ECE 209- Advanced and Secure Computer Architecture - UCLA
Professor Sehatbakhsh

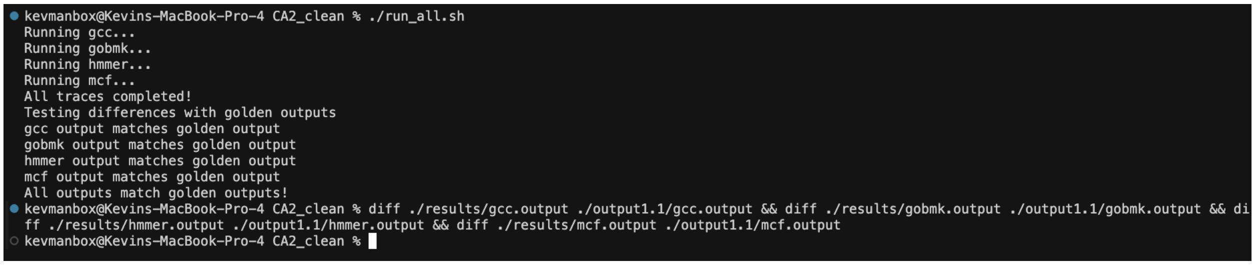
Computer Assignment 2: Backend Out-Of-Order Processor Simulator

Kevin Box

1. Proof of matching outputs

This screenshot in Figure 1 shows that the outputs match. The run_all.sh script can be found in the uploaded code. It runs each trace and saves the outputs in the results directory and then runs a diff for each output against the provided golden outputs. I then manually run the diff for each trace to again show that no differences exist between the two.

Figure 1: Screenshot of differences between simulator and golden outputs.



```
● kevmanbox@Kevins-MacBook-Pro-4 CA2_clean % ./run_all.sh
  Running gcc...
  Running gobmk...
  Running hmmer...
  Running mcf...
  All traces completed!
  Testing differences with golden outputs
  gcc output matches golden output
  gobmk output matches golden output
  hmmer output matches golden output
  mcf output matches golden output
  All outputs match golden outputs!
● kevmanbox@Kevins-MacBook-Pro-4 CA2_clean % diff ./results/gcc.output ./output1.1/gcc.output && diff ./results/gobmk.output ./output1.1/gobmk.output && di
  ff ./results/hmmer.output ./output1.1/hmmer.output && diff ./results/mcf.output ./output1.1/mcf.output
○ kevmanbox@Kevins-MacBook-Pro-4 CA2_clean % █
```

2. E2: Design Space Exploration

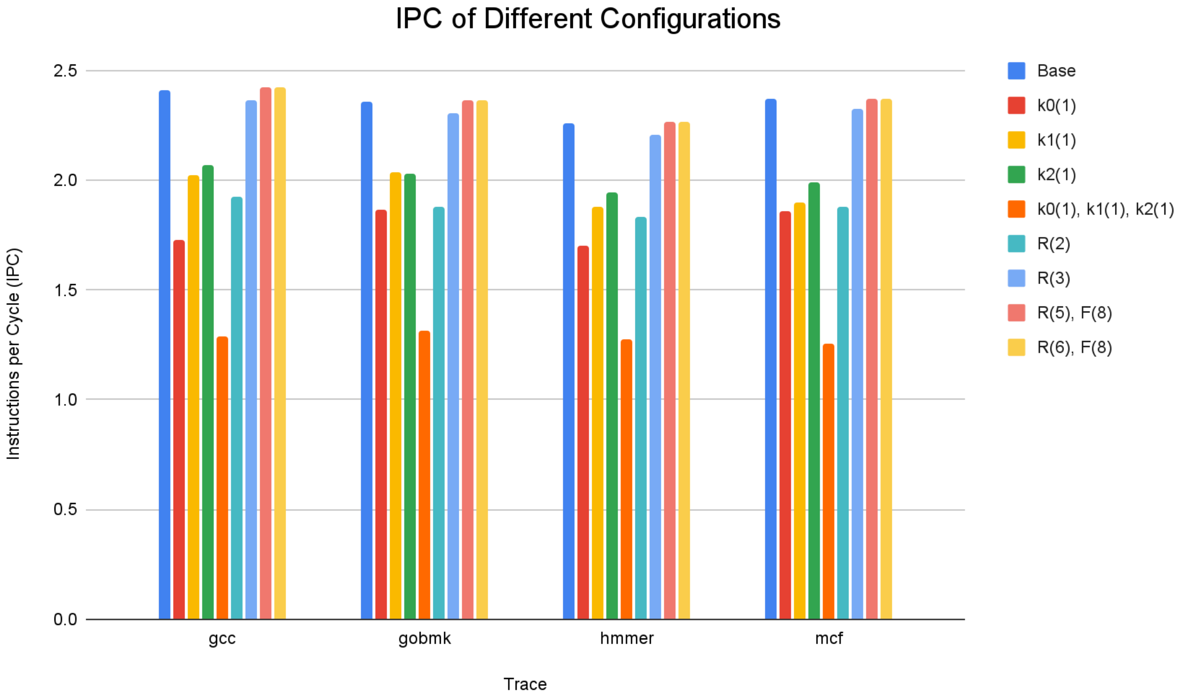
For the design space exploration I ran the run_expirements.sh script that is found in the uploaded code. This script runs many simulations according to the different configurations that one might want to run. It then saves the results based on configuration. From these experiments I took a base configuration that had a: fetch rate = 4, k0 = 2, k1 = 2, k2 = 2, result bus = 4. Then I took selected IPC results with parameters which varied off the base configuration, these selected can be found in table 1. A plot of these results can also be seen in Figure 2.

Table 1: Selected design space exploration results

	gcc	gobmk	hmmer	mcf	average
Base	2.411091	2.361777	2.262546	2.369388	2.3512005
k0(1)	1.727981	1.868565	1.702302	1.857079	1.78898175
k1(1)	2.024947	2.037324	1.879947	1.897785	1.96000075
k2(1)	2.07138	2.029839	1.943181	1.989496	2.008474
k0(1), k1(1), k2(1)	1.28861	1.316846	1.272621	1.258732	1.28420225
R(2)	1.925336	1.881822	1.834055	1.878852	1.88001625
R(3)	2.366752	2.304625	2.206385	2.3245	2.3245
R(5), F(8)	2.42207	2.364457	2.266906	2.369444	2.3245
R(6), F(8)	2.42207	2.364457	2.266906	2.369444	2.3245
R(8), F(8)	2.42207	2.364457	2.266854	2.369444	2.35570625
maximum	2.42207	2.364457	2.266906	2.369444	2.35570625

95% of max	2.3009665	2.24623415	2.1535607	2.2509718	2.237920938

Figure 2: Plot of selected design space exploration results



From the listed results we can see that the configuration with the least hardware (as measured by total number of FUs (k0+k1+k2) and result buses (R)) that gets within >95% of the max is the chosen base configuration with one less result bus, the R(3) configuration. This can be seen in the selected results listed in Table 1 where R(3) performs better than 95% of the max. This is because decreasing the number of result buses from 4 to 3 does not hurt the performance significantly, ~0.05 IPC decrease, however decreasing just one functional unit from 2 to 1 impacts the performance greatly, ~ .4 IPC decrease across all traces and different configurations not seen in selected results. As such I would choose the configuration of fetch rate = 4, k0 = 2, k1 = 2, k2 = 2, result bus = 3 for the given requirements of being within 95% of the max.