2 Booth Multiplier

I proceeded with the idea described in the document uploaded on Moodle. I used a variable indPair to maintain the state of the machine (the set of indices from b that we were to use in the current iteration). indPair = '1' meant the multiplication had just begun and b(3) and b(2) are used; indPair = '0' meant we were to use b(1) and b(0) and add it to the partial product computed previously. The variable sequentially alternates between 0 and 1, producing outputs every two clock cycles. The code is split into four processes, inspired by the simpleStateMachine.vhd file provided earlier.
The first handles the change of the variables that depend on the clock (indPair and nextIndPair)
The next two handle the logical implications of a certain value of indPair (ie. selecting the bits from b, the cumulative sum 'prevResult' and fixing the signal toAdd).
The last process handles the communication of 'sum' to 'result' port.
I have also used two 8-bit adders to sum up 'prevResult' and toAdd (and store in 'sum'), and also to procure 3A by summing 4A and A, stored in a variable temp3, used if necessary by one of the middle processes.
Each multiplication requires two clock cycles to complete (and one for reset in the beginning of the simulation).