Problem Out-of-Order Execution

1 Question A

	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	3	4	FMUL.D	r0	f0	f1
I_2	0	4	9	10	FADD.D	r1	f2	r0

2 Question B

	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	3	4	FMUL.D	r0	f0	f1
I_2	0	4	9	10	FADD.D	r1	f2	r0
I_3	5	10	15	16	FADD.D	r0	r1	f0

3 Question C

	Enter ROB	Issue	WB	Commit	OP	Dest	Src1	Src2
I_1	-1	0	2	3	FLD	r0	x1	-
I_2	0	1	3	4	FLD	r1	x1	-
I_3	1	4	7	8	FMUL.D	r2	r0	r1
I_4	2	8	13	14	FADD.D	r3	f2	r2
I_5	4	5	8	15	FLD	r0	x1	-
I_6	5	14	19	20	FADD.D	r1	r3	r0

Problem Virtual Memory & Aliasing Problem

1 Question A

$\frac{2048\times 8}{64} = 256 = 2^8$ entries/page

32GB=2^{35} Byte

$2^{35}/2048 = 2^{24}$ pages

$2^{24} = (2^8)^3$

3 levels page table

2 Question B

Two different cache entries holding data for the same physical address.

$L + b \leq k$

Since use physical address can avoid aliasing problem, and use physical tag iff $L + b \leq k$.

3 Question C

$\frac{256 \times 64}{4096} = 4$

at least 4-associative

Problem Branch Prediction

1 Question A

	Instruction	Counter	Prediction	Actual
i = 0	blt x2 x5, skip 1	01	not taken	taken
	bge x2, x5, skip2	01	not taken	not taken
	bnez x4, loop	01	not taken	taken
i = 1	blt x2 x5, skip 1	10	taken	not taken
	bge x2, x5, skip2	00	not taken	taken
	bnez x4, loop	10	taken	taken
i = 2	blt x2 x5, skip 1	01	not taken	taken
	bge x2, x5, skip2	01	not taken	not taken
	bnez x4, loop	11	taken	taken
i = 3	bnez x4, loop	10	taken	not taken
	bge x2, x5, skip2	00	not taken	taken
	bnez x4, loop	11	taken	not taken

Branch	Accuracy
blt	0%
bge	50%
bnez	50%
overall	33.33%

2 Question B

	Instruction	Global History	Counter 0	Counter 1	Prediction	Actual
i = 0	blt x2 x5, skip 1	0	01	01	not taken	taken
	bge x2, x5, skip2	1	01	01	not taken	not taken
	bnez x4, loop	0	01	01	not taken	taken
i = 1	blt x2 x5, skip 1	1	10	01	not taken	not taken
	bge x2, x5, skip2	0	01	00	not taken	taken
	bnez x4, loop	1	10	01	not taken	taken
i = 2	blt x2 x5, skip 1	1	10	00	not taken	taken
	bge x2, x5, skip2	1	10	00	not taken	not taken
	bnez x4, loop	0	10	10	taken	taken
i = 3	bnez x4, loop	1	10	01	not taken	not taken
	bge x2, x5, skip2	0	10	00	taken	taken
	bnez x4, loop	1	11	10	taken	not taken

Branch	Accuracy
blt	50%
bge	75%
bnez	25%
overall	50%

3 Question C

The second branch get the most benefit from global history.

Since the second branch is highly depends on the result of the first branch.

4 Question D

Since if we use Branch History Table (BHT) only, we still need to get target when ID. Adding Branch Target Buffer (BTB) can enable us to get target when IF, and thus we can brunch when IF, avoiding one cycle stall after ID get target.