

# 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

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

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

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

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

# 12.1 FIR数字滤波器设计

## 12.1.1 FIR滤波器原理

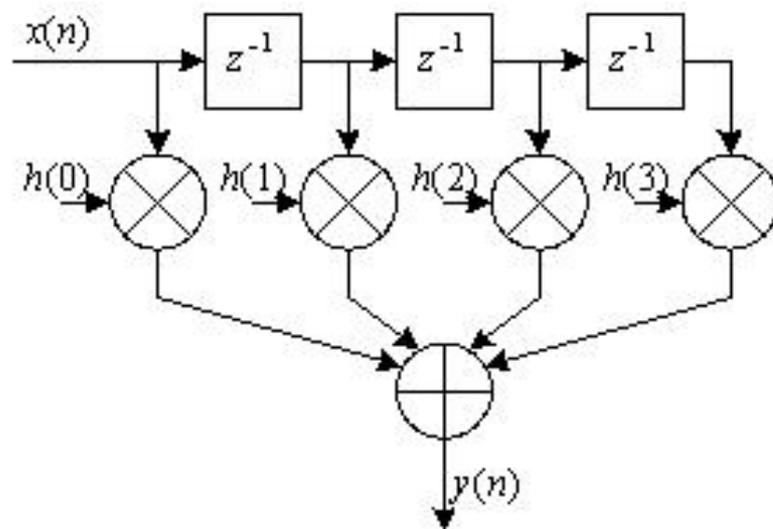


图 12-1 3阶 FIR 滤波器结构

# 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

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

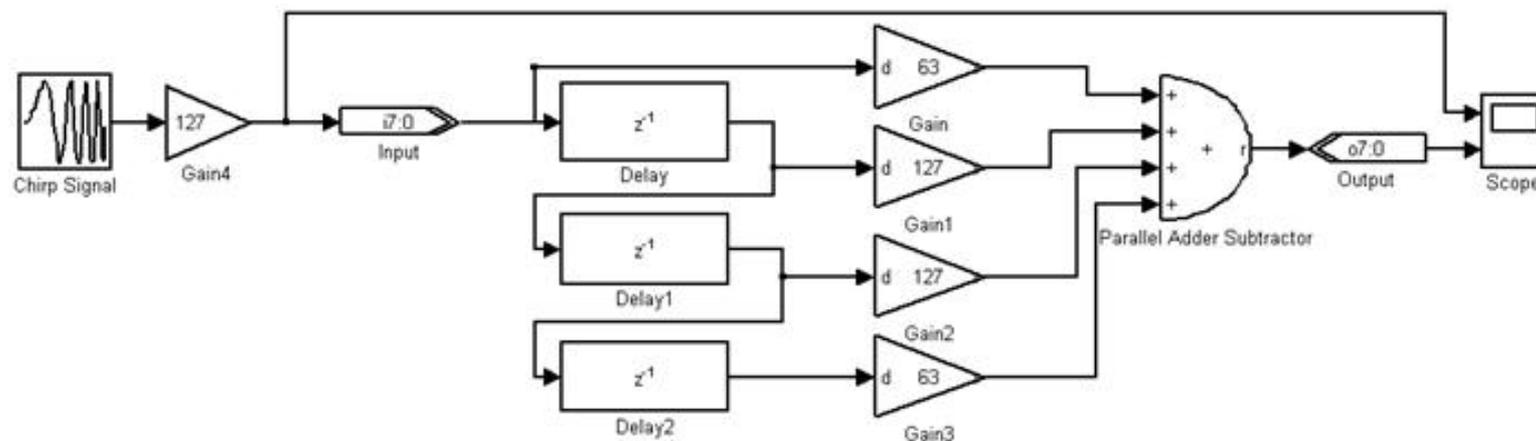


图 12-2 带有仿真信号模块的 3 阶滤波器模型

# 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

使用Interpret vectors parameters as 1-D

#### 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 设为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

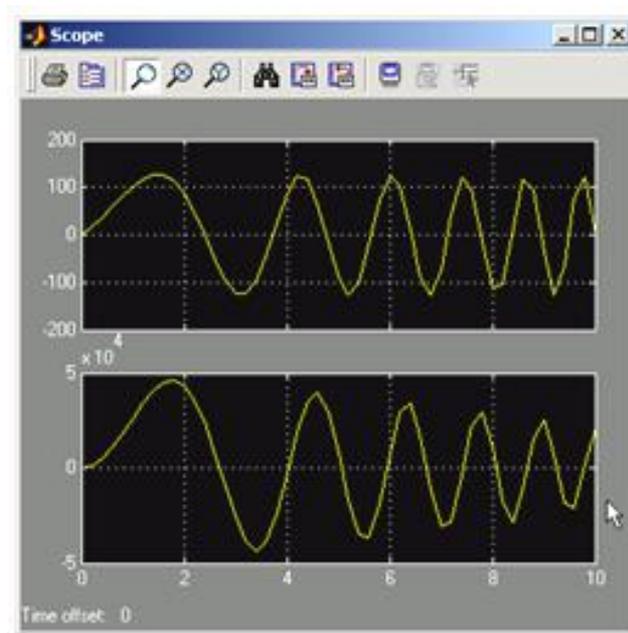


图 12-3 FIR 滤波器仿真结果

# 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

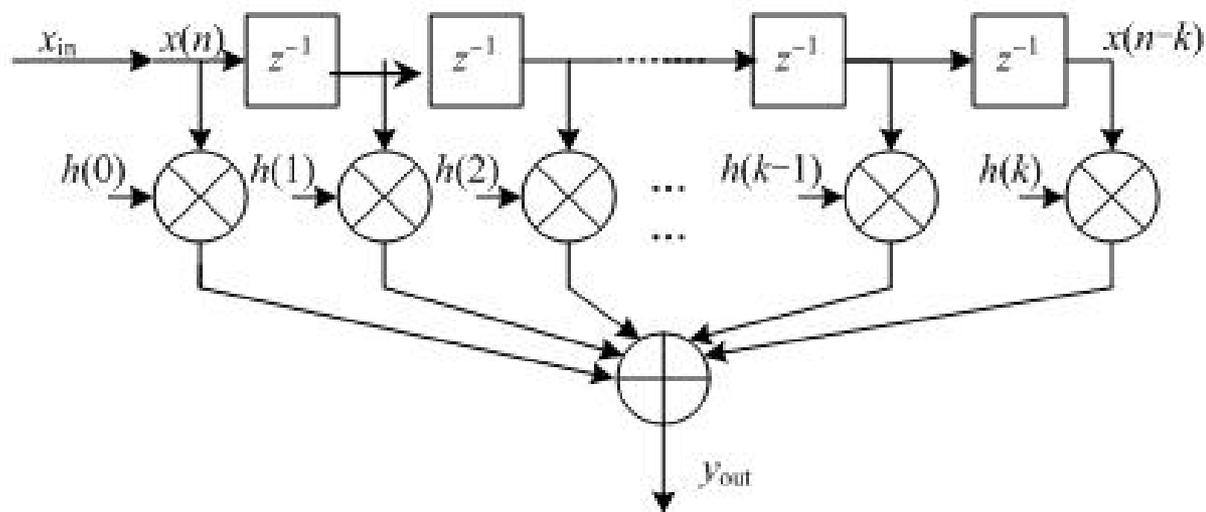


图 12-4 直接 I 型 FIR 滤波器结构

# 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

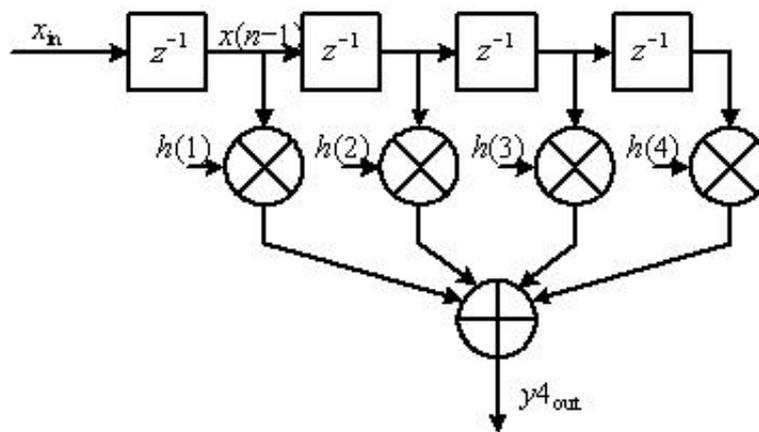


图 12-5 直接 I 型 4 阶 FIR 滤波器节

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

# 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

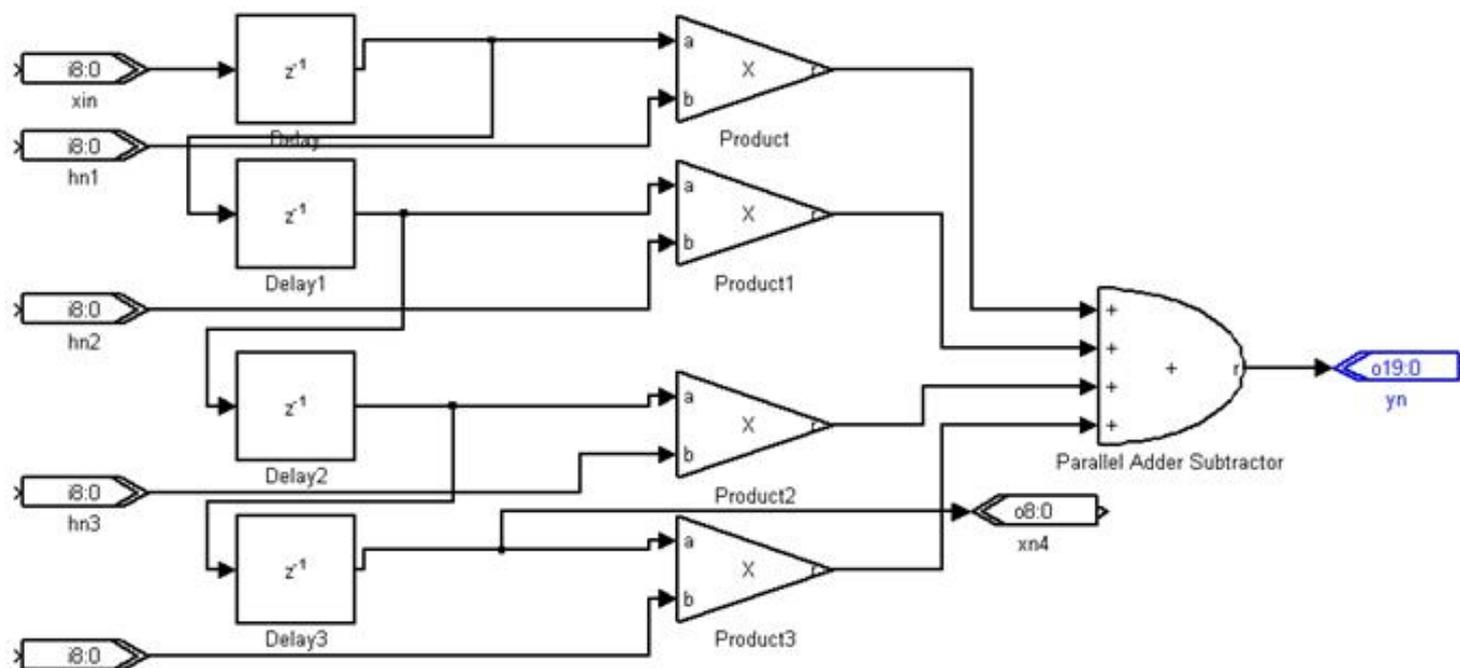


图 12-6 直接 I 型 4 阶 FIR 滤波器节

## 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

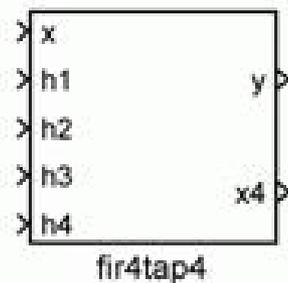


图 12-7 fir4tap4 子系统

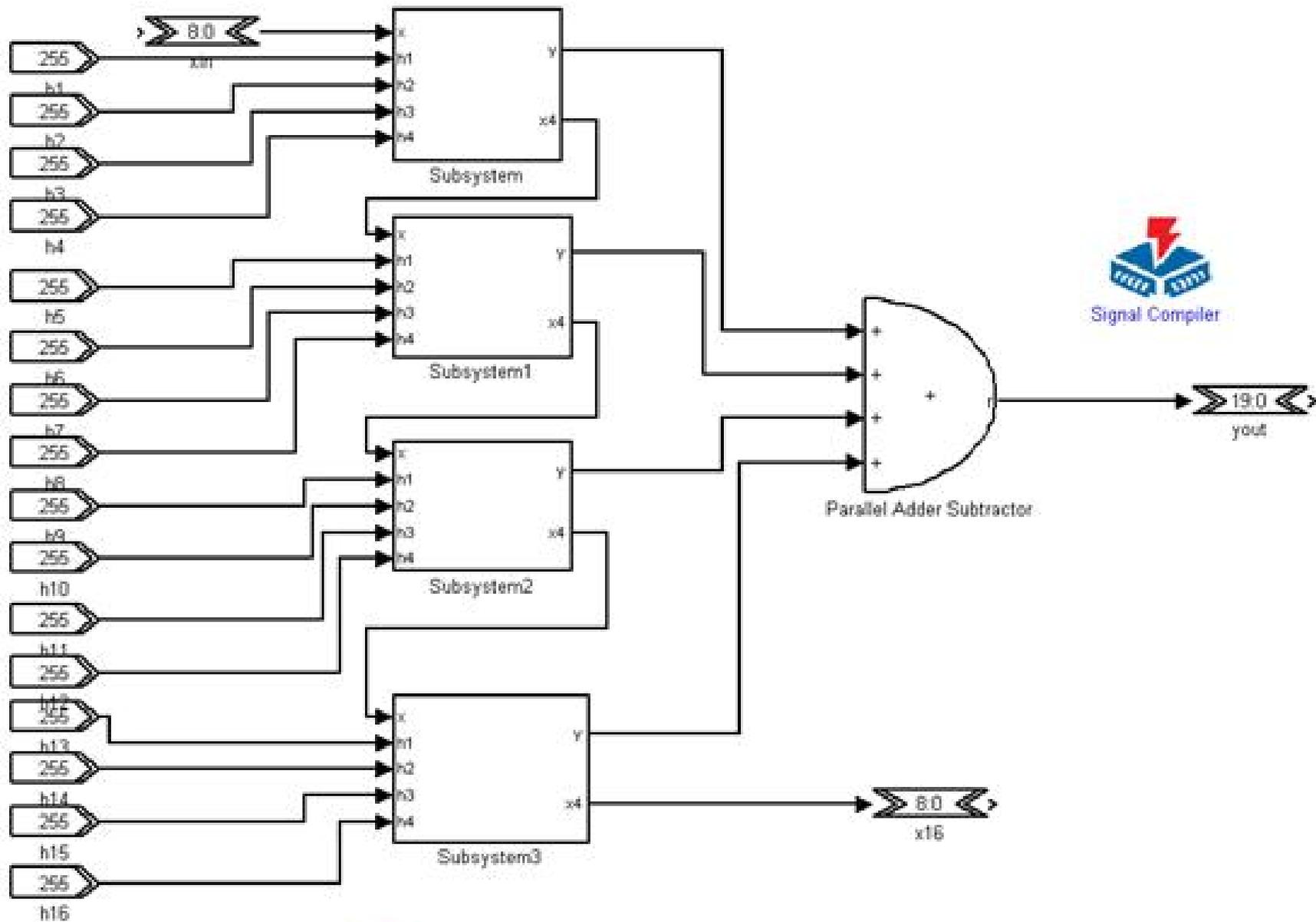


图 12-8 16 阶直接 I 型 FIR 滤波器模型

# 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

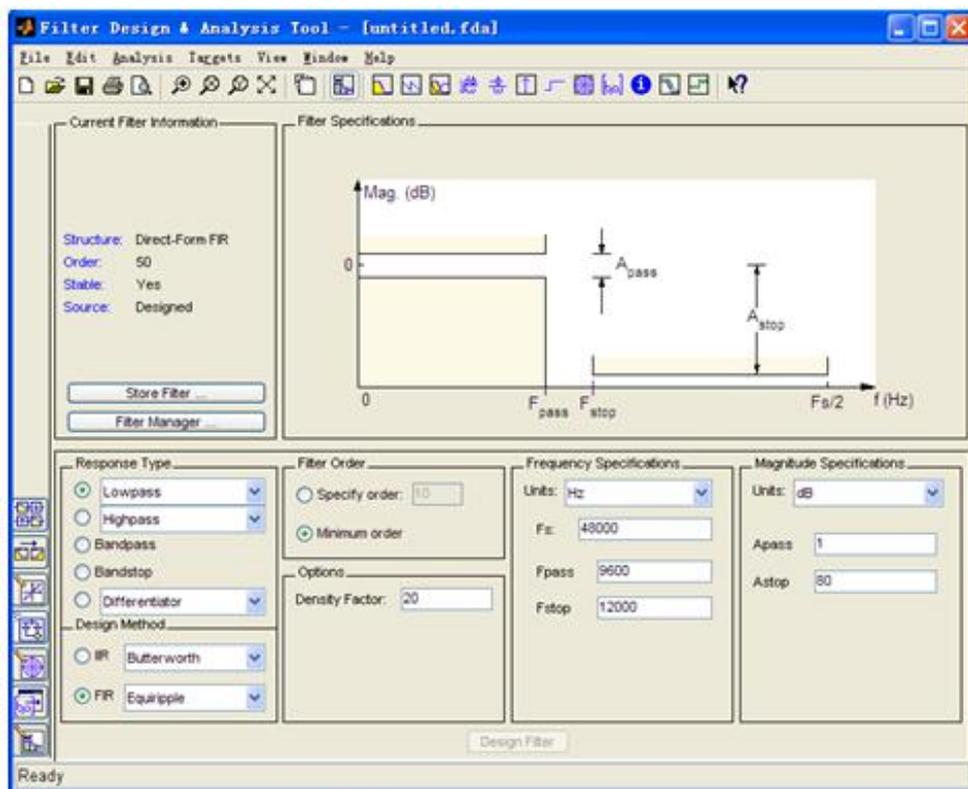


图 12-9 FDATool 界面

# 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

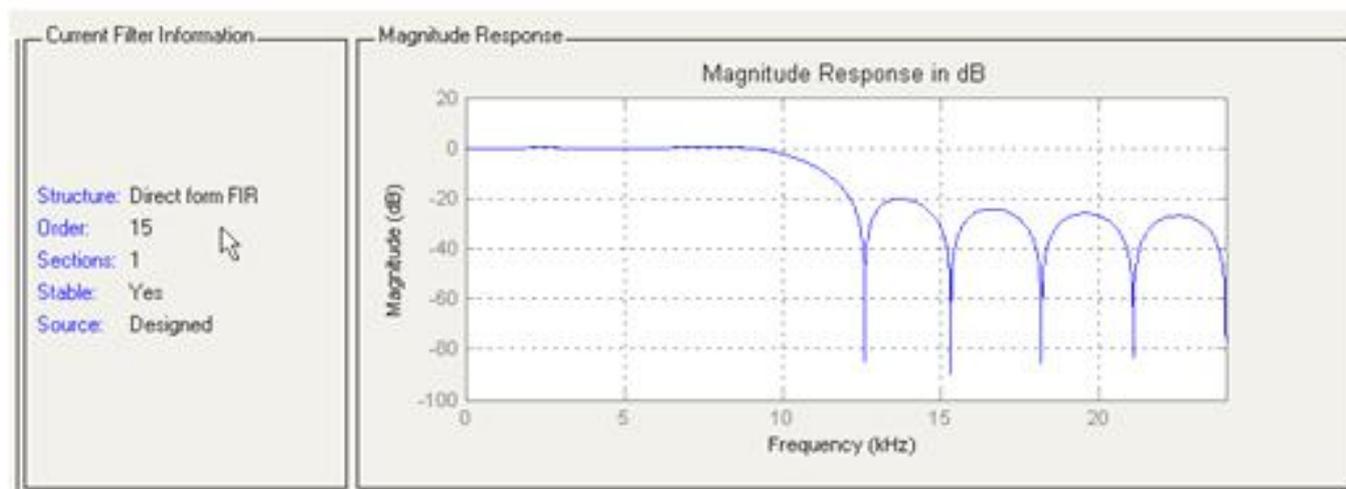


图 12-10 FIR 滤波器的幅频响应

# 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

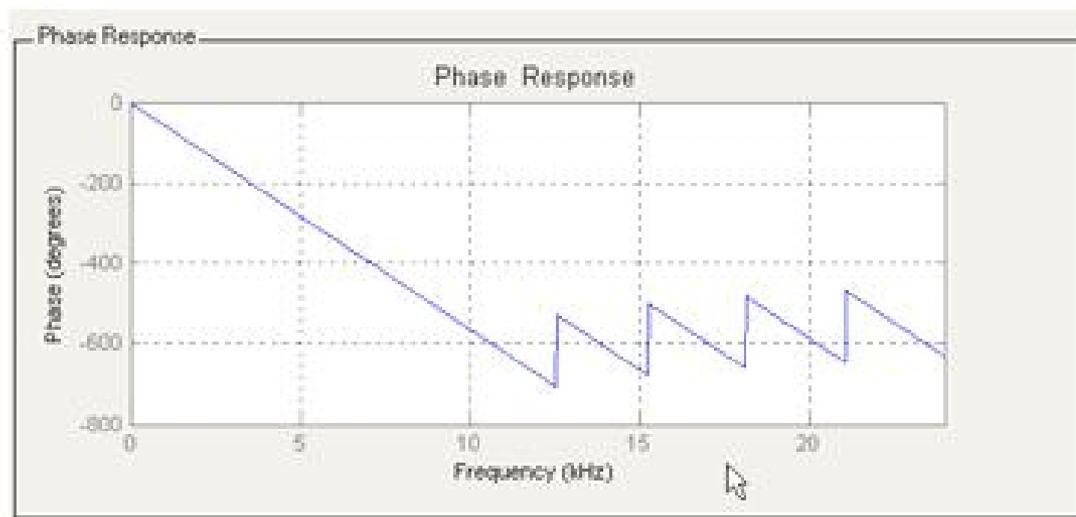


图 12-11 FIR 滤波器的相频响应

# 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

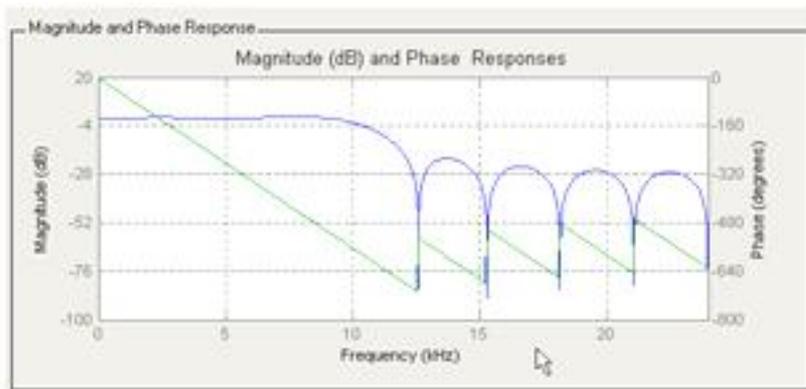


图 12-12 幅频响应与相频响应比较

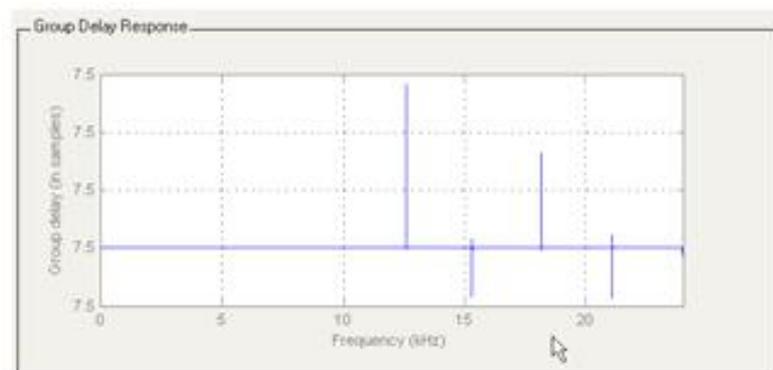


图 12-13 FIR 滤波器的群延时

# 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

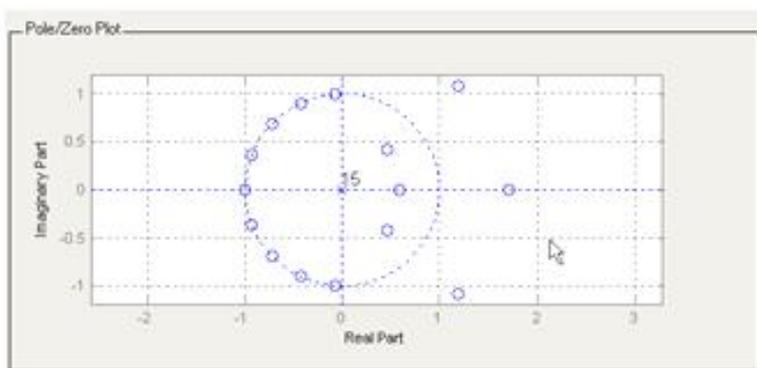


图 12-14 FIR 滤波器的零极点

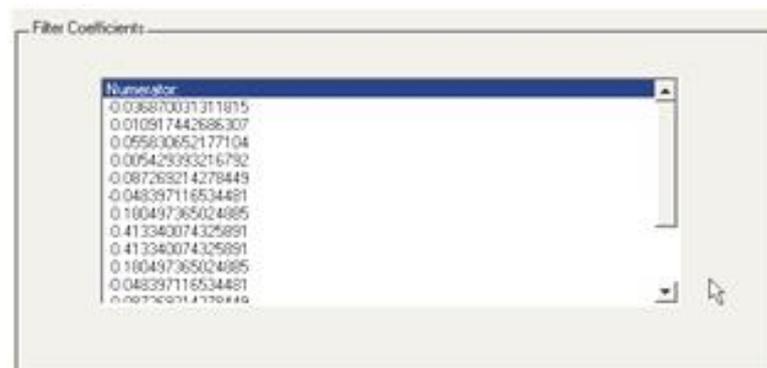


图 12-15 FIR 滤波器系数

# 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

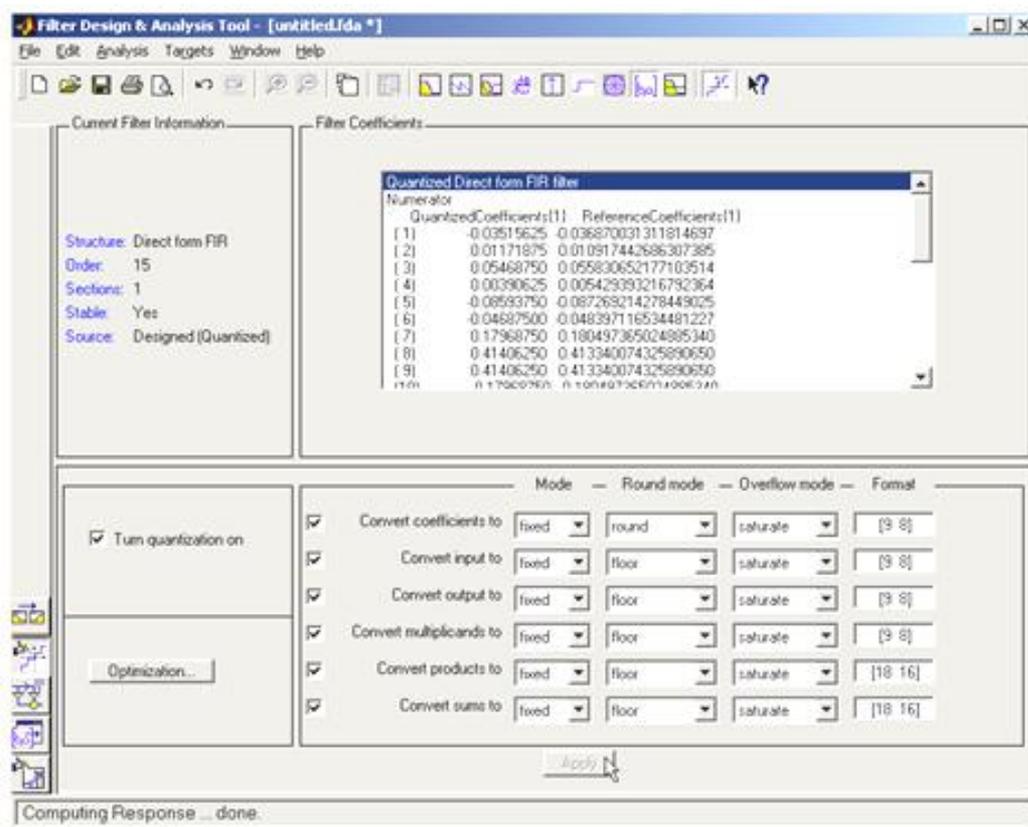


图 12-16 量化参数设置

# 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 12-17 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

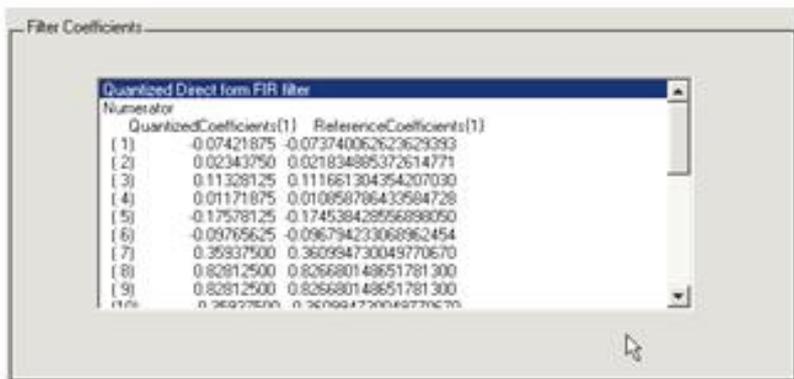


图 12-18 量化后系数

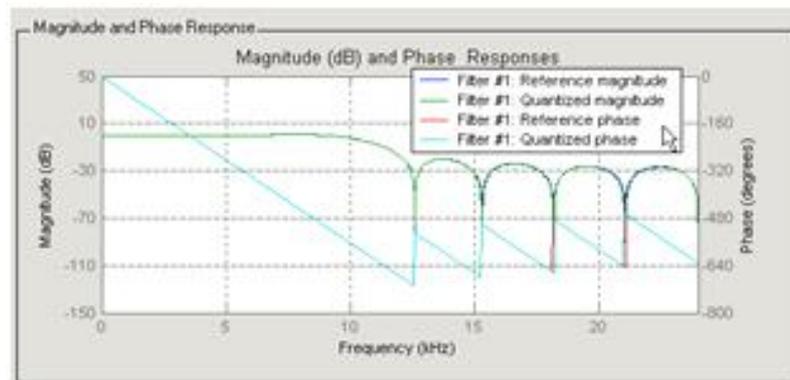


图 12-19 量化后幅频、相频响应

# 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

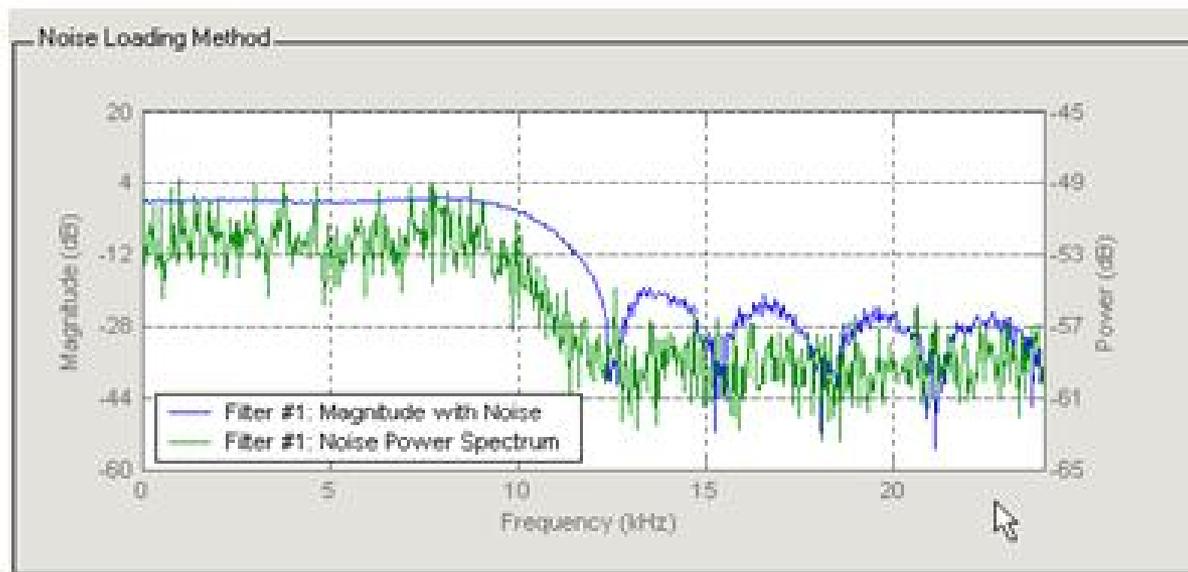


图 12-20 量化后噪声分析

# 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

### 6. 修改FIR滤波器模型添加参数

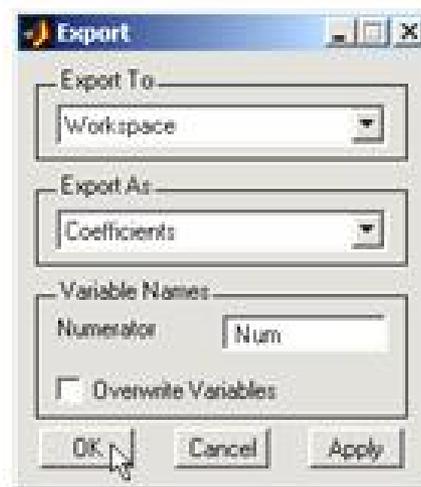


图 12-21 导出对话框

### 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

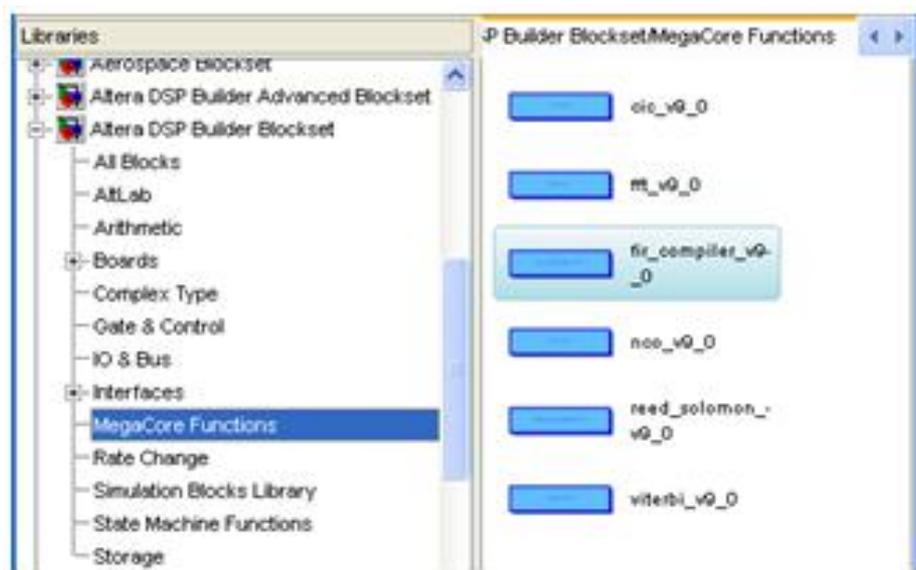


图 12-22 IP Core 模块库

# 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

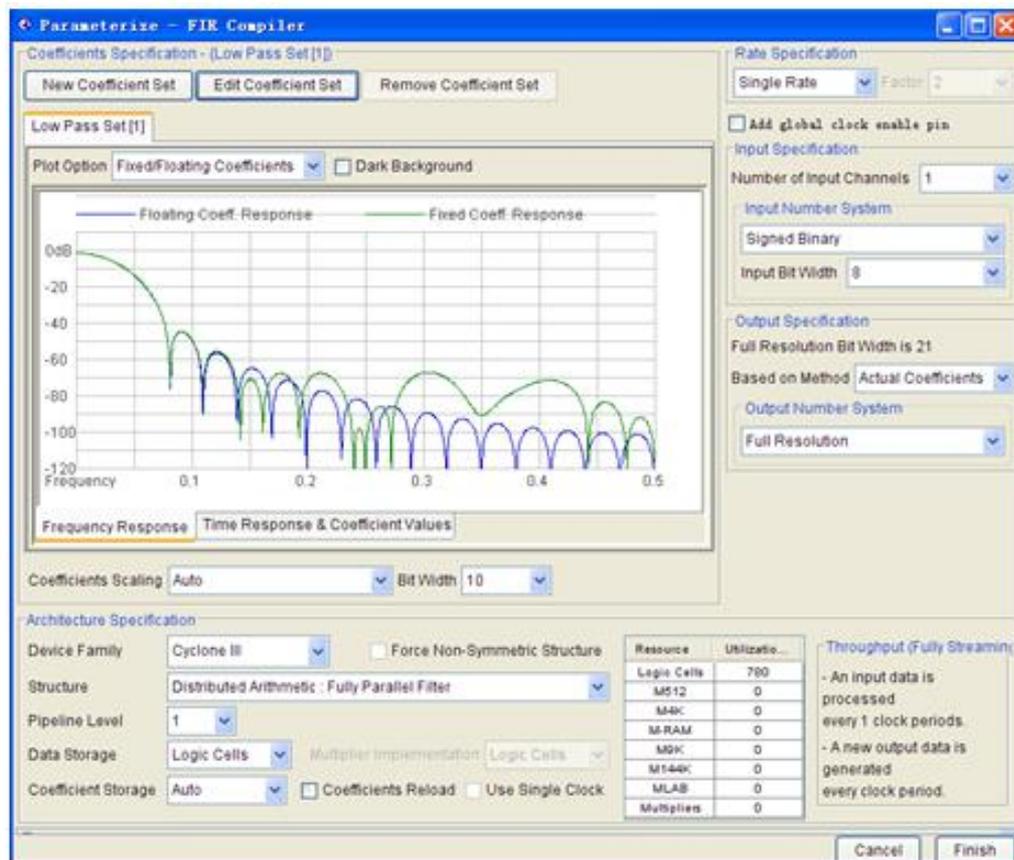


图 12-24 确定 FIR 滤波器系数

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

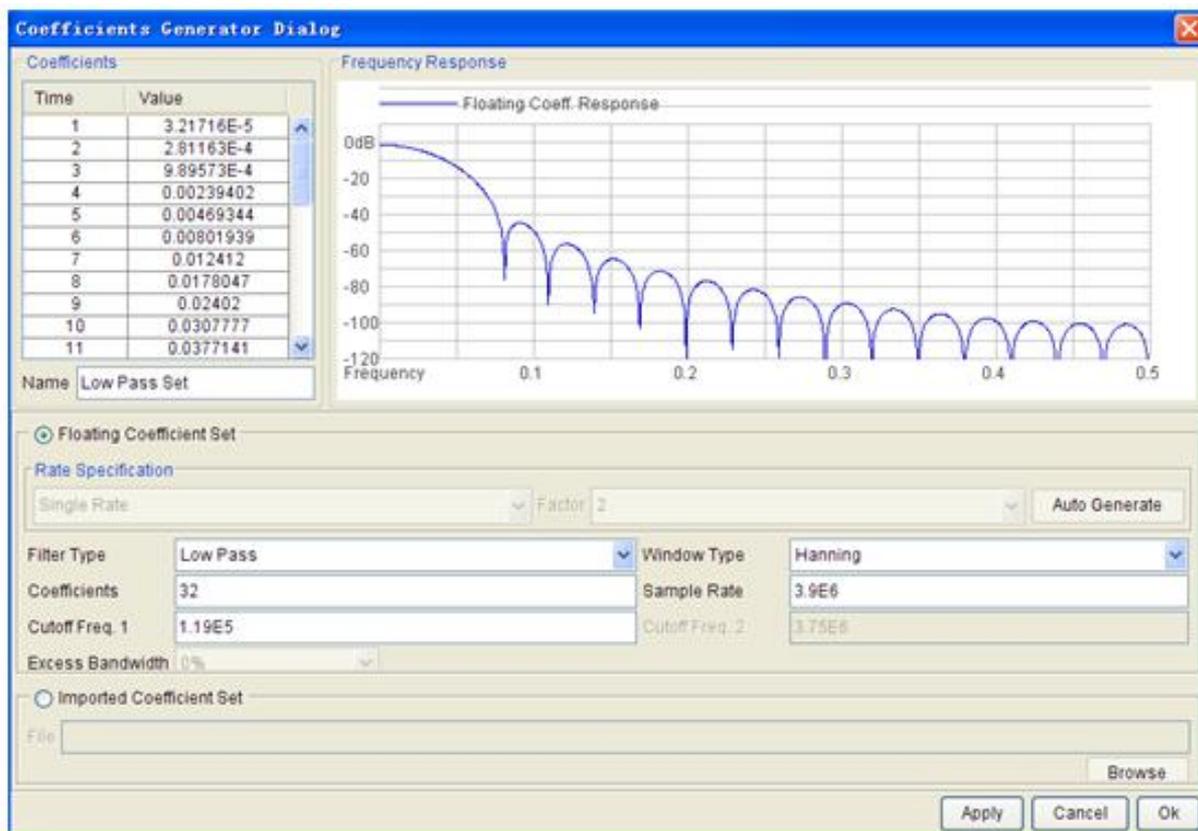


图 12-25 确定 FIR 工作方式

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

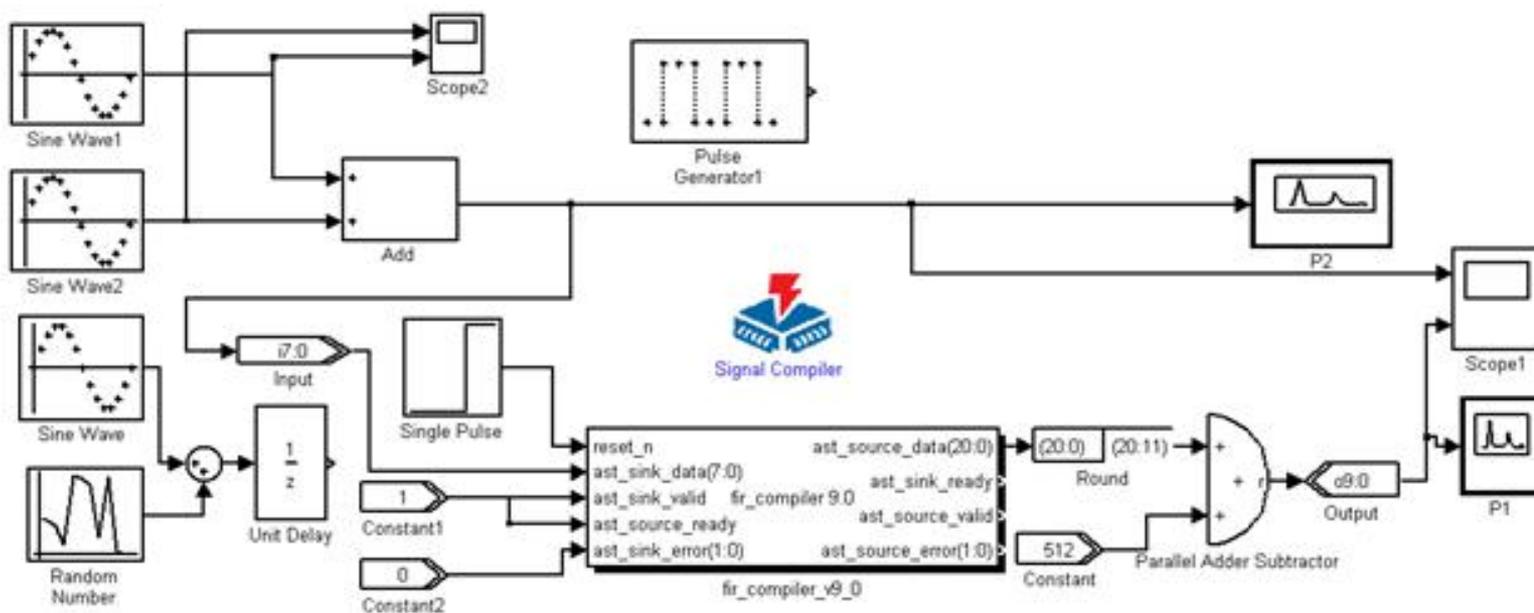


图 12-26 FIR 滤波器核的测试电路模型

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

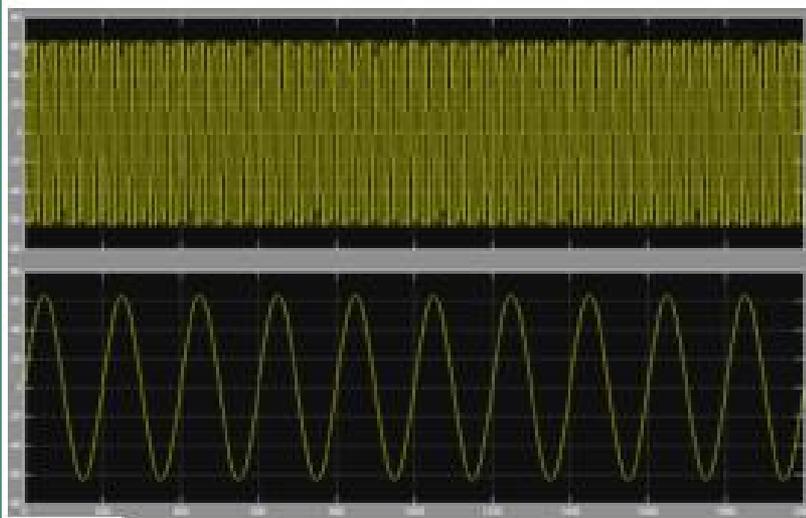


图 12-27 Scope2 显示波形

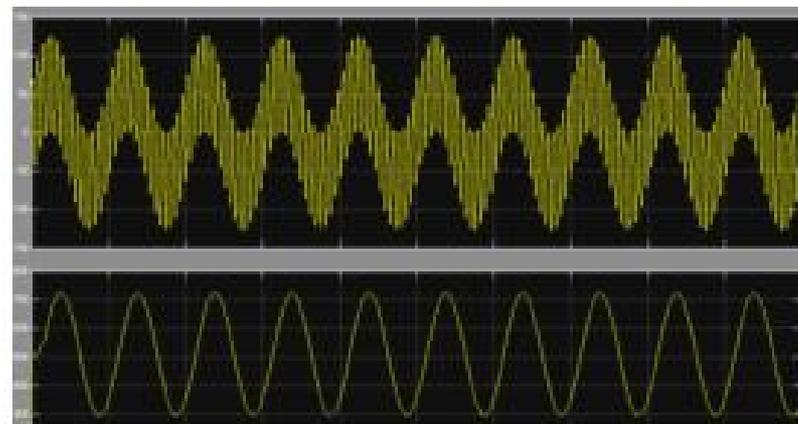


图 12-28 Scope1 显示波形

# 12.1 FIR数字滤波器设计

## 12.1.4 使用FIR IP Core设计FIR滤波器

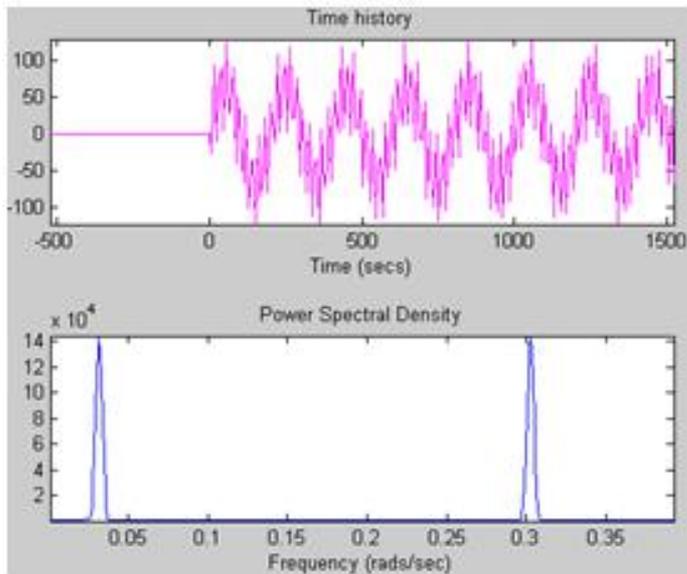


图 12-29 P2 频谱仪显示波形

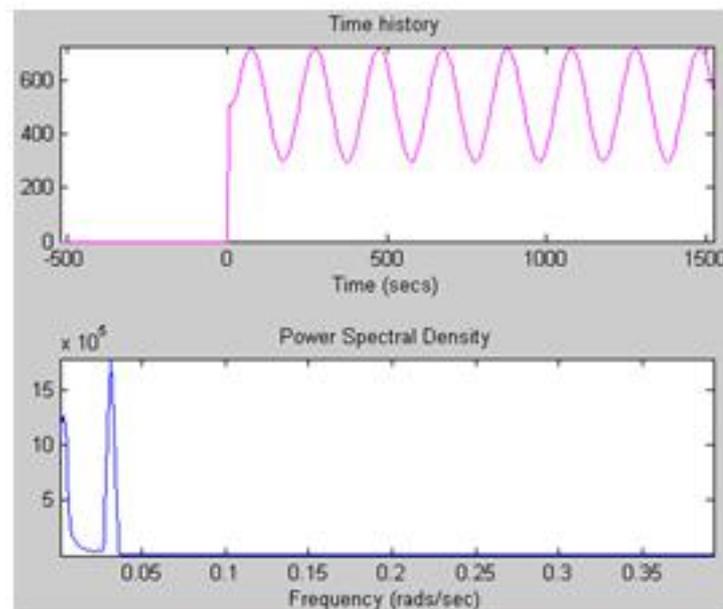


图 12-30 P1 频谱仪显示波形

# 12.2 HDL模块插入仿真及其设计

## 1. 完成VHDL设计

### 【例 12-1】

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
Entity fir_vhdl is
Port(clock      : in std_logic;  sclr : in std_logic:='0';
      data_in   : in std_logic_vector(15 downto 0);
      data_out  : out std_logic_vector(32 downto 0));
end fir_vhdl;
```

# 12.2 HDL模块插入仿真及其设计

## 1. 完成VHDL设计

### 【例 12-2】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY lpm;
USE lpm.lpm_components.all;
ENTITY final_add IS
PORT ( data, datab : IN STD_LOGIC_VECTOR (32 DOWNT0 0);
      Clock, aclr : IN STD_LOGIC;
      Result : OUT STD_LOGIC_VECTOR (32 DOWNT0 0) );
END final_add;
```

# 12.2 HDL模块插入仿真及其设计

## 1. 完成VHDL设计

### 【例 12-3】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;
ENTITY four_mult_add IS
PORT( clock0   : IN STD_LOGIC   := '1';
      dataa_0  : IN STD_LOGIC_VECTOR (15 DOWNTO 0) := (OTHERS=>'0');
      aclr3    : IN STD_LOGIC   := '0';
      datab_0 : IN STD_LOGIC_VECTOR (13 DOWNTO 0) := (OTHERS=>'0');
      datab_1 : IN STD_LOGIC_VECTOR (13 DOWNTO 0) := (OTHERS=>'0');
      datab_2 : IN STD_LOGIC_VECTOR (13 DOWNTO 0) := (OTHERS=>'0');
      datab_3 : IN STD_LOGIC_VECTOR (13 DOWNTO 0) := (OTHERS=>'0');
      shiftouta : OUT STD_LOGIC_VECTOR (15 DOWNTO 0);
      result   : OUT STD_LOGIC_VECTOR (31 DOWNTO 0) );
END four_mult_add;
```

# 12.2 HDL模块插入仿真及其设计

## 2. 调入HDL Import模块

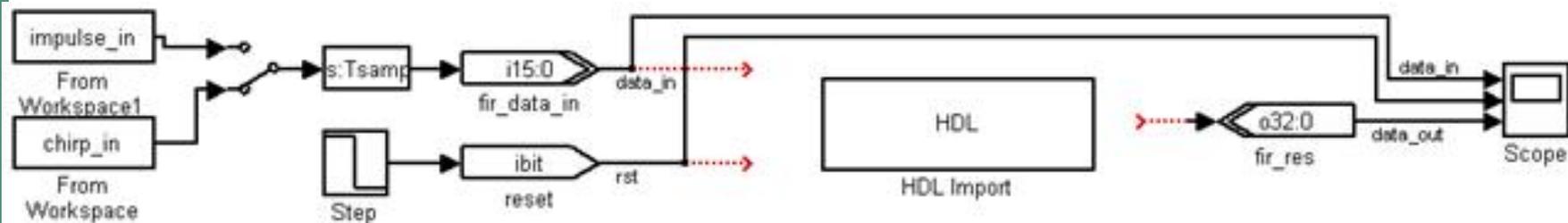


图 12-31 在一个 Simulink 空模型中调入一个 HDL Import 模块

# 12.2 HDL模块插入仿真与设计

## 3. 加入VHDL设计文件

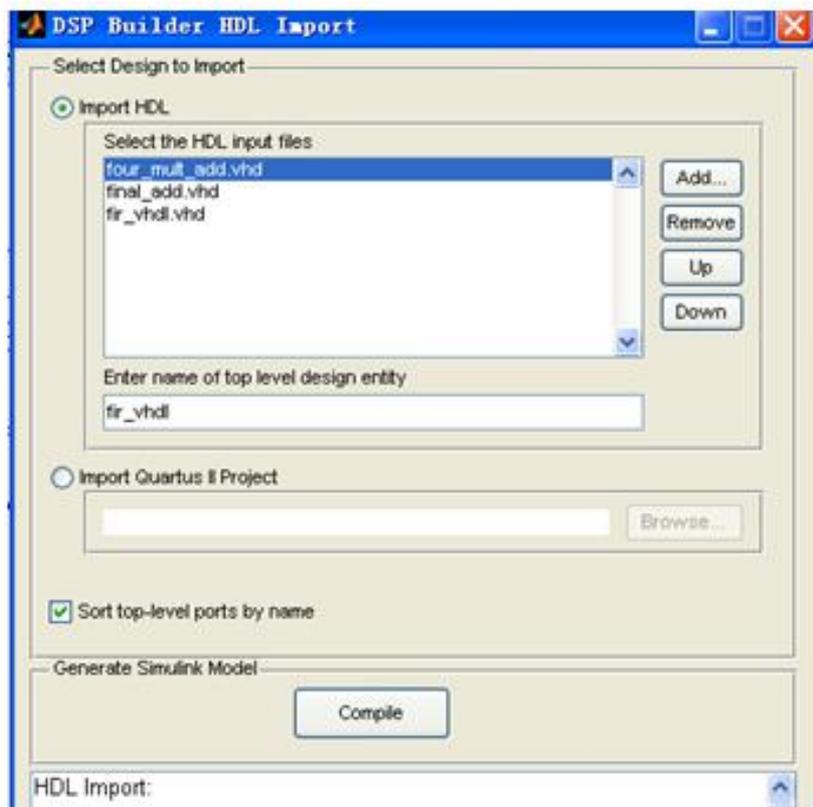


图 12-32 浏览到 3 个 VHDL 文件



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

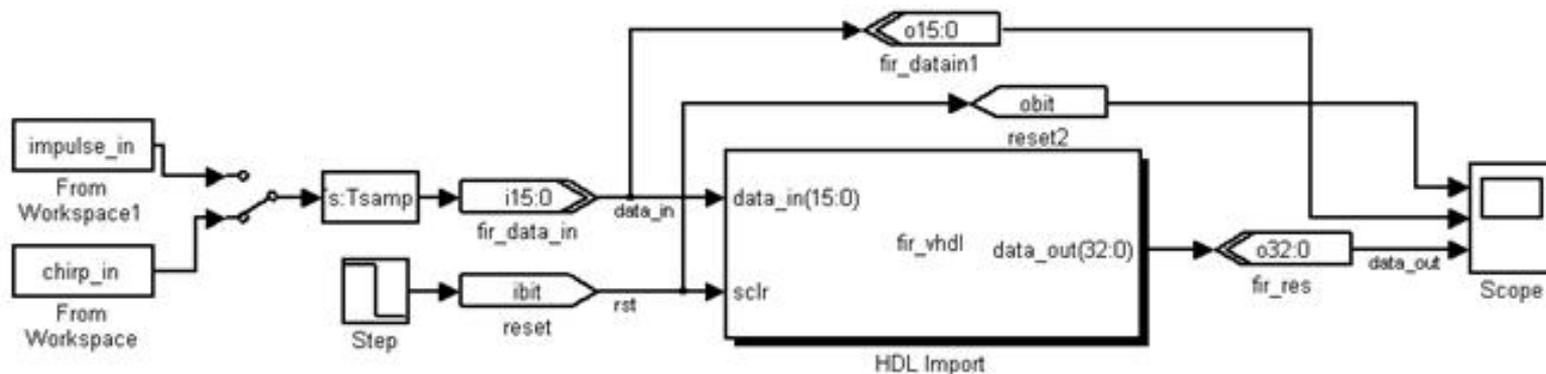


图 12-34 构成一个完整设计

# 12.2 HDL模块插入仿真与设计

## 4. 仿真

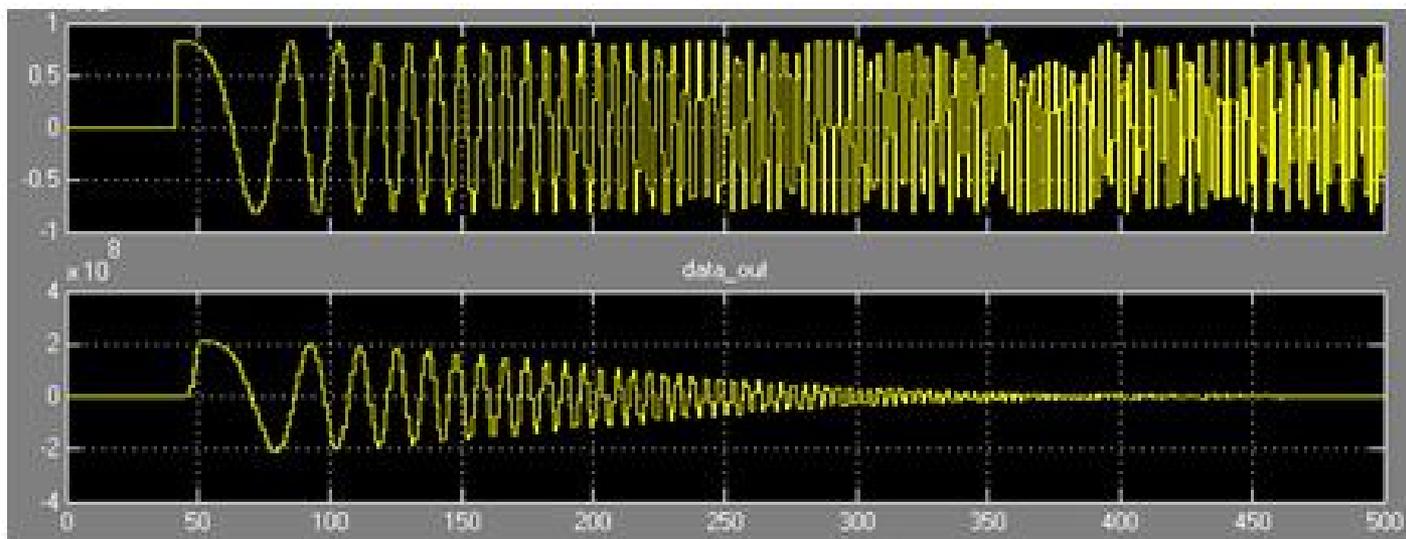


图 12-35 图 12-34 模型的仿真波形

# 习 题

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

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

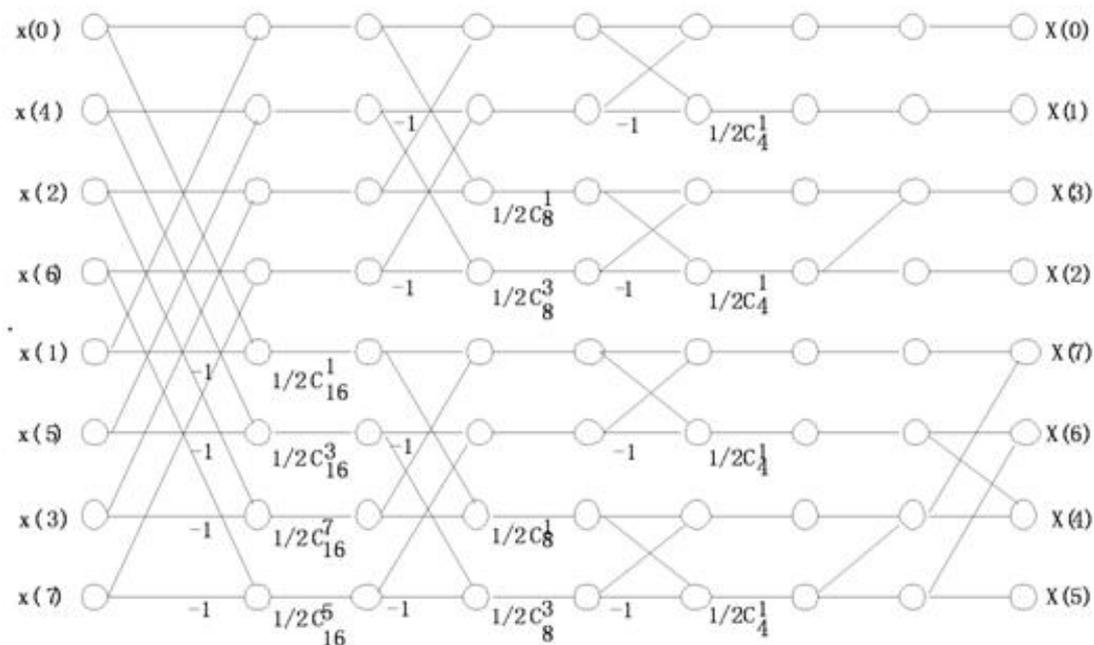


图 12-41 Lee 氏快速 DCT 算法

# 实验与设计

## 实验12-1 FIR数字滤波器设计实验

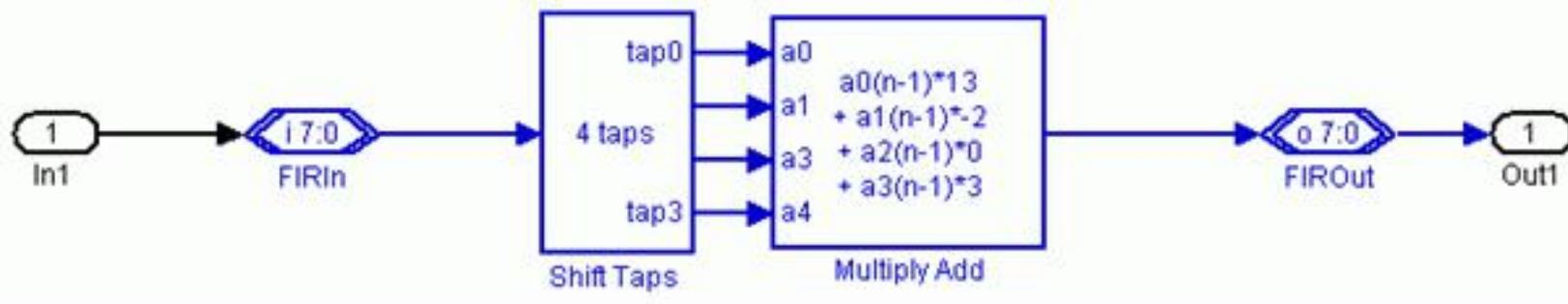


图 12-42 Shift Taps 模块和 Multiply Add 模块应用

## 实验12-2 调制解调模块设计实验

## 实验12-3 HDL Import模块应用实验