

VLSI DSP Chapter2 Homework

171180545 王静之

为了使用 LPM 方法求解迭代边界(iteration bound), 我根据 LPM 算法原理编写了一个 python 小程序, 并使用书上例题进行了验证, 代码如下:

```
lpm.py > ...
1  import numpy as np
2  #171180545 wjz
3  num_delay = 4    #延迟的数量
4  L = np.zeros((num_delay,num_delay,num_delay))
5  L[:, :, 0] = np.array([[-1,0,-1,-1],[4,-1,0,-1],[5,-1,-1,0],[5,-1,-1,-1]]) #观察获得L(1)
6  K_forselect = range(1,num_delay+1)
7
8  for d in range(1,num_delay):
9      for i in range(0,num_delay):
10         for j in range(0,num_delay):
11             max_list = [-1]
12             K_temp = list(K_forselect)
13             for k in K_forselect:
14                 if (L[i,k-1,0]==-1 or L[k-1,j,d-1]==-1):
15                     K_temp.remove(k)
16             for k in K_temp:
17                 max_list.append(L[i,k-1,0]+L[k-1,j,d-1])
18             L[i,j,d] = max(max_list)
19
20  bound_list = []
21  for d in range(0,num_delay):
22      for i in range(0,num_delay):
23          if (L[i,i,d] != -1):
24              bound_list.append(L[i,i,d]/(d+1))
25  iter_bound = max(bound_list)
26  print("iteration bound is",iter_bound)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Python

iteration bound is 2.0

PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> █

其中 L 是一个三维矩阵储存了 d 个平面矩阵的运算结果
以下三题均是在此程序中计算得到。

3. 图中加法和乘法分别需要 1 u.t. 和 2 u.t. 的运算时间。

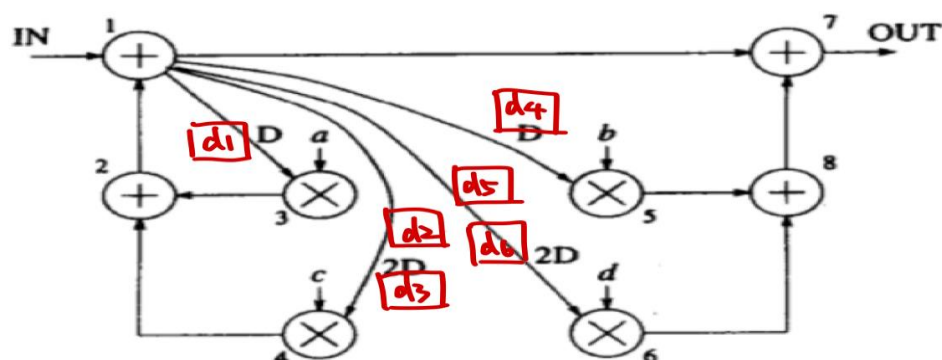


图2-14 biquad滤波器

根据定义，观察到有 6 个延迟单元，编号后得到从 d_i 到 d_j 只有 0 个延迟单元的 $L(1)$:

```
[ 4  4 -1  4  4 -1]
[-1 -1  0 -1 -1 -1]
[ 4  4 -1  4  4 -1]
[-1 -1 -1 -1 -1 -1]
[-1 -1 -1 -1 -1  0]
[-1 -1 -1 -1 -1 -1]
```

将 $L(1)$ 带入到程序中，得到计算结果：

```
26 print([round(i,2) for i in bound_list])
27 print("iteration bound is",iter_bound)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

iteration bound is 4.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> & C:/Users/10595/Anaconda3/pyt
/大三/大三下/VLSIDSP/LPM/lpm.py
[4.0, 4.0, 2.0, 2.0, 4.0, 2.67, 2.67, 4.0, 3.0, 3.0, 4.0, 3.2, 3.2, 4.0, 3.33, 3.33]
iteration bound is 4.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM>

故本题迭代边界 $T_{\omega} = 4 u.t.$

4. 图中加法和乘法分别需要 1 u.t. 和 2 u.t. 的运算时间。

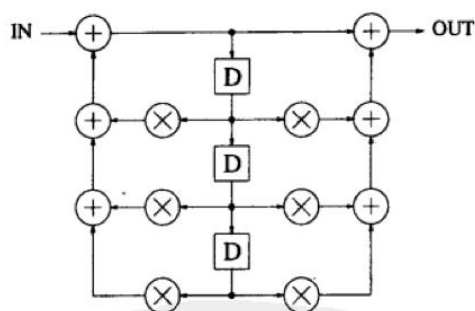


图2-15 3阶直接形式IIR滤波器

图中共有 3 个延迟单元，从上至下编号为 d_1, d_2, d_3 ，则观察可得 $L(1)$:

```
[ 4  0 -1 ]
[ 5 -1  0 ]
[ 5 -1 -1 ]
```

将其代入程序中：

得本题迭代边界 $T_{\omega} = 4 u.t.$

```

24         bound_list.append(L[i,i,d]/(d+1))
25     iter_bound = max(bound_list)
26     print([round(i,2) for i in bound_list])
27     print("iteration bound is",iter_bound)

```

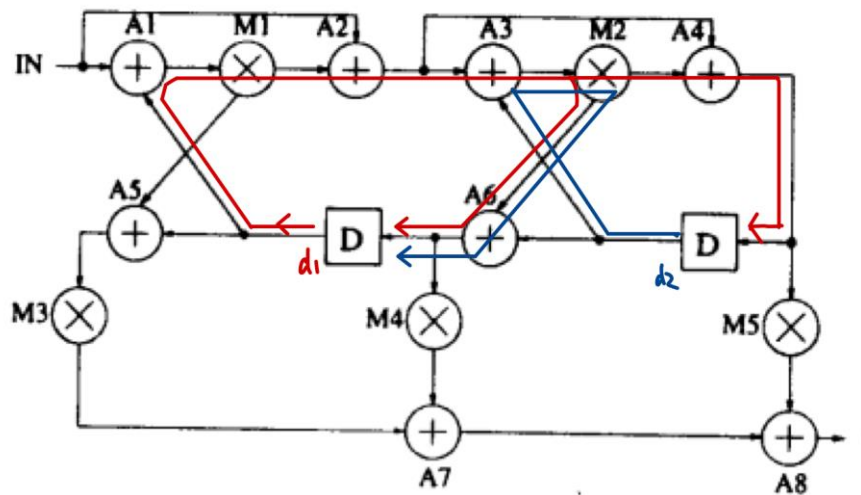
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

iteration bound is 4.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> & C:/Users/10595/Desktop/学习/大三/大三下/VLSIDSP/LPM/lpm.py
[4.0, 4.0, 2.5, 4.0, 3.0, 1.67]
iteration bound is 4.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> 

```

6. 图中加法和乘法分别需要 1 u.t. 和 2 u.t. 的运算时间。



图中共有 2 个延时单元， $L(1)$ 为：

$$[8 \ 8]$$

$$[4 \ 4]$$

计算 $L(2)$ 为：

$$[16 \ 16]$$

$$[12 \ 12]$$

故本题迭代边界为 $T_{\infty} = \max\left\{\frac{8}{1}, \frac{4}{1}, \frac{16}{2}, \frac{12}{2}\right\} = 8 \text{ u.t.}$

用代码验证一下，结果相同。

```

25     iter_bound = max(bound_list)
26     print([round(i,2) for i in bound_list])

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

iteration bound is 4.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> & C:/Users/10595/Desktop/学习/大三/大三下/VLSIDSP/LPM/lpm.py
[8.0, 4.0, 8.0, 6.0]
iteration bound is 8.0
PS C:\Users\10595\Desktop\学习\大三\大三下\VLSIDSP\LPM> 

```