Configuration Files and Cells

The end result (as you know) of the design flow is a *configuration file* or *bit file* that is used to program the FPGA.

For SRAM-based, EEPROM and FLASH FPGAs, the *bit file* contains both *configuration data* and *configuration commands*.

The commands tell the FPGA what to do with the data.

For anitfuse, the file contains only configuration data.

Configuration cells are present in the device to allow the interconnect, I/O and other features of the FPGA to be programmed.

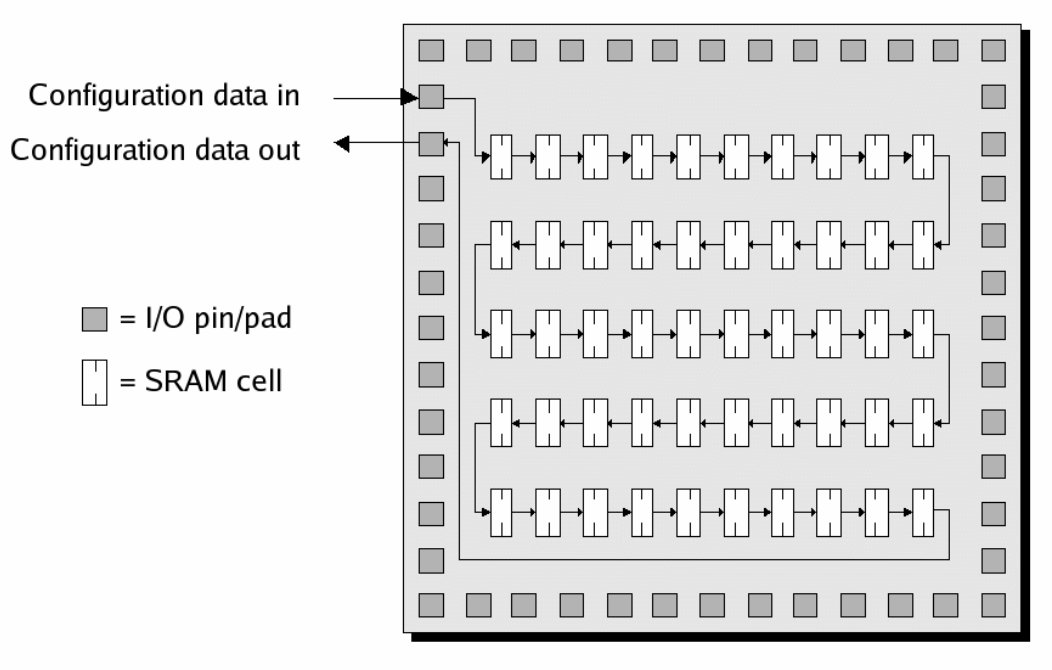
For example, within the PLB, the MUX's *select input* needs a configuration cell.

The register needs configuration cells to specify whether it is to act as a FF or a latch, whether it latches using a positive- or negative-going clk, and whether it is initialized to a 0 or a 1.

Configuring the FPGA

A 4-input LUT contains 16 configuration cells.

The configuration cells are typically connected in a long *scan chain*.



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

The *scan chain* (when programmed in this mode) has an input and an output.

The output is used if multiple FPGAs are *daisy-chained*.

Embedded RAMs are implemented as latches and are part of the scan chain.

Configuring the FPGA

Note that this is a simplistic view of the FPGA's internal organization.

In reality, the scan chain is made from latches, not FFs.

Latches are **half** the size - saves a lot of real estate with 25 million.

Also, *frames* of 1024 bits are clocked into a set of FFs and loaded in parallel to a *frame* of latches as the file is loaded.

Also, in some FPGAs, the very long scan chain is actually divided into *multiple smaller chains*, that are loaded by the *configuration port*.

The *configuration port* has several modes, controlled by dedicated pins.

Mode Pins	Mode
0 0	Serial load with FPGA as master
0 1	Serial load with FPGA as slave
1 0	Parallel load with FPGA as master
1 1	Parallel load with FPGA as slave

Each vendor defines this differently.

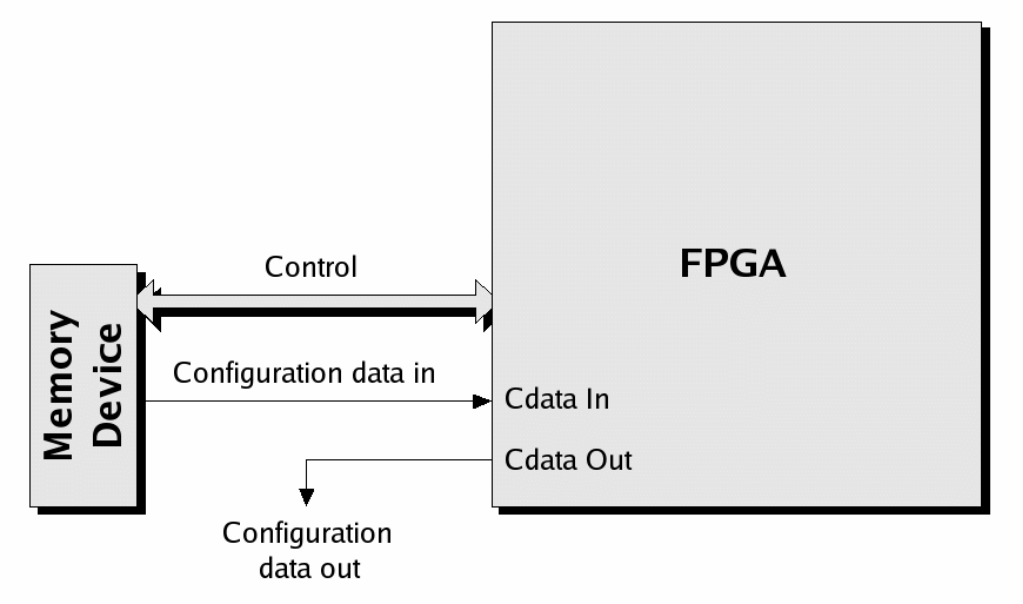
The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

Configuring the FPGA

Other pins are used to tell the FPGA to commence with the configuration, and to report an error and that the configuration is complete.

The pins dedicated to the *configuration port* can be reused as general purpose I/O once the configuration is complete.

The **serial load with FPGA as master** is the simplest mode.



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

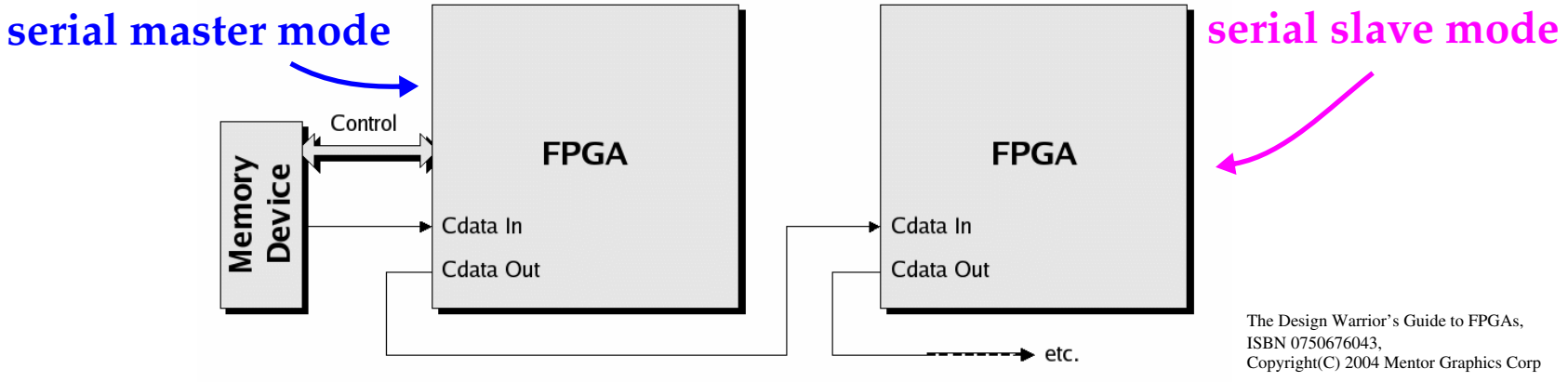
The *memory device* is usually a special FLASH with a single data out pin.

Configuration Modes of the FPGA

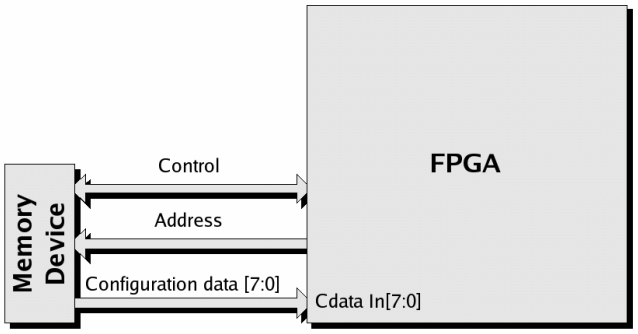
The FPGA controls the FLASH device through a reset and clock pin.

In this mode, there are no addresses that need to be issued by the FPGA.

As indicated, the *Cdata Out* pin is used when daisy-chaining.



In contrast, **Parallel load with FPGA as master** uses addresses.



Configuration Modes of the FPGA

Here, the FPGA maintains an internal counter that generates the byte addresses.

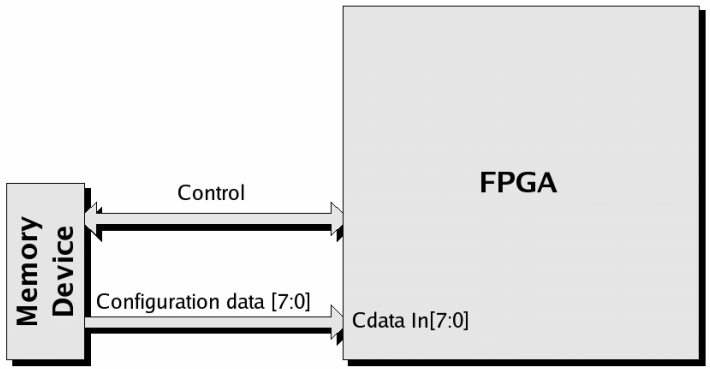
The advantage is reduced configuration time and off-the-shelf memory chips.

However, in the early days, the fetched byte was serially clocked into the scan chain, so it wasn't faster.

Similar to *serial mode*, these pins can be reused as general purpose I/Os.

Unfortunately, this is not often used because of the load associated with the hard-wired memory device.

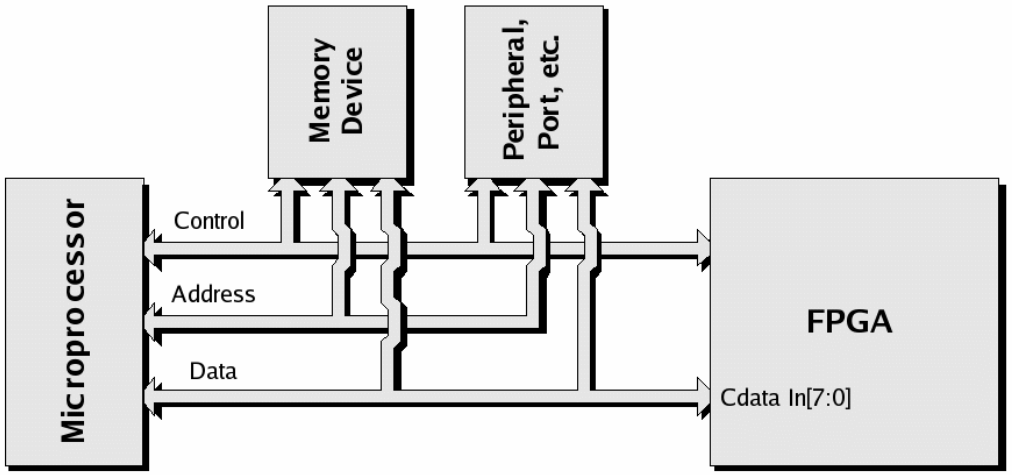
Modern devices use a variation of the *serial load*.



Parallel load without the need to send addresses

Configuration Modes of the FPGA

Boards that incorporate microprocessors can use the **parallel load with FPGA as slave** mode (or **serial load with FPGA as slave**).



The Design Warrior's Guide to FPGAs,
ISBN 0750676043,
Copyright(C) 2004 Mentor Graphics Corp

In this case, the microprocessor fetches the byte data and loads the FPGA.

This allows a great deal of flexibility w.r.t. how the chip is programmed.

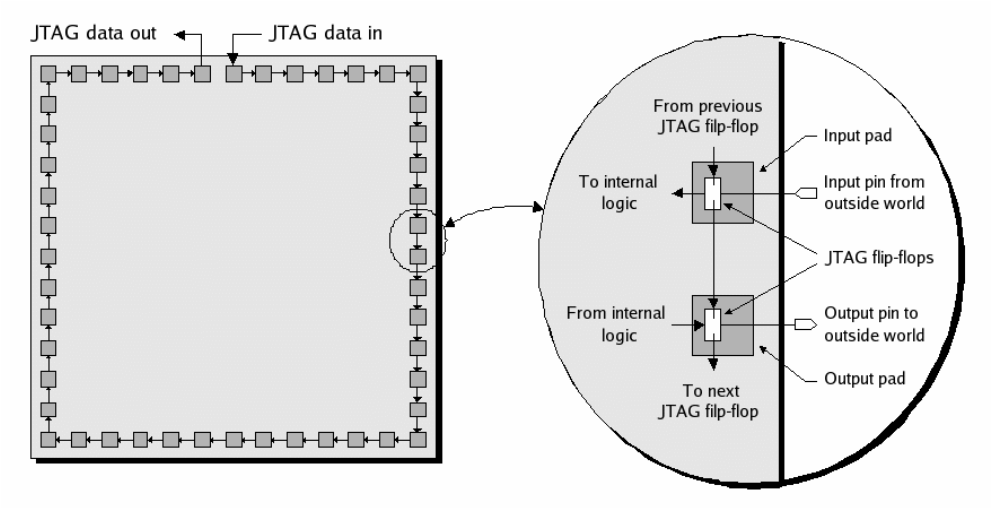
The **JTAG port** can also be used to program the FPGA.

Joint Test Action Group -> JTAG.

JTAG was originally introduced to enable *boundary scan*, which is used to test PCBs and the chips on the board.

Configuration Modes of the FPGA

The JTAG port has several dedicated pins, two of which are for data input and output, and an internal *scan chain* the connects to all FPGA I/O pins.



For *boundary scan* testing, data is scanned into the I/O scan registers, the chip operates on the data and the results are scanned out for verification.

The JTAG port also contains a command register (not shown) that can be loaded to inform the FPGA to get its configuration data from the JTAG port.

Thus, typically there are 3 mode pins in today's FPGA to allow this mode to be specified (and another, see text for an embedded processor mode).