Parking Lot Management System

He Zean [1] Li Zinan [1] Tang Xinyu[1]

[1]Dept. of Computer Science and Engineering, SUSTech

Dec. 23, 2021

Outline

SUSTech Southern University of Science and Technology

Outline

SUSTech Southern University of Science and Technology

Introduction

Outline

SUSTech Southern University of Science and Technology

Music Player: IO ports

```
module musicplayer(
            input clk,
            input music_sel,
            input music_en,
            output reg music_frac_ext = 0
        );
```

- clk: port Y18 (板载时钟, 100MHz)
- music_sel: controlled by the main module
- music_en: bgm on/off, controlled by a switch
- music_frac_ext: port A19 (蜂鸣器)

Music Player: How does it work?

Tuning

Frequencies for equal-tempered scale, $A_4 = 440$ Hz

Other tuning choices, A_4 =
432 434 436 438 440 442 444 446

Speed of Sound = 345 m/s = 1130 ft/s = 770 miles/hr
More about Speed of Sound

("Middle C" is C_4)

Note	Frequency (Hz)	Wavelength (cm)
C_0	16.35	2109.89
$C^{\#}_0/D^{b}_0$	17.32	1991.47
D_0	18.35	1879.69
$D^{\#}_0/E^{b}_0$	19.45	1774.20
E_0	20.60	1674.62
F_0	21.83	1580.63
$F^{\#}_0/G^{b}_0$	23.12	1491.91
G_0	24.50	1408.18
$G^{\#}_0/A^{b}_0$	25.96	1329.14
A_0	27.50	1254.55
$A^{\#}_0/B^{b}_0$	29.14	1184.13
B_0	30.87	1117.67
C_1	32.70	1054.94
$C^{\#}_1/D^{b}_1$	34.65	995.73
D_1	36.71	939.85

```
parameter stop = 0;

parameter do_lo = 382234;
parameter re_lo = 340530;
parameter me_lo = 303379;
parameter fa_lo = 286352;
parameter so_lo = 255102;
parameter la_lo = 227272;
parameter si_lo = 202478;

parameter do = 191110;
parameter re = 170265;
parameter me = 151685;
parameter fa = 143172;
parameter so = 127551;
parameter la = 113636;
parameter si = 101239;

parameter do_hi = 95557;
parameter re_hi = 85131;
parameter me_hi = 75844;
parameter fa_hi = 71586;
parameter so_hi = 63776;
parameter la_hi = 56818;
parameter si_hi = 50619;
```

Retrieved from https://pages.mtu.edu/%7Esuits/notefreqs.html

$$cnt_{do} \times \frac{1_{sec}}{clk} = \frac{1_{sec}}{freq_{C_4}} \qquad \text{Parameter } do \approx cnt_{do}$$

SUSTech Southern University of Science and Technology

Music Player: How does it work?

```
reg [18:0] freq_cnt;

always @ (posedge clk)begin
    if(music_en)begin
        if (freq_cnt >= freq)begin
            freq_cnt = 0;
            music_frac_ext = ~music_frac_ext;
        end
        else
            freq_cnt = freq_cnt + 1;
    end
end
```

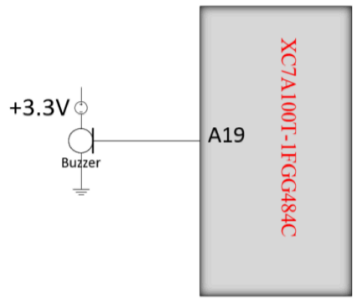

图 1-10 蜂鸣器连接电路图

除了上述的各种视觉输出部件，Minisys 实验板上还配置了一个蜂鸣器用作声音输出部件。与主芯片的连接方式如图 1-10 所示。主芯片通过 A19 管脚向蜂鸣器输出一个电信号，该信号的频率由用户决定。在该信号驱动下，蜂鸣器内部发生机械振动，发出相应频率的声音。

Music Player: Encoding music

```
always @ (paisp)begin
    if (!music_en)
        freq = stop;
    else begin
        case(song_midi[paisp * 5 +:5])
            'd1    : freq = do_lo;
            // other 20 cases of tones
            default: freq = stop;
        endcase
    end
end
```

- 22 tones: $C_3 \sim B_3$ (lo), $C_4 \sim B_4$ ("Middle C" is C_4), $C_5 \sim B_5$ (hi), plus a *stop*.
- $\lceil \log_2 22 \rceil = 5$, thus in our "musical score", 5 bits are used to represent one beat.

```
parameter SONG_1_LEN = 93;
parameter SONG_1_MON = 'b000110101001010100010101010001010011000101000100010100010001001111000100110100010010110001001101000

parameter SONG_2_LEN = 76;
parameter SONG_2_MON = 'b001100100001001010010000001001010100101101011000000101101100010100101000000010010100001000010010000
```

Music Player: Switching beats

```
always @ (posedge clk)begin
    if (music_en)begin
        if(paicg >= pai_gap)begin
            paicg <= 0;
            if (paisp == 0)begin
                if(music_rep)
                    paisp = song_len;
            end
            else
                paisp = paisp - 1;
        end
        else begin
            paicg = paicg + 1;
        end
    end
end
```

Outline

SUSTech Southern University of Science and Technology

Demonstration

- dfs
- bfs

Outline

SUSTech Southern University of Science and Technology