ProtoCompiler 試題

Q1: 請列出 ProtoCompiler 流程有哪些步驟? 每個步驟的用途? (20%)

ANS:

1. **Create a database and Compile a Design**
   (1) Create database inside testcase directory
   (2) Run the diagnostic compiler (report rtl_diagnostics) to make sure your RTL is clean
   (3) Compile Design
   (4) Check schematic
2. **Create a Target System for Partition (TSS)**
   (1) Run Pre_partition with tss file and toplevel constraint file
   (2) Check area estimation report、clock summary
   (3) Check unconstrainted、constraint syntax
3. **Partiton the Design**
   (1) Run Partition
   (2) Review pp0 schematic
   (3) Run system route which does trace assignment and hstdm insertion on the partitioned design
4. **Generate result files for each Partition (individual FPGA's)**
   (1) Run system generate with toplevel fdc file as input and the path where the individual projects will be created
   (2) Check the worst slack
5. **Synthesize individual FPGAs**
   (1) Running synthesis on Individual FPGA's
   (2) After synthesis the files needed for PAR are exported to directory 'vivado_hdl'. Run vivado through the following command.
   (3) check the untilization and the timing reports after RnR.
6. **Run P&R and Generate bit files for individual FPGAs**

Q2: Identify design constraints 中可以設定那些 trigger mode, 並且描述這幾種 trigger mode 的差異? (10%)

ANS:

可以設定 Simple triggering、Complex-Counter Triggering、State-Machine Triggering

Complex-Counter Triggering :

- Cycles : captures data after a trigger event occurs
- Events : captures data after multiple trigger events
- PulseWidth : Captures data after successive trigger events
- Watchdog : captures data when the trigger event does not occur for a specified number of cycles

State-Machine Triggering

- More efficient use of buffer by triggering on event sequence such as long idle cycles within a bus protocol
- Programmed dynamically during the debug session
- Optional counter available
- Create complex, design-specific triggers

Q3: TSS 檔案用途是什麼? 需要宣告什麼? (10%)

ANS:
Target system specification is the process of defining the hardware configuration you are going to use for prototyping

The target system specification file is a Tcl file

TSS will basically have the following information

- No of FPGAs in the system
- Connection details for daughter cards, when used
- Board voltage configuration details
- Clock connection
- Connection details between the FPGAs

Q4: PCF 檔案用途是什麼? 需要宣告什麼? (10%)

ANS:

將電路重新分配位置，以達到最佳化。

Needs to define:

1. Net attribute
2. Assign port

Q5: Partition 階段完成後, 我們需要看哪些 report 和確認那些內容? (10%)

ANS:
Check bin usage summary

Check TDM ratio

Check multi-Hop report and global route summary

Check area estimation report、clock summary

Check unconstrainted、constraint syntax

Q6: Partition report 發現有 Multi-Hop 和 cut-clock 時, 我們該如何處理? (10%)

ANS:
1. Multi-Hop optimization:
   partition stage:
   (1) normal effort is default
   (2) high effort -10x increase partition runtimes system route stage:route multi hop paths at lowest tdmm_ratio

2. cut-clock:
   (1) enable automatic clock gate replication
   (2)PCF command for Manual replication

Q7: System route 階段完成後, 我們需要看那些 report 和確認那些內容? (10%)

ANS:
When timing estimation is completed, we can check AP655(post-optiminzation routing summary)

Q8: 當 system route 階段, 發現 slack post tdm 是負值, 我們可以怎麼解決? (5%)

ANS:
Step1 : Add some external cables in TSS file
Step2 : Change tdm_control -max_ratio from 8 to 4

Q9: System generate 階段完成後, 我們需要看那些 report 和確認那些內容? (5%)

ANS:
Check the worst slack
Ideally, there shouldn’t be any violation related to inter-FPGA paths if Post-Optimization Routing Summary is clean

Q10: 當 Vivado P&R 階段發生 congestion 時, 我們可以怎麼處理? (10%)

ANS:

First, we can check the degree of congestion. We can generate a design congestion report through Tcl commands based on the DCP generated after layout. There may be several reasons：

(1) Too many MUXFs:

Solution:

1.Using modular synthesis techniques, set MUXF_REMAPPING for specific modes.

2. Use the -remap option during the opt_design phase.

3. Set the MUXF_REMAP attribute to true for a specific MUXF

(2) long carry chain

Solution:

1. Use the -remap option during the opt_design phase

2. Set the CARRY_REMAP attribute for a specific MUXF

(3) **too many control sets**

Solution：

1.Set CONTROL_SET_THRESHOLD for specific modes using modular synthesis

2. In the opt_design phase, use -control_set_merge to merge equivalent control sets

3. In the opt_design phase, use merge_equivalent_drivers to merge equivalent control sets, including non-control logic

(4) **Too many LUT integrations**

Solution:

1. Set LUT_COMBINING for specific modes using modular synthesis

2. Set the LUTNM attribute of the LUT to empty