Protocol Specification

EVLA Correlator Monitor & Control

Virtual Correlator Interface – VCI

RFS Document: **A25201N0000**

Revision :3.16

Sonja Vrcic, October 31, 2013

National Research Council Canada
Herzberg Institute of Astrophysics
Dominion Radio Astrophysical Observatory

P.O. Box 248, 717 White Lake Rd
Penticton, B.C., Canada
V2A 6J9

Table of Contents

List of Figures

List of Tables

List of XML Examples

List of Abbreviations and Acronyms

Abbreviations and Acronyms	
BB	Baseband
BLB	Baseline Board
CBE	Correlator Backend
CC	Correlator Chip
CCC	Correlator Chip Cell, the smallest unit that can be allocated for a single product. Can produce 128 complex lags, i.e. 64 spectral channels.
CCQ	Correlator Chip Quad, consists of 4 CCCs.
CM	Configuration Mapper, software package running on MCCC
CMCS	Correlator Monitoring and Control System
CMIB	Correlator Module Interface Board
CPCC	Correlator Power Control Computer
DP	Data Path
EMCS	EVLA Monitor & Control System.
EVLA	The Expanded Very Large Array
FO	Fibre Optic
FORM	Fibre Optic Receiver Module
ID	Identifier
LTA	Long Term Accumulator (described in Baseline Board LTA FPGA RFS)
LB	Lag Block = 128 complex lags
MCCC	Master Correlator Control Computer
MHIT	Minimum Hardware Integration Time (defined in Correlator Chip RFS)
MJD	Modified Julian Day
MLID	Module Location ID
M&C System	Monitor & Control System
SB	Subband: output of a Station Board Filter.

Stb	Station Board
TCP/IP	Transmission Control Protocol / Internet Protocol
TEX	Tone Extraction
VCI	Virtual Correlator Interface
VLA	Very Large Array
VLBA	Very Long Baseline Array
VLBI	Very Long Baseline Interferometry
VSI	VLBI Standard Interface
XML	Extensible Markup Language

Definitions

Subarray	A collection of antennas that are part of an observation or experiment.
Quadrant	A group of Baseline Boards (two racks) that process data for a Baseband pair.
Julian Day (JD)	The Julian Day (JD) is defined as the number of days since noon UT on Jan 1, 4713 B.C. So 1/1/4713 BC is Julian Day 0.
Modified Julian Day (MJD)	The Modified Julian Day (MJD) numbering system is defined as MJD = JD - 2,400,000.5. The '.5' moved the start of the day back to midnight and the 2,400,000 made MJD 0 correspond to Nov 17, 1858. This allows specifying recent and near future dates using only 5 digits.
UT	Universal Time (UT) is a timescale based on the rotation of the Earth. It is a modern continuation of the Greenwich Mean Time (GMT), i.e., the mean solar time on the meridian of Greenwich, England, which is the conventional 0-meridian for geographic longitude. GMT is sometimes used, incorrectly, as a synonym for UTC. The old GMT has been split, in effect into UTC and UT1.
UT0	Universal Time 0 (Greenwich time without polar motion taken into account)
UT1	UT1 is computed by correcting UT0 for the effect of polar motion on the longitude of the observing site. UT1 is the same everywhere on Earth, and defines the true rotation angle of the Earth with respect to a fixed frame of reference. Since the rotational speed of the earth is not uniform, UT1 has an uncertainty of plus or minus 3 milliseconds per day.
UTC	Coordinated Universal Time (UTC) is a high-precision atomic time standard. UTC has uniform seconds defined by International Atomic Time (TAI), with leap seconds announced at irregular intervals to compensate for the earth's slowing rotation, and other discrepancies. The leap seconds allow UTC to closely track Universal Time (UT), which is a time standard based on the earth's angular rotation, rather than a uniform passage of seconds. Time zones around the world are expressed as positive or negative offsets from UTC.
VCI Client	A computer system on the Monitor & Control side of the VCI interface.

Revision History

Revision	Date	Changes/Notes	Author
Draft 1.0	October 6, 2003	First draft.	Sonja Vrcic
Draft 1.1	October 8, 2003	Updated after internal review.	Sonja Vrcic
Revision 2.0	November 19, 2003	Updated after review at s/w meeting in Socorro.	Sonja Vrcic
Revision 2.1	December 8. 2003	Changes due to architecture refinement: o Added Station Board parameter – threshold for delay model and tone phase model alarm generation o Added Board Configuration Request o Added "Initial Report Time" to the Status Request message. o Added parameter "overwrite sister configuration" to the Station Board Configuration Request. o Renamed Swap Configurations message to Rollback Configuration Request.	Sonja Vrcic
Revision 2.2	February 8. 2004	Changes to accommodate e-Merlin requirements: o Whenever possible identifier is defined as 32 bit integer in the range 0 to 2^31-2 (0xFFFFFFFE). Zero usually means "not defined" or "unknown". 2^31-1 (0xFFFFFFFF) means "invalid". o Added name string for all identifiers. Max. length of the name string is 32 characters.	Sonja Vrcic
Revision 2.3	October 28, 2004	Corrected integer identifier range which was, by mistake, defined in the range o to 2^32-2, instead of 2^31-2. Changed definition of the parameter Activation Time in Configuration Request messages, so that the notion of "act upon received messages trigger" is introduced. Rollback and Swapping parameters (Dwell A and Dwell B) moved to the Configuration Request header. Re-arranged some parameters in the Station Board Configuration Request.	Sonja Vrcic
Revision 2.4	November 02, 2004	Added parameter "Number of subband correlators" to the subband Configuration Request Added definition for hardware integration time, LTA integration time and backend integration time.	Sonja Vrcic

Revision	Date	Changes/Notes	Author
Revision 2.5	February 04, 2005	Updated after the review of the memo18: Added the parameter “polarization pair mate” to the BaseBand configuration.	Sonja Vrcic
Revision 2.6	March 18, 2005	Added parameters “Use mixer: yes/no” and “flip subband sideband: yes/no” to the Station Board subband Configuration Request. “Single phase center” and “fringe rotation” are specified per BaseBand, not per Station Board. Removed references to the Correlator Chip autocorrelation mode (not supported in the first release).	Sonja Vrcic
Revision 2.7	March 30, 2006	Updated Station Board Configuration Request. Message type IDs: replaced integer with XML tags. Replaced term “Alert” with Log.	Sonja Vrcic
Revision 2.8	14 Septemb er 2006	Updated TIMECODE related parameters to reflect h/w design changed (TIMECODE is now generated on the Station Board). Updated Station Board Configuration Request parameters to match the XML Schema defined for the message. XML tags for the message types start with a lower case character.	Sonja Vrcic
Revision 2.9	15 Novemb er 2006	Updated after face-to-face meeting in Socorro: Added two optional parameters to the VCI Configuration Activation Trigger: Mapping Time and Query Request. Format for the date and time in the VCI messages: • UT displayed in format defined by ISO 8601, or • MJD+fraction, displayed as IEEE double-precision. • Preferred format for real-time messages (models) is WIDAR internal format (two 32-bit integers). Delay Models are sent via MCCC (not directly to CMIBs). The parameter “model expiry time” or “valid interval” has been removed from the delay model and tone phase model. Instead of specifying the “model expiry time” in each model, user may specify how often delay models will be transmitted as a part of the baseband configuration message. This change has been introduced in order to reduce the length of the delay models. Removed all references to output file names; Station Board does not write output data to files.	Sonja Vrcic

Revision	Date	Changes/Notes	Author
Revision 3.0	5. December 2007	Major update due to the changes of the requirements and the new WIDAR connectivity scheme The new requirements as specified at the meeting with Bryan Butler and David Harland in May 2007 in Socorro: • EVLA Monitor & Control System (Executor) should be able to specify the list of antennas that belong to a subarray and list of basebands/subbands/products that apply to all the antennas in a subarray. This is a shift from the initial requirement to specify configuration for each Station Board individually. • The antenna to Station Board connectivity will be specified either as a part of a subarray configuration or in advance. The session establishment and maintenance messages have been removed from the document; it seems that EVLA designers do not see a need for a session oriented VCI protocol.	Sonja Vrcic
Revision 3.1	14. April 2008	Updated after detailed review.	Sonja Vrcic
Revision 3.2	30. April 2008	Changes and requirements after review with Michael Rupen and David Harland. This release is not ready for general review. Backend parameters and functionality in some other areas have not been defined.	Sonja Vrcic
Revision 3.3	25. June 2008	New diagram showing Station Board Input. Not marked with green marker: The titles of the tables that list content of XML elements have been updated to reflect position in the XML (element) hierarchy. Table titles are marked with green only if there is a change in the content of the table. The description of the Delay Models and Phase Models for Tone Extraction has been updated to improve readability and the parameters have been re-arranged to match the actual XML schema, there is no major changes in the content of those messages.	Sonja Vrcic
Revision 3.4	17. November 2008	Updated after review of the version 3.3 For easier reading, all previously used markers were reset.	Sonja Vrcic
Revision 3.5	18. November 2008	Updated the range for the recirculation factor. Consequently, recirculation factor has been updated in all the XML examples; those changes are not marked with yellow marker. Section 6.4.10.3, Baseline Board Pairs, the rules and examples have been updated.	Sonja Vrcic

Revision	Date	Changes/Notes	Author
Revision 3.6	21. September 2010	Document updated to reflect changes introduced during implementation and testing. Changes are marked with green marker. To remove clutter and save some ink, color marker indicating changes from previous releases was removed. Added: i. Attribute *re-configure entire Baseline Boards* ii. Spectral window related attributes needed for BDF iii. Switched power related attributes iv. List of stations to *stationInputOutput* v. Two algorithms for auto-correlation products vi. XML element mcMonitorControl Added more XML examples and replaced some of the examples from the previous version with real-life examples (i.e. XML generated by software). Some sectionms were updated to provide better and more precise definition and description of the protocol.	Sonja Vrcic
Revision 3.7	10 December, 2010	Description and examples for “modify subarray”. Configuration for: - Filter gain (baseband) - Signed sum of the local oscillators - SSLO (baseband) - Epoch for DumpTrig (subarray)	Sonja Vrcic
Revision 3.8	03 February, 2011	Correlator configuration for the summed array – XML element summedArray.	Sonja Vrcic
Revision 3.8	16. February, 2011	Minor updates for Summed Array.	Sonja Vrcic
Revision 3.9	April 12, 2011	Replace subarrayId with configId (required to match attributes used bu other subsystems). Until proven to work, CM software will support both attributes: if configId is specified, it will use configId. If configId is not specified it will look for subarrayId and use that attribute. Note: All XML examples that contain element subarray have been updated to replace subarrayId with configId. For simplicity change is not marked with yellow marker. Element subband: replaced element autoCorrMode with attribute autoCorrMode (this simplifies the schema).	Sonja Vrcic

Revision	Date	Changes/Notes	Author
Revision 3.10	September 29, 2011	Updated information related to Logs and Alerts. Removed all references to 'alarm' which was not EVLA terminology. Removed log priority levels higher than 'ERROR' (not used by CM and not used in EVLA system (to my knowledge)). Added a new CM Monitor & Control command 'delete sybarray' – this command deletes all instances of a subarray with the specified config ID, active and/or pending. Updated info related to 7-bit correlation. No changes in VCI protocol. Implemented algorithm for 7-bit Baseline Board configuration is simpler that initially proposed (and very similart to 4-bit configuration).	Sonja Vrcic
Revision 3.11	Oktobar, 2011	Modified format for WBC products.	Sonja Vrcic
Revision 3.12	January, 2012	Station packing algorithm 'minPack' modified to work for up to 32 stations. No changes to the XML schema, modified only the actual configuration of the Baseline Boards . Correction: the upper limit for the minimum hardware integration time is 500 micro seconds.	Sonja Vrcic
Revision 3.13	August 2012	In this version of the document many chapters were modified with intention to improve and expand explanations, only sections where functionality changed or was added are marked with yellow marker. Noise diode parameters are not set via VCI and have been removed from the VCI schema. Elements and attributes related to STB data products and monitoring have been removed from this document and XML schema. It has been agreed upon that Station Boards data products (input counts, switched power, etc.) are automatically configured and turned ON when a subarray is created. STB data products are collected for every subarray and are available on demand. There is no need to explicitly set parameters related to STB data products except for RFI detection and excision. Ability to pass Station Board and/or Baseline Board configuration through VCI (i.e. via Configuration Mapper) was never implemented, it is not required and the relevant sections have been removed from the document. Added chapter for cmStatus (see CM Status). Interested readers shoud re-read all the chapters related to Baseline Board configuration for polarization products. Information for stationPacking=fourPerRowColumn (maxPack) has been expanded. Previously implemented functionality remains the same. There are significant changes related to stationPacking=onePerRowColumn (minPack). Algorithm to	Sonja Vrcic

Revision	Date	Changes/Notes	Author
		calculate the number of Baseline Boards has been updated. Added attribute recPhase (phase for recirculation) to element subband/polProducts/productIntegration. VDIF – attributes frameSize and frameDelay were updated. Still need to update: • List of referencies • XML examples	
Revision 3.14	April 10, 2013	Updated description for inter-frame delay attributes: frameSchedulingAlgorithm, interFrameDelay and randomDelay. Wideband Correlator (WBC) related changes: • WBC products are now by default disabled. - Added optional attribute noWbcProducts to element *baseband*. New attribute can be used to disable all WBC products. - Added optional attribute status to element *wpp*. New attribute can be used to enable/disable individual WBC products. SummedArray related changes: Removed default values for most optional attributes from the XMLschema. Reason: When default value for optional attribute is specified in XML schema, JAXB generated code automatically adds optional attributes to the generated XML element This is OK for 'create subarray'. Message 'modify subarray' should contain only attributes that should be modified, not the full set of attributes. For the same reason some attributes that were required are now declared as optional. CM implements the following: Create subarray: if optional attribute is not specified in the received VCI message, CM sets the default value Modify subarray: if optional attribute is not specified in the received VCI message, the correlator configuration for that attribute does not change. Removed chapter 'timers' from the document; in the EVLA system originator of the VCI rquests does not listen to VCI responses and therefore does not implement proposed timers. Re-organized material related to WBC products and regular polarization products. Number of other minor changes (typos, format, etc.). Chapers and tables where functionality has been modified or expanded are marked with green marker.	Sonja Vrcic

Revision	Date	Changes/Notes	Author
Revision 3.15	July 09, 2013	Revision 3.14 was never published; the same marker is used to indicate changes for 3.14 and 3.15. The following functionality has been added in this release: 1. stationPacking 'twoPerRowColumn' (midPack) 2. burst mode 3. inter frame delay mode = minDelay 4. RFI related parameters A number of sections has been updated to fix typos and other mistakes, and/or to improve description. Green marker is used to indicate changes of functionality, not editorial changes.	Sonja Vrcic
Revision 3.15	July 12. 2013	Improved description and added diagram for stationPacking=fourPerRow (maxPack) productPacking = minPack See Figure 6-12 and Figure 6-13 on page 121.	Sonja Vrcic
Revsioin 3.16	Oct. 21, 2013	Added phase binning parameters.	Sonja Vrcic

1 Introduction

This document defines the protocol specification for the Virtual Correlator Interface (VCI) for the WIDAR Correlator.

This version of the document describes Virtual Correlator Interface:

1. **As defined in XML Schema VCI version 3.16, and**
2. **As implemented by VCI Configuration Mapper (CM) in October 2013.**

The Virtual Correlator Interface (VCI) is the interface between the correlator and the Monitor & Control System. The VCI protocol is the set of procedures and messages that allow the Monitor & Control System to configure, control and monitor the correlator. This specification defines format and content of the messages that flow over the VCI and expected behaviour of the systems that communicate over the VCI.

VCI is a machine-to-machine interface; it is assumed that Monitor & Control System implements additional layer of software that translates VCI messages into a user friendly form. In further text, the Monitor & Control System will be referred to as the EVLA Monitor & Control System (EMCS).

Expanded VLA Project (EVLA) is described in the EVLA Project Book [1]. Chapter 8 of the EVLA Project Book describes the correlator. EVLA Correlator User Manual [17] describes the correlator in more detail.

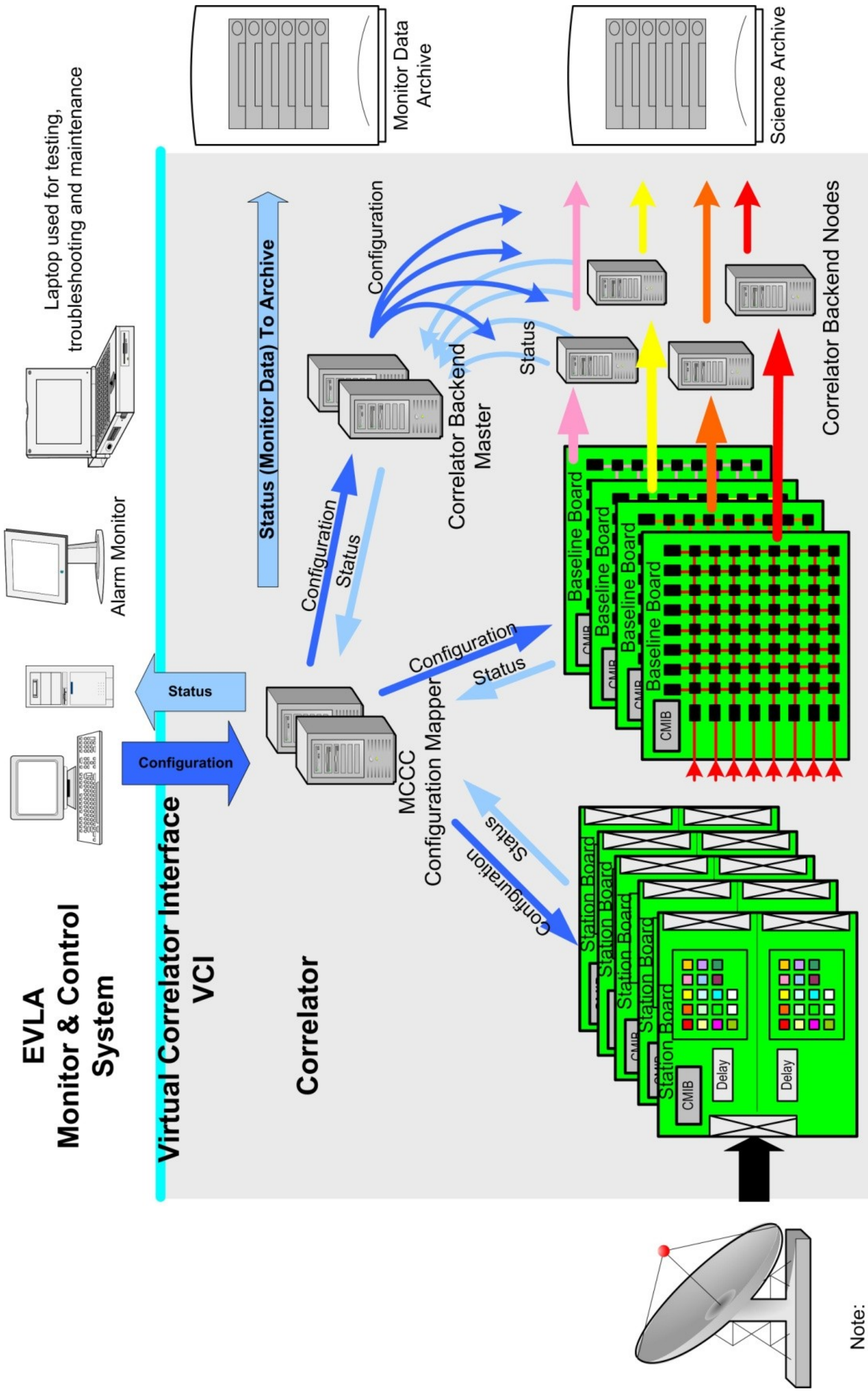
Figure 1-1 is a conceptual diagram; it shows high-level architecture of the correlator and flow of the monitor and control data. As shown in the diagram, the Master Correlator Control Computer (MCCC) handles VCI messages:

- MCCC receives configuration and control data specified by the EMCS and translates it into a lower level implementation-oriented format used for communication among the correlator components.
- MCCC monitors status of all the correlator components and maintains overall system status, which is reported over the VCI.

Note: The bulk of monitor data will be sent directly to the archive by the individual correlator subsystems.

This document does not define which part of the EVLA Monitor & Control System generates VCI messages. VCI protocol is designed to allow more than one source of VCI messages

Figure 1-1 Virtual Correlator Interface



Note:

- The full EVLA configuration consists of 128 Station Boards and 128 Baseline Boards.
- MCCC and Correlator Backend Master may be running on the same computer system. In any case, 1+1 redundancy will be provided for both.
- Number of the Correlator Backend Nodes is to be determined.
- The diagram shows configuration where each subband is sent to a different CBE node; that does not need to be the case, a CBE node will be able to receive and process output products for more than one subband.

2 Overview

2.1 VCI Peers

VCI is the interface between the EMCS and correlator. The EMCS and correlator are VCI peers.

Flow of information over the VCI interface is asymmetric:

- EMCS is the originator of requests.
- The correlator, or more precisely the MCCC, receives and processes requests, and generates responses, status reports, logs and alerts.

This document is a protocol specification and not an architecture or design document; this document does not define which part of the EMCS generates a particular message or implements a particular functionality. The same can be said about the correlator. However, on the explicit request from the readers, the document has been updated to specify which correlator component implements a particular behaviour. Within the EVLA correlator, VCI protocol is implemented as a process running on the MCCC. More precisely, VCI interface is implemented by VCI Configuration Mapper (CM).

An effort has been made to explicitly specify the hardware or software component that performs a particular action, however when the document refers simply to ‘the correlator’ (as in the: correlator rejects configuration), the functionality is implemented in the CM.

2.2 Types of Messages

Messages exchanged between the VCI peers may be classified as follows:

1. Configuration request, used by the EMCS to change the correlator configuration.
2. Configuration query, used by the EMCS to check whether a particular configuration can be activated at the specified time. Configuration query does not result in the configuration change.
3. Real-Time Control messages, used by the EMCS to specify delay and phase models for on-going observations.
4. Log / Alert messages, used by the correlator to report significant events.
5. Status Report messages, used by the correlator to report the correlator status.

2.3 Format

VCI messages are encoded as XML elements. An XML Schema is defined for each message type. XML Schema is a part of the protocol specification.

At the time when this revision is published software that implements VCI protocol is still under development. As a part of development process XML Schema for the VCI messages is being modified and improved.

The latest version of the XML Schema for the VCI messages can be found in the NRAO software repository in the directory:

trunk/widar/schema/vci

As a convention, when referring to an XML element using its tag, this document uses *italics*.

2.4 Error Handling

Both the EMCS and the correlator CM perform syntax and semantic checks for all received VCI messages.

If CM receives an unexpected, malformed or erroneous VCI message, CM raises an alert and generates a log. CM also generates either a negative acknowledgement or reject message. At this time EMCS does not listen to nor process VCI responses generated by the CM; those messages are transmitted to the well-known group address[1] and can be displayed using CM GUI (a.k.a. VCI Client) which can be launched from the WIDAR GUI web-page.

If the EMCS detects an error in a message received from the CM (or the correlator in general), it should notify the operator. There is no point in notifying CM, since all the messages generated by the correlator are either responses or reports.

A VCI response and/or log generated as a direct consequence of a received VCI message, contains the original VCI message. If it is impossible or impractical to include the whole message, CM includes the header of the received message.

[1] See section 13.7 on page 201.

2.5 Extensibility and Future Compatibility

Additional functionality may be added to the VCI in the future.

The version of the protocol is defined in the schema and is specified in each VCI message. This allows VCI peers to easily identify the system that uses unsupported or unknown version.

Whenever possible, changes in the VCI protocol should maintain interoperability with previously implemented software.

2.6 Communication infrastructure

The physical connection between the EMCS and correlator is 1Gb/sec or 100Mb/sec Ethernet.

Correct operation of VCI requires reliable and ordered delivery of messages. That may be achieved either by using a reliable transport infrastructure or by defining and implementing the VCI protocol so that all the messages are acknowledged and retransmitted if the acknowledgement is not received in due time.

In the EVLA system VCI messages are transmitted using HTTP (Hypertext Transfer Protocol)[1].

For each request received over VCI , CM generates a positive or negative acknowledgement. An acknowledgement message informs the originator that the request has been received, not only by the destination device, but also by the process which is responsible for handling VCI requests[2]. A negative acknowledgement contains a log record generated during the message parsing and processing.

---

[1] HTTP is well known and widely used protocol specified by IETF RFC2616. The protocol specification can be found at: http://www.ietf.org/

[2] VCI Configuration Mapper.

3 Status and Priorities

Detailed requirements and functionality still need to be defined in some areas. Sections that describe functionality that has not been implemented were not updated. Detailed requirements and list of parameters have still to be defined for the following:

1. Fast switching between two configurations can be used to obtain auto-correlation products for all the antennas, using time-multiplexing. Plan is to set Baseline Boards so that at any time Baseline Board generates auto-correlation products for a half of the stations. This feature can be implemented so that:

    a. CM periodically sends to BaselineBaord CMIB 'delta' configuration for the Correlator Chips that produce auto-correlation products. Functionalty would be implemented by CM, CMIB software would be completely unaware of the 'switching between two configurations' as a feature.

    b. Baseline Board configuration transmitted by CM could specify two alternative configurations for the Correlator Chips that produce auto-correlation products and an indication how often CMIB should switch between those two configurations. This would require addition to the Baseline Baord XML schema and some work on CMIB and CM software.

    Option b) is preffered. Option a) would increase number and frequency of Baseline Board re-configurations. Also when 'delete subarray' is received, CM would have to revoke from the CMIB queue all 'delta' configurations; all this is duable but could destabilize the system.

2. Gating (for pulsars). Gating was assigned a low priority and has not been implemented (CMIB and VCI parameters to be defined).

3. Tone extraction. Tone extraction was assigned a low priority and has not been implemented (CMIB and VCI parameters to be defined)

4. Radar Mode (saving row data for one subband per Baseband Pair). Radar mode was assigned a low priority and has not been implemented (CMIB and VCI parameters to be defined).

5. Disable/enable correlation (output products) for individual stations and/or baselines. Requirements are known but this has been assigned very low priority.

Time needed to re-configure the correlator is less than 6 seconds which meats the requirements.

4 Operation

This section defines the procedures used by the VCI peers. The details of the protocol are defined in Section 4.1.13.

4.1 Configuration

Configuration is the core functionality of the VCI. Naturally, the flow of information is asymmetrical: the VCI Client generates Configuration Requests, and the correlator generates acknowledgements and responses.

4.1.1 Commands

Some of the parameters are, in fact, commands, such as:

- Clear the queue of received VCI messages,
- Clear activation queue (discard configurations that have not been activated so far),
- Remove all subarrays (release correlator resources),
- Remove all stations (release Station Board resources).

4.1.2 Configuration of a Scan or Experiment

Configuration of a scan or experiment consists of:

- **Station** configuration, which assigns a Station ID to a group of Station Boards that receive input from the same antenna (or other source). Station is defined as a list of Basebands. For each Baseband, the message has to specify: Baseband ID, source of input data, Station Board Data Path and polarization.
- **Subarray** configuration, which consists of the list of stations, description of input data (basebands) and desired output products (subbands, polarization products). Subarray configuration also specifies Baseline Boards to be used.
- **Activation Trigger** that specifies activation time for all configuration messages with the same Activation ID.

4.1.3 Subarray Configuration

Subarray configuration is specified as a list of stations (antennas) that belong to the subarray and specification of the basebands, subbands and output products. Typically, all the stations in a subarray are configured in the same way. However, subarray may consist of antennas (stations) that are not identical. Antennas (stations) that belong to the same subarray may have different baseband configuration.

By default, all the baselines in a subarray are enabled, i.e. at the activation time the correlator begins to generate output products for all the baselines.

In a subarray where baseband/subband configuration is not identical for all the stations, some baselines are permanently disabled; for example: subbands that have different bandwidth cannot be correlated.

A subarray is identified by the Config ID (ASCII string).

A subarray cannot be replaced by another subarray with the same Config ID. Previously configured subarray has to be removed (deleted), before a new subarray with the same Config ID can be created.

CM does not allow user to add or remove stations to/from active subarray. Also, CM does not allow user to add basebands and subbands to the stations that belong to an active subarray. Only a subset of subarray parameters can be modified.

VCI Configuration Mapper has no concept of the duration of a scan; the correlator keeps processing input data and generating output products as long as configuration remains active.

A subarray which is no longer needed must be de-activated explicitly, by sending "delete subarray".

4.1.4 Configuration Request Type

VCI protocol defines the following Configuration Request types:

1. Element ***stationHw*** assigns Station ID to Station Boards that receive input from the same antenna. Baseband ID is assigned to a Station Board Data Path. Station ID must be unique within the system (the correlator) and Baseband ID must be unique for a station.

2. Element ***subarray*** consist of:

    - List of stations that belong to the subarray,
    - Specification of input data (basebands) and
    - Specification of the desired output products.

Content and format for VCI Configuration Request is defined in the Chapter 6.

4.1.5 Acknowledgement

When a VCI Request is received, CM performs basic syntactic and semantic checks. Using XML terminology, CM checks whether received message is well-formed and valid.

If an error is detected in the received VCI Request, CM:

- Raises an alert.
- Generates a log.
- Creates and sends back to the originator a Negative Acknowledgement (Nack), which contains a copy of the received message and reason for rejection.
- Discards received VCI Request.

A VCI Request can contain configuration for more than one station. If an error is detected in one message, CM will reject a complete vciRequest.

If the message has been successfully parsed and validated, and no errors have been detected during initial checking, CM creates and sends back to the originator an Acknowledgement (Ack). This message is an acknowledgement that the request has been received and that it passed initial tests, i.e. the XML content is well-formed and valid. Acknowledged messages are added to the Configuration Queue where they stay until configuration mapping is performed.

4.1.6 Mapping Time

Configuration Mapping Time is the time when the messages with the same Activation ID are removed from the Configuration Queue, processed and translated (or mapped) into the correlator configuration.

Mapping Time can be specified in the Activation Trigger.

If the Mapping Time is not specified, mapping is performed immediately upon receipt of the Activation Trigger.

The ability to explicitly specify Mapping Time has been introduced to ensure (or at least increase probability) that all configuration messages that belong to the same scan or experiment are received before the mapping is performed. For example: Mapping Time may be used to postpone mapping in the case when several devices in the EMCS originate messages that belong to the same scan.

4.1.7 Mapping Order

At Mapping Time, messages with the same Activation ID are removed from the Configuration Queue and the "mapping" is performed.

If the Configuration Queue contains more than one message with the same Activation ID, messages are processed (mapped) in following order:

1. Messages that have parameter "mapping order" assigned to them are processed first, starting from the message with the lowest mapping order[1]. If there are several messages with the same mapping order, they are processed in the order in which they were received.

2. Messages that have no mapping order assigned to them are processed after all the messages that have mapping order assigned to them. Messages that have no "order" assigned to them are processed in the order in which they were received (and placed in the configuration queue).

The algorithm implemented by the Configuration Mapper is determinisitic, i.e. mapping of the same VCI message (or set of messages) always produces the same hardware configuration. The only exception is allocation of the rows/columns on the Baseline Board, where result may be different if some of the rows/columns are alredy used for another subarray.

4.1.8 Activation Time

Activation Time is specified in the Activation Trigger. VCI Client should always specify time in the future, so that CM has enough time to perform semantic and syntactic checks and distribute configuration to the CMIBs and Backend. If there is not enough time to configure all the components, activation is delayed.

Time needed to re-configure the correlator will be empirically determined and results will be added to this document.

If the specified Activation Time is in the future, CM accepts the Activation Trigger, no matter how soon, or how distant the Activation Time is.

If CM receives Activation Trigger with an expired Activation Time, configuration will be activated as soon as possible.

Activation Time is an optional attribute, if not specified, it is interpreted as: activate configuration as soon as possible.

[1] Message order is a specified as positive integer.

For every subarray, CM generates a log which indicates both the Activation Time specified in the Activation Trigger and the actual activation time.

For the active configuration the Activation Time as specified in the Activation Trigger and the actual activation time are displayed on the CM web page.

4.1.9 Observation Time

Observation time may be specified as a part of a subarray configuration.

If not specified, it is assumed that observation time is the same as system time.

The system time is the time of the operating system on the computers where the CM and CMIB software are running.

4.1.10 Acceptance

At Mapping Time, messages with the same Activation ID are removed from the Configuration Queue and analysed to check if the correlator can be configured as specified.

If an error or inconsistency is detected, or if there are no enough resources to obtain specified products, CM:

- Raises an alert.
- Generates a log.
- Creates and transmits message VCI Reject.
- Discards all configuration messages for the specified activation time.

After careful consideration it has been decided that the initially planned parameter "reject level" should not be implemented. It is not a requirement for CM to accept partial configurations. If CM can not accept configuration as specified in the request, configuration is rejected.

If the correlator can be configured as specified, new configuration is placed in the Activation Queue and VCI Accept message is transmitted. Message VCI Accept contains Activation ID and Configuration ID; it may also contain logs generated during the "mapping" of the configuration requests.

Unlike the Ack and Nack messages that are sent to the originator, Accept and Reject messages are transmitted as multicast messages to the well-known multicast group[1].

---

[1] See section 13.7 on page 201.

4.1.11 Activation

At Activation Time, CM removes the configuration from the activation queue; it becomes 'the active configuration'. CM then transmits queries to check the status of the affected boards and generates Activation Report. In the same fashion as Accept and Reject, Activation Report is transmitted as a multicast message.

4.1.12 Activation Failure

If any of the correlator subsystems fails to activate the configuration, the correlator generates alert and log. Alert message identifies the component that failed to activate configuration.

The sub-system that failed to activate new configuration in most cases still has the previous configuration. Overall, the correlator configuration can be considered as inconsistent. This situation may be amended only when a new configuration is activated; or when the affected component returns to a known state.

In some cases activation failure should be easy to amend, but it may require human intervention.

4.1.13 Subarray Configuration – An Example

This section provides an example of a VCI Request message for the subarray that consists of three antennas (stations).

The **vciRequest** in XML Example 4-1includes:

- Station hardware configuration, which assigns Station IDs and Baseband IDs to the Station Board hardware.
- Subarray configuration has only one instance of the element *stationInputOutput*, i.e. all the stations in the subarray are configured in the same way. Element *stationInputOutput* contains:

    1. List of stations that belong to a subarray.
    2. Baseband parameters (*bbParams*) that specify the source of input data and polarization.
    3. List of baseband pairs, with the list of subbands for each baseband and list of polarization products for each subband.

The correlator configured using vciRequest as specified in XML Example 4-1 would produce the following output products:

a) Cross-correlation products: polarization products: R*R, R*L, L*R, L*L; 64 spectral channels per product for all baselines. In this example there are 3 baselines: 1*2, 1*3 and 2*3.

In general, number of baselines can be calculated as follows:

numBaselines = (numAntennas * (numAntennas-1)) / 2

b) Auto-correlation products: Auto-correlation algorithm is defined as: “Get all auto-correlation products for every other station, starting from the station with the lowest ID.”

In this example, the correlator would produce products A*A, A*B and B*B for stations 1 and 3. In the case of ‘single dish products’ i.e. when correlating antenna with itself, products A*B and B*A are identical, only one is required (and produced).

XML Example 4-1 vciRequest: station hardware, subarray and activation trigger, 8-bit IQ


```
<widar:vciRequest desc="StationHwAndSubarray" timeStamp="2009-09-01T11:16:12.000-07:00"
                  msgId="97" xmlns:widar="http://www.nrc.ca/namespaces/widar">

  <widar:stationHw sid="1" name="Station1" activationId="example1" action="add" msgId="6">
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-0" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-0" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-4" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-4" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-0" dataPath="0"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-0" dataPath="1"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-4" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-4" dataPath="1"/>
    <widar:antenna type="EVLA" id="ea01" name="Antenna1"/>
  </widar:stationHw>

  <widar:stationHw sid="2" name="Station2" activationId="example1" action="add" msgId="7">
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-1" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-1" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-5" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-5" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-1" dataPath="0"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-1" dataPath="1"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-5" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-5" dataPath="1"/>
    <widar:antenna type="EVLA" id="ea02" name="Antenna2"/>
  </widar:stationHw>
  <widar:stationHw sid="3" name="Station3" activationId="example1" action="add" msgId="8">
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-2" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-2" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-6" dataPath="0"/>
```

```
        <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-6" dataPath="1"/>
        <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-2" dataPath="0"/>
        <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-2" dataPath="1"/>
        <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-6" dataPath="0"/>
        <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-6" dataPath="1"/>
        <widar:antenna type="EVLA" id="ea03" name="Antenna3"/>
    </widar:stationHw>

    <widar:subArray configId="vciDoc-ExampleOne" activationId="example1"  action="create" msgId="9">

      <widar:stationInputOutput sid="list">

        <widar:station sid="1"/>
        <widar:station sid="2"/>
        <widar:station sid="3"/>

        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="1"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="2"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="3"/>

        <widar:baseBand inQuant="8" bw="1024000000"  swbbName="AC_8BIT" bbB="2" bbA="0">

            <widar:subBand sbid="1" swIndex="1" centralFreq="64000000" bw="128000000"
                             rqNumBits="4">

                    <widar:polProducts>
                        <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                        <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                        <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                        <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
                        <widar:blbProdIntegration recirculation="1" minIntegTime="250"
                                    ltaIntegFactor="4000" ccIntegFactor="1" cbeIntegFactor="1"/>
                        <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="1"/>
                        <widar:stationPacking  algorithm="maxPack"/>
                        <widar:productPacking algorithm="maxPack"/>
                        <widar:autoCorrSubset algorithm="halfStationsMaxProd"/>
                    </widar:polProducts>
                </widar:subBand>

            </widar:baseBand>
      </widar:stationInputOutput>
    </widar:subArray>

    <widar:activationTrigger msgId="2" activationId="example1"  activationTime="2010-06-11T16:56:00"/>

</widar:vciRequest>
```

4.2 VCI Query

A VCI configuration can be transformed into a query by adding a single attribute.

If the Activation Trigger contains '*query=yes*', VCI messages are processed as query; which means that CM will perform the mapping and generate report (Accept or Reject), but will not change the correlator configuration. In other words, Configuration Mapper will check if configuration can be activated at the specified time, but will not activate the configuration (i.e. will not forward configuration to other components).

A VCI query has been introduced to provide a means for VCI users to check in advance whether a set of VCI messages is valid and whether it can be activated at the specified time.

4.3 Models

Messages that contain delay models are generated by the EMCS in real-time, i.e. during the scan or experiment.

To reduce overhead, messages that contain Delay Models may contain an arbitrary number of models.

To minimize copying and reformatting within the correlator, it is recommended that a single message should contain Delay Models for a pair of Basebands and TEX Phase Models for all the sub-bands on a single Station Board.

In the EVLA system Delay Models are send directly to CMIBs.

Tone Extraction has not been implemented, therefore phase models for tone extraction (TEX) have not been implemented either.

4.3.1 Delay Models

WIDAR correlator uses two types of delay models:

1. **Baseband Delay Model** used for wideband delay tracking. Usually, the same Delay Model applies for both Basebands in a polarized pair; but nothing prevents user to specify a different Delay Model for each Baseband. If each of the Basebands that belong to a pair is processed on a different Station Board[1], both Station Boards should receive delay models for both basebands.

2. **Subband Delay Model**s can be used to offset the delay center on the sky from the main delay center on a subband bases, or to phase up the array (e.g. for VLBI).

---

[1] This is always the case for 8-bit initial quantization.

Subband delay tracking is implemented in the Station Board Filter and can be turned for each subband independently. A differential Subband Delay Model may be defined for each subband of each baseband. Total of 36 Subband Delay Models may be specified per Station Board, one model for each Station Board Filter.[1]

All delay models for one Station Board can (and should) be transmitted as one message.

In general, each model is expected to run for 10 seconds and thus 10 seconds of delays are calculated by CMIB software and fed into a buffer. If this buffer expires before a new buffer is submitted, CMIB software generates a log (this is not necessarily an error condition).

VCI Delay Model is considered valid until replaced by another model. Until a new model arrives (and its activation time passes), the previous model is continuously evaluated and used. If a new model is not received for more than 5 minutes, the CMIB model task will quit evaluating and turn to idle state. For detailed description refer to documents that describe Station Board CMIB software and Delay Model Handler.

4.3.2 Phase Model for Tone Extraction

Station Board hardware has the ability to extract one tone (frequency) per subband.

VCI interface for tone extraction is not implemented.

Proposed functionality for tone extraction is described in section 6.4.17. Content of the Phase Model for Tone Extraction is described in section 9.2.

4.4 Event Reporting

CM and other correlator components report significant events using EVLA alerts and logs. Logs are used for debugging and toubleshooting. Alerts are used to inform operator and/or user that a problem occurred.

WIDAR alerts are handled by Alert Handler running on the MCCC. Alert Handler generates alerts in the EVLA alert format and sends alerts to the well-known multicast address so that they can be displayed on the operator screen. For details regarding EVLA alerts refer to documents describing EVLA Monitor & Control System.

CM and CMIB software generate logs that are stored in log files. CM and CMIB log files are implemented as circular lists, so that old logs eventually get overwritten by new logs.

VCI Configuration Mapper implements the following functionality:

- Logging level is configurable parameter (see Table 6-38 *cmMonitorControl*).

[1] Station Board has two banks of 18 filters; all the filters in the same bank get the same input data i.e. the same Baseband. A filter output data stream is called subband.

- Log Record is stored in the log file only if its priority is the same as, or higher than, the current CM logging level.
- If Log Record priority level is ERROR CM also generates an alert.
- Logging of received and transmitted messages can be turned on/off[1].

Detailed information regarding CM logging is available in CM User Manual [18].

4.5 Status Reporting

Except for the alerts that are transmitted by all sub-systems, status reporting in the EVLA sustem is 'pull' based. EVLA implements REST architecture, where each device maintains its own status and is able to report status on demand.

CM status query is a part of VCI interface. On request, CM can report:

- status of CM configurable parameters,
- content of the queues and
- status of active (current) correlator configuration.

CM status report does not include operational status for all the boards; CM reports status only for the boards affected by the last configuration change.

Operation status of the correlator hardware can be obtained from the CMIBHOST at

http://cmibhost.aoc.nrao.edu/

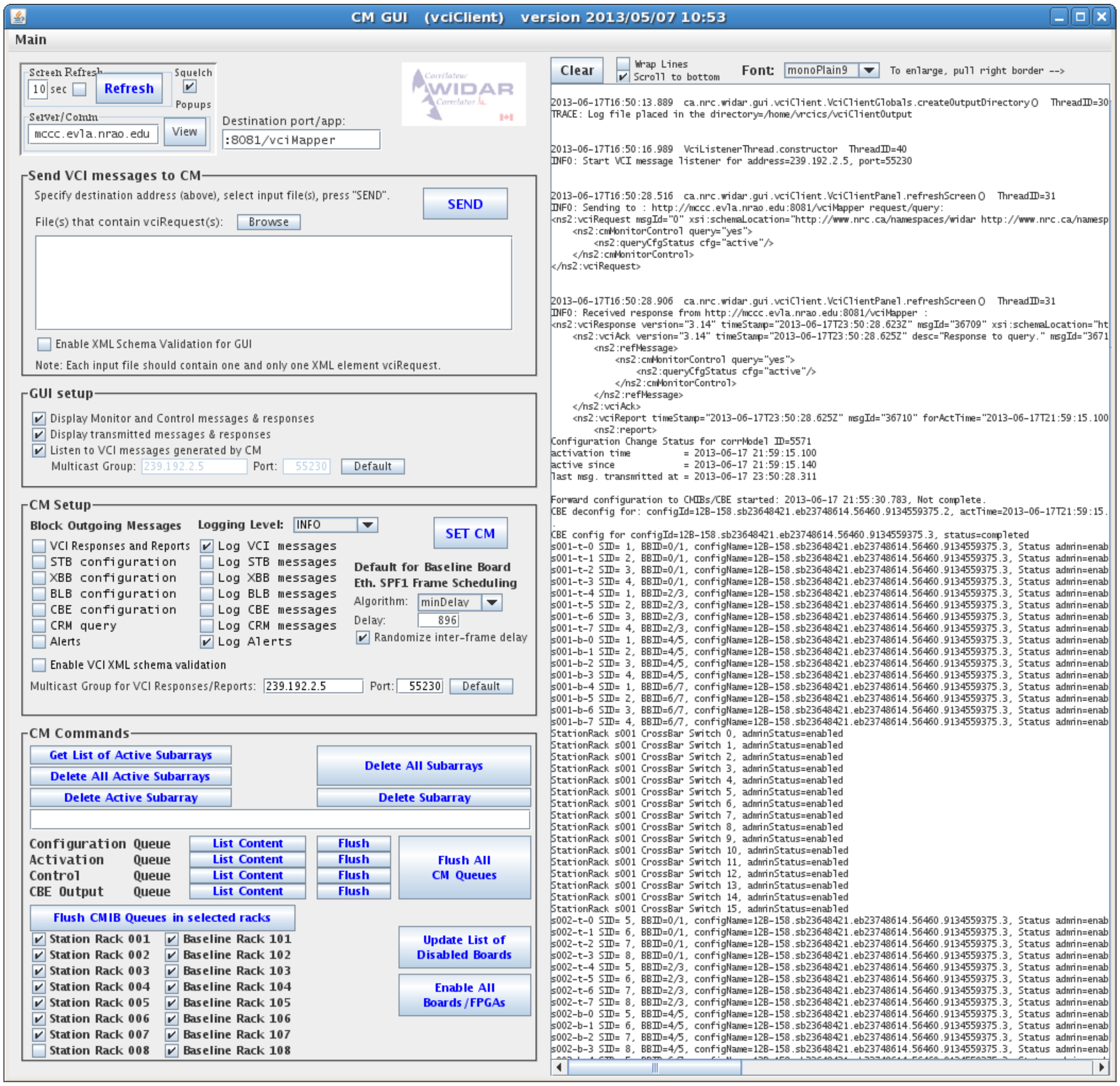
or from individual boards using various GUI-based applications.

Status of CM configurable parameters can be obtained using VCI CM query or via HTML interface.

GUI-based tool vciClient.jar (known as CM GUI) can be used to display and change CM setup.

[1] Logging of received/transmitted messages can be enabled/disabled via CM GUI.

Figure 4-1 CM GUI



5 Protocol

The section on VCI operation describes scenarios that involve exchange of messages among VCI peers. This section specifies message content and procedures for processing the messages.

5.1 VCI Procedures

VCI defines messages, parameters and procedures in the following areas:

1. Configuration,
2. Real-time control,
3. Event reporting, and
4. Status reporting.

The sections that follow describe messages and procedures.

5.2 Encoding

VCI uses XML to encode the information carried in the VCI messages.

VCI Protocol Data Unit is a well-formed XML document.

See XML Example 4-1 on page 34 for an example of VCI message.

XML Schemata, which is the blueprint or template for XML messages, is defined for all VCI Protocol Data Units. XML Schemata defines XML tags (names of XML elements) and content (children and attributes of the elements). For attributes that can take only a predefined set of values, XML schema defines the range.

XML schemata are not included in this document; they are stored in the NRAO software repository, in the directory:

trunk/widar/schema/vci

A copy of VCI XML schemata is available on the NRAO WIDAR web-server:

http://www.aoc.nrao.edu/asg/widar/schemata

When URL of the XML schema is specified in VCI message, general purpose XML parser can verify not only whether a received VCI message is well-formed, but also whether all the required parameters are specified, and whether the value of the specified required and optional parameters is within supported range (in XML terminology: it performs validation).

Additional XML schema file defines content and format of messages used to get CM status in HTML format (to be displayed in a general purpose web browser).

trunk/widar/schema/cmStatus/cmStatus.xsd

XML element cmStatus can be used to get the list of active subarrays with the list of stations/antennas that belong to each subarray. This is used to provide information to MCAF.

5.3 Units

Whenever possible, attribute values in VCI messages are specified in basic units, e.g.: Hertz, seconds, etc.

5.4 Date and Time

Date and time in VCI messages may be specified in one of the following formats:

1. UT (Universal Time) displayed as ISO 8601 (format defined for dateTime in the XML Schema Part 2: Datatypes Second Edition [9]),

2. MJD + fraction displayed as IEEE double-precision floating point number (format defined for double in the XML Schema Part 2: Datatypes Second Edition [9]). Scientific Data Model (SDM) stores time as MJD in nanoseconds (i.e. as integer value).

ISO 8601 should be used whenever possible, that's an international standard endorsed by XML Schema and is human readable.

5.5 Conventions

This chapter describes conventions used in this document.

Content of VCI messages, more precisely XML elements and attributes, are described using tables. Each table describes single XML element.

Each table has the following columns:

- Parameter – one of the following: XML element (tag) or XML attribute (name).
- R/O – indicates wheter attribute or element is required or optional.
- Range – for attributes defines type and range.
- Descrption – texual description of parameters (attributes).

Note: In release 3.14, term ‘required’ replaced previously used term ‘mandatory’. XML schema uses term ‘required’ to convey that parameter must be present (i.e. must be specified). Since VCI interface uses XML encoded messages and XML schema is used to formally define content and format of VCI messages it is appropriate to use XML terminology.

In addition, some tables in this document have:

1. column ‘modify’ to identify attributes and elements that can be modified using message ‘modify subarray’, and/or
2. column ‘must be same for all stations’ to identify attributes and elements that must be the same for all stations in a subarray (otherwise correlation can not be performed).

Rows that contain XML elements (as opposed to attributes) are shaded gray. Each element is described in more detail in a different table (reference is provided in the column description).

When referring to an XML element using its tag, this document uses *italics*.

Location of files in NRAO software repository is described starting with:

trunk/widar/

When checking out code and other files from NRAO software repository, checkout command will create directory ***widar*** in the current user directory. This document *does not* include instructions on how to check out code from NRAO software directory.

5.6 Messages

Communication over VCI is asymmetric:

- EVLA Management & Control System (EMCS) originates requests, and
- VCI Configuration Mapper (CM) originates responses.

So called, “envelope” elements are used to group several VCI messages and transmit them as one. VCI defines two types of envelope elements:

1. VCI Request
2. VCI Response.

VCI Request is generated by the users (EMCS, VCI client). It can contain:

- configuration requests,
- configuration queries,
- CM setup (configuration),
- CM status query.

Table 6-1 lists the parameters of the elements vciRequest and vciResponse.

For each parameter of each VCI message, this document specifies the following:

1. Parameter name,
2. Range,
3. Short description, and
4. Usage Indication: required (R) or optional (O).

Required parameters must be specified. If one of the required parameters is not specified, received message is rejected by CM. When rejecting a message CM: raises an alert, generates log, generates and sends back to the originator a Negative Acknowledgement (Nack).

Optional parameters are specified when the originator of the VCI Request chooses so. This document and, in many cases, XML schema, define the default value for optional parameters. Default value is the value that is used by CM if the value is not explicitly specified in the VCI Request.

Some of the parameters that are marked as optional may become required when a particular output product or functionality is requested. It is not possible to specify such dependency in XML schema; this document should be used as a companion to the XML Schema.

Table 5-1 VCI Message Header

Parameter	R/O[1]	Range	Description
Message Type	R	vciRequest vciResponse vciAck vciNack vciAccept vciReject vciLog vciReport	XML tag.
Message ID	R	1..2^16-1	XML attribute. 16-bit positive integer.
Time Stamp	R	UT	XML attribute. Time when the message was transmitted (or created) by the originator. UT displayed in format defined by ISO 8601. At the review of this document, there has been some discussion on whether UT and ISO8601 define range or format. In the rest of this document UT is specified as range, while the format (ISO 8601) has been moved to the description.
Version	O	3.9 3.10 3.11 3.12 3.13 3.14 3.15 3.16	VCI protocol version. **Corresponds to the version (revision) of this document.** Protocol version is an optional parameter, if specified and XML schema validation is on, version must be one of the values specified here.
Descriptor	O	ASCII String	ASCII string that may be used to provide additional "human readable" information. Not used by CM.

5.7 Message ID

Message Identifier is a 16-bit positive integer assigned by the originator of the message. VCI protocol specification does not define rules for the assignment of the Message IDs.

[1] Required / Optional

Message ID should be assigned to every individually transmitted message (XML element) and to every message (XML element) that can be activated (or acted upon) independently.

A VCI message generated as a direct reply to a received VCI message includes Message ID of the message that triggered transmission of a reply or notification.

A log record generated to report erroneous or unexpected message or inability to act as specified in the received request includes Message ID when possible.

Message ID has been introduced to allow for tracking of the messages within the system, for debugging and troubleshooting. WIDAR correlator does not require Message ID to be a unique number, nor does it require Message IDs to be assigned in any particular sequence. It is up to the user (EMCS) to decide how it should be used.

6 VCI Request

A vciRequest is an "envelope element" used to transmit one or more of the following:

1. Configuration Request for stations and/or sybarrays,
2. Configuration Query for stations and/or subarrays,
3. CM status query,
4. CM configuration change request.

Table 6-1 lists attributes and elements (messages) that can be specified in the VCI Request. More than one element can be specified in a single vciRequest.

Table 6-2 lists parameters common to all the elements that can be specified as children of a VCI Request. With notable exception of the XML element *cmMonitorControl*, **Activation ID** is assigned to each element. **Activation time** is specified in Activation Trigger only.

vciRequest may contain elements with different Activation IDs and several Activation Triggers.

Message ID and Time Stamp have been added mostly as tools used for debugging during testing and system integration, and for troubleshooting in the normal operation mode.

6.1 Configuration Request

A configuration request may contain:

a) One or more instances of element stationHw and/or
b) Only one instance of element subarray and/or
c) One or more instances of ActivationTtrigger.

If a complete configuration consists of several instances of stationHw, one subarray and an Activation Trigger, those messages can be transmitted in one vciRequest or separately (e.g. each in a different instance of vciRequest).

Mapping time can be specified in the Activation Trigger. If the mapping time is not specified, mapping is performed immediately upon receipt of the ActivationTrigger. Attribute mapping time has been introduced to delay mapping in the case when different configuration messages (e.g. stationHw and subarray) are generated by different processes.

6.2 Configuration Query

If an ActivationTrigger contains *query='yes'*, all the messages with the same Activation ID are treated as a query. When a query is received, CM performs mapping and generates a report (accept or reject) but *does not* activate configuration (no configuration or status change in the correlator).

Table 6-1 VCI Request

Parameter/Group	R/O	Range	Description
XML tag	R	vciRequest	XML tag for the VCI Configuration Request.
Message ID	R	Integer	XML attribute. Message Identifier specified by the originator. Used as a reference in responses and logs.
Time Stamp	R	UT	XML attribute. Time when the message was generated (or transmitted). UT displayed in format defined by ISO 8601.
Version	O	3.9 3.10 3.11 3.12 3.13 3.14 3.15	Version of the VCI protocol. Corresponds to the version (revision) of this document.
Activation Trigger	O	XML element. Used to specify activation time for elements with the same Activation ID. See section 6.5.	
StationHardware	O	XML element. Station ID can be in the range 1 to 255, theoretically up to 255 stations can be specified. This element is used to assign Station ID and Baseband ID to a specific Station Board input data stream (baseband) and to identify Station Board hardware that should be used to process input data. See section 6.3.	
Subarray	O	XML element. Used to specify list of stations that belong to the subarray, input data (basebands) and desired output products. See section 6.4.	
CM Monitor & Control	O	XML element. Used to set CM parameters and query CM status. See section 6.6 on page 181.	

Table 6-2 Parameters common for the children of the VCI Request

Parameter	R/O	Range	Description
XML tag	R	stationHw subarray activationTrigger cmMonitorControl	Elements that can be specified in the vciRequest (children of the element vciRequest).
Message ID	R	$1..2^{\wedge 16}-1$	XML attribute. Used as a reference in response messages. It is up to the users to decide how to use this identifier.
Time Stamp	R	UT	XML attribute. The time when the message was created (or transmitted) . UT displayed in format defined by ISO 8601.
Activation ID	R	String (up to 32 char)	XML attribute. Activation ID, used to identify elements that should be activated as one configuration (at the same time).
Mapping Order	O	Positive Integer	XML attribute. At mapping time, all the requests with the same Activation ID are sorted according to the “Mapping Order”, Message with the lowest Mapping Order is processed first. Requests with no Mapping Order are added to the end of the queue and are processed in order in which they were received (they are processed after all the messages with assigned Mapping Order). As a consequence, if Mapping Order is not used at all, requests are processed in the order in which they were received. Mapping Order was introduced to provide for predictable and reproducible result of configuration mapping.

6.3 Station: Hardware

Element *stationHw* is used to identify Station Boards that belong to a particular station; it specifies the following:

- Station ID (assigned by the EMCS).
- List of Basebands. A maximum of 8 Basebands can be defined for a station. This is used to assign Baseband ID to a Station Board Data Path. Baseband ID must be unique for a station; the same Baseband ID can not be assigned to two different Station Board Data Paths, even if they receive input from the same source (as is the case for EVLA antenna with 8-bit samplers).
- Antenna, which is the source of input data. MCCC, Station Boards and Baseline Boards do not require antenna specific information; the Correlator Backend may need some of the antenna parameters in order to include them into output data.

Reasons for grouping station hardware related parameters into a separate element are:

1. It is expected that connections among the EVLA antennas and Station Boards will be fairly stable. In order to connect an antenna to a different set of Station Boards the antenna patch panel and other hardware has to be re-configured.
2. Baseband IDs rarely change and there is no need to specify BBIDs for every subarray.

However, nothing prevents user to transmit stationHw configuration with every subarray.

XML Example 6-1 is a typical configuration for an EVLA antenna.

XML Example 6-1 Station Hardware

```
<widar:stationHw sid="1" name="R1-St1(1)" activationId="CreateStations-8Racks"
action="add" msgId="106">
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-0" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-0" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-4" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-4" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-0" dataPath="0"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-0" dataPath="1"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-4" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-4" dataPath="1"/>
    <widar:antenna type="EVLA" id="ea01" name="Antenna1"/>
</widar:stationHw>
```

Note: ***Noise diode parameters are not set via VCI .***

Table 6-3 ***stationHw***

Station Parameter	R/O	Range	Description
XML tag	R	stationHw	XML tag. This element is used to assign Baseband ID (BBID) to Station Boards.
Station ID	R	1..255	XML attribute. Together with Baseband ID and Subband ID, Station ID is inserted in the output data stream of each filter (subband). For definition of the Station ID and other identifiers see EVLA System Numbering Plan [8] .
Name	O	ASCII String	XML attribute. If specified, name will be saved by the CM, but the name cannot be used instead of the Station ID to refer to a station.
Activation ID	R	String	XML attribute. Used to identify the activation trigger that applies to this message.
Mapping Order	O	Positive Integer	XML attribute. See Table 6-2.
Baseband Hardware	O	XML element. Up to 8 Basebands can be specified. See Table 6-4.	
Antenna	O	XML element. Antenna, which is source of the input data. See Table 6-5.	

Table 6-4 ***stationHw / basebandHw***

Baseband Parameter	R/O	Range	Description
XML tag	R	basebandHw	XML tag. Assigns Baseband ID to Station Board Data Path.
Baseband ID	R	0..7	XML attribute. Baseband ID (BBID), as defined in [8].
Station Board MLID	R	Rack-crate-slot	XML attribute. Module Location ID (MLID) as defined in [8].
Data Path	R	0, 1	XML attribute. Data Path Identifier (DPID) as defined in [8]. EMCS chooses the Station Board Data Path (i.e. filter bank and associated hardware) to be used for the Baseband.

6.3.1 Antenna

EMCS may specify Antenna ID and type. CM does not use Antenna ID and type to derive configuration parameters; this information, if provided, is used for information only (to be displayed when reporting configuration status). This element is specified as a part of element *stationHw*, it (typically) changes only when antenna configuration and/or connections between antenna and Station Board change.

Local oscillator is specified as part of subarray configuration (see 6.4.2), since it (typically) changes for each scan.

Table 6-5 ***stationHw / antenna***

Parameter	R/O	Range	Description
XML tag	R	antenna	XML Tag. Antenna configuration.
Antenna ID	R	ASCII String	XML attribute. Antenna ID. Maximum length 32 characters.
Name	O	ASCII String	XML attribute. ASCII name assigned to antenna.
Type	R	EVLA	XML attribute. Antenna type.

6.4 Subarray

Subarray configuration consists of:

a) List of stations (antennas),

b) Description of input data, and

c) Description of desired products.

Table 6-6 lists content of the XML element *subarray*.

XML element *subarray* can be used:

1. To add (configure) a new subarray,
2. To modify some of the parameters of an existing subarray,
3. To remove (de-configure) an existing subarray.

The terms “existing subarray” and “existing configuration” in this document means: a subarray (a configuration) that will be active at the specified activation time.

When creating a new subarray, list of stations cannot contain stations that already belong to another subarray. It is necessary to explicitly delete a subarray before the same stations can be assigned to another subarray.

Usually, all the stations that belong to the same subarray have identical configuration; if that is the case, there is no need to specify basebands, subbands and products for each station individually. Configuration that applies for all the stations can be specified only once. If configuration for some stations differes from the rest of the subarray, configuration for those stations may be specified individually.

Unless otherwise specified, CM assumes that correlation should be performed for all the baselines in a subarray (i.e. by default, all the baselines are enabled).

Table 6-6 ***subarray***

Subarray Parameter	R/O	Range	Description
XML tag	R	subarray	This element can be used to add a new subarray, and to modify or delete existing subarray.
Config ID	R	String	XML attribute. Config ID is used to identify subarray. By default, correlation is performed for all the stations that belong to the same subarray. Note: in release 3.9 attribute subarray ID has been replaced with config ID.
Subarray Name	O	String	XML attribute. Optional parameter used for display only.
Scan ID	O	String	XML attribute. ASCII string. Used for reference only, not used by the correlator.
Observation Time	O	Date and time in ISO9601 format. Default = UT	XML attribute. Allows VCI Client to set observation time to be different from the actual (wall clock) time.
Action	O	Create, Modify, Delete Default=Create	XML attribute. This parameter specifies what the correlator should do with the specified subarray.
Activation ID	R	String	XML attribute. See Table 6-37.
Mapping Order	O	Positive Integer	XML attribute. See Table 6-2.
Re-configure Complete Baseline Boards	O	Boolean, default=false	XML attribute. Normally, a subarray configuration re-configures only those row/columns on the Baseline Board that are affected by the configuration change. This parameter was introduced to “tidy up” remnants of the previous configurations and tests.
Epoch for DUMPTRIG	O	Date and time in ISO9601 format.	Epoch for DUMPTRIG. Applies for all basebands and subbands in the sybarray. ISO9601 format allows for millisecond precision; however scans can only be specified to begin on system ticks (10 ms). Default= 1970-01-01T00:00:00.00.000
List of stations	O	XML element. List of stations that belong to the subarray. The maximum number of Station IDs in the WIDAR correlator is 255, theoretically up to 255 stations can be specified. See Table 6-7	
Station Input and Output	O	XML element. Describes input and output data: basebands, subbands and output products. Usually, the same configuration applies for all the stations, but it can be specified for each station individually or for a subset of stations. See Table 6-10.	
Baseline	O	XML element. Disable/enable baselines. Not implemented. If not specified, all the baselines in the subarray are enabled. See 6.4.20.	

Note that Activation ID, Config ID and Scan ID have different significance:

- **Activation ID** is used to group the messages that should be activated at the same time. Activation time is specified in the Activation Trigger, which may be specified in the same vciRequest message.
- **Config ID** indicates which stations belong to the same subarray. By default, correlation is performed for all the stations that have the same Config ID. In other words, Config ID is used to identify baselines where correlation is required.
- **Scan ID** is specified as a part of a subarray configuration and is saved to archive with configuration and monitor data. WIDAR correlator uses Scan ID only as a label that accompanies control data.

For a simple example of a subarray configuration refer to XML Example 4-1 on page 34.

6.4.1 Modify Subarray

Only a subset of subarray parameters can be modified:

- SSLO (signed sum of local oscillators), can be specified for each baseband of each station, using one of the following:
  subarray/listOfStations/station/bb
  subarray/stationInputOutput/station/bb

- Sideband can be specified for each individual baseband (left/right or x/y), and applies for all the stations in a subarray; or for a group of stations that have the same configuration. Sideband is an attribute of element *bbParams*.
  subarray/listOfStations/bbParams
  subarray/stationInputOutput/bbParams

- Filter gain and/or desired RMS can be specified for a baseband pair. Filter gain and RMS apply for all subbands and for all stations in a subarray or for a group of stations that have the same configuration.
  subarray/listOfStations/baseBand
  subarray/stationInputOutput/baseBand

- Wade-Band Correlator (WBC) products:
  - ➢ can be specified and enabled/disabled using element
    subArray/baseBand/wpp
  - o WBC products for a baseband pair can be disabled (de-configured) using
    subArray/baseBand attribute *noWbcProducts*

- Summed array:
  - Subset of stations to be excluded from the sum can be changed (modify must contain complete list of stations to be excluded, not the delta).
  - Transmission of VDIF packets can be enabled/disabled (if VDIF was configured in “create subarray”).
  - Filter gain related parameters (automatic gain contol and target RMS) can be modified for data stream transmitted via VDIF and/or for data streams used to produce the auto-correlation products.

Other parameters, if specified in the modify request, are ignored. In other words, if VCI request ‘modify subarray’ contains parameters that cannot be modified, the request will be accepted and only the parameters that can be modified (as listed above) will be modified. This, somewhat relaxed, syntax checking has been explicitly requested by NRAO.

In the same manner, if VCI request “modify subarray” contains references to basebands that were not specified in “create subarray”, the request is accepted and unknown basebands are ignored. Modify cannot be used to add additional basebands or subbands.

Filter gain changes are specified per baseband and apply for all used filters. Unused (idle) filters are not affected.

XML Example 6-2 Modify filter gain for all stations in a subarray

```
<widar:vciRequest  desc="Modify_Subarray" xmlns:widar="http://www.nrc.ca/namespaces/widar">
    <widar:subArray configId="Subarray-7" activationId="Modify-all" action="modify" >
        <widar:stationInputOutput sid="all">
            <widar:baseBand bbB="2" bbA="0" defaultFilterGain="" stage1Rms="200.0" />
            <widar:baseBand bbB="3" bbA="1" stage1Rms="200.0"  />
        </widar:stationInputOutput>
    </widar:subArray>
    <widar:activationTrigger  activationId="Modify-all" activationTime="2010-08-17T18:25:00"/>
</widar:vciRequest>
```

XML Example 6-3 Modify filter gain for a subset of stations

```
<widar:vciRequest  desc="Modify_subset" xmlns:widar="http://www.nrc.ca/namespaces/widar">
    <widar:subArray configId="Test" activationId="Modify-Subset" action="modify" >
        <widar:stationInputOutput sid="list">
            <widar:station sid="1"/>
            <widar:station sid="2"/>
            <widar:baseBand bbB="2" bbA="0" defaultFilterGain=""/>
            <widar:baseBand bbB="3" bbA="1" defaultFilterGain=""/>
        </widar:stationInputOutput>
    </widar:subArray>
    <widar:activationTrigger activationId="Modify-Subset" activationTime="2010-11-01T18:55:00"/>
</widar:vciRequest>
```

XML Example 6-4 Modify enable VDIF, enable AGC, modify RMS for AGC

```xml
<widar:vciRequest  xmlns:widar="http://www.nrc.ca/namespaces/widar">
    <widar:subArray  configId="sub-34" activationId="modify-VDIF-AGC" action="modify">
        <widar:stationInputOutput sid="all">
            <widar:baseBand bbB="2" bbA="0">
                <widar:subBand sbid="0" swIndex="1" centralFreq="64000000" bw="128000000">
                    <widar:summedArray sid="100" >
                        <widar:vdif vdifEnableA="true" stationId="123" agcEnabled="true" agcRms="2.3"/>
                    </widar:summedArray>
                </widar:subBand>
            </widar:baseBand>
        </widar:stationInputOutput>
    </widar:subArray>
    <widar:activationTrigger activationId="modify-VDIF-AGC"/>
</widar:vciRequest>
```

XML Example 6-5 Modify SSLO, sideband and WBC products

```xml
<widar:vciRequest xmlns:widar="http://www.nrc.ca/namespaces/widar">
    <widar:subArray configId="vciDoc-ExampleOne" activationId="Modify-3" action="modify">
        <widar:stationInputOutput sid="list">
            <widar:station sid="1">
                <widar:bb bbid="0" localOsc="-888.0" freqShift="-20.0"/>
                <widar:bb bbid="1" localOsc="42640"  freqShift="-25.6"/>
                <widar:bb bbid="2" localOsc="1.0e8"   freqShift="1024000.0"/>
                <widar:bb bbid="3" localOsc="-888.0" freqShift="-60.0"/>
            </widar:station>
            <widar:bbParams bbid="0" sideband="upper"/>
            <widar:bbParams bbid="1" sideband="upper"/>
            <widar:baseBand bbB="0" bbA="2">
                <widar:wpp id="0" correlation="A*A" spectralChannels="64" integFactor="100"/>
                <widar:wpp id="2" correlation="B*B" spectralChannels="64" integFactor="100"/>
            </widar:baseBand>
        </widar:stationInputOutput>
        <widar:stationInputOutput sid="list">
            <widar:station sid="2"/>
            <widar:station sid="3"/>
            <widar:bbParams sideband="lower" bbid="0"/>
            <widar:bbParams sideband="lower" bbid="1"/>
        </widar:stationInputOutput>
    </widar:subArray>

    <widar:activationTrigger activationId="Modify-3" activationTime="2010-11-17T18:25:00"/>
</widar:vciRequest>
```

6.4.2 List of Stations

XML element *listOfStations* can be used to specify list of stations that belong to a subarray.

When creating a new subarray, list of stations may be specified either using XML element *listOfStations* (as described here) or by adding stations to the XML element *stationInputOutput*. *The latter method, introduced in release 3.6, is preffered.*

When deleting (de-activating) a subarray, list of stations is not needed, all the stations that belong to the subarray will be released and returned to the pool of unassigned (or unused) stations.

Only stations that are already known, i.e. that were previously configured using element *stationHw*, can be specified in the list of stations. StationHw configuration can be transmitted only once (e.g. when the CM is started) and used until there is a change in antenna-to-Station Board connectivity. If, at the mapping time, CM finds that the hardware configuration for some of the stations is not known, configuration is rejected.

Only an unassigned station, i.e. one that is not a member of any other subarray, can be added to a subarray. If a station already belongs to a different subarray, the subarray to which it currently belongs must be deleted before it can be included in any other subarray. **Ability to add and remove antennas (stations) to/from a subarray is not required in the EVLA system and is not implemented.** The only way to remove a station from a subarray is to delete the entire subarray. The only way to add a station to a subarray is to delete the subarray and then create it again with a new (complete) list of stations.

Note that, at the mapping time, CM does not consider the current correlator configuration, but the correlator configuration as it will be at the specified activation time.

Table 6-7 ***subarray / listOfStations***

Parameter	R/O	Range	Description
XML tag	M	listOfStations	XML element.
Type	O	add default = add	**Add** is the only valid value - this parameter has become obsolete and could be removed.
Station	O	XML element. Station to be included in the subarray. The maximum number of elements in the list is the maximum number of different Station IDs in the WIDAR correlator (255). See Table 6-8.	

6.4.3 Station

XML element *station* is used to specify the list of stations that belong to a subarray. It can be specified in the *listOfStations* or in *stationInputOutput*.

This element is used to specify the signed sum of local oscillators (SSLO). Normally, SSLO is specified in the delay models, but in some cases (mostly for tests and experiments) users may want to specify SSLO via “create subarray” and/or “modify subarray”.

SSLO is different for each baseband of each station.

SSLO is specified as two decimal numbers:

1. Local oscillator offset and
2. Frequency shift.

Table 6-8 ***subarray / listOfStations / station***

Station Parameter	R/O	Range	Description
XML tag	R	station	XML element. Used to add /remove stations to/from subarray.
Station ID	R	1..255	Station ID as specified in [8].
name	O	ASCII string	ASCII string assigned to a station.
bb	O	XML element used to specify local oscillator and frequency shift.	

Table 6-9 ***subarray / listOfStations / station / bb***

Station Parameter	R/O	Range	Description
XML tag	R	bb	XML element. Used to add /remove stations to/from subarray.
BB ID	R	0..7	Baseband ID as specified in [8].
localOsc	O	Decimal number	Local oscillator offset in Herz. Can be specified in create and modify subarray.
freqShift	O	Decimal number	Frequency shift in Hertz. Can be specified in create and modify subarray.

6.4.4 Station: Input and Output

Element *stationInputOutput* is used to specify basebands, subbands and output products. Typically, the same basebands, subbands and products are defined for all the stations in a subarray. If that is not the case, an instance of the element stationInputOutput must be specified for each group of stations (antennas) that have the same configuration.

Table 6-10 ***subarray / stationInputOutput***

Station Parameter	R/O	Range	Description
XML tag	R	stationInputOutput	Description of input and output data specifies: a) input data (basebands), b) Station Board output (subbands) c) desired output products.
Station ID (SID)	R	1..255, all, list	Value “all” can be used if the list of stations is specified in the element subarray (pre v3.6 format). See XML Example 6-6. Value “list” when stations to which the configuration applies are listed within element *stationInputOutput*. See XML Example 6-7. Integer is specified if configuration applies to a single station. Preffered method: list
Station	O	XML element. Station ID of the station to which configuration applies. If SID=’list’at least one station must be specified. For the SID range see Table 6-8.	
bbParams	O	XML element. Baseband parameters (contains parameters that must be specified for individual basebands, and not per pair). See section 6.4.4.1.	
Baseband	O	XML element. See 6.4.5. Up to 4 baseband pairs may be specified.	

XML Example 6-6 is a subarray configuration in pre-3.6 format: list of stations is specified in the element *subarray*, while *stationInputOutput* uses “sid=all”.

XML Example 6-7 shows the same configuration in format introduced in Release 3.6. This format applies to all VCI and CM releases including and after 3.6. List of stations is specified within element *stationInputOutput*.

<u>**Advantages of the 3.6 format become obvious when a subarray consist of antennas with different samplers (3-bit and 8-bit).**</u>

XML Example 6-6 Subarray configuration - pre-v3.6 format.

```
<widar:subArray configId="A-1-0" activationId="Create-A-1-0" action="create"  msgId="295">

        <widar:listOfStations action="add">
            <widar:station sid="1"/>
            <widar:station sid="2"/>
            <widar:station sid="3"/>
            <widar:station sid="4"/>
        </widar:listOfStations>

        <widar:stationInputOutput sid="all">

        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="1"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="2"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="3"/>

        <widar:baseBand inQuant="8" bw="1024000000" swbbName="AC_8BIT" bbB="2" bbA="0">
           <widar:subBand sbid="10" swIndex="1" centralFreq="64000000" bw="128000000"
                                                 rqNumBits="4">
                    <widar:polProducts>
                        <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                        <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                        <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                        <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
                        <widar:blbProdIntegration recirculation="1"
                                                        minIntegTime="250"
                                                        ltaIntegFactor="4000"
                                                        ccIntegFactor="1"
                                                       cbeIntegFactor="1"/>
                        <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
                        <widar:stationPacking  algorithm="maxPack"/>
                        <widar:productPacking algorithm="maxPack"/>
                        <widar:autoCorrSubset algorithm=" halfStationsMaxProd "/>
                    </widar:polProducts>
            </widar:subBand>
        </widar:baseBand>

    </widar:stationInputOutput>

</widar:subarray>
```

XML Example 6-7 Subarray configuration – format introduced in Release 3.6.

```
<widar:subArray configId="A-1-0" activationId="Create-A-1-0" action="create"  msgId="12">

        <widar:stationInputOutput sid="list" >

        <widar:station sid="1"/>
        <widar:station sid="2"/>
        <widar:station sid="3"/>
        <widar:station sid="4"/>

        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="1"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="2"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="3"/>

        <widar:baseBand inQuant="8" bw="1024000000" swbbName="AC_8BIT" bbB="2" bbA="0">
            <widar:subBand sbid="10" swIndex="1" centralFreq="64000000" bw="128000000"
                                                  rqNumBits="4">
                <widar:polProducts>
                    <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                    <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                    <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                    <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
                    <widar:blbProdIntegration recirculation="1"
                                              minIntegTime="250"
                                              ltaIntegFactor="4000"
                                              ccIntegFactor="1"
                                              cbeIntegFactor="1"/>
                    <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
                    <widar:stationPacking  algorithm="maxPack"/>
                    <widar:productPacking algorithm="maxPack"/>
                    <widar:autoCorrSubset algorithm="halfStationsMaxProd"/>
                </widar:polProducts>
            </widar:subBand>
        </widar:baseBand>

    </widar:stationInputOutput>

</widar:subArray>
```

6.4.4.1 Element bbParams

There are 8 sources of the input data on the Station Board:

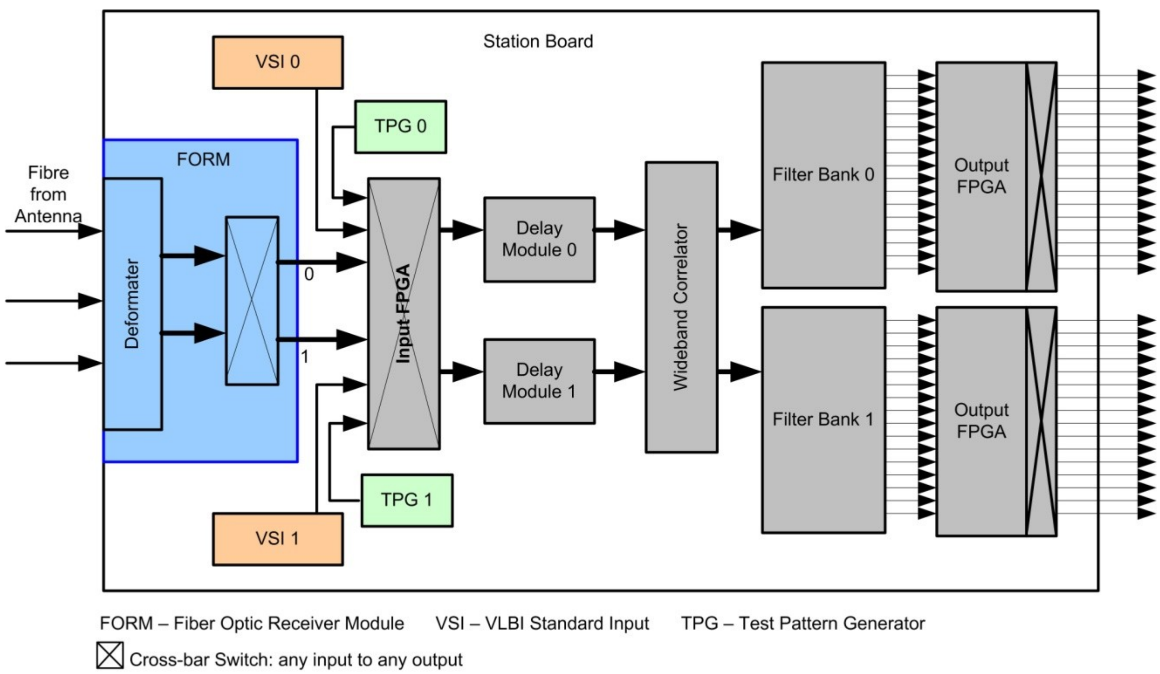
- Fibre Optic Receiver Module (FORM) is an optical to digital converter. FORM is mounted on the Station Board as a daughter card and has *two* outputs: 0 and 1. Any of the data streams in the optical input can be forwarded to any FORM output.

- There are *two* VSI FPGAs on the Station Board; either of the two VSI FPGAs can be used to feed VLBI recorded data to either of the two Station Board Data Paths.

- *Two* Test Pattern Generators (TPGs) are implemented in the Station Board Input FPGA. Output of either of the two TPGs can be forwarded to either of the two Station Board Data Paths.

- *Two* Delay Model Test Vector generators, one per Delay Module. Filters can use Delay Module Test Vector generated by the Delay Module which is located on the same data path.

Figure 6-1 Station Board Input



FORM outputs (0 and 1) are connected to the Station Board Input FPGA. Station Board Input FPGA implements a cross-bar switch that can forward data from either of the two FORM outputs to either of the two Station Board Data Paths.

Station Board input from an EVLA antenna with 3-bit samplers consists of two Basebands. FORM performs optical to digital conversion and de-multiplexing and can forward either of the two Basebands (input data streams) to either of the two Station Board Data Paths.

Input from an EVLA antenna with 8-bit samplers consists of one input data stream. FORM can forward input data to either of the two FORM outputs, or to both.

N.B. Both FORM and Station Board Input Chip implement cross-bar switch and are able to forward either of the two inputs to either of the two outputs; it is up to the EMCS to decide whether switching, if needed, is performed in the FORM or in the Station Board Input FPGA.

Although mounted on the Station Board, the Fiber Optic Receiver Module (FORM) has its own monitor & control interface; it is up to the EMCS to ensure that FORM configuration is consistent with configuration specified in bbParams (which is mapped into Station Board configuration).

XML Example 6-9 is a configuration for EVLA antennas with 3-bit samplers[1].

XML Example 6-10 is a configuration for EVLA antennas with 8-bit samplers[1].

Note that the same *stationHw* configuration, specified in XML Example 6-8, applies for both.

XML Example 6-8 stationHw configuration


```
<widar:stationHw sid="1" name="R1-St1(1)" activationId="CreateStations-8Racks"
                 action="add" msgId="106">
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-0" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-0" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-4" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-4" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-0" dataPath="0"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-0" dataPath="1"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-4" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-4" dataPath="1"/>
    <widar:antenna type="EVLA" id="ea01" name="Antenna1"/>
</widar:stationHw>
```


[1] In this document quantization performed by antenna samplers is referred to as 'initial quantization' (IQ), as opposed to re-quantization (RQ) performed in Station Board filters (per subband).

Table 6-11 ***subarray / stationInputOutput / bbParams***

Baseband Parameter	R / O	Range	Modify	Description
XML tag	R	bbParams		Baseband parameters that can, and usually are, different for basebands in a baseband pair.
Baseband ID	R	0..7	No	Baseband ID as specified in [8]. Only “known” Baseband IDs, i.e. Baseband IDs that are configured in stationHw, can be specified.
Source Type	R	FORM VSI TestPatternGenerator dmTestVector	No	Device type for the source of input data. Station Board implements two types of test pattern generators: Two TestPatternGenerators are implemented in the Input FPGA. Each test pattern generator can be turned on/off independently; output of each test pattern generator can be forwarded to any data path. Each of the 2 Delay Modules implements test vector generator and each can be turned on/off independently.
Source ID	R	0, 1	No	Identifier for the source of input data. sourceType=FORM: identifier of the FORM output (FOID); sourceType=VSI: identifier of the VSI chip (as defined in hardware); sourceType=Test Pattern Generator (TPG): identifier of the TPG implemented in the Station Board Input FPGA. For sourceType=dmTestVector: irrelevant. Filters can use only test vector generated by the Delay Module located on the same Data Path.
Polarization	R	R, L, X, Y	No	Polarization ID (POLID) as defined in [8]. Used by the correlator to differentiate basebands which belong to the same pair.
Sideband	O	Upper, Lower	Yes	Baseband sideband. Upper means input is arranged so that samples start from lower frequency. Lower means input is arranged so that samples start from higher frequency. Can be specified both in create subarray and modify subarray.
Phase model insertion	O	Early, Late	No	CM transparently passes this parameter to CMIB software. If not specified in ‘create subarray’ CM sets the default value: o late for Station Boards in slots 0-3 o early for Station Boards in slots 4-7. This parameter is used to specify if phase model for the baseband is delivered early or late in the 10ms window. This information is used by CMIB software to allow ANDing of phase models from different station boards on the crossbar boards. For detailed explanation refer to CMIB software documentation.

Parameters Source Type, Source ID and Polarization are marked as required, they must be specified in ‘create subarray’.

In ‘modify subarray’ only Baseband ID and Phase model insertion should be specified. Other parameters, if specified, are ignored.

XML Example 6-9 EVLA antennas with 3-bit samplers (stationHw as in XML Example 6-8)

```
<stationInputOutput sid="list">

  <station sid="1"/>
  <station sid="2"/>
  <station sid="3"/>
  <station sid="4"/>
  <station sid="5"/>

  <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
  <bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L" bbid="1"/>
  <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="2"/>
  <bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L" bbid="3"/>
  <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="4"/>
  <bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L" bbid="5"/>
  <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="6"/>
  <bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L" bbid="7"/>

  <baseBand inQuant="3" bw="2048000000" swbbName="A1C1_3BIT" bbB="1" bbA="0">

    <subBand sbid="1" swIndex="1"
             centralFreq="64000000" bw="128000000" rqNumBits="4" >
      <polProducts>
        <pp spectralChannels="64" id="1" correlation="A*A"/>
        <pp spectralChannels="64" id="2" correlation="A*B"/>
        <pp spectralChannels="64" id="3" correlation="B*A"/>
        <pp spectralChannels="64" id="4" correlation="B*B"/>
        <blbProdIntegration recirculation="1" minIntegTime="210.0"
                            ltaIntegFactor="4000" ccIntegFactor="1"/>
        <blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
        <stationPacking  algorithm="maxPack"/>
        <productPacking algorithm="maxPack"/>
        <autoCorrSubset algorithm="allStationsMaxProd"/>
      </polProducts>
    </subBand>

  </baseBand>

</stationInputOutput>
```

XML Example 6-10 EVLA antennas with 8-bit samplers (stationHw as in XML Example 6-8)

```
<stationInputOutput sid="list">

    <station sid="1"/>
    <station sid="2"/>
    <station sid="3"/>
    <station sid="4"/>
    <station sid="5"/>

    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="1"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="2"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="3"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="4"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="5"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="6"/>
    <bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="7"/>

    <baseBand inQuant="8" bw="1024000000" swbbName="AC_8BIT" bbB="2" bbA="0">

        <subBand sbid="1" swIndex="1"
                  centralFreq="64000000" bw="128000000" rqNumBits="4">
            <polProducts>
                <pp spectralChannels="64" id="1" correlation="A*A"/>
                <pp spectralChannels="64" id="2" correlation="A*B"/>
                <pp spectralChannels="64" id="3" correlation="B*A"/>
                <pp spectralChannels="64" id="4" correlation="B*B"/>
                <blbProdIntegration recirculation="1" minIntegTime="210.0"
                                    ltaIntegFactor="4000" ccIntegFactor="1"/>
                <blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
                <stationPacking  algorithm="maxPack"/>
                <productPacking algorithm="maxPack"/>
                <autoCorrSubset algorithm="allStationsMaxProd"/>
            </polProducts>
        </subBand>

    </baseBand>

</stationInputOutput>
```

6.4.5 Baseband

XML element *baseband* describes a pair of Station Board input data streams (baseband pair), and desired output products.

All parameters (except Baseband ID) apply to both basebands. Parameters that can be different for each baseband are specified using element bbParams (e.g. polarization).

A station can have up to 4 baseband pairs.

Support for individual basebands is not provided.

Table 6-12 ***subarray / stationInputOutput / baseband***

BaseBand Pair Parameter	R/O	Range	Modify	Description
XML Tag	R	baseband		XML element.
bbA	R	Baseband ID: 0-7	No	Baseband IDs as specified in *stationHw*. Letters A and B are used to differentiate basebands configured as polarized pair.
bbB	R	Baseband ID: 0-7	No	
swBbName	R	A1C1_3BIT A2C2_3BIT B1D1_3BIT B2D2_3BIT AC_8BIT BD_8BIT	No	Baseband name for spectral window. Required for BDF.
Bandwidth	R	2048000000 1024000000 512000000 256000000 128000000 64000000 32000000 16000000 8000000 4000000 2000000 1000000 500000 250000 125000 62500 31250	No	Baseband bandwidth in Hz. For 8-bit input maximum bandwidth is 1024000000
Initial quantization	R	1..8	No	Number of bits per sample in the input data stream.

BaseBand Pair Parameter	R/O	Range	Modify	Description
Single phase center	O	Yes / No Default: Yes	No	YES: all subbands have the same phase center, i.e. delay models will be specified for Basebands only. NO: each subband can has its own phase center. In addition to the Baseband delay models, differential subband delay models will be supplied.
swPwrEpoch	O	*time* *default = 0*	No	Epoch for switched power. Format ISO8601.
swPwrInteg	O	Integer	No	Integration time in milliseconds.
Default filter gain	O	none	Yes	If present, this parameter is forwarded to Station Boards. Station Board software built-in algorithm sets filter gain based on the description of input data and filter setup. Can be specified in create subarray and modify subarray. See section 6.4.6 on page 71.
Stage 1 RMS	O	Decimal number	Yes	Desired RMS. If specified, RMS value is forwarded to Station Boards. Station Board software measures RMS and adjusts filter gain to achieve desired RMS. If not present, filter gain remains as previously set. Filter gain attributes can be specified in create and modify subarray (per baseband pair).
Stage 2 RMS	O	Decimal number	Yes	
Stage 3 RMS	O	Decimal number	Yes	
Stage 4 RMS	O	Decimal number	Yes	
Requnatizer RMS	O	Decimal number	Yes	
noWbcProducts	O	*String*	Yes	This attribute is only relevant in 'modify subarray'. If present this attribute will cause CM to deconfigure all WBC products for the baseband pair. Content of the string is irrelevant; best practice is to use empty string: noWbcProducts="". See XML Example 16-7 and XML Example 16-8 on page 229.
binningPeriod	O	*milliseconds*	No	Period for phase binning. Must be specified if phase binning is ON for at last one subband. Specified in milliseconds as floating point number.
binMaxHwIntegTime	O	*Milliseconds* *Default=500.0*	No	Maximum integration time for phase binning. Relevant only when phase binning is used. If not specified the default value of 500 miliseconds is used.Specified in milliseconds as floating point number.
Subband	O	XML element	Yes	XML element. Up to 18 subbands. *See section 6.4.8*
phaseBinning	O	XML element	No	XML element. There can be up to 2000 instances per baseband. See Table 6-13 below and section 6.4.14 on page 159.

BaseBand Pair Parameter	R/O	Range	Modify	Description
Pulsar Gating	O	XML element	-	XML element. **Not implemented.** *See section 6.4.18*
WBC Product	O	XML element	Yes	XML element. Wideband correlation product. See section 6.4.7 on page 72.

Table 6-13 ***subarray / stationInputOutput / baseband/ phaseBinning***

Phase Binning Parameter	R/O	Range	Modify	Description
XML Tag	R	`phaseBinning`	No	XML element.
phase	O	`0.0 to 1.0` `Default: 0.0`	No	Start of the first phase bin relative to epoch as a fraction of the binning period. Optional parameter, if not specified it is assumed that phase is 0.0.
binWidth	R	`0.0 to 1.0`	No	Bin width as a fraction of the binning period.
numBins	O	`1 to 2000` `Default: 1.`	No	Number of adjacent phase bins that have the same width. If not specified, CMIB assumes one.

6.4.6 Filter Gain

Filter gain related parameters are listed in Table 6-12 on page 68.

On the Station Board, incoming data stream (baseband) is forwarded to a filter bank that consists of 18 filters. Output of 16 filters can be forwarded to Baseline Boards, where correlation and integration is performed. All 18 filters in the filter bank have identical design (hardware and firmware) but each filter can be individually configured. Station Board hardware, firmware and software allow user to set filter gain for each stage of each filter independently.

Note:

- **Gain is specified per baseband pair, not for individual subbands (filters).**[1]

- **If a VCI message does not contain parameters related to filter gain, filter gain setup does not change. In other words, CM does set the 'default values' for filter gain.**

Each Station Board filter has four stages. Number of stages used for a specific configuration (baseband/subband) is determined by CMIB software based on input and output bandwidth. At each stage, the output of the filter summers (up to 32 bits, depending on stage but no loss of bits) is multiplied by the 16 bit filter gain and symmetrically rounded (and clipped) to 16 bits. Output of one of the four filter stages is selected as output of the filter, multiplied by the re-quantizer gain and symmetrically rounded (and clipped) to the number of bits requested for the output samples.The sum-of-squares (power) of the any of the four stages and the re-quantizer output is measured and can be used by the software to calculate the RMS of the specified signal. RMS stands for Root Mean Square. RMS value of a set of values is the square root of the arithmetic mean of the squares of the original values. More detailed description is provided in document Station Board Filter FPGA RFS [7].

Parameter "set default filter gain", if specified in VCI message, is forwarded to Station Board(s). Station Board CMIB software uses built-in algorithm to derive filter gain for the used stages of the used filters. Unused filters and unused filter stages are not affected. Alternatively (or in addition), user can specify desired RMS for each stage individually. Any subset of RMS values (including *none* and *all*) can be specified for any baseband. For example, if RMS is specified for stage 1 only, Station Board software measures RMS after stage 1, compares the calculated value with the desired value, and adjusts stage 1 gain accordingly. Gains for other stages in the same filter are not affected.

Filter gain related parameters are optional; user can specify any or all the parameters in the same instance of the element *baseband*. When both "set default filter gain" and desired RMS (for one or more stages) are specified, Station Board software first sets the

[1] If in the future that becomes a requirement, same or similar attributes can be added to element *subband*.

default filter gain; then it measures RMS and adjusts filter gain to acheive desired RMS. This allows user to set default gain for all the stages and modify filter gain for a subset of stages using the same message.

6.4.7 Wide Band Correlator

Wide Band Correlator (WBC) on the Station Board can produce correlation products for wideband input (basebands).

For EVLA antennas with 3-bit samplers (3-bit initial quantization) Station Board input consists of two basebands, configured as polarization pair. Wide Band Correlator on the Station Board can be used to get auto-correlation products for those two basebands and cross-corelation products between the basebands, i.e. products A*A, B*B, A*B and B*A[1].

For EVLA antennas with 8-bit samplers (8-bit initial quantization), each Station Board gets different polarization; wideband correlator can get auto-correlation products A*A and/or B*B. It is not possible to get polarization products A*B or B*A.

Wide Band Correlator can produce 64 lags at a time, which gives 64 spectral channels for the auto-correlation products (A*A and B*B) and 32 spectral channels for the cross-correlation products (A*B and B*A).

More spectral channels per product can be obtained using time-multiplexing; lags can be obtained in blocks of 64 lags per interrupt and “stiched” together. This results in loss of sensitivity; e.g. if a single cross-correlation product with 64 spectral channels is desired, every second data sample will be correlated.

Integration factor is the number of times a complete lag set is obtained and integrated, before it can be off-loaded.

Station Board CMIB software implements algorithm that can produce multiple products and/or more spectral channels per product in the following order: first all the lags for the first product (product with the smallest Product ID), then all the lags for the second product, and so on, till the end of the list. When the full set of lags is collected for all products, software increments integration counter, and repeats the process. It also checks whether the specified integration factor has been reached, if so, it switches to the other buffer and starts from the beginning. Two buffers are used, so that, while one set of data is being collected, users have access to the other.

In the VCI request WBC products are specified per baseband pair; configuration applies for all the stations in a subarray. If required, VCI protocol could be updated to allow users to specify WBC products for individual stations.

[1] Usually only one cross-product A*B or B*A is required.

Create subarray:

- By default WBC products are disabled. More precisely: if not explicitly specified in *create subarray* WBC products are disabled.
- There is no need to specify attribute *noWbcProducts* in *create subarray*; WBC products are disabled by default.
- If *create subarray* for the same baseband pair specifies both attribute *noWbcProducts* and at least one instance of element *wpp*, element *wpp* overrides the attribute *noWbcProducts*.

Modify subarray:

- If *modify subarray* contains <baseband noWbcProducts=' ' bbA='2' bbA='0' /> all WBC products for the baseband pair 0/2 will be disabled.
- If *modify subarray* contains at least one instance of the element *wpp*, new configuration will completely replace previously configured WBC products specified either in *create subarray* or in previously received *modify subarray*.
- To disable one product while leaving others unchanged, use
  <wpp id='1' status="disable" />
- If message *modify subarray* contains neither element *wpp* nor attribute *noWbcProducts*, WBC configuration remains unchanged.
- If message *modify subarray* for the same baseband pair specifies both attribute *noWbcProducts* and at least one instance of element *wpp*, element *wpp* overrides the attribute *noWbcProducts*.

Table 6-14 ***subarray / baseband / wpp***

Product parameter	R/O	Range	Must be same for all stations	Description
XML tag	R	wpp	-	XML element.
Product ID	R	Positive integer, range [0..31]	No	Product ID assigned by the originator of the message.
Correlation	R	A*A A*B B*A B*B	No	Data streams to be correlated.
Number of spectral channels	O	64*n, where n is in range [1..64] Default=64	No	Number of spectral channels per product.
Integration Factor	O	Positive integer Default=1	No	Number of complete data sets to be integrated.
Status	O	enable, disable Default=enable	No	This attribute can be used in ‘modify subarray’ to disable previously enabled products.

XML Example 6-11 WBC Products for 3-bit Initial Qunatization

```
<widar:baseBand inQuant="3" bw="1024000000" swbbName="A1C1_8BIT" bbB="3" bbA="2">
   <widar:subBand sbid="1" swIndex="1" centralFreq="64000000" bw="128000000" rqNumBits="4">
      <widar:polProducts>
         <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
         <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
         <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
         <widar:blbProdIntegration recirculation="1" minIntegTime="250.0" ltaIntegFactor="4000"
                                                                ccIntegFactor="1"/>
         <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
         <widar:stationPacking  algorithm="maxPack"/>
         <widar:productPacking algorithm="maxPack"/>
         <widar:autoCorrSubset algorithm="halfStationsMaxProd"/>
      </widar:polProducts>
   </widar:subBand>
   <widar:wpp id="1" correlation="A*A"  spectralChannels="64" integFactor="100"/>
   <widar:wpp id="2" correlation="B*B"  spectralChannels="64" integFactor="100"/>
   <widar:wpp id="4" correlation="A*B"  spectralChannels="64" integFactor="100"/>
</widar:baseBand>
```

6.4.8 Subband

WIDAR Station Board has two banks of 18 filters. Wideband input data stream received from the antenna (more precisely, from the Fibre Optic Receiver Module - FORM) can be decimated into up to 36 narrow band streams, called subbands.

On the Station Board, output of any filter can be forwarded to any output wafer. Also, output of a single filter can be forwarded to any number of output wafers. This means that output of any filter (any subband) can be forwarded to any Baseline Board, and that output of a single filter can be forwarded to any number of Baseline Boards (up to and including all Baseline Boards in the correlator).

However, only 16 out of 18 Station Board output wafers are connected to Crossbar Boards and through them to Baseline Boards. In other words, output of 16 filters per filter bank can be forwarded to Baseline Boards and correlated.

Remaining two filters in each filter bank can be used:

- As backup. CM software is able to 'rout around' filters that are declared as disabled. Even if up to two filters in the same filter bank fail, Station Board still can produce 16 subbands.
- For calibration. In the case where the correlated subband is either very narrow or strongly affected by RFI, one of spare filters can be used to gather calibration (switched power) data. If needed, both spare filters can be used for calibration.

It is possible to define and configure a subband that has no output products. In such case there is no need to forward output of the filter (subband) to Baseline Boards.

For each baseband pair there can be:

- **up to 16 subbands that require output products and**
- **up to 2 subbands that don't require (Baseline Board generated) output products.**

Each Station Board filter is configured independently; therefore each subband can have different bandwidth, central frequency, number of bits for re-quantizer, and other parameters.

The following limitation applies:
Different Subband ID must be assigned to each filter in the same filter bank.

This is not hardware limitation; this limitation is enforced by software. Rationale: Each station board output stream should have unique set of identifiers SID/BBID/SBID.

For each subband, VCI configuration can include description of desired polarization products and a list of Baseline Boards that should be used to produce those products. Any pair of Baseline Boards can be assigned to any subband. Some restrictions apply. See section 6.4.10.

EVLA antennas with 3-bit samplers can have up to 4 baseband pairs per antenna (station). User may define up to 16 subbands per baseband pair. Therefore, the maximum number of subbands per subarray is 16*4 = 64.

EVLA antennas with 8-bit samplers can have up to 2 baseband pairs per antenna (station). User may define up to 32 subbands per baseband pair. Therefore, the maximum number of subbands per subarray is 32*2 = 64.

128 Baseline Boards in the EVLA correlator are arranged as 64 pairs, each pair may be used to correlate different subband. Alternatively, in order to get products with more spectral channels, several BlB pairs may be assigned to a single subband. In other words, user can trade number of subbands for number of spectral channels.

6.4.8.1 Subband Bandwidth and Central Frequency

It is assumed that Baseband[1] is adjusted so that frequency range starts from zero. This section describes the rules that should be followed when choosing subband bandwidth and central frequency. Note however that **CM does not enforce these rules.**

The following acronyms are used in this section:

```
BBbw  – baseband bandwidth
SBbw  – subband bandwidth
SBcf  – subband central frequency
```

- **Subband bandwidth and central frequency must be defined so that subband is within baseband, i.e.**

  `((SBcf – SBbw/2) >= 0) and ((SBcf + SBbw/2) =< BBbw)`

- **Subband central frequency must be defined so that subbands do not cross 128MHz slots.**

- **If subband bandwidth is less than 128MHz, and *stage 2 mixer is used*, subband can be placed anywhere within 128MHz slot.**

- **If subband bandwidth is less than 128MHz, and *stage 2 mixer is not used*, subband must be placed in the slots defined by the subband bandwidth**, i.e. the following must apply:

  `SBcf = 128MHz*m + (SBbw*n) + SBbw/2,`

  where:

  n is an integer in the range [0, (128MHz/SBbw)-1][2]

  m is an integer in the range [0, 15]

  and

  `((SBcf-SBbw/2)>=128MHz*m) and ((SBcf + SBbw/2)=<(128MHz*(m+1))`

VCI protocol allows user to explicitly specify whether Stage 2 Mixer is to be used or not. For a discussion related to use of the Stage 2 Mixer, please refer to the "Programmer's Guide to EVLA Correlator System Timing, Synchronization, Data Products and Operation" [2].

---

[1] Baseband - Input data stream from antenna or any other source.

[2] Square brackets indicate that range includes the lower and upper limits, in this particular case: 0 and (128MHz/SBbw)-1 are included.

Table 6-15 ***subarray / stationInputOutput / baseband / subband***

Subband parameter	R / O	Range	Same for all stations	Modify	Description
XML tag	R	subBand	-		subBand is XML element.
subband ID	R	Integer in the range 0..17	Yes	No	Subband Identifier.
swIndex	R	Integer	Yes	No	Spectral window index. swIndex is passed to CBE and used when writing data in BDF. **For each baseband and each value of total integration time swIdexes must form a list of consecutive integers starting with 1.(** In order to be placed in the same BDF products must have the same total integration time. For more information refer to BDF and EVLA Correlator Backend Documention.)
name	O	ASCII string	No	No	For display only.
Bandwidth	R	128000000 64000000 32000000 16000000 8000000 4000000 2000000 1000000 500000 250000 125000 62500 31250	Yes	No	Subband bandwidth in Hz. The maximum value that can be specified for a subband bandwidth is BBbw/16 or 128MHz, whichever is smaller. The minimum value that can be specified for a subband bandwidth is BBbw/4096 or 31,250Hz, whichever is bigger.
Central Frequency	R	Decimal number	Yes	No	Central frequency in Hz. For example: 32000000.123 Must be specified so that subband is within baseband. Subband can not "cross" 128MHz slots. Restrictions that apply to subbands with bandwidth less than 128MHz and use of Stage 2 Mixer, are described in the text.
Re-quantizer	R	Integer in the range [1..8]	Yes	No	Number of bits for re-quantizer, i.e. the number of bits per sample in the Station Board output data stream. Baseline Board can handle only 4 bit and 7 bit data streams (subband), which means that, if Baseline Board products are required, 4 bit or 7 bit re-quantization must be used.

Subband parameter	R / O	Range	Same for all stations	Modify	Description
Use Stage 2 Mixer	O	Yes, No Default=Yes if SBbw<128MHz	Yes	No	Stage 2 mixer can be used only when subband width < 128MHz. For subbands with SBbw=128MHz this parameter is irrelevant. When the mixer is OFF, subbands must be placed in the fixed slots. When the mixer is ON, subbands may be positioned anywhere within 128MHz slots.
Mixer Phase Error Correction	O	Yes, No if Stage 2 Mixer = Yes Default = Yes Otherwise irrelevant.	Yes	No	Fractional Sample Delay Error Correction This parameter allows VCI Client to explicitly specify whether or not the Stage 2 mixer should update the phase error and model (or in other words: perform fractional sample delay error correction). If "on": stage2 mixer updates phase error and models; phase error is set to zero. This parameter is relevant only when Stage 2 Mixer is "on" (subband width must be < 128MHz). **Not implemented.** **Disabled by the CMIB software.**
Signal to noise ratio (STNR)	O	Integer in the range 0..100 Default=0	Yes	No	Percent of signal in the input data. If not specified, it is assumed that the input is noise dominated (i.e. that percentage of signal in the input is 0). Used to determine scaling factors (for each of the filter stages and for the re-quantizer). For more information refer to StB Filter Chip RFS [7]. **Not implemented.**
Phase for pulsar gating	O	Integer in the range 0..100	Yes	No	Specifies the shift of the pulsar gate in fractions of cycle. Required when pulsar gating is on. Otherwise irrelevant. See 6.4.18. **Not implemented.**
phaseBinning	O	On, off Default=off	Yes	No	Relevant only when binningPeriod is specified for the baseband, otherwise ignored.
binningOffset	O	0.0 to 1.0 Default=0.0	Yes	No	Phase (offset) specified as fraction of the binning period. Can be used to specify different phase for each subband. When specified binningOffest is added to the phase specified in the element phaseBinning.

Subband parameter	R / O	Range	Same for all stations	Modify	Description
Frame scheduling algorithm	O	dontSet, setDelay, minDelay. Default: minDelay	Yes	Yes	Algorithm used to determine inter-frame delay for transmission of frames generated by the Baseline Boards for frames transmitted via port SPF1. dontSet – do not change inter-frame delay on the Baseline Boards (use previously set values). setDelay – set value specified by the attribute interFrameDelay. If interFrameDelay not specified set 896. minDelay – CM determines the minimum inter-frame delay that can be used based on the number of Baseline Boards that transmit to the same destination. The default for the inter-frame delay algorithm is configurable parameter and can be changed via CM GUI.
Inter-frame delay	O	8-bit number Default:896	Yes	Yes	Inter-frame delay for port SPF1 in microseconds. Relevant only if algorithm is setDelay, otherwise ignored. If randomDelay=ON must be multiple of 4 microseconds. Max.value 1 millisecond (1000 microseconds). Default:896 If randomDelay=OFF Can be multiple of 1 microsecond. Max.value 255 microseconds.
Randomize Delay	O	on, off Default = on	Yes	Yes	Enable/disable randomization of inter-frame delay. Relevant only if algorithm is setDelay, otherwise ignored. If not specified, CM assumes that randomizer should be on. If ON Baseline Board uniformly randomizes the inter-frame delay between transmitted packets, over the range of 0 to 2xIFD (Inter Frame Delay). If ON, inter-frame delay must be multiple of 4 microseconds, maximum value is 1 millisecond. If OFF, no time randomization occurs, inter-frame delay can be specified as multiple of 1 microsecond. Maxumum value is 255 microseconds.
RFI Detection Level	O	0 .0 to 1.0 Default: 1.0	No	Yes	Implemented in the Station Board Filter chip. Each data sample is compared to RFI Detection Level. If data sample is outside ± RFI Detection Level range, RFI detection coumter is incremented and, if RFI Blanking Duration is > 0, data is invalided (for the specified duration). For more details refer to [7]. If not specified, RFI parameters are set as follows: RFI Detection Level = 1.0, i.e. no RFI detection, RFI Blanking Duration= 0.0, i.e. no blanking.
RFI Blanking Duration	O	Decimal number in seconds. Default: 0	No	Yes	

Subband parameter	R/O	Range	Same for all stations	Modify	Description
Polarization Products	O	See section 6.4.9	Yes	No	XML element.
Summed Array	O	See section 6.4.16	Yes	Yes	XML element.
Tone Extraction	O	See section 6.4.17	No	No	XML element. **Not implemented**.
Radar Mode	O	See section 6.4.19	No	No	XML element. **Not implemented**.

6.4.9 Polarization Products

XML element polProducts can be specified as a child of element *subband.*

Polarization products specified in the same instance of the element *polProducts* use the same integration time, recirculation factor, station packing algorithm, product packing algorithm, subset of auto-correlation products and the same set of Baseline Boards. Also, they must have the same number of spectral channels.

Note:

1) **In VCI message number of spectral channels is specified per product.**

2) **Current implementation of CM requires that all products in an instance of the element *polProducts* have the same number of spectral channels.**

Initally, BDF (Binary Data Format) is not able to handle output data where polarization products for the same baseline have different number of spectral channels, therefore CM is not required to handle that cace, on the contrary, CM should enforce the rule that all products for the same baseline must have the same number of spectral channels.

Baseline Boards to be used must be explicitly specified.

Baseline Board allocation is discussed in section 6.4.10.

XML Example 6-12 ***subband / polProducts***

```xml
<polProducts>
    <productIntegration recirculation="1" minIntegTime="250"   ccIntegFactor="1"
                                          ltaIntegFactor="4000"  cbeIntegFactor="10" />
    <pp id="1" correlation="A*A" spectralChannels="64"/>
    <pp id="2" correlation="A*B" spectralChannels="64"/>
    <pp id="3" correlation="B*A" spectralChannels="64"/>
    <pp id="4" correlation="B*B" spectralChannels="64"/>
    <blbPair quadrant="1" firstBlbPair="0" numBlbPairs="1"/>
    <stationPacking algorithm="maxPack"/>
    <productPacking algorithm="maxPack"/>
    <autoCorrSubset algorithm="halfStationsMaxProd" />
</polProducts>
```

Table 6-16 ***subarray / stationInputOutput / baseband / subband / polProducts***

polProducts parameter or child element	R/O	Range	Same for all stations	Description
XML tag	R	polProducts	-	XML tag
autoCorrMode	O	On/off Default=off	Yes	XML attribute. If ON, produces only auto-correlation products. In this mode, each Correlator Chip produces auto-correlation for one input stream – number of lags is fixed and is 2048. In this mode the Correlator Chip produces a half of the lag chain, which means that number of spectral channels is 2048. See 6.4.13.
Polarization Product (pp)	O	See Table 6-17 on page 84.	Yes	XML element, used to specify desired polarization product. Multiple instances of this element can be specified.
Product Integration	R	See Table 6-19 on page 87.	Yes	XML element – defines recirculation factor and integration time. A single instance of this element can be specified.
Station Packing	O	See 6.4.10.6 on page 103. Default=maxPack	Yes	Algorithm for allocation of rows/columns on Baseline Boards. A single instance of this element can be specified.
Product Packing	O	See Table 6-22 on page 106. Default=maxPack	Yes	Agorithm for allocation of the Correlator Chips on Baseline Boards. Relevant only for configurations that require more than one CCC per pol.product. A single instance of this element can be specified.
AutoCorr Subset	O	See Table 6-23 on page 109. Default = halfStationsMaxProd	Yes	A subset of auto-correlation products to be obtained. More often than not, the correlator cannot obtain all the auto-correlation products all the time. This element allows EMCS to choose which subset of auto-correlation products will be produced. A single instance of this element can be specified.
Baseline Board Pair	O	See Table 6-20 on page 96.	Yes	XML element. Baseline Boards to be used to produce specified products. More than one instance of this element can be specified.

Table 6-17 ***subarray / stationInputOutput / baseband / subband / polProducts / pp***

Product parameter	R/O	Range	Must be same for all stations	Description
XML tag	R	pp	-	XML element.
Product ID	R	Positive integer	Yes	Product ID assigned by the originator of the message.
Correlation	R	A*A A*B B*A B*B	Yes	Data streams to be correlated. There can be up to 4 products per subband. For each subband, polarization products must have the same number of spectral channels and different correlation type. CM will reject configuration if this is not true. This restriction is imposed by the Correlator Backend and BDF.
Number of spectral channels	O	64*n, where n is in the range [1..4096] Default: 64 for autocorrelation mode 2048	Yes	Number of spectral channels per product. Number of spectral channels can not be less than 64 or more than 262,144. Number of spectral channels that can be obtained for a particular configuration depends on several factors. For more information see text below.

6.4.9.1 Number of spectral channels

This version of CM requires that all the products specified for the same subband have the same number of spectral channels.

A Correlator Chip Cell (CCC) is the minimum unit that can be assigned to a polarization product. CCC can produce 64 spectral channels for a single polarization product.

Number of spectral channels per product may be increased by:

1. using more CCCs for the same product and/or
2. using recirculation.

Number of spectral channels per product must be an integer multiple of 64.

This topic is discussed in more detail in chapters 6.4.10.9 to 6.4.10.11.

6.4.9.2 Product Integration

Station Board output wafer has 4 wires: two sampled data streams (subband pair), a control stream and a clock.

Control stream generated on the Station Board is often referred to as DUMPTRIG (Dump Trigger), it controls:

1. Recirculation,
2. Integration of correlation products in the Correlator Chip,
3. Dumping of correlation products from the Correlator Chip into Long Term Accumulator (LTA), and
4. Integration in the Long Term Accumulator and dumping/transmission to the Correlator Back End (CBE).

Each Station Board can generate up to 16 different DUMPTRIG streams, which means that each of 16 Station Board output wafers can be assigned different control signal. In other words, each subband can use different recirculation factor and product integration parameters.

For a particular subband, all the stations in a subarray should be assigned the same recirculation factor and integration parameters. In other words, the same subband from all the stations should be accompanied by the identical DUMPTRIG signal. The correlator (CM) rejects configuration where different recirculation factors or integration times are assigned to different stations in the same subarray.

Table 6-19 lists parameters related to product integration.

Fundamental parameter for product integration is the Minimum Hardware Integration Time (MHIT). MHIT must be same for all the products on the same Correlator Chip.

The minimum value for the Minimum Hardware Integration Time is specified in Table 6-18.

The maximum value for the Minimum Hardware Integration Time is 500 microsecond.

Table 6-18 Minimum allowed value for the Minimum Hardware Integration Time

	Dump output of a single CCC per Correlator Chip	***Dump output of more than one CCC per Correlator Chip***
Single CCC dump per LTA frame ***(LTA_integation = 1)***	`2.5 µsec`	`Num.CCCs`[1] `* 2.5 µsec` [1] Number of CCCs being dumped per Correlator Chip.
Multiple CCC dumps per LTA frame ***(LTA_integation > 1)***	`15 µsec`	`200 µsec`

Product integration time is defined as integer multiple of the Minimum Hardware Integration Time. Product integration is performed in three stages:

a) in the Correlator Chip,

b) in the Long Term Accumulator (LTA) and

c) in the Correlator Backend.

User can specify 3 integration factors, as follows:

- Correlator Chip Integration Factor determines Correlator Chip Integration Time, as follows: ccIntegTime = MHIT * ccIntegFactor.

  Except for phase binning, Correlator Chip Integration Time should be:

  - **200 μsec if recirculation is used and**
  - **400 μsec if recirculation is not used.**

- LTA Integration Factor determines LTA Integration Time, as follows: ltaIntegTime= ccIntegTime * ltaIntegFactor.

  The maximum allowed LTA Integration Time is 1 second.

- Backend Integration Factor determines Backend Integration Time, as follows: cbeIntegTime=ltaIntegTime*cbeIntegFactor.

  CM does not enforce limit for the Backend Integration Time.

For more detailed description of the Correlator Chip, LTA and Backend Integration Time please refer to [5], [6] and [1], respectively.

Product integration time for XML Example 6-12 (page 82) can be calculated as follows:

Minimum Hardware Integration Time	250 μsec
Correlator Chip Integration Time	250 μsec * 1 = 250 μsec
LTA Integration Time	250 μsec * 4000 = 1 sec
CBE Integration Time	1 sec * 10 = 10 sec

Total Baseline Board product integration time is limited by the Baseline Board interface which is used to transmit products. Using Gigabit Ethernet, a Baseline Board can transmit cca 11000 frames per second. In the case when there is a single CCC dump per frame, a complete Baseline Board dump consists of 1024 frames (64 Correlator Chips * 16 CCCs), which means that everything can be dumped every 100msec (cca). For more detailed information please refer to the “Programmer’s Guide to EVLA Correlator System Timing, Synchronization, Data Products and Operation” [2].

Table 6-19 ***subarray / stationInputOutput / baseband / subband / polProducts / blbProdIntegration***

parameter	R / O	Range	Same for all stations	Description
XML tag	R	productIntegration	-	XML tag.
Recirculation factor	O	1..256 Default=1	Yes	Dynamic recirculation can be used to get polarization products with more spectral channels. Using the same number of CCCs and recirculation factor of 2 one can double the number of spectal channels. Depending on the data rate, use of recirculation can cause loss of sensitivity. For more information refer to [2] . Recirculation factor=1 means no recirculation.
Phase for recirculation	O	Serial, parallel Default=serial	Yes	Relevant only when recFactor > 1. Serial phase is the default. **Parallel phase can be used for:** stationPacking=onePerRowColumn -autoCorrAlgorithm=allStationsMaxProd , for up to 262144 spectral channels per product. -autoCorrAlgorithm=halfStationsMaxProd, for up to 16384 spectral channels per product. stationPacking=twoPerRowColumn -autoCorrAlgorithm=allStationsMaxProd , for up to 16384 spectral channels per product. **Otherwise, phase for recirculation must be serial.**
Minimum Hardware Integration Time	R	2.5 - 4000	Yes	Minimum hardware integration time (MHIT) in micro seconds. Value must be integer/64MHz.
Correlator Chip Integration Factor	O	Positive Integer Default: 1	Yes	Correlator Chip Integration Time specified as an integer multiple of the Minimum Hardware Integration Time (MHIT).
LTA Integration Factor	O	Positive Integer Default: 1	Yes	LTA Integration Time specified as an integer multiple of the Hardware Integration Time. According to [2] the LTA Integration Time should be 1second. The upper limit on the integration time is determined by the 32-bit data valid counter, and is approximately 16.7 seconds. For more detailed explanation refer to [2].
Backend Integration Factor	O	Positive Integer Default: 1	Yes	Backend Integration Time specified as an integer multiple of LTA Integration Time. Range TBD.
Burst Mode	O		Yes	Burst mode parameters in Table 6-26 on pg. 162.

6.4.10 Baseline Board Configuration for Polarization Products

User must explicitly specify Baseline Boards to be used for each subband and group of products. The decision to let user choose Baseline Boards was made to avoid boards with known or suspected problems during testing and installation. This decision is not irreversible, but change would require additional work on CM software.

In the EVLA correlator, Baseline Boards are arranged as follows:

- There are two crates in the Baseline Rack and 8 slots in each crate.
- Crates are assigned IDs 0 (top) and 1 (bottom).
- Slots are assigned IDs in the range 0 to 7.[1]

Baseline Board can be identified using Module Locator Identifier (MLID).

MLID is defined as *rack-crate-slot, e.g . s001-t-0, b101-t-0.*

Station Boards are connected to Baseline Boards via Cross-bar Boards; connections among the boards are fixed. Switches implemented in the Cross-bar Board are used to forward Station Board output to different Baseline Boards.

Connections between Station Boards, Cross-bar Boards and Baseline Boards, organize Baseline Boards in four Quadrants.

1) There are four Baseline Quadrants in the EVLA correlator.
2) Each Quadrant consists of 16 Baseline Board Pairs.
3) Baseline Boards in adjacent slots are connected as pairs. Input data from BlB in slot 1 is forwarded to BlB in slot 0, and so on. Figure 6-2 is a conceptual diagram of the Baseline Board Rack.

Baseline Board in the EVLA correlator can be identified using Quadrant ID, BlB Pair ID and position in the BlB pair: *quadrant-blbPair-position.*

Range:

```
quadrant [1..4]
blbPair  [0..15]
position [0,1].
```

When switches in the Cross-bar Boards are in the default position (straight-through), a **Baseline Board Pair *n***, where ***n*** is an integer in the range [0..15], receives input from the **Station Board Output Wafer *n***, as follows:

[1] Numbers adjacent to the Baseline Board input cables are Station IDs provided to illustrate that each Baseline Board receives input for up to 32 stations.

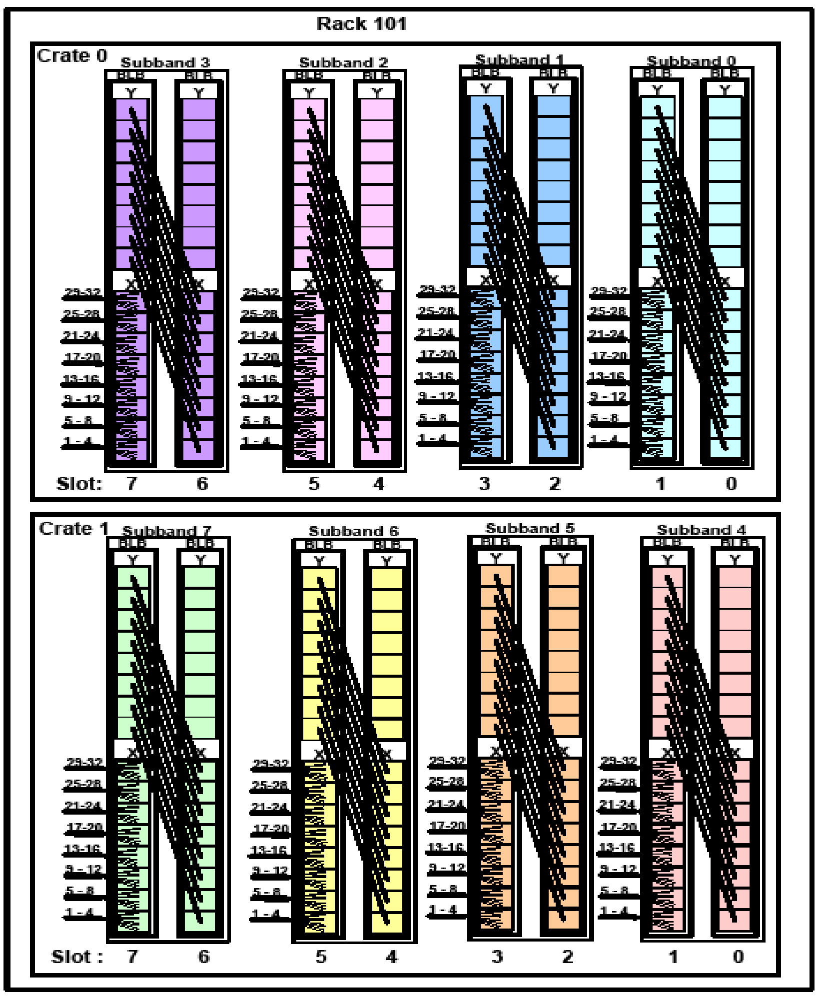
1. BlB pair **Q1/Pn** from Stations Boards in **slots t-0 to t-3**
2. BlB pair **Q2/Pn** from Stations Boards in **slots t-4 to t-7**
3. BlB pair **Q3/Pn** from Stations Boards in **slots b-0 to b-3**
4. BlB pair **Q4/Pn** from Stations Boards in **slots b-3 to b-7**

This applies for all Station Racks.

In the "nominal" or "default" configuration, each BlB Pair gets different subband:

1. Quadrant 1 BlB Pair 0 receives Subband 0 for Basebands 0/1,
2. Quadrant 1 BlB Pair 1 receives Subband 1 for Basebands 0/1,
3. Quadrant 1 BlB Pair 2 receives Subband 2 for Basebands 0/1, etc.

Figure 6-2 Baseline Rack



6.4.10.1 4-Bit Correlation

Figure 6-3 is a conceptual diagram of the EVLA correlator configuration for the EVLA antennas with 3-bit initial quantization and 4-bit correlation:

- There are 32 stations.
- Each station has 4 Baseband Pairs.
- Each station has 4 Station Boards, one Station Board per BB pair.
- Each Station Board produces 16 subbands per Baseband Pair; those 16 subbands are forwarded to the Baseline Boards that belong to the same Quadrant.

The Station IDs, Baseband IDs and Subband IDs shown in the diagram are typical or nominal configuration. Station IDs, Baseband IDs and Subband IDs are assigned by the user; the correlator does not impose any restrictions, except for the following:

Appendix A Station ID must be unique in the correlator, i.e. there cannot be two or more stations with the same ID,

Appendix B Combination of Baseband ID/Subband ID must be unique for a station.

For more information regarding Subband ID refer to section 6.4.8.

Figure 6-4 shows configuration for a Baseline Board Pair from Figure 6-3. Station IDs, Baseband IDs and Subband IDs do not need to be assigned as shown in the diagram; this is just a “nominal” configuration.

Figure 6-3 EVLA Correlator configured for 32 Antennas (Stations), 3-Bit Initial Quantization (sampling), 4 EVLA polarization pairs, 4 WIDAR baseband pairs, 16 subbands per baseband pair, 4-bit re-quantization (correlation). There is one Baseline Board Pair per Subband; it can be used to produce one of the following: 4 Polarization Products @ 64 Spectral Channels per product, or 2 Plarization Products @ 128 spectral channels per product, or 1 Polarization Product @ 256 spectral channels. Recirculation can be used to get more spectral channels.

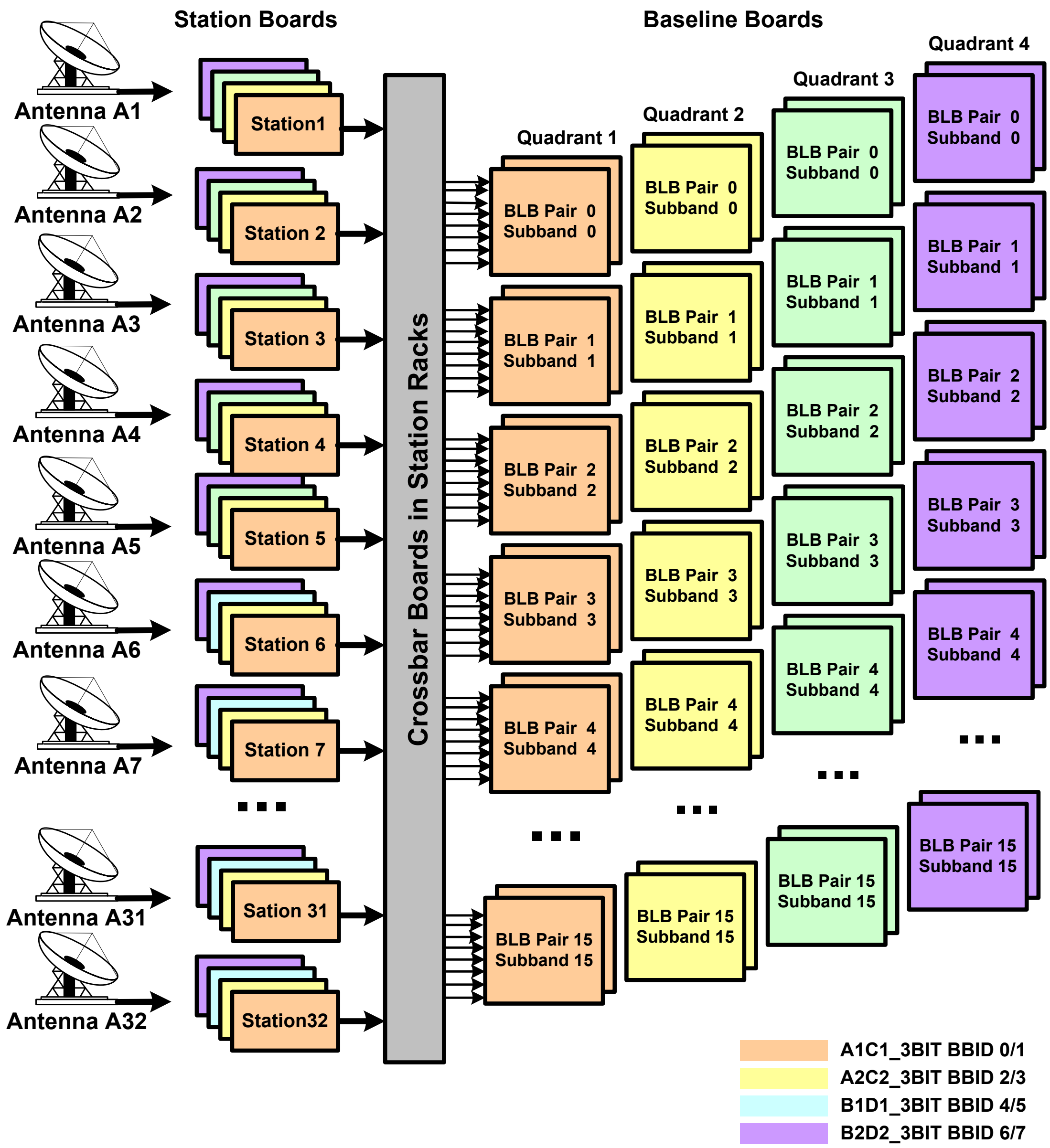


Figure 6-4 Baseline Board Pair - CorrChip array configured for stationPacking=maxPack, autoCorrAlgorithm=halfStationsMaxProd, starting from the station with the lowest SID. There are 4 baselines per CorrChip. Each Baseline can use 4 CCCs (one CCQ).

One subarray, 32 stations

2 Baseline Boards can produce cross-correlation products for all 32 stations and auto-correlation products for 16 of those 32 stations.

1st Baseline Board

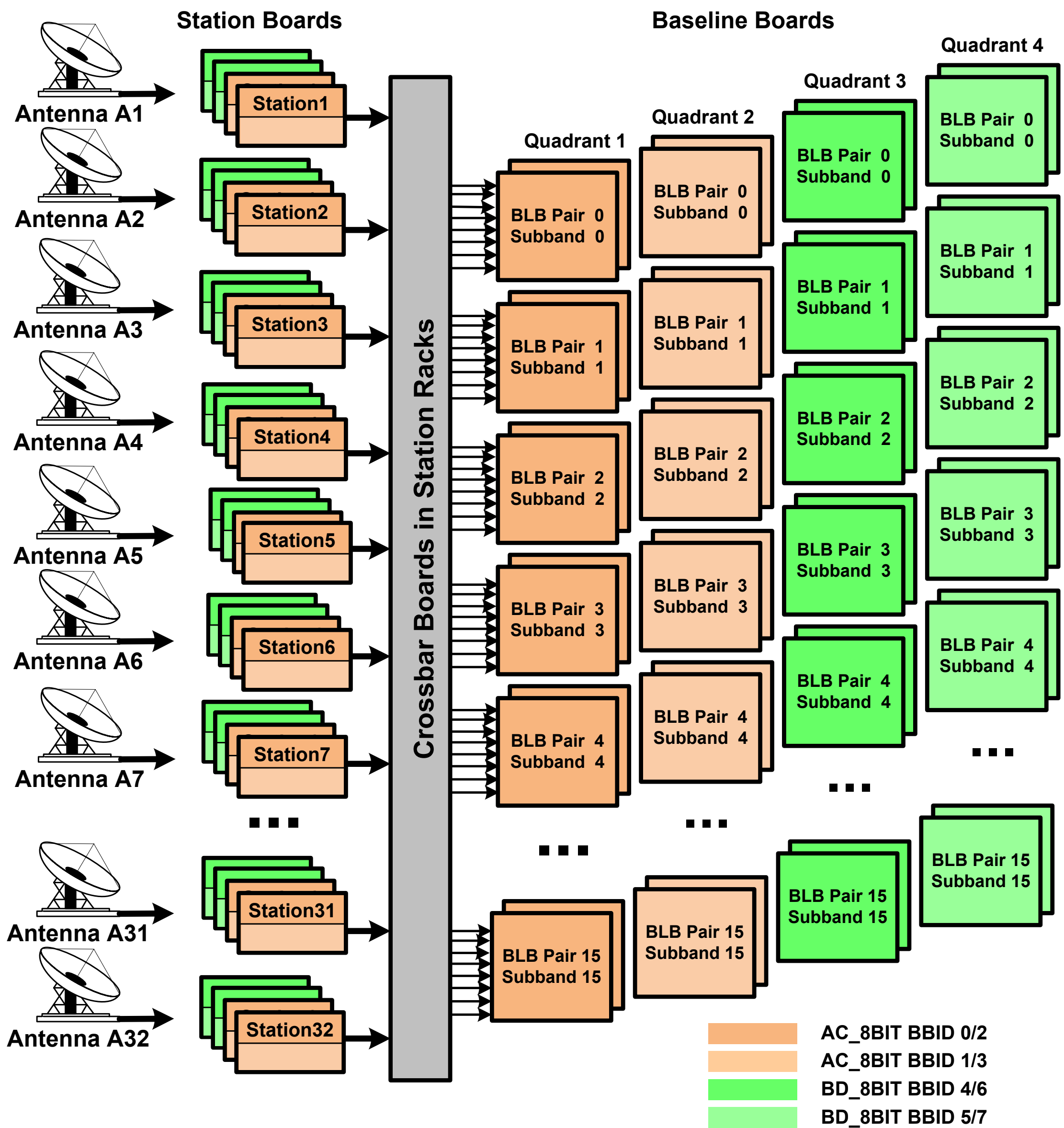
	29 30 31 32 X7	25 26 27 28 X6	21 22 23 24 X5	17 18 19 20 X4	13 14 15 16 X3	9 10 11 12 X2	5 6 7 8 X1	1 2 3 4 X0
30 29 Y7 32 21	29 * 29 29 * 30 29 * 31 29 * 32	25 * 29 25 * 30 25 * 31 25 * 32	21 * 29 21 * 30 21 * 31 21 * 32	17 * 29 17 * 30 17 * 31 17 * 32	13 * 29 13 * 30 13 * 31 13 * 32	9 * 29 9 * 30 9 * 31 9 * 32	5 * 29 5 * 30 5 * 31 5 * 32	1 * 29 1 * 30 1 * 31 1 * 32
26 25 Y6 28 27	26 * 29 26 * 30 26 * 31 26 * 32	25 * 25 25 * 26 25 * 27 25 * 28	21 * 25 21 * 26 21 * 27 21 * 28	17 * 25 17 * 26 17 * 27 17 * 28	13 * 25 13 * 26 13 * 27 13 * 28	9 * 25 9 * 26 9 * 27 9 * 28	5 * 25 5 * 26 5 * 27 5 * 28	2 * 25 1 * 26 1 * 27 1 * 28
22 21 Y5 24 23	22 * 29 22 * 30 22 * 31 22 * 32	22 * 25 22 * 26 22 * 27 22 * 28	21 * 21 21 * 22 21 * 23 21 * 24	17 * 21 17 * 22 17 * 23 17 * 24	13 * 21 13 * 22 13 * 23 13 * 24	9 * 21 9 * 22 9 * 23 9 * 24	5 * 21 5 * 22 5 * 23 5 * 24	1 * 21 1 * 22 1 * 23 1 * 24
18 17 Y4 20 19	18 * 29 18 * 30 18 * 31 18 * 32	18 * 25 18 * 26 18 * 27 18 * 28	18 * 21 18 * 22 18 * 23 18 * 24	17 * 17 17 * 18 17 * 19 17 * 20	13 * 17 13 * 18 13 * 19 13 * 20	9 * 17 9 * 18 9 * 19 9 * 20	5 * 17 5 * 18 5 * 19 5 * 20	1 * 17 1 * 18 1 * 19 1 * 20
14 13 Y3 16 15	13 * 29 13 * 30 13 * 31 13 * 32	14 * 25 14 * 26 14 * 27 14 * 28	14 * 21 14 * 22 14 * 23 14 * 24	14 * 17 14 * 18 14 * 19 14 * 20	13 * 13 13 * 14 13 * 15 13 * 16	9 * 13 9 * 14 9 * 15 9 * 16	5 * 13 5 * 14 5 * 15 5 * 16	1 * 13 1 * 14 1 * 15 1 * 16
10 9 Y2 12 11	10 * 29 10 * 30 10 * 31 10 * 32	10 * 25 10 * 26 10 * 27 10 * 28	10 * 21 10 * 22 10 * 23 10 * 24	10 * 17 10 * 18 10 * 19 10 * 20	10 * 13 10 * 14 10 * 15 10 * 16	9 * 9 9 * 10 9 * 11 9 * 12	5 * 9 5 * 10 5 * 11 5 * 12	1 * 9 1 * 10 1 * 11 1 * 12
6 5 Y1 8 7	6 * 29 6 * 30 6 * 31 6 * 32	6 * 25 6 * 26 6 * 27 6 * 28	6 * 21 6 * 22 6 * 23 6 * 24	6 * 17 6 * 18 6 * 19 6 * 20	6 * 13 6 * 14 6 * 15 6 * 16	6 * 9 6 * 10 6 * 11 6 * 12	5 * 5 5 * 6 5 * 7 5 * 8	1 * 5 1 * 6 1 * 7 1 * 8
2 1 Y0 4 3	2 * 29 2 * 30 2 * 31 2 * 32	2 * 25 2 * 26 2 * 27 2 * 28	2 * 21 2 * 22 2 * 23 2 * 24	2 * 17 2 * 18 2 * 19 2 * 20	2 * 13 2 * 14 2 * 15 2 * 16	2 * 9 2 * 10 2 * 11 2 * 12	2 * 5 2 * 6 2 * 7 2 * 8	1 * 1 1 * 2 1 * 3 1 * 4

2nd Baseline Board

31 32 29 30 X7	27 28 25 26 X6	23 24 21 22 X5	19 20 17 18 X4	15 16 13 14 X3	11 12 9 10 X2	7 8 5 6 X1	3 4 1 2 X0	
31 * 30 31 * 31 31 * 32 32 * 30	27 * 29 27 * 30 27 * 31 27 * 32	23 * 29 23 * 30 23 * 31 23 * 32	19 * 29 19 * 30 19 * 31 19 * 32	15 * 29 15 * 30 15 * 31 15 * 32	11 * 29 11 * 30 11 * 31 11 * 32	7 * 29 7 * 30 7 * 31 7 * 32	3 * 29 3 * 30 3 * 31 3 * 32	32 31 30 29
28 * 29 28 * 30 28 * 31 28 * 32	27 * 26 27 * 27 27 * 28 28 * 26	23 * 25 23 * 26 23 * 27 23 * 28	19 * 25 19 * 26 19 * 27 19 * 28	15 * 25 15 * 26 15 * 27 15 * 28	11 * 25 11 * 26 11 * 27 11 * 28	7 * 25 7 * 26 7 * 27 7 * 28	3 * 25 3 * 26 3 * 27 3 * 28	28 27 26 25
24 * 29 24 * 30 24 * 31 24 * 32	24 * 25 24 * 26 24 * 27 24 * 28	23 * 22 23 * 23 23 * 24 24 * 22	19 * 21 19 * 22 19 * 23 19 * 24	15 * 21 15 * 22 15 * 23 15 * 24	11 * 21 11 * 22 11 * 23 11 * 24	7 * 21 7 * 22 7 * 23 7 * 24	3 * 21 3 * 22 3 * 23 3 * 24	24 23 22 21
20 * 29 20 * 30 20 * 31 20 * 32	20 * 25 20 * 26 20 * 27 20 * 28	20 * 21 20 * 22 20 * 23 20 * 24	19 * 18 19 * 19 19 * 20 20 * 18	15 * 17 15 * 18 15 * 19 15 * 20	11 * 17 11 * 18 11 * 19 11 * 20	7 * 17 7 * 18 7 * 19 7 * 20	3 * 17 3 * 18 3 * 19 3 * 20	20 19 18 17
16 * 29 16 * 30 16 * 31 16 * 32	16 * 25 16 * 26 16 * 27 16 * 28	16 * 21 16 * 22 16 * 23 16 * 24	16 * 17 16 * 18 16 * 19 16 * 20	15 * 14 15 * 15 15 * 16 16 * 14	11 * 13 11 * 14 11 * 15 11 * 16	7 * 13 7 * 14 7 * 15 7 * 16	3 * 13 3 * 14 3 * 15 3 * 15	16 15 14 13
12 * 29 12 * 30 12 * 31 12 * 32	12 * 25 12 * 26 12 * 27 12 * 28	12 * 21 12 * 22 12 * 23 12 * 24	12 * 17 12 * 18 12 * 19 12 * 20	12 * 13 12 * 14 12 * 15 12 * 16	11 * 10 11 * 11 11 * 12 12 * 10	7 * 9 7 * 10 7 * 11 7 * 12	3 * 9 3 * 10 3 * 11 3 * 12	12 11 10 9
8 * 29 8 * 30 8 * 31 8 * 32	8 * 25 8 * 26 8 * 27 8 * 28	8 * 21 8 * 22 8 * 23 8 * 24	8 * 17 8 * 18 8 * 19 8 * 20	8 * 13 8 * 14 8 * 15 8 * 16	8 * 9 8 * 10 8 * 11 8 * 12	7 * 6 7 * 7 7 * 8 8 * 6	3 * 5 3 * 6 3 * 7 3 * 8	8 7 6 5
4 * 29 4 * 30 4 * 31 4 * 32	4 * 25 4 * 26 4 * 27 4 * 28	4 * 21 4 * 22 4 * 23 4 * 24	4 * 17 4 * 18 4 * 19 4 * 20	4 * 13 4 * 14 4 * 15 4 * 16	4 * 9 4 * 10 4 * 11 4 * 12	4 * 5 4 * 6 4 * 7 4 * 8	3 * 2 3 * 3 3 * 4 4 * 2	4 3 2 1

Correlator Chip on the diagonal: receive X and Y input from the same group of 4 stations

Figure 6-5 Correlator configured for 32 Antennas (Stations), 8-Bit Initial Quantization (sampling), 2 EVLA polarization pairs, 4 WIDAR baseband pairs, 16 subbands per baseband pair, 4-bit re-quantization (correlation). There is one Baseline Board Pair per subband. This is example of a configuration where each Baseline Board Pair is used for different subband.



6.4.10.2 7-Bit Correlation

Figure 6-6 is a conceptual diagram of the EVLA correlator configured for 8-bit initial quantization (sampling) and 7-bit re-quantization and correlation:

- There are 2 EVLA Baseband Pairs per station,
- 2 EVLA polarization pairs can be configured as 4 WIDAR Baseband Pairs.
- Each EVLA Baseband Pair uses two Station Boards, one Station Board for each polarization.
- 4 Baseline Board Pairs are needed to get 7-bit correlation products for up to 32 stations in stationPacking=maxPack mode.
- All the Baseline Boards (4 Quadrants) are used to get 7-bit correlation products for a single Baseband Pair (16 subbands).

Initial quantization (3-bit or 8-bit) and re-qunatization (4-bit or 7-bit) can be selected independently from each other:

1. Initial quantization is specified per baseband pair and describes input received from antenna.
2. Re-quantization is selected in the Station Board filter and can be independently selected for each subband of each baseband.

In the Baseline Board Recirculation Controller each 7-bit data stream received from Station Board is devided into two 4-bit data streams, referred to as: LSN – Least Significant Nibble, MSN – Most Significant Nibble. LSN stream carries 4 least significant bits of 7-bit sample; MSN stream carries 3 most significant bits of 7-bit sample.

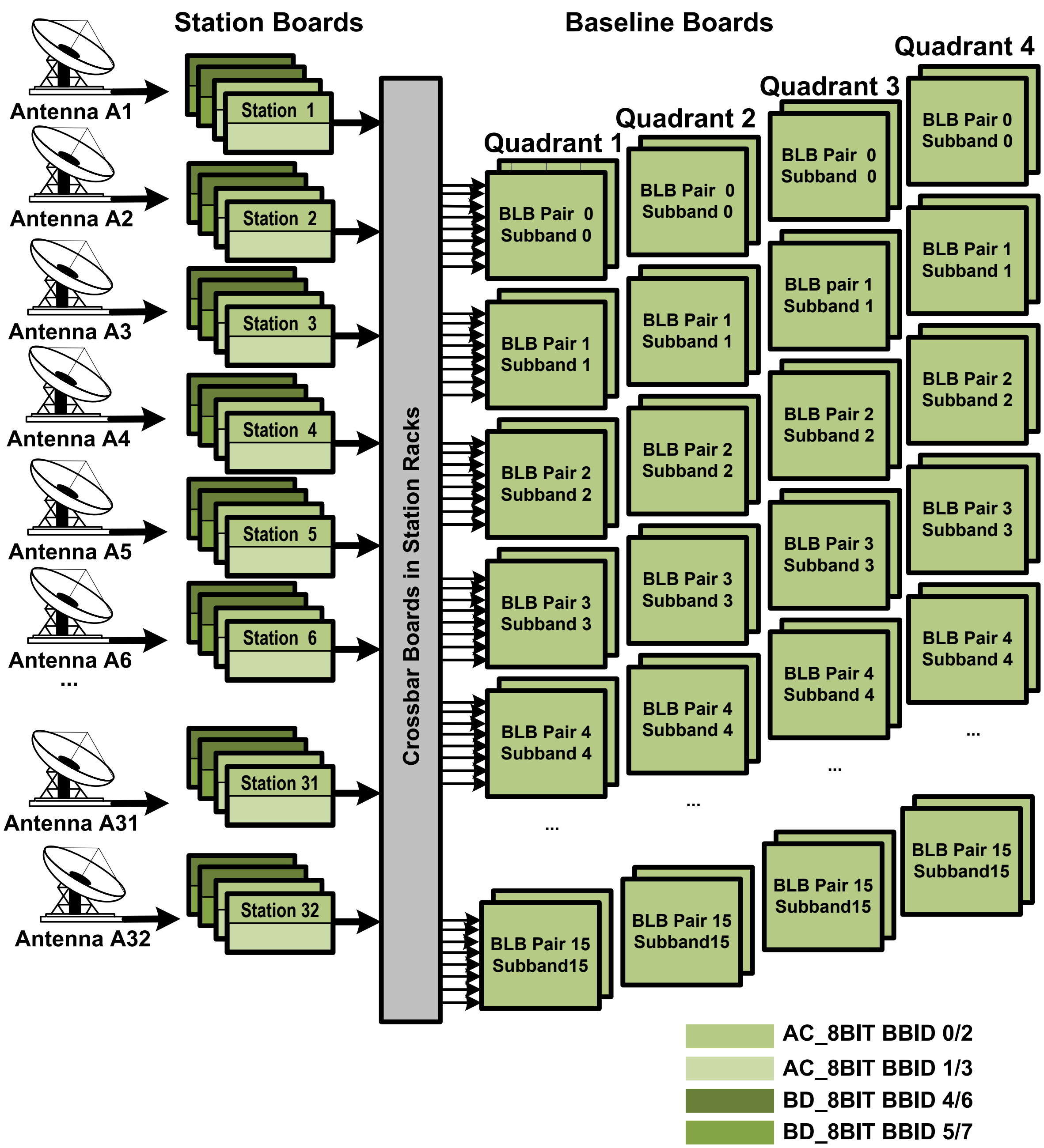
Each For 7-bit correlation each polarization product is configured as 4 sub-products:

1. LSN*LSN,
2. LSN*MSN,
3. MSN*LSN,
4. MSN*MSN.

7-bit polarization products are configured in the same manner as 4-bit products (see Figure 6-4), only, in this case, each Baseline Board Pair is configured to produce one sub-product (LSN*LSN, LSN*MSN, MSN*LSN or MSN*MSN).

Number of Baseline Boards needed to get plarization products is calculated in the same manner as for 4-bit correlation and multiplied by 4.

Figure 6-6 EVLA Correlator configured for 32 Antennas (Stations), 8-Bit Initial Quantization (sampling), 2 EVLA polarization pairs, 4 WIDAR baseband pairs, 16 Subbands, 7-Bit Re-Quantization. 7-bit correlation uses 4 BlB Pairs per subband. In this example all Baseline Boards are used to get pol.products for a single baseband pair (16 subbands). Four BlB Pairs can produce: 4pp @ 64 Spectral Channels or 2pp @ 128 spectral channels or 1pp @ 256 spectral channels. Recirculation can be used to get more spectral channels.



6.4.10.3 Baseline Board Pair

Table 6-20 ***subarray / stationInputOutput / baseband / subband / polProducts / blbPair***

subband parameter	R/O	Range	Must be same for all stations	Description
XML Tag	R	blbPair	n/a	XML tag. This XML element is used to specify Baseline Board Pair(s) to be used for a specific subband.
quadrant	R	1..4	Yes	Quadrant ID. Identifier of the correlator Quadrant.
firstBlbPair	R	0..15	Yes	Index of the first Baseline Board Pair to be used.
numBlbPairs	R	1..16	Yes	Number of consecutive Baseline Board Pairs to be used. In the EVLA correlator each Quadrant has 16 Baseline Board Pairs.

In general any subband can be forwarded to any Baseline Board Pair, with the following restrictions:

1. For a baseband pair each subband must be correlated on a Baseline Board Pair with different BlB Pair ID. For example, for BB Pair 0/1, if subband 0 uses BlB pair Q1P10, BlB Pairs Q2P10, Q3P10 or Q4P10 cannot be used to correlate other subbands of the same baseband pair[1].

2. Two Baseline Boards connected as a pair receive the same input (up to 32 subband pairs), which means that a Baseline Board Pair can not get more than 32 different subband pairs. Those subbands may or may not belong to the same subarray.

---

[1] This limitation is due to the fact that each Station Board output wafer is connected to four Baseline Board Pairs. Those 4 BlB Pairs have the same Pair ID but belong to different Quadrants. For example, Output Wafer 0, on all the Station Boards, is connected to Baseline Board Pairs: Q1P0, Q2P0, Q3P0 and Q4P0.

Additional limitation applies if a Baseline Board Pair is used to get products for more than one subarray:

- Subarrays can not share rows and columns on the Baseline Board.

Rationale: It is a hardware limitation that all the products on the same Correlator Chip must use the same MHIT (see section 6.4.9.2). Different subarrays may use different Minimum Hardware Integration Time – MHIT (and generally different DUMPTRIG). In order to simplify CM and observation preparation software, it has been decided that subarrays should not share rows/columns on the Baseline Boards.

For stationPacking algoritm=maxPack (4 stations per row/column), when calculating number of BlB rows/columns required for a subarray, number of antennas (stations) is rounded up to a multiple of 4. For example, a subarray that consists of 9 antennas would use 3 rows/columns on the Baseline Board.

numRowsColumns=(numStations/4)

if ((numStations Modulo 4) > 0) numRowsColumns = numRowsColumns+1;

XML Example 6-13 shows Baseline Board Pair allocation for 5 subbands: subband 0 needs only one BlB Pair, subband 1 needs two BlB Pairs, subbands 2, 3, 4 and 5 need 4 BlB Pairs per subband.

XML Example 6-13 Baseline Board Pairs

```
<widar:baseBand bbA="0" bbB="2" swbbName="AC_8BIT" bw="1024000000" inQuant="8"  >

    <widar:subBand sbid="0" swIndex="1"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
        <widar:polProducts>
            <widar:blbProdIntegration recirculation="1" minIntegTime="210.0" ltaIntegFactor="4000"
                                     ccIntegFactor="1" />
            <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
            <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
            <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
            <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
            <widar:blbPair quadrant="1" firstBlbPair="0" numBlbPairs="1" />
        </widar:polProducts>
    </widar:subBand>

    <widar:subBand sbid="1" swIndex="2" bw="128000000" centralFreq="256000000" rqNumBits="4" >
        <widar:polProducts>
            <widar:blbProdIntegration recirculation="1" minIntegTime="210.0" ltaIntegFactor="4000"
                                     ccIntegFactor="1" />
            <widar:pp spectralChannels="128" id="1" correlation="A*A"/>
            <widar:pp spectralChannels="128" id="2" correlation="A*B"/>
            <widar:pp spectralChannels="128" id="3" correlation="B*A"/>
            <widar:pp spectralChannels="128" id="4" correlation="B*B"/>
            <widar:blbPair quadrant="1" firstBlbPair="2" numBlbPairs="1" />
```

```
            <widar:blbPair quadrant="2" firstBlbPair="2" numBlbPairs="1" />
        </widar:polProducts>
    </widar:subBand>

    <widar:subBand sbid="2" swIndex="3"  bw="128000000" centralFreq="256000000" rqNumBits="4" >
        <widar:polProducts>
            <widar:blbProdIntegration recirculation="1" minIntegTime="210.0" ltaIntegFactor="4000"
                                      ccIntegFactor="1" />
            <widar:pp spectralChannels="256" id="1" correlation="A*A"/>
            <widar:pp spectralChannels="256" id="2" correlation="A*B"/>
            <widar:pp spectralChannels="256" id="3" correlation="B*A"/>
            <widar:pp spectralChannels="256" id="4" correlation="B*B"/>
            <widar:blbPair quadrant="1" firstBlbPair="4"   numBlbPairs="1" />
            <widar:blbPair quadrant="2" firstBlbPair="4"   numBlbPairs="1" />
            <widar:blbPair quadrant="3" firstBlbPair="4"   numBlbPairs="1" />
            <widar:blbPair quadrant="4" firstBlbPair="4"   numBlbPairs="1" />
        </widar:polProducts>
    </widar:subBand>

    <widar:subBand sbid="3" swIndex="4"  bw="128000000" centralFreq="192000000" rqNumBits="4" >
        <widar:polProducts>
            <widar:blbProdIntegration recirculation="1" minIntegTime="210.0" ltaIntegFactor="4000"
                                      ccIntegFactor="1" />
            <widar:pp spectralChannels="256" id="1" correlation="A*A"/>
            <widar:pp spectralChannels="256" id="2" correlation="A*B"/>
            <widar:pp spectralChannels="256" id="3" correlation="B*A"/>
            <widar:pp spectralChannels="256" id="4" correlation="B*B"/>
            <widar:blbPair quadrant="1" firstBlbPair="5" numBlbPairs="4" />
        </widar:polProducts>
    </widar:subBand>

    <widar:subBand sbid="4"  swIndex="5" bw="128000000" centralFreq="256000000" rqNumBits="4" >
        <widar:polProducts>
            <widar:blbProdIntegration recirculation="1" minIntegTime="210.0" ltaIntegFactor="4000"
                                      ccIntegFactor="1" />
            <widar:pp spectralChannels="256" id="1" correlation="A*A"/>
            <widar:pp spectralChannels="256" id="2" correlation="A*B"/>
            <widar:pp spectralChannels="256" id="3" correlation="B*A"/>
            <widar:pp spectralChannels="256" id="4" correlation="B*B"/>
            <widar:blbPair quadrant="1" firstBlbPair="10" numBlbPairs="2" />
            <widar:blbPair quadrant="2" firstBlbPair="10" numBlbPairs="2" />
        </widar:polProducts>
    </widar:subBand>

</widar:baseBand>
```

6.4.10.4 Baseline Board Singleton

All the examples for the assignment of Baseline Boards in section 6.4.10 discuss cases when Baseline Boards are assigned and used as pairs.

In some cases, a single Baseline Board may be able to produce all the products for one subband.

- For auto-correlation mode.
- If a single product is required for stationPacking=minPack, BlB stacking is not used, and the subarray consists of a small number of antennas; a single Baseline Board may be able to produce all the products.
- In some cases, stationPacking=minPack uses odd number of Baseline Boards, leaving one of the boards unused.

However, these cases are rarely used and auto-correlation mode is not fully implemented. Using each Baseline Board in a pair for a different subband is possible, but there would be a number of restrictions.

At this time Baseline Boards must be assigned as pairs.

6.4.10.5 Baseline Board Allocation - Example

Figure 6-7 is a resource allocation grid that shows allocation of Baseline Boards for Basebands/Subbands. Here is how the grid could be used:

- By placing a black dot in a specific square user selects Subband ID;
- Each square represents a Baseline Board;
- Dotted line separates Baseline Boards that are connected as a pair.
- Color is used to show which Baseline Boards belong to which subband;
- The number of channels is the total available per subband pair;
- The number of channels per product is is not specified in this diagram.

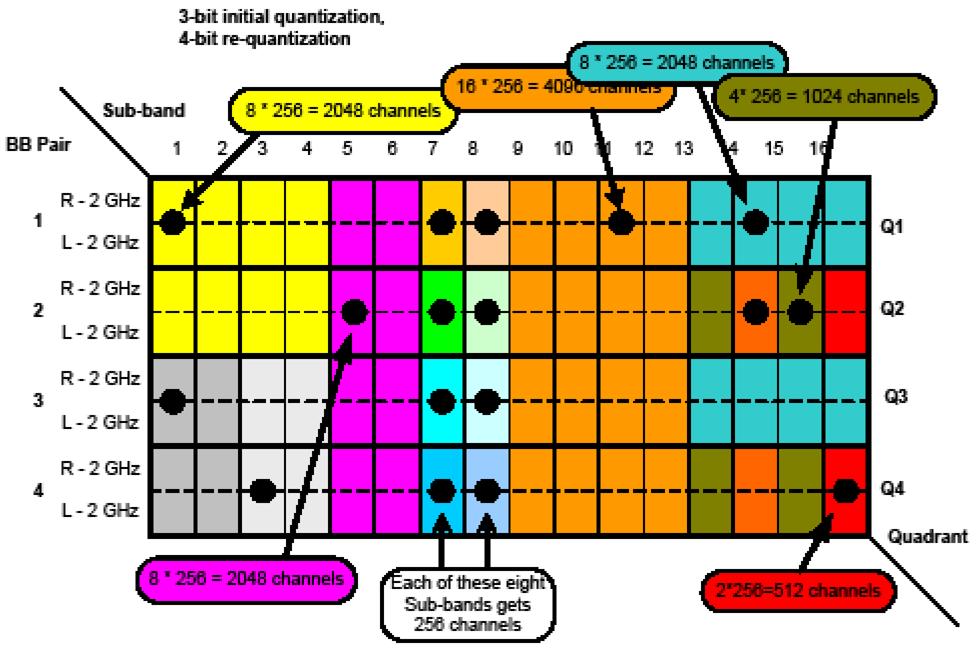
Figure 6-8 shows allocation of the Station Boards and Baseline Boards for the basebands/subbands specified in Figure 6-7. Note the differences in notation used in the resource allocation grid and by the WIDAR correlator:

- Resource allocation grid uses Basebands Pair (BlBPs) identifiers 1, 2, 3 and 4, while WIDAR correlator uses Baseband Identifiers (BBID)s in the range [0..7].
- In the resource allocation grid, Subband IDs start from 1, while in the WIDAR correlator Subbands IDs start from 0.

Figure 6-8 shows 4 Station Boards that belong to the same station, i.e. receive input from the same antenna. Typically, all the stations in a subarray are configured in the same way. As shown in the diagram, there are two banks of 18 filters on each Station Board. Typically, the correlator software chooses which filter is used for which subband, although VCI protocol allows user to explicitly select the filter.

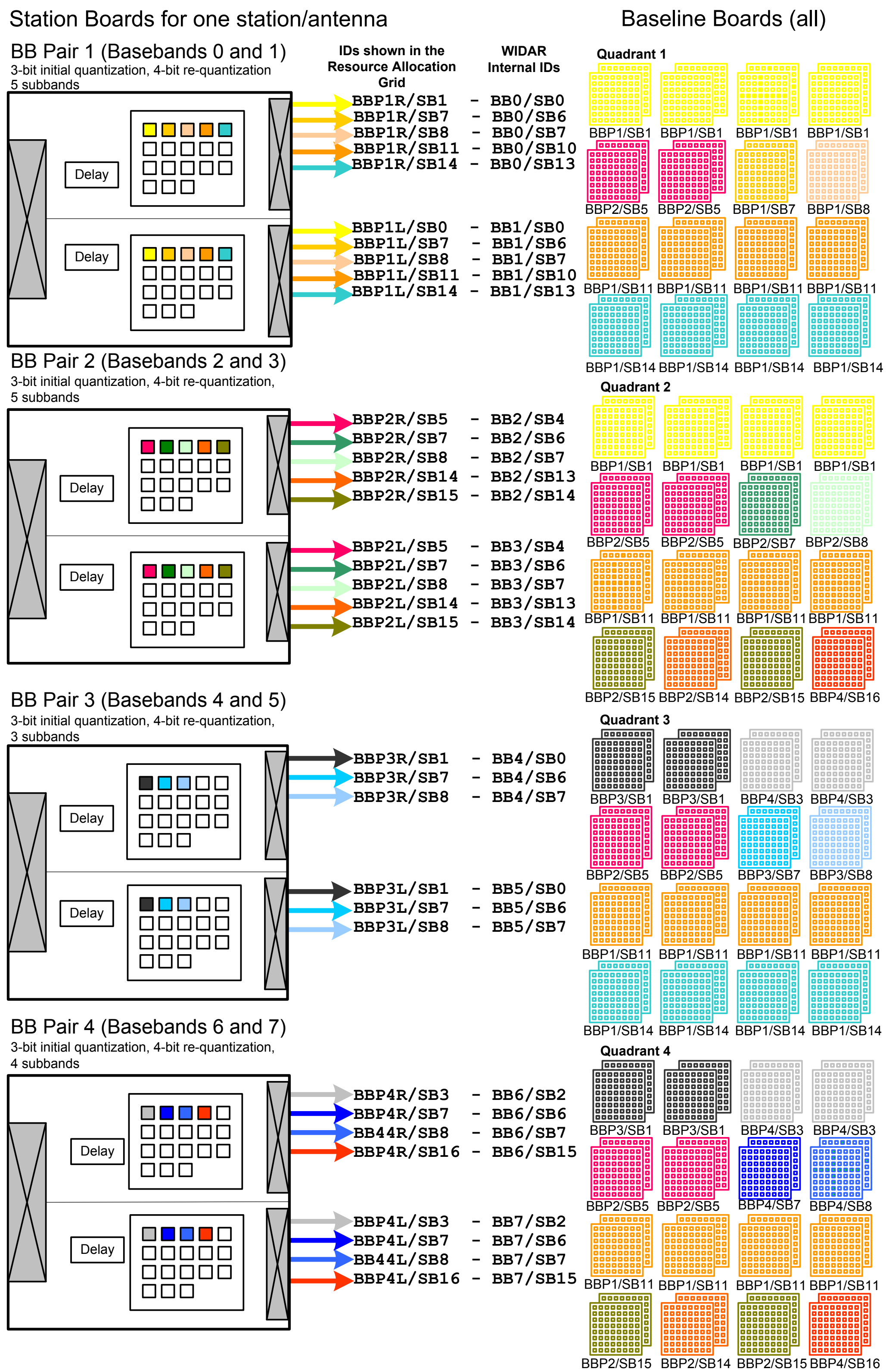
XML Example 16-1 on page 215 is an example of the XML configuration for subbands from Figure 6-7. Polarization products as specified in XML Example 16-1 are just an example, other combinations are possibile.

Figure 6-7 Resource Allocation Grid – an Example for 3-bit IQ / 4-bit RQ



Each black dot represents a particular combination of frequency and bandwidth.

Figure 6-8 Station Board and Baseline Board allocation for Figure 6-7



6.4.10.6 Station Packing Algorithm

Using XML element “stationPacking” user can choose one of the following:

1. fourPerRowColumn (maxPack) : In this mode each Baseline Board row/column gets input from 4 antennas, which is the maximum possible number of stations per row/column. This mode is used in order to minimize number of used Baseline Boards (and rows/columns per Baseline Board), leaving resources to be used by other subbands and subarrays.

2. twoPerRowColumn (midPack): In this mode each Baseline Board row/column gets input from 2 antennas. In this mode each Correlator Chip is used for different baseline. It is possible to get up to 1024 spectral channels per baseline without recirculation and Baseline Board Stacking. This mode can be used to get more spectral channels or to reduce number of used Correlator Chip Cells per Baseline Board (for faster dumping).

3. onePerRowColumn (minPack): In this mode each Baseline Board row/colum gets input from 1 antenna. Each Correlator Chip is used for different polarization product. It is possible to get up to 1024 spectral channels per product without recirculation and Baseline Board Stacking.This mode can be used to get more spectral channels or to reduce number of used Correlator Chip Cells per Baseline Board (to allow for faster dumping).

Figure 6-9 on page 105 shows Baseline Board Correlator Chip Array configured for station packing algorithm maxPack (fourPerRowColumn) and minPack (onePerRowColumn) for a subarray that consists of 8 antennas.

XML Example **6-14** and XML Example 6-15 show corresponding VCI XML configuration messages. As shown in Figure 6-9, this parameter can cause a significant change in Baseline Board configuration:

- stationPacking=maxPack (fourPerRowColumn) uses only two rows/columns on the Baseline Boards, leaving remaining 6 rows/columns to be used for other subarray(s). This configuration can produce up to 256 spectral channels per baseline (without recirculation).
- stationPacking=minPack (onePerRowColumn) uses all 8 rows/columns on the Baseline Boards, but has 16 times more lags per baseline. It is possible to get up to 4096 spectral channels per baseline (up to 1024 spectral channels per pol. product).

Figure 6-4 on page 92 shows a pair of Baseline Boards configured for ‘maxPack’ mode. Baseline Boards are configured for a single subarray that has 32 antennas (stations). Each row/column on the Baseline Board gets input from 4 antennas and each Correlator Chip produces products for 4 baselines.

Detailed description for all station packing modes is provided in chapters 6.4.10.9, 6.4.10.10 and 6.4.10.11.

Table 6-21 ***subarray / stationInputOutput / baseband / subband / polProducts / stationPacking***

subband parameter	R/O	Range	Must be same for all stations	Description
XML tag	R	stationPacking	n/a	stationPacking is an XML element, it is used to choose the algorithm used to configure Baseline Boards.
algorithm	O	fourPerRowColumn maxPack twoPerRowColumn midPack onePerRowColumn minPack Default=fourPerRowColumn (maxPack)	Yes	Algorithm for allocation of Baseline Board resources (rows and columns on the Baseline Board). Default in this case means: value assumed when element *stationPacking* is not specified. maxPack = fourPerRowColumn midPack = twoPerRowColumn minPack = onePerRowColumn

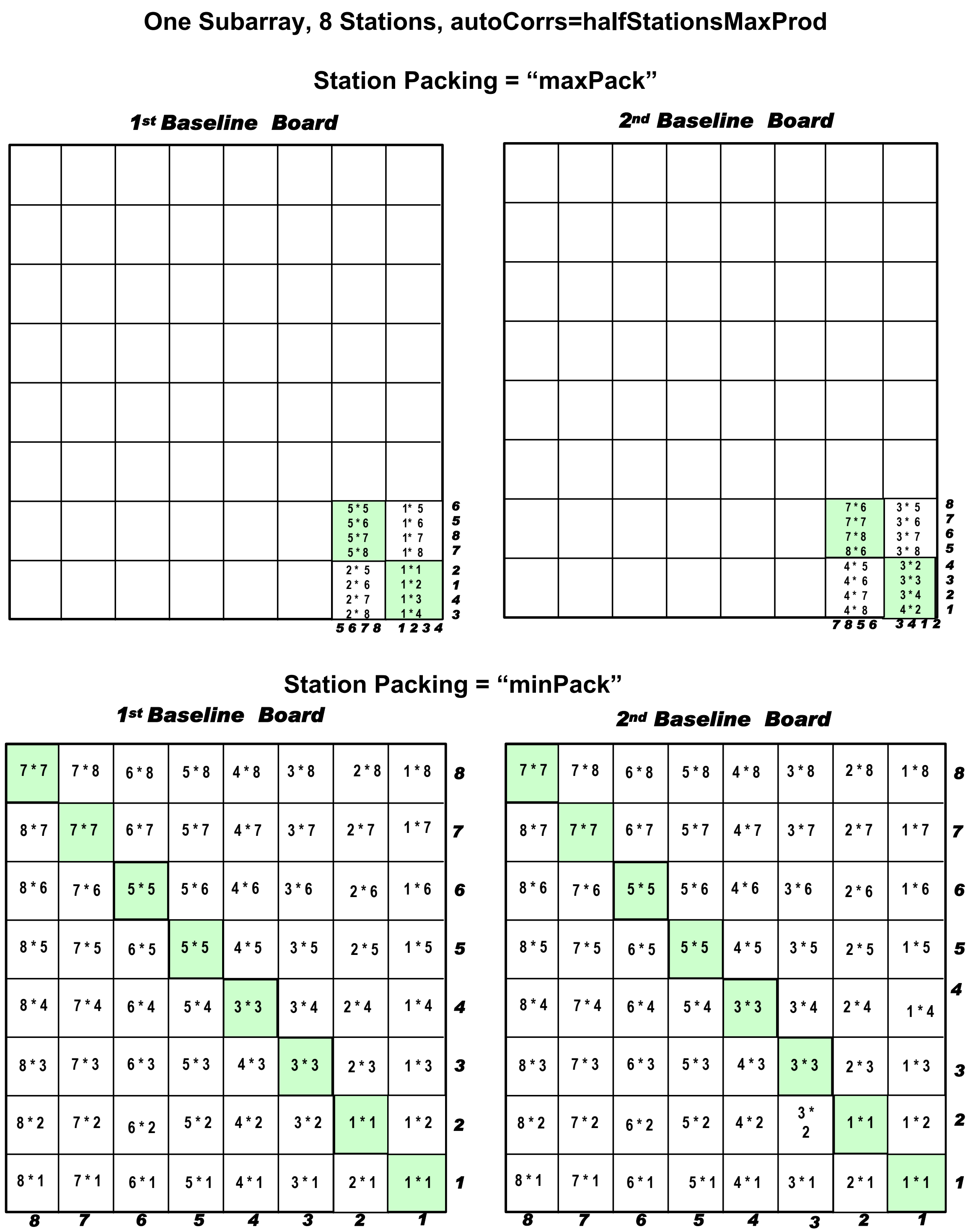
XML Example 6-14 Station Packing = MaxPack, 4pp @ 64 spectral channels

```
<baseBand bbA="2" bbB="3" name="BBP2" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
  <subBand sbid="0"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
       <polProducts>
           <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"
                                               ltaIntegFactor="10"  cbeIntegFactor="10">
           <pp id="1" correlation="A*A" spectralChannels="64"/>
           <pp id="2" correlation="A*B" spectralChannels="64"/>
           <pp id="3" correlation="B*A" spectralChannels="64"/>
           <pp id="4" correlation="B*B" spectralChannels="64"/>
           <blbPair quadrant="2" firstBlbPair="0" numBlbPairs="1" />
           <stationPacking algorithm="maxPack" />
       </polProducts>
   </subBand></baseBand >
```

XML Example 6-15 Station Packing = MinPack, 4 pp @ 64 spectral channels

```
<baseBand bbA="2" bbB="3" name="BBP2" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
  <subBand sbid="0"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
       <polProducts>
           <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"
                                              ltaIntegFactor="10"   cbeIntegFactor="10"/>
           <pp id="1" correlation="A*A" spectralChannels="64"/>
           <pp id="2" correlation="A*B" spectralChannels="64"/>
           <pp id="3" correlation="B*A" spectralChannels="64"/>
           <pp id="4" correlation="B*B" spectralChannels="64"/>
           <blbPair quadrant="2" firstBlbPair="0" numBlbPairs="1" />
           <stationPacking algorithm="minPack"/>
       </polProducts>
   </subBand></baseband>
```

Figure 6-9 Station Packing: Baseline Board Configuration



6.4.10.7 Product Packing Algorithm

In addition to choice of station packing, which determines how baselines are packed (or arranged) on the Baseline Boards, CM also allowes user to choose:

- to maximize the number of Correlator Chip Cells (CCCs) used on each Baseline Board (per baseline) in order to minimize the number of used Baseline Boards, or
- to minimize the number of Correlator Chip Cells (CCCs) used on each Baseline Board (per baseline), in order to minimize amont of data generated by individual boards (for faster dumping).

Table 6-22 ***subarray / stationInputOutput / baseband / subband / polProducts / productPacking***

subband parameter	R/O	Range	Must be same for all stations	Description
XML tag		productPacking	n/a	productPacking is an XML element, it is used to specify which algorithm should be used to configure Baseline Boards.
algorithm	R	maxPack, minPack Default=maxPack	Yes	Algorithm to be used to configure lag chain for the products. Default in this case defines the value which is assumed if the element *productPacking* is not specified.

Product packing *maxPack* means: use Correlator Chip Cells assigned to each baseline so that products are 'packed' as tight as possible, in order to minimize the number of Baseline Boards used for a particular subband. In this mode, if there are two BlB Pairs per subband and one BlB Pair is enough to get all the products; CM would use only one BlB Pair.

Product packing *minPack* means: use as many Baselines Boards as possible to minimize the number of Correlator Chip Cells used on individual boards.

Total number of used Correlator Chip Cells is same for both modes; this parameter changes the number of Correlator Chip Cells used per Baseline Board.

For productPacking=minPack, resulting correlator configuration depends on the number of Baseline Boards assigned per subband (assigned in VCI message).

Example:

For stationPacking=maxPack there is one CCQ per baseline per Basline Board Pair. In this example, there are 4 pol.products; each with 128 spectral channels. To get one pol.porduct with 128 spectral channels, without recirculation, we need 2 CCCs; total of 4*2= 8 CCCs per baseline. One Baseline Board Pair has 4 CCCs per baseline. The minimum number of Baseline Board Pairs for this configuration is 2.

In XML Example 6-17 user asked for productPacking=minPack but did not increase the number of Baseline Board Pairs; resulting configuration would be the same as for 'maxPack'. To reduce the number of cells used per Correlator Chip user must increase number of Baseline Board Pairs per subband.

XML Example 6-16 Product Packing = MaxPack

```xml
<baseBand bbA="4" bbB="5" name="BBP3" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
  <subBand sbid="0"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
        <polProducts>
            <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5"
                                                  ltaIntegFactor="10"    cbeIntegFactor="10"/>
            <pp id="1" correlation="A*A" spectralChannels="128"/>
            <pp id="2" correlation="A*B" spectralChannels="128"/>
            <pp id="3" correlation="B*A" spectralChannels="128"/>
            <pp id="4" correlation="B*B" spectralChannels="128"/>
            <blbPair quadrant="3" firstBlbPair="0" numBlbPairs="2" />
            <productPacking algorithm="maxPack"./>
        </polProducts>
    </subBand>
</baseband>
```

XML Example 6-17 Product Packing = MinPack

```xml
<baseBand bbA="4" bbB="5" name="BBP3" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
  <subBand sbid="2"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
        <polProducts>
            <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5"
                                                  ltaIntegFactor="10"    cbeIntegFactor="10"/>
            <pp id="1" correlation="A*A" spectralChannels="128"/>
            <pp id="2" correlation="A*B" spectralChannels="128"/>
            <pp id="3" correlation="B*A" spectralChannels="128"/>
            <pp id="4" correlation="B*B" spectralChannels="128"/>
            <blbPair quadrant="3" firstBlbPair="2" numBlbPairs="2" />
            <productPacking algorithm="minPack"./>
        </polProducts>
    </subBand>
</baseband>
```

6.4.10.8 Auto-Correlation Products

There are three methods to obtain auto-correlation products:

1. Auto-correlation products can be configured as a part of subarray configuration. On the Baseline Board, Correlator Chips on the diagonal of the Correlator Chip Array can be used to produce auto-correlation products. In some modes, EVLA correlator can not obtain full set of auto-correlation products for all antennas; that's why VCI protocol allows user to select a subset of auto-correlation products. Table 6-23 below lists available options.

2. The second method is to place Correlator Chips in the autocorrelator mode. By placing the whole column in the autocorrelator mode it is possible to obtain auto-correlation products for 8 input data streams (4 pairs of subbands). For more details see Section 6.4.13 below.

3. Wideband auto-correlation products (for baseband) can be obtained on the Station Board. For detailed description refer to section 6.4.7.

Table 6-23 lists parameters used to specify desired subset of auto-correlation products for the first method. A more detailed explanation of supported algorithms is provided later in the document, when describing various configuration modes.

Figure 6-4 on page 92 is an example of a Baseline Board Pair configuration for 32 station subarray. In the diagram, each Baseline Board is represented by 8*8 matrix of the Correlator Chips. Numbers along the top and left edge of each matrix are Station IDs, while numbers in the quadrants (n*m) are baselines. Each Correlator Chip receives up to 4 subband pairs (the same subband for up to 4 stations) on X and Y inputs. All input data streams are the same subband pair for stations identified by the Station ID.

Figure 6-4 shows that a Baseline Board Pair can produce:

- Cross-correlation products for 32 stations; and
- Auto-correlation products for a half of the stations (in this case 16).

Table 6-23 ***subarray / stationInputOutput / baseband / subband / polProducts / autoCorrelation***

subband parameter	R/O	Range	Subarray [1]	Description
XML tag	R	autoCorr	n/a	autoCorr is an XML element used to choose the subset of auto-correlation products.
algorithm	R	halfStationsMaxProd (1) timeMuxHalfStnMaxProd (2) allStationsMinProd (3) allStationsMaxProd (4) crossCorrOnly(5) autoCorrOnly (6) BB pair default = halfStationsMaxProd (1)	Yes	Desired subset of auto-correlation products: (1)Produce all the auto-correlation products for half of the stations. See parameter “startFrom”. (2) Time multiplex configurations (1) for supported values of the parameter startFrom. **Not implemented.** (3) Produce at least one auto-correlation product for each station. Number of products varies depends on other configuration parameters. **Not implemented.** (4) Get complete set of auto-correlation products. (5) Produce cross-products only, i.e. do not correlate baselines where x and y input come from the same antenna (station). (6) Auto-correlation only. In this case auto correlation products can be obtained for all the stations.
startFrom	O	lowestSID scndLowestSID Default= lowestSID	Yes	Can be specified if algorithm is halfStationsMaxProd (Baseband Pair) or halfStationsOneProd. Sort stations accordi ng to Station ID. lowestId means get auto-correlation products for every second station starting from the station with the lowest Station ID. scndLowestId means get auto-correlation products for every second station starting from the station with second lowest Station ID.
dwellTime	O	Positive integer	Yes	Relevant for the algorithm that performs time multiplexing; specifies for how many seconds at a time the correlator should produce each subset of products; in other words, how often to switch between different configurations.

[1] The parameter must be the same for all the stations in a subarray.

Figure 6-9 on page 105 shows stationPacking=minPack configuration for 8 stations; there are 2 Correlator Chips per baseline on each baseline Board. Correlator Chips on the diagonal get X and Y input from the same station and can produce auto-correlation products; however there is only one Correlator Chip per station. In other words, Baseline Board has twice as many resources for cross-correlations than for auto-correlations. Additional Baseline Board resources are needed to obtain auto-correlation products for all the stations (antennas).

User may choose:

1) to use more Baseline Boards to get all auto-correlation products for all the stations, or

2) to get all auto-correlation products for half of the stations, or

3) to get a subset of auto-correlation products for all the stations.

XML Example 6-18 shows subband configuration with and without element *autoCorr*. Note that correlator configuration for the Subbands 0 and 1 in XML Example 6-18 would be the same: for subband 0 auto-corr. algorithm is not explicitly specified, CM would use the default: halfStationsMaxProd starting from the lowest Station ID.

XML Example 6-18 Auto-Correlation Algorithm

```
<stationInOutData  stationID="all">
    <baseband bbA="0" bbB="1" bw="1024MHz" inQuant="8bit">
        <subband id="0" bw="128MHz" centralFreq="64000000" reQuant="4bit">
            <polProducts>
                        <productIntegration  recirculation="4" minIntegTime="250" ccIntegFactor="5"
                                                    ltaIntegFactor="10" cbeIntegFactor="10" />
                        <pp id="1" correlation="A*A" spectralChannels="512"/>
                        <pp id="2" correlation="A*B" spectralChannels="512"/>
                        <pp id="3" correlation="B*A" spectralChannels="512"/>
                        <pp id="4" correlation="B*B" spectralChannels="512"/>
                        <blbPair quadrant="1" firstBlbPair="0" numBlbPairs="4" />
            </polProducts>
        </subband>
        <subband id="1" bw="128MHz" centralFreq="64000000" reQuant="4bit" >
            <polProducts>
                        <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"
                                                    ltaIntegFactor="10"   cbeIntegFactor="10" />
                        <pp id="1" correlation="A*A" spectralChannels="512"/>
                        <pp id="2" correlation="A*B" spectralChannels="512"/>
                        <pp id="3" correlation="B*A" spectralChannels="512"/>
                        <pp id="4" correlation="B*B" spectralChannels="512"/>
                        <blbPair quadrant="1" firstBlbPair="4" numBlbPairs="4" />
                         <autoCorr algorithm="allStationsMinProd" startFrom="lowestProdID" />
            </polProducts>
        </subband>
        <subband id="2" bw="128MHz" centralFreq="64000000" reQuant="4bit"  >
                <polProducts>
                        <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"
                                                    ltaIntegFactor="10"   cbeIntegFactor="10" />
                        <pp id="1" correlation="A*A" spectralChannels="512"/>
                        <pp id="2" correlation="A*B" spectralChannels="512"/>
                        <pp id="3" correlation="B*A" spectralChannels="512"/>
                        <pp id="4" correlation="B*B" spectralChannels="512"/>
                        <blbPair quadrant="1" firstBlbPair="8" numBlbPairs="4" />
                        <autoCorr algorithm="halfStationsMaxProd" startFrom="scndLowestSID"  >
                </polProducts>
        </subband>

    </baseband>
</stationInOutData>
```

6.4.10.9 Four Stations per Row/Column (maxPack)

In this mode it is possible to get up to 4 polarization products per baseline on a single Baseline Board Pair for up to 32 stations (antennas). Figure 6-4 on page 92 shows a pair of Baseline Boards configured for a subarray that has 32 antennas (stations). Each row/column on the Baseline Board gets input from 4 antennas and each Correlator Chip produces products for 4 baselines.

Number of Baseline Board Pairs for each subband depends on:
- number of bits to correlate,
- number of spectral channels per product,
- number of products,
- recirculation factor,
- product packing mode and
- subset of auto-correlation products.

stationPackingAlgorithm=fourPerRowColumn

4-Bit Correlation

For stationPackingAlgorithm=fourPerRowColumn (maxPack) and 4-bit correlation, a pair of Baseline Boards, without recirculation, can produce 4 polarization products, 64 spectral channels per product for up to 32 stations.

Station packing algorithm fourPerRowColumn (maxPack) arranges Baseline Board input so that each Correlator Chip receives input (subband pair) for 4 different stations.

1. Correlator Chip has 4 Correlator Chip Quads (CCQs); each CCQ performs correlation for a different pair of stations (antennas).
2. CCQ consists of 4 Correlator Chip Cells (CCCs). Each CCC can produce 128 complex lags, or 64 spectral channels for a single product.
3. Each of 4 CCCs in a CCQ can be used to produce different polarization product, but CCCs can also be concatenated so that two, three or four CCCs all produce lags for the same polarization product.

For 4-bit correlation, *no recirculation* and station packing algorithm fourPerRowColumn (maxPack), a pair of Baseline Boards can produce:
1. 4 polarization products per baseline @ 64 spectral channels, or
2. 2 polarization products per baseline @ 128 spectral channels, or
3. 1 polarization product per baseline @ 256 spectral channels.

The list above shows the maximum number of products and spectral channels that can be produced on a single BlB pair.

Number of spectral channels per product must be:

- Lower or equal to the maximum listed above, and
- Integer multiple of 64.

For example, all of the following is valid:

- 2 polarization products @ 64 spectral channels, or
- 1 polarization product @ 64 spectral channels, or
- 1 polarization product @ 128 spectral channels, or
- 1 polarization product @ 192 spectral channels.

stationPackingAlgorithm=fourPerRowColumn

Dynamic Recirculation

Recirculation can be used to get more spectral channels per product. When using recirculation, in this mode, the following restrictions apply:

- Maximum number of spectral channels per product: 16384
- Phase for recirculation must be serial.

Term 'dynamic recirculation' is used to refer to the case where, in order to obtain more spectral channels, the same data samples are recirculated, i.e. repeatedly sent through the Correlator Chip, each time with a different delay. Recirculation factor specifies how many times data samples are recirculated. For detailed information regarding recirculation plase refer to [16].

Number of spectral channels that can be obtained with a given number of Baseline Boards can be calculated as follows:

Number of spectral channels without recirculation * recirculation factor

Valid range for recirculation factor is 1 to 256.

Recirculation factor of one (1) means no recirculation (the default value).

To avoid loss of sensitivity, recirculation factor should be the same or lower than the subband decimation factor (othervise samples will be dropped). For a subband bandwith=128MHz/n, the maximum recirculation factor that does not cause loss of sensitivity is n.

For example:

1. for subband bandwidth=128MHz, the maximum recirculation factor that does not cause loss of sensitivity is 1, i.e. no recirculation.
2. for subband bandwidth=64MHz, the maximum recirculation factor that does not cause loss of sensitivity is 2 (since 128/2=64).

<u>Recirculation factor must be choosen so that number of spectral channels per product does not exceede 16384.</u>

<u>16384 is the maximum number of spectral channels per product that can be obtained in stationPacking=fourPerRowColumn (maxPack) mode.</u>

The limit on the number of spectral channels is imposed by hardware, more precisely by Recirculation Controller. Explanation:

1. To get more than 4pp@64 spectral channels or 2pp@128 spectral channels or 1pp@256 spectral channels we have to use recirculation (either dynamic or static).
2. Each row/column on the Baseline Board gets input from 4 stations, which means, if recirculation is used, Recirculation Controller has to perform recirculation for 4 pairs of data streams (8 data streams).

For products with less than 16384 spectral channels, Recirculation Controller is capable of performing recirculation for 4 pairs of data streams. For products with more than 16384 spectral channels, Recirculation Controller can perfirm recirculation for only 2 pairs of data streams.

stationPackingAlgorithm=fourPerRowColumn

Static Recirculation

(Baseline Board Stacking)

To avoid use of dynamic recirculation (which causes loss of sensitivity), one can use several Baseline Board Pairs per subband. This is referred to as static recirculation or Baseline Board Stacking. Term static recirculation is used in WIDAR Baseline Board documentation, while term Baseline Board Stacking is used when describing Baseline Board allocation.

For example, for stationPacking=fourPerRowColumn:

- 1 Baseline Board Pair can produce 4pp@64 spectral channels,
- 4 Baseline Board Pairs can produce 4pp@256 spectral channels.

For 4pp@256 spectral channels, Correlator Chips on all four Baseline Board Pairs are configured in the same way, but Recirculation Controlers are configured so that each Baseline Board Pair produces different segment of the lag chain:

```
1st BlB Pair: lags   0 to  64
2nd BlB Pair: lags  64 to 127
3rd BlB Pair: lags 128 to 195
4th BlB Pair: lags 196 to 255.
```

Each Baseline Board Pair produces the same segment for all polarization products.

For Baseline Board Stacking, number of spectral channels can be calculated as:

Number of spectral channels produced by a single BlB pair * number of BlB pairs.

There is no limit on the number of BlB pairs that can be assigned to a single subband. All the Baseline Boards in the system (64 pairs) can be assigned to a single subband. If that is the case, for station packing algorithm fourPerRowColumn (maxPack), the maximum number of spectral channels per product, without dynamic recirculation, is:

4pp per baseline: 64 * 64 = 4096

2pp per baseline: 128 * 64 = 8192

1pp per baseline: 256 * 64 = 16384

It is possible to use both dynamic and static recirculation at the same time and get more spectral channels, however, **16384 is the maximum number of spectral channels per product that can be obtained in stationPacking=fourPerRowColumn (maxPack) mode.** This limit is imposed by Recirculation Controller. For detailed explanation see the section on dynamic recirculation (above).

stationPackingAlgorithm=fourPerRowColumn

7-Bit Correlation

For 7-bit correlation, in the Baseline Board Recirculation Controller, each input data stream (subband) is devided in two 4-bit streams (upper and lower nibble); correlation is perfomed for each combination:

a) upper*upper,
b) upper*lower,
c) lower*upper,
d) lower*lower.

Obviously, four sub-products are required for each polarization product; as a consequence, four times more Baseline Board Pairs are needed to get the same polarization products. For example, for ‘four stations per row/column’ 4 BlB pairs are needed to get 4 polarization products @ 64 spectral channels.

Figure 6-6 on page 95 shows that, for 7-bit correlation, 64 BlB pairs are needed to get 4 polarization products for 16 subbands.

Baseline Boards are configured in the same way as for 4-bit correlation, with exception of “nibble selection” in the Recirculation Controller which is used to select upper or lower nibble.

stationPackingAlgorithm=fourPerRowColumn

GUI Screenshots

Figure 6-10 shows a Correlator Chip configured for stationPacking=fourPerRowColumn (maxPack). Correlator Chip {0,0} is on the diagonal, it is configured to produce 4 polarization products for 3 baselines and 3 auto-correlation products for one station (station 3 in the picture). Auto-correlation products R*L and L*R are identical and only one product is required. Auto-correlation algorithm is halfStationsMaxProd, starting from the lowest ID. Configuration for this Correlator Chip would be the same for autoCorr=allStationsMaxProd, but additional rows/columns would be used to get auto-correlation products for Station 5.

Figure 6-11 is a screenshot of the Baseline Board Map GUI; shows Correlator Chip Cells and Correlator Chip input for a subarray with 25 stations, 4 polarization products per baseline.

Baseline Board b101-t-1 is the 1st Baseline Board in the Baseline Board Pair Q1P0.

Figure 6-10 Correlator Chip Configuration for stationPacking=fourPerRowColumn (maxPack), autoCorr=halfStationsMaxPack, starting from the lowest Station ID.

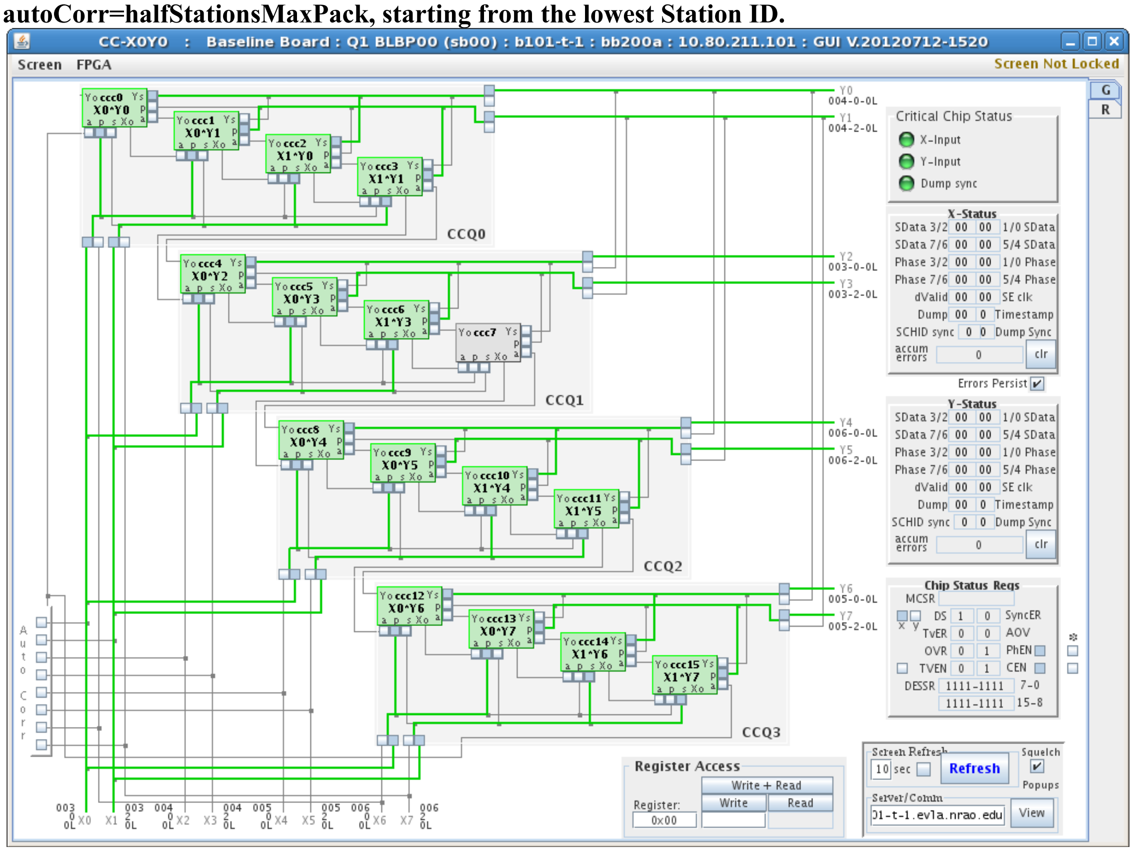
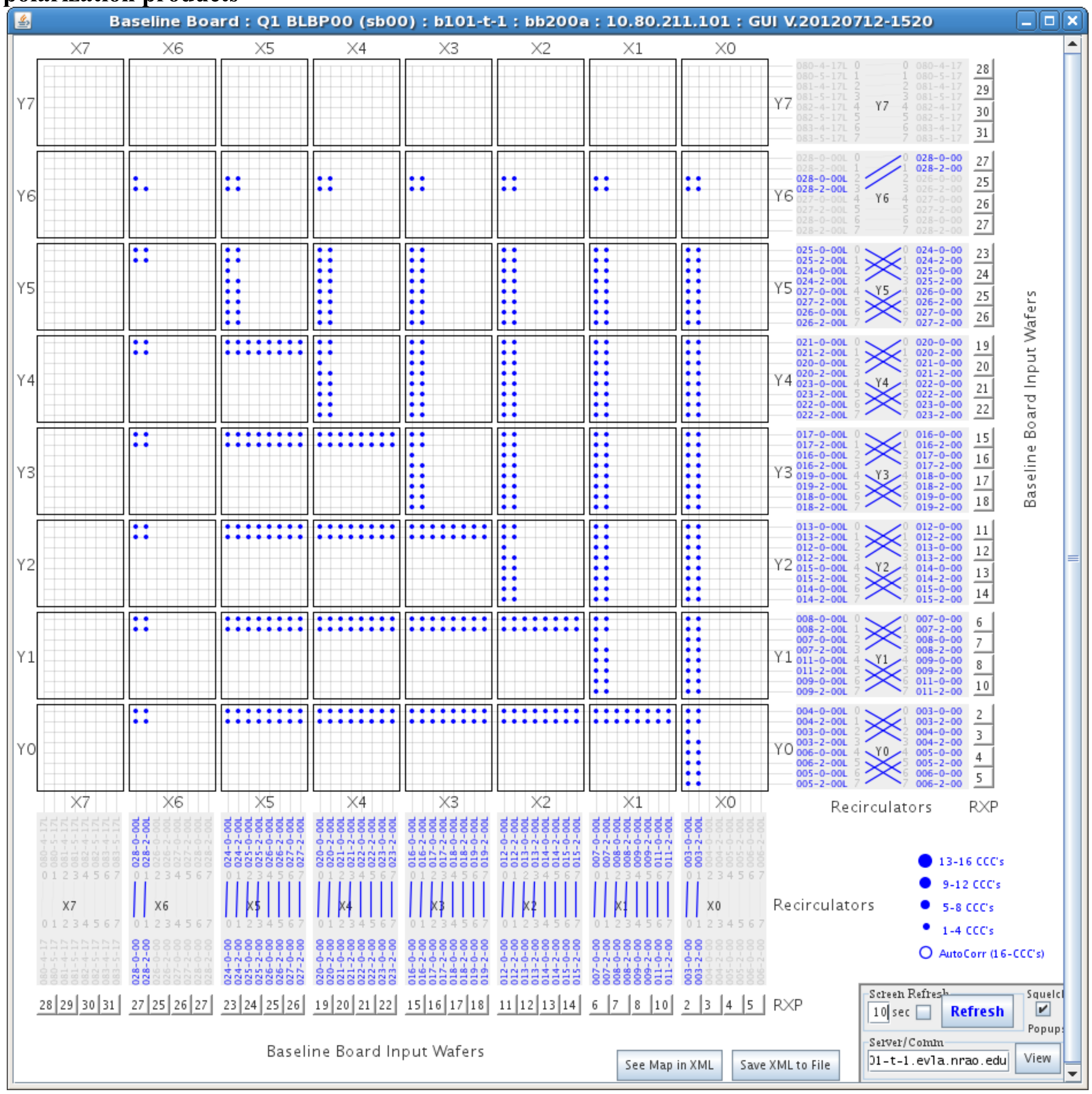


Figure 6-11 Baseline Board GUI Map: an overview of Baseline Board configuration for a subarray with 25 antennas, stationPacking=fourPerRowColumn, autoCorrs=halfStationsMaxProd 4 polarization products



stationPackingAlgorithm=fourPerRowColumn

productPackingAlgorithm=maxPack

productPackingAlgorithm=maxPack means: configure Correlator Chips to minimize number of used Baseline Boards.

For stationPacking=fourPerRowColumn and stationPacking=maxPack CM uses the minimum possible number of Baseline Boards. Based on the number of products per baseline and number of CCCs required per product, CM determines the maximum size of the lag chain segment.

In this mode there are 4 CCCs per baseline per Baseline Board Pair.

- For 4pp the maximum size of a lag chain segment is 1 CCC.
- For 2pp the maximum size of a lag chain segment is 2 CCCs.
- For 1pp the maximum size of a lag chain segment is 4 CCCs.

Restriction: All segments must be of the same size.

The minimum number of Baseline Board Pairs can be calculated as follows:

```
Number of Correlator Chip Cells per product:

numCCCs = numSpectralChannels / 64 / recirculationFactor

4 polarization products:

numBlbPairs = numCCCs

2 polarization products:

if ((numCCCs modulo 2) == 0)      numBlbPairs = numCCCs/2
else                              numBlbPairs = numCCCs

1 polarization product:

if      ((numCCCs modulo 4) == 0) numBlbPairs = numCCCs/4
else if ((numCCCs modulo 2) == 0) numBlbPairs = numCCCs/2
else                              numBlbPairs = numCCCs
```

Examples:

- For 4pp@512 spectral channels, no recirculation, number of CCCs is 512/64=8. There is only one CCC per product per baseline per BlB Pair, 8 BlB Pirs are needed to get all the products. No matter how many Baseline Board Pairs were assigned to this subband in the VCI message, CM uses 8 BlB Pairs. If less than 8 BlB Pairs are assigned, configuration is rejected.

- For 4p@448 spectral channels, no recirculation, number of CCCs is 448 / 64 = 7. There is one CCC per product per baseline per BlB Pair. Seven Baseline Board Pairs are needed to get specified products.

- For 2pp@512 spectral channels, no recirculation, number of CCCs is 512/64=8. There are 2 CCCs per product per baseline per BlB Pair. 4 BlB Pairs are needed to get products.
- For 2pp@488 spectral channels, no recirculation, number of CCC is 448 / 64 = 7. There are 2 CCCs per product per baseline per BlB Pair, *however, since seven cannot be devided by 2, each segment can use only one CCC and 7 BlB Pairs are needed to get products!*

stationPackingAlgorithm=fourPerRowColumn

productPackingAlgorithm=minPack

This mode is used to minimize the number of used cells per Correlator Chip, in order to reduce the amount of data generated by each Baseline Board (for fast dumping).

For stationPacking=fourPerRowColumn there is one CCQ (4 CCCs) per baseline on a BlB Pair.

Number of CCCs to be used per product is determined based on the number of spectral channels per product and recirculation factor.

For **4 polarization products per baseline** there is one CCC per product per baseline on a BlB Pair, which means that 64 spectral channels per product can be obtained using one BlB Pair. Using single BlB Pair, more spectral channels can be obtained using dynamic recirculation. The minimum number of BlB Pairs required to configure polarization products can be calculated from the number of spectral channels and recirculation factor.

If configuration requires more than one CCC per product, Baseline Board Stacking must be used; product lag chain is devided into 128 lags segments (which corresponds to 64 spectral channels). The minimum number BlB Pairs required for a subband is equal to the number of lag chain segment per product.

- If the number of BlB Pairs assigned to the subband (in the VCI message) is *equal to* the minimum number of BlB pairs required, configuration uses 4 CCQs per Correlator Chip and 4 CCCs per CCQ (per baseline); in total 16 CCCs per Correlator Chip (maximum).
- If the number of BlB Pairs assigned to the subband is *two times* the minimum number of BlB pairs, in order to minimize number of CCCs used per Correlator Chip, CM would configure only two polarization products per baseline on the same Correlator Chip. Such configuration uses 4 CCQs (for 4 baselines), 2 CCCs per CCQ; in total 8 CCCs per Correlator Chip.
- If the number of BlB Pairs assigned to the subband is *four times* the minimum number of BlB pairs required, each polarization product can be configured on a different BlB Pair. Such configuration uses 4 CCQs per Correlator Chip, one CCC per CCQ, in total 4 CCCs per Correlator Chip. See Figure 6-12.

- If the number of BlB Pairs assigned to the subband is *eight times* the minimum number of BlB pairs required, number of used CCCs per Correlator Chip is further reduced so that only 2 baselines (instead of 4) are configured per Correlator Chip. Such configuration uses 2 CCQs per Correlator Chip, one CCC per CCQ; in total 2 CCCs per Correlator Chip.
- If the number of BlB Pairs assigned to the subband is *sixteen times* the minimum number of BlB pairs required, number of used CCCs per Correlator Chip can be further reduced: each polarization product of each baseline can be configured on a different BlB Pair. Such configuration uses 1 CCC per CCQ and one CCQ per Correlator Chip; total of 1 CCC per Correlator Chip. See Figure 6-12.

For **2 polarization products per baseline** on each BlB Pair there are two CCCs per product per baseline, which means that 128 spectral channels per product can be obtained using single BlB Pair. More spectral channels can be obtained using dynamic recirculation. The minimum number of Blb Pairs required to configure polarization products can be calculated from the number of spectral channels and recirculation factor.

If the number of BlB Pairs assigned to a subband (in the VCI message) is *equal to* the minimum number of BlB pairs required, configuration uses 4 CCQs per Correlator Chip (for 4 baselines) and 4 CCCs per CCQ (2 per product); in total 16 CCCs per Correlator Chip (maximum). For 2pp with 64 spectral channels per product, 8 CCCs per Correlator Chip would be used.

If the number of BlB Pairs assigned in the VCI message is greater then the minimum, CM would try to minimize the number of used CCCs per Correlator Chip.

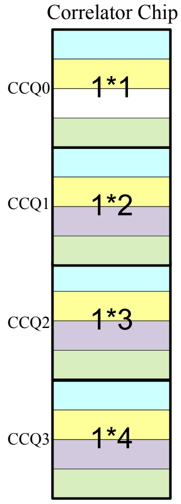
- First, CM would minimize the size of the lag chain segment so that only one CCC per product is used per Correlator Chip. For a given number of BlB pairs, CM calculates the minimum size for the lag chain segments. If the lag chain segment size is 128 lags, such configuration would use 4 CCQs per Correlator Chip, 1 CCCs per CCQ; in total 4 CCCs per Correlator Chip.
- If there are still unused BlB Pairs, CM would try to configure each polarization product on a different BlB Pair, in the same manner as for 4 pol. products per baseline.
- And finally CM would try to configure each baseline on a different BlB Pair, in the same manner as for 4 polarization products per baseline.

See Figure 6-13.

For **1 polarization product per baseline** there are 4 CCCs per product per baseline on a BlB Pair, which means that 256 spectral channels per product can be obtained using a single BlB Pair. If product packing mode is minPack and the number of BlB Pairs assigned in the VCI message is greater then the minimum, CM would try to minimize the number of used CCCs per Correlator Chip in the same manner as for 2 polarization products per baseline.

Figure 6-12 stationPacking=fourPerRowColumn, productPacking=minPack, 4pp@64 spectral chans.

Example 1:
Correlator Chip x0y0



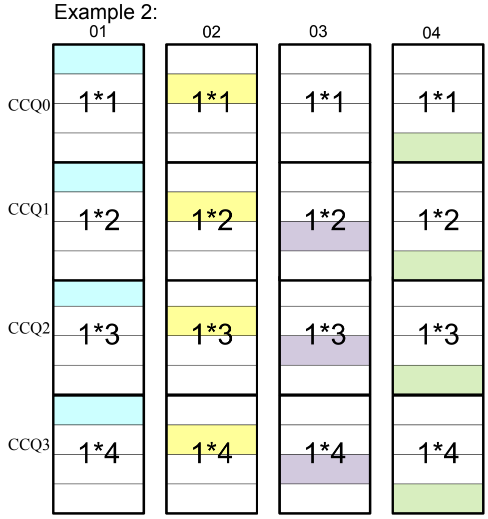
Three examples for
stationPacking = fourPerRowColumn (maxPack),
4 pol.products per baseline,
64 spectral channels per product.
Needs 1 CCC per product, 4 CCCs per baseline.

Not used
Pol.Product R*R
Pol.Product R*L
Pol.Product L*R
Pol.Product L*L

Example 1 VCI configuration: productPacking=minPack, 1BlB Pair
Configuration is the same as for ProductPacking=maxPack;
Uses 1 BlB Pair, one CCQ per baseline.
A Correlator Chip can produce 4 polarization products for 4 baselines.

Diagram shows Correlator Chip x0y0.
Each rectangle represents a Correlator Chip Cell (CCC).
There are 16 CCCs per Correlator Chip, organized as 4 Correlator Chip Quads (CCQs).
Each CCQ has 4 CCCs. Each CCQ gets input from a pair of antennas (X and Y).
For example: 1*2 indicates that CCQ correlates input from antennas 1 and 2.
Color is used to indicate polarization products.

Example 2:



Example 2: productPacking=minPack, 4BlB Pairs
To minimize the number of used CCCs per Correlator Chip,
each polarization product is configured on a different BlB Pair.
This configuration uses 4 CCCs per Correlator Chip.
Diagram shows Correlator Chip x0y0 on four Baseline Boards.

Example 3: productPacking = minPack, 16 BlbPairs
Each polarization product for each baseline can be placed on a
different BlB Pair.
This configuration uses one CCC per Correlator Chip.
Diagram shows Correlator Chip x0y0 on 16 Baseline Boards.

Example 3:

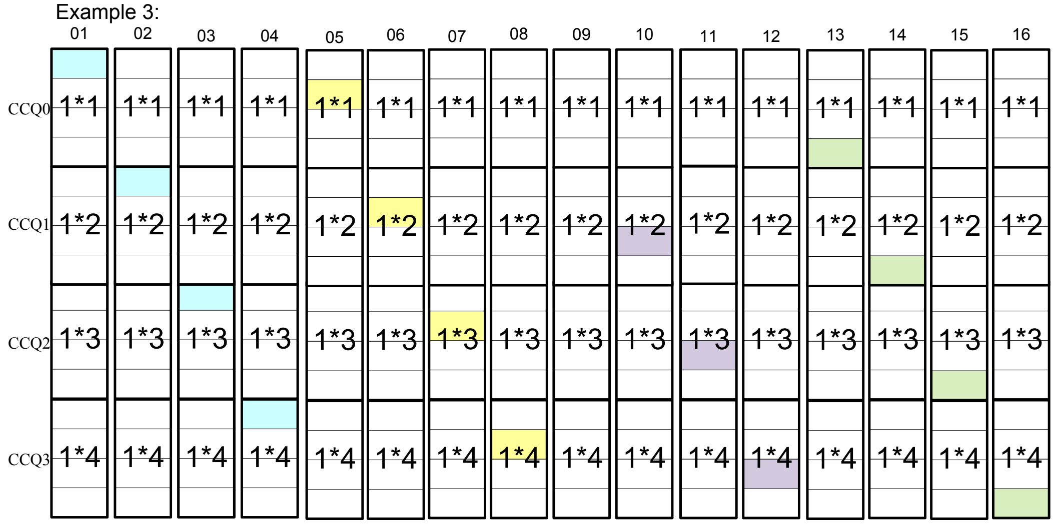
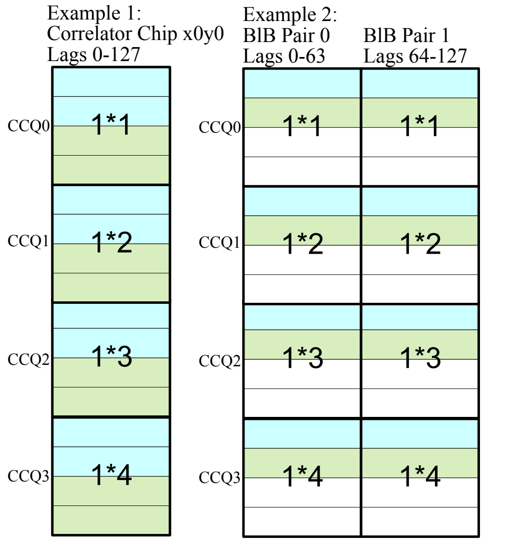


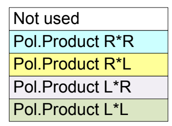
Figure 6-13 stationPacking=fourPerRowColumn, productPacking=minPack, 2pp@128 s. chans.

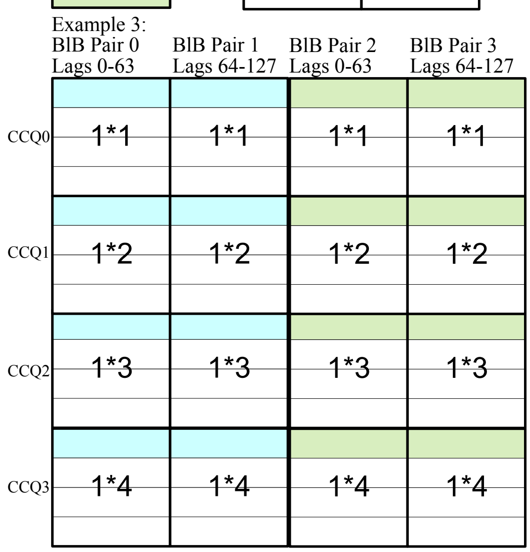


stationPacking = fourPerRowColumn (maxPack),
2 polarization products per baseline,
128 spectral channels per product, no recirculation.
Needs 2 CCCs per product, 4 CCCs per baseline.

Example 1: productPacking=minPack, 1 BlB Pair.
A Correlator Chip can produce 2 pol. products for 4 baselines.
Configuration is the same as for productPacking=maxPack.

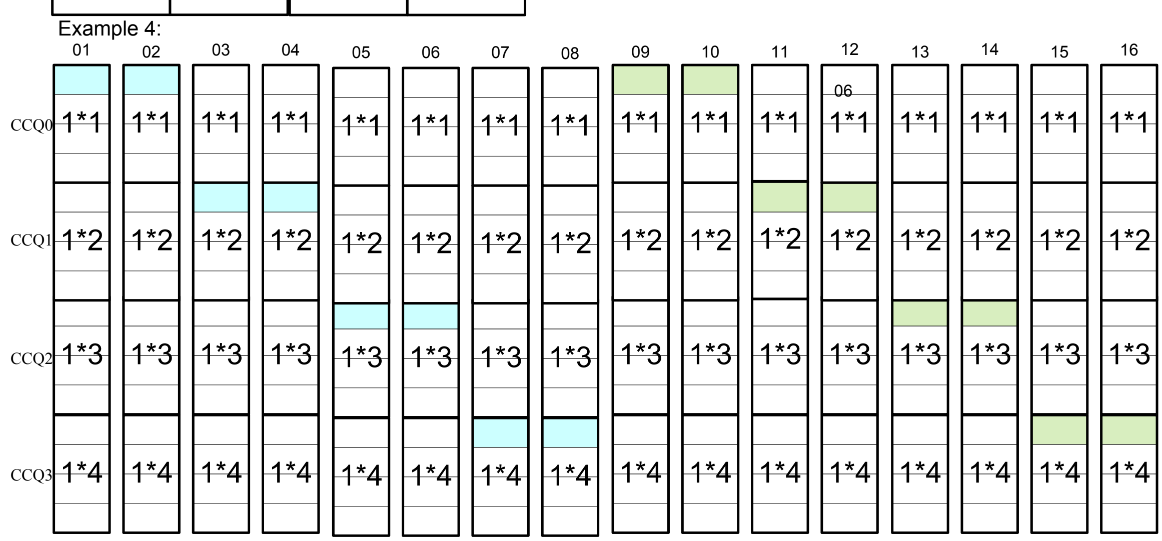
Example 2: productPacking=minPack, 2 BlB Pairs.
There are 2 CCQs per baseline; to minimize the number of CCCs used on each Baseline Board, each BlB Pair is configured to produce a half of the lag chain for both products.
Diagram shows Correlator Chip x0y0 on 2 Baseline Boards.





Example 3: productPacking=minPack, 4 Blb Pairs
4 CCQs (16 CCCs) per baseline are available.
This configuration uses one CCC per baseline on each BlB Pair.
Diagram shows Correlator Chip x0y0 on 4 Baseline Boards.

Example 4: productPacking = minPack, 16 BlB Pairs
Each polarization product for each baseline can be placed on a different BlB Pair.
This configuration uses one CCC per Correlator Chip.
Diagram shows Correlator Chip x0y0 on 16 Baseline Boards.



stationPackingAlgorithm=fourPerRowColumn (maxPack)

Auto-Correlation Products

Figure 6-4 on page 92 shows a Baseline Board Pair configured for *stationPacking=fourPerRowColumn*, 32 stations.

Correlator Chips on the diagonal each have one CCQ for auto-correlation products and are configured to produce auto-correlation products for half of the stations; more precisely, for every second station starting from the station with the lowest ID.

Configuration shown in Figure 6-4 is the default, which means that if the algorithm (mode) for auto-correlation products is not explicitly specified in the VCI message, Baseline Boards will be configured in this manner.

Alternatively, user can choose to get auto-correlations for complementary set of stations, i.e. for every second station starting with the station with the second lowest ID.

A complete set of auto-correlations for all the stations in a subarray can be obtained (using keyword allStationsMaxProd) for up to 26 stations, but not for a subarray that consists of 27 stations.

If autoCorr=allStationsMaxProd is specified for a subarray with more than 26 stations, configuration is *not* rejected, CM changes to mode to autoCorr=halfStationsMaxProd and proceeds with configuration.

Ratonal: In the EVLA system, number of antennas is not assigned when the configuration script is created, most of the time when creating VCI script user is not sure how many antennas will be used for a particular subarray.

Time-multiplexing between two sets of 'halfStationsMaxProd' could be implemented to provide complete set of auto-correaltion products for all the stations. Time-multiplexing between two Correlator Chip configurations on the Baseline Board diagonal could be implemented either in CM or in CMIB software. In either case CM would have to prepare configuration messages for CMIB, but the periodic switching between two Correlator Chip configurations should be implemented by CMIB software, to avoid periodic retransmission of the same messages, long lineup of messages in CMIB input queues, issues with aborting observations, and similar.

A mode where a single auto-correlation product would be generated for each station (maxStationsMinProd) was initially planned but is not implemented (it is not considered useful). In such mode, product A*A would be obtained for half of the stations and product B*B for the other half. In the same manner as for 'halfStationsMaxProd' , product A*A would be configured for every second station, starting from the lowest station ID. Product B*B would be configured for every second station starting from the second lowest ID.

6.4.10.10 Two Stations per Row/Column (midPack)

For stationPackingAlgorithm=twoPerRowColumn (midPack) Baseline Boards are configured so that each row/column gets input for two stations (antennas), there is one Correlator Chip per baseline, which means that up to 1024 spectral channels per baseline can be obtained without recirculation.

Using recirculation and/or Baseline Board stacking, the maximum number of spectral channels per product is as follows:

- **If phase for recirculation is serial, up to 262,144 spectral channels per product.**
- **If phase for recirculation is parallel, up to 16,384 spectral channels per product.**

Number of Baseline Board Pairs for each subband depends on:
- number of antennas,
- number of bits to correlate,
- number of spectral channels per product,
- number of products,
- recirculation factor,
- product packing mode and
- subset of auto-correlation products.

Figure 6-14 and Figure 6-15 show Correlator Chip Array on four Baseline Board Pairs configured for a subarray with 27 antennas. Figure 6-14 shows the first Baseline Board for all 4 pairs. Figure 6-15 shows the second Baseline Board for all 4 pairs. Each square represents a Correlator Chip; numbers in the squares are baselines. Numbers on the side and bottom are Station IDs in the Recirculation Controller input and Correlator Chip input.

All the modes supported for stationPacking=maxPack (fourPerRowColum) are also supported in this mode, namely:
- 4-bit correlation and 7-bit correlation,
- Dynamic and static recirculation (Baseline Board Stacking) ,
- ProductPacking maxPack and minPack,
- AutoCorrAlgorithm: halfStationsMaxProd and allStationsMaxProd.

Table 6-24 shows the minimum number of Baseline Board Pairs required for:
- 4-bit correlation,
- productPacking=maxPack,
- no recirculation,
- auto-correlation products halfStationsMaxProd and allStationsMinProd.

Table 6-24 The minimum number of Baseline Board Pairs for stationPacking=twoPerRowColumn (midPack).

	halfStationsMaxProducts	allStationsMaxProducts
1 BlbPair	Up to 16 stations	Up to 10 stations
3 BlbPairs	17 to 24 stations	8 to 22 stations
4 BlbPairs	Up to 32 stations	22 to 27 stations

For 7-bit correlation: multiply the number of BlB Pairs neded for 4-bit correlation by 4.

For static recirculation (Baseline Board Stacking): multiply the number of BlB Pairs by the number of lag chain segments.

Number of segments can be determined as follows:

For 4 pol.products: (number of spectral channels per product / 256)

For 2 pol.products: (number of spectral channels per product / 512)

For 1 pol.product: (number of spectral channels per product / 1024)

If number of spectral channels per polarization product is not multiple of 1024/512/256 (depending on the number of product), find the maximum size of the lag chain segment that can be used, given that all segments must be of the same size.

Figure 6-14 stationPacking=twoPerRowColumn, the 1st Baseline Board for 4 Baseline Board Pairs. Subarray has 27 stations. Each Correlator Chip is used for different baseline or station.

Black font is configuration for autoCorrAlgorithm=halfStationsMaxProd.

Black and blue font is configuration for autoCorrAlgorithm=allStationsMaxProd.

Pair 2, Baseline Board 1 (upper left); Pair 3, Baseline Board 1 (upper right); Pair 4, Baseline Board 1 (lower left); Pair 1, Baseline Board 1 (lower right)

RC Input X/Y	CC Input Y	6 - - -	2 - - -	27 - - -	25 26 26 26	23 24 24 24	21 22 22 22	19 20 20 20	17 18 18 18	15 - - -	13 - - -	11 - - -	9 - - -	7 - - -	5 - - -	3 - - -	1 - - -	CC Input Y	RC Input X/Y
6 8 - -	6 - - -	**6*6**								15*24	13*24	11*24	9*24	7*24	5*24	3*24	1*24	24 - - -	15 24 16 24
2 4 - -	2 - - -		**2*2**							15*23	13*23	11*23	9*23	7*23	5*23	3*23	1*23	23 - - -	13 23 14 23
27 - - -	27 - - -				25*27	23*27	21*27	19*27	17*27	15*22	13*22	11*22	9*22	7*22	5*22	3*22	1*22	22 - - -	11 22 11 23
25 26 26 26	25 26 26 26				**25*26**	23*25	21*25	19*25	17*25	15*21	13*21	11*21	9*21	7*21	5*21	3*21	1*21	21 - - -	9 21 10 21
23 24 24 24	23 24 24 24				23*26	**23*24**	21*23	19*23	17*23	15*20	13*20	11*20	9*20	7*20	5*20	3*20	1*20	20 - - -	7 20 8 20
21 22 22 22	21 22 22 22				21*26	21*24	**21*22**	19*21	17*21	15*19	13*19	11*19	9*19	7*19	5*19	3*19	1*19	19 - - -	5 19 6 19
19 20 20 20	19 20 20 20				19*26	19*24	19*22	**19*20**	17*19	15*18	13*18	11*18	9*18	7*18	5*18	3*18	1*18	18 - - -	3 18 4 18
17 18 18 18	17 18 18 18				17*26	17*24	17*22	17*20	**17*18**	15*17	13*17	11*17	9*17	7*17	5*17	3*17	1*17	17 - - -	1 17 2 17
15 26 16 -	26 - - -	**26*26**								**15*16**	13*15	11*15	9*15	7*15	5*15	3*15	1*15	15 16 16 16	15 16 16 16
13 22 14 24	22 - - -		**22*22**							13*16	**13*14**	11*13	9*13	7*13	5*13	3*13	1*13	13 14 14 14	13 14 14 14
11 18 12 20	18 - - -			**18*18**						11*16	11*14	**11*12**	9*11	7*11	5*11	3*11	1*11	11 12 12 12	11 12 12 12
9 14 10 16	14 - - -				**14*14**					9*16	9*14	9*12	**9*10**	7*9	5*9	3*9	1*9	9 10 10 10	9 10 10 10
7 10 8 12	10 - - -					**10*10**				7*16	7*14	7*12	7*10	**7*8**	5*7	3*7	1*7	7 8 8 8	7 8 8 8
5 27 6 27	27 - - -	15*27	13*27	11*27	9*27	7*27	5*27	3*27	1*27	5*16	5*14	5*12	5*10	5*8	**5*6**	3*5	1*5	5 6 6 6	5 6 6 6
3 26 4 26	26 - - -	15*26	13*26	11*26	9*26	7*26	5*26	3*26	1*26	3*16	3*14	3*12	3*10	3*8	3*6	**3*4**	1*3	3 4 4 4	3 4 4 4
1 25 2 25	25 - - -	15*25	13*25	11*25	9*25	7*25	5*25	3*25	1*25	1*16	1*14	1*12	1*10	1*8	1*6	1*4	**1*2**	1 2 2 2	1 2 2 2
	CC Input X →	15 26 26 26	13 22 22 22	11 18 18 18	9 14 14 14	7 10 10 10	5 - - -	3 - - -	1 - - -	15 16 16 16	13 14 14 14	11 12 12 12	9 10 10 10	7 8 8 8	5 6 6 6	3 4 4 4	1 2 2 2		

Figure 6-15 stationPacking=twoPerRowColumn, the 2nd Baseline Board for 4 Baseline Board Pairs. Subarray has 27 stations. Each Correlator Chip is used for different baseline or station.

Black font is configuration for autoCorrAlgorithm=halfStationsMaxProd.

Black and blue font is configuration for autoCorrAlgorithm=allStationsMaxProd.

RC Input X/Y	CC Input Y	Pair 2, Baseline Board 2								Pair 3, Baseline Board 2								CC Input Y	RC Input X/Y
		8 - - -	4 - - -	27 - - -	26 25 25 25	24 23 23 23	22 21 21 21	20 19 19 19	18 17 17 17	16 - - -	14 - - -	12 - - -	10 - - -	8 - - -	6 - - -	4 - - -	2 - - - -		
6 8 - -	8 - - -	8*8								16*24	14*24	12*24	10*24	8*24	6*24	4*24	2*24	24 - - -	15 24 16 24
2 4 - -	4 - - -		4*4							16*23	14*23	12*29	10*23	8*23	6*23	4*23	2*23	23 - - -	13 23 14 23
27 - - -	27 - - -			27*27	26*27	24*27	22*27	20*27	18*27	16*22	14*22	12*22	10*22	8*22	6*22	4*22	2*22	22 - - -	11 22 12 22
25 26 - -	25 26 26 26				25*25	24*25	22*25	20*25	18*25	16*21	14*21	12*21	10*21	8*21	6*21	4*21	2*21	21 - - -	9 21 10 21
23 24 - -	23 24 24 24				24*26	23*23	22*23	20*23	18*23	16*20	14*20	12*20	10*20	8*20	6*20	4*20	2*20	20 - - -	7 20 8 20
21 22 - -	21 22 22 22				22*26	22*24	21*21	20*21	18*21	16*19	14*19	12*19	10*19	8*19	6*19	4*19	2*19	19 - - -	5 19 6 19
19 20 - -	20 19 19 19				20*26	20*24	20*22	19*19	18*19	16*18	14*18	12*18	10*18	8*18	6*18	4*18	2*18	18 - - -	3 18 4 18
17 18 - -	18 17 17 17				18*26	18*24	18*22	18*20	17*17	16*17	14*17	12*17	10*17	8*17	6*17	4*17	2*17	17 - - -	1 17 2 17
15 26 16 -										15*15	14*15	12*15	10*15	8*15	6*15	4*15	2*15	16 15 15 15	15 16 - -
13 22 14 24	24 - - -		24*24							14*16	13*13	12*13	10*13	8*13	6*13	4*13	2*13	14 13 13 13	13 14 - -
11 18 12 20	20 - - -			20*20						12*16	12*14	11*11	10*11	8*11	6*11	4*11	2*11	12 11 11 11	11 12 - -
9 14 10 16	16 - - -				16*16					10*16	10*14	10*12	9*9	8* 9	6* 9	4* 9	2* 9	10 9 9 9	9 10 - -
7 10 8 12	12 - - -					12*12				8*16	8*14	8*12	8*10	7*7	6*7	4*7	2*7	8 7 7 7	7 8 - -
5 27 6 27	27 - - -	16*27	14*27	12*27	10*27	8*27	6*27	4*27	2*27	6*16	6*14	6*12	6*10	6*8	5*5	4*5	2*5	6 5 5 5	5 6 - -
3 26 4 26	26 - - -	16*26	14*26	12*26	10*26	8*26	6*26	4*26	2*26	4*16	4*14	4*12	4*10	4*8	4*6	3*3	2*3	4 3 3 3	3 4 - -
1 25 2 25	25 - - -	16*25	14*25	12*25	10*25	8*25	6*25	4*25	2*25	2*16	2*14	2*12	2*10	2*8	2*6	2*4	1*1	2 1 1 1	1 2 - -
		16 - - -	14 24 24 24	12 20 20 20	10 16 16 16	8 12 12 12	6 - - -	4 - - -	2 - - -	16 15 15 15	14 13 13 13	12 11 11 11	10 9 9 9	8 7 7 7	6 5 5 5	4 3 3 3	2 1 1 1		
		Pair 4, Baseline Board 2								Pair 1, Baseline Board 2									

6.4.10.11 One Station per Row/Column (minPack)

For stationPackingAlgorithm=onePerRowColumn (minPack):

- Baseline Boards are configured so that each row/column gets input for one station.
- There are 4 Correlator Chips per baseline on each Baseline Board Pair.
- Each polarization product is configured on a different Correlator Chip.
- Up to 1024 spectral channels per product can be obtained without recirculation.

Using recirculation and/or Baseline Board stacking it is possible to get up to 262,144 spectral channels per product.

Number of Baseline Board Pairs for each subband depends on:

- number of antennas,
- number of bits to correlate,
- number of spectral channels per product,
- number of products,
- recirculation factor,
- product packing mode and
- subset of auto-correlation products.

Figure 6-16 is a conceptual diagram for Baseline Boards configured for a subarray with up to 32 antennas. The diagram shows only the first Baseline Board in each pair. Numbers on the side and bottom are Station IDs.

All the modes supported for stationPacking=maxPack (fourPerRowColum) are also supported in this mode:

4. 4-bit correlation and 7-bit correlation,
5. Dynamic and static recirculation (Baseline Board Stacking) ,
6. ProductPacking maxPack and minPack,
7. Subset of auto-correlation products: halfStationsMaxProd and allStationsMaxProd.

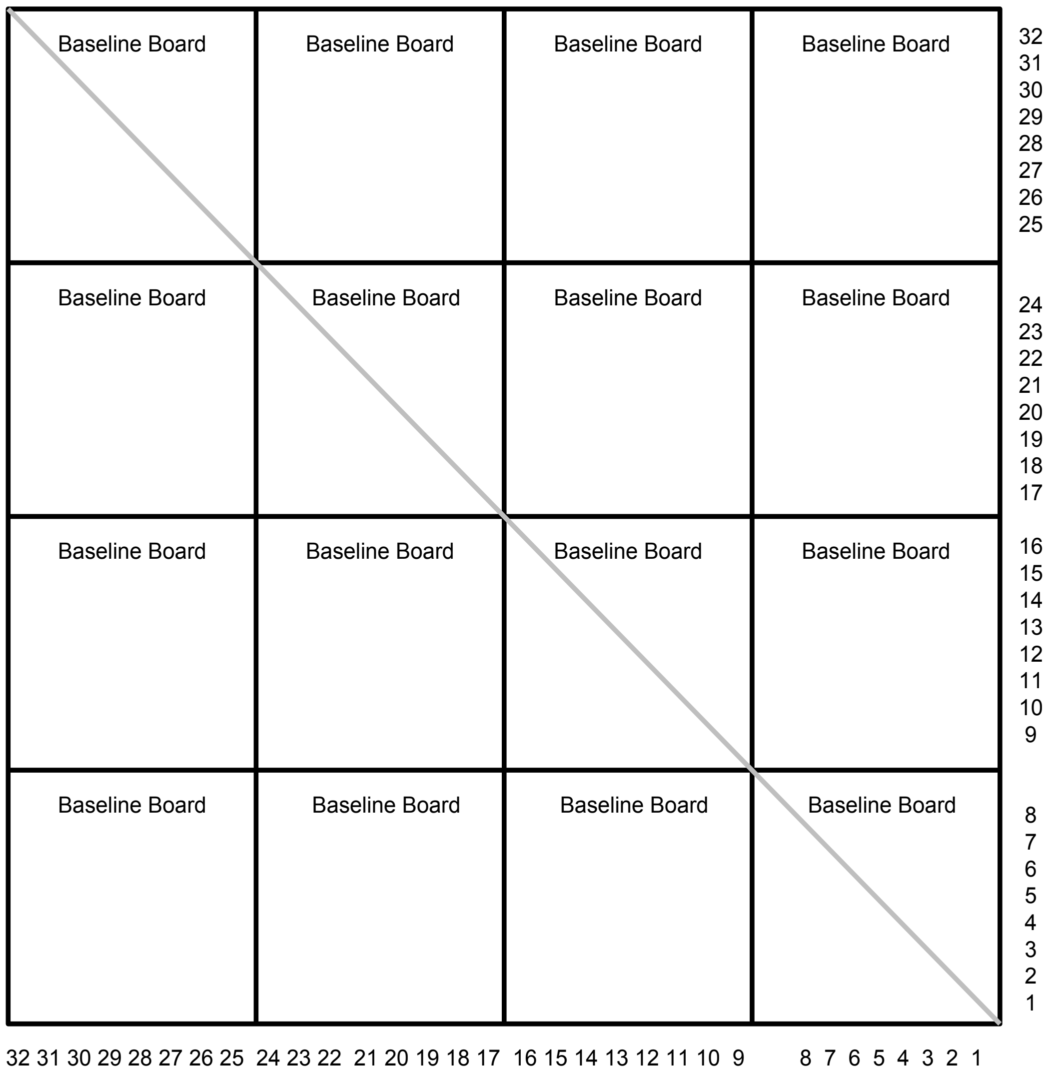
stationPackingAlgorithm=onePerRowColumn

Product Packing

For productPacking=maxPack, in order to minimize number of Baseline Board Pairs used per subband, CM determines the maximum size of the lag chain.

For productPacking=minPack, CM determines the minimum size of the lag chain, i.e. the minimum number of CCCs that can be used per Correlator Chip for a given number of BlB Pairs.

Figure 6-16 Baseline Boards for stationPacking=minPac 32 antennas.



stationPackingAlgorithm=onePerRowColumn

autoCorrAlgorithm=halfStationsMaxProd

In this mode:

- Auto-correlation products are produced for half of the stations, or more precisely for:
  - Every second station starting from the station with the lowest Station ID, or
  - Every second station starting from the station with the second lowest Station ID.
- The maximum number of spectral channels per product is 262,144.
- For polarization products with 16,384 spectral channels or less, phase for recirculation can be either serial or parallel.
- More than 16,834 spectral channels per polarization product can be obtained if phase for recirculation is serial.

Figure 6-17 and Figure 6-18 show how baselines are arranged on the Baseline Boards for a subarray with 16 stations. In this mode, each polarization product is configured on a different Correlator Chip. Color is used to indicate which Correlator Chip produces which product.

Correlator Chips on the diagonal produce auto-correlation products. There are 4 polarization products per baseline. Auto-correlation products A*A, A*B and B*B are configured for every second station, starting from the station with the lowest Station ID. In the same manner as for stationPacking=fourPerRowColumn, one use choose a complementary subset of stations, starting from the second lowest ID, in this example from Station 2.

Explanation regarding the maximum number of spectral channels per product:

- To get more than 16834 spectral channels per product it is necessary to use recirculation.
- For a product with more than 16834 spectral channels, Recirculation Controller can recirculate only two (out of four) pairs of data streams.
- When phase for recirculation is parallel, Recirculation Controller uses two internal data streams for each recirculated input data stream (one for data and the other for phase).
- In order to get all auto-correlation products for a half of the stations, in every second row/column on the 1st Baseline Board in a pair, Correlator Chip on the diagonal is used for an auto-correlation product for a station from the previous (or the next) row/column; in other words: input from two stations must be recirculated. If phase for recirculation is serial, Recirculation Controller has to recirculate two pairs of data streams; up to 262,144 spectral channels per product can be obtained in this mode. If phase for recirculation is parallel, Recirculation Controller has to recirculate data and phase for two input stations, i.e. four pairs of data streams; and the maximum number of spectral channels per product is 16834.

Number of Baseline Boards for stationPacking=onePerRowColumn halfStationsMaxProd

4polarization products, up to 1024 spectral channels per product

Number of Baseline Board Pairs on the diagonal:
```
dNumBlbPairs = (numStations / 8)
if ((numStations Modulo 8) > 0) dNumBlbPairs++
```

Number of Baseline Boards for baselines only:
```
For (n=1; n<dNumBlbPairs; n++) {
    numRows = numProducts * (numStations-(8*n))
    numBLBs = numRows/8
    if ((numRows Modulo 8) > 0) numBlbs++
    bNumBlbs = bNumBlbs + numBlbs
}

totalNumBlbPairs=dNumBlbPairs+(bNumBlbs/2)+(bNumBlbs Mod 2)
```

2pp or 1pp, up to 1024 spectral channels per product

Number of Baseline Boards on the diagonal:
```
dNumBlbs = (numStations / 8)
if ((numStations Modulo 8) > 0) dNumBlbs++
```

Number of Baseline Boards for baselines only:
```
For (n=1; n<dNumBlbs; n++) {
    numRows = numProducts * (numStations-(8*n))
    numBLBs = numRows/8
    if ((numRows Modulo 8) > 0) numBlbs++
    bNumBlbs = bNumBlbs + numBlbs
}

totalNumBlbs= (dNumBlbs+ bNumBlbs)
totalNumBlbPairs = (totalNumBlbs/2)+(totalNumBlbs Mod 2)
```

For polarization products that require more than 16 CCCs, calculate number of lag chain segments and multiply the number of Baseline Boards needed for one segment with the number of lag chain segments.

The usual constrains apply for the size of lag chain segments:

- All segments must be of the same size,
- Minimum segment size is 1 CCC.
- Maximum segment size is 16 CCC.

Figure 6-17 The 1st Baseline Board for 4 Baseline Board Pairs for stationPacking=onePerRowColumn (minPack), autoCorrs=halfStationsMaxProd, 16 stations. In this configuration each Correlator Chip produces one polarization product for one baseline.

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 1 (upper left) — Pair 3, Baseline Board 1 (upper right) — Pair 4, Baseline Board 1 (lower left) — Pair 1, Baseline Board 1 (lower right)

RC output X/Y	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	RC output X/Y
16 15 15 15	15*15	15*16	14*16	13*16	12*16	11*16	10*16	9*16	8*16	7*16	6*16	5*16	4*16	3*16	2*16	1*16	8 16 16 16
15 16 16 16	15*16	15*15	14*15	13*15	12*15	11*15	10*15	9*15	8*15	7*15	6*15	5*15	4*15	3*15	2*15	1*15	7 15 15 15
14 13 13 13	14*16	14*15	13*13	13*14	12*14	11*14	10*14	9*14	8*14	7*14	6*14	5*14	4*14	3*14	2*14	1*14	6 14 14 14
13 14 14 14	13*16	13*15	13*14	13*13	12*13	11*13	10*13	9*13	8*13	7*13	6*13	5*13	4*13	3*13	2*13	1*13	5 13 13 13
12 11 11 11	12*16	12*15	12*14	12*13	11*11	11*12	10*12	9*12	8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	4 12 12 12
11 12 12 12	11*16	11*15	11*14	11*13	11*12	11*11	10*11	9*11	8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	3 11 11 11
10 9 9 9	10*16	10*15	10*14	10*13	10*12	10*11	9*9	9*10	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10	2 10 10 10
9 10 10 10	9*16	9*15	9*14	9*13	9*12	9*11	9*10	9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9	1 9 9 9
8 16 16 16	8*16	8*15	8*14	8*13	8*12	8*11	8*10	8* 9	7*7	7*8	6*8	5*8	4*8	3*8	2*8	1*8	8 7 7 7
7 15 15 15	7*16	7*15	7*14	7*13	7*12	7*11	7*10	7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7	7 8 8 8
6 14 14 14	6*16	6*15	6*14	6*13	6*12	6*11	6*10	6* 9	6*8	6*7	5*5	5*6	4*6	3*6	2*6	1*6	6 5 5 5
5 13 13 13	5*16	5*15	5*14	5*13	5*12	5*11	5*10	5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5	5 6 6 6
4 12 12 12	4*16	4*15	4*14	4*13	4*12	4*11	4*10	4* 9	4*8	4*7	4*6	4*5	3*3	3*4	2*4	1*4	4 3 3 3
3 11 11 11	3*16	3*15	3*14	3*13	3*12	3*11	3*10	3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3	3 4 4 4
2 10 10 10	2*16	2*15	2*14	2*13	2*12	2*11	2*10	2* 9	2*8	2*7	2*6	2*5	2*4	2*3	1*1	1*2	2 1 1 1
1 9 9 9	1*16	1*15	1*14	1*13	1*12	1*11	1*10	1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1	1 2 2 2
CC Input X	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 6-18 The 2nd Baseline Board for 4 Baseline Board Pairs for stationPacking=onePerRowColumn (minPack), autoCorrs=halfStationsMaxProd, 16 stations. In this configuration each Correlator Chip produces one polarization product for one baseline.

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 2 | Pair 3, Baseline Board 2

RC output X/Y																	RC output X/Y
16 15 15 15		15*16	14*16	13*16	12*16	11*16	10*16	9*16	8*16	7*16	6*16	5*16	4*16	3*16	2*16	1*16	8 16 16 16
15 16 16 16	15*16	15*15	14*15	13*15	12*15	11*15	10*15	9*15	8*15	7*15	6*15	5*15	4*15	3*15	2*15	1*15	7 15 15 15
14 13 13 13	14*16	14*15		13*14	12*14	11*14	10*14	9*14	8*14	7*14	6*14	5*14	4*14	3*14	2*14	1*14	6 14 14 14
13 14 14 14	13*16	13*15	13*14	13*13	12*13	11*13	10*13	9*13	8*13	7*13	6*13	5*13	4*13	3*13	2*13	1*13	5 13 13 13
12 11 11 11	12*16	12*15	12*14	12*13		11*12	10*12	9*12	8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	4 12 12 12
11 12 12 12	11*16	11*15	11*14	11*13	11*12	11*11	10*11	9*11	8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	3 11 11 11
10 9 9 9	10*16	10*15	10*14	10*13	10*12	10*11		9*10	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10	2 10 10 10
9 10 10 10	9*16	9*15	9*14	9*13	9*12	9*11	9*10	9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9	1 9 9 9
8 16 16 16	8*16	8*15	8*14	8*13	8*12	8*11	8*10	8* 9		7*8	6*8	5*8	4*8	3*8	2*8	1*8	8 7 7 7
7 15 15 15	7*16	7*15	7*14	7*13	7*12	7*11	7*10	7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7	7 8 8 8
6 14 14 14	6*16	6*15	6*14	6*13	6*12	6*11	6*10	6* 9	6*8	6*7		5*6	4*6	3*6	2*6	1*6	6 5 5 5
5 13 13 13	5*16	5*15	5*14	5*13	5*12	5*11	5*10	5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5	5 6 6 6
4 12 12 12	4*16	4*15	4*14	4*13	4*12	4*11	4*10	4* 9	4*8	4*7	4*6	4*5		3*4	2*4	1*4	4 3 3 3
3 11 11 11	3*16	3*15	3*14	3*13	3*12	3*11	3*10	3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3	3 4 4 4
2 10 10 10	2*16	2*15	2*14	2*13	2*12	2*11	2*10	2* 9	2*8	2*7	2*6	2*5	2*4	2*3		1*2	2 1 1 1
1 9 9 9	1*16	1*15	1*14	1*13	1*12	1*11	1*10	1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1	1 2 2 2
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Pair 4, Baseline Board 2 | Pair 1, Baseline Board 2

Figure 6-19 and Figure 6-20 show Baseline Boards configured for a subarray that has 9 stations (antennas). This example shows that, depending on the number of antennas, stationPacking=onePerRowColum may result in very inefficient use of Baseline Boards.

Figure 6-19 onePerRowColumn, halfStationsMaxProd, 9 stations, 4 polarization products – 4 Baseline Board Pairs, 1st Baseline Board.

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 1 | Pair 3, Baseline Board 1

						9*9									
							9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
							8* 9	7*7	7*8	6*8	5*8	4*8	3*8	2*8	1*8
							7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7
							6* 9	6*8	6*7	5*5	5*6	4*6	3*6	2*6	1*6
							5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5
							4* 9	4*8	4*7	4*6	4*5	3*3	3*4	2*4	1*4
							3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3
							2* 9	2*8	2*7	2*6	2*5	2*4	2*3	1*1	1*2
							1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1
							9	8	7	6	5	4	3	2	1

Pair 4, Baseline Board 1 | Pair 1, Baseline Board 1

Figure 6-20 onePerRowColumn, halfStationsMaxProd, 9 stations, 4polarization products – 4 Baseline Board Pairs, 2nd Baseline Board.

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 2 **Pair 3, Baseline Board 2**

							9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
							8* 9		7*8	6*8	5*8	4*8	3*8	2*8	1*8
							7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7
							6* 9	6*8	6*7		5*6	4*6	3*6	2*6	1*6
							5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5
							4* 9	4*8	4*7	4*6	4*5		3*4	2*4	1*4
							3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3
							2* 9	2*8	2*7	2*6	2*5	2*4	2*3		1*2
							1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1
							9	8	7	6	5	4	3	2	1

Pair 4, Baseline Board 2 **Pair 1, Baseline Board 2**

Products on Baseline Board Pairs 3 and 4 in Figure 6-19 and Figure 6-20 can be configured on a single Baseline Board. Figure 6-21 shows optimized configuration; all 4 polarization products are configured on the same Baseline Board, which means that 3 Baseline Board Pairs are used for a subarray that has 9 antennas. *This is how CM configures Basline Boards*.

Figure 6-21 3rd Baseline Board Pair for oneStationPerRowColumn, 9 stations, 4pp

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 3, Baseline Board 2								Pair 3, Baseline Board 1							
								8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
								8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
								8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
								8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9

Figure 6-22 3rd Baseline Board Pair for oneStationPerRowColumn, 12 stations, 4pp

Pair 3, Baseline Board 2								Pair 3, Baseline Board 1							
8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10
8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10
8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10
8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10
8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9
8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9

Figure 6-23 is a snapshot of GUI that provides overview of the Baseline Board configuration for a subarray with 9 stations (antennas), single subband, 4 polarization products, 1024 spectral channels. GUI shows Baseline Boards as configured by CM. In this case the whole lag chain can be configured on the same Correlator Chip, no need to use Baseline Board Stacking. CM uses 3 BlB Pairs:

- BlB Pair q1p0 (Quadrant 1, BlB Pair 0) is configured as shown for Baseline Board Pair 1 in Figure 6-19 and Figure 6-20.
- BlB Pair q2p0 is configured as shown for BlB Pair 2 in Figure 6-19 and Figure 6-20. This pair is used for auto-correlation products for station 9. Two auto-correlation products are placed on the 1st BlB, the 3rd product is placed on the second BlB.
- BlB Pair, q3p0 is configured as shown in Figure 6-21. In this case 4 rows are needed to get 4 polarization products. Note that unused 4 rows on the Baseline Board 1 Figure 6-21 cannot be used for anything else. Baseline Board 2 can be used for the 2nd lag chain segment when Baseline Board stacking is used.

Figure 6-23 onePerRowColumn, halfStationsMaxProd, 9 stations, 4pp @ 1024 * recFactor

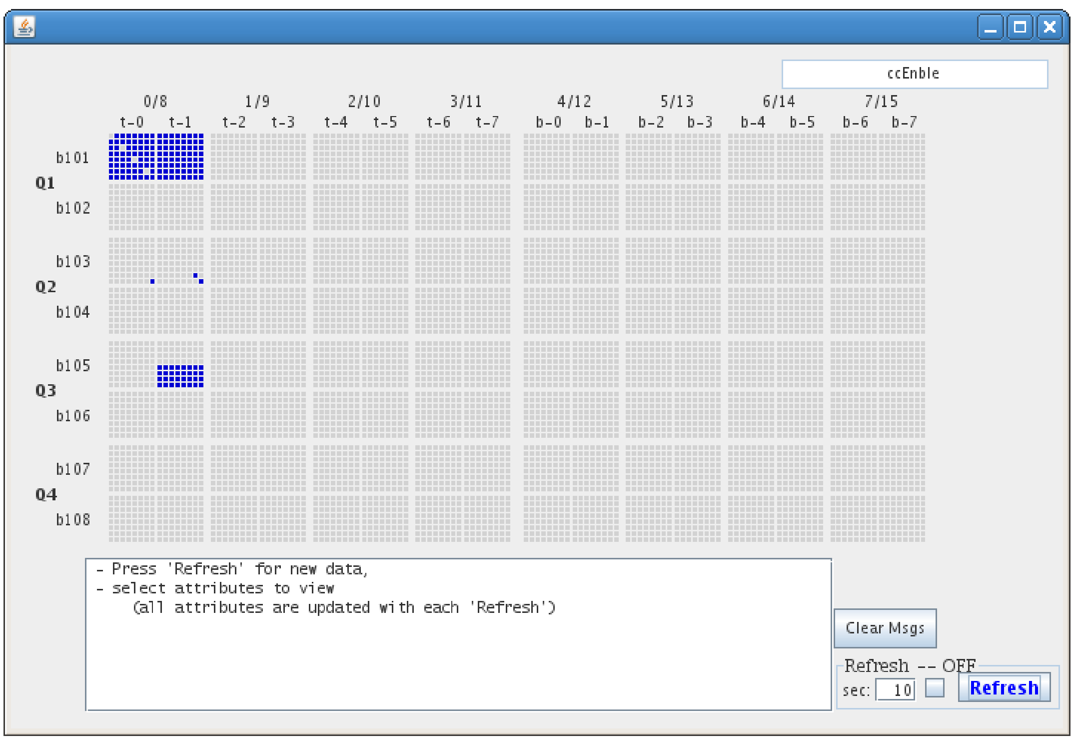


Figure 6-24 is a snapshot of a GUI that provides overview of the Baseline Board configuration for 9 stations, single subband, 4 polarization products, 4096 spectral channels per product. Each polarization product is configured on a different Correlator Chip; 1024 spectral channels can be obtained using single Correlator Chip (more with recirculation). In this case 4096 spectral channels are obtained using Baseline Board stacking (static recirculation). Lag chain is devided in 4 segments and every segment is configured on different Correlator Chip.

Baseline Board Pairs q1p0, q2p0, q3p0 and q1p1 are configured as Baseline Board Pair 1 in Figure 6-19 and Figure 6-20. Each Baseline Board Pair is used for different lag chain segment.

Baseline Board Pair q2p1 is configured as Baseline Board Pair 2 in Figure 6-19 and Figure 6-20. All 4 segments are configured on the same Baseline Board Pair. Two rows/columns are needed for each segment; CM configures all 4 segments on the same BlB Pair.

Baseline Board Pairs q3p2 and q1p2 are configured as Baseline Board 1 in Figure 6-21. Each Baseline Board is used for different segment.

Figure 6-24 onePerRowColumn, halfStationsMaxProd, 9 stations, 4pp @ 4096 *recFactor; Example of Baseline Boards stacking – 4 lag chain segments.

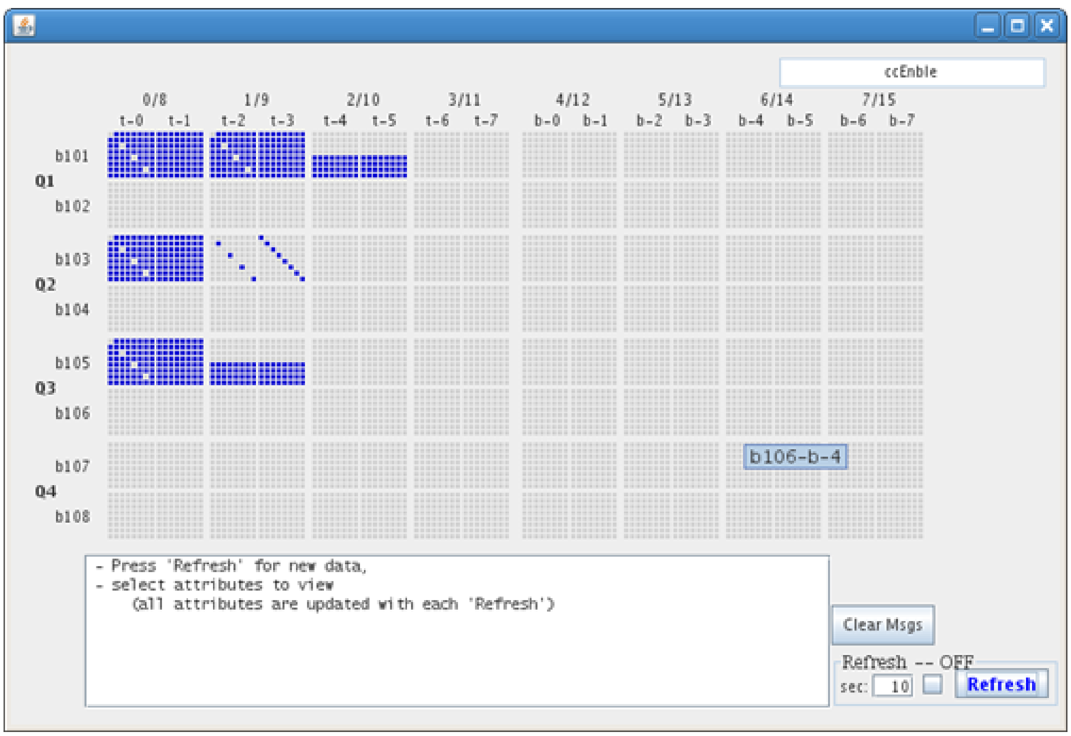


Figure 6-25 to Figure 6-30 are GUI screenshots, they are examples of Baseline Board configuration for stationPacking=minPack, halfStationsMaxProd.

Figure 6-25 oneStationPerRowColumn, halfStationsMaxProd, 27 stations, 4pp @ 1024 * recFactor

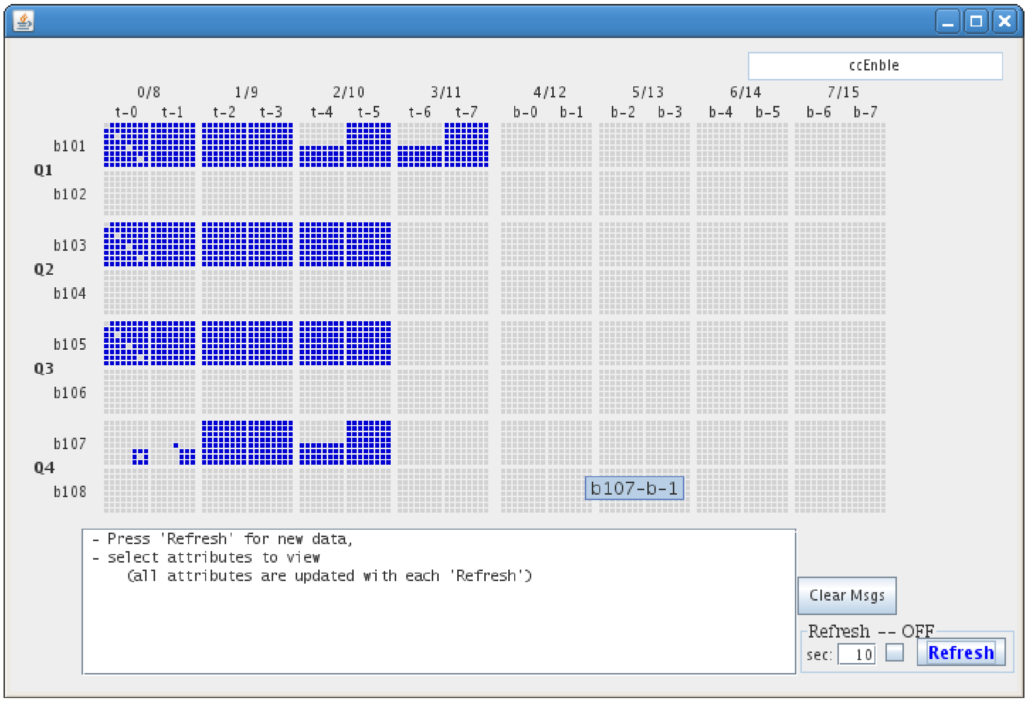


Figure 6-26 oneStationPerRowColumn, halfStationsMaxProd, 27 stations, 4pp@4096 * recFactor. Example of Baseline Board stacking – 4 lag chain segments.

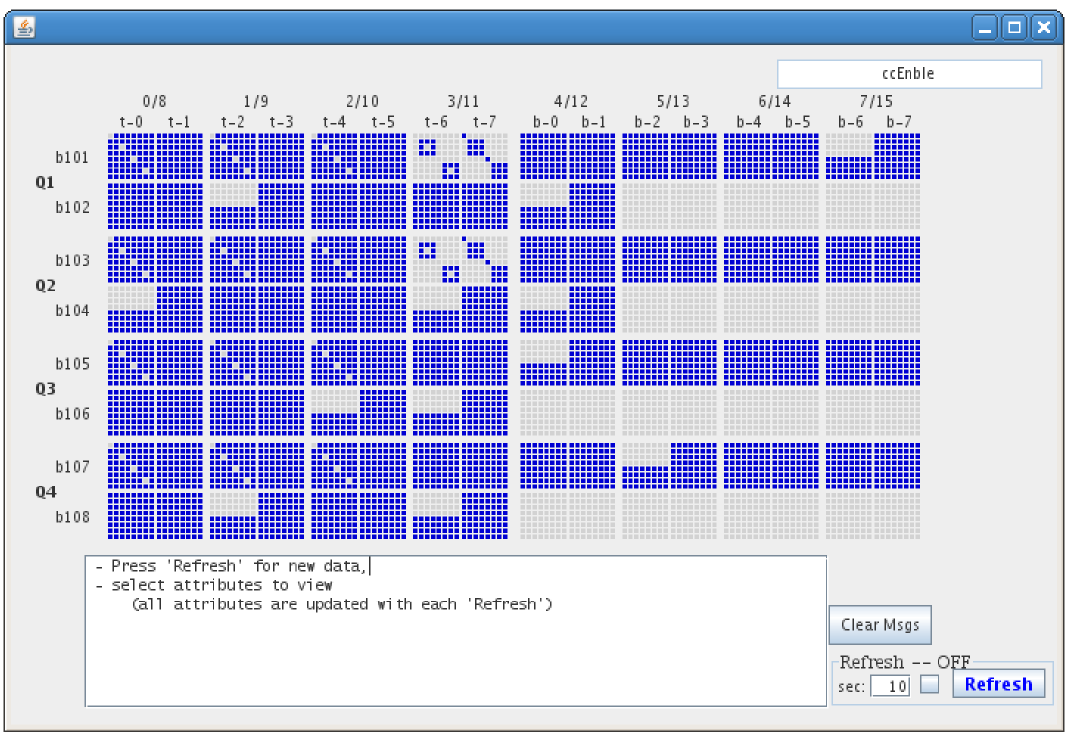


Figure 6-27 oneStationPerRowColumn, halfStationsMaxProd, 9 stations, 2pp@1024* recFactor

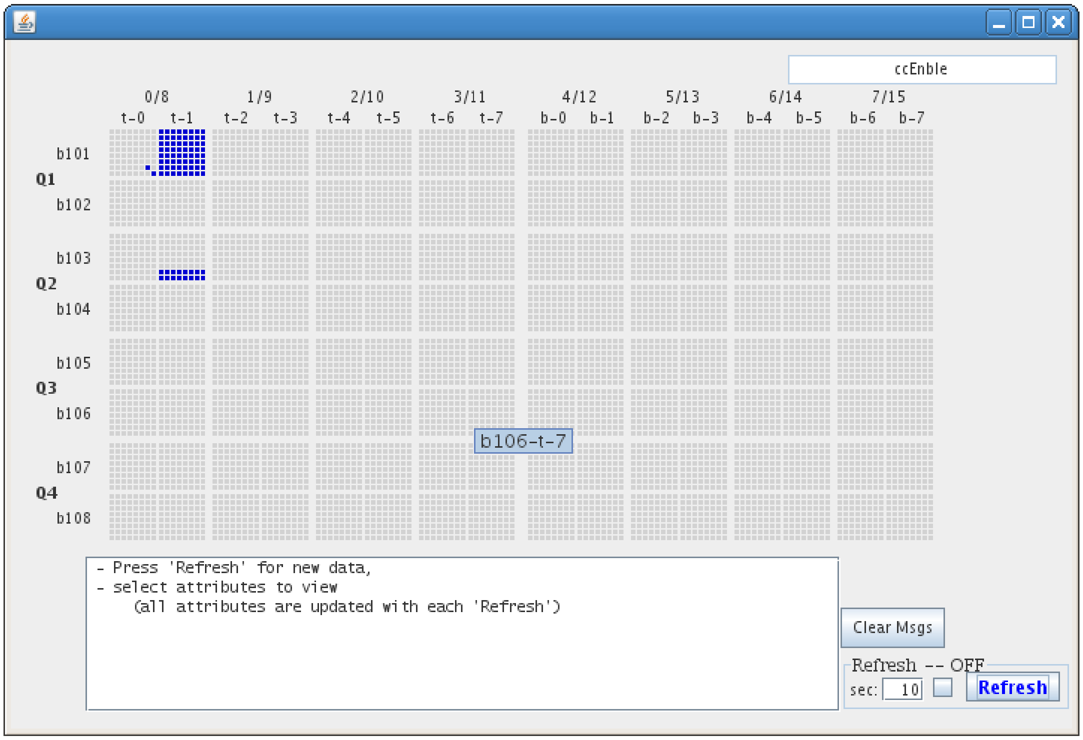


Figure 6-28 oneStationPerRowColumn, halfStationsMaxProd, 9 stations, 2pp @ 5120 * recFactor. Example of Baseline Board stacking – 5 lag chain segments.

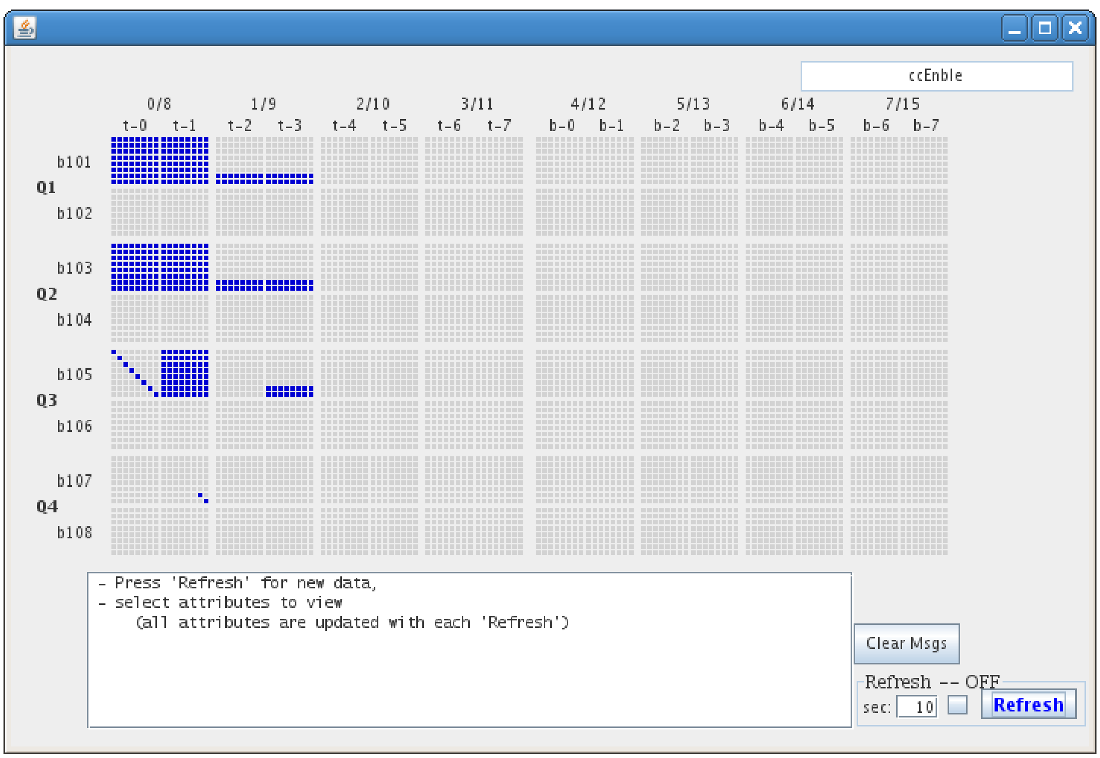


Figure 6-29 9 oneStationPerRowColumn, halfStationsMaxProd, 9 stations, 1pp@2048 * recFactor. Example of Baseline Board stacking – 2 lag chain segments.

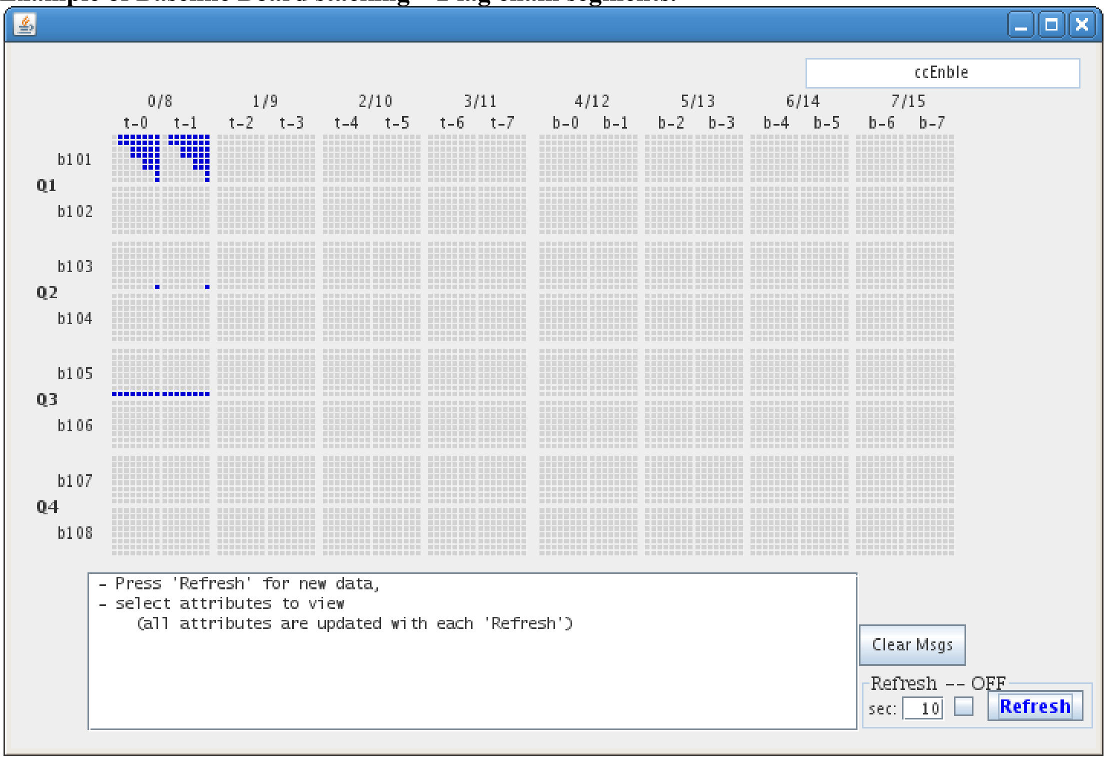
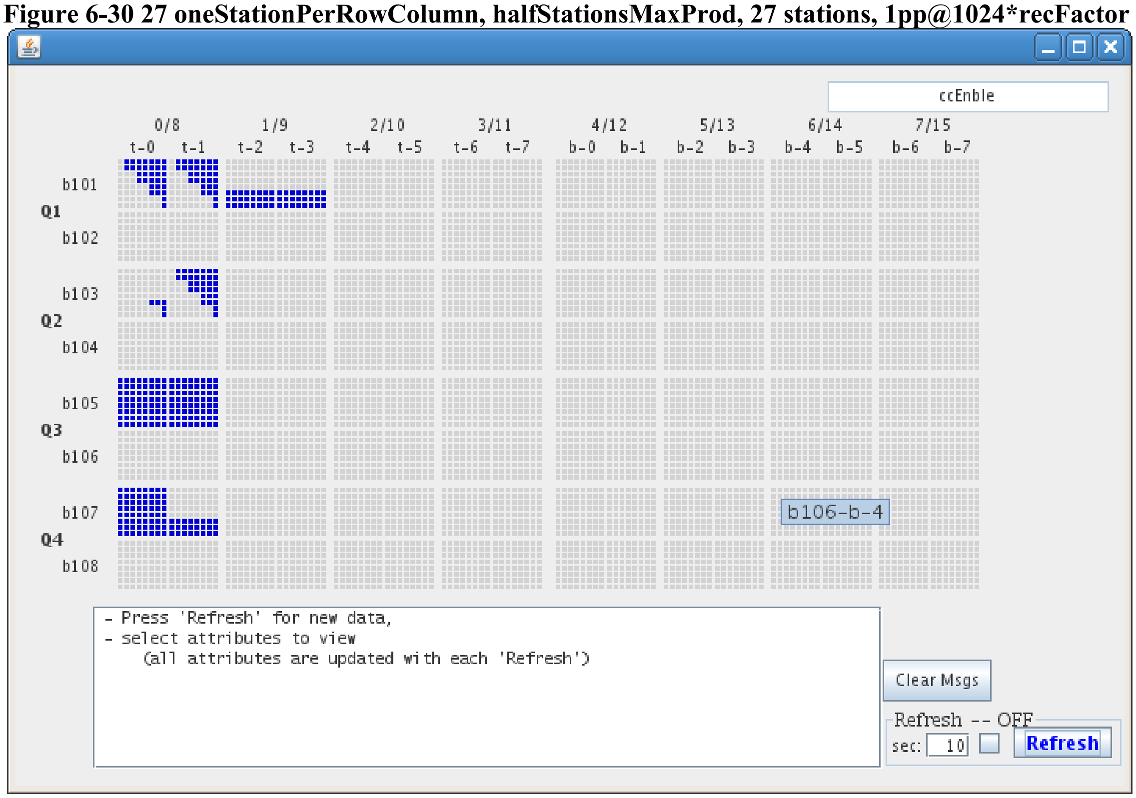


Figure 6-30 27 oneStationPerRowColumn, halfStationsMaxProd, 27 stations, 1pp@1024*recFactor



stationPackingAlgorithm=onePerRowColumn

autoCorrAlgorithm=allStationsMaxProd

All auto-correlation products for all stations in a subarray.

In this mode, Baseline Board Pairs on the diagonal are configured as follows:

- the 1st BlB in a pair produces auto-correlation product A*A for all the stations,
- the 2nd BlB in a pair produces auto-correlation product B*B for all the stations.
- Additional Baseline Board Pair(s) are used to get auto-correlation product A*B for all the stations.

Boards that are not on the diagonal are configured in the same manner as for 'halfStationsMaxProd'.

See Figure 6-31, Figure 6-32 and Figure 6-33 for an example of 16 station subarray.

In this mode:

1. **The maximum number of spectral channels per product is 262,144.**
2. **Phase for recirculation can be either serial or parallel for any number of spectral channels per product (up to, and including, 262,144).**

Number of Baseline Boards for stationPacking=onePerRowColumn, allStationsMaxProd

4pp , up to 1024 spectral channels per product (without recirculation)

Number of Baseline Board Pairs on the diagonal:

```
numRows = numStations +(numStations/2)+(numStations modulo 2)
dNumBlbPairs = (numRows / 8)
if ((numRows modulo 8) > 0) dNumBlbPairs++
```

Number of Baseline Boards for baselines only:

```
For (n=1; n<dNumBlbPairs; n++) {
    numRows = numProducts * (numStations-(8*n))
    numBLBs = numRows/8
    if ((numRows Modulo 8) > 0) numBlbs++
    bNumBlbs = bNumBlbs + numBlbs
}
totalNumBlbPairs=dNumBlbPairs +(bNumBlbs/2)+(bNumBlbs modulo 2)
```

2pp, up to 1024 spectral channels per product (without recirculation)

Number of Baseline Board Pairs on the diagonal:

```
numRows = numStations
dNumBlbPairs = (numRows / 8)
if ((numRows modulo 8) > 0) dNumBlbPairs++
```

Number of Baseline Boards for baselines only:

```
For (n=1; n<dNumBlbPairs; n++) {
    numRows = numProducts * (numStations-(8*n))
    numBLBs = numRows/8
    if ((numRows Modulo 8) > 0) numBlbs++
    bNumBlbs = bNumBlbs + numBlbs
}
totalNumBlbPairs=dNumBlbPairs +(bNumBlbs/2)+(bNumBlbs modulo 2)
```

1pp, up to 1024 spectral channels per product (without recirculation)

Number of Baseline Boards on the diagonal:

```
dNumBlbs = (numStations / 8)
if ((numStations Modulo 8) > 0) dNumBlbs++
```

Number of Baseline Boards for baselines only:

```
For (n=1; n<dNumBlbs; n++) {
    numRows = numProducts * (numStations-(8*n))
    numBlbs = numRows/8
    if ((numRows modulo 8) > 0) numBlbs++
    bNumBlbs = bNumBlbs + numBlbs
}
totalNumBlbs= (dNumBlbs+ bNumBlbs)
totalNumBlbPairs = (totalNumBlbs/2)+(totalNumBlbs modulo 2)
```

Figure 6-31 oneStationPerRowColumn, allStationsMaxProd, 16 stations, 4pp, 1st Baseline Board

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 1 **Pair 3, Baseline Board 1**

16*16	15*16	14*16	13*16	12*16	11*16	10*16	9*16	8*16	7*16	6*16	5*16	4*16	3*16	2*16	1*16	16
15*16	15*15	14*15	13*15	12*15	11*15	10*15	9*15	8*15	7*15	6*15	5*15	4*15	3*15	2*15	1*15	15
14*16	14*15	14*14	13*14	12*14	11*14	10*14	9*14	8*14	7*14	6*14	5*14	4*14	3*14	2*14	1*14	14
13*16	13*15	13*14	13*13	12*13	11*13	10*13	9*13	8*13	7*13	6*13	5*13	4*13	3*13	2*13	1*13	13
12*16	12*15	12*14	12*13	12*12	11*12	10*12	9*12	8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	12
11*16	11*15	11*14	11*13	11*12	11*11	10*11	9*11	8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	11
10*16	10*15	10*14	10*13	10*12	10*11	10*10	9*10	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10	10
9*16	9*15	9*14	9*13	9*12	9*11	9*10	9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9	9
8*16	8*15	8*14	8*13	8*12	8*11	8*10	8* 9	8*8	7*8	6*8	5*8	4*8	3*8	2*8	1*8	8
7*16	7*15	7*14	7*13	7*12	7*11	7*10	7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7	7
6*16	6*15	6*14	6*13	6*12	6*11	6*10	6* 9	6*8	6*7	6*6	5*6	4*6	3*6	2*6	1*6	6
5*16	5*15	5*14	5*13	5*12	5*11	5*10	5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5	5
4*16	4*15	4*14	4*13	4*12	4*11	4*10	4* 9	4*8	4*7	4*6	4*5	4*4	3*4	2*4	1*4	4
3*16	3*15	3*14	3*13	3*12	3*11	3*10	3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3	3
2*16	2*15	2*14	2*13	2*12	2*11	2*10	2* 9	2*8	2*7	2*6	2*5	2*4	2*3	2*2	1*2	2
1*16	1*15	1*14	1*13	1*12	1*11	1*10	1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1	1
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Pair 4, Baseline Board 1 **Pair 1, Baseline Board 1**

Figure 6-32 oneStationPerRowColumn, allStationsMaxProd, 16 stations, 2nd Baseline Board

Legend:
Blue - pol.product R*R
Red - pol.product R*L
Green - pol.product L*L
Violet - pol.product L*R

Pair 2, Baseline Board 2 **Pair 3, Baseline Board 2**

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
16*16	15*16	14*16	13*16	12*16	11*16	10*16	9*16	8*16	7*16	6*16	5*16	4*16	3*16	2*16	1*16	16
15*16	15*15	14*15	13*15	12*15	11*15	10*15	9*15	8*15	7*15	6*15	5*15	4*15	3*15	2*15	1*15	15
14*16	14*15	14*14	13*14	12*14	11*14	10*14	9*14	8*14	7*14	6*14	5*14	4*14	3*14	2*14	1*14	14
13*16	13*15	13*14	13*13	12*13	11*13	10*13	9*13	8*13	7*13	6*13	5*13	4*13	3*13	2*13	1*13	13
12*16	12*15	12*14	12*13	12*12	11*12	10*12	9*12	8*12	7*12	6*12	5*12	4*12	3*12	2*12	1*12	12
11*16	11*15	11*14	11*13	11*12	11*11	10*11	9*11	8*11	7*11	6*11	5*11	4*11	3*11	2*11	1*11	11
10*16	10*15	10*14	10*13	10*12	10*11	10*10	9*10	8*10	7*10	6*10	5*10	4*10	3*10	2*10	1*10	10
9*16	9*15	9*14	9*13	9*12	9*11	9*10	9*9	8* 9	7* 9	6* 9	5* 9	4* 9	3* 9	2* 9	1* 9	9
8*16	8*15	8*14	8*13	8*12	8*11	8*10	8* 9	8*8	7*8	6*8	5*8	4*8	3*8	2*8	1*8	8
7*16	7*15	7*14	7*13	7*12	7*11	7*10	7* 9	7*8	7*7	6*7	5*7	4*7	3*7	2*7	1*7	7
6*16	6*15	6*14	6*13	6*12	6*11	6*10	6* 9	6*8	6*7	6*6	5*6	4*6	3*6	2*6	1*6	6
5*16	5*15	5*14	5*13	5*12	5*11	5*10	5* 9	5*8	5*7	5*6	5*5	4*5	3*5	2*5	1*5	5
4*16	4*15	4*14	4*13	4*12	4*11	4*10	4* 9	4*8	4*7	4*6	4*5	4*4	3*4	2*4	1*4	4
3*16	3*15	3*14	3*13	3*12	3*11	3*10	3* 9	3*8	3*7	3*6	3*5	3*4	3*3	2*3	1*3	3
2*16	2*15	2*14	2*13	2*12	2*11	2*10	2* 9	2*8	2*7	2*6	2*5	2*4	2*3	2*2	1*2	2
1*16	1*15	1*14	1*13	1*12	1*11	1*10	1* 9	1*8	1*7	1*6	1*5	1*4	1*3	1*2	1*1	1

Pair 4, Baseline Board 2 **Pair 1, Baseline Board 2**

Figure 6-33 oneStationPerRowColumn, allStationsMaxProd, 16 stations, 4pp, 5th BIB Pair – auto-correlation product R*L for all the stations.

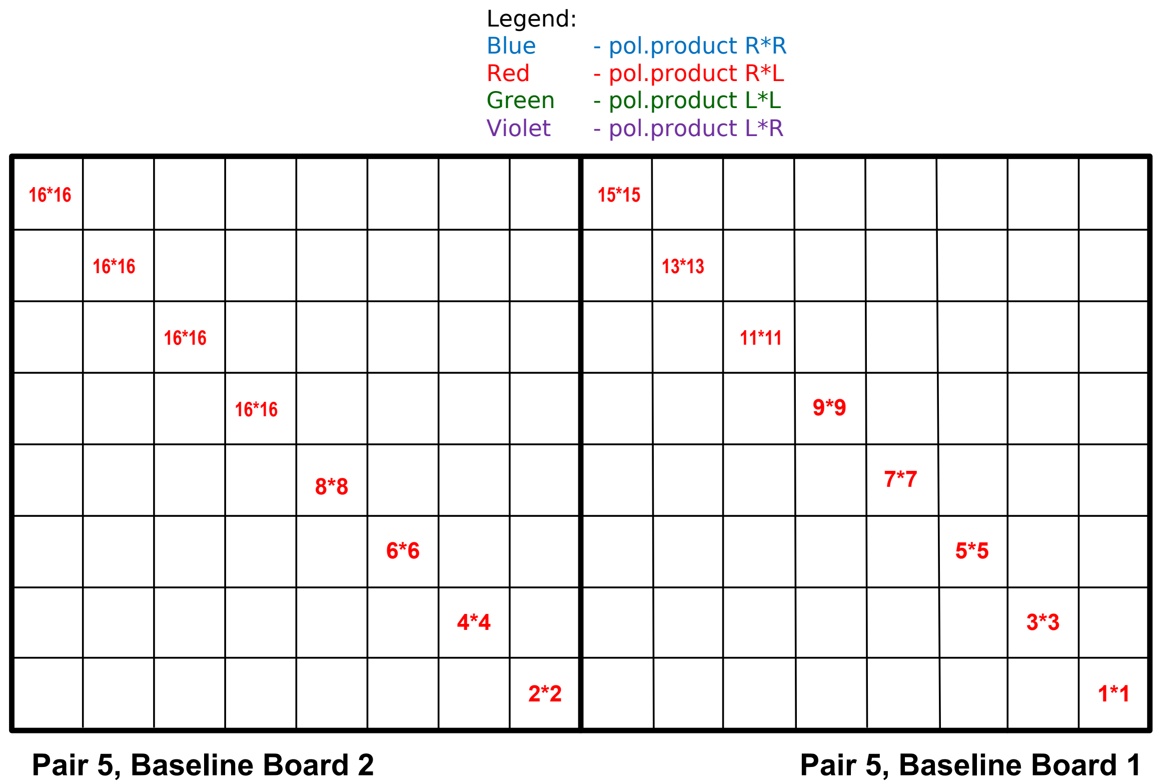


Figure 6-34 to Figure 6-39 are GUI screenshots; they are examples of the Baseline Board configuration for stationPacking=onePerRowColumn, allStationsMaxProd.
All except the last one (Figure 6-39) are examples of configuration where the whole lag chain for each product can be configured on a single Correlator Chip (no Baseline Board Stacking).

Figure 6-34 oneStationPerRowColumn, allStationsMaxProd, 4 stations, 4pp@1024*recFactor

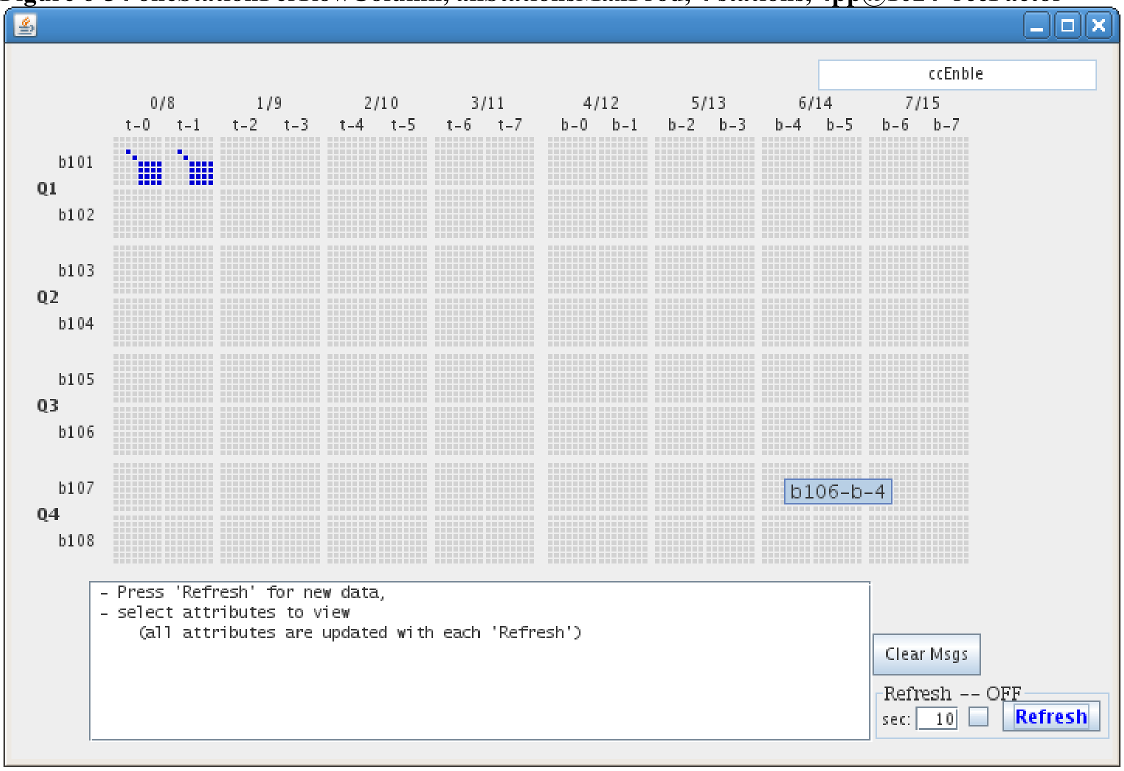


Figure 6-35 oneStationPerRowColumn, allStationsMaxProd, 9 stations, 4pp@1024*recFactor

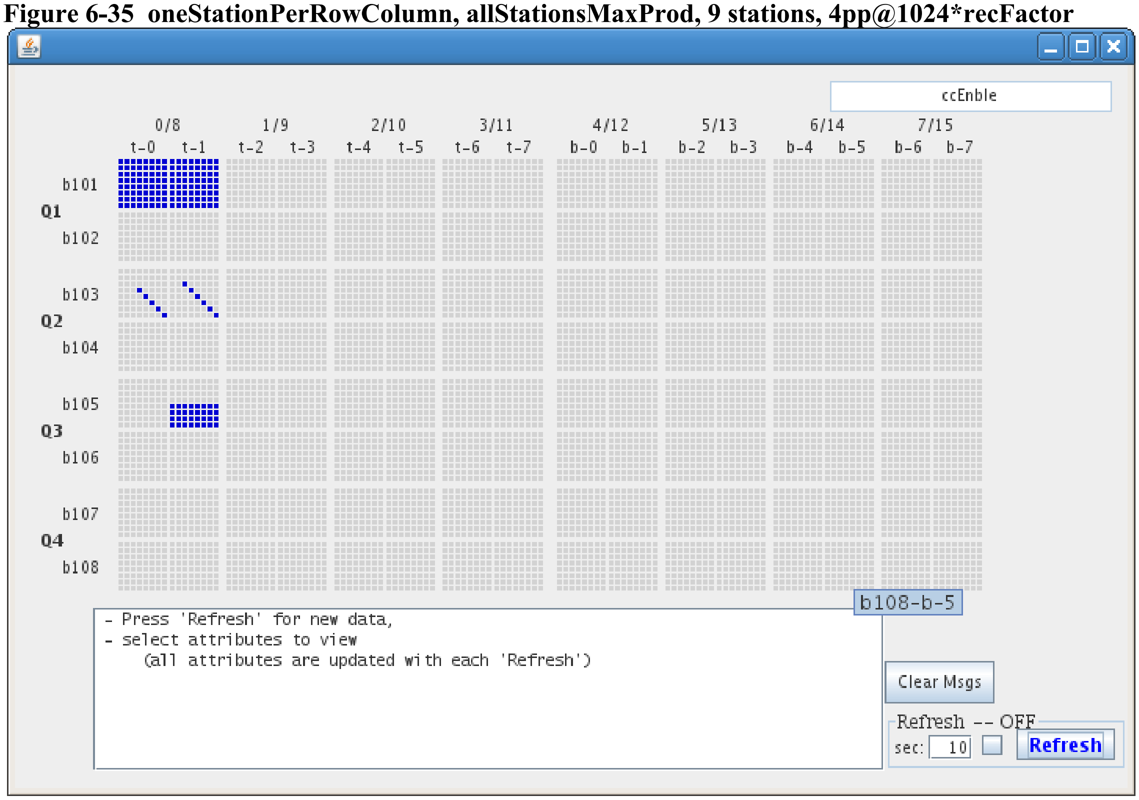


Figure 6-36 oneStationPerRowColumn, allStationsMaxProd, 9 stations, 2pp@1024*recFactor

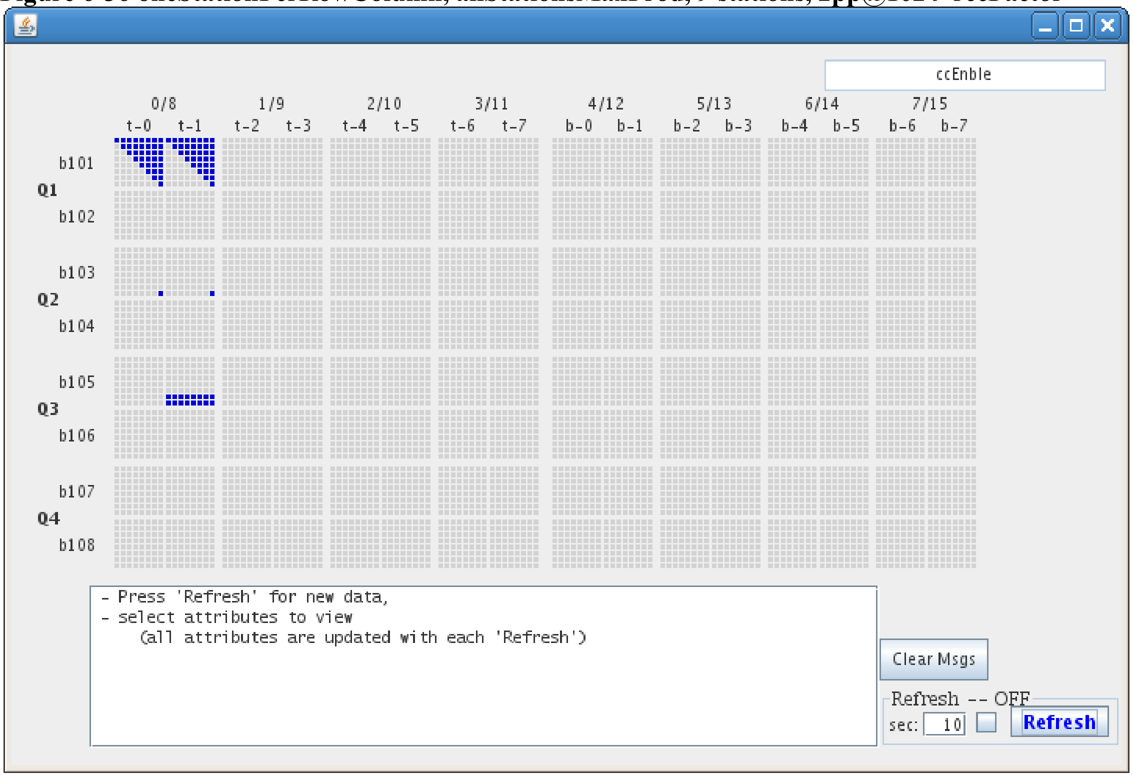


Figure 6-37 oneStationPerRowColumn, allStationsMaxProd, 9 stations, 1pp@1024*recFactor

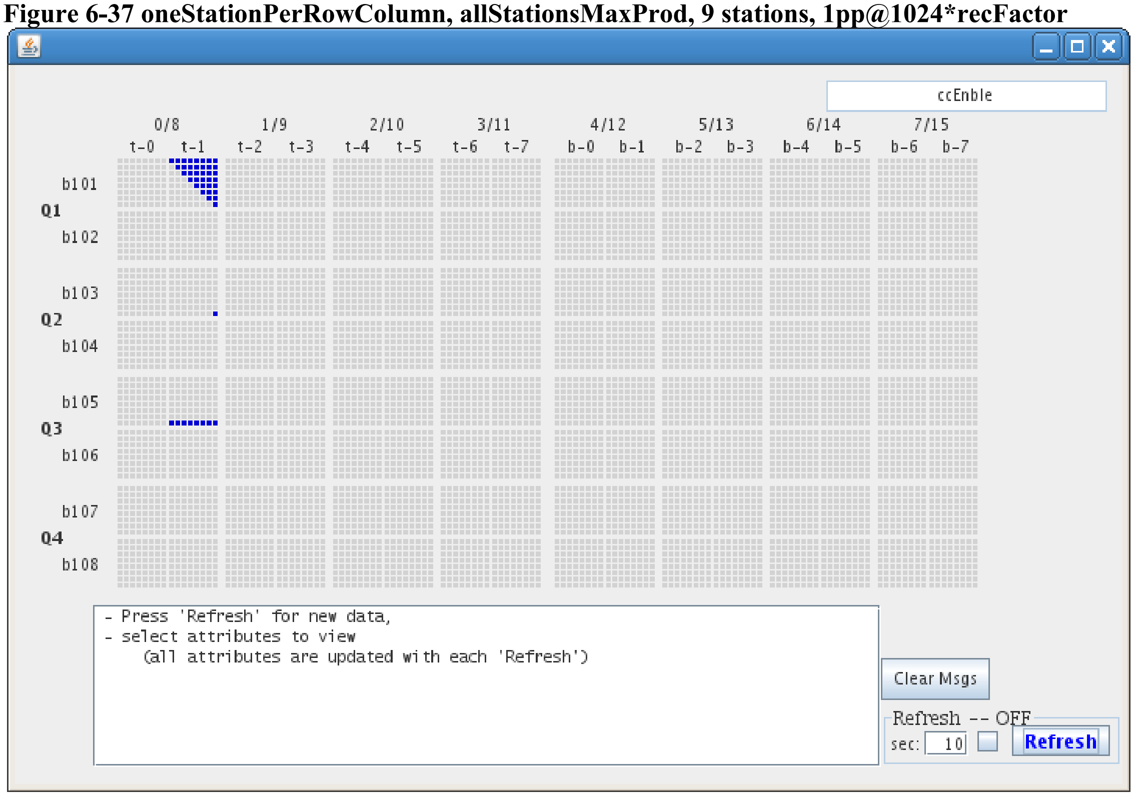


Figure 6-38 oneStationPerRowColumn, allStationsMaxProd, 27 stations, 4pp@1024*recFactor

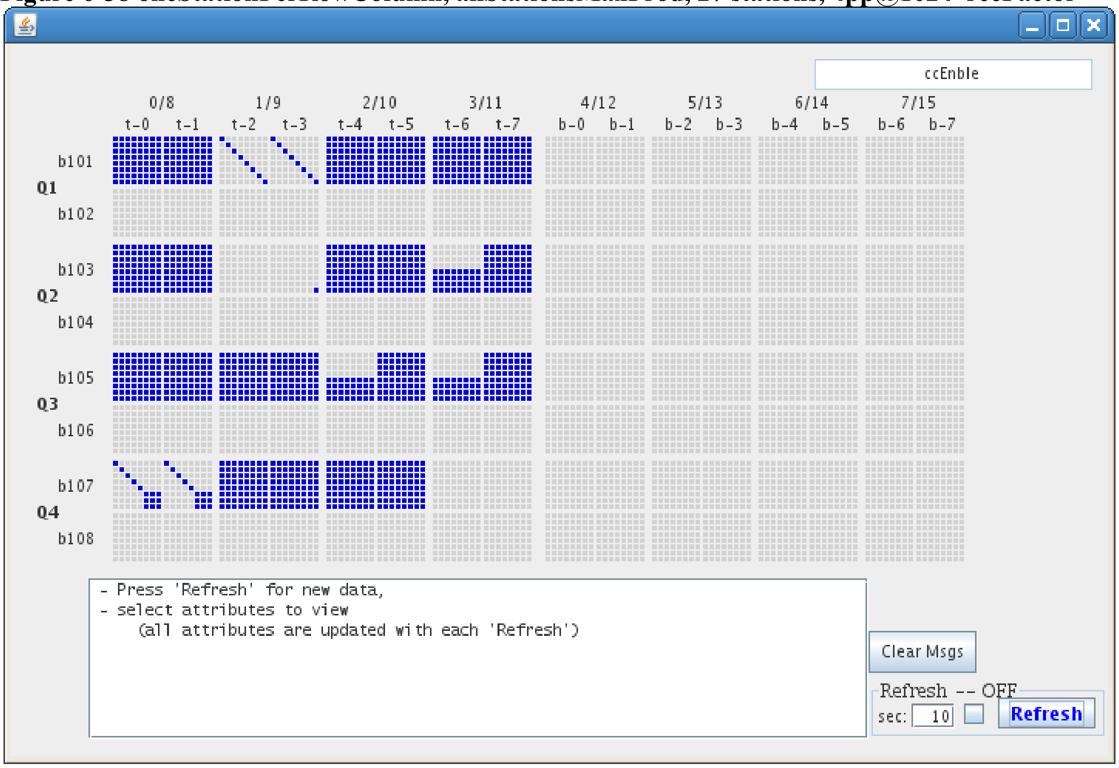
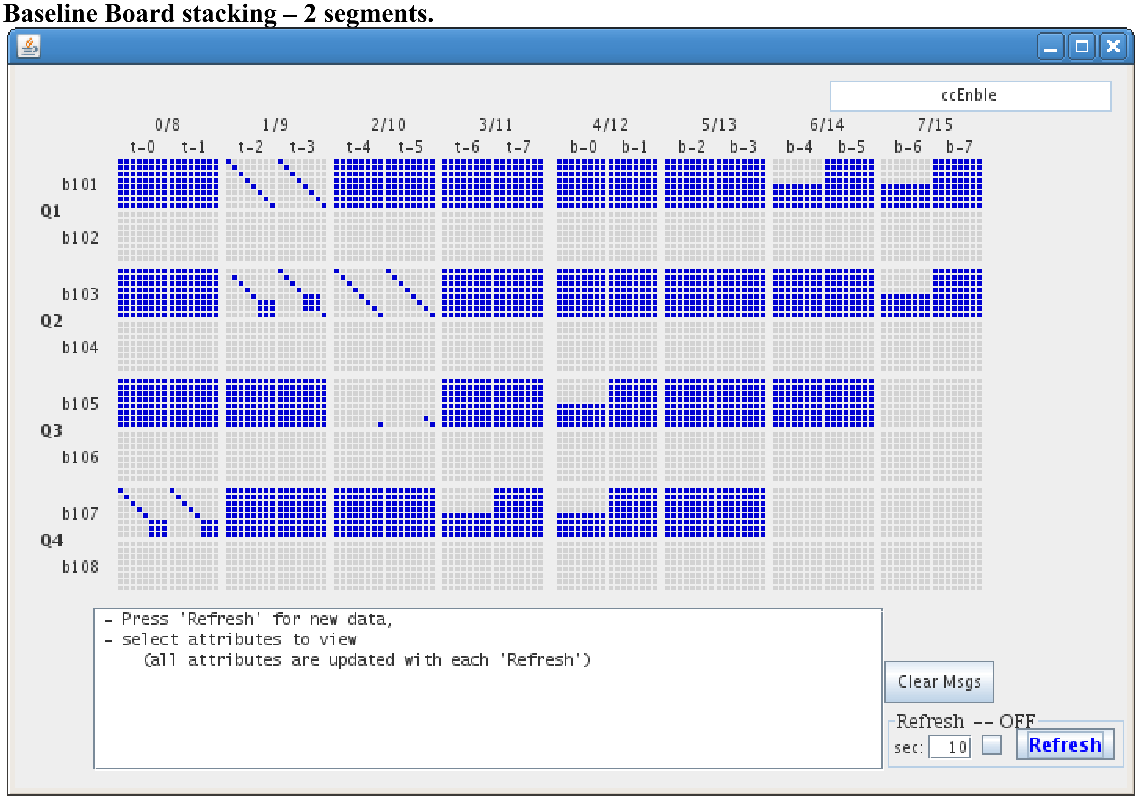


Figure 6-39 oneStationPerRowColumn, allStationsMaxProd, 27 stations, 4pp@2048*recFactor, Baseline Board stacking – 2 segments.



6.4.11 Correlating Antennas with Different Initial Qunatization

EVLA antennas have 3-bit samplers and 8-bit samplers. A subarray may consist of antennas with different samplers (initial quantization).

Input from an EVLA antenna with 3-bit samplers consists of 4 polarized pairs. See Station 2 in Figure 6-40 for an example of 4 Station Boards that process input from an EVLA antenna with 3-bit samplers. There are 4 Baseband Pairs:

- A1C1 - BBID 0/1,
- A2C2 - BBID 2/3,
- B1D1 - BBID 4/5,
- B2D2 - BBID 6/7.

Station 1 in the same picture gets input from an EVLA antenna with 8-bit samplers. It has 2 polarized pairs:

1. AC – BBID 0/1 and 2/3
2. BD – BBID 4/5 and 6/7

Input for Station Boards that belongs to Station 1 is a single Baseband (not polarized pair). Since a Station Board has two filter banks, the same input can be forwarded to both filter banks, which means that there can be up to 32 subbands per Baseband. Subbands are identified using three IDs: Station ID, Baseband ID and Subband ID. A different BBID is assigned to each filter bank. Note that BB0 and BB2 in Figure 6-40 get the same input data. In this example, BBIDs for Station 1 are not assigned sequentially, as for Station 2. In order to get correlation products for Stations 1 and 2 (1*2), BBIDs for Station 1 must be assigned so that Baseband 0 and 1 are a polarized pair (as for Station 2).

To correlate input from antennas with 3-bit samplers and 8-bit samplers BBIDs must be assigned so that basebands (and subbands) to be correlated have identical IDs. It is not possible to correlate Baseband pair 0/1 with Baseband pair 0/2. To fulfil this requirement, use element *stationHw* to assign BBIDs accordingly for antennas with 3-bit and 8-bit samplers.

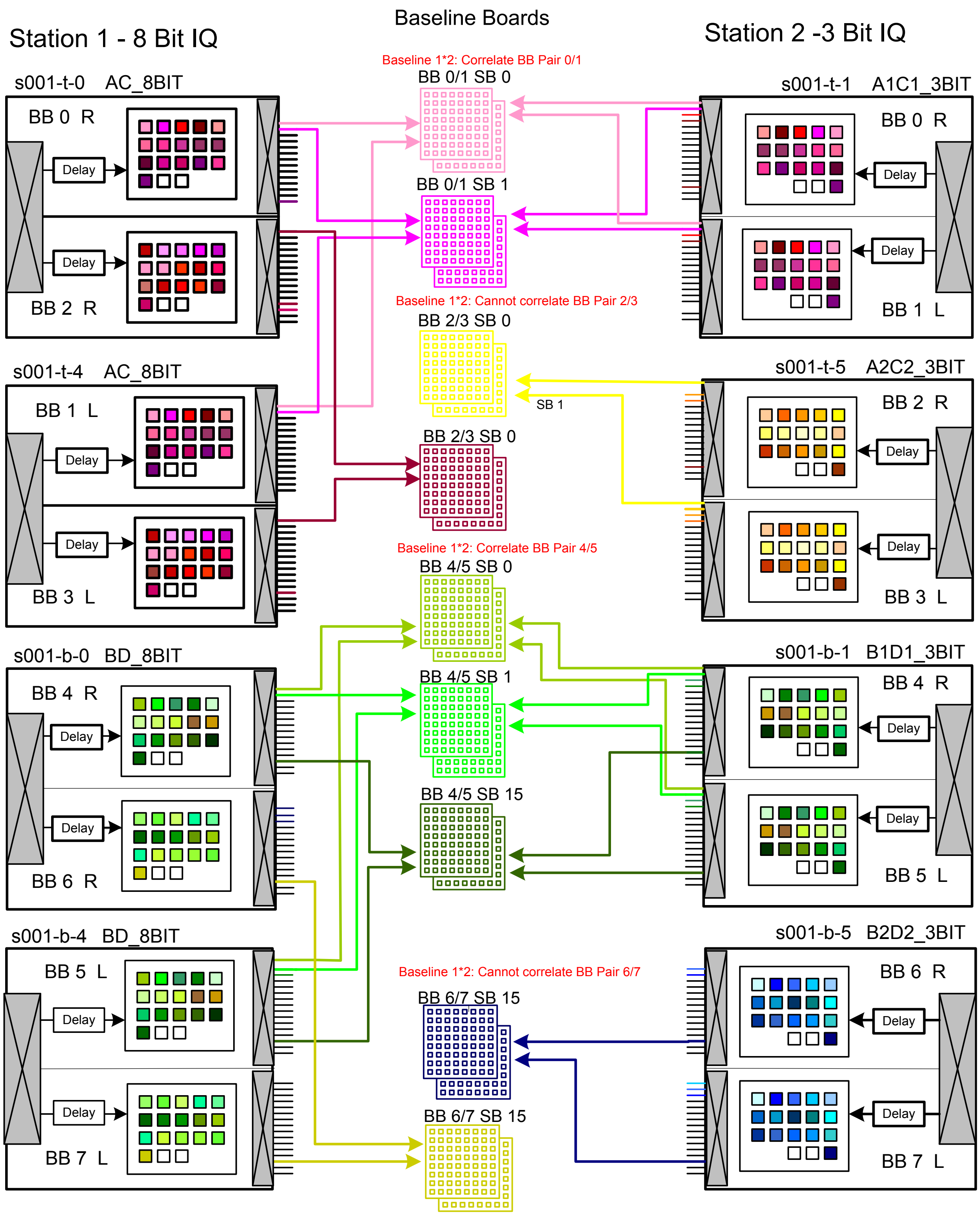
XML Example 6-19 is an example of the VCI Request for a subarray that consists of two antennas: 1 and 2. Antenna 1 has 8-bit samplers while Antenna 2 has 3-bit samplers. There are two instances of the element *stationInputOutput*, one for each type of antenna. Note that subband parameters and products must be identical in both.

Additional antenna (Antenna *n*) could be added to the subarray by adding

```
<widar:station sid="n"/>
```

to either instance of the element *stationInputOutput*.

Figure 6-40 Correlating input from 3-bit and 8-bit antennas



XML Example 6-19 Correlating 3-bit and 8-bit antennas


```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<widar:vciRequest desc="Correlate_antennas_with_3bit_and_8_bit_samplers"
                  timeStamp="2009-09-11T11:16:12.328-07:00" msgId="297"
                 xmlns:widar="http://www.nrc.ca/namespaces/widar">

<widar:stationHw sid="1" name="Station1-8bitIQ" activationId="3IQ*8IQ"  action="add" >
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-0" dataPath="0"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-0" dataPath="1"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-4" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-4" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-0" dataPath="0"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-0" dataPath="1"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-4" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-4" dataPath="1"/>
    <widar:antenna type="EVLA" id="Antenna_1" name="Antenna1"/>
</widar:stationHw>
<widar:stationHw sid="2" name="Station2-3bitIQ" activationId="3IQ*8IQ" action="add" >
    <widar:baseBandHw bbid="0" stationBoardMlid="s001-t-1" dataPath="0"/>
    <widar:baseBandHw bbid="1" stationBoardMlid="s001-t-1" dataPath="1"/>
    <widar:baseBandHw bbid="2" stationBoardMlid="s001-t-5" dataPath="0"/>
    <widar:baseBandHw bbid="3" stationBoardMlid="s001-t-5" dataPath="1"/>
    <widar:baseBandHw bbid="4" stationBoardMlid="s001-b-1" dataPath="0"/>
    <widar:baseBandHw bbid="5" stationBoardMlid="s001-b-1" dataPath="1"/>
    <widar:baseBandHw bbid="6" stationBoardMlid="s001-b-5" dataPath="0"/>
    <widar:baseBandHw bbid="7" stationBoardMlid="s001-b-5" dataPath="1"/>
    <widar:antenna type="EVLA" id="Antenna_2" name="Antenna2"/>
</widar:stationHw>

<widar:subArray configId="3bit8bitCorr" activationId="3IQ*8IQ" action="create" >

    <widar:stationInputOutput sid="list">
        <widar:station sid="1"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                        bbid="0"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                        bbid="2"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L"
                        bbid="1"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L"
                        bbid="3"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                        bbid="4"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                        bbid="6"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L"
                        bbid="5"/>
        <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L"
                        bbid="7"/>
```

```
    <widar:baseBand inQuant="8" bw="1024000000"
                         swbbName="AC_8BIT" bbB="1" bbA="0">

        <widar:subBand sbid="1" swIndex="1"
                              centralFreq="64000000" bw="128000000" rqNumBits="4">
            <widar:polProducts>
                <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
                <widar:blbProdIntegration recirculation="1"
                      minIntegTime="250" ltaIntegFactor="4000" ccIntegFactor="1" />
                <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="1"/>
                <widar:stationPacking  algorithm="maxPack"/>
                <widar:productPacking algorithm="maxPack"/>
                <widar:autoCorrSubset algorithm="crossCorrOnly"/>
            </widar:polProducts>
        </widar:subBand>

    </widar:baseBand>
</widar:stationInputOutput>

<widar:stationInputOutput sid="list">
    <widar:station sid="2"/>
    <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                    bbid="0"/>
    <widar:bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L"
                    bbid="1"/>
    <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                    bbid="2"/>
    <widar:bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L"
                    bbid="3"/>
    <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                    bbid="4"/>
    <widar:bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L"
                     bbid="5"/>
    <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R"
                     bbid="6"/>
    <widar:bbParams sourceType="FORM" sourceId="1" sideband="lower" polarization="L"
                    bbid="7"/>

    <widar:baseBand inQuant="3" bw="2048000000"
                         swbbName="A1C1_3BIT" bbB="1"  bbA="0">

            <widar:subBand sbid="1" swIndex="1"
                              centralFreq="64000000" bw="128000000" rqNumBits="4">
            <widar:polProducts>
                <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
```

```
                    <widar:blbProdIntegration recirculation="1"
                            minIntegTime="250" ltaIntegFactor="4000" ccIntegFactor="1" />
                    <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="1"/>
                    <widar:stationPacking  algorithm="maxPack"/>
                    <widar:productPacking algorithm="maxPack"/>
                    <widar:autoCorrSubset algorithm="crossCorrOnly"/>
                </widar:polProducts>
            </widar:subBand>

        </widar:baseBand>

    </widar:stationInputOutput>

</widar:subArray>

<widar:activationTrigger  activationId="3IQ*8IQ"  activationTime="2010-09-11T16:56:00"/>

</widar:vciRequest>
```

6.4.12 Inter-Frame Delay

Frames generated on the Baseline Board Long Term Accumulators (LTAs) are transmitted via Ethernet port SPF1. All 64 LTAs transmit frames via the same SPF1 port. Usually more than one Baseline Board sends frames to the same destination. To minimazie loss of data due to retransmissions, Ethernet FPGA allows user to insert a delay between frame transmissions. Furthermore, to avoid situation where all Baseline Boards that transmit to the same destination are synchronized (i.e. attempt transmission at the same time), inter-frame delay may be randomized, so that the actual delay is randomly choosen on each Baseline Baord independently as a number in the range ±interFrameDelay. For detailed description of the Ethernet FPFGA functionality please refer to the GBit Ethernet Chip V2 RFS [19].

Inter-frame delay can be specified in the VCI message for each subband of each subarray. If not specified in the VCI message, CM uses the default value. Parameters related to inter-frame delay are defined in Table 6-15 on page 78.

CM implements three inter-frame delay related modes:

- dontSet – which means do not change inter-frame delay (use previously set inter-frame delay). This mode is used when inter-frame delay is set from some other source (not via CM).
- setDelay – set value explicitelyspecified in the VCI message (attribute interFrameDelay). When this mode is used, VCI message should specify inter-frame delay in micro seconds and status of the 'randomizer' (on/off).
- minDelay – determine the minimum inter-frame delay that can be used based on the number of Baseline Boards that send frames to the same destination. In this mode inter-frame delay and status of the randomizer are determined by CM.

The default mode and, for setDelay, the default for the inter-frame delay and randomizer are configurable parameters of the CM and can be changed via CM GUI. **Initally, default mode is set to minDelay.**

As a part of the mapping process, CM determines inter-frame delay mode and value for each subband of each subarray and assigns those parameters to rows/columns on the Baseline Boards.

For minDelay mode, inter-frame delay is determined based on the number of Baseline Boards that send frames to the same destination. Each time when a subarray is created or deleted CM re-calculates inter-frame delay for minDelay for all destinations. For each destination (IP address) CM determines the number of Baseline Boards that transmit frames to that destination and calculates inter-frame delay as follows:

```
ifd = (numBoards==1) ? 0 : ((((numBoards-1)*9.1)+4)/4*4)
if (ifd > 1020) { ifd = 1020 }
```

When forwarding configuration to Baseline Boards, for each Baseline Board, CM determines inter-frame delay mode & value, as follows:

a) If at least one row/column on a Baseline Board uses mode dontSet, CM does not set inter-frame delay (Baseline Board will use previously set value or value set by some other source).

b) If at least one row/column on a Baseline Board uses mode setDelay, CM sets inter-frame delay as specified for that row/column. If more than one row/column uses setDelay, CM sets the longest inter-frame delay for mode setDelay. CM sets randomize inter-frame delay switch as assigned for the longest inter-frame delay.

c) Otherwise CM uses inter-frame delay as determined for minDelay. For a Baseline Board that sends frames to two or more different destinations, CM uses the longest inter-frame delay calculated for minDelay. If inter-frame delay is zero, CM disables randomization of the inter-frame delay, otherwise CM enables randomization.

6.4.13 Autocorrelator Mode

In autocorrelator mode all 16 cells (CCCs) in a Correlator Chip are concatenated and used to produce auto-correlation product for a single input data stream (product A*A or B*B).

Lag chain for the auto-correlation product is symmetric; in autocorrelator mode Correlator Chip produces a half of the lag chain (2048 complex lags) for product with 2048 spectral channels (4096 complex lags).

Cross-products R*L and L*R *can not* be obtained in autocorrelator mode.[1]

There are 8 Correlator Chips in the Baseline Board column and 8 input data streams per column; each Correlator Chip can produce auto-correlation product for a different input data stream, which means that 8 Correlator Chips in a column can produce auto-correlation products for all 8 input streams (4 polarized pairs).

The following restrictions apply in autocorrelator mode:

1. Recirculation is not suppored. Recirculation factor, if specified, must be 1.
2. Number of spectral channels is 2048.
3. Re-qunatization (number of bits to correlate) must be 4.

XML Example 6-20 is an example of VCI configuration for autocorrelator mode. In the same fashion as for the regular polarization products (baselines), user must specify product integration parameters and Baseline Boards to be used.

Number of Baseline Board columns needed depends on the number of stations in a subarray. If N is the number of stations in a subarray, number of Baseline Board columns needed for the autocorrelator mode can be calculated as follows:

```
if ((N MOD 4) = 0)  than    numColumns = (N/4)
else                        numColumns = (N/4)+1
```

A Baseline Board can produce auto-correlation products for one subband pair for up to 32 stations.

Autocorrelator mode is not fully implemented. *CM is able to configure the correlator, but cannot configure CBE (products generated in autocorrelator mode contain only half of the lag chain; additional parameters must be specified in the CBE XML schema to support this mode.*

[1] Recirculation FPGA could be modified to provide for R*L and L*R products in the autocorrelator mode. For now, there is no plan to implement that feature.

XML Example 6-20 Autocorrelator Mode


```
<basebandPair bbA="4" bbB="5" bw="2048MHz" inQuant="3bit">
        <subband id="0" bw="128MHz" centralFreq="64000000" reQuant="4bit" >
            <polProducts autoCorrMode="on" >
                <productIntegration  recirculation="1" minIntegTime="200" ccIntegFactor="5"
                                                      ltaIntegFactor="10"   cbeIntegFactor="20"/>
                <pp id="1" correlation="A*A" />
                <pp id="4" correlation="B*B" />
                <blbPair quadrant="3" firstBlbPair="0" numBlbPairs="1" />
            </polProducts>
        </subband>
</basebandPair>
```

6.4.14 Phase Binning

Correlator Chip has two banks of 1000 bins that can be used for phase binning. In the case when up to 1000 bins are defined per binning period, one bank is used for binning while data from the other bank is being transmitted. Alternatively, user may specify up to 2000 bins per period, in which case binning must pause during transmission.

Binning period and bins are specified per baseband, but binning can be turned on for individual subbands, so that some of the subbands can be used for binning while others are used to produce 'regular' correlation products. However, it is not possible to get both for the same subband.

The following parameters are specified per baseband and apply for all the subbands where phase binning is turned on:

a) Binning period
b) Maximum hardware integration time
c) Number of bins and bin width.

Binning status (on/off) and phase for the first bin can be specified per subband.

Table 6-25 ***Phase binning parameters***

Optional parameters of the XML element **baseBand.**	
binningPeriod	Binning period in microseconds.Must be specified when binning is ON for at least one subband.
binMaxHwIntegTime	Relevant only when phase bining is used. If not specified CMIB uses the default value of 500 micro seconds.
Parameters of the XML element phaseBinning. There can be up to 2000 instances of the element phaseBinning per baseBand.	
phase	Start of the first phase bin relative to epoch as fraction of the binning period. Range [0.0 to 1.0]. This is an optional parameter, if not specified it is assumed that phase is 0.0.
binWidth	Bin width as fraction of the binning period. Range [0.0 to 1.0]. Must be specified.
numBins	Number of adjacent phase bins that have the same width. If not specified, CMIB assumes one.

Optional parameters of the XML element **subBand.**	
phaseBinning	Relevant only when binningPeriod is specified in the baseBand. Range [on, off]. If not specified it is assumed that phase binning is off.
binningOffset	Phase (offset) specified as fraction of the binning period. Range [0.0 to 1.0]. If not specified assumed to ne 0.0. This parameter can be used to specify different phase for each subband. When specified binningOffest is added to phase specified in the element phaseBinning.

XML Example 6-21 Phase Binning

```
<widar:baseBand inQuant="8" bw="1024000000"
            swbbName="AC_8BIT" bbB="2" bbA="0"  binningPeriod="1000000.0">

    <widar:phaseBinning phase="0.1" binWidth="0.01" numBins="3"/>
    <widar:phaseBinning phase="0.5" binWidth="0.03" numBins="1"/>
    <widar:phaseBinning phase="0.6" binWidth="0.02" numBins="4"/>

    <widar:subBand sbid="1" swIndex="1" centralFreq="64000000"
                          bw="128000000"
                          rqNumBits="4"
                          phaseBinning="on">
      <widar:polProducts>
        <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
        <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
        <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
        <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
        <widar:blbProdIntegration recirculation="1"
                          ltaIntegFactor="4"
                          cbeIntegFactor="1" />
        <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
        <widar:stationPacking algorithm="maxPack"/>
        <widar:productPacking algorithm="maxPack"/>
        <widar:autoCorrSubset algorithm="halfStationsMaxProd"/>
      </widar:polProducts>
    </widar:subBand>
  </widar:baseBand>
```

6.4.15 Burst Mode

In burst mode, correlator is operating in the regime where sustained data transfer, at the requested rate, is not possible; loss of either observing time or bandwidth is inevitable:

a) Loss of observing time occurs when, while waiting for data to be transferred to the Correlator Backend, no data is being processed or produced by the correlator.

b) Loss of bandwidth occurs when one or more subband correlators are burst-capturing data, while other sub-correlators are transferring their data to the Backend. Time multiplexing may be performed either for subbands or for baselines.

The document "Programmer's Guide to EVLA Correlator System Timing, Synchronization, Data Products and Operation" [2] provides a rather detailed discussion of the capabilities and proposed uses for the burst mode.

Correlator hardware and firmware provide support for two modes of operation:

1. DUMPTRIG controlled mode, where the commands embedded in the DUMPTRIG signal control burst duration, start times and blanking times.

2. CMIB controlled mode, where the DUMPTRIG control sequence is continuous, and the software enables/disables baselines, using a predefined algorithm and parameters specified by the VCI Client.

At this time (Release v3.15) software (CMIB and CM) implements only DUMPTRIG controlled burst mode. Time multiplexing is performed for subbands.

Burst mode parameters are specified per subband in the XML element blbProdIntegration. See Table 6-26.

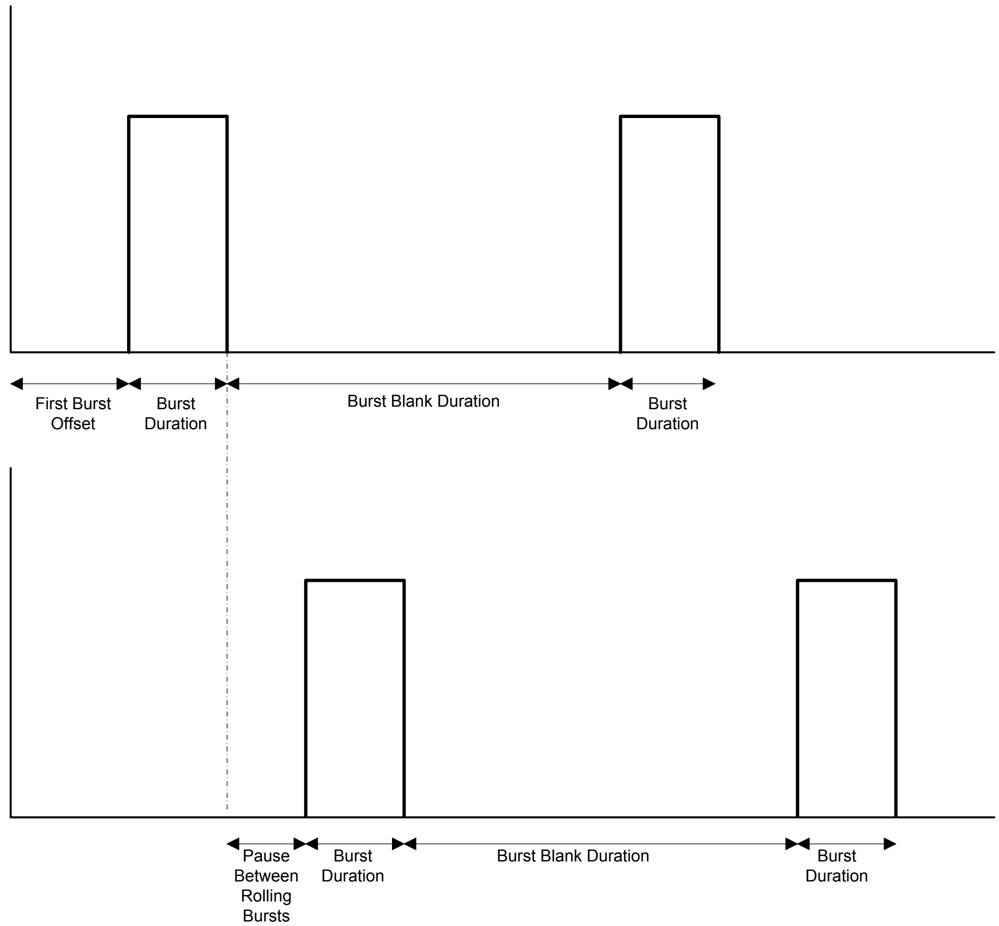
For the complete list of blbProdIntegration parameters see Table 6-19 on page 87.

Term "rolling bursts" is used to describe configuration where two or more sets of Baseline Boards are used to generate short bursts of output data for the same subband in time-multiplex mode. For'rolling bursts' the same subband is forwarded to up to 16 sets of Baseline Boards, the same products are configured on all sets, but each set of Baseline Boards gets different DUMPTRIG. Burst duration and blank duration are same for all, only the offset is different for each set of Baseline Boards. Figure 6-41 is an example for two 'rolling bursts'.

Table 6-26 ***subarray / stationInputOutput / baseband / subband / polProducts / blbProdIntegration***

Subband parameter	R/O	Range	Description
XML Tag	R	blbProdIntegration	XML element used to specify parameters that control product integration and dumping.
burstDuration	O	Integer Default=0	Burst duration as integer multiple of integration time. Zero means burst mode is not enabled.
burstBlankDuration	O	Integer Default=0	Duration for the period between the bursts (when Baseline Baord does not transmit data) as integer multiple of integration time.Relevant only if burstDuration > 0. Must be specified if burstDuration > 0.
firstBurstOffset	O	Integer Default=0	Offset for the first burst as integer multiple of integration time. Relevant only if burstDuration>0.
numRollingBursts	O	Integer Default=0	Number of rolling bursts. Relevant only if burstDuration>0. Determines how many sests of Baseline Boards are configured for this subband.
pauseBetweenBursts	O	Integer Default=0	Pause between rolling bursts as integer multiple of integration time. Relevant only if burstDuration>0 and rolling bursts are used.

Figure 6-41 Burst Mode Parameters for number of rolling bursts = 2.



6.4.16 Summed Array

Baseline Board input consists of 32 HM connectors. HM connector has four-pair "wafer" carrying: two sampled data streams, a control stream and a clock. In general, eachBaseline Baord input wafer carries data and control streams for different antenna. For polarized input, two data streams carry different polarization for the same antenna.

Baseline Board is able to produce phased sum of input data streams. The sum of input data is produced in the RXP FPGA; an RXP FPGA can produce phased sum for one input data stream (A or B) for all input wafers or for a subset of input wafers. For detailed information related to signal processing refer to the document "Baseline Board RXP FPGA, Requirements and Functional Specification" [12]. In short, RXP FPGA first applies the phase for each station individually and then calculates the sum.

There are two RXP FPGAs on the Baseline Board, which means that a single Baseline Board can produce the sum for both input data streams (A and B[1]).

In the WIDAR correlator, Baseline Boards are connected as pairs, so that two adjacent Baseline Boards receive the same input. If the sum of both A and B stream is required, CM uses the lower RXP on both Baseline Boards in a pair, so that the 1st Baseline Board produces the sum for data stream A and the 2nd Baseline Board produces the sum for the data stream B. Reasons for this choice will be discussed later, when discussing VDIF and auto-correlation products for the sum of the array.

Phased sum of a subarray can be forwarded to the following outlets:

- The sum of stations in VDIF Frame Format Version A.5.8 [15] can be embedded in UDP/IP packets and transmitted over 1Gig Ethernet to the destination specified by the user.

- Auto-correlation products for the sum can be sent to the CBE in the same fashion as 'regular' polarization products.

- The sum may be routed to the lower two rows of the Baseline Board X ERNI connector, for use by an external piece of hardware.

By default, the sum includes all the stations that belong to subarray.

Alternatively, user can specify the list of stations to be excluded from the sum.

---

[1] VCI uses "generic" identifies A and B, polarization can be defined as Rand L, or X and Y.

Re-quantization

RXP chip always uses 4-bit, 2's complement requnatizer for data destined for the correlator chip array and tX ERNI connector.

For data destined for VDIF packets, RXP chip implements selectable 1, 2, 4 or 8-bit offset binary re-qunatizer.

When using 4-bit re-quantizer for VDIF, summed array output for all destinations uses the same re-quantizer hardware, as a consequence, parameters related to gain and Automatic Gain Control (AGC) in elements *vdif* and *cc* must have identical values. CM will reject configuration if that is not the case.

Table 6-27 lists parameters for the XML element *summedArray*. All parameters, except SID, are optional. Default values (used if parameter is not specified in the VCI message) are specified in the table. SID is used to uniquely identify the sum. There can be up to two instances of the element *summedArray* per subband.

The sum is produced if the VCI "create subarray" contains element *summedArray*.[1] To get Baseline Boards to transmit VDIF and/or auto-correlation products, products must be explicitly enabled using XML elements shown in Table 6-27.

[1] Element *summedArray* is specified per subband.

Table 6-27 create: *subarray / stationInputOutput / baseband / subband / summedArray*

Subband parameter	R/O	Range	Description
XML tag		summedArray	XML element. Can be specified only in "create subarray". If element *summedArray* is specified, phased sum of input data streams (subband) is enabled. Functionality for modify subarray is described later in the text.
Station ID (SID)	R	33..255	One of 3 WIDAR identifiers (SID, BBID, SBID) used to identify a data stream. In this case SID is assigned to the sum. SID/BBID/SBID are inserted into lag frames generated for the auto-correlation products. BBID and SBID are specified in elements baseband and subband. See XML Example 6-22 on page 172.
Exclude stations	O	list of SIDs	List of stations to be excluded from the sum. The list should contain only stations that belong to the subarray. Stations that do not belong to the subarray are excluded by default.
Zero fill invalid data	O	true / false Default: true	True – Replace invalid data with zeros. Default. False – Do not modify invalid data.
Auto integration	O	true / false Default=true	Automatically set window parameters.
Apply window integrations in h/w	O	true / false Default=true	This attribute controls whether or not window integrations are applied in h/w. Default: ON. This attribute corresponds to the RXP FPGA register ATWINCR-1. For detailed explanation refer to RXP FPGA RFS [12]
6dB headroom	O	true / false Default=false	If ON (true) an additional bit is added to the window selection so that 6dB of headroom is built in, resulting in a lower probability of data overflow. For detailed explanation see register ATWINCR-0, field Push in RXP FPGA RFS [12].
Integration time	O	Positive integer Default=10 sec.	Relevant only when auto integration is disabled. Integration time in seconds for manually controlled integrations.
Continuous integration	O	true / false Default=true	Relevant only if auto integration disabled. If auto integration is disabled user can select either continuous integration (true) or one time integration (false).
blbPair	O	XML element. Baseline Boards to be used for the sum of the subarray. Baseline Boards may be specified for *vdif* and auto-correlation products (*cc*) individually. If not specified at all, CM will use the Baseline Boards assigned for the polarization products. See Table 6-20.	
vdif	O	XML element. Transmit the sum in VDIF format. See Table 6-28.	
cc	O	XML element. Auto-correlation products for the sum of stations. See Table 6-29.	

6.4.16.1 VDIF

Phased sum of a subarray can be sent to the designated destination in VLBI Data Interchange Format (VDIF) Version A.5.8. VDIF frames are embedded in UDP/IP packets and transmitted over 1Gig Ethernet to the destination specified by the user.

Baseline Board has two output ports, SFP1 and SFP2. Polarization products (produced by the Correlator Chips and Long Term Accumulators) can be transmitted only via SFP1. VDIF packets can be transmitted either via SFP1 or via SFP2. To simplify the configuration, it has been decided that VDIF packets will be always transmitted via port SFP2.

All VDIF packets produced on the same Baseline Board are sent to the same destination.

CM configures Baseline Boards so that the sums for different polarizations (right and left) are generated on different boards. For that purpose, CM uses the lower RXP on both boards. This allows users to select different destination addresses for different polarizations.[1]

For the summed array data transmitted in VDIF packets RXP uses selectable 1, 2, 4 or 8-bit offset binary re-qunatizer. If the number of bits for re-quantizer is not explicitly specified by the user, 2-bit re-quantizer is used for VDIF output.

If the number of bits for VDIF re-qunatizer is 4, requantizers for VDIF and CC use the same gain and power meter, consequentially, re-qunatizer gain and AGC setup must be the same for VDIF and CC output. This rule is enforced by CM.

VDIF frame size can be specified by the user. VDIF frame size must be selected so that there are an integer number of frames per second. If not specified by the user, CM sets the default value (1250).

---

[1] This also allows user to get auto-correlation products (A*A and B*B) for the sum using row/column 7 on the Baseline Board, which in turn simplifies Baseline Board configuration in the case that auto-correlation products for the summed data are requested. See discussion related to auto-correlation products for the summed array later in the document.

Table 6-28 create: *subarray / stationInputOutput / baseband / subband / summedArra / vdif*

VDIF parameter	R/O	Range	Description
XML tag		*vdif*	XML element.
vdifEnableA	O	true / false Default =false	Transmission of VDIF packets is controlled (enabled/disabled) for each polarization individually. True – TransmitVDIF packets False – Do not transmit VDIF packets If element *vdif* is not present, VDIF transmission is disabled for A and B.
vdifEnableB	O		
stationId	R	0..65535	Station ID for VDIF frames. 16-bit identifier. Not to be confused with WIDAR Station ID (SID).
aThread	O	Integer Range [0-1023]	10-bit integer. If not specified, CM will create Thread ID based on the BBID, SBID and polarization, as follows: 1-bit polarization (MSB – Most Significant Bit) 3-bits BBID 1-bit unused 5-bits SBID (LSB – Least Significant Bit) Plarization bit is assigned as follows: For R and X = 0, for L and Y = 1
bThread	O		
frameSize	O	250-2000 Default=1250	VDIF frame size in 32-bit words. Must be set so that there are an integer number of frames per second, otherwise frame build synchronization errors will occur.
numBits	O	1,2,4,8 Default=2	Number of bits per sample (for requantizer). If only one link to the GigE FPGA is used: 1 bit samples: 32 samples per word. Max sample rate: 256 Ms/s 2 bit samples: 16 samples per word. Max sample rate: 256 Ms/s 4 bit samples: 8 samples per word. Max sample rate: 128 Ms/s 8 bit samples: 4 samples per word. Max sample rate: 64 Ms/s If VDIF transmission is enabled on both RXPs, max sample rate is half of above shown values.
aPacketDelay	O	0..255	Delay from the start of the packet from RXP to start of the packet on the Ethernet, in milliseconds. If not specified in VCI message, delay is calculated by CM as: vdifCounter*6. (This functionality was implemented in May 2012 but was not documented.) VDIF counter starts from zero and is incremented for every board that generates VDIF frames.
bPacketDelay			
epoch	R	Integer , Range [0.. 63]	Reference epoch for VDIF frame header.
epochOffset	R	30-bit integer	Number of seconds to align EVLA time to VLBA time.

VDIF parameter	R/O	Range	Description
aDestMAC	O	MAC address Default: 12:34:56:78:90:00	Destination address for the Baseband A VDIF frames. If not specified, use the default address. Note: At this time only one MARK5 recorder is available.
aDestIP	O	IP address Default: 192.168.0.21	
aDestPort	O	Integer Default: 12002	
bDestMAC	O	MAC address Default: same as address for A	Destination address for Baseband B VDIF frames. If not specified use the same address as for Baseband A. Note: Destination for A and B can be different only if the sums for A and B are produced on different Baseline Boards. When possible, CM will configure the Baseline Boards so that the sum for different polarizations is produced on different Baseline Boards. When at least one BlB Pair is assigned to a subband and the subarray has exclusive use of the BlB Pair, the sums for A and B can be produced on different BlBs and destination for the VDIF packets can be different for A and B.
bDestIP	O	IP address Default: same as address for A	
bDestPort	O	Integer Default: same as port for A	
requnatGain	O	0 - 250	Requantizer gain. Relevant only if AGC is disabled.
agcEnabled	O	true / false Default=true	Enable/disable Automatic Gain Control (AGC) for the summed array data. By default AGC is enabled.
agcRms	O	Decimal number.	Automatic Gain Control : Target RMS Relevant only if AGC is enabled. Default: For 1-bit samples: not relevant. For 2-bit samples: 1.8 For 4-bit samples: 2.7 For 8-bit samples: not relevant.
blbPair	O	XML element. Baseline Boards to be used to generate VDIF products. If not specified here, use BlBs specified in the element *summedArray*. If not specified at all, CM will use the Baseline Boards assigned for the polarization products. See Table 6-20.	

6.4.16.2 X ERNI Connector

Phased sum of a subarray can be routed to the lower two rows of the Baseline Board X ERNI connector, for use by an external piece of hardware.

To enable this feature add element cc to the VCI configuration for *summedArray*.

6.4.16.3 Auto-correlation Products for the Sum

Auto-correlation products for the phased sum of a subarray can be generated and sent to the CBE in the same fashion as 'regular' polarization products (for baselines and antennas).

Table 6-29 lists configurable parameters (XML attributes.

The list of products, integration time and Baseline Boards specified for the 'regular' polarization products are also used for the auto-correlation products for the sum. Correlator Backend stores auto-correlation products for the sum in the same BDF file with other products for the same subband (and the same subarray); consequently, the number of products, number of spectral channels per product and integration time must be the same for all correlation products.

To get the auto-correlation products, the sum generated in RXP FPGA must be routed to the correlator chip array.

The sum produced by the lower RXP is routed to the row/column 7, input 3, stream B. The sum produced by the upper RXP is routed to the row/column 3, input 3, stream B.

The sum for both streams (A and B) and the auto-correlation products (A*A, A*B, B*B) for the sum, can be configured on a single Baseline Board. The following Correlator Chips would be used: {3,3}, {3,7}, {7, 3} and {7,7} for that purpose. Two rows/columns on a Baseline Board would be used exclusively for the sum of the subarray, leaving only 6 rows/columns to be used for baselines.

In general, it is possible to get polarization products for the baselines and for the sum on the same Correlator Chip. However, in practice, this would require CM to implement a rather sophisticated algorithm in order to ensure that configuration is valid. Although this eventually may be implemented, presently, rows and columns on the Baseline Boards are either exclusively used for the sum or exclusively used for the baselines.

<u>Auto-correlation products for the sum are supported only for the stationPacking=fourPerRowColumn (maxPack).</u>

Reason: For station packing modes 'one station per row/column' and 'two stations per row/column', in some cases, the 2nd Baseline Board in a pair does not get input for all the stations in a subarray. In general, Baseline Board configuration is not as predictable as for the 'maxPack' mode. It would be possible to implement support for the auto-correlations for the sum for all the packing modes, if that becomes priority.

Auto-correlation products for the sum of the subarray are configured in row/column 7, when auto-correlations for the sum are enabled row/column 7 is used exclusively for the sum. The lower RXP on the 1st Baseline Board in a pair is used to generate the sum for stream A; the lower RXP on the 2nd Baseline Board in a pair is used to generate the sum for stream B. Products are configured on the 2nd Baseline Board Correlator Chip [7,7]. Auto-correlation products for the sum are configured in the same manner as other products (the same number of the lag chain segments and cells per Correlator Chip are used for all products). Note that this means that only one CCQ is used in the Correlator Chip [7,7].

For configurations that use Baseline Board Stacking, CM could use more cells per Correlator Chip for the sum if that is required (and desired) functionality.

Table 6-29 create: *subarray / stationInputOutput / baseband / subband / summedArray / cc*

Subband parameter	R/O	Range	Description
XML tag	R	cc	XML element. If present, the phased sum of stations is forwarded to the correlator chip array and X ERNI connector. If summedArray does not contain this element, the sum is not forwarded to the correlator chip array and X ERNI connector.
Requnatizer gain	O	0 - 250	Requantizer gain. Relevant only if AGC is disabled.
agcEnable	O	true / false Default=True	Enable/disable Automatic Gain Control (AGC) for the summed array data. By default AGC is enabled.
agcRms	O	Decimal number.	Automatic Gain Control : Target RMS Relevant only if AGC is enabled. Default:2.7

XML Example 6-22 Summed ArrayConfiguration (VDIF and auto-correlation products)

```
<subArray configId="test" action="create" activationId="test-c" >
  <stationInputOutput sid="list" >
    <station sid="1" />
    <station sid="2" />
    <station sid="3" />
    <bbParams bbid="0" sourceType="FORM" sourceId="0" polarization="R" />
    <bbParams bbid="2" sourceType="FORM" sourceId="0" polarization="L" />

    <baseBand bbA="0" bbB="2" swbbName="AC_8BIT" bw="1024000000" inQuant="8" >
      <subBand sbid="0" swIndex="1" bw="128000000" centralFreq="64000000" rqNumBits="4" >
          <polProducts>
                <pp id="1" correlation="A*A" spectralChannels="64"/>
                <pp id="2" correlation="A*B" spectralChannels="64"/>
                <pp id="3" correlation="B*A" spectralChannels="64"/>
                <pp id="4" correlation="B*B*" spectralChannels="64"/>
                <blbProdIntegration recirculation="1" minIntegTime="250.0" ccIntegFactor="1"
                                    ltaIntegFactor="4" cbeIntegFactor="1"/>
                <blbPair quadrant="1" firstBlbPair="0" numBlbPairs="1"/>
          </polProducts>
          <summedArray sid="120"  >
                <vdif  vdifEnableA="true" vdifEnableB="true" stationId="65500"
                      epoch="61" epochOffset="10"    />
                <cc  />
          </summedArray>
      </subBand>
    </baseBand>
  </stationInputOutput>
</subArray>
```

6.4.16.4 Modify Summed Array

XML element *summedArray* may be used to modify previously configured sum.

Only a subset of attributes can be modified, as follows:

- List of stations (antennas) included in the sum. By default, it is assumed that all stations that belong to the subarray should be included in the sum. Attribute exclude stations can be used to specify stations to be excluded from the list.
- Transmission of the VDIF packets can be disabled/enabled for each polarization independently.
- Automatic Gain Control can be enabled/disabled and target RMS can be changed.

Tables below list XML elements and attributes that can be used to modify summedArray configuration.

Table 6-30 modify: ***subarray / stationInputOutput / baseband / subband / summedArray***

Subband parameter	R/O	Range	Description
XML tag		summedArray	XML element. Can be specified only in “create subarray”. If element *summedArray* is specified, phased sum of input data streams (subband) is enabled. Functionality for modify subarray is described later in the text.
Station ID (SID)	R	33..255	summedArray to be modified.
Modify Exclude Stations	O	Boolean Default=False	This attribute is relevant in ‘modify subarray’ only. If true, list of excluded stations received before (in create or modify subarray) is replaced with the new list (specified in the same message). Attribute excludeStations is defined as a list; empty list means ‘all stations should be included in the sum’.
Exclude stations	O	list of SIDs	List of stations to be excluded from the sum. The list should contain only stations that belong to the subarray. This list completely replaces previously received list of stations.
vdif	O	XML element. See Table 6-31.	
cc	O	XML element. See Table 6-32.	

Table 6-31 modify: ***subarray / stationInputOutput / baseband / subband / summedArra / vdif***

VDIF parameter	R/O	Range	Description
XML tag		*vdif*	XML element.
vdifEnableA	O	true / false Default: noChange	Can be used to enable/disable transmission of (previously configured) VDIF packets. If element *vdif* is not present, or attribute is not present VDIF status does not change.
vdifEnableB	O		
stationId	R	0..65535	Used to identify VDIF affected by the change (16 bits).
requnatGain	O	0 – 250 Default: noChange	Requantizer gain. Relevant only if AGC is disabled. If not specified no change.
agcEnabled	O	true / false Default: noChange	Enable/disable Automatic Gain Control (AGC). If not specified AGC status does not change.
agcRms	O	Decimal number. Default: noChange	Target RMS for the Automatic Gain Control. Relevant only if AGC is enabled. If not specified, no change.

Table 6-32 modify: ***subarray / stationInputOutput / baseband / subband / summedArray / cc***

Subband parameter	R/O	Range	Description
XML tag	R	Cc	XML element.
Requnatizer gain	O	0 – 250 Default: noChange	Requantizer gain. Relevant only if AGC is disabled.
agcEnable	O	true / false Default: noChange	Enable/disable Automatic Gain Control (AGC).
agcRms	O	Decimal number. Default: noChange	Target RMS for the Automatic Gain Control. Relevant only if AGC is enabled. If not specified, no change.

6.4.17 Tone Extraction

Not implemented.

This section is outdated and should be updated when requirements are defined.

Station Board hardware has the ability to extract one tone (frequency) per subband. Extraction of more than one tone per subband can be implemented in software, using time-multiplexing, as follows:

- On each 10msec interrupt, Station Board software changes tone to be extracted.
- Station Board software goes through the list of the tones to be extracted in round-robin fashion, starting from the lowest frequency.

The tone (frequency) to be extracted could be specified in the Phase Model for Tone Extraction, in real-time. Content and format of the phase model for tone extraction is described in Section 9.2. Static parameters for tone extraction could be specified per subarray (and apply for all stations in a subarray). If not same for all the stations in a subarray, tone extraction could be specified for each station individually using element station in the list of stations. The same XML element, described in Table 6-33 could be used in both cases.The maximum number of the tones that can be extracted for a single subband is specified in Section 9.2 which defines the content of the Phase Model for Tone Extraction. The limit on the number of tones is not imposed by hardware; it is an arbitrary number, and may be evaluated and changed if needed.

Table 6-33 ***subarray / stationInputOutput / baseband / subband / toneExtraction***

Subband Parameter	R/O	Range	Description
XML tag	R	toneExtraction	This element specifies parameters that do not change during a scan or experiment.
Number of tones to be extracted	R	0..32 Default = 0	Number of tones to be extracted for the subband. This parameter will be needed so that Station Board software can allocate memory in advance and generate alert if tone extraction phase models are not received for the specified number of tones.
Dwell time	O	Positive integer Default = 1	Dwell time specified as multiple of 10 msec interrupts. Relevant if more than one tone per subband is to be extracted.
Integration factor	R	Positive integer Default = 10	Specifies how many times value should be added before the data can be off-loaded.

6.4.18 Gating

Not implemented.

This section is outdated and should be updated when requirements are defined.

The specified "gate" is applied to the baseband. When gating is enabled, signal outside the "gate" is marked as invalid. Typically, the same set of gating parameters applies for all the stations in a subarray.

When gating is *on* for a pair of basebands, *pulsar gating phase,* which specifies the shift of the pulsar gate in fractions of period, should be specified for each subband.[1]

Model for pulsar gating may be either specified in advance or in real-time.

Table 6-34 ***subarray / stationInputOutput / baseband / pulsarGating***

Baseband Parameter	R/O	Range	Description
XML Tag	R	pulsarGating	XML element.
Status	R	On / Off	Turns pulsar gating on or off
Gate Model	O	XML element. Model for pulsar gate could be specified as a part of configuration (in advance) or in real-time.	
Period	R	TBD	Pulsar period.
Gate Width	R	0..100	Gate width specified in fractions of cycle.
Epoch	R	UT (ISO8601)	The time when the pulsar period begins.
ModelCoeff	O	XML element. Coefficient for the pulsar gate model. Number of coefficients per model is not limited.	
Coefficient	R	Floating point number	Coefficent for the pulsar gating model.

[1] Alternatively, user could specify frequency and dispersion for a baseband, and leave to the correlator to calculate the phase for each subband.

6.4.19 Radar Mode

Not implemented.

This section is outdated and should be updated when requirements are defined.

Output of a single filter on a Station Board Filter Bank can be captured and transmitted to the predefined destination. This ability to save raw data is known as “radar mode”. Destination address for the captured data can be defined as configurable parameter.

Unlike other output products, where duration can not be specified in advance, radar mode parameter allows VCI Client to specify desired duration for the raw data saving.

Radar mode can be turned on for the subbands where the bandwidth does not exceed 500 kHz.

Radar mode can be turned on for a subband which is not correlated[1].

Table 6-35 ***subarray / stationInputOutput / baseband / radarMode***

Subband Parameter	R/O	Range	Description
XML tag	R	radarMode	XML element
Status	R	On / Off	Turn radar mode on / off
destination	O	IP address	Destination address for the saved row data. If not specified, data is sent to the pre-defined address (system attribute).
duration	O	Non negative integer	Duration in 10milli second interrupts. If status is ON, should be positive integer.

[1] The same applies for all Station Board output products.

6.4.20 Enable/Disable Baseline

Not implemented.

This functionality could be added to ‘create subarray’ and ‘modify subarray’ to allow user to disable correlation (and generation of products) for a particular baseline or station.

By default, all baselines are enabled.

Disabling a baseline woulds not release allocated Correlator Chip Cells (CCCs); Baseline Board configuration would remain the same, only correlation and “dumping” of the output products would be disabled; so that baselines can be disabled and enabled ‘on the fly’.

Table 6-36 ***subband / baseline***

Baseline parameters	R/O	Range	Description
XML tag		baseline	XML element.
status	M	Enable Disable	Disable / enable output products for the specified baseline(s).
xStation	M	1-255	If two stations are specified, disable / enable baseline X*Y. If one station is specified, disable / enable all the baselines for station X.
yStation	O	1-255	

6.5 Activation Trigger

The Activation Trigger is used to specify the activation time for a group of configuration messages that have the same Activation ID. Table 6-37 lists the parameters of the Activation Trigger.

6.5.1 Mapping Time

At Mapping Time, Configuration Mapper searches the Configuration Queue for all the previously received configuration messages with the specified Activation ID, removes them from the queue, and generates the detailed configuration for the Station Boards and Baseline Boards. The configuration is forwarded to the Station Baord and Baseline Baord CMIBs as soon as possible.

Mapping Time *may* be specified in the Activation Trigger. When the "Mapping Time" is *not* explicitly specified, mapping is performed immediately upon receipt of the Activation Trigger.

Mapping Time has been introduced to accommodate requirements of the EMCS where configuration for each antenna may be generated (and transmitted) by a different process and/or CPU and the acknowledgments are ignored. VCI Client may explicitly specify Mapping Time to postpone the *mapping* of the configuration, i.e. the translation of the VCI configuration parameters into the actual correlator configuration.

6.5.2 Activation Time

Activation Time specifies when the correlator should start processing observed data and generating output products. Activation Time applies to all pending Configuration Request messages[1] with the same Activation ID. If Activation Time is not specified, configuration is activated as soon as possible.

Activation Time applies for all the Configuration Request messages with the same Activation ID received before the Mapping Time. In other words, configuration messages received after the Activation Trigger and before the Mapping Time will be processed at the Mapping Time.

In order to ensure that the configuration can be activated at specified time, configuration messages and Activation Trigger should be transmitted at least 6 seconds in advance.

If mapping time is specified, activation time must be after the mapping time.

Table 6-37 Activation Trigger

[1] In this context "pending configuration request" means a previously received Configuration Request message for which activation time has not been specified.

Name	R/O	Range	Description
Message Type	R	activationTrigger	XML element Activation Trigger.
Activation ID	R	ASCII String	Activation Identifier, used as a reference. All pending Configuration Messages with the same Activation ID will be processed ("mapped") as a group at the mapping time, and activated at the activation time.
Activation Time	O	UT	UT displayed in format defined by ISO 8601. Time when the correlator should start processing observed data and generating output products. Activation Time applies to all the pending Configuration Request messages with the same Activation ID. If activation time is not specified, configuration is activated as soon as possible.
Mapping Time	O	UT	UT displayed in format defined by ISO 8601. Time when the configuration messages with the specified Activation ID should be removed from the Configuration Queue and "mapping" of the VCI configuration into detailed hardware configuration should be performed. If Mapping Time is not specified, mapping is performed immediately upon receipt of the Activation Trigger.
Query	O	Yes, No Default: No	If present, this parameter indicates that the configuration should not be activated. The correlator should perform "mapping", generate accept/reject message and discard the configuration.

6.6 CM Monitor & Control

VCI protocol defines a set of parameters and commands that can be used to monitor and control the software that implements the VCI interface.

XML element *cmMonitorControl* is used to configure and monitor parameters of the WIDAR software application which implements VCI interface: VCI Configuration Mapper (CM).

Unlike the configuration for stations and subarrays, which is used to instruct the correlator how to processes observed data and which output products to generate; the XML element *cmMonitorControl* is used to control CM functions, such as:

- Enable/disable logging, set logging level,
- Enable/disable XML schema validation,
- Enable/disable vciReport generation,
- List content of CM queues and flush (clear) CM queues.
- Block transmission of messages for selected destination (Station Boards, Baseline Boards, Crossbar Boards, CBE, CRM, VCI Reponses and Reports).
- Flush (clear) CMIB queues (per rack).
- Delete all subarrays (active and scheduled).
- Delete active or scheduled subarray for specified Config ID.
- Load list of administratively disabled boards and components from CRM Data Base.
- Set all boards and components (FPGAs/ASICs) as administratively enabled. This can be used to reverse effect of CRM Data Base query.

XML element *cmMonitorControl* does not specify Activation ID; it is not possible to assign activation time for this message; configuration change is performed immediately upon receipt of a *cmMonitorControl* message.

Rationale: Given the nature of the functions it controls, there is no need to schedule the changes in advance. Quite contrary, that could be source of confusion.

Tables below list parameters that can be specified / monitored using this message.

6.6.1 Delete Subarray

In release 3.10 of the VCI protocol, a new command that allows user to delete all instances of the specified subarray, active and/or pending has been implemented. Unlike other commands in this group, this command affects the correlator configuration (and not only CM), but was added to allow user(s) to delete subarray without checking the actual subarray status (active or pending). Instead of Config ID, user can specify 'all' to delete all the subarrays.

Table 6-38 *cmMonitorControl*

Name	R/O	Range	Description
Message Type	R	*cmMonitorControl*	XML element for CM Monitor & Control
query	O	Yes, No Default = "Yes"	XML attribute. If 'yes" cmMonitorControl is treated as "queryOnly"; CM setup does not change, CM generates a response message to report the current CM setup. See Appendix B for an example of the quary and response messages.
crmQuery	O	Yes, No Default="No"	XML attribute. Use to instruct CM to transmit CRM query, i.e. to obtain list of boards and components that are administratively disabled in CRM Data Base.
VCI Schema Validation	O	Yes, No	XML attribute. Disable/enable VCI schema validation. This attribute is optional, but there is no default value. If the attribute is not specified in the request, CM setup for VCI schema validation will not be affected.
transmitMessagesForSTBs	O	Yes, No	XML attribute. Can be used to administratively disable messages for all Station Boards[1]. If not present, the status of the CM parameters is not affected.
transmitMessagesForXBBs	O	Yes, No	XML attribute. Can be used to administratively disable messages for all Cross-Bar Boards[1]. If not present, the status of the CM parameters is not affected.
transmitMessagesForBLBs	O	Yes, No	XML attribute. Can be used to administratively disable messages for all Baseline Boards.If not present, the status of the CM parameters is not affected.
transmitMessagesForCBE	O	Yes, No	XML attribute. Can be used to disable messages for CBE. If not present, CM status remains the same.

[1] If the board is administratively disabled, configuration and other messages for the board are not transmitted. The same applies for CBE and CRM.

Name	R/O	Range	Description
transmitMessagesForCRM	O	Yes, No	XML attribute. Can be used to disable messages transmitted to CRM (CRM query).
CM Alerts	O	XML element cmAlerts. Attribute transmit='yes/no' can be used to disable/enable generation of CM alerts.	
CM Configuration Queue	O	XML element cfgQueue. CM configuration queue commands: - List - Flush queue. CM configuration queue holds VCI messages before the mapping is performed.	
CM Activation Queue	O	XML element actQueue. CM activation queue commands: - List - Flush queue. CM activation queue holds aconfiguration after the mapping has been successfully performed.	
CM Control Queue	O	XML element ctrlQueue. CM control queue commands: - List - Flush queue. CM places received cmMonitorControl messages into Control Queue.	
CM Queue for Backend destined messages	O	XML element cbeQueue. CM control queue commands: - List - Flush queue. Messages destined for CBE are placed into this queue. Messages for CBE are transmitted one at a time. CM does not transmit next message until previously transmitted message has been confirmed.	
Set CM logging parameters	O	XML element cmLogging. See Table 6-39	
Generating vciReports	O	XML element vicReporting. See Table 6-41.	
Configuration Status	O	XML element queryCfgStatus. Request for configuration status report for a particular configuration. See Table 6-41	
Delete Subarray	O	XML element cmDeleteSubarray. Delete subarray with the specified Config ID. 'All' means delete all active and pending subarrays.	
Flush CMIB Queues	O	XML element cmFlushCmibQueues. Contains list of racks where CMIB queues should be flushed. Attributes are s001 to s008, and b101 to 108. Example s001='yes' means flush CMIB queues in Station Rack 001.	

Table 6-39 ***cmLogging***

Name	R/O	Range	Description
Message Type	R	*cmLogging*	XML element used to specify and report CM logging parameters.
Logging level	O	TRACE DEBUG INFO NOTICE WARNING ERROR	Used to specify the desired logging level for CM. If not specified logging level remains as is.
Log to file	O	Yes, No	Disable/enable logging to log file.
Log VCI messages	O	Yes, No	Enable/disable logging of received and transmitted VCI messages in the logging file[1].
Log Station Board messages	O	Yes, No	Enable/disable logging of received and transmitted Station Board messages in the logging file[1].
Log Cross-Bar Board messages	O	Yes, No	Enable/disable logging of received and transmitted Cross-Board messages in the logging file[1].
Log Baseline Board messages	O	Yes, No	Enable/disable logging of received and transmitted Baseline Board messages in the logging file[1].
Log CBE messages	O	Yes, No	Enable/disable logging of received and transmitted CBE messages in the logging file[1].
Log CRM messages	O	Yes, No	Enable/disable logging of received and transmitted CRM messages in the logging file[1].

[1] For this command to take effect, logging to file must be enabled.

Table 6-40 ***vciReporting***

Name	R/O	Range	Description
Message Type	R	*vciReporting*	XML element used to set and report CM parameters related to generation and transmission of VCI Status Reports.
Transmit	R	Yes, No	Disable/enable generation of VCI Configuration Status Reports. For description and definition of VCI Configuration Status Reports see Section 7.5.
Destination IP address	O	IP address	IP address - destination for the VCI Reports generated by CM. BY default VCI Reports are transmitted as multicast messages. If not specified destination address remains as is.
Destination Port	O	Integer, valid port number	Destination port for the vcireports generated by the CM. If not specified, port remains as is.

Table 6-41 Request Configuration Status Report: ***queryCfgStatus***

Name	R/O	Range	Description
Message Type	R	*queryCfgStatus*	XML element used to request configuration status report. Note: All 3 attributes are optional, but one (and only one) should be specified. User can specify: cfg=active or cfg=next or activation time or CorrModel ID. If more than one identifies is specified, CM uses the first valid identifier, in following order: 1. cfg 2. actTime 3. CorrModel ID
cfg	O	Active, Next	Identify the configuration for which user wants report: i. active or ii. next (the 1st configuration in the activation queue).
actTime	O	UT	Activation time for which user wants to get report.

Name	R/O	Range	Description
CorrModel ID	O	Integer	Used to identify the configuration for which user wants to get report. CorrModel ID is internal CM parameter. This option is useful for debugging.

XML Example 6-23 CM Response to Status Query


```xml
<ns2:vciResponse version="3.14" timeStamp="2012-07-10T00:17:59.109Z" msgId="1"
    xsi:schemaLocation="http://www.nrc.ca/namespaces/widar
   http://www.aoc.nrao.edu/asg/widar/schemata/vci/vci3-11/vci/vciResponse.xsd"
   xmlns:ns2="http://www.nrc.ca/namespaces/widar"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

   <ns2:vciAck version="3.14" timeStamp="2012-07-10T00:17:59.110Z"
               desc="Response to query." msgId="3">
      <ns2:refMessage>
         <ns2:cmMonitorControl query="yes">
            <ns2:queryCfgStatus cfg="active"/>
         </ns2:cmMonitorControl>
      </ns2:refMessage>
   </ns2:vciAck>
   <ns2:vciReport timeStamp="2012-07-10T00:17:59.110Z" msgId="2"
                  forActTime="2012-07-10T00:17:08.000Z">
      <ns2:report>
      Configuration Change Status for corrModel ID=0
      activation time            = 2012-07-10 00:17:08.000
      active since               = 2012-07-10 00:17:07.310
      last msg. transmitted at   = 2012-07-10 00:17:58.437
      No configuration change.
      Notice: Station Boards   are administratively disabled.
      Notice: Cross-bar Boards are administratively disabled.
      Notice: Baseline Boards  are administratively disabled.
      Notice: Communication with CBE is administratively disabled.
      </ns2:report>
   </ns2:vciReport>
   <ns2:cmMonitorControl vciSchemaValidation="no" sendQueryToCRM="no"
                              sendConfigToCBE="no"     sendConfigToXBBs="no"
                           sendConfigToBLBs="no"     sendConfigToSTBs="no">
      <ns2:cfgQueue>Configuration Queue empty.</ns2:cfgQueue>
      <ns2:actQueue>Activation Queue empty.</ns2:actQueue>
      <ns2:ctrlQueue>CM Control queue is empty.</ns2:ctrlQueue>
      <ns2:cmLogging logXAlerts="yes"  logCrmMessages="no" logCbeMessages="no"
                     logXbbMessages="no" logBlbMessages="no"
                     logStbMessages="no"  logVciMessages="yes"
```

```
                          logToFile="yes" level="INFO"/>
        <ns2:vciReporting destPort="55230" destIpAddress="239.192.2.5"
                           transmit="yes"/>
        <ns2:cmAlerts transmit="no"/>
    </ns2:cmMonitorControl>
</ns2:vciResponse>
```

7 VCI Response

The correlator (CM) may originate the following configuration related messages:

a) Acknowledgement (Ack)

b) Negative Acknowledgement (Nack)

c) Accept

d) Reject

e) Report

The above listed messages are transmitted within a VCI Response, which is an "envelope" message and may contain more than one response message.

Acknowledgement and Negative Acknowledgement are transmitted immediately upon receipt of the request. Ack and Nack messages are transmitted as HTTP response and sent to the originator of the vciRequest.

Configuration Accept, Reject and Report are transmitted as multicast messages to the

Multicast group: 239.192.2.5

Port: 55230

Rationale:

- Messages Accept and Reject are used to report success or failure of the mapping for a group of messages with the same Activation ID. It is not necessary that all the messages that should be activated at the same time come from the same source. Actually, it was envisioned that, in the EVLA Monitor & Control, one application will send Station configuration and the other Subarray configuration. Both applications, and perhaps even the 3rd one, may be interested to know whether or not the configuration was accepted.
- More than one application may be monitoring status of correlator configuration.
- If all reports are sent to well-known[1] multicast group, there is no need for CM to store the address of the originator for eachVCI message.

[1] See section 13.7 on page 201.

7.1 Acknowledgement

An acknowledgment is generated by the Configuration Mapper (CM) as a direct response to a received configuration request. The acknowledgment is used to notify the originator of the message being acknowledged that:

1. The request has been received.
2. The basic syntactic and semantic checks are successfully passed. Using XML terminology: message is well formed, and if schema validation is ON, message is valid.
3. The message has been stored in the queue and is waiting Activation Trigger.

Table 7-1 vciAck / vciNack

Parameter	R/O	Range	Description
XML tag	R	vciAck vciNack	Acknowledgement Negative Acknowledgement
Time Stamp	O	UT	UT displayed in format defined by ISO 8601. Time when the message was generated.
Message ID	R	1..2^31-2	Message ID.
Version	O	3.15	Version of the VCI Protocol. Corresponds to the version (revision) of this document.
Descriptor	O	String	Optional text description or explanation.
XML message being acknowledged	O	See section 13.2	In most cases vciAck and The received message may be added to the Acknowledgement.
Log	O	Table 10-1	Log generated while processing the acknowledged message.

7.2 Negative Acknoweldgement (Nack)

Nack message is generated by the Configuration Mapper (CM) as a direct response to a received VCI request. The Nack is a negative acknowledgement; it informs originator that the message was discarded and the reason why the message was discarded. The format and content of the negative acknowledgement is the same as positive acknowledgment (Table 7-1), only XML tag is different (vciNack).

7.3 Accept

A VCI Accept message is generated by the Configuration Mapper (CM) as a response to a received Activation Trigger. The VCI Accept conveys the following:

1. The Activation Trigger has been received,
2. All the messages with the same Activation ID have been successfully mapped into correlator configuration and will be activated at the specified Activation Time.

The VCI Accept message is transmitted to a well-known multicast address (see Well-known Numbers on page 210).

If the configuration cannot be not be activated exactly as specified (i.e. if the original request was modified by the Configuration Mapper), VCI Accept contains relevant information; more precisely it contains a Log Record with detailed information.

The VCI Accept message is not a guarantee that the configuration will be activated; due to configuration changes that mat take place between the acceptance and activation of the configuration the accepted configuration may fail at activation time. If that is the case, Configuration Mapper will generate alert.

Table 7-2 vciAccept / vciReject

Parameter	R/O	Range	Description
Message Type	R	vciAccept vcireject	Configuration Accepted Configuration rejected
Time Stamp	O	UT	UT displayed in format defined by ISO 8601. Time when the message was generated.
Version	O	3.15	Version of the VCI Protocol. Corresponds to the version (revision) of this document.
Activation Time	R	UT	UT displayed in format defined by ISO 8601. Time when the configuration will be activated (from the received Activation Trigger).
Ref.Message	R	1..2^31-2	Message ID of the received Activation Trigger.
Log	O	Table 10-1	Log message(s) generated by the Configuration Mapper.

XML Example 7-1 vciAccept

```xml
<ns2:vciResponse version="3.9" timeStamp="2010-09-16T01:41:50.725Z" desc="widar"
msgId="102" xsi:schemaLocation="http://www.nrc.ca/namespaces/widar
http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciResponse.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ns2:vciAccept version="3.9" timeStamp="2010-09-16T01:41:50.725Z"
desc="mappingCompleted" msgId="100">
      <ns2:refMessage>
         <ns2:subArray msgId="295" activationId="Delete-A-1-0" action="delete"
                                          configId="A-1-0"/>
      </ns2:refMessage>
      <ns2:vciLog msgId="5" timeStamp="2010-09-16T01:41:50.725Z" code=" " level="INFO">
         <ns2:originator thread="9" method="sendCfgToCbe()"
                          class="ca.nrc.widar.vciMapper.VciConfigMapperGlobals"
                          componentID=""
                          componentType="ca.nrc.widar.vciMapper.VciConfigMapper"/>
         <ns2:description>Configure CBE administratively disabled.</ns2:description>
      </ns2:vciLog>
  </ns2:vciAccept>

  <ns2:vciAccept version="3.9" timeStamp="2010-09-16T01:41:50.725Z"
                 desc="mappingCompleted" msgId="101">
      <ns2:refMessage>
         <ns2:activationTrigger activationTime="2010-08-17T18:27:00"
                                activationId="Delete-A-1-0" msgId="296"/>
      </ns2:refMessage>
  </ns2:vciAccept>

</ns2:vciResponse>
```

7.4 Reject

A VCI Reject message is generated by the Configuration Mapper (CM) as a response to a received Activation Trigger.

The VCI Reject conveys the following:

- Activation Trigger has been received,
- The configuration will not be activated,
- All the messages with the same Activation ID have been discarded, and
- The reason why the configuration has been rejected.

The format of the VCI Reject message is the same as the format of the VCI Accept; only a VCI Reject message must contain at least one Log to inform EVLA M&C why the configuration was rejected.

The VCI Accept message is transmitted to a well-known multicast address (see Well-known Numbers on page 210).

7.5 Status Report

VCI Status Report is transmitted as a response to a query (cmMonitorControl/queryCfgStatus) after.

In the same manner as other VCI messages, configuration status is transmitted as XML message, but the report itself is an ASCII string, i.e. can be displayed to the user or operator as received.

Rationale: EVLA Monitor & Control System at this time does not provide an application that displays the correlator status in user friendly format. The only application that can be used to monitor the correlator/CM status is, so called, VCI Client, a GUI interface which displays received messages as text.

XML Example 7-2 is an example of CM configuration status report.

XML Example 7-2 Configuration Status Report

```
<ns2:vciResponse version="3.15" timeStamp="2013-07-02T19:41:08.268Z" msgId="1032"
xsi:schemaLocation="http://www.nrc.ca/namespaces/widar  http://www.aoc.nrao.edu/asg/widar/schemata/vci/vci3-13/vci/vciResponse.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ns2:vciAck version="3.15" timeStamp="2013-07-02T19:41:08.270Z" desc="Response to query." msgId="1034">
        <ns2:refMessage>
            <ns2:cmMonitorControl query="yes">
                <ns2:queryCfgStatus cfg="active"/>
            </ns2:cmMonitorControl>
        </ns2:refMessage>
    </ns2:vciAck>
    <ns2:vciReport timeStamp="2013-07-02T19:41:08.270Z" msgId="1033" forActTime="2013-07-02T19:26:15.000Z">
        <ns2:report>
Configuration Change Status for corrModel ID=160
activation time           = 2013-07-02 19:26:15.000
active since              = 2013-07-02 19:26:15.149
last msg. transmitted at = 2013-07-02 19:41:07.691
Forward configuration to CMIBs/CBE started: 2013-07-02 19:26:10.898, Not complete.
CBE config for configId=A.1, status=completed
s001-t-0 SID= 1, BBID=0/1, configName=A.1, Status admin=enabled, oper=running, used=no , config=active, info=OK, tCount=1
s001-t-1 SID= 2, BBID=0/1, configName=A.1, Status admin=enabled, oper=running, used=no , config=active, info=OK, tCount=1
s001-t-2 SID= 3, BBID=0/1, configName=A.1, Status admin=enabled, oper=running, used=no , config=active, info=OK, tCount=1
s001-t-3 SID= 4, BBID=0/1, configName=A.1, Status admin=enabled, oper=running, used=no , config=active, info=OK, tCount=1
StationRack s001 CrossBar Switch 0, adminStatus=enabled
StationRack s001 CrossBar Switch 1, adminStatus=enabled
BLB[1][00][1]=b101-t-0, configName=A.1, Status admin=enabled, oper=configured, used=yes , config=active, info=OK, tCount=1
BLB[1][00][0]=b101-t-1, configName=A.1, Status admin=enabled, oper=configured, used=yes , config=active, info=OK, tCount=1
</ns2:report>
    </ns2:vciReport>
    <ns2:cmMonitorControl vciSchemaValidation="no" sendQueryToCRM="yes" sendConfigToCBE="yes" sendConfigToXBBs="yes"
```

```
                              sendConfigToBLBs="yes" sendConfigToSTBs="yes">
        <ns2:cfgQueue>Configuration Queue empty.</ns2:cfgQueue>
        <ns2:actQueue>Activation Queue empty.</ns2:actQueue>
        <ns2:ctrlQueue>CM Control queue is empty.</ns2:ctrlQueue>
        <ns2:cbeOutputQueue>CM  CBE output queue is empty.</ns2:cbeOutputQueue>
        <ns2:cmLogging logXAlerts="yes" logCrmMessages="yes" logCbeMessages="no" logXbbMessages="no" logBlbMessages="no"
                      logStbMessages="no" logVciMessages="yes" logToFile="yes" level="INFO"/>
        <ns2:vciReporting destPort="55230" destIpAddress="239.192.2.5" transmit="yes"/>
        <ns2:cmAlerts transmit="yes"/>
        <ns2:ifdDefault randomOn="yes" delay="896" mode="minDelay"/>
    </ns2:cmMonitorControl>
</ns2:vciResponse>
```

8 CM Status

Additional XML schema file defines content and format of messages used to display CM status in the form of web-page (HTML format):

trunk/widar/schema/cmStatus/cmStatus.xsd

The following address can be used to display CM status in the general purpose browser:

http://mccc.evla.nrao.edu:8081/vciMapper

HTML links on the web-page can be used to display more detailed information. Those links use queries defined in XML schema file cmStatus.xsd. Queries defined in cmStatus.xsd can be entered in the browser navigation line, for example, to display active configuration enter:

http://mccc.evla.nrao.edu:8081/vciMapper?<corrModel desc='active' />

The following queries can be entered via a browser:

Display active configuration (CM object CorrModel):

```
<corrModel desc='active' />
```

Display more detailed list of the subarrays in the active configuration:

```
<corrModel desc='active' query="subarrays_html" />
```

Display the list of subarrays with the list of subbands and boards used for each station and subband:

```
<corrModel desc='active' query="subarrays_boards_html" />
```

Display the list of Station Boards, Baseline Baords or Cross-bar Boards modified by active configuration:

```
<corrModel desc='active' query="STB" />

<corrModel desc='active' query="BLB" />

<corrModel desc='active' query="XBB" />
```

Descriptor 'active' can be replaced by: next, previous, or first in the activation queue. Also, instead of the descriptor one can enter ID of the corrModel. For the full list of parameters see the XML schema cmStatus.xsd

For example, to display configuration for corrModel ID=155 enter the following:

<corrModel id='155' query="XBB" />

Most users will use the web-page to get CM status information. Direct use of cmStatus.xsd is for the expert user who understands how CM software is implemented.

All above mentioned queries get response in HTML format. In addition, XML element cmStatus can be used to get the list of active subarrays in XML format. This is used to provide information to MCAF.

Here is an example of such quiry and CM response:

Query:

<corrModel desc='active' query="subarrays_xml" />

CM Response:


```
<cmStatus>
   <corrModel id="155">
      <subarray configId="collimation.56168.997910659724.18">
         <station antenna="ea01" sid="1"/>
         <station antenna="ea02" sid="2"/>
         <station antenna="ea04" sid="4"/>
         <station antenna="ea06" sid="6"/>
         <station antenna="ea07" sid="7"/>
         <station antenna="ea10" sid="10"/>
         <station antenna="ea11" sid="11"/>
         <station antenna="ea12" sid="12"/>
         <station antenna="ea13" sid="13"/>
         <station antenna="ea14" sid="14"/>
         <station antenna="ea16" sid="16"/>
         <station antenna="ea17" sid="17"/>
         <station antenna="ea18" sid="18"/>
      </subarray>
   </corrModel>
</cmStatus>
```

9 Models

Delay models and phase models for tone extraction are real-time control messages supplied during the scan or experiment. Models are generated and transmitted by the EMCS.

The real-time control messages (models) flow in only one direction: from the EMCS to the correlator. The correlator does not acknowledge receipt of the models. If models are not received as expected the correlator will generate an alert.

There are three types of real-time control messages:

1. Delay Model for delay tracking,
2. Phase Model for tone extraction,
3. Gate Model for (pulsar) gating.

For an overview of the (pulsar) gating functionality refer to section 6.4.18. Detailed requirements and functionality for (pulsar) gating are still to be defined; other two models are described below.

9.1 Delay Model

A Delay Model message can contain multiple models. It is recommended that a single XML message should contain all the models for one Station Board, so that the MCCC can forward the models to the Station Boards with no reformatting.

For EVLA antennas with 8-bit initial quantixation, each Station Board has to receive models for both basebands that are part of a pair.

Each Station Board must generate phase models for all delay models it receives.

The maximum number of models that can be specified per one Station Board is 38:

- Up to 2 Baseband Delay Models, and
- Up to 36 Subband Delay Models (18 per baseband).

Sub-band delay tracking is optional.

The delay model is specified as a set of coefficients of a polynomial:

$$\tau(t) = C_0 + C_1(t-t_0) + C_2(t-t_0)^2 + C_3(t-t_0)^3 + \ldots.$$

The number of coefficients can vary.

Refer to the document “Programmers’ Guide to EVLA Correlator System Timing, Synchronization, Data Products and Operation” [2] for sign conventions and detailed discussion regarding delay tracking.

Table 9-1 ***vciStbDelayModel***

Parameter	R/O	Range	Description
Message Type	R	vciStbDelayModel	Delay Model message
Station ID	R	1..255	Identifier of the station.
Activation Epoch	R	UT	The time when the model should be activated. Format: ISO0601.
Epoch_t_o	O	MJD	Epoch of the polynomial. If not specified, it is assumed that the epoch of the model is the same as the activation epoch.
Baseband Delay Model	R	bbDelayModel See Table 9-2	XML element. Baseband Delay Model must be specified for each baseband of each station in a subarray. It is recommended that a single message should contain all the models that are destined to the same Station Board, i.e. all the models for a Baseband Pair (for EVLA 3-bit samplers). For EVLA antennas with 8-bit samplers, each baseband is processed on a different Station Board, but models may still be grouped per pair.
sslo	R	Non-negative number	Sum of antenna local oscillators. Specified in MHZ.
fShift	R	Non-negative number	WIDAR frequency shift. Specified in kHz.

Table 9-2 ***vciStbDelayModel / bbDelayModel***

Parameter	R/O	Range	Description
Model Type	R	bbDelayModel	Delay model for a single baseband
BB ID	R	0..7	Identifier of the baseband.
Coefficients	R	64bit floating point number	The coefficients of the polynomial. At least one coefficient must be specified. The number of coefficients in the model may vary. A coefficient is 64 bit (8 byte) floating-point number displayed as IEEE double-precision floating-point number.
sbDelayModel	O	See Table 9-3	XML element. Subband delay model. A baseband can have up to 18 subbands, which means that up to 18 subband delay models may be specified for a single baseband.

Table 9-3 ***vciStbDelayModel / bbDelayModel / sbDelayModel***

Parameter	R/O	Range	Description
Model Type	R	sbDelayModel	Subband delay model. Subband delay model s specified as a differential model with respect to baseband model.
SB ID	R	0..17	Identifier of the subband for which the delay model applies.
Coefficients	R	64bit floating point number	The coefficients of the polynomial. The number of coefficients may vary. A coefficient is 64 bit (8 byte) floating-point number displayed as IEEE double-precision floating-point number.

9.2 Phase Model for Tone Extraction

In the same fashion as the Delay Model message, the Phase Model for Tone Extraction (TEX Phase Model) can contain multiple models. To minimize the processing time of the MCCC, the TEX Phase Model messages should contain models for one Station Board, so that they can be forwarded (by the MCCC) to the Station Boards without reformatting.

Station Board hardware can extract one frequency at a time. Extraction of more than one tone per subband may be performed in time-sharing manner.

Software imposed limit on the number of tones that can be extracted per subband is 32.

Note: **tone extraction has not been implemented**.

Table 9-4 Phase Model for Tone Extraction

Parameter	R/O	Range	Description
Message Type	R	vciTexModel	Tone Phase Model for Tone Extraction (TEX).
Activation Epoch	R	UT	The time when the model becomes valid. Format: ISO 8601.
Epoch_t_o	O	MJD	Epoch of the polynomial. If not specified, it is assumed that the epoch of the polynomial is the same as the epoch of the model.
The following parameters are specified for each Subband where tone extraction is desired:			
Station ID	R	1 .. 255	Station identifier.
BB ID	R	0..7	Baseband identifier.
SB ID	R	0..17	Subband identifier.
The following parameters are specified for each tone to be extracted:			
Tone ID	R	String	ID of the tone to be extracted. User may specify the frequency or any other text. Note: ID is assigned "on-the-fly". The number of tones to be extracted is specified as a part of the subarray configuration (for each subband independently); so that the correlator knows in advance how many tones will be extracted and can prepare data structures and raise an alert if the number of the models that are received does not match the number specified in advance.
Coefficient	R	64 bit floating point number	The coefficient of the polynomial. This parameter can be specified more than once. The number of coefficients may vary, at least three coefficients should be specified: C_0 – phase C_1 – frequency (tone) to be extracted C_2 – phase rate Coefficients are specified as IEEE double-precision floating-point numbers.

10 Logs

All correlator components generate logs that are stored in the log files to be used for debugging and troubleshooting.

In the debugging mode, VCI Configuration Mapper allows user to select format for the generated log (XML or text) and the destination.

The CM version running on the MCCC generates logs in text format.

CM generated vciLog message contains at minimum:

1. time and date stamp,
2. Event ID,
3. priority.

Depending on the cause and nature of the event, additional parameters may be added to the vciLog message, for example:

- Type and ID of the component, which is the originator of the message.
- Message ID of the received VCI message that caused the generation of the Log,
- Type and ID of the component that caused or detected the event, etc.

Table 10-1 lists the parameters of the vciLog message.

Table 10-1 Log Message

Parameter	R/O	Range	Description
type	R	vciLog	XML tag
Time Stamp	R	UT	Date and time when the event was detected. The time stamp is generated by the component that has first detected and reported the event. UT in ISO 8601 format.
Level	R	See section 10.1	Logging level.
Event Description	O	ASCII String	A short textual description.
Component	O	String	Identifier of the component, device and/or object that caused the event or that is affected by the event. Present only when it can be identified. In general, a message should include the identifier for the lowest possible level. E.g. instead of reporting that a board failed, log should indicate the hardware or software component that caused the failure.
Host	R	List to be defined	Identifier of the host which originated the log (host is a component that has IP address assigned to it)
Originator	R	ASCII String	H/w device or s/w entity. The list of h/w and s/w originators will be finite but rather lengthy; it will be defined for each correlator component.
Logger	O	ASCII String	For Java components only: Name of the Java Logger.
Log	O	Table 10-1	May include another log generated by the same or some other component.
Property	O	ASCII String	Used to specify the name and value of a property that caused log generation.
Throwable	O	ASCII String	Java Throwable (exception) as a string. Included when software failure is detected. Can be fairly long.

XML Example 10-1 VCI Log generated by the CM - XML format


```
<ns2:vciLog msgId="4" timeStamp="2010-09-16T00:55:53.999Z" code=" " level="INFO">
        <ns2:originator thread="9" method="parsePolProducts()"
                          class="ca.nrc.widar.vciMapper.CorrModel" componentID=""
                           componentType="ca.nrc.widar.vciMapper.VciConfigMapper"/>
         <ns2:description>
                Mapping SUCCESSFUL for:
                PolProductGroup[[ Subarray=A-1-0 BBID=0/2 SBID=10,
                                        recFactor=1,
                                        minHwIntegTime=250.0,
                                        ccIntegFactor=1,
                                         ltaIntegFactor=  4000,
                                        cbeIntegFactor=1,
                                         stationPacking =MAX_PACK,
                                         productPacking=MAX_PACK,
                                         autoCorrs=ALL_STATIONS_MAX_PROD]
                  PolProduct[ id=1, correlation=A_A, numChannels=64, bitsToCorrelate=4]
                  PolProduct[ id=2, correlation=A_B, numChannels=64, bitsToCorrelate=4]
                  PolProduct[ id=3, correlation=B_A, numChannels=64, bitsToCorrelate=4]
                  PolProduct[ id=4, correlation=B_B, numChannels=64, bitsToCorrelate=4]
                  Stations: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
                  BLBs:  + assigned and used; | assigned, not used.
                        0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
                   Qadrant1: ++  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
                   Qadrant2: --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
                   Qadrant3: --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
                   Qadrant4: --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
]
        </ns2:description>
 </ns2:vciLog>
```


Example for CM log in text format :

```
2011-09-27T20:11:10.862  ca.nrc.widar.vciMapper.VciConfigMapperGlobals.setBlockCfgForCbe()
ThreadID=1
NOTICE: CM setup change: Block Configuration Messages for CBE = true
```

10.1 Priority

Table 10-2 lists the priority level supported by the VCI protocol. Priority or importance of the event is indicated as a string, where TRACE indicates the lowest severity, while ERROR indicates the highest severity. TRACE and DEBUG are used to log events that should be logged during testing and debugging only.

Table 10-2 Log Priority

Type	Description
Trace	Used to log events that may be significant during development and testing.
Debug	Debugging messages.
Information	Used to report an event that is significant enough to be reported, but is not perceived as failure, error or problem. E.g. may be used to inform the EMCS that the status of an object has changed from inactive to active.
Notice	Similar as information, but requires more attention, or relates to a large component or scarce resource.
Warning	Used to report out-of-tolerance state or a state that can cause error or failure if an action is not taken.
Error	Used to report an illegal or undefined state.

11 Alerts

VCI Configuration Mapper (CM) generates alerts in the EVLA format. Alert format and content is described elsewhere and is not part of this protocol specification.

VCI Configuration Mapper (CM) generates the following alerts:

1. config-failure-vci
2. config-failure-cbe
3. config-failure-cmib
4. software-error

Alert message for ‘config-failure’ includes Config ID.

Alert message for ‘software-error’ includes a short text that identifies the problem.

12 Station Board Data Products and Monitor Data

Station Board data products, listed in Table 12-1 are automatically enabled for each subarray (configuration). Station Board data products can be obtained from the Station Boards. The mechanisam for accessing StB Data Products is described elsewhere.

Table 12-1 Station Board Data Products

Data Type	Description
Input State Counts XML tag: stbInpStateCountsTable	Wideband input state counts. Collected per baseband. Hardware can collect state counts for up to 16 states at the same time. For initial quantization of 1,2,3 and 4 bits, Station Board can collect counts for all the states all the time. For input data stream that has more than 4 bits, the input state counts are collected in round robin fashion; software reads counts for 16 states per interrupt and stores them until state counts for all the states are collected. For 8-bit input, 16 interrupts are needed to collect the state counts for all the states, i.e. for each individual state, state counts are collected only 1/16 of time. For integration factor > 1, software collects counts for all the states once, and then starts from the beginning. By default, collection of input state counts is ON, integration factor is 100. Total integration time depends on the number of states, for 3-bit input it is 1 second, for 8-bit input it is 16 seconds.
Clip Counts XML tag: stbClipCntTable	Collected per subband. Station Board Filter has 4 stages. The number of filter stages used for a particular subband depends on the baseband and subband badwidth. Hardware can count clipped samples for one stage at a time. Software collects count of clipped samples for all used stages in round robin fashion. For more details refer to the Station Board Filter Chip RFS [7]. By default, collection of clip counts is ON, integration factor is 100. Total integration time depends on the number of stages used for each subband, and can be 1,2,3, or 4 seconds.
Power Counts XML tag: stbPowerTable	Collected per subband. Station Board Filter collects the measurements of: a) power in the signal before re-quantization for the off and on states of the noise diode; and b) power measurements made after re-quantization (for post-quantizer measurements state of the noise diode is ignored). The number of corresponding valid samples is also provided. For detailed description refer to Station Board Filter Chip RFS [7]. The exact list of data included with the power measurements is defined elsewhere. By default, collection of power counts is ON. Integration factor is 100 (total integration time 1 second).

Data Type	Description
RFI Counts XML tag: stbRfiTable	Counts the number of RFI detections. Collected per subband. RFI Detection Level and RFI Blanking Duration are configurable VCI parameters; see Table 6-15 on page 78. Implemented in the Station Board Filter chip. Each data sample is compared to the detection level. The counter is incremented if sample is outside ± RFI Detection Level range. If RFI Blanking Duration is greater than zero, data is invalidated for at least the blanking duration. If RFI is detected during the invalidated interval, the interval is extended by the blanking duration. For more details refer to [7].
Output State Counts XML tag: stbOutStateCountsTable	State counts for the output data stream of a Station Board Filter. Collected per subband. Hardware collects state counts for one state at a time, an algorithm that collects counts for all the states in round robin fashion is implemented in software. By default, collection of output state counts is ON, integration factor is 100 . Total integration time depends on the number of bits per sample in the filter chip output, for 4-bit subband it is 16 seconds, for 7-bit subband total integration time is 128 seconds.

At this time content and format of the monitor data is defined only for the Station Board. Table 12-2 lists the monitor data generated by the Station Board.

Station Board monitor data is always ON and can be obtained directly from the Station Board.

Table 12-2 Station Board Monitor Data

Data Type	Description
Time Interval Counts stbTicTable	Time Interval Counts can be used for error detection. The time interval counters measure the time interval between the data tick and the system tick. Those numbers should be constant.
Error Counts stbErrorCntTable	Station Board software error counters.
CRC stbCrcTable	CRC error counters for individual links.

13 Name Spaces

The following sections address name space management for the following identifiers used by the VCI protocol:

1. Message ID
2. Message Type
3. Activation ID
4. Scan ID
5. Antenna ID

13.1 XML Namespace

XML Schema describes content and format of the VCI messages. XML Schema defines VCI message content as elements and attributes, and assigns name to each element and attribute.

XML Schemas that define content of VCI messages are part of the protocol specification.

An XML namespace is a collection of element and attribute names identified by the Unified Resource Identifier (URI). The URI (Universal Resource Locator) does not need to point to a real location (e.g. web-page, filesystem, directory or file), quite often this is used to indicate which organization or authority has defined the XML schema. XML schema for VCI messages defines name space "widar", as follows:

xmlns:widar="http://www.nrc.ca/namespaces/widar"

In addition, the schema defines the URI for the target namespace, which tells the recipient (or reader) of the XML document where to find the name space definition (i.e. the XML schema). XML schema for the VCI protocol is available in the NRAO software repository and on the NRAO server:

http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciRequest.html

13.2 Message Type

The message type is specified as an XML tag.

XML elements that have attribute Activation ID and can be individually activated are in this document refered to as VCI messages, for example vciRequest and vciResponse are VCI messages.

XML elements that have no attribute "Activation ID" can not be individually activated and are always transmitted as children of the elements that have "Activation ID" assigned to them.

1. VCI messages that can be generated by the VCI Client:
    - 1.1. vciRequest (envelope)
        - 1.1.1. activationTrigger
        - 1.1.2. stationHw
        - 1.1.3. subarray
        - 1.1.4. cmMonitorControl
2. VCI messages that can be generated by the correlator (CM):
    - 2.1. vciResponse
        - 2.1.1. vciAck
        - 2.1.2. vciNack
        - 2.1.3. vciAccept
        - 2.1.4. vciReject
        - 2.1.5. vciReport
        - 2.1.6. vciLog

13.3 Message ID

Message identifier is an integer. Message IDs are assigned independently for each direction. The WIDAR correlator and EMCS are free to assign the Message IDs according to their needs.

The Message ID name space is not defined by this protocol specification.

13.4 Activation ID

The Activation ID is an ASCII string. The Activation ID name space is not defined in this document. The originator of the configuration messages (VCI Client) is free to assign Activation Identifiers according to own needs.

13.5 Scan ID

The Scan Identifier is an ASCII string. The Scan ID name space is not defined in this document. The originator of the configuration messages (VCI Client) is free to assign Scan Identifiers according to own needs.

13.6 Antenna ID

Antenna Identifiers are assigned by the VCI Client. The correlator does not use Antenna IDs, if received Antenna ID is stored with other configuration parameters for reference only.

13.7 Well-known Numbers

This section lists the well-known numbers used by the VCI peers.

VCI Accept, Reject and Report are transmitted as multicast messages to

Multicast group: 239.192.2.5

Port: 55230

Multicast group and port are configurable parameters og the VCI Configuration Mapper; group and port can be modified using vciRequest/cmMonitorControl (and via CM GUI).

14 References

[1] The VLA Expansion Project, project book, NRAO http://www.aoc.nrao.edu/evla/pbook.shtml

[2] Programmer's Guide to EVLA Correlator System Timing, Synchronization, Data Products and Operation, Users Guide, DRAO Document A25290N0000, Brent Carlson, NRC.

[3] EVLA Correlator Monitor and Control System, Test Software, and Backend Software Requirements and Design Concepts, NRC-EVLA Memo #15, Brent Carlson, NRC.

[4] Assigned Numbers, IETF, RFC 1700, October 1994.

[5] EVLA Correlator Chip RFS, DRAO Document A25082N0000, Brent Carlson, NRC.

[6] Baseline Board LTA Controller FPGA RFS, DRAO Document A25091N000, Brent Carlson, NRC.

[7] Station Board Filter Chip RFS, DRAO Document A25044N0000, David Fort, NRC.

[8] EVLA Correlator System Numbering Plan, DRAO Document A25010N0002, Brent Carlson, NRC.

[9] XML Schema Part 2: Datatypes Second Edition, W3C recommendation, October 2004. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

[10] ISO 8601:2000 Second Edition, ISO (International Organization for Standardization), Representations of dates and times, Second edition, 2000-12-15.

[11] EVLA Station Board VSI Test FPGA RFS, DRAO Document A25050N0000, David Fort, NRC.

[12] Baseline Board RXP (Re-timing, X-bar and Phasing) FPGA RFS, DRAO Document A25093N0000, Brent Carlson, NRC.

[13] Protocol Specification, HM Gbps Cable Signaling Specification, Protocol Document: A25022N0041, Revision 1.2, August 18, 2005, Carlson, B.

[14] VLBI Data Interchange Format (VDIF) Specification, Ratified 26 June 2009, VDIF Task Force, Version 1.0.

[15] Interface Control Document, HM Gbps Cable Physical Specification, ICD: A25022N0040, Revision DRAFT, April 13, 2004, Zhang, H.

[16] Baseline Board Recirculation Controller RFS, Brent Carlson, 25090N0000

[17] EVLA Correlator User Manual, Brent Carlson, DRAO Document 25010N0010

[18] VCI Configuration Mapper (CM) User Manual, Sonja Vrcic, DRAO Document A25201N0001.

[19] Baseline Board GBit Ethernet Chip V2, Requirements and Functional Specification, Document: A25092N0001, Carlson B.

15 VCI Request - Summary of XML Elements

```
<vciRequest>

    <stationHw>
        <baseBandHw/>
        <antenna/>
    </stationHw>

    <subArray>
        <listOfStations>
            <station >
                <bb />
            </station>
        </listOfStations>
        <stationInputOutput>
            <station     />
            <bbParams />
            <baseBand>
                <phaseBinning/>
                <subBand>
                    <polProducts>
                        <pp/>
                        <blbProdIntegration/>
                        <blbPair/>
                        <stationPacking/>
                        <productPacking/>
                        <autoCorrSubset/>
                    </polProducts>
                    <summedArray>
                        <vdif>
                            <blbPair/>
                        </vdif>
                        <cc />
                        <blbPair/>
                    </summedArray>
                </subBand>
                <wpp/>
            </baseBand>
        </stationInputOutput>
        <baseline/>
    </subArray>
    <activationTrigger/>
</vciRequest>
```

16 XML Examples

XML Example 16-2 to XML Example 16-5 show VCI configuration request (*vciRequest/stationHw*) transmitted by the VCI Client, and the corresponding responses and report generated by CM. These messages are real-world examples copied for the VCI Client log.

XML Example 16-6 is an example of the CM status query and the corresponding report. These messages are real-world examples copied from the VCI Client log.

XML Example 16-1 VCI Request for the configuration shown in Figure 6-7 and Figure 6-8


```
<vciRequest xmlns="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="nraoRepository\widar\schema\vci\vciRequest.xsd" msgId="234" timeStamp="2009-12-16T09:30:47" version="3.9" >

    <!- Example for the station hardware configuration. It is assumed that all the stations are configured in the same way. -->

    <stationHw sid="1"  activationId="Example1" mappingOrder="1" action="add" timeStamp="2009-12-16T09:30:47" msgId="321">
        <baseBandHw bbid="0" stationBoardMlid="s001-t-0" dataPath="0"/>
        <baseBandHw bbid="1" stationBoardMlid="s001-t-0" dataPath="1"/>
        <baseBandHw bbid="2" stationBoardMlid="s001-t-4" dataPath="0"/>
        <baseBandHw bbid="3" stationBoardMlid="s001-t-4" dataPath="1"/>
        <baseBandHw bbid="4" stationBoardMlid="s001-b-0" dataPath="0"/>
        <baseBandHw bbid="5" stationBoardMlid="s001-b-0" dataPath="1"/>
        <baseBandHw bbid="6" stationBoardMlid="s001-b-4" dataPath="0"/>
        <baseBandHw bbid="7" stationBoardMlid="s001-b-4" dataPath="1"/>
        <noiseDiode status="enable" periodMilliSec="10.0" dutyCycle="50.0" phase="0.0"/>
        <antenna type="EVLA" id="Antenna1" />
    </stationHw>

    <subArray configId="Example1"  action="create" scanId="3bitIQ-4bitRQ"  msgId="432" activationId="Example1"
                                                   timeStamp="2009-12-16T09:30:47"  mappingOrder="100" >
        <listOfStations action="add">
            <station sid="1" / >
            <station sid="2" / >
            <station sid="3" / >
            <station sid="4" / >
        </listOfStations>
```

```xml
<!--  Subband bandwidth, central frequency, number of products, and product integration
      are not specified in the grid; the values shown here are arbitrarily choosen by the author.  -->

<stationInputOutput sid="all">

  <bbParams bbid="0" sourceType="FORM" sourceId="0" polarization="R" sideband="upper" />
  <bbParams bbid="1" sourceType="FORM" sourceId="1" polarization="L" sideband="upper" />
  <bbParams bbid="2" sourceType="FORM" sourceId="0" polarization="R" sideband="upper" />
  <bbParams bbid="3" sourceType="FORM" sourceId="1" polarization="L" sideband="upper" />
  <bbParams bbid="4" sourceType="FORM" sourceId="0" polarization="R" sideband="upper" />
  <bbParams bbid="5" sourceType="FORM" sourceId="1" polarization="L" sideband="upper" />
  <bbParams bbid="6" sourceType="FORM" sourceId="0" polarization="R" sideband="upper" />
  <bbParams bbid="7" sourceType="FORM" sourceId="1" polarization="L" sideband="upper" />

  <baseBand bbA="0" bbB="1" name="BBP1" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >

    <subBand sbid="0"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
      <polProducts>
        <productIntegration 1"0" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="512"/>
        <pp id="2" correlation="A*B" spectralChannels="512"/>
        <pp id="3" correlation="B*A" spectralChannels="512"/>
        <pp id="4" correlation="B*B" spectralChannels="512"/>
        <blbPair quadrant="1" firstBlbPair="0" numBlbPairs="4" />
        <blbPair quadrant="2" firstBlbPair="0" numBlbPairs="4" />
      </polProducts>
    </subBand>
```

```
<subBand sbid="6"  bw="128000000" centralFreq="192000000" rqNumBits="4" >
    <polProducts>
        <productIntegration recirculation="8" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="512"/>
        <pp id="2" correlation="A*B" spectralChannels="512"/>
        <pp id="3" correlation="B*A" spectralChannels="512"/>
        <pp id="4" correlation="B*B" spectralChannels="512"/>
        <blbPair quadrant="1" firstBlbPair="6" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="7"  bw="128000000" centralFreq="320000000" rqNumBits="4" >
    <polProducts>
        <productIntegration 1"0" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="64"/>
        <pp id="2" correlation="A*B" spectralChannels="64"/>
        <pp id="3" correlation="B*A" spectralChannels="64"/>
        <pp id="4" correlation="B*B" spectralChannels="64"/>
        <blbPair quadrant="1" firstBlbPair="7" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="10"  bw="128000000" centralFreq="448000000" rqNumBits="4" >
    <polProducts>
        <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="1024"/>
        <pp id="2" correlation="A*B" spectralChannels="1024"/>
        <pp id="3" correlation="B*A" spectralChannels="1024"/>
        <pp id="4" correlation="B*B" spectralChannels="1024"/>
        <blbPair quadrant="1" firstBlbPair="8" numBlbPairs="4" />
        <blbPair quadrant="2" firstBlbPair="8" numBlbPairs="4" />
        <blbPair quadrant="3" firstBlbPair="8" numBlbPairs="4" />
        <blbPair quadrant="4" firstBlbPair="8" numBlbPairs="4" />
```

```xml
        </polProducts>
    </subBand>
    <subBand sbid="13"  bw="128000000" centralFreq="576000000" rqNumBits="4" >
        <polProducts>
            <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
            <pp id="1" correlation="A*A" spectralChannels="1024"/>
            <pp id="2" correlation="B*B" spectralChannels="1024"/>
            <blbPair quadrant="1" firstBlbPair="12" numBlbPairs="4" />
            <blbPair quadrant="3" firstBlbPair="12" numBlbPairs="4" />
        </polProducts>
    </subBand>
</baseBand>

<baseBand bbA="2" bbB="3" name="BBP2" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
    <subBand sbid="4"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
        <polProducts>
            <productIntegration 1"0" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
            <pp id="1" correlation="A*A" spectralChannels="512"/>
            <pp id="2" correlation="A*B" spectralChannels="512"/>
            <pp id="3" correlation="B*A" spectralChannels="512"/>
            <pp id="4" correlation="B*B" spectralChannels="512"/>
            <blbPair quadrant="1" firstBlbPair="4" numBlbPairs="2" />
            <blbPair quadrant="2" firstBlbPair="4" numBlbPairs="2" />
            <blbPair quadrant="3" firstBlbPair="4" numBlbPairs="2" />
            <blbPair quadrant="4" firstBlbPair="4" numBlbPairs="2" />
        </polProducts>
    </subBand>
```

```
<subBand sbid="6"  bw="128000000" centralFreq="192000000" rqNumBits="4" >
    <polProducts>
        <productIntegration recirculation="8" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="512"/>
        <pp id="2" correlation="A*B" spectralChannels="512"/>
        <pp id="3" correlation="B*A" spectralChannels="512"/>
        <pp id="4" correlation="B*B" spectralChannels="512"/>
        <blbPair quadrant="2" firstBlbPair="6" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="7"  bw="128000000" centralFreq="320000000" rqNumBits="4" >
    <polProducts>
        <productIntegration 1"0" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="64"/>
        <pp id="2" correlation="A*B" spectralChannels="64"/>
        <pp id="3" correlation="B*A" spectralChannels="64"/>
        <pp id="4" correlation="B*B" spectralChannels="64"/>
        <blbPair quadrant="2" firstBlbPair="7" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="13"  bw="128000000" centralFreq="448000000" rqNumBits="4" >
    <polProducts>
        <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
        <pp id="1" correlation="A*A" spectralChannels="128"/>
        <pp id="2" correlation="A*B" spectralChannels="128"/>
        <pp id="3" correlation="B*A" spectralChannels="128"/>
        <pp id="4" correlation="B*B" spectralChannels="128"/>
        <blbPair quadrant="2" firstBlbPair="13" numBlbPairs="1" />
        <blbPair quadrant="4" firstBlbPair="13" numBlbPairs="1" />
    </polProducts>
</subBand>
```

```
        <subBand sbid="14"  bw="128000000" centralFreq="576000000" rqNumBits="4" >
            <polProducts>
                <productIntegration recirculation="2" minIntegTime="250" ccIntegFactor="2"  ltaIntegFactor="10"  cbeIntegFactor="10" />
                <pp id="1" correlation="A*A" spectralChannels="1024"/>
                <pp id="2" correlation="B*B" spectralChannels="1024"/>
                <blbPair quadrant="2" firstBlbPair="12" numBlbPairs="1" />
                <blbPair quadrant="4" firstBlbPair="12" numBlbPairs="1" />
                <blbPair quadrant="2" firstBlbPair="14" numBlbPairs="1" />
                <blbPair quadrant="4" firstBlbPair="14" numBlbPairs="1" />
            </polProducts>
        </subBand>
    </baseBand>

    <baseBand bbA="4" bbB="5" name="BBP3" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
        <subBand sbid="0"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
            <polProducts>
                 <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10" />
                <pp id="1" correlation="A*A" spectralChannels="256"/>
                <pp id="2" correlation="A*B" spectralChannels="256"/>
                <pp id="3" correlation="B*A" spectralChannels="256"/>
                <pp id="4" correlation="B*B" spectralChannels="256"/>
                <blbPair quadrant="3" firstBlbPair="0" numBlbPairs="2" />
                <blbPair quadrant="4" firstBlbPair="0" numBlbPairs="2" />
            </polProducts>
        </subBand>
        <subBand sbid="6"  bw="128000000" centralFreq="192000000" rqNumBits="4" >
            <polProducts>
                 <productIntegration recirculation="8" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10">
                <pp id="1" correlation="A*A" spectralChannels="512"/>
                <pp id="2" correlation="A*B" spectralChannels="512"/>
                <pp id="3" correlation="B*A" spectralChannels="512"/>
```

```
            <pp id="4" correlation="B*B" spectralChannels="512"/>
            <blbPair quadrant="3" firstBlbPair="6" numBlbPairs="1" />
        </polProducts>
    </subBand>
    <subBand sbid="7"  bw="128000000" centralFreq="320000000" rqNumBits="4" >
        <polProducts>
             <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10">
            <pp id="1" correlation="A*A" spectralChannels="64"/>
            <pp id="2" correlation="A*B" spectralChannels="64"/>
            <pp id="3" correlation="B*A" spectralChannels="64"/>
            <pp id="4" correlation="B*B" spectralChannels="64"/>
            <blbPair quadrant="3" firstBlbPair="7" numBlbPairs="1" />
        </polProducts>
    </subBand>
</baseBand>

<baseBand bbA="6" bbB="7" name="BBP4" bw="2048MHz" inQuant="3" singlePhaseCenter="yes" >
    <subBand sbid="2"  bw="128000000" centralFreq="64000000" rqNumBits="4" >
        <polProducts>
             <productIntegration recirculation="2" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10">
            <pp id="1" correlation="A*A" spectralChannels="512"/>
            <pp id="2" correlation="A*B" spectralChannels="512"/>
            <pp id="3" correlation="B*A" spectralChannels="512"/>
            <pp id="4" correlation="B*B" spectralChannels="512"/>
            <blbPair quadrant="3" firstBlbPair="2" numBlbPairs="2" />
            <blbPair quadrant="4" firstBlbPair="2" numBlbPairs="2" />
        </polProducts>
    </subBand>
```

```
<subBand sbid="6"  bw="128000000" centralFreq="192000000" rqNumBits="4" >
    <polProducts>
         <productIntegration recirculation="8" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10">
        <pp id="1" correlation="A*A" spectralChannels="512"/>
        <pp id="2" correlation="A*B" spectralChannels="512"/>
        <pp id="3" correlation="B*A" spectralChannels="512"/>
        <pp id="4" correlation="B*B" spectralChannels="512"/>
        <blbPair quadrant="4" firstBlbPair="6" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="7"  bw="128000000" centralFreq="320000000" rqNumBits="4" >
    <polProducts>
         <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10"/>
        <pp id="1" correlation="A*A" spectralChannels="64"/>
        <pp id="2" correlation="A*B" spectralChannels="64"/>
        <pp id="3" correlation="B*A" spectralChannels="64"/>
        <pp id="4" correlation="B*B" spectralChannels="64"/>
        <blbPair quadrant="4" firstBlbPair="7" numBlbPairs="1" />
    </polProducts>
</subBand>
<subBand sbid="15"  bw="128000000" centralFreq="1984000000" rqNumBits="4" >
    <polProducts>
         <productIntegration recirculation="1" minIntegTime="250" ccIntegFactor="5" ltaIntegFactor="10" cbeIntegFactor="10">
        <pp id="1" correlation="A*A" spectralChannels="256"/>
        <pp id="2" correlation="B*B" spectralChannels="256"/>
        <blbPair quadrant="2" firstBlbPair="15" numBlbPairs="1" />
        <blbPair quadrant="4" firstBlbPair="15" numBlbPairs="1" />
    </polProducts>
</subBand>
    </baseBand>
</subArray>
```

XML Example 16-2 VCI Configuration Request: stationHw, transmitted by VCI Client


```xml
<ns2:vciRequest version="3.9" timeStamp="2009-09-01T11:16:12.328-07:00" desc="StationHw-Station1-3bitIQ" msgId="100"
xsi:schemaLocation="http://www.nrc.ca/namespaces/widar http://www.nrc.ca/namespaces/widar/vciRequest.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <ns2:stationHw msgId="106" action="add" activationId="stationHwFor1-3IQ" name="Station1-3bitIQ" sid="1">
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-0" bbid="0"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-0" bbid="2"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-4" bbid="1"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-4" bbid="3"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-0" bbid="4"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-0" bbid="6"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-4" bbid="5"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-4" bbid="7"/>
        <ns2:noiseDiode phase="0.0" dutyCycle="50.0" periodMilliSec="10.0" status="enable"/>
        <ns2:antenna name="Antenna1" id="Antenna_1" type="EVLA"/>
    </ns2:stationHw>

    <ns2:activationTrigger activationId="stationHwFor1-3IQ" msgId="130"/>

</ns2:vciRequest>
```

XML Example 16-3 VCI Response: Acknowledgement, transmitted by CM


```
<ns2:vciResponse version="3.9" timeStamp="2010-09-20T20:01:24.489Z" msgId="11"
xsi:schemaLocation="http://www.nrc.ca/namespaces/widar  http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciResponse.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ns2:vciAck version="3.9" timeStamp="2010-09-20T20:01:24.490Z" desc="Added message(s) to the Configuration Queue" msgId="12">
    <ns2:refMessage>
      <ns2:activationTrigger activationId="stationHwFor1-3IQ" msgId="130"/>
    </ns2:refMessage>
  </ns2:vciAck>
  <ns2:vciAck version="3.9" timeStamp="2010-09-20T20:01:24.492Z" desc="Added message(s) to the Configuration Queue" msgId="13">
    <ns2:refMessage>
      <ns2:stationHw msgId="106" action="add" activationId="stationHwFor1-3IQ" name="Station1-3bitIQ" sid="1">
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-0" bbid="0"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-0" bbid="2"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-4" bbid="1"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-4" bbid="3"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-0" bbid="4"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-0" bbid="6"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-4" bbid="5"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-4" bbid="7"/>
        <ns2:noiseDiode phase="0.0" dutyCycle="50.0" periodMilliSec="10.0" status="enable"/>
        <ns2:antenna name="Antenna1" id="Antenna_1" type="EVLA"/>
      </ns2:stationHw>
    </ns2:refMessage>
  </ns2:vciAck>
</ns2:vciResponse>
```

XML Example 16-4 VCI Response: Configuration accepted, transmitted by CM


```xml
<ns2:vciResponse version="3.9" timeStamp="2010-09-20T20:01:28.713Z" desc="widar" msgId="16"
xsi:schemaLocation="http://www.nrc.ca/namespaces/widar  http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciResponse.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ns2:vciAccept version="3.9" timeStamp="2010-09-20T20:01:28.712Z" desc="mappingCompleted" msgId="14">
    <ns2:refMessage>
      <ns2:stationHw msgId="106" action="add" activationId="stationHwFor1-3IQ" name="Station1-3bitIQ" sid="1">
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-0" bbid="0"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-0" bbid="2"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-t-4" bbid="1"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-t-4" bbid="3"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-0" bbid="4"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-0" bbid="6"/>
        <ns2:baseBandHw dataPath="0" stationBoardMlid="s001-b-4" bbid="5"/>
        <ns2:baseBandHw dataPath="1" stationBoardMlid="s001-b-4" bbid="7"/>
        <ns2:noiseDiode phase="0.0" dutyCycle="50.0" periodMilliSec="10.0" status="enable"/>
        <ns2:antenna name="Antenna1" id="Antenna_1" type="EVLA"/>
      </ns2:stationHw>
    </ns2:refMessage>
  </ns2:vciAccept>
  <ns2:vciAccept version="3.9" timeStamp="2010-09-20T20:01:28.712Z" desc="mappingCompleted" msgId="15">
    <ns2:refMessage>
      <ns2:activationTrigger activationId="stationHwFor1-3IQ" msgId="130"/>
    </ns2:refMessage>
  </ns2:vciAccept>

</ns2:vciResponse>
```

XML Example 16-5 VCI Report: Configuration Status Report transmitted by CM


```
<ns2:vciResponse version="3.9" timeStamp="2010-09-20T20:01:28.767Z" desc="widar" msgId="18"
xsi:schemaLocation="http://www.nrc.ca/namespaces/widar  http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciResponse.xsd"
xmlns:ns2="http://www.nrc.ca/namespaces/widar" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ns2:vciReport timeStamp="2010-09-20T20:01:28.767Z" msgId="17" forActTime="2010-09-20T20:01:27.000Z">

    <ns2:report>
     Configuration Change Status for corrModelID=1
     activation time          = 2010-09-20 20:01:27.000
     active since             = 2010-09-20 20:01:28.715
     last msg. transmitted at = 2010-09-20 20:01:28.766

     Forward configuration to CMIBs/CBE started: 2010-09-20 20:01:28.722, Not complete.
     Notice: Station Boards   are administratively disabled.
     Notice: Cross-bar Boards are administratively disabled.
     Notice: Baseline Boards  are administratively disabled.
     Notice: Communication with CBE is administratively disabled.

     STB=s001-t-0 SID=1, BBID=0/2, configId=, Status admin=Disabled, oper=unknown, used=No , configChange=ComponentDisabled,
                                                  desc=Administratively disabled, transmissions=0
     STB=s001-t-4 SID=1, BBID=1/3, configId=, Status admin=Disabled, oper=unknown, used=No , configChange=ComponentDisabled,
                                                  desc=Administratively disabled, transmissions=0
     STB=s001-b-0 SID=1, BBID=4/6, configId=, Status admin=Disabled, oper=unknown, used=No , configChange=ComponentDisabled,
                                                  desc=Administratively disabled, transmissions=0
     STB=s001-b-4 SID=1, BBID=5/7, configId=, Status admin=Disabled, oper=unknown, used=No , configChange=ComponentDisabled,
                                                  desc=Administratively disabled, transmissions=0
     </ns2:report>

  </ns2:vciReport>

</ns2:vciResponse>
```

XML Example 16-6 CM Monitor & Control: query and response

```
<ns2:vciRequest msgId="0"
      xsi:schemaLocation="http://www.nrc.ca/namespaces/widar
                              http://www.nrc.ca/namespaces/widar/vciRequest.xsd"
      xmlns:ns2="http://www.nrc.ca/namespaces/widar"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <ns2:cmMonitorControl query="yes">
        <ns2:queryCfgStatus cfg="active"/>
    </ns2:cmMonitorControl>

</ns2:vciRequest>
```

```
<ns2:vciResponse version="3.6" timeStamp="2010-09-16T19:03:53.122Z" msgId="1"
       xsi:schemaLocation="http://www.nrc.ca/namespaces/widar
            http://www.aoc.nrao.edu/asg/widar/schemata/vci/vciResponse.xsd"
       xmlns:ns2="http://www.nrc.ca/namespaces/widar"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ns2:vciAck version="3.6" timeStamp="2010-09-16T19:03:53.123Z" desc="Response to query." msgId="3">
        <ns2:refMessage>
            <ns2:cmMonitorControl query="yes">
                <ns2:queryCfgStatus cfg="active"/>
            </ns2:cmMonitorControl>
        </ns2:refMessage>
    </ns2:vciAck>
    <ns2:vciReport timeStamp="2010-09-16T19:03:53.123Z" msgId="2" forActTime="2010-09-16T19:03:47.000Z">
        <ns2:report>
            Configuration Change Status for corrModelID=0
            activation time          = 2010-09-16 19:03:47.000
            active since             = 2010-09-16 19:03:47.701
            last msg. transmitted at = 2010-09-16 19:03:51.708

            No configuration change.
            Notice: Station Boards   are administratively disabled.
            Notice: Cross-bar Boards are administratively disabled.
            Notice: Baseline Boards  are administratively disabled.
            Notice: Communication with CBE is administratively disabled.
      </ns2:report>
    </ns2:vciReport>
    <ns2:cmMonitorControl vciSchemaValidation="no" sendQueryToCRM="no"  sendConfigToCBE="no"
                          sendConfigToXBBs="no"    sendConfigToBLBs="no" sendConfigToSTBs="no">
        <ns2:cfgQueue>Configuration Queue empty.</ns2:cfgQueue>
        <ns2:actQueue>Activation Queue empty.</ns2:actQueue>
        <ns2:ctrlQueue>CM Control queue is empty.</ns2:ctrlQueue>
        <ns2:cmLogging logCrmMessages="no" logCbeMessages="no"  logXbbMessages="no" logBlbMessages="no"
                       logStbMessages="no" logVciMessages="yes" logToFile="yes"     level="INFO"/>
        <ns2:vciReporting destPort="55230" destIpAddress="239.192.2.5" transmit="yes"/>
    </ns2:cmMonitorControl>
</ns2:vciResponse>
```

XML Example 16-7 Create Subarray – Configure and enable WBC products

```
<widar:vciRequest version="3.14" timeStamp="2012-11-15T11:16:12.328-07:00" msgId="297"
xmlns:widar="http://www.nrc.ca/namespaces/widar">
    <widar:subArray action="create" configId="wbc-example" activationId="create-wbc-example" msgId="291">
        <widar:stationInputOutput sid="list">
            <widar:station sid="1"/>
            <widar:station sid="2"/>
            <widar:station sid="3"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="0"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="1"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="2"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="3"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="4"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="R" bbid="5"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="6"/>
            <widar:bbParams sourceType="FORM" sourceId="0" sideband="lower" polarization="L" bbid="7"/>
            <widar:baseBand inQuant="8" bw="1024000000" swbbName="AC_8BIT" bbB="2" bbA="0">
                <widar:subBand sbid="1" swIndex="1" centralFreq="64000000" bw="128000000" rqNumBits="4">
                    <widar:polProducts>
                        <widar:pp spectralChannels="64" id="1" correlation="A*A"/>
                        <widar:pp spectralChannels="64" id="2" correlation="A*B"/>
                        <widar:pp spectralChannels="64" id="3" correlation="B*A"/>
                        <widar:pp spectralChannels="64" id="4" correlation="B*B"/>
                        <widar:blbProdIntegration recirculation="1" minIntegTime="250.0" ltaIntegFactor="4000" ccIntegFactor="1"/>
                        <widar:blbPair quadrant="1" numBlbPairs="1" firstBlbPair="0"/>
                        <widar:stationPacking algorithm="fourPerRowColumn"/>
                        <widar:productPacking algorithm="maxPack"/>
```

```xml
                        <widar:autoCorrSubset algorithm="halfStationsMaxProd"/>
                    </widar:polProducts>
                </widar:subBand>
                <widar:wpp id="1" correlation="A*A" spectralChannels="64" integFactor="100"/>
                <widar:wpp id="2" correlation="B*B" spectralChannels="64" integFactor="100"/>
            </widar:baseBand>
        </widar:stationInputOutput>
    </widar:subArray>
    <widar:activationTrigger msgId="295" activationId="create-wbc-example"/>
</widar:vciRequest>
```

XML Example 16-8 Modify Subarray – Disable WBC products

```xml
<widar:vciRequest version="3.14" timeStamp="2009-09-01T11:16:12.328-07:00" msgId="397"
xmlns:widar="http://www.nrc.ca/namespaces/widar">

<widar:subArray action="modify" configId="wbc-example" activationId="modify-wbc-example"  msgId="391">
    <widar:stationInputOutput sid="all">
        <widar:baseBand inQuant="8" bw="1024000000" swbbName="AC_8BIT" bbB="2" bbA="0"  noWbcProducts=" "/>
    </widar:stationInputOutput>
</widar:subArray>

<widar:activationTrigger msgId="395" activationId="modify-wbc-example" />
</widar:vciRequest>
```

Index