EECS 470 F25 Midterm Review Session Worksheet

Peppa Pig's Power, Performance, and Pipelining Problems (PPPPPP):

The 5-stage in-order pipeline is modified such that branches are resolved in the WB stage instead of the MEM stage. Is this a safe modification? Why or why not.

Qualcomm is working to improve the energy-efficiency of its next-generation mobile processor. The baseline processor consumes 12W. They are considering adding a new low-power mode that shuts off 75% of the on chip caches. The low power mode incurs a 10% performance hit, but reduces power consumption to 8W. Should they add the new low power mode? Why or why not?

Consider a program where 25% of its execution is serial and the remainder is "embarrassingly parallel" (i.e., its performance scales linearly in the number of cores for an arbitrary number of cores). The performance of the serial portion of the program is directly proportional to memory access latencies. What is the speedup if parallelized across 8 cores?

P6 Problems:

From where can an instruction receive its source register values (Hint: there are 3 places)? What about in R10K?

Bradley's #1 Fan Club is implementing a P6 architecture for their 470 final project. They want to write to the ARF on complete rather than commit. Is this a good idea? Why or why not?

Given the following state of a P6-style machine, walk the machine forward three cycles. Assume that all operations have a 1 cycle latency, and there are no structural hazards on functional units. The machine is 2-way superscalar. Values of ROB #0 and #2 are broadcasted on the CDB on cycle 3. The tail of the ROB points to the next free entry

There are two more instructions that need to be dispatched: Inst 4: R4 = R2 + R1 Inst 5: R4 = R4 + R4

After cycle 2 (initial state)

ROB							
ht	Entry #	Inst #	R	V	S	X	C
h	0	0	R1	[R1]	1	2	
	1	1	R2				
	2	2	R3	[R3]	1	2	
	3	3	R1				
t	4						

Map Table	
Reg	T
R1	ROB #3
R2	ROB #1
R3	ROB #2
R4	

Reservation Station						
Entry #	Inst #	T	T1	T2	V1	V2
1	3	ROB #3	ROB #0	ROB #2		
2	1	ROB #1	ROB #0			[R4]
3						

CDB	
T	V

After cycle 3:

ROB							
ht	Entry #	Inst #	R	V	S	X	C
	0						
	1						
	2						
	3						
	4						

Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station						
Entry #	Inst #	T	T1	T2	V1	V2
1						
2						
3						

CDB	
T	V

After cycle 4:

ROB							
ht	Entry #	Inst #	R	V	S	X	C
	0						
	1						
	2						
	3						
	4						

Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station						
Entry #	Inst #	T	T1	T2	V1	V2
1						
2						
3						

CDB	
T	V

After cycle 5:

ROB							
ht	Entry #	Inst #	R	V	S	X	C
	0						
	1						
	2						
	3						
	4						

Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station						
Entry #	Inst #	T	T1	T2	V1	V2
1						
2						
3						

CDB	
T	V

R10K Problems:

When is the register file written to?

When can a physical register be freed? Why can it be freed then?

How are branch mispredictions handled (assume no early branch resolution/fast branch recovery)?

Given the following state of a R10K-style machine, walk the machine forward four cycles. The machine is 2-way superscalar. There is one add functional unit with a 1 cycle latency, and one fully pipelined multiplier with a 2-cycle latency. The ROB has been prepopulated with the instructions to be executed. There are no further instructions. Feel free to cross out/replace entries in the given tables. Start at cycle 1.

After cycle 1:

ROB							
ht	Entry #	Instruction	T	Told	S	X	C
ht	1	R1 = R1 + R2					
	2	R2 = R3 * R4					
	3	R3 = R1 + R1					
	4	R4 = R4 * R4					

Map Table	
Reg	T+
R1	P1+
R2	P2+
R3	P3+
R4	P4+

Arch. Map Table	
Reg	T
R1	P1
R2	P2
R3	P3
R4	P4

Reservation Station				
Entry #	Operation	T	T1	T2
1				
2				
3				

CDB
T

Free List
P5, P6, P7, P8

After cycle 2:

ROB							
ht	Entry #	Instruction	T	Told	S	X	C
	1	R1 = R1 + R2					
	2	R2 = R3 * R4					
	3	R3 = R1 + R1					
	4	R4 = R4 * R4					

Map Table	
Reg	T+
R1	
R2	
R3	
R4	

Arch. Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station				
Entry #	Operation	T	T1	T2
1				
2				
3				

CDB
T

Free List

After cycle 3:

ROB							
ht	Entry #	Instruction	T	Told	S	X	C
	1	R1 = R1 + R2					
	2	R2 = R3 * R4					
	3	R3 = R1 + R1					
	4	R4 = R4 * R4					

Map Table	
Reg	T+
R1	
R2	
R3	
R4	

Arch. Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station				
Entry #	Operation	T	T1	T2
1				
2				
3				

CDB
T

Free List

After cycle 4:

ROB							
ht	Entry #	Instruction	T	Told	S	X	C
	1	R1 = R1 + R2					
	2	R2 = R3 * R4					
	3	R3 = R1 + R1					
	4	R4 = R4 * R4					

Map Table	
Reg	T+
R1	
R2	
R3	
R4	

Arch. Map Table	
Reg	T
R1	
R2	
R3	
R4	

Reservation Station				
Entry #	Operation	T	T1	T2
1				
2				
3				

CDB
T

Free List