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

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

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

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

Decode		Window	Execute			Write		Cycle
I1	I2							1
I3	I4	*I1,I2*	I1	I2				2
I5	I6	*I3,I4*	I1		I3	I2		3
		I4,I5,I6		I6	I4	I1	I3	4
		I5		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.
      lwz   r8=a(r1)              #load a
      lwz   r12=b(r1,4)           #load b
      lwz   r9=c(r1,8)            #load c
      lwz   r10=d(r1,12)          #load d
      lwz   r11=e(r1,16)          #load e
      cmpi  cr0=r8,0              #compare immediate
      bc    ELSE,cr0/gt=false     #branch if bit false
IF:
      add   r12=r8,r12            #add
      add   r12=r12,r9            #add
      add   r12=r12,r10           #add
      add   r4=r12,r11            #add
      stw   a(r1)=r4              #store
      b     OUT                   #unconditional branch
ELSE:
      subf  r12=r12,r8            #subtract
      subf  r12=r9,r12            #subtract
      subf  r12=r10,r12           #subtract
      subf  r4=r12,r11            #subtract
      stw   a(r1)=r4              #store
OUT:
```

(b) Assembly code

Figure 14.13 Code Example with Conditional Branch [WEIS94]

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	lwz	r8=a(r1)	F	D	E	C	W											
	lwz	r12=b(r1,4)	F	·	D	E	C	W										
	lwz	r9=c(r1,8)	F	·	·	D	E	C	W									
	lwz	r10=d(r1,12)	F	·	·	·	D	E	C	W								
	lwz	r11=e(r1,16)	F	·	·	·	·	D	E	C	W							
	cmpi	cr0=r8,0	F	·	·	·	·	·	D	E								
	bc	ELSE,cr0/gt=false	F	·	·	·	S											
IF:	add	r12=r8,r12	F	·	·	·	·	·	·	D'	E	W						
	add	r12=r12,r9			F	·	·	·	·	·	D	E	W					
	add	r12=r12,r10			F	·	·	·	·	·	·	D	E	W				
	add	r4=r12,r11									F	·	D	E	W			
	stw	a(r1)=r4									F	·	·	D	E	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

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	lwz	r8=a(r1)	F	D	E	C	W											
	lwz	r12=b(r1,4)	F	·	D	E	C	W										
	lwz	r9=c(r1,8)	F	·	·	D	E	C	W									
	lwz	r10=d(r1,12)	F	·	·	·	D	E	C	W								
	lwz	r11=e(r1,16)	F	·	·	·	·	D	E	C	W							
	cmpi	cr0=r8,0	F	·	·	·	·	·	D	E								
	bc	ELSE,cr0/gt=false	F	·	·	·	S											
IF:	add	r12=r8,r12	F	·	·	·	·	·	·	D'								
	add	r12=r12,r9			F	·	·	·	·	·								
	add	r12=r12,r10			F	·	·	·	·	·								
	add	r4=r12,r11																
	stw	a(r1)=r4																
	b	OUT																
ELSE:	subf	r12=r8,r12									F	D	E	W				
	subf	r12=r12,r9									F	·	D	E	W			
	subf	r12=r12,r10									F	·	·	D	E	W		
	subf	r4=r12,r11									F	·	·	·	D	E	W	
	stw	a(r1)=r4									F	·	·	·	·	D	E	C
OUT:																		

(b) Incorrect prediction: Branch was taken

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

Figure 14.14 Branch Prediction: Not Taken [WEIS94]