



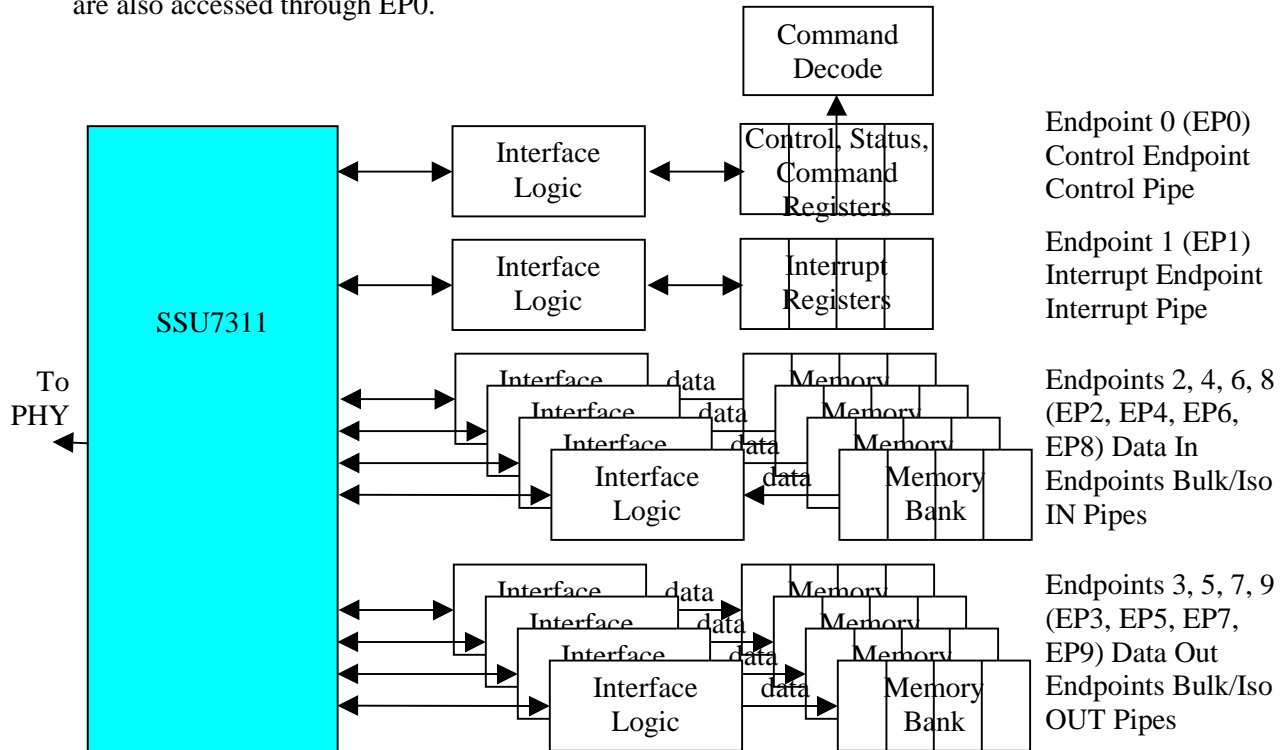
## SSU7311: USB 1.1 Device Core

The SSU7311 USB 1.1 Device Core provides USB 1.1 Device functionality. It includes the required endpoints plus eight bulk/iso transfer endpoints. This USB 1.1 Device Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact [CustomerService@auroravlsi.com](mailto:CustomerService@auroravlsi.com).

Features are summarized:

- Ten endpoints:
  - EP0- control endpoint, accepts SETUP, IN, and OUT control transactions
  - EP1- interrupt endpoint, accepts IN and OUT interrupt transactions
  - EP2, EP4, EP6, EP8- IN endpoints; accept IN bulk and isochronous transactions
  - EP3, EP5, EP7, EP9- OUT endpoints; accept OUT bulk and isochronous transactions
- Serial interface to PHY transceiver chip
- Accepts stalls from the application logic

The application uses the eight bulk/iso pipes for data transfer. Endpoints 2, 4, 6, and 8 (EP2, EP4, EP6, and EP8) send data over bulk/iso IN pipes. Endpoints 3, 5, 7, and 9 (EP3, EP5, EP7, and EP9) receive data from bulk/iso OUT pipes. Data typically resides in SRAM memory banks at the bulk/iso endpoints. The interrupt pipe is used for event notification- interrupts for events such as media change, media no longer ready, etc. The application posts interrupts in the Interrupt Registers, and the USB Host reads the interrupt registers with IN transactions to EP1. All the USB, class specific, and vendor specific commands are decoded and executed as register transfers through Control Endpoint- EP0. Configuration, interface, endpoint status registers, etc. are also accessed through EP0.



SSU7311- USB 1.1 Device Core



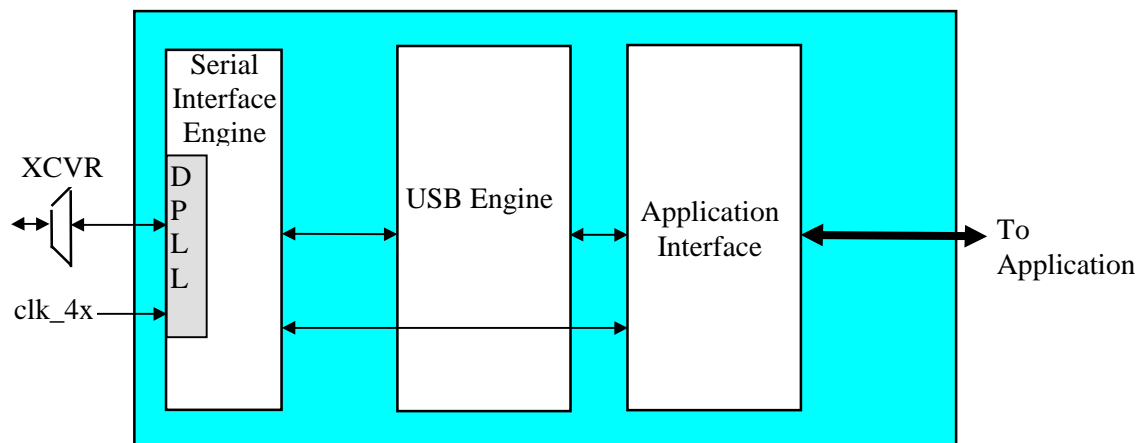
A block diagram of the SSU7311 is shown below.

There are three major blocks in SSU7311, the Serial Interface Engine, USB Engine, and the Application Interface.

The Serial Interface Engine interfaces with a standard USB PHY transceiver on the line side. A DPLL is used to extract the clock from the received data stream. The Serial Interface Engine converts the received serial data stream into parallel bytes and delivers them to the USB Engine. In transmit mode, it converts the parallel bytes from the USB Engine into a serial data stream, and sends this serial data to the PHY transceiver. As part of the USB protocol, it performs SYNC detection, bit stuffing/unstuffing, PID decoding, NRZI encoding/decoding, and CRC checking/generation for token and data packets. It also monitors the line to detect reset, end of packet (EOP), start of packet (SOP), and idle conditions. In transmit mode, it generates SOP, EOP, and resume signaling. The Serial Interface Engine keeps track of the byte boundaries.

The USB Engine keeps track of a transaction on TXVALID, from SOP to EOP. On SOP, it checks the validity of the address and endpoint, and initiates the appropriate data transaction based on the status of the endpoint FIFOs. It handles the data retry mechanism using data toggling and generates appropriate handshakes.

The Application Interface provides a simple mechanism to interface to the user's logic. All endpoint FIFOs, registers, and any other memory elements are in the application logic. This interface allows direct access to all these endpoint FIFOs, registers, and other memory elements. It controls the addressing and control signals for the SSU7311 side of these memory elements. Setting and clearing of stall conditions are controlled by the application logic through the Application Interface. Each endpoint is controlled independently. This allows simultaneous access to any number of endpoints.



The core is delivered as a synthesizable RTL Verilog model. Deliverables include:

- RTL Verilog source code model of the core
- Verilog testbench and test cases
- Synthesis scripts examples
- Complete detailed documentation and training class notes