Mark6/Flexbuff Effelsberg installation notes

Hardware and Software status

Middle of May a Mark6 that was pre-installed and tested in the VLBI lab in Bonn was shipped to Effelsberg and installed in the Faradayroom. It is running on Debian Squeeze (6.0) and has the following software version installed:

dplane version 1.17-1
mark6-sc-module 0.13-1
python-cplane 1.0.20-1
python-m6utils 1.0.7-2

The two onboard Ethernet ports have connection to the MPIfR network and the eVLBI network. A VNC-Server is running to provide a remote-accessible platform where programs can run locally on the machine without remote logins.

Both VSI ports of the Effelsberg DBBC are connected to an external Fila10G SA. The Mark5B+ gets a copy of the incoming VSI1 from the Fial10G, which is connected to COM2 of the DBBC for remote control via the DBBC server software (v105). The first 10 GE optical fibre adapter of the Fila10G is connected to eth3 of the Mark6 and the second one to the Harrobox just because it was there.

The Mark6 came with 4 disk modules, 32 TB each, that are supposed to stay in Effelsberg and are not being shipped. In this configuration the recorder is used not much different from a Flexbuff.

First tests

First test were done using the d- and cplane software following the Haystack “Mark6 usage examples” that are given on the Mark6 Documentation page:

http://www.haystack.mit.edu/tech/vlbi/mark6/documentation.html

It starts with initializing the modules in each slot, one after each other, and finally combine them into one group. This was done using the "da-client" software

mod_init=2:8:TEST0001:sg:new
mod_init=2:8:TEST0002:sg:new
mod_init=3:8:TEST0003:sg:new
mod_init=4:8:TEST0004:sg:new
group=new:1234
group=open:1234

>> mstat?
<< !mstat?0:0:1234:1:TEST0001/32000/4/8:8:8:31989:32000:open:ready:sg:1234:2:TEST0002/32000/4/8:8:8:31989:32000:open:ready:sg:1234:3:TEST0003/32000/4/8:8:8:31989:32000:open:ready:sg:1234:4:TEST0004/32000/4/8:8:8:31989:32000:open:ready:sg;
>>

An input stream need to be defined from which the data is coming that should be written on the modules. This needed a bit of time, because the actual package size needs to be correct otherwise the data will be rejected.

The Fila10G was configured in TVG mode, 64 MHz clock, VDIF format, payload size of 8192 bytes. Including the 32 byte header each package has then a size of 8224 bytes. However someting is added and the packages that arrive at the Mark6 have 8274 bytes, as one can check using wireshark, or tcpdump.

So, the final input stream that worked to record data was:

input_stream=add:fila10g:vdif:8224:50:42:eth3:172.16.3.2:46227::
input_stream=commit;
input_stream?

>> input_stream?
<< !input_stream?0:0:fila10g:vdif:8224:50:42:eth3:172.16.3.2:46227:0;

172.16.3.2 is the address of the Fila10G and 46227 is the port where it is sending. The parameters can be specified, but are not really used by cplane, it is just grabbing everything that has the right package size from eth3.

Eth3 itself has the 172.16.3.1 and the Fila10G is configure to send the data to this address including the correct arp hardware address.

With this configuration recording and checking the data worked fine:

record=on:10::no000007:test:ef
record?
list?
rtime?
scan_check?
scan_info?

Using jive5ab for recording

Once this was done and the systems seemed to behave like a recorder, jive5ab was installed to test recording with from the Field System.

wget http://www.jive.eu/~verkouter/evlbi/jive5ab-2.6.0.tar.gz

tar xzf jive5ab-2.6.0.tar.gz
cd jive5ab-2.6.0
make B2B=64
su
mark B2B=64 install

This will compile a 64 bit jive5ab version for your Mark6/Flexbuff that is immediately ready to run.

/opt/jivemark5a -m 3

Field System version FS-9.11.8 already knows about the Flexbuff and it can be configured in equip.ctl. The IP address goes into the mk5add.ctl, as usual. To configure the Fila10G at the same time as the DBBC the Field System backend option "dbbc/fila10g" was chosen. The recorder is still a Mark5B+, which does not depend on the Fial10G configuration, but just gets a copy of the incoming VSI data.

To allow parallel recordings from a single Field System the snp- and prc-files for each expriment are processed with Simon Caseys script fila5cflex-v005.py (http://www.jive.eu/~verkouter/flexbuff/). The cmd2fila10g commands are ignored, but the sy=cmd2flexbuff.py allow to start, stop, and test the Flexbuff recordings and the same time as the normal Mark5B+ ones. With that a typical setup-section in a FS procedure file look like:

```
fila10g_mode=0xFFFFFFFF,,16.000
mk5b_mode=ext,0xFFFFFFFF,,16.000
sy=cmd2flexbuff.py "net_protocol=udps:32000000:256000000:4; mtu=8070;
record = mk6:0 "
sy=cmd2flexbuff.py "set_disks=1:2:3:4"
sy=cmd2flexbuff.py mode=VDIF_8000-512-16-2
```

This sets the basic parameters for the FiLa10G and the FlexBuff, that are needed for successful FlexBuff recordings. The default data port for jive5ab is 2630, therefore the FiLa10g has to be configured to send data to this port (See Fila10G notes below).The corresponding sections in the snap-file contain:

```
!2015.155.14:51:50
preob
!2015.155.14:52:00
disk_pos
disk_record=on
sy=cmd2flexbuff.py record = on : n15c2_Ef_No0018 ;
disk_record
data_valid=on
midob
!2015.155.15:00:00
data_valid=off
disk_record=off
sy=cmd2flexbuff.py record = off
disk_pos
postob
checkmk5
```

The last jive5ab 2.6.1-dev development version also supports "scan_set" and "scan_check" and "disk2file" commands. However for paralell recordings it is simpler to use the disk2filefb.py from S. Casey.

FiLa10G configuration

FS-9.11.8 supports direct command to be send to the FiLa10G. It actually sends the commands to the DBBC server and the server does the communication via a serial port from the DBBC PC. The FS comes with a default configuration for the FiLa10G that is stored in fila10g.prc. There is a 32 MHz clock section and a 64 Mhz clock section. The current 32 MHz settings at Effelsberg are:

```
define  fila10g_32    15146143412
"customize inputselect, IPs, gateways, MACs, and timesync
"for your station
fila10g=inputselect vsi1
fila10g=vsi_samplerate 32000000
fila10g=arp off
```

```
fila10g=tengbcfg eth0 ip=172.16.3.2 gateway=172.16.3.1
fila10g=tengbcfg eth0 mac=00:60:dd:44:46:c3
fila10g=tengbcfg eth0 nm=27
fila10g=tengbcfg eth1 ip=10.88.1.125 gateway=10.88.1.126
fila10g=tengbcfg eth1 mac=ba:cd:af:e4:be:e1
fila10g=tengbcfg eth1 nm=27
fila10g=splitmode off
fila10g=vdif_station Ef
fila10g=destination 0 172.16.3.1:2630
fila10g=destination 1 10.88.1.126:2630
fila10g=reset
"fila10g=timesync
enddef
```

The internal GPS receiver of the FiLa10G SA is connected to an antenna and therewith gets the GPS time internally. Current status is, that it was synced on May 28 before the session started and is still in sync since then (June 9, 2015). Not a really long time yet, but it was used and reconfigured for every expiment since then and about 60 TB were recorded to the FlexBuff.