14-Bit Dual Port DA Output Module AN9767

User Manual

ALINX

Table of Contents

Part1: Dual Port DA Module General Description

14-bit dual port DA output module, use AD9767 chip of Analogy Devices, which supporting independent dual ports, 14-bit, 125MPS DA. The module reserves a 40-pin female header that to connect FPGA development kit, 2 BNC connectors to output analogy signals.

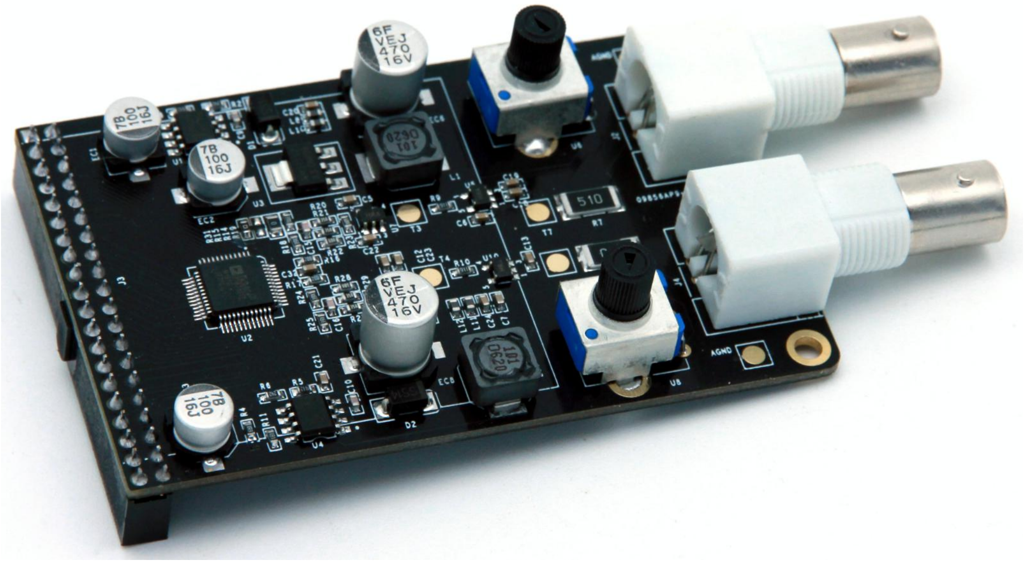


Figure 1-1: AN9767 module product photo (front side)

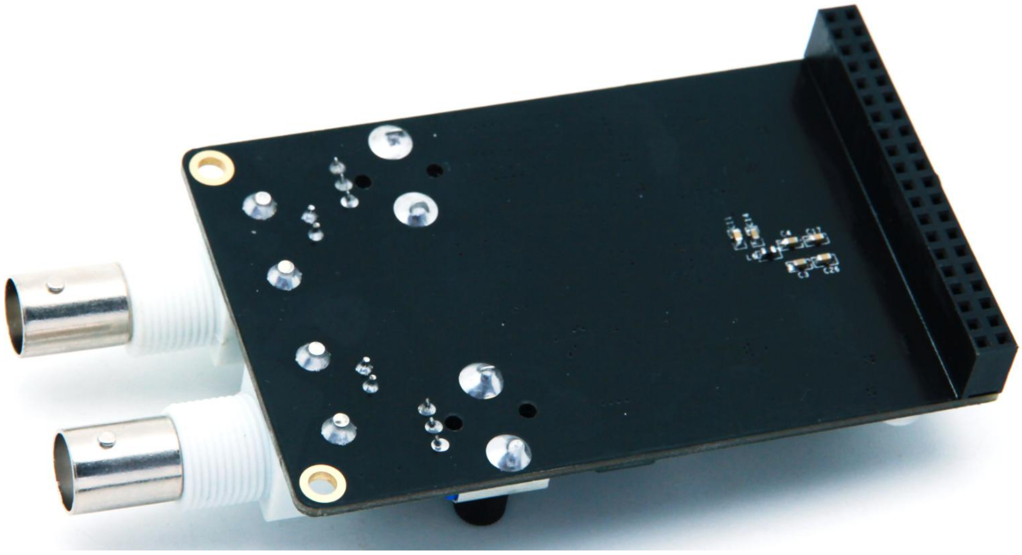


Figure 1-2: AN9767 module product photo (back side)

Part 1.1: AN9767 Module Detail Parameter

AN9767 dual port DA module detail parameter listed as below:

- DA chip：AD9767
- Channel：2-channel
- DA bits：14-Bits
- DA update rate：125 MSPS；
- Output Voltage range：-5V~+5V；
- PCB layers of Module：4-Layer，independent power layer and GND layer
- Module Interface：40-pin 0.1 spacing female header, download direction
- Ambient Temperature (with power applied： -40°~85°, all the chips on module to meet the industrial requirements.
- Output interface：2-Port BNC analog output interface (use the BNC line connect to the oscilloscope directly）

Part 1.2: AN9767 Module Size Dimension

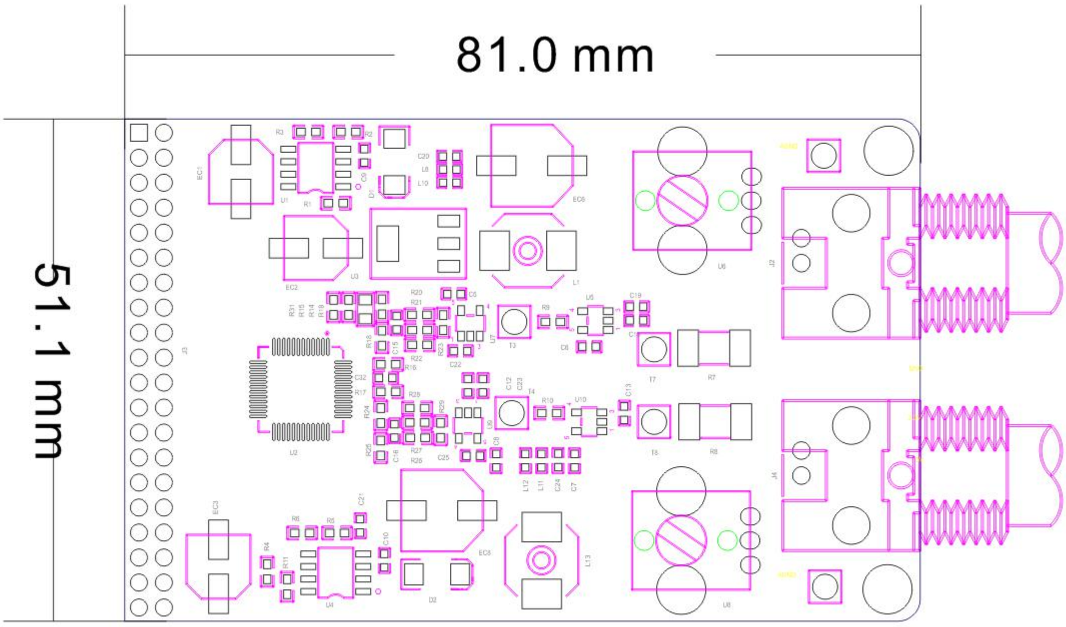


Figure 1-2: AN9767 dual port DA Module Dimensions

Part 2 Dual Port DA Module Function Description

Part 2.1 AN9767 Module Block Diagram

Figure 2-1: AN9767 Module Block Diagram as below:

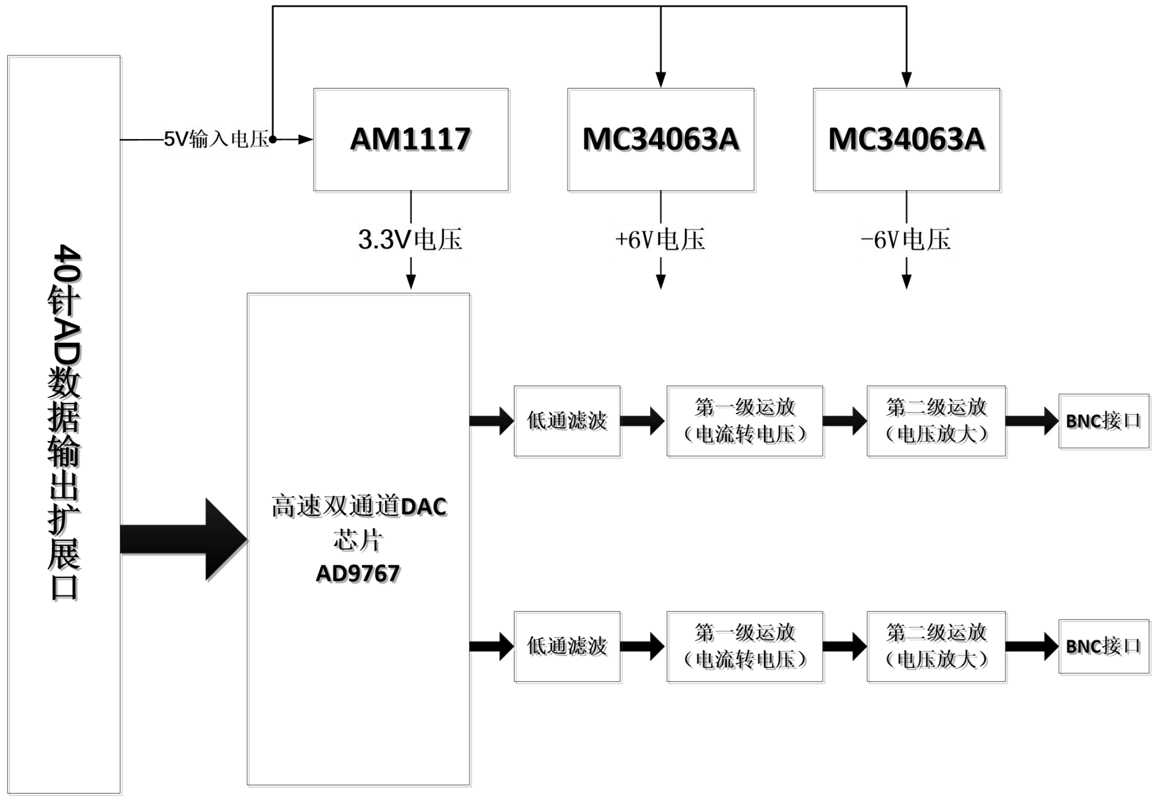


Figure 2-1: AN9767 Module Block Diagram

Part 2.2: AD9767 Chip Introduction

The AD9767 is a dual port, high speed, two channel, 14-bit CMOS DAC. It integrates two high quality 14-bit TxDAC+ cores, a voltage reference and digital interface circuitry into a 48-lead LQFP package. The AD9767 offers exceptional ac and dc performance while supporting updates rates up to 125 MSPS. Figure 2-2: The functional block diagram of AD9767 as below:

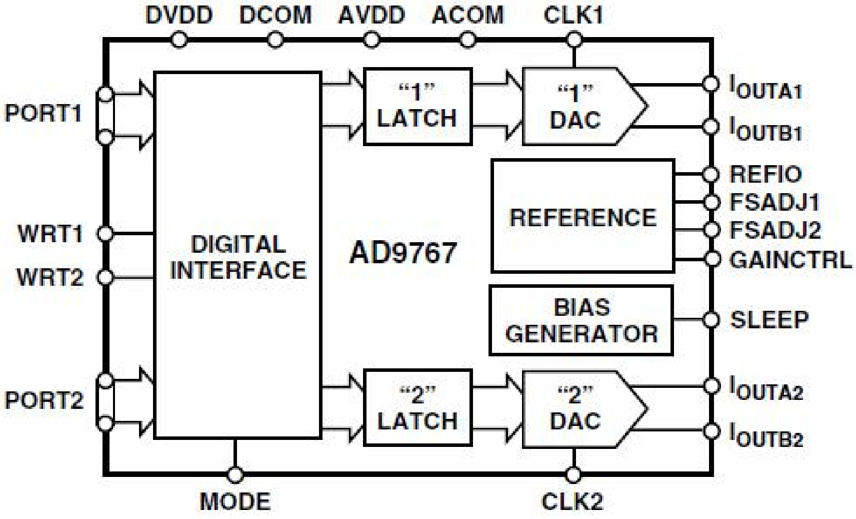


Figure 2-2: The functional block diagram of AD9767

Part 2.3: DAC Transfer Function

Both DACs in the AD9767 provide complementary current outputs, IoutA and IoutB。IoutA will provide a near full-scale current output, IoutFS, when all bits are high (i.e., DAC CODE=16383) while IoutB, the complementary output, provides no current. The current output appearing at IoutA and IoutB is a function of both the input code and IoutFS can be expressed as :

$$I_{OUTA} = (DAC\ CODE / 16384) \times I_{OUTFS}$$
$$I_{OUTB} = (16383 - DAC\ CODE)/16384) \times I_{OUTFS}$$

Where IoutFS=32 x Iref. In the design of AN9767, Iref is set by the resistor of R16，if the is R16=19.2K，then the value of Iref is 0.625mA, IoutFS is 20mA. The output current of AD9767 is converted to voltage -1V~+1V through the first stage operational amplifier AD6045. Figure 2-3: The Output current convert to voltage

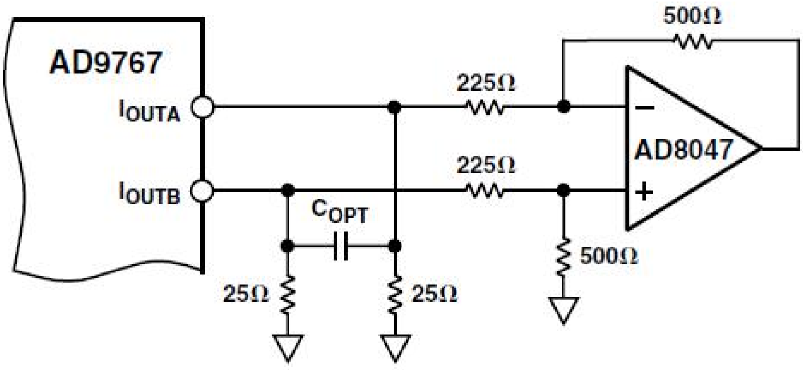


Figure 2-3: The Output current convert to voltage by first stage OP

The voltage of -1V~+1V after the conversion of the first stage operational amplifier is converted to a higher amplitude voltage signal by the second stage operational amplifier. The amplitude of the second operational amplifier can be change by adjusting the adjustable resistor on the board. With the second stage operation amplifier, the output of range of the analog signal as high as -5V~+5V

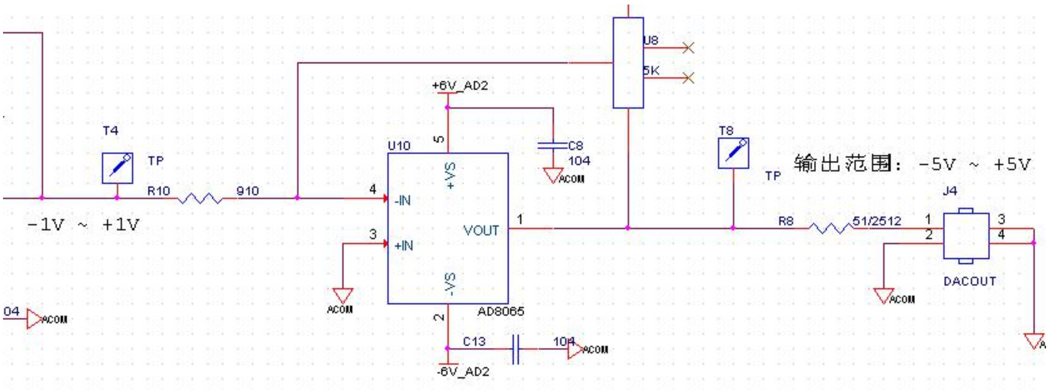


Figure 2-4: The Output Voltage Amplified by second stage OP

DAC data input value	AD9767current output	The output of first stage OP	The output of second stage OP
3fff(14 bits are high）	+20mA	-1V	+5V
0(14 bits are low）	-20mA	+1V	-5V
2000（Median）	0mA	0V	0V

Table 2-1: The input digital signal and current output from OP

Part 2.4: Current and Voltage Conversion and Amplifier

The digital inputs of AD9767 chip is used in Dual mode or Interleaved mode by setting PIN (MODE)。In the design of AN9767 module, the AD9767 chip operated in Dual mode. The DA digital inputs of dual ports are separated and independent. Time specifications for dual port mode are give in figure 2-5:

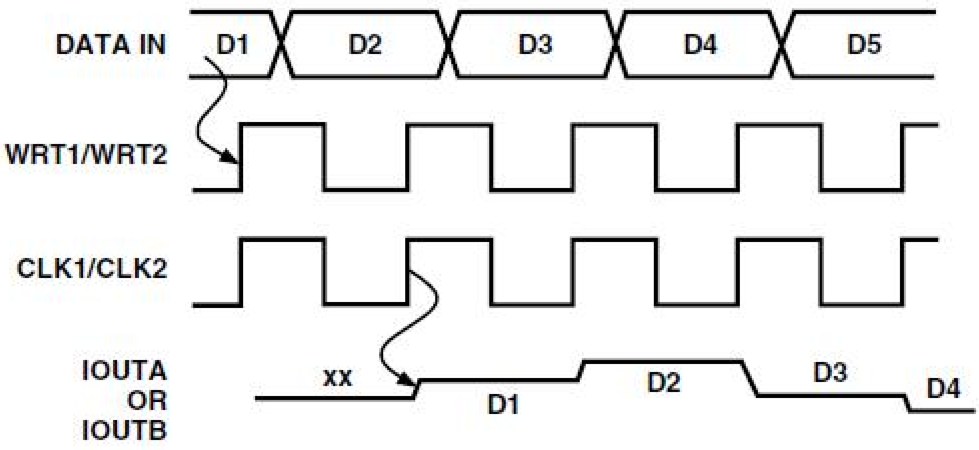


Figure 2-5: Dual Mode Timing

The DA data entered to the AD9766 and converted to DA with the rising edge of CLK and WRT.

Part2.5: AN9767 Module 40-pin Female Header Pins Description

Pin	Pin Name	Description
1	GND	Digital Ground
2	+5V	5V power input
3	P1_DATA13	Bit 13 of DAC Port 1 data
4	P1_DATA12	Bit 12 of DAC Port 1 data
5	P1_DATA11	Bit 11 of DAC Port 1 data
6	P1_DATA10	Bit 10 of DAC Port 1 data

7	**P1_DATA9**	Bit 9 of DAC Port 1 data
8	**P1_DATA8**	Bit 8 of DAC Port 1 data
9	**P1_DATA7**	Bit 7 of DAC Port 1 data
10	**P1_DATA6**	Bit 6 of DAC Port 1 data
11	**P1_DATA5**	Bit 5 of DAC Port 1 data
12	**P1_DATA4**	Bit 4 of DAC Port 1 data
13	**P1_DATA3**	Bit 3 of DAC Port 1 data
14	**P1_DATA2**	Bit 2 of DAC Port 1 data
15	**P1_DATA1**	Bit 1 of DAC Port 1 data
16	**P1_DATA0**	Bit 0 of DAC Port 1 data
17	**P1_WRT**	Input write signal for DAC port 1
18	**P1_CLK**	Clock input for DAC port 1
19	**P2_CLK**	Clock input for DAC port 2
20	**P2_WRT**	Input write signal for DAC port 2
21	**P2_DATA13**	Bit 13 of DAC Port 2 data
22	**P2_DATA12**	Bit 12 of DAC Port 2 data
23	**P2_DATA11**	Bit 11 of DAC Port 2 data
24	**P2_DATA10**	Bit 10 of DAC Port 2 data
25	**P2_DATA9**	Bit 9 of DAC Port 2 data
26	**P2_DATA8**	Bit 8 of DAC Port 2 data
27	**P2_DATA7**	Bit 7 of DAC Port 2 data
28	**P2_DATA6**	Bit 6 of DAC Port 2 data
29	**P2_DATA5**	Bit 5 of DAC Port 2 data
30	**P2_DATA4**	Bit 4 of DAC Port 2 data
31	**P2_DATA3**	Bit 3 of DAC Port 2 data
32	**P2_DATA2**	Bit 2 of DAC Port 2 data
33	**P2_DATA1**	Bit 1 of DAC Port 2 data

34	P2_DATA0	Bit 0 of DAC Port 2 data
35	-	NC
36	-	NC
37	GND	Digital Ground
38	GND	Digital Ground
39	-	NC
40	-	NC

Part 3: Sine Wave Generation Program Description

We provide two DA testing programs of AN9767 module. One is sine wave generation program and another is triangular wave generation program. Regarding the triangular wave generation program is simple, so not provide the detail introduction. Here is the sine wave generation program introduction.

The sine wave test program reads the sine wave data stored in a ROM inside the FPGA, and then outputs the sine wave data to the AN9767 module for digital-to-analog conversion, thereby obtaining a sine wave analog signal. The schematic diagram of the sine wave test program as follow:

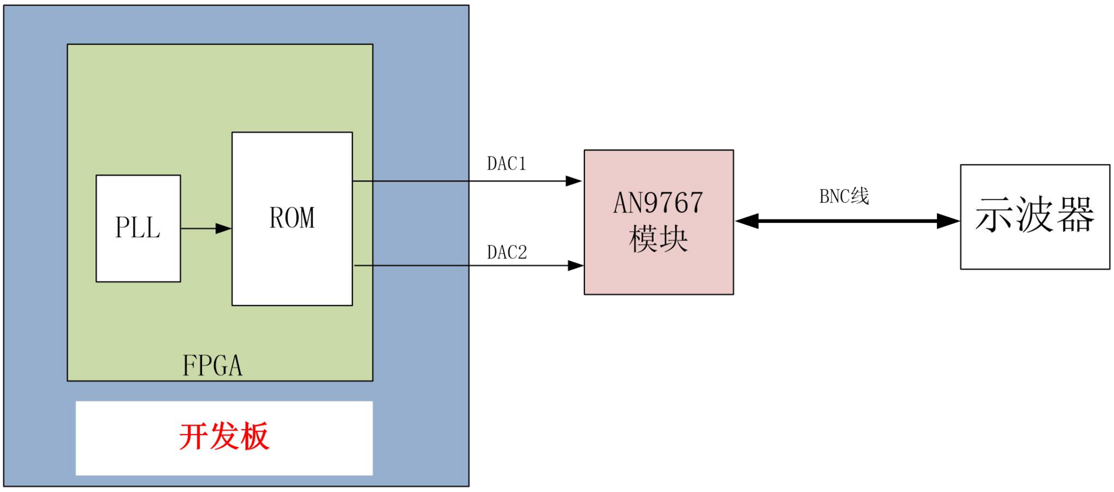
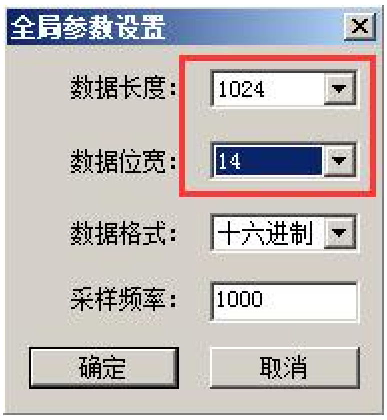


Figure 3-1: The schematic diagram of the sine wave test program

Part 3.1: Generate ROM Initialization file

In the program we will use a ROM for storing 1024 14-bit sine wave data. First we need to prepare the ROM initialization file (if it is an ALTERA development board, it is a mif file, if it is a Xilinx development board, it is a coe file). The following is a method for generating a sine wave ROM data file：

Firstly, Open the Guagle_wave in the “Software Tools”, select the menu “View” ->”Global Parameter Setting”，set the parameter as below:



全局参数设置

数据长度：1024

数据位宽：14

数据格式：十六进制

采样频率：1000

Figure 3-2: Global Parameter Setting

Select the menu “Set Waveform” and set to sine wave

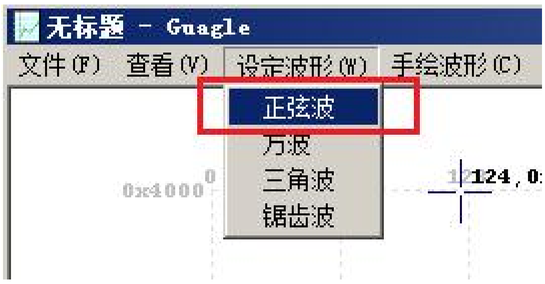


Figure 3-3: Set Waveform to Sine Wave

Select the menu “View”and set to data carve.

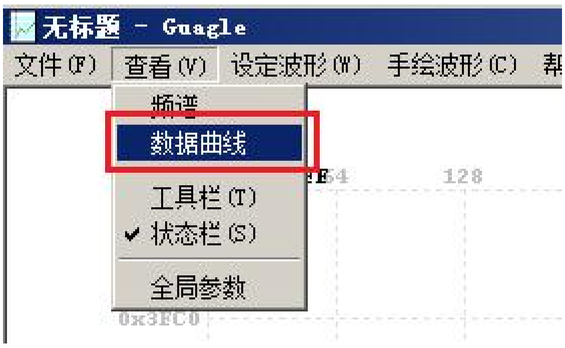


Figure 3-4: Set to Data Curve

Save the file in mif format and named sin1024.mif (Pls. pay attention do not select save as, or the file will be blank)

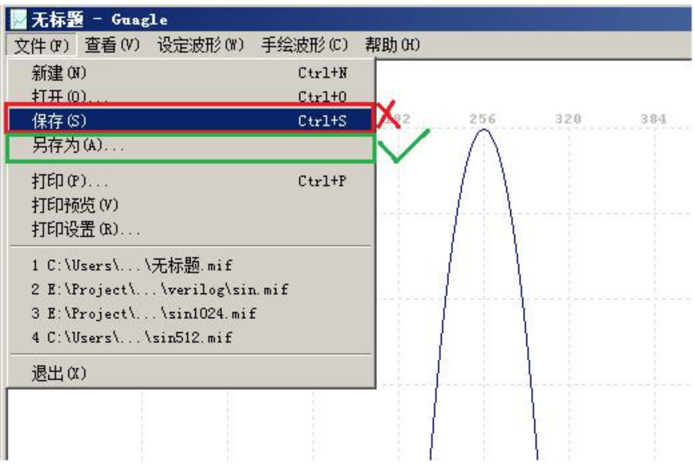
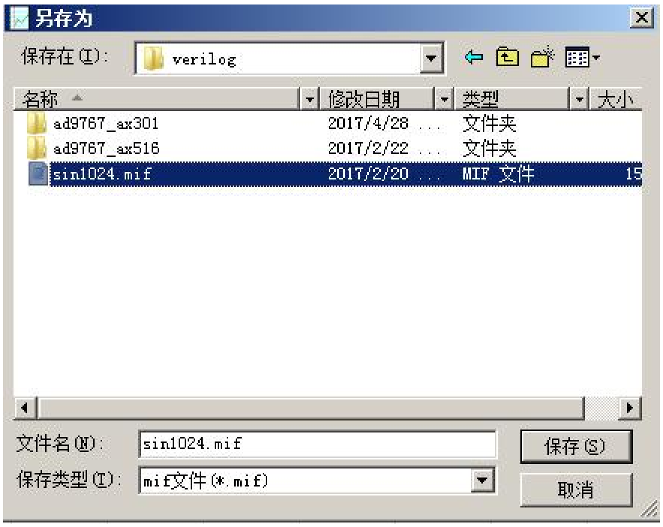


Figure 3-5: Select Save as to Save File

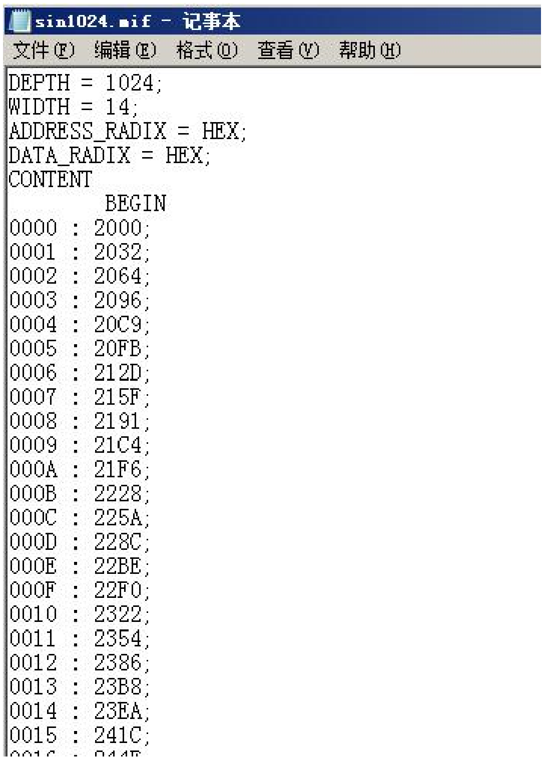


文件名(N): sin1024.mif

保存类型(T): mif文件(*.mif)

Figure 3-6: Saved File and named

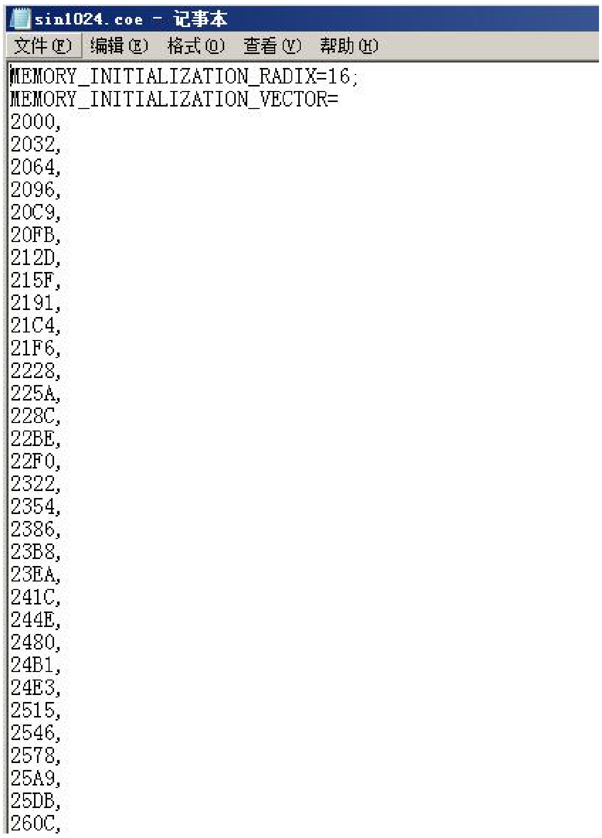
Then the mif file generated. For the Altera FPGA development kits, the mif file is the ROM initialization file. Open the sin1024.mif file with text and check the date format and content inside.



```
DEPTH = 1024;
WIDTH = 14;
ADDRESS_RADIX = HEX;
DATA_RADIX = HEX;
CONTENT
        BEGIN
0000 : 2000;
0001 : 2032;
0002 : 2064;
0003 : 2096;
0004 : 20C9;
0005 : 20FB;
0006 : 212D;
0007 : 215F;
0008 : 2191;
0009 : 21C4;
000A : 21F6;
000B : 2228;
000C : 225A;
000D : 228C;
000E : 22BE;
000F : 22F0;
0010 : 2322;
0011 : 2354;
0012 : 2386;
0013 : 23B8;
0014 : 23EA;
0015 : 241C;
```

Figure 3-7: Open the sin1024.mif file with Text

For the Xilinx FPGA development kits, the user also needs to modify it to Coe's file format. Because the format of the initialization file coe of Xilinx ROM is different from the format of the mif file, we need to re-edit the generated mif with the text file or excel into the following format (hexadecimal, 1024 data length):

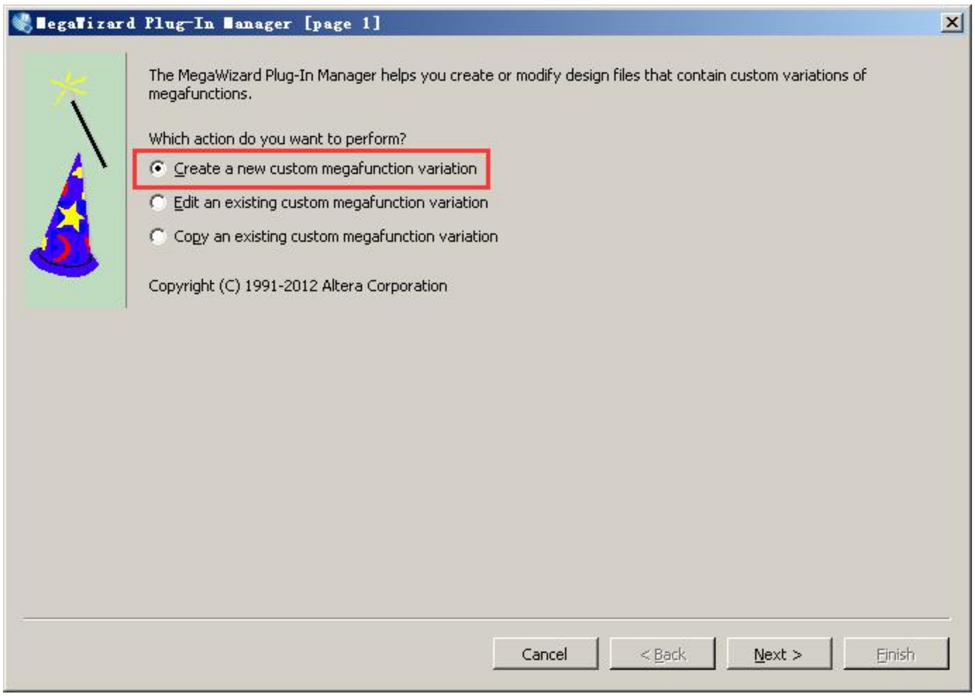


```
MEMORY_INITIALIZATION_RADIX=16;
MEMORY_INITIALIZATION_VECTOR=
2000,
2032,
2064,
2096,
20C9,
20FB,
212D,
215F,
2191,
21C4,
21F6,
2228,
225A,
228C,
22BE,
22F0,
2322,
2354,
2386,
23B8,
23EA,
241C,
244E,
2480,
24B1,
24E3,
2515,
2546,
2578,
25A9,
25DB,
260C,
```

Figure 3-8: Re-edit to Coe's file format

Part 3.2: Add ROM IP & Configuration Initialization file

Let's take the AX301 development board as an example to introduce the generation and configuration of ROM IP. Let's create a new Quartus project and add a ROM IP to the project. Select the Tool->MegaWizard Plug-In Manager, and popped up MegaWizard Plug-in Manager window to choose the first on "Create a new custom megafunction variation".



MegaWizard Plug-In Manager [page 1]

The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.

Which action do you want to perform?

● Create a new custom megafunction variation

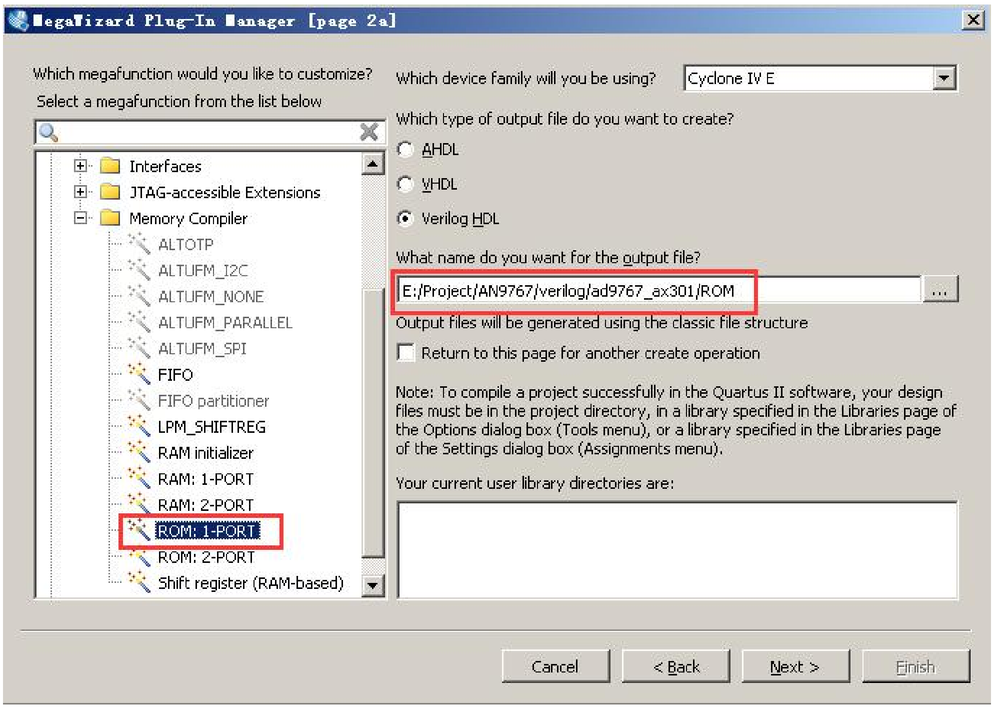
○ Edit an existing custom megafunction variation

○ Copy an existing custom megafunction variation

Copyright (C) 1991-2012 Altera Corporation

Figure 3-9: MegaWizard Plug-In Manager [page 1] Window

Select ROM:1-PORT, then input the direction and named the file that save the IP, here named the file as rom.



MegaWizard Plug-In Manager [page 2a]

Which megafunction would you like to customize?
Select a megafunction from the list below

Which device family will you be using? Cyclone IV E

Which type of output file do you want to create?

○ AHDL

○ VHDL

● Verilog HDL

What name do you want for the output file?

E:/Project/AN9767/verilog/ad9767_ax301/ROM

Output files will be generated using the classic file structure

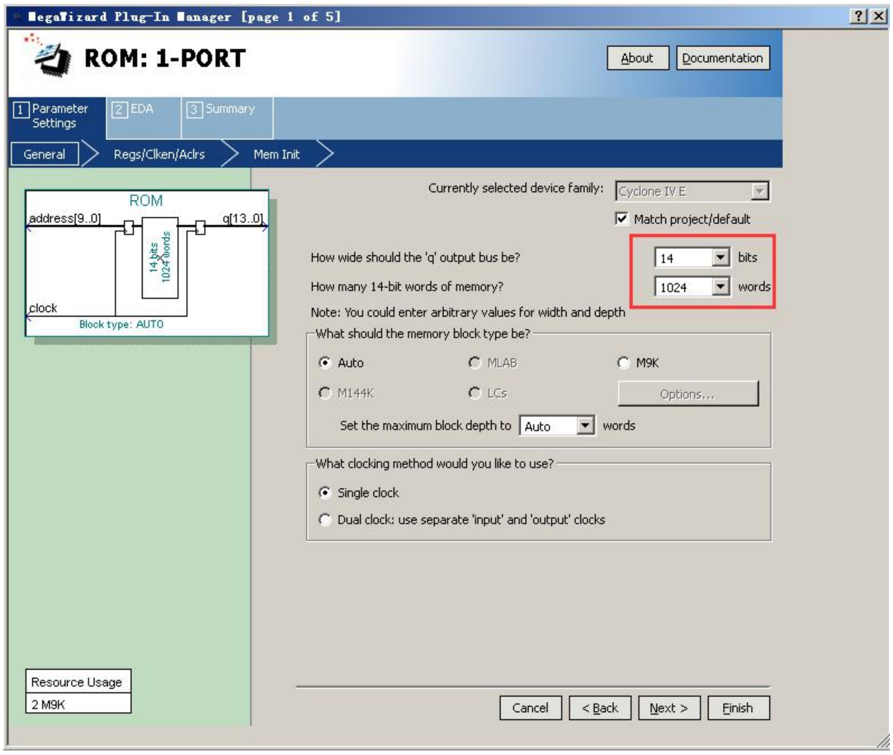
□ Return to this page for another create operation

Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in a library specified in the Libraries page of the Options dialog box (Tools menu), or a library specified in the Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

Figure 3-10: MegaWizard Plug-In Manager [page 2a] Window

The data width of the selected ROM is 14 bits, and the depth of the data is 1024 data. This setting is exactly the same as the size of the mif file generated in previous.



Currently selected device family: Cyclone IV E

☑ Match project/default

How wide should the 'q' output bus be? 14 bits

How many 14-bit words of memory? 1024 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?

● Auto ○ MLAB ○ M9K
○ M144K ○ LCs Options...

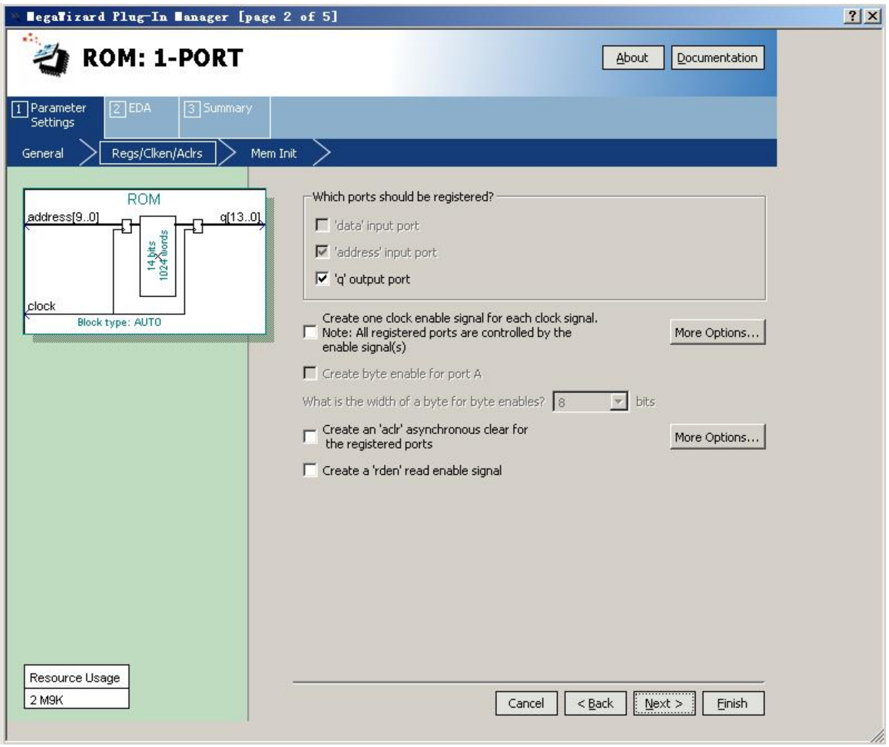
Set the maximum block depth to Auto words

What clocking method would you like to use?

● Single clock
○ Dual clock: use separate 'input' and 'output' clocks

Figure 3-11: MegaWizard Plug-In Manager [page 1 of 5] Window

keep the default settings, click the Next button



Which ports should be registered?

- □ 'data' input port
- ☑ 'address' input port
- ☑ 'q' output port

□ Create one clock enable signal for each clock signal. Note: All registered ports are controlled by the enable signal(s) — More Options...

□ Create byte enable for port A

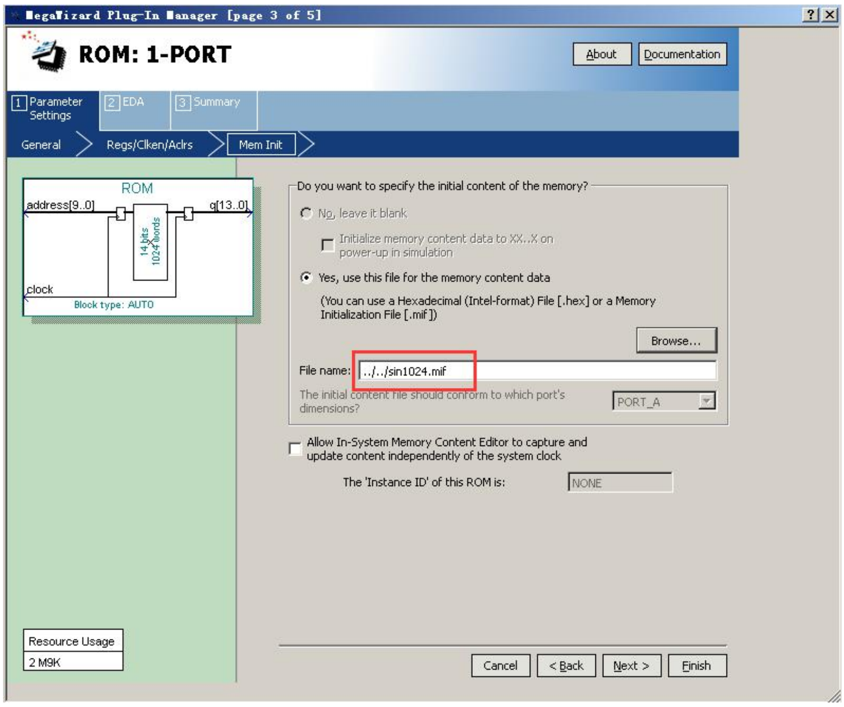
What is the width of a byte for byte enables? 8 bits

□ Create an 'aclr' asynchronous clear for the registered ports — More Options...

□ Create a 'rden' read enable signal

Figure 3-12: MegaWizard Plug-In Manager [page 2 of 5] Window

Click browse button and select the ROM initialization file sin1024.mif



Do you want to specify the initial content of the memory?

- ○ No, leave it blank
  - □ Initialize memory content data to XX..X on power-up in simulation
- ● Yes, use this file for the memory content data

(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

File name: ../../sin1024.mif

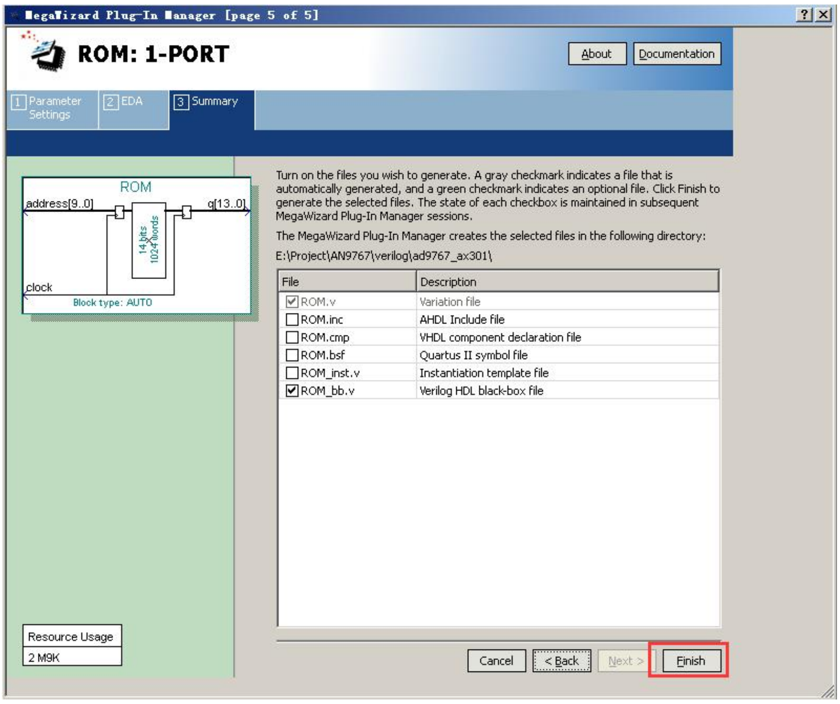
The initial content file should conform to which port's dimensions? PORT_A

□ Allow In-System Memory Content Editor to capture and update content independently of the system clock

The 'Instance ID' of this ROM is: NONE

Figure 3-13: MegaWizard Plug-In Manager [page 2 of 5] Window

Click Finish button, completed the addition of ROM IP



Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a green checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:

E:\Project\AN9767\verilog\ad9767_ax301\

File	Description
☑ ROM.v	Variation file
□ ROM.inc	AHDL Include file
□ ROM.cmp	VHDL component declaration file
□ ROM.bsf	Quartus II symbol file
□ ROM_inst.v	Instantiation template file
☑ ROM_bb.v	Verilog HDL black-box file

Resource Usage
2 M9K

Figure 3-14: MegaWizard Plug-In Manager [page 5 of 5] Window

Click Finish button, completed the addition of ROM IP

Regarding the addition and configuration of the ROM IP of the Xilinx FPGA development board, we will not introduce it here. The configuration method is similar to the configuration of ALTERA.

Part 3.3: Write a Sine Wave Generation Program

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Sine Wave Generation Program—2 port output -10V ~ +10V
//////////////////////////////////////////////////////////////////////////////////
module ad9767_test(
    input clk,              //fpga clock
    output da1_clk,                 //DA1 clock signal

    output da1_wrt,                 //DA1 write date signal

    output [13:0] da1_data,          //DA1 data


    output da2_clk,                 //DA2 clock signal

    output da2_wrt,                 //DA2 write date signal

    output [13:0] da2_data           //DA2 data


    );


reg [9:0] rom_addr;


wire [13:0] rom_data;

wire clk_50;

wire clk_125;


assign da1_clk=clk_125;

assign da1_wrt=clk_125;

assign da1_data=rom_data;


assign da2_clk=clk_125;

assign da2_wrt=clk_125;

assign da2_data=rom_data;


//DA output sin waveform

always @(negedge clk_125)

begin

     rom_addr <= rom_addr + 1'b1 ;   //if the sampling number is 1024, the output frequency of
```

```
sine wave is 125/1024=122Khz
        // rom_addr <= rom_addr + 4 ;   // if the sampling number is 256, the output frequency of sine
wave is 125/256=488Khz
        // rom_addr <= rom_addr + 128 ; // if the sampling number is 8, the output frequency of sine
wave is 125/8=15.6Mhz

    end

    ROM ROM_inst (
      .clock     (clk_125), // input clka
      .address (rom_addr), // input [8 : 0] addra
      .q            (rom_data) // output [7 : 0] douta
    );

    PLL PLL_inst(
     .areset   (1'b0),
     .inclk0   (clk),
     .c0          (clk_50),
     .c1          (clk_125),
     .locked    ()
     );

    Endmodule
```

Figure 3-15: Sine Wave Generation Program

The program is relatively simple, a 125M DA output clock is generated by a PLL IP, and then 1024 data stored in the ROM is cyclically read and output to the DA data lines of channel 1 and channel 2. In the program, select a sine wave with different frequencies by adding 1 or adding 4, or adding 128 to the address.

Part4: Hardware Connection

The hardware connection between the AN9767module and FPGA development are easy. The 40-pin female headers of module plug into the expansion board of FPGA development kit. The figure 4-1 is the expansion IOs J1 of ALINX Series FPGA development kit AX301 and AN9767 dual port DA output module hardware connection as below: (if connected to the expansion IOs J2, need to reassign the pins)

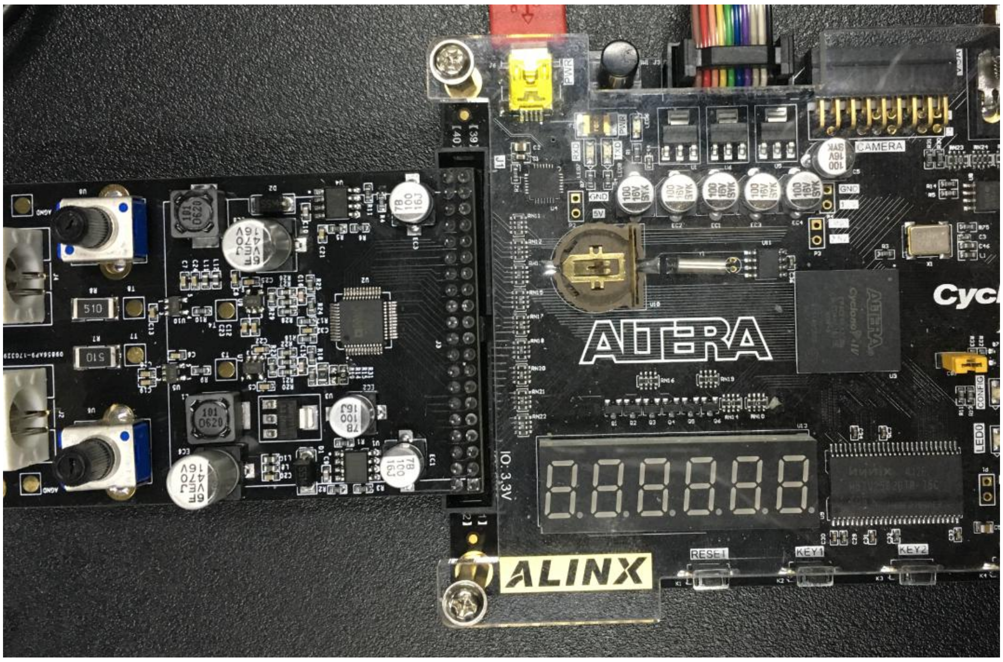


Figure 4-1: Hardware connection to ALINX Series FPGA Board

Then use the BNC cable provided to connect the analog output of the AN9767 module (channel 1 or 2) to the oscilloscope display.

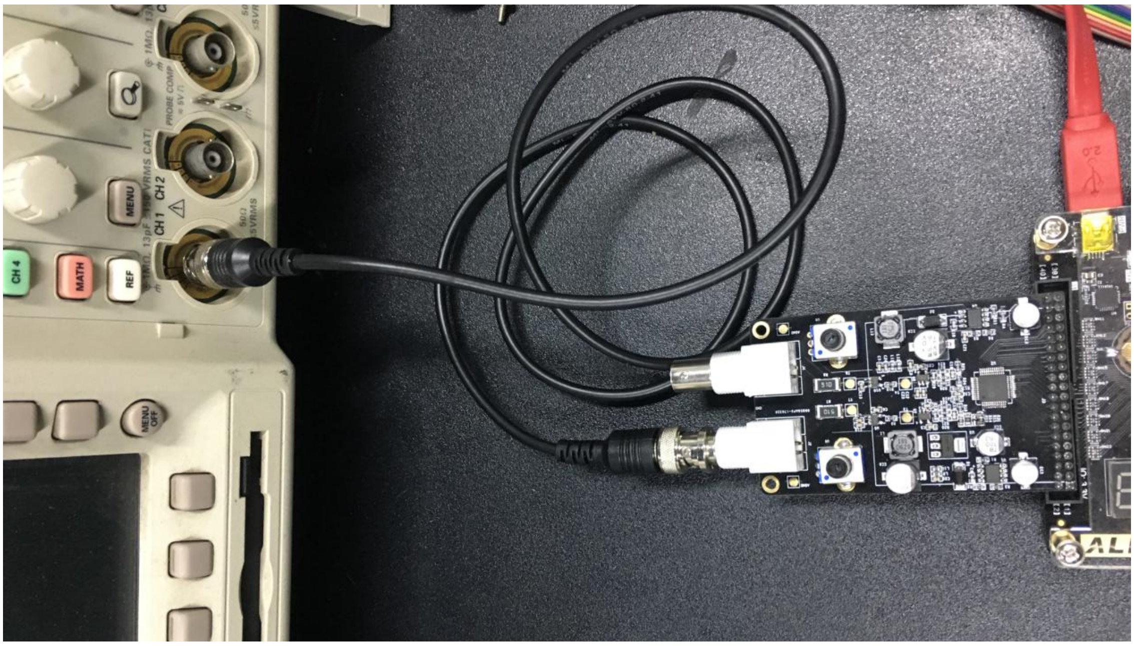


Figure 4-2: Connect the BNC Cable to oscilloscope display

Power on the FPGA Develop board, download the program, and observe the analog signal output waveform of the DA module from the oscilloscope.

Part 5: Waveform Experiment

Down load the .sof file（down load the .bit file for Xilinx FPGA board）to the FPGA development kit. if used the BNC cable to connect the AN9767 module to oscilloscope. Then we can observe the sine wave from the oscilloscope.

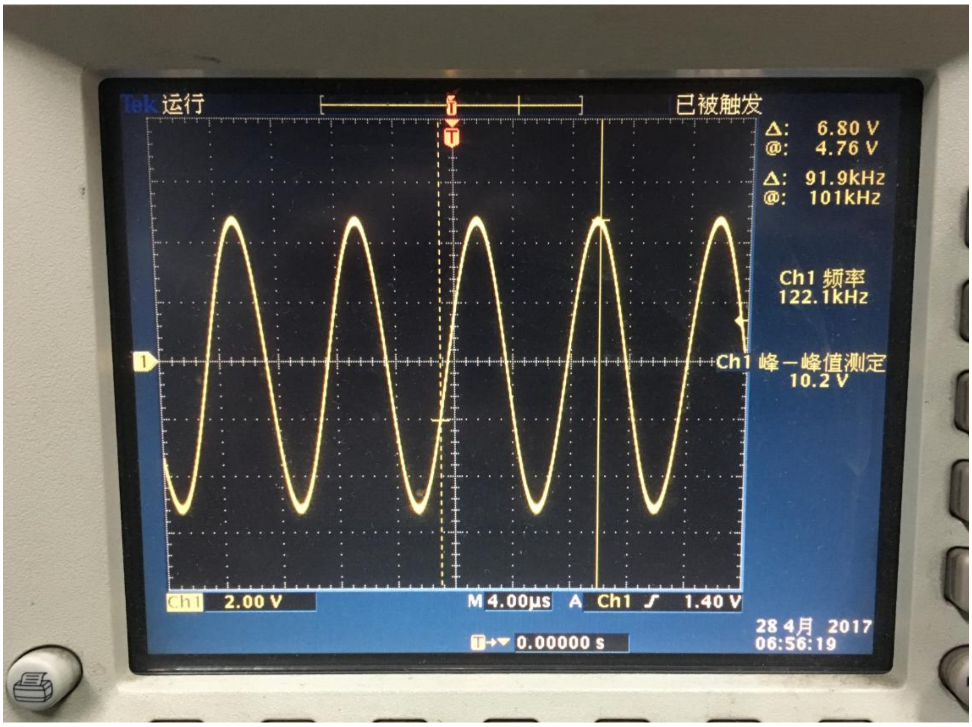
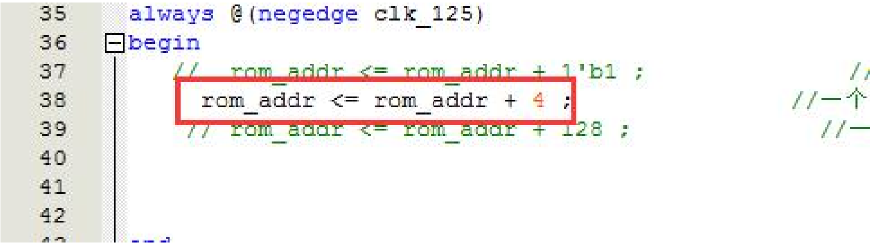


Figure 5-1: Observe the Sine Wave from the Oscilloscope

If we change the address in the program to +4, the output of the sine wave will be 256, and the frequency of the output sine wave will increase by 4 times.



```
always @(negedge clk_125)
begin
    //   rom_addr <= rom_addr + 1'b1 ;                //
      rom_addr <= rom_addr + 4 ;                    //一个
     // rom_addr <= rom_addr + 128 ;                  //一
```

Figure 5-2: Change the address in the program to +4

After the program is modified, re-download to the FPGA, the frequency of the sine wave becomes higher, and the waveform displayed by the oscilloscope is as follows:

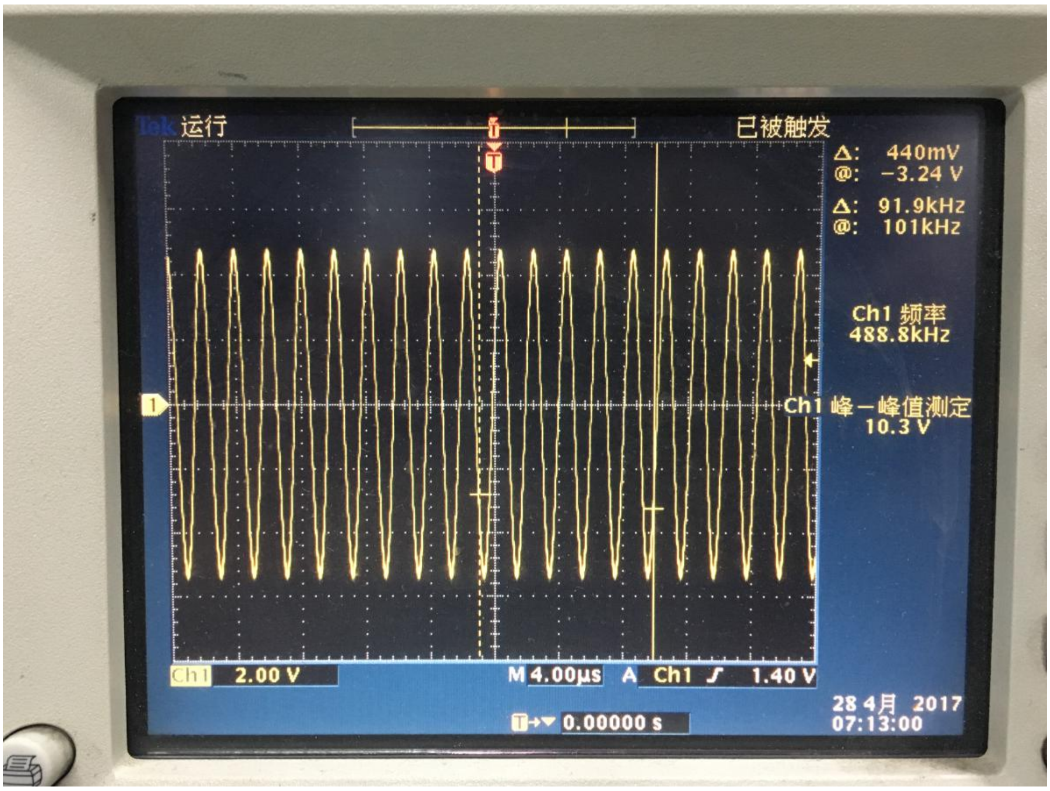


Figure 5-3: Observe the Sine Wave from the Oscilloscope

The user can also change the amplitude of the 2 channel output waveforms by adjusting the adjustable resistor on the AN9767 module.

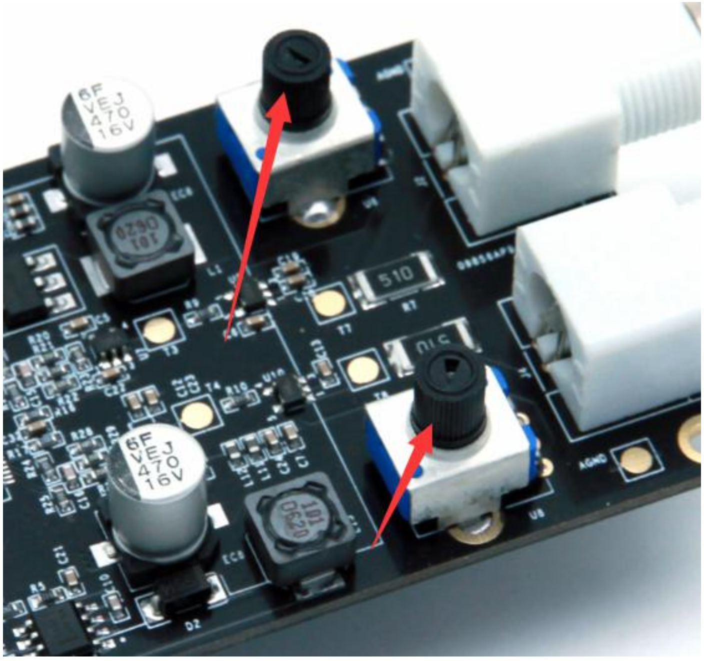


Figure 5-4: Adjustable resistor on the AN9767 module