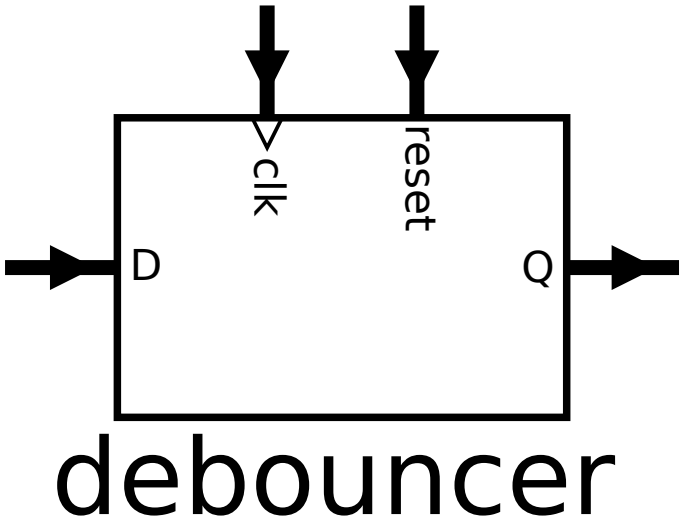
debouncer

The **debouncer** filters short transient pulses from a signal, ensuring that there are no accidental transitions that may result in unwanted behavior.



Port descriptions:

- Input port `clk` a clock with a 10ns period
- Input port `reset` an asynchronous reset port.
- Input port `D` is the debouncer input.
- Output port `Q` is the debouncer output.

The internal signals are:

- Multi-bit signal `counter` counts down to the time where the input has been stable for long enough to merit a transit a change in the output. The number of bits in this counter can be customized through the use of a generic field.
- Signal `i_D` buffers the input port `D`.
- Signal `i_Q` buffers the output port `Q`.

Behavioral description:

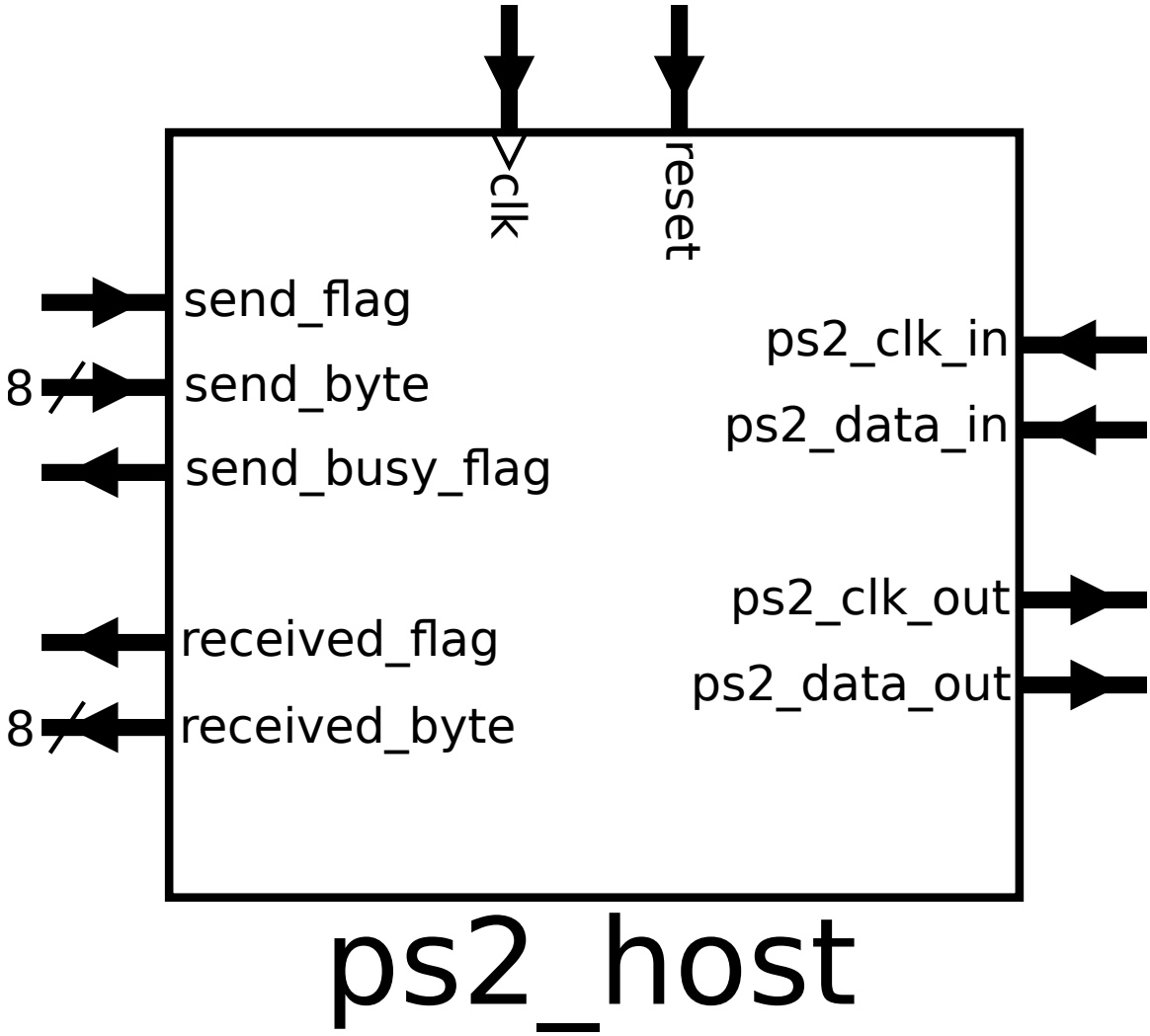
When `reset` is high, the input `D` writes straight through to the output.

The value of `i_D` lags that of `D` by 1 clock cycle, so `D` and `i_D` can be used to detect changes in the input `D`.

When a change has been detected, `counter` is reset to its maximum value. This maximum value can be customized through the use of a generic field.

When a change is not detected, if the counter is 0, then the input writes through to the output, otherwise the counter decreases by 1.

ps2_host



Port descriptions:

- Input port `clk` a clock with a 10ns period
- Input port `reset` an asynchronous reset port.
- Input port `ps2_clk_in` is the input ps2 clock connection. **This port connects to a remote device.**
- Input port `ps2_data_in` is the input ps2 data connection. **This port connects to a remote device.**
- Output port `ps2_clk_out` is the output ps2 clock connection. **This port connects to a remote device.**
- Output port `ps2_data_out` is the output ps2 data connection. **This port connects to a remote device.**

- Output 8-bit port `received_byte` returns bytes that the host receives from the ps2 connection.
- Output port `received_flag` is a flag that is normally `0`, but emits a **1 clock period** pulse on the clock cycle where `received_byte` is returning the byte received from the ps2 connection.
- Input 8-bit port `send_byte` is the byte that is to be sent over the ps2 connection.
- Input port `send_flag` is given a **1 clock period** pulse on the clock cycle where `send_byte` is set to the byte that is to be sent over the ps2 connection.
- Output port `send_busy_flag` is `1` when an existing byte is queued to be sent or is being sent, and no new bytes are currently being accepted.

The internal signals are:

- Signal `i_ps2_clk` is the input `ps2_clk_in` signal filtered through a debouncer to eliminate transient pulses less than 1μs in duration.
- Signal `i_ps2_data` is the input `ps2_data_in` signal filtered through a debouncer to eliminate transient pulses less than 1μs in duration.
- Signal `i_ps2_clk_d1` is `i_ps2_clk` delayed by 1 clock cycle to detect falling edges in `i_ps2_clk`.
- Signal `i_ps2_clk_falling_edge` is high when the current value of `i_ps2_clk` is 0, while the value on the previous clock cycle is 1.
- Signal `i_direction` has 3 states: `S_DIR_IDLE`, `S_DIR_RECEIVE`, `S_DIR_SEND`. These states denote the following:
  - `S_DIR_IDLE` denotes the state where no data is begin received or transmitted.
  - `S_DIR_RECEIVE` denotes the state where a byte is being received.
  - `S_DIR_SEND` denotes the state where a byte is being sent. Note that a single byte can be primed for sending even if the state is currently `S_DIR_RECEIVE`. When the state returns to `S_DIR_IDLE`, it will then be set to `S_DIR_SEND` and transmission will begin immediately.
- Signal `i_data_received_state` has 6 states: `S_REC_IDLE`, `S_REC_BITS`, `S_REC_PARITY`, `S_REC_STOP`, `S_REC_DONE`, `S_REC_ERROR`. These states denote the following:
  - `S_REC_IDLE` denotes the state where no byte is being received. This is the state when `i_direction` is not `S_DIR_RECEIVE`. When `i_direction` is `S_DIR_IDLE`, and a falling ps2 clock edge has been detected, this signals a transition of `i_data_received_state` to either `S_REC_BITS` or `S_REC_ERROR` depending on the ps2 data line (a valid start bit should always be 0).
  - `S_REC_BITS` denotes the state where data bits are expected to be sent over the ps2 data line. A new bit is read with each falling ps2 clock edge, and is registered. When a total of 8 bits have been received in this state, sate `S_REC_PARITY` begins.
  - `S_REC_PARITY` denotes the state where the parity bit is now expected. When this bit arrives, the next state is either `S_REC_STOP` or `S_REC_ERROR` depending on the status of the parity bit.
  - `S_REC_STOP` denotes the state where the stop bit is now expected. When this bit arrives, the next state is either `S_REC_DONE` or `S_REC_ERROR` depending on whether of not the stop bit is correctly 1 or not.
  - `S_REC_DONE` denotes a successful receipt of a byte from the remote device. **This state lasts for exactly 1 clock cycle**, after which the next state is `S_REC_IDLE`.
  - `S_REC_ERROR` denotes a failed receipt of a byte from the remote device. **This state lasts for exactly 1 clock cycle.**, after which the next state is `S_REC_IDLE`.

- 8-bit signal `i_received_data_buffer` stores the bits received from the ps2 data line.
- 3-bit signal `i_received_data_bit_index` stores an index from 0 to 7, the index of the bit that is being expected.
- Signal `i_received_data_parity_bit` stores the current parity as the bits are received.
- 17-bit signal `i_receive_time_out_counter` is a counter that counts the clock ticks since the last falling ps2 clock edge. If 1ms passes without a falling ps2 clock edge, it will be assumed that there was an error in transmission, and the receipt is aborted, turning `i_data_received_state` to `S_REC_ERROR`, eventually returning `i_direction` to `S_DIR_IDLE`.
- Signal `i_send_busy_flag` is an internal register for the `send_busy_flag` output port.
- Signal `i_data_send_state` has 9 states: `S_SEND_IDLE`, `S_SEND_INHIBIT`, `S_SEND_BITS`, `S_SEND_PARITY`, `S_SEND_STOP`, `S_SEND_PS2_RELEASE`, `S_SEND_ACKNOWLEDGE`, `S_SEND_DONE`, `S_SEND_ERROR`. These states denote the following:
  - `S_SEND_IDLE` denotes the state where no byte is being currently sent. This is the state when `i_direction` is not `S_DIR_SEND`. When `i_direction` becomes `S_DIR_SEND`, the state transitions to `S_SEND_INHIBIT`, the ps2 clock is forced low, and a countdown counter (`i_send_inhibitor_counter`) for the inhibit pulse is set to 150μs.
  - `S_SEND_INHIBIT` denotes the state where the ps2 clock is being inhibited by being forced to 0. This state lasts until the inhibitor counter, `i_send_inhibitor_counter`, reaches 0 wherein the ps2 clock is released (set to 1) and the ps2 data port is taken control of and forced low. The next state is `S_SEND_BITS`.
  - `S_SEND_BITS` denotes the state where a bit is assigned to the ps2 data port on each falling ps2 clock edge. After the 8 bits comes `S_SEND_PARITY`.
  - `S_SEND_PARITY` denotes the state where the parity bit is sent on the falling ps2 clock edge. Next is `S_SEND_STOP`.
  - `S_SEND_STOP` denotes the state where the stop bit, namely 1, is sent on the falling ps2 clock edge. Next is `S_SEND_ACKNOWLEDGE`.
  - `S_SEND_ACKNOWLEDGE` denotes the state where the ps2 data port is to receive a 0 on the last falling ps2 clock edge. If this 0 is received, the next state is `S_SEND_DONE`, otherwise the next state is `S_SEND_ERROR`.
  - `S_SEND_DONE` denotes a successful transmission of a byte to the remote device. **This state lasts for exactly 1 clock cycle**, after which the next state is `S_SEND_IDLE`.
  - `S_SEND_ERROR` denotes a failed transmission of a byte to the remote device. **This state lasts for exactly 1 clock cycle.**, after which the next state is `S_SEND_IDLE`. The transmission is immediately re-attempted.
- 8-bit signal `i_send_data_buffer` stores the bits that are to be sent to the ps2 data line.
- 3-bit signal `i_send_data_bit_index` stores an index from 0 to 7, the index of the bit that is being sent.
- Signal `i_send_data_parity_bit` stores the current parity as the bits are sent.
- 15-bit signal `i_send_inhibitor_counter` is a counter that counts down to the end of the inhibitor pulse.