Partial Reconfiguration Design Considerations

Objectives

- After completing this module, you will be able to:
  - Identify Partial Reconfiguration design requirements and guidelines
  - Describe the Partial Reconfiguration design recommendations
  - Describe the PR tools flow requirements
  - State the clocking considerations for the PR designs

Outline

- Tools Features and Functionality
- Design Recommendations
- PR Tool Flow Requirements
- Clocking Considerations
- Summary

Tools Support

- Device support
  - Virtex®-4, Virtex-5, and Virtex-6 families
  - Virtex-7 family support available starting 13.3 release
- PR is supported via the PlanAhead™ tool or command line only
  - No Project Navigator support
- Floorplanning is required to define reconfigurable regions, per element type
  - For greatest efficiency, align to frame/clock region boundaries when possible

Tools Support

- Standard timing constraints are supported, and additional timing budgeting capabilities are available if needed
- A unique set of Design Rule Checks (DRCs) has been established to guide users on a successful path to design completion
- Not all implementation options are available to the PR flow
  - The -global_opt option to the MAP command (as well as its child options, the -power option to the MAP command) and SmartGuide™ technology cannot be used with partitions or PR because these techniques perform optimization across the entire design

Tools and Design Performance

- Performance metrics will vary from design to design
  - Expect 10% degradation in clock frequency
  - Expect to not exceed 80% slices in packing density
- Longer design run times
  - MAP will display the greatest impact, but NGDBuild and PAR could also show the effects of processing a PR design
- Routing challenges can occur if the reconfigurable region is too small or is constructed of non-rectangular shapes
- Encryption support for partial bit files
  - Available natively for the Virtex-6 FPGA and Virtex-7
    - The ICAP must be used with 8-bit interface. Reconfiguration through external configuration ports is not permitted
  - Available via an IP core for the Virtex-5 FPGA (see XAPP887)

Outline

- Tools Features and Functionality
- **Design Recommendations**
- PR Tool Flow Requirements
- Clocking Considerations
- Summary

Design Recommendations

- The following design recommendations will help you to manage, enhance, and enable the PR flow
  - Utilize synchronous design techniques
  - Follow stringent hierarchy guidelines
  - Register inputs and outputs of RM
  - Control RM fanout manually
  - Use static decoupling logic
  - Reset in-coming RM
- In addition, a PR design must consider the initiation of Partial Reconfiguration as well as the delivery of partial bit files, either within the FPGA or as part of the system design

Synchronous Design

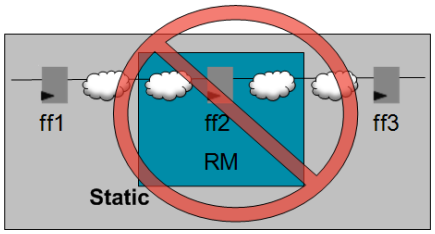
- Synchronous design methodology
  - One clock, one edge (all flip-flops use rising or falling edge)
  - Use D-type flip-flops
  - Partition hierarchy along functional lines while minimizing the interaction of blocks
  - Register the outputs of each behavioral block
  - Use clock enables in place of multiple clocks
  - Synchronize asynchronous signals to the “single” clock (synchronization circuits)

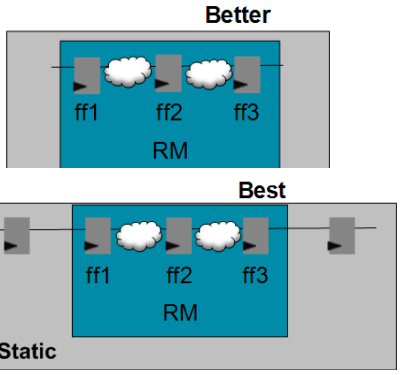
Design Hierarchy

- The number of partitions is important
  - Reconfigurable Partitions are required in PR design
  - Static partitions are based on need for design preservation techniques
  - Too many static partitions can affect performance because there are no optimizations across partition boundaries
    - Nor are there optimizations across RPs
- Do not use constants as inputs to a partition
  - Constants are not propagated across boundaries in a bottom-up synthesis flow
  - Using input constants to disable unused features of a design in a partition can cause utilization and quality-of-results issues
    - Unused logic is optimized away in the flat flow, but not when partitions are used

Static-RP Interface

- Register inputs and outputs of Reconfigurable Modules (RMs)
  - DRCs in the PlanAhead tool (Partition DRC)
  - Highly recommended to register the RM drivers and loads

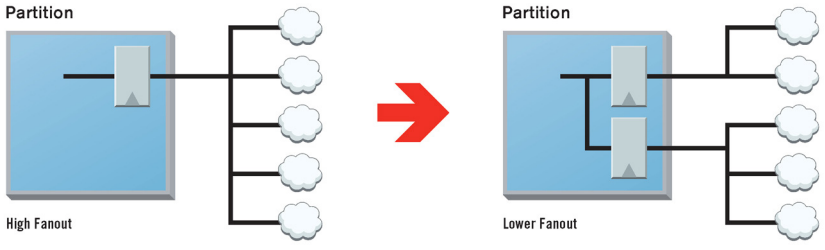




- Combinatorial logic will not be combined across partition boundaries
- Best implementation will have register-to-register delays between static logic and RM I/O
  - Static registers will act as decoupling logic

Fanout

- Manage fanout on output pins explicitly
  - Register duplication is not used on partition outputs

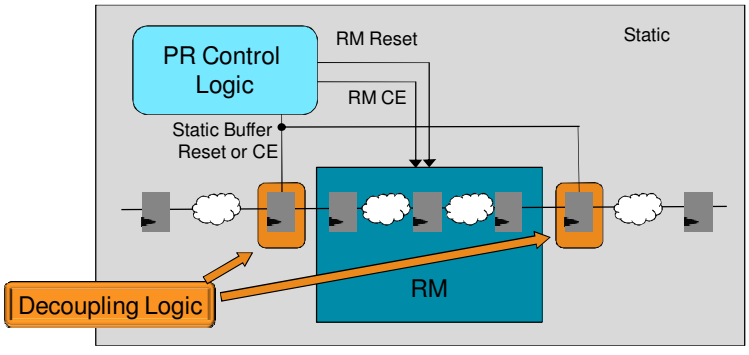


In-coming RM

- Static logic might be “confused” by the loss of signals
  - As in case of a processor system
- Decoupling logic is highly recommended
  - Disconnects the reconfigurable region from the static portion of the design during the act of PR
    - If the reconfigurable element is an output of the FPGA, the decoupling should be performed off chip
- No “Global Set/Reset” function is available for partial reconfiguration
  - Nor is there a “Global 3-State” when an RP contains I/O
  - Responsibility falls to the design
    - Explicitly reset the in-coming RM logic before use

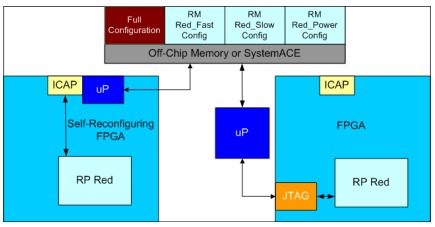
Decoupling and Reset Considerations

- PR control logic (in static partition) can produce enable and reset signals that
  - Clear and disable the incoming buffers from injecting any data into the RP
  - Holds the RM in reset until after the RP has been fully configured
  - Enables the outputs of the RM upon release of the reset



Reconfiguration Control

- A PR design must consider the initiation of Partial Reconfiguration as well as the delivery of partial bit files, either within the FPGA or as part of the system design
  - Mechanism
    - On-chip state machine, processor, or other logic
    - Off-chip microprocessor or other controller
  - Location
    - Flash memory
    - Compact Flash card
    - Network
  - Resource
    - SelectMAP, serial, JTAG configuration ports
    - Internal Configuration Access Port (ICAP)



Outline

- Tools Features and Functionality
- Design Recommendations
- **PR Tool Flow Requirements**
- Clocking Considerations
- Summary

Design Flow Requirements

- Use a top-level UCF to define global constraints for both static and all RPs
  - RM UCF needed only for path exceptions
- Each RM destined to fit in a specific RP must be named identically and posses identical ports
- Static and each RM must be synthesized independently
  - Reconfigurable Modules should be synthesized with I/O insertion disabled
- Store each RM in a separate folder
- Multiple configurations are built to create at least one occurrence of each RM

Design Flow Requirements

- Create RP area group boundaries 5–10% larger than an equivalent flat design
  - Makes it easier to meet timing, routing, and area constraints
  - Utilize the largest as the basis for the area group
- Align frame boundaries to clock regions
  - Plan your layout to avoid multiple RPs within a single frame
    - DRC will flag errors
- When creating bitstreams, use the -b switch to create rawbit files (indicates bit file size)

RM Naming Guidelines

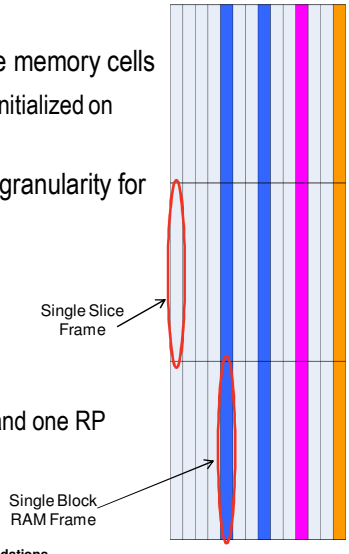
- Naming
  - Each RM destined to fit in a specific RP must be named identically and posses identical ports
    - Even if not all the ports are used in each PR module
    - Identical naming is required so that each RM connects with the static logic properly
  - Each *instantiation* of a given module must have a unique module name

Synthesis Process

- Each reconfigurable module must be independently synthesized (using bottom-up synthesis flow)
  - Static modules can be synthesized together to form one or more netlists
  - Each module can have different synthesis options
- Disable automatic I/O insertion for each module
  - In most cases, the ports will connect to higher level entities
  - I/Os can be instantiated if they are to be reconfigured
- Reconfigurable modules must be stored in separate directories

Reconfigurable Frames

- User-programmable features regulated by volatile memory cells
  - These memory cells (configuration memory) are initialized on power up by a bitstream
- One "reconfigurable frame" represents the finest granularity for partial reconfiguration
  - Frames are homogeneous; they contain only one kind of resource
    - CLBs or block RAMs, for example
  - One clock region tall
  - A "reconfigurable frame" can contain static logic, and one RP
  - Tools within the PlanAhead software help the designer create partitions



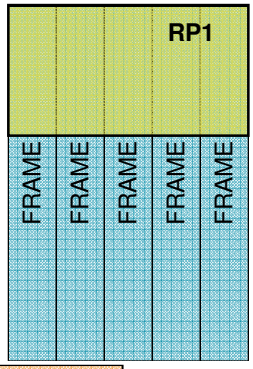
Reconfigurable Regions

- Base reconfigurable frames vary by family and logic type
  - *These are the smallest regions that can be selected for reconfiguration*
  - Bit file sizes for each of these resource types will vary

Family	Slice	BRAM	DSP	IOB
Virtex-6 Family	40 CLB x1 col	8 RAMB36	16	80 (1 bank)
Virtex-5 Family	20 CLB x1 col	4 RAMB36	8	40 (1 bank)
Virtex-4 Family	16 CLB x1 col	4 RAMB16 + 4 FIFO 16	8	32 (1 bank)
Spartan Family	NOT supported			

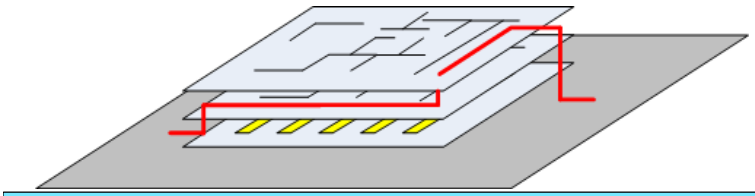
Frames and Area Groups

- Even if a full reconfigurable frame is not completely selected, the entire frame will be reconfigured
  - Static logic that is placed in reconfigurable frames will simply be overwritten with the same data
    - Silicon is designed to guarantee no glitching
  - DRCs will ensure that dynamic logic (SRL, LUTRAM) is not placed in the frames to be reconfigured
  - Static/global routing is free to go anywhere
  - Reconfigurable regions must not overlap each other
    - No two RPs can exist on the same reconfiguration frame



RM Routing Restrictions

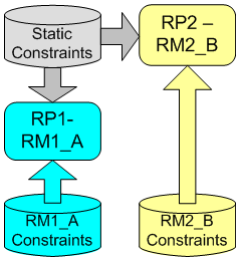
- Static routes are maintained for all RMs
  - RP ranges can have both static and RM routes within them
    - Static routes have precedence with all others being implemented in context
- RM routes must be fully contained within the RP boundary
  - For larger RMs it may be more difficult to route and/or meet timing



Recommendations:
- Make the RP boundary 5–10% larger than an equivalent flat design
- Run Place & Route on the most demanding RM first

General Constraints

- If constraints are valid for all configurations, place in the top-level (static) UCF
  - Full pinout, I/O parameters, global timing, or static logic constraints, for example
  - Static logic constraints are pushed to all the Reconfigurable Modules
- If the constraints are unique to a Reconfigurable Module, create an RM-specific UCF
  - For path exceptions
    - RM constraints cannot be included with the static logic constraints
      - Use separate UCFs for constraints unique to specific RMs
      - The PlanAhead tool manages constraint files on a per-module basis



Module-level Constraints

Recommendations:

- Use top-level UCF to define global constraints
  - Constraints valid for static & RMs
- RM UCF needed only for path exceptions
  - Two options available:
    - Embed constraints in the netlist during synthesis
    - Import UCF when RM is added to the project

Outline

- Tools Features and Functionality
- Design Recommendations
- PR Tool Flow Requirements
- **Clocking Considerations**
- Summary

Global Clocks

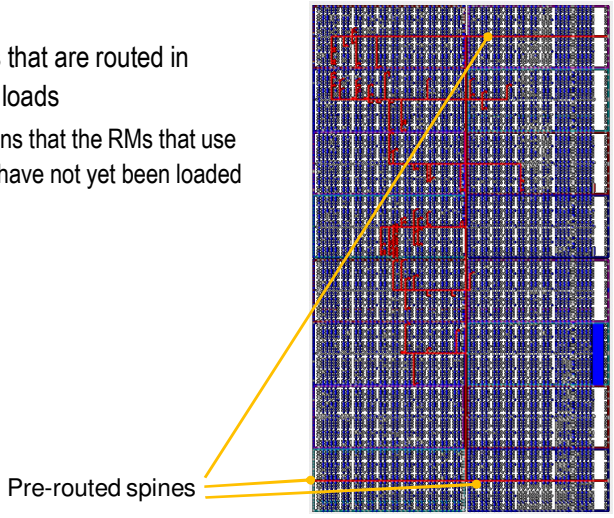
- Each implementation is independent of each other
  - Locations of resources in other RMs are unknown; therefore, clock usage is also unknown for all RMs
- Because the synchronous elements in the other RMs is unknown
  - All global clocks that drive an RM are routed to a clock spine
    - For every clock region
    - Multiple global clocks are driven to where area group ranges overlap
  - Spines are reserved regardless of loads in RMs
    - This limits the number of clock spines for static logic
- Available clock spines per region
  - Virtex®-4 FPGAs have 8
  - Virtex-5 FPGAs have 10
  - Virtex-6 FPGAs have 12

Clock Resource Determination

- All partition pins are derived from the:
  - VHDL entity port list
  - Verilog module list
- All RMs must have identical names and ports
  - Regardless of whether all the ports are used or not

Global Clocks Pre-routing

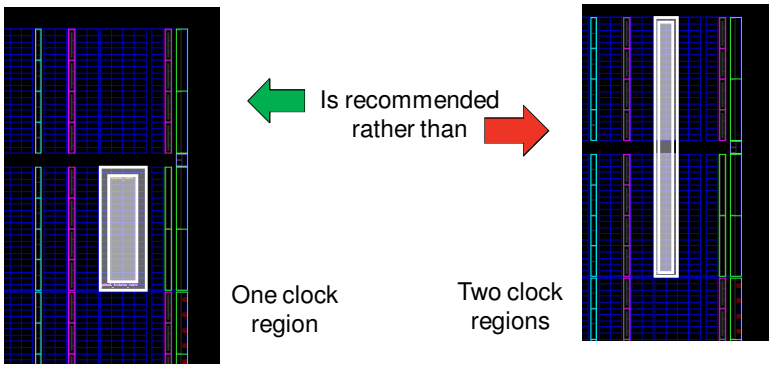
- Note the spines that are routed in regions with no loads
  - This just means that the RMs that use these clocks have not yet been loaded



Global Clocks

Recommendation:

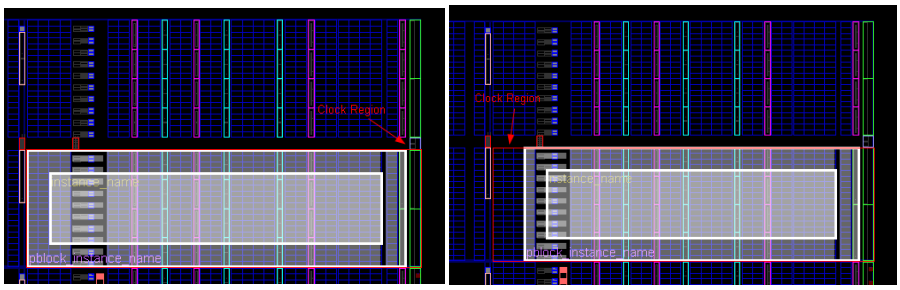
- When possible, add frames to an RP range in the same clock region rather than adding an additional clock region to avoid clock starvation



Global Clocks

Recommendation:

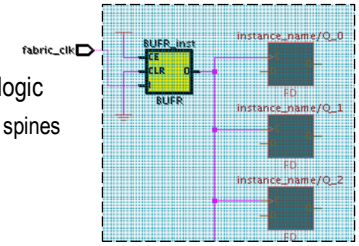
- Static logic will not be placed within RP regions
  - Therefore, for RPs with high clock utilization, occupy an entire clock region



Regional Clocks

Recommendation:

- Completely contain the BUFR and all its loads within one partition (static or an RP) to minimize skew and delay
- If you are crossing partition boundaries, ensure that your regional clock is completely and accurately constrained
- RMs can include BUFRs
  - Some architectures require insertion of proxy logic
    - V4 and V6 have pre-routed regional clocks and spines
    - V5 requires proxy logic insertion



I/O Clocks

- I/O clock spines are not pre-routed
  - Therefore, proxy logic is applied to these spines
- I/O clocks have dedicated routes to I/OLOGICs
  - Hence, they are unroutable after proxy logic is inserted

Requirement:
BUFIOs and all their loads must be in the same partition (static or RP)

Outline

- Tools Features and Functionality
- Design Recommendations
- PR Tool Flow Requirements
- Clocking Considerations
- **Summary**

Summary

- Following the PR design guidelines and recommendations will enable your PR design success
- Partial Reconfiguration is more than just a change in flow
  - It has more stringent design rules
- Follow the PR design recommendations to manage, enhance, and enable the PR flow
  - Utilize recommended design and tool flow techniques
- Follow the PR design flow recommendations for ease of implementation
  - Partial Reconfiguration requires a partitioned, bottom-up compilation approach
    - This prevents optimizations offered by both the synthesis and implementation tools

Summary

- Global clocks must be pre-routed so that every version of every RM and static logic in each clock region is supplied with the necessary clock(s)
  - Clock requirements are determined by the tools as every RM must have identical ports—even if the RM does not use some of those ports
- If more global clocks are required in a given clock region than are available, the design will not route properly
  - Each architecture supports a different number of global clocks per region