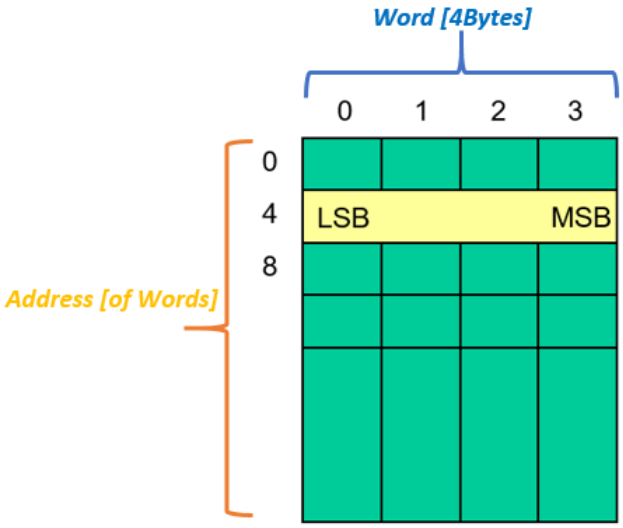
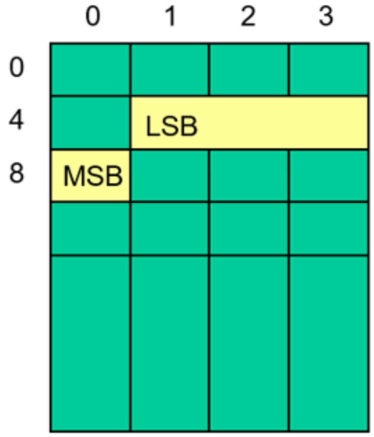
VHDL_DUT_after vs. ***VHDL_DUT_before***

The difference between ***VHDL_DUT_after*** and ***VHDL_DUT_before*** designs based on the way we access to a real Alignment Memory. Our case of 32-bit Word alignment, the physical access address is to be padded with 2-bit of zero value, means that $\boldsymbol{AddressValue} \ \% \ \mathbf{4} = \mathbf{0}$ (as in ***VHDL_DUT_after*** version while the memory access in ***VHDL_DUT_before*** is byte access – you can explore the results through their given STP files).

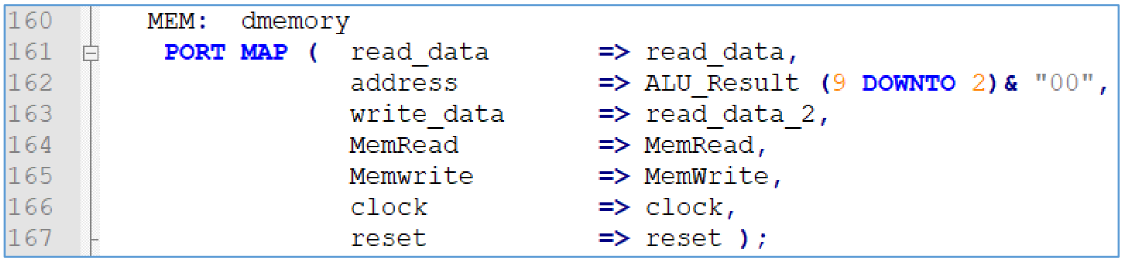
Note: the memory access with ***VHDL_DUT_before*** design in ModelSim works well (simulation case behaves that each address value associated with a Word).



Aligned memory



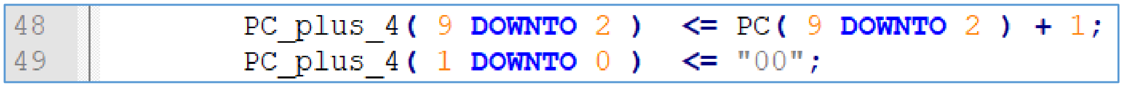
Unaligned memory



```
   MEM:  dmemory
    PORT MAP (  read_data       => read_data,
                address         => ALU_Result (9 DOWNTO 2)& "00",
                write_data      => read_data_2,
                MemRead         => MemRead,
                Memwrite        => MemWrite,
                clock           => clock,
                reset           => reset );
```

See ***DMEMORY.vhd***

In ***VHDL_DUT_after***

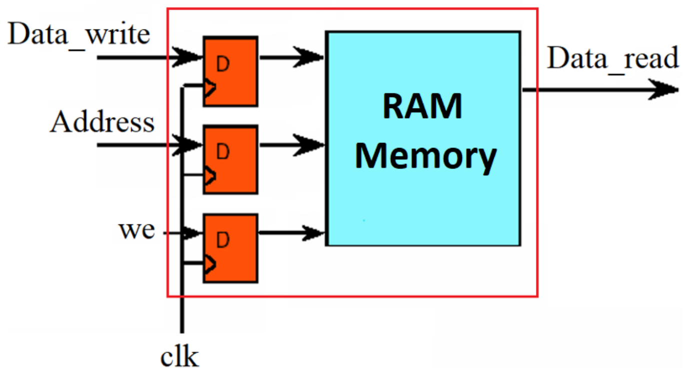


```
        PC_plus_4( 9 DOWNTO 2 )  <= PC( 9 DOWNTO 2 ) + 1;
        PC_plus_4( 1 DOWNTO 0 )  <= "00";
```

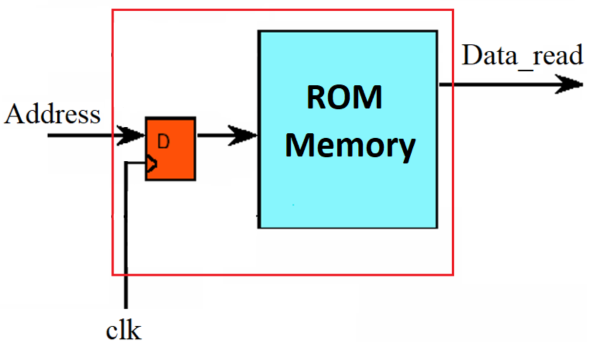
See ***IFETCH.vhd***

In ***VHDL_DUT_after***

The given I-Chace (ROM type) and D-Chace (RAM type) memories are 4kB wide each:



Single Port RAM with unregistered output



Single Port ROM with unregistered output