

Figure 1: Slave Parallel and EPROM Interface Circuit Diagram

Note:

1. BUSY is only needed when the oscillator is running at a frequency greater than 50 MHz.

Slave Parallel Mode

Slave Parallel mode uses a byte-wide bidirectional access port for reading and writing the configuration data of a Spartan-II FPGA. It is very similar to the Express mode used by Spartan-XL FPGAs. The following signals are used in Slave Parallel mode.

D[7:0]

D[7:0] comprise the 8-bit bidirectional data bus.

CCLK

CCLK is the configuration clock input signal used by the internal configuration logic.

PROGRAM

The PROGRAM input signal resets the internal configuration logic and re-initializes the internal configuration memory.

DONE

The DONE output indicates the completion of configuration and the beginning of the Startup sequence.

INIT

The bidirectional open-drain INIT pin is used to hold off configuration initialization, and it indicates when CRC errors occur in the configuration data.

WRITE

The WRITE input is a write strobe that must be asserted and held throughout the loading of data.

BUSY

When $\overline{\text{CS}}$ is asserted, BUSY output indicates whether the current byte is being loaded or ignored. When $\overline{\text{CS}}$ is not asserted, BUSY is disabled and pulled High.

$\overline{\text{CS}}$

The $\overline{\text{CS}}$ input is the Chip Select signal used to enable the FPGA's configuration interface when the Slave Parallel mode is selected.

M2, M1, and M0

The MODE pins M[2:0] must be set to select the desired configuration mode. For Slave Parallel configuration these lines must be driven High, High, and Low, respectively.

Configuration Process Steps

The following are the steps for performing a Slave Parallel configuration.

Reset the Configuration Logic

If configuration follows power-up, or a recycling of the power source, then the configuration logic will automatically be initialized. However, if the FPGA is to be reconfigured without the recycling of power, then the PROGRAM input must be asserted Low for at least 300 ns to reset the configuration logic.

Time-out Start of Configuration

After the release of the $\overline{\text{PROGRAM}}$ input, or the powering up of the FPGA, the $\overline{\text{INIT}}$ output stays Low while the internal configuration memory is automatically cleared.

Unlike previous generations of FPGAs, the Spartan-II device does not require an additional time-out period following initialization. Configuration may begin as soon as $\overline{\text{INIT}}$ has gone High. The Spartan-II FPGA uses the first three clock cycles after $\overline{\text{INIT}}$ has gone High to initialize the configuration circuitry; however it is padded with eight dummy bytes at the beginning of the data stream to account for this.

Loading Configuration Data

To begin configuration, the $\overline{\text{WRITE}}$ and $\overline{\text{CS}}$ inputs must be asserted Low and held. Each byte is loaded on the rising edge of CCLK. If BUSY was Low during the CCLK transition then the byte was accepted. If it was High then the byte was ignored and must be reloaded on the next clock cycle. BUSY is a synchronous signal. Therefore, CCLK can not be suspended until BUSY is de-asserted.

For configuration clocks speeds below 50 MHz, BUSY is guaranteed to be inactive, and thus may be ignored entirely, and removed from the interface design.

End of Configuration

The DONE output transitions High to indicate the end of configuration and the beginning of the startup sequence. During the startup sequence the D[7:0], $\overline{\text{WRITE}}$, BUSY, $\overline{\text{INIT}}$, and $\overline{\text{CS}}$ pins all become user I/O. If any of these pins are used by the configured design, then any driving source used only for configuration purposes must be disabled so as not to contend.

Depending on the startup options selected in the BitGen configuration option template, part of the Xilinx development software, completing the startup phase will require as many as eight CCLK cycles after DONE has transitioned High. However, since the example shown below uses a free-running oscillator, this is not a concern. The Spartan-II FPGA will complete the startup phase even if the CS signal is de-asserted.

The PROMMAP Design

The CPLD design PROMMAP, shown in Figure 2, consists of an address counter, a write control register, some tristate buffers and random logic. The PROGRAM input acts as a global reset. DONE acts as a global tristate. INIT and BUSY hold off the address counter from incrementing while the FPGA initializes or processes data.

Address Counter

The address counter needs to be large enough to accommodate the address bus width of the PROM. In Figure 1 a 1-Mbyte (8-Mbit) PROM is shown which requires a 20-bit address bus. For connecting multiple PROMs see "Interfacing Multiple PROMs" on page 5.

Write Control Register

The \overline{CS} and \overline{WRITE} input signals to the FPGA must be asserted and held Low for configuration. If only one FPGA is being targeted for configuration, these two signals may be derived from the same source. For targeting multiple FPGAs see "Multiple FPGAs in a Parallel Chain" on page 6.

Configuration must be delayed until the \overline{INIT} signal has transitioned High indicating that FPGA initialization has completed.

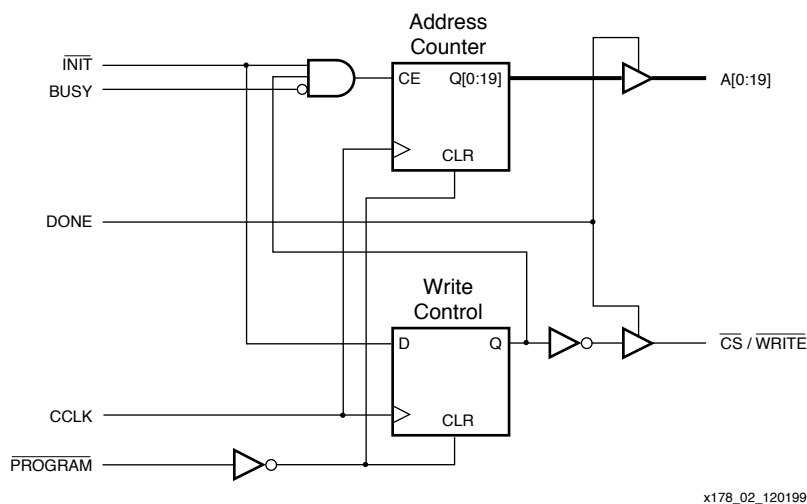


Figure 2: PROMMAP Interface Design

Oscillator

The configuration logic of the FPGA was designed to work with a free-running oscillator. In this design the maximum frequency for the oscillator is constrained by the access time of the PROM (T_{ACC}) plus the setup time for the Slave Parallel data inputs (T_{SMDCC}).

$$\text{Oscillator Frequency} = \frac{1}{T_{ACC} + T_{SMDCC}}$$

PROMMAP Design Files

The PROMMAP design was built in an XC9536VQ44-10 and tested with an XCV300BG432 and an AT27c080. VHDL and Verilog source code examples, as well as a Foundation Project Schematic, for the design can be downloaded from the Xilinx website at <ftp://ftp.xilinx.com/pub/applications/xapp/xapp137.zip>.