

Requirement-Based Design Methods for Adaptive Communications Links

Juan Antonio Carballo*, Kevin Nowka*, Seung-Moon Yoo*, Ivan Vo*, Clay Cranford†, Robert Norman†

*IBM Austin Research Laboratory, Austin, TX

†IBM Systems and Technology, Raleigh, NC

ABSTRACT

High-speed communications link cores must consume low-power, feature low bit-error-rates (BER), and address many applications. We present a methodology to design adaptive link architectures, whereby the link's internal logic complexity, frequency, and supply are simultaneously adapted to application requirements. The requirement space is mapped to the design space using requirements measurement circuits and configurable logic blocks. CMOS results indicate that power savings of 60% versus the worst case are possible, while the area overhead is kept under 5%.

Categories and Subject Descriptors: B.4.3 [Input/Output & Data Communications]: Interconnections (Subsystems).

General Terms: Design, Performance, Measurement.

Keywords: Energy Efficient Design, Communication Architectures.

1. INTRODUCTION

Wired communications hardware is composed of chips that may include tens to hundreds of serial communication cores. Each core is a set of transmitters (serializers) or receivers (deserializers) working at multi-Gigabit/second speeds [8]. These cores implement the lowest physical communication layer and have direct contact with the off-chip communication channel, whose materials and distances may vary widely. Specifications require these cores to consume low power while meeting tough BER requirements [9]. Such a tough trade-off has made the design of these cores very complicated. Receivers in particular are most complicated, since they have to implement power-hungry Clock-and-Data Recovery (CDR) algorithms. Because logic power scales well with supply voltage, CDR algorithms are increasingly implemented with semi-custom logic [6]. However, as this logic becomes more complex, power becomes difficult to control. For example, a 64-state filtering state machine running at GHz speeds may be needed. In theory, power could be reduced by voltage supply reduction but high speed requirements coupled with logic complexity makes voltage scaling difficult to apply. Due to design complexity and limited design resources a single, "conservative" CDR circuit is typically implemented to ensure a high BER over all applications. Logic is designed based on the most stringent requirement, and the power supply and frequency are high to meet speed requirements. Thus the circuit often consumes more power than necessary for most applications.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2004, June 7–11, 2004, San Diego, California, USA
Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

This paper presents a method to design adaptive power management architectures for communication links, whereby the link's CDR internal logic complexity, clock frequencies, and voltage supply levels are simultaneously adapted to its application requirements. By application requirements we mean the difficulty of meeting a given BER specification for the given link. Difficulty is determined by transmission media parameters and physical and system connection characteristics that degrade the data signal and/or reduce timing margin. These variables include jitter and transmitter-referred frequency offset. Adaptability allows the CDR circuitry to consume low power when requirements are positive (e.g., channel quality is high), while avoiding extra design variants or over-conservative design. This method is applicable to any given environment, including proprietary backplanes and standards-based links (e.g., InfiniBand). It is also applicable to other fields within communications system design.

The remainder of the paper is organized as follows. First, relevant background in link architecture design is presented. Second, our method for designing adaptive architectures is presented. Third, the design and impact of key power reduction levers are discussed. Finally, results for a multi-protocol serial transceiver core and conclusions are presented.

2. PRIOR WORK

CDR architectures consist of circuit blocks that recover clock and data information from an incoming serial data stream [4]. The quality with which this information is recovered largely determines the bit error rate (BER) system performance. To achieve low BER under increasingly tougher bandwidth, power, and cost specifications, two types of sections are generally used [7]. First, feed-forward sections over-sample the input stream and then make a decision about the data value using a voting scheme. This technique suppresses high frequency jitter of limited amplitude. Second, feedback sections generate signals that indicate whether the sampling edge is at its expected position, earlier, or later in time. These signals are typically used to control the output phases of a Phase-Locked-Loop (PLL), used to sample the data and complete the CDR feedback loop. This loop takes care of low frequency jitter, and is the most power-hungry logic in the link. Unfortunately, existing architecture approaches feature fixed, complex loops whose functionality does not change with channels or applications. Generally no configurability exists in the CDR loop. Outside the loop, only very limited configurability exists for equalization, testing, or debugging purposes [1].

3. REQUIREMENT-BASED DESIGN METHODS FOR ADAPTIVE LINKS

Fixed CDR implementations are very power-inefficient for many common applications. As described in Figure 1, our method consists of providing an efficient mapping from the link requirements space R_S to the link design space D_S as a set of power modes $M \subset R_S \times D_S$.

This mapping can be done automatically on-core (a fully adaptive implementation) or may be done from outside the core (by controlling the configuration through on-chip or off-chip pins). In the remainder of this paper, we will assume a fully adaptive implementation. Our method is based on a power management architecture template that can be effectively adapted to the requirements of the communications system where it is embedded. Figure 2 depicts this architecture in a serial link receiver.

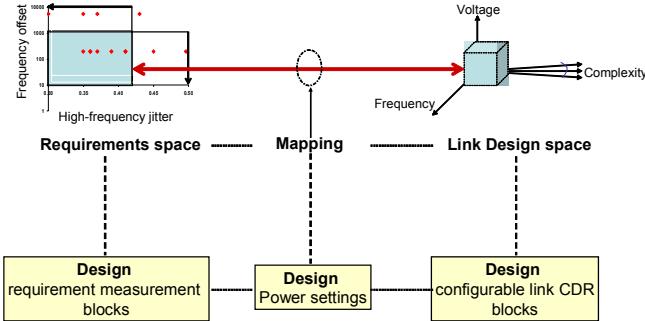


Figure 1 Design of adaptive communications links.

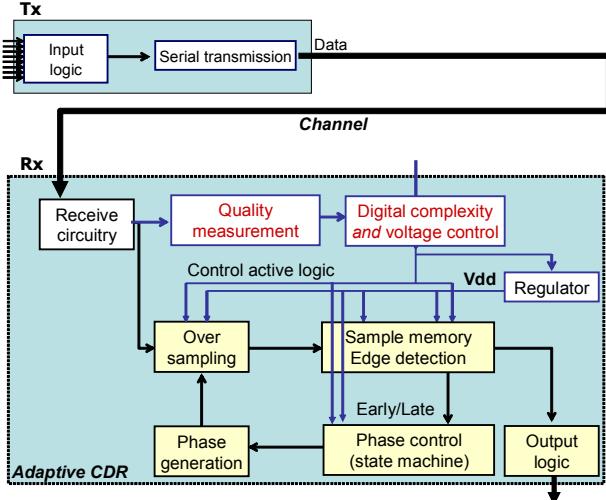


Figure 2 Adaptive CDR-based receiver architecture.

The method used to map the requirements space to the design space is based on the following design tasks:

- **Design application requirements measurement.** First, the requirements space R_S is defined. Then, a *quality measurement* block is designed. This block estimates requirements by measuring the characteristics of the communications system where the receiver is embedded. These characteristics are quantitative components of the “difficulty” of the system.
- **Design configurable link.** The CDR loop is decomposed into a “canonical” set of blocks (“Over-sampling”, “Sample memory + Edge Detection”, “Phase Control”, and “Phase Generation” in the figure) including all considered feed-forward and feedback functions. Each block is individually configurable in terms of its logic complexity (via “Control Active logic” in the figure), frequency (also “Control Active logic”), and/or supply voltage (V_{dd} in the figure). A multi-dimensional, discrete design space D_S is thus defined, from which any design point is selectable. Logic

complexity is selectable by making logic configurable. Frequency is selectable through a simple clock selection scheme. Finally, supply voltage is selectable by attaching one or more digitally-controlled on-core regulators.

- **Design power modes setting.** Requirement subspaces are selected and mapped to link design subspaces via power modes. Power mode setting is implemented by a digital *complexity and voltage control* block. Based on requirements measurements, this block decides what power mode is appropriate for the BER requirements while avoiding unnecessary power, and sets the mode on the CDR blocks.

Power modes are defined with a simple heuristic procedure (others are possible). For each design dimension or parameter in D_S , BER performance is analyzed to understand its impact on each requirement. Based on this analysis, low-power values are initially selected for each parameter. If a certain percentage (e.g., >75%) of applications are satisfied and distinguishable via measurement, a power mode is defined by those applications and all current design values. Otherwise, higher parameter values are selected. This procedure is repeated until R_S is covered. For each mode, switching power savings can be estimated by expressing power consumption as a function of CDR circuit characteristics:

$$P \propto K \cdot f \cdot V_{dd} \quad (1)$$

where V_{dd} is the voltage provided by the regulator; f is the average frequency at which the logic runs; and K is proportional to the average logic switching capacitance. Reducing logic complexity reduces the number of gates doing CDR computations. Therefore, it lowers power as switching capacitance is reduced. Reducing frequency also helps as power is approximately proportional to logic frequency. Finally, reducing voltage supply helps because power is approximately proportional to voltage, assuming an efficient on-core linear voltage regulator is used [3]. (An on-chip regulator is chosen because it allows the core to use the global VLSI power supply, leaving core interfaces unmodified; off-chip supply control would increase cost, even though it would result in power savings that grow with the square of voltage.) Unfortunately, delay depends on V_{dd} too. If V_{dd} is low and logic complexity and/or frequency are high, timing violations will occur. Thus low V_{dd} values can only be used when low-power modes are selected and thus critical paths are effectively shorter.

Next we describe the requirement and CDR design spaces that enable the design of the adaptive link, including its power modes.

3.1 Requirement space

The requirement space R_S contains the values for key system signal characteristics affecting BER that at least one considered application is subject to. We consider a two-dimensional space, as illustrated in Figure 3 (each application is shown as a point). The horizontal axis shows the percentage of the signal eye lost in system-induced jitter, and the vertical axis shows the transmitter-referred peak frequency offset that the receiver needs to compensate for in asynchronous applications. (We assume peak offset is measured over a microsecond-range interval, including a small fixed offset and a periodic variation, sometimes called “spread-spectrum clocking” or SSC. For simplicity, we will refer to it as “peak frequency offset”.) Based on the figure, difficult applications are in the top-right area in the graph, while easier applications are in the bottom-left area. Adapting the CDR characteristics to each of these regions can save significant power.

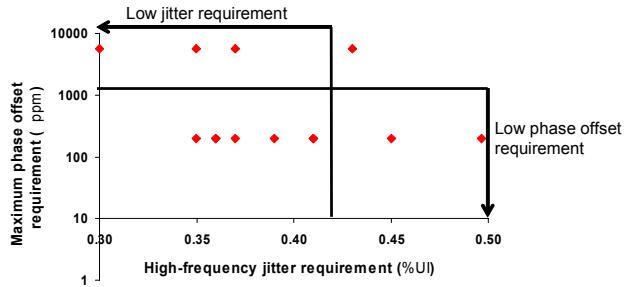


Figure 3 Jitter and frequency offset space.

Focusing on the jitter dimension, Figure 4 illustrates the difficulty of satisfying a BER specification for a set of 3.125 Gbit/sec applications (a1-a14 in the figure). (This data was obtained by system simulations on detailed models of the communications system from Figure 2, including the CDR circuitry and all sources of signal degradation. Random input data and Matlab/C modeling style are used. The environment is calibrated with hardware data.)

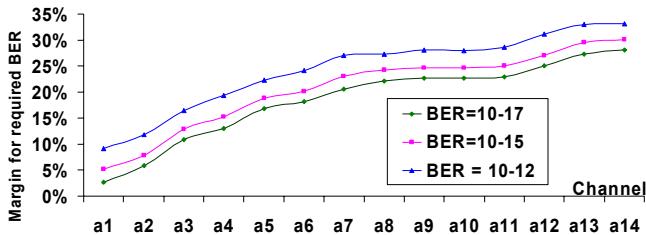


Figure 4 Difficulty of meeting BER requirements.

The difficulty of each application is expressed on the vertical axis as the allowable jitter at the receiver that meets a given BER. The applications toward the left in the chart correspond to difficult channels with high induced signal-degrading jitter. The intuition behind our approach is that requirements to the right side of the chart can be satisfied using a low-power mode link setting.

3.2 Design space

The CDR design space D_S contains all CDR configurations to which requirements may be mapped. Each power mode is given by a mapping to a point in D_S that defines a configuration for each CDR block. Based on potential block configurations, the following parameters may be selectable (more are possible):

- **Voltage-related parameters:** the global logic voltage supply.
- **Frequency-related parameters:** (a) the over-sampling rate; (b) the averaging rate for sample memory and edge detection; (c) the rate or frequency at which the phase control machine runs.
- **Logic-complexity parameters:** (a) the finite-state machine function, (b) the edge detection block's averaging function, and (c) the resolution (number of steps) of the phase generator.

Our implementation only considers the parameters in Table 1.

For a given application, average BER performance (BER_{CDR}) depends to a first approximation on the first three parameters, while power consumption (P_{CDR}) will depend on all four:

$$BER_{CDR} \approx f_0(s, c, l), \quad P_{CDR} \approx f_p(s, c, l, V_{dd}) \quad (2)$$

These functions will generally be monotonic in their variables, thereby making automated optimization feasible. (BER estimation is

done as described in Section 3.1, and power estimation is based on calibrated logic power models.) Thus adaptability is provided by (a) measuring BER requirements and (b) setting the values of CDR design space parameters s , c , l , and V_{dd} based on the measurement. Next we describe requirements measurement design and power mode design based on this configurable CDR.

Selectable parameter	Symbol	Possible values	Effect on performance
Over-sampling rate	s	2 (per bit) 3 (per bit)	High value → use for high-jitter environments
State machine complexity (given by number of states)	c	1 (64 states) 2 (32 states) 3 (16 states) 4 (8 states)	High value → use for high-jitter environments
Loop latency (frequency of machine and rest of logic)	l	1 (highest rate) 2 (half-rate) 4 (quarter-rate)	Low value → use for high peak frequency offset
Logic voltage supply	V_{dd}	1.2V – 0.8V	High value → use complex/fast FSM

Table 1 Configurable CDR block parameters.

4. REQUIREMENTS MEASUREMENT

Automatically measuring application requirements requires a design technique that results in low overhead on-core integration, and provides good estimates of the key components of system quality. Figure 5 shows our approach to this problem.

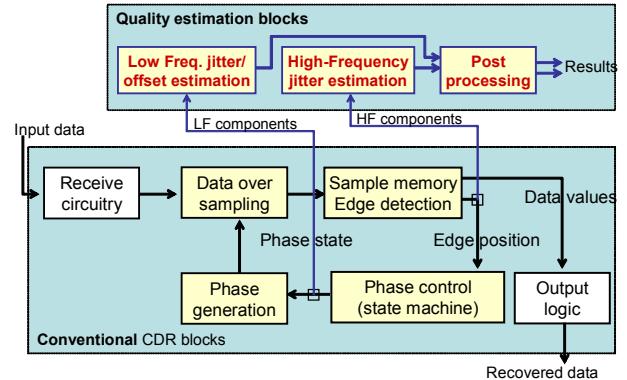


Figure 5 Measurement of application requirements.

Our method consists of designing a simple quality measurement block that selects certain digital signals inside the CDR loop and processes their values to obtain estimates of high-frequency and low-frequency jitter in the incoming signal. The key observation is that, in modern CDR loops, some of the work involved in measuring and separating jitter components is already done. Figure 6 depicts a simplified digital implementation of this measurement block. Two design steps are performed here:

- **Design signal selection and initial estimation blocks.** First, output signals from the CDR's over-sampling section are selected as inputs to estimate high-frequency jitter components. Specifically, signals from the edge correlation logic ("Early" and "Late") are selected. To overcome their limited resolution, these signals are aggregated over n bits by using a filtering-type operator (cumulative sum). Similarly, signals from the feedback section are selected as inputs to estimate low-frequency jitter components, which in turn provide a measure of peak frequency

offset. Specifically, signals from the phase control state machine logic ("phase-up" and "phase-down") are selected and filtered over m bits, and the result is roughly proportional to the long term drift of the center sampling point. m and n are typically tens of thousands.

- **Design post-processing block.** Based on the two measures, post-processing logic is used to produce two final indicators j_H and j_L . This logic is necessary because if the application's peak frequency offset is high, the high-frequency jitter estimator may include a component of low-frequency jitter. To correct for this component a simple post-processing linear subtraction block is used. The value to be subtracted is a small multiple, k_p , of the low-frequency jitter estimator. (To implement the multiplication, a small shift register is often accurate enough.)

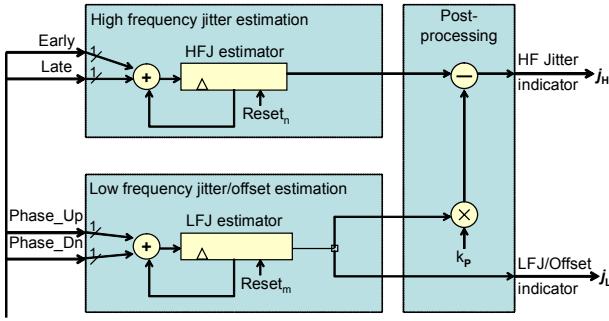


Figure 6 Digital requirements measurement circuit.

k_p , n , and m are easily calibrated by hardware-correlated simulation. Unlike other approaches [2,5,11], these measurement blocks are fully digital and can be synthesized with conventional ASIC methodologies with little area overhead (less than 2%).

Figure 7 shows the output of the high-frequency jitter measurement logic as a function of BER performance for a set of 3.125 Gbit/sec applications. (The results in this section were obtained using the simulation method described in Section 3.1, and the vertical axes have been scaled by a constant factor for confidentiality.) As the figure indicates, with a suitable threshold, the circuit can readily identify difficult applications with very low jitter margin. Thus the output of this logic may be used as an effective jitter requirement indicator to assess application difficulty. (All applications in this experiment had the same frequency offset.) For easy applications with low channel-induced jitter, the indicator does not produce as much differentiation. The lowest power mode will suffice for these applications.

Peak-frequency offset between transmitter and receiver may be significant in certain applications (0.5% or more). Figure 8 shows the output of the low-frequency jitter measurement logic as a function of the peak-frequency offset for our set of applications. Each dot in the figure is a cluster of applications with the same peak frequency offset. In our implementation we consider two clusters (see circles): very low-offset applications ($\leq 0.05\%$), and high-offset ($\geq 0.5\%$) applications. The graph indicates that the measurement logic is an effective estimator for this requirement.

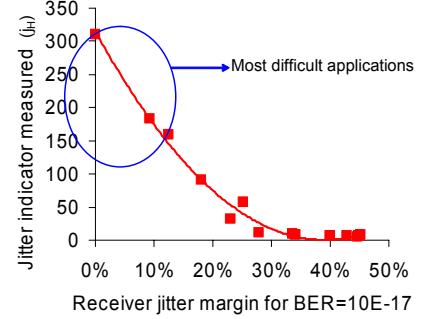


Figure 7 High frequency jitter measurement.

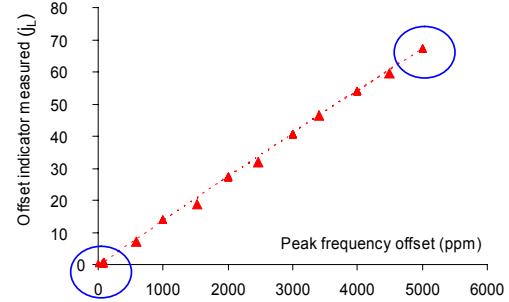


Figure 8 Frequency offset measurement accuracy.

5. ADAPTIVE POWER MODES

In this section, we examine the design of each configurable design space parameter, and the impact of each parameter on BER performance that leads to the definition of power modes. (Simulations were run as described in Section 3.1.)

5.1 Design dimension: over-sampling rate

We consider two possible values for this parameter: two and three samples per incoming bit. This parameter can be designed to be selectable on-chip using a digitally controllable phase generator (PLL), over-sampling unit, and sample memory. It has a strong impact on BER performance when the application's high-frequency jitter is high, because more samples per bit mean higher sampling resolution, and thus high-frequency signal edge movements can be averaged out more effectively. Figure 9 shows the impact of over-sampling rate on BER performance for the most difficult application channels.

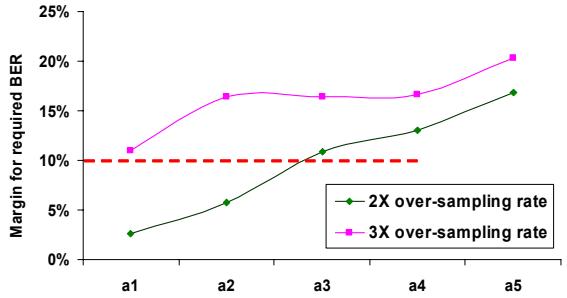


Figure 9 Impact of over-sampling rate on BER.

5.2 Design dimension: filtering algorithm

The key element in the phase control logic is a finite-state machine (FSM) that determines whether the sampling phases need to be pushed forward or backward based on edge detection information.

The functionality of this FSM is critical for BER performance, because it implements a non-linear filtering scheme that tracks long-term phase shifts while filtering out unwanted system-induced jitter. A complex FSM with many states consumes more power than a simple one but may result in improved jitter tolerance. However, the impact on BER is critical only when the application's high-frequency jitter is very high. Smaller FSMs can meet BER requirements for many applications. Based on this observation, we consider four possible FSMs: a 64-state machine, a 32-state machine, a 16-state machine, and an 8-state machine.

Figure 10 shows the impact of FSM type on BER performance for 4 example application channels. Each smaller FSM mimics the state diagram structure of the larger one, but reduces the number of states proportionally (see Figure 10 (a)). For example, assume the 64-state FSM has 4 clusters of states or "levels", with each level including 16 states. When the FSM is at the lowest level, the sampling phases have been repeatedly found late with respect to the incoming signal, and thus the FSM will order frequent increases in sampling phase. Conversely, at the highest levels frequent phase decreases will be ordered. The 32-state FSM can be built on the same principle, but using 2 instead of 4 state "levels". This smaller FSM can track most of the long-term phase drift, but filters a little less unwanted jitter than the larger FSM.

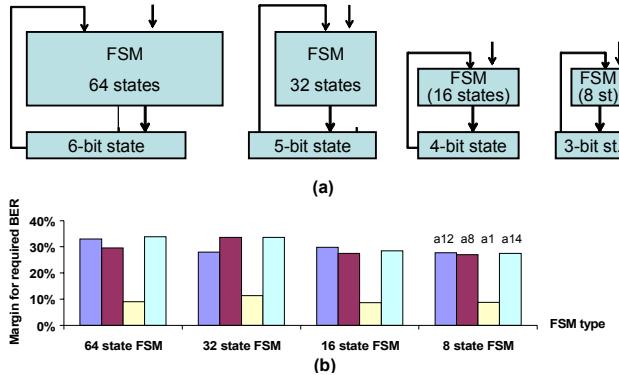


Figure 10 Effect of FSM complexity (a) on BER (b).

As Figure 10(b) indicates, using the smaller FSMs for these applications has no significant impact on BER performance. According to our measurements only the hardest 10% applications require the most complex machine (have under 5-10% margin) – similar to the over-sampling case in Section 5.1. This statement was valid regardless of the peak frequency offset associated with the application. Additionally, these four FSM versions have a lot of functionality in common. As a result, the area overhead of implementing a configurable state machine with multiple settings is low, around 1% per extra setting in average.

5.3 Design dimension: loop latency

The third configuration parameter is the frequency at which the phase control logic operates, i.e., the clock frequency with which edge averaging and subsequent state machine operation happens. We refer to this setting as the loop latency, since it impacts the minimum time required for a new state machine output (and hence phase increase or decrease) to be produced. This setting has a significant impact on BER performance only for applications with high peak frequency offset: a low-enough latency is critical for the BER loop to catch up with fast phase/frequency shifts. As Figure 11 (top) shows, this setting can be easily implemented through simple

clock selection (plus simple configurability of edge averaging logic, not shown). The area overhead of this configurability is under 1%. We consider 3 possible values for this setting: highest (standard) clock frequency, half frequency, and quarter frequency. (Note that the highest frequency may be a fraction, 1/n, of the incoming data frequency.) Figure 11 (bottom) shows the impact of this design dimension on BER performance for the 90% "easy" applications. When peak frequency offset is low (<0.05%), the average BER performance for all applications is not significantly affected. However, when peak frequency offset is high (0.5% or higher), the standard frequency must be selected.

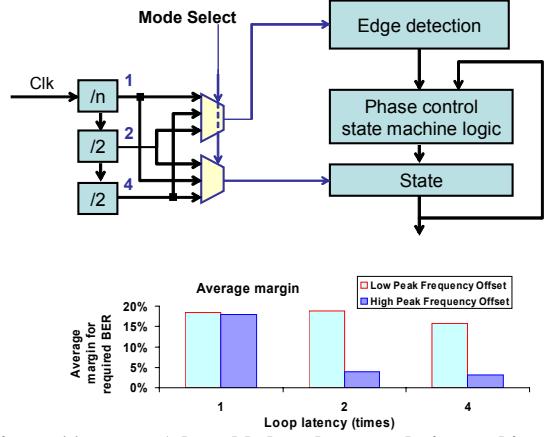


Figure 11 Adaptable loop latency: design and impact.

5.4 Design dimension: voltage supply

The last selectable feature is supply voltage. While it does not directly impact BER performance significantly, the supply has impact on the maximum logic complexity and frequency that can practically be supported for a given technology. Therefore, this voltage setting is selected based on the previously described settings (more complex adaptive regulators could be used [12]). In our implementation, we consider digitally-selectable voltages from 1.2V to 0.8V, with 1.2V the standard value. Through timing and circuit simulations using IBM's ASIC tools, we found that the voltage can be dropped at least 20% (to 1V) when either the FSM type or the loop latency are set to their lower-power settings. Figure 12 illustrates power savings from lower supply voltage.

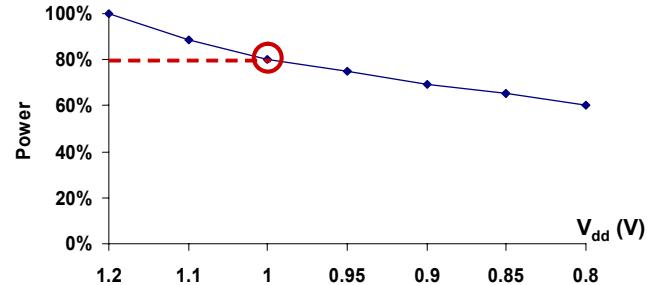


Figure 12 Measured power savings versus supply.

Laboratory-measured results are shown for an on-chip receiver logic fabricated with a 0.13μm CMOS technology. Power savings over 20% are possible by setting the regulator output to 1V (see graph). Simulations (not shown) predict similar results for a newer technology. Regulator area overhead is around 1%.

6. ADAPTIVE 3.125 GBIT/S CORE

Based on the method and the results shown in the previous section, we have defined three key power modes for our adaptive 3.125 Gigabit/sec serial link core implementation. Each mode corresponds to a setting for each of the four parameters explored in this paper: over-sampling rate, FSM type/complexity, frequency-induced loop latency, and voltage supply. Table 2 summarizes the settings for these power modes, and their corresponding logic power savings (see Table 1 for notation).

Table 2 Power modes in 3.125 Gbit/s link core.

Power mode	s	ϵ	l	Vdd (V)	J_H (jitter) indicator	J_H (offset) indicator	Power savings
High	3	1	1	1.2	>160	N/A	0
Std	2	2	1	1	<160	>10	48%
Low	2	2	4	1	<160	<10	63%

Figure 13 shows the estimated contribution of each of the four adaptive power features to total power savings, for the described power modes.

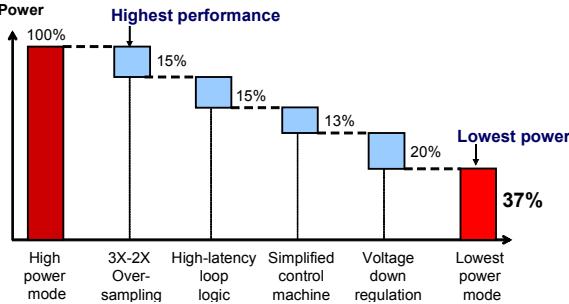


Figure 13 Power savings drivers for power modes.

The data in Table 2 and Figure 13 are based on hardware-calibrated power simulations performed using IBM's ASIC tools. We have verified the functionality of our architecture implementation in statically and dynamically-adaptive versions. In the dynamic version, the receiver initializes itself with a known input bit sequence to set its power mode and then works in that mode unless further calibration is requested. Figure 14 shows a simplified depiction of the methodology used.

7. CONCLUSIONS

Increasingly difficult bandwidth and BER requirements are making it very difficult to develop VLSI serial communications cores with acceptable power efficiency. This paper has proposed a novel method to design serial communications architectures whereby the core signal recovery logic can be adapted to the application requirements. This technique is based on mapping the application requirement space to the CDR design space. This mapping allows the CDR's logic complexity, internal frequency, and voltage supply to be adapted so power consumption is reduced while requirements are met. For the self-adaptive version, we have presented a set of low-overhead on-chip blocks that automatically measure aspects of the incoming signals, such as high-frequency jitter and frequency offset variation, to estimate the application requirements.

Experimental results indicate that power savings of over 60% are possible while the area overhead is kept small (around 5% overall). This method supports both self-adaptive and calibration-based approaches. It may also be applicable to analog blocks and to other areas of communications SoC design (off-chip and on-chip [10]).

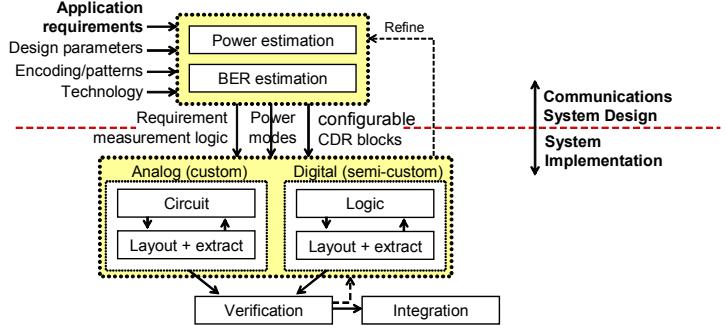


Figure 14 Adaptive link design methodology.

8. REFERENCES

- [1] W. Dally and J. Poulton, "Transmitter Equalization for 4 Gbits/second Signaling", Proc. Hot Interconnects Symposium, August 1996, pp. 29-39.
- [2] W. Dalal and D. Rosenthal, "Measuring Jitter of High Speed Data Channels Using Undersampling Techniques", Proc. International Test Conference 1998.
- [3] T. Endoh, et al., "An On-chip 96.5% Current Efficiency CMOS Linear Regulator Using a Flexible Control Technique of Output Current", IEEE JSSC, 36:1, Jan 2001, pp. 34 -39.
- [4] R. Farjad-Rad et al., "A 0.3- μ m CMOS 8-Gigabit/s 4-PAM Serial Link Transceiver", IEEE JSSC, vol. 35, issue 5, May 2000, pp 757-764.
- [5] J. Hamre, "Jitter Measurement Using a Statistically Locked Loop", US Patent 5481563.
- [6] D.-K. Jeong, "High Speed Serial Link for Fully Duplexed Data Communication", US Patent US6107946.
- [7] C.-K. Ken Yang, et al., "A 0.5 μ m CMOS 4Gb/s Serial Link Transceiver with Data Recovery Using Oversampling", IEEE JSSC, 33-5, May 1998, pp. 713 -722.
- [8] J.M. Khouri and K.R. Lakshmikumar, "High Speed Serial Transceivers for Data Communication Systems", IEEE Communications Magazine, 39:7, July 2001, pp. 160 -165.
- [9] M.-J.E. Lee, et al., "An 84-mW 4-Gb/s Clock and Data Recovery Circuit for Serial Link Applications", Symposium on VLSI Circuits, 2001, pp. 149 -152.
- [10] V.Raghunathan et al., "A Survey of Techniques for Energy Efficient On-chip Communication", Proc. DAC 2003.
- [11] Y.Shohet, "Jitter Measurement Device", US Patent 4975634.
- [12] G.-Y. Wei and M. Horowitz, "A Fully Digital, Energy-efficient, Adaptive Power-supply Regulator", IEEE JSSC, 34:4, April 1999, pp. 520 -528.