



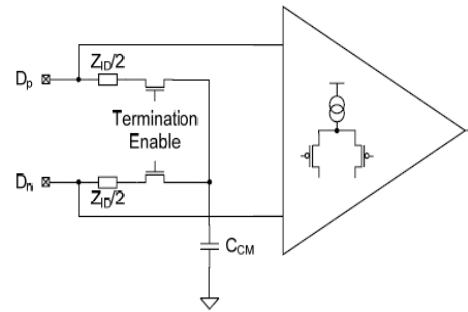
High Speed Differential Signal Receiver IP for PHY

RCRX01

Description

The RCRX01 is LVDS receiver IP designed using 0.18um CMOS process. It can be configured and interfaced with other CMOS PHY IPs using multilevel signaling and supports data or clock signals reception up to 1Gbps (max). The LVDS receiver accepts differential input levels and translates them into standard single ended output. The receiver receives High-Speed data correctly while rejecting common-mode interference. During operation of the receiver, termination impedance is required between the Dp and Dn pins of the receiver.

Functional Diagram



Applications

- High speed receivers
- High speed PHY
- Clock Drivers
- Camera and Display Tx/Rx PHY

Key Features

- Low Power CMOS Design
- On chip terminations
- Reconfigurable for use in various high speed PHY designs
- Supports reduced swing LVDS for low EMI
- Portable to other CMOS foundries



High Speed Differential Signal Receiver IP for PHY

RCRX01

AC Specification

Parameter	Description	Min	Nom	Max	Units	Notes
$\Delta V_{CMRX(HF)}$	Common-mode interference beyond 450 MHz			100	mV	2
$\Delta V_{CMRX(LF)}$	Common-mode interference 50MHz – 450MHz	-50		50	mV	1, 4
C_{CM}	Common-mode termination			60	pF	3

DC Specification

Vcc = 1.8 V ± 10%; TA = -40°C to 125°C

Parameter	Description	Min	Nom	Max	Units
$V_{CMRX(DC)}$	Common Mode Voltage HS Rx Mode	70		330	mV
V_{IDTH}	Differential input high Threshold			70	mV
V_{IDTL}	Differential input low Threshold	-70			mV
V_{IHHS}	Single-ended input high Threshold			460	mV
V_{ILHS}	Single-ended input low Threshold	-40			mV
V_{TERM_EN}	Single-ended threshold for HS termination enable			450	mV
Z_{ID}	Differential input impedance	80	100	125	Ω

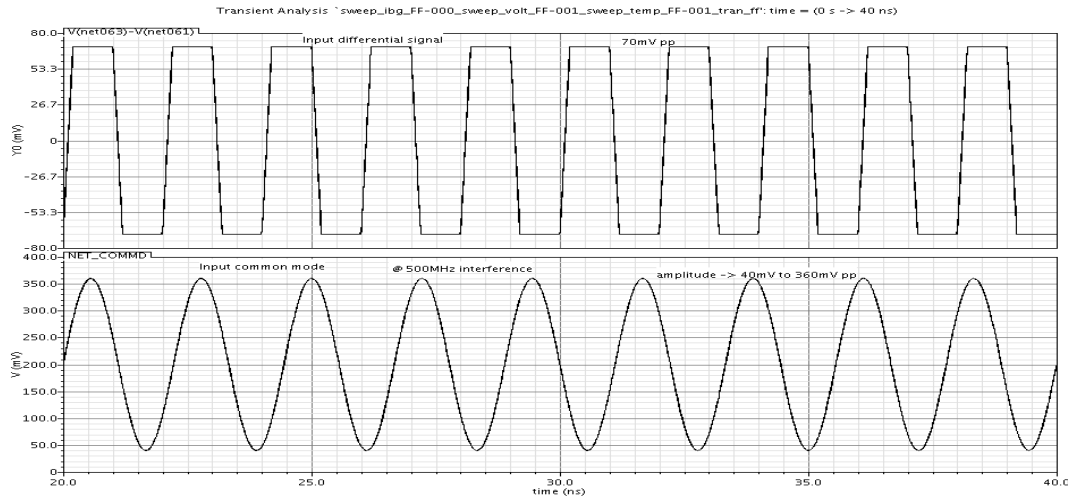


High Speed Differential Signal Receiver IP for PHY

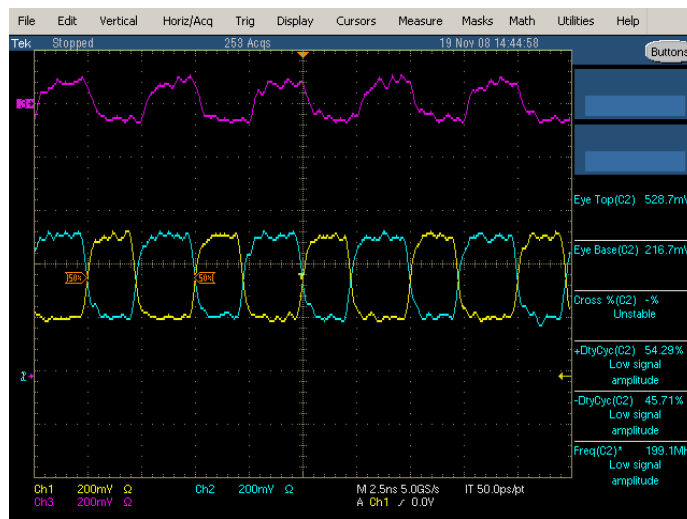
RCRX01

Simulation Result

The result is shown for 70mV differential input about common mode interference Of 340mV peak to peak at 500MHz interference frequency.



Silicon data:



Differential signal input (Dp-Dn) and output