WP161 (v1.0) June 17, 2002

Comparing Virtex-II and Stratix Logic Utilization

By: Frederic Rivoallon

New programmable logic circuits provide an ever-growing set of architectural elements. Advanced FPGAs are not solely made of look-up tables and flip-flops anymore, and today's logic fabrics are best described as “feature rich.” This trend requires sophisticated algorithms in both synthesis and implementation tools to provide optimal performance and logic utilization by leveraging these new hardware features. This document highlights how the Xilinx Virtex™-II family provides 25% better logic utilization compared to Altera®’s Stratix™ device family as a result of fabric features and advanced software algorithms.

The Logic Fabric

Building Blocks

The basic logic element in Stratix is called a “Logic Element” or “LE.” It is composed of a 4-input look-up table (LUT), a flip-flop, and carry-chain circuitry for arithmetic functions. The basic logic element in a Virtex-II device is composed of the same elements as the Stratix LE as well as additional functionality, such as a function expander (MUXF) and an arithmetic cell (MULT_AND). Furthermore, the Virtex LUT is usable as a 16-bit memory element or as a 16-bit shift register. These additional logic elements in Virtex-II devices significantly improve logic utilization and design performance as described in the following paragraphs.

MUXF

The function expander cell represents a 2:1 mux. It can be used to build wide multiplexers. For example, using the MUXF, only four LUTs are required to implement an 8:1 mux. The function expander is not limited to multiplexers and can be used for many other logic functions. For example, a MUXF combined with two LUTs can implement any function of five inputs, thereby, implementing a full 5-input LUT. Another example (in Figure 1) shows a 9-input function mapped onto two LUTs, plus one function expander for the Virtex-II device, whereas the same function requires three LUTs for Stratix (which equates to three LEs).

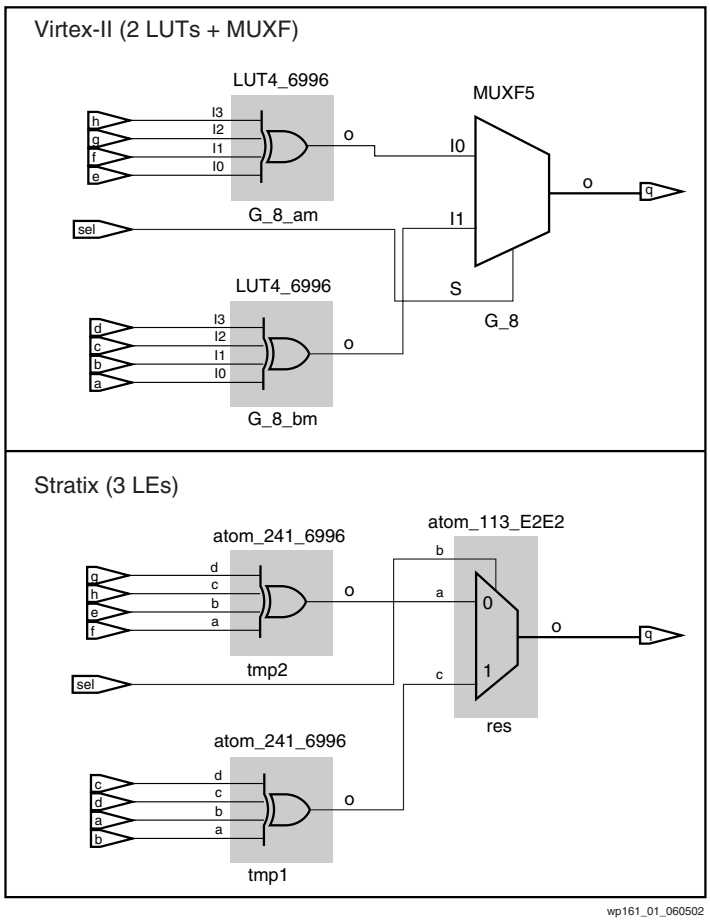

Figure 1: **9-Input Function Mapped to Virtex-II and Stratix Devices**

MULT_AND

The arithmetic cell, MULT_AND, reduces the amount of logic necessary to implement a multiplier along with LUTs (when dedicated multipliers are not used). In addition, some 5-input functions can be mapped onto a single LUT using the MUL_AND cell and the carry chain. For example, loadable up and down counters implemented using the MULT_AND cell utilize only one LUT per bit instead of two LUTs per bit as in Stratix. Figure 2 shows a loadable counter mapped to a Virtex-II and a Stratix device:

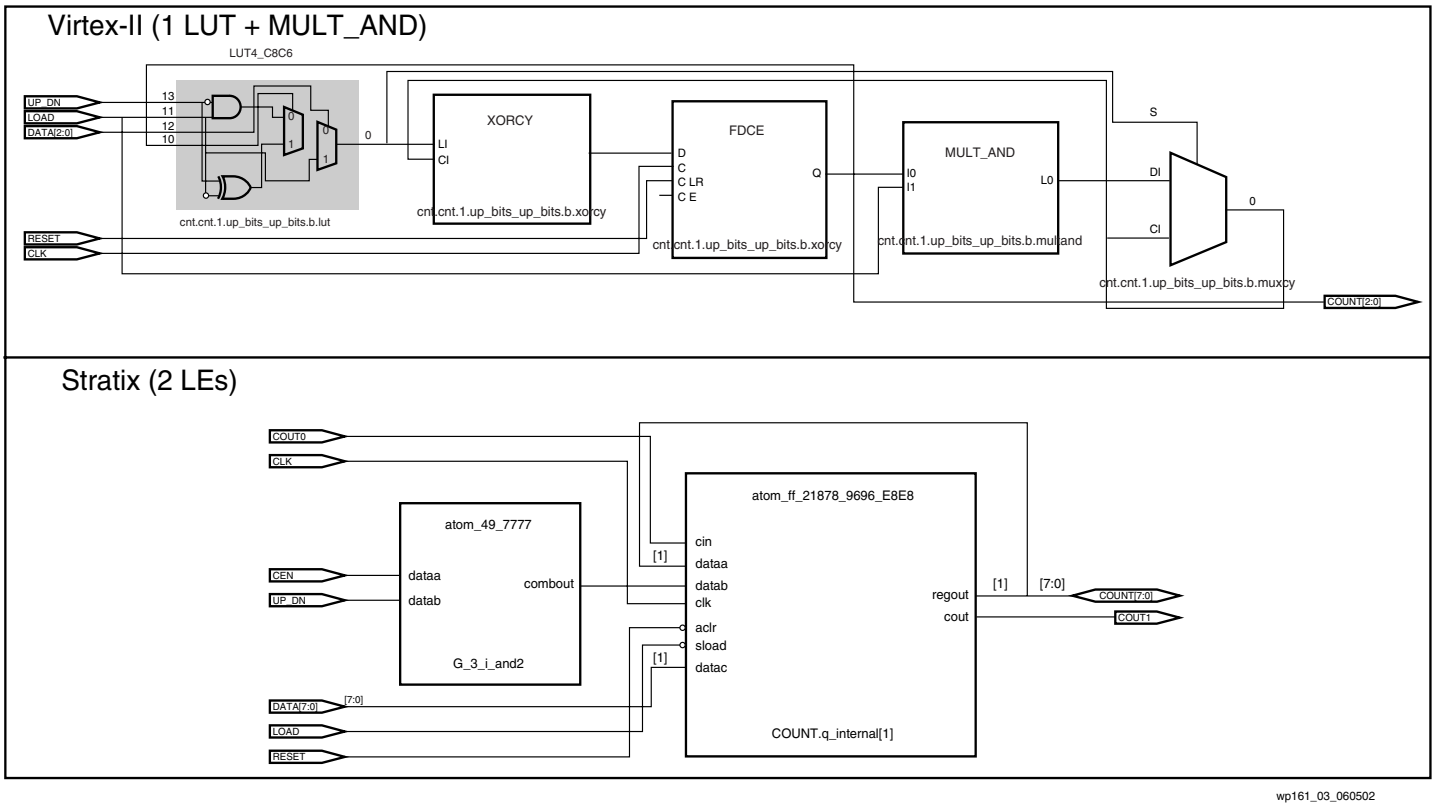

Figure 2: **Loadable Up/down Counter Mapped to a Virtex-II Device and a Stratix Device**

RAM

The RAM mode capability of the LUT is ideal for small memories and saves considerable amount of logic for register file implementations that would otherwise consume LUTs and flip-flops. True asynchronous reading of the memory is also available, providing a flexible storage element suitable to various HDL coding styles.

SRL

In its shift register mode (SRL), the LUT implements very efficient fixed delay lines or elastic buffers without the need of a dedicated memory block. For example, in a Virtex-II design, a 16-bit shift register is implemented with one LUT and one flip-flop, whereas the same shift register in Stratix would take up to 16 LEs or a memory block.

Figure 3 shows a shift register mapped to a Virtex-II device and a Stratix device.

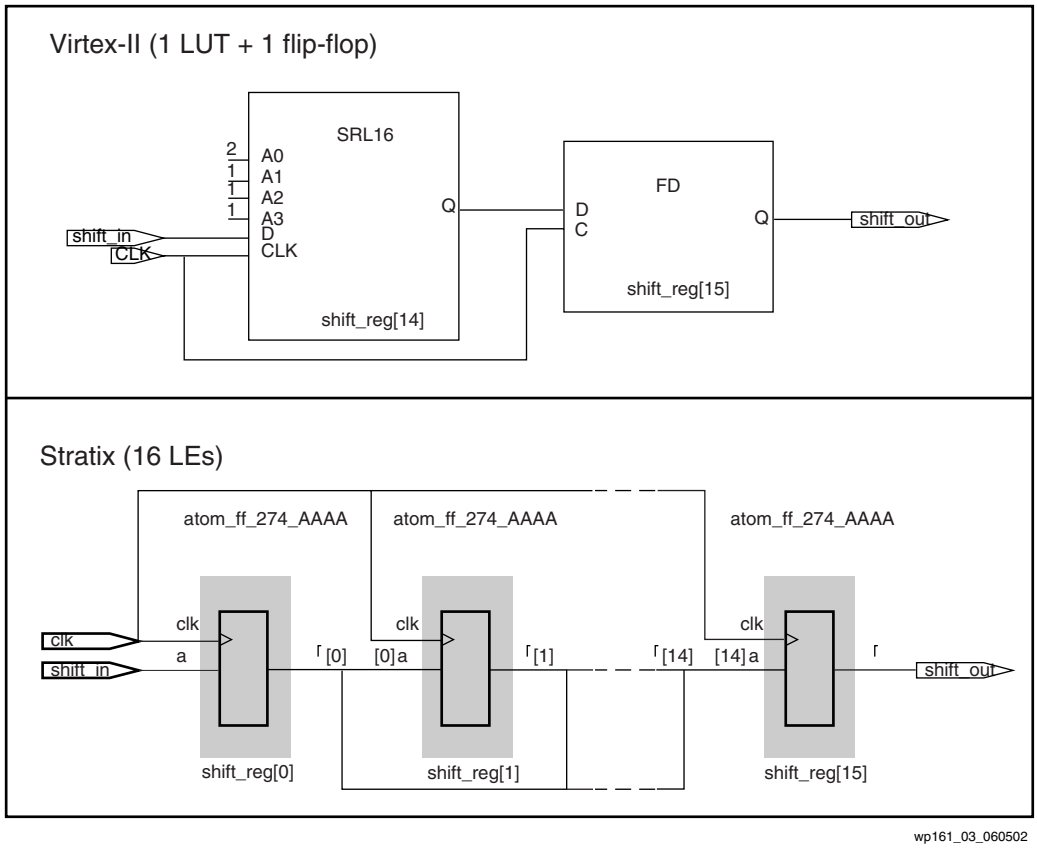

Figure 3: **A 16-Bit Shift Register Mapped to a Virtex-II Device and a Stratix Device**

Logic Connectivity

In Virtex-II fabric, the LUT and flip-flop can be used independently without restrictions. In Stratix devices, a LUT cannot be used with its flip-flop in all circumstances, because one input pin of the LUT is shared with the path that has direct access to the flip-flop. By default, when the flip-flop is not fed by any logic, the LUT in that LE is unavailable to the rest of the design. As a remedy, Quartus® II tools provides a register packing option (off by default) to enable the packing of LUTs along with the flip-flop. This still does not allow LUTs using 4-inputs to be packed because the connectivity restriction is still present. Figure 4 shows LUT to flip-flop connectivity in both Virtex-II and Stratix devices.

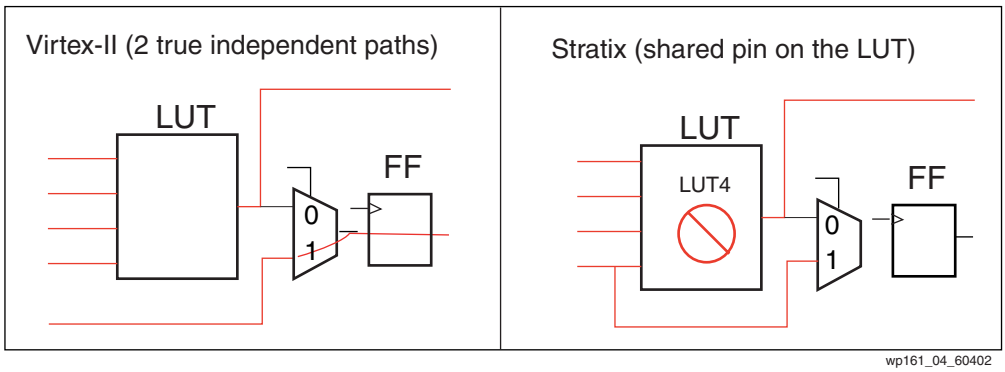

Figure 4: **Connectivity LUT to Flip-Flop in Virtex-II and Stratix Devices**

Stratix devices provide a register chaining option that enables chains of flip-flops to be placed into a LAB (a unit of 10 LEs) with LEs that are not using flip-flops. The scope of this feature is limited to shift register functions only. This feature also implies a placement constraint where all the LUTs within a single LAB would not use their flip-flops.

Customer Design Benchmarks

A suite of 40 customer designs has been synthesized and implemented for Virtex-II and Stratix. Results indicate that Virtex-II logic has 25% better utilization than Stratix. The contribution of the logic fabric building blocks and logic connectivity is 12% and 13%, respectively.

Building Blocks

Eighty-seven percent of the designs considered use the above mentioned Virtex-II cells (MUXF, MULT_AND, SRL). As a result, 12% fewer LUTs in average were required to implement these designs than Stratix.

Logic Connectivity

Stratix uses an additional 13% more LUTs in average than Virtex-II logic just to route signals to the flop in the LE. In one instance, over 47% of the LUTs are wasted (see Figure 5). These additional LUTs are reported on the "uses cells as routing" line at the end of the Synplify report (targeting Stratix):

```
...
Total LUTs:  37067 of 79040 (46%)
Logic resources:  54727 ATOMs of 79040 (69%)
...
  with no input combinational logic: 17660 (uses cell for routing)
...
```

Figure 5: **Excerpt from Synplify PRO Report File .srr Targeting Stratix**

Implementation with ISE and Quartus II Tools

Xilinx ISE development tools separate unrelated functions and assign them to different clusters (called a slice) on the fabric. This avoids conflicting placement constraint and guarantees optimal performance. As the device gets full, powerful algorithms pack unrelated logic into common clusters. This gradual process ensures that the device is utilized at its best, with minimal impact to design performance.

In the Altera development tools, Quartus II v2.0 sp1, the packing of logic requires a special option. With this option turned on, packing is limited because an unrelated LUT using its 4-inputs and a flip-flop cannot be merged together in any "LE," and the limited packing comes at a cost to design performance. By enabling register packing in Quartus II for Stratix, Virtex-II logic utilization was 15% better than Stratix.

Conclusion

The Virtex-II advanced building block features and superior logic connectivity, together with advanced synthesis and software mapping algorithms that perform automatic packing lead to 25% better logic utilization compared to Stratix.

Table 1 compares the density of equivalent Stratix and Virtex-II devices This table enables users to determine the size of the programmable device required:

Table 1: **Density Comparison in Virtex-II and Stratix Devices**

Virtex-II	Equivalent LE/ Logic Elements	Stratix
	10,570	EP1S10
XC2V1000	12,800	
	18,460	EP1S20
XC2V2000	19,200	
	25,660	EP1S25
XC2V2000	26,880	
	32,470	EP1S30
XC2V3000	38,840	
	41,250	EP1S40
	57,120	EP1S60
XC2V4000	57,600	
	79,040	EP1S80
XC2V6000	84,480	
	114,140	EP1S120
XC2V8000	116,480	

Notes:

1. The equivalent number of LEs for Virtex-II devices is obtained by applying a coefficient of 1.25 to the number of LUTs in the device. The XC2V1000 contains 10,240 LUTs, the XC2V2000 21,504 LUTs, the XC2V3000 28,672 LUTs, the XC2V4000 46,080 LUTs, the XC2V6000 67,584 LUTs, the XC2V8000 93,184 LUTs.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06 / 17 / 02	1.0	Initial Xilinx release.