Application Note 119
Embedding the 1-Wire® Master

www.dalsemi.com

INTRODUCTION

The DS1WM 1-Wire Master, termed 1WM, was created to facilitate host CPU communication with devices over a 1-Wire bus without concern for bit timing. This application note shows how to incorporate the 1-Wire Master into a user's ASIC design. The DS89C200 referred to in this document is a theoretical micro controller. It is assumed the reader has knowledge of the DS1WM 1-Wire Master and Dallas Semiconductor's 1-Wire protocol. For more detailed information see [1] **Book of iButton Standards** and [2] **DS1WM Datasheet**.

LIBRARIES

To compile the Verilog version of the 1-Wire Master, IEEE.std_logic_1164 is the only library required. The VHDL version requires both IEEE.std_logic_1164 and work.std_arith libraries.

CONNECTIONS

The following table lists the wires that needed to be connected for proper operation of the 1-Wire Master.

PIN	OPERATION
DQ	Open Drain 1-Wire Bus Connection
DATA	Bi-directional 8 Bit Data Bus
ADDRESS	3 Bit Address Bus
ADS_bar	Address Strobe
EN_bar	Instance Enable
RD_bar	Read Data Strobe
WR_bar	Write Data Strobe
INTR	Interrupt Detection
CLK	System Clock
MR	Mater Reset

Under most circumstances, DQ will be connected directly to a bi-directional I/O pad. If no address strobe is available in the system, the ADS_bar may be tied low making the address latch transparent. The EN_bar signal should be generated by address decoding logic external to the 1WMaster module. If the 1WM is the only instance on the data bus, EN_bar may be tied low. The system clock wired to CLK must be between 3.2 and 128 MHz. For detailed operation of all connections see the DS1WM 1-Wire Master datasheet.

INSTANCE

The following is an example of how to create a 1-Wire Master instance in Verilog.

```
module DS89C200 (...top level list...);

wire [7:0] DB;
wire [2:0] ADDR;
wire sysclk, read_bar,
     write_bar, master_reset,
     interrupt, addr_strobe;
wire DQ_OUT;

supply1 Tie1;
supply0 Tie0;

cpu xcpu(.CLK(sysclk),
         .DB(DB),
         .EXTRD_BAR(read_bar),
         .EXTWR_BAR(write_bar),
         .EXTADDR(ADDR),
         .RESET(master_reset),
         .EXTINTR(interrupt),
         .ADDR_ST(addr_strobe),
         ... other I/O signals ...);

onewiremaster xonewiremaster(
         .ADDRESS(ADDR),
         .ADS_bar(addr_strobe),
         .EN_bar(Tie0),
         .RD_bar(read_bar),
         .WR_bar(write_bar),
         .DATA(DB),
         .INTR(interrupt),
         .CLK(sysclk),
         .DQ(DQ_OUT),
         .MR(master_reset) );

... rest of design ...
```

All signals generated by xcpu meet the 1-Wire Master timing requirements. The EN_bar signal is tied low because there is no other addressable logic on the data bus. The DQ_OUT signal is wired directly to an I/O pad.

SIMULATION

An example of a Synopsys run script for this design is listed below. This script is for the Verilog version of the 1WM. A VHDL script would look very similar with .v files replaced with .hdl ones.

```
sh date

read -f verilog ../verilog/onewiremaster.v
read -f verilog ../verilog/DS89C200.v // all other modules
read -f verilog ../verilog/modules.v

include "../timing_scripts/DS89C200.const"

report_file = DS89C200.report

compile
```

```
report_design > report_file
report_area >> report_file
report_cell >> report_file
report_constraint -all_violators
      -verbose >> report_file
report_timing -path full -delay max
      -max_paths 10 >> report_file

write -f db -hier -o DS89C200.db

write -f verilog -hier -o DS89C200.v

write_sdf DS89C200.sdf -version 1.0
      -context verilog

quit
```

The example timing constraints file included in the script, "../timing_scripts/DS89C200.const", also contains some 1-Wire Master specific information. It would contain something very similar the following listing:

```
... other module constraints ...

current_design xonewiremaster

set_max_area 130000

create_clock sysclk -name sysclk
      -period 25 -waveform {0.0 12.5}
set_dont_touch_network sysclk
set_driving_cell -cell i1s4 -library
      csm0_5_worst -pin OUT all_inputs()
set_load 3.0 all_outputs()

<individual timing constraints
     (set_input_delay/set_output_delay)
      etc >
```

These examples are very generic. The actual runscript and constraint files would be generated by the engineer to meet the timing requirements of the specific design. One thing to bear in mind, the timing in the 1-Wire Master block is not entirely synchronous by design. The DQ output is synchronized to CLK, but the bus read/write timing will only be synchronous to CLK if the CPU uses CLK to generate RD_bar, WR_bar and ADS_bar. See the specification for the timing relationships for these signals.

REFERENCES:

[1] **Book of iButton Standards**, Dallas Semiconductor, online at http://www.ibutton.com/iButtons/standard.pdf
[2] **DS1WM Datasheet**, Dallas Semiconductor, online at http://www.dalsemi.com/DocControl/PDFs/1WM.pdf