Application Note December 20, 2007 AN1356.0

Table of Contents

Introduction

The shared 3-wire serial interface is a common interface developed for Intersil optical products, including LDD, PDIC and PMIC, or other receiving devices on the OPU. It is electrically compatible to the existing 3-wire serial interface. The purpose is to allow multiple devices to use the same 3 wires, thus minimizing the lines required between the main board and the OPU, while still allowing the high speed 3-wire interface. Devices from other vendors may use this interface specification to be compatible with this 3-wire serial bus.

All devices are always receptive to the device enable number. But, only one selected device at a time can have it's serial bus enabled for other read or write activity for control, parameter, or data. Any or all of the devices can be otherwise enabled for their other functions. It is only the access to the other serial space that is restricted. Thus one controller is able to control multiple devices of various types with only 3 serial wires. There is no provision at this time for allowing the controlling device to be changed from one device to another.

The three wires are SEN (serial enable, which is device high-impedance input pin and controller output pin), SCLK (serial clock, which is device high-impedance input pin, and controller output pin), and SDATA (serial data, which is device bi-direction pin and controller bi-directional pin. During a controller write cycle, the SDATA line is driven by the controller and received by the device. During a controller read cycle, the controller drives during the direction and address part of the transfer, and receives during the data portion of the transfer. The device drives the SDATA line during the data portion of a read cycle by the controller. The device circuitry may use hysteresis or other means to reduce the effects of noise on the interface.

Serial Programing

The serial programming word is 16 bits long comprising a direction bit, followed by a 7-bit address, followed by an 8-bit data byte. It is the same as in existing devices that are not compatible with this serial bus.

The Direction bit defines if a serial write or read will occur. When the direction bit is 0, a write cycle (controller to device) will occur. When the direction bit is 1, a read cycle (device to controller) will occur.

Write Cycle

During a controller write, all of the devices will read the bus and accept a write to the device select register (DSR). But only the selected device will allow the write to pass to any other register. The controller drives all three lines during the write cycle.

Read Cycle

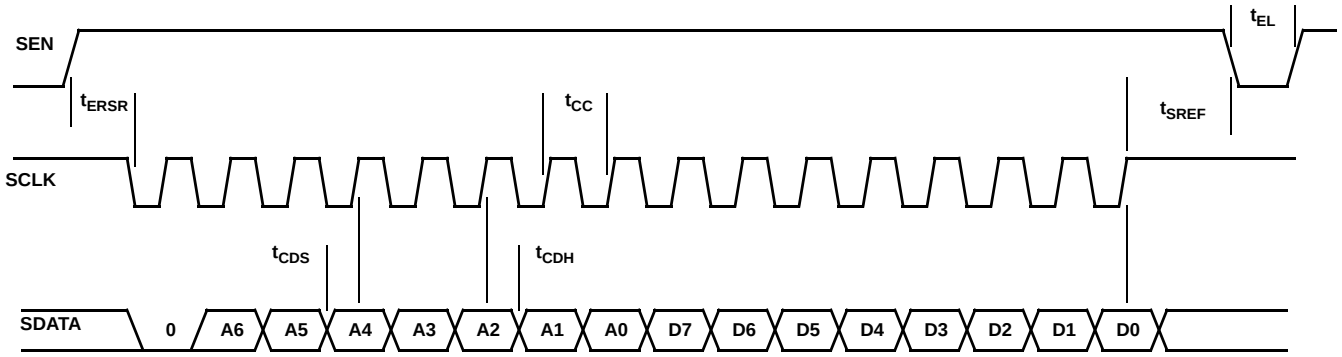
During a controller read, only the selected device will respond by sending the data pointed to by the address portion of the read cycle. Check the timing diagram closely to note the phase of the data, and the clocking of the last data bit.

Shared Timing Requirements

In the system, there are different serial interface speed requirements, and it is expected that different devices will be capable of different speeds. All devices must be capable of correct write and read performance from “DC” up to their rated speed. Read or write activity cannot be allowed on the bus at a speed faster than the slowest device on the bus. Thus if it is desired to operate one device at say 50Mhz, it cannot be done if another device rated at only 20Mhz is on the bus.

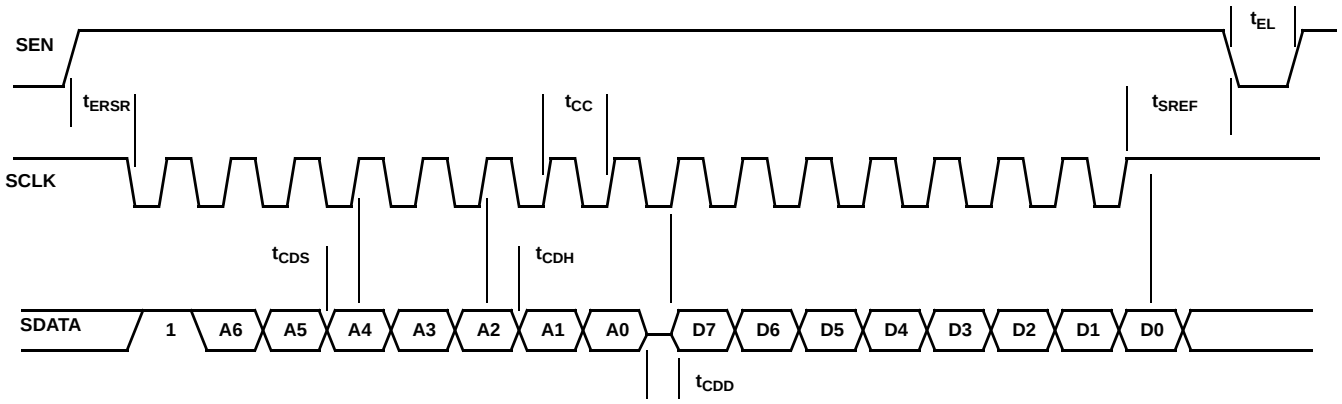
If it is desired to use a 50Mhz device at 50Mhz, and a 20Mhz device at 20Mhz, then a separate SEN line must be used; one to the 50Mhz device and another to the 20Mhz device. Both devices can share the SCLK and SDATA lines. It is required that all devices do not respond in any way to SCLK and/or SDATA activity when its SEN line is low.

Write Timing Diagram



- SCLK is high when SEN rises or falls.
- The controller changes SDATA at the same time that SCLK falls.
- The device clocks in SDATA on the rising edge of SCLK.

Read Timing Diagram



- SCLK is high when SEN rises or falls.
- The controller changes SDATA direction and address bits at the same time that SCLK falls.
- The device clocks in SDATA direction and address on the rising edge of SCLK.
- The controller releases the SDATA bus on the 9th falling SCLK edge.
- The device clocks out data on SDATA starting on the 9th rising edge of SCLK.
- The controller clocks in data on the SCLK falling edge.
- The controller clocks in D0 after the last SCLK rising edge, on a hidden SCLK falling edge.

Typical Serial Interface Specifications

TABLE 1. TYPICAL TIMING SPECIFICATION FOR SERIAL INTERFACE

PARAMETER	DESCRIPTION	CONDITIONS	MIN	TYP	MAX	UNIT
F_{SER}	SCLK operating range		0		$1/T_{CC}$	MHz
t_{EL}	SEN “L” time		$3*T_{CC}$			ns
t_{ERSR}	SEN rising edge to the first SCLK falling edge		$0.5*T_{CC}$			ns
t_{CDS}	SDIO set up time		$0.3*T_{CC}$			ns
t_{CDH}	SDIO hold time		$0.3*T_{CC}$			ns
t_{SREF}	Last SCLK rising edge to SEN falling edge		$0.5*T_{CC}$			ns
t_{CC}	SCLK cycle time		T_{CC}			ns
Duty	SCLK “H” duty cycle		40	50	60	%
t_{CDD}	SDIO output delay				$0.7*T_{CC}$	ns

Register Definition

TABLE 2. RECOMMENDED REGISTER LOCATIONS

ADDRESS	CONTENTS
00h	ID0 (read only)
01h	ID1 (read only)
02h	ID2 (Read only)
03h	Device Select Register. Write with delayed read-back
04h-0Fh	Reserved for future multi-chip protocol, checksum, etc.

Device Select Register (DSR)

The one register of the shared serial interface specification that is required to allow multiple devices to share the three interface lines is the Device Select Register (DSR). It is located at location 03h. Regardless of any paging schemes that might be used, the DSR will always accept a write to location 03h.

Device Enable Number (DEN)

Each device has an 8 bit Device Enable Number (DEN). The DEN need not be readable from a register location, except for test purposes. If the number in the DSR equals the DEN, then the serial bus of that device will be enabled for further read/write activity to and from the controller. If the number in the DSR does not equal the DEN, no register in the device can be read back by the controller, and only the DSR may be written to. Regardless of the contents of the DSR, the DSR for all devices will be written to when the controller writes to the DSR address (03h). All devices on a serial bus must have the same address for the DSR. When the device serial is not enabled by the DEN in the DSR, the SDATA must be in a high impedance state to allow the controller or other devices to drive the SDATA line.

From the viewpoint of the device, the DEN is the key to enabling it’s serial bus. The DEN should be a metal maskable option, or have some other hardware method of being locked into the device.

From the standpoint of the OPU designer, the DEN is a number that is assigned to a physical location on the OPU that uses the shared serial bus. In this way, it is not required to have a vast library of DEN numbers for various devices. The bare minimum for any application is that there is a different DEN number for each location on a particular shared serial bus.

Under this protocol of DEN assignment, the same device from the same vendor can be used multiple times on a shared serial bus, as long as each device has a different DEN number. If these DEN numbers are fixed in metal, then they are not really identical. But if the DEN is specified by voltage(s) on a pin(s), then they could be identical devices. In addition, the user of the shared serial bus is responsible for getting the device vendors to assign the DEN numbers of the user’s choosing. In this way, multiple vendors can use the shared serial bus. In spite of this, Intersil has proposed an initial assignment of DEN numbers for the OPU application. This proposed assignment is shown later in the document.

Device ID Registers

Locations 00h, 01h, and 02h are reserved to store a unique number for each device. The concept here is that the serial firmware can read the ID from a selected device, and from the ID be able to know what revision level and what device is present. The device ID allows the controller to make firmware decisions based on which device is located at the DEN. When the otherwise same device has it's DEN changed in metal, the device ID should also be changed so that the two devices can be differentiated. This provides a means to check that the correct device is located in the designated physical location on the shared serial bus. As for all other registers, the Device ID can only be accessed when the serial bus is enabled by having the DEN written to the DSR.

Device DEN and DSR Verification

It is prudent to verify that the device responds correctly to the DEN at device test. This involves making sure that the device disables the serial bus when the DEN is not in the DSR, and that it does operate when the DEN is in the DSR. Verifying that the serial bus operates correctly when the DEN is in the DSR can be done in the normal manner. First write the DEN to the DSR, then test the device serial bus. This type of verification cannot be done after the device is on a shared serial bus.

But when any other number is in the DSR, the serial bus should be inactive. How can it be verified that the serial bus is not active with some other number? One way would be to first write the DEN to the DSR, then write a test number to some other register, then write a non DEN to the DSR, then try to read the other register, then write the DEN to the DSR, then read the other register. Repeat this for all non DEN numbers. Use various other registers and test numbers to increase the probability of finding a bad case.

A partial helper to testing the DSR is to use a delayed DSR register. On every write operation to the DSR, write the previous contents of the DSR to a delayed DSR register. Then when the serial bus is enabled, the read of the delayed DSR register verifies that the DSR register recorded the number correctly. Although this is sufficient to verify the read and write operation of the DSR, it does not verify the totality of the serial and DSR function.

System Operation

- Upon power up, all devices on the serial bus should come up with a 00h in the DSR. No devices should use 00h as their DEN. Nor should any device use FFh as their DEN. Thus all devices power up with their serial busses disabled.
- After power up, the controller should verify that the proper devices are located in their proper locations. This can be done by enabling each DEN in sequence, and reading the Device ID from each device.
- Then the controller simply writes out the DEN of the desired device to address 03h. All devices read in the DEN, but only the device with the matching DEN will have it's serial bus enabled for further read / write activity. The controller only needs to write the DEN to the DSR once to enable all further read / write activity to that device.
- Next the controller writes out the DEN of the next device to be programmed. This enables the serial of the next device with the matching DEN, and disables the serial of the previous device. The previous device can still do it's other system functions, because all the DEN and DSR do is enable and disable the serial bus. All the registers of the next device can now be set, and the controller can then move on the third device.

DEN Number Assignment

It is expected that we cannot formulate a method that is perfect in which changes may not be required in the future. The specification can be revised. DEN numbers can be changed in metal. Even the 8 bit DSR can be expanded if necessary. Given these arguments, following is the proposed DEN number assignment for the OPU.

It is the responsibility of the pick-up manufacturer to request numbers for pick-up functions. Thus a given pick-up design will request number 03h for the LDD, 22h for the PDIC, 61h for the EEPROM, etc. In the case of a high volume product like an OPU, it is an inconvenience to have different OPU vendors use different DEN numbers, because it causes the device manufacturer to have different metal and different part numbers. But the volumes are usually high enough that it is not a significant issue.

In the case of someone using the serial bus for low volume applications, it is desirable for all device users to use the same DEN to increase the volume of manufactured parts with the same DEN. For the low volume application, it is suggested that the device manufacturer specify the DEN, rather than the device user.

The firmware will need to know the DEN numbers on its serial lines. It can do this automatically by cycling through the possible DEN numbers and getting a response from the various chips connected. Or, someone can burn the numbers into the EEPROM that the firmware uses.

Ultimately, it is the responsibility of the pick-up manufacturer to ensure that the same number is not used twice on the same pick-up. Looking far into the future, after the number space is full, numbers will be either re-assigned, or the number space expanded. It is intended that this number assignment and specification is a public, read only, document available on the Intersil web page. Requests for changes will be accepted and considered.

TABLE 3. PROPOSED DEN NUMBER ASSIGNMENT

DEN NUMBERS	DEVICE TYPE	COMMENTS
00h, FFh	none	Not used. Some change on the interface ensures that a DSR is present!
01h	LDD1	Single output LDD
02h	LDD2	Dual output LDD
03h	LDD3	Triple output LDD
04h	LDD4	Quad output LDD
05h	LDD1B	Second single output LDD
06h	LDD2B	Second dual output LDD
07h	LDD3B	Second triple output LDD
08h	LDD4B	Second quad output LDD
09h - 1FH	LDD space	Future expansion
20h	PDIC space	Left blank just to be compatible
21h	PDIC1	Single color PDIC
22h	PDIC2	Dual color PDIC
23h	PDIC3	Triple color PDIC
24h	PDIC4	Quad color PDIC
25h	PDIC1B	Second single color PDIC
26h	PDIC2B	Second dual color PDIC
27h	PDIC3B	Second triple color PDIC
28h	PDIC4B	Second quad color PDIC
29h – 3Fh	PDIC future expansion	
40h	PMIC space	Left blank just to be compatible
41h	PMIC1	Single color PMIC
42h	PMIC2	Dual color PMIC
43h	PMIC3	Triple color PMIC
44h	PMIC4	Quad color PMIC
45h	PMIC1B	Second single color PMIC
46h	PMIC2B	Second dual color PMIC
47h	PMIC3B	Second triple color PMIC
48h	PMIC4B	Second quad color PMIC
49h-5Fh	PMIC future expansion	
60h	EEPROM space	Left blank just to be compatible
61h	EEPROM1	
62h-7F	EEPROM future expansion	

Backward Compatibility

Some of the Intersil WSG LDD (ISL58xxx) uses a page register, located at 02h, which also deselect the serial bus if page 3 (bits 4 and 5 = 11) is set. In the WSG LDD's, the page select bits serve a dual function. Setting bits 4 and 5 of Reg 02 disables WSG LDD serial interface. Perhaps at a future date, the WSG LDD's will employ this serial bus feature.

If WSG LDD and other devices are mixed in an OPU, the following procedure is recommended for initialization:

1. Write 00 to register 03h (DSR), disabling serial interface of all the devices except WSG LDD (since register 03h is a read-only status register for WSG LDD, write operation does not affect its contents).
2. The host communicates with WSG LDD.
3. Write xx11_xxxx to register 02h, disabling WSG LDD serial interface.
4. Write the other device DEN to register 03h to start communication with some other device.