1. 解：

答案按照教科书中所述流水线结构予以求解。该流水线有两个特点：（1）数据相关的判断及阻塞是在 ID 级进行的；（2）图中没有画出的前递路径，则认为不存在该类型前递。例如图 5.13 中，RAW 相关在 WB 和 ID 两级流水上产生的冲突不会自动消除。存在的是运算指令 EX 到 EX 的前递，以及 MEM 到 EX 的前递。

```
LW      R1, 0(R0)
LW      R2, 4(R0)
ADD     R3, R1, R2      ;a=b+e
SW      R3, 12(R0)
LW      R4, 8(R0)
ADD     R5, R1, R4      ;c=b+f
SW      R5, 16(R0)
```

	1	2	3	4	5	6	7	8	9
LW R1, 0(R0)	**IF**	**ID**	**EX**	**MEM**	**WB**				
LW R2, 4(R0)		**IF**	**ID**	**EX**	**MEM**	**WB**			
ADD R3, R1, R2			**IF**	**ID**	**ID**	**ID**	**ID**	**EX**	**MEM**
SW R3, 12(R0)				**IF**	**IF**	**IF**	**IF**	**ID**	**EX**
LW R4, 8(R0)								**IF**	**ID**
ADD R5, R1, R4									**IF**
SW R5, 16(R0)									

	10	11	12	13	14	15
LW R1, 0(R0)						
LW R2, 4(R0)						
ADD R3, R1, R2	**WB**					
SW R3, 12(R0)	**MEM**	**WB**				
LW R4, 8(R0)	**EX**	**MEM**	**WB**			
ADD R5, R1, R4	**ID**	**ID**	**EX**	**MEM**	**WB**	
SW R5, 16(R0)	**IF**	**IF**	**ID**	**EX**	**MEM**	**WB**

排序前共需要 15 拍。

重排后的指令序列为：

```
LW      R1, 0(R0)
LW      R2, 4(R0)
LW      R4, 8(R0)
ADD     R5, R1, R4      ;c=b+f（先算后 LW 出来的）
SW      R5, 16(R0)
ADD     R3, R1, R2      ;a=b+e（再算先 LW 出来的）
SW      R3, 12(R0)
```

	1	2	3	4	5	6	7	8	9	10	11	12
LW R1, 0(R0)	IF	ID	EX	MEM	WB							
LW R2, 4(R0)		IF	ID	EX	MEM	WB						
LW R4, 8(R0)			IF	ID	EX	MEM	WB					
ADD R5, R1, R4				IF	ID	ID	EX	MEM	WB			
SW R5, 16(R0)					IF	IF	ID	EX	MEM	WB		
ADD R3, R1, R2							IF	ID	EX	MEM	WB	
SW R3, 12(R0)								IF	ID	EX	MEM	WB

通过指令重排，上述指令序列只需要 12 拍，减少了 3 拍。

利用了存在 MEM 到 EX 的前递，但是没有 WB 到 EX 前递的特点。

2. 解：

MIPS 代码如下：

```
LW      R1, 0(R0)      ; load A
LW      R2, 4(R0)      ; load B
ADD     R3, R1, R2     ; A=A+B
SUB     R4, R3, R2     ; C=A-B
SW      R3, 0(R0)      ; store A
SW      R4, 8(R0)      ; store C
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LW R1, 0(R0)	IF	ID	EX	MEM	WB										
LW R2, 4(R0)		IF	ID	EX	MEM	WB									
ADD R3, R1, R2			IF	ID	ID	ID	ID	EX	MEM	WB					
SUB R4, R3, R2				IF	IF	IF	IF	ID	ID	ID	ID	EX	MEM	WB	
SW R3, 0(R0)								IF	IF	IF	IF	ID	EX	MEM	WB
SW R4, 8(R0)												IF	ID	ID	ID

	16	17	18
LW R1, 0(R0)			
LW R2, 4(R0)			
ADD R3, R1, R2			
SUB R4, R3, R2			
SW R3, 0(R0)			
SW R4, 8(R0)	EX	MEM	WB

3. 解：本题假设采用图 5.15 所示结构

假设此处的浮点运算均为单精度浮点运算。数组的长度为 N(N<2^{15}-1)。

```
        ADDIU   R3, R0, N       ; N
        SLL     R3, R3, 2       ; N*4
```

	ADDU R3, R1, R3	; R3=R1+N*4
Loop:	LWC1 F1, 0(R1)	; load X(i)
	LWC1 F2, 0(R2)	; load Y(i)
	MUL.S F3, F0, F1	; a*X(i)
	ADD.S F4, F2, F3	; a*X(i)+Y(i)
	ADDIU R1, R1, 4	; i++ for X
	ADDIU R2, R2, 4	; i++ for Y
	BNE R1,R3, Loop	
	SWC1 F4, -4(R1)	; store to X(i)

	1	2	3	4	5	6	7	8	9
LWC1 F1, 0(R1)	IF	ID	EX	MEM	WB				
LWC1 F2, 0(R2)		IF	ID	EX	MEM	WB			
MUL.S F3, F0, F1			IF	ID	EX1	EX2	EX3	EX4	MEM
ADD.S F4, F2, F3				IF	ID	ID	ID	ID	EX1
ADDIU R1, R1, 4					IF	IF	IF	IF	ID
ADDIU R2, R2, 4									IF
BNE R1,R3, Loop									
SWC1 F4, -4(R1)									

	10	11	12	13	14	15	16	17
LWC1 F1, 0(R1)								
LWC1 F2, 0(R2)								
MUL.S F3, F0, F1	WB							
ADD.S F4, F2, F3	EX2	EX3	MEM	WB				
ADDIU R1, R1, 4	ID	ID	EX	MEM	WB			
ADDIU R2, R2, 4	IF	IF	ID	EX	MEM	WB		
BNE R1,R3, Loop			IF	ID	EX	MEM	WB	
SWC1 F4, -4(R1)				IF	ID	EX	MEM	WB

4. 解：//对此题中(2)(3)两小问也假设 WB 和 ID 能通过旁路消除相关，与第(1)小问相同。

(1) 无旁路部件，前 99 次循环的流水线时空图如下

	1	2	3	4	5	6	7	8	9	10	11
LD R1, 0(R2)	IF	ID	EX	MEM	WB						
DADDI R1, R1, 4		IF	ID	ID	ID	EX	MEM	WB			
SD R1, 0(R2)			IF	IF	IF	ID	ID	ID	EX	MEM	WB
DADDI R2, R2, 4						IF	IF	IF	ID	EX	MEM
DSUB R4, R3, R2									IF	ID	ID
BNEZ R4, Loop										IF	IF
LD R1, 0(R2)											

	12	13	14	15	16	17	18	19	20	21
LD R1, 0(R2)										
DADDI R1, R1, 4										
SD R1, 0(R2)										
DADDI R2, R2, 4	**WB**									
DSUB R4, R3, R2	**ID**	**EX**	**MEM**	**WB**						
BNEZ R4, Loop	**IF**	**ID**	**ID**	**ID**	**EX**	**MEM**	**WB**			
（延迟槽）										
LD R1, 0(R2)						**IF**	**ID**	**EX**	**MEM**	**WB**

前 99 次循环，每次都是预测错，每个循环 16 个时钟周期。最后一个循环，预测对，但整个循环最后一条指令写回需要 18 个时钟周期。

共计：16×99+18=1602 时钟周期

(2) 有旁路，预测 taken，前 99 次循环流水线时空图如下：

	1	2	3	4	5	6	7	8	9	10	11	12	13
LD R1, 0(R2)	**IF**	**ID**	**EX**	**MEM**	**WB**								
DADDI R1, R1, 4		**IF**	**ID**	**ID**	**EX**	**MEM**	**WB**						
SD R1, 0(R2)			**IF**	**IF**	**ID**	**EX**	**MEM**	**WB**					
DADDI R2, R2, 4					**IF**	**ID**	**EX**	**MEM**	**WB**				
DSUB R4, R3, R2						**IF**	**ID**	**EX**	**MEM**	**WB**			
BNEZ R4, Loop							**IF**	**ID**	**EX**	**MEM**	**WB**		
（延迟槽）													
LD R1, 0(R2)									**IF**	**ID**	**EX**	**MEM**	**WB**

前 99 次循环，执行一个循环需要 8 拍。最后一次循环完整执行完需要 11 拍。

共计：8×99+11=803 时钟周期

(3) 仍认为分支预测 taken

```
Loop:
    LD      R1, 0(R2)
    DADDI   R2, R2, #4
    DSUB    R4, R3, R2
    DADDI   R1, R1, #4
    BNEZ    R4, Loop
    SD      -4(R2), R1
```

	1	2	3	4	5	6	7	8	9	10	11
LD R1, 0(R2)	**IF**	**ID**	**EX**	**MEM**	**WB**						
DADDI R2, R2, 4		**IF**	**ID**	**EX**	**MEM**	**WB**					
DSUB R4, R3, R2			**IF**	**ID**	**EX**	**MEM**	**WB**				
DADDI R1, R1, 4				**IF**	**ID**	**EX**	**MEM**	**WB**			
BNEZ R4, Loop					**IF**	**ID**	**EX**	**MEM**	**WB**		
SD -4(R2), R1						**IF**	**ID**	**EX**	**MEM**	**WB**	
LD R1, 0(R2)							**IF**	**ID**	**EX**	**MEM**	**WB**

前 99 次循环，执行一个循环需要 6 拍。最后一次循环完整执行完需要 10 拍。

整个循环共计：6×99+10=604 时钟周期

5. 解：

空操作指令(nop 指令)，其不改变程序可见寄存器、状态寄存器以及内存的状态，以及用于等待需要一定周期执行的操作。nop 指令的作用，常见的有：取指的强制访存对齐（memory alignment），防止相关风险（hazard），以及用于填充延迟槽（branch delay slot）。

6. 解：

7. 解：

(1) ALU 模块

```
module alu_module
(
    input  [ 3:0] aluop,
    input  [31:0] in1,
    input  [31:0] in2,
    output [31:0] out
);

wire slt_res;
assign slt_res = (in1[31] && !in2[31])   & 1'b1 |
                 (in1[31] && in2[31] )   & !(in1<in2) |
                 (!in1[31] &&  !in2[31]) & (in1<in2) |
                 (!in1[31] && in2[31])   & 1'b0;

assign out =
  {32{aluop==4'b0000}} & (in1+in2) |
  {32{aluop==4'b0001}} & (in1-in2) |
  {32{aluop==4'b0010}} & {31'b0, slt_res} |
  {32{aluop==4'b0011}} & {31'b0, in1<in2} |
  {32{aluop==4'b0100}} & (in1&in2) |
  {32{aluop==4'b0101}} & (in1|in2) |
  {32{aluop==4'b0110}} & (in1^in2) |
  {32{aluop==4'b0111}} & ~(in1|in2) |
  {32{aluop==4'b1000}} & (in1<<in2) |
  {32{aluop==4'b1010}} & (in1>>in2) |
  {32{aluop==4'b1011}} & (in1>>>in2); //verilog2001 中算术右移的新表达方式

endmodule
```