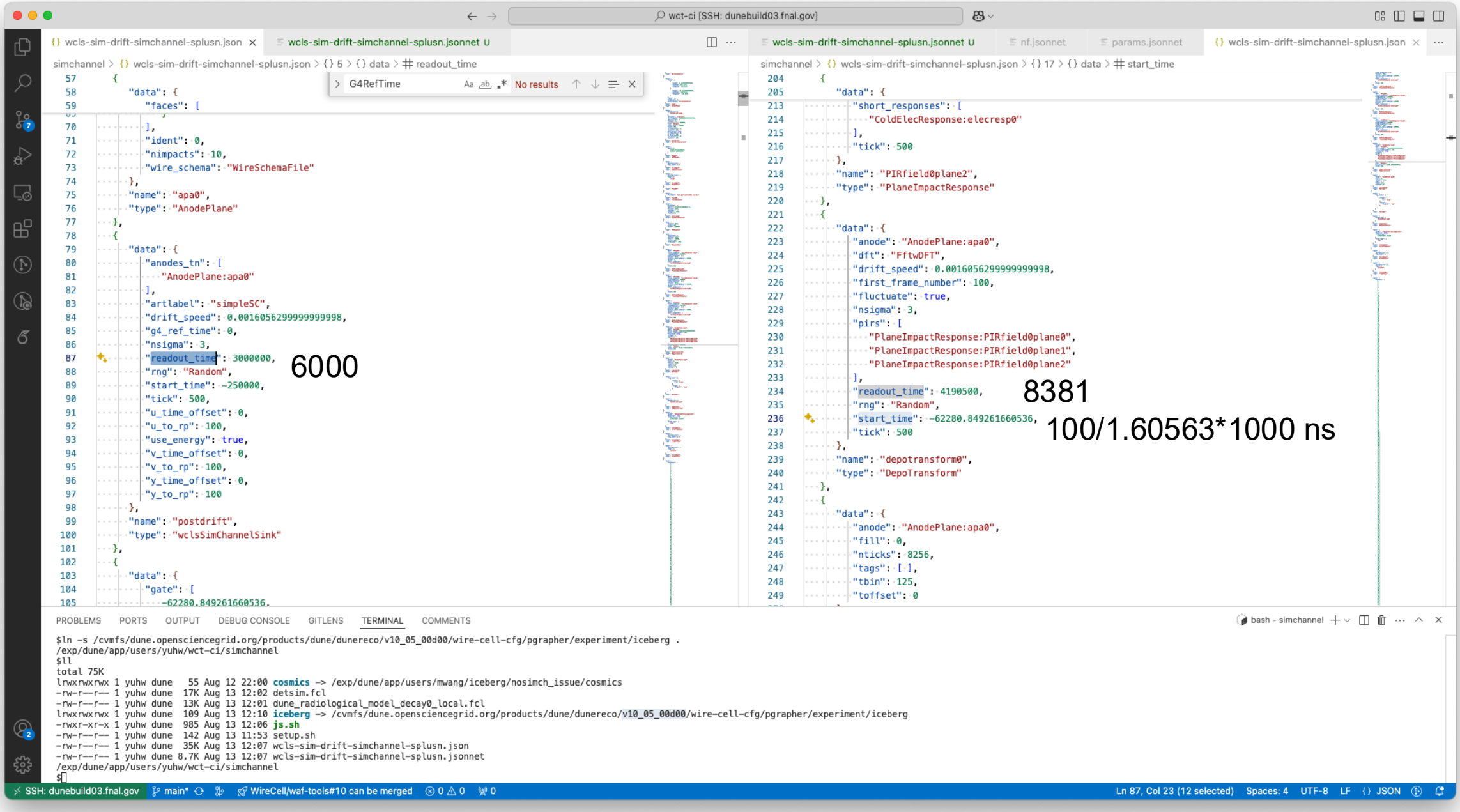
ICEBERG

v10_05_00d00


wct-ci [SSH: dunebuild03.fnal.gov]
wcls-sim-drift-simchannel-splusn.json
wcls-sim-drift-simchannel-splusn.jsonnet U
simchannel > wcls-sim-drift-simchannel-splusn.json > 5 > data > readout_time
G4RefTime
No results
"readout_time": 3000000,
6000
wcls-sim-drift-simchannel-splusn.jsonnet U
nf.jsonnet
params.jsonnet
wcls-sim-drift-simchannel-splusn.json
simchannel > wcls-sim-drift-simchannel-splusn.json > 17 > data > start_time
"readout_time": 4190500,
8381
"start_time": -62280.849261660536,
100/1.60563*1000 ns
PROBLEMS PORTS OUTPUT DEBUG CONSOLE GITLENS TERMINAL COMMENTS
bash - simchannel
$ln -s /cvmfs/dune.opensciencegrid.org/products/dune/dunereco/v10_05_00d00/wire-cell-cfg/pgrapher/experiment/iceberg .
/exp/dune/app/users/yuhw/wct-ci/simchannel
$ll
total 75K
lrwxrwxrwx 1 yuhw dune 55 Aug 12 22:00 cosmics -> /exp/dune/app/users/mwang/iceberg/nosimch_issue/cosmics
-rw-r--r-- 1 yuhw dune 17K Aug 13 12:02 detsim.fcl
-rw-r--r-- 1 yuhw dune 13K Aug 13 12:01 dune_radiological_model_decay0_local.fcl
lrwxrwxrwx 1 yuhw dune 109 Aug 13 12:10 iceberg -> /cvmfs/dune.opensciencegrid.org/products/dune/dunereco/v10_05_00d00/wire-cell-cfg/pgrapher/experiment/iceberg
-rwxr-xr-x 1 yuhw dune 985 Aug 13 12:06 js.sh
-rw-r--r-- 1 yuhw dune 142 Aug 13 11:53 setup.sh
-rw-r--r-- 1 yuhw dune 35K Aug 13 12:07 wcls-sim-drift-simchannel-splusn.json
-rw-r--r-- 1 yuhw dune 8.7K Aug 13 12:07 wcls-sim-drift-simchannel-splusn.jsonnet
/exp/dune/app/users/yuhw/wct-ci/simchannel
SSH: dunebuild03.fnal.gov
main*
WireCell/waf-tools#10 can be merged
Ln 87, Col 23 (12 selected)
Spaces: 4
UTF-8
LF
JSON

DepoTransform/ductor

```
sim: super.sim {

  // For running in LArSoft, the simulation must be in fixed time mode.
  fixed: true,

  // The "absolute" time (ie, in G4 time) that the lower edge of
  // of final readout tick #0 should correspond to.  This is a
  // "fixed" notion.
  local tick0_time = if std.objectHas(params, 'G4RefTime') then params.G4RefTime else 0,

  // Open the ductor's gate a bit early.
  local response_time_offset = $.det.response_plane / $.lar.drift_speed,
  local response_nticks = wc.roundToInt(response_time_offset / $.daq.tick),

  ductor: {
    nticks: $.daq.nticks + response_nticks,
    readout_time: self.nticks * $.daq.tick,
    start_time: tick0_time - response_time_offset,
  },

  // To counter the enlarged duration of the ductor, a Reframer
  // chops off the little early, extra time.  Note, tags depend on how
  reframer: {
    tbin: response_nticks,
    nticks: $.daq.nticks,
  },

},
```

My understanding:

- Depo times are G4 times drifted at the response plane
- Here the start/during decide a windown to include which depos

DepoFluxWriter

```
local wcls_depoflux_writer = g.pnode({
  type: 'wclsDepoFluxWriter',
  name: 'postdrift',
  data: {
    anodes: [wc.tn(anode) for anode in tools.anodes],
    field_response: wc.tn(tools.field),
    tick: 0.5 * wc.us,
    window_start: params.sim.tick0_time, // -205 * wc.us,
    window_duration: self.tick * params.daq.nticks,
    nsigma: 3.0,

    reference_time: -1700 * wc.us - self.window_start, // target is tick 410 should be 3400

    //energy: 1, # equivalent to use_energy = true
    simchan_label: 'simpleSC',
    sed_label: if (savetid == 'true') then 'ionandscint' else '',
    sparse: false,
  },
}, nin=1, nout=1, uses=tools.anodes + [tools.field]);
```

- input depos have times at resp. plane
- shift time to collection plane
- window start/duration are for times at collection plane

SimChannelSink

```
local wcls_simchannel_sink = g.pnode({
  type: 'wclsSimChannelSink',
  name: 'postdrift',
  data: {
    artlabel: 'simpleSC', // where to save in art::Event
    anodes_tn: [wc.tn(anode) for anode in tools.anodes],
    rng: wc.tn(rng),
    tick: 0.5 * wc.us,
    start_time: -0.25 * wc.ms,
    readout_time: self.tick * 6000,
    nsigma: 3.0,
    drift_speed: params.lar.drift_speed,
    u_to_rp: 100 * wc.mm, // 90.58 * wc.mm,
    v_to_rp: 100 * wc.mm, // 95.29 * wc.mm,
    y_to_rp: 100 * wc.mm,
    u_time_offset: 0.0 * wc.us,
    v_time_offset: 0.0 * wc.us,
    y_time_offset: 0.0 * wc.us,
    g4_ref_time: fcl_params.G4RefTime, // -250 * wc.us,
    use_energy: true,
  },
}, nin=1, nout=1, uses=tools.anodes);
```

```
void SimChannelSink::save_as_simchannel(const WireCell::IDepo::pointer& depo)
{
  // Binning tbins(m_readout_time/m_tick, m_start_time, m_start_time+m_readout_time);

  /*  Start the gate ealier for the depos between the response
   *  plane and the anode plane. Those depos are anti-drifted
   *  to the reponse plane, so the start time is earlier.
   *  c.f. jsonnet config in wirecell toolkit: params.sim.ductor
   */
  // double response_plane = 10.0 * units::cm;
  double response_time_offset = m_response_plane / m_drift_speed;
  int response_nticks = (int)(response_time_offset / m_tick);
  Binning tbins(m_readout_time / m_tick + response_nticks,
                m_start_time - response_time_offset,
                m_start_time + m_readout_time);
```

g4_ref_time is used to shift the tdc filled in SimChannel

```
  unsigned int temp_time = (unsigned int)((tdc - m_g4_ref_time) / m_tick);
  charge = abs(charge);
  if (charge > 1) {
    sc.AddIonizationElectrons(
      id, temp_time, charge, xyz, energy * abs(charge / depo->charge()));
  }
}
```

Brookhaven National Laboratory

fixing

```
/exp/dune/app/users/yuhw/dunereco
$git diff
diff --git a/dunereco/DUNEWireCell/iceberg/params.jsonnet b/dunereco/DUNEWireCell/iceberg/params.jsonnet
index 1ec5b352..4093135f 100644
--- a/dunereco/DUNEWireCell/iceberg/params.jsonnet
+++ b/dunereco/DUNEWireCell/iceberg/params.jsonnet
@@ -86,7 +86,7 @@ function(params) base {
     // The "absolute" time (ie, in G4 time) that the lower edge of
     // of final readout tick #0 should correspond to.  This is a
     // "fixed" notion.
-    local tick0_time = if std.objectHas(params, 'G4RefTime') then params.G4RefTime else 0,
+    tick0_time: if std.objectHas(params, 'G4RefTime') then params.G4RefTime else 0,

     // Open the ductor's gate a bit early.
     local response_time_offset = $.det.response_plane / $.lar.drift_speed,
@@ -95,7 +95,7 @@ function(params) base {
     ductor: {
       nticks: $.daq.nticks + response_nticks,
       readout_time: self.nticks * $.daq.tick,
-      start_time: tick0_time - response_time_offset,
+      start_time: $.sim.tick0_time - response_time_offset,
     },

     // To counter the enlarged duration of the ductor, a Reframer
diff --git a/dunereco/DUNEWireCell/iceberg/wcls-sim-drift-simchannel-splusn.jsonnet b/dunereco/DUNEWireCe
nnet
index 0282b8e9..9c41d4df 100644
--- a/dunereco/DUNEWireCell/iceberg/wcls-sim-drift-simchannel-splusn.jsonnet
+++ b/dunereco/DUNEWireCell/iceberg/wcls-sim-drift-simchannel-splusn.jsonnet
@@ -138,9 +138,9 @@ local wcls_simchannel_sink = g.pnode({
     artlabel: 'simpleSC',  // where to save in art::Event
     anodes_tn: [wc.tn(anode) for anode in tools.anodes],
     rng: wc.tn(rng),
-    tick: 0.5 * wc.us,
-    start_time: -0.25 * wc.ms,
-    readout_time: self.tick * 6000,
+    tick: params.daq.tick,
+    start_time: params.sim.tick0_time,
+    readout_time: self.tick * params.daq.nticks,
     nsigma: 3.0,
     drift_speed: params.lar.drift_speed,
     u_to_rp: 100 * wc.mm,  // 90.58 * wc.mm,
```

after fixing