Спецификация Real Time Clock (RTC).

Оглавление

1. [Введение 1](#_bookmark0)
   1. [Назначение 1](#_bookmark1)
   2. [Основные функции 1](#_bookmark2)
2. [Функциональное описание 2](#_bookmark3)
   1. [Структурная схема 2](#_bookmark4)
3. [Внешние интерфейсы 2](#_bookmark5)
4. [Синхронизация и сброс 3](#_bookmark6)
5. [Карта памяти 4](#_bookmark7)
6. [Описание портов 7](#_bookmark8)
7. [Прерывания 8](#_bookmark9)
8. [Функциональное описание: режимы и сценарии работы 8](#_bookmark10)
   1. [Инициализация 8](#_bookmark11)
   2. [Сброс 8](#_bookmark12)
   3. [Описание работы 8](#_bookmark13)
9. [История версий 9](#_bookmark14)

# Введение

## Назначение

Real Time Clock (RTC) – цифровые часы/календарь, которые выводят секунды, минуты, часы в 24- и 12-часовом формате с индикацией AM/PM, число, день недели, месяц, год с учетом високосного. Имеет встроенный генератор тактового сигнала частотой 1 Гц, а также имеет возможность использования внешнего тактового сигнала. RTC имеет в своем составе блок обработчика прерываний, контролирующий их работу.

## Основные функции

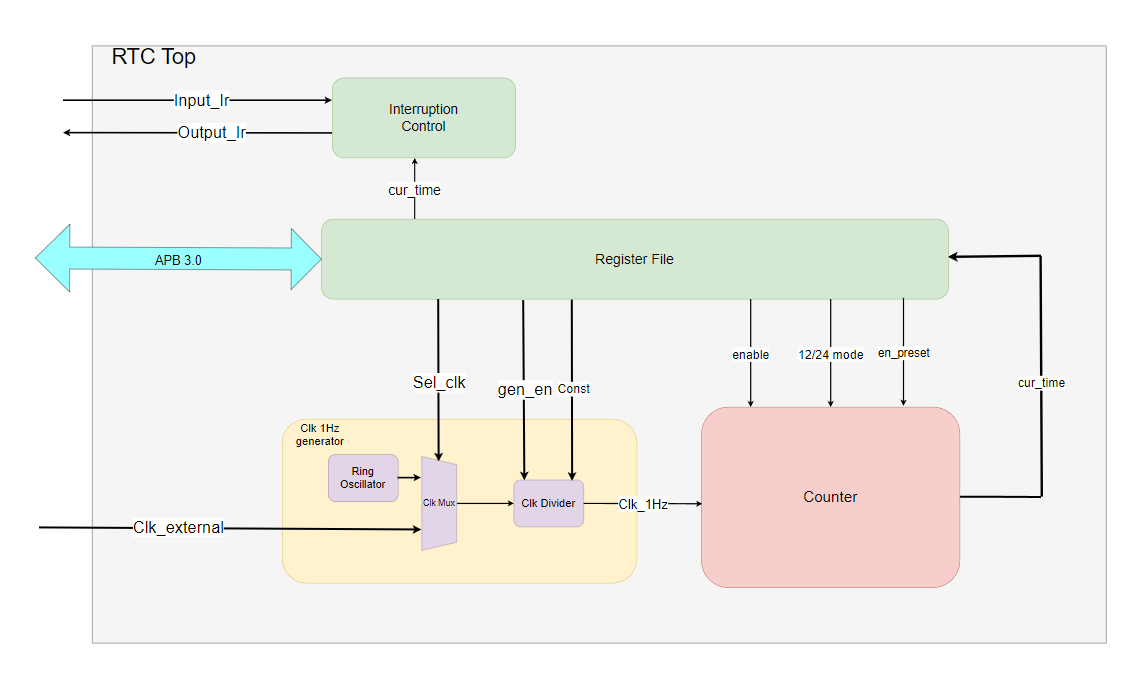
* + - RTC выводит секунды, минуты, часы в 24- и 12-часовом формате с индикацией AM/PM, число, день недели, месяц, год с учетом високосного.
    - Возможность установки точки отсчета.
    - Имеет внутренний генератор тактового сигнала 1 Гц.
    - Работа с внешними и внутренними прерываниями.
    - APB 3.0 interface.

# Функциональное описание

## Структурная схема

На рисунке 1 приведена структурная схема RTC. В его состав входит:

* + - Clk 1 Hz Generator – генератор тактового сигнала частотой 1 Гц;
    - Counter – счетчик времени и даты;
    - Register File – регистровый файл, хранящий в себе основные данные.
    - Interruption Control – блок обработки прерываний.



*Рисунок 1. Структурная схема RTC*

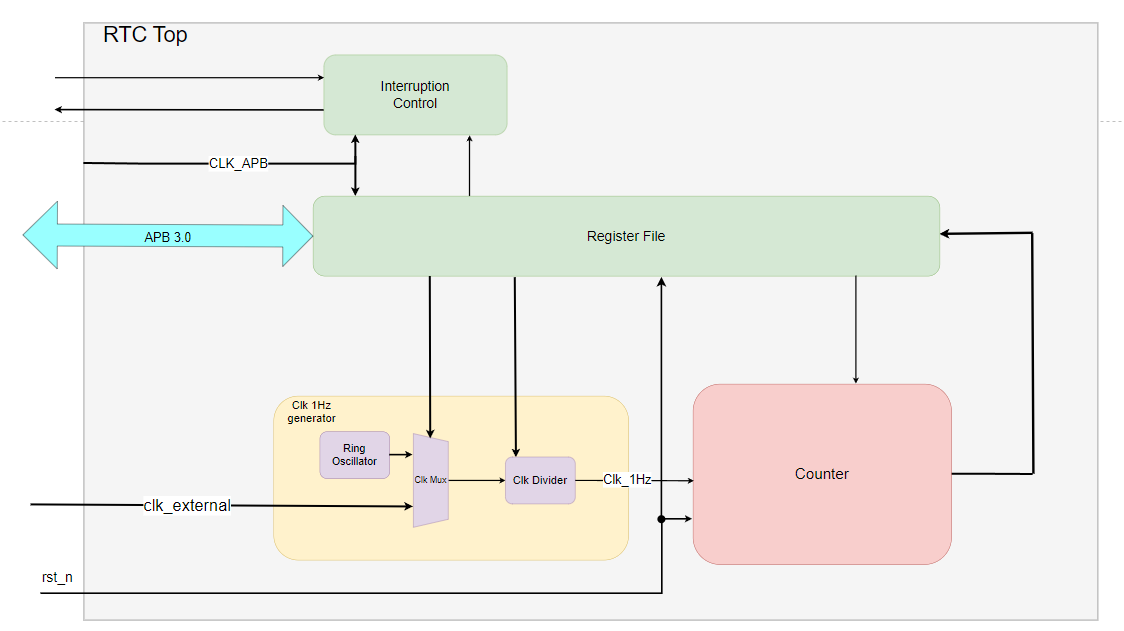
# Внешние интерфейсы

RTC использует интерфейс APB 3.0.

# Синхронизация и сброс

RTC использует несколько тактовых сигналов. Блок счетчика – counter использует сигнал clk\_1Hz, создаваемый встроенным генератором тактового сигнала, есть возможность использовать внешний тактовый сигнала clk\_external. Блок Register File использует тактовый сигнал интерфейса APB 3.0 – clk\_apb.

Сброс счетчика осуществляется внешним сигналом сброса rst\_n.



*Рисунок 2. Блок-схема тактирования и сброса*

# Карта памяти

В таблице 1 представлены адреса памяти RTC. Ширина слова – 32 бита.

*Таблица 1. Таблица адресов памяти устройства*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address Offset | Name | Description | Field | Width | Functionality |
| Enable register – contain enable signal for counter | | | | | |
| 0x00 | Enable\_reg | Enable register | Enable[0] | 1 | Enable signal for counter |
| Config register – contain configuration signals | | | | | |
| 0x04 | config\_reg | Config Register | gen\_en[0] | 1 | Clock generator enable signal (active high) |
| sel\_clk[1] | 1 | Clock select signal (0 –  internal, 1 -  external) |
| sel\_mode[2] | 1 | 12/24 mode select signal (0 – 24h, 1 – 12h) |
| en\_preset[3] | 1 | Enable setup preset point for counter (0 – disable, 1 – enable) |
| Current register – contain current time and date value | | | | | |
| 0x08 | cur\_sec\_reg | Current Seconds Register | Sec[5:0] | 6 | Current second (00-59) |
| 0x0C | cur\_min\_reg | Current Minutes Register | Min[5:0] | 6 | Current minute (00-59) |
| 0x10 | cur\_hours\_reg | Current Hours Register | Hour[4:0] | 5 | Current hour (00-24/1-12) |
| mode\_12\_24[5] | 1 | Current mode (0 – 24h, 1 – 12h) |
| mode\_AM\_PM[6] | 1 | AM/PM flag (active if mode\_12\_24 = 1,  0 – AM, 1 - PM) |
| 0x14 | cur\_day\_of\_week\_reg | Current Day of Week Register | day\_of\_week[2:0] | 3 | Current DoW (1-  7, 1 – Sunday etc…) |
| 0x18 | cur\_day\_of\_month\_reg | Current Day of Month Register | day\_of\_month[4:0] | 5 | Current DoM (01-  31, depends on current month) |
| 0x1C | cur\_month\_reg | Current Month Register | Month[3:0] | 4 | Current month (01-12) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0x20 | cur\_year\_reg | Current Year Register | Year[11:0] | 12 | Current year (0000-4096) |
| Constant register – contain constant for clk divider to achieve 1 Hz clock signal | | | | | |
| 0x24 | const\_reg | Constant for divider register | Const[31:0] | 32 | Constant for divider |
| Initial register – contain initial value for counter | | | | | |
| 0x28 | init\_sec\_reg | Initial Seconds Register | Sec[5:0] | 6 | Initial second (00-59) |
| 0x2C | init\_min\_reg | Initial Minutes Register | Min[5:0] | 6 | Initial minute (00-59) |
| 0x30 | init\_hours\_reg | Initial Hours Register | Hour[4:0] | 5 | Initial hour (00- 24/1-12) |
| mode\_12\_24[5] | 1 | Initial mode (0 – 24h, 1 – 12h) |
| mode\_AM\_PM[6] | 1 | Initial AM/PM flag (active if mode\_12\_24 = 1,  0 – AM, 1 - PM) |
| 0x34 | init\_day\_of\_week\_reg | Initial Day of Week Register | day\_of\_week[2:0] | 3 | Initial DoW (1-7, 1 – Sunday etc…) |
| 0x38 | init\_day\_of\_month\_reg | Initial Day of Month Register | day\_of\_month[4:0] | 5 | Initial DoM (01-  31, depends on current month) |
| 0x3C | init\_month\_reg | Initial Month Register | Month[3:0] | 4 | Initial month (01-  12) |
| 0x40 | init\_year\_reg | Initial Year Register | Year[11:0] | 12 | Initial year (0000-4096) |
| Interrupt input register – contain data saved from the counter when an external interrupt signal appears | | | | | |
| 0x44 | ir\_in\_sec\_reg | Interrupt input Seconds Register | Sec[5:0] | 6 | Saved second (00-59) |
| 0x48 | ir\_in\_min\_reg | Interrupt input Minutes Register | Min[5:0] | 6 | Saved minute (00-59) |
| 0x4C | ir\_in\_hours\_reg | Interrupt input Hours Register | Hour[4:0] | 5 | Saved hour (00- 24/1-12) |
| mode\_12\_24[5] | 1 | Saved mode (0 – 24h, 1 – 12h) |
| mode\_AM\_PM[6] | 1 | Saved AM/PM flag (active if mode\_12\_24 = 1,  0 – AM, 1 - PM) |
| 0x50 | ir\_in\_day\_of\_week\_reg | Interrupt input Day of | day\_of\_week[2:0] | 3 | Saved DoW (1-7, |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Week Register |  |  | 1 – Sunday etc…) |
| 0x54 | ir\_in\_day\_of\_month\_reg | Interrupt input Day of Month Register | day\_of\_month[4:0] | 5 | Saved DoM (01-  31, depends on current month) |
| 0x58 | ir\_in\_month\_reg | Interrupt input Month Register | Month[3:0] | 4 | Saved month (01-12) |
| 0x5C | ir\_in\_year\_reg | Interrupt input Year Register | Year[11:0] | 12 | Saved year (0000-4096) |
| Interrupt output register – contain time point data, when the value on the counter turns out to be equal to the value in these registers, an interrupt signal is generated | | | | | |
| 0x60 | ir\_out\_sec\_reg | Interrupt output Seconds Register | Sec[5:0] | 6 | Ir second (00-  59) |
| 0x64 | ir\_out\_min\_reg | Interrupt output Minutes Register | Min[5:0] | 6 | Ir minute (00-59) |
| 0x68 | ir\_out\_hours\_reg | Interrupt output Hours Register | Hour[4:0] | 5 | Ir hour (00-24/1- 12) |
| mode\_12\_24[5] | 1 | Ir mode (0 – 24h, 1 – 12h) |
| mode\_AM\_PM[6] | 1 | Ir AM/PM flag (active if mode\_12\_24 = 1,  0 – AM, 1 - PM) |
| 0x6C | ir\_out\_day\_of\_week\_reg | Interrupt output Day of Week Register | day\_of\_week[2:0] | 3 | Ir DoW (1-7, 1 –  Sunday etc…) |
| 0x70 | ir\_out\_day\_of\_month\_reg | Interrupt output Day of Month Register | day\_of\_month[4:0] | 5 | Ir DoM (01-31,  depends on current month) |
| 0x74 | ir\_out\_month\_reg | Interrupt output Month Register | Month[3:0] | 4 | Ir month (01-12) |
| 0x78 | ir\_out\_year\_reg | Interrupt output Year Register | Year[11:0] | 12 | Ir year (0000-  4096) |

# Описание портов

В таблице 2 представлены порты ввода-вывода RTC.

*Таблица 2. Таблица портов ввода-вывода устройства*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Port | Width | Direction | Clock | Description |
| Clocks | | | | |
| clk\_apb\_i | 1 | Input | - | APB clock signal |
| clk\_external | 1 | Input | - | External clock signal for counter |
| clk\_gen | 1 | - | - | Ring Oscillator clock signal |
| Resets | | | | |
| rst\_n\_i | 1 | Input | - | Counter reset signal, active low |
| Bus APB | | | | |
| paddr | 32 | Input | clk\_apb\_i | Master address data |
| psel | 1 | Input | clk\_apb\_i | Master select slave |
| penable | 1 | Input | clk\_apb\_i | Master enable |
| pwrite | 1 | Input | clk\_apb\_i | Master type transaction(write/read) |
| pwdata | 32 | Input | clk\_apb\_i | Master write data |
| prdata | 32 | Output | clk\_apb\_i | Slave read data |
| pready | 1 | Output | clk\_apb\_i | Slave read data ready |
| Interruptions | | | | |
| Input\_Ir | 1 | Input | clk\_apb\_i | Input interruption |
| Output\_Ir | 1 | Output | clk\_apb\_i | Output interruption |

# Прерывания

В RTC предусмотрено два типа прерываний:

1. Генерация прерывания Output\_Ir наружу при равенстве полей Current Register и Interrupt Output Register.
2. При поступлении сигнала прерывания Input\_Ir извне сохраняется значение из полей Current Register в Interrupt Input Register.

# Функциональное описание: режимы и сценарии работы

## Инициализация

Блок переходит в рабочее состояние при неактивном уровне сигнала сброса rst\_n (1), активном уровне сигнала enable (1) – сигнал активации счетчика, активным уровнем сигнала gen\_en (1) – включения генератора тактового сигнала. Перед включением сигнала enable необходимо задать режим счета и отображения времени (mode). Если начальная точка отсчета не установлена (en\_preset = 0), то при включении сигнала enable, счетчик установится в стандартное значение в зависимости от состояния управляющего сигнала mode – 01.01.2000 Sunday 00.00.00 (24h, если mode = 0), или 01.01.2000 Sunday 12.00.00 AM (12h, если mode = 1), для того чтобы задать стартовую точку необходимо установить сигнал en\_preset = 1, после чего в счетчик запишется начальная точка из Register File (Initial register).

## Сброс

Сброс RTC осуществляется внешним сигналом rst\_n (активный уровень 0).

## Описание работы

После этапа инициализации, счетчик работает по внутреннему тактовому сигналу clk\_1Hz, и записывает текущее время в Register File (Current register).

Возможно использовать либо внутренне сгенерированный тактовый сигнал, либо внешний тактовый сигнал clk\_external. Выбор источника тактового сигнала осуществляется сигналом sel\_clk. При изменении тактового сигнала необходима подстройка частоты, путем изменения константы для делителя частоты.

# История версий

1.0