White Paper

Stratix II vs. Virtex-4 Density Comparison

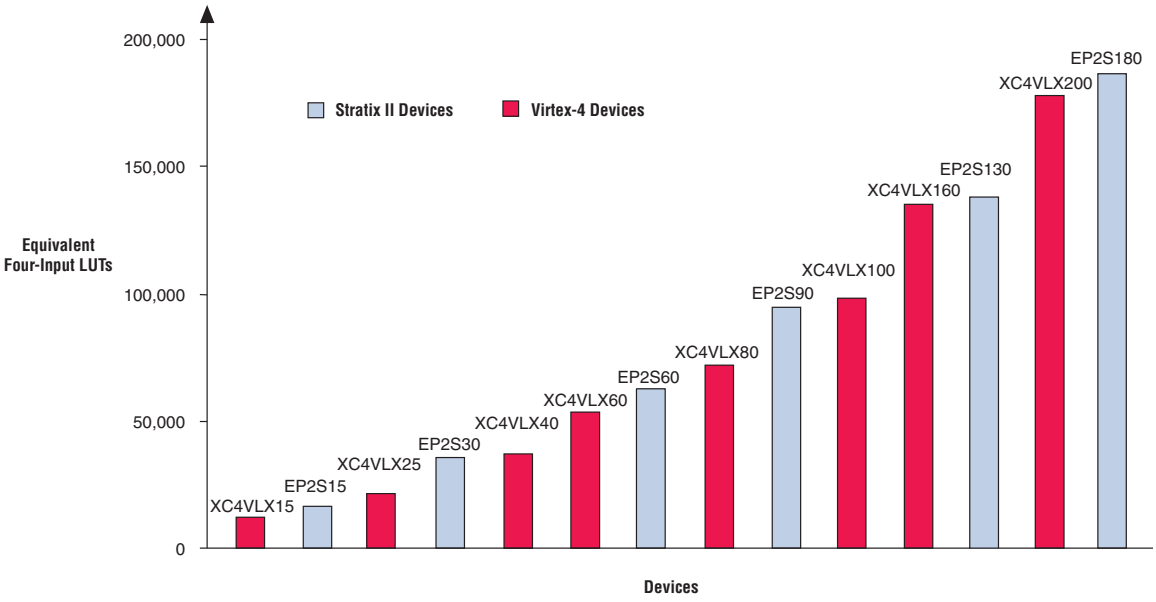
Introduction

Altera® Stratix® II devices are built using a new and innovative logic structure called the adaptive logic module (ALM) to make Stratix II devices the industry's biggest and fastest FPGAs. The ALM packs more combinational logic into less area, providing a higher logic density than a standard four-input look-up table (LUT) architecture and more logic per register. With the Stratix II ALM architecture, the Quartus® II software can reduce the number of levels of logic as well as the routing requirements for a Stratix II design. Altera has shown results of performance benchmarking that demonstrate an average of 18% performance benefit of this new architecture as compared to the Xilinx Virtex-4 product family. This measurement is based on Xilinx's -12 speed grade Virtex-4 devices versus Altera's production -3 speed grade Stratix II devices.

New capacity benchmarks with over 70 real-world designs were run to measure the capacity of Stratix II devices compared to Virtex-4 devices. The primary difference between the capacity and performance benchmarks is that the maximum performance for both Stratix II and Virtex-4 devices is constrained to be equal for capacity benchmarks, and the goal is to use the least amount of logic (i.e., minimize area). The capacity benchmarks provide an average relative measure of capacity for Altera's ALM and Xilinx's slice architectures. Based on the benchmarking results, one Altera Stratix II ALM has an equivalent logic capcaity of 1.3 Xilinx slices.

Figure 1 compares the resources available in Stratix II and Virtex-4 devices. The data used in Figure 1 is shown in Table 4.

Figure 1. Stratix II vs. Virtex-4 LX Logic Capacity Comparison.



The data in Figure 1 provides a well defined capacity relationship between the Altera and Xilinx architectures, enabling designers to accurately compare logic capacity. Figure 1 demonstrates that product naming does not provide an accurate measure of relative capacity between these device families. For example, the EP2S130 device offers more logic than the LX160 device, and the EP2S180 device offers more logic than the LX200 device.

Stratix II ALMs are 30% more logic efficient than the Virtex-4 slices because of two factors.

- The Stratix II ALM contains a new adaptive LUT (ALUT) that can be configured into a six-input function, seven-input function, or shared logic with independent outputs. Each ALM also contains embedded enhanced adder blocks to perform small-area and high-performance arithmetic functions, such as adder trees.
- The Quartus II software provides leading-edge area optimization algorithms, resulting in better area utilization in Stratix II FPGAs while maintaining equivalent performance compared to Virtex-4 FPGAs for the same design. The Quartus II software can also provide higher performance for Stratix II devices over ISE for Virtex-4 devices.

This white paper explains how Stratix II devices achieve a higher logic efficiency. This white paper also compares the technical implementation of a variety of design building blocks such as adder trees, wide functions, and barrel shifters in Stratix II and Virtex-4 architectures.

The Virtex-4 slice architecture is similar to Virtex, Virtex-E, Virtex-II, Virtex-II Pro, Spartan-II, Spartan-IIE, Spartan-3 and Spartan-3E device architectures. Therefore, the 30% logic efficiency advantage that Stratix II ALMs provide over Virtex-4 devices is equally applicable for comparison with all these Xilinx families.

The most useful and accurate representation of logic usage in a design is through FPGA software compilation. Designs must go through the full place and route procedures in the FPGA vendor's software to obtain the true utilization for a particular design.

Capacity of a Logic Unit in FPGAs

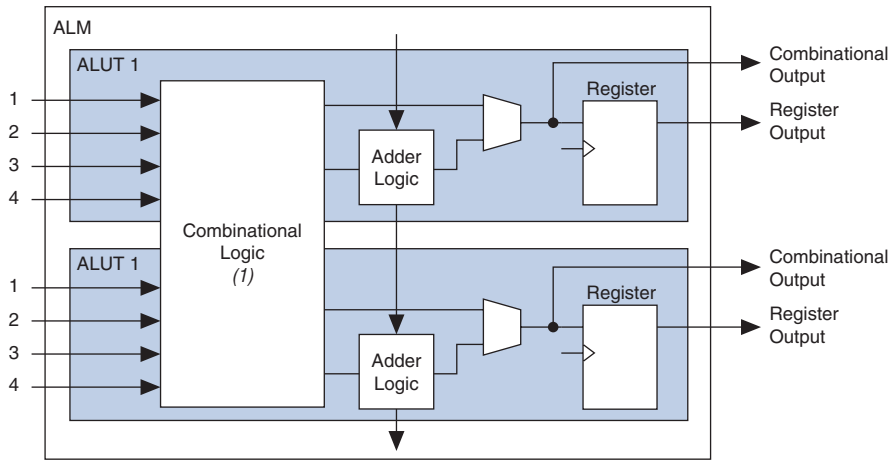
The fundamental logic architectures across FPGA vendors are different and there is no direct one-to-one mapping to compare the logic unit of one FPGA architecture to another. The only effective method to compare FPGA capacity is to complete full place and route compilations using the appropriate software tools.

The efficiency of logic utilization of an FPGA architecture depends on the following factors:

- The logic capacity of a single logic unit
- The structure of the design, such as whether the design includes multiplexing, wide functions, or arithmetic functions
- The embedded functions that are present in the FPGA, such as a DSP block or embedded RAM
- The effectiveness of the synthesis tool
- The quality of the place and route software

The Stratix II device logic unit is the ALM. A single ALM contains two ALUTs, which provide two independent and highly flexible combinational outputs, two adder logic blocks, and two registers. See Figure 2.

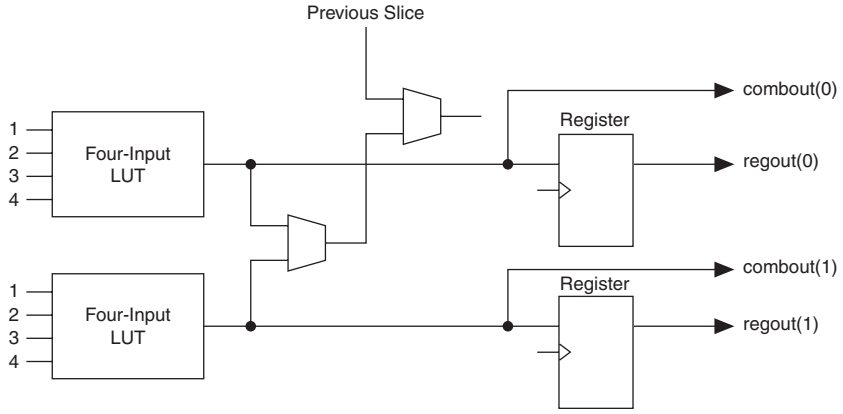
Figure 2. Stratix II ALM



Note to Figure 2:
(1) The combinational logic is adaptively divided between the two ALUTs.

The Virtex-4 device logic unit is the slice. A Virtex-4 slice can be divided into two half-slices. Each slice consists of two 4-input LUTs, embedded multiplexers, carry logic, and two registers (see Figure 3). Each LUT can also be configured into a 16 × 1 RAM or a 16-bit shift register. However, when the LUT is configured as RAM, the LUT can no longer perform logic functions.

Figure 3. Virtex-4 Slice



Based on extensive benchmark analysis done using real designs and a full synthesis, place and route design flow targeted for minimum area, the ratio of Stratix II ALMs to Virtex-4 slices is shown in Table 1.

Table 1. Normalized Relative Logic Capacity Note (1)

Stratix II	Virtex-4
1 ALM	1.3 slices

Note to Table 1:
(1) Actual measurements were done on ALUTs and half-slices.

The logic capacity of one Stratix II ALM is equivalent to that of 1.3 Virtex-4 slices. Therefore, on average, if a given function requires four slices in a Virtex-4 device, it only requires three Stratix II ALMs. The following sections provide the evidence and technical explanation of how Stratix II devices achieve such a high logic efficiency.

Benchmark Data

Altera performed extensive benchmarking to determine the logic usage comparison between Stratix II and Virtex-4 devices. The designs used in this benchmark are actual designs using generic VHDL or Verilog HDL. These designs come from a full spectrum of applications, ranging from networking and wireless applications to industrial and consumer electronics. This benchmark is designed to re-create the diversity of real situations in the field to avoid any bias toward a particular market segment or any bias toward Altera technology.

Virtex-4 devices are used in this benchmark, but this comparison applies to all Virtex-based products because of their similar logic structures.

The benchmarking methodology used to generate the empirical data found in Table 1 and Figure 4 is based on Altera's standard benchmarking methodology as defined in the *FPGA Performance Benchmarking Methodology White Paper*. As shown in Table 2, the benchmarking used the Synplify Pro 8.0 synthesis tool and the Quartus II software version 5.0 and the ISE software version 7.1i SP1 for place and route. In addition to optimizing for performance, the Quartus II and ISE softwares were used to minimize area utilization and maximize density. The Quartus II software settings are listed in Table 3, including *Perform WYSIWYG Resynthesis, Optimize for Area, Restructure Multiplexers, and aggressive Register Packing.* Any ISE *map* or *par* options that saved device area were used. After using this methodology with over 70 real designs, the results show that on average, Virtex-4 uses 30% more slices than ALMs used in Stratix II devices. Therefore, a single Stratix II ALM contains 30% more logic capacity than a Virtex-4 slice. Individual logic utilization results are shown for each design in Figure 4.

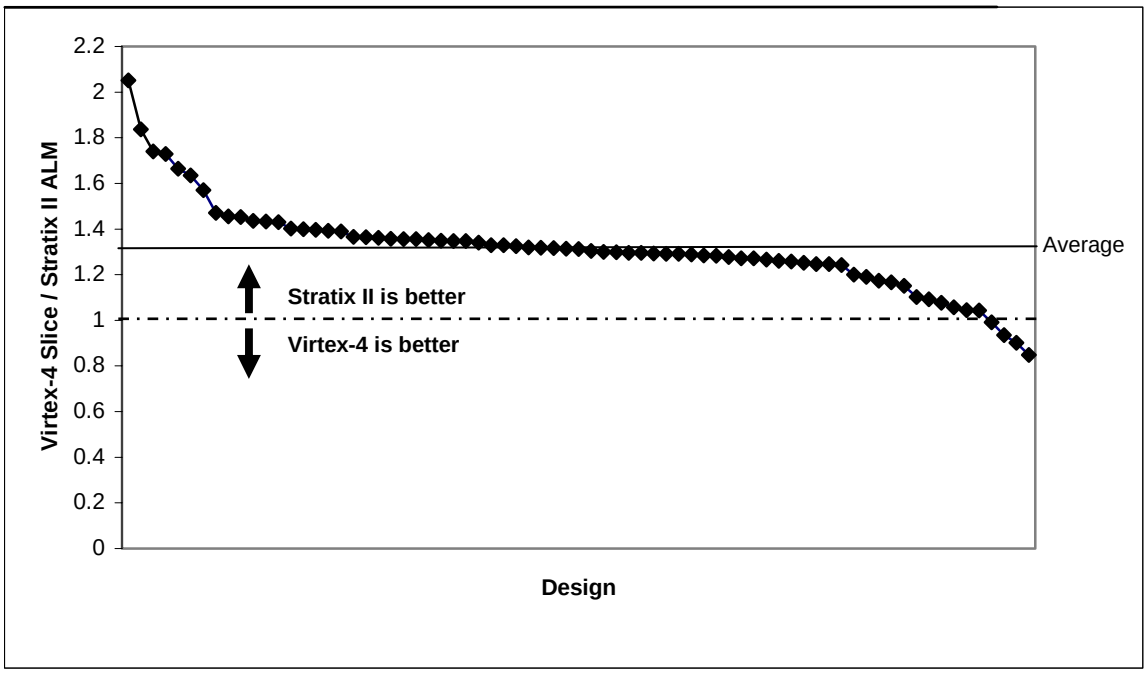
Table 2. Benchmark Parameters

Parameter	Altera	Xilinx
Synthesis tool	Synplicity Synplify Pro 8.0	Synplicity Synplify Pro 8.0
Place and route tool	Quartus II version 5.0	ISE version 7.1i SP1
Device	Stratix II	Virtex-4

Table 3. Quartus II Area Optimization Options

Quartus II Option	Stratix II Benchmark Settings
Perform WYSIWYG resynthesis (On/Off)	On
Optimization technique (Area Balanced/Speed)	Area
Restructure multiplexers (On/Off/Auto)	On
Auto packed registers	Minimize area with chains

Figure 4. Stratix II vs. Virtex-4 Logic Utilization Benchmark



The Y-axis of the graph in Figure 4 is the logic utilization ratio, and each point along the X-axis represents a single design. For each design, the logic utilization ratio is defined as the number of Virtex-4 slices needed to implement the design divided by the number of Stratix II ALMs needed to implement the same design. A logic utilization ratio below 1.0 means that more Stratix II ALMs are used than Virtex-4 slices for a given design, while data points above 1.0 represent more efficient Stratix II logic utilization. Based on this data, designs consume an average of 30% more Virtex-4 slices than Stratix II ALMs.

Logic utilization can vary greatly with the structure of the design. Some designs have large differences in logic utilization between Stratix II and Virtex-4, and others have similar total utilization. However, on average, most designs are implemented more efficiently in the Stratix II architecture.

Stratix II vs. Virtex-4 Devices

Based on the benchmark result with an equivalency of 1 ALM to 1.3 slices, Table 4 compares the Stratix II and Virtex-4 device density normalized to four-input LUTs. Table 4 provides information on equivalent device density when comparing Stratix II and Virtex-4 devices. However, because of variations in design structures and logic utilization, you should still compile your design in the Quartus II software and ISE software tools to obtain the actual device utilization.

Table 4. Stratix II vs. Virtex-4 LX Equivalent Device Matchup

Stratix II				Virtex-4 LX		
Device	**ALMs**	**Equivalent Slices *(1)***	**Equivalent Four-Input LUTs *(2)***	**Device**	**Slices**	**Equivalent Four-Input LUTs *(3)***
EP2S15	6,240	8,112	16,224	XC4VLX15	6,144	12,288
				XC4VLX25	10,752	21,504
EP2S30	13,552	17,617	35,235			
				XC4VLX40	18,432	36,864
EP2S60	24,176	31,429	62,858	XC4VLX60	26,624	53,248
				XC4VLX80	35,840	71,680
EP2S90	36,384	47,299	94,598	XC4VLX100	49,152	98,304
EP2S130	53,016	68,921	137,842	XC4VLX160	67,584	135,168
EP2S180	71,760	93.288	186,576	XC4VLX200	89,088	178,176

Notes to Table 4:
(1) This information is based on empirical benchmark data. One ALM is equivalent to 1.3 slices.
(2) Each Altera Stratix II ALM contains 2.6 equivalent four-input LUTs.
(3) Each Xilinx Virtex-4 slice contains 2.0 equivalent four-input LUTs as referenced in the Virtex-4 User Guide version 1.3.

Stratix II Adaptive Logic Module

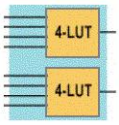
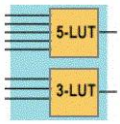
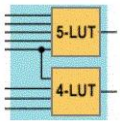
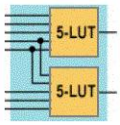
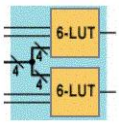
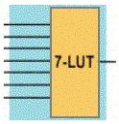
This section provides information on Stratix II ALMs and the available features.

ALM Adaptive Combinational Logic Support

Each ALM can be configured to perform combinational logic functions or logic-and-arithmetic operations. Each ALM contains a variety of LUT-based resources that can be adaptively divided between two logical outputs. With up to eight inputs to the combinational logic block, one ALM can implement various combinations of two functions.

Table 5 shows the available combinational logic configurations in an ALM. Refer to the *Stratix II Device Handbook* for detailed architecture descriptions.

Table 5. ALM Adaptive Combinational Logic Configurations

Configuration	Description
	One Stratix II ALM can be configured to implement two independent four-input (or smaller) LUTs. This configuration can be viewed as the "backward-compatibility" mode. Designs that are optimized for the traditional four-input LUT (4-LUT) FPGAs can easily be migrated to the Stratix II family.
	One Stratix II ALM can be configured to implement a five-input LUT (5-LUT) and a three-input LUT (3-LUT). The inputs to the two LUTs are independent of each other. The 3-LUT can be used to implement any logic function that has three or fewer inputs. Therefore, 5-LUT-2-LUT is also allowed.
	One Stratix II ALM can be configured to implement a five-input LUT and a four-input LUT. One of the inputs is shared between the two LUTs. The 5-LUT has up to four independent inputs. The 4-LUT has up to three independent inputs. The sharing of inputs between LUTs is very common in FPGA designs, and the Quartus II software automatically seeks logic functions that are structured in this manner.
	One Stratix II ALM can be configured to implement two five-input LUTs (5-LUTs). Two of the inputs between the LUTs are common, and up to three independent inputs are allowed for each 5-LUT.
	A Stratix II ALM can implement any six-input function (6-LUT). If there are two six-input functions that have the same logic operation and four shared inputs, then the two six-input functions can be implemented in one Stratix II ALM. For example, a 4x2 crossbar switch that has four data input lines and two sets of unique select signals requires four LEs in the Stratix family. In the Stratix II family, this function only requires one ALM. Another example is a six-input AND gate. An ALM can implement two six-input AND gates that have four common inputs.
	One Stratix II ALM in the extended mode can implement a subset of a seven-variable function. The Quartus II software automatically recognizes the applicable seven-input function and fits it into an ALM. Refer to the *Stratix II Device Handbook* for detailed information about the types of seven-input functions that can be implemented in an ALM.

ALM Arithmetic Support

In addition to the adaptive combinational logic support, each ALM contains two full adders to perform various arithmetic operations. ALMs support arithmetic mode and shared arithmetic mode.

Table 6 shows a summary of the two arithmetic operation configurations in an ALM. Refer to the *Stratix II Device Handbook* for detailed architecture descriptions.

Table 6. ALM Arithmetic Operation Configurations

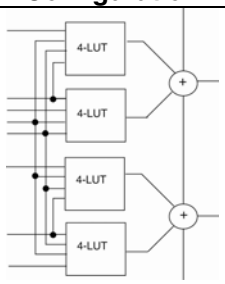
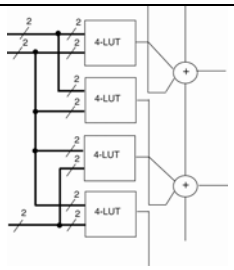
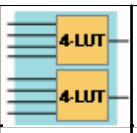
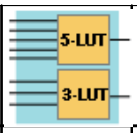
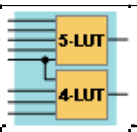
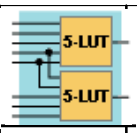
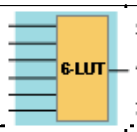
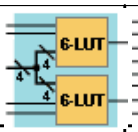
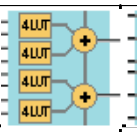
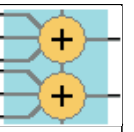
Mode	Configuration	Description
Arithmetic mode		An ALM can be configured to implement complex logic-arithmetic-combined operations. The arithmetic operation is computed by the two dedicated full adders, leaving the LUTs available to perform pre-adder logic. For example, data selection logic before the summation can be easily combined with the summation operation in the same ALM, which improves the performance and reduces the logic resource usage.
Shared arithmetic mode		An ALM can be configured to perform complex arithmetic operations by allowing the sharing of LUT resources between the ALUTs. For example, one ALM can be configured to perform two three-input adders. Allowing the summation of three numbers at once greatly reduces the number of summation stages in an adder tree, thus improving the performance and reducing logic resource usage.

Table 7 provides a summary of comparison for all modes shown in Table 5.

Table 7. Summary of Comparison For all Possible Modes Note (1)

Function								
Stratix II	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM	1 ALM
Virtex-4*	1 Slice	1.5 Slices	1.5 Slices	1.5-2Slices	2 Slices	2-4 Slices	1-2 Slices	2 Slices

Note to Table 7:
(1) This table applies to Virtex-based products including Virtex, Virtex-E, Virtex-II, Virtex II Pro, Virtex-4, Spartan-II, Spartan-IIE, Spartan-3 and Spartan-3E.

Stratix II vs. Virtex-4 Logic Structure

The innovative and flexible Stratix II ALM structure efficiently implements designs. This section describes the implementation of various common functions, which contribute to the 30% logic efficiency over Virtex-4 architecture.

Wide Fan-In Functions

Stratix II devices have a new, adaptive LUT structure that allows the device to efficiently implement functions larger than four inputs and complex equations with shared inputs. The result of this large-input LUT is higher performance and more efficient logic resource utilization. EDA synthesis tools perform the

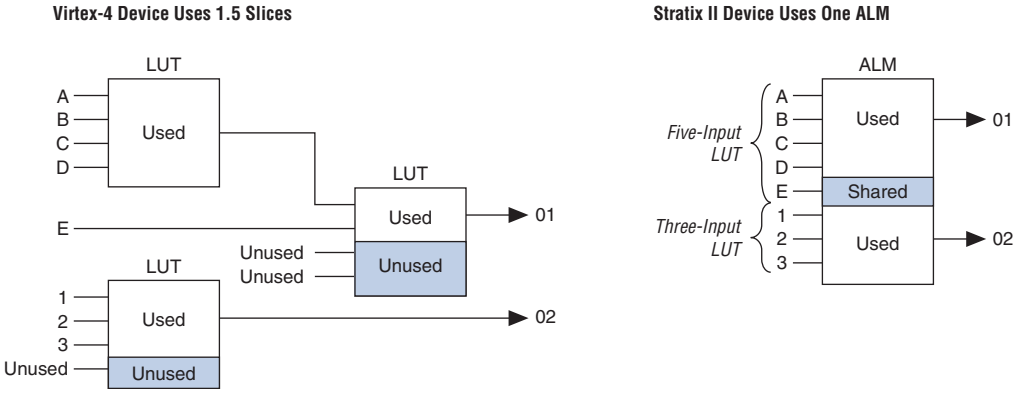
mapping, which is generally transparent to the designer. However, manual optimization is also available for advanced users. The following sections discuss Stratix II configuration modes that provide LUTs with more than four inputs.

Five-Input LUT Sharing

Stratix II ALMs can share inputs between the two ALUTs inside an ALM, allowing one ALM to implement an independent five-input function and an independent three-input function. This feature prevents the rest of the logic in the ALM from being wasted when implementing five-input functions. Each Virtex-4 slice contains two independent four-input LUTs. However, the Virtex-4 LUTs cannot share the input lines. When part of the input is used, the rest of the LUT goes unused.

The example in Figure 5 consists of two independent functions: a five-input LUT and a three-input LUT. In Virtex-4, these functions must be implemented separately in two different slices, with unused inputs wasted. Stratix II devices efficiently map both functions in the same ALM, using the entire LUT for two separate functions.

Figure 5. Stratix II Five-Input LUT Sharing Capability



Six-Input Functions

Each Stratix II ALM can be configured to implement one arbitrary six-input LUT function. The larger logic capacity per LUT allows the Stratix II architecture to map large and complex logic functions more efficiently than other traditional four-input LUT architectures do, including Virtex-4 devices. A Virtex-4 device requires at least two four-input LUTs to implement a six-input function, but typically requires more. The number of four-input LUTs required to implement a six-input function depends on the equation.

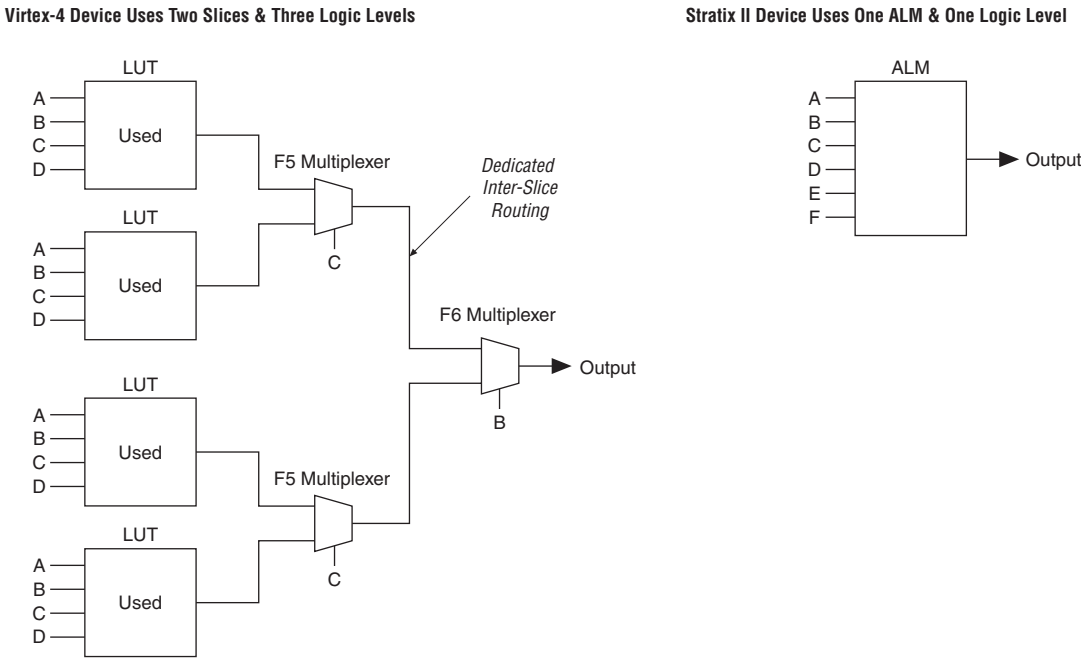
The following equation shows a six-input function example.

$$f(A,B,C,D,E,F) = \overline{ABCDEF} \mid \overline{AB}CD\overline{EF} \mid \overline{A}B\overline{C}DEF \mid \overline{A}BC\overline{D}E\overline{F} \mid ABC\overline{DEF} \mid AB\overline{C}D\overline{EF} \mid A\overline{B}CDE\overline{F} \mid A\overline{BCDEF}$$

Virtex-4 devices require two slices to implement this function, while Stratix II devices only require one ALM for the same function. Stratix II devices not only use less area but are also faster because the entire logic operation is packed inside one ALM without additional routing to stitch LUTs together.

Figure 6 shows a six-input function implemented in both a Stratix II device and a Virtex-4 device.

Figure 6. Six-Input Function Comparison between Stratix II & Virtex-4 Devices



Another example of efficient six-input LUT usage is the crossbar switch as discussed in the "Multiplexer & Crossbar Switch" section.

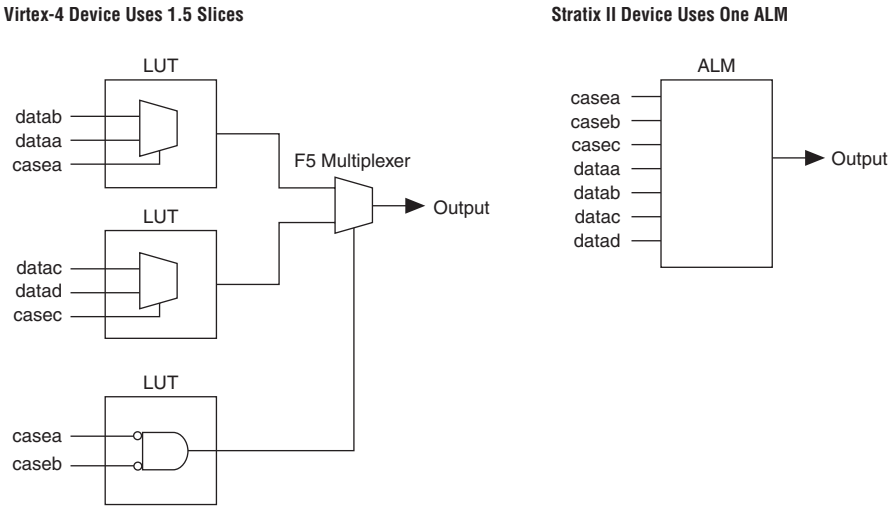
Seven-Input Functions

Stratix II ALMs can also be configured to implement certain seven-input functions if some of the inputs are shared. One example is a cascaded if-then-else function that is commonly found in a typical design:

```
begin
      if (casea)
            out = dataa;
      else if (caseb)
            out = datab;
      else if (casec)
            out = datac;
      else out = datad;
end
```

Virtex-4 devices require 1.5 slices and one 2-to-1 multiplexer to implement this if-else statement. Stratix II devices can implement this function in one ALM, taking advantage of the shared seven-input LUT mode in the ALM. This ALM capability provides smaller area usage and faster performance. Figure 7 shows the block diagram implementation in Stratix II and Virtex-4 devices.

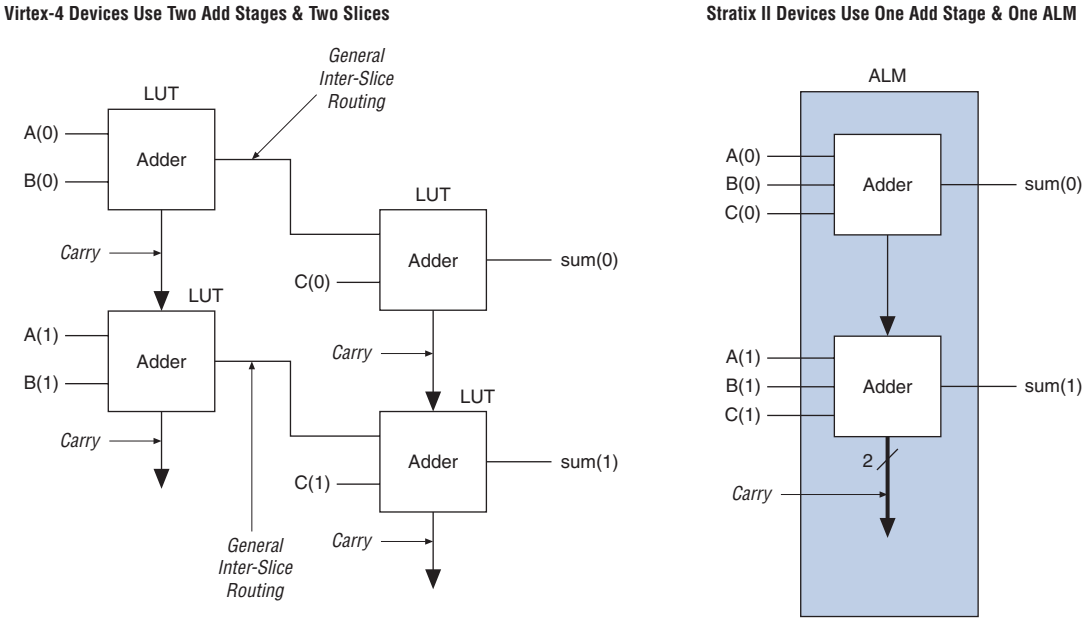
Figure 7. Seven-Input Function Comparison between Stratix II & Virtex-4 Devices



Adder Tree

Adder trees are often found in DSP applications, such as the correlators in 3G wireless base station channel cards or the partial-product summation in logic-based multipliers. Stratix II ALMs offer a performance and logic efficiency advantage in adder tree implementations by supporting the summation of three 1-bit numbers in one step. The Virtex-4 logic structure only supports two-input add operations. Figure 8 shows the Stratix II device and Virtex-4 device implementations of a simple two-bit, three-input adder.

Figure 8. Three-Input Adder Comparison Between Stratix II & Virtex-4 Devices



When a three-input adder is implemented in a Virtex-4 device, the device generates the sum by repeatedly adding two numbers each time. This binary-tree style summation uses a lot of logic resources. The performance of this binary-tree style summation is limited by the delay of each add stage in addition and the delay of the programmable routing between each add stage.

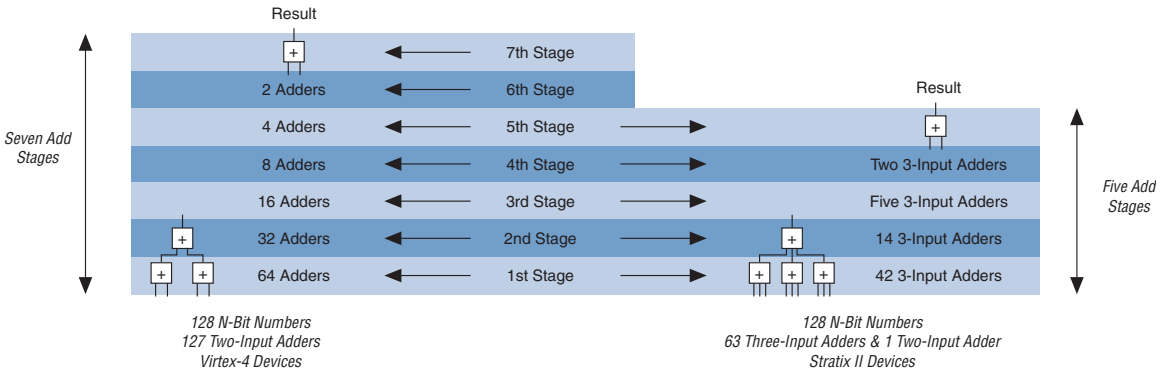
The three-input adder support (in shared arithmetic mode) in Stratix II ALMs can collapse the two add stages into one, improving the area usage. The performance is also improved by eliminating one add stage and the extra programmable routing. The number of add stage reductions can be estimated through the following equation.

$$\text{Add Stage Reduction} = \text{ceiling}(\log_2(n)) - \text{ceiling}(\log_3(n))$$

where n is the total number of input numbers

When the number of adder tree inputs increases, the number of summation stages increases. Stratix II devices provide area savings and performance gains by adding three numbers at once in the ALMs. Figure 9 shows the binary implementation of a 128-number adder tree. Implementing this design in a Stratix II device reduces the total height of the tree by two add stages and improves performance by reducing both the logic propagation and routing delay.

Figure 9. 128-Number, 16-Bit per Number Adder Tree Comparison between Stratix II & Virtex-4 Devices Notes (1), (2), (3)



Notes to Figure 8:
(1) To simplify the logic utilization calculation, the size of all adders in each add stage is assumed to be N-bit wide.
(2) The 127th and 128th input numbers in the Stratix II implementation are fed into the third and fourth add stage, respectively.
(3) This figure only shows a 1-bit operation. Each number can be *n*-bits wide.

Table 8 compares the logic resource utilization of a 128-number, 16-bit adder tree in both Stratix II devices and Virtex-4 devices.

Table 8. 128-Number, 16-Bit per Number Adder Tree Comparison between Stratix II & Virtex-4 Devices

Design	Logic Utilization	
	Stratix II	Virtex-4
Pipeline adder tree	780 ALMs	1,192 slices
Non-pipeline adder tree	605 ALMs	1,080 slices

Complex Arithmetic with Logic Function Generator

Simple arithmetic operations such as additions and subtractions are common in designs. These operations often have associated logic functions in front of them. Consider the following function as an example:

`If select=0, then A +/- B, else A +/- C`
A, *B*, and *C* are all n-bit numbers, and `+/-` is controlled by the `addsub` signal.

For each bit of data, Virtex-4 devices require one LUT to multiplex the *B* and *C* inputs based on the *select* line. Then, another LUT adds or subtracts *A* to or from the chosen input. One Stratix II ALM can perform both the multiplexing and adding, or multiplexing and subtracting, for two bits. Therefore, the Stratix II architecture achieves a reduction in logic levels over Virtex-4 devices. See Figure 10.

Figure 10. 8-Bit Data-Select-Before-Addition/Subtraction Comparison between Stratix II & Virtex-4 Devices

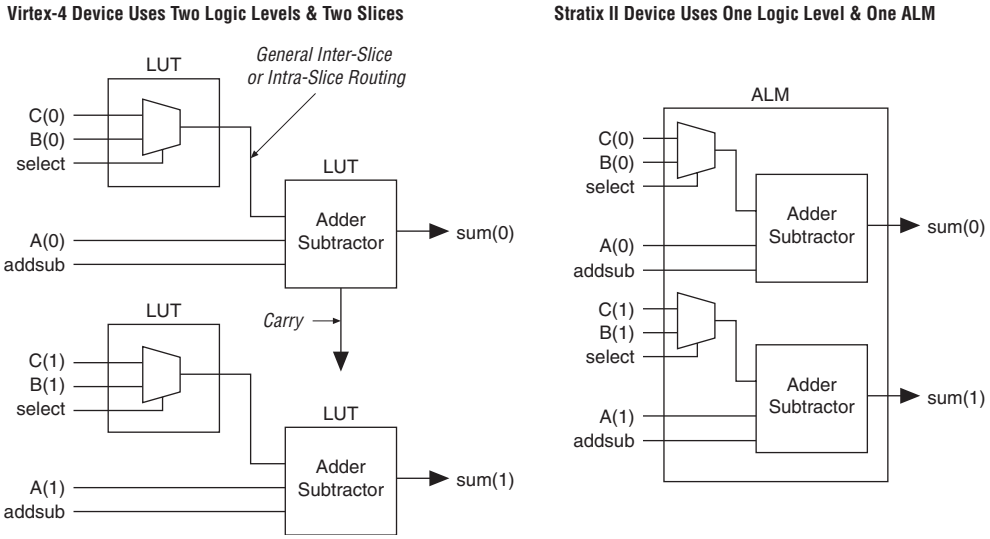


Table 9 shows the performance comparison of an 8-bit data-select-before-addition/subtraction between Virtex-4 and Stratix II devices.

Table 9. 8-Bit Data-Select-Before-Addition/Subtraction Comparison between Stratix II & Virtex-4 Devices

Design	Logic Utilization	
	Stratix II	Virtex-4
8-bit data-select-before-addition/subtraction	4.5 ALMs	8.5 slices

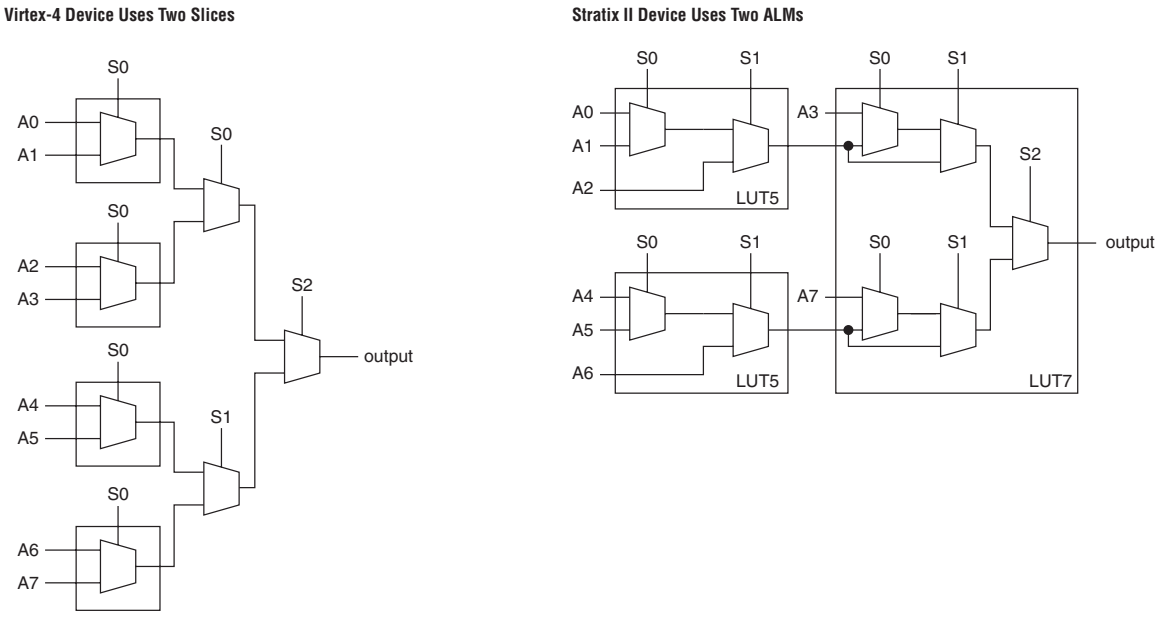
Multiplexer & Crossbar Switch

Virtex-4 device slices have embedded 2-to-1 multiplexers, called F5, F6, F7, and F8 multiplexers. These multiplexers can reduce the logic resource usage for building multiplexers larger than 4-to-1 or some decoding functions.

The Stratix II ALM has a more flexible input-output relationship than the Virtex-4 architecture. The Stratix II device architecture can efficiently implement large multiplexers, barrel shifters, crossbar switches, decoders, and state machines. Stratix II devices consume less logic resources than Virtex-4 devices and produce higher f_{MAX}.

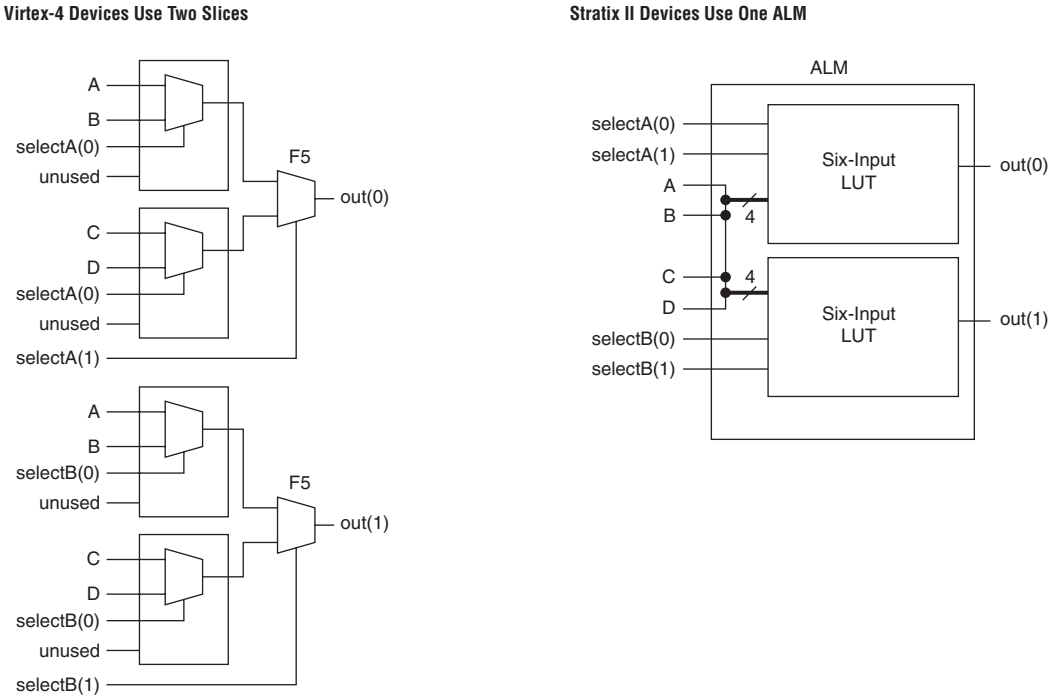
The following is a comparison of an 8-to-1 multiplexer between Virtex-4 devices and Stratix II devices. The Virtex-4 architecture uses the embedded multiplexer in the slices to build the multiplexers. The 8-to-1 multiplexer uses two stitched slices. Stratix II devices use two ALMs, using the flexible five-input and seven-input LUT configurations to encompass all 11 total input lines.

Figure 11. 8-to-1 Multiplexer Implementation Comparison Between Stratix II & Virtex-4 Devices



Although the synthesis of a multiplexer consumes about the same logic count, Stratix II devices can implement a complex multiplexer, such as a crossbar switch, more efficiently. For example, a 4 × 2 crossbar switch can be implemented in Stratix II technology by using one ALM. However, the same design would take two slices in Virtex-4 devices. See Figure 12.

Figure 12. 4 × 2 Crossbar Switch Implementation Comparison Between Stratix II & Virtex-4 Devices



Crossbar switch functions normally have multiple outputs and require multiple select lines from the same inputs. In traditional four-input LUT architectures, the function generator for the multiple output lines must be replicated. Even though Virtex-4 devices have embedded multiplexers that help pack the function into one slice, the device still requires two slices to cover the same function for a different set of select and output lines. Stratix II ALMs can pack two functions in a single ALM because the two six-input LUTs implement the same function and have several common inputs. Table 10 lists the resources required for an 8-to-1 multiplexer and a 4 × 2 crossbar switch.

Table 10. 4 × 2 Crossbar Switch Comparison between Stratix II & Virtex-4 Devices

Design	Logic Utilization	
	Stratix II	Virtex-4
8-to-1 multiplexer	Two ALMs	Two slices
4 × 2 crossbar switch	One ALM	Two slices

Shift Registers

Many designs use shift registers, and Altera recognizes their importance by providing efficient architecture support for shift register implementations.

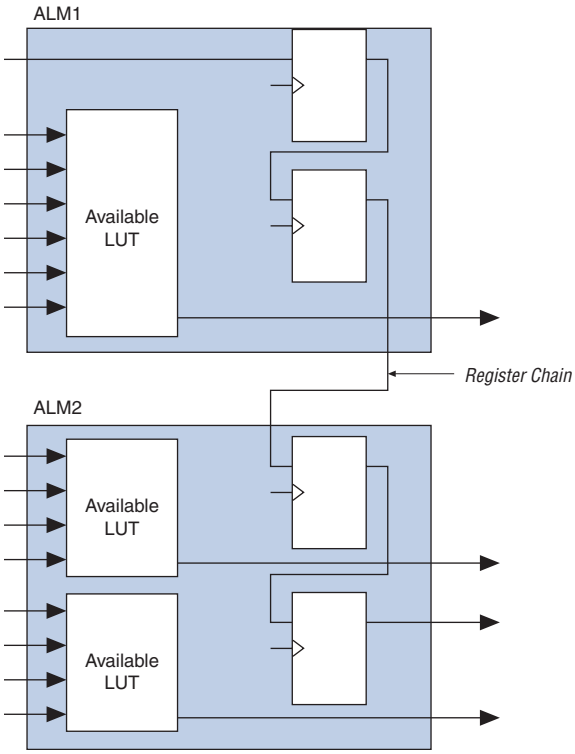
Stratix II FPGAs use an efficient shift register implementation through one of the following:

- M512 or M4K RAM blocks for wide or deep shift registers
- Dedicated ALM register chain for small shift registers

Most shift registers operate on data buses and are very wide. These shift registers can be efficiently mapped into the M512 RAM blocks with minimal logic usage. Each M512 block can contain up to 576 bits of FIFO buffer or RAM data, and each Stratix II device has up to 930 M512 blocks.

For small shift registers, Stratix II devices have dedicated register chains that connect the ALM registers without using the ALM combinational logic block. Therefore, when the register is used, the combinational logic block is still available for any arbitrary logic functions and can have an independent combinational output. This capability is called register packing. As a result, small shift registers effectively consume zero logic in Stratix II devices. See Figure 13.

Figure 13. Stratix II Register Chain Example



Each Virtex-4 LUT can be configured into a 16-bit shift register, called SRL16. When the LUT is configured as an SRL16 shift register, the LUT is no longer available as logic, so the logic resource utilization increases with SRL16 usage. Wide, deep, and small shift registers are implemented the same way in Virtex-4 devices using SRL16 shift registers.

Table 11 shows the different configurations of shift registers and their respective logic utilization in Stratix II and Virtex-4 devices.

Table 11. Shift Register Utilization Comparison between Stratix II & Virtex-4 Devices

Design	Logic Utilization	
	Stratix II	Virtex-4
18-bit wide x 32 deep	One M512 block and six ALMs	18 slices
1-bit wide x 4 deep	Four registers, 0 ALM	0.5 slices

Barrel Shifter

Barrel shifters are needed whenever the beginning of the received data is not known. For example, a SONET Framer repeatedly performs a search and compare (with the A1A2 code) until it finds the A1A2 code, which provides the start of the frame location.

Barrel shifters are multiplexers. Due to the efficient mapping of the six-input and seven-input LUTs in Stratix II ALMs, barrel shifters in Stratix II devices use fewer logic resources and run faster than Virtex-4 devices. Table 12 displays the logic usage of a Stratix II barrel shifter compared to that of a barrel shifter in Virtex-4.

Table 12. Barrel Shifter Comparison between Stratix II & Virtex-4 Devices

Design	Logic Utilization	
	Stratix II	Virtex-4
16-bit barrel shifter	19 ALMs	34 slices

Conclusion

Stratix II adaptive logic architecture provides a flexible and powerful logic mapping, which produces highly efficient logic use. Stratix II ALMs can perform six-input functions and some seven-input functions, have the capability of shared input LUTs, and contain an embedded ternary adder circuit. These features enable efficient implementation of many common functions, such as complex arithmetic, multiplexers, crossbar switches, barrel shifters, or wide fan-in functions.

Based on real-world design benchmarking, a single Stratix II ALM contains 30% more logic capacity than a Xilinx Virtex-4 slice. As a result, Stratix II devices are larger and more logic efficient than Virtex-4.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com