UCCP Spectrum Analyser

Register Description Document

Filename   :   UCCP Spectrum Analyser.Register Description Document.docx

Version   :   1.0.59 Not Issued - Live Document

Issue Date   :   17 Oct 2013

Author   :   Imagination Technologies Ltd

Contents

IF{{Internal}}

Document History

Issue	Date	Changes/Comments
1.0.59	02 Jul 2013	External Issue.

END{{Internal}}

IF{{Internal}}

1. Introduction

This document describes registers in Imagination Technologies' UCCP Spectrum Analyser IP, which is intended for use in conjunction with Imagination Technologies' TV demodulator IP. The purpose of the Spectrum Analyser IP is to provide a convenient means of measuring the power spectral density in the signal received from a TV tuner. The Spectrum Analyser IP may be used to detect signalling, to assist in scanning a frequency band searching for valid signals, to measure tuner performance, or for other purposes.

The Spectrum Analyser IP is controlled via "virtual registers" in a similar fashion to Imagination Technologies' demodulator IP. The standard set of control registers for a TV demodulator is outlined in reference [1]. In the case of the Spectrum Analyser IP, a sub-set of these registers are used along with a simplified form of behavioural model.

2. System Operation

Please refer to reference [1] sections 1 and 2 for a general description of the virtual register system and command register operation.

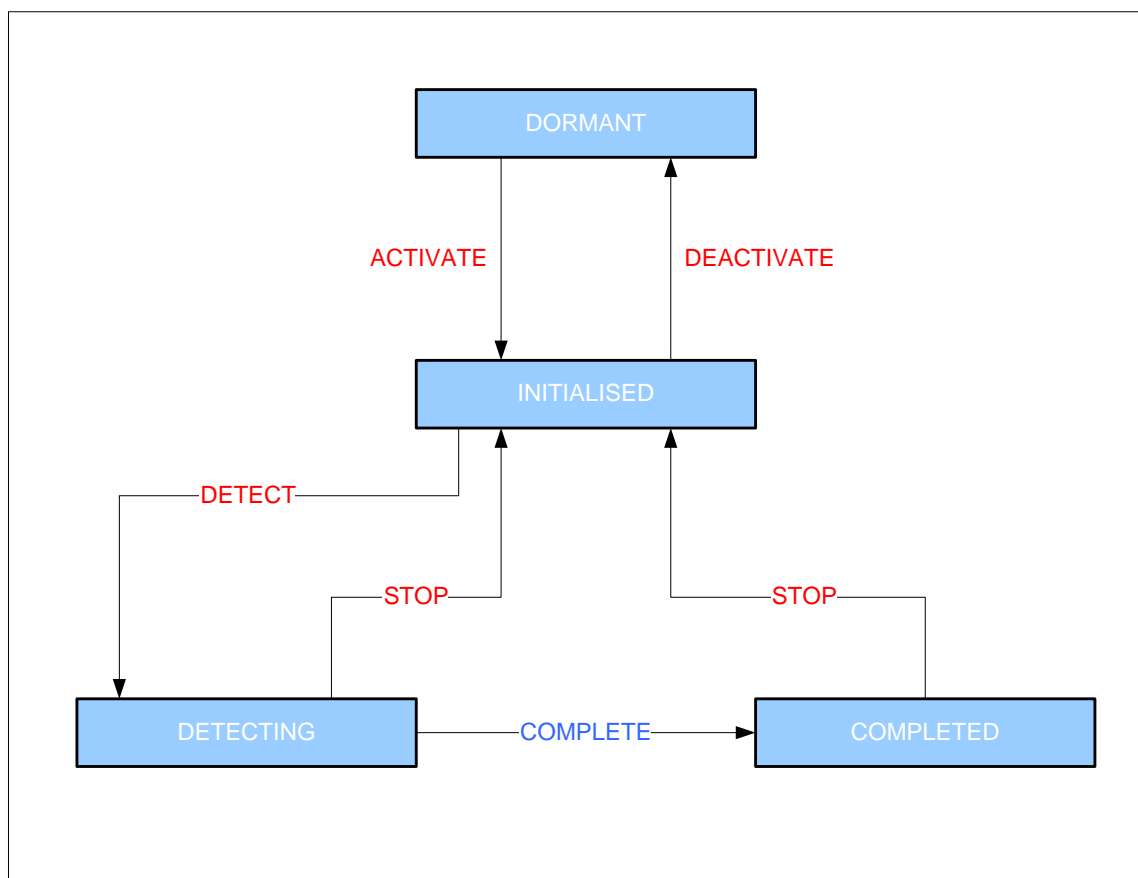The Spectrum Analyser IP uses a cut-down behavioural model illustrated in Figure 1:

Figure 1: Spectrum Analyser Behavioural Model

The transitions in the state machine are driven by two external *commands* (generated by the controlling application) and one internally generated *event*. The behaviour of the system is further controlled by values written to control registers associated with the Spectrum Analyser IP. The full set of control registers is described in section 3.

2.1. Basic Operating Sequence

To perform a spectrum scan, the sequence of operations is as follows:

1. Controller activates an instance of the Spectrum Analyser. The UCCP software returns an instance identifier, which is subsequently used to identify the particular spectrum analyser target. The activation operation corresponds to the state transition from *DORMANT* to *INITIALISED*. At this point, the virtual register interface is operational.

2. The controller monitors the operation of the spectrum analyser by observing the value in the `TV_REG_STATE` register.

3. Controller sets up the scan parameters using the API registers.

4. Controller writes `TV_CMD_DETECT` to the `TV_REG_CONTROL` register, causing the spectrum analyser to initially enter the *DETECTING* state. This starts the scan process.

5. When the scan is complete, the state machine shall move to the *COMPLETED* state. In this state, results of the scan are available.

6. The system can be stopped by the controller writing `TV_CMD_STOP` to the `TV_REG_CONTROL` register. This shall cause a return to the *INITIALISED* state.

7. Once the system returns to the *INITIALISED* state, the spectrum analyser can be deactivated, which releases resources for use by TV demodulators etc.

2.2. API Terminology

Commands	
ACTIVATE	Prepare for use, allocate memory, resources etc.
DEACTIVATE	Remove from active system, Release resources.
DETECT	Run a scan
STOP	Return to INITIALISED state.
States	
DORMANT	Standard not available for use. No resources have been allocated. Virtual register API not available
INITIALISED	Resources allocated successfully, The virtual register interface is avalable.
DETECTING	Scan in progress
COMPLETED	Scan results available
Events	
COMPLETE	Scan completed

2.3.   Tuner integration

TV demodulators support two system design models for integrating tuners. The behaviour of some of the API registers varies according to which model is used.

2.3.1.   Locally controlled tuners

In this design model the RF tuner is controlled completely autonomously by the UCCP firmware.   In the case of a TV demodulator system, changes to RF frequency and bandwidth are requested by the controlling application via the `TV_REG_TUNER_FREQ` and `TV_REQ_TUNER_BW` registers. The currently selected values are made available to the controller via the `TV_REG_ACTIVE_TUNER_FREQ` and `TV_REG_ACTIVE_TUNER_BW` registers. Often these registers do no more than mirror the values requested by the controller, but some systems adjust tuner frequency and bandwidth as part of their signal tracking functionality. Such changes are reflected back via the `TV_REG_ACTIVE_TUNER_FREQ` and `TV_REG_ACTIVE_TUNER_BW` registers as they happen.

In the case of the spectrum analyser IP, the host sets up the `TV_REG_TUNER_FREQ` and `TV_REQ_TUNER_BW` registers prior to a scan. `TV_REG_TUNER_FREQ` indicates the starting frequency for the scan, and `TV_REQ_TUNER_BW` sets the tuner bandwidth to be used when capturing each spectral fragment during the scan. Once the scan is running, the spectrum analyser IP has autonomous control of tuner frequency and bandwidth.

The UCCP firmware controls the tuner via an abstracted Tuner API, which is implemented by a tuner driver module in the software. To support a different RF tuner, a new version of the UCCP software must be built with a suitable tuner driver.

2.3.2.   Host controlled tuners

In this design model the RF tuner is largely controlled by the host processor. Typically, frequency and bandwidth selection are made by the host using its own hardware interface. Only analogue AGC control is usually performed directly by the UCCP firmware. The host controlled design model may be preferred for a number of reasons, including:

- The system integrator already has peripheral device interface hardware available for the host processor.
- Support for different tuners is achieved by including appropriate tuner drivers in the host application. This software development environment may be more familiar to the system integrators.
- It is possible to support different tuners without changing the UCCP software at all.

In this design model, the interpretation of the `TV_REG_TUNER_FREQ` and `TV_REQ_TUNER_BW` registers is somewhat different, rather than being *requests* for the UCCP firmware to modify the tuner, they are treated as *notifications* to the firmware that the host *has modified* the relevant tuner settings.

Similarly, the interpretation of the `TV_REG_ACTIVE_TUNER_FREQ` and `TV_REG_ACTIVE_TUNER_BW` registers is different in this model. Rather than being notifications that the UCCP firmware has changed the tuner settings, these registers are now considered to be requests to the host to make changes. The host is expected to reply to these requests with the updated values via the `TV_REG_TUNER_FREQ` and `TV_REQ_TUNER_BW` registers. This allows UCCP systems to manage any delay in the host responding to the tune requests.

Tuner Frequency Grid

Many tuners limit the available frequency settings to discrete values – typically on an equally spaced "grid". In cases where the UCCP firmware requests frequency updates, the requests are limited to those which will move the tuner frequency from one grid position to another. This avoids the possibility of flooding the communications channel with small frequency adjustments, that would effectively be ignored by the host. The host notifies the UCCP of its frequency grid restrictions by writing to the `TV_REG_TUNER_GRID_BASE` and `TV_REG_TUNER_GRID_INCR` registers. Frequency grid filtering can be disabled by setting `TV_REG_TUNER_GRID_INCR` to 0 or 1. When requesting tuner updates, the UCCP firmware:

- Snaps its frequency requests to the nearest frequency grid position.
- Filters out frequency requests that do not result in a change of gridded value.

The frequency grid control applies only when the host controlled tuner design model is in use. The grid control registers are ignored when local tuner control is in use (any "gridding" requirements are managed internally by the local tuner driver).

2.3.3. Tuner control model selection

Which tuner control model is in use is fixed and determined at build time. The end user need not be concerned with *how* the selection is made. The system designer just needs to know *which* model is in use. This section is, thus, for information only.

When the host controlled model is in use, the UCCP firmware is built with a stub driver in which most tuner control functions are omitted. Typically, the only active function in the driver is the AGC control setting function since, even in host controlled designs, AGC control is usually performed by the UCCP firmware.

It is the presence or absence of a "tune" function in the driver, which determines the behaviour of the UCCP firmware – if it is present, the local control model is used; if it is absent, the host controlled model is used.

Where several UCCP-based systems are made available to run on the same (host-controlled) platform, it is possible to build all with the same generic tuner driver.

2.3.4. Tuner register behaviour summary

Table 1 summarises the register behaviour for the two tuner control models.

Register	Local Tuner Control	Host Tuner Control
`TV_REG_TUNER_FREQ`	Host requests a tuner frequency.	Host notifies that it has changed tuner frequency.
`TV_REG_TUNER_BW`	Host requests a tuner bandwidth.	Host notifies that it has changed tuner bandwidth.
`TV_REG_ACTIVE_TUNER_FREQ`	UCCP firmware notifies host that it has changed tuner frequency.	UCCP firmware requests host to change tuner frequency.
`TV_REG_ACTIVE_TUNER_BW`	UCCP firmware notifies host that it has changed tuner bandwidth.	UCCP firmware requests host to change tuner bandwidth.

Register	Local Tuner Control	Host Tuner Control
TV_REG_TUNER_GRID_BASE	Ignored	Set by host to define tuner frequency grid.
TV_REG_TUNER_GRID_INCR	Ignored	Set by host to define tuner frequency grid.

Table 1 Tuner register behaviour summary

The TV_REG_TUNER_FREQ and TV_REG_TUNER_BW registers are linked such that the two values are always managed in pairs. Writing to TV_REG_TUNER_BW has no effect until a value is also written to TV_REG_TUNER_FREQ. Then, both values are processed by the UCCP firmware together. We recommend that the host always write both values, bandwidth first, even if only one has changed.

Similarly, the UCCP firmware will always write to both TV_REG_ACTIVE_TUNER_BW and TV_REG_ACTIVE_TUNER_FREQ (bandwidth first) even if only one of the values has changed. The host need only monitor TV_REG_ACTIVE_TUNER_FREQ to be notified of changes to either or both registers.

3. Register Set

3.1. Identification Registers

3.1.1. Common TV API Version

Address: TV_REG_API_VERSION (0)
Access: Read only

Bit(s)	Symbol	Description
31:16	Reserved	
15:8	TV_API_ID_MAJOR	API version ID, major part
7:0	TV_API_ID_MINOR	API version ID, minor part

3.1.2. Spectrum Analyser Firmware Version

Address: SA_REG_VERSION (1)
Access: Read only

Bit(s)	Symbol	Description
31:16	Reserved	
15:8	SA_VERSION_MAJOR	Spectrum Analyser version ID, major part
7:0	SA_VERSION_MINOR	Spectrum Analyser version ID, minor part

3.1.3.   Firmware Build ID

Address: SA_BUILD_ID (2)
Access: Read only

Bit(s)	Symbol	Description
31:0	SA_BUILD_ID	A unique identifier for a particular build of the firmware

3.2.   Basic System Control

System control is compatible with the control system for a TV demodulator, using similar commands and register addresses.

3.2.1.   System Control

Address: TV_REG_CONTROL (3)
Access: Read/write
Initial Value: 0xFFFFFFFF

The control register is the primary operating control for the Spectrum Analyser IP. Command values written to this register control the major operating modes of the demodulator.

This register is a command register as described in [1].

Symbol	Value	Description
TV_CMD_NULL	0	Null command – this command has no effect.
TV_CMD_STOP	1	Stop command – operations stop and the system returns to the "Initialised" state.
TV_CMD_DETECT	2	Detect command – the system enters the *DETECTING* state and starts a scan.

3.2.2.   System State

Address: SA_REG_STATE (4)
Access: Read/write
Initial Value: 0

The state register indicates the current operating state of the TV demodulator.

Symbol	Value	Description
SA_STATE_DORMANT	0	The *DORMANT* state is a pseudo-state, representing a system that is not available for use. This value does not normally appear in the state register since the register itself is not valid for a system in the DORMANT state.
SA_STATE_INITIALISED	1	The Spectrum Analyser is in the *INITIALISED* state – it is idle but ready for use.
SA_STATE_DETECTING	2	The Spectrum Analyser is in the *DETECTING* state, during which a scan is in progress.
SA_STATE_COMPLETED	3	The Spectrum Analyser is in the *COMPLETED* state, during which the scan results may be read by the controlling application.

3.3.   Tuner Control

Usually these registers are written while the system is in the *INITIALISED* state. Once a scan has been started, further updates to the tuner are initiated by the spectrum analyser IP itself. The precise meaning of these registers depends on whether the locally controlled tuner model or the host controlled tuner model is in use (Section 2.3).

3.3.1.   Tuner Frequency

Address: TV_REG_TUNER_FREQ (5)
Access: Read/write
Initial Value: 0xFFFFFFFF
In the *INITIALISED* state, this register is used to set the starting RF tuner frequency for a scan. For a host-controlled tuner, it is also used for confirmation that a requested frequency has been set, as described in section 2.3.2. It is a command register as described in [1].

Bit(s)	Symbol	Description
31:0	TV_REG_TUNER_FREQ	Tuner frequency in Hz

3.3.2.   Tuner Bandwidth

Address: TV_REG_TUNER_BW (6)
Access: Read/write
Initial Value: 0
In the *INITIALISED* state, this register is used to set the RF tuner bandwidth to use during a scan. For a host-controlled tuner, it is also used for confirmation that a requested bandwidth has been set, as described in section 2.3.2.

Its operation is linked to TV_REG_TUNER_FREQ in two ways:

• Updates to the tuner bandwidth will not become effective until TV_REG_TUNER_FREQ is written also; i.e. bandwidth changes and centre frequency changes are always made together.

• Completion of combined frequency and bandwidth update operations is indicated in TV_REG_TUNER_FREQ. You should check that TV_REG_TUNER_FREQ is ready for an update before writing to TV_REG_TUNER_BW. A bandwidth update may be missed if a frequency update (which implies a bandwidth update also) is still in progress.

Bit(s)	Symbol	Description
31:0	TV_REG_TUNER_BW	Tuner bandwidth in Hz

3.3.3.   Tuner Grid Base

Address: TV_REG_TUNER_GRID_BASE (7)
Access: Read/write
Initial Value: 0
This register is used (in conjunction with TV_REG_TUNER_GRID_INCR) to describe a RF frequency grid to the demodulator.

When a demodulator requests that the host makes a RF frequency change, it will limit such requests to frequencies satisfying: $F = Base \pm N \times Increment$, where N is an integer. This has two benefits:

• Frequency selections can be limited to those that the tuner is able to provide.

• Frequency selections are effectively filtered, limiting traffic across the register API.

Frequency gridding/filtering can be disabled by setting `TV_REG_TUNER_GRID_INCR` to 0 or 1.

Bit(s)	Symbol	Description
31:0	TV_REG_TUNER_GRID_BASE	Tuner RF frequency grid base in Hz

3.3.4. Tuner Grid Increment

Address: TV_REG_TUNER_GRID_INCR (8)
Access: Read/write
Initial Value: 0

This register is used (in conjunction with `TV_REG_TUNER_GRID_BASE`) to describe a RF frequency grid to the demodulator.

When a demodulator requests that the host makes a RF frequency change, it will limit such requests to frequencies satisfying: $F = Base \pm N \times Increment$, where N is an integer. This has two benefits:

- Frequency selections can be limited to those that the tuner is able to provide.
- Frequency selections are effectively filtered, limiting traffic across the register API.

Frequency gridding/filtering can be disabled by setting `TV_REG_TUNER_GRID_INCR` to 0 or 1.

Bit(s)	Symbol	Description
31:0	TV_REG_TUNER_GRID_BASE	Tuner RF frequency grid base in Hz

3.3.5. Active Tuner Frequency

Address: TV_REG_ACTIVE_TUNER_FREQ (11)
Access: Read/write
Initial Value: 0

For a host-controlled tuner this register is used to request changes of tuner frequency. In the case where the tuner is controlled directly by the spectrum analyser system, this register may be ignored.

Bit(s)	Symbol	Description
31:0	TV_REG_ACTIVE_TUNER_FREQ	Active tuner RF frequency in Hz

3.3.6. Active Tuner Bandwidth

Address: TV_REG_ACTIVE_TUNER_BW (12)
Access: Read/write
Initial Value: 0

For a host-controlled tuner this register is used to request changes of tuner bandwidth. In the case where the tuner is controlled directly by the spectrum analyser system, this register may be ignored.

Bit(s)	Symbol	Description
31:0	TV_REG_ACTIVE_TUNER_BW	Active tuner RF bandwidth in Hz

3.4.   Front End Override Registers

A number of internal SCP parameters must be matched to the chosen tuner and to the particular TV standards to be demodulated using the UCCP system.

- If the system implements local tuner control, these parameters are managed automatically and the controlling program need not be concerned with them.
- If the system implements a host-controlled tuner, the UCCP must be informed of appropriate SCP parameters. The front end override register group is used to load these parameters.

The entire group of front end override parameters must be updated as a consistent set. Writing to these registers has no effect on the TV demodulator until the `TV_REG_FE_GROUP` register is written, to define the particular parameter set to be updated.

These registers should only be updated in the *INITIALISED* state.

3.4.1.   Front End Group

Address: TV_REG_FE_GROUP (16)
Access: Read/write
Initial Value: 0xFFFFFFFF

The `TV_REG_FE_GROUP` register is used as a trigger to load the other `TV_REG_FE_xxx` registers into the demodulator. It also implements a handshake mechanism so that an application can avoid overwriting values in these registers before they have been accepted by the demodulator.

The spectrum analyser system only implements one group of front-end registers so 0 is the only valid value to write to this register.

This register is a command register as described in [1].

Bit(s)	Symbol	Description
31:0	TV_REG_FE_GROUP	Write 0 to update the spectrum analyser with the values in the other TV_REG_FE_XXX register values.

3.4.2.   Front End Control

Address: TV_REG_FE_CTRL (17)
Access: Read/write
Initial Value: 0

Bits	Description
31:10	Reserved
9	Inverted spectrum. 0: Normal spectrum 1: Inverted spectrum
8	Complex IF. 0: Real IF output from tuner. 1: Complex IF output from tuner.
7:1	Reserved
0	ADC sample format. 0: Signed 2's complement. 1: Offset binary

3.4.3.   ADC Sample Rate

Address: TV_REG_FE_ADCSAMP (18)
Access: Read/write
Initial Value: 0

Bits	Description
31:0	ADC Sample rate in sample/s

3.4.4.   IF Frequency

Address: TV_REG_FE_IF (19)
Access: Read/write
Initial Value: 0

Bits	Description
31:0	Tuner IF frequency in Hz

3.4.5.   AGC Sample Period

Address: TV_REG_FE_AGC_NORMAL (21)
Access: Read/write
Initial Value: 0

Bits	Description
31:18	Reserved
17:0	Normal AGC period (as a sample count)

3.4.6.   SCP FIR Filter Coefficients

Address: TV_REG_FE_FIR_NARROW_0 through TV_REG_FE_FIR_NARROW_03 (23-26)
Access: Read/write
Initial Value: 0

FIR filter coefficients in these registers take values in the range -512 to +511 representing coefficients in the range -2.0 to +1.9999.

Coefficients are packed sequentially, three to a register, with coefficients 0…2 in TV_REG_FE_FIR_NARROW_0, coefficients 3…5 in TV_REG_FE_FIR_NARROW_1 etc.

Bits	Description
31:30	Reserved
29:20	Coefficient C_{3X+2} sign extended to 10-bits.
19:10	Coefficient C_{3X+1} sign extended to 10-bits.
9:0	Coefficient C_{3X+0} sign extended to 10-bits.

3.5.   Scan Parameters

All of the register values in this section should be set up prior to running a scan, in the *INITIALISED* state.

3.5.1.   Scan Range

Address: SA_SCAN_RANGE (0x8000)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_SCAN_RANGE	Frequency range between start and end points of the scan, in Hz.

3.5.2.   Scan Resolution

Address: SA_SCAN_RESOLUTION (0x8001)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_SCAN_RESOLUTION	Frequency interval between spectral measurements, in Hz. This will be rounded to fit TV_REG_TUNER_GRID_INCR and give a whole number of readings per tuning step. The rounded value can be read after beginning a scan.

3.5.3.   Tuning Step

Address: SA_TUNING_STEP (0x8002)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31	SA_TUNING_STEP_AUTO	1: Spectrum analyser will autonomously choose appropriate tuning positions according to the tuner bandwidth and scan range settings 0: Spectrum analyser will tune with a spacing set by SA_TUNING_STEP. This may be desirable for example if the tuning positions are required to be located according to defined TV channel allocations.
30:0	SA_TUNING_STEP	Increment between tuning positions, in Hz, used when SA_TUNING_STEP_AUTO = 0. Note that this value must be less than 90% of the tuner bandwidth setting in TV_REG_TUNER_BW and will be rounded to the nearest TV_REG_TUNER_GRID_INCR setting.

3.5.4.   Measurement Control

Address: SA_MEASUREMENT_CONTROL (0x8003)
Access: Read/write
Initial Value: 0x00000200

Bit(s)	Symbol	Description
31:10	Reserved	

Bit(s)	Symbol	Description
9	SA_ENABLE_DC_COMP	Setting this bit to 1 enables a system for correcting for measurement error at DC. This is useful in the case of a zero-IF tuner, where the signal level at DC may be corrupted by residual DC in the analogue circuits prior to the ADCs.
8:6	SA_WINDOW	The window function applied to each analysis block before the Fast Fourier Transform is performed. The following options are available: 0: Rectangular 1: Hamming 2: Hanning Others: reserved
5:3	SA_SIGNAL_TYPE	The type of signal expected at the Spectrum Analyser input. This field is used to control the scaling of the signal through the measurement calculations. If the signal type is unknown then this field should be set to indicate NOISE. The following options are available: 0: NOISE 1: SINEWAVE Others: reserved
2:0	SA_MAX_PEAK_WIDTH	Defines a maximum width for detection of spectral peaks reported in the SA_MAX_POWER_REG_N registers. When a peak value is found, other peaks will be ignored in a region of frequencies around that peak. This region size is given by +/- (SA_MAX_PEAK_WIDTH * SA_SCAN_RESOLUTION). Note this register setting does not affect the scan results obtained from SA_REG_OUT_SPECTRUM_PTR.

3.5.5. Averaging Period

Address: SA_AVERAGING_PERIOD (0x8004)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:16	Reserved	
15:0	SA_AVERAGING_PERIOD_M	SA_AVERAGING_PERIOD_M is an integer and SA_AVERAGING_PERIOD_N is an enumerated type, such that the total number of blocks averaged for each spectral measurement is given by SA_AVERAGING_PERIOD_M * (2^(SA_AVERAGING_PERIOD_N+1)). A larger number of blocks averaged gives more noise immunity but extends the time taken for the scan.
7:0	SA_AVERAGING_PERIOD_N	

3.5.6. IF Gain Override

Address: SA_IF_GAIN_OVERRIDE (0x8005)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:17	Reserved	
16	SA_OVERRIDE_IF_GAIN	SA_OVERRIDE_IF_GAIN=1 causes the Spectrum Analyser application to use the value of IF gain specified in

Bit(s)	Symbol	Description
		the field SA_IF_GAIN. This feature can be used to ensure that the power spectral density measured at one tuner centre frequency can be compared directly with the power spectral density measured at a different tuner centre frequency. SA_OVERRIDE_IF_GAIN=0 causes the Spectrum Analyser application to use the value of IF gain calculated by the AGC system.
15:0	SA_IF_GAIN	IF gain value for use when SA_OVERRIDE_IF_GAIN=1. This gain value is restricted to positive values in the range in the range 0..65535, corresponding to the minimum and maximum gains, respectively, that the tuner can apply. The gain value is directly proportional to the logarithm of the expected gain to be applied to the incoming signal. I.e., the value is proportional to the gain in dB; however the translation to an absolute gain value is tuner-dependent. Note that a suitable setting can be obtained by reading SA_MIN_IF_GAIN after a scan has been performed.

3.6.   Tuner Characteristics

The following 4 registers allow the characteristics of the tuner channel filter to be communicated to the spectrum analyser system. This allows these characteristics to be allowed for when calculating the overall spectrum. The information may be obtained from the data sheet for the tuner.

3.6.1.   Tuner Filter 3dB point

Address: SA_TUNER_3DB_POINT (0x8006)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_TUNER_3DB_POINT	Frequency spacing between the centre of the passband of the tuner's channel filter and the 3dB attenuation frequency for that filter, in Hz

3.6.2.   Tuner Filter 6dB point

Address: SA_TUNER_6DB_POINT (0x8007)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_TUNER_6DB_POINT	Frequency spacing between the centre of the passband of the tuner's channel filter and the 6dB attenuation frequency for that filter, in Hz

3.6.3.   Tuner Filter 9dB point

Address: SA_TUNER_9DB_POINT (0x8008)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description

Bit(s)	Symbol	Description
31:0	SA_TUNER_9DB_POINT	Frequency spacing between the centre of the passband of the tuner's channel filter and the 9dB attenuation frequency for that filter, in Hz

3.6.4. Tuner Filter 12dB point

Address: SA_TUNER_12DB_POINT (0x8009)
Access: Read/write
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_TUNER_12DB_POINT	Frequency spacing between the centre of the passband of the tuner's channel filter and the 12dB attenuation frequency for that filter, in Hz

3.7. Data Output

The output spectrum is obtained by following a pointer supplied in the SA_REG_OUT_SPECTRUM_PTR register. The spectrum is stored as an array of 16-bit words. Each 16-bit word contains a signed value in Q7.8 format and represents spectral power in dB at a particular point on the frequency axis. The values are arranged in ascending frequency order with increasing address. The lowest frequency index (representing the frequency programmed into the TV_REG_TUNER_FREQ register) is located at the lowest address; and the frequency increment between each value is set by the SA_SCAN_RESOLUTION register. The number of values available is obtained by reading the SA_OUT_SPECTRUM_LEN register; it will correspond to the range set by the SA_SCAN_RANGE register, up to a maximum of 8192 values. A resolution and range combination that exceeds the maximum will trigger a failure. The SA_FAILURE_CODE register should be checked before proceeding.

The output spectrum values are only meaningful by reference against each other; the dB values have no absolute meaning. ~~0dB represents the lowest spectral power recorded in the scan~~.

However, if the tuner provides an RSSI (Received Signal Strength Indicator) measure, then this value is recorded when tuned to the highest-power frequency detected, and reported in the SA_MAX_RSSI_REG register. This gives the possibility of translating the detected dB value at this frequency to an absolute signal power value. If no RSSI value is available, then the SA_OUT_MIN_IF_GAIN_REG value can be used. This reports the analogue IF gain setting for the highest-power frequency detected, which can be used as a rough indicator of the absolute signal power.

3.7.1. Output Spectrum Pointer

Address: SA_REG_OUT_SPECTRUM_PTR (0x800A)
Access: Read only
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:30	SA_OUT_SPECTRUM_MEM_TYPE	Memory region within which the output spectrum may be located: 0: Global RAM 1: External RAM 2: Reserved 3: Reserved

Bit(s)	Symbol	Description
29:0	SA_OUT_SPECTRUM_PTR	Pointer to the start of the spectrum output, expressed as an offset in words from the start of the region specified by SA_OUT_SPECTRUM_MEM_TYPE. (Words are 24-bit for global RAM, 32-bit for other memory types)

3.7.2.   Output Spectrum Length

Address: SA_OUT_SPECTRUM_LEN (0x800B)
Access: Read only
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_OUT_SPECTRUM_LEN	Length of the output spectrum.

3.7.3.   Maximum RSSI

Address: SA_MAX_RSSI_REG (0x800C)
Access: Read only
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:16	SA_MAX_RSSI	Maximum RSSI value recorded from the tuner. This value will only be valid if the tuner provides an RSSI output. This is a signed 16-bit integer indicating an approximate signal strength value in dBm.
15:0	SA_MAX_RSSI_INDEX	Frequency index at the centre of the tuning position where the maximum RSSI value was recorded. This is an index into the output vector obtained from SA_OUT_SPECTRUM_PTR. A value of 0xFFFF indicates that the RSSI feature is unsupported.

3.7.4.   Reference IF Gain

Address: SA_REF_IF_GAIN_REG (0x800D)
Access: Read only
Initial Value: 0x00000000

Bit(s)	Symbol	Description
31:0	SA_REF_IF_GAIN	Reference analogue gain used by the IF AGC sub-system during the scan. This may be used as an indication of received signal power.
		This gain value is restricted to positive values in the range in the range 0..65535, corresponding to the minimum and maximum gains, respectively, that the tuner can apply. The gain value is directly proportional to the logarithm of the expected gain to be applied to the incoming signal. I.e., the value is proportional to the gain in dB; however the translation to an absolute gain value is tuner-dependent.

3.8. Scan Results Summary

These registers provide a simple analysis of the scan results for use by applications (such as Unicable signalling) which have no need to use the detailed scan results.

3.8.1. Spectral Power Peaks

Address: SA_MAX_POWER_REG_0 through SA_MAX_POWER_REG_7 (0x800E through 0x8015)
Access: Read only
Initial Value: 0x00000000

These registers record the 8 largest independent values of spectral power found in the scan. To ensure that each peak is independent, peaks which are adjacent to another higher peak are excluded. The "adjacent" region is defined by the SA_MAX_PEAK_WIDTH register. SA_MAX_POWER_REG_0 records the highest spectral power found, whereas SA_MAX_POWER_REG_7 records the 8th highest power found.

Bit(s)	Symbol	Description
31:24	Reserved	
23:16	SA_MAX_POWER_N	Spectral power peak value found in the scan, in dB
15:0	SA_MAX_POWER_INDEX_N	Frequency index where this spectral power peak was recorded. This is an index into the output vector obtained from SA_OUT_SPECTRUM_PTR.

3.8.2. Failure code

Address: SA_FAILURE_CODE (0x8016)
Access: Read only
Initial Value: 0x00000000

The failure code register should be read on commencing a scan to ensure settings are valid. Zero indicates no failure.

Bit(s)	Symbol	Description
31:1	Reserved	
0	SA_SCAN_SIZE_EXCEEDS_AVAILABLE_MEMORY	Not enough memory to perform scan with given resolution and range settings.

4. References

[1] Imagination Technologies "UCCP Common TV API Virtual Register Interface"