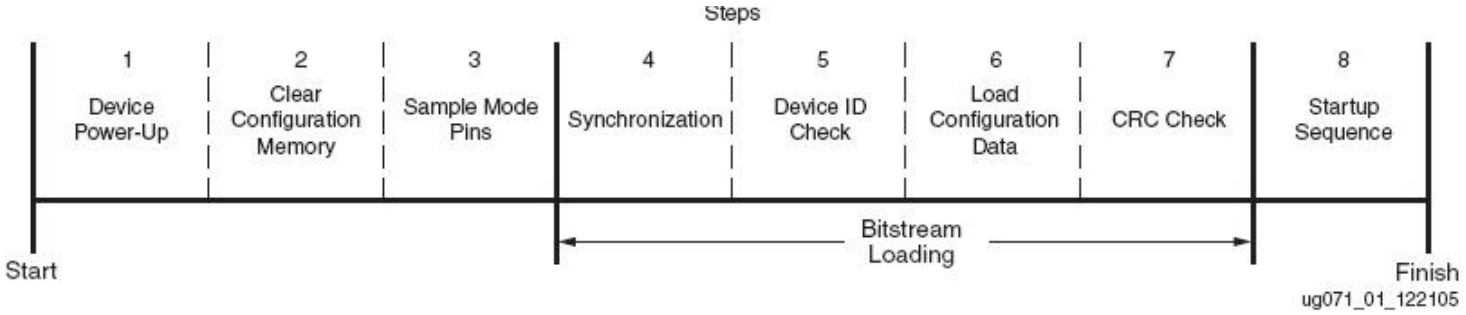
FPGA In-System Configuration

Haibo Wang
ECE Department
Southern Illinois University
Carbondale, IL 62901

FPGA Operation Modes

- FPGAs normally have two operation models: "configuration mode" or "user mode".
- Immediately after power-up, FPGAs are automatically in configuration mode and all their outputs are at high impedance states
- FPGAs can be also switched to configuration model by activating configuration pins (e.g. applying low voltage at PROG_B pin)
- Example: Virtex-4 power-up sequences

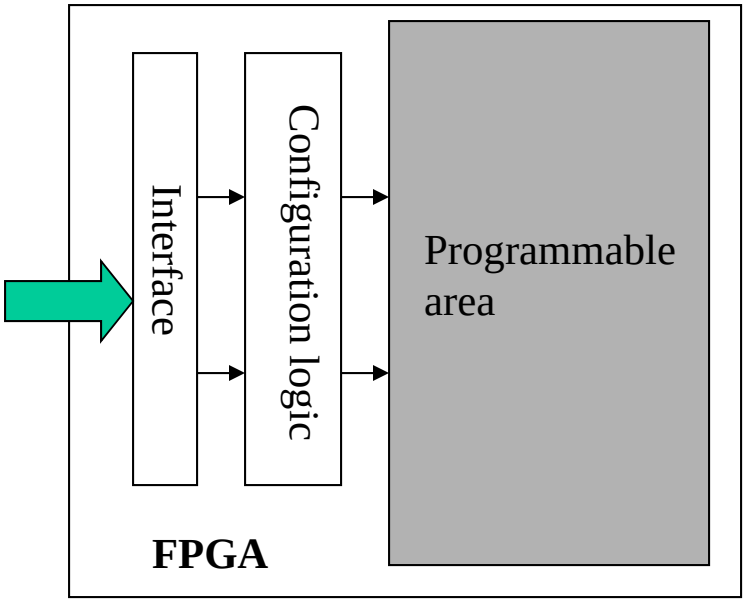


FPGA Configuration Methods

- FPGAs will switch to user mode after configuration. The configuration methods normally include:
  - Download configuration bitstream from a PC
  - An on-board microcontroller sends configuration bitsteam to FPGA
  - FPGA is configured by data from on-board boot PROM
- Common FPGA configuration interfaces include:
  - The JTAG interface
  - Synchronous serial interface
  - Synchronous parallel interface
  - …

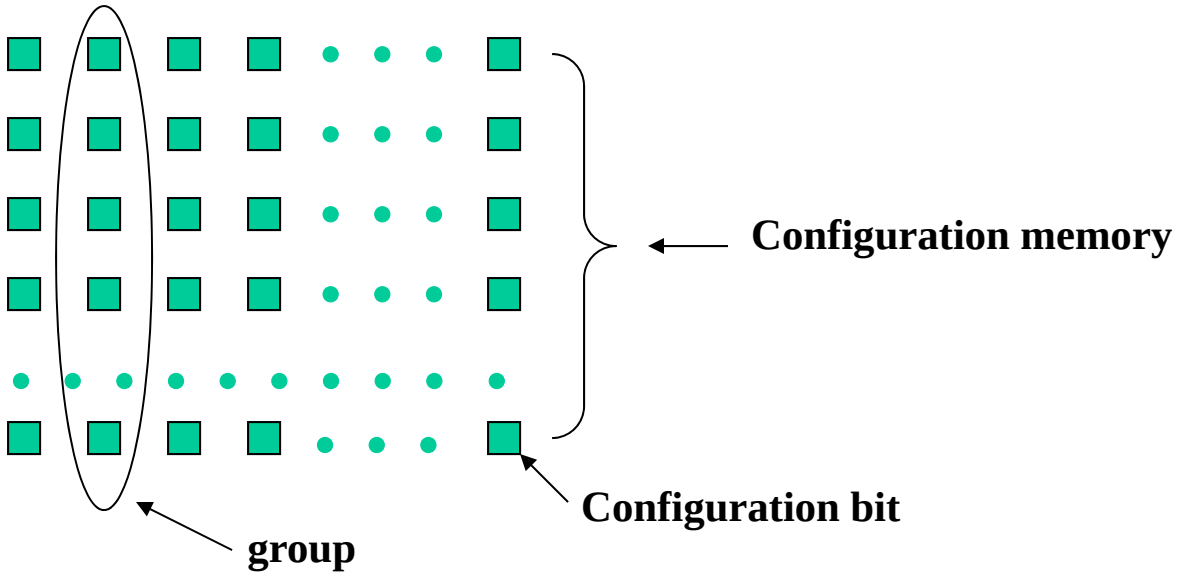
Overview of FPGA Configuration Mechanism

- An FPGA can be partitioned into non-programmable and programmable area.
- Non-programmable area includes: all or parts of configuration interface and configuration logic.
- Programmable area includes: CLBs, portion of IOBs, routing resources, etc.
- In reconfiguration mode, configuration logic gets configuration bitstream from interface circuits and write them into proper locations in configuration memory.



FPGA Configuration Memory

- FPGA configuration memory can be visualized as a rectangular array of bits.
- Configuration bits are arranged into groups, e.g. frames and columns in Xilinx FPGAs
- Addresses are assigned to configuration bit groups such that they can be selectively accessed by configuration logic.



FPGA Configuration Logic

- Configuration logic contains a set of registers, which control the operation of configuration logic and reflect the status of configuration operation
- Addresses are assigned to registers for accessing registers.
- In a configuration operation, controls registers are first loaded with proper values before configuration bits arriving
- Example: Registers in Xilinx Spartan 3 configuration logic

Name	Mnemonic	Read/Write	Binary Address
Cyclic Redundancy Check	CRC	R/W	00000
Frame Address Register	FAR	R/W	00001
Frame Data Input Register	FDRI	W	00010
Frame Data Output Register	FDRO	R	00011
Command Register	CMD	R/W	00100
Control Register	CTL	R/W	00101
Mask Register	MASK	R/W	00110
Status Register	STAT	R	00111
Legacy Output Register	LOUT	W	01000
Configuration Options Register	COR	R/W	01001
Multiple Frame Write Register	MFWR	W	01010
Frame Length Register	FLR	R/W	01011
(Reserved)	–	–	01100
(Reserved)	–	–	01101
Product IDCODE Register	IDCODE	R/W	01110
Partial Reconfiguration Register	SNOWPLOW	W	01111

FPGA Bitstream Composition

- FPGA bitstreams normally include three parts
  1. First a synchronization word
  2. Packages of commands and data for writing or reading registers in configuration logic (configuration memory is updated through registers in configuration logic)
  3. Data that used to perform error checking

- All the data that will be written into registers and configuration memories are encapsulated into packages. Each package starts with a package header.

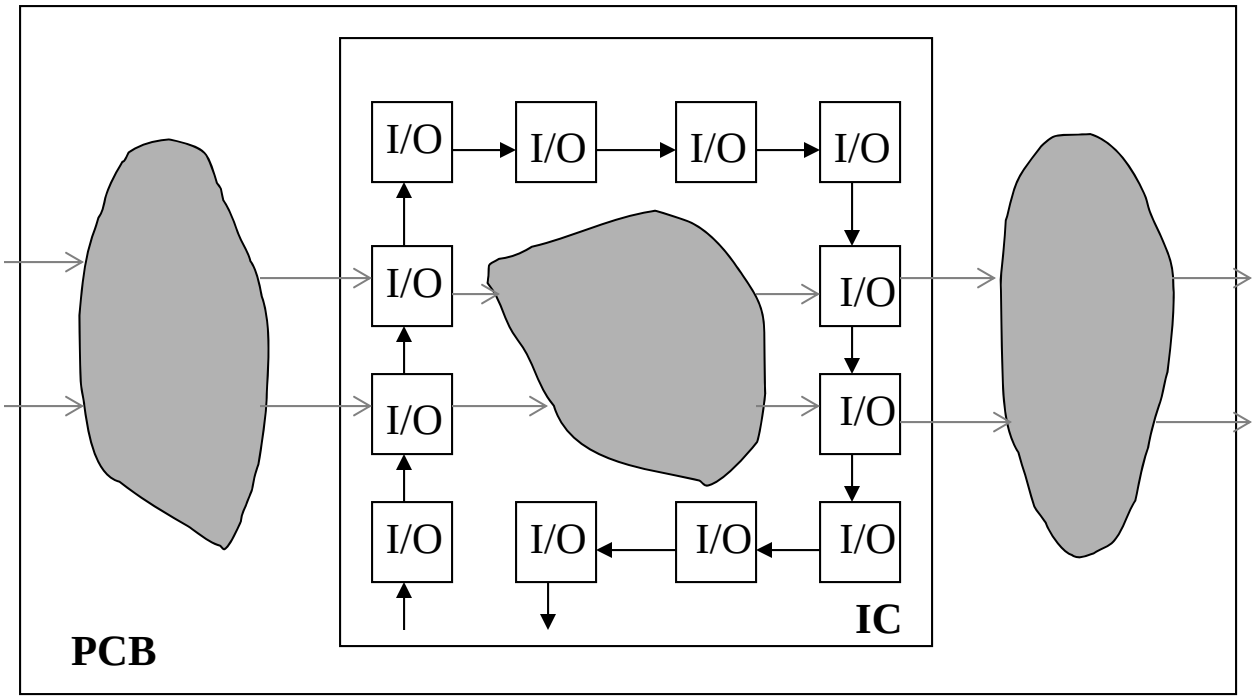
- Example: Xilinx Spartan 3 type-1 package header

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Type			Op		Register Address														RSVD		Word Count											
0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x	x	x	x

Figure 6: **Type 1 Header**

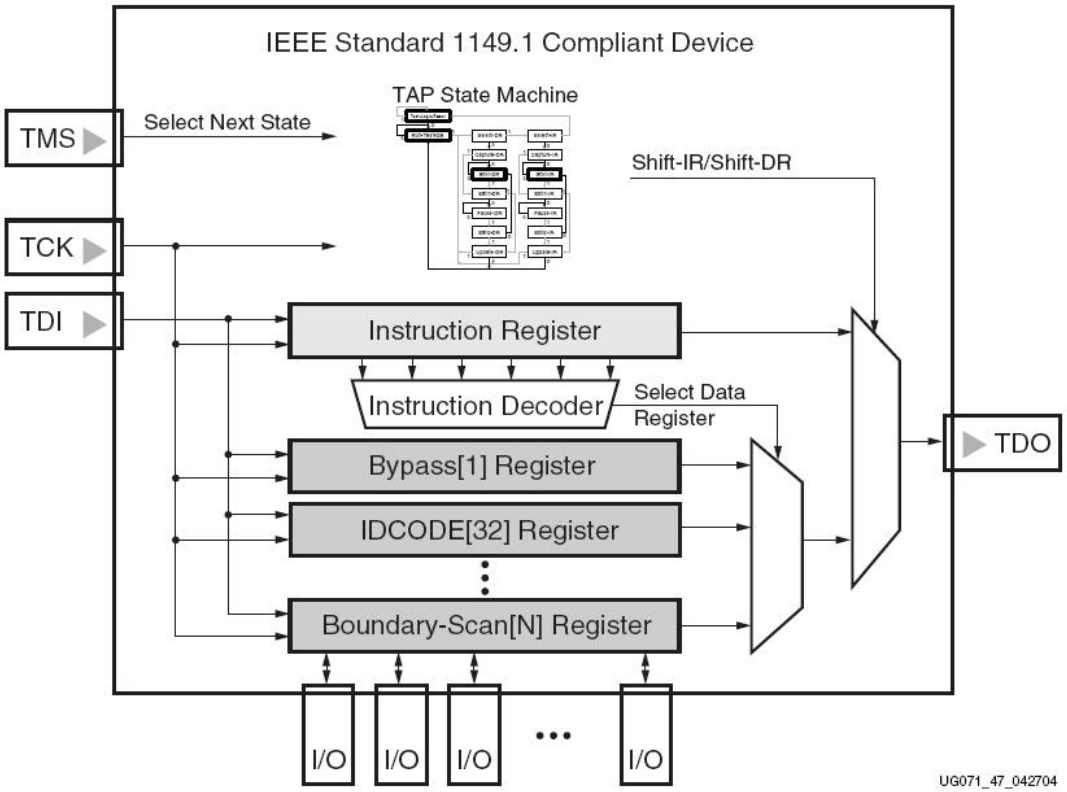
FPGA JTAG Interface

- The JTAG interface is originally designed for testing purpose. It provides a mechanism to shift testing vectors into IC I/O ports and shift circuit responses from IC I/O ports.



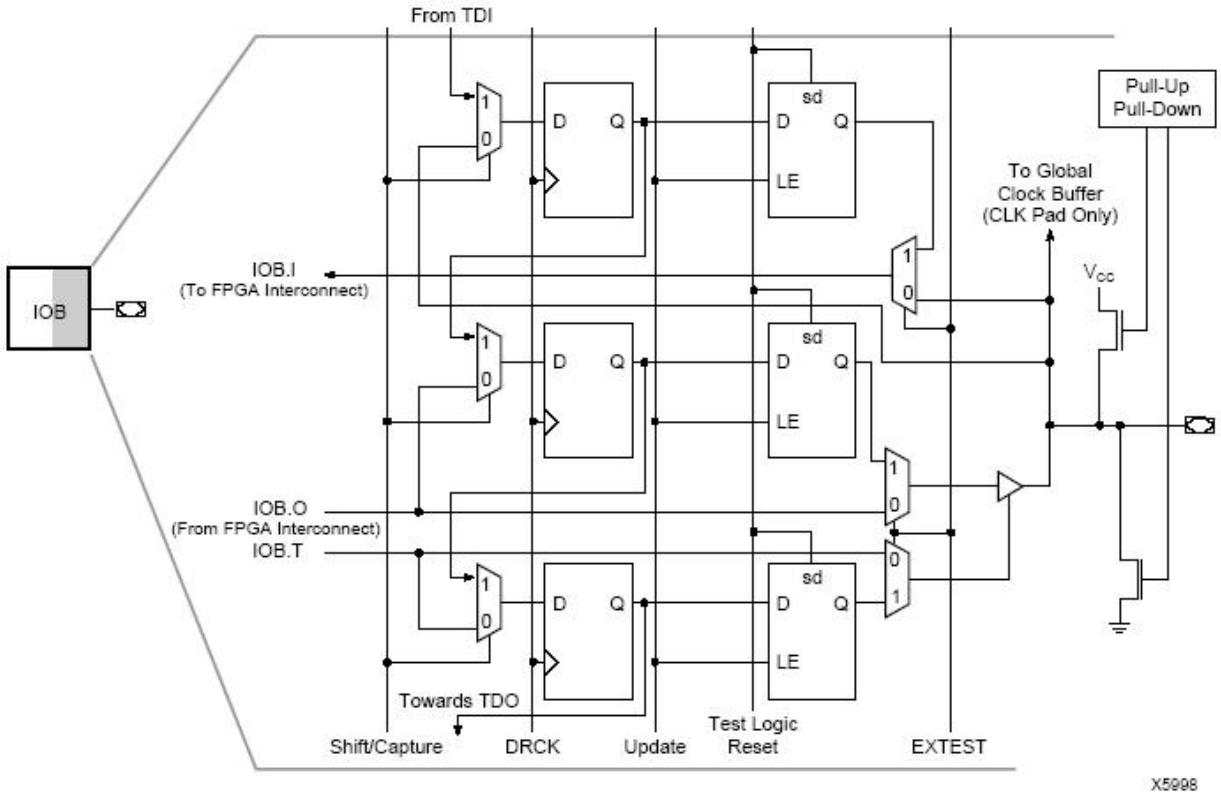
FPGA JTAG Interface

- A JTAG interface normally includes four pins: TDI, TDO, TCK, TMS
- Example: JTAG interface in Xilinx FPGAs



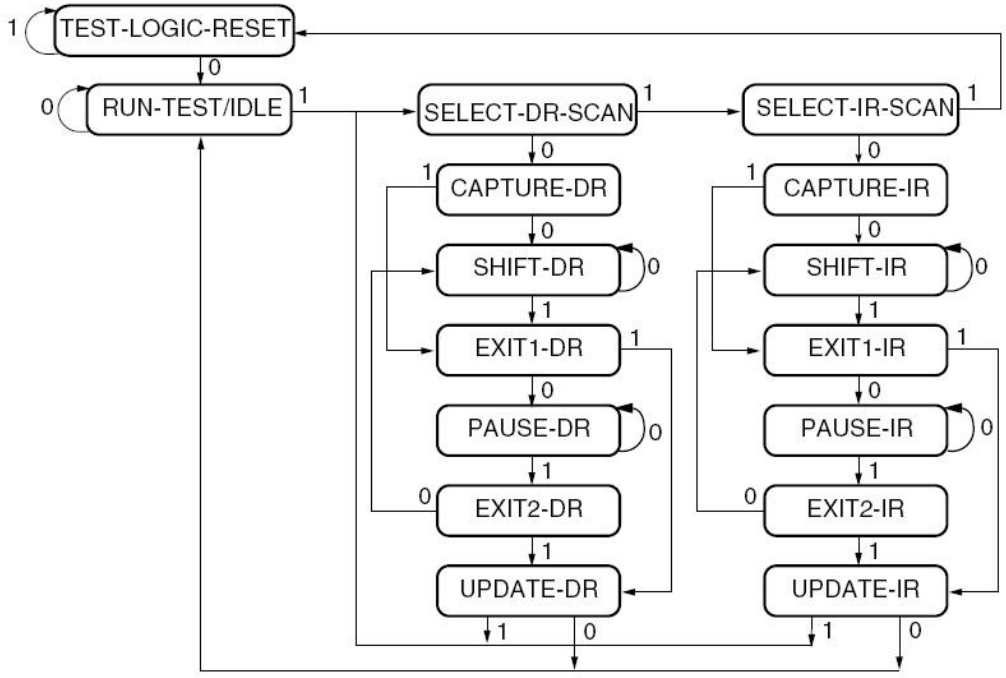
FPGA JTAG Boundary-Scan Chain

- D flip-flops and multiplexers are added to FPGA IO cells to implement JTAG boundary-scan chain.
- Example: Each Xilinx FPGA IOB contains three bits of the boundary-scan chain.



JTAG TAP State Machine

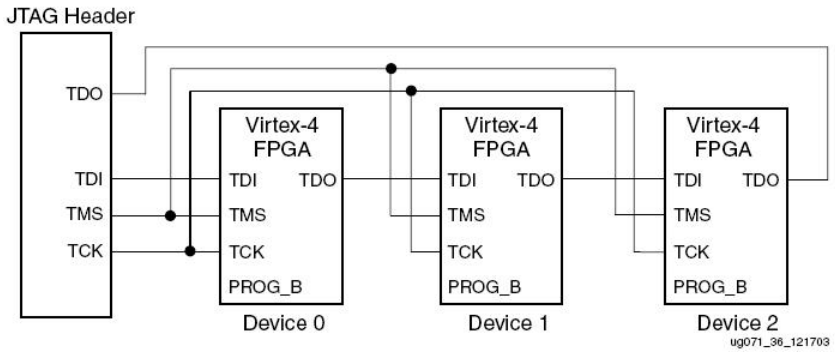
- Data shifting operation in a JTAG scan chain is controlled by a Test Access Port (TAP) state machine.
- State transitions of the TAP FSM is controlled by TMS and TCK



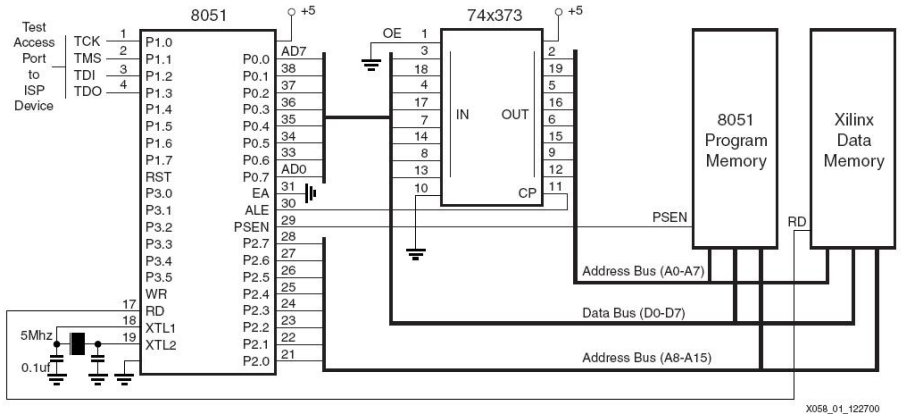
NOTE: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

FPGA configuration via Boundary-scan chain

- Example: configuring multiple Virtex-4 devices via JTAG chian



- The JTAG header can be implemented using a CPLD or microprocessor.

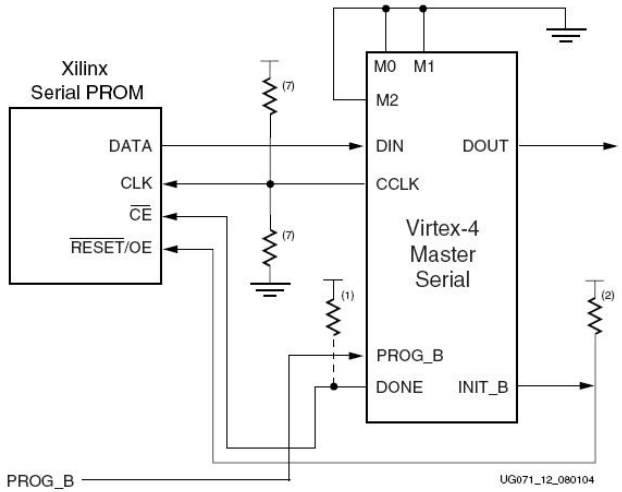


Other Configuration Interfaces in Xilinx FPGAs

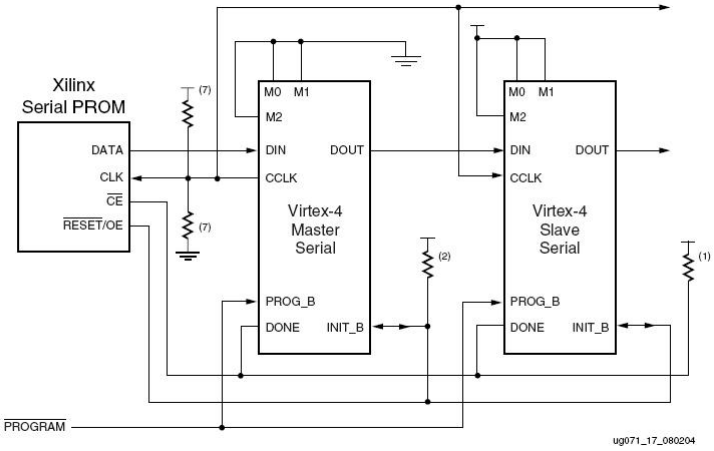
- Select-MAP:
  - External clock is needed
  - Data is loaded one-byte per clock cycle
  - It is desirable when configuration speed is a concern
- Master-Serial Model:
  - Using internal clock
  - Data is loaded one-bit per clock cycle
- Slave-Serial Model:
  - External clock is needed
  - Data is loaded one-bit per clock cycle
  - Allow daisy-chain configuration
- Configuration mode is selected by applying proper values at model selection input pins M[2:0]

Serial Configuration Examples

- Master serial mode configuration:

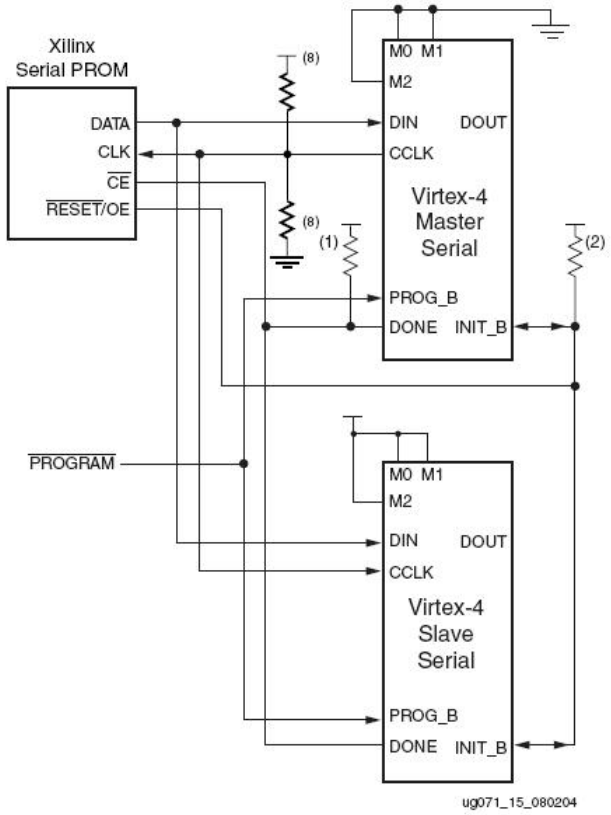


- Master/slave serial mode daisy chain configuration:



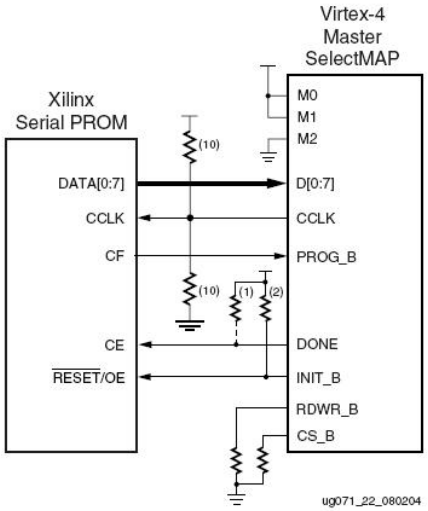
Serial Configuration Examples

- Ganged serial mode configuration:

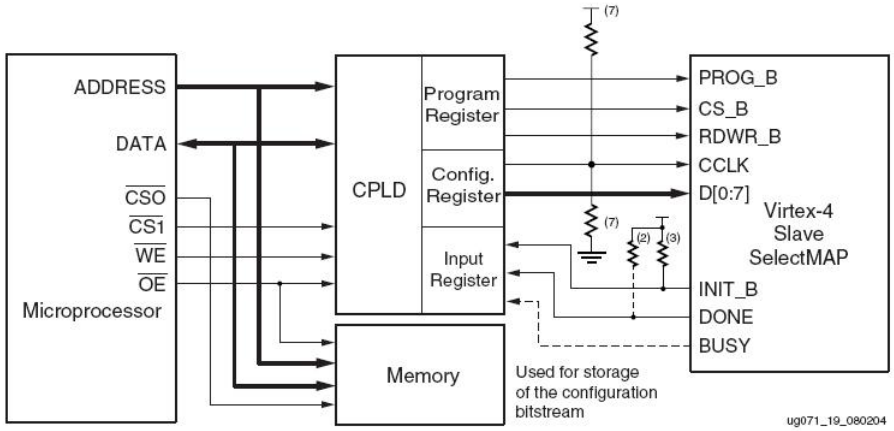


Parallel Configuration Examples

- Master SelectMAP configuration:

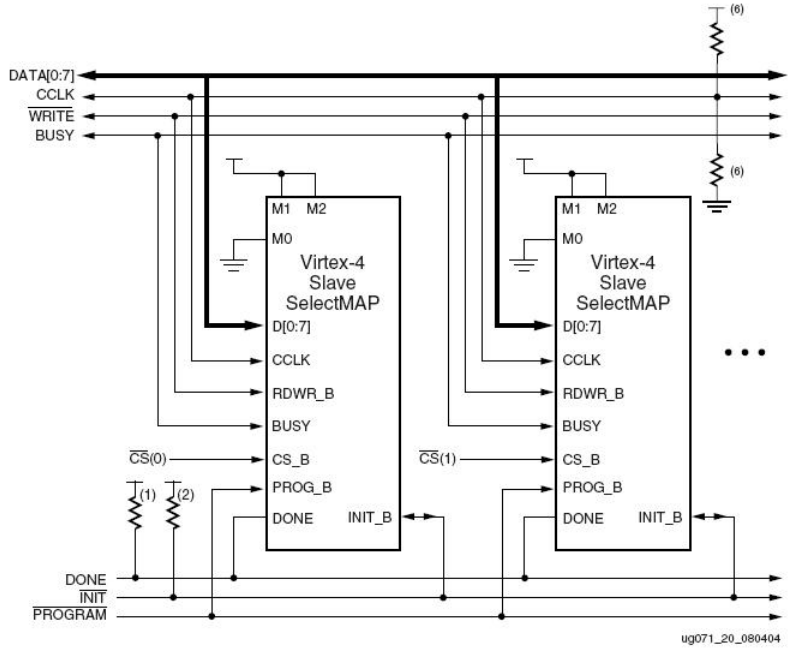


- Slave SelectMAP configuration:

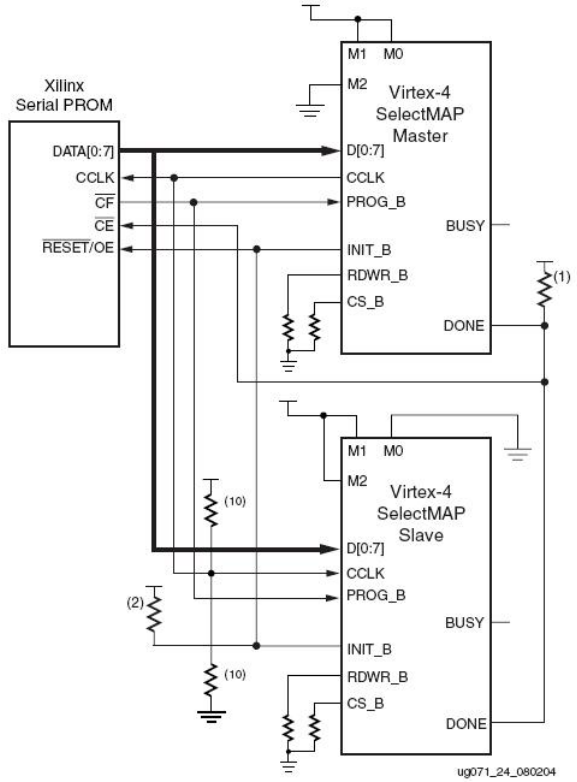


Parallel Configuration Examples

- Multiple slaves in SelectMAP configuration:



- Ganged slaves in SelectMAP configuration:

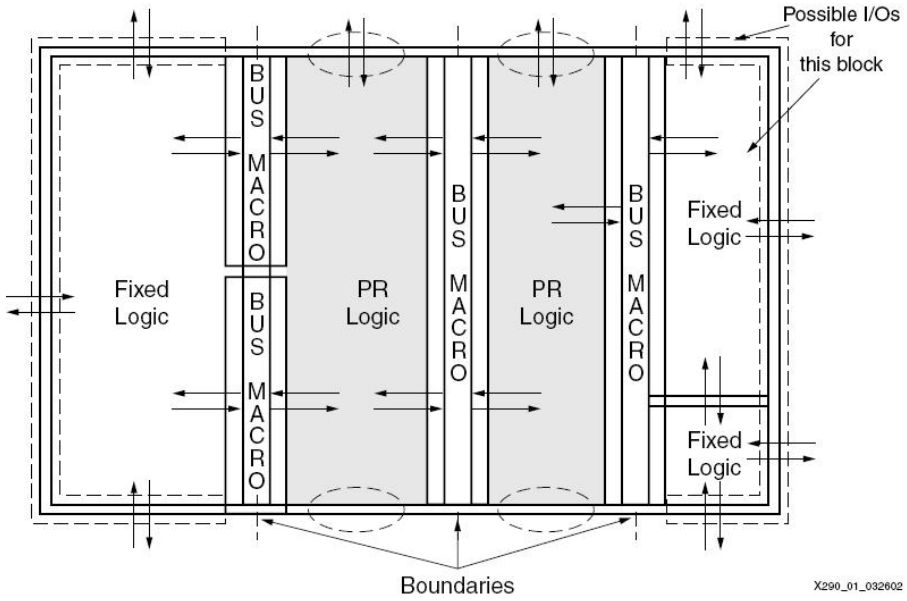


FPGA Partial Reconfiguration

- Partial reconfiguration is a design process, which allows a limited, predefined portion of an FPGA to be reconfigured while the remainder of the device continues to operate.
- Applications
  - In-the-field hardware upgrades and updates to remote sites
  - Runtime reconfiguration
  - Adaptive hardware algorithms
  - Continuous service applications
- Other Advantages
  - Reduced device count
  - Reduced power consumption
  - More efficient use of available board space

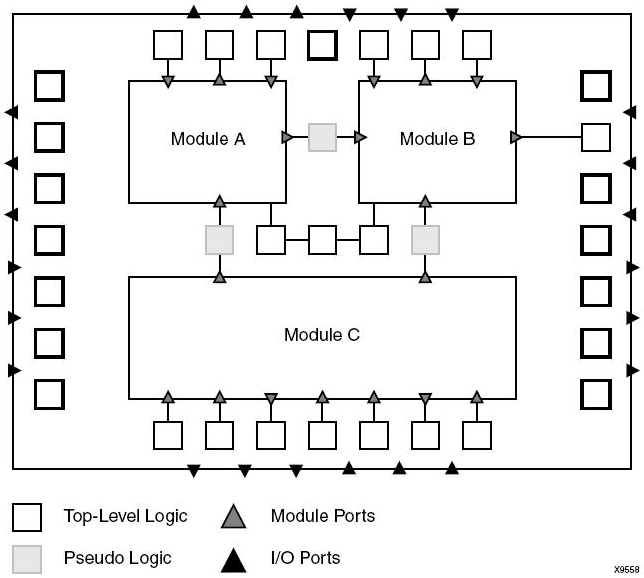
Example: module-based partial reconfiguration approach for Virtex FPGAs

- The chip layout is partitioned into fixed and reconfigurable areas.
- The reconfigurable module height is always the full height of the device.
- Reconfigurable modules communicate with other modules, both fixed and reconfigurable, by using a special bus macro.
- Static portions of the design do not rely on the state of the module under reconfiguration while reconfiguration is taking place.

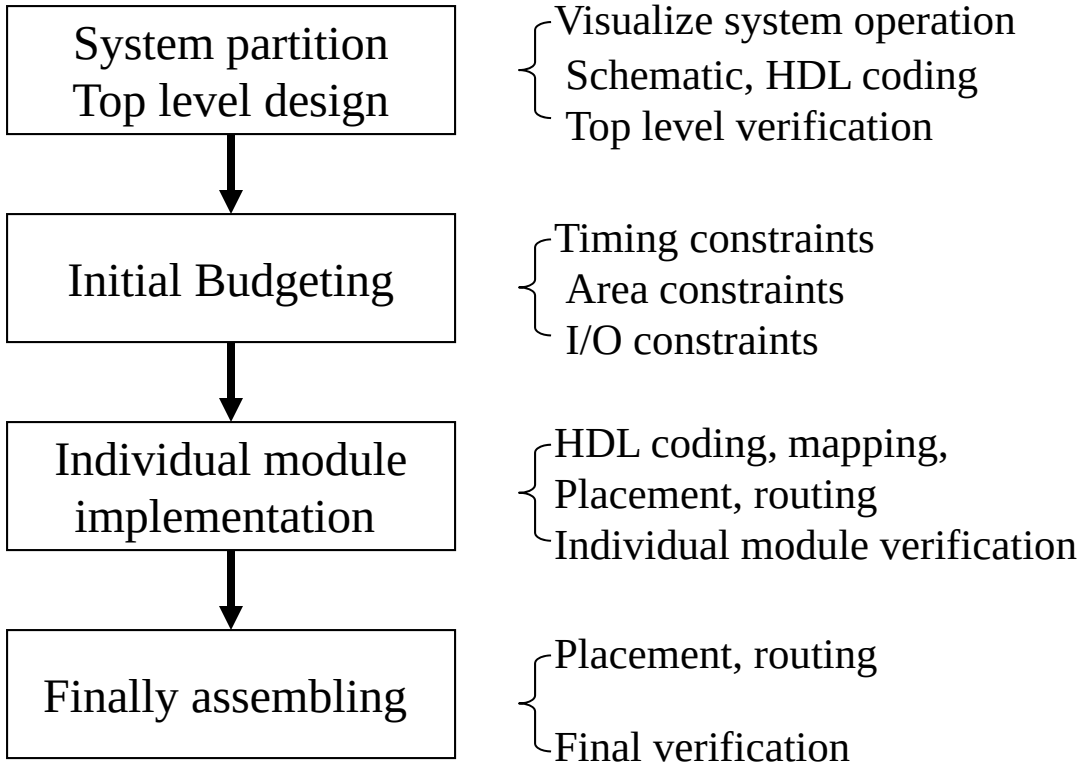


Modular Design

- Top-down design approach
- Design activities start from partitioning a complex system into several self-contained sub-design (modules)
- The top level of the design contains global logics (e.g. clock, I/O circuits) and instantiated modules.
- At the top level, instantiated modules are treated as black-boxes and only communications (ports) between modules are described.



Modular Design Flow



An example of Top-level Verilog Code

```
module top (clk, rst, in1, in2, out1, out2);
input clk, rst, in1, in1, in2;
output out1, out2;
wire clk_buf, a, b, c, d, e, f, h;

// clock circuit
IBUFG ibuf_dll (.I(clk), .O(clk_buf));
CLKDLL dll_1 (.CLKIN(clk_buf),
…..

// global logic
assign b =  a*in1;
…..

// Instantiation  module
M1 insta_1 (.in1(a), .in2(b), .out(c));
M2 insta_2 (.rst(rst), .clk(clk_buf),
.in1(e), .in2(f), .out(h));
…
endmodule
```

```
module M1 (in1, in2, out);
input in1, in2;
output out;
endmodule


module M2 (rst, clk, in1, in2, out);
input rst, clk, in1, in2;
output out;
endmodule
```

Initial Budgeting

- Tasks in initial budgeting
  - Position global logic
  - Size and position each module on the target chip
  - Position the input and output ports for each module
  - Budget initial timing constraints