This pcore originates from opb_bram_if_cntlr_v1_00_a

- added generic MY_NUM_WRITE_ENABLES to bram_if.vhd
- added generic MY_NUM_WRITE_ENABLES to opb_bram_if_cntlr.vhd
- modified attribute NUM_WRITE_ENABLES to equal MY_NUM_WRITE_ENABLES in opb_bram_if_cntlr.vhd
- added feedback mux for bram data
- inserted one clk delay for ack during write access

NOTE: In order to have only 1 bram enumerated, the address range on the controller must
      be 2K. I.E. Base=0xfffff800, High=0xffffffff

This pcore still relies on platgen to enumerate the bram. The controller is a 32bit device
so the datalanes match one for one. Read accesses haven't changed from the original bram cntlr
so they continue to be 3 clks. Write accesses have been extending so now the access time to
the bram to 3 clks. During a write access the cycles are as follows:
  - 1st clk presents address to the bram
  - 2nd clk reads data from the bram, decodes the BE signals, and muxes the matching data
    lanes. So the new data that contains a mix of orig. bram contents and opb write data is
    presented to the bram.
  - 3rd clk writes the new mux'd data to the bram and acks the opb bus.

MHS Usage:

```
    BEGIN opb_bram_if_cntlr
     PARAMETER INSTANCE = opb_bram_if_cntlr_1
     PARAMETER HW_VER = 1.00.b
     PARAMETER c_BASEADDR = 0xFFFFf800
     PARAMETER c_HIGHADDR = 0xffffffff
     BUS_INTERFACE SOPB = opb
     BUS_INTERFACE PORTA = porta
    END

    BEGIN bram_block
     PARAMETER INSTANCE = bram_block_1
     PARAMETER HW_VER = 1.00.a
     BUS_INTERFACE PORTA = porta
    END
```