VLBA Sensitivity Upgrade Memo 39

Dual RDBE IF and Channel Assignments

Walter Brisken & Matthias Bark

Draft 12 April 2013

1 Introduction

The VLBA Sensitivity Upgrade Project had from the beginning the requirement to record channels from up to 4 IFs simultaneously. Implementation of this will be through a combination of a flexible IF matrix switch (the T450 or “4 × 4 switch”), two RDBEs (each with two samplers), a network switch, and at least one data recorder. This memo describes how the RDBEs can be configured for all cases with up to four active Intermediate Frequency (IF) signals.

The combining of data from the two RDBEs in the network switch requires use of a data format where there is a 1 to 1 mapping of data frames to ethernet frames and where each data frame is sufficiently self-identifying such that the correlator can assemble the baseband data for correlation properly. The VLBI Data Interchange Format (VDIF; see http://vlbi.org/vdif/) has been chosen for this purpose and is currently being implemented in the Digital Down-Converter (DDC) personality. It is recommended here to only make use of the single-channel-per-thread variant of the VDIF format.

VDIF allows per channel specification of sample rate. The scheme outlined in this document will allow this full generality though not all components of the full system (RDBE firmware and correlator specifically) may be able to support such general mode from the outset.

2 Notation

1. N is the maximum number of channels each RDBE can produce. For the real-sampled initial DDC personality $N = 4$. A future complex-sampled DDC personality is likely to yield $N = 8$.

2. α, β, γ, and δ are the abstract names of the four IFs. There will be some mapping from these to physical IFs A, B, C and D based on the number of channels each physical IF will produce. The mapping between physical IFs and the RDBE inputs is accomplished in a fully general manner with the T450 matrix switch.

3. n_α, n_β, n_γ, and n_δ represent the number of channels to attach to α, δ, γ, and δ, respectively. Here it is stipulated that $n_\alpha \geq n_\beta \geq n_\gamma \geq n_\delta$.

4. The total number of channels is $n = n_\alpha + n_\beta + n_\gamma + n_\delta$. For the two RDBE case, $n \leq 2N$.

5. An RDBE configuration is represented as $(P, Q : r, s)$ where P and Q are the IFs to send to RDBE analog inputs 0 and 1 respectively and r and s are the number of channels to assign to those IFs. In some two- and three-IF configurations one IF is fed into both RDBEs. In these circumstances RDBE0 will take the lowest numbered channels and RDBE1 will take the highest numbered channels. A configuration of NA indicates “Not Allowed”; there is no way to support the channel distribution.

6. $\emptyset$ indicates an unconnected RDBE input.

3 Assignment table

This section tabulates IF switch settings and channel assignments for the case of two RDBEs each capable of generating N digital downconverter channels. The table covers cases for RDBEs with any value of N but not mixed cases (where the two RDBEs support different numbers of channels). In all cases where a single RDBE is capable of recording all channels only RDBE0 is used. Effort is made to minimize need to use multiple samplers on one IF, but there is not much room to circumvent this.

Active IFs	Channel distribution	RDBE0 config	RDBE1 config
1	$n_\alpha \leq N$	$(\alpha, \emptyset : n_\alpha, 0)$	
1	$n_\alpha > N$	$(\alpha, \emptyset : N, 0)$	$(\alpha, \emptyset : n_\alpha - N, 0)$
2	$n_\alpha + n_\beta \leq N$	$(\alpha, \beta : n_\alpha, n_\beta)$	
2	$n_\alpha \leq N$ and $n_\alpha + n_\beta > N$	$(\alpha, \emptyset : n_\alpha, 0)$	$(\beta, \emptyset : n_\beta, 0)$
2	$n_\alpha > N$	$(\alpha, \emptyset : N, 0)$	$(\alpha, \beta : n_\alpha - N, n_\beta)$
3	$n_\alpha \leq N$ and $n_\beta + n_\gamma \leq N$	$(\alpha, \emptyset : n_\alpha, 0)$	$(\beta, \gamma : n_\beta, n_\gamma)$
3 (app. B)	$n_\alpha > N$ or $n_\beta + n_\gamma > N$	$(\alpha, \beta : N - n_\beta, n_\beta)$	$(\alpha, \gamma : n_\alpha + n_\beta - N, n_\gamma)$
4 (app. C)	$n_\alpha + n_\delta \leq N$ and $n_\beta + n_\gamma \leq N$	$(\alpha, \delta : n_\alpha, n_\delta)$	$(\beta, \gamma : n_\beta, n_\gamma)$
4	Other cases	NA	NA

Note: There is no possible way to configure the RDBE pair with channels coming from four active IFs if any one IF has N or more channels (i.e., $n_\alpha \geq N$ cannot be accommodated).

4 Hardware implementation

Four IF cables to IF switch to two RDBEs to network switch to Mark5C.

5 Software implementation

The selection of channels and downstream identification of data has software requirements in `sched`, `vex2script`, the executor, and the correlator.

5.1 sched

`sched`'s role is simply to define in a `.vex` file the channels to be observed and verify that the configuration is legal. All channels for a configuration should end up in a FREQ block. Until vex2 is ratified and implemented, a modified Mark5B track layout will be used to describe the bitstreams. The track number slot will be used to hold the thread ID for that channel. The same thread ID should be given to both the sign and magnitude "tracks" here. The `track_frame_format` parameter should be set to `VDIF5032`. The number 5032 refers to the VDIF data frame size to use. Our current plan is to implement VDIF with exactly 5000 bytes of payload data and 32 bytes of header (see appendix A).

Sched will also need to continue to verify that the IF configuration requested by the observer is legal.

5.2 vex2script

`vex2script` will have the responsibility for implementing the information in the above table to assign IF (pairs) to RDBE(s) and assign channels. Because individual channels are self-identifying based on the thread ID mapping, vex2script does not need to coordinate its algorithm for channel assignment with any other code.

5.3 executor

The executor will follow the LO/IF prescription provided by `vex2script`.

5.4 DiFX

The correlator will use the `.vex` file channel-to-thread mapping in the decoding of data. No dual-RDBE specific changes should be needed at the correlator.

A VDIF data frame size choice

This appendix documents the choosing of 5000 byte VDIF data payload sizes. The Mark5C recorder 10 GbE interface assumes no packets larger than 9000 bytes will be received, so that should be considered a hard limit, regardless of the UDP standard which allows 64k.

VDIF packet size numerology is a bit complicated and has implications for minimum data rates. The payload size for a VDIF packet must be a multiple of 8 bytes. I assume here that 2 bits per sample will be the default, but more won't change the logic (1-bit data will challenge the low data rates even more, but no one would consider observing with only 1 bit samples at low bandwidths!)

In one VDIF thread containing 1 channel one must have a multiple of 32 samples in one frame. Since there must be an integer number of frames per second, and by convention (which no one is considering changing here!) sample rates are 2^k MHz, there must be no fewer than 500000 samples per second to legally frame one 2-bit VLBI channel into one VDIF thread. This puts a lower bound of 250 kHz bandwidth on the DDC in this mode (the same conclusion would be reached for Real or Complex data).

500000 samples will fit into 125000 bytes, so for this mode there would be 5^p frames per second, each consisting of $8 \times 5^{6-p}$ bytes. The possibilities are thus:

Frames per second	frame size (B)	comment
1	125000	Too big
5	25000	Too big
25	5000	Just right?
125	1000	3% framing overhead
625	200	16% framing overhead

The DDC personality wideband limit is 128 MHz bandwidth (256 Msps) resulting in 32000000 bytes per second. At 5000 bytes per packet, this yields 6400 frames per second which is not a problem. Note that the WIDAR firmware was developed to support 1000 byte data frame payloads.

B Example three IF vex file

Below is a portion of a hypothetical dual RDBE vex file:

```
$FREQ;
def DualRDBE8x32MHz;
     chan_def = :  2180.00 MHz : U :  32.00 MHz : &CH01 : &BBC01 : &U_Cal; *Rcp
     chan_def = :  2212.00 MHz : U :  32.00 MHz : &CH02 : &BBC02 : &U_Cal; *Rcp
     chan_def = :  2244.00 MHz : U :  32.00 MHz : &CH03 : &BBC03 : &U_Cal; *Rcp
     chan_def = :  2180.00 MHz : U :  32.00 MHz : &CH04 : &BBC04 : &U_Cal; *Lcp
     chan_def = :  2212.00 MHz : U :  32.00 MHz : &CH05 : &BBC05 : &U_Cal; *Lcp
     chan_def = :  8460.00 MHz : L :  32.00 MHz : &CH06 : &BBC06 : &U_Cal; *Lcp
     chan_def = :  8492.00 MHz : L :  32.00 MHz : &CH07 : &BBC07 : &U_Cal; *Lcp
     chan_def = :  8524.00 MHz : L :  32.00 MHz : &CH08 : &BBC08 : &U_Cal; *Lcp
enddef;

$IF;
def DualFreqRPolTone/1;
     if_def = &IF_A : A : R :  2900.0 MHz : L : 1 MHz ; *
     if_def = &IF_C : C : L :  2900.0 MHz : L : 1 MHz ; *
     if_def = &IF_D : D : L :  7900.0 MHz : U : 1 MHz ; *
enddef;

$BBC;
def 8BBCs;
     BBC_assign = &BBC01 :  1 : &IF_A;
     BBC_assign = &BBC02 :  2 : &IF_A;
     BBC_assign = &BBC03 :  3 : &IF_A;
     BBC_assign = &BBC04 :  4 : &IF_C;
     BBC_assign = &BBC05 :  5 : &IF_C;
     BBC_assign = &BBC06 :  6 : &IF_D;
     BBC_assign = &BBC07 :  7 : &IF_D;
     BBC_assign = &BBC08 :  8 : &IF_D;
enddef;

$TRACKS;
def VDIF.8thread2bit;
     track_frame_format = VDIF5032;
     fanout_def = X : &CH01 : sign : 1: 0;
     fanout_def = X : &CH01 :  mag : 1: 0;
     fanout_def = X : &CH02 : sign : 1: 1;
     fanout_def = X : &CH02 :  mag : 1: 1;
     fanout_def = X : &CH03 : sign : 1: 2;
     fanout_def = X : &CH03 :  mag : 1: 2;
     fanout_def = X : &CH04 : sign : 1: 3;
     fanout_def = X : &CH04 :  mag : 1: 3;
     fanout_def = X : &CH05 : sign : 1: 4;
     fanout_def = X : &CH05 :  mag : 1: 4;
     fanout_def = X : &CH06 : sign : 1: 5;
     fanout_def = X : &CH06 :  mag : 1: 5;
     fanout_def = X : &CH07 : sign : 1: 6;
     fanout_def = X : &CH07 :  mag : 1: 6;
     fanout_def = X : &CH08 : sign : 1: 7;
     fanout_def = X : &CH08 :  mag : 1: 7;
enddef;
```

For concreteness, this would result in the following table of channel mappings:

Name	Thread ID	IF	Pol.	Sideband	RDBE	Input
CH01	0	$\alpha =$A	R	U	0	0
CH02	1	$\alpha =$A	R	U	0	0
CH03	2	$\alpha =$A	R	U	0	0
CH04	3	$\gamma =$C	L	U	0	1
CH05	4	$\gamma =$C	L	U	1	1
CH06	5	$\beta =$D	L	L	1	0
CH07	6	$\beta =$D	L	L	1	0
CH08	7	$\beta =$D	L	L	1	0

C Example four IF vex file

Below is a portion of another hypothetical dual RDBE vex file and channel mappings:

```
$FREQ;
def DualRDBE8x32MHz;
     chan_def = :  2180.00 MHz : U :  32.00 MHz : &CH01 : &BBC01 : &U_Cal; *Rcp
     chan_def = :  2212.00 MHz : U :  32.00 MHz : &CH02 : &BBC02 : &U_Cal; *Rcp
     chan_def = :  8460.00 MHz : L :  32.00 MHz : &CH03 : &BBC03 : &U_Cal; *Rcp
     chan_def = :  8492.00 MHz : L :  32.00 MHz : &CH04 : &BBC04 : &U_Cal; *Rcp
     chan_def = :  2180.00 MHz : U :  32.00 MHz : &CH05 : &BBC05 : &U_Cal; *Lcp
     chan_def = :  2212.00 MHz : U :  32.00 MHz : &CH06 : &BBC06 : &U_Cal; *Lcp
     chan_def = :  8460.00 MHz : L :  32.00 MHz : &CH07 : &BBC07 : &U_Cal; *Lcp
     chan_def = :  8492.00 MHz : L :  32.00 MHz : &CH08 : &BBC08 : &U_Cal; *Lcp
enddef;

$IF;
def DualFreqRPolTone/1;
     if_def = &IF_A : A : R :  2900.0 MHz : L : 1 MHz ; *
     if_def = &IF_B : B : R :  7900.0 MHz : U : 1 MHz ; *
     if_def = &IF_C : C : L :  2900.0 MHz : L : 1 MHz ; *
     if_def = &IF_D : D : L :  7900.0 MHz : U : 1 MHz ; *
enddef;

$BBC;
def 8BBCs;
     BBC_assign = &BBC01 :  1 : &IF_A;
     BBC_assign = &BBC02 :  2 : &IF_A;
     BBC_assign = &BBC03 :  3 : &IF_B;
     BBC_assign = &BBC04 :  4 : &IF_B;
     BBC_assign = &BBC05 :  5 : &IF_C;
     BBC_assign = &BBC06 :  6 : &IF_C;
     BBC_assign = &BBC07 :  7 : &IF_D;
     BBC_assign = &BBC08 :  8 : &IF_D;
enddef;

$TRACKS;
def VDIF.8thread2bit;
     track_frame_format = VDIF5032;
     fanout_def = X : &CH01 : sign : 1: 0;
     fanout_def = X : &CH01 :  mag : 1: 0;
     fanout_def = X : &CH02 : sign : 1: 1;
     fanout_def = X : &CH02 :  mag : 1: 1;
     fanout_def = X : &CH03 : sign : 1: 2;
     fanout_def = X : &CH03 :  mag : 1: 2;
     fanout_def = X : &CH04 : sign : 1: 3;
     fanout_def = X : &CH04 :  mag : 1: 3;
     fanout_def = X : &CH05 : sign : 1: 4;
     fanout_def = X : &CH05 :  mag : 1: 4;
     fanout_def = X : &CH06 : sign : 1: 5;
     fanout_def = X : &CH06 :  mag : 1: 5;
     fanout_def = X : &CH07 : sign : 1: 6;
     fanout_def = X : &CH07 :  mag : 1: 6;
     fanout_def = X : &CH08 : sign : 1: 7;
     fanout_def = X : &CH08 :  mag : 1: 7;
enddef;
```

Name	Thread ID	IF	Pol.	Sideband	RDBE	Input
CH01	0	α =A	R	U	0	0
CH02	1	α =A	R	U	0	0
CH03	2	β =B	R	L	1	0
CH04	3	β =B	R	L	1	0
CH05	4	γ =C	L	U	1	1
CH06	5	γ =C	L	U	1	1
CH07	6	δ =D	L	L	0	1
CH08	7	δ =D	L	L	0	1