Convolution Kernel

Diagram

Description

A N-bit K x K convolution kernel processor for a M x M input matrix.
Uses a stride of 1 since with 1-stride output any other stride output can be extracted.
Uses array multipliers and carry lookahead adders via Vector Dot Product Units.
Note: The output has double the bit width of the input.
NOTE: To compile this file please set modelsim to use VHDL 2008 for all files. See:
https://forums.xilinx.com/t5/Synthesis/Array-of-Unconstrained-Array/td-p/681253
This is combinational circuit.
If enable != '1' then the output is set to high impedence.

Generics and ports

Table 1.1 Generics

Generic name	Type	Value	Description
bit_width	integer	4	Bit width for every integer.
input_matrix_size	integer	7	The matrix dimension M x M.
kernel_size_pow_of_2	integer	2	The kernel dimension 2^K x 2^K. This is powered by 2 to use logrithmic addition in Vector Dot Product Unit.

Table 1.2 Ports

Port name	Direction	Type	Description
Kernel	in	matrix_T(2 ** kernel_size_pow_of_2 - 1 downto 0)(2 ** kernel_size_pow_of_2 - 1 downto 0)(bit_width - 1 downto 0)	Input Kernel to use. Must be of dimension 2^K x 2^K.
Matrix	in	matrix_T(input_matrix_size - 1 downto 0)(input_matrix_size - 1 downto 0)(bit_width - 1 downto 0)	Input matrix to apply kernel on. Must have dimension of M x M.
D	out	matrix_T(input_matrix_size - 2 ** kernel_size_pow_of_2 downto 0)(input_matrix_size - 2 ** kernel_size_pow_of_2 downto 0)(2 * bit_width - 1 downto 0)	Output result A.B (dot product). Dimension is square matrix of size M - 2^K + 1. Output also has 2*bit_width for all integers.
enable	in	std_logic	If not = '1', output is high impedence.

Signals, constants and types

Signals

Name	Type	Description
FlatKernel	vector_T(2 ** (2 * kernel_size_pow_of_2) - 1 downto 0)(bit_width - 1 downto 0)	Stores flattned kernel
FlatMatrixPieces	matrix_T((input_matrix_size - 2 ** kernel_size_pow_of_2 + 1) ** 2 - 1 downto 0)(2 ** (2 * kernel_size_pow_of_2) - 1 downto 0)(bit_width - 1 downto 0)	Stores flattned matrix peices

Processes

- **FlattenKernel**: *(Kernel)*