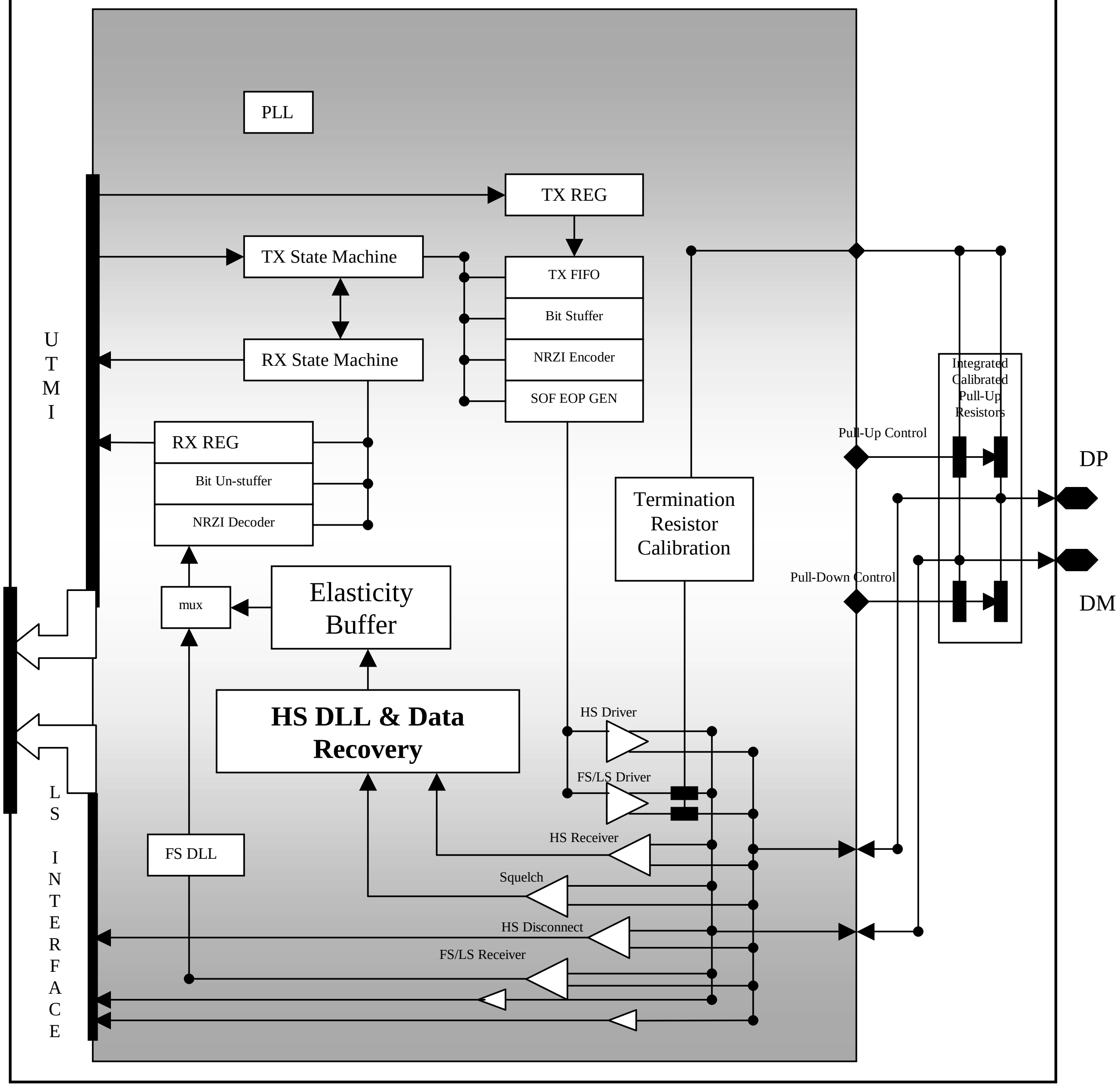
# SMS2100 USB 2.0 Device Transceiver

**SMS USB 2.0 HS/FS DEVICE PHY with optional LS feature**



## FEATURES

- Fully compliant with latest USB 2.0
- Innovative technique to recover clock from 480 Mbps data.
- High frequency PLL.
- Advanced High-Speed Transmitter and Receiver.
- Support High Speed HS, Full Speed FS, Low Speed LS modes.
- Integrated/Calibrated Termination Resistors
- Legacy USB 1.1 Interface

## GENERAL DESCRIPTION

USB 2.0 DEVICE Transceiver is a fully integrated PHY Core which is a super-set of DEVICE PHY with High Speed (HS), Full-Speed (FS) and Low-Speed Transceivers and is compliant with the USB 2.0 and USB 1.1 Specifications. It includes Clock/Data Recovery, on-chip PLL, Integrated & Calibrated Termination and Pull-Up Resistors with full Analog Transceiver functionality for the Complete USB 2.0 PHY as illustrated in the figure.
USB2 DEVICE Transceiver has standard UTMI so that ASIC vendors are isolated from the high speed and analog circuitry associated with the transceiver, thus reducing the design risk and speeding the design cycle.

The core's main blocks are clock/data recovery for FS/HS, PLL, transceiver state machines, data encoder/decoder and high-speed analog transceiver as can be seen in the main block diagram above.

### Functional Description

As shown in main block diagram, 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, serialization/de-serialization, bit stuffing/un-stuffing and other relevant functionality needed by a USB 2.0 Device PHY.

**Table 1: System Interface Signals**

| Name | Direction | Active | Description |
|---|---|---|---|
| CLK | Output | Rising-Edge | **Cloc**k. This output is used for clocking receive and transmit parallel data.<br>60 MHz HS/FS or HS Only, with 8-bit interface<br>30 MHz HS/FS or HS Only, with 16-bit interface |
| Reset | Input | High | **Reset.** Reset all state machines in the UTM. Reset doesn't stop the CLK output.<br>Reset Signal Should be activated simultaneously with SuspendM signal and It should be deactivated 1ms after SuspendM signal. See Figure 2. |
| XcvrSelect | Input | N/A | **Transceiver Select.** This signal selects between the FS and HS transceivers:<br>0: HS transceiver enabled<br>1: FS transceiver enabled |
| TermSelect | Input | N/A | **Termination Select.** This signal selects between the FS and HS terminations:<br>0: HS termination enabled<br>1: FS termination enabled |
| SuspendM | Input | Low | **SuspendM.** 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. Suspend signal stops the CLK. When Suspend is activated, CLK stops at its low level. After Suspend is deactivated, CLK starts with at least 40% Duty Cycle in less than 4ms.<br>0: Macrocell circuitry drawing suspend current<br>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**. **DP=**LineState[0] & **DM=**LineState[1]<br>DM DP Description<br>0 0 0: SE0<br>0 1 1: 'J' State<br>1 0 2: 'K' State<br>1 1 3: SE1 |
| OpMode(0-1) | Input | N/A | **Operational Mode.** These signals select between various operational modes:<br>[1] [0] Description<br>0 0 0: Normal Operation<br>0 1 1: Non-Driving<br>1 0 2: Disable Bit Stuffing and NRZI encoding<br>1 1 3: Reserved |

**Table 2: USB Interface Signals**

| Name | Direction | Active | Description |
|---|---|---|---|
| DP | Bidir | N/A | USB data pin Data+ |
| DM | Bidir | N/A | USB data pin Data– |
| Mode | Input | High | UTMI/USB 1.1 Mode Switch (1 == UTMI) |
| OEB | Input | Low | Output Enable for Full/Low Speed during USB 1.1Mode |
| Speed | Input | High | Full/Low Speed Selection During USB 1.1 Mode (1 == FS, 0 == LS)<br>In USB 1.1 Device Mode (Mode=0) Speed also Selects which Data Line will be Pull-Up for FS/LS Device Termination. |
| VMO | Input | N/A | Single Ended Data Driver Input USB 1.1 Mode |
| VPO | Input | N/A | Single Ended Data Driver Input USB 1.1 Mode |
| RCV | Output | N/A | Differential Receiver Output For USB 1.1 Mode |
| VM | Output | N/A | Single Ended Receiver Output For USB 1.1 Mode |
| VP | Output | N/A | Single Ended Receiver Output For USB 1.1 Mode |

**Table 3: Data Interface Signals (Transmit)**

| Name | Direction | Active | Description |
|---|---|---|---|
| DataIn0-7 | Input | N/A | **DataI**n. 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. |
| DataIn8-15 | Input | N/A | **DataI**n. 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. |
| TXValid | Input | High | **Transmit Vali**d. Indicates that the **DataIn** bus is valid. The assertion of Transmit Valid initiates SYNC on USB The negation of Transmit Valid initates EOP on USB. |
| TXValidH | Input | High | **Transmit Valid Hig**h. 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. |
| TXReady | Output | High | **Transmit Data Read**y. If **TXValid** is asserted, the SIE must always have data available for clocking in to the TX Holding Register on the rising edge of **CL**K. **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 SIE. |

**Table 4: Data Interface Signals (Receive)**

| Name | Direction | Active | Description |
|---|---|---|---|
| DataOut0-7 | Output | N/A | **DataOu**t. 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. |
| DataOut8-15 | Output | N/A | **DataOu**t. 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. |
| RXValid | Output | High | **Receive Data Vali**d. 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. |
| RXValidH | Output | High | **Receive Data Valid Hig**h. 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. |
| RXActive | Output | High | **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.**<br>0 Indicates no error.<br>1 Indicates that a receive error has been detected. |

**Table 5: Data Interface Signals (Other)**

| Name | Direction | Active | Description |
|---|---|---|---|
| DataBus16_8 | Input | High | **DataBus16_8**. Selects between 8 and 16 bit data transfers.<br>1 16-bit data path operation enabled. DataIn(8-15), DataOut(8-15),TXValidH,RXValidH operational. CLK=30MHz<br>0 8-bit data path operation enabled. DataIn(8-15), DataOut(8-15), TXValidH,RXValidH undefined. CLK=60MHz. |

**Table 6: System Interface Signals**

| Name | Direction | Active | Description |
|---|---|---|---|
| RefClk | Input | Pos-Edge | Reference Clock For PLL Input<br>48 MHz or 24 MHz (12 MHz is available if required) |
| Rext | Input | N/A | Bias Resistor Pin Should be connected to 12.3K %1 |
| VoREF | Output | N/A | Voltage Reference output **OPTIONAL** |
| Atest0 | Output | N/A | Analog Test Output 0 **OPTIONAL** |
| Atest1 | Output | N/A | Analog Test Output 1 **OPTIONAL** |

**Table 7: Powers**

| Name | Direction | Active | Description |
|---|---|---|---|
| AVDDRX | Input | N/A | Analog VDD for Receive |
| AVSSRX | Input | N/A | Analog VSS for Receive |
| AVDDTX | Input | N/A | Analog VDD for Transmit |
| AVSSTX | Input | N/A | Analog VSS for Transmit |
| AVDDC | Input | N/A | Analog VDD for Common **OPTIONAL** |
| AVSSC | Input | N/A | Analog VSS for Common **OPTIONAL** |
| DVDD | Input | N/A | Digital VCC From Core |
| DVSS | Input | N/A | Digital VSS From Core |

Note: Receive & Transmit Powers can be merged if necessary.

## USB2FE

Analog front end USB2FE includes high-speed differential driver and envelope detection/disconnection differential receiver.

**HIGH-SPEED RECEIVER—Data Sheet**

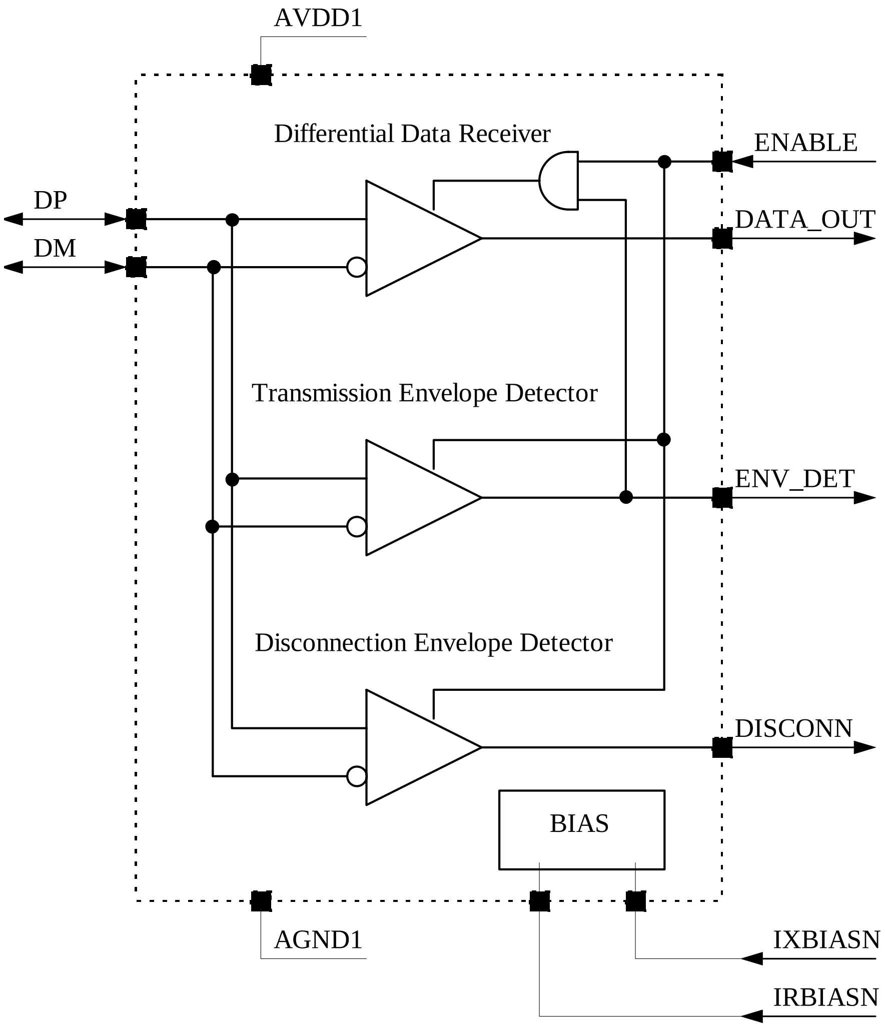

AVDD1
Differential Data Receiver
ENABLE
DP
DM
DATA_OUT
Transmission Envelope Detector
ENV_DET
Disconnection Envelope Detector
DISCONN
BIAS
AGND1
IXBIASN
IRBIASN

**Fig. 3: Block diagram of high-speed receiver**

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.”

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**

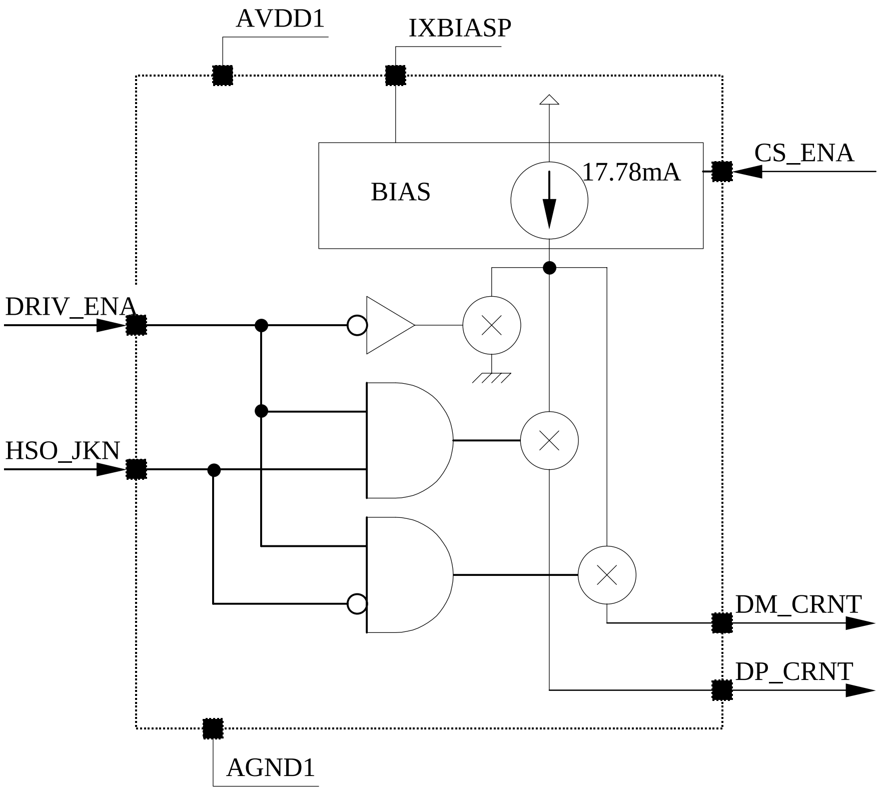


**Fig. 4: Block diagram of high-speed current driver**

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 HS DLL 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.

**PLL**

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

## BLOCK DESCRIPTIONS

### RECEIVER:

The Receiver includes the HS and FS Receive Portions of the Analog Front End and the Receive Portion of the Digital Block which includes HS Clock & Data Recovery, Elasticity Buffer, FS DLL, Receive FIFO and Receive State Machine Implementation which includes NRZI Decoding, Bit Un-Stuffing, UTMI Control Signal Generation.

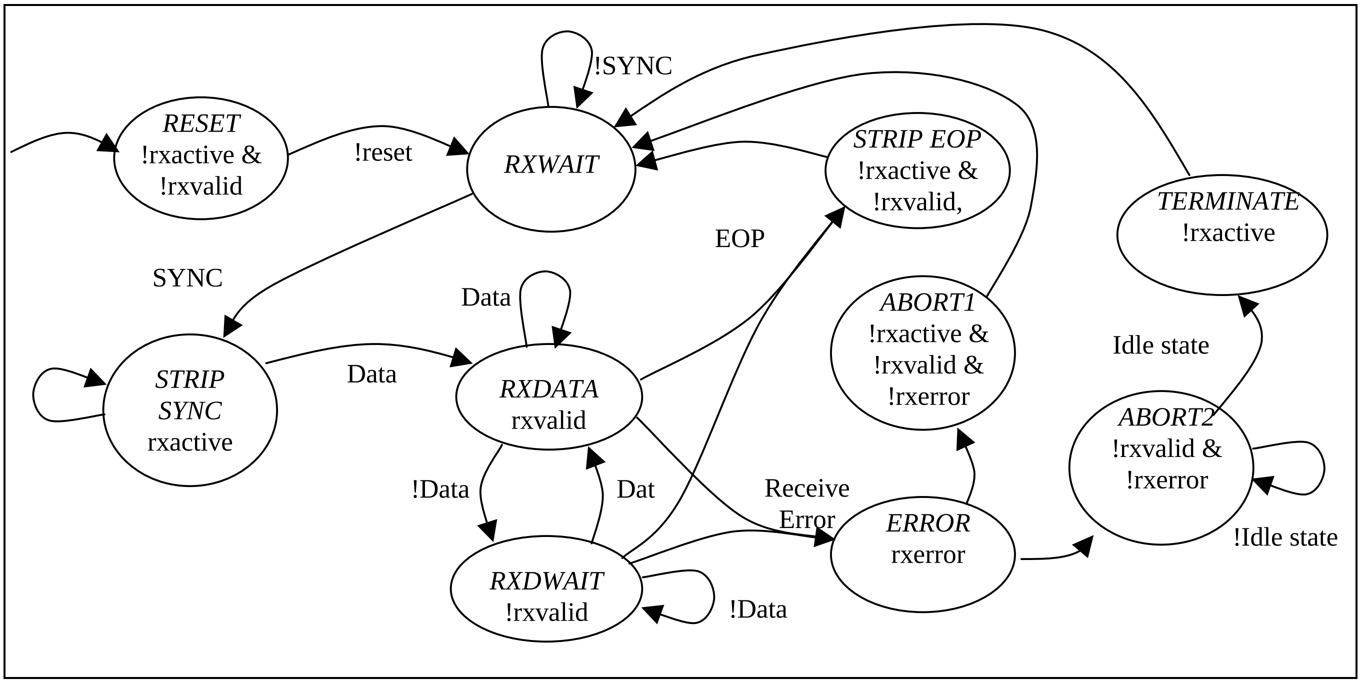
The receive path functions as follows:

1) AFE continuously monitors the line state for activity.
2) The digital block detects SYNC and strips it, at the same time turns on RxActive.
3) Received and Recovered Data is de-serialized, after being converted to NRZ and bit stuffing additions have been removed to generate individual bytes
4) Received bytes are put into Receive FIFO
5) Bytes are extracted from FIFO and given to UTMI with the control of RxValid and RxValidh signals

### 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 SYNC is detected, STRIP SYNC filters out SYNC pattern until the first valid data.
**RXWAIT:** When reset signal is gone, the Receive SM 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 5: Receive State Diagram**



## TRANSMITTER

The transmitter receives data from the UTMI under the control of TxReady, TxValid, TxValidh signals and drives the AFE to generate the correct form of signaling on the USB wire.

The Transmitter functions as follows:

1) The Device Function puts the transmit data on DataIn bus and turns on TxValid and/or TxValidh as appropriate.
2) The PHY samples TxValid and generates TxReady after putting the initial data into the Transmit FIFO
3) Individual bytes are taken from the TX FIFO and serialized for transmission while being NRZI encoded and Bit-Stuffed at the same time.
4) If, because of Bit-Stuffing, the TX FIFO becomes full, TxReady gets turned off which signals the Device Function to stop sending more data
5) The PHY drives turns on the appropriate Front End to generate the necessary Electrical signaling on the USB wire
6) As appropriate the PHY generates the SOF EOP and does the Disconnect Detection at the end of the packet

## 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**.
**SENDSYNC:** 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 Data Holding 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.

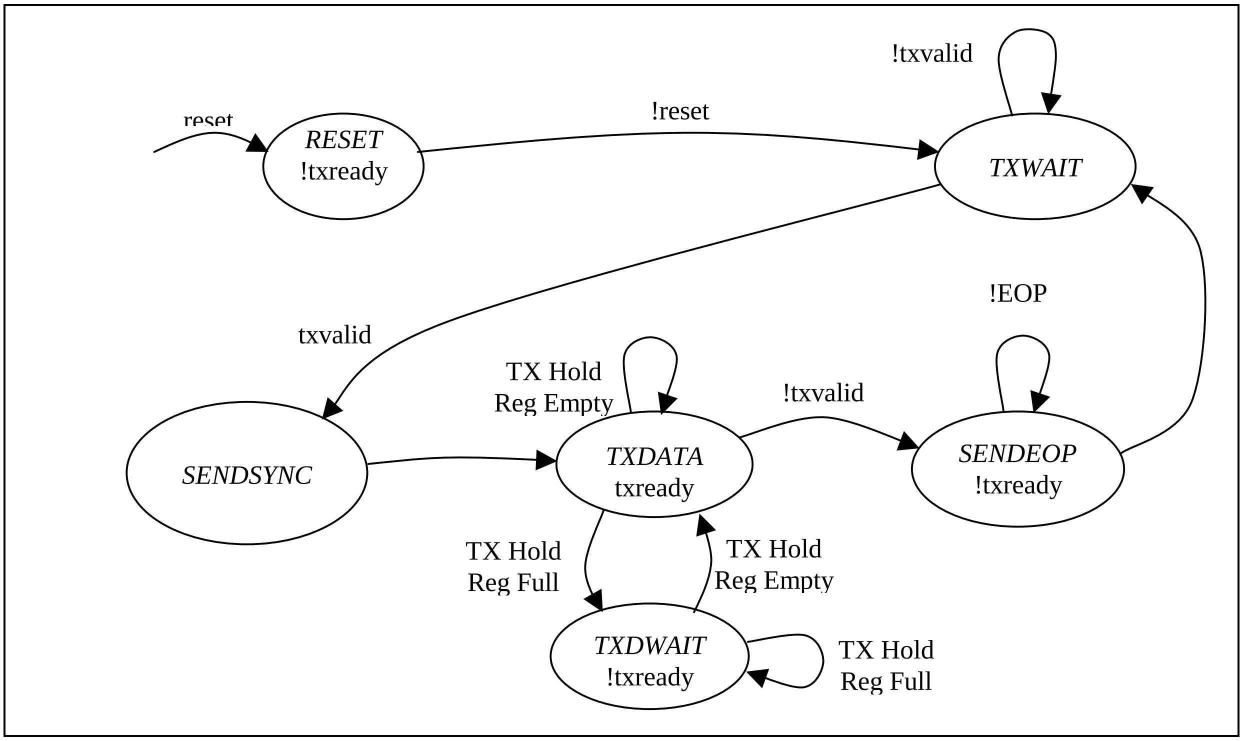
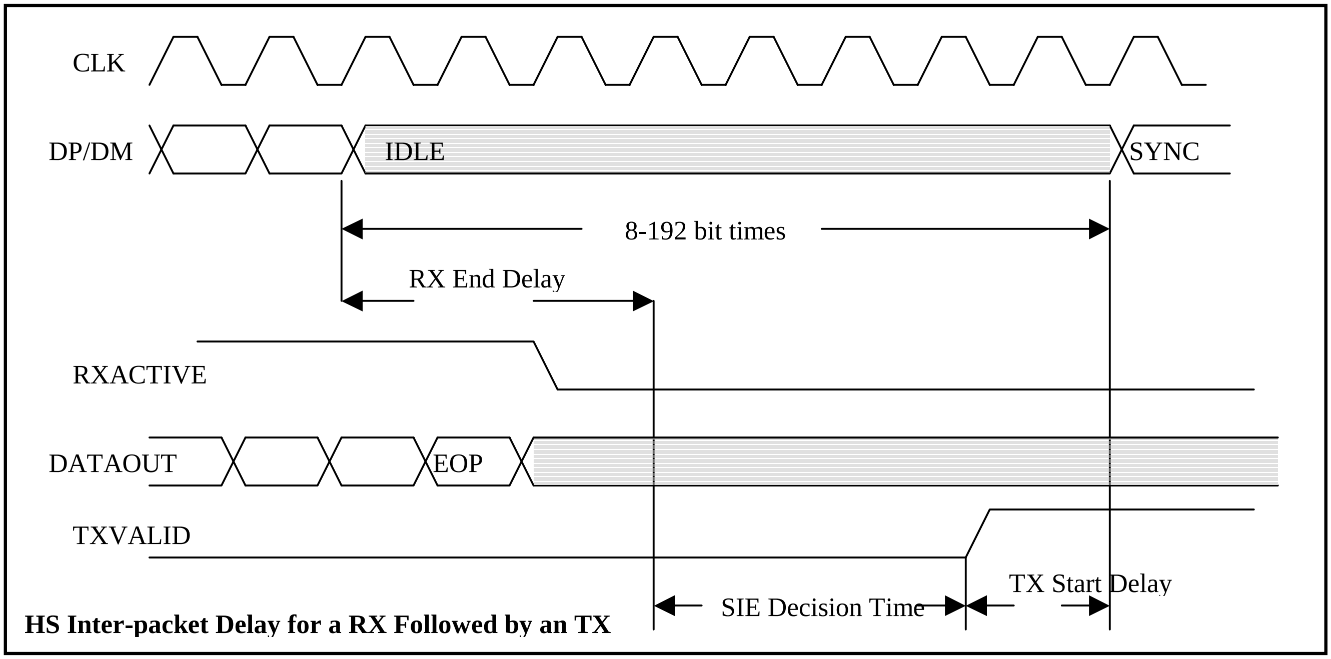


**Figure 6: Transmit State Diagram**

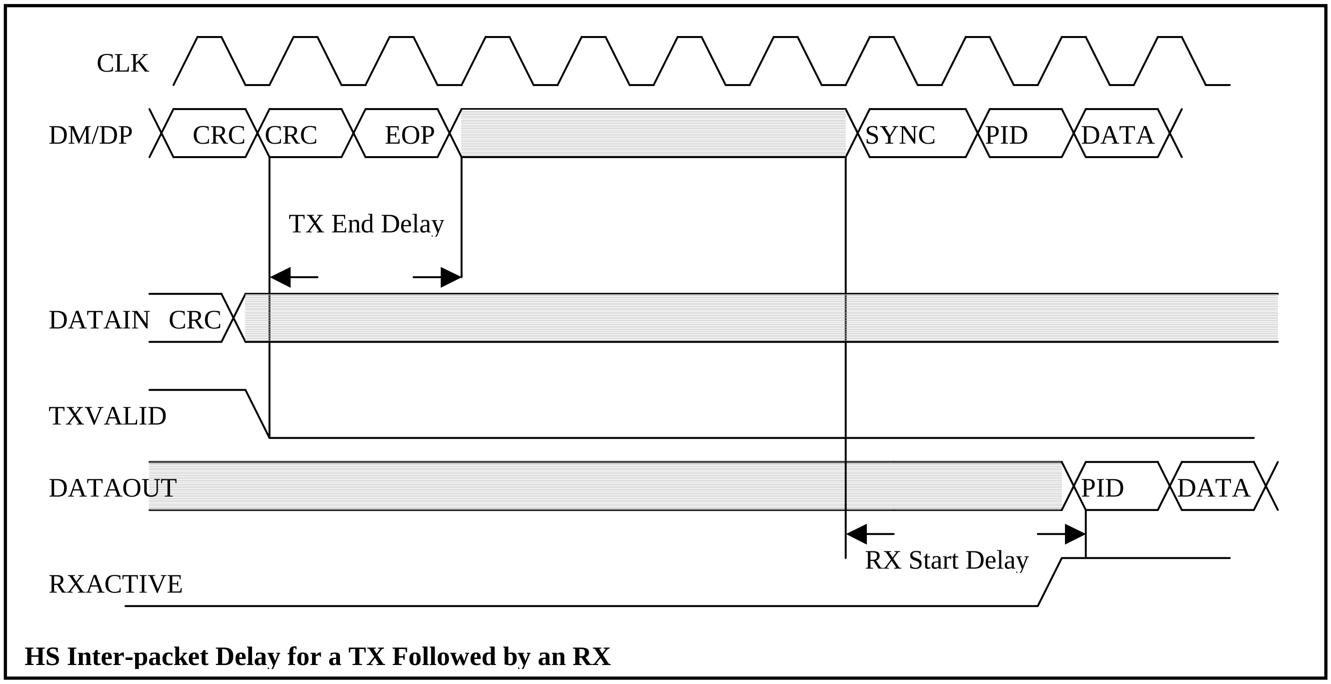
## TIMING CONSTRAINTS:

INTER-PACKET DELAY TIMING VALUES:



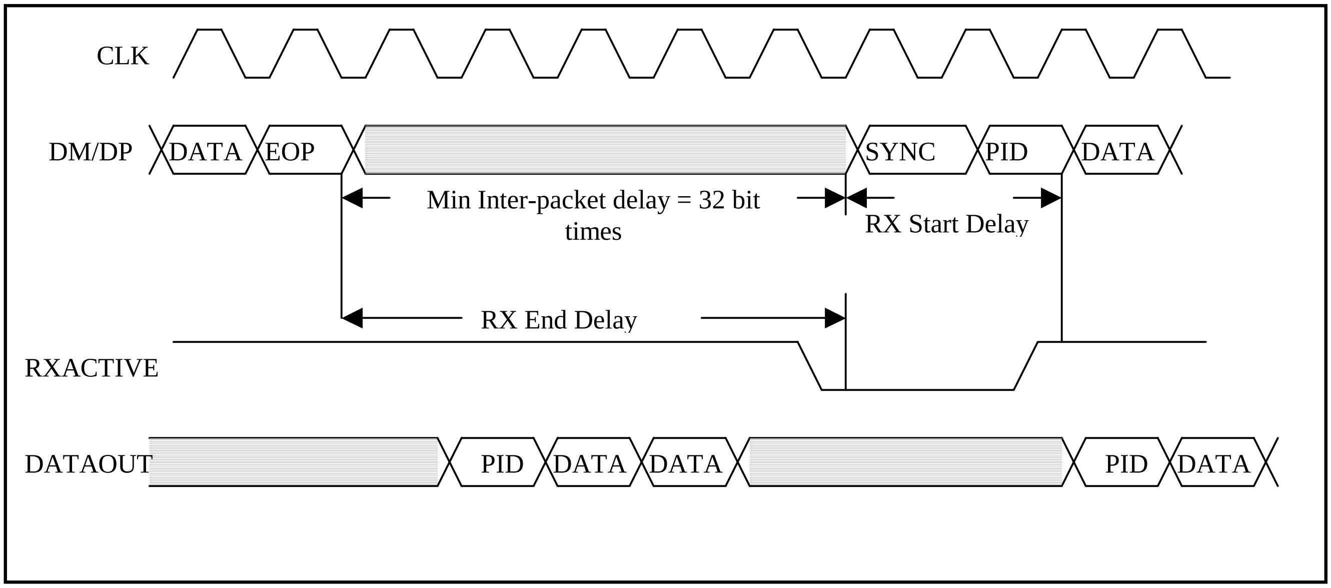
HS Inter-packet Delay for a RX Followed by an TX

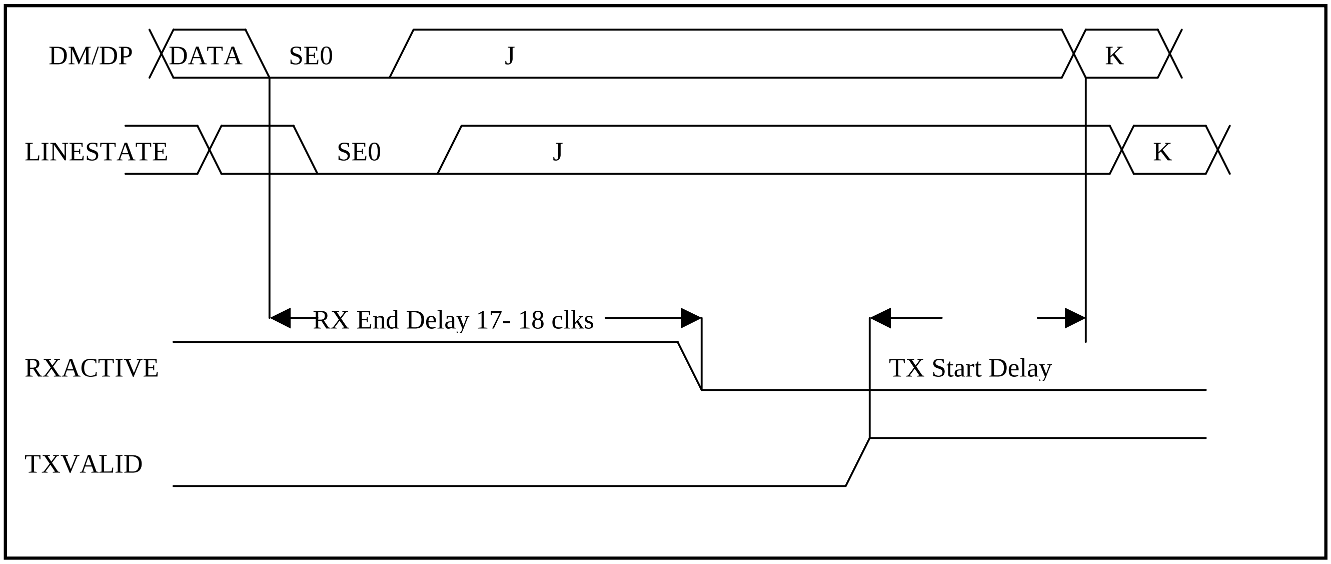
SMS2100 RX End Delay is 44 bit times to 63 bit times for 8-bit UTMI and 52 to 71 bits in 16-bit UTMI mode.
The Maximum TX Start Delay for SMS2100 is 16 bit times.



HS Inter-packet Delay for a TX Followed by an RX

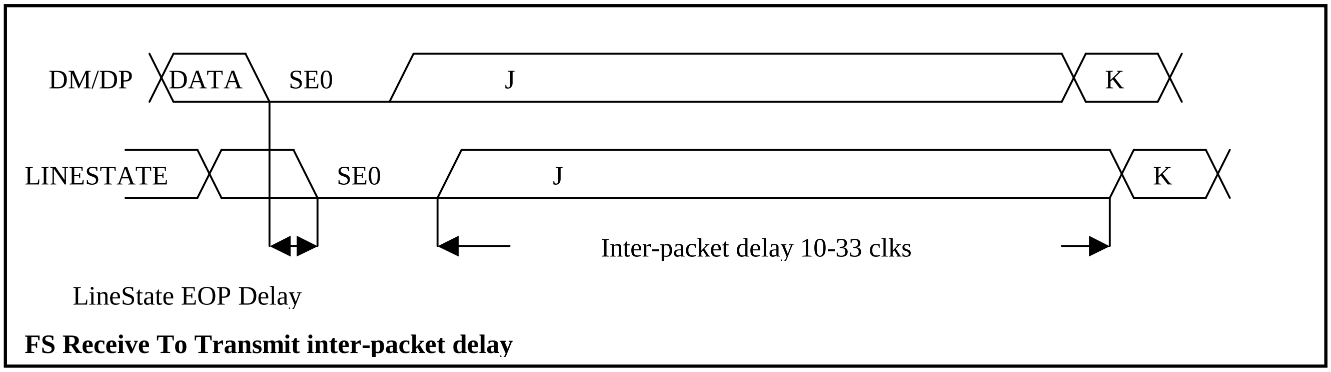
SMS2100 TX End Delay is 18 bit times to 40 bit times.
SMS2100 RX Start Delay is 40 bit times to 63 bit times.





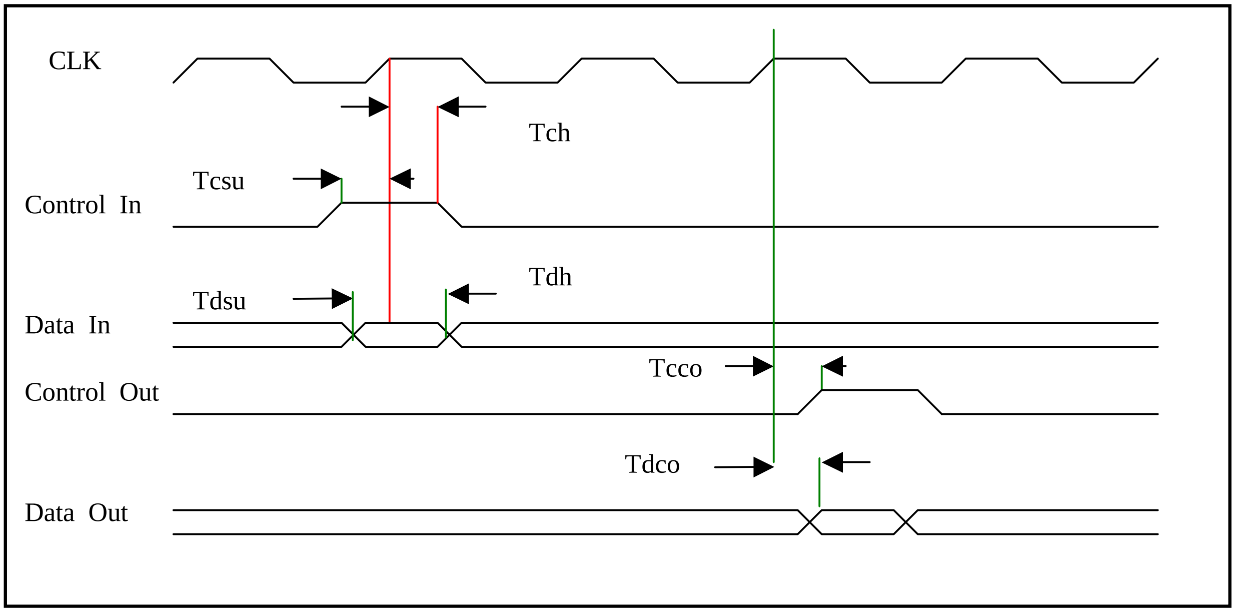
The SIE must utilize the LineState to determine the EOP transition from SE0 to the J-State.

FS SIE Decision time must be between 7 to 9 clks to ensure that 6.5 bit times FS inter-packet gap is met.



**FS Receive To Transmit inter-packet delay**

UTMI CONTROL SIGNAL TIMING VALUES:



**Tcsu: 3ns**
**Tch: 1ns**
**Tdsu: 3ns**
**Tdh: 1ns**
**Tcco: 2ns min, 4ns max**
**Tdco: 2ns min, 4ns max**

## ELECTRICAL CHARACTERISTICS:

PLEASE REFER TO ATTACHED SEPARATE ELECTRICAL SPECIFICATION DOCUMENT FOR ELECTRICAL CHARACTERISTICS OF SMS2100 DEVICE PHY.