

The background features five light purple circles arranged in two rows. The top row has three circles, and the bottom row has two circles. The text is overlaid on these circles.

# 第12章

# DSP Builder设计深入

# 12.1 FIR数字滤波器设计

## 12.1.1 FIR滤波器原理

$$H(z) = \sum_{k=0}^M b_k z^{-k} \quad (12-1)$$

$$y(n) = \sum_{i=0}^{L-1} x(n-i)h(i) \quad (12-2)$$

$$y(n) = x(n) * h(n) \quad (12-3)$$

# 12.1 FIR数字滤波器设计

## 12.1.1 FIR滤波器原理

$$h(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + h(3)x(n-3) \quad (12-4)$$

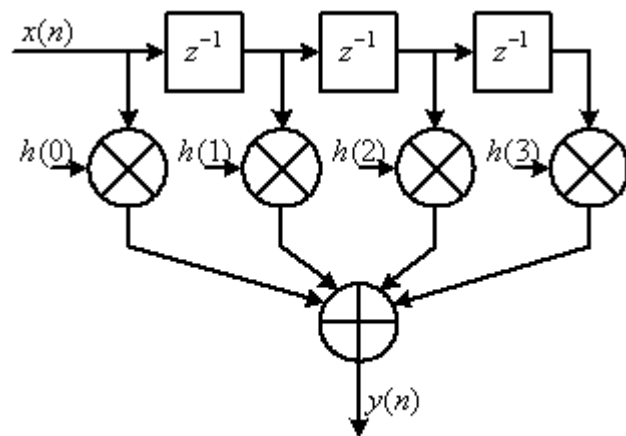


图 12-1 3阶 FIR 滤波器结构

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 1. 3阶常数系数FIR滤波器设计

$$h(n) = C_q (h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + h(3)x(n-3)) \quad (12-5)$$

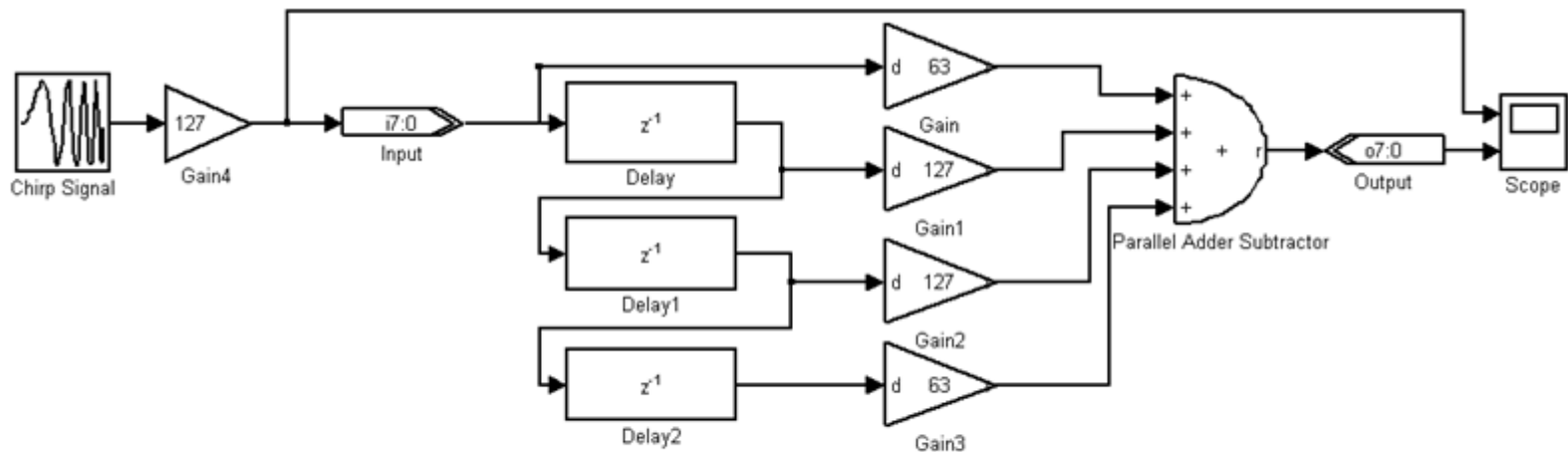


图 12-2 带有仿真信号模块的 3 阶滤波器模型

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 1. 3阶常数系数FIR滤波器设计

Chirp Signal模块：（Chirp Signal）

库：Simulink中Sources库

参数“Initial Frequency (Hz)”设为“0.1”；“Target time”设为“50”

参数“Frequency at target time (Hz)”设为“1”

参数“Interpret vectors parameters as 1-D”选中

Gain4模块：（Gain）

库：Simulink中Math Operations库

参数“Gain”设为“127”；“Multiplication”设为“Element wise (K.\*u)”

Scope模块：（Scope）

库：simulink中sinks库

参数“Number of Axes”为“2”

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 1. 3阶常数系数FIR滤波器设计

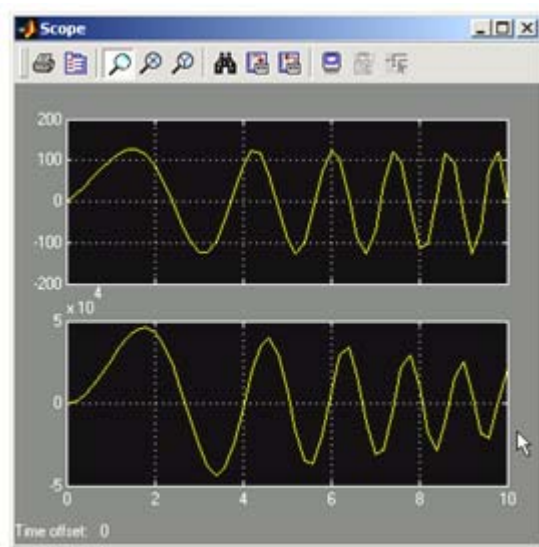


图 12-3 FIR 滤波器仿真结果

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 2. 4阶FIR滤波器节设计

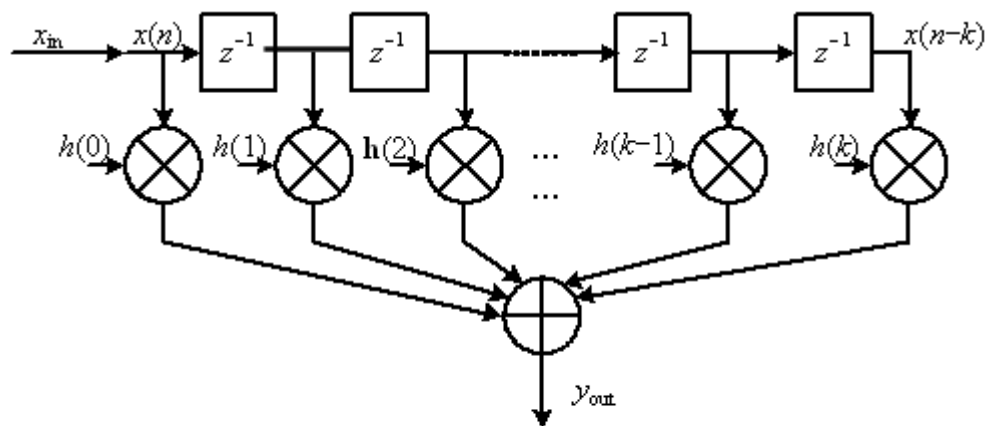


图 12-4 直接 I 型 FIR 滤波器结构

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 2. 4阶FIR滤波器节设计

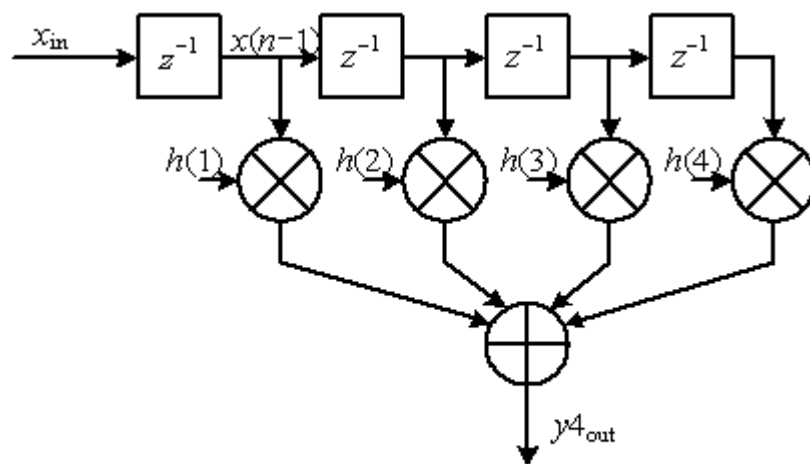


图 12-5 直接 I 型 4 阶 FIR 滤波器节

$$H(z) = h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} \quad (12-6)$$



# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 2. 4阶FIR滤波器节设计

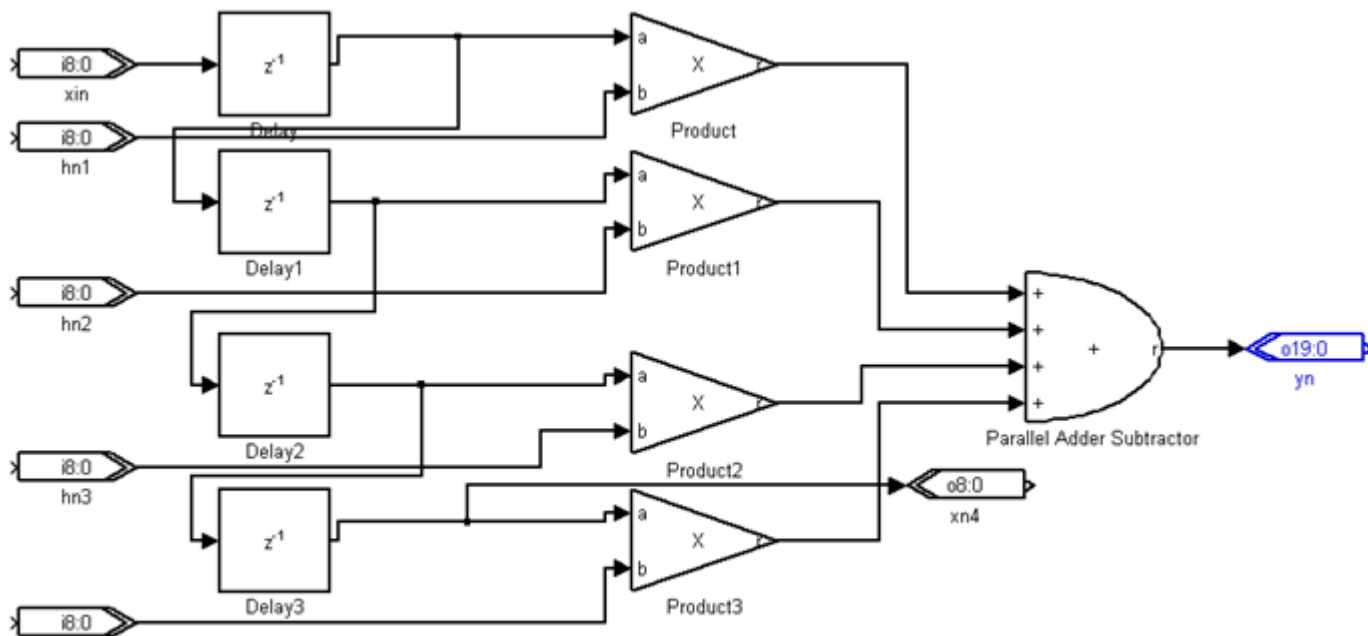


图 12-6 直接 I 型 4 阶 FIR 滤波器节

# 12.1 FIR数字滤波器设计

Xin、hn1、hn2、hn3、hn4模块：（Altbus）

库：Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”；“number of bits”设为“9”

yn模块：（Altbus）

库：Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”；“number of bits”设为“20”

xn4模块：（Altbus）

库：Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”；“number of bits”设为“9”

Parallel Adder Subtractor模块：（Parallel Adder Subtractor）

库：Altera DSP Builder中Arithmetic库

参数“Number of Inputs”设为“4”；“Add(+)Sub(-)”设为“++++”

使用“Pipeline”

参数“Clock Phase Selection”设为“1”

Delay、Delay1、Delay2、Delay3模块：（Delay）

库：Altera DSP Builder中Storage库

参数“Depth”设为“1”；“Clock Phase Selection”设为“1”

Product、Product1、Product2、Product3模块：（Product）

库：Altera DSP Builder中Arithmetic库

参数“Pipeline”设为“2”；“Clock Phase Selection”设为“1”；不选择“Use LPM”

# 12.1 FIR数字滤波器设计

## 12.1.2 使用DSP Builder设计FIR滤波器

### 3. 16阶FIR滤波器模型设计

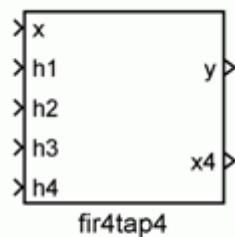


图 12-7 fir4tap 子系统

# 12.1 FIR数字滤波器设计

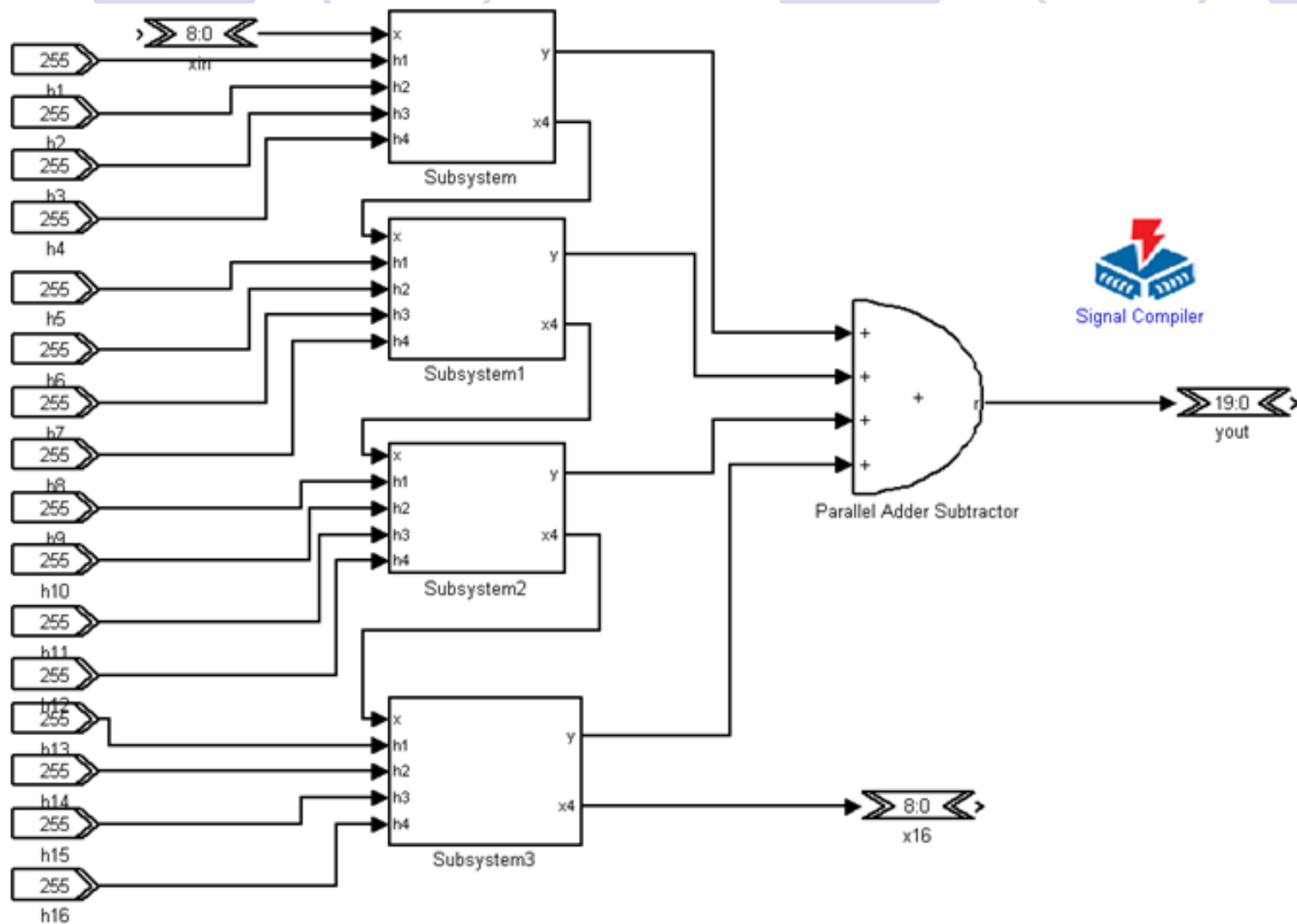


图 12-8 16 阶直接 I 型 FIR 滤波器模型

# 12.1 FIR数字滤波器设计

xin模块: (Altbus)

库: Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”; “number of bits”设为“9”

yout模块: (Altbus)

库: Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”; “number of bits”设为“20”

x16模块: (Altbus)

库: Altera DSP Builder中IO & Bus库

参数“Bus Type”设为“signed Integer”; “number of bits”设为“9”

Parallel Adder Subtractor模块: (Parallel Adder Subtractor)

库: Altera DSP Builder中Arithmetic库

参数“Number of Inputs”设为“4”; “Add(+) Sub(-)”设为“+++”; 使用“Pipeline”

参数“Clock Phase Selection”设为“1”

h1、h2、h3、h4、h5、h6、h7、h8、h9、

h10、h11、h12、h13、h14、h15、h16模块: (Constant)

库: Altera DSP Builder中IO & Bus库

参数“[Number of Bits].[]”设为“9”; “Constant Value”设为“255”

参数“Bus Type”设为“Signed Integer”

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 1. 打开MATLAB的FDATool

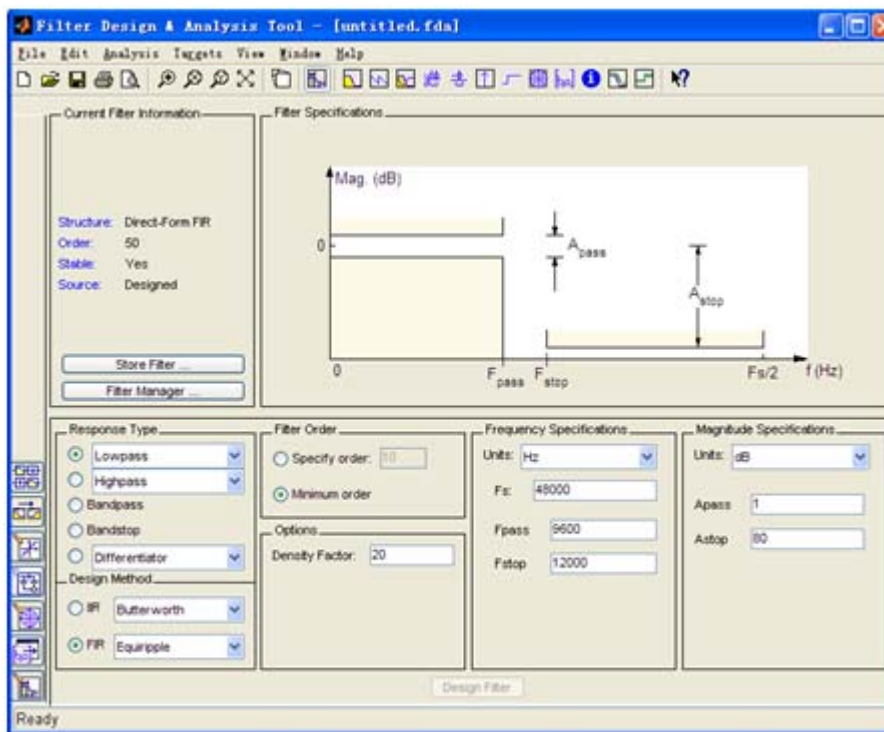


图 12-9 FDATool 界面

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 2. 选择Design Filter

$$H(z) = \sum_{k=1}^{16} b_k z^{-k} \quad (12-7)$$

$$H(z) = z^{-1} \sum_{k=0}^{15} b_k z^{-k} \quad (12-8)$$

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 3. 滤波器分析

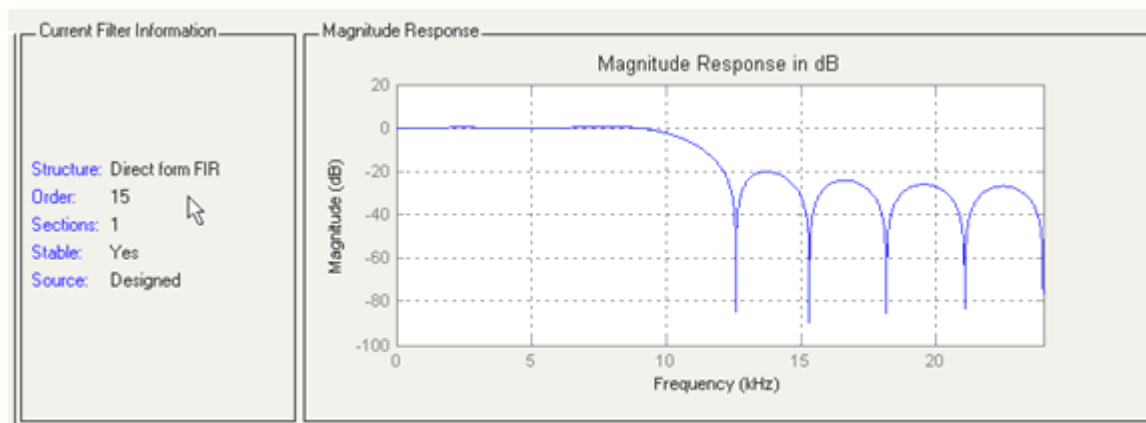


图 12-10 FIR 滤波器的幅频响应



# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 3. 滤波器分析

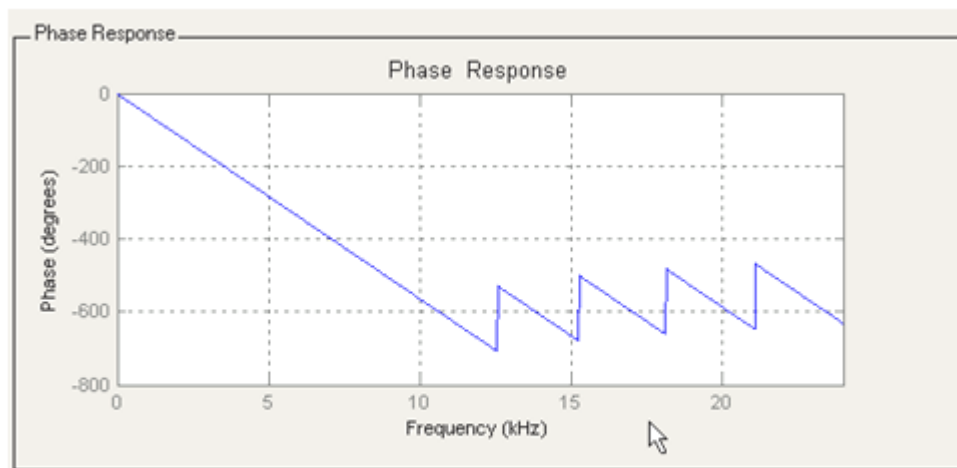


图 12-11 FIR 滤波器的相频响应

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 3. 滤波器分析

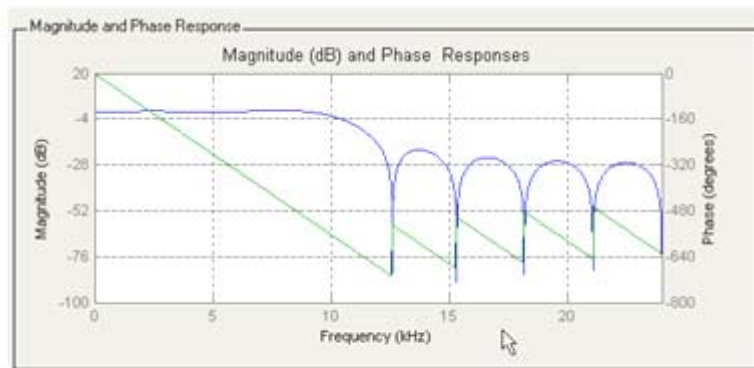


图 12-12 幅频响应与相频响应比较

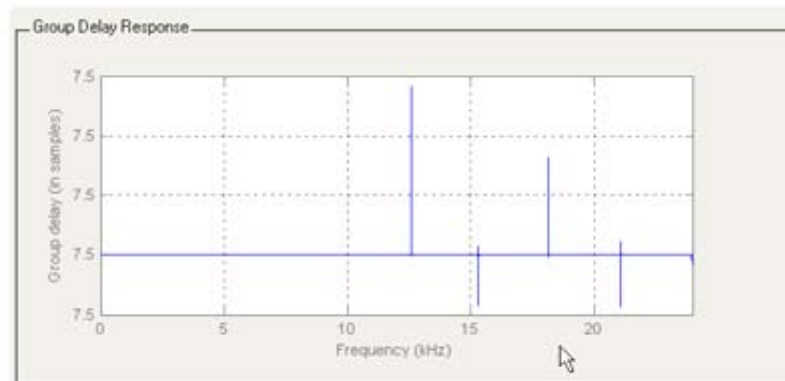


图 12-13 FIR 滤波器的群延时

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 3. 滤波器分析

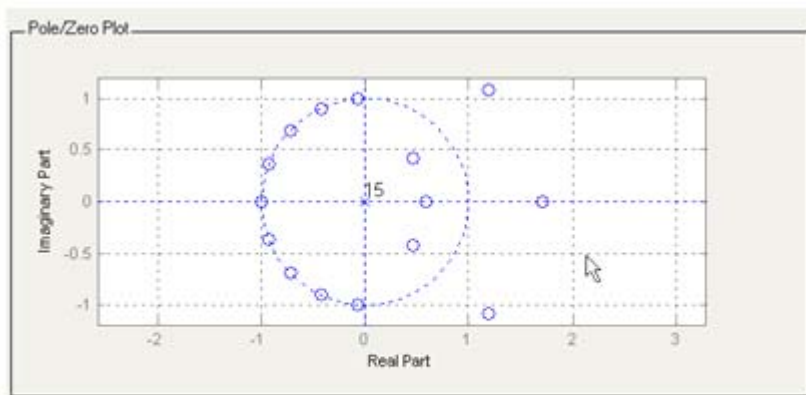


图 12-14 FIR 滤波器的零极点

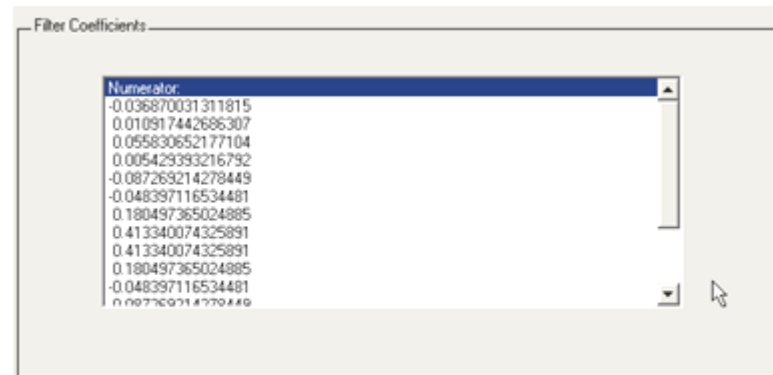


图 12-15 FIR 滤波器系数

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 4. 量化

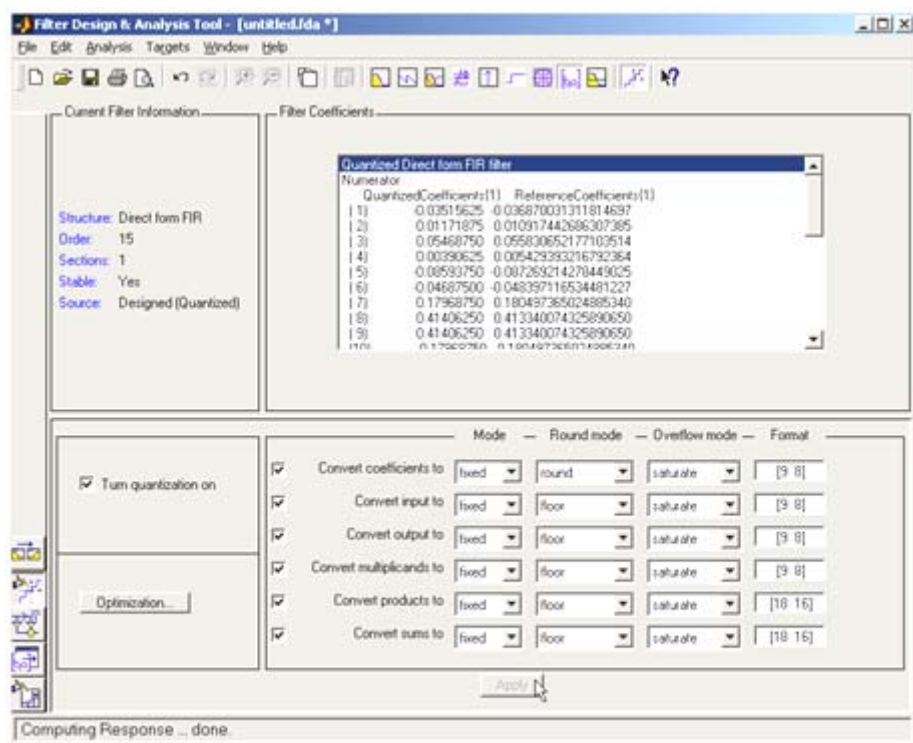


图 12-16 量化参数设置

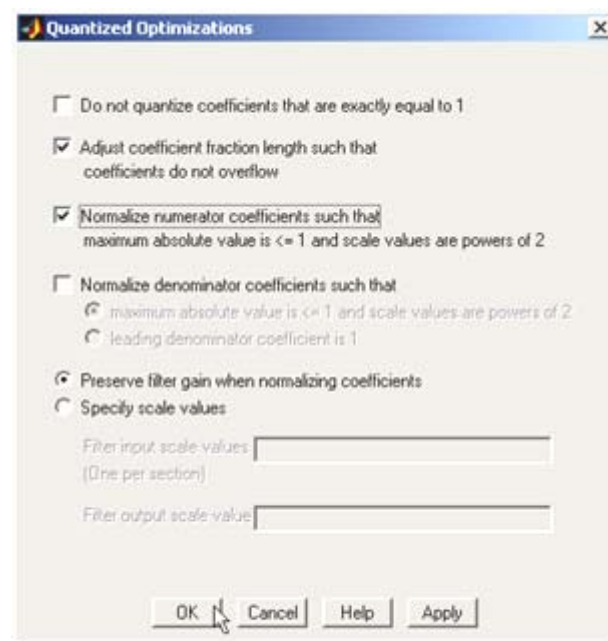


图 12-17 量化优化设置

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 4. 量化

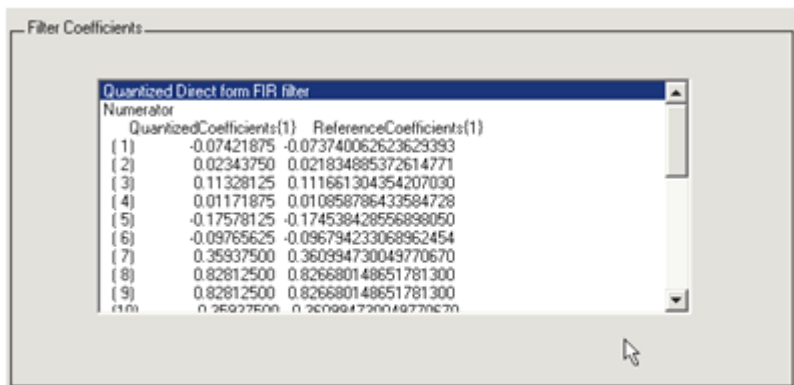


图 12-18 量化后系数

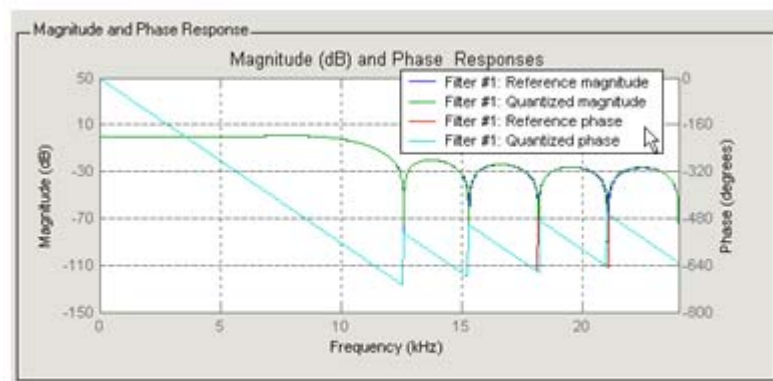


图 12-19 量化后幅频、相频响应

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 4. 量化

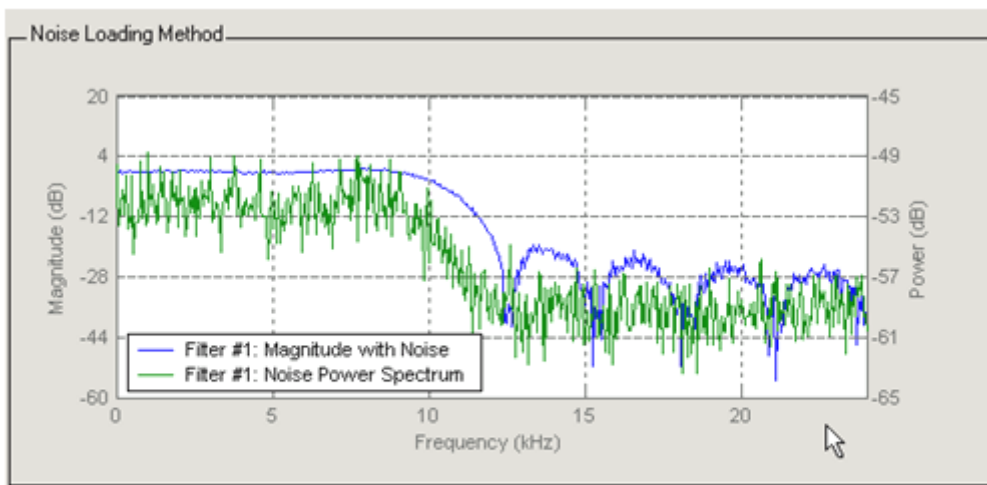


图 12-20 量化后噪声分析

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 5. 导出滤波器系数

```
Num =  
-0.0742    0.0234    0.1133    0.0117   -0.1758   -0.0977  
 0.3594    0.8281    0.8281    0.3594   -0.0977   -0.1758  
 0.0117    0.1133    0.0234   -0.0742
```

```
>> Num*(2^8)  
ans =  
Columns 1 through 10  
-19    6   29    3  -45  -25   92  212  212   92  
Columns 11 through 16  
-25  -45    3   29    6  -19
```

# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

### 5. 导出滤波器系数

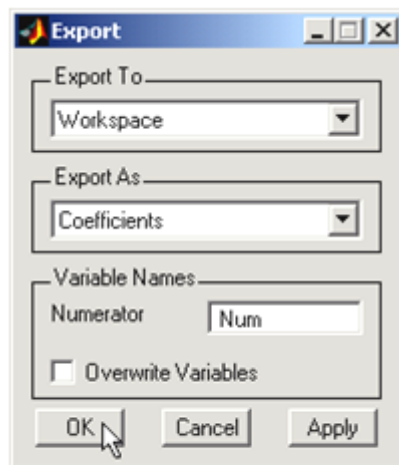


图 12-21 导出对话框



# 12.1 FIR数字滤波器设计

## 12.1.3 使用MATLAB的滤波器设计工具

6. 修改FIR滤波器模型添加参数

7. 导出滤波器系数的另一种方法

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

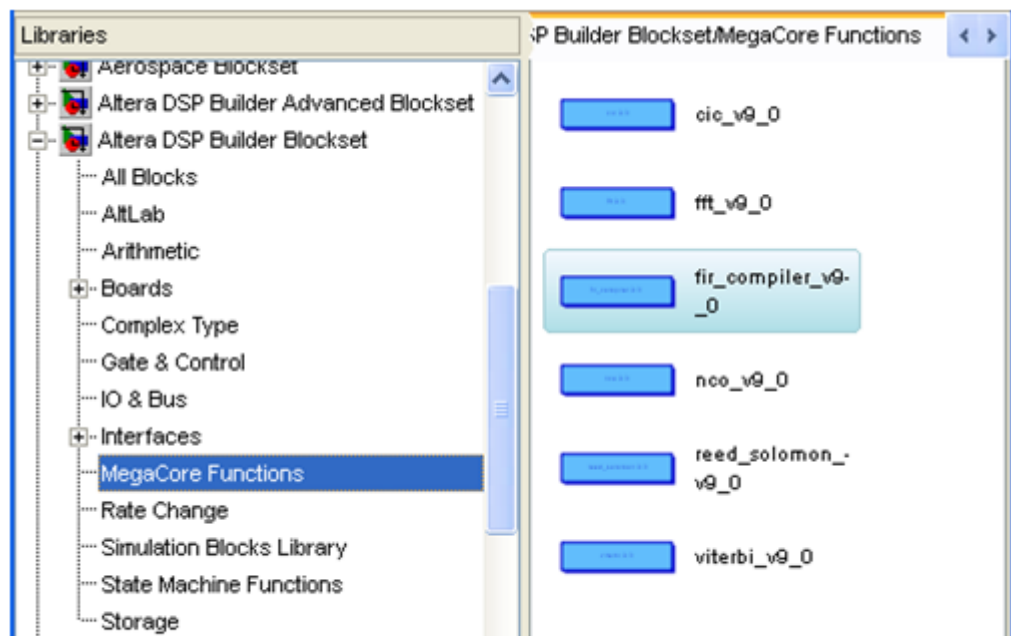


图 12-22 IP Core 模块库

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

1. FIR滤波器核的使用
2. 配置FIR滤波器器核



图 12-23 设置 FIR Core 参数

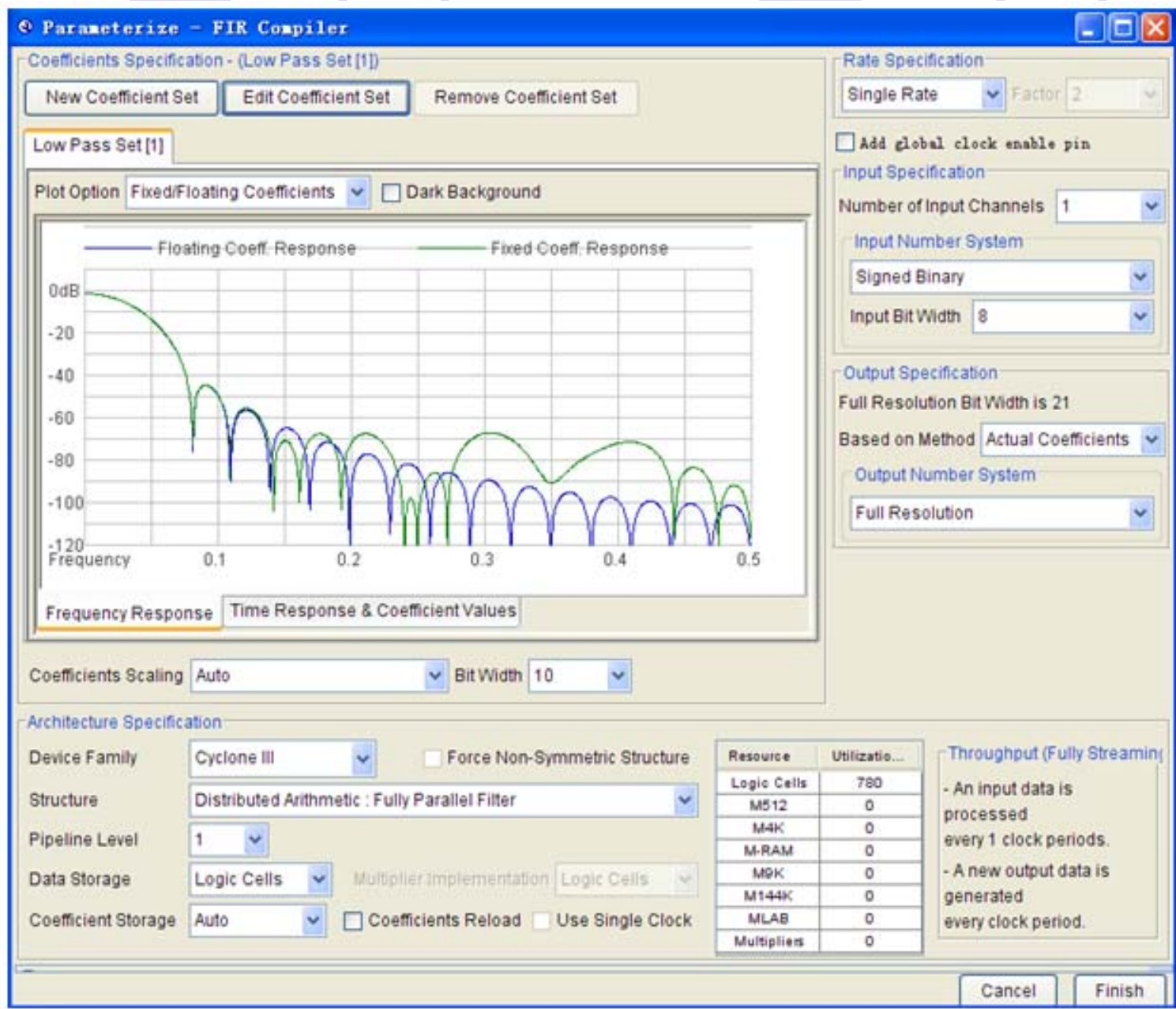


图 12-24 确定 FIR 滤波器系数

# 12.1 FIR数字滤波器设计

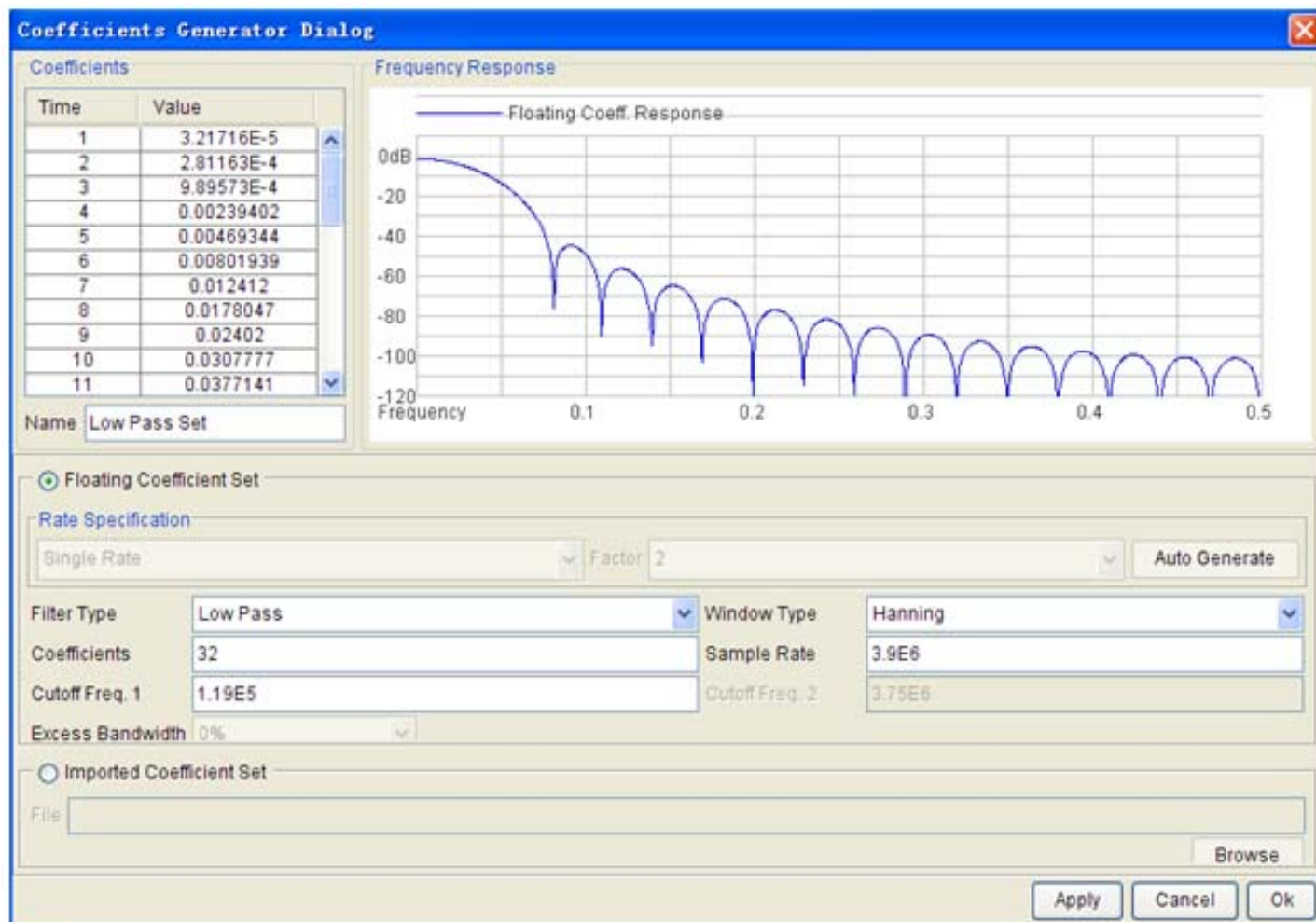


图 12-25 确定 FIR 工作方式

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

### 2. 配置FIR滤波器器核

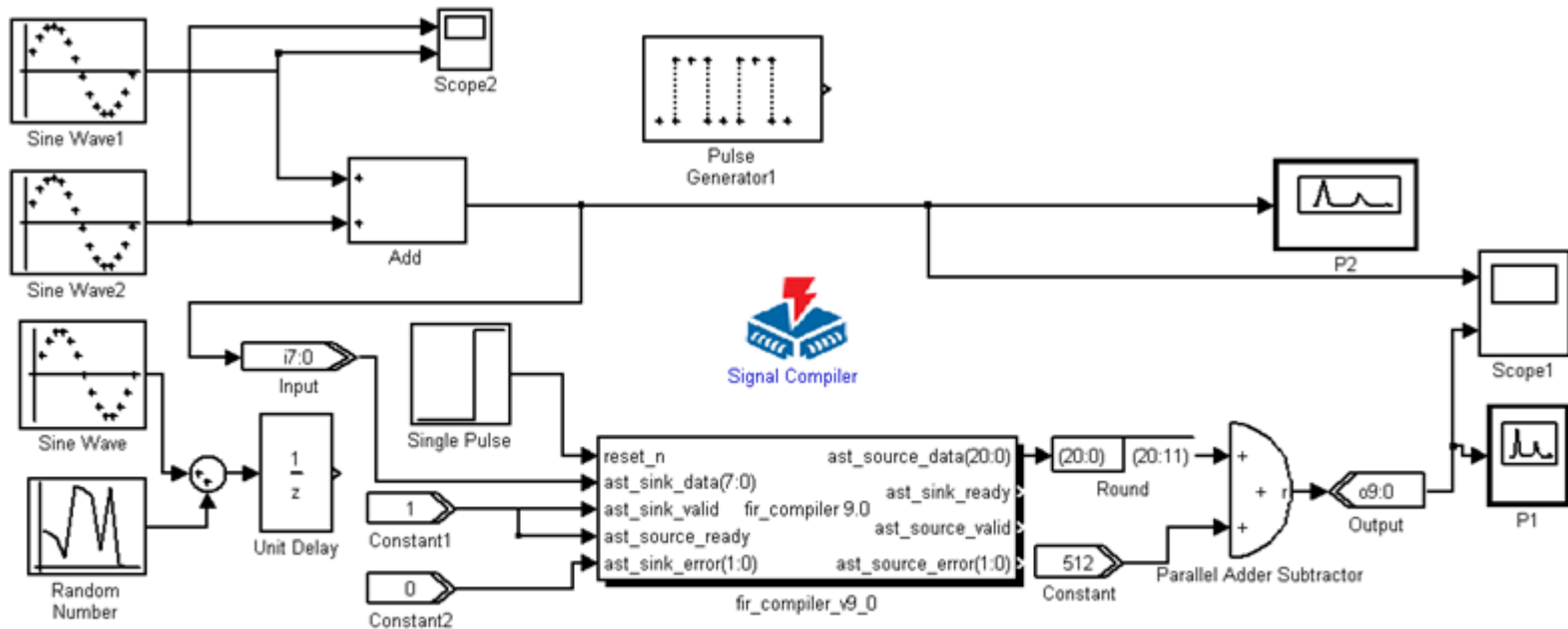


图 12-26 FIR 滤波器核的测试电路模型

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

### 2. 配置FIR滤波器器核

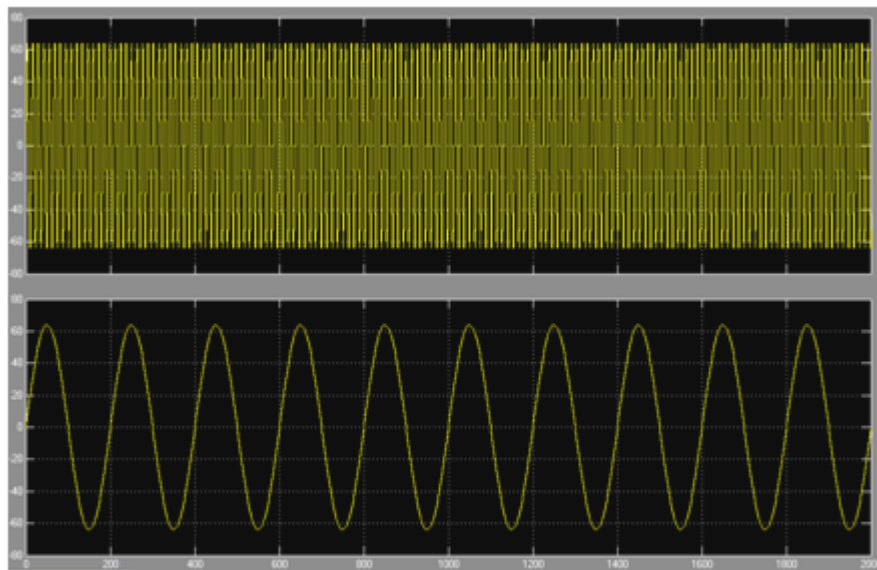


图 12-27 Scope2 显示波形

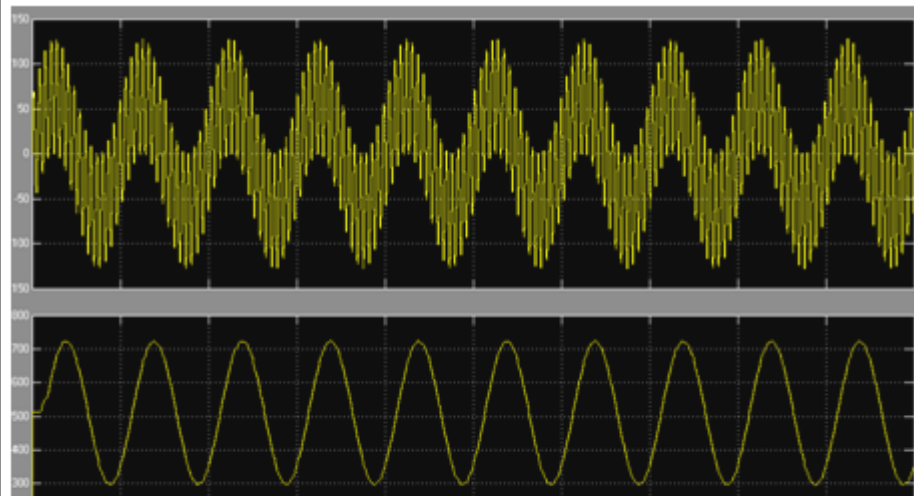


图 12-28 Scope1 显示波形

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

### 2. 配置FIR滤波器器核

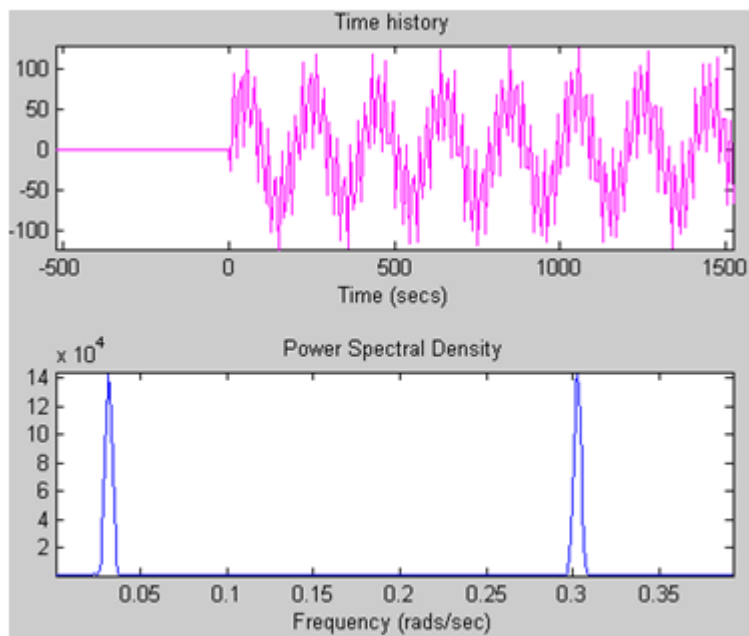


图 12-29 P2 频谱仪显示波形

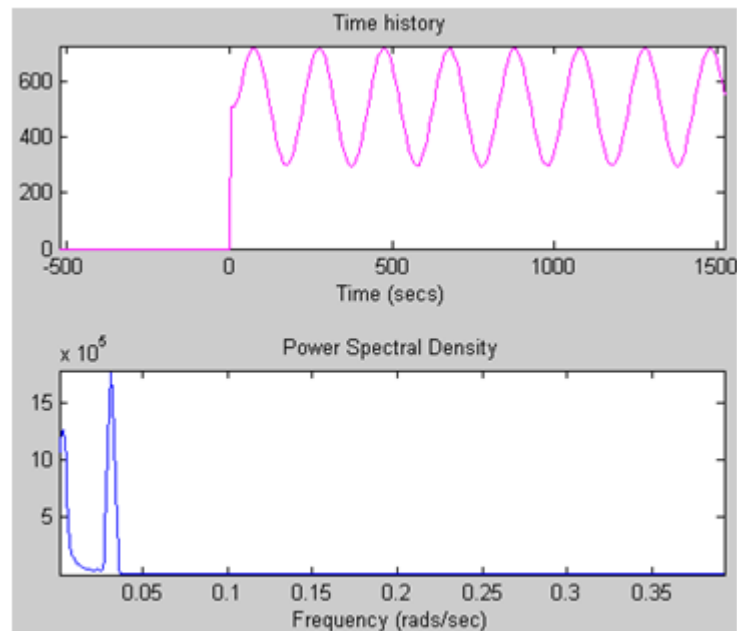


图 12-30 P1 频谱仪显示波形



# 12.2 HDL模块插入仿真与设计

## 1. 完成Verilog HDL设计

### 【例 12-1】

```
module MUX21a (A,B,S,Y);  
    input S ; input [7:0]A,B;    output [7:0]Y;  
    reg [7:0] Y;  
    always @(A,B,S) begin  
        if (S==0) Y = A; else Y = B;  
    end  
endmodule
```

# 12.2 HDL模块插入仿真与设计

## 2. 调入HDL Import模块

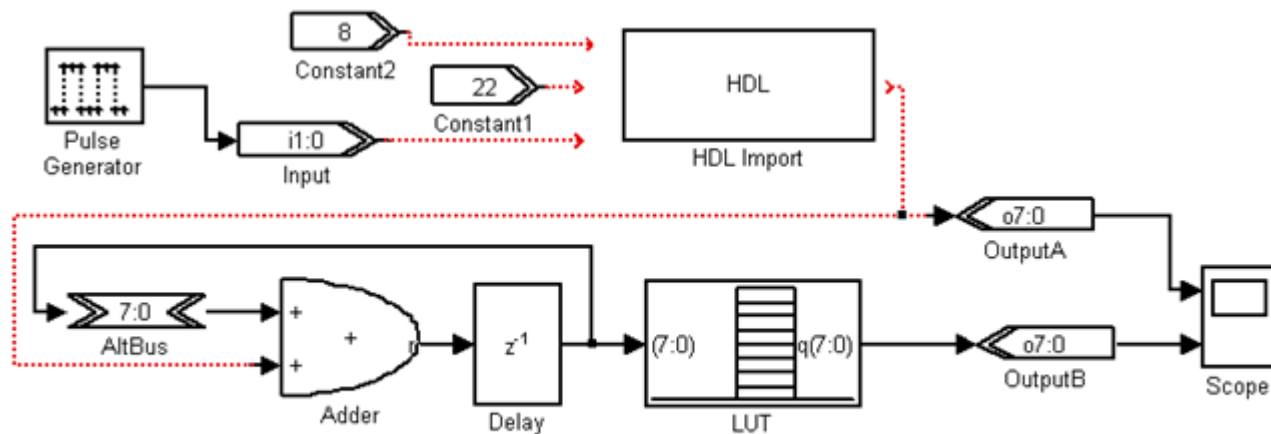


图 12-31 在 FSK 的 Simulink 模型中调入一个 HDL Import 模块

# 12.2 HDL模块插入仿真与设计

## 3. 加入Verilog设计文件

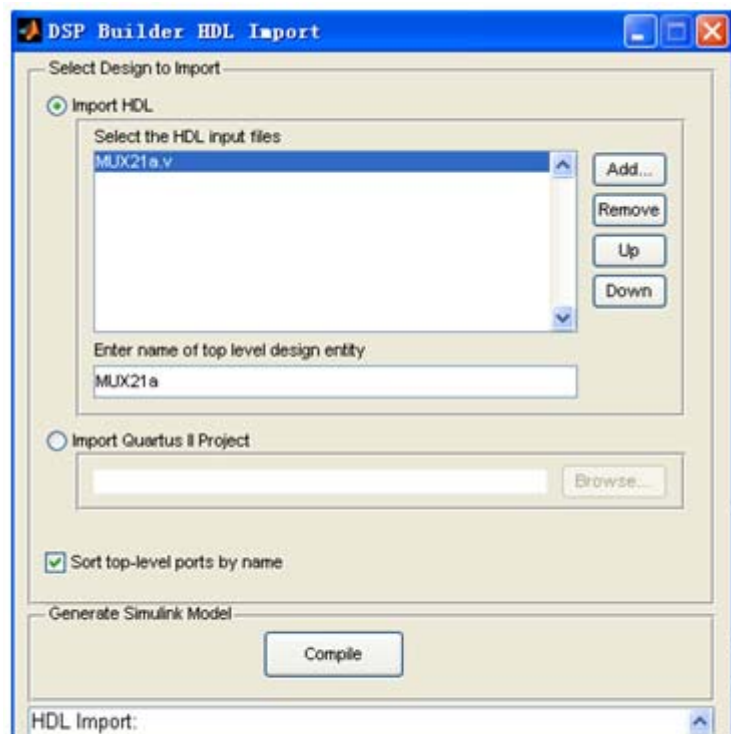


图 12-32 浏览到 3 个 VHDL 文件

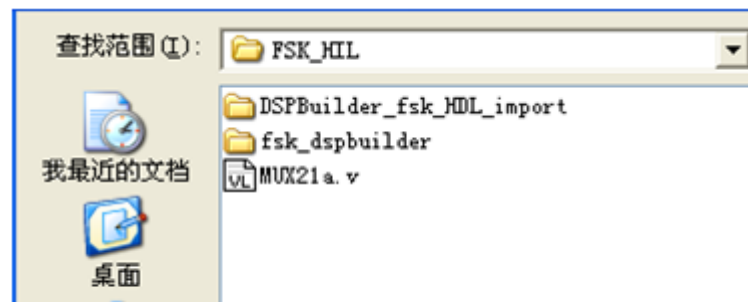


图 12-33 加入 3 个 FIR 设计文件

# 12.2 HDL模块插入仿真与设计

## 4. 仿真

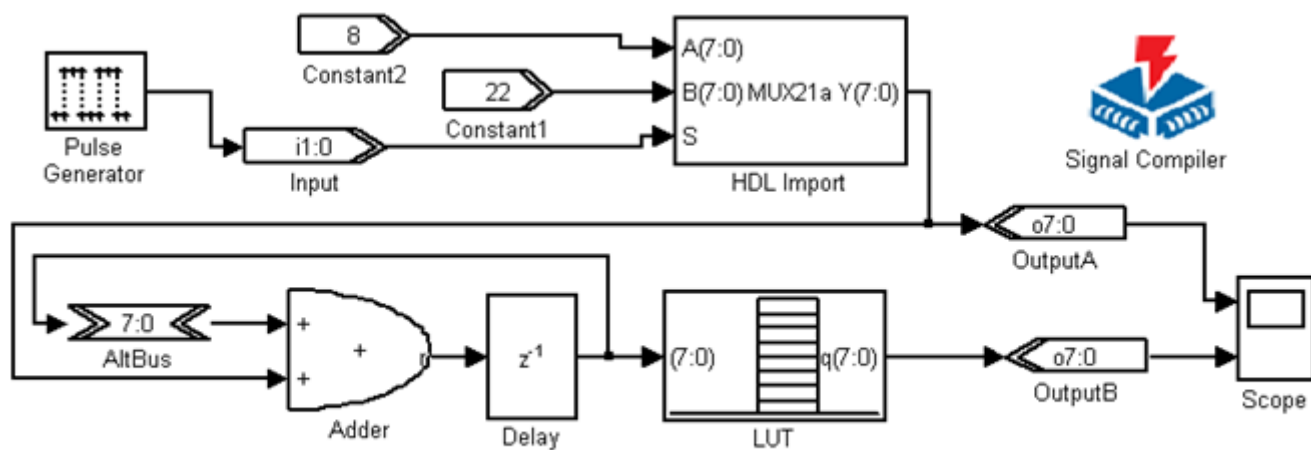


图 12-34 构成一个完整设计

## 12.3 正交幅度调制与解调模型设计

$$I(t) = a(t) \cos \omega_0 t ; \quad Q(t) = b(t) \sin \omega_0 t \quad (12-9)$$

$$X(t) = I(t) + Q(t) = a(t) \cos \omega_0 t + b(t) \sin \omega_0 t \quad (12-10)$$

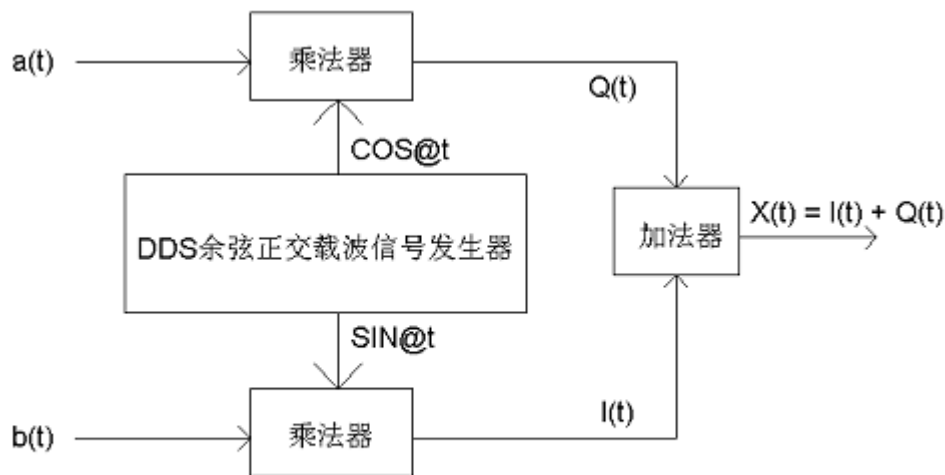


图 12-35 正交幅度调制原理图

## 12.3 正交幅度调制与解调模型设计

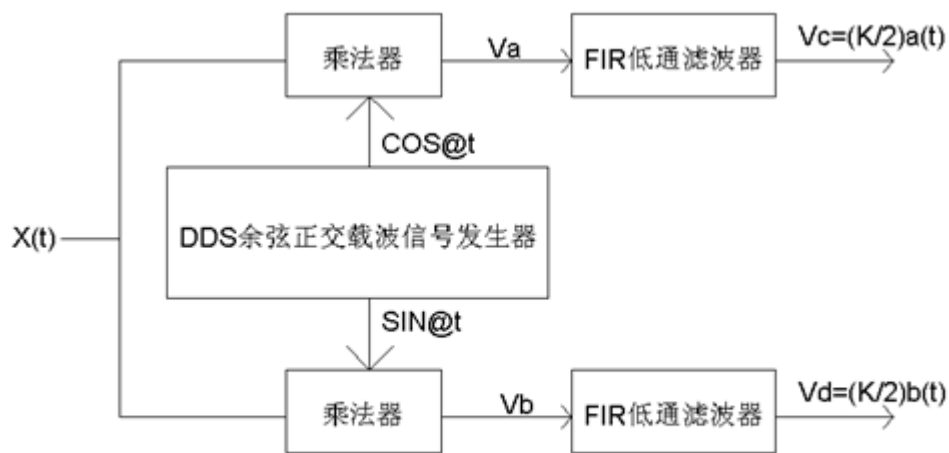


图 12-36 正交幅度信号解调原理图

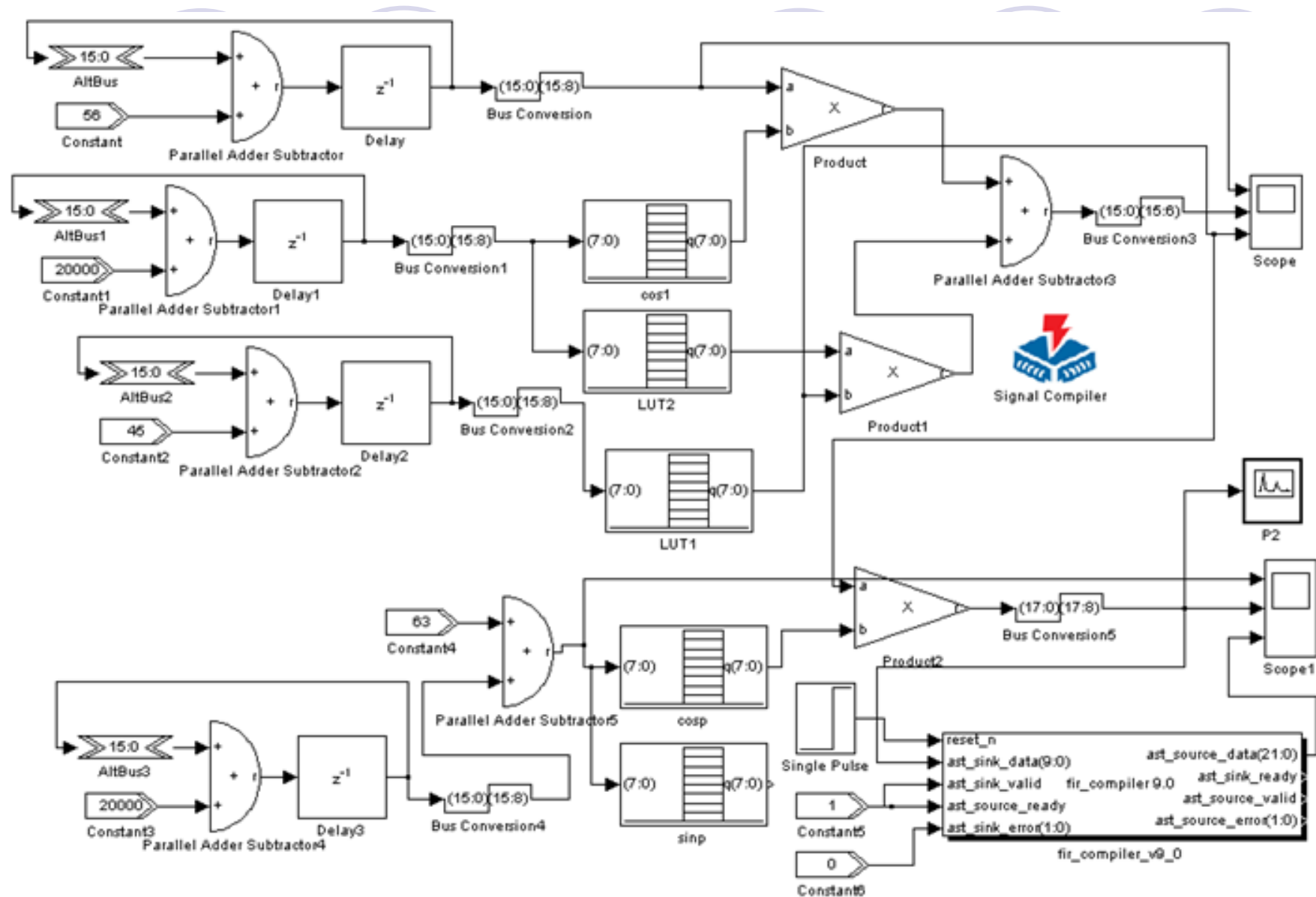


图 12-37 FIR 滤波器核的测试电路模型

## 12.3 正交幅度调制与解调模型设计

$$\begin{aligned}v_a &= X(t) \cos \omega_0 t = a(t) \cos^2 \omega_0 t + b(t) \sin \omega_0 t \cos \omega_0 t & (12-11) \\ &= \frac{1}{2}a(t) + \frac{1}{2}a(t) \cos 2\omega_0 t + \frac{1}{2}b(t) \sin 2\omega_0 t\end{aligned}$$

$$\begin{aligned}v_b &= X(t) \sin \omega_0 t = b(t) \sin^2 \omega_0 t + a(t) \sin \omega_0 t \cos \omega_0 t \\ &= \frac{1}{2}b(t) - \frac{1}{2}a(t) \cos 2\omega_0 t + \frac{1}{2}a(t) \sin 2\omega_0 t\end{aligned}$$

$$v_c = \frac{K}{2}a(t) \qquad v_d = \frac{K}{2}b(t) \qquad (12-12)$$



## 12.3 正交幅度调制与解调模型设计

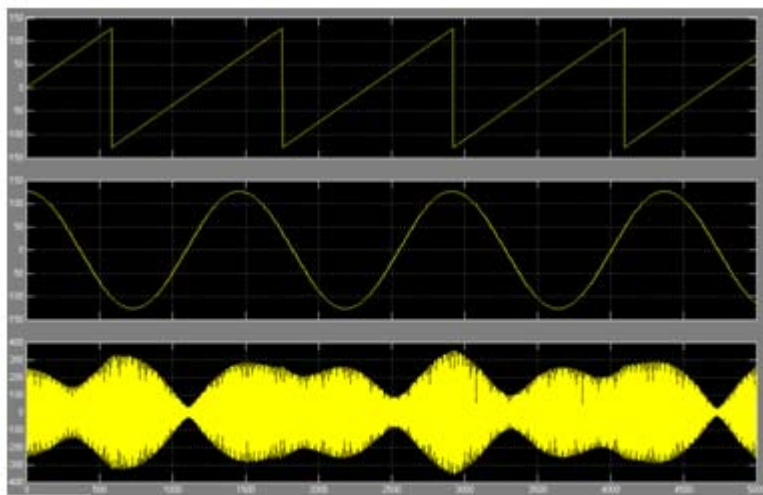


图 12-38 Scopel 显示后的波形

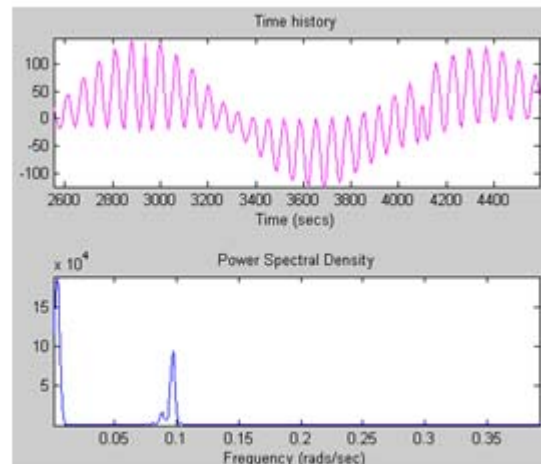


图 12-39 图 12-37 电路中频谱仪的输出波形

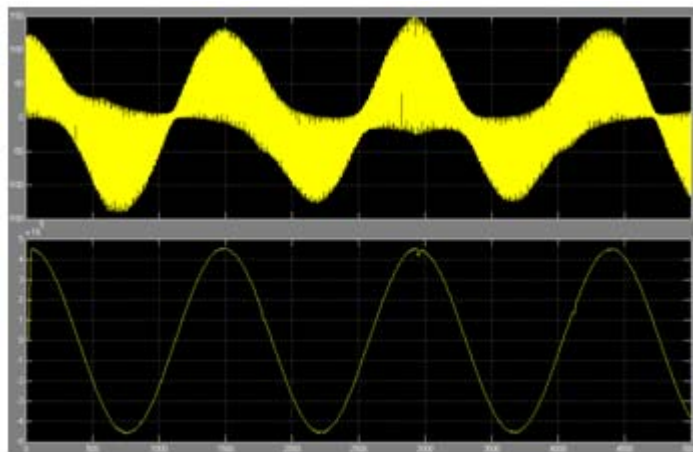


图 12-40 图 12-37 FIR 滤波后的输出波形

# 12.4 NCO IP核应用

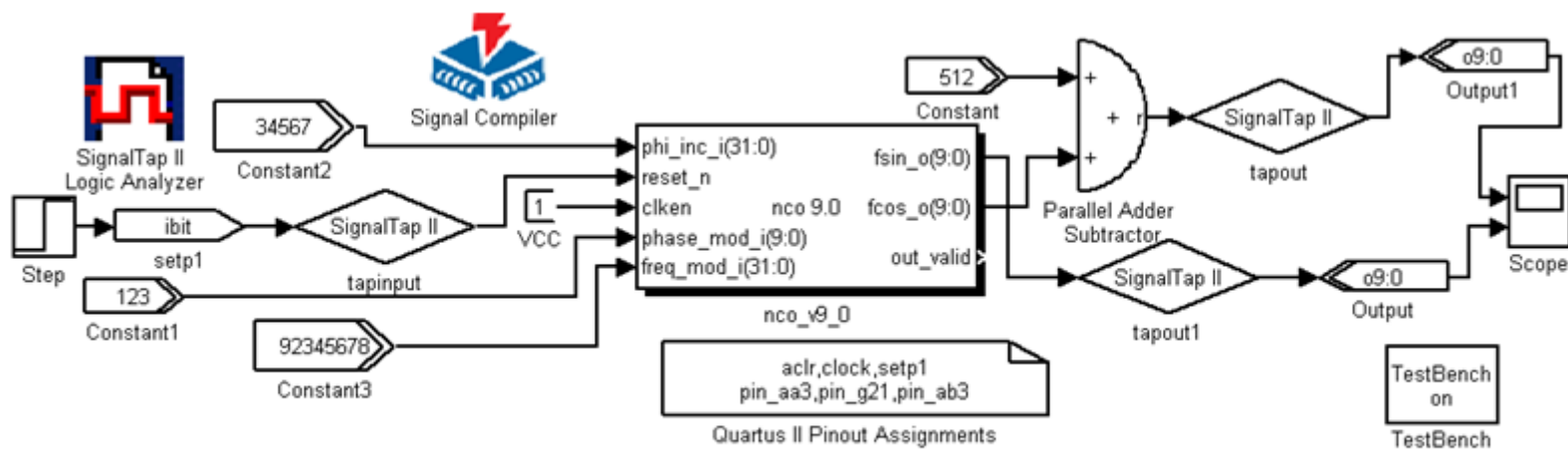


图 12-41 FIR 滤波器核的测试电路模型

# 12.4 NCO IP核应用

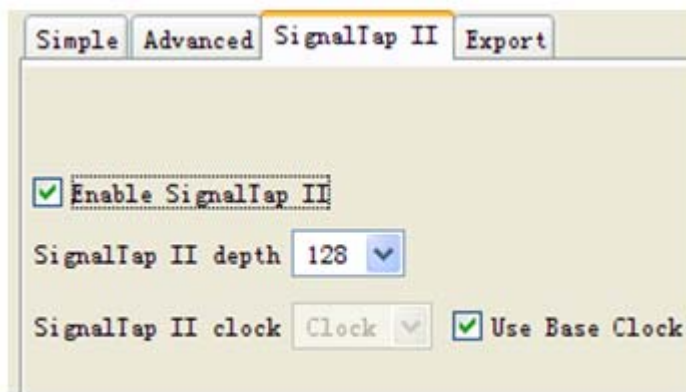


图12-42 选中Enable SignalTap II

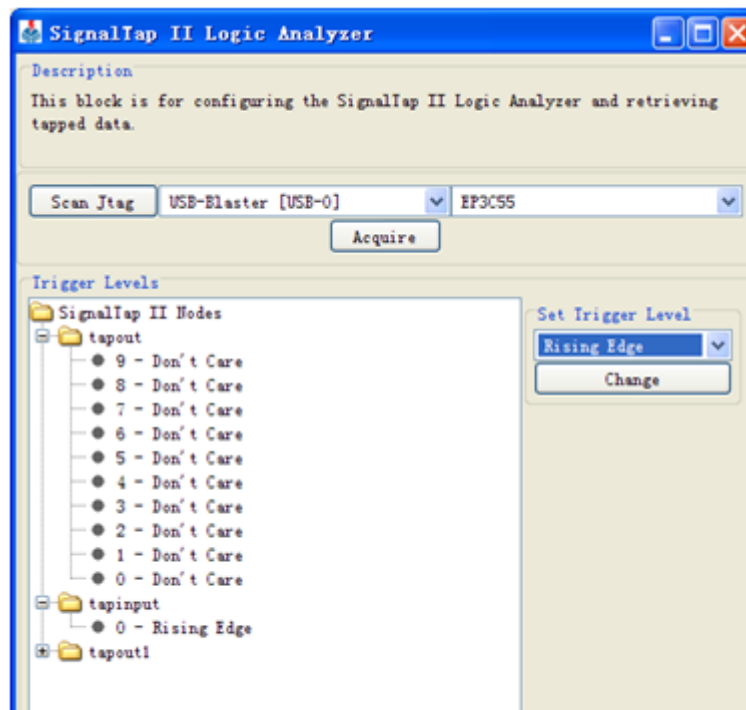


图12-43 SignalTap II观察节点

# 12.4 NCO IP核应用



图12-44 设置触发信号

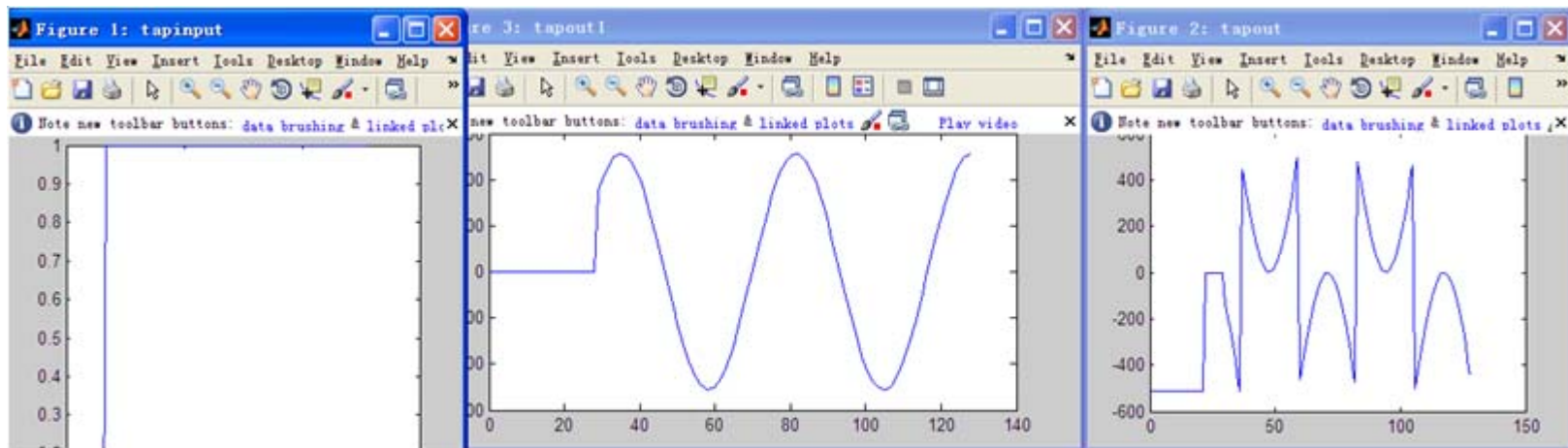


图12-45 Matlab画出的信号波形

# 12.5 基于IP的数字编译码器设计

表 12-1 可以纠正  $t$  个符号错误的 RS 码参数

参数名称	参 数 值	单 位
码长	$n = 2^m - 1$	符号
信息段	$k$	符号
监督段	$n - k = 2t$	符号
最小码距	$2t + 1$	符号
符号	$m$	比特

# 12.5 基于IP的数字编译码器设计



```
reset                                rsout(7:0)
rsin(7:0)                             sink_ena
sink_val                             reed_solomon 9.0  source_val
sink_sop                             source_sop
sink_eop                             source_eop
source_ena                            source_eop

reed_solomon_v9_0
```

图 12-46 RS Compiler 与 DSP Builder 集成

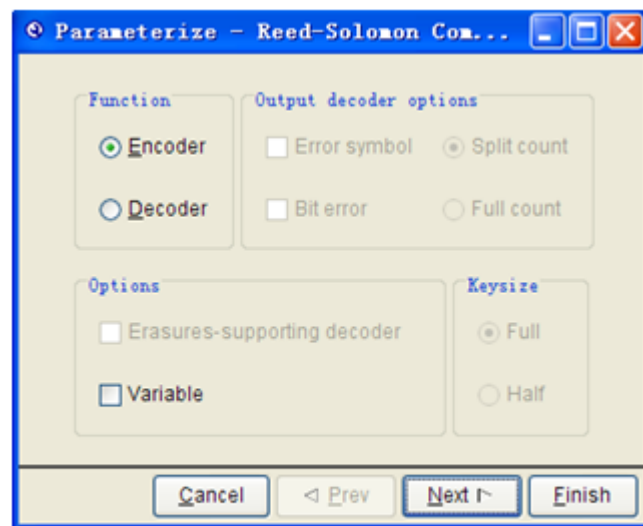


图 12-47 选择类型为 RS 编码器

# 12.5 基于IP的数字编译码器设计

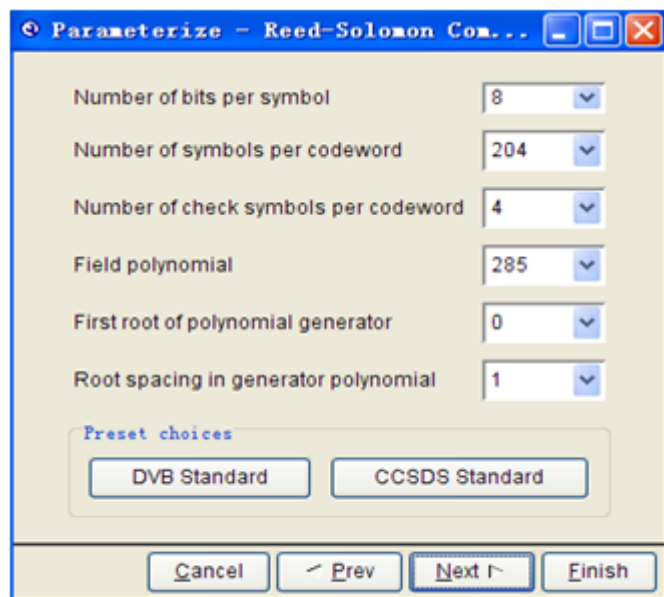


图 12-48 确定参数

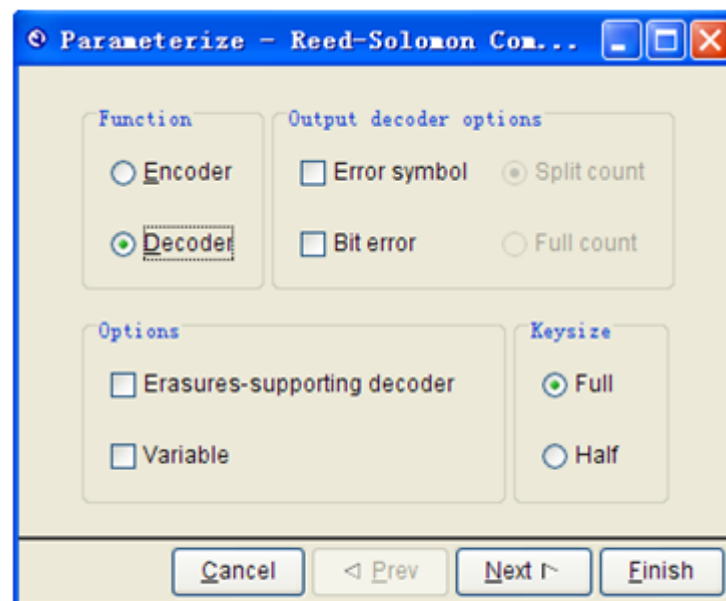


图 12-49 选择类型为 RS 译码器

# 习 题

$$x[i] = \sum_{k=0}^{31} X[k] * \cos\left[\frac{i+16}{64} \pi(2k+1)\right], i=0 \cdots 63$$

$$x[i] = \sum_{k=0}^{31} X[k] * \cos\left[\frac{i * \pi}{64} \pi(2k+1)\right], i=0 \cdots 31$$

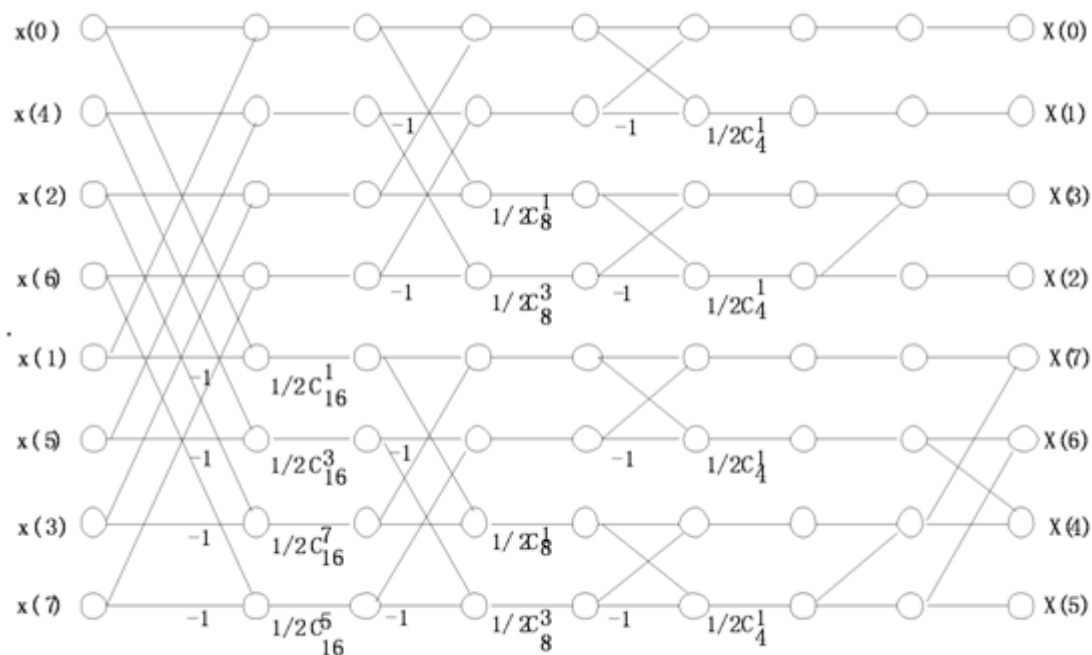


图 12-50 Lee's 快速 DCT 算法



# 实验与设计

## 实验12-1 FIR数字滤波器设计实验

$$h(0) = h(5) = 25, \quad h(1) = h(4) = 93, \quad h(2) = h(3) = 212$$

$$h(n) = C_q [h(0)(x(n) + x(n-5)) + h(1)(x(n-1) + x(n-4)) + h(2)(x(n-2) + x(n-3))]$$

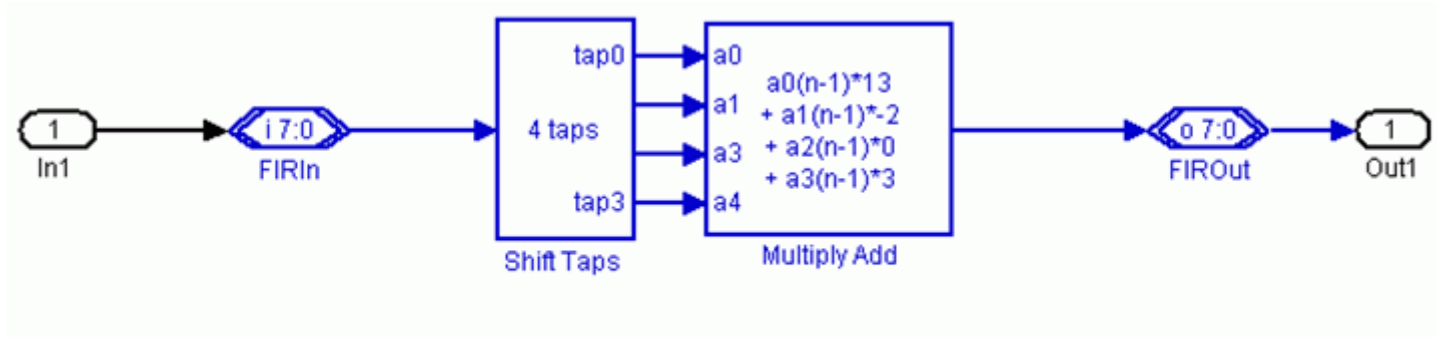


图 12-51 Shift Taps 模块和 Multiply Add 模块应用



# 实验与设计

实验12-2 编译码器与调制解调模块设计实验

实验12-3 HDL Import模块应用实验