

```

1  ┌────────────────── MODULE AJupiterExtended ───────────────────┐
    AJupiter extended with JupiterCtx. This is used to show that AJupiter implements XJupiter.
6  EXTENDS JupiterCtx
7  └──────────────────┐
8  VARIABLES cbuf, crec, sbuf, srec
10 vars  $\triangleq$   $\langle \textit{intVars}, \textit{ctxVars}, \textit{cbuf}, \textit{crec}, \textit{sbuf}, \textit{srec} \rangle$ 
12 Msg  $\triangleq$  [ack : Int, cop : Cop, oid : Oid]
13 └──────────────────┐
14 TypeOK  $\triangleq$ 
15    $\wedge$  TypeOKInt
16    $\wedge$  TypeOKCtx
17    $\wedge$  Comm(Msg)! TypeOK
18    $\wedge$  crec  $\in$  [Client  $\rightarrow$  Int]
19    $\wedge$  srec  $\in$  [Client  $\rightarrow$  Int]
20    $\wedge$  cbuf  $\in$  [Client  $\rightarrow$  Seq(Cop)]
21    $\wedge$  sbuf  $\in$  [Client  $\rightarrow$  Seq(Cop)]
22 └──────────────────┐
23 Init  $\triangleq$ 
24    $\wedge$  InitInt
25    $\wedge$  InitCtx
26    $\wedge$  Comm(Msg)! Init
27    $\wedge$  crec = [c  $\in$  Client  $\mapsto$  0]
28    $\wedge$  srec = [c  $\in$  Client  $\mapsto$  0]
29    $\wedge$  cbuf = [c  $\in$  Client  $\mapsto$   $\langle \rangle$ ]
30    $\wedge$  sbuf = [c  $\in$  Client  $\mapsto$   $\langle \rangle$ ]
31 └──────────────────┐
    Client c  $\in$  Client issues an operation op.
35 DoOp(c, op)  $\triangleq$ 
36   LET cop  $\triangleq$  [op  $\mapsto$  op, oid  $\mapsto$  [c  $\mapsto$  c, seq  $\mapsto$  cseq'[c], ctx  $\mapsto$  ds[c]]
37   IN    $\wedge$  crec' = [crec EXCEPT ![c] = 0]
38        $\wedge$  cbuf' = [cbuf EXCEPT ![c] = Append( $\textcircled{a}$ , cop)]
39        $\wedge$  state' = [state EXCEPT ![c] = Apply(op,  $\textcircled{a}$ )]
40        $\wedge$  Comm(Msg)! CSend([ack  $\mapsto$  crec[c], cop  $\mapsto$  cop, oid  $\mapsto$  cop.oid])
42 DoIns(c)  $\triangleq$ 
43    $\exists$  ins  $\in$  {op  $\in$  Ins : op.pos  $\in$  1 .. (Len(state[c] + 1)  $\wedge$  op.ch  $\in$  chins  $\wedge$  op.pr = Priority[c]} :
44      $\wedge$  DoOp(c, ins)
45      $\wedge$  chins' = chins  $\setminus$  {ins.ch}
47 DoDel(c)  $\triangleq$ 
48    $\exists$  del  $\in$  {op  $\in$  Del : op.pos  $\in$  1 .. Len(state[c])} :
49      $\wedge$  DoOp(c, del)
50      $\wedge$  UNCHANGED chins

```

```

52  $Do(c) \triangleq$ 
53    $\wedge DoCtx(c)$ 
54    $\wedge \vee DoIns(c)$ 
55      $\vee DoDel(c)$ 
56    $\wedge UNCHANGED \langle sbuf, srec \rangle$ 
57 |-----|
    Client  $c \in Client$  receives a message from the Server.
61  $Rev(c) \triangleq$ 
62    $\wedge Comm(Msg)!CRev(c)$ 
63    $\wedge crec' = [crec \text{ EXCEPT } ![c] = @ + 1]$ 
64    $\wedge LET \ m \triangleq Head(cincoming[c])$ 
65      $cBuf \triangleq cbuf[c]$ 
66      $cShiftedBuf \triangleq SubSeq(cBuf, m.ack + 1, Len(cBuf))$ 
67      $xcop \triangleq XformOpOps(COT, m.cop, cShiftedBuf)$ 
68      $xcBuf \triangleq XformOpsOp(COT, cShiftedBuf, m.cop)$ 
69   IN    $\wedge cbuf' = [cbuf \text{ EXCEPT } ![c] = xcBuf]$ 
70      $\wedge state' = [state \text{ EXCEPT } ![c] = Apply(xcop.op, @)]$ 
71    $\wedge RevCtx(c)$ 
72    $\wedge UNCHANGED \langle chins, sbuf, srec \rangle$ 
73 |-----|
    The Server receives a message.
77  $SRev \triangleq$ 
78    $\wedge Comm(Msg)!SRev$ 
79    $\wedge LET \ m \triangleq Head(sincoming)$ 
80      $c \triangleq ClientOf(m.cop)$ 
81      $cBuf \triangleq sbuf[c]$ 
82      $cShiftedBuf \triangleq SubSeq(cBuf, m.ack + 1, Len(cBuf))$ 
83      $xcop \triangleq XformOpOps(COT, m.cop, cShiftedBuf)$ 
84      $xcBuf \triangleq XformOpsOp(COT, cShiftedBuf, m.cop)$ 
85   IN    $\wedge srec' = [cl \in Client \mapsto$ 
86     IF  $cl = c$ 
87     THEN  $srec[cl] + 1$ 
88     ELSE 0]
89      $\wedge sbuf' = [cl \in Client \mapsto$ 
90     IF  $cl = c$ 
91     THEN  $xcBuf$ 
92     ELSE  $Append(sbuf[cl], xcop)]$ 
93      $\wedge state' = [state \text{ EXCEPT } ![Server] = Apply(xcop.op, @)]$ 
94      $\wedge Comm(Msg)!SSend(c, [cl \in Client \mapsto [ack \mapsto srec[cl], cop \mapsto xcop, oid \mapsto xcop.oid]])$ 
95    $\wedge SRevCtx$ 
96    $\wedge UNCHANGED \langle chins, cbuf, crec \rangle$ 
97 |-----|
98  $Next \triangleq$ 
99    $\vee \exists c \in Client : Do(c) \vee Rev(c)$ 

```

100 $\vee SRev$

102 $Fairness \triangleq$ There is no requirement that the clients ever generate operations.

103 $WF_{vars}(SRev \vee \exists c \in Client : Rev(c))$

105 $Spec \triangleq Init \wedge \Box[Next]_{vars} \wedge Fairness$

106 |-----|

107 $QC \triangleq$ Quiescent Consistency

108 $Comm(Msg)!EmptyChannel \Rightarrow Cardinality(Range(state)) = 1$

110 THEOREM $Spec \Rightarrow \Box QC$

111 |-----|

 \ * Modification History

 \ * Last modified Sat Dec 29 17:43:58 CST 2018 by hengxin

 \ * Created Thu Dec 27 21:15:09 CST 2018 by hengxin