

kb2freq.vhd

```
--kb2freq.vhd
--
--keyboard to frequency converter for keyboard tone generator
--keys: Q-I, A-K, Z-,

library IEEE;
use IEEE.std_logic_1164.all;
entity kb2freq is
    port(    scan: in std_logic_vector(7 downto 0);
            freq: out std_logic_vector(24 downto 0)); --MSB is valid bit
end kb2freq;

architecture behavioral of kb2freq is
begin
    process(scan)
    begin
        case scan is
            when x"15" => freq <= "110000000000000000000000"; --Q
            when x"1D" => freq <= "101000000000000000000000"; --W
            when x"24" => freq <= "100100000000000000000000"; --E
            when x"2D" => freq <= "100010000000000000000000"; --R
            when x"2C" => freq <= "100001000000000000000000"; --T
            when x"35" => freq <= "100000100000000000000000"; --Y
            when x"3C" => freq <= "100000010000000000000000"; --U
            when x"43" => freq <= "100000001000000000000000"; --I
            when x"1C" => freq <= "100000000100000000000000"; --A
            when x"1B" => freq <= "100000000010000000000000"; --S
            when x"23" => freq <= "100000000001000000000000"; --D
            when x"2B" => freq <= "100000000000100000000000"; --F
            when x"34" => freq <= "100000000000010000000000"; --G
            when x"33" => freq <= "100000000000001000000000"; --H
            when x"3B" => freq <= "100000000000000100000000"; --J
            when x"42" => freq <= "100000000000000010000000"; --K
            when x"1A" => freq <= "100000000000000001000000"; --Z
            when x"22" => freq <= "100000000000000000100000"; --X
            when x"21" => freq <= "100000000000000000010000"; --C
            when x"2A" => freq <= "100000000000000000001000"; --V
            when x"32" => freq <= "100000000000000000000100"; --B
            when x"31" => freq <= "100000000000000000000010"; --N
            when x"3A" => freq <= "100000000000000000000001"; --M
            when x"41" => freq <= "1000000000000000000000001"; --,
            when others => freq <= "000000000000000000000000"; --invalid
        end case;
    end process;
end behavioral;
```