Decode			Write			Cycle		
I1	I2							1
I3	I4	I1	I2					2
I3	I4	I1						3
	I4			I3		I1	I2	4
I5	I6			I4				5
	I6		15			I3	I4	6
			I6					7
						I5	I6	8

(a) In-order issue and in-order completion

Decode			Write			Cycle		
I1	I2							1
I3	I4	I1	I2					2
	I4	I1		I 3		I2		3
I5	I6			I4		I1	I3	4
	I6		15			I 4		5
			I6			I5		6
						I6		7

(b) In-order issue and out-of-order completion

Decode		Window	Execute			W	rite	Cycle	
I1	I2								1
I3	I4		<i>I1,I2</i>	I1	I2				2
I5	I6		<i>I3,I4</i>	I1		I3	I2		3
			<i>I4,I5,I6</i>		I6	I4	I1	I3	4
			<i>I5</i>		I5		I4	I6	5
							I5		6

(c) Out-of-order issue and out-of-order completion

Figure 14.4 Superscalar Instruction Issue and Completion Policies

```
if (a > 0)
a = a + b + c + d + e;
else
a = a - b - c - d - e;
```

(a) C code

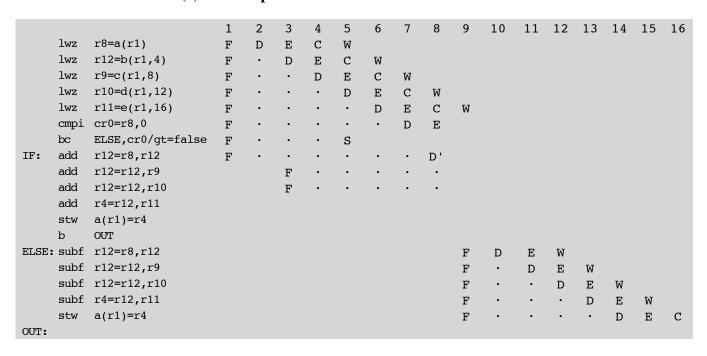
```
#r1 points to a,
                                 #r1+4 points to b,
                                 #r1+8 points to c,
                                 #r1+12 points to d,
                                 #r1+16 points to e.
                                 #load a
      lwz
             r8=a(r1)
             r12=b(r1,4)
                                 #load b
      lwz
                                 #load c
      lwz
             r9=c(r1,8)
      lwz
             r10=d(r1,12)
                                 #load d
             r11=e(r1,16)
                                 #load e
      lwz
                                 #compare immediate
      cmpi
             cr0=r8,0
                                 #branch if bit false
      bc
             ELSE,cr0/gt=false
IF:
                                 #add
      add
             r12=r8,r12
             r12=r12,r9
                                 #add
      add
                                 #add
      add
             r12=r12,r10
             r4=r12,r11
                                 #add
      add
             a(r1)=r4
                                 #store
      stw
             OUT
                                 #unconditional branch
ELSE:
      subf
             r12=r12,r8
                                 #subtract
             r12=r9,r12
      subf
                                 #subtract
             r12=r10,r12
                                 #subtract
      subf
      subf
             r4=r12,r11
                                 #subtract
             a(r1)=r4
                                 #store
      stw
OUT:
```

(b) Assembly code

Figure 14.13 Code Example with Conditional Branch [WEIS94]

```
2
                                       3
                                                5
                                                         7
                                                                      10
                                                                           11 12 13 14 15
                                                                                                 16
         r8=a(r1)
     lwz
                                   D
                                       Е
                                           С
                                                W
     lwz r12=b(r1,4)
                              F
                                                С
                                       D
                                           Е
                                                    W
          r9=c(r1,8)
     lwz
                              F
                                           D
                                                Е
                                                    С
                                                         W
     lwz r10=d(r1,12)
                              F
                                                D
                                                         С
     lwz r11=e(r1,16)
                                                    D
                                                         Ε
                                                             С
                                                                  W
     cmpi cr0=r8,0
                              F
     bc
          ELSE,cr0/gt=false F
                                                S
IF:
     add r12=r8,r12
     add r12=r12,r9
                                       F
                                                                  D
                                                                      Е
                                                                           \overline{\mathsf{W}}
     add r12=r12,r10
                                       F
                                                                           Ε
                                                                                W
          r4=r12,r11
     add
                                                                  F
                                                                           D
                                                                                Ε
                                                                                    W
     stw
         a(r1)=r4
                                                                                D
                                                                                    Ε
                                                                                         C
     b
          OUT
ELSE: subf r12=r8,r12
     subf r12=r12,r9
     subf r12=r12,r10
     subf r4=r12,r11
     stw a(r1)=r4
OUT:
```

(a) Correct prediction: Branch was not taken



(b) Incorrect prediction: Branch was taken

F = fetch C = cache access D = dispatch/decode W = writeback E = execute/address S = dispatch

Figure 14.14 Branch Prediction: Not Taken [WEIS94]