

## Introduction

The Excalibur™ Development Kit, featuring the Nios™ embedded processor, includes the software, hardware, accessories, and documentation necessary to create working embedded systems. Using the software and hardware included in the kit, you can specify the peripherals and bus configuration for your Nios embedded microcontroller as well as add custom hardware blocks to complete your embedded system. When you are finished creating a design, you can implement it in an Altera® device. To determine which Altera device best meets your design requirements, it is important to be able to estimate the device resources that are used and the operating frequency. This document can help you make these estimates for your Nios embedded processor design.

This document describes the LE usage and performance ( $f_{MAX}$ ) of the 32- and 16-bit Nios reference designs included in the Excalibur Development Kit, featuring the Nios embedded processor version 1.1.1, and explains how Altera calculated the results. The reference designs have a full-featured microcontroller with a memory interface and built-in peripherals; they demonstrate real-world embedded systems implemented on an Altera device. You can use the techniques described in this application note to calculate the resource usage and performance of your own Nios designs.



This application note assumes that you are familiar with the Nios embedded processor, Altera device architectures, and the Quartus® II development tools.

## Software Versions Used

This document provides data based on push-button compilation of the 32- and 16-bit reference designs, i.e., the designs were not optimized. Altera used the following software versions to calculate the results:

- Nios embedded processor version 1.1.1
- Quartus II software version 1.1



Refer to the Literature pages on the Altera web site at <http://www.altera.com> for Nios peripheral documentation. Refer to the Altera device family data sheets for more information about the Altera device you want to use.

## LE Usage & Performance

When designing with the Nios embedded processor version 1.1 or higher, you can target a number of Altera device families, including FLEX10K®, FLEX® 10KE, ACEX™, APEX™ 20K, APEX II, and Mercury™ devices. See [Table 1](#) for a description of device families and development tools supported.

Device Family	Development Tool Support
FLEX 10K, FLEX 10KE	MAX+PLUS II, Quartus II
ACEX 1K	MAX+PLUS II, Quartus II
APEX 20K	Quartus II
APEX II	Quartus II
Mercury	Quartus II

This section describes the LE usage and performance of the Nios processor, multiplication units, and peripherals in Altera devices.

### Reference Designs

The 32- and 16-bit reference designs implement systems that are comparable to most dedicated microcontrollers. [Table 2](#) shows the Nios processor configuration for the two designs. The designs are located in the `\Altera\Excalibur\nios_example_designs\reference_design_32_bit` and `Altera\Excalibur\nios_example_designs\reference_design_16_bit` directories.



The **Include Multiply-step (MSTEP) instruction** option is turned off in the 32-bit reference design. See [“Multiplication Units” on page 4](#) for the LE usages with this option turned on.

Processor Configuration	32-Bit CPU	16-Bit CPU
Register File Size	256	128
Data Path (Bits)	32	16
Address Bus Width	21	16
ALU Shifter	7-bit	3-bit
Hardware Multiply Assist	None	Not Applicable

[Table 3](#) shows the reference design peripherals; both designs use the same peripherals.

Peripheral	Type
LCD Port	11-Bit Output Parallel Input/Output (PIO)
LED Port	2-Bit Bidirectional
7-Segment LED Port	16-Bit Output PIO
Push-Button Port	12-Bit Input PIO
Timer	32-Bit Timer
UART	Fixed Baud Rate at 115200
Memory Interface	32- or 16-Bit SRAM (1)
Memory Interface	16-Bit Flash (2)
On-Chip ROM	1K with Boot Monitor

**Note:**

- (1) The 32-bit reference design addresses 256K of 32-bit SRAM memory; the 16-bit reference design addresses 32K of 16-bit SRAM memory.
- (2) The 32-bit reference design addresses 1MByte of flash memory; the 16-bit reference design addresses 16K of flash memory.

Table 4 shows the LE usage and  $f_{MAX}$  for the reference designs. The table provides information for the fastest speed grade devices for each family. Altera created the reference designs and compiled them using a standard tool flow without additional optimization. Additionally, no pin constraints were used when compiling the designs.

Device Family	Device	32-Bit			16-Bit		
		LEs	$f_{MAX}$	EAB/ESB Bits (1)	LEs	$f_{MAX}$	EAB/ESB Bits (1)
Mercury	EP1M120F484C5	2,602	59.08	27,744	1,784	70.51	15,200
APEX II	EP2A15F672C7	3,046	55.44	40,960	2,144	59.34	31,488
APEX 20KC	EP20K200CF484C7	3,022	53.15	27,616	2,124	54.09	15,040
APEX 20KE	EP20K200EFC484-1X	3,035	43.26	27,616	2,151	44.50	15,040
APEX 20K	EP20K200FC484-1X	3,018	40.86	27,616	2,132	48.28	15,040
FLEX 10KE	EPF10K100EQC208-1X	3,029	32.47	27,616	2,103	38.82	15,040
ACEX 1K	EP1K100FC484-1	3,042	31.35	27,616	2,119	38.31	15,040

**Note:**

- (1) EAB = embedded array block. ESB = embedded system block.

## Multiplication Units

The Nios CPU has three multiplication unit options. The overall LE usage depends on which option your design uses. Table 5 compares the LE usage of the options for three variants of the reference designs:

- 32-bit reference design without hardware assistance
- 32-bit reference design with the **Include Multiply-step (MSTEP) instruction** option turned on (integer multiplication is performed with 1 clock-cycle per bit)
- 32-bit reference design with the **Include Multiply (MUL) instruction** turned on ( $16 \times 16$  multiplication is performed in 2 clock-cycles)

Device Family	Device	Software Only Multiplication	MSTEP (1)	MUL (1)
Mercury	EP1M120F484C5	2,602	113	437
APEX II	EP2A15F672C7	3,046	124	432
APEX 20KC	EP20K200CF484C7	3,022	124	432
APEX 20KE	EP20K200EFC484-1X	3,035	147	449
APEX 20K	EP20K200FC484-1X	3,018	148	458
FLEX 10KE	EPF10K100EQC208-1X	3,029	221	741 (2)
ACEX 1K	EP1K100FC484-1	3,042	197	734 (2)

**Note:**

- (1) Add the LE count in this column to the LE count in the Software Only Multiplication column to obtain the total LEs using this option.
- (2) To compile the hardware multiplication option for FLEX 10K and ACEX 1K devices, you must turn off the **Auto Cascade Chains** option. To do so, choose **Assignment Organizer** (Tools menu); Expand **Options for Individual Nodes & Entities** in the **By Node** tab; Select **Auto Cascade Chains** in the **Name** drop-down list box, and select **Off** in the **Setting** drop-down list box.

## Peripherals

The Excalibur Development Kit, featuring the Nios embedded processor includes the following on-chip peripherals:

- On-chip memory (RAM or ROM)
- PIO ports
- Serial peripheral interface (SPI)
- Interval timer
- UART

The number of LEs each peripheral uses depends on which Altera device you use. [Table 6](#) shows the number of LEs used by a peripheral included in a 32-bit Nios design targeting either an APEX 20KE or FLEX 10KE device.



These LE counts may vary slightly, depending on the Quartus II on MAX+PLUS II Compiler optimization settings used.

Peripheral	APEX 20KE	FLEX 10KE
LCD Port	78	58
LED Port	16	27
7-Segment LED Port	47	39
Push-Button Port	74	115
Timer	244	292
UART	202	256

When you implement a Nios peripheral, you can configure it to meet your system's requirements. [Table 7](#) describes some of the peripheral configuration options—and how many LEs the option uses—for a 32-bit Nios design targeting either an APEX 20KE or FLEX 10KE device.

Peripheral	Option(s)	APEX 20KE	FLEX 10KE
UART	Fixed Baud Rate	202	191
	Variable Baud Rate	293	343
PIO	32-Bit, Tristate, No Capture, No IRQ	138	167
	16-Bit, Tristate, No Capture, No IRQ	81	79
	8-Bit, Tristate, No Capture, No IRQ	45	35
	32-Bit, Tristate, With Capture, No IRQ	297	300
	32-Bit, Tristate, No Capture, With IRQ Level	223	276
	32-Bit, Tristate, With Capture, With IRQ Level	256	406
	32-Bit, Input and Output Only, No Capture, No IRQ	28	39
	32-Bit, Both Input and Output, No Capture, No IRQ	95	104
Timer	No Options	244	292
SPI	Master, 8-Bit, MSB/LSB, Default Settings, 1 Slave	103	136
	Master, 8-Bit, MSB/LSB, Default Settings, 2 Slaves	108	140
	Slaver, 8-Bit, MSB/LSB, Default Settings	98	111
	Slave, 16-Bit, MSB/LSB, Default Settings	127	167

## Calculating LE Usage

When calculating the LE usage in “[LE Usage & Performance](#)” on page 2, Altera did not use any optimization techniques during synthesis or compilation. To duplicate the calculations, follow the steps in the following sections. You can use these steps to calculate the resource usage and performance of your own Nios designs or to benchmark designs or peripherals in Altera device families. This example uses the Mercury EPM120F484C5 device. The process for using other devices is similar.



This analysis uses the Quartus II software version 1.1. If you use a different version of the software, your results may vary by a few percent.

### Calculating LEs in the Quartus II Software

To find the LE usage for the 32-bit reference design implemented in a Mercury device, perform the following steps:

1. Run the Quartus II software.
2. Choose **Open Project** (File menu).
3. Browse to the `<path>Altera\Excalibur\nios_example_designs\reference_design_32_bit` directory.
4. Select the `reference_design_32_bit.quartus` project file. The `reference_design_32_bit.bdf` file opens.
5. Click **Open**.
6. Select the Altera device to target.
  - a. Choose **Compile Mode** (Processing menu).
  - b. Choose **Compiler Settings** (Processor menu).
  - c. Click the **Chips & Device** tab.
  - d. Select **Mercury** from the **Family** drop-down list box.
  - e. Click **Yes** in the Quartus II message box that warns you that the device family has changed. (You do not want to save any pin assignments.)
  - f. Select the **EPM120F484C5** device in the **Available devices** list box.
  - g. Click **OK**.

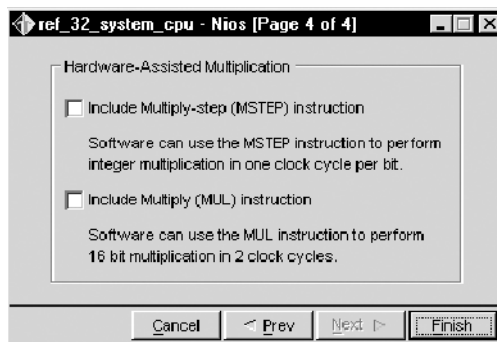
7. Set the parameters of the Nios embedded processor.



Skip this step if you want to calculate the LE usage and performance of the reference design as shipped with the kit.

- a. Double-click the `ref_32_system` symbol to launch the Nios System Builder wizard.
- b. Click **Edit CPU**.
- c. Click **Next** in wizard pages 1, 2, and 3.
- d. In wizard page 4, turn off all hardware-assisted multiplication you want to use by turning the **Include Multiply-step (MSTEP) instruction** or **Include Multiply (MUL) instruction** options off. See [Figure 1](#).

**Figure 1. Hardware Assisted Multiplication window**



- e. Click **Finish**.
8. Change the peripheral settings.



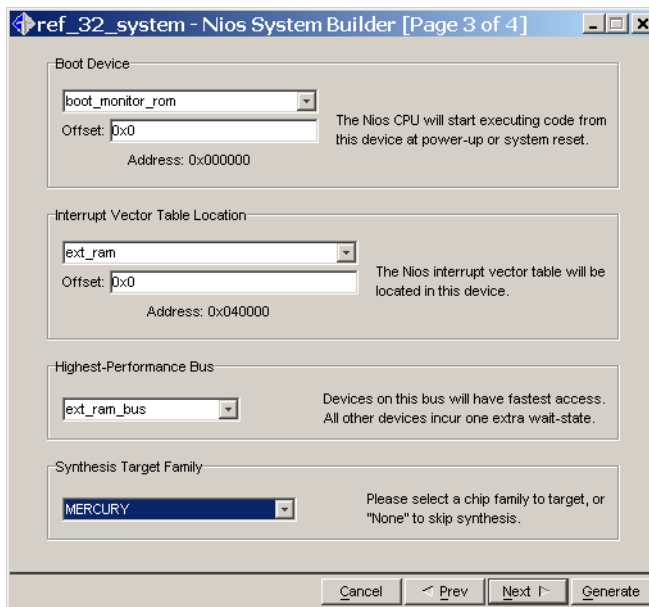
Skip this step if you want to calculate the LE usage and performance of the reference design as shipped with the kit.

- a. Select the name for the peripheral that you want to edit to highlight the peripheral.
- b. Click **Next**.
- c. Go through the wizard, changing the parameters.
- d. Click **Finish** when you are done making changes.

- When you are finished changing the CPU and peripheral parameters, click **Next** in the Nios System Builder.
- Choose the targeted device family. See [Figure 2](#).

---

**Figure 2. Nios System Builder (Page 3 of 4)**



- Click **Next**.
- Click **Generate**. The Nios System Builder begins generating the files needed to compile the design.
- Click **Finish** when the file generation completes to close the Nios System Builder.
- Choose **Start Compilation** (Processing menu) to compile the design.

15. When compilation completes, view the LE usage and performance.
  - a. The compilation summary shows the device resources used in the design.
  - b. Expand **Timing Analyses**.
  - c. Click **f<sub>max</sub> (not incl. delays to/from pins)** to view the **f<sub>MAX</sub>** for the design.

### Calculating Peripheral LE Usage

You cannot calculate the LE usage of the Nios peripherals directly; instead, you must compile your design with one set of settings and peripherals and then compile it a second time with changes. Then, compare the results to determine the LEs used. To calculate the LE usage, perform the following steps:

1. Perform the steps in [“Calculating LEs in the Quartus II Software” on page 6](#) to target a device and to find out how many LEs the design uses.
2. Change the design, for example, by removing a peripheral or changing its settings. Do not change the device selection.
3. Recompile the design to determine the LE usage.
4. Compare the LE usage found in step 1 and 3 to determine how many LEs were affected by the design change.

## Conclusion

This application note provides the LE usage and **f<sub>MAX</sub>** performance for the 32- and 16-bit Nios reference designs and common peripheral options. You can use the techniques described in this application note to determine the LE usage and performance of the Nios processor in your own designs.



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>  
**Applications Hotline:**  
(800) 800-EPLD  
**Literature Services:**  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Copyright © 2001 Altera Corporation. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services. All rights reserved.



I.S. EN ISO 9001