ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Talise Release 6.0
February 8, 2019

API: 3.6.0.5

ARM 6.0.2

STREAM: 1.2.17.2

GUI: 3.6.0.5

2/8/2019

GUI changes

- Added Frequency Hopping GUI tab:
  - Enabling the user to evaluate frequency hopping feature in GPIO or Non-GPIO mode.
- Support for *ADRV9008-x and ADRV9009 Profile Configuration Tool* (Talise_Configuration_wizard.exe) generated custom profiles for open market release.
  - Allows the user to load the custom profiles generated with the Profile Configuration Tool.
- Support for User Controlled Rx Data Gating
  - When the Rx stream runs, it is possible to get a glitch of spurious information just prior to valid data over the link. Some customers have indicated that this is an unwanted effect. The Stream Generator now takes in a delay setting "ADCSettlingDelay" which can mute the output of the Rx path for a period of time until the datapath settles out. The control is limited 0-5us. This mute control only applies to the RX path.
  - This will affect the Enable times of various paths (RX-TX-ORX). It is suggested for this delay setting to remain at the default value of zero if the initial glitches do not cause a system problem.

- Eratta:
  - The init.c is not fully complete and the user will need to extend it. Refer to the python script as a secondary source.

API Overview

- DLL/API Version updated to 3.6.0.4
- Stream Generator DLL updated to 1.2.17.2
- New APIs:
  - New API to support RF LO Sync (TALISE_enableMultichipRfLOPhaseSync) to be used in init sequenece instead of previous MCS function. Slightly different device init sequence. The previous MCS function is still valid if RF LO Sync is not required.
  - Added TALISE_serializerReset(). It should be called after MCS to reset clock dividers at serializer. This is called automatically in backward support init cases (called in TALISE_enableMultichipSync when enableMcs parameter = 0) , but exposed incase needed for a custom init sequence or using TALISE_enableMultichipRfLOPhaseSync.
  - Added API to enable/disable Digital DC offset
    - Added TALISE_setDigDcOffsetEn
    - Added TALISE_getDigDcOffsetEn
  - Updated Support to read in custom configuration file that is output from the *ADRV9008-x and ADRV9009 Profile Configuration Tool* (Talise_Configuration_wizard.exe).
  - Added support for ORx ADC stitching to be disabled by zeroing out merge filter in the init structure when ORx ADC Bw > 200MHz.
  - AD9528 API: added support for PLL2 R1 divider (previously forced this divider to 1 - limiting use cases)
  - AD9528 API: Updated ad9528pll2Settings_t init structure to add PLL2 R1 divider and updates to other members.
- Bug Fixes:
  - Tx PFIR sync clock divider updated in cases where Tx FIR interpolation = 4x. The clock generated was running too fast previously.
  - FPGA configuration: fixed bug to calculate FPGA's LMFC clock rate to cover broad market use cases. Previously it only worked for standard use cases.

API Function file split

talise	talise_agc	talise_arm	talise_cals	talise_error
TALISE_openHw	TALISE_setupRxAgc	TALISE_initArm	TALISE_runInitCals	TALISE_getErrCode
TALISE_closeHw	TALISE_getAgcCtrlRegisters	TALISE_writeArmProfile	TALISE_waitInitCals	TALISE_getErrorMessage
TALISE_resetDevice	TALISE_getAgcPeakRegisters	TALISE_loadArmFromBinary	TALISE_checkInitCalComplete	TALISE_getRegisteredErrorMessage
TALISE_initialize	TALISE_getAgcPowerRegisters	TALISE_loadAdcProfiles	TALISE_abortInitCals	
TALISE_shutdown	TALISE_setupDualBandRxAgc	TALISE_readArmMem	TALISE_getInitCalStatus	
TALISE_enableMultichipSync	TALISE_getDualBandLnaControls	TALISE_writeArmMem	TALISE_enableTrackingCals	
TALISE_programFir	TALISE_setRxAgcMinMaxGainIndex	TALISE_writeArmConfig	TALISE_getEnabledTrackingCals	
TALISE_calculateDigitalClocks	TALISE_resetRxAgc	TALISE_readArmConfig	TALISE_getPendingTrackingCals	
TALISE_verifyProfiles		TALISE_readArmCmdStatus	TALISE_rescheduleTrackingCal	
TALISE_setSpiSettings		TALISE_readArmCmdStatusByte	TALISE_setAllTrackCalState	
TALISE_verifySpiReadWrite		TALISE_waitArmCmdStatus	TALISE_getAllTrackCalState	
TALISE_initDigitalClocks		TALISE_sendArmCommand	TALISE_getTxLolStatus	
TALISE_setTxPfirSyncClk		TALISE_getArmVersion (deprecated)	TALISE_getTxQecStatus	
TALISE_setRxPfirSyncClk		TALISE_verifyArmChecksum	TALISE_getRxQecStatus	
TALISE_getApiVersion		TALISE_getArmVersion_v2	TALISE_getOrxQecStatus	
TALISE_getDeviceRev			TALISE_getRxHd2Status	
TALISE_getProductId			TALISE_waitForEvent	
TALISE_getMultiChipSyncStatus			TALISE_readEventStatus	
TALISE_enableMultichipRfLOPhaseSync			TALISE_resetExtTxLolChannel	
TALISE_serializerReset			TALISE_setRxHd2Config	
			TALISE_getRxHd2Config	
			TALISE_setDigDcOffsetMShift	
			TALISE_getDigDcOffsetMShift	
			TALISE_getDigDcOffsetEn	
			TALISE_setDigDcOffsetEn	

- Green functions are new public Talise API functions
- Yellow functions are not fully supported because no device profiles allow the dualband feature to be enabled. Dualband will be supported in a future release. This code was added at the same time as the supported Low IF to Zero IF feature which does have a new Rx profile.
- Red, prototype change

API Function file split

talise_gpio	talise_jesd204	talise_radioctrl	talise_rx	talise_tx
TALISE_setGpioOe	TALISE_setupSerializers	TALISE_loadStreamFromBinary	TALISE_programRxGainTable	TALISE_setTxAttenuation
TALISE_getGpioOe	TALISE_setupDeserializers	TALISE_setArmGpioPins	TALISE_programOrxGainTable	TALISE_getTxAttenuation
TALISE_setGpioSourceCtrl	TALISE_setupJesd204bFramer	TALISE_setRadioCtlPinMode	TALISE_setRxManualGain	TALISE_setDacFullScale
TALISE_getGpioSourceCtrl	TALISE_setupJesd204bDeframer	TALISE_getRadioCtlPinMode	TALISE_getRxGain	TALISE_enableTxNco
TALISE_setGpioPinLevel	TALISE_enableFramerLink	TALISE_setOrxLoCfg	TALISE_setObsRxManualGain	TALISE_setTxAttenCtrlPin
TALISE_getGpioPinLevel	TALISE_enableDeframerLink	TALISE_getOrxLoCfg	TALISE_getObsRxGain	TALISE_getTxAttenCtrlPin
TALISE_getGpioSetLevel	TALISE_enableSysrefToFramer	TALISE_radioOn	TALISE_setRxGainControlMode	TALISE_setPaProtectionCfg
TALISE_setGpioMonitorOut	TALISE_enableSysrefToDeframer	TALISE_radioOff	TALISE_setRxDataFormat	TALISE_getPaProtectionCfg
TALISE_getGpioMonitorOut	TALISE_readFramerStatus	TALISE_getRadioState	TALISE_getRxDataFormat	TALISE_enablePaProtection
TALISE_setGpIntMask	TALISE_readDeframerStatus	TALISE_setRxTxEnable	TALISE_getSlicerPosition	TALISE_getTxSamplePower
TALISE_getGpIntStatus	TALISE_setupDacSampleXbar	TALISE_getRxTxEnable	TALISE_setRxGainCtrlPin	TALISE_getPaProtectErrorFlags
TALISE_getGpIntMask	TALISE_setupAdcSampleXbar	TALISE_setTxToOrxMapping	TALISE_getRxGainCtrlPin	TALISE_clearPaProtectErrorFlags
TALISE_getTemperature	TALISE_enableFramerTestData	TALISE_setRfPllFrequency	TALISE_programDualBandLnaGainTable	
TALISE_setupAuxDacs	TALISE_injectFramerTestDataError	TALISE_getRfPllFrequency	TALISE_setGainTableExtCtrlPins	
TALISE_writeAuxDac	TALISE_enableDeframerPrbsChecker	TALISE_getPllsLockStatus	TALISE_getRxDecPower	
TALISE_setSpi2Enable	TALISE_clearDeframerPrbsCounters	TALISE_setRfPllLoopFilter (deprecated)		
TALISE_getSpi2Enable	TALISE_readDeframerPrbsCounters	TALISE_getRfPllLoopFilter (deprecated)		
TALISE_setGpio3v3Oe	TALISE_getDfrmIlasMismatch	TALISE_setPllLoopFilter		
TALISE_getGpio3v3Oe	TALISE_getDfrmIrqMask	TALISE_getPllLoopFilter		
TALISE_gpIntHandler	TALISE_setDfrmIrqMask	TALISE_setOrxLoSource		
TALISE_setAuxAdcPinModeGpio	TALISE_clearDfrmIrq	TALISE_getOrxLoSource		
TALISE_getAuxAdcPinModeGpio	TALISE_getDfrmIrqSource	TALISE_setFhmConfig		
TALISE_startAuxAdc	TALISE_framerSyncbToggle	TALISE_getFhmConfig		
TALISE_readAuxAdc		TALISE_setFhmMode		
TALISE_setGpio3v3SourceCtrl		TALISE_getFhmMode		
TALISE_getGpio3v3SourceCtrl		TALISE_setFhmHop		
TALISE_setGpio3v3PinLevel		TALISE_getFhmRfPllFrequency		
TALISE_getGpio3v3PinLevel		TALISE_getFhmStatus		
TALISE_getGpio3v3SetLevel		TALISE_setExtLoOutCfg		
		TALISE_getExtLoOutCfg		

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

API Function prototype changes

- No changes to existing function prototypes

API Structure changes

- AD9528 API:

```
/**
 * \brief Structure to hold AD9528 PLL2 settings
 */
typedef struct
{
    uint8_t rfDivider;  /* VCO divider: Valid range (3,4,5) */
    uint16_t n2Divider; /*!< PLL2 N2 Divider */
    uint8_t r1Divider;      /*! PLL2 R1 Divider */
    uint8_t totalNdiv; /*!< NDiv = 4*Bdiv + Adiv //Bdiv valid range (3 to 63), Adiv valid range (0-3) */

} ad9528pll2Settings_t;
```

- Added r1Divider member (ability to divide PLL2 input clock (1 to 31))
- Renamed nDivider and updated to two new fields n2Divider and totalNdiv for code clarity.

API ENUM changes

- No changes

New API functions

- **uint32_t TALISE_enableMultichipRfLOPhaseSync(taliseDevice_t *device, uint8_t enableDigTestClk);**

LOs on multiple chips can be phase synchronized to support active
antenna system and beam forming applications.This function should
be run after all transceivers have finished the TALISE_setRfPllFrequency(),
and before TALISE_runInitCals().

When the enableDigTestClk parameter = 1, this function will reset the MCS state
machine in the Talise device and switch the ARM to run on device clock scaled instead of HSDIGCLK
When the enableDigTestClk parameter = 0, this function will enable Mcs Digital Clocks Sync and JESD204 sysref,
switch the ARM back the HSDIGCLK.

Typical sequence:
1) Initialize all Talise devices in system using TALISE_initialize(),load the ARM and stream processor
2) Set the RF PLL frequency with TALISE_setRfPllFrequency
3) Run TALISE_enableMultichipRfLOPhaseSync with enableDigTestClk = 1 before TALISE_runInitCals()
4) Send at least 4 SYSREF pulses
5) Run TALISE_enableMultichipRfLOPhaseSync with enableDigTestClk = 0
6) Send at least 4 SYSREF pulses
7) Continue with init sequence ...Run initCals, bring up JESD204, etc

- **uint32_t TALISE_serializerReset(taliseDevice_t *device);**
  - Normally called by TALISE_enableMultichipSync(), but if using TALISE_enableMultichipRfLOPhaseSync, TALISE_serializerReset should be called after MCS, and before bringing up the JESD204 links.

New API functions

- **uint32_t TALISE_setDigDcOffsetEn(taliseDevice_t *device, uint8_t enableMask);**
- **uint32_t TALISE_getDigDcOffsetEn(taliseDevice_t *device, uint8_t *enableMask);**

  These functions are used to enable/disable Digital DC offset tracking if required.

```
Sets/Reads back Enable/ Disable channels Digital DC Offset and returns a mask of it.
 * The mask returned will be a combination of the following channel values ( ::taliseRxDcOffsettEn_t ).
 * By default RX BBDC tracking is ON and ORX BBDC tracking is OFF.  This API function can be used to change default behavior.
 *
 *    Channel                |  Value  |  Channel description
 * -------------------------|---------|--------------------------
 *  TAL_DC_OFFSET_ALL_OFF   |   0x00  | All channels are disabled
 *  TAL_DC_OFFSET_RX1       |   0x01  | Rx1 is enabled
 *  TAL_DC_OFFSET_RX2       |   0x02  | Rx2 is enabled
 *  TAL_DC_OFFSET_ORX1      |   0x04  | ORx1 is enabled
 *  TAL_DC_OFFSET_ORX2      |   0x08  | ORx2 is enabled
 *  TAL_DC_OFFSET_ALL_ON    |   0x0F  | All channels are enabled
```

New API Structures

- No New Structures added to API

New API ENUMs

```
 *  \brief Enum of Rx/ORx channels mask  for configuring (Enable /disable) DC offsets.
 */
typedef enum
{
    TAL_DC_OFFSET_ALL_OFF = 0x00,               /*!< Disable all the channels */
    TAL_DC_OFFSET_RX1 = 0x01,                   /*!< Enables Rx1  */
    TAL_DC_OFFSET_RX2 = 0x02,                   /*!< Enables Rx2  */
    TAL_DC_OFFSET_ORX1 = 0x04,                   /*!< Enables ORx1  */
    TAL_DC_OFFSET_ORX2 = 0x08,                   /*!< Enables ORx2  */
    TAL_DC_OFFSET_ALL_ON = 0x0F              /*!< Enables all the channels  */
}taliseRxDcOffsettEn_t;
```

Known Issues / Errata

1) GUI: The init.c is not fully complete and the user will need to extend it. Refer to the python script as a secondary source.