REORDERING FIFO

regular fifo with a twist.

Pinout :

pin	direction	job	notes
clk	input	system clock	
rst_n	input	active low async reset	
vldin	input	new data valid for writing	pulse
order[2:0]	input	position of this data relative to others	this is THE input, look below.
din[WID-1:0]	input	new data	
ok_to_write	output	back pressure to hold writing	
readout	input	read the current dout data	
dout[WID-1:0]	output	data that is read out.	top of the fifo is always present on dout. (not clock after read).
empty	output	data out is not valid	
count[15:0]	output	number of valid entries in this fifo	
Parameters			
WID	integer	Width of data in the fifo.	
DEPTH	integer	Number of entries in the fifo.	

Idea

This fifo can behave like regular synchronous fifo. But sometimes it needs to re-arrange the entries and output them not in the incoming order.

For this end, there is a new input : **order[2:0]** when it is zero (on vldin) nothing changes. First in First out. However when it is non-zero, it may start one of 3 distinct sequences

* **1, 0** : sequence of length two. second entry must be outputted first, followed by entry “1”
* **1,2,3,0 or 2,3,0,1 or 3,0,1,2** : entries should be outputted accordingly.
* **1,2,3,4,5,6,7,0 or 2,3,4,…. or 6,7,0,1,2,3,4,5 or 7,0,1,2,3,4,5,6**

Sequences can be of length 2 or 4 or 8.

*** Bonus point to anyone guessing where this order crawled from? ***

Constraints and relaxes

1. the DEPTH can be fixed at 8 or a little bigger.
2. Additional temporary entries are allowed, but less is better.
3. ask anything, so we can add more points here.

Sequence Example (4 reorders)

vldin data	order		readout	dout	
				empty	
1	0			1	
2	0			1	
3	0			1	
4	0		1	1	
			1	2	
			1	3	
				3	
12	2		1	4	needs to support write and read together
13	3			empty	
10	0			empty	
11	1			empty	
			1	10	
			1	11	
			1	12	
			1	13	
				empty	