

a) Load and branch instructions can forward from the memory stage, and all other instructions can forward from the execute stage. So to simulate forwarding, I simply marked instructions as completed earlier in the pipeline with the following function:

```
void forward(inst_t* i){
    i->donecycle = sim_cycle;
    i->status = DONE;
}
```

In the bottom of the `execute()` function, if the instruction wasn't a load and wasn't a branch instruction, then I forwarded it from the execute stage. In the bottom of the `memory()` function, I forwarded all instructions that reached it.

This method gave me an average speedup of each benchmark of 58.52%. The highest speedup was from Go with 1.75, compared to the lowest which was GCC with a value of 1.48. This means that Go has the most amount of load instructions and GCC has the lowest. The results surprised me because since FPPPP is largely matrix operations, I thought there would be many calculation dependencies.

b) To determine the number of stalls due to loading dependencies, every cycle I checked to see if the current cycle was a load or a branch instruction. If it was, I saved the destination registers and set a `load_last_cycle` flag.

Also every cycle, I checked to see if it wasn't a load or branch instruction and the `load_last_cycle` flag was set. If it was, then I knew last cycle was a load and I checked for dependencies by comparing the saved destination registers to the current source registers.

If there were any dependencies, I incremented a `g_total_dependency_stalls` counter for the number of stalls. To calculate CPI I also used the `sim_num_insn` counter in `sim-safe`.

$$CPI = 1.0 + 1 \times \frac{g_total_dependency_stalls}{sim_num_insn} \quad (1)$$

This method gave me roughly the same values for CPI, but not within 3% as desired.

Benchmark	CPI_{ii}	CPI_{calc}	Difference
Go	1.1898	1.2293	3.21%
GCC	1.1982	1.2116	1.10%
VPR	1.1662	1.2312	5.28%
FPPPP	1.2033	1.3070	7.94%

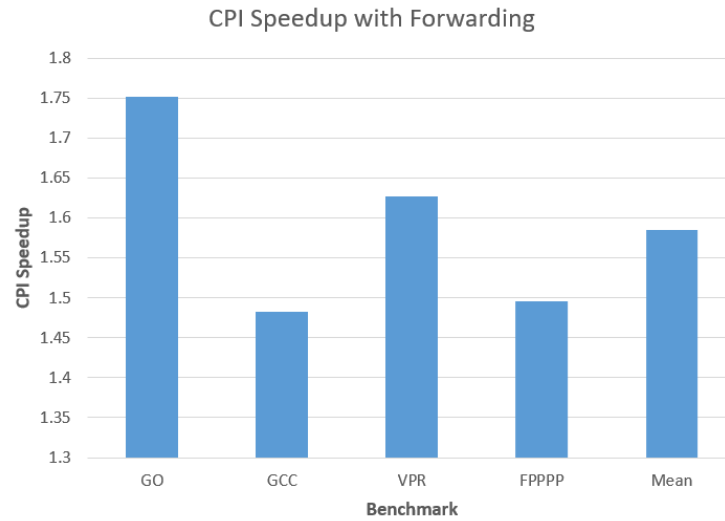


Figure 1: Speedup Due To Forwarding of the Benchmarks

Benchmark	CPI_i	CPI_{ii}	Speedup
Go	2.0835	1.1898	1.7511
GCC	1.7758	1.1982	1.4820
VPR	1.8974	1.1662	1.6269
FPPPP	1.7993	1.2033	1.4953
Geometric Mean			1.5852