ECE 611 – Assignment 2
Spring 2017
(1 week assignment, 100 points)
Due date 2/22/2016
Electrical and Computer Engineering
George Mason University

1. Objective

Observe the impact of unit resizing on the performance.

In this assignment you need to change the size of several processor units and get the IPC to observe the impact of various units with different sizes on the processor performance. The units are listed in table below:

Unit(s)	**Name in the shell script**	**Various size to study**
Fetch width	Core/Fetch/single_limit, Core/Fetch/total_limit	1, 2, 4
Decode width	Core/Decode/total_limit	
Commit width	Core/Commit/single_limit Core/Commit/total_limit	
Int. Phy. Reg. file	Core/Rename/int_rename_regs	32, 64, 96
Float Phy. Reg. file	Core/Rename/float_rename_regs	
ROB size	Thread/reorder_buffer_size	32, 64, 96
L1 Dcache (size & associativity)	Core/DCache/size_kb, Core/DCache/assoc	(16,4), (32,8)
L1 Icache (size & associativity)	Core/ICache/size_kb, Core/ICache/assoc	
Load and store queue size	Core/loadstore_queue_size	16, 32,48
Integer queue size	Core/Queue/int_queue_size	
floating point queue size	Core/Queue/float_queue_size	
L2 cache size	Global/Mem/L2Cache/size_kb 512	256, 512

As shown in the table, there are 6 sets of units. The design space includes 3x3x3x2x3x2 = 324 cases each for art_470, and bzip2_source benchmark. So, the total number of simulations are 648. Fetch, decode and commit widths are

changed together. The same case is applied to register files, caches and load/integer/float Qs.

This is just to find the effect of each of these resources on processor performance.

Please be aware that running the simulations with different L2 cache size (256KB and 512KB) is optional and comes with extra credit.

2- Instruction

1. **Shell generation using the python**. For each configuration and per benchmark you need to create a shell file. Since the number of experiments is large you may want to change the uploaded python script to generate all the cases, otherwise opening files one by one and editing what you need could take forever!
2. **Running the shells**. Run the generated shells.
3. **Extracting IPCs from the output files**. Since there are a large number of output files (results), it is recommended to use a python or etc. programming language to extract the data.

List of Deliverables:

Please include the followings in your report:

1- IPC of the simulated core with minimum and maximum sizes (weakest core and strongest core).

2- IPC graphs.

   a. Draw the IPC graph for a case where **each unit size varies** while the **other unit sizes are *minimum*** (12 graphs).

   b. Draw the IPC graph for a case where **each unit size varies** while the **other unit sizes are *Maximum*** (12 graphs)

3- Result analysis. Investigating the results, try to find the trend as you increase the size of each unit. Explain the performance behavior you observe. Comparing the results, try to find that which unit has more impact on the performance.

4- The script that you used for automation.

5- Simulation output files.

Once completed, submit a ZIPPED file with the following syntax assignmentnumber_first_last.zip where first_last is your first and last name. For instance, file should be named like this → assignment1_Bhoopal_Gunna.zip. Submit the zipped file through the blackboard.

WARNING: Points will be taken off if you do not follow the above instructions