

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 1/28

#include "micalCal0SD.hh"
#include "G4Step.hh"
#include "G4HCofThisEvent.hh"
#include "G4TouchableHistory.hh"
#include "G4ios.hh"
#include "TRandom.h"
#include "TMath.h"
// #include "micalDetectorParameterDef.hh"

#include "vect_manager.h"

#include "Randomize.hh"
#include "CLHEP/Random/RandGauss.h"
#define multiplicity 0

micalcal0SD::micalcal0SD(G4String name)
:G4VSensitiveDetector(name),
// numberInMO(16), numberInCH(7), numberInLA(140),
// numberInX(100), numberInY(100), numberInT(4), numberInCell(20000), InCell(0)
{
  G4String HCname;
  collectionName.insert(HCname="cal0Collect");
  cal0SDMessenger = new micalcal0SDMessenger(this);
  pAnalysis = MultiSimAnalysis::AnPointer;
  // inoHit_pointer = new InoHit_Manager();
  inoStripX_pointer = new InoStripX_Manager();
  inoStripY_pointer = new InoStripY_Manager();
  paradev = micalDetectorParameterDef::AnPointer;
  twopow31= pow(2,31);
  NewMultiplicity = 1;
  cout<<"-----"<<endl;
  if (NewMultiplicity){cout<<" Strip Multiplicity enabled "<<endl;}
  else{cout<<" Strip Multiplicity disabled "<<endl;}
  cout<<"-----"<<endl;
  SetTimeToDigiConv(0.1);
  SetSignalSpeed(0.15);
  SetCorrTimeSmear(0.7);
  SetUnCorrTimeSmear(0.7);
  SetRootRandom(1);
  // Define All the other parameters
}

micalcal0SD::~micalcal0SD() {
  for (unsigned ij=0; ij<inoStripX_pointer->InoStripX_list.size(); ij++) {
    if (inoStripX_pointer->InoStripX_list[ij]) {
      // cout <<"ij "<< ij<<" "<<inoStripX_pointer->InoStripX_list.size()<<endl;
      delete inoStripX_pointer->InoStripX_list[ij]; inoStripX_pointer->InoStripX
_list[ij]=0;
    }
  }

  inoStripX_pointer->InoStripX_list.clear();
  if (inoStripX_pointer) {delete inoStripX_pointer; inoStripX_pointer=0;}
  for (unsigned ij=0; ij<inoStripY_pointer->InoStripY_list.size(); ij++) {
    if (inoStripY_pointer->InoStripY_list[ij]) {
      delete inoStripY_pointer->InoStripY_list[ij]; inoStripY_pointer->InoStripY
_list[ij]=0;
    }
  }
  inoStripY_pointer->InoStripY_list.clear();
  if (inoStripY_pointer) {delete inoStripY_pointer; inoStripY_pointer=0;}
}

void micalcal0SD::Initialize(G4HCofThisEvent* HCE) {
  // cout<<"micalcal0SD::Initialize(..."<<endl;
  static int HCID = -1;
  cal0Collection = new micalcal0HitsCollection

```

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 2/28

(SensitiveDetectorName, collectionName[0]);
if (HCID<0) { HCID = GetCollectionID(0); }
HCE->AddHitsCollection(HCID, cal0Collection);

// InoHit_Manager* tmp_vect = InoHit_Manager::APointer;
InoHit_list.clear();
// inoHit_pointer->InoHit_list.clear();
paradev = micalDetectorParameterDef::AnPointer;

inoStripX_pointer->InoStripX_list.clear();
inoStripY_pointer->InoStripY_list.clear();

histxmn = histymn = histzmn = 100000.;
histxmx = histymx = histzmx = -100000.;

for (int ij=0; ij<3; ij++) {parino[ij] = paradev->GetParino(ij);
// cout<<"parino["<<ij<<"] = "<<parino[ij] << ", "<<paradev->GetParino(ij)<<e
ndl;
}
for (int ij=0; ij<3; ij++) {parlay[ij] = paradev->GetParlay(ij);}
// for (int ij=0; ij<3; ij++) {parmod[ij] = paradev->GetParmod(ij);}
for (int ij=0; ij<3; ij++) {parchm[ij] = paradev->GetParchm(ij);}
// for (int ij=0; ij<3; ij++) {parair[ij] = paradev->GetParair(ij);}
// for (int ij=0; ij<3; ij++) {parirnlay[ij] = paradev->GetParirnlay(ij);}
for (int ij=0; ij<3; ij++) {parcup[ij] = paradev->GetParcup(ij);}
// for (int ij=0; ij<3; ij++) {parg10[ij] = paradev->GetParg10(ij);}

for (int ij=0; ij<3; ij++) {parqurz[ij] = paradev->GetParqurz(ij);}
for (int ij=0; ij<3; ij++) {pargas[ij] = paradev->GetPargas(ij);}

for (int ij=0; ij<3; ij++) { parirlay[ij] = paradev->GetParirlay(ij);}
for (int ij=0; ij<3; ij++) { parhcoil[ij] = paradev->GetParhcoil(ij);}
for (int ij=0; ij<3; ij++) { parcoilssupport[ij] = paradev->GetParcoilssupport (i
j);}

nINODet = 1; //paradev->GetNumino();
gapino = 0; // paradev->GetGapino();

Xstrwd = paradev->GetXStrwd();
Ystrwd = paradev->GetYStrwd();

numberInX = paradev->GetnXStrip();
numberInY = paradev->GetnYStrip();

numberInMO = 1; //paradev->GetnModule();
numberInCH = 2; //paradev->GetnChamber();
numberInLA = paradev->GetnLayer();

if ( numberInMO >8) numberInMO=8;
if ( numberInCH >8) numberInCH=8;
if ( numberInLA >256) numberInLA=256;
// 12334457,1239075
// 1202219559

if (RootRandom==0) {
  gRandom->SetSeed(1327511442);
}
// cout<<"micalcal0SD::Initialize( complete..."<<endl;
}

G4bool micalcal0SD::ProcessHits(G4Step* aStep, G4TouchableHistory*) {
  G4double edep = aStep->GetTotalEnergyDeposit()/keV;
  // if (edep<1.e-6) return true;
  // edep = 100*keV;
  //G4double edep = aStep->GetTotalEnergyDeposit()/keV-aStep->GetNonIonizingEner
gyDeposit()/keV;
  // cout<<
  // G4cout <<"getname "<<GetName()<<G4endl;

```

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 3/28

// if (edep>0) G4cout <<"ical0cal0SD "<<aStep->GetTrack()->GetVolume()->GetNam
e())<<" x "<<aStep->GetTrack()->GetVolume()->GetLogicalVolume()->GetName()-<<" y "
<<aStep->GetTrack()->GetVolume()->GetLogicalVolume()->GetMaterial()->GetName()-<<
" "<<aStep->GetPreStepPoint()->GetPosition()-<<" "<<edep<<G4endl;

G4TouchableHistory* theTouchable = (G4TouchableHistory*)( aStep->GetPreStepPoi
nt()->GetTouchable() );
// cout<<"XXXXXXXXXXXXXXXXXXXXX"<<endl;
// for(int ij=0; ij<14; ij++) {
//   cout<<"ij "<<ij<<" physName "<<theTouchable->GetVolume(ij)->GetName()-<<"
material = "<<theTouchable->GetVolume(ij)->GetLogicalVolume()->GetMaterial()->G
etName()-<<endl;
// }
// cout<<"XXXXXXXXXXXXXXXXXXXXX"<<endl;
G4StepPoint* point = aStep->GetPreStepPoint(); //
G4int tmpint = theTouchable->GetCopyNumber( 7 );
// G4TouchableHandle touch = point->GetTouchableHandle();
// G4VPhysicalVolume* volum= touch->GetVolume();
// G4String name = volum->GetName();
// G4int copyNumber = touch->GetCopyNumber();
// G4LogicalVolume* lvolume= volum->GetLogicalVolume();
const G4Track* track = aStep->GetTrack();
// if (abs(track->GetDefinition()->GetPDGEncoding()) !=13) return t
rue;
int level = theTouchable->GetHistoryDepth();
// cout<<"particle "<<track->GetDefinition()->GetPDGEncoding()-<<" track ID
"<<track->GetParentID()-<<" "<</*track->GetTrackID<<*/" physical volume "<<n
ame<<" copy number "<<copyNumber<<" logical volume "<<lvolume->GetName
()-<<" level "<<level<<endl;

// cout << "aStep->GetPreStepPoint() = " << theTouchable->GetCopyNumber(7) <<
endl;

G4ThreeVector parmom = aStep->GetTrack()->GetMomentum();
// double trkPid=track->GetDefinition()->GetPDGEncoding();
double momentum= parmom.mag();
double polang = parmom.theta();
double aziang = parmom.phi();
// cout<<"momentum "<<momentum<<" theta "<<polang<<" phi "<<azi
ang<<" "<<edep<<endl;
/*if (edep>0)*/
// cout <<"ROHist " << level<<" "
// <<theTouchable->GetReplicaNumber(0)<<" "
// <<theTouchable->GetReplicaNumber(1)<<" "
// <<theTouchable->GetReplicaNumber(2)<<" "
// <<theTouchable->GetReplicaNumber(3)<<" "
// <<theTouchable->GetReplicaNumber(4)<<" "
// <<theTouchable->GetReplicaNumber(5)<<" "
// <<theTouchable->GetReplicaNumber(6)<<" "
// <<theTouchable->GetReplicaNumber(7)<<" "
// <<theTouchable->GetReplicaNumber(8)<<" "
// <<theTouchable->GetReplicaNumber(9)<<" "
// <<theTouchable->GetReplicaNumber(10)<<" "
// <<theTouchable->GetReplicaNumber(11)<<" "
// <<theTouchable->GetReplicaNumber(12)<<" "

// <<tan(polang)*cos(aziang)<<" "<<tan(polang)*sin(aziang)
<<" "<<edep<<endl;
// <<aStep->GetTrack()->GetTrackID()-<<" "
// <<aStep->GetTrack()->GetParentID()-<<" "
// <<aStep->GetTrack()->GetKineticEnergy()/GeV<<" "
// <<1./aStep->GetTrack()->GetTotalEnergy()/GeV<<" "
// <<aStep->GetPreStepPoint()->GetPosition()-<<" "
// <<setw(5)<<aStep->GetPreStepPoint()->GetGlobalTime()-<<" "
// <<setw(5)<<aStep->GetPreStepPoint()->GetLocalTime()-<<" "
// <<setw(5)<<aStep->GetPreStepPoint()->GetProperTime()-<<G4endl;

if(edep==0.) return false;

```

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 4/28

//GMAA ParentID() should be reoved, for the time being keep it for the test of
algorithms
if (level <9) {
  G4cout <<"Hits are not in the sensitive vol"<<G4endl;
  return false;
}

//20/02/2009 for visualisation plots

G4ThreeVector glbpos = 0.5*(aStep->GetPreStepPoint()->GetPosition() + aStep->G
etPostStepPoint()->GetPosition()); //aStep->GetPreStepPoint()->GetPosition();

float tmpx = (1/m)*glbpos.x();
float tmpy = (1/m)*glbpos.y();
float tmpz = (1/m)*glbpos.z();

if (tmpx >histxmx) histxmx = tmpx;
if (tmpx <histxmn) histxmn = tmpx;

if (tmpy >histymx) histymx = tmpy;
if (tmpy <histymn) histymn = tmpy;

if (tmpz >histzmx) histzmx = tmpz;
if (tmpz <histzmn) histzmn = tmpz;

// G4int tmpint = theTouchable->GetCopyNumber( 8 );
// G4int nInCH = 0;//tmpint%8; // theTouchable->GetCopyNumber( 6
) ;

G4int nInMO = 0;//int(tmpint/8); // theTouchable->GetCopyNumber( 5 );
G4int nInCH = theTouchable->GetCopyNumber( 8 );
G4int nInLA = theTouchable->GetCopyNumber( 9 );

G4int nInDT = 0;//theTouchable->GetCopyNumber( 10 );
// pAnalysis->timeAsciiOutput<<"ievt = "<<pAnalysis->ievt<<endl;
// pAnalysis->timeAsciiOutput<<"*****"<<endl;
// for(int ixj=0; ixj<14; ixj++) {
//   cout <<"theTouchable->GetVolume("<<ixj<<"->GetName() "<<theTouchable->Ge
tVolume(ixj)->GetName()-<<" "<<nInLA<<endl;
// }
// pAnalysis->timeAsciiOutput <<"G4int tmpint = theTouchable->GetCopyNumber( 7
) = "<<tmpint<<endl;

// pAnalysis->timeAsciiOutput <<"G4int nInCH = tmpint%8 = "<<nInCH<<endl;
// theTouchable->GetCopyNumber( 6 );
// pAnalysis->timeAsciiOutput <<"G4int nInMO = int(tmpint/8) = "<<nInMO<<endl;
// theTouchable->GetCopyNumber( 5 );
// pAnalysis->timeAsciiOutput<<"G4int nInLA = theTouchable->GetCopyNumber( 8 )
= "<<nInLA<<endl;
// pAnalysis->timeAsciiOutput<<"G4int nInDT = theTouchable->GetCopyNumber( 9 )
= "<<nInDT<<endl;
// pAnalysis->timeAsciiOutput<<"
"<<endl;

G4double atime = aStep->GetPreStepPoint()->GetGlobalTime()/ns;
// cout<<"geantTimeStamp = "<<atime<<" "<<aStep->GetPreStepPoint()->GetGlobal
Time()/ns<<endl;
G4int nInT = G4int(atime/125.); //(2*ns)); //(5*ns)); //(10*ns)); //(5*ns));
//maximum of 40 ns

G4ThreeVector localpos = theTouchable->GetHistory()->GetTopTransform().Transfo
rmPoint(glbpos); // 0.5*(aStep->GetPreStepPoint()->GetPosition() + aStep->GetPos
tStepPoint()->GetPosition());

// cout<<"glb "<<glbpos<<" loc "<<localpos<<" "<<glbpos-localpos<<endl;

// cout<<"XXXXXXXXXXXXXXXXXXXXX"<<endl;
// cout<<"atime "<<atime<<" nInT "<<nInT<<endl;
// cout<<"localpos "<< 1.e-1*localpos.x()-<<" "<< 1.e-1*localpos.y()-<<" "<< 1.e

```

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 5/28

-1*localpos.z()<<endl;
// nInT = 0; // 04/02/2009

const G4int MxStrip=3; //1; //GMA230621
const G4int UsedMxStrip=1;

//GMA This is only for test purpose, actual smearing and storing is done in micalcal0SD::EndOfEvent (G4HCofThisEvent*)
G4double CorrTimeSmr = G4RandGauss::shoot(0,TimeCorrSmr);
G4double atimeX = atime + CorrTimeSmr + G4RandGauss::shoot(0,TimeUnCorrSmr);
G4double atimeY = atime + CorrTimeSmr + G4RandGauss::shoot(0,TimeUnCorrSmr);

// cout<<"SmearedTime = "<<atime + CorrTimeSmr <<endl;
// cout<<"atimeX = "<<atimeX<<endl;
// cout<<"atimeY = "<<atimeY<<endl;

//Time shift due to propagation of signal in strip //Bring this through datab
ase
// double sigXspeed = 0.15*ns; // 5ns/m; 0.15ns/strip
// double sigYspeed = 0.15*ns; // 5ns/m; 0.15ns/strip
// atimeX += (Ytpos - Ymin)*0.005*ns; //5 ns/m
// atimeY += (Xtpos - Xmin)*0.005*ns;

G4int nInX[MxStrip]={-1, -1, -1}; //GMA230621
G4int nInY[MxStrip]={-1, -1, -1};

G4double yy = pargas[1] + localpos.y(); // /m; // /cm; //GMA factor 100 for me
ter to cm
nInY[0] = int(yy/Ystrwd);
G4double xx = pargas[0] + localpos.x(); // /m; // /cm;
nInX[0] = int(xx/Xstrwd);
// cout<<"nInX[0] "<<nInX[0] <<" "<<nInY[0]<<" "<<nInLA<<endl;

for (int ix = 0; ix < UsedMxStrip; ix++) {
    if(!multiplicity && ix>0) continue;
    // Aug3109: multiplicity of hits is put off, just to make the tracks comparabe
    to tracks from earlier code.
    if (nInX[ix] <0) continue;
    for (int iy = 0; iy < UsedMxStrip; iy++) {
        if(!multiplicity && iy>0) { continue;}
        if (nInY[iy] <0) {continue;}
        // if (ix>0 || iy>0) continue; //RANDOM

        // cout<<"nINODet "<<nINODet<<" "<<numberInCH<<" "<<numberInMO<<" "<<numbe
rInLA<<" "<<numberInX<<" "<<numberInY<<endl;
        // cout <<"Wrong numbers "<<nInLA<<" "<<nInX[ix]<<" "<<nInY[iy] <<" "<<aSt
ep->GetPreStepPoint()->GetPosition()<<" "<<localpos<<endl;

        double ShiftInX = parafdef->GetINOrroomPos(0)+ parafdef->GetStackPosInRoom(0)
+ parafdef->GetShiftInX();
        double ShiftInY = parafdef->GetINOrroomPos(1)+ parafdef->GetStackPosInRoom(1)
+ parafdef->GetShiftInY();
        double ShiftInZ = parafdef->GetINOrroomPos(2) + parafdef->GetStackPosInRoom(
2)+ parafdef->GetShiftInZ(nInLA);

        //      cout<<"ShiftInXYZ "<<ShiftInX<<" "<<ShiftInY<<" "<<ShiftInZ<<en
dl;

        double ddl = (-pargas[0] + Xstrwd*(nInX[ix]+0.5) + ShiftInX);
        double shift2y = (parafdef->GetnStack()>1) ? (2*nInCH-1)*
parchm[1] : 0; //GMA one/two RPC in a layer
        double dd2 = (shift2y -pargas[1] + Ystrwd*(nInY[iy]+0.5) + ShiftInY);
        // double dd3 = ZLayerPos[nInLA]/m;

        //      cout<<"recxpos "<<ddl<<" genxpos "<<glbpos.x()<<endl;
        // cout<<"recypos "<<dd2<<" genypos "<<glbpos.y()<<endl;

pAnalysis->pPosX->Fill(0.1*ddl - 0.1*glbpos.x());

```

```

Jul 16, 21 18:23      micalCal0SD.cc      Page 6/28

pAnalysis->pPosY->Fill(0.1*ddl - 0.1*glbpos.x());

if(nInDT <0 || nInDT >=nINODet ||
nInCH <0 || nInCH >=numberInCH ||
nInMO <0 || nInMO >=numberInMO ||
nInLA <0 || nInLA >=numberInLA ||
nInX[ix] <0 || nInX[ix] >=numberInX ||
nInY[iy] <0 || nInY[iy] >=numberInY){
    // cout <<"Wrong numbers "<<ix<<" "<<iy<<" "<<nInCH<<" "<<nInMO<<" "<<nI
nLA<<" "<<nInX[ix]<<" "<<nInY[iy]<<" "<<nInT <<" "<<aStep->GetPreStepPoint()->Ge
tPosition()<<" "<<localpos<<endl;
    continue;
}

unsigned long detid = nInDT; //2bit
detid<<=8;
detid +=nInLA;
detid<<=3;
detid +=nInMO;
detid<<=3;
detid +=nInCH;
detid<<=7;
detid +=nInX[ix];
detid<<=7;
detid +=nInY[iy];

// cout<<"nInDT,nInLA,nInMO,nInCH,nInX[ix],nInY[iy] "<<nInDT<<" "<<nInLA<<
" "<<nInMO<<" "<<nInCH<<" "<<ix<<" "<<nInX[ix]<<" "<<iy<<" "<<nInY[iy]<<endl;
// cout<<"XXXXXXXXXXXXXXXXXXXXX"<<endl;
int oldCellId = -1;
for (int ij=0; ij<InCell; ij++) {
    if (detid ==CellDetID[ij]) {oldCellId = ij;}
}
// cout<<" oldCellId "<<oldCellId<<" "<<InCell<<" "<<numberInCell<<endl;
if (oldCellId ==-1 && InCell <numberInCell -1 ) {
    micalcal0Hit* newHit = new micalcal0Hit();
    // cout<<"detid = "<< detid%128 <<" nInY[iy] = "<<nInY[iy]<<endl;
    // cout<<" 1 "<<"atime = "<<atime<<" GetTime() "<<" pdgid "<<aStep->Get
Track()->GetDefinition()->GetPDGEncoding()<<" GetPdgid "<<" oldcellid "<<oldCell
Id<<endl;
    // cout<<" 1 aStep->GetTrack() "<<aStep->GetTrack()->GetTrackID()<<" "<<
aStep->GetTrack()->GetParentID()<<endl;
    // cout<<" 1 xpos = "<<glbpos.x()<<" y = "<<glbpos.y()<<" z = "<<glbpos.
z()<<endl;

    newHit->SetHitId(detid);
    int pdgid = aStep->GetTrack()->GetDefinition()->GetPDGEncoding();
    newHit->Setpdgid(pdgid);
    newHit->SetEdep(edep);
    newHit->SetTime(atime);
    newHit->SetPos(glbpos); // 0.5*(aStep->GetPreStepPoint()->GetPosition()
+ aStep->GetPostStepPoint()->GetPosition());
    newHit->SetLocalXPos(localpos.x());
    newHit->SetLocalYPos(localpos.y());
    //      newHit->SetLocalPos(localpos);
    newHit->SetMom( aStep->GetTrack()->GetMomentum());

    InCell = cal0Collection->insert( newHit );
    CellDetID[InCell-1] = detid;

    // double MCxx = 0.0;
    // double MCyy = 0.0;
    // double MCzz = 0.0;

    //      G4ThreeVector glb = 0.5*(aStep->GetPreStepPoint()->GetPosition()
+ aStep->GetPostStepPoint()->GetPosition());
    // MCxx = 1.e-3*glb.x();
    // MCyy = 1.e-3*glb.y();
    // MCzz = 1.e-3*glb.z();

```

Jul 16, 21 18:23 micalCal0SD.cc Page 7/28

```

//cout<<"MCxx "<<MCxx<<"      MCyy "<<MCyy<<"      MCzz "<<MCzz<<endl;
//
//GMA Need from micalcal0HitCollection
//GMA This is only for test purpose, actual smearing and storing is done
in micalcal0SD::EndOfEvent(G4HCofThisEvent*)
G4int xstripid = 0;
xstripid<=2; //1;
xstripid +=nInDT;

xstripid<=8; // 2;
xstripid +=nInLA;
xstripid<=3; // 8;
xstripid +=nInMO;

xstripid<=3;
xstripid +=nInCH;

xstripid<=7; //3;
xstripid +=nInX[ix];

xstripid<=5;
xstripid +=0; // nInT;

xstripid<=3;
xstripid +=TMath::Min(int(edep/16),7);

G4int ystripid = 1;

ystripid<=2; //1;
ystripid +=nInDT;

ystripid<=8;
ystripid +=nInLA;

ystripid<=3;
ystripid +=nInMO;

ystripid<=3;
ystripid +=nInCH;

ystripid<=7; //3;
ystripid +=nInY[iy];

ystripid<=5;
ystripid +=0; // nInT;

ystripid<=3;
ystripid +=TMath::Min(int(edep/16),7);

InoStrip Xstrip;
InoStrip Ystrip;

Xstrip.Setpdgid(pdgid);
Ystrip.Setpdgid(pdgid);

Xstrip.SetPlaneView(0);
Ystrip.SetPlaneView(1);

Xstrip.SetStrip(numberInX*numberInMO*nInDT+numberInX*nInMO+nInX[ix]);
Ystrip.SetStrip(numberInY*nInCH+nInY[iy]);

Xstrip.SetPlane(nInLA);
Ystrip.SetPlane(nInLA);

//      G4ThreeVector posvec = aStep->GetPreStepPoint()->GetPosition();
//GMA14 Define only once
//GMA 250808

```

Jul 16, 21 18:23 micalCal0SD.cc Page 8/28

```

//      double ShiftInX = parafdef->GetShiftInX();
//      double ShiftInY = parafdef->GetShiftInY();
//      double ShiftInZ = parafdef->GetShiftInZ(nInLA);
double ShiftInX = parafdef->GetInOroomPos(0)+parafdef->GetStackPosInRoom(0)
) + parafdef->GetShiftInX();
double ShiftInY = parafdef->GetInOroomPos(1)+parafdef->GetStackPosInRoom(1)
) + parafdef->GetShiftInY();
double ShiftInZ = parafdef->GetInOroomPos(2) +parafdef->GetStackPosInRoom(
2)+ parafdef->GetShiftInZ(nInLA);
// // ShiftInX = parafdef->GetShiftInX();
// // ShiftInY = parafdef->GetShiftInY();
// // ShiftInZ = parafdef->GetShiftInZ();

// cout<<"ShiftInX "<<0.1*ShiftInX<<" "<<0.1*ShiftInY<<" "<<0.1*ShiftInZ
<<endl;

// double xpos = (1/m)*(nInDT-1)*(2*parino[0]+gapino) - parlay[0] +
(2*nInMO+1)*parmod[0] -pargas[0] + Xstrwd*(nInX[ix]+0.5) + ShiftInX); //GMA use
global variables (for all these three co-ordinates)
// //0.01 is the conversion factor for cm t m
// double ypos = (1/m)*(- parmod[1] + (2*nInCH+1)*parchm[1] -pargas[1]
+ Ystrwd*(nInY[iy]+0.5) + ShiftInY);
// double zpos = (1/m)*(-(numberInLA-1)*(parirlay[2]+parlay[2])+(nInLA)*
2*(parirlay[2] + parlay[2]) + ShiftInZ); //AAR:** changes for Central Iron Layer
**

// cout <<"Glo Position "<<1000* xpos<<" "<<1000*ypos<<" "<<1000*zpos<<"
glb "<<glbpos<<" Diff_x "<<1000*xpos-glbpos.x()<<" Diff_y "<<1000*ypos-glbpos.y
()<<endl;

//      cout<<"parlay[0]"<<parlay[0]<<"gapino"<<gapino<<"pariino[0]"<<p
arino[0]<<"nInMO"<<nInMO<<"parmod[0]"<<parmod[0]<<"nindt"<<nInDT<<endl;

double xpos = (1/m)*( -pargas[0] + Xstrwd*(nInX[ix]+0.5) + ShiftInX);
//GMA use global variables (for all these three co-ordinates)
//0.01 is the conversion factor for cm t m
double shift2y = (parafdef->GetnStack())>1) ? (2*nInCH-1)*parchm[1] : 0; /
/GMA one/two RPC in a layer
double ypos = (1/m)*( shift2y -pargas[1] + Ystrwd*(nInY[iy]+0.5) + Shift
InY);
double zpos = (1/m)*(-(numberInLA-1)*(parirlay[2]+parlay[2])+(nInLA)*2*(
parirlay[2] + parlay[2]) + ShiftInZ); //AAR:** changes for Central Iron Layer **

//      cout <<"Glo Pos "<<nInLA<<" "<<aStep->GetTrack()->GetMomentum()
.mag()<<" "<<aStep->GetTrack()->GetMomentum().theta()<<" "<<aStep->GetTrack()->G
etMomentum().phi()<<" "<<1000* xpos<<" "<<1000*ypos<<" "<<1000*zpos<<" glb "<<gl
bpos<<endl; // " Diff_x "<<1000*xpos-glbpos.x()<<" Diff_y "<<1000*ypos-glbpos.y()
<<endl;

cout <<"Glo Pos "<<pdgid<<" "<<edep<<" "<<nInLA<<" "<<1000* xpos<<" "<<1000
*ypos<<" "<<1000*zpos<<" glb "<<glbpos<<" Diff_x "<<1000*xpos-glbpos.x()<<" Diff_y "<<10
00*ypos-glbpos.y()<<" "<<aStep->GetTrack()->GetMomentum()<<endl;

//localpos

//      double lo_xpos = (1/m)*( -pargas[0] + Xstrwd*(nInX[ix]+0.5)); /
/GMA use global variables (for all these three co-ordinates)
//0.01 is the conversion factor for cm t m
//      double lo_ypos = (1/m)*( -pargas[1] + Ystrwd*(nInY[iy]+0.5));
//      double lo_zpos = (1/m)*(-(numberInLA-1)*(parirlay[2]+parlay[2])+(
nInLA)*2*(parirlay[2] + parlay[2]) + ShiftInZ); //AAR:** changes for Central Ir
on Layer **

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 9/28

```

//      cout <<"local Position "<<1000* lo_xpos<<" "<<1000*lo_ypos<<" "<
<1000*lo_zpos<<" loc "<<localpos<<" Diff_x "<<1000*lo_xpos-localpos.x()<<" Diff_
y "<<1000*lo_ypos-localpos.y()<<endl<<endl;

//

Xstrip.SetXYPos(xpos);
Ystrip.SetXYPos(ypos);

Xstrip.SetZPos(zpos);
Ystrip.SetZPos(zpos);

Xstrip.SetMomentum(momentum);
Xstrip.SetTheta(polang);
Xstrip.SetPhi(aziang);

//Do not need it.
Ystrip.SetMomentum(momentum);
Ystrip.SetTheta(polang);
Ystrip.SetPhi(aziang);

Xstrip.SetTrueTime(atime/0.1);
Ystrip.SetTrueTime(atime/0.1);

Xstrip.SetSmrTime((atimeX+(nInY[iy]+0.5)*SignalSpeed)/0.1);
Ystrip.SetSmrTime((atimeY+(nInX[ix]+0.5)*SignalSpeed)/0.1);

// cout<<"nInY[iy] = "<<nInY[iy]<<endl;
// cout<<"nInX[ix] = "<<nInX[ix]<<endl;
// cout<<"sigSpeed*nInY = "<<(nInY[iy]+0.5)*SignalSpeed <<endl;
// cout<<"sigSpeed*nInX = "<<(nInX[ix]+0.5)*SignalSpeed <<endl;
// cout<<"Non digitised X-Time = "<<(atimeX+(nInY[iy]+0.5)*SignalSpeed)
<endl;
// cout<<"Non digitised Y-Time = "<<(atimeY+(nInX[ix]+0.5)*SignalSpeed)
<endl;
// cout<<"Digitised X-Time = "<< int((atimeX+(nInY[iy]+0.5)*SignalSpeed)
/0.1) <<endl;
// cout<<"Digitised Y-Time = "<< int((atimeY+(nInX[ix]+0.5)*SignalSpeed)
/0.1) <<endl;

Xstrip.SetPulse(edep);
Ystrip.SetPulse(edep);

InoHit tmpHit(&Xstrip, &Ystrip);
InoHit_list.push_back(tmpHit);
//-----ascii_output

if (pAnalysis->isVisOut==1&&(pAnalysis->InputOutput==0 ||pAnalysis->InputOutput==3 ||pAnalysis->InputOutput==5)) {
    if(aStep->GetTrack()->GetTrackID()==1 && aStep->GetTrack()->GetParentID()==0 ) {
        pAnalysis->H->NPrimHits=2; //Number of Triplet events
        pAnalysis->Hp= pAnalysis->H->AddHits(0,0); // add a track object //V
ALGRIND
        pAnalysis->Hp->TrackType=-101; // Track Type: -1: hits, -2: clulster,
-3: triplet, -4: track
        pAnalysis->Hp->ParCode= 13;
        pAnalysis->Hp->PrimHitNum= 0; // Hit Number
        pAnalysis->Hp->ZZ=Xstrip.GetPlane();
        pAnalysis->Hp->XX=Xstrip.GetXYPos();
        pAnalysis->Hp->YY=Ystrip.GetXYPos();
        // hh++;
    } else {
        pAnalysis->H->NPrimHits=2; //Number of Triplet events
        pAnalysis->Hp= pAnalysis->H->AddHits(0,0); // add a track object //V

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 10/28

```

ALGRIND
        pAnalysis->Hp->TrackType=-101; // Track Type: -1: hits, -2: clulster,
-3: triplet, -4: track
        pAnalysis->Hp->ParCode= 1;
        pAnalysis->Hp->PrimHitNum= 1; // Hit Number
        pAnalysis->Hp->ZZ=Xstrip.GetPlane();
        pAnalysis->Hp->XX=Xstrip.GetXYPos();
        pAnalysis->Hp->YY=Ystrip.GetXYPos();
    }
    //      cout<<"ParentID " << aStep->GetTrack()->GetParentID();
    //-----

}
if (oldCellId >=0) {
    (*cal0Collection)[oldCellId]->AddEdep(edep);
    if (atime < (*cal0Collection)[oldCellId]->GetTime()) {
        //      cout<<"*****"
*****"<<endl;
        //      cout<<"atime = "<<atime<<" GetTime() "<<(*cal0Collection)[oldCellId]->GetTime()<<" pdgid "<<aStep->GetTrack()->GetDefinition()->GetPDGEncoding()<<" GetPdgid "<<(*cal0Collection)[oldCellId]->Getpdgid()<<" oldcellid "<<oldCellId<<endl;
        //      cout<<" aStep->GetTrack() "<<aStep->GetTrack()->GetTrackID()<<" "<<aStep->GetTrack()->GetParentID()<<endl;
        //      G4ThreeVector tmppos1 = 0.5*(aStep->GetPreStepPoint()->GetPosition() + aStep->GetPostStepPoint()->GetPosition());
        //      G4ThreeVector tmppos2 = (*cal0Collection)[oldCellId]->GetPos();
        //      cout<<" 2 xpos = "<<tmppos1.x()<<" y = "<<tmppos1.y()<<" z = "<<tmppos1.z()<<endl;
        //      cout<<" 3 xpos = "<<tmppos2.x()<<" y = "<<tmppos2.y()<<" z = "<<tmppos2.z()<<endl;

        // cout<<"*****"
*****"<<endl;

        (*cal0Collection)[oldCellId]->SetTime(atime);
    }
    //      //      if (atimeX < (*cal0Collection)[oldCellId]->GetTimeX()) {
    //      //      //      (*cal0Collection)[oldCellId]->SetTimeX(atimeX);
    //      //      }
    //      //      if (atimeY < (*cal0Collection)[oldCellId]->GetTimeY()) {
    //      //      //      (*cal0Collection)[oldCellId]->SetTimeY(atimeY);
    //      //      }
    //      } // for (int iy = 0; iy<MxStrip; iy++) {
    //      } //for (int ix = 0; ix<MxStrip; ix++) {
    return true;
}

void micalcal0SD::EndOfEvent(G4HCofThisEvent*) {
    // cout<<"EndOfEvent::() {..."<<endl;

    int ntriglay = 4;
    int TrgLayer[4] = {6,7,8,9};
    int TrgDataX[12] = {0,0,0,0,0,0,0,0,0,0,0,0};
    int TrgDataY[12] = {0,0,0,0,0,0,0,0,0,0,0,0};
    int trigStoreX = 0;
    int trigStoreY = 0;
    InCell = 0;

    //Time shift due to propagation of signal in strip
    // double sigXspeed = 0.15*ns; // 5ns/m; 0.15ns/strip
    // double sigYspeed = 0.15*ns; // 5ns/m; 0.15ns/strip

    //20/02/2009 visualisation variables
    int ihst = pAnalysis->ihist;

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 11/28

```

if (pAnalysis->isVisOut>=2) {

    histxmn -= (1/m)*Xstrwd;
    histxmx += (1/m)*Xstrwd;

    histymn -= (1/m)*Ystrwd;
    histymx += (1/m)*Ystrwd;

    histzmn -= (1/m)*parlay[2];
    histzmx += (1/m)*parlay[2];

    int nbinx = int(m*(histxmx - histxmn)/Xstrwd);
    int nbiny = int(m*(histymx - histymn)/Ystrwd);
    int nbinz = int(m*(histzmx - histzmn)/parlay[2]);

    if (ihst < pAnalysis->nhistmx-1 && pAnalysis->isVisOut==2) {
        char name[100];
        sprintf(name, "gens_list_%i", ihst);
        cout <<"name "<<name<<endl;
        pAnalysis->gens_list[0][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

        sprintf(name, "hits_list_%i", ihst);
        pAnalysis->gens_list[1][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

        sprintf(name, "clus_list_%i", ihst);
        pAnalysis->gens_list[2][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

        sprintf(name, "trip_list_%i", ihst);
        pAnalysis->gens_list[3][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

        sprintf(name, "find_list_%i", ihst);
        pAnalysis->gens_list[4][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

        sprintf(name, "fitr_list_%i", ihst);
        pAnalysis->gens_list[5][ihst] = new TH3F(name, name, nbinx, histxmn, histxmx, nbiny, histymn, histymx, nbinz, histzmn, histzmx);

    }
}

cout <<"tmphitlist filled :cal0SD "<< InoHit_list.size()<<endl;
for (unsigned i=0; i<InoHit_list.size(); i++) {
    // cout<<"pAnalysis->hitDist->Fill(InoHit_list["<<i<<"].GetZPlane());"<<
    InoHit_list[i].GetZPlane() <<endl;
    pAnalysis->hitDist->Fill(InoHit_list[i].GetZPlane());
}

InoHit_list.clear();
cout <<"tmphitlist clear "<< endl;//inoHit_pointer->InoHit_list.size()<<G4endl;

int iMnT = 10000; //Should we use these at all ? GMA151001
int iMxT = -10000;
double eMx = 100;
int nHits = 0;

const G4int MxStrip=1; //GMA230621

if (pAnalysis->InputOutput ==3 || pAnalysis->InputOutput ==4) {
    pAnalysis->inputRootFile->cd();
    if (pAnalysis->FirstEvt+pAnalysis->ievent< pAnalysis->inputEventTree->GetEntries()) {
        pAnalysis->inputEventTree->GetEntry(pAnalysis->FirstEvt+pAnalysis->ievent+
+);
    }
}

```

Friday July 16, 2021

micalCal0SD.cc

Jul 16, 21 18:23

micalCal0SD.cc

Page 12/28

```

    } else {
        cout<<"\n Error: Event no. greater than total no. of entries in the input file. \n";
        exit(1);
    }

    for (int rrl=0; rrl<cal0Collection->entries(); rrl++) {
        cout<<rrl<<" ";
        (*cal0Collection)[rrl]->Print();
    }

    cout <<"siminput "<< pAnalysis->nsimht<<endl;
    cout<<"Before loop: "<<cal0Collection->entries()<<endl;
    for (unsigned ij=0; ij<pAnalysis->nsimht; ij++) {
        micalcal0Hit* newHit = new micalcal0Hit();
        G4ThreeVector mom(pAnalysis->simpv[ij], pAnalysis->simv[ij], pAnalysis->simv[ij]);
        G4ThreeVector pos(pAnalysis->simvx[ij], pAnalysis->simvy[ij], pAnalysis->simv[ij]);
        newHit->SetHitId(pAnalysis->detid[ij]);
        newHit->Setpdgid(pAnalysis->simpdgid[ij]);
        newHit->SetEdep(pAnalysis->simenr[ij]);
        newHit->SetTime(pAnalysis->simtime[ij]);

        newHit->SetPos(pos);
        newHit->SetMom(mom);

        newHit->SetLocalXPos(pAnalysis->simlocvx[ij]);
        newHit->SetLocalYPos(pAnalysis->simlocvy[ij]);
        cout<<"ij "<<ij<<" "<<pos<<endl;
        // newHit->Print();
        // cout <<"newhits "<< newHit->GetTime()<<endl;
        cal0Collection->insert(newHit);
    }

    cout<<"cal0Collection->size "<<cal0Collection->entries()<<endl;
    for (int rrl=0; rrl<cal0Collection->entries(); rrl++) {
        cout<<rrl<<" ";
        (*cal0Collection)[rrl]->Print();
    }

    if (pAnalysis->isVisOut==1&&pAnalysis->InputOutput==3) {
        for (unsigned ij=0; ij<pAnalysis->ngent; ij++) {
            pAnalysis->H->NParticles++;
            pAnalysis->Hp = pAnalysis->H->AddHits(0,0); // add a track object
            pAnalysis->Hp->TrackType=-14; // Track Type: -1: hits, -2: clulster, -3: triplet, -4: track -14: particle info
            pAnalysis->Hp->ParCode=pAnalysis->pidin[ij]; // track Number
            //pAnalysis->Hp->ZZ= (7.50 + pAnalysis->poszin[ij]/100 - 0.356)/(0.048*2); // vertex z incase of particle info
            pAnalysis->Hp->ZZ= (((numberInLA*(parirlay[2]+parlay[2])*cm/m-parlay[2]) -pAnalysis->poszin[ij]*cm/m))/((parirlay[2]+parlay[2])*2*(1/m)); // vertex z incase of particle info
            pAnalysis->Hp->XX=pAnalysis->posxin[ij]*cm/m; // vertex x incase of particle info
            pAnalysis->Hp->YY=pAnalysis->posyin[ij]*cm/m; // vertex y incase of particle info
            pAnalysis->Hp->pmag=pAnalysis->momin[ij]; // vertex y incase of particle info
            pAnalysis->Hp->pt=pAnalysis->thein[ij]; // vertex y incase of particle info
            pAnalysis->Hp->pp=pAnalysis->phiin[ij]; // vertex y incase of particle info
        }
    }

    pAnalysis->pRootFile->cd();
}

if (pAnalysis->InputOutput <=4) {
    if (pAnalysis->InputOutput==2) {
        pAnalysis->pRootFile->cd();
        pAnalysis->nsimht = cal0Collection->entries();
        // cout<<" "<<cal0Collection->entries()<<" "<<pAnalysis->nsimht<<endl
    }
}

```

6/14

Jul 16, 21 18:23 **micalCal0SD.cc** Page 13/28

```

;
    if (pAnalysis->nsimht > pAnalysis->nsimhtmx) pAnalysis->nsimht = pAnalysis->
nsimhtmx;
    for (int ij=0; ij< cal0Collection->entries() && ij<(int)pAnalysis->nsimht;
ij++) {
        pAnalysis->detid[ij] = (*cal0Collection)[ij]->GetHitId();
        pAnalysis->simpdgid[ij] = (*cal0Collection)[ij]->Getpdgid();
        pAnalysis->simtime[ij] = (*cal0Collection)[ij]->GetTime();
        pAnalysis->simenr[ij] = (*cal0Collection)[ij]->GetEdep();

        G4ThreeVector posvec1 = (*cal0Collection)[ij]->GetPos();
        pAnalysis->simvx[ij] = posvec1.x();
        pAnalysis->simvy[ij] = posvec1.y();
        pAnalysis->simvz[ij] = posvec1.z();

        G4ThreeVector momvec = (*cal0Collection)[ij]->GetMom();
        pAnalysis->simpx[ij] = momvec.x();
        pAnalysis->simpy[ij] = momvec.y();
        pAnalysis->simpz[ij] = momvec.z();

        pAnalysis->simlocvx[ij] = (*cal0Collection)[ij]->GetLocalXPos();
        pAnalysis->simlocvy[ij] = (*cal0Collection)[ij]->GetLocalYPos();

        if (ij >= (int)pAnalysis->nsimhtmx) break; ; //redundant
    }
    pAnalysis->pEventTree->Fill();
} else {
    int nstripX = int(1.999*pargas[0]/Xstrwd)+1;
    int nstripY = int(1.999*pargas[1]/Ystrwd)+1;
    // cout<<"lcal0Collection->entries() "<<cal0Collection->entries()<<endl;
    for (int ij=0; ij<cal0Collection->entries(); ij++) {
        // GMA Use 90% efficiency for a hit, use poper value
        // float xx = gRandom->Rndm(0);
        // if (xx>0.9) continue;

        // if (corrIneffi < CorrIneffiPar) continue;
        //GMA or use Poission function fo efficiency
        // On the average, need ~24 eV to produce an electron-ion pair

        double edep = (*cal0Collection)[ij]->GetEdep();

        eMx +=edep;

        nHits++;

        // int a4 = gRandom->Poisson(edep/0.024);
        // if (a4 ==0) continue;

        G4ThreeVector posvec2 = (*cal0Collection)[ij]->GetPos();

        if (ihst < pAnalysis->nhistmx-1 && pAnalysis->isVisOut>=2) {
            pAnalysis->gens_list[0][ihst]->Fill((1/m)*posvec2.x(), (1/m)*posvec2.y(
), (1/m)*posvec2.z());
            // cout<<"pAnalysis->gens_list[0][ihst]->Fill();"<<endl;
            vectGr tmpgr;
            tmpgr.x = (1/m)*posvec2.x();
            tmpgr.y = (1/m)*posvec2.y();
            tmpgr.z = (1/m)*posvec2.z();
            tmpgr.dx = 0;
            tmpgr.dy = 0;
            tmpgr.dz = 0;
            if (pAnalysis->isVisOut==3) pAnalysis->gens_vect[0].push_back(tmpgr);
        }

        G4int nInX[MxStrip]={-1}; //, -1, -1; //GMA230621
    }
}

```

Jul 16, 21 18:23 **micalCal0SD.cc** Page 14/28

```

        G4int nInY[MxStrip]={-1}; //, -1, -1;

        unsigned long detid = (*cal0Collection)[ij]->GetHitId();

        // if (gRandom->Rndm(0) > UnCorrYIneffiPar) {nInY[0] = detid%128;} //nIn
Y;
        // detid>=7;
        // if (gRandom->Rndm(0) > UnCorrXIneffiPar) {nInX[0] = detid%128;} //nIn
X;

        nInY[0] = detid%128;
        detid>=7;
        nInX[0] = detid%128;

        if (nInX[0] <0 && nInY[0] <0) { continue;}
        detid>=7;
        int iRPCMod = detid;

        int nInCH = detid%8; //nInCH;
        detid>=3;
        int nInMO = detid%8; //nInMO;
        detid>=3;
        int nInLA = detid%256; //nInLA;
        detid >=8;
        int nInDT = detid%4; //nInDT;

        if(pAnalysis->collatedIn) {
            CorrIneffiPar = pAnalysis->inefficiency_corx[nInLA]->GetBinContent(nIn
X[0]+1,nInY[0]);
        }
        if(gRandom->Rndm(0) < CorrIneffiPar) continue;
        //Gaussian smearing and binning of timing performances
        // GMA 05/02/2009 need to put value from hardware
        int pdgid = (*cal0Collection)[ij]->Getpdgid();
        double atime = (*cal0Collection)[ij]->GetTime();
        // double atimeX = (*cal0Collection)[ij]->GetTimeX();
        // double atimeY = (*cal0Collection)[ij]->GetTimeY();

        // atime +=G4RandGauss::shoot(0,1.0*ns); //Timing resolution is 1ns

        G4double CorrTimeSmr = G4RandGauss::shoot(0,TimeCorrSmr);

        G4double tmpatimeX = atime + SignalSpeed*(nInY[0]+0.5) + CorrTimeSmr; //
+ G4RandGauss::shoot(0,TimeUnCorrSmr);
        G4double tmpatimeY = atime + SignalSpeed*(nInX[0]+0.5) + CorrTimeSmr; //
+ G4RandGauss::shoot(0,TimeUnCorrSmr);

        int nInT = int(atime/TimeToDigiConv); // Assuming Minimum scale of timin
g ~100 ps = 0.1 ns
        if (nInT < iMnT) { iMnT = nInT;}
        if (nInT > iMxT) { iMxT = nInT;}
        G4double gapX = (pargas[0] + (*cal0Collection)[ij]->GetLocalXPos() - nIn
X[0]*Xstrwd)/Xstrwd - 0.5;
        G4double gapY = (pargas[1] + (*cal0Collection)[ij]->GetLocalYPos() - nIn
Y[0]*Ystrwd)/Ystrwd - 0.5;
        int nxmul=1;
        int nymul=1;
        if (nInX[0] >=0 && NewMultiplicity) {
            // cout<<"Hello X World"<<endl;
            if(pAnalysis->collatedIn) {
                nxmul = GetRandomXY(gapX,pAnalysis->strp_xmulsim_cor[nInLA]);
            } else {
                double arand=gRandom->Rndm();
                if (arand<0.1) { //10% case three strip hits
                    nxmul=3;
                } else {
                    if (gRandom->Rndm(0) < 3.2*gapX*gapX) {
                        nxmul=2;
                    } else {
                        nxmul=1;
                    }
                }
            }
        }
    }
}

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 15/28

```

    }
}
if (nxmul==3) { //10% case three strip hits
    nInX[1] = nInX[0] + 1; //int(gapX/abs(max(1.e-12,gapX)));
    nInX[2] = nInX[0] - 1; //int(gapX/abs(max(1.e-12,gapX)));
} else if (nxmul==2) {
    // f(x) = ax**2, => a=3.2
    nInX[1] = nInX[0] + int(gapX/(max(1.e-12,abs(gapX))));
}
} // if (nInX[0] >=0 && NewMultiplicity) {
if (nInY[0] >=0 && NewMultiplicity) {
    if(pAnalysis->collatedIn) {
        nymul = GetRandomXY(gapY,pAnalysis->strp_ymulsim_cor[nInLA]);
    } else {
        double arand=gRandom->Rndm();
        if (arand<0.1) {nymul = 3;}
        else {
            if (gRandom->Rndm(0) < 3.2*gapY*gapY) {nymul=2;}
            else {nymul=1;}
        }
    }
    if (nymul==3) { //10% case three strip hits
        nInY[1] = nInY[0] + 1; //int(gapY/abs(max(1.e-12,gapY)));
        nInY[2] = nInY[0] - 1; //int(gapY/abs(max(1.e-12,gapY)));
    } else if (nymul==2) {
        nInY[1] = nInY[0] + int(gapY/(max(1.e-12,abs(gapY))));
    }
}
}
// if (gRandom->Rndm(0) > UnCorrXIneffiPar) {
for (int ix=0; ix<MxStrip; ix++) {
    if(!NewMultiplicity && ix>0) continue;
    if (nInX[ix] <0 || nInX[ix]>=nstripX) continue;
    if(pAnalysis->collatedIn && nInLA!=5) {
        UnCorrXIneffiPar = pAnalysis->inefficiency_uncx[nInLA]->GetBinContent(nInX[ix]+1,nInY[0]+1);
    }
    if(gRandom->Rndm(0) < UnCorrXIneffiPar) continue;
    double trigeffiX = 0.0;
    if(pAnalysis->collatedIn) {
        trigeffiX = pAnalysis->triggereffi_xevt[nInLA]->GetBinContent(nInX[ix]+1,nInY[0]+1);
        for(int trglx=0; trglx<ntriglay; trglx++) {
            if((nInLA == TrgLayer[trglx]) && (G4UniformRand()<(trigeffiX))) {
                TrgDataX[TrgLayer[trglx]]++;
            }
        }
    } // if(pAnalysis->collatedIn) {
G4double atimeX = tmpatimeX + G4RandGauss::shoot(0,TimeUnCorrSmr);
int nInTX = int(atimeX/TimeToDigiConv);
int iold = 0;

for (unsigned jk=0; jk<inoStripX_pointer->InoStripX_list.size(); jk++)
{
    InoStrip* Xstrip =inoStripX_pointer->InoStripX_list[jk];
    if (Xstrip->GetRPCMod()==iRPCMod &&
        Xstrip->GetStrip()%numberInX==nInX[ix]) {
        inoStripX_pointer->InoStripX_list[jk]->AddPulse(edep); //GMA151001
        for large multiplcty share this energy
        if (inoStripX_pointer->InoStripX_list[jk]->GetSmrTime() >nInTX) {
            inoStripX_pointer->InoStripX_list[jk]->SetSmrTime(nInTX);
        }
        iold = 1; break;
    }
}
}

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 16/28

```

//GMA Take precaution of these one strip hits
// 1. Segement direction, for X/Y Z-value might be different
// consequently direction
// 2.
//
//
if (iold==0) {
    // cout <<"ixx "<< ix<<" "<<nInX[ix]<<endl;
    InoStrip * Xstrip = new InoStrip(); //VALGRIND
    Xstrip->SetStrip(numberInX*numberInMO*nInDT+numberInX*nInMO+nInX[ix]);
};

Xstrip->SetpdgId( pdgid);
Xstrip->SetTrueTime(nInT);
Xstrip->SetSmrTime(nInTX); //nInT

cout <<"xtime "<< nInT<<" "<<nInTX<<" "<<Xstrip->GetTrueTime()<<" "<<Xstrip->GetSmrTime()<<" "<<pdgid<<" "<<Xstrip->GetpdgId()<<endl;

Xstrip->SetPulse(edep);
Xstrip->SetRPCMod(iRPCMod);

G4ThreeVector trkmom = (*cal0Collection)[ij]->GetMom();

Xstrip->SetMomentum(trkmom.mag());
Xstrip->SetTheta(trkmom.theta());
Xstrip->SetPhi(trkmom.phi());

G4ThreeVector posvec3 = (*cal0Collection)[ij]->GetPos();
// cout<<"X_posvec3 "<<posvec3<<endl;
Xstrip->SetGenPosX(posvec3.x());
Xstrip->SetGenPosY(posvec3.y());
Xstrip->SetGenPosZ(posvec3.z());

G4int xstripid = 0;
xstripid<=2;
xstripid +=nInDT;

xstripid<=8;
xstripid +=nInLA;

xstripid<=3;
xstripid +=nInMO;

xstripid<=3;
xstripid +=nInCH;

xstripid<=7;
xstripid +=nInX[ix];

xstripid<=5;
xstripid +=0; //nInT;

xstripid<=3;
xstripid +=TMath::Min(int(edep/16),7);

Xstrip->SetId(xstripid);

// cout <<"iold = "<<iold<<"", nInLA = "<<nInLA<<"", nInX =
"<<nInX[ix]<<endl;
pAnalysis->DeadStripX->Fill(nInX[ix]);
// cout<<"pAnalysis->DeadStripX->Fill(nInX);="<<nInX[ix]<
<endl;
inoStripX_pointer->InoStripX_list.push_back(Xstrip);
} // if (iold==0)
} // for (int ix=0; ix<MxStrip; ix++)

// if (gRandom->Rndm(0) > UnCorrYIneffiPar) {

```


Jul 16, 21 18:23

micalCal0SD.cc

Page 17/28

```

for (int jy=0; jy<MxStrip; jy++) {
    if(!NewMultiplicity && jy>0) continue;
    if (nInY[jy] <0 || nInY[jy]>=nstripY) continue;

    if(pAnalysis->collatedIn && nInLA!=5) {
        UnCorrYIneffiPar = pAnalysis->inefficiency_uncy[nInLA]->GetBinContent(nInX[0]+1,nInY[jy]+1);
    }
    if(gRandom->Rndm(0) < UnCorrYIneffiPar) continue;

    double trigeffiY = 0.0;
    if(pAnalysis->collatedIn) {
        trigeffiY = pAnalysis->triggereffi_yevt[nInLA]->GetBinContent(nInX[0]+1,nInY[jy]+1);
        for(int trgly=0; trgly<ntriglay; trgly++) {
            if((nInLA == TrgLayer[trgly]) && (G4UniformRand()<(trigeffiY))) {
                TrgDataY[TrgLayer[trgly]]++;
            }
        }
    } // if(pAnalysis->collatedIn) {

    G4double atimeY = tmpatimeY + G4RandGauss::shoot(0,TimeUnCorrSmr);
    int nInTY = int(atimeY/TimeToDigiConv);

    int iold = 0;
    for (unsigned jk=0; jk<inoStripY_pointer->InoStripY_list.size(); jk++)
    {
        InoStrip* Ystrip =inoStripY_pointer->InoStripY_list[jk];
        if (Ystrip->GetRPCmod()==iRPCMod &&
            Ystrip->GetStrip()>numberInY==nInY[jy]) {
            inoStripY_pointer->InoStripY_list[jk]->AddPulse(edep);
            if (inoStripY_pointer->InoStripY_list[jk]->GetSmrTime() >nInTY) {
                inoStripY_pointer->InoStripY_list[jk]->SetSmrTime(nInTY);
            }
            iold = 1; break;
        }
    }
    if (iold==0) {
        InoStrip* Ystrip = new InoStrip(); //VALGRIND
        Ystrip->SetStrip(numberInY*nInCH+nInY[jy]);

        Ystrip->Setpdgid(pdgid);
        Ystrip->SetTrueTime(nInT);
        Ystrip->SetSmrTime(nInTY);
        // cout <<"ytime "<< nInT<<" "<<nInTY<<" "<<Ystrip->GetTrueTime()<<"
        "<<Ystrip->GetSmrTime()<<" "<<pdgid<<" "<<Ystrip->Getpdgid()<<endl;

        Ystrip->SetPulse(edep);
        Ystrip->SetRPCmod(iRPCMod);
        G4ThreeVector trkmom = (*cal0Collection)[ij]->GetMom();

        Ystrip->SetMomentum(trkmom.mag());
        Ystrip->SetTheta(trkmom.theta());
        Ystrip->SetPhi(trkmom.phi());

        G4ThreeVector posvec = (*cal0Collection)[ij]->GetPos();
        // cout<<"Y_posvec "<<posvec<<endl;

        Ystrip->SetGenPosX(posvec.x());
        Ystrip->SetGenPosY(posvec.y());
        Ystrip->SetGenPosZ(posvec.z());

        G4int ystripid = 1;

        ystripid<=2;
        ystripid +=nInDT;

        ystripid<=8;
        ystripid +=nInLA;

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 18/28

```

        ystripid<=3;
        ystripid +=nInMO;

        ystripid<=3;
        ystripid +=nInCH;

        ystripid<=7;
        ystripid +=nInY[jy];

        ystripid<=5;
        ystripid +=0; //05/01/2009 nInT;

        ystripid<=3;
        ystripid +=TMath::Min(int(edep/16),7);

        Ystrip->SetId(ystripid);

        // cout <<"iold = "<<iold<<"", nInLA = "<<nInLA<<"", nInY =
        "<<nInY[jy]<<endl;
        pAnalysis->DeadStripY->Fill(nInY[jy]);
        // cout<<"pAnalysis->DeadStripY->Fill(nInY[jy]);"<<nInY<<endl;
        inoStripY_pointer->InoStripY_list.push_back(Ystrip);
    } // if (iold==0)
    } //for (int jy=0; jy<MxStrip; jy++)
    // } //if (gRandom->Rndm(0) > UnCorrYIneffiPar)
    // cout<<"hihihi "<<ij<<endl;
    } //for (int ij=0; ij<cal0Collection->entries(); ij++)
    //Add noise hits
    //GMA use proper noise hits
    // Assume total noise hits is 100 in whole detector
    // cout <<"Size2 "<< inoStripX_pointer->InoStripX_list.size()<<" "<<inoStr
    ipY_pointer->InoStripY_list.size()<<endl;

    double genxpos=pAnalysis->posxin[0]; //Position of neutrino vertex
    double genypos=pAnalysis->posyin[0];
    double genzpos=pAnalysis->poszin[0];

    int igenchm = ((genypos+8.0)/2.0); //Y-direction
    int igenlay = ((genzpos+7.5)/0.096);
    int igendt = (genxpos+24.0)/16.0;
    int igenmod = ((genxpos+24.0)/2.0)-8*igendt;

    for (int ij=0; ij<RandomNoisePar; ij++) { //RANDOM
        const int nrandom = 15;
        float randvar[nrandom];
        gRandom->RndmArray(nrandom, randvar);

        // int nInT = min(int(4*randvar[0]), 3);
        //GMA 05/02/2009
        int nInT = int((iMxT-iMnT+200)*randvar[0])-100; //+-10ns within actual
        hits

        double atime = TimeToDigiConv*nInT;

        G4double CorrTimeSmr = G4RandGauss::shoot(0,TimeCorrSmr);
        G4double atimeX = atime + CorrTimeSmr + G4RandGauss::shoot(0,TimeUnCorrS
        mr);

        G4double atimeY = atime + CorrTimeSmr + G4RandGauss::shoot(0,TimeUnCorrS
        mr);

        int nInY = min(int(nstripY*randvar[1]),nstripY-1);
        int nInX = min(int(nstripX*randvar[2]),nstripX-1);

        int nInTXX = int((atimeX+SignalSpeed*(nInY+0.5))/TimeToDigiConv);
        int nInTTY = int((atimeY+SignalSpeed*(nInX+0.5))/TimeToDigiConv);

        // int nInCH = min(int(numberInCH*randvar[3]),numberInCH-1);
        // int nInMO = min(int(numberInMO*randvar[4]),numberInMO-1);
        // int nInLA = min(int(numberInLA*randvar[5]),numberInLA-1);

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 19/28

```

// int nInDT = int(3*randvar[7]);

int nInCH = int(igenchm + 3*(randvar[3]-0.5)+0.5);
int nInMO = int(igenmod + 3*(randvar[4]-0.5)+0.5);
int nInLA = int(igenlay + 40*(randvar[5]-.5)+0.5);
int nInDT = igendt;

if ((nInCH<0 || nInCH >=numberInCH) ||
    (nInMO<0 || nInMO >=numberInMO) ||
    (nInLA<0 || nInLA >=numberInLA)) continue;

float edep = gRandom->Exp(eMx/max(1,nHits)); //Exponential distribution

int ihitxy=2; //both hit
if (randvar[8]<0.25) {ihitxy=0;} else if (randvar[8]<0.50) {ihitxy=1;}

int iRPCMod = nInDT;
iRPCMod<=8;
iRPCMod +=nInLA;
iRPCMod<=3;
iRPCMod +=nInMO;
iRPCMod<=3;
iRPCMod +=nInCH;
int ioldx = 0;

for (unsigned jk=0; jk<inoStripX_pointer->InoStripX_list.size(); jk++) {
    InoStrip* Xstrip =inoStripX_pointer->InoStripX_list[jk];
    if (Xstrip->GetRPCmod()==iRPCMod &&
        Xstrip->GetStrip() %numberInX==nInX) {
        inoStripX_pointer->InoStripX_list[jk]->AddPulse(edep);
        if (inoStripX_pointer->InoStripX_list[jk]->GetSmrTime() >nInTXX) {
            inoStripX_pointer->InoStripX_list[jk]->SetSmrTime(nInTXX);
        }
        ioldx = 1; break;
    }
}
int ioldy = 0;
for (unsigned jk=0; jk<inoStripY_pointer->InoStripY_list.size(); jk++) {
    InoStrip* Ystrip =inoStripY_pointer->InoStripY_list[jk];
    if (Ystrip->GetRPCmod()==iRPCMod &&
        Ystrip->GetStrip() %numberInY==nInY) {
        inoStripY_pointer->InoStripY_list[jk]->AddPulse(edep);
        if (inoStripY_pointer->InoStripY_list[jk]->GetSmrTime() >nInTTY) {
            inoStripY_pointer->InoStripY_list[jk]->SetSmrTime(nInTTY);
        }
        ioldy = 1; break;
    }
}
//
// GMA Noise hits may be correlated, may not be
// Here used equal distribution of correlated and uncorrelated hits
// Use proper value from detector information

if (ihitxy%2==0 && ioldx==0) {
    InoStrip* Xstrip = new InoStrip();
    Xstrip->SetStrip(numberInX*numberInMO*nInDT+numberInX*nInMO+nInX);
    Xstrip->SetRPCmod(iRPCMod);

    Xstrip->SetTrueTime(nInT);
    Xstrip->SetSmrTime(nInTXX);
    Xstrip->SetPulse(edep);

    G4int xstripid = 0;
    xstripid<=2;
    xstripid +=nInDT;

    xstripid<=8;
    xstripid +=nInLA;

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 20/28

```

xstripid<=3;
xstripid +=nInMO;

xstripid<=3;
xstripid +=nInCH;

xstripid<=7;
xstripid +=nInX;

xstripid<=5;
xstripid +=0; // nInT;

xstripid<=3;
xstripid +=TMath::Min(int(edep/16),7);

Xstrip->SetId(xstripid);

inoStripX_pointer->InoStripX_list.push_back(Xstrip);
} // if (ihitxy%2==0 && ioldx==0)

if (ihitxy>0 && ioldy==0) {
    InoStrip* Ystrip = new InoStrip();
    Ystrip->SetStrip(numberInY*nInCH+nInY);
    Ystrip->SetRPCmod(iRPCMod);
    Ystrip->SetTrueTime(nInT);
    Ystrip->SetSmrTime(nInTTY);
    Ystrip->SetPulse(edep);

    G4int ystripid = 1;

    ystripid<=2; //1;
    ystripid +=nInDT;

    ystripid<=8;
    ystripid +=nInLA;

    ystripid<=3;
    ystripid +=nInMO;

    ystripid<=3;
    ystripid +=nInCH;

    ystripid<=7;
    ystripid +=nInY;

    ystripid<=5;
    ystripid +=0; // nInT;

    ystripid<=3;
    ystripid +=TMath::Min(int(edep/16),7);

    Ystrip->SetId(ystripid);

    inoStripY_pointer->InoStripY_list.push_back(Ystrip);
} // if (ihitxy>0 && ioldy==0)
} //for (int ij=0; ij<100; ij++)
// cout <<"Size1 "<< inoStripX_pointer->InoStripX_list.size()<<" "<<inoStr
ipY_pointer->InoStripY_list.size()<<endl;
//All these %tage and [5] should be either from database of thorough messan
ger class
//2% time will have five consecutive strip hit
//5% time will every eighth strip
//3% time correlated noise in X/Y Strips
const int nConseStr=4;
vector<InoStrip*> tmp_striplist;
int nsizeX = inoStripX_pointer->InoStripX_list.size();
int nsizeY = inoStripY_pointer->InoStripY_list.size();
for (int ixy=0; ixy<2; ixy++) {

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 21/28

```

tmp_striplist.clear();
if (ixy==0) {
    for (int ix=0; ix<nsize; ix++) {
        tmp_striplist.push_back(inoStripX_pointer->InoStripX_list[ix]);
    }
} else {
    for (int ix=0; ix<nsize; ix++) {
        tmp_striplist.push_back(inoStripY_pointer->InoStripY_list[ix]);
    }
}

for (unsigned int jk=0; jk<tmp_striplist.size(); jk++) {
    InoStrip* Xstrip = tmp_striplist[jk];
    int nInX[nConseStr]={-1,-1,-1,-1};
    int nNoise=0;
    double xrnd = gRandom->Rndm(0);

    int idetid = tmp_striplist[jk]->GetId();
    int istrip = ((idetid>>8)&0x7f);
    double time = tmp_striplist[jk]->GetSmrTime();

    if (xrnd < CorrNoisePar1) {
        nNoise=4;
        nInX[0] = istrip+1;
        nInX[1] = istrip+2;
        nInX[2] = istrip-1;
        nInX[3] = istrip-2;
        for (int ix=0; ix<nConseStr; ix++) {
            if (nInX[ix] > (nstripX-1)) { nInX[ix] -=nConseStr; }
            if (nInX[ix] < 0 ) {nInX[ix] +=nConseStr; }
        }
    } else if (xrnd< CorrNoisePar1+CorrNoisePar2) {
        nNoise=3;
        int irem=istrip%8;
        int iint=int(istrip/32);
        int counter[4]={irem+32*iint, irem+8+32*iint, irem+16+32*iint, irem+
24+32*iint};

        nNoise =0;
        for (int ix=0; ix<nNoise+1; ix++) {
            if (istrip!=counter[ix]) {
                nInX[nNoise] = counter[ix];
                nNoise++;
            }
        }
    } else if (xrnd < CorrNoisePar1+CorrNoisePar2+CorrNoisePar3) {
        nNoise =1;
        nInX[0] = -istrip;
    }

    for (int ix=0; ix<nNoise; ix++) {
        InoStrip* xstrip = new InoStrip(Xstrip);

        xstrip->SetSmrTime(time+G4RandGauss::shoot(0,TimeUnCorrSmr));
        xstrip->SetPulse(-100.0);
        if (nInX[ix]>=0) {
            unsigned int tmpid = idetid;
            int iext =tmpid&0xff;
            tmpid>>=15;
            //          int strp = tmpid&0x7f;
            //          tmpid>>=7;
            tmpid<<=7;
            tmpid +=nInX[ix];
            tmpid<<=8;
            tmpid +=iext;
            xstrip->SetId(tmpid);
            if (ixy==0) {
                inoStripX_pointer->InoStripX_list.push_back(xstrip);
            } else {

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 22/28

```

        inoStripY_pointer->InoStripY_list.push_back(xstrip);
    }
} else { //Y correlation
    unsigned int tmpid = idetid + (ixy==0) ? twopow31 : -twopow31;
    xstrip->SetId(tmpid);
    if (ixy==0) {
        inoStripY_pointer->InoStripY_list.push_back(xstrip);
    } else {
        inoStripX_pointer->InoStripX_list.push_back(xstrip);
    }
}
} // for (int ix=0; ix<nNoise; ix++)
} // for (int jk=0; jk<nsize; jk++)
}
tmp_striplist.clear();

for(int trgi=0; trgi<numberInLA; trgi++) {
    if (TrgDataX[trgi]>0) {
        trigStoreX++;
    }
    if (TrgDataY[trgi]>0) {
        trigStoreY++;
    }
} // for(int trgi=0; trgi<numberInLA; trgi++) {
} //else of if (pAnalysis->InputOutput==2)

cout <<"Size "<< inoStripX_pointer->InoStripX_list.size()<<" "<<inoStripY_poi
nter->InoStripY_list.size()<<endl;

if (pAnalysis->InputOutput==1 || pAnalysis->InputOutput==4) {
    pAnalysis->pRootFile->cd();
    pAnalysis->trigx = trigStoreX;
    pAnalysis->trigy = trigStoreY;
    pAnalysis->ndigih = inoStripX_pointer->InoStripX_list.size()
+ inoStripY_pointer->InoStripY_list.size();
    if (pAnalysis->ndigih >pAnalysis->ndigihmx) pAnalysis->ndigih =pAnalysi
s->ndigihmx;
    for (unsigned ij=0; ij<inoStripX_pointer->InoStripX_list.size() && ij<pAna
lysis->ndigih ; ij++) {
        pAnalysis->stripid[ij] =inoStripX_pointer->InoStripX_list[ij]->GetId();
        pAnalysis->digipdgid[ij] =inoStripX_pointer->InoStripX_list[ij]->GetpdgI
d();
        pAnalysis->digitime[ij] = inoStripX_pointer->InoStripX_list[ij]->GetSmrT
ime();
        pAnalysis->digitruetime[ij] = inoStripX_pointer->InoStripX_list[ij]->Get
TrueTime();
        pAnalysis->digienr[ij] =inoStripX_pointer->InoStripX_list[ij]->GetPulse(
);
        pAnalysis->digivx[ij] =inoStripX_pointer->InoStripX_list[ij]->GetGenPosX
();
        pAnalysis->digivy[ij] =inoStripX_pointer->InoStripX_list[ij]->GetGenPosY
();
        pAnalysis->digivz[ij] =inoStripX_pointer->InoStripX_list[ij]->GetGenPosZ
();

        G4ThreeVector trkmom(1000,1,1);
        trkmom.setMag(inoStripX_pointer->InoStripX_list[ij]->GetMomentum());
        trkmom.setTheta(inoStripX_pointer->InoStripX_list[ij]->GetTheta());
        trkmom.setPhi(inoStripX_pointer->InoStripX_list[ij]->GetPhi());

        pAnalysis->digipx[ij] = trkmom.x();
        pAnalysis->digipy[ij] = trkmom.y();
        pAnalysis->digipz[ij] = trkmom.z();
        if (ij >=pAnalysis->ndigihmx) break; //redundant

        cout<<"ij "<<ij<<" "<<pAnalysis->digivx[ij]<<" "<<pAnalysis->digivy[ij]<<"
"<<pAnalysis->digivz[ij]<<" "<<(pAnalysis->stripid[ij]>>8)<<endl;
    }
}

```

Jul 16, 21 18:23	micalCal0SD.cc	Page 23/28
<pre> unsigned jk = inoStripX_pointer->InoStripX_list.size(); for (unsigned ij=0; ij<inoStripY_pointer->InoStripY_list.size() && jk< pAnalysis->ndigihit; ij++, jk++) { pAnalysis->stripid[jk] = inoStripY_pointer->InoStripY_list[ij]->GetId(); pAnalysis->digipdgid[jk] = inoStripY_pointer->InoStripY_list[ij]->GetpdgId(); pAnalysis->digitime[jk] = inoStripY_pointer->InoStripY_list[ij]->GetSmrTime(); pAnalysis->digitruetime[jk] = inoStripY_pointer->InoStripY_list[ij]->GetTrueTime(); pAnalysis->digienr[jk] = inoStripY_pointer->InoStripY_list[ij]->GetPulse(); pAnalysis->digivx[jk] = inoStripY_pointer->InoStripY_list[ij]->GetGenPosX(); pAnalysis->digivy[jk] = inoStripY_pointer->InoStripY_list[ij]->GetGenPosY(); pAnalysis->digivz[jk] = inoStripY_pointer->InoStripY_list[ij]->GetGenPosZ(); G4ThreeVector trkmom(1000,0,0); trkmom.setMag(inoStripY_pointer->InoStripY_list[ij]->GetMomentum()); trkmom.setTheta(inoStripY_pointer->InoStripY_list[ij]->GetTheta()); trkmom.setPhi(inoStripY_pointer->InoStripY_list[ij]->GetPhi()); pAnalysis->digipx[jk] = trkmom.x(); pAnalysis->digipy[jk] = trkmom.y(); pAnalysis->digipz[jk] = trkmom.z(); if (jk >= pAnalysis->ndigihitmx) break; //redundant cout<<"jk "<<jk<<" "<<pAnalysis->digivx[jk]<<" "<<pAnalysis->digivy[jk]<<" "<<pAnalysis->digivz[jk]<<" "<<(pAnalysis->stripid[jk]>>8)<<endl; } cout<<"digioutput "<<pAnalysis->ndigihit<<" "<<pAnalysis->trigx<<" "<<pAnalysis->trigy<<endl; pAnalysis->pEventTree->Fill(); } else { //read diginint file //if (pAnalysis->InputOutput <=4) pAnalysis->inputRootFile->cd(); if(pAnalysis->FirstEvt+pAnalysis->ievent< pAnalysis->inputEventTree->GetEntries()){ pAnalysis->inputEventTree->GetEntry(pAnalysis->FirstEvt+pAnalysis->ievent+1); } else { cout<<"\n Error: Event no. greater than total no. of entries in the input file.\n"; exit(1); } for(unsigned ij=0;ij<pAnalysis->ngent;ij++) { if (pAnalysis->isVisOut==1) { pAnalysis->H->NParticles++; pAnalysis->Hp= pAnalysis->H->AddHits(0,0); // add a track object //VALGRIND pAnalysis->Hp->TrackType=-14; // Track Type: -1: hits, -2: clulster, -3: triplet, -4: track -14: particle info pAnalysis->Hp->ParCode=pAnalysis->pidin[ij]; // track Number //pAnalysis->Hp->ZZ= (7.50 + pAnalysis->poszin[ij]/100 - 0.356)/(0.048*2); // vertex z incase of particle info pAnalysis->Hp->ZZ= ((numberInLA*(parirlay[2]+parlay[2])*cm/m-parlay[2])-(pAnalysis->poszin[ij]*cm/m))/((parirlay[2]+parlay[2])*2*(1/m)); // vertex z incase of particle info pAnalysis->Hp->XX=pAnalysis->posxin[ij]*cm/m; // vertex x incase of particle info pAnalysis->Hp->YY=pAnalysis->posyin[ij]*cm/m; // vertex y incase of particle info pAnalysis->Hp->pmag=pAnalysis->momin[ij]; // vertex y incase of particle info pAnalysis->Hp->pt=pAnalysis->thein[ij]; // vertex y incase of particle info pAnalysis->Hp->pp=pAnalysis->phiin[ij]; // vertex y incase of particle info </pre>		

Jul 16, 21 18:23	micalCal0SD.cc	Page 24/28
<pre> nfo } } //----- for(unsigned ij=0;ij<pAnalysis->ndigihit;ij++) { unsigned istrp = pAnalysis->stripid[ij]; InoStrip* Xstrip = new InoStrip(); //VALGRIND Xstrip->SetId(istrp); Xstrip->SetpdgId(pAnalysis->digipdgid[ij]); Xstrip->SetSmrTime(pAnalysis->digitime[ij]); Xstrip->SetTrueTime(pAnalysis->digitruetime[ij]); Xstrip->SetPulse(pAnalysis->digienr[ij]); G4ThreeVector tmp3v(pAnalysis->digipx[ij], pAnalysis->digipy[ij], pAnalysis->digipz[ij]); Xstrip->SetMomentum(tmp3v.mag()); Xstrip->SetTheta(tmp3v.theta()); Xstrip->SetPhi(tmp3v.phi()); Xstrip->SetGenPosX(pAnalysis->digivx[ij]); Xstrip->SetGenPosY(pAnalysis->digivy[ij]); Xstrip->SetGenPosZ(pAnalysis->digivz[ij]); cout<<"G_Position "<<ij<<" "<< pAnalysis->digivx[ij]<<" "<< pAnalysis->digivy[ij]<<" "<< pAnalysis->digivz[ij]<<endl; if ((istrp>>31)==0) { //Most significant bit is X/Y inoStripX_pointer->InoStripX_list.push_back(Xstrip); } else { inoStripY_pointer->InoStripY_list.push_back(Xstrip); } } pAnalysis->pRootFile->cd(); } //if (pAnalysis->InputOutput <=4) //Fill up positions, layer etc of all strips from stripid for futher use if (pAnalysis->InputOutput==0 pAnalysis->InputOutput ==3 pAnalysis->InputOutput==5) { cout<<" "<<endl<<".....Entered Reconstruction....."<<endl; double ShiftInX = paradev->GetINOrroomPos(0) + paradev->GetStackPosInRoom(0) + paradev->GetShiftInX(); double ShiftInY = paradev->GetINOrroomPos(1) + paradev->GetStackPosInRoom(1) + paradev->GetShiftInY(); double ShiftInZ = paradev->GetINOrroomPos(2) + paradev->GetStackPosInRoom(2) + paradev->GetShiftInZ(0); // ShiftInX = paradev->GetShiftInX(); // ShiftInY = paradev->GetShiftInY(); // ShiftInZ = paradev->GetShiftInZ(); // cout<<"X Side ShiftInXYZ "<<ShiftInX<<" "<<ShiftInY<<" "<<ShiftInZ<<endl; // cout<<inoStripX_pointer->InoStripX_list.size()<<endl; for (unsigned ij=0; ij<inoStripX_pointer->InoStripX_list.size(); ij++) { unsigned istrp = inoStripX_pointer->InoStripX_list[ij]->GetId(); double energy = istrp%8; istrp >>=8; int nInX = istrp%128; istrp >>=7; int iRPCMod = istrp%65536; // 2**16 int nInCH = istrp%8; istrp >>=3; int nInMO = istrp%8; istrp >>=3; int nInLA = istrp%256; </pre>		

Jul 16, 21 18:23	micalCal0SD.cc	Page 25/28
<pre> istrp>=8; int nInDT = istrp%4; istrp>=2; inoStripX_pointer->InoStripX_list[ij]->SetPlaneView(istrp); inoStripX_pointer->InoStripX_list[ij]->SetPlane(nInLA); inoStripX_pointer->InoStripX_list[ij]->SetRPCMod(iRPCMod); inoStripX_pointer->InoStripX_list[ij]->SetStrip(numberInX*numberInMO*nInDT +numberInX*nInMO+nInX); double xpos = (1/m)*(-pargas[0] + Xstrwd*(nInX+0.5) + ShiftInX); // double xpos = (1/m)*((nInDT-1)*(2*parino[0]+gapino) - parlay[0]] + (2*nInMO+1)*parmod[0] -pargas[0] + Xstrwd*(nInX+0.5) + ShiftInX); double ypos = (1/m)*(ShiftInY + (paradef->GetnStack())> 1) ? (2*nInCH-1)*parchm[1] : 0) ; //double ypos = (1/m)*(-parino[2] + 2*(parhcoil[2]+parcoilsupport[2]) + 2* (nInLA+1)*parirlay[2] + (2*nInLA+1)*(parlay[2])); double ypos = (1/m)*(-(numberInLA-1)*(parirlay[2]+parlay[2]))+(nInLA)*2*(pa rirlay[2] + parlay[2]) + ShiftInZ); //AAR:** changes for Central Iron Layer ** //if smearing //RANDOM inoStripX_pointer->InoStripX_list[ij]->SetXYPos(xpos); inoStripX_pointer->InoStripX_list[ij]->SetZPos(zpos); // cout<<" X "<<nInLA<<" "<<ij<<" nInX "<<nInX<<" x pos "<<1000*inoStripX_pointer->InoStripX_list[ij]->GetXYPos()<<" "<<inoStripX_po inter->InoStripX_list[ij]->GetGenPosX()<<" diffX "<<1000*xpos -inoStripX_pointer ->InoStripX_list[ij]->GetGenPosX()<<" zpos "<<1000*inoStripX_pointer->InoStripX_ list[ij]->GetZPos()<<" "<<inoStripX_pointer->InoStripX_list[ij]->GetGenPosZ()<<" diffZ "<<1000*zpos - inoStripX_pointer->InoStripX_list[ij]->GetGenPosZ()<<endl; pAnalysis->pPosX->Fill(100*xpos - 0.1*inoStripX_pointer->InoStripX_list[ij]->GetGenPosX()); pAnalysis->pPosZ->Fill(100*zpos - 0.1*inoStripX_pointer->InoStripX_list[ij]->GetGenPosZ()); pAnalysis->pPosXX->Fill(100*xpos, 100*xpos - 0.1*inoStripX_pointer->InoStr ipX_list[ij]->GetGenPosX()); pAnalysis->pPosZZ->Fill(100*zpos, 100*zpos - 0.1*inoStripX_pointer->InoStr ipX_list[ij]->GetGenPosZ()); // cout<<"pAnalysis->(pPosX,pPosZ,pPosXX,pPosZZ)->Fill();"<<endl; if (energy >100000 abs(ypos)>100000) cout <<"ypos "<<ypos<<" "<<energy<<e ndl; } // cout<<inoStripY_pointer->InoStripY_list.size()<<endl; for (unsigned ij=0; ij<inoStripY_pointer->InoStripY_list.size(); ij++) { unsigned istrp = inoStripY_pointer->InoStripY_list[ij]->GetId(); double energy = istrp%8; istrp >=8; int nInY = istrp%128; istrp>=7; int iRPCMod = istrp%65536; // 2**16 int nInCH = istrp%8; istrp>=3; int nInMO = istrp%8; istrp>=3; int nInLA = istrp%256; istrp>=8; int nInDT = istrp%4; istrp>=2; inoStripY_pointer->InoStripY_list[ij]->SetPlaneView(istrp); inoStripY_pointer->InoStripY_list[ij]->SetPlane(nInLA); inoStripY_pointer->InoStripY_list[ij]->SetRPCMod(iRPCMod); inoStripY_pointer->InoStripY_list[ij]->SetStrip(numberInY*nInCH+nInY); </pre>		

Jul 16, 21 18:23	micalCal0SD.cc	Page 26/28
<pre> double xpos = (1/m)*(ShiftInX); // double xpos = (1/m)*((nInDT-1)*(2*parino[0]+gapino) - parlay[0]] + (2*nInMO+1)*parmod[0] + ShiftInX); double shift2y = (paradef->GetnStack())>1) ? (2*nInCH-1)* parchm[1] : 0; //GMA one/two RPC in a layer double ypos = (1/m)*(shift2y -pargas[1] + Ystrwd*(nInY+0.5) + ShiftInY); // double ypos = (1/m)*(- parmod[1] + (2*nInCH+1)*parchm[1] -parga s[1] + Ystrwd*(nInY+0.5) + ShiftInY); double ypos = (1/m)*(-(numberInLA-1)*(parirlay[2]+parlay[2]))+(nInLA)*2*(pa rirlay[2] + parlay[2]) + ShiftInZ); //AAR:** changes for Central Iron Layer ** inoStripY_pointer->InoStripY_list[ij]->SetXYPos(ypos); inoStripY_pointer->InoStripY_list[ij]->SetZPos(zpos); // cout<<" Y "<<nInLA<<" "<<ij<<" ypos "<<1000*ino StripY_pointer->InoStripY_list[ij]->GetXYPos()<<" "<<inoStripY_pointer->InoStrip Y_list[ij]->GetGenPosY()<<" diffY "<<1000*ypos - inoStripY_pointer->InoStripY_li st[ij]->GetGenPosY()<<" zpos "<<100*inoStripY_pointer->InoStripY_list[ij]->GetZP os()<<" "<<inoStripY_pointer->InoStripY_list[ij]->GetGenPosZ()<<" diffZ "<<1000* zpos - inoStripY_pointer->InoStripY_list[ij]->GetGenPosZ()<<endl; pAnalysis->pPosY->Fill(100*ypos - 0.1*inoStripY_pointer->InoStripY_list[ij]->GetGenPosY()); pAnalysis->pPosZ->Fill(100*zpos - 0.1*inoStripY_pointer->InoStripY_list[ij]->GetGenPosZ()); pAnalysis->pPosYY->Fill(100*ypos, 100*ypos - 0.1*inoStripY_pointer->InoStr ipY_list[ij]->GetGenPosY()); pAnalysis->pPosZZ->Fill(100*zpos, 100*zpos - 0.1*inoStripY_pointer->InoStr ipY_list[ij]->GetGenPosZ()); // cout<<"pAnalysis->(pPosY,pPosZ,pPosYY,pPosZZ)->Fill();"<<endl; // cout<<"micalcal0SD::EndOfEvent(G4HCofThisEvent*) {....}"<<endl; if (energy >100000 abs(xpos)>100000) cout <<"xpos "<<xpos<<" "<<energy<<e ndl; } // cout<<"...}EndofEvent() "<<endl; } void micalcal0SD::clear() { } void micalcal0SD::DrawAll() { } void micalcal0SD::PrintAll() { } void micalcal0SD::SetCorrTimeSmear(G4double val) { cout<<"void micalcal0SD::SetCorrTimeSmear(G4double "<<val<<)"<<endl; // cout<<"CorrTimeSmear = "<<val<<endl; // cout<<"...}"<<endl; TimeCorrSmr = val; pAnalysis->SetCorrTimeError(val); } void micalcal0SD::SetUnCorrTimeSmear(G4double val) { cout<<"void micalcal0SD::SetUnCorrTimeSmear(G4double "<<val<<)"<<endl; // cout<<"UnCorrTimeSmear = "<<val<<endl; // cout<<"...}"<<endl; TimeUnCorrSmr = val; pAnalysis->SetUnCorrTimeError(val); } void micalcal0SD::SetCorrInefficiency(G4double val) { </pre>		

Jul 16, 21 18:23

micalCal0SD.cc

Page 27/28

```

cout<<"void micalcal0SD::SetCorrInefficiency(G4double "<<val<<")<<endl;
// cout<<"CorrInefficiency = "<<val<<endl;
// cout<<"..."<<endl;
CorrIneffiPar = val;
}

void micalcal0SD::SetUnCorrXInefficiency(G4double val) {
cout<<"void micalcal0SD::SetUnCorrXInefficiency(G4double "<<val<<")<<endl;
// cout<<"UnCorrXInefficiency = "<<val<<endl;
UnCorrXIneffiPar = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetUnCorrYInefficiency(G4double val) {
cout<<"void micalcal0SD::SetUnCorrYInefficiency(G4double "<<val<<")<<endl;
// cout<<"UnCorrYInefficiency = "<<val<<endl;
UnCorrYIneffiPar = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetTimeToDigiConv(G4double val) {
cout<<"void micalcal0SD::SetTimeToDigiConv(G4double "<<val<<")<<endl;
// cout<<"TimeToDigiConv = "<<val<<endl;
// cout<<"..."<<endl;
TimeToDigiConv = val;
pAnalysis->SetTimeToDigiConvVal(val);
}

void micalcal0SD::SetSignalSpeed(G4double val) {
cout<<"void micalcal0SD::SetSignalSpeed(G4double "<<val<<")<<endl;
// cout<<"SignalSpeed = "<<val<<endl;
// cout<<"..."<<endl;
SignalSpeed = val;
pAnalysis->SetSignalSpeedVal(val);
}

void micalcal0SD::SetCorrNoise1(G4double val) {
cout<<"void micalcal0SD::SetCorrNoise1(G4double "<<val<<")<<endl;
// cout<<"CorrNoisePar1 = "<<val<<endl;
CorrNoisePar1 = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetCorrNoise2(G4double val) {
cout<<"void micalcal0SD::SetCorrNoise2(G4double "<<val<<")<<endl;
// cout<<"CorrNoisePar2 = "<<val<<endl;
CorrNoisePar2 = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetCorrNoise3(G4double val) {
cout<<"void micalcal0SD::SetCorrNoise3(G4double "<<val<<")<<endl;
// cout<<"CorrNoisePar3 = "<<val<<endl;
CorrNoisePar3 = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetRandomNoise(G4int val) {
cout<<"void micalcal0SD::SetRandomNoise(G4int "<<val<<")<<endl;
// cout<<"RandomNoisePar = "<<val<<endl;
RandomNoisePar = val;
// cout<<"..."<<endl;
}

void micalcal0SD::SetRootRandom(G4int val) {
cout<<"void micalcal0SD::SetRootRandom(G4int "<<val<<")<<endl;
cout<<"-----"<<endl;
if (val) {cout<<" Root Random Enabled "<<endl;}
else {cout<<" Root Random Disabled "<<endl;}
}

```

Jul 16, 21 18:23

micalCal0SD.cc

Page 28/28

```

cout<<"-----"<<endl;

// cout<<"RootRandom = "<<RootRandom<<endl;
RootRandom = val;
// cout<<"..."<<endl;
}

int micalcal0SD::GetRandomXY(double& GapX, TH2D* tmphistx) {
double sumX = G4UniformRand();
int xbinf = tmphistx->GetXaxis()->FindBin(GapX);
int nmult = -1;
int iiter = 0;
int nmusedhits = 3;
while(nmult<=0) {
sumX = G4UniformRand();
double valY = 0.0;
for (int ijf=0; ijf<=nmusedhits+1; ijf++) {
valY += tmphistx->GetBinContent(xbinf, ijf+1);
if (valY > sumX) {
nmult = ijf;
break;
}
} // for (int ijf=0; ijf<=nmusedhits; ijf++) {
if (iiter++==100) {
nmult = 1;
}
} // while(nmult<=0) {

return nmult;
}

```