Technical Reference for Altium's LatticeECP™ Daughter Board DB32

Summary

This reference document provides detailed information on Altium's LatticeECP daughter board DB32, including the physical FPGA device it offers and any additional resources available to an FPGA design targeting that device.

Technical Reference
TR0151 (v2.0) April 28, 2008

The target FPGA devices to which a design can be downloaded reside on separate satellite boards, referred to as daughter boards. These boards plug-in to Altium's Desktop NanoBoard NB2DSK01. By keeping each programmable device on its own daughter board, engineers can easily change the target project architecture, while keeping the NB2DSK01 a truly FPGA vendor-independent, reconfigurable hardware platform.

Altium's LatticeECP™ daughter board DB32 provides an LFECP33E-3FN672C device, as well as a range of on-board memories available for use by a design running within that device.

Note: All daughter boards available from Altium may be plugged into either the Desktop NanoBoard NB2DSK01 or the NanoBoard-NB1.

For information on the Desktop NanoBoard NB2DSK01, refer to the document *TR0143 Technical Reference Manual for Altium's Desktop NanoBoard NB2DSK01*.

For information on the complete range of daughter boards currently available, and additional documentation specific to each, go to *www.altium.com/nanoboard/resources*.

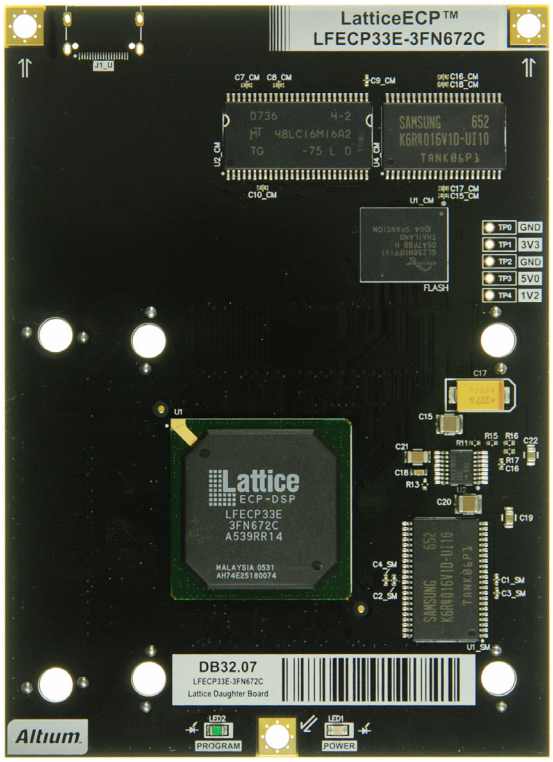


Figure 1. Altium's LatticeECP daughter board DB32.

Key features

- LatticeECP FPGA (LFECP33E-3FN672C)
- On-board memories available for use by FPGA design:
  - 256K x 32-bit common-bus SRAM (1MByte)
  - 16M x 32-bit common-bus SDRAM (64MByte)
  - 16M x 16-bit common-bus Flash memory (32MByte)
  - Dual 256K x 16-bit independent SRAM (512KByte each)
- 1-Wire® memory device used to store board ID and related information
- Three 100-way connectors for attachment to NB2DSK01 motherboard. These connectors provide:
  - Interface to resources on the NB2DSK01 motherboard and plugged-in peripheral boards
  - SPI bus interface
  - I2C bus interface
  - 1-Wire bus interface
  - JTAG, power and additional control lines from the motherboard.

Functional overview

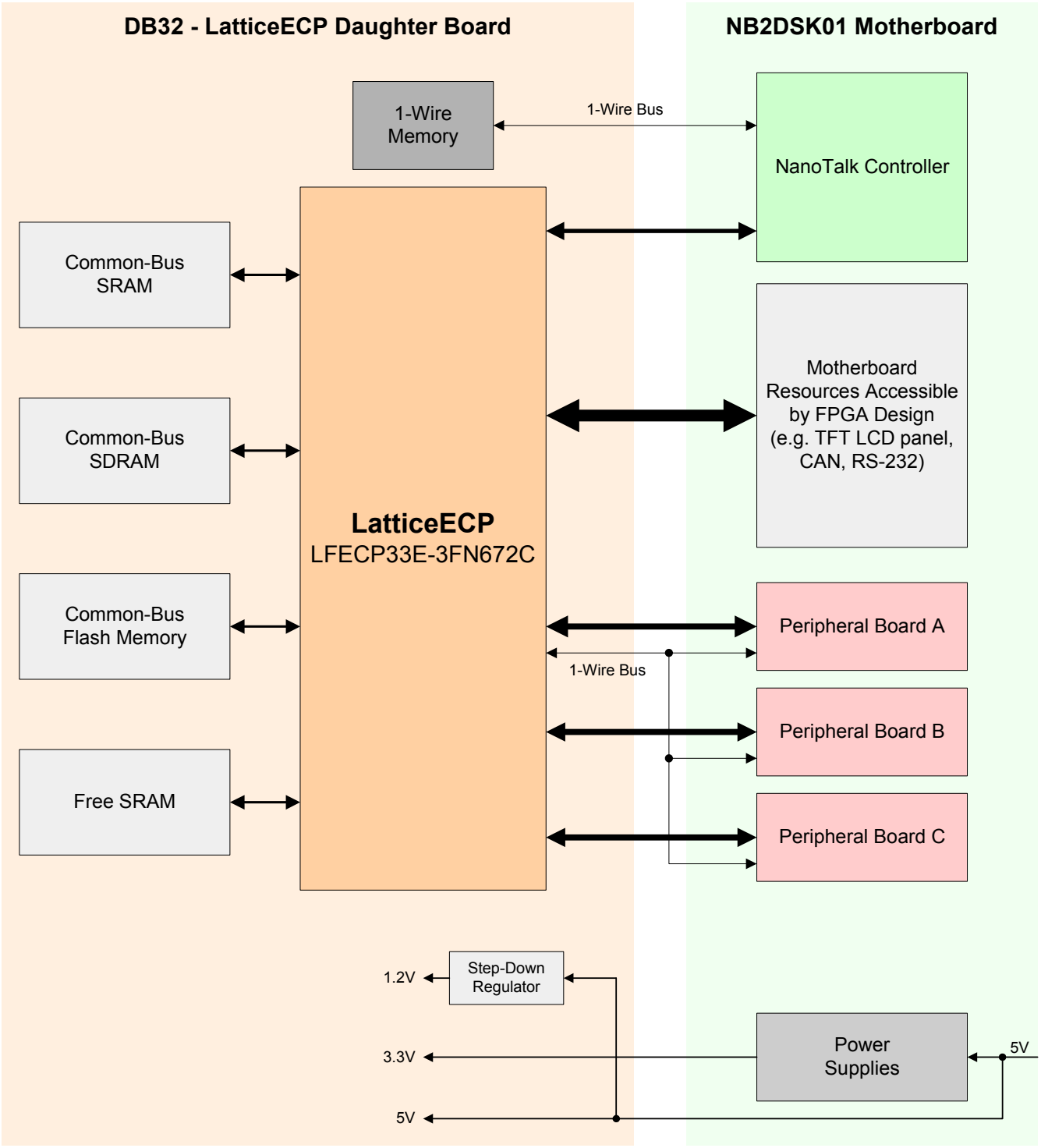


Figure 2. LatticeECP daughter board (DB32) block diagram.

3-connector daughter board

Daughter boards available with the Desktop NanoBoard NB2DSK01 and beyond each have three connectors (top, bottom and left). These are referred to as '3-connector' daughter boards. Those available previously for the NanoBoard-NB1 have two such connectors (top and bottom) and are referred to as '2-connector' daughter boards. The LatticeECP daughter board (DB32) is a 3-connector daughter board.

3-connector daughter boards also have three holes that align with threaded standoffs on the NB2DSK01. These can be used to affix the board securely to the motherboard.

Attachment to the NB2DSK01 motherboard

The daughter board is mounted onto the NB2DSK01 motherboard by plugging its three 100-way Male docking connectors into the motherboard's corresponding three 100-way Female docking connectors – designated `HDR_T1`, `HDR_L1` and `HDR_B1`. These are referred to as `'NANOCONNECT'` interfaces. The board can be securely fixed in place using the available threaded standoffs.

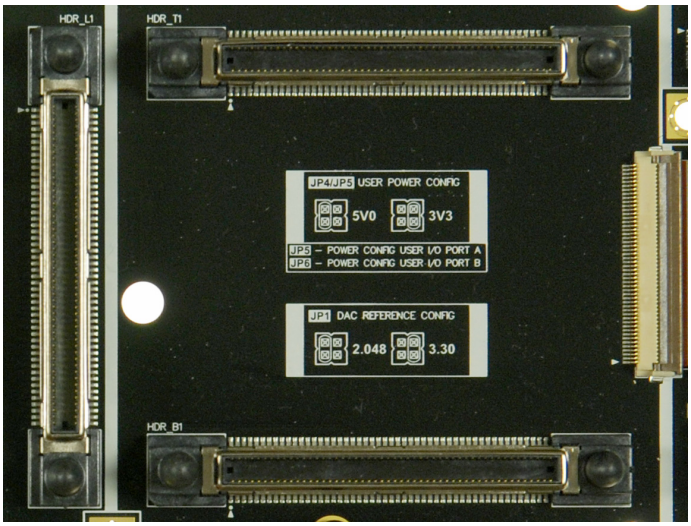


Figure 3. Daughter board connectors on the NB2DSK01 motherboard – the docking point for a satellite daughter board.

Accessing motherboard resources

The daughter board connectors on the motherboard map I/O resources on the NB2DSK01 directly to the pins of the daughter board FPGA device, as if that device were mounted directly on the motherboard.

For more information on the NB2DSK01 motherboard resources made available to a daughter board FPGA device, see the section *Resources accessible from an FPGA design*, in the document *TR0143 Technical Reference Manual for Altium's Desktop NanoBoard NB2DSK01*.

Accessing peripheral board resources

Resources on the plug-in peripheral boards are also made available to the daughter board FPGA device. The corresponding I/O pins from each peripheral board connector – 50 each – are wired directly to pins of the daughter board connectors on the motherboard:

- Peripheral Board A and B connector signals are wired to daughter board connector `HDR_L1`
- Peripheral Board C connector signals are wired to daughter board connector `HDR_T1`

These signals are subsequently wired to I/O pins of the daughter board's FPGA device. By keeping the signals generic – based on the peripheral board connector position and not on the resources – the peripheral boards can be attached to the NB2DSK01 in any of the three peripheral board positions.

Note: Some peripheral boards (e.g. PB01 – the Audio / Video Peripheral Board) are twice the width of other available peripheral boards and, as such, can only be connected to the NB2DSK01 using the `'PERIPHERAL BOARD A'` or `'PERIPHERAL BOARD C'` connectors.

For more information on peripheral boards, see the section *Peripheral Boards*, in the document *TR0143 Technical Reference Manual for Altium's Desktop NanoBoard NB2DSK01*. For information on the range of peripheral boards currently available and additional documentation specific to each, go to *www.altium.com/nanoboard/resources*.

Additional services

In addition to the user-available I/O on the motherboard and peripheral boards, the daughter board connectors provide pins for a series of other functions, including implementation of the NanoTalk communications protocol, power, and programming of the FPGA device. The following sections detail these additional signals, in relation to the daughter board connectors on the daughter board itself.

Hard JTAG signals

All daughter board devices that are JTAG-equipped are connected to signals `FPGA_TMS`, `FPGA_TCK`, `FPGA_TDI` and `FPGA_TDO`. This allows the NB2DSK01 and Altium Designer to address the daughter board hardware using the JTAG protocol.

Soft JTAG signals

Four FPGA I/O pins are reserved for JTAG signals that are utilized by the FPGA design. Altium Designer uses JTAG IP to communicate directly with the FPGA fabric, allowing applications to be debugged live. These signals (`NEXUS_TMS`, `NEXUS_TCK`, `NEXUS_TDI` and `NEXUS_TDO`) are derived in the NB2DSK01's NanoTalk Controller, which is implemented in a Xilinx Spartan-3 on the motherboard.

For more information on the JTAG communications, refer to the article *AR0130 PC to NanoBoard Communications*.

Daughter board identification signals

FPGA devices from different manufacturers and families require differing auto-configuration processes, so it is necessary for the NanoTalk Controller to be able to identify the family.

Board identification is achieved through use of a 1-Wire slave memory device located on the daughter board. This device carries more than enough storage capacity to hold information such as Board ID, physical Device ID, board revision, and so on. For more information, see the section *Board ID memory*, later in this document.

To satisfy NanoBoard-NB1 compatibility requirements, four signals (`FPGA_ID0..FPGA_ID3`) are hardwired on the daughter board to provide the required identification to the NanoTalk Controller. For the LatticeECP daughter board (DB32), `FPGA_ID[1..0]` are tied to 3.3V and `FPGA_ID[3..2]` are tied to `GND`, giving an `FPGA_ID` value of `3h`.

SPI bus interface

The NB2DSK01 SPI system involves a variety of SPI-compatible slave resources, located across the hardware system – on the NB2DSK01 motherboard itself and also on certain peripheral boards that plug in to the motherboard. These SPI resources are accessible by three distinct SPI masters:

- Altium Designer (via the parallel or USB connection)
- The firmware – more specifically a TSK3000A processor therein – loaded onto the motherboard's Spartan-3 FPGA device (the NanoTalk Controller)
- The design loaded onto the currently plugged-in daughter board FPGA device.

Providing the required SPI bus arbitration between the masters, and access to the SPI devices, is the NB2DSK01's SPI Controller. The Controller, which is part of the NanoBoard firmware, plays the role of multiplexer/router – determining which master has access to the SPI bus and which SPI slave device is selected for communications.

From an FPGA design perspective, the NB2DSK01's SPI Controller provides an SPI path from the daughter board to each of the SPI slave resources resident in the system.

Daughter board connector signals `SPI_DIN`, `SPI_DOUT`, `SPI_CLK`, `SPI_SEL` and `SPI_MODE` provide the required connectivity from the daughter board. During operation, the daughter board FPGA design communicates with the NB2DSK01's SPI Controller to establish a path between the design and a specific motherboard/peripheral board SPI device.

For detailed information on the Desktop NanoBoard's SPI communications system, refer to the document *AP0163 SPI Communications on the Desktop NanoBoard NB2DSK01*.

1-Wire® bus interface

A 1-Wire serial bus interface signal is provided (`ONE_WIRE_DB_PB`) which is connected through to each of the NB2DSK01's peripheral board connectors. This provides the ability to communicate directly from a processor in an FPGA design, with one or more slave 1-Wire compatible devices located across plugged-in peripheral boards (where such devices exist). As the 1-Wire bus is made available to all three peripheral board sites it allows development of a dedicated network of 1-Wire devices – a micro-LAN if you will.

Daughter board power signals

Daughter board connectors `HDR_T` and `HDR_B` provide two power supplies to the daughter board, as well as ground signals. The power supply voltages – sourced from the NB2DSK01 motherboard – are 5V and 3.3V.

Note: The maximum current available to the daughter board will depend upon the number and type of peripheral boards attached to the motherboard, as well as the power requirements of each individual rail. In any case, the total connector contact rating of 2.0A for both the 3.3V and 5V rails is not to be exceeded. The maximum return ground current for all rails should not exceed 3A.

Daughter board FPGA control signals

Daughter board connector `HDR_T` handles various signals between the NanoTalk Controller and the daughter board FPGA device, that are used to control that device. Such signals include detection of an FPGA device when a daughter board is plugged-in to the motherboard, as well as signals used to actually program the device with an FPGA design.

A word about changing daughter boards

When changing FPGA daughter boards, please take care not to damage the connectors that attach the daughter board to the NB2DSK01. The following procedure is recommended:

1. Ensure that the NB2DSK01's power is switched off.
2. If using a 3-connector daughter board, ensure that any screws affixing it to the motherboard are removed.
3. Grip the two sides of the daughter board between your thumb and fingers and gently pull the daughter board upwards. Gently rocking the daughter board from side to side can help loosen the connectors.
4. As the daughter board disengages, ensure that you keep the board parallel to the motherboard and pull it straight up until all connectors are fully disengaged.
5. Install a different daughter board by gently positioning it so that its connectors are aligned with the corresponding connectors on the motherboard – the posts of which slot through the holes in the daughter board.

   If using a 2-connector daughter board, place its locator 'legs' into the corresponding holes in the motherboard.

   Once located, press the daughter board firmly onto the motherboard. If using a 3-connector daughter board, such as the LatticeECP daughter board (DB32), you can secure it to the motherboard using the available threaded standoffs.
6. Switch the NB2DSK01's power on.

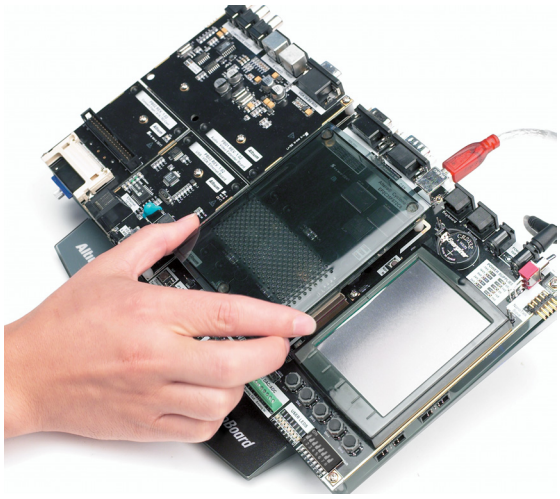


Figure 4. Switch daughter boards to change the target architecture.

The system software interrogates the NB2DSK01 at regular intervals to determine the FPGA device installed. If you change daughter boards, the system will automatically detect the change and show the correct device in the **Devices** view. When the **Devices** view is active you can force the system to poll the NB2DSK01 by pressing the **F5** key.

Daughter board resources

The resources on the LatticeECP daughter board (DB32) can be divided into the following key areas:

- General board resources – including connectors, power supplies and LEDs
- Target FPGA device – the primary element of the board, into which an FPGA design is programmed
- User resources – resources made available for use by the FPGA design itself, such as memory storage devices.

The following sections take a closer look at these three areas.

Note: For each of the following resources, reference is made to the corresponding sheet(s) of the daughter board schematics on which the circuitry of interest can be found. A pdf document of these schematics – *DB32 LatticeECP (LFECP33E-3FN672C) Daughter Board Schematics.pdf* – can be found at *www.altium.com/nanoboard/resources*.

General board resources

The following is a list of general resources found on the DB32.

Docking connectors

The DB32 is a 3-connector daughter board and, as such, has three 100-way Male connectors. These are used to connect the daughter board to the NB2DSK01 motherboard, which has corresponding 100-way Female connectors (`'NANOCONNECT'` interfaces).

The connectors on the daughter board are designated `HDR_T`, `HDR_L` and `HDR_B`.

Note: `HDR_T` and `HDR_B` connectors are wired such that the daughter board may be attached to either the Desktop NanoBoard NB1DSK01 or the NanoBoard-NB1.

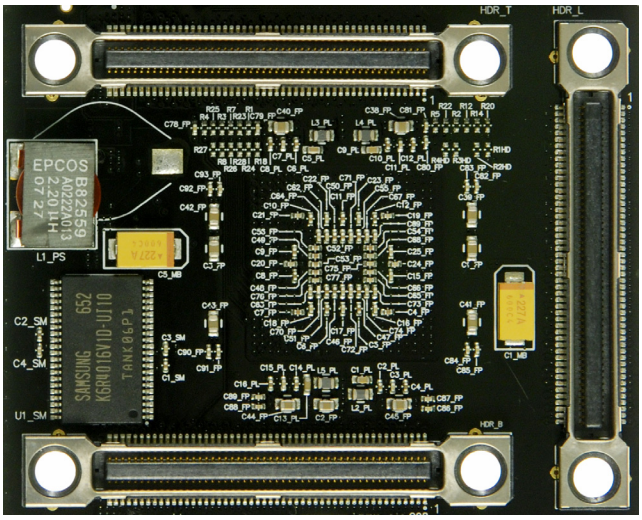


Figure 5. Docking connectors on the daughter board, used to attach the board to the NB2DSK01 motherboard.

Location on board

The docking connectors (`HDR_T`, `HDR_L` and `HDR_B`) are located on the solder side of the board.

Schematic reference

The docking connectors can be found on sheet `DB_MotherBoardConnectors.SchDoc` (entitled *NB2 to Daughter Board Connectors*) of the daughter board schematics.

Board ID memory

Board identification is handled courtesy of a DS2406 device (from Dallas Semiconductor). Although the device is actually a dual-addressable switch, it is used for the additional 1kbit of memory that it possesses.

The DS2406 is a 1-Wire® compatible device, primarily used to contain a code with which to identify the daughter board when plugged in. The NanoTalk Controller interrogates this device over a single wire, the associated signal of which arrives at the Controller as `ONE_WIRE_DBID`.

The DS2406 supports half-duplex communication at a rate of up to 16.3kbits/s. Although power for the device is sourced over the same single wire as the data itself – courtesy of a parasitic capacitor which charges when the 1-Wire bus signal line is High – the device is also fed from the daughter board's 3.3V supply, to ensure continuous power.

Similar devices on all peripheral boards and 3-connector daughter boards allow the NanoTalk Controller to detect which specific boards are available to the system.

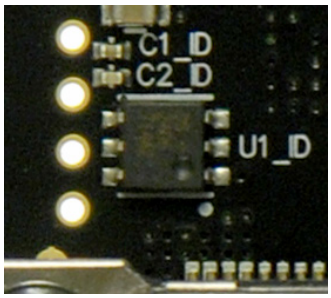


Figure 6. 1-Wire memory used to contain the ident code for the board.

Location on board

The DS2406 device (designated `U1_ID`) is located on the solder side of the board, to the top-left of the 100-way docking connector `HDR_T`.

Schematic reference

The 1-Wire memory device can be found on sheet `1WB_DS2406_EPROM.SchDoc` (entitled *1-Wire Bus ID*) of the daughter board schematics.

Further device information

For more information on the DS2406 device, refer to the datasheet (`DS2406.pdf`) available at *www.maxim-ic.com*.

Power

In addition to the 3.3V and 5V supply voltages and ground signals fed to the daughter board from the NB2DSK01 motherboard, the daughter board also has its own 1.2V supply. This supply is generated by passing the regulated 5V supply through a low-voltage step-down regulator – a MAX1831 device, from Maxim.

The daughter board provides an array of test points for checking the integrity of each of the board's power supplies, as well as GND.

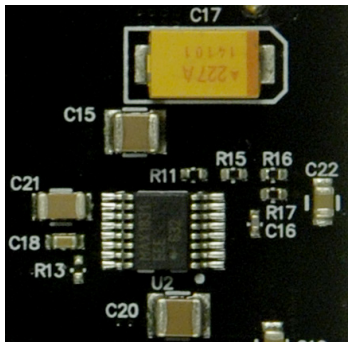


Figure 7. MAX1831 (U2) device.

Location on board

The MAX1831 device (designated `U2`) is located on the component side of the board, to the right of the LatticeECP device.

The power supply test points (designated `TP0 – TP4`) are also located on the component side of the board, to the right of the Flash memory device (`U1_CM`).

Schematic reference

The power supply-related circuitry can be found on the following sheets of the daughter board schematics:

- `PSU.SchDoc` (entitled *Power Supply Top Level*)
- `PSU_MAX1831_1V2_ALT.SchDoc` (entitled *Power Supply MAX1831 (1V2)*).

Further device information

For more information on the MAX1831 device, refer to the datasheet (`MAX1830-MAX1831.pdf`) available at *www.maxim-ic.com*.

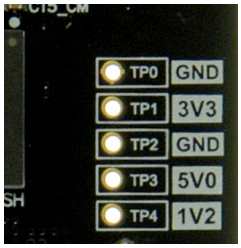


Figure 8. Power supply test points.

Power and Program LEDs

The daughter board provides the following two LEDs:

- **Power LED** – this will light (RED) when the daughter board is correctly plugged into the motherboard and the NB2DSK01's power is switched on. It signifies presence of the 3.3V supply to the board.
- **Program LED** – this will light (GREEN) when the target device on the daughter board has been successfully programmed with an FPGA design.

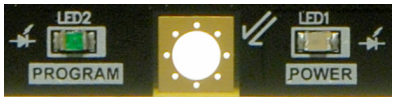


Figure 9. Status and Power LEDs.

Location on board

The two LEDs, labeled `'POWER'(LED1)` and `'PROGRAM'(LED2)` respectively, are located at the bottom of the board, on the component side.

Schematic reference

The LEDs can be found on sheet `DB_LEDS.SchDoc` (entitled *Daughter Board LEDs*) of the daughter board schematics.

Target FPGA device

The LFECP33E-3FN672C device on the daughter board is a member of the 1.2V ECP family of FPGAs. The ECP provides a low-cost, high-density solution for applications such as those targeted to the consumer electronics industry.

The entire ECP family includes five devices offering densities ranging from 6,100 to 32,800 LUTs. The LFECP33E-3FN672C offers 32,800 LUTs. Table 1 provides an information summary for this device.

Note: In order to use the ECP device on this daughter board, you will need to install the relevant Vendor tools – Lattice® ispLever® from *www.latticesemi.com*. The ispLever Starter software can be downloaded but does require a license. Check the website for licensing options.

Figure 10. LatticeECP (LFECP33E-3FN672C).

Table 1. Feature summary for the ECP FPGA device.

Feature	Description
Device Name	LFECP33E-3FN672C
Vendor	Lattice
Family	ECP
Package	672-Ball Fine Pitch Ball Grid Array (fpBGA672) – Lead Free
Speed Grade	3
Temperature Grade	Commercial
Pin Count	672
Maximum User I/O Pins	496
Max. Differential I/O Pairs	248
LUTs	32.8K
Embedded (Block) RAM	498K bits (over 54 blocks)
Distributed RAM	131K bits
DSP Blocks	8
Embedded Multipliers	64 (9x9); 32 (18x18); 8 (36x36)
Clock Managers (PLLs)	4
Global Clock Resources	4
Configuration Memory Required	8,089,600 bits
On-Chip Termination Support	No

Supported single-ended I/O standards

Table 2 lists the single-ended I/O standards supported by the LFECP33E-3FN672C device.

Table 2. Supported single-ended I/O standards.

I/O Standard	Description
HSTL15 Class I	High-Speed Transceiver Logic (1.5V) Class I
HSTL15 Class III	High-Speed Transceiver Logic (1.5V) Class III
HSTL18 Class I	High-Speed Transceiver Logic (1.8V) Class I
HSTL18 Class II	High-Speed Transceiver Logic (1.8V) Class II
HSTL18 Class III	High-Speed Transceiver Logic (1.8V) Class III
LVTTL	Low-Voltage Transistor-Transistor Logic (3.3V)
LVCMOS12	Low-Voltage Complementary Metal-Oxide Semiconductor (1.2V)
LVCMOS15	Low-Voltage Complementary Metal-Oxide Semiconductor (1.5V)
LVCMOS18	Low-Voltage Complementary Metal-Oxide Semiconductor (1.8V)
LVCMOS25	Low-Voltage Complementary Metal-Oxide Semiconductor (2.5V)
LVCMOS33	Low-Voltage Complementary Metal-Oxide Semiconductor (3.3V)
PCI	Peripheral Component Interconnect (33MHz, 3.3V)
SSTL18 Class I	Stub Series Terminated Logic (1.8V) Class I
SSTL2 Class I	Stub Series Terminated Logic (2.5V) Class I
SSTL2 Class II	Stub Series Terminated Logic (2.5V) Class II
SSTL3 Class I	Stub Series Terminated Logic (3.3V) Class I
SSTL3 Class II	Stub Series Terminated Logic (3.3V) Class II

Supported differential I/O standards

Table 3 lists the differential I/O standards supported by the LFECP33E-3FN672C device.

Table 3. Supported pseudo-differential I/O standards.

I/O Standard	Description
BLVDS	Bus Low-Voltage Differential Signaling (2.5V)
Differential HSTL15 Class I	Differential High-Speed Transceiver Logic (1.5V) Class I
Differential HSTL15 Class III	Differential High-Speed Transceiver Logic (1.5V) Class III
Differential HSTL18 Class I	Differential High-Speed Transceiver Logic (1.8V) Class I
Differential HSTL18 Class II	Differential High-Speed Transceiver Logic (1.8V) Class II
Differential HSTL18 Class III	Differential High-Speed Transceiver Logic (1.8V) Class III
Differential SSTL18 Class I	Differential Stub Series Terminated Logic (1.8V) Class I
Differential SSTL2 Class I	Differential Stub Series Terminated Logic (2.5V) Class I
Differential SSTL2 Class II	Differential Stub Series Terminated Logic (2.5V) Class II
Differential SSTL3 Class I	Differential Stub Series Terminated Logic (3.3V) Class I
Differential SSTL3 Class II	Differential Stub Series Terminated Logic (3.3V) Class II
LVDS	Low-Voltage Differential Signaling

LVPECL	Low-Voltage Positive Emitter-Coupled Logic
RSDS	Reduced Swing Differential Signaling (2.5V)

Location on board

The ECP device (designated `U1`) is located on the component side and in the lower half of the board.

Schematic reference

The ECP device and related circuitry can be found on the following sheets of the daughter board schematics:

- `DEVICES.SchDoc` (entitled *FPGA, LEDs and SRAM Memory*)
- `FPGA.SchDoc` (entitled *FPGA Connections*)
- `FPGA_NonIO.SchDoc` (entitled *FPGA Power and Programming*)
- `Bypass_FPGA_1V2.SchDoc` (entitled *FPGA Bypass Capacitors for 1V2*)
- `Bypass_FPGA_3V3.SchDoc` (entitled *FPGA Bypass Capacitors for 3V3*).

Further device information

For more information on the LFECP33E-3FN672C device, refer to the datasheet (`HB1000.pdf`) available at *www.latticesemi.com*.

User resources

The following sections detail the resources on the DB32 which are accessible by, and available for use within, a design running on the board's target FPGA.

Accessing resources from an FPGA design

Before taking a closer look at the FPGA-accessible resources on the daughter board, it is worth taking time to elaborate just exactly how access to these resources is made possible.

Normally you would use ports to connect from the nets in a design to the pins on the FPGA. However, since the connectivity from the FPGA to the memory components on the daughter board is fixed by the routing, there is no need to place ports and then define the net-to-pin mapping. Instead, Altium Designer provides special components that can be placed – allowing the daughter board's memory resources to be easily incorporated in designs. These components, which can be thought of as design interface components and are commonly referred to as port components, are available in the FPGA DB Common Port-Plugin integrated library (`\Library\Fpga\FPGA DB Common Port-Plugin.IntLib`).

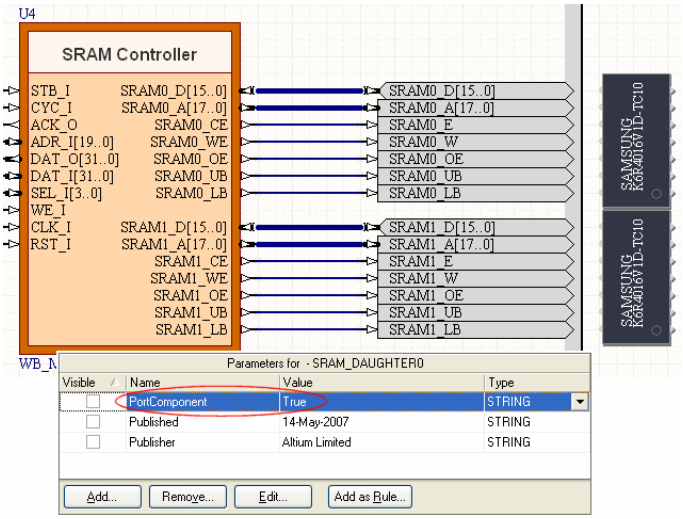


Figure 11. Example port components.

The port components automatically establish connectivity between the resource and FPGA IO pins, allowing the same design to be built for different FPGA devices from different manufacturers. They are placed on the top sheet of the FPGA design project (`*.PrjFpg`), instead of ports. They are recognized as being external to the FPGA design by the presence of the `PortComponent = True` parameter. They are automatically converted to ports during synthesis.

Converting a port component to standard ports

It can often be clearer to represent the signal lines to the physical pins of the FPGA device using standard ports, rather than the port components. Altium Designer provides a conversion command to quickly change a port component to standard ports. From the schematic document, this command can be accessed by:

- Choosing **Tools » Convert » Convert Part To Ports** from the main menus and clicking on the required port component you wish to convert.
- Right-clicking over the required port component and choosing **Part Actions » Convert Part To Ports** from the menu that appears.

After launching the command (and choosing the port component if applicable), the component will be converted into electrically equivalent ports.

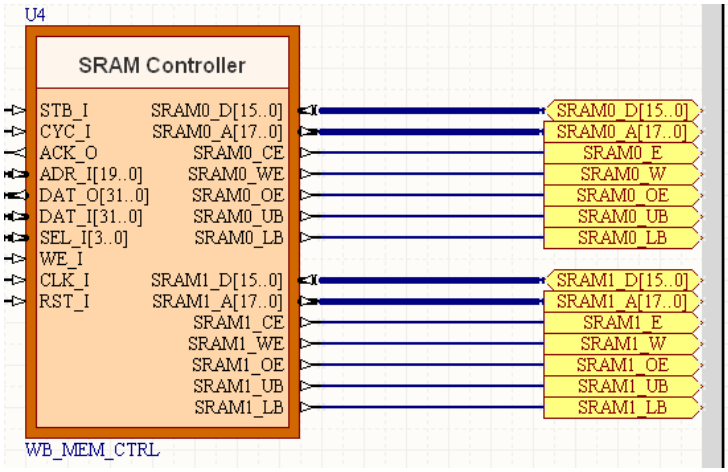


Figure 12. Converted port components.

Common-Bus SRAM

The DB32 includes Static RAM as part of the common-bus block of memory resources available to the FPGA.

The SRAM is provided in the form of two 4Mbit, high-speed CMOS SRAM devices. Each device is organized as 256K x 16 bits – combined together to give 256K x 32-bit storage (1MByte). Both devices are powered by the daughter board's 3.3V supply.

Although the devices require 18 address lines (SRAM_A[19..2]), the common bus interface makes provision for a nineteenth address line (SRAM_A20), giving the daughter board the flexibility to accommodate 512 x 16-bit devices, should they require to be fitted.

Figure 13. Common-bus SRAM (`U4_CM`).

Location on board

The common-bus SRAM device `U4_CM` is located on the component side of the board, to the right of the common-bus SDRAM device `U2_CM`.

The common-bus SRAM device `U5_CM` is located in a corresponding position on the solder side of the board, to the left of the common-bus SDRAM device `U3_CM`.

Figure 14. Common-bus SRAM (`U5_CM`).

Schematic reference

The common-bus SRAM devices can be found on sheet `SRAM_256Kx32_TSOP44_1.SchDoc` (entitled *256K x 32 SRAM – TSOP44 x 2*) of the daughter board schematics.

The common-bus memory block and interface wiring can be found on sheet `NB2_CommonMemory.SchDoc` (entitled *Common-Bus Memory Block*).

Common-Bus SDRAM

The DB32 includes Synchronous Dynamic RAM as part of the common-bus block of memory resources available to the FPGA.

The SDRAM is provided in the form of two MT48LC16M16A2TG devices (from Micron Technology). Each device is a 256Mbit, high-speed CMOS SDRAM, organized as 16M x 16 bits (4M x 16 bits x 4 banks) – combined together to give 16M x 32-bit storage (64MByte). Both devices are powered by the daughter board's 3.3V supply.

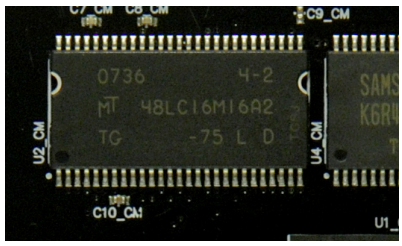


Figure 15. Common-bus SDRAM (`U2_CM`).

Location on board

The common-bus SDRAM device `U2_CM` is located on the component side of the board, to the left of the common-bus SRAM device `U4_CM`.

The common-bus SDRAM device `U3_CM` is located in a corresponding position on the solder side of the board, to the right of the common-bus SRAM device `U5_CM`.

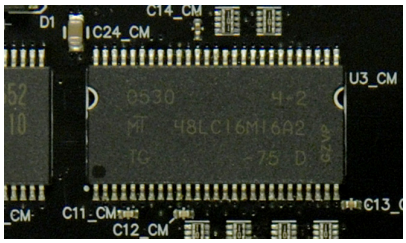


Figure 16. Common-bus SDRAM (`U3_CM`).

Schematic reference

The common-bus SDRAM devices can be found on sheet `SDRAM_MT48LC16M16A2TG_16Mx32.SchDoc` (entitled *16M x 32 SDRAM TSOP54 x 2*) of the daughter board schematics.

The common-bus memory block and interface wiring can be found on sheet `NB2_CommonMemory.SchDoc` (entitled *Common-Bus Memory Block*).

Further device information

For more information on the MT48LC16M16A2TG device, refer to the datasheet (*256MSDRAM.pdf*) available at *www.micron.com*.

Common-Bus Flash memory

The DB32 includes Flash memory as part of the common-bus block of memory resources available to the FPGA.

The Flash memory is provided in the form of an S29GL256N11FFIV10 device (from Spansion). The Page Mode 256Mbit device is manufactured using 110nm MirrorBit™ technology and is organized as 16M x 16 bits (32MByte). It is powered by the daughter board's 3.3V supply and offers a 110ns access time.

Although this device can operate with 8-bit or 16-bit data bus widths, it has been fixed at 16-bit for the daughter board.

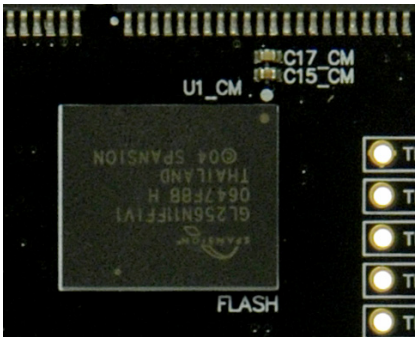


Figure 17. Common-bus Flash memory.

Location on board

The common-bus Flash memory device (designated `U1_CM`) is located on the component side of the board, directly below the common-bus SRAM and to the left of the power supply test points.

Schematic reference

The common-bus Flash memory device can be found on sheet `FLASH_S29GL256N11FFIV10_16Mx16.SchDoc` (entitled *16M x 16 Flash Memory (BGA)*) of the daughter board schematics.

The common-bus memory block and interface wiring can be found on sheet `NB2_CommonMemory.SchDoc` (entitled *Common-Bus Memory Block*).

Further device information

For more information on the S29GL256N11FFIV10 device, refer to the datasheet (*s29gl-n_00_b3_e.pdf*) available at *www.spansion.com*.

Accessing common-bus memory devices

Table 4 summarizes the available design interface components that can be placed from the `FPGA DB Common Port-Plugin.IntLib` for access to, and communications with, any or all of the common-bus memory resources on the daughter board.

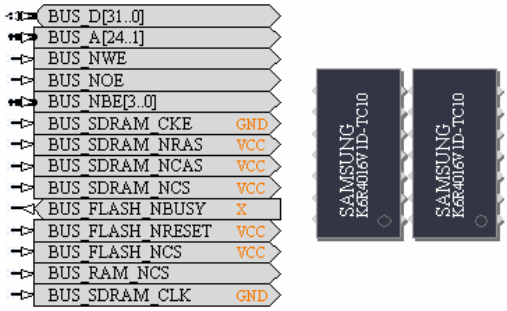
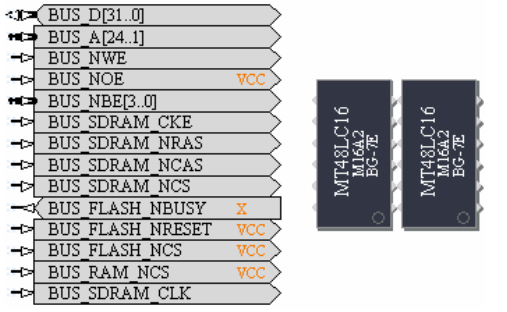
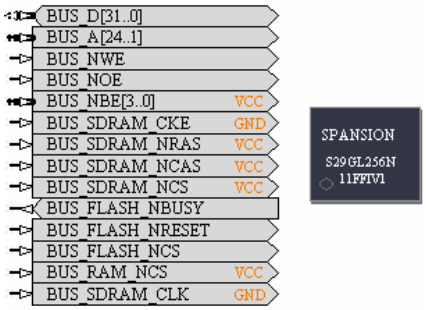
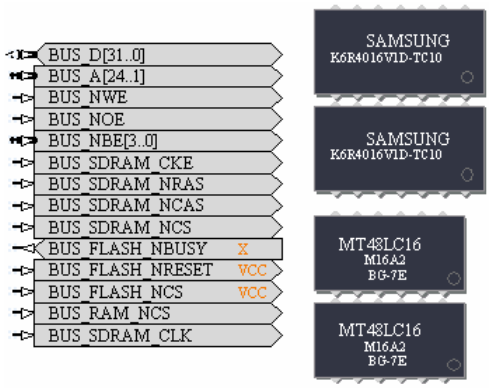
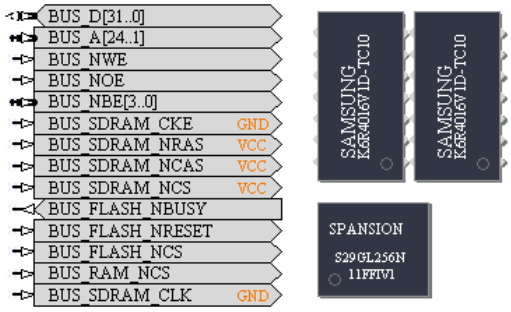
Only ONE common-bus related port component can be placed in an FPGA design. The design interface component used will depend on which particular memory resource(s) you wish to access, and how you have configured the Shared Memory Controller – the intermediate design peripheral that sits between a processor in the design and the common-bus memory on the daughter board.

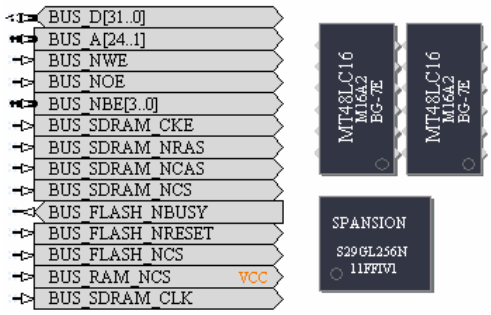
The Shared Memory Controller has an option to control the visibility of unused pins – in terms of its physical memory interface. By default, this option is enabled and all pins of the interface will be shown. In terms of wiring, it is more convenient to leave all interface pins visible and wire all pins to the corresponding ports of the SHARED_MEM_DAUGHTER port component. Those signals not required for the design, in accordance with the Controller's configuration, are internally handled. For example, active Low outputs relating to unused memory types are tied to '1', unused inputs are ignored and, if not using the SDRAM, the associated clock output signal will be tied to '0'.

However, you may not be using all three memory types in your design, or may prefer to hide the 'clutter' of pins associated with memory types that are not being used. You would then place the port component that truly corresponds to the memory you are wishing to access, only wiring from the Shared Memory Controller to those pins of relevance. For the remaining pins of the port component chosen, use the orange guidance text on each unused port to terminate it correctly:

- `VCC` – connect the unused port to `VCC`
- `GND` – connect the unused port to `GND`
- `X` – place a No ERC directive on the unused port.

Table 4. Common-bus memory port-plugin components.

Component Symbol	Component Name	Description
	SHARED_SRAM_DAUGHTER	Place this component to interface to the daughter board's common-bus SRAM.
	SHARED_SDRAM_DAUGHTER	Place this component to interface to the daughter board's common-bus SDRAM.
	SHARED_FLASH_DAUGHTER	Place this component to interface to the daughter board's common-bus Flash memory.
	SHARED_SRAM_SDRAM_DAUGHTER	Place this component to interface to the daughter board's common-bus SRAM and SDRAM.
	SHARED_SRAM_FLASH_DAUGHTER	Place this component to interface to the daughter board's common-bus SRAM and Flash memory.

	SHARED_SDRAM_FLASH_DAUGHTER	Place this component to interface to the daughter board's common-bus SDRAM and Flash memory.
	SHARED_MEM_DAUGHTER	Place this component to interface to all of the daughter board's common-bus memory resources.

For information on available peripheral devices – used to provide the control interface between a processor in the design and the daughter board memory resource – go to *www.altium.com/nanoboard/resources*.

Independent SRAM

The DB32 also includes independent Static RAM as part of the memory resources available to the FPGA device, or more specifically the design running within. The term 'independent' is used in this case to distinguish this SRAM – which is interfaced using dedicated address and data lines – from the SRAM that is accessed over a common bus (also used to access on-board SDRAM and Flash memory).

The SRAM is provided in the form of two 4Mbit, high-speed CMOS SRAM devices. Each device is organized as 256K x 16 bits and powered by the daughter board's 3.3V supply.

The devices are accessed separately, giving two distinct 256K x 16-bit storage areas (512KByte each, 1MByte in total).

Although the devices require 18 address lines (SRAM_A[17..0]), the interface makes provision for a nineteenth address line (SRAM_A18), giving the daughter board the flexibility to accommodate 512 x 16-bit devices, should they require to be fitted.

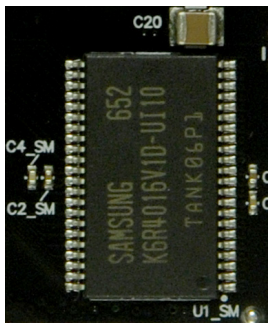


Figure 18. Independent SRAM (`U1_SMA`).

Location on board

The independent SRAM device `U1_SMA` is located on the component side of the board, towards the bottom-right corner.

The independent SRAM device `U1_SMB` is located in a corresponding position on the solder side of the board, towards the bottom-left corner.

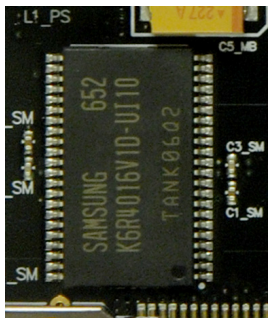


Figure 19. Independent SRAM (`U1_SMB`).

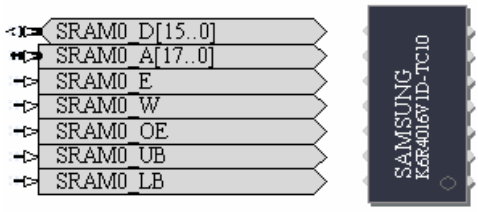
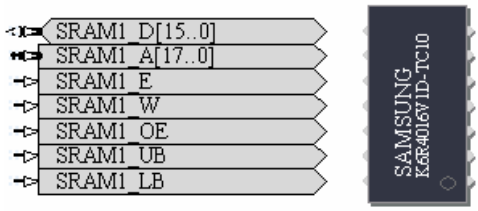
Schematic reference

The independent SRAM device can be found on sheet `SRAM_256Kx16_TSOP44.SchDoc` (entitled *256K x 16-Bit SRAM*) of the daughter board schematics.

Design interface component

Table 5 summarizes the available design interface components that can be placed from the `FPGA DB Common Port-Plugin.IntLib` for access to, and communications with, the independent SRAM on the daughter board.

Table 5. Independent SRAM port-plugin components.

Component Symbol	Component Name	Description
	SRAM_DAUGHTER0	Place this component to interface to the independent SRAM device U1_SMA on the daughter board.
	SRAM_DAUGHTER1	Place this component to interface to the independent SRAM device U1_SMB on the daughter board.

For information on available peripheral devices – used to provide the control interface between a processor in the design and the daughter board memory resource – go to *www.altium.com/nanoboard/resources*.

Desktop NanoBoard NB2DSK01 Constraint system

The process of mapping or constraining a design to its physical implementation is done by creating constraint files – files that specify implementation detail such as the target device, the port-to-pin mapping, pin IO standards, and so on. The minimum information required to synthesize a design is the device specification.

Sets of constraint files are targeted to a design by creating a configuration, which is simply a named list of constraint files.

Setting up to implement a design on the NB2DSK01 is simplified through use of an auto-configuration feature, whereby a target configuration for the FPGA design project is automatically created. The required constraint files are automatically determined and added to this configuration, based on the IDs of the hardware (motherboard, daughter board and peripheral boards) in the system.

For more information on the concept of configurations and constraints, and their role in design portability, refer to the article *AR0124 Design Portability, Configurations and Constraints*.

For more detailed information on the Desktop NanoBoard NB2DSK01 constraint system, including auto-configuration, refer to the application note *AP0154 Understanding the Desktop NanoBoard NB2DSK01 Constraint System*.

Revision History

Date	Version No.	Revision
01-Nov-2007	1.0	Initial release
15-Jan-2008	1.1	Updated for Altium Designer 6.9.
28-Apr-2008	2.0	Updated for Altium Designer Summer 08