```
MODULE AJupiterExtended
```

*AJupiter* extended with *JupiterCtx*. This is used to show that *AJupiter* implements *XJupiter*.

```
EXTENDS JupiterCtx, BufferStateSpace   TODO: To extend AJupiter

VARIABLES cbuf, crec, sbuf, srec, cincomingXJ, sincomingXJ
varsEx ≜ ⟨intVars, ctxVars, cbuf, crec, sbuf, srec, cincomingXJ, sincomingXJ⟩

AJMsgEx ≜ [ack : Nat, cop : Cop, oid : Oid]
commXJ ≜ INSTANCE CSComm WITH Msg ← Seq(Cop),
                              cincoming ← cincomingXJ, sincoming ← sincomingXJ

TypeOKEx ≜
    ∧ TypeOKInt
    ∧ TypeOKCtx
    ∧ commXJ!TypeOK
    ∧ crec ∈ [Client → Nat]
    ∧ srec ∈ [Client → Nat]
    ∧ cbuf ∈ [Client → Seq(Cop)]
    ∧ sbuf ∈ [Client → Seq(Cop)]

InitEx ≜
    ∧ InitInt
    ∧ InitCtx
    ∧ commXJ!Init
    ∧ crec = [c ∈ Client ↦ 0]
    ∧ srec = [c ∈ Client ↦ 0]
    ∧ cbuf = [c ∈ Client ↦ ⟨⟩]
    ∧ sbuf = [c ∈ Client ↦ ⟨⟩]

DoOpEx(c, op) ≜
    LET cop ≜ [op ↦ op, oid ↦ [c ↦ c, seq ↦ cseq[c]], ctx ↦ ds[c]]
    IN  ∧ crec′ = [crec EXCEPT ![c] = 0]
        ∧ cbuf′ = [cbuf EXCEPT ![c] = Append(@, cop)]
        ∧ SetNewAop(c, op)
        ∧ Comm!CSend([ack ↦ crec[c], cop ↦ cop, oid ↦ cop.oid])
        ∧ commXJ!CSend(cop)

ClientPerformEx(c, m) ≜
    LET xform ≜ xFormShift(COT, m.cop, cbuf[c], m.ack + 1)
    IN  ∧ cbuf′ = [cbuf EXCEPT ![c] = xform.xops]
        ∧ crec′ = [crec EXCEPT ![c] = @ + 1]
        ∧ SetNewAop(c, xform.xop.op)

ServerPerformEx(m) ≜
    LET c ≜ ClientOf(m.cop)
```

```
48              xform ≜ xFormShift(COT, m.cop, sbuf[c], m.ack + 1)
49               xcop ≜ xform.xop
50      IN      ∧ srec′ = [cl ∈ Client ↦ IF cl = c THEN srec[cl] + 1 ELSE 0]
51              ∧ sbuf′ = [cl ∈ Client ↦ IF cl = c THEN xform.xops ELSE Append(sbuf[cl], xcop)]
52              ∧ SetNewAop(Server, xcop.op)
53              ∧ Comm!SSend(c, [cl ∈ Client ↦ [ack ↦ srec[cl], cop ↦ xcop, oid ↦ xcop.oid]])
54              ∧ commXJ!SSendSame(c, xcop)
55 ├──────────────────────────────────────────────────────────────────────────┤
56  DoEx(c) ≜
57          ∧ DoInt(DoOpEx, c)
58          ∧ DoCtx(c)
59          ∧ UNCHANGED ⟨sbuf, srec⟩

61  RevEx(c) ≜
62      ∧ RevInt(ClientPerformEx, c)
63      ∧ RevCtx(c)
64      ∧ commXJ!CRev(c)
65      ∧ UNCHANGED ⟨sbuf, srec⟩

67  SRevEx ≜
68      ∧   SRevInt(ServerPerformEx)
69      ∧   SRevCtx
70      ∧   commXJ!SRev
71      ∧   UNCHANGED ⟨cbuf, crec⟩
72 ├──────────────────────────────────────────────────────────────────────────┤
73  NextEx ≜
74      ∨  ∃ c ∈ Client : DoEx(c) ∨ RevEx(c)
75      ∨  SRevEx

77  FairnessEx ≜
78      WF_varsEx(SRevEx ∨ ∃ c ∈ Client : RevEx(c))

80  SpecEx ≜ InitEx ∧ □[NextEx]_varsEx  ∧ FairnessEx
81 ├──────────────────────────────────────────────────────────────────────────┤
82  QC ≜  Quiescent Consistency
83      Comm!EmptyChannel ⇒ Cardinality(Range(state)) = 1

85  THEOREM SpecEx ⇒ □QC
86 └──────────────────────────────────────────────────────────────────────────┘
    \* Modification History
    \* Last modified Sat Jan 12 21:09:03 CST 2019 by hengxin
    \* Created Thu Dec 27 21:15:09 CST 2018 by hengxin
```