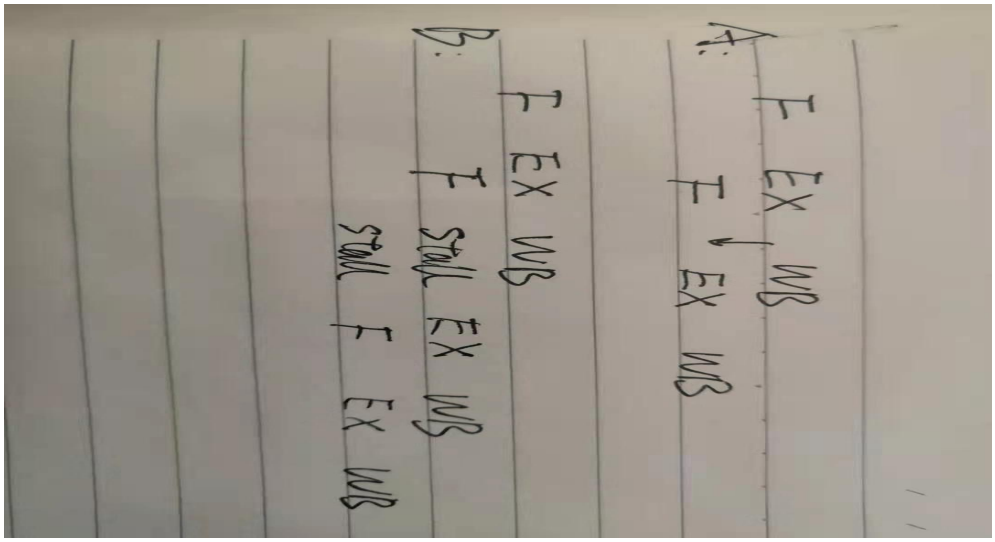
ECEN676_LAB2

tiandi

Feburary 2021

1

I will choose Architecture A because it has higher IPC. When data hazard happens, architecture with forwarding path saves one cycle which would be wasted in architecture with stall path. This can be seen in the graph below.



2

2.1 Which registers account for most of the weight in the tail?

For dependency distance >= 80, the registes that counts most are:
Black-Scholes: rbp
Bodytrack: rbp
Cholesky: r8
Ferret: rsp

FFT: rsi
Fluidanimate: r13

2.2 How to explain this behavior?

To explain this behavior, we first consider the usual use cases for the above registers. For rbp and rsp, they are usually reserved for storing stack frame pointer. Reads and writes on these two happens infrequently, hence longer dependency distance.

2.3 Potential architectural changes?

3

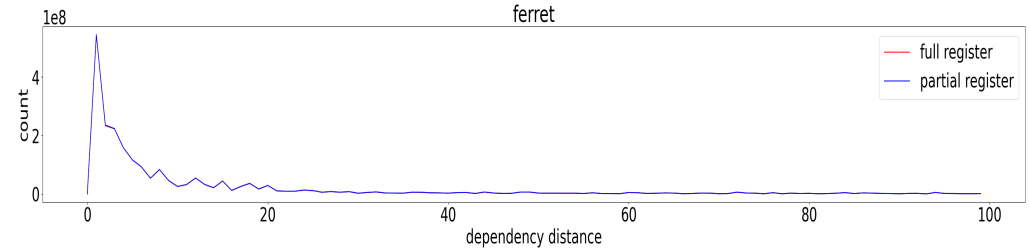
3.1 Explain the trade-off?

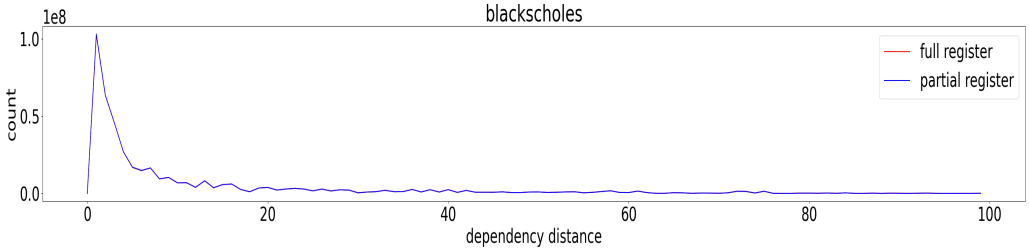
The trade-off lies between simplicity, performance and accuracy. If partial register reads and writes are taken into account, every time a full register read happens, we need to look at the full register writes and all partial registers writes, find the most recent one and calculate the dependency distance. On the other hand, every time a partial register read happens, we need to look at the partial register writes and full register writes.

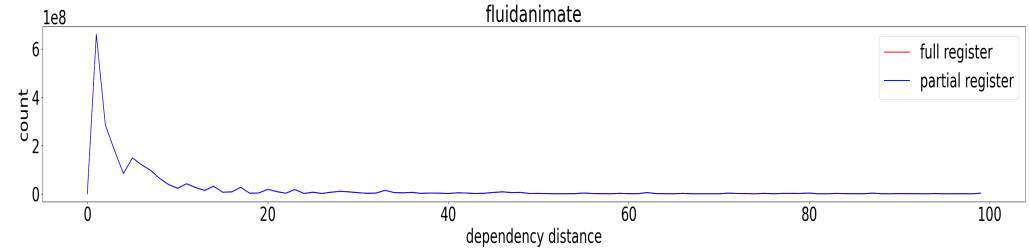
The benefit of taking partial register reads and writes into account is higher accuracy. However, the implementation is more complex(about 30 more lines of code) and the instrumentation runs slower, this is because for each instruction, we need to execute more instrumentation code.

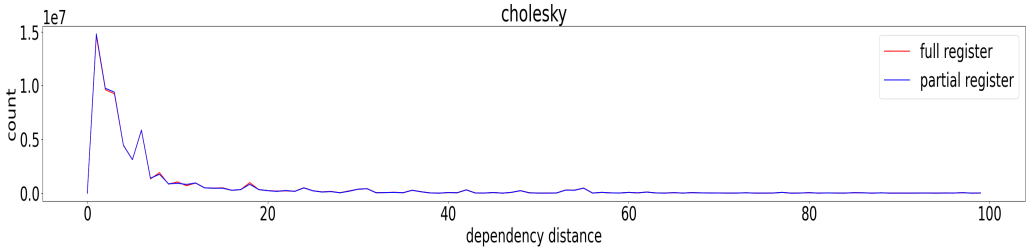
3.2 Simulation results

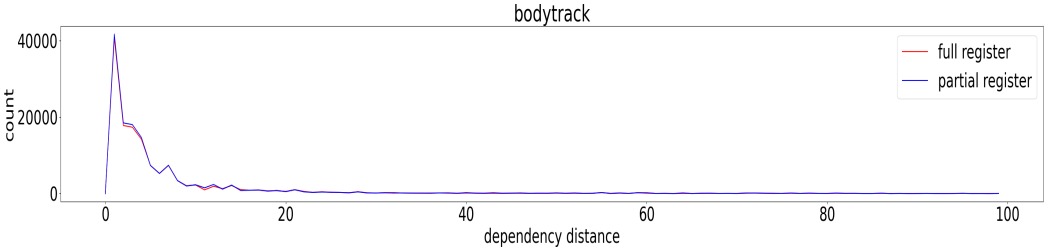
The simulation results of the two dependency distance model are shown below. It can be seen that the difference is small enough to ignore.

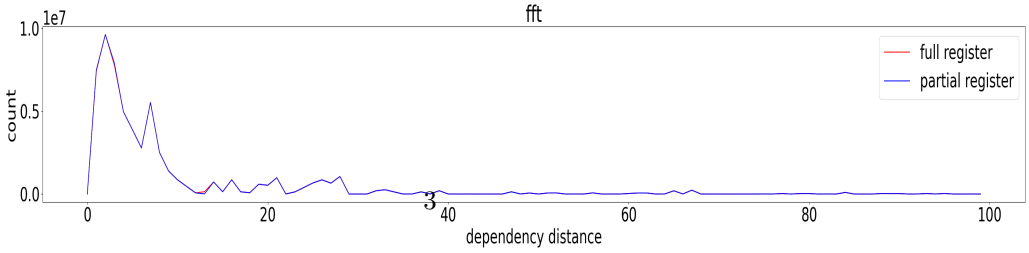

ferret
1e8
count
dependency distance
full register
partial register


blackscholes
1e8
count
dependency distance
full register
partial register


fluidanimate
1e8
count
dependency distance
full register
partial register


cholesky
1e7
count
dependency distance
full register
partial register


bodytrack
count
dependency distance
full register
partial register


fft
1e7
count
dependency distance
full register
partial register

3.3 Is the trade-off acceptable?

Yes, from the figure above it is clear that the difference is small enough to ignore. This might because partial registers are used very little in real practise.

4

Machine A has more general purpose registers. With more GPRs, toward computing the final results, the CPU can carry out more intermediate results which causes longer dependency distance. On the contrary, if less intermediate results can be carried, many GPRs must be reused which causes shorter dependency distance.