Ethan Wong

Midterm MIS1B

1. Increased register file size - reduce spilling
increase register field, decrease others

a) Instruction count could decrease as ~~we have to access memory less with reduced spilling~~ Same

b) CPI could increase in the case of I-type, where it might ~~take more steps to use the immediate.~~ Same

c) CT should decrease

2. New transistor with lower latency

a) IC should stay the same, didn't recompile

b) CPI could decrease as instructions may be faster with the new transistor

c) CT should be same, software changes shouldn't affect CT on hardware side

3. Compiler avoids single complex multicycle
Now uses simple instructions that run in fewer cycles

a) IC increase, more simple instructions to replace complex ones

b) CPI decrease, simpler instructions need fewer cycles

c) CT decrease, could have shorter cycle time for simpler instruction

4.

R type	LW	SW	Branch
7	9	6	5
40%	30%	10%	20%

$(0.4 \times 7) + (0.3 \times 9) + (0.1 \times 6) + (0.2 \times 5) = 7.1$ CPI

5 One multiplier for odd bits, one for even bits

a) 9 nanoseconds for each iteration of the multiplication unit (original one)
multiply two 32 bit integers means 31 additions/shifts
$31 \times 9 = 279$ nanoseconds

b) For the new design, we are working in parallel so we effectively "halve" the time of the original multiplier.
We then add 9 nanoseconds to compute the sum.
There will be 2 16-bit multiplications that get added. So effectively it is
$(15 \times 9) + 9 =$

I will just use normal letters,
the greek symbols confuse me
Alpha = A, Beta = B, Gamma = C, Delta = D, Epsilon = E

6. Delay: (K+6)T, where K is the fan-in of the gate.

G0: A0B0
↳ 2 input AND, 8T

P0: A0 ^ B0
↳ 2 input XOR, 8T

G25: $G_E + G_D P_E + G_C P_D P_E + G_B P_C P_D P_E + G_A P_B P_C P_D P_E + C_0 P_A P_B B_C P_D P_E$
↳ 8T (to generate G, P) + 12T (6AND + 12T (6OR) = 32T

P25: $P_A P_B P_C P_D P_E$
↳ 8T + 11T (5AND) = 19T

G_α: The G that govern a 5bit CLA should be the same.
Thus, G_α and G_{25} should be the same as they are both 5bit CLA (even though G25 "bit" is technically another 5bit CLA)

P_α: Same reasoning as G_α. P25 and P_α should be the same.

C25: $G_E + G_D P_E + G_C P_D P_E + G_B P_C P_D P_E + G_A P_B P_C P_D P_E + C_{20} P_A P_B P_C P_D P_E$
↳ 12T + 12T + 19T = 43T

C50: $G_E + G_D P_E + G_C P_D P_E + G_B P_C P_D P_E + G_A P_B P_C P_D P_E + C_{25} P_A P_B P_C P_D P_E$
↳ C25 takes 43T + 12T + 12T = 67T

C45: $G_D + G_C P_D + G_B P_C P_D + G_A P_B P_C P_D + C_{40} P_A P_B P_C P_D$
↳ 11T + 11T + 32T = 54T

C49: $G_{48} + G_{47} P_{48} + G_{46} P_{47} P_{48} + G_{45} P_{46} P_{47} P_{48} + C_{45} P_{45} P_{46} P_{47} P_{48}$
↳ 54T + 11T + 11T = 76T

S49: $(A_{49} \wedge B_{49}) \wedge C_{49}$ =
↳ 76 + 8 = 84T

Max delay is finding S49, 84T

8. 200 billion instructions 3GHz

LW	SW	R-type	Branch	Jump
40%	10%	30%	15%	5%

CPI = 1. (single cycle)

CT = $\frac{1}{3} \times 10^{-9}$

IC = 200 billion

ET = $1 \times (\frac{1}{3} \times 10^{-9}) \times 200 \text{ billion} = 66.67 \text{ s}$

CPI = 1 (still single cycle)

CT = $(\frac{1}{3} \times 10^{-9}) + (0.3 \times 10^{-9})$

← additional 300 pico seconds

IC = 200 billion

25% of loads

25% of 40% → 10%

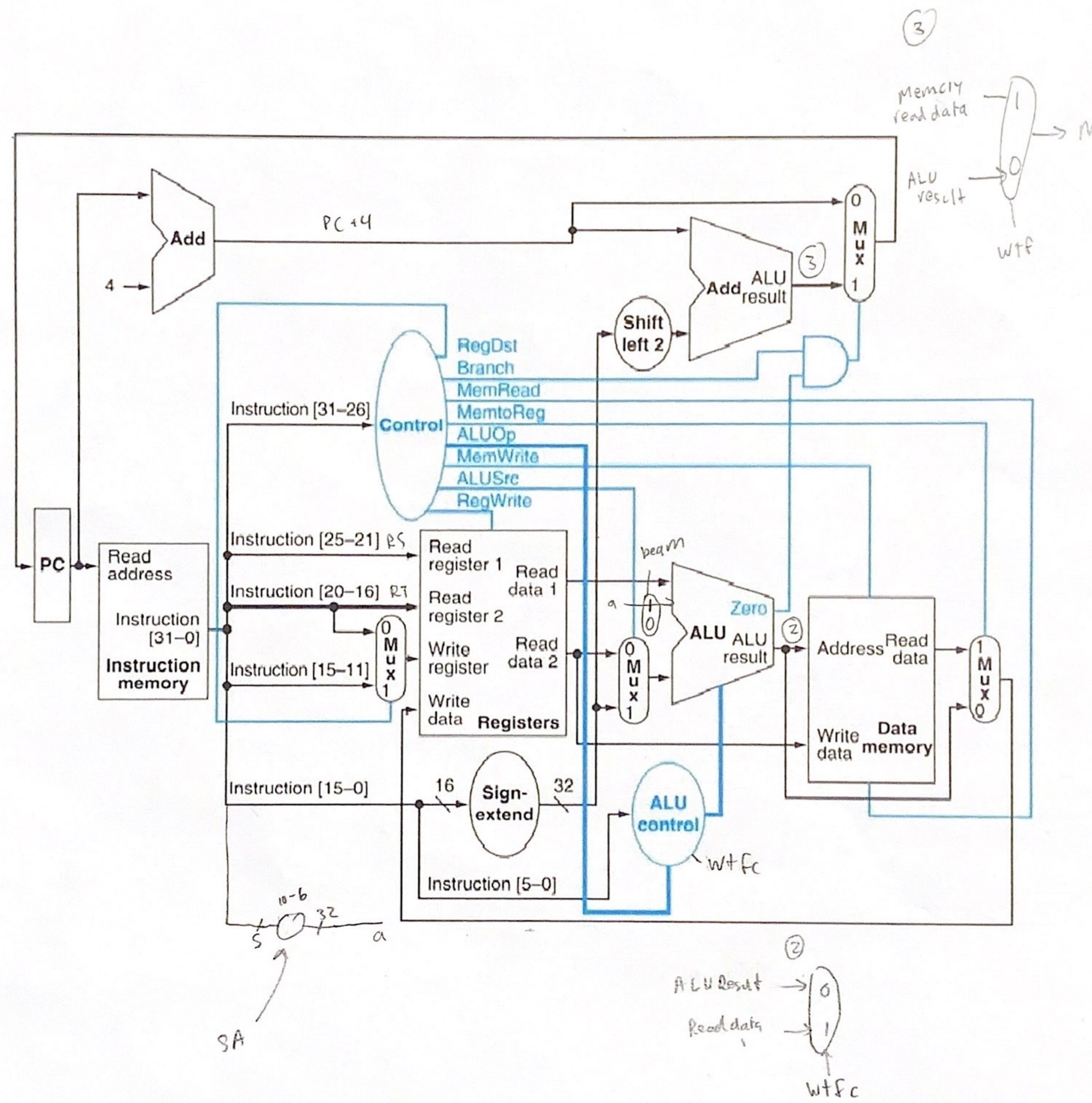
Now we have 90% of original

ET = $1 \times [(\frac{1}{3} \times 10^{-9}) + (0.3 \times 10^{-9})] \times (0.9 \times 200 \text{ billion})$

ET = 114 s

9. a = b Zero = 1

a ≠ b Zero = 0 → invert will be 1


PC
Read address
Instruction [31–0]
Instruction memory
Add
4
PC+4
Instruction [31–26]
Control
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [25–21] RS
Instruction [20–16] RT
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
Shift left 2
Add
ALU result
Mux
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory
beqm
a
10-6
5
32
SA
wtfc
ALU Result
Read data
Memory read data
ALU result
Mux 1
wtf

Main Controller					
Input or Output	Signal Name	R-format	lw	sw	Beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

I can't submit the data path for some reason, I will submit it with the rest of my work

ALU Controller						WTFC
Opcode	**ALUOp**	**instruction**	**function**	**ALU Action**	**ALUCtrl**	
Lw	00	load word	XXXXXX	add	010	0
Sw	00	store word	XXXXXX	add	010	0
Beq	01	branch equal	XXXXXX	subtract	110	0
R-type	10	add	100000	add	010	0
R-type	10	subtract	100010	subtract	110	0
R-type	10	AND	100100	AND	000	0
R-type	10	OR	100101	OR	001	0
R-type	10	SLT	101010	SLT	111	0
R-type	10	WTF	Unique	subtract	110	1