FEATURES
- Fully compliant with latest USB2.0 spec version 2.0.
- Innovative technique to recover clock from 480Mbps data.
- High frequency PLL.
- Advanced high-speed transmitter and receiver.
- Support High Speed HS, Full Speed FS modes.
- Interface with standard SIE.

APPLICATIONS
- High-speed serial link at 12Mbps, 480Mbps.

GENERAL DESCRIPTION
USB 2.0 transceiver is a fully integrated Clock Data Recovery with on-chip PLL. USB2 has standard UTMI with SIE so that other ASIC vendors are isolated from the high speed and analog circuitry associated with the transceiver, thus reducing the design risk and fastening the design cycle.

The chip has 5 major blocks: clock data recovery for FS/HS, PLL, transceiver state machines, data encoder/decoder and high-speed analog transceiver.

**Figure 1. USB2 Transceiver Block Diagram**
**Functional Description**

As shown in Figure 1, the USB2 transceiver’s main responsibility is to transmit data onto the line and to receive data and to recover clock correctly from the receive data. The front interface of USB2 transceiver is the cable, the back interface of USB2 transceiver is the UTMI to the serial interface engine that handles packet recognition, transaction sequencing, ...

### Table 1: System Interface Signals

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Active</th>
<th>Description</th>
</tr>
</thead>
</table>
| CLK        | Output    | Rising-Edge | **Clock.** This output is used for clocking receive and transmit parallel data.  
|            |           |         | 60 MHz HS/FS or HS Only, with 8-bit interface  
|            |           |         | 30 MHz HS/FS or HS Only, with 16-bit interface  
|            |           |         | 48 MHz FS Only, with 8-bit interface  
|            |           |         | 6 MHz LS Only, with 8-bit interface  |
| Reset      | Input     | High    | **Reset.** Reset all state machines in the UTM.                             |
| XcvrSelect | Input     | N/A     | **Transceiver Select.** This signal selects between the FS and HS transceivers:  
|            |           |         | 0: HS transceiver enabled  
|            |           |         | 1: FS transceiver enabled  
|            |           |         | This signal is not provided in HS Only and LS Only transceiver implementations. |
| TermSelect | Input     | N/A     | **Termination Select.** This signal selects between the FS and HS terminations:  
|            |           |         | 0: HS termination enabled  
|            |           |         | 1: FS termination enabled  
|            |           |         | This signal is not provided in HS Only and LS Only transceiver implementations. |
| SuspendM   | Input     | Low     | **Suspend.** Places the Macrocell in a mode that draws minimal power from supplies. Shuts down all blocks not necessary for Suspend/Resume operation. While suspended, **TermSelect** must always be in FS mode to ensure that the 1.5K pull-up on DP remains powered.  
|            |           |         | 0: Macrocell circuitry drawing suspend current  
|            |           |         | 1: Macrocell circuitry drawing normal current  |
| LineState(0,1) | Output | N/A | **Line State.** These signals reflect the current state of the single-ended receivers. They are combinatorial until a "usable" CLK is available then they are synchronized to CLK. They directly reflect the current state of the DP (LineState[0]) and DM (LineState[1]) signals:  
|            |           |         | DM DP Description  
|            |           |         | 0 0 0: SE0  
|            |           |         | 0 1 1: ‘J’ State  
|            |           |         | 1 0 2: ‘K’ State  
|            |           |         | 1 1 3: SE1  |
| OpMode(0-1) | Input | N/A | **Operational Mode.** These signals select between various operational modes:  
|            |           |         | [1] [0] Description  
|            |           |         | 0 0 0: Normal Operation  
|            |           |         | 0 1 1: Non-Driving  
|            |           |         | 1 0 2: Disable Bit Stuffing and NRZI encoding  
|            |           |         | 1 1 3: Reserved  |

### Table 2: USB Interface Signals

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Active</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>Bidir</td>
<td>N/A</td>
<td>USB data pin Data+</td>
</tr>
</tbody>
</table>
## Table 3: Data Interface Signals (Transmit)

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Active</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataIn0-7</td>
<td>Input</td>
<td>N/A</td>
<td>DataIn: 8-bit parallel USB data input bus. When DataBus16_8 = 1 this bus transfers the low byte of 16-bit transmit data. When DataBus16_8 = 0 all transmit data is transferred over this bus.</td>
</tr>
<tr>
<td>DataIn8-15</td>
<td>Input</td>
<td>N/A</td>
<td>DataIn: An 8-bit parallel USB data input bus that transfers the high byte of 16-bit transmit data. These signals are only valid when DataBus16_8 = 1.</td>
</tr>
<tr>
<td>TXValid</td>
<td>Input</td>
<td>High</td>
<td>Transmit Valid. Indicates that the DataIn bus is valid. The assertion of Transmit Valid initiates SYNC on USB. The negation of Transmit Valid initiates EOP on USB.</td>
</tr>
<tr>
<td>TXValidH</td>
<td>Input</td>
<td>High</td>
<td>Transmit Valid High. When DataBus16_8 = 1, this signal indicates that the DataIn(8-15) bus contains valid transmit data. This signal is ignored when DataBus16_8 = 0. This signal is not provided in 8-Bit transceiver implementations.</td>
</tr>
<tr>
<td>TXReady</td>
<td>Output</td>
<td>High</td>
<td>Transmit Data Ready. If TXValid is asserted, the SIE must always have data available for clocking in to the TX Holding Register on the rising edge of CLK. TXReady is an acknowledgement to the SIE that the UTM has clocked the data from the bus. This acknowledgement also indicates to the SIE that it must present the data for the next transfer on the bus. If TXValid is negated, TXReady can be ignored by the SIE.</td>
</tr>
</tbody>
</table>

## Table 4: Data Interface Signals (Receive)

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Active</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataOut0-7</td>
<td>Output</td>
<td>N/A</td>
<td>DataOut: 8-bit parallel USB data output bus. When DataBus16_8 = 1 this bus transfers the low byte of 16-bit receive data. When DataBus16_8 = 0 all receive data is transferred over this bus.</td>
</tr>
<tr>
<td>DataOut8-15</td>
<td>Output</td>
<td>N/A</td>
<td>DataOut: An 8-bit parallel USB data output bus that transfers the high byte of 16-bit receive data. These signals are only valid when DataBus16_8 = 1.</td>
</tr>
<tr>
<td>RXValid</td>
<td>Output</td>
<td>High</td>
<td>Receive Data Valid. Indicates that the DataOut bus has valid data. The Receive Data Holding Register is full and ready to be unloaded. The SIE is expected to latch the DataOut bus on the clock edge.</td>
</tr>
<tr>
<td>RXValidH</td>
<td>Output</td>
<td>High</td>
<td>Receive Data Valid High. When DataBus16_8 = 1 this signal indicates that the DataOut(8-15) bus is presenting valid receive data. This signal is ignored when DataBus16_8 = 0. This signal is not provided in 8-Bit transceiver implementations.</td>
</tr>
<tr>
<td>RXActive</td>
<td>Output</td>
<td>High</td>
<td>Receive Active. Indicates that the receive state machine has detected SYNC and is active. RXActive is negated after a Bit Stuff Error or an EOP is detected. RXError Output High Receive Error. 0 Indicates no error. 1 Indicates that a receive error has been detected.</td>
</tr>
</tbody>
</table>
USB2FE
Analog front end USB2FE includes high-speed differential driver and envelope detection/disconnection differential receiver.

**HIGH-SPEED RECEIVER—Data Sheet**

The high-speed receiver primarily consists of three functional elements:
1) A differential data receiver.
2) A transmission envelope detector.
3) A disconnection envelope detector.

The differential data receiver receives the high-speed (480Mb/s) differential data from the USB and converts it to a single-ended NRZI signal (DATA_OUT) suitable for use as an input to standard CMOS digital cells. The receiver is disabled and powered down (with digital output forced to “0”) when either the ENABLE input to the high-speed receiver is set to “0” or the output of the transmission envelope detector is “0.”

The transmission envelope detector monitors the amplitude of the high-speed differential data signal from the USB. A differential amplitude greater than 150mVp, indicating valid data, sets the digital output ENV_DET to “1,” while an amplitude less than 100mVp, indicating invalid data, resets ENV_DET to “0.”

Fig. 2: Block diagram of high-speed receiver
The disconnection envelope detector also monitors the amplitude of the differential data signal from the USB, albeit with a somewhat different operation and different signal levels of interest. A differential signal amplitude greater than 625mV, indicating disconnection, forces output DISCONN to “1,” while a differential amplitude less than 525mV causes DISCONN to remain at “0.” The disconnection envelope detector is powered down (with DISCONN forced to “0”) when the ENABLE input is set to “0.”

**HIGH-SPEED CURRENT DRIVER—Data Sheet**

![Block diagram of high-speed current driver](image)

**Fig. 3: Block diagram of high-speed current driver**

The high-speed current driver is used for high-speed data transmission. For those times when the driver is not transmitting, a standby mode (set by DRIV_ENA=0) is available which allows the 17.78mA current to be directed internally to ground (AGND1) and places the driver in a high-output-impedance state. The transition between standby and non-standby modes does not exhibit much delay, and use of the standby/non-standby modes ensures that the driver meets the required accuracy starting with the first symbol of a packet.

**HSDLL**

The HSDLL is a delay line, characterized to sample data at 480Mhz to detect the data transition. This information is used by a HSDLL state machine to position recover clock at the midpoint of the data pulse to assure optimal setup and hold time. To further correct the clock discrepancy, which can be up to 500ppm between transmitter and receiver, a 24-bit elasticity buffer is used to store enough data in it to avoid overflow or underflow. A 12-bit threshold is reached before the receiver state machine starts using the data.

**FSDLL**

FS DLL is a 4-time over-sampling clock recovery at 48Mhz to recover 12Mbps data. The HS detection circuit detects what speed the downstream port is. Once detected, either FS data or HS data are multiplexed to be NRZI-decoded back into NRZ data for receive state machine.
TRANSCEIVER STATE MACHINES

The transceiver state machines follow the handshake protocol from UTMI to interface with SIE. The state machines also control NRZI encoder/decoder and bit stuffer/unstuffer for data correctness.

RECEIVE STATE MACHINE

RESET: The assertion of reset signal forces the Receive State Machine into RESET state.

RXWAIT: When reset signal is gone, the Receive State Machine goes to RXWAIT state to wait for SYNC pattern.

STRIP SYNC: When a SYNC is detected, STRIP SYNC starts filter out SYNC pattern until the first valid data.

RXWAIT: When reset signal is gone, the Receive State Machine goes to RXWAIT state to wait for SYNC pattern.

RXDATA: After 8 bits of valid serial data is received the state machine enters the RXDATA state, where the data is loaded into the RX Holding Register on the rising edge of CLK and rxvalid is asserted. The SIE must clock the data out on the next rising edge of CLK.

RXDWAIT: Each time 8 stuffed bits are accumulated the state machine will enter the RXDWAIT state, negating rxvalid thus skipping a byte time.

STRIP EOP: When the EOP is detected the state machine will enter the STRIP EOP state and negate rxactive and rxvalid. After the EOP has been stripped the Receive State Machine will reenter the RXWAIT state and begin looking for the next packet.

ERROR: At any time if there’s an error due to bit stuff error, buffer overflow, buffer underflow, loss of sync, alignment error, … the state machine will enter ERROR state.

ABORT: Abort current packet due to receive error.

Figure 4: Receive State Diagram
TRANSMIT STATE MACHINE

RESET: The assertion of reset signal forces the Transmit State Machine into RESET state.

TXWAIT: When reset signal is gone, the Transmit State Machine goes to TXWAIT state to wait for txvalid.

SENSSYNC: When txvalid is detected, SENDSYNC starts sending out SYNC pattern.

TXDATA: When the transmitter is ready for the first byte of the packet (PID), it will enter the TXDATA state, assert txready and load the TX Holding Register.

TXDWAIT: When there is no more data, the state machine goes to TXDWAIT and remains there until the TX Hold register is available for more data.

SENDEOP: When txvalid is negated the transmit state machine enters the SENDEOP state where it sends the EOP. After the EOP is transmitted the Transmit State Machine returns to the TXWAIT state, looking for more data.

PLL
PLL requires a 48Mhz or 24 MHz (12 MHz is also available if required) crystal clock as a reference and produces 480Mhz clock for the HSDL portion of the USB2 transceiver. This 480Mhz clock is further divided into 60Mhz, 48Mhz, 30Mhz, 12Mhz, 6Mhz and 1.5Mhz for the FS, LS portions and UTMI interface. These 60Mhz, to 6Mhz clocks have a frequency accuracy of 500ppm and duty cycle accuracy of 1%.