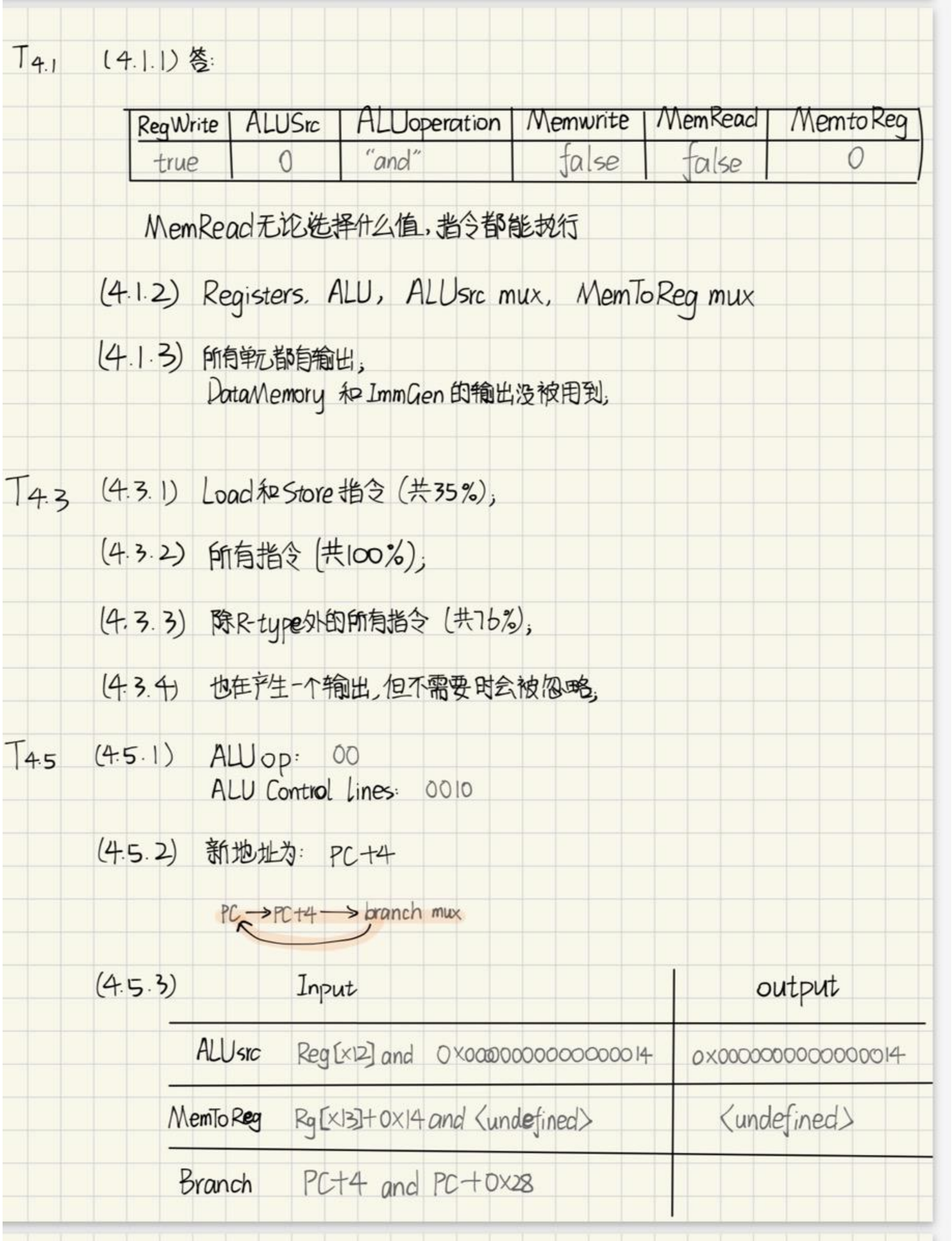


T4.1 (4.1.1) 答:

RegWrite	ALUSrc	ALUoperation	Memwrite	MemRead	MemtoReg
true	0	"and"	false	false	0

MemRead无论选择什么值，指令都能执行

(4.1.2) Registers, ALU, ALUsrc mux, MemToReg mux

(4.1.3) 所有单元都有输出；
DataMemory 和 ImmGen 的输出没被用到；

T4.3 (4.3.1) Load和Store指令（共35%）；

(4.3.2) 所有指令（共100%）；

(4.3.3) 除R-type外的所有指令（共76%）；

(4.3.4) 也在产生一个输出，但不需要时会被忽略；

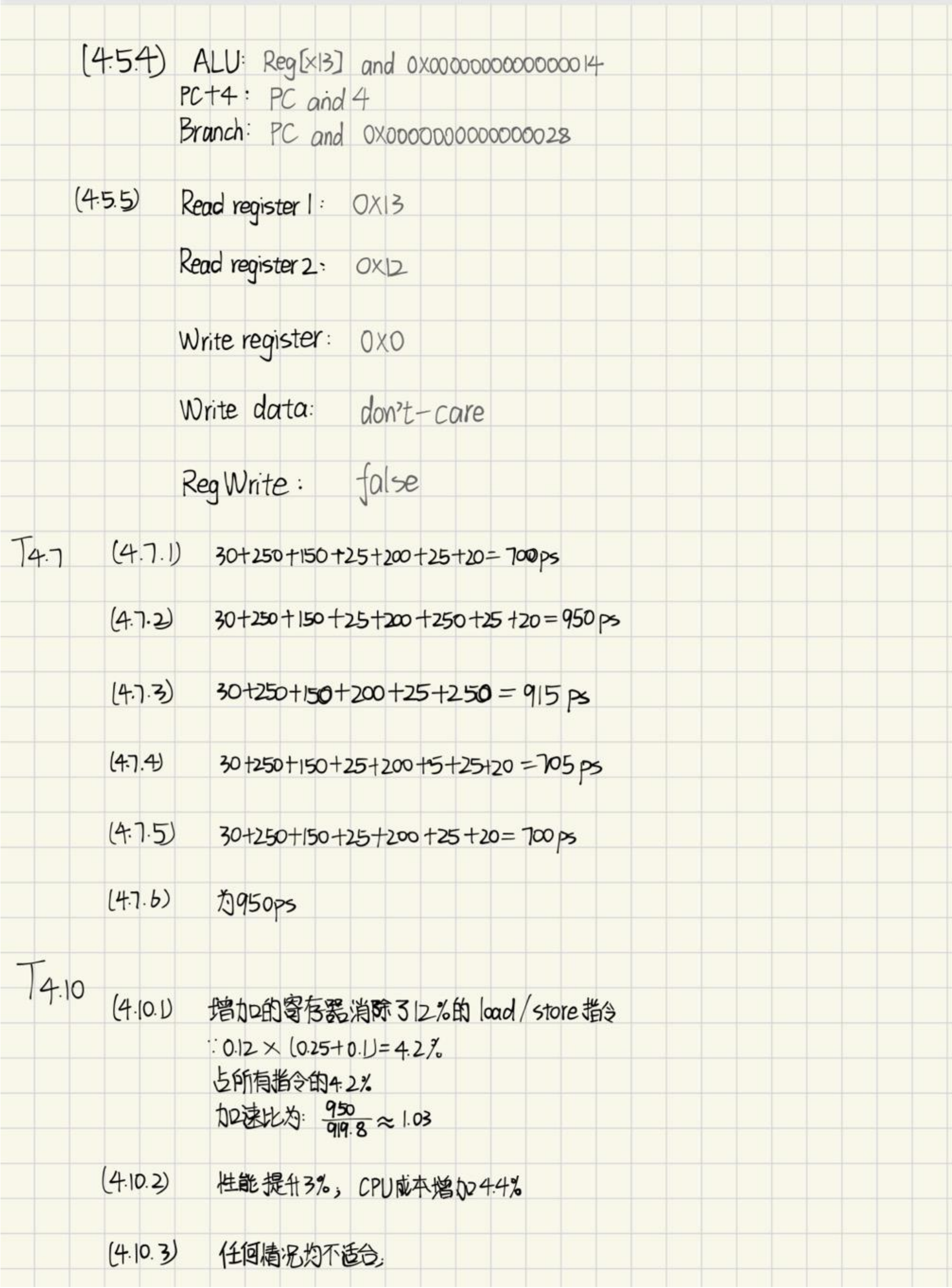
T4.5 (4.5.1) ALU op: 00
ALU Control lines: 0010

(4.5.2) 新地址为: PC+4

$$\begin{tikzcd}
PC \arrow[r] & PC+4 \arrow[r] & \text{branch mux} \arrow[ll, bend left]
\end{tikzcd}$$

(4.5.3)

	Input	output
ALUsrc	Reg[x12] and 0x0000000000000014	0x0000000000000014
MemToReg	Rg[x13]+0x14 and <undefined>	<undefined>
Branch	PC+4 and PC+0x28	



(4.5.4) ALU: Reg[x13] and 0X0000000000000014
PC+4: PC and 4
Branch: PC and 0X0000000000000028

(4.5.5) Read register 1: 0X13

Read register 2: 0X12

Write register: 0X0

Write data: don't-care

RegWrite: false

T4.7 (4.7.1) 30+250+150+25+200+25+20=700ps

(4.7.2) 30+250+150+25+200+250+25+20=950ps

(4.7.3) 30+250+150+200+25+250=915ps

(4.7.4) 30+250+150+25+200+5+25+20=705ps

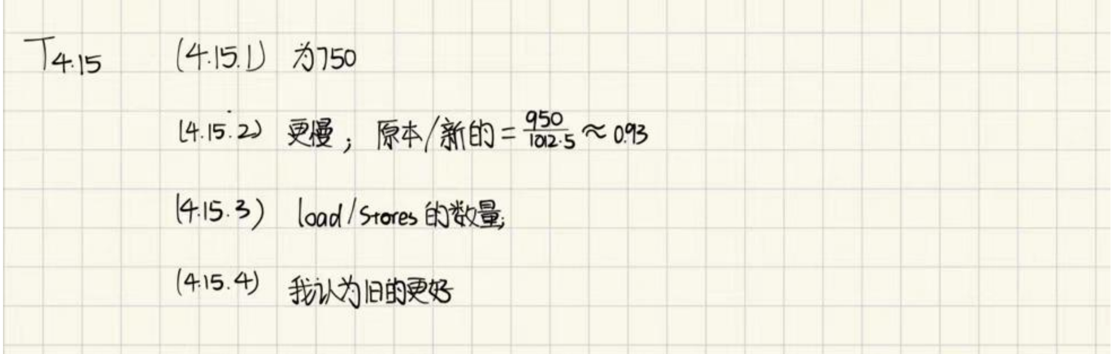
(4.7.5) 30+250+150+25+200+25+20=700ps

(4.7.6) 为950ps

T4.10 (4.10.1) 增加的寄存器消除了12%的load/store指令
∵ 0.12 × (0.25+0.1)=4.2%
占所有指令的4.2%
加速比为: $\frac{950}{919.8} \approx 1.03$

(4.10.2) 性能提升3%，CPU成本增加4.4%

(4.10.3) 任何情况均不适合;



T4.15 (4.15.1) 为750

(4.15.2) 更慢；原本/新的 = $\frac{950}{1012.5} \approx 0.93$

(4.15.3) load/Stores 的数量;

(4.15.4) 我认为旧的更好