ETIN45 – DSP Design

Laboratory Manual

Lab 1: Pipelining

Mojtaba Mahdavi

Jan. 2018

1. Lab preparations

Read Chapter 1 and the first two pages of Chapter 2 in the *ac_datatypes_uv.pdf* document (downloadable from the course homepage). Make sure you understand the concepts of the datatypes ac_int and ac_fixed. Read pages 41-44 in the *bluebook* (downloadable from the course homepage) to get familiar with the concepts of pipelining and Initiation Interval (II).

NOTE:
You should perform all tasks and also fill Table I. Then, ask the TA to check your results to get approved in this lab.

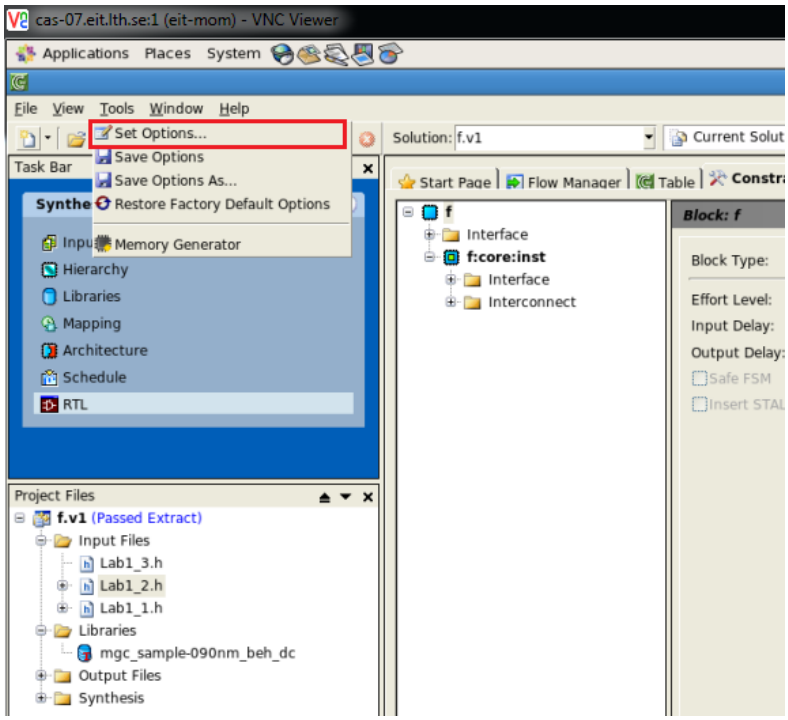
2. Design Walkthrough

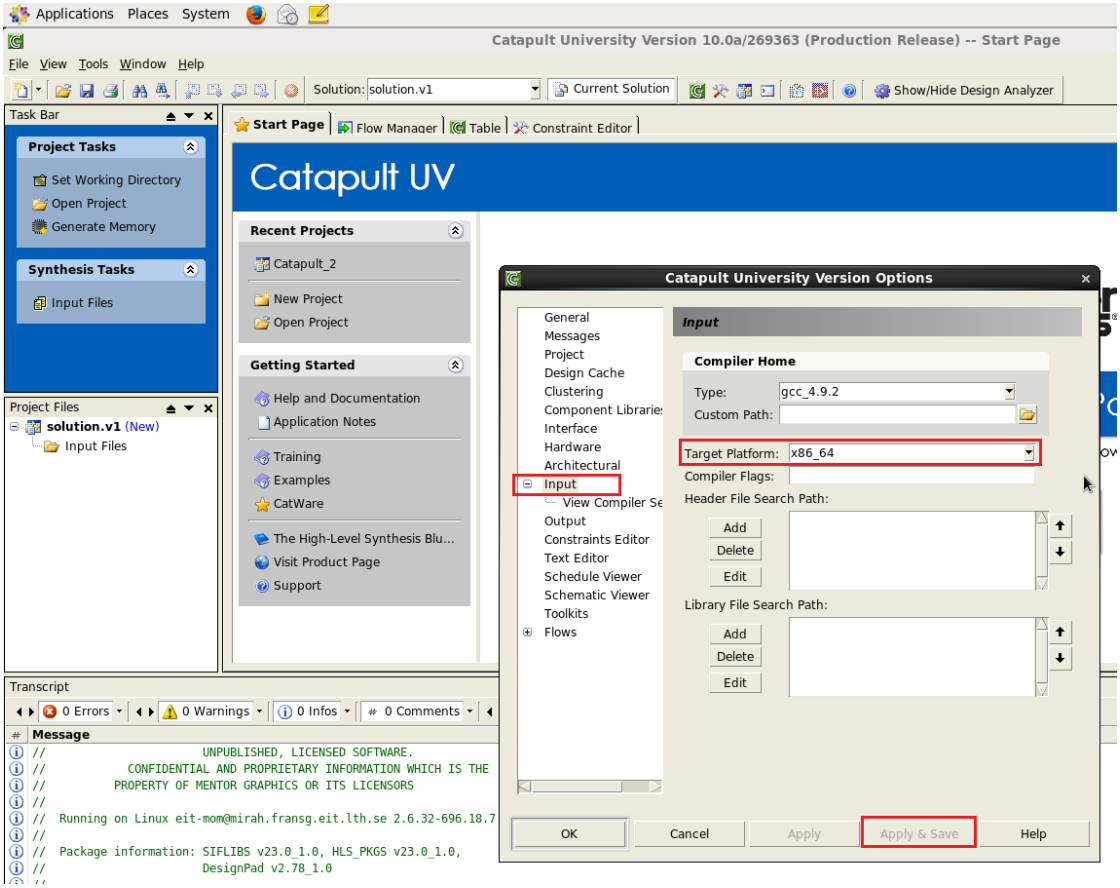
1. Create a directory for lab1.
2. Open a new terminal and run the following command in the command prompt:

```
source /usr/local-eit/cad2/mentor/cat100a/setup.efd
```

3. Invoke Catapult from your lab1 directory by typing “catapult” in the terminal and press enter.

4. **Do the following setting,** as shown in the following figures:


cas-07.eit.lth.se:1 (eit-mom) - VNC Viewer
Applications Places System
File View Tools Window Help
Set Options...
Save Options
Save Options As...
Restore Factory Default Options
Memory Generator
Solution: f.v1
Current Solut
Task Bar
Hierarchy
Libraries
Mapping
Architecture
Schedule
RTL
Start Page
Flow Manager
Table
Interface
f:core:inst
Interface
Interconnect
Block: f
Block Type:
Effort Level:
Input Delay:
Output Delay:
Safe FSM
Project Files
f.v1 (Passed Extract)
Input Files
Lab1_3.h
Lab1_2.h
Lab1_1.h
Libraries
mgc_sample-090nm_beh_dc
Output Files
Synthesis


Applications Places System
Catapult University Version 10.0a/269363 (Production Release) -- Start Page
File View Tools Window Help
Solution: solution.v1
Current Solution
Show/Hide Design Analyzer
Task Bar
Project Tasks
Set Working Directory
Open Project
Generate Memory
Synthesis Tasks
Input Files
Start Page
Flow Manager
Table
Constraint Editor
Catapult UV
Recent Projects
Catapult_2
New Project
Open Project
Getting Started
Help and Documentation
Application Notes
Training
Examples
CatWare
The High-Level Synthesis Blu...
Visit Product Page
Support
Project Files
solution.v1 (New)
Input Files
Catapult University Version Options
General
Messages
Project
Design Cache
Clustering
Component Libraries
Interface
Hardware
Architectural
Input
View Compiler Se
Output
Constraints Editor
Text Editor
Schedule Viewer
Schematic Viewer
Toolkits
Flows
Input
Compiler Home
Type: gcc_4.9.2
Custom Path:
Target Platform: x86_64
Compiler Flags:
Header File Search Path:
Add
Delete
Edit
Library File Search Path:
Add
Delete
Edit
OK
Cancel
Apply
Apply & Save
Help
Transcript
0 Errors
0 Warnings
0 Infos
0 Comments
Message
// UNPUBLISHED, LICENSED SOFTWARE.
// CONFIDENTIAL AND PROPRIETARY INFORMATION WHICH IS THE
// PROPERTY OF MENTOR GRAPHICS OR ITS LICENSORS
//
// Running on Linux eit-mom@mirah.fransg.eit.lth.se 2.6.32-696.18.7
//
// Package information: SIFLIBS v23.0_1.0, HLS_PKGS v23.0_1.0,
// DesignPad v2.78_1.0

5. Choose “File->New->File” from the menu.
6. Type the following code into the file (a function that multiplies three numbers and returns the result):

```
int f(int a, int b, int c){
  return a*b*c;
}
```

7. Save the file as “lab1.h” in the lab1 directory.
8. Choose “Add Input Files” from the Task Bar and choose the lab1.h file.
9. Click the “Hierarchy” icon in the task bar.
10. Select “int” and set it as the Top level of your design. This is to indicate catapult that this will be our top-level design.
11. Click “Apply”.
12. Click the “Libraries” icon in the task bar set the following settings:
    - RTL Synthesis Tool: Design Compiler
    - Technology: 90nm library
    - Compatible Libraries Base ASIC library
13. Click “Apply”.
14. Click the “Mapping” icon in the task bar. Then, select ‘core’ in the Module window and set the clock frequency of the design to 50 MHZ.
15. Click “Apply”.
16. In the task bar window, click on the “Architecture” and select “main”, which is located in the ‘Module’ tab in Constraints Editor (“f->core->main”).
17. Select the pipeline option/checkbox and set the Initial Interval to 1. <explain>
18. Click “Apply”.
19. Click “RTL” in the task bar window to generate the corresponding RTL of your design.

3. Assignment 1 – Design Analysis

There are a number of files and figures that can be inspected to analyze the result. In the following section, you will familiarize with the Catapult GUI and be able to analyze the result in detail.

3.1. Task 1 - Scheduling

Click “Schedule” in the “Synthesis Tasks” to see the scheduled operation of your design. Try to analyze the scheduling view and figure out what the individual elements in the graph represent by comparing the original C++ code and the graph. You can see that the computations are finished in 1 clock cycle which is

the latency of the design. Double click on one of the multipliers marked by "*". What happened?

3.2. Task 2 - Schematic

Analyze the schematic view by opening the schematic in the Project Files as follows:

- f.v1-> Output Files-> Schematics-> RTL

Try to figure out what the individual elements in the schematic represent by comparing the original C++ code and the schematic. Do you see the multipliers? Enter the "f:core" block by double clicking on it. Analyze the result of the "f:core" block. Compare with the original code. Is the result expected?

3.3. Task 3 – Area and Timing report

Analyze the area and timing by opening the RTL report:

- f.v1-> Output Files-> Reports -> RTL

Look in the "Bill Of Materials" section and report the total area? Which HW components are present in the design and what is the area of those? Look in the "Timing Report" section. There are two values (unit is ns) "Max Delay" and "Slack". The "Max Delay" refers to the maximum time for the operation to complete. The "Slack" is the margin to the clock period specified earlier (50Mhz). What is the clock period of the design? Add the values of "Max Delay" and "Slack". Is the result what you expected? Fill in the corresponding cells in Table I.

3.4. Task 4: VHDL code

Analyze the VHDL code by opening the rtl.vhdl file:

- f.v1-> Output Files-> VHDL -> rtl.vhdl

This is the resulting vhdl code that has been generated by the tool. This file is typically used in subsequent synthesis steps by either and ASIC or FPGA tools. Since a tool generates this code, it is not meant for human analysis. Nevertheless, in a small design like the example it should not be too difficult. Try to find the two multipliers in the code by identifying the row in the code where the multiplications are.

4. Assignment 2 – Design Modifications

In this part of the lab you will do some modifications to the design by changing some constraints, such as clock frequency and Initialization Interval (II), and the code. You will re-run the tool with the modified settings and analyze the result similarly to the previous assignment ("Design Analysis").

4.1. Task 1: Pipeline

The concept of pipelining was introduced in the preparation assignments. Change the clock frequency to 100MHz in the "Constraints Editor":

- Click "Mapping".
- Select "core" in the Module tab.
- Click Apply.

Run the tool again (Click on "Generate RTL"). Analyze the result by inspecting different views, logs etc. (like you did previously in the "Design Analysis")? What is the latency of the design (after how many clock cycle is the operation done). Write the corresponding results in Table I.

4.2. Task 2: Initiation Interval

The concept of II was introduced in the preparation assignments. Change the II from II=1 to II=2 in Constraints Editor. Run the tool again (Click on "Generate RTL"). Analyze the result by inspecting different views, logs etc. and write the corresponding results in Table I. (like you did previously in the "Design Analysis")?

4.3. Task 3: Datatypes

Different datatypes were introduced in the preparation assignments. What is the bitwidth of the datatypes in the example?

Your task is to change the datatypes to 8 bit for all parameters (Tip: add `#include ''ac_int.h''` at the top of the file and change all "int" to "ac_int" with proper template parameters).

Set the clock frequency to 50 MHZ. Run the tool for two scenarios: II=1 and II=2. Analyze the result by inspecting different views, logs etc. (like you did previously in the "Design Analysis")? What is the total area and the "Max delay"? Write the corresponding results in Table I.

5. Assignment 3 – Complex Multiplier

In this assignment you will implement complex multiplication and try out different constraints. Use this empty function for your design:

```
void cplx_mult(int &a_re, int &a_im, int &b_re, int
&b_im, int &out_re, int &out_im){
     out_re = 0; //Modify this part of the code
     out_im = 0; //Modify this part of the code
}
```

5.1. Task 1: Analyze with Default Setting

Try to analyze the design with clock frequency of 100 MHZ and II=1 and write the corresponding results in Table I.

5.2. Task 2: Area Minimization

For a given clock speed, 100MHz, try to minimize the design in area. What were your settings and how large was the design? What is the data throughput of your design (the number of complex multiplications per clock cycle)? Fill in Table I with your results.

5.3. Task 3: Throughput Maximization

Now try to optimize the design for throughput – maximize the complex multiplications per clock cycle. Report all requested results in Table I.

Table I. Synthesis results of different runs for design in Lab 1.

	Frequency	II	Latency	Throughput	Max Delay	Area
3.3-Task3						
4.1-Task1						
4.2-Task2						
4.3-Task3		1				
4.3-Task3		2				
5.1-Task1						
5.2-Task2						
5.3-Task3						