

- V opisu vezja deklariraj 8-bitni standardni vektor **reg**, ki bo predstavljal notranji register. Vrednost registra v vmesniku naj določa faktor skaliranja 14-bitnega analognega vhodnega signala (`adc_i`). Vhodni signal predstavlja vrednosti vzorčene z enim kanalom ADC, ki so zapisane v dvojiškem komplementu, zato jih najprej pretvori v 14-bitni notranji signal tipa `signed`. Vrednost v registru vmesnika pa naj bo 8-bitno nepredznačeno število, ki ga pretvori v 9-bitno predznačeno število. Rezultat množenja je 23-bitni vektor, ki ga skrči nazaj na 14-bitni izhod (`adc_o`). Če pride pri množenju do prekoračitve območja 14-bitnih vrednosti, naj gre izhodna vrednost v pozitivno ali negativno nasičenje.
- Opiši logiko vmesnika za sistemsko vodilo, prek katerega bo procesor nastavljal vrednost registra **reg**. Naredi sinhroni proces (`rising_edge(clk_i)`), ki ob aktivnem signalu `rstn_i='0'` postavi register na 0, ob signalu `sys_wen='1'` in pogoju `sys_addr(19 downto 0)=X"00000"` pa prenese v register spodnjih 8 bitov iz vodila **sys_wdata**. Dodaj še asinhrono logiko za branje registra in kontrolne izhode:

```

sys_err <= '0';

pbusr: process(sys_addr, sys_wen, sys_ren, reg)
begin
    if (sys_wen or sys_ren)='1' then
        sys_ack <= '1';
    end if;
    if sys_addr(19 downto 0)=X"00000" then
        sys_rdata <= X"000001" & reg;
    else
        sys_rdata <= (others=>'0');
    end if;
end process;

```

- Preveri sintakso in naredi simulacijo komponente.