A Fast Constant Coefficient Multiplier for the XC6200

XAPP 082 August 24, 1997 (Version 1.0)

Summary

This application note presents a high performance constant coefficient multiplier for the Xilinx XC6200 FPGA. The design provides performance and density by using dynamic reconfiguration, allowing changes to coefficients without the need for complete reconfiguration.

Xilinx Family

- XC6200

Demonstrates

- A constant coefficient multiplier.
- Efficient XC6200 implementation
- Use of dynamic reconfiguration.

Table of Contents

Introduction

The Xilinx XC6200 is the first commercial FPGA designed to specifically work in close cooperation with a microprocessor. The microprocessor sees the XC6200 as a block of RAM, within which the user designed registers and configuration data appear. The data bus width is programmably selectable to 8, 16 or 32 bits. A feature of the XC6200 that reduces configuration time is that it supports the simultaneous writing of many words in configuration memory with the same data. An additional feature is the ability to group non-adjacent registers within the user design into a single word for transfer to and from the processor.

The XACTStep Series 6000 CAD software package, together with a PCI bus resident development system [2], is used in the development and debugging of applications for the XC6200.

Constant Coefficient Multiplication

Constant coefficient multiplication is a common function in many Digital Signal Processing (DSP) applications. One such application is Finite Impulse Response (FIR) filtering. With FIR, the parameters to the filter are changed much less frequently than the input data values, hence constant coefficients can be used to represent them.

The multiplier described here takes an 8-bit input value and multiplies it by a constant 8-bit coefficient to get a 16-bit unsigned result.

An efficient way of implementing this fixed coefficient multiplier is by using a Look Up Table (LUT) based FPGA[5]. The design works by substituting the networks of adders with lookup tables to perform most of the multiplication. Consider the multiplication of a 4-bit value and an 8-bit constant; as there are only 16 different values that can be represented by 4-bits, there can only be 16 possible answers. Thus any multiplication between a four-bit variable and an eight-bit constant can be implemented by a 16 entry lookup table. With a maximum possible result of 3,825 in an unsigned 4 by 8-bit multiplication, each LUT entry must be large enough to accommodate a 12-bit result. Therefore, both the entry elements and the output of the LUT should be at least 12-bits wide.

For an 8-bit by 8-bit constant multiplier, two of these 4-bit by 8-bit multipliers can be arranged as shown in Figure 1. The mathematics behind this arrangement are shown in Figure 2.

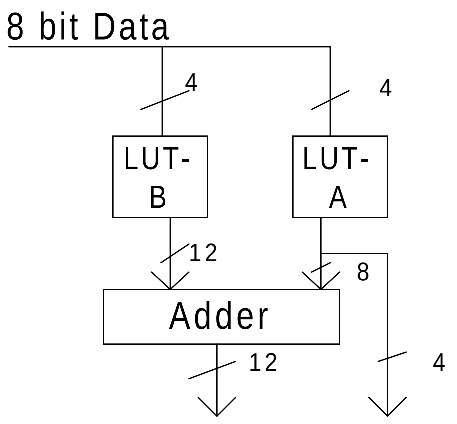

Figure 1: LUT Based Multiplier

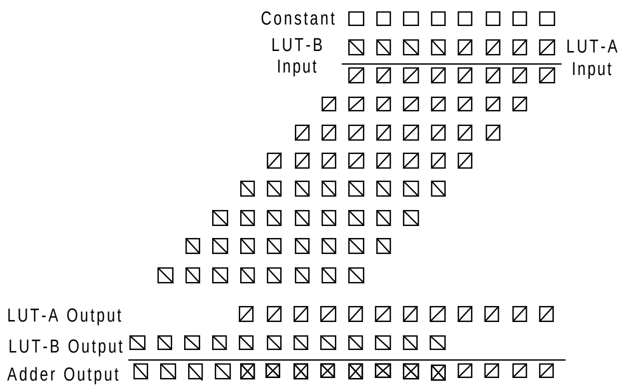

Figure 2: Maths of LUT Based Multiplier

Physical Implementation

The hardware components required to implement the 8x8-bit multiplier are shown in Figure 3. Pipeline registers have been added to both the LUT's and the adders to increase the throughput.

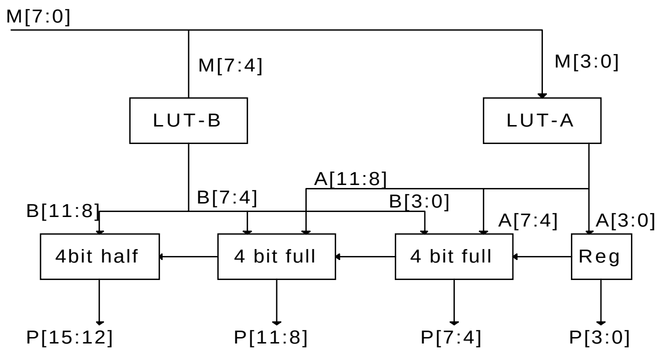

Figure 3: Multiplier Architecture

The XC6200, unlike the XC4000E, does not use LUT's to implement logic functions. Instead, its function units are built around 2:1 multiplexers. These function units can implement a register and either a 2:1 multiplexer or a combinational 2-input logic gate. Being able to implement both a register and a logic function means that the cost of pipelining in this technology is very low.

A 4-input lookup table can be implemented, as shown in Figure 4, using 2:1 multiplexers and registers.

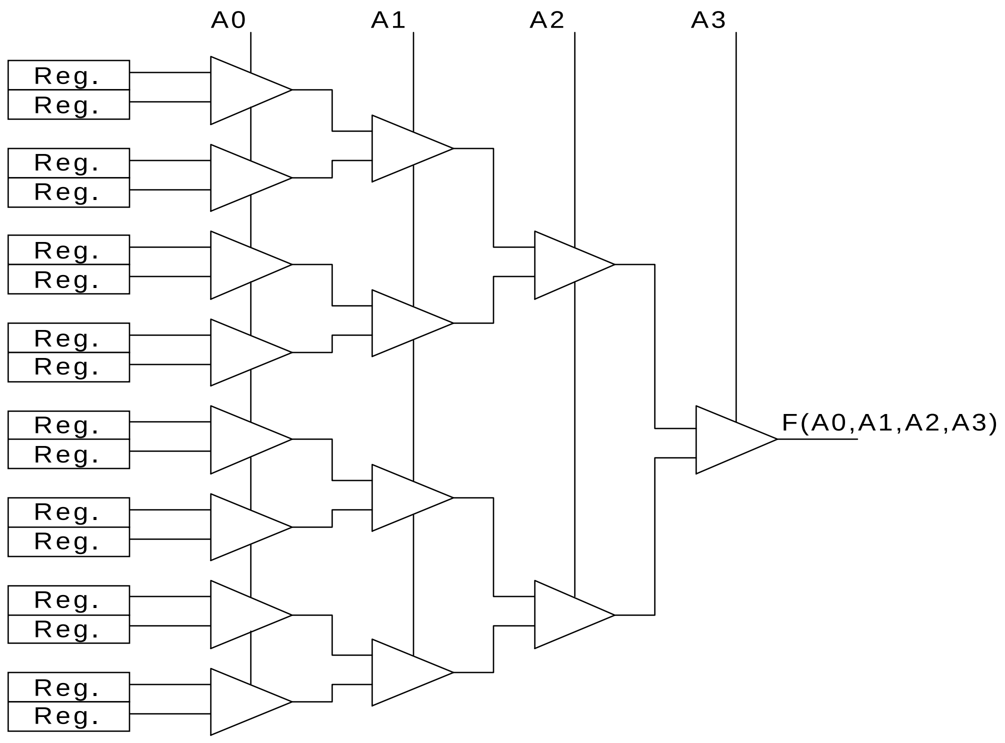

Figure 4: Building LUT's from Muxes

An improvement of this approach implements LUTs by replacing the 16 user registers with four 2-input gate functions as shown in Figure 5. Each of the gates require four bits of configuration memory to encode a logic gate truth table; so instead of user registers, configuration memory is used.

In a complete LUT, the individual 4-input LUT's are laid out in single rows of 8 cells; these are then stacked vertically to form the 12-bit output. This extremely regular layout, as shown in Figure 6, makes it easy to determine which parts of the circuit need to be reconfigured at run time. It is worth noting that because of the pipelining scheme some horizontal rows of cells, those corresponding to the most significant bits, have additional registers in them.

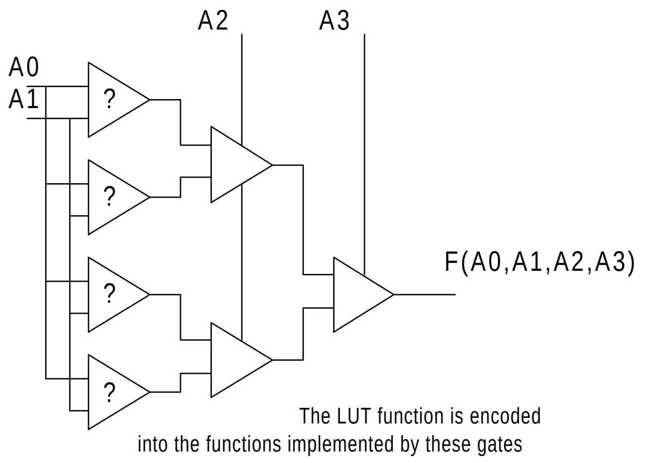
A2
A3
A0
A1
?
F(A0,A1,A2,A3)
The LUT function is encoded
into the functions implemented by these gates

Figure 5: Optimised LUT Construction

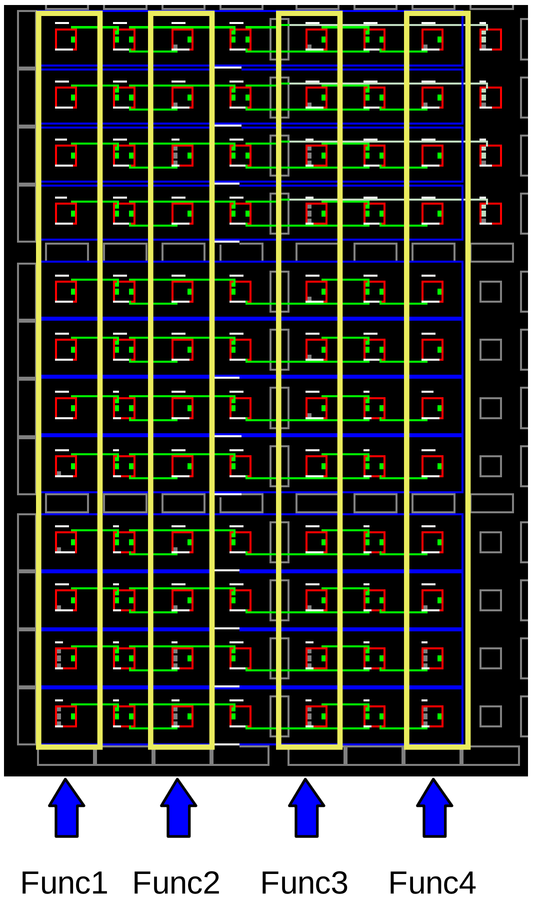

Figure 6: LUT layout on XC6200

Changing Coefficients

Figure 6 shows the layout of the lookup table on the XC6200. The XC6216 memory map is designed so that the control bits for functionally related resources on the chip occur in logically adjacent bits and words of control memory.

The vertical columns, labeled Func1 through Func4 (see Figure 6), contain the 4 gates whose functions are changed according to the lookup table. To change the function of a gate, up to 12 cells need to be reconfigured. The control bits for each of these cells require only one byte, thus a 32-bit data bus can simultaneously write to 4 vertically adjacent cells in one cycle.

A coefficient change, on a multiplier consisting of two LUTs, would require up to 96 cells to be reconfigured. Therefore it would take 24 write cycles, or 960 nano seconds with a 50Mhz clock, to completely reprogram the multiplier.

It is worth bearing in mind that, in practice, coefficient changes may not always require a complete reprogramming of the circuit; in fact, only a few cells may need reconfiguration. This would make reconfiguring the multiplier, on average, much faster.

Multiplier Performance

The multiplier runs at 75MHz using the *-2* speed grade. The equivalent design on the XC4000E at grade *-3* runs at 65Mhz. (Note that the speed grade numbering system is different.)

Cost-wise, the design uses 280 cells on the XC6200 and 25 CLB's on the XC4000E. The most significant advantage is the speed at which its coefficients can be changed: the XC6200 reconfigures in less than 1µs.

The Dedicated Interface

On both these devices, LUT configuration data can be changed without reconfiguring the chip. This is achieved by allowing user logic to address LUT RAM. On the XC4000E this technique requires a significant amount of additional logic, which must be added to the user design to control the LUT RAM and to interface with the external source of data (e.g. a microprocessor). This results in routing resources being tied up between the IOB's and the LUT RAM control logic.

Unlike the XC4000E, the XC6200 has a built-in dedicated processor interface. This interface includes the necessary extra circuitry to perform this function without tying up routing resources.

Increasing Density

The layout described above is designed to preserve regularity in order to make dynamic reconfiguration to change coefficients fast and efficient. If speed of reconfiguration can be sacrificed however, the design can be implemented in a much smaller area. This is done by co-optimizing the logic in the groups of 12 shared input LUT's.

One way of achieving this would be to exploit the fact that the first two LUT input variables require only 5 unique functions. Also, of the 16 possible functions, the second eight are the logical inverses of the first; this means that the first eight functions can be implemented simply by inverting the corresponding multiplexer input in the second level of the tree. The remaining eight can then be further reduced to five by observing that three of the functions - ZERO, A0 and A1 - are trivial and therefore do not require logic gates. Thus we need only calculate the 5 non trivial functions and route these, alongside the two input variables, to the 12 sets of muxes to build the LUT.

These changes almost half the area used by an XC6216 LUT: only *3*12+5=41* cells are used, as opposed to *7*12=84* cells in the previous design. The downside of this is that a different route pattern must be created for each possible LUT; this leads to longer reconfiguration times. The number of gates used can be further reduced to 30 by using complex logic optimization algorithms, but these significantly complicate the routing.

Summary

This application note gave a design on the XC6200 fine-grained multiplexer based FPGA for a constant coefficient multiplier. The design provides better performance and greater density than multipliers on conventional FPGAs.

In addition the ability of the XC6200 architecure to dynamically reconfigure small portions of the device very quickly makes the look up table based technique much more attractive since coefficients or other parameters stored in the lookup tables can be changed rapidly. The configuration values required for a particular coefficient can be calculated rapidly on the fly by a microprocessor - without the need for complex CAD tools.

References

[1] Xilinx Inc, "XC6200 FPGA Family Advanced Product Description", Available from Xilinx Inc. 2100 Logic Drive San Jose CA.

[2] Wayne Luk and Nabeel Shirazi, "Modelling and Optimising Run Time Reconfigurable Systems", Proc. IEEE Symposium on FPGA's for Custom Computing Machines, Napa CA 1996.

[3] Tom Kean, "Configurable Logic: A Dynamically Programmable Cellular Architecture and its VLSI Implementation" Phd Thesis CST-62-89, University of Edinburgh, Dept. Computer Science.

[4] Tom Kean and John Gray, "Configurable Hardware: A New Paradigm for Computation", Advanced Research in VLSI, Proc. Decennial Caltech Conference, MIT Press 1989.

[5] Ken Chapman, "Fast Integer Multipiers fit in FPGA's", EDN 1993 Design Idea Winner, EDN May 12th 1994.

Limitations And Restrictions