IRQ Generator

Design Specification

June 30th, 2017

Table of Contents

1. Introduction

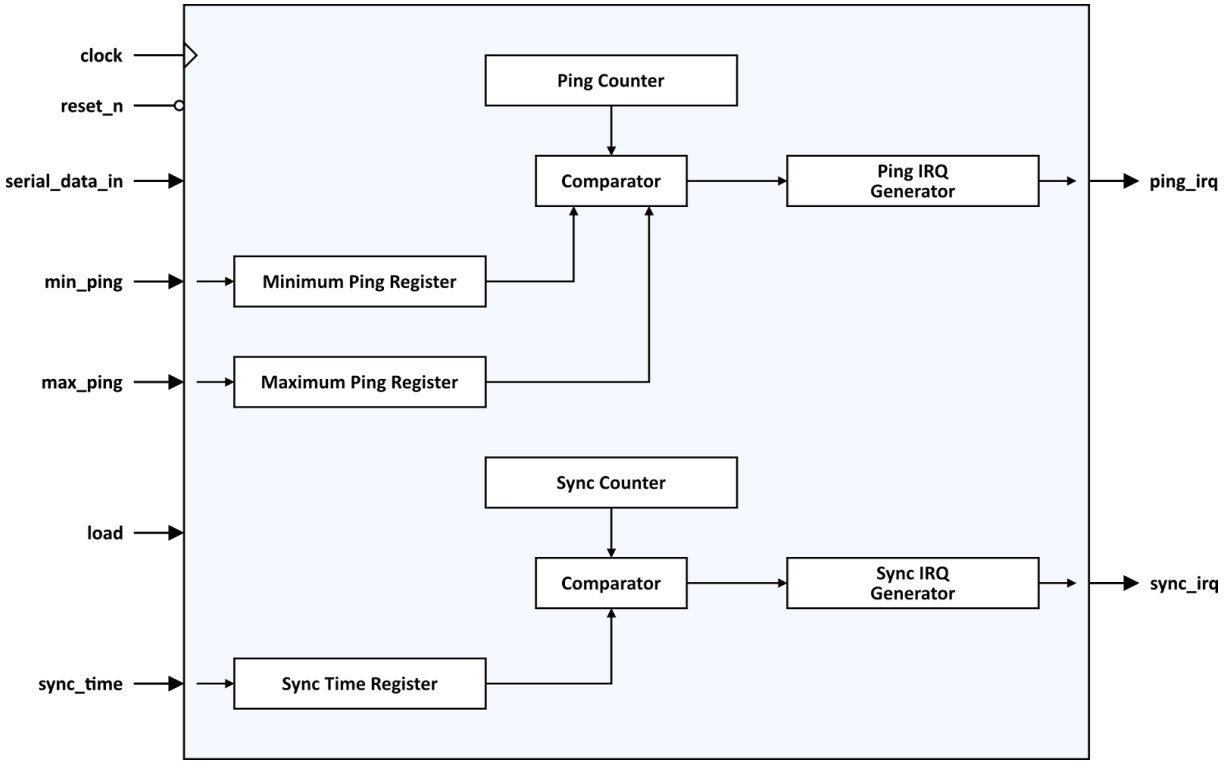
The IRQ generator is a module that generates two types of interrupt requests (IRQs):

- **Ping IRQs, which are pulses generated at the falling edge of `serial_data_in`, if the pulse's width is within a configurable range;**
- **Sync IRQs, which are pulses generated periodically, after a rising edge of `serial_data_in`, as long as `serial_data_in` stays high for at least a minimum configurable number of clock cycles.**

2. Block diagram

Below is a simplified schematic of the block diagram, presenting the main functional blocks inside the module.

Figure 2.1 - IRQ generator block diagram



3. Operation

3.1. Loading the configuration

After a reset, all the outputs and the internal registers will be equal to 0 and the module will start in idle mode. It remains so until the first configuration is loaded. During this period, no IRQs are generated.

To load a configuration, the `load` signal must be asserted and the values of `min_ping`, `max_ping` and `sync_time` will be stored in the internal registers and further used to decide when to generate the IRQs. `min_ping` and `max_ping` are used to generate ping IRQs, while `sync_time` is used to generate sync IRQs. If the `load` signal is not asserted, changing the values at the input will not affect the operation of the module.

3.2. IRQ types

There are two types of IRQs: ping IRQs and sync IRQs.

3.2.1. Ping IRQs

The ping IRQ is a one clock cycle pulse that is generated on the falling edges of pulses on the serial data input. The ping IRQ is generated only if the width of the input pulse, expressed in clock cycles, has a value between the minimum and the maximum ping times which were loaded from `min_ping` and `max_ping` inputs. No ping IRQ is generated if the minimum ping time is larger than the maximum ping time. Setting both values to zero will result in disabling the ping IRQs.

3.2.2. Sync IRQs

Sync IRQs are one clock cycle pulses which are generated periodically on the `sync_irq` output. If the serial data input is asserted for a number of clock cycles greater than or equal to `sync_time_reg + 1`, a one clock cycle pulse is generated every `sync_time_reg + 1` cycles while the serial data input remains asserted. `sync_time_reg` is the last value loaded from the `sync_time` input. If the value in `sync_time_reg` is equal to 0, the sync IRQ will be asserted continuously, until the serial data input falls to logic 0.

4. Waveforms

In the waveforms below, the signals marked with "▸" are the module's inputs, while the signals marked with "◂" are the module's outputs. `ping_counter` and `sync_counter` are the internal counters used in the IRQ generation logic.

Figure 4.1 - Ping IRQs timing diagram

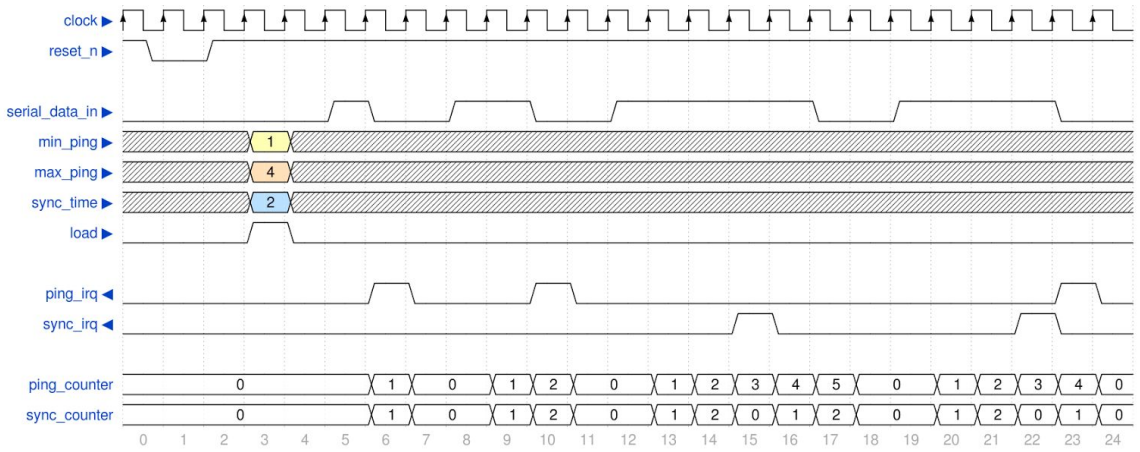
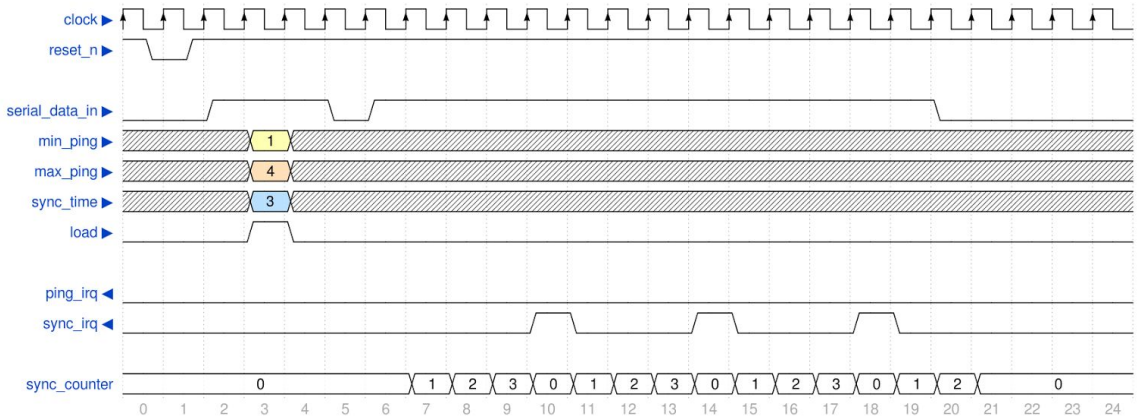


Figure 4.2 - Sync IRQs timing diagram



5. Port List

Table 5.1 - Port list

Name	Width	Direction	Description
clock	1	Input	System clock.
reset_n	1	Input	Active low asynchronous reset.
serial_data_in	1	Input	Serial data input.
min_ping	16	Input	Minimum ping time.
max_ping	16	Input	Maximum ping time.
sync_time	16	Input	Sync time.
load	1	Input	When asserted, the values of `min_ping`, `max_ping` and `sync_time` inputs are stored in the internal registers.
ping_irq	1	Output	Ping interrupt request.
sync_irq	1	Output	Sync interrupt request.