FIR Filter Core User Guide

UG-CORE-FIR-v1.5
June 2023
www.elitestek.com

Contents

Introduction **3**

Features **3**

Device Support **3**

Resource Utilization and Performance **3**

Installing the Core **5**

Functional Description **6**

Ports 7

FIR Filter Operations 8

Customizing the FIR Filter **11**

FIR Filter Testbench **12**

Revision History **12**

Introduction

The FIR Filter core allows you to easily create an FIR filter within 易灵思 FPGAs to perform various signal processing such as, removing unwanted noise, low-pass filtering, band-pass filtering, and high-pass filtering.

Features

- Supports single-rate, polyphase interpolator, and polyphase decimator filter types
- FIR filter with up to:
  - 2048 coefficients with up to 32 bit coefficient width
  - 16 channels with up to 32 bit data width
- Supports signed binary for input and output
- Supports time division multiplexing and parallel data channel
- Verilog HDL RTL and simulation testbench

Device Support

Table 1: FIR Filter Core Device Support

FPGA Family	Supported Device
Trion	All
钛金系列	All

Resource Utilization and Performance

Note: The resources and performance values provided are based on some of the supported FPGAs. These values are just guidance and change depending on the device resource utilization, design congestion, and user design.

The resources are based on the following parameter settings:

- DATA_WIDTH = 8
- COEF_WIDTH = 8
- NUM_CHANNEL = 1
- TDM = 1
- USE_RAM_BLOCK = 1
- NUM_INTERPOLATION = 2 (Polyphase Interpolation filter only)
- NUM_ DECIMATION = 2 (Polyphase Decimation filter only)

Table 2: 钛金系列 Resource Utilization and Performance

FPGA	Filter Type / NUM_TAB	Logic and Adders	Flip-flops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity® Version[1]
Ti60 F225 C4	Single rate / 8	206	218	1	8	450	2021.2
	Single rate / 32	845	824	1	32	420	

[1] Using Verilog HDL.

FPGA	Filter Type / NUM_TAB	Logic and Adders	Flip-flops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity® Version[1]
	Single rate / 128	3,356	3,230	1	128	370	
	Polyphase Interpolation / 8	191	256	1	8	358	
	Polyphase Interpolation / 32	734	766	1	32	315	
	Polyphase Interpolation / 128	2,861	2,788	1	128	271	
	Polyphase Decimation / 8	242	250	1	8	379	
	Polyphase Decimation / 32	858	875	1	32	283	
	Polyphase Decimation / 128	3,266	3,392	1	128	250	

The resources are based on the following parameter settings:

- DATA_WIDTH = 8
- COEF_WIDTH = 8
- NUM_CHANNEL = 1
- TDM = 1
- USE_RAM_BLOCK = 1
- NUM_INTERPOLATION = 2 (Polyphase Interpolation filter only)
- NUM_ DECIMATION = 2 (Polyphase Decimation filter only)

Table 3: Trion Resource Utilization and Performance

FPGA	Mode / Width (bit) / Latency	Logic Utilization (LUTs)	Registers	Memory Blocks	Multipliers	f_{MAX} (MHz)	Efinity® Version[1]
T120 BGA324 C4	Single rate / 8	206	218	1	8	115	2021.2
	Single rate / 32	845	824	1	32	115	
	Single rate / 128	3,356	3,230	1	128	105	
	Polyphase Interpolation / 8	191	256	1	8	135	
	Polyphase Interpolation / 32	734	766	1	32	110	
	Polyphase Interpolation / 128	2,861	2,788	1	128	105	
	Polyphase Decimation / 8	242	250	1	8	149	
	Polyphase Decimation / 32	875	858	1	32	115	
	Polyphase Decimation / 128	3,392	3,266	1	128	101	

[1] Using Verilog HDL.

Installing the Core

The FIR Filter core is an Early Access core and it is not included in the Efinity IP Manager. To obtain the core, download the **efx_fir_filter-v**<version>**.zip** file from the Support Center.

The file contains:

File or Folder	Description
/efx_fir_filter.sv	FIR Filter core RTL source file.
coef.hex	Coefficient set text file.
/testbench	Contains simulation testbench files.

To install the FIR Filter core:

1. Unzip and copy the files into your project directory.
2. Instantiate the FIR Filter core in your top-level wrapper file.

   You can refer to the **efx_fir_filter_top.sv** wrapper file included in the downloaded IP files. Ensure to check the UUID used is the latest one.

   Example:

```
module top (
     <ports>
 );

efx_fir_filter_<uuid> #  (
    <parameters>
)  fir_inst  (
    <ports>
);
endmodule
```

   The `<uuid>` is included in the efx_fir_filter.sv file. For example, `` `define IP_UUID _0 `` then the `<uuid>` is `0`.

   Note: You can also customize the core using parameters in your design file. The core uses the default settings if no parameter is set.

3. Copy the included **coef.hex** file into the same folder as your Efinity project file.
4. Open your project in the Efinity software, click **File** > **Edit Project**, and in the **Design** tab, add design file and select **efx_fir_filter.sv**.

   Note: If required, update the Synopsys Design Constraints (**.sdc**) file before compiling your design.

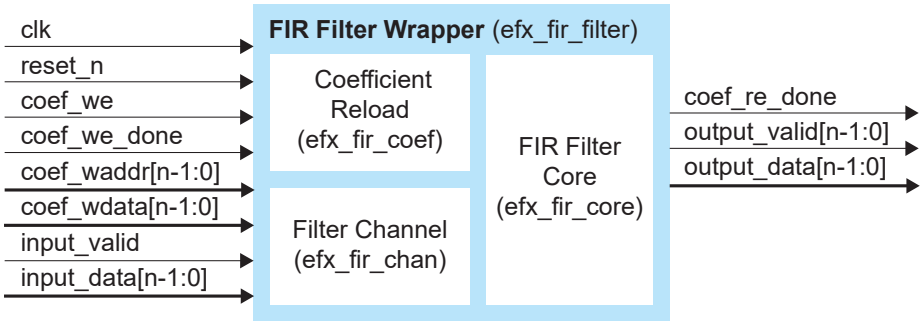
5. Compile your project.

Functional Description

The FIR Filter core consists of the following blocks:

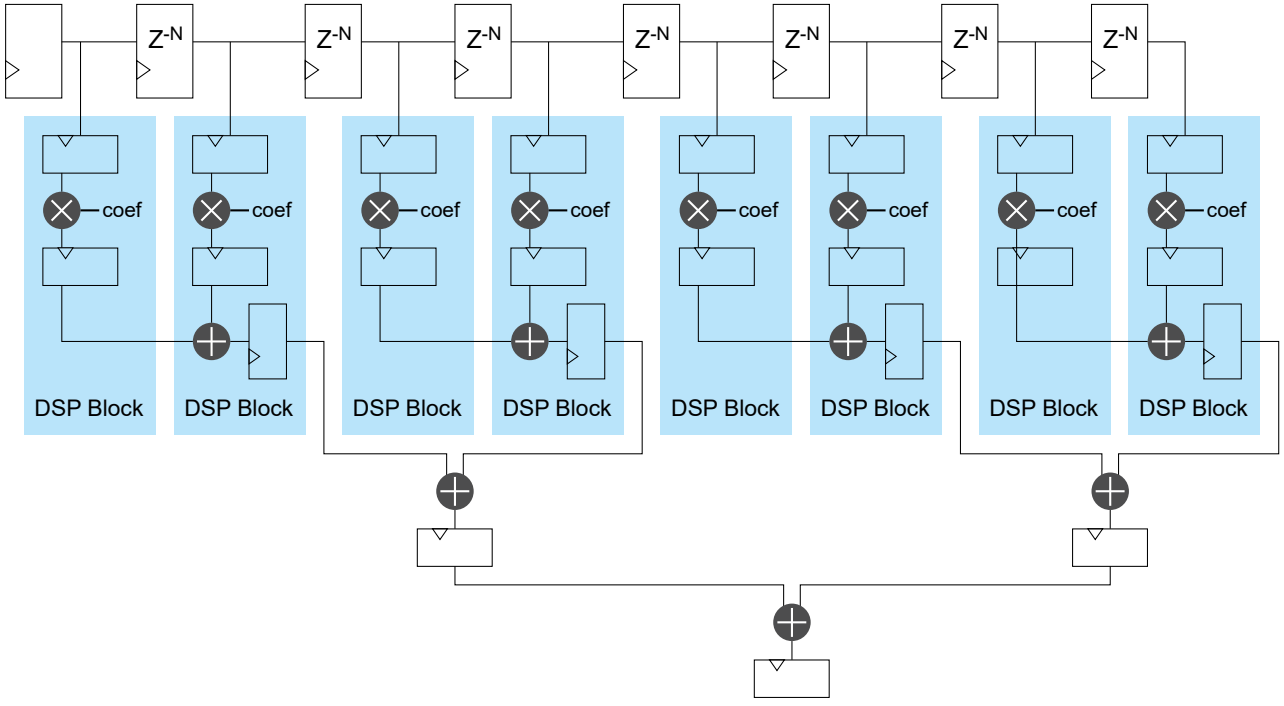
- FIR Filter Wrapper—Top wrapper file that instantiates internal blocks and performs output data repacking
- FIR Filter core—FIR filter multiplication and accumulation logic
- Coefficient Reload—Coefficient set reload logic and a RAM block that stores coefficient set
- Filter Channel—Generates valid signal for channels

Figure 1: FIR Filter System Block Diagram



The FIR Filter core uses the FIR direct form structure. In a direct form implementation, an adder tree sums the outputs of the individual multipliers The following figure represents the FIR direct form structure. In this implementation, the chainout adders of the DSP blocks perform the first round of addition, halving the number of required adders outside the DSP blocks.

Figure 2: Direct Form FIR Filter



Ports

Table 4: FIR Filter Ports

Port	Direction	Description
clk	Input	IP core clock signal.
reset_n	Input	IP core asynchronous active low reset signal.
coef_we	Input	Write enable signal to write a coefficient value to the RAM.
coef_we_done	Input	Assert this signal for one clock cycle to indicate that coefficient reload operation is done. The IP core proceeds to read out the coefficient value from the RAM.
coef_waddr[n-1:0]	Input	Write address to coefficient RAM. n =ADDR_WIDTH Where ADDR_WIDTH is the address width of the coefficient RAM
coef_wdata[n-1:0]	Input	Coefficient write value to the RAM. n =COEF_WIDTH
coef_re_done	Output	This signal is asserted high when the coefficient set from the RAM is completely read out by the IP core.
input_valid	Input	Assert this signal high to indicate that the input_data is valid.
input_data[n-1:0]	Input	Input data. n =PHYWIRECNT_IN*DATA_WIDTH Where PHYWIRECNT_IN = NUM_CHANNEL/TDM
output_valid[n-1:0]	Output	This signal is asserted high to indicate the valid output data. n = PHYWIRECNT_OUT
output_data [n-1:0]	Output	Output data. n = PHYWIRECNT_OUT*(DATA_WIDTH+COEF_WIDTH+CARRY_BIT) Where for: FILTER_TYPE=0—PHYWIRECNT_OUT = NUM_CHANNEL/TDM FILTER_TYPE=1—PHYWIRECNT_OUT = (NUM_INTERPOLATION/TDM) * NUM_CHANNEL FILTER_TYPE=2—PHYWIRECNT_OUT = NUM_CHANNEL/TDM

FIR Filter Operations

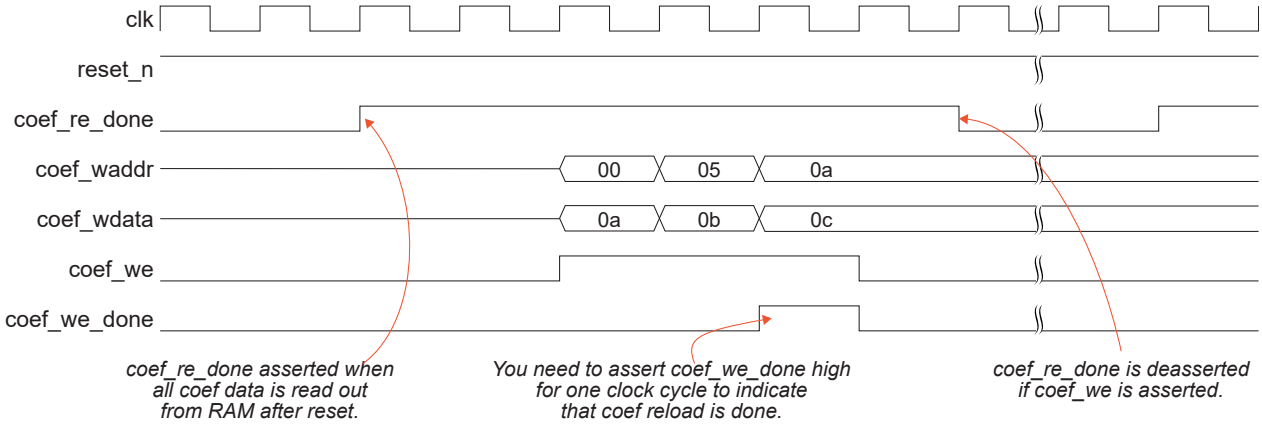
The following wave forms illustrates the FIR Filter filter operations.

Coefficient Reload

Parameters:

- NUM_CHANNEL = 1
- CLOCK_FREQ_MHZ = 100
- SAMPLE_RATE_MSPS = 100
- DATA_WIDTH = 8
- COEF_WIDTH = 8

Figure 3: Coefficient Reload Example Waveform

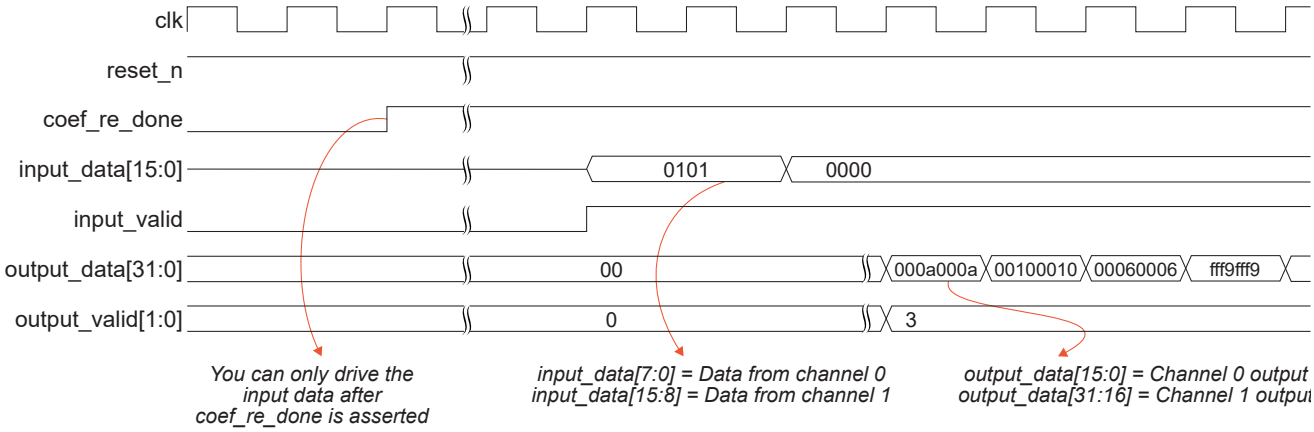


Single Rate Filter

Parameters:

- NUM_CHANNEL = 2
- CLOCK_FREQ_MHZ = 100
- SAMPLE_RATE_MSPS = 100
- DATA_WIDTH = 8
- COEF_WIDTH = 8

Figure 4: Single Rate Filter Example Waveform



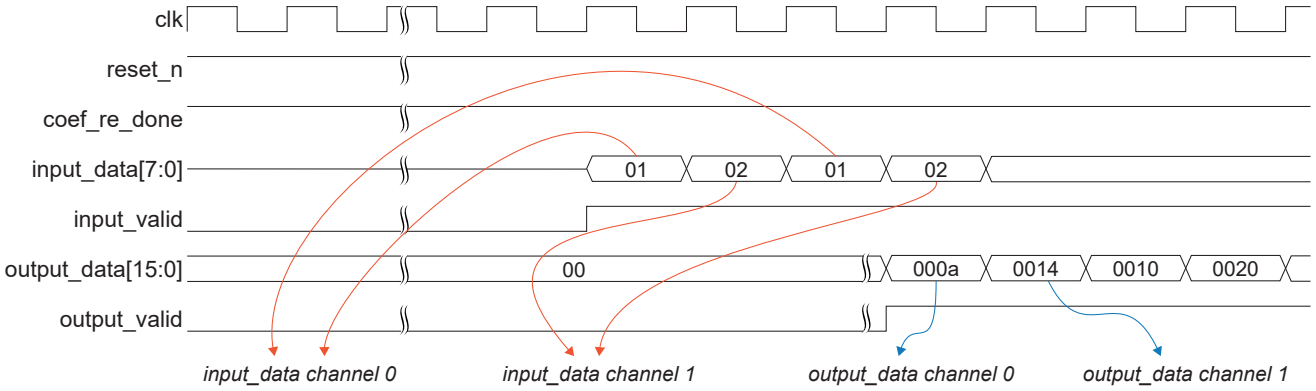
Single Rate Filter with Time Division Multiplexing of 2

Parameters:

- NUM_CHANNEL = 2
- CLOCK_FREQ_MHZ = 100

- SAMPLE_RATE_MSPS = 50
- DATA_WIDTH = 8
- COEF_WIDTH = 8

Figure 5: Single Rate Filter with Time Division Multiplexing of 2 Example Waveform

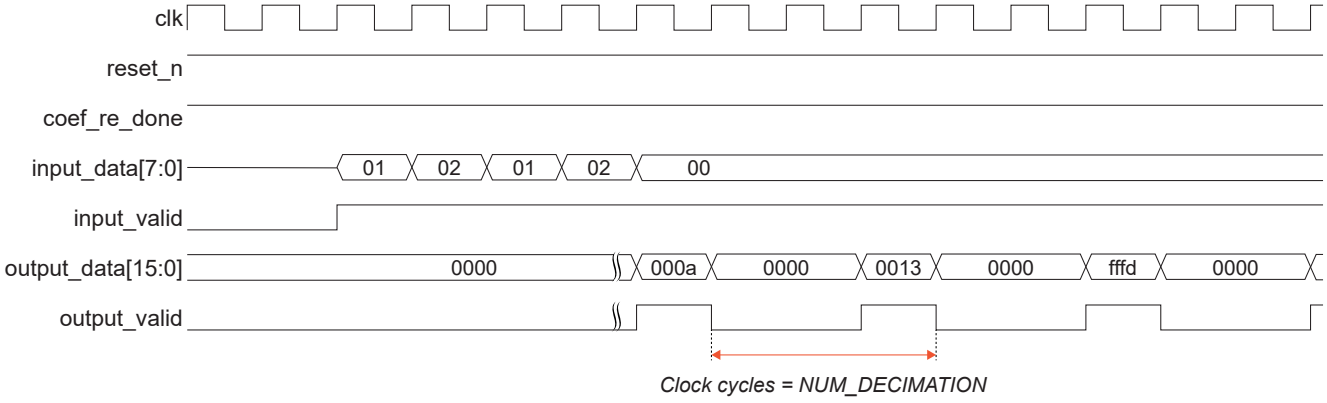


Polyphase Decimator Filter

Parameters:

- NUM_CHANNEL = 1
- CLOCK_FREQ_MHZ = 100
- SAMPLE_RATE_MSPS = 100
- DATA_WIDTH = 8
- COEF_WIDTH = 8
- NUM_DECIMATION = 3

Figure 6: Polyphase Decimator Filter Example Waveform

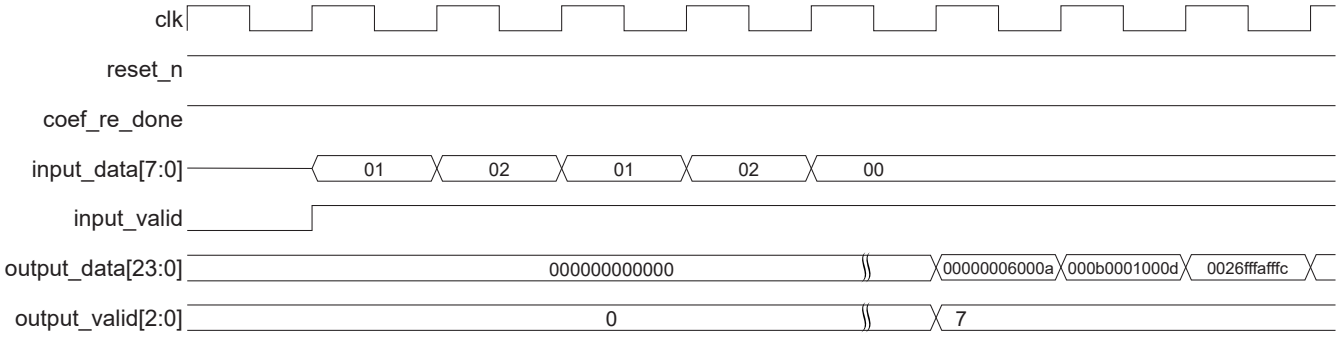


Polyphase Interpolation Filter

Parameters:

- NUM_CHANNEL = 1
- CLOCK_FREQ_MHZ = 100
- SAMPLE_RATE_MSPS = 100
- DATA_WIDTH = 8
- COEF_WIDTH = 8
- NUM_INTERPOLATION = 3

Figure 7: Polyphase Interpolation Filter Example Waveform



Customizing the FIR Filter

The FIR Filter core has parameters so you can customize its function.

Table 5: FIR Filter Core Parameters

Parameters	Options	Description
NUM_TAB	2 - 640	Defines the number of coefficient. This value must be the multiple of DECIMATION and INTERPOLATION FACTOR.
NUM_CHANNEL	1 - 16	Defines the number of physical channel. Default: 1
FILTER_TYPE	0, 1, 2	Defines the filter type. 0: Single rate (Default) 1: Polyphase interpolation 2: Polyphase decimation
NUM_DECIMATION	1 - 64	Defines the number of data points to remove between the original samples for polyphase decimation. Default: 1
NUM_INTERPOLATION	1 - 64	Defines the number of extra points to generate between the original samples for polyphase interpolation. Default: 1
CLOCK_FREQ_MHZ	See description	Defines the IP core clock frequency in MHz. CLOCK_FREQ_MHZ and SAMPLE_RATE_MBPS are used to determine the time division multiplexing (TDM) value, where, TDM = CLOCK_FREQ_MHZ/SAMPLE_RATE_MBPS The FIR Filter core supports TDM value of 1, 2, 3, or 4. Ensure that you set the CLOCK_FREQ_MHZ and SAMPLE_RATE_MBPS values accordingly. Default: 100
SAMPLE_RATE_MBPS	See description	Input data sample rate in MBPS. CLOCK_FREQ_MHZ and SAMPLE_RATE_MBPS are used to determine the time division multiplexing (TDM) value, where, TDM = CLOCK_FREQ_MHZ/SAMPLE_RATE_MBPS The FIR Filter core supports TDM value of 1, 2, 3, or 4. Ensure that you set the CLOCK_FREQ_MHZ and SAMPLE_RATE_MBPS values accordingly. Default: 100
DATA_WIDTH	8 - 32	Defines the data width. Default: 8
COEF_WIDTH	8 - 32	Defines the coefficient width. Default: 8
USE_RAM_BLOCK	0, 1	Enables RAM block for coefficient set. When this parameter is disabled, registers are be used to store the coefficient set. 0: Disable 1: Enable (default)
CARRY_BIT	0, 1, 2, 3, 4, 5, 6	Additional carry bit on the output data width. Default: 1

FIR Filter Testbench

The FIR Filter includes a simulation testbench which allow you to simulate the IP core. It also includes an input **.txt** file, a coefficient set **.hex** file, and a **modelsim.do** file.

The testbench file, **top_tb.sv**, takes in and translate the values from the input text file, **input.txt**, as the input data for the IP core. The coefficient set text file, **coef.hex**, is loaded into the RAM in the IP core upon out of reset.

The output data from the IP core is then translate into a text file, **output.txt**.

You can compare the **output.txt** value with the Matlab result for expected output.

Revision History

Table 6: Revision History

Date	Version	Description
June 2023	1.5	Added Device Support section. (DOC-1234) Improved Installing the Core description. (DOC-1287)
May 2023	1.4	Improved steps to install the core. (DOC-1244)
February 2023	1.3	Added note about the resource and performance values in the resource and utilization table are for guidance only.
April 2022	1.2	Editorial fixes.
January 2022	1.1	Updated output_data port width. Added USE_RAM_BLOCK and CARRY_BIT parameters. Added FIR Filter operation waveforms. Added core resource utilization.
December 2020	1.0	Initial release.