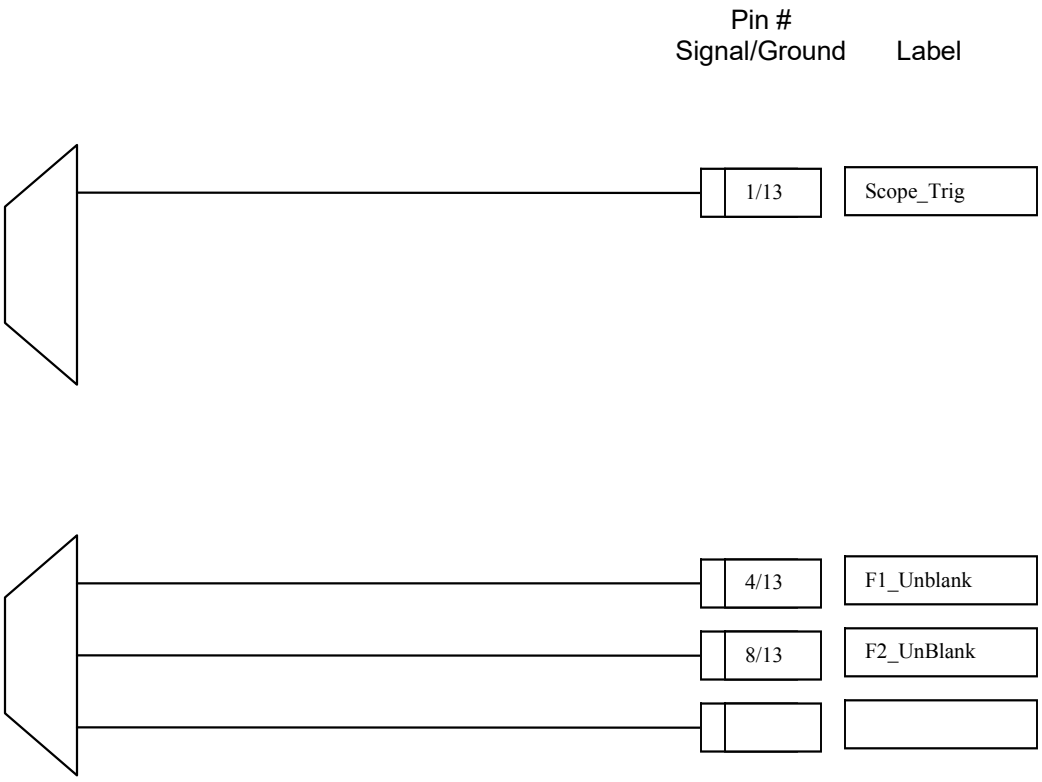
S/N 37863 DB25 LP Connectors



BP2 LP Outputs*

DB-25 pin	Signal	Config.con Assignment	Event Module
1	LP20		EM_Sys_3
2	LP21		
3	LP22		
4	LP23	F1_UnBlank	
14	GND	GND	
15	GND	GND	
16	LP22 inv	N/A	
17	LP23 inv	N/A	
5	LP20		EM_Sys_4
6	LP21		
7	LP22		
8	LP23	F2_UnBlank	
18	GND	GND	
19	GND	GND	
20	LP22 inv	N/A	
21	LP23 inv	N/A	
9	LP20		EM_Sys_5
10	LP21		
11	LP22		
12	LP23		
22	GND	GND	
23	GND	GND	
24	LP22 inv	N/A	
25	LP23 inv	N/A	
13	GND	GND	

* Pins / Signal addresses with no Config.con assignment, N/A, or GND are available for use. Contact Tecmag Support for assistance with enabling these signals.

BP1 LP Outputs*

DB-25 pin	Signal	Config.con Assignment	Event Module
1	LP20	Scope_Trig	EM_Sys_0
2	LP21		
3	LP22		
4	LP23		
14	GND	GND	
15	GND	GND	
16	LP22 inv	N/A	
17	LP23 inv	N/A	
5	LP20		EM_Sys_1
6	LP21		
7	LP22		
8	LP23		
18	GND	GND	
19	GND	GND	
20	LP22 inv	N/A	
21	LP23 inv	N/A	
9	LP20		EM_Sys_2
10	LP21		
11	LP22		
12	LP23		
22	GND	GND	
23	GND	GND	
24	LP22 inv	N/A	
25	LP23 inv	N/A	
13	GND	GND	

* Pins / Signal addresses with no Config.con assignment, N/A, or GND are available for use. Contact Tecmag Support for assistance with enabling these signals.

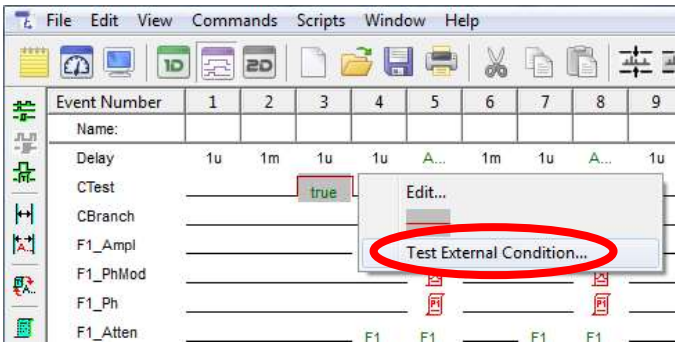
Hardware Interface for Conditional Branching:

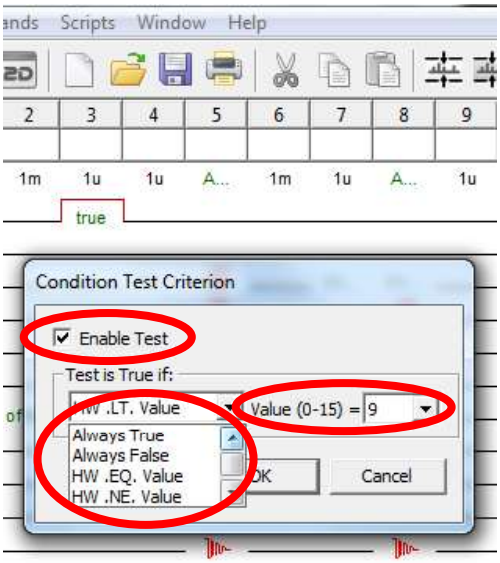
A DB-25 connector labeled “EXT IN” is provided on the Redstone back panel to interface external hardware signals for conditional branch control. Currently four bits are tested allowing 16 values (“conditions”).

EXT IN	
Bit	**Pin**
0	1
1	2
2	3
3	4
GND	25

CTest Logic Conditions:

The TNMR Sequence Editor provides a number of logic options for conditional branch testing.





“Test is True if:”	**Expression**
Always True	True for all Values
Always False	False for all Values
HW .EQ. Value	=
HW .NE. Value	≠
HW .GT. Value	>
HW .LT. Value	<
HW .GE. Value	≥
HW .LE. Value	≤

Example 2: Conditional ("Gated") Acquisition

The following example shows how to maintain a steady-state magnetization while acquiring signals only when an external hardware condition is satisfied. This is useful, for example, in respiratory gating, where data are only acquired during a restricted phase of the subject's breathing cycle.

In the example, events 8-13 and 14-19 have exactly the same time duration. Depending upon the state of the external hardware, the sequence will either:

- execute events 8-13, then branch to event 1 (branch2 – target2, in red), without incrementing the counters, and then repeat the same scan, or
- branch from 8 to 14 (branch1 – target1, in blue), then execute events 14-19, then increment the 1D/nD counters and advance to the next scan.

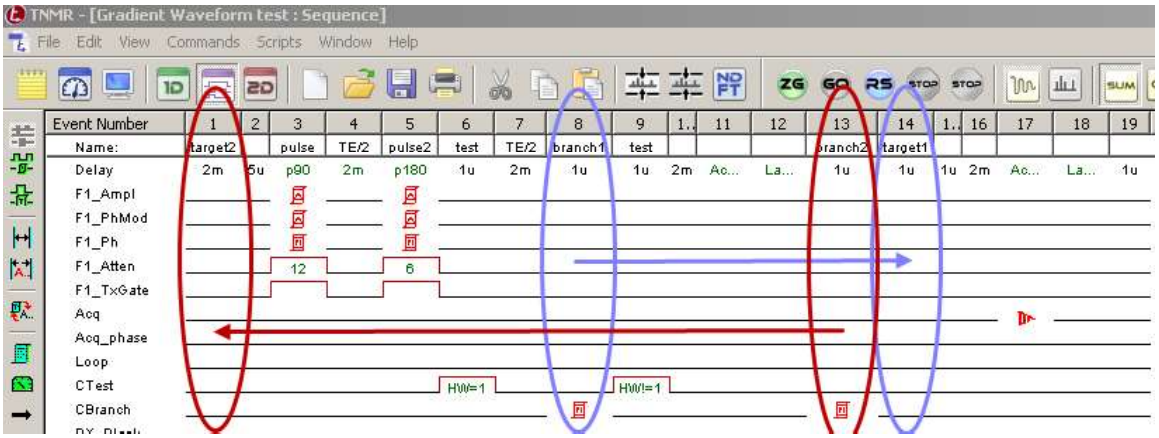
Pulse Programming notes:

There are two condition tests, in events 6, and 9, with opposite logic. The test in event 6 is true if hardware bit 0 is high (logic "1"), while the test in event 9 is false.

The CBranch table in event 8 has one entry, "14". The table in event 13, likewise has one entry, "1".

Be sure to count the events executed along each path and set the durations equal in order to maintain constant timing of the sequence.

(Note that this example assumes that the external condition bit is constant for the duration of the scan. This can be achieved, for example, by using an LP output in event 18 to latch in a new value).

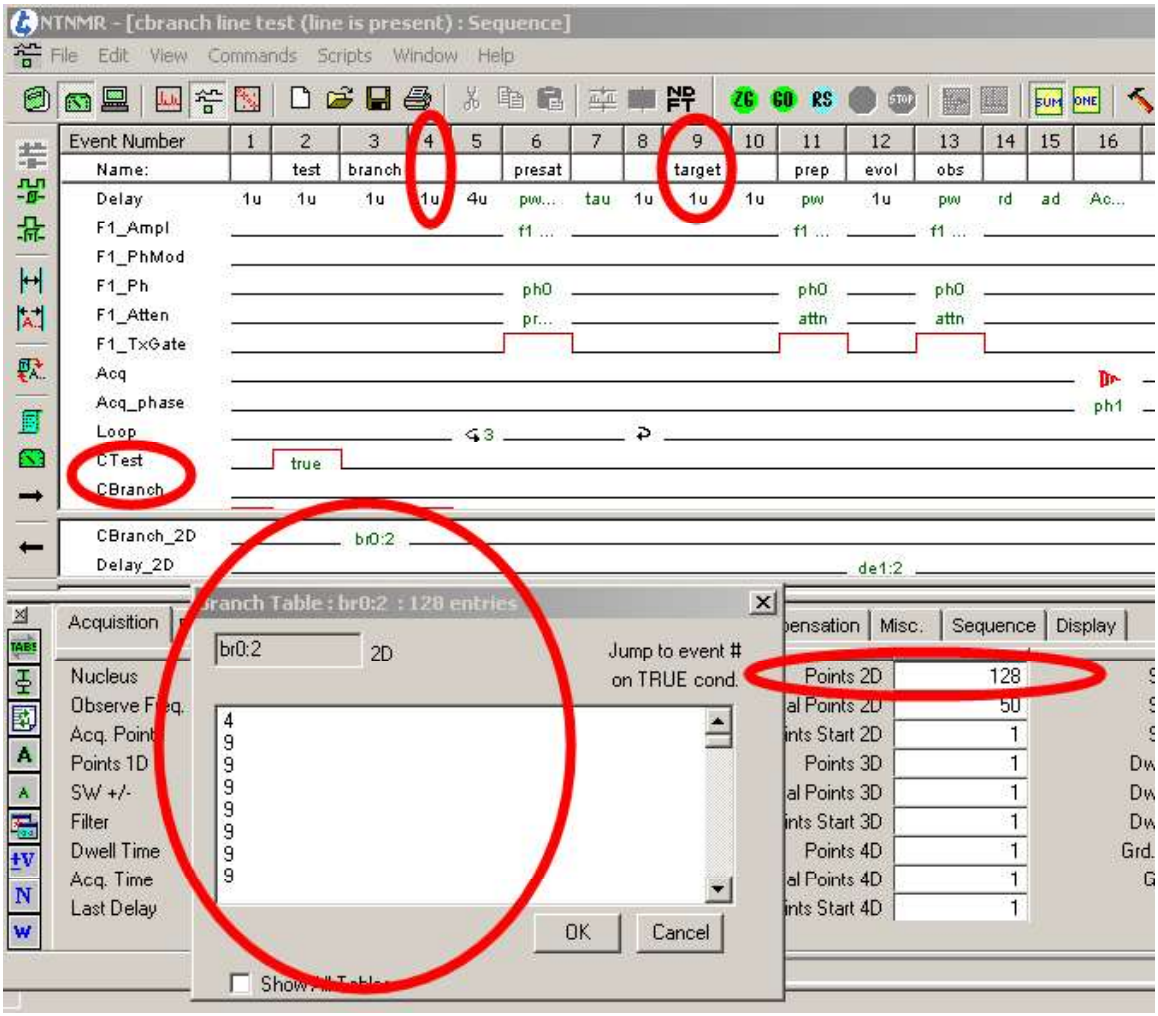


Example 1: 2D Preparation Sequence

The following example performs the presaturation loop (events 5-7) only once for each 2D plane. On the first scan, events 4-8 are executed (the branch is from event 3 to 4). On all subsequent scans, a branch from event 3 to 9 causes events 4-8 to be skipped. Note the extra event (#4) inserted before the loop start to provide a target for the first jump.

Pulse Programming procedure:

In event 2, place "Always True" on the CTest line.
In event 3, place a 2D table on the CBranch line.
Put the following entries in the table: 4, 9, 9, 9…. (the target event numbers, total 128 entries)



New Features with EM-III (continued)

Conditional Branching:
Conditional branching allows the execution of events in a pulse sequence in a different order, depending upon some **condition**. Such conditions might include:

- Scan count values
- Presence or absence of an external (hardware) signal (future)

The changed order of events results from a **branch**, from a branch event to a target event (The target is normally some event other than the next one in the sequence).

To allow the increased flexibility in pulse programming that results from conditional branches, some hardware and software features have been added to the Redstone system:

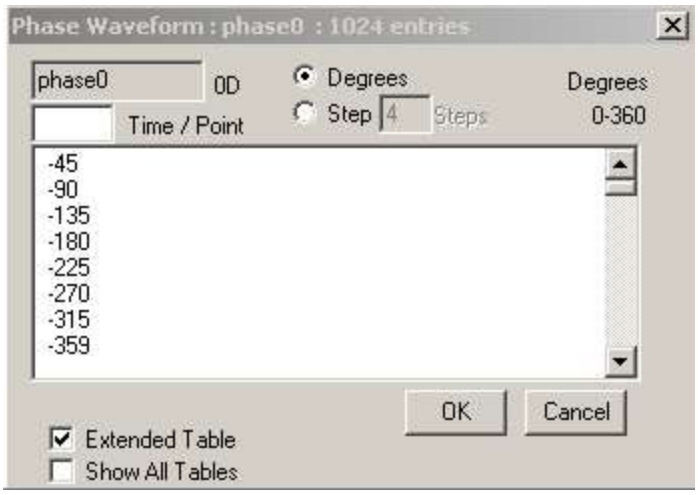
- A hardware state or condition input distributed to every EM-III (implemented with PCB rev C)
- A **CTest** sequence line to specify when the condition input is to be tested or detected, and what sort of test to perform. A CT event specified on this line will set an internal flag bit = 1 if the condition is true. The flag state is stored and persists until the next CT event occurs.
- A **CBranch** sequence line to specify the branch destinations. Tables on this line indicate the target event(s) for a branch. A branch occurs if the flag (from CTest) is true when the branch event is executed.

Some rules:

- Conditional branches apply only to Redstone/EM-III systems.
- Minimum time for CB/CT events is 1μs
- The condition must be tested BEFORE the branch event.
- The branch event and its target are always executed.
- Branches are always specified in tables. All branches are specified by the target event number as shown in the sequence editor. (If the number of events in the sequence is changed, the sequence editor will attempt to update the target event numbers, and warn the user if this is not possible).
- No branching INTO or OUT OF a loop (an entire loop may be skipped, however).
- No branching into another scan: i.e. beyond the START or END-OF-SCAN events for the current scan.
- No branch can skip any slow-io event (frequency hop, gradient rotation, or acquisition).
- nD tables: the highest dimension overrides any lower dimension branches (tables replace rather than add).
- A branch may not have a waveform table on the same event. However, the branch's target may have a waveform.
- A branch may not occur in the same event as an external trigger event.
- There is no protection against infinite loops

Limitations:

There are presently no software protections against illegal branches, infinite loops, or other problems that might cause the pulse programmer to lock up. Users must test their sequence under safe conditions (e.g. with the power amplifiers disabled).



To specify extended table behavior, check the box in the table editor, as shown above. This feature is available for "Icon types" A3, E3, G3, P3, R3, and W3.

Other notes:

- All tables in the same dimension, on the same event, and on the same module (e.g. amplitude and phase), must have the same setting of the "extended" flag. The compiler creates a composite table for this event, and uses the flag value from the last table it processes.
- The number of table entries should be different from the corresponding Scans nD or Points nD, in order to be useful.

New Features with EM-III

The Tecmag Redstone system is based on the latest Event Module—the EM-III. Below are detailed a few of the new features made possible with the EM-III including *Extended Tables, Conditional Branching,* and *Simultaneous Tables.*

Extended Tables:

Certain table types may now be "extended". This means that the table pointer is no longer reset along with the associated scan counter. Thus, (depending upon the number of entries) the table entries can become "asynchronous" from the scan counter. For example, suppose it is desired to apply a pseudo-random phase to the transmitter and receiver on each excitation, to accomplish so-called "rf-spoiling" of the transverse magnetization. In such a case, it would be undesirable for this phase pattern to repeat itself synchronously with the phase encoding in a 2D experiment (e.g., every 4th record). If we use an extended table, we can accomplish this, as illustrated in the chart below, which compares the phase from the following tables:

- Standard 1D table (0º, 180º)
- Standard 2D table (0º, 90º, 180º, 270º)

  vs.
- Extended 1D table (117º, 234º, 351º, 108º, 225º, 342º, 99º, 216º, 333º, 90º, 207º, 324º, 81º)

1D scan #	2D scan #	Phase from standard 1D table	Phase from standard 2D table	Net phase from standard tables	Phase from 1D extended table
1	1	0	0	**0**	**117**
2	1	180	0	**180**	**234**
1	2	0	90	**90**	**351**
2	2	180	90	**270**	**108**
1	3	0	180	**180**	**225**
2	3	180	180	**0**	**342**
1	4	0	270	**270**	**99**
2	4	180	270	**90**	**216**
1	5	0	0	**0**	**333**
2	5	180	0	**180**	**90**
1	6	0	90	**90**	**207**
2	6	180	90	**270**	**324**
1	7	0	180	**180**	**81**
2	7	180	180	**0**	**117**
1	8	0	270	**270**	**234**
2	8	180	270	**90**	**351**

In this example, 2D records 1, 5, and so on have the same phase with the standard tables. The 1D extended table repeats on the second scan of the 7th record, resulting in far less correlation with record number. Note that because this table is a **1D extended table**, the pointer increments with the **1D scan counter**.

Two Channel 5-bit Digital Attenuator

Introduction:

This Redstone incorporates a two channel digital attenuator module which incorporates two Hittite Microwave Corp. HMC470LP3 digital attenuators. Five-bit control is provided via the System Cluster backplane. The software interface is accomplished using the Ext_Gain line of the TNMR pulse sequence editor.

The attenuator allows 1.0 dB incremental attenuation from 0 dB to 31 dB (5-bits). However, due to the logic of the device, software control is presented as a "Gain" since a value of 31 (all 5 bits high logic) delivered to the device corresponds to 0 dB attenuation while a value of 0 (all 5 bits low) results in maximum attenuation. *NOTE: The two channels are controlled together using the same sequence editor control.*

Hardware Installation:

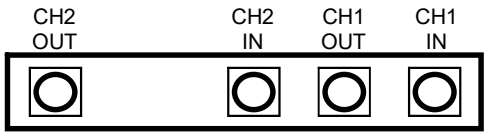
The attenuator hardware is factory installed as an internal component of the Redstone. It is installed in the System Cluster, and inserted in the cable path between the RX1 input on the Redstone back panel and the RX1 analog receiver input. All cabling has been installed in the factory; no additional installation is required. *NOTE: The SMA Port Assignments are illustrated below.*

Software Installation:

The /tnmr/config/config.con file has already been modified at the factory to incorporate the Ext_Gain control line for the pulse sequence.

Operating Instructions:

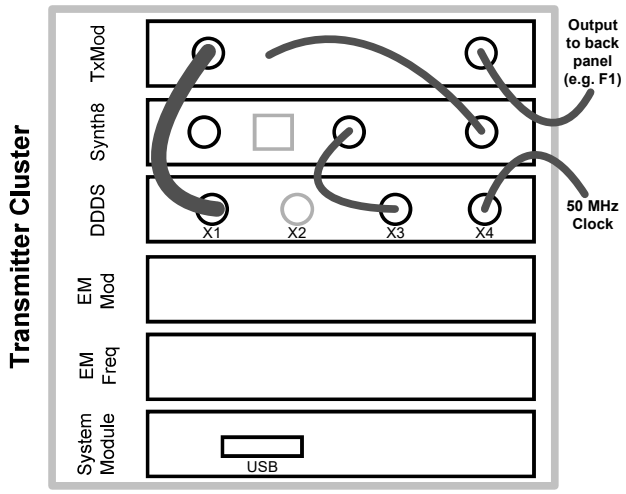
1. Add a "Gain" value or variable on the Ext_Gain line during every acquisition event in the TNMR pulse sequences.
2. For maximum signal, enter a value of 31. For minimum signal, use a variable set to zero. For 10 dB attenuation, enter 31 – 10 = 21.



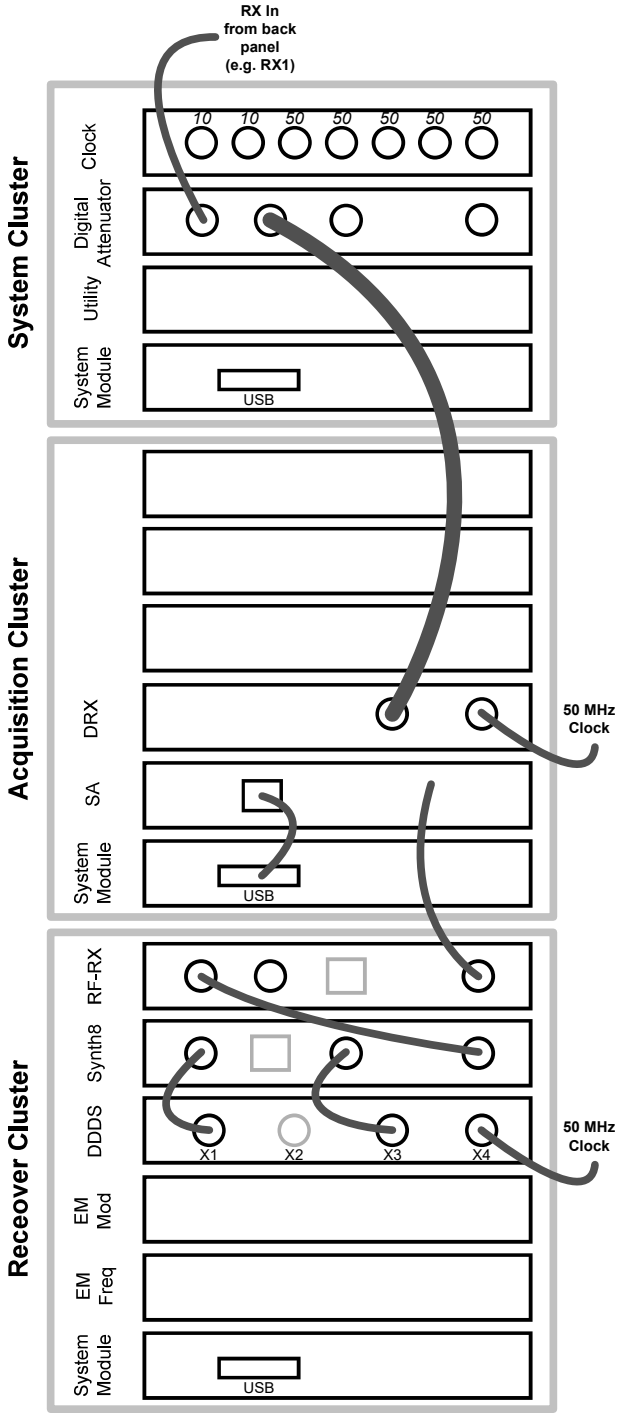
Tecmag Two Channel Digital Attenuator
SMA Port Assignments

Direct Digital Detection Cabling

Transmitter (Low Frequency)
***Bypass Synth8**

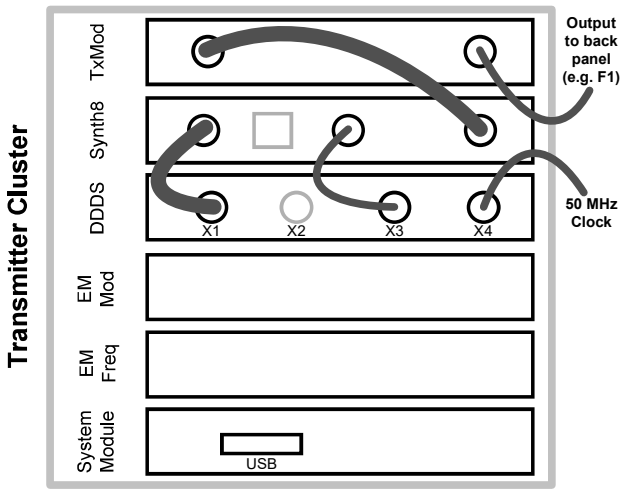


Receiver (Direct Digital Detection)

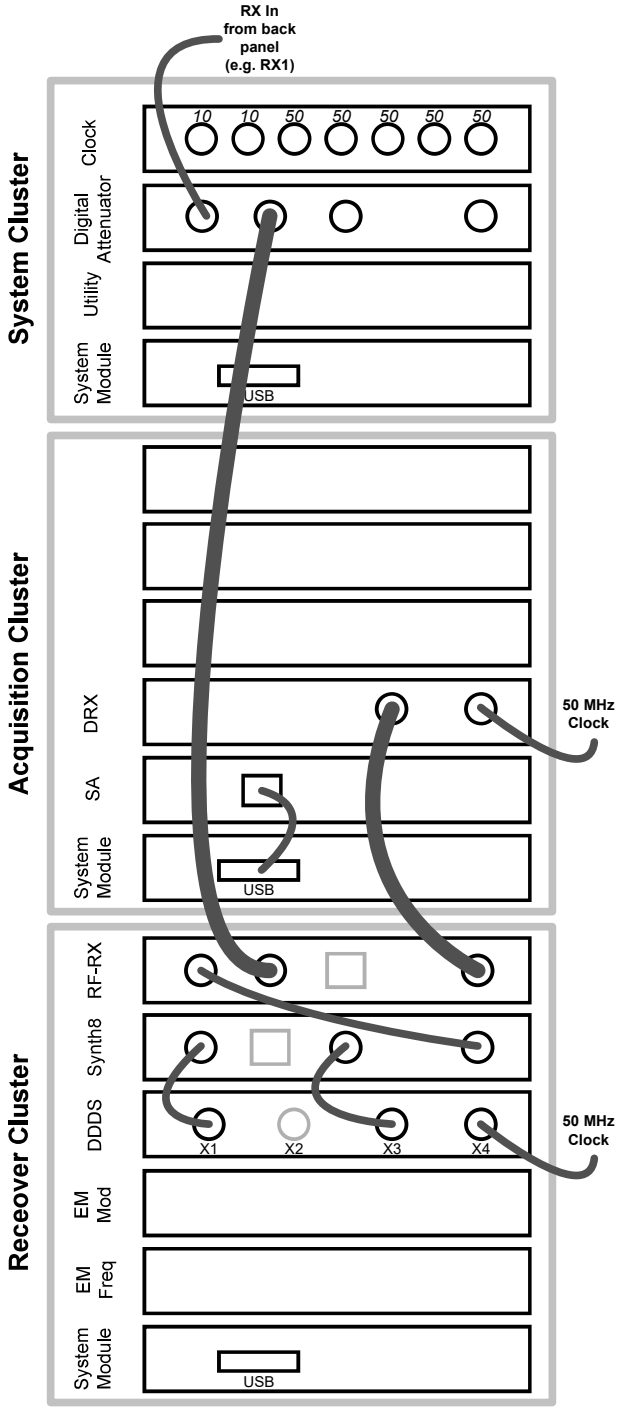


High Frequency (default) Cabling

Transmitter (High Frequency)



Receiver (High Frequency)



Converting to Direct Digital Detection (DDD) Configuration:

Redstone S/N 37863 is shipped in a "High Frequency" configuration in which both transmitters are capable of 0.5 to 500 MHz output, and an analog receiver is used over this range. It is possible, however, to reconfigure the Redstone to allow "Low Frequency" F1 transmitter output while bypassing the analog receiver for Direct Digital Detection (DDD). This low-frequency DDD mode is recommended for operation below 500 kHz; the F2 transmitter and analog receiver are not used in DDD mode.

To switch between configurations, it is necessary to make (a) hardware changes in the form of changing the cabling within the Redstone, (b) software changes in the form of modifying the /tnmr/config configuration files, and (c) TNMR preference changes.

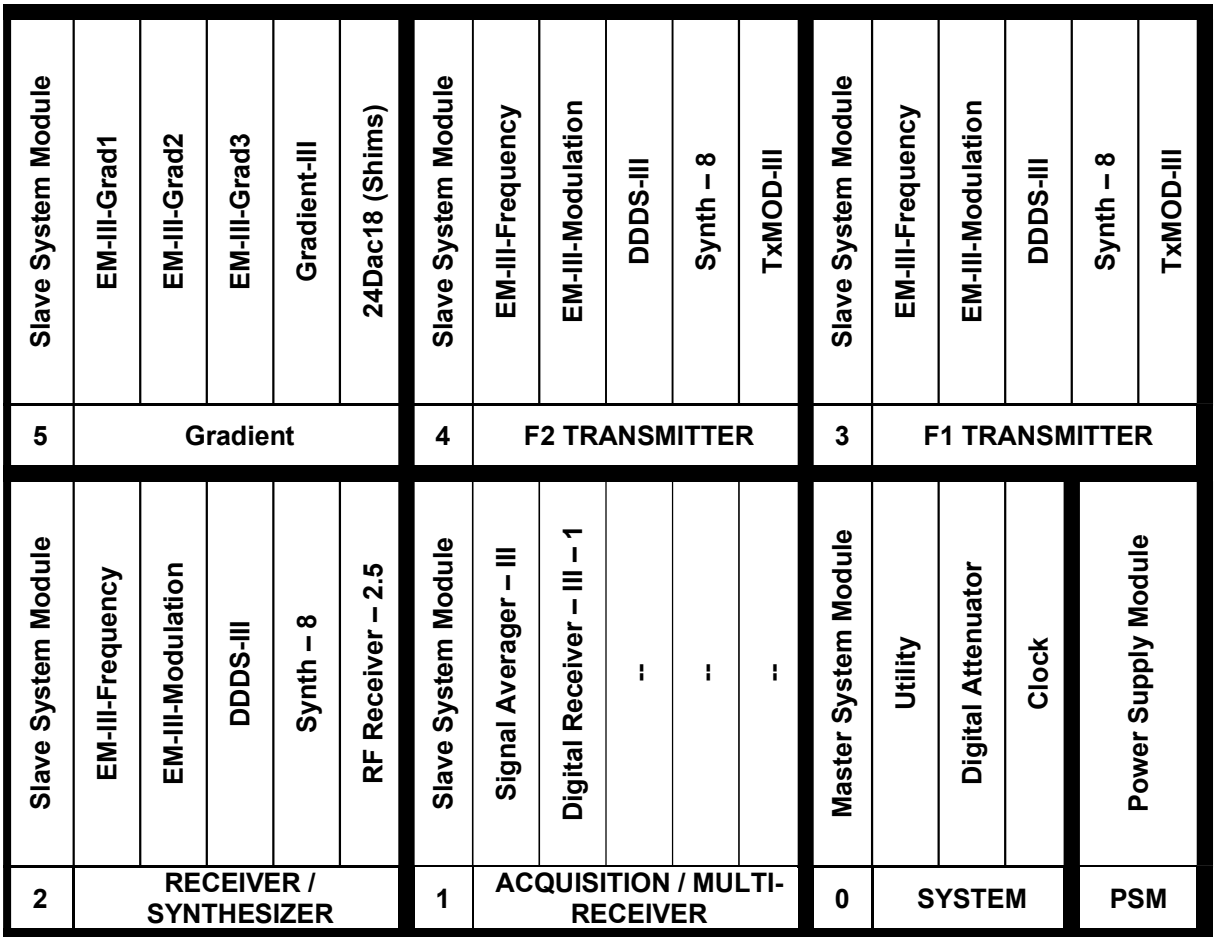
(a) **Hardware:** The diagrams on the following two pages will guide you in making the necessary cabling changes. Changes are indicated by bolder lines. An 8mm wrench can be used when working with the SMA connectors. It is recommended that the power to the unit be turned OFF while making the changes. To access the cables, remove the top cover by removing the screws along the sides and top of the cover.

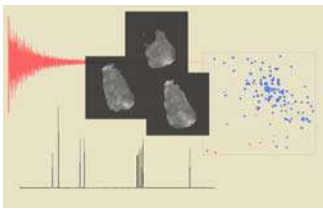
(b) **Software:** The software changes are performed by simply copying the appropriate files from the /tnmr/config subfolders to the root /tnmr/config folder. Restart TNMR after changing configuration files.
- For direct digital detection (DDD) mode, use the Apollo.ini and config.con files in the /tnmr/config/DDD subfolder.
- For high frequency mode, use the Apollo.ini and config.con files in the /tnmr/config/HF subfolder.

(c) **Prefereneces:** A simple preference change is necessary when switching between high frequency and DDD mode. In TNMR open the Edit Menu and click Preferences. On the Acquisition / Console tab, locate the "Reverse Direct Dimension" option box. Set the preference as follows:
- DDD Mode: Checked
- HF Mode: Unchecked

You can verify the software configuration by choosing Commands | Configuration Commands | Hardware Status. The Serial Number field at the top of the message box will report the serial number and configuration (See screenshot below).

Chassis Layout Diagram:





System Specific Information

The table below lists configuration information about the system which may be system specific. This table should be used as a reference when reviewing system specifications and capabilities found in the "Hardware Reference Manual".

Consult the "Hardware Reference Manual" for all system specifications.

General	
System	Redstone HF-2 MRI
Serial Number	37863
Hardware	
F1 & F2 Channels	
Freq Range	0.5 MHz - 500 MHz*
Phase/Amplitude Mod Type	EM-III
Freq Control Type	EM-III
Gradients	
Gradient Type	EM-III
Rotation Option	Yes
B_0 Compensation	No

Phase Reset

In order to ensure phase synchronization between all transmitters and receivers, all pulse sequences need to begin with a recommended 5 μs event (minimum duration) for the following sequence lines:

F0_PhRst F1_PhRst

RX_PhRst F2_PhRst

* Lower frequency operation is possible in a Direct Digital Detection configuration.