СВД
Встраиваемые Системы

Измерение характеристик реального времени ЗОСРВ «Нейтрино»

Владимир Махилёв

Руководитель отдела промышленных проектов

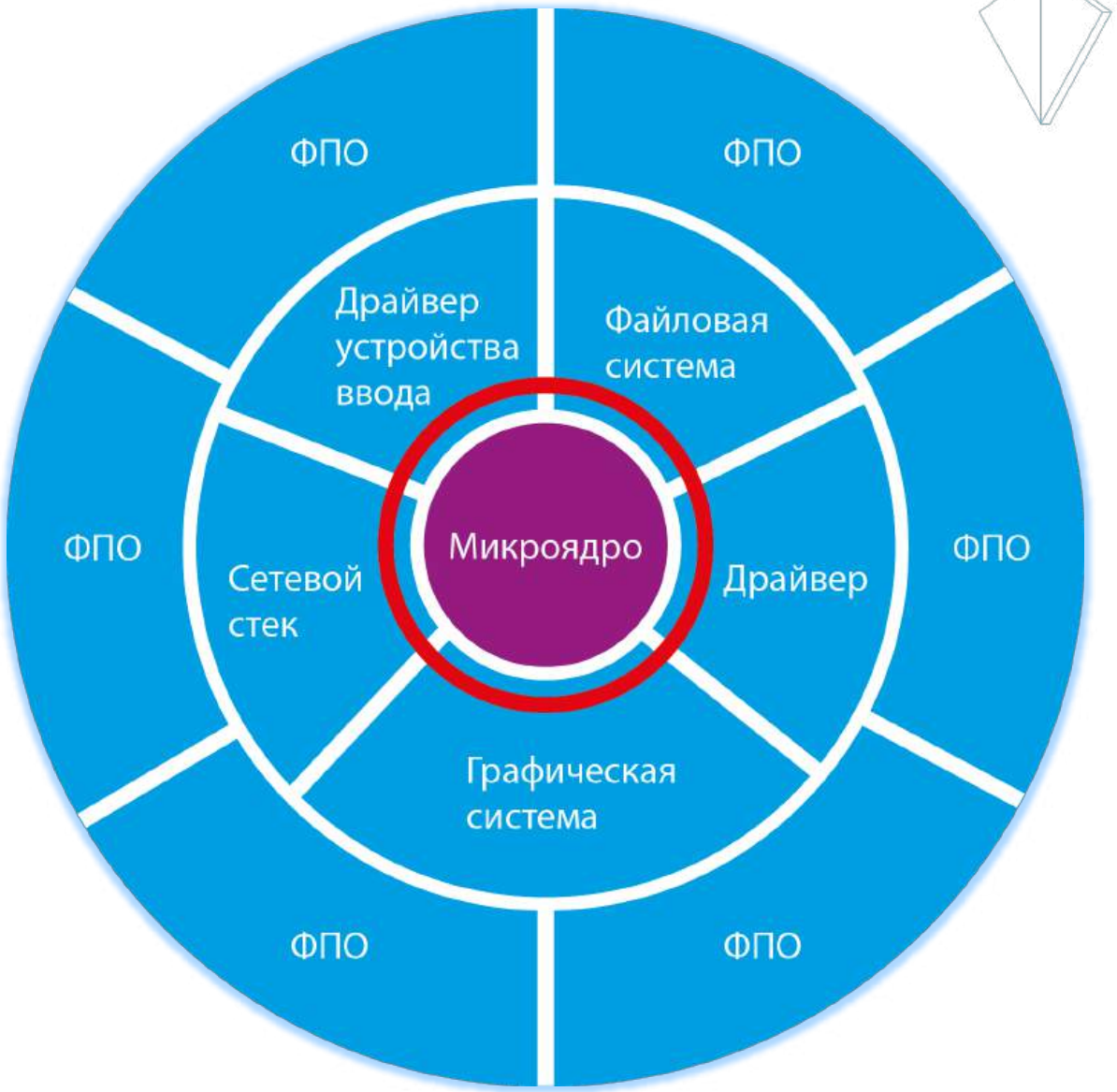
О нас

- «СВД ВС» основана в 2002 году
- Основное направление - разработка ОС реального времени «Нейтрино»
- Офис в Санкт-Петербурге
- Лицензии
  - Минобороны – разработка СЗИ
  - ФСТЭК – разработка СЗИ
  - Минпромторг – разработка вооружений и военной техники
  - Комитет по образованию Санкт-Петербурга – образовательная деятельность

СВД
Встраиваемые Системы

ОСРВ «Нейтрино»: Микроядерная архитектура

- **Микроядро отвечает только за передачу сообщений (IPC), системные вызовы, прерывания, исключения**
- **Изолированное адресное пространство для всех процессов, драйверов, системных служб и сервисов**
- **Сбой в процессе не влияет на микроядро и процесс может быть перезапущен, включая драйверы**
- **Б’ольшая часть кода ядра вытесняемая, включая системные вызовы, что обеспечивает предсказуемое поведение и время отклика**
- **Динамический запуск и остановка драйверов без перезагрузки всей системы**
- **Масштабируемость и встраиваемость**



План доклада

- **Метод замеров характеристик реального времени**
  - Общие понятия: задержка обработки прерывания, задержка планирования и получения управления в пользовательском потоке
  - Стенд для замеров
- **Анализ результатов**
  - Время реакции на внешние прерывания
  - Максимальная выдерживаемая частота внешних прерываний без потерь
  - Сравнение характеристик с другими платформами
- **Дополнительные практические вопросы**
  - Распределение прерываний по процессорным ядрам
  - Установка порога срабатывания прерывания
  - Трасса системных событий

Характеристики реального времени

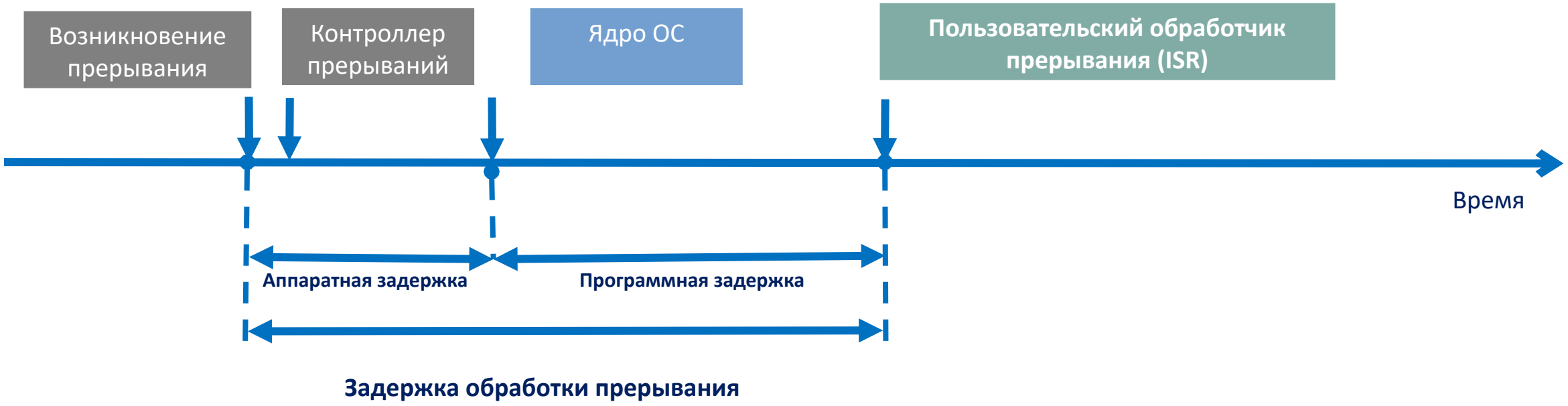
- **Время реакции на внешнее событие (прерывание)**
  - Задержка
  - Разброс значений (джиттер)
- **Пропускная способность системы**
  - Частота внешних прерываний
  - Количество потерянных прерываний

Обработка прерывания: два подхода

Процесс может:

- **Зарегистрировать функцию-обработчик, которую ядро будет вызывать при прерывании**
  - Выполняется в контексте ядра
  - Ограниченный функционал
  - Обработчик должен быть максимально коротким
- **Получать уведомление о возникновении прерывания**
  - Выполняется в контексте процесса
  - Полный функционал

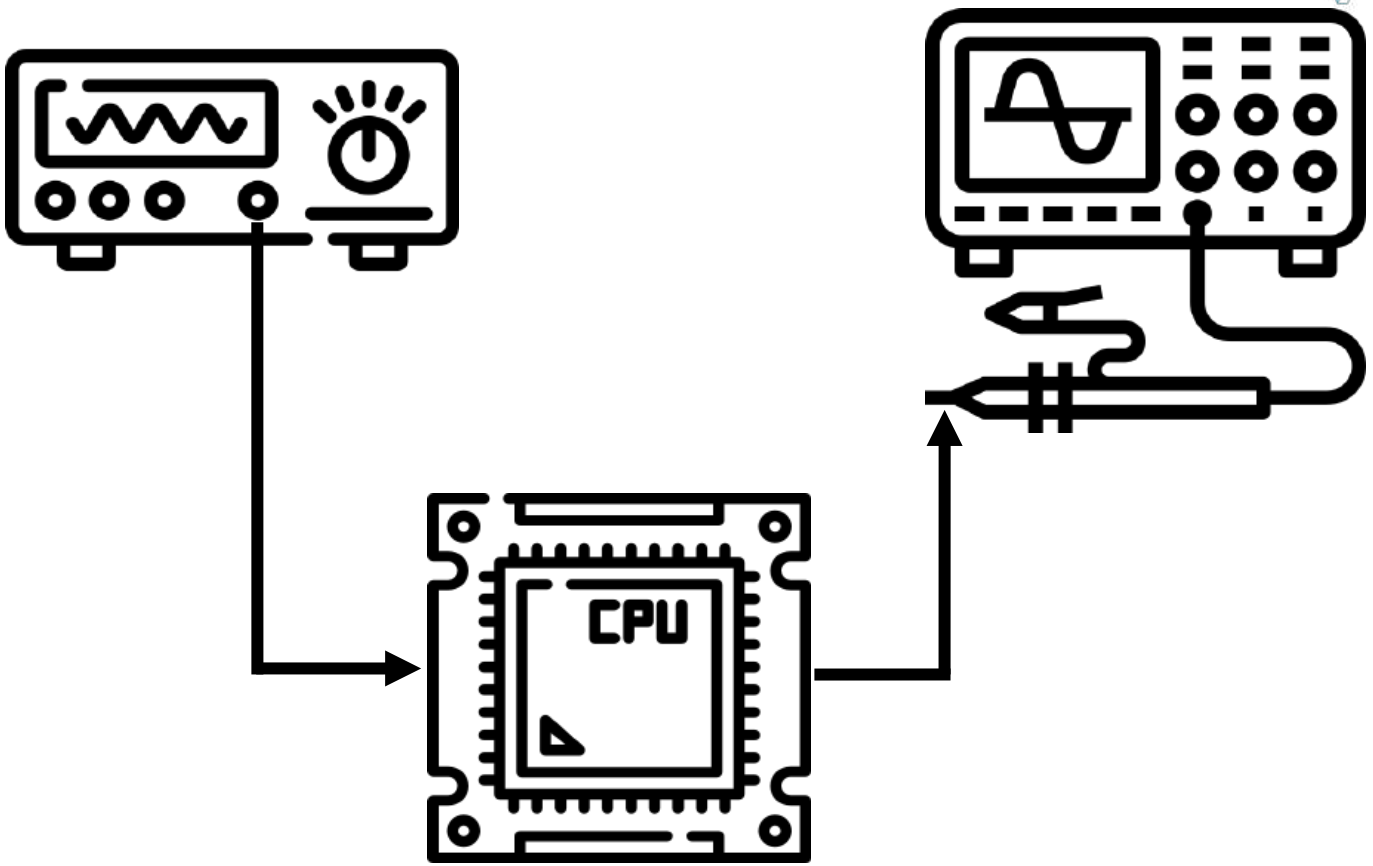
Задержка обработки прерывания



Задержка планирования и получения управления в пользовательском потоке

Стенд для замеров характеристик реального времени

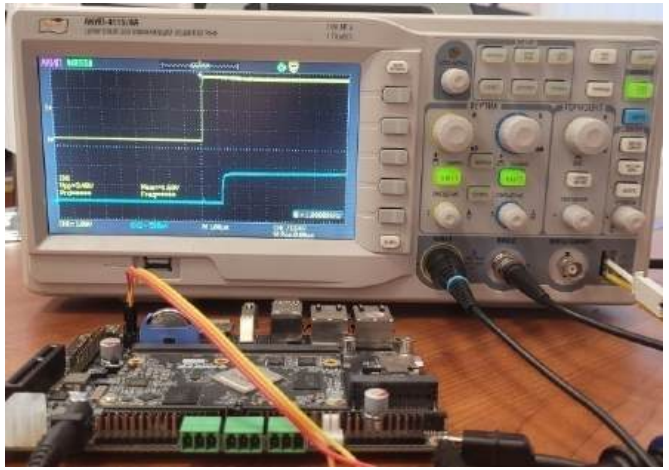
- **Генератор**
  - Прямоугольный импульс (меандр) с заданной частотой
  - Коэффициент заполнения 50%
  - Пакетный режим с заданием количества импульсов
- **Целевая система**
  - Вход: сигнал от генератора
  - Выход: сигнал для осциллографа
- **Осциллограф**
  - 1 вход: генератор
  - 2 вход: целевая система
  - Режим накопления данных

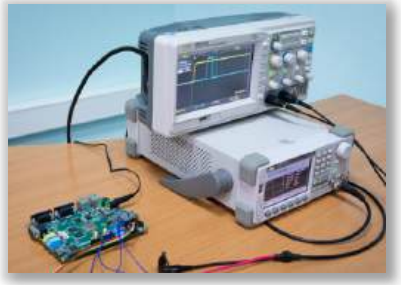


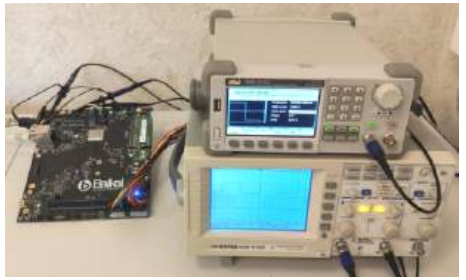
Иконки от flaticon.com

Стенд для замеров характеристик реального времени

- **Генератор АКИП-3413**
  - Пакетный Burst режим до 1 миллиона импульсов
- **Осциллограф АКИП-4115, АКИП 72204**
  - 2 канала
  - Режим накопления данных
- **Целевая система**
  - Процессорный модуль RK3568 с GPIO

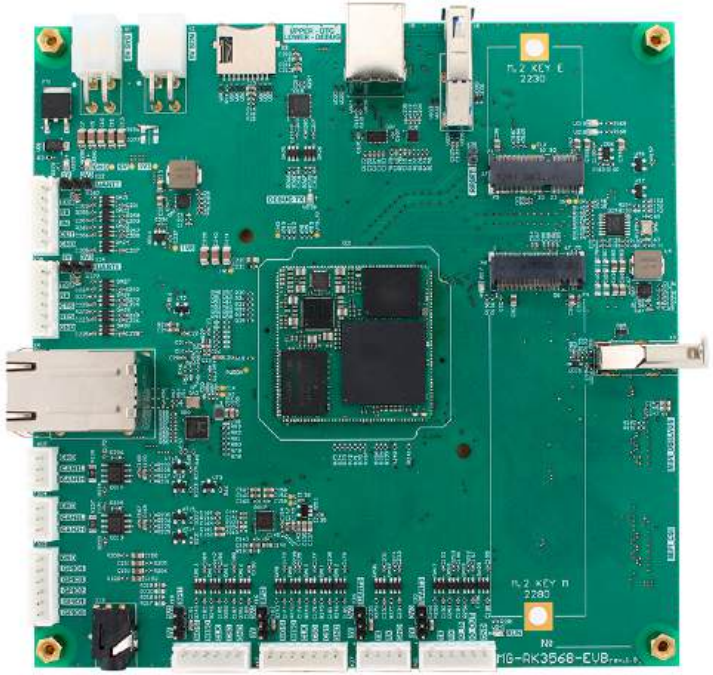






Целевая система: Diasom DS-RK3568-EVB

- **Четырехядерный процессор ARM Cortex A55**
  - Частота CPU 816МГц
- **Контроллер прерываний ARM GICv3**
  - Приоритеты прерываний
  - Распределение прерываний по ядрам
- **Состав тестового образа «Нейтрино»**
  - Ядро ОС
  - Минимальный набор сервисов
  - Тестовая программа работы с GPIO
- **Штатные прерывания системы**
  - Таймер 1мс
  - Межядерные IPI



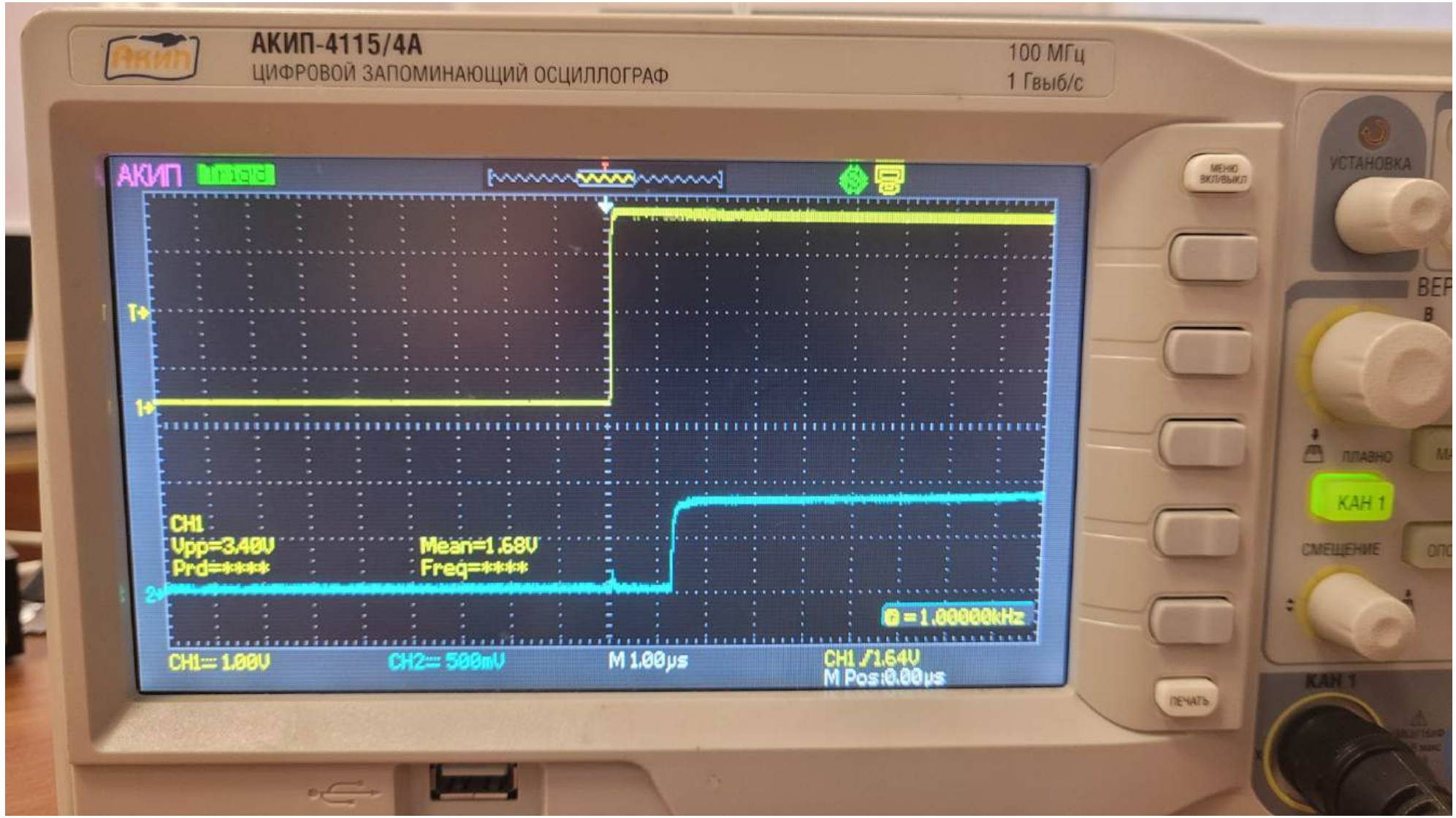
Программный цикл обработки

- **Обработчик прерывания**
  - Установка линии GPIO

- **Основной цикл**
  - Ожидание прерывания от входа GPIO
  - Эмуляция нагрузки (для более наглядного импульса на выходе)
  - 2 сценария: с обработчиком InterruptAttach и событием InterruptAttachEvent

```c
const struct sigevent *interrupt_handler (void *area, int id)
{
    gpio_pin_set(gpio_base, OUT_PIN_NUM);

    gpio_intr_clear(gpio_base, IN_PIN_NUM);

    current_irq_counter++;

    return (&int_event);
}


int main( int argc, char **argv )
{
    if (thread_test)
        id = InterruptAttachEvent( INTR_NUM, &int_event,_NTO_INTR_FLAGS_TRK_MSK);
    else
        id = InterruptAttach( INTR_NUM, &interrupt_handler, NULL, 0,  _NTO_INTR_FLAGS_TRK_MSK);

    if (!thread_test)
    {
        printf("Interrupt handler \n");
        while (current_irq_counter < total_irq_num)
        {
            if (InterruptWait(0, NULL) == -1)
                break;

            current_handler_counter++;

            if (busy_emulate)
                nanospin_ns(busy_time);

            gpio_pin_clear(gpio_base, OUT_PIN_NUM);
        }
    }
```

Стенд для замеров характеристик реального времени

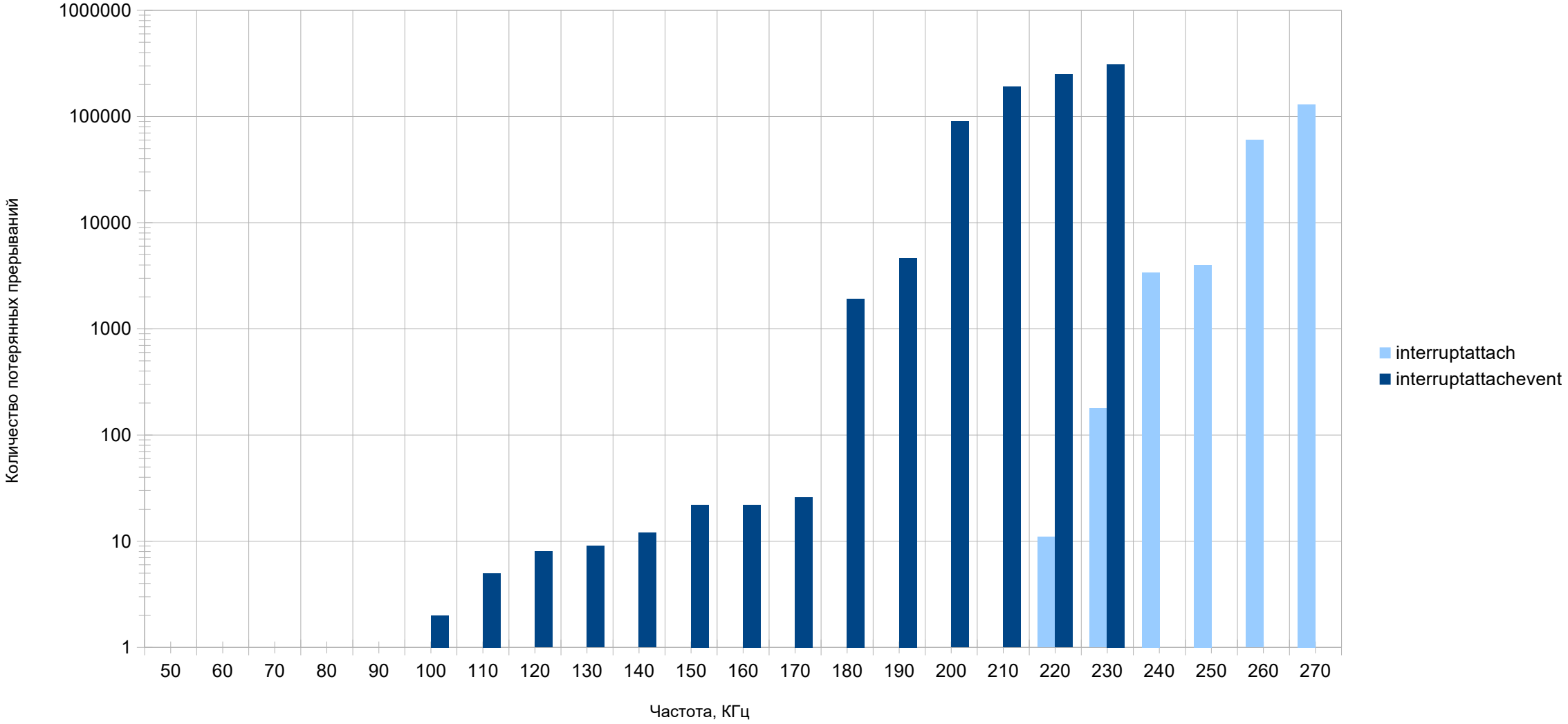


Замер характеристик реального времени

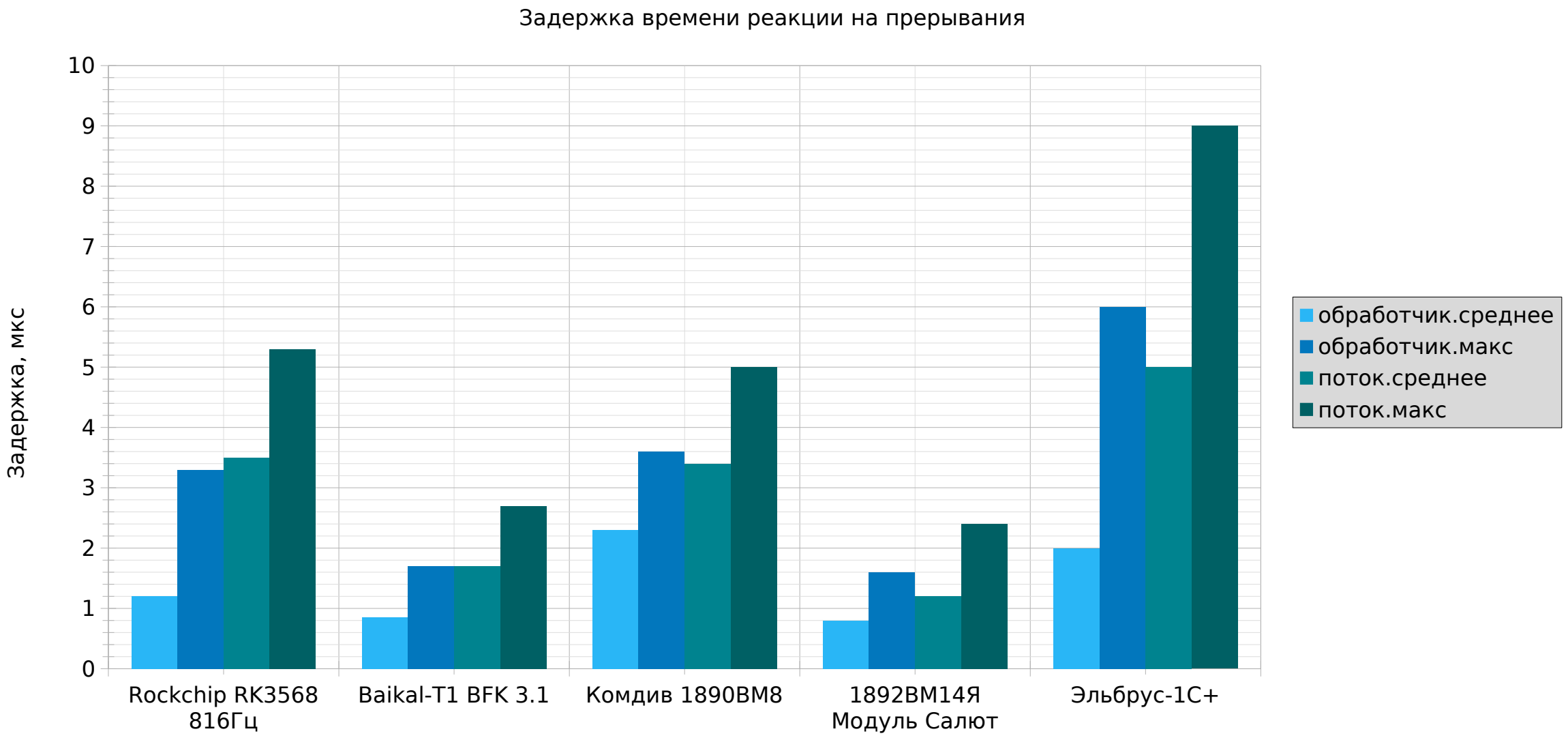
Характеристики реального времени «Нейтрино» на модуле Diasom DS-RK3568-EVB

Среднее время реакции на прерывание в обработчике	1 мкс
Максимальное время реакции на прерывание в обработчике	**3.3 мкс**
Среднее время реакции на прерывание в пользовательском потоке	3.5 мкс
Максимальное время реакции на прерывание в пользовательском потоке	**5.3 мкс**
Период обработки прерываний без потерь в обработчике	5 мкс (200кГц)
Период обработки прерываний без потерь в пользовательском потоке	11.1 мкс (90кГц)

Количество потерянных прерываний из 1 миллиона



Сравнение характеристик реального времени для разных аппаратных платформ



Распределение прерываний по разным процессорным ядрам

	Прерывание приходит на 1 ядро	Прерывание приходит на 0 ядро (по-умолчанию)
Среднее время реакции на прерывание в обработчике	1.9 мкс	1 мкс
Максимальное время реакции на прерывание в обработчике	**2.1 мкс**	3.3 мкс
Среднее время реакции на прерывание в пользовательском потоке	8 мкс	3.5 мкс
Максимальное время реакции на прерывание в пользовательском потоке	11 мкс	5.3 мкс
Период обработки прерываний без потерь в обработчике	**3.85 мкс (260кГц)**	5 мкс (200кГц)

Установка порога срабатывания прерываний для привилегированного пользовательского потока

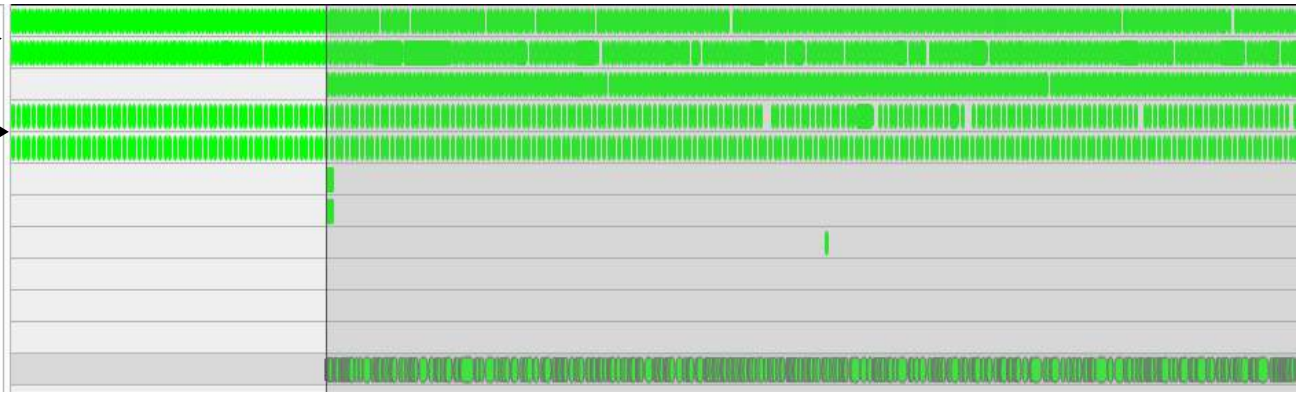
- **Установка порога срабатывания прерываний для привилегированного потока**
  - На основе приоритетов прерываний
  - Во время активного исполнения потока не срабатывают прерывания ниже заданного приоритета
  - При переключении контекста ядро сохраняет и восстанавливает порог срабатывания, т.е. другие потоки продолжают работать штатно
- **Организация критической секции кода**
  - Поток получает только требуемые прерывания
  - Увеличение получаемого процессорного времени
  - Повышение предсказуемости поведения
- **Подводные камни**
  - Запрет прерываний
  - Незначительное увеличение накладных расходов (включается опцией ядра)
  - Поддерживается для систем с GICv3 и Эльбрус

Установка порога срабатывания прерываний для привилегированного пользовательского потока

Прикладные прерывания

Прерывание таймера

Interrupt 0x98			
Interrupt 0x98	4108		
Interrupt 0x98	192528		
Interrupt 0x1b			
Interrupt 0x1b	1		
Interrupt 0x3b			
Interrupt 0x3b	6		
Interrupt 0x4dda40	2013973688		
proc/boot/procnto-smp-instr	1	0	0
proc/boot/mksh	49165	0	1
proc/boot/mksh->proc/boot/on->proc/boot/test_cpr	192528	0	49165
Thread 2	192528	2	192528

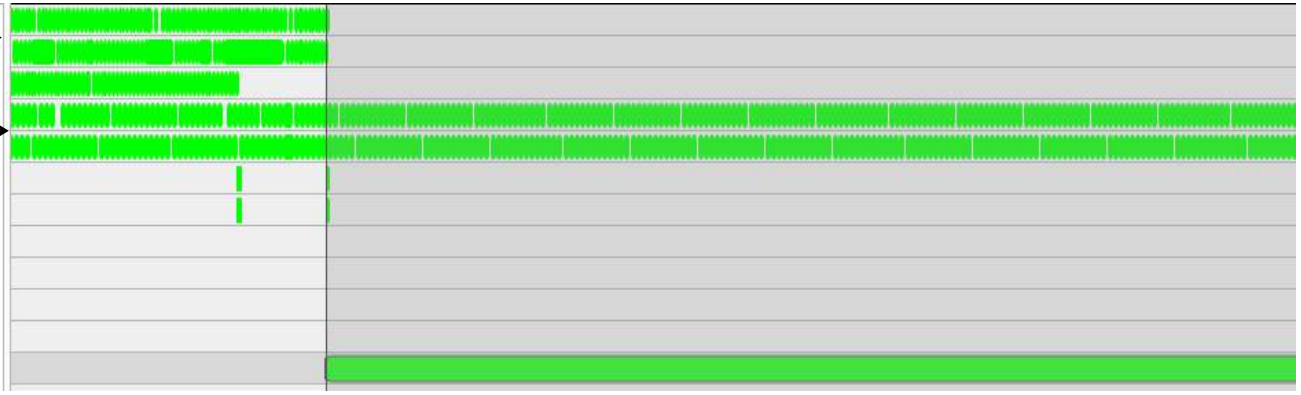


Прерываемый поток

Прикладные прерывания

Прерывание таймера

Interrupt 0x98			
Interrupt 0x98	4108		
Interrupt 0x98	192528		
Interrupt 0x1b			
Interrupt 0x1b	1		
Interrupt 0x3b			
Interrupt 0x3b	6		
Interrupt 0x4dda40	2013973688		
proc/boot/procnto-smp-instr	1	0	0
proc/boot/mksh	49165	0	1
proc/boot/mksh->proc/boot/on->proc/boot/test_cpr	204816	0	49165
Thread 2	204816	2	204816



TraceKev — инструмент анализа системной трассы событий

- **Специализированный инструмент**, позволяющий упростить анализ системных трасс, собранных с узлов под управлением ЗОСРВ «Нейтрино»
  - Используется технология т. н. «системного профилирования»
  - Разработка инструмента началась в 2023 году
- **Типовые задачи**:
  - Анализ характеристик реального времени приложений
  - Исследование процессов взаимодействия между компонентами программных комплексов
  - Поиск причин аномального поведения (пропущенные прерывания, нарушения
  - в циклограммах и т.д.)
- **Основные функции**
- **Визуальное отображение** информации о системных событиях, содержащейся в файлах .kev (Kernel Events)
  - Группировка, масштабирование, поиск и комплексная фильтрация отображаемой информации
  - Базовая аналитика (временные интервалы, взаимосвязь между событиями)
- **Основные характеристики**
  - Программное ядро и интерфейс собственной разработки (C++, Boost, Qt)
  - Высокая производительность операций визуализации и поиска/фильтрации
    - Возможность обработки больших системных трасс (сотни МБ)

TraceKev – трасса системных событий

Neutrino-System-Trace

Файл Инструменты Помощь

- Neutrino-System-Trace
  - Заголовок файла трассы
  - Таймлайн
  - Фильтры
    - Фильтр событий
    - Фильтр источников
    - Поиск событий

Таймлайн

Имя процесса	PID
Interrupt 0x1b	
Interrupt handler 0x1b	1
Interrupt 0x42	
Interrupt handler 0x42	282
proc/boot/procnto-smp-instr	1
Thread 1	1
Thread 2	1
Thread 3	1
Thread 4	1
Thread 6	1
Thread 7	1
Thread 8	1
Thread 10	1
Thread 11	1
Thread 13	1
Thread 14	1
Thread 15	1
Thread 16	1
Thread 17	1
Thread 18	1
proc/boot/slogger	819
Thread 1	819
proc/boot/pipe	819
Thread 1	819
Thread 2	819
Thread 3	819
proc/boot/random	819
Thread 1	819

8 00:03:58.088 023 125 00:03:58.094 108 041 00:03:58.100 192 958 00:03:58.106 277 875 00:03:58.112 362 791 00:03:58.118 447 708 00:03:58.124 532 625 00:03:58

Метка времени	CPU №	Источник события	Тип	Параметры
00:03:58.100 309 250	0	Interrupt 0x1b	int_enter	interruptNumber = 0x1b; kernelFlag = 0x600d94ec
00:03:58.100 309 541	0	Interrupt handler 0x1b pid: 1	int_handler_enter	pid = 1; interruptNumber = 0x1b; ip = 0x600ad7c0; area = 0x0
00:03:58.100 311 000	0	Interrupt handler 0x1b pid: 1	int_handler_exit	interruptNumber = 0x1b; sigEvent = INTR EVENT
00:03:58.100 313 041	0	Interrupt 0x1b	int_exit	interruptNumber = 0x1b; kernelFlag = 0x1
00:03:58.100 399 083	0	Interrupt 0x42	int_enter	interruptNumber = 0x42; kernelFlag = 0x600d94ec
00:03:58.100 400 541	0	Interrupt handler 0x42 pid: 282634	int_handler_enter	pid = 282634; interruptNumber = 0x42; ip = 0x401950; area = 0x0

00:03:58.100 313 041

Абсолютное

СВД
Встраиваемые Системы

Спасибо за внимание!

Владимир Махилёв

Руководитель отдела промышленных проектов

ул. Кузнецовская, д. 19,
г. Санкт-Петербург
+7 (812) 346-89-56
www.kpda.ru
support@kpda.ru