

---

**Драйвер файловой подсистемы**

Драйвер устройства файловой подсистемы отвечает за обработку файла, доступного после заданной контрольной точки, в глобальной файловой системе HAL.

**Создание элемента устройства**

Создание и регистрация файловой системы очень похожа на создание и регистрацию устройства с символьным режимом. Чтобы сделать доступной файловую систему, создайте элемент структуры `alt_dev` (см. "Драйверы устройств с символьным режимом" на стр. 7-5). Есть одно отличие в том, что имя поля устройства описывает контрольную точку файловой подсистемы. Конечно, вам нужно предоставить все необходимые функции доступа к файловой подсистеме, такие как `read()` и `write()`, также как и в случае с устройством с символьным режимом.

Если вы не предоставите реализацию для `fstat()`, то по умолчанию будет возвращаться значение для устройства с символьным режимом, которое является некорректным для файловой подсистемы.

**Регистрирование устройства файловой подсистемы**

Вы сможете зарегистрировать файловую подсистему, используя следующую функцию:

```
int alt_fs_reg (alt_dev* dev)
```

Эта функция имеет один входной аргумент, который является регистрируемой структурой устройства. При успешной регистрации, возвращаемое значение – нуль. Отрицательное возвращаемое значение означает, что устройство не зарегистрировано.

После того, как файловая подсистема зарегистрирована в файловой системе HAL, вы получаете доступ к нему через HAL API и стандартную библиотеку ANSI Си. Контрольная точка для файловой подсистемы – это имя, заданное в структуре `alt_dev`.

За дополнительной информацией обратитесь к главе "[Разработка программ с использованием слоя аппаратной абстракции](#)" в настольной книге программиста под Nios II.

**Драйверы устройств таймеров**

В этой секции описываются драйверы системного тактового сигнала и временной метки.

**Драйвер системного тактового сигнала**

Модели устройства тактового сигнала необходим драйвер для генерирования периодических тактовых импульсов. В системе может быть только один системный тактовый сигнал. Вы реализуете драйвер системного тактового сигнала в качестве процедуры обслуживания прерывания (ISR) для периферии таймера, которая генерирует периодическое прерывание. Драйвер должен предоставить периодический вызов следующей функции:

```
void alt_tick (void)
```

Ожидается, что `alt_tick()` вызывается в контексте исключения.

Для регистрирования наличия драйвера системного тактового сигнала, вызовите следующую функцию:

```
int alt_sysclk_init (alt_u32 nticks)
```

---

В аргументе `nticks` указывается количества системных тактов в секунду, которое определено вашим драйвером системного тактового сигнала. Возвращаемое значение – нуль при успехе, другом случае - не нуль.

За дополнительной информацией обратитесь к главе "[Обработка исключений](#)" в настольной книге программиста под Nios II.

### ***Драйвер временной метки***

Драйвер временной метки предлагает реализацию трёх функций временной метки: `alt_timestamp_start()`, `alt_timestamp()` и `alt_timestamp_freq()`.

В системе может быть только один драйвер временной метки.

За дополнительной информацией обратитесь к главе "[Разработка программ с использованием слоя аппаратной абстракции](#)" и к главе "[Справка по HAL API](#)" в настольной книге программиста под Nios II.