

SiGe Prototype Chip Design Implementing CMOS Fixed Bit-Load Drivers and Receivers for Next Generation High-Speed Board-Level Interconnect

Student Designers:

*Jason D. Bakos, Amit Gupta, Leo Salavo
University of Pittsburgh*

Faculty Designers:

*Donald Chiarulli
University of Pittsburgh*

Abstract

We present our design and simulation results for a test chip that implements fixed bit-load drivers and receivers. The link architecture that is formed using these circuits offers several improvements over differential (LVDS-standard) links, including a substantial decrease in die area and power consumption. These designs were fabricated as part of a multi-project die in IBM's .5um 5HP SiGe process.

1. Introduction

Figure 1 shows our multi-project test die, manufactured in IBM's .5 um 5HP SiGe process. In this paper, we present a novel design for high-speed drivers and receivers which are located on the periphery of this die.

1.1. Introduction and Motivation for Fixed-Load Drivers and Receiver Circuits

We are investigating a new encoding method for high performance serial data channels. This new method has comparable performance characteristics to LVDS channels with up to 40% better power efficiency and a 1/3 reduction in pad count. The key components are a new driver circuit and termination network that can support both single and multi-drop link configurations and it is electrically compatible with standard LVDS receiver circuits. In the new system we replace the familiar 2-channel per bit format of differential signaling with an n-channel system in which all of the code symbols in the channel

must conform to an "n choose m" encoding rule. This system supports significantly higher code density while it duplicates the signal-to-noise characteristics of differential signaling. High performance serial data channels are of increasing importance for applications such as chip-to-chip PCB interconnect and disk drive data-channels as smaller features and lower voltages drive the economics of CMOS design away from traditional single-ended, bit-per-wire I/O.

In the new encoding, we replace the familiar 2-wire per bit format of a differential channel with an n-wire system in which all code symbols in the channel must conform to an n choose m (written nCm) code rule. In an nCm code, each n -bit symbol encoding must have exactly m 1-bits and $n-m$ 0-bits. We refer to this method as fixed bit-load signaling based on the constant number of 1-bits in the channel at all times. Fixed bit-load channels share the high performance electrical attributes of differential signaling while supporting a much larger code symbol set in the

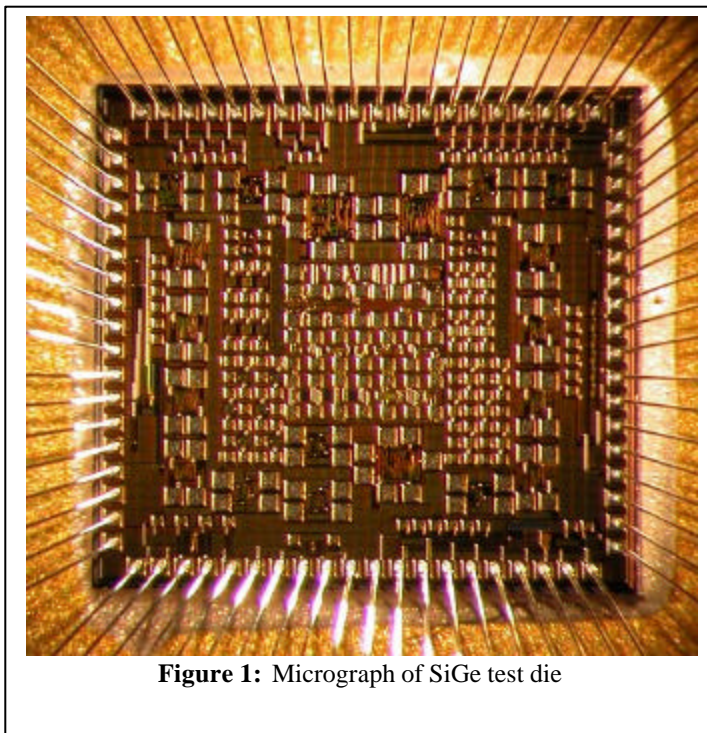


Figure 1: Micrograph of SiGe test die

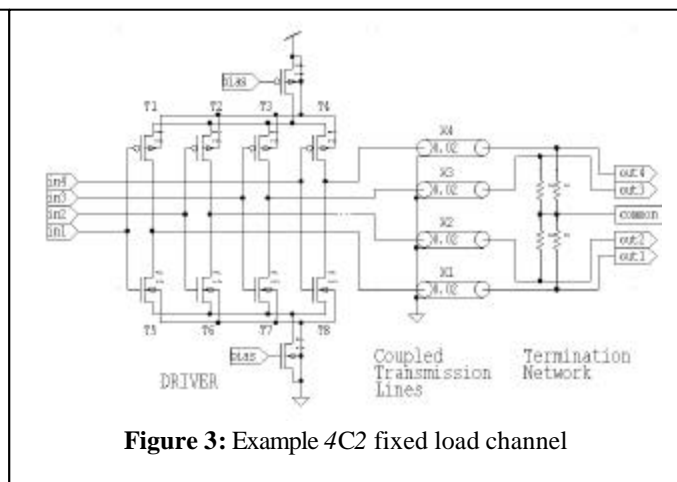
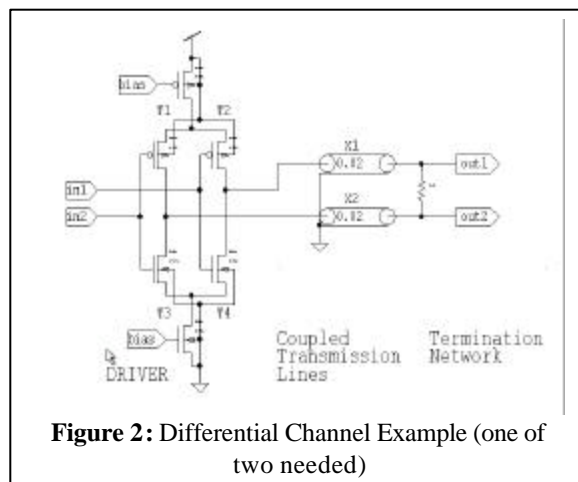
channel. This larger code set can be exploited in several ways including improvements in power efficiency and lower pad count, increased effective link bandwidth, or for low overhead ECC and channel control protocols.

For example, a 4 choose 2 (4C2) channel uses a 4-wire interconnect such that at any time exactly 2 of the wires will be energized with 1-bits. When compared to the same four wires configured as two differential channels, the 4C2 channel has roughly 25% greater information throughput capacity. This is because the 4C2 channel can transmit any of the code symbols in the set {0011,0101,0110,1001,1010,1100} while the two differential channels are restricted to the code symbols {0101,1010,0110,1001}. This corresponds to 2.5 bits of throughput for the 4C2 channel versus 2 bits for the differential channels.

Now consider the two driver circuits and example channels shown in figure 2 and figure 3. Figure 2 shows a model of a differential channel. Figure 3 is a model of a 4C2 channel. Both consist of a driver circuit, a set of coupled transmission lines, and a termination network between outputs of the transmission lines. Both of the driver circuits operate by steering a constant supply current between legs of the circuit and thus through paths in the termination network. One such path for the differential driver, corresponding to a 1-bit input, is configured such that T1 and T4 are turned on and T2 and T3 are turned off. This configuration connects the bias current supply to the bottom transmission line X2 through the termination resistor to X1 and then to ground via T4. In the opposite state, T2 and T3 turn on, T1 and T4 turn off, and the current flow reverses direction. The net effect is that the polarity of the voltage drop across the termination resistor is reversed between the two states as shown in figure 3. In a similar configuration of the 4C2 circuit for sending 0110, T1, T4, T6, and T7 are turned on and T2, T3, T5, and T8 are turned off. Current is steered through the termination network from X1 and X4 to X2 and X3. All of the resistors in the termination network have the same value and are sized to match a 100 ohm load to the transmission line. Since every code symbol is encoded with exactly two 1-bits, the common node in the center of the termination network has a constant voltage. The outputs are sensed across each resistor between the transmission line output and this common mode reference. Like the differential case, the data is sensed as the voltage polarity across this resistor and thus the two circuits can use the same type of receivers.

This comparison is important because it clearly demonstrates that fixed load channels have substantial encoding advantages yet they retain the electrical characteristics of a differential channel. These characteristics are specifically highlighted below.

- **Better power efficiency** can be achieved since nCm code symbols are encoded with fewer '1' bits used to send the same information. Given that both fixed bit-load and LVDS use current-mode logic, energy is expended for each '1' bit transmitted in order to charge the transmission lines and to maintain current flow through the termination resistors. By reducing the number of 1-bits in the encoding, we reduce the overall power consumption. The savings can be substantial. For example a 5C2 channel can send 3-bits of information per symbol using only two 1-bits per symbol encoding. An equivalent set of three differential channels requires three 1-bits. Thus the 5C2 implementation is 33% more power efficient.
- **Less silicon area and lower pad/wire count** is required for an nCm channel than an equivalent set of differential channels. The overall width of the nCm channel is less. For example a 6C3 channel can send 4-bits of information



using 6 I/O pads/wires for the link. An equivalent set of 4 differential channels uses 8 pads/wires. The 6C3 implementation uses 25% (6 versus 8) fewer I/O pads.

- **Higher Effective Bandwidth** is available because more information is delivered to the receiver per symbol. The larger symbol set of an nCm channel means that each code word imparts more information per symbol received. For example, a comparison of 4 differential channels to an 8C4 link shows that each message on the differential channel transmits 1 of 16 possible symbols, or 4bits. The 8C4 link transmits 1 of 70 possible code symbols which corresponds to over 6-bits per message. Thus, the same link has 1.5 times higher effective bandwidth.
- **Low noise in the driver circuit** comes from the current steering design of the driver that operates with constant current in all code states. A transition from one code state to the other does not introduce switching noise to the power supply.
- **Coupled transmission line behavior** will occur assuming that the link transmission lines or printed circuit board are properly designed and the signals are routed in parallel and in close proximity. The embedded set of transmission lines will be electromagnetically coupled resulting in lower loss and greater signal integrity.
- **Common mode noise rejection in a pair-wise differential receiver** is achieved because the link output signal as delivered to the receiver is a relative voltage drop across the termination resistor network. The receiver sees each encoding state as a change in the polarity of this voltage. Common mode noise appears identically on both sides of the resistor in either state. Thus, it is cancelled by differential receiver circuitry, substantially enhancing the signal to noise ratio of the output.
- **Minimal overhead ECC and Channel control** comes from two characteristics of the design. First, single bit error detection is built into the architecture and can be implemented directly in the receivers. Additional ECC and channel control codes can be implemented with no addition power or area overhead using the excess code symbols not assigned to data in the channel.

1.2. Fixed Bit-load Coding

In this section we present some basic combinatorial relationships that govern fixed bit-load encoding and show how these relationships affect the power and area requirements of selected encodings. We also introduce the notion of code utilization in fixed bit-load codes and suggest several mechanisms for exploiting unused code symbols in a fixed bit load channel.

Consider the set X such that $X_{nm} = \{x_{nm} : x \in nCm\}$. In other words, X_{nm} is the set of all code symbol encodings in an nCm channel. The size of X_{nm} , the number of available code symbols, is

$$f\{X_{nm}\} = \frac{n!}{(n-m)!m!}$$

For any value of n , the number of nCm code symbols is maximum when $m=floor(n/2)$, that is when the number of ones and zero bits are equal (or within one for odd values of n).

Regardless of the number of code symbols for the channel, each code symbol must be mapped to a binary data value at the inputs and outputs of the link. Since incoming and outgoing data will always be an integer number of binary bits, the effective bit width, bit_{eff} , the number of bits coming into and out of the channel before encoding and after decoding is given by:

$$bit_{eff} = floor(log_2(f\{X_{nm}\}))$$

Using effective bit width as a metric, Table 1 compares the relative power consumption, pad count, and code utilization for several fixed bit-load channel configurations to a set of differential channels with an equivalent effective bit width. Specifically, the relative power consumption of the two links is computed as $P_{eff} = m/bit_{eff}$, the ratio of m , the number of wires energized to '1' in the nCm channel, to bit_{eff} , the number of differential channels required to send the same information, each having one wire energized. The relative pad count is computed as $RP = n/(2*bit_{eff})$, the ratio of n , the number of wires in the nCm channel to $2*bit_{eff}$, the number of differential channels required to send the same information times two wires per channel.

From the data in the table it is clear that a 30-40% improvement in power efficiency and pad utilization can be achieved with relatively small values of n . Further, by selecting odd values of n , it is possible to trade smaller pad count reductions for greater power efficiency.

Channel type	Available Code symbols $f\{X_{mn}\}$	Effective bit width bit_{eff}	Relative Power Consumption $P_{eff} = m/bit_{eff}$	Relative Pad Count $RP = n/(2 * bit_{eff})$	% Code Utilization
2C1	2	1	100%	100%	100%
4C2	6	2	100%	100%	66%
5C2	10	3	66%	83%	80%
6C3	20	4	75%	75%	80%
7C3	35	5	60%	71%	91%
8C4	70	6	66%	66%	91%
10C5	252	7	71%	71%	51%
11C5	462	8	63%	69%	55%
12C6	924	9	66%	66%	55%

Table 1: Comparison fixed load channels to equivalent single ended channels

The rightmost column in the table is a measure of the number code symbols left over after encoding all of the $2^{bit_{eff}}$ input data words to symbols in the nCm encoding. These unused code words are available for error checking, protocol support, or other link management functions. The number of excess codes varies significantly between different channel sizes. However, the effectiveness of these codes for ECC can be greatly enhanced by encoding the ECC over multiple code words in temporal or spatial sequences. In other words, complex ECC coding can be implemented with no additional code bits and no additional channel requirements other than an nCm word sequence. Such sequences can be implemented either in time, over a single link, or in space, on a set of parallel links.

1.3. Driver Design

2, 4, 6, and 8 – channel nCm current-steering drivers were laid out for this design, using standard NFET and PFET cells from the IBM SiGe 5HP library. The 2 and 4 channel driver designs correspond to those shown in Figures 2 and 3. 6- and 8- channel drivers were designed as scaled-up versions of the 4-channel driver by adding additional current-steering “legs” for each additional output. One biasing transistor was added for every two outputs. In these designs, the goal is to size the transistors such that, for any channel in the “on” state, the corresponding driver leg must be capable of sourcing up to 10 mA of current (assuming full biasing voltage) at the maximum possible speed. Simulation results suggested that the P-transistors had to be, at the least, $W=180\ \mu\text{m}$ with the minimum channel length of $0.5\ \mu\text{m}$. Thus, all PFETs were designed to be 4-fingered with each finger being $45\ \mu\text{m}$ in width. Simulation results also suggested that the best rise/fall time was reached with the NFETs was precisely half the width of the PFETs. Thus, the NFETs were sized to be $90\ \mu\text{m}$ (2 fingers, $45\ \mu\text{m}$ each). These transistor sizes are consistent throughout the cells, including biasing transistors.

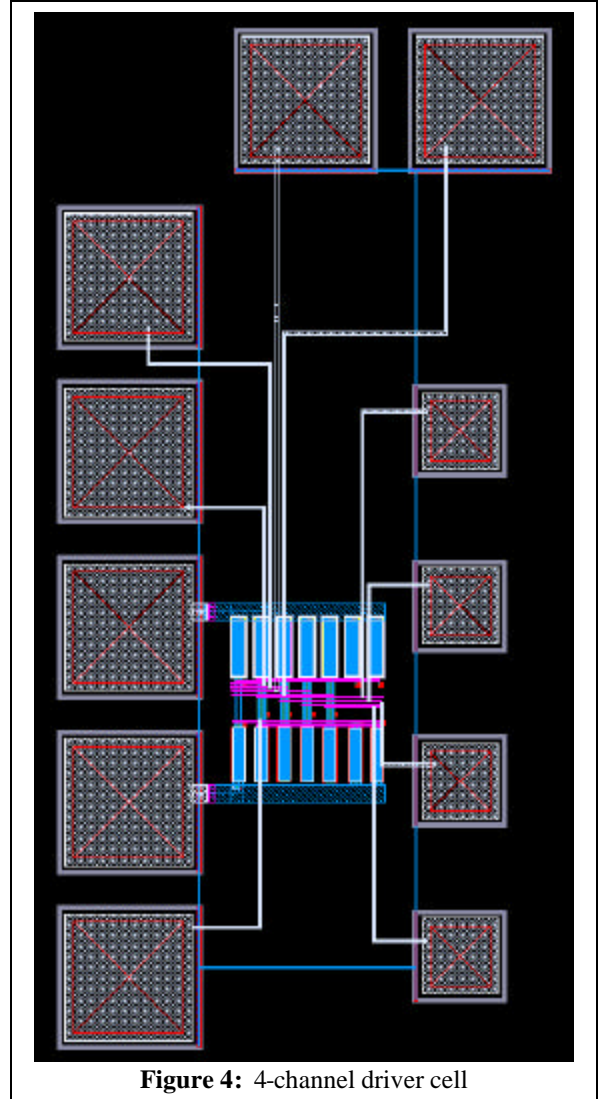


Figure 4: 4-channel driver cell

The drivers are physically situated on the periphery of the die, as close to the wirebond pads as possible. The outputs of the driver are connected to standard 110 μm square wirebond pad cells. Power and ground for each driver is brought in from wirebond pads as well. The 6- and 8- channel drivers have one additional redundant power and ground connection. The CMOS-level digital inputs to the drivers are provided by 65 μm square probe pads, which are located within the driver cells. Figure 4 shows the layout for the 6-channel driver.

1.4. Receiver Design

The schematic for the differential receiver is shown in Figure 6. The receiver conforms to the LVDS standard [1]. The transistor sizes for the receiver match those of the driver. 2-, 4-, 6-, and 8-channel receivers were designed by instantiating one differential receiver cell for each channel. The receiver cells amplify the differential signal created between any particular channel and the common point in the resistor network any the corresponding nCm link. According to the LVDS standard, the differential voltage of the input may be as low as 100 mV. The inputs to the receivers (including the channels and common point connection) are formed using the standard 110 μm square wirebond pad cells, aligned on the periphery of the die. As with the drivers, power and ground are also connected to the wirebond pads. The 6- and 8- channel receiver have redundant power and ground connections. The digital outputs of the receivers are sampled using 65 μm square probe pads. The layout for the 4-channel receiver is shown in Figure 5.

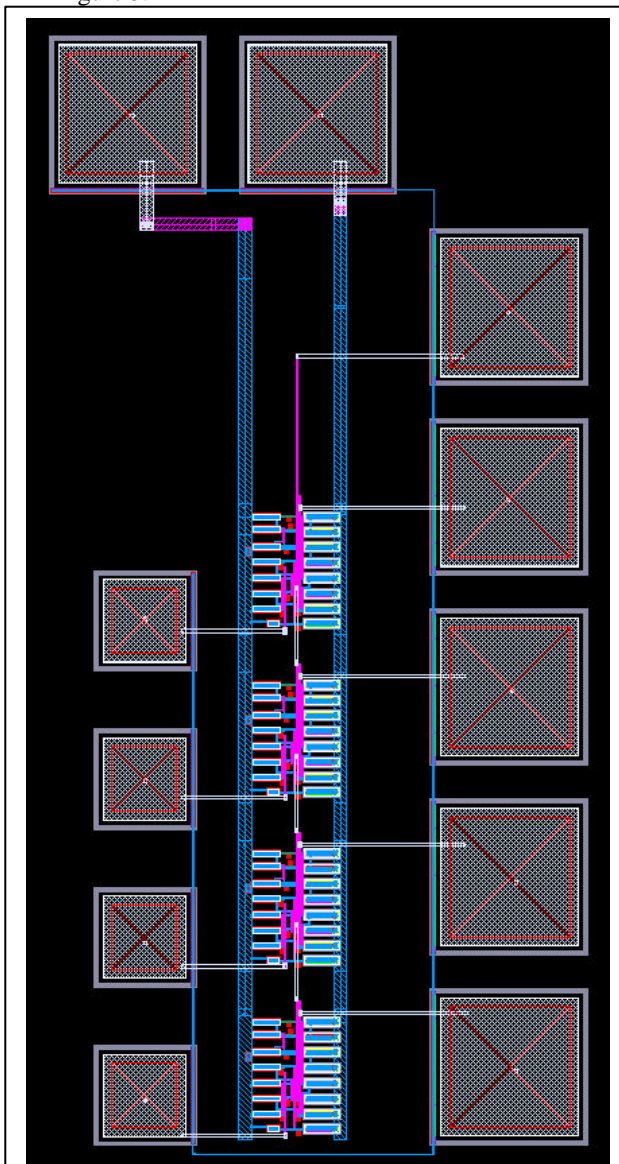


Figure 5: Layout of 4-channel receiver

The main disadvantage of this receiver design is that it is significantly slower than the driver designs. In future work, our goal is to test faster differential receiver designs with the fixed-load drivers.

1.5. Simulation Setup and Results

We have made preliminary investigations into two of the areas listed above. In the first study, we designed and simulated several implementations of nCm channels and tested the relative bandwidth of each for a 25cm (10in) channel length. In the second study we have done a static electric field analysis of three coupled transmission line structures as candidates for PCB architectures (as topological plots of the field intensity). In the third study, we study the effect of noise on the bit-error-rate of a 4C2 link.

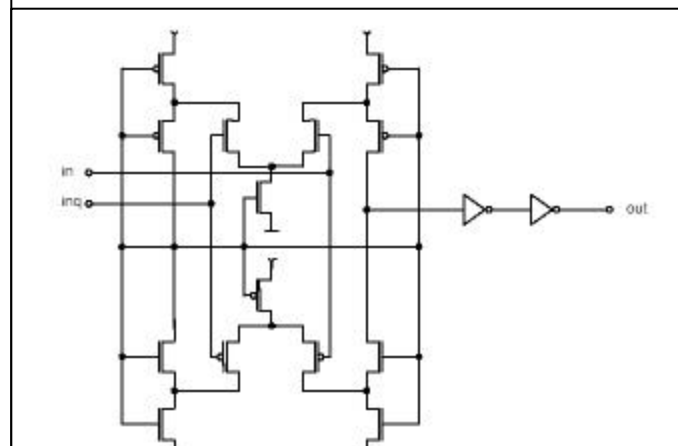


Figure 6: 2-input differential receiver [1]

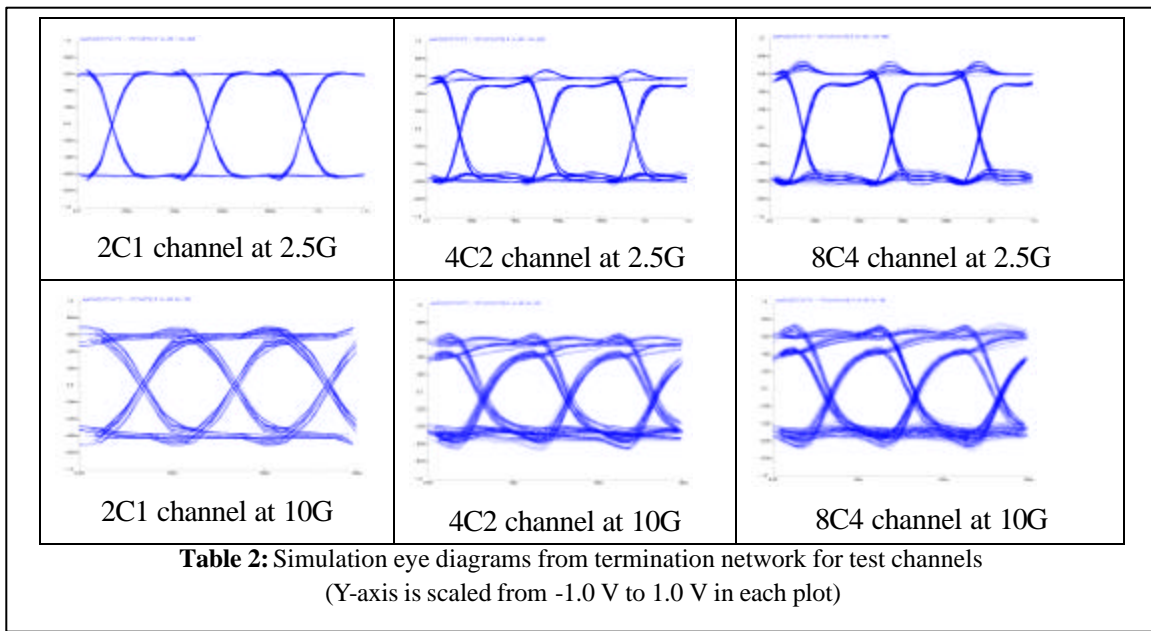
1.5.1. Driver Bandwidth Testing for 2C1, 4C2, and 8C4 Links

Figure 3, above, shows an example of a 4C2 link. Like the differential case, this driver circuit operates in a “current steering” mode with the total driver current regulated by the biasing transistor. Thus, for any code word, exactly two of the p-transistors, T1-T4 will be turned on and two off. Each of the n-transistors T5 through T8 will be set to the complement of its partner. In any such configuration the driver will source current to exactly two of the four transmission lines and sink current from the other two.

The termination network at the output of the transmission lines terminates all of the lines to single node through the termination resistors R1-R4. All four resistors have the same value and this design presents the same termination load to all of the transmission lines regardless of state.

As shown in figure 3 above, the voltage states for zero and one reverse polarity depending on whether the driver is sourcing or sinking current through the transmission line associated with the channel. Since the common node of the termination network is balanced between and connected to all of the other outputs, common mode noise rejection is still available at the differential receiver input. In fact, each of the outputs is fully compatible with commercial standard LVDS receivers. This is an important feature since it suggests that receiver circuitry for these links will be scalable to wide (multi-bit) links with no loss of performance.

In our simulation we compare the performance of a 2C1, 4C2, and 8C4 links for 2.5Gb/s and 10Gb/s. In this study, we simulated the driver schematics. Table 2 shows eye-diagrams for our 2-, 4-, and 8-channel driver designs. Input was a random sequence of valid code words. In the simulation, we can see the effects of increased link width on performance due to the parasitic capacitance associated with the biasing transistor(s) and the fanouts at the drains of the n-transistors and p-transistors. The plots are based on Cadence Spectre simulation runs for .5um IBM SiGe HP5 transistors for the driver circuits, and simulation models built into Spectre for the transmission lines and termination network. The significance of these results is the comparable performance at each bandwidth between the differential (2C1) channel and the 4C2 and 8C4 channels. This suggests that there is little degradation in performance for the wider channels.



1.5.2. Static Field Analysis for 2C1 and 4C2 Links

In a second study we have looked at the static electric field distribution for a conventional 2-conductor coupled transmission line and two possible configurations of a 4-conductor link in a 4C2 channel. These results are shown in Table 3. Symmetrical modes are not shown. For example for the 2C1 case (a standard 2-conductor couple transmission line) there is symmetrical field, for both states. In the 4 line case, the linear structure has a strong asymmetry between the modes that would likely result in variations between the individual line coupling based on code sequences. The simple 2D structure introduces symmetry in two of the three modes, but is still not completely symmetrical.

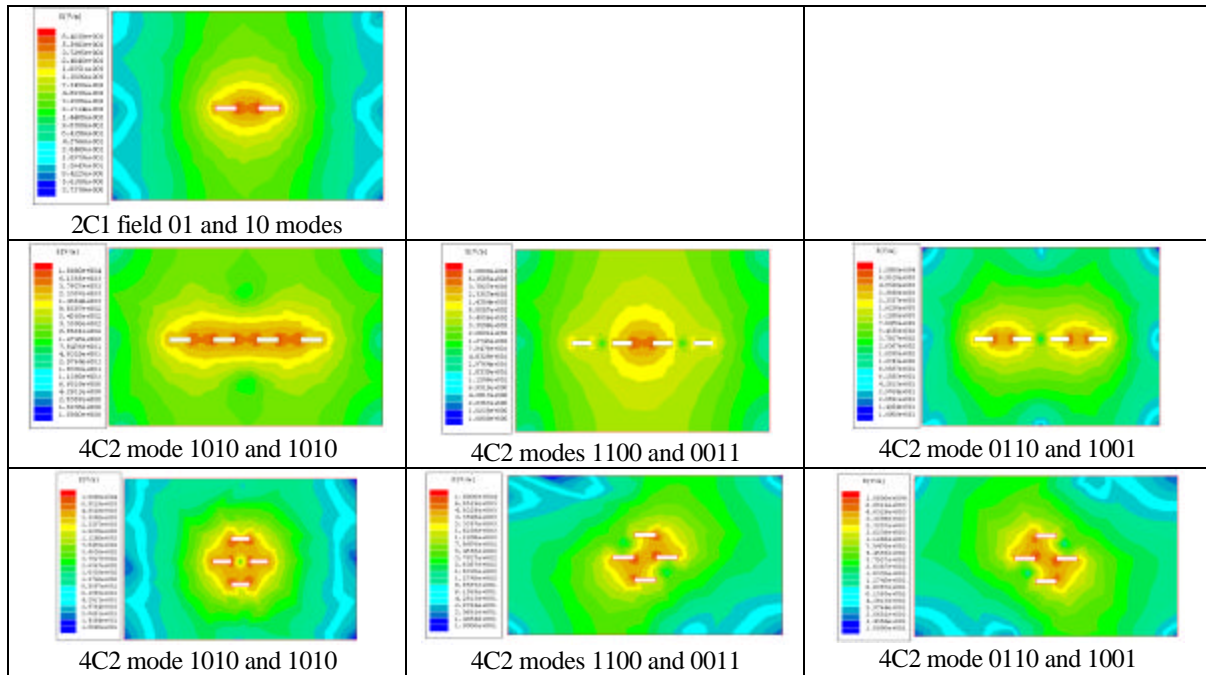


Table 3: Static field topologies for selected links and coupled transmission line designs

1.5.3. Effects of Common-mode Noise on Bit-Error Rate of 4C2 Link

In a third study we looked at the effect of common-mode noise on the bit error rate of a 4C2 link. We set up a simulation model that includes a complete link, including extracted layout netlists of a 4C2 driver and 4C2 receiver, along with an ideal resistor termination network on the input of the receiver and a generated 2D stripline coupled transmission line model between the driver and the termination network. The receiver netlists were included in the simulation for loading purposes only. The results of the simulation are the transient responses of the receiver input voltages only. This simulation assumes that the receivers are ideal LDVS receivers that can support up to 10 Gbs sampling.

The transmission line topology matches that of the second row of Table 3. The transmission line model was setup to be 25 cm lossless striplines. The generated model includes complete capacitance and inductance matrices for each of the four transmission lines. This model was created using Cadence's transmission line modeler.

We ran simulations for a consistent set of 10,000 random codewords at various bias voltages and speeds. We recorded the number of bit errors in each symbol as a function of latency (transmission delay). In this case, latency is the difference in time from when the output is sampled to when the input is sampled. This is illustrated in Figure 7.

Bit errors are modeled according to the LVDS-standard [1], which states that in order for a bit to be transmitted correctly, the voltage difference must be $\geq 100\text{mV}$ for a 1-bit and $\leq -100\text{mV}$ for a 0-bit at the time of sampling.

We repeated these simulations after adding white Gaussian noise to the bias input of the drivers (to simulate the addition of common-mode noise). The noise source follows a Normal distribution with a mean of 0 and a standard deviation of 1. This noise is multiplied by 5 before being added to the bias voltage for the driver. This yields a noise which typically ranges between 10 mV and 50 mV, which equates to approximately 1-5% of the total bias voltage.

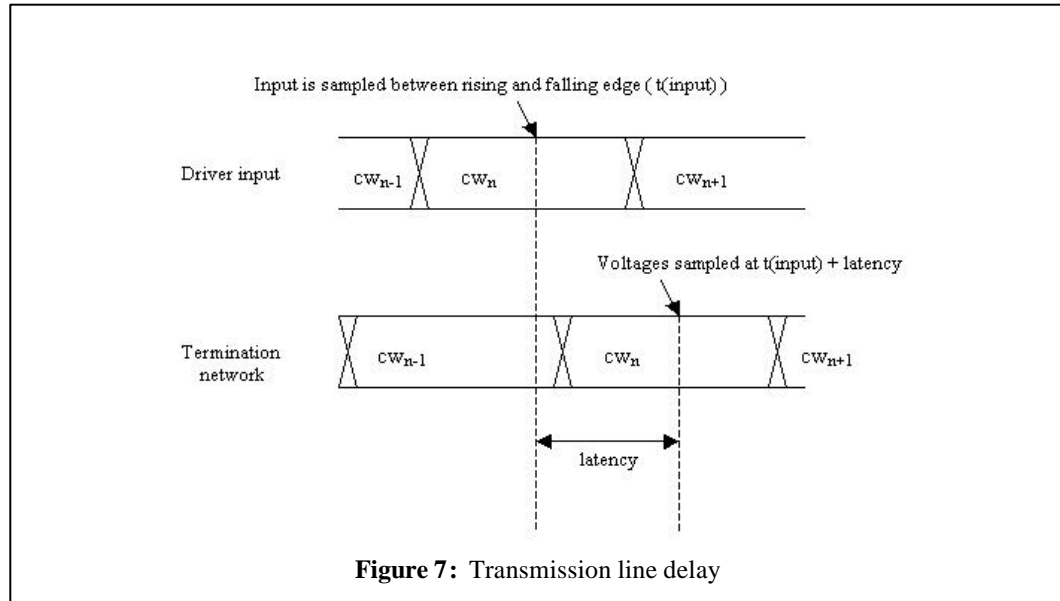


Figure 7: Transmission line delay

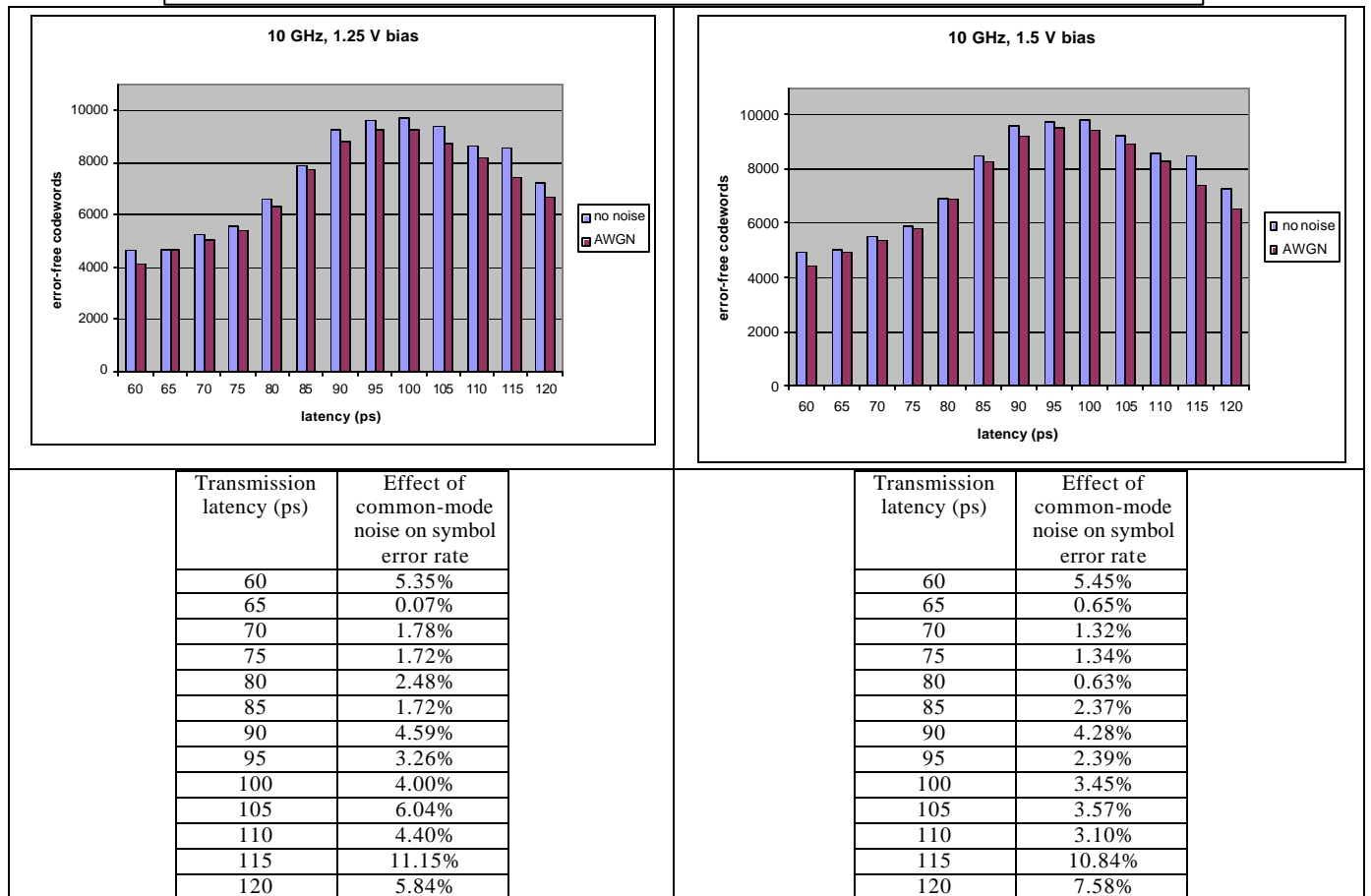


Table 4: Effect of noise on channel bit error rate

Table 4 shows the results of the bit error rate simulations. The charts show the number of error-free codewords vs. the latency at which the results were sampled. The table underneath each plot shows the relative difference between the number of error-free code words in the non-noise and noise simulations. This value is computed as:

$$(cw_correct_{no_noise} - cw_correct_{noise}) / total_codewords.$$

These simulations show that the effect of common-mode noise on the bit error rate of the link was very low, with a $\leq 4\%$ difference for optimal latencies for all simulations.

2. Conclusions and Future Work

We have successfully designed and simulated the fixed-load drivers described in this paper. Simulation results and analysis indicate that all of the design goals listed in section 1.1 (those that were explicitly tested) have been met.

The test die has been manufactured. However, at the time of writing, we have not completed assembly of our prototype PCB that will allow us to collect test results from multiple chip-to-chip configurations transmission line topologies. Once this is completed, we expect to have extensive test results from our prototype system.

3. References

[1] Stefan Hirsch and Hans-Jörg Pfeleiderer, “*CMOS receiver circuits for high-speed data transmission according to LVDS-standard*” Proceedings of SPIE Vol. 5117 (2003).