Dual Priority Fifo

interface

pin	direction	job
clk	input	clock
rst_n	input	reset (active neg). async reset.
vldin	input	new data valid
hipri	input	new data is hi priority
din[WID-1:0]	input	new data
readout	input	pop the current data out. disregard if empty. on next clock next data appears.
dout[WID-1:0]	output	current valid data (if not empty, undefined otherwise.
empty	output	fifo is empty. reading is blocked.
full	output	fifo is full. writing in is blocked
count[15:0]	output	number of valid entries in fifo

Parameters (Verilog params)

fifo code is defined by two parameters:

1. **WID** : width of data entry.
2. **DEPTH** : total number of entries in the fifo. Best if it is not divided between priorities.

Description

New entry is written on each vldin pulse. Entry is qualified by high priority input. If it is high priority, it will appear on the output (dout) ahead of all previous low priorities (This may take few clocks). Order within priorities should be observed.

Assignment

write code for dual priority fifo. write test bench (tb.v) to test basic functionality. Observe high priority entries overcome lower priority. Idea: First, test all entries of the same priority.

Help stuff

You can start by looking at “syncfifo.v” which is ubiquitous synchronous fifo in Ilia’s vlsistuff git hub repository. Clone the repository and then do “find” for this file name inside it.