CREDITNET SWITCH USER GUIDE

Version 1.0
Gary Depelteau [depeltea@bnr.ca]
Thomas NG [ngt@bnr.ca]
Bell Northern Research

This document

This document is the user guide of the CreditNet ATM switch co-developed by Harvard University and Bell-Northern Research.

1.0 Introduction

This is the User Guide for Creditnet ATM Switch (also known as ARPA Switch). This guide describes the installation procedure of the switch software, the configuration of the GPU, the start-up script file, GPU memory upgrade. It also includes a printed version of the on-line manual pages of the switch software.

2.0 Release Notes

The latest version of Creditnet Switch SW was released on August 24 95. This note serves to highlight some of the new features of this release. It also serves to explain the partitioning of the S/W developed by BNR, how to find functions that you may be looking for and how to use the man pages.

The main features of this new release of Creditnet Switch S/W are:

1. A more complete set of SCU Drivers and diagnostics.
2. A new module which performs complete SCU initialization to default initial conditions.
3. A sysInitCtl() function which now performs all the CM and PM initialization and diagnostics for which BNR is responsible.
4. A set of Interrupt Service Routines and some monitoring capabilities which report on the status of switch interrupts received by the GPU from the SCU. The only interrupts are the SCU CDM Chalkmark search stale CADR interrupt, SCU CDM Chalkmark search done interrupt and the SCU CTC interval timing complete interrupt.
5. A set of PCC routines were added. That includes fillPcc, testChecksums, several pcc data bus tests and a repeat mode on PCC.
6. A small routine for non-flow control VC setup (nfvc.c) and a function for setting up PM (ij.c)were added as examples.
7. On-line manpages for all S/W that have been written. They are associated with the modules listed below.

2.1 Partition of S/W

The partition of CreditNet Switch S/W is based upon individual hardware block, i.e. GPU, SCM and PM.

2.2 New Release Contents

This new release can be found in:YOUR_BASE_DIR/Release and it includes the following major object files:

gpu.o: composed of scmUtil, flashCtl, gpuUIinterface, xmemTest, gpuDiag, alias, cell.

scm.o: composed of scuDrv, scuInit, scuIntLib, intMonLib, scuSCoreTest, scuSMDiag, scuSDiag, scuHVDiag, sysInitCtl.

gpuVW.o: linked to VxWorks so there is no need to download this. Its composed of sccLib, gpuWdLib, gpuNmiLib.

gpuVW_res.o: linked to VxWorks bootroms so there is no need to download this. It is replaced when the actual VxWorks image including gpuVW.o is downloaded. It is composed of fDramTest, scmUtil_res, gpuDiag_res, sccLib, gpuWdLib, gpuNmiLib.

nfvc.o: composed of nfvc, a function for setting up non-flow control VC.

pmAll.o: composed of pccUtil, pmCtrl, vxServer, pmVCsetup, pccDiag.

pmConf.o: composed of PM configuration functions.

2.3 WARNING

Mixing different versions of the Creditnet Switch SW probably will not work since there are some very painful work-around needed to avoid dependencies.

3.0 Installation procedures

A file called Released.tar.zip (or Released.tar.gz) that contains some essential SW for the switch to run including the VxWorks image is shipped along with the switch. The installation procedure of the SW is as below:

- In case of a zipped file, i.e. with filename Release.tar.zip, do the following:
  % unzip Released.tar.zip
  % tar xvf Released.tar
- In case of a gzip'ed file, i.e. with the name Release.tar.gz, do the following:
  % gunzip Released.tar.gz
  % tar xvf Released.tar

It should then produce a directory tree such as the one below:

```
./          DOCUMENTATION/   PM.rom/   TAGS     UI/          man/              startup.scr*    vcset.scr
../         GPU/             SCH/      TLG/     include/     nfvc.scr          switchConf.c
COMIF/      Makefile*        SCM/      TOOLS/   loadup.scr   scuHVEncDecTest   switchConf.o

COMIF:
./          Makefile       comClient.o   comServ.img*   comServ.out*   comServ.x1*
../         comClient.c*   comServ.c*    comServ.o      comServ.x

DOCUMENTATION:
./     ../                 HW_DOCS/                 SW_DOCS/

DOCUMENTATION/HW_DOCS:
./     ../

DOCUMENTATION/SW_DOCS:
./     VxWorks/                 scuDiagnostics/           userGuide/
../    gpuDiagnostics/          systemInitialization/

DOCUMENTATION/SW_DOCS/VxWorks:
./     ../     README*

DOCUMENTATION/SW_DOCS/gpuDiagnostics:
./     GpuDiagnostics.fbbook*   GpuDiagnosticsDD.doc*     GpuDiagnosticsTI.doc*
../    GpuDiagnostics.ps*       GpuDiagnosticsFN.doc*     GpuDiagnosticsTOC.doc*

DOCUMENTATION/SW_DOCS/scuDiagnostics:
./     ScuDiagnostics.fbbook*   ScuDiagnosticsDD.doc*     ScuDiagnosticsTI.doc*
../    ScuDiagnostics.ps*       ScuDiagnosticsFN.doc*     ScuDiagnosticsTOC.doc*

DOCUMENTATION/SW_DOCS/systemInitialization:
./     ../     SysInit.fbbook*   SysInit.ps*   SysInitDD.doc*   SysInitFN.doc*   SysInitTI.doc*   SysInitTOC.doc*

DOCUMENTATION/SW_DOCS/userGuide:
./     Darpa.userguide.TOC.doc   Darpa.userguide.ps          Darpa.userguide.complete.ps      Darpa.userguide.manpage.pcc.doc
../    Darpa.userguide.book      Darpa.userguide.intro.doc   Darpa.userguide.manpage.scu.doc

GPU:
./          flashes/       gpuVW.o*      ij.o      intAll.o*      nfvc.o      pmAll.o*      vxWorks*
```

```
../          gpu.o*         gpuVW_res.o*    int1.o     lc.o       nfvc.o.bak   pmConf.c*   vxWorks.sym*

GPU/flashes:
./           README              br0.bin*        br1.bin*        cardid_gpu0.bin*
../          bootrom_res.bin*    br0.hex*        br1.hex*        cardid_gpu0.hex*

PM.rom:
./           ../            bootDpram.img*   bootRom.rom*   bootSCH.img    checksum.out*

SCH:
./              bootSCH.x          i_handle_SCH.s*   schAlgo1.s    schAlgo8.o
../             bootSCH.x1*        i_table_SCH.o     schAlgo2.c*   schAlgo8.s
Makefile*       boot_dmn_SCH.o     i_table_SCH.s*    schAlgo2.o    schWW.o
Makefile~*      boot_dmn_SCH.s*    out               schAlgo2.s    schWW.s
SCH.c*          checksum.sun*      schAlgo0.o*       schAlgo3.o    schWatchdog.o
SCH.ld*         dmn_ftbl_SCH.o     schAlgo0.s*       schAlgo3.s    schWatchdog.s
SCH.o           dmn_ftbl_SCH.s*    schAlgo0.s.bak*   schAlgo4.o    startup_SCH.o
SCHdpram.inc*   dmn_ptbl_SCH.o     schAlgo1.c*       schAlgo4.s    startup_SCH.s*
bootSCH.img     dmn_ptbl_SCH.s*    schAlgo1.c~*      schAlgo6.o    x
bootSCH.out*    i_handle_SCH.o     schAlgo1.o        schAlgo6.s    xice.cfg*

SCM:
./              ../             scm.o*

TLG:
./              ../             tlg.o*

TOOLS:
./              ../             checksum.sun*

UI:
./              ../             combined*        mainprog*        switchDaemon*    uiconfig.f
uihelp.f        vxshell*        wish*

include:
./              comIF.h*          intMonLib.msg.h*     pmCtrl.h*               scuHVDiag.h           scuSDiag.msg.h*
../             configFLEX.h*     ispUtil.h            pmVCsetup.h*            scuHVDiag.msg.h*      scuSMDiag.h
PCCregions.h*   dctest.h*         isplsi.h             regions.h*sculnit.h*    scuSMDiag.msg.h
SCHError.h*     flashCtl.h*       mcpPcMemoryMap.h*    sccLib.h*sculnit.msg.h  scuWDbmIntegTest.h
SCHregions.h*   generateCell.h*   nfvc.h               schMemoryMap.h*         scuIntLib.h*          simpleCom.h*
bootError.h*    gpuDiag.h*        pcMemoryMap.h*       scmUtil.h               scuIntLib.msg.h*      sysInitCtl.h
bus.h*          gpuNmiLib.h*      pcServices.h*        scuDaisyChainTest.h*    scuPortCardTest.h*    tlgDriver.h
cell.h          gpuWdLib.h*       pccDiag.h*           scuDrv.h                scuSCoreTest.h*       tucServices.h*
comError.h*     intMonLib.h*      pccUtil.h*           scuFifoIntegTest.h      scuSDiag.h            xmemTest.h*

man:
./              ../             cat1/            cat2/            man1/            man2/
wrs.an*

man/cat1:
./              generateCell.1*   ispUtil.1*              scuDrv.1*              scuPortCardTest.1*     switchConf.1*
../             gpuDiag.1*        pccUtil.1               scuFifoIntegTest.1*    scuSCoreTest.1*        sysInitCtl.1*
alias.1*        gpuNmiLib.1*      sccLib.1*               scuHVDiag.1*           cuSDiag.1*
fDramTest.1*    gpuWdLib.1*       scmUtil.1*              scuInit.1*             scuSMDiag.1*
flashCtl.1*     intMonLib.1*      scuDaisyChainTest.1*    scuIntLib.1*           scuWDbmIntegTest.1*

man/cat2:
./                          sccOutputChar.2*                 scuCtcStopRegRead.2*       scuSCdmCmSrchTest.2*
../                         sccPrintHexWord.2*               scuCurChalkRead.2*         scuSCdmCmUpdTest.2*
Copyright.2*                sccPrintString.2*                scuCurChalkSet.2*          scuSCdmMonFATest.2*
cellCrc8.2*                 sccWait.2*                       scuDaisyChainTest.2*       scuSChalkmarkCheck.2*
checkPortCardExist.2        scmChkDramParity.2*              scuDbmCapture.2*           scuSCifCrc8Test.2*
crc16.2*                    scmChkFlashSig.2*                scuDbmInit.2*              scuSCifDataPathTest.2*
displayChecksums.2          scmMarch6Test.2*                 scuDbmSet.2*               scuSCifFifoTest.2*
downloadToPcc.2*            scmMicrosecondWait.2*            scuDbmSetup.2*             scuSCoreTest.2*
dp.2*                       scmRegTest.2*                    scuDefDSInit.2*            scuSCtcIt1Test.2*
dw.2*                       scmVerify.2*                     scuFanAlarmISR.2*          scuSCtcIt2Test.2*
fDramTest.2*                scmWalkA1Test.2*                 scuFifoIntegTest.2*        scuSDiagCtl.2*
fakeFLMGetNextCADR.2*       scmWalkD0Test.2*                 scuFifoSeqTest.2*          scuSFifoStatCheck.2*
fakeFLMInit.2*              scmWalkD1Test.2*                 scuFlmCmbfCountRead.2*     scuSFlmBasicTest.2*
fillPcc.2                   scu64BCellDecode.2*              scuFlmCntlRegRead.2*       scuSFlmIntegTest.2*
flashAddRamSize.2*          scuACAlarmISR.2*                 scuFlmCntlRegWrite.2*      scuSMCdmConCapTest.2*
flashBootParamsSet.2*       scuCMCompareCell.2*              scuFlmEmptyISR.2*          scuSMCdmRegTest.2*
flashCardIdUsage.2*         scuCMTestEntireCore.2*           scuFlmFLArrayPush.2*       scuSMCifRegTest.2*
flashCheckErase.2*          scuCMTestGeneratePatterns.2*     scuFlmFLCInit.2*           scuSMCtcRegTest.2*
flashCopyFromRam.2*         scuCMTestLostAddresses.2*        scuFlmFLPush.2*            scuSMDbmRamTest.2*
flashDLoad.2*               scuCMTestOneAddress.2*           scuFlmFailISR.2*           scuSMFlmRamStatTest.2*
flashDataPoll.2*            scuCMWalkingTest.2*              scuFlmFreeListInit.2*      scuSMFlmRegTest.2*
flashErase.2*               scuCbtErrISR.2*                  scuFlmFullISR.2*           scuSMPbusRegTest.2*
flashProgAltBank.2*         scuCdmCntlRegRead.2*             scuFlmHalt.2*              scuSMRivcLutTest.2*
flashSetRamSize.2*          scuCdmCntlRegWrite.2*            scuFlmHaltClear.2*         scuSMSharBuffTest.2*
generateCell.2*             scuCdmInit.2*                    scuFlmInit.2*              scuSMWDbmRamTest.2*
gpuCardIdTest.2*            scuCdmStaleCadrRegRead.2*        scuFlmPrime.2*             scuSMWivcLutTest.2*
gpuCntrlRegTest.2*          scuCdmStatRegRead.2*             scuFlmReset.2*             scuSRegCheck.2*
gpuDiagCtl.2*               scuCellCpyRamToSb.2*             scuFlmStatRegRead.2*       scuSSBByteCheck.2*
```

gpuDipSwTest.2*
gpuEDramTest.2*
gpuEepromTest.2*
gpuErrRegTest.2*
gpuEtherTest.2*
gpuExtBusTest.2*
gpuFlSwapTest.2*
gpuNmiInit.2*
gpuNmiInitIsr.2*
gpuNmiMon.2*
gpuNmiMonStart.2*
gpuNmiMonStop.2*
gpuNmiMonTask.2*
gpuSccTest.2*
gpuTimIntTest.2*
intCntrMon.2*
intMonInit.2*
intMonPerSet.2*
ispEnValCalc.2*
ispProgram.2*
ispRead.2*
ispVerify.2*
pccAddressTest.2
pccMarch6Test.2
pccWalkD0Test.2
pccWalkD1Test.2
printCellStruct.2*
rFifoAlFullFlag.2*
ramDebug.2*
repeatModeOff.2
repeatModeOn.2
resetScheduler.2*
rp.2*
rs.2*
runPccMemoryTest.2*
runPccService.2
sccInit.2*
scuCellCpySbToRam.2*
scuCellXferFromCore.2*
scuCellXferToCore.2*
scuCellXferToFromCore.2*
scuChalkSearchDisable.2*
scuChalkSearchEnable.2*
scuChalkSearchReset.2*
scuCifCntlRegRead.2*
scuCifCntlRegWrite.2*
scuCifHPWFifoStatRead.2*
scuCifInit.2*
scuCifLPWFifoStatRead.2*
scuCifRFifoStatRead.2*
scuCifRcvCrc8ErrISR.2*
scuCifRdPoll.2*
scuCifRdWrPoll.2*
scuCifRivcFifoRead.2*
scuCifStatRegRead.2*
scuCifWivcFifoRead.2*
scuCifWrPoll.2*
scuCoreCellRead.2*
scuCoreCellWrite.2*
scuCoreClkAlarmISR.2*
scuCoreMemoryCompareCell.2*
scuCoreMemoryTestEntireCore.2*
scuCoreMemoryTestGeneratePatterns.2*
scuCoreMemoryTestLostAddresses.2*
scuCoreMemoryTestOneAddress.2*
scuCoreMemoryWalkingTest.2*
scuCoreParErrISR.2*
scuCrmInit.2*
scuCtcInit.2*
scuCtcIntTimProg.2*
scuCtcIntTimRead.2*
scuCtcRealTimProg.2*
scuCtcStartRegRead.2*
scuCtcStatRegRead.2*
scuFlmWrapISR.2*
scuFullPortCardTest.2*
scuGetCellFromCore.2*
scuGetCellsFromCore.2*
scuHVCdmPETest.2*
scuHVCifPETest.2*
scuHVCrmBFStatTest.2*
scuHVCrmPETest.2*
scuHVCtcItPETest.2*
scuHVDS1TogCheck.2*
scuHVEncDecTest.2*
scuHVFlmCmbfPETest.2*
scuHVIdLogTest.2*
scuHVIdleTest.2*
scuHVPMIntTest.2*
scuHVPMResetTest.2*
scuHVWCadrPETest.2*
scuHwInit.2*
scuInitCtl.2*
scuInitPortCardTest.2*
scuIntInit.2*
scuIntMon.2*
scuItTimCompISR.2*
scuIvcInit.2*
scuIvcSet.2*
scuIvcSetup.2*
scuMap8To7.2*
scuPSAlarmISR.2*
scuParCrcInit.2*
scuPifInit.2*
scuPutCellInCore.2*
scuRFifoAlFullISR.2*
scuRFifoMidISR.2*
scuReadXCells.2*
scuRivcLutInit.2*
scuRivcLutSet.2*
scuRivcLutSetup.2*
scuShortPortCardTest.2*
scuStaleCadrISR.2*
scuVerCorePat.2*
scuVerify.2*
scuWDbmCapture.2*
scuWDbmIntegTest.2*
scuWDbmMultiCastTest.2*
scuWDbmWalkingOneTest.2*
scuWDbmWalkingZeroTest.2*
scuWFifoHPAlFullISR.2*
scuWFifoHPMidISR.2*
scuWFifoLPAlFullISR.2*
scuWFifoLPMidISR.2*
scuWivcLutInit.2*
scuWivcLutSet.2*
scuWivcLutSetup.2*
scuWriteXCells.2*
setOwnershipTable.2*
setPM.2*
setPort.2*
swapCell.2*
sysInitCtl.2*
testChecksums.2
ur.2*
wdFeeder.2*
wdFeederHookAdd.2*
wdFeederHookDelete.2*
wdFeederHookShow.2*
wdFeederInit.2*
wdFeederTask.2*
wdRestartFeeder.2*
wdStopFeeder.2*
wp.2*
ws.2*

man/man1:
./
../
alias.1*
fDramTest.1*
flashCtl.1*
generateCell.1*
gpuDiag.1*
gpuNmiLib.1*
gpuWdLib.1*
intMonLib.1*
ispUtil.1*
pccUtil.1*
sccLib.1*
scmUtil.1*
scuDaisyChainTest.1*
scuDrv.1*
scuFifoIntegTest.1*
scuHVDiag.1*
scuInit.1*
scuIntLib.1*
scuPortCardTest.1*
scuSCoreTest.1*
scuSDiag.1*
scuSMDiag.1*
scuWDbmIntegTest.1*
switchConf.1*
sysInitCtl.1*

man/man2:
./
../
Copyright.2*
cellCrc8.2*
checkPortCardExist.2
crc16.2*
displayChecksums.2
downloadToPcc.2*
dp.2*
dw.2*
fDramTest.2*
fakeFLMGetNextCADR.2*
fakeFLMInit.2*
fillPcc.2
flashAddRamSize.2*
flashBootParamsSet.2*
flashCardIdUsage.2*
flashCheckErase.2*
flashCopyFromRam.
flashDLoad.2*
flashDataPoll.2*
flashErase.2*
flashProgAltBank.2*
flashSetRamSize.2*
generateCell.2*
gpuCardIdTest.2*
gpuCntrlRegTest.2*
gpuDiagCtl.2*
gpuDipSwTest.2*
gpuEDramTest.2*
gpuEepromTest.2*
gpuErrRegTest.2*
gpuEtherTest.2*
gpuExtBusTest.2*
gpuFlSwapTest.2*
gpuNmiInit.2*
gpuNmiInitIsr.2*
gpuNmiMon.2*
gpuNmiMonStart.2*
sccInit.2*
sccOutputChar.2*
sccPrintHexWord.2*
sccPrintString.2*
sccWait.2*
scmChkDramParity.2*
scmChkFlashSig.2*
scmMarch6Test.2*
scmMicrosecondWait.2*
scmRegTest.2*
scmWalkA1Test.2*
scmWalkD0Test.2*
scmWalkD1Test.2*
scu64BCellDecode.2*
scuACAlarmISR.2*
scuCMCompareCell.2*
scuCMTestEntireCore.2*
scuCMTestGeneratePatterns.2*
scuCMTestLostAddresses.2*
scuCMTestOneAddress.2*
scuCMWalkingTest.2*
scuCbtErrISR.2*
scuCdmCntlRegRead.2*
scuCdmCntlRegWrite.2*
scuCdmInit.2*
scuCdmStaleCadrRegRead.2*
scuCdmStatRegRead.2*
scuCellCpyRamToSb.2*
scuCellCpySbToRam.2*
scuCellXferFromCore.2*
scuCellXferToCore.2*
scuCellXferToFromCore.2*
scuChalkSearchDisable.2*
scuChalkSearchEnable.2*
scuChalkSearchReset.2*
scuCifCntlRegRead.2*
scuCifCntlRegWrite.2*
scuCifHPWFifoStatRead.2*
scuCifInit.2*
scuCtcStopRegRead.2*
scuCurChalkRead.2*
scuCurChalkSet.2*
scuDaisyChainTest.2*
scuDbmCapture.2*
scuDbmInit.2*
scuDbmSet.2*
scuDbmSetup.2*
scuDefDSInit.2*
scuFanAlarmISR.2*
scuFifoIntegTest.2*
scuFifoSeqTest.2*
scuFlmCmbfCountRead.2*
scuFlmCntlRegRead.2*
scuFlmCntlRegWrite.2*
scuFlmEmptyISR.2*
scuFlmFLArrayPush.2*
scuFlmFLCInit.2*
scuFlmFLPush.2*
scuFlmFailISR.2*
scuFlmFreeListInit.2*
scuFlmFullISR.2*
scuFlmHalt.2*
scuFlmHaltClear.2*
scuFlmInit.2*
scuFlmPrime.2*
scuFlmReset.2*
scuFlmStatRegRead.2*
scuFlmWrapISR.2*
scuFullPortCardTest.2*
scuGetCellFromCore.2*
scuGetCellsFromCore.2*
scuHVCdmPETest.2*
scuHVCifPETest.2*
scuHVCrmBFStatTest.2*
scuHVCrmPETest.2*
scuHVCtcItPETest.2*
scuHVDS1TogCheck.2*
scuHVEncDecTest.2*
scuSCdmCmSrchTest.2*
scuSCdmCmUpdTest.2*
scuSCdmMonFATest.2*
scuSChalkmarkCheck.2*
scuSCifCrc8Test.2*
scuSCifDataPathTest.2*
scuSCifFifoTest.2*
scuSCoreTest.2*
scuSCtcIt1Test.2*
scuSCtcIt2Test.2*
scuSDiagCtl.2*
scuSFifoStatCheck.2*
scuSFlmBasicTest.2*
scuSFlmIntegTest.2*
scuSMCdmConCapTest.2*
scuSMCdmRegTest.2*
scuSMCifRegTest.2*
scuSMCtcRegTest.2*
scuSMDbmRamTest.2*
scuSMFlmRamStatTest.2*
scuSMFlmRegTest.2*
scuSMPbusRegTest.2*
scuSMRivcLutTest.2*
scuSMSharBuffTest.2*
scuSMWDbmRamTest.2*
scuSMWivcLutTest.2*
scuSRegCheck.2*
scuSSBByteCheck.2*
scuShortPortCardTest.2*
scuStaleCadrISR.2*
scuVerCorePat.2*
scuVerify.2*
scuWDbmCapture.2*
scuWDbmIntegTest.2*
scuWDbmMultiCastTest.2*
scuWDbmWalkingOneTest.2*
scuWDbmWalkingZeroTest.2*
scuWFifoHPAlFullISR.2*
scuWFifoHPMidISR.2*

gpuNmiMonStop.2*
gpuNmiMonTask.2*
gpuSccTest.2*
gpuTimIntTest.2*
intCntrMon.2*
intMonInit.2*
intMonPerSet.2*
ispEnValCalc.2*
ispProgram.2*
ispRead.2*
ispVerify.2*
pccAddressTest.2
pccMarch6Test.2
pccWalkD0Test.2
pccWalkD1Test.2
printCellStruct.2*
rFifoAlFullFlag.2*
ramDebug.2*
repeatModeOff.2
repeatModeOn.2
resetScheduler.2*
rp.2*
rs.2*
runPccMemoryTest.2*
runPccService.2
scuCifLPWFifoStatRead.2*
scuCifRFifoStatRead.2*
scuCifRcvCrc8ErrISR.2*
scuCifRdPoll.2*
scuCifRdWrPoll.2*
scuCifRivcFifoRead.2*
scuCifStatRegRead.2*
scuCifWivcFifoRead.2*
scuCifWrPoll.2*
scuCoreCellRead.2*
scuCoreCellWrite.2*
scuCoreClkAlarmISR.2*
scuCoreMemoryCompareCell.2*
scuCoreMemoryTestEntireCore.2*
scuCoreMemoryTestGeneratePatterns.2*
scuCoreMemoryTestLostAddresses.2*
scuCoreMemoryTestOneAddress.2*
scuCoreMemoryWalkingTest.2*
scuCoreParErrISR.2*
scuCtcInit.2*
scuCtcIntTimProg.2*
scuCtcIntTimRead.2*
scuCtcRealTimProg.2*
scuCtcStartRegRead.2*
scuCtcStatRegRead.2*
scuHVFlmCmbfPETest.2*
scuHVldLogTest.2*
scuHVldleTest.2*
scuHVPMIntTest.2*
scuHVPMResetTest.2*
scuHVWCadrPETest.2*
scuHwInit.2*
scuInitCtl.2*
scuInitPortCardTest.2*
scuIntInit.2*
scuIntMon.2*
scuItTimComplSR.2*
scuIvcInit.2*
scuIvcSet.2*
scuIvcSetup.2*
scuMap8To7.2*
scuPSAlarmISR.2*
scuPifInit.2*
scuPutCellInCore.2*
scuRFifoAlFullISR.2*
scuRFifoMidISR.2*
scuReadXCells.2*
scuRivcLutInit.2*
scuRivcLutSet.2*
scuRivcLutSetup.2*
scuWFifoLPAlFullISR.2*
scuWFifoLPMidISR.2*
scuWivcLutInit.2*
scuWivcLutSet.2*
scuWivcLutSetup.2*
scuWriteXCells.2*
setOwnershipTable.2*
setPM.2*
setPort.2*
swapCell.2*
sysInitCtl.2*
testChecksums.2
ur.2*
wdFeeder.2*
wdFeederHookAdd.2*
wdFeederHookDelete.2*
wdFeederHookShow.2*
wdFeederInit.2*
wdFeederTask.2*
wdRestartFeeder.2*
wdStopFeeder.2*
wp.2*
ws.2*

3.1 Configuration procedures of GPU

This section describes the process of booting the CreditNet Switch up to the point where Harvard University Application SW may be downloaded. It assumes that the BNR portion of the Switch SW has been installed in a directory visible on the Switch Workstation. This section aims to bring out the effort required to get the switch to boot but does not give an over all or detailed view of complete switch initialization. For this type of information please refer to “CreditNet Switch System Initialization”.

The first step is to get the GPU on the switch to download the VxWorks image from the workstation via the ethernet port of the GPU. It requires communicating with the GPU via its RS-232 serial port. The following is a step by step guide for initializing the board’s booting parameters. After it is done, the GPU will instead download VxWorks from the workstation using the provided IP address and target node name through the ethernet port. Note that anything preceded by // is a comment.

The next thing to do is to run a startup script called *startup.scr* which downloads BNR developed SW. The script file will execute switch diagnostics and some initialization on the portcards. After the diagnostics and initialization are finished, run a function called *flashBootParamsSet()* to ‘permanently’[1] change the GPU boot parameters in its Flash EEPROMS. Here’s how it should go:

```
*****************************************************************
                    Booting GPU960
                  From Primary Bootrom Bank

Testing current bootrom bank............Passed!
Testing Card ID EEPROM..................Passed!
Testing dynamic RAM (fast test).........Passed!
Testing real-time system clock..........Passed!
Testing ethernet controller.............Passed!
```

1. The boot parameters are stored in flash memory, i.e. the parameters are retained in the memory when the switch is powered off. The flash memory can be reprogrammed using the *flashBootParamsSet()* function as many times as required.

```
Testing serial controller...............Passed!
Testing NMI error source register.......Failed!
Testing control register................Passed!
Testing flash bank swapping.............Passed!

Round-robin scheduling started - time-slices are 400 milliseconds

GPU sanity watchdog feeding task started every 500 milliseconds.

NMI ISR installed and NMIs enabled.



                    VxWorks System Boot


Copyright 1984-1993  Wind River Systems, Inc.



CPU: GPU960
Version: 5.1.1
BSP version: 1.0 Beta-4
Creation date: Tue Mar 7 15:26:32 EST 1995


// YOU MUST HIT A KEY TO STOP THE BOOT PROCESS AND CHANGE THE
// BOOT PARAMETERS BEFORE THE FOLLOWING COUNT DOWN EXPIRES.
// THE BEST WAY IS TO START HITTING THE SPACE BAR AS SOON AS
// THE ABOVE DIAGNOSTICS FINISH.

Press any key to stop auto-boot...
 1
[VxWorks Boot]: c    // THIS COMMAND STARTS BOOT PARAMETERS CHANGE DIALOG
                       // YOU SHOULD TYPE IN YOUR OWN STUFF WHEREVER YOU
                      // SEE STUFF IN CAPITAL LETTERS.

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device          : ei
processor number     : 0
host name            : bcars782b                                   // PLACE YOUR HOST NAME HERE
file name            : /bnr/3rdparty/glan_sw/WindRiver/vw/config/gpu2960/vxWorks   //  BASE DIR/Release/GPU/vxWorks
inet on ethernet (e) : 47.53.32.8:ffff0000                          // THE IP ADDR YOU SUPPLY FOR GPU
inet on backplane (b):
host inet (h)        : 47.53.32.1                                  // THE IP ADDR OF THE HOST WORKSTATION
gateway inet (g)     :
user (u)             : nobody                                      // THE USER ID YOU USE TO DOWNLOAD SW
ftp password (pw) (blank = use rsh): nobody                        // PASSWORD CORRESPONDING TO ABOVE
flags (f)            : 0x8
target name (tn)     : gpu5                                        // THE HOST NAME OF YOU GPU
startup script (s)   :
other (o)            :


[VxWorks Boot]: @                        // INSTRUCT THE GPU TO CONTINUE BOOTING WITH NEW PARAMS

// IF EVERYTHING IS OK YOU SHOULD SEE THE FOLLOWING:


boot device          : ei
processor number     : 0
host name            : YOUR HOST NAME
file name            : YOUR BASE DIR/Release/GPU/vxWorks
inet on ethernet (e) : THE IP ADDRESS YOU SUPPLY FOR GPU
host inet (h)        : THE IP ADDRESS OF YOUR HOST WORKSTATION
user (u)             : THE USER ID YOU USE TO DOWNLOAD SW
ftp password (pw)    : PASSWORD CORRESPONDING TO ABOVE
flags (f)            : 0x8
target name (tn)     : THE HOST NAME OF YOU GPU

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading... 492216 + 34104 + 53568                       // THIS MEANS YOU'VE DOWNLOADED VXWORKS
```

Starting at 0x10001000...

Round-robin scheduling started - time-slices are 400 milliseconds

GPU sanity watchdog feeding task started every 500 milliseconds.

NMI ISR installed and NMIs enabled.

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Mounting NFS file systems from host bcars782b for target gpu5:
...done
Loading symbol table from bcars782b:/bnr/3rdparty/glan_sw/WindRiver/vw/config/gpu2960/vxWorks.sym ...done

```
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
      ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]         (R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]              ]]]]
 ]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]     ]]              ]]]]
 ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]   ]]]]]]]]]  ]]]] ]] ]]]]  ]]   ]]]]]
 ]]]]     ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]   ]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]    ]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]    ]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]    ]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]    ]]]]]]]  ]]]]    ]]]]  ]]]] ]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks version 5.1.1
 ]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.4
 ]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-1993

                           Board: GPU960.  Processor #0.
                          Memory Size: 0x400000.  BSP version 1.0 Beta-4.
```

->

// THE ABOVE IS THE VXWORKS PROMPT. IF YOU TYPE <help> IT WILL
// GIVE YOU SOME HELP. OK, IF YOU GET THIS FAR YOUR DOING PRETTY
// GOOD BUT THERE'S MORE.

// NOW RUN OUR STARTUP SCRIPT TO LOAD BNR SW AND RUN THE
// BNR INITIALIZATION AND DIAGNOSTICS SW BY TYPING THE
// FOLLOWING AT THE PROMPT.

-><YOUR BASE DIR/Release/startup.scr

```
#
# Master startup script
#       For release only.  Not for development use.
#
cd “YOUR BASE DIR/Release”
value = 0 = 0x0

# GPU related modules
#       Gary, (except xmemTest and gpuUIInterface by Thomas)
ld < GPU/gpu.o
value = 272625504 = 0x103fef60

# Port Card related modules
#       Ihab
ld < GPU/pmConf.o
value = 272384816 = 0x103c4330
ld < GPU/pmAll.o
value = 272383344 = 0x103c3d70 = _test_count + 0x8c

# SCM related modules and integration tests
```

```
#       Gary, (except generateCell, scuSCoreTest, scuPortCardTest,
#         scuDaisyChainTest by Mark)
ld < SCM/scm.o
value = 272613520 = 0x103fc090


sysInitCtl()



              System Initialization Controller

send my name gpu5 to server
My name echoed is gpu5
Connects to bcars782b thru local port 5131

GPU Diagnostics failed.

// YOU SHOULD BE ABLE TO SEE SOME INFO ON PORT CARD INITIALIZATION
// AND DIAGNOSTICS AROUND HERE AS WELL. IT DOES NOT APPEAR HERE
// BECAUSE THERE WAS NO PORT CARD ON THE WHEN THIS IS GENERATED

SCU interrupt initialization completed.

Switch interrupt monitoring started.

Performing PM Diagnostics/Initialization.

 Error while Access Slot Number : 1
PM Diagnostics/Initialization failed.

SCU Initialization Complete.

value = 0 = 0x0

->  // NOW LETS CHANGE THE BOOT PARAMETERS PERMANENTLY IN THE GPU's FLASH CARD ID.
-> flashBootParamsSet

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device          : ei
processor number     : 0
host name            : bcars782b                                  // YOUR HOST NAME
file name            : /bnr/3rdparty/glan_sw/WindRiver/vw/config/gpu2960/vxWorks  // BASE DIR/Release/GPU/vxWorks[1]
inet on ethernet (e) : 47.53.32.8:ffff0000                        // THE IP ADDRESS YOU SUPPLY FOR GPU
inet on backplane (b):
host inet (h)        : 47.53.32.1                                 // THE IP ADDRESS OF HOST WORKSTATION
gateway inet (g)     :
user (u)             : nobody                                     // THE USER ID YOU USE TO DOWNLOAD SW
ftp password (pw) (blank = use rsh): nobody                       // PASSWORD CORRESPONDING TO ABOVE
flags (f)            : 0x8
target name (tn)     : gpu5                                       // THE HOST NAME OF YOUR GPU
startup script (s)   :
other (o)            :


Card ID EEPROM updated with the following new boot parameters.

boot device          : ei
processor number     : 0
host name            : YOUR HOST NAME
file name            : YOUR BASE DIR/Rease/GPU/vxWorks
inet on ethernet (e) : THE IP ADDRESS YOU SUPPLY FOR GPU
host inet (h)        : THE IP ADDRESS OF YOUR HOST WORKSTATION
user (u)             : THE USER ID YOU USE TO DOWNLOAD SW
ftp password (pw)    : PASSWORD CORRESPONDING TO ABOVE
flags (f)            : 0x8
target name (tn)     : THE HOST NAME OF YOUR GPU

value = 0 = 0x0
->
```

1. The absolute path of the vxWorks image file.

```
*****************WE'RE DONE!!!**************************************************
```

For more information regarding the standard VxWorks booting process, boot ROM commands, boot parameters or configuring a host workstation to support a VxWorks target please see the "VxWorks Programmer's Guide", Chapter 2.

4.0 Startup Scripts

The same scripts still exist but have been modified for this release.The script is named 'startup.scr'. It is intended to be the script that performs all of BNR's SW downloading and initialization.The other script which is in the release directory, 'loadup.scr' performs the same duties as 'startup.scr' plus some additional initialization duties for switch debugging or demoing. People using this should be warned that correctness of this script is only guaranteed up to and including the call to sysInitCtl(). We are not too familiar with anything beyond that point and are unsure that it is still working correctly. It does not have any problems finding or loading SW but there could still be some differences that we do not understand.

WARNING: Depending on when you received the BNR portion of the SW, the script 'loadup.scr' may or may not download the BNR SW in the correct order. If it does not load these modules successfully, it should be edited to load the BNR modules in the same order as that seen in 'startup.scr'.

Here is the startup script that is shipped along with the Release SW, note that the following path has to be changed:

```
#
# Master startup script
#      For release only.  Not for development use.
#
# the following line contains the base directory where the switch software are installed,
# change it to the absolute path of your base directory **********************************
cd "/bnr/3rdparty/glan_sw/src/Release"

# GPU related modules
#      Gary, (except xmemTest and gpuUIInterface by Thomas)
ld < GPU/gpu.o

# Switch Configuration
# Ihab and Gary
ld < switchConf.o

# Port Card related modules
#      Ihab
ld < GPU/pmAll.o

# SCM related modules and integration tests
#      Gary, (except generateCell, scuSCoreTest, scuPortCardTest,
#        scuDaisyChainTest by Mark)
ld < SCM/scm.o


sysInitCtl()
```

Additional object files and function calls for initialization can be added freely to the startup script file, providing the above object files are loaded first. The switch will not function properly without these object files.

5.0 Man Pages Usage

The man pages for BNR developed SW are very similar to those found on-line for VxWorks and in their reference manual. They were generated automatically using a couple of scripts from WindRiver using only the comments in the corresponding source code. On-line man pages for the BNR SW exist for the modules in gpu.o, scm.o, gpuVW.o, pmAll.o and gpuVW_res.o. For a list of these modules, see Section 2.0 on page 4, "New Release Contents".

The man pages can be found in: YOUR_BASE_DIR/Release/man

It is recommend to either add the above path to your man path or add the following alias to your .cshrc (or wherever):

alias cnsman   man -M YOUR_BASE_DIR/Release/man

If you want to view the man pages for any of the modules just enter:

cnsman [1] moduleName   eg. cnsman scuDrv

This will list all the functions in the module (which would be useful if you can't remember a function name or the module it is in) as well as how to use them and a brief overview of the entire module.

If you want to view the man pages for a particular function enter:

cnsman [2] functionName   eg. cnsman scuCellXferToCore

6.0 Upgrade

6.1 S/W Upgrade

Future S/W releases will be in the form as gzip'ed tarred files. Small patches may be available as well. Note that any minor upgrade will require a temporary shutdown of the switch due to the fact that vxWorks dynamically links objects and upgrading object modules during operation will very likely crash the system.

6.2 GPU RAM - Upgrading/Changing Size

It may be desirable to change the amount of RAM on the GPU. This section describes how to determine the amount of GPU RAM installed, change the amount of GPU RAM and re-configure VxWorks to make use of a different amount of RAM.

The GPU is designed to use either 4 1M DRAM SIMMs or 4 4M DRAM SIMMs for a total of 4 and 16M respectively. The SIMMs should be 30-pin with 9 data bits.

6.2.1 Determining Amount of GPU RAM Installed

The amount of GPU RAM installed can be determined by examining the part numbers on the SIMMs. The 1M parts have part numbers ending in '1000BJ' while the 4M parts have part numbers ending in '4000BJ'. If it is necessary to acquire a set of 4M parts, one choice would be Samsung's KM41C4000BJ-7 or KM41C4000BJ-6 (70ns or 60ns parts will do). Other suppliers will also do.

Please note that although the GPU has 8 DRAM SIMM slots, it has only been debugged to use the first 4 slots.

6.2.2 Changing DRAM SIMMs

In order to remove an existing DRAM SIMM, two tabs, visible through circular holes at either end of the SIMM, must be depressed before the SIMM can slid out of its slot.

6.2.3 Reconfiguring VxWorks for Different RAM Sizes

VxWorks learns of the amount of RAM available to it on a GPU by reading a location in the GPU's Card ID Flash which indicates the amount of RAM available to VxWorks. In order to change this amount, function *flashSetRamSize*() should be used. For more information on *flashSetRamSize*(), please see the man page either on-line or later in this document. For example, in order to change the RAM size to 8M, at the VxWorks prompt, simply type flashSetRamSize(0x800000) and wait until the VxWorks prompt returns (it may take several seconds) then reboot the GPU so that VxWorks can make use of the additional RAM.

WARNING: BNR SW only supports use of up to 8M of GPU RAM. This is in order to make use of a faster i960 assembly call instruction which uses relative addressing which reaches only 8M. Do not configure a GPU to use more than 8M of RAM, even if 16M is installed, unless otherwise notified. At the current time neither Harvard University nor BNR can foresee the need to use more than 8M of RAM on the GPUs. Unfortunately, 4 2M SIMMs can not be purchased since the 2M SIMM parts do no exist.

7.0 Support

Please contact Ihab Mahna[ihab@bnr.ca] (for PM related stuff), Thomas NG [ngt@bnr.ca] (UI, TLG) or Gary Depelteau [delpeltea@bnr.ca] (SCU, GPU) if you have any questions or problems.

8.0 List of Functions

The following is a list of functions in the Creditnet Switch S/W (not in alphabetical order):

8.1 GPU

These functions can be found in gpu.o and gpuVW.o

8.1.1 Flash Memory Functions

fDramTest()	-	Perform a Fast memory test
flashAddRamSize()	-	Flash Add RAM Size.
flashBootParamsSet()	-	Flash Boot Parameter Set.
flashCardIdUsage()	-	Flash Card ID Usage.
flashCheckErase()	-	Check flash bank was erased.
flashCopyFromRam()	-	Copy nbytes from RAM to a flash bank.

flashDataPoll()	-	Polls a flash location.
flashDLoad()	-	Download a file from a workstation.
flashErase()	-	Erase a flash bank.
flashProgAltBank()	-	Program alternate flash bank.
flashSetRamSize()	-	Flash Set RAM Size.

8.1.2 GPU Testing Functions

gpuCardIdTest()	-	GPU Card ID Test
gpuCntrlRegTest()	-	GPU Control Register Test
gpuDiagCtl()	-	GPU Diagnostics Controller
gpuDipSwTest()	-	GPU DIP Switch Test
gpuEDramTest()	-	GPU Extended DRAM Test
gpuEepromTest()	-	GPU EEPROM Test
gpuErrRegTest()	-	GPU NMI Error Source Register Test
gpuEtherTest()	-	Test the i82596ca ethernet controller
gpuExtBusTest()	-	GPU External Bus Test
gpuFlSwapTest()	-	GPU Flash Swap Test
gpuSccTest()	-	Test the 85C30 SCC on the GPU
gpuTimIntTest()	-	GPU Real Time Clock Test

8.1.3 NMI and General Interrupts Functions

gpuNmiInit()	-	GPU NMI Initialization
gpuNmiInitIsr()	-	GPU NMI Initialization ISR
gpuNmiMon()	-	GPU NMI Monitor
gpuNmiMonStop()	-	Stop NMI counter monitoring
gpuNmiMonStart()	-	Start NMI counter monitoring
gpuNmiMonTask()	-	The GPU NMI Monitoring Task
intCntrMon()	-	Interrupt Counters Monitor
intMonInit()	-	Initialize Interrupt monitoring
intMonPerSet()	-	Set the Interrupt Monitor Period (seconds)

8.1.4 Lattice Chip Programming Functions

ispEnValCalc()	-	Calculate the ISP Programming Register Value
ispProgram()	-	Programs an ISP Lattice Device
ispRead()	-	Reads an ISP Lattice Device
ispVerify()	-	Verifies an ISP Lattice Device

8.1.5 Serial Port Functions

sccInit()	-	SCC initialization.
sccOutputChar()	-	Output a single character via the SCC (Serial Communication).
sccPrintHexWord()	-	Output a 32-bit hexadecimal number.
sccPrintString()	-	Output a string via the SCC.
sccWait()	-	Poll until SCC is no longer transmitting a char.

8.1.6 Watchdog Functions

wdFeeder()	-	Watchdog feeder function
wdFeederHookAdd()	-	Add hook routine to WD feeder task
wdFeederHookDelete()	-	Delete WD feeder task hook routine
wdFeederHookShow()	-	Displays WD hook routines
wdFeederInit()	-	Initialize watchdog feeder library
wdFeederTask()	-	Watchdog feeder task
wdRestartFeeder()	-	Re-enable the feeding of the watchdog
wdStopFeeder()	-	Force the disabling of watchdog feeds

8.2 SCM

These functions can be found in scm.o:

8.2.1 Miscellaneous functions

scmMicrosecondWait()	-	Wait n microseconds
scuACAlarmISR()	-	SCU AC Alarm ISR
scuCbtErrISR()	-	SCU Cell Bus-tap Error ISR
scuCoreClkAlarmISR()	-	SCU Core Memory Clock Alarm ISR
scuCoreParErrISR()	-	SCU Core Memory Parity Error ISR
scuFanAlarmISR()	-	SCU Fan Alarm ISR
scuIntInit()	-	SCU Interrupt Initialization
scuIntMon()	-	SCU Interrupt Monitor
scuItTimCompISR()	-	SCU Interval Timer Timing Complete ISR
scuPSAlarmISR()	-	SCU Power Supply Alarm ISR
scuSDiagCtl()	-	SCU Start-up Diagnostics Controller
scuSRegCheck()	-	SCU Register Bit Check
scuStaleCadrISR()	-	SCU Stale CADR ISR
scuVerify()	-	SCU Verify

8.2.2 Cell Manipulation and Share buffer functions

crc16()	-	CRC16 Calculation
cellCrc8()	-	8 bit CRC calculation
generateCell()	-	Generate Cell
printCellStruct()	-	Print Cell Structure
scu64BCellDecode()	-	SCU 64-Byte Cell Decode to 56-Bytes
scuCellCpyRamToSb()	-	Copy Cell GPU RAM->Shared Buffer
scuCellCpySbToRam()	-	Copy Cell Shared Buffer->GPU RAM
scuCellXferFromCore()	-	SCU Cell Transfer from Core Memory
scuCellXferToCore()	-	SCU Cell Transfer to Core Memory
scuCellXferToFromCore()	-	SCU Cell Transfer to/from Core
scuCoreCellRead()	-	SCU Core Cell Read
scuCoreCellWrite()	-	SCU Core Cell Write
scuGetCellsFromCore()	-	SCU Get Cells From Core
scuGetCellFromCore()	-	SCU Get Cell From Core
scuMap8To7()	-	SCU Map 8 TO 7 (byte)
scuPutCellInCore()	-	SCU Put Cell In Core
scuReadXCells()	-	Read X Cells from the Core Memory
scuSMSharBuffTest()	-	SCU SM Diag. Shared Buffer Test
scuSSBByteCheck()	-	SCU Shared Buffer Byte Check
scuVerCorePat()	-	SCU Verify Core Pattern
scuWriteXCells()	-	Write X Cells into the Core Memory
swapCell()	-	Swap Cell

8.2.3 CIF Functions

scuCifCntlRegRead()	-	SCU Read from CIF Control Register
scuCifCntlRegWrite()	-	SCU Write to CIF Control Register
scuCifInit()	-	SCU Cell Interface Initialization
scuCifHPWFifoStatRead()	-	SCU CIF HP W_IVC FIFO Status Read
scuCifLPWFifoStatRead()	-	SCU CIF LP W_IVC FIFO Status Read
scuCifRdPoll()	-	SCU CIF Read Cycle Poll
scuCifRdWrPoll()	-	SCU CIF Read/Write Cycle Poll
scuCifRFifoStatRead()	-	SCU CIF R_IVC FIFO Status Read
scuCifRivcFifoRead()	-	SCU CIF R_IVC FIFO Read
scuCifStatRegRead()	-	SCU Read from CIF Status Register
scuCifWrPoll()	-	SCU CIF Write Cycle Poll
scuCifWivcFifoRead()	-	SCU CIF W_IVC FIFO Read
scuSCifDataPathTest()	-	SCU CIF Data Path Test
scuSCifFifoTest()	-	SCU CIF FIFO Test

scuSFifoStatCheck()	-	SCU CIF FIFO Status Check
scuCifRcvCrc8ErrISR()	-	SCU CIF Receive CRC8 Error ISR
scuSCifCrc8Test()	-	SCU CIF CRC8 Test
scuSMCifRegTest()	-	SCU SM CIF Register Test

8.2.3.1 IVC Monitor functions

scuRFifoAlFullISR()	-	SCU R_IVC FIFO Almost Full ISR
scuRFifoMidISR()	-	SCU R_IVC FIFO Mid-range ISR
scuRivcLutInit()	-	SCU R_IVC Lookup Table Initialization
scuRivcLutSet()	-	SCU R_IVC Lookup Table Entry Set
scuRivcLutSetup()	-	SCU R_IVC Lookup Table Setup
scuSMWivcLutTest()	-	SCU SM Diag. W_IVC Lookup Table Test
scuSMRivcLutTest()	-	SCU SM Diag. R_IVC Lookup Table Test
scuWFifoHPAlFullISR()	-	SCU HP W_IVC FIFO Almost Full ISR
scuWFifoHPMidISR()	-	SCU HP W_IVC FIFO Mid-range ISR
scuWFifoLPAlFullISR()	-	SCU LP W_IVC FIFO Almost Full ISR
scuWFifoLPMidISR()	-	SCU LP W_IVC FIFO Mid-range ISR
scuWivcLutInit()	-	SCU W_IVC Lookup Table Initialization
scuWivcLutSet()	-	SCU W_IVC Lookup Table Entry Set
scuWivcLutSetup()	-	SCU W_IVC Lookup Table Setup

8.2.4 CDM functions

scuCdmCntlRegRead()	-	SCU Read from CDM Control Register
scuCdmCntlRegWrite()	-	SCU Write to CDM Control Register
scuCdmStaleCadrRegRead()	-	SCU CDM Stale CADR Reg. Read
scuCdmStatRegRead()	-	SCU Read from CDM Status Register
scuChalkSearchDisable()	-	SCU CDM Chalkmark Search Disable
scuChalkSearchEnable()	-	SCU CDM Chalkmark Search Enable
scuChalkSearchReset()	-	SCU CDM Chalkmark Search Reset
scuCurChalkRead()	-	SCU Current Chalkmark Read
scuCurChalkSet()	-	SCU Current Chalkmarks Set
scuIvcSetup()	-	SCU IVC Setup
scuIvcSet()	-	SCU Single IVC Setup
scuIvcInit()	-	SCU IVC Initialization
scuSCdmCmSrchTest()	-	SCU CDM Chalkmark Search Test
scuSCdmCmUpdTest()	-	SCU CDM Chalkmark Updating Test
scuSCdmMonFATest()	-	SCU CDM Monitoring and FA Test
scuSChalkmarkCheck()	-	SCU Chalkmark Check
scuSMCdmConCapTest()	-	SCU SM CDM Configuration/Capture Test
scuSMCdmRegTest()	-	SCU SM CDM Register Test

8.2.5 CTC functions

scuCtcInit()	-	SCU Cell Time Counter Initialization
scuCtcIntTimProg()	-	SCU CTC Interval Timer Program
scuCtcIntTimRead()	-	SCU CTC Interval Timer Register Read
scuCtcRealTimProg()	-	SCU CTC Real-time Timer Program
scuCtcStartRegRead()	-	SCU CTC Start Register Read
scuCtcStatRegRead()	-	SCU CTC Status Register Read
scuCtcStopRegRead()	-	SCU CTC Stop Register Read
scuSCtcIt1Test()	-	SCU CTC Interval Timer Test 1
scuSCtcIt2Test()	-	SCU CTC Interval Timer Test 2
scuSMCtcRegTest()	-	SCU SM CTC Register Test

8.2.6 DBM functions

scuDbmCapture()	-	SCU Distribution Bitmap Capture
scuDbmInit()	-	SCU Distribution Bitmap Initialization
scuDbmSet()	-	SCU Distribution Bitmap Entry Set
scuDbmSetup()	-	SCU Distribution Bitmap Setup

scuSMDbmRamTest()	-	SCU SM DBM RAM Test
scuSMWDbmRamTest()	-	SCU SM Working DBM RAM Test
scuWDbmCapture()	-	SCU Working Distribution Bitmap Capture
scuWDbmIntegTest()	-	SCU wDBM Integration Test
scuWDbmMultiCastTest()	-	SCU wDBM Multi-Cast Test
scuWDbmWalkingOneTest()	-	SCU wDBM Walking One Test
scuWDbmWalkingZeroTest()	-	SCU wDBM Walking Zero Test

8.2.7 FLM Functions

fakeFLMInit()	-	Initialize FLM emulator
fakeFLMGetNextCADR()	-	get the next address out of the FLM emulator
scuFlmCmbfCountRead()	-	SCU FLM CMBF Counter Read
scuFlmCntlRegRead()	-	SCU Read from FLM Control Register
scuFlmCntlRegWrite()	-	SCU Write to FLM Control Register
scuFlmEmptyISR()	-	SCU FLM Free List Empty ISR
scuFlmFailISR()	-	SCU FLM Free List Fail ISR
scuFlmFLArrayPush()	-	SCU FLM Free List Array Push
scuFlmFLCInit()	-	SCU FLM Free List Complete Initialization
scuFlmFLPush()	-	SCU FLM Free List Push
scuFlmFreeListInit()	-	SCU FLM Free List Initialization
scuFlmFullISR()	-	SCU FLM Full ISR
scuFlmHalt()	-	SCU FLM Halt
scuFlmHaltClear()	-	SCU FLM Halt Clear
scuFlmInit()	-	SCU FLM Initialization
scuFlmPrime()	-	SCU FLM Prime
scuFlmReset()	-	SCU FLM Reset
scuFlmStatRegRead()	-	SCU Read from FLM Status Register
scuFlmWrapISR()	-	SCU FLM Free List Wrap ISR
scuSFlmBasicTest()	-	SCU Flm Basic Functionality Test
scuSMFlmRamStatTest()	-	SCU SM Diag. FLM RAM and Status Test
scuSMFlmRegTest()	-	SCU SM FLM Register Test
scuSFlmIntegTest()	-	SCU FLM Integrity Test

8.2.8 Miscellaneous Initialization and Configuration

scuCdmInit()	-	SCU Cell Distribution Manager Initialization
scuDefDSInit()	-	SCU Default Data Structure Initialization
scuHwInit()	-	SCU Hw Initialization Initialization
scuInitCtl()	-	SCU Initialization Control
scuPifInit()	-	SCU Processor Interface Initialization
sysInitCtl()	-	System Initialization Controller

8.2.9 Miscellaneous SCM/SCU Tests

scmMarch6Test()	-	SCM March 6 Memory Test
scmRegTest()	-	SCM Register Test
scmWalkA1Test()	-	SCM Address-bit Walking One Test
scmWalkD0Test()	-	SCM Data-bit Walking Zero Test
scmWalkD1Test()	-	SCM Data-bit Walking One Test
scmChkDramParity()	-	SCM Check for GPU DRAM Parity Error
scmChkFlashSig()	-	SCM Check Flash Signature
scuCMCompareCell()	-	SCU CM Compare Cell
scuCMTestEntireCore()	-	SCU CM Test Entire Core
scuCMTestGeneratePatterns()	-	SCU CM Test Generate Patterns
scuCMTestLostAddresses()	-	SCU CM Test Lost Addresses
scuCMTestOneAddress()	-	SCU CM Test One Address
scuCMWalkingTest()	-	SCU CM Walking Test
scuDaisyChainTest()	-	SCU Daisy Chain Test
scuFifoIntegTest()	-	SCU FIFO Integration Test
scuFifoSeqTest()	-	SCU FIFO Sequential Test

scuHVCdmPETest()	-	SCU HV CDM PE Test (N.T.)
scuHVCifPETest()	-	SCU HV CIF PE Test (N.T.)
scuHVCrmBFStatTest()	-	SCU HV CRM BF Status Test (N.T.)
scuHVCrmPETest()	-	SCU HV CRM PE Test (N.T.)
scuHVCtcItPETest()	-	SCU HV CTC IT PE Test (N.T.)
scuHVDS1TogCheck()	-	SCU HV DIP Switch Toggle Check
scuHVEncDecTest()	-	SCU HV Encoded/Decoder Test
scuHVFlmCmbfPETest()	-	SCU HV FLM CMBF Count PE Test (N.T.)
scuHVIdleTest()	-	SCU HV Idle Test (N.T.)
scuHVIdLogTest()	-	SCU HV ID/LOG Test
scuHVPMIntTest()	-	SCU HV PM Interrupt Test (N.T.)
scuHVPMResetTest()	-	SCU HV PM Reset Test
scuHVWCadrPETest()	-	SCU HV W_CADR PE Test (N.T.)
scuInitPortCardTest()	-	SCU Initialization Port Card Test
scuSCoreTest()	-	SCU Start-up CM Test
scuSMPbusRegTest()	-	SCU SM P-Bus Register Test

8.3 PM

These functions can be found in pmAll.o:

8.3.1 General PCC functions

checkPortCardExist()	-	Checks if a port card is present in the specified slot
displayChecksums()	-	display portcard checksums
repeatModeOff()	-	turn PCC repeat mode off
repeatModeOn()	-	turn PCC repeat mode on
runPccService()	-	Run a PCC service
setPort()	-	Sets the current port - Required for many PCC functions
ur()	-	Update ROM

8.3.2 Memory Access

downloadToPcc()	-	Download to PCC
dp()	-	Dump PCC memory to the screen with specified address, width, and length
fillPcc()	-	Fill pcc memory.
rp()	-	Read from a PCC memory location
wp()	-	Write to a PCC memory location

8.3.3 PCC test

pccAddressTest()	-	PCC Address Test
pccMarch6Test()	-	PCC March 6 Memory Test
pccWalkD1Test()	-	PCC Data-bit Walking One Test
pccWalkD0Test()	-	PCC Data-bit Walking Zero Test
runPccMemoryTest()	-	Run a PCC memory test
testChecksums()	-	test portcard checksums

8.3.4 Scheduler specific

rs()	-	Read Scheduler
ws()	-	Write Scheduler
resetScheduler()	-	Reset Scheduler

9.0 Memory maps

Individual memory maps for SCM and PM can be found in the following files:

SCM - scu.h, scu.h.ramAdrs

PCM - PCCregions.h, SCHregions.h, mcpPcMemoryMap.h, pcMemoryMap.h and schMemoryMap.h.

10.0 Glossary

ATM	:Asynchronous Transfer Mode
BNR	:Bell Northern Research
BTL	:Bus Transceiver Logic
CCB	:Cell Control Bus
CDM	:Cell Departure Monitor
CIF	:Cell Interface
CM	:Core Memory or Credit Manager on PCC
CRM	:Core Memory Resource Manager
DM	:Data Manager
FIFO	:First-In First-Out
FLM	:Free List Manager
GPU	:General Processor Unit
IVC	:Internal Virtual Connection
P-BUS	:Processor Bus
PCC	:port Card Controller
PM	:Port Module
SCC	:Serial Communication Controller
SCH	:Scheduler on PCM
SCM	:Switch Control Module (consists of a GPU and an SCU)
SCU	:Switch Control Unit
TBD	:To Be Determined
TC	:Terminator Card
TLG	:Test Load Generator