



# 第11章



# DSP Buider设计初步

# 11.1 MATLAB/DSP Builder及其设计流程

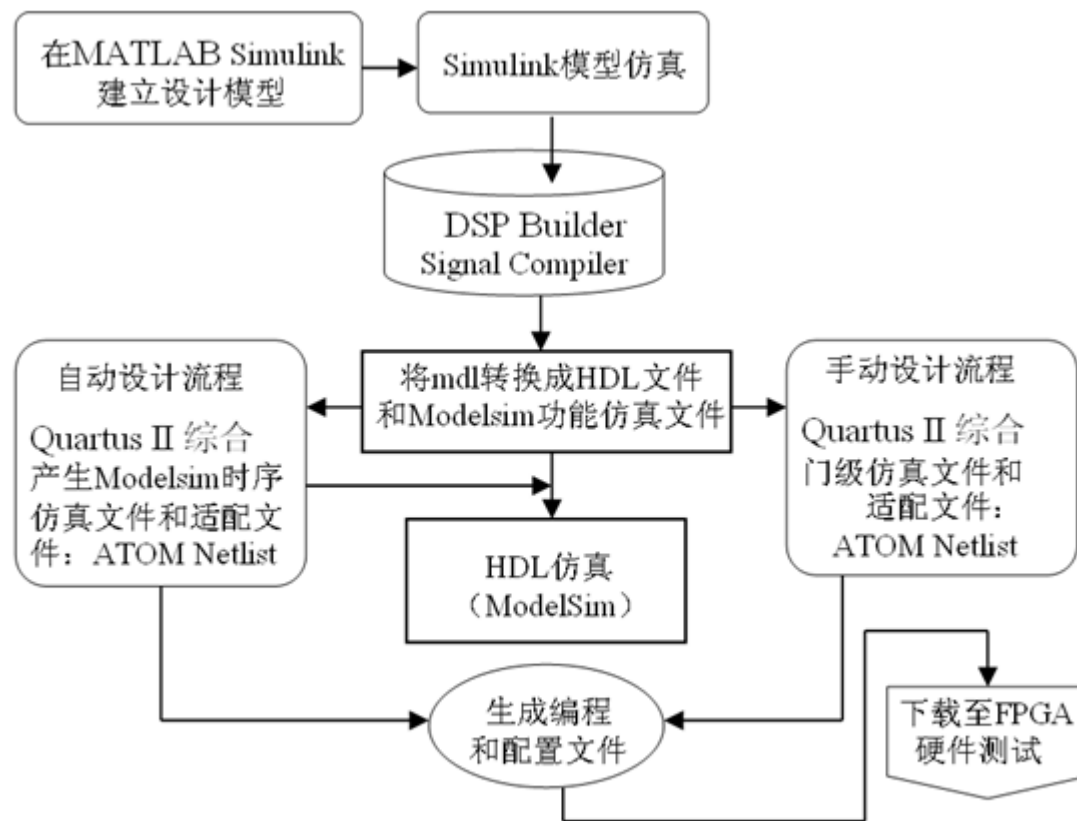


图 11-1 基于 MATLAB、DSP Builder 和 Quartus II 等工具完成设计的流程图

# 11.2 正弦信号发生器设计

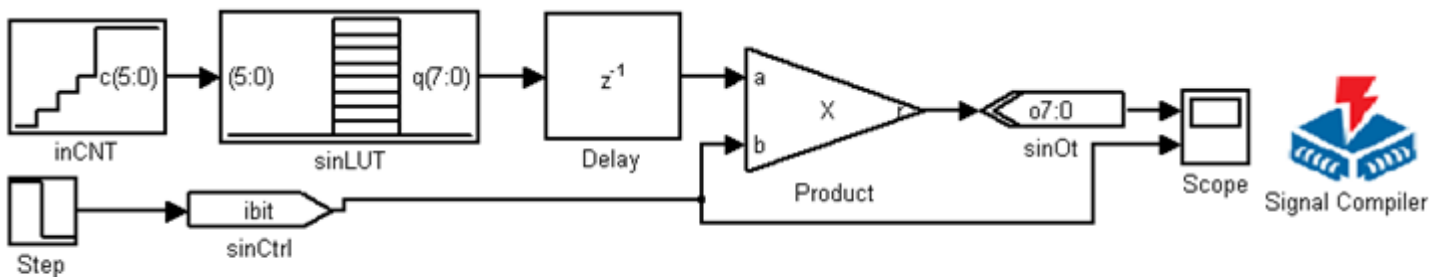


图 11-2 正弦波发生模块原理图

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 1. 打开MATLAB环境

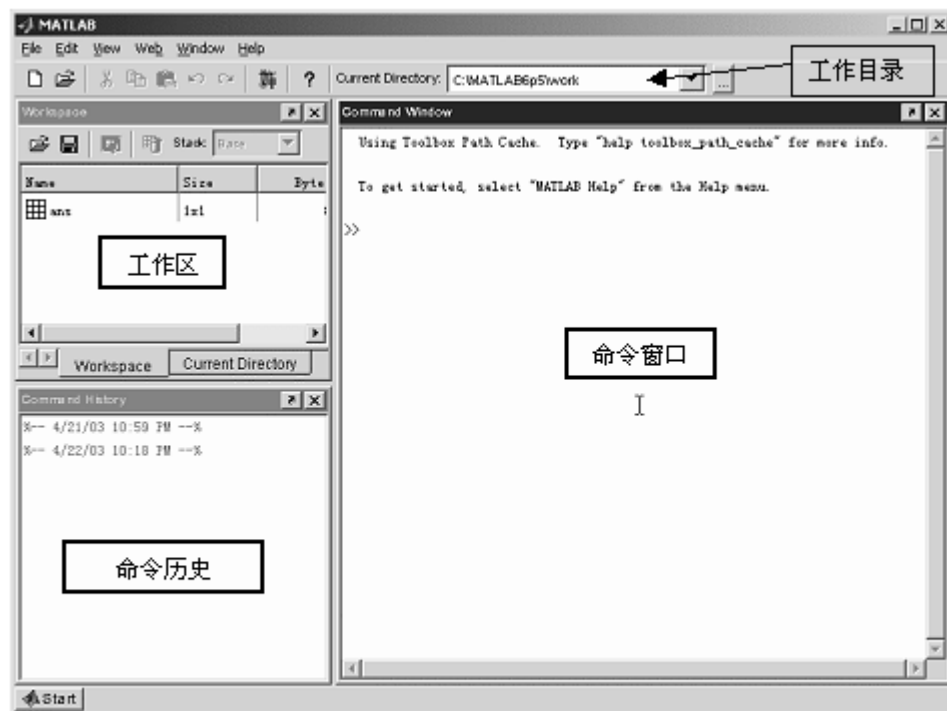


图 11-3 MATLAB 界面

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

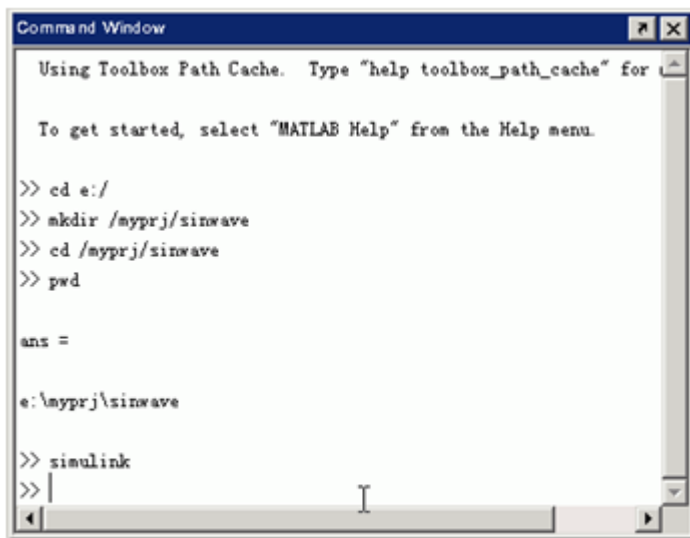
### 2. 建立工作库

```
cd e:/  
    mkdir /myprj/sinwave  
cd /myprj/sinwave
```

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 3. 了解Simulink库管理器



```
Command Window
Using Toolbox Path Cache. Type "help toolbox_path_cache" for
To get started, select "MATLAB Help" from the Help menu.

>> cd e:/
>> mkdir /myprj/sinwave
>> cd /myprj/sinwave
>> pwd

ans =

e:\myprj\sinwave

>> simulink
>> |
```

图 11-4 打开 Simulink

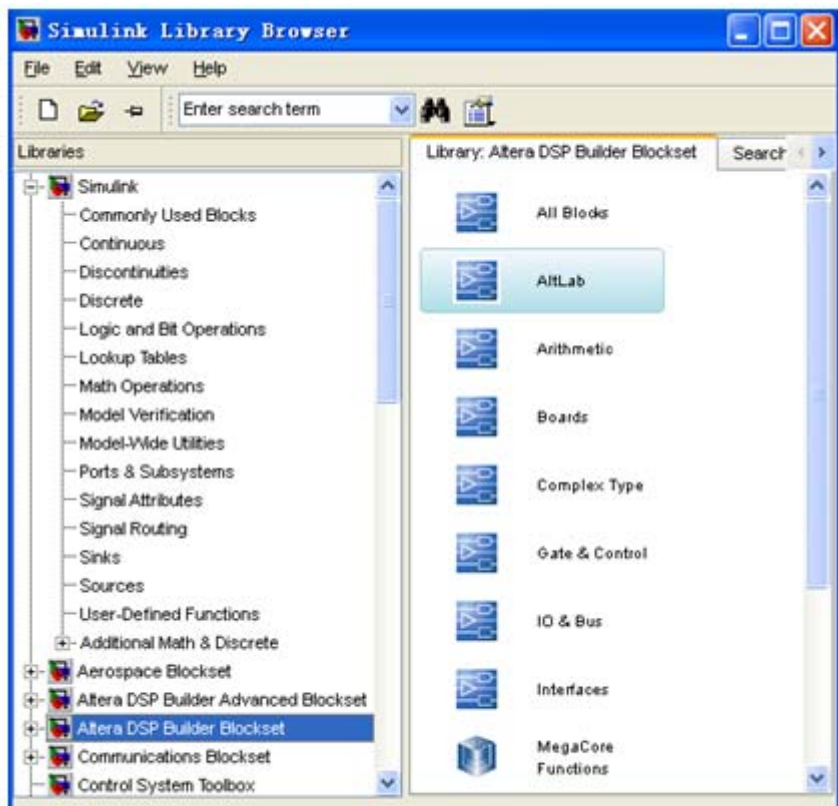


图 11-5 Simulink 库管理器

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 4. Simulink的模型文件



图 11-6 建立新模型

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 5. 放置SignalCompiler

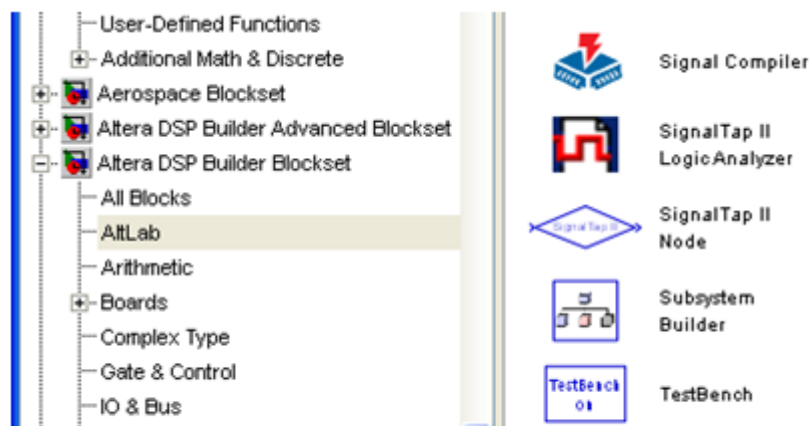


图 11-7 放置 SignalCompiler



# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 6. 放置Increment Decrement

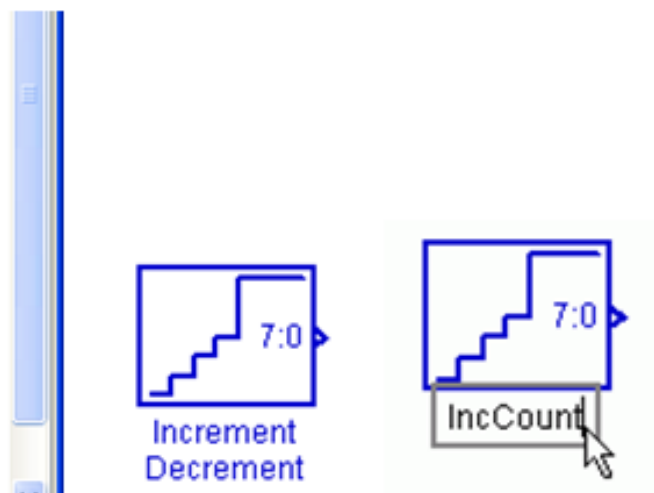


图 11-8 递增递减模块改名为 IncCount

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 7. 设置IncCount

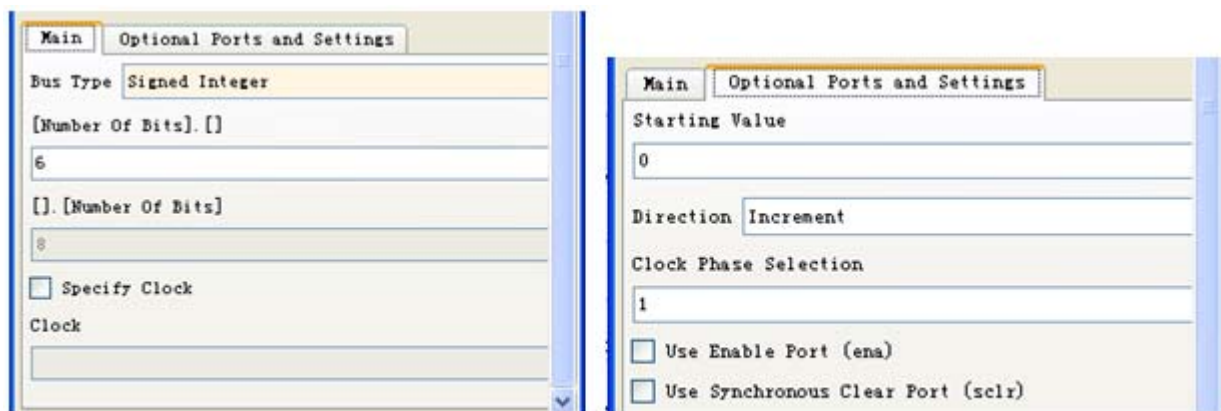


图 11-9 设置递增递减模块



图 11-10 LUT 模块

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 8. 放置正弦查找表 (SinLUT)

`sin([起始值:步进值:结束值])`

`127*sin(0:2*pi/(2^6):2*pi)` (11-1)

`511*sin(0:2*pi/(2^6):2*pi) + 512` (11-2)

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 9. 放置Delay模块

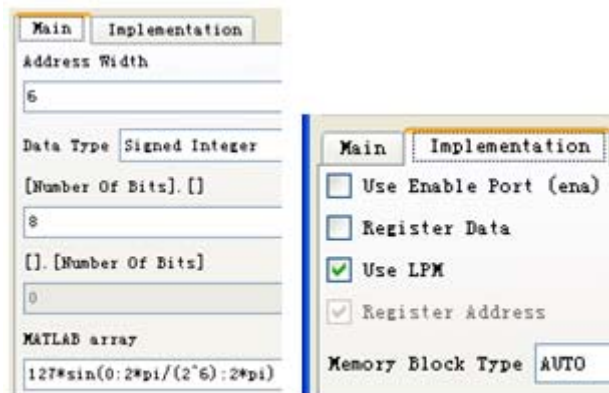


图 11-11 设置 SinLUT

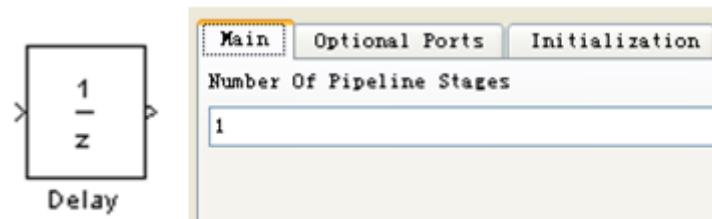


图 11-12 Delay 模块及其参数设置窗口

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 10. 放置端口sinCtrl

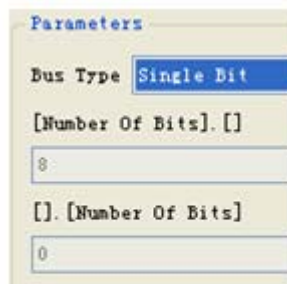


图 11-13 设置参数

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 11. 放置Product模块

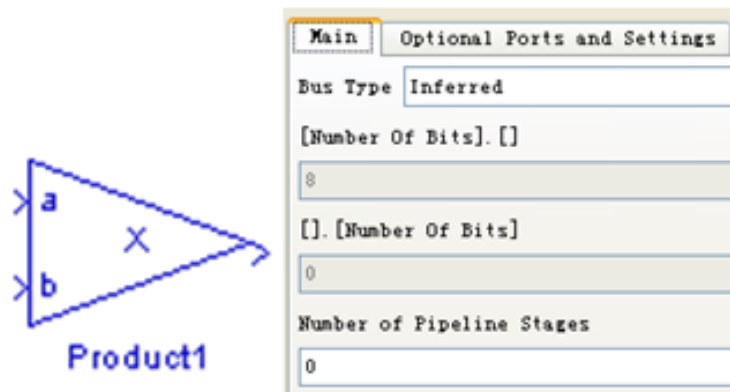


图 11-14 设置乘法单元

# 11.2 正弦信号发生器设计

## 11.2.1 建立设计模型

### 12. 放置输出端口sinOt

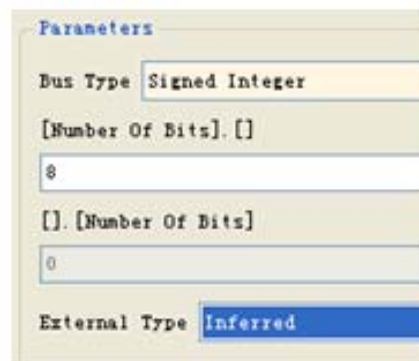


图 11-15 设置输出端口

```
STD_LOGIC_VECTOR(7 DOWNTO 0)
```

### 13. 设计文件存盘

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 1. 加入仿真步进模块

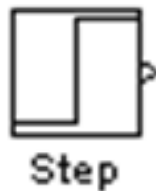


图 11-16 Step 模块



# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 2. 添加波形观察模块并设置参数



图 11-17 Scope 模型

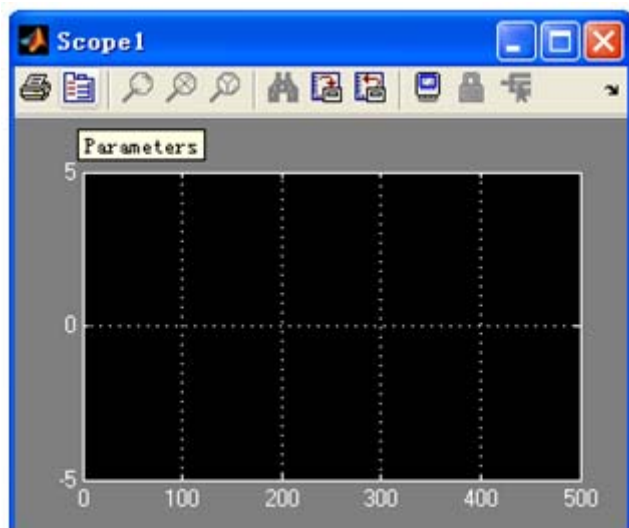


图 11-18 Scope 初始显示

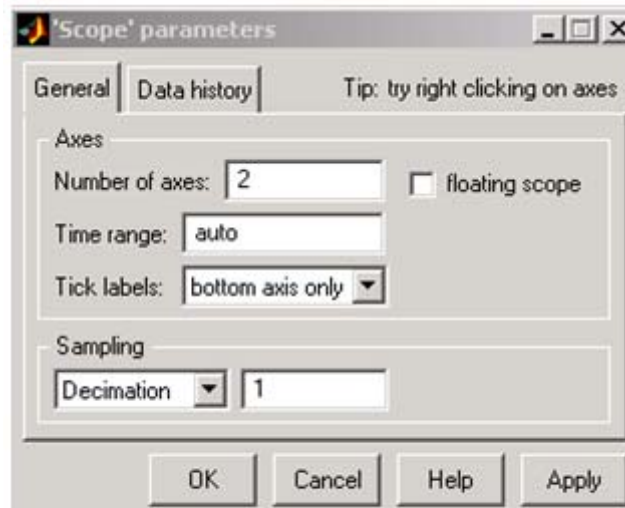


图 11-19 设置 Scope 参数

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 3. 设置仿真激励



图 11-20 设置 Step

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 3. 设置仿真激励



图 11-21 Simulink 仿真设置

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 4. 启动仿真

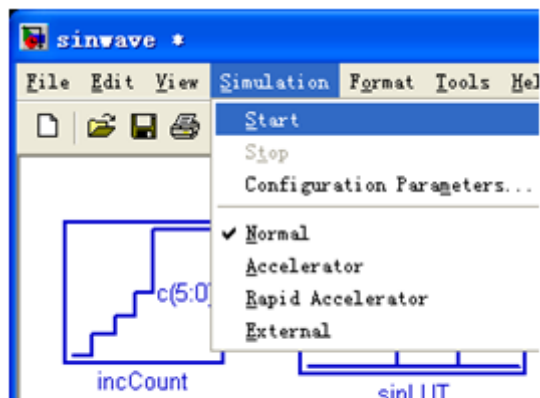


图 11-22 Simulink 仿真开始

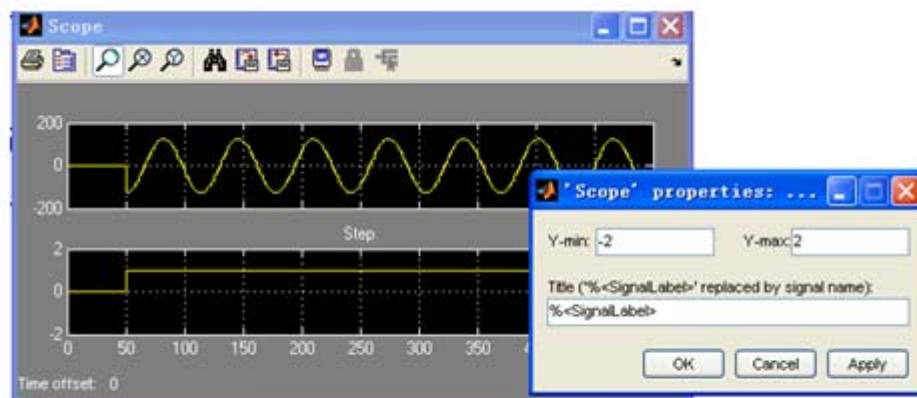


图 11-23 Scope 模块输出波形

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 5. 设计成无符号数据输出

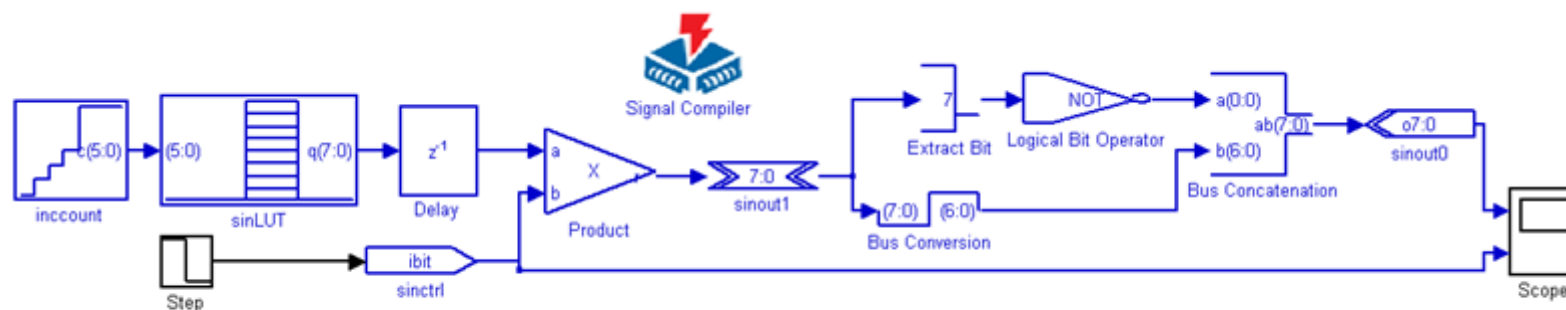


图 11-24 有符号输出改为无符号输出电路

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 6. 各模块功能说明

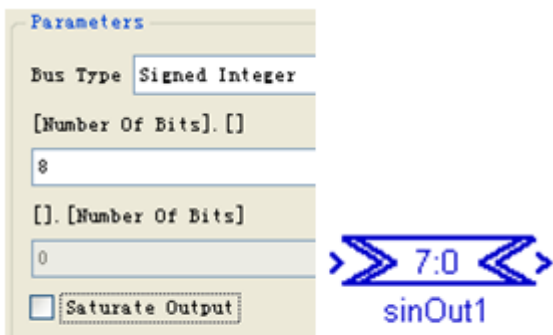


图 11-25 AltBus 模块 SinOut1 的设置

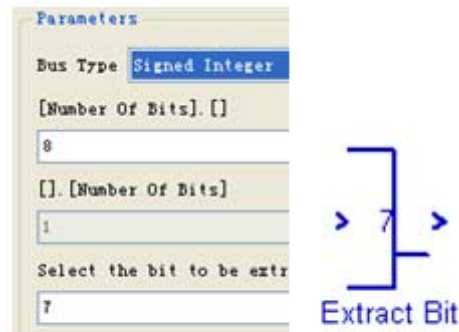


图 11-26 ExtractBit 模块设置

# 11.2 正弦信号发生器设计

## 11.2.2 Simulink模型仿真

### 6. 各模块功能说明

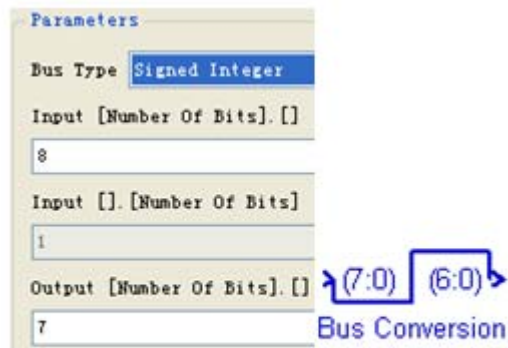


图 11-27 BusConversion 模块设置

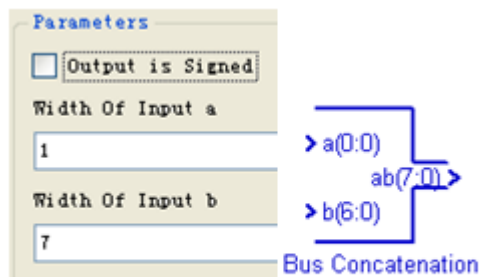


图 11-28 BusConcatenation 模块设置

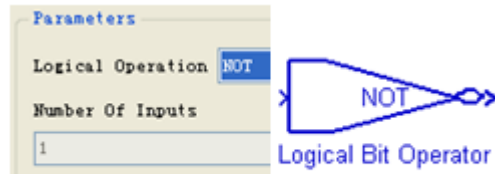


图 11-29 NOT 模块设置

# 11.2 正弦信号发生器设计

## 11.2.3 SignalCompiler使用方法

### 1. 分析当前的模型

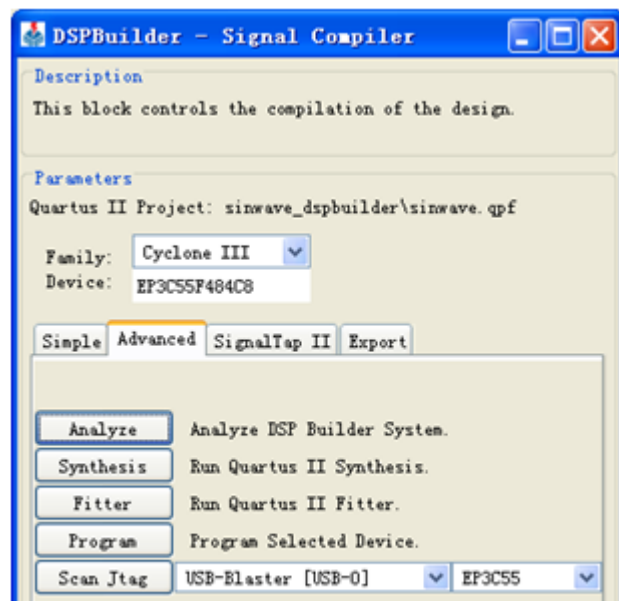


图 11-30 SignalCompiler 界面



# 11.2 正弦信号发生器设计

## 11.2.3 SignalCompiler使用方法

### 2. 设置SignalCompiler

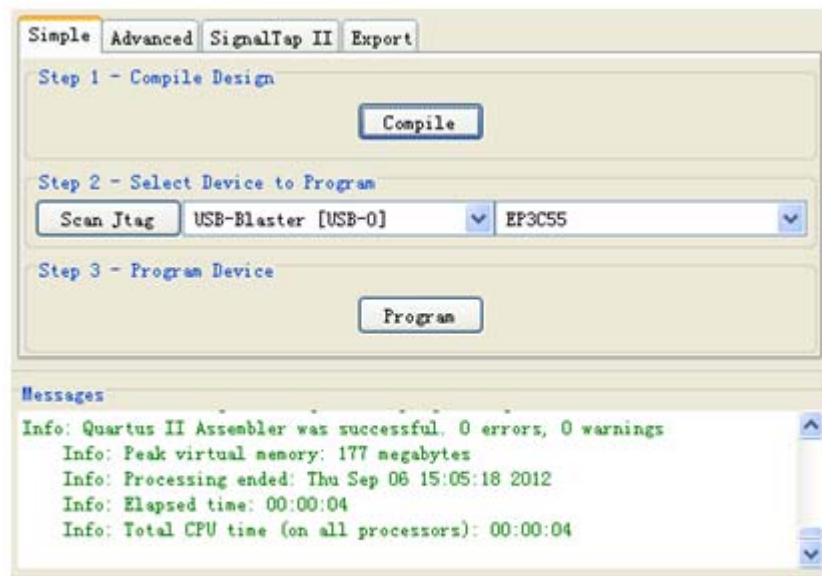


图 11-31 Sinout 工程处理信息

# 11.2 正弦信号发生器设计

## 11.2.4 使用ModelSim进行RTL级仿真

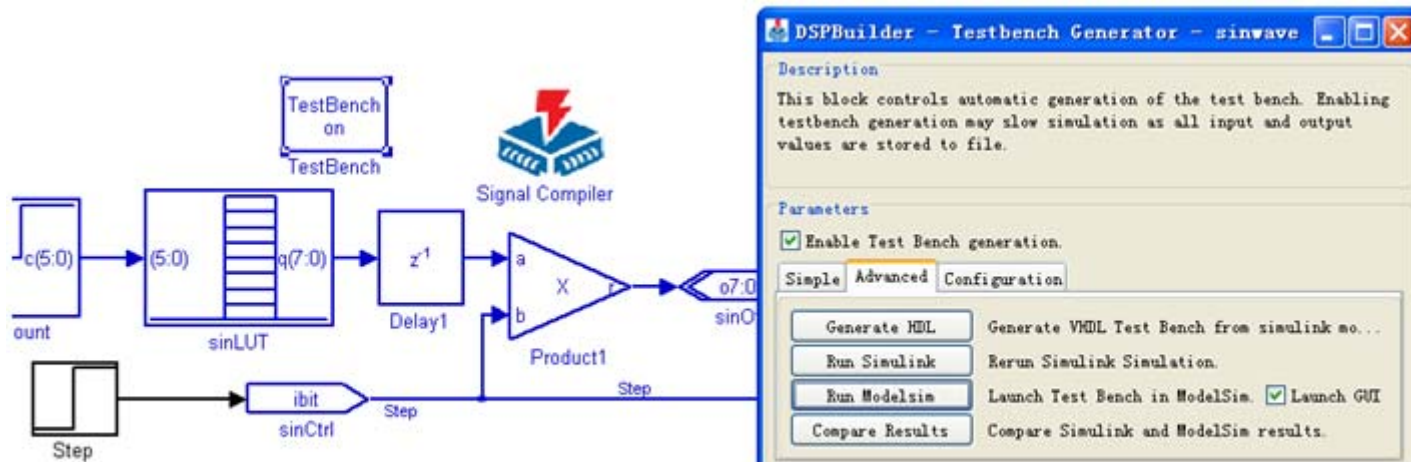


图 11-32 在 Simout 模型中启动 TestBench 模块

# 11.2 正弦信号发生器设计

## 11.2.4 使用ModelSim进行RTL级仿真

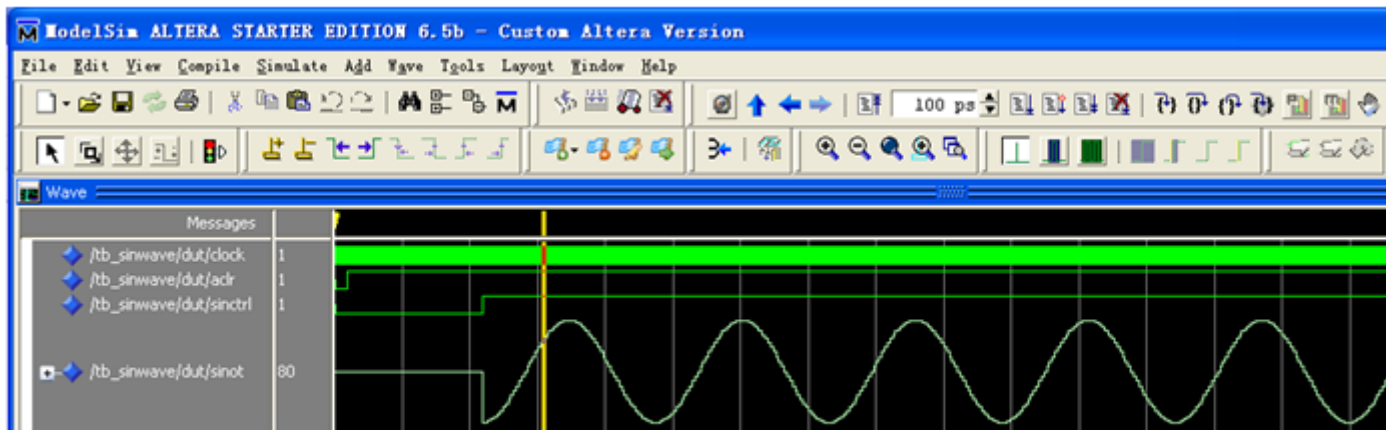


图 11-33 Modelsim 显示仿真结果波形

# 11.2 正弦信号发生器设计

## 11.2.5 使用Quartus II实现时序仿真

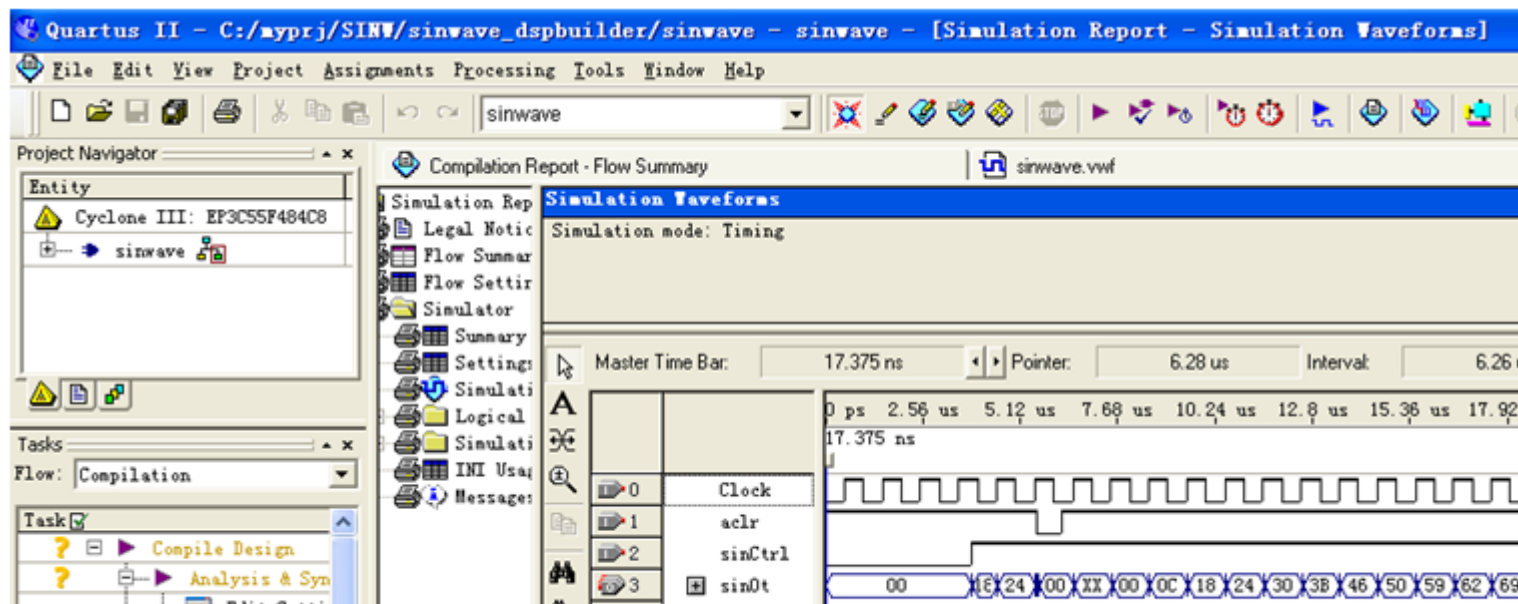


图 11-34 打开 Quartus II 工程进行编译和时序仿真

# 11.2 正弦信号发生器设计

## 11.2.6 硬件测试与硬件实现

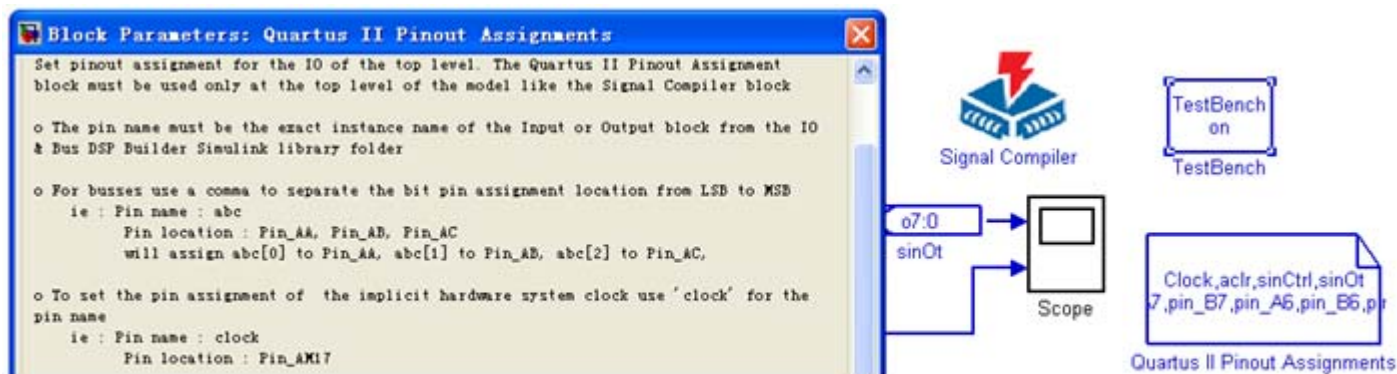


图 11-35 添加 Quartus II Pinout Assignments 模块

# 11.2 正弦信号发生器设计

## 11.2.6 硬件测试与硬件实现

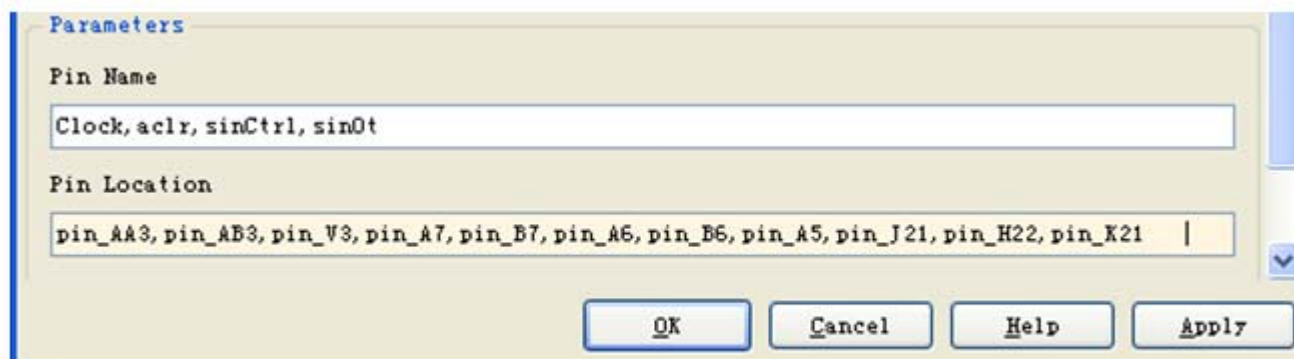


图 11-36 Quartus II Pinout Assignments 模块分配引脚

# 11.3 DSP Builder 层次化设计

## 11.2.6 硬件测试与硬件实现

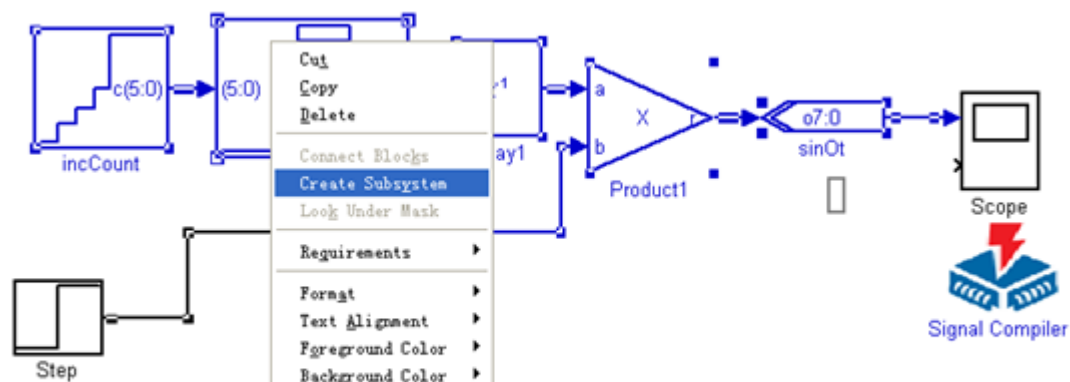


图 11-37 准备建立 Subsystem

# 11.3 DSP Builder 层次化设计

## 11.2.6 硬件测试与硬件实现

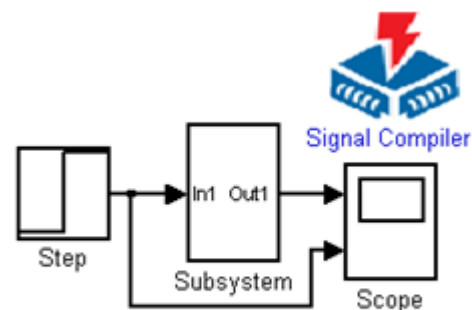


图 11-38 建立 Subsystem 后

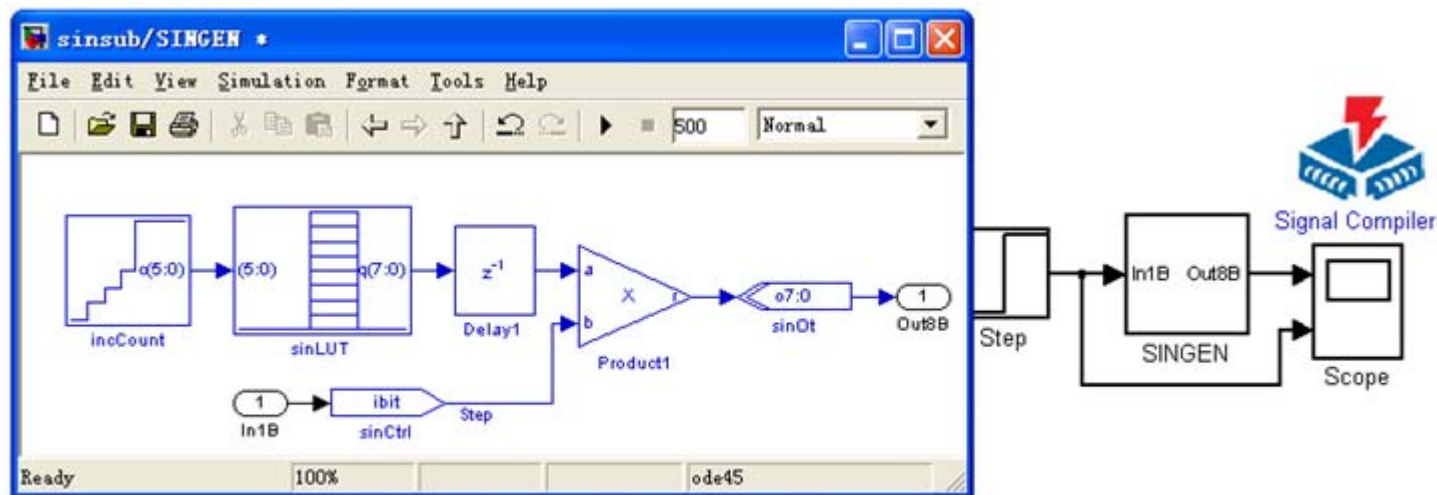


图 11-39 subint/Subsystem 子系统图



# 11.3 DSP Builder 层次化设计

## 11.2.6 硬件测试与硬件实现

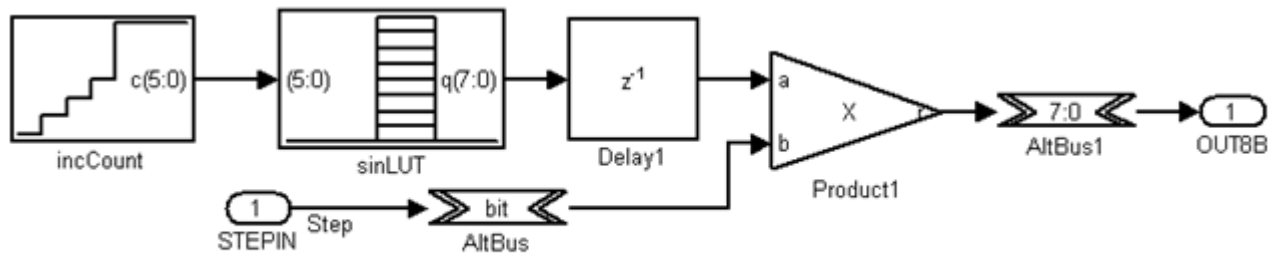


图 11-40 更改了端口和端口名的子系统图

# 11.3 DSP Builder 层次化设计

## 11.2.6 硬件测试与硬件实现

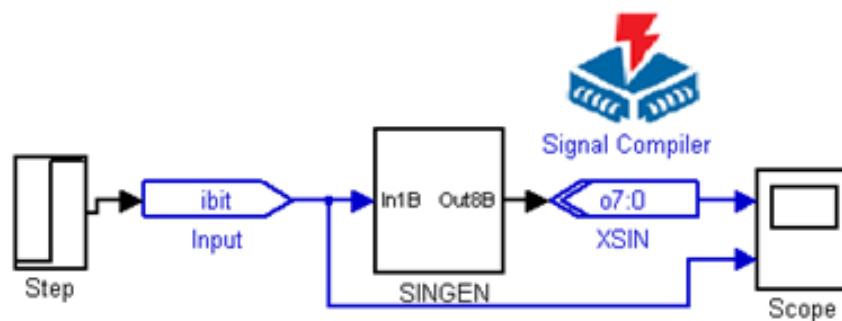


图 11-41 顶层设计图

- 1) Altbody 模块 (Altbody 模块)  
库: Altera DSP Builder 中 IO&Bus 库  
参数 Bus Type 设为 Signed Integer; 参数 number of bits 设为 8; 其余为 0。
- 2) Adder 模块 (Parallel Adder Subtractor 模块)  
库: Altera DSP Builder 中 Arithmetic 库  
参数 Add(+)Sub(-)设为 +; 参数 Number of Inputs 设: 2  
Clock Phase Selection 设 1; 使用 Enable Pipeline
- 3) xsin 模块 (Output 模块)  
库: Altera DSP Builder 中 IO&Bus 库  
参数 Bus Type 设为 Unsigned Integer  
参数 number of bits 设为 8; External Type 选择 Inferred
- 4) Input 模块 (Input 模块)  
库: Altera DSP Builder 中 IO&Bus 库; 参数 Bus Type 设为 Single Bit
- 5) Constant 模块 (Constant 模块)  
库: Altera DSP Builder 中 IO&Bus 库  
参数 Bus Type 设为 Signed Integer; 参数 Constant Value 设为 127;  
其余默认;
- 6) Mux 模块  
库: simulink 中 Signal Routing 库  
参数 Number of Inputs 设为 2 ; 参数 Display Option 选择 bar.
- 7) Display 模块; 库: simulink 中 sinks 库

# 11.3 DSP Builder 层次化设计

## 11.2.6 硬件测试与硬件实现

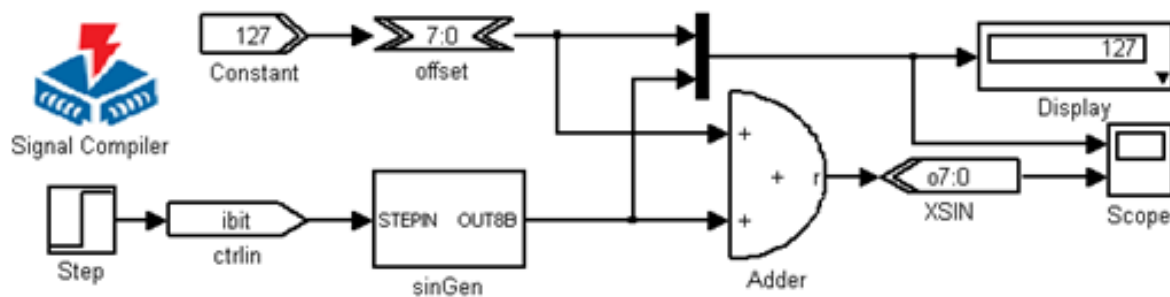


图 11-42 含 Subsystem 的新的 subint 模型

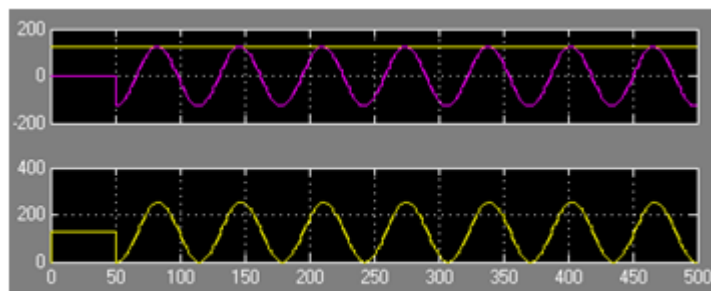


图 11-43 新的 subint 模型的仿真波形图

# 11.4 基于DSP Builder的DDS设计

## 11.4.1 DDS模块设计

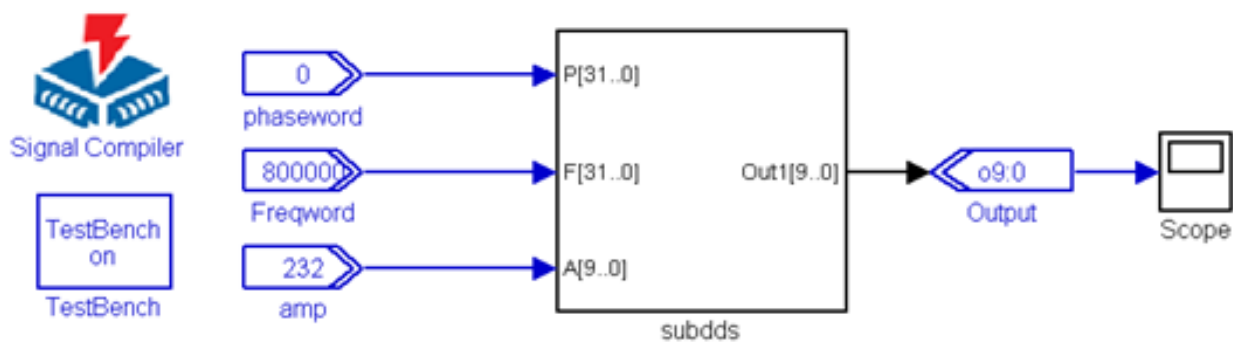


图 11-44 DDS 系统

# 11.4 基于DSP Builder的DDS设计

## 11.4.1 DDS模块设计

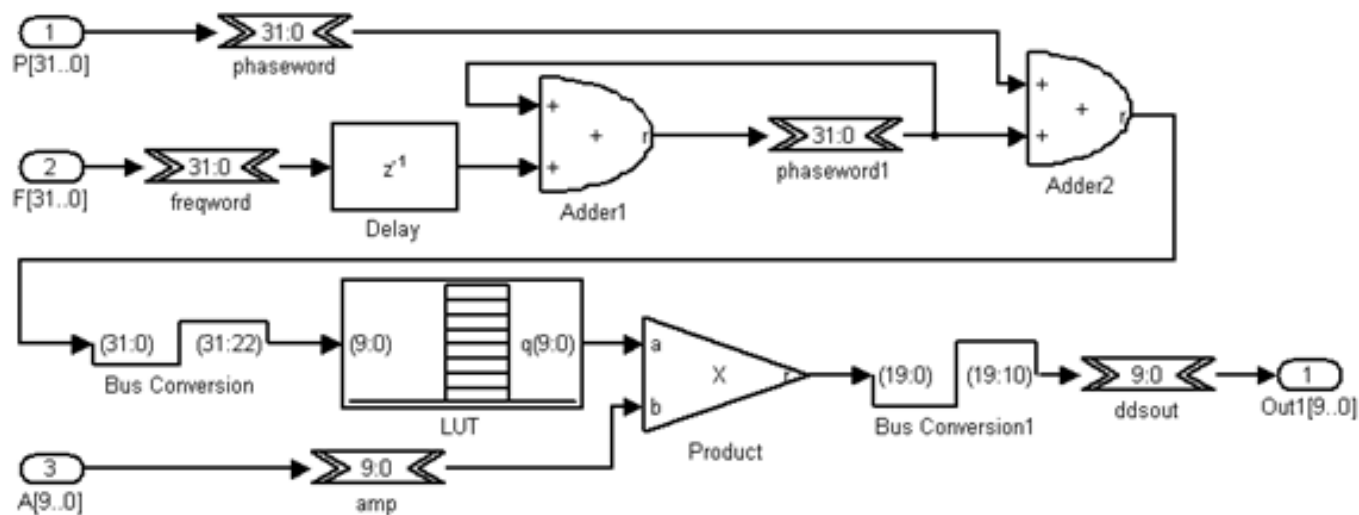


图 11-45 DDS子系统 SubDDS

# 11.4 基于DSP Builder的DDS设计

## 11.4.1 DDS模块设计

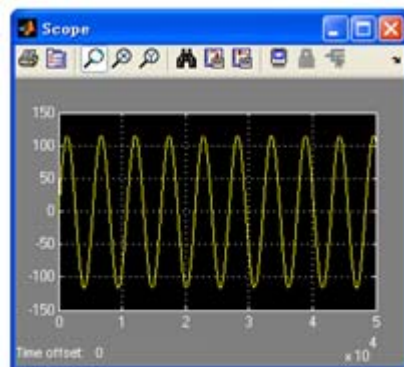


图 11-46 DDS 模型输出波形

1) freqword模块 (Altbus模块)

库: Altera DSP Builder 中 IO&Bus 库

参数 Bus Type 设为 Signed Integer

参数 number of bits 设为 32

2) Phaseword 模块: 与 freqword 模块相同

3) Amp 模块 (Altbus): 与 freqword 模块相同, 但参数 number of bits 设为 10

4) ddsout 模块 (Altbus): 与 Amp 模块相同

# 11.4 基于DSP Builder的DDS设计

## 11.4.1 DDS模块设计

- 1) Adder1模块 (Parallel Adder Subtractor模块)
    - 库: Altera DSP Builder 中 Arithmetic 库
    - 参数 Number of Inputs 设为 2; Add(+) Sub(-) 设为 ++
    - 选择 Enable Pipeline ; 参数 Clock Phase Selection 设为 1
  - 2) Delay 模块 (Delay 模块)
    - 库: Altera DSP Builder 中 Storage 库
    - 参数 Depth 设为 1 ; 参数 Clock Phase Selection 设为 1
  - 3) Phaseword1模块: 设置与freqword模块相同
- 
- 1) Adder2模块 (Parallel Adder Subtractor模块) 设置与Adder1相同
  - 2) BusConversion 模块 (BusConversion 模块)
    - 库: Altera DSP Builder 中 IO&Bus 库
    - 参数 Input Bus Type 设为 Signed Integer;
    - 参数 Input [number of bits].[ ] 设为 32
    - 参数 Output [number of bits].[ ] 设为 10
    - 参数 Input Bit Connected to Output LSB 设为 22



# 11.4 基于DSP Builder的DDS设计

## 11.4.1 DDS模块设计

### 1) Product模块 (Product模块)

库: Altera DSP Builder 中 Arithmetic 库

参数 Number of Pipeline Stages 设为 2

参数 Clock Phase Selection 设为 1; 选择 Use LPM

### 2) BusConversion1模块 (BusConversion 模块)

库: Altera DSP Builder 中 IO&Bus 库

参数 Input Bus Type 设为 Signed Integer

参数 Input [number of bits].[] 设为 20

参数 Output [number of bits].[] 设为 10

参数 Input Bit Connected to Output LSB 设为 10

### 3) LUT 模块 (LUT 模块)

库: Altera DSP Builder Blockset 中 Storage 库;

参数 Data Type 设为 Signed Integer; 参数 Address Width 设为 10

参数 [number of bits].[] 设为 10

参数 MATLAB Array 设为 “511\*sin([0:2\*pi/(2^10):2\*pi])” ; 使用 Use LPM

# 11.4 基于DSP Builder的DDS设计

## 11.4.2 FSK调制器设计

(1) 直接调频法。

(2) 频率键控法。

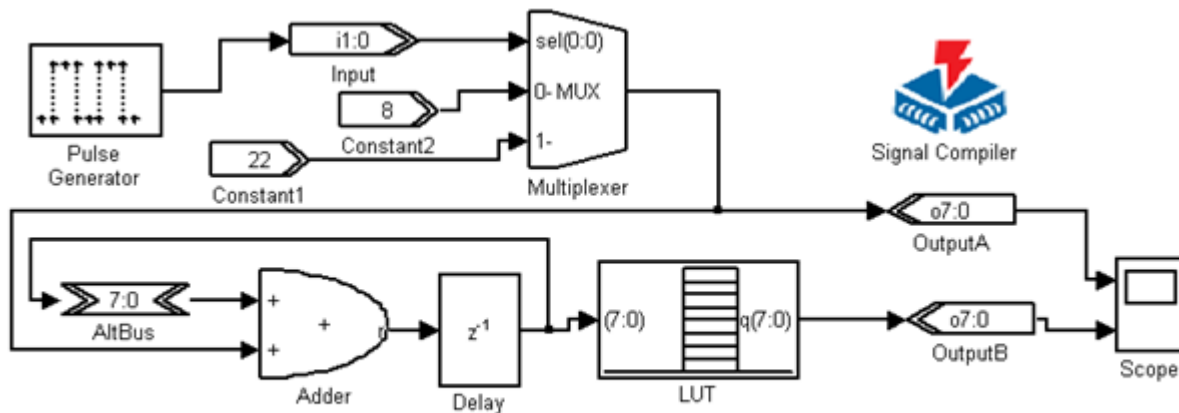


图 11-47 FSK 调制器模型

# 11.4 基于DSP Builder的DDS设计

## 11.4.2 FSK调制器设计

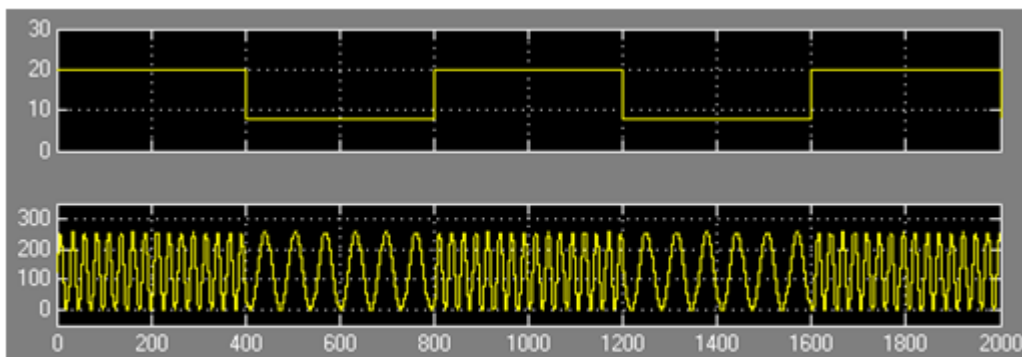


图 11-48 FSK 的 Simulink 仿真结果

# 11.4 基于DSP Builder的DDS设计

## 11.4.3 正交信号发生器设计

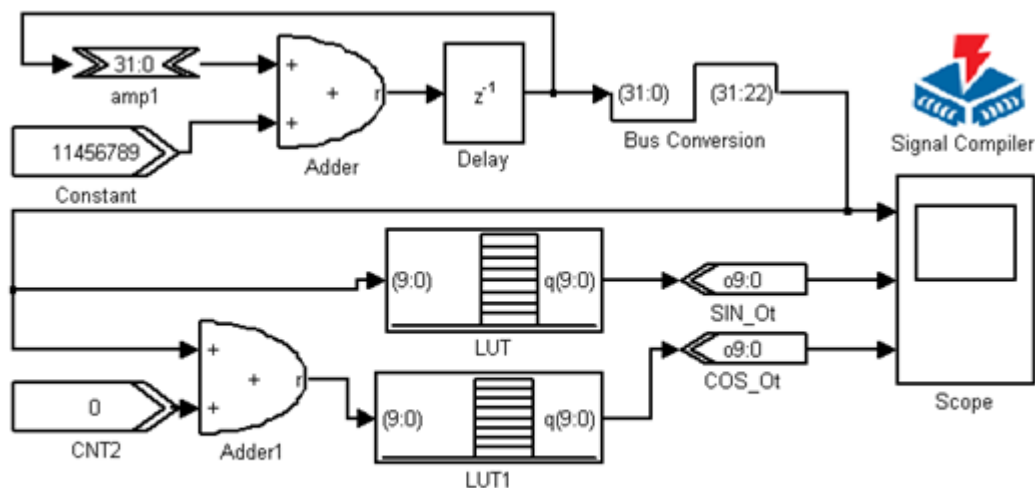


图 11-49 正交信号发生器 MDL 模型

# 11.4 基于DSP Builder的DDS设计

## 11.4.3 正交信号发生器设计

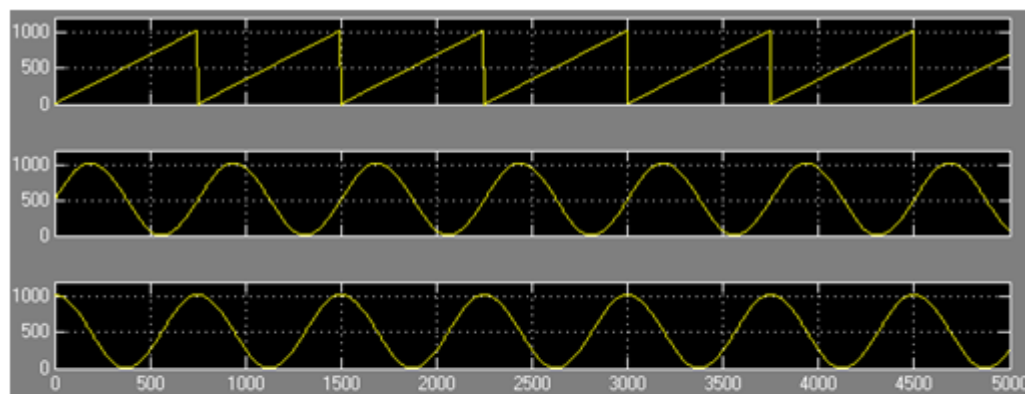


图 11-50 正交信号发生器输出信号波形

# 11.4 基于DSP Builder的DDS设计

## 11.4.4 数控移相信号发生器设计

## 11.4.5 幅度调制信号发生器设计

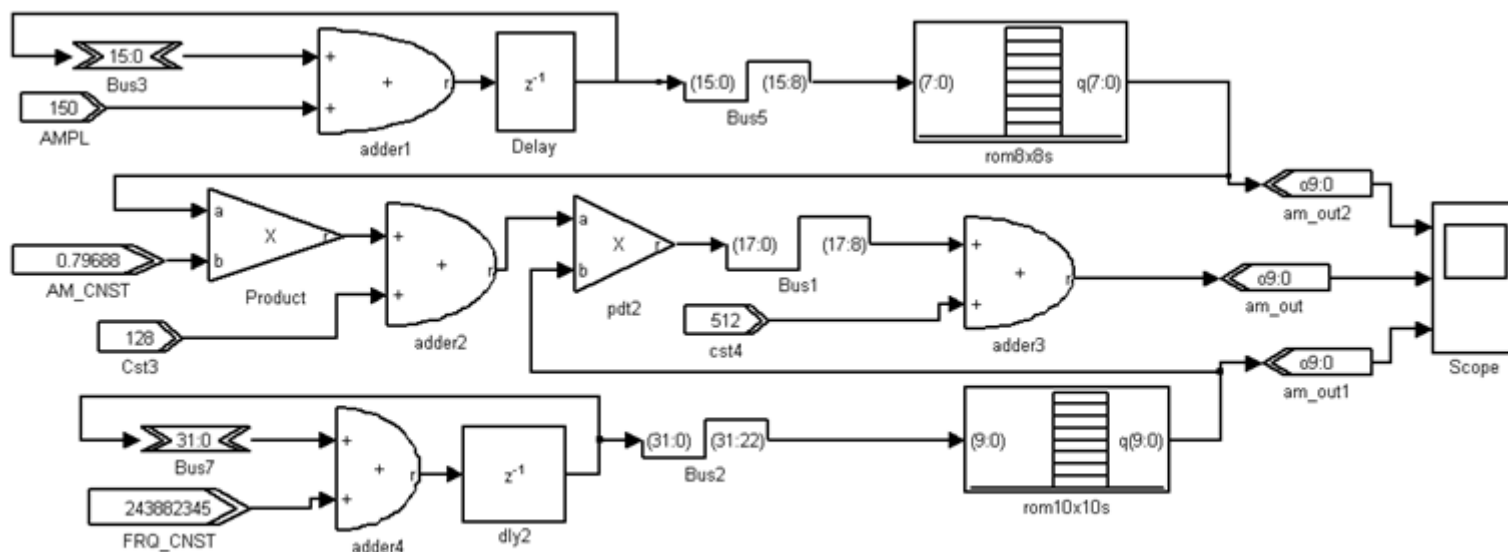


图 11-51 AM 发生器模型

# 11.4 基于DSP Builder的DDS设计

## 11.4.5 幅度调制信号发生器设计

$$F = F_{dr} \cdot (1 + F_{am} \cdot m) \quad (11-3)$$

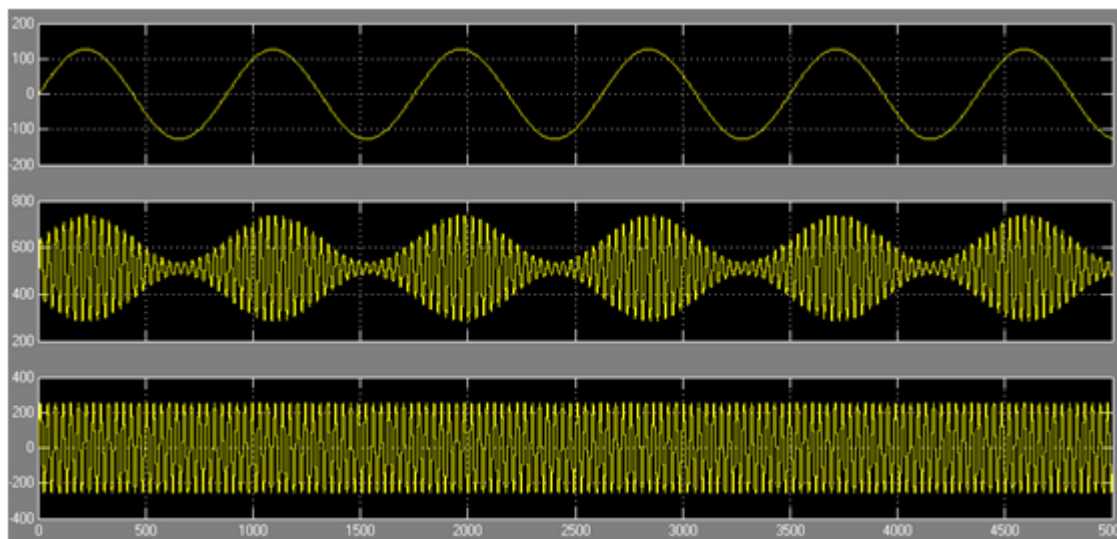


图 11-52 AM 模型仿真波形

# 11.5 数字编码与译码器设计

## 11.5.1 伪随机序列

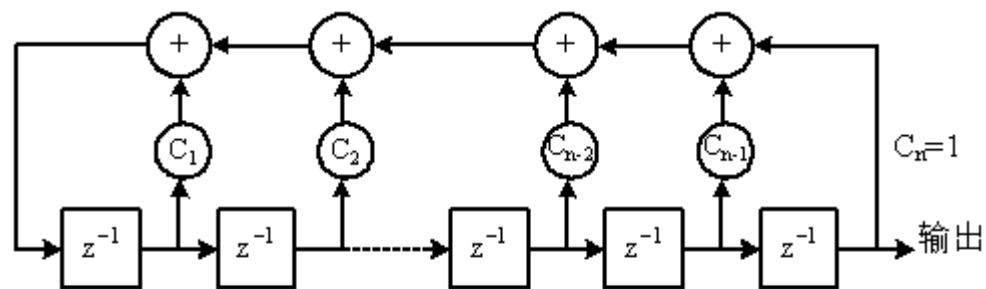


图 11-53 线性反馈移位寄存器的构成

$$F(x) = \sum_{i=0}^n C_i x^i \quad (11-4)$$

$$x^5 + x^2 + 1 \quad (11-5)$$



# 11.5 数字编码与译码器设计

## 11.5.1 伪随机序列

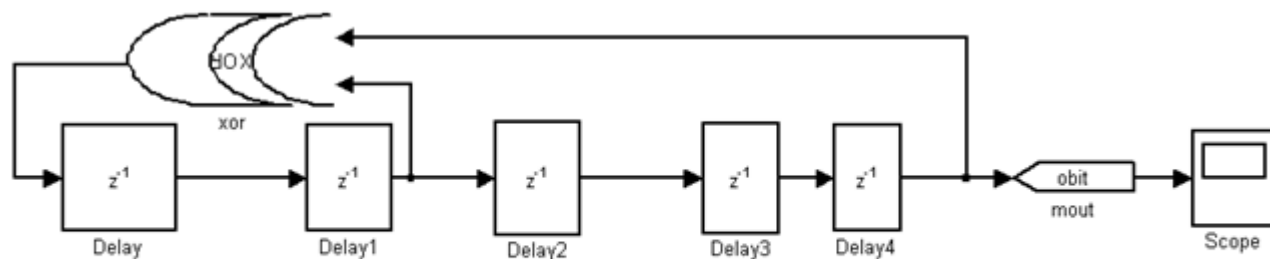


图 11-54 m 序列发生器模型

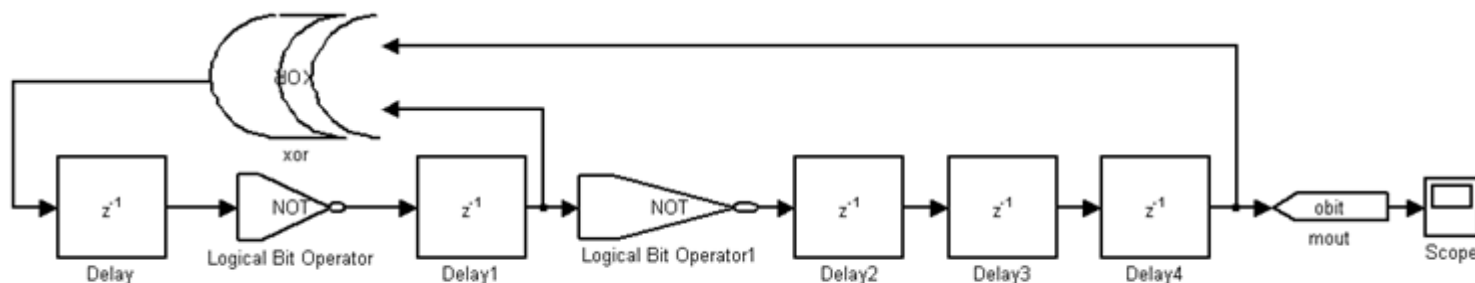


图 11-55 修改后的 m 序列发生器模型

# 11.5 数字编码与译码器设计

## 11.5.1 伪随机序列

Xor 模块: (Logical Bit Operator)

库: Altera DSP Builder 中 Gate & Control 库

参数 “Logical Operator” 设为 “xor”; 参数 “Number of Inputs” 设为 “2”

Not、Not1 模块: (Logical Bit Operator)

库: Altera DSP Builder 中 Gate & Control 库; 参数 “Logical Operator” 设为 “not”

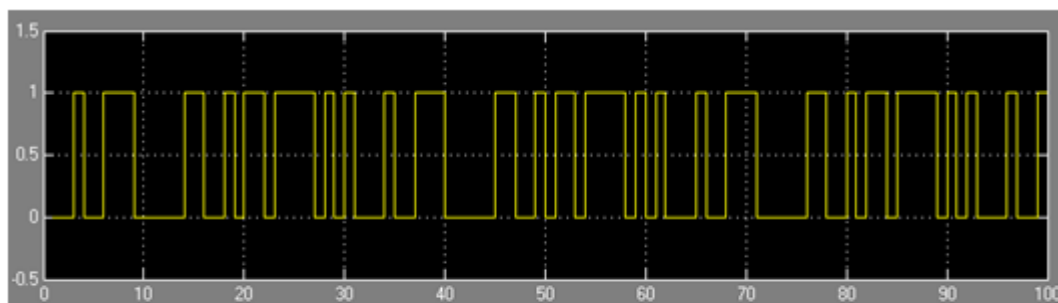


图 11-56 m 序列发生器 Simulink 仿真结果

# 11.5 数字编码与译码器设计

## 11.5.2 帧同步检出

$$R(j) = \sum_{i=1}^{n-j} x_i x_{x+j} = \begin{cases} n, & j=0 \\ 0, \pm 1, & 0 < j < n \\ 0, & j \geq n \end{cases} \quad (11-6)$$

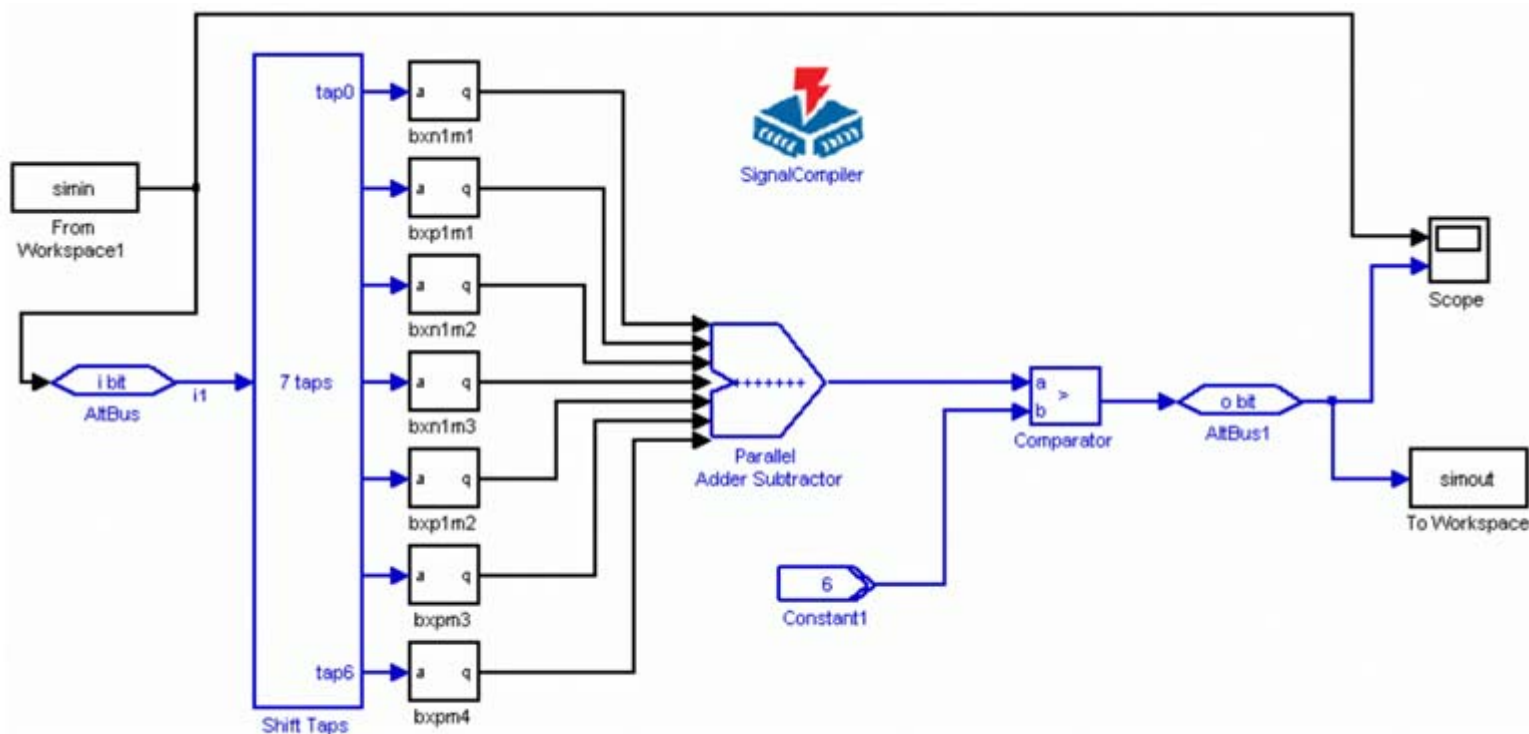


图 11-57 帧同步检出模型

# 11.5 数字编码与译码器设计

## 11.5.2 帧同步检出

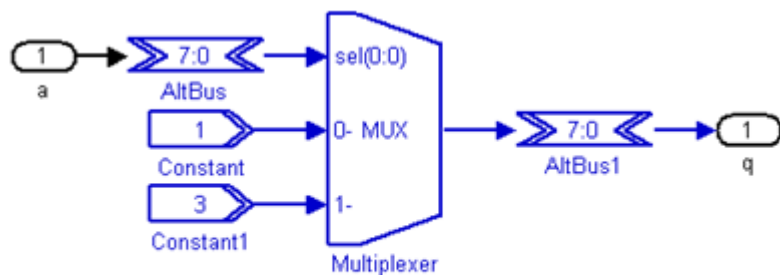


图 11-58 子系统 bxn1m

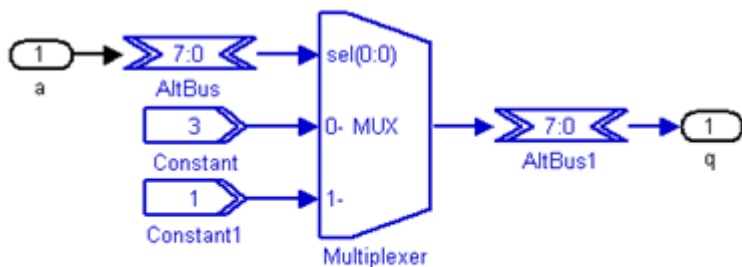


图 11-59 子系统 bxp1m

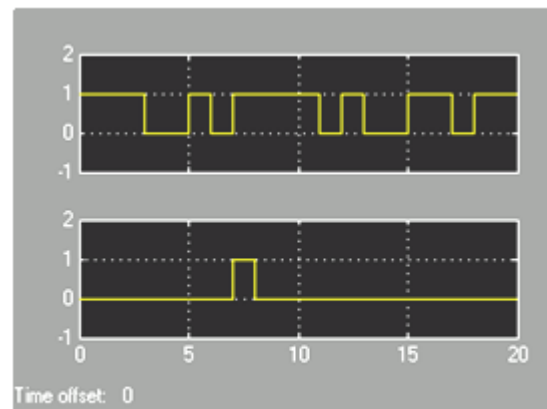


图 11-60 帧同步的巴克码检测仿真结果

# 11.6 HIL 硬件仿真

## 11.6.1 HIL 仿真流程

### 1. 首先完成一个 Simulink 模型设计

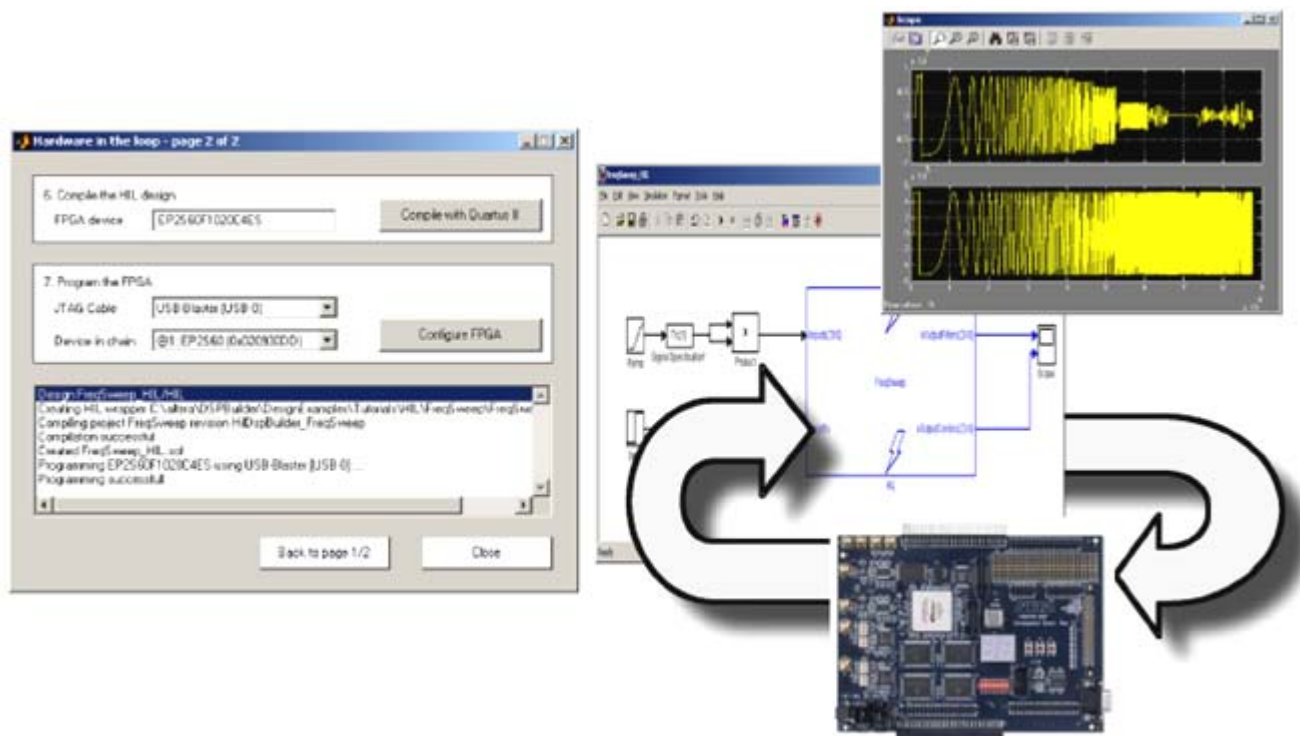
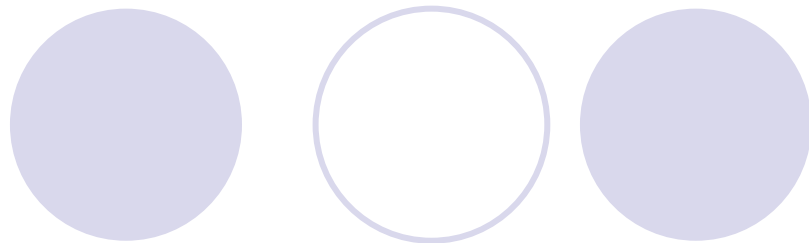


图 11-61 插入 HIL 的 Simulink 模型硬件仿真说明图

# 11.6 HIL硬件仿真



## 11.6.1 HIL仿真流程

### 1. 首先完成一个Simulink模型设计

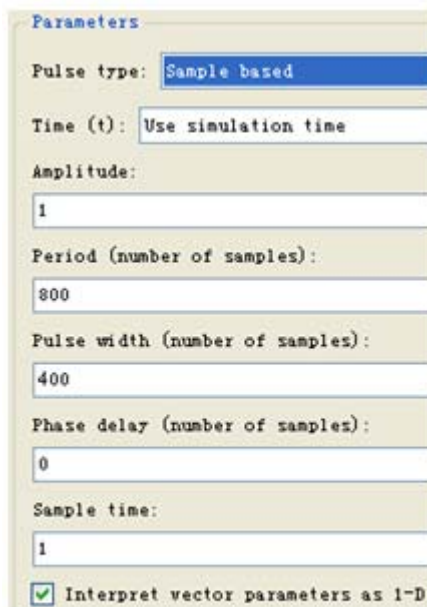


图 11-62 脉冲发生器设置



图 11-63 Simulink 仿真参数设置

# 11.6 HIL硬件仿真

## 11.6.1 HIL仿真流程

1. 首先完成一个**Simulink**模型设计

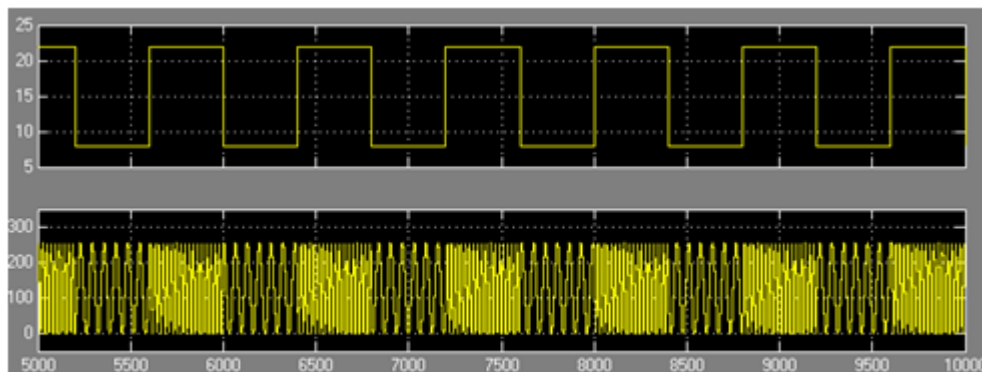


图 11-64 FSK 发生器 Simulink（软件）仿真波形

# 11.6 HIL硬件仿真

## 11.6.1 HIL仿真流程

### 2. 通过DSP Builder转化成Quartus II的工程



图 11-65 SignalCompiler 对 FSK 信号发生器进行转换、综合和适配



# 11.6 HIL硬件仿真

## 11.6.1 HIL仿真流程

### 3. 用HIL模块取代设计模型的所有电路

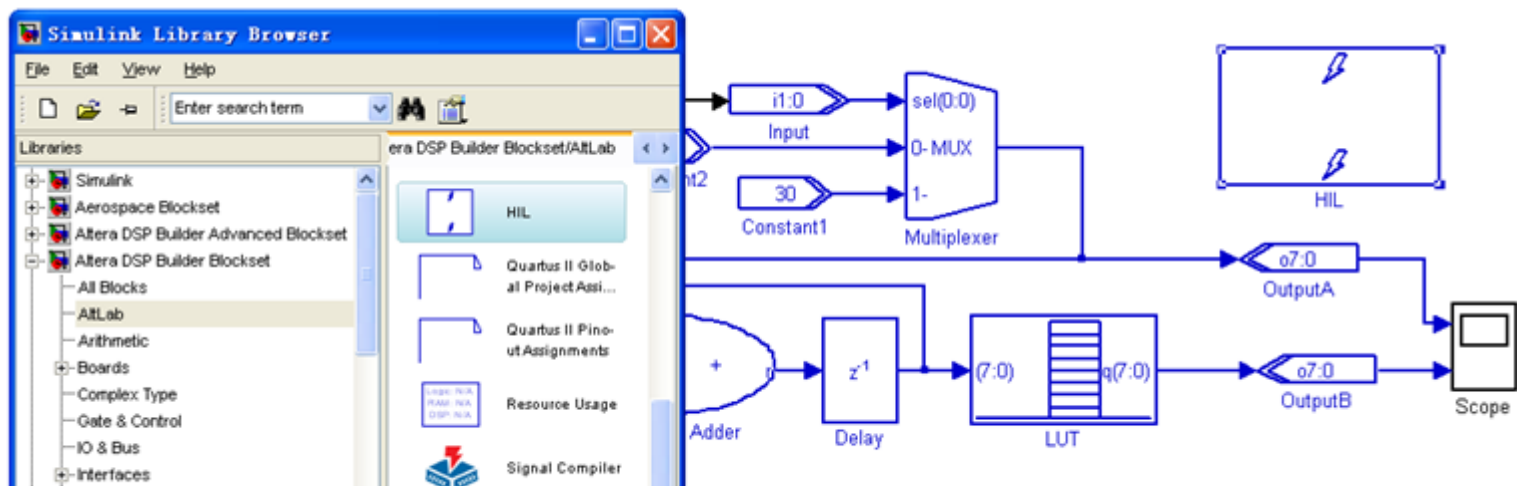


图 11-66 向 Simulink 图中拖入 HIL 模块

# 11.6 HIL硬件仿真

## 4. HIL模块参数设置

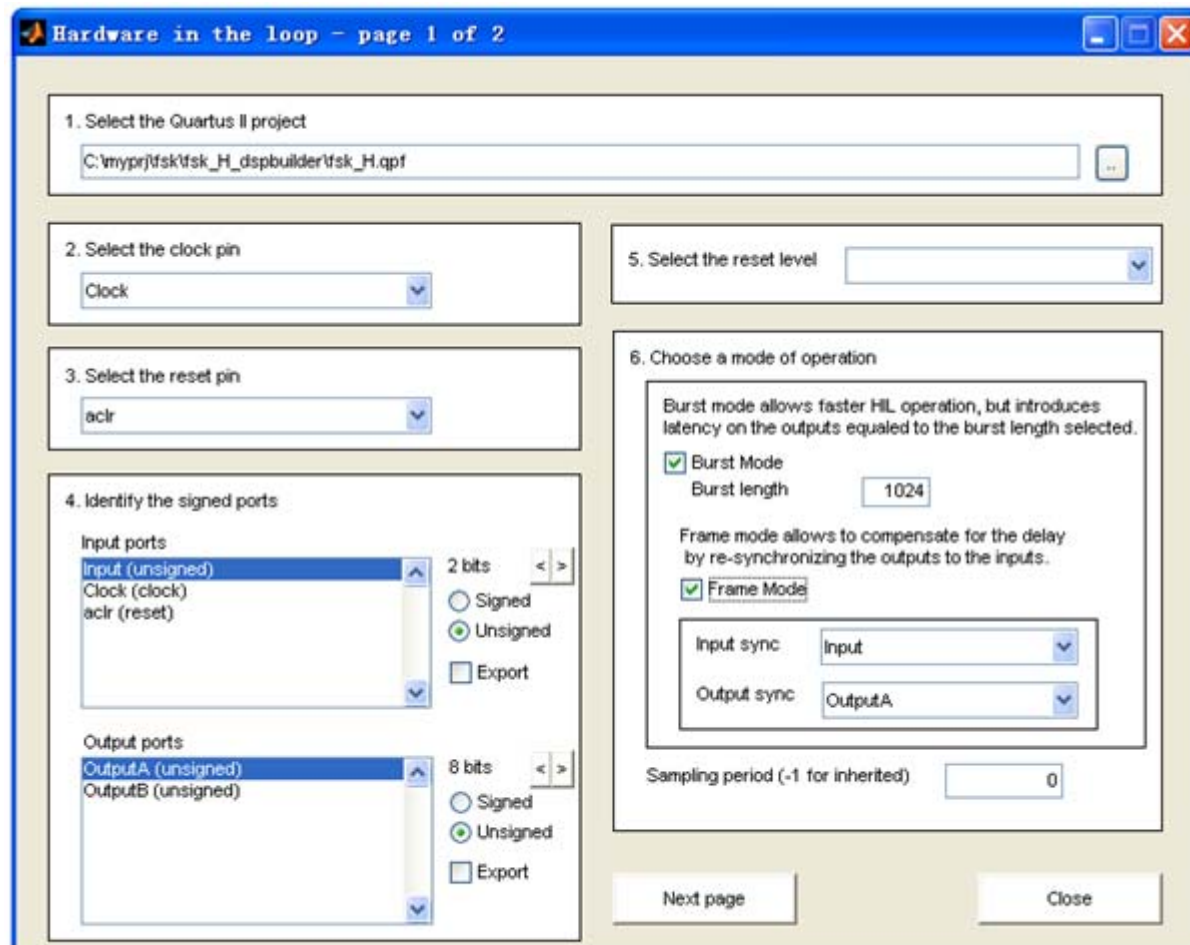


图 11-67 HIL 模块工程加载与参数设置窗口

# 11.6 HIL硬件仿真

## 4. HIL模块参数设置

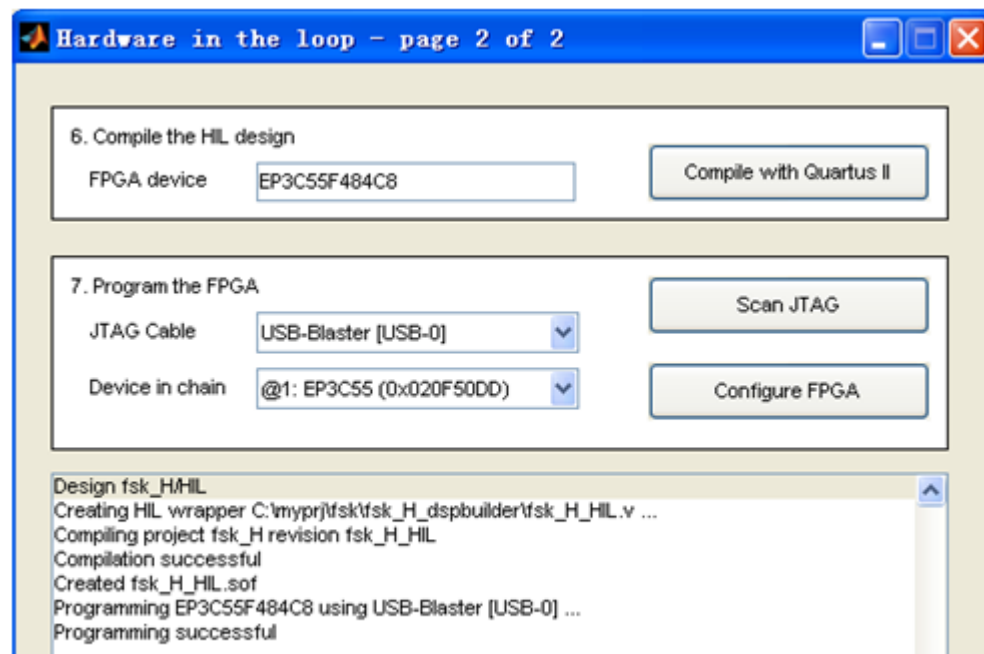


图 11-68 HIL 模块编译与编程窗口

# 11.6 HIL硬件仿真

## 11.6.1 HIL仿真流程

### 5. 进行HIL硬件仿真

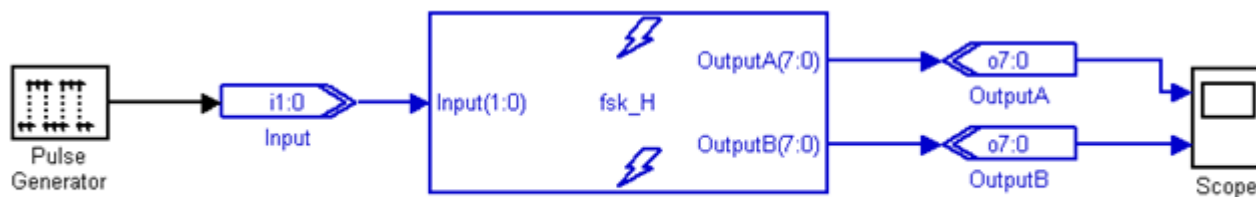


图 11-69 加入了 HIL 模块的扫频滤波电路模型

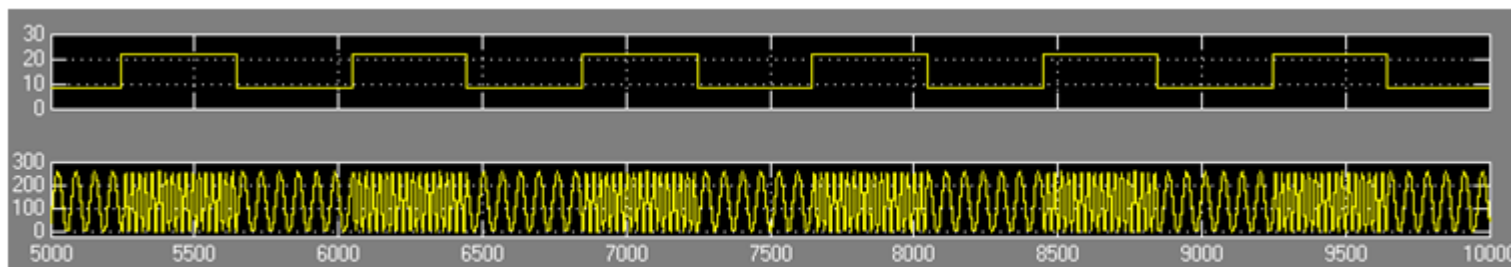


图 11-70 加入了 HIL 模块的扫频滤波电路模型的硬件仿真波形

# 实验与设计

## 11-1 利用MATLAB/DSP Builder设计基本电路模块

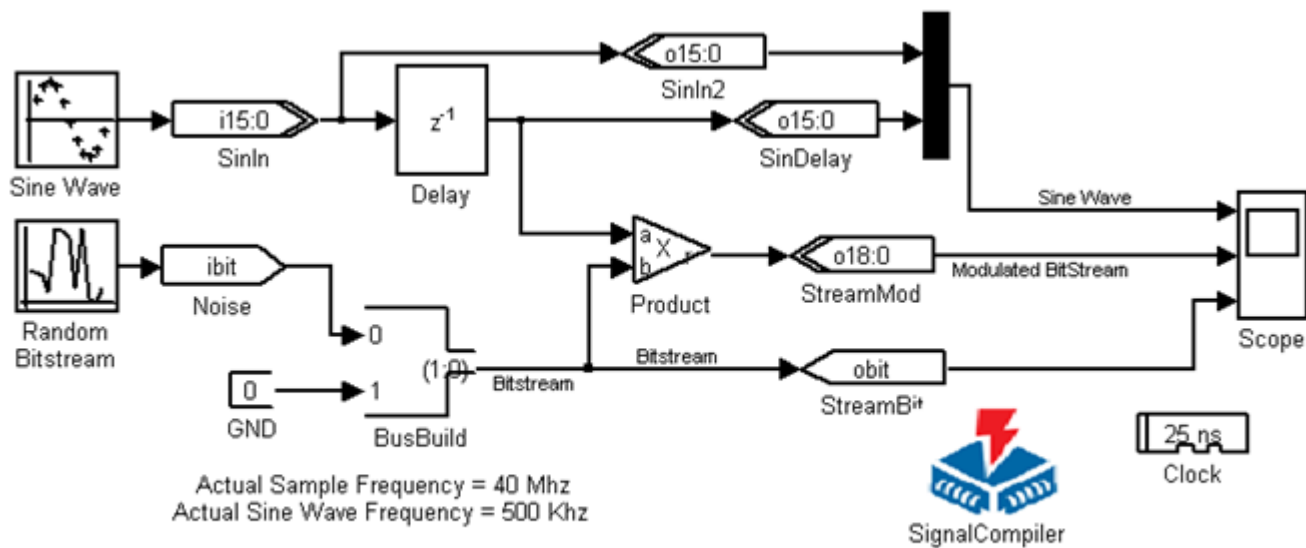


图 11-71 正弦调制信号模型

# 实验与设计

## 11-1 利用MATLAB/DSP Builder设计基本电路模块

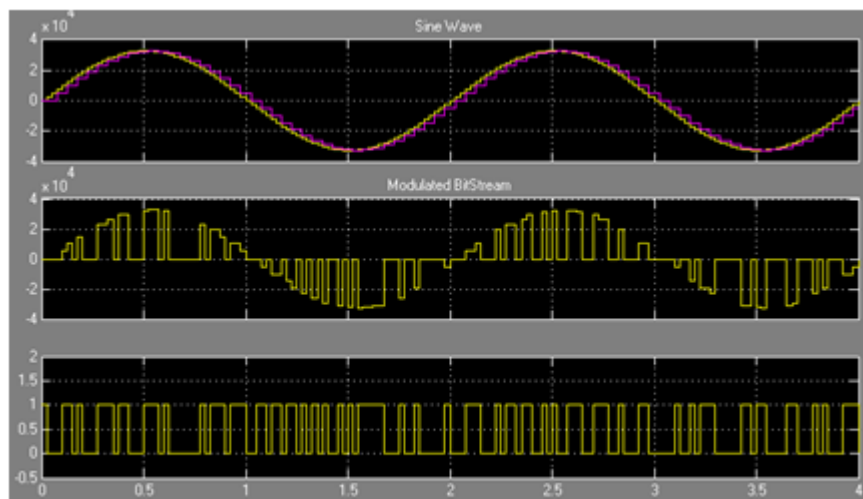


图 11-72 正弦调制信号仿真波形

# 实验与设计

## 11-2 基于DSP Builder的DDS应用模型设计

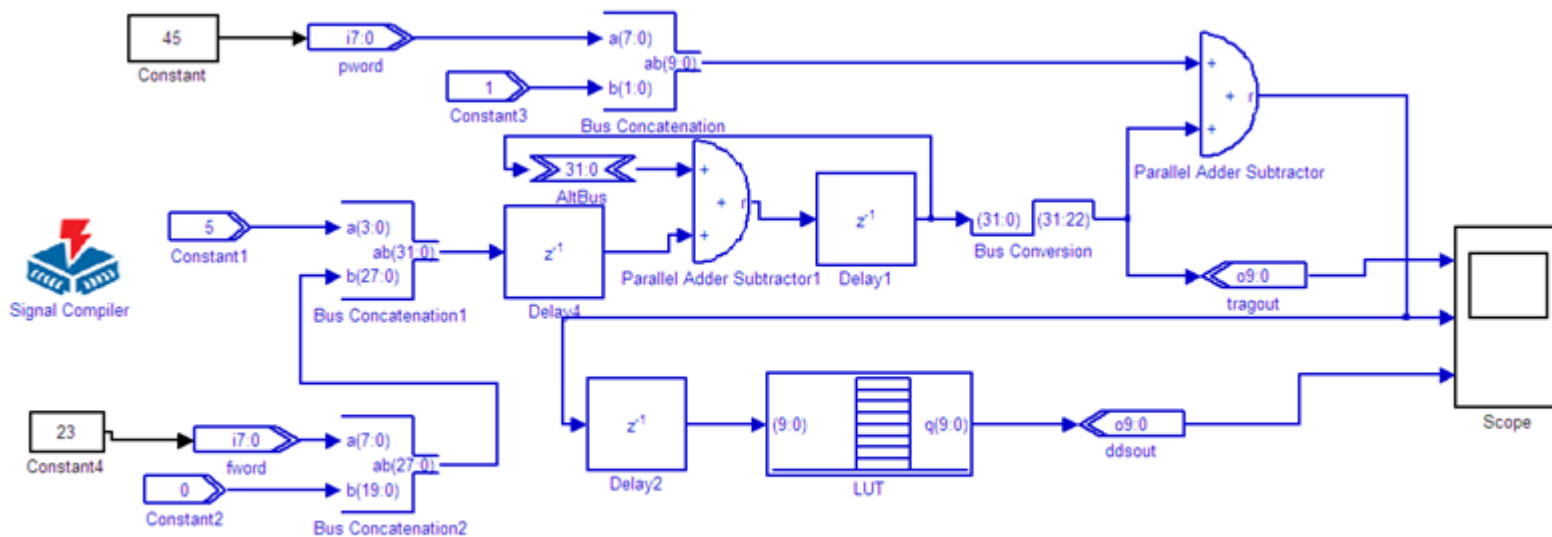


图 11-73 端口数据类型和位数变换后的 DDS 模型

# 实验与设计

## 11-2 基于DSP Builder的DDS应用模型设计

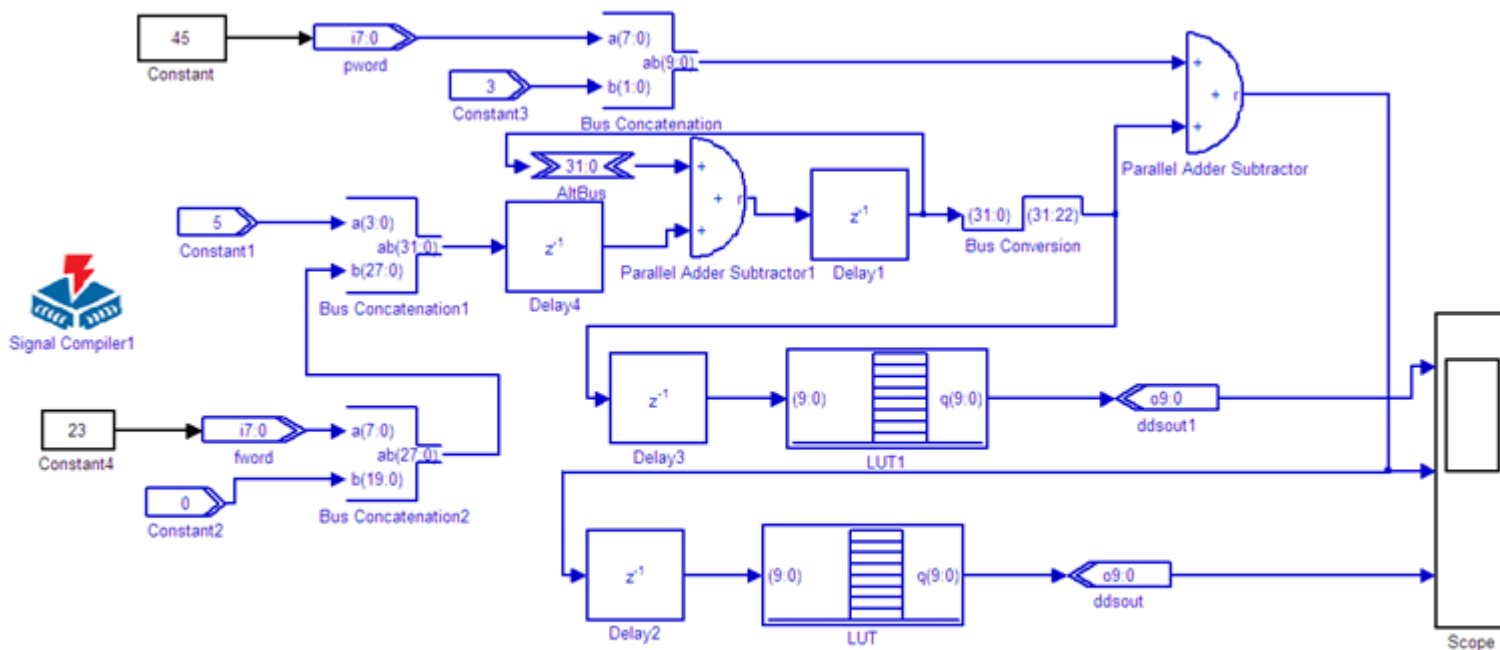


图 11-74 数字移相信号发生器



# 实验与设计

## 11-2 基于DSP Builder的DDS应用模型设计

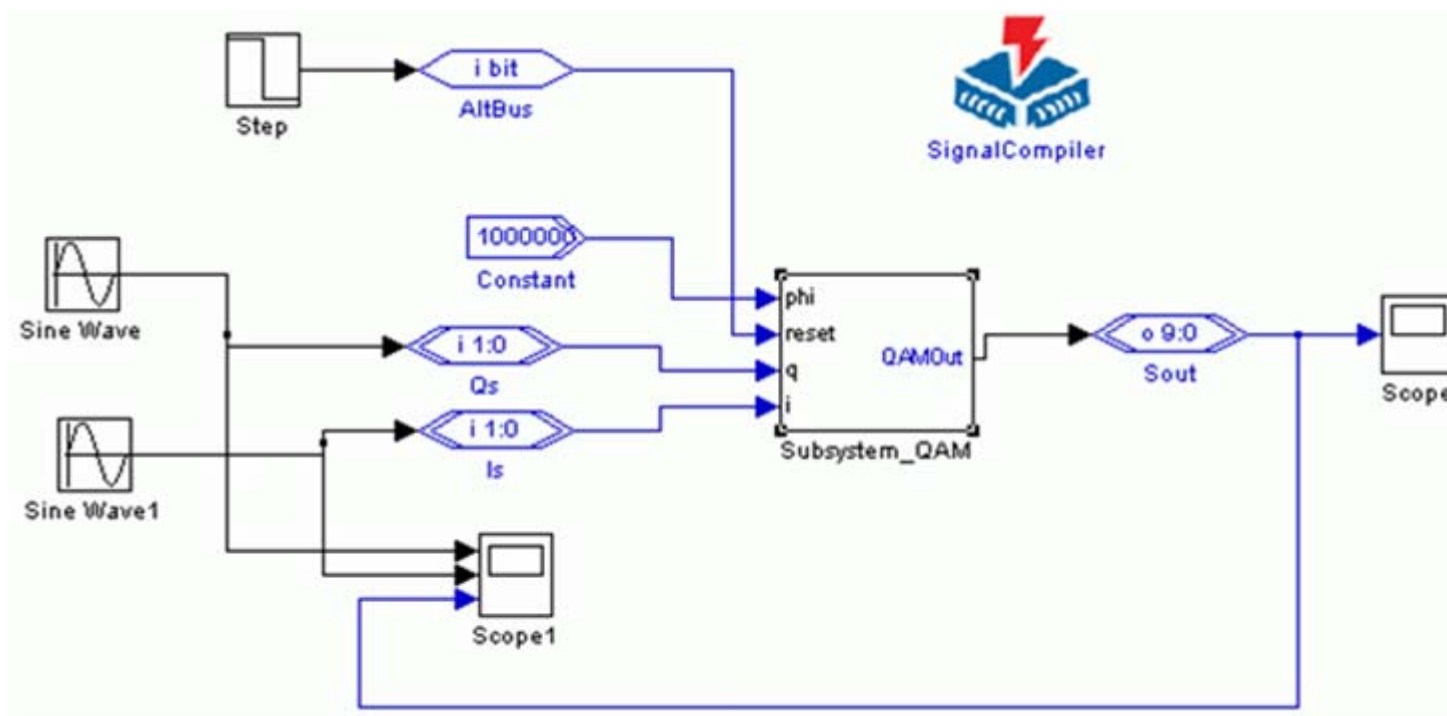


图 11-75 QAM 模型

# 实验与设计

## 11-3 编译码器设计实验

## 11-4 HIL硬件环仿真实验

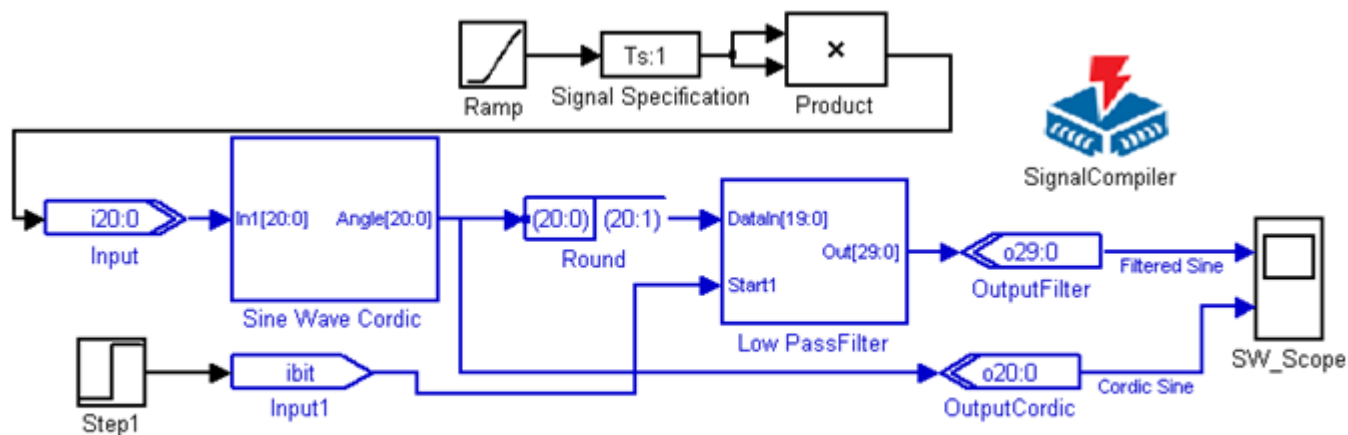


图 11-76 扫频滤波信号发生器模型 FreqSweep.mdl

# 实验与设计

## 11-4 HIL硬件环仿真实验

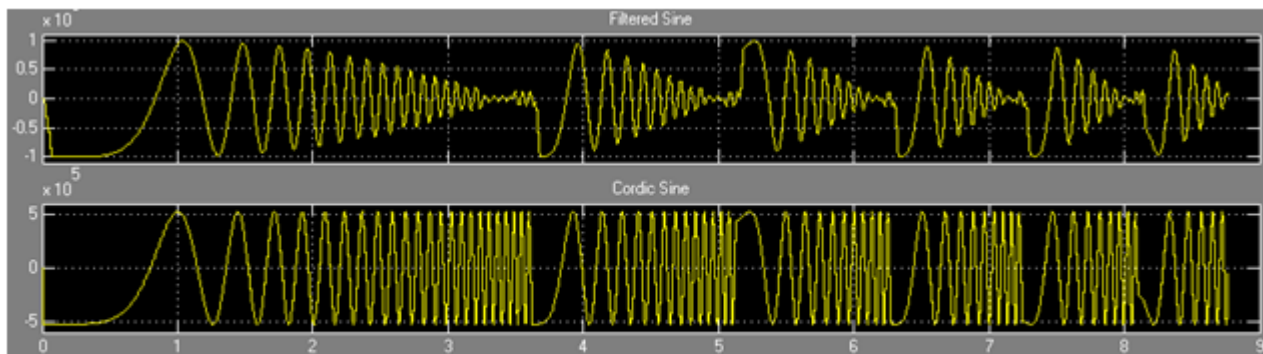


图 11-77 FreqSweep.mdl 模型的 Simulink 仿真波形

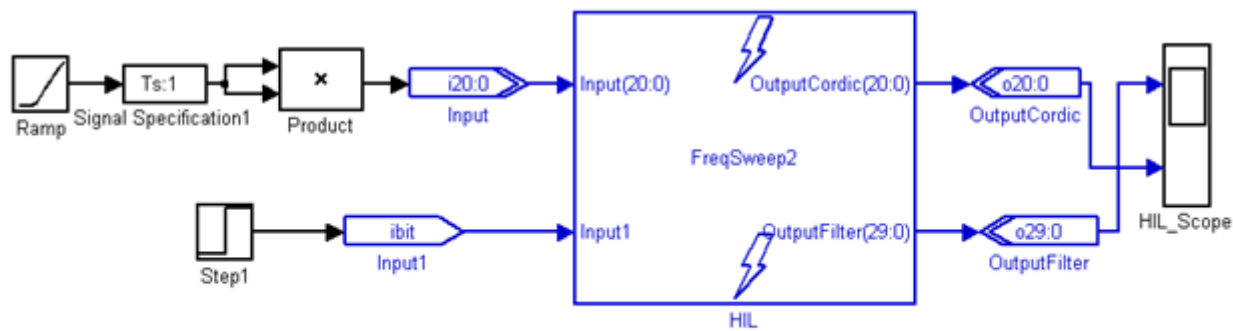


图 11-78 FreqSweep\_HIL.mdl 模型电路

# 实验与设计

## 11-5 DSP Builder 状态机应用实验

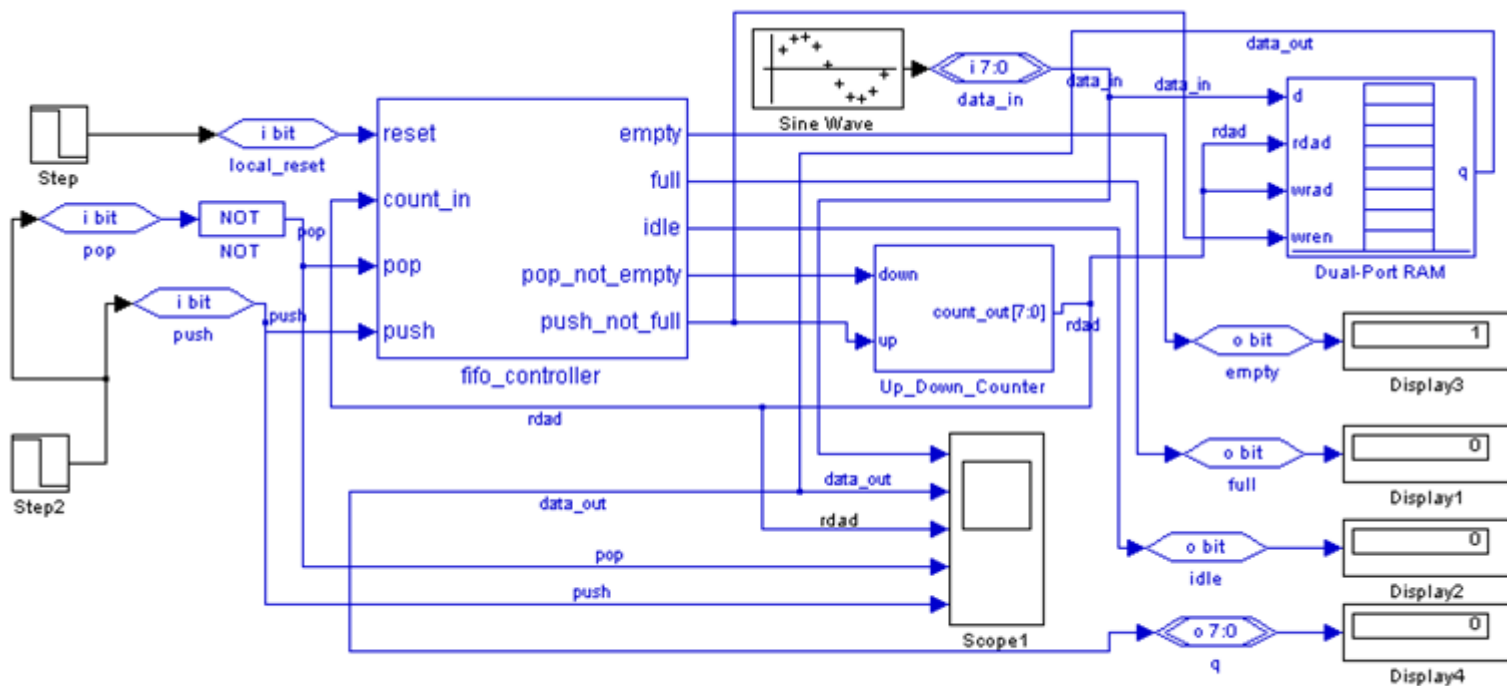


图 11-79 由状态机模块和双口 RAM 构成的 FIFO 存储器电路 fifo\_control.mdl

# 实验与设计

## 11-5 DSP Builder状态机应用实验

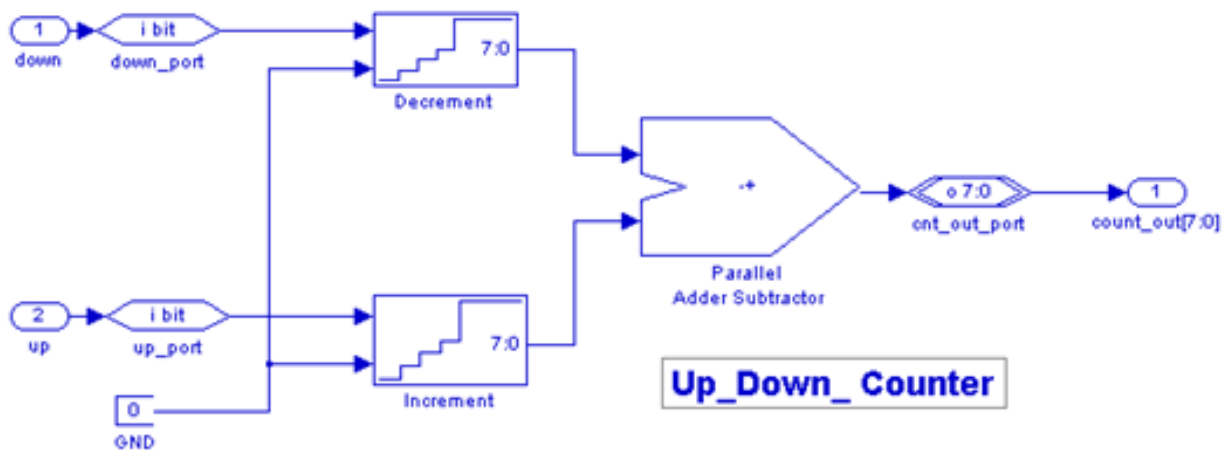


图 11-80 加减计数器模块 Up\_Down\_Counter 内部电路