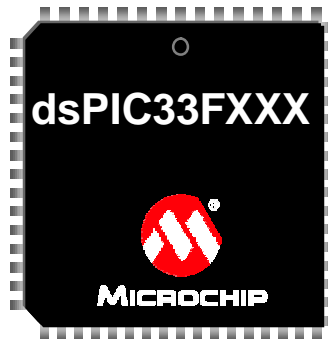


# Цифровые порты В/В дают метод для тестирования...



I/O выводы имеют защитные ESD диоды

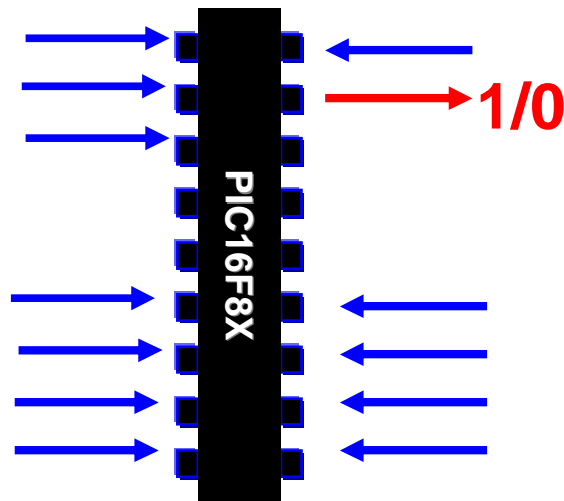
# “Stuck at” Error



Вы пишете “1”  
НО  
читаете “0”

Причина: ошибки монтажа, выход из строя внешних элементов

# Перекрестные ошибки замыкания в шине



1) Запишите “1” на один вывод.

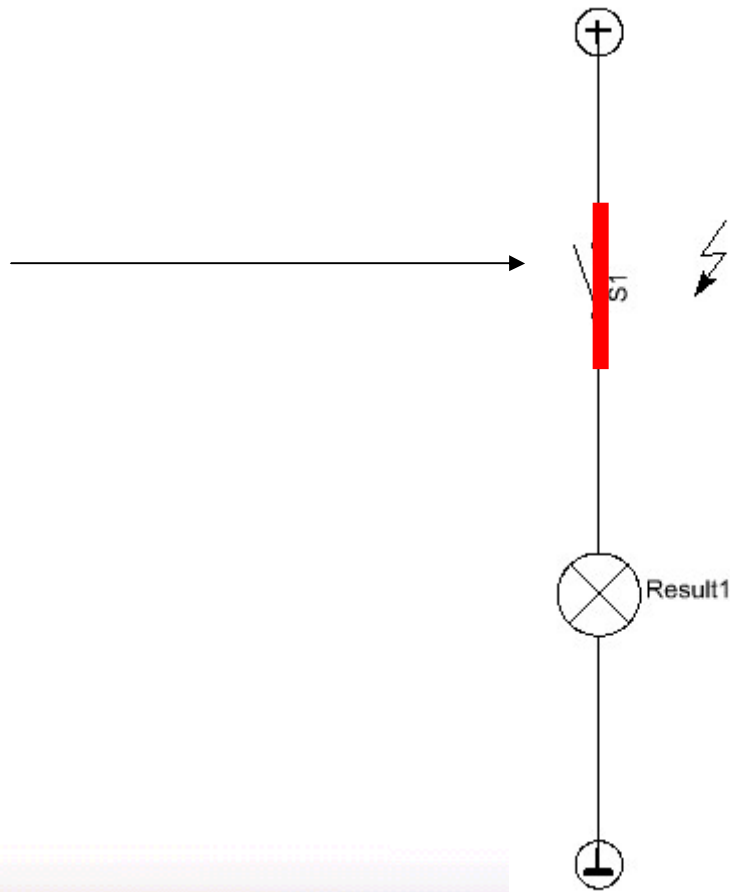
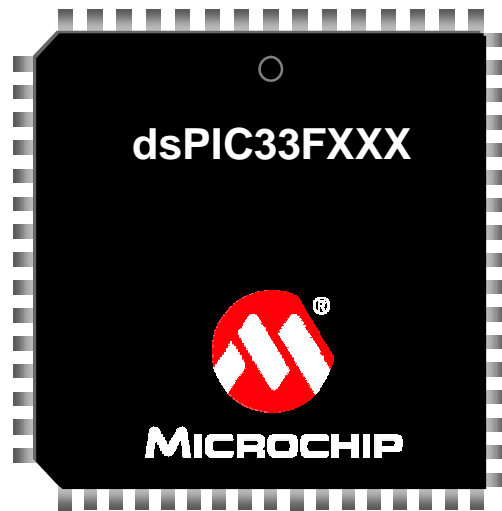
2) Прочитайте все остальные порты.

3) Запишите “0” на тот же вывод

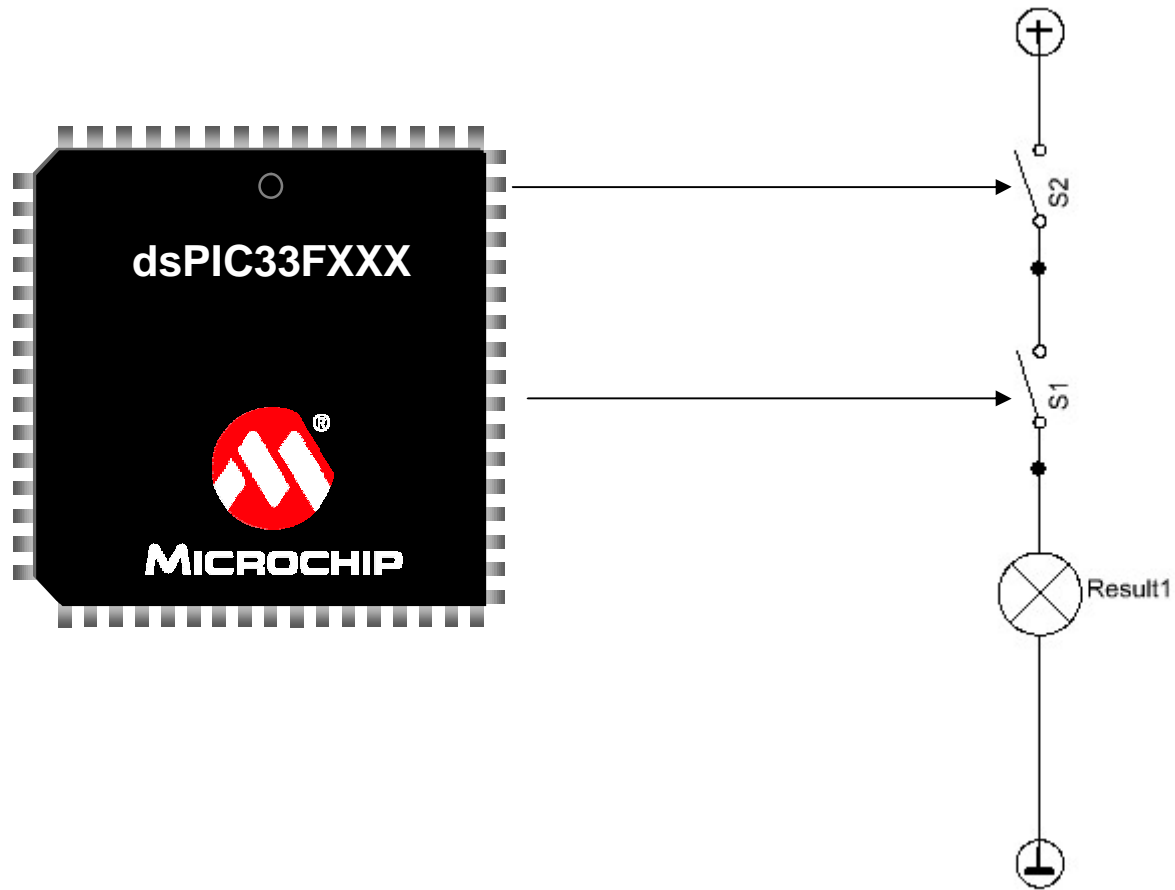
4) Прочитайте все порты снова и сравните с предыдущим значением.

*Гуляющая “1” и “0” по всем портам В/В*

# Что если дефект в ключе?

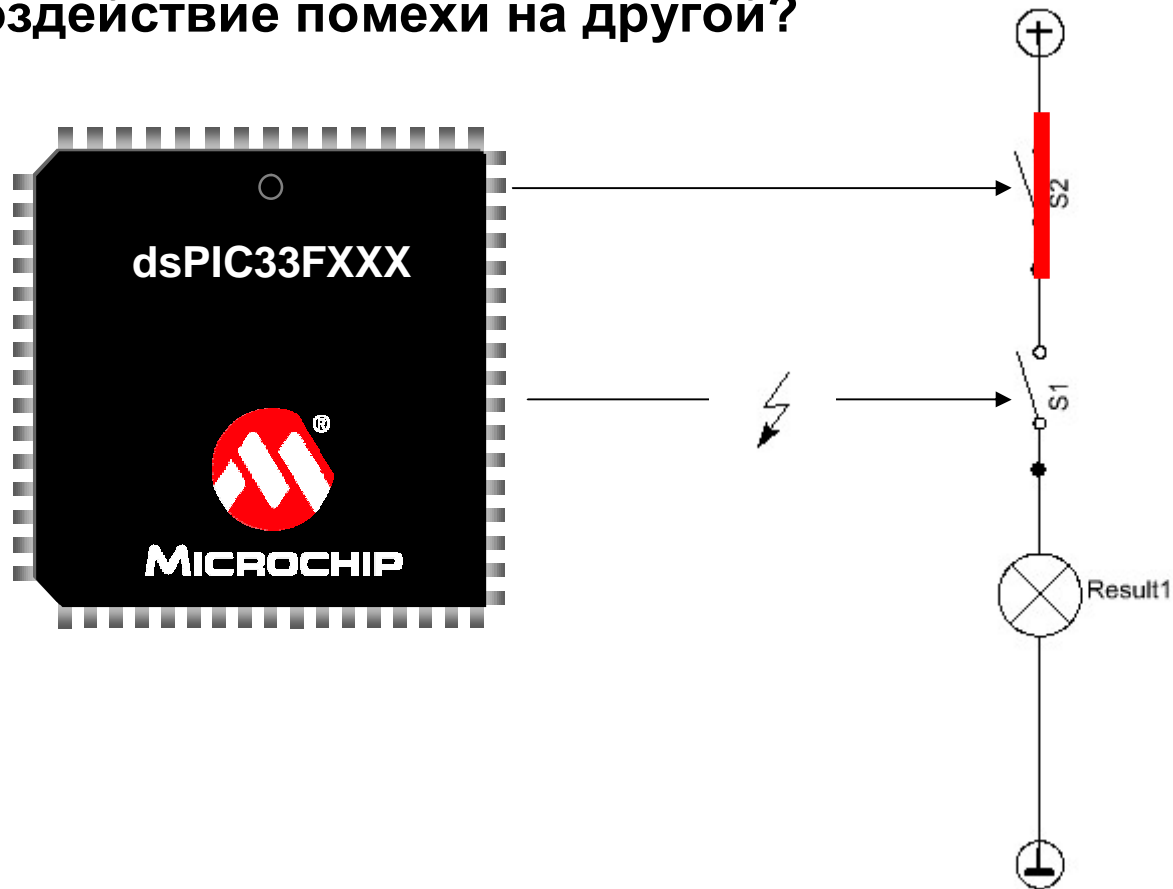


# Второй ключ (транзистор)...

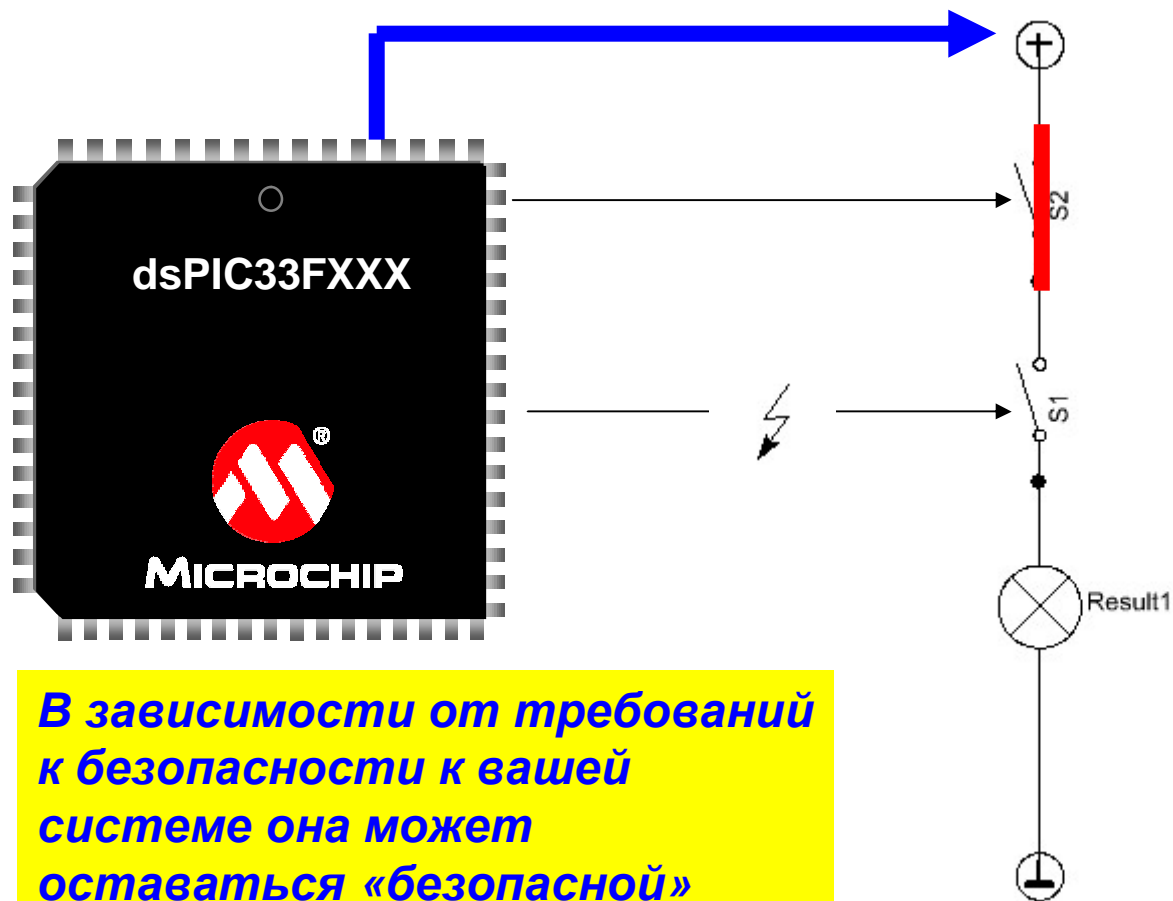


# Второй ключ (транзистор)...но...

Что если выход из строя одного и  
воздействие помехи на другой?



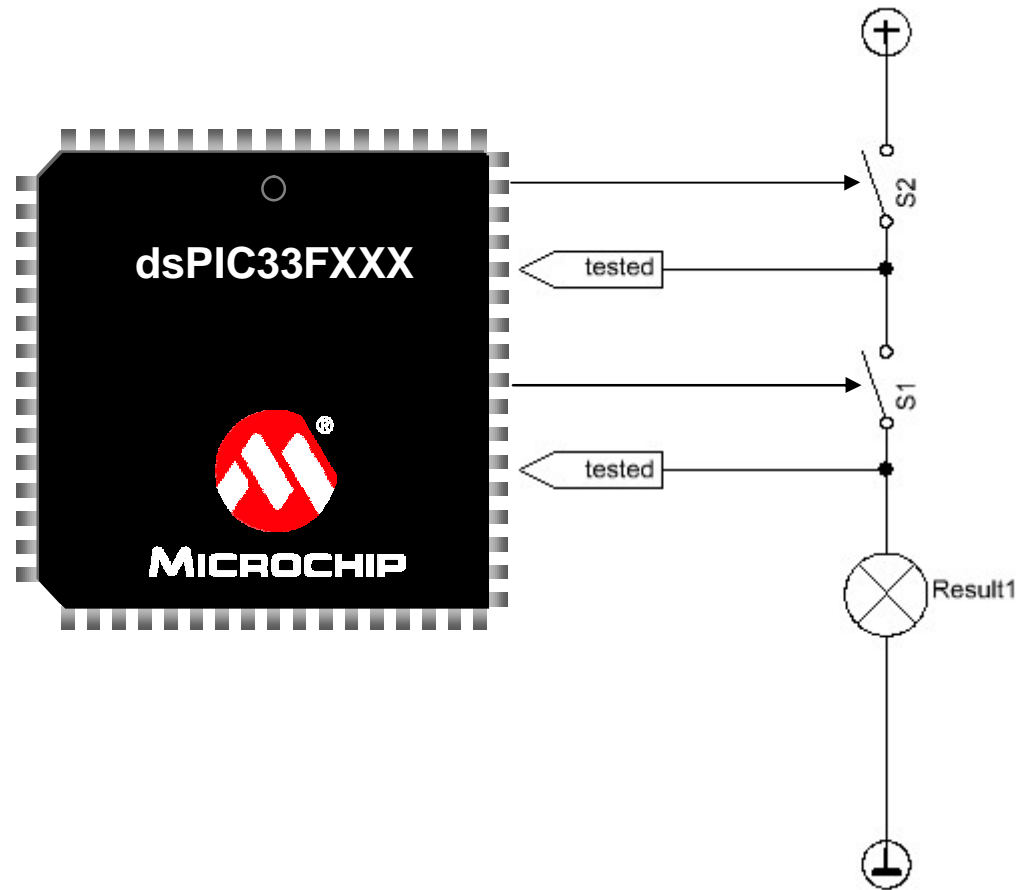
# Хорошо, если вы можете управлять питанием!



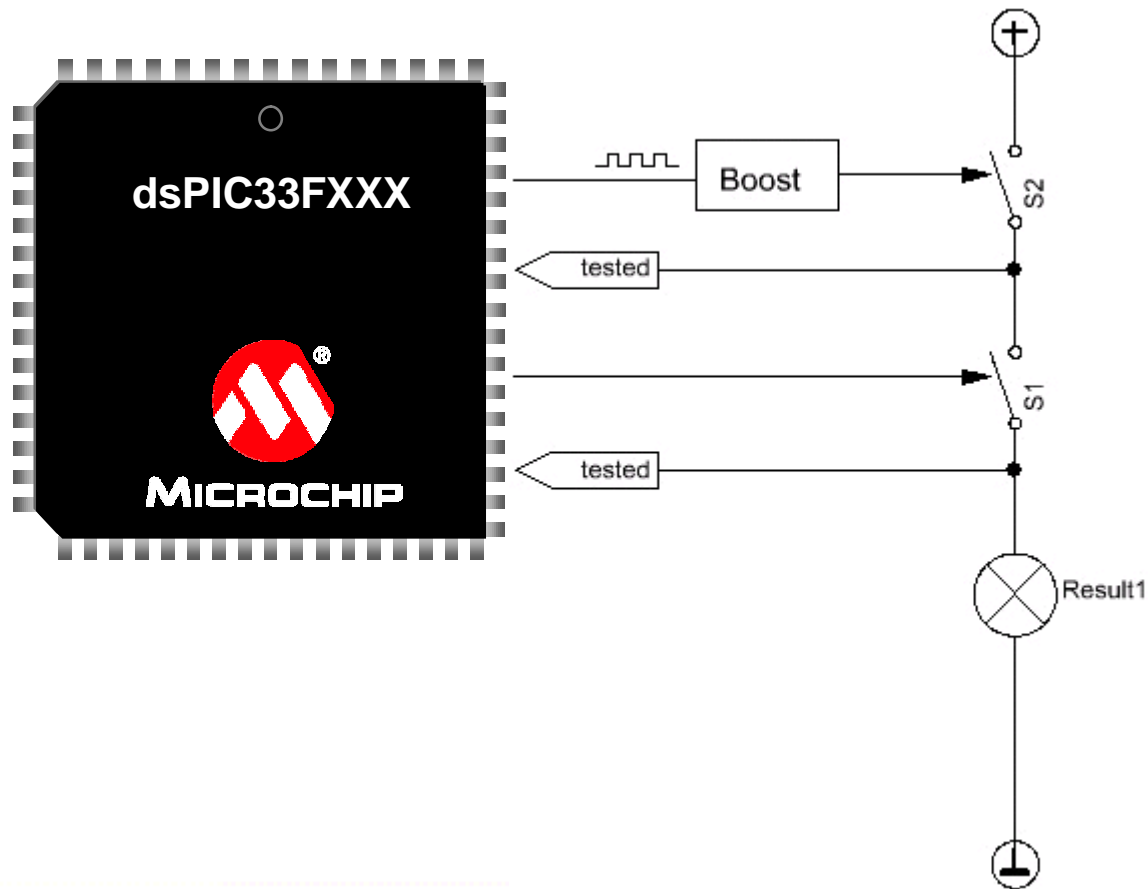
*В зависимости от требований к безопасности к вашей системе она может оставаться «безопасной» даже если есть 2 ошибки в системе!*



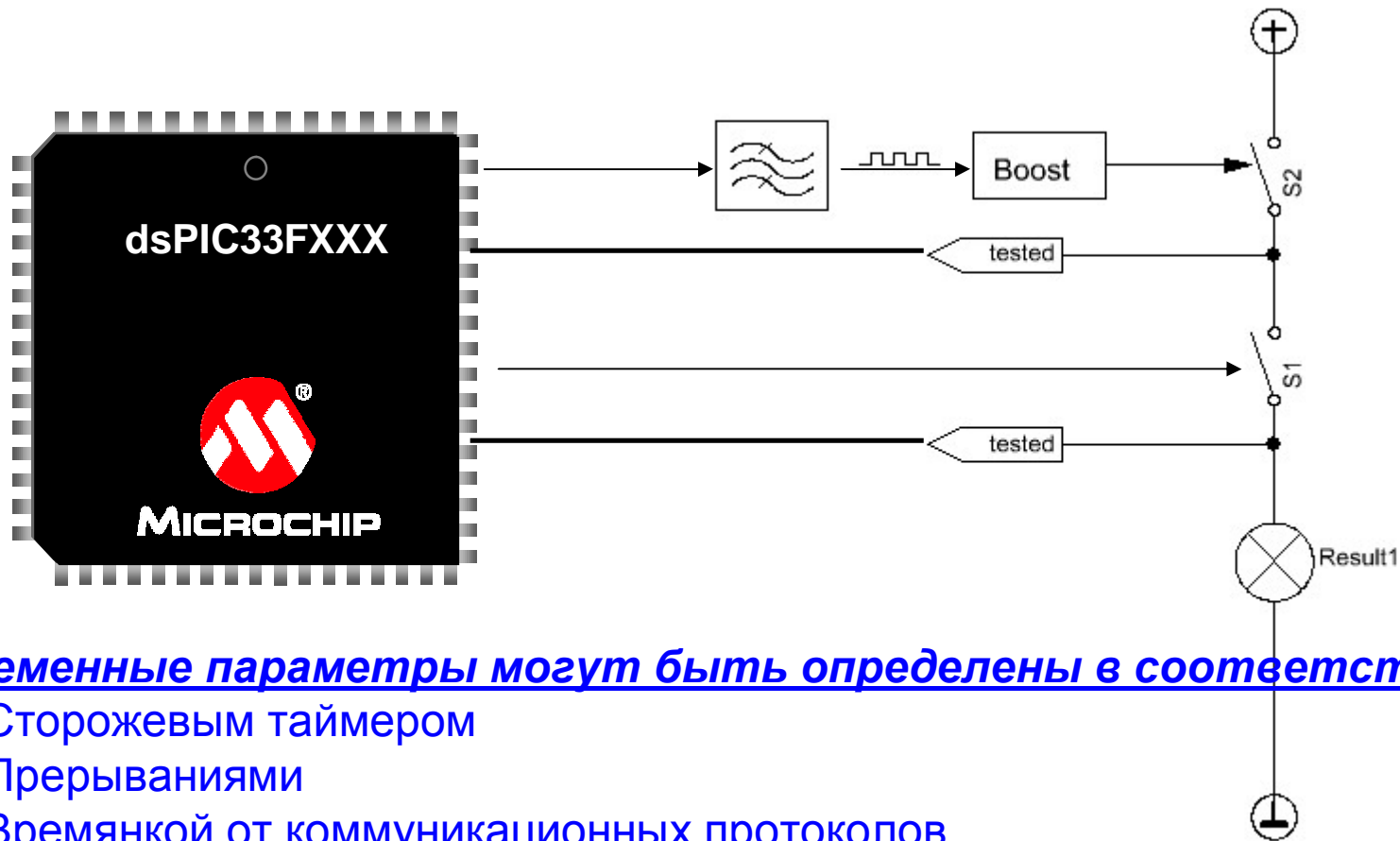
# Второй ключ (транзистор) помогает тестировать...



# Управлять одним ключом напряжением, которое не возможно получить случайно...



# Как вы узнаете что **частота** корректна для преобразователя?

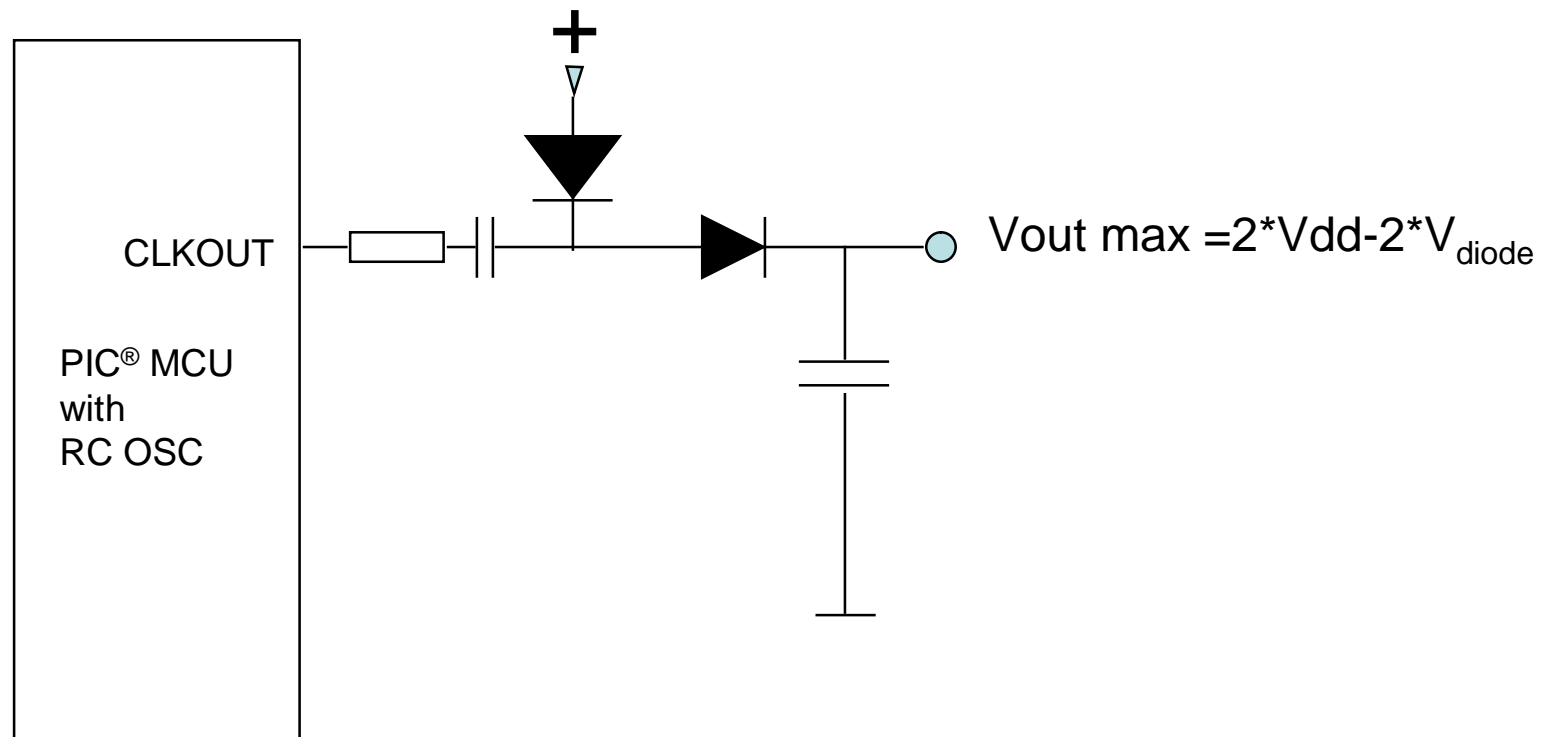


Временные параметры могут быть определены в соответствии с:

- 1) Сторожевым таймером
- 2) Прерываниями
- 3) Времянкой от коммуникационных протоколов

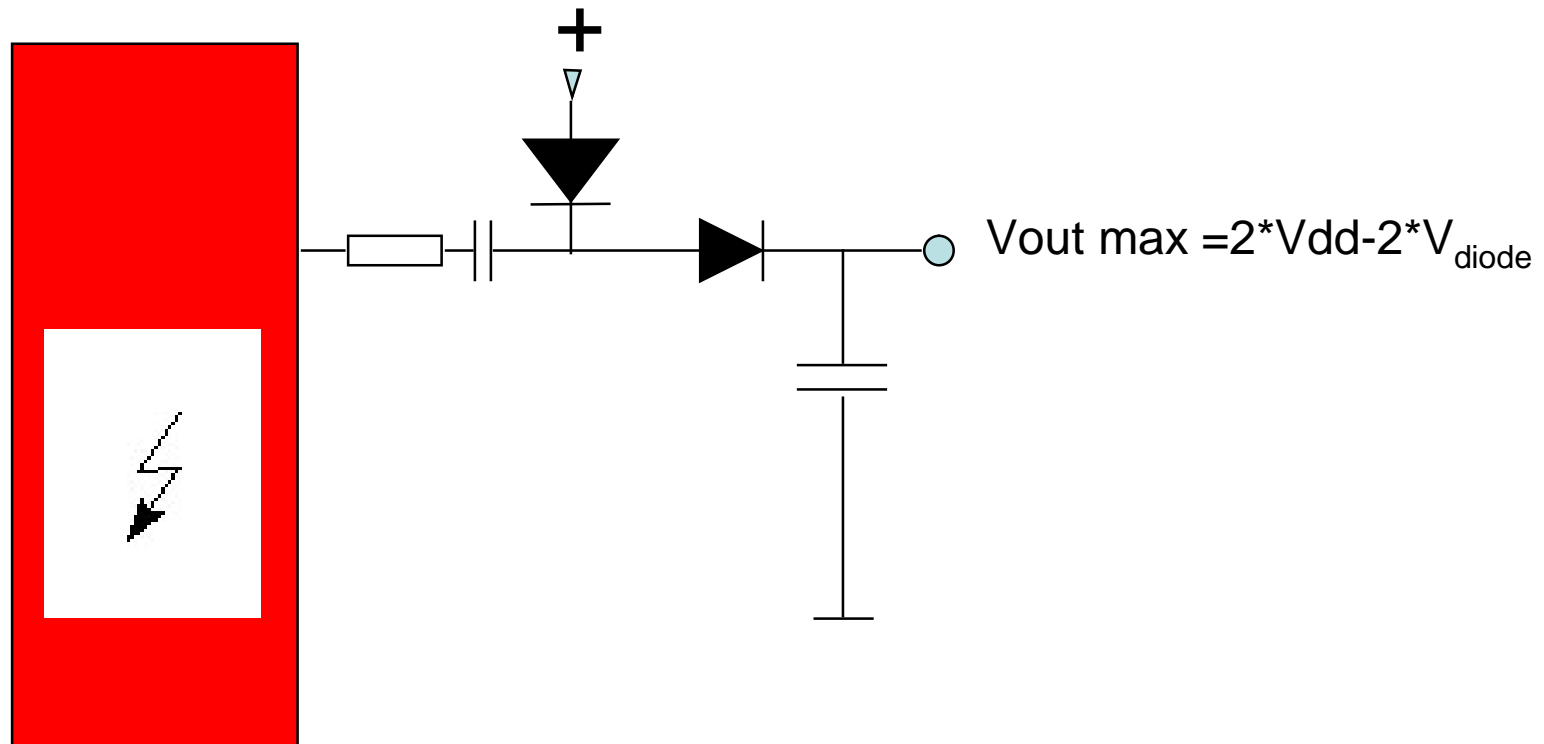
Аппаратный трюк:

# удвоитель напряжения с использованием CLKOUT

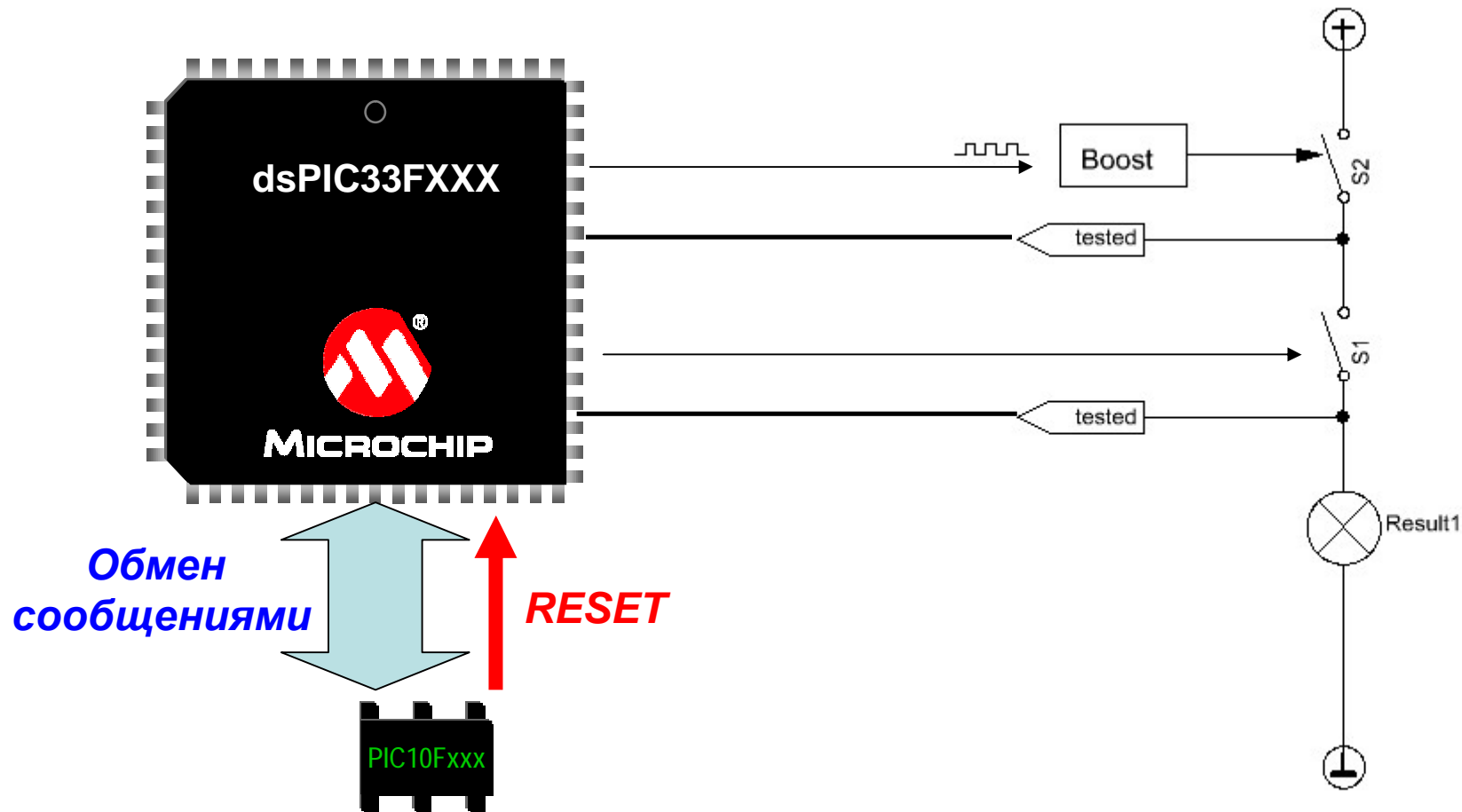


# Аппаратный трюк: No CLKOUT

будет продолжать «тикать»  
при повисшей программе

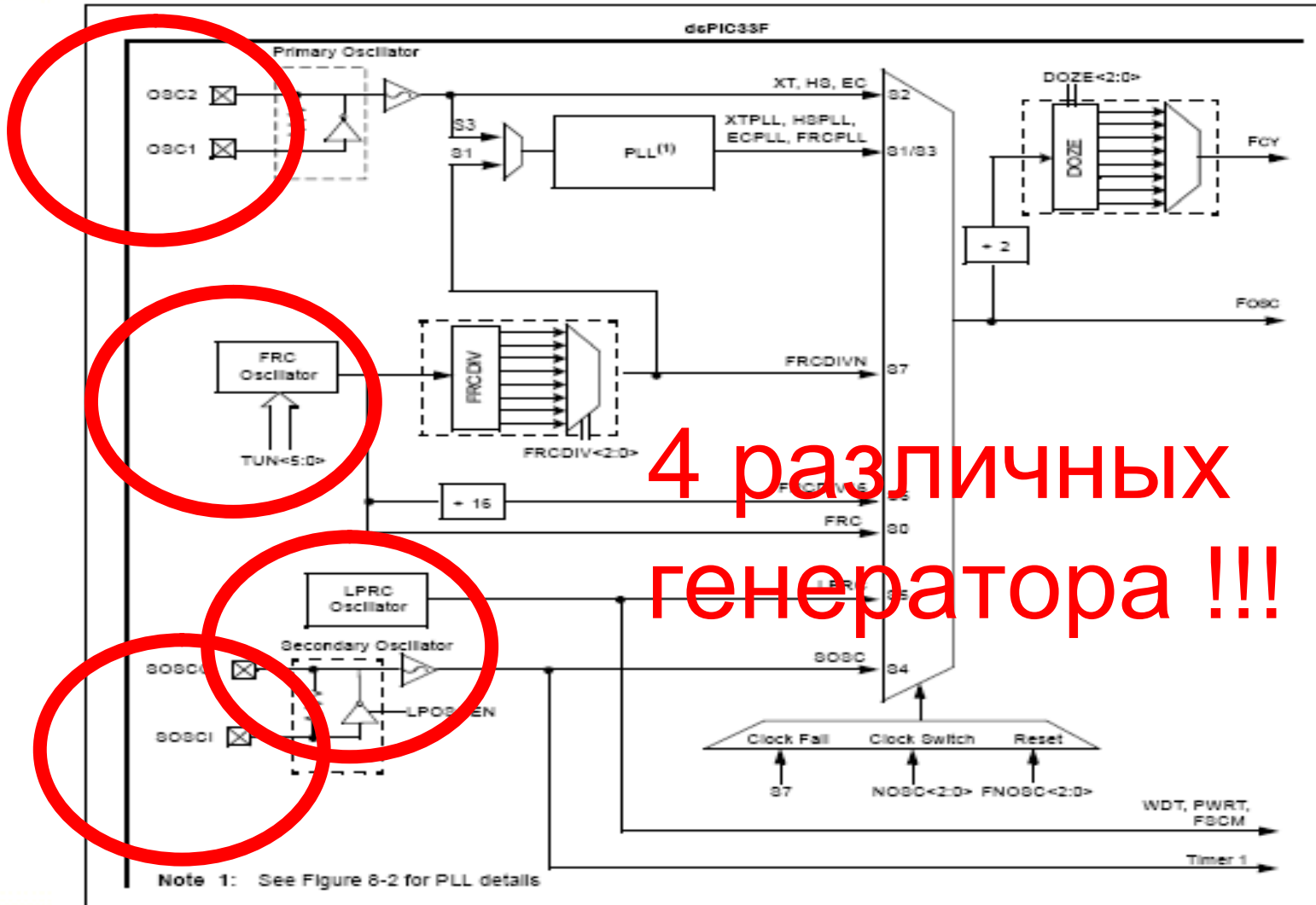


# Нужно доказательство что софт работает или сказать "STOP"



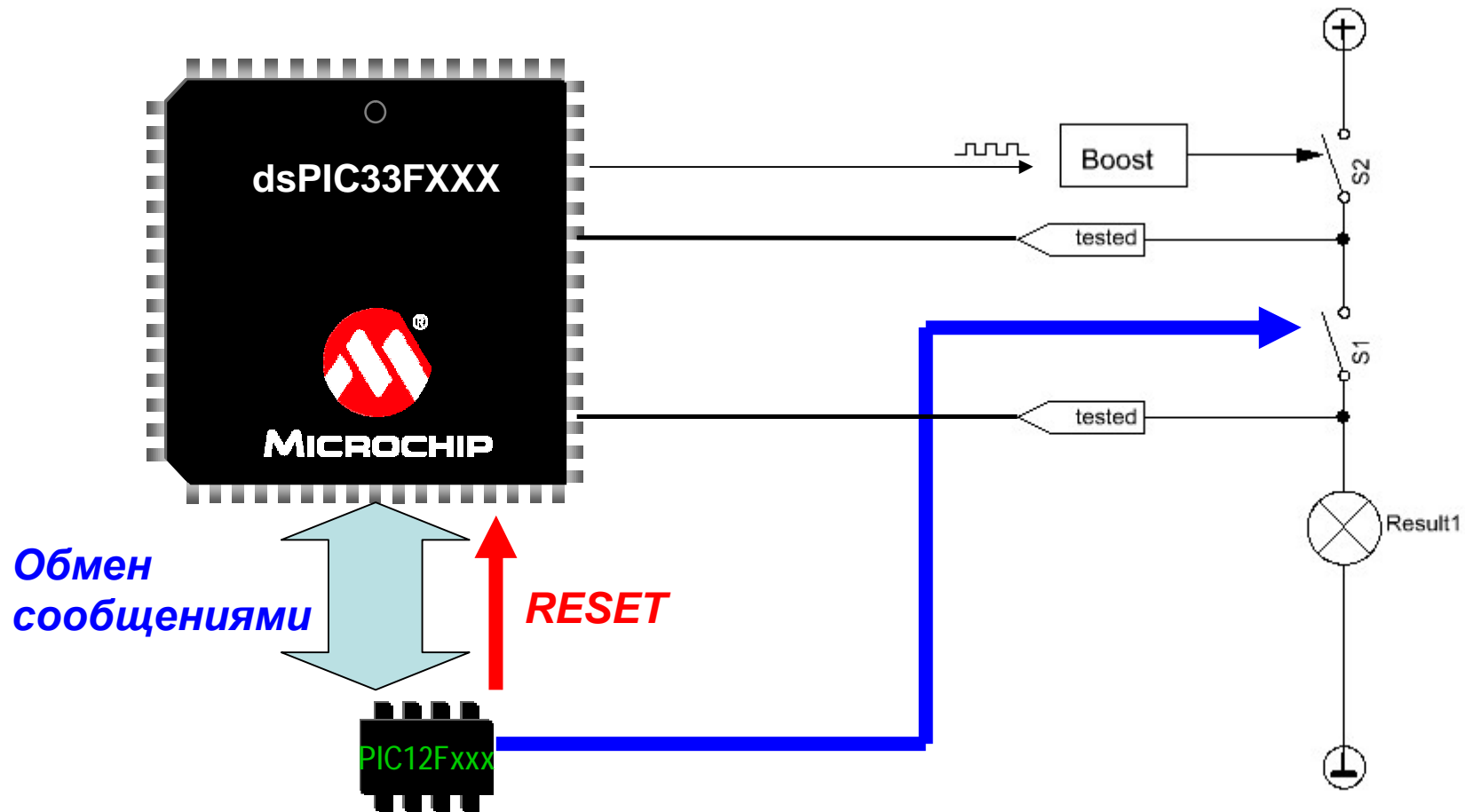
# Источники тактирования dsPIC33 – безопасные генераторы

FIGURE 8-1: dsPIC33FJXXXGPX06/X08/X10 OSCILLATOR SYSTEM DIAGRAM



4 различных генератора !!!

# Убедиться что софт работает и „Все ОК“





# Comparator Voltage Reference

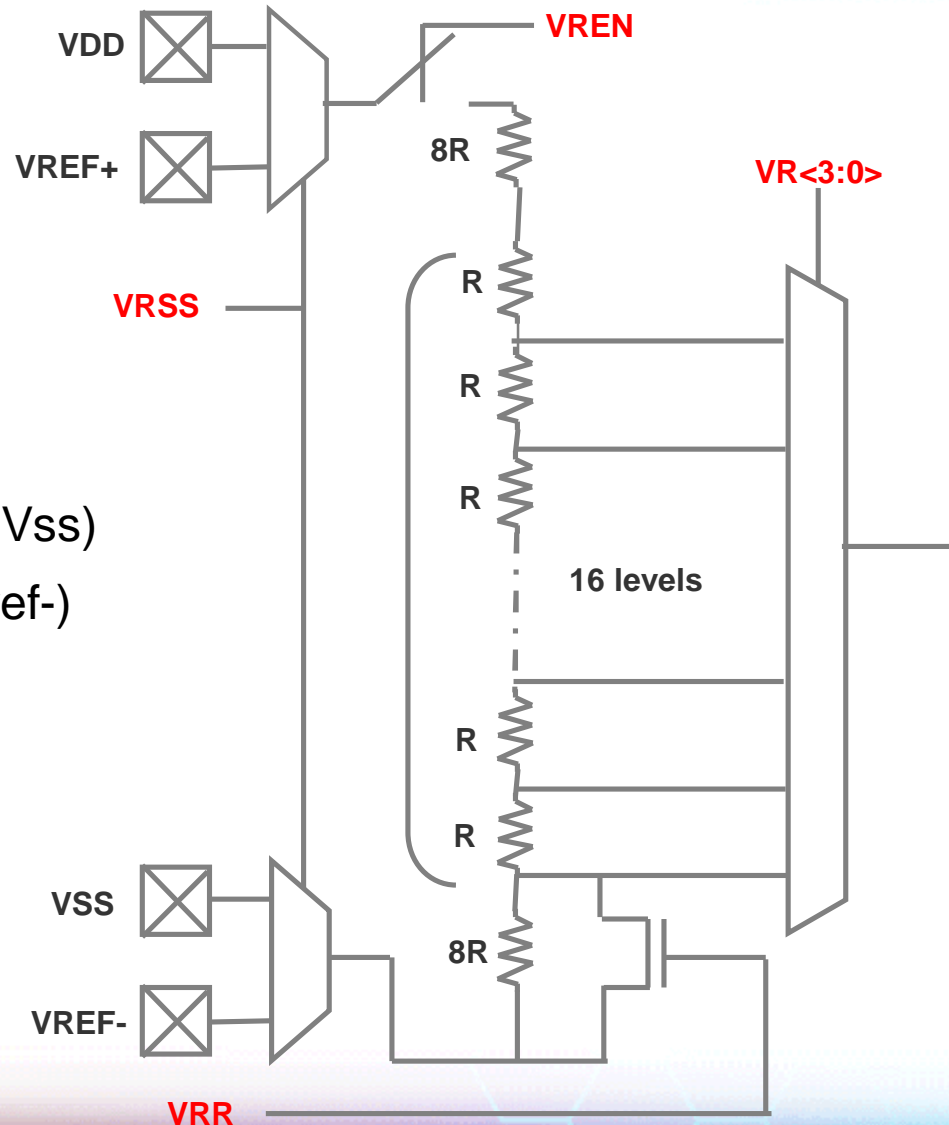
## Опора на делителе

- | 2 k $\Omega$  резисторы
- | 4-bit разрешение
- | 2 различных диапазона

## Опорное напряжение

- | Напряжение питания (Vdd, Vss)
- | Внешние выводы (Vref+, Vref-)

***Позволяет  
динамически  
тестировать  
входы***



# АЦП

## Разрешение

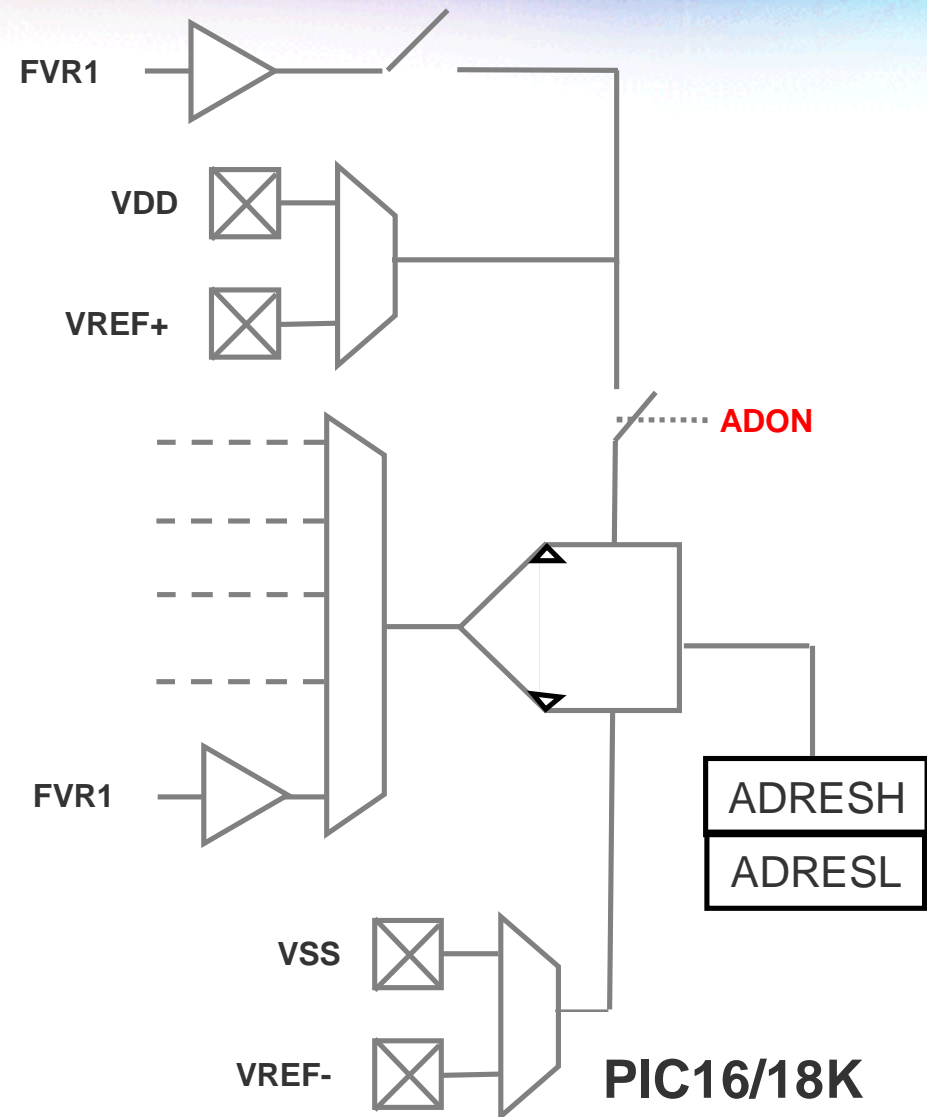
- | 8-bit
- | 10-bit
- | 12-bit

## Частота преобразования

- | 30 ksps for IC12/PIC16/18
- | 100 ksps for PIC18J
- | 500 ksps for PIC24J
- | 100 ksps for PIC16/18K
- | 500ksps for PIC24K
- | 500ksps, 1Msps, 2Msps for dsPIC33

## Опора

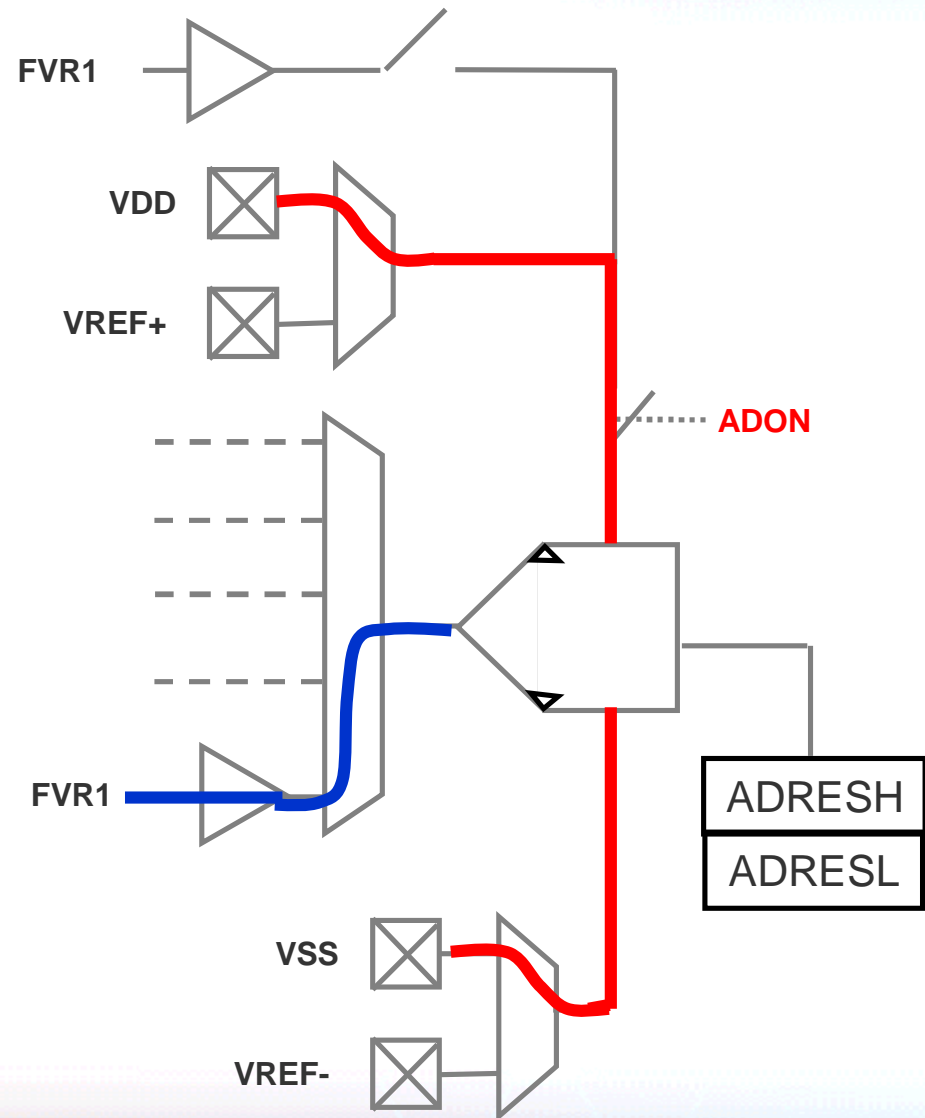
- | VDD/VSS
- | Внешние выводы
- | Фиксированное напряжение как канал
- | Фиксированное напряжение как положительный источник



# Определение VDD

## Возможность:

- Динамически тестировать входы
- Измерять напряжение питания



# Безопасность Системы

- | **Безопасность это вопрос системы в целом НЕ ТОЛЬКО микроконтроллера...**
- | **Программная поддержка так же является важной (специфические методы, которые дают уверенность в правильности работы контроллера)**



# MICROCHIP 2010

---

## MASTERS Conference

# Программные методы обеспечения отказоустойчивости

# Два заблуждения относительно отказоустойчивости ПО

1. **Отказоустойчивым ПО должно быть только в ответственных системах**
2. **99.99(9)% отказоустойчивости обеспечивается схемотехнически**

# Почему не все можно решить схемотехнически?

- | Восстановление работы после пропадания питания
- | Сбой прошивки
- | Ограничения по себестоимости, габаритам, массе
- | Тенденция: заменять схемотехнические узлы программными, а не наоборот

# Предпосылки к обеспечению отказоустойчивости ПО

- | Сильные наносекундные помехи
- | Внешние условия: температура, влага, агрессивные среды
- | Механические воздействия: удары, деформация
- | Утечка заряда носителей информации
- | Порча внешних компонентов и контактов: старение, брак, загрязнение
- | Ошибки монтажа
- | **Ошибки ПО**
- | Нарушение правил эксплуатации



# Стадии появления ошибок ПО

| Техническое задание

| Проектирование

| Кодирование

| **Сторонние библиотеки**

| **Транслятор**

| **Программатор**

| **Микроконтроллер**

Не зависят  
от  
разработчика

# Сложности при создании отказоустойчивого ПО

- | Требуется намного больше времени, сил и затрат
- | Ограничения инструментов
- | Сложности тестирования
- | Отсутствие мотивации: никто не скажет «спасибо»

# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | Блокировка ответственных участков кода
- | Сторожевой таймер
- | Восстановление после сбоя
- | Безопасный режим
- | Резервирование

# Самодиагностика

- | **Предписывается различными стандартами, например:**
  - | IEC 60730 (МЭК 60730) annex H
    - | Class B
    - | Class C
  - | UL1998 (американский стандарт)
    - | Class 1
    - | Class 2

# Самодиагностика

## Однократно или периодически проверяются:

- | Регистры
- | Программный счетчик
- | Прерывания
- | Генератор
- | ROM
- | EEPROM
- | RAM
- | Стек
- | Адресация
- | АЦП и аналоговый мультиплексор
- | сторожевой таймер

# Microchip AN1229

- | Описывается проведение диагностики в соответствии с IEC 60730 (Annex H) для устройств **Class B**
- | Предоставляются библиотеки с исходными кодами для 16-bit/32-bit контроллеров
- | Сертифицирована ассоциацией VDE
- | Готовятся библиотеки и их сертификация для 8-bit контроллеров

# AN1229

- | **Предоставляет функции для диагностики:**
  - | Регистров общего назначения
  - | Программного счетчика
  - | Генератора
  - | Памяти Flash/EEPROM
  - | Памяти RAM и стека
- | **Описывает рекомендации по диагностике:**
  - | Прерываний
  - | Периферии ввода/вывода
  - | Аналогового мультиплексора
  - | АЦП

## Именованние функций:

SSL\_<xx>bitsFamily\_<yy>test[\_<zz>]

Семейство:  
16 или 32

Объект тестирования:

CPU\_Register  
ROM  
RAM  
RAM\_STACK  
CLOCK

Доп. инф:  
тип теста



# AN1229 – Регистры общего назначения

```
int SSL_16bitsFamily_CPU_RegisterTest()  
int SSL_32bitsFamily_CPU_RegisterTest()
```

- | Запись/чтение **0x5555**
- | Запись/чтение **0xAAAA**

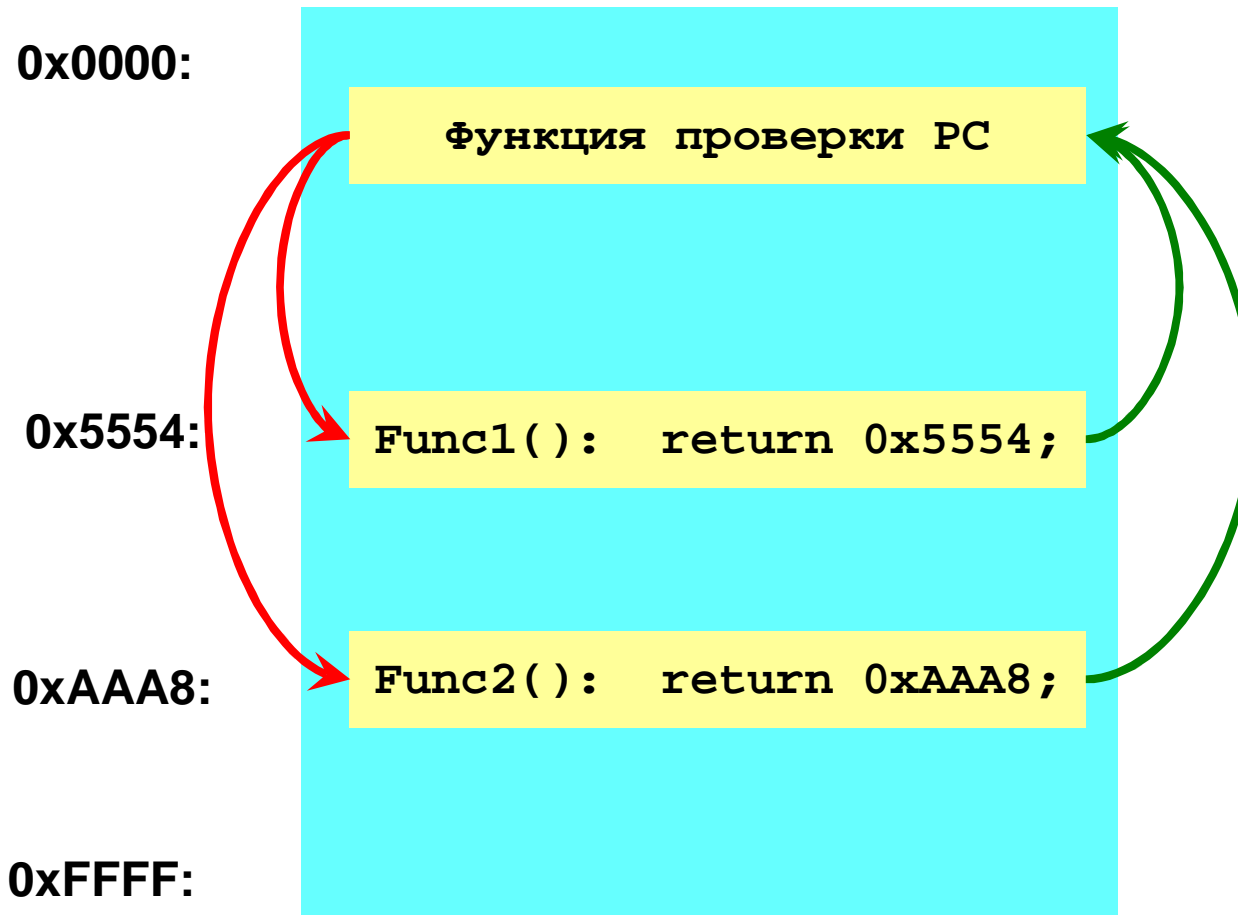
# AN1229 - Программный счетчик

```
int SSL_16bitsFamily_PCtest()  
int SSL_32bitsFamily_PCtest()
```

- | Вызов функций, расположенных в различных областях памяти
- | Сравнение уникальных возвращаемых значений с ожидаемыми
- | Требуется настройка скрипта линкера для размещения тестовых функций

# AN1229 – Программный счетчик

## ROM



# AN1229 – Генератор

```
int SSL_16bitsFamily_CLOCKtest()  
int SSL_16bitsFamily_CLOCKtest_LineFreq()  
int SSL_32bitsFamily_CLOCKtest()  
int SSL_32bitsFamily_CLOCKtest_LineFreq()
```

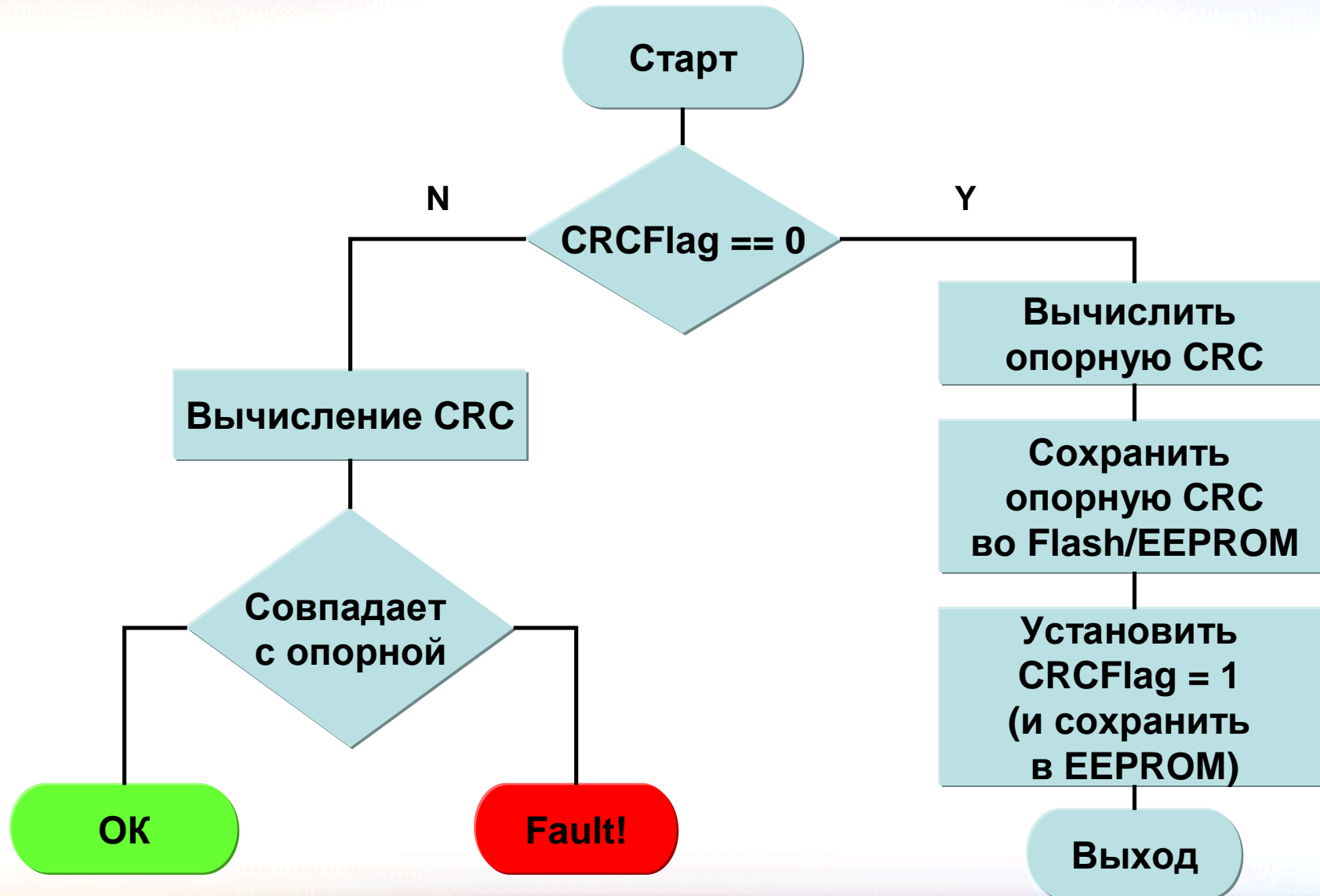
- | Выполняется проверка на гармоники и субгармоники
- | Требуется опорная частота:
  - | **Secondary oscillator**
  - | **Сеть ~50/60 Гц**

# AN1229 - Flash/EEPROM

```
int SSL_16bitsFamily_Flashtest_CRC16  
int SSL_16bitsFamily_EEPROMtest_CRC16  
int SSL_32bitsFamily_Flashtest_CRC16  
    (uReg32 startAddress,  
    uReg32 endAddress,  
    unsigned int init_CRC_Value)
```

- | Обнаруживает сбои в одиночных битах
- | Используется CRC16 или CRC32
- | Для хранения контрольной суммы рекомендуется использовать Flash или EEPROM
- | Позволяет вычислять CRC для различных участков памяти

# AN1229 – Flash



# AN1229 – RAM и стек

```
int SSL_16bitsFamily_RAMtest_MarchC
int SSL_16bitsFamily_RAMtest_MarchC_Minus
int SSL_16bitsFamily_RAMtest_MarchB
int SSL_16bitsFamily_RAMtest_CheckerBoard
int SSL_16bitsFamily_RAM_STACKtest_MarchC
    (int *ramStartAddress, int ramSize)
```

(... И ДЛЯ 32-БИТНЫХ )

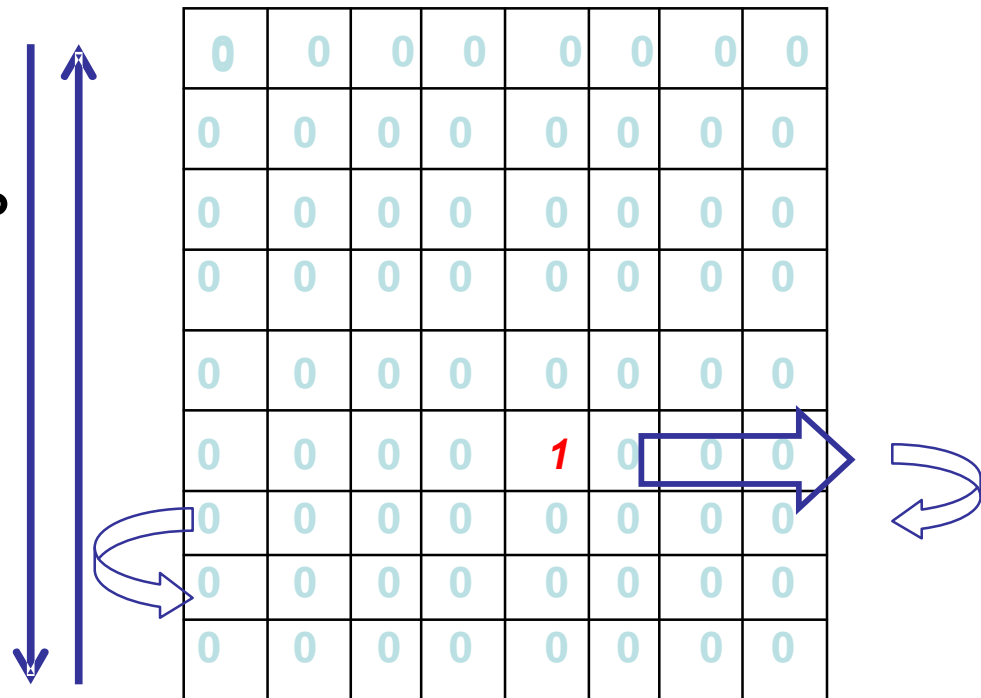
# AN1229 – RAM и стек

- | Обнаруживает сбои в одиночных битах
- | Проверки:
  - | **Только статические ошибки**
- | Тесты методом «Марш»:
  - | **March C**
  - | **March C-**
  - | **March B**



# AN1229 – RAM

- | Бегущая «1»  
проходит всю память
- | Проверка изменения  
остальных ячеек
- | Тест разрушает  
данные!



# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | **Контроль выполнения программы**
- | Блокировка ответственных участков кода
- | Сторожевой таймер
- | Восстановление после сбоя
- | Безопасный режим
- | Резервирование

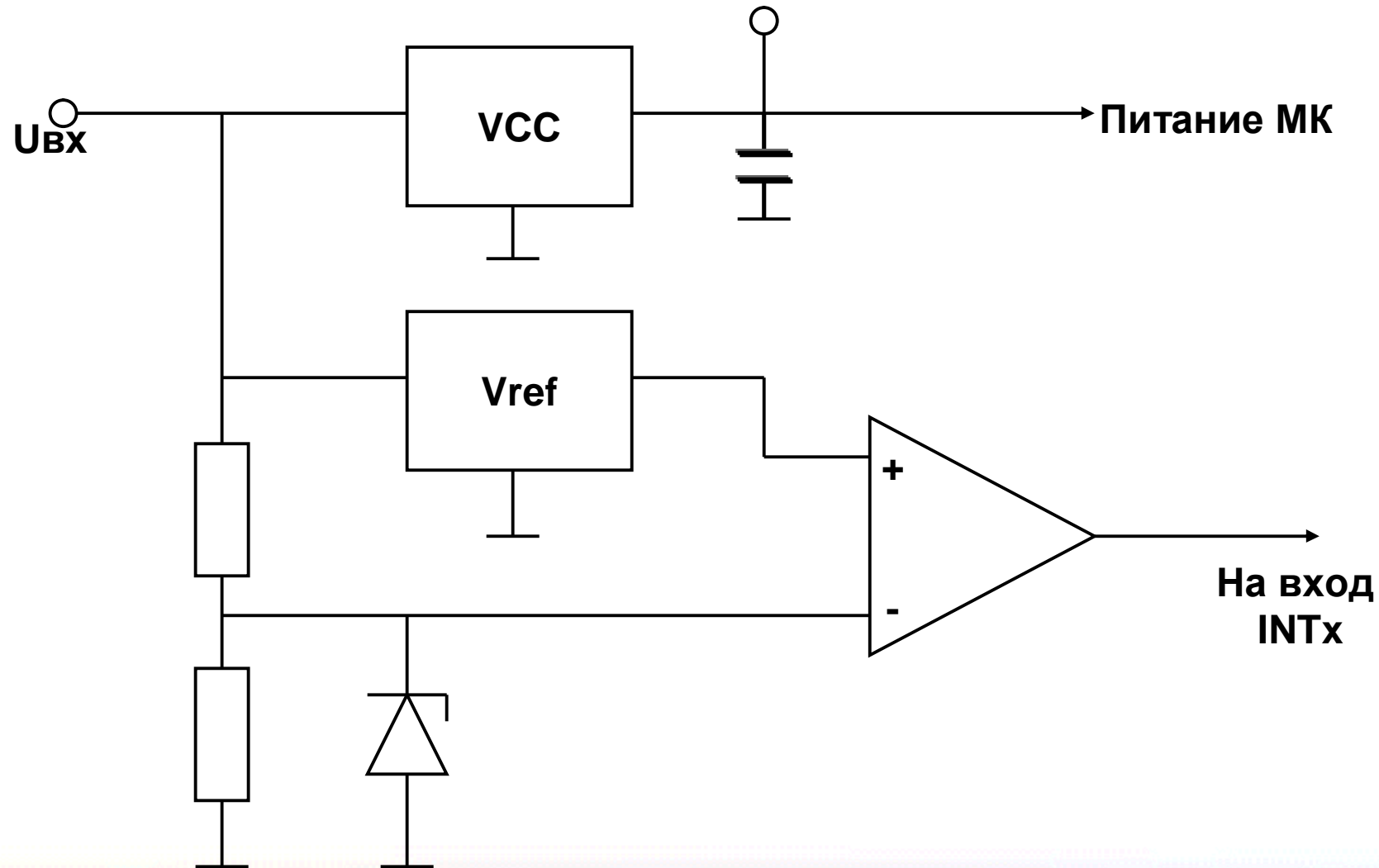
# Контроль выполнения программы

- | **Аппаратные средства**
  - | (не дать программе выполнить некорректные действия)
- | **Контроль периферии**
  - | (не дать программе работать со сбитыми настройками)
- | **Контроль данных**
  - | (не дать программе работать с некорректными данными)
- | **Временные проверки**

# Контроль выполнения программы

- | **Аппаратные средства**
  - | Трапы
    - | Сбой генерации
    - | Ошибка адреса
    - | Математическая ошибка
    - | Ошибка стека
  - | Контроль питания с помощью компаратора (с прерыванием)
  - | WDT – не является средством контроля выполнения программы

# Контроль питания с помощью компаратора



# Контроль выполнения программы

- | **Периодическая проверка периферии:**
  - | Настройки периферийных модулей
  - | Направление портов ввода/вывода
  - | Запрет неиспользуемых прерываний
  - | Проверка разрешения используемых прерываний
  - | Счетчики возникновения прерываний (AN1229)

# Контроль выполнения программы

- | **Контроль данных:**
  - | **Входные аргументы**
    - | Область допустимых значений
    - | Достоверность
  - | **Результаты функций**
    - | Обратный расчет
    - | Параллельный расчет другим способом
- | **Все операции с внешними данными должны сопровождаться контрольными суммами**
- | **Защита внутренних ответственных данных контрольными суммами**

# Контроль выполнения программы

- | **Временные проверки**
  - | **Сбой тактовой частоты**
  - | **«Мусор» во входящих извне данных**
  - | **Время выполнения функций**



# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | **Блокировка ответственных участков кода**
- | Сторожевой таймер
- | Восстановление после сбоя
- | Безопасный режим
- | Резервирование

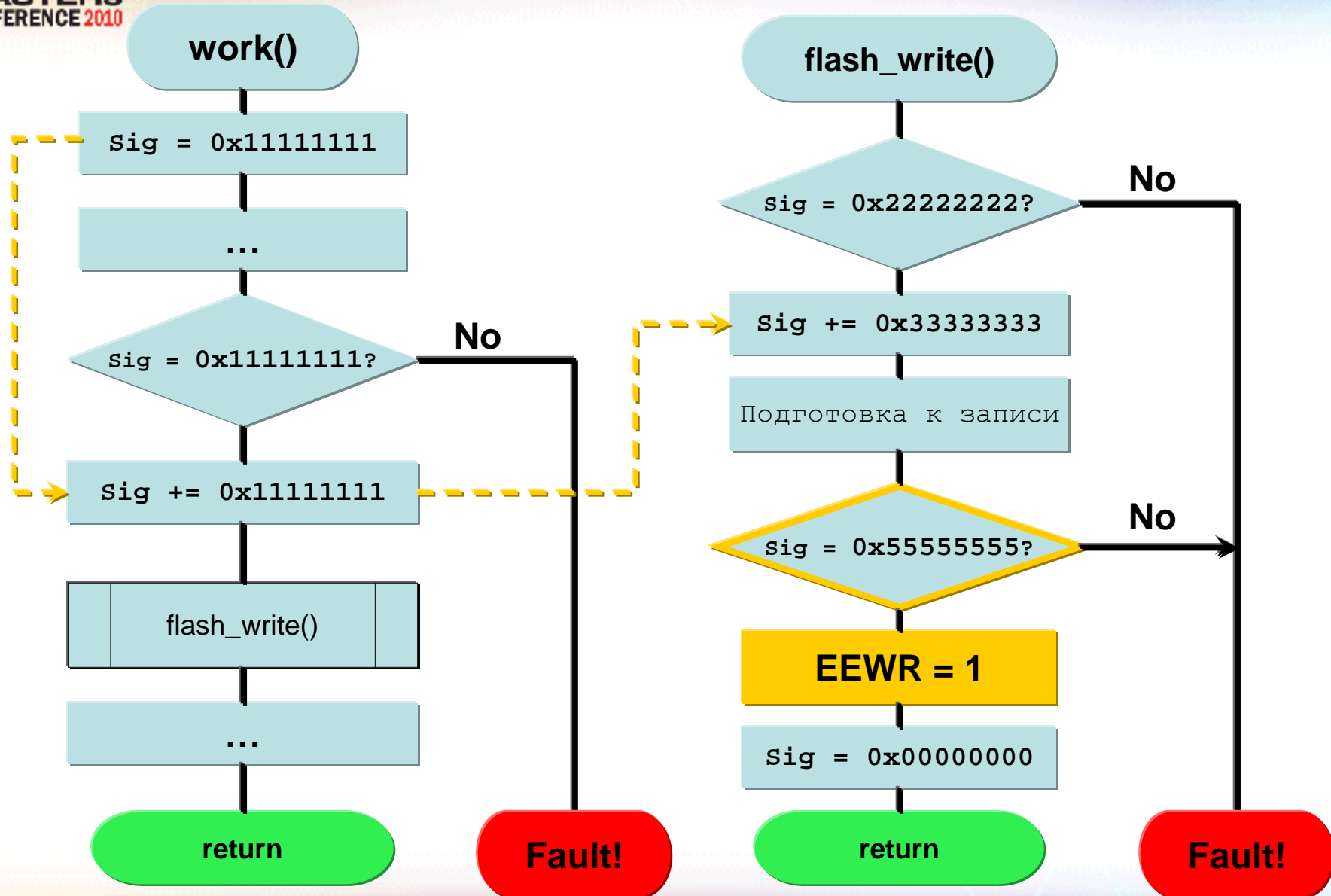
# Блокировка ответственных участков кода

- | **Что блокировать:**
  - | Запись в программную память
  - | Настройку контролирующих узлов:  
WDT, BOD, прерывания
  - | Вход в безопасный режим

# Блокировка ответственных участков кода

- | **Как блокировать:**
  - | Контроль стека
  - | Проверка состояния программы
  - | Сигнатуры
  - | Контроль времени

# Пример применения сигнатур



# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | Блокировка ответственных участков кода
- | **Сторожевой таймер**
- | Восстановление после сбоя
- | Безопасный режим
- | Резервирование

# WDT

- | Аппаратный механизм защиты от сбоев
- | Всегда предпочтительнее обнаруживать и устранять сбои программно, WDT – последний рубеж

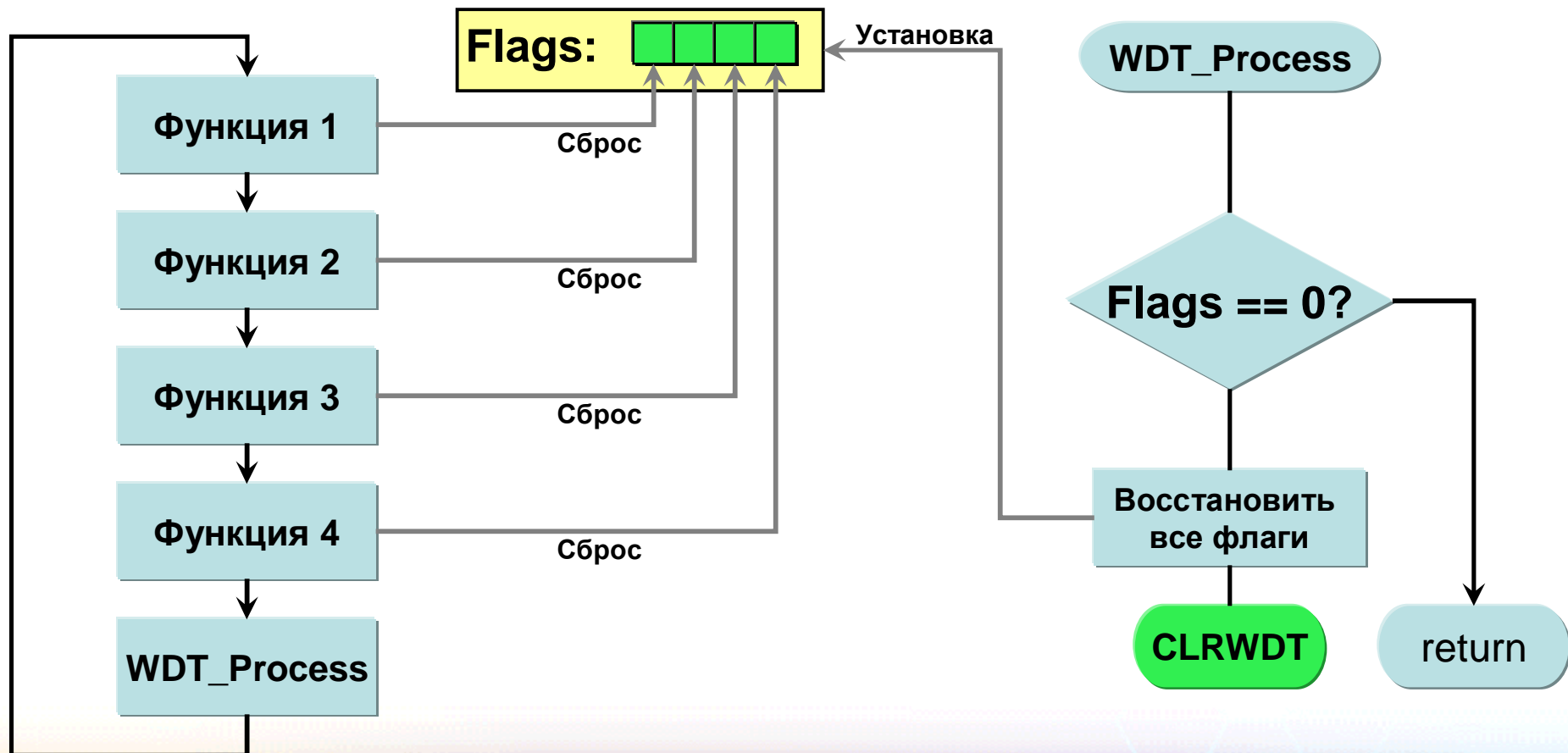
- | **Конструктивное исполнение**
  - | Внутренний
  - | Внешний
    - | Специализированная микросхема
    - | Схема на дискретных компонентах
    - | Схема на МК
- | **Типы**
  - | Традиционный
  - | Оконный
  - | Смешанный

- | **Методы обнуления**
  - | **Нельзя обнулять безусловно!**
  - | Обнуление с проверкой флагов
  - | Периодическое обнуление с контролем состояния



# WDT: Методы обнуления

## Обнуление с проверкой флагов



# WDT: Методы обнуления

## Обнуление с проверкой флагов

### | Преимущества:

- | Простота
- | Наглядность
- | Нетребовательность к ресурсам

### | Недостатки:

- | Обязывает все функции выполняться, даже когда это не нужно
- | Требует, чтобы все функции успели выполниться быстрее интервала WDT

# WDT: Методы обнуления

## Периодическое обнуление с контролем состояния

- | Контроль за состоянием работы
- | Не требует обязательной отработки функции в пределах одного периода WDT
- | Позволяет функции выполняться дольше одного периода WDT
- | Требует тщательной проработки порядка проверок

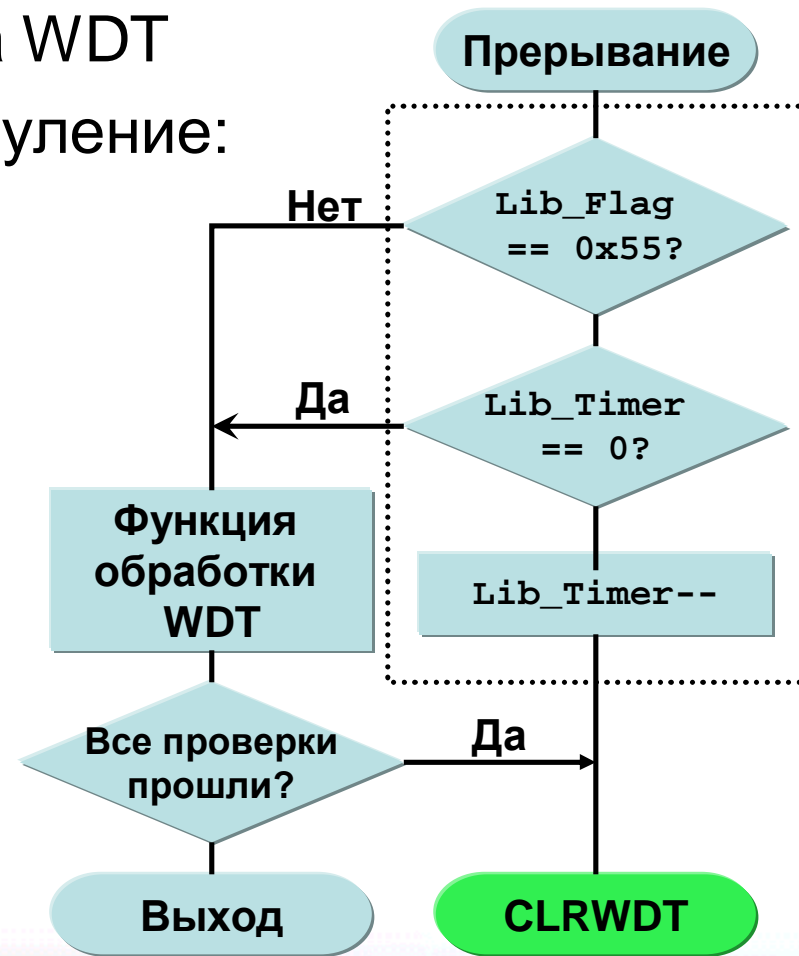
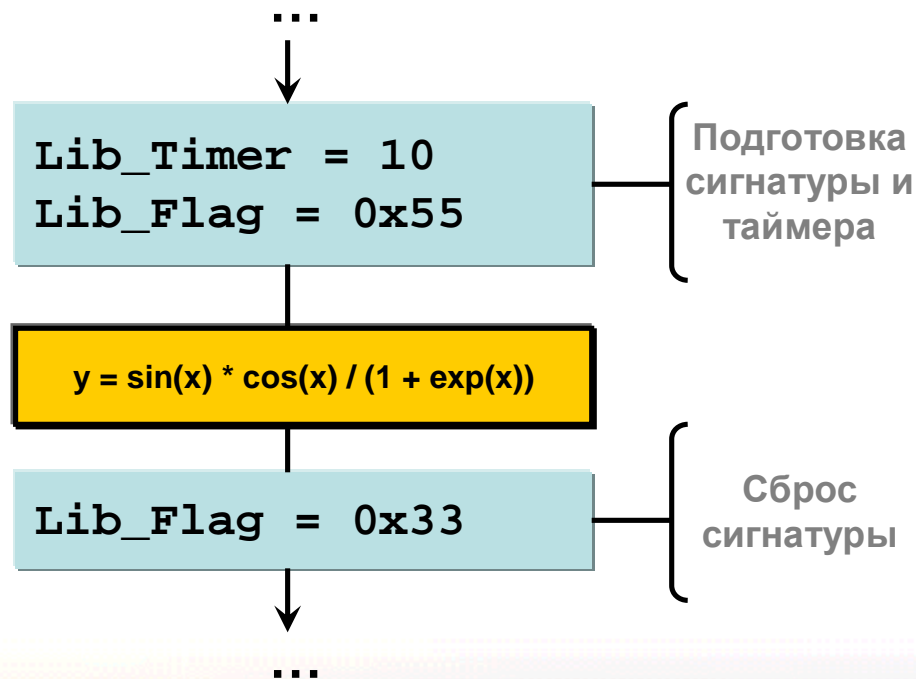
## | Выбор интервала

- | Максимально быстрая реакция на сбой
- | Время выполнения проверок для обнуления WDT д.б. намного меньше интервала
- | Период WDT должен быть на порядок больше времени выполнения критических узлов кода

# WDT

## Работа со сторонними библиотеками:

- ▮ Либо увеличение периода WDT
- ▮ Либо принудительное обнуление:



# WDT

- | **Работа под управлением RTOS**
  - | Для каждой задачи – своя функция проверки
  - | WDT обнуляется, только если все функции вернули положительный результат

# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | Блокировка ответственных участков кода
- | Сторожевой таймер
- | **Восстановление после сбоя**
- | Безопасный режим
- | Резервирование

# Восстановление после сбоя

- | Поведение зависит от стадии, на которой был обнаружен сбой
  - | Сброс по WDT
  - | Сброс по внешним причинам
  - | Сбоем обнаружен программными средствами (стек, настройки и т.п.)



# Восстановление после сбоя

- | **Что делать:**
  - | **При сбросе**
    - | Протоколирование
    - | Вспоминание последнего состояния
    - | Переход в безопасный режим
  - | **При программном обнаружении**
    - | Попытка восстановления работоспособности
    - | Протоколирование
    - | Программный сброс контроллера
    - | Переход в безопасный режим

# Восстановление после сбоя

- | **Протоколирование**
  - | При сбросе
  - | При обнаружении сбоя программными средствами
  - | Периодическое:
    - | в энергонезависимую память
    - | внешним устройствам по каналам СВЯЗИ

# Восстановление после сбоя

- | **Что протоколируем:**
  - | Счетчик сбоев (или сбросов)
  - | Причины (POR, BOR, WDT, Soft)
  - | Переменные состояния программы
  - | Содержимое стека
  - | Дата и время
  - | Под RTOS: дескрипторы задач

# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | Блокировка ответственных участков кода
- | Сторожевой таймер
- | Восстановление после сбоя
- | **Безопасный режим**
- | Резервирование

# Безопасный режим

## | Зачем нужен

- | Перевести внешние узлы в безопасное состояние
- | Сообщить оператору о неисправности
- | Не дать устройству висеть в неопределенном состоянии

# Безопасный режим

## | Характеристики

- | Хранение в памяти двух копий функции безопасного режима с контрольными суммами
- | Расчет на резервный генератор
- | Использование минимума ресурсов

# Средства повышения отказоустойчивости ПО

- | Самодиагностика
- | Контроль выполнения программы
- | Блокировка ответственных участков кода
- | Сторожевой таймер
- | Восстановление после сбоя
- | Безопасный режим
- | Резервирование

# Резервирование

- | **Резервирование данных**
  - | Избыточность
  - | Дублирование
- | **Резервирование кода**
  - | N-версирование
  - | Дублирование



# Резервирование данных

- | **Избыточность**

- | Контрольные суммы
- | Использование ячеек большей разрядности

- | **Дублирование**

- | При записи в энергонезависимую память
- | При передаче данных вовне (обязательно использование протоколов с подтверждением)
- | Ответственные блоки памяти

# Резервирование кода

- | **N-версирование**
  - | Одни и те же вычисления разными способами
  - | Проверка вычислений обратными действиями:  $X = Y * Z \Rightarrow Y = X / Z$
- | **Дублирование**
  - | По две копии ответственных функций
  - | Выбор рабочей копии – по совпадающей CRC

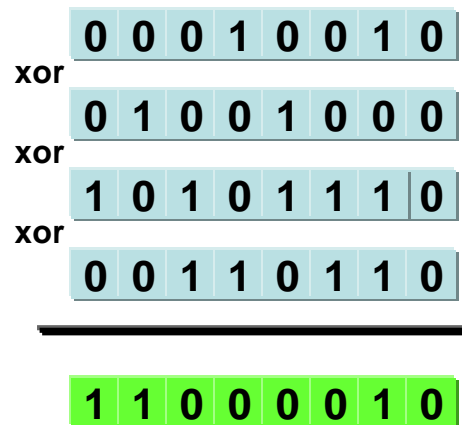
# О контрольных суммах

- | Топология элементов памяти такова, что наиболее вероятны сбои в одинаковых битах
- | Предпочтительнее вычислять методом циклического избыточного кода
- | Не рекомендуется применять простые математические операции (например, XOR, ADD, ADD с инверсией и т.д.), т.к. они неэффективны при возникновении двойных, тройных и т.д. ошибок

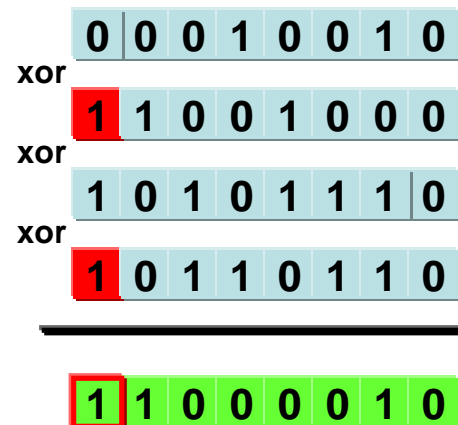
# О контрольных суммах

Пример ненадежности контрольной суммы XOR.

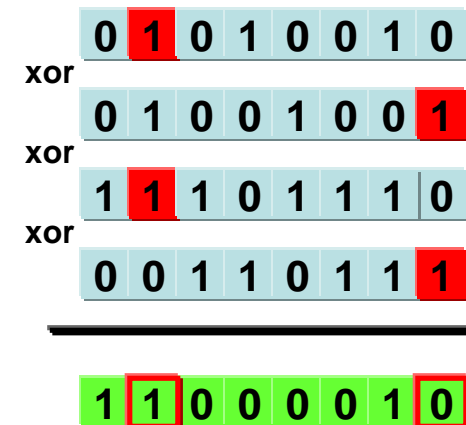
Исходные данные: 0x12, 0x48, 0xAE, 0x36



Сходится при  
отсутствии  
ошибок



Сходится при  
двойной ошибке



Сходится при  
многократной  
ошибке

# О контрольных суммах

Пример ненадежности контрольной суммы **ADD**.

Исходные данные: 0x12, 0x48, 0xAE, 0x36

$$\begin{array}{r}
 00010010 \\
 + 01001000 \\
 + 10101110 \\
 + 00110110 \\
 \hline
 00111110
 \end{array}$$

Сходится при  
отсутствии  
ошибок

$$\begin{array}{r}
 00010010 \\
 + 11001000 \\
 + 10101110 \\
 + 10110110 \\
 \hline
 00111110
 \end{array}$$

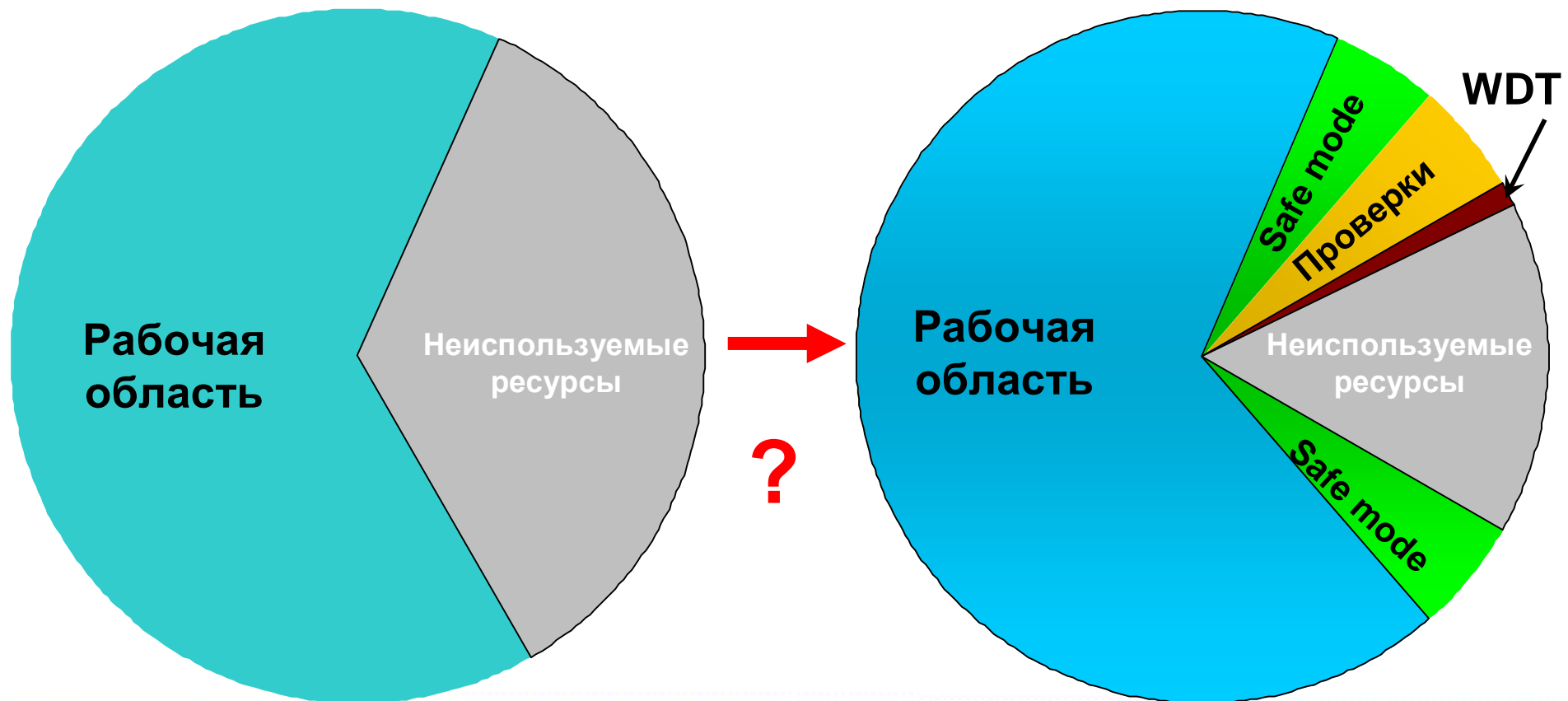
Сходится при  
двойной ошибке

$$\begin{array}{r}
 10010011 \\
 + 01001000 \\
 + 10101110 \\
 + 00110110 \\
 \hline
 10111111
 \end{array}$$

Сходится при  
многократной  
ошибке

# «Кто будет сторожить сторожей?»

Код становится больше. Область уязвимости увеличивается???



# «Кто будет сторожить сторожей?»

- | Сбой может случиться в любом месте и в любое время
- | При наличии проверок снижается область уязвимости (вместо рабочей области уязвима только область проверок и WDT)
- | Две копии безопасного режима спасают от одиночного сбоя (желательно, чтобы они располагались в различных областях памяти)



# Спасибо!