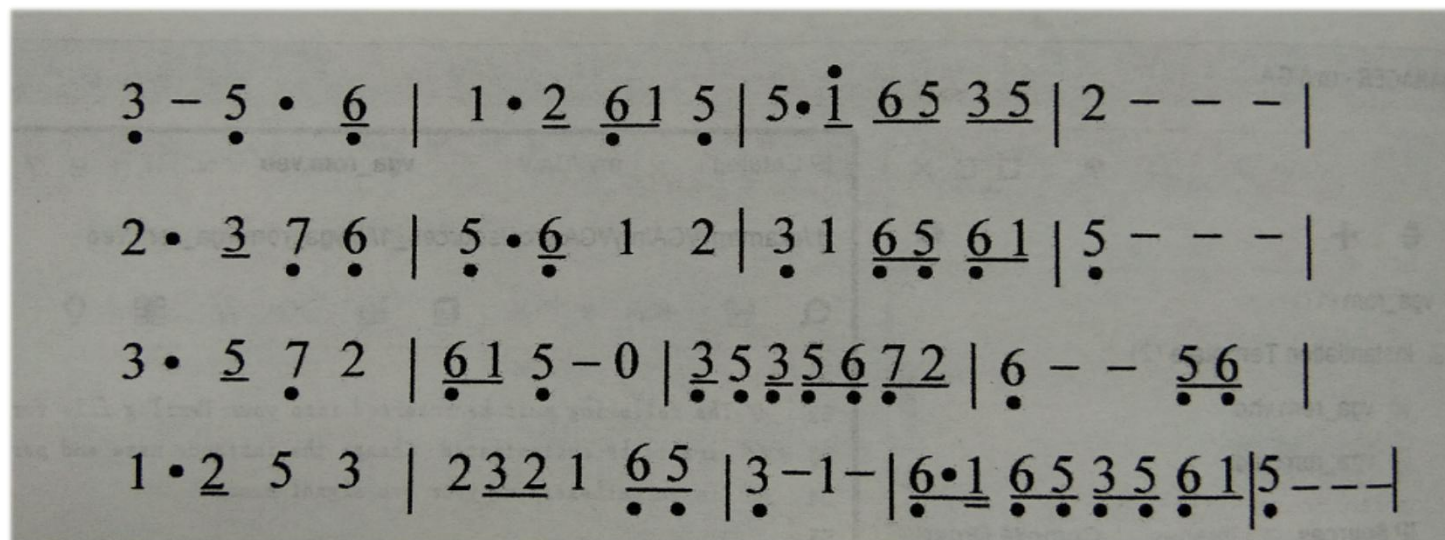


音乐播放器的实现

实现原理：组成乐曲的每个音符的频率值（音调）及其持续的时间（音长）是乐曲能连续演奏所需的两个基本数据，因此，只要控制输出到扬声器的激励信号的频率的高低和持续时间，就可以使扬声器发出连续的乐曲声。

本次实验采用FPGA器件驱动扬声器构成一个乐曲演奏电路，演奏的乐曲选择“梁祝”片段，其曲谱如下：



音乐播放器的实现

1.音调的控制

频率的高低决定了音调的高低。音乐的十二平均率规定：每两个八度音（如简谱中的中音1和高音1）之间的频率相差一倍。在两个八度音之间，又可以分为12个半音，每两个半音的频率比为 $\sqrt[12]{2}$ 。另外，音名A（简谱中的低音6）的频率为440Hz，音名B到C之间、E到F之间为半音，其余为全音。由此可以计算出简谱中从低音1到高音1之间每个音名对应的频率如表所示：

音名	频率/Hz	音名	频率/Hz	音名	频率/Hz
低音1	261.6	中音1	523.3	高音1	1046.5
低音2	293.7	中音2	587.3	高音2	1174.7
低音3	329.6	中音3	659.3	高音3	1319.5
低音4	349.2	中音4	699.5	高音4	1396.9
低音5	392	中音5	784	高音5	1568
低音6	440	中音6	880	高音6	1760
低音7	493.9	中音7	987.8	高音7	1975.5

音乐播放器的实现

1.音调的控制

所有不同频率的信号都是从同一个基准频率分配得到的。由于音阶频率多为非整数，而分频系数又不能为小数，故必须将计算得到的分频四舍五入取整。若基准频率过低，则由于分频比太小，四舍五入取整后的误差较大；若基准频率过高，虽然误差变小，但分频数将变大。实际的设计综合考虑这两方面的因素，在尽量减小频率误差的前提下取合适的基准频率。本例选取6MHz为基准频率，若无6MHz的时钟频率，则可以先分频得到6MHz（或近似6MHz），或者换一个新的基准频率。实际上，只要各个音名间的相对频率关系不变，C作1与D作1演奏出的音乐听起来都不会太走调。

为了减少输出的偶次谐波分量，最后输出到扬声器的波形应为对称方波，因此在达到扬声器之前，有一个二分频的分配器。下表的分频比就是从6MHz频率分频得到的3MHz频率基础上计算出来的。

音乐播放器的实现

1.音调的控制

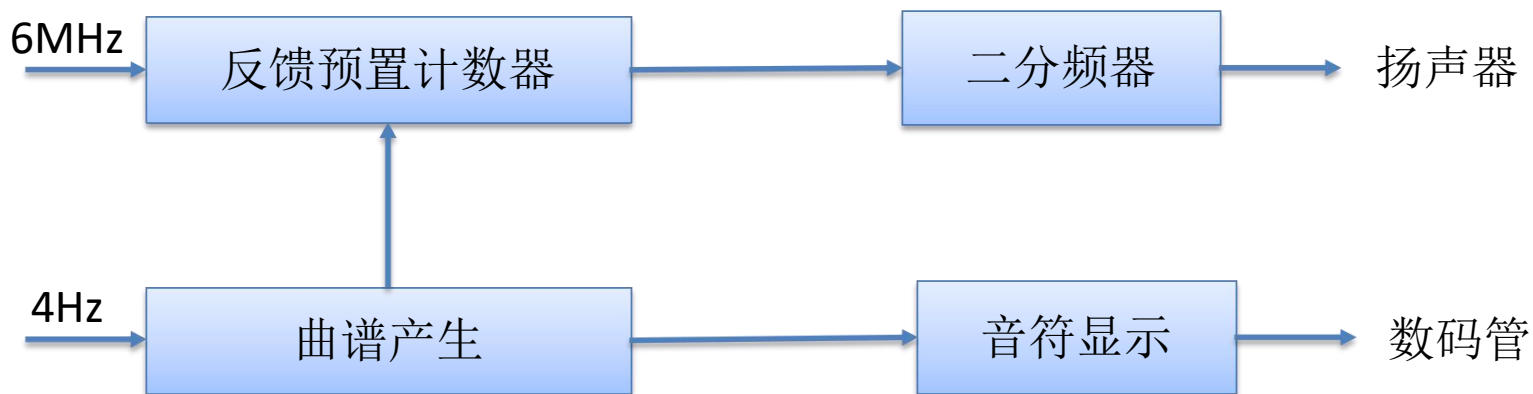
本例需要演奏的是“梁祝”乐曲，该乐曲各音阶频率对应的分频比及预置数如下所示：

音名	分频比	预置数	音名	分频比	预置数
低音1	11468	4915	中音5	3827	12556
低音2	10215	6168	中音6	3409	12974
低音3	9102	7281	中音7	3037	13346
低音4	8591	7792	高音1	2867	13516
低音5	7653	8730	高音2	2554	13829
低音6	6818	9565	高音3	2274	14109
低音7	6073	10310	高音4	2148	14325
中音1	5736	10647	高音5	1913	14470
中音2	5111	11272	高音6	1705	14678
中音3	4552	11831	高音7	1519	14864
中音4	4289	12094	休止符	0	16383

音乐播放器的实现

2.音长的控制

音符的持续时间根据乐曲的速度及每个音符的节拍数来确定。本例演奏的“梁祝”片段，最短的音符为四分音符，如果将全音符的持续时间设为1s，则只需要提供一个4Hz的时钟频率即可产生四分音符的时长。



乐曲演奏电路的原理框图

音乐播放器的实现

2.音长的控制

乐谱产生电路用来控制音乐的音调和音长。控制音调通过设置计数器的预置数来实现，预置不同的数值就可以使计数器产生不同频率的信号，从而产生不同的音调。控制音长是通过控制计数器预置的停留时间来实现的，预置数停留的时间越长，该音符演奏的时间就越长。每个音符的演奏时间都是0.25s的整数倍，对于节拍较长的音符，如二分音符，在记谱时将该音名连续记录两次即可。

音符显示电路用来显示乐曲演奏时对应的音符。可用数码管显示音符，实现演奏的动态显示。在本例中，HiGH[3:0]、MED[3:0]、LOW[3:0]等信号分别用于显示高音、中音和低音音符。为了使演奏能循环进行，需要另外设置一个时长计数器，当乐曲演奏完成时，保证能自动从头开始演奏。

音乐播放器部分引脚约束代码

```
set_property PACKAGE_PIN T1 [get_ports speaker]  
set_property PACKAGE_PIN P17 [get_ports sys_clk]
```