1. An ASIC has 16x 32-bit input ports and a 16-bit register A with each bit representing data valid on each port. E.g. if ports 0, 7, 9 have valid input, register A would have a value of 0x0281.  
     
   Design and implement Verilog code such that on each positive clock edge, the values of all valid ports would be packed into a buffer of 16x32b entries, and skipping all invalid ports in between. The total number of valid ports would also be stored into register B.  
     
   E.g. if register A = 0x0281 on cycle 0, your code would generate the following output on cycle 1:  
   * Register B = 3
   * Buffer[0] = Port 0  
     Buffer[1] = Port 7  
     Buffer[2] = Port 9
   * Buffer[15:3] = 0