

EF_I2S

Two-wire I2S synchronous serial interface, compatible with I2S specification.

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_I2S_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),  
    .ws(ws),  
    .sck(sck),  
    .sdi(sdi)  
);
```

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.

External IO interfaces

IO name	Direction	Width	Description
ws	output	1	Word Select; this signal indicates the boundary between left and right audio channels
sck	output	1	Serial Clock; this provides the timing for data transfer, synchronizing the transmission of bits.
sdi	input	1	Serial Data Input; this line carries the audio data being sent to the device.

Interrupt Request Line (irq)

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

Implementation example

The following table is the result for implementing the EF_I2S IP with different wrappers using Sky130 HD library and [OpenLane2](#) flow.

Module	Number of cells	Max. freq
EF_I2S	2433	116
EF_I2S_APB	2716	142
EF_I2S_AHBL	2799	135
EF_I2S_WB	2941	136

The Programmer's Interface

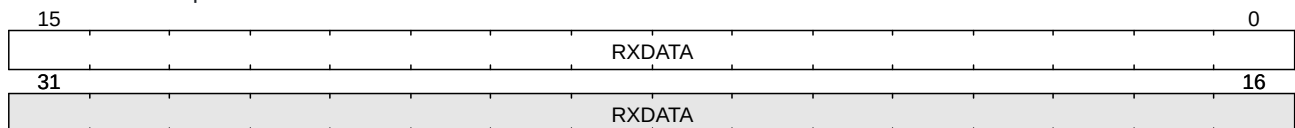
Registers

Name	Offset	Reset Value	Access Mode	Description
RXDATA	0000	0x00000000	r	The received sample
PR	0004	0x00000000	w	The Prescaler register; used to determine the sck signal frequency . $Prescaler = clkfreq / (2xsckfreq) - 1$.
AVGT	0008	0x00000000	w	The Average threshold
ZCRT	000c	0x00000000	w	The ZCR threshold
CTRL	0010	0x00000000	w	Control Register; 0:Enable, 1:FIFO Enable.
CFG	0014	0x00000201	w	Configuration Register, 0-1: Channels to read, '01': right, '10': left, '11': Both (stereo) 2: Sign Extend 3: Left Justify 4-8: Sample Size (0-31)
RX_FIFO_LEVEL	fe00	0x00000000	r	RX_FIFO Level Register
RX_FIFO_THRESHOLD	fe04	0x00000000	w	RX_FIFO Level Threshold Register
RX_FIFO_FLUSH	fe08	0x00000000	w	RX_FIFO Flush Register
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

Name	Offset	Reset Value	Access Mode	Description
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

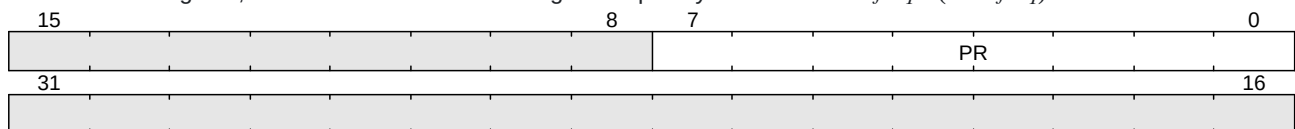
RXDATA Register [Offset: 0x0, mode: r]

The received sample



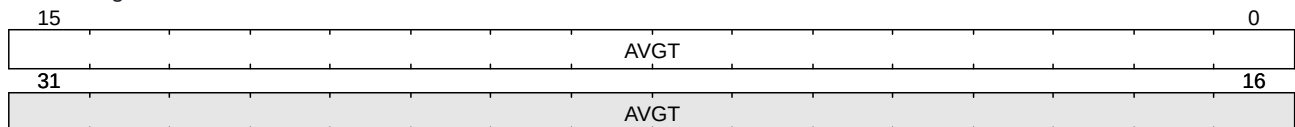
PR Register [Offset: 0x4, mode: w]

The Prescaler register; used to determine the sck signal frequency . $Prescaler = clkfreq / (2xsckfreq) - 1$.



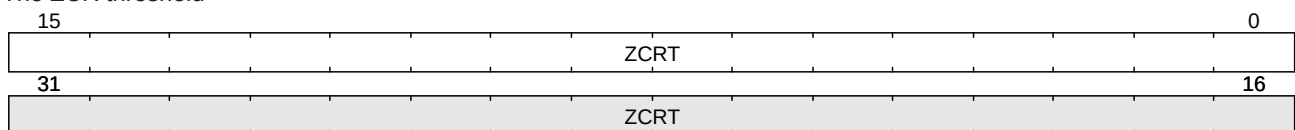
AVGT Register [Offset: 0x8, mode: w]

The Average threshold



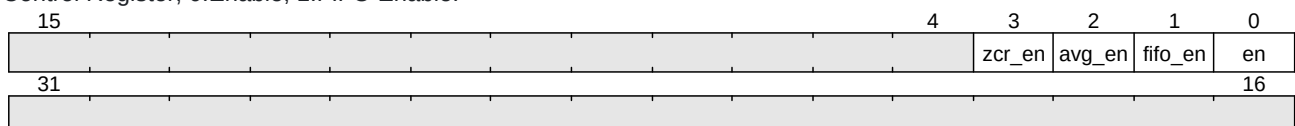
ZCRT Register [Offset: 0xc, mode: w]

The ZCR threshold



CTRL Register [Offset: 0x10, mode: w]

Control Register; 0:Enable, 1:FIFO Enable.

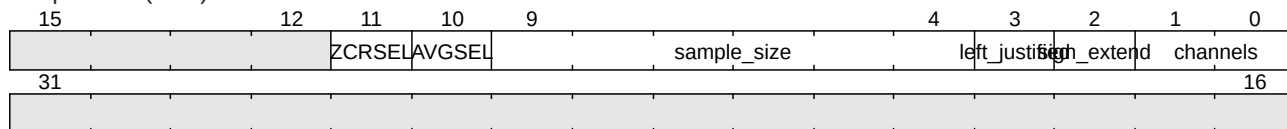


bit	field name	width	description
0	en	1	Enable
1	fifo_en	1	Fifo Enable
2	avg_en	1	Averaging enable

bit	field name	width	description
3	zcr_en	1	ZCR enable

CFG Register [Offset: 0x14, mode: w]

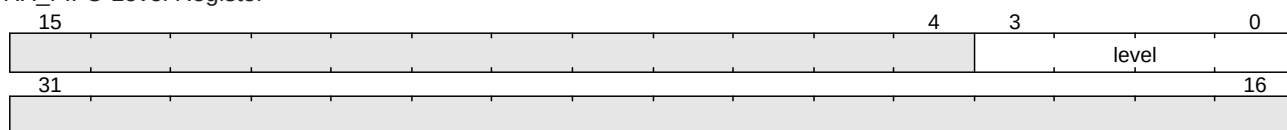
Configuration Register, 0-1: Channels to read, '01': right, '10': left, '11': Both (stereo) 2: Sign Extend 3: Left Justify 4-8: Sample Size (0-31)



bit	field name	width	description
0	channels	2	Channels to read, '01': right, '10': left, '11': Both (stereo)
2	sign_extend	1	Sign Extend
3	left_justified	1	Left justified flag
4	sample_size	6	Sample size
10	AVGSEL	1	Select the number of samples to average
11	ZCRSEL	1	Select the number of samples to calculate the ZCR

RX_FIFO_LEVEL Register [Offset: 0xfe00, mode: r]

RX_FIFO Level Register



bit	field name	width	description
0	level	4	FIFO data level

RX_FIFO_THRESHOLD Register [Offset: 0xfe04, mode: w]

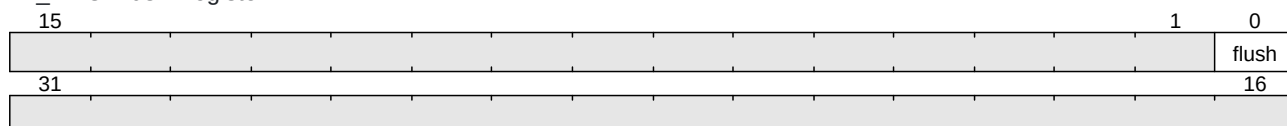
RX_FIFO Level Threshold Register



bit	field name	width	description
0	threshold	4	FIFO level threshold value

RX_FIFO_FLUSH Register [Offset: 0xfe08, mode: w]

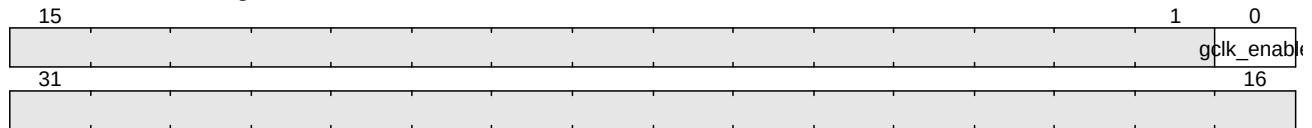
RX_FIFO Flush Register



bit	field name	width	description
0	flush	1	FIFO flush

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	FIFOE	1	Receive FIFO is Empty
1	FIFOA	1	FIFO level is above the set level threshold
2	FIFOE	1	Receive FIFO is Full.
3	AVGF	1	The avg is above the threshold.
4	ZCRF	1	The ZCR is above the threshold.
5	VADF	1	The Voice Activity Detector flag; active when both ZCR & AVG flags are active.

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__d1clkp_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_I2S can be found in the [Drivers](#) directory in the [EFIS](#) (Efabless Firmware Interface Standard) repo. EF_I2S driver documentation is available [here](#). You can also find an example C application using the EF_I2S 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_I2S`.

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 github.com/efabless/EF_I2S`

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.



Module Parameters

Parameter	Description	Default Value
DW	FIFO data width	32
AW	FIFO Address width; Depth=2^AW	4

Ports

Port	Direction	Width	Description
ws	output	1	Word Select; this signal indicates the boundary between left and right audio channels
sck	output	1	Serial Clock; this provides the timing for data transfer, synchronizing the transmission of bits.
sdi	input	1	Serial Data Input; this line carries the audio data being sent to the device.
fifo_en	input	1	FIFO enable

Port	Direction	Width	Description
fifo_rd	input	1	Read from FIFO signal
fifo_level_threshold	input	AW	FIFO Threshold
fifo_flush	input	1	FIFO Flush
fifo_full	output	1	FIFO is full flag
fifo_empty	output	1	FIFO is empty flag
fifo_level	output	AW	The current FIFO level
fifo_level_above	output	1	FIFO level is above threshold flag
fifo_rdata	output	32	Data read from FIFO
sign_extend	input	1	Flag to show if input data is sign extended
left_justified	input	1	Flag to show if input data is left justified
sample_size	input	6	The sample size of input data
sck_prescaler	input	8	The clock prescaler
avg_threshold	input	32	The samples average threshold
avg_flag	output	1	Flag raised when the samples average is above a threshold
avg_en	input	1	Enable average feature
avg_sel	input	1	0 - 32 Samples, 1 - 64 Samples
zcr_threshold	input	32	The samples ZCR threshold
zcr_flag	output	1	Flag raised when the samples ZCR is above a threshold
zcr_en	input	1	Enable average feature
zcr_sel	input	1	0 - 32 Samples, 1 - 64 Samples
vad_flag	output	1	The VAD flag
channels	input	2	Channels used (left, right, or stereo)
en	input	1	Enable signal