

DE23-Lite Board

User Manual



CONTENTS

Chapter 1	<i>DE23-Lite Board</i>	3
1.1	Package Contents	4
1.2	DE23-Lite Resource Package	4
1.3	Getting Help	5
Chapter 2	Introduction to the DE23-Lite Board	6
2.1	Layout and Components	6
2.2	Block Diagram of the DE23-Lite Board	8
Chapter 3	Using the DE23-Lite Board	9
3.1	Settings of FPGA Configuration Mode	9
3.2	Configuration of Agilex 3 FPGA on DE23-Lite	10
3.3	Board Status Elements	17
3.4	Clock Circuitry	18
3.5	User Push-buttons, Switches, LEDs and 7-Segment Displays	18
3.6	7-segment Displays	21
3.7	2x20 GPIO Expansion Header	22
3.8	HDMI Output	24
3.9	SDRAM Memory	26
3.10	UART	28
3.11	A/D Converter and 2x5 Header	29
Chapter 4	Examples of Designs	30
4.1	DE23-Lite Factory Configuration	31
4.2	SDRAM Test in Verilog	32
4.3	SDRAM_Test_NiosV	34
4.4	HDMI_out RTL in Verilog	37

4.5 UART Test in Verilog	38
4.6 ADC RTL Demonstration	39
4.7 ADC NiosV Reading	43
Chapter 5 Programming the QSPI Flash	46
5.1 Programming Flow	46
5.2 Programming Steps with Batch Files	47
5.3 Programming Steps with Quartus GUI	49
Chapter 6 Appendix	59
6.1 Revision History	59
6.2 Copyright Statement	59

Chapter 1

DE23-Lite Board

The DE23-Lite is a compact and cost-effective development board powered by the Altera Agilex™ 3 FPGA, purpose-built for digital logic and embedded system design. Featuring onboard SDRAM, HDMI output, rich I/O headers, integrated ADC, and user interface components, the board provides a robust and versatile platform for both education and prototyping.

With essential connectivity—including USB Type-C, GPIO, switches, LEDs, and 7-segment displays—the DE23-Lite enables developers and students to quickly implement and test real-world designs. It offers an ideal balance of simplicity, performance, and power efficiency, making it well-suited for image and video processing, classroom instruction, and FPGA edge applications.

The DE23-Lite extends the capabilities of the previous DE10-Lite by offering upgraded I/O flexibility, improved peripheral integration, and the benefits of a modern Agilex 3 architecture—providing greater scalability for next-generation educational and prototyping needs.

1.1 Package Contents

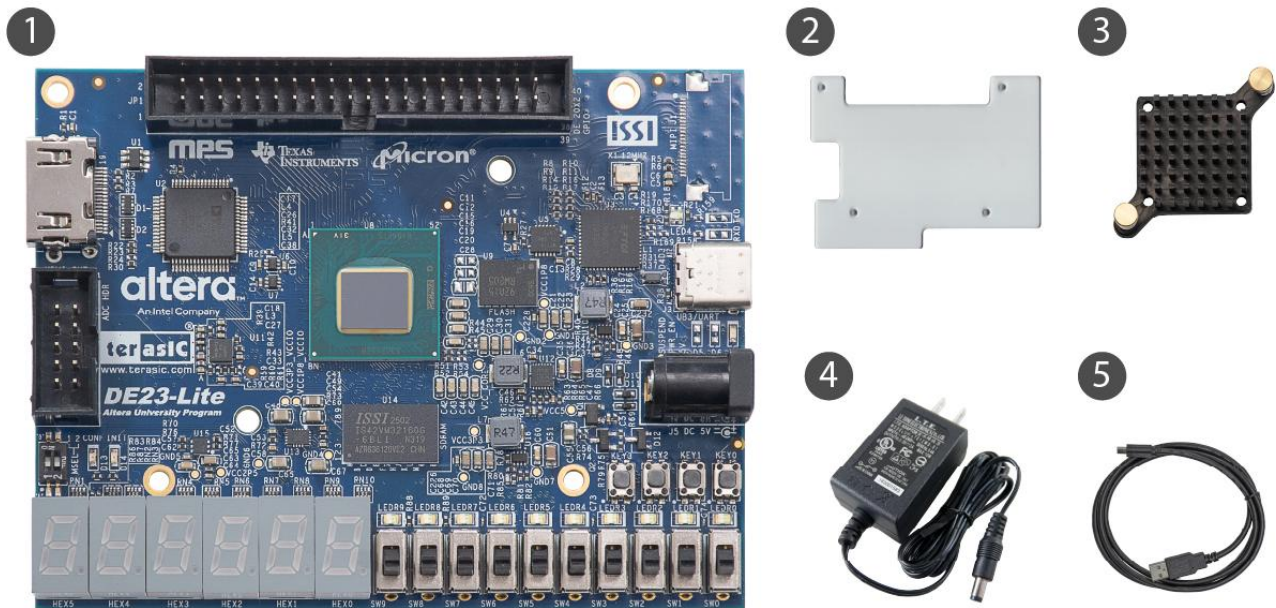


Figure 1-1 The DE23-Lite package contents

The DE23-Lite package includes:

1. DE23-Lite Board
2. Heatsink (Installed)
3. Acrylic (Installed)
4. Type-C USB Cable
5. DC 5V/2A Power Supply

1.2 DE23-Lite Resource Package

The DE23-Lite Resource Package contains all the documents and supporting materials associated with DE23-Lite, including the user manual, reference designs and device datasheets. Users can download this resource package from the link: <http://DE23-Lite.terasic.com/cd>.

The developers can create their Quartus project based on the **golden_top** Quartus project included in this Resource Package. The **golde_top** Quartus project is placed in the folder: *Demonstrations/golden_top*.

1.3 *Getting Help*

Here are the addresses where you can get help if you encounter any problems:

- Terasic Technologies
- No.80, Fenggong Rd., Hukou Township, Hsinchu County, 303035 Taiwan

Email: support@terasic.com

Tel.: +886-3-575-0880

Website: DE23-Lite.terasic.com

Chapter 2

Introduction to the DE23-Lite Board

This chapter provides an introduction to the design and features of the board.

2.1 Layout and Components

Figure 2-1 and Figure 2-2 shows a photograph of the board that illustrates its layout and the location of connectors and key components.

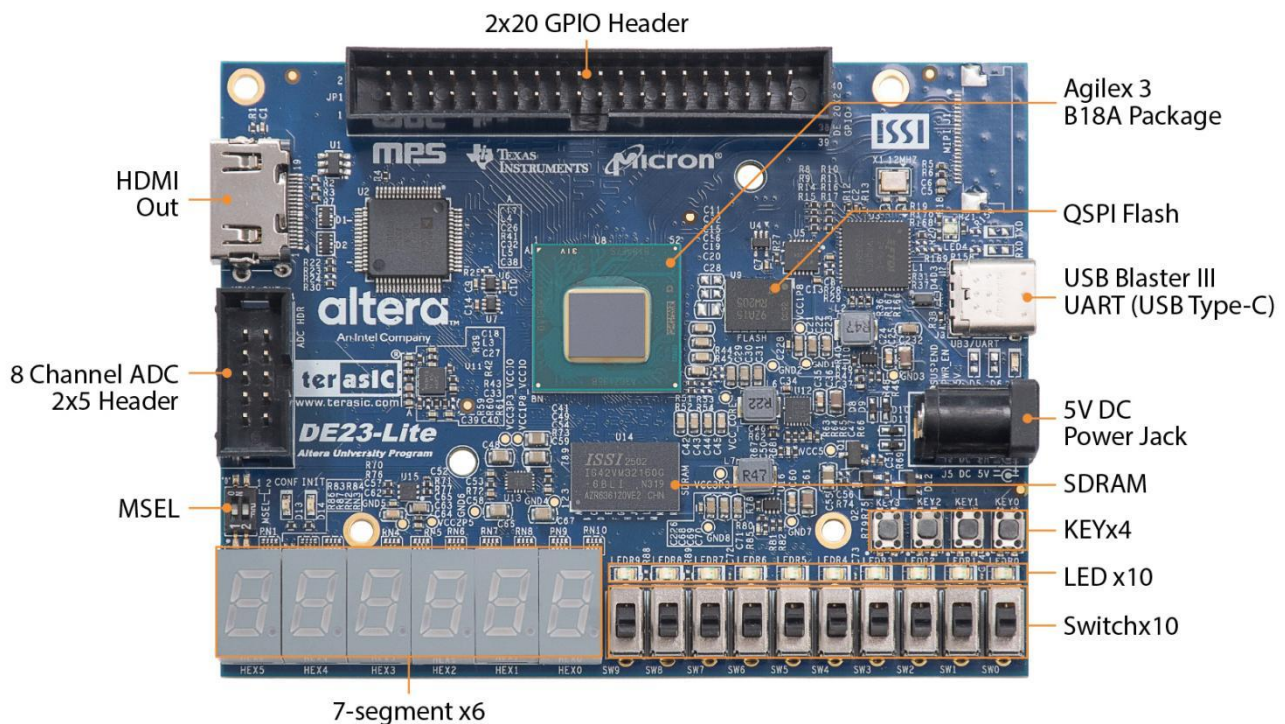


Figure 2-1 DE23-Lite board (top view)

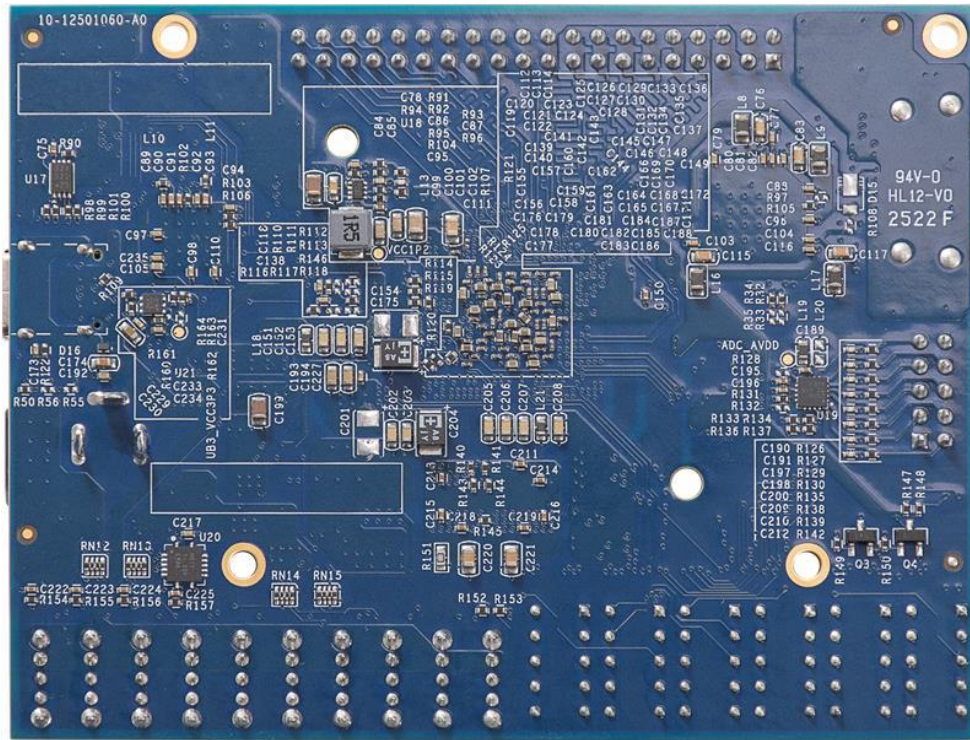


Figure 2-2 DE23-Lite board (bottom view)

The DE23-Lite board has many features that allow users to implement a wide range of designed circuits, from simple circuits to multimedia projects.

The following hardware is provided on the board:

- **FPGA: Agilex 3 A3CZ135BB18AE7S Device**
 - 135,110 Logic Elements
 - 6.89 Mbit M20K, 1.4 Mbit MLAB
 - 368 18x19 Multipliers
 - 4 IO PLL, 8 Fabric I/O PLL
- **Programming/Debug and Configuration**
 - On-Board USB Blaster III (Type C USB connector)
 - ASx4 Mode with 128Mbit QSPI Flash
- **Memory Device**
 - 64MB SDRAM, x32 bits data bus
- **Communiation**
 - UART
- **Display**
 - HDMI Output

- **Expansion Connectors**
 - 2x5 ADC Header
 - One 2x20 GPIO Connector (voltage level: 3.3V)
- **Switches/Buttons/LEDs/7-Segment**
 - 10 LEDs
 - 10 Slide Switches
 - 6 7-Segment Display
 - 4 Push Buttons
- **Power**
 - 5V DC input or Type-C Connector

2.2 Block Diagram of the DE23-Lite Board

Figure 2-3 is the block diagram of the board. All the connections are established through the Agilex 3 FPGA device to provide maximum flexibility for users. Users can configure the FPGA to implement any system design.

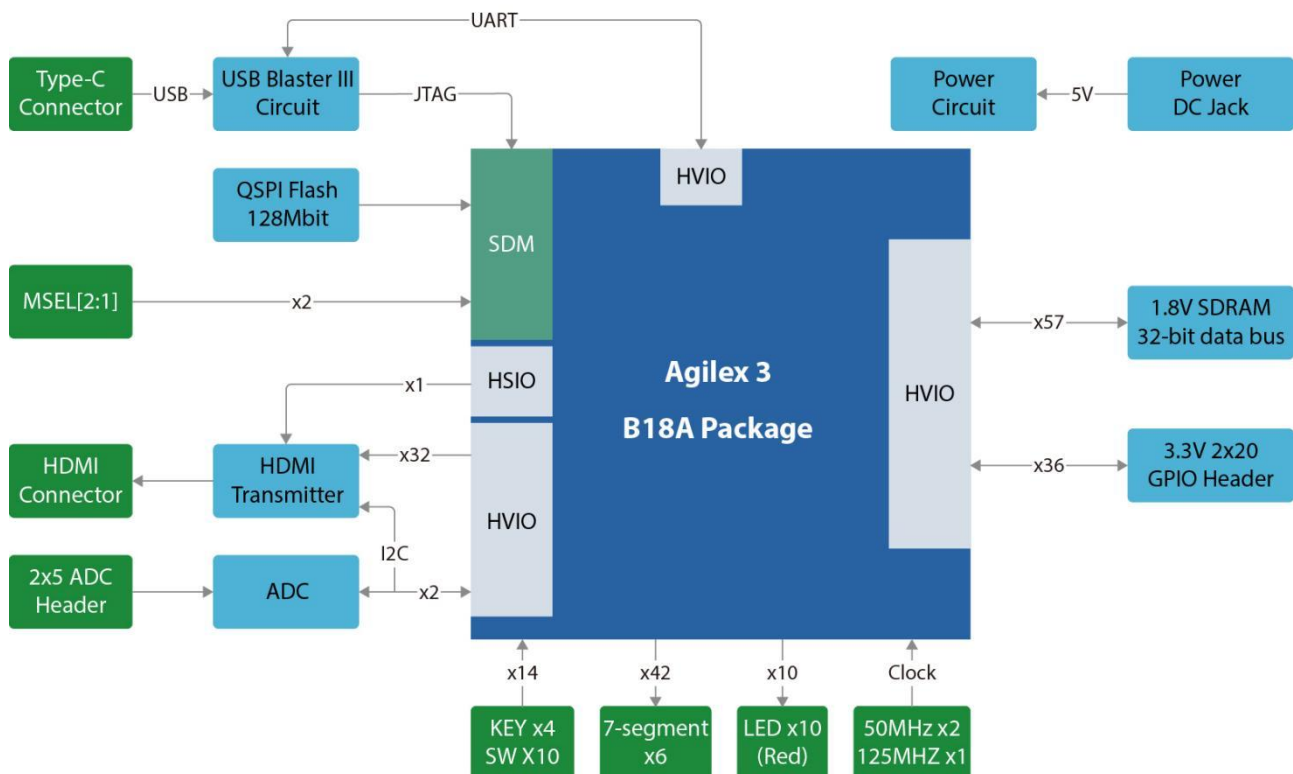


Figure 2-3 Block diagram of DE23-Lite

The components shown in Figure 2-3 are described in detail in Section 2.1.

Chapter 3

Using the DE23-Lite Board

This chapter provides instructions for using the board and describes its peripherals.

3.1 Settings of FPGA Configuration Mode

Figure 3-1 shows MSEL[2:1] setting of Active Serial (AS) Fast mode, which is the default setting on the DE23-Lite. When the board is powered on, the FPGA is configured from QSPI Flash, which is pre-programmed with a default configuration.

The MSEL[2:1] switches are used to select the configuration scheme, implemented as a 2-pin DIP switch **SW10** on the DE23-Lite board.

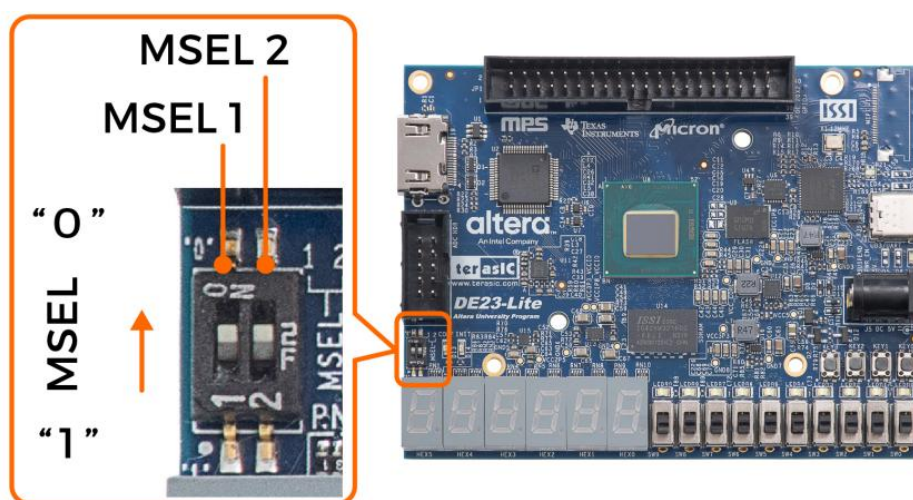


Figure 3-1 DIP switch (SW10) setting of Active Serial (AS) mode on DE23-Lite board

Table 3-1 shows the relation between MSEL[2:1] and DIP switch (SW10).

Table 3-1 FPGA Configuration Mode Switch (SW10)

<i>Board Reference</i>	<i>Signal Name</i>	<i>Description</i>	<i>Default AS Mode</i>
SW10.1	MSEL1	Use these pins to set the FPGA	ON ("0")
SW10.2	MSEL2	Configuration scheme	ON ("0")

Table 3-2 shows the MSEL description.

Table 3-2 MSEL Pin Settings for FPGA Configuration of DE23-Lite

<i>MSEL[2:1]</i>	<i>Configuration Scheme</i>	<i>Description</i>
00	AS Fast	FPGA configured from QSPI Flash (default)
11	JTAG	You can configure the FPGA using the dedicated JTAG interface and circuit.

3.2 Configuration of Agilex 3 FPGA on DE23-Lite

There are two programming methods supported by DE23-Lite:

1. JTAG programming: This is named after the IEEE standards Joint Test Action Group. The configuration bitstream is downloaded directly to the Agilex 3 FPGA. The FPGA will retain its status as long as the power is applied to the board; the configuration information will be lost when the power is turned off.
2. AS programming: The other programming method is Active Serial configuration. The configuration bitstream is written to the quad serial configuration device (QSPI Flash), which provides non-volatile storage for the bit stream. The information is retained within QSPI Flash even if the DE23-Lite board is turned off. When the board is powered on, the configuration data in the QSPI Flash device is automatically loaded into the Agilex 3 FPGA.

■ JTAG Chain on DE23-Lite Board

Figure 3-2 Block diagram of the JTAG chain shows the JTAG interface of DE23-Lite board, which uses the USB Blaster III circuit to connect to the host PC. Users can configure or debug the Agilex 3 FPGA on the board through the USB Type-C interface and Quartus software on the host PC.

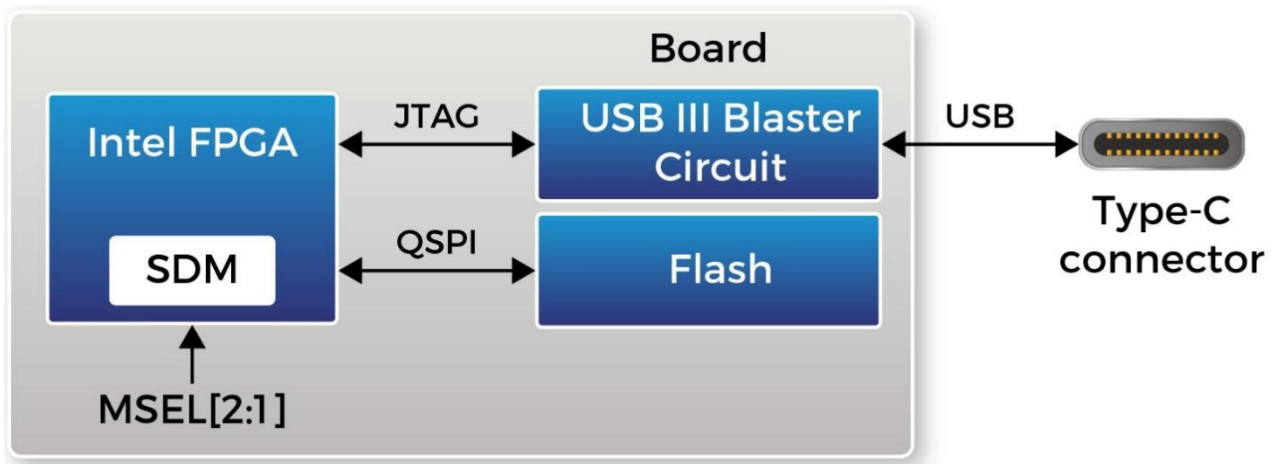


Figure 3-2 Block diagram of the JTAG chain

■ Configuring the FPGA in JTAG Mode

The following explains step-by-step how to program the FPGA in JTAG mode.

1. Make sure the Quartus Pro v25.1 and the driver of **USB Blaster III** are installed on your Host.
2. Set MSEL to 11(JTAG programming mode), use a Type-C USB cable to connected the board(J3) to PC, and plug the 5V DC adapter to J5. Open the Quartus Programmer tool, make sure the USB blaster III (“DE23-Lite[USB-x]”) is found in “Hardware Setup..” tab(as shown in **Figure 3-3**).

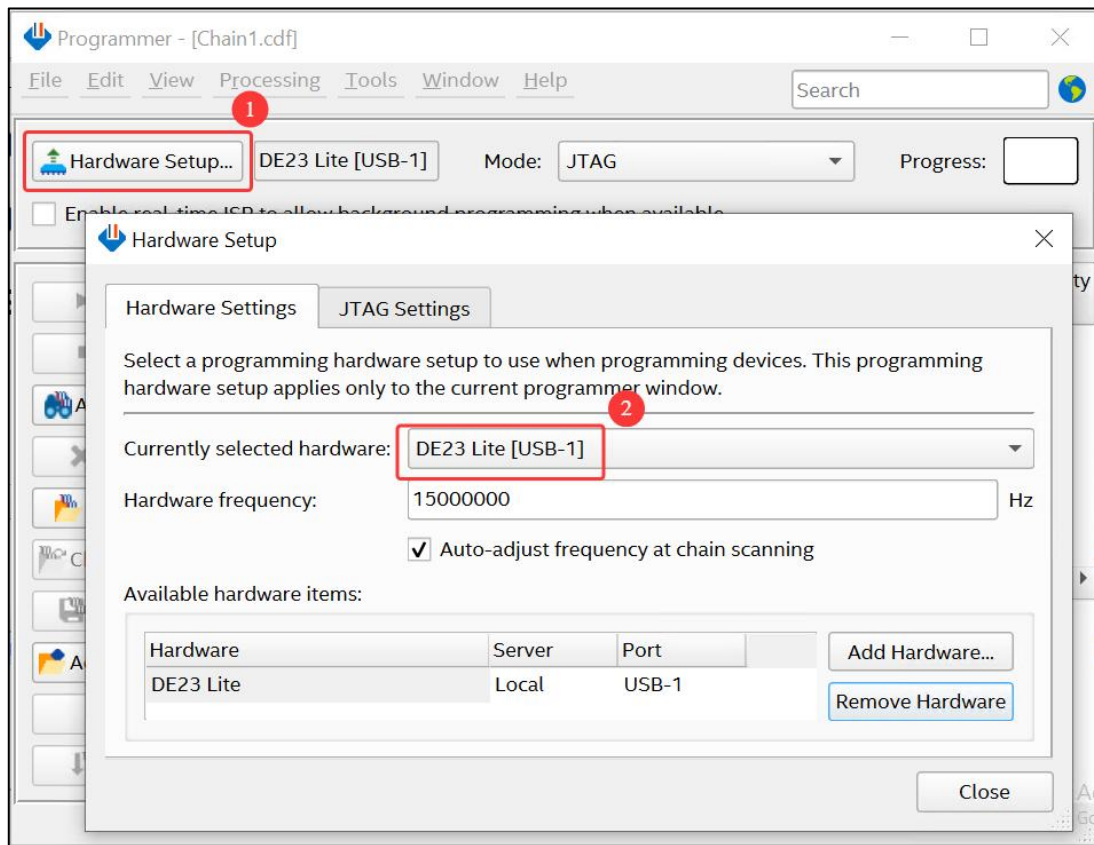


Figure 3-3 USB blaster III is found in Programmer

3. Open the Programmer and click “**Auto Detect**”, as shown in **Figure 3-4**.

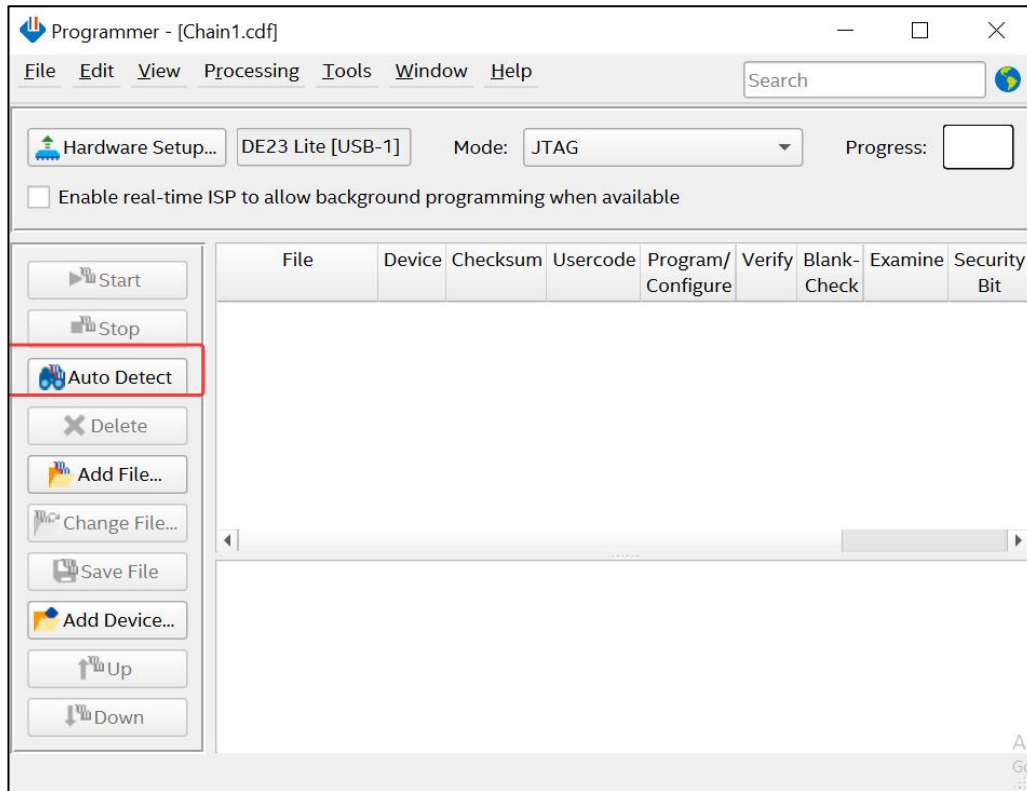


Figure 3-4 Detect FPGA device in JTAG mode

4. The Agilex 3 FPGA should be detected, as shown in [Figure 3-5](#).

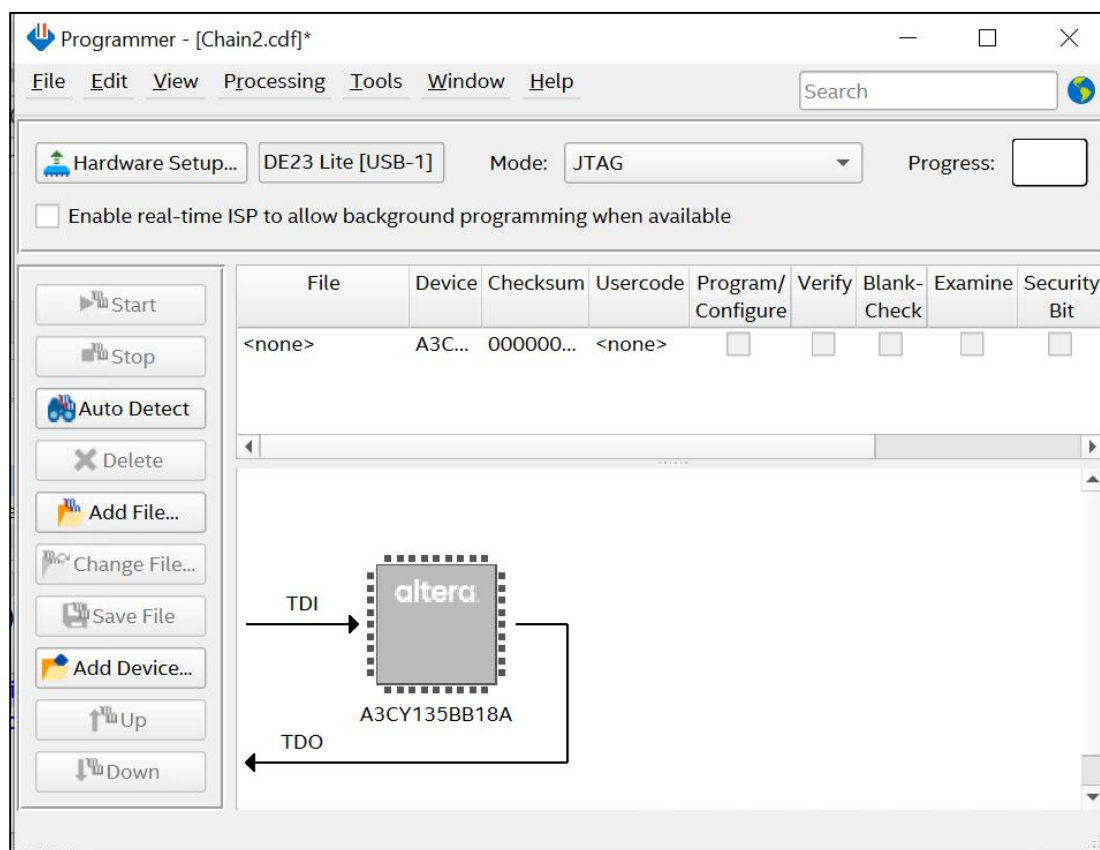


Figure 3-5 FPGA detected in Quartus programmer

5. Right click on the FPGA device and open the .sof file to be programmed, as shown in **Figure 3-6**.

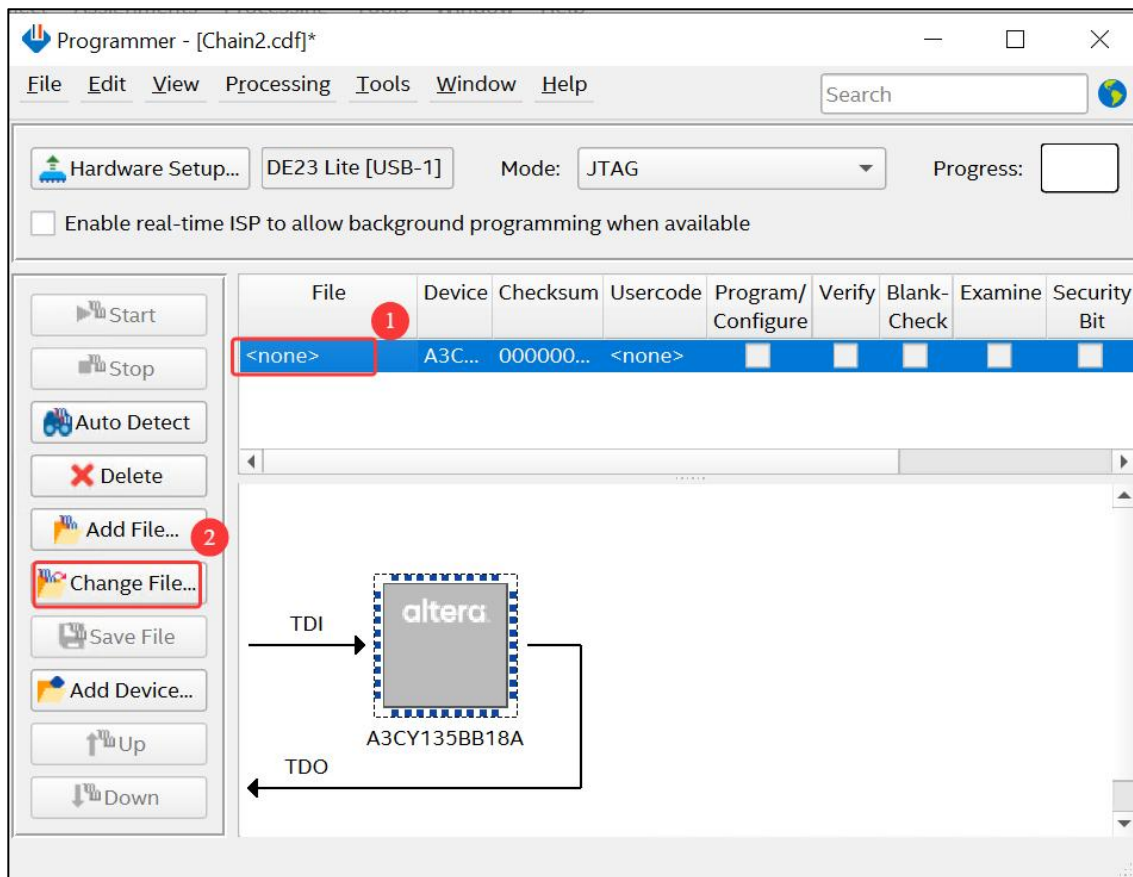


Figure 3-6 Open the .sof file to be programmed into the FPGA device

6. Select the .sof file to be programmed, as shown in **Figure 3-7**.

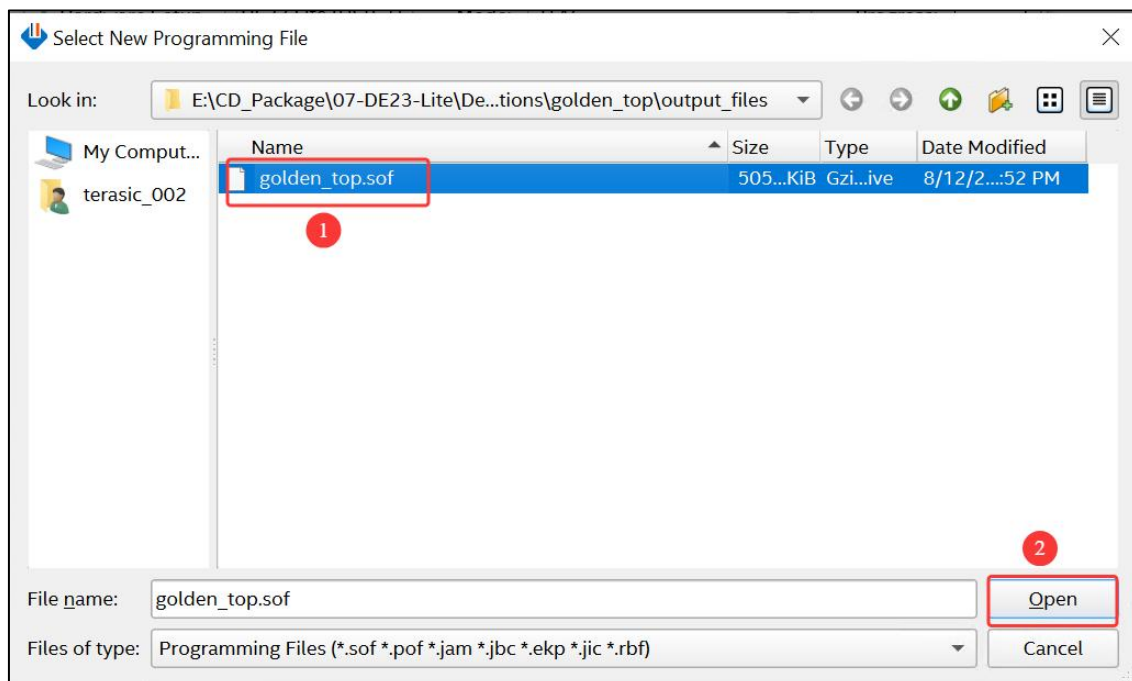


Figure 3-7 Select the .sof file to be programmed into the FPGA device

7. Click the “Program/Configure” checkbox and click “Start” button to download the .sof file into the FPGA device, as shown in **Figure 3-8** and **Figure 3-9**.

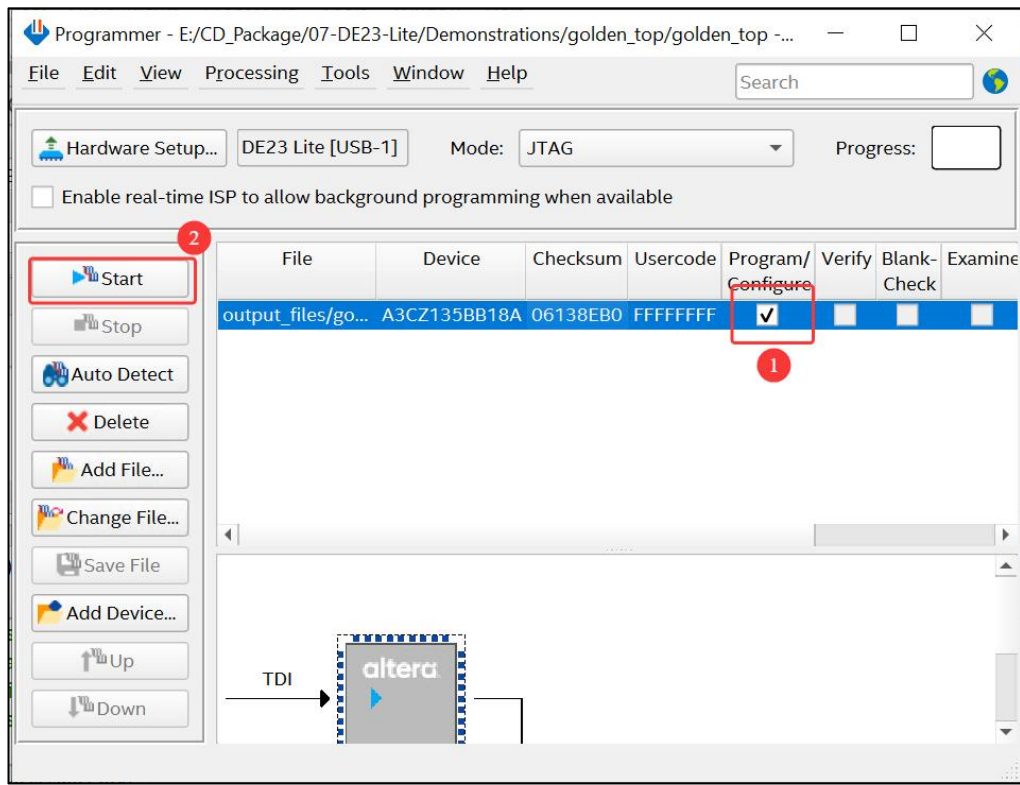


Figure 3-8 Program .sof file into the FPGA device

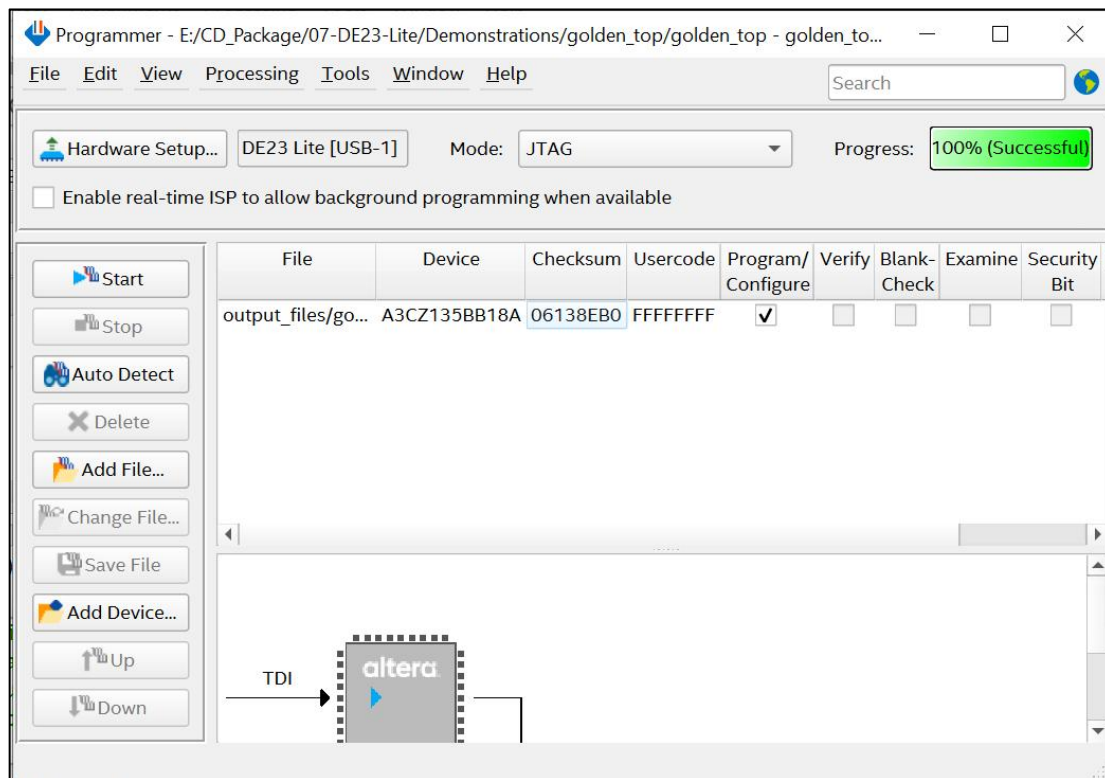


Figure 3-9 Program .sof file successfully

3.3 Board Status Elements

In addition to the four LEDs that the FPGA can control, there are three indicators that indicate the board status (see Figure 3-10). Table 3-3 lists the details. There also is one Jtag indicator that indicate the Jtag status(see Table 3-4).

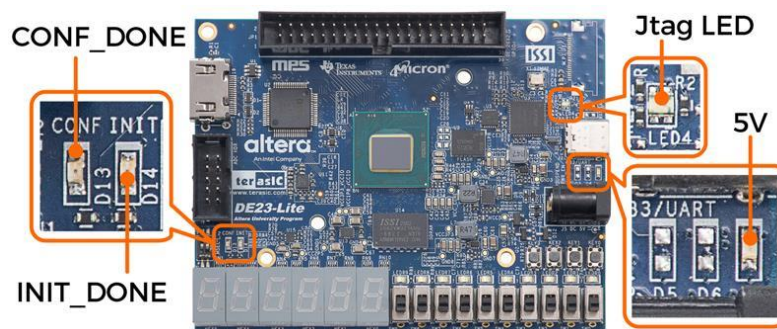


Figure 3-10 LED Indicators on the DE23-Lite

Table 3-3 LED Indicators

Board Reference	LED Name	Description
D7	5V	Illuminates when 5V power is active.
D14	INIT_DONE	Illuminates when the board initialization process finished for user

		model.
D13	CONF_DONE	Illuminates when the FPGA is successfully configured.

Table 3-4 JTAG blaster LED(LED4) status

Color	Meaning
Off	No power / not connected / suspend mode
Blue	Connected, not in use
Green	Connected, an application is using JTAG, no traffic
Green flickering	Connected, data is moving through the JTAG interface
Purple flashing	Identify function has been triggered on this cable

3.4 Clock Circuitry

Figure 3-11 shows the default frequency of all external clocks fed to the Agilex 3 FPGA. Two programmable BAW clock generators are used to distribute clock signals with low jitter. The four 50MHz clock signals connected to the FPGA are used as clock sources for user logic. A 125MHz clock is used for FPGA configuration bank (OSC_CLK1). For peripheral devices, one 25MHz clock is fed to the clock input of Gigabit Ethernet PHY. The pin assignments of clock inputs to FPGA I/O pins are listed in **Table 3-5**.

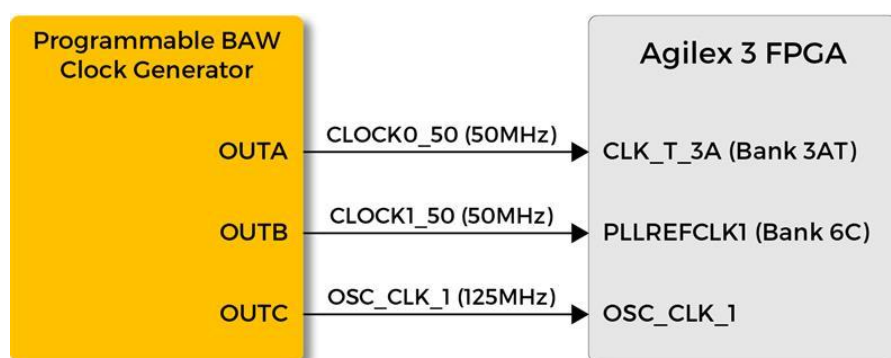


Figure 3-11 Block diagram of the clock distribution on the DE23-Lite

Table 3-5 Pin Assignment of Clock Inputs

Signal Name	FPGA Pin No.	Description	I/O Standard
CLOCK0_50	PIN_K43	50 MHz clock input	1.2V
CLOCK1_50	PIN_A8	50 MHz clock input	3.3-V LVCMOS

3.5 User Push-buttons, Switches, LEDs and 7-Segment Displays

The board has four push-buttons connected to the FPGA, as shown in **Figure 3-12**. A Schmitt

trigger circuit acts as a switch debouncer for the connected buttons. The four buttons named KEY0, KEY1, KEY2, and KEY3 coming out of the Schmitt trigger device are connected directly to the Agilex 3 FPGA. Each push-button generates a low logic level when it is pressed (Active low). Since the push-buttons are debounced, they can be used as reset inputs in a circuit.

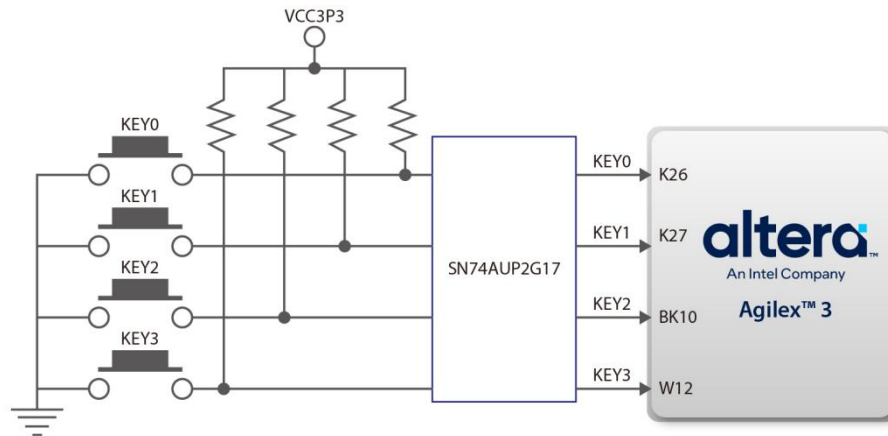


Figure 3-12 Connections between the push-buttons and the Agilex 3 FPGA

There are ten slide switches connected to the Agilex 3 FPGA, as shown in **Figure 3-13**. These switches are not debounced and may be used as level-sensitive data inputs to a circuit. Each switch is connected directly and individually to the FPGA. When the switch is set to the DOWN position (towards the edge of the board), it sends a low logic level to the FPGA. When the switch is set to the UP position, a high logic level is sent to the FPGA.

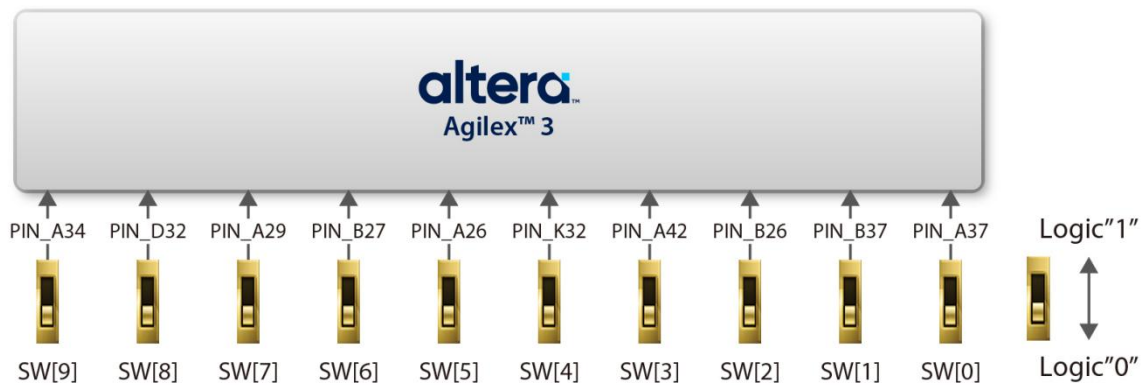


Figure 3-13 Connections between the slide switches and the Agilex 3 FPGA

There are also ten user-controllable LEDs connected to the FPGA. Each LED is driven directly and individually by the Agilex 3 FPGA; driving its associated pin to a “**low**” logic level turn the LED “**on**”. **Figure 3-14** shows the connections between LEDs and Agilex 3 FPGA. **Table 3-6**, **Table 3-7** and **Table 3-8** list the pin assignment of user push-buttons, switches, and LEDs.

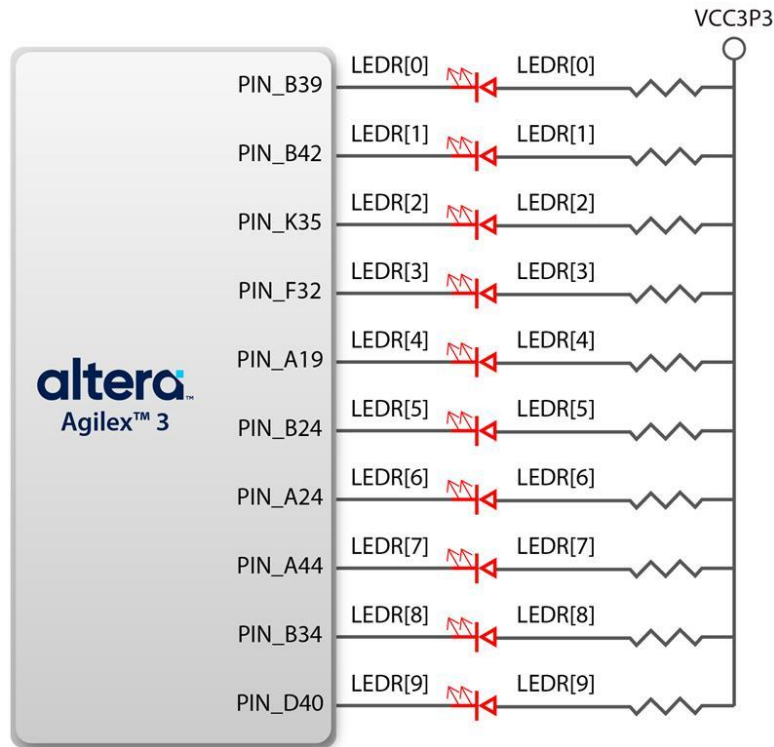


Figure 3-14 Connections between the LEDs and the Agilex 3 FPGA

Table 3-6 Pin Assignments of Slide Switches

Signal Name	FPGA Pin No.	Description	I/O Standard
SW[0]	PIN_A37	Slide Switch[0]	1.2V
SW[1]	PIN_B37	Slide Switch[1]	1.2V
SW[2]	PIN_B26	Slide Switch[2]	1.2V
SW[3]	PIN_A42	Slide Switch[3]	1.2V
SW[4]	PIN_K32	Slide Switch[4]	1.2V
SW[5]	PIN_A26	Slide Switch[5]	1.2V
SW[6]	PIN_B27	Slide Switch[6]	1.2V
SW[7]	PIN_A29	Slide Switch[7]	1.2V
SW[8]	PIN_D32	Slide Switch[8]	1.2V
SW[9]	PIN_A34	Slide Switch[9]	1.2V

Table 3-7 Pin Assignments of Push-buttons

Signal Name	FPGA Pin No.	Description	I/O Standard
KEY[0]	PIN_K26	Push-button[0]	3.3V
KEY[1]	PIN_K27	Push-button[1]	3.3V
KEY[2]	PIN_BK10	Push-button[2]	3.3V
KEY[3]	PIN_W12	Push-button[3]	3.3V

Table 3-8 Pin Assignments of LEDs

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR[0]	PIN_B39	Red LED [0]	3.3V
LEDR[1]	PIN_B42	Red LED [1]	3.3V
LEDR[2]	PIN_K35	Red LED [2]	3.3V
LEDR[3]	PIN_F32	Red LED [3]	3.3V
LEDR[4]	PIN_A19	Red LED [4]	3.3V
LEDR[5]	PIN_B24	Red LED [5]	3.3V
LEDR[6]	PIN_A24	Red LED [6]	3.3V
LEDR[7]	PIN_A44	Red LED [7]	3.3V
LEDR[8]	PIN_B34	Red LED [8]	3.3V
LEDR[9]	PIN_D40	Red LED [9]	3.3V

3.6 7-segment Displays

The DE23-Lite board has six 7-segment displays ideal for displaying numbers. **Figure 3-15** shows the connection of the seven segments (common anode) to pins on the Agilex 3 FPGA. Each segment can be turned on by applying a low logic level from the FPGA.

Each segment in a display is indexed from 0 to 6, with corresponding positions given in **Figure 3-15**. **Table 3-9** shows the FPGA pin assignments for the 7-segment displays.

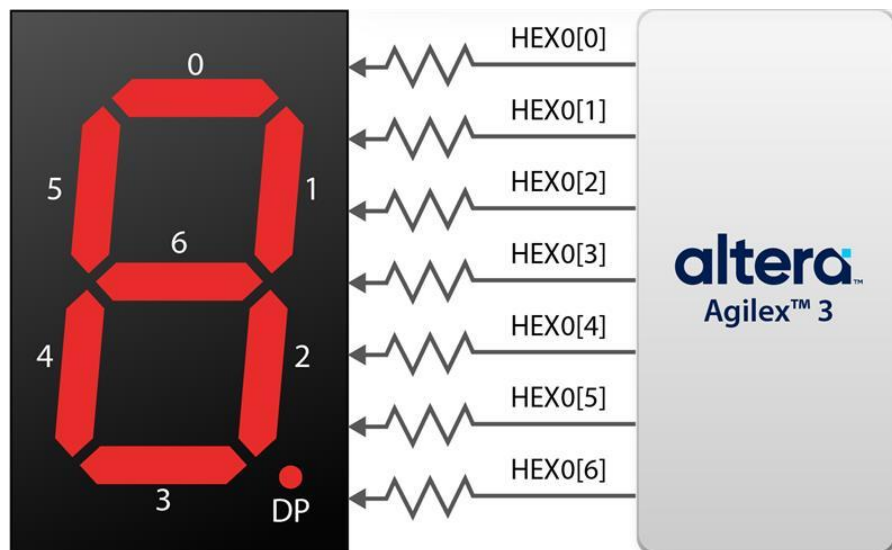


Figure 3-15 Connections between the 7-segment display HEX0 and the Agilex 5 SoC FPGA

Table 3-9 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_L51	Seven Segment Digit 0[0]	1.2V
HEX0[1]	PIN_J51	Seven Segment Digit 0[1]	1.2V
HEX0[2]	PIN_G51	Seven Segment Digit 0[2]	1.2V

HEX0[3]	PIN_J52	Seven Segment Digit 0[3]	1.2V
HEX0[4]	PIN_M51	Seven Segment Digit 0[4]	1.2V
HEX0[5]	PIN_M52	Seven Segment Digit 0[5]	1.2V
HEX0[6]	PIN_R51	Seven Segment Digit 0[6]	1.2V
HEX1[0]	PIN_F40	Seven Segment Digit 1[0]	1.2V
HEX1[1]	PIN_K40	Seven Segment Digit 1[1]	1.2V
HEX1[2]	PIN_N52	Seven Segment Digit 1[2]	1.2V
HEX1[3]	PIN_V51	Seven Segment Digit 1[3]	1.2V
HEX1[4]	PIN_R52	Seven Segment Digit 1[4]	1.2V
HEX1[5]	PIN_V52	Seven Segment Digit 1[5]	1.2V
HEX1[6]	PIN_G52	Seven Segment Digit 1[6]	1.2V
HEX2[0]	PIN_H43	Seven Segment Digit 2[0]	1.2V
HEX2[1]	PIN_F35	Seven Segment Digit 2[1]	1.2V
HEX2[2]	PIN_A31	Seven Segment Digit 2[2]	1.2V
HEX2[3]	PIN_B29	Seven Segment Digit 2[3]	1.2V
HEX2[4]	PIN_H35	Seven Segment Digit 2[4]	1.2V
HEX2[5]	PIN_BN24	Seven Segment Digit 2[5]	1.2V
HEX2[6]	PIN_BN19	Seven Segment Digit 2[6]	1.2V
HEX3[0]	PIN_BM22	Seven Segment Digit 3[0]	1.2V
HEX3[1]	PIN_BH13	Seven Segment Digit 3[1]	1.2V
HEX3[2]	PIN_BM3	Seven Segment Digit 3[2]	1.2V
HEX3[3]	PIN_AY6	Seven Segment Digit 3[3]	1.2V
HEX3[4]	PIN_AU6	Seven Segment Digit 3[4]	1.2V
HEX3[5]	PIN_AU4	Seven Segment Digit 3[5]	1.2V
HEX3[6]	PIN_AW12	Seven Segment Digit 3[6]	1.2V
HEX4[0]	PIN_AT9	Seven Segment Digit 4[0]	1.2V
HEX4[1]	PIN_AT12	Seven Segment Digit 4[1]	1.2V
HEX4[2]	PIN_AP9	Seven Segment Digit 4[2]	1.2V
HEX4[3]	PIN_BK7	Seven Segment Digit 4[3]	1.2V
HEX4[4]	PIN_BM2	Seven Segment Digit 4[4]	1.2V
HEX4[5]	PIN_BL2	Seven Segment Digit 4[5]	1.2V
HEX4[6]	PIN_BH7	Seven Segment Digit 4[6]	1.2V
HEX5[0]	PIN_BH3	Seven Segment Digit 5[0]	1.2V
HEX5[1]	PIN_BF12	Seven Segment Digit 5[1]	1.2V
HEX5[2]	PIN_BF9	Seven Segment Digit 5[2]	1.2V
HEX5[3]	PIN_BB12	Seven Segment Digit 5[3]	1.2V
HEX5[4]	PIN_BB9	Seven Segment Digit 5[4]	1.2V
HEX5[5]	PIN_E2	Seven Segment Digit 5[5]	1.2V
HEX5[6]	PIN_K3	Seven Segment Digit 5[6]	1.2V

3.7 2x20 GPIO Expansion Header

The board has one 40-pin expansion header. The header has 36 user pins connected directly to the

Agilex 3 FPGA. It also includes DC +5V (VCC5), DC +3.3V (VCC3P3), and two GND pins. The maximum power consumption allowed for a daughter card connected to one GPIO ports is shown in [Table 3-10](#).

Table 3-10 Voltage and Max. Current Limit of Expansion Header

Supplied Voltage	Max. Current Limit
5V	1A
3.3V	1.5A

[Figure 3-16](#) shows the I/O distribution of the GPIO connector, and the [Figure 3-17](#) indicates the pin 1 location of the expansion headers. [Table 3-11](#) shows the pin assignment of the GPIO header.

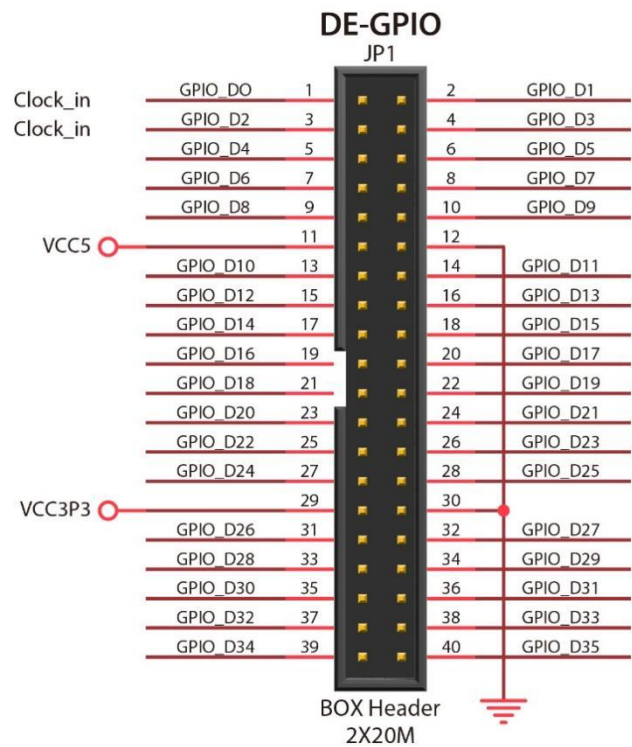


Figure 3-16 Pin arrangement of the GPIO expansion header

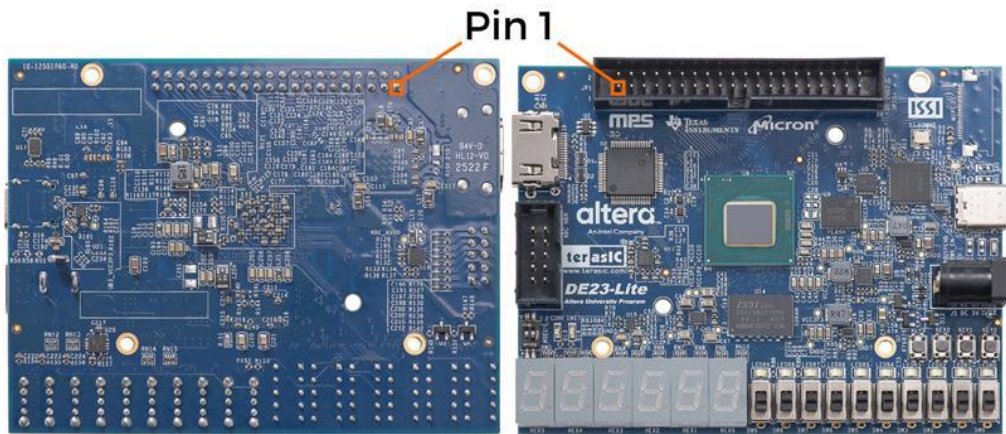


Figure 3- 17 Pin1 location of the GPIO expansion header

Table 3- 11 Pin Assignment of Expansion Header

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
GPIO_D[0]	PIN_B3	GPIO Connection 0[0]	3.3V
GPIO_D[1]	PIN_C2	GPIO Connection 0[1]	3.3V
GPIO_D[2]	PIN_D18	GPIO Connection 0[2]	3.3V
GPIO_D[3]	PIN_F3	GPIO Connection 0[3]	3.3V
GPIO_D[4]	PIN_W9	GPIO Connection 0[4]	3.3V
GPIO_D[5]	PIN_K7	GPIO Connection 0[5]	3.3V
GPIO_D[6]	PIN_F7	GPIO Connection 0[6]	3.3V
GPIO_D[7]	PIN_H7	GPIO Connection 0[7]	3.3V
GPIO_D[8]	PIN_B8	GPIO Connection 0[8]	3.3V
GPIO_D[9]	PIN_B5	GPIO Connection 0[9]	3.3V
GPIO_D[10]	PIN_K10	GPIO Connection 0[10]	3.3V
GPIO_D[11]	PIN_F10	GPIO Connection 0[11]	3.3V
GPIO_D[12]	PIN_F13	GPIO Connection 0[12]	3.3V
GPIO_D[13]	PIN_D10	GPIO Connection 0[13]	3.3V
GPIO_D[14]	PIN_K13	GPIO Connection 0[14]	3.3V
GPIO_D[15]	PIN_H13	GPIO Connection 0[15]	3.3V
GPIO_D[16]	PIN_U9	GPIO Connection 0[16]	3.3V
GPIO_D[17]	PIN_D3	GPIO Connection 0[17]	3.3V
GPIO_D[18]	PIN_B11	GPIO Connection 0[18]	3.3V
GPIO_D[19]	PIN_A9	GPIO Connection 0[19]	3.3V
GPIO_D[20]	PIN_B14	GPIO Connection 0[20]	3.3V
GPIO_D[21]	PIN_A11	GPIO Connection 0[21]	3.3V
GPIO_D[22]	PIN_U12	GPIO Connection 0[22]	3.3V
GPIO_D[23]	PIN_A14	GPIO Connection 0[23]	3.3V
GPIO_D[24]	PIN_B19	GPIO Connection 0[24]	3.3V
GPIO_D[25]	PIN_B16	GPIO Connection 0[25]	3.3V
GPIO_D[26]	PIN_F18	GPIO Connection 0[26]	3.3V
GPIO_D[27]	PIN_P9	GPIO Connection 0[27]	3.3V
GPIO_D[28]	PIN_F26	GPIO Connection 0[28]	3.3V
GPIO_D[29]	PIN_D26	GPIO Connection 0[29]	3.3V
GPIO_D[30]	PIN_K21	GPIO Connection 0[30]	3.3V
GPIO_D[31]	PIN_F21	GPIO Connection 0[31]	3.3V
GPIO_D[32]	PIN_H27	GPIO Connection 0[32]	3.3V
GPIO_D[33]	PIN_H21	GPIO Connection 0[33]	3.3V
GPIO_D[34]	PIN_K18	GPIO Connection 0[34]	3.3V
GPIO_D[35]	PIN_F27	GPIO Connection 0[35]	3.3V

3.8 HDMI Output

The development board provides a high performance HDMI transmitter via the Analog Devices ADV7513, which incorporates HDMI v1.4 features, including 3D video support and 165MHz

support for all video formats up to 1080p and UXGA. The ADV7513 is controlled via a serial I2C bus interface, which is connected to pins on the Agilex 3 FPGA. A schematic diagram of the audio circuitry is shown in **Figure 3-18**. Detailed information on using the ADV7513 HDMI TX is available on the manufacturer's website, or under the Datasheets\HDMI folder in the DE23-Lite Resource Package.

Table 3-12 lists the HDMI Interface pin assignments and signal names relative to the FPGA.

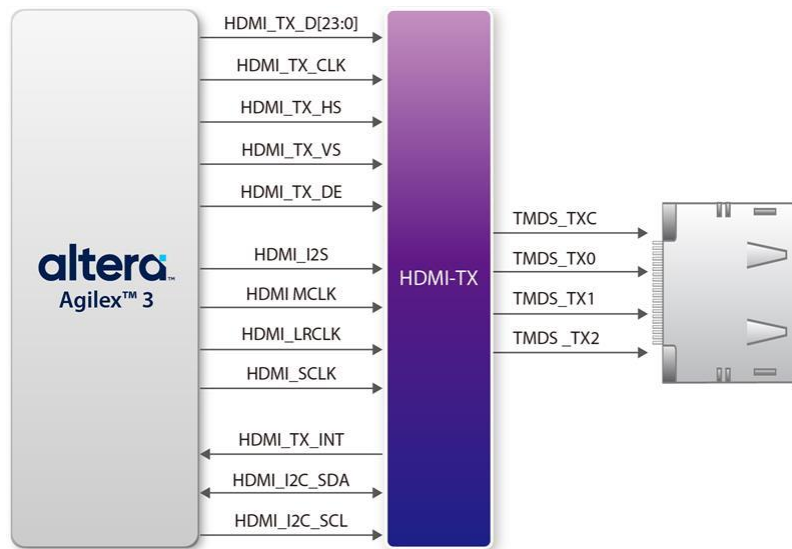


Figure 3-18 Connections between the FPGA and HDMI Transmitter Chip

Table 3-12 Pin Assignment of HDMI

Signal Name	FPGA Pin No.	Description	I/O Standard
HDMI_TX_D0	PIN_AC2	Video Data bus	3.3V
HDMI_TX_D1	PIN_N1	Video Data bus	3.3V
HDMI_TX_D2	PIN_P4	Video Data bus	3.3V
HDMI_TX_D3	PIN_V2	Video Data bus	3.3V
HDMI_TX_D4	PIN_P6	Video Data bus	3.3V
HDMI_TX_D5	PIN_AA6	Video Data bus	3.3V
HDMI_TX_D6	PIN_AD6	Video Data bus	3.3V
HDMI_TX_D7	PIN_AA4	Video Data bus	3.3V
HDMI_TX_D8	PIN_AB9	Video Data bus	3.3V
HDMI_TX_D9	PIN_AC1	Video Data bus	3.3V
HDMI_TX_D10	PIN_AF2	Video Data bus	3.3V
HDMI_TX_D11	PIN_AF1	Video Data bus	3.3V
HDMI_TX_D12	PIN_AG2	Video Data bus	3.3V
HDMI_TX_D13	PIN_AK1	Video Data bus	3.3V
HDMI_TX_D14	PIN_AK2	Video Data bus	3.3V
HDMI_TX_D15	PIN_AM6	Video Data bus	3.3V
HDMI_TX_D16	PIN_AR1	Video Data bus	3.3V

HDMI_TX_D17	PIN_AR2	Video Data bus	3.3V
HDMI_TX_D18	PIN_AV1	Video Data bus	3.3V
HDMI_TX_D19	PIN_AE12	Video Data bus	3.3V
HDMI_TX_D20	PIN_AH9	Video Data bus	3.3V
HDMI_TX_D21	PIN_AJ4	Video Data bus	3.3V
HDMI_TX_D22	PIN_AJ6	Video Data bus	3.3V
HDMI_TX_D23	PIN_AH12	Video Data bus	3.3V
HDMI_TX_CLK	PIN_A22	Video Clock	3.3V
HDMI_TX_DE	PIN_Y1	Data Enable Signal for Digital Video.	3.3V
HDMI_TX_HS	PIN_V1	Horizontal Synchronization	3.3V
HDMI_TX_VS	PIN_R2	Vertical Synchronization	3.3V
HDMI_TX_INT	PIN_AL12	Interrupt Signal	3.3V
HDMI_I2S0	PIN_M1	I2S Channel 0 Audio Data Input	3.3V
HDMI_MCLK	PIN_T6	Audio Reference Clock Input	3.3V
HDMI_LRCLK	PIN_L1	Audio Left/Right Channel Signal Input	3.3V
HDMI_SCLK	PIN_J2	I2S Audio Clock Input	3.3V
FPGA_I2C_SCL	PIN_M2	FPGA I2C Clock	3.3V
FPGA_I2C_SDA	PIN_N2	FPGA I2C Data	3.3V

3.9 SDRAM Memory

The board features 64MB of SDRAM with a single 64MB (16M×32) SDRAM chip. The chip connects to the FPGA with 32 data lines, 13 address lines and control lines. This chip uses the 1.8V LVCMOS signaling standard. Connections between the FPGA and SDRAM are shown in [Figure 3-19](#), and the pin assignment is listed in [Table 3-13](#).



Figure 3-19 Connections between the FPGA and SDRAM

Table 3-13 Pin Assignment of SDRAM

Signal Name	FPGA Pin No.	Description	I/O Standard
DRAM_ADDR[0]	PIN_BN26	SDRAM Address[0]	1.8-V LVCMOS

DRAM_ADDR[1]	PIN_BH27	SDRAM Address[1]	1.8-V LVCMOS
DRAM_ADDR[2]	PIN_BM26	SDRAM Address[2]	1.8-V LVCMOS
DRAM_ADDR[3]	PIN_BH18	SDRAM Address[3]	1.8-V LVCMOS
DRAM_ADDR[4]	PIN_BH26	SDRAM Address[4]	1.8-V LVCMOS
DRAM_ADDR[5]	PIN_BK18	SDRAM Address[5]	1.8-V LVCMOS
DRAM_ADDR[6]	PIN_BH21	SDRAM Address[6]	1.8-V LVCMOS
DRAM_ADDR[7]	PIN_BH32	SDRAM Address[7]	1.8-V LVCMOS
DRAM_ADDR[8]	PIN_BK26	SDRAM Address[8]	1.8-V LVCMOS
DRAM_ADDR[9]	PIN_BH35	SDRAM Address[9]	1.8-V LVCMOS
DRAM_ADDR[10]	PIN_BN27	SDRAM Address[10]	1.8-V LVCMOS
DRAM_ADDR[11]	PIN_BN29	SDRAM Address[11]	1.8-V LVCMOS
DRAM_ADDR[12]	PIN_BM24	SDRAM Address[12]	1.8-V LVCMOS
DRAM_DQ[0]	PIN_BM51	SDRAM Data[0]	1.8-V LVCMOS
DRAM_DQ[1]	PIN_BM50	SDRAM Data[1]	1.8-V LVCMOS
DRAM_DQ[2]	PIN_BK50	SDRAM Data[2]	1.8-V LVCMOS
DRAM_DQ[3]	PIN_BN47	SDRAM Data[3]	1.8-V LVCMOS
DRAM_DQ[4]	PIN_BM47	SDRAM Data[4]	1.8-V LVCMOS
DRAM_DQ[5]	PIN_BH50	SDRAM Data[5]	1.8-V LVCMOS
DRAM_DQ[6]	PIN_BL51	SDRAM Data[6]	1.8-V LVCMOS
DRAM_DQ[7]	PIN_BH46	SDRAM Data[7]	1.8-V LVCMOS
DRAM_DQ[8]	PIN_BN37	SDRAM Data[8]	1.8-V LVCMOS
DRAM_DQ[9]	PIN_BM37	SDRAM Data[9]	1.8-V LVCMOS
DRAM_DQ[10]	PIN_BN39	SDRAM Data[10]	1.8-V LVCMOS
DRAM_DQ[11]	PIN_BM44	SDRAM Data[11]	1.8-V LVCMOS
DRAM_DQ[12]	PIN_BM42	SDRAM Data[12]	1.8-V LVCMOS
DRAM_DQ[13]	PIN_BM45	SDRAM Data[13]	1.8-V LVCMOS
DRAM_DQ[14]	PIN_BN42	SDRAM Data[14]	1.8-V LVCMOS
DRAM_DQ[15]	PIN_BN45	SDRAM Data[15]	1.8-V LVCMOS
DRAM_DQ[16]	PIN_BM9	SDRAM Data[16]	1.8-V LVCMOS
DRAM_DQ[17]	PIN_BC1	SDRAM Data[17]	1.8-V LVCMOS
DRAM_DQ[18]	PIN_BN5	SDRAM Data[18]	1.8-V LVCMOS
DRAM_DQ[19]	PIN_BJ1	SDRAM Data[19]	1.8-V LVCMOS
DRAM_DQ[20]	PIN_BJ2	SDRAM Data[20]	1.8-V LVCMOS
DRAM_DQ[21]	PIN_BG2	SDRAM Data[21]	1.8-V LVCMOS
DRAM_DQ[22]	PIN_BC2	SDRAM Data[22]	1.8-V LVCMOS
DRAM_DQ[23]	PIN_BG1	SDRAM Data[23]	1.8-V LVCMOS
DRAM_DQ[24]	PIN_BN8	SDRAM Data[24]	1.8-V LVCMOS
DRAM_DQ[25]	PIN_BN11	SDRAM Data[25]	1.8-V LVCMOS
DRAM_DQ[26]	PIN_BM8	SDRAM Data[26]	1.8-V LVCMOS
DRAM_DQ[27]	PIN_BM14	SDRAM Data[27]	1.8-V LVCMOS
DRAM_DQ[28]	PIN_BM11	SDRAM Data[28]	1.8-V LVCMOS
DRAM_DQ[29]	PIN_BN16	SDRAM Data[29]	1.8-V LVCMOS
DRAM_DQ[30]	PIN_BN14	SDRAM Data[30]	1.8-V LVCMOS
DRAM_DQ[31]	PIN_BM19	SDRAM Data[31]	1.8-V LVCMOS
DRAM_BA[0]	PIN_BH43	SDRAM Bank Address[0]	1.8-V LVCMOS

DRAM_BA[1]	PIN_BM29	SDRAM Bank Address[1]	1.8-V LVCMOS
DRAM_DQM[0]	PIN_BE4	SDRAM byte Data Mask[0]	1.8-V LVCMOS
DRAM_DQM[1]	PIN_BA2	SDRAM byte Data Mask[1]	1.8-V LVCMOS
DRAM_DQM[2]	PIN_BE6	SDRAM byte Data Mask[2]	1.8-V LVCMOS
DRAM_DQM[3]	PIN_BD1	SDRAM byte Data Mask[3]	1.8-V LVCMOS
DRAM_RAS_N	PIN_BM31	SDRAM Row Address Strobe	1.8-V LVCMOS
DRAM_CAS_N	PIN_BK40	SDRAM Column Address Strobe	1.8-V LVCMOS
DRAM_CKE	PIN_BH40	SDRAM Clock Enable	1.8-V LVCMOS
DRAM_CLK	PIN_BK32	SDRAM Clock	1.8-V LVCMOS
DRAM_WE_N	PIN_BM34	SDRAM Write Enable	1.8-V LVCMOS
DRAM_CS_N	PIN_BN34	SDRAM Chip Select	1.8-V LVCMOS

3.10 UART

The board provides a UART interface for users to communicate and transfer data with the Agilex 3 FPGA through the host. The interface is implemented via the FT2232H chip in the USB Blaster III circuit.

Connecting a USB cable between the DE23-Lite board's Type-C connector and the host enables both USB Blaster III and FPGA UART functions. Serial driver isn't required, users need to make sure the USB Blaster III driver is installed before using the UART function. After connecting the USB cable, normally USB Blaster III and a COM port number will both show in PC Device Manager.

Figure 3-20 shows the connections between the Agilex 3 FPGA, FT2232H chip and the USB Type-C connector. **Table 3-14** lists the pin assignments of the UART interface connected to the FPGA.

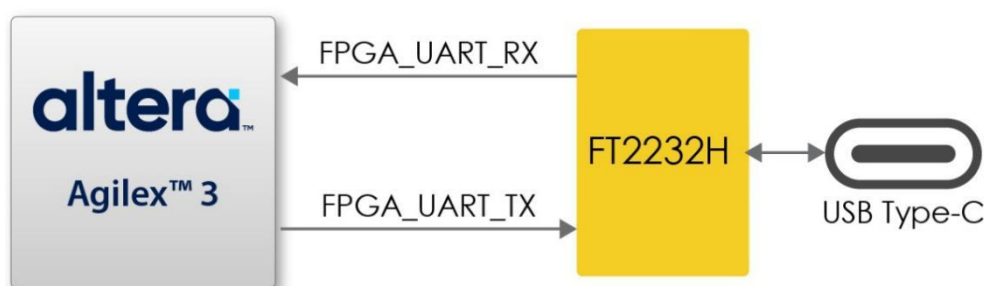


Figure 3-20 Connections between the FPGA and Type-C UART connector

Table 3-14 Pin Assignment of UART

Signal Name	FPGA Pin No.	Description	I/O Standard
FPGA_UART_TX	PIN_J1	UART Transmitter	3.3-V LVCMOS
FPGA_UART_RX	PIN_BH10	UART Receiver	3.3-V LVCMOS

3.11 A/D Converter and 2x5 Header

The DE23-Lite features the TI TLA2528, an 8-channel, multiplexed, 12-bit successive approximation register (SAR) analog-to-digital converter capable of sampling rates up to 140 KSPS. The analog input range for all input channels extends from 0 V to 5.5V . The I2C interface allows SCL(Serial Clock Line) to operate at frequencies up to 3.4 MHz (high-speed mode). The ADC can be configured to accept eight single-ended analog input signals at ADC_IN0 through ADC_IN7. Alternatively, these pins can be reconfigured as digital inputs or outputs in GPIO mode. These eight signals are routed to a 2×5 header, as shown in **Figure 3-21**.

More information about the A/D converter chip is available in its datasheet, which can be found on the manufacturer's website or in the directory \datasheet\ADC of the DE23-Lite resource package.

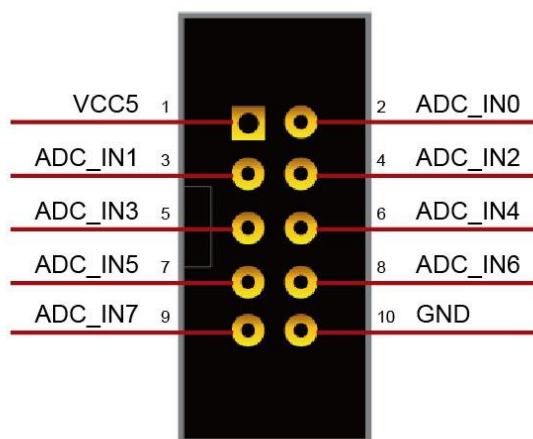


Figure 3-21 Signals of the 2x5 Header

Figure 3-22 shows the connections between the FPGA, 2×5 header, and the A/D converter. **Table 3-15** shows the pin assignment of the A/D converter.

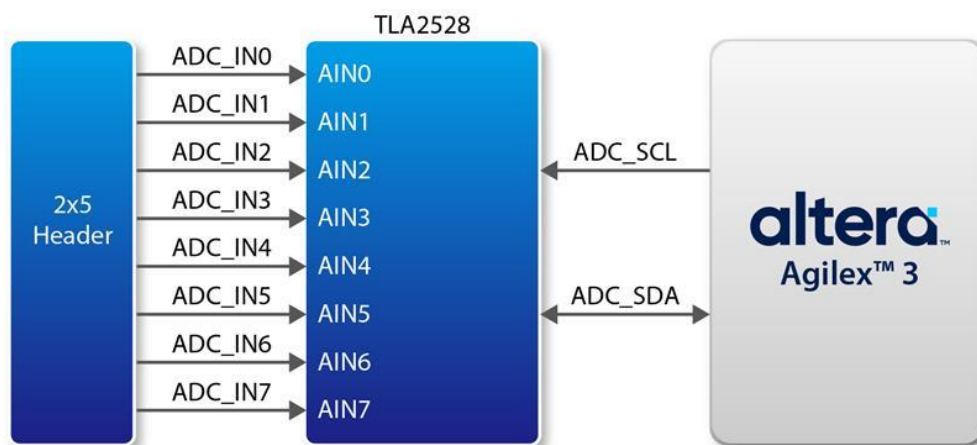


Figure 3-22 Connections between the FPGA, 2x5 header and the A/D converter

Table 3-15 Pin Assignment of ADC

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
ADC_SCL	PIN_M2	I2C clock output	3.3V
ADC_SDA	PIN_N2	I2C data inout	3.3V

Chapter 4

Examples of Designs

This chapter provides examples of advanced designs implemented by RTL or Qsys on the **DE23-Lite board**. These reference designs cover the features of peripherals connected to the FPGA, such as SDRAM, HDMI, ADC and UART. All the associated files can be found in the directory **\Demonstrations** of DE23-Lite Resource Package.

■ Installation of Demonstrations

To install the demonstrations on your computer:

Copy the folder Demonstrations to a local directory of your choice. It is important to make sure the path to your local directory contains NO space. Otherwise, it will lead to error in Nios.

Note: Quartus Pro v25.1 is required for all DE23-Lite demonstrations, including the Agilex 3 FPGA device.

4.1 DE23-Lite Factory Configuration

The DE23-Lite board has a default configuration bitstream pre-programmed, which demonstrates some of the basic features on board. The setup required for this demonstration and the location of its files are shown below **Figure 4-1**.

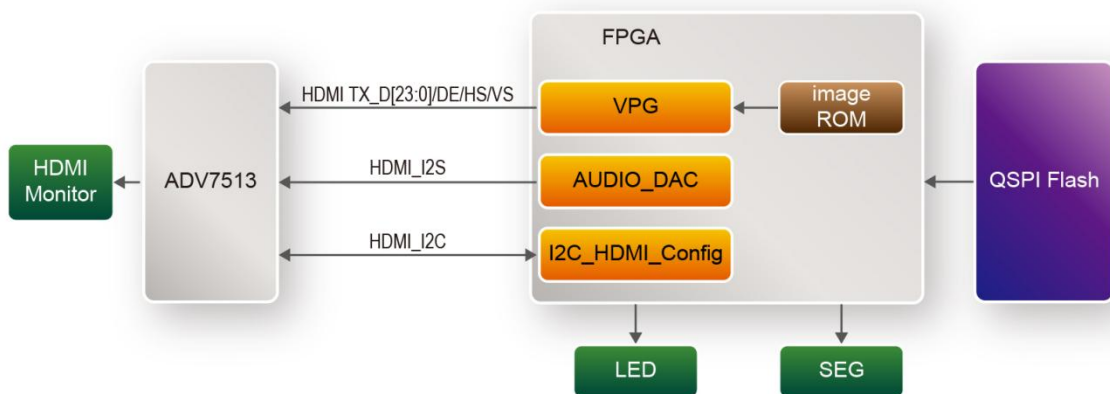


Figure 4-1 Block Diagram of the default demo

■ Design Tools

- Quartus Prime 25.1 Pro Edition

■ Demonstration Source Code

- Quartus Project directory: \Demonstrations\HDMI_ASx4
- Bitstream used: golden_top.sof or golden_top.jic

■ Demonstration Batch File

Demo Batch File Folder: \Demonstrations\HDMI_ASx4\demo_batch

The demo batch file includes following files:

- Demo Batch File : test.bat
- Convert jic Batch File: convert.bat
- Flash Batch File: flash_program.bat
- FPGA Configure File: golden_top.sof, golden_top.jic

■ Demonstration Setup and Instructions

- Set MSEL to 00, use a Type-C USB cable to connected the board(J3) to PC, and plug the 5V DC adapter to J5. Use HDMI cable to connect the board to HDMI monitor. If

necessary (that is, if the default factory configuration is not currently stored in the QSPI device), download the bit stream to the board via JTAG interface.

- You should now be able to observe the ten user LEDRs are blinking, the HEX0~HEX5 display 0~F in a loop, and the HDMI monitor displays the board image.
- The test.bat is executed to program .sof to the FPGA, the flash_program.bat is executed to program the .jic to the flash, the convert.bat is executed to convert .sof to .jic file then program the .jic to the flash.
- If users want to program a new design into the QSPI flash device, the easiest method is to copy the new .sof file to the demo_batch folder and execute the convert.bat to program the flash device.

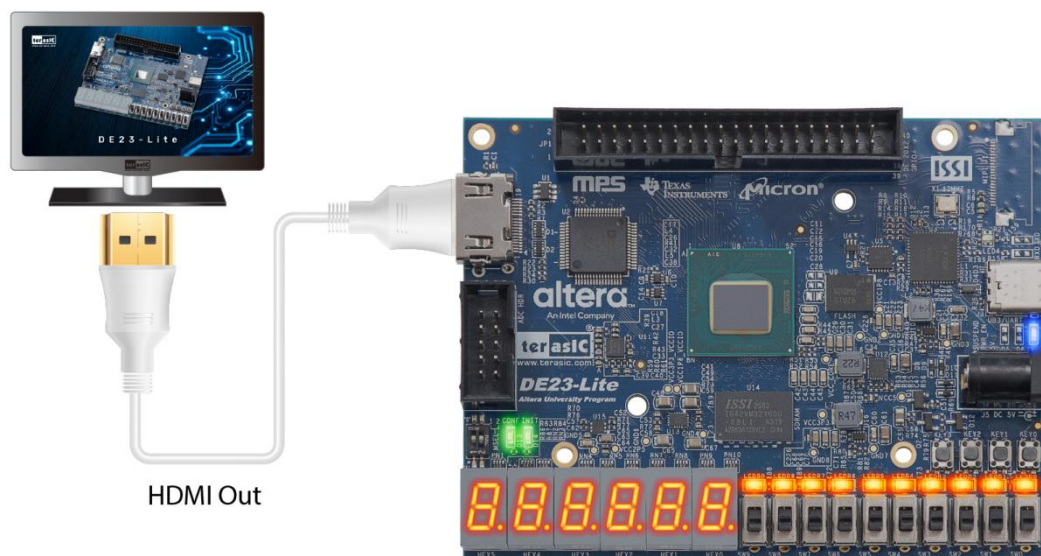


Figure 4-2 Setup for the default demo

4.2 SDRAM Test in Verilog

DE23-Lite resource package offers another SDRAM test with its test code written in Verilog HDL. The memory size of the SDRAM is 16Mx32bit.

■ System Block Diagram

Figure 4-3 shows the function block diagram of this demonstration. The SDRAM controller uses 50 MHz as a reference clock and generates 125MHz as the memory clock.

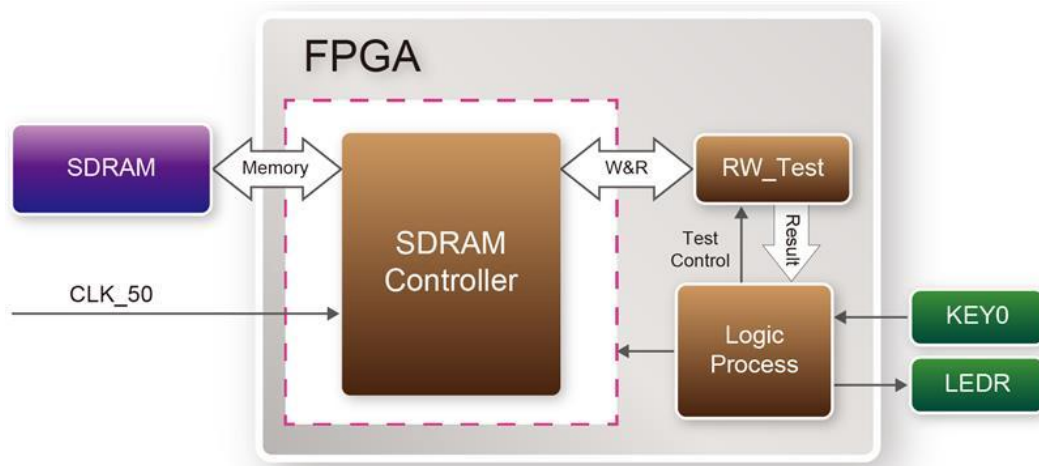


Figure 4- 3 Block Diagram of the SDRAM test in Verilog

RW_test module writes the entire memory with a test sequence first before comparing the data read back with the regenerated test sequence, which is same as the data written to the memory. KEY0 triggers test control signals for the SDRAM, and the LEDRs will indicate the test result according to [Table 4-1](#).

■ Design Tools

- Quartus Prime 25.1 Pro Edition

■ Demonstration Source Code

- Quartus Project directory: \Demonstrations\SDRAM_RTL
- Bitstream used: golden_top.sof

■ Demonstration Batch File

Demo Batch File Folder: \Demonstrations\SDRAM_RTL\demo_batch

The demo batch file includes following files:

- Demo Batch File : test.bat
- FPGA Configure File: golden_top.sof

■ Demonstration Setup and Instructions

- Please make sure both Quartus II and UBIII driver are installed on the host PC.
- use a Type-C USB cable to connected the board(J3) to PC,and plug the 5V DC adapter to J5.
- Execute the demo batch file “ test.bat” from the directory SDRAM_RTL_Test\demo_batch.
- Press KEY0 on the board to start the verification process. When KEY0 is pressed, the LEDR [2:0] should turn on.Then release KEY0, LEDR1 and LEDR0 should start blinking.
- After approximately 6 seconds, LEDR0 should stop blinking and stay ON to indicate the

test is PASS. Table 4-1 lists the status of LEDR indicators.

- If LED0 is still blinking after approximately 8 seconds, the SDRAM test is NG.
- If LEDR1 is not blinking, it means 50MHz clock source is not working.
- Press KEY0 again to repeat the SDRAM test.

Table 4-1 Status of LED Indicators

Name	Description
LEDR2	Indicate Reset operation
LEDR1	Blinks when the 50 MHz clock source is functioning normally
LEDR0	ON if the test is PASS after releasing KEY0

4.3 SDRAM_Test_NiosV

Many applications use SDRAM as a temporary storage solution. This demonstration provides hardware and software designs to illustrate how to perform SDRAM memory access in Platform Designer (formerly Qsys). We describe how the SDRAM Controller is used to access the single SDRAM chip (U14) on the FPGA board, and how the Nios V processor is used to read and write the SDRAM for hardware verification. The SDRAM controller handles the complex aspects of using the SDRAM, such as initializing the memory devices, managing SDRAM banks, and Keeping the devices refreshed at appropriate intervals.

■ System Block Diagram

Figure

4-4

shows the system block diagram of this demonstration. In Platform Designer, a PLL is used. The PLL uses 50MHz from the on-board clock generator as a reference clock and generates a 80MHz clock for the SDRAM Controller and the SDRAM chip.

One SDRAM Controller is used in the demonstration, configured to control a total memory space of 64MB (16Mx32bit x 1 chip). The Nios V processor is used to perform the memory test. The Nios V software program runs in On-Chip Memory. A PIO Controller is used to monitor button status, which triggers the start of the memory test.

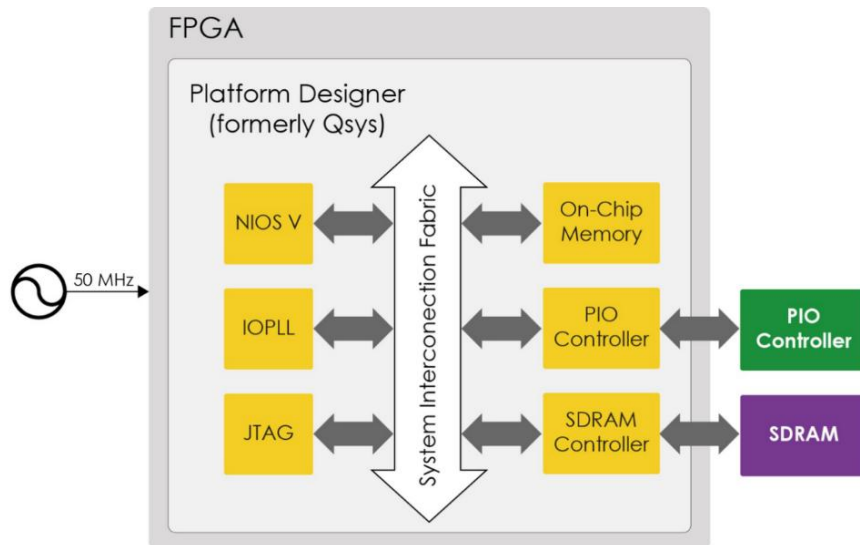


Figure 4-4 Block diagram of the SDRAM Basic Demonstration

The system flow is controlled by a Nios V program. First, the Nios V program writes test patterns into the entire 64MB of SDRAM. Then, it calls the Nios V system function,

`alt_dcache_flush_all()`, to ensure all data has been written from the cache to the SDRAM. Finally, it reads data from the SDRAM for verification. Since the full test may take a long time, a quick test option is also provided. The Nios V program writes a fixed pattern to the address and data lines and reads it back for verification. The program displays progress in the Nios V terminal during the write/read process. When the verification process is complete, the result is displayed in the terminal.

■ Design Tools

Quartus Prime 25.1 Pro Edition
Ashling* RiscFree* IDE for Altera

■ Demonstration Source Code

Quartus Project directory: SDRAM_Test_NiosV
Nios V Project workspace: SDRAM_Test_NiosV/software

■ Demonstration File Locations

Demo batch directory: SDRAM_Test_NiosV\demo_batch

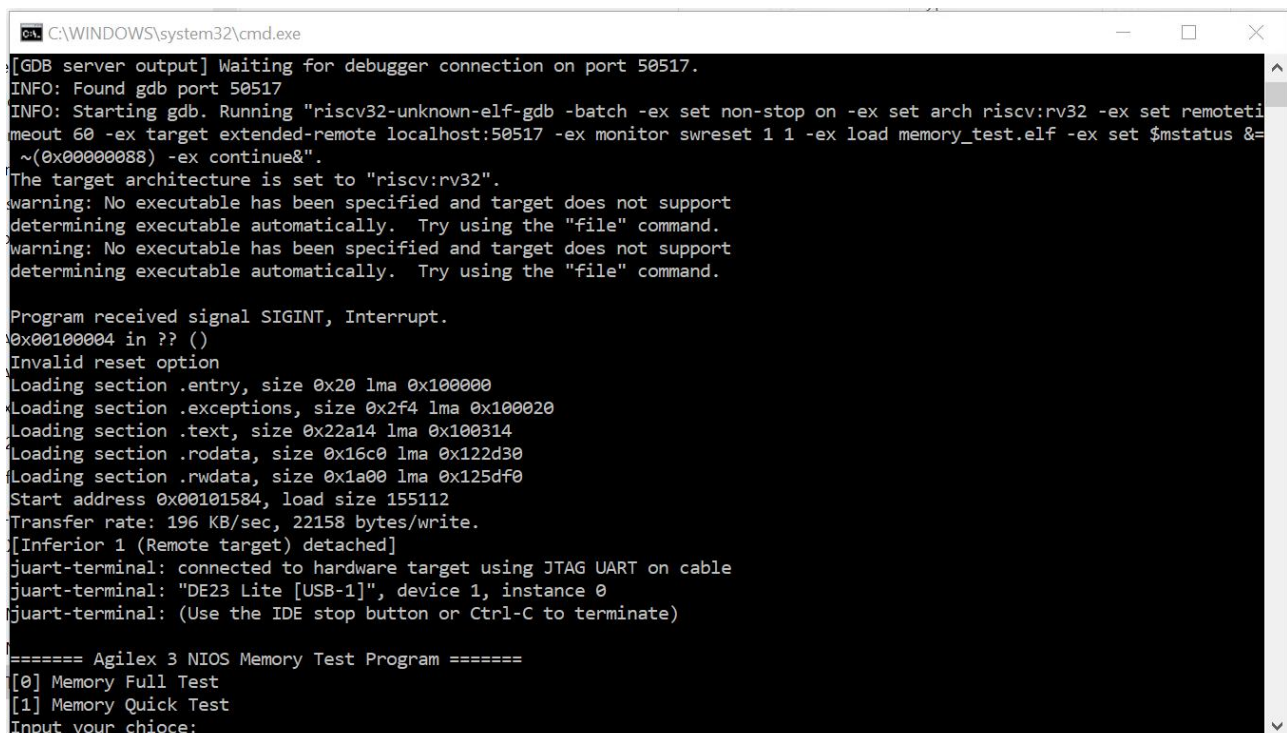
■ The folder includes the following files:

- Batch file: test.bat
- FPGA configuration file: golden_top.sof
- Nios V bin file: memory_test.elf

■ Demonstration Setup

Please follow below procedures to set up the demonstrations.

1. Make sure Quartus Prime and Ashling RiscFree IDE are installed on your PC.
2. Power on the FPGA board.
3. Use a USB cable to connect the PC and the FPGA board, and install the USB Blaster III driver if necessary.
4. Execute the demo batch file test.bat located in the SDRAM_Test_NiosV\demo_batch folder.
5. After the Nios V program is downloaded and executed successfully, a prompt message will be displayed in the Nios V Command Shell and show the test result as shown in **Figure 4-5**.
6. Full Test: For the full SDRAM test, input key '0' and press 'Enter' in the Nios V Command Shell terminal. The program will display progress and result information. Press KEY0~KEY1 on the FPGA board to start the SDRAM verification process, and press KEY0 to continue the test.
7. Quick Test: For the SDRAM quick test, input key '1' and press 'Enter' in the Nios V Command Shell terminal. The program will display progress and result information. Press KEY0~KEY1 on the FPGA board to start the SDRAM verification process, and press KEY0 to continue the test.



```
C:\WINDOWS\system32\cmd.exe
[GDB server output] Waiting for debugger connection on port 50517.
INFO: Found gdb port 50517
INFO: Starting gdb. Running "riscv32-unknown-elf-gdb -batch -ex set non-stop on -ex set arch riscv:rv32 -ex set remoteti
meout 60 -ex target extended-remote localhost:50517 -ex monitor swreset 1 1 -ex load memory_test.elf -ex set $mstatus &=
~(0x00000088) -ex continue&".
The target architecture is set to "riscv:rv32".
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
Program received signal SIGINT, Interrupt.
0x00100004 in ?? ()
Invalid reset option
Loading section .entry, size 0x20 lma 0x100000
Loading section .exceptions, size 0x2f4 lma 0x100020
Loading section .text, size 0x22a14 lma 0x100314
Loading section .rodata, size 0x16c0 lma 0x122d30
Loading section .rdata, size 0x1a00 lma 0x125df0
Start address 0x00101584, load size 155112
Transfer rate: 196 KB/sec, 22158 bytes/write.
[Inferior 1 (Remote target) detached]
Juart-terminal: connected to hardware target using JTAG UART on cable
Juart-terminal: "DE23 Lite [USB-1]", device 1, instance 0
Juart-terminal: (Use the IDE stop button or Ctrl-C to terminate)

===== Agilix 3 NIOS Memory Test Program =====
[0] Memory Full Test
[1] Memory Quick Test
Input your chioice:
```

Figure 4-5 SDRAM Basic Demonstration

4.4 HDMI_out RTL in Verilog

This demonstration outputs a color bar to an HDMI monitor using the DE23-Lite board. **Figure 4-6** shows the block diagram of the design.

When the demonstration bitstream is loaded into FPGA. The I2C_HDMI_Config block will configure the HDMI transmitter chip(ADV7513) via I2C bus (I2C slave address=0x72). When KEY3 is pressed, the AUDIO_DAC block generates a 1 kHz sine-wave tone, which is sent via I2S to the ADV7513. The VPG block outputs the color-bar video signal to the ADV7513.

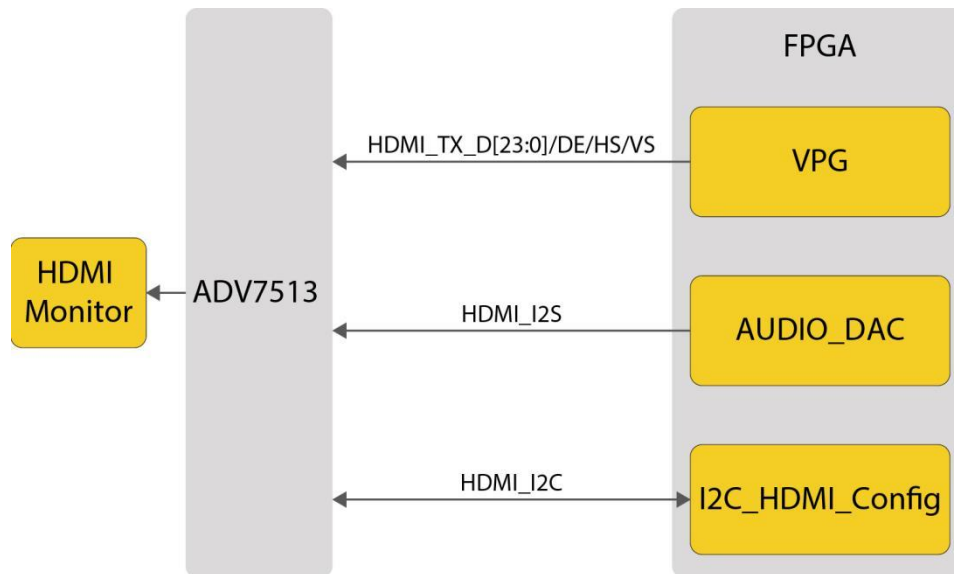


Figure 4-6 Block Diagram of the HDMI_out_RTL in Verilog

■ Design Tools

- Quartus Prime 25.1 Pro Edition

■ Demonstration Source Code

- Quartus Project directory: HDMI_out_RTL
- Bitstream used: golden_top.sof

■ Demonstration Batch File

Demo Batch File Folder: HDMI_out_RTL\demo_batch

The demo batch file includes following files:

- Demo Batch File : test.bat
- FPGA Configure File: golden_top.sof

■ Demonstration Setup and Instructions

- Please make sure both Quartus Pro and UBIII driver are installed on the host PC.
- Connect the HDMI TX(J2) of the DE23-Lite board to a HDMI interface monitor.

- Connect a USB cable to the Type-C USB connector (J3) on the DE23-Lite board and the host PC, and plug the 5V DC adapter to J5.
- Load the bitstream into the FPGA by executing the batch file 'test.bat' from the directory \HDMI_out_RTL\demo_batch\.
- Figure 4-5 Setup for the HDMI_out_RTL demonstration Setup for the HDMI_out_RTL demonstration

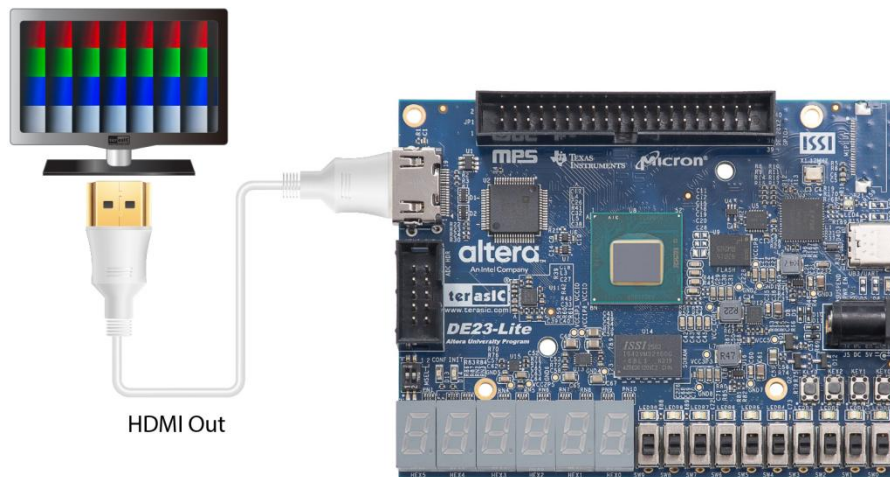


Figure 4-7 Setup for the HDMI_out_RTL demonstration

4.5 UART Test in Verilog

DE23-Lite resource package offers UART loopback test code written in Verilog HDL.

■ System Block Diagram

Figure 3-8 shows the function block diagram of this demonstration. As shown in the block diagram, FPGA_UART_RX is the UART signal output from the FT2232 on the DE23-Lite and input to the FPGA. Conversely, FPGA_UART_TX is the UART signal transmitted from the FPGA and input to the FT2232 on the DE23-Lite. KEY0 functions as a cutoff switch for the UART loopback.

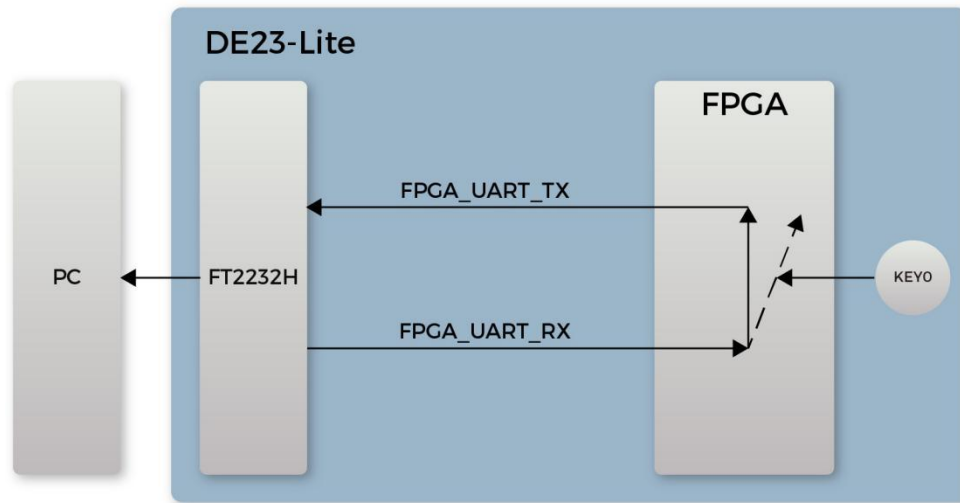


Figure 4-8 Block Diagram of the UART in Verilog

■ Design Tools

- Quartus Prime 25.1 Pro Edition

■ Demonstration Source Code

- Quartus Project directory: UART
- Bitstream used: golden_top.sof

■ Demonstration Batch File

Demo Batch File Folder: UART\demo_batch

The demo batch file includes following files:

- Demo Batch File : test.bat
- FPGA Configure File: golden_top.sof

■ Demonstration Setup and Instructions

- Please make sure both Quartus Pro and UBIII driver are installed on the host PC.
- use a Type-C USB cable to connected the board(J3) to PC,and plug the 5V DC adapter to J5.
- Execute the demo batch file “ test.bat” from the directory UART\demo_batch.
- Use a UART communication tool such as PuTTY on the PC. Select the correct COM port (the COM port number can be found in Windows Device Manager), set the baud rate to 115200, and open the UART terminal window. Typing characters from the PC keyboard should result in echoed characters appearing in the terminal. Press KEY0 to verify whether the UART transmission is interrupted or not.

4.6 ADC RTL Demonstration

This demonstration illustrates steps to evaluate the performance of the 8-channel 12-bit analog-to-digital converter (ADC) LTA2528. The DC 5.0V on the 2x5 header is used to drive the

analog signals by a additional Voltage Divider Board. The input voltage is scaled down to eight discrete levels, approximately from 4V to 0.5V, and each level is connected to ADC input channels ADC_IN0 through ADC_IN7, respectively. The 12-bit measurement is displayed on the Signal Tape Logic Analyzer , And use the switch to toggle and display the measured voltage of each channel on the DE23-Lite's 7-segment display, in units of mV.

Figure 4-9 shows the block diagram of this demonstration. Design an I2C IP core in RTL style to communicate with the LTA2528.

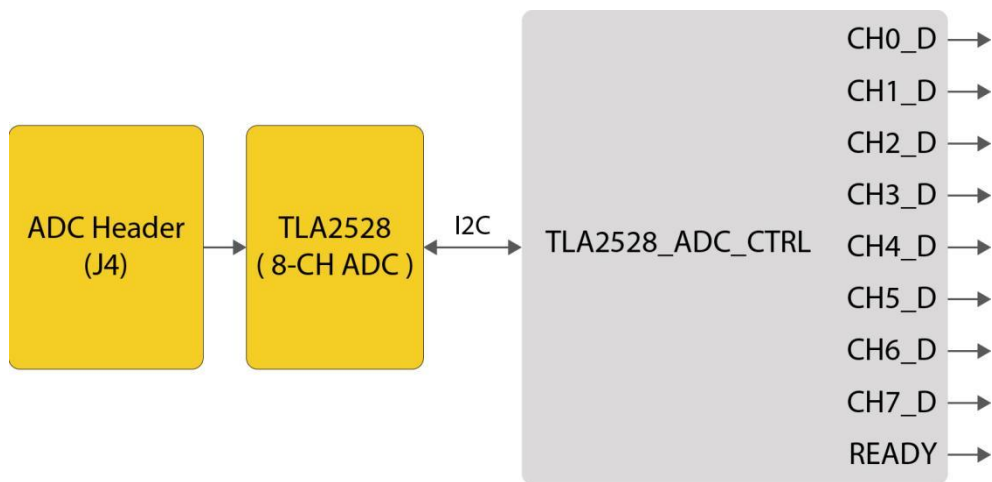


Figure 4-9 Block diagram of ADC reading

Figure 3-21 depicts the pin arrangement of the 2x5 header. This header is the input source of ADC convertor in this demonstration. Users apply 8 individual voltages (each < 5V) to ADC_IN0 to ADC_IN7 as inputs for ADC conversion. The FPGA will read the associated register in the convertor via serial interface and translates value to be displayed on the Signal Tape Logic Analyzer and display it on the 7-segment display.

Figure 4-10 shows the block diagram of ADC chip TLA2528. The TLA2528 is an easy-to-use, 8-channel, multiplexed, 12-bit, up to 140Msps, successive approximation register analog-to-digital converter (SAR ADC). The TLA2528 communicates via an FPGA_I2C(Slave address =0x20) .

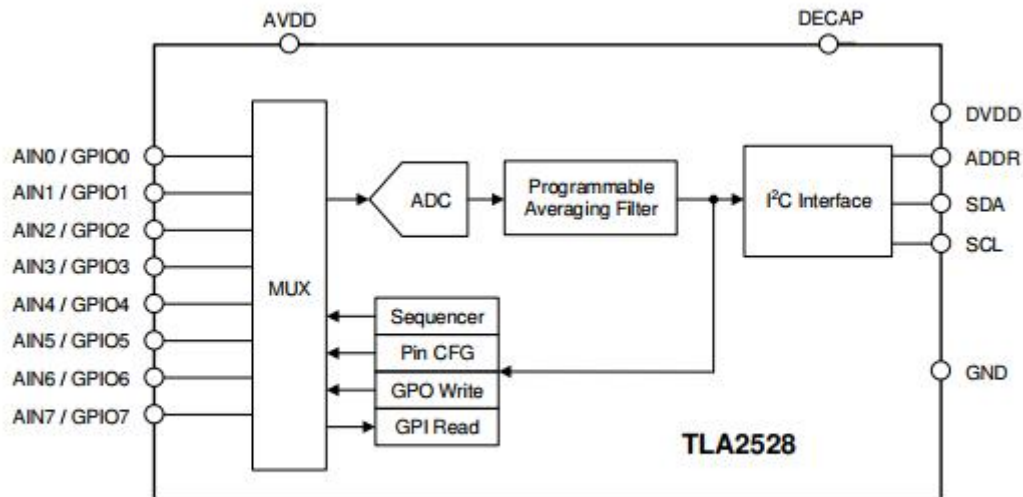


Figure 4- 10 TLA2528 Block Diagram

■ System Requirements

- DE23-Lite board x1
- Voltage Divider Board x1

■ Demonstration Source Code

- Bitstream used: golden_top.sof
- Quartus Project directory:ADC_RTL

■ Demonstration Batch File

Demo Batch File Folder: ADC_RTL\demo_batch

The demo batch file includes following files:

- Batch File for USB III: test.bat
- FPGA Configure File: golden_top.sof
- Signal Tape File:stp1.stp

■ Demonstration Setup

Please follow below procedures to set up the demonstration.

1. First, prepare an additional voltage divider board(as shown in **Figure 4-11**) . Then, plug it into the J4 header on the DE23-Lite.
2. Connect a USB cable to the Type-C USB connector (J3) on the DE23-Lite board and the host PC,and plug the 5V DC adapter to J5.
3. Execute the demo batch file test.bat to load the bitstream to the FPGA.
4. When SW[2:0] = 0, the voltage measured from ADC_IN0 (in mV) is shown on the 7-segment display. You can toggle SW[2:0] to select other channels.
5. You can also run **stp1.stp** to display the converted values of all ADC channels in the Quartus Signal Tap Analyzer., as shown in **Figure 4-12**.

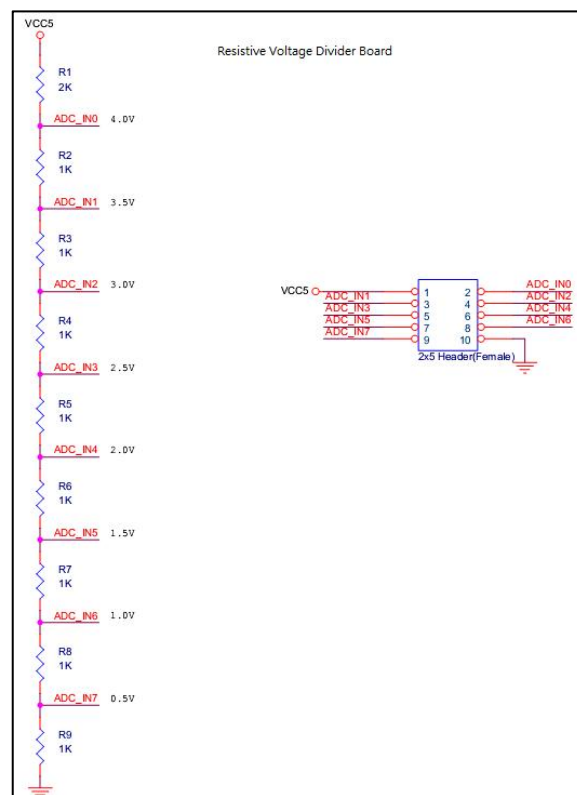


Figure 4- 11 The additional voltage divider board is set up for the ADC_RTL demonstration

⊕ u_TLA2518_ADC_CTRL CH0_D[15..0]	3985
⊕ u_TLA2518_ADC_CTRL CH1_D[15..0]	3487
⊕ u_TLA2518_ADC_CTRL CH2_D[15..0]	2987
⊕ u_TLA2518_ADC_CTRL CH3_D[15..0]	2491
⊕ u_TLA2518_ADC_CTRL CH4_D[15..0]	1989
⊕ u_TLA2518_ADC_CTRL CH5_D[15..0]	1492
⊕ u_TLA2518_ADC_CTRL CH6_D[15..0]	999
⊕ u_TLA2518_ADC_CTRL CH7_D[15..0]	497

Figure 4- 12 ADC Channel 0~7 reading (unit: mV)

4.7 ADC NiosV Reading

This demonstration illustrates steps to evaluate the performance of the 8-channel 12-bit analog-to-digital converter (ADC) LTA2528. The DC 5.0V on the 2x5 header is used to drive the analog signals by a trimmer potentiometer. The voltage should be adjusted within the range between 0 and 5.0V. The 12-bit voltage measurement is displayed on the NIOS terminal.

Figure 3-13 illustrates the block diagram of this demonstration. The NIOS V processor communicates with the LTA2528 sensor using the Avalon I²C (Host) Core IP. The NIOS V application runs entirely from on-chip memory. For developers working with NIOS V processors, Altera provides an I²C (Host) Core API that facilitates communication with I²C devices. The corresponding API define in the header file `altera_avalon_i2c.h`.

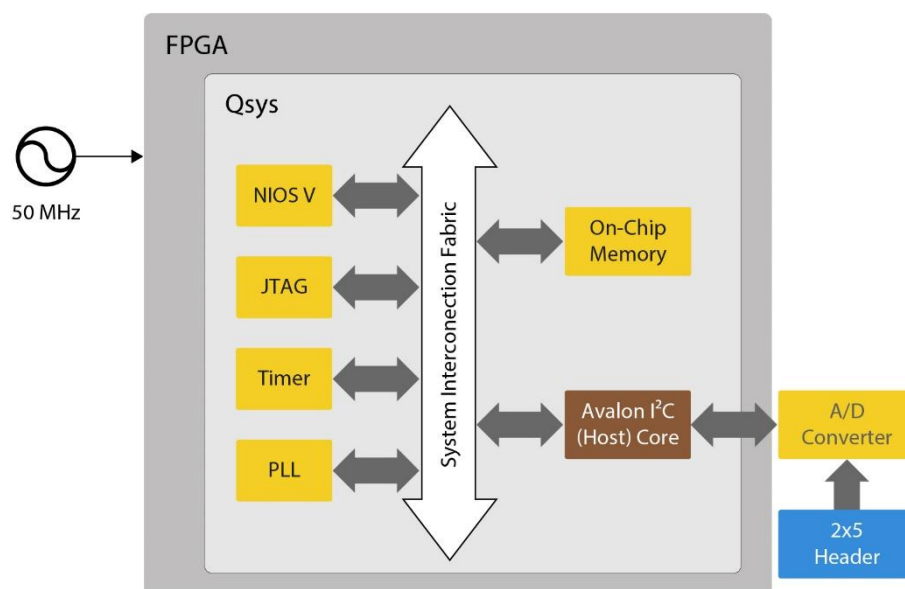


Figure 4- 13 Block diagram of ADC reading

Figure 3-21 depicts the pin arrangement of the 2x5 header. This header is the input source of ADC

converter in this demonstration. Users can connect a trimmer to the specified ADC channel (ADC_IN0 ~ ADC_IN7) that provides voltage to the ADC convert. The FPGA will read the associated register in the converter via serial interface and translates it to voltage value to be displayed on the NIOS Terminal.

Figure 4-10 shows the block diagram of ADC chip TLA2528. The TLA2528 is an easy-to-use, 140ksps, 8-channel, multiplexed, 12-bit, successive approximation register analog-to-digital converter (SAR ADC). The TLA2518 communicates via an I2C compatible interface and supports averaging multiple data samples with a single start of conversion.

■ System Requirements

- DE23-Lite board x1
- Trimmer Potentiometer x1
- Wire Strip x3

■ Demonstration File Locations

- Hardware project directory: ADC_NiosV
- Bitstream used: golden_top.sof
- Software project directory: ADC_NiosV\software
- Demo batch file : ADC_NiosV\demo_batch\test.bat

■ Install Ashling RiscFree IDE

Before executing this demo with NIOS V core, users need to install **Ashling RiscFree IDE** for Altera FPGAs to ensure that this batch file can be executed correctly. The file name should be *RiscFreeSetup-<Quartus Version>-windows.exe*. Users can find it under the [Quartus Pro download page](#) (Individual Files tab).

■ Demonstration Setup

Please follow below procedures to set up the demonstrations.

6. Connect the trimmer to one of eight for ADC channel on the 2x5 header, as shown in **Figure 4-14**, as well as the +5V and GND signals. The setup shown below is connected to ADC channel 0. *Note, the input voltage do no exceeds 5.0V.*
7. Connect a USB cable to the Type-C USB connector (J3) on the DE23-Lite board and the host

- PC, and plug the 5V DC adapter to J5.
8. Execute the demo batch file test.bat to load the bitstream and software execution file to the FPGA.
 9. Voltage measurements for all eight channels will be displayed on the NIOS Terminal, as shown in Figure 4- 15.

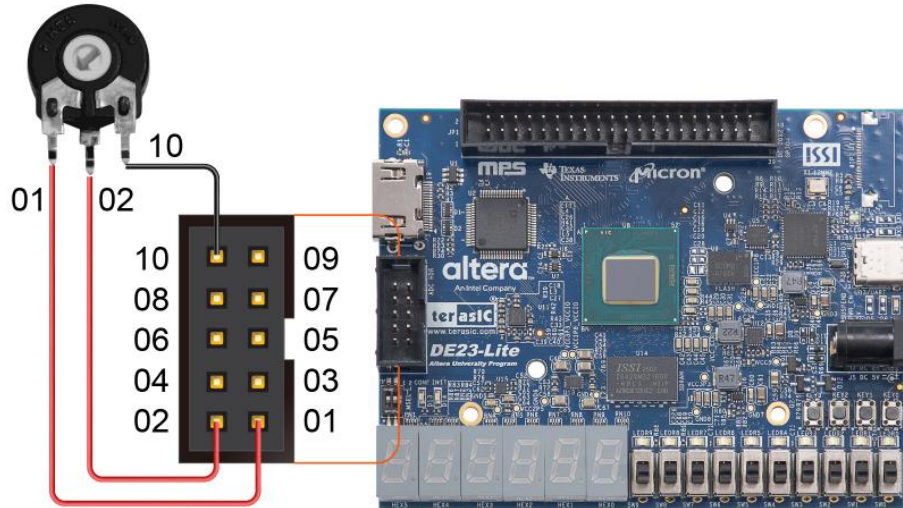


Figure 4- 14 Hardware setup for the ADC reading demonstration

```

C:\WINDOWS\system32\cmd.exe
Program received signal SIGINT, Interrupt.
0x00040004 in ?? ()
OK. Software reset asserted
Loading section .entry, size 0x20 lma 0x40000
Loading section .exceptions, size 0x2f4 lma 0x40020
Loading section .text, size 0x2ef04 lma 0x40314
Loading section .rodata, size 0x2da0 lma 0x6f220
Loading section .rwdata, size 0x1934 lma 0x738f4
Start address 0x000413f0, load size 211180
Transfer rate: 518 KB/sec, 26397 bytes/write.
[Inferior 1 (Remote target) detached]
===== 0 =====
ch0=3.62V (0b94h)
ch1=4.87V (0f95h)
ch2=4.97V (0fe5h)
ch3=4.98V (0ff0h)
ch4=3.64V (0ba5h)
ch5=4.88V (0fa0h)
ch6=2.59V (0849h)
ch7=2.31V (0762h)
===== 1 =====
ch0=3.62V (0b93h)
ch1=4.86V (0f8eh)
ch2=4.97V (0fe8h)
ch3=4.98V (0ff1h)
ch4=3.94V (0c9bh)
ch5=4.91V (0fb3h)
ch6=2.94V (0969h)
ch7=3.11V (09f2h)
===== 2 =====

```

Figure 4- 15 ADC Eight Channel Reading

Chapter 5

Programming the QSPI Flash

DE23-Lite board features an QSPI Flash memory connected to FPGA, allowing users to program their FPGA configuration files into the QSPI flash. This setup enables the FPGA's Secure Device Manager (SDM) to automatically load the configuration from QSPI Flash upon power-up. This chapter will describe the necessary tools, environment, and detailed steps involved in the programming process.

5.1 Programming Flow

Figure 5-1 below illustrates the components/software and programming flow for the entire programming process. The flow is as follows:

1. Users need to compile their projects in Quartus Prime to generate a stream file (.sof).
2. Next, use the **Quartus Prime Programming File Generator** to convert the .sof file into a .jic file.
3. Users can use the **Quartus Prime Programmer** on the host computer and the USB Blaster circuit on the DE23-Lite to transfer the .jic file from the host to the FPGA on the DE23-Lite.
4. The FPGA's internal Secure Device Manager (SDM) then programs the received .jic file into the QSPI Flash through the Active Serial x4 interface.

Note : If users program their own project into the QSPI flash, the factory code in the QSPI flash will be erased. If default factory functionality is required, please reprogram the factory flash file into your board.

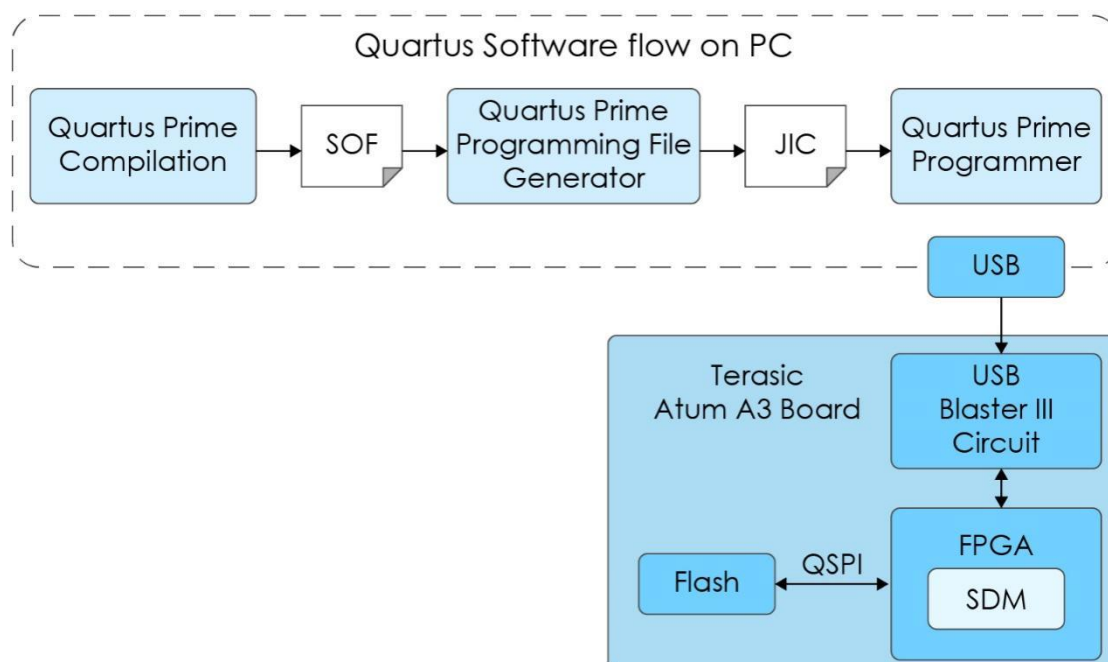


Figure 5-1 Programming Flow

5.2 Programming Steps with Batch Files

This section provides a guide on automating the conversion of your .sof (SRAM Object File) to .jic (JTAG Indirect Configuration File) format, and subsequently programming the .jic file into the QSPI flash memory, all through the use of batch files.

Terasic has pre-configured the necessary Quartus Prime commands within these batch files. This eliminates the need for users to manually launch the Quartus Prime software or input commands, significantly simplifying the programming process.

These convenient batch files are located in the following directory: Resource Package\Demonstrations\HDMI_ASx4\demo_batch

For a detailed overview of the files and their corresponding descriptions within this directory, please consult **Table 5-1**.

Table 5-1 File Descriptions in demo_batch folder

File Name	Description
convert.bat	A batch script that automates the conversion of a .sof file to a .jic file.
flash_program.bat	A batch script designed to streamline the programming of a .jic file into the QSPI flash memory.
golden_top.jic	The .jic file generated from the conversion of golden_top.sof.
golden_top.sof	The source .sof file intended for conversion.
test.bat	A batch script used to download the .sof file directly to the FPGA.

Here are the steps for programming:

1. Copy the files in the following path to your host PC:
Resource Package\Demonstrations\HDMI_ASx4\demo_batch
2. Rename the .sof file of your project to **golden_top.sof** and overwrite the existing file in *demo_batch*.
3. Execute **convert.bat** to convert **golden_top.sof** to **golden_top.jic**.
4. Verify that the MSEL Settings Switch on DE23-Lite is set to AS mode (MSEL[2:1] = 2'b00) as shown in **Figure 5-2**.
5. Ensure that the USB blaster connector (J3) of your DE23-Lite is connected to your host PC with a USB cable (see **Figure 5-3**) and the board's power is powered on.
6. Execute **flash_program.bat**, wait for the programming to complete (see **Figure 5-4**), then you can restart the board's power to check if your project runs correctly.

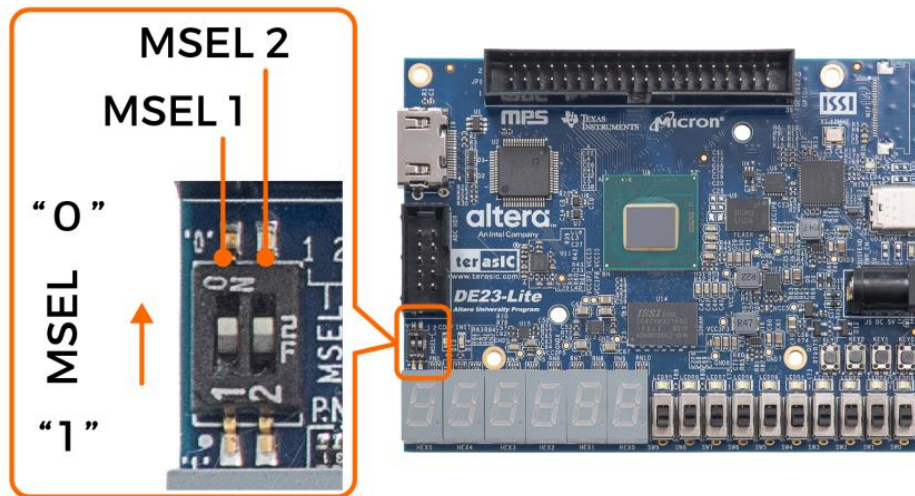


Figure 5-2 SW2 for FPGA Configuration Mode

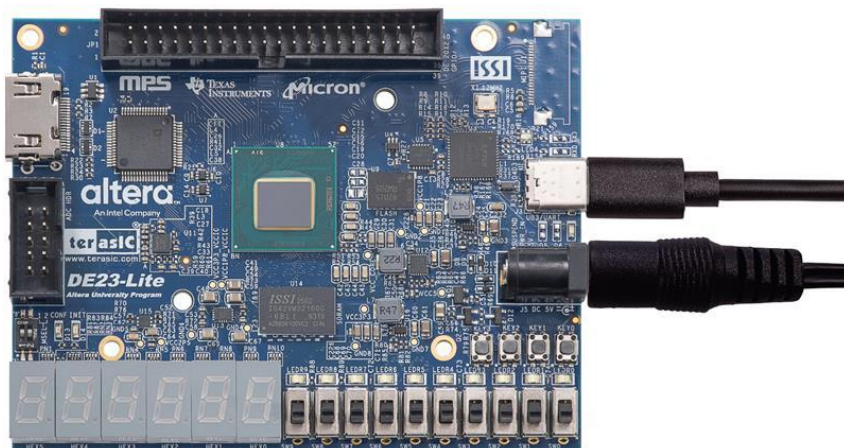


Figure 5-3 Power and USB cable connection for the board

```

C:\WINDOWS\system32\cmd.exe
Info: and other software and tools, and any partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Altera Program License
Info: Subscription Agreement, the Altera Quartus Prime License Agreement,
Info: the Altera IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Altera and sold by Altera or its authorized distributors. Please
Info: refer to the Altera Software License Subscription Agreements
Info: on the Quartus Prime software download page.
Info: Processing started: Fri Aug 15 15:56:29 2025
Info: System process ID: 18296
Info: Command: quartus_pgm -m jtag -c 1 -o pvi:golden_top.jic
Info (213045): Using programming cable "DE23 Lite [USB-1]"
Info (213011): Using programming file golden_top.jic with checksum 0xE04B4FC4 for device A3CZ135BB18A@1
Info (209060): Started Programmer operation at Fri Aug 15 15:56:40 2025
Info (18942): Configuring device index 1
Info (18943): Configuration succeeded at device index 1
Info (19094): Erasing flash chip select 0 at device index 1
Info (19096): Programming flash chip select 0 at device index 1
Info (19097): Verifying flash chip select 0 at device index 1
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Fri Aug 15 15:57:19 2025
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 1413 megabytes
Info: Processing ended: Fri Aug 15 15:57:19 2025
Info: Elapsed time: 00:00:50
Info: System process ID: 18296

```

Figure 5-4 Programmed successfully

5.3 Programming Steps with Quartus GUI

This section explains how to use the Quartus Prime Programmer (GUI) to convert the jic file and program it into the QSPI Flash.

1. Open the Quartus Prime Programmer and select **File → Programming File Generator...** (see Figure 5-5).

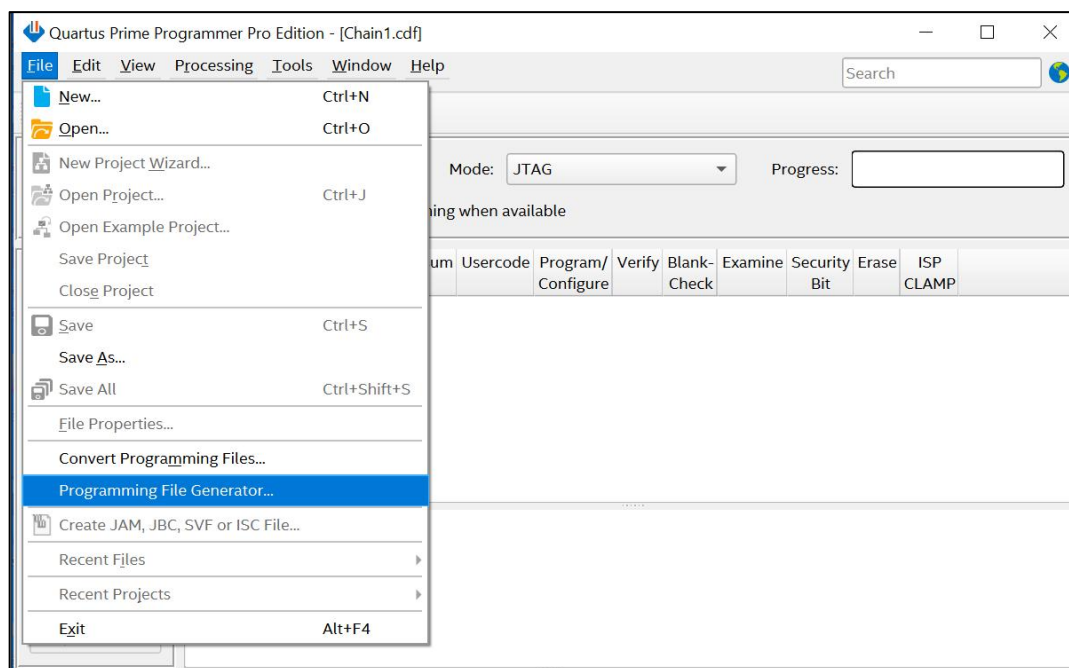


Figure 5-5 Open the Programming File Generator

When the Programming File Generator window appears (as shown in **Figure 5-6**):

- Set Device Family to Agilex 3.
- Set Configuration Mode to Active Serial x4.
- In the **Output Files** tab / **Output Directory**, choose the desired path to save the .jic file. Do not use the default Quartus path as it may cause errors during file generation.
- In the **Output Files** tab, check the JTAG Indirect Configuration File (.jic) option.

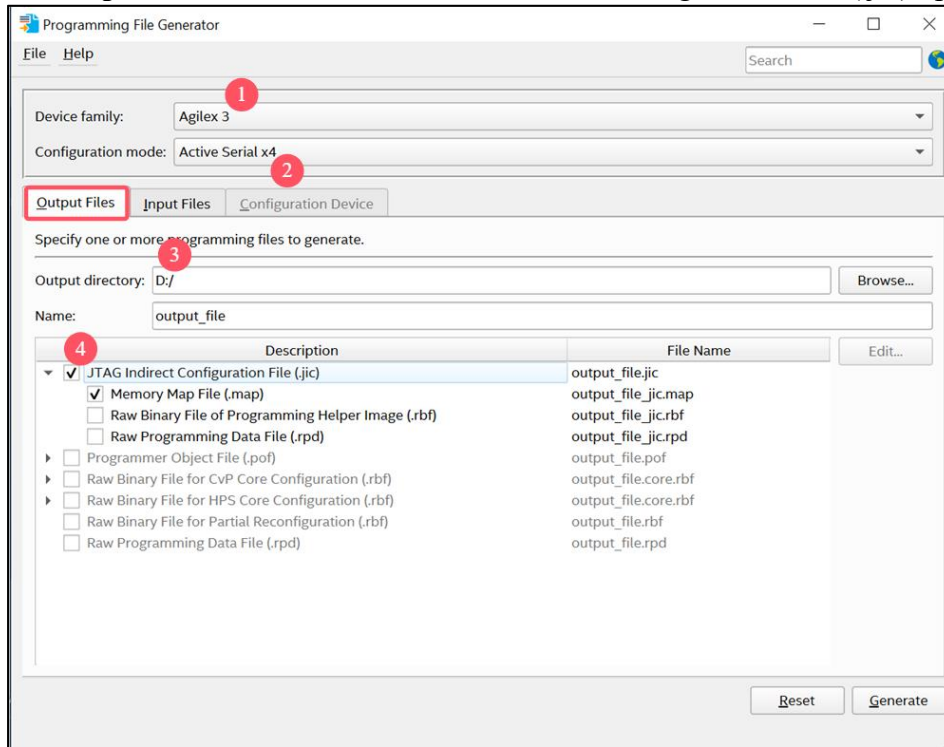


Figure 5-6 Setting output file

In the **Input Files** tab, click **Add Bitstream...** and select the .sof file you wish to convert. Once selected, it will appear in the window (see **Figure 5-7**).

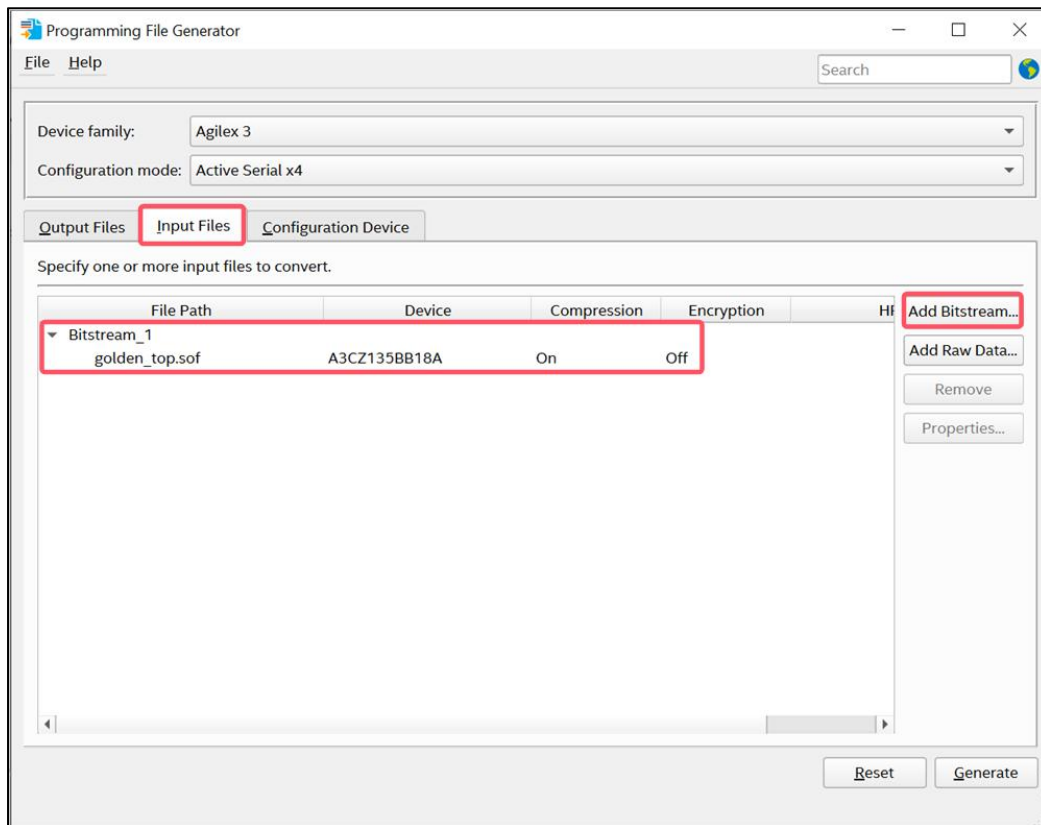


Figure 5-7 Setting input files

In the **Configuration Device** tab, click **Select....** In the appeared **Select Devices** window, check **Agilex 3** and **A3CZ135BB18A**, then click **OK** to complete the setup(see **Figure 5-8**).

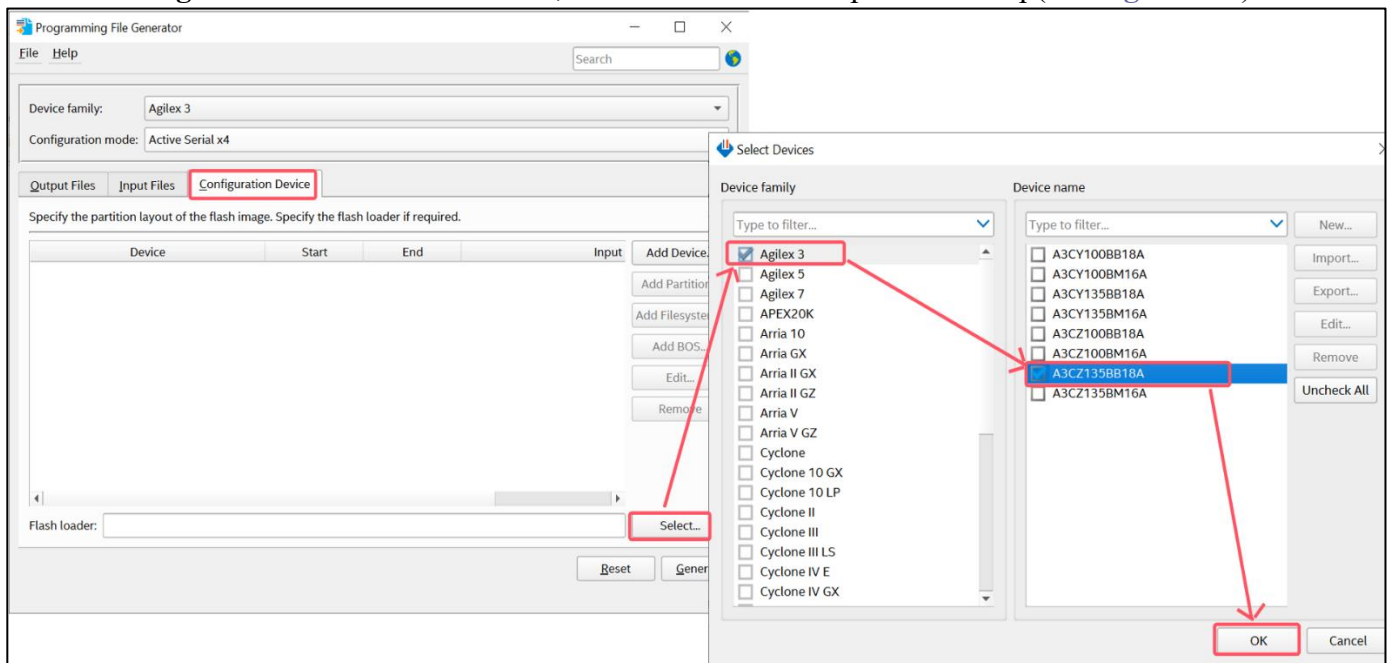


Figure 5-8 Setting Flash loader

In the **Configuration Device** tab, click **Add Device....** In the **Configuration Device** window, select **QSPI128** and then click **OK** to finish(see **Figure 5-9**).

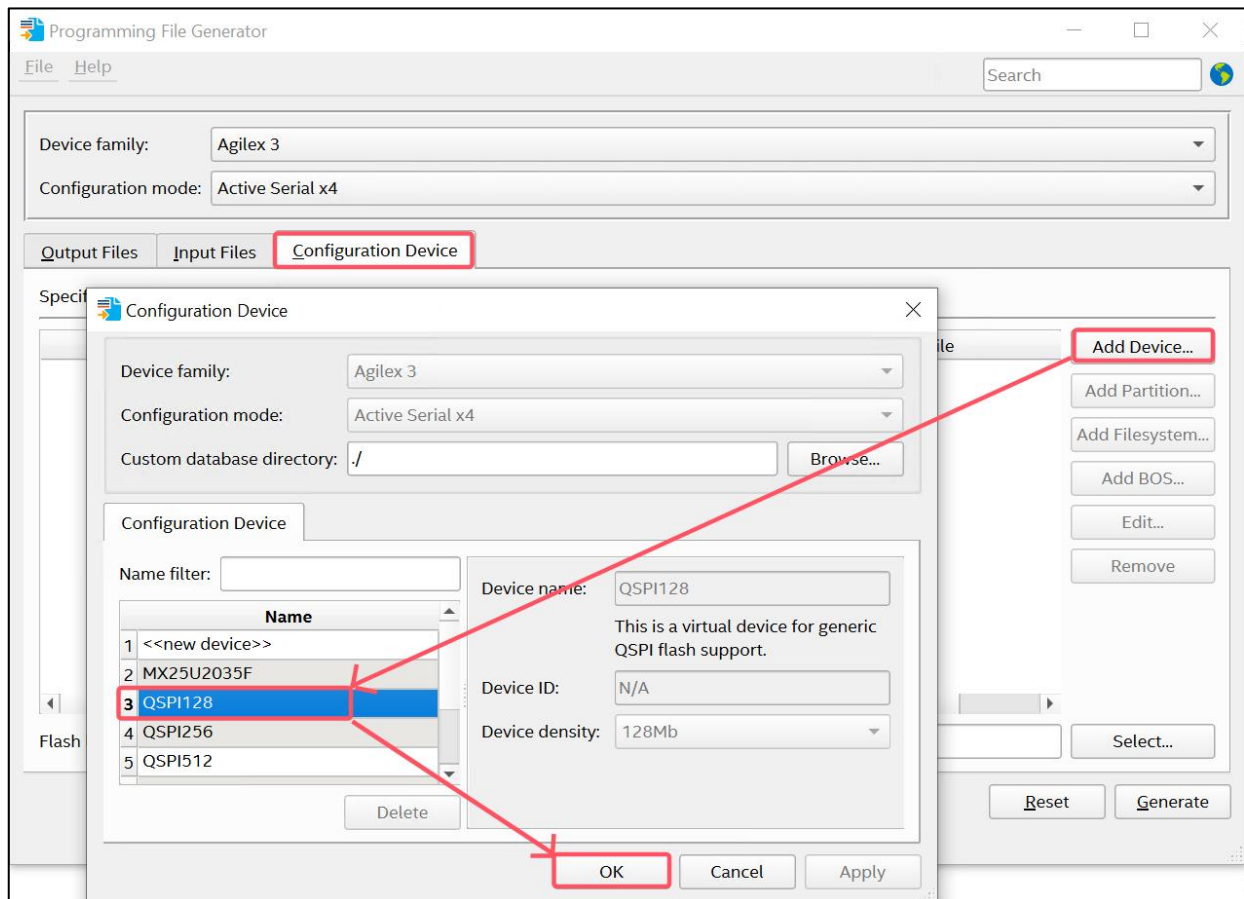


Figure 5-9 Setting Configuration Device

Next, in the **Configuration Device** tab, select the newly added QSPI128 and click **Add Partition....** In the Add Partition window, choose the previously added .sof file under **Input File** and click **OK** to complete the setup(see [Figure 5-10](#)).

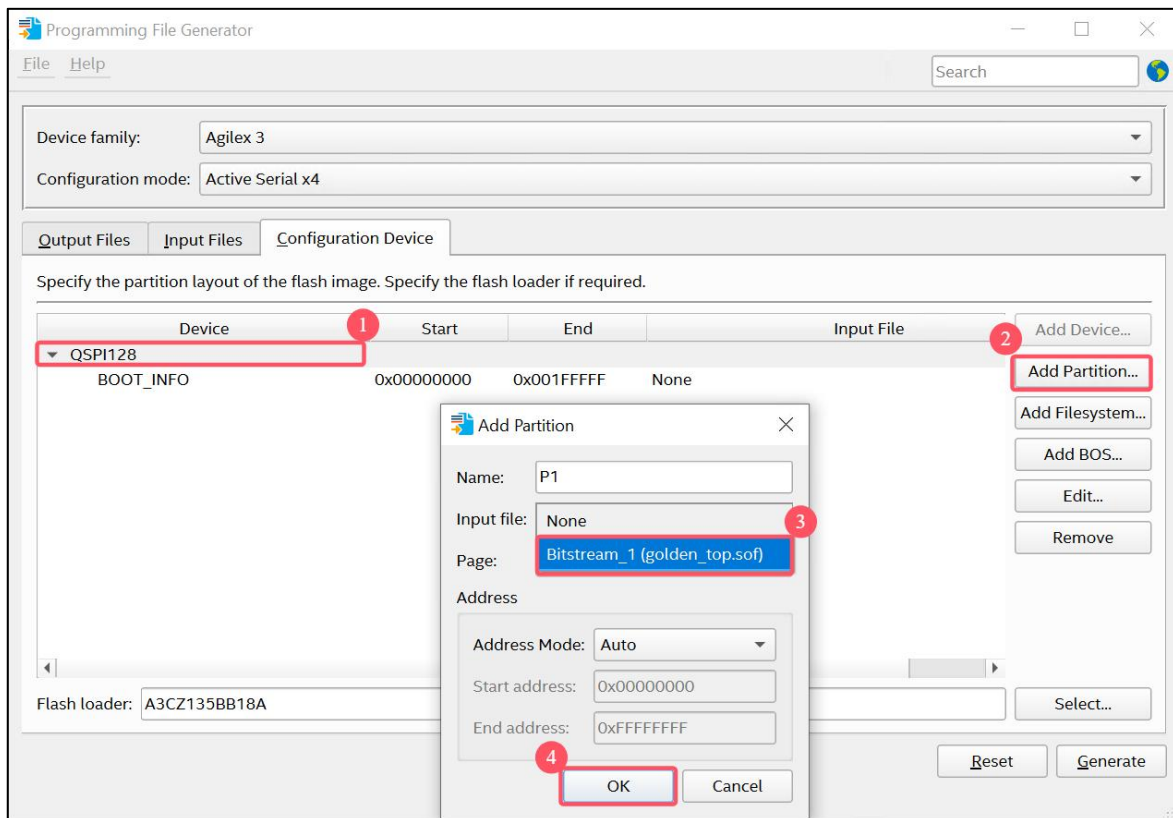


Figure 5-10 Setting Partition

Finally, click **Generate** to produce the .jic file(see [Figure 5-11](#)). When the window displays "Successfully generated output file(s)" (see [Figure 5-12](#)), it means the file conversion was successful.

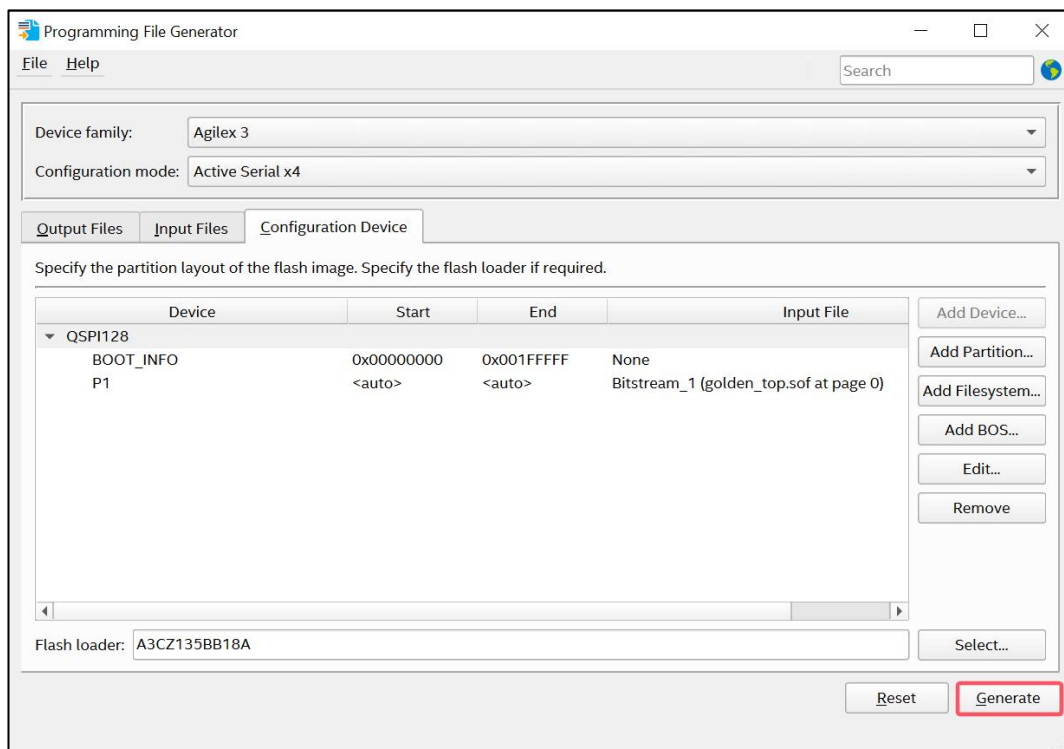


Figure 5- 11 Generate .jic file

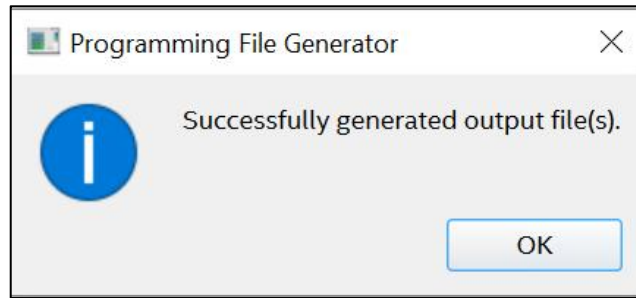


Figure 5- 12 Generated file successfully

2. Verify that the MSEL Settings Switch on DE23-Lite is set to AS mode (MSEL[2:1] = 2'b00) as shown in [Figure 5- 2](#).

Ensure that the USB blaster connector (J3) of your DE23-Lite is connected to your host PC with a USB cable (see [Figure 5- 3](#)) and that the board's power is turned on.

3. Return to the **Quartus Prime Programmer** window. First, ensure that the USB Blaster is detected. If it is not, click **Hardware Setup** to select it (see [Figure 5- 13](#)).

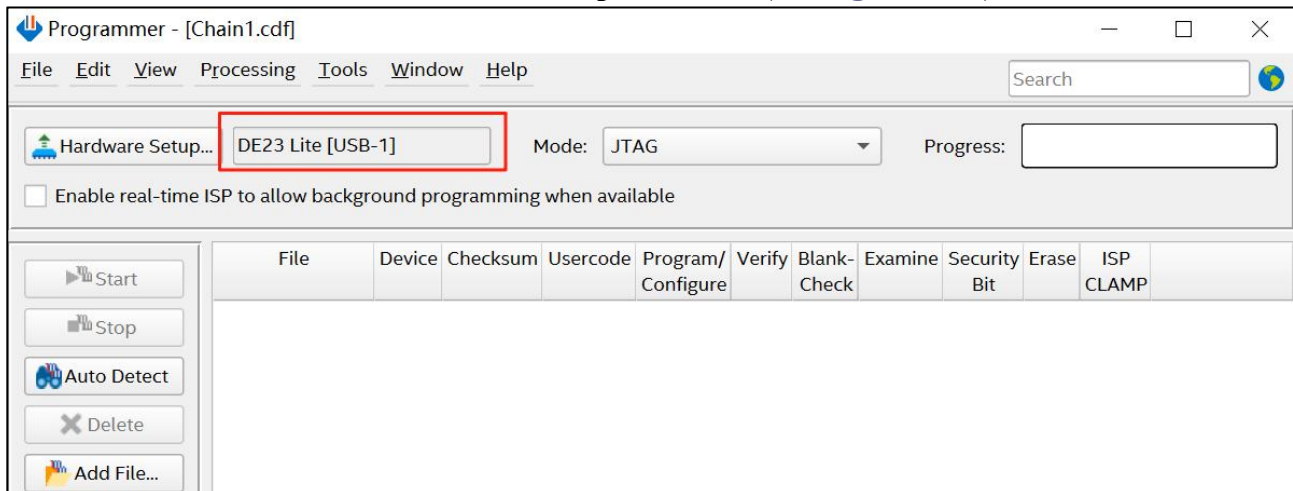


Figure 5- 13 Setting USB blaster

Click **Auto Detect**. This command lists the devices attached to the board's JTAG chain(see [Figure 5- 14](#)).

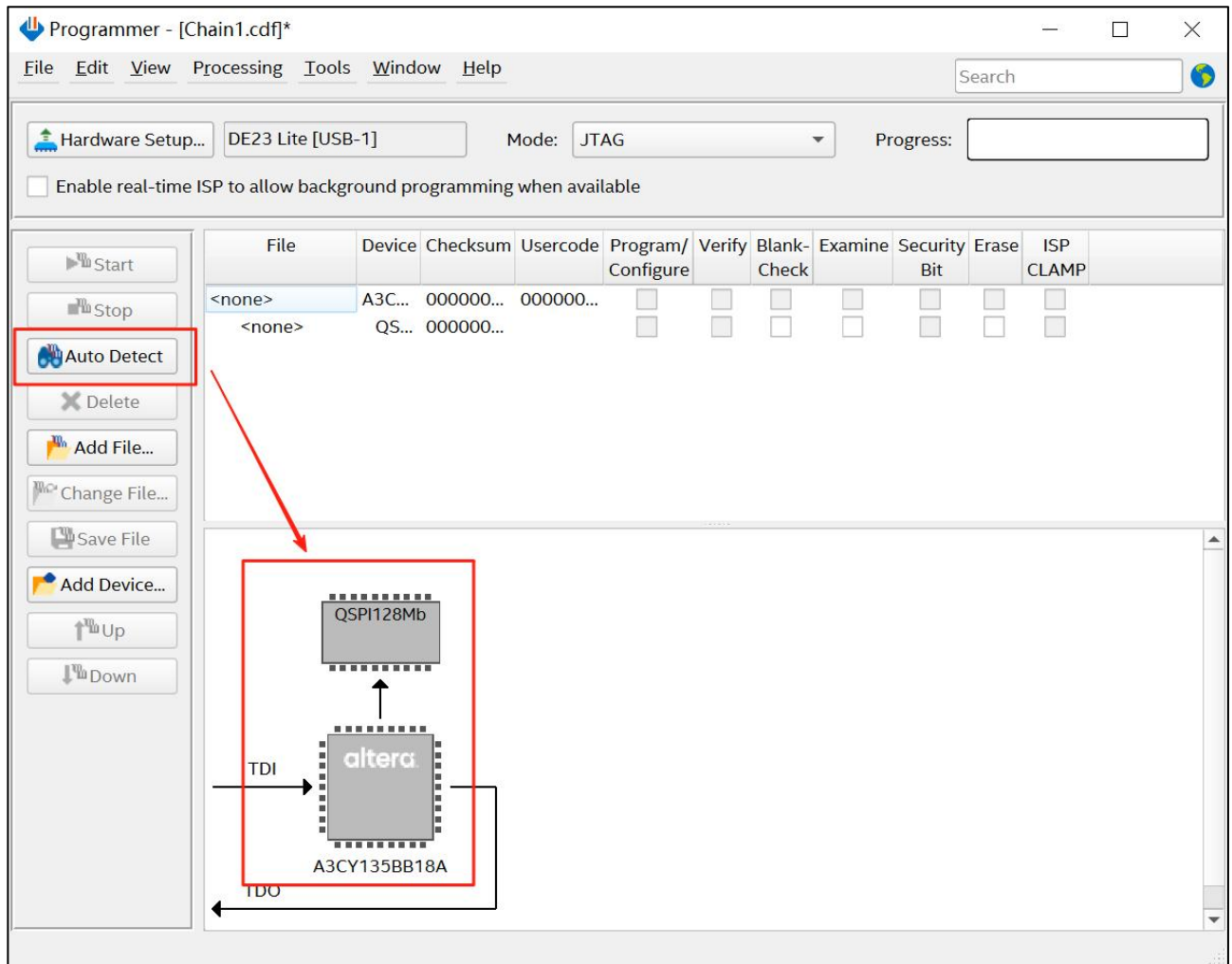


Figure 5-14 Detect FPGA device

Select the FPGA device that appears, then click Add File... to choose the .jic file to be programmed(see [Figure 5-15](#)).

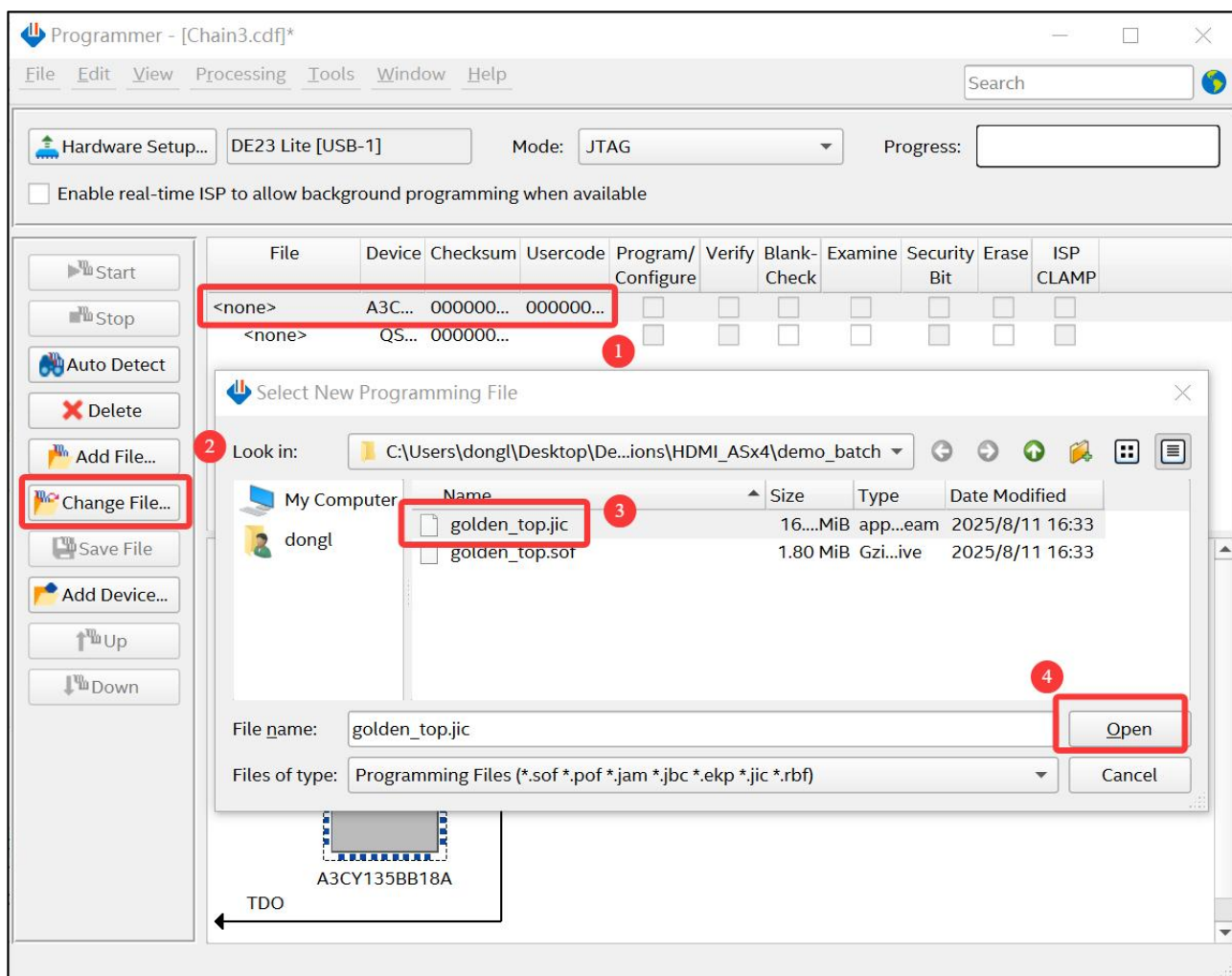


Figure 5-15 Add .jic file

Enable the **Program/Configure** option for the MT25QU128 device corresponding to the added file. Click **Start** to program the selected file to the flash(see [Figure 5-16](#)).

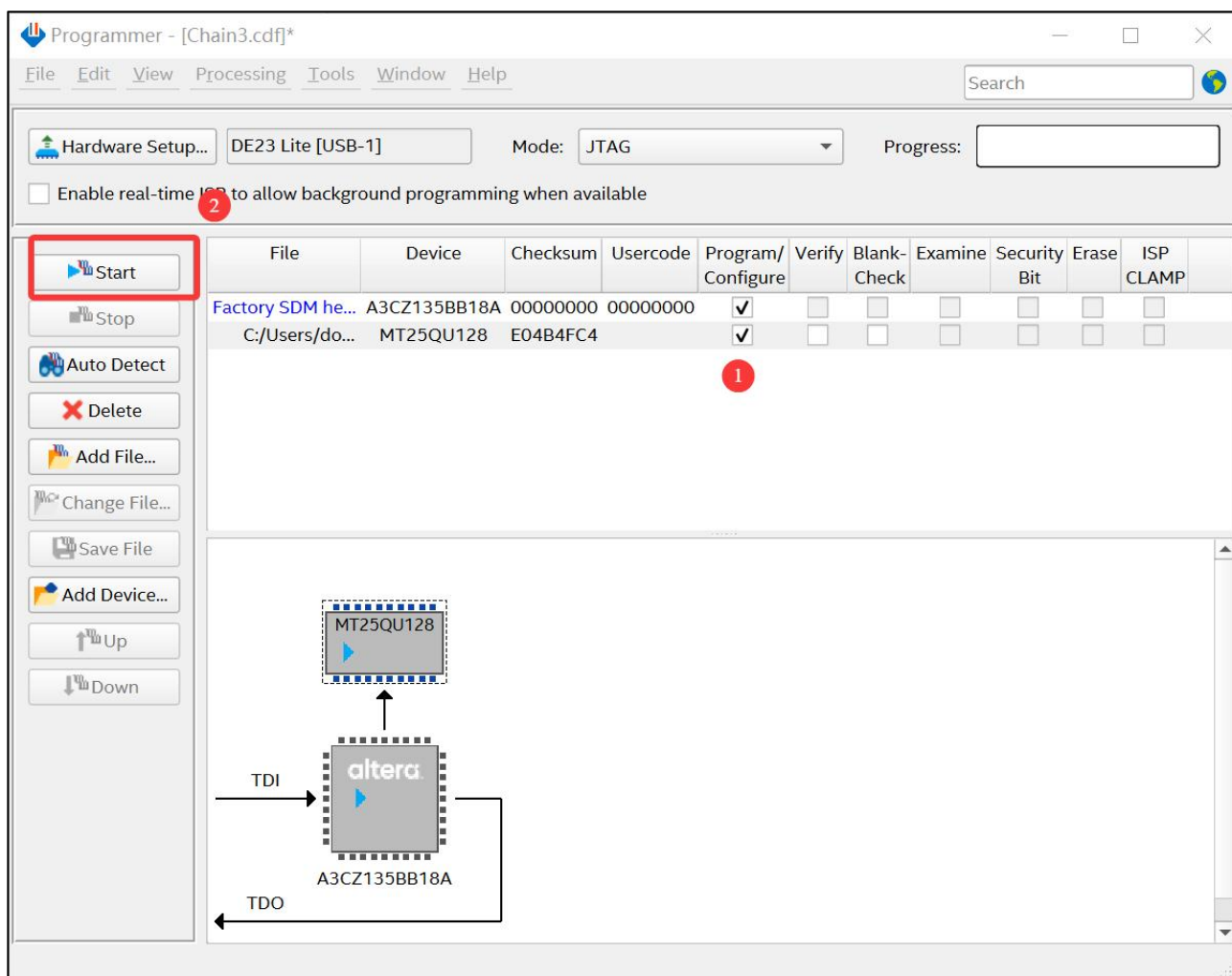


Figure 5-16 Enable programming file

Programming is complete when progress reaches 100%. Power cycle the board to verify your design(see [Figure 5-17](#)).

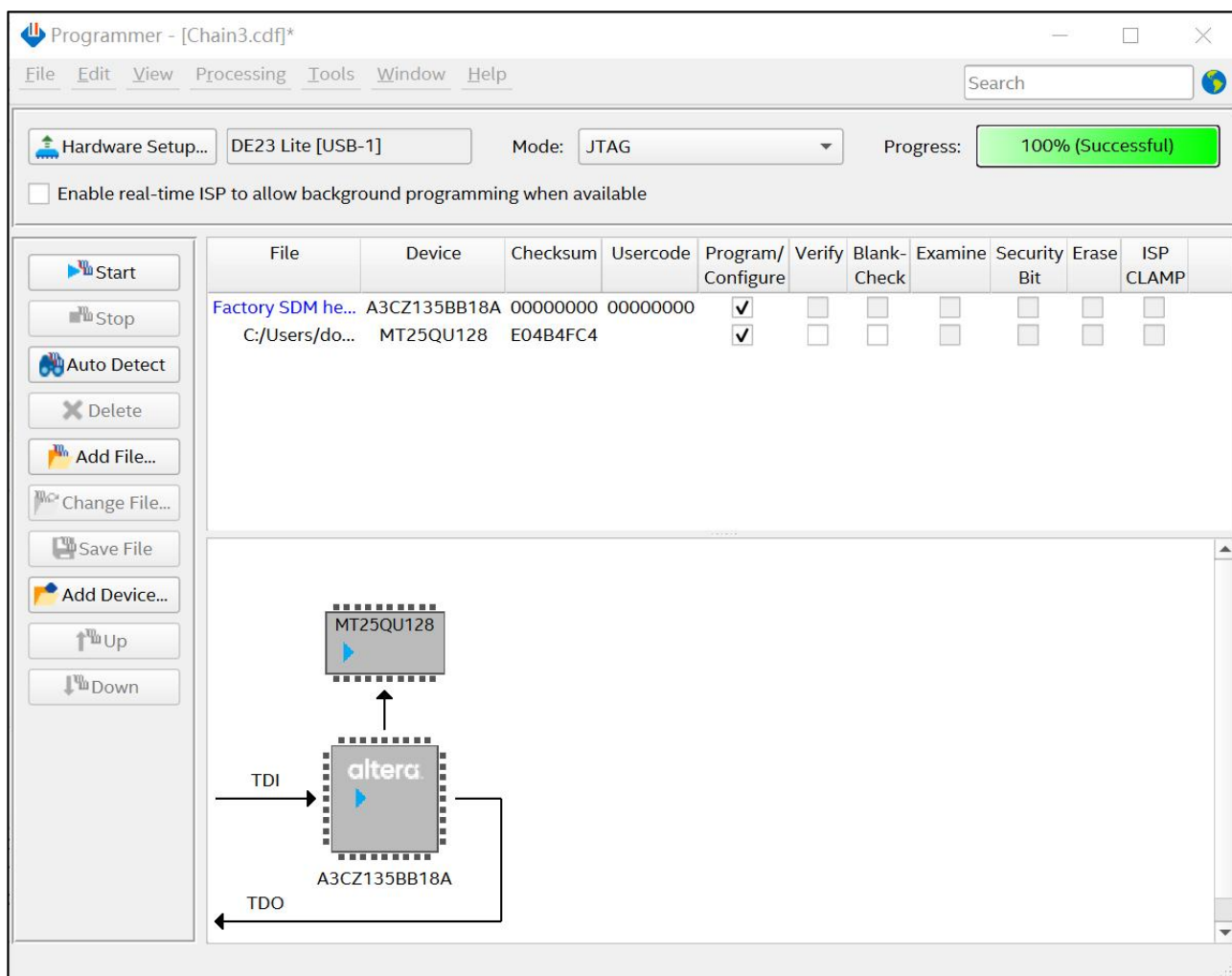


Figure 5-17 .jic file programmed successfully

Chapter 6

Appendix

6.1 *Revision History*

Version	Change Log
V1.0	Initial version
V1.1	Remove Powering the DE23-Lite Board – Safety Guidelines
V1.2	Add the SDRAM_Test_NiosV

6.2 *Copyright Statement*

Copyright © Terasic Inc. All Rights Reserved.