PROBLEMS SOLVED

William Bricken

August 2002

==========

PERFORMANCE PROBLEMS SOLVED

==========

"My timing is to slow to meet the bus protocol"

-- Competitor's solution: spend two months of engineering expertise to hand place and route the design

-- BTC solution: guaranteed minimum logic processing at .5 ns per level automatically
(400 MHz per five levels of logic)

-- BTC solution: customize bus interface protocol in reconfigurable logic

==========

"Your one million system gates accommodate only 160K logic gates"

-- Competitor's solution: Designs are so different that we can't tell you how much logic will fit into one our chips. Besides if you hire an place and route expert for a few months, you can fit more logic.

-- BTC's solution: System gates are not a relevant measure. 100K available Comesh logic gates means 70K user gates after place an route.

==========

"My design saturates the routing resources and wastes half the chip area."

-- Competitor's solution: Our next generation chips have much more routing.

-- BTC's solution: More routing means more expense. Our modern hardware architecture uses blocks of 300 gates with five levels of logic, removing the need to high overhead routing resources.

==========

==========

"I have an exotic design that cannot meet performance specifications on your chip."

-- Competitor's solution: Simplify your design, please.

-- BTC's solution: Our software can reconfigure your design to meet performance specifications automatically.

==========

"I need a low-cost solution for millions of gates, but I'm forced to use many of your chips. Cross chip communication is killing my performance."

-- Competitor's solution: We're working on it.

-- BTC's solution: Our scalable architecture permits millions of user gates on the same chip, without performance loss.

==========

"My board requires fixed pin assignments, but your chip won't let me do that."

-- Competitor's solution: Try redesigning your board.

-- BTC's solution: We accommodate any pin assignments for your i/o logic, and you can change them anytime without degrading the performance.

==========

==========

COST PROBLEMS SOLVED

==========

"My design is too big for the FPGA chip"

-- Competitor's solution: buy a more expensive chip

-- BTC solution: set the design parameters for the size you need
expect 30% logic reduction compared to competition
can tradeoff a longer critical path if more logic compression is desired

==========

"A thousand gates of usable logic ends up cost me about five dollars. For 100K gates, that's $500!"

-- Competitor's solution: Those are our prices; try an EDA tool to optimize your design.

-- BTC's solution: We provide better performance at very competitive prices.

==========

"I'm paying for many hardware options, like an 18-bit ASIC multiplier, that I'm not using"

-- Competitor's solution: That's wheat we put on our chips so that they appeal to a wider customer base.

-- BTC's solution: For most customers, our reconfigurable performance is so good that you will not need specialized hardware. We can provide custom logic for high volume applications.

==========

"I have to pay tens of thousands of dollars to get software that makes you chips usable."

-- Competitor's solution: We're in the hardware business, go to an EDA company if you want better software performance.

-- BTC's solution: Exceptionally easy-to-use software with superior performance at very low cost is provided when you buy our chips.

==========

==========

"We have to buy exorbitant rates to expert consultants just to get your hardware to work proficiently."

-- Competitor's solution: Perhaps your design is too complicated

-- BTC's solution: Hardware/software codesign eliminates the need for expertise by automating high performance layout.

==========

"Your chip costs include unreasonable overheads to support software development and legacy products."

-- Competitor's solution: Those are our costs.

-- BTC's solution: Modern codesigned hardware and software keeps our performance high and our costs low.

==========

==========

DESIGN PROBLEMS SOLVED

==========

"Every time I iterate my design, the FPGA timing changes"

-- Competitor's solution: sorry, that's how FPGAs work

-- BTC solution: guaranteed 300MHz stable timing for design changes (for less than six levels of logic)

==========

"Your architecture makes it difficult to meet specifications for our data rich application."

-- Competitor's solution: Try our hybrid product with embedded data processing hardware.

-- BTC's solution: The performance specifications for our chip apply to all types of circuitry. You do not have to limit your design.

==========

"Why do I have to use an equivalence checker for each design change?"

-- Competitor's solution: That's how the design tool chain works.

-- BTC's solution: Our formal system guarantees that you cannot make a design change that alters the functionality of your design.

==========

"Our designer did not have the time to do an efficient design."

-- Competitor's solution: That's your fault.

-- BTC's solution: Regardless of the quality of your design, so long as it is functionally correct, our software will turn it into a highly optimized, extremely efficient design that can meet your performance specifications.

==========

==========

"Our designer sketched out an FSM state machine, how can I test its performance?"

-- Competitor's solution: Have your designer write an HDL file, then run it through an EDA logic synthesizer, then pass those results through our layout software, and if you have not made any mistakes, it will run. When you change it, it will run differently.

-- BTC's solution: Submit the state machine to our software and it will run on our chip immediately. Modest changes in the machine will not change the runtime performance.

==========

==========

SOFTWARE PROBLEMS SOLVED

==========

"That damned software is so difficult to use, and it never gives me the results I want"

-- Competitor's solution: We're working on it, why don't you try the next software revision.

-- BTC's solution: Describe the functionality you want and the specifications it must meet and we will generate a circuit that satisfies your needs. Iterating the design configuration is not necessary.

==========

"We have to buy a dozen different software tools and train engineers in a dozen different skills to get from specification to performing hardware."

-- Competitor's solution: It's a complex process that needs many different design steps.

-- BTC's solution: Complex design and layout is an accretion of twenty years of FPGA evolution. Our modern hardware/software codesign shows that getting from specification to high performance hardware is easy.

==========

==========

"Our synthesis results are completely degraded when we go to place and route"

-- Competitor's solution: Try to iterate between synthesis and routing until you find a good solution.

-- BTC's solution: Hardware/software codesign integrates synthesis and layout, providing a automated solution that combines superior synthesis with superior chip utilization.

==========

"Hand layout is too difficult cause we do not understand how to partition our functionality. When we try different partitions, we get vastly different performance results."

-- Competitor's solution: Our EDA partners can help you.

-- BTC's solution: EDA support for hardware is expensive and not that good. Our software knows the best way to partition for our hardware, giving you stable high performance without design effort.

==========

==========

HARDWARE PROBLEMS SOLVED

==========

"We're stuck with a product that was designed ten years ago cause it is too difficult and expensive to upgrade to your newer chips."

-- Competitor's solution: We're still supporting products from ten years ago, do even try to upgrade.

-- BTC's solution: We take any netlist specification and upgrade to better performance at a lower cost, automatically.

==========