```
- Module AJupiterImplXJupiter -
 1 [
    We show that AJupiter (specifically, AJupiterExtended) implements XJupiter.
 5 EXTENDS A Jupiter Extended, State Space
    Variables c2ss, s2ss
    varsImpl \stackrel{\triangle}{=} \langle varsEx, c2ss, s2ss \rangle
10 F
     TypeOKImpl \triangleq
11
          \land TypeOKEx
12
          \land \forall c \in Client : IsSS(c2ss[c]) \land IsSS(s2ss[c])
13
14 |
    InitImpl \triangleq
15
          \wedge InitEx
16
          \land c2ss = [c \in Client \mapsto EmptySS]
17
          \land s2ss = [c \in Client \mapsto EmptySS]
18
19 F
    Client c \in Client issues an operation op.
    DoOpImpl(c, op) \triangleq
23
                  cop \stackrel{\Delta}{=} [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq'[c]], ctx \mapsto ds[c]]
24
                \wedge crec' = [crec \text{ EXCEPT } ! [c] = 0]
25
                \wedge cbuf' = [cbuf \ EXCEPT \ ![c] = Append(@, cop)]
26
                \land state' = [state \ EXCEPT \ ![c] = Apply(op, @)]
27
                \land Comm(Msg)! CSend([ack \mapsto crec[c], cop \mapsto cop, oid \mapsto cop.oid])
28
                \land commXJ! CSend(cop)
29
                \wedge c2ss' = [c2ss \text{ except } ![c] =
30
                                    @ \oplus [node \mapsto \{ds'[c]\},\]
31
                                           edge \mapsto \{[from \mapsto ds[c], to \mapsto ds'[c], cop \mapsto cop]\}]
32
33
                \land Unchanged s2ss
34
     DoInsImpl(c) \triangleq
36
         \exists ins \in \{op \in Ins : op.pos \in 1 .. (Len(state[c]) + 1) \land op.ch \in chins \land op.pr = Priority[c]\}:
37
             \land DoOpImpl(c, ins)
38
             \wedge chins' = chins \setminus \{ins.ch\}
39
     DoDelImpl(c) \triangleq
41
         \exists del \in \{op \in Del : op.pos \in 1 .. Len(state[c])\}:
42
             \land DoOpImpl(c, del)
43
             ∧ UNCHANGED chins
44
    DoImpl(c) \triangleq
46
          \wedge DoCtx(c)
47
          \land \lor DoInsImpl(c)
48
             \vee DoDelImpl(c)
49
          \land UNCHANGED \langle sbuf, srec \rangle
```

```
51 F
    RevImpl(c) \triangleq
52
               RevEx(c)
53
               LET m \stackrel{\triangle}{=} Head(cincoming[c])
54
                      cBuf \stackrel{\triangle}{=} cbuf[c]
55
                      cShiftedBuf \stackrel{\Delta}{=} SubSeq(cBuf, m.ack + 1, Len(cBuf))
                       xform \stackrel{\triangle}{=} xFormCopCopsSS(m.cop, cShiftedBuf) [lss, xss]
57
                     c2ss' = [c2ss \text{ EXCEPT } ! [c] = @ \oplus xform.xss]
              UNCHANGED s2ss
59
60
     SRevImpl \triangleq
61
          \land \ SRevEx
62
          \land \ \mathtt{LET} \ m \ \triangleq \\
                           Head(sincoming)
63
                   c \triangleq ClientOf(m.cop)
64
                   cBuf \triangleq sbuf[c]
65
                   cShiftedBuf \stackrel{\triangle}{=} SubSeq(cBuf, m.ack + 1, Len(cBuf))
66
                    xform \stackrel{\Delta}{=} xFormCopCopsSS(m.cop, cShiftedBuf) [lss, xss]
67
                   s2ss' = [cl \in Client \mapsto
68
                                   If cl = c then s2ss[cl] \oplus xform.xss else s2ss[cl] \oplus xform.lss]
69
70
          \land Unchanged c2ss
71
     NextImpl \triangleq
72
          \lor \exists c \in Client : DoImpl(c) \lor RevImpl(c)
73
          \vee SRevImpl
74
     FairnessImpl \triangleq
76
         WF_{varsImpl}(SRevImpl \lor \exists c \in Client : RevImpl(c))
77
    SpecImpl \stackrel{\triangle}{=} InitImpl \wedge \Box [NextImpl]_{varsImpl} \wedge FairnessImpl
    XJ \stackrel{\triangle}{=} \text{INSTANCE } XJupiter \text{ WITH}
81
                      cincoming \leftarrow cincoming XJ, sincoming \leftarrow sincoming XJ
82
    THEOREM SpecImpl \Rightarrow XJ!Spec
     \* Modification History
     * Last modified Sun Dec 30 16:48:09 CST 2018 by hengxin
     \* Created Sat Dec~29~18:36:51~CST~2018 by hengxin
```