

Qualification and Integration of Complex I/O in SoC Design Flows

Jay Abraham, Guruprasad Rao

Magma Design Automation,
Silicon Correlation Division

Abstract

Low power, high speed, and reduced cost requirements force integration of specialized Intellectual Property (IP) like complex I/O blocks on a System on Chip (SoC). Today designers have access to a variety of specialized IP blocks and cells for use in SoC design flows. Complex I/O appear in a myriad of standards such as USB 1.0/1.1/2.0, IEEE 1394 a/b (FireWire), SSTL, HSTL, PCI-X, LVDS, and more. These new standards are driven by consumer's demand for bandwidth and capability, and the industry's desire to reuse proven design blocks in vastly different applications and domains [1]. Integration of these specialized IP blocks introduces increased complexity to design flows. For example, digital designs must now consider the analog like properties of some complex I/O. This paper discusses the uniqueness of embedding complex I/O in a SoC. The features and properties that differentiate complex I/O from standard design practices will be described. Finally methodologies for characterizing and building accurate digital abstractions of I/O will be presented.

Introduction

With increased use of specialized I/O in SoC designs, design automation tools struggle to perform efficiently. In order for these tools to perform optimally, automation tools require correct and accurate abstracted models that describe the functionality, timing, power, electrical, signal integrity, and other properties of the I/O device. Due to the analog like behavior of complex I/O in digital designs and the unpredictable behavior of nanometer silicon, the modeling of complex I/O often results in shortcuts and approximations. This means that design automation tools like synthesis, place and route, static timing, and power analysis rely on inaccurate models, which leaves them vulnerable to design closure failure when performing sign-off with static timing analysis (STA) or SPICE simulations.

To mitigate these inaccuracies, SoC design teams apply guardband and over-design. However, over-designing to accommodate timing inaccuracies results in increased die area and increased manufacturing costs. Over-designing to accommodate power inaccuracies

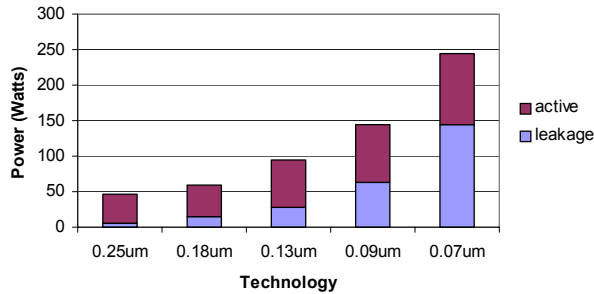
impacts chip pin-out and packaging costs. With the increased popularity of wireless and other portable devices, SoC power consumption in both active and standby states creates a critical issue for designers. Low power design techniques often focus on reducing power for the internal gates, despite the fact that I/O cells in an SoC design consume significant power. The need to drive large pad capacitances and board traces, as well as switching activity on busses results in increased power consumption. It is possible for I/O cells to consume as much as 50% of total power [2, 3]. Algorithmic techniques for bus pattern coding are being developed to reduce active power consumption [2]. Power reduction of an SoC will depend on accurate power consumption analysis of all components including I/O.

Nanometer technology trends

The complexities of cotemporary SoC designs require that analysis tools fully comprehend the details of the underlying nanometer technology. With silicon technology persistently proving Moore's Law, it is imperative that the complex electrical effects of nanometer silicon be understood and abstracted. These effects include resistive voltage drop which impacts circuit performance, increased leakage currents, noise in the form of cross talk and glitch, increased inductance and mutual inductance, electromigration and other reliability concerns, and intradie process variations.

Interest in leakage power is of increasing concern. Industry data indicates that leakage power is becoming prevalent in nanometer technology (see Figure 1) [4]. Leakage power consumption in standby mode—the mode in which a portable system spends the majority of its time—is becoming the significant contributor to battery power consumption. This is the result of leakage currents, which include reverse-bias-source or drain-diode currents, drain-to-source weak-inversion currents, and tunneling currents. The primary effect of these leakage-current components is to create a current flow from VDD to VSS through the transistor. This flow will occur even though the transistor is logically in the off state [4]. This complicates power analysis during active periods of the circuitry (even in nonstandby mode) because leakage

paths can be state dependent and leakage currents are masked by active switching currents.



Source: Microprocessor power consumption, Intel

Figure 1: Leakage power consumption trends[4]

In order for design tools to analyze nanometer electrical effects in a reasonable amount of time, characterization and model generation must be able to build abstract models that efficiently capture these behaviors.

Designing with Complex I/O

Comparing both the design and operation of a modern I/O cell against a standard logic cell demonstrates the substantial differences between the two. Yet many SoC designers are forced to work with I/O cells as if they were standard logic cells. Interface cells behave much differently than standard logic cells because they electrically connect to the board-level world. As a result, the modeling of I/O cells must include packaging constraints and complex parasitics, electro static discharge (ESD) protection concerns, analog inputs and outputs, voltage translation and over-voltage protection, and SI concerns.

Due to the wide variety of environments and electronic equipment in use today, a variety of signaling standards exist for the definition of I/O. The Joint Electron Device Engineering Council (JEDEC) prescribes standards for many different I/O types. However, almost all I/O devices require a common set of parameters to define I/O signal levels. These are [5]:

- VDD—chip supply voltage
- VDDQ—driver supply voltage
- VREF—input reference voltage
- VTT—termination voltage
- VIH/VIL—correct high/low input levels
- VOH/VOL—correct high/low output levels

With I/O driving bond wire and board traces, transmission line effects concern many designers. The types of networks include point-to-point networks consisting of a single driver and receiver pair, and multiple-load networks in which a single driver must supply a signal to

multiple external devices (for example, a microprocessor application providing a single address to several other devices).

Complex I/O characteristics

In this section, we describe specific characteristics of simple and complex I/O cells that are relevant to full chip timing and power analysis. I/O cells place substantial demands on the simple techniques that are typically available for modeling and characterization.

Differential Pins

Differential signals are more immune to line noise and distortions due to reflections and material effects. Differential signals are provided in pairs which transition about a common mode voltage and are found in I/O standards like USB, FireWire, LVDS, and PCI-Express. The circuitry and voltage waveforms are shown in Figure 2 for an LVDS (low voltage differential signal) I/O cell.

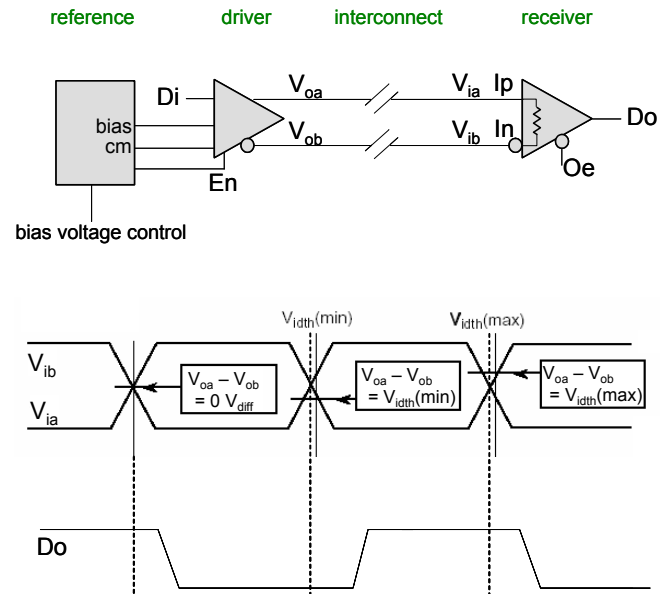


Figure 2: Circuitry & waveforms for a LVDS cell

The voltage swing on the differential input pin pair can be partial around a common mode reference. The term partial implies that the swing can be truncated to reduce current and power requirements. The resulting input signal is constructed from the difference between the two signals, hence the term differential[6]. These pins are specified as two separate ports in the timing and power model. The designer can acquire the delay from a differential input pin pair to an output pin on the cell

using the crossover point between the two signals as the trigger, where the crossover point is:

- 50% point on the differential waveform
- 50% point on the (+) or (–) waveform
- The smallest (largest) of the 50% point on either waveforms

Similarly, the input slew of the signal can be represented in several ways:

- The specified slew drives the + or the – signal and the fixed slew drives the other signal.
- The specified slew (matched) drives both signals.
- A pre-characterized differential output driver generates a table of input signal slews vs. difference signal slews. Designers refer to this table to obtain a difference signal of the desired slew.

Figure 3 demonstrates the variation in delay for and LVDS 0.13um cell that is observed when the two signals on the differential pair are not driven with matched slews. The y-axis demonstrates that delay can vary by as much as 45% if slews are not correctly matched.

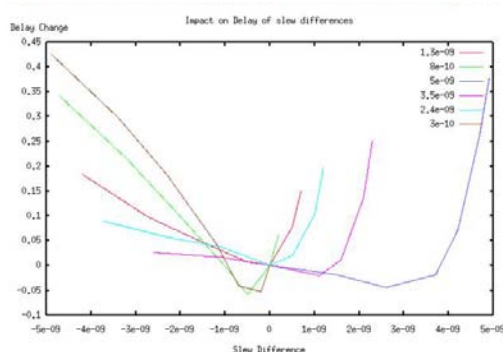


Figure 3: Delay vs. input slew for a receiver (mismatched on differential pins)

For the same cell, Figure 4 demonstrates the variability in delay due to selection of VDI (difference between the + and – differential signals) and COMMON (common mode voltage around which the + and – differential signals swing). The delay measured from the 50% point on the difference input waveform to the 50% point on the output waveform can vary from 1ns to 1.7ns depending upon the selection of VDI and COMMON. The characterization tool must allow the user to specify the appropriate VDI and COMMON voltage values to ensure that realistic delay values are provided in the model.

Differential output signals have similar characteristics to differential input signals, except the swing on the output signals is not known a-priori. Furthermore, the

swing is extremely sensitive to process, voltage, and temperature variations. To accurately model the delay and output slew characteristic, it is important to dynamically determine this swing on a simulation-by-simulation basis. The delay from an input pin to a differential output pin pair on the cell can be acquired using the same techniques as the input pin pair. The output load on the differential pair can be represented using one of the following options:

- Load the + or the – signal using the specified load and the other signal can be left unloaded or loaded with a fixed value.
- Load both signals using the specified load (matched loads).

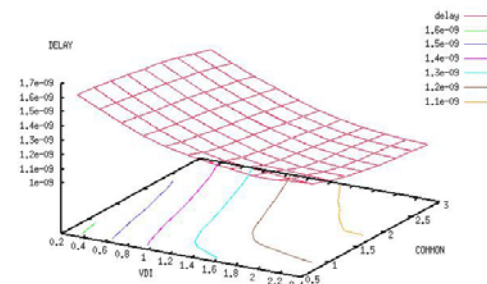


Figure 4: Delay vs. different differential input swings & varying common modes for a receiver

Figure 5 demonstrates that delay for an output differential driver can vary from +30% to -80% when compared with the matched load case.

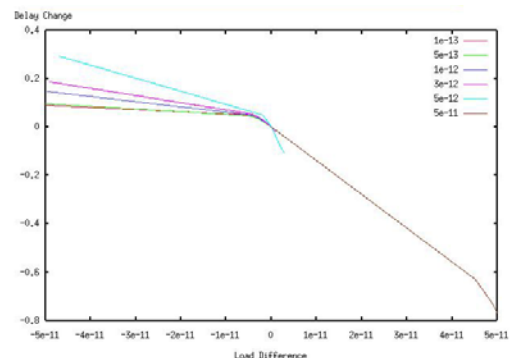


Figure 5: Delay vs. load for a driver (mismatched on differential pins)

Figure 6 demonstrates the differences in delay when only one of the signals is selected as the target of the acquisition vs. using the difference signal. Using the + (positive) or – (negative) signal severely underestimates or overestimates the delay when compared to dynamically determining the 50% point on the difference waveform.

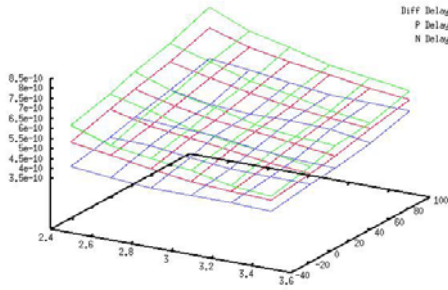


Figure 6: Driver delay vs. different trigger points

Differential swing sensitivity also varies as a function of voltages and temperatures. It is difficult to capture this sensitivity a-priori. The characterization tool must dynamically determine the swing (amplitude) of the output signal. In the figure, the differential output signal swing varies from 0.53 to 0.75 volts with a chip side voltage variation of 2.5 to 3.6V and a temperature variation of -40 to 100°C. The positive and negative output signal components demonstrate a similar sensitivity to voltage and temperature (see Figure 7).

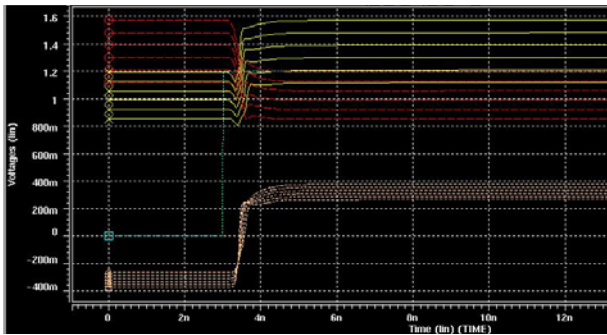


Figure 7: SPICE plot for driver varying chip side voltage (2.5 - 3.6V) & temperature (-40 to 100°C)

Modal Pins

Modal pins can switch the cell between different drive strength outputs by controlling the pull-up and pull-down behavior of the cell and they can also explicitly alter the functionality of the cell. When modal pins do not affect the functionality of the cell, care must be exercised when determining the range over which the delay arc is characterized. If the modal pin affects the drive strength of the output pin, different load ranges must be characterized to accommodate the required accuracy within the table. It is not uncommon to see variations in maximum load of more than an order of magnitude. See Figure 8 for an example I/O cell with mode pins that are controlled by a register.

Slew rate control is a common characteristic of modern, general-purpose, high-bandwidth interface standards (e.g. USB 1.0/1.1/2.0). Slew rate control provides

a constant current driver (thus providing frequency independent power consumption) in the face of large variations in data transfer rates. Additionally slew rate control on the driver minimizes radiated noise cross talk [7]. Slew rate control requires mode-specific, load range specification to obtain the characteristics of the driver because, depending on the mode, slew rate control generally begins at a specific point. Determining the point at which this compensation is initiated is critical for accurately characterizing the delay arc.

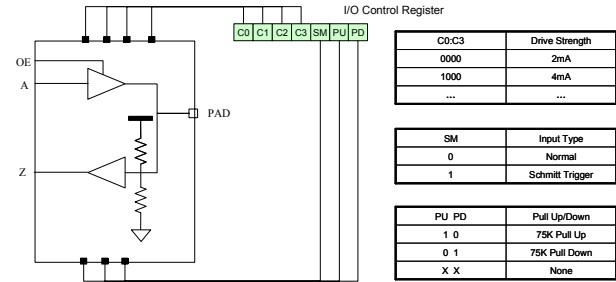


Figure 8: I/O cell with mode pins [6]

Increasing Input Slew (0.01-1.2ns) ↓	Increasing Output Load (50-300pF) →				
	4.34	4.34	4.33	4.35	4.35
	4.34	4.34	4.33	4.35	4.35
	4.34	4.34	4.33	4.35	4.35
	4.34	4.34	4.33	4.35	4.35
	4.34	4.34	4.33	4.35	4.35

Table 1: Output slew as a function of output load and input slew

Table 1 shows the typical behavior of a USB 1.1 I/O cells that has slew rate control. The table shows output slew for a transmit data pin as a function of increasing output load (50-300pF on transmit data pin) with increasing input slew (10ps-1.2ns on data pin). Note that with increasing output load, the slew rate circuitry automatically is able to maintain the same output slew rate. E.g. for fixed input slew, the first row in the table shows output slew to be at 4.33-4.35ns (20ps difference). Incidentally, the USB 1.1 specification requires output slew on D+ and D- pins to be between 4-20ns. The data in Table 2 shows that the designers have designed this USB 1.1 cell to spec.

Simultaneous switching

I/O cells are typically in noisy peripheral regions of the chip. Also, simultaneous switching creates self-induced noise. Banks of I/O cells typically handle the signals of a bus that sends and receives parallel data. If multiple signals switch simultaneously, the current draw

from a power pin that is shared across a set of these I/O cells can cause bounce on the supplies. This bounce has significant impact on the delay [8].

Power

Power requirements create limits in a variety of SoC design applications. Clearly, mobile applications that are dependent on battery power require low power design techniques to extend battery life. In addition, power consumption now limits fixed applications like packet switch designs for the telecom industry. Mikenberg et. al. indicates that with every new switch fabric design, there is an increased amount of power that is used for inter-chip signal transportation [9] (see Table 2).

Switch Size	16 X 16	16 X 16	32 X 32
Throughput (Gb/s)	6	32	64
Total Power (W)	6	12	25
I/O Power (W)	1	4	12
I/O Power (%)	16%	33%	50%

Table 2: Single SoC Power Consumption [9]

According to Rent's rule [10], the number of pins in a design, C_p , can be modeled as a function of the number of logic gates, C_g , as follows; $C_p = r_p (C_g)^\beta$. Where r_p is a constant and $\beta < 1$. Higher transmission rates require additional pins and logic, which results in increased power consumption by the I/O [9]. 50% power consumption by I/O cells is a very significant percentage of total power consumption. Therefore, the accurate characterization of power on I/O cells is critical for determining the design's viability. Since the I/O cell provides a translation from the reduced swing to the larger swing on the core side (for non-differential pins), it is important to accurately track the sources and sinks of power.

For cell-based designs, power is estimated in a full chip environment by using the results of typical stimulus sequences. A Verilog (or VHDL) simulation that executes a stimulus sequence to generate a change dump file (.vcd) can drive analysis tools. Analysis tools use models that accurately depict power and energy consumed during individual transitions. The analysis tools then perform the bookkeeping required to determine the following:

- Switching states—energy consumed by parasitic capacitances both inside and outside the cell,
- Hidden states—energy consumed on input transitions that do not cause output transitions, and
- Leakage states—steady states that draw residual current due to subthreshold leakage.

Characterizing all potential states of an I/O cell is impractical. The designer must quickly determine the representative arcs that capture the variations in energy, and generate models that describe the appropriate energy consumed for each transition and state. For accurate power characterization the designer must deal with several issues.

- A path might exist from a core-side pin to a chip-side output with a mirror or the complement of the input signal (or both) appearing on pins on the core side. These signals might be used as synchronization triggers for sophisticated self correcting protocols that allow the core side to know when to expect a response back from another chip on the bus. In order to determining the power consumed by a transition, the designer must know how to load these mirror pins on the core side.
- Chip-side driving pins supply significant current during a transition. While designers can ignore core-side driving pin current sources without significant loss of accuracy, the same cannot be assumed on the chip side.
- In order to determine the energy supplied, the designer must determine the period over which the current draw from the sources is integrated.
- Cells might require an initialization sequence before the energy is measured. The variation in the energy measured with and without initialization sequences may exceed two to three times the correct energy consumed during the transition.

Leakage can be a critical power component for I/O cells. I/O cells with termination resistors, pull-ups and pull-downs, leak more than standard cells. Accurate characterization of leakage for different states enables understanding the overall power consumed during a stimulus sequence, and ensures that quiescent or idle states do not consume more than the advertised power. Designers must verify the advertised numbers provided in vendor datasheets across a representative region of power, voltage, and temperature (PVT) corners.

Required characterization and modeling

To analyze complex I/O cell behaviors and measure various aspects of the cell, the I/O cell must be simulated using tools and an environment that reflects the actual operating environment. Some measurements, such as dissipated power and propagation delay, can be used in models to enable chip-level timing sign-off. Other measurements are required to validate a design's compliance with its electrical specification. The characterization and modeling of complex I/O requires three stages for efficient processing and accurate results. These stages as

shown in Figure 9 are planning, characterization, and model generation and verification.

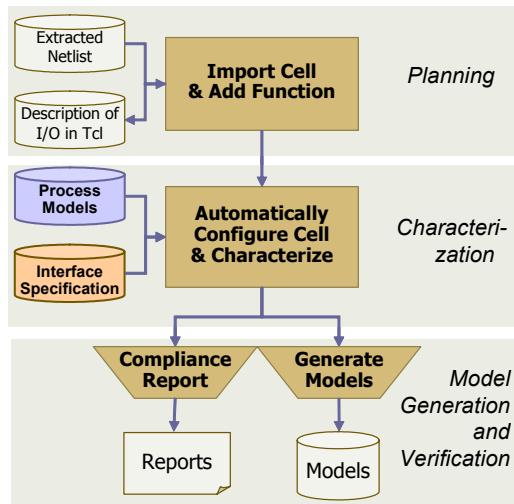


Figure 9: Required characterization methodology for complex I/O

During the planning phase, the designer must fully elaborate the complex I/O to determine the function, timing arcs, power arcs, state dependency, etc. and also generate required stimulus needed to invoke the SPICE simulator. A simulation deck that encapsulates the complex I/O circuit must accommodate all secondary pins. The designer must stub all unused pins, connect realistic input devices (e.g. active drivers), and utilize realistic output load harnesses (e.g. inductive loads). Furthermore, power supplies might have to reflect extracted inductance from the pad ring. Once all of the simulations are complete, the designer can acquire the data in either time specific measurements or waveform capture. Finally, once models are generated, the user must verify the results. Ideally, the user builds test harnesses for the complex I/O, performs measurements of the I/O device, and then cross compares the results with the generated model and the appropriate I/O specification.

Summary and Conclusion

SoC designs utilize a wide variety of specialized IP that include complex I/O cells. Integration of specialized IP components introduces complexities to the design process. Fundamentally, the design flow is dependent on high-quality function and performance models that abstract the functional and electrical behavior of the IP. Therefore the quality of the design is dependent on these models.

In this paper we have shown that generation of high quality complex I/O models can be a tedious and error prone task. Complex I/O introduces analog effects into

purely digital designs. Designers must contend with differential signals and nonfunctional modal pins. Additionally, care must be taken to accurately model leakage power that is becoming prevalent in nanometer technology. Incorrect capture of the behavior of complex I/O will lead to gross errors when abstracting timing and power models. This in turn will lead to errors in the design construction and analysis of the SoC.

Generation of accurate and correct timing, power, and SI models of complex I/O depend on automated flows. In this paper we have introduced the complexities and time consuming processes required to build these models. Automation is the key to providing designers with accurate and correct complex I/O. We briefly described required automated characterization and modeling flows for complex I/Os. These flows enable improvements in the design and analysis of nanometer technology SoCs.

Acknowledgements

I would like to thank Jason McCampbell and Scott Swarts for their help with researching, testing, and running experiments for this paper.

References

- [1] *Bigger faster ASICs are a specialty*, Electronicstalk, Texas Instruments Case Study, November 2000.
- [2] *Bus-Invert Coding for Low Power I/O*, IEEE Transactions on VLSI Systems, Mircea R. Stan and Wayne P. Burleson, Vol. 3, No. 1, March 1995.
- [3] *Power Estimation and Minimization of Digital Signal Processing Systems*, University of Illinois-Champaign Thesis, Sumant Ramprasad, January 1999.
- [4] *Managing Leakage in Mobile Operations*, Integrated Systems Design, Stephen King and Jerry Frenkil, December 5, 2001.
- [5] *High Speed CMOS Design Styles*, Kluwer Academic Publishers, Kerry Bernstein et al, 1999.
- [6] *FS9000A 0.25um Standard Cell Databook*, Faraday Technology Corp., www.dreamtel.com.
- [7] *Universal Serial Bus Specification Revision 2.0*, www.usb.org, April 2000.
- [8] *Estimation of On-Chip Simultaneous Switching Noise in VDSM CMOS Circuits*, International Conference on Modeling and Simulation of Microsystems, K.T. Tang and E.G. Friedman, March 2000.
- [9] *Current Issues in Packet Switch Design*, HOTNETS, Cyriel Mikenberg et al, 2002.
- [10] *Circuits Interconnections and Packaging for VLSI*, Addison-Wesley Pub. Co., H.B. Bakgolu, 1990.