```
COMMENT

A "simple" implementation of the Izhikevich neuron.
Equations and parameter values are taken from
  Izhikevich EM (2007).
  "Dynamical systems in neuroscience"
  MIT Press

Equation for synaptic inputs taken from
  Izhikevich EM, Edelman GM (2008).
  "Large-scale model of mammalian thalamocortical systems."
  PNAS 105(9) 3593-3598.

Example usage (in Python):
  from neuron import h
  sec = h.Section(name=sec) # section will be used to calculate v
  izh = h.Izhi2007b(0.5)
  def initiz () : sec.v=-60
  fih=h.FInitializeHandler(initz)
  izh.Iin = 70  # current clamp

Cell types available are based on Izhikevich, 2007 book:
    1. RS - Layer 5 regular spiking pyramidal cell (fig 8.12 from 2007 book)
    2. IB - Layer 5 intrinsically bursting cell (fig 8.19 from 2007 book)
    3. CH - Cat primary visual cortex chattering cell (fig 8.23 from 2007 book)
    4. LTS - Rat barrel cortex Low-threshold  spiking interneuron (fig 8.25 from 2007
book)
    5. FS - Rat visual cortex layer 5 fast-spiking interneuron (fig 8.27 from 2007
book)
    6. TC - Cat dorsal LGN thalamocortical (TC) cell (fig 8.31 from 2007 book)
    7. RTN - Rat reticular thalamic nucleus (RTN) cell  (fig 8.32 from 2007 book)

ENDCOMMENT

: Declare name of object and variables
NEURON {
  POINT_PROCESS Izhi2007b
  RANGE C, k, vr, vt, vpeak, u, a, b, c, d, Iin, celltype, alive, cellid, verbose,
derivtype, delta, t0
  NONSPECIFIC_CURRENT i
}

: Specify units that have physiological interpretations (NB: ms is already declared)
UNITS {
  (mV) = (millivolt)
  (uM) = (micrometer)
}

: Parameters from Izhikevich 2007, MIT Press for regular spiking pyramidal cell
PARAMETER {
  C = 1 : Capacitance
  k = 0.7
  vr = -60 (mV) : Resting membrane potential
  vt = -40 (mV) : Membrane threhsold
  vpeak = 35 (mV) : Peak voltage
  a = 0.03
  b = -2
  c = -50
  d = 100
  Iin = 0
  celltype = 1 : A flag for indicating what kind of cell it is,  used for changing the
dynamics slightly (see list of cell types in initial comment).
  alive = 1 : A flag for deciding whether or not the cell is alive -- if it's dead,
acts normally except it doesn't fire spikes
  cellid = -1 : A parameter for storing the cell ID, if required (useful for
diagnostic information)
}

: Variables used for internal calculations
ASSIGNED {
  v (mV)
  i (nA)
  u (mV) : Slow current/recovery variable
  delta
  t0
  derivtype
}

: Initial conditions
INITIAL {
  u = 0.0
  derivtype=2
  net_send(0,1) : Required for the WATCH statement to be active; v=vr initialization
done there
}

: Define neuron dynamics
BREAKPOINT {
  delta = t-t0 : Find time difference
  if (celltype<5) {
    u = u + delta*a*(b*(v-vr)-u) : Calculate recovery variable
  }
  else {
     : For FS neurons, include nonlinear U(v): U(v) = 0 when v<vb ; U(v) = 0.025(v-vb)
when v>=vb (d=vb=-55)
     if (celltype==5) {
       if (v<d) {
        u = u + delta*a*(0-u)
       }
       else {
        u = u + delta*a*((0.025*(v-d)*(v-d)*(v-d))-u)
       }
     }

     : For TC neurons, reset b
     if (celltype==6) {
       if (v>-65) {b=0}
       else {b=15}
       u = u + delta*a*(b*(v-vr)-u) : Calculate recovery variable
     }

     : For TRN neurons, reset b
     if (celltype==7) {
       if (v>-65) {b=2}
       else {b=10}
       u = u + delta*a*(b*(v-vr)-u) : Calculate recovery variable
     }
  }

  t0=t : Reset last time so delta can be calculated in the next time step
  i = -(k*(v-vr)*(v-vt) - u + Iin)/C/1000
}

FUNCTION derivfunc () {
  if (celltype==5 && derivtype==2) { : For FS neurons, include nonlinear U(v): U(v) = 0 when v<vb ; U(v) = 0.025(v-vb) when v>=vb (d=vb=-55)
    derivfunc = a*(0-u)
  } else if (celltype==5 && derivtype==1) { : For FS neurons, include nonlinear U(v): U(v) = 0 when v<vb ; U(v) = 0.025(v-vb) when v>=vb (d=vb=-55)
    derivfunc = a*((0.025*(v-d)*(v-d)*(v-d))-u)
  } else if (celltype==5) {
    VERBATIM
    hoc_execerror("izhi2007b.mod ERRA: derivtype not set",0);
    ENDVERBATIM
  } else {
    derivfunc = a*(b*(v-vr)-u) : Calculate recovery variable
  }
}

: Input received
NET_RECEIVE (w) {
  : Check if spike occurred
  if (flag == 1) { : Fake event from INITIAL block
    if (celltype == 4) { : LTS cell
      WATCH (v>(vpeak-0.1*u)) 2 : Check if threshold has been crossed, and if so, set flag=2
    } else if (celltype == 6) { : TC cell
      WATCH (v>(vpeak+0.1*u)) 2
    } else { : default for all other types
      WATCH (v>vpeak) 2
    }
    : additional WATCHfulness
    if (celltype==6 || celltype==7) {
      WATCH (v> -65) 3 : change b param
      WATCH (v< -65) 4 : change b param
    }
    if (celltype==5) {
      WATCH (v> d) 3  : going up
      WATCH (v< d) 4  : coming down
    }
    v = vr  : initialization can be done here
  : FLAG 2 Event created by WATCH statement -- threshold crossed for spiking
  } else if (flag == 2) {
    if (alive) {net_event(t)} : Send spike event if the cell is alive
    : For LTS neurons
    if (celltype == 4) {
      v = c+0.04*u : Reset voltage
      if ((u+d)<670) {u=u+d} : Reset recovery variable
      else {u=670}
     }
    : For FS neurons (only update v)
    else if (celltype == 5) {
      v = c : Reset voltage
     }
    : For TC neurons (only update v)
    else if (celltype == 6) {
      v = c-0.1*u : Reset voltage
      u = u+d : Reset recovery variable
     }  else {: For RS, IB and CH neurons, and RTN
      v = c : Reset voltage
      u = u+d : Reset recovery variable
     }
  : FLAG 3 Event created by WATCH statement -- v exceeding set point for param reset
  } else if (flag == 3) {
    : For TC neurons
    if (celltype == 5)        { derivtype = 1 : if (v>d) u'=a*((0.025*(v-d)*(v-d)*(v-
d))-u)
    } else if (celltype == 6) { b=0
    } else if (celltype == 7) { b=2
    }
  : FLAG 4 Event created by WATCH statement -- v dropping below a setpoint for param 
reset
  } else if (flag == 4) {
    if (celltype == 5)        { derivtype = 2  : if (v<d) u==a*(0-u)
    } else if (celltype == 6) { b=15
    } else if (celltype == 7) { b=10
    }
  }
}
```