```
main:
daddui r1,r0,0          FDEMW 5
daddui r2,r0,100         FDEMW 1
loop:
l.d f1,v1(r1)             FDEMW 1
l.d f2,v2(r1)              FDEMW 1
add.d f7,f1,f2              FDSEEMW 3
l.d f3,v3(r1)                FSDEsMW 1
div.d f8,f7,f3                 FDSSddddddddMW 9
l.d f4,v4(r1)                   FSSDEMW 0
l.d f5,v5(r1)                      FDEMW 0
mul.d f9,f4,f5                      FDSmmmmmmmmMW 4
l.d f6,v6(r1)                        FSDEMW 0
div.d f10,f9,f6                        FDSSSSSSddddddddMW 8
add.d f11,f8,f10                        FSSSSSSDSSSSSSSEEMW 2
s.d f11,v7(r1)                                 FSSSSSSSDESMW 1
daddui r1,r1,8                                         FDSEMW 1
daddi r2,r2,-1                                          FSDEMW 1
bnez r2,loop                                              FSDEMW 2
Halt                                                        Fxxxx 1

5 + 1 + (1+1+3+1+9+4+8+2+1+1+1+2+1) * 100 = 3506
```

Computer Architectures -- example

Question 2

Considering the same loop-based program, and assuming the following processor architecture for a superscalar MIPS64 processor implemented with multiple-issue and speculation:

- issue 2 instructions per clock cycle
- jump instructions require 1 issue
- handle 2 instructions commit per clock cycle
- timing facts for the following separate functional units:
    i. 1 Memory address 1 clock cycle
    ii. 1 Integer ALU 1 clock cycle
    iii. 1 Jump unit 1 clock cycle
    iv. 1 FP multiplier unit, which is pipelined: 8 stages
    v. 1 FP divider unit, which is not pipelined: 8 clock cycles
    vi. 1 FP Arithmetic unit, which is pipelined: 2 stages
- Branch prediction is always correct
- There are no cache misses
- There are 2 CDB (Common Data Bus).

- Complete the table reported below showing the processor behavior for the 2 initial iterations.
-

# iteration		Issue	EXE	MEM	CDB x2	COMMIT x2
1	l.d f1,v1(r1)	1	2M	3	4	5
1	l.d f2,v2(r1)	1	3M	4	5	6
1	add.d f7,f1,f2	2	6A		8	9
1	l.d f3,v3(r1)	2	4M	5	6	9
1	div.d f8,f7,f3	3	9D		17	18
1	l.d f4,v4(r1)	3	5M	6	7	18
1	l.d f5,v5(r1)	4	6M	7	8	19
1	mul.d f9,f4,f5	4	9X		17	19
1	l.d f6,v6(r1)	5	7M	8	9	20
1	div.d f10,f9,f6	5	18D		26	27
1	add.d f11,f8,f10	6	27A		29	30
1	s.d f11,v7(r1)	6	8M			30
1	daddui r1,r1,8	7	8I		9	31
1	daddi r2,r2,-1	7	9I		10	31
1	bnez r2,loop	8	11J			32
2	l.d f1,v1(r1)	9	10M	11	12	32
2	l.d f2,v2(r1)	9	11M	12	13	33
2	add.d f7,f1,f2	10	14A		16	33
2	l.d f3,v3(r1)	10	12M	13	14	34
2	div.d f8,f7,f3	11	17D			
2	l.d f4,v4(r1)	11				
2	l.d f5,v5(r1)	12				
2	mul.d f9,f4,f5	12				
2	l.d f6,v6(r1)	13				
2	div.d f10,f9,f6	13				
2	add.d f11,f8,f10	14				
2	s.d f11,v7(r1)	14				
2	daddui r1,r1,8	15				
2	daddi r2,r2,-1	15				
2	bnez r2,loop	16				

ASS FUCKING

Computer Architectures -- example

Question 2

Considering the same loop-based program, and assuming the following processor architecture for a superscalar MIPS64 processor implemented with multiple-issue and speculation:

- issue 2 instructions per clock cycle
- jump instructions require 1 issue
- handle 2 instructions commit per clock cycle
- timing facts for the following separate functional units:
  - i. 1 Memory address 1 clock cycle
  - ii. 1 Integer ALU 1 clock cycle
  - iii. 1 Jump unit 1 clock cycle
  - iv. 1 FP multiplier unit, which is pipelined: 8 stages
  - v. 1 FP divider unit, which is not pipelined: 8 clock cycles
  - vi. 1 FP Arithmetic unit, which is pipelined: 2 stages
- Branch prediction is always correct
- There are no cache misses
- There are 2 CDB (Common Data Bus).

○ Complete the table reported below showing the processor behavior for the 2 initial iterations.

○

# iteration		Issue	EXE	MEM	CDB x2	COMMIT x2
1	l.d f1,v1(r1)	1	2M	3	4	5
1	l.d f2,v2(r1)	1	3M	4	5	6
1	add.d f7,f1,f2	2	6A		8	9
1	l.d f3,v3(r1)	2	4M	5	6	9
1	div.d f8,f7,f3	3	9D		17	18
1	l.d f4,v4(r1)	3	5M	6	7	18
1	l.d f5,v5(r1)	4	6M	7	8	19
1	mul.d f9,f4,f5	4	9X		17	19
1	l.d f6,v6(r1)	5	7M	8	9	20
1	div.d f10,f9,f6	5	25D		33	34
1	add.d f11,f8,f10	6	34A		36	37
1	s.d f11,v7(r1)	6	8M			37
1	daddui r1,r1,8	7	8I		9	38
1	daddi r2,r2,-1	7	9I		10	38
1	bnez r2,loop	8	11J			39
2	l.d f1,v1(r1)	9	10M	11	12	39
2	l.d f2,v2(r1)	9	11M	12	13	40
2	add.d f7,f1,f2	10	14A		16	40
2	l.d f3,v3(r1)	10	12M	13	14	41
2	div.d f8,f7,f3	11	17D		25	41
2	l.d f4,v4(r1)	11	13M	14	15	42
2	l.d f5,v5(r1)	12	14M	15	16	42
2	mul.d f9,f4,f5	12	17X		25	43
2	l.d f6,v6(r1)	13	15M	16	18	43
2	div.d f10,f9,f6	13	33D		41	44
2	add.d f11,f8,f10	14	42A		44	45
2	s.d f11,v7(r1)	14	16M			45
2	daddui r1,r1,8	15	16I		18	46
2	daddi r2,r2,-1	15	18I		19	46
2	bnez r2,loop	16	20			47