RMS block

This block calculates the RMS value of a series of numbers. The RMS calculation is performed according to:

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}.$$

The interface has a flag to indicate the beginning of a sequence of numbers. The input is a one flag push interface, and the output is a two flag pull interface. The input has a two bit control code. The Code is:

Code	Function
00	X value
01	Remove X value from RMS calculation
10	Produce result and continue (X is part of RMS)
11	Produce result and clear (X is part of RMS)

Your code should calculate the number of samples seen, the sum of squares, and when a result is required, The RMS value. Removing an X value requires decrementing n, and subtracting X^2 from the sum.

The samples are 32 bit signed numbers. The result of X2 is 64 bits. The sum is 72 bits. Dividing by n should yield 64 bits, and the square root will yield 32 bits.

The divide and square root cannot be implemented with verilog operators (They will not synthesize).

Divide and square root algorithms can be found on the web. They are about the same. Do not copy code from the web. If two submission have similar code, they receive 0... The divide and square root must be pipelined (A lot of stages... >60) . The correct square root is the largest integer which squared is less than or equal to the number. The system can request a result every clock cycle.

You should have at least a 32 entry output interface FIFO.

The following table describes the interface signals:

Name	Dir	Bits	Description
clk	In	1	Positive edge system clock
rst	In	1	Active high reset signal (Asynchronous)
pushin	In	1	Signal indicating presence of input data/commands
cmdin	In	2	Describes what is to be done with the data
Xin	In	32	Data into the block

Name	Dir	Bits	Description
pullout	In	1	A pull signal from the testbench
stopout	Out	1	Indicates there is not valid results available
Xout	Out	32	Results from the RMS calculation