

15-1 認識聲音

聲音是一種波動，聲音的振動會引起空氣分子有節奏的振動，使周圍的空氣產生疏密變化，形成疏密相間的縱波，因而產生了聲波。人耳可以聽到的聲音頻率範圍在 20Hz~20kHz 之間。常用的聲音輸出裝置如蜂鳴器 (buzzer) 及揚聲器 (loudspeaker)，如圖 15-1(a)所示為蜂鳴器，可以分成有源蜂鳴器和無源蜂鳴器兩種，有源又稱為自激式，內含驅動電路，必須加直流電壓，而且只能發出單一固定頻率的聲音輸出。無源又稱為它激式，沒有內部驅動電路，加上不同頻率的交流訊號可以產生不同頻率的聲音輸出。如圖 15-1(b)所示為揚聲器或稱為喇叭，輸出功率較蜂鳴器大，音質也較蜂鳴器好。



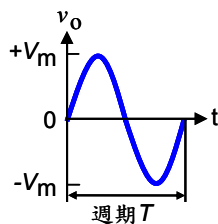
(a) 蜂鳴器



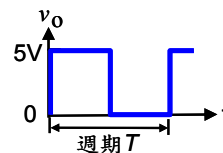
(b) 揚聲器

圖 15-1 聲音輸出裝置

如圖 15-2(a)所示正弦波為組成聲音的基本波形，聲音音量與振幅成正比；聲音音調與頻率成正比，與週期成反比；聲音音長與輸出時間成正比。在數位電路中常以圖 15-2(b)所示方波來模擬正弦波，方波是由奇次諧波 (harmonic) 所組成，由於數位電路的頻寬有限，方波只能以有限的頻寬來合成，諧波的數量越多，波形更接近方波。



(a) 正弦波



(b) 方波

圖 15-2 聲音信號

15-2 單音產生器實習

□ 功能說明：

使用 CPLD 晶片，設計產生單音頻率的數位電路。利用兩個按鍵開關控制 CPLD 產生 10Hz、1kHz 等兩種單音頻率輸出，當按下 SW0 鍵時，輸出 10Hz 單音訊號；當按下 SW1 鍵時，輸出 1kHz 單音訊號。

□ 電路方塊圖：

如圖 15-3 所示為單音產生器電路方塊圖，包含除頻電路、按鍵電路、多工電路等三個部份。

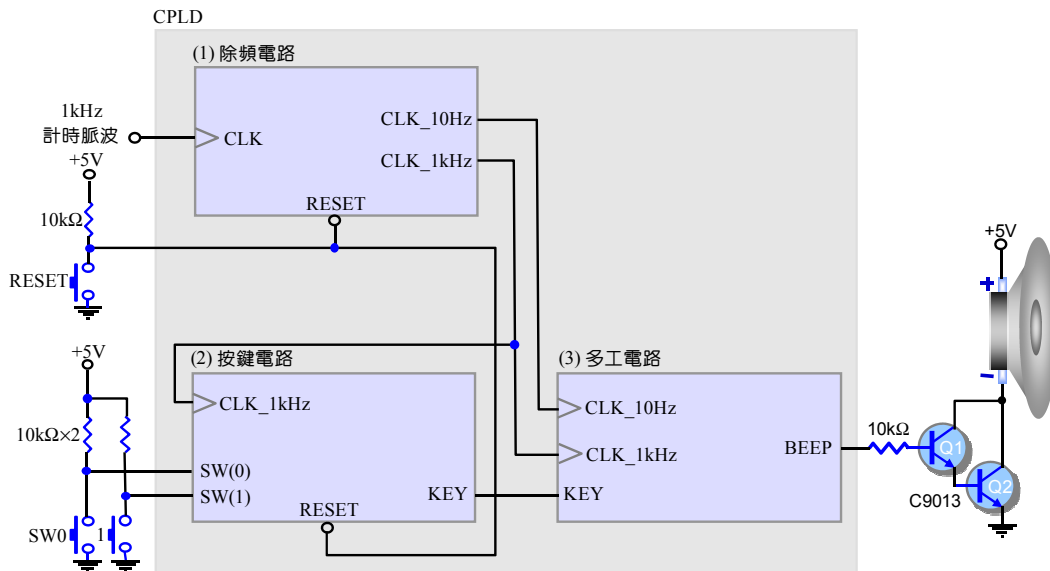


圖 15-3 單音產生器電路方塊圖

輸入使用 1kHz 計時脈波訊號，加入按鍵電路的脈波輸入端 CLK_1kHz。經由 CPLD 除頻電路除頻產生 10Hz 脈波，並且加入多工電路。按鍵電路的功能是檢測並儲存按鍵狀態至 KEY 中。多工電路的功能是依按鍵狀態輸出 10Hz、1kHz 等單音脈波訊號至喇叭。

RESET、CLK、SW、BEEP 等為外部 port 訊號；而 CLK_10Hz、CLK_1kHz、KEY 等則為內部 signal 訊號。

□ 程式： tone.vhd

```

library ieee;                                --使用 ieee 零件庫。
use ieee.std_logic_1164.all;                 --使用 std_logic_1164 套件。
use ieee.std_logic_arith.all;               --使用 std_logic_arith 套件。
use ieee.std_logic_unsigned.all;           --std_logic_unsigned 套件。
entity TONE is
  port
    ( RESET,CLK:in std_logic;                --重置輸入，計時脈波輸入。
      SW:in std_logic_vector(1 downto 0);     --按鍵輸入。
      BEEP:out std_logic );                  --聲音輸出。
  end TONE;
  architecture arch of TONE is
    signal KEY:std_logic_vector(1 downto 0);  --按鍵狀態。
    signal DIV1,DIV2,CNT1:integer range 0 to 100;--除 100。
    signal CLK_10Hz:std_logic;               --10Hz 脈波訊號。
    signal CLK_1kHz:std_logic;              --1kHz 脈波訊號。
  begin
    -- (1)除頻電路
    process (CLK, RESET)
  begin
    if RESET='0' then                       --按 RESET 鍵?
      DIV1<=100;                              --除 100。
      DIV2<=DIV1/2;                            --50%工作週期 10Hz 脈波。
    elsif CLK'event and CLK='1' then      --計時脈波正緣?
      if CNT1=DIV1 then                    --除 100。
        CNT1<=1;
      else
        CNT1<=CNT1+1;
      end if;
      if (CNT1=DIV1) or (CNT1=DIV2) then  CNT1=100? 或 CNT1=50?
        CLK_10Hz<=not CLK_10Hz;            --10Hz 脈波。
      end if;
    end if;
  end process;

```

-- (2) 按鍵電路

```

process (CLK, RESET)
begin
  if RESET='0' then -- 按 RESET 鍵?
    KEY<="11"; -- 按鍵狀態初始值。
  elsif CLK'event and CLK='1' then -- 計時脈波正緣?
    if SW(0)='0' or SW(1)='0' then -- 按下任意鍵?
      KEY<=SW; -- 儲存按鍵狀態。
    end if;
  end if;
end process;

```

-- (3) 多工電路

```

case KEY is
  when "10" => BEEP<=CLK_10Hz; -- 按 SW0 鍵，輸出 10Hz 脈波。
  when "01" => BEEP<=CLK; -- 按 SW1 鍵，輸出 1kHz 脈波。
  when others => BEEP<='0'; -- 未按任意鍵，輸出邏輯 0。
end case;
end process;
end arch;

```

□ 時序波形：

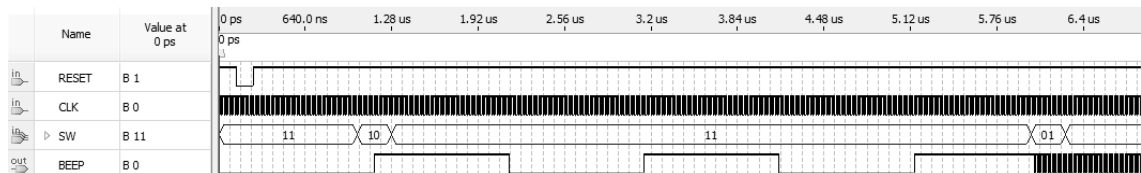


圖 15-4 單音產生器電路的時序波形



1. 使用 CPLD 晶片，設計單音產生器的數位電路。利用三個按鍵開關控制 CPLD 產生 10Hz、100Hz、1kHz 等三種單音頻率輸出，當按下 SW0 鍵時，輸出 10Hz 訊號；當按下 SW1 鍵時，輸出 100Hz 訊號；當按下 SW2 鍵時，輸出 1kHz 訊號。(tone1.vhd)
2. 接續上題，增加三個 LED 指示輸出脈波訊號，當按下 SW0 鍵時，輸出 10Hz 訊號且 LED0 亮；當按下 SW1 鍵時，輸出 100Hz 訊號且 LED1 亮；當按下 SW2 鍵時，輸出 1kHz 訊號且 LED2 亮。(tone2.vhd)

15-5 電子音樂盒實習

□ 功能說明：

使用 CPLD 晶片，設計播放如圖 15-12 所示鋼琴入門音樂—小蜜蜂 (Little Bee) 第一段音符的數位電路。演奏速度為每分鐘 300 拍，每拍為 200ms。

小蜜蜂 (Little Bee)																				
	5	3	3	-		4	2	2	-		1	2	3	4		5	5	5	-	
	5	3	3	-		4	2	2	-		1	3	5	5		3	-	-	-	
	2	2	2	2		2	3	4	-		3	3	3	3		3	4	5	-	
	5	3	3	-		4	2	2	-		1	3	5	5		1	-	-	-	

圖 15-12 小蜜蜂簡譜

□ 電路方塊圖：

如圖 15-13 所示為電子音樂盒電路方塊圖，包含音符選擇電路、音符產生電路、多工電路等三個部份。

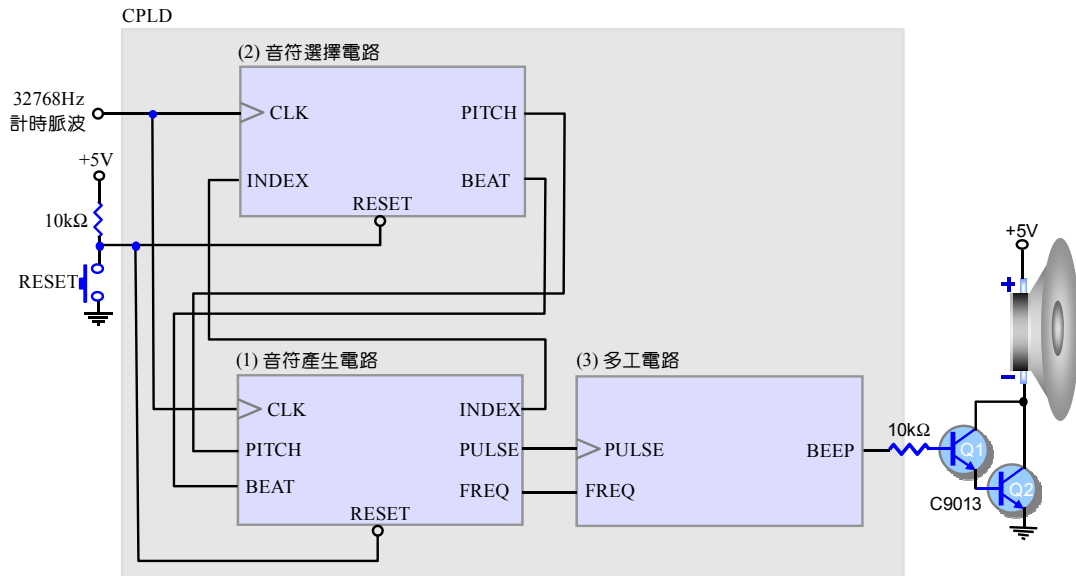


圖 15-13 電子音樂盒電路方塊圖

輸入使用 32768Hz 計時脈波訊號，加入音符選擇電路及音符產生電路的脈波輸入端 CLK。音符產生電路的功能是依音調 PITCH 及節拍 BEAT 產生所須音符數值。音符選擇電路的功能是依索引值 INDEX 選擇所須音符的音調及節拍，並將其加入音符產生電路輸入端。多工電路的功能是依不同音符產生音符訊號輸出。

RESET、CLK、BEEP 等為外部 port 訊號；而 PITCH、INDEX、PULSE、FREQ 等則為內部 signal 訊號。

□ 程式： music.vhd

```

library ieee;                                --使用 ieee 零件庫。
use ieee.std_logic_1164.all;                 --使用 std_logic_1164 零件庫。
use ieee.std_logic_arith.all;               --使用 std_logic_arith 零件庫。
use ieee.std_logic_unsigned.all;           --std_logic_unsigned 零件庫。
entity MUSIC is
  port
  ( RESET:in std_logic;                      --重置輸入。
    CLK:in std_logic;                        --32768Hz 計時脈波輸入。
    BEEP:out std_logic );                   --聲音輸出。
end MUSIC;
architecture arch of MUSIC is
  signal PULSE:std_logic;                   --聲音訊號輸出。
  signal PITCH:integer range 0 to 8;        --音調。
  signal BEAT,LEN:integer range 0 to 4;    --節拍。
  signal COUNT:integer range 0 to 127;     --音符除數。
  signal FREQ:integer range 0 to 127;      --音符除數。
  signal INDEX:integer range 0 to 25;      --音樂總音符數。
  signal DIVIDER:integer range 0 to 16384; --節拍計時器。
begin
  --(1) 音符產生電路
  process (CLK, RESET)
  begin
    case PITCH is
      when 1=>FREQ<=32768/523;              --中音音符 C (Do) 頻率除數。
      when 2=>FREQ<=32768/587;              --中音音符 D (Re) 頻率除數。
    end case;
  end process;
end arch;

```

when 3=>FREQ<=32768/659;	-- 中音音符 E(Mi) 頻率除數。
when 4=>FREQ<=32768/698;	-- 中音音符 F(Fa) 頻率除數。
when 5=>FREQ<=32768/784;	-- 中音音符 G(So) 頻率除數。
when 6=>FREQ<=32768/880;	-- 中音音符 A(La) 頻率除數。
when 7=>FREQ<=32768/988;	-- 中音音符 B(Si) 頻率除數。
when 8=>FREQ<=32768/1046;	-- 高音音符 C(Do) 頻率除數。
when others =>FREQ<=0;	
end case ;	
if RESET='0' then	-- 按 RESET 鍵?
PULSE<='0';	-- 清除音符脈波訊號。
DIVIDER<=0;	-- 清除除頻器。
LEN<=0;	-- 清除節拍值。
elsif CLK'event and CLK='1' then	-- 計時脈波正緣?
if FREQ=0 then	-- 音符頻率除數為 0?
COUNT<=0;	
else	-- 音符頻率除數不為 0。
COUNT<=COUNT+1;	-- 開始計算並產生音符訊號。
if COUNT=FREQ/2 then	
PULSE<= not PULSE;	
COUNT<=0;	
end if ;	
end if ;	
if DIVIDER>=6553 then	-- 1 拍時間等於 6553/32768 秒。
DIVIDER<=0;	
if LEN<BEAT then	
LEN<=LEN+1;	
else	
LEN<=0;	
INDEX<=INDEX+1;	-- 下一個音符。
if INDEX>25 then	-- 已取完所有音符?
INDEX<=0;	-- 重新讀取第一個音符。
end if ;	
end if ;	
else	
DIVIDER<=DIVIDER+1;	-- 節拍計時器加 1。

```

    end if;
end if;
end process;
-- (2) 音符選擇電路
process (CLK, RESET)
begin
    case INDEX is
        when 0=>    PITCH<=5;           -- 音符 so。
                   BEAT<=1;           -- 1 拍。
        when 1=>    PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 2=>    PITCH<=3;           -- 音符 mi。
                   BEAT<=1;           -- 1 拍。
        when 3=>    PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 4=>    PITCH<=3;           -- 音符 mi。
                   BEAT<=2;           -- 2 拍。
        when 5=>    PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 6=>    PITCH<=4;           -- 音符 fa。
                   BEAT<=1;           -- 1 拍。
        when 7=>    PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 8=>    PITCH<=2;           -- 音符 re。
                   BEAT<=1;           -- 1 拍。
        when 9=>    PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 10=>   PITCH<=2;           -- 音符 re。
                   BEAT<=2;           -- 2 拍。
        when 11=>   PITCH<=0;           -- 間隔音。
                   BEAT<=0;
        when 12=>   PITCH<=1;           -- 音符 do。
                   BEAT<=1;           -- 1 拍。
        when 13=>   PITCH<=0;           -- 間隔音。
                   BEAT<=0;
    end case;
end process;

```



```

when 14=>  PITCH<=2;           -- 音符 re。
              BEAT<=1;           -- 1 拍。
when 15=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when 16=>  PITCH<=3;           -- 音符 mi。
              BEAT<=1;           -- 1 拍。
when 17=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when 18=>  PITCH<=4;           -- 音符 fa。
              BEAT<=1;           -- 1 拍。
when 19=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when 20=>  PITCH<=5;           -- 音符 so。
              BEAT<=1;           -- 1 拍。
when 21=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when 22=>  PITCH<=5;           -- 音符 so。
              BEAT<=1;           -- 1 拍。
when 23=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when 24=>  PITCH<=5;           -- 音符 so。
              BEAT<=2;           -- 2 拍。
when 25=>  PITCH<=0;           -- 間隔音。
              BEAT<=0;
when others=>  PITCH<=0;       -- 結束。
                  BEAT<=0;
end case;
end process;
-- (3) 多工電路
with FREQ select
    BEEP<= '0' when 0,           -- 靜音。
              PULSE when others; -- 輸出音符訊號。
end arch;

```

 練習

1. 使用 CPLD 晶片，設計播放如圖 15-12 所示小蜜蜂 (Little Bee) 音樂的數位電路。演奏速度為每分鐘 300 拍，每拍為 200ms。(music1.vhd)
2. 使用 CPLD 晶片，設計單一按鍵控制播放如圖 15-12 所示小蜜蜂 (Little Bee) 音樂及圖 15-14 所示小星星 (Little Star) 音樂的數位電路。按鍵切換靜音→小蜜蜂→小星星等三種變化。(music2.vhd)

小星星 (Little Star)																				
	1	1	5	5		6	6	5	-		4	4	3	3		2	2	1	-	
	5	5	4	4		3	3	2	-		5	5	4	4		3	3	2	-	
	1	1	5	5		6	6	5	-		4	4	3	3		2	2	1	-	

圖 15-14 小星星簡譜



充電站

1. **音樂 IC**：市面上有現成的音樂 IC 可供選擇，常用音樂 IC 編號 HT3810 系列是一顆歌曲產生器的單晶片，採用 CMOS 技術，只需少量的外接元件，即可產生美妙的音樂。HT3810 音樂 IC 的工作電壓範圍：2.4V~5.0V，為一 TO-92 包裝。如圖 15-15 所示音樂 IC 應用電路，當音樂 IC 連接直流電源後，其輸出可以直接驅動一顆壓電式蜂鳴器或喇叭，或者是透過 NPN 電晶體來驅動 8Ω 喇叭，產生內建音樂。使用 8051 的 P1.0 埠腳來控制，當 P1.0=1 時，喇叭輸出音樂 IC 的內建音樂，當 P1.0=0 時，喇叭沒有聲音。

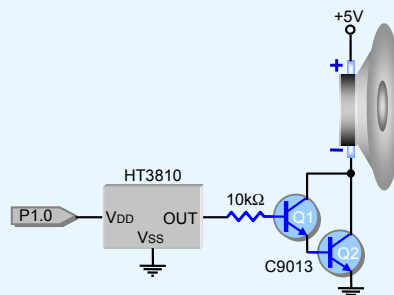


圖 15-15 音樂 IC 應用電路