Benchmark Designs For The Quartus University Interface Program (QUIP)

Version 1.1

February 6, 2008

Table of Contents:

1. Overview

This document is intended for developers working with the Quartus University Interface Program (QUIP), and complements the Benchmarking [1] and Synthesis Interfaces [2] documents for researchers using QUIP as a front-end tool.

As part of the QUIP release, Altera is providing a number of VHDL and Verilog benchmark designs for use in benchmarking design flows, architectures and algorithms. These benchmarks are documented here. Some benchmarks can be converted to BLIF using Quartus Integrated Synthesis (see [2]) and used with academic tools such as SIS. Others that contain embedded memory or other blocks such as DSP cannot be converted to traditional BLIF. However, they can be used in other flows such as the hierarchical BLIF flow also described in [2]. Additional benchmarks that require this particular flow are available on Alan Mishchenko's web page [6]. Designs will be added to this web-page in between QUIP releases. The characteristics of the initially available designs on that web-page are described in [5].

All of the benchmarks in this design-set compile through the Quartus flow can be used via the VQM interface.

Any questions on the content in this document should be sent to quip@altera.com for advice. For fastest response, please reference the name of the document so that the question can be routed to the most appropriate person.

2. Benchmark Designs

The benchmark characteristics are shown in Tables 1-7. The location of the benchmarks is in the directory benchmarks/ in the QUIP release.

For each benchmark, we show the design-name, the number of I/O pins, LEs, DSP blocks, memory (in Kbits) for a standard Quartus II 5.0 compile to Stratix, and a brief description of the benchmark.

The current design set has 60+ designs, of which 12 are over 3000 LEs in size and 4 over 10,000 LEs in size. For the designs translatable to BLIF there are 5 more than 3000 LEs, but all of the 10,000+ sized designs have either DSP or RAM issues with BLIF conversion.

A wealth of designs of virtually any size can also be created using the DSP-Builder tool. We have not included any such designs here.

Table 1 gives characteristics of simple HDL designs for multiplexors, barrel shifters and crossbars. These designs can be generate very discontinuous synthesis, and you should watch. For example, one could choose to synthesize a barrel shifter into a DSP-block (multiply by constant), which would generate 0 logic cells. The number of LEs is shown for Stratix 4-LUTs. Table 1 also shows a simple finite state machine written in Verilog. This example is more to have an example of state-machine processing in Quartus than as a benchmark.

Table 2 gives characteristics of designs from free-ip.com. There are two cordic-related designs and a RISC CPU.

Table 3 gives characteristics of opencore.org designs related to encryption – des, aes and blowfish.

Table 4 gives characteristics of the opencore.org designs related to control logic. These include processors, memory and dma control, vga, I/O, etc.

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
barrel16	yes	38	129	0	0	
barrel16a	yes	26	107	0	0	Alt. Implementation
barrel32	yes	71	322	0	0	
barrel64	yes	136	882	0	0	
mux32_16bit	yes	54	853	0	0	Bus of muxes
mux64_16bit	yes	87	1702	0	0	
mux8_128bit	yes	140	1667	0	0	
mux8_64bit	yes	76	835	0	0	
xbar_16x16	yes	97	176	0	0	One-bit 16x16
ts_mike_fsm	yes	15	16	0	0	Simple FSM

Table 1. Simple HDL designs

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
fip_cordic_cla	yes	53	412	0	0	
fip_cordic_rca	yes	53	425	0	0	Alt. Implementation
fip_risc8	no	113	2381	0	384	RAM

Table 2. Designs from free-ip.com

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
oc_aes_core	no	388	1680	0	32768	RAM
oc_aes_core_inv	no	389	1947	0	34176	RAM
oc_des_area_opt	yes	189	691	0	0	
oc_des_des3area	yes	304	1135	0	0	
oc_des_des3perf	no	298	15990	0	2744	RAM
os_blowfish	no	585	1527	0	67168	RAM
oc_des_perf_opt	yes	185	5336	0	0	

Table 3. Opencore.org encryption-type designs

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
oc_miniuart	no	27	186	0	0	
oc_minirisc	no	389	635	0	1024	RAM
oc_oc8051	yes	189	3031	0	4608	RAM, Asynch signals
oc_sdram	yes	304	271	0	0	Asynch signals
oc_ssram	no	298	111	0	32	RAM
oc_vga_lcd	no	585	2207	0	32640	RAM, Asynch signals
oc_hdlc	no	82	646	0	2048	RAM, Asynch signals
oc_mem_ctrl	yes	267	3972	0	0	Asynch signals
oc_mips	no	201	4340	0	1152	RAM
oc_wb_dma	yes	444	3479	0	0	Asynch signals
os_sdram16	yes	83	320	0	0	Asynch signals
oc_ata_ocidec1	yes	125	540	0	0	Asynch signals
oc_ata_ocidec2	yes	125	588	0	0	Asynch signals
oc_ata_ocidec3	no	130	1045	0	224	RAM, Asynch signals
oc_ata_v	yes	125	290	0	0	
oc_ata_vhd_3	no	130	1040	0	224	RAM, Asynch signals
oc_ethernet	no	211	2607	0	9216	RAM, Asynch signals
oc_fcmp	yes	70	153	0	0	
oc_fpu	yes	110	6967	8	0	
oc_gpio	yes	141	220	0	0	Asynch signals
oc_hdlc	no	82	640	0	2048	RAM, Asynch signals
oc_i2c	yes	33	293	0	0	Asynch signals
oc_pavr	no	52	4212	0	32768	RAM, Asynch signals
oc_pci	no	367	2439	0	1720	RAM, Asynch signals
oc_rtc	yes	93	283	0	0	Asynch signals
oc_aquarius	no	35	6329	8	131072	RAM, Asynch Signals

Table 4. Opencore.org processor, control-type designs

Table 5 gives characteristics of the opencore.org designs related to DSP, and audio/video processing. The cordic, DCT and other designs are in this table, as well as some video compression algorithms.

Table 6 gives characteristics of "other" designs collected from the public domain. Currently, there are only four academic designs here. In future QUIP releases we hope this to be the growth area for the design set.

Table 7 gives characteristics of auto-generated VHDL and Verilog test cases (Section 6 of [4]). These designs are not specifically provided as benchmarks, but they are quite good at testing easy optimizations (sweep and register removal) in Quartus, and might be of some benefit for the same in exercising academic tools.

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
oc_video_compression_systems_dct	no	24	36419	0	18688	RAM, Asynch signals
oc_video_compression_systems_huffman_dec	yes	23	649	0	0	Asynch signals
oc_video_compression_systems_huffman_enc	yes	23	613	0	0	Asynch signals
oc_video_compression_systems_jpeg	no	47	32287	0	16640	RAM, Asynch signals
oc_correlator	yes	85	478	0	0	Asynch signals
oc_dct_slow	yes	24	306	0	0	Asynch signals
oc_cfft_1024x12	no	68	1655	0	24576	RAM, Asynch Signals
oc_simple_fm_receiver	no	22	826	0	8192	RAM
oc_cordic_p2r	yes	82	1016	0	0	
oc_cordic_r2p	yes	74	1426	0	0	

Table 5. Opencore.org audio/video/DSP designs.

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
uoft_raytracer [3]	no	772	35473	144	54758	RAM, Asynch signals
radar12	no	5	6230	78	269152	12 ch radar processing
radar20	no	5	13501	80	487744	Same, 20 channels
pong [7]	no	32	661	0	0	Video game "Pong"

Table 6. Academic Designs

Benchmark	BLIF?	I/O	LEs	DSP	Mem	Comments
nut_000	Yes	314	631	0	0	DSP/SR suppressed
nut_001	no	208	1382	0	2120	RAM
nut_002	Yes	122	500	0	0	DSP/SR suppressed
nut_003	Yes	466	770	0	0	DSP/SR suppressed
nut_004	Yes	321	337	0	0	DSP/SR suppressed

Table 7. Auto-generated VHDL/Verilog

3. Using the Designs

Researchers should be aware of various aspects of synthesizing and reporting benchmarking results on these designs.

Quartus Integrated Synthesis has many logic optimization settings, and the results you receive will change with different performance requirements, settings on speed/area preferences in technology mapping, use of physical synthesis, etc. Area numbers will change significantly when using the BLIF-output flow, because this will disable the use of any carry-chain hardware in the logic cell and synthesize all adders into LUTs. Thus, the area numbers listed in the tables above are a rough guideline only, and apply to default settings.

Before doing any benchmarking against Quartus, please consult the Benchmarking guide that is part of the QUIP release [1]. It is also important to carefully list any changes made to Quartus settings so that results can be duplicated by follow-up research.

Some designs are notoriously sensitive to delay/area tradeoffs. For example, many of the encryption algorithms such as DES have two dramatic local-minima in synthesis. For example, when synthesized for speed the design can be half the size and twice the speed. Full-system designs are less likely to have this behavior, as the effects are averaged out to single-digit swings between area and speed.

Some of the designs with DSP and RAM can synthesize these into LEs using appropriate settings. The setting:

set_global_assignment -name AUTO_SHIFT_REGISTER_RECOGNITION OFF

(added to your QSF file) will prevent Quartus from synthesizing longer shift-registers into 512b memories. In general it is better to do this, but for BLIF benchmarking you will need to suppress it to avoid the memory synthesis. Note, that you don't need to suppress it if you are using the hierarchical BLIF flow described in [2].

Similarly, you can force Quartus to not use the hard DSP multiplier blocks, even when they exist in the architecture, with:

set_global_assignment -name DSP_BLOCK_BALANCING "LOGIC ELEMENTS".

In both cases you can just add the setting to the QSF file. These changes to the default synthesis settings can have dramatic effects on the results, and researchers should carefully look at the report files provided by Quartus to understand the issues.

4. Future Enhancements

In future releases of QUIP the benchmark set will be expanded. To facilitate this, researchers are encouraged to nominate their public-domain benchmarks to add to the design set. Contact quip@altera.com or any Altera researchers affiliated with QUIP.

As the design-set expands, we would also ideally like to provide more documentation on the designs – e.g. design documents, block-diagrams – and classify the designs by their application domain.

5. References

[1] Benchmarking Using the Quartus University Interface Program (QUIP). Document "quip_benchmarking.pdf" in the QUIP package.

[2] Synthesis Design Flows Using the Quartus University Interface Program (QUIP). Document "quip_synthesis_interface.pdf" in the QUIP package.

[3] J. Fender and J. Rose, "A High-Speed Ray Tracing Engine Built on a Field-Programmable System,", in IEEE Int'l Conf. On Field-Programmable Technology (FPT 2003), pp. 188-195, 2003. See also http://www.eecg.utoronto.ca/~jayar/benchmarks/bench.html .

[4] B. Ratchev, M. Hutton, G. Baeckler and B. van Antwerpen, "Verifying the Correctness of FPGA Logic Synthesis Algorithms", in Proc. ACM/SIGDA Int'l Symposium on FPGAs (FPGA 2003), pp.127-135, 2003.

[5] J. Pistorius, M. Hutton, A. Mishchenko, R. Brayton, "Benchmarking Logic Synthesis for FPGAs", in Proc. Int. Workshop on Logic and Synthesis (IWLS 2007), pp. 230-237, 2007.

[6] http://www.eecs.berkeley.edu/~alanmi/benchmarks/altera

[7] D. Koch, C. Haubelt and J. Teich: "Efficient Hardware Checkpointing – Concepts, Overhead Analysis, and Implementation", in Proc. ACM/SIGDA Int'l Symposium on FPGAs (FPGA 2007), pp. 188-196, 2007.