# Rate Divider
# Gateware Documentation

Version 2.0

February 9, 2023

René Geißler
r.geissler@gsi.de

GSI Helmholtzzentrum für Schwerionenforschung GmbH

# Contents

# Resources

The code of this project and also the source of this documentation are under version control in a Git repository whose upstream is:
https://git.gsi.de/BEA_HDL/Rate_Divider_Gateware.
The relevant branch is *master*.

Common code is included as a Git submodule of the main Git repository. The upstream of the submodule is:
https://git.gsi.de/BEA_HDL/FPGA_Common.
The relevant branch is *master*.

Installation scripts to set up a Gitlab runner for continuous integration including all necessary software to build the gateware can be found in a Git repository whose upstream is:
https://git.gsi.de/BEA_HDL/Gitlab_Runner_Setup_Centos_7.
The relevant branch is *master*.

# 1 Introduction

This document describes the gateware (= FPGA firmware) implementation of the Rate Divider.

The Rate Divider is intended to be be used as an add-on for a Struck SFMC-AD-2500-14-2 2.5 GSPS ADC card.

The Rate Divider generates the trigger pulses for the Struck card by gating the incoming RF pulses. This allows the Struck card to strobe only the data of interest.

Vaibhav Jain's original documentation can be found here:
https://git.gsi.de/va.jain/rate_divider_sis18/-/wikis/Rate-Divider-for-SIS18

Harald Bräuning's original specification can be found here:
https://git.gsi.de/BEA_HDL/Rate_Divider_Gateware/-/blob/master/doc/LogicModuleSpec.pdf

# 2 Common FPGA based projects documentation

This project incorporates the code from the *FPGA_Common* Git repository which is used in multiple projects. The documentation of the common features can be found here:

## 2.1 Common monitoring and control features

Documentation about the register bank, the architecture information storage and the observer can be found here:
https://git.gsi.de/BEA_HDL/FPGA_Common#2-common-monitoring-and-control-features

## 2.2 FPGA Observer

There is a expert GUI that can be used together with multiple projects:
https://git.gsi.de/BEA_HDL/FPGA_Common#3-fpga-observer

## 2.3 Build flow and simulation

You can find instructions on how to build and simulate the gateware here:
https://git.gsi.de/BEA_HDL/FPGA_Common#4-build-flow-and-simulation

## 2.4 Helper scripts

You can find usefull scripts here:
https://git.gsi.de/BEA_HDL/FPGA_Common#5-helper-scripts

## 2.5 Continuous integration environment

Information about the continuous integration setup can be found here:
https://git.gsi.de/BEA_HDL/FPGA_Common#6-continuous-integration-environment

## 2.6 Programming and hardware configuration

You can find instructions on how to program the FPGA and configure other hardware here:
https://git.gsi.de/BEA_HDL/FPGA_Common#7-programming-and-hardware-configuration

# 3 Theory of operation

The gateware generates an internal gate signal, which controls the generation of trigger pulses out of incoming RF pulses. The rate of the pulses can be reduced by any number in the range between 1 and $2^{32}$.

Only complete RF pulses will be passed to the output:

- if the RF signal is high when the gate starts, the first incomplete pulse will be ignored
- if the RF signal is high when the gate stops, the gate will be extended until the RF pulse is completed

Due to the FPGA processing, the pulses of the trigger output will be delayed relative to the RF input pulses by one clock cycle of 8 ns. Additionally, there is a maximum strobing delay of one clock cycle, so that the total digital processing delay ranges between 8 ns and 16 ns. The same applies to the generation of the gate signal after receiving a signal from the FTRN.

Input signal states that are shorter than 8 ns cannot be detected reliably by the gateware.

The gate signal can be generated in two different ways determined by the configuration register *gate mode*:

## 3.1 Variable length gate mode

In this mode, the duration of the gate signal is determined by the high time of the FTRN input signal.

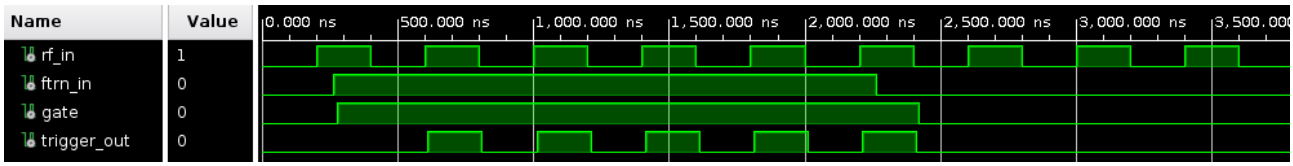


Figure 3.1: Variable length gate mode, no rate division, no delay

As you can see, the incomplete pulse at the beginning is ignored and the gate signal is extended to let the last begun pulse pass completely. In the following figure, a division of two is configured.

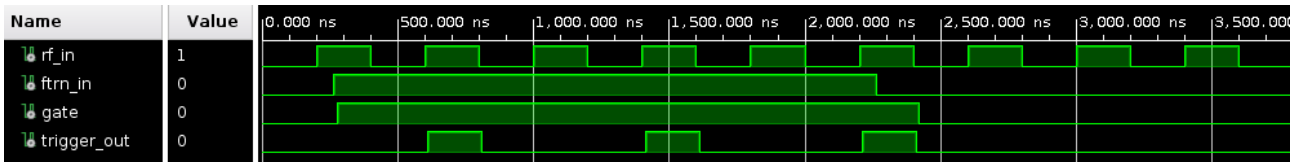


Figure 3.2: Variable length gate mode, division by 2, no delay

In the following figure, additional start and stop delays to the gate are configured.

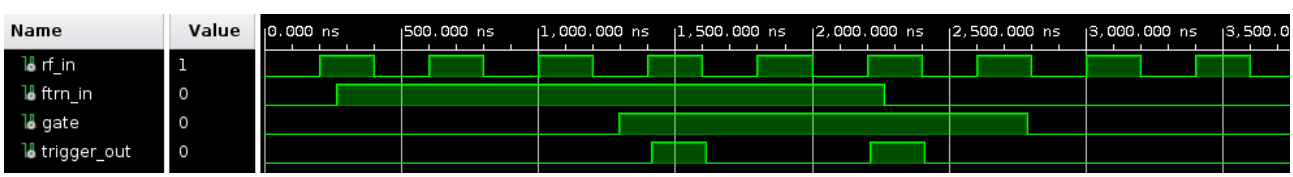


Figure 3.3: Variable length gate mode, division by 2, start delay 1042 ns, stop delay 512 ns

## 3.2 Fixed length gate mode

In this mode, the duration of the gate signal is determined by the configuration register *gate length*.

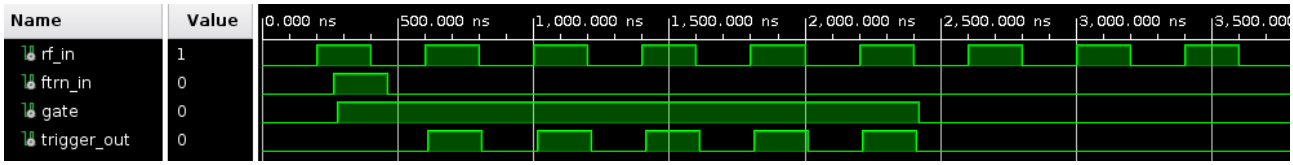


Figure 3.4: Fixed length gate mode, gate length 2048 ns, no rate division, no delay

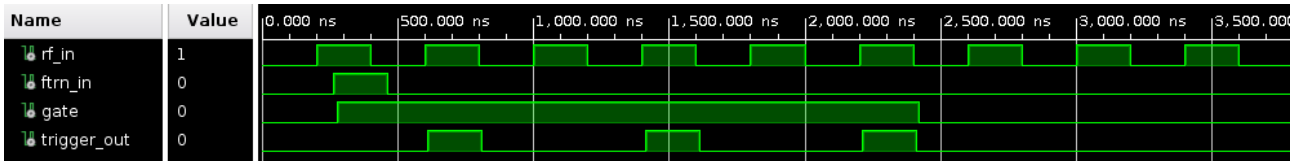


Figure 3.5: Fixed length gate mode, gate length 2048 ns, division by 2, no delay

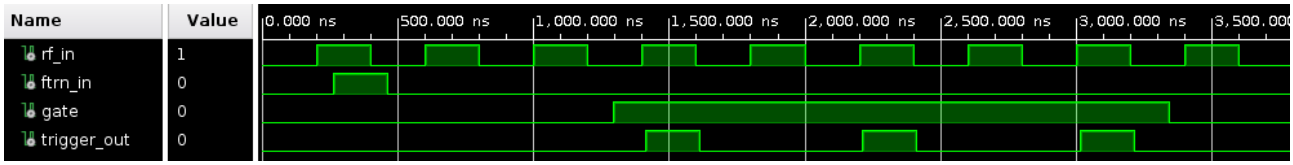


Figure 3.6: Fixed length gate mode, gate length 2048 ns, division by 2, start delay 1024 ns

# 4 Connections

## 4.1 FMC DIO LEMO Connectors

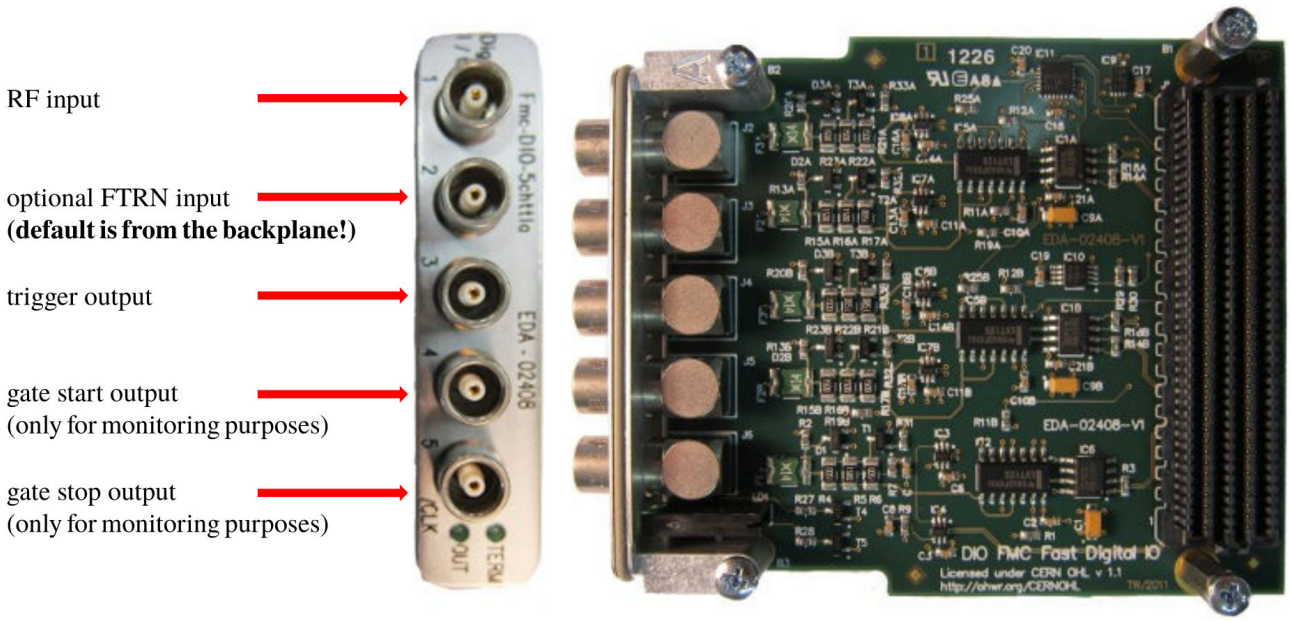


Figure 4.1: FMC DIO connections

## 4.2 Backplane

The following two input signals can be connected via the backplane:

| MLVDS line | signal | description |
|---|---|---|
| 0 | FTRN input | signal coming from the Timing Receiver, from which the gate signal is derived |
| 1 | optional RF input | only for testing purposes, enabled by the register *RF input select* |

As a default, the FTRN signal is configured to be driven by MLVDS line 0. The FTRN signal can optionally be driven by LEMO connector 2 by changing the register *FTRN input select*.

# 5 Gateware software interface

## 5.1 Necessary MCH settings

There are two different AFC versions used for the Rate Dividers. Version 2.0 connects to the PCIe lanes 8..11, whereas version 3.1 connects to the lanes 4..7.

To support both of them, the following settings should be applied to the MCH:

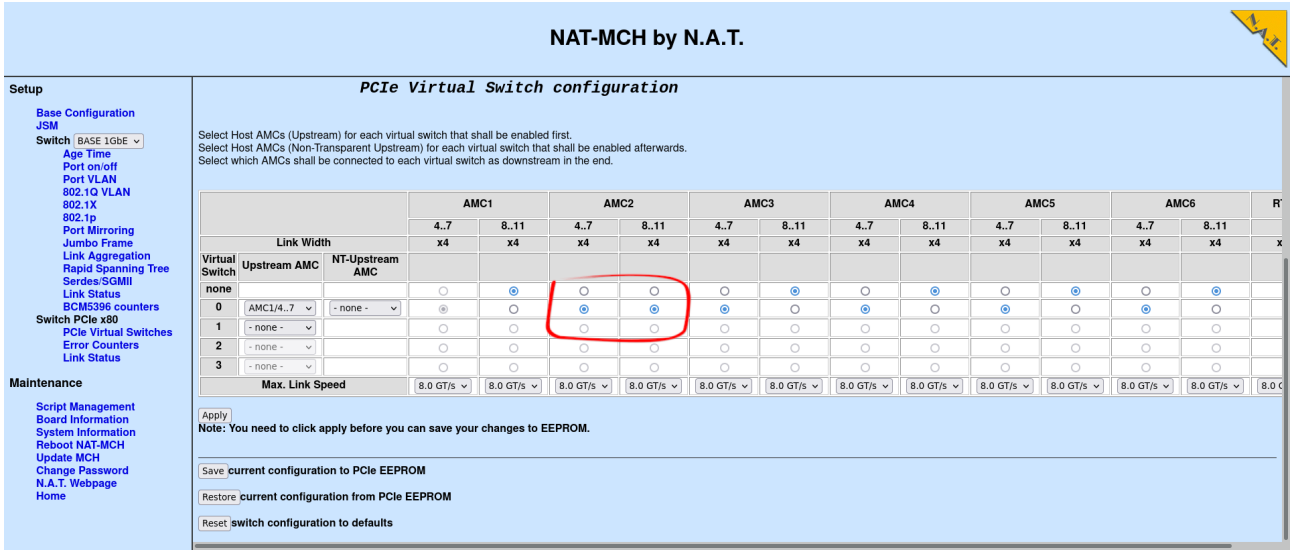


Figure 5.1: MCH settings

Keep in mind that changing the PCIe settings will also affect the ethernet port numbering of the FEC.

If you have defined a custom boot image in which the ethernet port was set to e.g. `ip=enp15s0:dhcp`, you might have to change it to `ip=enp16s0:dhcp`.

## 5.2 PCIe Driver

The communication between the gateware and the software takes place via a PCIe driver by Xilinx called XDMA.

Read and write accesses are mapped to virtual file accesses:

- `/dev/xdma0_c2h_0` for read accesses
- `/dev/xdma0_h2c_0` for write accesses

### 5.2.1 Reading from a register

Example in C:

```
uint32_t address = 0x00000000;
int fd = open("/dev/xdma0_c2h_0", O_RDWR);
lseek(fd, address, SEEK_SET);
uint64_t value;
read(fd, &value, sizeof(uint64_t));
```

### 5.2.2 Writing to a register

It is important to write the whole register width of 64 bits. If a register has less than 64 bits, the unused MSBs have to be written to any value. 32 bit write accesses will not have any effect.

Example in C:

```
uint64_t value = 42;
uint32_t address = 0x00000400;
int fd = open("/dev/xdma0_h2c_0", O_RDWR);
lseek(fd, address, SEEK_SET);
write(fd, &value, sizeof(uint64_t));
```

## 5.3 Register map

### 5.3.1 Status registers

The following status registers can be read by software:

| index | address | bits | radix | description |
|---|---|---|---|---|
| 0 | 0x00000000 | 1 | binary | gate status |
| 124 | 0x000003E0 | 32 | hex | build timestamp |
| 125 | 0x000003E8 | 57 | unsigned | FPGA serial number |
| 126 | 0x000003F0 | 64 | hex | module ID |
| 127 | 0x000003F8 | 64 | unsigned | magic number |

Table 5.1: List of status registers

#### 0: gate status

This register holds the status of the gate signal, which is also output on the DIO 3 LEMO connector.

#### 124: build timestamp

Time when the bitstream was created.

Format:

| bits 31 - 27 | bits 26 - 23 | bits 22 - 17 | bits 16 - 12 | bits 11 - 6 | bits 5 - 0 |
|---|---|---|---|---|---|
| day | month | year (last two decimal digits) | hours | minutes | seconds |

#### 125: FPGA serial number

The XDMA PCIe driver by Xilinx numbers the devices randomly and is not able to identify the slot number of an AMC board. This register holds the FPGA's unique serial number and can be used to identify an AMC board.

#### 126: module ID

The module ID can be used to identify the type of the current bitstream. The value of the module ID of the rate divider gateware is either 0x0002000200010003 or 0x0002010300010003, depending on the AFC board version.

The fields are defined as follows:

| bits 63 - 56 | bits 55 - 48 | bits 47 - 40 | bits 39 - 32 | bits 31 - 16 | bits 15 - 0 |
|---|---|---|---|---|---|
| minor gateware version | major gateware version | minor board version | major board version | developer ID | project ID |

Since the AFC board versions 2.0 and 3.1 are used in this project, the board version can be either 0x0002 or 0x0103.

Here is an incomplete list of project IDs:

| project ID | project name |
|---|---|
| 0x0001 | Cryring BPM |
| 0x0002 | UniMon |
| 0x0003 | Rate Divider |
| 0x0004 | BLoFELD |
| 0x0005 | Resonant Transformer |
| 0x8001 | Red Pitaya |

#### 127: magic number

The magic number can be used to determine if the gateware uses the expected register format.
The value of this register is the same for all module IDs: 0xBADEAFFEDEADC0DE.

### 5.3.2 Configuration registers

The following registers can be written by software:

| index | address | bits | radix | description | default value |
|---|---|---|---|---|---|
| 0 | 0x00000400 | 32 | unsigned | divisor minus 1 | 0x00000000 |
| 1 | 0x00000408 | 1 | binary | gate mode | 0x0 |
| 2 | 0x00000410 | 32 | unsigned | gate length | 0x00000000 |
| 3 | 0x00000418 | 32 | unsigned | gate start delay | 0x00000000 |
| 4 | 0x00000420 | 32 | unsigned | gate stop delay | 0x00000000 |
| 16 | 0x00000480 | 1 | binary | RF input select | 0x0 |
| 17 | 0x00000488 | 1 | binary | FTRN input select | 0x1 |
| 18 | 0x00000490 | 8 | hex | MLVDS direction | 0x00 |
| 19 | 0x00000498 | 1 | binary | test signal enable | 0x00 |
| 20 | 0x000004A0 | 32 | unsigned | test RF toggle period | 0x00000063 |
| 21 | 0x000004A8 | 32 | unsigned | test FTRN toggle period | 0x000003e7 |
| 127 | 0x000007F8 | 1 | binary | reset | 0x0 |

Table 5.2: List of configuration registers

#### 0: divisor minus 1

This parameter defines the division ratio of the incoming RF pulses.
At a value of 0, all incoming pulses inside the gate will be passed to the output.
At a value of e.g. 9, the {1st, 11th, 21th, ...} of the incoming pulses inside the gate will be passed to the output.

#### 1: gate mode

If the gate mode is 0, the gate length will be defined by the length of the incoming pulse from the FTRN.
In this case the value of register 2 *gate length* will be ignored.

If the gate mode is 1, the gate length will be defined by register 2 *gate length*.

#### 2: gate length

The value of this register is only relevant if register 1 *gate mode* is set to a value of 1.
The gate length is adjustable with the granularity of the FPGA clock frequency of 125 MHz. 1 LSB corresponds to 8 ns.

#### 3: gate start delay

Additional delay between the rising edge of the FTRN input signal and the rising edge of the gate signal. 1 LSB corresponds to a delay of 8 ns.

#### 4: gate stop delay

Additional delay to the falling edge of the gate signal. 1 LSB corresponds to a delay of 8 ns.
The value of this register will be ignored if register 1 *gate mode* is set to a value of 1.

#### 16: RF input select

This register is intended for testing purposes only and can be left at its default value for normal operation.

| value | RF input source |
|---|---|
| 0 | FMC DIO LEMO connector 1 |
| 1 | MicroTCA backplane, MLVDS line 1 |

### 17: FTRN input select

The default connection of the FTRN signal is via the backplane.

| value | RF input source |
|---|---|
| 0 | FMC DIO LEMO connector 2 |
| 1 | MicroTCA backplane, MLVDS line 0 |

### 18: MLVDS direction

This register is intended for testing purposes only. The direction of each of the eight MLVDS pins is individually adjustable:

| value | direction |
|---|---|
| 0 | input |
| 1 | output |

### 19: test signal enable

This register is intended for testing purposes only. It enables the generation of test signals which allow standalone tests of the rate divider. If enabled, a test RF signal will be output on DIO 4 and a test FTRN signal will be output on DIO 5. Additionally, the two signals can be output on the MLVDS lines 1 and 0, if the corresponding pins are configured as outputs via register 18 *MLVDS direction*.

### 20: test RF toggle period

This register is intended for testing purposes only. It contains the period after which the test RF signal will toggle. 1 LSB corresponds to 8 ns. The test signal is a continuous rectangle signal.

### 21: test FTRN toggle period

This register is intended for testing purposes only. It contains the period after which the test FTRN signal will toggle. 1 LSB corresponds to 8 ns. The test signal is a continuous rectangle signal.

### 127: reset

Resets the FPGA, which includes resetting the configuration registers to their default values. This register will also automatically be resetted to zero.

# 6 List of Devices

| location | FEC | MCH | AMC slot nr | AFC version | AFC serial nr | FPGA serial nr | commen |
|---|---|---|---|---|---|---|---|
| ELR | sddsc079 | sdmch019 | 2 | 2.0 | - | 0x00408c8522c3004 | no SDRA |
| ELR | sddsc080 | sdmch018 | 2 | 2.0 | - | 0x06408c8522c300c | - |
| TBD | TBD | TBD | TBD | 3.1 | 35233 | 0x054b48160e47054 | no SDRA |

Table 6.1: List of Rate Divider devices in use

The Rate Divider gateware does not use the SDRAM on the AFC board.
The FPGA serial number is not printed anywhere but can only be read from the status register *FPGA serial number*.