

# Занятие 4

## Тактирование приложений

### Разделы

- A. Цикл с тактированием (timed loop)
- B. Программное тактирование
- C. Аппаратное тактирование
- D. Отклик на событие

# Тактирование циклов

- Обеспечивает режим ожидания для предоставления процессорного времени потоками с низкими приоритетами
- Уменьшает джиттер приложения
- Программное или аппаратное тактирование цикла
  - Программный джиттер обычно больше чем аппаратный джиттер
  - Программный джиттер может составлять около 15 мкс

# Тактирование циклов

## Программное тактирование (для всех RT платформ)

- Время привязывается к таймеру операционной системы (миллисекундный таймер)
- Длительность итерации цикла задается при помощи функций ожидания  
Функции ожидания маскируют программный джиттер кода, но вносят свой собственный джиттер

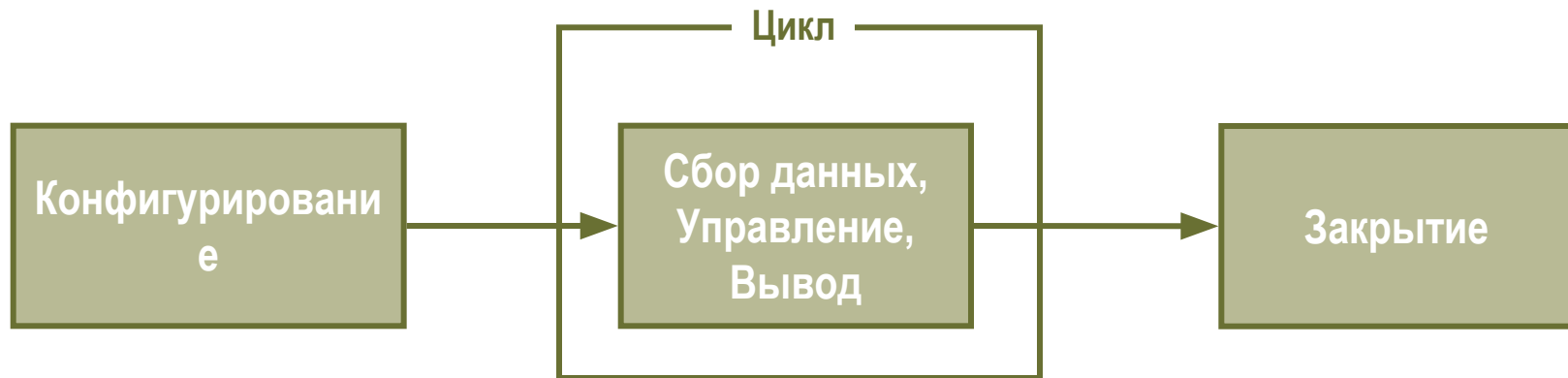
## Аппаратное тактирование (не доступно на [с]FP)

- Время привязывается к процессорному таймеру (микросекундный таймер) или к внешнему аппаратному таймеру
- Аппаратные таймеры не зависят от таймера операционной системы
- Аппаратный джиттер зависит от точности аппаратного таймера

# Программное тактирование

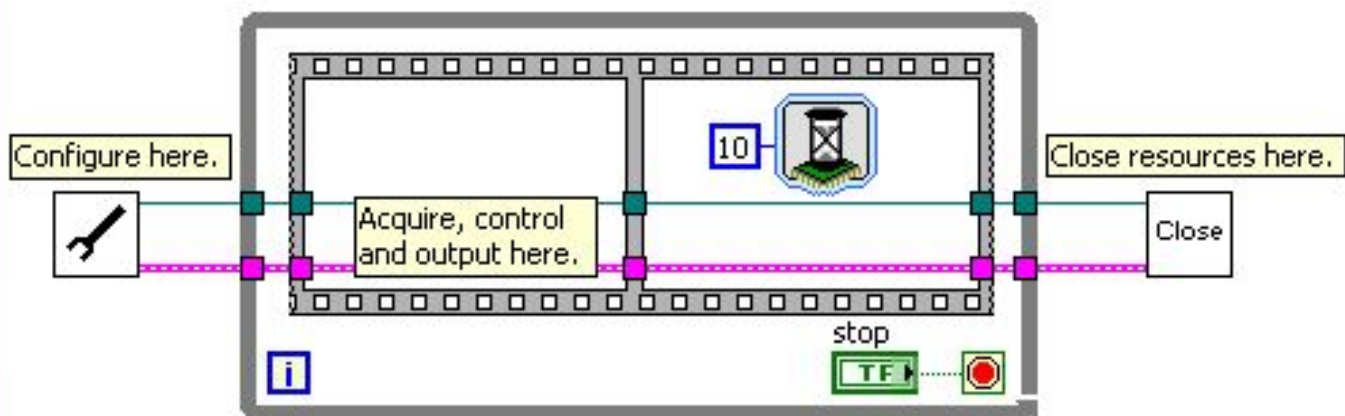
Три стандартных метода программного тактирования:

- Вставить функцию ожидания *Wait* в цикл
- Вставить функцию ожидания *Wait Until Next Multiple* в цикл
- Заместить обычный цикл **циклом с тактированием**

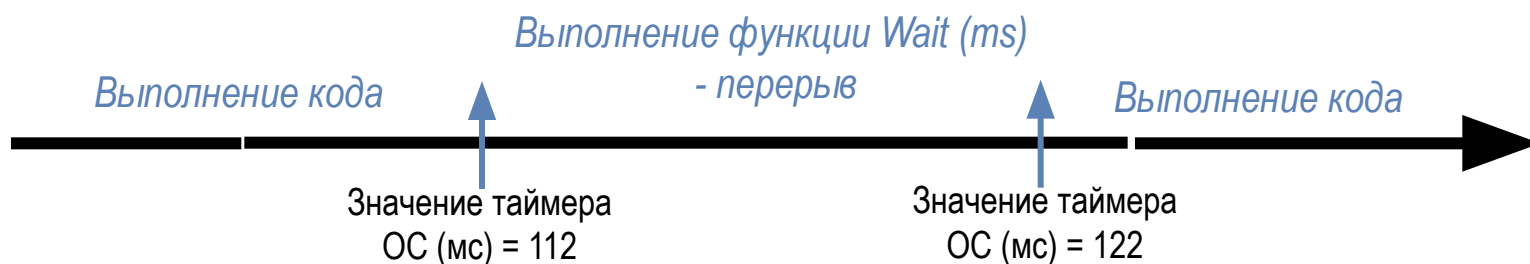


Архитектура приложения управления в модуле Real-Time

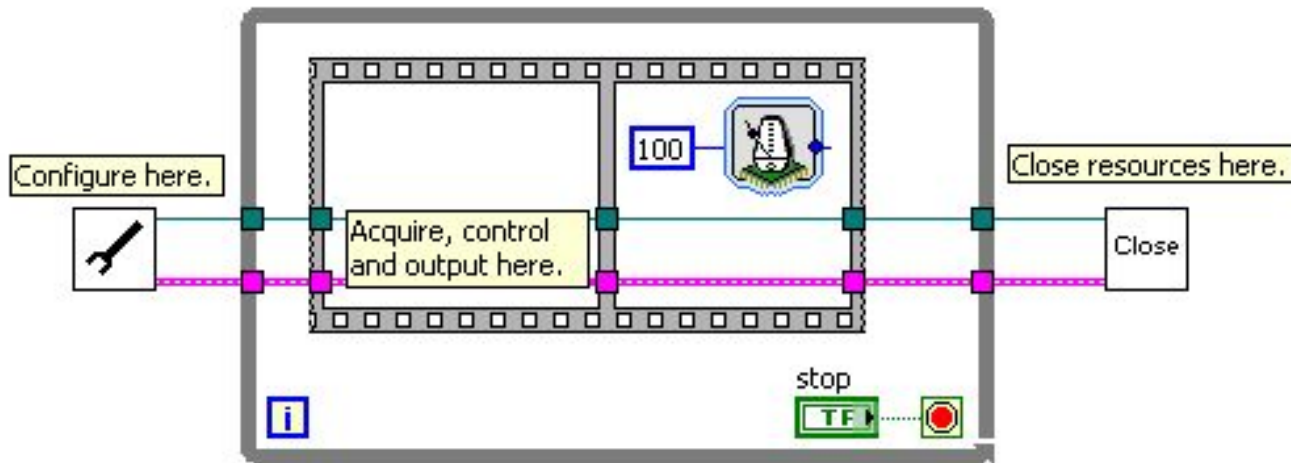
# Программное тактирование — функция Wait



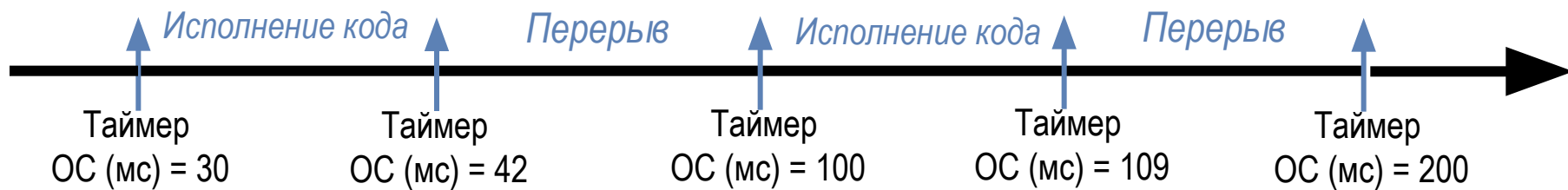
- Переводит VI в режим ожидания на заданное
- Не используйте параллельно с кодом критическим по времени
- Время исполнения кода может варьироваться, поэтому время выполнения цикла может также варьироваться



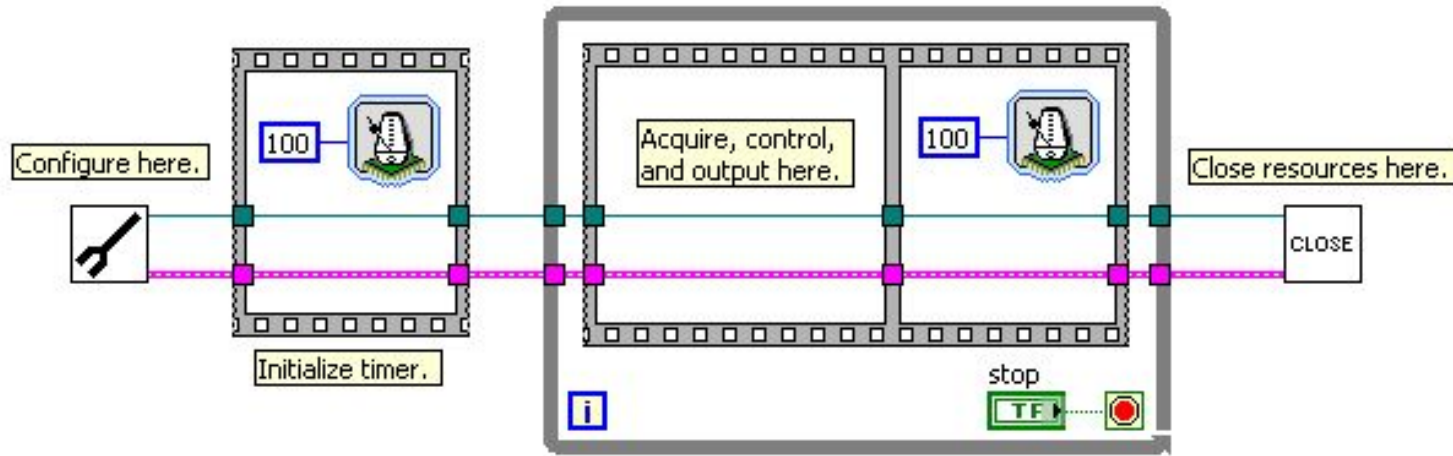
# Программное тактирование – функция Wait Until Next Multiple



- Поток прерывает свое исполнение и переходит в режим ожидания на время, кратное значению входа **Count (mSec)**
- Первая итерация цикла не определена



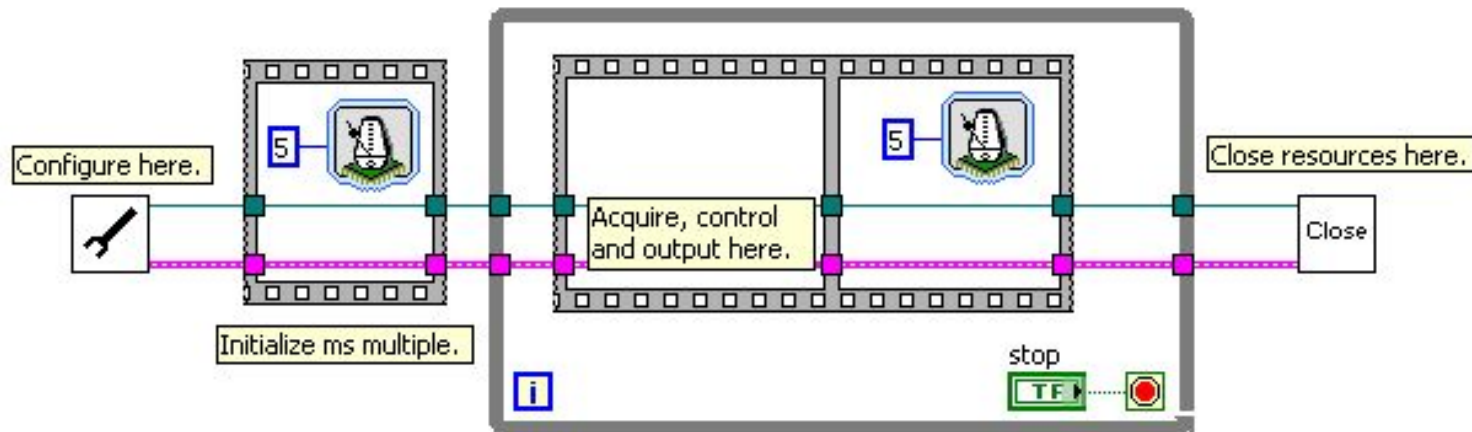
# Программное тактирование – Wait Until Next Multiple



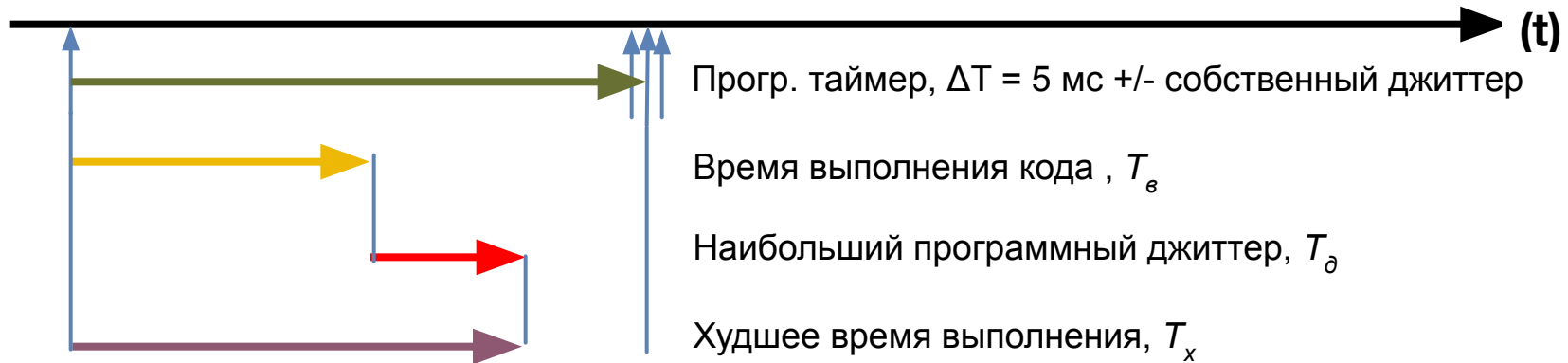
- Добавьте функцию Wait перед циклом для инициализации таймера
- Такой метод определяет время выполнения первой итерации цикла



# Программный тактирование—Wait Until Next Multiple



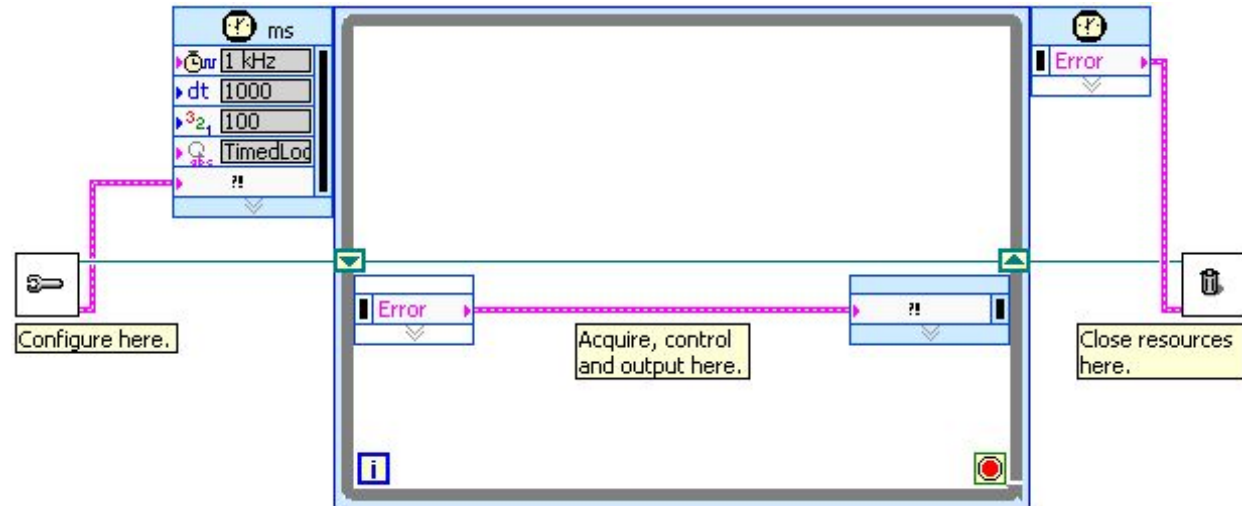
$T_x$  (худшее время выполнения) <  $\Delta T$  (кратное времени функции Wait + джиттер)





# Программный тактирование - цикл с тактированием

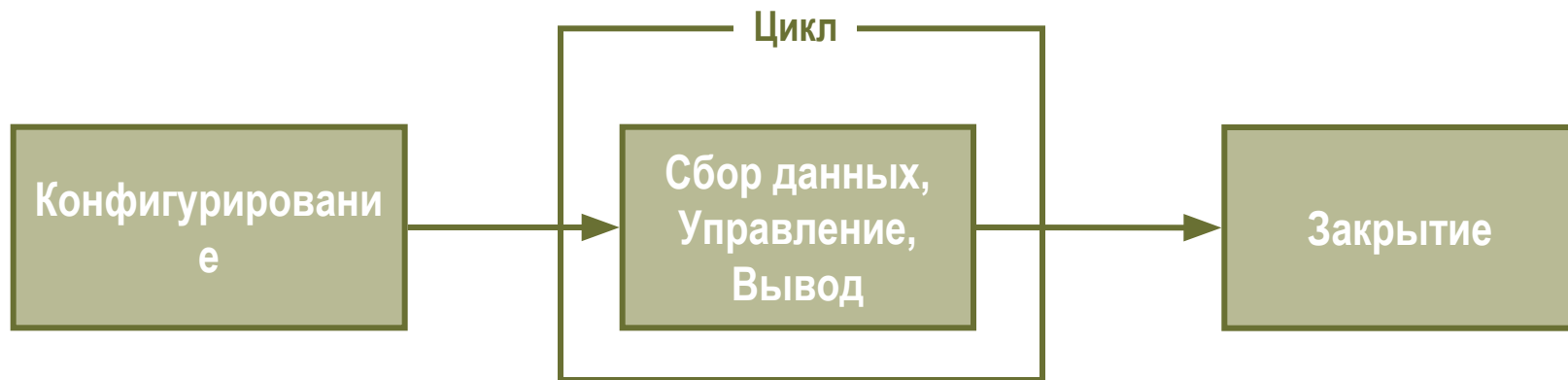
- При необходимости может использоваться вместо циклов For или While
- Выбор миллисекундного таймера означает к программное тактирование
- Следует использовать цикл с тактированием в следующих случаях:
  - Несколько циклов с разными частотами
  - Динамическое изменение параметров тактирования



# Аппаратное тактирование – не доступно для платформ [с]FieldPoint

Существуют четыре стандартных метода аппаратного тактирования

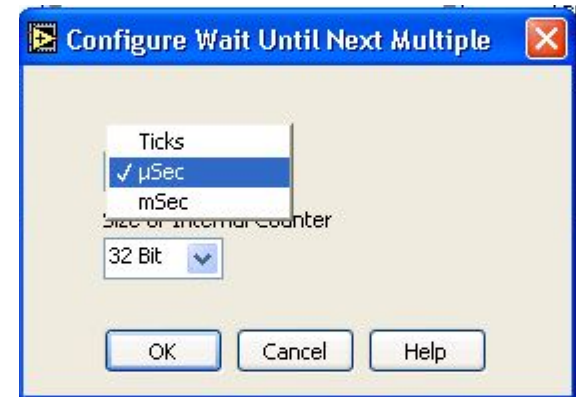
- Помещение в цикл функции Wait с мкс разрешением
- Помещение в цикл функции Wait Until Next Multiple с мкс разрешением
- Замена обычного цикла циклом с тактированием ( с мкс разрешением) или внешним таймером
- Использование внешнего таймера DAQmx



Архитектура приложения управления в модуле Real-Time

# Аппаратное тактирование – функции с мкс разрешением

- Выберите мкс таймер для функций Wait, Wait Until Next Multiple или для цикла с тактированием
- Доступные частоты цикла в этом случае составят: 1 МГц, 500 КГц, ~333КГц, 250 КГц, 200 КГц, ~167 КГц и т.д.
- Как и в случае программного тактирования, худшее время исполнения кода цикла должно быть меньше периода цикла  $\Delta T$
- Используйте архитектуру построения приложения, как и случае программного тактирования



# Аппаратное тактирование – цикл с тактированием

Выберите мкс таймер

Или  
подсоединитесь к  
внешнему  
таймеру

**Configure Timed Loop**

**Loop Timing Source**

Use Built-In Timing Source

Source Type

- 1 kHz Clock
- 1 MHz Clock**
- 1 kHz <reset at structure start>
- 1 MHz <reset at structure start>

Source name

1 MHz

Use Timing Source Terminal

**Loop Timing Attributes**

Period: 1000 us

Priority: 100

**Advanced Timing**

Deadline: -1 us

Timeout (ms): -1

Offset / Phase: 0 us

Loop name: L234756

**Action on Late Iterations**

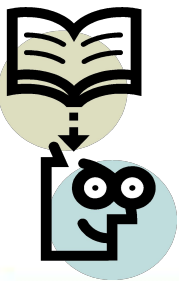
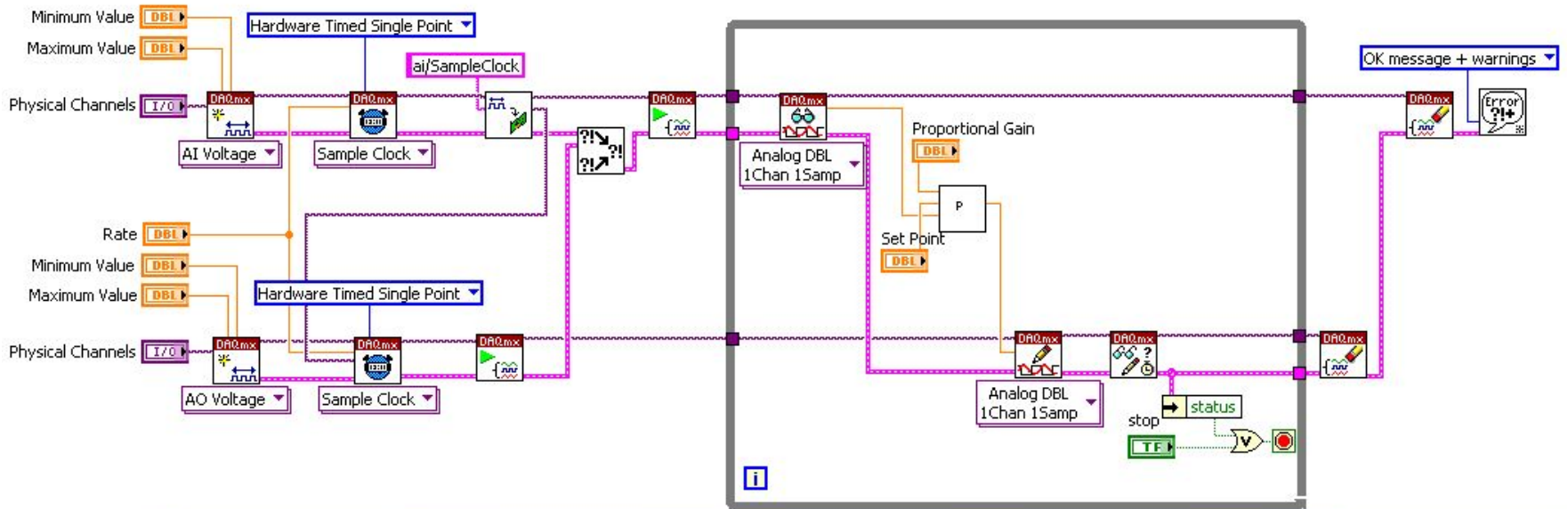
- Discard missed periods
- Maintain original phase

OK Cancel Help

**Frame Timing Source**

This structure does not have multiple frames. To add multiple frames, right click on the border of the loop and select one of the "Add Frame" menu items.

# Аппаратное тактирование — DAQmx



## Пример: NI Example Finder

Hardware Input and Output»DAQmx»Control»General»  
PID Control-Single Channel.vi

# Упражнение 4-1

Программный и аппаратный режим ожидания

Время на выполнение: 10 мин.

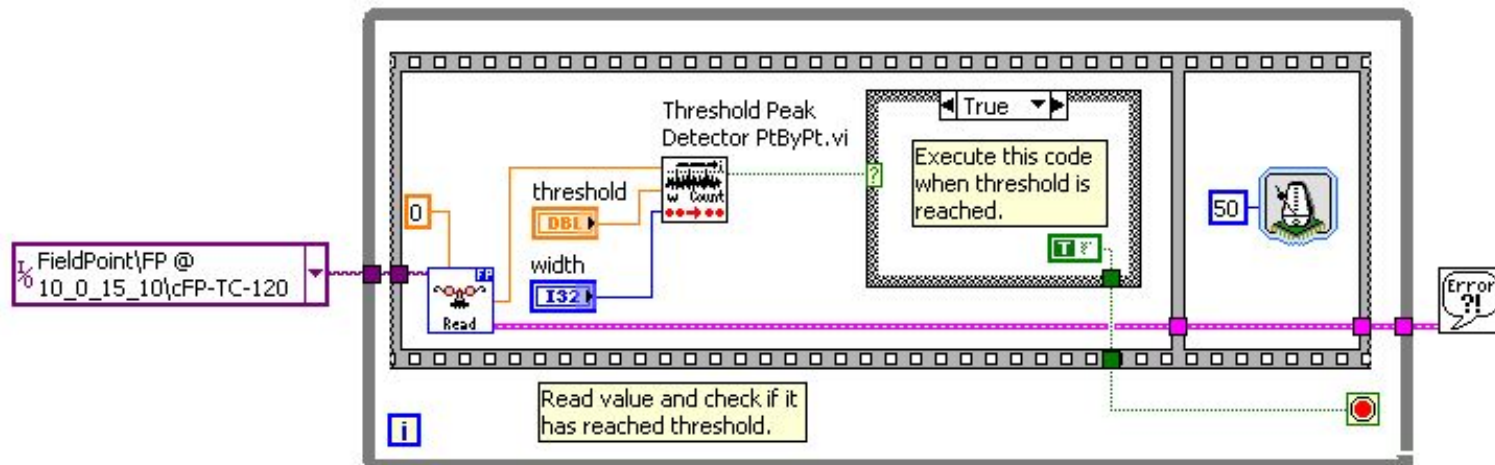
OBJECTIVE

Используйте существующий код для задания программного или аппаратного тактирования.

# Отклик на событие (event) – мониторинг событий

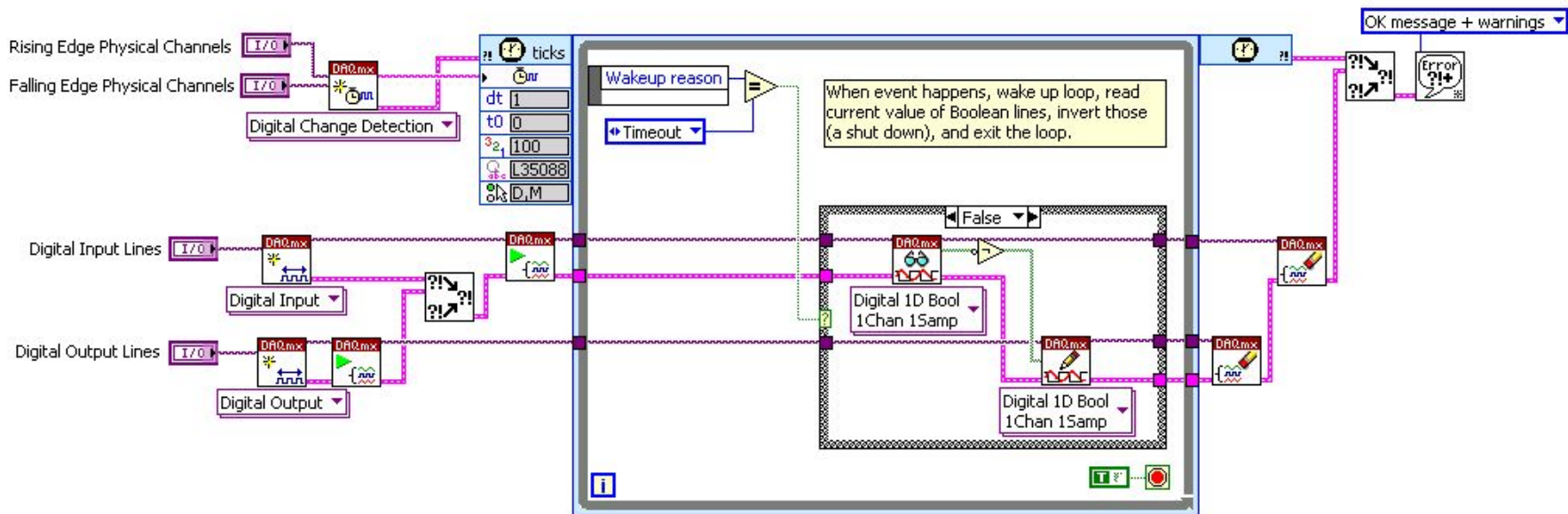
Используйте point-by-point VI для мониторинга следующих событий:

- Запись на жесткий диск по триггеру
- Возникновение триггера тревоги
- Запуск вычислений



# Отклик на событие – определение изменения состояния

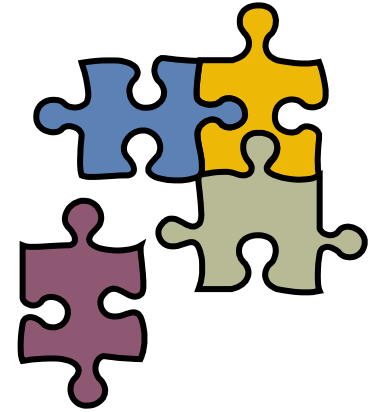
- Обработка обычных событий
- Необходимо использовать устройство цифрового ввода/вывода, которое поддерживает определение изменения состояния





# Упражнение 4-2

VI с критическим приоритетом



Время на исполнение: 60 мин.

OBJECTIVE

Разработка виртуального прибора с критическим приоритетом

# Резюме

- Какой таймер имеет лучшее разрешение – таймер процессора или таймер операционной системы?
- Какие преимущества имеет цикл с тактированием?