⚙️ Key Optimization Techniques in VLSI FIR Filters:

* Symmetric Coefficient Sharing: Reduces number of multipliers.
* Distributed Arithmetic (DA): Replaces multipliers with LUTs.
* Bit-serial/Multiplierless Architecture: Reduces dynamic power.
* Clock Gating: Disables unused logic to save power.
* Pipelining: Balances speed and power tradeoff.

🗒️ Example: FIR Filter VHDL Code with Low-Power Awareness (Simple 4-Tap Filter)

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;


entity fir_filter is

   Port ( clk    : in  STD_LOGIC;

          reset  : in  STD_LOGIC;

          enable : in  STD_LOGIC;

          x_in   : in  STD_LOGIC_VECTOR(7 downto 0);

          y_out  : out STD_LOGIC_VECTOR(15 downto 0));

end fir_filter;


architecture Behavioral of fir_filter is


   type coeff_array is array (0 to 3) of integer := (2, 4, 4, 2); -- symmetric coefficients

   signal x_reg : array (0 to 3) of STD_LOGIC_VECTOR(7 downto 0);

   signal acc   : integer := 0;


begin
```

```
process(clk, reset)
begin
   if reset = '1' then
      x_reg(0) <= (others => '0');
      x_reg(1) <= (others => '0');
      x_reg(2) <= (others => '0');
      x_reg(3) <= (others => '0');
      acc <= 0;
      y_out <= (others => '0');

   elsif rising_edge(clk) then
      if enable = '1' then  -- clock gating control
         x_reg(3) <= x_reg(2);
         x_reg(2) <= x_reg(1);
         x_reg(1) <= x_reg(0);
         x_reg(0) <= x_in;

         acc <= coeff_array(0) * to_integer(unsigned(x_reg(0))) +
               coeff_array(1) * to_integer(unsigned(x_reg(1))) +
               coeff_array(2) * to_integer(unsigned(x_reg(2))) +
               coeff_array(3) * to_integer(unsigned(x_reg(3)));

         y_out <= std_logic_vector(to_unsigned(acc, 16));
      end if;
   end if;
end process;
```

```
end Behavioral;
```