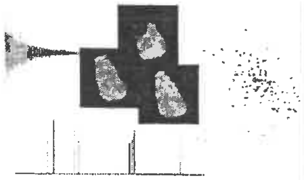


Hardware Reference Manual Table of Contents

1

>>> NEXT>>>

Hardware Descriptions

<<< PREV <<<

2

>>> NEXT >>>

Hardware Related Parameters & Pulse Sequence Line Descriptions

<<< PREV <<<

3

>>> NEXT >>>

Module Descriptions

<<< PREV <<<
4
>>> NEXT >>>

Hardware Specifications

<<< PREV <<<
5
>>> NEXT >>>

Accessing the User Definable Pulse Programmer Lines

<<< PREV <<<
6
>>> NEXT >>>

Frequency Control in Apollo/DSPect/Orion Systems

<<< PREV <<<

7

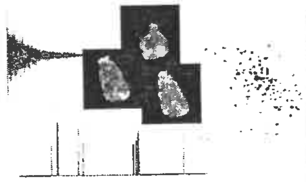
>>> NEXT >>>

System & Module Block Diagrams

<<< PREV <<<

A

Appendices



1 - Hardware Descriptions

1.1 Apollo - Low Frequency

>>> NEXT>>>

Introduction:
The Apollo is a compact, modular, multiple DSP, Windows NT based console, single RF channel that can be equipped with multiple digital receivers, and gradient waveform generators. Various options allow the Apollo LF/MF consoles to be configured for NMR, NQR or MRI applications.

RF Transmitter:
The RF transmitter channel is comprised of a DSP driven DDS based (Direct Digital Synthesizer) frequency synthesizer and a transmitter modulator board. The synthesizer board produces small angle phase shifts and the transmitter board supplies amplitude modulation. The final output from the transmitter has a maximum nominal output of 1V p-p into 50 ohms.

Digital Receiver:
The receiver is composed of two sections. The first RF section demodulates the signal to an intermediate frequency (IF) of 11.25 MHz. This section has been designed to maintain high dynamic range without sacrificing recovery time. In the second section, direct digitization and over sampling of the IF signal with 12-bit resolution (14-bit optional), followed by digital filtering with a bandwidth from ±2 Hz to ±300 kHz, eliminates baseline roll and quadrature detection errors.

For use of coil arrays, multiple digital receivers can be added to the Apollo console. A low noise preamplifier is not integrated into the receiver section of the Apollo to allow for end user flexiblility.

Pulse Programmer and Data Acquisition System:
The DSP driven pulse programmer has 100 ns timing resolution, 300 ns minimum pulse width, unlimited number of loop counters, and 1024 events. For each RF channel, there are eight spare TTL pulse lines available to control other equipment. If desired, additional control lines can be added in groups of 24. For synchronizing pulse sequences with external events, an external trigger input is provided.

The signal averager is controlled by an embedded PowerPC processor and is equipped with 4 Mword complex (2048 x 2048) memory, dedicated real-time display memory, and ultra fast upload capability. The signal averager memory is expandable to 32 Mword complex (512 x 256 x 256) memory with commercially available SIMMs.

Gradient Waveform Generators:
For imaging or diffusion applications, three gradient waveform generators are provided as an option. Each waveform generator has its own DSP with 3072 point library waveform memory and an optically isolated 18-bit DAC to control a gradient power supply. The optically isolation reduces noise and eliminates ground loops. Pre-emphasis time constants are computer adjusted and calculated on-the-fly by the DSP's. Bo compensation is also available.

Computer:
The APOLLO console uses an external Pentium class computer equipped with a large capacity hard disk, a CD-ROM drive, accelerated video, 10/100 Base T Ethernet, a keyboard and a mouse.

Windows NT operating system and NTNMR spectrometer control software are preloaded and customized to the specific configuration. NTNMR offers powerful automation capabilities and an easy to use graphic interface. Instrument control, pulse sequence editing, data processing, and printing are accomplished through the NTNMR interface. Consult the NTNMR manual for software details.

Module List (* = optional module that may not be present on all systems)

Module Name	Qty	Comment
Event Module	5	1 master (DB25), 2 slave, 2 SIO
Shape Module*	1	2 if B0 compensation option installed.
DAC-18*	3	Supplied in optically linked, external box. 6 if B0 compensation options installed.
LF Synthesizer (O1/O2)	1	Frequency range and phase shifting resolution specifications dependent on LF or MF system.
Digital Receiver	1	
Digital Receiver Utility	1	

RF Receiver	1	
Signal Averager	1	
Transmitter Modulator	1	
Clock	1	
System Interface	1	

<<< PREV <<<
1.2
>>> NEXT >>>

Apollo - High Frequency

Introduction:
The Apollo is a compact, modular, multiple DSP, Windows NT based console, single RF channel that can be equipped with up to 4 state of the art RF synthesizers, multiple digital receivers, and gradient waveform generators. Various options allow the Apollo LF/MF consoles to be configured for NMR, NQR or MRI applications.

RF Transmitter:
The RF transmitter channel is comprised of a DSP driven DDS based (Direct Digital Synthesizer) frequency synthesizer and a transmitter modulator board. The synthesizer board produces small angle phase shifts and the transmitter board supplies amplitude modulation. The final output from the transmitter has a maximum nominal output of 1V p-p into 50 ohms.

Digital Receiver:
The receiver is composed of two sections. The first RF section demodulates the signal to an intermediate frequency (IF) of 11.25 MHz. This section has been designed to maintain high dynamic range without sacrificing recovery time. In the second section, direct digitization and over sampling of the IF signal with 12-bit resolution (14-bit optional), followed by digital filtering with a bandwidth from ±2 Hz to ±300 kHz, eliminates baseline roll and quadrature detection errors.

For use of coil arrays, multiple digital receivers can be added to the Apollo console. A low noise preamplifier is not integrated into the receiver section of the Apollo to allow for end user flexiblility.

Pulse Programmer and Data Acquisition System:
The DSP driven pulse programmer has 100 ns timing resolution, 300 ns minimum pulse width, unlimited number of loop counters, and 1024 events. For each RF channel, there are eight spare TTL pulse lines available to control other equipment. If desired, additional control lines can be added in groups of 24. For

synchronizing pulse sequences with external events, an external trigger input is provided.

The signal averager is controlled by an embedded PowerPC processor and is equipped with 4 Mword complex (2048 x 2048) memory, dedicated real-time display memory, and ultra fast upload capability. The signal averager memory is expandable to 32 Mword complex (512 x 256 x 256) memory with commercially available SIMMs.

Gradient Waveform Generators:
For imaging or diffusion applications, three gradient waveform generators are provided as an option. Each waveform generator has its own DSP with 3072 point library waveform memory and an optically isolated 18-bit DAC to control a gradient power supply. The optically isolation reduces noise and eliminates ground loops. Pre-emphasis time constants are computer adjusted and calculated on-the-fly by the DSP's. Bo compensation is also available.

1
0
ERROR

Computer:
The APOLLO console uses an external Pentium class computer equipped with a large capacity hard disk, a CD-ROM drive, accelerated video, 10/100 Base T Ethernet, a keyboard and a mouse.

Windows NT operating system and NTNMR spectrometer control software are preloaded and customized to the specific configuration. NTNMR offers powerful automation capabilities and an easy to use graphic interface. Instrument control, pulse sequence editing, data processing, and printing are accomplished through the NTNMR interface. Consult the NTNMR manual for software details.

Module List (* = optional module that may not be present on all systems)

Module Name	**Qty**	**Comment**
Event Module		
Shape Module*	1	2 if B0 compensation option installed.
DAC-18*	3	Supplied in optically linked, external box. 6 if B0 compensation options installed.
LF Synthesizer (O1/O2)		Frequency range and phase shifting resolution specifications dependent on LF or MF system.

Digital Receiver	1	
Digital Receiver Utility	1	
RF Receiver	1	
Signal Averager	1	
Transmitter Modulator		
Clock	1	
System Interface	1	

<<< PREV <<<

1.3

>>> NEXT >>>

Orion Description

The ORION is a multi-purpose Data Acquisition System for NMR, NQR or MRI applications. Its numerous options meet the needs of any solid-state, imaging or spectroscopy experiment. Digital Signal Processors (DSP) have been used throughout the unit to optimize flexibility and expandability. The ORION is conveniently packaged in a compact 4U 19" rack mountable chassis.

The basic version of the Orion includes:

• a Pulse Programmer with 100 ns resolution, 300 ns minimum pulse width, unlimited number of loop counters, 80 control lines, and 1024 events. Each group of 24 output digital lines, or Event Module, is controlled by its own DSP. Individual lines can be user assigned and grouped to the various instrument control functions.

• a Signal Averager with an embedded PowerPC, 4 Mword complex memory (2048 X 2048), dedicated real-time display memory, ultra fast data upload capability, and a choice of dual channel ADC's. The signal averager memory is expandable to 32 Mword complex memory (512 x 256 x 256) with commercially available SIMMs.

Signal averager options are:
- F12: 12-bit, 1.0 μs/complex point (1 MHz).
- S16: 16-bit, 2.0 μs/complex point (500 kHz).

• an external Pentium type computer with a large capacity hard disk drive, a CD-ROM drive, accelerated video, Fast Ethernet, dual high speed PCI Interface cards, a keyboard and a mouse. Please contact Tecmag for the latest computer configuration.

• Windows NT operating system and NTNMR spectrometer control software preloaded and customized to the specific configuration. NTNMR offers powerful automation capabilities and an easy to use graphic interface. Multi-dimensional acquisitions and processing up to 4D are standard features. Since each system includes a NTNMR software site license, low cost personal computers can be used for off-line processing. For expanded 3D and 4D processing / display, NT-NMRpipe and NT-NMRdraw are bundled with each ORION.

Options that can be added to the basic ORION are:

Additional Event Modules, each containing 24 digital lines for up to 224 user-defined pulse programmer lines.

Shape Modules for generating shaped RF or gradient pulses. Each shape module contains three DSP waveform generators with high speed optically isolated 18-bit DACs. For imaging and diffusion applications, pre-emphasis calculations are computed by the DSP's in real time. Gradient Control and B0 compensation packages are available.

PTS Interface Modules.

Module List (* = optional module that may not be present on all systems)

Module Name	**Qty**	**Comment**
Event Module	4	
Shape Module*	1	2 if B0 compensation option installed.
DAC-18*	3	Supplied in optically linked, external box. 6 if B0 compensation options installed.
ADC Board	1	
Signal Averager	1	
Clock	1	
System Interface	1	

<<< PREV <<<
1.4
>>> NEXT >>>

DSPect Description

The DSPect is a multiple Digital Signal Processor (DSP), Windows NT based Data Acquisition Upgrade for Bruker NMR spectrometers with Aspect 3000 computers. The DSPect Real-Time NMR System is designed to replace the Aspect 3000 computer including the associated computer peripherals, digitizer, pulse programmer, offset oscillators and Fourier filters. All the existing functions including the use of the SCM (System Controlled Microprocessor) are retained.

The DSPect data acquisition system contains:

• a DSP Pulse Programmer with 100 ns resolution, 300 ns minimum pulse width, unlimited number of loop counters and 1024 events.

• a Signal Averager with an embedded PowerPC and 4 Mword complex (2048 X 2048) memory, dedicated real-time display memory, ultra fast data upload capability, and a choice of dual channel ADC's. The signal averager memory can be expanded to 32 Mword complex (512 X 256 X 256) memory using commercially available SIMMs.

Signal averager options are:
-F12:12-bit, 1.0 μs/complex point (1 MHz).
-S16:16-bit, 2.0 μs/complex point (500 kHz).

• an external Pentium type computer with a large capacity hard disk drive, a CD-ROM drive, accelerated video, Fast Ethernet, a keyboard and a mouse. Please contact Tecmag for the latest computer configuration.

• Fourier Filters, DDS offset oscillators, Bruker slow I/O, and Bruker interface modules.

The DSPect is entirely plug compatible with the original cables of the spectrometer. It has the same P2, P1, I3, A1 and SDC (Slow Device Channel) connectors as found on the Aspect 3000. The DSPect can be installed in less than an hour without any special tools on Bruker AC, AF, AM, NR, WM and WP spectrometers.

All the capabilities of the original spectrometer, such as lock control or shim adjustment, are retained by the DSPect. Also, all the functions of the SCM unit are under direct control of the DSPect. The O1/O2 offset oscillators are generated by direct digital synthesizers increasing instrument stability.

With NTNMR, the powerful NMR software included with the DSPect, instrument and experiment parameters can be set through pop-up windows. Any task such as locking, shimming, experiment setup, processing, and printing can be automated through NMRScripts(r) including all the functions of the SCM unit. NMRscripts can be written in Visual Basic or any programming language supporting OLE. Multi-dimensional acquisitions and processing up to 4D are standard features.

The NTNMR "RF Console" Window allows setup of instrument control commands. The functions include temperature, decoupler power and mode selection, F1 power and band selection, lock control, preamplifier selection, receiver gain, and transmitter offsets. This window provides a convenient setup of instrument parameters.

The NTNMR "Shim Panel" Window allows setup of computer shimming on the lock signal or FID. The lock level and the FID can be viewed simultaneously. Shim files can be saved and recalled later. On Bruker Aspect 3000 computer systems with a System Controlled Microprocessor (SCM) unit, the keypad can also be used for autolocking and shimming.

For imaging systems or spectrometers with a selective excitation unit, optional Shape Modules are available for generating shape RF or gradient pulses. Each waveform generator has its own DSP with 3072 point library waveform memory and an optically coupled 18-bit DAC. The optically isolation reduces noise and eliminates ground loops. Pre-emphasis time constants are computer adjusted and calculated on-the-fly by the DSP's.

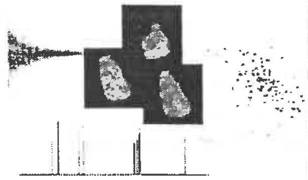
Module List (* = optional module that may not be present on all systems)

Module Name	Qty	Comment
Event Module		
Shape Module*	1	2 if B0 compensation option installed.
DAC-18*	3	Supplied in optically linked, external box. 6 if B0 compensation options installed.
LF Synthesizer (O1/O2)		Frequency range and phase shifting resolution specifications dependent on LF or MF system.
Digital Receiver	1	
Digital Receiver Utility	1	

RF Receiver	1	
Signal Averager	1	
Transmitter Modulator		
Clock	1	
System Interface	1	

DSPect Specifications

See the 'Hardware Specifications' chapter.



2 - Hardware Related Parameters & Pulse Sequence Lines

<<< PREV <<<
2.1
>>> NEXT >>>

Introduction

The following sections contain descriptions of the hardware related parameters that appear in the dashboard (and in the console for the case of DSPect systems) as well as descriptions of the pulse sequence lines that appear in the graphical pulse sequence editor. See the NTNMR manual for Dashboard items not discussed below.

<<< PREV <<<
2.2
>>> NEXT >>>

Apollo Systems - Hardware Related Dashboard Parameters

Dashboard Parameter	Allowed Values	Description
Receiver Gain	0 - 10,000	Controls the gain of the RF receiver section. Non-linear.
Acq. Points	2 - 64k	Sets the actual number of complex points to be acquired for each acquisition event in the sequence.
Points 1D	--	Calculated by NTNMR to be the total number of complex points acquired in the experiment.
SW +/-	300 - 22500000 Hz	The actual SW that is set is determined by NTNMR. User input numbers are converted to valid numbers for the digital receiver.
Dwell Time	1 / SW	
Filter		
F1 Freq. (F2, F3, F4)	System dependent values in MHz.	The actual frequency of the specified transmitter channel. F1 Freq. = F1 Base Freq. + F1 Offset Freq.
F1 Base (F2, F3, F4)	System dependent values in MHz.	A software only parameter that can be used to define a standard 'base' frequency.
F1 Offset Freq. (F2, F3, F4)	Values in (+/-) kHz.	A software only parameter that can be used to set an offset in kHz from the current frequency.

0 – 11010
accepted by SW.

Z where
O adjusted
?

Sequence Variables	variable	Sequence parameters are user definable variables that appear in the sequence tab on the dashboard. These variables may vary depending on the sequence. See below for the standard variables used by Tecmag in the provided sequences. Also see the chapter on pulse programming in the NTNMR manual for details on using variables.

<<< PREV <<<

2.3 Apollo Systems - Pulse Sequence Line Descriptions*

>>> NEXT >>>

The following descriptions are for a system equipped with a single RF transmitter channel. In the case of systems equipped with multiple RF transmitter channels, the 'F1' lines will be duplicated using F2, F3, and F4.

**Not all lines present on all systems*

F1_TxGate

Allowed Values:

On/Off gate
1 = on, 0 = off

Sequence Usage:

TX style on/off gate
1D - 4D tables
User defined sequence variables

Description:

Gates on and off the RF transmitter pulse for the F1 transmitter channel.

F1_Ph

Allowed Values:

x, y, -x, -y
0, 1, 2, 3
0, 90, 180, 270

Sequence Usage:

Hard coded icons x, y, -x, y
Tables with 0, 1, 2, 3
Tables with 0, 90, 180, 270
User defined dashboard variables, values of 0, 1, 2, 3

Description:

F1 transmitter phase cycling of 0, 90, 180, 270.

F1_PhMod

Allowed Values:

Any floating point value, positive or negative

Sequence Usage:

Hard coded values
Shape tables
1D - 4D tables
User defined dashboard variables

Description:

F1 phase modulation tables and fine phase control

F1_Ampl

Allowed Values:

0 to 100, floating point

Sequence Usage:

Hard coded values
Shape tables
1D - 4D tables
User defined dashboard variables

Description:

F1 amplitude modulation and fine amplitude control

F1_Atten

Allowed Values:

0 to 63.5, in steps of 0.5 (units are assumed and should not be entered).

Sequence Usage:

Hard coded values
Shape tables
1D - 4D tables
User defined dashboard variables

Description:

F1 attenuation, 0 to 63.5 dB in steps of 0.5 dB

F1_H/L

Allowed Values:

On/Off gate
1 = Low, attenuator active, 0 = High

Sequence Usage:

TX style on/off gate
1D - 4D tables
User defined sequence variables

Description:

Toggles on/off a user adjustable attenuator. This line is used as a high/low power control. The 'high' attenuation level can be user set via a potentiometer on the TX Modulator board. See the block diagram for information.

F1_Freq_I

Allowed Values:

Positive and negative values in Hz (units are implied and should not be used)

Sequence Usage:

1D - 4D Tables
User defined sequence variables

Description:

Adds the value indicated to the dashboard parameter F1 Freq. (*not* to F1 Base or F1 Offset).

This line, in combination with F1_Freq_Q, can be used to perform a real time frequency hop in the form of an offset to the current F1 frequency. Frequency hops on the F1_Freq_I line must be duplicated on the F1_Freq_Q line. Frequency hops are 'latched', meaning that to return to the current Dashboard frequency, a value of '0' must be programmed on both the F1_Freq_I and F1_Freq_Q lines.

Due to a sequence compiler limitation, a table must be used to enter a value of '0' in the F1_Freq_I line.

F1_Freq_I in the pulse sequence in *not* equivalent to F1 Offset in the dashboard.

F1_Freq_Q

See F1_Freq_I above.

F1_PhRst

Allowed Values:
On/Off gate
1 = on, 0 = off
Sequence Usage:
TX style on/off gate
1D - 4D tables
User defined sequence variables
Description:
Resets the phase to an absolute value of zero. This line should be set on for one event (approximately 1u) at the start of the pulse sequence.

F1_UnBlank

Allowed Values:
On/Off gate
1 = on, 0 = off
Sequence Usage:
TX style on/off gate
1D - 4D tables
User defined sequence variables
Description:
A spare TTL line on the master event module to be used to control amplifier blanking.

Gr_Shape

Allowed Values:

-100 to 100, floating point

Sequence Usage:

Shape tables

User defined sequence variables

Description:

Read gradient shape. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'X'~~ 'Z' channel.

Gr_Amp

Allowed Values:

-100 to 100, floating point

Sequence Usage:

TX style on/off gate

Shape tables

1D - 4D tables

User defined sequence variables

Description:

Read gradient amplitude. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'Y'~~ 'Z' channel.

Gp_Shape

Allowed Values:

-100 to 100, floating point

Sequence Usage:

Shape tables

User defined sequence variables

Description:

Phase gradient shape. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'Y'~~ 'Z' channel.

Gp_Amp

Allowed Values:

-100 to 100, floating point

Sequence Usage:

TX style on/off gate
Shape tables
1D - 4D tables
User defined sequence variables

Description:

Phase gradient amplitude. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'Y'~~ 'Z' channel.

Gs_Shape

Allowed Values:

-100 to 100, floating point

Sequence Usage:

Shape tables
User defined sequence variables

Description:

Slice gradient shape. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'Y'~~ 'Z' channel.

Gs_Amp

Allowed Values:

-100 to 100, floating point

Sequence Usage:

TX style on/off gate
Shape tables
1D - 4D tables
User defined sequence variables

Description:

Slice gradient amplitude. Note that the Dashboard parameter 'Grad. Orientation' can be used to reassign the sequence lines. By default (Grad. Orientation = XYZ) the Gr lines are assigned to the hardware 'X' channel, the Gp lines are assigned to the hardware 'Y' channel, and the ~~Gp~~ Gs lines are assigned to the hardware ~~'Y'~~ 'Z' channel.

Acq

Allowed Values:
Acq. Points in Dashboard
Sequence Usage:
Acquisition icon
Description:
Instructs hardware to begin acquiring data

Acq_phase

Allowed Values:
x, y, -x, y
0, 1, 2, 3
Sequence Usage:
Acquisition phase

Description:
Acquisition phase

RX_Blank

Allowed Values:
On/Off gate
1 = on, 0 = off
Sequence Usage:
Receiver blank
Description:
The RF receiver (analog section of the receiver) should be blanked when not detecting an NMR signal. Typical the RX_Blank line would be set to the on position in all sequence events except for the event immediately preceding the acquisition and the acquisition event (ACQ line).

DRX_Gate

not there!! 2.1

Allowed Values:
On/Off gate
1 = on, 0 = off
Sequence Usage:
Digital Receiver Gating
Description:

The DRX_Gate line activates the digital receiver section of the receiver and begins acquiring data. The DRX_Gate should be turned on during the acquisition event (ACQ line).

RX_Freq_I

Allowed Values:

Positive and negative values in Hz (units are implied and should not be used)

Sequence Usage:

1D - 4D Tables

User defined sequence variables

Description:

? Error ?

This line, in combination with RX_Freq_Q, can be used to perform a real time frequency hop in the form of an offset to receiver frequency. Frequency hops on the RX _Freq_I line must be duplicated on the F1_Freq_Q line. Frequency hops are 'latched', meaning that to return to the current Dashboard frequency, a value of '0' must be programmed on both the RX _Freq_I and RX _Freq_Q lines.

Due to a sequence compiler limitation, a table must be used to enter a value of '0' in the RX _Freq_I line.

RX_Freq_Q

Allowed Values:

Positive and negative values in Hz (units are implied and should not be used)

Sequence Usage:

1D - 4D Tables

User defined sequence variables

Description:

? Error ?

This line, in combination with RX_Freq_I, can be used to perform a real time frequency hop in the form of an offset to receiver frequency. Frequency hops on the RX _Freq_Q line must be duplicated on the F1_Freq_I line. Frequency hops are 'latched', meaning that to return to the current Dashboard frequency, a value of '0' must be programmed on both the RX _Freq_I and RX _Freq_Q lines.

Due to a sequence compiler limitation, a table must be used to enter a value of '0' in the RX _Freq_Q line.

RX_PhRst

Allowed Values:
On/Off gate
1 = on, 0 = off

Sequence Usage:
TX style on/off gate
1D - 4D tables
User defined sequence variables

Description:
Resets the phase to an absolute value of zero. This line should be set on for one event (approximately 1u) at the start of the pulse sequence.

Ext_Trig

Allowed Values:
On/Off gate
1 = wait

Sequence Usage:
External trigger

Description:
This line in the pulse sequence can be used to synchronize the spectrometer on an external source. The pulse programmer will be held in a wait state while the Ext_Trig line is high until an appropriate signal is detected on the External Trigger input on the back of the DSPect. *See the External Trigger section below.*

Loop

Description:
See the NTNMR Reference manual for a description of the loop line

3 - Module Descriptions

<<< PREV <<<
3.1
>>> NEXT >>>

Introduction

The following section contains brief descriptions of the various modules that are contained in the Apollo/Orion/DSPect systems. Block diagrams each module are presented in the **Block Diagrams** section and complete system specifications are given in the **Hardware Specifications** section.

> *The **Hardware Specification** section should be consulted for system specifications and all specifications in the **Hardware Specifications** section override those listed below in the Module Descriptions section.*

<<< PREV <<<
3.2
>>> NEXT >>>

Descriptions

Event Module

Description:

The heart of the pulse programmer is the Event Module. The Event Module is a DSP-based pulse programmer module with 24 output lines. Most event modules are defined as 'master' or as 'slave'. Typically, each system has only one master, but may have many slaves. The master event modules controls the timing for all real time events in the system, including events that occur on slave modules. This architecture allows virtually unlimited expansion of control lines. Additional event modules may be present is a system for the specific control of other devices. in

Error

Some event modules provide logic in the form of DB25 connectors or in some cases Burndy connectors. All logic lines that are user accessible are positive TTL logic.

Specifications:

resolution	100ns
minimum event	300ns

maximum events	1024	
loop counters	unlimited	
# of outputs	24	
output	TTL positive	
output impedance:		
	lines on backplane:	1 M-ohm
	lines on mezzanine:	25 ohm
shape tables:		
	min time/pt	0.1us
	max time/pt	2184.5us

/

0

Output Map:

The table below outlines the location of the output logic lines on the Event Module. Note that in the case of an Event Module that is defined as a master, not all lines are available externally as some lines are reserved for internal use. The exact lines that are used internally on the master is system dependent. The reserved lines can be determined by consulting the [EM Master] key in the 'Apollo.ini' file, located in the 'system' directory. Also note that some lines on both slave and master modules may have been assigned at the factory in the config.con.

DB25 Pin Number	Output Line Assignment
01	0
02	1
03	2
04	3
05	4
06	5
07	6
08	7
09	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
21	20
22	21
23	22
24	23
25	GND

Event Module Mezzanine Boards (P2, I3P1, SDC, DB25)

Description:

Event modules that bring logic lines to the back panel of the unit are fitted with a mezzanine board. Application specific mezzanine boards exist with Burndy-style connectors (DSPect) and with DB25 connectors. See the above section for the output map for Event Modules fitted with the DB25 mezzanine board.

Specifications:

output impedance	25 ohm

Shape Module (SM1 - 3Ch , SM2 - 2Ch)

Description:

The shape module is a DSP-based waveform generator with either 2 or 3 channels. Each channel communicates with a DAC via an optical link. Pre-emphasis calculations are performed on-the-fly by the DSP with three time constants.

Preemphasis Calculation Formula

$$S_{x\_out}(t) = \{ S_{x\_in}(t) [A_{0x} + A_{1x} \exp(-t / T_{1x}) + A_{2x} \exp (-t / T_{2x}) + A_{3x} \exp (-t / T_{3x})] + DC_x \} / 100;$$

$$S_{y\_out}(t) = \{ S_{y\_in}(t) [A_{0y} + A_{1y} \exp(-t / T_{1y}) + A_{2y} \exp (-t / T_{2y}) + A_{3y} \exp (-t / T_{3y})] + DC_y \} / 100;$$

$$S_{z\_out}(t) = \{ S_{z\_in}(t) [A_{0z} + A_{1z} \exp(-t / T_{1z}) + A_{2z} \exp (-t / T_{2z}) + A_{3z} \exp (-t / T_{3z})] + DC_z \} / 100.$$

Where $S_{in}(t)$ is the input of preemphasis calculation, and $S_{out}(t)$ is the output of preemphasis calculation;
$A_{0x}, \ldots, A_{3x}, A_{0y}, \ldots, A_{3y}, A_{0z}, \ldots, A_{3z}, DC_x, DC_y, DC_z$: [-100, +100];
$|A_{0x} + A_{1x} + A_{2x} + A_{3x}| <= 100;$
$|A_{0y} + A_{1y} + A_{2y} + A_{3y}| <= 100;$
$|A_{0z} + A_{1z} + A_{2z} + A_{3z}| <= 100;$
T_{1x}, T_{1y}, T_{1z} : [20us, 4000us];
T_{2x}, T_{2y}, T_{2z} : [1ms, 200ms];
T_{3x}, T_{3y}, T_{3z} : [0.05s, 10s].

Specifications:

Resolution	18 bits
Max points/waveform	2048
Waveform memory	3072 pts
min time/pt	3.3us
max time/pt	550ms
minimum delay between shape tables	300ns

18 & 20 Bit DAC Board

Description:

The waveform from the DSP-based shape module is transferred via an optical link to the DAC board. Waveforms are output from the DAC board via BNC coax or twinax connectors.

Specifications:

Max Output	+/- 10 V, adjustable
Resolution	18 or 20 bits
Data Output Rate	3.3 us / pt.
Load Impedance	> 2k Ohm

ADC - 16bit & 12bit

Description:

The ADC board is used in systems without digital receivers. Two version are available - 12 bit, 1MHz and 16 bit, 500 kHz. The ADC board samples that demodulated (AF) NMR signal source from an appropriate quadrature (real and imaginary) RF receiver.

Specifications:

16 bit: DEX-2

Max Input	-5 to +5V
Max Sampling Rate	500 kHz

12 bit:

Max Input	-2.5 to +2.5V
Max Sampling Rate	1 MHz

12 bit board connections

O	Ch1 Input
O	Ch2 Input

16 bit board connections

O	Ch1 Input
o	Ch1 gain adjust
o	Ch1 offset adjust
O	Ch2 Input
o	Ch2 gain adjust
o	Ch2 offset adjust

Signal Averager

Description:

The signal averager performs all basic data acquisition and memory management functions, including sum-to-memory for averaging. An embedded PowerPC (TM) 403 processor controls all functions. Incoming data is processed through a FIFO to prevent loss of data and permit acquisition in burst mode (under development 6/98). Separate memory blocks are provided to store the current scan as well as the summed scans for uploading to the host during or after a scan. Data memory uses conventional SIMM modules, with 32Mbytes standard. The signal averager consists of two modules: SA-Mem, the memory module, and SA-PPC, the logic module.

Specifications:

memory	32 mb standard, expandable to 256 mb
memory type	72 pin SIMM, non-parity, 60 ns minimum

PTS interface

Description:

PTS synthesizers with BCD interfaces (*not* GPIB) can be controlled by the combination of a standard event module equipped with a DB25 mezzanine board and a PTS interface module. One event module is capable of controlling up to four PTSs; however, each PTStobecontrolled requires a PTS interface module. The PTS interface module connects directly to the back of the PTS and connects via a ribbon cable to the controlling event module.

Error

Synthesizer (Low Frequency Apollo Systems & DSPect Systems)

Description:

The DDDS (referred to as the Dual DDS or sometimes the O1/O2 board) comes in several versions dictated by the specific application. For DSPect systems where he O1/O2 board replaces the original Bruker O1/O2 board, there are two versions: 8 bit TTL which produces kHz range TTL level outputs, and standard 8 bit which produces RF outputs in the low MHz ranges. All other systems use a version of the 16 bit module that is designed for a particular frequency range.

Error

Specifications:

output	0 dBm nominal
freq range:	10kHz - 12 MHz
	12 Mhz - 24 MHz
	38 MHz - 62 MHz

DDDS Board Connections:

O	50 MHz in
O	DDS1 out
O	DDS2 out

Synthesizer (High Frequency Apollo Systems)

Description:

The HF synthesizer (5-450 MHz) consists of 3 boards plus the associated controls (EMs): Synth5, 900Gen, and DDDS. Coarse frequency control is obtained from the Synth5, which can be set in steps of 3.125 MHz (under direct control from the bus, no EM being associated with it). Fine control of frequency, real-time frequency changes, and phase modulation are done using the direct digital synthesizers (DDS) on the DDDS board. The 900Gen provides the appropriate bumper frequency.

900 Board Connections

O	900 MHz out
O	900 MHz test
O	Q input
O	I input
O	50 MHz out
O	50 MHz in

Synth5 Board Connections

O	PLL out (test only)
O	900 MHz in
O	RF out
O	50 MHz in

TX Modulation

Description:

The TX modulation module is responsible for RF shape and amplitude. There is one TX Modulation board per RF channel on Apollo systems. Through a series of multipliers and DACs, the RF input signal from a Synth5 modules is modified for output.

Specifications:

Gain	1, (0 dBm in -> 0 dBm out)
switching time	< 20 ns
output impedance	50 ohm
nominal imput	0 dBm

Digital Receiver

Description:

The Apollo series receiver is composed of two sections. The first section demodulates the RF signal to an intermediate frequency of 11.25 MHz (see the IF recceiver section). The second section directly digitizes the signal. Over sampling, Undersampling, direct digital quadrature detection, and real-time digital filtering techniques are employed.

Specifications:

ADC	14 bit, 1MHz

DRX Utility Board

Description:

The digital receiver utility board performs several functions related to the digital receiver and acquisition: 1) 45 MHz and 11.25 MHz outputs are generated for the digital receiver, 2) supplies control logic for the RF receiver section, and 3) the external trigger signal is processed and supplied to the master event module.

Connections:

O	RX in, receiver reference frequency (TX +/- 11.25 MHz)
O	Filter input
O	Filter output to digital receiver
O	*External Trigger input
O	NMR Signal @ IF, demodulated signal from RF receiver section

*External Trigger Input

The external trigger input can be positive or negative logic. The external trigger circuit is capable of providing a positive or negative trigger signal to the master event module. Currently, however, the master event module MUST have a negative control signal. Using the jumpers U15 and U21 the external trigger circuit on the DRX Utility board can be configured to trigger on the rising or falling edge (U15) of the input and to provide negative of positive logic (U21) to the master event module.

U15 (rising or falling edge)

	F		R
falling edge	--	--	
rising edge		--	--

U21 (positive or negative <u>output</u> logic)

	N		P
negative	--	--	
positive		--	--

RF (Analog) Receiver

Description:

The RF/IF receiver demodulates the NMR signal to 11.25 MHz for detection by the digital receiver.

Specifications:

Gain	10 - 70 dB
Bandwidth	4 MHz
Noise Figure (no preamp)	6 dB
Maximum Input Power	-37 dBm (10 mV p-p, 50 ohm)
Dynamic Range	70 dB

Clock

Description:

The clock board obtains 10 MHz either from an on-board oscillator (high-stability oven-controlled crystal oscillator, OCXO, or a temperature-compensated crystal oscillator, TCXO), or from an external source. A 15 MHz TTL clock signal is generated for use by the DSP processors on the event modules. 50 MHz sine-wave signals are generated for the various synthesizer components. A 10 MHz sinewave output is also available to drive a secondary clock. The block diagram shows the relationship of the various signals.

Connections:

O	10 MHz in/out
O	10 MHz out*
O	50 MHz out
O	50 MHz out**
O	50 MHz out**
O	50 MHz out**

* Square wave output on Orion/DSPect systems

** Not present on Orion/DSPect systems

Converting from Internal to External Clock

Internal/external 10 MHz clock conversion is accomplished by switching the jumper on JP1, JP2, and JP3. When operating in external mode a clean, 1 V p-p (50 ohms), sine wave should be supplied to the 10 MHz in port. System performance and reliability will suffer with poor quality 10 MHz external clock signals.

	1	2	3
external	----	----	
internal		----	----

PCI Interface Cards

Description:

There are two PCI cards: PCI-SA and PCI-SI. The PCI-SA card facilitates communication between the PC and the signal averager only. PCI-SA is used for uploading data to the PC during and after data acquisition. The PCI-SI card is responsible for all other communication between the PC and the system. PCI-SI uploads pulse program data from the PC to the system interface board.

Card location in the host PC: The location of the PCI cards in the PC is important. Systems are pre-configured by Tecmag and, unless absolutely necessary, the cards should not be moved. The specific slot location is not critical. However, the order of the cards in the PCI slots is important. The order of the cards can be reversed under certain circumstances by editing the "PCI reverse" setting in the "Apollo.ini" file and by editing the Windows NT Registry.

System Interface

Description:

The system interface board distributes data received from the PC via the PCI-SI card to the various modules in the system.

ours : (mixed?) G2 Phase mod. / Grad. type / Ampl. mod G1 for frequ. control

4 - Hardware Specifications

Introduction

This chapter outlines hardware specifications for the Apollo series hardware. Note that some specifications are dependent on the hardware configuration. Consult the System Information document in the Installation manual to verify your exact hardware configuration.

In general, systems may be 'G2-based' or 'G1-based', although there is the possibility of the system being 'mixed'. The System Information document will indicate the type of system for each major system component. It will also indicate the frequency range of the RF channels present and the type of gradient system present, if applicable.

Section 4.2 (Sequence Related Specifications) contains specifications for 'G2-based' systems and for 'G1-based' systems. All other specifications are the same for all systems.

If the System Information document indicates that the Phase Mod and Amplitude Mod types are 'G2', then consult the G2-based table for RF related Sequence specifications. If G1 type is indicated for these functions, consult the G1 Sequence specifications. Similarly, if G2 gradients are specified consult the G2-based table for Gradient specifications and if G1 gradients are specified consult the G1 table.

If your system was shipped prior to 9/20/99, G1 specifications apply.

<<< PREV <<<

4.1

>>> NEXT >>>

Acquisition Related Specifications

Specification	Value	Notes
No. Points 1D	even, ≥ 2, and: ≤ 2^{28} =256M (ADC) ≤ 2^{29} (DRX) =512M ≤ 4096 (DRX, fast mode) ≤ 1024 (SA burst mode)	Complex points (all values subject to total point limits, see below)
No. Points 2D	≤ 64k-1	(subject to total point limits, see below)
Total Points per expt.	≤ SA memory (bytes)/8	Max= 256 Mbytes = 32M complex Std = 32Mbytes = 4M complex 256x256x64 = 32Mbyte 512x256x256 = 256Mbyte
No. Scans 1D	< 64k	
No. Scans 1D without overflow	about 65536	With standard 16-bit precision, and signal filling ADC
Real-time display	~32k points (transient mode) full data set (sum mode)	exact length varies according to factoring, if data length > 32k. sum mode may show acquisition in progress
Upload times - 450 MHz Host	typ 15-20 µsec/ complex point	For large data sets

<<< PREV <<<

4.2

>>> NEXT >>>

Sequence Related Specifications - G2-based Systems

Specification/Limitation	Value	Notes
Time resolution	100 ns	
Minimum event duration	100 ns	
Minimum event duration	100 ns	
Maximum event duration	143.165s	
Max events in sequence	1024	
Min time/point - Phase/Amplitude modulation	100 ns	
Max time/point - Phase/Amplitude modulation	143.165 s	
Max points/waveform - Phase/Amplitude modulation	200,000	The max size and min time/pt are inter-related and dependent on the FIFO size for the board
Max total waveform pts - Phase/Amplitude modulation	200,000	
Max points/table - Phase/Attenuation	200,000	
Max total points/table - Phase/Attenuation	200,000	
Max Frequency table entries	1024	
Min time/point - Gradient waveform	3.3 µs	
Max time/point - Gradient waveform	143.165 s	
Max points/waveform - Gradient waveform	200,000	
Max total waveform pts - Gradient waveform	200,000	
Min delay between consecutive Shape tables	0 ns	
Max number of independent loops	unlimited	
Maximum loop counter	16777216	
Max number of nested loops	32	
External trigger latency	370ns +/- 17 ns	rising edge to next event

Sequence Related Specifications - G1-based Systems

Specification/Limitation	Value	Notes
Time resolution	100 ns	
Minimum event duration	300 ns	
Minimum event duration	300 ns	
Maximum event duration	143.165s	
Max events in sequence	1024	
Min time/point - Phase/Amplitude modulation	33.333... ns	
Max time/point - Phase/Amplitude modulation	2184.5 µs	
Max points/waveform - Phase/Amplitude modulation	2048	
Max total waveform pts - Phase/Amplitude modulation	3072	
Max points/table - Phase/Attenuation	2048	
Max total points/table - Phase/Attenuation	3072	
Max Frequency table entries	1024	
Min time/point - Gradient waveform	3.3 µs	
Max time/point - Gradient waveform	550 ms	
Max points/waveform - Gradient waveform	2048	
Max total waveform pts - Gradient waveform	3072	
Min delay between consecutive Shape tables	300 ns	except DMOD
Max number of independent loops	512	
Maximum loop counter	16383	
Max number of nested loops	4	
External trigger latency	370ns +/- 17 ns	rising edge to next event

<<< PREV <<<

4.3

>>> NEXT >>>

ADC Related Specifications*

Specification	Value	Notes
Min Dwell Time	1 μsec (12-bit) 2 μsec (16-bit)	
Min Recycle Delay	4 μsec 2.2 μsec/pt (burst)	Also the minimum time between consecutive acquisitions in one scan.
Single-point acq - ADC	≥ 4 μs / point	
Input Range	+/- 2.5v	
Input Capacitance	17 pF between conversions 5 pF during conversions	

*Applies to *non-Digital Receiver* only, (i.e. Orion and DSPect).

<<< PREV <<<

4.4

>>> NEXT >>>

Analog Receiver Section Specifications

Specification	Value	Notes
Input impedance	50 Ω	nominal
IF Bandwidth	> 4 MHz at -3dB > 3 MHz at -1 dB	
RF Bandwidth	0.5-450 MHz	
gain	> 70dB	
gain control	50 dB 8-bit variable gain x1, x10 fixed gain	Non linear from 1-100, linear above 100 (to 10000) - NTNMR values.
Input saturation	-20 dBm 80mV p-p	
Max output level	+/- 1.5 V in 50 Ω	

<<< PREV <<<

4.5

>>> NEXT >>>

Digital Receiver Section Specifications

Specification	Value	Notes
Min Dwell Time	1 μsec or 0.300 μsec	
Min Recycle Delay	< 40 μsec + 1 dwell	
	1 μsec/pt (DRX fast mode)	At dwell < 1 μsec
Single-point acq	≥ 1 μs / point	
Input level	2 Vpp (+10 dBm) 1Vpp for direct detection	
Spurious signals	-112 dB @ n*BW / 64	
	-72 dB @ +/- 250 kHz	
Spectral widths (+/-)	1.667 MHz to 305 Hz (10.0 MHz / 2N, where 3 ≤ N ≤ 16384) N = P * Q * R, and P≤16, Q≤32, R≤32 are integers	
Passband width, - 3dB	+/- 1.75 MHz	
IF Frequency	DC to 12.5 MHz	
Spurs, SA related	-130 dB (sw= +/- 100 kHz)	

which is it then?

<<< PREV <<<

4.6

>>> NEXT >>>

Transmitter Section Specifications

Specification	Value	Notes
RF Bandwidth	5-450 MHz (+/- 3 dB) (usable to 500kHz (-10 dB))	
phase shift time	100 ns	may be further limited by amplifiers or filters
phase shift delay	300 ns	
ampl change time	100 ns	may be further limited by amplifiers or filters
output level	2 Vpp max	Att=0, Ampl=100.0
gating transient	3%	
leakage when blanked	-96 dB @ 5 MHz -40 dB @ 300 MHz	
leakage with zero output (unblanked)	-50 dB @ 5 MHz -30 dB @ 300 MHz	
gain	1 dB @ 20 MHz	100% amplitude, no atten

<<< PREV <<<

4.7

>>> NEXT >>>

Synthesizer Specifications - Apollo LF Systems

Specification	Value	Notes
range - direct, -3dB	200 Hz - 12.0 MHz	
range - doubled, -3dB	36 kHz - 24.8 MHz	
-6dB	20 kHz - 25.0 MHz	
output, minimum	+4 dBm = 1V p-p	> 2V pp achievable at midband for use as receiver LO
resolution, freq (step size)	.025 Hz	allows for exact frequency matching with possible doubler on other channels
resolution, phase	12 bit	
SFDR	>45 dB	frequency dependent

<<< PREV <<<

4.8

>>> NEXT >>>

Synthesizer Specifications - Apollo HF Systems

Specification	Value	Notes
range, -3dB	5-450 MHz	
output, minimum	0 dBm	
resolution (step size)	.025 Hz	allows for exact frequency matching with possible doubler on other channels
resolution, phase	12 bit	
SFDR	40 dB	
900Gen output, dBm	+1 to -3 dBm	905-912.5 MHz Allow for cable loss when measuring: 5' RG174 ~ -3dB Measured with 2.6v pp input from O1O2.
900Gen SFDR (+/- 20 MHz)	40 dB	

<<< PREV <<<

4.9

>>> NEXT >>>

Miscellaneous System Specifications

Specification	Value	Notes
External Trigger Input level	TTL	
External Trigger Latency, rising edge to start of next event	370ns +/- 17 ns	
Max data set size	ca 256x256x64 or 1K x 1K x 4 (32 Mbyte standard SA Memory)	ca 512 x 256 x 256 with 256 Mbyte SA memory
Data upload time from Signal Averager	15-20 µs/complex point typ.	Host computer will impact data transfer speed.
Clock Drift, ppm/day	+/- 5.0 e-4	
Clock Temp stability, ppm	+/- .01	
TTL output lines		25 ohm line drivers

<<< PREV <<<

4.10

>>> NEXT >>>

Environmental Specifications

Specification	Value	Notes
Mains voltage (specified when ordered)	120 VAC, 60 Hz single phase, or 240 VAC, 50 Hz single phase	
Maximum voltage variation	+/- 10%.	
Current requirements:		
Apollo-LF (1 channel)	120 V = 2 A 240 V = 1 A	
Apollo-HF (1 or channels)	120 V = 4 A 240 V = 2 A	
Recommended operating temperature	+15 to +25 C (59 to 77 F)	
Temperature stability	+/- 1 C	
Heat generated (console only)	LF = 180 W HF (2ch) =450 W	

5 - Accessing the User Definable Pulse Programmer Lines

<<< PREV <<<

5.1

>>> NEXT >>>

Apollo.ini File

Each module in the system is defined in the Apollo.ini file. Depending on the module type, there may be several different parameters associated with a module section in the Apollo.ini file. Although the Apollo.ini is somewhat complex and generally should not be edited by the user, there is some information contained in this file that is required for adding user definable pulse programmer lines. The config.con file (see below) requires a "Module Prefix" for each pulse sequence ~~line~~. The modules prefix that has been assigned to each module in the system ca be found in the [ModuleNames] section of the Apollo.ini file.

1 Error

<<< PREV <<<

5.2

>>> NEXT >>>

Config.con File

The 'config.con' file is a text file used by NTNMR to format the Pulse Sequence Window. This file must be located in the 'system' folder in the NTNMR folder. This file contains the interface information necessary to assign output lines from the Tecmag pulse programmer for spectrometer control functions. In general this file should not be edited by the user. However, there may be cases, such as the construction of a home built spectrometer based on a Tecmag Pulse Programmer, where it may be necessary to edit the 'config.con' file.

See the next page for an example config.con file.

Example Config.con File

```
;file name = config.con
;
;Definition of LP lines
;
;    -- File Format --
;
; 1 = <Name>
; 2 = <Module Prefix>
; 3 = <Start Address>
; 4 = <Number of Bits>
; 5 = <Minimum delay>
; 6 = <Maximum Delay>
; 7 = <Delay Unit>
; 8 = <Icon type>
; 9 = <Visible?>
; 10 = < Private Data>
; Any text after ";" on each line is ignored.
; Fields are delimited by either spaces or tabs.

; <1>        <2> <3><4><5><6><7> <8> <9> <10>
OBS_RF       ES1 20  1  1  0  0  TX   1   0; OBS_RF gate (p1_SPF1)
OBS_PH       ES1 19  2  1  0  0  PH   1   0; obs. Phase= LP19,LP18
Acq          IO3 1   2  0  0  0  ACQ  1   0;
Acq_phase    IO3 3   2  0  0  0  PH   1   0 ;
DEC_RF       ES1 21  1  1  0  0  TX   1   0 ; Dec_RF gate
DEC_PH       ES1 15  2  1  0  0  PH   1   0 ; decouple phase
BB_MOD       ES1 22  1  1  0  0  TX   1   0 ;
RCP4         EM1  3  1  1  0  0  TX   1   0
;ATN2        EM1  8  2  1  0  0  TX   1   0 ; I3_ATN3
;SPOIL       EM1  1  1  1  0  0  TX   1   0
Rec_Gate     ES1 12  1  1  0  0  TX   1   0;
Gx_Shape     GM1 31 16  0  0  0  SH   1   0;
Gx_Amp       GM1 15 16  0  0  0  GR   1   0;
Gy_Shape     GM2 31 16  0  0  0  SH   1   0;
Gy_Amp       GM2 15 16  0  0  0  GR   1   0;
Gz_Shape     GM3 31 16  0  0  0  SH   1   0;
Gz_Amp       GM3 15 16  0  0  0  GR   1   0;
SH_RF_A      GM5 31 16  0  0  0  SH   1   0;
RFA_Amp      GM5 15 16  0  0  0  GR   1   0;
SH_RF_B      GM6 31 16  0  0  0  SH   1   0;
RFB_Amp      GM6 15 16  0  0  0  GR   1   0;
```

The semi-colon character is used to comment a line. Everything after the ";" on any line is ignored. Individual lines can be commented out. (As in the "ATN2" and "SPOIL" lines in the example above.

NTNMR creates the sequence window with the lines in the order that they appear in the config.con file. This is only the default order and lines can be repositioned in the sequence window.

The fields in the file are labeled as <1>, <2>, … <8> for informational purposes only. The fields are described below in reference to the labels in the above example.

<1> Line name. This is the exact text that will appear in the pulse sequence window when NTNMR is run.

<2> Module prefix. This tells the software what type of module this line in the pulse sequence is controlling. There are several module types: EM (*event module*), ES (*event module slave*), GM (*gradient module*), etc. and each module has a number assigned to it, EM1, IO3, GM4, etc. Consult the hardware documentation for more information about the various modules types and their function.

<3> Start address. This is an integer number that defines the starting address for the pulse programmer lines to use for the current line in the pulse sequence.

<4> Number of Bits. This is an integer number that defines the number of pulse programmer lines to use starting from the Start Address. For example, in the config.con example shown above shows the line Gx_Shape with a starting address of "31" and a number of bits value of 16. This means that there are 16 bits assigned to the control of the GM1 module. The 16 bits are 31, 30, 29, …16.

<5>, <6>, <7> These fields are currently unused.

<8> Icon Library. This field contains a two letter code that assigns the type of icon to display in the NTNMR graphical sequence display. The various icon type and their function are discussed in the next section.

<9> Visible. This flag determines whether or not the defined line will be visible in the graphical pulse sequence editor in NTNMR. 1 = visible, 0 = invisible.

<10> Private Data. This field is used for various line-specific data assignments. For example, this field may be used to indicate the resolution of the transmitter phase shifter, the resolution of the transmitter attenuator, etc.

In the Pulse Sequence Window, the **Delay**, **ACQ**, **ACQ Phase**, and **Loop** row labels are added to the timing diagram automatically by the software, they are not part of the 'config.con' file.

<<< PREV <<<

5.3

>>> NEXT >>>

Icon Libraries

Each icon library used in the config.con file is outlined in the table below. For specific examples of the icons that appear in the pulse sequence window see the section above entitled "Defining and Editing Pulse Sequences."

LIBRARY	ICONS	INFORMATION
TX	gate on, gate off	used for controlling rf transmitter channels, attenuators, external devices
PH	fixed phases (+x, +y, -x, -y) and 1 - 4D tables	used for controlling rf phase shifts and for signal averager phase cycling.
ACQ	Acquisition	indicates the start of data acquisition by the signal averager
SH	Shape	Allows the creation of waveform (i.e. shape) tables for gradients or for RF pulse shaping
GR	Gradient	Gradient amplitude control
PS	Shape	Phase modulation
RM	Shape	Transmitter amplitude modulation
AT	Shape	Transmitter attenuator
ET	gate on, gate off	activate the external trigger
LD	Shape, gate	Used to assign a group of lines to be used for clocking out a table of values one value at a time.

6 - Frequency Control in Apollo/Orion/DSPect Systems

<<< PREV <<<
6.1
>>> NEXT >>>

Introduction

Frequency control can be accomplished manually (e.g. using knobs if a PTS synthesizer is present), or under software control from NTNMR by one of the following means:

- Tecmag PTS synthesizer interface.
- Tecmag Dual DDS board.
- Tecmag HF synthesizer.
- Combinations of the above (Compound Synthesizers).
- Bruker SDC using Tecmag DDS for offsets.

Each method will be discussed in a separate section. Additionally, the use of one or more synthesizer systems (of any of the above types) with a receiver as

- First local oscillator (LO) or
- Detector oscillator (IF).

will be discussed. Following the discussion, a list of all available .INI file settings is given.

<<< PREV <<<
6.2
>>> NEXT >>>

Setting Frequencies in NTNMR

NTNMR permits four spectrometer channels to be defined (**F1....F4**). The channel associated with the receiver is specified by the dashboard **Obs Channel** parameter. The connection between the hardware and the software is made by the **ChannelNum** keywords in the .INI file. The receiver channel does not have a number. It is important to have only one device for each ChannelNum, and to have channels numbered contiguously starting at 1. Failure to observe these requirements will cause unpredictable results or software crashes.

Frequencies can be set from the dashboard. The settings for any individual channel will be sent to the hardware when <ENTER> is pressed after editing the frequency value, or when the cursor is moved to another dashboard field. Dashboard frequencies for all defined channels (F1....F4 and any associated

? where

ah: can be customized

receivers) will be sent to the hardware when a Zero/Go command is issued, before data acquisition. The dashboard permits use of "base" and "offset" frequency values, but this is strictly for user convenience and has no meaning whatever to the hardware.

Changing frequency during a sequence ("hopping") is also permitted under certain circumstances. As of software release 805xx, this is only possible on channels incorporating a DDDS board. The board is listed in **config.con** (type "O1" or "O2") to create a line in the sequence editor (separate lines for each port can also be defined, if appropriate). This entry refers to the *simple* synthesizer module, even if it is part of a *compound* synthesizer. To change the frequency, a frequency shift (in Hz, relative to the dashboard value) is entered on the relevant line at the appropriate event in the sequence. A table (1D...nD) of frequencies can also be used. The frequency remains at the new setting until another value is encountered (it does not return to zero shift at the end of the sequence line!). Care must be used to assure phase coherence when using this feature.

<<< PREV <<<

6.3

>>> NEXT >>>

The INI File Entries

In order to invoke the correct software routines, and perform the correct calculations, NTNMR must know what hardware is present, and how it is configured (e.g. frequency multiplication or shifting, assignment to transmit or receive, etc.). This is the function of the file **..\system\apollo.ini.** In case phase or frequency modulation is required during a pulse sequence, the hardware must also be described to the pulse sequence generator. This is the function of the file **..\system\config.con**.

In NTNMR, a synthesizer can be either a simple or a compound type. A **simple synthesizer** will have a *single* block of entries in the .INI file, and consist of a single controllable device (although there may be additional fixed oscillators, multipliers and so on to achieve the final frequency). The PTS, BrukerPTS, O1O2, and Synth5 board are examples. A **compound synthesizer** has two or more controllable elements which together determine the overall frequency (also possibly incorporating fixed oscillators, etc.). For each compound synthesizer, a total of *three* blocks of entries are required in the .INI file: one for the compound device, and one for each of the two (sub-)synthesizers which compose it. The sub-synthesizers may be simple devices, or compound devices in their own right. (But fixed frequency oscillators, multipliers, and other non-controllable devices are not sub-synthesizers.) Examples of compound synthesizers are the Tecmag HF synthesizer (Synth5 + DDDS) and Bruker synthesizers (BrukerPTS + DDDS).

The [ModuleNames] section of the file is used to specify all of the components to be included in the NMR instrument. The left side of each entry in this section (e.g. EF1) is a "prefix" used for entry in config.con, if required, or for reference by other modules in the system. The right side of the entry indicates the section of the .INI file which gives the details. Each component appearing on the right side of an entry in [ModuleNames] must have its own section. Thus, for compound synthesizers, at least three entries would be required. The following fragments from a .INI file illustrate both types - SY1 is a simple synthesizer, and SY2 is a compound synthesizer:

```
[ModuleNames]
; Modules for a simple synthesizer and a compound
synthesizer
SY1 = Synth1
SY2 = Synth2
OF2 = Offset2
MA2 = Main2

[Synth1]
ModuleType = 0102
ChannelNum = 1

[Synth2]
ModuleType = BrukerSynth
ChannelNum = 2

[Offset2]
ModuleType = 0102

[Main2]
ModuleType = BrukerPTS
```

<<< PREV <<<

6.4

>>> NEXT >>>

Troubleshooting with the Log File

NTNMR records certain actions for troubleshooting and other purposes in the Apollo Log File. The level of detail for this log is controlled by the .INI file LogLevel parameter (value =1....5) for each device. Lower values will suppress the messages generated at higher levels. For frequency-setting devices, the following events are logged at each level (a general outline, not always strictly enforced!):

Level	Event	Software Module
1	Instrument creation, reset, destruction	ApolloInstrument
		various
	Fatal errors	
2	Initialization	each module, at Reset
	Run Sequence	SysInterface

3	Major functions (SetFreq...)	O1O2, PTSSyn, Synth5, etc.
4	Reading INI parameters Details of major functions	various
5	Sequence compilation	various

Higher LogLevel values may cause a slight slowdown of some functions, such as "Zero and Go" (but no effect on running sequences). The Log file is usually found in **C:\Temp\Alog.log**, and can be inspected with NotePad. A new log is created each time NTNMR is started, and the old log file is not saved.

<<< PREV <<<

6.5

>>> NEXT >>>

Complex Mixing Schemes

NTNMR allows the user to configure control devices for complex mixing schemes, including frequency multipliers, combination of more than one frequency-control device by mixing, and use of fixed offset oscillators ("bumpers"). The actual output frequency is entered in the dashboard, and each device is then set according to the required formula to achieve the desired output.

Two software mechanisms for accomplishing this are incorporated: 1) implementation of simple multipliers and offsets in **any** simple synthesizer device (e.g. PTS, or DDDS boards, requiring only the correct parameters in the .INI file); and 2) definition of more complex systems in software modules **(compound synthesizers)** which control 2 or more devices and assign different parts of the frequency task to each (e.g. Bruker or Tecmag synthesizers).

The basic mixing / multiplying scheme (which can be applied to any synthesizer device) uses four user-specified parameters:

```
[MySynthesizer]
Mult1 = 2.0                 (Default = 1.0)
Mult2 = 1.0                 (Default = 1.0)
OffsetMHz1 = 50.0           (Default = 0.0)
OffsetMHz2 = 0.0            (Default = 0.0)
```

The output frequency will be controlled according to the above parameters and the following equation:

F1(final output frequency) = { [(device setting)*Mult1 + OffsetMHz1]*Mult2 + OffsetMHz2 }

Thus, using the example parameters, a spectrometer frequency of 61.35 MHz is achieved by setting the device (e.g. PTS or DDDS) to 5.675 MHz. The hardware output of 5.675 MHz is doubled (Mult1) and mixed with a fixed oscillator frequency of 50 MHz (OffsetMHz1) to produce the final output of 61.35, as specified in the dashboard.

For "hops" during a sequence, the hop direction may be reversed by the mixing scheme (if the lower sideband is used, the frequency will be inverted). In this case, the HopDirection parameter may be used to correct the sign of the hop.

The compound frequency case will be discussed under the Tecmag Synthesizer heading.

<<< PREV <<<

6.6

>>> NEXT >>>

PTS Control

Introduction

PTS synthesizers (product of Programmed Test Sources, Inc.) are principally used with the **Orion** systems (PTSs used in Bruker systems have been modified, and will be treated separately). One or more PTS units are connected to an Event Module (EM) board (and DB-25 mezzanine adaptor) via PTS Adaptor modules and a cable. The adaptor attaches to the Centronics connector on the rear of the PTS and contains the "address" information. Upto 4 PTS's can be connected to each EM/DB-25. Each adaptor is configured to respond to one of four different control lines in the cable via DIP switches (jumpers on older models).

The PTS operates as a *simple* synthesizer in NTNMR. Each PTS requires a separate block in the .INI file, although up to 4 PTSs on a single EM will reference the same EM block. For a system in which the PTS is the primary frequency control, the PTS adaptor address will be the same as the spectrometer channel number (F1 has address 1, etc.). In other cases, however, this may not be true, particularly when one or more PTS units is a part of a *compound* synthesizer.

Sample .ini file entries for PTS

In this example, 2 PTS units are connected. For demonstration purposes, these are shown as configured for F2 and F3 (F1 is some other synthesizer). Note that the addresses *do not match* the channel numbers in this example.

```
[ModuleNames]
; -- Synthesizer objects --
TX1 = TXChannel1
TX2 = TXChannel2
TX3 = TXChannel3
; Single EM controls upto 4 PTS's
EMP = EM_PTS

[TXChannel1]
; some other device
ChannelNum  = 1          ; This is F1

[TXChannel2]
ModuleType  = PTS
Control1    = EMP        ; Controlled by EM_PTS
SeqResource = PTS.seq
MaxFreqMHz  = 160        ; Max freq for this PTS in MHz
ChannelNum  = 2          ; This is F2
```

```
PTSAddress   = 1          ; First PTS on this cable
LogLevel     = 5

[TXChannel3]
ModuleType   = PTS
Control1     = EMP        ; Controlled by EM_PTS
SeqResource  = PTS.seq
MaxFreqMHz   = 250        ; Max freq for this PTS in MHz
ChannelNum   = 3          ; This is F3
PTSAddress   = 2          ; 2nd PTS on this cable
LogLevel     = 5

[EM_PTS]
ModuleType    = SlowIO
BaseAddr      = 0x0F70
EXEFile       = EM-Sio.exe
TicksPer100n  = 3
LogLevel      = 5
```

Corresponding config.con file entries:

PTS synthesizers cannot be controlled in real time.

PTS Adaptor interface details

The control uses parallel data with latches, as follows: 8 data, 3 address, and 1 select line are used to write to the PTS. Each 8 bits of data represents 2 BCD digits depending on the address

```
ADDRESS      BCDH, BCDL
--------------------
0            100M  10M
1            1M    100K
2            10K   1K
3            100   10
4            1     0.1
```

The select line is used as a strobe (active high). An additional line (RemoteEnable#) is used to enable the remotely programmed frequency. It may be left low by default.

```
----------------------------------------------
 PTS Adaptor - LP assignments on EM board
----------------------------------------------
PTS_BCDL    LP3..0      ; BCD data low
PTS_BCDH    LP7..4      ; BCD data high
PTS_Addr    LP10..8     ; PTS Frequency register select
```

```
PTS1_Sel     LP11        ; Select line PTS1
PTS2_Sel     LP12        ; Select line PTS2
PTS3_Sel     LP13        ; Select line PTS3
PTS4_Sel     LP14        ; Select line PTS4

PTS1_RE      LP15        ; Remote Enable PTS1
PTS2_RE      LP16        ; Remote Enable PTS2
PTS3_RE      LP17        ; Remote Enable PTS3
PTS4_RE      LP18        ; Remote Enable PTS4
```

<<< PREV <<<

6.7

>>> NEXT >>>

The Dual DDS (DDDS or "O1O2")

Introduction

The **Apollo** series depends principally on the **Dual DDS** board for phase and frequency control. The DSPect system also uses this board.

This board contains two Analog Devices AD9830 Direct Digital Synthesizer chips (DDS, sometimes also called Numerically Controlled Oscillator or NCO). Each chip permits frequency setting in very small increments according to a binary 32 bit value. When using the Tecmag-standard 50 MHz clock, the frequencies available are those which satisfy the following relationship (N is a positive integer less than 2^{31}):

$$F = N * (50\ \text{MHz}) / 2^{32}$$

The resolution is thus about 0.01164 Hz. Direct numerical phase modulation with 12-bit resolution is available, as is an absolute phase reset. The chips permit additional features which are not currently used, such as rapid switching between previously stored frequencies or phases. The output data from the NCO are converted to analog form by a 10-bit DDS, and after amplification and low pass filtering, two independent outputs are available from the board.

Several versions of the DDDS board exist, allowing different control interfaces (8-bit, 16-bit, depending on the Altera program and the presence of a data latch chip); and incorporating various frequency multipliers, mixers, switches and filters. In some cases, only one output port is provided (the components for the second port are omitted). By programming both oscillators with the same frequency and a 90 degree phase shift, I/Q modulation (single-sideband modulation) is facilitated. This feature is used in the Apollo HF system.

DDDS boards are used as *simple* synthesizers in Apollo LF systems (either direct or with on-board multipliers), and Apollo MF systems (with on-board 50 MHz "bumper"). They also form one element of *compound* synthesizers in Apollo HF systems (with a 900Gen and Synth5 board) and DSPect systems (with a Bruker-modified PTS, fixed-frequency bumper and multipliers in the original Bruker console).

Sample .ini file entries

In this example, the O1 port of an O1O2 board is used to control frequency and phase for NTNMR **F1** (ChannelNum = 1). The **frequency** is controlled in real time by event module EF1, and **phase** by event module EP1. (Notice that the phase control is not mentioned directly at all in connection with this device - but it is specified in config.con). In this example, the raw frequency is to be doubled and added to a 50 MHz bumper.

Since the second port is used for a second synthesizer, but the same EM controls both, it is not generally possible to perform real-time switching on both ports in the same sequence. To permit switching on both ports, 4 lines are required in config.con, as shown. This is not a hardware limitation, but a compiler limitation.

```
[ModuleNames]
; Modules for a simple synthesizer
SY1 = Synth1
SY2 = Synth2
EF1 = EM_Freq_1
EP1 = EM_Phase_1
[EM_Master]
; -- Specify LP lines to TRIGGER all other modules --
;23=ScanInc to SysIntf board
SY1          = 17
SY2          = 17

[Synth1]
ModuleType = 0102
ChannelNum = 1
SeqResource = 012_8Bit_AD.seq
Control1 = EF1
NumPorts = 1  ; Only one DDS is used
PortNum  = 1  ; 1 or 2 depending on which DDS is involved
              ; (01 or 02)
LogLevel = 4  ;
Mult1 = 2.0   ; DDS frequency to be doubled by hardware
Mult2 = 1.0   ; (optional, default = 1.0)
OffsetMHz1 = 50.0  ; Offset frequency added in hardware
("bumper")
OffsetMHz2 = 0.0   ; (optional, default = 0.0)
HopDirection = 1.0   ; (optional, default = 1.0)
```

Corresponding config.con file entries:

F1	SY1	23	12	0	0	0	01	1	0;
F1_d	SY1	11	12	0	0	0	02	0	0;
RX_d	SY2	23	12	0	0	0	01	0	0;
RX	SY2	11	12	0	0	0	02	1	0;

Note that the second and third lines are “dummy” lines and are invisible to the user

<<< PREV <<<

6.8

>>> NEXT >>>

The Tecmag HF Synthesizer

Introduction

Apollo HF systems use Tecmag synthesizers. These consist of 3 boards: O1O2 Dual DDS, 900Gen, and Synth5 PLL, together with the associated event modules. Both channels of the Dual DDS are used in an I / Q modulation format: i.e. both DDSs are set to the same frequency, with the phase 90 degrees apart. The output of the DDSs, with any associated phase modulation or shifting, is mixed on the 900Gen board with a 900.0 MHz signal (generated by multiplying the 50 MHz clock by 18). The upper sideband of the resulting signal is filtered to produce a high-resolution modulated signal in the 905-912.5 MHz range. Separately, a PLL (phase-locked loop) synthesizer on the Synth5 board generates a CW signal in the 900-1400 MHz range, with 5 MHz resolution. The 905 MHz signal is subtracted in a mixer on the Synth5 to produce the final output at the desired frequency. A programmable 15 dB attenuator allows the output level to be equalized at all frequencies within a dB or two.

Both the DDS module and the PLL module must be controlled. So this is a *compound* synthesizer, and 3 modules are defined in the .INI file. Additionally, phase and frequency control for the DDS board is required, each provided by an event module (EM). The Synth5 has its own bus address and is controlled directly, rather than through an EM.

Sample .ini file entries

In this example, the synthesizer output is used directly, so the multiplication values and offsets are defaulted.

```
[ModuleNames]
; Modules for a simple synthesizer
TS1 = TecSyn1
S51 = Synth5_1
IQ1 = DDDS_1
EF1 = EM_Freq_1
EP1 = EM_Phase_1


[TecSyn1]
; Top level - compound synthesizer
ModuleType = TecSyn
Control1    = TS1     ; Coarse steps (PLL board)
Control2    = IQ1     ; Fine steps (DDS board)
OffsetMax      = 10.0
OffsetMin      = 6.0
IdlerSideband = 1.0
```

```
IdlerFreq     = 900.0
ChannelNum = 1       ; Associate with F1
AttenuationFile = SY1.AFF
LogLevel   = 5


[Synth5_1]
ModuleType     = Synth5
BaseAddr       = 0x0100 ; Base address of Syn5 board
ReferenceFreq = 50.0
RefDivisor     = 16
LogLevel       = 5


[DDDS_1]
ModuleType   = O1O2
SeqResource = O12_IQMod.seq ;
Control1     = EF1            ; Controlled by EM_Freq_1
NumPorts     = 2
PortNum      = 1
LogLevel     = 5
```

Corresponding config.con file entries:

```
F1_I      IQ1    11   12   0    0    0    O1    1
0;
F1_Q      IQ1    23   12   0    0    0    O2    1
0;
```

For hopping, the hop frequenc(ies) (offsets from dashboard, in Hz) must be entered on BOTH lines of the sequence.

<<< PREV <<<

6.9

>>> NEXT >>>

Other Compound Synthesizers

Introduction

Other compound synthesizers are possible. For example, to reach frequencies beyond the range of the Tecmag synthesizer, a PTS might be used in a transverter (the same frequency being used to upconvert the transmitter and downconvert the receiver). Since both the PTS and the Tecmag synthesizer are controllable, a large number of possibilities exist that will result in the desired output frequency. One possible scheme is to use the PTS for "coarse" steps (a few hundred MHz), and the Tecmag synthesizer for the fine steps. A special software module (CSynthSpecial) has been written to accomplish this.

The CSynthSpecial software module uses the CoarseStartMHz and CoarseStepMHz values to assign a frequency to the coarse synthesizer (PTS in the example). For example, if the desired frequency is 751.3 MHz, and the mixing scheme uses the upper sideband (MixerSideband = +1), while the minimum fine frequency is 100 MHz (FineMHzMin=100), the PTS will be set to the nearest coarse step below (751.3-100 = 651.3). If the coarse steps are 200 MHz starting at 100 MHz, the result will be (2 x 200 + 100 = 500 MHz, < 651.3 MHz). The fine synthesizer is then set to the difference (751.3 - 500 = 251.3 MHz).

Transverters

A trick must be used for the transverter case (where the same oscillator is used for both transmit and receiver conversions). The receiver LO synthesizer is defined in exactly the same way as the transmitter synthesizer, except for the FineMHzMin and FineMHzMax values. These are both altered by exactly the receiver IF offset (actually the second IF in the case of a transverter!). For example if the IF is 12.5 MHz, and the receiver uses high-side injection, the receiver synthesizer will use FineMHzMin/FineMHzMax values of 112.5/412.5 MHz instead of 100/400 MHz. Thus, when the transmitter is set to XXX MHz, using a coarse value of YYY MHz, the receiver will be set 12.5 MHz higher, but the requested value for the coarse synthesizer will still be YYY MHz. The second request, by the receiver, to set the coarse synthesizer will thus have no net effect on the transmitter frequency. See the section on the receiver, below, for more information.

Sample .ini file entries

In this example, the synthesizer output is used directly for F1, so the multiplication values and offsets are defaulted.

```
[ModuleNames]
; Extra EM - for transverter PTS, etc.
EX1 = EM_Ext_1
; -- Synthesizer Modules for TX Channel 1 --
TX1 = TXChannel1
PT1 = PTSSyn1
TS1 = TecSyn1
S51 = Synth5_1
IQ1 = DDDS_1
EF1 = EM_Freq_1
EP1 = EM_Phase_1


[EM_Ext_1]
ModuleType    = SlowIO
BaseAddr      = 0x0F70
EXEFile       = EM-Sio.exe
TicksPer100n = 3
Verify        = 1
LogLevel      = 2


[TXChannel1]
;Top level compound synth - combines TecSyn and PTS
ModuleType       = SpecialSynth
Control1         = PT1        (Coarse synthesizer, PTS)
Control2         = TS1        (Fine synthesizer, Tecmag Synth)
CoarseStepMHz = 200.0         (MHz, default = 750 MHz)
CoarseStartMHz = 100.0        (MHz, default = 0 MHz)
MixerSideband = 1             (+/-1, upper/lower sideband, default = 1)
FineMHzMax   = 400.0          (Fine offset, MHz, default = 400 MHz)
FineMHzMin   = 100.0          (Fine offset, MHz, default = 10 MHz)
ChannelNum   = 1              (Associate with dashboard F1)
LogLevel     = 2

[PTSSyn1]
; PTS - provides coarse steps
ModuleType    = PTS
Control1      = EX1        ; Controlled by EM_Ext_1
SeqResource   = PTS.seq
PTSAddress    = 1          ; PTS Adaptor address switch
setting
MaxFreqMHz    = 1000       ; Max freq for the PTS in Hz
LogLevel      = 5

[TecSyn1]
; Second level compound synth - provides fine steps
ModuleType       = TecSyn
Control1         = S51      ; Synth5
Control2         = IQ1      ; Dual DDS
AttenuationFile = TX1.AFF
OffsetMax        = 12.5
OffsetMin        = 6.0
```

```
IdlerSideband   = 1.0
IdlerFreq       = 900.0
LogLevel        = 2

[Synth5_1]
ModuleType    = Synth5
BaseAddr      = 0x0100 ; Base address of Syn5 board
ReferenceFreq = 50.0
RefDivisor    = 10     ; 5 MHz steps
LogLevel      = 5

[DDDS_1]
ModuleType  = O1O2
SeqResource = O12_IQMod.seq ;
Control1    = EF1          ; Controlled by EM_Freq_1
NumPorts    = 2
PortNum     = 1
LogLevel    = 5
```

Corresponding config.con file entries:

The Tecmag synthesizer (actually the dual DDS!) is used for real-time frequency hops:

```
F1_I     IQ1     11    12    0     0     0     O1     1
0;
F1_Q     IQ1     23    12    0     0     0     O2     1
0;
```

For hopping, the hop frequenc(ies) (offsets from dashboard, in Hz) must be entered on BOTH lines of the sequence.

<<< PREV <<<

6.10

>>> NEXT >>>

DSPect (Bruker) Systems

Introduction

The **DSPect** series uses DDDS ("O1O2") boards as "offset synthesizers" to replace the original equipment in Bruker AC and AM systems. Typically, one port is used for the observe frequency, together with a Bruker-modified PTS unit as a compound synthesizer. The second port is used for the decoupler, with an external bumper and multiplier, as a simple synthesizer, or with a second PTS as a compound synthesizer. The details of the multiplication schemes depend on the exact spectrometer configuration.

Sample .ini file entries for DSPect (Bruker AC300 with decoupler)

In this example, the O1 port of a DDDS board is used to control frequency and phase for NTNMR **F1** (ChannelNum = 1). The base frequency is controlled by a "Bruker PTS" connected to the SDC. The frequency (offset frequency) is controlled in real time by Event Module EF1, and phase by Event Module EP1. The board has an Altera chip programmed for 8-bit control interface (the data buffer/latch chip must also be installed in this case), and the appropriate sequence resource is specified to correspond to this.

The O2 port of the DDDS board is used to control frequency and phase for NTNMR **F2** (ChannelNum = 2). This port's output is tripled and added to a bumper to obtain the proton decoupler frequency (see **complex mixing schemes**, elsewhere). Note also that O22 and O12 actually refer to the same physical board!

The DDDS must be triggered for frequency changes during a sequence. The trigger is specified in the [EM_Master] section. The number refers to the LP line on the master.

```
[ModuleNames]
; Modules for AC300 frequency control
TX1 = TXChannel1
TX2 = TXChannel2
PT1 = PTS_1
O12 = DDDS_1
O22 = DDDS_2
EF1 = EM_Freq_1
EP1 = EM_Phase_1
SDC = Bruker_SDC

[EM_Master]
; -- Specify control lines to all other modules --
O12          = 21
```

```
O22           = 21

[TXChannel1]
ModuleType = BrukerSynth
Control1   = PT1
Control2   = O12
ChannelNum = 1
LogLevel = 4  ;

[TXChannel2]
ModuleType = BrukerSynth
Control2   = O22
ChannelNum = 1
LogLevel = 4  ;


[PTS_1]
ModuleType = BrukerPTS
Control1   = SDC
LogLevel   = 5

[DDDS_1]
ModuleType  = O102
SeqResource = O12_AD_8Bit.seq
Control1    = EF1       ; Controlled by Module EF1
NumPorts    = 1
PortNum     = 1
HopDirection = -1
LogLevel    = 5

[DDDS_2]
ModuleType  = O102
SeqResource = O12_AD_8Bit.seq
Control1    = EF1       ; Controlled by Module EF1
NumPorts    = 1
PortNum     = 2
Mult1       = 3.0
OffsetMHz1  = 290.0
LogLevel    = 5
```

Corresponding config.con file entries:

```
O1     O12   23  12  0  0  0  O1  1  0;
O1_d   O12   11  12  0  0  0  O2  0  0;
O2_d   TX2   23  12  0  0  0  O1  0  0;
O2     TX2   11  12  0  0  0  O2  1  0;
```

The two lines with "_d" are invisible dummy lines required by the sequence compiler (as of 80601).

<<< PREV <<<

6.11

>>> NEXT >>>

The Receiver

Introduction

This section applies principally to **Apollo** series units with a Tecmag RF/IF receiver module. However, frequency control is available for other receivers as well.

If a receiver module is defined in the .INI file, NTNMR will call the SetFrequency function for this module immediately after setting F1 (or the channel specified by the "Observe Ch" dashboard field). The actual frequency of F1 (in other words, the physically-achievable frequency closest to the requested F1 value) will be passed to the receiver. In turn, if an IF synthesizer (detector frequency) is defined (as for Tecmag systems with a digital receiver), the IF synthesizer will be set. Finally, if a receiver local oscillator (LO) is declared, it will then be set as well, using the F1 frequency (from NTNMR - dashboard) **and** the IF frequency (from the .INI file) with the selected sideband (see below). If the resulting physically-achievable frequency is not an exact match for the F1 frequency, an error will be logged (but not displayed), as this will cause problems with phase coherence and signal averaging.

Sideband selection is made using two parameters: RFSideband (= +/-1) and SidebandChangeMHz (= floating point frequency in MHz). The normal situation is to use the lower sideband (LSB), with the LO frequency **above** the observe frequency (e.g. 311.35 MHz LO for 300.1 MHz observe frequency). However, there may be an upper limit to the synthesizer frequency which is less than required for the maximum observe frequency (e.g. 499.95 MHz observe frequency with a PTS-500). In this case, we must switch to the upper sideband (USB). For example, we could specify SidebandChangeMHz = 400.0 to use USB for proton and fluorine, and LSB for all other nuclei on an 11.7T magnet.

There could be a case in which one wishes to invert the sidebands for some reason, such as avoiding interference from the image frequency (the Tecmag IF receiver has essentially no image rejection). For such a case, RFSideband = -1 (use LSB, above default change frequency = 0) should be used, otherwise, RFSideband = +1 (use USB, above default change frequency = 13.75, LSB below). The default change frequencies can be overridden by specifying an actual value as in the previous example.

The correct sideband will be used by NTNMR for purposes of setting the acquisition phase (i.e. the sense of rotation will be reversed for the lower sideband) and (eventually) for automatically setting the frequency axis.

INI File Parameters for Receiver Frequency Control

Optional parameters:

`[Receiver]`	
`Control3 = IF1`	(module prefix for IF synthesizer)
`Control4 = LO1`	(module prefix for LO synthesizer)
`RFSideband = 1`	(+/-1, +1 = upper sideband, LO < RF. Default = -1)
`SidebandChangeMHz = 120`	(frequency where receiver sideband changes.
Default =	
	0/13.75, depending on RFSideband)
`IFFrequency = 11250000`	(Frequency in Hz. Default = 0)

<<< PREV <<<

6.12

>>> NEXT >>>

Summary: INI File Parameters for Synthesizers

General

Required parameters common to **all synthesizer** systems:

`ModuleType = O1O2` (Hardware type: O1O2, PTS, BrukerPTS, Synth5 **(simple)** or TecSyn, BrukerSynth, SpecialSynth **(compound)**)

`ChannelNum = 1` (Associates hardware with software F1...F4 if required)

`Control1 = EF1` (Module prefix of Event Module for frequency control. Must be declared in [ModuleNames] section of .INI file. Additional control devices may be required)

`LogLevel = 4` (1-5, detail level of information recorded in Alog.log)

Optional parameters common to **all synthesizer** systems:

`Mult1 = 2.0` (For frequency conversion systems. Default = 1.0)

`Mult2 = 1.0` (For frequency conversion systems. Default = 1.0)

`OffsetMHz1 = 50.0` (For frequency conversion systems. Default = 0.0)

`OffsetMHz2 = 0.0` (For frequency conversion systems. Default = 0.0)

`HopDirection = 1.0` (+/- 1. Accomodates sideband inversion in mixing schemes for frequency hops in a sequence. Default = 1.0)

Parameters for simple synthesizer systems

Required parameters for **ModuleType=O1O2**:

`SeqResource = O12_8Bit_AD.seq` (Depends on the Altera chip and application)

Optional parameters for **ModuleType=O1O2**:

`DDSType = AD_16bit` (only useful value right now)

`DDSClock = 50000000` (Hz, default = 50000000)

`NumPorts = 1` (1-2, independent or I/Q mode, default = 2)

`PortNum = 1` (1-2, output jack ID for independent mode, default = 1)

`HopDirection= 1` (+/-1, change sign of hops in sequence for Bruker systems, default = +1)

Required parameters for **ModuleType=PTS**:

`Control1 = EPT` (Module prefix of Event Module for control cable)

`SeqResource = PTS.seq` (Depends on the application)
`MaxFreqMHz = 160` (160, 250, 300, 500, etc.)

Optional parameters for **ModuleType=PTS**:

`PTSAddress = 1` (1-4, depending on adaptor setting, default = ChannelNum)

Required parameters for **ModuleType=BrukerPTS**:

`Control1 = SD1` (Module prefix of Bruker SDC connected to the PTS)

Optional parameters for **ModuleType= BrukerPTS**:

none

Required parameters for **ModuleType=Synth5**:

`BaseAddr = 0x0100` (Backplane address = 0x0X00, depends on the application)
`RefDivisor = 16` (1-16, depends on the application)
`ReferenceFreq = 50` (MHz, depends on the application)
`Attenuator = 4` (dB, 0-15, as required)

Optional parameters for **ModuleType= Synth5**:

none

Parameters for compound synthesizer systems

Required parameters for **ModuleType=TecSyn**:

`Control1 = SY1` (Module prefix of Synth5)
`Control2 = 001` (Module prefix of DDDS)

Optional parameters for **ModuleType= TecSyn**:

`IdlerFreq = 900.0` (Any, MHz, default = 900 MHz)
`IdlerSideband = 1` (+/-1, upper/lower sideband for 900Gen, default = 1)
`OffsetMax = 10.0` (900Gen offset, MHz, default = 12.0 MHz)
`OffsetMin = 7.0` (900Gen offset, MHz, default = 7.0 MHz)
`AttenuationFile = TX1.AFF` (Filename for level equalization attenuator values)

Required parameters for **ModuleType=BrukerSynth**:

`Control1 = PT1` (Module prefix of Bruker PTS)
`Control2 = 001` (Module prefix of DDDS)

Optional parameters for **ModuleType= BrukerSynth**:

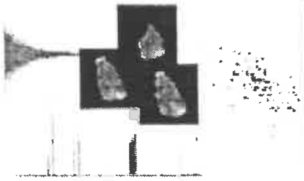
`none`	(But the BrukerHardware section must be filled out)

Required parameters for **ModuleType=SpecialSynth**:

`Control1 = SY1`	(Module prefix of Coarse generator)
`Control2 = O01`	(Module prefix of Fine generator)

Optional parameters for **ModuleType= SpecialSynth**:

`CoarseStepMHz = 750.0`	(MHz, default = 750 MHz)
`CoarseStartMHz = 750.0`	(MHz, default = 0 MHz)
`MixerSideband = 1`	(+/-1, upper/lower sideband, default = 1)
`FineMHzMax = 400.0`	(Fine offset, MHz, default = 400 MHz)
`FineMHzMin = 10.0`	(Fine offset, MHz, default = 10 MHz)



7- System & Module Block Diagrams

<<< PREV <<<

7.1

>>> NEXT >>>

Introduction

This chapter contains module and system block diagrams for Apollo, DSPect, and Orion systems. Due to the extensive use of DSPs and other programmable surface mount technology, schematics are not included.

<<< PREV <<<

7.2

>>> NEXT >>>

System Block Diagrams

Click below to jump to a block diagram.

Apollo - Low Frequency

Apollo - High Frequency

DSPect

Orion

<<< PREV <<<

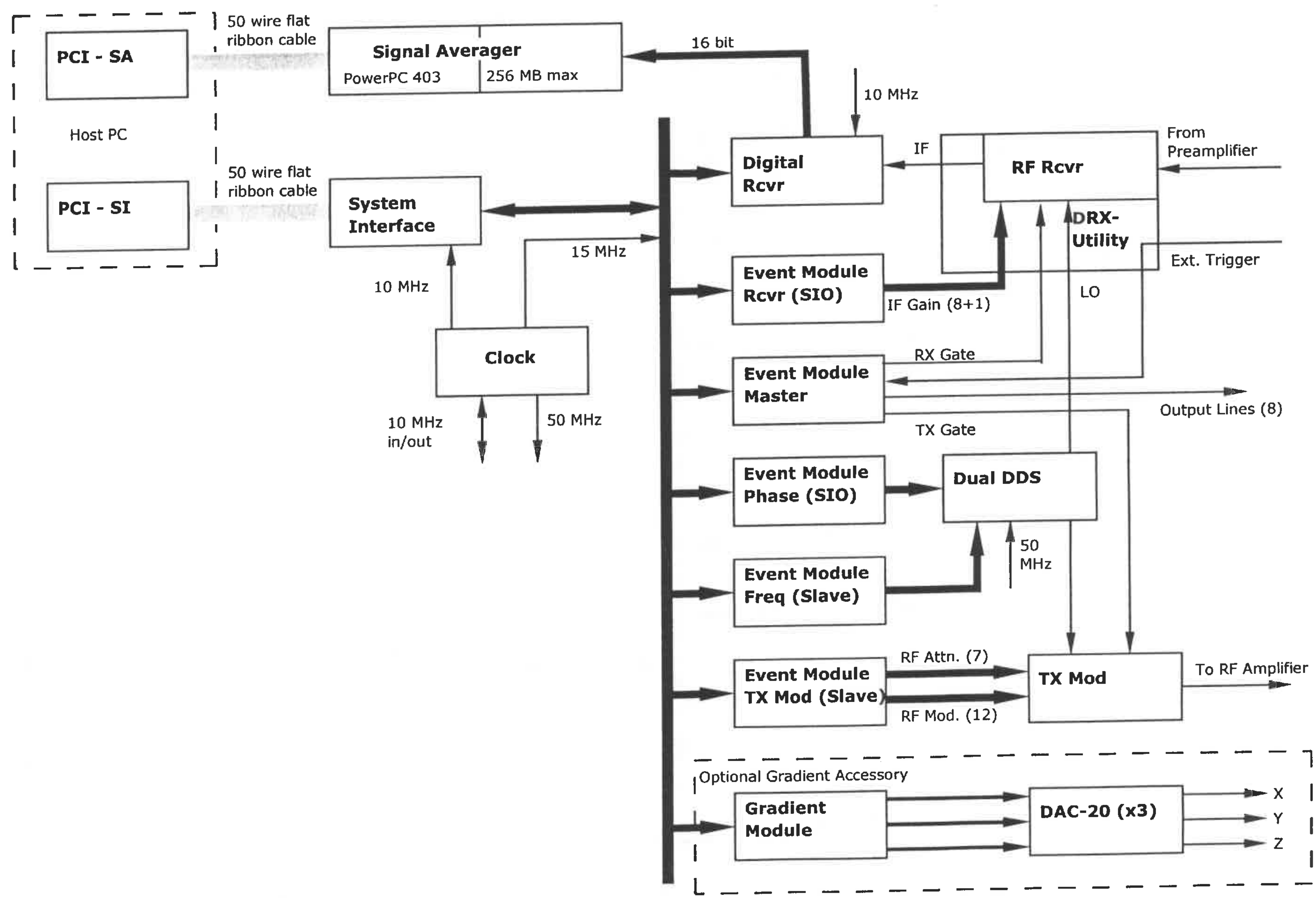
7.3

Module Block Diagrams

Click below to jump to a block diagram.

Apollo Low Freq - Block Diagram

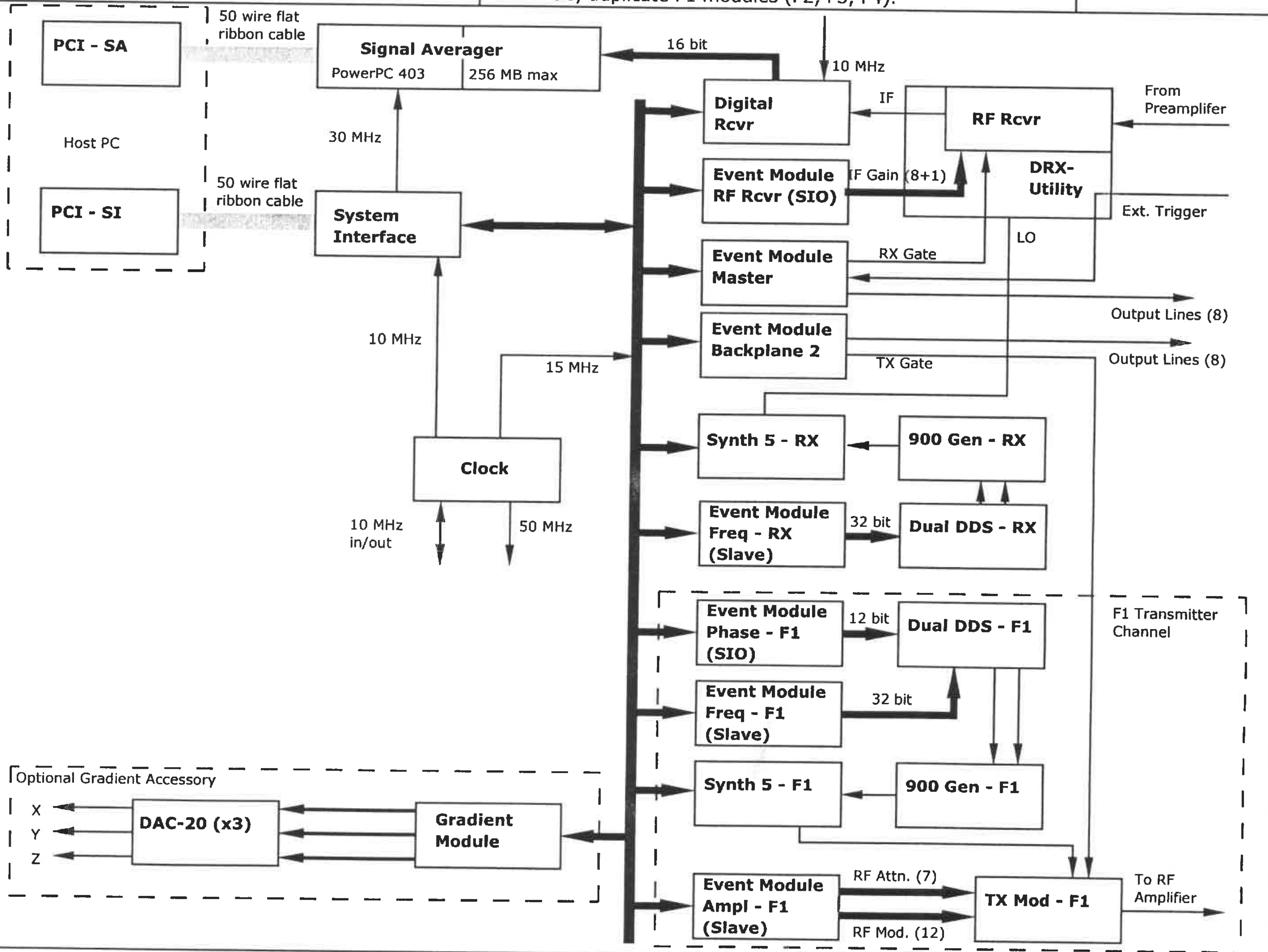
DBF90113



Apollo High Freq - Block Diagram

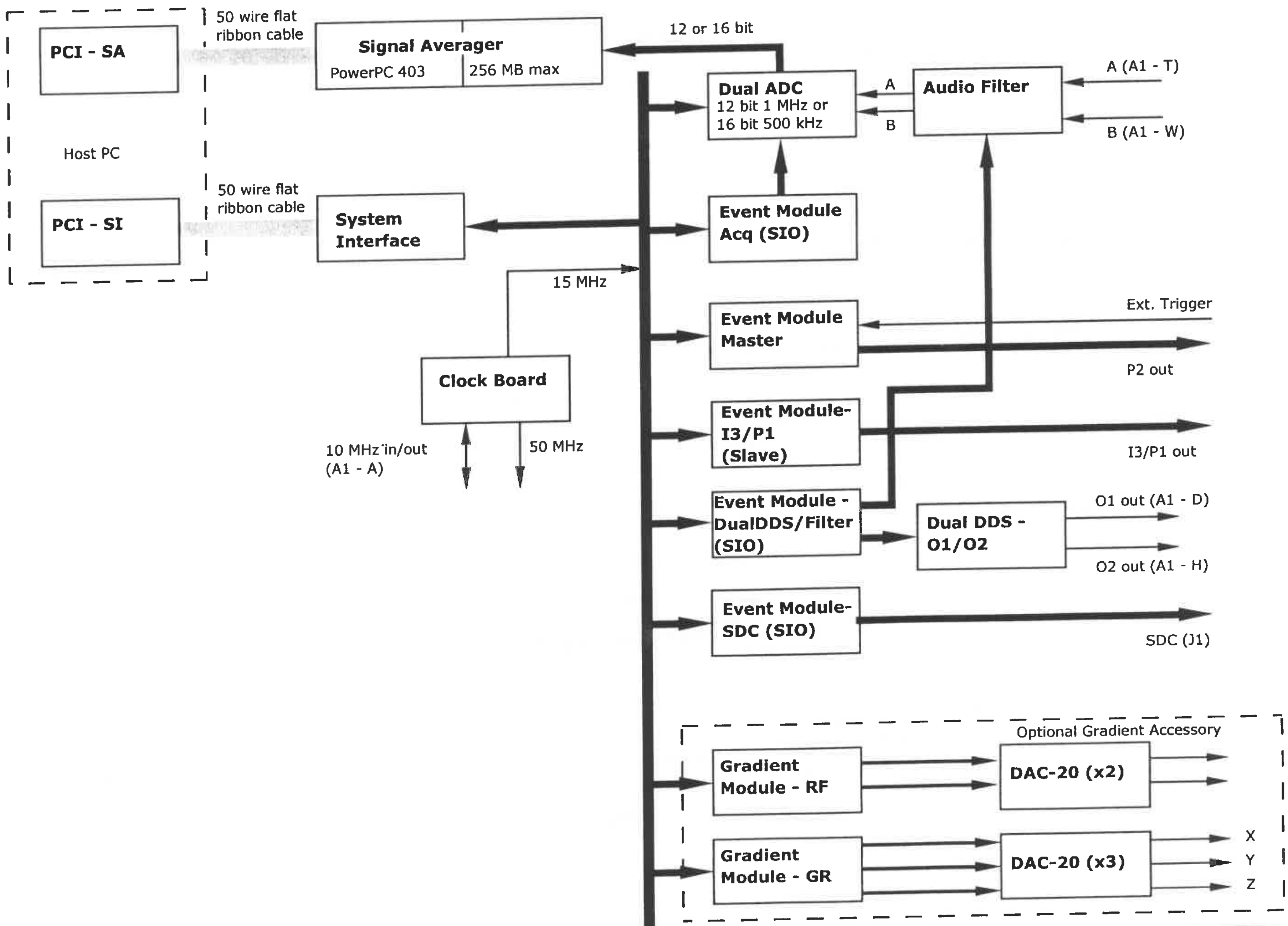
Single transmitter channel shown. For additional transmitter channels, duplicate F1 modules (F2, F3, F4).

DBF90113
CDG 012799



DSPect - Block Diagram

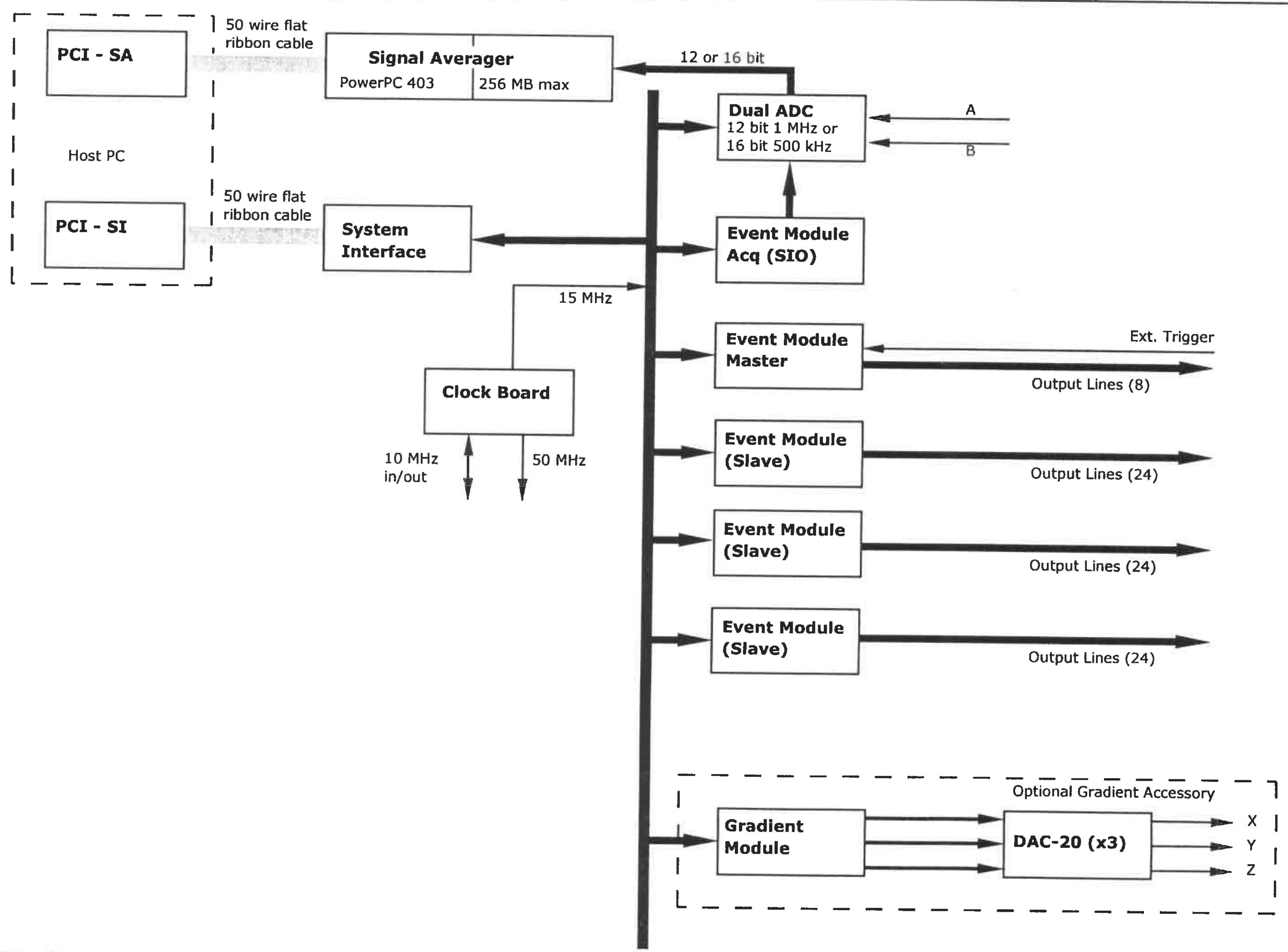
DBF90113



Orion - Block Diagram

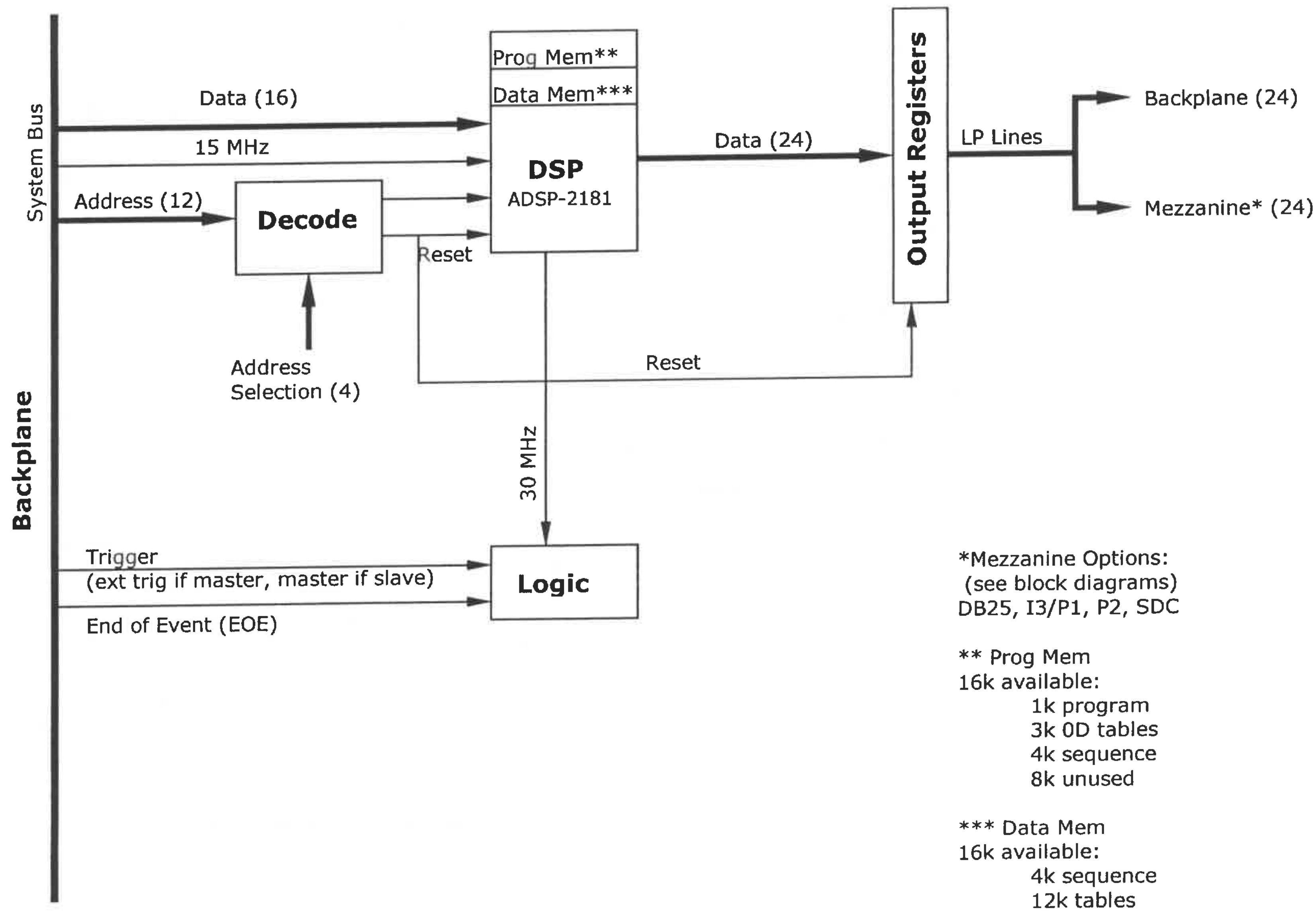
Additional Event Modules may be added for additional output lines

DBF90113



Event Module I - Block Diagram

DBF81215
DBF81223



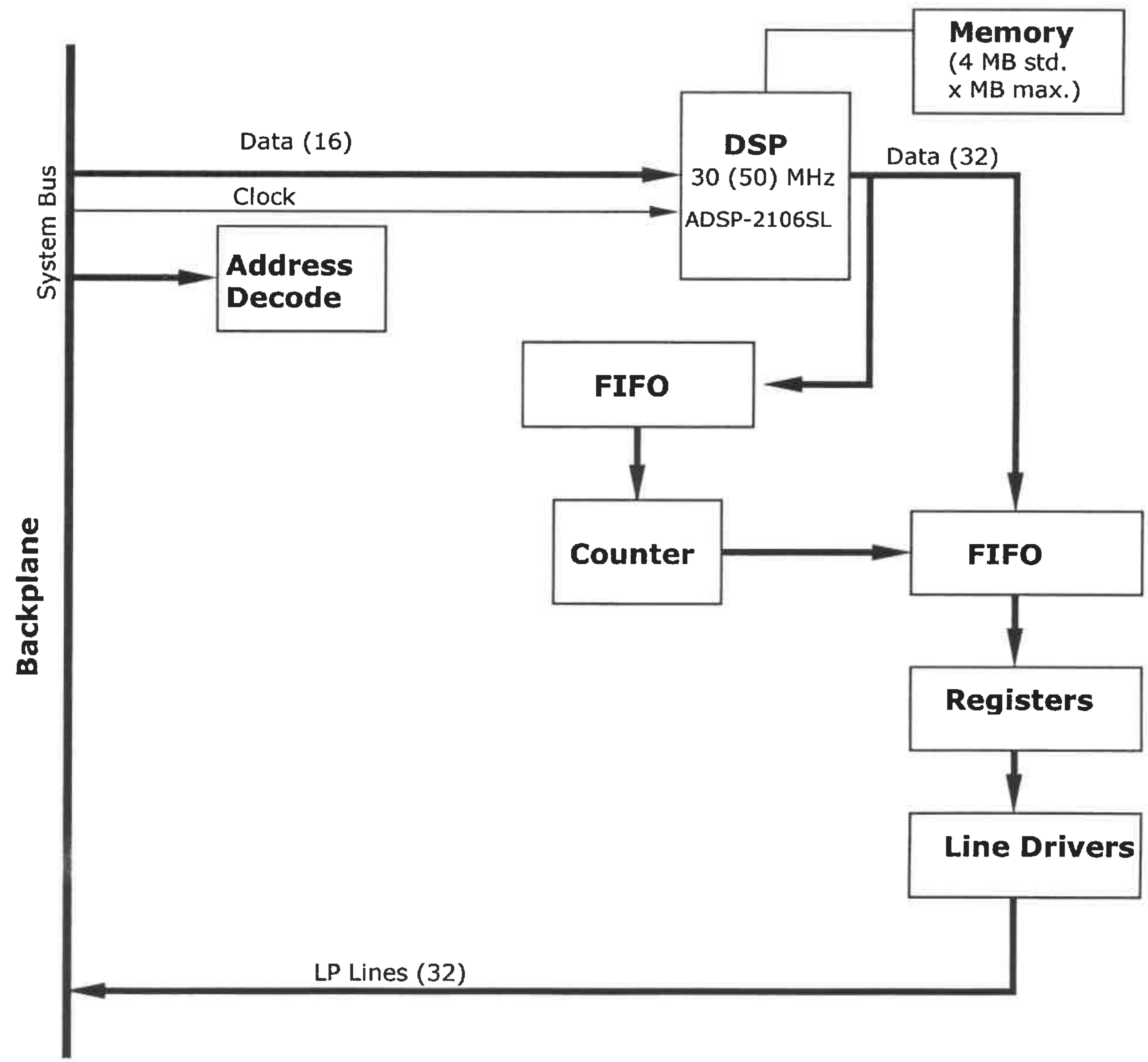
*Mezzanine Options:
(see block diagrams)
DB25, I3/P1, P2, SDC

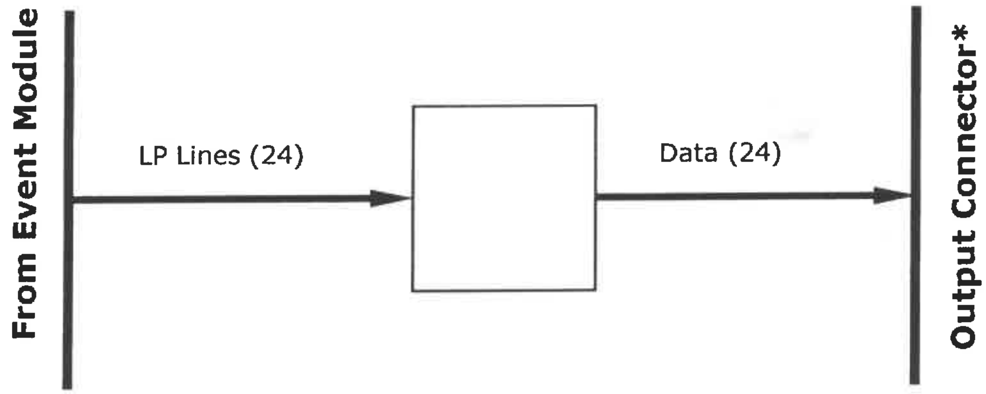
** Prog Mem
16k available:
1k program
3k 0D tables
4k sequence
8k unused

*** Data Mem
16k available:
4k sequence
12k tables

Event Module II - Block Diagram

DBF81215
DBF81221





*Output Connector Options (see page 2 for connector output maps)

DB25

pin	output
1	LP00
2	LP01
3	LP02
4	LP03
5	LP04
6	LP05
7	LP06
8	LP07
9	LP08
10	LP09
11	LP10
12	LP11
13	LP12
14	LP13
15	LP14
16	LP15
17	LP16
18	LP17
19	LP18
20	LP19
21	LP20
22	LP21
23	LP22
24	LP23
25	GND

I3/P1

pin	output
I3-H	LP17 (ATN3)
I3-A,B,C,D	NC
I3-E,F,J,K	NC
I3-L,M,N,P	NC
I3-R,S,T,U	NC
I3-V,W,X	NC
P1-A	LP? (SPF2B)
P1-B	LP20 (SPF1)
P1-C	LP? (DSPRFS1, AUTOSTOP)
P1-D	LP18 (PPF1)
P1-E	LP? (SPF1B)
P1-F	GND
P1-H	LP23 (ATN1)
P1-J	LP14 (PPF2)
P1-K	LP? (DF/DN, DECON)
P1-L	LP16 (ATN2)
P1-M	LP13 (EP B)
P1-N	GND
P1-P	LP12 (EP A)
P1-R	LP21 (SPF2)
P1-S	LP22 (MODON)
P1-U	LP11 (EPC)
P1-V	GND
P1-X	LP10 (PFPE)
P1-T,W	NC

P2

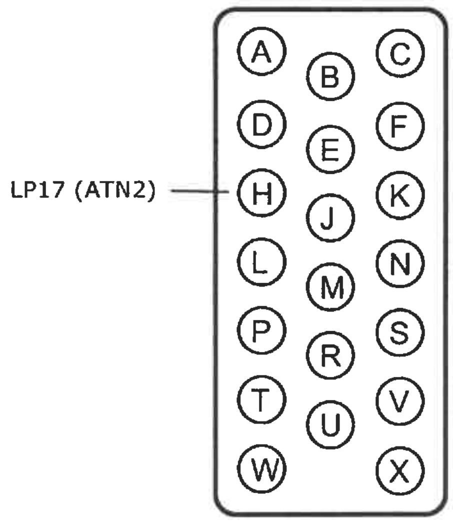
pin	output
P2-D	LP9 (RCP10)
P2-E	LP8 (RCP9)
P2-H	LP7 (RCP8)
P2-J	LP6 (RCP7)
P2-L	LP5 (RCP6)
P2-M	LP4 (RCP5)
P2-P	LP3 (RCP4)
P2-R	LP2 (RCP3)
P2-T	LP1 (RCP2)
P2-U	LP0 (RCP1)
P2-A,B,C,F	NC
P2-K,N,S,V	NC
P2-W,X NC	NC

SDC

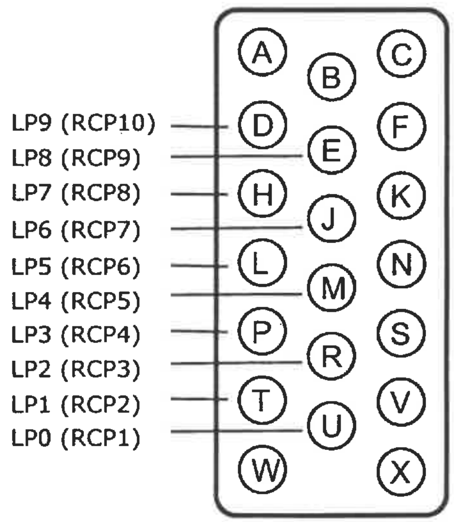
pin	output
1	LP00 (KW0)
2	LP01 (KW1)
3	LP02 (KW2)
4	LP03 (KW3)
5	LP04 (KW4)
6	LP05 (KW5)
7	LP06 (KW6)
8	LP07 (KW7)
9	LP08 (KW8)
10	LP09 (KW9)
11	LP10 (KW10)
12	LP11 (KW11)
13	LP12 (SDA0)
14	LP13 (SDA1)
15	LP14 (SDA2)
16	LP15 (SDA3)
17	LP16 (SDA4)
18	LP17 (SDA5)
19	LP18 (SDA6)
20	GND (GND)
21	+5V (+5V)
22	LP19 (SDFLAG-)
23	LP20 (TALK)
24	GND (GND)
25	LP23 (STROBE-)

DBF81215
DBF81223

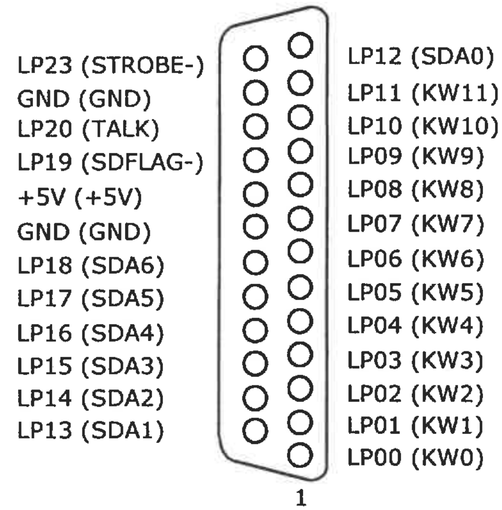
I3



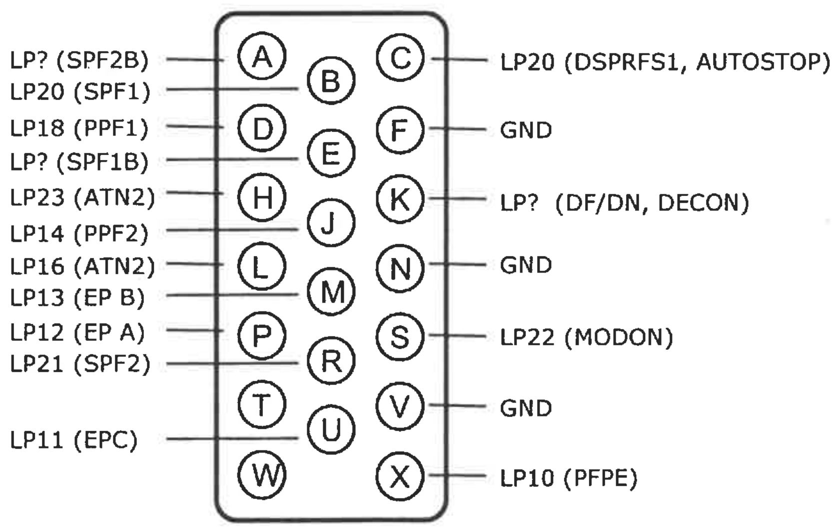
P2



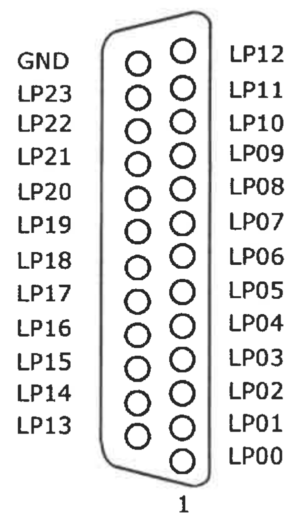
SDC

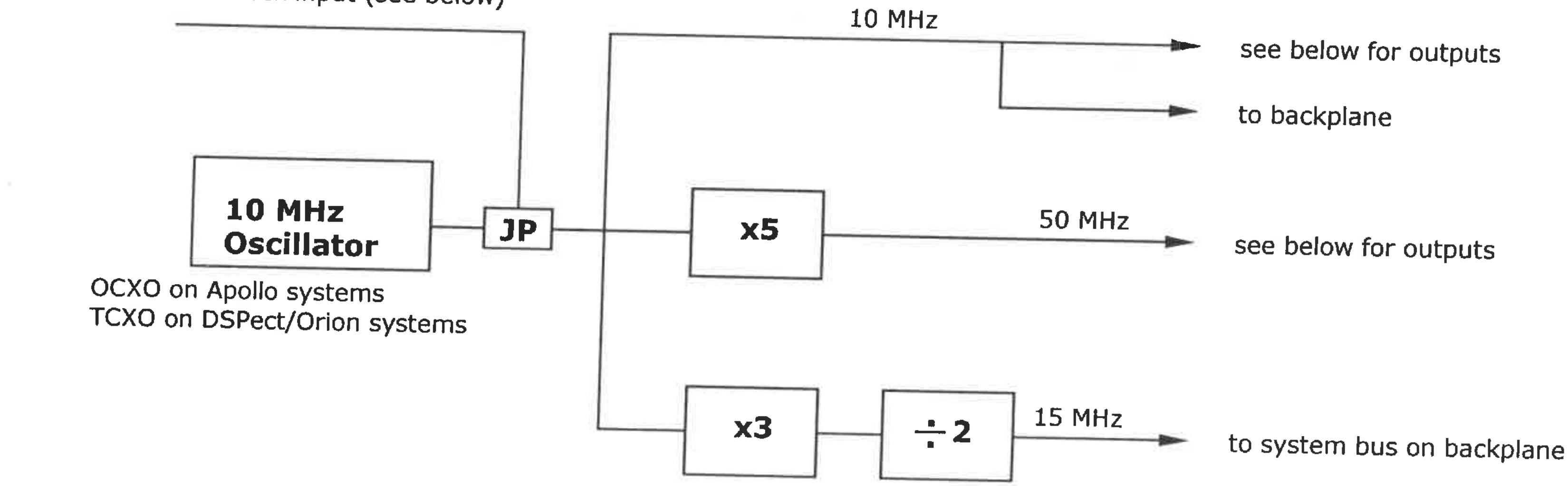


P1



DB25





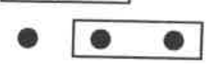
OCXO on Apollo systems
TCXO on DSPect/Orion systems

Outputs

- 10 MHz in/out*
- 10 MHz out**
- 50 MHz out
- 50 MHz out***
- 50 MHz out***
- 50 MHz out***

*The 10 MHz port can be configured as an input or an ouput, allowing system synchronization to an external 10 MHz reference by changing the jumpers as indicated below. Note that there are three locations on the board where the jumper must be changed: JP1, JP2, and JP3.

external clock


internal clock


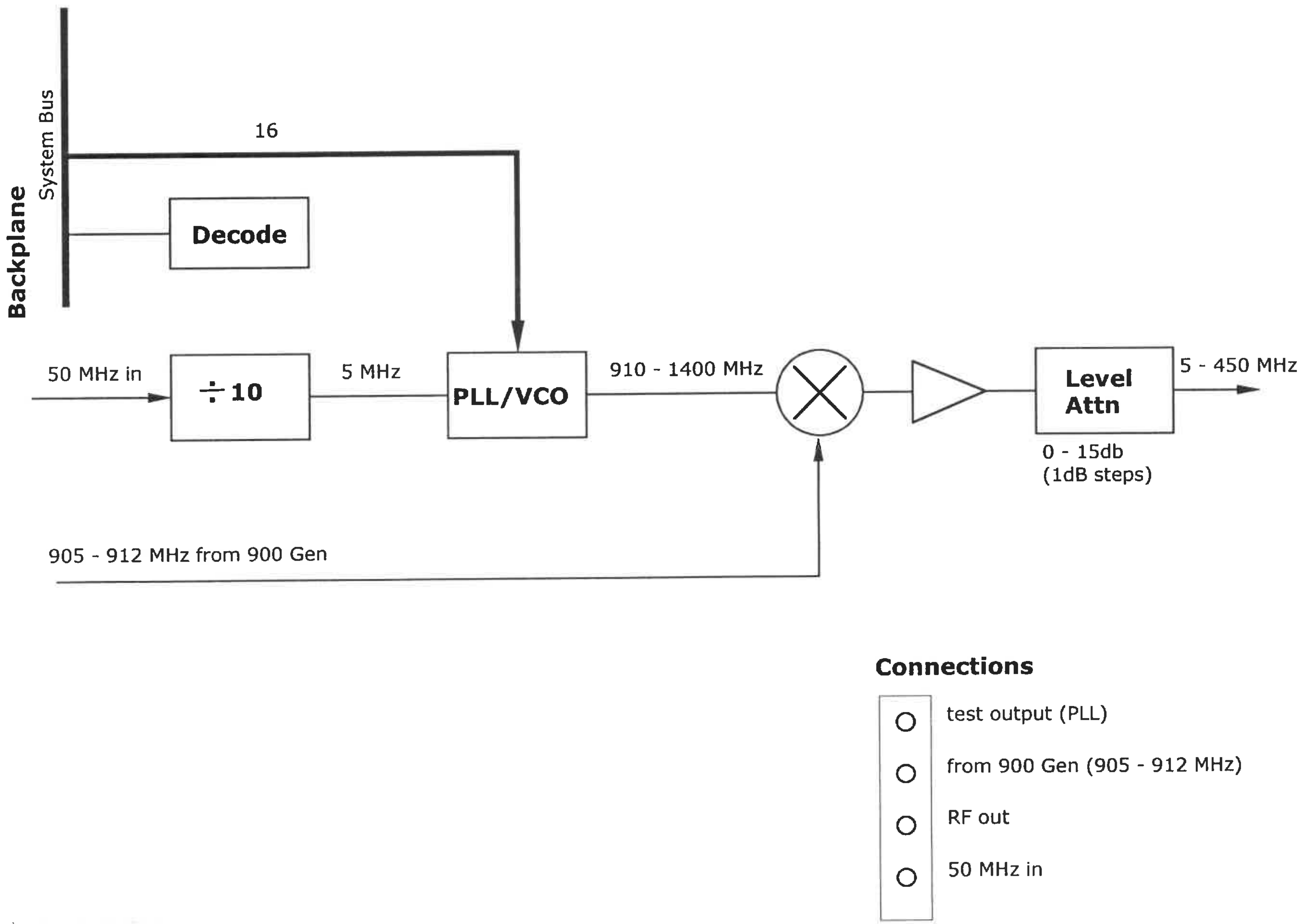
** Square wave output on Orion/DSPect systems

*** Not present on Orion/DSPect systems

Syn...

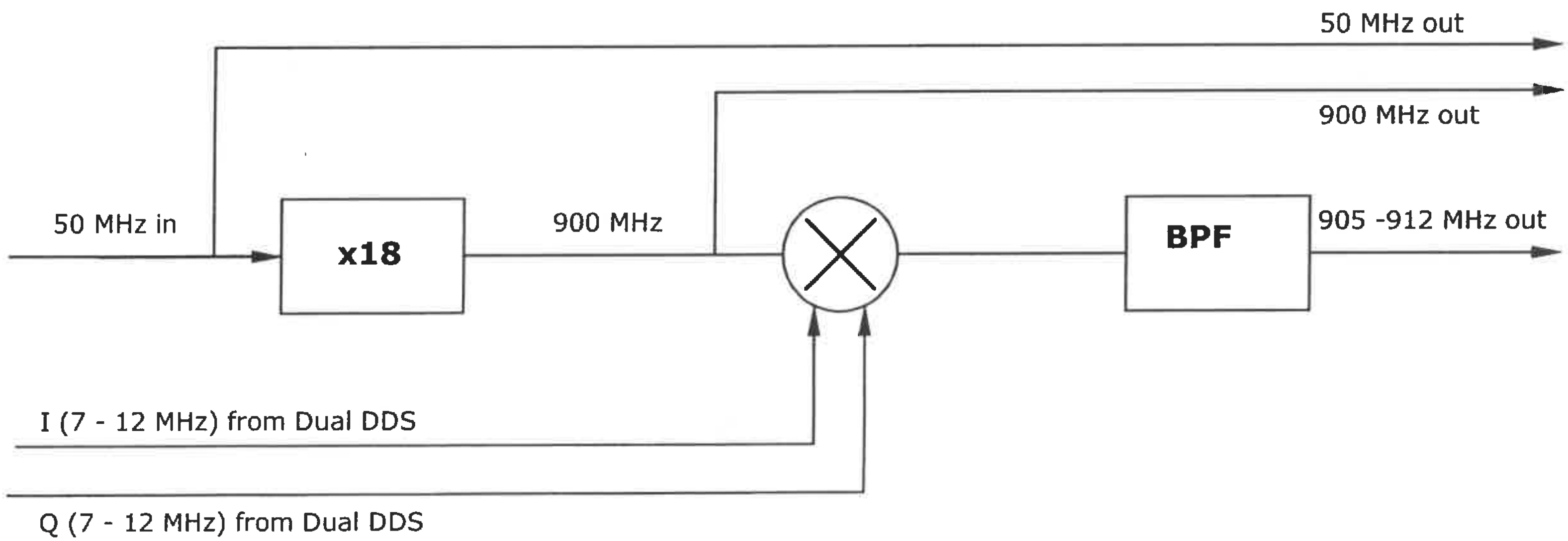
Synth 5 - Block Diagram

DBF90106
DBF90113



900 Gen - Block Diagram

DBF90106
DBF90113

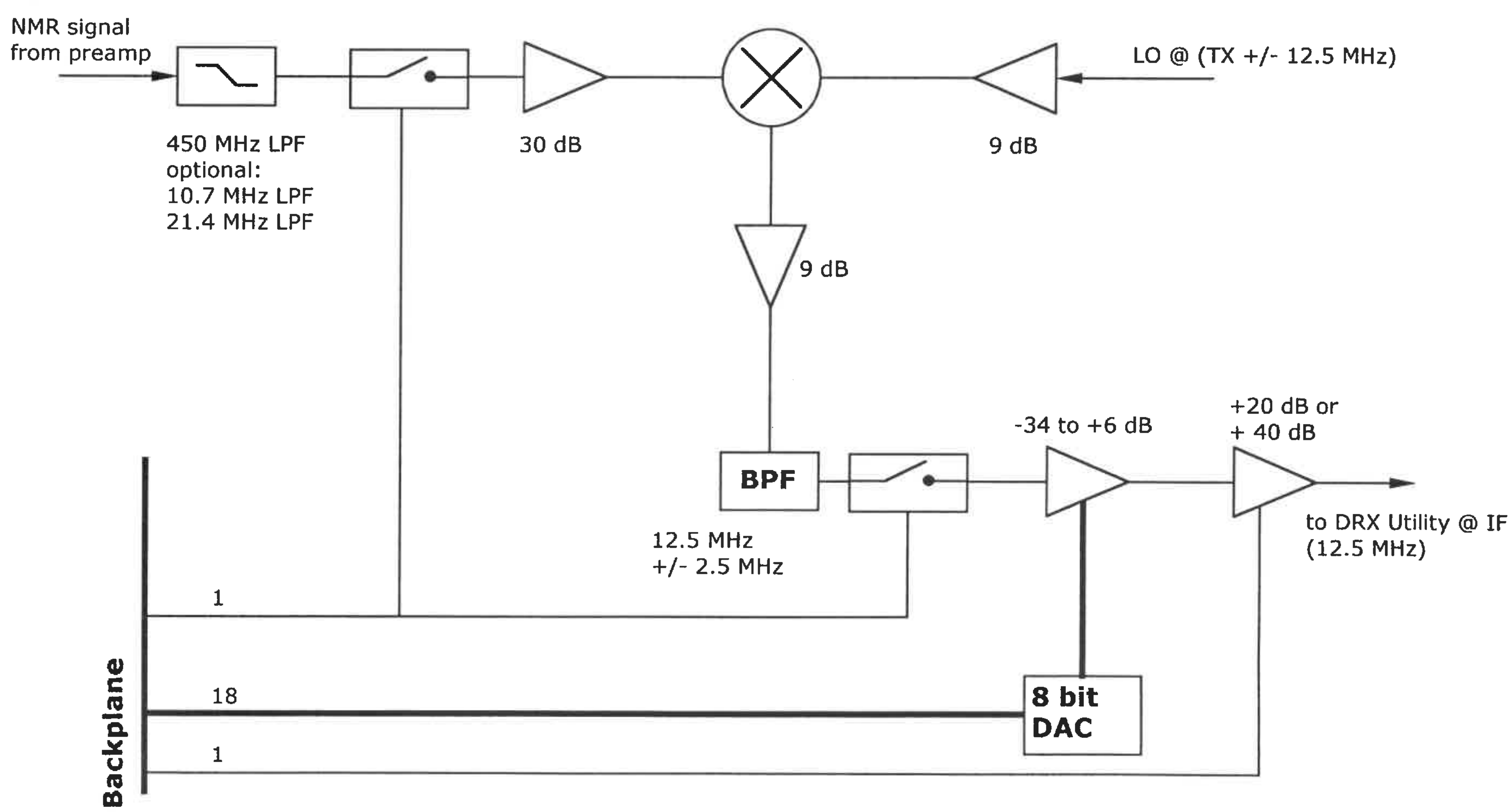


Connections

- 900 MHz out
- 900 MHz test
- Q input from Dual DDS
- I input from Dual DDS
- 50 MHz out
- 50 MHz in

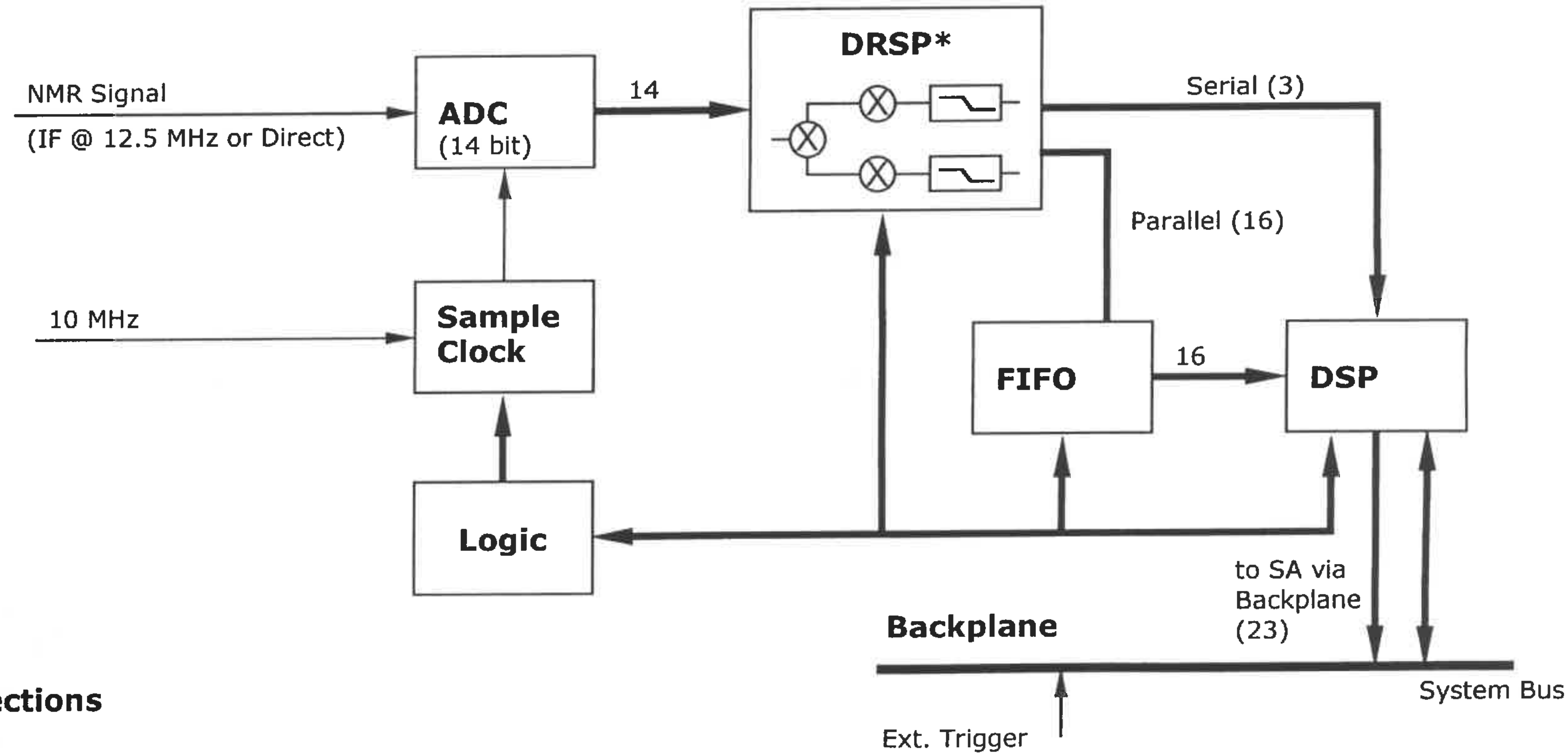
RF Rcvr - Block Diagram

DBF90106
DBF90113



Digital Receiver II - Block Diagram

DBF81215
DBF81221



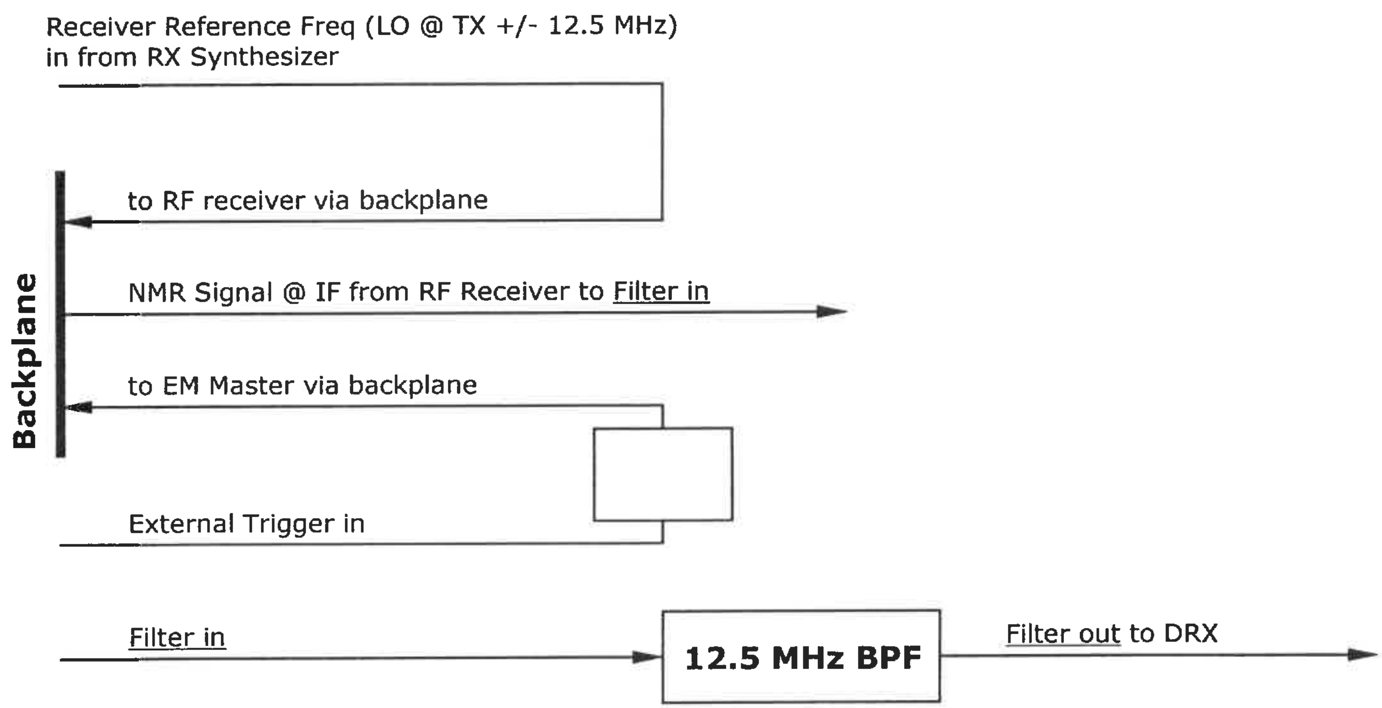
Connections

- NMR signal in @ IF
- Not used
- Clock in (10 MHz)

* DRSP Functions:
Digital Quadrature Detection
Acquisition Phase Cycling
Digital Filtering

DRX Utility - Block Diagram

DBF90113

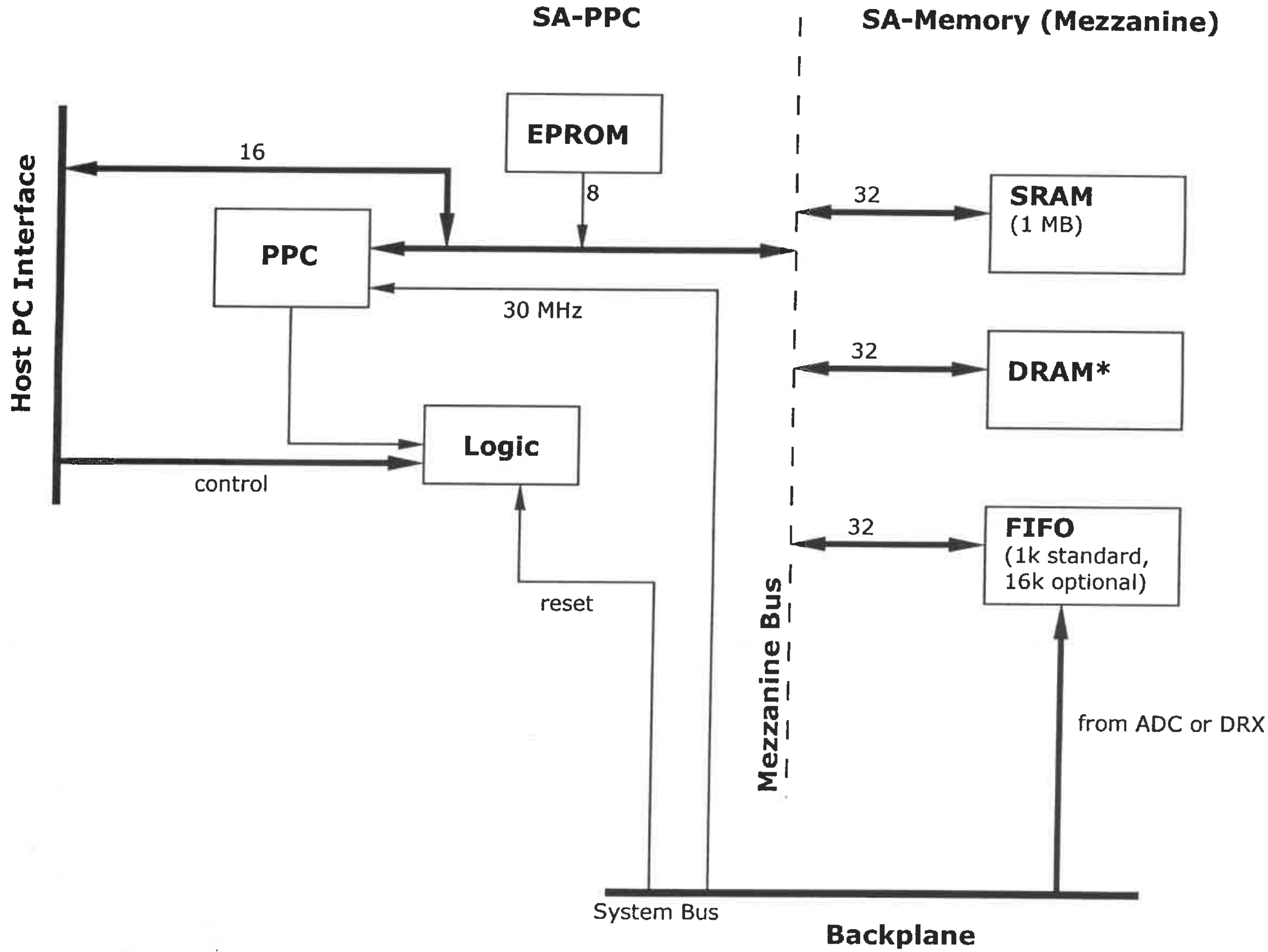


Connections

O	Receiver Reference Freq in (LO @ TX +/- 12.5 MHz)
X	
O	FIlter in
O	Filter out (to DRX)
O	External Trigger in
O	NMR Signal @ IF freq (demodulated signal from RF receiver)

Signal Averager - Block Diagram

DBF81215
DBF81221



* DRAM:
2 SIMM Slots
Max 128 MB per slot

Backplane - System Bus Lines

DBF81215

	A	B	C
1	15 MHz	+5v	15MHz
2	GND	GND	GND
3		STR	
4	A08	A00	
5	A09	A01	
6	A10	A02	
7	A11	A03	
8	A12	A04	
9	A13	A05	
10		A06	
11		A07	
12	RESET-	GND	
13	INITACK-	+5v	
14	WRITE-	D00	
15	READ-	D01	
16		D02	
17		D03	
18		D04	
19		D05	
20		D06	
21		D07	
22		GND	
23		D08	
24		D09	
25		D10	
26		D11	
27		D12	
28		D13	
29		D14	
30		D15	
31	-15v	GND	-15v
32	+15v	+5v	+15v

Tecmag Y2K Compatibility Statement

1. Software Products:

All Tecmag products operate under either Windows NT or the MacOS. Since both of these operating systems are Y2K compliant and Tecmag relies on the respective OS vendors for all date and time functions, there are no known Y2K issues with any version of NTNMR, MacNMR, or MacFID according to the definition below.

2. Hardware Products:

According to the definition below, all Tecmag hardware is Y2K compliant.

Definition of "Year 2000 Compliance"

To the best of our knowledge, we certify that neither the performance nor the functionality of any Tecmag product is affected by dates prior to, during, and after the year 2000.

In particular:

- No value for current date will cause any interruption in operation.
- Date-based functionality will behave consistently for dates prior to, during and after year 2000.
- In all interfaces and data storage, the century in any date is specified either explicitly or by unambiguous algorithms or inferencing rules.
- Year 2000 is recognized as a leap year.