Interfacing an External Processor to the SL811HS/S

Introduction

The SL811HS is a dual-role capable, i.e., host or peripheral, embedded USB controller. As such, it is designed to be easily connected to a variety of external embedded processors ranging from an 8051 to a StrongARM as a memory mapped peripheral. The signal descriptions and transaction methods described here equally apply to the SL811S peripheral only device. This application note describes the typical methods used to connect the SL811HS/S to an embedded processor. Example circuits and signal descriptions are provided that should help you become more confident that your design will work the first time around.

This application note also describes a typical configuration of support circuitry needed when a USB controller is incorporated into a USB-enabled embedded system. Two configurations are demonstrated including host-only and peripheral-only.

Signal Basics

The SL811HS/S incorporates an industry-standard address/data bus. The requirements of the embedded processor signals are laid out in the following list.

- Active LOW CHIP SELECT signal
- Active LOW READ signal
- Active LOW WRITE signal
- Active HIGH INTERRUPT signal
- Address bus or GPIO
- Data bus, at least 8-bits wide
- GPIO to drive various signals such as RESET, USB bus power enable, various resistors…etc. The number of GPIO required is dependent on the controller's configuration. See the schematics later in this document for more information.

CHIP SELECT (nCS) – nCS is used to enable the SL811HS/S interface and read or write the SL811HS/S registers/memory. nCS essentially signals that the transaction is intended for "this chip" as opposed to another one that might share the same read or write signals. nCS must be asserted by the embedded processor for at least 65 ns during a transaction in order for the transaction to be valid. With some embedded processors this may require that the firmware set an additional number of wait-states so that nCS does not cycle too fast. Wait-state generation is processor dependent, so no information will be given here on how to set additional wait states. If the SL811HS/S is the only IC on the embedded processor's data bus, nCS can be continuously asserted.

READ (nRD) – nRD is an active LOW signal driven by the embedded processor that is used to signal a register or memory read. Before a read can take place, the desired address to read must be written into the SL811HS/S. During a read nCS must also be asserted in order for the SL811HS/S to recognize the assertion of nRD. The minimum pulse width of the nRD pulse is 65 ns. 65 ns after the assertion of nRD the D[7:0] signals switch from hi-z to driving mode and drive the data bus until 5 ns after nRD is deasserted. The minimum spacing in between nRD assertions is 85 ns.

WRITE (nWR) – nWR is an active LOW signal driven by the embedded processor that is used to write an address, register, or memory location. In conjunction with nWR, nCS must also be asserted in order for the SL811HS/S to recognize the assertion of the nWR signal. nWR is asserted LOW for a minimum of 65 ns. Data is written from the embedded processor to the SL811HS/S on the rising edge of nWR. The data must remain valid on the bus for 5 ns after nWR is deasserted in order for it to be properly latched by the SL811HS/S. The minimum spacing between nWR assertions is 85 ns.

ADDRESS (A0) – The A0 signal is driven by the embedded processor and is used in conjunction with the nWR signal to define a write as an address pointer or data. If A0 is LOW during a write, the write goes to an address pointer register. If A0 is HIGH, the write goes to a register or memory location pointed to by the address pointer register. For instance, if we want to write to the register at address 00h we would first perform a write with D[7:0] set to 00h and A0 set LOW. Then we would perform another write with D[7:0] set to the register value and A0 set HIGH. The value of A0 must be held for 10 ns after the assertion of nWR in order for the write to be properly recognized. Typically this signal would be connected to address bit 0 on an 8-bit processor, address bit 1 on a 16-bit processor, or address bit 2 on a 32-bit processor.

In some cases the embedded processor may be using an older Intel®-type bus with multiplexed address and data pins, and will typically have an address latch enable (ALE) signal. If this is the case, an external flip-flop will be required to latch the value of the A0 pin on the ALE edge (edge may depend on the particular processor) as shown in *Figure 1*.

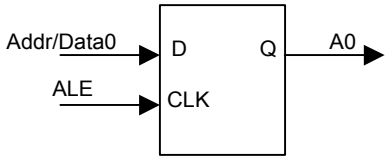

Figure 1. Using ALE on a Multiplexed Bus

INTERRUPT (INTRQ) – The interrupt signal is asserted HIGH by the SL811HS/S during a programmable interrupt event. Some examples may include the completion of a transaction or the connection of a new peripheral device. INTRQ is asserted HIGH until the interrupt event is cleared by writing to the associated interrupt clearing register in the SL811HS/S. The interrupt polarity is not programmable, so an

external inverter may be required if a particular processor does not support active HIGH interrupt signaling.

DATA BUS (D[7:0]) – The SL811HS/S bidirectional data bus is used to transfer data in and out of registers or memory. The data bus is normally held in a high-impedance state unless both nCS and nRD are asserted during a read transaction. The D[7:0] pins should be connected to the least significant byte of the embedded processor's data bus.

RESET (NRST) – Reset must be asserted LOW at power-on by the embedded processor or an external POR circuit. NRST is asserted for 16 clock cycles of the CLK signal. Further transactions with the SL811HS/S should not take place before 16 cycles of CLK after NRST is deasserted.

ROLE (M/S) – This signal determines the operating role for the SL811HS at the assertion of an external reset. At the assertion of NRST the value of the M/S pin is latched into the internal M/S register bit. If M/S is held LOW, the SL811HS acts as a USB host. If M/S is held HIGH, the SL811HS is a USB peripheral. During normal operation the M/S bit does not have any effect on the operation of the SL811HS. The operating role of the SL811HS may be changed without external reset by software running on the embedded processor that changes the SL811HS internal M/S register bit.

DMA SIGNALS (nDACK, nDRQ) – These peripheral-only DMA related signals are typically not used with an embedded processor so they are not described here. Please see the SL811HS/S data sheet for more details on nDACK and NDRQ.

The SL811HS/S I/Os are 5-volt tolerant, but will only drive its own I/Os to 3.3 volts. As long the embedded processor has TTL or 3.3-volt CMOS level inputs the SL811HS/S interface should not require voltage translation buffering.

Figure 2 shows a typical example of the connection of a SL811HS/S to generic embedded processor bus.

Example Transactions

Two example transactions are shown in *Figures 3* and *4*. *Figure 3* shows a simple write transaction where a register or memory location is being written to.

Figure 4 shows a simple read transaction of a register or memory location.

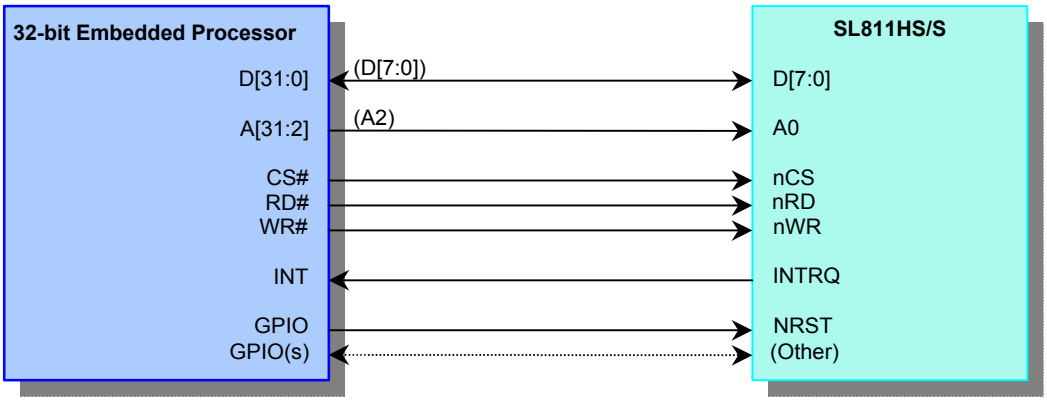

Figure 2. Example Connection of a 32-bit Embedded Processor to the SL811HS/S

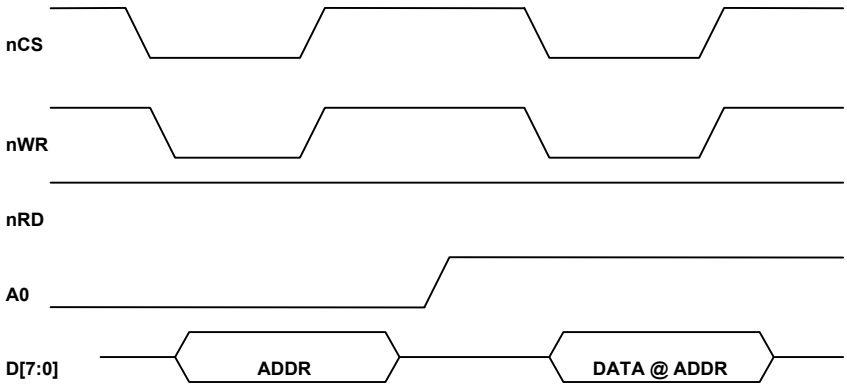

Figure 3. Example Write Transaction (not to scale)

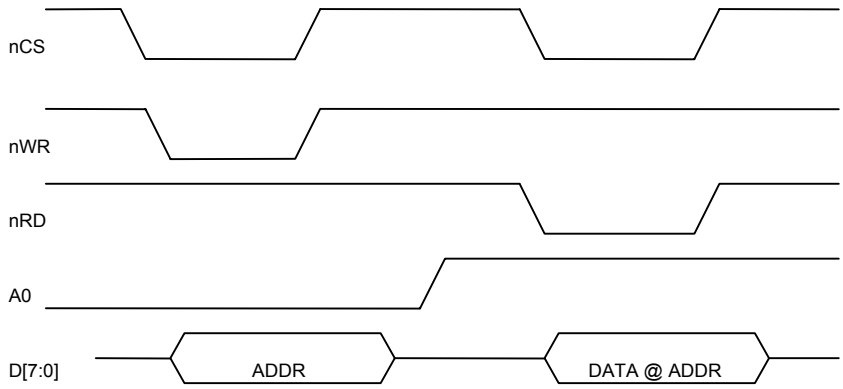

Figure 4. Example Read Transaction (not to scale)

Example SL811HS/S Circuits

Two typical circuit configurations are shown in *Figures 5* and *6*. The 48-pin version of the SL811HS is shown in each of the figures; however the same pin configurations apply to the 28-pin package. *Figure 5* represents a typical USB embedded host with power protection and all associated components. The clock may be supplied from a 3.3-volt 12-/48-MHz CMOS oscillator or a 12-/48-MHz crystal. The chosen clock source must meet the jitter and accuracy requirements of the USB 2.0 specification, meaning that driving the clock from an external processor timer/counter output may or may not be possible. Reset is generated via a GPIO on the embedded processor. Reset could also be generated from a dedicated POR circuit.

Figure 6 shows a typical USB peripheral configuration for the SL811HS or SL811S (minus the M/S pin). In this situation the USB data line pull-up resistor must be able to be disabled while power is disconnected; therefore two GPIO are required on the embedded processor. In *Figure 4* the pull-up resistor is connected to D+, meaning that the circuit is configured for full-speed USB operation. If low-speed USB operation is desired, the pull up resistor should be connected to the D– signal. Total capacitance on the USB Vbus must be below 10 µF to meet the USB 2.0 specification. A voltage regulator that converts the Vbus 5 volts to 3.3 volts is not shown in the schematic, but may be required if the device is bus powered.

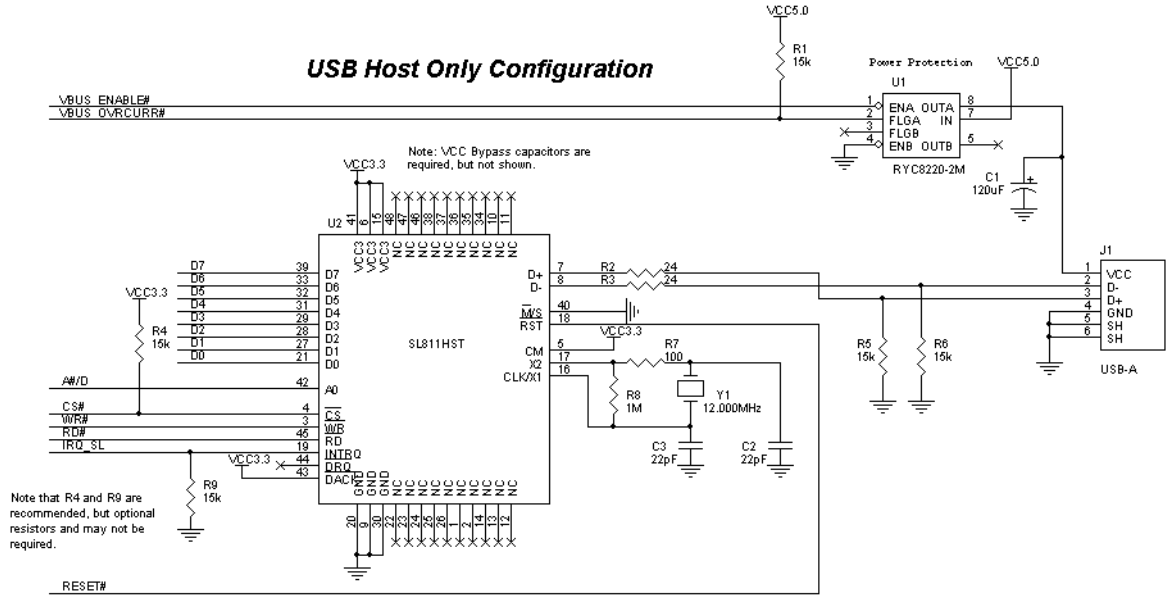

Figure 5. SL811HS in a Typical USB Host Configuration

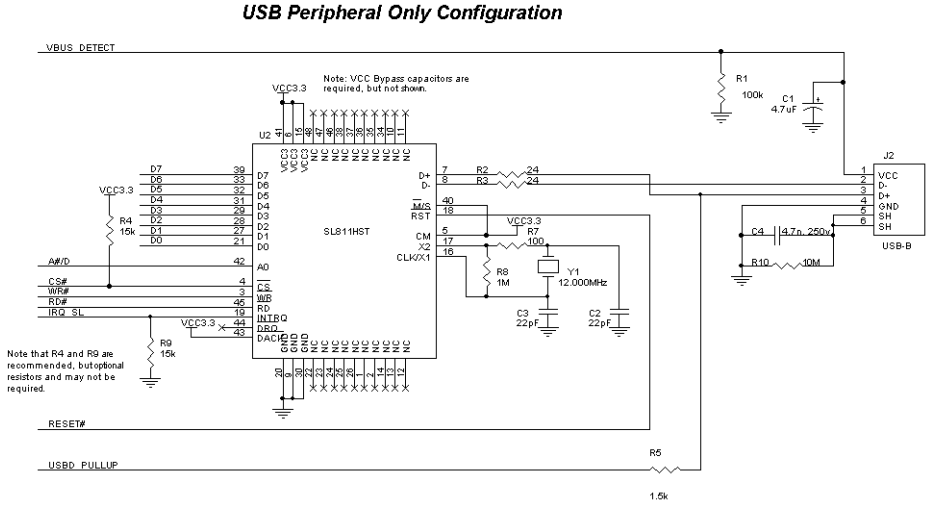

Figure 6. SL811HS/S in a Typical USB Peripheral Configuration

Conclusion

The SL811HS/S offers an easy-to-use interface for embedded processors. The use of standard data bus signals and 5-volt tolerant I/Os allow the SL811HS/S to connect to most embedded processors without glue-logic. For further questions and assistance please contact Cypress USB applications support.

approved dsg 12/3/02