Chapter 1 : Skew-Tolerant Circuit Design - David Harris - Google Books

Skew-Tolerant Circuit Design presents a systematic way of achieving the same goal and puts it in the hands of all designers. This book clearly presents skew-tolerant techniques and shows how they address the challenges of clocking, latching, and clock skew.

We then discuss technology scaling over the next several generations, from SIA and other predictions, and how our wire metrics trend over that time. We will examine the delay and bandwidth lim We will examine the delay and bandwidth limitations of both long global wires and short local wires and discuss architectural design techniques that help us avoid the limitations of scaled wires. Show Context Citation Context Historical FO4s per clock, machines. One architectural method for increasing processor performance involves increasing the frequency by implementing deeper pipelines. This paper will show that the branch misprediction latency is the single largest contributor to performance degradation as pipelines are stretched, and therefore branch prediction and fast branch recovery will continue to increase in importance. We will also show that higher performance cores, implemented with longer pipelines for example, will put more pressure on the memory system, and therefore require larger on-chip caches. ACM , " Power consumption has become a major constraint in the de-sign of processors today. To optimize a processor for energy-efficiency requires an examination of energy-performance trade-offs in all aspects of the processor design space, including both architectural and circuit design choices. In this pa-per, we apply an integrated architecture-circuit optimiza-tion framework to map out energy-performance trade-offs of several different high-level processor architectures. We show how the joint architecture-circuit space provides a trade-off range of approximately 6. We then show that many of the designs in this space come at very high marginal costs. Our results show that, for a large range of design objectives, voltage scaling is effective in efficiently trading off performance and energy, and that the choice of optimal architecture and circuits does not change much during voltage scaling. This type of design essentially removes all clocking overheads that are found in conventional designs. For this method to work, all logic must be a monotonic function of its input domino logic , so Oklobdzija , " We present a systematic approach for deriving a clocked storage element suitable for "time borrowing" and absorption of clock uncertainties. We explain how to compare different clocked storage elements with each other, and discuss issues related to power consumption and low-pow We explain how to compare different clocked storage elements with each other, and discuss issues related to power consumption and low-power designs. Finally, results of comparisons among representative designs are presented. Energy and performance models for clocked and asynchronous communication by Kenneth S. IEEE , " Parameterized first-order models for throughput, energy, and bandwidth are presented in this paper. Models are developed for many common pipeline methodologies, including clocked flopped, clocked time-borrowing latch protocols, asynchronous two-cycle, four-cycle, delay-insensitive, and source synchr Models are developed for many common pipeline methodologies, including clocked flopped, clocked time-borrowing latch protocols, asynchronous two-cycle, four-cycle, delay-insensitive, and source synchronous. The paper focuses on communication costs which have the potential to throttle design performance as scaling continues. The models can also be applied to logic. The equations share common parameters to allow apples-to-apples comparisons against different design targets and pipeline methodologies. By applying the parameters to various design targets, one can determine when unclocked communication is superior at the physical level to clocked communication in terms of energy for a given bandwidth. Comparisons between protocols at fixed targets also allow designers to understand tradeoffs between implementations that have a varying degree of timing assumptions and design requirements. Table 4 shows the throughput models. Each equation was independentlyderived and is highlyparameterized for delays and variations. Setup, skew, and flop delays create the overhead for clocking. The DI protocols are also relatively simple because theysequentiallyexecute each logic and communication operation. Smith , "

Chapter 2 : CiteSeerX â€” Citation Query Skew-Tolerant Circuit Design

Skew-tolerant domino circuits use multiple overlapping clocks to eliminate latches, removing hard edges and hiding the sequencing overhead. This thesis presents a systematic approach to skew-tolerant circuit design, combining both.

Given these formulae, hold times, and practical clock generation issues, we conclude that four-phase skew-tolerant domino circuits are a good way to build systems. We then return to general domino design issues, including monotonicity, footed and unfooted dynamic gates, and noise. Chapter 4 puts together static and domino circuits into a coherent skew-tolerant circuit design methodology. It looks at the interface between the two circuit families and shows that the static-to-domino interface must budget clock skew, motivating the designer to build critical rings entirely in domino for maximum performance. It describes the use of timing types to verify proper connectivity in static circuits, then extends timing types to handle skew-tolerant domino. Finally, it addresses issues of testability and shows that scan techniques can serve both latches and skew-tolerant domino in a simple and elegant way. None of these skew-tolerant circuit techniques would be practical if providing the necessary clocks introduced more skew than the techniques could handle. Chapter 5 addresses clock generation and distribution. Many experienced designers reflexively cringe when they hear schemes involving multiple clocks because it is virtually impossible to route more than one high-speed clock around a chip with acceptable skew. Instead, we distribute a single clock across the chip and locally produce the necessary phases with the final-stage clock drivers. We analyze the skews from these final drivers and conclude that although the delay variation is nonnegligible, skew-tolerant circuits are on the whole a benefit. In addition to tolerating clock skew, good systems minimize the skew that impacts each path. By considering the components of clock skew and dividing a large die into multiple clock domains, we can budget smaller amounts of skew in most paths than we must budget across the entire die. By this point, we have developed all the ideas necessary to build fast skew-tolerant circuits. With a little practice, skew-tolerant circuit design is no harder than conventional techniques. However, it is impossible to build multimillion transistor ICs unless we have tools that can analyze and verify our circuits. In particular, we need to be able to check if our circuits can meet timing objectives given the actual skews between clocks in various domains. Chapter 6 addresses this problem of timing analysis, extending previous formulations to handle multiple domains of clock SkeW. Although such design is not difficult, it has been ignored by the bulk of engineers and computer-aided design systems because flip-flops were adequate when cycle times were long and thus sequencing overhead was small. High-end microprocessors push circuit performance to the limit and will benefit from skew-tolerant domino circuits to reduce overhead. Applicationspecific integrated circuits will generally have less aggressive frequency targets and are unlikely to employ domino circuits until signal integrity tools improve. Nevertheless, many ASICs will be fast enough that flip-flop overhead becomes significant, and a switch to skew-tolerant latches may make design easier. Given these trends, skew-tolerant circuit design should be an exciting area in the coming years. Suppose the setup and clock-to-Q delays of the flip-flops are 1. Assume there is no clock skew. You are targeting a 0. If the operating frequency is to be MHz, what fraction of the cycle time is wasted for sequencing overhead? Repeat if the operating frequency is 1 GHz. Your latches have setup and dock-to-Q delays of 1. They have a D-to-Q delay of 1. The flip-flops in your cell 3 0 1 Introduction library have a setup time of ps and a clock-to-Q delay of ps for the expected loading. You are budgeting clock skew of ps. How much time in nanoseconds is available for useful multiplier logic? Assume you will use the LDE flip-flop ignore the A, B, C, and 1 inputs used for level-sensitive scan in the E performance level driving a load of four standard loads. Extract the maximum setup time and clockto-Q i. Your computation requires 3 ns to complete. You are considering various pipeline organizations in which the computation is done in 1, 2, 3, or 4 clock cycles in the hopes that breaking the computation into multiple cycles would allow a faster clock rate. Make a table showing the maximum possible clock frequency for each option. Also show the total latency of the computation in picoseconds for each option. In practice, you may have time remaining at the end of a cycle for half a gate delay; this time goes unused because it is impossible to build half a gate. Redo the exercise if on average 50 ps at the end of each cycle goes unused due to imbalanced logic. How fast is microprocessor performance increasing? How about clock rate? How long does it take for performance to double? Copyright by International Business Machines. Have these trends accelerated or decelerated relative to the performance increases between and How do they compare with the SIA roadmap in Table 1. Consider using a pipeline built with traditional domino circuits. The pipeline requires 1 ns of logic in each cycle. Suppose the setup time of each latch is 90 ps and the clock skew is ps. What is the cycle time? What fraction of the cycle is lost to sequencing overhead? You are weighing whether to recommend static CMOS circuits or traditional domino circuits. You have determined that the setup time and D-to-Q delay of your latches are approximately equal. If you use domino circuits, you determine it will require only 0. Divide the space into two regions based on whether static CMOS or traditional domino circuits offer higher operating frequencies for the given logic delay and skew. Assume there is no imbalanced logic.

Chapter 3 : Skew-Tolerant Circuit Design [Book]

If you have this book go ahead and post it here and your listing will appear for all students at your school who have classes requiring this specific book.

Chapter 4 : calendrierdelascience.com | Skew-Tolerant Circuit Design (ebook), David Harris | | Boeken

Skew-Tolerant Circuit Design or any of the above titles can be purchased directly from Morgan Kaufmann via the Web at calendrierdelascience.com or by phone at

Chapter 5 : Skew-Tolerant Circuit Design - PDF Free Download

Skew-Tolerant Circuit Design David Harris Page 3 of 50 Sequencing Overhead in Flip-Flop Systems Ideally, each full clock cycle T c is available for useful computation But we must keep token in current pipestage from catching up with token in next.

Chapter 6 : Skew-Tolerant Circuit Design - O'Reilly Media

- Skew Tolerant Circuit Design â€¢ This is a book by David Harris that expanded on his thesis, and discusses much of the material in this (and the previous lecture).

Chapter 7 : CiteSeerX â€" Skew-Tolerant Circuit Design

D. Z. Pan Skew-Tolerant Design 3 Clock Distribution â€¢ On a small chip, the clock distribution network is just a wire - And possibly an inverter for clkb.

Chapter 8 : Skew-Tolerant Circuit Design : David Harris :

Get this from a library! Skew-tolerant circuit design. [David Harris] -- As advances in technology and circuit design boost operating frequencies of microprocessors, DSPs and other fast chips, new design challenges continue to emerge.

Chapter 9 : Skew-Tolerant Circuit Design - | SlugBooks

Skew Tolerant Circuit Design Clock skew wikipedia, in circuit design clock skew can be caused by many different things, such as wire interconnect length, temperature variations, variation in.