Tagger pipeline TDCs for T-counters

- We had two sets of TDCs for T-counters:
  - Multi-hit LeCroy 1877. 500 ps LSB. Double pulse resolution 25 ns. A hardware coincidence of $TL_i x TR_i$ as an input.
  - Pipeline TDC from CAEN. 25 ps LSB. Double pulse resolution 5 ns. Individual channel for each TL_i and Tr_i. We need to arrange $TL_i x TR_i$ in software.
- We can take advantage of that to cross check results with two versions of tagger reconstruction.
- Experience with these new TDCs in recent CLAS runs is very positive.
- A development of the software for PRIMEX have been started when we took data a year ago.

```
----------------------------
revision 1.18
date: 2004/10/08 19:17:37;  author: pasyuk;  state: Exp;  lines: +12 -3
1. change type of 'val' in HARDWARE bank to 'unsigned int'
2. add TAGTLHIT and TAGTRHIT banks
----------------------------
----------------------------
revision 1.15
date: 2004/08/31 03:07:42;  author: mikewood;  state: Exp;  lines: +6 -2
Added new bank definitons for TAGTL and TAGTR (raw left and right T-counter TDCs)
----------------------------
```

bank definitions

```
<bank name="tagt" comment="Tagger T-counter raw TDCs after Left-Right Coincidence" typedef="primRawData_t"
size="primRawDataHit_t" bankflag="BANKFLAG_RAWBANK" />

<bank name="tagtl" comment="Tagger T-counter raw TDCs (Left)" typedef="primRawData_t" size="primRawDataHit_t"
bankflag="BANKFLAG_RAWBANK" />

<bank name="tagtr" comment="Tagger T-counter raw TDCs (Right)" typedef="primRawData_t"
size="primRawDataHit_t" bankflag="BANKFLAG_RAWBANK" />

<bank name="tage" comment="Tagger E-counter raw TDCs" typedef="primRawData_t" size="primRawDataHit_t"
bankflag="BANKFLAG_RAWBANK" />

<struct name="TAG_HIT_t">
<member type="int" name="id" comment="T-counter number"/>
<member type="float" name="t" comment="nanoseconds"/>
<member type="int" name="tdc" comment="TDC counts"/>
<member type="float" name="E" comment="Energy in GeV"/>
<member type="int" name="status"/>
</struct>

<struct name="TAGTLR_HIT_t">
<member type="int" name="id" comment=""/>
<member type="int" name="tdc" comment="TDC counts with reference time subtracted"/>
<member type="float" name="t" comment="nanoseconds"/>
</struct>

<bank name="tagehit" rowdef="TAG_HIT_t" comment="Tagger E-channel hits"/>

<bank name="tagthit" rowdef="TAG_HIT_t" comment="Tagger T-channel hits"/>

<bank name="tagtlhit" rowdef="TAGTLR_HIT_t" comment="Tagger left T-counter hits"/>

<bank name="tagtrhit" rowdef="TAGTLR_HIT_t" comment="Tagger right T-counter hits"/>

<bank name="tagm" comment="reconstructed tagger bank">
<column name="energy" type="float" comment="Energy of the photon in GeV" />
<column name="t" type="float" comment="T-counter time (ns)" />
<column name="e_t" type="float" comment="E-counter time (ns)" />
<column name="status" type="int" comment="Status (not yet used)" />
<column name="tid" type="int" comment="T channel Id" />
<column name="eid" type="int" comment="E channel Id" />
<column name="Thit" type="int" comment="index of hit in tagt_hit bank" />
<column name="Ehit" type="int" comment="index of hit in tage_hit bank" />
</bank>
```

```
//  routine name: tagM_GetTcounterHits
  if(HI_RES_TDC){
    for(i=0;i<banks->TAGTL->bank.nrow;i++){
      idL  = banks->TAGTL->hit[i].id;
      tdcL = banks->TAGTL->hit[i].val;
      tL  = (float)(tdcL-TcL_offset[idL-1]);
      tL *= TcL2ns[idL-1]; /* convert to nanoseconds  */
      for(j=0;j<banks->TAGTL->bank.nrow;j++){
idL  = banks->TAGTR->hit[i].id;
tdcR = banks->TAGTR->hit[j].val;
tR  = (float)(tdcR-TcR_offset[idR-1]);
tR *= TcR2ns[idR-1]; /* convert to nanoseconds  */
tdiff = fabs(tL-tR); /* left,right time difference */
if((idL==idR) && (tdiff<TAG_LR_WINDOW)){
  t_hits[n].status = 0;
  t_hits[n].id  = id;
  t_hits[n].tdc = (tdcL+tdcR)/2;
  t_hits[n++].t = (tL+tR)/2.0;     /* nanoseconds  */
}
      }
    }
  }
  else{
    for(i=0;i<banks->TAGT->bank.nrow;i++){
      id  = banks->TAGT->hit[i].id;
      t_hits[n].status = 0;
      tdc = banks->TAGT->hit[i].val;
      t  = (float)(tdc-Tc_offset[id-1]);
      t *= Tc2ns;   /* convert to nanoseconds  */
      t_hits[n].id  = id;
      t_hits[n].tdc = tdc;
      t_hits[n++].t = t; /* nanoseconds  */
    }
  }
```

```
//  routine name: tagM_MakeTLRcounterHitsBanks
//
//  RCS ID string.  Don't edit next line! CVS takes care of it.
//  $Id: tagM_MakeTLRCounterHitsBanks.c,v 1.2 2005/07/12 13:49:51 eclinton Exp $
//
//_begin_doc
//
//  Documentation for subroutine tagger_evnt
//
//  Purpose:  Read Tagger left-right T-counter TDC info from TAGTL&TAGTR banks,
//            subtract reference channel and fill TAGTLHIT&TAGTRHIT banks
//  --------
//
//  Calling Sequence:
//  ----------------
//
//  Input Parameters:  (Name - Type - Meaning)
//  ----------------
//
//  Output Parameters:  (Name - Type - Meaning)
//  -----------------
//
//  Called from:
//  ------------
//
//  Other routines:
//  ---------------
//
//  Notes:
//  ------
//
//  Author:   Eugene Pasyuk      Created:  Thu Oct 12 15:19:04 EDT 2004
//  -------
//
//  Major revisions:
//  ----------------
//
//_end_doc
```

Status

- translation to TAGTL/TAGTR banks in codaIO - DONE
- conversion from TDC counts to ns – almost DONE
- Left/Right matching. Makes bank similar to TAGTHIT – TO DO
- The rest of the reconstruction should go the same way as we had before: make new bank (TAGMLR?) bank, sort of a clone of TAGM – TO DO