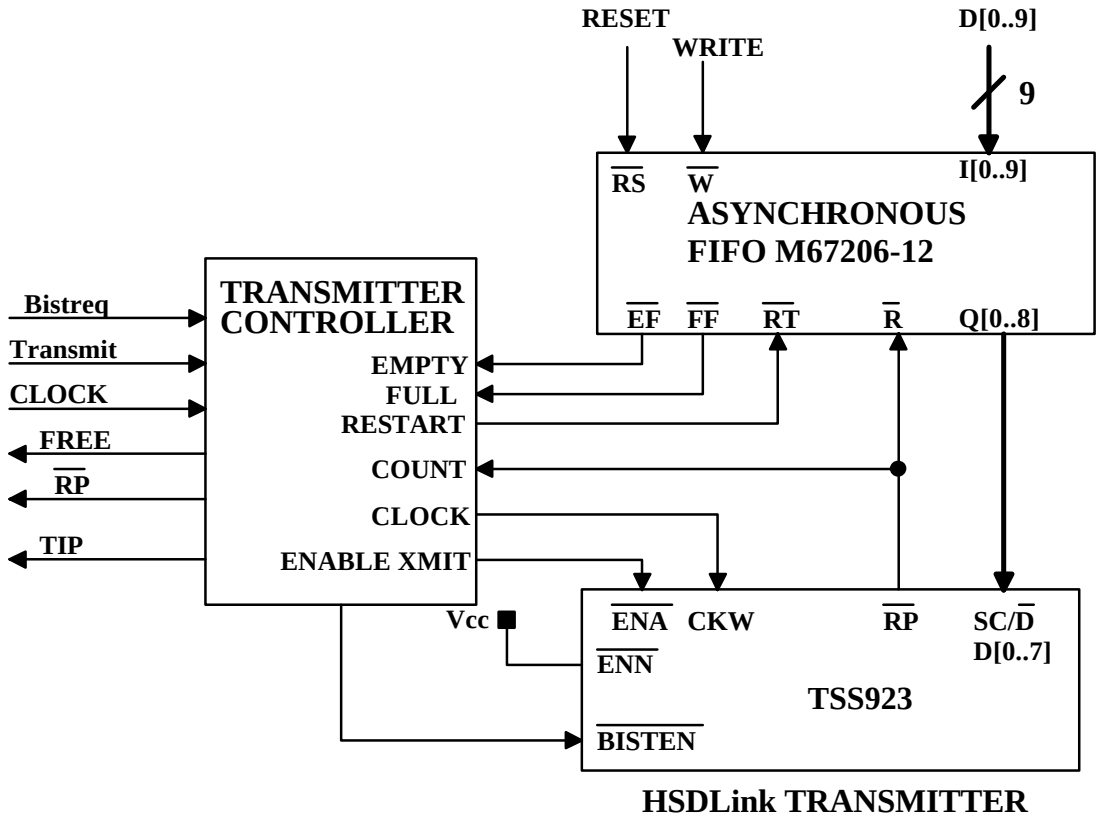
TSS923 & TSS933 HSDLink with FIFO

Using Asynchronous FIFO

This note describes the design considerations of a high-speed serial transmitter (TSS923) / receiver (TSS933) with FIFO (First In First Out) data buffers.

Transmitter interface

The transmitter interface consists of a new single 16Kx9 M67206-12 interfacing directly to the HSDLink transmitter. A transmitter controller supplies the control signals to both FIFO and the HSDLink transmitter. The architecture of the controller can be implemented in a PLD or FPGA.



.

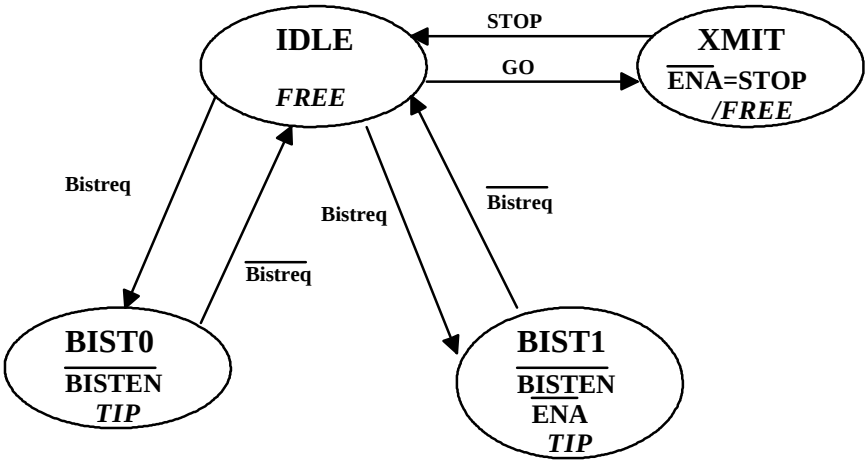
The transmitter controller will be in charge of supervising the data flow by managing the $\overline{FF}$ (Full Flag) and the $\overline{EF}$ (Empty Flag) indication. The FIFO write port interfaces directly to a 9-bit data bus. Data are written into the FIFO by asserting the $\overline{W}$ signal. Data may be written at any time as long as the FIFO is not full (as indicated by the FIFO Full Flag) and a FIFO reset cycle is not in progress. The FIFO read port interfaces to the HSDLink transmitter parallel port. Control of this interface is done by the transmitter controller state machine. The interface state machine is under the control of a higher level controller responsible for the serial protocol and data bus/FIFO transactions.

The interface controller can be considered as a simple state machine. While the state machine waits in the IDLE state, HSDLink will transmit SYNC characters ($\overline{ENN}$ ='1', $\overline{ENA}$='1'). When the Transmit signal is asserted by the higher-level controller, the transmitter state machine goes to the XMIT state. This state asserts FREE signal and allows reads of the 9-bit words out of the FIFO into the HSDLink transmitter until a STOP condition is detected

(the FIFO is empty or the transmit signal is deasserted). Reading the data from the FIFO is accomplished by asserting

$\overline{ENA}$ low. Assertion of $\overline{ENA}$ causes data to be latched into the HSDLink transmitter. After a STOP condition is detected, the state machine returns to the IDLE state and deasserts the FREE signal.

If the transmitter is in IDLE state, the higher level can assert BISTREQ signal to place it into a BIST0 state. This BIST0 state will assert $\overline{BISTEN}$ ($\overline{ENA}$ is let to high level) to initiate transmission of the repeating character 1010101010 and will also assert TIP (Test In Progress) signal to '1. If BISTREQ is deasserted then state machine returns to IDLE mode and TIP signal is deasserted. The next request on BISTREQ when transmitter is into IDLE mode will place the state machine into BIST1 state. The BIST1 state asserts $\overline{BISTEN}$ and $\overline{ENA}$ to start BIST pattern generation. The higher level could monitor the number of BIST loops by counting the number of pulses on $\overline{RP}$. After the desired number of BIST loops made, the higher level could deassert BISTREQ and then the state machine returns back to IDLE state.



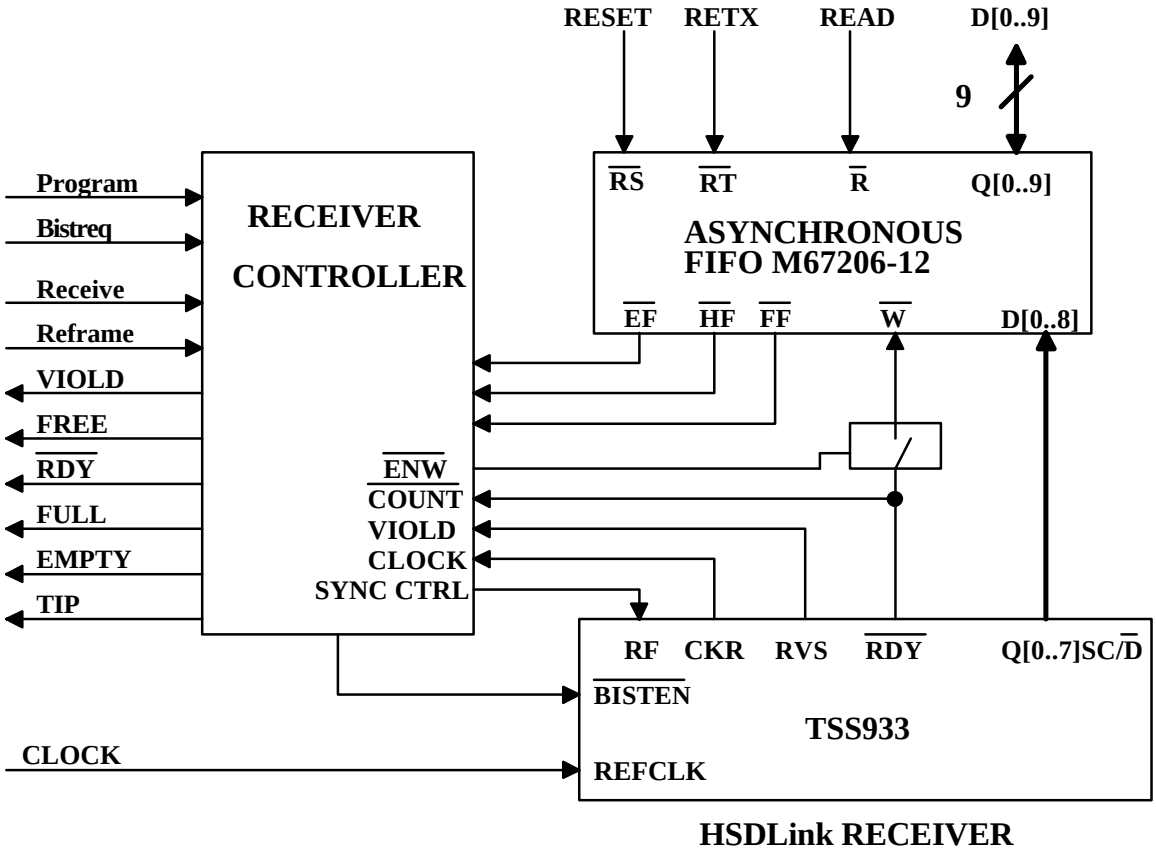
Empty = $\overline{EF}$
GO = Transmit.$\overline{Bistreq}$
STOP=$\overline{Transmit}$+Empty

Critical timings

FIFO access time (ta=12ns) plus data setup time for HSDLink (tsd = 5ns) has to be less than 60% of the CKW cycle.

Receiver Interface

The receiver interface uses also a new single M67206-12 to buffer the parallel data presented by the HSDLink receiver. The HSDLink receiver interface is capable of up to 320Mbits/s and then writing 9-bit words in the FIFO. Words in the FIFO can be read up to 70 MBytes/s. A higher-level controller is responsible for coordinating the receiver interface and bus transactions according to the serial link protocol .

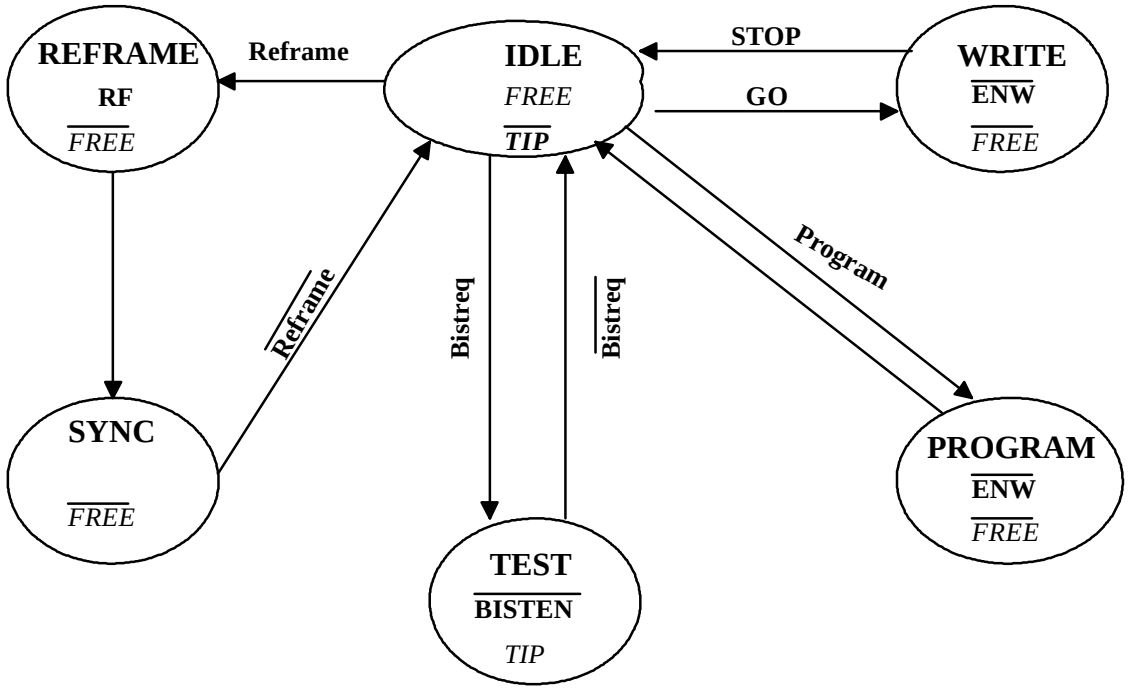


HSDLink RECEIVER

The HSDLink serial receiver must synchronize itself with the proper word alignment of the incoming data. Assertion of the HSDLink RF (ReFrame) input forces HSDLink to synchronize its internal bit counter with the boundary of a received K28.5 (SYNC) character. HSDLink will answer by asserting $\overline{RDY}$ low when the first K28.5 is received in the incoming flow. The receiver state machine controller should be designed to synchronize HSDLink at the beginning of data reception.

The receiver state machine answers to control signals from a higher-level controller. The higher-level controller initializes data reception by asserting the Receive signal to the receiver state machine. 9-bit words from the HSDLink parallel port are stored into the 67206 FIFO each time $\overline{RDY}$ is asserted low. $\overline{RDY}$ will pulse low when new data are available at the HSDLink parallel port and will be high when multiple SYNC codes are received. $\overline{RDY}$ will avoid to fill the FIFO with SYNC codes. If $\overline{FF}$ is set, the FIFO will ignore writes. The 67206 has one programmable flag ($\overline{HF}$). which can generate Almost Full or Almost Empty flags. The distance that this flag becomes active from the empty and full FIFO is programmed during RESET cycle using $\overline{RS}$ (ReSet) pin. The distance can be chosen such that $\overline{HF}$ is asserted when a fixed packet length of data has been received. The higher-level control will be in charge of reading the data packet when the previous flag ($\overline{HF}$) will be set. This flag can be set to compensate the response latency or regulate the data flow.

The receiver state machine can be described as follow



Empty = HF+EF
Full = HF+FF
STOP = Receive
GO = Receive.Reframe.Bistreq
STOP = ENW

The Receive signal allows the state machine to store word into the FIFO when /RDY pulses low. If Receive is deasserted, the reception will be stopped immediately. The Reframe signal asks the state machine to synchronize the HSDLink receiver to the serial data in. The BISTREQ signal forces the HSDLink receiver to enter BIST mode and the Program signal allows the state machine to write a word into the FIFO internal program register. The FREE signal output is asserted when the state machine is in IDLE state.

The REFRAME state takes place by the assertion of Reframe from the IDLE state. The REFRAME state is used to synchronize the receiver to the incoming serial data stream. When the state machine asserts RF, the HSDLink synchronizes its bit counter with received SYNC characters. $\overline{RDY}$ pulse low when the first SYNC character is available. The state machine will return to IDLE state when $\overline{Reframe}$ is deasserted. Data reception is initiated by asserting the Receive signal when the state machine is in IDLE state. The controller will jump to the WRITE state and store data with $\overline{RDY}$ low pulse until Receive is deasserted. Then state machine returns back to IDLE state. It is during reception that the higher-level controller has to manage the data flow using programmable flags. The BIST state is used when BISTREQ is asserted. During BIST writing to the FIFO is disabled by using an analog switch controlled by $\overline{ENW}$ signal. The W signal will be held high using a pull up when it is disconnected from $\overline{RDY}$. This $\overline{RDY}$ signal will pulse once per BIST loop and should be used to count the number of BIST loops received.

The Program state write the program word into the FIFO internal program register. This state is entered from the WAIT state at the command of the higher-level controller. The program word will be entered into the FIFO during the Reset cycle ($\overline{RS}$)

.

Critical timings:

The state machine delay for generating output control signals from valid inputs is t_{PD}. The FIFO enable set-up time is t_{SEN}= 5ns. If t_{CKR} is 30ns, then $t_{PD} <= 1/2\ t_{CKR} - t_{SEN} - 3ns = 7ns$

A pipeline register inserted between FIFO and HSDLink receiver will increase this t_{PD} allowing to use fast but not very fast PLD or FPA to built the receiver controller.