

EF_WDT32

A simple 32-bit WatchDog Timer (WDT). It can be used to interrupt the CPU to:

Indicate a software malfunction. Wakeup the CPU after a certain sleep time. The WDT is a simple 32-bit down-counter which is decremented every system clock cycle. When it reached zero, the counter is loaded with the value stored in the load register and the TO flag is raised..

The WDT can be enabled/disabled. When it is disabled the counter is loaded with the value in the load register.

The wrapped IP

APB, AHBL, and Wishbone wrappers are provided. All wrappers provide the same programmer's interface as outlined in the following sections.

Wrapped IP System Integration

Based on your use case, use one of the provided wrappers or create a wrapper for your system bus type. For an example of how to integrate the wishbone wrapper:

```
EF_WDT32_WB INST (  
    .clk_i(clk_i),  
    .rst_i(rst_i),  
    .adr_i(adr_i),  
    .dat_i(dat_i),  
    .dat_o(dat_o),  
    .sel_i(sel_i),  
    .cyc_i(cyc_i),  
    .stb_i(stb_i),  
    .ack_o(ack_o),  
    .we_i(we_i),  
    .IRQ(irq),  
);
```

Wrappers with DFT support

Wrappers in the directory `/hdl/rtl/bus_wrappers/DFT` have an extra input port `sc_testmode` to disable the clock gate whenever the scan chain testmode is enabled.

Interrupt Request Line (irq)

This IP generates interrupts on specific events, which are described in the [Interrupt Flags](#) section bellow. The IRQ port should be connected to the system interrupt controller.

Implementation example

The following table is the result for implementing the EF_WDT32 IP with different wrappers using Sky130 HD library and [OpenLane2](#) flow.

Module	Number of cells	Max. freq
EF_WDT32	TBD	TBD
EF_WDT32_APB	TBD	TBD
EF_WDT32_AHBL	TBD	TBD

Module	Number of cells	Max. freq
EF_WDT32_WB	TBD	TBD

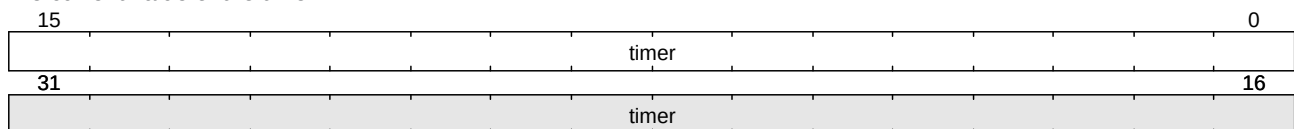
The Programmer's Interface

Registers

Name	Offset	Reset Value	Access Mode	Description
timer	0000	0x00000000	r	The current value of the timer
load	0004	0x00000000	w	The value loaded into the timer when it reaches zero
control	0008	0x00000000	w	Writing 0 disables the WDT and writing 1 enables it
IM	ff00	0x00000000	w	Interrupt Mask Register; write 1/0 to enable/disable interrupts; check the interrupt flags table for more details
RIS	ff08	0x00000000	w	Raw Interrupt Status; reflects the current interrupts status; check the interrupt flags table for more details
MIS	ff04	0x00000000	w	Masked Interrupt Status; On a read, this register gives the current masked status value of the corresponding interrupt. A write has no effect; check the interrupt flags table for more details
IC	ff0c	0x00000000	w	Interrupt Clear Register; On a write of 1, the corresponding interrupt (both raw interrupt and masked interrupt, if enabled) is cleared; check the interrupt flags table for more details
GCLK	ff10	0x00000000	w	Gated clock enable; 1: enable clock, 0: disable clock

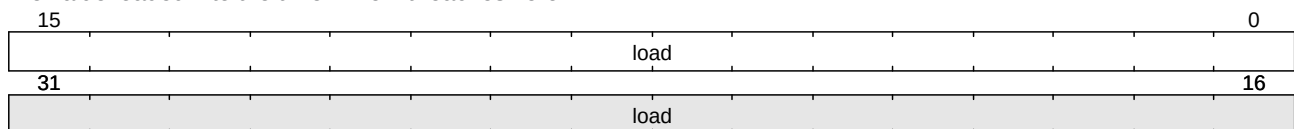
timer Register [Offset: 0x0, mode: r]

The current value of the timer



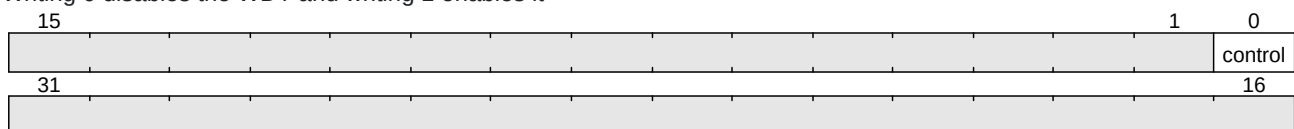
load Register [Offset: 0x4, mode: w]

The value loaded into the timer when it reaches zero



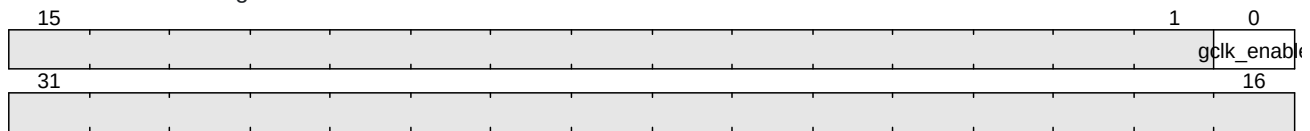
control Register [Offset: 0x8, mode: w]

Writing 0 disables the WDT and writing 1 enables it



GCLK Register [Offset: 0xff10, mode: w]

Gated clock enable register



bit	field name	width	description
0	gclk_enable	1	Gated clock enable; 1: enable clock, 0: disable clock

Interrupt Flags

The wrapped IP provides four registers to deal with interrupts: IM, RIS, MIS and IC. These registers exist for all wrapper types.

Each register has a group of bits for the interrupt sources/flags.

- IM [offset: 0xff00]: is used to enable/disable interrupt sources.
- RIS [offset: 0xff08]: has the current interrupt status (interrupt flags) whether they are enabled or disabled.
- MIS [offset: 0xff04]: is the result of masking (ANDing) RIS by IM.
- IC [offset: 0xff0c]: is used to clear an interrupt flag.

The following are the bit definitions for the interrupt registers:

Bit	Flag	Width	Description
0	WDTTO	1	Time out flag

Clock Gating

The IP includes a clock gating feature that allows selective activation and deactivation of the clock using the `GCLK` register. This capability is implemented through the `ef_util_gating_cell` module, which is part of the common modules library, [ef_util_lib.v](#). By default, the clock gating is disabled. To enable behavioral implementation clock gating, only for simulation purposes, you should define the `CLKG_GENERIC` macro. Alternatively, define the `CLKG_SKY130_HD` macro if you wish to use the SKY130 HD library clock gating cell, `sky130_fd_sc_hd__dclkp_4`.

Note: If you choose the [OpenLane2](#) flow for implementation and would like to enable the clock gating feature, you need to add `CLKG_SKY130_HD` macro to the `VERILOG_DEFINES` configuration variable. Update OpenLane2 YAML configuration file as follows:

```
VERILOG_DEFINES:  
- CLKG_SKY130_HD
```

Firmware Drivers:

Firmware drivers for EF_WDT32 can be found in the [Drivers](#) directory in the [EFIS](#) (Efabless Firmware Interface Standard) repo. EF_WDT32 driver documentation is available [here](#). You can also find an example C application using the EF_WDT32 drivers [here](#).

Installation:

You can install the IP either by cloning this repository or by using [IPM](#).

1. Using IPM:

- [Optional] If you do not have IPM installed, follow the installation guide [here](#)
- After installing IPM, execute the following command `ipm install EF_WDT32` .

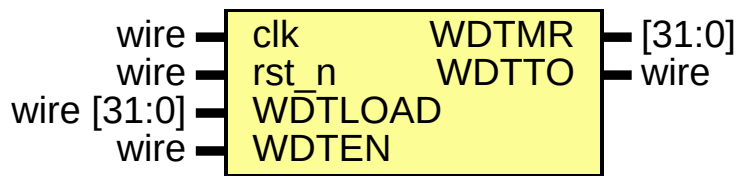
Note: This method is recommended as it automatically installs [EF_IP_UTIL](#) as a dependency.

2. Cloning this repo:

- Clone [EF_IP_UTIL](#) repository, which includes the required modules from the common modules library, [ef_util_lib.v](#).
`git clone https://github.com/efabless/EF_IP_UTIL.git`
- Clone the IP repository `git clone https://github.com/efabless/EF_WDT32`

The Wrapped IP Interface

NOTE: This section is intended for advanced users who wish to gain more information about the interface of the wrapped IP, in case they want to create their own wrappers.



Ports

Port	Direction	Width	Description
WDTMR	output	32	The current value of the timer
WDTLOAD	input	32	The value loaded into the timer when it reaches zero
WDTTO	output	1	Time out flag
WDTEN	input	1	watchdog timer enable