ECE 6100, Adv. Computer Architecture

Project 2: Tomasulo Algorithm Pipelined Processor

Check-point 3 Report

Anisha Gartia, GTID: 903136557

A simulator for Tomasulo Algorithm pipelined processor has been designed in this project. We completed design of the vanilla-no frills simulator in checkpoint1. For checkpoint 2, we added two repair schemes to the simulator, viz. Re-order Buffer(ROB), and Checkpoint Repair(CPR).

For checkpoint 3, Experiments for number of FUs for each type =1, 2 and 3 was run. Thus we have 27 combinations for number of FUs of each type as follows:

K0	K1	K2	K0	K1	K2	K0	K1	K2
1	1	1	2	1	1	3	1	1
		2			2			2
		3			3			3
	2	1		2	1		2	1
		2			2			2
		3			3			3
	3	1		3	1		3	1
		2			2			2
		3			3			3

For each of the above combinations, experiments were run for the number of result buses ranging from 1 to total number of FUs. These experiments were conducted for fetch rate of 4 and 8. Repair schemes used were ROB and CPR Thus number of experiments = 162 * 2 * 2 = 648 for each trace file. Exception rate used is 333.

Total number of experiments = 648*4 = 2592.

The above experiments have been run for each of the four traces give viz. mcf.100k, gcc.100k, gobmk.100k and hmmer.100k.

Presented in this report, are the observations for each of the traces, followed by an analysis and reasoning for the observations.

1. Observations for mcf.100k.trace

Maximum Retired IPC = 3.083914

Hardware used to obtain highest retired IPC is as follows:

K0	K1	K2	Number of result buses
3	3	3	9

a. Experiments that resulted in >95% of highest IPC are:

k0	k1	k2	r	f	s	Retired IPC	Total hardware used
3	**3**	**3**	**4**	**4**	**2**	**2.989626**	**13**
3	3	3	4	8	2	3.007217	13
3	3	3	5	4	2	3.048843	14
3	3	3	5	8	2	3.073393	14
3	3	3	6	4	2	3.052007	15
3	3	3	6	8	2	3.081348	15
3	3	3	7	4	2	3.053405	16
3	3	3	7	8	2	3.083629	16
3	3	3	8	4	2	3.053218	17
3	3	3	8	8	2	3.083914	17
3	3	3	9	4	2	3.053218	18
3	3	3	9	8	2	3.083914	18

Thus Retired IPC of (>95%) can be achieved with the lowest hardware as
{k0=3, k1=3, k2=3, result buses = 4}

These were obtained for CPR scheme, and exception rate of 333.

b. Comparison between ROB and CPR repair scheme:

Average rate of retired IPC		Best retired IPC rate	
ROB	CPR	ROB	CPR
1.76607	1.927235	2.642217	3.083914

We observe that with exception rate of 333, **CPR scheme has better performance.**

2. Observations for gcc.100k.trace

Maximum Retired IPC = 3.146336
Hardware used to obtain highest retired IPC is as follows:

K0	K1	K2	Number of result buses
3	3	3	7

a. Experiments that resulted in >95% of highest IPC are:

k0	k1	k2	r	f	s	Retired IPC	Total hardware used
3	**3**	**3**	**4**	**4**	**2**	**3.011776**	**13**
3	3	3	4	8	2	3.024437	13
3	3	3	5	4	2	3.102411	14
3	3	3	5	8	2	3.126857	14
3	3	3	6	4	2	3.117207	15
3	3	3	6	8	2	3.14416	15
3	3	3	7	4	2	3.117207	16
3	3	3	7	8	2	3.146336	16
3	3	3	8	4	2	3.114683	17
3	3	3	8	8	2	3.14169	17
3	3	3	9	4	2	3.114683	18
3	3	3	9	8	2	3.14169	18

Thus Retired IPC of (>95%) can be achieved with the lowest hardware as
{k0=3, k1=3, k2=3, result buses = 4}

These were obtained for CPR scheme, and exception rate of 333.

b. Comparison between ROB and CPR repair scheme:

Average rate of retired IPC		Best retired IPC rate	
ROB	CPR	ROB	CPR
1.751852488	1.953188497	2.663542	3.146336

We observe that with exception rate of 333, **CPR scheme has better performance.**

3. Observations for gobmk.100k.trace

Maximum Retired IPC = 2.820874
Hardware used to obtain highest retired IPC is as follows:

K0	K1	K2	Number of result buses
3	3	3	9

a. Experiments that resulted in >95% of highest IPC are:

k0	k1	k2	r	f	s	Retired IPC	Total hardware used
3	**3**	**3**	**4**	**4**	**2**	**2.735679**	**13**
3	3	3	4	8	2	2.745237	13
3	3	3	5	4	2	2.789867	14
3	3	3	5	8	2	2.805049	14
3	3	3	6	4	2	2.802298	15
3	3	3	6	8	2	2.820477	15
3	3	3	7	4	2	2.801277	16
3	3	3	7	8	2	2.819046	16
3	3	3	8	4	2	2.803633	17
3	3	3	8	8	2	2.820715	17
3	3	3	9	4	2	2.803162	18
3	3	3	9	8	2	2.820874	18

Thus Retired IPC of (>95%) can be achieved with the lowest hardware as
{k0=3, k1=3, k2=3, result buses = 4}

These were obtained for CPR scheme, and exception rate of 333.

b. Comparison between ROB and CPR repair scheme:

Average rate of retired IPC		Best retired IPC rate	
ROB	CPR	ROB	CPR
1.803281954	1.908598093	2.593428	2.820874

We observe that with exception rate of 333, **CPR scheme has better performance.**

4. Observations for hmmer.100k.trace

Maximum Retired IPC = 2.819171
Hardware used to obtain highest retired IPC is as follows:

K0	K1	K2	Number of result buses
3	3	3	7

a. Experiments that resulted in >95% of highest IPC are:

k0	k1	k2	r	f	s	Retired IPC	Total hardware used
3	**3**	**3**	**4**	**4**	**2**	**2.732913**	**13**
3	3	3	4	8	2	2.74408	13
3	3	3	5	4	2	2.783311	14
3	3	3	5	8	2	2.802578	14
3	3	3	6	4	2	2.797952	15
3	3	3	6	8	2	2.818535	15
3	3	3	7	4	2	2.798735	16
3	3	3	7	8	2	2.819171	16
3	3	3	8	4	2	2.794667	17
3	3	3	8	8	2	2.815203	17
3	3	3	9	4	2	2.794667	18
3	3	3	9	8	2	2.815203	18

Thus Retired IPC of (>95%) can be achieved with the lowest hardware as
{k0=3, k1=3, k2=3, result buses = 4}

These were obtained for CPR scheme, and exception rate of 333.

b. Comparison between ROB and CPR repair scheme:

Average rate of retired IPC		Best retired IPC rate	
ROB	CPR	ROB	CPR
1.69674934	1.834527185	2.443196	2.819171

We observe that with exception rate of 333, **CPR scheme has better performance.**

ANALYSIS

a. Analysis for choice of hardware to obtain retired IPC > 95%:

As we can see from the observations, the general trend is that rate of retired IPC increases with increase in hardware. This can be explained as more number of FUs gives the system more opportunity for parallelism. For example, if there are 5 instructions that require FU k2 in the scheduling queue that are ready to fire, if we have 5 FUs of type k2, all 5 can fire simultaneously in the same cycle. Now, again, if we have 5 result buses, all 5 instructions can potentially retire can retire simultaneously (if there are no other instructions of lower instruction number that have completed execution). Also, with higher hardware capability, it will be pointless to have 5 FUs but just one new instruction available at schedule Q per cycle. Thus we see an increase in IPC with higher fetch rate as well. For traces mcf.100k and

gobmk.100k we obtain highest retired IPC when most hardware is used (i.e. k0=k1=k2= 3, result buses = 9). On the other hand, for traces gcc.100k and hmmer.100k we observe highest retired IPC for k0=k1=k2 = 3, result buses = 7. This can be explained as the case we mentioned above where the number of instructions available for retirement is lesser than the number of result buses. Thus, beyond this even if we increase number of result buses, we observe constant retired IPC rate.

b. Analysis for choice of repair scheme:

We observe a better average as well as maximum rate of retired IPC for CPR as compared to ROB. This can be reasoned as ROB in a way stalls instructions that are ready to retire if there is an instruction that was fetched before it that has not been completed. In CPR on the other hand, instructions retire as soon as they are deleted from the scheduling queue. And this deletion from scheduling queue can happen out of order. It is only stored to backup 2 in order. Thus the setup for exception doesn't interfere in Tomasulo's original algorithm at all. Hence this gives us a better rate of retired instructions per cycle.