Table of Contents

Reference Link

- AtlasSiliconRodGroup
- Andreas's bookmark
- Pixel DSP Workshop 29 October 2007~ 9 Nov 2007
    - some educational material for NewDSP design and implementation
- Trevor Vickey's SiRod page: link to documents and manuals in 2003

Talk

- 02-11-14_PM_SctPixROD SCT-Pixel RPD, P. Moretini, block diagram explains the design of ROD and DAQ software
- Genova Pixelweek talk - DAQ/electronics talk from Genova in Pixel Week between 1995~2005
- ppt Douglas Ferguson, Writing a Primitive or Task. Five easy steps to make a primitive.

Before you start

Software installation

- Install Code Composer Studio 3.1 - compiler and simulator for TMS320C6X series. We use C6201 (fixed-point) for one master DSP and C6701 (floating-point) for four slave DSPs
    - ♦ TMS320C6000 Code Composer Studio Manuals
        - ◊ SPRU509C: Code Composer Studio Getting Started Guide, Provides basic procedures in program development flow order to help you begin programming Code Composer Studio.
        - ◊ Texas Instruments: Windows application software
        - ◊ We don't have evaluation board
- Install windows version CVS manager (option) - check out the source codes
    - ♦ WinCvs 2.0.2.4 (released 2005-08-15) - it can be used to check out/in the source codes and built binaries into CVS in Windows.
    - ♦ python 2.5.2 - it's required by WinCvs
        - ◊ WinCVS could not find Python 2.1 installed on your system - One have to add the settings after starting WinCVS. Admin -> Preferences -> WinCvs -> Pyton DLL = C:\WINDOWS\system32\python25.dll
- John's FPGA Binaries

Education

- The DSP handbook Algorithms, Applications and Design Techniques, Andrew Bateman, Iain Paterson-Stephens
- ROD - Read Out Driver is core of the off-detector read out device. To understand how it controls Front End electronics and receives data. We have to learn on-line detector archietecture.
- FE and MCC
    - ♦ pdf: An Introduction to ATLAS Pixel Module Calibration Procedures by John Richardson - Give an overview of command structure and data flow of MCC and give threshold scan as an example.
    - ♦ ATL-IP-QP-0144: ATLAS Pixel Module Electrical Tests by LBL in 2004, introduction to electrical tests of single module - bare module, groups of modules on PCB and cut out from PCB and mount on the stave or sector. Use TurboDAQ, a standard alone software and hardware test system. We probably have one in the SR1.
    - ♦ pdf: 12/9/2003, ATLAS On-detector Electronics Architecture by Kevin Einsweiler. This document explains the MCC, register, bits and pins.
- RODs and Bocs
    - ♦ ROD user's manual March 11 2008 - describes RODs in low level
    - ♦ Michael Levi's page: documents for design of SCT/!PIXEL ROD since 1999
- C Programming knowledge requirements
    - ♦ The Function Pointer Tutorials
    - ♦ void pointers from Physics 2101 - Scientific Computing by Randy Kobes
    - ♦ Bitwise operation: NOT "~" (tilde), OR "|" (pipe), XOR "^" (caret), AND "&" (ampersand), shift ">>", "<<"
- bit or words
    - ♦ word_twiki: one word = 4 bytes = 4 x (8 bits) = 32 bits
    - ♦ Byte: one bytes = 8 bits
- Tutorial
    - ♦ Texas Instrument tutorial
    - ♦ DSP Tutor
    - ♦ Development of Electronic Module at CERN
    - ♦ Physics 351 Fall 2007 Introduction to Digital Electronics

- NFS CCLI-A&I: David Waldo's Digital Signal Processing Laboratory for Real-Time Systems Design and Implementation
    - Basic introduction of assembly and C and concept of DSP
    - Use TMS320C6701 EVM boards as examples
- Paolo Alto news: Routers, Growth in World Markets for FPGA, DSP, and ASIC Driven by Demand for Higher Flexibility in Medical Imaging Equipment
    - The increasing requirement of high-end processors and field-programmable gate array (FPGA) in imaging equipment for high performance computing capabilities drives global demand for FPGAs, digital signal processors (DSPs) and application-specific integrated circuits (ASICs).
    - New analysis from Frost & Sullivan (http://www.semiconductors.frost.com), World Markets for FPGA, DSP and ASIC in Medical Imaging Equipment, finds that the market earned revenues of $602.87 million in 2006 and estimates this to reach $917.15 million in 2011.
    - Although FPGAs have begun to replace DSP in most applications, there exists a trend to use them as co-processors. There still exists a healthy market for DSPs with a small reduction in the number of units used.
    - While FPGA does most of the computing, DSP uses its complementing capabilities to offload some of the computations done prior to the image processing. On the other hand, some medical OEMs prefer to use application-specific integrated circuit (ASIC) to implement specific functionalities in their equipment.
- slides: Sergio Carrato's slides for TMS320C6x DSP Design Workshop
    - slides to talk about optimisation and DSP architecture
- DSP's existence and distinction

PRAGMA, CODE_SECTION

- spra824: TI: Memeory Allocation Techniques in System with Dynamic Swapping of Application Codes - The C compiler uses two pragmas for Code and Data sections
- html You can selectively place C functions in internal or external memory by using #pragma directives.
- Optimization on TI DSP system - describes DSP program development flow, Code Optimization, Memory Optimization
- CCS, open a project, master.cmd - defines the input of memory section and master.map deines the output of memory usage after compilation

```
SECTIONS {
/*      .text    > IPROG   /* all code for program added by RTS lib & CSL */
      .sect0   "iprog"   > IPROG
      .sect1   "idata"   > IDATA
      .sect2   "iheap"   > IHEAP
      .sect3   "xprog"   > XPROG
      .sect4   "xdata"   > XDATA
}
```

Compose and Build DSP code in Code Composer Studio 3.1

- Setup
  - Setup CCStudio v3.1 -> C6201 Device Simulator - TMS320C6201
  - Setup CCStudio v3.2 -> can't find C62xx family. Is it the reason why we just use CCStudio 3.1?
- check out the code into directory C:\projects\RodDaq-PitTestVersion\RodDaq_checkIn

```
CVSROOT=:ssh:isscvs.cern.ch:
PATH=/local/reps/atlaspixeldaq/
Module=RodDsp
Tag=NewDsp-Release-1-4
```

- start code composer. check the system configuration.
  - D. Ferguson's talk
    - Master DSP: tms320c6201
    - Slave DSP: tms320c6701
- Build tilab - Why do we need tilab?

```
Menu> Project-> Open -> C:\projects\RodDaq-PitTestVersion\RodDaq_checkIn\NewDsp\tilib\tili
Menu> Project-> Build
------------------------------------
[Archiving...] "C:\CCStudio_v3.1\C6000\cgtools\bin\ar6x" @"Debug.lkf"
  ==>  new archive 'C:/Projects/siRodDaq/NewDsp/tilib/tilib.lib'
  ==>  building archive 'C:/Projects/siRodDaq/NewDsp/tilib/tilib.lib'
Build Complete,
  0 Errors, 0 Warnings, 0 Remarks.
```

- Build MASTER

```
Menu> Project > Open > C:\projects\RodDaq-PitTestVersion\RodDaq_checkIn\NewDsp\Master\
Ignore All warning message. We'll configure it later.

Cursor point to the master_pix.pjt and right click->Build Options
In the General tab
Initial build steps: C:\projects\RodDaq-PitTestVersion\RodDaq_checkIn\NewDsp\dspAux\birthd
Final build steps: C:\projects\RodDaq-PitTestVersion\RodDaq_checkIn\NewDsp\dspAux\coff2\co

Menu>Project-> Build
-------------------------  master_pix.pjt - Custom  -------------------------
c:\projects\siRodDsp\dspAux\birthday\bin\birthday.exe birthday.c
The system cannot find the path specified.

Build Complete,
  1 Errors, 18 Warnings, 0 Remarks.
  Please view the build output for errors and warnings for any custom steps.
```

- run simulator
- The file sizes are always: 512kb for master.bin and 225 kb for slave.dld

Linux

- Use WINE compiler in the linux
- copy C6000 family to linux and keep the same directory structure.

```
CCSTudio_v3.1/C6000/cgtools/bin
```

```
CCSTudio_v3.1/C6000/cgtools/include
CCSTudio_v3.1/C6000/cgtools/lib
```

- WINE environment setup
  - ♦ config/Makefile.cfg

```
PROJECT_DRIVE=f
CCS_DRIVE=g
CCS_DIR=CCStudio_v3.1
CCS_PATH=/home/${USER}/$(CCS_DIR)
TOOLCHAIN=C6000
WINEPREFIX=$(ROD_DAQ)/NewDsp/winecfg
CC=export WINEPREFIX=$(WINEPREFIX);wine $(CCS_PATH)/$(TOOLCHAIN)/cgtools/bin/cl6x.ex
AR=export WINEPREFIX=$(WINEPREFIX);wine $(CCS_PATH)/$(TOOLCHAIN)/cgtools/bin/ar6x.ex
COFF2FLASH:=export WINEPREFIX=$(WINEPREFIX);wine ../dspAux/coff2/coffToFlash2/Releas

COFF2DLD:=export WINEPREFIX=$(WINEPREFIX);wine ../dspAux/coff2/coffToDld2/Release/co
%.obj:   %.c
   $(CC) $(OPTIONS)  $<
%.obj:   %.asm
```

  - ♦ Makefile

```
        mkdir -p ../../projects_wks/pixelRod/tilib/obj
        mkdir -p ../../projects_wks/pixelRod/master/obj
        mkdir -p ../../projects_wks/pixelRod/slave/obj
        mkdir -p ../../projects_wks/pixelRod/tilib/asm
        mkdir -p ../../projects_wks/pixelRod/master/asm
        mkdir -p ../../projects_wks/pixelRod/slave/asm
 wineprefixcreate --prefix ./winecfg
        cd ./winecfg/dosdevices ; rm -f $(PROJECT_DRIVE)\: ; ln -sf ../../../.. $(PR
        cd ./winecfg/dosdevices ; rm  -f $(CCS_DRIVE)\: ;ln -sf $(CCS_PATH)/.. $(CCS
         if [ ${TOOLCHAIN} = "c6000"   ]; then sed -i  s/'"PATH"=str(2):"C:\\\\windo
        if [ ${TOOLCHAIN} = "C6000"   ]; then sed -i  s/'"PATH"=str(2):"C:\\\\window
```

  - ♦ It generates winecfg directory with required configuration. Sometimes, wineprefixcreate function can fail. For example,

In the file winecfg/system.reg, the "PATH" is not pointed to user's own CCSTudio_v3.1.

New DSP codes

Feature

- Goals of the new code base Travis's talk in 2005 Sep:
  - Ease of maintenance and extensibility
  - More intuitive
  - Better documented
  - Modular SCT and PXL specific pieces plug into a common DSP code-base

History

- June 2005 pixelweek_dsp_050620.pdf Rod-Pixel-1-43
  - original Software infrastructure for both Pixel & SCT developed by Douglas Ferguson (Wisconsin), leaving
  - replacement from Wisconsin: PostDoc Trevor Vickey + Student Alden Stradling
  - Pixel matters: AK, Joe Virzi
  - SCT matters: request for manpower (A. Barr)
- 2005-08-29 15:52:00 The first version is checked in by Alden Stradling
- 2006-03-31 - 2006-10-25 js Virzi

Under Development

- Jun 20 2008
  - Paolo: Error handling for better debug
  - Paolo: Cross-talk within module groups - data transfer via EoC might have cross-talk
- Tuesday 01 July New DSP code by Jed Biesiada

Description

- NewDsp/MASTER

```
| header | src code |
|        | main.c           |
|        | commandChannelMaster.c    |
|        | leds.c           |
|        | requestChannelMaster.c    |
|        | skeleton.c           |
|        | taskStarts.c               |
|        | userExtension.c        |
|        | userExtensionDynamic.c    |
|        | validate.c           |
| bocStructure.h    |             |
| mainpage.h       |             |
| bitStream.h       |  bitStream.c      |
| boc.h          |  boc.c         |
| controller.h       |  controller.c      |
| diagnostics.h       |  diagnostics.c      |
| efb.h          |  efb.c         |
| emif.h       |  emif.c              |
| evtmem.h       |  evtmem.c       |
| flash.h        |  flash.c         |
| formatter.h       |  formatter.c      |
| group.h        |  group.c         |
| inmem.h        |  inmem.c         |
| masterUtils.h       |  masterUtils.c      |
```

```
| peripherals.h       |  peripherals.c      |
| quickStatusInfo.h |  quickSlaveXfer.c    |
| registers.h       |  registers.c      |
| router.h       |  router.c       |
| serialPorts.h       |  serialPorts.c      |
| slave.h        |  slave.c         |
| system.h       |  system.c       |
| talkback.h       |  talkback.c      |

Joseph Salvatore VIRZI
jsvirzi
LBL

stradling
Alden STRADLING
University of Wisconsin

hgray
Heather GRAY
Columbia University

weingar
Jens WEINGARTEN
Universitaet Dortmund

wittgen
Jens Dopke
jdopke
Bergische Universitaet Wuppertal

Matthias WITTGEN
wittgen
SLAC

aschrein
Alexander SCHREINER
University of Iowa
```

- NewDsp/MASTER/PIX/

```
boc_scanm.h
leak_scan.h
module.h
moduleTests.h
quickStatus.h
scanCtrl_2.h
scanCtrl.h
streams.h
```

FAQ

How to check the version of DSP code for each ROD?

- We have to execute the command in the SBC which contains the ROD for interest since it needs to create VME communications. The usage in SR1:

```
pixrcc03:~/daq/RodDaq/RodUtils/RodTests/RodCheckVerDsp 5 12 14 17 21
pixrcc04:~/daq/RodDaq/RodUtils/RodTests/RodCheckVerDsp 11 15 20
```

- Make sure you get following output:

```
++++++++++++ ROD(12)   S/N 213
**** PRM Version (12) = f17
```

```
**** RCF Version (12) = f2e  PIXEL
**** FMT Version (12) = f20p
```

If you query the non-existing ROD on that crate you will get:

```
**** PRM Version (11) = fff
**** RCF Version (11) = fff  PIXEL
**** FMT Version (11) = fffp
```

- Andreas' instructions

```
Subject: DSP version check
From: Andreas Korn <AKorn@lbl.gov>
Date: Fri, 6 Jun 2008 14:14:18 +0200
To: Jed Biesiada <Jed.Biesiada@cern.ch>

Hi,

I just hacked together a tool to check firmware on the ROD:
RodDaq/RodUtils/RodTests/RodCheckVerDsp.cxx

This tool wortks with booth DSP versions.
It has hardcoded default verisions and complaints if they are different.
Usage :
   RodCheckVerDsp 7 14


Output:

**** RTR Version (7) = f1f  PIXEL
**** old DSP Version (150) =

++++++++++++ ROD(14)   S/N 48
**** PRM Version (14) = f11
+++++++++ WRONG PRM Firmware (f17 / e60) +++++++++

**** RCF Version (14) = f2b  PIXEL
+++++++++ WRONG RCF Firmware (f2e) +++++++++
**** FMT Version (14) = f1fp f1fp f1fp f1fp f1fp f1fp f1fp f1fp
**** EFB Version (14) = f21  PIXEL
+++++++++ WRONG EFB Firmware (f22) +++++++++
**** RTR Version (14) = f1f  PIXEL
**** new DSP Version
```

Project 1

- Make pxiel level scan variabiles working in DSP level, e.g. VCal (has beeb done), TDAC (7bit), FDAC (3bit)
- The configure() function is called:

```
static void setupMaskStage(ScanCtrl_2 *scanCtrl, int loop) {
        PixelRegister iMask, oMask, xtalk_off, xtalk_on, *pixelRegister;
   iMask = (1 << pixel_enable) | (1 << pixel_select) | (1 << pixel_kill);
   oMask = (1 << pixel_enable) | (1 << pixel_select) | (1 << pixel_kill) | (1 << pixel_hitl
   oMask = ~oMask;
   xtalk_off = (1 << pixel_select) | (1 << pixel_kill); // Hopefully fires the the Calibra
   xtalk_on = (1 << pixel_enable) | (1 << pixel_kill);  // Reads out...

        Get groupList from scanCtrl.
        Iterate group, get module
        Iterate module, get scanConfig,
           set values of scanConfig:
```

```
            iterate front end chips on each module,
            iterate pixelRegister on each front end chip, pixelRegister is a 2D array [C
         module->configure(module, module->scanCfg);
}

primFxns.c:
int primFxn_sendModuleConfig(PrimData *primData) {
   print(VerboseLevelDebug, "sending module(%d) configuration\n", moduleId);

   status = module->configure(module, moduleCfg);
   return status;
}
```

- configure is defined at:

```
module.c
int configure(Module *module, ModuleConfig *cfg) {
      for(bit=0;bit<N_PIXEL_REGISTER_BITS;++bit) {
         writePixelRegister(bitStream, fe, bit);
         sport->sendWait(sport, bitStream);
      }
}
```

- Current codes:

```
PixLib/PixController/PixScan.h
ScanType: TDAC_TUNE, TDAC_TUNE_ITERATED, GDAC_TUNE, TDAC_FAST_TUNE, FDAC_TUNE, IF_TUNE
enum HistogramType  { TDAC_T, TDAC_THR, TDAC_LOOP, OCCUPANCY_THR, GDAC_T, GDAC_THR, FDAC_T
 ScanParam      {TDACS = SCAN_TDACS, FDACS = SCAN_FDACS,GDAC = SCAN_GDAC, STROBE_DURATION =
enum EndLoopAction  {TDAC_TUNING, TDAC_FAST_TUNING, GDAC_TUNING, T0_SET, FDAC_TUNING,IF_TU

OCCUPANCY histogram (default)
runType=NORMAL_SCAN (default)
-> DIGITAL_TEST, ANALOG_TEST
    no loop at all
-> THRESHOLD_SCAN
    VAL dsp loop


-> TDAC_TUNE, TDAC_TUNE_ITERATED, GDAC_TUNE
    first loop - DSP loop change Vcal with Scurve fitting
    second loop- non DSP loop, change TDACs (1-70) or GDAC

-> TDAC_FAST_TUNE
    one loop on TDACS

-> FDAC_TUNE || IF_TUNE
    one loop
-> T0_SCAN
   DSP loop 0 , param= STROBE_DELAY;
   nonDSP loop 1, param= TRIGGER_DELAY;
-> TIMEWALK_MEASURE
    //STROBE_DELAY is DSP but VCAL is non-DSP
    // Andreas spent sometime to restructure new DSP code which can match the structure in
    // new DSP now treats everything in one big loop.
    // if we want two loop DSP, we have to do some works. since slave DSP can not understa

m_runType = RAW_EVENT
-> BOC_RX_DELAY_SCAN, BOC_V0_RX_DELAY_SCAN, BOC_THR_RX_DELAY_SCAN, BOC_THR_DEL_LONG, BOC_R

m_runType =  FMT_COUNT
BOC_THR_DEL_FAST, SHAPESCAN, BOC_TUNE, OPTO_TUNE, BOC_LINKSCAN

m_runType = FMT_COUNT_LINKSCAN
BOC_LINKSCAN

m_runType = IN_LINK_SCAN
```

```
BOC_INLINKSCAN
```

THRESHOLD SCAN

```
  } else if (presetName == THRESHOLD_SCAN) {
PixController/PixScan.cxx
 } else if (presetName == THRESHOLD_SCAN) {
    m_repetitions = 100;  // Global scan config, defin in PixScan.h
    m_strobeDuration = 500;
    m_maskStageSteps = 32;
    m_digitalInjection = false;
    m_loopActive[0] = true;
    m_loopParam[0] = VCAL;
    m_dspProcessing[0] = true;
    m_dspFEbyFE=false;
    setLoopVarValues(0, 0, 200, 201);
    m_loopAction[0] = SCURVE_FIT;
    m_dspLoopAction[0] = true;
    m_histogramFilled[OCCUPANCY] = true;
    m_histogramKept[OCCUPANCY] = false;
    m_histogramFilled[SCURVE_MEAN] = true;
    m_histogramKept[SCURVE_MEAN] = true;
    m_histogramFilled[SCURVE_SIGMA] = true;
    m_histogramKept[SCURVE_SIGMA] = true;
    m_histogramFilled[SCURVE_CHI2] = true;
    m_histogramKept[SCURVE_CHI2] = true;
}
void PixModuleGroup::scanLoopStart(int nloop, PixScan *scn) {
  // This method prepares the module group for the beginning of a particular loop.
  // The typical operation to perform  in this phase is the setting of the loop scan
  // variable to the appropriate initial value.
  // Do nothing else if loop is disabled or handled by the ROD
   if (scn->getLoopActive(nloop)) {
    switch (scn->getLoopAction(nloop)) {
    case PixScan::SCURVE_FIT:
      break;
}

//does nloop here mean nth loop? we have only 0,1,2 3 loop. Right?
void PixModuleGroup::scanLoopEnd(int nloop, PixScan *scn) {
  // This method will perform the end-of-loop actions.
  // Typically this method will upload histograms or fit results from the
  // ROD and store them in the appropriate PixScan structures.

  // Backward comaptibility for apps without scanTerminate
  if (nloop == 0 && m_execToTerminate) scanTerminate(scn);

  // Nothing to do if the loop is inactive, unless it's loop 0
  if (!scn->getLoopActive(nloop) && nloop != 0) return;

  // Nothing to do for loop 0 if loop 1 is executed by the ROD??
 //schsu:  number of loops =0, means no loop, then, it's just mask stage. just wait for DSP. it i
  if (nloop == 0 && scn->getDspProcessing(1)) return;

  // Restore original scan variables if needed
  // who change the OB_VISET?? is it done by BOC scan???
  if (scn->getLoopParam(nloop) == PixScan::OB_VISET) {
    std::cout << "PixModuleGroup::scanLoopEnd nloop, ok OB_VISET "<< std::endl;
    resetViset();
  }
  // Download histograms with the FILLED and KEPT flags (for loop 0 and 1 if executed by the ROD)
  if ((nloop == 0 && scn->getDspProcessing(0)) ||
      (nloop == 1 && scn->getDspProcessing(1)) ||
      (nloop == 0 && !scn->getLoopActive(0))) {
```

```
    // Download dsp histograms
    std::map<std::string, int> &hTypes = scn->getDspHistoTypes();
  // PisScan.h{ return m_dspHistogramTypes; } where it is initialized in PixScan::initConfig(), w
    std::map<std::string, int>::iterator it;
    for (it = hTypes.begin(); it != hTypes.end(); ++it)  {
      std::string name = (*it).first;
      PixScan::HistogramType type = (PixScan::HistogramType)((*it).second);
      if (scn->getHistogramFilled(type) && scn->getHistogramKept(type)) {
     //would it be true that a histogram kept means it is filled?
        for (int mod=0; mod<32; mod++) {
          scn->downloadHisto(m_pixCtrl, mod, type);
         //PixScan::downloadHisto(PixController *ctrl, unsigned int mod, HistogramType type);
        // this will merge histogram, Histo that you want to merge if you run Fe by FE
        //mergeFEHistos(), addHisto()
        //BTW, where the download is done? why some of them need to do fitting?
        }
      }
    }
  }
 // what happen now is the RAW histograms

  // Execute the end-loop action
  switch (scn->getLoopAction(nloop)) {
  case PixScan::SCURVE_FIT:
    if (scn->getDspLoopAction(nloop)) {
      for (int mod=0; mod<32; mod++) {
        scn->downloadHisto(m_pixCtrl, mod, PixScan::SCURVE_MEAN);
        scn->downloadHisto(m_pixCtrl, mod, PixScan::SCURVE_SIGMA);
        scn->downloadHisto(m_pixCtrl, mod, PixScan::SCURVE_CHI2);
      }
    } else {
      for (int mod=0; mod<32; mod++) {
        scn->calcThr(m_pixCtrl, mod, scn->scanIndex(2), scn->scanIndex(1));
      }
    }
    break;

 if ((nloop == 0 && scn->getDspProcessing(0)) ||
      (nloop == 1 && scn->getDspProcessing(1)) ||
      (nloop == 0 && !scn->getLoopActive(0))) {
    RodPixController *rod = dynamic_cast<RodPixController*>(m_pixCtrl);
    rod->stopScan();
  }

  // Update scan termination time
  scn->scanTimeEnd();
}


}
```

- moduleConfig.h - defines pixel register, module register

enum { pixel_hitbus, pixel_select, pixel_enable, pixel_tdac0, pixel_tdac1, pixel_tdac2, pixel_tdac3, pixel_tdac4, pixel_tdac5, pixel_tdac6, pixel_fdac0, pixel_fdac1, pixel_fdac2, pixel_kill };

#define MCC_CSR 0x0 #define MCC_LV1 0x1 #define MCC_FEEN 0x2 #define MCC_WFE 0x3 #define MCC_WMCC 0x4 #define MCC_CNT 0x5 #define MCC_CAL 0x6 => This is the one #define MCC_PEF 0x7

- What does this mean?

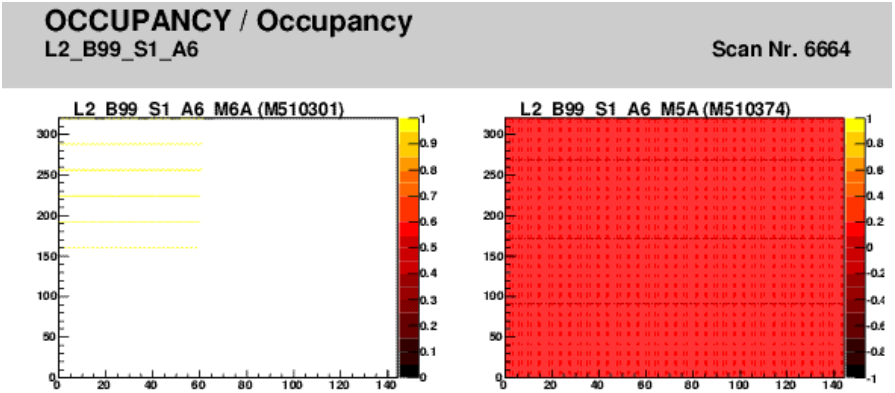
```
#if 0
```

```
typedef struct {
   unsigned short int hitBus : 1;
   unsigned short int select : 1;
   unsigned short int mask : 1; /* enable readout */
   unsigned short int tdac : 7;
   unsigned short int fdac : 3;
   unsigned short int kill : 1; /* preamp. 1 = enable preamp */
} PixelRegister;

#else

typedef UINT16 PixelRegister;

#endif
```

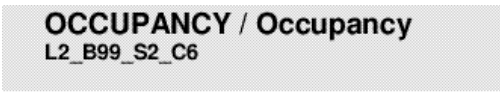
Digital Test

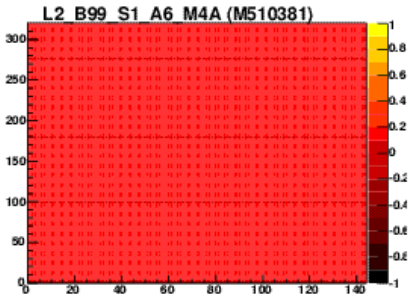
- Digital test with ROD 20
  - ♦ NewDsp just provides none sense results
  - ♦ OldDsp seems working

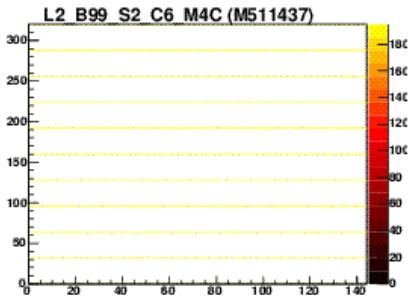
NewDsp, s6664, TOOTHPIX-2008-C2(Mod), TOOTHPIX-BOC (Conn)

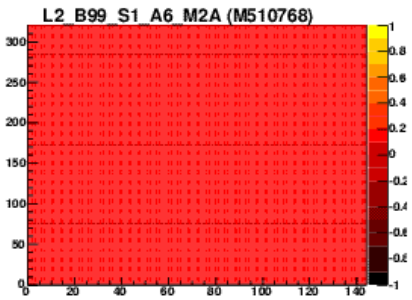


NewDsp, s6664



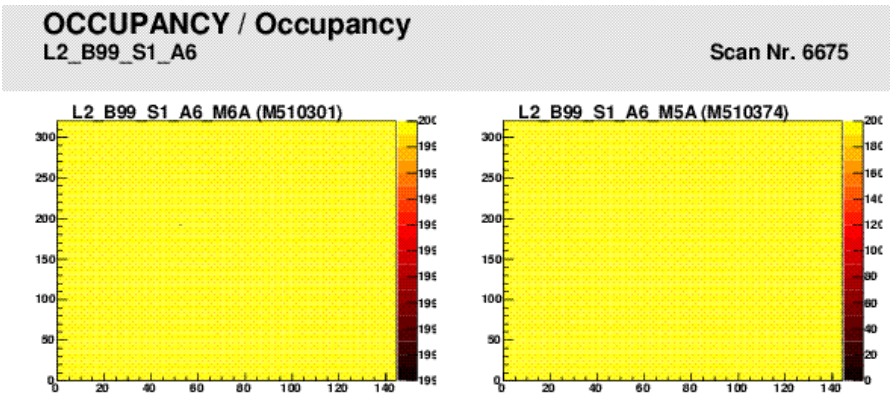


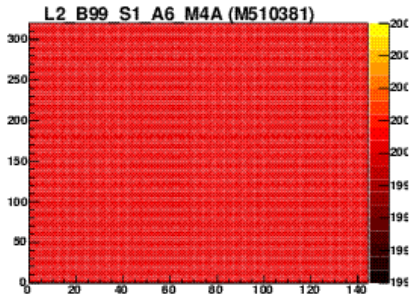


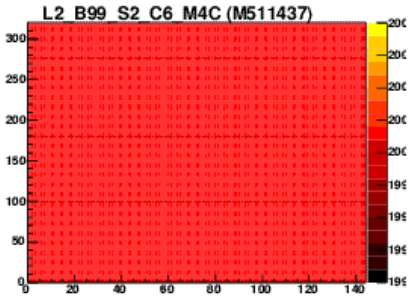


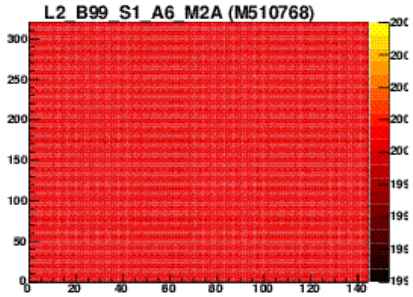
OldDsp, s6675, TOOTHPIX-V3 (Mod) TOOTHPIX-DEFAULT (Conn)

OldDsp, s6675



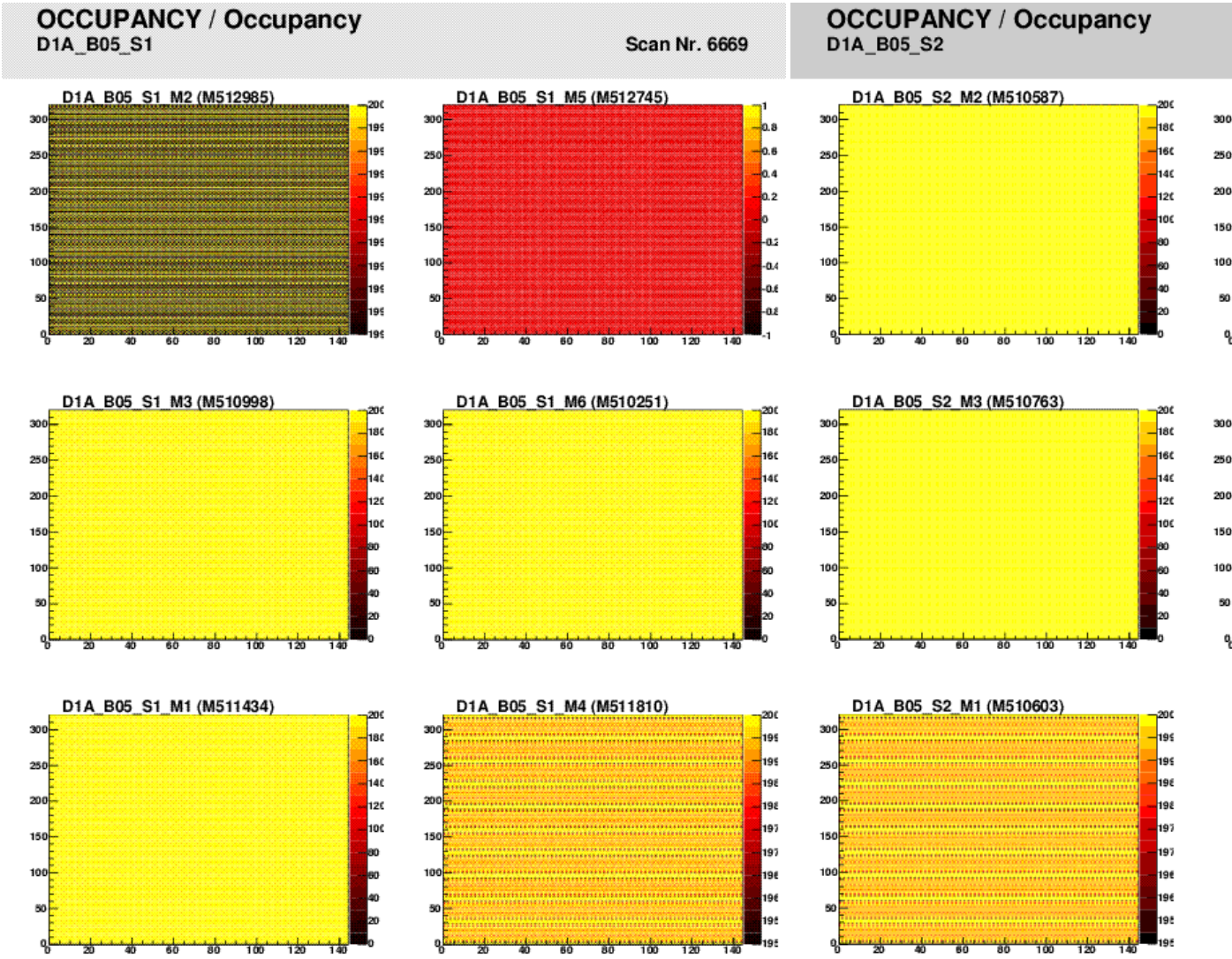






- Digital test with ROD 14
  - ROD 14 has a worse BOC.
  - OldDSP crash immediately on DigitalTest and NewDsp gives results not make sense
  - We should do a BOC scan, make sure RX_THR and RX_DElay are in the Error Free Region
  - If it's not, we should do a BOC Tunning, which will do a 3D tunning, VIset, RX_THR, and RX_Delay. It's a known problem that saved configuration not working. We could try JensAnalysis code which will save tunned configuration or we could try CT code (but not support anymore)

NewDsp, s6669, TOOTHPIX-2008-C2(Mod), TOOTHPIX-BOC (Conn) NewDsp, s6669



New DSP works need to be done

Scan Type	OldDSP	NewDSP
Basic		
DIGITAL_TEST		?
ANALOG_TEST		?

THRESHOLD_SCAN		?
INTIME_THRESH_SCAN		
TOT_CALIB		?
TDAC_TUNE		
TDAC_TUNE_ITERATED		
TDAC_FAST_TUNE		
INCREMENTAL_TDAC_SCAN		
GDAC_TUNE		
FDAC_TUNE		
IF_TUNE		

Module property		
LEAK_SCAN_DSP	X	Alex

STO_DSP		slow turn on

Timing		
TIMEWALK_MEASURE		
T0_SCAN		
CROSSTALK_SCAN		

BOC, runType=RAW_EVENT		
BOC_RX_DELAY_SCAN		Jens
BOC_THR_RX_DELAY_SCAN		
BOC_V0_RX_DELAY_SCAN		
BOC_RX_THR_SCAN		
BOC_THR_DEL_LONG		
BOC_THR_DEL_FAST		
BOC_THR_DEL_DSP		

Runtype= FMT_COUNT		
BOC_TUNE		
OPTO_TUNE		
MSR_SCAN		Mark to Space Ratio
BPM_FINEPHASE		
SHAPESCAN		

| TunType = FMT_COUNT_LINKSCAN

BOC_LINKSCAN		
BOC_INLINKSCAN		

New	
BCID	Mattias

Structure

- SctPixelRod::RodModule *m_rod;

SLink dump

- One can dump raw data from ROD via Robin or Filar card. Since Robin dump has a limitation on stop dumping after 800 events. Here shows how to do filar dump
- In SR1, the filar card is installed in ros01. We can check the status of filar card here

```
[ros01] ~ > cat /proc/filar
Card 0:
=======
       |       |       |          |free      |     |        |       |          |free
       |       |       |entries in|entries in|     |        |       |entries in|entries in
Channel|present|enabled|ACK FIFO  |REQ FIFO  |LDOWN|Overflow|   XOFF|OUT FIFO  |IN FIFO
-------|-------|-------|----------|----------|-----|--------|-------|----------|----------
      0|    yes|    yes|         0|       >14|  yes|      no| is off|         0|       200
      1|    yes|     no|         0|       >14|  yes|      no| is off|         0|       200
      2|    yes|     no|         0|       >14|  yes|      no| is off|         0|       200
      3|    yes|     no|         0|       >14|  yes|      no| is off|         0|       200

It tells us we have enabled channel connected to channel 0
```

- We can use this command to reset filar card after plugin the SLink fiber in the back of crate

```
[ros01] ~ > filarscope
```

- Example usage

```
[ros01] ~ > filarscope
This is FILARSCOPE running on a card of revision 47

Select an option:
  1 Print help           2 Dump PCI conf. space  3 Dump PCI MEM registers
  4 Write to a register  5 Read ACK FIFO         6 Reset the FILAR
  7 Reset the S-Link     8 Dump a buffer         9 SLIDAS test
 10 Configure FILAR     11 Set latency counter  12 Test DMA protocol
 13 Record data
  0 Quit
Your choice [1] :7
Reset channel 0 (1=yes  0=no): [1] :1
Waiting for link to come up...
Reset channel 1 (1=yes  0=no): [1] :1
Waiting for link to come up...
Reset channel 2 (1=yes  0=no): [1] :0
Reset channel 3 (1=yes  0=no): [0] :0

After setting, check the proc file again, we have two SLink connection
Card 0:
=======
       |       |       |          |free      |     |        |       |          |free
       |       |       |entries in|entries in|     |        |       |entries in|entries in
Channel|present|enabled|ACK FIFO  |REQ FIFO  |LDOWN|Overflow|   XOFF|OUT FIFO  |IN FIFO
-------|-------|-------|----------|----------|-----|--------|-------|----------|----------
      0|    yes|    yes|         0|       >14|   no|      no| is off|         0|       200
      1|    yes|    yes|         0|       >14|   no|      no| is off|         0|       200
      2|    yes|    yes|         0|       >14|  yes|      no| is off|         0|       200
      3|    yes|    yes|         0|       >14|  yes|      no| is off|         0|       200
```

- Here is the command to dump raw data. We have to start it before the start of the scan and stop it in the end of the scan by doing Ctrl-c:

```
To dump data from channel 0, we should use -o 1 as input parameter
[ros01] ~ > filar_slink_dst -o 1 -f /daq/results/NewDspDebug/DIGITAL_TEST/filar_S7625_digi
```

- The output ignored b0f and e0f words.

How to check configuration

```
> ${PIX_LIB}/Examples/DbServer_monitor --dump --dbPart PixelInfr_schsu
Successfully connected to DB Server PixelDbServer
 entering ready remote
 server is already enable for reading!!

- DbServer has 2 domains

  -> domain   Configuration-SR1 has 2 tags

     -> tag   NEWDSP-SCHSU-BOC (content: BOC )  has 631 objects

     -> tag   TOOTHPIX-ORIG (content: MOD )  has 5940 objects

  -> domain   Connectivity-SR1 has 1 tags

     -> tag   TOOTHPIX-V4 (content: NULL )  has 172 objects
```

ROD DSP in other detectors

Silicon ROD

- proceeding 2006: A Read-out Driver for Silicon Detectors in ATLAS Vickey, T
- J. Instrum. 3 (2008) P01003: The Data Acquisition and Calibration System for the ATLAS Semiconductor Tracker

MDT ROD

Liquid Argon

- 2007 JINST: ATLAS liquid argon calorimeter back end electronics
- DSP 6202: Simion LArg Columbia
- pdf LAr Monitoring Online and Offline Requirements
- CaloMonitoring: online/offline monitoring and profiling

Hadronic Tile

- Valencia Tilecal doc: TMS320C6414 DSP
- JINST 2 T02001: Algorithms for the ROD DSP of the ATLAS Hadronic Tile Calorimeter, B Salvachua et al 2007
- pdf DSP online algorithms for the ATLAS TileCal Read-Out Drivers
- talk Andrea Dotti: Online monitoring at the testbeam and for commissioning

CSC ROD

- Donovan Hawkins' thesis for CSC ROD
- iEEE Off-Detector Electronics for a High-Rate CSC Detector, TMS320C6203 DSP
- Michael Schernau
- Steve Pier
- CSC-ROD

-- ShihChiehHsu - 18 Jun 2008

This topic: Main > ShihChiehHsuDSP
Topic revision: r37 - 2009-02-14 - ShihChiehHsu