**Name:** saif\_pipeline\_stage

**Main Features:**

* Conforms to the [SAIF RTL Standard Interface](../../rtl_standard_interfaces/doc/saif_rsi.doc" \o "/components/rtl/rtl_standard_interfaces/doc/saif_rsi.doc; Reference Outlink) (RSI).
* Optionally pipelines (registers) the interface between two SAIF components to improve timing.
* Configurable datawidth and register location.

**Block Diagram:**

downstream\_cts

upstream\_rts

upstream\_cts

data\_in

ctr\_in

rtr\_out

**SAIF**

**Pipeline**

**Stage**

cts\_in

rts\_out

downstream\_rts

data\_out

Figure 1: Connection Block Diagram

**Application Information:**

The saif\_pipeline\_stage is an optimized, fully tested method of registering SAIF signals to break long asynchronous paths for data or control signals.

One typical use may attach ethernet\_saif data to data\_in/data\_out(31 downto 0), and the EOF, DF, and SOF signals to data\_in/data\_out(34 downto 32) -- like in this example that can be pasted into code with a few edits:

pipeline\_stg1 : saif\_pipeline\_stage

generic map

(

DATA\_WIDTH => 35,

STAGE\_TYPE => 1

)

port map

(

data\_in(31 downto 0) => upstream\_data\_out,

data\_in(32) => upstream\_sof\_out,

data\_in(33) => upstream\_df\_out,

data\_in(34) => upstream\_eof\_out,

rtr\_out => upstream\_cts,

ctr\_in => upstream\_rts,

data\_out(31 downto 0) => downstream\_data\_in,

data\_out(32) => downstream\_sof\_in,

data\_out(33) => downstream\_df\_in,

data\_out(34) => downstream\_eof\_in,

rts\_out => downstream\_ctr,

cts\_in => downstream\_rtr,

);

The STAGE\_TYPE generic allows a user to select where the register is in relation to the inputs and outputs. Set STAGE\_TYPE = 0 will asynchronously connect the input SAIF signals to the output SAIF signals. It could be handy for a user to populate this component to all the places on a SAIF data path where extra pipelining is potentially needed. The user then can turn ON/OFF individual pipeline stage through the STAGE\_TYPE generic to fine-tune the timing performance of the data path without the need of modifying the code.

**Interface:**

Table 1: Generics

|  |  |  |
| --- | --- | --- |
| **Name** | **Range** | **Description** |
| DATA\_WIDTH | 1 to 1200 | Width of data\_in & data\_out busses. |
| STAGE\_TYPE | 0 to 1 | 1: All outputs are registered.  0: No pipelining. |

Table 2: Signals

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Direction** | **Width** | **Description** |
| clk | in | 1 | System clock |
| reset | in | 1 | Synchronous system reset |
| SAIF Input Interface | | | |
| data\_in | in | DATA\_WIDTH | SAIF Data input bus |
| ctr\_in | in | 1 | Clear to receive |
| rtr\_out | out | 1 | Ready to receive |
| SAIF Output Interface | | | |
| data\_out | out | DATA\_WIDTH | SAIF Data output bus |
| rts\_out | out | 1 | Ready to send |
| cts\_in | in | 1 | Clear to send |

**Timing Specifications and Diagrams:**

See the SAIF RSI.

**Approximate Size:**

(2\*DATA\_WIDTH + 3 flipflops) \* STAGE\_TYPE