# STi5518

## SINGLE CHIP SET-TOP BOX DECODER WITH ENHANCED AUDIO

REGISTER MANUAL

# 1 Introduction

This manual describes the bit format and functionality of the STi5518 registers. *Register map* on page 5 lists all of the memory mapped registers, there is a table for each group of registers and the registers are sorted in address order within the tables. In the subsequent chapters, the registers are described individually in detail. The register map quotes the absolute register addresses, and the individual register descriptions give the address offset to the base address. The table below lists the base addresses for each group of registers. The final chapter contains an alphabetical index of registers. In the electronic version of this manual, the index and table of contents are hypertext linked into the register descriptions.

## 1.1 Accessing registers

Registers can be examined and set by the *devlw* (device load word) and *devsw* (device store word) instructions; they cannot be accessed using memory instructions. All registers located in region 2 of the peripheral address space. The registers of each module are grouped in the address space, usually in a 4 Kbyte block. In the register descriptions, the addresses are given as offsets from the base of the appropriate block. Table 1 lists the register block base variables and gives their values.

All unused register map locations and unused register bits are reserved, and only the value “0” must be written to any of these locations or bits. The values read from these locations or bits are undefined.

| Variable | Base address value | Block |
|---|---|---|
| ASC0BaseAddress | 0x20003000 | Asynchronous serial controller ASC0 / SmartCard 0 |
| ASC1BaseAddress | 0x20004000 | Asynchronous serial controller ASC1 |
| ASC2BaseAddress | 0x20005000 | Asynchronous serial controller ASC2 |
| ASC3BaseAddress | 0x20006000 | Asynchronous serial controller ASC3 / SmartCard 1 |
| AudioBaseAddress | 0x00000200 | Audio decoder |
| BMBaseAddress | 0x20026000 | Block move DMA controller |
| CacheBaseAddress | 0x00004000 | Cache configuration |
| Clock GeneratorBaseAddress | 0x00000100 | Clock Generator |
| DencBaseAddress | 0x00000600 | Digital Encoder |
| EMIBaseAddress | 0x00002000 | Programmable CPU Memory Interface |
| IntControllerBase | 0x20000000 | Interrupt controller |
| InterruptLevelBase | 0x20011000 | Interrupt level controller |
| IRBBaseAddress | 0x2000A200 | Infra red blaster |
| LinkBaseAddress | 0x20038000 | Link and front-end interface |
| LPCBaseAddress | 0x20000400 | Low power and Watchdog |
| ModemBaseAddress | 0x20027000 | Modem control |
| MPEGDMA0BaseAddress | 0x20024000 | MPEGDMA controller |
| MPEGDMA1BaseAddress | 0x20025000 | MPEGDMA controller |
| MPEGDMA2BaseAddress | 0x20030000 | LINK (SDAV MPEGDMA2) controller |
| PIO0BaseAddress | 0x2000C000 | PIO port 0 controller |
| PIO1BaseAddress | 0x2000D000 | PIO port 1 controller |
| PIO2BaseAddress | 0x2000E000 | PIO port 2 controller |
| PIO3BaseAddress | 0x2000F000 | PIO port 3 controller |
| PIO4BaseAddress | 0x20010000 | PIO port 4 controller |
| PIO5BaseAddress | 0x20001000 | PIO port 5 controller |
| PIO5BaseAddress | 0x2000A100 | PIO port 5 controller |
| PWMBaseAddress | 0x2000B000 | PWM and counter module |
| SSC0BaseAddress | 0x20009000 | Synchronous serial controller SSC0 |
| SSC1BaseAddress | 0x2000A000 | Synchronous serial controller SSC1 |

**Table 1 Register block base variables**

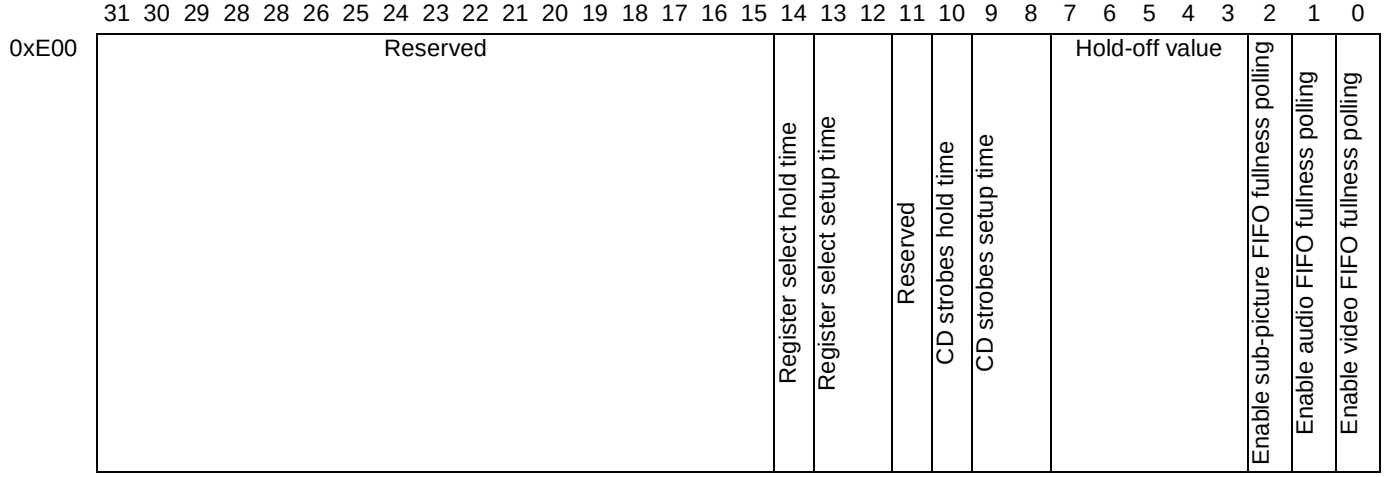
| Variable | Base address value | Block |
|---|---|---|
| SubPictureBaseAddress | 0x00000400 | Sub-picture decoder |
| TTxtBaseAddress | 0x2000A300 | Teletext |
| VideoBaseAddress | 0x00000000 | MPEG video decoder |

**Table 1 Register block base variables**

## 1.2 Audio/video access control

Audio and video access is controlled by the register MPEG_CONTROL, described here.

### MPEG_CONTROL MPEGAV buffer control register




Address: 0xE00
Type: R/W
Reset value: 0

***Description***

This register controls the speed of CPU accesses to the audio, video, sub-picture and DENC registers. It can also control the nature and speed of accesses to the CD FIFOs. For normal applications this register should be left in its default state. The register can be accessed by 8-bit word reads or word writes at the address 0xE00[7:0] and 0xE01[14:8].

- Bits 14:12 slow down register accesses by adding extra cycles on the internal register select signals.
- Bits 10:8 slow down CD FIFO accesses by adding extra cycles on the internal FIFO strobe signals.
- The 5-bit hold-off value enables the user to define a minimum wait time between two CD FIFO accesses. (The delay is in units of system clock cycles.)

| Bitfield | Description |
|---|---|
| Register select hold time | Extra cycle with register select high. |
| Register select setup time | Number of extra cycles with registers select low. |
| CD strobes hold time | Extra cycle with FIFO strobe high. |

| Bitfield | Description |
|---|---|
| CD strobes setup time | Number of extra cycles with FIFO strobe low. |
| Hold-off value | Minimum wait time before sending next CD FIFO access. |
| Enable sub-picture FIFO fullness polling | Enable FIFO fullness polling.<br>If the FIFO fullness polling enable bit is set, the CD FIFO fullness is examined on every byte transfer. If the FIFO is full then the transfer will be stalled. If the enable is not set, then data will be transferred regardless of the FIFO fullness. Hence there is a risk of FIFO overflow if the incoming bandwidth is too high. |
| Enable audio FIFO fullness polling | |
| Enable video FIFO fullness polling | |

## 1.3 Synchronization of video decoder registers

There are two types of video decoder register: *synchronized* and *unsynchronized*.

- *Synchronized* registers only change value in response to an internal event, either DSYNC or VSYNC, depending on the register. These registers are double-banked; during the write cycle the new value is loaded into a master register, and on the occurrence of the synchronizing event this value is loaded into a slave register, at which time the new value is available to the circuit. If a synchronized register is read, the value returned is that held in the master register.
- *Unsynchronized* registers change their value immediately they are written to.

Some registers are non synchronized registers with edge triggered write (on end of write cycle). This is to avoid glitches on internal signals during write cycles.

# 2 Register map

This chapter lists all the memory mapped registers in grouped into modules and in address order. The contents of each register are given in the individual register descriptions, and the addresses are absolute addresses. The column headed **Bits** gives the number of significant bits. For serial registers, only 8 bits should be read or written at each access.

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| ASC_0_BAUDRATE | 0x20003000 | 16 | R/W | ASC 0 Baud rate generator/reload |
| ASC_0_TXBUFFER | 0x20003004 | 9 | W | ASC 0 Output buffer |
| ASC_0_RXBUFFER | 0x20003008 | 9 | R | ASC 0 Input buffer |
| ASC_0_CONTROL | 0x2000300C | 13 | R/W | ASC 0 Control register |
| ASC_0_INTENABLE | 0x20003010 | 9 | R/W | ASC 0 Enable interrupts |
| ASC_0_STATUS | 0x20003014 | 11 | R | ASC 0 Interrupt status |
| ASC_0_GUARDTIME | 0x20003018 | 8 | R/W | ASC 0 Delay before transmitter empty flag |
| ASC_0_TIMEOUT | 0x2000301C | 8 | R/W | ASC 0 Time out register |
| ASC_0_TXRESET | 0x20003020 | 0 | W | ASC 0 Reset output FIFO |
| ASC_0_RXRESET | 0x20003024 | 0 | W | ASC 0 Reset input FIFO |
| ASC_0_RETRIES | 0x20003028 | 8 | R/W | ASC 0 number of retries on transmission |

**Table 2 Asynchronous Serial Controller 0 (ASC)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| ASC_1_BAUDRATE | 0x20004000 | 16 | R/W | ASC 1 Baud rate generator/reload |
| ASC_1_TXBUFFER | 0x20004004 | 9 | W | ASC 1 Output buffer |
| ASC_1_RXBUFFER | 0x20004008 | 9 | R | ASC 1 Input buffer |
| ASC_1_CONTROL | 0x2000400C | 13 | R/W | ASC 1 Control register |
| ASC_1_INTENABLE | 0x20004010 | 9 | R/W | ASC 1 Enable interrupts |
| ASC_1_STATUS | 0x20004014 | 11 | R | ASC 1 Interrupt status |
| ASC_1_GUARDTIME | 0x20004018 | 8 | R/W | ASC 1 Delay before transmitter empty flag |
| ASC_1_TIMEOUT | 0x2000401C | 8 | R/W | ASC 1 Time out register |
| ASC_1_TXRESET | 0x20004020 | 0 | W | ASC 1 Reset output FIFO |
| ASC_1_RXRESET | 0x20004024 | 0 | W | ASC 1 Reset input FIFO |
| ASC_1_RETRIES | 0x20004028 | 8 | R/W | ASC 1 number of retries on transmission |

**Table 3 Asynchronous Serial Controller 1 (ASC)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| ASC_2_BaudRate | 0x20005000 | 16 | R/W | ASC 2 Baud rate generator/reload |
| ASC_2_TxBuffer | 0x20005004 | 9 | W | ASC 2 Output buffer |
| ASC_2_RxBuffer | 0x20005008 | 9 | R | ASC 2 Input buffer |
| ASC_2_Control | 0x2000500C | 13 | R/W | ASC 2 Control register |
| ASC_2_IntEnable | 0x20005010 | 9 | R/W | ASC 2 Enable interrupts |

**Table 4 Asynchronous Serial Controller 2 (ASC)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| ASC_2_Status | 0x20005014 | 11 | R | ASC 2 Interrupt status |
| ASC_2_Guardtime | 0x20005018 | 8 | R/W | ASC 2 Delay before transmitter empty flag |
| ASC_2_Timeout | 0x2000501C | 8 | R/W | ASC 2 Time out register |
| ASC_2_TxReset | 0x20005020 | 0 | W | ASC 2 Reset output FIFO |
| ASC_2_RxReset | 0x20005024 | 0 | W | ASC 2 Reset input FIFO |
| ASC_2_Retries | 0x20005028 | 8 | R/W | ASC 2 number of retries on transmission |

**Table 4 Asynchronous Serial Controller 2 (ASC)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| ASC_3_BaudRate | 0x20006000 | 16 | R/W | ASC 3 Baud rate generator/reload |
| ASC_3_TxBuffer | 0x20006004 | 9 | W | ASC 3 Output buffer |
| ASC_3_RxBuffer | 0x20006008 | 9 | R | ASC 3 Input buffer |
| ASC_3_Control | 0x2000600C | 13 | R/W | ASC 3 Control register |
| ASC_3_IntEnable | 0x20006010 | 9 | R/W | ASC 3 Enable interrupts |
| ASC_3_Status | 0x20006014 | 10 | R | ASC 3 Interrupt status |
| ASC_3_Guardtime | 0x20006018 | 8 | R/W | ASC 3 Delay before transmitter empty flag |
| ASC_3_Timeout | 0x2000601C | 8 | R/W | ASC 3 Time out register |
| ASC_3_TxReset | 0x20006020 | 0 | W | ASC 3 Reset output FIFO |
| ASC_3_RxReset | 0x20006024 | 0 | W | ASC 3 Reset input FIFO |
| ASC_3_Retries | 0x20006028 | 8 | R/W | ASC 3 number of retries on transmission |

**Table 5 Asynchronous Serial Controller 3 (ASC)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| AUD_VERSION | 0x00000200 | 8 | RO | Hardware version |
| AUD_IDENT | 0x00000201 | 8 | RO | Identify |
| AUD_SFREQ | 0x00000205 | 8 | R/WS | Sampling frequency |
| AUD_EMPH | 0x00000206 | 2 | R/W | Emphasis |
| AUD_INTEL [7:0] | 0x00000207 | 16 | R/W | Interrupt enable |
| AUD_INTE [15:8] | 0x00000208 | 16 | R/W | Interrupt enable |
| AUD_INT [7:0] | 0x00000209 | 16 | R/W | Interrupt |
| AUD_INT [15:8] | 0x0000020A | 16 | R/W | Interrupt |
| AUD_SIN_SETUP | 0x0000020C | 8 | R/W | Input data setup |
| AUD_CAN_SETUP | 0x0000020D | 8 | R/W | A/D converter setup |
| AUD_ERROR | 0x0000020F | 8 | RO | ERROR code |
| AUD_SOFTRESET | 0x00000210 | 8 | W | Soft reset |
| AUD_PLLPCM | 0x00000212 | 3 | R/W | PCM PLL disable |
| AUD_PLAY | 0x00000213 | 1 | R/W | Play |
| AUD_MUTE | 0x00000214 | 1 | R/W | Mute |
| AUD_PLLMASK | 0x00000218 | 1 | W | PCMCLK mask for half sampling frequency |

**Table 6 Audio Decoder (AUD)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| AUD_SYNCSTATUS | 0x00000240 | 4 | RO | Synchronization status |
| AUD_ANCCOUNT | 0x00000241 | 8 | RO | Ancillary data |
| AUD_HEAD[31:24] | 0x00000242 | 8 | RO | HEADER 4 register |
| AUD_HEAD[23:16] | 0x00000243 | 8 | RO | HEADER 3 register |
| AUD_HEADLEN[15:8] | 0x00000244 | 8 | RO | Frame length |
| AUD_HEADLEN[7:0] | 0x00000245 | 8 | RO | Frame length |
| AUD_PTS[32] | 0x00000246 | 8 | R/W | PTS |
| AUD_PTS[31:24] | 0x00000247 | 8 | R/W | PTS |
| AUD_PTS[23:16] | 0x00000248 | 8 | R/W | PTS |
| AUD_PTS[15:8] | 0x00000249 | 8 | R/W | PTS |
| AUD_PTS[7:0] | 0x0000024A | 8 | R/W | PTS |
| AUD_STREAMSEL | 0x0000024C | 3 | R/W | STREAM selection |
| AUD_DECODESEL | 0x0000024D | 5 | R/W | Decoding algorithm |
| AUD_VOLUME0 | 0x0000024E | 8 | R/W | Left/right balance |
| AUD_PACKET_LOCK | 0x0000024F | 8 | R/W | Packet lock |
| AUD_AUDIO_ID_EN | 0x00000250 | 8 | R/W | Enable audio ID |
| AUD_AUDIO_ID | 0x00000251 | 8 | R/W | Audio ID |
| AUD_AUDIO_ID_EXT | 0x00000252 | 8 | R/W | Audio extension |
| AUD_SYNC_LOCK | 0x00000253 | 8 | R/W | SYNC lock |
| AUD_PCMDIVIDER | 0x00000254 | 8 | R/W | Divider for PCM clock |
| AUD_PCM_CONF | 0x00000255 | 7 | R/W | PCM configuration |
| AUD_PCM_CROSS | 0x00000256 | 6 | R/W | Cross PCM channels |
| AUD_LDLY | 0x00000257 | 8 | R/W | Left channel |
| AUD_TM_SPEED | 0x00000257 | 8 | R/W | Audio trick-mode speed |
| AUD_RDLY | 0x00000258 | 8 | R/W | Right channel |
| AUD_CDLY | 0x00000259 | 8 | R/W | Centre channel |
| AUD_SUBDLY | 0x0000025A | 8 | R/W | Subwoofer channel |
| AUD_LSDLY | 0x0000025B | 8 | R/W | Left surround channel |
| AUD_RSDLY | 0x0000025C | 8 | R/W | Right surround channel |
| AUD_DLYUPDATE | 0x0000025D | 8 | R/W | PCM delay update |
| AUD_SPDIF_CMD | 0x0000025E | 8 | R/W | IEC958 control |
| AUD_SPDIF_CAT | 0x0000025F | 8 | R/W | Category code |
| AUD_SPDIF_CONF | 0x00000260 | 8 | R/W | IEC958 PCMCLK divider |
| AUD_SPDIF_STATUS | 0x00000261 | 7 | R/W | IEC status bit |
| AUD_PDEC | 0x00000262 | 3 | R/W | Post decoder register |
| AUD_VOLUME1 | 0x00000263 | 8 | R/W | L/R surround balance |
| AUD_PL_AB | 0x00000264 | 8 | R/W | Pro logic auto balance |
| AUD_PL_DWNX | 0x00000265 | 8 | R/W | Pro logic decoder downmix |
| AUD_OCFG | 0x00000266 | 4 | R/W | Output configuration |
| AUD_CHAN_IDX | 0x00000267 | 3 | R/W | PCM scale factor |
| AUD_PCM_BTONE | 0x00000268 | 8 | R/W | PCM beep tone frequency |
| AUD_DECODE_LFE | 0x00000268 | 8 | R/W | Decode LFE for AC3 setup |

**Table 6 Audio Decoder (AUD)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| AUD_MP_SKIP_LFE | 0x00000268 | 8 | R/W | Channel skip for MPEG setup |
| AUD_COMP_MOD | 0x00000269 | 8 | R/W | Compression mode for AC3 setup |
| AUD_MP_PROG_NO | 0x00000269 | 8 | R/W | Program number for MPEG setup |
| AUD_AC3_HDR | 0x0000026A | 8 | R/W | High dynamic range for AC3 setup |
| AUD_MP_DRC | 0x0000026A | 1 | R/W | High dynamic range for MPEG setup |
| AUD_AC3_LDR | 0x0000026B | 8 | R/W | Low dynamic range |
| AUD_AC3_CRC_OFF | 0x0000026C | 8 | R/W | Repeat count for AC3 setup |
| AUD_MP_CRC_OFF | 0x0000026C | 8 | R/W | CRC check-off for MPEG setup |
| AUD_AC3_KARAMODE | 0x0000026D | 8 | R/W | Karaoke downmix for AC3 setup |
| AUD_MP_MC_OFF | 0x0000026D | 2 | R/W | Multichannel for MPEG setup |
| AUD_AC3_DUALMODE | 0x0000026E | 8 | R/W | Dual downmix for AC3 setup |
| AUD_MP_DUAL | 0x0000026E | 8 | R/W | Dual downmix for MPEG setup |
| AUD_AC3_DOWNMIX | 0x0000026F | 8 | R/W | Downmix for AC3 setup |
| AUD_MP_DOWNMIX | 0x0000026F | 6 | R/W | Downmix for MPEG setup |
| AUD_DWSMODE | 0x00000270 | 8 | R/W | Downsampling filter |
| AUD_SOFTVER | 0x00000271 | 8 | R/W | Software version |
| AUD_RUN | 0x00000272 | 1 | R/W | RUN decoding |
| AUD_SKIP_MUTE_CMD | 0x00000273 | 5 | R/W | Skip or mute commands |
| AUD_SKIP_MUTE_VALUE | 0x00000274 | 8 | R/W | Skip frames or mutes blocks of frame |
| AUD_SPDIF_REP_TIME | 0x00000275 | 8 | R/W | IEC958 repetition time of a pause frame |
| AUD_STATUS0 | 0x00000276 | 8 | RO | AC-3 status register |
| AUD_MP_STATUS0 | 0x00000276 | 8 | RO | MPEG status register0 |
| AUD_LPCM_STATUS0 | 0x00000276 | 7 | RW | LPMC status register0 |
| AUD_STATUS1 | 0x00000277 | 4 | RO | AC-3 status register 1 |
| AUD_MP_STATUS1 | 0x00000277 | 4 | RO | MPEG status register1 |
| AUD_LPCM_STATUS1 | 0x00000277 | 7 | RW | LPMC status register1 |
| AUD_STATUS2 | 0x00000278 | 8 | RO | AC-3 status register 2 |
| AUD_MP_STATUS2 | 0x00000278 | 4 | RO | MPEG status register2 |
| AUD_LPCM_STATUS2 | 0x00000278 | 8 | RW | LPMC status register3 |
| AUD_STATUS3 | 0x00000279 | 4 | RO | AC-3 status register 3 |
| AUD_MP_STATUS3 | 0x00000279 | 8 | RO | MPEG status register3 |
| AUD_STATUS4 | 0x0000027A | 5 | RO | AC-3 status register 4 |
| AUD_MP_STATUS4 | 0x0000027A | 8 | RO | MPEG status register4 |
| AUD_STATUS5 | 0x0000027B | 8 | RO | AC-3 status register 5 |
| AUD_MP_STATUS5 | 0x0000027B | 8 | RO | MPEG status register5 |
| AUD_STATUS6 | 0x0000027C | 5 | RO | AC-3 status register 6 |
| AUD_STATUS7 | 0x0000027D | 8 | RO | AC-3 status register 7 |
| AUD_SPDIF_LATENCY | 0x0000027E | 8 | R/W | Latency value |
| AUD_SPDIF_DTDI | 0x0000027F | 8 | R/W | IEC958 data type information |

**Table 6 Audio Decoder (AUD)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| BMDMA_SRCADDRESS | 0x20026000 | 32 | W | Block move DMA source address |
| BMDMA_DESTADDRESS | 0x20026004 | 32 | W | Block move DMA destination address |
| BMDMA_COUNT | 0x20026008 | 16 | W | Block move DMA counts |
| BMDMA_INTEN | 0x2002600C | 1 | R/W | Block move DMA interrupt enable |
| BMDMA_STATUS | 0x20026010 | 2 | R | Block move DMA status |
| BMDMA_INTACK | 0x20026014 | 32 | W | Block move DMA interrupt acknowledge |
| BMDMA_ABORT | 0x20026018 | 32 | W | Block move DMA abort |

**Table 7 Block move DMA (BMDMA)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| CAC_CACHECONTROL0 | 0x00004000 | 8 | R/W | Cacheability of 0xC0000000 to 0xC007FFFF |
| CAC_CACHECONTROL1 | 0x00004100 | 8 | R/W | Cacheability of 0xC0200000 to 0xC027FFFF |
| CAC_CACHECONTROL2 | 0x00004200 | 8 | R/W | Cacheability of 0x40000000 to 0x4007FFFF |
| CAC_CACHECONTROL3 | 0x00004300 | 4 | R/W | Cacheability of 0x40000000 to 0x7FFFFFFF |
| CAC_DCACHENOTSRAM | 0x00004400 | 1 | W | Select data cache or extra SRAM |
| CAC_INVALIDATEDCACHE | 0x00004500 | 1 | W | Invalidate the data cache |
| CAC_FLUSHDCACHE | 0x00004600 | 1 | W | Flush the data cache |
| CAC_ENABLEICACHE | 0x00004700 | 1 | W | Enable the instruction cache |
| CAC_INVALIDATEICACHE | 0x00004800 | 1 | W | Invalidate the instruction cache |
| CAC_CACHESTATUS | 0x00004900 | 7 | R | Cache status |
| CAC_CACHECONTROLLOCK | 0x00004A00 | 1 | R/W | Lock the cache configuration |

**Table 8 Cache Control (CAC)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| CKG_LPC_DIV | 0x000001CE | 8 | R/W | Low-power clock divider |
| CKG_PLL_CNT | 0x000001CF | 6 | R/W | Clock bypass PLL phase and PLL controller |
| CKG_CCAUD | 0x000001D0 | 3 | R/W | Clock controller for audio clock |
| CKG_DIVAUD | 0x000001D1 | 4 | R/W | Clock divider for audio clock |
| CKG_CCAUXDENC | 0x000001D2 | 3 | R/W | Auxiliary clock controller |
| CKG_DIVAUXDENC | 0x000001D3 | 4 | R/W | Denc Test clock divider |
| CKG_CCMCK | 0x000001D4 | 3 | R/W | Clock controller for MEMCLK |
| CKG_DIVMCK | 0x000001D5 | 4 | R/W | Clock divider for MEMCLK |
| CKG_SFREQSMC_SDIV | 0x000001D6 | 3 | R/W | SDIV reg: SMC frequency synthesizer |
| CKG_SFREQSMC_PE0 | 0x000001D7 | 8 | R/W | PE reg (LSB): SMC frequency synthesizer |
| CKG_SFREQSMC_PE1 | 0x000001D8 | 8 | R/W | PE reg (MSB): SMC frequency synthesizer |
| CKG_SFREQSMC_MD | 0x000001D9 | 5 | R/W | MD reg: SMC frequency synthesizer |
| CKG_SMC_CNT | 0x000001DA | 2 | R/W | Smart card clock control |

**Table 9 Clock generator (CKG)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| CKG_CCDENC | 0x000001DC | 3 | R/W | Clock Control for the DENC Clock |
| CKG_CCST20 | 0x000001DD | 3 | R/W | Clock controller for ST20 clock |
| CKG_DIVST20 | 0x000001DE | 4 | R/W | Clock divider for ST20 clock |
| CKG_PREDIVPLL | 0x000001DF | 8 | R/W | Pre-divider ratio set-up of the PLL |
| CKG_FBKDIVPLL | 0x000001E0 | 8 | R/W | Feedback divider ratio set-up of the PLL |
| CKG_POSTDIVPLL | 0x000001E1 | 6 | R/W | Post divider ratio and lock detector |
| CKG_PLLSETUP | 0x000001E2 | 6 | R/W | PLL start-up control and charge pump control |
| CKG_IDDQPAD_C | 0x000001E3 | 4 | R/W | Iddq mode, I/O hsync, vsync |
| CKG_SFREQAUD_SDIV | 0x000001E4 | 3 | R/W | SDIV reg: AUD frequency synthesizer |
| CKG_SFREQAUD_PE0 | 0x000001E5 | 8 | R/W | PE reg (LSB): AUD frequency synthesizer |
| CKG_SFREQAUD_PE1 | 0x000001E6 | 8 | R/W | PE reg (MSB): AUD frequency synthesizer |
| CKG_SFREQAUD_MD | 0x000001E7 | 5 | R/W | MD reg: AUD frequency synthesizer |
| CKG_AUD_CNT | 0x000001E8 | 1 | R/W | Sets audio frequency control through ST20 or MMDSP |
| CKG_SFREQAUX_SDIV | 0x000001E9 | 3 | R/W | SDIV reg: AUX frequency synthesizer |
| CKG_SFREQAUX_PE0 | 0x000001EA | 8 | R/W | PE reg (LSB): AUX frequency synthesizer |
| CKG_SFREQAUX_PE1 | 0x000001EB | 8 | R/W | PE reg (MSB): AUX frequency synthesizer |
| CKG_SFREQAUX_MD | 0x000001EC | 5 | R/W | MD reg: AUX frequency synthesizer |
| CKG_AUX_CNT | 0x000001ED | 1 | R/W | Enable AUX frequency synthesizer |

**Table 9 Clock generator (CKG)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| DEN_CFG0 | 0x00000600 | 8 | R/W | DENC configuration 0 |
| DEN_CFG1 | 0x00000601 | 8 | R/W | DENC configuration 1 |
| DEN_CFG2 | 0x00000602 | 8 | R/W | DENC configuration 2 |
| DEN_CFG3 | 0x00000603 | 8 | R/W | DENC configuration 3 |
| DEN_CFG4 | 0x00000604 | 8 | R/W | DENC configuration 4 |
| DEN_CFG5 | 0x00000605 | 7 | R/W | DENC configuration 5 |
| DEN_CFG6 | 0x00000606 | 8 | R/W | DENC configuration 6 |
| DEN_CFG7 | 0x00000607 | 8 | R/W | DENC configuration 7 |
| DEN_CFG8 | 0x00000608 | 8 | R/W | DENC configuration 8 |
| DEN_STA | 0x00000609 | 8 | R | DENC Status |
| DEN_IDFS1 | 0x0000060A | 8 | R/W | DENC Increment for Digital Frequency Synthesizer |
| DEN_IDFS2 | 0x0000060B | 8 | R/W | DENC Increment for Digital Frequency Synthesizer |
| DEN_IDFS3 | 0x0000060C | 8 | R/W | DENC Increment for Digital Frequency Synthesizer |
| DEN_PDFS1 | 0x0000060D | 8 | R/W | Static phase offset for digital frequency synthesizer |
| DEN_PDFS2 | 0x0000060E | 8 | R/W | Static phase offset for digital frequency synthesizer |
| DEN_WSS1 | 0x0000060F | 8 | R/W | WSS data registers |
| DEN_WSS2 | 0x00000610 | 8 | R/W | WSS data registers |
| DEN_DAC13 | 0x00000611 | 8 | R/W | DAC1 and DAC3 multiplying factors |
| DEN_DAC45 | 0x00000612 | 8 | R/W | DAC4 and DAC5 multiplying factors |

**Table 10 Digital encoder (DEN)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| DEN_DAC6C | 0x00000613 | 8 | R/W | DAC6 and C multiplying factors |
| DEN_LJMP1 | 0x00000615 | 8 | R/W | Line jump |
| DEN_LJMP2 | 0x00000616 | 8 | R/W | Line jump |
| DEN_LJMP3 | 0x00000617 | 8 | R/W | Line jump |
| DEN_CID | 0x00000618 | 8 | R | DENC identification number |
| DEN_VPS1 | 0x00000619 | 8 | R/W | VPS data registers |
| DEN_VPS2 | 0x0000061A | 8 | R/W | VPS data registers |
| DEN_VPS3 | 0x0000061B | 8 | R/W | VPS data registers |
| DEN_VPS4 | 0x0000061C | 8 | R/W | VPS data registers |
| DEN_VPS5 | 0x0000061D | 8 | R/W | VPS data registers |
| DEN_VPS6 | 0x0000061E | 8 | R/W | VPS data registers |
| DEN_CGMS1 | 0x0000061F | 4 | R/W | CGMS data registers |
| DEN_CGMS2 | 0x00000620 | 8 | R/W | CGMS data registers |
| DEN_CGMS3 | 0x00000621 | 8 | R/W | CGMS data registers |
| DEN_TTX1 | 0x00000622 | 8 | R/W | Teletext block definition |
| DEN_TTX2 | 0x00000623 | 8 | R/W | Teletext block definition |
| DEN_TTX3 | 0x00000624 | 8 | R/W | Teletext block definition |
| DEN_TTX4 | 0x00000625 | 8 | R/W | Teletext block definition |
| DEN_TTXM | 0x00000626 | 8 | R/W | Teletext block mapping |
| DEN_CCF1 | 0x00000627 | 8 | R/W | Closed caption characters/extended data for field 1 |
| | 0x00000628 | 8 | R/W | Closed caption characters/extended data for field 1 |
| DEN_CCF2 | 0x00000629 | 8 | R/W | Closed caption characters/extended data for field 2 |
| | 0x0000062A | 8 | R/W | Closed caption characters/extended data for field 2 |
| DEN_CLF1 | 0x0000062B | 5 | R/W | Closed caption/extended data line insert for field 1 |
| DEN_CLF2 | 0x0000062C | 5 | R/W | Closed caption/extended data line insert for field 2 |
| DEN_REG_45...63 | 0x0000062D-3F | | | RESERVED |
| DEN_REG_64 | 0x00000640 | 3 | R/W | TTX_conf |
| DEN_REG_65 | 0x00000641 | 8 | R/W | DAC2MULT&TTXS |
| DEN_REG_65...68 | 0x00000642-4 | | | RESERVED |
| DEN_REG_69 | 0x00000645 | 8 | R/W | Brightness |
| DEN_REG_70 | 0x00000646 | 8 | R/W | Contrast |
| DEN_REG_71 | 0x00000647 | 8 | R/W | Saturation |
| DEN_YCOUT | 0x000001F9 | 1 | R/W | Make YC data bus accessible on PIO4 |

**Table 10 Digital encoder (DEN)**

**Programmable CPU Memory Interface (EMI)**

| REGISTER | Address | Size | Read/ Write | Description |
|---|---|---|---|---|
| EMI_CONFIGDATA0BANK0 | 0x00002000 | 16 | R/W | EMI bank 0 configuration data register 0<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA1BANK0 | 0x00002004 | 16 | R/W | EMI bank 0 configuration data register 1<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA2BANK0 | 0x00002008 | 16 | R/W | EMI bank 0 configuration data register 2<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA3BANK0 | 0x0000200C | 16 | R/W | EMI bank 0 configuration data register 3<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA0BANK1 | 0x00002010 | 16 | R/W | EMI bank 1 configuration data register 0<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA1BANK1 | 0x00002014 | 16 | R/W | EMI bank 1 configuration data register 1<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA2BANK1 | 0x00002018 | 16 | R/W | EMI bank 1 configuration data register 2<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA3BANK1 | 0x0000201C | 16 | R/W | EMI bank 1 configuration data register 3<br>SDRAM, DRAM and peripheral memory formats |
| EMI_CONFIGDATA0BANK2 | 0x00002020 | 16 | R/W | EMI bank 2 configuration data register 0<br>peripheral memory format |
| EMI_CONFIGDATA1BANK2 | 0x00002024 | 16 | R/W | EMI bank 2 configuration data register 1<br>peripheral memory format |
| EMI_CONFIGDATA2BANK2 | 0x00002028 | 16 | R/W | EMI bank 2 configuration data register 2<br>peripheral memory format |
| EMI_CONFIGDATA3BANK2 | 0x0000202C | 16 | R/W | EMI bank 2 configuration data register 3<br>peripheral memory format |
| EMI_CONFIGDATA0BANK3 | 0x00002030 | 16 | R/W | EMI bank 3 configuration data register 0<br>peripheral memory format |
| EMI_CONFIGDATA1BANK3 | 0x00002034 | 16 | R/W | EMI bank 3 configuration data register 1<br>peripheral memory format |
| EMI_CONFIGDATA2BANK3 | 0x00002038 | 16 | R/W | EMI bank 3 configuration data register 2<br>peripheral memory format |
| EMI_CONFIGDATA3BANK3 | 0x0000203C | 16 | R/W | EMI bank 3 configuration data register 3<br>peripheral memory format |
| EMI_CONFIGLOCKBANK0 | 0x00002040 | 1 | W | Write protection bit. When set, makes **EMI_ConfigData0-3** for **Bank0** read only. |
| EMI_CONFIGLOCKBANK1 | 0x00002044 | 1 | W | Write protection bit. When set, makes **EMI_ConfigData0-3** for **Bank1** read only. |
| EMI_CONFIGLOCKBANK2 | 0x00002048 | 1 | W | Write protection bit. When set, makes **EMI_ConfigData0-3** for **Bank2** read only. |
| EMI_CONFIGLOCKBANK3 | 0x0000204C | 1 | W | Write protection bit. When set, makes **EMI_ConfigData0-3** for **Bank3** read only. |
| EMI_CONFIGSTATUS | 0x00002050 | 8 | R | Status information |
| EMI_SDRAMMODEREG0 | 0x00002058 | 7 | W | SDRAM Mode Register for either the case of one SDRAM bank or SDRAM in bank0, when two SDRAM banks. |
| EMI_SDRAMMODEREG1 | 0x0000205C | 7 | W | SDRAM Mode Register in the case of SDRAM in bank1 only when two SDRAM banks. |

**Table 11 Programmable CPU Memory Interface (EMI)**

| REGISTER | Address | Size | Read/ Write | Description |
|---|---|---|---|---|
| EMI_DRAMINITIALIZE | 0x00002060 | 1 | W | Initialize any (S)DRAM in the system.<br>In Slave Mode, the EMI3 should not initialize DRAM or SDRAM devices in the system, nor should it refresh any S/DRAM device. Therefore this register should never be written |
| EMI_CONFIGPADLOGIC | 0x00002070 | 14 | R/W | Padlogic configuration data register. |

**Table 11 Programmable CPU Memory Interface (EMI)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| FEI_IK | 0x20038000 + 00, 01, 02, 03, 04 | 8 | W & R [1] | Intermediate Key |
| FEI_TWH | 0x20038000, + 10, 11, 12, 13, 14 | 8 | W0 | Temporary Word High |
| FEI_TWL | 0x20038000, + 18, 19, 1A, 1B, 1C | 8 | W0 | Temporary Word Low |
| FEI_IKL | 0x20038007 | 8 | W0 | Internal Key Load Register |
| FEI_IF | 0x2003803F | 6 | W0 | Internal Function |
| FEI_DEC_DAT | 0x20038780 | 8 | R/W | |
| FEI_DEC_IDX | 0x20038784 | 6 | WO | |
| FEI_DEC_STA | 0x20038784 | 2 | RO | |
| FEI_SUB | 0x20038788 | 16 | RO | |
| FEI_SFF | 0x2003878C | 8 | RO | |
| FEI_GCF | 0x20038790 | 16 | RO | |
| FEI_SLG | 0x20038798 | 16 | R/W | |
| FEI_REV | 0x2003879C | 8 | RO | Revision ID |
| FEI_ATAPI_CFG | 0x200387A0 | 8 | R/W | ATAPI interface configuration |
| FEI_SE_RST | 0x20038800 | 1 | WO | Reset Module |
| FEI_SE_SAC | 0x20038804 | 1 | R/W | Sector Address Check |
| FEI_SE_MOD | 0x20038808 | 1 | R/W | Sector Processor Mode |
| FEI_SE_CAPEN | 0x2003880C | 1 | R/W | Capturing Enable |
| FEI_SE_EMR | 0x20038810 | 1 | R/W | Error Mode |
| FEI_SE_IR | 0x20038814 | 8 | R/W | Interrupt |
| FEI_SE_STAT | 0x20038818 | 1 | RO | Status |
| FEI_SE_CAP_H | 0x20038820 | 8 | R/W | Address of first sector /subcode block to be captured |
| FEI_SE_CAP_L | 0x20038824 | 8 | R/W | Address of first sector /subcode block to be captured |
| FEI_SE_LST_H | 0x20038828 | 8 | R/W | Address of (last + 1) sector /subcode block to be captured |
| FEI_SE_LST_L | 0x2003882C | 16 | R/W | Address of (last + 1) sector/subcode block to be captured |
| FEI_SE_CUR_H | 0x20038830 | 8 | RO | Address of sector/subcode block actually entering the SP |
| FEI_SE_CUR_L | 0x20038834 | 16 | RO | Address of sector /subcode block entering the SP |
| FEI_SE_HE_H | 0x20038860 | 16 | R/W | CPR_MAI Bits of Current Sector actually entering the SP |
| FEI_SE_HE_M | 0x20038864 | 16 | R/W | CPR_MAI Bits of Current Sector actually entering the SP |
| FEI_SE_HE_L | 0x20038868 | 16 | R/W | CPR_MAI Bits of Current Sector actually entering the SP |

**Table 12 Front-end interface (FEI)**

1. Read access in some conditions

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| INC_HANDLERWPTR0 | 0x20000000 | 32 | R/W | Interrupt handler 0 work space pointer |
| INC_HANDLERWPTR1 | 0x20000004 | 32 | R/W | Interrupt handler 1 work space pointer |
| INC_HANDLERWPTR2 | 0x20000008 | 32 | R/W | Interrupt handler 2 work space pointer |
| INC_HANDLERWPTR3 | 0x2000000C | 32 | R/W | Interrupt handler 3 work space pointer |
| INC_HANDLERWPTR4 | 0x20000010 | 32 | R/W | Interrupt handler 4 work space pointer |
| INC_HANDLERWPTR5 | 0x20000014 | 32 | R/W | Interrupt handler 5 work space pointer |
| INC_HANDLERWPTR6 | 0x20000018 | 32 | R/W | Interrupt handler 6 work space pointer |
| INC_HANDLERWPTR7 | 0x2000001C | 32 | R/W | Interrupt handler 7 work space pointer |
| INC_TRIGGERMODE0 | 0x20000040 | 3 | R/W | Interrupt 0 trigger mode |
| INC_TRIGGERMODE1 | 0x20000044 | 3 | R/W | Interrupt 1 trigger mode |
| INC_TRIGGERMODE2 | 0x20000048 | 3 | R/W | Interrupt 2 trigger mode |
| INC_TRIGGERMODE3 | 0x2000004C | 3 | R/W | Interrupt trigger mode |
| INC_TRIGGERMODE4 | 0x20000050 | 3 | R/W | Interrupt 3 trigger mode |
| INC_TRIGGERMODE5 | 0x20000054 | 3 | R/W | Interrupt 4 trigger mode |
| INC_TRIGGERMODE6 | 0x20000058 | 3 | R/W | Interrupt 5 trigger mode |
| INC_TRIGGERMODE7 | 0x2000005C | 3 | R/W | Interrupt 6 trigger mode |
| INC_MASK | 0x200000C0 | 17 | R/W | Interrupt enable mask |
| INC_SET_MASK | 0x200000C4 | 17 | W | Set a bit of the interrupt enable mask |
| INC_CLEAR_MASK | 0x200000C8 | 17 | W | Clear a bit of the interrupt enable mask |
| INC_PENDING | 0x20000080 | 8 | R/W | Interrupt pending |
| INC_SET_PENDING | 0x20000084 | 8 | W | Set a bit of the **Pending** register |
| INC_CLEAR_PENDING | 0x20000088 | 8 | W | Clear a bit of the **Pending** register |
| INC_EXEC | 0x20000100 | 8 | R/W | Interrupts executing |
| INC_SET_EXEC | 0x20000104 | 8 | W | Set a bit of the **Exec** register |
| INC_CLEAR_EXEC | 0x20000108 | 8 | W | Clear a bit of the **Exec** register |

**Table 13 Interrupt control (INC)**

| | | | | |
|---|---|---|---|---|
| INC_INT0PRIORITY | 0x20011000 | 3 | R/W | Internal interrupt 0 priority |
| INC_INT1PRIORITY | 0x20011004 | 3 | R/W | Internal interrupt 1 priority |
| INC_INT2PRIORITY | 0x20011008 | 3 | R/W | Internal interrupt 2 priority |
| INC_INT3PRIORITY | 0x2001100C | 3 | R/W | Internal interrupt 3 priority |
| INC_INT4PRIORITY | 0x20011010 | 3 | R/W | Internal interrupt 4 priority |
| INC_INT5PRIORITY | 0x20011014 | 3 | R/W | Internal interrupt 5 priority |
| INC_INT6PRIORITY | 0x20011018 | 3 | R/W | Internal interrupt 6 priority |
| INC_INT7PRIORITY | 0x2001101C | 3 | R/W | Internal interrupt 7 priority |
| INC_INT8PRIORITY | 0x20011020 | 3 | R/W | Internal interrupt 8 priority |

**Table 14 Interrupt level control**

| | | | | |
|---|---|---|---|---|
| INC_INT9PRIORITY | 0x20011024 | 3 | R/W | Internal interrupt 9 priority |
| INC_INT10PRIORITY | 0x20011028 | 3 | R/W | Internal interrupt 10 priority |
| INC_INT11PRIORITY | 0x2001102C | 3 | R/W | Internal interrupt 11 priority |
| INC_INT12PRIORITY | 0x20011030 | 3 | R/W | Internal interrupt 12 priority |
| INC_INT13PRIORITY | 0x20011034 | 3 | R/W | Internal interrupt 13 priority |
| INC_INT14PRIORITY | 0x20011038 | 3 | R/W | Internal interrupt 14 priority |
| INC_INT15PRIORITY | 0x2001103C | 3 | R/W | Internal interrupt 15 priority |
| INC_INT16PRIORITY | 0x20011040 | 3 | R/W | Internal interrupt 16 priority |
| INC_INT17PRIORITY | 0x20011044 | 3 | R/W | Internal interrupt 17 priority |
| INC_INT18PRIORITY | 0x20011048 | 3 | R/W | Internal interrupt 18 priority |
| INC_INT19PRIORITY | 0x2001104C | 3 | R/W | Internal interrupt 19 priority |
| INC_INT20PRIORITY | 0x20011050 | 3 | R/W | Internal interrupt 20 priority (reserved) |
| INC_INT21PRIORITY | 0x20011054 | 3 | R/W | Internal interrupt 21 priority (reserved) |
| INC_INT22PRIORITY | 0x20011058 | 3 | R/W | Internal interrupt 22 priority (reserved) |
| INC_INT23PRIORITY | 0x2001105C | 3 | R/W | Internal interrupt 23 priority (reserved) |
| INC_INT24PRIORITY | 0x20011060 | 3 | R/W | External interrupt 0 priority |
| INC_INT25PRIORITY | 0x20011064 | 3 | R/W | External interrupt 1 priority |
| INC_INT26PRIORITY | 0x20011068 | 3 | R/W | External interrupt 2 priority |
| INC_INT27PRIORITY | 0x2001106C | 3 | R/W | External interrupt 3 priority |
| INC_INT28PRIORITY | 0x20011070 | 3 | R/W | External interrupt 4 priority |
| INC_INT29PRIORITY | 0x20011074 | 3 | R/W | External interrupt 5 priority |
| INC_INT30PRIORITY | 0x20011078 | 3 | R/W | External interrupt 6 priority |
| INC_INPUTINTERRUPTS | 0x2001107C | 23 | R | Input interrupt status |
| INC_SRC0_TRIGGERMODE | 0x200110BC | 3 | R/W | Interrupt source 0 trigger mode |
| INC_SRC1_TRIGGERMODE | 0x200110C0 | 3 | R/W | Interrupt source 1 trigger mode |
| INC_SRC2_TRIGGERMODE | 0x200110C4 | 3 | R/W | Interrupt source 2 trigger mode |
| INC_SRC3_TRIGGERMODE | 0x200110C8 | 3 | R/W | Interrupt source 3 trigger mode |
| INC_SRC4_TRIGGERMODE | 0x200110CC | 3 | R/W | Interrupt source 4 trigger mode |
| INC_SRC5_TRIGGERMODE | 0x200110D0 | 3 | R/W | Interrupt source 5 trigger mode |
| INC_SRC6_TRIGGERMODE | 0x200110D4 | 3 | R/W | Interrupt source 6 trigger mode |
| INC_SRC7_TRIGGERMODE | 0x200110D8 | 3 | R/W | Interrupt source 7 trigger mode |
| INC_SRC8_TRIGGERMODE | 0x200110DC | 3 | R/W | Interrupt source 8 trigger mode |
| INC_SRC9_TRIGGERMODE | 0x200110E0 | 3 | R/W | Interrupt source 9 trigger mode |
| INC_SRC10_TRIGGERMODE | 0x200110E4 | 3 | R/W | Interrupt source 10 trigger mode |
| INC_SRC11_TRIGGERMODE | 0x200110E8 | 3 | R/W | Interrupt source 11 trigger mode |
| INC_SRC12_TRIGGERMODE | 0x200110EC | 3 | R/W | Interrupt source 12 trigger mode |
| INC_SRC13_TRIGGERMODE | 0x200110F0 | 3 | R/W | Interrupt source 13 trigger mode |
| INC_SRC14_TRIGGERMODE | 0x200110F4 | 3 | R/W | Interrupt source 14 trigger mode |
| INC_SRC15_TRIGGERMODE | 0x200110F8 | 3 | R/W | Interrupt source 15 trigger mode |
| INC_SRC16_TRIGGERMODE | 0x200110FC | 3 | R/W | Interrupt source 16 trigger mode |
| INC_SRC17_TRIGGERMODE | 0x20011100 | 3 | R/W | Interrupt source 17 trigger mode |
| INC_SRC18_TRIGGERMODE | 0x20011104 | 3 | R/W | Interrupt source 18 trigger mode |
| INC_SRC19_TRIGGERMODE | 0x20011108 | 3 | R/W | Interrupt source 19 trigger mode |

**Table 14 Interrupt level control**

| | | | | |
|---|---|---|---|---|
| INC_SRC20_TRIGGERMODE | 0x2001110C | 3 | R/W | Interrupt source 20 trigger mode (reserved) |
| INC_SRC21_TRIGGERMODE | 0x20011110 | 3 | R/W | Interrupt source 21 trigger mode (reserved) |
| INC_SRC22_TRIGGERMODE | 0x20011114 | 3 | R/W | Interrupt source 22 trigger mode (reserved) |
| INC_SRC23_TRIGGERMODE | 0x20011118 | 3 | R/W | Interrupt source 23 trigger mode (reserved) |
| INC_SRC24_TRIGGERMODE | 0x2001111C | 3 | R/W | External interrupt source 0 trigger mode |
| INC_SRC25_TRIGGERMODE | 0x20011120 | 3 | R/W | External interrupt source 1 trigger mode |
| INC_SRC26_TRIGGERMODE | 0x20011124 | 3 | R/W | External interrupt source 2 trigger mode |
| INC_SRC27_TRIGGERMODE | 0x20011128 | 3 | R/W | External interrupt source 3 trigger mode |
| INC_SRC28_TRIGGERMODE | 0x2001112C | 3 | R/W | External interrupt source 4 trigger mode |
| INC_SRC29_TRIGGERMODE | 0x20011130 | 3 | R/W | External interrupt source 5 trigger mode |
| INC_SRC30_TRIGGERMODE | 0x20011134 | 3 | R/W | External interrupt source 6 trigger mode |
| INC_SRC_CLEAR_MASK | 0x200111B4 | 31 | W | Clear interrupt-source enable register |
| INC_SRC_SET_MASK | 0x200111B8 | 31 | W | Set interrupt-source enable mask |
| INC_SRC_MASK | 0x200111BC | 31 | R/W | Interrupt-source enable mask |
| INC_SELNOTINV | 0x200111C8 | 3 | R/W | External interrupt pin is set to active high or active low |
| INC_EN_INT | 0x200111CC | 3 | R/W | Enable external interrupt to wake-up the low-power controller |
| INC_SRC_STATUS | 0x200111DC | 31 | R | Gives the latched status of an interrupt source |
| INC_SRC_CLEAR | 0x200111EC | 31 | R | Clears the INC_SRC_STATUS register |

**Table 14 Interrupt level control**

| Register | Address | Bits | Access | Description | |
|---|---|---|---|---|---|
| IRB_TX_PRE_SCALER_IR | 0x00 | 8 | W/R | Clock pre-scaler selection | RC Transmit Interface registers |
| IRB_TX_SUB_CARRIER_IR | 0x04 | 16 | W/R | sub-carrier freq. programming | |
| IRB_TX_SYM_TIME_IR[1] | 0x08 | 16 | W | symbol time programming | |
| IRB_TX_ON_TIME_IR[1] | 0x0C | 16 | W | symbol on time programming | |
| IRB_TX_INT_EN_IR | 0x10 | 3 | W/R | Transmit Interrupt enable register | |
| IRB_TX_INT_STATUS_IR | 0x14 | 4 | R | Transmit Interrupt status register | |
| IRB_TX_EN_IR | 0x18 | 1 | W/R | RC transmit enable register | |
| IRB_TX_CLR_UNDERRUN_IR | 0x1C | 1 | W | Clears the underrun status | |
| IRB_RX_ON_TIME_IR[1] | 0x40 | 16 | R | Received pulse time capture | RC Receive Interface registers for Infrared signals |
| IRB_RX_SYM_TIME_IR[1] | 0x44 | 16 | R | Received symbol time capture | |
| IRB_RX_INT_EN_IR | 0x48 | 4 | W/R | Receive Interrupt enable register | |
| IRB_RX_INT_STATUS_IR | 0x4C | 5 | R | Receive Interrupt status register | |
| IRB_RX_EN_IR | 0x50 | 1 | W/R | RC receive enable register | |
| IRB_RX_MAX_SYM_TIME_IR | 0x54 | 16 | W/R | Maximum RC symbol time register | |
| IRB_RX_CLR_OVERRUN_IR | 0x58 | 1 | W | Clears the overrun status | |
| IRB_RX_NOISE_SUPPRESS_WIDTH_IR | 0x5C | 8 | W/R | Reserved | |
| IRB_RC_IRDA_CONTROL | 0x60 | 1 | W/R | Reserved | |

**Table 15 Infra red blaster (IRB)**

| Register | Address | Bits | Access | Description | |
|---|---|---|---|---|---|
| IRB_RX_SAMPLING_RATE_COMMON | 0x64 | 4 | W/R | Sampling frequency division for UHF and IR frequencies. | Common to both RC and UHF receivers |
| IRB_RX_ON_TIME_UHF[1] | 0x80 | 16 | R | Received pulse time capture | RC Receive Interface registers for UHF signals |
| IRB_RX_SYM_TIME_UHF[1] | 0x84 | 16 | R | Received symbol time capture | |
| IRB_RX_INT_EN_UHF | 0x88 | 4 | W/R | Receive Interrupt enable register | |
| IRB_RX_INT_STATUS_UHF | 0x8C | 5 | R | Receive Interrupt status register | |
| IRB_RX_EN_UHF | 0x90 | 1 | W/R | RC receive enable register | |
| IRB_RX_MAX_SYM_TIME_UHF | 0x94 | 16 | W/R | Maximum RC symbol time register | |
| IRB_RX_CLR_OVERRUN_UHF | 0x98 | 1 | W | Clears the overrun status | |
| IRB_RX_NOISE_SUPPRESS_WIDTH_UHF | 0x9C | 8 | W/R | Noise suppression width | |

**Table 15 Infra red blaster (IRB)**

1. These locations have a quadruple buffer (i.e. a FIFO of length 4 words).

| Name | Address | Bits | Access | Description |
|---|---|---|---|---|
| LNK_STREAM_CONF_1 | 0x20038580 | 32 | R/W | Link configuration 1 |
| LNK_STREAM_CONF_2 | 0x20038600 | 32 | R/W | Link configuration 2 |
| LNK_STREAM_CONF_3 | 0x20038680 | 32 | R/W | Link configuration 3 |
| LNK_STREAM_ENn | 0x20038F00 +4n | 1 | R/W | Stream n enable |
| LNK_STAT | 0x20038F80 | 21 | R/W | Status register |
| LNK_STAT_FIFO | 0x20038F84 | 32 | R | FIFO status word |
| LNK_PACKET_LENGTH | 0x20038F88 | 12 | R/W | Number of bytes per packet |
| LNK_TIME_OUT | 0x20038F8C | 6 | R/W | AR threshold to reset the block |
| LNK_MODE | 0x20038F90 | 9 | R/W | Mode |
| LNK_PCR_STREAM | 0x20038F94 | 6 | R/W | PCR stream |
| LNK_AF1-0 | 0x20038F98 | 32 | R | Adaptation field bytes 0 to 3 |
| LNK_AF1-0 | 0x20038F9C | 32 | R | Adaptation field bytes 4 to 7 |
| LNK_V_PTS | 0x20038FA0 | 32 | R | Video time stamp |
| LNK_A_PTS | 0x20038FA4 | 32 | R | Audio time stamp |
| LNK_PCR | 0x20038FA8 | 32 | R | PCR |
| LNK_PCR_EXT | 0x20038FAC | 9 | R | PCR extension |
| LNK_AR_SIZE | 0x20038FB0 | 6 | R/W | AR size |
| LNK_SDAV_CONF | 0x20038FB4 | 27 | R/W | SDAV configuration |
| LNK_SDAV_DMA_EN | 0x20038FB8 | 6 | R/W | SDAV DMA enable |
| LNK_SDAV_DATA | 0x20038FBC | 32 | R/W | SDAV data |
| LNK_EN_LINK | 0x20038FC0 | 1 | R/W | Enable link |
| LNK_EXTRA_BITS | 0x20038FC8 | 13 | R/W | Extra bits |
| LNK_HDD_ADDSTOP | 0x20038904 | 16 | R/W | Write limit address |

**Table 16 Link (LNK)**

| Name | Address | Bits | Access | Description |
|---|---|---|---|---|
| LNK_HDD_ADDRHIGH | 0x2003890C | 9 | R/W | High part of the incoming address |
| LNK_HDD_ADDRBANK | 0x20038908 | 4 | R/W | High part of the incoming address |
| LNK_HDD_BUFSIZE | 0x20038900 | 13 | R/W | Buffer size |

**Table 16 Link (LNK)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| LPM_TIMER[31:0] | 0x20000400 | 32 | R/W | Low power timer least significant word |
| LPM_TIMER[32:63] | 0x20000404 | 32 | R/W | Low power timer most significant word |
| LPM_TIMERSTART | 0x20000408 | 1 | W | Low power timer start |
| LPM_ALARM[31:0] | 0x20000410 | 32 | R/W | Low power alarm least significant word |
| LPM_ALARM[32:39] | 0x20000414 | 8 | R/W | Low power alarm most significant word |
| LPM_ALARMSTART | 0x20000418 | 1 | W | Low power alarm start |
| LPM_SYSPLL | 0x20000420 | 2 | R/W | System clock PLL |
| LPM_SYSRATIO | 0x20000500 | 6 | R | System clock ratio |
| LPM_WDENABLE | 0x20000510 | 1 | R/W | Watchdog enable |
| LPM_WDFLAG | 0x20000514 | 1 | R | Watchdog flag |

**Table 17 Low Power Module (LPM)**

| Register | Address | Bits | Access | Notes |
|---|---|---|---|---|
| MOD_CONTROL | 0x20027000 | 2 | R/W | MAFEIF control |
| MOD_STATUS | 0x20027004 | 6 | R | MAFEIF status |
| MOD_INT_ENABLE | 0x20027008 | 3 | R/W | Interrupt enable |
| MOD_ACK | 0x2002700C | 8 | W | Acknowledge |
| MOD_BUFFER_SIZE | 0x20027010 | 7 | R/W | Buffer size |
| MOD_MAFE_CTRL | 0x20027014 | 8 | W | MAFE control |
| MOD_MAFE_STATUS | 0x20027018 | 8 | R | MAFE status |
| MOD_RECEIVE0_POINTER | 0x20027020 | 30 | R/W | Receive_memory_buffer_0 start address |
| MOD_RECEIVE1_POINTER | 0x20027024 | 30 | R/W | Receive_memory_buffer_1 start address |
| MOD_TRANSMIT0_POINTER | 0x20027028 | 30 | R/W | Transmit_memory_buffer_0 start address |
| MOD_TRANSMIT1_POINTER | 0x2002702C | 30 | R/W | Transmit_memory_buffer_1 start address |

**Table 18 Modem analog front-end interface (MOD)**

| Register | Address | Bits | Access | Notes |
|---|---|---|---|---|
| MPEGDMA0_BURSTSIZE | 0x20024000 | 5 | W | The number of bytes to be transferred in one burst |
| MPEGDMA0_HOLDOFF | 0x20024004 | 5 | W | Holdoff for MPEG decoder: range 0 to 31, where 0=0 delay cycles |

**Table 19 MPEGDMA0 (MPEGDMA)**

| Register | Address | Bits | Access | Notes |
|---|---|---|---|---|
| MPEGDMA0_ABORT | 0x20024008 | 1 | W | Abort all operation |
| MPEGDMA0_WHICHDEC | 0x2002400C | 2 | W | DMA destination pointer |
| MPEGDMA0_STATUS | 0x20024010 | 2 | R | Interrupt status register |
| MPEGDMA0_INTACK | 0x20024014 | 1 | W | Interrupt acknowledge register |
| MPEGDMA0_SRCADD | 0x20024018 | 32 | W | DMA source pointer |
| MPEGDMA0_CNTRL | 0x2002401C | 2 | R/W | Interrupt control register |
| MPEGDMA0_BLSIZE | 0x20024020 | 16 | W | Data block dimension to be transferred |

**Table 19 MPEGDMA0 (MPEGDMA)**

| Register | Address | Bits | Access | Notes |
|---|---|---|---|---|
| MPEGDMA1_BURSTSIZE | 0x20025000 | 5 | W | The number of bytes to be transferred in one burst |
| MPEGDMA1_HOLDOFF | 0x20025004 | 5 | W | Holdoff for MPEG decoder: range 0 to 31, where 0=0 delay cycles |
| MPEGDMA1_ABORT | 0x20025008 | 1 | W | Abort all operation |
| MPEGDMA1_WHICHDEC | 0x2002500C | 2 | W | DMA destination pointer |
| MPEGDMA1_STATUS | 0x20025010 | 2 | R | Interrupt status register |
| MPEGDMA1_INTACK | 0x20025014 | 1 | W | Interrupt acknowledge register |
| MPEGDMA1_SRCADD | 0x20025018 | 32 | W | DMA source pointer |
| MPEGDMA1_CNTRL | 0x2002501C | 2 | R/W | Interrupt control register |
| MPEGDMA1_BLSIZE | 0x20025020 | 16 | W | Data block dimension to be transferred |

**Table 20 MPEGDMA1 (MPEGDMA)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| OGT_CTL | 0x00000040 | 7 | R/W | Control register |
| OGT_LUT | 0x00000041 | 8 | R/W | Main look-up table |
| OGT_LUT_H1 | 0x00000042 | 8 | R/W | Highlight 1 look-up table |
| OGT_LUT_H2 | 0x00000043 | 8 | R/W | Highlight 2 look-up table |
| OGT_XDI | 0x00000044 | 2 | R/W | Active area horizontal position offset |
| OGT_XDI | 0x00000045 | 8 | R/W | Active area horizontal position offset |
| OGT_YDI | 0x00000046 | 2 | R/W | Active area vertical position offset |
| OGT_YDI | 0x00000047 | 8 | R/W | Active area vertical position offset |
| OGT_HL2XO | 0x00000048 | 2 | R/W | Active area vertical position offset |
| OGT_HL2XO | 0x00000049 | 8 | R/W | Active area vertical position offset |
| OGT_HL2YO | 0x0000004A | 2 | R/W | Highlight 2 area, start Y |
| OGT_HL2YO | 0x0000004B | 8 | R/W | Highlight 2 area, start Y |
| OGT_HL2XI | 0x0000004C | 2 | R/W | Highlight 2 area, end X |
| OGT_HL2XI | 0x0000004D | 8 | R/W | Highlight 2 area, end X |
| OGT_HL2YI | 0x0000004E | 2 | R/W | Highlight 2 area, end Y |
| OGT_HL2YI | 0x0000004F | 8 | R/W | Highlight 2 area, end Y |

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| OGT_STAT | 0x00000050 | 2 | R | Status register |

**Table 21 Overlay graphics and text (OGT)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| OSD_ACT | 0x0000003E | 7 | R/W | Active Signal |
| OSD_CFG | 0x00000091 | 3 | R/W | OSD Configuration |
| OSD_BDW | 0x00000092 | 6 | R/W | OSD Boundary Weight |
| OSD_OBP | 0x0000002B | 14 | Serial R/W | OSD Bottom Field Pointer |
| OSD_OTP | 0x0000002A | 14 | Serial R/W | OSD Top Field Pointer |

**Table 22 On-screen display**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| PES_CF1 | 0x00000040 | 7 | R/W | PES Audio Decoding Control |
| PES_CF2 | 0x00000041 | 8 | R/W | PES Video Parser Control |
| PES_TM1 | 0x00000042 | 8 | RO | DSM Trick Mode |
| PES_TM2 | 0x00000043 | 2 | RO | PES Parser Status |
| PES_TS [7:0] | 0x00000049 | 8 | RO | PES Time Stamps |
| PES_TS [15:8] | 0x0000004A | 8 | RO | PES Time Stamps |
| PES_TS [23:16] | 0x0000004B | 8 | RO | PES Time Stamps |
| PES_TS [31:24] | 0x0000004C | 8 | RO | PES Time Stamps |
| PES_TS [32:TSA] | 0x0000004D | 2 | RO | PES Time Stamps |

**Table 23 PES Parser (PES)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P0OUT | 0x2000C000 | 8 | R/W | PIO 0 output |
| PIO_SET_P0OUT | 0x2000C004 | 8 | W | Set bits of P0Out |
| PIO_CLEAR_P0OUT | 0x2000C008 | 8 | W | Clear bits of P0Out |
| PIO_P0IN | 0x2000C010 | 8 | R | PIO 0 input |
| PIO_P0C0 | 0x2000C020 | 8 | R/W | PIO 0 configuration 0 |
| PIO_SET_P0C0 | 0x2000C024 | 8 | W | Set bits of P0C0 |
| PIO_CLEAR_P0C0 | 0x2000C028 | 8 | W | Clear bits of P0C0 |
| PIO_P0C1 | 0x2000C030 | 8 | R/W | PIO 0 configuration 1 |
| PIO_SET_P0C1 | 0x2000C034 | 8 | W | Set bits of P0C1 |
| PIO_CLEAR_P0C1 | 0x2000C038 | 8 | W | Clear bits of P0C1 |
| PIO_P0C2 | 0x2000C040 | 8 | R/W | PIO 0 configuration 2 |
| PIO_SET_P0C2 | 0x2000C044 | 8 | W | Set bits of P0C2 |
| PIO_CLEAR_P0C2 | 0x2000C048 | 8 | W | Clear bits of P0C2 |
| PIO_P0COMP | 0x2000C050 | 8 | R/W | PIO 0 input comparison |

**Table 24 Parallel I/O 0 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_SET_P0COMP | 0x2000C054 | 8 | W | Set bits of P0Comp |
| PIO_CLEAR_P0COMP | 0x2000C058 | 8 | W | Clear bits of P0Comp. |
| PIO_P0MASK | 0x2000C060 | 8 | R/W | PIO 0 input comparison mask |
| PIO_SET_P0MASK | 0x2000C064 | 8 | W | Set bits of P0Mask |
| PIO_CLEAR_P0MASK | 0x2000C068 | 8 | W | Clear bits of P0Mask |

**Table 24 Parallel I/O 0 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P1OUT | 0x2000D000 | 8 | R/W | PIO 1 output |
| PIO_SET_P1OUT | 0x2000D004 | 8 | W | Set bits of P1Out |
| PIO_CLEAR_P1OUT | 0x2000D008 | 8 | W | Clear bits of P1Out |
| PIO_P1IN | 0x2000D010 | 8 | R | PIO 1 input |
| PIO_P1C0 | 0x2000D020 | 8 | R/W | PIO 1 configuration 0 |
| PIO_SET_P1C0 | 0x2000D024 | 8 | W | Set bits of P1C0 |
| PIO_CLEAR_P1C0 | 0x2000D028 | 8 | W | Clear bits of P1C0 |
| PIO_P1C1 | 0x2000D030 | 8 | R/W | PIO 1 configuration 1 |
| PIO_SET_P1C1 | 0x2000D034 | 8 | W | Set bits of P1C1 |
| PIO_CLEAR_P1C1 | 0x2000D038 | 8 | W | Clear bits of P1C1 |
| PIO_P1C2 | 0x2000D040 | 8 | R/W | PIO 1 configuration 2 |
| PIO_SET_P1C2 | 0x2000D044 | 8 | W | Set bits of P1C2 |
| PIO_CLEAR_P1C2 | 0x2000D048 | 8 | W | Clear bits of P1C2 |
| PIO_P1COMP | 0x2000D050 | 8 | R/W | PIO 1 input comparison |
| PIO_SET_P1COMP | 0x2000D054 | 8 | W | Set bits of P1Comp |
| PIO_CLEAR_P1COMP | 0x2000D058 | 8 | W | Clear bits of P1Comp |
| PIO_P1MASK | 0x2000D060 | 8 | R/W | PIO 1 input comparison mask |
| PIO_SET_P1MASK | 0x2000D064 | 8 | W | Set bits of P1Mask |
| PIO_CLEAR_P1MASK | 0x2000D068 | 8 | W | Clear bits of P1Mask |

**Table 25 Parallel I/O 1 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P2OUT | 0x2000E000 | 8 | R/W | PIO 2 output |
| PIO_SET_P2OUT | 0x2000E004 | 8 | W | Set bits of P2Out |
| PIO_CLEAR_P2OUT | 0x2000E008 | 8 | W | Clear bits of P2Out |
| PIO_P2IN | 0x2000E010 | 8 | R | PIO 2 input |
| PIO_P2C0 | 0x2000E020 | 8 | R/W | PIO 2 configuration 0 |
| PIO_SET_P2C0 | 0x2000E024 | 8 | W | Set bits of P2C0 |
| PIO_CLEAR_P2C0 | 0x2000E028 | 8 | W | Clear bits of P2C0 |
| PIO_P2C1 | 0x2000E030 | 8 | R/W | PIO 2 configuration 1 |

**Table 26 Parallel I/O 2 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_SET_P2C1 | 0x2000E034 | 8 | W | Set bits of P2C1 |
| PIO_CLEAR_P2C1 | 0x2000E038 | 8 | W | Clear bits of P2C1 |
| PIO_P2C2 | 0x2000E040 | 8 | R/W | PIO 2 configuration 2 |
| PIO_SET_P2C2 | 0x2000E044 | 8 | W | Set bits of P2C2 |
| PIO_CLEAR_P2C2 | 0x2000E048 | 8 | W | Clear bits of P2C2 |
| PIO_P2COMP | 0x2000E050 | 8 | R/W | PIO 2 input comparison |
| PIO_SET_P2COMP | 0x2000E054 | 8 | W | Set bits of P2Comp |
| PIO_CLEAR_P2COMP | 0x2000E058 | 8 | W | Clear bits of P2Comp |
| PIO_P2MASK | 0x2000E060 | 8 | R/W | PIO 2 input comparison mask |
| PIO_SET_P2MASK | 0x2000E064 | 8 | W | Set bits of P2Mask |
| PIO_CLEAR_P2MASK | 0x2000E068 | 8 | W | Clear bits of P2Mask |

**Table 26 Parallel I/O 2 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P3OUT | 0x2000F000 | 8 | R/W | PIO 3 output |
| PIO_SET_P3OUT | 0x2000F004 | 8 | W | Set bits of P3Out |
| PIO_CLEAR_P3OUT | 0x2000F008 | 8 | W | Clear bits of P3Out |
| PIO_P3IN | 0x2000F010 | 8 | R | PIO 3 input |
| PIO_P3C0 | 0x2000F020 | 8 | R/W | PIO 3 configuration 0 |
| PIO_SET_P3C0 | 0x2000F024 | 8 | W | Set bits of P3C0 |
| PIO_CLEAR_P3C0 | 0x2000F028 | 8 | W | Clear bits of P3C0 |
| PIO_P3C1 | 0x2000F030 | 8 | R/W | PIO 3 configuration 1 |
| PIO_SET_P3C1 | 0x2000F034 | 8 | W | Set bits of P3C1 |
| PIO_CLEAR_P3C1 | 0x2000F038 | 8 | W | Clear bits of P3C1 |
| PIO_P3C2 | 0x2000F040 | 8 | R/W | PIO 3 configuration 2 |
| PIO_SET_P3C2 | 0x2000F044 | 8 | W | Set bits of P3C2 |
| PIO_CLEAR_P3C2 | 0x2000F048 | 8 | W | Clear bits of P3C2 |
| PIO_P3COMP | 0x2000F050 | 8 | R/W | PIO 3 input comparison |
| PIO_SET_P3COMP | 0x2000F054 | 8 | W | Set bits of P3Comp |
| PIO_CLEAR_P3COMP | 0x2000F058 | 8 | W | Clear bits of P3Comp |
| PIO_P3MASK | 0x2000F060 | 8 | R/W | PIO 3 input comparison mask |
| PIO_SET_P3MASK | 0x2000F064 | 8 | W | Set bits of P3Mask |
| PIO_CLEAR_P3MASK | 0x2000F068 | 8 | W | Clear bits of P3Mask |

**Table 27 Parallel I/O 3 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P4OUT | 0x20010000 | 8 | R/W | PIO 4 output |
| PIO_SET_P4OUT | 0x20010004 | 8 | W | Set bits of P4Out |

**Table 28 Parallel I/O 4 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_CLEAR_P4OUT | 0x20010008 | 8 | W | Clear bits of P4Out |
| PIO_P4IN | 0x20010010 | 8 | R | PIO 4 input |
| PIO_P4C0 | 0x20010020 | 8 | R/W | PIO 4 configuration 0 |
| PIO_SET_P4C0 | 0x20010024 | 8 | W | Set bits of P4C0 |
| PIO_CLEAR_P4C0 | 0x20010028 | 8 | W | Clear bits of P4C0 |
| PIO_P4C1 | 0x20010030 | 8 | R/W | PIO 4 configuration 1 |
| PIO_SET_P4C1 | 0x20010034 | 8 | W | Set bits of P4C1 |
| PIO_CLEAR_P4C1 | 0x20010038 | 8 | W | Clear bits of P4C1 |
| PIO_P4C2 | 0x20010040 | 8 | R/W | PIO 4 configuration 2 |
| PIO_SET_P4C2 | 0x20010044 | 8 | W | Set bits of P4C2 |
| PIO_CLEAR_P4C2 | 0x20010048 | 8 | W | Clear bits of P4C2 |
| PIO_P4COMP | 0x20010050 | 8 | R/W | PIO 4 input comparison |
| PIO_SET_P4COMP | 0x20010054 | 8 | W | Set bits of P4Comp |
| PIO_CLEAR_P4COMP | 0x20010058 | 8 | W | Clear bits of P4Comp |
| PIO_P4MASK | 0x20010060 | 8 | R/W | PIO 4 input comparison mask |
| PIO_SET_P4MASK | 0x20010064 | 8 | W | Set bits of P4Mask |
| PIO_CLEAR_P4MASK | 0x20010068 | 8 | W | Clear bits of P4Mask |

**Table 28 Parallel I/O 4 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PIO_P5OUT | 0x2000D000 | 8 | R/W | PIO5 output |
| PIO_SET_P5OUT | 0x2000D004 | 8 | W | Set bits of P5Out |
| PIO_CLEAR_P5OUT | 0x2000D008 | 8 | W | Clear bits of P5Out |
| PIO_P5IN | 0x2000D010 | 8 | R | PIO 5 input |
| PIO_P5C0 | 0x2000D020 | 8 | R/W | PIO 5 configuration 0 |
| PIO_SET_P5C0 | 0x2000D024 | 8 | W | Set bits of P5C0 |
| PIO_CLEAR_P5C0 | 0x2000D028 | 8 | W | Clear bits of P5C0 |
| PIO_P5C1 | 0x2000D030 | 8 | R/W | PIO 5 configuration 1 |
| PIO_SET_P5C1 | 0x2000D034 | 8 | W | Set bits of P5C1 |
| PIO_CLEAR_P5C1 | 0x2000D038 | 8 | W | Clear bits of P5C1 |
| PIO_P5C2 | 0x2000D040 | 8 | R/W | PIO 5 configuration 2 |
| PIO_SET_P5C2 | 0x2000D044 | 8 | W | Set bits of P5C2 |
| PIO_CLEAR_P5C2 | 0x2000D048 | 8 | W | Clear bits of P5C2 |
| PIO_P5COMP | 0x2000D050 | 8 | R/W | PIO 5 input comparison |
| PIO_SET_P5COMP | 0x2000D054 | 8 | W | Set bits of P5Comp |
| PIO_CLEAR_P5COMP | 0x2000D058 | 8 | W | Clear bits of P5Comp |
| PIO_P5MASK | 0x2000D060 | 8 | R/W | PIO 5 input comparison mask |
| PIO_SET_P5MASK | 0x2000D064 | 8 | W | Set bits of P5Mask |
| PIO_CLEAR_P5MASK | 0x2000D068 | 8 | W | Clear bits of P5Mask |

**Table 29 Parallel I/O 5 (PIO)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| PWM_0VAL | 0x2000B000 | 9 | R/W | PWM 0 pulse width |
| PWM_1VAL | 0x2000B004 | 9 | R/W | PWM 1 pulse width |
| PWM_2VAL | 0x2000B008 | 9 | R/W | PWM 2 pulse width |
| PWM_3VAL | 0x2000B00C | 9 | R/W | PWM 3 pulse width |
| PWM_INTENABLE | 0x2000B054 | 9 | R/W | PWM interrupt enable |
| PWM_INTSTATUS | 0x2000B058 | 9 | R | PWM interrupt status |
| PWM_INTACK | 0x2000B05C | 9 | W | PWM interrupt acknowledge |
| PWM_CONTROL | 0x2000B050 | 11 | R/W | PWM control register |
| PWM_COUNT | 0x2000B060 | 8 | R/W | PWM output counter |
| PWM_0CAPTUREEDGE | 0x2000B030 | 2 | R/W | PWM 0 capture event definition |
| PWM_1CAPTUREEDGE | 0x2000B034 | 2 | R/W | PWM 1 capture event definition |
| PWM_2CAPTUREEDGE | 0x2000B038 | 2 | R/W | PWM 2 capture event definition |
| PWM_3CAPTUREEDGE | 0x2000B03C | 2 | R/W | PWM 3 capture event definition |
| PWM_0CAPTUREVAL | 0x2000B010 | 32 | R | PWM 0 capture value |
| PWM_1CAPTUREVAL | 0x2000B014 | 32 | R | PWM 1 capture value |
| PWM_2CAPTUREVAL | 0x2000B018 | 32 | R | PWM 2 capture value |
| PWM_3CAPTUREVAL | 0x2000B01C | 32 | R | PWM 3 capture value |
| PWM_0COMPAREVAL | 0x2000B020 | 32 | R/W | PWM 0 compare value |
| PWM_1COMPAREVAL | 0x2000B024 | 32 | R/W | PWM 1 compare value |
| PWM_2COMPAREVAL | 0x2000B028 | 32 | R/W | PWM 2 compare value |
| PWM_3COMPAREVAL | 0x2000B02C | 32 | R/W | PWM 3 compare value |
| PWM_0COMPAREOUTVAL | 0x2000B040 | 1 | R/W | PWM 0 compare output value |
| PWM_1COMPAREOUTVAL | 0x2000B044 | 1 | R/W | PWM 1 compare output value |
| PWM_2COMPAREOUTVAL | 0x2000B048 | 1 | R/W | PWM 2 compare output value |
| PWM_3COMPAREOUTVAL | 0x2000B04C | 1 | R/W | PWM 3 compare output value |
| PWM_CAPTURECOUNT | 0x2000B064 | 32 | R/W | PWM capture/compare counter |

**Table 30 PWM and counter module (PWM)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| SCI_SC0CLKVAL | 0x20007000 | 5 | W | SmartCard 0 clock |
| SCI_SC0CLKCON | 0x20007004 | 2 | W | SmartCard 0 clock control |

**Table 31 SmartCard Interface 0 (SCI)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| SCI_SC1CLKVAL | 0x20008000 | 5 | W | SmartCard 1 clock |
| SCI_SC1CLKCON | 0x20008004 | 2 | W | SmartCard 1 clock control |

**Table 32 SmartCard Interface 1 (SCI)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| SPD_CTL1 | 0x00000400 | 6 | R/W | Control Register 1 |
| SPD_SPR | 0x00000401 | 1 | R/W | Soft Reset |
| SPD_CTL2 | 0x00000402 | 2 | R/W | Control Register 2 |
| SPD_LUT | 0x00000403 | 8 | W | Main Lookup Table |
| SPD_XD0 [9:8] | 0x00000404 | 2 | R/W | Sub-picture X Offset |
| SPD_XD0 [7:0] | 0x00000405 | 8 | R/W | Sub-picture X Offset |
| SPD_YD0_1 [9:8] | 0x00000406 | 2 | R/W | Sub-picture Y Offset |
| SPD_YD0_2 [7:0] | 0x00000407 | 8 | R/W | Sub-picture Y Offset |
| SPD_HLSX [9:8] | 0x0000040C | 2 | R/W | Highlight Region Start X |
| SPD_HLSX [7:0] | 0x0000040D | 8 | R/W | Highlight Region Start X |
| SPD_HLSY [9:8] | 0x0000040E | 2 | R/W | Highlight Region Start Y |
| SPD_HLSY [7:0] | 0x0000040F | 8 | R/W | Highlight Region Start Y |
| SPD_HLEX [9:8] | 0x00000410 | 2 | R/W | Highlight Region End X |
| SPD_HLEX [7:0] | 0x00000411 | 8 | R/W | Highlight Region End X |
| SPD_HLEY [9:8] | 0x00000412 | 2 | R/W | Highlight Region End Y |
| SPD_HLEY [7:0] | 0x00000413 | 8 | R/W | Highlight Region End Y |
| SPD_HCOL | 0x00000414 | 8 | R/W | Highlight Region Color |
| SPD_HCOL | 0x00000415 | 8 | R/W | Highlight Region Color |
| SPD_HCN | 0x00000416 | 8 | R/W | Highlight Region Contrast |
| SPD_HCN | 0x00000417 | 8 | R/W | Highlight Region Contrast |
| SPD_SXD0 [9:8] | 0x00000424 | 2 | R/W | Sub-picture Display Area |
| SPD_SXD0 [7:0] | 0x00000425 | 8 | R/W | Sub-picture Display Area |
| SPD_SYD0 [9:8] | 0x00000426 | 2 | R/W | Sub-picture Display Area |
| SPD_SYD0 [7:0] | 0x00000427 | 8 | R/W | Sub-picture Display Area |
| SPD_SXD1 [9:8] | 0x00000428 | 2 | R/W | Sub-picture Display Area |
| SPD_SXD1 [7:0] | 0x00000429 | 8 | R/W | Sub-picture Display Area |
| SPD_SYD1 [9:8] | 0x0000042A | 2 | R/W | Sub-picture Display Area |
| SPD_SYD1 [7:0] | 0x0000042B | 8 | R/W | Sub-picture Display Area |

**Table 33 Sub-picture decoder (SPD)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| SSC_0_Con | 0x2000900C | 11 | R/W | SSC 0 control |
| SSC_0_I2C | 0x20009018 | 5 | R/W | SSC 0 I2C control |
| SSC_0_TBuf | 0x20009004 | 16 | W | SSC 0 transmit buffer |
| SSC_0_RBuf | 0x20009008 | 16 | R | SSC 0 receive buffer |
| SSC_0_SlAd | 0x2000901C | 16 | W | SSC 0 slave address |
| SSC_0_BRG | 0x20009000 | 10 | R/W | SSC 0 baud rate generation |
| SSC_0_Stat | 0x20009014 | 10 | R | SSC 0 status |
| SSC_0_IEn | 0x20009010 | 9 | R/W | SSC 0 interrupt enable |

**Table 34 Synchronous serial controller 0 (SSC)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| SSC_1_CON | 0x2000A00C | 11 | R/W | SSC 1 control |
| SSC_1_I2C | 0x2000A018 | 5 | R/W | SSC 1 $I^2C$ control |
| SSC_1_TBUF | 0x2000A004 | 16 | W | SSC 1 transmit buffer |
| SSC_1_RBUF | 0x2000A008 | 16 | R | SSC 1 receive buffer |
| SSC_1_SLAD | 0x2000A01C | 16 | W | SSC 1 slave address |
| SSC_1_BRG | 0x2000A000 | 10 | R/W | SSC 1 baud rate generation |
| SSC_1_STAT | 0x2000A014 | 10 | R | SSC 1 status |
| SSC_1_IEN | 0x2000A010 | 9 | R/W | SSC 1 interrupt enable |

**Table 35 Synchronous serial controller 1 (SSC)**

| REGISTER | Address | Bits | Access | Description |
|---|---|---|---|---|
| TTXT_ABORT | 0x2000A324 | 1 | W | Teletext abort |
| TTXT_ACKODDEVEN | 0x2000A320 | 1 | W | Teletext acknowledge odd or even |
| TTXT_DMAADDRESS | 0x2000A300 | 32 | R/W | Teletext DMA address |
| TTXT_DMACOUNT | 0x2000A304 | 11 | R/W | Teletext DMA count |
| TTXT_INTENABLE | 0x2000A31C | 3 | R/W | Teletext interrupt enable |
| TTXT_INTSTATUS | 0x2000A318 | 3 | R | Teletext interrupt status |
| TTXT_OUTDELAY | 0x2000A308 | 9 | R/W | Teletext output delay |

**Table 36 Teletext (Ttxt)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| USD_BMS[15:8] | 0x00000009 | 8 | R/W | Block move size |
| USD_BMS[7:0] | 0x00000009 | 8 | R/W | Block move size |
| USD_BRP[19:16] | 0x00000088 | 4 | R/W | Memory Read Pointer |
| USD_BRP[15:8] | 0x00000089 | 8 | R/W | Memory Read Pointer |
| USD_BRP[7:0] | 0x0000008A | 8 | R/W | Memory Read Pointer |
| USD_BWP[19:16] | 0x0000008C | 4 | R/W | Memory write pointer |
| USD_BWP[15:8] | 0x0000008D | 8 | R/W | Memory write pointer |
| USD_BWP[7:0] | 0x0000008E | 8 | R/W | Memory write pointer |
| USD_BSK[19:16] | 0x000000A4 | 8 | R/W | Block skip |
| USD_BSK[15:8] | 0x000000A5 | 8 | R/W | Block skip |
| USD_BSK[7:0] | 0x000000A6 | 8 | R/W | Block skip |

**Table 37 Block move (USD)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| VID_CFG_MCF | 0x00000000 | 7 | R/W | Memory refresh interval |
| VID_CFG_CCF | 0x00000001 | 7 | R/W | Chip configuration |
| VID_CTL | 0x00000002 | 5 | R/W | Decoding Control |
| VID_TIS | 0x00000003 | 6 | W | Task Instruction |
| VID_PFH | 0x00000004 | 8 | R/W | Picture F-parameters Horizontal |
| VID_PFV | 0x00000005 | 8 | R/W | Picture F-parameters Vertical |
| VID_PPR1 | 0x00000006 | 7 | R/W | Picture Parameters 1 |
| VID_PPR2 | 0x00000007 | 7 | R/W | Picture Parameters 2 |
| VID_DFP [14:8] | 0x0000000C | 7 | R/W | Displayed Luma Frame Pointer |
| VID_DFP [7:0] | 0x0000000D | 8 | R/W | Displayed Luma Frame Pointer |
| VID_RFP[14:8] | 0x0000000E | 7 | R/W | Reconstructed Frame Pointer |
| VID_RFP[7:0] | 0x0000000F | 8 | R/W | Reconstructed Frame Pointer |
| VID_FFP [13:8] | 0x00000010 | 6 | R/W | Forward Luma Frame Pointer |
| VID_FFP [7:0] | 0x00000011 | 8 | R/W | Forward Luma Frame Pointer |
| VID_BFP [14:8] | 0x00000012 | 7 | R/W | Backward Frame Pointer |
| VID_BFP [7:0] | 0x00000013 | 8 | R/W | Backward Frame Pointer |
| VID_VBG[14:8] | 0x00000014 | 7 | R/W | Start of Video Bit Buffer |
| VID_VBG[7:0] | 0x00000015 | 8 | R/W | Start of Video Bit Buffer |
| VID_VBL[14:8] | 0x00000016 | 7 | R | Video Bit Buffer Level |
| VID_VBL[7:0] | 0x00000017 | 8 | R | Video Bit Buffer Level |
| VID_VBS[14:8] | 0x00000018 | 7 | R/W | Video Bit Buffer Stop |
| VID_VBS[7:0] | 0x00000019 | 8 | R/W | Video Bit Buffer Stop |
| VID_VBT[14:8] | 0x0000001A | 7 | R/W | Video Bit Buffer Threshold |
| VID_VBT[7:0] | 0x0000001B | 8 | R/W | Video Bit Buffer Threshold |
| VID_ABG [14:8] | 0x0000001C | 7 | R/W | Start of audio bit buffer |
| VID_ABG [7:0] | 0x0000001D | 8 | R/W | Start of audio bit buffer |
| VID_ABL [14:8] | 0x0000001E | 7 | R | Audio Bit Buffer Level |
| VID_ABL [7:0] | 0x0000001F | 8 | R | Audio Bit Buffer Level |
| VID_ABS [14:8] | 0x00000020 | 7 | R/W | Audio Bit Buffer Stop |
| VID_ABS [7:0] | 0x00000021 | 8 | R/W | Audio Bit Buffer Stop |
| VID_ABT [14:8] | 0x00000022 | 7 | R/W | Audio Bit Buffer Threshold |
| VID_ABT [7:0] | 0x00000023 | 8 | R/W | Audio Bit Buffer Threshold |
| VID_DFS | 0x00000024 | 15 | Serial R/W | Decoded Frame Size |
| VID_DFW | 0x00000025 | 8 | R/W | Decoded Frame Width |
| VID_XFW | 0x00000028 | 8 | R/W | Displayed Frame Width |
| VID_PAN [10:8] | 0x0000002C | 3 | R/W | Pan/scan horizontal vector integer part |
| VID_PAN [7:0] | 0x0000002D | 8 | R/W | Pan/scan horizontal vector integer part |
| VID_PTH[13:8] | 0x0000002E | 6 | R/W | Panic threshold |
| VID_PTH[7:0] | 0x0000002F | 8 | R/W | Panic threshold |
| VID_STL | 0x00000030 | 2 | R/W | Launch start-code detector FIFO |

**Table 38 Video decoder (VID)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| VID_CWL | 0x00000031 | 1 | R/W | Launch compress data write |
| VID_656 | 0x00000032 | 1 | R/W | Enable 656 mode |
| VID_SRA | 0x00000035 | 1 | R/W | Audio Soft Reset |
| VID_CFG_DRC | 0x00000038 | 6 | R/W | SDRAM configuration |
| VID_SRV | 0x00000039 | 1 | R/W | Video Soft Reset |
| VID_CFG_GCF | 0x0000003A | 8 | R/W | General configuration |
| VID_STA | 0x0000003B | 8 | R | Status |
| VID_ITM | 0x0000003C | 8 | R/W | Interrupt Mask |
| VID_ITS | 0x0000003D | 8 | R | Interrupt Status |
| VID_TP_VID_LDP | 0x0000003F | 3 | R/W | Load Pointer |
| VID_YDS[8] | 0x00000046 | 1 | R/W | Display Y End |
| VID_YDS[7:0] | 0x00000047 | 8 | R/W | Display Y End |
| VID_SPREAD | 0x0000004E | 20 | Serial R/W | Sub-picture Read Pointer |
| VID_SPWRITE | 0x0000004F | 20 | Serial R/W | Sub-picture Write Pointer |
| VID_SPB[10:8] | 0x00000050 | 3 | R/W | Sub-picture Buffer Begin |
| VID_SPB[7:0] | 0x00000051 | 8 | R/W | Sub-picture Buffer Begin |
| VID_SPE[11:8] | 0x00000052 | 4 | R/W | Sub-picture Buffer End |
| VID_SPE[7:0] | 0x00000053 | 8 | R/W | Sub-picture Buffer End |
| VID_TRF[12:8] | 0x00000056 | 4 | R/W | Temporal Reference |
| VID_TRF[7:0] | 0x00000057 | 8 | R/W | Temporal Reference |
| VID_DFC [14:8] | 0x00000058 | 7 | R/W | Displayed Chroma Frame Pointer |
| VID_DFC [7:0] | 0x00000059 | 8 | R/W | Displayed Chroma Frame Pointer |
| VID_RFC[14:8] | 0x0000005A | 7 | R/W | Reconstructed Chroma Frame Pointer |
| VID_RFC[7:0] | 0x0000005B | 8 | R/W | Reconstructed Chroma Frame Pointer |
| VID_FFC [14:8] | 0x0000005C | 7 | R/W | Forward Chroma Frame Pointer |
| VID_FFC [7:0] | 0x0000005D | 8 | R/W | Forward Chroma Frame Pointer |
| VID_BFC [14:8] | 0x0000005E | 7 | R/W | Backward Chroma Pointer |
| VID_BFC [7:0] | 0x0000005F | 8 | R/W | Backward Chroma Pointer |
| VID_ITM | 0x00000060 | 8 | R/W | Interrupt Mask |
| VID_ITM | 0x00000061 | 8 | R/W | Interrupt Mask |
| VID_ITS | 0x00000062 | 8 | R | Interrupt Status |
| VID_ITS | 0x00000063 | 8 | R | Interrupt Status |
| VID_STA2 | 0x00000064 | 8 | R | Status |
| VID_STA3 | 0x00000065 | 8 | R | Status |
| VID_HDF | 0x00000066 | 16 | Serial R | Header Data FIFO |
| VID_CDCOUNT | 0x00000067 | 24 | Serial R | Bit Buffer Input Counter |
| VID_SCDCOUNT | 0x00000068 | 24 | Serial R | Bit Buffer Output Counter |
| VID_HDS | 0x00000069 | 4 | R/W | Header Search |
| VID_LSO | 0x0000006A | 8 | R/W | SRC Luminance Offset |
| VID_LSR [7:0] | 0x0000006B | 8 | R/W | SRC Luma Resolution |
| VID_CSO | 0x0000006C | 8 | R/W | SRC Chrominance Offset |
| VID_LSR [8] | 0x0000006D | 1 | R/W | SRC Luma Resolution |

**Table 38 Video decoder (VID)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| VID_YDO[8] | 0x0000006E | 1 | R/W | Display Y Offset |
| VID_YDO[7:0] | 0x0000006F | 8 | R/W | Display Y Offset |
| VID_XDO[9:8] | 0x00000070 | 2 | R/W | Display X Offset |
| VID_XDO[7:0] | 0x00000071 | 8 | R/W | Display X Offset |
| VID_XDS[9:8] | 0x00000072 | 2 | R/W | Display X End |
| VID_XDS[7:0] | 0x00000073 | 8 | R/W | Display X End |
| VID_DCF | 0x00000074 | 6 | R/W | Display Configuration |
| VID_DCF | 0x00000075 | 6 | R/W | Display Configuration |
| VID_QMW | 0x00000076 | 8 | W | Quantization Matrix Data |
| VID_TST | 0x00000077 | 1 | R/W | Test register |
| VID_REV | 0x00000078 | 8 | R | Device revision |
| VID_LCK | 0x0000007B | 1 | R/W | Locks registers VID_CFG_MCF, VID_CFG_CCF and VID_CFG_DRC |
| VID_TP_SCD_RD[15:8] | 0x00000081 | 8 | Read only | SCD pointer VLD load address |
| VID_TP_SCD_RD[7:0] | 0x00000082 | 8 | Read only | SCD pointer VLD load address |
| VID_TP_CD_RD[19:16] | 0x00000083 | 4 | Read only | CD pointer current address |
| VID_TP_CD_RD[15:8] | 0x00000084 | 8 | Read only | CD pointer current address |
| VID_TP_CD_RD[0:7] | 0x00000085 | 8 | Read only | CD pointer current address |
| VID_SCN | 0x00000087 | 6 | R/W | Pan/Scan Vertical Vector |
| VID_OUT | 0x00000090 | 4 | R/W | Output of 4:2:2 display |
| VID_BCK_Y | 0x00000098 | 8 | R/W | Background color Y |
| VID_BCK_U | 0x0000009A | 8 | R/W | Background color U |
| VID_BCK_V | 0x00000099 | 8 | R/W | Background color V |
| VID_MWV | 0x0000009B | 8 | R/W | Color mix between background color and video |
| VID_TP_SCD [16] | 0x000000C0 | 1 | R/W | SCD pointer load address |
| VID_TP_SCD [15:8] | 0x000000C1 | 8 | R/W | SCD pointer load address |
| VID_TP_SCD [7:0] | 0x000000C2 | 8 | R/W | SCD pointer load address |
| VID_TP_VLD_RD [19:16] | 0x000000C3 | 4 | Read only | VLD pointer current address |
| VID_TP_VLD_RD [15:8] | 0x000000C4 | 8 | Read only | VLD pointer current address |
| VID_TP_VLD_RD [7:0] | 0x000000C5 | 8 | Read only | VLD pointer current address |
| VID_TP_CD [16] | 0x000000CA | 1 | R/W | CD pointer load address |
| VID_TP_CD [15:8] | 0x000000CB | 8 | R/W | CD pointer load address |
| VID_TP_CD[7:0] | 0x000000CC | 8 | R/W | CD pointer load address |
| VID_TP_SCD_CURRENT [19:16] | 0x000000CD | 4 | Read only | SCD pointer current address |
| VID_TP_SCD_CURRENT [15:8] | 0x000000CE | 8 | Read only | SCD pointer current address |
| VID_TP_SCD_CURRENT | 0x000000CF | 8 | Read only | SCD pointer current address |
| VID_TP_VLD [15:8] | 0x000000D4 | 8 | R/W | VLD pointer load address |
| VID_TP_VLD [7:0] | 0x000000D5 | 8 | R/W | VLD pointer load address |
| VID_DCF | 0x000000D6 | 3 | R/W | Display Configuration |
| VID_TP_CDLIMIT [16] | 0x000000E0 | 1 | R/W | CD write limit address |

**Table 38 Video decoder (VID)**

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| VID_TP_CDLIMIT [15:8] | 0x000000E1 | 8 | R/W | CD write limit address |
| VID_TP_CDLIMIT [7:0] | 0x000000E2 | 8 | R/W | CD write limit address |
| VID_CMOD | 0x000000EA - 0x000000EE | 40 | R/W | Configure chrominance of block-row |
| VID_YMOD | 0x000000EF - 0x000000F3 | 40 | R/W | Configure luminance of block-row |

**Table 38 Video decoder (VID)**

# 3 Asynchronous serial controller (ASC)

## ASC_*n*_BAUDRATE ASC*n* baud rate generator

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | Reserved | | | | | | | | | | | | | | | | ReloadVal | | | | | | | | | | | | | | | |

Address: *ASCnBaseAddress* + 0x00
Type: Read/write
Reset value: 1

***Description***

The **ASC**_*n*_**BaudRate** register is the dual-function baud rate generator and reload value register. A read from this register returns the content of the 16-bit counter/accumulator; writing to it updates the 16-bit reload register.

If the **Run** bit of the control register is 1, then any value written in the **ASC**_*n*_**BaudRate** register is immediately copied to the timer. However, if the **Run** bit is 0 when the register is written, then the timer will not be reloaded until the first CPU clock cycle after the **Run** bit is 1.

The mode of operation of the baud rate generator depends on the setting of the **BaudMode** bit in the **ASC**_*n*_**Control** register. The 2 modes are described in the *Asynchronous Serial Controller* section of the ***STi5518*** *Datasheet*.

**Mode 0**

When the **BaudMode** bit in the **ASC**_*n*_**Control** register is set to 0, the baud rate and the required reload value for a given baud rate can be determined by the following formulae:

$$BaudRate = \frac{f_{CPU}}{16 \times ASCBaudRate}$$

$$ASCBaudRate = \frac{f_{CPU}}{16 \times BaudRate}$$

where: *ASCBaudRate* represents the content of the **ASC**_*n*_**BaudRate** register, taken as an unsigned 16-bit integer, $f_{CPU}$ is the frequency of the CPU.

Mode 0 shoud be used for all baud rates below 19.2K baud.

Table 39 lists commonly used baud rates with the required reload values and the approximate deviation errors for an example baud rate with a CPU clock of 60 MHz.

| Baud rate | Reload value (exact) | Reload value (integer) | Reload value (hex) | Approx. deviation error |
|---|---|---|---|---|
| 38.4 K | 97.656 | 98 | 0062 | 0.35% |
| 19.2 K | 195.313 | 195 | 00C3 | 0.16% |
| 9600 | 390.625 | 391 | 0091 | 0.1% |
| 4800 | 781.250 | 781 | 030D | 0.03% |
| 2400 | 1562.500 | 1563 | 061B | 0.03% |
| 1200 | 3125.000 | 3125 | 0C35 | 0.00% |
| 600 | 6250.000 | 6250 | 186A | 0.00% |

**Table 39 Mode 0 Baud rates**

| Baud rate | Reload value (exact) | Reload value (integer) | Reload value (hex) | Approx. deviation error |
|---|---|---|---|---|
| 300 | 12500.000 | 12500 | 30D4 | 0.00% |
| 75 | 50000.000 | 50000 | C350 | 0.00% |

**Table 39 Mode 0 Baud rates**

**Mode 1**

When the **BaudMode** bit in the **ASC**_*n*_**Control** register is set to 1, the baud rate is given by::

$$BaudRate = \frac{ASCBaudRate \times f_{CPU}}{16 \times 2^{16}}$$

where: $f_{CPU}$ is the CPU clock frequency and *ASCBaudRate* is the value written to the **ASC**_*n*_**BaudRate** register..

| Baud rate | Reload value (exact) | Reload value (integer) | Reload value (hex) | Approx. deviation error |
|---|---|---|---|---|
| 115200 | 2013.266 | 2013 | 07DD | 0.01% |
| 96000 | 1677.722 | 1678 | 068E | 0.02% |
| 38.4 K | 671.089 | 671 | 029F | 0.02% |
| 19.2 K | 335.544 | 336 | 0150 | 0.14% |

**Table 40 Mode 1 Baud rates**

Mode 1 shoud be used for baud rates of 19.2 K and above as it has a lower deviation error than Mode 0 at higher frequencies.

## ASC_*n*_CONTROL ASC*n* control register

| | 31 30 29 28 | 27 26 25 24 | 23 22 21 20 | 19 18 17 16 | 15 14 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 3 | 2 1 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | Reserved | | | | | Baud Mode | Cts Enable | Fifo Enable | SC Enable | Rx Enable | Run | Loop Back | Parity Odd | Stop Bits | Mode |

Address: *ASCnBaseAddress* + 0x0C
Type: Read/write
Reset value: 0

***Description***

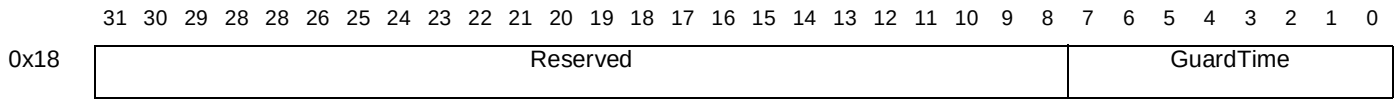
The **ASC**_*n*_**CONTROL** register controls the operating mode of the UART ASC*n* and contains control bits for mode and error check selection, and status flags for error identification.

Programming the mode control field (**Mode**) to one of the reserved combinations may result in unpredictable behavior. Serial data transmission or reception is only possible when the baud rate generator run bit (**Run**) is set to 1. When the **Run** bit is set to 0, **TxD** will be 1. Setting the **Run** bit to 0 will immediately freeze the state of the transmitter and receiver. This should only be done when the ASC is idle.

Serial data transmission or reception is only possible when the baud rate generator **Run** bit is set to 1. A transmission is started by writing to the transmit buffer register **ASC**_*n*_**TXBUFFER**.

| Bitfield | Description |
|---|---|
| Mode | ASC mode control: Mode2:<br>000: RESERVED<br>001: 8-bit data<br>010: RESERVED<br>011: 7-bit data + parity<br>100: 9-bit data<br>101: 8-bit data + wake up bit<br>110: RESERVED<br>111: 8-bit data + parity |
| StopBits | Number of stop bits selection:<br>00: 0.5 stop bits<br>01: 1 stop bits<br>10: 1.5 stop bits<br>11: 2 stop bits |
| ParityOdd | Parity selection:<br>0: Even parity (parity bit set on odd number of ‘1’s in data)<br>1: Odd parity (parity bit set on even number of ‘1’s in data) |
| LoopBack | Loopback mode enable bit:<br>0: Standard transmit/receive mode<br>1: Loopback mode enabled |
| Run | Baudrate generator run bit:<br>0: Baudrate generator disabled (ASC inactive)<br>1: Baudrate generator enabled |
| RxEnable | Receiver enable bit:<br>0: Receiver disabled<br>1: Receiver enabled |
| SCEnable | SmartCard enable bit:<br>0: SmartCard mode disabled<br>1: SmartCard mode enabled |
| FifoEnable | FIFO enable bit:<br>0: FIFO disabled<br>1: FIFO enabled |
| CtsEnable | CTS enable bit:<br>0: CTS ignored<br>1: CTS enabled |
| BaudMode | Baud rate generation mode:<br>0: Baud counter decrements, ticks when it reaches 1<br>1: Baud counter added to itself, ticks when there’s a carry |

## ASC_*n*_GUARDTIME ASC*n* guard time




Address: *ASCnBaseAddress* + 0x18
Type: Read/write
Reset value: 0

***Description***

The **ASC_n_GUARDTIME** register enables the user to delay the assertion of the interrupt **TxEmpty** by a programmable number of baud clock ticks. The value in the register is the number of baud clock ticks to delay assertion of **TxEmpty**. This value must be in the range 0 to 255.

## ASC_*n*_INTENABLE ASC*n* interrupt enable

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | Reserved | | | | | | | | | | | | | | | | | | | | | | | RHF | TOI | TNE | OE | FE | PE | THE | TE | RBE |

Address: *ASCnBaseAddress* + 0x10
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| RBE | **RxBufFullIE:** Receiver buffer full interrupt enable:<br>0: Receiver buffer full interrupt disable<br>1: Receiver buffer full interrupt enable |
| TE | **TxEmptyIE:** Transmitter empty interrupt enable:<br>0: Transmitter empty interrupt disable<br>1: Transmitter empty interrupt enable |
| THE | **TxHalfEmptyIE:** Transmitter buffer half empty interrupt enable:<br>0: Transmitter buffer half empty interrupt disable<br>1: Transmitter buffer half empty interrupt enable |
| PE | **ParityErrorIE:** Parity error interrupt enable:<br>0: Parity error interrupt disable<br>1: Parity error interrupt enable |
| FE | **FrameErrorIE:** Framing error interrupt enable:<br>0: Framing error interrupt disable<br>1: Framing error interrupt enable |
| OE | **OverrunErrorIE:** Overrun error interrupt enable:<br>0: Overrun error interrupt disable<br>1: Overrun error interrupt enable |
| TNE | **TimeoutNotEmptyIE:** Time-out when not empty interrupt enable:<br>0: Time-out when input FIFO or buffer not empty interrupt disable<br>1: Time-out when input FIFO or buffer not empty interrupt enable |
| TOI | **TimeoutIdleIE:** Time-out when the receiver FIFO is empty interrupt enable:<br>0: Time-out when the input FIFO or buffer is empty interrupt disable<br>1: Time-out when the input FIFO or buffer is empty interrupt enable |
| RHF | **RxHalfFullIE:** Receiver FIFO is half full interrupt enable:<br>0: Receiver FIFO is half full interrupt disable<br>1: Receiver FIFO is half full interrupt enable |

## ASC_*n*_RETRIES ASC*n* number of retries on transmission

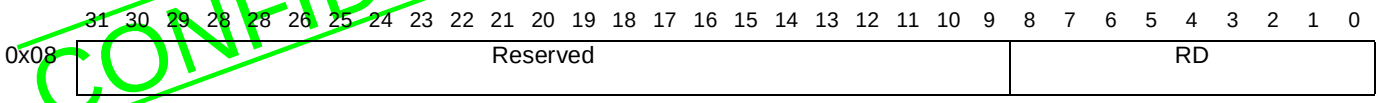
| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | Reserved | | | | | | | | | | | | | | | | | | | | | | | | Number of Retries | | | | | | | |

Address: *ASCnBaseAddress* + 0x28
Type: Read/write
Reset value: 1

***Description***

The **ASC**_*n*_**Retries** register defines the number of transmissions attempted on a piece of data before the UART discards the data. If a transmission still fails after 'Number of Retries' the **Nacked** bit is set in the **ASC**_*n*_**Status** register where it can be read and acted on by software. This register does not have to be re-initialized after a Nack error.

## ASC_*n*_RxBUFFER ASC*n* receive buffer




Address: *ASCnBaseAddress* + 0x08
Type: Read only
Reset value: 0

***Description***

Serial data reception is only possible when the baud rate generator **Run** bit in the **ASC**_*n*_**Control** register is set to 1.

| Bitfield | Description |
|---|---|
| RD[6:0] | Receive buffer data **D0 to D6** |
| RD[7] | Receive buffer data **D7**, or parity error bit - depending on the operating mode (the setting of the **Mode** bit of the **ASCControl** register). |
| RD[8] | Receive buffer data **D8**, or parity error bit, or wake-up bit - depending on the operating mode (the setting of the **Mode** field of the **ASCControl** register)<br>If the **Mode** field selects an 8-bit frame then this bit is undefined. Software should ignore this bit when reading 8-bit frames |

## ASC_*n*_RxRESET ASC*n* receive FIFO reset

Address: *ASCnBaseAddress* + 0x24
Type: Write only

***Description***

Reset the receiver FIFO. The ‘registers’ **ASC**_*n*_**RxReset** have no actual storage associated with them. A write of any value to one of these registers resets the corresponding receiver FIFO.

## ASC_*n*_STATUS ASC*n* interrupt status

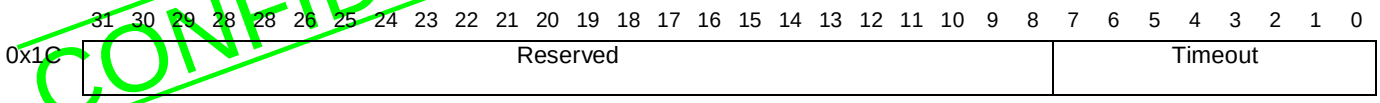
| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | Reserved | | | | | | | | | | | | | | | | | | | | NKD | TF | RHF | TOI | TNE | OE | FE | PE | THE | TE | RBF |

Address: *ASCnBaseAddress* + 0x14
Type: Read only
Reset value: 3 (i.e. Rx buffer full and Tx buffer empty)

***Description***

| Bitfield | Description |
|---|---|
| RBF | **RxBufFull:** Receiver FIFO not empty (FIFO operation) or buffer full (double-buffered operation):<br>0: Receiver FIFO is empty or buffer is not full<br>1: Receiver FIFO is not empty or buffer is full |
| TE | **TxEmpty:** Transmitter empty flag:<br>0: Transmitter is not empty<br>1: Transmitter is empty |
| THE | **TxHalfEmpty:** Transmitter FIFO at least half empty flag or buffer empty:<br>0: The FIFOs are enabled and the transmitter FIFO is more than half full (more than 8 characters) or the FIFOs are disabled and the transmit buffer is not empty.<br>1: The FIFOs are enabled and the transmitter FIFO is at least half empty (8 or less characters) or the FIFOs are disabled and the transmit buffer is empty |
| PE | **ParityError:** Input parity error flag:<br>0: No parity error<br>1: Parity error |
| FE | **FrameError:** Input frame error flag, i.e.stop bits not found:<br>0: No framing error<br>1: Framing error |
| OE | **OverrunError:** Overrun error flag:<br>0: No overrun error<br>1: Overrun error, i.e. data received when the input buffer is full |
| TNE | **TimeoutNotEmpty:** Time-out when the receiver FIFO or buffer is not empty:<br>0: No time-out or the receiver FIFO or buffer is empty<br>1: Time-out when the receiver FIFO or buffer is not empty |
| TOI | **TimeoutIdle:** Time-out when the receiver FIFO or buffer is empty:<br>0: No time-out or the receiver FIFO or buffer is not empty<br>1: Time-out when the receiver FIFO or buffer is empty |
| RHF | **RxHalfFull:** Receiver FIFO is half full:<br>0: The receiver FIFO contains 8 characters or less<br>1: The receiver FIFO contains more than 8 characters |
| TF | **TxFull:** Transmitter FIFO or buffer is full:<br>0: The FIFOs are enabled and the transmitter FIFO is empty or contains less than16 characters or the FIFOs are disabled and the transmit buffer is empty<br>1: The FIFOs are enabled and the transmitter FIFO contains 16 characters or the FIFOs are disabled and the transmit buffer is full |
| NKD | **Nacked:** Transmission failure acknowledgement by receiver:<br>0: Data transmitted successfully<br>1: Data transmission unsuccessful (data ‘nacked’ by SmartCard) |

## ASC_*n*_TIMEOUT ASC*n* time out




Address: *ASCnBaseAddress* + 0x1C
Type: Read/write
Reset value: 0

***Description***

The time-out period in baud rate ticks. The ASC contains an 8-bit time-out counter, which reloads from **ASC**_*n*_**Timeout** whenever one or more of the following is true:

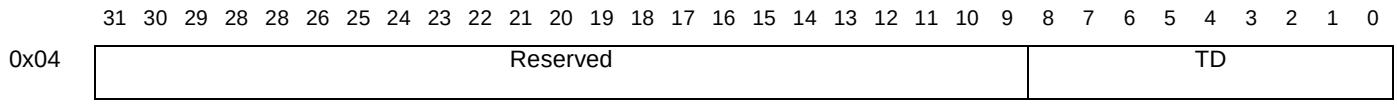
- **ASC**_*n*_**RxBuffer** is read;
- the ASC is in the middle of receiving a character;
- **ASC**_*n*_**Timeout** is written to.

If none of these conditions hold the counter decrements towards 0 at every baud rate tick.

The **TimeoutNotEmpty** bit of the **ASC**_*n*_**Status** register is 1 when the input FIFO is not empty and the time-out counter is zero.The **TimeoutIdle** bit of the **ASC**_*n*_**Status** register is 1 when the input FIFO is empty and the time-out counter is zero.

When the software has emptied the input FIFO, the time-out counter will reset and start decrementing. If no more characters arrive, when the counter reaches zero the **TimeoutIdle** bit of the **ASC**_*n*_**Status** register will be set.

## ASC_*n*_TxBUFFER ASC*n* transmit buffer




Address: *ASCnBaseAddress* + 0x04
Type: Write only
Reset value: 0

***Description***

A transmission is started by writing to the transmit buffer register **ASC**_*n*_**TxBuffer**. Serial data transmission is only possible when the baud rate generator **Run** bit in the **ASC**_*n*_**Control** register is set to 1.

Data transmission is double-buffered or uses a FIFO, so a new character may be written to the transmit buffer register before the transmission of the previous character is complete. This allows characters to be sent back-to-back without gaps.

| Bitfield | Description |
|---|---|
| TD[6:0] | Transmit buffer data **D0 to D6** |
| TD[7] | Transmit buffer data **D7**, or parity bit - depending on the operating mode (the setting of the **Mode** field of the **ASC-Control** register). |
| TD[8] | Transmit buffer data **D8**, or parity bit, or wake-up bit or undefined - depending on the operating mode (the setting of the **Mode** field of the **ASCControl** register).<br>If the **Mode** field selects an 8-bit frame then this bit should be written as 0. |

### ASC_*N*_TXRESET ASC*n* transmit FIFO reset

Address: *ASCnBaseAddress* + 0x20
Type: Write only

***Description***

Reset the transmitter FIFO. The ‘registers’ **ASC**_*n*_**TxReset** have no actual storage associated with them. A write of any value to one of these registers resets the corresponding transmitter FIFO.

# 4 Audio decoder (AUD)

The following table describes codes used in the right-hand column of Table 41: *Audio decoder register map by function* on page 39:

| Code | Description |
|---|---|
| (a) | Register modification is ALWAYS taken into account by the audio decoder.<br>Any change to these registers is taken into account immediately. |
| (b) | Register modification is taken into account AFTER EVERY DECODED DATA BLOCK or JUST AFTER RESET (soft or hard).<br>The decoded block is related to the granularity of the computation in the audio decoder software.<br>A block is 256 samples in Dolby Digital, 96 samples in MPEG, 80 samples in LPCM/PCM. |
| (f) | Register modification is taken into account AFTER EVERY DATA FRAME.<br>A frame is: 1152 samples in MPEG I/II, 1536 samples in Dolby Digital, 384 samples in MPEG-1 layer 1, 80 samples in LPCM/PCM. |
| (r) | Register modification is taken into account ONLY WHEN THE DSP IS RUN AFTER RESET (soft or hard). |
| (1) | The delay registers are updated when bit 0 of the AUD_UPDATE register is set to 1; the trick-mode registers are updated when bit 1 of the AUD_UPDATE register is set to 1. |
| (2) | The volume is updated when CHAN_IDX is set to the appropriate value. |

The following tables list the register map by address and function, then each audio decoder register is described individually.

| Register function | HEX | DEC | Name |
|---|---|---|---|
| VERSION | 0x00 | 0 | AUD_VERSION |
| | 0x01 | 1 | AUD_IDENT |
| | 0x71 | 113 | AUD_SOFVER |
| SETUP + INPUTS | 0x0C | 12 | AUD_SIN_SETUP (a) |
| | 0x0D | 13 | AUD_CAN_SETUP (a) |
| PCM CONFIGURATION | 0x54 | 84 | AUD_PCMDIVIDER (b) |
| | 0x55 | 85 | AUD_PCMCONF (b) |
| | 0x56 | 86 | AUD_PCMCROSS (b) |
| DAC AND PLL CONFIGURATION | 0x05 | 5 | AUD_SFREQ (f) |
| | 0x06 | 6 | AUD_EMPH (f) |
| | 0x12 | 18 | AUD_PLLPCM (a) |
| | 0x18 | 24 | AUD_PLLMASK (a) |
| CHANNEL DELAY SETUP | 0x57 | 87 | AUD_LDLY (1) |
| | 0x58 | 88 | AUD_RDLY (1) |
| | 0x59 | 89 | AUD_CDLY (1) |
| | 0x5A | 90 | AUD_SUBDLY (1) |
| | 0x5B | 91 | AUD_LSDLY (1) |
| | 0x5C | 92 | AUD_RSDLY (1) |
| | 0x5D | 93 | AUD_UPDATE (f) |

**Table 41 Audio decoder register map by function**

| Register function | HEX | DEC | Name |
|---|---|---|---|
| SPDIF OUTPUT SETUP | 0x5E | 94 | AUD_SPDIF_CMD(r) |
| | 0x5F | 95 | AUD_SPDIF_CAT(f) |
| | 0x60 | 96 | AUD_SPDIF_CONF(b) |
| | 0x61 | 97 | AUD_SPDIF_STATUS(b) |
| | 0x75 | 117 | AUD_SPDIF_REP_TIME(b) |
| | 0x7E | 126 | AUD_SPDIF_LATENCY(f) |
| | 0x7F | 127 | AUD_SPDIF_DTDI(f) |
| COMMAND | 0x10 | 16 | AUD_SOFTRESET(a) |
| | 0x13 | 19 | AUD_PLAY(a) |
| | 0x14 | 20 | AUD_MUTE(a) |
| | 0x72 | 114 | AUD_RUN(a) |
| | 0x73 | 115 | AUD_SKIP_MUTE_CMD(f) |
| | 0x74 | 116 | AUD_SKIP_MUTE_VALUE(f) |
| INTERRUPT | 0x07, 08 | 7, 8 | AUD_INTE(a) |
| | 0x09, 0A | 9, 10 | AUD_INT(a) |
| INTERRUPT STATUS | 0x40 | 64 | AUD_SYNC_STATUS |
| | 0x41 | 65 | AUD_ANCCOUNT |
| | 0x42 | 66 | AUD_HEAD4 (f) |
| | 0x43 | 67 | AUD_HEAD3(f) |
| | 0x44, 45 | 68, 69 | AUD_HEADLEN(f) |
| | 0x46 - 4A | 70 | AUD_PTS(f) |
| | 0x0F | 15 | AUD_ERROR |
| DECODING ALGORITHM | 0x4C | 76 | AUD_STREAMSEL(r) |
| | 0X4D | 77 | AUD_DECODESEL(r) |
| SYSTEM SYNCHRONIZATION | 0x4F | 79 | AUD_PACKET_LOCK(r) |
| | 0x50 | 80 | AUD_ID_EN(a) |
| | 0x51 | 81 | AUD_ID(a) |
| | 0x52 | 82 | AUD_ID_EXT(a) |
| | 0x53 | 83 | AUD_SYNC_LOCK(r) |
| POST DECODING AND PRO LOGIC | 0x62 | 98 | AUD_PDEC(b) |
| | 0x64 | 100 | AUD_PL_AB(b) |
| | 0x65 | 101 | AUD_PL_DWNX(b) |
| | 0x70 | 112 | AUD_DWSMODE(b) |
| BASS REDIRECTION | 0x4E | 78 | AUD_VOLUME0(2) |
| | 0x63 | 100 | AUD_VOLUME1(2) |
| | 0x66 | 102 | AUD_OCFG(b) |
| | 0x67 | 103 | AUD_CHAN_IDX(b) |

**Table 41 Audio decoder register map by function**

| Register function | HEX | DEC | Name |
|---|---|---|---|
| Dolby Digital configuration | 0x68 | 104 | AUD_AC3_DECODE_LFE (b) |
| | 0X69 | 105 | AUD_AC3_COMP_MOD (b) |
| | 0x6A | 106 | AUD_AC3_HDR (b) |
| | 0x6B | 107 | AUD_AC3_LDR (b) |
| | 0x6C | 108 | AUD_AC3_RPC (b) |
| | 0x6D | 109 | AUD_AC3_KARAMODE (b) |
| | 0x6E | 110 | AUD_AC3_DUALMODE (b) |
| | 0x6F | 111 | AUD_AC3_DOWNMIX (b) |
| | 0x76 | 118 | AUD_AC3_STATUS0 (f) |
| | 0x77 | 119 | AUD_AC3_STATUS1 (f) |
| | 0x78 | 120 | AUD_AC3_STATUS2 (f) |
| | 0x79 | 121 | AUD_AC3_STATUS3 (f) |
| | 0x7A | 122 | AUD_AC3_STATUS4 (f) |
| | 0x7B | 123 | AUD_AC3_STATUS5 (f) |
| | 0x7C | 124 | AUD_AC3_STATUS6 (f) |
| | 0x7D | 125 | AUD_AC3_STATUS7 (f) |
| MPEG configuration | 0x68 | 104 | AUD_MP_SKIP_LFE (b) |
| | 0x69 | 105 | AUD_MP_PROG_NUMBER (b) |
| | 0x6E | 106 | AUD_MP_DUALMODE (b) |
| | 0x6A | 110 | AUD_MP_DRC (b) |
| | 0x6C | 108 | AUD_MP_CRC_OFF (b) |
| | 0x6D | 109 | AUD_MP_MC_OFF (b) |
| | 0x6F | 111 | AUD_MP_DOWNMIX (b) |
| | 0x76 | 118 | AUD_MP_STATUS0 (f) |
| | 0x77 | 119 | AUD_MP_STATUS1 (f) |
| | 0x78 | 120 | AUD_MP_STATUS2 (f) |
| | 0x79 | 121 | AUD_MP_STATUS3 (f) |
| | 0x7A | 122 | AUD_MP_STATUS4 (f) |
| | 0x7B | 123 | AUD_MP_STATUS5 (f) |
| DTS configuration | 0x76 | 118 | AUD_DTS_STATUS0 (f) |
| | 0x77 | 119 | AUD_DTS_STATUS1 (f) |
| | 0x78 | 120 | AUD_DTS_STATUS2 (f) |
| | 0x79 | 121 | AUD_DTS_STATUS3 (f) |
| PCM beep-tone configuration | 0x68 | 104 | AUD_PCM_BTONE (b) |
| Audio trick-mode configuration | 0x57 | 87 | AUD_TM_SPEED (1) |

**Table 41 Audio decoder register map by function**

## 4.1 Version registers

### AUD_IDENT Identify

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x01 | | | | | | | | |

Address: *AudioBaseAddress* + 0x01
Type: RO
SW Reset: NA
HW Reset: NA

***Description***

IDENT is a read-only register and is used to identify the IC on an application board. IDENT always has the value “0xAC”.

### AUD_SOFVER Software version

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x71 | | | | | | | | |

Address: *AudioBaseAddress* + 0x71
Type: R/W
SW Reset: 0x47
HW Reset: 0x47

***Description***

This register gives the version of micro-code which is running on the device. For STi5518 cut A0, the value is 0x4B. For subsequent cuts, the value is given in application note7152120, STi5518 How to move from cutA0.

### AUD_VERSION Version

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x00 | | | | | | | | |

Address: *AudioBaseAddress* + 0x00
Type: RO
SW Reset: 0x20
HW Reset: 0x20

***Description***

This register gives the audio hardware version (binary decimal encoded). For STi5518 cut A0, the value is 0x30. For subsequent cuts, the value is given in application note 7152120, STi5518 How to move from cutA0.

## 4.2 Setup & input registers

### AUD_SIN_SETUP Input data setup

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0C | | | | | | | | |

Address: *AudioBaseAddress* + 0x0C
Type: R/W
SW Reset: NC
HW Reset: 0

***Description***

This register is reserved and must be set to 1.

### AUD_CAN_SETUP A/D converter setup

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0D | | | | | | | | |

Address: *AudioBaseAddress* + 0x0D
Type: R/W
SW Reset: NC
HW Reset: 0

***Description***

This register is reserved and must be set to 0.

## 4.3 PCM configuration resisters

### AUD_PCMDIVIDER Divider for PCM clock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x54 | | | | | | | | |

Address: *AudioBaseAddress* + 0x54
Type: R/W
SW Reset: UND
HW Reset: UND

***Description***

The PCM divider must be set according to the formula below, where DAC_SCLK is the bit clock for the DAC. When Div is set to 0, DAC_SCLK is equal to DAC_PCMCLK:

Div = (DAC_PCMCLK/ (2 x DAC_SCLK)) -1

When the internal PLL is used, DAC_PCMCLK=384 x fs or 256 x fs. If DAC_PCMCLK = 384 x fs, the formula becomes:

Div = (192 x Fs/DAC_SCLK) -1

If DAC_SCLK is 32 x Fs (common case with the 16 bit DAC), Div must be set to 5.

| PCM divider value | Mode description |
|---|---|
| 5 | DAC_PCMCLK = 384Fs, DAC is 16-bit mode |
| 3 | DAC_PCMLK = 256 Fs, DAC is 16-bit mode |
| 2 | DAC_PCMLK = 384 Fs, DAC is 32-bit mode |
| 1 | DAC_PCMLK = 256 Fs, DAC is 32-bit mode |

### AUD_PCMCONF PCM configuration

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x55 | | ORD | DIF | INV | FOR | SCL | PREC[1:0] | |

Address: *AudioBaseAddress* + 0x55
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| PREC[1:0] | PCM Precision<br>00: 16 bit mode (16 slots)<br>01: 18 bit mode (32 slots)<br>10: 20 bit mode (32 slots)<br>11: 24 bit mode (32 slots) |
| SCL | When set, the polarity of DAC_SCLK is inverted, the PCM outputs and DAC_LRCLK are stable for the DACs on the falling edge of DAC_SCLK.<br>When reset, PCM outputs and DAC_LRCLK are stable on the rising edge of DAC_SCLK. |
| FOR | 0: $I^2S$ format<br>1: Sony format |
| INV | 0: left channel is output when DAC_LRCLK is low ($I^2S$ format).<br>1: left channel is output when DAC_LRCLK is high (Sony format). |
| DIF | If set to zero, left padded. |
| ORD | PCM Order: only significant in 16-bit mode. When set, LSB is sent first. When reset, MSB is sent first. |

## AUD_PCMCROSS Cross PCM channels

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x56 | | | CLR[1:0] | | CSW[1:0] | | LRS[1:0] | |

Address: *AudioBaseAddress* + 0x56
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

The PCMCROSS register only acts if bit PFC of register AUD_SPDIF_DTDI is set.

| Bitfield | Description |
|---|---|
| LRS[1:0] | Cross left and right surround. |
| CSW[1:0] | Cross centre and subwoofer. |
| CLR[1:0] | Cross left and right channels.<br>00: Left channel is mapped on the left output, Right channel is mapped on the Right output.<br>01: Left channel is duplicated on both outputs.<br>10: Right channel is duplicated on both outputs.<br>11: Right channel and Left channel are toggled. |

## 4.4 DAC and PLL configuration registers

### AUD_SFREQ Sampling frequency

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x05 | | | | | | | | |

Address: *AudioBaseAddress* + 0x05
Type: R/WS
SW Reset: NC
HW Reset: 0

***Description***

This status register holds the code of the current sampling frequency. If the audio stream is encoded (Dolby Digital, MPEG) or packetized (DVD_LPCM), the sampling frequency is automatically read in the audio stream and written into this register by the audio DSP.

For PCM stream or CDDA, this register is written to by the application. The value in AUD_SFREQ corresponds to the following frequencies:

| Value | 0 | 1 | 2 | 3 | 4 | 8 | 9 | 10 | 11 (MP3 only) |
|---|---|---|---|---|---|---|---|---|---|
| Frequency (kHz) | 48 | 44.1 | 32 | not used | 96 | 24 | 22.05 | 16 | 12 (MP3 only) |

### AUD_EMPH Emphasis

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x06 | | | | | | | E[1:0] | |

Address: *AudioBaseAddress* + 0x06
Type: R/WS
SW Reset: NC
HW Reset: 0

***Description***

This register is used in MPEG, DVD_LPCM or CDDA modes; it is not supported in Dolby Digital.

In MPEG and DVD_LPCM modes, its register value is extracted from the bitstream. When the emphasis status changes (by setting bit DEM of the AUD_INT register), an interrupt is generated. In CDDA mode, the register value must be updated by the application.

The de-emphasis filter specified here is applied only if bit DEM of the AUD_PDEC register is set.

| Bitfield | Description |
|---|---|
| E[1:0] | 00: none, 01: 50/15µs, 10: reserved, 11: CCITT J.17 |

## AUD_PLLPCM PCM PLL disable

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x12 | | | | | | 256 | RP | DP |

Address: *AudioBaseAddress* + 0x12
Type: R/W
SW Reset: NC
HW Reset: 1

***Description***

When a Hard reset occurs the PLL is disabled. The host must set RP to 1 and set DP to 0 to enable it. When a Soft reset occurs the register remains unchanged.

| Bitfield | Description |
|---|---|
| DP | 0: Internal PCM PLL is used<br>1: Internal PCM PLL is deactivated, PCM_CLK pin is an input |
| RP | Run PLL. When high the DAC PLL is running. |
| 256 | 0: PCMCLK = 384 x fs<br>1: PCMCLK = 256 x fs |

## AUD_PLLMASK PCMCLK mask for half sampling frequency

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x18 | | | | | | | | HALF_FS |

Address: *AudioBaseAddress* + 0x18
Type: W
SW Reset: NC
HW Reset: 0

***Description***

| Bitfield | Description |
|---|---|
| HALF_FS | If the incoming bitstream is encoded with half sampling frequency, the device generates a PCM clock (for audio DAC)<br>1: At 256 x half_fs or 384 x half_fs (half_fs is equal to 24KHz, 22.05KHz, 16KHz).<br>0: At 256 x fs or 384 x fs (fs is equal to 48KHz, 44.1KHz, 32KHz).<br>This function is mainly use for DAC frequency adaptation. |

## 4.5 Channel delay set-up registers

The unit for the register delay contents is a group of 16 samples. The maximum delay is 35 ms (i.e. the sum of the delays on the six channels is limited by 35ms). E.g., when the sampling frequency is 48kHz, the sum of all the delays must be less than 105 = 35 ms * 48 KHz /16 samples.
When only one surround channel is present (in Pro Logic or other mode), the right surround delay must be cleared, and the left delay channel is used for both surround channels.

### AUD_LDLY Left channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x57 | | | | | | | | |

Address: *AudioBaseAddress* + 0x57
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on left channel, expressed in number of group of 16 samples. LDLY = delay (ms) * Fs (kHz) / 16

### AUD_RDLY Right channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x58 | | | | | | | | |

Address: *AudioBaseAddress* + 0x58
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on right channel, expressed in number of group of 16 samples. RDLY = delay (ms) * Fs (kHz) / 16

### AUD_CDLY Centre channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x59 | | | | | | | | |

Address: *AudioBaseAddress* + 0x59
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on center channel, expressed in number of group of 16 samples. CDLY = delay (ms) * Fs (kHz) / 16

## AUD_SUBDLY Subwoofer channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5A | | | | | | | | |

Address: *AudioBaseAddress* + 0x5A
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on subwoofer channel, expressed in number of group of 16 samples. SUBDLY = delay (ms) * Fs (kHz) / 16

## AUD_LSDLY Left surround channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5B | | | | | | | | |

Address: *AudioBaseAddress* + 0x5B
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on left surround channel, expressed in number of group of 16 samples. LSDLY = delay (ms) * Fs (kHz) / 16

## AUD_RSDLY Right surround channel delay

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5C | | | | | | | | |

Address: *AudioBaseAddress* + 0x5C
Type: R/W
SW Reset: NC
HW Reset: UND

Delay on right surround channel, expressed in number of group of 16 samples. RSDLY = delay (ms) * Fs (kHz) / 16 Note that when only one surround channel is used, this register must be reset at initialization.

## AUD_UPDATE PCM delay update

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5D | | | | | | | TM | DLY |

Address: *AudioBaseAddress*+ 0x5D
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

| Bitfield | Description |
|---|---|
| DLY | This bit must be set to 1 to force the DSP to update its delays values (read from the audio delay registers).<br>0: Delay values held in the audio delay registers are NOT updated in the DSP (i.e. the DSP will keep the delay values set previously).<br>1: The delay values held in the audio delay registers are updated in the DSP (i.e. the DSP will use the new values). This bit is automatically reset to zero after it the update has been carried out. |
| TM | This bit must be set to 1 to force the DSP to update the trick-mode speed (read from reg. AUD_TM_SPEED).<br>0: Audio trick-mode value held in the AUD_TM_SPEED register is NOT updated in the DSP (i.e. the DSP will keep the value set previously)<br>1: Audio trick-mode value held in the AUD_TM_SPEED register is updated in the DSP (i.e. the DSP will use the new values). This bit is automatically reset to zero after it the update has been carried out. |

## 4.6 SPDIF output set-up

### AUD_SPDIF_CMD SPDIF control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5E | | | | | | | | |

Address: *AudioBaseAddress* + 0x5E
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register is the control register. Several modes are available, the mode is selected by value:

| Value | Description | |
|---|---|---|
| 0 | Off mode: | The SPDIF is not working, the output line is idle. |
| 1 | Muted mode: | The outputs are PCM null data. |
| 2 | PCM mode: | The outputs are PCM data. Only the first two decoded channels (Left and Right) are transmitted. |
| 3 | Encoded: | The compressed bitstream is transmitted (See IEC61937 standard) |

### AUD_SPDIF_CAT Category code

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5F | CATCODE | | | | | | | |

Address: *AudioBaseAddress* + 0x5F
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

The table below defines the category codes, values not listed are reserved.

| Category code | Description | |
|---|---|---|
| 0 0 0 0 0 0 0 | General | |
| 1 0 0 0 0 0 0 | Experimental | |
| X X X 0 0 0 0 | Reserved | |
| X X X 1 0 0 0 | Solid State Memory | |
| 0 0 0 0 1 0 0 | Broadcast reception of dig. audio | Japan |
| 1 1 0 0 1 0 0 | Broadcast reception of dig. audio | United states |
| 0 0 0 1 1 0 0 | Broadcast reception of dig. audio | Europe |
| 1 0 0 0 1 0 0 | Broadcast reception of dig. audio | Electronic Software delivery |
| X X X X 1 0 0 | Broadcast reception of dig. audio | All other states are reserved |
| 0 0 0 0 0 1 0 | Digital / Digital converters and signal processing | PCM encoder/decoder |
| 0 1 0 0 0 1 0 | Digital / Digital converters and signal processing | Digital sound sampler |
| 0 0 1 0 0 1 0 | Digital / Digital converters and signal processing | Digital signal mixer |

**Table 42 Category codes**

| Category code | Description |
|---|---|
| 0 0 1 1 0 1 0 | Digital / Digital converters and signal processing Sample rate converter |
| X X X X 0 1 0 | Digital / Digital converters and signal processing All other states are reserved |
| X X 0 0 1 1 0 | A/D converter W/o copyright |
| X X 1 0 1 1 0 | A/D converter W/ copyright (using copy and L bits) |
| X X X 1 1 1 0 | Broadcast reception of dig. audio |
| 0 0 0 0 0 0 1 | Laser optical CD - Compatible with IEC 908 |
| 0 0 0 1 0 0 1 | Laser optical CD - Not compatible with IEC 908 (Magneto optical) |
| X X X X 0 0 1 | Laser optical All other states are reserved |
| 0 0 0 0 1 0 1 | Musical instruments, microphones, etc. Synthesizer |
| 0 0 0 1 1 0 1 | Musical instruments, microphones, etc. Microphone |
| X X X X 1 0 1 | Musical instruments, microphones, etc. All other states are reserved |
| 0 0 0 0 0 1 1 | Magnetic tape or disks DAT |
| 0 0 0 1 0 1 1 | Magnetic tape or disks Digital audio sound VCR |
| X X X X 0 1 1 | Magnetic tape or disks All other states are reserved |
| X X X X 1 1 1 | Reserved |
| 7 | Only cat. codes XXXX100, XXX1110, XXXX001 -> L bit |
| 0 | Original, commercially pre-recorded data |
| 1 | No indication of 1st generation or higher |
| 7 | All other categories |
| 0 | No indication of 1st generation or higher |
| 1 | Original, commercially pre-recorded data |

**Table 42 Category codes**

## AUD_SPDIF_CONF SPDIF PCMCLK divider

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x60 | LAT | SM=0 | Bit5 | DIV[4:0] | | | | |

Address: *AudioBaseAddress* + 0x60
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| DIV[4:0] | This field is the DAC_PCMCLK divider. It must be set according to the formula:<br>in 16 bit mode: IECDIV=(1+PCMDIV)/2-1; in 32 bit mode: IECDIV=PCMDIV |
| Bit5 | Must be set to 1 |
| SM | SYNC MUTE Mode, must be set to zero. |
| LAT | Configures the latency mode between the SPDIF output (in mode compressed) and the Audio output.<br>0: Auto-Latency: The latency is the transmission time for 2/3 of the payload, plus the time to decode an audio block.<br>For MPEG Auto-Latency, the latency is the following time depending of the sampling frequency in the incoming bitstream: MPEG 48KHz: 20.90ms, MPEG 44.1KHz: 22.95ms, MPEG 32KHz: 32.53ms.<br>1: User-programmable latency - the AUD_SPDIF_LATENCY register is used. |

The table below shows the relationship between the value of the IEC divider and the value of the PCM divider.

| PCM Divider Value | Mode Description | IEC Divider Value |
|---|---|---|
| 5 | DAC_PCMCLK = 384Fs, DAC is 16-bit mode | 2 |
| 3 | DAC_PCMLK = 256 Fs, DAC is 16-bit mode | 1 |
| 2 | DAC_PCMLK = 384 Fs, DAC is 32-bit mode | 2 |
| 1 | DAC_PCMLK = 256 Fs, DAC is 32-bit mode | 1 |

## AUD_SPDIF_STATUS SPDIF status bit

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x61 | | SFR | | | | PRE | COP | COM |

Address: *AudioBaseAddress* + 0x61
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register is used to set the value of the status bit in the IEC958 data stream.

| Bitfield | Description |
|---|---|
| COM | Compress data bit.<br>1: compressed mode<br>0: non compressed mode. |
| COP | 1: copy allowed<br>0: copy not allowed |
| PRE | 1: output has pre-emphasis<br>0: output does not have pre-emphasis |
| SFR | 0000: if sampling frequency = 44.1KHz<br>0010: if sampling frequency = 48KHz<br>0011: if sampling frequency = 32KHz<br>1010: if sampling frequency = 96KHz |

## AUD_SPDIF_REP_TIME SPDIF repetition time of a pause frame

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x75 | | | | | | | | |

Address: *AudioBaseAddress* + 0x75
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

In compressed mode, a burst of pause frames is sent when there are no more data to transmit (due to an error or a gap in the incoming bitstream, for example). This register sets the size of a pause frame in IEC frames: Dolby Digital =4, MPEG=32 and DTS=3.

## AUD_SPDIF_LATENCY Latency value

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7E | Value | | | | | | | |

Address: *AudioBaseAddress* + 0x7E
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

If bit LAT of register AUD_SPDIF_CONF is set, a delay can be configured between the output of IEC61937 in compressed mode and the output of the audio decoder. To configure a latency (in unit of seconds) this register has to be set according the following formula:

Value = L x $F_S$/8 where, L=Latency in s and FS=Sampling frequency in Hz

The minimum latency delay is 0; the maximum latency delay is the time to decode a frame:

| Bitfield | Description |
|---|---|
| Value | Dolby Digital: L = 1536 samples / Sampling frequency<br>MPEG: L = 1152 samples / Sampling frequency |

## AUD_SPDIF_DTDI SPDIF data-type information

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7F | PFC | DTD | | INF | | | | |

Address: *AudioBaseAddress* + 0x7F
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

<table>
<tr><th>Bitfield</th><th>Description</th></tr>
<tr><td>DTDI[4:0]</td><td>In Dolby Digital mode:
<table>
<tr><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td colspan="3">BSMOD[2..0]</td></tr>
</table>
In MPEG mode:
<table>
<tr><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>DR</td><td>K</td></tr>
</table></td></tr>
<tr><td>DTD</td><td>1: Data-type dependent information used for the SPDIF in compressed mode, can be set by the user. Refer to IEC958 standard for more information.<br>0: Transmitted DTDI are extracted from the stream.</td></tr>
<tr><td>PFC</td><td>1: PCMCROSS function enabled</td></tr>
</table>

## 4.7 Audio command registers

### AUD_SOFTRESET Soft reset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | | | | | | | | |

Address: *AudioBaseAddress* + 0x10
Type: W0
SW Reset: NA
HW Reset: NA

***Description***

When bit 0 of this register is set, a soft reset occurs. The command registers and the interrupt registers listed below are cleared.The decoder goes into idle mode and the volumes are cleared.

Command registers: AUD_MUTE, AUD_RUN, AUD_PLAY, AUD_SKIP_MUTE_CMD and AUD_SKIP_MUTE_VALUE

Interrupt registers: AUD_INTE, AUD_INT and AUD_ERROR

### AUD_PLAY Play

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x13 | | | | | | | | PLAY |

Address: *AudioBaseAddress* + 0x13
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

The PLAY command is treated according to the state of the decoder:

- When in idle mode, the PLAY value is not taken into account by the decoder.
- When in init mode, the PLAY value is not taken into account by the decoder.
- When in decode mode, PLAY enables the decoding, see table below:

| PLAY value | MUTE value | DAC_SCLK, DAC_LRCLK state | DAC_PCMOUT | Decoding |
|---|---|---|---|---|
| 0 | 0 | Not running | 0 | No |
| 0 | 1 | Running | 0 | No |
| 1 | 0 | Running | Decoded samples | Yes |
| 1 | 1 | Running | 0 | Yes |

### AUD_MUTE Mute

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x14 | | | | | | | | MUTE |

Address: *AudioBaseAddress* + 0x14

Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

The MUTE command is handled differently according to the state of the decoder:

- When in idle mode after hardware reset, setting MUTE to “1” automatically runs the DAC_SCLK and DAC_LRCLK clocks and outputs them to the DACs.
- When playing, setting MUTE to “1” mutes the PCM outputs.
- The AUD_MUTE register has no effect on the SPDIF output.

## AUD_RUN RUN decoding

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x72 | | | | | | | | RUN |

Address: *AudioBaseAddress* + 0x72
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

This register enables to exit from idle mode. After a soft or hard reset the decoder is in idle mode. It stays in this mode until the RUN is set.

In run mode the decoder takes into account the state of all the configuration registers and begins to decode.

The AUD_RUN register can only be reset by the SOFTRESET or REBOOT commands.

## AUD_SKIP_MUTE_CMD Skip or mute commands

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x73 | RESERVED | | | REB | PAU | BLK | SKP | GMUT |

Address: *AudioBaseAddress* + 0x73
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

This register cannot be used in MP3 decoding mode.The register is taken in account at a beginning of decoding a frame.

<table>
<tr><th>Bitfield</th><th>Description</th></tr>
<tr><td>GMUT</td><td>Global mute command.<br>1: PCM and S/PDIFoutputs are muted. In compressed mode, the S/PDIF transmits bursts of pause-frames<br>In non-compressed mode, the IEC60958 transmits exactly the same data as the decoder.<br>0: Normal operation.</td></tr>
<tr><td>SKP</td><td>Skip Frame. If set, the decoder skips the number of frames set in the SKIP_MUTE_VALUE register. This command is useful for Audio-Video synchronization when the audio decoder is in late. Once the frame skipped, the decoder clears this bit. For IEC60958/61937:<br>- In compressed mode, the IEC61937 do not transmit the skipped frames.<br>- In non-compressed mode, the IEC60958 do not transmit the skipped frame as the decoder. Note: The skip command has no effects in PCM or CDDA mode.</td></tr>
<tr><td>BLK</td><td>Soft-Mute block of frames. If set, the decoder mutes the number of blocks set in the SKIP_MUTE_VALUE register. This command is useful for Audio-Video synchronization when the video decoder is in late. Once the blocks muted, the decoder clears this bit. For IEC60958/61937:<br>- In compressed mode, the IEC61937 transmits pause frames burst. The value of repetition time and Gap length parameter are given is the following table.<br>- In non-compressed mode, the IEC60958 mutes exactly the same amount of data as the decoder.
<table>
<tr><td></td><td>Repetition Time</td><td>Gap Length Parameter</td></tr>
<tr><td>Dolby Digital</td><td>4</td><td>SKIP_MUTE_VALUE x 256</td></tr>
<tr><td>MPEG</td><td>32</td><td>SKIP_MUTE_VALUE x 96</td></tr>
</table></td></tr>
<tr><td>PAU</td><td>Pause command. If set, the decoder mutes during the time of decoding a frame, then and forever it tests if the bit has been reset by host. If the bit is still set, it mutes during the time of decoding a block of frame and so on. If reset, it plays the incoming stream. For IEC60958/61937:<br>- In compressed mode, the IEC61937 transmits a Pause frames burst which duration is a time of decoding a frame with gap_length_parameter_1. Then and forever, it test if the bit is still set. If the bit is still set, it transmits a Pause frame which duration is a time of decoding a block frame with gap-length parameter-2 and so on. If reset, it transmits the next frame.<br>- In non-compressed mode, the IEC60958 mutes exactly the same amount of data as the decoder.
<table>
<tr><td></td><td>Repetition Time</td><td>Gap Length Parameter_1</td><td>Gap Length parameter_2</td></tr>
<tr><td>Dolby Digital</td><td>4</td><td>6 x 256</td><td>256</td></tr>
<tr><td>MPEG Layer2</td><td>32</td><td>12 x 96</td><td>96</td></tr>
<tr><td>MPEG1 Layer1</td><td>32</td><td>4 x 96</td><td>96</td></tr>
</table></td></tr>
<tr><td>REB</td><td>Reboot command.<br>If set, the decoder soft mutes the next frame. Then the decoder cannot receive new data and it is reinitialized.<br>The decoder clears the Reboot bit and the Run register. Then, the decoder is idle and a new configuration, in user registers, can be loaded by host. To restart the decoder, the Run register must be set. The clocks are never stopped, the decoder transmits null data to the DAC in idle state. For IEC60958/61937:<br>- In compressed mode, the IEC61937 transmits a burst of pause-frames, of duration equal to gap_length_parameter_1. Then it transmits some pause-frames with gap_paremeter length equal to 0, until the decoder is restarted. The size of pause-frame is defined in the AUD_SPDIF_REP_TIME register.<br>- In non-compressed mode, the IEC60958 transmits exactly the same data as the decoder. The IEC60958/61937 is never stopped.<br>For correct termination of the reboot procedure, two frames must be sent to the audio decoder.</td></tr>
</table>

## AUD_SKIP_MUTE_VALUESkip frames or mutes blocks of frame

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x74 | | | | | | | | |

Address: *AudioBaseAddress* + 0x74
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

This register works according to soft mute block of frames (bit BLK) or skip frame (bit SKP) in the AUD_SKIP_MUTE_CMD register.

If the command SKIP is selected, the decoder skips n frames, and then clears the register.

If the command AUD_SOFT_MUTE_BLOCK is selected, the decoder mutes n blocks, and then clears the register. n is the value of the AUD_SKIP_MUTE_VALUE register. The following table gives the number of sample in a block or in a frame.

| | **Samples by frames** | **Samples by blocks** |
|---|---|---|
| Dolby Digital | 1536 | 256 |
| MPEG layer II | 1152 | 96 |
| MPEG layer I | 384 | 96 |

## 4.8 Interrupt registers

### AUD_INTE Interrupt enable

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x08 | INTE[15:8] | | | | | | | |
| 0x07 | INTE[7:0] | | | | | | | |

Address: *AudioBaseAddress* + 0x08 - 0x07
Type: R/W
SW Reset: 0
HW Reset: 0

***Description***

The audio decoder contains a 16 bit interrupt register associated with a 16 bit “enable” register. A bit set in this register will enable the corresponding interrupt. The interrupt associated with each bit is given in the register INT description.

### AUD_INT Interrupt

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0A | RESERVED | | | | FIO | FBE | FBF | PCM |
| 09 | ANC | PTS | BOF | DEM | SFR | ERR | HDR | SYN |

Address: *AudioBaseAddress* + 0x0A - 0x09
Type: RO
SW Reset: 0
HW Reset: 0

***Description***

An interrupt is signalled whenever one of the bits of INT become set. This can only occur if the corresponding bit is set in the INTE register. The Table below shows the condition indicated by each bit.

| Bitfield | Description |
|---|---|
| SYN | Change in Synchronization Status [1] |
| HDR | Valid Header Registered [1] |
| ERR | Error Detected [1] |
| SFR | Sampling frequency changed [2] |
| DEM | De-emphasis Changed [2] |
| BOF | First Bit of New Frame at Output Stage [2] |
| PTS | First Bit of New Frame with PTS at Output Stage [1] |
| ANC | Ancillary Data Registered [1] |
| PCM | PCM Output Underflow [2] |
| FBF | Frame Buffer Full: The frame buffer memory contains 2 frames: one decoded, and one parsed for next decoding |
| FBE | The frame buffer memory contains 1 frame which begins to be decoded. The next frame begins to be parsed |

| Bitfield | Description |
|---|---|
| FIO | FIFO Input has Overflowed [2] |
| TBD | Reserved |
| TBD | Reserved |
| TBD | Reserved |
| TBD | Reserved |

1. Cleared when the MSB of the corresponding register is read, or when a reset occurs. Affected registers are listed in the following table.
2. Cleared when the MSB of the interrupt register is read or when a reset occurs.

| Address | Name |
|---|---|
| 0x0F | AUD_ERROR |
| 0x40 | AUD_SYNCSTATUS |
| 0x41 | AUD_ANCCOUNT |
| 0x42 | AUD_HEAD 4 |
| 0x46 | AUD_PTS |

## 4.9 Interrupt status registers

### AUD_SYNC_STATUS Synchronization status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x40 | | | | | PAC | | FRA | |

Address: *AudioBaseAddress* + 0x40
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

This register indicates the status of the audio parser for synchronization. It is used in conjunction with PACKET_LOCK and SYNCK_LOCK registers. On read the synchronization status interrupt bit is cleared (INT.SYN is cleared).

| Bitfield | Description |
|---|---|
| FRA | Frame Status<br>0 0: Research audio synchronization<br>0 1: Wait for confirmation - a synchro word has been detected but the parser has not yet detected SYNC-LOCK+1 synchro words.<br>1 0: Synchronized - SYNC_LOCK + 1 synchro words have been detected<br>1 1: Not used |
| PAC | Packet Status<br>0 0: Research packet synchronization word<br>0 1: Wait for confirmation - - a synchro word has been detected but the parser has not yet detected PACKET_LOCK+1 synchro words.<br>1 0: Synchronized - PACKET_LOCK + 1 synchro words have been detected<br>1 1: Not used |

### AUD_ANCCOUNT Ancillary data

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x41 | | | | | | | | |

Address: *AudioBaseAddress* + 0x41
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

This value gives the number of ancillary data in the stream. The ancillary data interrupt bit ANC of the AUD_INT register is cleared by a read.

## AUD_HEAD4 HEADER 4 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| AC_3 | 0 | 0 | 0 | 0 | 0 | BSMOD | | |
| MPEG-2 | 0 | 0 | 0 | 0 | 0 | 0 | DR | K |
| OTHER | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Address: *AudioBaseAddress* + 0x42
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

This register contains header data HEAD[31:24]. The contents depend on the type of the frame.HEAD4[7:3] = 00000, in all cases.

When the host reads this register, the corresponding interrupt bit (HDR) is cleared.

Dolby Digital

| Bitfield | Description |
|---|---|
| HEAD4[2:0] | BSMOD if an Dolby Digital frame |

MPEG-2

| Bitfield | Description |
|---|---|
| HEAD4[2] | 0 |
| HEAD4[1] | DR=1 Dynamic range exists |
| HEAD4[0] | K=0 in normal mode, K=1 in Karaoke mode. |

OTHER

In all other types of frame HEAD4[2:0] = “000”

## AUD_HEAD3 HEADER 3 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x43 | 0 | 0 | 0 | DTYPE | | | | |

Address: *AudioBaseAddress* + 0x43
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

This register contains header data HEAD[23:16].HEAD3[7:5] = “000”, in all cases HEAD3[4:0] = DTYPE

DTYPE is the data type and is defined as follows:

| Bit | Description | |
|---|---|---|
| DTYPE | 0000: | Null data or Linear PCM |
| | 0001: | Dolby Digital |
| | 0100: | MPEG-1 Layer I |
| | 0101: | MPEG-1 Layer II or MPEG-2 word extension |
| | 0110: | MPEG-2 Layer II with extension |
| | 1001: | MPEG-2 Layer II low sample rate |

*Note This register can not detect the data-type of data in a stream.*

## AUD_HEADLEN Frame length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x44 | HEADLEN[15:8] | | | | | | | |
| 0x45 | HEADLEN[7:0] | | | | | | | |

Address: *AudioBaseAddress* + 0x44 - 0x45
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

The HEADLEN register contains the bit length of the compressed data frame HEAD[15:0]. HEADER registers are all updated as soon as the decoder begins to decode a frame.

## AUD_PTS PTS

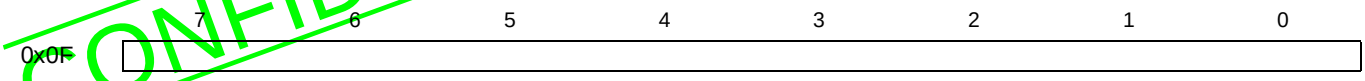
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x46 | | | | | | | | PTS[32] |
| 0x47 | PTS[31:24] | | | | | | | |
| 0x48 | PTS[23:16] | | | | | | | |
| 0x49 | PTS[15:8] | | | | | | | |
| 0x4A | PTS[7:0] | | | | | | | |

Address: *AudioBaseAddress* + 0x46 to 0x4A
Type: R/W
SW Reset: UND
HW Reset: UND

***Description***

When the PTS interrupt is activated, a new PTS value is stored in this register. Once the PTS[32] value is read bit PTS of the AUD_PTS register is cleared.

## AUD_ERROR ERROR code




Address: *AudioBaseAddress* + 0x0F
Type: RO
SW Reset: 0
HW Reset: 0

***Description***

This is a status register, when read by the ST20, this and the corresponding interrupt register are cleared. This register 7-bit register is ANDed with 0x7F to get the correct value. The value in the ERROR register indicates the type of error that has occurred. These errors are defined in the table below.

| Error Name | Value |
|---|---|
| **Dolby Digital Decoding** | |
| NO-ERROR | 0 |
| EXPAND_DELTA_PAST_END_ARRAY | 1 |
| XDCALL_TRY_TO_REUSE_REMAT_FLG | 2 |
| XDCALL_TRY_TO_REUSE_COUPLING_STRA | 3 |
| XDCALL_CANT_COUPLE_IN_DUAL_MODE | 4 |
| XDCALL_TRY_TO_REUSE_CPL_LEAK | 5 |
| XDCALL_TRY_TO_REUSE_SNR | 6 |
| XDCALL_TRY_TO_REUSE_BIT_ALLOC | 7 |
| XDCALL_TRY_TO_REUSE_COUPLING_EXPONENT_STRA | 8 |
| XDCALL_TRY_TO_REUSE_EXPONENT_STRA | 9 |
| XDCALL_TRY_TO_REUSE_LFE_EXPONENT_STRA | 10 |
| XDCALL_CHBWCOD_IS_TOO_HIGH | 11 |
| BSI_ERR_REV | 12 |
| BSI_ERR_CHANS | 13 |
| CRC_NOT_VALID | 14 |
| **Packet Synchronization** | |
| SYNCHRO_PACKET_NOT_FOUND | 16 |
| BAD_MPEGI_RESERVED_WORD | 17 |
| BAD_MPEG2_RESERVED_WORD | 18 |
| BAD_LPCM_SYNCHRO | 19 |
| UNKNOWN_STREAM_ID | 20 |
| MARKER_ERROR | 21 |
| UNKNOWN_SUB_STREAM_ID | 22 |
| IEC958_INPUT_MISMATCH_CONF | 23 |
| IEC958_MPEG2_LAYERI_NOT_SUPPORTED | 24 |
| IEC958_PAUSE_FRAME_NOT_SUPPORTED | 25 |
| IEC958_BAD_DATA_TYPE_DEPENDANT | 26 |
| MISMATCH _HOST_SEL_CONFIGURATION | 27 |
| **Audio Synchronization** | |
| SYNCHRO_AUDIO_NOT_FOUND | 32 |
| BAD_CRC_AC3 | 33 |

| | |
|---|---|
| BAD_LPCM_QUANTIZATION_WORDLENGTH | 34 |
| BAD_AUDIO_SAMPLING_FREQUENCY | 35 |
| BAD_MPEG_LAYER | 36 |
| MPEG_BITRATE_FREE_FORMAT | 37 |
| NOT_SUPPORTED_AC-3_FRMSIZECOD | 38 |
| BAD_CRC_MPEG_EXTENDED | 39 |
| BAD_MPEG_EXTENDED_RESERVED_BIT | 40 |
| MPEG_EXTENDED_SYNC_NOT_FOUND | 41 |
| MPEG_EXTENDED_LENGTH_TOO_SMALL | 42 |
| **DTS Transmission** | |
| BAD_SAMPLES_PER_CHANNEL | 44 |
| BAD_FRAME_BIT_SIZE | 45 |
| **MPEG Decoding** | |
| MPEG_EXTENSION_ERROR | 48 |
| MPEG_MC_MUTE | 49 |
| NOT USED | 50 |
| NOT USED | 51 |
| MPEG_LAYER_ERROR | 52 |
| MPEG_CHCONFIG_ERROR | 53 |
| MPEG_MC_PREDICTION_ERROR | 54 |
| MPEG_CRC_ERROR | 55 |
| MPEG_EXT_CRC_ERROR | 56 |
| MPEG_TOO_SMALL_FOR_MC_HEADER | 57 |
| MPEG_BITRATE_ERROR | 58 |
| **Miscellaneous** | |
| IEC_958_READ_ERROR | 64 |
| MPEG_FB_BYPASS_AREA_ERROR | 65 |
| SKIPPING_BITS_IN_FB_ERROR | 66 |
| LATENCY_TOO_BIG | 67 |
| SKIP_MUTE_ERROR | 68 |
| UNKNOW_SFREQ_FOR_LATENCY | 69 |
| LATENCY_TOO_SMALL | 70 |

## 4.10 Decoding algorithm registers

The table below shows how the AUD_STREAMSEL and AUD_DECODESEL registers should be programmed for different types of bitstream.

| AUD_STREAMSEL (0x4C) | AUD_DECODESEL (0x4D) | Mode |
|---|---|---|
| 0 | 0 | MPEG2 PES carrying Dolby Digital (ATSC) |
| 0 | 1 | MPEG2 PES carrying MPEG1 frames |
| 0 | 2 | MPEG2 PES carrying MPEG2 frames |
| 1 | 0 | MPEG2 PES carrying Dolby Digital frames for DVD |
| 1 | 2 | MPEG2 PES carrying MPEG2 frames for DVD |
| 1 | 3 | MPEG2 PES carrying linear PCM for DVD |
| 1 | 6 | MPEG2 PES carrying DTS frames for DVD |
| 2 | 1 | MPEG1 packet carrying MPEG1 audio |
| 3 | 0 | Dolby Digital frames elementary streams |
| 3 | 1 | MPEG1 frame elementary streams |
| 3 | 2 | MPEG2 frame elementary stream |
| 3 | 3 | Stereo PCM (16bits samples) |
| 3 | 4 | Pink noise generator |
| 3 | 5 | CDDA (Stereo PCM 16 bits samples) |
| 3 | 6 | DTS elementary stream |
| 3 | 7 | Activate PCM beep tone |
| 3 | 8 | MP3 elementary streams |
| 5 | 0 | IEC61937 Input with Dolby Digital frames |
| 5 | 1 | IEC61937 Input with MPEG1 frames |
| 5 | 2 | IEC61937 Input with MPEG2 frames |
| 5 | 6 | IEC61937 Input with DTS frames |

**Table 43 STREAMSEL and DECODESEL programming definitions**

### AUD_DECODESEL Decoding algorithm

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4D | | | | | DEC | | | |

Address: *AudioBaseAddress* + 0x4D
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register identifies the audio data-type.

| Bitfield | Description |
|---|---|
| DEC | 000: Dolby Digital Decoding<br>001: MPEG1<br>010: MPEG2<br>011: PCM/LPCM<br>100: PINK NOISE generator<br>101: CD_DA<br>110: DTS<br>111: PCM beep tone generator<br>1000: MP3 |

## AUD_STREAMSEL STREAM selection

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4C | | | | | | STRSEL | | |

Address: *AudioBaseAddress* + 0x4C
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| STRSEL | 000: PES<br>001: PES DVD<br>010: Packet MPEG1<br>011: Elementary Stream/IEC60958<br>100: Reserved<br>101: SPDIF IEC61937 |

## 4.11 System synchronization registers

### AUD_PACKET_LOCK Packet lock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4F | | | | | | | | |

Address: *AudioBaseAddress* + 0x4F
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register specifies the number of supplementary packet synchro words that the packet parser must detect before it is considered as synchronized, and can send data to the audio parser (max=1, min=0). In this way, stream data can not be sent to the audio parser instead of packet sync words.

PACKET_LOCK = 0: the packet parser is synchronized when it has detected one packet synchro word.

PACKET_LOCK = 1: the packet parser is synchronized when it has detected two packet synchro words.

### AUD_ID_EN Enable audio ID

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x50 | | | | | | | | |

Address: *AudioBaseAddress* + 0x50
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

If set to 1, the audio decoder decodes only the stream corresponding to the stream-id or sub-stream-id of the packet layer. This selection is done through AUDIO_ID or AUDIO_ID_EXT registers. If set to 0, the decoder decodes all the audio packets.

### AUD_ID Audio ID

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x51 | | | | | | | | |

Address: *AudioBaseAddress* + 0x51
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

When decoding packets, it is possible to specify an identifier for a selected program. AUDIO_ID must be written with the packet ID. This feature is enabled when the register AUDIO_ID_EN is set, and only packets with matching ID are decoded.

For MPEG1 packets or PES, the 5 LSB bits are significant. For DVD PES (LPCM, Dolby Digital or MPEG), the 3 LSB bits are significant (see audio pack definition in DVD specifications).

These bits correspond to the stream number defined in the STREAM_ID field of the audio packet header, except for DVD, Dolby Digital or LPCM packets, where they correspond to the stream number defined in the SUB_STREAM_ID field.

## AUD_ID_EXT Audio extension

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x52 | | | | | | | | |

Address: *AudioBaseAddress* + 0x52
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

The 3 LSB bits of this register are significant. In case of DVD MPEG2 audio with extension bitstream (see DVD specifications), this register is used to select the stream defined in the STREAM_ID of the packets containing MPEG2 extension bit stream data.

## AUD_SYNC_LOCK SYNC lock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x53 | | | | | | | | |

Address: *AudioBaseAddress* + 0x53
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register specifies the number of supplementary audio synchro words that the audio parser must detect before it is considered as synchronized, and can send data to the decoder. In this way, stream data can not be sent to the decoder instead of audio sync words. Max value = 3; min value = 0.

SYNC_LOCK = 0, the audio parser is synchronized when it has detected one audio synchro word.

SYNC_LOCK = n > 0, when the audio parser has detected one synchro word, it waits until it detects n supplementary audio sync words. When it has detected (SYNC_LOCK+1) sync words, it sends the data to the decoder.

## 4.12 Post decoding and pro logic registers

### AUD_PDEC Post decoder control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x62 | Reserved | | DEM | DCF | Reserved | | | PL |

Address: *AudioBaseAddress* + 0x62
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register controls the post decoder operations.

| Bitfield | Description |
|---|---|
| PL | When high Pro Logic decoding is forced, when low the PL decoder is activated only if the output of the previous decoding stage is Pro Logic encoded. |
| DCF | When high the DC filter is activated. |
| DEM | When high the de-emphasis filter is activated. |
| Reserved | Set to zero |

### AUD_PL_AB Pro logic auto balance

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x64 | | | | | | | | |

Address: *AudioBaseAddress* + 0x64
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register enables the auto balance function of the pro-logic decoder. The default value is zero (auto balance off).

### AUD_PL_DWNX Pro logic decoder downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x65 | | | | | | | | |

Address: *AudioBaseAddress* + 0x65
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This value in this register controls the function of the Pro Logic Downmix.

| Value | Comment |
|---|---|
| 0, 1, 2 | Pro Logic is disabled |
| 3 | 3/0 (L, C, R) three stereo |
| 4 | 2/1 (L, R, S) phantom |
| 5 | 3/1 (L, C, R, S) |
| 6 | 2/2 (L, R, S, S) phantom |
| 7 | 3/2 (L, C, R, S, S) |

Phantom mode means that the center is not used.

## AUD_DWSMODE Downsampling filter

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x70 | | | | | | | | |

Address: *AudioBaseAddress* + 0x70
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register controls the downsampling filter. When decoding a 96kHz DVD-LPCM stream, it might be necessary to downsample the stream to 48kHz***.***

| Value | Mode |
|---|---|
| 0 | Automatic (according to bitstream) |
| 1 | Force Downsampling |
| 2 | Suppress Downsampling |

## 4.13 Bass redirection registers

### AUD_VOLUME0 Volume of first channel

Address: 0x4E
Type: RWS
SW Reset: 0
HW Reset: UND

***Description***

This register reads or writes the attenuation that is applied to the channel selected by CHAN_IDX. The attenuation is kdB where k is the contents of the register.

- If CHAN_IDX = 0, then VOLUME0 can be written with the attenuation that will be applied to Left channel.
- If CHAN_IDX = 1, then VOLUME0 can be written with the attenuation that will be applied to Center channel.
- If CHAN_IDX = 2, then VOLUME0 can be written with the attenuation that will be applied to Left Surround channel.
- If CHAN_IDX = 5, then reading VOLUME0 provides the attenuation that is applied to Left channel.
- If CHAN_IDX = 6, then reading VOLUME0 provides the attenuation that is applied to Center channel.
- If CHAN_IDX = 7, then reading VOLUME0 provides the attenuation that is applied to Left Surround channel.
- For other values of CHAN_IDX, then the contains of this register is meaningless.

### AUD_VOLUME1 Volume of second channel

Address: 0x63
Type: RWS
SW Reset: 0
HW Reset: UND

***Description***

This register reads or writes the attenuation that is applied to the channel selected by CHAN_IDX. The attenuation is -kdB where k is the contents of the register.

- If CHAN_IDX = 0, then VOLUME1 can be written with the attenuation that will be applied to Right channel.
- If CHAN_IDX = 1, then VOLUME1 can be written with the attenuation that will be applied to Subwoofer channel.
- If CHAN_IDX = 2, then VOLUME1 can be written with the attenuation that will be applied to Right Surround channel.
- If CHAN_IDX = 5, then reading VOLUME1 provides the attenuation that is applied to Right channel.
- If CHAN_IDX = 6, then reading VOLUME1 provides the attenuation that is applied to Subwoofer channel.
- If CHAN_IDX = 7, then reading VOLUME1 provides the attenuation that is applied to Right Surround channel.
- For other values of CHAN_IDX, then the contents of this register are meaningless.

## AUD_OCFG Output configuration

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x66 | LFE | | | | | OCFG_num | | |

Address: *AudioBaseAddress* + 0x66
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register specifies the bass redirection scheme (see Dolby specifications for further information on this scheme). In this table, LP means low-pass filter, HP means high-pass filter.

| Bitfield | Description |
|---|---|
| OCFG_num | Configuration number for bass management. For OCFG_num equal to 2 or 3, the subwoofer can be output if bit LFE = 1. For all other values of OCFG_num, bit LFE has no effect<br>000: ALL, All channels are rounded according to the selected output precision, (24b -> 16b, 24 -> 18b.) and scaled (volume control) only.<br>001: LSW, Low frequencies are extracted from the six input channels and redirected to the subwoofer.<br>SUB = LP (L + R + LS + Rs + C + LFE)<br>Low frequencies are removed from all channels<br>L = HP (L) R = HP (R) C = HP (C)<br>LS = HP (LS) RS = HP (RS)<br>010: LLR, Low frequencies are extracted from C, LFE, LS and RS channels and redirected to left and right channels if subwoofer is not output or to the subwoofer.<br>C =HP (C), Rs = HP (Rs), Ls = HP (Ls)<br>if subwoofer is output, SUB = LP(LFE + C + Ls+ Rs), L = L, R = R<br>if subwoofer is not output, L = L + LP (C + LFE + LS + RS), R = R + LP (C + LFE + LS + RS)<br>011: SLP, Low frequencies are redirected to the left, right and surround channels or can be output on the subwoofer.<br>if subwoofer is output, SUB = LFE, L = L + LP(C), R = R + LP(C), Ls = Ls, Rs = Rs<br>if subwoofer is not output, L = L + LP(C) + LFE, R = R + LP(C) + LFE, Ls = Ls + LFE, Rs = Rs + LFE.<br>101: BYP, All channels are directly routed to PCM outputs. |
| LFE | 0: Subwoofer is not output<br>1: Subwoofer is output |

## AUD_CHAN_IDX Channel Index

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x67 | Reserved | | | | | CHAN_IDX | | |

Address: 0x67
Type: R/W
SW Reset: 4
HW Reset: UND

***Description***

This register identifies the pair of channels and the type of access:

| Bitfield | value | Channel pair | Access | comment |
|---|---|---|---|---|
| CHAN_IDX | 0 | Left and Right | write | |
| | 1 | Center and Subwoofer | write | |
| | 2 | Left surround and right surround | write | |
| | 3 | not used | not used | |
| | 4 | no pair selected | none | Indicates that volume can be read or written |
| | 5 | Left and Right | read | |
| | 6 | Center and Subwoofer | read | |
| | 7 | Left surround and right surround | read | |

- To read a volume, the register AUD_CHAN_IDX must be set to the appropriate value. The DSP indicates that the attenuation is readable through registers AUD_VOLUME0 and AUD_VOLUME1 by changing automatically the AUD_CHAN_IDX to value 4.
- To write a volume, the attenuation of the pair of channel should be written in AUD_VOLUME0 and AUD_VOLUME1 registers. Then the AUD_CHAN_IDX register is written to the appropriate value. The attenuation is updated on the next audio block and AUD_CHAN_IDX value is automatically changed to 4.

## 4.14 Dolby Digital configuration registers

### AUD_AC3_DECODE_LFEDecode LFE

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x68 | | | | | | | | |

Address: *AudioBaseAddress* + 0x68
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

When this register is set to 1, the device decodes LFE channel (if present).

### AUD_AC3_COMP_MOD Compression mode

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x69 | | | | | | | | |

Address: *AudioBaseAddress* + 0x69
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

The value of this register defines the compression mode. In custom A mode, the dialog normalization function is not done by the audio decoder, it has to be done by an external analog part. In all other modes the normalization is done by audio decoder.

| Value | Meaning |
|---|---|
| 0 | Custom A (Analog) |
| 1 | Custom D (Digital) |
| 2 | Line Out |
| 3 | RF Mode |

### AUD_AC3_HDR High dynamic range

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6A | | | | | | | | |

Address: *AudioBaseAddress* + 0x6A
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register corresponds to the Dynamic range scale factor for high level signals, also called cut factor in the Dolby specifications.

HDR = 255 * Cut Factor (in decimal), where the cut factor is a fractional number between 0 and 1. It is used to scale the dynamic range control word for high-level signals that would otherwise tend to be reduced.

When HDR = 0xff (cut factor = 1.0), the high level signals reduction is the one given in the stream. A value of zero disables the high-level compression. This word is ignored if the compression mode is set to RF mode.

## AUD_AC3_LDR Low dynamic range

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6B | | | | | | | | |

Address: *AudioBaseAddress* + 0x6B
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register corresponds to the Dynamic range scale factor for low level signals, also called boost factor in the Dolby specifications.

LDR = 255 * BoostFactor (in decimal), where the boost factor is a fractional number between 0 and 1.0.

The boost factor scales the dynamic range control-word for low-level signals that would otherwise tend to be amplified. When LDR = 0xff (boost factor = 1.0), and the low level signals amplification is maximum. A value of zero disables the low-level amplification. This word is ignored if the compression mode is set to RF mode.

## AUD_AC3_RPC Repeat count

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6C | | | | | | | | |

Address: *AudioBaseAddress* + 0x6C
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

When a CRC error is detected, previous blocks can be repeated or muted. This register specifies the number of audio blocks to repeat before muting. If this is zero, then blocks are muted until the next frame is decoded

## AUD_AC3_KARAMODE Karaoke downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6D | | | | | | | | |

Address: *AudioBaseAddress* + 0x6D
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

Downmix mode when a karaoke bit stream is received. A Karaoke bitstream can be composed of 5 channels, which are: L(left), R(right), M(Music), V1(Vocal 1), V2(Vocal 2). There are two major modes when receiving a Karaoke bitstream: aware and capable.

- When in 'aware' mode (AUD_KARAMODE = 0), a predefined downmix is applied on all incoming channels.
- When in 'capable' mode (AUD_KARAMODE = 4, 5, 6, 7), the user can choose to reproduce or not the two incoming vocal channels, V1 and V2.

An additional mode is added (AUD_KARAMODE = 3) to allow multi-channel reproduction. In this case, the downmix specified by the AUD_DOWNMIX and AUD_DUALMODE registers is applied. The following table summaries the different modes:

| Value | Mode | Comment |
|---|---|---|
| 0 | Aware | Left = L + clev*M + slev*V1, Right = R + clev*M + slev*V2 |
| 1 | | Not used |
| 2 | | Not used |
| 3 | Multichannel | Consider bitstream as multi-channel:<br>Perform downmix according to DOWNMIX and DUALMODE registers |
| 4 | Capable | Do not reproduce V1, V2: Left = L + clev*M, Right = R + clev*M |
| 5 | | Reproduction V1 only: Left = L + clev*M + 0.707*V1, Right = R + clev*M + 0.707V1 |
| 6 | | Reproduction V2 only: Left = L + clev*M + 0.707*V2, Right = R + clev*M + 0.707V2 |
| 7 | | Reproduction V1, V2: Left = L + clev*M + V1, Right = R + clev*M + V2 |

Left = Output Channel,
Right = Output Channel, L, R, M, V1, V2 = Input Channels (coded in Dolby Digital karaoke bitstream),
clev = Center Mix Level (value provided in the bitstream),
slev = Surround Mix Level (value provided in the bitstream). For further information ref. to annex C of ATSC standard "Digital Audio Compression (AC-3)".

## AUD_AC3_DUALMODE Dual downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6E | | | | | | | | |

Address: *AudioBaseAddress* + 0x6E
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

This register allows additional downmix to be set when in 2/0 output mode or when receiving a "Dual mode" incoming bitstream (example: A disk with 2 different languages on channel 1 and channel 2). In the following table, channel 1 and 2 represent the output channels after downmix performed with AUD_AC3_DOWNMIX.

This register enables Mono downmix when AUD_DOWNMIX = 2 and AUD_DUALMODE = 3.

| Value | Comment |
|---|---|
| 0 | Output as Stereo |
| 1 | Output Channel 1 on both output L/R |
| 2 | Output Channel 2 on both output L/R |
| 3 | Mix Channel 1 and 2 to monophonic and output on both L/R |

## AUD_AC3_DOWNMIX Downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6F | | | | | | | | |

Address: *AudioBaseAddress* + 0x6F
Type: R/W
SW Reset: NC
HW Reset: UND

***Description.***

| Value | Comment |
|---|---|
| 0 | 2/0 Dolby Surround (LT, RT) |
| 1 | 1/0 (C) |
| 2 | 2/0 (L, R) |
| 3 | 3/0 (L, C, R) |
| 4 | (L, R, S) |
| 5 | 3/1 (L, C, R, S) |
| 6 | 2/2 (L, R, LS, RS - Dolby Phantom Mode |
| 7 | 3/2 (L, C, R, $L_S$, $R_S$) |

*Note In notation, 3/2 represents 3 front speakers and 2 surround speakers.*

## AUD_AC3_STATUS0 Dolby Digital status register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | Not used | | fs_cod | | | Bitrate code | | |

Address: *AudioBaseAddress* + 0x76
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Bitrate code | Code identifying the bitrate. Bitrate[4..0] = frmsizecod[5..1] |
| fs_cod | Code identifying the sampling frequency |

## AUD_AC3_STATUS1 Dolby Digital status register 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x77 | Reserved | | | | LFE | Acmod | | |

Address: *AudioBaseAddress* + 0x77
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Acmod | Audio coding mode. Indicates which channels are in use. |
| LFE | Indicates if LFe channel is present in the stream |

## AUD_AC3_STATUS2 Dolby Digital status register 2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x78 | Bsmod | | | Bsid | | | | |

Address: *AudioBaseAddress* + 0x78
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Bsid | Bit stream identification, indicates the version of the standard |
| Bsmod | Bbit stream mode, indicates the type of service |

## AUD_AC3_STATUS3 Dolby Digital status register 3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x79 | Reserved | | | | Cmixlevel | | SurMixlevel | |

Address: *AudioBaseAddress* + 0x79
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Cmixlevel | Downmix level of center channel |
| SurMixlevel | Downmix level of surround channel |

## AUD_AC3_STATUS4 Dolby Digital status register 4

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7A | Reserved | | | Dsurmod | | Copyright | Origbs | Lancode |

Address: *AudioBaseAddress* + 0x7A
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Lancode | When at 1, indicates that a language code is provided in the stream |
| Origbs | When at 1, indicates that the stream is an original |
| Copyright | When at 1, indicates that the stream is protected by copyright |
| Dsurmod | In 2/0 mode, indicates if the stream is Dolby surround encoded |

## AUD_AC3_STATUS5 Dolby Digital status register 5

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7B | Lancode | | | | | | | |

Address: *AudioBaseAddress* + 0x7B
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains the code of the language of the audio service, extracted from the stream.

## AUD_AC3_STATUS6 Dolby Digital status register 6

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7C | Reserved | | | Dialog Normalization (see Dolby specifications) | | | | |

Address: *AudioBaseAddress* + 0x7C
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains the code indicating the dialog normalization level extracted from the stream.

## AUD_AC3_STATUS7 Dolby Digital status register 7

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7D | Room type | | | Mix level | | | | Audprodie |

Address: *AudioBaseAddress* + 0x7D
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This register contains bit stream information extracted from the stream.

| Bitfield | Description |
|---|---|
| Audprodie | Audprodie: if set, indicates that room type and mix level are provided |
| Mix level | If audprodie is set, mix level indicates the sound level |
| Room type | If audprodie is set, mix level indicates the sound level |

## 4.15 MPEG configuration registers

### AUD_MP_SKIP_LFE Channel skip

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x68 | Reserved | | | | | | | |

Address: *AudioBaseAddress* + 0x68
Type: R/W
SW Reset: 0x00
HW Reset: UND

***Description***

When this register is set to 1, the LFE channel is skipped. When this register is set to 0 the LFE channel is decoded (if present).

### AUD_MP_PROG_NUMBERProgram number

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x69 | Reserved | | | | | | | Prog |

Address: *AudioBaseAddress* + 0x69
Type: R/W
SW Reset: 0x00
HW Reset: UND

***Description***

When the stream is in Second Stereo mode, this register specifies which program is played.

| Bitfield | Description |
|---|---|
| Prog | Select program #0 or #1 where 0: L0,R0 in front channels, 1: L2,R2 in front channels |

### AUD_MP_DUALMODE MPEG setup dual mode

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6E | | | | | | | | |

Address: *AudioBaseAddress* + 0x6E
Type: R/W
SW Reset: 0x00
HW Reset: UND

***Description***

The MPEG DUAL_MODE is active in downmix mode 1 and 9.

| Value | Comment |
|---|---|
| 0 | Output as Stereo |
| 1 | Output Channel 1 on both outputs L/R |
| 2 | Output Channel 2 on both outputs L/R |
| 3 | Mix Channel 1 and 2 to monophonic, and output on both L/R |

## AUD_MP_DRC Dynamic range control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6A | | | | | | | | DRC |

Address: *AudioBaseAddress* + 0x6A
Type: R/W
SW Reset: 0x00
HW Reset: UND

***Description***

When bit DRC=1, dynamic range control is enabled. The dynamic range is set according to the data transmitted in the DVD MPEG stream.

## AUD_MP_CRC_OFF CRC_CHECK_OFF

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6C | | | | | | | | |

Address: *AudioBaseAddress* + 0x6C
Type: R/W
SW Reset: NC
HW Reset: UND

When register is set to 1, the CRC in MPEG frame is not checked. When register is set to 0, the CRC in MPEG frame is checked if exists. If a CRC error occurs, the decoder soft mutes the frame (but does note stop).

## AUD_MP_MC_OFF Multi-channel

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6D | Reserved | | | DEN | Reserved | | | MC |

Address: *AudioBaseAddress* + 0x6D
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| MC | When MC=1, the multi-channel part of the bitstream is not decoded, only the MPEG-1 compatible bitstream is decoded. Bit MC must be set to 1 for an MPEG-1 bitstream. |
| DEN | De-normalization: set DEN=0 for MPEG1 signals, and set DEN=1 for MPEG2 multi-channel signals<br>When DEN=1, MPEG2 multi-channel signals L, C, R, LS and RS can be de-normalized. The signals must first be inverse-weighted then multiplied by the de-normalization factor. This undoes the attenuation carried out at the encoder side to avoid overload when calculating the compatible signals (see MPEG 13818-3 specifications). |

## AUD_MP_DOWNMIX MPEG downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6F | | | | | | | | |

Address: *AudioBaseAddress* + 0x6F
Type: R/W
SW Reset: 0x08
HW Reset: UND

***Description***

In the table below, $L_O$, $R_O$, $C_O$, $Ls_O$, $Rs_O$ represent the output channels after downmix, and L, R, C, $L_S$, $R_S$ are the audio channels.

The coefficients $K_j$, $K_C$, $K_r$, $K_S$, depend on the number of input channels. In the above table, the equations are given for a 5 channels input bitstream. If the input bitstream does not contain five channels (L, C, R, $L_S$, $R_S$), the coefficient “Kj” corresponding to the channel not present is equal to 0. If the MPEG bitstream contains only one surround channel (S), replace ($K_S$ x ($L_S$ + $R_S$)), ($K_S$ x $L_S$ and ($K_S$ x $R_S$) by ($K_S$ x S) in the above equations.

| Value | Output Mode | Comment |
|---|---|---|
| 0x00 | 1/0 (C) = Mono | $C_O = K_j$ x L + C + $K_r$ x R + $K_S$ ($L_S$ + $R_S$) |
| 0x01 | 2/0 (L, R) = Stereo | $L_O = (L + K_C \times C + K_S \times L_S)/(1 + K_C + K_S)$,<br>$R_O = (R + K_C \times C + K_S \times R_S)/(1 + K_C + K_S)$ |
| 0x02 | 3/0 (L, C, R) | $L_O = L + K_S \times L_S$, $R_O = R + K_S \times R_S$, $C_O = C$ |
| 0x03 | 2/1 (L, R, S) | $L_O = L + K_C \times C$, $R_O = R + K_C \times C$, $Ls_O = Rs_O = K_S \times (L_S + R_S)$ |
| 0x04 | 3/1 (L, C, R, S) | $L_O = L$, $R_O = R$, $C_O = C$, $Ls_O = Rs_O = K_S \times (L_S + R_S)$ |
| 0x05 | 2/2 (L, R, $L_S$, $R_S$) | $L_O = L + K_C \times C$, $R_O = R + K_C \times C$, $Ls_O = L_S$, $Rs_O = R_S$ |
| 0x06 | 3/2 (L, C, R, $L_S$, $R_S$) | $L_O = L$, $R_O = R$, $C_O = C$, $Ls_O = L_S$, $Rs_O = R_S$ |
| 0x09 | 2/0 (Dolby surround $L_T$, $R_T$) | $L_T = (L + 0.707C - 0.707 \times 0.5 (LS + R_S))$ /2.414,<br>$R_T = (R + 0.707C + 0.707 \times 0.5 (L_S + R_S))$ /2.414 |
| 0x0A | 2/0 Karaoke capable: V1 ON, V2 ON | Lk = L + 0.707 A1 + 0.707 G, Rk = R + 0.707 A2 + 0.707 G |
| 0x0B | 2/0 Karaoke Capable: V1 ON, V2 OFF | Lk = L + 0.707 A1 + 0.707 G, Rk = R + 0.707 G |

| | | |
|---|---|---|
| 0x0C | 2/0 Karaoke Capable: V1 OFF, V2 ON | Lk = L + 0.707 G, Rk = R + 0.707 A2 + 0.707 G |
| 0x0D | 2/0 Karaoke Capable: V1 OFF, V2 OFF | Lk = L + 0.707 G, Rk = R + 0.707 G |
| 0x0E | 2/0 Karaoke Capable: V1 ON, V2 OFF (Dolby Digital like) | Lk = L + 0.707 A1 + 0.707 G, Rk = R + 0.707 A1 + 0.707 G |
| 0x0F | 2/0 Karaoke Capable: V1 OFF, V2 ON (Dolby Digital like) | Lk = L + 0.707 A2 + 0.707 G, Rk = R + 0.707 A2 + 0.707 G |
| 0x1A | 3/0 Karaoke Capable: V1 ON, V2 ON | Lk = L + 0.707 A1, Ck = G, Rk = R + 0.707 A2 |
| 0x1B | 3/0 Karaoke Capable: V1 ON, V2 OFF | Lk = L + 0.707 A1, Ck = G, Rk = R |
| 0x1C | 3/0 Karaoke Capable: V1 OFF, V2 ON | Lk = L, Ck = G, Rk = R + 0.707 A2 |
| 0x1D | 3/0 Karaoke Capable: V1 OFF, V2 OFF | Lk = L, Ck = G, Rk = R |
| 0x1E | 3/0 Karaoke Capable: V1 ON, V2 OFF (Dolby Digital like) | Lk = L, Ck = G + A1, Rk = R |
| 0x1F | 3/0 Karaoke Capable: V1 OFF, V2 ON (Dolby Digital like) | Lk = L, Ck = G + A2, Rk = R |

## AUD_MP_STATUS0 MPEG status register 0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | ID | LAY[1:0] | | P | BRI[3:0] | | | |

Address: *AudioBaseAddress* + 0x76
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| BRI[3:0] | Bit rate index |
| P | Protection Bit |
| LAY[1:0] | Layer |
| ID | Identifier |

## AUD_MP_STATUS1 MPEG status register 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x77 | SFR[1:0] | | PAD | PRI | MOD[1:0] | | MEX[1:0] | |

Address: *AudioBaseAddress* + 0x77
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| MEX[1:0] | Mode Extension |
| MOD[1:0] | Mode |
| PRI | Private Bit |
| PAD | Padding Bit |
| SFR[1:0] | Sampling Frequency |

## AUD_MP_STATUS2 MPEG status register 2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x78 | not used | | | | C | OCB | EMP[1:0] | |

Address: *AudioBaseAddress* + 0x78
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| EMP[1:0] | Emphasis rate index |
| OCB | Original/Copy Bit |
| C | Copyright |

## AUD_MP_STATUS3 MPEG status register 3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x79 | CEN[1:0] | | SUR[1:0] | | LFE | AMX | DEM[1:0] | |

Address: *AudioBaseAddress* + 0x79
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| DEM[1:0] | Dematrix procedure |
| AMX | Audio mix |
| LFE | LFE |
| SUR[1:0] | Surround |
| CEN[1:0] | Centre |

## AUD_MP_STATUS4 MPEG status register 4

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7A | EXT | NML[2:0] | | | MFS | MLY | CIB | CIS |

Address: *AudioBaseAddress* + 0x7A
Type: RO
SW Reset: UND

HW Reset: UND

| Bitfield | Description |
|---|---|
| CIS | Copyright ID Start |
| CIB | Copyright ID Bit |
| MLY | Multi-lingual Layer |
| MFS | Multi-lingual FS |
| NML[2:0] | Number of Multi-lingual Channels |
| EXT | Extension bitstream present |

### AUD_MP_STATUS5 MPEG status register 5

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7B | | | | | | | | |

Address: *AudioBaseAddress* + 0x7B
Type: RO
SW Reset: UND
HW Reset: UND

***Description***

The number of extended ancillary data bytes is contained in this register

## 4.16 Pink noise generation registers

### AUD_PN_DOWNMIX Pink noise downmix

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6F | | | RS | LS | LFE | C | R | L |

Address: *AudioBaseAddress* + 0x6F
Type: R/W
SW Reset: NC
HW Reset: UND

***Description***

| Bitfield | Description |
|---|---|
| L | 1: Left channel contains pink noise<br>0: Left channel is forced to zero |
| R | 1: Right channel contains pink noise<br>0: Right channel is forced to zero |
| C | 1: Center channel contains pink noise<br>0: Center channel is forced to zero |
| LFE | 1: LFE channel contains pink noise<br>0: LFE channel is forced to zero |
| LS | 1: Left surround channel contain pink noise<br>0: Left surround channel is forced to zero |
| RS | 1: Right surround channel contains pink zero<br>0: Right surround channel is forced to zero |

After this processing, the AUD_OCFG stage is applied on these channels. AUD_OCFG must be configured to 0 and attenuation on all channels must be set to 10dB attenuation.

The other values must not be used because low frequency extraction must not be done when generating pink noise.

Pink noise selection is made through the AUD_STREAMSEL and AUD_DECODSEL registers.

## 4.17 DTS

### AUD_DTS_STATUS0 DTS status 0 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | NBLKS | | | | | | | |

Address: 0x76
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This status register contains the number of PCM samples in the stream. The value is extracted from the stream, and can be read after each BOF interrupt.

| Bitfield | Description |
|---|---|
| NBLKS | The number of PCM samples is: (NBLKS+1) * 32 |

### AUD_DTS_STATUS1 DTS status 1 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x77 | Reserved | | AMODE | | | | | |

Address: 0x77
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This status register contains the arrangement mode of the audio channels. It defines the number and type of conveyed channels. The value AMODE is extracted from the incoming stream and can be read after each BOF interrupt.

| AMODE value | N° of channels | Arrangement |
|---|---|---|
| 0 | 1 | A |
| 1 | 2 | A + B (dual mono) |
| 2 | 2 | L + R (stereo) |
| 3 | 2 | (L+R) + (L-R) |
| 4 | 2 | LT + RT |
| 5 | 3 | C + L + R |
| 6 | 3 | L + R + S |
| 7 | 4 | C + L + R + S |
| 8 | 4 | L + R + SL + SR |

| AMODE value | N° of channels | Arrangement |
|---|---|---|
| 9 | 5 | C + R + L + SL + SR |
| 10 | 6 | CL + CR + L + R + SL + SR |
| 11 | 6 | C + L + R + LR + RR + OV |
| 12 | 6 | CF + CR + LF + RF + LR + RR |
| 13 | 7 | CL + C + CR + L + R + SL + SR |
| 14 | 8 | CL + CR + L + R + SL1 + SL2 + SR1 + SR2 |
| 15 | 8 | CL + C + CR + L + R + SL + S + SR |
| 16/63 | - | user defined |

The following abbreviations have been used in the table above:

L: left, R: Right, C: Center, S: Surround, F: Front, R: Rear, T: Total, OV: overhead, A: channel1 (when in dual mono mode), B: channel2. I.e.: CR = Center_right, RR = Rear_right, SL1 = Surround_Left1, CF = Center_Front.

## AUD_DTS_STATUS2 DTS status 2 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x78 | Reserved | | | | SFREQ | | | |

Address: 0x78
Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This status register contains a code corresponding to the sampling frequency of the stream. The value SFREQ is extracted from the incoming stream and can be read after each BOF interrupt.

| SFREQ value | Source sampling frequency | SFREQ value | Source sampling frequency |
|---|---|---|---|
| 0 | invalid | 8 | 44.1 kHz |
| 1 | 8 kHz | 9 | 88.2 kHz |
| 2 | 16 kHz | 10 | 176.4 kHz |
| 3 | 32 kHz | 11 | 12 kHz |
| 4 | 64 kHz | 12 | 24 kHz |
| 5 | 128 kHz | 13 | 48 kHz |
| 6 | 11.025 kHz | 14 | 96 kHz |
| 7 | 22.05 kHz | 15 | 192 kHz |

## AUD_DTS_STATUS3 Status 3 register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x79 | Reserved | | | | RATE | | | |

Address: 0x79

Type: RO
SW Reset: NC
HW Reset: UND

***Description***

This status register contains a code corresponding to the transmission bit rate of the stream. The value RATE is extracted from the incoming stream and can be read after each BOF interrupt.

| Rate value | Transmission bitrate | Rate value | Transmission bitrate | Rate value | Transmission bitrate | Rate value | Transmission bitrate |
|---|---|---|---|---|---|---|---|
| 0 | 32 kbps | 8 | 256 kbps | 16 | 960 kbps | 24 | 1536 kbps |
| 1 | 56 kbps | 9 | 320 kbps | 17 | 1024 kbps | 25 | 1920 kbps |
| 2 | 64 kbps | 10 | 384 kbps | 18 | 1152 kbps | 26 | 2048 kbps |
| 3 | 96 kbps | 11 | 448 kbps | 19 | 1280 kbps | 27 | 3072 kbps |
| 4 | 112 kbps | 12 | 512 kbps | 20 | 1344 kbps | 28 | 3840 kbps |
| 5 | 128 kbps | 13 | 576 kbps | 21 | 1408 kbps | 29 | open |
| 6 | 192 kbps | 14 | 640 kbps | 22 | 1411.2 kbps | 30 | variable |
| 7 | 224 kbps | 15 | 768 kbps | 23 | 1472 kbps | 31 | lossless |

## 4.18 PCM beep-tone registers

### AUD_PCM_BTONE PCM beep tone frequency

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x68 | | | | | | | | |

Address: *AudioBaseAddress* + 0x68
Type: R/W
HW Reset: 0
SW Reset: UND

***Description***

The value in this register sets the PCM beep tone frequency according to the formula:

$$\text{Beep\_tone frequency} = (F_s/2)/(\text{Register\_value} + 1)$$

## 4.19 Audio trick-mode register

### AUD_TM_SPEED Audio trick-mode speed

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x57 | SPEED | | | | | | | |

Address: 0x57
Type: R/W
HW Reset: 0
SW Reset: UND

***Description***

This register sets the audio trick-mode speed according to the table below. After setting this register, the register bit AUD_UPDATE.TM must be set to 1 to update the DSP with the new values. After the update, AUD_UPDATE.TM is automatically reset to zero.

| Bitfield | Description |
|---|---|
| SPEED | 00000000: normal speed (no audio trick mode)<br>00000001: slow forward (two times slower)<br>00000010: very slow forward (three times slower)<br>10000000: fast forward (two times faster)<br>01000000:very fast forward (three times faster)<br>All other bit combinations are reserved. |

## AUD_LPCM_STATUS0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | EMPH_FLAG | MUTE_FLAG | Reserved | FRAME_NUM | | | | |

Address: 0x76
Type: R/W
Reset value: UND

***Description***

| Bitfield | Description |
|---|---|
| FRAME_NUM | frame number of the first access unit in the group of audio frames |
| Reserved | Set to 0 |
| MUTE_FLAG | 0: mute off, 1: mute on |
| EMPH_FLAG | Emphasis status after the first access unit: 0: emphasis off; 1: emphasis on |

## AUD_LPCM_STATUS1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | Word_Length | | Samp_Freq | | Reserved | Aud_Channels | | |

Address: 0x77
Type: R/W
Reset value: UND

***Description***

| Bitfield | Description |
|---|---|
| Aud_Channels | Number of audio channels:000=1 channel (mono), 001=2 channels (stereo), 010=3 channels, 011=4 channels, 100=5 channels,101=6 channels, 110=7 channels, 111=8 channels |
| Reserved | Set to 0 |
| Samp_Freq | Sampling frequency: 00=48kHz, 01=96kHz, 10=reserved, 11=reserved |
| Word_Length | Audio sample length: 00=16 bits, 01=20 bits, 10=24 bits, 11=reserved |

## AUD_LPCM_STATUS2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | Dyn_Range_Control | | | | | | | |

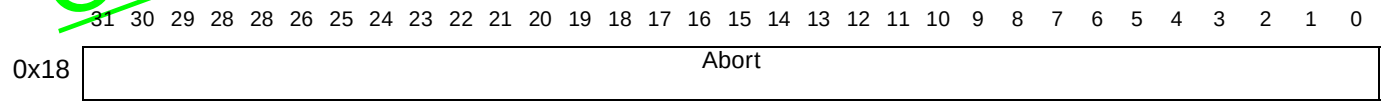
Address: 0x78
Type: RW
Reset value: UND

***Description***

This register sets the dynamic range compression from the first access unit. For the hexadecimal value 0x80, dynamic range control is not set. For all other values, the dynamic range control is (24.082 - 6.0206 * X - 0.2007 * Y)dB, where X = dynamic_range_control[7..5] and Y = dynamic_range_control[4..0].

# 5 Block move DMA (BMDMA)

## BMDMA_ABORT Block move DMA abort

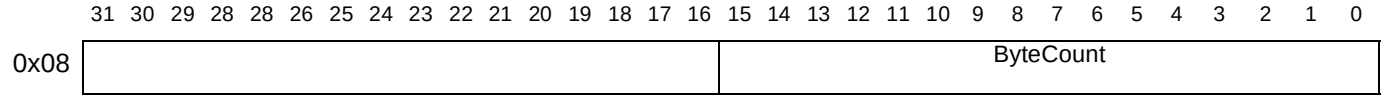



Address: *BMBaseAddress* + 0x18
Type: Write only

***Description***

Any write to this register aborts the current block move operation. The abort may take a few system clock cycles to complete depending on the access time of the memory and the Active bit in BMDMA_Status should be polled to check that the DMA has been aborted.

## BMDMA_COUNT Block move DMA counts

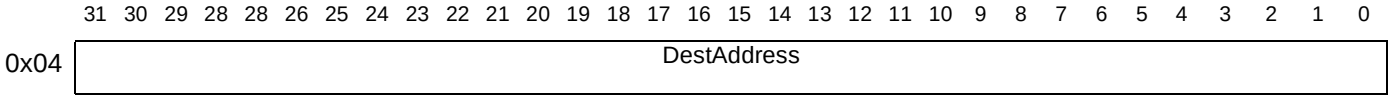



Address: *BMBaseAddress* + 0x08
Type: Write only

***Description***

The number of bytes to block move. Writing to this register starts the block move.

## BMDMA_DESTADDRESSBlock move DMA destination address

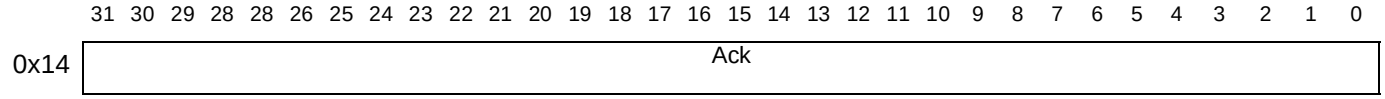



Address: *BMBaseAddress* + 0x04
Type: Write only

***Description***

The address of the base of the block move destination area.

## BMDMA_INTACK Block move DMA interrupt acknowledge

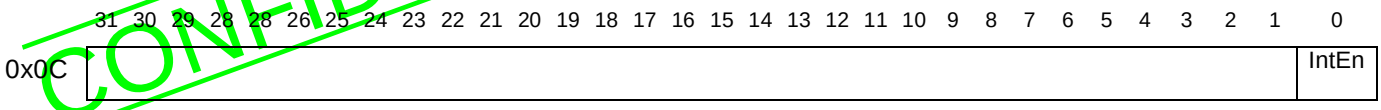



Address: *BMBaseAddress* + 0x14
Type: Write only

***Description***

Any write to this register acknowledges the interrupt and resets the Interrupt bit in the BMDMA_Status register.

## BMDMA_INTEN Block move DMA interrupt enable

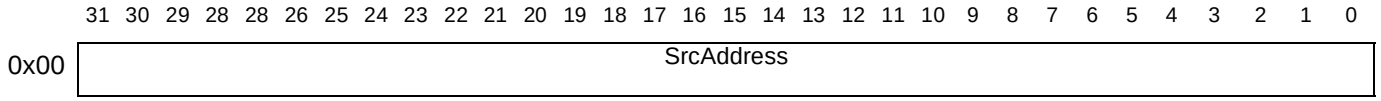



Address: *BMBaseAddress* + 0x0C
Type: Read/write

***Description***

Interrupt enable; enabled when set to 1, masked when set to 0.

## BMDMA_SRCADDRESS Block move DMA source address

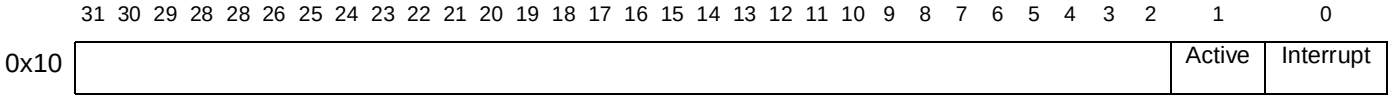



Address: *BMBaseAddress* + 0x00
Type: Write only

***Description***

The address of the base of the block move source data.

## BMDMA_STATUS Block move DMA status




Address: *BMBaseAddress* + 0x10
Type: Read only

***Description***

| Bitfield | Description |
|---|---|
| Interrupt | Interrupt pending when set to 1. |
| Active | Block move in progress when this bit is 1. |
| Reserved | Read data undefined. |

# 6 Cache control (CAC)

## CAC_CACHECONTROL0 Cache control 0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x000 | Cacheable7 | Cacheable6 | Cacheable5 | Cacheable4 | Cacheable3 | Cacheable2 | Cacheable1 | Cacheable0 |

Address: *CacheBaseAddress* + 0x000
Type: Read/write
Reset value: 0

***Description***

This register defines the data cacheability of part of region 1, i.e. whether the data cache will be used when accessing data at addresses in the range 0xC0000000 to 0xC007FFFF.

Each bit in the register defines whether a certain 64 Kbyte block of addresses will be cached. If the appropriate bit is set then the block will be cached; otherwise it will not be cached.

| Bitfield | Block start | Block end |
|---|---|---|
| Cacheable0 | 0xC0000000 | 0xC000FFFF |
| Cacheable1 | 0xC0010000 | 0xC001FFFF |
| Cacheable2 | 0xC0020000 | 0xC002FFFF |
| Cacheable3 | 0xC0030000 | 0xC003FFFF |
| Cacheable4 | 0xC0040000 | 0xC004FFFF |
| Cacheable5 | 0xC0050000 | 0xC005FFFF |
| Cacheable6 | 0xC0060000 | 0xC006FFFF |
| Cacheable7 | 0xC0070000 | 0xC007FFFF |

## CAC_CACHECONTROL1 Cache control 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x100 | Cacheable7 | Cacheable6 | Cacheable5 | Cacheable4 | Cacheable3 | Cacheable2 | Cacheable1 | Cacheable0 |

Address: *CacheBaseAddress* + 0x100
Type: Read/write
Reset value: 0

***Description***

This register defines the data cacheability of part of region 1, i.e. whether the data cache will be used when accessing data in addresses in the range 0xC0200000 to 0xC027FFFF.

Each bit in the register defines whether a certain 64 Kbyte block of addresses will be cached. If the appropriate bit is set then the block will be cached; otherwise it will not be cached.

| Bitfield | Block start | Block end |
|---|---|---|
| Cacheable0 | 0xC0200000 | 0xC020FFFF |
| Cacheable1 | 0xC0210000 | 0xC021FFFF |
| Cacheable2 | 0xC0220000 | 0xC022FFFF |
| Cacheable3 | 0xC0230000 | 0xC023FFFF |
| Cacheable4 | 0xC0240000 | 0xC024FFFF |
| Cacheable5 | 0xC0250000 | 0xC025FFFF |
| Cacheable6 | 0xC0260000 | 0xC026FFFF |
| Cacheable7 | 0xC0270000 | 0xC027FFFF |

## CAC_CACHECONTROL2 Cache control 2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x200 | Cacheable7 | Cacheable6 | Cacheable5 | Cacheable4 | Cacheable3 | Cacheable2 | Cacheable1 | Cacheable0 |

Address: *CacheBaseAddress* + 0x200
Type: Read/write
Reset value: 0

***Description***

This register defines the data cacheability of part of region 3, i.e. whether the data cache will be used when accessing data in addresses in the range 0x40000000 to 0x4FFFFFFF, provided bit Cacheable0 of CacheControl3 is set.

If bit Cacheable0 of CacheControl3 is 0 then the address range 0x40000000 to 0x4007FFFF cannot be data cached. If this bit is set to 1 then each bit of CacheControl2 defines the data cacheability of one 64 Kbyte sub-block, as shown below. If the appropriate bit of CacheControl2 is set then the address range will be cached; otherwise it will not be cached.

| Bitfield | Block start | Block end |
|---|---|---|
| Cacheable0 | 0x40000000 | 0x4000FFFF |
| Cacheable1 | 0x40010000 | 0x4001FFFF |
| Cacheable2 | 0x40020000 | 0x4002FFFF |
| Cacheable3 | 0x40030000 | 0x4003FFFF |
| Cacheable4 | 0x40040000 | 0x4004FFFF |
| Cacheable5 | 0x40050000 | 0x4005FFFF |
| Cacheable6 | 0x40060000 | 0x4006FFFF |
| Cacheable7 | 0x40070000 | 0x4007FFFF |

## CAC_CACHECONTROL3 Cacheability of region 3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x300 | | | | | Cacheable3 | Cacheable2 | Cacheable1 | Cacheable0 |

Address: *CacheBaseAddress* + 0x300
Type: Read/write
Reset value: 0

***Description***

This register controls the data cacheability of the four EMI banks.

Clearing bit Cacheable0 of CacheControl3 to 0 will make EMI bank 0, the address range 0x40000000 to 0x4FFFFFFF, completely not cacheable by the data cache. If this bit is set to 1 then the bottom 512 Kbyte is controlled by CacheControl2 and the rest of the bank is fully cacheable.

Setting any of bits Cacheable3-1 of CacheControl3 to 1 will make EMI bank 3 to 1 respectively completely cacheable by the data cache; clearing these bits to 0 will make the corresponding bank not cacheable.

| Bitfield | EMI bank | Block start | Block end |
|---|---|---|---|
| **Cacheable**3 | Bank 3 | 0x70000000 | 0x7FFFFFFF |
| **Cacheable**2 | Bank 2 | 0x60000000 | 0x6FFFFFFF |
| **Cacheable**1 | Bank 1 | 0x50000000 | 0x5FFFFFFF |
| **Cacheable**0 | Bank 0 | 0x40000000 | 0x4FFFFFFF |

## CAC_CACHECONTROLLOCK Cache control lock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xA00 | | | | | | | | Lock |

Address: *CacheBaseAddress* + 0xA00
Type: Read/write
Reset value: 0

***Description***

The cache configuration can be locked by writing a 1 to the CacheControlLock register bit. The lock makes CacheControl, DCacheNotSRAM, EnableICache and CacheControlLock not writable. Reset of this flag is only performed by a hardware reset. This bit should be set to 1 after all the cache configuration registers have been written.

## CAC_CACHESTATUS Cache status register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x900 | | ICache Ready | DCache Ready | Flushing DCache | Invalidating ICache | Invalidating DCache | Enable ICache | DCache NotSRAM |

Address: *CacheBaseAddress* + 0x900
Type: Read only
Reset value: 0

***Description***

The **CacheStatus** register is read only, and shows the current state of the caches.

| Bitfield | Description |
|---|---|
| DCacheNotSRAM | Data cache or SRAM:<br>0: SRAM, 1: Data cache. |
| EnableICache | Instruction cache enabled:<br>0: Disabled, 1: Enabled. |
| InvalidatingDCache | Invalidating data cache:<br>0: Not invalidating, 1: Invalidating. |
| InvalidatingICache | Invalidating instruction cache:<br>0: Not invalidating, 1: Invalidating. |
| FlushingDCache | Flushing instruction cache:<br>0: Not flushing, 1: Flushing. |
| DCacheReady | Data cache ready:<br>0: The data cache is busy performing an operation.<br>1: The data cache is ready to perform another operation. |
| ICacheReady | Instruction cache ready:<br>0: The instruction cache is busy performing an operation.<br>1: The instruction cache is ready to perform another operation. |

## CAC_DCACHENOTSRAM Select data cache or SRAM

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x400 | | | | | | | | DCacheNotSRAM |

Address: *CacheBaseAddress* + 0x400
Type: Write only
Reset value: 0

***Description***

It is possible to select either data cache or an extra 2 Kbyte of on-chip SRAM. This is done by writing to the CAC_DCacheNotSRAM register. The default is to enable the extra on-chip SRAM. It is not recommended to change the selection other than during booting of the application. Set the DCacheNotSRAM bit to 1 to select data cache mode. The default value of 0 selects SRAM operation. Do not access locations 0x80001000 to 0x800017FF when using the data cache. The cache invalidate bit should be set before enabling the cache.

## CAC_ENABLEICACHE Enable the instruction cache

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x700 | | | | | | | | Enable |

Address: *CacheBaseAddress* + 0x700
Type: Write only
Reset value: 0

***Description***

The instruction cache can be selected by writing 1 to the CAC_EnableICache register; the default condition is no instruction cache. The instruction cache must be enabled before it is used. The cache should be invalidated immediately before enabling it.

## CAC_FLUSHDCACHE Flush the data cache

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x600 | | | | | | | | Flush |

Address: *CacheBaseAddress* + 0x600
Type: Write only

***Description***

Flushing the cache means forcing a write-back to memory of every dirty line in the cache. A dirty line is a line of cache that has been written to since it was loaded or last written back. Only the data cache can be flushed; the instruction cache never needs flushing since it is read only. To flush the data cache, set the CAC_FlushDCache register to 1. It is automatically reset to 0 on completion of the task. Any memory accesses that are cacheable which were started before the flush of the data cache is complete, will be blocked until it is completed. It is not recommended to change selection from data cache to SRAM during operation. However, if it is necessary to do so, it is essential to flush the cache to maintain memory integrity before making the change.

## CAC_INVALIDATEDCACHE Invalidate the data cache

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x500 | | | | | | | | Invalidate |

Address: *CacheBaseAddress* + 0x500
Type: Write only

***Description***

Invalidating a cache marks every line as not containing valid data. This is done by setting the CAC_InvalidateDCache register to 1. This register is automatically reset to 0 on completion of the task. Changing from SRAM to data cache should normally only be performed during the initialization stage of an application. However, if it is necessary to do so at other times, it is essential to invalidate the cache contents when making the change by setting the invalidate bit first and then enabling the cache. Any memory accesses that are cacheable which are started before the data cache invalidation is complete will be blocked until it is completed.

## CAC_INVALIDATEICACHE Invalidate the instruction cache

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x800 | | | | | | | | Invalidate |

Address: *CacheBaseAddress* + 0x800
Type: Write only

***Description***

Invalidating a cache marks every line as not containing valid data. This is done by setting the CAC_InvalidateICache register to 1. This register is automatically reset to 0 on completion of the task. Any instruction fetches that are cacheable and were started before completion of the invalidation of the instruction cache, will be blocked until it is completed. If the instruction cache is enabled, the cache contents will be random and must be invalidated by setting the invalidate bit first before enabling the cache.

# 7 Clock generator (CKG)

## 7.1 Introduction

The clock generator registers are mapped into the ST20 memory space (Video decoder register) following the formula:

@ST20 = ClockGeneratorBaseAddress + @A8=1 + @ clock_generator_register

All of the clock generator registers are unsynchronized; they change their value immediately.

## 7.2 Register map

| Register | Address | Bits | Access | Description |
|---|---|---|---|---|
| CKG_LPC_DIV | 0x000001CE | 8 | R/W | Low-power clock divider |
| CKG_PLL_CNT | 0x000001CF | 6 | R/W | Clock bypass PLL phase and PLL controller |
| CKG_CCAUD | 0x000001D0 | 3 | R/W | Clock controller for audio clock |
| CKG_DIVAUD | 0x000001D1 | 4 | R/W | Clock divider for audio clock |
| CKG_CCAUXDENC | 0x000001D2 | 3 | R/W | Auxiliary clock controller |
| CKG_DIVAUXDENC | 0x000001D3 | 4 | R/W | Denc Test clock divider |
| CKG_CCMCK | 0x000001D4 | 3 | R/W | Clock controller for MEMCLK |
| CKG_DIVMCK | 0x000001D5 | 4 | R/W | Clock divider for MEMCLK |
| CKG_SFREQSMC_SDIV | 0x000001D6 | 3 | R/W | SDIV reg: SMC frequency synthesizer |
| CKG_SFREQSMC_PE0 | 0x000001D7 | 8 | R/W | PE reg (LSB): SMC frequency synthesizer |
| CKG_SFREQSMC_PE1 | 0x000001D8 | 8 | R/W | PE reg (MSB): SMC frequency synthesizer |
| CKG_SFREQSMC_MD | 0x000001D9 | 5 | R/W | MD reg: SMC frequency synthesizer |
| CKG_SMC_CNT | 0x000001DA | 2 | R/W | Smart card clock control |
| CKG_CCDENC | 0x000001DC | 3 | R/W | Clock Control for the DENC Clock |
| CKG_CCST20 | 0x000001DD | 3 | R/W | Clock controller for ST20 clock |
| CKG_DIVST20 | 0x000001DE | 4 | R/W | Clock divider for ST20 clock |
| CKG_PREDIVPLL | 0x000001DF | 8 | R/W | Pre-divider ratio set-up of the PLL |
| CKG_FBKDIVPLL | 0x000001E0 | 8 | R/W | Feedback divider ratio set-up of the PLL |
| CKG_POSTDIVPLL | 0x000001E1 | 6 | R/W | Post divider ratio and lock detector |
| CKG_PLLSETUP | 0x000001E2 | 6 | R/W | PLL start-up control and charge pump control |
| CKG_IDDQPAD_C | 0x000001E3 | 4 | R/W | Iddq mode, I/O hsync, vsync |
| CKG_SFREQAUD_SDIV | 0x000001E4 | 3 | R/W | SDIV reg: AUD frequency synthesizer |
| CKG_SFREQAUD_PE0 | 0x000001E5 | 8 | R/W | PE reg (LSB): AUD frequency synthesizer |
| CKG_SFREQAUD_PE1 | 0x000001E6 | 8 | R/W | PE reg (MSB): AUD frequency synthesizer |
| CKG_SFREQAUD_MD | 0x000001E7 | 5 | R/W | MD reg: AUD frequency synthesizer |
| CKG_AUD_CNT | 0x000001E8 | 1 | R/W | Sets audio frequency control through ST20 or MMDSP |
| CKG_SFREQAUX_SDIV | 0x000001E9 | 3 | R/W | SDIV reg: AUX frequency synthesizer |
| CKG_SFREQAUX_PE0 | 0x000001EA | 8 | R/W | PE reg (LSB): AUX frequency synthesizer |
| CKG_SFREQAUX_PE1 | 0x000001EB | 8 | R/W | PE reg (MSB): AUX frequency synthesizer |
| CKG_SFREQAUX_MD | 0x000001EC | 5 | R/W | MD reg: AUX frequency synthesizer |
| CKG_AUX_CNT | 0x000001ED | 1 | R/W | Enable AUX frequency synthesizer |

**Table 44 Clock generator (CKG)**

## 7.3 Clock controller and clock divider registers

For the divider registers, only multiple of 2 are available:

- value 0 : Clock divided is PLL clock.
- value 1 : Clock divided is PLL clock/2.
- value 2 : Clock divided is PLL clock/4.
- value 3 : Clock divided is PLL clock/6 ...

Five clocks are derived form the PLL clock, each of them can be controlled and programmed separately.

### CKG_CCAUD Clock controller for audio clock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD0 | ENA | DV2 | BYP | Unused | | | | |

Address: ClockGenBaseAddress + 0xD0
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BYP | Bypass. The Clock output is bypassed and is connected to Pixclk.<br>0: Bypassed, 1: Not bypassed |
| DV2 | Divide - by -2. The output of the divider is divided by two.<br>0: Not divided by 2, 1: Divided by 2 |
| ENA | Enable. Controls whether or not the divider is enabled or disabled for low power.<br>0: Divider is disabled, 1: Divider is enabled |

### CKG_DIVAUD Clock divider for audio clock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD1 | NUM[3:0] | | | | Unused | | | |

Address: ClockGenBaseAddress + 0xD1
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| NUM[3:0] | Contains the value of the divider (Described in PLL Strategy and Clocks distribution) |

### CKG_CCAUXDENC Auxiliary clock controller

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD2 | ENA | DV2 | BYP | Unused | | | | |

Address: ClockGenBaseAddress + 0xD2

Type: Serial R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BYP | Bypass. The Clock output is bypassed and is connected to Pixclk<br>0: Bypassed, 1: Not bypassed |
| DV2 | 0: The output of the divider is not divided by 2, 1: The output of the divider is divided by 2 |
| ENA | 0: Low power divider is disabled, 1: Low power divider is enabled |

## CKG_DIVAUXDENC Denc Test clock divider

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD3 | NUM[3:0] | | | | Unused | | | |

Address: ClockGenBaseAddress + 0xD3
Type: Serial R/W
Reset value: 0

***Description***

This value in this register sets the fractional divider for the Denc Test clock.

## CKG_CCMCK Clock controller for MEMCLK

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD4 | ENA | DV2 | BYP | Unused | | | | |

Address: ClockGenBaseAddress + 0xD4
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BYP | Bypass. The Clock output is bypassed and is connected to Pixclk.<br>0: Bypassed, 1: Not bypassed |
| DV2 | The output of the divider is divided by two.<br>0: Not divided by 2, 1: Divided by 2 |
| ENA | Controls whether or not the divider is enabled or disabled for low power.<br>0: Divider is disabled, 1: Divider is enabled |

## CKG_DIVMCK Clock divider for MEMCLK

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD5 | NUM[3:0] | | | | Unused | | | |

Address: ClockGenBaseAddress + 0xD5
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| NUM[3:0] | Contains the value of the divider (Described in PLL Strategy and Clocks distribution) |

## CKG_CCDENC DENC clock controller

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xDC | ENA | DV2 | BYP | Unused | | | | |

Address: ClockGenBaseAddress + 0xDC
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BYP | Bypass. The Clock output is bypassed and is connected to Pixclk.<br>0: Bypassed, 1: Not bypassed |
| DV2 | Divide by 2. The clock input is divided by two.<br>0: Not divided by 2, 1: Divided by 2 |
| ENA | Enable. Controls whether or not the clock is enabled or disabled for low power.<br>0: Clk input is enabled, 1: Clk input (clk27MHz) is disabled |

## CKG_CCST20 Clock controller for ST20 clock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xDD | ENA | DV2 | BYP | Unused | | | | |

Address: ClockGenBaseAddress + 0xDD
Type: R/W
Reset value: 0xa0

***Description***

The ST20 Clock is enabled and divided (PLL clock divided by 4) when PLL is locked.

| Bitfield | Description |
|---|---|
| BYP | Bypass. The Clock output is bypassed and is connected to Pixclk.<br>0: Bypassed, 1: Not bypassed |
| DV2 | Divide - by -2. The output of the divider is divided by two.<br>0: Not divided by 2, 1: Divided by 2 |
| ENA | Enable. Controls whether or not the divider is enabled or disabled for low power.<br>0: Divider is disabled, 1: Divider is enabled |

## CKG_DIVST20 Clock divider for ST20 clock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xDE | NUM[3:0] | | | | Unused | | | |

Address: ClockGenBaseAddress + 0xDE
Type: R/W
Reset value: 0x20

***Description***

The ST20 Clock is enabled and divided (PLL clock divided by 4) when PLL is locked.

| Bitfield | Description |
|---|---|
| NUM[3:0] | Contains the value of the divider (Described in PLL Strategy and Clocks distribution). |

## CKG_LPC_DIV Low-power clock divider

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xCE | NUM[7:0] | | | | | | | |

Address: ClockGenBaseAddress + 0xCE
Type: Serial R/W
Reset value: 0x7F --> 212 KHz

***Description***

The 27MHz clock is divided by the value contained in this register. the minimum frequency for the Low Power Clock is 105 KHz.

## 7.4 PLL controller registers

Separate registers are used to program the clock frequency, and to control PLL setup.

### CKG_PREDIVPLL Pre-divider ratio set-up of the PLL

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xDF | M[7:0] | | | | | | | |

Address: ClockGenBaseAddress + 0xDF
Type: R/W
Reset value: 0x12

***Description***

| Bitfield | Description |
|---|---|
| M[7:0] | Pre-divider ratio setup of the PLL |

### CKG_FBKDIVPLL Feedback divider ratio set-up of the PLL

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE0 | N[7:0] | | | | | | | |

Address: ClockGenBaseAddress + 0xE0
Type: R/W
Reset value: 0xA2

***Description***

| Bitfield | Description |
|---|---|
| N[7:0] | Feedback divider ratio setup of the PLL |

### CKG_POSTDIVPLL Post divider ratio and lock detector

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE1 | P[2:0] | | | Setup[8:6] | | | Unused | |

Address: ClockGenBaseAddress + 0xE1
Type: R/W
Reset value: 0x30

***Description***

| Bitfield | Description |
|---|---|
| Setup[8:6] | Lock detector threshold > 4ns |
| P[2:0] | Post-divider ratio setup of the PLL |

### CKG_PLLSETUP PLL start-up control and charge pump control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE2 | Setup[5:4] | | Setup[3:0] | | | | Reserved | |

Address: ClockGenBaseAddress + 0xE2
Type: R/W
Reset value: 0x7C

***Description***

This register controls the Pre-divider, Post-divider, Feedback-divider and Setup of the PLL. There are four registers which are accessed at different address.

| Bitfield | Description |
|---|---|
| Setup[3:0] | Charge-pump control > 3.61µA |
| Setup[5:4] | Start-up control > 1.5v |

## 7.5 Configuration registers

### CKG_IDDQPAD_CNT Iddq mode, I/O hsync, vsync

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE3 | CF7 | CF6 | CF5 | CF4 | Reserved | | | |

Address: ClockGenBaseAddress + 0xE3
Type: Serial R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| CF4 | Controls the /output state of the Vsync/pwm2 pin.<br>0: Pwm2 is an output, 1: Vsync is an output |
| CF5 | Controls the output state of the Hsync/pwm0 pin.<br>0: Pwm0 pin is an output, 1: Hsync pin is an output |
| CF6 | Force test clock for MMDSP.<br>0: MMDSP clock, 1: Force test clock. |
| CF7 | IDDQ mode control<br>0: No IDDQ Mode, 1: IDDQ Mode |

## CKG_PLL_CNT Clock bypass PLL phase and PLL controller

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xCF | CF7 | CF6 | CF5 | CF[4:3] | | CF2 | Unused | |

Address: ClockGenBaseAddress + 0xCF
Type: R/W for CF[7:6], RO for CF[5:2]
Reset value: 0x38

***Description***

| Bitfield | Description |
|---|---|
| CF2 | State of sys_clk_off signal (From TPMAC)<br>0: All clocks are enabled<br>1: Clocks are turned off |
| CF[4:3] | State of what is required by the ST20 to control the PLL in stand-by mode (corresponds to lp_sys_pll_reg<1:0>)<br>00: Off<br>01: PLL reference on<br>10: PLL power on<br>11: PLL on |
| CF5 | State of Lock signal (PLL).<br>0: Lock is off, 1: Lock is on |
| CF6 | Disable the PLL<br>0: Enable - default value, 1: Disable. |
| CF7 | Control the Bypassclk Phase.<br>0: Bypassclk, 1: Bypassclk inverted |

## CKG_SMC_CNT Smart card clock control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xDA | SMC | ENA | Unused | | | | | |

Address: ClockGenBaseAddress + 0xDA
Type: Serial R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| ENA | Enable the Frequency Synthesizer for the Smart Card Clock<br>0: Frequency Synthesizer is on - default value<br>1: Frequency Synthesizer is off. |
| SMC | Control the source of the Smart Card Clock<br>0: PIO1_2 is used as external Source - default value<br>1: Clock from Frequency Synthesizer is used |

### CKG_AUX_CNT Enable AUX frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xED | ENA | Reserved | | | | | | |

Address: ClockGenBaseAddress + 0xED
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| ENA | 0: Frequency Synthesizer on,1: Frequency Synthesizer off |

### CKG_AUD_CNT Sets audio frequency control through ST20 or MMDSP

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE8 | SRG | Reserved | | | | | | |

Address: ClockGenBaseAddress + 0xE8
Type: Serial R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| SRG | Controls the value of frequency register<br>0: MMDSP programs the Frequency Synthesizer - default value<br>1: ST20 via Clock Generator controls the Frequency Synthesizer. |

## 7.6 SMC clock frequency synthesizer

### CKG_SFREQSMC_SDIV SDIV reg: SMC frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD6 | SDIV[2:0] | | | Reserved | | | | |

Address: ClockGenBaseAddress + 0xD6
Type: R/W
Reset value: 0

***Description***

This register controls the SDIV values, combined with PE and MD registers.

### CKG_SFREQSMC_PE PE reg: SMC frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD8 | PE[15:8] | | | | | | | |
| 0xD7 | PE[7:0] | | | | | | | |

Address: ClockGenBaseAddress + 0xD7 (PE[7:0]), 0xD8 (PE[15:8])
Type: R/W
Reset value: 0

***Description***

This register controls the PE values, combined with SDIV and MD registers.

### CKG_SFREQSMC_MD MD reg: SMC frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD9 | MD[4:0] | | | | | Unused | | |

Address: ClockGenBaseAddress + 0xD9
Type: R/W
Reset value: 0

***Description***

This register controls the MD values, combined with SDIV and PE registers.

## 7.7 Auxiliary clock frequency synthesizer

For programming, this clock is the same as the PCM clock. The Frequency synthesizer uses 24 bits (1x16 + 1x5 + 1x3) to program the frequency. 3 x 8 bits registers are inserted.

### CKG_SFREQAUX_SDIV SDIV reg: AUX frequency synthesizer

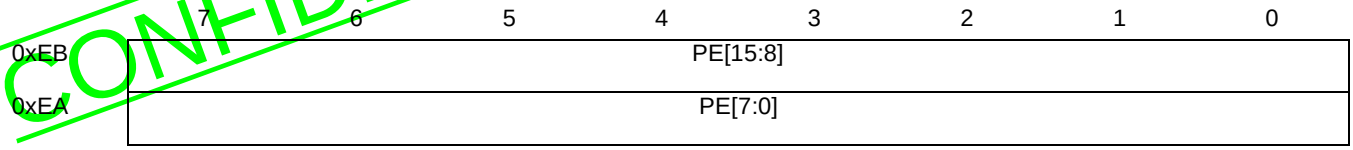
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE9 | SDIV[2:0] | | | Unused | | | | |

Address: ClockGenBaseAddress + 0xE9
Type: R/W
Reset value: 0

***Description***

This register controls the SDIV values, combined with PE and MD registers.

### CKG_SFREQAUX_PE PE reg : AUX frequency synthesizer

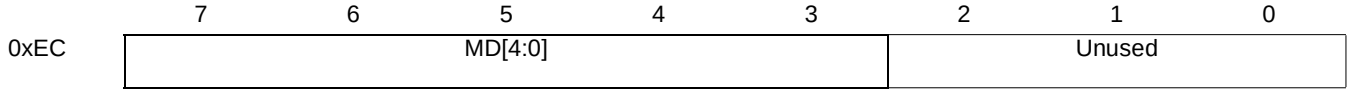



Address: ClockGenBaseAddress + 0xEA (PE[7:0]), 0xEB (PE[15:8])
Type: R/W
Reset value: 0

***Description***

This register controls the PE values, combined with SDIV and MD registers.

### CKG_SFREQAUX_MD MD reg: AUX frequency synthesizer




Address: ClockGenBaseAddress + 0xEC
Type: R/W
Reset value: 0

***Description***

This register controls the MD values, combined with SDIV and PE registers.

## 7.8 Audio clock frequency synthesizer

The frequency synthesizer uses the CKG_SFREQAUD_SDIV, CKG_SFREQAUD_PE and CKG_SFREQAUD_MD registers to program the frequency, and the 1-bit Selreg register to validate it. Table 45 lists the register settings versus audio frequency values.

| Audio frequency | SDIV (hex) | MD (hex) | PE (hex) |
|---|---|---|---|
| 384 x 32KHz | 4 | 11 | 3600 |
| 384 x 44.1KHz | 3 | 19 | 3EB2 |
| 384 x 48KHz | 3 | 17 | 4800 |
| 256 x 32KHz | 4 | 1A | 5100 |
| 256 x 44.1KHz | 4 | 13 | 6F05 |
| 256 x 48KHz = 384 x 32KHz | 4 | 11 | 3600 |
| 256 x 96KHz | 3 | 11 | 3600 |
| 384 x 96KHz | 2 | 17 | 4800 |
| 256 x 192KHz | 2 | 11 | 3600 |
| 192 x 192KHz = 384 x 96KHz | 2 | 17 | 4800 |

**Table 45 Audio frequency vaues**

## CKG_SFREQAUD_SDIV SDIV reg: AUD frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE4 | SDIV[2:0] | | | | Reserved | | | |

Address: ClockGenBaseAddress + 0xE4
Type: Serial R/W
Reset value: 0

***Description***

This register controls the SDIV values, combined with PE and MD registers.

## CKG_SFREQAUD_PE PE reg : AUD frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE5 | PE[7:0] | | | | | | | |
| 0xE6 | PE[15:8] | | | | | | | |

Address: ClockGenBaseAddress + 0xE5 (CKG_SFREQAUD_PE0), 0xE6 (CKG_SFREQAUD_PE1)
Type: Serial R/W
Reset value: 0x0 1st Register - 0x0 2nd Register

***Description***

This register controls PE values, combined with SDIV and MD registers.

## CKG_SFREQAUD_MD MD reg: AUD frequency synthesizer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE7 | MD[4:0] | | | | | Reserved | | |

Address: ClockGenBaseAddress + 0xE7
Type: Serial R/W
Reset value: 0

***Description***

This register controls the MD values, combined with SDIV and PE registers.

# 8 Digital encoder (DEN)

## 8.1 Introduction

Where teletext is not implemented, registers related to teletext should not be used and register bit DEN_CFG6.TTX_EN must be left at the reset value of zero.

## 8.2 Register descriptions

### DEN_CFG0 Configuration0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0 | std1 | std0 | sync2 | sync1 | sync0 | polh | polv | freerun |

Address: DencBaseAddress + 0x0
Type: Serial R/W
Reset value: 10010010

***Description***

| Bitfield | Description |
|---|---|
| std[1:0] | Standard<br>0 0: PAL BDGHI<br>0 1: PAL N (see bit set-up)<br>1 0: NTSC M [1] [2]<br>1 1: PAL M |
| sync[2:1:0] | Configuration<br>0 0 0: ODDEV-only based SLAVE mode (frame locked)<br>0 0 1: F based SLAVE mode (frame locked)<br>0 1 0: ODDEV + HSYNC based SLAVE mode (line locked)<br>0 1 1: F+ H based SLAVE mode (line locked)<br>1 0 0: VSYNC-only based SLAVE mode (frame locked) [3]<br>1 0 1: VSYNC + HSYNC based SLAVE mode (line locked)<br>1 1 0: MASTER mode<br>1 1 1: AUTOTEST mode (color bar pattern) |
| polh | Synchro: active edge of HSYNC selection (when input) or polarity of HSYNC (when output)<br>0: HSYNC is a negative pulse (128 Tckref wide) or falling edge is active<br>1: HSYNC is a positive pulse (128 Tckref wide) or rising edge is active |
| polv | Synchro: active edge of ODDEV/VSYNC selection (when input)<br>0: Falling edge of ODDEV flags start of field1 (odd field) or VSYNC is active low<br>1: Rising edge of ODDEV flags start of field1 (odd field) or VSYNC is active high |
| freerun | Bit “freerun” is taken into account in ODDEV-only, VSYNC-only or F-based slave modes. It is irrelevant to other synchronization modes.<br>Synchro: active edge of ODDEV/VSYNC selection (when input)<br>0: Disabled<br>1: Enabled |

1. The standard on hardware reset is NTSC; any standard modification automatically selects the right parameters for correct subcarrier generation.
2. If bit ‘secam’ from register DEN_CFG7 is set, ‘std1’ and ‘std0’ bits are not taken into account.
3. In VSYNC-only based slave mode (sync[2:0]=”100”), HSYNC is nevertheless needed as an input.

## DEN_CFG1 Configuration1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x01 | blkli | flt1 | flt0 | syncok | coki | setup | cc2 | cc1 |

Address: DencBaseAddress + 0x01
Type: Serial R/W
Reset value: 01000100

### *Description*

| Bitfield | Description |
|---|---|
| cc[2:1] | 00:[2] No closed caption/extended data encoding<br>01: Closed caption/extended data encoding enabled in field 1 (odd)<br>10: Closed caption/extended data encoding enabled in field 2 (even)<br>11: Closed caption/extended data encoding enabled in both fields |
| setup | Pedestal enable<br>0:[2] Blanking level and black level are identical on all lines (ex: Arg. PAL-N, Japan NTSC-M, PAL-BDGHI)<br>1: Black level is 7.5 IRE above blanking level on all lines outside VBI (ex: Paraguayan and Uruguayan PAL-N). In all cases, gain factor is adjusted to obtain the required chrominance levels. |
| coki | Color killer<br>0:[2] Color ON<br>1: Color suppressed on CVBS output signal (CVBS=YS) but color still present on C output<br>For color suppression on chroma DAC 'C', see register DEN_CFG5.bkdac2. |
| syncok | Sync signal availability (analog & digital) for input synchronization loss with free-run inactive freerun=0<br>0:[2] No synchro output signals<br>1: Output synchros available on YS, CVBS and, when applicable, HSYNC (if output port), ODDEV (if output port),: i.e same behavior as free-run except that video outputs are blanked in the active portion of the line |
| flt[1:0] | U/V Chroma filter bandwidth selection. Typical applications for 3dB bandwidth are given for each flt bit configuration.<br>00: $f_{-3dB}$=1.1MHz low def NTSC filter<br>01: $f_{-3dB}$=1.3MHz low def PAL filter<br>10:[2] $f_{-3dB}$=1.6MHz high def. NTSC filter (ATSC compliant) & PAL M/N (ITU-R 624.4 compliant)<br>11: $f_{-3dB}$=1.9MHz high def. PAL filter: Rec 624 - 4 for PAL BDG/I compliant |
| blkli [1] | Vertical Blanking Interval selection for active video lines area.<br>0:[2] ('partial blanking') Only the following lines inside the vertical interval are blanked:<br>NTSC-M: lines [1..9], [263(half)..272] (525-SMPTE), PAL-M: lines [523..6], [260(half)..269] (525-CCIR)<br>other PAL: lines [623(half)..5], [311..318] (625-CCIR)<br>This mode preserves embedded VBI data within the incoming YCrCb, e.g. Teletext (lines [7..22] and [320..335]), Wide Screen signalling (full line 23), Video Programing Service (line16), etc.).<br>1: ('full blanking') All lines inside VBI are blanked<br>NTSC-M: lines [1..19], [263(half)..282] (525-SMPTE), PAL-M: lines [523..16], [260(half)..279] (525-CCIR)<br>other PAL: lines [623(half)..22], [311..335] (625-CCIR) |

1. **blkli** must be set to '0' when closed captions are to be encoded on the following lines:
   in 525/60 system: before line 20(SMPTE) or before line 283(SMPTE)
   in 625/50 system: before line 23(CCIR) or before line 336(CCIR)
2. DEFAULT mode when denc_nrst pin is active (LOW level)

## DEN_CFG2 Configuration2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x02 | nintrl | enrst | bursten | set4:4:4 | selrst | rstosc_buf | valrst1 | valrst0 |

Address: *DencBaseAddress* + 0x02
Type: Serial R/W
Reset value: 00100000

***Description***

| Bitfield | Description |
|---|---|
| valrst[0:1] [1] | 00:[2] Automatic reset of the oscillator every line<br>01: Automatic reset of the oscillator every 2nd field<br>10: Automatic reset of the oscillator every 4th field<br>11: Automatic reset of the oscillator every 8th field |
| rstosc_buf [3] | Software phase reset of DDFS (Direct Digital Frequency Synthesizer) buffer<br>0:[2] No reset<br>1: When a 0-to-1 transition occurs, either the hard-wired default phase value, or the value loaded in the DEN_PDFS1/2 register (according to bit 'selrst'), is put to phase buffer. This value is loaded into accumulator (phase of sub-carrier) when bits **ph_rst_osc** from DEN_CFG8 are programmed, or when the standard changes or softreset occurs. |
| selrst | Selects reset values for Direct Digital Frequency Synthesizer<br>0:[2] Hardware reset values for subcarrier oscillator phase (see DEN_PDFS1/2 register descriptions for values)<br>1: Loaded reset values selected (see contents of the DEN_PDFS1/2 registers) |
| set4:4:4 | Selects the 4:2:2 or 4:4:4 video input from the mixing unit<br>0:[2] YCrCb (4:2:2) input :<br>The presence of OSD on the DENC output can be selected or deselected by OSD block header field S.<br>1: YCrCb (4:4:4) input : In this mode OSD is always present on the DENC output. |
| bursten | Chrominance burst control<br>0: Burst is turned off on CVBS, chrominance output is not affected<br>1:[2] Burst is enabled |
| enrst | Cyclic update of DDFS phase<br>0:[2] No cyclic subcarrier phase reset<br>1: Cyclic subcarrier phase reset depending of valrst1 and valrst0 (see below) |
| nintrl [4] | Non-interlaced mode select<br>0:[2] Interlaced mode (625/50 or 525/60 system)<br>1: Non-interlaced mode(2x312/50 or 2x262/60 system) |

1. **valrst[1:0]** is taken into account only if bit 'enrst' is set. Resetting the oscillator means here forcing the value of the phase accumulator to its nominal value to avoid accumulating errors due to the finite number of bits used internally. The value to which the accumulator is reset is either the hard-wired default phase value or the value loaded in the DEN_PDFS1/2 registers (according to bit 'selrst'), to which a $0^0$, $90^0$, $180^0$, or $270^0$ correction is applied according to the field and line on which the reset is performed.Note: If SECAM is performed the oscillator is reset every line.
2. Default mode when **denc_nrst** pin is active (LOW level)
3. **rstosc_buf** is automatically set back to '0' after the buffer is loaded
4. **nintrl** update is internally taken into account at the beginning of the next frame. In SECAM mode, only the interlaced mode is available.

## DEN_CFG3 Configuration3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x03 | entrap | trap_4.43 | encgms | ck_in_phase | del2 | del1 | del0 | enwss |

Address: DencBaseAddress + 0x03
Type: Serial R/W
Reset value: 00000000

***Description***

| Bitfield | Description |
|---|---|
| enwss | WSS encoding enable<br>0:[4] Disabled<br>1: Enabled |
| del[2:0] | Delay on luma path with reference to chroma path on 4:2:2 inputs<br>0 1 0: + 2 pixel delay on luma<br>0 0 1: + 1 pixel delay on luma<br>0 0 0: [4] + 0 pixel delay on luma<br>1 1 1: - 1 pixel delay on luma<br>1 1 0: - 2 pixel delay on luma<br>Other configurations: + 0 pixel delay on luma |
| ck_in_phase [1] | Choice of active edge of **denc_ref_ck** (master clock) that samples incoming YCrCb data.<br>This bit can be used to analyze the cause of application synchronization problems.<br>0:[4] **denc_ref_ck** falling edge<br>1: **denc_ref_ck** rising edge |
| encgms [2] | CGMS encoding enable<br>0:[4] Disabled<br>1: Enabled |
| trap_4.43 | Enable trap filter: Trap_4.43 is taken into account only if bit **entrap** is set<br>0:[4] Select the trap filter centered around 3.58 MHz<br>1: Select the trap filter centered around 4.43 MHz |
| entrap [3] | Enable trap filter<br>0:[4] Trap filter disabled<br>1: Trap filter enabled |

1. This bit is typically used when synchronization problems appear in application.
2. When encgms is set to 1, Closed-Captions/Extended Data Services should not be programmed on lines 20 and 283 (525/60, SMPTE line number convention).
3. When SECAM is performed trap filter is always enabled (entrap =’1’).
4. Default mode when **denc_nrst** pin is active (LOW level)

## DEN_CFG4 Configuration4

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x04 | syncin_ad1 | syncin_ad0 | syncout_ad1 | syncout_ad0 | aline | del4_[2:0] | | |

Address: *DencBaseAddress* + 0x04
Type: Serial R/W
Reset value: 00000000

***Description***

| Bitfield | Description |
|---|---|
| del4[2:0] | Delay on luma path w.r.t chroma path on YUV and RGB outputs when 4:4:4 input is used<br>0 1 0: + 2 pixel delay on luma<br>0 0 1: + 1 pixel delay on luma<br>0 0 0: [4] + 0 pixel delay on luma<br>1 1 1: - 1 pixel delay on luma<br>1 1 0: - 2 pixel delay on luma<br>Other configurations: + 0 pixel delay on luma |
| aline | Video active line duration control<br>0:[1] Full digital video line encoding (720 pixels - 1440 clock cycles)<br>1: Active line duration follows ITU-R/SMPTE ‘analog’ standard requirements |
| syncout_ad[1:0] | Adjustment of outgoing sync signals. Used to ensure correct interpretation of incoming video samples as Y, Cr or Cb when the encoder is master and supplies sync signals<br>00:[1] Nominal<br>01: +1 ckref<br>10: +2 ckref<br>11: +3 ckref |
| syncin_ad[1:0] | Adjustment of incoming sync signals. Used to ensure correct interpretation of incoming video samples as Y, Cr or Cb when the encoder is slaved to incoming sync signals (incl. ‘F/H’ flags stripped off ITU-R656/D1 data)<br>00:[1] Nominal<br>01: +1 ckref<br>10: +2 ckref<br>11: +3 ckref |

1. Default mode when **denc_nrst** pin is active (LOW level)

## DEN_CFG5 Configuration5

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x05 | selrst_inc | bkdac1 | bkdac2 | bkdac3 | bkdac4 | bkdac5 | bkdac6 | dacinv |

Address: DencBaseAddress + 0x05
Type: Serial R/W
Reset value: 00000000

***Description***

| Bitfield | Description |
|---|---|
| dacinv | 'Inverts' DAC codes to compensate for an inverting output stage in the application<br>0:[1] Non inverted DAC inputs (outputs)<br>1: Inverted DAC inputs (outputs) |
| bkdacN | Blanking of DACs (N = 1,2,3,4,5 or 6)<br>0:[1] DAC N in normal operation<br>1: DAC N input code forced to black level (if RGB, UV or C) or blanking level (if Y or CVBS) depending on conf_out bits of the DEN_CFG8 register. |
| selrst_inc | Choice of Digital Frequency Synthesizer increment after soft-reset or when ph_rst_mode = '01' (see the DEN_CFG8 register)<br>0:[1] Hard wired value (depending on TV standard)<br>1: Soft (value from DEN_IDFS1/2/3 registers) |

1. Default mode when **denc_nrst** pin is active (LOW level)

## DEN_CFG6 Configuration6

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x06 | softreset | jump | dec_ninc | free_jump | cfc1 | cfc2 | ttx_en | maxdyn |

Address: DencBaseAddress + 0x06
Type: Serial R/W
Reset value: 00010000

***Description***

| Bitfield | Description |
|---|---|
| max_dyn[1] | Max dynamic magnitude allowed on YCrCb inputs for encoding<br>0:[5] 10hex to EBhex for Y, 10hex to E0hex for chrominance (Cr,Cb)<br>1: 01hex to FEhex for Y, Cr and Cb |
| ttx_en | Teletext enable bit<br>0:[5] Disabled<br>1: Enabled |
| cfc[1:0] | Color frequency control via CFC line<br>0 0:[5] Update mode disabled (update is carried out by loading DEN_IDFS1/2/3 registers)<br>0 1: Update of increment for DDFS just after serial loading via CFC<br>1 0: Update of increment for DDFS on next active edge of HSYNC<br>1 1: Update of increment for DDFS just before next color burst |
| jump: [2]<br>dec_ninc:<br>free_jump<br><br>(bits[6:5:4]) | Update mode<br>0 0 0: Normal mode (no line skip/insert capability) ITU-R (CCIR): 313/312 or 263/262 non-interlaced: 312/312 or 262/262<br>0 x 1: [5] Manual mode for line insert (“dec_ninc”=0) or skip (“dec_ninc”=1) capability. Both fields of all the frames following the writing of this value are modified according to “Iref” and “Itar” bits of DEN_LJPM1/2/3 registers (by default, “Iref”=0 and “Itar”=1 giving the normal mode above).<br>1 0 0: Automatic line insert mode. The 2nd field of the frame following the writing of this value is increased. Line insertion is done after line 245 in 525/60 and after line 290 in 625/50. “Iref” and “Itar” are ignored.[3]<br>1 1 0 Automatic line skip mode. The 2nd field of the frame following the writing of this value is decreased. Line suppression is done after line 245 in 525/60 and after line 290 in 625/50. “Iref” and “Itar” are ignored.[3]<br>1 x 0: DO NOT USE! |
| softreset [4] | Software reset<br>0:[5] No reset<br>1: Software reset |

1. EAV and SAV words are replaced by blanking values before being fed to the luminance and chrominance processing.
2. Bit jump is automatically reset after use.
3. Two lines are skipped (inserted) in 525/60 and four lines in 625/50 standards.
4. Bit **softreset** is automatically reset after internal reset generation. Software reset is active for 4 CKREF periods. When softreset is activated, all the device is reset as with hardware reset except for the first nine user registers (DEN_CFG0 to DEN_CFG8).
5. Default mode when denc_nrst pin is active (LOW level)

## DEN_CFG7 Configuration7

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x07 | secam | gen_secam | inv_phi_secam | not used | setupYUV | uv_lev | envps | sqpix |

Address: DencBaseAddress + 0x07
Type: Serial R/W
Reset value: 00000000

***Description***

| Bitfield | Description |
|---|---|
| sqpix | Square pixel mode enable: This mode is not available in secam standard.<br>0:[1] Disable<br>1: Enable |
| envps | VPS encoding enable<br>0:[1] Disable<br>1: Enable |
| uv_lev | UV output level<br>0:[1] Same peak to peak amplitude for both U and V (default value - 0.7 Vpp for 100/0/100/0 color bar pattern)<br>1: U and V outputs as defined by ITU-R624-4 |
| setupYUV | Pedestal enable on YUV outputs:<br>This bit is significant only if register DEN_CFG8 bit conf_out(1:0) = '01' (YUV outputs). If this is the case, the Y output on dac5_g_y has a 7.5 IRE pedestal. Furthermore, if uv_lev = '1' then the U and V levels also depend on the value of setupYUV.<br>0:[1] Disable<br>1: Enable |
| inv_phi_secam | Inversion of sub-carrier phase<br>0:[1] 0, 0, π,... in odd fields and π, π, 0 in even fields<br>1: π, π, 0 in odd fields and 0, 0, π,... in even fields |
| gen_secam /secam | Where the bit order is secam - gen_secam:<br>00[1]: Standard selected by std1 and std0 bits of DEN_CFG0 (PAL or NTSC)<br>01: Genlock slave mode using external sync extraction<br>10: Secam standard, the sub-carrier phase sequence start-point is line1<br>11: Secam standard, the sub-carrier phase sequence start-point is line23<br>If master mode is performed in the secam standard, bit "secam" must be set before sync2, sync1 and sync0 bits of DEN_CFG0. If DEN_CFG0 is not programmed, then a soft reset must be performed after programming secam. In secam, the trap filter should always be enabled (see DEN_CFG3 register). |

1. Default value

## DEN_CFG8 Configuration8

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x08 | ph_rst_mode1 | ph_rst_mode0 | conf_out1 | conf_out0 | blk_all | ttx_notmv | xxx | xxx |

Address: DencBaseAddress + 0x08
Type: Serial R/W
Reset value: 00100000

***Description***

| Bitfield | Description |
|---|---|
| ttx_notmv | ***WARNING!*** *After reset, the default value of this bit is zero, however, for devices using teletext, this bit* ***must*** *then be reprogrammed to 1. This is to avoid the occurrence of teletext glitches in Secam mode and loss of teletext data in all modes.*<br>Priority of ancillary data on a VBI line. Note, higher priority data overwrites lower priority data.<br>0[1] Priority is: CGMS > Closed caption > Macrovision[2] > WSS > VPS > Teletext<br>1 Priority is: Teletext > CGMS > Closed caption > WSS > VPS > Macrovision |
| blk_all | Blanking of all video lines<br>0[1] Disabled<br>1 Enabled (all inputs ignored - 80hex instead of Cr and Cb and 10hex instead of Y and Y4) |
| conf_out[1:0] | DENC output configuration |
| ph_rst_mode[1:0] | Sub-carrier phase reset mode.<br>In modes “01” and “10”, this bit is automatically reset to “00” after oscillator reset.<br>00 [1] Disabled<br>01 Enabled - phase is updated with value from phase buffer register (see DEN_CFG2 bit rstosc_buf) on the beginning of the next video line. Increment is updated with hard or soft value depending on selreg_inc value (see DEN_CFG5)<br>10 Enabled - phase is updated with value from the DEN_IDFS1/2/3 registers on the next increment updating by cfc (depending on cfc loading moment and DEN_CFG6 cfc(1:0) bits.<br>11 Enabled - phase is reset after detecting of rst bit on cfc line, up to 9 denc_ref_ck after loading of cfc’s LSB. |

| | | dac1 | dac2 | dac3 | dac4 | dac5 | dac6 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | Y | C | CVBS | C | Y | CVBS |
| 0 | 1 | Y | C | CVBS | V | Y | U |
| 1 | x[1] | Y | C | CVBS | R | G | B |

If conf_out(1:0) = ‘01’ and register DEN_CFG2 bit set_4:4:4 = ‘1’, then the dac5 output Y comes from the Y4 input.

1. Default value
2. If there is no Macrovision programmed on the line, then VBI line blanking (when register DEN_CFG1 bit blki=1) will over-write VPS or teletext, and no VPS or teletext data will be encoded.

## DEN_STA Status (read only)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x09 | hok | atfr | buf2_free | buf1_free | fieldct2 | fieldct1 | fieldct0 | jump |

Address: DencBaseAddress + 0x09
Type: Serial R0
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| jump | Indicates whether a frame length modification has been programmed at '1' from programming of bit 'jump' to end of frame(s) concerned<br>default = 0 Refer to DEN_CFG6 and DEN_LJPM1/2/3 registers |
| fieldct[2:0] | Digital field identification number<br>000 Indicates field 1<br>...<br>111 Indicates field 8<br>NOTE: fieldct[0] also represents the odd/even information (odd='0', even='1') |
| buf1_free | Closed caption registers access condition for field 1<br>Same as buf2_free but concerns field 1.<br>Reset value:= 1 (access authorized)[2] |
| buf2_free | Closed caption registers access condition for field 2<br>Closed caption data for field 2 is buffered before being output on the relevant TV line; buf2_free is reset if the buffer is temporarily unavailable. If the microcontroller can guarantee that the DEN_CCF2 registers are never written more than once between two frame reference signals, then bit 'buf2_free' will always be true (set). Otherwise, closed caption field2 registers (DEN_CCF2) access might be temporarily forbidden by resetting bit 'buf2_free' until the next field2 closed caption line occurs.<br>Note that this bit is false (reset) when 2 pairs of data bytes are awaiting to be encoded, and is set back immediately after one of these pairs has been encoded (so at that time, encoding of the last pair of bytes is still pending)<br>Reset value:= 1 (access authorized)[2] |
| atfr | Frame synchronization flag<br>0[2] Encoder not synchronized<br>1 In slave mode: encoder synchronized |
| hok[1] | Hamming decoding of frame sync flag embedded within ITU-R656 / D1 compliant YCrCb streams<br>0 Consecutive errors<br>1[2] A single or no error |

1. Signal quality detector is issued from Hamming decoding of EAV,SAV from YCrCb
2. DEFAULT mode when denc_nrst pin is active (LOW level)

## DEN_IDFS1/2/3 Increment for digital frequency synthesizer

These three registers contain the 24-bit increment used by the DDFS, **ONLY IF** bit 'selrst_inc' (register DEN_CFG5) equals '1'. They generate the phase of the subcarrier, i.e. the address that is supplied to the sine ROM. It, therefore, customizes the synthesized subcarrier frequency: 1 LSB ~ 1.6 Hz

To validate use of these registers instead of the hard-wired values:

- Load the registers with the required value
- Set bit 'selrst_inc' to 1 (register DEN_CFG5)
- Perform a software reset (register DEN_CFG6)

*Note The values loaded in DEN_IDFS1/2/3 are taken into account after a software reset, and* **ONLY IF** *bit 'selrst_inc'='1' (register DEN_CFG5)*
*These registers are never reset and must be explicitly written, to ensure that they contain sensible information.*

On hardware or software reset the DDFS is initialized with a hardwired increment, independent of DEN_IDFS1/2/3. These hardwired values cannot be read out of the DENC.

| | d23 | d22 | d21 | d20 | d19 | d18 | d17 | d16 |
|---|---|---|---|---|---|---|---|---|
| 0x0A | | | | | | | | |
| | d15 | d14 | d13 | d12 | d11 | d10 | d9 | d8 |
| 0x0B | | | | | | | | |
| | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 |
| 0x0C | | | | | | | | |

Address: DencBaseAddress+ 0x0A(DEN_IDFS1), 0x0B (DEN_IDFS2), 0x0C(DEN_IDFS3)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description | | |
|---|---|---|---|
| DEN_IDFS1/2/3 | Value | Frequency synthesized | Ref.Clock |
| | d(23:0): 21F07C hexa for NTSC M | f=3.5795452 MHz | 27 MHz |
| | d(23:0): 2A098B hexa for PAL BGHIN | f=4.43361875MHz | 27 MHz |
| | d(23:0): 21F694 hexa for PAL N | f=3.5820558 MHz | 27 MHz |
| | d(23:0): 21E6F0 hexa for PAL M | f=3.57561149 MHz | 27 MHz |
| | d(23:0): 255554 hexa for NTSC M square pixel | f=3.5795434 MHz | 24.545454 MHz |
| | d(23:0): 26798C hexa for PAL BGHIN square pixel | f=4.43361867 MHz | 29.5 MHz |
| | d(23:0): 1F15C0 hexa for PAL N square pixel | f=3.58205605 MHz | 29.5 MHz |
| | d(23:0): 254AD4 hexa for PAL M square pixel | f=3.57561082 MHz | 24.545454 MHz |

## DEN_PDFS1/2 Phase_dfs

These registers are never reset and must be explicitly written to.

If bit 'selrst'=0 (e.g. after a hardware reset) the phase offset used every time the DDFS is reinitialized is a hard-wired value. The hard-wired values *cannot be read out of the DENC*. These are:

D9C000hex for PAL BDGHI, N, M,
1FC000hex for NTSC-M,
000000hex (blue lines)
43C000hex (red lines) for SECAM

**When secam = '0' (DEN_CFG8)**

*Static phase offset for digital frequency synthesizer (10 bits only)*

| | - | - | - | - | - | - | o23 | o22 |
|---|---|---|---|---|---|---|---|---|
| 0x0D | | | | | | | | |
| | o21 | o20 | o19 | o18 | o17 | o16 | o15 | o14 |
| 0x0E | | | | | | | | |

Address: DencBaseAddress + 0x0D (DEN_PDFS1), 0x0E (DEN_PDFS2)
Type: Serial R/W
Reset value: Undefined

*Under certain circumstances* (detailed below), these registers contain the 10 MSBs of the value with which the phase accumulator of the DDFS is initialized after a 0-to-1 transition of bit 'rstosc' (DEN_CFG2), or after a standard change, or when cyclic phase readjustment has been programmed (see bits valrst[1:0] of DEN_CFG2). The 14 remaining LSBs loaded into the accumulator in these cases are all '0's (defining the phase reset value with a $0.35^{o}$ accuracy).

To validate use of these registers instead of the hard-wired values:

- Load the registers with the required value
- Set bit 'selrst' to 1 (DEN_CFG2)
- Perform a software reset (DEN_CFG6), or set DEN_CFG6 bit rstosc_buf to "1". This puts the soft-phase value into a tampon register and sets register DEN_CFG8 bit ph_rst_mode[1:0] to put this value into an accumulator at the beginning of the next line.

**When secam = '1' (DEN_CFG8)**

*Static phase offset for digital frequency synthesizer for two SECAM sub-carriers (8 MSB only) (blue lines* - DEN_PDFS1 *and red lines* - DEN_PDFS2*)*

| | b21 | b20 | b19 | b18 | b17 | b16 | b15 | b14 |
|---|---|---|---|---|---|---|---|---|
| 0x0D | | | | | | | | |
| | r21 | r20 | r19 | r18 | r17 | r16 | r15 | r14 |
| 0x0E | | | | | | | | |

Address: DencBaseAddress + 0x0D (DEN_PDFS1), 0x0E (DEN_PDFS2),
Type: Serial R/W
Reset value: Undefined

These registers contain the 8 bits (21 to14) of the value with which the phase accumulator of the DDFS is initialized on **every line** in SECAM mode. The phase is calculated with 1.4° accuracy as:

(Blue lines) DEN_PDFS1 * 16384 (dec), or DEN_PDFS1 * 4000 (hex)

(Red lines) (256 + DEN_PDFS1 + DEN_PDFS2) * 16384 (dec), or (100 + DEN_PDFS1 + DEN_PDFS2) * 4000 (hex)

To validate use of these registers instead of the hard-wired values, follow the PAL and NTSC mode.

## DEN_WSS1/2 WSS_BIT[15:0]: WSS data registers

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0x0F | wss15 | wss14 | wss13 | wss12 | wss11 | wss10 | wss9 | wss8 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x10 | wss7 | wss6 | wss5 | wss4 | wss3 | wss2 | wss1 | wss0 |

Address: DencBaseAddress + 0x0F(DEN_WSS1), 0x10 (DEN_WSS2)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description | |
|---|---|---|
| wss[0] | 0: | No subtitles within teletext |
| | 1: | Subtitles within teletext |
| wss[2:1] | 0 0: | No open subtitles |
| | 0 1: | Subtitles in active image area |
| | 1 0: | Subtitles out of image area |
| | 1 1: | reserved |
| wss[7:3] | reserved = 0 | |
| wss[11:8] [1] | 1 0 0 0: | full format 4:3 |
| | 0 0 0 1: | box 14:9 center |
| | 0 0 1 0: | box 14:9 top |
| | 1 0 1 1: | box 16:9 center |
| | 0 1 0 0: | box 16:9 top |
| | 1 1 0 1: | > box 16:9 center |
| | 1 1 1 0: | full format 4:3 (shoot and protect 14:9 center) |
| | 0 1 1 1: | full format 16:9 (anamorphic) |
| wss[12] | 0: | Camera mode |
| | 1: | Film mode |
| wss[15:13] | reserved = 0 | |

1. wss11 is an odd parity bit

## DEN_DAC13 Dac1 and dac3 multiplying factors

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x11 | dac1_mult[3:0] | | | | dac3_mult[3:0] | | | |

Address: DencBaseAddress + 0x11
Type: Serial R/W
Reset value: 10001000

***Description***

| Bitfield | Description |
|---|---|
| dac3_mult[3:0] | multiplying factor on dac3_cvbs digital signal before the D/A converters with 3.125% step<br>0 0 0 0: 75.00% dac3_cvbs value compared to default<br>0 0 0 1: 78.125% dac3_cvbs value compared to default<br>0 0 1 0: 81.25% dac3_cvbs value compared to default<br>0 0 1 1: 84.375% dac3_cvbs value compared to default<br>.. .. .. .. ...<br>1 0 0 0: [1] 100% dac3_cvbs value compared to default<br>.. .. .. .. <br>1 1 1 1: 121.875% dac3_cvbs value compared to default |
| dac1_mult[3:0] | multiplying factor on dac1_y digital signal before the D/A converters with 3.125% step<br>0 0 0 0: 75.000% dac1_y value compared to default<br>0 0 0 1: 78.125% dac1_y value compared to default<br>0 0 1 0: 81.250% dac1_y value compared to default<br>0 0 1 1: 84.375% dac1_y value compared to default<br>.. .. .. .. ...<br>1 0 0 0: [1] 100% dac1_y value compared to default<br>.. .. .. .. ...<br>1 1 1 1: 121.875% dac1_y value compared to default |

1. Default value

## DEN_DAC45 DAC4 and DAC5 multiplying factors

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x12 | dac4_mult[3:0] | | | | dac5_mult[3:0] | | | |

Address: DencBaseAddress + 0x12
Type: Serial R/W
Reset value: 10001000

***Description***

| Bitfield | Description | | |
|---|---|---|---|
| dac5_mult[3:0] | Multiplying factor on dac5_g_y digital signal before the D/A converters:<br>dac5_g_y value (in% of default value) | | |
| | Value | if G(conf_out=1x) | if Y(conf_out=0x) |
| | 0 0 0 0: | 80.49% | 75.000% |
| | 0 0 0 1: | 82.93% | 78.125% |
| | 0 0 1 0: | 85.37% | 81.250% |
| | 0 0 1 1: | 87.81% | 84.375% |
| | .. .. .. .. | ... | ... |
| | 1 0 0 0: [1] | 100% | 100% |
| | .. .. .. .. | | |
| | 1 1 1 1: | 117.07% | 121.875% |
| dac4_mult[3:0] | Multiplying factor on dac4_r_v_c digital signal before the D/A converters:<br>dac4_r_v_c value (in% of default value) | | |
| | Value | if R (conf_out=1x) | if V or C (conf_out=0) |
| | 0 0 0 0: | 80.49% | 75.000% |
| | 0 0 0 1: | 82.93% | 78.125% |
| | 0 0 1 0: | 85.37% | 81.250% |
| | 0 0 1 1: | 87.81% | 84.375% |
| | .. .. .. .. | ... | ... |
| | 1 0 0 0: [1] | 100% | 100% |
| | .. .. .. .. | | |
| | 1 1 1 1: | 117.07% | 121.875% |

1. DEFAULT mode when denc_nrst pin is active (LOW level)

## DEN_DAC6C DAC6 and C multiplying factors

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x13 | dac6_mult[3:0] | | | | c_mult[3:0] | | | |

Address: DencBaseAddress + 0x13
Type: Serial R/W
Reset value: 10000000

***Description***

| Bitfield | Description | | |
|---|---|---|---|
| c_mult[3:0] | Multiplying factor on C digital output (before D/A convertors) and on color part of CVBS signal: | | |
| | Value | Factor value (c_mult) | |
| | 0 0 0 0: | 1.000000 (1.000000 dec) | |
| | 0 0 0 1: | 1.000001 (1.015625 dec) | |
| | 0 0 1 0: | 1.000010 (1.031250 dec) | |
| | 0 0 1 1: | 1.000011 (1.046875 dec) | |
| | .. .. .. .. | ... | |
| | 1 1 1 1: | 1.001111 (1.234375 dec) | |
| dac6_mult[3:0] | Multiplying factor on dac6_b_cvbs digital signal before the D/A converters:<br>dac6_b_cvbs value (in% of default value) | | |
| | Value | If B(conf_out=1x) | If CVBS (conf_out=01) |
| | 0 0 0 0: | 80.49% | 75.000% |
| | 0 0 0 1: | 82.93% | 78.125% |
| | 0 0 1 0: | 85.37% | 81.250% |
| | 0 0 1 1: | 87.81% | 84.375% |
| | .. .. .. .. | ... | ... |
| | 1 0 0 0: [1] | 100% | 100% |
| | .. .. .. .. | | |
| | 1 1 1 1: | 117.07% | 121.875% |

Default peak to peak amplitude of U and V outputs corresponds to 70% of default Y or CVBS peak to peak amplitude if 100/0/100/0 color bar pattern is inputted. In other words, when Iref is set to deliver 1Vpp for CVBS on DAC3 for example (and dac3_mult = “1000”), when switched to U, DAC6 delivers 0.7 Vpp. If bit ‘uv_lev’ from register DEN_CFG7 is set, default peak to peak amplitude is 86% (80% if setupYUV=’1’) for V output and 67% (57% if setupYUV=’1’) for U output, of default Y or CVBS peak to peak amplitude if 100/0/100/0 color bar pattern is inputted, according to ITU-R 624-4 definition of UV signals.

Default peak to peak amplitude of RGB outputs corresponds to 70% of default Y or CVBS peak to peak amplitude if 100/0/75/0 color bar pattern is inputted. In other words, when Iref is set to deliver 1Vpp for CVBS on DAC3 for example (and dac3_mult = “1000”), when switched to B, DAC6 delivers 0.7 Vpp.

## DEN_LJPM1/2/3 CLIC_I_REG = ltarg[8:0] and lref[8:0]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x15 | ltarg8 | ltarg7 | ltarg6 | ltarg5 | ltarg4 | ltarg3 | ltarg2 | ltarg1 |
| 0x16 | ltarg0 | lref8 | lref7 | lref6 | lref5 | lref4 | lref3 | lref2 |
| 0x17 | lref1 | lref0 | - | - | - | - | - | - |

Address: DencBaseAddress+ 0x15(DEN_LJPM1), 0x16(DEN_LJPM2), 0x17(DEN_LJPM3)
Type: Serial R/W
Reset value: Undefined

These registers can be used to jump from a reference line (end of that line) to a target line of the SAME FIELD. However, not all lines can be skipped or repeated without problems. This functionality should be USED WITH CAUTION.

| Bitfield | Description |
|---|---|
| lref[8:0] | Binary format of the reference line from which a jump is required.<br>Default value: lref[8:0]:= 000000000 |
| ltarg[8:0] | Binary format of the target line. Default value: ltarg[8:0]:= 000000001 |

## DEN_CID CHIPID (read only): DENC version identification number

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x18 | CHIPID | | | | | | | |

Address: DencBaseAddress + 0x18
Type: Serial RO
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| CHIPID | 1001 0000 (for V9, 1000 0000 for V8) |

## DEN_VPS1 VPS: VPS Data registers

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x19 | s1 | s0 | xxx | xxx | cni3 | cni2 | cni1 | cni0 |
| 0x1A | np7 | np6 | d4 | d3 | d2 | d1 | d0 | m3 |
| 0x1B | m2 | m1 | m0 | h4 | h3 | h2 | h1 | h0 |
| 0x1C | min4 | min3 | min3 | min2 | min1 | min0 | c3 | c2 |
| 0x1D | c1 | c0 | np5 | np4 | np3 | np2 | np1 | np0 |
| 0x1E | pt7 | pt6 | pt5 | pt4 | pt3 | pt2 | pt1 | pt0 |

Address: DencBaseAddress + 0x19(DEN_VPS1), 0x1A(DEN_VPS2), 0x1B(DEN_VPS3), 0x1C(DEN_VPS4), 0x1D(DEN_VPS5), 0x1E(DEN_VPS6)

Type: Serial R/W

Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| pt[7:0] | Program type, binary |
| np[5:0] | Network or program CNI (Country and Network Identification) |
| c[3:0] | Country, binary |
| min[5:0] | Minute, binary |
| h [4:0] | Hour, binary |
| m[3:0] | Month, binary |
| d[4:0] | Day, binary |
| np[7:6] | Network or program CNI |
| cni[3:0] | 4 bits of CNI Reserved for enhancement of VPS |
| S[1:0] | Sounds<br>0 0: don’t know<br>0 1: mono<br>10: stereo<br>1 1: dual sound |

## DEN_CGMS1/2/3 CGMS_BIT[1:20]: CGMS data registers (20 bits only)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1F | - | - | - | - | b1 | b2 | b3 | b4 |
| 0x20 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 |
| 0x21 | b13 | b14 | b15 | b16 | b17 | b18 | b19 | b20 |

Address: DencBaseAddress + 0x1F (DEN_CGMS1), 0x20 (DEN_CGMS2), 0x21 (DEN_CGMS3)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| b15 - b20 | CRC (not internally computed) |
| b11 - b14 | Word2 |
| b7 - b10 | Word1 |
| b4 - b6 | Word0B |
| b1 - b3 | Word0A |

## DEN_TTX1-4/M TTX__[1:4]_DEF Teletext block definition

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x22 | fp_ttxt | ttxdel2 | ttxdel1 | ttxdel0 | ttx_l6 | ttx_l7 | ttx_l8 | ttx_l9 |
| 0x23 | ttx_l10 | ttx_l11 | ttx_l12 | ttx_l13 | ttx_l14 | ttx_l15 | ttx_l16 | ttx_l17 |
| 0x24 | ttx_l18 | ttx_l19 | ttx_l20 | ttx_l21 | ttx_l22 | ttx_l23 | ttx_l318 | ttx_l319 |
| 0x25 | ttx_l320 | ttx_l321 | ttx_l322 | ttx_l323 | ttx_l324 | ttx_l325 | ttx_l326 | ttx_l327 |
| 0x26 | ttx_l328 | ttx_l329 | ttx_l330 | ttx_l331 | ttx_l332 | ttx_l333 | ttx_l334 | ttx_l335 |

Address: DencBaseAddress + 0x22(DEN_TTX1), 0x23(DEN_TTX2), 0x24(DEN_TTX3), 0x25(DEN_TTX4), 0x26(DEN_TTXM)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ttx_lX | Each of these bits enables teletext on line X (ITU-R line numbering and 625 line systems).<br>Teletext line selections for other standards refer to the table below: |
| ttxdel[2:0] | Teletext data latency: The encoder clocks the first Teletext data sample on the (2+txdl[2:0])$^{th}$ rising edge of the master clock, following the rising edge of TTXS (teletext synchronization signal, supplied by the encoder). 1 0 0: default value |
| fp_ttxt | Full page teletext mode enable<br>0 0:[1] Disabled<br>0 1: Enabled |

1. Default mode when denc_nrst pin is active (low level).

| TTX__[1:4]_DEF bitfield | PAL BDGHIN line (CCIR 625 line numbering) | PAL M line (CCIR 525 line numbering) | NTSC M line (SMPTE 525 line numbering) |
|---|---|---|---|
| ttx_l6 | 6 | 6 | 9 |
| ttx_l7 | 7 | 7 | 10 |
| ttx_l8 | 8 | 8 | 11 |
| ttx_l9 | 9 | 9 | 12 |
| ttx_l10 | 10 | 10 | 13 |
| ttx_l11 | 11 | 11 | 14 |
| ttx_l12 | 12 | 12 | 15 |
| ttx_l13 | 13 | 13 | 16 |
| ttx_l14 | 14 | 14 | 17 |
| ttx_l15 | 15 | 15 | 18 |
| ttx_l16 | 16 | 16 | 19 |
| ttx_l17 | 17 | 17 | 20 |
| ttx_l18 | 18 | 18 | 21 |
| ttx_l19 | 19 | 19 | 22 |
| ttx_l20 | 20 | 20 | 23 |
| ttx_l21 | 21 | 21 | 24 |
| ttx_l22 | 22 | 22 | 25 |
| ttx_l23 | 23 | 23 | 26 |
| ttx_l318 | 318 | 268 | 271 |
| ttx_l319 | 319 | 269 | 272 |
| ttx_l320 | 320 | 270 | 273 |
| ttx_l321 | 321 | 271 | 274 |
| ttx_l322 | 322 | 272 | 275 |
| ttx_l323 | 323 | 273 | 276 |
| ttx_l324 | 324 | 274 | 277 |
| ttx_l325 | 325 | 275 | 278 |
| ttx_l326 | 326 | 276 | 279 |
| ttx_l327 | 327 | 277 | 280 |

**Table 46 Teletext block definition for PAL and NTSC standards**

| TTX__[1:4]_DEF bitfield | PAL_BDGHIN line (CCIR 625 line numbering) | PAL M line (CCIR 525 line numbering) | NTSC M line (SMPTE 525 line numbering) |
|---|---|---|---|
| ttx_l328 | 328 | 278 | 281 |
| ttx_l329 | 329 | 279 | 282 |
| ttx_l330 | 330 | 280 | 283 |
| ttx_l331 | 331 | 281 | 284 |
| ttx_l332 | 332 | 282 | 285 |
| ttx_l333 | 333 | 283 | 286 |
| ttx_l334 | 334 | 284 | 287 |
| ttx_l335 | 335 | 285 | 288 |

**Table 46 Teletext block definition for PAL and NTSC standards**

## DEN_CCF1 CCCF1: closed caption characters/extended data for field 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x27 | opc11 | c117 | c116 | c115 | c114 | c113 | c112 | c111 |
| 0x28 | opc12 | c127 | c126 | c125 | c124 | c123 | c122 | c121 |

Address: DencBaseAddress + 0x27 (DEN_CCF11), 0x28 (DEN_CCF12)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| c11[7:1] | First byte to encode in field1 |
| opc11 | Odd-parity bit of US-ASCII 7-bit character c11[7:1] |
| c12[7:1] | Second byte to encode in field1 |
| opc12 | Odd-parity bit of US-ASCII 7-bit character c12[7:1] |

- DEFAULT value: none, but closed captions enabling without loading these registers will issue a null character.
- DEN_CCF1 registers are never reset.

## DEN_CCF2 CCCF2: closed caption characters/extended data for field 2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x29 | opc21 | c217 | c216 | c215 | c214 | c213 | c212 | c211 |
| 0x2A | opc22 | c227 | c226 | c225 | c224 | c223 | c222 | c221 |

Address: DencBaseAddress + 0x29 (DEN_CCF21), 0x2A (DEN_CCF22)
Type: Serial R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| c21[7:1] | First byte to encode in field2 |
| opc21 | Odd-parity bit of US-ASCII 7-bit character c21[7:1] |
| c12[7:1] | Second byte to encode in field2 |
| opc22 | Odd-parity bit of US-ASCII 7-bit character c22(7:1) |

- DEFAULT value: none, but closed captions enabling without loading these registers will issue a null character.
- DEN_CCF2 registers are never reset.

## DEN_CLF1 CCLIF1: closed caption/extended data line insertion for field 1

This register programs the TV line number of the closed caption/extended data encoded in field 1:

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x2B | xx | xx | xx | l14 | l13 | l12 | l11 | l10 |

Address: DencBaseAddress + 0x2B
Type: Serial R/W
Reset value: 00001111

**525/60 system: (525-SMPTE line number convention)**

Only lines 10 to 22 should be used for closed caption or extended data services (lines 1 to 9 contain the vertical sync pulses with equalizing pulses).

| Bitfield | Description | |
|---|---|---|
| l1[4:0] | 0 0 0 0 0: | No line selected for closed caption encoding |
| | 0 0 0 x x: | Do not use these codes |
| | .... | |
| | i code line (i+6) | (SMPTE) selected for encoding |
| | .... | |
| | 11111: | line 37 (SMPTE) selected |

**625/50 system: (625-CCIR/ITU-R line number convention)**

Only lines 7 to 23 should be used for closed caption or extended data services.

| Bitfield | Description | |
|---|---|---|
| l1[4:0] | 0 0 0 0 0: | No line selected for closed caption encoding |
| | .... | |
| | i code line (i+6) | (CCIR) selected for encoding (i>0) |
| | .... | |
| | 11111: | line 37 (CCIR) selected |

DEFAULT value = 01111 line 21 (525/60, 525-SMPTE line number convention), which corresponds to line 21 in l625/50 system,(625-CCIR line number convention)

## DEN_CLF2 CCLIF2: closed caption/extended data line insertion for field 2

This register programs the TV line number of the closed caption/extended data encoded in field 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x2C | xx | xx | xx | l24 | l23 | l22 | l21 | l20 |

Address: DencBaseAddress + 0x2C
Type: Serial R/W
Reset value: 00001111

**525/60 system: (525-SMPTE line number convention)**

Only lines 273 to 284 should be used for closed caption or extended data services (preceding lines contain the vertical sync pulses with equalizing pulses), although it is possible to program over a wider range.????

| Bitfield | Description | |
|---|---|---|
| l2[4:0] | 0 0 0 0 0: | No line selected for closed caption encoding |
| | 0 0 0 x x: | Do not use these codes |
| | i line (269 +i) | (SMPTE) selected for encoding |
| | .... | |
| | 01111: | Line 284 (SMPTE) selected for encoding |
| | 11111: | Line 289 (SMPTE) |

*Note If cgms is allowed on lines 20 and 283 (525/60, 525-SMPTE line number convention), closed captions should not be programmed on these lines.*

**625/50 system: (625-CCIR line number convention)**

Only lines 319 to 336 should be used for closed caption or extended data services (preceding lines contain the vertical sync pulses with equalizing pulses), although it is possible to program over a wider range.????

| Bitfield | Description | |
|---|---|---|
| l2[4:0] | 0 0 0 0 0: | No line selected for closed caption encoding |
| | i line (318 +i) | (CCIR) selected for encoding |
| | .... | |
| | 10010: | Line 336 (CCIR) selected for encoding |
| | 11111: | Line 349 (CCIR) |

DEFAULT value:= 01111 line 284 (525/60, 525-SMPTE line number convention) this value also corresponds to line 333 in 625/50 system, (625-CCIR line number convention)

## DEN_REG 45...63 Reserved

## DEN_REG_64 TTX_CONF

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x40 | ttxt100IRE | ttxt_abcd[1:0] | | xxx | xxx | xxx | xxx | xxx |

Address: DencBaseAddress + 0x40
Type: Serial R/W
Reset value: 01010000

***Description***

| Bitfield | Description |
|---|---|
| ttxt_abcd[1:0] | Teletext standard selection<br>0 0 Teletext A<br>0 1 Teletext B Teletext B in 625 line systems is known as World System Teletext<br>1 0 [1] Teletext C Teletext C in 525 line systems is known as NABTS<br>1 1 Teletext D |
| ttxt100IRE | Teletext waveform amplitude<br>0[1] 70 IRE<br>1 100 IRE |

1. DEFAULT mode when denc_nrst pin is active (LOW level)

*Note Do not change the value of bits 4 to 0.*

## DEN_REG_65 DAC2MULT&TTXS

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x41 | dac2_mult[3:0] | | | | ttx_mask_off | xxx | bcs_en_4 | bcs_en_2 |

Address: DencBaseAddress + 0x41
Type: Serial R/W
Reset value: 10000010

***Description***

| Bitfield | Description |
|---|---|
| bcs_en_2 | Brightness, Contrast and Saturation control by registers DEN_REG_69 to DEN_REG_71 on 4:2:2 input<br>0:[1] Disable<br>1: Enable |
| bcs_en_4 | Brightness, Contrast and Saturation control by registers DEN_REG_69 to DEN_REG_71 on 4:4:4 input<br>0: Disable<br>1:[1] Enable |
| ttx_mask_off | Masking of 2 bits in Teletext framing code, depending on selected teletext standard<br>0:[1] Enable<br>1: Disable |
| dac2_mult[3:0] | Multiplying factor on dac1_c digital signal before the D/A converters with 3.125% step<br>Bitvalue dac2_c value compared to default<br>0 0 0 0: 75.000%<br>0 0 0 1: 78.125%<br>0 0 1 0: 81.250%<br>0 0 1 1: 84.375%<br>... ...<br>1 0 0 0:[1] 100%<br>...<br>1 1 1 1: 121.875% |

1. DEFAULT mode when denc_nrst pin is active (LOW level)

## DEN_REG_69 Brightness

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x45 | b[7:0] | | | | | | | |

Address: DencBaseAddress + 0x45
Type: Serial R/W
Reset value: 10000000

The register contents are used by the following formula to adjust the luminance intensity of the display video image:

$$Y_{out} = Y_{in} + b - 128$$

Where Yin is 8-bit input luminance and Yout is the result of ‘Brightness’ operation (still on 8 bits).

This value is saturated at 235 (16) or 254 (1) according to DEN_CFG6 bit “maxdyn”, b: brightness (unsigned value with center at 128, default 128)

## DEN_REG_70 Contrast

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x46 | c[7:0] | | | | | | | |

Address: DencBaseAddress + 0x46
Type: Serial R/W
Reset value: 00000000

The register contents are used by the following formula to adjust the relative difference between the display image higher and lower intensity luminance values:

$$Y_{out} = \frac{(Y_{in} - 128)(c + 128)}{128} + 128$$

Where, Yin is 8-bit input luminance, Yout is the result of ‘Contrast’ operation (still on 8 bits). This value is saturated at 235 (16) or 254 (1) according to DEN_CFG6 bit ‘maxdyn’, c: contrast (2’s complement value from -128 to 127, default 0)

## DEN_REG_71 Saturation

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x47 | s[7:0] | | | | | | | |

Address: DencBaseAddress + 0x47
Type: Serial R/W
Reset value: 10000000

The register contents are used by the following formula to adjust the color intensity of the displayed video image:

$$Cbout = \frac{s(Cbin - 128)}{128} + 128$$

$$Crout = \frac{s(Crin - 128)}{128} + 128$$

Where Crin and Cbin are the 8-bit input chroma, Crout and Cbout are the result of ‘Saturation’ operation (still on 8 bits)

This value is saturated at 240 (16) or 254 (1) according to DEN_CFG6 bit ‘maxdyn’, s: saturation value (unsigned value with centre at 128, default 128).

## DEN_YCOUT Make YC data bus accessible on PIO4

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x19F | Reserved | | | YCOUT | Reserved | | | |

Address: VideoBaseAddress + 0x19F
Type: R/W
Reset value: 0

| Bitfield | Description |
|---|---|
| YCOUT | 0: YC pads receive POI4 functional outputs<br>1: DENC is bypassed and YC data bus is accessible on PIO4 pads |

# 9 Programmable CPU memory interface (EMI)

## 9.1 DRAM registers

### EMI_CONFIGDATA0BANK0&1(DRAM format)

| | 15 14 | 13 | 12 | 11 10 | 9 8 | 7 6 5 | 4 3 | 2 1 0 |
|---|---|---|---|---|---|---|---|---|
| 0x00 & 0x10 | Reserved | DataDriveDelay | BusReleaseTime | SubBanks | SubBankSize | ShiftAmount | PortSize | DeviceType |

Address: *EMIBaseAddress* + 0x00 & 0x10
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| DeviceType | 000 = DRAM. Sets the format of the configuration register |
| PortSize | 00 = reserved, 01 = 32 bit, 10 = 16 bit, 11 = 8 bit |
| ShiftAmount | Column address width (0 = 7, 1 = 8...) |
| SubBankSize | 00 = 256k, 01 = 1M, 10 = 4M, 11 = 16M words |
| SubBanks | 00 =1, 01 = 2, 10 = 4, 11 = reserved |
| BusReleaseTime | 1 or 2 cycles |
| DataDriveDelay | 0-1 phases |

### EMI_CONFIGDATA1BANK0&1(DRAM format)

| | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0x04 & 0x14 | RASbits |

Address: *EMIBaseAddress* + 0x04 & 0x14
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| RASbits | Page address mask for address bits 22:7 |

The RASbits are used by the EMI to calculate whether the pending EMI access request is in the same page (row address) as the current access. If this is the case, the next access is a page mode access.

The 1s in the mask correspond to the row address bits to be compared, and are used to check that the pending access is in the same row as the current access. Therefore, a number of zeros are programmed in the LSBs of this register that correspond to the size of the DRAM column address. The remaining register bits are programmed to 1.

For example : Two 16Mbit 1M x 16 bit wide data port DRAMs are connected in parallel to make the memory 32 bits wide, and are connected onto an EMI bank configured as 32 bits wide. The DRAM has a row address size of 12 bits and a column address size of 8 bits.

The register should be programmed as : RASbits22:7 = 1111 1111 1111 1000 ( 0xFFF8)

This sets zeros in address lines MemAddr9-7. Address lines below MemAddr7 are by default zero. The zeros set the mask of the DRAM column size (8 bits). Zeros go from MemAddr9 to MemAddr2, as we are in 32 bit mode : this is 8 bits!

Note however that the data port size of the EMI bank affects the programming of the RASbits value.

For example : if instead one 16Mbit 1M x 16 bit wide data port DRAM, with a row address size of 12 bits and a column address size of 8 bits, is connected to the EMI, onto an EMI bank configured as 16 bits wide, the value will change to be :

RASbits22:7 = 1111 1111 1111 1100 ( #FFFC)

MemAddr bits 6 to 1 are zero by default (MemAddr1 is now included in the calculation as we are in 16 bit mode). Then zeros go from MemAddr8 to MemAddr1, as we are in 16 bit mode : this is then 8 bits (for an 8 bit column).

## EMI_CONFIGDATA2BANK0&1(DRAM format)

| | 15 | 14 13 12 11 10 9 8 | 7 6 5 | 4 3 2 1 | 0 |
|---|---|---|---|---|---|
| 0x08 & 0x18 | Reserved | RASBits | PrechargeTime | RefreshInterval | RefreshRASedge |

Address: *EMIBaseAddress* + 0x08 & 0x18
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| RefreshRASedge | 1 or 2 cycles after start of refresh |
| RefreshInterval | (1 - 16) * 128 cycles |
| PrechargeTime | 1 - 8 cycles |
| RASbits | Page address mask for address bits 29:23 Note: these bits should normally be set to all 1. |

## EMI_CONFIGDATA3BANK0&1(DRAM format)

| | 15 14 13 12 | 11 | 10 9 | 8 | 7 6 | 5 4 | 3 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x0C & 0x1C | Reserved | RAS-time | CAStime | RASe1Time | RASe2Time | CASe1Time | CASe2Time | Multi-byte | Latch-Point |

Address: *EMIBaseAddress* + 0x0C & 0x1C
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| LatchPoint | 0 = end of CAStime. 1 = 1 cycle before end of CAStime |
| Multibyte | When set, enables byte addressing using CAS strobes |
| CASe2Time | Rising edge of CAS. 0-3 phases before end of CAStime - in phases |
| CASe1Time | Falling edge of CAS. 1-4 phases after start of CAStime - in phases |
| RASe2Time | Rising edge of RAS. 0-3 phases before end of RAStime - in phases |
| RASe1Time | Falling edge of RAS. 1-2 phases after start of RAStime - in phases |
| CAStime | 2 cycles + 0-3 cycles |
| RAStime | 1 or 2 cycles |

## 9.2 Peripheral registers

### EMI_CONFIGDATA0BANK 0, 1, 2 & 3 (peripheral format)

| 15 14 13 | 12 11 | 10 9 | 8 7 | 6 5 | 4 3 | 2 1 0 |
|---|---|---|---|---|---|---|
| DataDriveDelay | BusRelease-Time | CSactive | OEactive | BEactive | Portsize | DeviceType |

Address: *EMIBaseAddress* + 0x00, 0x10, 0x20 & 0x30
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| DeviceType | 001 = peripheral. Sets the format of the configuration register |
| Portsize | 0 = reserved, 01 = 32 bit, 10 = 16 bit, 11 = 8 bit |
| BEactive | 00: Inactive<br>01: Active during read<br>10: Active during write<br>11: Active during read and write |
| OEactive | 00: Inactive<br>01: Active during read<br>10: Active during write<br>11: Active during read and write |
| CSactive | 00: Inactive<br>01: Active during read<br>10: Active during write<br>11: Active during read and write |
| BusReleaseTime | 0-3 cycles |
| DataDriveDelay | 0-7 phases |

### EMI_CONFIGDATA1BANK 0, 1, 2 & 3 (peripheral format)

| 15 14 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|
| AccessTimeRead | CSe1TimeRead | CSe2TimeRead | OEe1TimeRead | OEe2TimeRead | BEe1TimeRead | BEe2TimeRead |

Address: *EMIBaseAddress* + 0x04, 0x14, 0x24 & 0x34
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| BEe2TimeRead | Rising edge of BE. 0-3 phases before end of access cycle |
| BEe1TimeRead | Rising edge of BE. 0-3 phases after start of access cycle |
| OEe2TimeRead | Rising edge of OE. 0-3 phases before end of access cycle |
| OEe1TimeRead | Falling edge of OE. 0-3 phases after start of access cycle |
| CSe2TimeRead | Rising edge of CS. 0-3 phases before end of access cycle |
| CSe1TimeRead | Falling edge of CS. 0-3 phases after start of access cycle |
| AccessTimeRead | 2 cycles + 0-15 cycles |

## EMI_CONFIGDATA2BANK 0, 1, 2 & 3 (peripheral format)

| 15 14 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|
| AccessTimeWrite | CSe1TimeWrite | CSe2TimeWrite | OEe1TimeWrite | OEe2TimeWrite | BEe1TimeWrite | BEe2TimeWrite |

Address: *EMIBaseAddress* + 0x08, 0x18, 0x28 & 0x38
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| BEe2TimeWrite | Rising edge of BE. 0-3 phases before end of access cycle |
| BEe1TimeWrite | Rising edge of BE. 0-3 phases after start of access cycle |
| OEe2TimeWrite | Rising edge of OE. 0-3 phases before end of access cycle |
| OEe1TimeWrite | Falling edge of OE. 0-3 phases after start of access cycle |
| CSe2TimeWrite | Rising edge of CS. 0-3 phases before end of access cycle |
| CSe1TimeWrite | Falling edge of CS. 0-3 phases after start of access cycle |
| AccessTimeWrite | 2 cycles + 0-15 cycles |

## EMI_CONFIGDATA3BANK 0, 1, 2 & 3 (peripheral format)

| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | |
|---|---|
| Reserved | LatchPoint |

Address: *EMIBaseAddress* + 0x0C, 0x1C, 0x2C & 0x3C
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| LatchPoint | 0 = end of access cycle,<br>1 = 1 cycle before end of access cycle,<br>2 = 2 cycles before end of access cycle |

## 9.3 SDRAM registers

### EMI_CONFIGDATA0BANK0&1(SDRAM format)

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 & 0x10 | Reserved | | DataDriveDelay | BusReleaseTime | SubBanks | | SubBankSize | | ShiftAmount | | | PortSize | | DeviceType | | |

Address: *EMIBaseAddress* + 0x00 & 0x10
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| DeviceType | 010 = SDRAM. Sets the format of the configuration register |
| PortSize | 00 = reserved, 01 = 32 bit, 10 = 16 bit, 11 = 8 bit |
| ShiftAmount | Column address width (0 = 7, 1 = 8...) |
| SubBankSize | 00 = 16Mbit, 01 = 32 Mbit, 10 =64 Mbit, 11 = 128 Mbit |
| SubBanks | 00 =1, 01 = 2, 10 = 4, 11 = reserved |
| BusReleaseTime | 1 or 2 cycles |
| DataDriveDelay | 0-1 phases |

### EMI_CONFIGDATA1BANK0&1(SDRAM format)

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 & 0x14 | RASbits | | | | | | | | | | | | | | | |

Address: *EMIBaseAddress* + 0x04 & 0x14
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| RASbits | Page address mask for address bits 22:7 |

The RASbits are used by the EMI to calculate if the pending EMI access request is in the same page (row address) as the current access. If this is the case, the next access is a page mode access (where page mode for SDRAM means that sequential read/write commands can be made without a preceeding row activate command).

The 1s in the mask correspond to the row address bits to be compared, and are used to check that the pending access is in the same row as the current access. Therefore, a number of zeros are programmed in the LSBs of this register that correspond to the size of the SDRAM column address. The remaining register bits are programmed to 1.

For example : a 64Mbit 32 bit wide data port SDRAM with a row address size of 12 bits and a column address size of 8 bits, configured onto an EMI bank 32 bits wide.

The register must be programmed as : RASbits22:7 = 1111 1111 1111 1000 ( 0xFFF8)

This sets zeros in address lines MemAddr9-7. Address lines below MemAddr7 are by default zero. The zeros set the mask of the SDRAM column size (8 bits). Zeros go from MemAddr9 to MemAddr2, as we are in 32 bit mode : this is 8 bits!

Note however that the data port size of the EMI bank affects the programming of the RASbits value.

For example : if instead a 64Mbit 16 bit wide data port SDRAM with a row address size of 12 bits and a column address size of 8 bits, configured onto an EMI bank 16 bits wide, the value will change to be :

RASbits22:7 = 1111 1111 1111 1100 ( #FFFC)

MemAddr bits 6 to 1 are zero by default (MemAddr1 is now included in the calculation as we are in 16 bit mode). Then zeros go from MemAddr8 to MemAddr1, as we are in 16 bit mode : this is then 8 bits (for an 8 bit column).

## EMI_CONFIGDATA2BANK0&1(SDRAM format)

| | 15 | 14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 | 0 |
|---|---|---|---|---|
| 0x08 & 0x18 | Reserved | RASBits | PrechargeTime | RefreshInterval | reserved |

Address: *EMIBaseAddress* + 0x08 & 0x18
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| RefreshInterval | (1 - 16) * 128 cycles |
| PrechargeTime | 1 - 8 cycles |
| RASbits | Page address mask for address bits 29:23. Note: these bits should normally be set to all 1. |

## EMI_CONFIGDATA3BANK0&1(SDRAM format)

| | 15 | 14 | 13 12 11 10 | 9 8 | 7 6 | 5 4 | 3 | 2 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x0C & 0x1C | Reserved | ModeSet-Delay | RefreshTime | ActivateTo-Read | ActivateToWrite | CASlatency | SDram Banks | WriteRecovery-Time | Res. |

Address: *EMIBaseAddress* + 0x0C & 0x1C
Type: R/W

***Description***

| Bitfield | Definition |
|---|---|
| WriteRecoveryTime | 00 = reserved, 01 = 1 cycle, 10 = 2 cycles, 11 = 3 cycles |
| SDRAM Banks | The number of internal memory banks in SDRAM organisation<br>0 = 2 SDRAM banks (16Mbit devices), 1 = 4 SDRAM banks (64Mbit devices) |
| CASlatency | 1 to 4 cycles |
| ActivateToWrite | 1 or 4 cycles |
| ActivateToRead | 1 to 4 cycles |
| RefreshTime | 1 to 16 cycles |
| ModeSetDelay | 3 or 4 cycles |

## EMI_CONFIGLOCKBANK0-3

| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 | 0 |
|---|---|
| not used | EMIConfigData0-3 |

Address: *EMIBaseAddress* + 0x40, 0x44, 0x48 & 0x4C
Type: Write only

***Description***

| Bitfield | Definition |
|---|---|
| EMIConfigData0-3 | Write protection, when set this bit makes the bank read only |

## EMI_CONFIGSTATUS EMI status

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x50 | Reserved | | | | | | | | EMI_ConfigStatus[7:0] | | | | | | | |

Address: *EMIBaseAddress* + 0x50
Type: Read only

***Description***

| Bitfield | Definition |
|---|---|
| EMI_ConfigStatus[0] | When set all the four configuration registers for bank0 have been written at least once |
| EMI_ConfigStatus[1] | When set all the four configuration registers for bank1 have been written at least once |
| EMI_ConfigStatus[2] | When set all the four configuration registers for bank2 have been written at least once |
| EMI_ConfigStatus[3] | When set all the four configuration registers for bank3 have been written at least once |
| EMI_ConfigStatus[4] | When set configuration registers for bank0 locked |
| EMI_ConfigStatus[5] | When set configuration registers for bank1 locked |
| EMI_ConfigStatus[6] | When set configuration registers for bank2 locked |
| EMI_ConfigStatus[7] | When set configuration registers for bank3 locked |

## EMI_DRAMINITIALIZE SDRAM system intialize

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x60 | Reserved | | | | | | | | | | | | | | | Bit0 |

Address: *EMIBaseAddress* + 0x60
Type: Write only

***Description***

In Slave Mode, the EMI3 should not initialize DRAM or SDRAM devices in the system, nor should it refresh any S/DRAM device. Therefore this register should never be written.

| Bitfield | Definition |
|---|---|
| Bit0 | Initializes the system SDRAM |

## EMI_SDRAMMODEREG0 SDRAM mode register

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x58 | Reserved | | | | | | | | | Latency mode | | | Burst type | | Burst length | |

Address: *EMIBaseAddress* + 0x58
Type: W
Reset value: 0x22

***Description***

This register is used by the EMI to program the SDRAM mode register. This is performed during the SDRAM initialize sequence, started by a write to the EMI_DRAMINITIALIZE register. The EMI performs a mode register set command to the SDRAM, in which the data in the EMI_SDRAMMODREG0 register is copied onto the bottom 16 address lines of the EMI.

SDRAM mode register for a single SDRAM bank (bank 0 or 1), or for SDRAM bank 0 when there are two SDRAM banks programmed in the EMI.

| Bitfield | Descritpion |
|---|---|
| Burst length | Default 010 : this field is not required to be programmed, as the EMI does not use SDRAM burst mode |
| Burst type | Default 0 : this field is not required to be programmed, as the EMI does not use SDRAM burst mode |
| Latency mode | Default 010 : this latency value must match the value programmed into the CASlatency field in the corre-sponding EMI_CONFIGDATA3 register, for the EMI bank being used<br>Valid codes for SDRAM are : 010 = SDRAM CAS latency of 2, 011 = SDRAM CAS latency of 3 |

## EMI_SDRAMMODEREG1 SDRAM mode register

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x5C | Reserved | | | | | | | | | Latency mode | | | Burst type | Burst length | | |

Address: *EMIBaseAddress* + 0x5C
Type: W
Reset value: 0x22

***Description***

This register is used by the EMI to program the SDRAM mode register. This is performed during the SDRAM initialize sequence, started by a write to the EMI_DRAMINITIALIZE register. The EMI performs a mode register set command to the SDRAM, in which the data in the EMI_SDRAMMODREG1 register is copied onto the bottom 16 address lines of the EMI.

SDRAM mode register for SDRAM bank 1, when there are two SDRAM banks programmed in the EMI.

| Bitfield | Descritpion |
|---|---|
| Burst length | Default 010 : this field is not required to be programmed, as the EMI does not use SDRAM burst mode |
| Burst type | Default 0 : this field is not required to be programmed, as the EMI does not use SDRAM burst mode |
| Latency mode | Default 010 : this latency value must match the value programmed into the CASlatency field in the EMI_CONFIGDATA3 register, for EMI bank 1.<br>Valid codes for SDRAM are : 010 = SDRAM CAS latency of 2, 011 = SDRAM CAS latency of 3 |

## EMI_CONFIGPADLOGIC_EMI configuration

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x70 | WaitCycles | | Reserved | | CPADL [11] | Res. | CPADL[9:7] | | | Res. | CPADL [5] | Res. | CPADL[3:1] | | | CPADL [0] |

Address: *EMIBaseAddress* + 0x70
Type: R/W
Reset value: 0

### *Description*

| Bitfield | Definition |
|---|---|
| CPADL[0] | 0: cas0 mapped on notCPU_CAS0<br>1: cpu_addr[22] mapped on notCPU_CAS0 |
| CPADL[3:1] | Encode DRAM or SDRAM configuration & sub-decoding of bank3[1]:<br>000: NO (S)DRAM - sub-decode bank3 not enabled<br>001: NO (S)DRAM - sub-decode bank3 enabled<br>010: 1 (S)DRAM in bank0 - sub-decode bank3 not enabled<br>011: 1 (S)DRAM in bank0 - sub-decode bank3 enabled<br>100: 1 (S)DRAM in bank1 - sub-decode bank3 not enabled<br>101: 1 (S)DRAM in bank1 - sub-decode bank3 enabled<br>110: (S)DRAM in bank0 with sub-decoding - sub-decode bank3 not enabled<br>111: 2(S)DRAM in bank0 and bank1 - sub-decode bank3 not enabled |
| CPADL[5] | 0: asynchronous MemWait<br>1: synchronous MemWait |
| CPADL[9:7] | SubBankSize in bank3:<br>000=4Mbit, 001=8Mbit, 010=16Mbit, 011=32Mbit, 100=64Mbit, 101=1Gbit, 110,111=Reserved |
| CPADL[11] | 0: no SDRAM in the EMI, 1: SDRAM in the EMI |
| WaitCycles | Sets the number of EMI wait cycles to be inserted during peripheral access to the upper part of EMIbank3:<br>00: Zero wait cycles, 01: 1 wait cycle, 10: 2 wait cycles, 11: 3 wait cycles |

1. (S)DRAM means DRAM or SDRAM: it is not possible to have SDRAM and DRAM in the system at the same time: so the configuration refer to only DRAM OR only SDRAM in bank0,1.

# 10 Front end interface (FEI)

## 10.1 ATAPI interface

### FEI_ATAPI_CFG ATAPI interface configuration

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 | 7 | 6 5 4 3 2 | 1 | 0 |
|---|---|---|---|---|---|
| 0x7A0 | Reserved | READY | ADDR_POL | ENABLE | PADS_CTL |

Address: *LinkBaseAddress +* 0x7A0
Type: R/W except for bit 7 which is read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| PADS_CTL | 0: Set ATAPI mode inactive, 1: Set ATAPI DIOR/DIOW pads active.<br>The ST20 PIO1 and PIO2 must be programmed as outputs when using the ATAPI interface. |
| ENABLE | 0: Disables ATAPI interface, 1: Enables ATAPI interface (Not used for STB applications) |
| ADDR_POL | Configurethe ATAPI interface (these bits correspond to EMI_addr<20:16>). |
| READY | This bit is read only<br>0: Not ready for new sector, 1: Ready for new sector |

## 10.2 Sector processor

The addresses below are quoted as the offset to the LinkBaseAddress.

### FEI_SE_RST Reset module

| 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 | 0 |
|---|---|
| | SE_RST[0] |

Address: *LinkBaseAddress +* 0x800
Type: W
Reset value: 1

***Description***

| Bitfield | Description |
|---|---|
| SE_RST[0] | 1: The sector processor and all FEI are reset. This bit must be used if parallel or FEC (in DVD mode) inputs are also selected.<br>0: This BID is write only. This bit resets all the FEI except decryption cell. |

### FEI_SE_SAC Sector address check

| 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 | 0 |
|---|---|
| | SE_SAC[0] |

Address: *LinkBaseAddress +* 0x804

Type: R/W
Reset value: 1

***Description***

| Bitfield | Description |
|---|---|
| SE_SAC[0] | 0 : no check 1 : continuity check |

## FEI_SE_MOD Sector processor mode

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SE_MOD[1:0] |

Address: *LinkBaseAddress +* 0x808
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| SE_MOD[1:0] | 00: DVD<br>01: VCD Mode 2 form 2<br>10: VCD Mode 2 form 1<br>11: CD-DA |

## FEI_SE_CAPEN Capturing enable

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SE_CAPEN[0] |

Address: *LinkBaseAddress +* 0x080C
Type: R/W
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| SE_CAPEN[0] | 0 : Disable Capturing (sector synchronized)<br>1 : Enable Capturing (sector synchronized) |

## FEI_SE_EMR Error mode

Address: *LinkBaseAddress +* 0x810
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| SE_EMR[0] | 0 : Continue<br>1 : Stop. To restart the sector processor, SE_CAPEN[0] has to be set |

## FEI_SE_IR Interrupt

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | SE_IR[7:0] | | | | | | | |

Address: *LinkBaseAddress +* 0x814
Type: R/W
Reset value: 0

***Description***

Write : enable interrupt.

Read : interrupt status and interrupt clear.

| Bitfield | Description |
|---|---|
| SE_IR[0] | CAPEND, sector with address in SE_LST minus one is captured |
| SE_IR[1] | SECCAP, sector with address in SE_CAP is captured |
| SE_IR[2] | RDERR, read error |
| SE_IR[3] | NPRENAV, New Pre NV_PCK |
| SE_IR[4] | EOF, VCD End Of File |
| SE_IR[5] | EOR, VCD End Of Record |
| SE_IR[6] | T, VCD Trigger |
| SE_IR[7] | Short Sector |

## FEI_SE_STAT Status

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SE_STAT[0] |

Address: *LinkBaseAddress +* 0x818
Type: R
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| SE_STAT[0] | 1 : Busy reading<br>0 : Not busy reading |
| SE_STAT[1] | 1 : Busy capturing<br>0 : Not capturing |

## FEI_SE_CAP_H Address of first sector /subcode block to be captured

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | SE_CAP_H | | | | | | | |

Address: *LinkBaseAddress +* 0x820
Type: R/W
Reset value: Undefined

## FEI_SE_CAP_L Address of first sector /subcode block to be captured

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | SE_CAP_L | | | | | | | |

Address: *LinkBaseAddress +* 0x824
Type: R/W
Reset value: Undefined

***Description***

These two registers program the first sector / subcode block to be captured. The value type depends on the sector processor mode : DVD (24 bits sector number in DVD sector header), VCD (24 bits MSF in VCD sector header), CD-DA (24 bits MSF in Q sub channel).

SE_CAP_H = SE_CAP [23...16]

SE_CAP_L = SE_CAP [15...0]

## FEI_SE_LST_H Address of (last + 1) sector /subcode block last to be captured

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | SE_LST_H | | | | | | | |

Address: *LinkBaseAddress +* 0x828
Type: R/W
Reset value: Undefined

## FEI_SE_LST_L Address of (last + 1) sector /subcode block to be captured

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | SE_LST_L | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x82C
Type: R/W
Reset value: Undefined

***Description***

These two registers are used to program the last sector / subcode block of the range to be captured.

This sector is not captured. Capture ends at LST-1. The type of value depends on sector processor mode : DVD (24 bits sector number in DVD sector header), VCD (24 bits MSF in VCD sector header), CD-DA (24 bits MSF in Q sub channel).

SE_LST_H=SE_LST [23...16]

SE_LST_L=SE_LST [15...0]

## FEI_SE_CUR_H Address of actual sector /subcode block actually entering the SP

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | SE_CUR_H | | | | | | | |

Address: *LinkBaseAddress +* 0x830
Type: R
Reset value: Undefined

## FEI_SE_CUR_L Address of actual sector /subcode block entering the SP

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | SE_CUR_L | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x834
Type: R
Reset value: Undefined

***Description***

These two registers are used to read the current sector / subcode block address entering the SP.

The type of read data depends on sector processor mode : DVD (24 bits sector number in DVD sector header), VCD (24 bits MSF in VCD sector header), CD-DA (24 bits MSF in Q sub channel).

SE_CUR_H=SE_CUR [23...16]

SE_CUR_L=SE_CUR [15...0]

### FEI_SE_HE_H CPR_MAI bits of current sector actually entering the SP

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | SE_HE_REG_H | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x860
Type: R/W
Reset value: Undefined

***Description***

SE_HE_REG_H[15...0]=CPR_MAI[47...32]

### FEI_SE_HE_M CPR_MAI bits of current sector actually entering the SP

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | SE_HE_REG_M | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x864
Type: R/W
Reset value: Undefined

***Description***

This register is only significant in DVD Mode. SE_HE_REG_M[15...0]=CPR_MAI[31...16]

### FEI_SE_HE_L CPR_MAI bits of current sector actually entering the SP

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | SE_HE_REG_L | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x868
Type: R/W
Reset value: Undefined

***Description***

This register is only significant in DVD Mode: SE_HE_REG_L[15...0]=CPR_MAI[15...0]

## 10.3 Front-end interface

### FEI_REV Revision ID

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | REV | | | | | | | |

Address: *LinkBaseAddress +* 0x79C
Type: RO

***Description***

This register gives the Front End Interface revision. For STi5518 cut A0, the value is 0x30. For susequent cuts, the value is given in application note 7152120, STi5518 How to move from cutA0.

## FEI_GCF Front-end interface configuration

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | FEI[15:0] | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0x790
Type: R/W
Reset value: 0x01E0

***Description***

| Bitfield | Default | Description |
|---|---|---|
| FEI_GCF[0] | 0 | If set to 1, the DVD front-end-interface is bypassed. Data goes directly from the FEC interface to the acquisition RAM. This bit must be programmed to 1 to ensure compatability with the STi5500. |
| FEI_GCF[1] | 0 | If set to 1, the data are input using the parallel interface. |
| FEI_GCF[2] | 0 | If set to 0, the next sector will not be taken into account. This flag is usefull when the track-buffer is full. When this bit is set to 0, the current read sector n is not affected. To restart the sector processor (if used), it is necessary to set SE_CAPEN[0] and to send from the basic engine a sequence containing at least sector n (if addresses of sectors to be captured have not been modified). In case of parallel interface, reset ting this bit means desactivating PARA_REQ output signal. In case of FEC interface, this bit has no effect since there is no handshake. |
| FEI_GCF[3] | 0 | Polarity of PARA_STR input signal of parallel interface. If set to 1, rising edge is used to strobe the data. |
| FEI_GCF[4] | 0 | Polarity of PARA_REQ output signal of parallel interface. If set to 1, PARA_REQ = 1 means that the device can accept data. |
| FEI_GCF[5] | 1 | Decryption cell is bypassed. |
| FEI_GCF[6] | 1 | 1: The I2S interface enters the sector processor<br>0: The serial interface (or FEC interface) enters the sector processor (DVD mode only) |
| FEI_GCF[7] | 1 | Decryption cell software reset. This bit must be set and then reset to reset the decryption cell. The keys are lost on this action. After a hard reset, this bit is automatically set. |
| FEI_GCF[8] | 1 | Pin 20 In/Out. If set to 1, Pin 20 is an input, if not the Pin 20 is on output. |
| FEI_GCF[9] | 0 | Use NRSS - this bit must be used in conjunction with FEI_GCF[8]. |
| FEI_GCF[10] | 0 | When FEI_GCF[11] is set, this bit is controls the active polarity of the external start sector signal. |
| FEI_GCF[11] | 0 | External start sector signal (PARA_SYNC or SER_SYNC)<br>If set to 1, the external start-sector signal generates an internal start-sector signal. |
| FEI_GCF[12] | 0 | If set to 1, data coming from the FEC interface enters the CSS decryption block, FEI_GCF[13] and FEI_SLG must be programmed accordingly. |
| FEI_GCF[13] | 0 | 1: The first 12 bytes and last bytes (after 2060) are not sent to the decryption cell. However, these bytes will be transfered by DMA. This initialization is done accordingly to FEI_SLG and FEI_GCF[12] registers.<br>0: All sector bytes enter the decryption cell (FEI_SLG must be set to 2048) |
| FEI_GCF[14] | 0 | Polarity of PARA_DVALID output signal of parallel interface.<br>If set to 1, PARA_DVALID = 1 means data sent by the front end are valid. |
| FEI_GCF[15] | 0 | Use of PARA_DVALID signal. If set to 1, PARA_DVALID controls parallel data input. |

After programming this register, perform a soft-reset using register FEI_SE_RST.

### FEI_SLG

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | Sector_length | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x798
Type: R/W
Reset value: Undefined

***Description***

This register is only used when data are input using the parallel interface. The value to be written is the same than the value to be written in the "DMA_PLG" register. This value is used internally to generate a sector start synchronisation, what is done automatically when the sector-processor is used.

### FEI_SUB

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | Subcodes | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress +* 0x788
Type: R
Reset value: Undefined

***Description***

This register contents the 16 bits subcode value from the subcode FIFO. Reading this register increments the read pointer of the subcode FIFO. If the FIFO is empty, the read cycle is not granted. It is necessary to read FEI_SFF before accessing FEI_SUB to be sure subcodes are available.

### FEI_SFF

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | Level | | | | | | | |

Address: *LinkBaseAddress +* 0x78C
Type: R
Reset value: Undefined

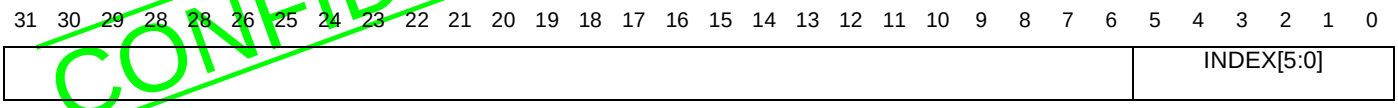
***Description***

This register indicates the actual filling level of the subcode FIFO (number of 16 bits words).

## 10.4 Decryption-cell Registers

The decryption cell internal registers are accessed by first writting an index to the Index Register (DEC_IDX) and then reading or writing data to the Data Register (DEC_DAT).

## FEI_DEC_IDX

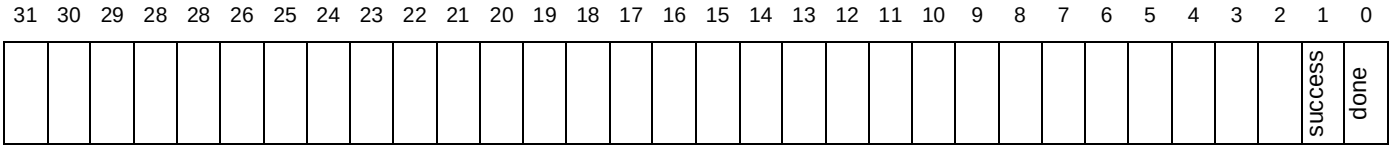



Address: *LinkBaseAddress +* 0x784
Type: W
Reset value: Undefined

***Description***

This index register is used to select intermediate key register, internal key load register, temporary word register or internal function register.

## FEI_DEC_STA

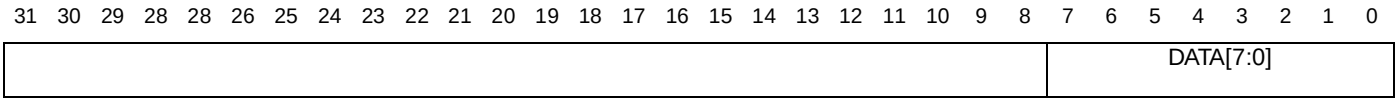



Address: *LinkBaseAddress +* 0x784
Type: R
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| done | Indicates completion of a function. A logic 1 indicates the function is complete. |
| success | Indicates the success or the failure of the function being performed. The value is only valid after the function is complete. This bit is valid only for function 4. A logic 1 indicates success. |

## FEI_DEC_DAT




Address: *LinkBaseAddress +* 0x780
Type: R/W
Reset value: Undefined

***Description***

This data register is used to load intermediate key, internal keys, temporary word, to select internal functions.

## FEI_TWH Temporary word high

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TEMP_HIGH | | | | | | | |

Address: *LinkBaseAddress +* 0 x 10, 11, 12, 13, 14
Type: W0
Reset value: Undefined

***Description***

This register is loaded with input values by the ST20 and is the source register for DVD ROM functions. This register is write only.

## FEI_TWL Temporary word low

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TEMP_LOW | | | | | | | |

Address: *LinkBaseAddress +* 0 x 18, 19, 1A, 1B, 1C
Type: WO
Reset value: Undefined

***Description***

This register is loaded with input values by the ST20 and is the source register for DVD ROM functions. This register is write only.

## FEI_IKL Internal key load register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| KEY[7:0] | | | | | | | |

Address: *LinkBaseAddress +* 0x07
Type: W
Reset value: Undefined

***Description***

This register is used in function 3 as a mean to lead the disk key into the device for later decryption. This register is write only. The key is loaded 1 byte at a time by writing each byte in turn to this register.

## FEI_IK Intermediate key

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INTER_KEY[7:0] | | | | | | | |

Address: *LinkBaseAddress +* 0 x 00, 01, 02, 03, 04
Type: W, R under conditions
Reset value: Undefined

***Description***

This register is a temporary result register used to store transient and intermediate values. This register is writeable but readable only at the completion of function 1. This register is loaded with the least significant byte at the lowest address. This register is used only with a DVD ROM drive.

## FEI_IF Internal function

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | Option | | Mult | Function | | |

Address: *LinkBaseAddress +* 0x3F
Type: W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| Function[2:0] | 0 : Verify drive response (use for DVD ROM only)<br>1 : Generate decoder response (use for DVD ROM only)<br>2 : Generate bus key (use for DVD ROM only)<br>3 : Load Keys<br>4 : Decrypt Disc Keys<br>5 : Decrypt Title Keys<br>6 : Decrypt Data<br>7 : Reset (will abort all other modes - keys preserved) |
| Mult | Writing a 1 in this bit position causes the cell to reinitialize at the end of a block of data and continue decrypting data. Writing a 0 will cause the device to terminate decryption at the end of the current block. |
| Option | Selects function 6 option of full DVD blocks or pre-parsed blocks with headers removed. ????????? |
| Option[5:4] | 00 : Fixed length - full length block (2048 bytes).<br>01 : Fixed length - pack header removed (2034 bytes). |

# 11 Interrupt control registers (INC)

## INC_CLEAR_EXEC Clear a bit of the Exec register

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | Clr | Reserved | | | | | | | | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr |

Address: *IntControllerBase* + 0x108
Type: Write only

***Description***

When set, bits 0-7 of this register clear the corresponding interrupt exec bit (IntEx) in register INC_Exec. There is one bit for each of the 8 interrupt levels, where bit 0 corresponds to bit INTEx0 etc... Bit 16 of this register clears all of the bits of the INC_Exec register.

Note, users must not use this register, as its operation is undefined.

## INC_CLEAR_MASK Clear a bit of the interrupt enable mask

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | Clr | Reserved | | | | | | | | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr |

Address: *IntControllerBase* + 0xC8
Type: Write only

***Description***

When set, bits 0-7 of this register clear the corresponding interrupt enable bit (IntEn) in register INC_Mask. There is one bit for each of the 8 interrupt levels, where bit 0 corresponds to bit IntEn0 etc... Bit 16 of this register clears all of the bits of the INC_Mask register.

## INC_CLEAR_PENDING Clear a bit of the Pending register

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | Clr | Reserved | | | | | | | | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr |

Address: *IntControllerBase* + 0x88
Type: Write only

***Description***

When set, bits 0-7 of this register clear the corresponding interrupt pending bit (PendInt) in register INC_Pending. There is one bit for each of the 8 interrupt levels, where bit 0 corresponds to bit PendInt0 etc... Bit 16 of this register clears all of the bits of the INC_Pending register.

## INC_EXEC Interrupts executing

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | IntEx7 | IntEx6 | IntEx5 | IntEx4 | IntEx3 | IntEx2 | IntEx1 | IntEx0 |

Address: *IntControllerBase* + 0x100
Type: Read/write
Reset value: 0

***Description***

This register records whether interrupt levels 0-7 are currently executing and pre-empting interrupts. The bit is set when the CPU starts running code for that interrupt, where IntEx0 corresponds to interrupt level 0 etc... The highest priority interrupt bit is reset once the interrupt handler executes a return from interrupt instruction (*iret*).

This register is specified as being read/write, but users should not attempt to write to this register, as the device operation is undefined.

## INC_HANDLERWPTR*n* Interrupt handler work space pointer

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | HandlerWptr0[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x04 | HandlerWptr1[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x08 | HandlerWptr2[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x0C | HandlerWptr3[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x10 | HandlerWptr4[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x14 | HandlerWptr5[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x18 | HandlerWptr6[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |
| 0x1C | HandlerWptr7[31:2] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | P |

Address: *IntControllerBase* + 0x00 (INC_HandlerWptr0) to 0x1C (INC_HandlerWptr7)
Type: Read/write
Reset value: Undefined

***Description***

This register points to the work space of the corresponding interrupt handler (HandlerWptr0 corresponds to interrupt level0...). The base of the work space is 32-bit word aligned, so the two least significant bits of the 32-bit address are always zero, and are not held. Each register contains a priority bit **P** which determines whether the interrupt is at a higher or lower priority than the high-priority process queue.

Before the interrupt is enabled by writing a 1 in the INC_Mask register, the software must ensure that there is a valid HandlerWptr in the register.

| Bitfield | Description |
|---|---|
| P | Sets the priority of the interrupt. If this bit is set to 0, the interrupt is a higher priority than the high priority process queue; if this bit is 1, the interrupt is a lower priority than the high priority process queue. 0: High priority, 1: Low priority |
| HandlerWptr[31:2] | The 30 most significant bits of the address of the work space of the interrupt handler. |

## INC_MASK Interrupt enable mask

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xC0 | Reserved | | | | | | | | | | | | | | | GloEn | Reserved | | | | | | | | IntEn7 | IntEn6 | IntEn5 | IntEn4 | IntEn3 | IntEn2 | IntEn1 | IntEn0 |

Address: *IntControllerBase* + 0xC0
Type: Read/write
Reset value: 0

***Description***

This register selectively enables or disables interrupts from the interrupt level controller, connected to each of the eight interrupt levels. The global interrupt disable bit, disables all interrupt levels, whatever the state of the individual interrupt mask bits.

The INC_Pending register contains a pending flag for each interrupt level. The INC_Mask register masks the INC_Pending register to control what interrupts the CPU, while continually monitoring interrupts.

On start-up, INC_Mask is initialized to zeros so all interrupts are disabled both globally and individually. When a 1 is written to the GloEn bit, the individual interrupt channels are still disabled. To enable an interrupt channel, a 1 must also be written to the corresponding IntEn bit.

| Bitfield | Description |
|---|---|
| GloEn | 1: Interrupt setting determined by the corresponding IntEn bit.<br>0: All interrupts disabled |
| IntEn[7:0] | 1: Interrupt enabled, 0: Interrupt disabled. |

INC_Mask is mapped onto registers INC_Set_Mask and INC_Clear_Mask so that bits can be set or cleared individually.

## INC_PENDING Interrupt pending

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x80 | PendInt7 | PendInt6 | PendInt5 | PendInt4 | PendInt3 | PendInt2 | PendInt1 | PendInt0 |

Address: *IntControllerBase* + 0x80
Type: Read/write
Reset value: 0

***Description***

This register contains one bit per interrupt level. A read of this register examines the state of the interrupt controller; a write can explicitly trigger an interrupt.

A bit is set when the triggering condition for an interrupt level is met. All bits are independent so that several bits can be set in the same cycle. Once a bit is set, a further triggering condition will have no effect. The triggering condition is independent of INC_Mask.

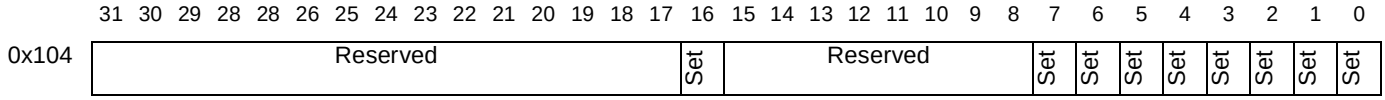
The highest priority interrupt bit is reset, once the interrupt controller has made an interrupt request to the CPU.

The interrupt controller receives interrupt requests and makes an interrupt request to the CPU when it has a pending interrupt request of higher priority than the currently executing interrupt handler.

If the software needs to write or clear some bits of the INC_Pending register, the interrupts should be masked (by writing or clearing the INC_Mask register) before writing or clearing the INC_Pending register. The interrupts can then be unmasked.

The INC_Pending register is mapped onto registers INC_Set_Pending and INC_Clear_Pending so that bits can be set or cleared individually.

## INC_SET_EXEC Set a bit of the Exec register



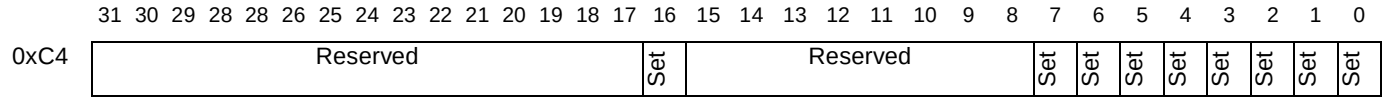

Address: *IntControllerBase* + 0x104
Type: Write only

***Description***

This register sets bits of the INC_EXEC register individually. Writing a ‘1’ in this register sets the corresponding bit in the INC_EXEC register, a ‘0’ leaves the bit unchanged.

Note: do not write to this register as device operation is undefined.

## INC_SET_MASK Set an interrupt enable mask

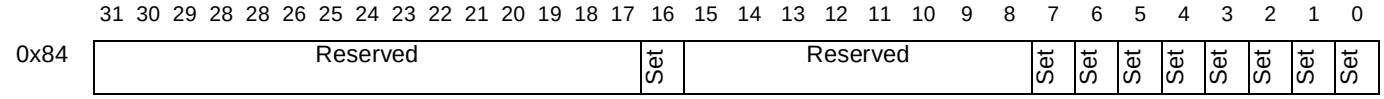



Address: *IntControllerBase* + 0xC4
Type: Write only

***Description***

This register sets bits of the INC_Mask register individually. Writing a ‘1’ in this register sets the corresponding bit in the INC_Mask register, a ‘0’ leaves the bit unchanged.

## INC_SET_PENDING Set a bit of the Pending register




Address: *IntControllerBase* + 0x84
Type: Write only

***Description***

This register sets bits of the INC_Pending register individually. Writing a ‘1’ in this register sets the corresponding bit in the INC_Pending register, a ‘0’ leaves the bit unchanged.

## INC_TRIGGERMODE Interrupt trigger mode

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 | 2 1 0 |
|---|---|---|
| 0x40 | Reserved | Trigger0 |
| 0x44 | Reserved | Trigger1 |
| 0x48 | Reserved | Trigger2 |
| 0x4C | Reserved | Trigger3 |
| 0x50 | Reserved | Trigger4 |
| 0x54 | Reserved | Trigger5 |
| 0x58 | Reserved | Trigger6 |
| 0x5C | Reserved | Trigger7 |

Address: *IntControllerBase* + 0x40 (INC_TriggerMode0) to 0x5C (INC_TriggerMode7)
Type: Read/write
Reset value: Undefined

***Description***

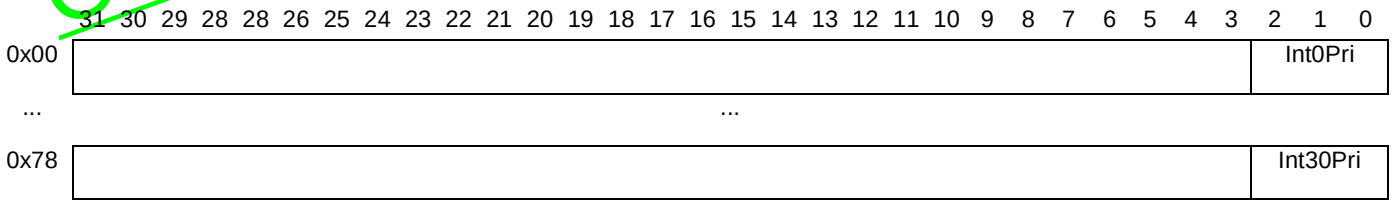
These registers control the triggering conditions of the interrupts. Each interrupt channel can be programmed to trigger on rising or falling edges or high or low levels on the incoming interrupt signal.

| Bitfield | Description |
|---|---|
| Trigger[7:0] | 000: No trigger mode<br>001: High level - triggered while input high<br>010: Low level - triggered while input low<br>011: Rising edge - low to high transition<br>100: Falling edge - high to low transition<br>101: Any edge - triggered on rising and falling edges<br>110: No trigger mode<br>111: No trigger mode |

Level triggering is different from edge triggering in that if the input is held at the triggering level, a continuous stream of interrupts is generated.

# 12 Interrupt-level control registers (INC)

## INC_INT*n*PRIORITY Interrupt priority



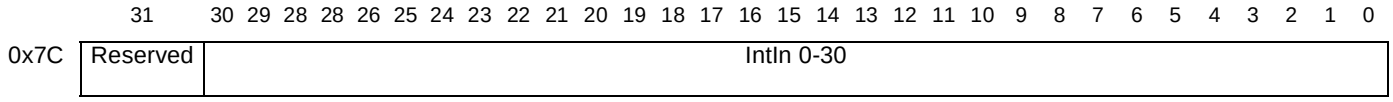

Address: *InterruptLevelBase* + 0x00 (INC_Int0Priority), 0x04 (INC_Int1Priority) up to 0x78 (INC_Int30Priority)
Type: Read/write
Reset value: 000

***Description***

The value in this register sets the priority of internal interrupts, from 0 - 7. Each register is word aligned. There is one register for each interrupt source from N=0 to N=30.

| Bitfield | Description |
|---|---|
| IntnPri[2:0] | 000: 0 (lowest priority)<br>001: 1<br>010: 2<br>011: 3<br>100: 4<br>101: 5<br>110: 6<br>111: 7 (highest priority) |

## INC_INPUTINTERRUPT Interrupt status




Address: *InterruptLevelBase* + 0x7C
Type: Read only
Reset value: 0

***Description***

This register identifies the source of an interrupt, coming into the interrupt level controller. Each register bit corresponds to an individual interrupt source (bits 0-30 to interrupt sources 0-30). The STi5518 Data Sheet gives a table of interrupt assignments, this can be used to identify the cause of interrupt. When the bit value is 1, the corresponding interrupt source is high; when the bit value is 0, the corresponding interrupt source is low. Note this register is not latched, but is simply a synchronized version of the interrupt source signal, reflecting its current logic state.

## INC_SRC_TRIGGERMODE Interrupt-source trigger mode

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0xBC | Trigger |
| 0xC0 | Trigger |
| 0xC4 | Trigger |
| 0x... | Trigger |
| 0x134 | Trigger |

Address: *InterruptLevelBase* + 0xBC (Interrupt N=0), 0xC0 (Interrupt N=1) to 0x134 (Interrupt N=30)
Type: Read/write
Reset value: 001 (high level)

***Description***

This register sets the interrupt trigger for each of the 31 Interrupt Level Controller interrupt sources. Register INC_TriggerMode sets the interrupt trigger for each of the 8 interrupt levels (set by the Interrupt Controller). These two registers can be used together to set the trigger mode on each of the 31 interrupt inputs and the 8 interrupt levels. Backward compatibility with the STi5500, STi5505 and STi5508 can be maintained by not programming this feature so that the trigger mode is high level and can then be set on the 8 interrupt levels only. This is the default after reset.

The table of interrupt assignments is given in the Data Sheet. Each interrupt source can be programmed to trigger on rising or falling edges, or high or low levels on the incoming interrupt signal, as described in the table below.

| Bitfield | Description |
|---|---|
| Trigger | 000: No trigger mode<br>001: High level - triggered while input high<br>010: Low level - triggered while input low<br>011: Rising edge - low to high transition<br>100: Falling edge - high to low transition<br>101: Any edge - triggered on rising and falling edges<br>110: No trigger mode<br>111: No trigger mode |

In level-triggering, when the input is held at the triggering level a continuous stream of interrupts is generated.

## INC_SRC_CLEAR_MASK Clear interrupt-source enable mask

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1B4 | | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr | Clr |

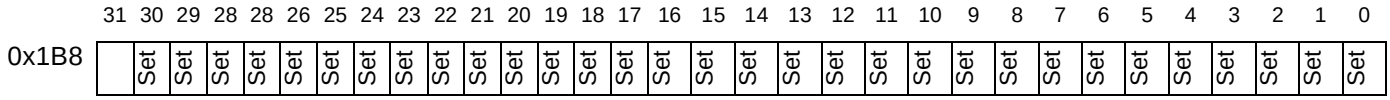
Address: *InterruptLevelBase* + 0x1B4
Type: Write only
Reset value: Undefined

***Description***

When set, these bits clear the corresponding interrupt enable bits in register INC_SRC_MASK. Bits 0-30 of this register can be set individually and correspond directly to bits 0-30 of INC_SRC_MASK (where bit0 corresponds to interrupt source N=0 etc.).

Register INC_Clear_Mask, in the interrupt controller, performs this same function on each of the 8 interrupt levels. These registers can be used in conjunction, or backwards compatibility with the STi5500, STi5505 and STi5508 can be maintained by not using this register.

## INC_SRC_SET_MASK Set the interrupt-source enable mask



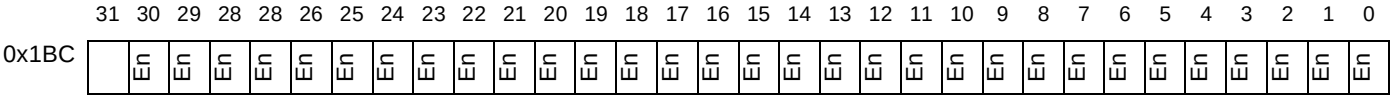

Address: *InterruptLevelBase* + 0x1B8
Type: Write only
Reset value: Undefined

***Description***

When set, these bits set the corresponding bits in register INC_SRC_MASK. Bits 0-30 of this register can be set individually and correspond directly to bits 0-30 of INC_SRC_MASK (where bit0 corresponds to interrupt source N=0 etc.).

Register INC_Set_Mask, in the interrupt controller, performs this same function for each of the 8 interrupt levels. These registers can be used in conjunction, or backwards compatibility with the STi5500, STi5505 and STi5508 can be maintained by not using this register.

## INC_SRC_MASK Interrupt source enable mask




Address: *InterruptLevelBase* + 0x1BC
Type: Read/write
Reset value: 0x7FFFFFFF

***Description***

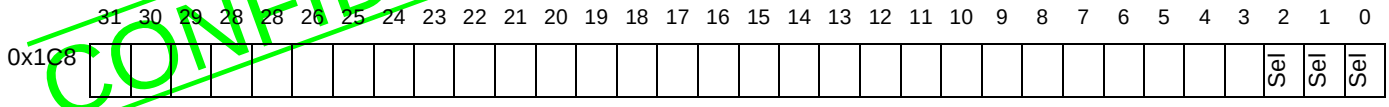
This register selectively enables or disables individual interrupt sources. There is one bit for each interrupt source, where bit 0 corresponds to interrupt source N=0, etc. Writing a'1' to a bit in this register, enables the corresponding interrupt source.

The INC_SRC_STATUS register contains a pending flag for each interrupt source. The INC_SRC_MASK register masks the INC_SRC_STATUS register to control which interrupt sources reach the CPU. (via the interrupt controller)

INC_SRC_MASK is mapped onto registers INC_SRC_SET_MASK and INC_SRC_CLEAR_MASK.

Register INC_Mask, in the interrupt controller, performs the same function as INC_SRC_MASK, but on the 8 interrupt levels. The two registers can be used together, or backwards compatibility with the STi5500, STi5505 and STi5508 can be maintained by not using this register.

## INC_SELNOTINV External Wake-Up interrupt pin is set to active high or active low

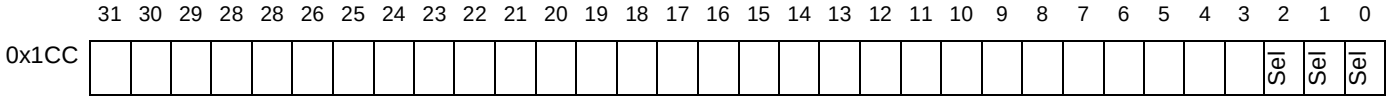



Address: *InterruptLevelBase* + 0x1C8
Type: Read/write
Reset value: 111

***Description***

The three bits of this register correspond to the external interrupt pins IRQ0 to IRQ2 (pins 127 to 125 resp.). When a bit is set to 1, then the corresponding pin is active high, otherwise it is active low. This register is used in conjunction with INC_EN_INT. This register is used to select the active level for the external interrupt pins, to wake up the CPU from low power mode. It does not affect the normal power operating mode interrupt triggering, in the interrupt level controller.

## INC_EN_INT Enable external interrupt to wake-up the low-power controller

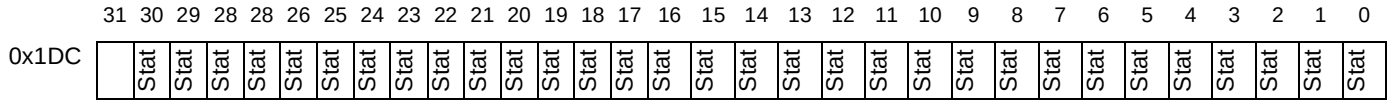



Address: *InterruptLevelBase* + 0x1CC
Type: Read/write
Reset value: 0

***Description***

The three bits of this register correspond to the external interrupt pins IRQ0 to IRQ2 (pins 127 to 125 resp.). When a bit is set, then the corresponding pin can be used to wake-up the CPU from low power mode, when entered by the low power controller.

## INC_SRC_STATUS Triggered status of an interrupt source



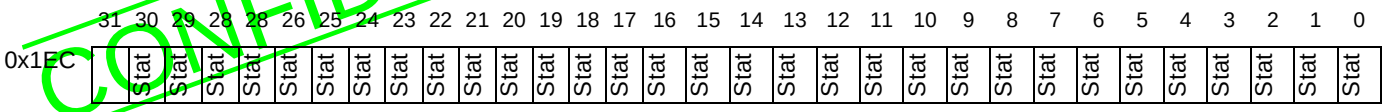

Address: *InterruptLevelBase* + 0x1DC
Type: Read only
Reset value: 0

***Description***

This register identifies the source of an interrupt, as it is after processing by the interrupt level controller trigger detection logic. Each bit corresponds to an individual interrupt source N (from 0 to 30). A bit is set when the triggering condition for that interrupt source is met. If the corresponding source bit is set to have an edge sensitive trigger mode, then the bit is only cleared explicitly by writing to the corresponding bit in the INC_SRC_CLEAR register. If the bit is set to be active high level, then the corresponding bit in the register remains set to 1,until the interrupt source is taken low again, when it is reset to 0. If the bit is set to be active low level, then the corresponding bit in the register remains set to 1 until the interrupt source is taken high again, when it is reset to 0.

Note the status register is set when the trigger mode condition is met, even if the corresponding bit of the INC_SRC_MASK register is reset. In this case no interrupt level output is generated for the interrupt controller.

## INC_SRC_CLEAR Clears the interrupt-source status register




Address: *InterruptLevelBase* + 0x1EC
Type: Write only
Reset value: Undefined

***Description***

When set, individual bits of this register clear the corresponding bits of the INC_SRC_STATUS register, where bit 0 of this register corresponds to bit 0 of INC_SRC_STATUS etc.

# 13 IR transmitter/receiver (IRB)

This section describes the RC transmitter and receiver registers, the RC and UHF receiver and control registers and the noise suppression registers of the IR transmitter/receiver. Although the IR RC receiver and UHF RC receiver registers are held at different addresses, their register descriptions are identical and are only given once for each pair of registers, in *RC receiver registers* on page 172.

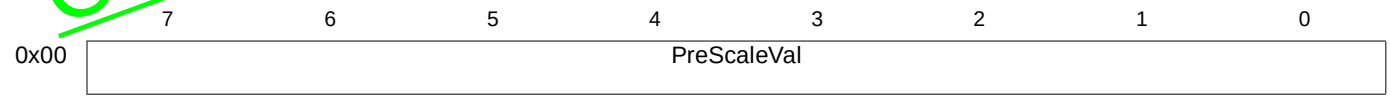
| Register | Address | Bits | Access | Description | |
|---|---|---|---|---|---|
| IRB_TX_PRE_SCALER_IR | 0x00 | 8 | W/R | Clock pre-scaler selection | RC Transmit Interface registers |
| IRB_TX_SUB_CARRIER_IR | 0x04 | 16 | W/R | sub-carrier freq. programming | |
| IRB_TX_SYM_PERIOD_IR[1] | 0x08 | 16 | W | symbol period programming | |
| IRB_TX_ON_TIME_IR[1] | 0x0C | 16 | W | symbol on time programming | |
| IRB_TX_INT_EN_IR | 0x10 | 3 | W/R | Transmit Interrupt enable register | |
| IRB_TX_INT_STATUS_IR | 0x14 | 4 | R | Transmit Interrupt status register | |
| IRB_TX_EN_IR | 0x18 | 1 | W/R | RC transmit enable register | |
| IRB_TX_CLR_UNDERRUN_IR | 0x1C | 1 | W | Clears the underrun status | |
| IRB_RX_ON_TIME_IR[1] | 0x40 | 16 | R | Received pulse time capture | RC Receive Interface registers for Infrared signals |
| IRB_RX_SYM_PERIOD_IR[1] | 0x44 | 16 | R | Received symbol period capture | |
| IRB_RX_INT_EN_IR | 0x48 | 4 | W/R | Receive Interrupt enable register | |
| IRB_RX_INT_STATUS_IR | 0x4C | 5 | R | Receive Interrupt status register | |
| IRB_RX_EN_IR | 0x50 | 1 | W/R | RC receive enable register | |
| IRB_RX_MAX_SYM_PERIOD_IR | 0x54 | 16 | W/R | Maximum RC symbol period register | |
| IRB_RX_CLR_OVERRUN_IR | 0x58 | 1 | W | Clears the overrun status | |
| IRB_RX_NOISE_SUPPRESS_WIDTH_IR | 0x5C | 8 | W/R | Reserved | |
| IRB_RC_IRDA_CONTROL | 0x60 | 1 | W/R | Reserved | |
| IRB_RX_SAMPLING_RATE_COMMON | 0x64 | 4 | W/R | Sampling frequency division for UHF and IR frequencies. | Common to RC & UHF receivers |
| IRB_RX_ON_TIME_UHF[1] | 0x80 | 16 | R | Received pulse time capture | RC Receive Interface registers for UHF signals |
| IRB_RX_SYM_PERIOD_UHF[1] | 0x84 | 16 | R | Received symbol period capture | |
| IRB_RX_INT_EN_UHF | 0x88 | 4 | W/R | Receive Interrupt enable register | |
| IRB_RX_INT_STATUS_UHF | 0x8C | 5 | R | Receive Interrupt status register | |
| IRB_RX_EN_UHF | 0x90 | 1 | W/R | RC receive enable register | |
| IRB_RX_MAX_SYM_PERIOD_UHF | 0x94 | 16 | W/R | Maximum RC symbol period register | |
| IRB_RX_CLR_OVERRUN_UHF | 0x98 | 1 | W | Clears the overrun status | |
| IRB_RX_NOISE_SUPPRESS_WIDTH_UHF | 0x9C | 8 | W/R | Noise suppression width | |

**Table 47 Infrared transmitter/receiver (IRB)**

1. These locations have a quadruple buffer (i.e. a FIFO of length 4 words).

## 13.1 RC transmitter registers

### IRB_TX_PRE_SCALER_IR Clock prescaler

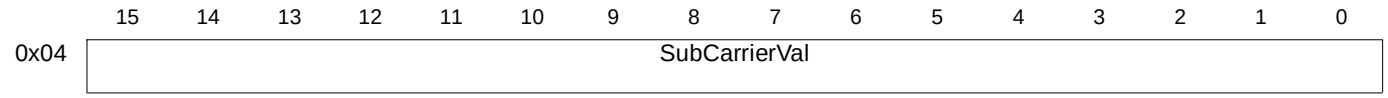



Address: *BaseAddress* + 0x00
Type: Read/write
Reset value: 0x00

***Description***

This register selects the value of the pre-scaler for clock division. The pre-scaled clock frequency is obtained by dividing the system clock frequency by PreScaleVal. It determines the transmit sub-carrier resolution, see IRB_TX_SUB_CARRIER_IR below.

### IRB_TX_SUB_CARRIER_IR Sub-carrier frequency programming

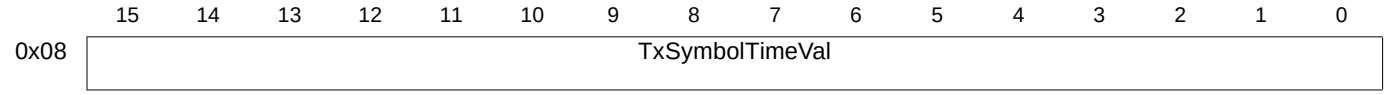



Address: *BaseAddress* + 0x04
Type: Read/write
Reset value: 0x00

***Description***

This register determines the RC transmit sub-carrier frequency. The pre-scaled clock frequency divided by (SubCarrierVal x 2) gives the sub-carrier frequency, which has a 50% duty cycle.

### IRB_TX_SYM_PERIOD_IR symbol-time programming




Address: *BaseAddress* + 0x08
Type: Write
Reset value: 0x0000

***Description***

The value in this register gives the symbol-time (symbol period) in periods of the sub-carrier clock. It must be programmed sequentially with the register *IRB_TX_ON_TIME_IR* on page 170. This register is quadruple buffered.

## IRB_TX_ON_TIME_IR Symbol ON time programming

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | TxOnTimeVal | | | | | | | | | | | | | | | |

Address: *BaseAddress* + 0x0C
Type: Write
Reset value: 0x0000

***Description***

The value in this register gives the symbol ON time (pulse duration) in periods of the sub-carrier clock. This register is quadruple buffered.

*Note* *The registers* IRB_TX_SYM_PERIOD_IR *and* IRB_TX_ON_TIME_IR *act as a single register set. They must be programmed sequentially as a pair to latch in the data.*

## IRB_TX_INT_EN_IR Transmit interrupt enable register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | Reserved | | TxFifoIrq | | Reserved | | | TxIrqEnable |

Address: *BaseAddress* + 0x10
Type: Read/write
Reset value: 0x00

***Description***

| Bit field | Function |
|---|---|
| TxIrqEnable | Select the Transmit interrupt enable/ disable:<br>0 - Interrupt Disable, 1 - Interrupt Enable |
| TxFifoIrq[1:0] | Select the Transmit FIFO fullness interrupt:<br>00 - Invalid<br>01 - Single word empty<br>10 - Half empty (two words empty)<br>11 - Buffer empty (three words empty) |
| Reserved | Set to logic ‘0’ |

## IRB_TX_INT_STATUS_IR Transmit Interrupt status register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x14 | Reserved | | TxFifoStatus | | Reserved | | TxUnderRun Status | TxIrqStatus |

Address: *BaseAddress* + 0x14
Type: Read
Reset value: 0x00

***Description***

This register is also updated when data is written into the registers IRB_TX_SYM_PERIOD_IR and IRB_TX_IR_ON_TIME_IR.

| Bit Field | Function |
|---|---|
| TxIrqStatus | Transmit interrupt status:<br>1 - Interrupt enabled |
| TxUnderRunStatus | Transmit underrun status:<br>0 - No under run, 1 - Under run occurred |
| TxFifoStatus[1:0] | Transmit FIFO fullness status:<br>00 - FIFO Full, not empty<br>01 - One block empty in FIFO<br>10 - Two blocks empty in FIFO<br>11 - Three blocks empty in FIFO |
| Reserved | Set to logic ‘0’ |

## IRB_TX_EN_IR RC transmit enable register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x18 | Reserved | | | | | | | TxEnable |

Address: *BaseAddress* + 0x18
Type: Read/write
Reset value: 0x00

***Description***

This register enables the RC transmit processor. When it is set to ‘1’ and there is data in the transmit FIFO, then the RC processor is transmitting.

## IRB_TX_CLR_UNDERRUN_IR Clears the underrun status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1C | Reserved | | | | | | | ClrUnderRun |

Address: *BaseAddress* + 0x1C
Type: Write
Reset value: 0x00

***Description***

This register must be set to “1” as part of the procedure for clearing the underrun flag in the register IRB_TX_INT_STATUS_IR. No data is transmitted until this flag has been cleared.

## 13.2 RC receiver registers

If not explicitly stated the following registers are common to both the RC-IR receiver and the RC-UHF receiver. The first address given is the RC-IR receiver (IR). The registers are distinguished by the suffix "_IR" for the IR receiver and "_UHF" for the UHF receiver.

### IRB_RX_ON_TIME Received pulse time capture

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x40<br>0x80 | RxOnTimeVal | | | | | | | | | | | | | | | |

Address: *BaseAddress* + 0x40 (IR) *BaseAddress* + 0x80 (UHF)
Type: Read
Reset value: 0x0000

***Description***

The value in this register is the detected duration (in microseconds) of the received RC pulse. It must be read sequentially with register *IRB_RX_SYM_PERIOD* on page 172. The register is quadruple buffered.

### IRB_RX_SYM_PERIOD Received symbol period capture

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x44<br>0x84 | RxSymbolTimeVal | | | | | | | | | | | | | | | |

Address: *BaseAddress* + 0x44 (IR) *BaseAddress* + 0x84 (UHF)
Type: Read
Reset value: 0x0000

***Description***

This register holds the detected time (in micro seconds) between the start of two successive received RC pulses. It is quadruple buffered.

NOTE: The registers IRB_RX_SYM_PERIOD and IRB_RX_IR_ON_TIME act as a register set. A new value can only be read after reading both registers sequentially.

### IRB_RX_INT_EN Receive Interrupt enable register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x48<br>0x88 | Reserved | | RxFifoIrq[1:0] | | Reserved | | LastSymbol-IrqEnable | RxIrqEnable |

Address: *BaseAddress* + 0x48 (IR) *BaseAddress* + 0x88 (UHF)
Type: Read/write
Reset value: 0x00

***Description***

To inhibit all these interrupts the RxIrqEnable bit (register bit 0) must be set to ‘0’.

| Bit Field | Function |
|---|---|
| RxIrqEnable | Select the receive interrupt enable/disable:<br>0 - Interrupt disable, 1 - Interrupt enable |
| LastSymbolIrqEnable | Select Interrupt Enable/disable on last symbol:<br>1- Generate interupt on last symbol received |
| RxFifoIrq[1:0] | Select the Receive FIFO fullness interrupt:<br>00 - Invalid<br>01 - One word available for read<br>10 - Two words available for read (half full)<br>11 - Three words available for read (FIFO full) |
| Reserved | set to logic “0” |

## IRB_RX_INT_STATUS Receive Interrupt status register

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4C<br>0x8C | Reserved | | RxFifoStatus | | Reserved | RxOverRun-Status | LastSymbol-IrqStatus | RxIrqStatus |

Address: *BaseAddress* + 0x4C (IR) *BaseAddress* + 0x8C (UHF)
Type: Read
Reset value: 0x00

***Description***

| Bit Field | Function |
|---|---|
| RxIrqStatus | Receive interrupt status:<br>1 - Interrupt Active |
| LastSymbolIrqStatus | Last symbol interrupt status:<br>1 - Interrupt Active |
| RxOverRunStatus | Receive overrun status:<br>0 - No overrun, 1 - Overrun occurred |
| RxFifoStatus[1:0] | Receive FIFO fullness status:<br>00 - FIFO empty<br>01 - One word in FIFO<br>10 - Two words in FIFO<br>11 - Three words in FIFO |
| Reserved | Set to logic “0” |

## IRB_RX_EN RC receive enable register

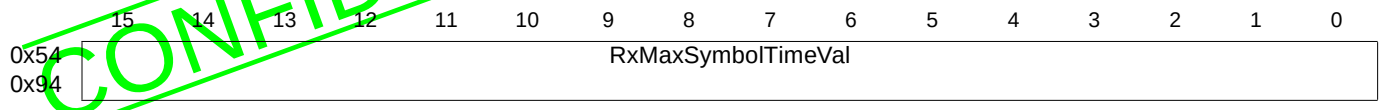
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x50<br>0x90 | Reserved | | | | | | | RxEnable |

Address: *BaseAddress* + 0x50 (IR) *BaseAddress* + 0x90 (UHF)
Type: Read/write
Reset value: 0x00

***Description***

When this register is set to “1” the RC receive section is enabled to read incoming data.

### IRB_RX_MAX_SYM_PERIOD Maximum RC symbol period register

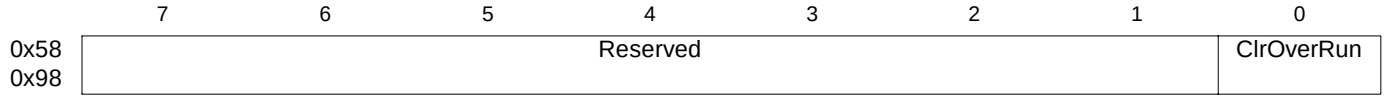



Address: *BaseAddress* + 0x54 (IR) *BaseAddress* + 0x94 (UHF)
Type: Read/write
Reset value: 0x0000

***Description***

The value in this register sets the maximum symbol period (in microseconds) which is necessary to define the time-out for recognizing the end of the symbol stream.

### IRB_RX_CLR_OVERRUN Clears the overrun status



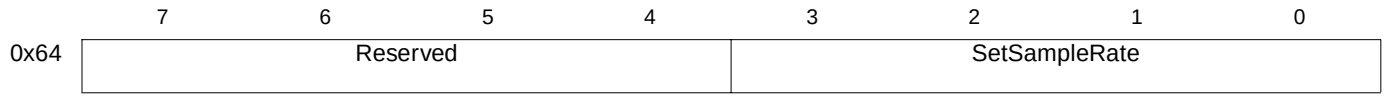

Address: *BaseAddress* + 0x58 (IR) *BaseAddress* + 0x98 (UHF)
Type: Write
Reset value: 0x00

***Description***

This register must be set to “1” as part of the procedure for clearing the overrun flag in the register IRB_RX_INT_STATUS. No new data is written into the receive FIFO while this flag is set.

## 13.3 RC and UHF receiver control

### IRB_RX_SAMPLING_RATE_COMMON Sampling frequency division for UHF and IR




Address: *BaseAddress* + 0x64
Type: Read/write
Reset value: 0x00

***Description***

The value in this register is the divisor which sets the sampling rate for the RC receive sections to 10MHz. It must be set to “5” with an IR transmitter/receiver system clock of 50MHz or to “6” with a clock of 60MHz. The register is common to both the IR and the UHF receive processors.

## 13.4 Noise suppression register

### IRB_RX_NOISE_SUPPRESS_WIDTH_UHF Noise suppression width

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x9C | NoiseSuppressWidth | | | | | | | |

Address: *BaseAddress* + 0x9C (UHF)
Type: Read/write
Reset value: 0x00

***Description***

The value, in microseconds, in this register determines the maximum width of noise pulses which the filter suppresses.

# 14 Link (LNK)

## 14.1 Introduction

The link interface address space is 4kbytes (1024*32 bit words), and all resources are accessed as 32-bit words. In this chapter, all addresses point to bytes. The chapter is split into two sections, one describes the FRAM registers, and the other the link registers. The registers are in order of ascending address.

The following table shows the arrangement of registers in the 4kbytes memory map.

| Address | Size | Resource |
|---|---|---|
| *LinkBaseAddress* - *LinkBaseAddress* + 0x77C | 480 Words | FRAM |
| *LinkBaseAddress*+0x780 - *LinkBaseAddress*+0xEFC | 480 Words | Not Used |
| *LinkBaseAddress*+0xF00 - *LinkBaseAddress*+0xFC4 | 49 Words | Registers |
| *LinkBaseAddress*+0xFC8 - *LinkBaseAddress*+0xFFC | 15 Words | Not used |

**Table 48 Global address map**

## 14.2 FRAM registers

The following table describes the arrangement of the FRAM (filter RAM) registers.

| Address | Size | Function |
|---|---|---|
| *LinkBaseAddress* - *LinkBaseAddress* + 0x4FC | 320 Words | Filter Data |
| *LinkBaseAddress* + 0x500 - *LinkBaseAddress* + 0x57C | 32 Words | Descrambling Keys (reserved) |
| *LinkBaseAddress* + 0x580 - *LinkBaseAddress* + 0x5FC | 32 Words | Stream Configuration #1 |
| *LinkBaseAddress* + 0x600 - *LinkBaseAddress* + 0x67C | 32 Words | Stream Configuration #2 |
| *LinkBaseAddress* + 0x680 - *LinkBaseAddress* + 0x6FC | 32 Words | Stream Configuration #3 |
| *LinkBaseAddress* + 0x700 - *LinkBaseAddress* + 0x77C | 32 Words | IRQ Registers |

**Table 49 Link interface FRAM address map**

Eight different keys set are defined. Each key set contains 2 keys of 64 bit each, mapped as follows:

| Address | Key set | Key Parity | Key bits | Type |
|---|---|---|---|---|
| 0x20002500 | 0 | EVEN | 31:0 (LSW) | R/W |
| 0x20002504 | 0 | EVEN | 63:32 (MSW) | R/W |
| 0x20002508 | 0 | ODD | 31:0 (LSW) | R/W |
| 0x2000250C | 0 | ODD | 63:32 (MSW) | R/W |
| 0x20002510 | 1 | EVEN | 31:0 (LSW) | R/W |
| 0x20002514 | 1 | EVEN | 63:32 (MSW) | R/W |
| 0x20002518 | 1 | ODD | 31:0 (LSW) | R/W |
| 0x2000251C | 1 | ODD | 63:32 (MSW) | R/W |
| ... | ... | ... | ... | ... |
| 0x20002578 | 7 | ODD | 31:0 (LSW) | R/W |
| 0x2000257C | 7 | ODD | 63:32 (MSW) | R/W |

**Table 50 Descrambling keys memory map**

## LNK_STREAM_CONF_1 Stream configuration 1

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOF_IRQ | EOS_IRQ | AF_IRQ | OUTPUT_PACKET | STREAM_TO_BUFFER | ERROR_PATT | PES_DES_EN | PES_NSEC | SEC_F_INV | SEC_F_INV_REG | | | FILTER_LENGTH | | | | FILTER_NB | | | | | | FRAM_ADDRESS | | | | | | | | | |

Address: *LinkBaseAddress* + 0x580, 584, 588, 58B...5F0 (32 words of 4 bytes)
Type: Read/write
Reset value: Not defined

***Description***

| Bitfield | Description |
|---|---|
| FRAM_ADDRESS | Filter Start Address in FRAM |
| FILTER_NB | Number of Filters |
| FILTER_LENGTH | Size of Filters |
| SEC_F_INV_REG | Offset in filter for 'not equal' filter |
| SEC_F_INV | 1: Equal Mode, 0: Not Equal Mode |
| PES_NSEC | 1: Stream PES, 0: Stream SECTION |
| PES_DES_EN | 1: Enables PES Level Descrambling, 0: No PES Level Descrambling |
| ERROR_PATT | 1: Insert Error Pattern on CC Error, 0: No Insertion of Error Code |
| STREAM_TO_BUFFER | 1: Enable Transfer to DMA Buffer, 0: Bytes are Only Read for SDAV |
| OUTPUT_PACKET | 1: Send Packet to SDAV, 0: Ignored by SDAV |
| AF_IRQ | DVB: IRQ Mask on LNK_AFIRQ (not used on STI5508) |
| EOS_IRQ | DVB: IRQ Mask on end of Section<br>DSS: End of Packet<br>DVD: End of Sector |
| EOF_IRQ | DVB: IRQ Mask on end of Section Filtering<br>DSS: End of Conditional Filtering<br>DVD: Config. has been loaded |

## LNK_STREAM_CONF_2 Stream configuration 2

| 31 | 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 | 10 9 8 7 | 6 5 4 3 2 1 0 |
|---|---|---|---|
| INCREMENT | BUFFER_SIZE | STOP_ADDR | START_ADDR | DMA_BANK_ADDR | DMA_HIGH_ADDR |

Address: *LinkBaseAddress* + 0x600, 604, 608, 60B...670 (32 words of 4 bytes)
Type: Read/write
Reset value: Not defined

***Description***

| Bitfield | Description |
|---|---|
| DMA_HIGH_ADDR | DMA address(21:15) |
| DMA_BANK_ADDR | DMA address(31:28) |
| START_ADDR | DMA start address(14:8) |
| STOP_ADDR | DMA stop address(14:5) |
| BUFFER_SIZE | DMA circular buffer size: |
| INCREMENT | 1: The start address is set by register bit LNK_STREAM_CONF_3.DMA_LOW_ADDR, and is incremented after each access, independent of the start and stop addresses.<br>0: All of the data is written to the address specified by register bits LNK_STREAM_CONF_2.DMA_HIGH_ADDR and LNK_STREAM_CONF_3.DMA_LOW_ADDR. |

| BUFFER_SIZE(2:0) | Buffer Size kBytes | Stop Precision Bytes |
|---|---|---|
| 000 | 256 | 1 |
| 001 | 512 | 1 |
| 010 | 1 | 1 |
| 011 | 1.5 | 2 |
| 100 | 2 | 4 |
| 101 | 3 | 8 |
| 110 | 4.5 | 16 |
| 111 | 8 | 32 |

### LNK_STREAM_CONF_3 Stream configuration 3

| 31 30 29 28 | 27 26 25 24 23 22 21 20 19 18 17 16 | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|---|
| CC_COUNTER_IN / SUSPEND | DMA_LOW_ADDR | TRANSFER_LENGTH |

Address: *LinkBaseAddress* + 0x0x680, 684, 688, 68B...6F0 (32 words of 4 bytes)
Type: Read/write
Reset value: Not defined

***Description***

| Bitfield | Description |
|---|---|
| TRANSFER_LENGTH | DVB - PES (When PES Scrambling): remaining Bytes of the PES Header, if the Header Exceeds 184 Bytes<br>DVB - SEC: Remaining Bytes in Current Section<br>DSS: Bit 0 = HD[1] (Toggle Bit) |
| DMA_LOW_ADDR | DMA address(14:0) |
| SUSPEND | DVB - PES: 1=Bypass First CC Check, 0=CC Check Takes Place<br>DVB - SEC: 1=Section Cut, 0=No Section Cut<br>DSS: 1=Hunt Mode, 0=No Hunt Mode |
| CC_COUNTER_IN | Current Continuity Counter Value |

## 14.3 Link registers

### LNK_STREAM_EN*n* Stream enable

| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 | 0 |
|---|---|
| Not Used | SE |

Address: *LinkBaseAddress* + 0xF00 + 4*n*
Type: Read/write
Reset value: 0

***Description***

The stream enable registers enable and disable the streams. Each of the 32 streams has one register, and stream *n* is enabled by LNK_STREAM_ENn which is at the address given by: *TransportDemuxBase* + 0xF00 + 4*n*

| Bitfield | Description | |
|---|---|---|
| SE | STREAM_EN | 1: Enables the stream *n,* 0: Disables the stream *n* |

## LNK_STAT Status

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | FRA | | | | | X | TE | FE | FWA | | | | | Not Used | | | | | | FR | FO | FNE |

Address: *LinkBaseAddress* + 0xF80
Type: Read only except bit 2, which is read/write
Reset value: Undefined except bit 2, which is reset to 0

***Description***

| Bitfield | Description | | Type | Reset |
|---|---|---|---|---|
| FNE | FIFO_NOT_EMPTY | 1: FIFO is not empty | R | x |
| FO | FIFO_OVERFLOW | 1: FIFO has overflowed | R | x |
| FR | FIFO_RESET | 1: FIFO pointers are reset | R/W | 0 |
| FWA | FIFO_WR_ADDR | Actual write pointer position of IRQ FIFO | R | x |
| FE | F_Error | 1: Error specified by the FEC interface | R | x |
| TE | TRANSPORT_ERROR | 1: Error specified in the TP header | R | x |
| X | Not used | | | |
| FRA | FIFO_RD_ADD | Actual read pointer position of IRQ FIFO | R | x |

## LNK_STAT_FIFO FIFO status

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOF | EOS | IF | AF | AO | DO | BS | SU | SO | PCR | x | SN | | | | | CC | | | | PT | | | OM | TM | | | | | | | |

Address: *LinkBaseAddress* + 0xF84
Type: Read only
Reset value: Undefined

***Description***

The register is the FIFO status word. For DVB, if bit AF=1 then LNK_STAT_FIFO(15:0) is set to the first 2 bytes of the AF. For bit EOS, the delay between the Interrupt generation and the actual transfer of the last byte of the section to memory can be up to 4 µs.

| Bitfield | Description | |
|---|---|---|
| TM | TARGET_MATCH | For DVD:<br>TM[7]: is '1' if the potential SC is at end of the sector<br>TM[6:5]:contains the two MSBs of the FRAM address for this sector<br>TM[4:0]:contains the number of SCs in a sector<br><br>Shows the 8 lowest target matches |
| OM | OTHER_MATCH | At least 1 match in the top 24 targets |
| PT | Packet type | For DVD:<br>000: Unknown packet type<br>001: if video packet is found<br>010: if navigation packet is found<br>011: if subpicture packet is found<br>100: Pack_start_code was not found!<br>101: if MPEG audio packet is found<br>110: if AC3 audio packet is found<br>111: if LPCM audio packet<br><br>For DVB:<br>Bit 11=PAYLOAD_UNIT_START_INDICATOR<br>Bit 10=SCRAMBLING_INDICATOR<br>Bit 9=KEY_INDICATOR (0: even; 1: odd)<br><br>For DSS: Bit 11=HD(3), Bit 10=HD(2), Bit 9=KEY_INDICATOR (0: odd; 1: even) |
| CC | Continuity counter | Continuity counter |
| SN | STREAM_NUMBER | Stream number of packet |
| X | Not Used | |
| PCR | PCR | 1: PCR has been latched |
| SO | SDAV_OVERFLOW | 1: SDAV overflow if SDAV_underflow = 0, else interrupt for EXTRA_BITS |
| SU | SDAV_UNDERFLOW | 1: SDAV Underflow if **SDAV_overflow** = 0, else interrupt for EXTRA_BITS |
| BS | BAD_SEC | 1: Bad section received |
| DO | DMA_OVERFLOW | 1: Storage buffer overflow |
| AO | AR_OVERFLOW | 1: Acquisition RAM overflow |
| AF | AF | 1: Adaptation field received |
| IF | INCOMPLETE_FILTER | 1: Filtering not complete at end of pack |
| EOS | END_OF_SECTION | 1: End of section transfer |
| EOF | END_OF_FILTER | 1: End of section filtering |

## LNK_PACKET_LENGTH Packet length

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | | | | | | | PACKET_LENGTH | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xF88
Type: Read/write
Reset value: 0xBC

***Description***

| Bitfield | Description |
|---|---|
| PACKET_LENGTH | Number of bytes per packet |

## LNK_TIME_OUT Time out

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | | | | | | | | | | | | | TIME_OUT | | | | | |

Address: *LinkBaseAddress* + 0xF8C
Type: Read/write
Reset value: 0x38

***Description***

| Bitfield | Description |
|---|---|
| TIME_OUT | AR threshold to reset the block |

## LNK_MODE Mode

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | E_SCD | Reserved | | | DFB | PM | BO | NC | LS | DVD_M | DSS_M | DVB_M |

Address: *LinkBaseAddress* + 0xF90
Type: Read/write
Reset value: 0x001

***Description***

The table below defines the mode register. DVB_MODE, DSS_MODE and DVD_MODE are mutually exclusive, so one of them must be set to 1.

| Bitfield | Description |
|---|---|
| DVB_M | DVB mode<br>1: DVB Mode |
| DSS_M | DSS mode<br>1: DSS Mode |
| DVD_M | DVD mode<br>1: DVD Mode |
| LS | Link SDAV<br>0: Input from link/channel, 1: Input from SDAV |
| NC | NRSS card<br>0: No NRSS card, 1: NRSS Card is Used |
| BO | BIT order<br>0: LSB first, 1: MSB first |
| PM | PCM mode<br>0: PCM mode is disabled, 1: PCM mode is enabled |
| DFB | Disable final burst<br>0: The last transfer can be a full burst, 1: No full burst for the last transfer |
| E_SCD | Enable start-code detector<br>0: Start-code detector disabled, 1: Start-code detector enabled |

## LNK_PCR_STREAM PCR stream

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | PIM | PSN | | | | |

Address: *LinkBaseAddress* + 0xF94
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| PSN | PCR_STREAM_NB: Stream which contains PCR |
| PIM | PCR_IRQ_MASK: Interrupt mask on PCR_IRQ. An interrupt is generated if this bit is set to 1. |

## LNK_AF1-0 Adaptation field

| | 31 30 29 28 27 26 25 24 | 23 22 21 20 19 18 17 16 | 15 14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 0 |
|---|---|---|---|---|
| 0xF98 | AF_7 | AF_6 | AF_5 | AF_4 |
| 0xF9C | AF_3 | AF_2 | AF_1 | AF_0 |

Address: *LinkBaseAddress* + 0xF98 (LNK_AF0) - 0xF9C (LNK_AF1)
Type: Read only
Reset value: Undefined

***Description***

The table below defines the LNK_AF registers, which hold the first eight bytes of the adaptation field. The interrupt is generated if mask = 1.

| Bitfield | Description | | Type |
|---|---|---|---|
| AF_7 | AF1 | LNK_AFByte #7 | R |
| AF_6 | | LNK_AFByte #6 | R |
| AF_5 | | LNK_AFByte #5 | R |
| AF_4 | | LNK_AFByte #4 | R |
| AF_3 | AF0 | LNK_AFByte #3 | R |
| AF_2 | | LNK_AFByte #2 | R |
| AF_1 | | LNK_AFByte #1 | R |
| AF_0 | | LNK_AFByte #0 | R |

## LNK_V_PTS Video time stamp

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V_PTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xFA0
Type: Read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| V_PTS | Actual video PTS (90 kHz) |

## LNK_A_PTS Audio time stamp

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A_PTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xFA4
Type: Read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| A_PTS | Actual audio PTS (90 kHz) |

## LNK_PCR PCR

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xFA8
Type: Read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| PCR | Actual PCR (90 kHz) |

## LNK_PCR_EXT PCR extension

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | | | | | | | | | | PCR_EXT | | | | | | | | |

Address: *LinkBaseAddress* + 0xFAC
Type: Read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| PCR_EXT | Actual PCR extension (27 MHz) |

## LNK_AR_SIZE AR size

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not used | | | | | | | | | | | | | | | | | | | | | | | | | | AR_SIZE | | | | | |

Address: *LinkBaseAddress* + 0xFB0
Type: Read/write
Reset value: 0x3E

***Description***

| Bitfield | Description |
|---|---|
| AR_SIZE | Programmable AR size. |

## LNK_SDAV_CONF SDAV configuration

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xFB4 | | | | | | SE | SDC | SDH | IS | SDE | THRESHOLD | | | | | | S | D | DAM | EC | EXTRA_BITS | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xFB4
Type: Read/write
Reset value: Undefined except bits 12, 14, 15 and 22 reset to 0.

***Description***

| Bitfield | Description |
|---|---|
| EXTRA_BITS | Inserted into the SDAV packet headers |
| EC | EXT_CLK<br>1: External clock (OSC_IN pin)<br>0: internal clock (SYS_CLK) |
| DAM | DES_AR_MODE Active only when PCM_MODE = 0<br>1: Input from descrambler<br>0: Input from acq. RAM (scrambled) |
| D | DIRECTION<br>1: Output mode, 0: Input mode |
| S | SDAV_1394<br>1: IEEE 1394<br>0: SDAV (OSC_IN: 49.152MHz) |
| THRESHOLD | SDAV buffer threshold to start output transfer |
| SDE | SDAV_ENABLE<br>1: SDAV enabled<br>0: SDAV disabled - x |
| IS | 1394_IS_SLOW<br>1: 0MHz < 1394_CLK < 40MHz<br>0: 40MHz < 1394_CLK < 60MHz |
| SDH | SDAV_HEADER_ENABLE<br>In IEEE 1394 mode enable the header:<br>1: Packet has a STi5518 header<br>0: Packet has no STi5518 header |
| SDC | AD_SDAV_CLK_SWITCH<br>For the SDAV clock switch Mechanism:<br>1: Before setting PCM_CLK_SWITCH to 1<br>0: Before resetting PCM_CLK_SWITCH to 0 |
| SE | STUFFING_ENABLE<br>ONLY in P1394 and DSS, active high<br>1: Packet has 10 bytes of stuffing appended<br>0: Packet has no stuffing |

## LNK_SDAV_DMA_EN SDAV DMA enable

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | | | | | | | | | | | | | IIF | LBP | | FBP | | DE |

Address: *LinkBaseAddress* + 0xFB8
Type: Read only
Reset value: Undefined except bit 0 reset to 0

***Description***

| Bitfield | Description | |
|---|---|---|
| DE | DMA_EN | Enable insertion at next opportunity |
| FBP | FIRST_BYTE_POSIT | Defines which bytes are active in the first word of the DMA<br>00: DMA_DATA[31:0]<br>01: DMA_DATA[31:8]<br>10: DMA_DATA[31:16]<br>11: DMA_DATA[31:24] |
| LBP | LAST_BYTE_POSIT | Defines which bytes are active in the last word of the DMA<br>11: DMA_DATA[31:0]<br>10: DMA_DATA[23:0]<br>01: DMA_DATA[15:0]<br>00: DMA_DATA[7:0] |
| IIF | INSERT_IF_EMPTY | 1: Insertion of CPU packets only if no transfer from the Transport Stream Demultiplexor to the STi5508 |

## LNK_SDAV_DATA SDAV data

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SDAV_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *LinkBaseAddress* + 0xFBC
Type: Read/write
Reset value: Undefined

***Description***

| Bitfield | Comment |
|---|---|
| SDAV_DATA | SDAV data write port |

## LNK_EN_LINK Enable link

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | EL |

Address: *LinkBaseAddress* + 0xFC0
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| EL_ | Enable Transport Stream Demultiplexor |

## LNK_PCM_CONF PCM configuration

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | DIV | | | | | | FORMAT | DIF | INV_IRCLK | INV_SCLK | ORD | PCM_1818 | QUANT | | NB_AUDIO_CH | | | CLK_SWITCH | PLAY | MUTE |

Address: *LinkBaseAddress* + 0xFC4
Type: Read/write
Reset value: Bits zero and one are reset to 0, all other bits have undefined reset

***Description***

| Bitfield | Description |
|---|---|
| PCM_MUTE | Mute function reset=0 |
| PCM_PLAY | Play function reset=0 |
| CLK_SWITCH | Switches between the two SDAV clocks.<br>For the SDAV's clock switch resynchronization mechanism, it's a soft RESET, active high: Activated before the SDAV_OUT_CLK is changed or after it was changed, to freeze the SDAV outputs. (To guarantee any glitches when SDAV_OUT_CLK is switched) In PCM_MODE: (same as PCM_MUTE = 0 and PCM_PLAY = 0. But no DMA_REQ are generated). |
| NB_AUDIO_CH | Number of audio channels in the audio packet for linear PCM:<br>000: 1 Channel (mono: left) (right = left), 001 to 111: ACH0 is left, ACH1 is right. |
| QUANT | Quantification of Linear PCM Audio Packet:<br>00: 16 bits, 01: 20 bits, 10: 24 bits, 11: Invalid |
| PCM_18 | Output coding<br>0:16-bit mode, 1:18-bit mode |
| ORD | Has significance only in 16-bit mode<br>0: MSB first, 1: LSB first |
| INV_SCLK | Polarity of SCLK. PCM_DATA, LRCLK are output<br>0: Just after the falling edge of SCLK, 1: Just after the rising edge of SCLK |
| INV_IRCLK | Polarity of the LRCLK<br>0: Left: LRCLK = 1, Right: LRCLK = 0, 1: Left : LRCLK = 0, Right : LRCLK = 1 |
| DIF | This bit is only used in 18-bit mode:<br>0: The 18-bit PCM _DATA, w/14 sign extension bits, is right aligned,<br>1: According to PCM_FORMAT, the 18-bit PCM_DATA is left aligned. |
| FORMAT | This bit is only used if bits PCM_18 = 1 and DIF=1:<br>0: Standard format, left aligned: '0' are inserted on the right.<br>1: Compatible I2S format: '0' is inserted before the MSB on the left,13 '0' are inserted to the right of the LSB. |
| DIV | Divider register for SLCK generation:<br>f(SCLK) = f(PCM_CLK) / 2*(PCM_DIV + 1) |

## LNK_EXTRA_BITS Extra bits

| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 | 9 8 7 6 5 4 3 2 1 | 0 |
|---|---|---|
| Not used | EBI | EBM |

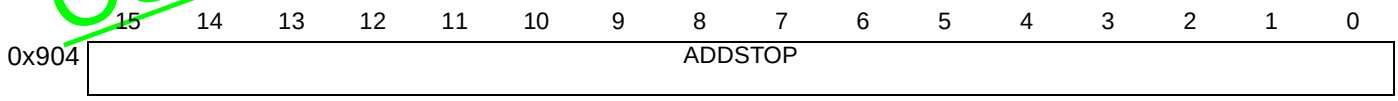
Address: *LinkBaseAddress* + 0xFC8
Type: Read/write
Reset value: Undefined except bit 0 reset to 0.

***Description***

| Bitfield | Description |
|---|---|
| EBM | EXTRA_BITS_IRQ_MASK: Interrupt mask for the extra bits<br>1: Enable interrupt on change of extra bits, 0: Disable interrupt |
| EBI[4:1] | EXTRA_BITS_INPUT: Incoming EXTRA_BYTES: Copy guard information |
| EBI[5:12] | EXTRA_BYTES_INPUT: Incoming EXTRA_BYTES: Playback rate control |

## 14.4 HDD buffer control

### LNK_HDD_ADDSTOP Write limit address

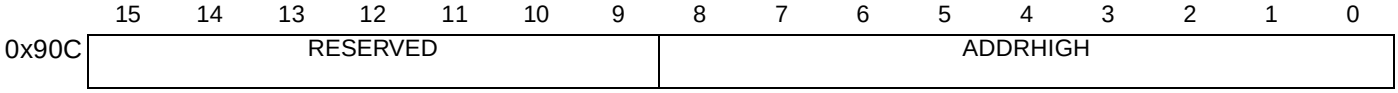



Address: *LinkBaseAddress* + 0x904
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| ADDSTOP | This register causes an HDD link interrupt (interrupt assignment N=29) when the buffer reaches a pre-defined fullness. This parameter must be set sufficiently in advance of reaching the last read address, to avoid overwriting data. |

### LNK_HDD_ADDRHIGH High part of the incoming address

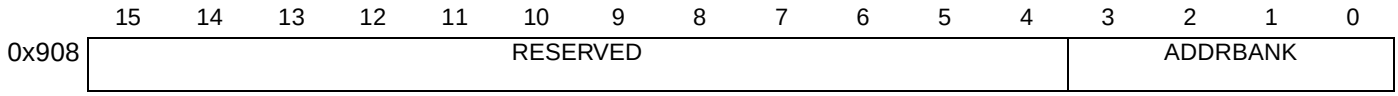



Address: *LinkBaseAddress* + 0x90C
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| ADDRHIGH | This value is used in conjunction with LNK_HDD_ADDRBANK to match the high part of the incoming address. It must be programmed to the same value as the corresponding link reg LNK_STREAM_CONF_2.DMA_HIGH_ADDR.<br>The 2 LSBs must be programmed to zero, as must the following bits: LNK_STREAM_CONF_3.DMA_LOW_ADDR[25:26] and LNK_STREAM_CONF_2.START_ADDR[16:17]. |

### LNK_HDD_ADDRBANK High part of the incoming address




Address: *LinkBaseAddress* + 0x908
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| ADDRBANK | This value is used in conjunction with LNK_HDD_ADDRHIGH to match the high part of the incoming address. This register should be programmed to the same value as the corresponding link register LNK_STREAM_CONF_2.DMA_BANK_ADDR. |

## LNK_HDD_BUFSIZE Buffer size

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x900 | RESERVED | | | | | | BUFSIZE | | | | | | | | | |

Address: *LinkBaseAddress* + 0x900
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BUFSIZE | Defines the buffer size, where buffer size = (LNK_HDD_BUFSIZE + 1) x 8KBytes.<br>When LNK_HDD_BUFSIZE is reached, the next address is filled by register LNK_STREAM_CONF_2 bits DMA_BANK_ADDR and DMA_HIGH_ADDR. |

# 15 Low power module (LPM)

## LPM_TIMER Low power timer

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0x400 | LPTimer[31:0] |
| 0x404 | LPTimer[63:32] |

Address: *LPCBaseAddress* + 0x400 ( LPTimer[31:0])and 0x404(LPTimer[63:32])
Type: Read/write
Reset value: UND

***Description***

These registers are the least significant and most significant words of the low-power timer register. These enable the least significant or most significant word to be written independently without affecting other words.

When either word of the register is written, the low power timer is stopped and the new value in this register is available to be written to the low power timer.

## LPM_TIMERSTART Low power timer start

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0x408 | Timer Start |

Address: *LPCBaseAddress* + 0x408
Type: Write only
Reset value: UND

***Description***

A write of any value to this register starts the low power timer counter. The counter is stopped and the register reset if either word of register LPM_TIMER is written. Setting LPM_TimerStart to zero does not stop the timer.

## LPM_ALARM Low power alarm

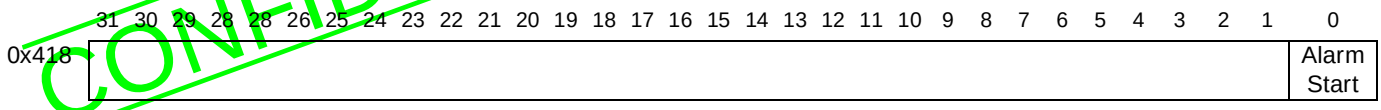
| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0x410 | LPAlarm[31:0] |
| 0x414 | LPAlarm[39:32] |

Address: *LPCBaseAddress* + 0x410 (LPAlarm[31:0]) and 0x414 (LPAlarm[39:32])
Type: Read/write
Reset value: UND

***Description***

These registers are the least significant and most significant words of the low power alarm. They are used to program the alarm register.

## LPM_ALARMSTART Low power alarm start

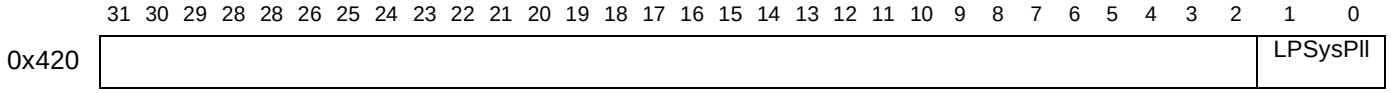



Address: *LPCBaseAddress* + 0x418
Type: Write only
Reset value: UND

***Description***

Any write to this register starts the low power alarm counter. The counter is stopped and the LPM_START register reset if either word of the LPM_TIMER register is written.

## LPM_SYSPLL System clock PLL



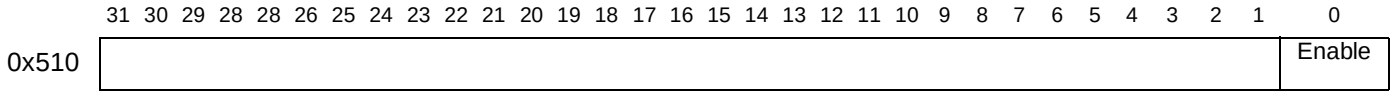

Address: *LPCBaseAddress* + 0x420
Type: Read/write
Reset value: UND

***Description***

This register controls the System Clock PLL operation when low-power mode is entered.

| Bitfield | Description |
|---|---|
| LPSysPll | 00: PLL off, 01: Illegal, 10: Illegal, 11: PLL on |

## LPM_WDENABLE Watchdog enable

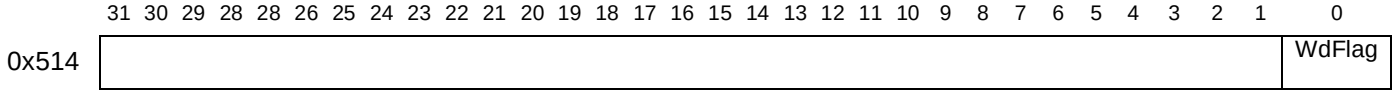



Address: *LPCBaseAddress* + 0x510
Type: Read/write
Reset value: 0

***Description***

Setting this register enables the low power alarm counter to be used as a watchdog timer.

## LPM_WDFLAG Watchdog flag




Address: *LPCBaseAddress* + 0x514
Type: Read only

***Description***

| Bitfield | Description |
|---|---|
| WdFlag | 0: Watchdog reset has not occurred.<br>1: Watchdog reset has occurred. |

# 16 Modem analog front-end interface (MOD)

## MOD_CONTROL MAFEIF control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x00 | Reserved | | | | | | start | run |

Address: ModemBaseAddress + 0x00
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| run | 1: The MAFEIF to start exchanging data with the AFE.<br>0: The MAFEIF will stop after completing the exchange of the current buffer-load of samples. |
| start | Indicates which of the two pairs of memory buffer pointers it should start off using:<br>0: Indicates Recieve0_pointer and Transmit0_pointer<br>1: Indicates Recieve1_pointer and Transmit1_pointer |

## MOD_STATUS MAFEIF status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x04 | Reserved | | missed | overflow | last | ctrl_empty | complete | idle |

Address: ModemBaseAddress + 0x04
Type: R
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| idle | 0: indicates that the MAFEIF is exchanging data with the AFE.<br>1: indicates that the run bit is low and the MAFEIF is not exchanging data. After the software clears the run bit, the MAFEIF only goes idle when it has finished exchanging the current buffer-load of samples. |
| complete | Set to '1' when a buffer-load of samples has been exchanged.<br>Cleared by writing to the Acknowledge register. |
| ctrl_empty | Set to '0' by writing to MOD_MAFE_CTRL.<br>Set to '1' when the MAFEIF has completed the control/status exchange. |
| last | Indicates the last pair of buffer pointers used by the DMA. |
| overflow | 1: Indicates that overflow has occurred (the MAFEIF has completed the exchange of another buffer-load of samples before the software has got round to acknowledging the previous buffer-load).<br>Cleared by writing to MOD_ACK. |
| missed | 1: Indicates that the memory latency is too high, causing a sample to be missed (the MAFEIF is exchanging samples faster than they can be read/written to/from the memory buffers).<br>Cleared by writing to MOD_ACK. |

## MOD_INT_ENABLE Interrupt enable

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x08 | Reserved | | | | | Intenable2 | Intenable1 | Intenable0 |

Address: ModemBaseAddress + 0x08
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| Intenable[2:0] | Enables interrupts connected to MAFEIF_status(2:0).<br>1: Indicates that the corresponding interrupt is enabled. |

## MOD_ACK Acknowledge

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0C | ACK | | | | | | | |

Address: ModemBaseAddress + 0x0C
Type: W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ACK | Clears the overflow, missed and complete flags in the MOD_MAFEIF_STATUS register. |

## MOD_BUFFER_SIZE Buffer size

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | SIZE | | | | | | | Reserved |

Address: ModemBaseAddress + 0x10
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| SIZE | This value must be a multiple of 2, it sets the buffer size (the number of 16-bit samples in a buffer). |

## MOD_MAFE_CTRL MAFE control

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | CNTVAL | | | | | | | | | | | | | | | |

Address: ModemBaseAddress + 0x14
Type: W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| CNTVAL | This number is the control value to send out to the MAFE |

## MOD_MAFE_STATUS MAFE status

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | STATUS | | | | | | | | | | | | | | | |

Address: ModemBaseAddress + 0x18
Type: R
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| STATUS | This number is the status value received from the MAFE. |

## MOD_RECEIVE0_POINTER Receive_memory_buffer_0 start address

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 | 1 0 |
|---|---|---|
| 0x20 | ADDR | Reserved |

Address: ModemBaseAddress + 0x20
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ADDR | This number is the start address of the receive_memory_buffer_0. |

## MOD_RECEIVE1_POINTER Receive_memory_buffer_1 start address

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 | 1 0 |
|---|---|---|
| 0x24 | ADDR | Reserved |

Address: ModemBaseAddress + 0x24
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ADDR | This number is the start address of the receive_memory_buffer_1. |

## MOD_TRANSMIT0_POINTER Transmit_memory_buffer_0 start address

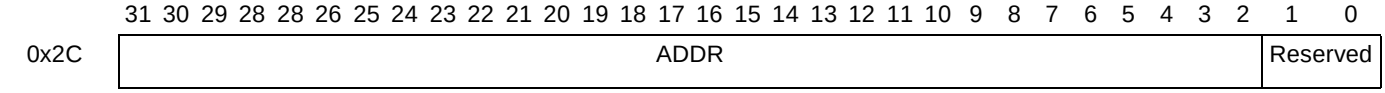
| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 | 1 0 |
|---|---|---|
| 0x28 | ADDR | Reserved |

Address: ModemBaseAddress + 0x28
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ADDR | This number is the start address of the transmit_memory_buffer_0. |

## MOD_TRANSMIT1_POINTER Transmit_memory_buffer_1 start address




Address: ModemBaseAddress + 0x2C
Type: R/W
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| ADDR | This number is the start address of the transmit_memory_buffer_1. |

# 17 MPEG DMA (MPEGDMA)

The MPEGDMA registers are accessed by bits 2 to 5 of the *dmacnt_and_peraddr* (peripheral address), inclusively.

The term “MPEGDMAn” refers to both MPEGDMA0 and MPEGDMA1 registers. These two set of registers have identical descriptions but different base addresses as given in Table 1: *Register block base variables* on page 2.

## MPEGDMAn_ABORT Abort the DMA operation

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x08 | Reserved | | | | | | | ABORT |

Address: *MPEGDMAnBaseAddress* + 0x08
Type: W
Reset value: 0

***Description***

While a data transfer is in progress, writing a “1” to this register will halt the data transfer and reset the DMA engine.

## MPEGDMAn_BLSIZE Data block dimension to be transferred

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | BLSIZE | | | | | | | | | | | | | | | |

Address: *MPEGDMAnBaseAddress* + 0x20
Type: W
Reset value: Undefined

***Description***

This register contain the overall block size to be transferred. Writing to this register starts the data transfer operation, it assumes that registers MPEGDMA_SRC_ADD and MPEGDMA_WHICHDEC have already been set. Writing to this register while the DMA engine is active has no effect and the new value is discarded.

Attempting to read a write only register will result in a return of all zeros.

## MPEGDMAn_BURSTSIZENumber of bytes to be transferred in one burst

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x00 | Reserved | | | BURSTSIZE | | | | |

Address: *MPEGDMAnBaseAddress* + 0x00
Type: W
Reset value: Undefined

***Description***

This register holds the number of bytes to be copied in one burst. It is a 5 bit number and allows burst sizes in the range 1 to 31($2^5$ - 1) bytes, zero means a 32 byte transfer will be performed.

Writing to this register while the data transfer engine is active will have effect only at the end of the present burst transfer.

## MPEGDMAn_CNTRL Interrupt control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1C | Reserved | | | | | | CNTRL2 | CNTRL1 |

Address: *MPEGDMAnBaseAddress* + 0x1C
Type: R/W
Reset value: Undefined

***Description***

This register masks the interrupt signal.

| Bitfield | Definition |
|---|---|
| CNTRL1 | 0: Interrupt disabled (Default), 1: Interrupt enabled |
| CNTRL2 | 0: Interrupt Mode - set to zero always |

## MPEGDMAn_HOLDOFF Holdoff for MPEG decoder

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x04 | Reserved | | | HOLDOFF | | | | |

Address: *MPEGDMAnBaseAddress* + 0x04
Type: W
Reset value: Undefined

***Description***

The value in this register is the hold-off time in cycles. After sampling the notcdreq_XX signal active, the signal is ignored until the burst size in bytes has been transferred, the last write cycle of the burst has completed, and the hold-off time has expired from the last write cycle completion.

Writing to this register while the data transfer engine is active will have effect only at the end of the present burst transfer.

## MPEGDMAn_INTACK Interrupt acknowledge

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x14 | Reserved | | | | | | | INTACK |

Address: *MPEGDMAnBaseAddress* + 0x14
Type: W
Reset value: 0

***Description***

Writing to this register acknowledges any pending interrupt and clears it; writing to this register at any other time has no effect.

## MPEGDMAn_SRCADD DMA source pointer

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | SRCADD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *MPEGDMAnBaseAddress* + 0x18
Type: W
Reset value: Undefined

***Description***

This register holds the address of the first byte of the block to be transferred. The data transfer operation can handle byte alignment. The contents of this register are unchanged after a transfer operation.

Writing to this register while the DMA engine is active has no effect and the new value is discarded.

## MPEGDMAn_STATUS Interrupt status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | Reserved | | | | | | STATUS2 | STATUS1 |

Address: *MPEGDMAnBaseAddress* + 0x10
Type: R
Reset value: Undefined

***Description***

This register gives the status of the interrupt and DMA engine.

| Bitfield | Description |
|---|---|
| STATUS1 | 0: No Interrupt, 1: Interrupt pending. |
| STATUS2 | 0: Inactive, 1: Active/busy. |

## MPEGDMAn_WHICHDECDMA destination pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0C | Reserved | | | | | | WHICHDEC | |

Address: *MPEGDMAnBaseAddress*+ 0x0C
Type: W
Reset value: Undefined

***Description***

This register holds the number of the MPEG decoder of the destination of the data transfer is placed in this register.Writing to this register while the data transfer engine is active will have effect only at the end of the present burst transfer.

The address placed on the memory bus as in the table below:

| Decoder | Destination address |
|---|---|
| 0 | Video |
| 1 | Audio |
| 2 | SubPicture |
| 3 | Not allowed |

# 18 On-screen display (OSD)

## OSD_ACT Active Signal

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | OAM | OAD[5:0] | | | | | |

Address: *VideoBaseAddress* + 0x3E
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| OAM | *OSD Active signal mode.* When this bit is set, OSD active signal is an input. When it is reset, OSD active signal is an output. Note that OSD active signal must never be driven when VID_CTL.EVI=1 and VID_OSD.OAM=0. |
| OAD[5:0] | *OSD Active signal delay.* These bits are used to define the delay of the OSD active signal corresponding to output OSD pixels. |

## OSD_BDW OSD Boundary Weight

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | OSD_BDW[5:0] | | | | | |

Address: *VideoBaseAddress* + 0x92
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BDW[5:0] | *OSD Boundary Weight.* These bits represent the value of mix_weight at the horizontal border of an OSD region or at a horizontal border of a transparent region within an OSD region. The OSD_BDW value is used in OSD filter formulae and has therefore no meaning when the OSD is in normal mode. |

## OSD_CFG OSD Configuration

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | | ENA | NOR | FIL |

Address: *VideoBaseAddress* + 0x91
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| FIL | *OSD Filter mode.* This bit is used to select the OSD filter mode. When this bit is set the anti-flutter filter mode will be active. When this bit is reset the anti-flicker filter mode will be active. Note that to activate one of the two filter modes the bit OSD_CFG.NOR has to be set. |
| NOR | *OSD Normal mode.* This bit is used to select the OSD normal mode. When this bit is reset the OSD normal mode will be active, which means no filter modes specified in OSD CFG.FIL will be active. When OSD_CFG.NOR is set, then the filter mode specified in OSD_CFG.FIL will be active. |
| ENA | *OSD Enable.* When this bit is set the OSD is enabled, when this bit is reset the OSD will be inactive. |

## OSD_OBP OSD Bottom Field Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | | | OBP[13:8] | | | | | |
| 2nd cycle | OBP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x2B
Type: Serial read/write
Reset value: 0
Synchronization: VSYNC bottom

***Description***

This register is written or read in two cycles: first cycle: OBP[13:8], second cycle: OBP[7:0]

The circularity is reset by a hardware reset or a bottom field VSYNC. The register holds the start address, in units of 256 bytes of the current OSD specification buffer for the bottom field. This specification will be decoded during bottom fields when OSD is enabled.

## OSD_OTP OSD Top Field Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | | | OTP[13:8] | | | | | |
| 2nd cycle | OTP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x2A
Type: Serial read/write
Reset value: 0
Synchronization: VSYNC top

***Description***

This register is written or read in two cycles: first cycle: OTP[13:8], second cycle: OTP[7:0]

The circularity is reset by a hardware reset or a top VSYNC. The register holds the start address, in units of 256 bytes, of the current OSD specification buffer for the top field. This specification will be decoded during top fields when OSD is enabled.

# 19 Overlay graphics & text (OGT)

## OGT_CTL Control Register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Not Used | Reserved | EnotO_pol | H2 | H1 | D | S | F |

Address: *SubPictureBaseAddress* + 0x40
Type: R/W
Reset value: 0
Synchronization: Vsync

***Description***

This register contains control bits for the overlay graphics and text.

| Bitfield | Description |
|---|---|
| F | Force_ogt command. When set, this bit indicates that the sub-picture unit is bypassed and VCD mode is active. |
| S | OGT decoder start command. When set, this bit it indicates the start of a new page. |
| D | Disable Display command. When set, this bit turns off the OGT display, but the decoding process continues. |
| H1 | Disable highlight region1. When set, this bit turns off highlighting inside the OGT display area. |
| H2 | Disable highlight region 2. When set, this bit turns off highlighting inside the OGT display area. |
| EnotO_pol | Sets the polarity of EnotO |

## OGT_LUT Main Look-up Table

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LUT[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x41
Type: R/W
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| LUT[7:0] | Writing to this register auto-increments the address in the main look-up table. Starting with color 0, for each color the Y component (8-bits) is written first followed by U, V and then K. The process continues for each color, up to 4 colors. |

## OGT_LUT_H1 Highlight 1 Look-up Table

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LUT_h1[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x42
Type: R/W
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| LUT_H1[7:0] | Writing to this register auto-increments the address in the highlight 1 look-up table.<br>For each color, starting with the color 0, the Y component (8-bits) is written first followed by U, V and then K.<br>The process continues for each color, up to 4 colors. |

## OGT_LUT_H2 Highlight 2 Look-up Table

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LUT_h2[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x43
Type: R/W
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| LUT_H2[7:0] | Writing to this register auto-increments the address in the highlight 2 look-up table.<br>For each color, starting with the color 0, the Y component (8-bits) is written first followed by U, V and then K.<br>The process continues for each color, up to 4 colors. |

## OGT_XDI Active area horizontal position offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x44 | Not Used | | | | | | XDI[9:8] | |
| 0x45 | XDI[7:0] | | | | | | | |

Address: SubPictureB*aseAddress* + 0x44 (OGT_XD1[9:8]) - 0x45 (OGT_XD1[7:0])
Type: R/W
Reset value: 0
Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| XDI[9:0] | This register value sets the horizontal offset of the left-hand side of the active OGT decode region.<br>The position is measured in number of pixels from the left-hand edge of the screen.<br>The true horizontal start position is XDI + OGT_XDO. |

## OGT_YDI Active area vertical position offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x46 | Not Used | | | | | | YDI[9:8] | |
| 0x47 | YDI[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x46 (OGT_YD1[9:8]) - 0x47 (OGT_YD1[7:0])
Type: R/W
Reset value: 0

Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| XDI[9:0] | This register value sets the vertical offset of the top of the active OGT decode region.<br>The position is measured in number of pixels from the top of the screen.<br>The true value of the vertical position of the top is YDI + OGT_YDO |

## OGT_HL2XO Highlight 2 area, Start X

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x48 | Not Used | | | | | | HL2XO[9:8] | |
| 0x49 | HL2XO[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x48 (OGT_HL2XO[9:8]) - 0x49 (OGT_HL2XO[7:0])
Type: R/W
Reset value: 0
Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| HL2XO[9:0] | This register value sets the horizontal position of the left-hand side of the highlight 2 region.<br>The position is measured in number of pixels from the left-hand side of the screen. |

## OGT_HL2YO Highlight 2 area, start Y

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4A | Not Used | | | | | | HL2YO[9:8] | |
| 0x4B | HL2YO[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x4A (OGT_HL2YO[9:8]) - 0x4B (OGT_HL2YO[7:0])
Type: R/W
Reset value: 0
Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| HL2YO[9:0] | This register value sets the vertical position of the top of the highlight 2 region.<br>The position is measured in number of pixels from the top of the screen. |

## OGT_HL2XI Highlight 2 area, end X

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4C | Not Used | | | | | | HL2XI[9:8] | |
| 0x4D | HL2XI[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x4C (OGT_HL2X1[9:8]) - 0x4D (OGT_HL2X1[7:0])
Type: R/W

Reset value: 0
Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| HL2XI[9:0] | This register value sets the horizontal position of the right-hand side of the highlight 2 region.<br>The position is measured in number of pixels from the left-hand edge of the highlight 2 OGT region. |

## OGT_HL2YI Highlight 2 area, End Y

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x4E | Not Used | | | | | | HL2YI[9:8] | |
| 0x4F | HL2YI[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x4E (OGT_HL2Y1[9:8]) - 0x4F (OGT_HL2Y1[7:0])
Type: R/W
Reset value: 0
Synchronization: Vsync

***Description***

| Bitfield | Description |
|---|---|
| HL2YI[9:0] | This register value sets the vertical position of the bottom of the highlight 2 region.<br>The position is measured in number of pixels from the top of the active highlight 2 region. |

## OGT_STAT Status register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Not Used | | | | | | R | E |

Address: *SubPictureBaseAddress* + 0x50
Type: R
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| E | RLD error. This bit is set when an RLD processing error is detected.<br>The RLD state machine automatically stops in its actual state. |
| R | Acknowledge not received after a request to the FIFO, the state machine is automatically reset into idle state. |

# 20 PES parser (PES)

## PES_CF1 PES Audio Decoding Control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x40 | SDT | | IAI | AUD_ID[4:0] | | | | |

Address: *VideoBaseAddress* + 0x40
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

This register controls the audio stream decoding and some miscellaneous parser functions.

| Bitfield | Description |
|---|---|
| AUD_ID | Audio Stream ID |
| IAI | Ignore Audio Stream ID |
| SDT | Store DTS not PTS.<br>When set, this bit causes the DTS stamps to be stored instead of the PTS stamps for the video parser. |

## PES_CF2 PES Video Parser Control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x41 | MOD[1:0] | | SS | IVI | VID_ID[3:0] | | | |

Address: *VideoBaseAddress* + 0x41
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

This register controls the video stream parser.

| Bitfield | Description |
|---|---|
| VID_ID[3:0] | Video Stream ID |
| IVI | Ignore Video Stream ID |
| SS | System Stream. This bit, when set, indicates that the stream sent to the parser is a system stream. To decode a pure video or audio stream this bit should be set to zero. |
| MOD[1:0] | 00: Automatic configuration. The parser will configure itself to decode the incoming stream.<br>01: MPEG-1 system stream with one data strobe input format<br>10: MPEG-2 PES separate data strobes input format (standard mode).<br>11: MPEG-2 whole PES video and whole PES audio, one after the other with single data strobe input format. |

## PES_TM1 DSM Trick Mode

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x42 | PES_TM1[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x42
Type: Read only
Reset value: 0
Synchronization: None

***Description***

This register stores the DSM trick mode bits. This register may be used only if the ES rate flag of the bit stream is set to zero.

## PES_TM2 PES Parser Status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x43 | | | | | | | M2 | DSA |

Address: *VideoBaseAddress* + 0x43
Type: Read only
Reset value: 2
Synchronization: None

***Description***

PES parser status register.

| Bitfield | Description |
|---|---|
| DSA | DSM association flag. This bit, when set indicates that the picture header present in the start code detector is associated to DSM values present in the **PES_TM1** register. |
| M2 | *MPEG-2 not MPEG-1.* This bit indicates, in automatic mode, if the current stream being decoded is an MPEG-2 or an MPEG-1 stream. |

## PES_TS PES Time Stamps

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x49 | PES_TS [7:0] | | | | | | | |
| 0x4A | PES_TS [15:8] | | | | | | | |
| 0x4B | PES_TS [23:16] | | | | | | | |
| 0x4C | PES_TS [31:24] | | | | | | | |
| 0x4D | | | | | | | TSA | PES_TS [32] |

Address: *VideoBaseAddress* + 0x49 (PES_TS [7:0]) to 0x4D (PES_TS [32:TSA])
Type: Read only
Reset value: 0
Synchronization: None

***Description***

These registers store the time stamps selected using the control bit in **PES_CF1**.

| Bitfield | Description |
|---|---|
| TSA | Time stamp association. When this bit is set it indicates that the picture to be decoded next (from which the header is available in the start code detector) has an associated time stamp available in **PES_TS**. |

# 21 Parallel input/output (PIO)

Each eight-bit PIO port has a set of eight-bit registers. Each of the eight bits of each register refers to the corresponding pin in the corresponding port.

## PIO_CLEAR_P*n*C2:0 Clear bits of P*n*C2:0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x28 | Clear_PC0[7:0] | | | | | | | |
| 0x38 | Clear_PC1[7:0] | | | | | | | |
| 0x48 | Clear_PC2[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x28 (PIO_Clear_PnC0), 0x38 (PIO_Clear_PnC1)and 0x48 (PIO_Clear_PnC2)

Type: Write only

***Description***

**PIO_Clear_P*n*C2:0** allows the bits of registers **PIO_P*n*C2:0** to be cleared individually. Writing a ‘1’ in one of these register clears the corresponding bit in the corresponding **PIO_P*n*C2:0** register, while a ‘0’ leaves the bit unchanged.

## PIO_CLEAR_P*n*COMP Clear bits of P*n*Comp

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Clear_PComp[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x58

Type: Write only

***Description***

**PIO_Clear_P*n*Comp** allows bits of **PIO_P*n*Comp** to be cleared individually. Writing a ‘1’ in this register clears the corresponding bit in the **PIO_P*n*Comp** register, while a ‘0’ leaves the bit unchanged.

## PIO_CLEAR_PnMASK Clear bits of P*n*Mask

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Clear_PMask[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x68

Type: Write only

***Description***

**PIO_Clear_P*n*Mask** allows bits of **PIO_P*n*Mask** to be cleared individually. Writing a ‘1’ in this register clears the corresponding bit in the **PIO_P*n*Mask** register, while a ‘0’ leaves the bit unchanged.

## PIO_CLEAR_PnOUT Clear bits of P*n*Out

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Clear_POut[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x08
Type: Write only

***Description***

**PIO_Clear_P*n*Out** allows bits of **PIO_P*n*Out** to be cleared individually. Writing a '1' in this register clears the corresponding bit in the **PIO_P*n*Out** register, while a '0' leaves the bit unchanged.

## PIO_PnC2:0 PIO configuration

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x20 | ConfigData0[7:0] | | | | | | | |
| 0x30 | ConfigData1[7:0] | | | | | | | |
| 0x40 | ConfigData2[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x20 (PIO_PnC0), 0x30 (PIO_PnC1) and 0x40 (PIO_PnC2)
Type: Read/write
Reset value: 0

***Description***

There are three configuration registers (PIO_PnC0, PIO_PnC1 and PIO_PnC2) for each port, which are used to configure the PIO port pins. Each pin can be configured as an input, output, bidirectional, or alternative function pin (if any), with options for the output driver configuration.

Three bits, one bit from each of the three registers, configure the corresponding bit of the port. The configuration of the corresponding I/O pin for each valid bit setting is given in the table below.

| **P*n*C2[*y*]** | **P*n*C1[*y*]** | **P*n*C0[*y*]** | **Bit *y* configuration** | **Bit *y* output** |
|---|---|---|---|---|
| 0 | 0 | 1 | Bidirectional | Open drain. |
| 0 | 1 | 0 | Output | Push-pull. |
| 0 | 1 | 1 | Bidirectional | Open drain. |
| 1 | 0 | 0 | Input | Hi-Z. |
| 1 | 0 | 1 | Input | Hi-Z. |
| 1 | 1 | 0 | Alternative function output | Push-pull. |
| 1 | 1 | 1 | Alternative function bidirectional | Open drain. |

**Table 51 PIO bit configuration encoding**

The **PIO_P*n*C** registers are each mapped onto two additional addresses **PIO_Set_P*n*C** and **PIO_Clear_P*n*C** so that bits can be set or cleared individually.

## PIO_PnCOMP PIO input comparison

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PComp[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x50
Type: Read/write
Reset value: 0

***Description***

The input compare register **PIO_P*n*Comp** can be used to cause an interrupt if the input value differs from a fixed value.

The input data from the PIO ports pins will be compared with the value held in **PIO_P*n*Comp**. If any of the input bits is different from the corresponding bit in the **PIO_P*n*Comp** register and the corresponding bit position in **PIO_P*n*Mask** is set to 1, then the internal interrupt signal for the port will be set to 1.

The compare function is sensitive to changes in levels on the pins. For the comparison to be seen as a valid interrupt by an interrupt handler, the change in state on the input pin must be longer in duration than the interrupt response time.

The compare function is operational in all configurations for each PIO bit, including the alternative function modes.

The **PIO_P*n*Out** register is mapped onto two additional addresses **PIO_Set_P*n*Out** and **PIO_Clear_P*n*Out** so that bits can be set or cleared individually.

## PIO_PnIN PIO input

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PIn[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x10
Type: Read only
Reset value: 0

***Description***

The data read from this register will give the logic level present on the input pins of the port at the start of the read cycle to this register. Each bit reflects the input value of the corresponding bit of the port. The read data will be the last value written to the register regardless of the pin configuration selected.

## PIO_PnMASK PIO input comparison mask

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PMask[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x60
Type: Read/write
Reset value: 0

***Description***

When a bit is set to 1, the compare function for the internal interrupt for the port is enabled for that bit. If the respective bit (7 to 0) of the input is different from the corresponding bit in the **PIO_P*n*Comp** register, then an interrupt is generated.

The **PIO_P*n*Mask** register is mapped onto two additional addresses **PIO_Set_P*n*Mask** and **PIO_Clear_P*n*Mask** so that bits can be set or cleared individually.

## PIO_PnOUT PIO output

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| POut[7:0] | | | | | | | |

Address: *PIOnBaseAddress* + 0x00
Type: Read/write
Reset value: 0

***Description***

This register holds output data for the port. Each bit defines the output value of the corresponding bit of the port.

The **PIO_P*n*Out** register is mapped onto two additional addresses **PIO_Set_P*n*Out** and **PIO_Clear_P*n*Out** so that bits can be set or cleared individually.

## PIO_SET_PnC2:0 Set bits of P*n*C2:0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x24 | Set_PC0[7:0] | | | | | | | |
| 0x34 | Set_PC1[7:0] | | | | | | | |
| 0x44 | Set_PC2[7:0] | | | | | | | |

Address: *PIOnBaseAddress* + 0x24 (PIO_Set_PnC0), 0x34 (PIO_Set_PnC1) and 0x44 (PIO_Set_PnC2)
Type: Write only

***Description***

PIO_Set_PnC2:0 allow the bits of registers PIO_PnC2:0 to be set individually. Writing a ‘1’ in one of these registers sets the corresponding bit in the corresponding PIO_PnC2:0 register, while a ‘0’ leaves the bit unchanged.

## PIO_SET_PnCOMP Set bits of P*n*Comp

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Set_PComp[7:0] | | | | | | | |

Address: *PIOnBaseAddress* + 0x54
Type: Write only

***Description***

PIO_Set_PnComp allows bits of PIO_PnComp to be set individually. Writing a ‘1’ in this register sets the corresponding bit in the PIO_PnComp register, while a ‘0’ leaves the bit unchanged.

## PIO_SET_PnMASK Set bits of P*n*Mask

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Set_PMask[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x64
Type: Write only

***Description***

PIO_Set_PnMask allows bits of PIO_PnMask to be set individually. Writing a ‘1’ in this register sets the corresponding bit in the PIO_PnMask register, while a ‘0’ leaves the bit unchanged.

## PIO_SET_PnOUT Set bits of P*n*Out

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Set_POut[7:0] | | | | | | | |

Address: *PIO*n*BaseAddress* + 0x04
Type: Write only

***Description***

PIO_Set_PnOut allows bits of PIO_PnOut to be set individually. Writing a ‘1’ in this register sets the corresponding bit in the PIO_PnOut register, while a ‘0’ leaves the bit unchanged.

# 22 PWM and counter module (PWC)

## PWM_nCAPTUREEDGE PWM *n* Capture event definition

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 | 1 0 |
|---|---|---|
| 0x30 | | PWM_0CaptureEdge |
| 0x34 | | PWM_1CaptureEdge |
| 0x38 | | PWM_2CaptureEdge |
| 0x3C | | PWM_3CaptureEdge |

Address: *PWMBaseAddress* + 0x30 (PWM_0CaptureEdge) to 0x3C (PWM_3CaptureEdge)
Type: Read/write
Reset value: Undefined

***Description***

The code in register PWM_nCaptureEdge defines what constitutes an event on input pin **CaptureIn***n*. Possible events are rising edge, falling edge, both or neither (in other words, disabled).

| Bitfield | Description |
|---|---|
| CaptureEdge | 01: Capture on rising edge<br>10: Capture on falling edge<br>11: Capture on rising or falling edge<br>00: Capture disabled |

## PWM_nCAPTUREVAL PWM *n* capture value

| | 31 30 29 28 28 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|---|---|
| 0x10 | PWM0CaptureVal |
| 0x14 | PWM1CaptureVal |
| 0x18 | PWM2CaptureVal |
| 0x1C | PWM3CaptureVal |

Address: *PWMBaseAddress* + 0x10 (PWM_0CAPTUREVAL0) to 0x1C (PWM_3CAPTUREVAL)
Type: Read only
Reset value: Undefined

***Description***

Each of the four capture value registers holds the 32-bit counter value at the time of the last event occurring at the corresponding CaptureIn pin. When an input event occurs on input CaptureInn, the value of the counter in register PWM_CAPTURECOUNT at that time is captured in register PWM_nCAPTUREVAL. The value can be any 32-bit value.

When an input event occurs, an interrupt is generated provided the register bit PWM_INTENABLE.IntN is set to 1. Register bit PWM_INTSTATUS.Intn becomes 1, and can be reset by writing 1 to register PWM_INTACK.IntAck.

The counter is not stopped nor reset by any of these events.

## PWM_nCOMPAREOUTVAL PWM *n* compare output value

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Compare OutVal2 |

Address: *PWMBaseAddress* + 0x40 to 0x4C
Type: Read/write
Reset value: Undefined

***Description***

This register holds the value which will be written to the PWM_COMPAREOUTN pin when the compare value in PWM_nCOMPAREVAL matches the counter value PWM_CAPTURECOUNT.

## PWM_nCOMPAREVAL PWM *n* compare value

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | PWM0CompareVal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x24 | PWM1CompareVal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x28 | PWM2CompareVal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2C | PWM3CompareVal | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *PWMBaseAddress* + 0x20 (PWM0CompareVal) to 0x2C (PWM3CompareVal)
Type: Read/write
Reset value: Undefined

***Description***

Each of the four compare registers PWM_nCOMPAREVAL in the module can be set to any 32-bit value. When the counter in register PWMCaptureCount reaches the value of register PWM_nCOMPAREVAL, two things happen:

1 An interrupt is generated if the register bit PWM_INTENABLE.IntN is set to 1. Register bit PWM_INTSTATUS.IntN becomes 1, and can be reset by writing 1 to register bit PWM_INTACK.IntAck.

2 Pin PWM_nCOMPAREOUT takes on the value set in register PWM_nCOMPAREVAL.

The counter is not stopped nor reset by any of these events.

## PWM_nVAL PWM *n* pulse width

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | | | | | | | | | | | | | | | | | | | | | | | | | PWM0Val | | | | | | | |
| 0x04 | | | | | | | | | | | | | | | | | | | | | | | | | PWM1Val | | | | | | | |
| 0x08 | | | | | | | | | | | | | | | | | | | | | | | | | PWM2Val | | | | | | | |
| 0x0C | | | | | | | | | | | | | | | | | | | | | | | | | PWM3Val | | | | | | | |

Address: *PW*MBaseAddress + 0x00 (PWM_0VAL) to 0x0C (PWM_3VAL)
Type: Read/write
Reset value: Undefined

***Description***

These registers hold the counter values, which are used to determine the width of the pulse generated on the PWMOut pin. PWMOut pulse width = (PWMVal + 1) x prescaled clock period

If PWMVal is 255 then PWMOut does not go low.

## PWM_CAPTURECOUNT PWM capture/compare counter

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CaptureCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *PWMBaseAddress* + 0x64
Type: Read/write
Reset value: Undefined

***Description***

This register holds the shared capture/compare counter used by all the capture and compare functions.

The capture/compare counter is clocked from the prescaled system clock. The prescaling factor, and therefore the period represented by one count, is determined by the value of field CaptureClkValue in register PWM_CONTROL. The factor can be from 1 to 32.

The counter is enabled by setting register bit PWM_CONTROL.CaptureEnable to 1. When it is disabled (PWM_CONTROL.CaptureEnable=0), none of the capture or compare functions work. PWM_CAPTURECOUNT can be read or written at any time.

When the capture/compare counter reaches its maximum count of #FFFFFFFF, it wraps round to count up from zero again.

## PWM_CONTROL PWM control register

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | Capture Enable | PWM Enable | Capture ClkValue | | | | | PWM ClkValue | | | |

Address: *PWMBaseAddress* + 0x50
Type: Read/write
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| PWMClkValue | PWM clock prescale factor 0-15 (divide clock by value + 1) |
| CaptureClkValue | Capture/compare clock prescale factor 0-31 (divide clock by value + 1) |
| PWMEnable | Enables PWM counter when = 1 |
| CaptureEnable | Enables capture/compare counter when = 1 |

## PWM_COUNT PWM output counter

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | Count | | | | | | | |

Address: *PWMBaseAddress* + 0x60
Type: Read/write (but see text)
Reset value: Undefined

***Description***

PWM output counter. The counter (in register PWM_COUNT) is enabled by setting register bit PWM_CONTROL.PWMEnable to 1. When it is disabled (PWM_CONTROL.PWMEnable=0), pin PWM_OUT is forced low. PWM_COUNT is writable at any time but can have a synchronization latency.

## PWM_INTACK PWM interrupt acknowledge

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | CmpIA3 | CmpIA2 | CmpIA1 | CmpIA0 | CptIA3 | CptIA2 | CptIA1 | CptIA0 | IntAck |

Address: *PWMBaseAddress* + 0x5C
Type: Write only

***Description***

| Bitfield | Description |
|---|---|
| IntAck | Interrupt acknowledge: write 1 to reset PWMInt to 0 |
| CmpIA0 | Compare 0 interrupt acknowledge: write 1 to reset status bit |
| CmpIA1 | Compare 1 interrupt acknowledge: write 1 to reset status bit |
| CmpIA2 | Compare 2 interrupt acknowledge: write 1 to reset status bit |
| CmpIA3 | Compare 3 interrupt acknowledge: write 1 to reset status bit |
| CptIA0 | Capture 0 interrupt acknowledge: write 1 to reset status bit. |
| CptIA1 | Capture 1 interrupt acknowledge: write 1 to reset status bit |
| CptIA2 | Capture 2 interrupt acknowledge: write 1 to reset status bit |
| CptIA3 | Capture 3 interrupt acknowledge: write 1 to reset status bit |

## PWM_INTENABLE PWM interrupt enable

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | CmpIE3 | CmpIE2 | CmpIE1 | CmpIE0 | CptIE3 | CptIE2 | CptIE1 | CptIE0 | IntEn |

Address: *PWMBaseAddress* + 0x54
Type: Read/write
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| IntEn | PWM counter overflow interrupt enable |
| CptIE0 | Capture 0 interrupt enable: 1 = enabled |
| CptIE1 | Capture 1 interrupt enable: 1 = enabled |
| CptIE2 | Capture 2 interrupt enable: 1 = enabled |
| CptIE3 | Capture 3 interrupt enable: 1 = enabled |
| CmpIE0 | Compare 0 interrupt enable: 1 = enabled |
| CmpIE1 | Compare 1 interrupt enable: 1 = enabled |
| CmpIE2 | Compare 2 interrupt enable: 1 = enabled |
| CmpIE3 | Compare 3 interrupt enable: 1 = enabled |

## PWM_INTSTATUS PWM interrupt status

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | Cmp3 | Cmp2 | Cmp1 | Cmp0 | Cpt3 | Cpt2 | Cpt1 | Cpt0 | Int |

Address: *PWMBaseAddress* + 0x58
Type: Read only
Reset value: Undefined

***Description***

| Bitfield | Description |
|---|---|
| Int | 1= PWM counter overflow |
| Cpt0 | Capture 0 interrupt: 1 = interrupt |
| Cpt1 | Capture 1 interrupt: 1 = interrupt |
| Cpt2 | Capture 2 interrupt: 1 = interrupt |
| Cpt3 | Capture 3 interrupt: 1 = interrupt |
| Cmp0 | Compare 0 interrupt: 1 = interrupt |
| Cmp1 | Compare 1 interrupt: 1 = interrupt |
| Cmp2 | Compare 2 interrupt: 1 = interrupt |
| Cmp3 | Compare 3 interrupt: 1 = interrupt |

# 23 SmartCard interface (SCI)

## SCI_n_CLKCON SmartCard *n* clock control

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved. Write 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Enable | Source |

Address: *SmartCard*n*BaseAddress* + 0x04
Type: Write only
Reset value: 0

***Description***

This register controls the source of the clock and determines whether the SmartCard clock output is enabled. The programmable divider and the output are reset when the enable bit is set to 0.

| Bitfield | Description |
|---|---|
| Enable | SmartCard clock generator enable bit.<br>0: Stop clock, set output low and reset divider.<br>1: Enable clock generator. |
| Source | Selects source of SmartCard clock.<br>0: Selects global clock.<br>1: Selects external pin. |

## SCI_n_CLKVAL SmartCard *n* clock

| 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | ScClkVal | | | | |

Address: *SmartCard*n*BaseAddress* + 0x00
Type: Write only
Reset value: 0

***Description***

This register determines the SmartCard clock frequency. The 5-bit value given in the register is multiplied by 2 to give the division factor of the input clock frequency. For example, if **Sc*n*SIkVal**=8 then the input clock frequency is divided by 16. The value “zero” must not be written into this register.

The divider is updated with the new value for the divider ratio on the next rising or falling edge of the output clock.

# 24 Sub-picture decoder (SPD)

## SPD_CTL1 Control Register 1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BOT | TOP | P | B | H | V | D | S |

Address: *SubPictureBaseAddress* + 0x00
Type: Read/write
Reset value: 0
Synchronization: VSYNC (field)

***Description***

This register contains control bits for the subpicture decoder.

| Bitfield | Description |
|---|---|
| S | *Sub-Picture Decoder Start command.* When set, this bit indicates the start of a new subpicture unit. The subpicture decoder will then reset the local time reference counter. The state is sampled on each VSYNC. |
| D | *Decoder Active.* This bit is set by the decoder when, at shadow register update, the Sub-picture start bit is sampled high. When the decoder active bit is reset decoding is disabled and can only be re-enabled by the decoder start bit. |
| V | *Display Active.* When reset, this bit turns off the sub-picture display, however, decoding still goes on even when the display is disabled. When decoding is disabled using the 'Decoder active' bit then the display is automatically disabled. |
| H | *Highlight Enable.* When set, this bit turns on highlighting inside the sub-picture display area. |
| B | *Bypass.* When set, this bit puts the run-length decoder into transparent mode. This allows standard 2-bit-per-pixel bitmaps to be fed into the sub-picture decoder. |
| P | *Sub-Picture Pause.* When this bit is set the 90KHz clock in sub-picture is paused. |
| TOP | When this bit is set to 1, the new subpicture command is taken into account one top field.<br>If TOP and BOT are both set to 0, subpicture command are executed on any field. |
| BOT | When this bit is set to 1, the new subpicture command is taken into account one bottom field. |

## SPD_CTL2 Control Register 2

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | Rc |

Address: *SubPictureBaseAddress* + 0x02
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

This register contains control bits for the subpicture decoder.

| Bitfield | Description |
|---|---|
| Rc | *Reset Lut #2.* When set, this bit resets the auto-increment address counter. |

## SPD_HCN Highlight Region Contrast

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x16 | e2[3:0] | | | | e1[3:0] | | | |
| 0x17 | p[3:0] | | | | b[3:0] | | | |

Address: *SubPictureBaseAddress* + 0x16 and 0x17
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These registers contain the highlight contrast map values.

## SPD_HCOL Highlight Region Color

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x14 | e2[3:0] | | | | e1[3:0] | | | |
| 0x15 | p[3:0] | | | | b[3:0] | | | |

Address: *SubPictureBaseAddress* + 0x14 and 0x15
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These registers contain the highlight color map values.

## SPD_HLEX Highlight Region End X

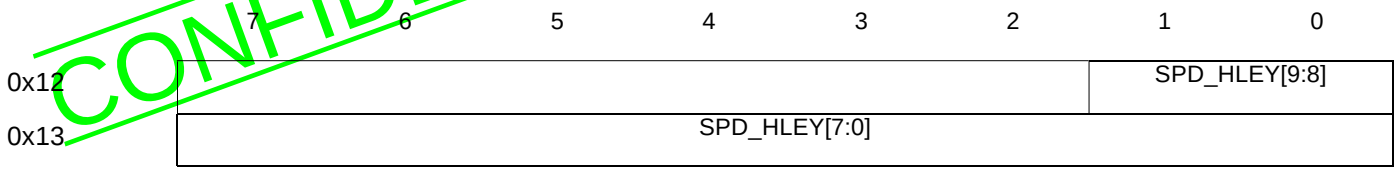
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | | | | | | | | SPD_HLEX[9:8] |
| 0x11 | SPD_HLEX[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x10 (SPD_HLEX[9:8]) and 0x11 (SPD_HLEX[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

Highlight region end position X-coordinate. This register is also used as the OGT highlight region1.

## SPD_HLEY Highlight Region End Y

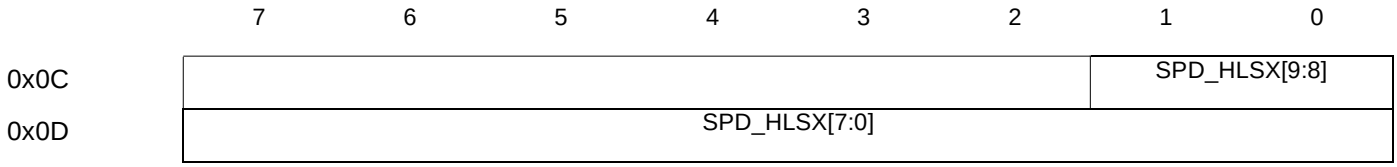



Address: *SubPictureBaseAddress* + 0x12 (SPD_HLEY[9:8]) and 0x13 (SPD_HLEY[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

Highlight region end position Y-coordinate. This register is also used as the OGT highlight region1.

## SPD_HLSX Highlight Region Start X

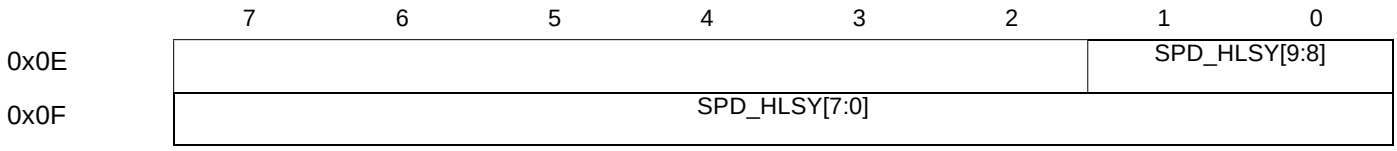



Address: *SubPictureBaseAddress* + 0x0C (SPD_HLSX[9:8]) and 0x0D (SPD_HLSX[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

Highlight region start position X-coordinate. This register is also used as the OGT highlight region1.

## SPD_HLSY Highlight Region Start Y




Address: *SubPictureBaseAddress* + 0x0E (SPD_HLSY[9:8]) and 0x0F (SPD_HLSY[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

Highlight region start position Y-coordinate. This register is also used as the OGT highlight region1.

## SPD_LUT Main Lookup Table

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPD_LUT[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x03
Type: Write
Reset value: 0
Synchronization: None

***Description***

This register allows input of the main lookup table. Writing to this register auto-increments the address in the lookup table. For each color, starting with color 0, the Y component (8-bit) is written first followed by U and V. The process continues for each color up to 16 colors.

## SPD_SPR Soft Reset

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | SPD_SPR |

Address: *SubPictureBaseAddress* + 0x01
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

This register resets the subpicture decoder.

| Bitfield | Description |
|---|---|
| SPR | *Sub Picture Reset*. When this bit is set the sub-picture decoder and its FIFOs are reset. Note that the sub-picture decoder is also reset by a hard reset or a global soft reset. |

## SPD_SXD0 Sub-picture display area

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x24 | | | | | | | | SPD_SXD0[9:8] |
| 0x25 | SPD_SXD0[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x24 (SPD_SXD0[9:8]) and 0x25 (SPD_SXD0[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description:***

These registers contain the parameters which define the subpicture display area within the subpicture decode area. The value represents an offset from the corresponding parameter used to define the subpicture decode area. For example: The true horizontal start position of the subpicture display will be equal to XDO + SXDO.

This register is also used for OGT, its value sets the horizontal position of the left-hand side of the active OGT decode region, the position is measured in number of pixels from the left-hand edge of the screen. The minimum value is 3.

## SPD_SXD1 Subpicture Display Area

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x28 | | | | | | | SPD_SXD1[9:8] | |
| 0x29 | SPD_SXD1[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x28 (SPD_SXD1[9:8]) and 0x29 (SPD_SXD1[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These registers contain the parameters which define the subpicture display area within the subpicture decode area. The value represents an offset from the corresponding parameter used to define the subpicture decode area.

This register is also used for OGT, its value sets the width of active OGT decode region, the position is measured in number of pixels from the left-hand edge of the OGT region.

## SPD_SYD0 Subpicture Display Area

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x26 | | | | | | | SPD_SYD0[9:8] | |
| 0x27 | SPD_SYD0[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x26 (SPD_SYD0[9:8]) and 0x27 (SPD_SYD0[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These registers contain the parameters which define the subpicture display area within the subpicture decode area. The value represents an offset from the corresponding parameter used to define the subpicture decode area.

This register is also used for OGT, its value sets the vertical position of the left-hand side of the active OGT decode region, the position is measured in number of pixels from the top edge of the screen.

## SPD_SYD1 Subpicture Display Area

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x2A | | | | | | | SPD_SYD1[9:8] | |
| 0x2B | SPD_SYD1[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x2A (SPD_SYD1[7:0]) and 0x2B (SPD_SYD1[9:8])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These registers contain the parameters which define the subpicture display area within the subpicture decode area. The value represents an offset from the corresponding parameter used to define the subpicture decode area.

This register is also used for OGT, its value sets the height of the active OGT decode region, the position is measured in number of pixels from the top edge of the OGT region.

## SPD_XD0 Sub-picture X Offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x04 | | | | | | | SPD_XD0[9:8] | |
| 0x05 | SPD_XD0[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x04 (SPD_XD0[9:8]) and 0x05 (SPD_XD0[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register value sets the horizontal position of the left hand side of the active subpicture decode region. The position is measured in number of pixels from the left hand edge of the screen.

## SPD_YD0 Sub-picture Y Offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x06 | | | | | | | SPD_YD0[9:8] | |
| 0x07 | SPD_YD0[7:0] | | | | | | | |

Address: *SubPictureBaseAddress* + 0x06 (SPD_YD0[9:8]) and 0x07 (SPD_YD0[7:0])
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register value sets the vertical position of the top of the active subpicture decode region. The position is measured in number of pixels from the top edge of the screen.

# 25 Synchronous serial controller (SSC)

## SSC_*n*_BRG SSC *n* baud rate generation

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BRG | | | | | | | | | | | | | | | |

Address: *SSCnBaseAddress* + 0x00
Type: Read/write
Reset value: 1

***Description***

This address is dual purpose. When reading, the current 16-bit counter value is returned. When a value is to this address, the 16-bit reload register is loaded with that value.

## SSC_n_CON SSC *n* control

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | LPB | EN | MS | SR | PO | PH | HB | BM | | | |

Address: *SSCnBaseAddress* + 0x0C
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| BM | SSC Data Width Selection<br>0000: Reserved. Do not use this combination.<br>0001: 2 bits<br>0010: 3 bits<br>... : ...<br>1111: 16 bits |
| HB | SSC Heading Control Bit<br>0: LSB first<br>1: MSB first |
| PH | SSC Clock Phase Control Bit<br>0: Pulse in second half cycle<br>1: Pulse in first half cycle |

| Bitfield | Description |
|---|---|
| PO | SSC Clock Polarity Control Bit<br>0: Clock idles at logic 0<br>1: Clock idles at logic 1 |
| SR | SSC Software Reset<br>0: Device is not reset<br>1: All functions are reset while this bit is set |
| MS | SSC Master Select Bit<br>0: Slave mode<br>1 Master mode |
| EN | SSC Enable Bit<br>0: Transmission and reception disabled<br>1: Transmission and reception enabled |
| LPB | SSC Loopback Bit<br>0: Disabled<br>1: Shift register output is connected to shift register input |

## SSC_n_I2C SSC *n* I²C control

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | AD10 | ACKG | STOPG | STRTG | I2CM |

Address: *SSC*n*BaseAddress* + 0x18
Type: Read/write
Reset value: 0

***Description***

| Bitfield | Description |
|---|---|
| I2CM | SSC I²C control bit<br>0: Disabled, 1: Enable I²C features |
| STRTG | SSC I²C generate START condition<br>0: Disabled, 1: Generate a START condition |
| STOPG | SSC I²C generate STOP condition<br>0: Disabled,1:Generate a STOP condition |
| ACKG | SSC I²C generate acknowledge bits<br>0: Disabled,1: Generate acknowledge bits when receiving |
| AD10 | SSC I²C 10-bit addressing control<br>0: Disabled,1: Use 10 bit addressing |

## SSC_*n*_IEn SSC *n* interrupt enable

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | ARBLEN | STOPEN | AASEN | | PEEN | REEN | TEEN | TIEN | RIEN |

Address: *SSC*n*BaseAddress* + 0x10
Type: Read/write
Reset value: 0

***Description***

This register holds the interrupt enable bits, which can be used to mask the interrupts.

| Bitfield | Description |
|---|---|
| RIEN | 0: Receiver Buffer Full Interrupt Enable<br>1: Receiver buffer interrupt enabled |
| TIEN | 0: Transmitter Buffer Empty Interrupt Enable<br>1: Transmitter buffer empty interrupt enabled |
| TEEN | 0: Transmit Error Interrupt Enable<br>1: Transmit error interrupt enabled |
| REEN | 0: Receive Error Interrupt Enable<br>1: Receive error interrupt enabled |
| PEEN | 0: Phase Error Interrupt Enable<br>1: Phase error interrupt enabled |
| AASEN | 0: $I^2C$ Addressed As Slave Interrupt Enable<br>1: Addressed as slave interrupt enabled |
| STOPEN | 0: $I^2C$ Stop Condition Interrupt Enable<br>1: Stop condition interrupt enabled |
| ARBLEN | 0: $I^2C$ Arbitration Lost Interrupt Enable<br>1: Arbitration lost interrupt enabled |

## SSC_n_RBUF SSC *n* receive buffer

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RD[15:0] | | | | | | | | | | | | | | | |

Address: *SSC*n*BaseAddress* + 0x08
Type: Read only
Reset value: 0

***Description***

Receive buffer data **D15** to **D0**.

## SSC_n_SLAD SSC *n* slave address

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | SL[9:7] | | | SL[6:0] | | | | | | |

Address: *SSC*n*BaseAddress* + 0x1C
Type: Write only
Reset value: 0

***Description***

The slave address is written into this register. If the address is a 10-bit address it is written into bits 9:0. If the address is a 7-bit address then it is written into bits 6:0.

## SSC_n_STAT SSC *n* status

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | BUSY | ARBL | STOP | AAS | CLST | PE | RE | TE | TIR | RIR |

Address: *SSC*n*BaseAddress* + 0x14
Type: Read only
Reset value: 2, i.e. all active bits clear except **TIR**.

***Description***

| Bitfield | Description |
|---|---|
| RIR | 0: Receiver buffer full flag<br>1: Receiver buffer full |
| TIR | 0: Transmitter buffer empty flag<br>1: Transmitter buffer empty |
| TE | 0: Transmit error flag<br>1: Transmit error set |
| RE | 0: Receive error flag<br>1: Receive error set |
| PE | 0: Phase error flag<br>1: Phase error set |
| CLST | 0: $I^2C$ clock stretch flag<br>1: Clock stretching in operation |
| AAS | 0: $I^2C$ addressed as slave flag<br>1: Addressed as slave device |
| STOP | 0: $I^2C$ stop condition flag<br>1: Stop condition detected |
| ARBL | 0: $I^2C$ arbitration lost flag<br>1: arbitration lost |
| BUSY | 0: $I^2C$ bus busy flag<br>1: $I^2C$ bus busy |

## SSC_n_TBUF SSC *n* transmit buffer

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TD[15:0] | | | | | | | | | | | | | | | |

Address: *SSC*n*BaseAddress* + 0x04
Type: Write only
Reset value: 0

***Description***

Transmit buffer data **D15** to **D0**.

# 26 Teletext interface (Ttxt)

Five dedicated DENC registers program the teletext encoding in various areas of the Vertical Blanking Interval (VBI) of each field. These registers are DEN_TTX1 to DEN_TTX4, and DEN_TTXM.

## TTXT_ABORT Teletext Abort

Address: *TtxtBaseAddress* + 0x24
Type: Write only

***Description***

This register is write only. Any write to this address causes the teletext interface to abort the current operation. The state of the teletext operation is reset, and the teletext data transfer is interrupted. The DMA engine is reset only after the current word read or write is complete.

## TTXT_ACKODDEVEN Teletext Acknowledge odd or even

Address: *TtxtBaseAddress* + 0x20
Type: Write only

***Description***

This register acknowledges the odd/even toggle interrupt. Any write to the **TTXT_ACKODDEVEN** register clears the **Odd** and **Even** bits of the **TTXT_INTSTATUS** register.

## TTXT_DMAADDRESS Teletext DMA address

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | DmaAddress | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Address: *TtxtBaseAddress* + 0x00
Type: Read/write
Reset value: Undefined

***Description***

The **TTXT_DMAADDRESS** register is a 32-bit register, which specifies the base address in memory for the DMA transfer from memory.

| Bit field | Function |
|---|---|
| DmaAddress | The base address for DMA transfer of data to or from memory. |

## TTXT_DMACOUNT Teletext DMA count

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | | | | | | | | | | | | | | | | | | | | | | DmaCount | | | | | | | | | | |

Address: *TtxtBaseAddress* + 0x04
Type: Read/write
Reset value: Undefined

***Description***

The **TTXT_DMACOUNT** register specifies the number of bytes to be transferred from memory during the DMA operation. A write to this register also starts the teletext output operation.

- For teletext output operation, this value must be: 46 bytes x *number_of_teletext_lines_to_send*
- For teletext input operation, the value must be: 42 bytes x *number_of_teletext_lines_to_receive*

## TTXT_INTENABLE Teletext Interrupt enable

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1C | | | | | | EvenEnable | OddEnable | InOutCompleteEn |

Address: *TtxtBaseAddress* + 0x1C
Type: Read/write
Reset value: Undefined

***Description***

The **TTXT_INTENABLE** register allows masking of the **TTXT_INTSTATUS** register.

| Bitfield | Function |
|---|---|
| InOutCompleteEn | Enable teletext input or output operation completed interrupt. |
| OddEnable | Enable odd field interrupt. |
| EvenEnable | Enable even field interrupt. |

## TTXT_INTSTATUS Teletext Interrupt status

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Even | Odd | InOut Complete |

Address: *TtxtBaseAddress* + 0x18
Type: Read only
Reset value: Undefined

***Description***

The **TTXT_INTSTATUS** register gives the current state of the teletext operations. If the appropriate bits in the interrupt enable register are set then interrupts can be driven by the state of this register.

| Bitfield | Function |
|---|---|
| InOutComplete | Teletext input or output operation completed. |
| Odd | Current (video encoder) field is ODD. |
| Even | Current (video encoder) field is EVEN. |

## TTXT_OUTDELAY Teletext Output delay

| | 31 | 30 | 29 | 28 | 28 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | | | | | | | | | | | | | | | | | | | | | | | | Delay | | | | | | | | |

Address: *TtxtBaseAddress* + 0x08
Type: Read/write
Reset value: Undefined

***Description***

This register programs the delay, in 27 MHz clock periods, from the rising edge of **TTXT_REQUEST** to the first valid teletext data bit, i.e. **TTXT_DATA** starting to transmit.

# 27 SDRAM block move (USD)

## USD_BMS Block move size

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | USD_BMS[15:8] | | | | | | | |
| 2nd cycle | USD_BMS[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x09
Type: Serial read/write
Reset value: 0

***Description***

This register holds the number of words to be moved in a block move. The second write to the register with USD_BMS non-zero enables (but does not launch) the block move mode. This register must be written before the associated USD_BRP. The register is reset by a hardware reset.

## USD_BRP Memory read pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x88 | | | | | USD_BRP[19:16] | | | |
| 0x89 | USD_BRP[15:8] | | | | | | | |
| 0x8A | USD_BRP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x88 (1st cycle), 0x89 (2nd cycle) & 0x8A (3rd cycle)
Type: Serial read/write
Reset value: Undefined

***Description***

This register holds the source address of the block to be moved. USD_BRP is an offset from the base of SDRAM in units of 64-bit words. It points to the base of the block to be copied. When USD_BMS is not zero, the third write to USD_BRP launches the block move process, taking into account the values in USD_BMS and USD_BWP. The pointer is reset by a hardware reset.

## USD_BWP Memory write pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x8C | | | | | USD_BWP[19:16] | | | |
| 0x8D | USD_BWP[15:8] | | | | | | | |
| 0x8E | USD_BWP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x8C (1st cycle), 0x8D (2nd cycle) & 0x8E (3rd cycle)
Type: Serial read/write
Reset value: Undefined

***Description***

When a block move is to be executed (i.e. USD_BMS is not zero), this register holds the destination address. USD_BWP defines an offset from the base of the SDRAM, in units of 64-bit words.This points to the base of the area to be written. The pointer is reset by a hardware reset.

## USD_BSK Block skip

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xA4 | | | | | | USD_BSK[18:16] | | |
| 0xA5 | USD_BSK[15:8] | | | | | | | |
| 0xA6 | USD_BSK[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xA4, 0xA5, 0xA6
Type: Read/write
Reset value: Undefined

***Description***

When a 2D block move is to be executed, this register holds the number of 64-bits words to skip before finding the beginning of the next line to move. As this register has 19 bits, it can contain negative values.

# 28 Video decoder (VID)

## VID_656 Enable 656 mode

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x32 | | | | | | | | E6M |

Address: *VideoBaseAddress* + 0x32
Type: R/W
Synchronization: Edge triggered

***Description***

| Bitfield | Description |
|---|---|
| E6M | Enable CCIR 656 mode:<br>0: Basic digital video output is sent to port YC[7:0].<br>1: Synchro patterns are inserted in video stream according to CCIR 656 recommendations. |

## VID_ABG Start of audio bit buffer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1C | Reserved | | | | ABG[14:8] | | | |
| 0x1D | ABG[7:0] | | | | | | | |

Address: *VideoBaseAddr*ess + 0x1C and 0x1D
Type: Read/write
Reset value: 0

***Description***

The register holds the starting address of the audio bitstream buffer, defined in units of 2 Kbits. If the audio bit buffer starts at address 0, then this register does not need to be set up, since its reset state is 0. When all the registers corresponding to the settings of the bit buffer is done (VID_ABG, VID_ABS), an audio soft reset must be done for values to be taken in account. In other words it must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

## VID_ABL Audio Bit Buffer Level

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1E | Reserved | | | | ABL[14:8] | | | |
| 0x1F | ABL[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x1E and 0x1F
Type: Read only
Reset value: 0

***Description***

This register holds the current level of occupation of the audio bit buffer, defined in units of 2 Kbits. It can be read at any time for the monitoring of the audio bit buffer level. When VID_ABL is greater than or equal to the value held in the VID_ABT register, the status bit VID_STA.ABF (audio bit buffer full) becomes set. When VID_ABL is zero, the status bit VID_STA.ABE (audio bit buffer empty) becomes set.

## VID_ABS Audio Bit Buffer Stop

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x20 | Reserved | | | | ABS[14:8] | | | |
| 0x21 | ABS[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x20 and 0x21
Type: Read/write
Reset value: 0

***Description***

This register holds the address of the top of the audio bit buffer, defined in units of 2 Kbits. The space allocated to the audio bit buffer starts at the address defined by the VID_ABG register, or, by default, 0. The end address of the audio bit buffer is: (128 x ABS) + 127

VID_ABS must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

## VID_ABT Audio Bit Buffer Threshold

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x22 | Reserved | | | | ABT [14:8] | | | |
| 0x23 | ABT [7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x22 and 0x23
Type: Read/write
Reset value: 0
Synchronization: Edge Triggered

***Description***

This register holds the level of occupancy of the audio bit buffer, in units of 2 Kbits, which when reached causes the status bit VID_STA.ABF to become set.

If the bit CFG_CCF.PBO is set, then transfer of data to the audio bit buffer is prevented if the bit buffer level is at or above the level defined in the VID_ABT register. If VID_ABT is set to a value equal to the top of the bit buffer, then this automatic mechanism will ensure that overflow never occurs.

## VID_BCK_Y Background color Y

| | 7 | 0 |
|---|---|---|
| 0x98 | **BCK_Y[7:0]** | |

Address: *VideoBaseAddress* + 0x98
Access: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register stores the value of the Y component of the background color mixable with the video or the still picture plane.

## VID_BCK_U Background color U

| | 7 | 0 |
|---|---|---|
| 0x99 | BCK_U[7:0] | |

Address: *VideoBaseAddress* + 0x9A
Access: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register stores the value of the U component of the background color mixable with the video or the still picture plane.

## VID_BCK_V Background color V

| | 7 | 0 |
|---|---|---|
| 0x9A | BCK_V[7:0] | |

Address: *VideoBaseAddress* + 0x99
Access: Read/write
Reset value: FF
Synchronization: VSYNC

***Description***

This register stores the value of the V component of the background color mixable with the video or the still picture plane.

## VID_BFC Backward Chroma Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5E | Reserved | BFC [14:8] | | | | | | |
| 0x5F | BFC [7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x5E and 0x5F
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the chrominance buffer of the backward prediction frame picture, defined in units of 256 bytes.

## VID_BFP Backward Frame Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x12 | Reserved | BFP [14:8] | | | | | | |
| 0x13 | BFP [7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x12 and 0x13

Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the luminance buffer of the backward prediction frame picture, defined in units of 256 bytes.

## VID_CDCOUNT Bit Buffer Input Counter

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | CDcount [23:16] | | | | | | | |
| 2nd cycle | CDcount [15:8] | | | | | | | |
| 3rd cycle | CDcount [7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x67
Type: Serial read only
Reset value: 0

***Description***

These three registers are accessed serially. They hold the number of bytes input to the bit buffer. The byte pointer is reset by a hardware reset, a global soft reset or a video soft reset.

## VID_CFG_CCF Chip Configuration

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | EOU | PBO | EC3 | EC2 | ECK | EDI | EVI |

Address: *VideoBaseAddress* + 0x01
Type: R/W
Reset value: 0
Synchronization: Edge Triggered

***Description***

| Bitfield | Description |
|---|---|
| EVI | Enable video interface. When this bit is reset the digital video output (YC7 -YC0) is put into its high impedance state and the internal PIXCLK disabled. This bit must be set for normal operation and for reduced power mode (if the display interface is used). It is reset in low power mode. |
| EDI | Enable SDRAM interface. When this bit is reset the SDRAM interface (DD63-DD0, AA8-AA0, RAS1, RAS0, CAS, OE and WE) and the signal CDREQ are put into their high impedance state. This bit must be set for normal operation and for reduced power mode. It is reset in low power mode. |
| ECK | Enable clocks. When this bit is reset, all internal clocks are disabled. This bit must be set for normal operation and for reduced power mode. It is reset in low power mode. |
| EC2 | Enable “clock 2". (“clock 2" is an internal clock controlling MPEG decoding). When this bit is reset, the internal “clock 2" is disabled.<br>This bit must be set for normal operation, and reset in reduced and low power modes. |
| EC3 | Enable “clock 3". (“Clock 3" is an internal clock controlling SDRAM accesses). When this bit is reset, the internal “clock 3" is disabled.<br>This bit must be set for normal operation and for reduced power mode (in reduced power mode, MPEG decoding is not running but SDRAM accesses - display, OSD... - are still active).<br>It is reset in low power mode (in low power mode SDRAM accesses are not active: no display, no SDRAM refresh). |
| PBO | Prevent bit buffer overflow. When this bit is set, bit buffer overflow (and thus the loss of data) is prevented by disabling the transfer of data from the compressed data FIFO to the bit buffer whenever the bit buffer level reaches the threshold defined in the VID_VBT or VID_ABT register. |
| EOU | Enable overflow/underflow errors. When this bit is reset overflow and underflow errors are not treated internally. This bit must be set for normal operation. |

## VID_CFG_MCF Memory Refresh Interval

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | RFI[6:0] | | | | | | |

Address: *VideoBaseAddress* + 0x00
Type: R/W
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| RFI[6:0] | SDRAM refresh interval. This is defined in units of 32 primary clock periods.<br>0: Refresh disabled<br>1: Refresh enabled. For example, if 2048 rows must be refreshed every 32 ms for a SDRAM clock of 100MHz, the following value must be stored in MCF.RFI[6:0]:<br>$32 \times \frac{10^{-3}}{2048} \times \frac{100 \times 10^{6}}{24} =$ 65decimal, 1000001binary |

## VID_CFG_DRC SDRAM configuration

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MY64 | | MRS | | P1 | P0 | ERQ | SDR |

Address: *VideoBaseAddress* + 0x38
Type: R/W

Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| SDR | Synchronous DRAM mode. This bit enables the Synchronous DRAM mode. To launch the automatic init sequence, it is necessary first to enable SDRAM interface (set bit SDR) and then set bit MRS with another write access. |
| ERQ | Enable Processes Requests. This bit when reset disables all processes requests to the local memory controller. This bit has to be set for normal operation. |
| P[1:0] | Clk3 phase to MemClk. The following procedure has to be executed to make sure the SDRAM interface is properly initialized: P[1:0] = 0, write 4 x 32 bits words to SDRAM, read back the 4 words from SDRAM, if they do not read back correctly, increment the value of P[1:0] until the read is correct. |
| MRS | Mode Register Set. Only for Synchronous DRAM use. When this bit is set the special power-on-reset sequence and mode register programming is carried out.When this bit is reset the procedure is inhibited. |
| MY64 | 1: Shared SDRAM is based on 64Mbit<br>0: Shared SDRAM is based on 16Mbit |

## VID_CFG_GCF General configuration

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3A | OnePixDel | A3Rq | A3DI | CDR | | SCK | A3M | TRST_ SPDIF |

Address: *VideoBaseAddress* + 0x3A
Type: R/W
Reset value: 0
Synchronization: None

***Description***

| Bitfield | Description |
|---|---|
| TRST_SPDIF | When this bit is set to 0, SPDIF is tristated. |
| A3M | This bit has to be set to 0 |
| SCK | Audio strobe clock select.<br>When set, this bit selects “clock 2" (“clock 2" is an internal clock controlling MPEG decoding) as serial strobe period. Otherwise, “clock 3" (“clock 3" is an internal clock controlling SDRAM accesses) is selected. |
| CDR | Compressed data register<br>00: External strobes, 01: Audio strobes, 10: Subpicture strobes, 11: Video strobes |
| A3DI | This bit has to be set to 0 |
| A3Rq | This bit has to be set to 0 |
| OnePixDel | 1: Shift one pixel to the right for block-to-row |

## VID_CSO SRC chrominance offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6C | CSO[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x6C
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set up with a value calculated from the fractional part of the pan vector. If no pan vector is defined, this register can be left in its reset (default) state. The method of calculation of the CSO value is given in the **VID_PAN** register description.

## VID_CTL Decoding Control

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x02 | ERU | ERS | | | | PRS | SRS | EDC |

Address: *VideoBaseAddress* + 0x02
Type: Read/write
Synchronization: Edge triggered

***Description***

| Bitfield | Description |
|---|---|
| EDC | Enable decoding: This bit must be set to allow decoding. |
| SRS | Soft reset: To generate a soft reset, this bit must be kept set for a duration of at least 80 SDRAM clock cycles (660 ns with a 121.5 MHz primary clock). |
| PRS | Pipeline reset: To generate a pipeline reset, this bit must be kept set for a duration of at least 4 SDRAM clock cycles (40 ns with a 100 MHz SDRAM clock).<br>Reserved. Write 0. |
| ERS | Enable pipeline reset on severe error: When this bit is set, a pipeline reset is automatically generated in case of Severe Error (more than DFS macroblocks decoded). |
| ERU | Enable pipeline reset on picture decode error: When this bit is set, a pipeline reset is automatically generated in case of Picture Decode Error (less than DFS macroblocks decoded). |

## VID_CWL CD write launch

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x31 | | | | | | | | CWL |

Address: *VideoBaseAddress* + 0x31
Type: R/W
Synchronization: Edge triggered

***Description***

| Bitfield | Description |
|---|---|
| CWL | Launch Compress Data Write FIFO reset procedure for trick-modes.<br>Automatically set to zero once reset procedure is completed |

## VID_DCF Display Configuration

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x74 | | FPO | BLL | BFL | FNF | FLY | ENPOL | FPM |
| 0x75 | | | EVD | | DSR | OSDPOL | | |
| 0xD6 | Reserved (Set to 0) | | | | | ZOOM4 | CFB | LFB |

Address: *VideoBaseAddress* + 0x74, 0x75 and 0xD6

Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

| Bitfield | Description |
|---|---|
| LFB | 0: The luma samples of the **field** are sent to the vertical filter.<br>1: The luma samples of the **frame** are sent to the vertical filter. |
| CFB | 0: The chroma samples of the **field** are sent to the vertical filter.<br>1: The chroma samples of the **frame** are sent to the vertical filter. |
| ZOOM4 | When this bit is set, only one line in two is sent from memory to the vertical filter. This mode is used for zoom/4 during play. |
| OSDPOL | *OSD polarity:* |
| FPM | *Enable display field:* When this bit is set to 1, the field displayed on the next VSYNC is that set by bit FPO. |
| ENPOL | *E-not-O polarity:* To be configured for block-to-row |
| FLY | 0: The current picture is displayed from external memory<br>1: Mode not supported |
| FNF | Frame not Field. This bit is only used during on-the-fly decoding. The bit must be set if a frame picture is going to be displayed, or reset if field picture are to be decoded on-the-fly. For classical decoding (field or frame) the bit is always 1. |
| BFL | *Blank First Line.* If this bit is set the first active line of a picture is blanked. (Used in letter box format display). |
| BLL | *Blank Last Line.* If this bit is set, the last active line of a picture is blanked. (Used in letter box format display). |
| FPO | *Field polarity* - 1 for bottom field, 0 for top field |
| DSR | *Disable SRC.* When this bit is set, both luminance and chrominance SRCs (sample rate converters) are disabled. In this case no horizontal filtering can occur, as would be required when the horizontal resolution of the decoded picture is equal to the horizontal resolution of the display. |
| EVD | *Enable video display.* When this bit is reset, the video output has a constant value of Y=16, $C_B$=$C_R$=128. OSD is still displayed. |

## VID_DFC Displayed Chroma Frame Pointer

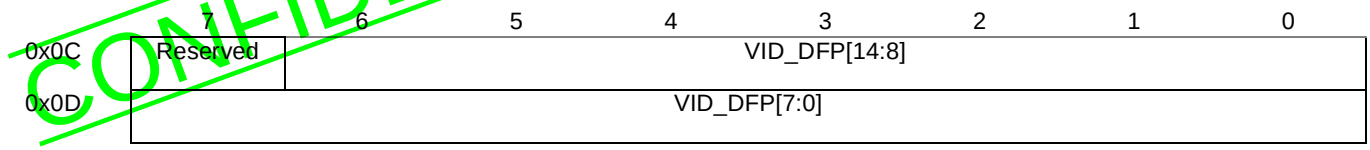
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x58 | Reserved | VID_DFC[14:8] | | | | | | |
| 0x59 | VID_DFC[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x58 and 0x59
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register holds the start address, defined in units of 256 bytes, of the chroma frame which is currently being displayed. When a new value is written this is used at the start of the next field. When VID_DFC is set to same value as VID_RFC (i.e. the decoder is writing the reconstructed picture into the buffer which is being displayed), bit VID_TIS.OVW must be set.

## VID_DFP Displayed Luma Frame Pointer



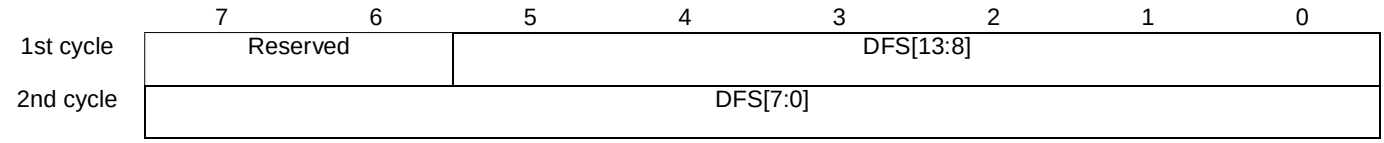

Address: *VideoBaseAddress* + 0x0C and 0x0D
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register holds the start address, defined in units of 256 bytes, of the luma frame which is currently being displayed. When a new value is written this is used at the start of the next field.

When VID_DFP is set to the same value as VID_RFP (i.e. the decoder is writing the reconstructed picture into the buffer which is being displayed), bit VID_TIS.OVW must be set.

## VID_DFS Decoded frame size



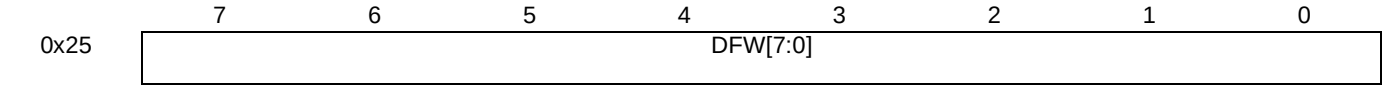

Address: *VideoBaseAddress* + 0x24
Type: Serial read/write
Reset value: 0
Synchronization: DSYNC

***Description***

The circularity is reset by a software reset.

| Bitfield | Description |
|---|---|
| DFS | This register field is set up with a value equal to the number of macroblocks in the decoded picture. This is derived from the *horizontal_size* and *vertical_size* values transmitted in the sequence header. |

## VID_DFW Decoded frame width




Address: *VideoBaseAddress* + 0x25
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register is set up with the width in macroblocks of the decoded picture. This is derived from the *horizontal_size* value transmitted in the sequence header.

## VID_FFC Forward Chroma Frame Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5C | Reserved | | | | FFC[14:8] | | | |
| 0x5D | FFC[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x5C and 0x5D
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the forward prediction chroma frame picture buffer, defined in units of 256 bytes.

## VID_FFP Forward Luma Frame Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x10 | | | FFP[13:8] | | | | | |
| 0x11 | FFP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x10 and 0x11
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the forward prediction luma frame picture buffer, defined in units of 256 bytes.

## VID_HDF Header Data FIFO

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | HDF[15:8] | | | | | | | |
| 2nd cycle | HDF[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x66
Type: Serial read only
Reset value: Undefined

***Description***

When the start code detector has found a start code, the header data FIFO must be read in order to identify the start code and if required to obtain the header data. The start code identification procedure is described in the STi5508 datasheet.

Before reading the header FIFO, status bit VID_STA.HFE should be checked to ensure that it is not empty. If bit VID_STA.HFF is set then the header FIFO contains at least 66 bytes of data.

As the start code detection is managed with 16-bit words, two successive reads are needed to access the whole 16-bit word; you must always perform an even number of reads before start code search restart (by software or with DSYNC signal).

The byte pointer is reset by a hardware reset, a global soft reset or a video reset.

## VID_HDS Header Search

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Read cycle | | | | | SCM | | | |
| Write cycle | | | | | | SOS | QMI | HDS |

Address: *VideoBaseAddress* + 0x69
Type: Read only bit SCM, write only HDS, QMI and SOS.
Reset value: 0
Synchronization: None

***Description***

This register controls the header search when it is written to, and returns the location of the start code when read.

| Bitfield | Description |
|---|---|
| HDS | Writing a 1 to this bit starts a header search. Completion of the header search is indicated by the setting of bit **VID_STA.SCH**. |
| QMI | This bit is used to control access to the inverse quantizer tables.<br>1: Select the intra table<br>0: Select the non-intra table.<br>For example, to write a new intra table, write VID_HDS.QMI = 1 then write 64 weights to VID_QMW. |
| SOS | Stop on First Slice. This bit when set allows the start code detector to stop on the first slice start code of a picture (0x00000101). To allow mismatches, this bit must be used in conjunction with **VID_HDS.SCM**. |
| SCM | Start Code on MSB. This bit indicates in which byte the start code is located. If this bit is set then VID_HDF[15:8] contains the start code; otherwise VID_HDF[7:0] contains the start code. |

## VID_ITM Interrupt Mask

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3C | CWR | ERR | SWR | TR_OK | ABF | SFF | AFF | ABE |
| 0x60 | PDE | SER | BMI | HFF | PNC | ERC | DEI | PII |
| 0x61 | PSD | VST | VSB | BBE | BBF | HFE | CFF | SCH |

Address: *VideoBaseAddress* + 0x3C, 0x60 and 0x61
Type: Read/write
Reset value: 0 (all interrupts disabled)
Synchronization: None

***Description***

Bit definitions are given in register*VID_STA* on page 259.

Any bit set in this register will enable the corresponding interrupt. An interrupt is generated whenever a bit in the **VID_STA** register changes from 0 to 1 and the corresponding mask bit is set.

## VID_ITS Interrupt Status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3D | CWR | ERR | SWR | TR_OK | ABF | SFF | AFF | ABE |
| 0x62 | PDE | SER | BMI | HFF | PNC | ERC | DEI | PII |
| 0x63 | PSD | VST | VSB | BBE | BBF | HFE | CFF | SCH |

Address: *VideoBaseAddress* + 0x3D, 0x62 and 0x63
Type: Read only
Reset value: 0

***Description***

Bit definitions are given in register*VID_STA* on page 259.

After the clocks have been enabled, the state changes to be the same as that of VID_STA. When a bit in the VID_STA register changes from 0 to 1, the corresponding bit in the VID_ITS register is set, irrespective of the state of VID_ITM. If the corresponding bit of VID_ITM is set, the interrupt is asserted. Reading the most significant byte of VID_ITS clears it, leaving IRQ in its de-asserted (high) state.

## VID_LCK Register lock

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x7B | | | | | | | | LDP |

Address: *VideoBaseAddress* + 0x7B
Type: R/W
Reset value: 0

***Description***

When bit LDP is set to 1, registers VID_CFG_MCF, VID_CFG_CCF and VID_CFG_DRC are locked into an non-writable state.

## VID_LSO SRC Luminance Offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x6A | LSO[7:0] | | | | | | | |

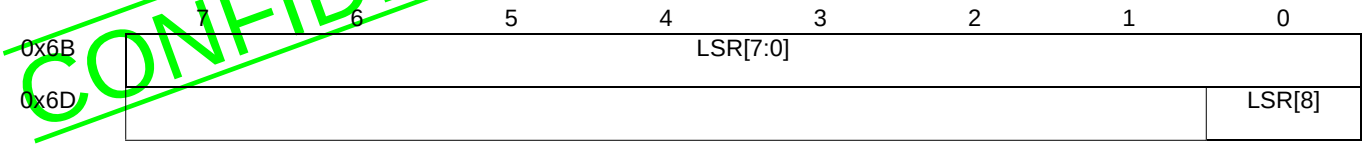
Address: *VideoBaseAddress* + 0x6A
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set up with a value calculated from the fractional part of the pan vector. If no pan vector is defined, this register can be left in its reset (default) state.

The method of calculation of the LSO value is given in the VID_PAN register description.

## VID_LSR SRC Luma/Chroma Resolution




Address: *VideoBaseAddress* + 0x6Band 0x6D
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

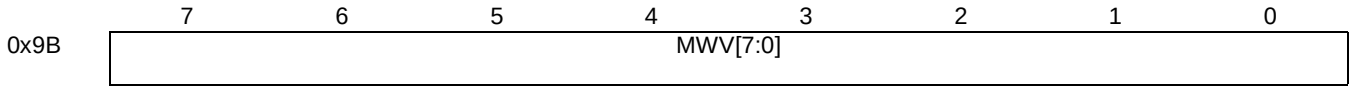
This register holds the upsampling factor of the luminance SRC (sample rate converter). It applies on chroma also.

The upsampling factor is equal to 256/LSR. Table 52 gives some examples of upsampling factors, where in each case the displayed picture has a nominal width of 720 pels. Also shown are the numbers of valid pels generated, “N”, calculated as shown in the STi5508 datasheet. Displayed picture widths other than 720 are supported.

| Decoded picture width | LSR | N |
|---|---|---|
| 640 | 228 | 715 |
| 640 | 227 | 718 |
| 544 | 193 | 717 |
| 544 | 192 | 721 |
| 480 | 170 | 717 |
| 480 | 169 | 722 |
| 352 | 125 | 713 |
| 352 | 124 | 719 |
| 704 | 250 | 717 |
| 704 | 249 | 720 |

**Table 52 Example upsampling factors**

## VID_MWV Mix Weight Video




Address: *VideoBaseAddress* + 0x9B
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register holds the value of the mix between the background color and the video. When **VID_MWV** = 0xFF then the background color is being displayed and at 0 the video is being displayed.

## VID_OUT Output of 4:2:2 display

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x90 | | | | | NOS | SPO | LAY | |

Address: *VideoBaseAddress* + 0x90
Type: Read/write
Reset value: $1100_2$
Synchronization: VSYNC

***Description***

This register controls the content of the 4:2:2 output from the display mixing unit and the order of the OSD and sub-picture planes. The bits work in combination as shown in Table 53.

| Bitfield | Description |
|---|---|
| LAY | The number of layers to be displayed in front of the video in the 4:2:2 output. |
| SPO | 0: OSD plane in front of Sub picture plane<br>1: Sub picture plane in front of OSD plane |
| NOS | 1: Do not include the OSD in the 4:2:2 output. |

| LAY | NOS | Sub-picture in front (SP0=1) | OSD in front (SP0=0) |
|---|---|---|---|
| 00-01 | Any | Video only | Video only |
| 10 | 0 | Video + OSD | Video + sub-picture |
| | 1 | Video only | Video + sub-picture |
| 11 | 0 | Video + OSD + sub-picture | Video + sub-picture + OSD |
| | 1 | Video + sub-picture | Video + sub-picture |

**Table 53 Encoding of LAY and NOS fields of VID_OUT**

## VID_PAN Pan/Scan Horizontal Vector Integer Part

| | 7 | 2 | 0 |
|---|---|---|---|
| 0x2C | | PAN[10:8] | |
| 0x2D | PAN[7:0] | | |

Address: *VideoBaseAddress* + 0x2C and 0x2D
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set up with the integer part of the horizontal pan/scan vector. The horizontal pan/scan vector defines, in the decoded picture, the location of the first displayed luminance sample relative to the first luminance sample in the line.

The **VID_LSO** and **VID_CSO** registers are set up with the fractional part of the horizontal pan/scan vector, as follows:

PSV = ⌊horizontal pan/scan vector⌋, where ⌊x⌋ indicates the integer part of x.

**VID_LSO** = 256 x (horizontal pan/scan vector - PSV)

**VID_CSO** = **VID_LSO** / 2 (+ 1 if PSV is odd)

## VID_PFH Picture F-parameters Horizontal

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x04 | BFH[3:0] | | | | FFH[3:0] | | | |

Address: *VideoBaseAddress* + 0x04
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register contains parameters of the picture to be decoded. These parameters are extracted from the bit stream.

| Bitfield | MPEG-1 | MPEG-2 |
|---|---|---|
| FFH | Set to *full_pel_forward_vector* of the picture header. | Set to *forward_horizontal_f_code* of the picture coding extension. |
| | Set to *forward_f_code* of the picture header. | |
| BFH | Set to *full_pel_backward_vector* of the picture header. | Set to *backward_horizontal_f_code* of the picture coding extension. |
| | Set to *backward_f_code* of the picture header. | |

## VID_PFV Picture F-parameters Vertical

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x05 | BFV[3:0] | | | | FFV[3:0] | | | |

Address: *VideoBaseAddress* + 0x05
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register contains parameters of the picture to be decoded. These parameters are extracted from the bitstream.

In MPEG-1 mode (i.e. when **VID_PPR2.MP2** is reset), **VID_PFV** is not used.

| Bitfield | Description |
|---|---|
| FFV | The *forward_vertical_f_code* of the picture coding extension. |
| BFV | The *backward_vertical_f_code* of the picture coding extension. |

## VID_PPR1 Picture Parameters 1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x06 | | OTF | PCT[1:0] | | DCP[1:0] | | PST[1:0] | |

Address: *VideoBaseAddress* + 0x06
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register contains parameters of the picture to be decoded. These parameters are extracted from the bit stream.

In MPEG-1 mode (i.e. when **VID_PPR2.MP2** is reset), only PCT has to be set; the other bits must be reset.

| Bitfield | Description |
|---|---|
| PST | Set to the *picture_structure* bits of the MPEG-2 picture coding extension.<br>00: Frame picture. This value is illegal in the MPEG-2 variable.<br>01: Top field.<br>10: Bottom field.<br>11: Frame picture. |
| DCP | Set equal to *intra_dc_precision* of the picture coding extension. The value “11”, defining a precision of 3 bits, is not allowed. |
| PCT | Set to the two least significant bits of *picture_coding_type* in the picture header. |
| OTF | When set this bit directs the decoded data in the block-to-row memory to be displayed directly. The picture data is not reconstructed in memory. |

## VID_PPR2 Picture Parameters 2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x07 | | MP2 | TFF | FRM | CMV | QST | IVF | AZZ |

Address: *VideoBaseAddress* + 0x07
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register contains parameters of the picture to be decoded. These parameters are extracted from the bitstream. In MPEG-1 mode, all bits must be reset to 0.

***Description***

| Bitfield | Description |
|---|---|
| AZZ | This bit is set equal to the *alternate_scan* bit of the picture coding extension. |
| IVF | This bit is set equal to the *intra_vlc_format* bit of the picture coding extension. |
| QST | This bit is set equal to the *q_scale_type* bit of the picture coding extension. |
| CMV | This bit is set equal to the *concealment_motion_vectors* bit of the MPEG-2 picture coding extension. It indicates that motion vectors are coded for intra macroblocks. |
| FRM | This bit is set equal to the *frame_pred_frame_dct* bit of the picture coding extension. |
| TFF | This bit is set equal to the *top_field_first* bit of the MPEG-2 picture coding extension. |
| MP2 | *MPEG-2 mode*. When this bit is set, the STi5508 expects an MPEG-2 video bitstream. If it is reset, then an MPEG-1 bitstream is expected. |

## VID_PTH Panic threshold

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x2E | | | FPAN | PEN | NFW | PTH[10:8] | | |
| 0x2F | PTH[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x2E and 0x2F
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

The panic threshold defines a block-to-row DRAM fullness level. In the block-to-row converter, there are two counters indicating the number of luma and chroma words stored. If the luma counter pointer is less than PTH or the chroma counter pointer is less than PTH divided by 2, a PANIC flag is set.

The panic threshold is programmed in units of DRAM cells.

| Bitfield | Description |
|---|---|
| PTH | Panic mode threshold. |
| NFW | Near Forward. This bit allows the user to select which prediction direction will be retained for bidirectional macrob-locks in panic mode. Forward if set, backward otherwise. |
| PEN | Panic mode enable. When set this bit allows the decoding to set the panic flag and enter the panic mode. |
| FPAN | Force Panic Mode. This bit, when set, forces panic mode while decoding. For test purposes only. |

## VID_QMW Quantization Matrix Data

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x76 | QMW[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x76
Type: Write only
Reset value: Undefined
Synchronization: None

***Description***

This address is used to load the quantization coefficients in the order in which they appear in the bit stream, i.e. zig-zag order. Bit VID_HDS.QMI defines whether an INTRA or INTER matrix is written.
For example, to write a new INTRA table, write VID_HDS.QMI = 1, and then write 64 weights to VID_QMW.

## VID_REV Device revision

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x78 | REV[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x78
Type: Read only

***Description***

This register gives the video cut number. For STi5518 cut A0, the value is 0x10. For subsequent cuts, the value is given in application note 7152120, STi5518 How to move from cutA0.

## VID_RFC Reconstructed Chroma Frame Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x5A | Reserved | | | | RFC[14:8] | | | |
| 0x5B | RFC[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x5A and 0x5B
Type: Read/write

Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the reconstructed (decoded) chroma frame picture buffer, defined in units of 256 bytes.

## VID_RFP Reconstructed Frame Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x0E | Reserved | | | | RFP[14:8] | | | |
| 0x0F | RFP[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x0E and 0x0F
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register holds the start address of the reconstructed (decoded) luma frame picture buffer, defined in units of 256 bytes.

## VID_SCDCOUNT Bit Buffer Output Counter

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | SCDcount [23:16] | | | | | | | |
| 2nd cycle | SCDcount [15:8] | | | | | | | |
| 3rd cycle | SCDcount [7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x68
Type: Serial read only
Reset value: 0

***Description***

These three registers are accessed serially. This register holds the number of 16-bit words output from the bit buffer into the Start Code Detector.

The byte pointer is reset by a hardware reset, a global soft reset or a video reset.

## VID_SCN Pan/Scan Vertical Vector

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x87 | | | SCN[5:0] | | | | | |

Address: *VideoBaseAddress* + 0x87
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register holds the vertical pan/scan vector in units of macroblock rows.

## VID_SPB Sub-picture Buffer Begin

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x50 | | | | | | SPB[10:8] | | |
| 0x51 | SPB[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x50 and 0x51
Type: Read/write
Reset value: 0
Synchronization: none

***Description***

This register holds the start address of the sub-picture circular buffer and is programmed in units of 64 bytes. The buffer should be aligned on a 1 Kbyte boundary.

## VID_SPE Sub-picture Buffer End

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x52 | Reserved | | | | SPE[11:8] | | | |
| 0x53 | SPE[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x52 and 0x53
Type: Read/write
Reset value: 0
Synchronization: none

***Description***

This register holds the stop address of the sub-picture circular buffer and is programmed in units of 64 bytes. The buffer should be aligned on a 1 Kbyte boundary.

## VID_SPREAD SubPicture Read Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | | | | | SPR[19:16] | | | |
| 2nd cycle | SPR[15:8] | | | | | | | |
| 3rd cycle | SPR[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x4E
Type: Serial read/write
Reset value: 0
Synchronization: VSYNC

***Description***

These three registers are accessed serially. The byte pointer is reset by a hardware reset. This is the absolute address in the memory. This register is used when the software needs to set the read address of the sub-picture decoder and is programmed in units of 64-bit words. This value is taken into account after a VSYNC.

## VID_SPWRITE SubPicture Write Pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st cycle | | | | | SPW[19:16] | | | |
| 2nd cycle | SPW[15:8] | | | | | | | |
| 3rd cycle | SPW[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x4F
Type: Serial read/write
Reset value: 0
Synchronization: None

***Description***

These three registers are accessed serially. The byte pointer is reset by a hardware reset. This is the absolute address in the memory. This register is used when the software needs to set the write address of the sub-picture decoder and is programmed in units of 64-bit words.

It is recommended to stop loading sub-picture data to the sub-picture decoder FIFO before changing the value of this register.

## VID_SRA Audio Soft Reset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x35 | | | | | | | | SRA |

Address: *VideoBaseAddress* + 0x35
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

In order to generate an audio soft reset (bit buffer), this bit must be kept set for a duration of at least 54 SDRAM clock cycles (540 ns with a 100 MHz primary clock).

## VID_SRV Video Soft Reset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x39 | | | | | | | | SRV |

Address: *VideoBaseAddress* + 0x39
Type: Read/write
Reset value: 0
Synchronization: none

***Description***

In order to generate a video soft reset, this bit must be kept set for a duration of at least 54 SDRAM clock cycles (540 ns with a 100 MHz primary clock).

## VID_STA Status

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3B | CWR | ERR | SWR | TR_OK | ABF | SFF | AFF | ABE |
| 0x64 | PDE | SER | BMI | HFF | PNC | ERC | DEI | PII |
| 0x65 | PSD | VST | VSB | BBE | BBF | HFE | BFF | SCH |

Address: *VideoBaseAddress* + 0x3B (VID_STA1), 0x64 (VID_STA2) and 0x65 (VID_STA3)
Type: Read only
Reset value: Undefined

This register contains a set of bits which represent the status of the decoder at any instant. Any change from 0 to 1 of any of these bits sets the corresponding bit of the **VID_ITS** register, and can thus potentially cause an interrupt.

The status vector is sampled internally at the start of the read cycle accessing the most significant byte of **VID_STA** (address 0x3B). VST, VSB and PSD are pulses and are unlikely ever to be read as a 1.

The status bits are described in table. The reset column shows the state after clocks have been enabled.

| Field | Description |
|---|---|
| SCH | Start Code Hit.<br>This bit is set whenever the first 16-bit word available in the header FIFO contains one of the start codes recognized. While data is being read from the header FIFO, this bit can be tested to determine whether the next word contains a start code. |
| BFF | Video Compressed Data (bit stream) FIFO Full.<br>This bit is set when the video CD FIFO is full. This bit is equivalent to the signal $\overline{\text{CDREQ}}$. |
| HFE | Header FIFO Empty.<br>This bit is set when the header FIFO is empty. |
| BBF | Video bit buffer full.<br>This bit is set when the bit buffer level (= **VID_VBL**) is greater than or equal to the value loaded into the **VID_VBT** reg. |
| BBE | Video bit buffer empty.<br>This bit is set when the bit buffer contains no data. |
| VSB | VSYNC Bottom. This bit is set for a short time at the beginning of the bottom field, corresponding to the rising edge of the B/$\overline{\text{T}}$ signal. |
| VST | VSYNC Top. This bit is set for a short time at the beginning of the top field, corresponding to the falling edge of the B/$\overline{\text{T}}$ signal. |
| PSD | Pipeline Starting to Decode.<br>This bit is set for a short period at the instant the pipeline starts decoding a picture. |
| PII | Pipeline Idle.<br>This bit is set when the device is not in the course of decoding a picture, i.e. when the pipeline is inactive. It becomes low when the decoding of a picture starts, and high when picture decoding is complete. |
| DEI | Decoder Idle.<br>This bit is set when the device is ready to decode a picture, i.e. when the pipeline is inactive and it has found the next picture start code. It becomes low when the decoding of a picture starts, and high when picture decoding is complete and the next picture start code has been found by to pipeline. |
| ERC | Error Concealment.<br>This bit is set when an error is detected in the bit stream and the mechanism of error concealment is active. |

| Field | Description |
|---|---|
| PNC | Panic.<br>This bit is set when decoding is in late compare to display. |
| HFF | Header FIFO Full.<br>This bit is set when the header FIFO contains at least 66 bytes. |
| BMI | Block Move Idle.<br>This bit is set when a block move operation has terminated. It is automatically reset at the start of a block move. |
| SER | Severe Error or overflow error.<br>This bit is set when more than the programmed number of macroblocks (defined by DFS) have been decoded, either due to a data or a programming error. Decoding is halted automatically when this error condition is detected. This bit is reset by all three types of reset. |
| PDE | Picture Decoding Error or underflow error.<br>This bit is set when less than the programmed number of macroblocks (defined by DFS) have been decoded, either due to a data or a programming error. Decoding is halted automatically when this error condition is detected. This bit is reset by all three types of reset. |
| ABE | Audio bit-buffer empty.<br>This bit is set when the audio bit buffer contains no data. |
| AFF | Audio compressed data (bit stream) FIFO full.<br>This bit is set when the audio CD FIFO is full. |
| SFF | Sub-picture compressed data (bit stream) FIFO full.<br>This bit is set when the sub-picture CD FIFO is full. |
| AFB | Audio bit-buffer full |
| TR_OK | Indicates that the VLR read pointer has been loaded into the memory controller, and that the VLD is ready to decode the selected picture. |
| SWR | CD Write Ready: set when CD write FIFO has been reset and is ready |
| ERR | Inconsistency error in PES parser |
| CWR | Start Code Detector Write Ready: set when SCD FIFO has been reset and is ready |

## VID_STL SDC trick-mode launch

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x30 | | | | | | | INR | STL |

Address: *VideoBaseAddress* + 0x30
Type: R/W
Synchronization: Edge triggered

***Description***

| Bitfield | Description |
|---|---|
| STL | 0: Launch Start Code Detector FIFO reset procedure for trick-modes.<br>Automatically set to zero once reset procedure is completed |
| INR | Inhibit re-decode mode:<br>0: Automatically start header code search after DSYNC.<br>1: Header code search cannot be launched automatically (niether by the block itself nor by the standard process). |

## VID_STR Video strobe

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x44 | Reserved | | | | | | | STR |

Address: *VideoBaseAddress* + 0x44
Type: W

***Description***

| Bitfield | Description |
|---|---|
| STR | Write to the internal strobe selected by register VID_CFG_GCF. CDR (either audio, video or subpicture). If VID_CFG_GCF. CDR=0, no internal strobes are selected and VID_STR cannot be used. |

## VID_TIS Task Instruction

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x03 | | SKP[1:0] | | | OVW | FIS | RPT | EXE |

Address: *VideoBaseAddress* + 0x03
Type: Write only
Reset value: 0
Synchronization: VSYNC

***Description***

This register contains 6 bits of the decoding task instruction.

| Bitfield | Description |
|---|---|
| SKP | These bits are part of the instruction register, and are thus synchronized with VSYNC. They define the skipping of one or two pictures. If skipping is required, then these bits must be set up as part of the instruction for the picture which will be decoded immediately after the skipped pictures.<br>00: No skip (default)<br>01: Skip one picture, decode next<br>10: Skip two pictures, decode next<br>11: Skip one picture then stop |
| OVW | This bit must be set when the displayed picture and the reconstructed picture share the same buffer (i.e. **VID_DFP** = **VID_RFP**). It enables the overwrite mode which ensures that the reconstructed picture does not overwrite data which has not yet been displayed. |
| FIS | Force instruction: if this bit is set, the task described by this register will be launch immediately. This bit is reset after its action. Its effect is the same as the effect of a VSYNC. |
| RPT | When this bit is set, the task duration is two VSYNC periods. When the frame display rate is equal to the picture decoding rate, **RPT** will generally always be high. In case of a task launched by **VID_TIS.FIS**, **RPT** is not taken in account. |
| EXE | When this bit is not set, no decoding or skipping task is executed for one or two VSYNC periods, depending on the state of **VID_TIS.RPT**. If set, the next task (decoding or skipping) is executed. **EXE** is internally cleared when the task starts its execution, i.e. setting this bit activates only one execution. |

## VID_TP_CD CD pointer load address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xCA | Not used | | | | | | | CD pointer address[16] |
| 0xCB | CD pointer address[15:8] | | | | | | | |
| 0xCC | CD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xCA, 0xCB, 0xCC
Type: R/W

***Description***

This register holds the CD pointer address during Trick Modes mode, defined in units of 512 bits.

## VID_TP_CD_RD CD pointer address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x83 | Not used | | | | CD pointer address[19:16] | | | |
| 0x84 | CD pointer address[15:8] | | | | | | | |
| 0x85 | CD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x85, 0x84, 0x83
Type: Read Only

***Description***

Holds the current CD pointer address, defined in units of 64 bits.

## VID_TP_CDLIMIT CD write limit address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xE0 | Not used | | | | | | | CD write limit address[16] |
| 0xE1 | CD write limit address[15:8] | | | | | | | |
| 0xE2 | CD write limit address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xE2, 0xE1, 0xE0
Type: R/W

***Description***

Contains the CD write limit value; the CD cannot write beyond this limit. The address is not latched and is a multiple of the CAS cycle length. In trick modes, a special bit-buffer management system is applied where the CD write flow is as follows:

CD pointer address = CD write limit = request CD disable.

For VLD read flow:

VLD pointer address = CD write limit => VLD request disable if FMUFLAGS/VSB is reset
VLD pointer address = CD pointer address => VLD request disable if FMUFLAGS/VSB is set

For SCD read flow:

VSCD pointer address = CD write limit => SD request disable

*Note CD write limit must be in the same format as the address used during comparisons.*

## VID_TP_SCD SCD pointer load address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xC0 | Not used | | | | | | | SCD pointer address[16] |
| 0xC1 | SCD pointer address[15:8] | | | | | | | |
| 0xC2 | SCD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xC0, 0xC1, 0xC2
Type: R/W

***Description***

Holds the address to load into the SCD pointer during Trick Modes. The address is defined in units of 512 bits.

## VID_TP_SCD_CURRENT SCD pointer current address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xCD | Not used | | | | SCD pointer address[19:16] | | | |
| 0xCE | SCD pointer address[15:8] | | | | | | | |
| 0xCF | SCD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xCD, 0xCE, 0xCF
Type: Read Only

***Description***

Holds the current SCD address, defined in units of 64 bits

## VID_TP_SCD_RD SCD pointer VLD load address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x81 | SCD pointer address for VLD[15:8] | | | | | | | |
| 0x82 | SCD pointer address for VLD[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x81, 0x82
Type: Read Only

***Description***

Holds the address to be loaded into the VLD pointer during trick modes. When the start-code detector finds a start-code, this register holds the bit buffer address, defined in units of 1Kbit.

## VID_TP_LDP Load pointer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x3F | | | | | | VSB | TM | LDP |

Address: *VideoBaseAddress* + 0x3F
Type: R/W
Reset value: Undefined

***Description***.

| Bitfield | Description |
|---|---|
| LDP | When this bit is set, the current start code detector pointer is stored in an internal register. When a PSC hit occurs, the current start code detector pointer has to be stored for further use by the VLD (if a B frame is to be processed on the fly). This bit has to be **set then reset,** before the PSD interrupt corresponding to the picture which has to be decoded on the fly (i.e. during the SCH interrupt) |
| TM | 0: Trick modes are not selected<br>1: Trick modes are selected |
| VSB | This bit stops a VLD read:<br>1: If VSB is set the VLD pointer address = CD pointer address<br>0: If VSB is reset the VLD pointer address = CD write limit address |

## VID_TP_VLD VLD pointer load address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xD4 | VLD pointer address[15:8] | | | | | | | |
| 0xD5 | VLD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xD4, 0xD5
Type: R/W

***Description***

This register holds the address to load to the VLD address pointer address during Trick Modes mode. It is defined in units of 1Kbits.

## VID_TP_VLD_RD VLD pointer current address

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xC3 | Not used | | | | VLD pointer address[19:16] | | | |
| 0xC4 | VLD pointer address[15:8] | | | | | | | |
| 0xC5 | VLD pointer address[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xC3, 0xC4, 0xC5
Type: Read Only

***Description***

Holds the current VLD pointer address, defined in units of 64 bits.

## VID_TRF Temporal Reference

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x56 | reserved | | | TML | DC2 | DTR | TRF[9:8] | |
| 0x57 | TRF[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x56 and 0x57
Type: Read/write
Reset value: 0
Synchronization: DSYNC

***Description***

This register contain extensions of the decoding task instruction. It is used only when decoding a B frame on the fly.

| Bitfield | Description |
|---|---|
| TRF | *Temporal reference.* This field holds the temporal reference of the current decoded B frame. On every B frame re-decode the VLD uses the temporal reference field **TRF** to re-synchronize on the previously decoded picture. **TRF** is used only when bit **DTR** is reset. |
| DTR | *Disable temporal reference comparison.* This bit is used only when decoding a B frame on the fly. On every B frame re-decode the VLD uses the temporal reference field **VID_TRF.TRF[9:0]** to re-synchronize on the previously decoded picture. When this bit is set this comparison mechanism is disabled. |
| DC2 | *Re-decode same B Frame twice.* When this bit is set it signals the VLD that the next picture will have to be decoded twice. This bit is used when decoding a B Frame on the fly. It has to be set by the user **before** the first PSD interrupt corresponding to a B frame and reset on the according PSD interrupt. |
| TML | *Trick_mode_launch*:<br>This signal is auto-re-initialized, that is it returns automatically to zero (it generates only a pulse). |

## VID_TST Test register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| set to zero | | | | | | | RBUSA7 |

Address: *VideoBaseAddress* + 0x77
Type: R/W WITH CARE
Reset value: 0
Synchronization: None

Description

| Bitfield | Description |
|---|---|
| RBUSA7 | This register is for test purposes only and should not be used otherwise.<br>1: the test registers can be written to<br>0: the test registers can not be written to |

## VID_VBG Start of Video Bit Buffer

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x14 | Reserved | VBG[14:8] | | | | | | |
| 0x15 | VBG[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x14 and 0x15
Type: Read/write
Reset value: 0
Synchronization: None

***Description***

The register holds the starting address of the video bit buffer, defined in units of 2 Kbits. If the video bit buffer starts at address 0, then this register does not need to be set up, since its reset state is 0. When all the registers corresponding to the settings of the bit buffer is done (VID_VBG, VID_VBS), an video soft reset must be done for values to be taken in account. In other words it must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

## VID_VBL Video Bit Buffer Level

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x16 | Reserved | VBL[14:8] | | | | | | |
| 0x17 | VBL[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x16 and 0x17
Type: Read only
Reset value: 0

***Description***

This register holds the current level of occupation of the video bit buffer, defined in units of 2 Kbits. It can be read at any time for the monitoring of the video bit buffer level. When VID_VBL is greater than or equal to the value held in the VID_VBT register, the status bit VID_STA.BBF (video bit buffer full) becomes set. When VID_VBL is zero, the status bit VID_STA.BBE (video bit buffer empty) becomes set.

## VID_VBS Video Bit Buffer Stop

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x18 | Reserved | VBS[14:8] | | | | | | |
| 0x19 | VBS[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x18 and 0x19
Type: Read/write

Reset value: 0
Synchronization: None

***Description***

This register holds the address of the top of the video bit buffer, defined in units of 2 Kbits. The space allocated to the video bit buffer starts at the address defined by the VID_VBG register, or, by default, 0. The end address of the video bit buffer is (in units of 64 bits words): (32 x VID_VBS) + 31

**VID_VBS** must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

## VID_VBT Video Bit Buffer Threshold

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x1A | Reserved | VBT[14:8] | | | | | | |
| 0x1B | VBT[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x1A and 0x1B
Type: Read/write
Reset value: 0
Synchronization: Edge triggered

***Description***

This register holds the level of occupancy of the video bit buffer, in units of 2 Kbits, which when reached causes the status bit VID_STA.BBF to become set, i.e. if VID_VBL ≥ VID_VBT, then VID_STA.BBF is set.

If the bit CFG_CCF.PBO is set, then transfer of data from the video CD FIFO to the bit buffer is prevented if the bit buffer level is at or above the level defined in the VID_VBT register. If VID_VBT is set to a value equal to the top of the bit buffer, then this automatic mechanism will ensure that overflow never occurs.

## VID_VFCMODE Configure chrominance of block-row

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xEA | HORIZDN | INTP | ZOOMOUT[37:36] | | OFFODD[35:32] | | | |
| 0xEB | OFFODD[31:24] | | | | | | | |
| 0xEC | OFFODD[23] | OFFEVEN[22:16] | | | | | | |
| 0xED | OFFEVEN[15:10] | | | | | | INCREMENT[9:8] | |
| 0xEE | INCREMENT[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xEA, 0xEB, 0xEC, 0xED, 0xEE
Type: R/W
Reset value: 0
Synchronization: VSYNC

***Description***

| Bitfield | Description |
|---|---|
| INCREMENT | The INCREMENT bit is used for downsampling up to zoom-out x2, and in conjunction with ZOOMOUT for fractional zoom-out between x2-x3, or x3-x4, for example x2.1, x2.2... etc.<br>Define the vertical ratio of the zoom, and given by the formula: $increment = \left(\frac{framestoresize}{desiredsize} \times 512\right) - 1$<br>Where *framestoresize* is the number of lines in the framestore, and *desiredsize* is the number of lines in the display region.<br>Ex, to display a 525 line framestore onto a 625 display region: $increment = \left(\frac{525}{625} \times 512\right) - 1 = 429$ |
| OFFEVEN | Vertical scans for even fields. Adjusts and superposes the luma and chroma filtering. The offset is a variable distance from the first line. |
| OFFODD | Vertical scans for odd fields. Adjusts and superposes the luma and chroma filtering. The offset is a variable distance from the first line. |
| ZOOMOUT | The INCREMENT bit is used for downsampling up to zoom-out x2, for zoom-out x3 and x4, ZOOMOUT must be used in as below. For zoom-out between x2-x3, or x3-x4, for example x2.1, x2.2... etc. ZOOMOUT must be used in conjunction with INCREMENT.<br>00: zoom-out up to x2<br>01: zoom-out by 2<br>10: zoom-out by 3<br>11: zoom-out by 4 |
| INTP | Interpolation field forces line repeat instead of integration (if 0, the top-line value is reproduced). |
| HORIZDN | Horizontal downsample by 2; neighboring samples are averaged.<br>NOTE: That no other post processing filter can be used at the same time as this function. The lines in the macroblock buffer RAM are read out as they are, however, the line offset part of start_offset is still used. |

## VID_VFLMODE Configure luminance of block-row

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xEF | HORIZDN | INTP | ZOOMOUT[37:36] | | OFFODD[35:32] | | | |
| 0xF0 | OFFODD[31:24] | | | | | | | |
| 0xF1 | OFFODD[23] | OFFEVEN[22:16] | | | | | | |
| 0xF2 | OFFEVEN[15:10] | | | | | | INCREMENT[9:8] | |
| 0xF3 | INCREMENT[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0xEF, 0xF0, 0xF1, 0xF2, 0xF3
Type: R/W
Reset value: 0
Synchronization: VSYNC

***Description***

| Bitfield | Description |
|---|---|
| INCREMENT | The INCREMENT bit is used for downsampling up to zoom-out x2, and in conjunction with ZOOMOUT for fractional zoom-out between x2-x3, or x3-x4, for example x2.1, x2.2... etc.<br>Define the vertical ratio of the zoom, and given by the formula: $increment = \left(\frac{framestoresize}{desiredsize} \times 512\right) - 1$<br>Where *framestoresize* is the number of lines in the framestore, and *desiredsize* is the number of lines in the display region.<br>Ex, to display a 525 line framestore onto a 625 display region: $increment = \left(\frac{525}{625} \times 512\right) - 1 = 429$ |
| OFFEVEN | Vertical scans for even fields. Adjusts and superposes the luma and chroma filtering. The offset is a variable distance from the first line. |
| OFFODD | Vertical scans for odd fields. Adjusts and superposes the luma and chroma filtering. The offset is a variable distance from the first line. |
| ZOOMOUT | The INCREMENT bit is used for downsampling up to zoom-out x2, for zoom-out x3 and x4, ZOOMOUT must be used in as below. For zoom-out between x2-x3, or x3-x4, for example x2.1, x2.2... etc. ZOOMOUT must be used in conjunction with INCREMENT.<br>00: zoom-out up to x2<br>01: zoom-out by 2<br>10: zoom-out by 3<br>11: zoom-out by 4 |
| INTP | Interpolation field forces line repeat instead of integration (if 0, the top-line value is reproduced). |
| HORIZDN | Horizontal downsample by 2; neighboring samples are averaged.<br>NOTE: That no other post processing filter can be used at the same time as this function. The lines in the macroblock buffer RAM are read out as they are, however, the line offset part of start_offset is still used. |

## VID_XDO Display X Offset

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0x70 | | | | | | | XDO[9:8] | |
| 0x71 | XDO[7:0] | | | | | | | |

Address: *VideoBaseAddress* + 0x70 to 0x71
Type: Read/write
Reset value: 0
Synchronization: VSYNC

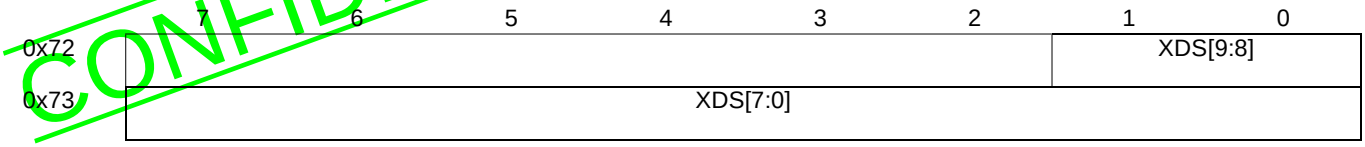
***Description***

**VID_XDO** is set up with a number defining the beginning of the left-hand border of the display. This offset is measured from the active (first) edge of HSYNC and is specified in units of PIXCLK cycles.

The horizontal offset, *XDO*', is given by: *XDO*' = **VID_XDO** + 4

The offset, *XDO*' cannot be less than 153 video decoder clock cycles.

## VID_XDS Display X End



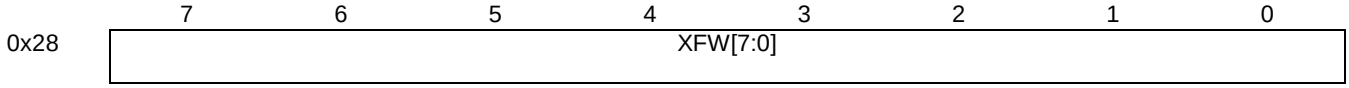

Address: *VideoBaseAddress* + 0x72 to 0x73
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set with a number defining the right-hand boundary of the picture display window, expressed in units of PIXCLK cycles. The value of **VID_XDS** must be equal to **VID_XDO** plus the width of the display window.

The actual offset, *XDS*', is given by: *XDS*' = **VID_XDS** + 4

## VID_XFW Displayed Frame Width

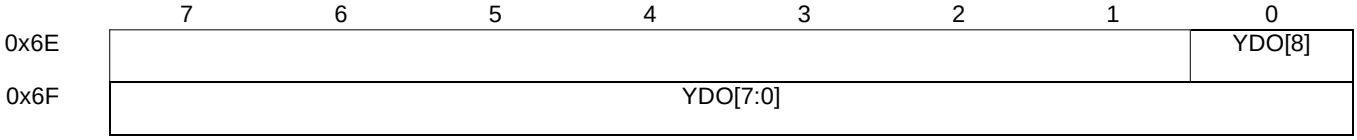



Address: *VideoBaseAddress* + 0x28
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set with a value equal to the width in macroblocks of the displayed picture. This is derived from the *horizontal_size* value transmitted in the sequence header.

## VID_YDO Display Y Offset



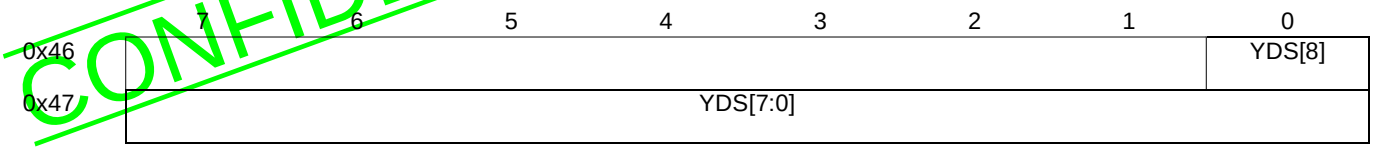

Address: *VideoBaseAddress* + 0x6E to 0x6F
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set up with a number defining the first line of the picture display, i.e. the top edge of the display window. Lines are counted from the active (first) edge of VSYNC.

In an interlaced display, the same value of **VID_YDO** must be used for both fields.

## VID_YDS Display Y End




Address: *VideoBaseAddress* + 0x46 to 0x47
Type: Read/write
Reset value: 0
Synchronization: VSYNC

***Description***

This register is set up with a number defining the bottom line of the display window. The value of YDS must be equal to:

**VID_YDO** + number of lines in picture counted in one field.

In an interlaced display, the same value of **VID_YDS** must be used for both fields.

# Index of registers