VOLUME / Sw2 (SWITCH 2)

REGOLAZIONE DEL GUADAGNO TOTALE

Come già dimostrato (vedi *blocco α*), $V_u < V_i$, quindi è presumibile che sia necessaria un'amplificazione del segnale totale $s[n]$. Ciò è realizzato mediante la moltiplicazione del segnale d'uscita per un parametro *v*, ottenendo così $v \cdot s[n]$, che quindi è regolabile, in ampiezza, a piacimento. Tale selezione del volume è ottenuta mediante un'appropriata combinazione dei micro-switches presenti nel blocco a Sw_2, che pilota il blocco *volume*, il quale effettua la moltiplicazione.

REALIZZAZIONE

Si è ritenuto sufficiente fornire all'utente una possibile scelta tra 4 diverse intensità sonore derivanti da una variazione di tensione pari a:

- 0×: muto;
- 1×: in-out;
- 2×: raddoppio;
- 3×: triplicazione.

CODICE ARCHITETTURALE

```
-- volume.vhd
-- Canziani Alfredo & Viviani Emanuele production
-- DEEI, Università degli studi di Trieste

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-- %                                                                            %
-- %                         LIBRERIE INCLUSE                                   %
-- %                                                                            %
-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-- %                                                                            %
-- %                           PORTE DI VOLUME                                  %
-- %                  (terminali di ingresso ed uscita)                         %
-- %                                                                            %
-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

entity volume is
    Port (   switch_vol    : in  std_logic_vector(1 downto 0);
             audio_in      : in  unsigned(15 downto 0);
             audio_out     : out unsigned(15 downto 0)
             );
end volume;

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
-- %                                                                            %
-- %                            FUNZIONAMENTO                                   %
-- %                                                                            %
-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
architecture Comportamento of volume is

begin
    process (switch_vol, audio_in)
    begin

    case (switch_vol) is  --gli switch si chiudono a massa e si aprono a Vcc
        when "11" =>
            audio_out <= "0000000000000000";    -- muto (x 0)

        when "01" =>
            audio_out <= audio_in;              -- In-Out (x 1)

        when "10" =>                            -- x 2
            audio_out <= audio_in + audio_in;

        when others =>                          -- x 3
            audio_out <= audio_in + audio_in + audio_in;

        end case;

    end process;
end Comportamento;
```