Advanced **L**ogic **S**ynthesis for **E**lectronics
http://www.ALSE-FR.com

Version 2018.03a

AVDB

Advanced Video Development Board

User's Manual

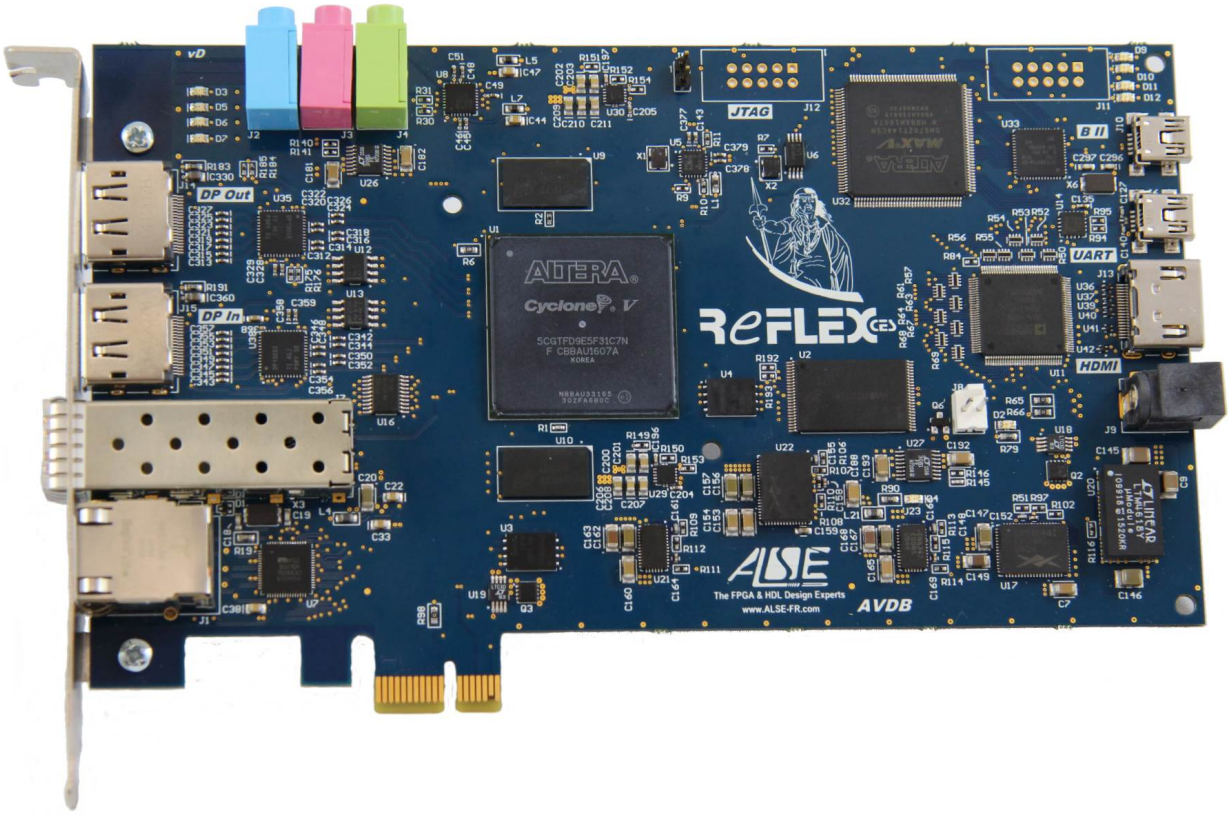


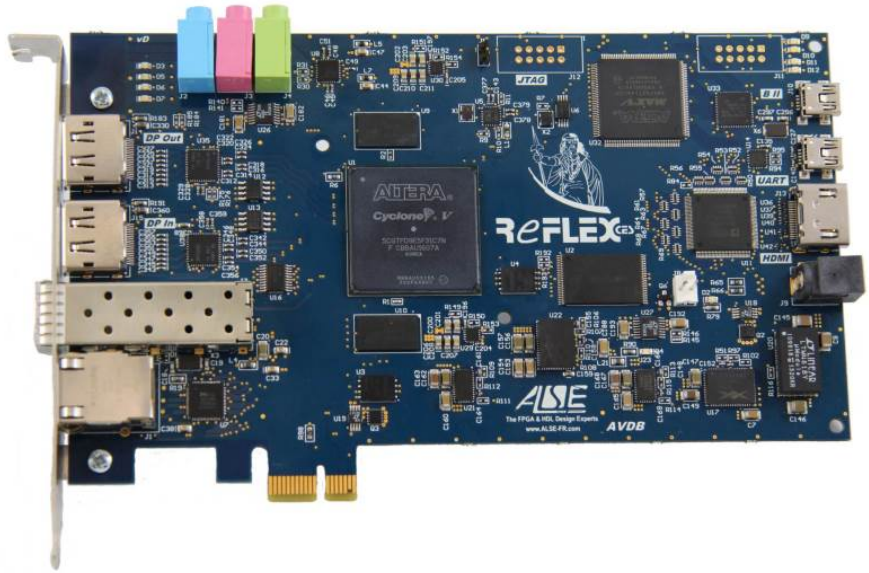
Table of Contents

Chapter 1
AVDB / Clovis kit Package

Thank you for choosing the AVDB Video Development Board designed by ALSE and manufactured by ReFLEX ces !

With standard connectivity, PC card form factor, PCIexpress Gen2, and powerful FPGA, AVDB is an affordable but capable platform to implement and deploy all kinds of video applications.

1 Package Contents



- AVDB board (above) with bracket & heatsink installed
- A 12V DC power supply
- A Quick Start Guide sheet
- A mini-USB cable (for USB Blaster II or UART use)
- An Ethernet RJ45 cable
- A “DisplayPort to HDMI” adapter

2 Getting Started

Follow the Instructions on the Quick Start Guide:

- **Register as an AVDB owner !**
  If you didn't do so already, please register at ReflexCES.
  You can also **send an email to info@alse-fr.com**.
  ALSE contact information is reproduced next page.
- Unpack carefully the board.
- Verify that the Heatsink is properly fitted on the FPGA with thermal compound.

  Using AVDB without proper cooling may harm the FPGA and the board, and is not covered by the warranty.
- Decide whether you want to use AVDB inside a PC (and likely use PCIexpress connectivity in your projects) or as a stand alone board.
- If you use AVDB **inside a PC**, the power can be supplied through the PCIexpress connector. You DO NOT have to use the Jack connector.
- If you use AVDB as a standalone board, use the provided power adapter :
  12V - 3A power supply with “+” in the center of the jack.

  Using an **incorrect** power supply **will damage** AVDB, and this is not covered by the warranty.
- If you want to use an **HDMI display**, you need to use the provided **DisplayPort to HDMI adapter**.
- For many Video projects, an **HDMI Video source** (without contents protection) will be useful. At ALSE, we use easily the **Raspberry Pi** with **OpenElec XBMC Media Center**.
  It allows to select accurately all kinds of video output formats and supports well full-HD streaming (even from a USB key). Alternatively, a PC can be a versatile video source.

3 Resources

To develop your own video applications and run them on AVDB, you will need the last version of this Manual, and you will also need the Intel-FPGA development tools, and Quartus Prime in particular.
Starting with version 16.0, you can use **Quartus Prime Lite Edition (free)**.
With previous versions (like 15.1), you had to use the *Subscription* Edition (not free).

As already mentioned on the previous page, to access more confidential information and to benefit from special offers reserved to AVDB owners, you should register as an AVDB user.

As a registered owner you will have access to:

- Detailed AVDB schematics
- Reference Design: Quartus Project + Top level template
- Free OpenCore+ Licenses for several ALSE IPs
- Free UART IP !
- Discounts for Video Training Courses
- Discounts for ALSE IP blocks: - Image & Video Compression/Decompression IPs,
  - ECC NAND Flash Controller, - Quad-SPI Controller, - HDMI In, - HDMI out,
  - Gigabit Ethernet Communication Kit (GEDEK), - I2C Controller, - UART Master port ...
- Discount for the Intel-FPGA Video IP Suite (VIP)
- Access to the latest technical information
- ALSE Technical support

As a registered AVDB owner, you are welcome to access ALSE Technical Support or IP Sales information by E-mail or by telephone (extended CET hours):

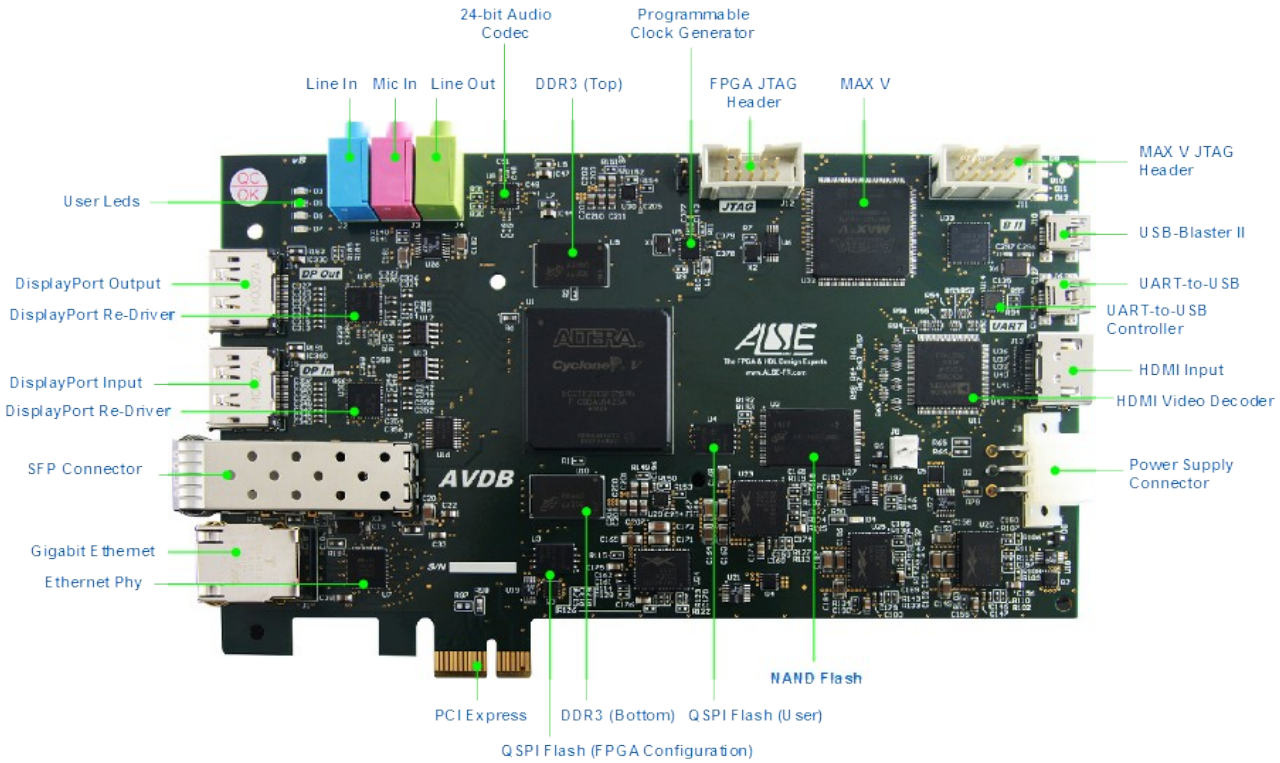
ALSE – Paris France
☎ Tel +33 1 84 16 32 32 (CET)
E-mail: info@alse-fr.com

Chapter 2
Introduction to AVDB

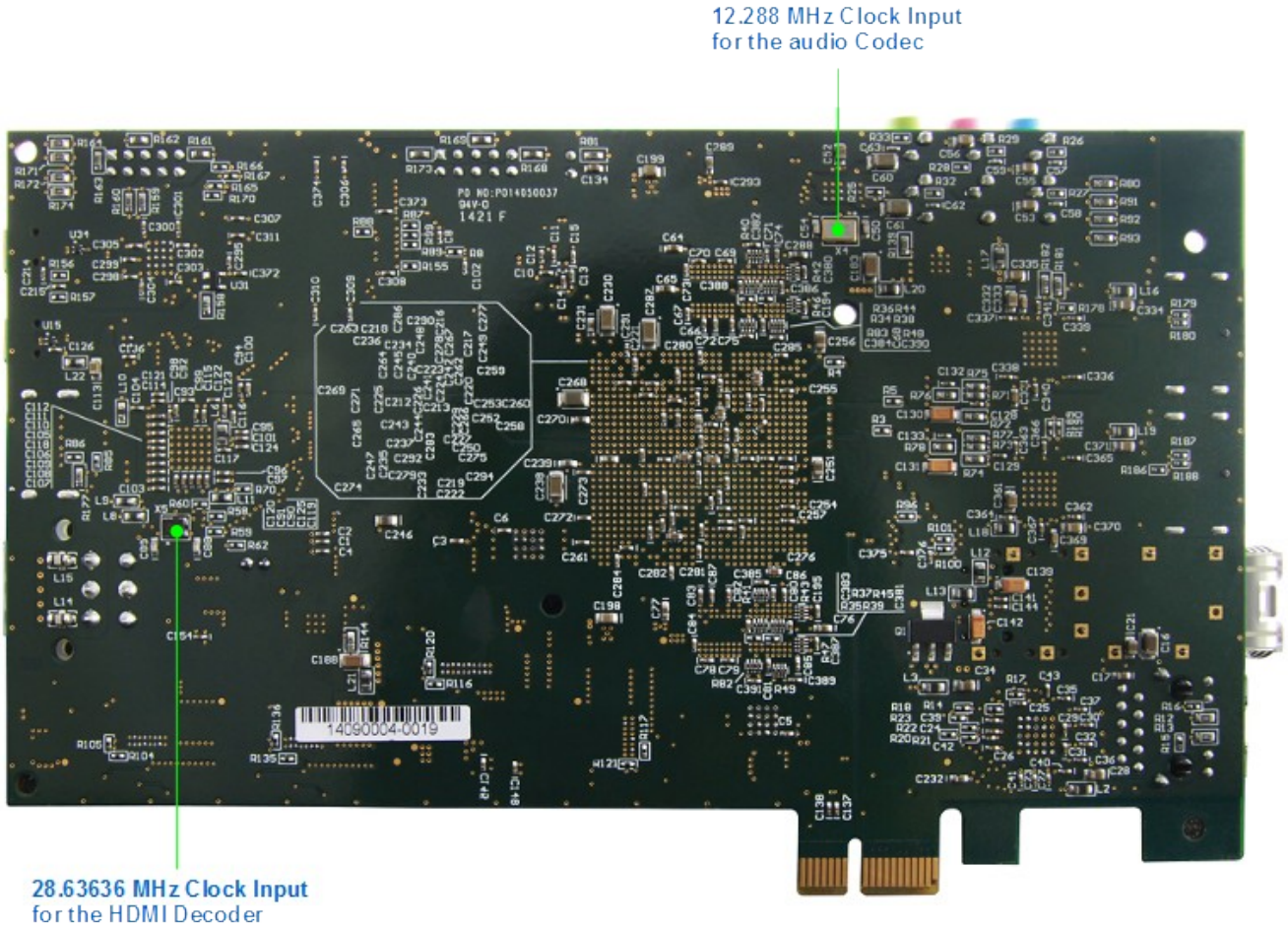
This chapter provides an introduction to the features and design characteristics of the board.

1 Layout and Components

The picture below shows the board and indicates the location of the connectors and key components.



The picture below shows the bottom side of AVDB which mainly hosts decoupling and passive components and a couple of oscillators.



2 Main Features

AVDB is a feature-rich board allowing a large range of designs and applications, especially for video and multimedia applications.

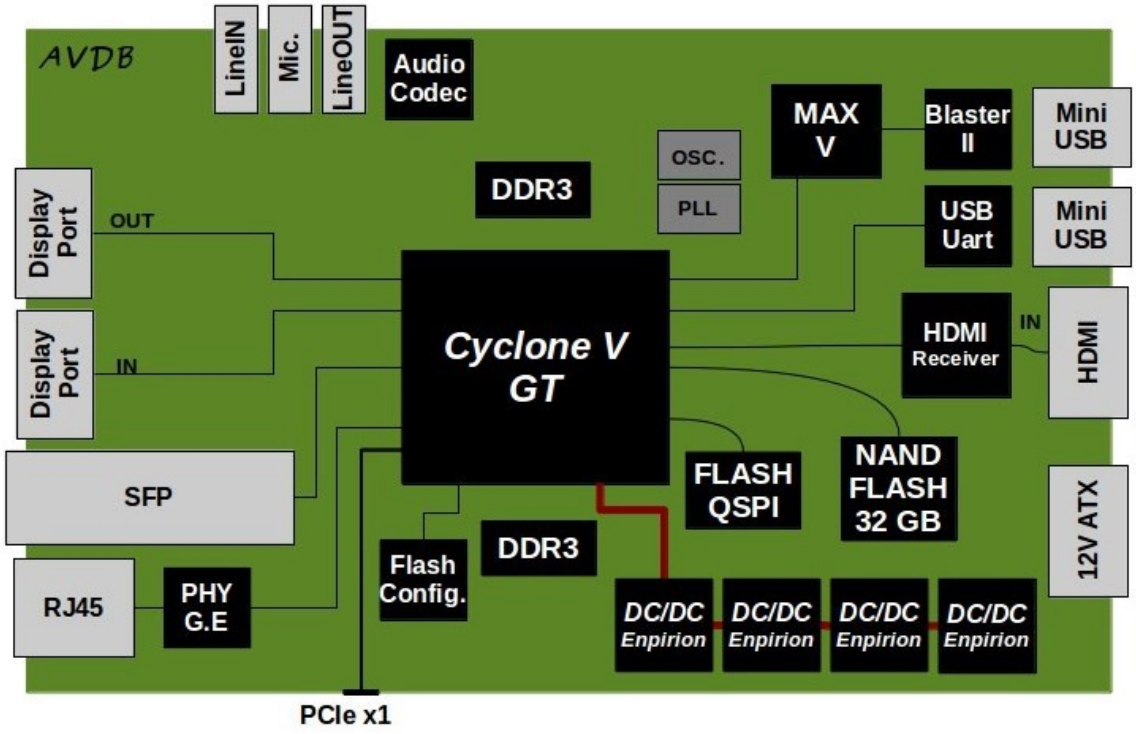
The main devices available are:

- **FPGA**
  Intel-FPGA Cyclone® V GT 5CGTFD9E5F31C7 device:
  113,560 logic elements (113560 ALM, 227120 registers)
  12,492,800 Memory bits (1220 M10K)
  342 DSP Blocks
  12 PLLs
  8 Fractional PLLs
  4 DLLs
  2 Hard Memory Controllers
  12 HSSI PMA RX Channels
  12 HSSI PMA TX Channels

- Serial configuration Flash - 256 Mbits Quad SPI from Micron
- User Quad SPI Nor Flash – 256 Mbits Quad SPI from Micron
- USB2 JTAG onboard adapter = USB Blaster II for programming and debugging (MAX V - 5M570ZT144C5N)
- One 50 MHz clock source
- HDMI Video Input, HDMI 1.4 Compliant, using Xpressview™ Advantiv HDMI Receiver ADV7619
- DisplayPort Video Input connector (v1.1 and DP v1.2 Compliant) with re-driver
- DisplayPort Video Output connector (and re-driver)
- 24-bit audio CODEC (SSM2603) with line-in, line-out, and mic-in jacks
- 2 x 4Gbits of 400MHz DDR3 SDRAM memory (16-bit data bus)
- 32 Gbits NAND Flash
- PCI Express Gen2 x1 lane (compatible with Gen1)
- 1 Gigabit Ethernet PHY (triple speed Micrel KSZ9021GN) with RJ45 connector
- 1 SFP Interface (for SDI adapter for example)
- SiliconLabs Programmable Clock Generator (Si5338A PLL)
- Uart over USB (FTDI FT230XQ) with Mini-B Connector
- 4 user LEDs
- Power consumption: 12V < 2A, either through PCIexpress edge connector or Jack and external Power Supply.

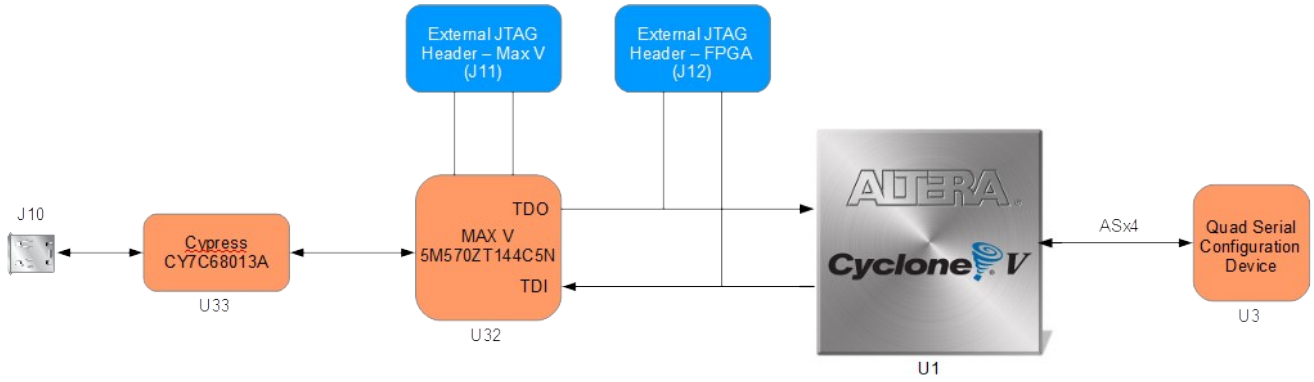
3 AVDB Block Diagram

The block diagram below show the various essential components and their connections.



Chapter 3
Using AVDB

1 FPGA Configuration



IMPORTANT !
For test and production purpose, AVDB has provision for two HE10x10 connectors (J11 & J12). However, **you should NOT try to use any of these connectors !**

You do not need an external programming adapter: **AVDB includes an on-board High Speed USB Blaster II JTAG Programming adapter** (**J10** mini USB connector) which can be used either:

- To program the Configuration Flash (U3)
- To download the FPGA directly
- To interact with the JTAG debugging tools in real time (SignalTap II, In-Ststem Memory Content Editor, In-System Sources and Probes, SystemConsole, NiosII debugger, JTAG Master Debug, etc).
- To transfer Debugging data from the FPGA to the PC at very high speed (Master debug mode accessible from the Master console utilities).

Every time the board is powered up, the Cyclone V FPGA tries to download its configuration from the external Quad-SPI Flash (EPCQ256-U3) using the Active Serial mode x4.

AVDB is normally delivered with a demo bitstream stored in the configuration Flash. This demo is described at the end of this manual and can be downloaded from ALSE web site.

Note that GEDEK (Gigabit Ethernet Communication Kit from ALSE) has an option to remotely program the configuration Flash (User & Safe configurations) through Ethernet. Contact ALSE.

2 Clock Circuitry

The following Table shows all the potential external clock signals connected to the Cyclone V GT FPGA on AVDB.

Pin Assignment of Clock Inputs

Signal Name	FPGA Pin	Description	I/O Standard	Frequency
EXT_CLK50MHZ	PIN_T23	50 MHz clock input	3.3V	50 MHz
DPS_CLK	PIN_N7 & PIN_P8	DP Reference Clock (Transceiver input clock)	IO_STANDARD LVDS	Programmable through Si5338
SFP_CLK	PIN_R7 & PIN_R8	SFP Reference Clock (Transceiver input clock)	1.5-V PCML	Programmable through Si5338
FPGA_CLK	PIN_J18 & PIN_H19	DDR3 HMC Reference Clock input (Top)	IO_STANDARD LVDS	Programmable through Si5338
FPGA_CLK2_N	PIN_AA15 & PIN_Y15	DDR3 HMC Reference Clock input (Bottom)	IO_STANDARD LVDS	Programmable through Si5338

3 Peripherals Connected to the FPGA

User LEDs (active high)

There are four user-controllable LEDs connected to the FPGA.

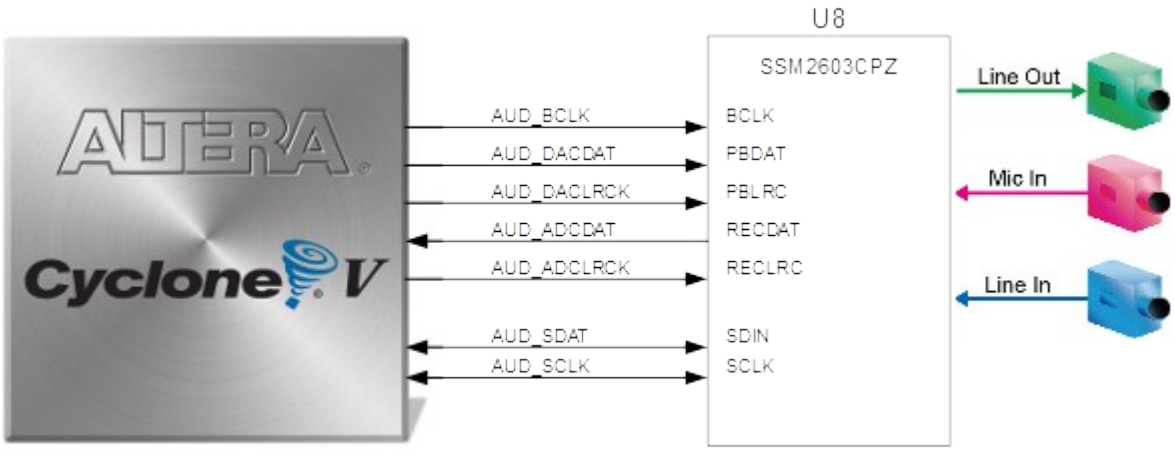
Each LED is driven directly and individually by the Cyclone V FPGA: driving its associated pin to a high logic level turns the LED on, driving to a logic low level turns it off.

The Table below lists the pin assignments of user LEDs.

Signal Name	FPGA Pin	Description	I/O Standard
LED0	PIN_AB26	LED D3 – Red LED	3.3V
LED1	PIN_Y26	LED D5 – Green LED	3.3V
LED2	PIN_AA26	LED D6 – Red LED	3.3V
LED3	PIN_Y25	LED D7 – Green LED	3.3V

Audio Codec

AVDB offers high-quality 24-bit audio using the SSM2603 Analog Devices audio CODEC (Encoder/Decoder). This device supports MICrophone-in, Line-in, and Line-out ports, with adjustable sample rate from 8 kHz to 96 kHz. The SSM2603 is controlled via serial I2C bus, which is connected to the Cyclone V FPGA The I2C address used on board for the audio Codec is 0x34.



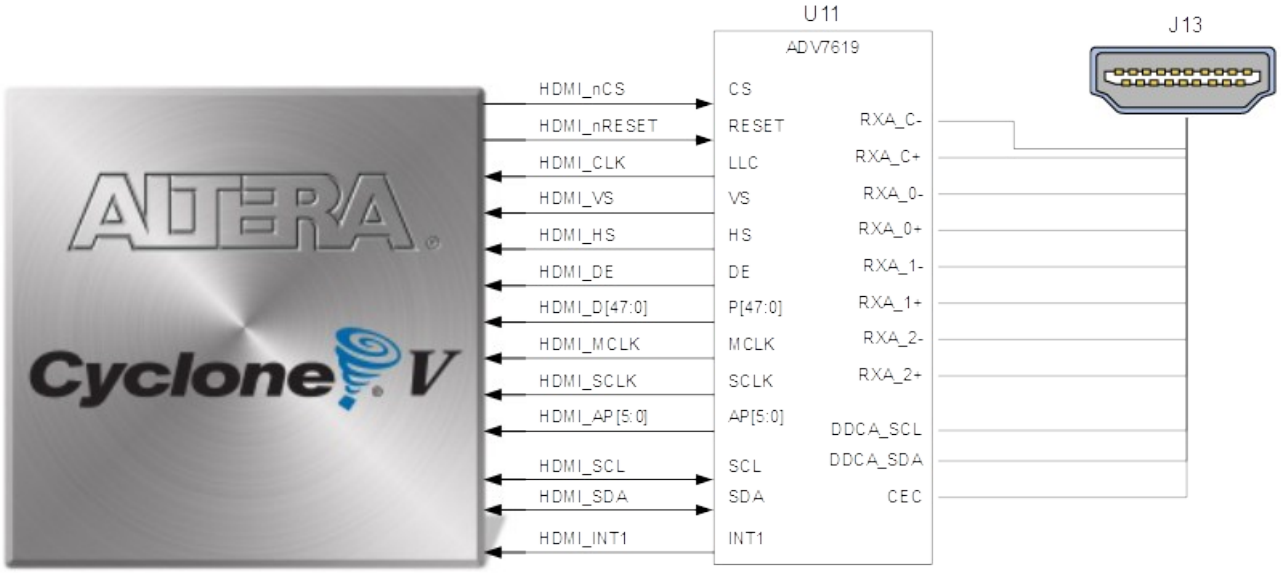
The associated pin assignment of the Audio Codec to the FPGA is listed in the Table below:

Signal Name	*FPGA Pin*	*Description*	*I/O Standard*
AUD_BCLK	**PIN_H24**	**Audio CODEC Bit Clock**	**3.3V**
AUD_DACDAT	**PIN_H25**	**Audio CODEC DAC Data**	**3.3V**
AUD_DACLRCK	**PIN_F25**	**Audio CODEC DAC Sampling Rate Clock**	**3.3V**
AUD_ADCDAT	**PIN_F26**	**Audio CODEC ADC Data**	**3.3V**
AUD_ADCLRCK	**PIN_E27**	**Audio CODEC ADC Sampling Rate Clock**	**3.3V**
AUD_SDAT	**PIN_J22**	**I2C Clock**	**3.3V**
AUD_SCLK	**PIN_J23**	**I2C Data**	**3.3V**

HDMI Input

AVDB includes an HDMI receiver: Dual Port Xpressview™ Advantiv HDMI Receiver, Analog Device ADV7619 (one port used) which is attached to the HDMI Connector (J13).

This video chip supports all TV formats defined in HDMI 1.4 specification and delivers to the FPGA the decoded stream through a video parallel interface.



The main features of the ADV7619 HDMI receiver chip are :

- HDMI 1.4a features receiver compliant (mandatory and additional 3D Video formats supported)
- Supporting link speeds up to 3 GHz (Maximum TMDS Clock frequency of 297 MHz)
- Supports input resolutions up to 4kx2k
- Supports up to 48-bit Deep Color with 36/20/24 bit support
- Digital audio output interface supporting High bit rate, Direct Stream Digital, S/PDIF (IEC 60958 compatible) and up to 4 I2S outputs

More information about the ADV7619 HDMI Receiver is available in its Datasheet, which can be found on the manufacturer's website (make sure you collect the latest version).

The HDMI receiver is controlled through the I2C Interface. The device I2C address of the HDMI receiver is 0x98.

Please refer to AVDB schematics for all the electrical details.

The pin assignment table between the FPGA and the ADV7619 video chip is next page.

Pin Assignment of HDMI Video input Codec

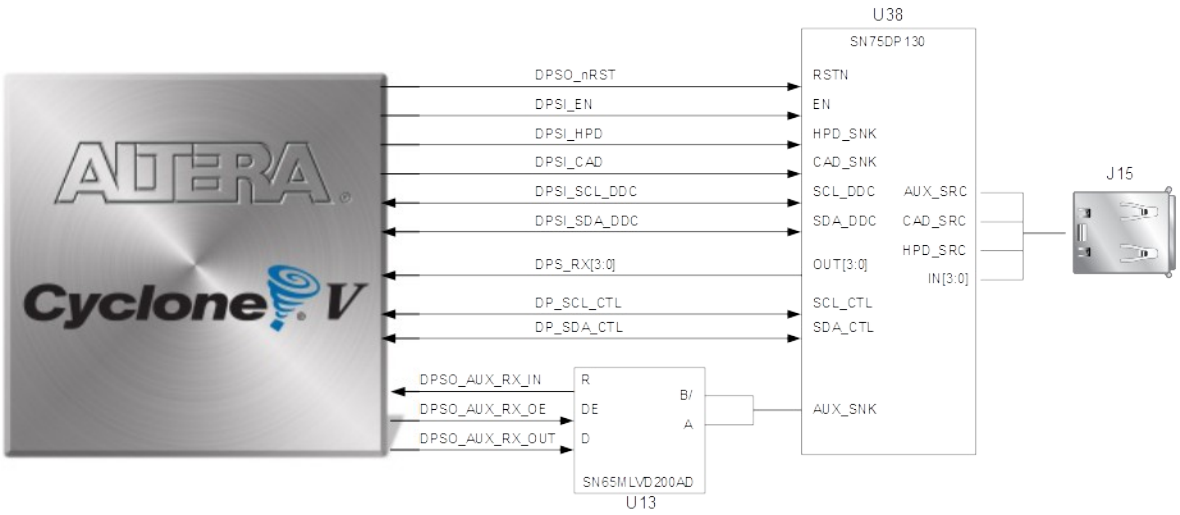
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
HDMI_nRESET	PIN_R25	Reset – active low	3.3V
HDMI_nCS	PIN_R27	Chip Select - active-low	3.3V
HDMI_INT1	PIN_R22	Interrupt signal	3.3V
HDMI_SCL	PIN_R21	I2C Serial Clock	3.3V
HDMI_SDA	PIN_R20	I2C Serial Data	3.3V
HDMI_CLK	PIN_P22	Pixel Output Clock	3.3V
HDMI_VS	PIN_N24	Vertical Synchronization Output	3.3V
HDMI_HS	PIN_N25	Horizontal Synchronization Output	3.3V
HDMI_DE	PIN_N30	Pixel Data Enable	3.3V
HDMI_D[47]	PIN_C29	Pixel Output Port	3.3V
HDMI_D[46]	PIN_C30	Pixel Output Port	3.3V
HDMI_D[45]	PIN_D27	Pixel Output Port	3.3V
HDMI_D[44]	PIN_D28	Pixel Output Port	3.3V
HDMI_D[43]	PIN_D29	Pixel Output Port	3.3V
HDMI_D[42]	PIN_D30	Pixel Output Port	3.3V
HDMI_D[41]	PIN_E28	Pixel Output Port	3.3V
HDMI_D[40]	PIN_E30	Pixel Output Port	3.3V
HDMI_D[39]	PIN_F28	Pixel Output Port	3.3V
HDMI_D[38]	PIN_F29	Pixel Output Port	3.3V
HDMI_D[37]	PIN_F30	Pixel Output Port	3.3V
HDMI_D[36]	PIN_G26	Pixel Output Port	3.3V
HDMI_D[35]	PIN_G29	Pixel Output Port	3.3V
HDMI_D[34]	PIN_G28	Pixel Output Port	3.3V
HDMI_D[33]	PIN_G27	Pixel Output Port	3.3V
HDMI_D[32]	PIN_H30	Pixel Output Port	3.3V
HDMI_D[31]	PIN_H27	Pixel Output Port	3.3V
HDMI_D[30]	PIN_H26	Pixel Output Port	3.3V
HDMI_D[29]	PIN_H29	Pixel Output Port	3.3V
HDMI_D[28]	PIN_J30	Pixel Output Port	3.3V
HDMI_D[27]	PIN_J29	Pixel Output Port	3.3V
HDMI_D[26]	PIN_J28	Pixel Output Port	3.3V
HDMI_D[25]	PIN_K21	Pixel Output Port	3.3V
HDMI_D[24]	PIN_K22	Pixel Output Port	3.3V
HDMI_D[23]	PIN_L21	Pixel Output Port	3.3V
HDMI_D[22]	PIN_L23	Pixel Output Port	3.3V
HDMI_D[21]	PIN_J25	Pixel Output Port	3.3V
HDMI_D[20]	PIN_J27	Pixel Output Port	3.3V
HDMI_D[19]	PIN_K27	Pixel Output Port	3.3V
HDMI_D[18]	PIN_K28	Pixel Output Port	3.3V
HDMI_D[17]	PIN_K30	Pixel Output Port	3.3V
HDMI_D[16]	PIN_K26	Pixel Output Port	3.3V
HDMI_D[15]	PIN_K25	Pixel Output Port	3.3V

HDMI_D[14]	PIN_L30	Pixel Output Port	3.3V
HDMI_D[13]	PIN_L25	Pixel Output Port	3.3V
HDMI_D[12]	PIN_L24	Pixel Output Port	3.3V
HDMI_D[11]	PIN_L26	Pixel Output Port	3.3V
HDMI_D[10]	PIN_L28	Pixel Output Port	3.3V
HDMI_D[9]	PIN_L29	Pixel Output Port	3.3V
HDMI_D[8]	PIN_M21	Pixel Output Port	3.3V
HDMI_D[7]	PIN_M22	Pixel Output Port	3.3V
HDMI_D[6]	PIN_M23	Pixel Output Port	3.3V
HDMI_D[5]	PIN_M27	Pixel Output Port	3.3V
HDMI_D[4]	PIN_M29	Pixel Output Port	3.3V
HDMI_D[3]	PIN_M28	Pixel Output Port	3.3V
HDMI_D[2]	PIN_N20	Pixel Output Port	3.3V
HDMI_D[1]	PIN_N22	Pixel Output Port	3.3V
HDMI_D[0]	PIN_N21	Pixel Output Port	3.3V
HDMI_MCLK	PIN_P29	Audio Master clock	3.3V
HDMI_SCLK	PIN_P28	Audio Serial Clock	3.3V
HDMI_AP[5]	PIN_P30	Audio Output pins	3.3V
HDMI_AP[4]	PIN_P25	Audio Output pins	3.3V
HDMI_AP[3]	PIN_N29	Audio Output pins	3.3V
HDMI_AP[2]	PIN_N27	Audio Output pins	3.3V
HDMI_AP[1]	PIN_N26	Audio Output pins	3.3V
HDMI_AP[0]	PIN_P20	Audio Output pins	3.3V

DisplayPort Video Input

AVDB also offers a DisplayPort Video input on connector **J15**.

Unlike HDMI input (which is decoded by a dedicated video chip see previous section) the DisplayPort video input signal only traverses a High speed DisplayPort Re-Driver (SN75DP130 from Texas Instruments) to reach the FPGA SerDes inputs. See below:



We notice on the diagram above that an extra tri-state buffer has been added on the low speed control lines.

The re-driver is used on the Sink main link signal path to improve signal integrity and ensure PHY Layer compatibility at the DisplayPort connector. The Re-Driver eliminates signal integrity issues when running at high bit rate.

To decode a video signal arriving on this input, all the protocol decoding and control management must be implemented in the FPGA. Intel-FPGA has an IP available for this purpose, and it has been tested successfully on AVDB. A Reference Design is available demonstrating the Intel-FPGA Display Port IP.

The pin assignment of the interconnects between the re-driver & tristate buffer and the FPGA is described in the Table below.

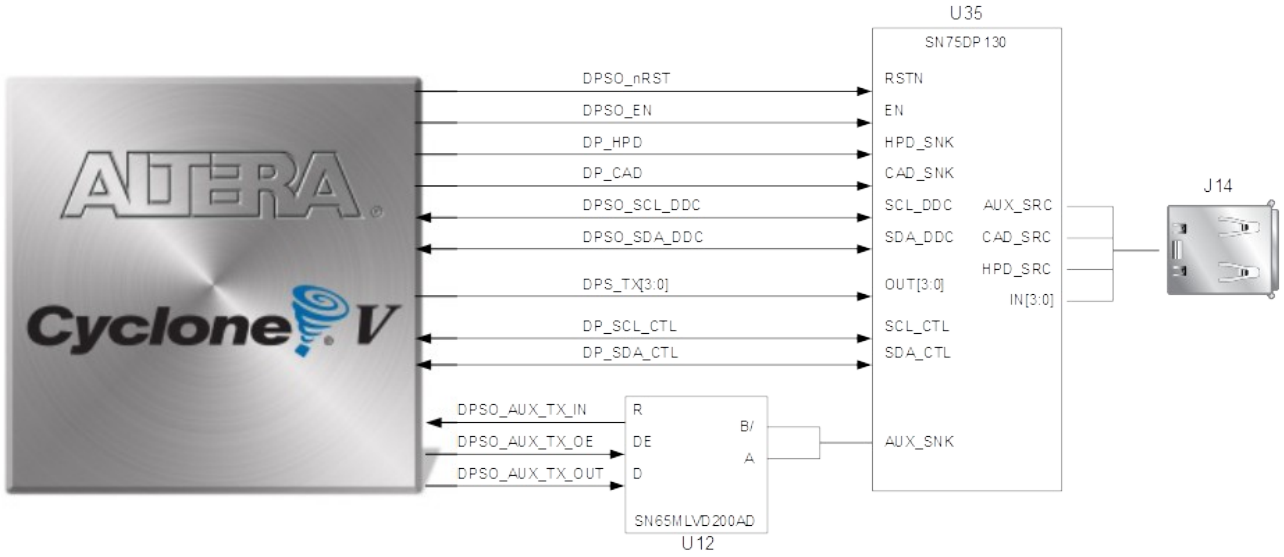
Pin Assignment of DisplayPort Interface

Signal Name	*FPGA Pin*	*Description*	*I/O Standard*
DPSO_nRST	PIN_U23	Re-Driver Device Reset (active low)	3.3V
DPSI_EN	PIN_Y22	Re-Driver Enable	3.3V
DPSI_HPD	PIN_AE25	Hot Plug Detect from Sink	3.3V
DPSI_CAD	PIN_AF25	Cable Adapter Detect	3.3V
DPSI_SCL_DDC	PIN_AE26	I2C Serial Data – Data Channel Configuration	3.3V
DPSI_SDA_DDC	PIN_AF26	I2C Serial Clock – Data Channel Configuration	3.3V
DPSO_AUX_RX_IN	PIN_AA25	Auxiliary Data Channel input	3.3V
DPSO_AUX_RX_OE	PIN_AA24	Auxiliary Data Channel output enable	3.3V
DPSO_AUX_RX_OUT	PIN_AA23	Auxiliary Data Channel output	3.3V
DPS_RX0_p	PIN_R2	DisplayPort main Link – Lane 0	1.5-V PCML
DPS_RX0_n	PIN_R1	DisplayPort main Link – Lane 0	1.5-V PCML
DPS_RX1_p	PIN_N2	DisplayPort main Link – Lane 1	1.5-V PCML
DPS_RX1_n	PIN_N1	DisplayPort main Link – Lane 1	1.5-V PCML
DPS_RX2_p	PIN_L2	DisplayPort main Link – Lane 2	1.5-V PCML
DPS_RX2_n	PIN_L1	DisplayPort main Link – Lane 2	1.5-V PCML
DPS_RX3_p	PIN_J2	DisplayPort main Link – Lane 3	1.5-V PCML
DPS_RX3_n	PIN_J1	DisplayPort main Link – Lane 3	1.5-V PCML
DP_SDA_CTL	PIN_AE23	I2C Serial Data – Re-Driver Configuration	3.3V
DP_SCL_CTL	PIN_AF24	I2C Serial Clock - Re-Driver Configuration	3.3V

DisplayPort Video Output

AVDB offers a Video output connector: **J14**. This is a DisplayPort Standard Connector.

The FPGA drives the output connector through a re-driver U35 (and a tri-state buffer for the low speed line).



Like for DP input, one can use the Intel-FPGA DisplayPort IP to drive a DisplayPort Monitor (eg).

Note however that, if using inside the FPGA an **HDMI** output IP (like the one available at ALSE), one can drive an HDMI monitor or TV by using a level translation adapter as the one coming with the board.

In practice, the DisplayPort output connector is very often used with HDMI peripherals.

The pin assignment of the interconnects between the FPGA and the redriver & tristate buffer is described in the Table next page.

Pin Assignment of DisplayPort Interface with the Re-Driver

Signal Name	*FPGA Pin*	*Description*	*I/O Standard*
DPS_CLK_p	**PIN_P8**	**Display Port Input reference clock – Serdes Input**	**LVDS**
DPS_CLK_n	**PIN_N7**	**Display Port Input reference clock – Serdes Input**	**LVDS**
DPS_TX0_p	**PIN_P4**	**DisplayPort Main Link – Lane 0**	**1.5-V PCML**
DPS_TX0_n	**PIN_P3**	**DisplayPort Main Link – Lane 0**	**1.5-V PCML**
DPS_TX1_p	**PIN_M4**	**DisplayPort Main Link – Lane 1**	**1.5-V PCML**
DPS_TX1_n	**PIN_M3**	**DisplayPort Main Link – Lane 1**	**1.5-V PCML**
DPS_TX2_p	**PIN_K4**	**DisplayPort Main Link – Lane 2**	**1.5-V PCML**
DPS_TX2_n	**PIN_K3**	**DisplayPort Main Link – Lane 2**	**1.5-V PCML**
DPS_TX3_p	**PIN_H4**	**DisplayPort Main Link – Lane 3**	**1.5-V PCML**
DPS_TX3_n	**PIN_H3**	**DisplayPort Main Link – Lane 3**	**1.5-V PCML**
DPSO_nRST	**PIN_U23**	**Re-Driver Device Reset (active low)**	**3.3V**
DPSO_EN	**PIN_U22**	**Re-Driver Enable**	**3.3V**
DP_HPD	**PIN_T25**	**Hot Plug Detect to the source**	**3.3V**
DP_CAD	**PIN_U26**	**Cable Adapter Detect**	**3.3V**

DPSO_AUX_TX_IN	**PIN_V24**	**Auxiliary Data Channel input**	**3.3V**
DPSO_AUX_TX_OE	**PIN_V25**	**Auxiliary Data Channel Output**	**3.3V**
DPSO_AUX_TX_OUT	**PIN_V26**	**Auxiliary Data Channel Output**	**3.3V**
DPSO_SDA_DDC	**PIN_V21**	**I2C Serial Data – Data Channel Configuration**	**3.3V**
DPSO_SCL_DDC	**PIN_U21**	**I2C Serial Clock – Data Channel Configuration**	**3.3V**
DP_SDA_CTL	**PIN_AE23**	**I2C Serial Data – Re-Driver Configuration**	**3.3V**
DP_SCL_CTL	**PIN_AF24**	**I2C Serial Clock - Re-Driver Configuration**	**3.3V**

DDR3

The FPGA is connected to two DDR3 memory chips ref **MT41J256M16HA-125** from Micron running at 2 x 400 MHz, which provides a raw Bandwidth of ~25.6 Gigabits/s. This memory (2 x 4 Gbits) can be used for any task requiring high Read/Write bandwidth and a large memory capacity.

The two DDR3 devices connected to the FPGA are identical. Each DDR3 device has a capacity of 4 Gbits and the data bandwidth is in 16-bit. The signals of each DDR3 devices are connected to independent dedicated Hard Memory Controllers (target speed is 400 MHz).

More information about the DDR3 memory Chips are available in the Micron Datasheet.

The Table below lists the DDR3 pin assignments.

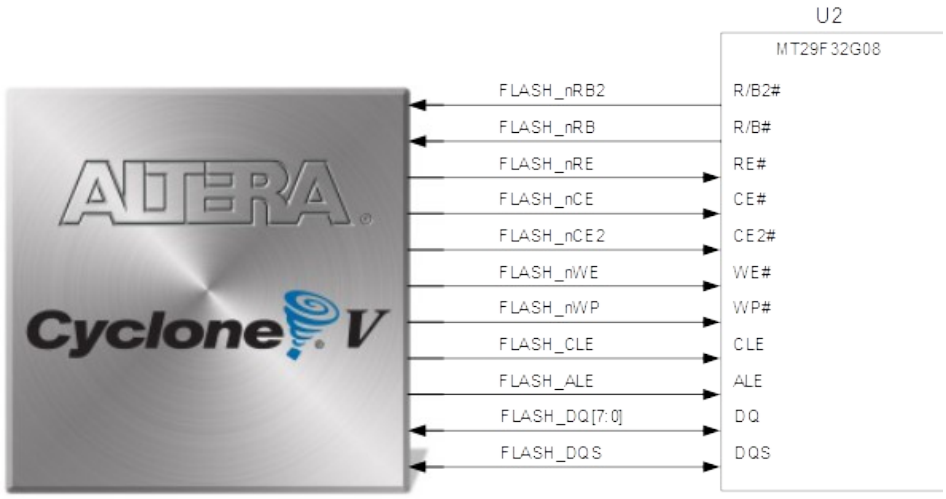
Signal Name	FPGA Pin	Description	I/O Standard
DDR3T_RST_N	PIN_B22	DDR3 Reset (active low)	1.5V
DDR3T_CLK[0]	PIN_M9	DDR3 Clock	DIFFERENTIAL 1.5-V SSTL Class I
DDR3T_CLK_N[0]	PIN_M8	DDR3 Clock	DIFFERENTIAL 1.5-V SSTL Class I
DDR3T_CKE[0]	PIN_A18	DDR3 Clock Enable	SSTL-15 Class I
DDR3T_BA[0]	PIN_A10	DDR3 Bank Address	SSTL-15 Class I
DDR3T_BA[1]	PIN_F15	DDR3 Bank Address	SSTL-15 Class I
DDR3T_BA[2]	PIN_E15	DDR3 Bank Address	SSTL-15 Class I
DDR3T_A[0]	PIN_B11	DDR3 Address	SSTL-15 Class I
DDR3T_A[1]	PIN_A11	DDR3 Address	SSTL-15 Class I
DDR3T_A[2]	PIN_F9	DDR3 Address	SSTL-15 Class I
DDR3T_A[3]	PIN_E10	DDR3 Address	SSTL-15 Class I
DDR3T_A[4]	PIN_F16	DDR3 Address	SSTL-15 Class I
DDR3T_A[5]	PIN_E16	DDR3 Address	SSTL-15 Class I
DDR3T_A[6]	PIN_D9	DDR3 Address	SSTL-15 Class I
DDR3T_A[7]	PIN_C10	DDR3 Address	SSTL-15 Class I
DDR3T_A[8]	PIN_E12	DDR3 Address	SSTL-15 Class I
DDR3T_A[9]	PIN_D13	DDR3 Address	SSTL-15 Class I
DDR3T_A[10]	PIN_B7	DDR3 Address	SSTL-15 Class I
DDR3T_A[11]	PIN_A8	DDR3 Address	SSTL-15 Class I
DDR3T_A[12]	PIN_B6	DDR3 Address	SSTL-15 Class I

DDR3T_A[13]	PIN_A6	DDR3 Address	SSTL-15 Class I
DDR3T_A[14]	PIN_E11	DDR3 Address	SSTL-15 Class I
DDR3T_CS_N[0]	PIN_J15	DDR3 Chip Select	SSTL-15 Class I
DDR3T_RAS_N[0]	PIN_B8	DDR3 Row Address Strobe	SSTL-15 Class I
DDR3T_CAS_N[0]	PIN_C9	DDR3 Column Address Strobe	SSTL-15 Class I
DDR3T_WE_N[0]	PIN_C7	DDR3 Write Enable (active low)	SSTL-15 Class I
DDR3T_DQ[0]	PIN_C15	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[1]	PIN_C16	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[2]	PIN_C11	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[3]	PIN_A13	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[4]	PIN_D12	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[5]	PIN_D17	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[6]	PIN_E17	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[7]	PIN_A14	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[8]	PIN_B17	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[9]	PIN_C17	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[10]	PIN_A16	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[11]	PIN_C14	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[12]	PIN_F18	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[13]	PIN_G18	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[14]	PIN_B18	DDR3 Data	SSTL-15 Class I
DDR3T_DQ[15]	PIN_A19	DDR3 Data	SSTL-15 Class I
DDR3T_DM[0]	PIN_B14	DDR3 Data Mask	SSTL-15 Class I
DDR3T_DM[1]	PIN_B19	DDR3 Data Mask	SSTL-15 Class I
DDR3T_DQS[0]	PIN_K17	DDR3 Data Strobe	SSTL-15 Class I
DDR3T_DQS_N[0]	PIN_J17	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3T_DQS[1]	PIN_K16	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3T_DQS_N[1]	PIN_L16	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3T_ODT[0]	PIN_B13	DDR3 On-Die Termination	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_RST_N	PIN_AK21	DDR3 Reset (active low)	1.5V
DDR3B_CLK[0]	PIN_Y13	DDR3 Clock	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_CLK_N[0]	PIN_AA14	DDR3 Clock	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_CKE[0]	PIN_AK18	DDR3 Clock Enable	SSTL-15 Class I
DDR3B_BA[0]	PIN_AH9	DDR3 Bank Address	SSTL-15 Class I
DDR3B_BA[1]	PIN_AH10	DDR3 Bank Address	SSTL-15 Class I
DDR3B_BA[2]	PIN_AJ10	DDR3 Bank Address	SSTL-15 Class I
DDR3B_A[0]	PIN_AJ12	DDR3 Address	SSTL-15 Class I
DDR3B_A[1]	PIN_AK12	DDR3 Address	SSTL-15 Class I
DDR3B_A[2]	PIN_AH11	DDR3 Address	SSTL-15 Class I
DDR3B_A[3]	PIN_AH12	DDR3 Address	SSTL-15 Class I
DDR3B_A[4]	PIN_AG13	DDR3 Address	SSTL-15 Class I
DDR3B_A[5]	PIN_AG14	DDR3 Address	SSTL-15 Class I
DDR3B_A[6]	PIN_AK10	DDR3 Address	SSTL-15 Class I
DDR3B_A[7]	PIN_AK11	DDR3 Address	SSTL-15 Class I

DDR3B_A[8]	PIN_AF11	DDR3 Address	SSTL-15 Class I
DDR3B_A[9]	PIN_AG11	DDR3 Address	SSTL-15 Class I
DDR3B_A[10]	PIN_AJ8	DDR3 Address	SSTL-15 Class I
DDR3B_A[11]	PIN_AK8	DDR3 Address	SSTL-15 Class I
DDR3B_A[12]	PIN_AJ7	DDR3 Address	SSTL-15 Class I
DDR3B_A[13]	PIN_AK7	DDR3 Address	SSTL-15 Class I
DDR3B_A[14]	PIN_AF13	DDR3 Address	SSTL-15 Class I
DDR3B_CS_N[0]	PIN_Y12	DDR3 Chip Select	SSTL-15 Class I
DDR3B_RAS_N[0]	PIN_AG9	DDR3 Row Address Strobe	SSTL-15 Class I
DDR3B_CAS_N[0]	PIN_AF9	DDR3 Column Address Strobe	SSTL-15 Class I
DDR3B_WE_N[0]	PIN_AK5	DDR3 Write Enable (active low)	SSTL-15 Class I
DDR3B_DQ[0]	PIN_AF15	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[1]	PIN_AE16	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[2]	PIN_AJ14	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[3]	PIN_AH15	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[4]	PIN_AE17	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[5]	PIN_AD17	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[6]	PIN_AJ15	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[7]	PIN_AF14	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[8]	PIN_AK17	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[9]	PIN_AK16	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[10]	PIN_AG17	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[11]	PIN_AJ18	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[12]	PIN_AG16	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[13]	PIN_AF16	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[14]	PIN_AJ19	DDR3 Data	SSTL-15 Class I
DDR3B_DQ[15]	PIN_AH20	DDR3 Data	SSTL-15 Class I
DDR3B_DM[0]	PIN_AE15	DDR3 Data Mask	SSTL-15 Class I
DDR3B_DM[1]	PIN_AH19	DDR3 Data Mask	SSTL-15 Class I
DDR3B_DQS[0]	PIN_Y16	DDR3 Data Strobe	SSTL-15 Class I
DDR3B_DQS_N[0]	PIN_AA16	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_DQS[1]	PIN_Y17	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_DQS_N[1]	PIN_Y18	DDR3 Data Strobe	DIFFERENTIAL 1.5-V SSTL Class I
DDR3B_ODT[0]	PIN_AH14	DDR3 On-Die Termination	DIFFERENTIAL 1.5-V SSTL Class I

NAND Flash

AVDB includes one **High Performance 32 Gigabits NAND Flash** ref MT29F32G08ABAAAWP from Micron. It can therefore be used to store up to 4 GBytes of User Application Data.



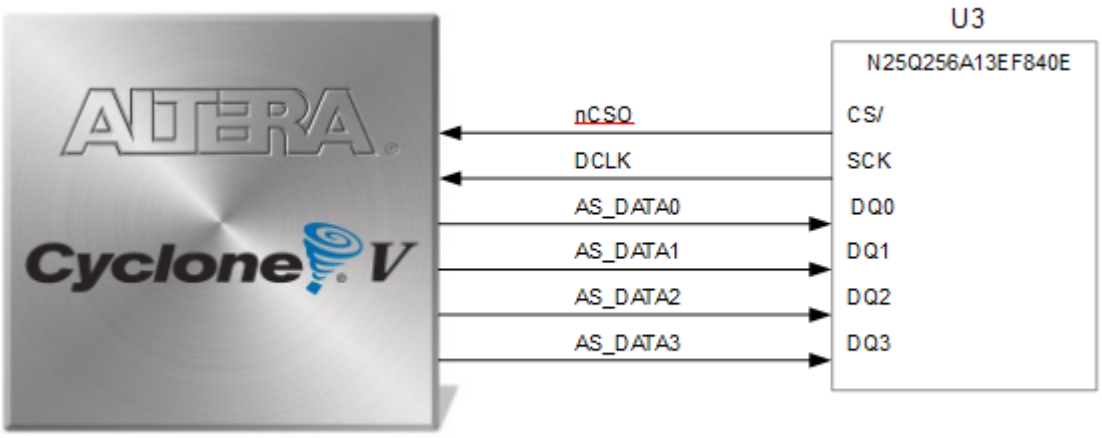
Note that NAND Flash memories are, by definition, not guaranteed without errors and must be driven through a **NAND Flash Controller IP** that provides the adequate level of **Error Check and Correct**. **ALSE has developed this NAND Flash Controller**. Contact ALSE.

The NAND Flash Pin Assignments re described in the Table below :

Signal Name	FPGA Pin	Description	I/O Standard
FLASH_nCE	PIN_AE28	Chip Enable (active low)	3.3V
FLASH_nCE2	PIN_AF30	Chip Enable (active low)	3.3V
FLASH_nWE	PIN_AG29	Write Enable (active low)	3.3V
FLASH_nRE	PIN_AE30	Read Enable (active low)	3.3V
FLASH_ALE	PIN_AF28	Address Latch Enable	3.3V
FLASH_CLE	PIN_AF29	Command Latch enable	3.3V
FLASH_DQ[7]	PIN_AB29	Data bus	3.3V
FLASH_DQ[6]	PIN_AC29	Data bus	3.3V
FLASH_DQ[5]	PIN_AC30	Data bus	3.3V
FLASH_DQ[4]	PIN_AD30	Data bus	3.3V
FLASH_DQ[3]	PIN_AH29	Data bus	3.3V
FLASH_DQ[2]	PIN_AJ30	Data bus	3.3V
FLASH_DQ[1]	PIN_AJ29	Data bus	3.3V
FLASH_DQ[0]	PIN_AJ28	Data bus	3.3V
FLASH_nRYBY	PIN_AD28	Ready/Busy signal (active low)	3.3V
FLASH_nRYBY2	PIN_AD29	Ready/Busy signal (active low)	3.3V
FLASH_nWP	PIN_AG28	Write Protect (active low)	3.3V
FLASH_DQS	PIN_AH30	Data Strobe	3.3V

FPGA Configuration Quad-SPI

It is a 256 Mbits Quad-SPI NOR Flash Memory, ref N25Q256A from Micron.
You can obtain more information about this Memory on the Micron Datasheet.



As previously mentioned, this memory is used by the FPGA to automatically retrieve its configuration at power up (Active-Serial x4 mode).

Once programmed, the FPGA could gain access to this memory at run-time. This is what the GEDEK Remote Update option takes advantage of, and allows the user to update the Safe or User bitstream image through Ethernet.

Connections between the FPGA and the SPI NOR Flash Memory

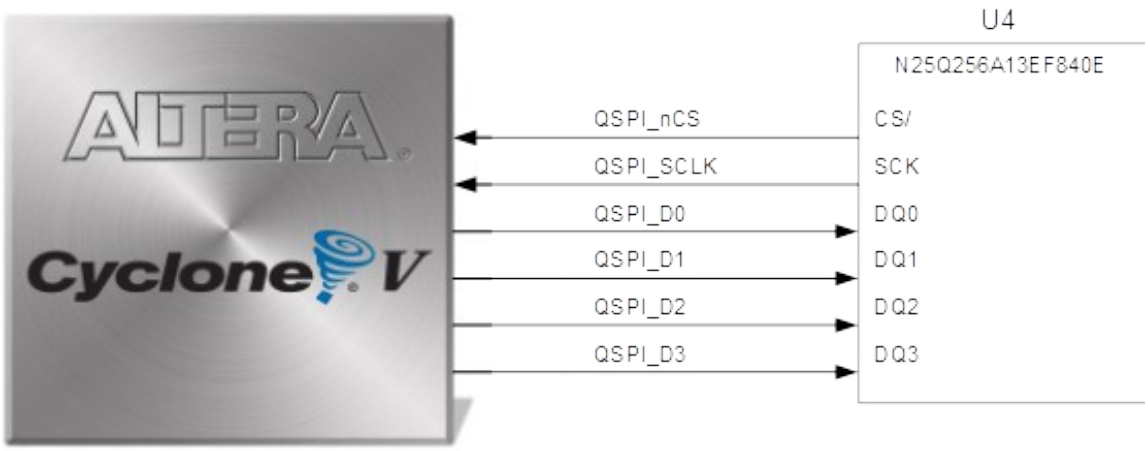
Signal Name	FPGA Pin	Description	I/O Standard
nCSO	PIN_AA7	Chip Select (active low)	3.3V
DCLK	PIN_T7	Serial Clock signal	3.3V
AS_DATA0	PIN_AG5	Input/Output Data	3.3V
AS_DATA1	PIN_AE5	Input/Output Data	3.3V
AS_DATA2	PIN_AE7	Input/Output Data	3.3V
AS_DATA3	PIN_AB7	Input/Output Data	3.3

Note to Designers: you cannot access the Configuration SPI Flash FPGA pins directly ! If you try, you will get an error from the Placer & Router. You have to use the **Intel-FPGA ASMI Parallel IP** (macro) provided by Intel-FPGA.

User Quad-SPI

This second Quad-SPI Flash is also a 256 Mbits Quad-SPI NOR Flash Memory, ref N25Q256A from Micron. You can obtain more information about this Memory on the Micron Datasheet.

It is entirely dedictaed to User Applications.



Note that ALSE has designed an **extremely efficient & High Performance Quad SPI Controller IP**.

Connections between the FPGA and the SPI NOR Flash Memory

Signal Name	FPGA Pin	Description	I/O Standard
QSPI_nCS	PIN_AE27	Chip Select (active low)	3.3V
QSPI_SCLK	PIN_AC27	Serial Clock signal	3.3V
QSPI_D0	PIN_AD27	Input/Output Data	3.3V
QSPI_D1	PIN_AG27	Input/Output Data	3.3V
QSPI_D2	PIN_AH27	Input/Output Data	3.3V
QSPI_D3	PIN_AC26	Input/Output Data	3.3V

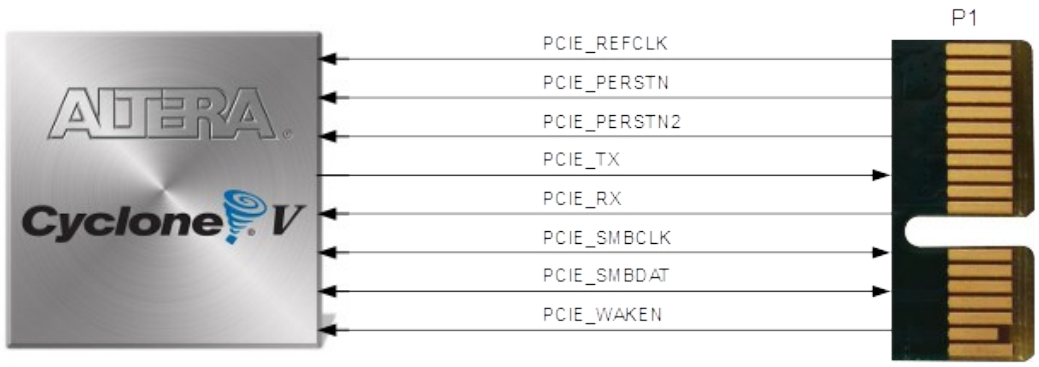
PCIexpress

AVDB can fit in a PC. It fits on any motherboard offering a PCI Express slot (either x1, x4 or x16, Gen1 or Gen2).

Utilizing the built-in transceivers of the Cyclone V GT, it is able to provide a fully integrated PCI Express compliant solution for Gen2 (5.0 Gbits/s) or Gen1 (2.5 Gbits/s) x1 lane applications.

The PCIE_REFCLK_p signal is a differential input that is driven from the PC motherboard through the PCIe edge connector.

The Power is also provided by the Motherboard to AVDB through the edge connector.



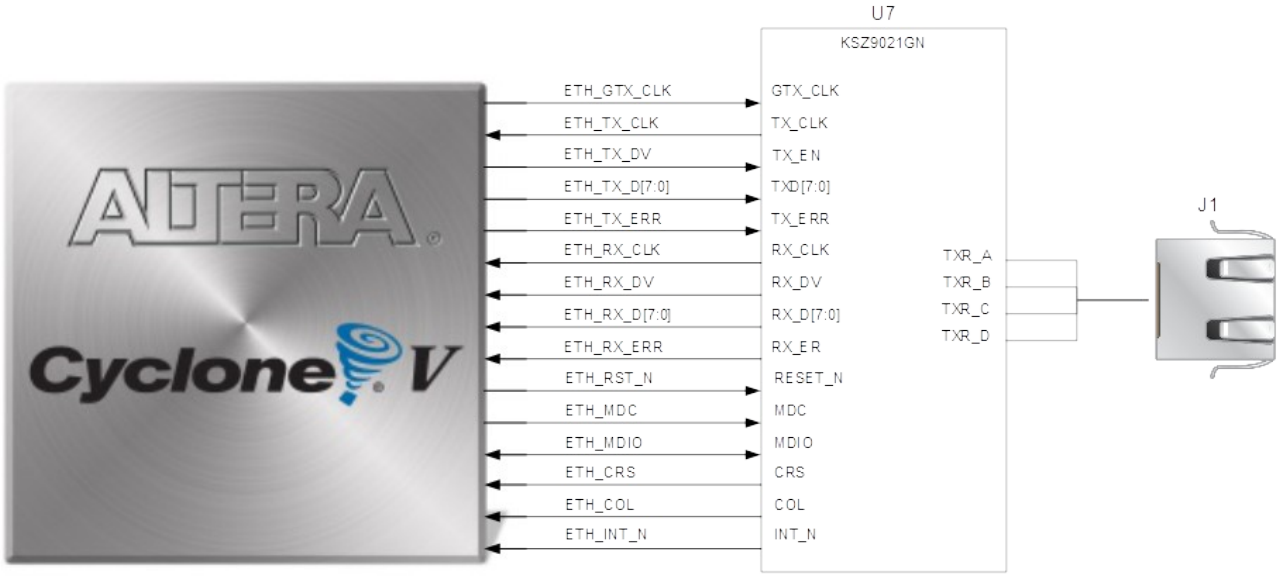
PCI Express Pin Assignments

Signal Name	FPGA Pin	Description	I/O Standard
PCIE_REFCLK_p	PIN_W8	Motherboard reference clock	1.5-V PCML
PCIE_REFCLK_n	PIN_W7	Motherboard reference clock	1.5-V PCML
PCIE_PERSTN	PIN_W24	Reset (active low)	3.3V
PCIE_PERSTN2	PIN_AD25	Reset2 (active low)	3.3V
PCIE_TX_p	PIN_AF4	Add-in card transmit bus	1.5-V PCML
PCIE_TX_n	PIN_AF3	Add-in card transmit bus	1.5-V PCML
PCIE_RX_p	PIN_AG2	Add-in card receive bus	1.5-V PCML
PCIE_RX_n	PIN_AG1	Add-in card receive bus	1.5-V PCML
PCIE_SMBCLK	PIN_AD23	SMB clock	3.3V
PCIE_SMBDAT	PIN_AC24	SMB data	3.3V
PCIE_WAKEN	PIN_AD24	Wake signal (active low)	3.3V

Gigabit Ethernet

AVDB includes a triple-speed 10/100/1000 Mbps Gigabit Ethernet PHY from Micrel, ref KSZ9021GN.

The KSZ9021GN supports GMII/MII MAC interface. To facilitate the timing convergence, a GMII scheme (Single Data Rate) has been implemented.



More information about the KSZ9021GN PHY chip can be found in the Micrel Datasheet, as well as application notes available on the manufacturer's website.

The Table below shows the pin interconnects between the FPGA and the Gigabit Ethernet PHY.

Connections between FPGA and Gigabit Ethernet PHY

Signal Name	FPGA Pin	Description	I/O Standard
ETH_GTX_CLK	PIN_T9	GMII Transmit Reference Clock	2.5V
ETH_TX_CLK	PIN_V10	MII Transmit Reference Clock	2.5V
ETH_TX_DV	PIN_AA8	GMII and MII transmit enable	2.5V
ETH_TX_D[7]	PIN_AB8	GMII transmit Data	2.5V
ETH_TX_D[6]	PIN_AG6	GMII transmit Data	2.5V
ETH_TX_D[5]	PIN_AG7	GMII transmit Data	2.5V
ETH_TX_D[4]	PIN_AG8	GMII transmit Data	2.5V
ETH_TX_D[3]	PIN_AH4	GMII and MII transmit Data	2.5V
ETH_TX_D[2]	PIN_AH5	GMII and MII transmit Data	2.5V
ETH_TX_D[1]	PIN_AH6	GMII and MII transmit Data	2.5V
ETH_TX_D[0]	PIN_AH7	GMII and MII transmit Data	2.5V
ETH_TX_ERR	PIN_AA9	GMII and MII transmit Error	2.5V
ETH_RX_CLK	PIN_AA11	GMII and MII Receive Reference Clock	2.5V
ETH_RX_DV	PIN_T11	GMII and MII receive Data Valid	2.5V
ETH_RX_D[7]	PIN_AF6	GMII receive Data	2.5V
ETH_RX_D[6]	PIN_AF7	GMII receive Data	2.5V
ETH_RX_D[5]	PIN_AD9	GMII receive Data	2.5V

ETH_RX_D[4]	PIN_Y10	GMII receive Data	2.5V
ETH_RX_D[3]	PIN_W10	GMII and MII receive Data	2.5V
ETH_RX_D[2]	PIN_V9	GMII and MII receive Data	2.5V
ETH_RX_D[1]	PIN_U12	GMII and MII receive Data	2.5V
ETH_RX_D[0]	PIN_U11	GMII and MII receive Data	2.5V
ETH_RX_ERR	PIN_R10	GMII and MII Receive Error	2.5V
ETH_RST_N	PIN_U9	Ethernet Transceiver Reset (active low)	2.5V
ETH_MDC	PIN_AC9	Management Data	2.5V
ETH_MDIO	PIN_AB9	Management Data Clock Reference	2.5V
ETH_CRS	PIN_AF8	GMII and MII Carrier Sense	2.5V
ETH_COL	PIN_T10	GMII and MII Collision Detected signal	2.5V
ETH_INT_N	PIN_V11	Ethernet Transceiver Interrupt (active low)	2.5V

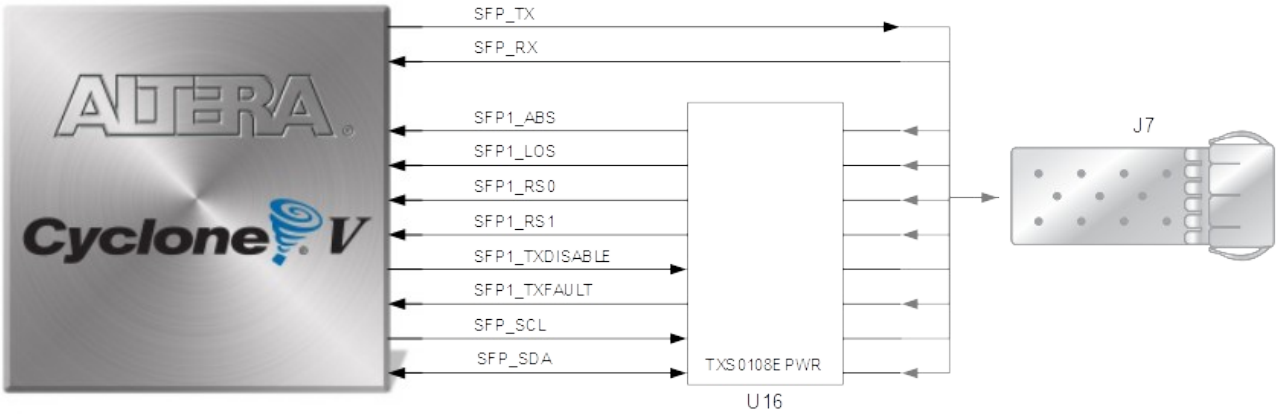
There are **two LEDs**, which represent the state of the Ethernet link negotiated by the PHY. The LED control signals from the PHY are connected to the LEDs on the RJ45 connector. The state and definition of the two LEDs are listed below.

State and Definition of LED Mode Pins

LED (State)		LED (Definition)		Link /Activity
LED2 (left)	LED1 (right)	LED2 (left)	LED1 (right)	
H	H	OFF	OFF	Link off
L	H	ON	OFF	1000 Link / No Activity
Toggle	H	Blinking	OFF	1000 Link / Activity (RX, TX)
H	L	OFF	ON	100 Link / No Activity
H	Toggle	OFF	Blinking	100 Link / Activity (RX, TX)
L	L	ON	ON	10 Link/ No Activity
Toggle	Toggle	Blinking	Blinking	10 Link / Activity (RX, TX)

SFP module

AVDB includes a standard SFP cage connected to Cyclone V GT Transceivers (SerDes).



The FPGA used on AVDB includes transceivers capable of supporting up to 6.144 Gbps.

As a consequence, you can use a wide range of SFP modules including but not limited to :

- RJ45 Ethernet
- Fiber Ethernet (2.5Gb/s)
- Optical SDI SFP
- 3G SDI modules
- HDMI/DVI modules
- etc

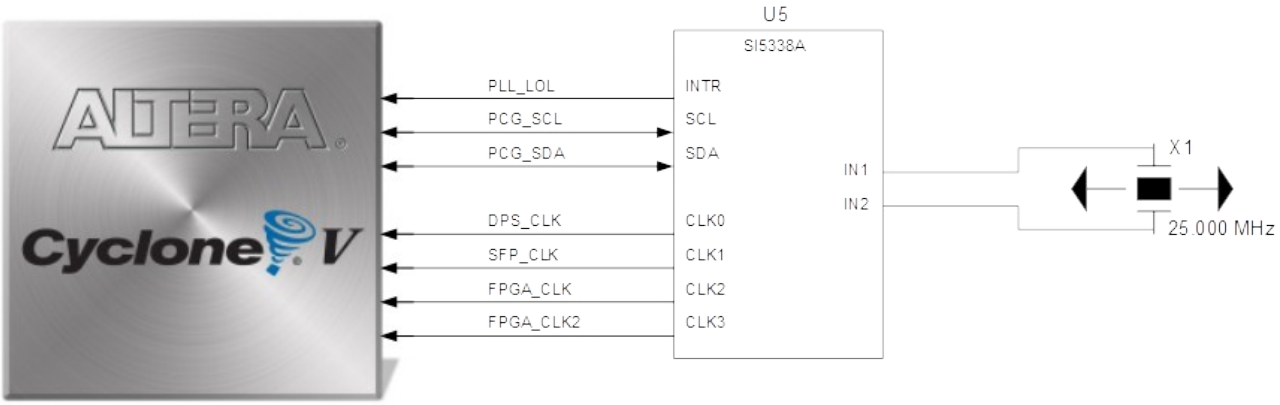
You can visit the Embrionix Web site to find a lot of such video SFP modules.

Connections between FPGA and SFP module

Signal Name	FPGA Pin	Description	I/O Standard
SFP_CLK	PIN_R8	Input clock of the FPGA Transceiver	1.5-V PCML
SFP_TX	PIN_Y4	Transmitter Data	1.5-V PCML
SFP_RX	PIN_AA2	Receiver data	1.5-V PCML
SFP1_ABS	PIN_J10	Module present indicator	1.5 V
SFP1_LOS	PIN_L9	Signal Loss Indicator	1.5 V
SFP1_RS0	PIN_K10	Rate Select0, controls the SFP Interface receiver	1.5 V
SFP1_RS1	PIN_N10	Rate Select1, controls the SFP Interface receiver	1.5 V
SFP1_TXDISABLE	PIN_F8	Turns off and disables the transmitter output	1.5 V
SFP1_TXFAULT	PIN_F6	Transmitter Fault	1.5 V
SFP_SCL	PIN_G8	Two wire serial interface clock line	1.5 V
SFP_SDA	PIN_G6	Two wire serial interface Data line	1.5 V

Programmable Clock Generator

AVDB includes one programmable oscillator from SiliconLabs ref Si5338A which is used to generate Reference Clock Signals for use by DisplayPort Transceivers, SFP Transceivers and DDR3 Hard Memory controllers. It receives a high stability oscillator clock at 25 MHz for reference.



For ease of use, this device comes already programmed at delivery, with a default configuration that is suitable to a lot of AVDB applications.

This configuration is summarized below:

```
#Input Mux = XoClk
#FDBK Input Frequency (MHz) = 25.000

#Output Clock 0
# Output Frequency (MHz) = 148.500

#Output Clock 1
# Output Frequency (MHz) = 125.000

#Output Clock 2
# Output Frequency (MHz) = 50.000

#Output Clock 3
# Output Frequency (MHz) = 50.000
```

However, this programmable clock generator remains completely programmable via an I2C Serial interface. You cannot however alter the *power up* configuration, but if you include a control block which activates when the FPGA starts, you can reprogram any desired configuration suitable for your specific application.

Note that **ALSE has designed a specific IP for this task.**

The interconnects between the FPGA and the Programmable oscillator are described next page.

Pin Assignment of the Programmable Clock Generator

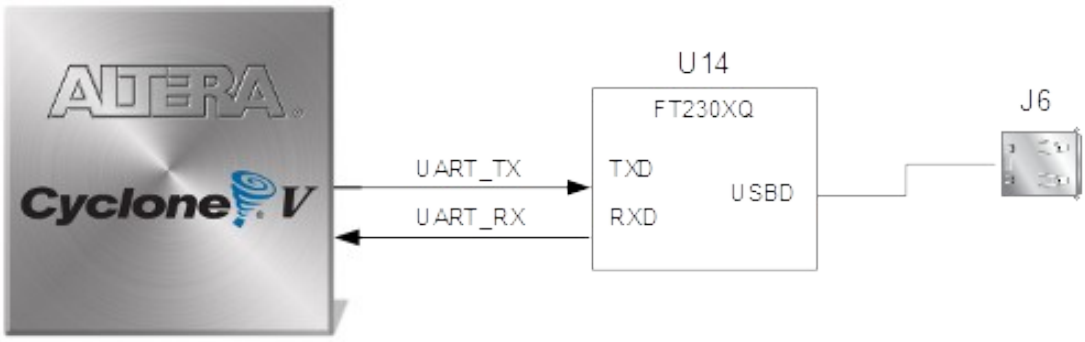
Signal Name	FPGA Pin	Description	I/O Standard
DPS_CLK_N	PIN_N7	DP Reference Clock (Transceiver input clock)	IO_STANDARD LVDS
DPS_CLK_P	PIN_P8	DP Reference Clock (Transceiver input clock)	IO_STANDARD LVDS
SFP_CLK_N	PIN_R7	SFP Reference Clock (Transceiver input clock)	1.5-V PCML
SFP_CLK_P	PIN_R8	SFP Reference Clock (Transceiver input clock)	1.5-V PCML
FPGA_CLK_N	PIN_J18	DDR3 HMC Reference Clock input (Top)	IO_STANDARD LVDS
FPGA_CLK_P	PIN_H19	DDR3 HMC Reference Clock input (Top)	IO_STANDARD LVDS
FPGA_CLK2_N	PIN_AA15	DDR3 HMC Reference Clock input (Bottom)	IO_STANDARD LVDS
FPGA_CLK2_P	PIN_Y15	DDR3 HMC Reference Clock input (Bottom)	IO_STANDARD LVDS
PLL_LOL	PIN_B28	Interrupt signal	3.3V
PCG_SCL	PIN_B29	I2C Serial Clock	3.3V
PCG_SDA	PIN_A29	I2C Serial Data	3.3V

UART over USB

AVDB includes one **UART over USB** interface connected for to the Cyclone V FPGA.

This interface doesn't support H/W flow control signals. The physical interface is implemented by a UART-USB bridge from FTDI chip ref **FT230XQ** and is available through a USB Mini-B connector **J6**.

More information about the chip, as well as miscellaneous drivers are available on the manufacturer's website.



Pin Assignment of UART over USB Interface

Signal Name	FPGA Pin	Description	I/O Standard
UART_TX	PIN_Y27	UART Transmit FPGA to FTDI	3.3V
UART_RX	PIN_W27	UART Receive from FTDI to FPGA	3.3V

Chapter 4
Reference Designs

1 Introduction

This chapter provides examples of advanced designs implemented in RTL or using Qsys and running AVDB (Advanced Video Development Board).

These Reference Designs cover a lot Video features, such as Video interfaces, DDR3 memory, and audio. All the associated bitstreams files are available on demand. Contact ALSE.

The “HDMI Bypass Reference Design” is available as a project for registered users. This permits to use it as a base for customized projects, while testing the various Intellectual Property blocks provided by ALSE (or Intel-FPGA) in OpenCore Plus mode.

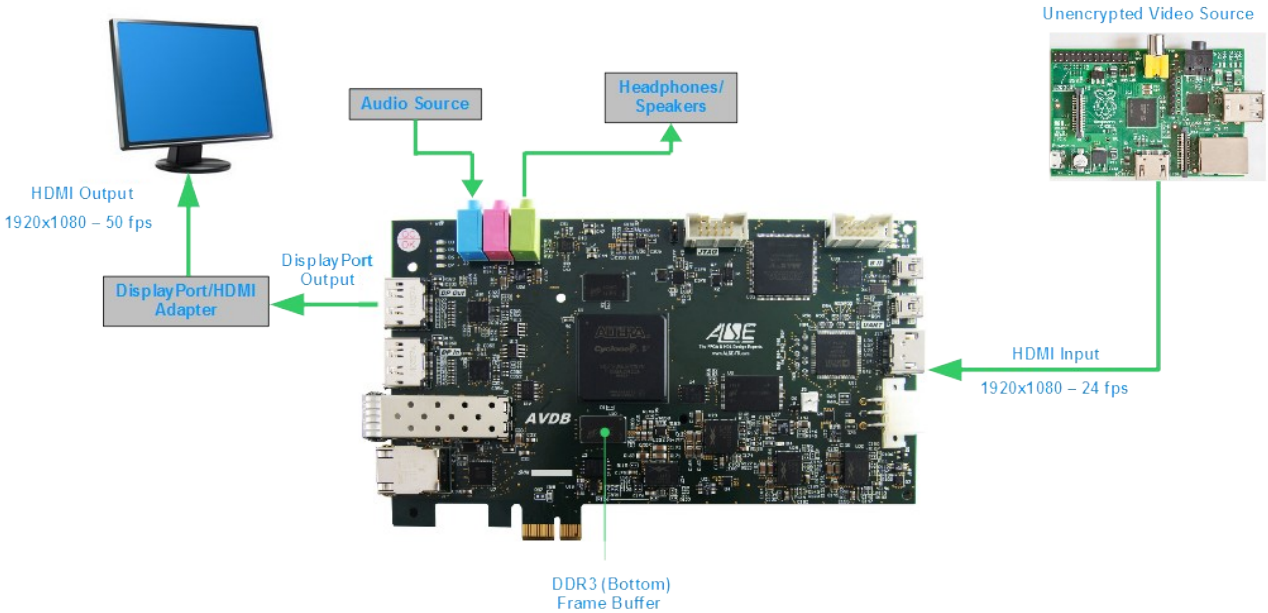
2 HDMI Bypass Ref Design

This demonstration turns AVDB into a “TV player” that sends video and audio to an HDMI TV or monitor. The video (& audio) comes from an HDMI non-encrypted video source player connected to AVDB's HDMI input port. The TV or monitor HDMI is attached to AVDB's DisplayPort connector using an HDMI adapter. The Audio Codec located on AVDB does play the HDMI source input audio and captures an audio source (if present) and plays it on the TV or Monitor.

Design features :

- Video input : non encrypted HDMI input – 1920x1080 @ 24 Fps – RGB on 8 bits (we typically use a Raspberry Pi)
- Video Output : HDMI through an adapter – 1920x1080 @ 50 Fps – RGB on 8 bits
- Audio Output : Line Output from the Codec – I2S Mode with Sampling rate @ 48 KHz
- Audio Input : Line input from the Codec – I2S Mode with Sampling rate @ 48 Khz

The Figure below shows the block diagram of the design.



The whole design fits in approximately 6,000 ALMs and has been built using Qys.

Once the FPGA is programmed :

- Video : a test picture with ALSE's logo should be displayed during a few seconds and then the Display screen should show the video read from the Video source.
- Audio : the audio input from the Video source should be available through the Display Screen speakers and also on the Board Headphone Speakers. If an audio source is connected to the Codec line input connector of the Board, it will replace the HDMI audio stream and be sent to the Video Display (and to the headphone/speakers).

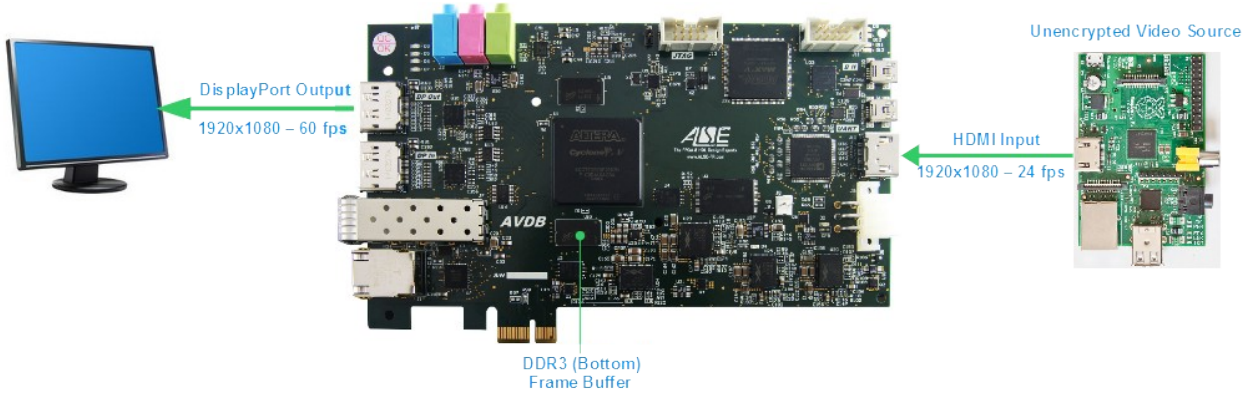
3 HDMI to DisplayPort Ref Design

This demonstration is a variant of the HDMI Bypass demo, with the Video output stream being displayed using the **DisplayPort** Video Standard.

Design features :

- Video input : non encrypted HDMI input – 1920x1080 @ 24 Fps – RGB on 8 bits (we typically use a raspberry Pi)
- Video Output : DisplayPort – 1920x1080 @ 60 Fps – RGB on 8 bits

The Figure below shows the block diagram of the design.



Once the FPGA is programmed :

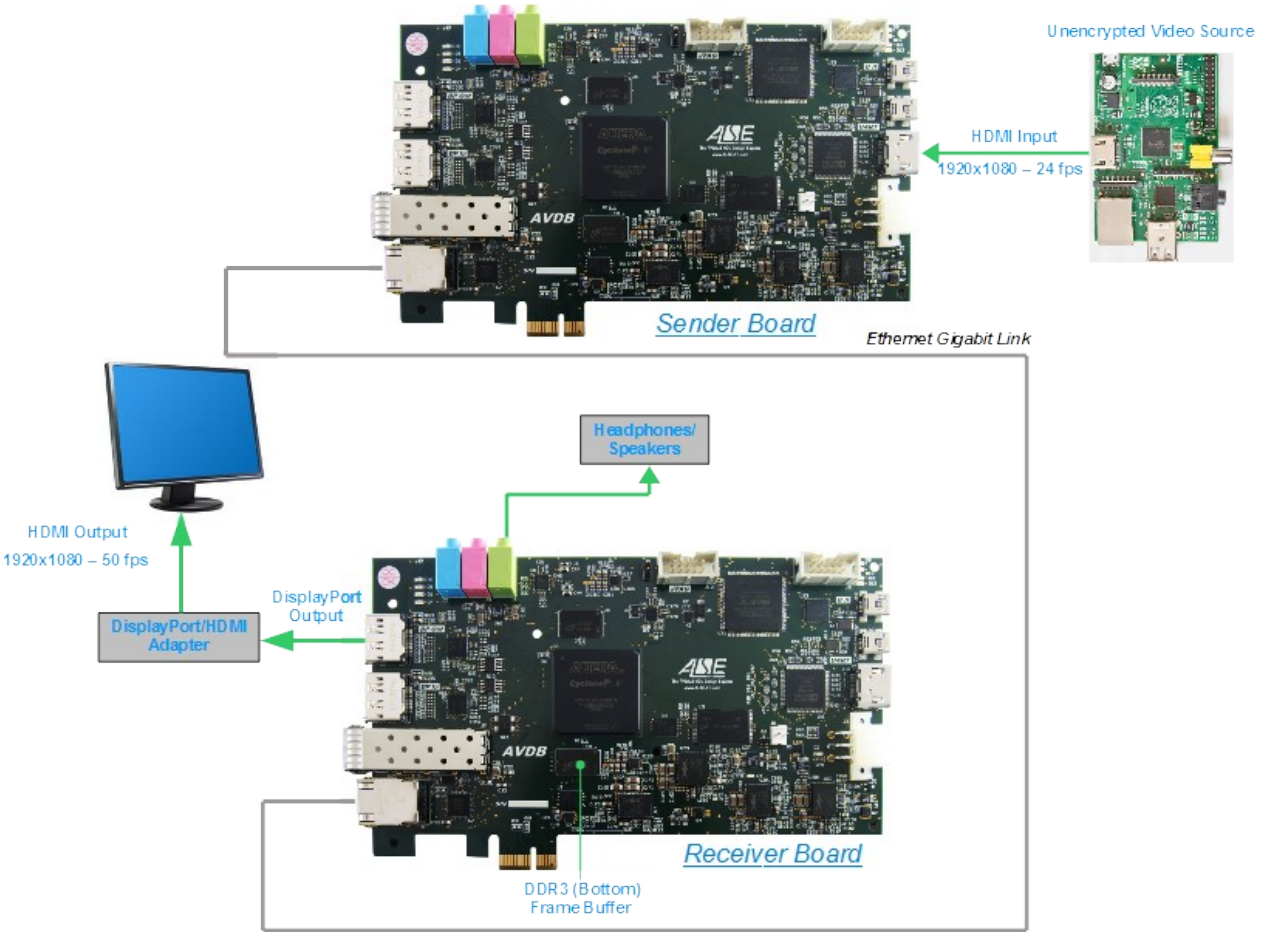
- Video : a test picture with ALSE's icon should be displayed during a few seconds and then the Display screen should show the video read from the Video source.

4 Video Streaming over Ethernet (4:2:0 no compression)

In this demonstration, a Full HD video stream (reduced to 4:2:0) is transmitted over Gigabit Ethernet through the **GEDEK** IP from ALSE (Gigabit Data Exchange Kit). This demo illustrates the efficiency and throughput of the GEDEK IP: the streamed data occupies close to 90% of the theoretical Gigabit bandwidth.

Two AVDB Kits are required for this demo :

- AVDB 1 (Sender) : receives a Full HD Video Stream on the HDMI Input Interface and transmits the video contents over Ethernet to the second AVDB.
- AVDB 2 (Receiver) : receives the Video Stream over Ethernet and displays it on the DisplayPort/HDMI Interface through a Frame Buffer (allowing Frame rate control).



NB : Please note that this setup also exists :

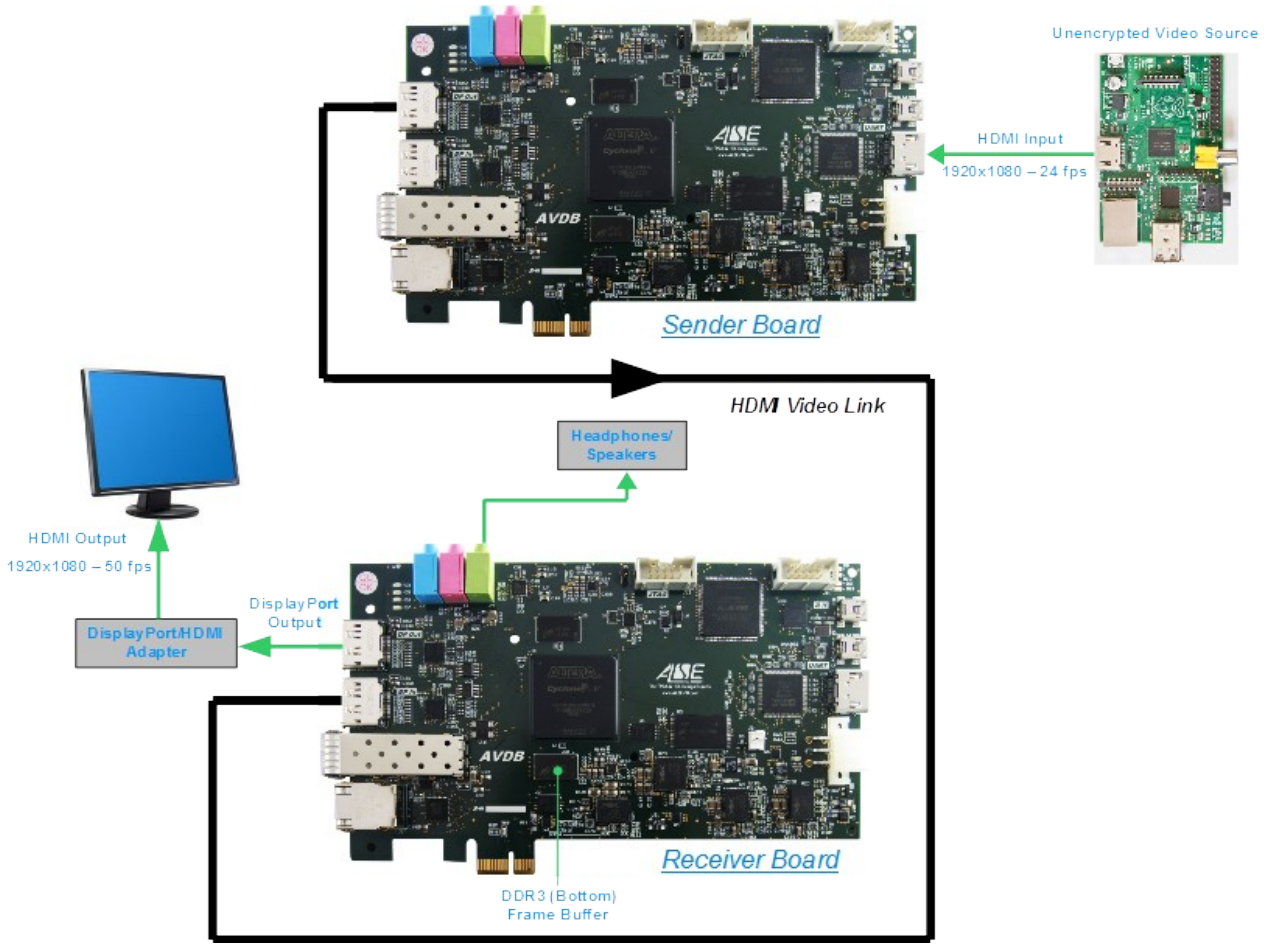
- With a JPEG encoder at input and a JPEG decoder in the receiver in order to reduce the Ethernet bandwidth used over the network.
- With the NEW **Wavelet Encoder/Decoder IPs developed by ALSE.** This new Codec provides extremely high quality and a very good compression rate.

5 Video Streaming over HDMI

This demonstration is a variant of the Video Streaming demo over Ethernet except that in this case the video stream is transmitted from one board to the other through the DisplayPort/HDMI Link.

This setup has been developed on AVDB to test and demonstrate ALSE's **HDMI receiver IP**.

Just like the Ethernet streaming demo video, the audio is also transported from the source to the TV or monitor through all IPs.

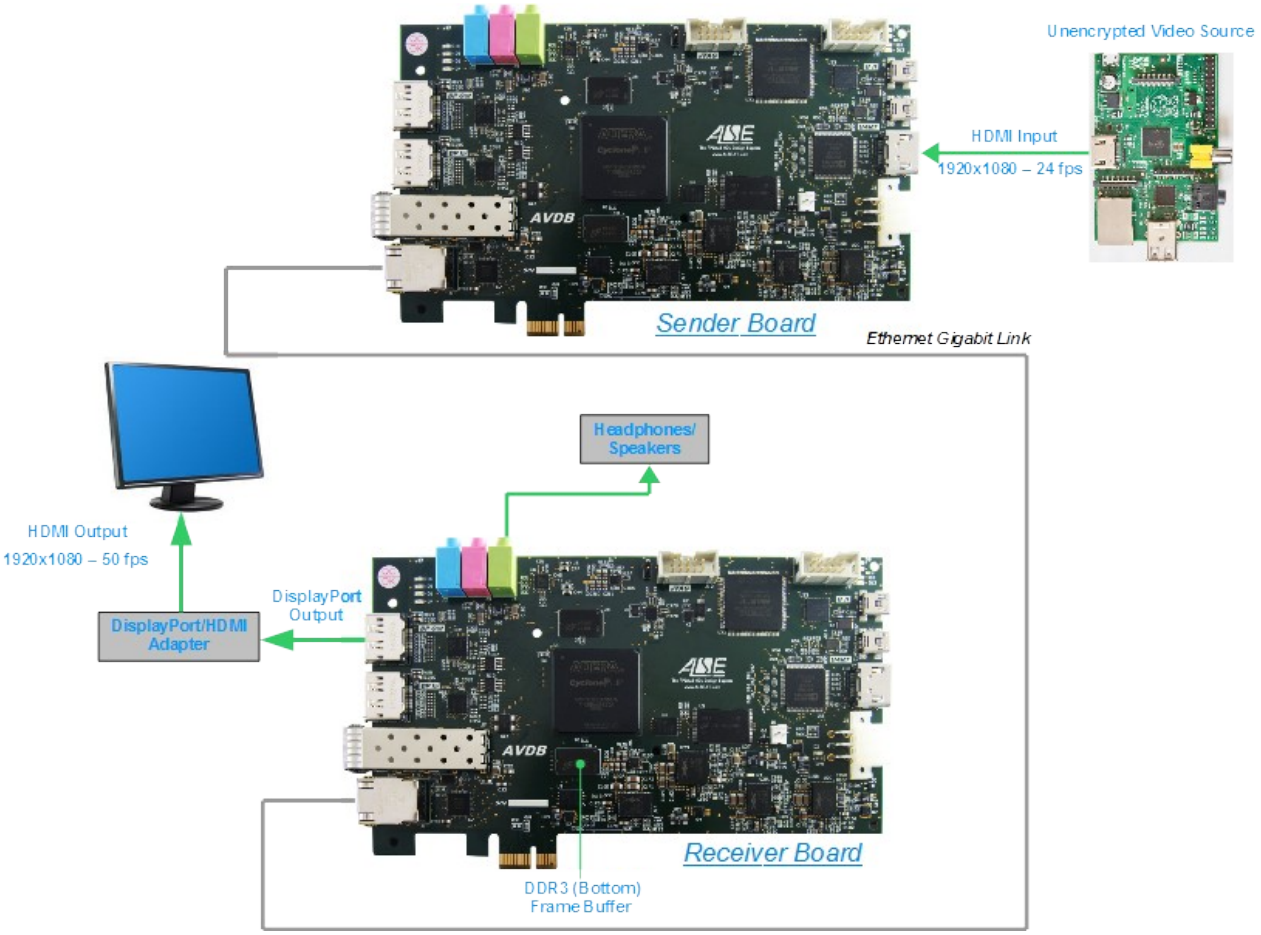


6 Video Streaming over Ethernet with wavelet compression - decompression

In this demonstration, a Full HD video stream (HDMI input) is received and compressed using ALSE's new wavelet Coder. The compression rate is high (close to 30) so the transmission over Gigabit Ethernet through the **GEDEK** IP occupies less than 8% of the theoretical GB bandwidth, thus allowing the use of Fast (100M Ethernet), or allowing to stream more than 10 HD videos in parallel ! This new wavelet Codec also provides high quality video.

Two AVDB Kits are required for this demo :

- AVDB 1 (Sender) : receives a Full HD Video Stream on the HDMI Input Interface and transmits the video contents over Ethernet to the second AVDB.
- AVDB 2 (Receiver) : receives the Video Stream over Ethernet and displays it on the DisplayPort/HDMI Interface through a Frame Buffer (allowing Frame rate control).



7 Video Streaming over Ethernet with intoPIX' TICO compression - decompression

In this demonstration, a **Full HD 60p** video stream (HDMI input) is received and compressed using i**ntoPIX' new TICO Codec**.

The compression rate is limited (~4) but the result after decompression is **visually lossless**.

The transmission over Gigabit Ethernet through the **GEDEK** IP occupies about 93% of the the theoretical GB bandwidth.

Two AVDB Kits are required for this demo :

- AVDB 1 (Sender) : receives a Full HD Video Stream on the HDMI Input Interface and transmits the video contents over Ethernet to the second AVDB.
- AVDB 2 (Receiver) : receives the Video Stream over Ethernet and displays it on the DisplayPort/HDMI Interface through a Frame Buffer (allowing Frame rate control).

You can get the bitstreams for free by asking ALSE or IntoPix.

If you are interested by this Codec, please contact i**ntoPIX**.

8 Video Training Course Lab4

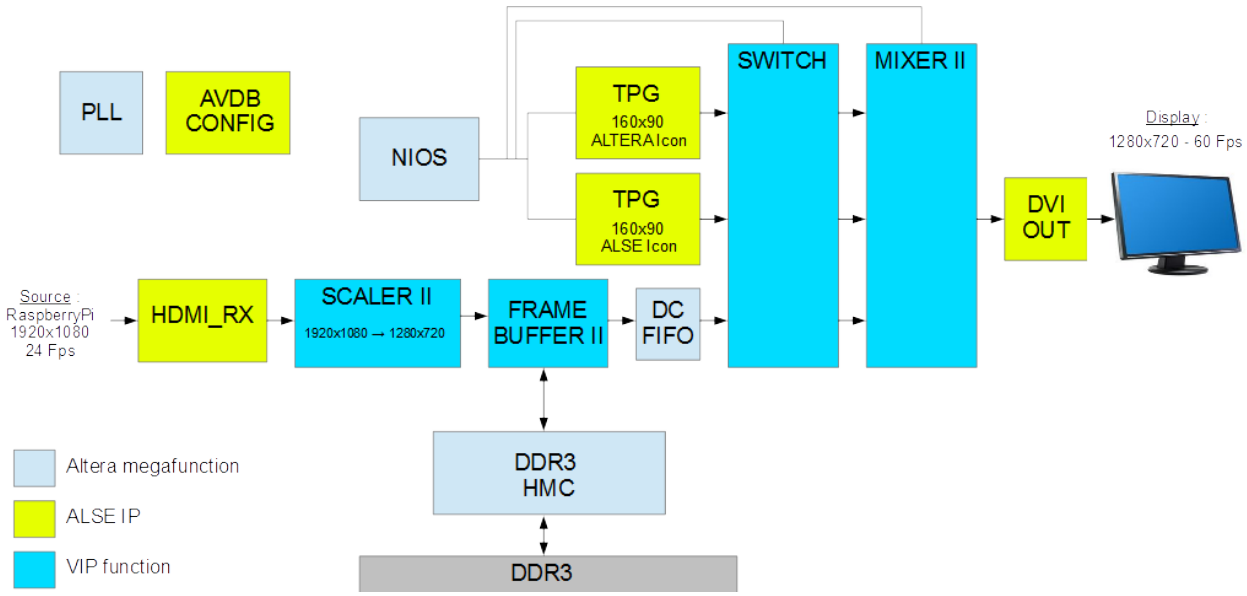
The following design uses several functions of the Video and Image Processing suite proposed by Intel-FPGA, as well as ALSE IPs.

IPs from the VIP suite :

- Scaler II
- Frame Buffer II
- Switch
- Mixer II

ALSE IPs :

- AVDB Configuration block
- Test Pattern Generator customized to generate Altera & ALSE logo.
- HDMI Receiver
- DVI Output



Video Input :

- HDMI mode – 1920x1080 – RGB – 24 Fps from the Raspberry Pi

Video Output :

- DVI Mode – 1280x720 – RGB – 60 Fps

Design description :
The NIOS embedded in the design is used to display every second the Altera & ALSE logos and to move them all around the display screen.

9 PCIexpress

We have developed several IPs using PCI-Express on Cyclone V for customers and we have tested them using AVDB (Gen2x1).

If you are interested in using PCI-Express, please contact ALSE.

10 ALSE Intellectual Property Blocks

In our Reference Designs, we use the following ALSE I.Ps :

- ALSE HDMI input through ADV7619
- ALSE HDMI in (with I2S audio out)
- ALSE HDMI out (with I2S input)
- GEDEK (Gigabit Ethernet hardware communication stack used for Video streaming)
- I2S Codec (SSM2603) Interface
- I2C controller
- NAND Flash Memory Controller with BCH ECC
- Quad-SPI Controller
- JPEG video compression engine
- JPEG video decompression engine
- Wavelet video compression engine
- Wavelet video decompression engine
- Frame Buffer (with rate changing)
- Video Test Pattern Generator
- Upscale / Downscale resizers
- PCI-Express

11 Intel-FPGA VIP suite

We have built many Labs that can be used in a training course about Video on FPGAs.

These Labs demonstrate the use of VIP suite blocks and are designed and tested on AVDB.

-=oOo=-

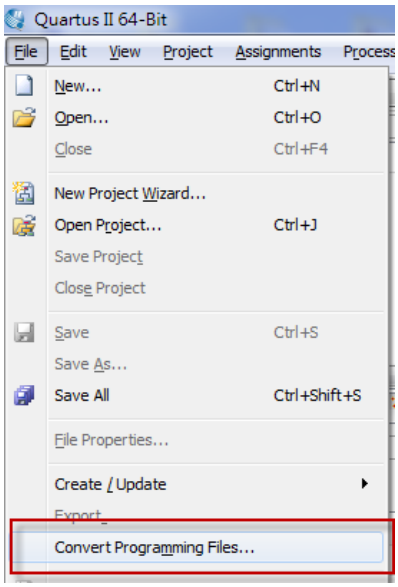
Chapter 5
Programming the Configuration Flash

1 Introduction

This chapter describes how to program the Quad Serial configuration (EPCQ) Flash using the Serial Flash Loader (SFL) function via the JTAG interface. Users can program EPCQ devices with a JTAG indirect configuration (.jic) file, which is converted from a user-specified SRAM object file (.sof) in Quartus. The .sof file is generated after the project compilation is successful. The steps of

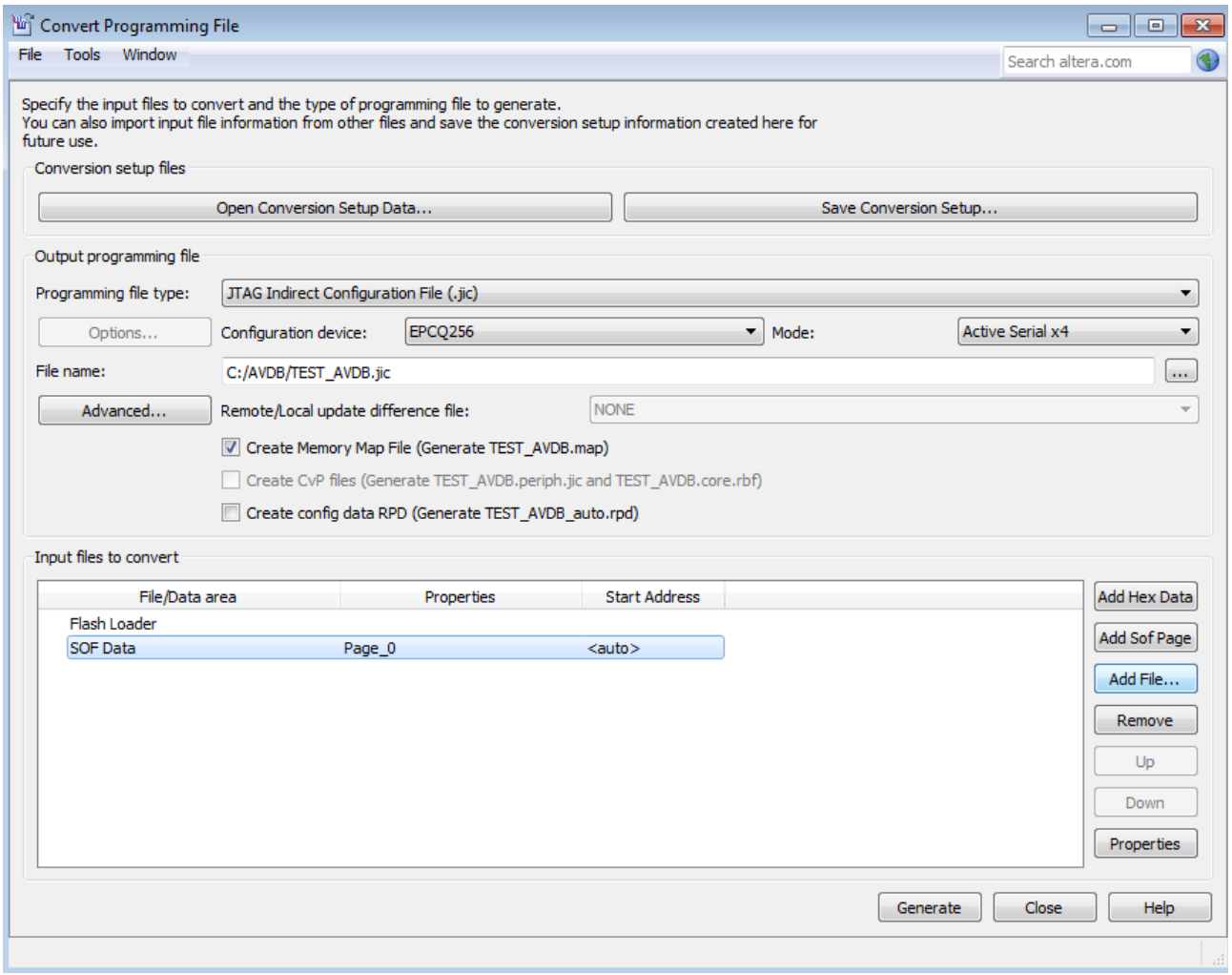
2 Converting the .SOF File into a .JIC File

1. Choose **Convert Programming Files** from the File menu of Quartus II, as shown below :

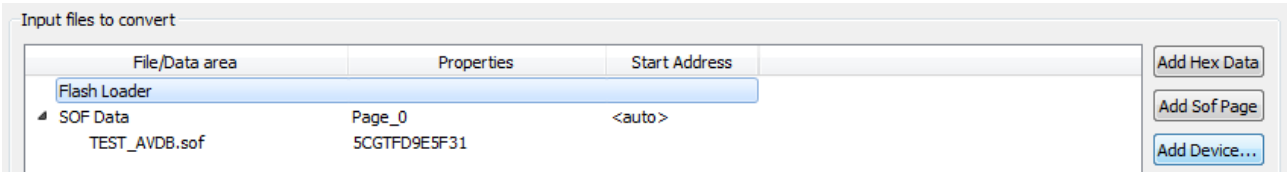


2. Select **JTAG Indirect Configuration File (.jic)** from the **Programming file type** field in the dialog of Convert Programming Files.
3. Choose **EPCQ256** from the **Configuration device** field.
4. Choose **Active Serial x4** from the **Model** filed.
5. Browse to the target directory from the **File name** field and specify the name of output file.

6. Click on the **SOF data** in the section of **Input files to convert**, as shown here below :

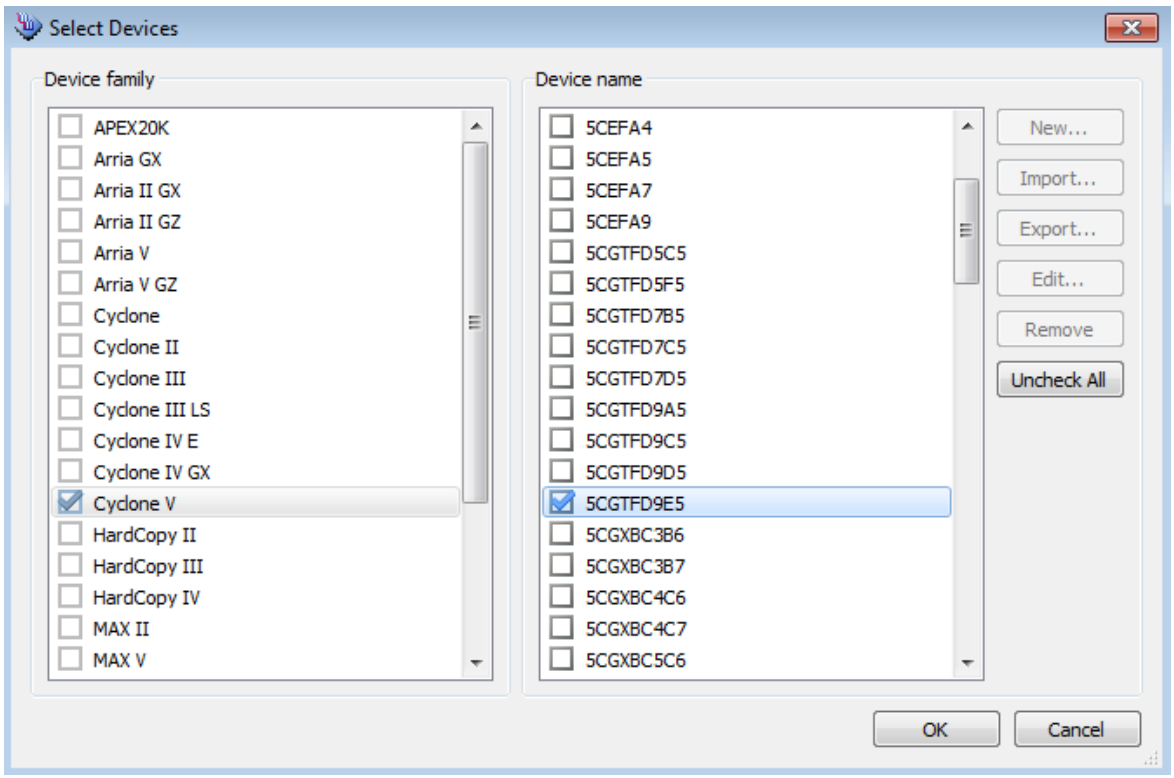


7. Click **Add File**.
8. Select the .sof to be converted to a .jic file from the Open File dialog.
9. Click **Open**.
10. Click on the **Flash Loader** and click **Add Device**, as shown below

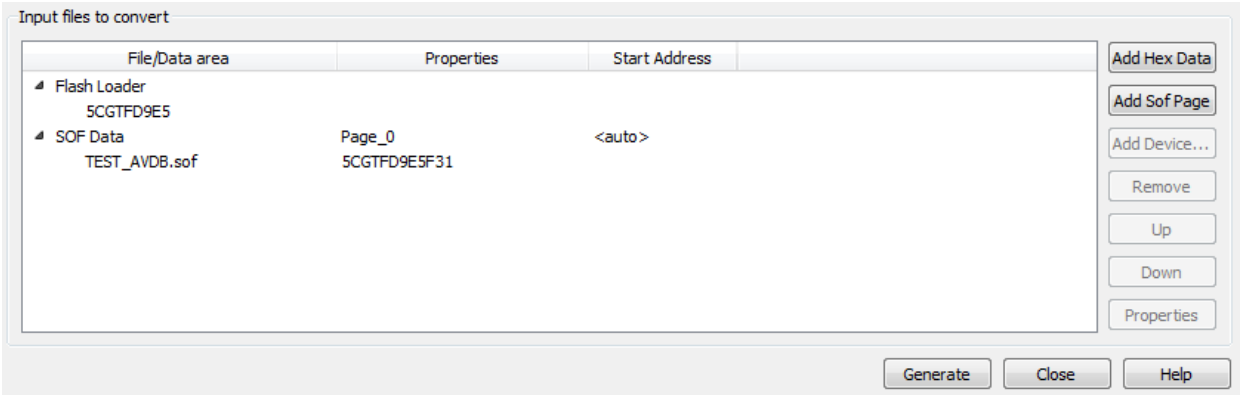


11. Click **OK** and the **Select Devices** page will appear.

12. Select the targeted FPGA to be programed into the EPCQ, as shown below :

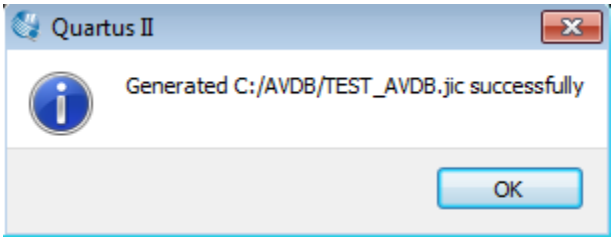


13. Click **OK** and the **Convert Programming Files** page will appear, as shown below :



14. Click **Generate**.

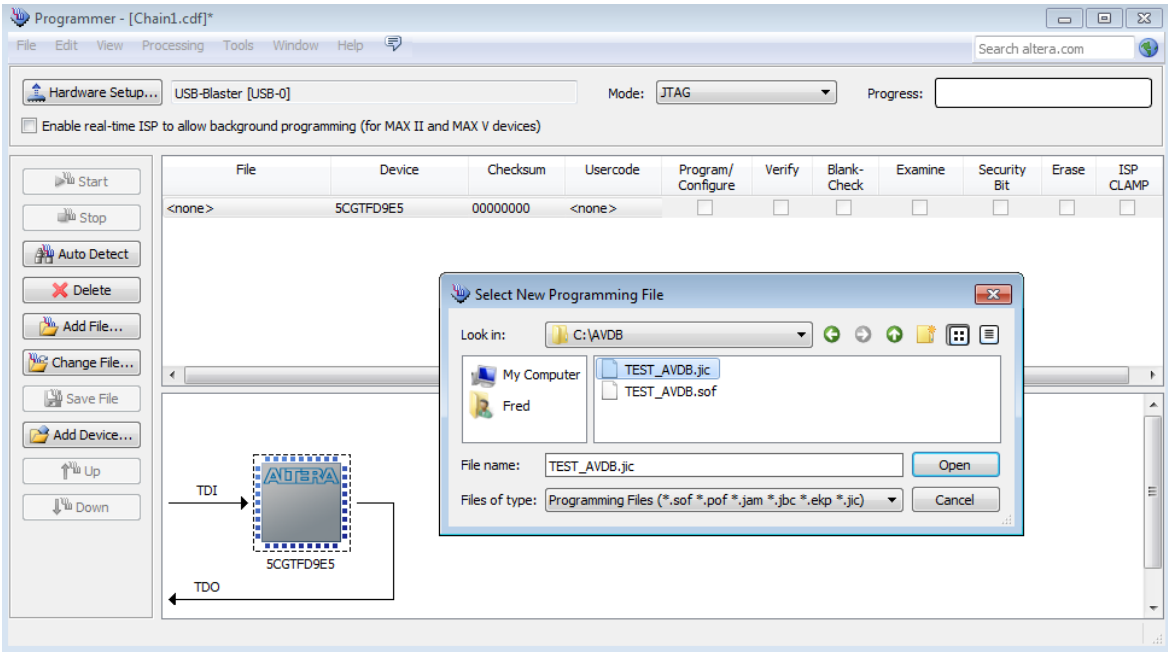
15. At the end of the conversion process the following message should appear :



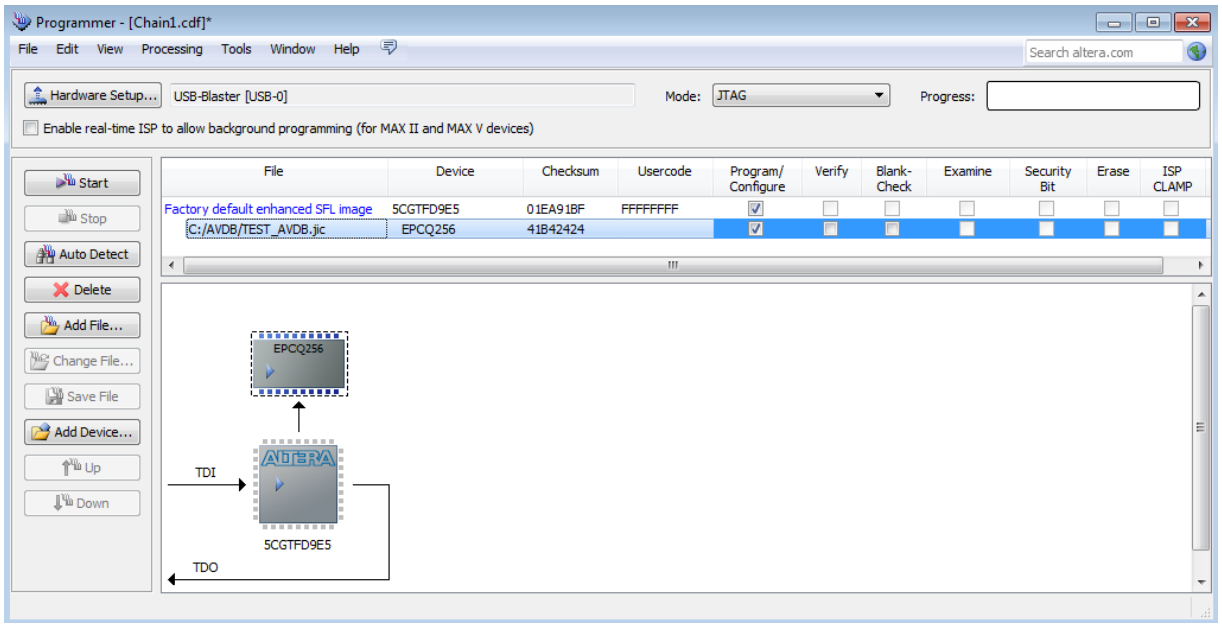
3 Programming the Configuration Flash

When the conversion of SOF file format to JIC file is complete, you can program the Serial Flash with the steps below.

1. Choose **Programmer** from the Tools menu and the **Chain.cdf** window will appear.
2. Click **Auto Detect** and then select the correct device : 5CGTFD9E5. The FPGA device should be detected**.**
3. Double click the “none” File and the **Select New Programming File** page will appear. Select the .jic file to be programmed:



4. Program the EPCQ device by clicking the corresponding **Program/Configure** box. A factory default SFL image will be loaded, as shown here below :



5. Click **Start** to program the EPCQ device. (note that you can also **Erase the Flash**).

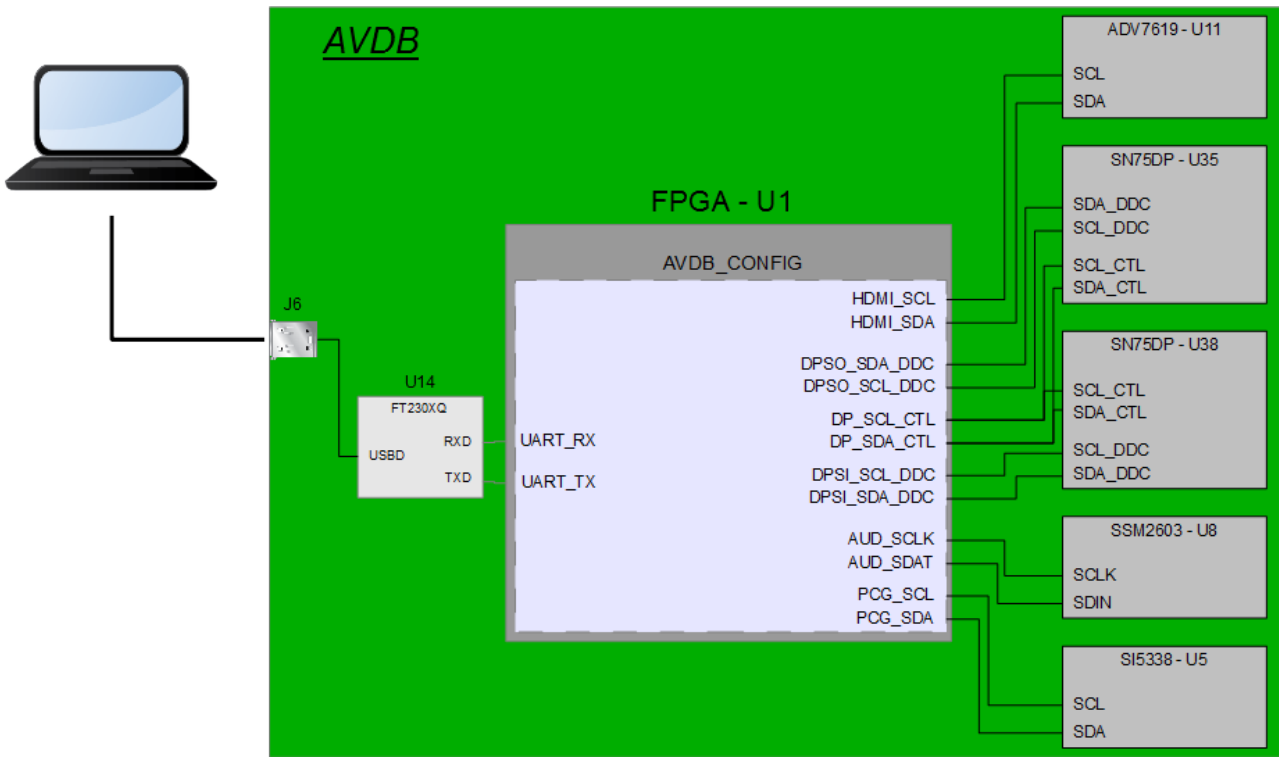
Chapter 6
AVDB Peripherals Configuration IP

1 Introduction

As we have seen, AVDB includes many peripheral devices that can be configured through I^2C links.

In several reference designs, we use a Configuration IP named AVDB_CONFIG that automatically configures these devices when the FPGA starts running.

It is also possible to change dynamically the settings of the devices through RS232 as represented below.



2 Peripherals Memory Map

A 32 bits address defines which device is addressed.

The memory map is represented in the Table below:

I2C Device	FPGA Internal Address on 32 bits
HDMI – ADV7619 (U11)	0x01XX00XX
• HDMI – IO	0x019800XX
• HDMI – CP	0x014400XX
• HDMI – REP	0x016400XX
• HDMI – DPLL	0x014C00XX
• HDMI – HDMI	0x016800XX
DisplayPort – SN75DP (U38) – CTL	0x020000XX
DisplayPort – SN75DP (U35) – CTL	0x040000XX
DisplayPort – SN75DP (U38) – DDC	0x080000XX
DisplayPort – SN75DP (U35) – DDC	0x100000XX
Audio Codec – SSM2603 (U8)	0x200000XX
Programmable Clock Generator – SI5338 (U5)	0x400000XX

3 Using the Peripherals Configuration IP

The I^2C devices can be selected in this block with a simple protocol based on ASCII characters.

The RS232 interface in the block AVDB_CONFIG is configured as follows :

- Baud rate : 115200 bauds
- 8 x Data bits
- 1 x Stop bit
- No parity used
- No Flow control used

The RS232 protocol used by this block is the following :

- Write access : **W XXXXXXXX XXXX <Enter>**
  - command character : either W or w,
  - spaces are optional
  - address on 32 bits in hexadecimal (must be 8 characters)
  - data on 16 bits in hexadecimal (must be 4 characters)
  - terminated by Enter (Carriage Return)
- Read Access : **R XXXXXXXX <Enter>**
  - command character : either R or r
  - space is optional
  - address on 32 bits in hexadecimal (must be 8 characters)
  - terminated by Enter (Carriage Return)

4 Example of use: Codec

For example to read the back the Digital Audio I/F of the Codec (register located at the address 0x07 in the Codec device) the following characters must be sent on the serial line:

> R20000007

The FPGA will then perform the read operation of the “Digital Audio I/F” register located in the Codec and will then return on the serial line the 16 bits value:

> 0002

The value returned corresponds to the value configured by the FPGA at Power Up.

For many Reference Designs provided, this is the following configuration:

- BCLK not inverted
- Enable slave mode (default)
- Output left and right channel data as normal (default)
- Normal PBLRC and RECLRC (default)
- 16 bits
- I^2S mode (default)

In order to use Sample on 24 bits instead of 16 bits for example through the Codec interface the content of the register “Digital Audio I/F” can be modified by sending the following characters on the serial line:

> W 20000007 000A