

```

### Log file for this test: /afs/hep.wisc.edu/cms/RCTlog/daffodil/RC_2004-08-26
.log ### Location of log file

#####
##### RC Test 5a - Backplane data paths #####
### Test run on 2004-08-26_17:51:21
### HOST computer is: daffodil
### Run in vmedia kumac: rc_backplane_path_slot2.txt
Please fill in the backplane data paths CHECKLIST.
#####

!!!! RC to be tested has to be in slot 2 !!!! IMPORTANT!
Device to open: /dev/btp96
Device to open: /dev/btp160 SBS successfully booted and
Device to open: /dev/btp64 talked to the cards in the crate
Enter command (help for usage)> The CCC, 1 EIC in slot 2, 6 RCs and
RCT boot succeeded with 9 cards. the RC to be tested in slot 2 should be plugged in
Enter command (help for usage)> Enter command (help for usage)> Enter command (h
elp for usage)>
Zero memories first.
Device to open: /dev/btp96
Device to open: /dev/btp160
Device to open: /dev/btp64
RCTCrate::initialize() : vmeReset() successful
RCTCrate::initialize() : Defined RCTClockControlCard 10000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 12000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 14000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 16000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 18000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 1b000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 1d000000
RCTCrate::initialize() : Defined RCTReceiverCard with address 1f000000
RCTCrate::initialize() : Defined RCTElectronIsolationCard with address 15000000
rctCrateTest: initialize() succeeded
Cards in the crate are: 54ba
RCTCrate::doZeroPatternTest() : Loading RC (f500, 12000000)
RCTCrate::doZeroPatternTest() : Verifying RC (f500, 12000000)
RCTCrate::doZeroPatternTest() : Loading RC (fa00, 14000000)
RCTCrate::doZeroPatternTest() : Verifying RC (fa00, 14000000)
RCTCrate::doZeroPatternTest() : Loading RC (0, 16000000)
RCTCrate::doZeroPatternTest() : Verifying RC (0, 16000000)
RCTCrate::doZeroPatternTest() : Loading RC (f600, 18000000)
RCTCrate::doZeroPatternTest() : Verifying RC (f600, 18000000)
RCTCrate::doZeroPatternTest() : Loading RC (f700, 1b000000)
RCTCrate::doZeroPatternTest() : Verifying RC (f700, 1b000000)
RCTCrate::doZeroPatternTest() : Loading RC (fe00, 1d000000)
RCTCrate::doZeroPatternTest() : Verifying RC (fe00, 1d000000)
RCTCrate::doZeroPatternTest() : Loading RC (f800, 1f000000)
RCTCrate::doZeroPatternTest() : Verifying RC (f800, 1f000000)
RCTCrate::doZeroPatternTest() : Loading EIC (b100, 15000000)
RCTCrate::doZeroPatternTest() : Verifying EIC (b100, 15000000)
rctCrateTest: All tests successful
Now start vmedia script rc_backplane_path_slot2

*****
*** this is vmedia script rc_backplane_path_slot2.txt ***
*****
for this test, the rc has to be in slot 2 Important
Continue <return> ? Exit <Ctrl-D> ? type <return> here
this rc has barcode
Device to open: /dev/btp96
Device to open: /dev/btp160
Device to open: /dev/btp64

```

**Check that the verification doesn't fail. When it does, leave the script (Ctrl-D) and vmedia (exit) and redo >run\_RC\_test5a**

16000006 -> 0000 **Compare this number with the RC bar code**  
Continue <return> ? Exit <Ctrl-D> ? **type <return> here**  
for this test, the crate has to be loaded with all seven rc's **Important**

12000006 -> F565  
14000006 -> FA05  
16000006 -> 0060  
18000006 -> F600  
1B000006 -> F760  
1D000006 -> FE00  
1F000006 -> F820  
12000000 -> 0202  
14000000 -> 0202  
16000000 -> 0202  
18000000 -> 0202  
1B000000 -> 0202  
1D000000 -> 0202  
1F000000 -> 0202

**These values should be read back**

initial setup done.  
start direct paths.

**Repeat:**

7f in \$440000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ? **Check the signals as specified in the checklist - when done type <return> for next signal**  
ff in \$440000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?

7f in \$4c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$4c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$540000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$540000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$5c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$5c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$640000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$640000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$6c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$6c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$740000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$740000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
7f in \$7c0000 -- next ?  
Continue <return> ? Exit <Ctrl-D> ?  
ff in \$7c0000

**Check 4 bits on U125 and 3 bits on U126; pattern 7F should result in 111 1111; pattern FF should result in 000 0000**

direct paths done. hit return to zero memory and exit.

Continue <return> ? Exit <Ctrl-D> ? **type <return> here**

VMEDia> exit **type 'exit' here**

Bye

##### RC Test 5a End #####