An FPGA Technologies Area Examination of the SHA-3 Hash Candidate Implementations

Brian Baldwin and William P. Marnane

Claude Shannon Institute for Discrete Mathematics, Coding and Cryptography.
Department of Electrical & Electronic Engineering,
University College Cork, Cork, IRELAND
Email: {brianb,liam}@eleceng.ucc.ie

Abstract. This paper presents an examination of the different FPGA architectures used to implement the various hash function candidates for the currently ongoing NIST-organised SHA-3 competition [1]. This paper is meant to be used as both a quick reference guide used in conjunction with the results table [2] as an aid in finding the best-fit FPGA for a particular algorithm, as well as a helpful guide for explaining the many different terms and measurement units used in the various FPGA packages.

1 Introduction

Round two of the NIST-run SHA-3 hash function competition [1] comprises fourteen different design competitors. NIST have stated that round two will include a hardware analysis. Attempts are currently being made to form a standard benchmark and a website was developed in conjunction with the SHA-3 zoo [3], to record and tabulate all the different designs, their methodologies and the tools used in the design.

Difficulties arise in comparison analysis when one considers that there are fourteen different designs and three different design methodologies: core functionality, full implementation and full implementation with external memory. Each of these can be further broken down into high-speed and low-area designs (and everything in between). Added to this is the wide range of different FPGA devices used in the implementation, each of which has different underlying technology and different standards of measurements both between different vendors and in some cases between different families of the same vendor, thus making any form of comparison extremely difficult.

For the purpose of this exercise, we assume the timing and speed grades of the different FPGAs are chosen for the fastest speed grade (which is both package and device dependent) as this value is not explicitly given for some of the implementations.

As FPGA hardware results produced by the different groups each take their area and timing measurements on different target platforms, we attempt to provide some form of reference for comparing the different hash function area results

on different FPGA technologies. Results for a particular hash function architecture on different FPGA platforms cannot be directly compared, since any two platforms have different underlying technologies. However, by looking at similarities between the different devices we try to define some standard measurement for a "ball-park" estimate.

2 FPGA Architecture

A Field Programmable Gate Array (FPGA) is an integrated circuit which is user-programmable, as opposed to an Application Specific Integrated Circuit (ASIC), which is customised by the manufacturer for a particular use. FPGAs are an attractive choice for implementing cryptographic algorithms, because of their low cost relative to ASICs. FPGAs are flexible when adopting security protocol upgrades, as they can be re-programmed in-place, and FPGAs also allow rapid prototyping of designs. The downsides however are they are larger in area and higher in power usage when compared to ASICs.

An FPGA can be described as an array of configurable logic blocks and interconnects, all of which can be programmed by the user to describe combinational logic circuitry. The components and connections which make up these logic blocks however, vary to a greater or lesser degree between different manufacturers, and even between different families of the same manufacturer.

Xilinx [4] and Altera [5] FPGAs are the two FPGA products used for the implementations of the hash functions. The two basic measurement standards of an FPGA in the case of a Xilinx device are configurable logic blocks (CLB) or slices, and for an Altera device they are, Logic Array Blocks (LAB), Adaptive Logic Modules (ALMs) or Logic Elements (LEs).

For a more indepth description of the different types of FPGAs, ASICs and microprocessors, the authors invite the reader to examine Chapter 3 of [6].

2.1 Xilinx FPGA

Configurable logic blocks (CLB) are organized in an array and are used to build combinatorial and synchronous logic designs. Each CLB element is tied to a switch matrix to access the general routing matrix. A CLB element comprises a number of similar slices, with fast local feedback within the CLB. These slices are split into columns with independent carry logic chains and common shift chain.

Each slice includes a number of multi-input Look-Up Tables (LUT), carry logic, arithmetic logic gates, wide function multiplexers and storage elements, namely D-type flip-flops. CLBs can be configured to operate as either a logic or a memory element. When operating as a logic element, the LUTs are programmed to operate as combinational logic, with a 1-bit register, or as a variable-tap shift register. As a memory element, each LUT is configured as an 2^n x 1-bit Distributed RAM block. The slices also contains logic that combines function

generators to provide multiplexing, sum of products (SOP) chains, shift registers and tri-state buffers used to drive on-chip buses.

In most cases, Xilinx CLBs share a similar make-up. There are distinctions though, as certain families are designed for different specific purposes. For example, the Virtex-II Pro can use each LUT as a 16 x 1-bit RAM. Others come with a specific designation; 'L' denoting a low power version for example. The underlying technology is quite similar however. For the Spartan3 onward, slices are seperated into those which include built-in RAM resources (SLICEM) and those which do not (SLICEL), with two of the four slices in a CLB being SLICEM and two being SLICEL. This allows each CLB to operate both as logic or memory.

The exception to this is the newer Virtex5. It incorperates larger LUTs in its CLB, and as such, it can achieve more varied functionality than the previous generation FPGAs (at a greater monetary cost). SLICEM logic for the Virtex5 can be configured as 64-bit Distributed RAM.

2.2 Altera FPGA

Equivalently, each Logic Array Blocks (LAB) consists of a number of Adaptive Logic Modules (ALMs) for the Stratix device family, carry chains, shared arithmetic chains, LAB control signals, local interconnect, and register chain connection lines, and a number of LEs for the Cyclone device family,

Each ALM contains a variety of LUT-based resources that can be divided between two combinational adaptive LUTs (ALUTs) and two registers. ALMs (and LEs) can also be configured for two different modes of operation; normal and dynamic arithmetic. Normal operates as standard combinational logic, while dynamic arithmetic configures the LUTs to operate as a dynamic Adder/Subtractor, Accumulator, or other arithmetic function.

In addition to the LUTs, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Altera FPGAs also contain embedded memory blocks of three different sizes of embedded SRAM. The smallest block, MLAB, can be used to implement small FIFO buffers and shift registers. Each ALM in an MLAB can be configured as a 16x2 block of simple dual port SRAM block.

For the lower cost Cyclone devices, each LE has four inputs, a four-input LUT, a register, and output logic.

Table 1 summarizes the logic resources in one CLB or one LAB. All of the CLBs and LABs in a given FPGA device are identical and each CLB (or slice equivalent) and LAB (or ALM or LE equivalent) can be implemented in one of the configurations listed above. For the Distributed RAM and Shift-Registers columns, the values given refer to the number of bits that one slice (or SLICEM) LUT can be configured to store. For example, each Spartan3 SLICEM LUT can be configured as a 1 x 16-bit shift register, with 4 SLICEM LUTs per CLB, thereby allowing 4 x 16-bit shift registers per CLB. These can also be configured in a smaller number of larger shift registers.

Table 1. Area logic resources in one CLB or LAB per FPGA Type

Xilinx	Slices	LUTs 4-input	LUTs 6-input	Storage Elements	Distributed RAM (bits)	Shift Registers (bits)
Virtex-II Pro	4	8	-	8	1x16	1x16
Spartan3	4	8	-	8	4x16	4x16
Virtex4	4	8	-	8	2x16	2x16
Virtex5	2	-	8	8	4x16	2x64
Altera	ALMs/LEs	LUTs 7-input	LUTs 4-input	Storage Elements	Distributed RAM (bits)	Shift Registers (bits)
Stratix III	10	10	-	10	20x16(MLAB)	20x16(MLAB)
Cyclone III	16	-	16	16	512x16(MLAB)	512x16(MLAB)

2.3 Memory and DSP Blocks

Some Xilinx FPGA devices also incorporate large amounts of block memory resources (BRAM) and dedicated multiplier or DSP blocks.

The BRAM complements the distributed memory resources that provide shallow RAM structures implemented using the CLBs. Implementing using BRAM (for example, S-boxes), can improve the clock frequency while also reducing the number of slices required. Note the BRAM usage does not show up in the CLB area result and so must be taken into account separately.

For the Xilinx devices, both the Virtex-II Pro and the Spartan3 contain dedicated 18-bit x 18-bit, twos complement signed multipliers. The Virtex4 has an equivalent DSP block comprising an 18x18, multiplier (with optional pipeline stages) and a built-in Accumulator(48-bit) and adder-subtracter called XtremeDSP. The Virtex5 also has a DSP block (DSP48E). In this case, each DSP48E slice contains a 25 x 18 multiplier, an adder, and an accumulator.

For Altera Stratix devices, as stated earlier, TriMatrix embedded memory blocks provide three different sizes of embedded SRAM. 320-bit MLAB blocks (simple dual-port memory) used to implement FIFO buffers and shift registers, 9-Kbit M9K blocks (true dual-port memory) that can be used for general purpose memory applications and 144-Kbit M144K blocks for processor code storage, packet and video frame buffering. While each embedded memory block can be independently configured to be a single or dual-port RAM, ROM, or shift register, and multiple blocks of the same type can be joined to produce larger memories with minimal timing penalty.

Table 2 summarizes the total area per device in slice and block memory available, in blocks and K-bits. All of the device types listed are taken from the hardware results page of the SHA-3 zoo. Only hardware results involving a particular device type of a family used for analysis are given. This can therefore

be used to calculate the total area usage of a hash function result in relation to the total area available for a given FPGA target architecture.

The I/O pin-count is also given for the purpose of calculating the maximum size of message blocks that can be processed and can be used to help calculate the maximum throughput that can be achieved [1]. In the case where multiple values are given, denoted by *, the number of I/O pins is package dependent.

Table 2. Total Area logic resources per FPGA Type

FPGA	Xilinx				
Virtex-II Pro	Slices	BRAM (Blocks)	BRAM (Kb)	Dedicated Multipliers	Max User I/O's
xc2vp40	19,392	192	3,456	192	72-108*
xc2vp50	23,616	232	4,176	232	144
xc2vp100	44,096	444	7,992	444	1040-1164*
Spartan3					
xc3s5000	33,280	104	1,872	104	633
xc3s1400AN	11,264	32	576	32	502
Virtex4				DSP Blocks	
xc4vlx100	49,152	240	4,320	96	960
Virtex5				DSP Blocks	
xc5vlx50	7,200	96	1,728	32	560
xc5vlx50t	7,200	120	2,160	48	480
xc5vlx110	17,280	256	4,608	64	800
xc5vlx220	34,560	384	6,912	128	800
FPGA	Altera				
Stratix III	ALMs	M144K-MLAB (Blocks)	Total RAM (Kb)	Dedicated Multipliers	Max User i/o's
EP3SE50	19,000	12-950	5,328	384	296-488*
EP3SL340	135,000	48-6750	16,272	576	744-1120*
Cyclone III	LEs	M9K (blocks)			
EP3C5	5,136	46	423.936	23	182
EP3C10	10,320	46	423.936	23	182

3 Comparison

As a common reference design, the core functionality module of the present standard SHA-256 module was developed, representative of a single operation for SHA-256 (which would be repeated 64 times for a full implementation). The

[1] Only the family and device are given. The package and speed grade of the various implementations are not taken into account. We refer the reader to the relevant data sheet [4] [5] for these differences and also any differences in BRAM or Multipliers between different FPGA family members

design takes in eight 32-bit words, as well as a 32-bit message word W_t and a 32-bit constant K_t, and outputs eight 32-bit words. Figure 1 shows a block diagram of the design. The architecture is basic; it doesn't use any pipelining or unrolling techniques. The design consists of registers, Adders and associated logic. This design was implemented on each of the different FPGAs using Xilinx ISE 9.2 and Altera Quartus II 9.1. They were then examined for area usage.

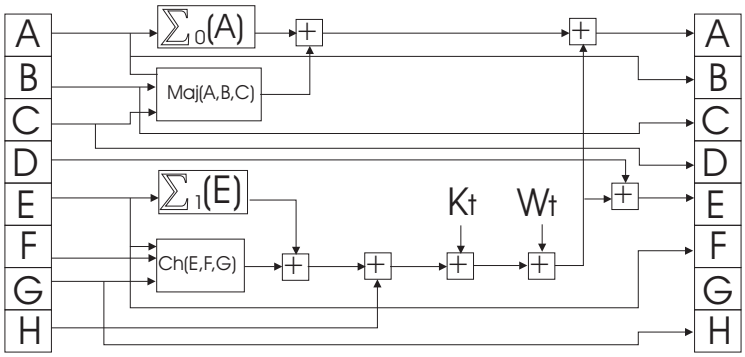

Fig. 1. SHA-256 Core Functionality

Table 3 shows the Post-Place-and-Route results obtained [2]. As can be seen from the table, the Virtex-II PRO, Spartan3 and Virtex4 can all be easily and directly compared, and each contain a roughly equivalent number of LUTs and Slices. The Virtex5, having slightly larger LUTs and slightly smaller Slices, reflects this in the table. For the Altera devices, as seen in Table 1 and Table 3, there is roughly a 2:3 ratio between ALMs and LEs.

Obviously these become more difficult to compare when including BRAM/MxK Blocks. Xilinx technology gives a gate-equivalent metric in an attempt to normalize the "gate count" of different objects by estimating how the element compares to a number of 2-input-NANDs, and therefore how it compares to an ASIC. This is however a very rough estimate. An alternative "rule of thumb" measurement for ASIC measurement, is to multiply the Slice count by seven or eight.

4 Conclusions

As can be seen from above, while some of the different platforms can be farily compared, i.e. all of the 4 input LUT Virtex FPGAs, some "loose" comparisons can be made between the rest of the target technology, i.e. those with equivalent LUTs or BRAM. This can be used to estimate whether or not a particular design

[2] Due to the large number of input and output signals, for smaller FPGAs, the Place and Route process will not be able to proceed past Mapping. However it will give correct area measurements before failing

Table 3. Total Area Usage for SHA-256 per FPGA Type

FPGA	Xilinx		
Virtex-II Pro	No. of 4 I/P LUTs	No. of Occupied Slices	No. of I/O's
xc2vp40	742	410	580
xc2vp50	742	410	580
xc2vp100	742	410	580
Spartan3			
xc3s5000	744	412	580
xc3s1400AN	740	408	580
Virtex4			
xc4vlx100	754	418	580
Virtex5	No. of 6 I/P LUTs	No. of Occupied Slices	No. of I/O's
xc5vlx50	654	499	580
xc5vlx50t	654	499	580
xc5vlx110	654	499	580
xc5vlx220	654	331	580
FPGA	Altera		
Stratix III	Combinational ALUTs	Logic Registers	No. of I/O's
EP3SE50	289	256	580
EP3SL340	Design not supported in 9.1		
Cyclone III	LEs	Logic Registers	No. of I/O's
EP3C5	545	256	580
EP3C10	545	256	580

can fit on a particular platform, thereby speeding up the development time to which an accurate analysis can be done. We also attempted to de-mystify the large number of abbreviations used by the vendors to describe their measurement units, and the scales of each.

References

1. http://csrc.nist.gov/groups/ST/hash/sha-3/index.html
2. http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo
3. http://ehash.iaik.tugraz.at/wiki/SHA-3_Hardware_Implementations
4. http://www.xilinx.com/support/documentation/
5. http://www.altera.com/literature/lit-index.html
6. Rodríguez-Henríquez, Francisco and Saqib, N. A. and Díaz-Pèrez, A. and Koc, Cetin Kaya, *Cryptographic Algorithms on Reconfigurable Hardware (Signals & Communication Technology)*, Springer-Verlag New York, Inc., 2006
7. National Institute of Standards and Technology, *FIPS PUB 197*, Advanced Encryption Standard, November 2001.