*IFRO DESIGN & VERIFICATION SUMMER SCHOOL*

# *General-Purpose Counter*

## *Design Activities*

A general-purpose counter can be used in different modes and configurations such as:

- Count different events inside a system,
- Count starting with a configurable event,
- Measure the distance between two events,
- Generate signals with a defined shape.

The proposed architecture for the general-purpose counter is described in Figure1.

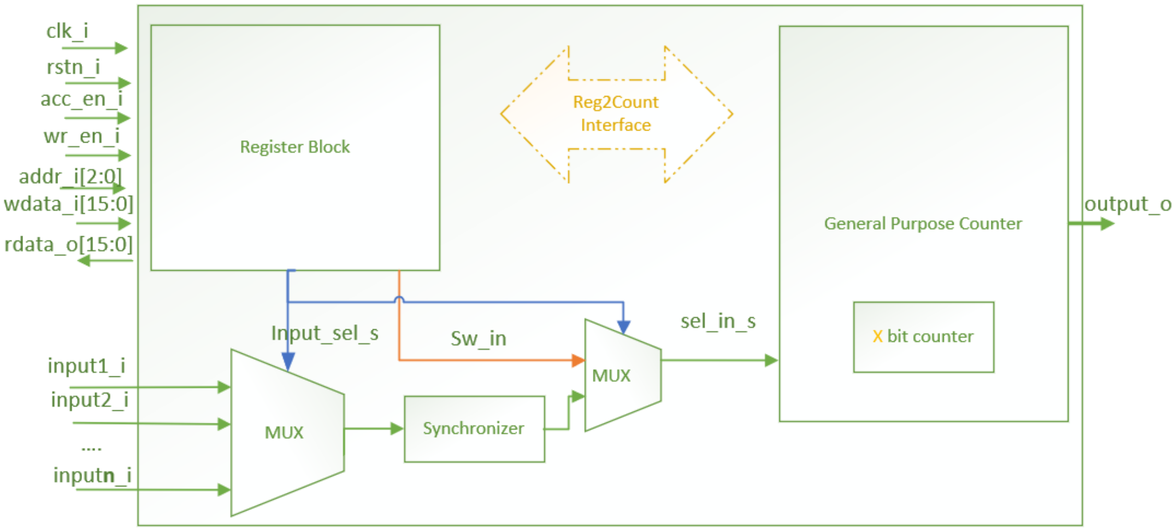


Figure1. General-purpose Counter Architecture

The top module should contain:

- A synchronous interface used to configure the counter,
- A register block which contains all the configuration bits,
- Synchronizer for the asynchronous inputs (inputx_i, x=1,n),
- One *single counter* and the logic needed to implement all the different operational modes.

The interface of the general-purpose counter is described in the table below.

| Port name | Size | Direction | Clock domain | Description |
|---|---|---|---|---|
| clk_i | 1 | input | n.a | Clock signal with target frequency @40Mhz |
| rstn_i | 1 | input | asynchronous | Asynchronous reset, active low |
| acc_en_i | 1 | input | clk_i | Access enable for synchronous transactions |
| wr_en_i | 1 | input | clk_i | Write enable. When set, a write is performed, when reset, a read is performed |
| addr_i | 3 | input | clk_i | Address accessed during a transaction |
| wdata_i | 16 | input | clk_i | Data to be written during write access |
| rdata_o | 16 | output | clk_i | Data read during read access |
| inputx_i* | 1 | input | asynchronous | Inputs that can be selected as an event source for the general-purpose counter |
| output_o | 1 | output | clk_i | Output calculated based on operation mode and out function |

*x = 1, n. n should be calculated

## Communication Interface

A protocol is a set of rules that define how data is transmitted and received over a network or communication channel. Protocols define the format, timing, sequencing, and error control for data communication. They also specify the types of messages that can be exchanged between devices, and how those messages are interpreted. Depending on the application for each they were defined, protocols are very different and can be grouped in different categories.

In this application, we will use a protocol with the following attributes:

- Synchronous
- Parallel

Synchronous communication requires that the clocks in the transmitting and receiving devices are synchronized – running at the same rate – so the receiver can sample the signal at the same time intervals used by the transmitter. Therefore, synchronous protocols always have an extra wire for the external clock signal.
In parallel communication, multiple bits of data are sent simultaneously, as oppose to serial communications which involves sending data one bit at a time.
The timing for write and read commands is shown in Figure2 and Figure3.

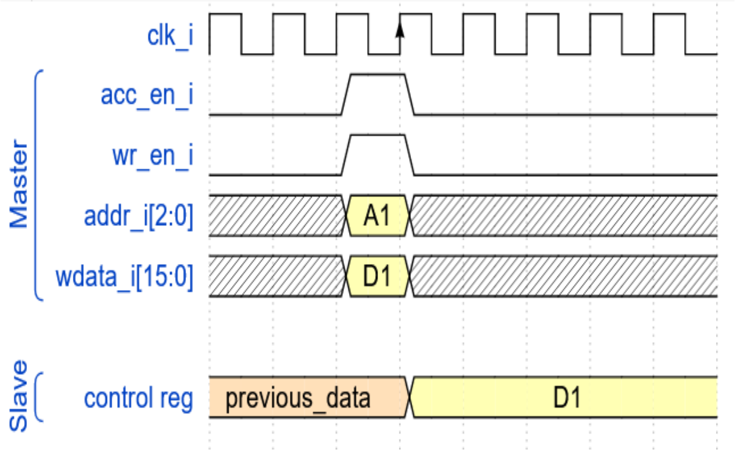


Figure2. Write acces

Each access takes one clock cycle. Writting to a read-only register has no effect. Reading from a write-only register will return a zero. When the module is not accesed (acc_en_i = 1'b0), it should return zero.

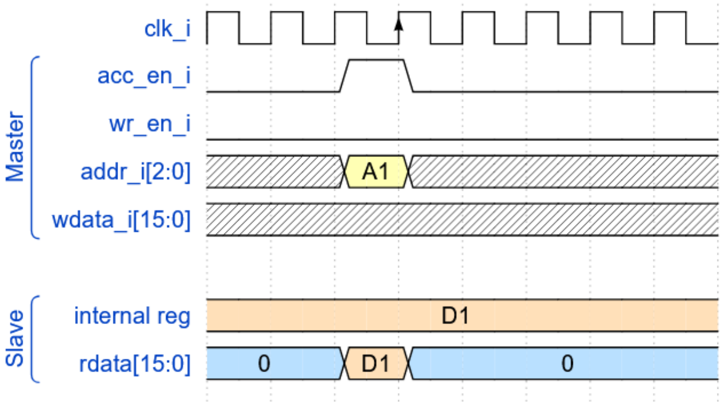
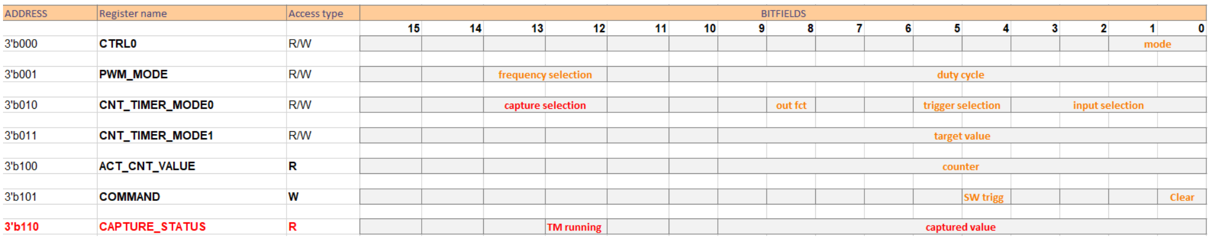


Figure3. Read acces

# *Configurability – register block.*

The register block will contain the configuration bits of the counter and will facilitate the access to the status of it. The module should implement the register map described in Figure4.



| ADDRESS | Register name | Access type | BITFIELDS | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3'b000 | CTRL0 | R/W | | | | | | | | | | | | | | | mode | |
| 3'b001 | PWM_MODE | R/W | | | frequency selection | | | | duty cycle | | | | | | | | | |
| 3'b010 | CNT_TIMER_MODE0 | R/W | | | capture selection | | | | | out fct | | | trigger selection | | input selection | | | |
| 3'b011 | CNT_TIMER_MODE1 | R/W | | | | | | | target value | | | | | | | | | |
| 3'b100 | ACT_CNT_VALUE | R | | | | | | | counter | | | | | | | | | |
| 3'b101 | COMMAND | W | | | | | | | | | | | | SW trigg | | | | Clear |
| 3'b110 | CAPTURE_STATUS | R | | | | TM running | | | captured value | | | | | | | | | |

Figure4. Register map

Each register is completely described in the following section. In PWM mode, only the PWM_MODE register is used to configure the general-purpose counter. For counter and timer mode the configuration registers are common (CNT_TIMER_MODE0/1). The default(reset) value is underlined in each register.

| **CTRL0** | | | |
|---|---|---|---|
| Bit # | Bit field name | Access | Description |
| [1:0] | Mode | R/W | These bits are used to enable the counter and set the operation mode:<br><u>2'b00 – module disabled</u><br>2'b01 – module in PWM mode<br>2'b10 – module in counter mode<br>2'b11 – module in timer mode |
| [15:2] | n.a | n.a | Not used |

| **PWM MODE** | | | |
|---|---|---|---|
| Bit # | Bit field name | Access | Description |
| [9:0] | Duty cycle | R/W | These bits define the duty cycle of the PWM signal as follows:<br><u>10'd0 – 0% duty cycle</u><br>10'd1023 – 100% duty cycle<br>The general formula of the duty cycle, based on the value of this register is:<br>Dec[xx xxxx xxxx]/1023)*100% duty cycle |
| [11:10] | n.a | n.a | Not used |
| [13:12] | Frequency selection | R/W | These bits define the frequency of the PWM signal as follows:<br><u>2'00 – 100 Hz</u><br>2'01 – 200 Hz<br>2'10 – 320 Hz<br>2'11 – 400 Hz |
| [15:14] | n.a | n.a | Not used |

| **Counter MODE0** | | | |
|---|---|---|---|
| Bit # | Bit field name | Access | Description |
| [3:0] | Input selection | R/W | These bits are used to select one of the 16 inputs as follows:<br>4<u>'b0000 – software input</u> |

| | | | |
|---|---|---|---|
| | | | 4’b0001 – input1<br>…<br>4’b1111 – input15 |
| [5:4] | Trigger selection | R/W | These bits are used to select the trigger option for the counter/time mode as follows:<br>2’b00 – rising edge used as trigger<br>2’b01 – falling edge used as trigger<br>2’b10 – both edges used as trigger<br>2’b11 – none (not used) |
| [7:6] | n.a | n.a | Not used |
| 8 | Out function | R/W | This bit defines the format of the output signal as follows:<br>1’b0 – output signal is active only one clock cycle (pulse generator)<br>1’b1 – output signal toggles when the according condition is met |
| [11:9] | n.a | n.a | Not used |
| [13:12] | Capture selection | R/W | These bits are used to select the capture event as follows:<br>2’b00 – rising edge used as trigger<br>2’b01 – falling edge used as trigger<br>2’b10 – both edges used as trigger<br>2’b11 – none (not used) |
| [15:14] | n.a | n.a | Not used |

**Counter MODE1**

| Bit # | Bit field name | Access | Description |
|---|---|---|---|
| [9:0] | Target value | R/W | These bits are used to set the target value for the counter. When this value is reached, the output will be activated according to the selected output function. The reset value for this is 0. Once the target value is reached, the internal counter will be reset automatically. |
| [15:10] | n.a | n.a | Not used |

**Actual Counter Value**

| Bit # | Bit field name | Access | Description |
|---|---|---|---|
| [9:0] | Counter | R | These bits reflect the current value of the internal counter. |
| [15:10] | n.a | n.a | Not used |

**Command**

| Bit # | Bit field name | Access | Description |
|---|---|---|---|
| 0 | Clear | W | This bit will clear the internal counter |
| [3:1] | n.a | n.a | Not used |
| 4 | SW trigger | W | This bit is used as software input and will be used by the internal counter if the selection input is set to 0. |
| [15:5] | n.a | n.a | Not used |

**Captured value**

| Bit # | Bit field name | Access | Description |
|---|---|---|---|
| [9:0] | Captured value | R | These bits are used to store the captured value (the value of the internal counter when the capture condition appears) |
| [11:10] | n.a | n.a | Not used |
| 12 | TM running | R | This bit is set when the counter/timer start counting (when start condition is detected). It will be cleared when the target value will be reached or when the clear command is used. |
| [15:13] | n.a | n.a | Not used |

# Operational modes

## PWM Mode

Pulse-width modulation (PWM) is any method of representing a signal as a rectangular wave with a varying duty cycle. PWM is useful for controlling the average power or amplitude delivered by an electrical signal. The average value of voltage (and current) fed to the load is controlled by switching the supply between 0 and 100% at a rate faster than it takes the load to change significantly. Different duty cycles are shown in Figure5.

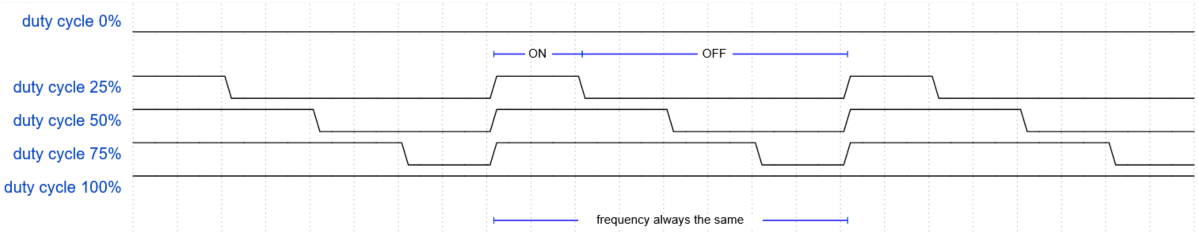


Figure5. Duty cycle

## Counter Mode

Counter mode will be used to count events on a selectable source. The current value of the counter can be read through a status register. When the maximum value is reached(configurable), the output will be activated according to the *out-function* bit.

Counter mode with target value =10'd3, output function set to 0 and trigger selection set to 0(rising edge) is shown in Figure6.

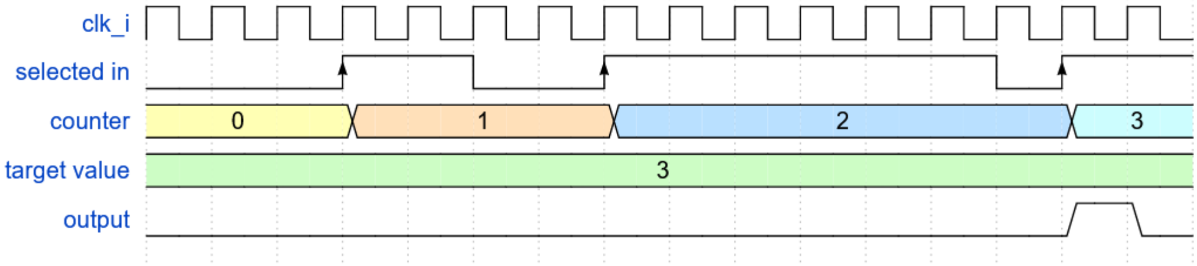


Figure6. Example in counter mode

Hint: Figure6,7,8 must be used only to understand the functionality of the general-purpose counter in different modes. The real timing (e.g. delay between rising edge on the selected input and counter increment, between counter saturation and output toggling) must be calculated according to the implementation and the synchronous design rules.

## Timer Mode

Timer mode will be used to count starting from a selectable event on a selectable source. From this start condition, the counting will be done using the internal clock. The current value of the counter can be read through a status register. When the maximum value is reached(configurable), the output will be activated according to the *out-function* bit. If another start condition is detected before the maximum value is reached, the counting will re-start.

Counter mode with target value =10’d3, output function set to 1(toggle function) and trigger selection set to 0(rising edge) is shown in Figure7.

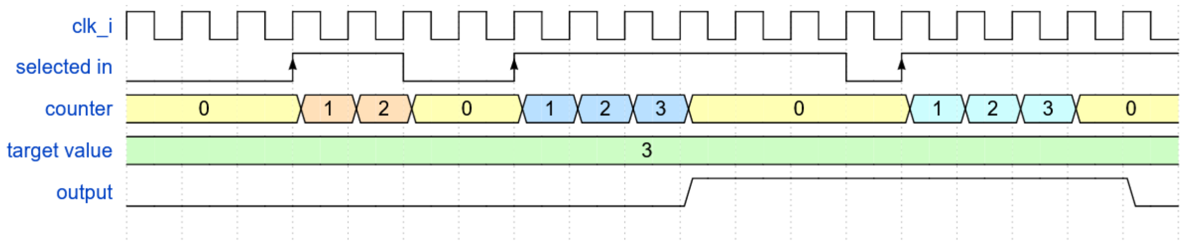


Figure7. Example in timer mode

## Extra option – Capture Mode

Capture mode is just an extra option for the already existing counter and timer modes. With this function an extra event, capture event, is defined and we will store the value of the running counter when this event is triggered. Output function is not affected by this feature. An extra status is also provided, while the counter is running (greater than zero).

An example for capture mode with trigger selection set as 0 (rising edge) and capture selection set to 1 (falling edge) is shown in Figure8.

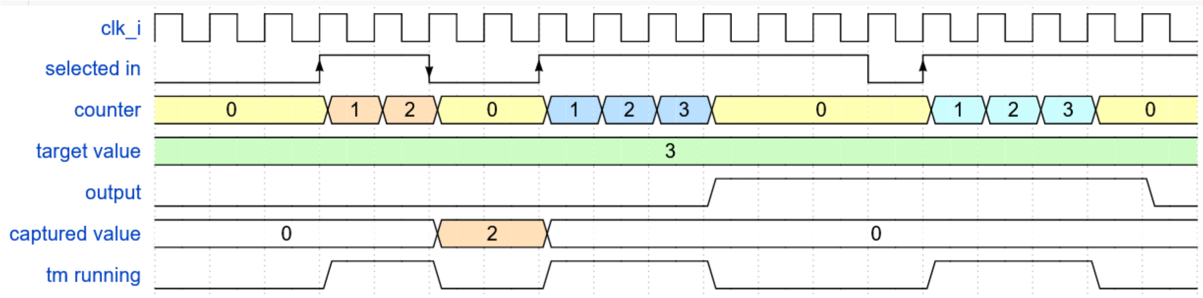


Figure8. Example in capture mode

# Task scheduler

## Day 1:

- Digital Design Introduction, HW vs SW, programming languages vs hardware description languages (mentors)
- Synchronous parallel protocol explanations & discussions (all)
- Interface + register block implementation (students)

## Day 2:

- Testbench for existing blocks (students)
- PWM mode explanations & discussions (all)
    1. Size of the counter (x from architecture)
    2. Tics or clock used for counting.
- Counter for PWM mode implementation (students)

## Day 3:

- Testbench for existing blocks(students)
- Counter mode & timer mode explanations & discussions (all)
    1. Synchronization
    2. Edge detector
- Counter mode implementation (students)
- Timer mode implementation(students)

## Day 4:

- Top level instantiations & tests for timer/counter mode(students)
- Digital Design Backend, RTL -> chip (mentors)