Parameterized Sorter

December 24, 2001

The Sorter accepts \( (N) \) patterns arriving upon rising edge of the CLOCK signal. Every pattern contains \( (M) \) bits. Sorter selects \( (P) \) best patterns out of \( (N) \) \( (P \ll N) \) based on value of each pattern: the larger the pattern, the better it is for sorting purpose. In case of several equal patterns the pattern with a largest number should be considered as the best one. For example, if pattern_7 = pattern_5 = pattern_2, then pattern_7 will be the first best, pattern_5 will be the second best and pattern_2 will be the third best.

The Sorter outputs \( (P) \) results, or Products, in ranked order, i.e. Product_1 always corresponds to the first best pattern, Product_2 corresponds to the second best pattern and so on. Each product comprises \( (N) \) bits and represents a number of the winning pattern in a binary code. Only one bit out of these \( (N) \) should be “1”, all the rest must be “0”. For example, if \( N=6, P=3 \), pattern_2 is the first best, pattern_4 is the second best and pattern_5 is the third best, then the products should be:

Product_1 = “000010”
Product_2 = “001000”
Product_3 = “010000”

In the example above, if only two patterns have a non-zero value, then Product_3=“000000”. If only Pattern_1 has a non-zero value, then Product_1=Product_2=“000000”. In general, if all input patterns arriving at a specific clock tick, are zeroes, then all Products corresponding to these patterns, should be zeroes as well.

All inputs and outputs should be numbered in descending order: PAT[I][M-1..0], PROD[j][N-1..0] etc. All outputs, as well as inputs, are latched into flip-flops (FF) on the rising edge of CLOCK (see block diagram on Fig.1). The total number of Sorter inputs is \( N*M+1 \) and total number of outputs is \( P*N \).

Sorting should be done in two stages (Fig.1). All possible comparisons between \( (N) \) patterns are performed at a first stage. Based on \( N*(N-1)/2 \) results of comparisons, \( (P) \) products are simultaneously selected at the second stage. Sorting project should be implemented in VHDL (prefered) or Verilog using \( (N) \), \( (M) \) and \( (P) \) variables. Two cases of special interest for the CSC Trigger Electronics should be functionally simulated and optimized for the best performance (maximum operating frequency). The first case is with \( (N=18, M=4 \) and \( P=3 \). The second case is with \( (N=36, M=7 \) and \( P=4 \).
Select 1st best product

Select (P) best product

Figure 1: Sorter Block Diagram