Viterbi and Reed–Solomon FEC Decoder

1. Description

Digital communication channels are inherently noisy, making transmission error control essential for reliable communication at low transmit power.

The TEMIC TSS902E is a single–chip Forward Error Correction decoder; it conforms to the MPEG–II transport layer protocol specified by ISO/IEC standard and FEC requirements of Digital Video Broadcasting (DVB) specification; its typical applications are DVB satellites, regenerative multi–media transmission satellites and military communications.

The TEMIC TSS902E capabilities rely on Viterbi and Reed–Solomon decoding algorithms to achieve extremely low bit–error rate (BER) on the transmitted data. Allowing discontinuous data blocks transmission, the TSS902E burst mode feature is unique.

The component is made of the following blocks:

- The inner decoder which performs the first level error detection and correction.
  This unit is made of a depuncturing block, a Viterbi decoder (k=7) and a synchronization/clock controller.
- The convolutional deinterleaver, l=12 bytes for RS (204, 188, T=8) configuration.
- The outer decoder performs the second level error protection, using a Reed Solomon (255, 239) error correcting process.
- The descrambler for energy dispersal removal.
- A micro–processor interface to setup the device and monitor the testability functions.

While monitoring the inner Viterbi decoder BER output, the phase and the depuncturing pattern are tuned until the Viterbi decoder proper alignment is found.

The Viterbi decoder output feeds the deinterleaver and Reed–Solomon decoder synchronization module. Once the synchronization words have been found, the deinterleaver, the outer Reed–Solomon decoder and the descrambler are properly aligned.

Each functional block may be by–passed, giving more flexibility to a system designer.

2. Features

2.1. General

- Compliant with ETS 300 421 for DVB, DVB–S.
- Compliant with ISO/IEC–CD 13818–1 MPEG–II transport layer protocol.
- Input code rate frequency up to 10 MBits/sec at 5V.
- On–chip Bit Error Rate monitoring.
- SEU immunity better than 30 MeV/mg/cm^2
- Total dose better than 50 Krad (Si).
- Supply voltage 3 to 5V.
- Power consumption 1W at 5V / 10MHz external clock frequency (code rate 7/8).
- 0.6 µm drawn CMOS, 3 metal layers.
- 132–pin MQFP.

2.2. Viterbi Decoder

- Selectable code rates $^1/_2$, $^2/_3$, $^3/_4$, $^5/_6$ and $^7/_8$ or automatic acquisition mode.
- Hard decision or 3–Bit soft–decision decoder inputs.
- Constraint length k = 7.
- E_b/N_0 for BER 2.10^{-4} (code rate $^1/_2$) 3.5 dB.

2.3. Synchronization controller

- Automatic synchronization capabilities for QPSK or BPSK.
- Responds to inverted synchronization byte.
- Programmable synchronization byte.

2.4. Convolutional deinterleaver

- Error protected frame length n = 204.
- Interleave depth I = 12.

2.5. Reed Solomon Decoder

- Supported programmable shortened code length K = 34 to 239, T = 8.
- Correction capability up to T = 8 bytes.

2.6. Descrambler (Energy Dispersal)

- Polynomial generator $q(x) = X^{15} + X^{14} + 1$.
- MPEG–II inverted synchronization byte.

3. Term definitions

Table 1: VITERBI

TERM	NAME	DEFINITION	RANGE
k	Encoding constraint length	Number of input bits contributing to the convolutional code.	7
	Traceback depth	Length of path through the trellis over which the Viterbi decoder computes a decoded bit value likelihood.	
	Puncturing	Transmission rate increasing process. Done by erasing some specific code bits before data transmission.	
	Code rate	Convolutional code input to output bits ratio.	2/3, 3/4, 5/6, 7/8

Table 2: CONVOLUTIONAL DEINTERLEAVER

TERM	NAME	DEFINITION	RANGE
I	Interleaving depth	Interleaved stream separation.	12

Table 3: REED–SOLOMON

TERM	NAME	DEFINITION	RANGE
K	Message length	Number of user data symbols in one message block.	$34 \leqslant K \leqslant 239$ bytes
R	Check symbols	Symbols appended to the user data to detect errors.	16 bytes
N	Codeword length	Sum of message and check symbols. $N = K + R$	$50 \leqslant N \leqslant 255$ bytes
T	Error corrections	Maximum number of error corrections performed by the RS decoder.	8

4. Functional description

This section describes the TSS902E device architecture and its constituents functionality.

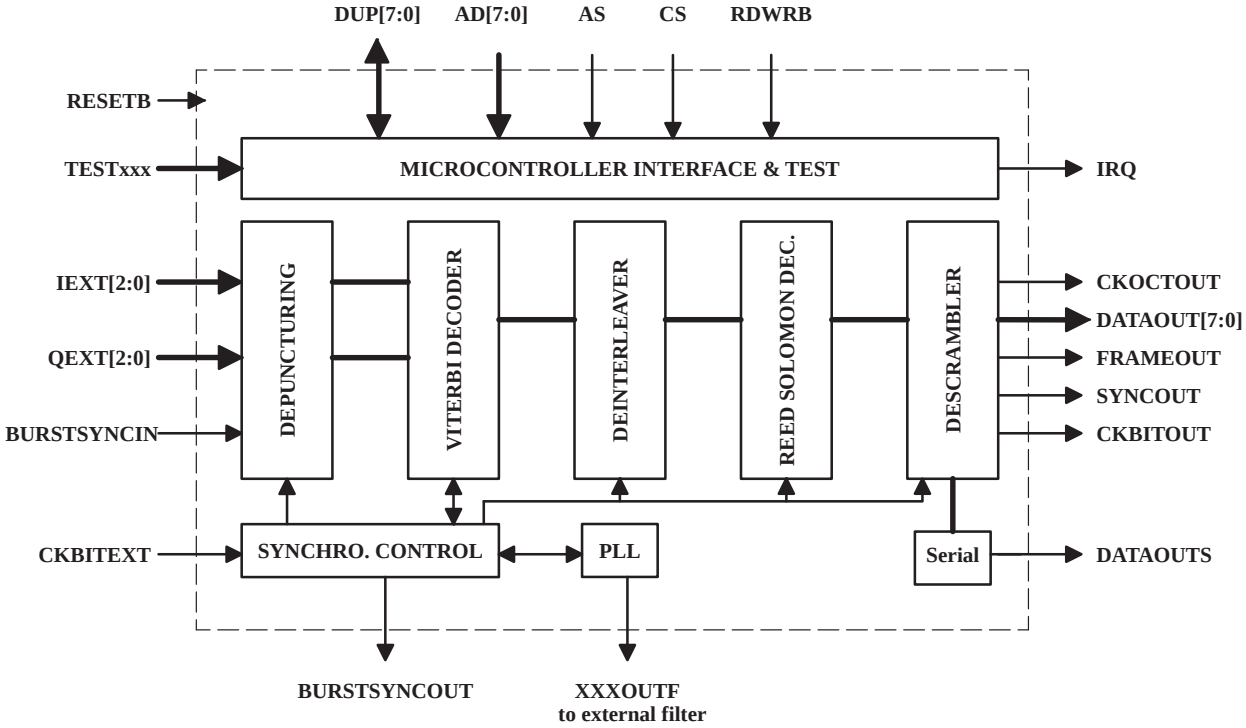


Figure 1: TSS902E block diagram

4.1. Synchronization

The *Synchronization Control* block is made of two parts called *Synchro–Bit* and *Synchro–Frame* which algorithms are described hereafter.

4.1.1. The Synchro–Bit module

The *Synchro–Bit* module performs the Depuncturing block and the Viterbi decoder synchronization by monitoring the Bit Error Rate (*BER*). The Viterbi *BER* calculation is done by the device during the averaging period whereas the monitoring period can be defined by the user in order to perform off–line statistical *BER* calculations.

The Depuncturing module adds missing bits according to the code rate. Since the code rate may be unknown, the Depuncturing block may initially use a bad rate if the automatic mode has been selected (see *RxVitControl* register – A/M and FRE bits). Furthermore, the Depuncturing process may be uncorrectly synchronized although using the right code rate. Both conditions will lead to an unsatisfactory *BER* value.

The Viterbi *BER* value is considered to be acceptable when it remains under the RxVitThreshold register value. This register is actually made of several registers, each associated to one code rate value (1/2, 2/3, 3/4, 5/6 and 7/8). If throughout the averaging period the *BER* value stays below the threshold, the *Synchro–Bit* block sets the *SBF* (Synchro Bit Found) flag within the *RxVitStatus* register, locks the PLL and allows the *Synchro–Frame* module to start the synchro word search.

If, on the other hand, in automatic mode, the Viterbi decoder *BER* value exceeds its programmed threshold during the averaging period, the device will enter in *sync–bit* search mode trying to resolve at first the input data phase ambiguity within the current code rate; if the *BER* value still is too high, the device will assume that the depuncturing process

was not performed on the right input bits (the bits assumed to be missing were not on the right position) and the depuncturing state–machine will enter next state; finally, if the *BER* value still remains unacceptable, the code rate value will be changed and the device will restart the overall search process.

Warning : if the programmed threshold value is too high, the bit synchronization may be found on a wrong position. In automatic mode, all the threshold registers must be properly programmed according to the expected noise and code rate. In semi–automatic mode, the user can just define the right value in the known code rate threshold register. In manual mode, the maximum threshold value can be programmed in the code rate threshold register; the use of FRE bit allows to find the bit synchronization for this code rate. (cf *RxVitControl* register usage)

A loss of synchronization restarts a search of frame synchro. The Data processing of blocks placed after sync–frame block is restarted as well.

The *Synchro–Bit* module can be bypassed by bit *SYBE* of the *RxRSVitAct* register. When set, the *SYBE* bit enables the synchronization between De–Puncturing and Viterbi blocks; when cleared, the *Synchro–Bit* module still maintains the SBF generation in order to allow other blocks to work; in this case the De–Puncturing rate is fixed (Rate = 1/2, no phase ambiguity).

4.1.2. The Synchro–Frame module

Starting on successful *Synchro–Bit* process completion, the *Synchro_Frame* process waits for the frame synchronization word (47h) or the transport packet one (B8h) by default. The sync words search duration is limited by a timeout value defined through the *RxSyncWCtl* register *TOV* bits and *RxVitSyncCompWord* register *SCW* bits. An unsuccessful sync words search produces an interrupt and sets the *RxVitRSInt0* register *TiO* flag.

Whenever either sync word or bitwise inverted sync word is found, a counter is incremented and compared to the number of consecutive sync words at the expected position required to get synchronized (*RxSyncWCtl* register *SY2/SY1/SY0* bits). The *Synchro–Frame* process remains in the sync word waiting loop until the expected number of consecutive sync words is reached, then the *Synchro–Frame* block sets the *RxVitStatus* register *LCK* bit (*Synchro–Frame* locked).

The default sync word value is 47h but it may be changed to any value through the *RxSyncWord* register. The sync condition may take into account the bitwise inverted sync word search or not (*RxVitSyncCompWord* register *CENA* bit). The bitwise inverted sync word occurs every 8 frames by default but this value may be changed from 1 to 15 (*RxVitSyncCompWord* register *SCW3* to *SCW0* bits).

A maximum number of allowed mismatching bits can be defined when expecting the sync word (*RxSyncWctl* register *MSY* bits).

The *Synchro–Frame* process may restart under the following external conditions:
- a general reset is applied to the device.
- the *Synchro_Frame* module receives a restart signal from the *Synchro–Bit* block.
- the bit synchronization is lost.
- the Reed–Solomon decoder is out of sync (*RxDeSyncWCtl* register *DRSM* bit); this condition may be masked.
- the descrambler module is out of sync (*RxDeSyncWCtl* register *DSCM* bit); this condition may be masked.

The *Synchro–Frame* module will lose sync again after lock, when the sync word is not found at the expected position for a number of consecutive frames which is specified in *RxVitDeSyncWCtl* register *DSY[2:0]*. The maximum number of mismatching bits for a missed sync word is programmable (*RxVitDeSyncWCtl* register *MDS[1:0]* bits).

A loss of synchronization restarts a search of frame synchro. The Data processing of blocks placed after sync–frame block is restarted as well.

The *Synchro–Frame* module can be bypassed by bit *SYE* of the *RxRSVitAct* register.

4.2. Depuncturing

The puncturing process is a way to change the code rate. Within a certain interval of code bits, some specific bits are simply not transmitted, i.e. erased. The De–Puncturing block must add these erased bits according to the transmission rate. This rate can be programmed in the *RxVitRate* register or automatically looked for by the Synchro_bit module.

The ETS 300 421 (Digital brodcasting systems for television, sound and data services; Framing structure, channel cading and modulation for 11/12 GHz satellite services) defines punctured codes as in Table 1.

Table 4: Punctured code definition

Original Coding			**Code rates**									
K	**G1 (X)**	**G2 (Y)**	**1/2**		**2/3**		**3/4**		**5/6**		**7/8**	
			P	dfree	P	dfree	P	dfree	P	dfree	P	dfree
			X:1		X:10		X:101		X:10101		X:1000101	
			Y:1		Y:11		Y:110		Y:11010		Y:1111010	
7	171_{oct}	133_{oct}		10		6		5		4		3
			I=X_1		I=$X_1Y_2Y_3$		I=X_1Y_2		I=$X_1Y_2Y_4$		I=$X_1Y_2Y_4Y_6$	
			Q=Y_1		Q=$Y_1X_3Y_4$		Q=Y_1X_3		Q=$Y_1X_3X_5$		Q=$Y_1Y_3X_5Y_7$	
			DePunctured Add Bit									
			No Bit		X_2X_4		X_2Y_3		$X_2Y_3X_4Y_5$		$X_2X_3X_4Y_5X_6Y_7$	
Note			1 = transmitted bit 0 = no transmitted bit									

For example, running with the 7/8 rate, the IEXT and QEXT inputs receive the following sequence:

IEXT = a1, b2, b4, b6

QEXT = b1, b3, a5, a7

At a coding rate of 7/8, the device has to generate 6 bitsat the proper position when it reads 8 bits in order to obtain 14 bits (i.e. 7 symbols). Depuncturing the above sequence gives:

IEXT = a1, **1**, **1**, **1**, a5, **1,** a7

QEXT = b1, b2, b3, b4, **1**, b6, **1**

If the code rate is known, the Viterbi decoder module may be set to work in manual mode (see the *RxVitControl* register, *A/M* and *FRE* bits); in that case, the code rate must be defined through the *RxVitRate* register.
If the code rate is unknown, the *Synchro–Bit* module can retrieve it by cycling through the 1/2, 2/3, 3/4, 5/6 and 7/8 code rates, looking for a valid pattern; the *Synchro–Bit* module may also consider a possible phase ambiguity or a depuncturing process error (the bits assumed to be missing were not at the right position). The depuncturing block can work in BPSK or QPSK modes (I and Q information being serial on IEXT input in BPSK mode, therefore the internal bit clock frequency is half the external one).

The depuncturing module can be bypassed by bit *DPE* of the *RxRSVitAct* register.

4.3. Viterbi decoder

The Viterbi decoder reads data from the depuncturing module I and Q outputs and produces decoded bit information.

The Viterbi block I[2:0] and Q[2:0] inputs may be configured to work in *soft–decision* or *hard–decision* mode by programming the *RxVitControl* register *H/S* bit.

In *soft–decision* mode, the 3–bit input words are compared to seven thresholds.

In *hard–decision* mode, only the IEXT and QEXT most significant bits are used; other bits are ignored.

The Viterbi decoder can be bypassed by the *RxRSVitAct* register *VE* bit.

The Viterbi decoder latency is 290 *Ckbitrate* clock periods. *Ckbitrate* is internally generated; its period is:

$$T = \frac{Text}{2 \times Rate}$$

Where *Text* is the *CKBITEXT* clock period. For instance, if *Text* value is 100ns and *Rate* is 3/4, *T* value is 67ns. Latency = 19.5us

The Viterbi module can be bypassed by bit *VE* of the *RxRSVitAct* register.

4.4. De–interleaver

This part describes the process of convolutional de–interleaver as it is defined into ETS 300 421 .
This device allows the error burst at the output of the inner decoder to be randomized on a byte basis in order to improve the burst error correction capability of the outer decoder (Reed–Solomon).
The principle of the interleaver is the following one : the interleaver is composed of I = 12 branches, cyclically connected to the input byte–stream by the input switch. Each branch is a First–In, Fist–Out (FIFO) shift register, with depth M*j cells (where M = 17 = N/I, N = 204 = error protected frame length, I = 12 = interleaving depth, j = branch index).
For synchronization purpose, the sync bytes (47h) and the inverted sync bytes (B8h) will always be routed in the branch ”0” of the interleaver (corresponding to a null delay)
The de–interleaver process is similar except that the branch indexes are inverted (i.e. j = 0 corresponds to the largest delay). The de–interleaver synchronization can be carried out by routing the first recognized sync byte in the ”0” branch.

The de–interleaver can be bypassed by bit *DIE* of the *RxRSVitAct* register.

4.5. Reed–Solomon decoder

The inner code has to be an 8–error–correcting Reed–Solomon code over GF(2.8) (Galois Field). The natural length of this code is 255 symbols and since it can correct up to 8 errors, the code has 16 check symbols. This RS8 code can be shortened by any number of symbols resulting in a code length N, 50<N<255 (including check symbols). The value of N will be selected by the user with the *RxRsFrameL* register. If there are more than 8 errors in a block, the error pattern is uncorrectable and a flag error is generated. (cf Descrambler block)

The device can also count the number of bit errors (when frames are successfully corrected) and store this value in *RxRsErrLsb* and *RxRsErrMsb* registers. It counts and stores the number of uncorrectable frames (more than 8 byte errors) in *RxRsUncorrect* register. These registers are updated at the end of a programmable monitoring period *RxRsFrameNb*.

If there are more than n consecutive uncorrectable frames (n specified in *RxRsDeSyncFN* register), a desynchronization frame signal is transmitted to Synchro–Frame module to start a new synchronization search. This signal can be masked by setting to one the *DRSM* bit of *RxRsDeSyncWCtl* register.

The Reed–Solomon can be bypassed by bit *RSE* of the *RxRSVitAct* register.

4.6. Descrambler

The energy dispersal is defined into ETS 300 421. The polynomial for the Pseudo Random Binary Sequence (PRBS) generator is : $1 + X^{14} + X^{15}$ and the loading sequence is "100101010000000".
A MPEG–2 frame is 188 bytes long and the first byte is called MPEG–2 sync byte (47h) (MSB first). A transport packet is a group of 8 packets (8 x 188) where the first MPEG–2 sync byte is bit–wise inverted 47h to B8h. The randomizer will start with the first bit (MSB) of the first byte following the inverted sync. If the inverted sync word is not found and in order to aid other synchronization functions during MPEG–2, the PRBS will continue but its output will be disabled, leaving the sync bytes unrandomized.
It has to be noticed that the Synchronization byte is provived by the *RxSyncWord* register. It is thus configurable.
This block is also in charge of setting the "transport_error_indicator". In case the Write Error Bit (WER) is set and non–correctable errors happen in Reed–Solomon decoder (FlagError is set), the MSB of the first byte following the sync byte is forced to "1" after descrambling. If WER is reset, data are not affected.
When the CENA bit is set (*RxSyncCompWord* register), the descrambler works as just described. When it is reset, the PRBS generator is reloaded at each frame. The synchronization byte is not inverted in this case.
If an error occured on the MPEG–2 sync (B8h), 8 frames would be lost. To avoid this trouble a proper "Mainframe" synchronization is created using the SCW value (Synchro Complement Word). As soon as one inverted sync is found, a counter is set using the SCW value written in *RxSyncCompWord* register (8 by default for MPEG–2). Then, it generates this synchronization. So, only one frame is actually lost instead of eight.

The descrambler can be bypassed by bit *EDE* of the *RxRSVitAct* register.

4.7. The burst mode

In order to indicate to the circuit that the received data is of Burst type, the user must set the *RxVitControl* register *BURSTM* bit.

4.7.1. De–Interleaving burst mode constraints

A burst packet may keep characteristics such as puncturing, Viterbi and RS coding. The de-interleaving process of packets will be significant only if the packets are made of n frames of 204 bytes. Likewise, the de-interleaving process will correctly achieve only if the burst packets and the inter-burst (duration between 2 packets) are multiple of

$$\frac{(I-1)\times I}{2} \times 17\ bytes \quad (\text{for } I = 12)$$

4.7.2. De-puncturing burst mode constraints

During a Burst Mode transmission, there are several possible operating modes for the depuncturing block:

4.7.3. Freewheel Mode

On each cycle, the received data, whether it is valid data or not, is depunctured according to the rate.
The *Freewheel* mode is activated by clearing the *RxVitRate* register *PBFrz* and *PBRst* bits.

4.7.4. Freeze Mode

Whenever the circuit receives invalid data, the state–machine, which handles the insertion of bits according to the rate, stops depuncturing and restarts when incoming data become valid.

The *Freeze* mode may be activated by setting the *RxVitRate* register *PBFrz* bit.

4.7.5. Restart Mode

Whenever the circuit receives invalid data, the state–machine, stops depuncturing, goes to the first step, and restarts when incoming data become valid. The external data input is then supposed to be a *a1/b1* pair.
This mode requires an inter-burst period greater or equal to 15 clock cycles
The *Restart* mode may be activated by setting the *RxVitRate* register *PBRst* bit.

4.7.6. Viterbi decoding burst mode constraints

Patent pending.

4.7.7. Synchro–Bit burst mode constraints

The Synchro-bit block can also be used during Burst Mode operation. However the errors issued from the Viterbi decoder will be taken into account only when they are recognized as being part of valid data. Since the Synchro-Frame process will start only after the Synchro-Bit is locked, the synchronization packets must be long enough to remove any ambiguity.

4.7.8. Synchro–Frame burst mode constraints

In addition to the frame synchronization process, the Synchro–Frame block organizes the Viterbi data output as bytes. When operating in Burst Mode, this serial to parallel conversion is the only useful operation, since the Reed–Solomon decoder uses the bytes and Synchro–Frame signal *SYNCOUT* to perform its decoding.
The Synchro–Frame block constraint is to have packets being multiple of 8 bits and an inter-burst of at least 7 bits.
The π ambiguity removal at the synchro-frame level can be done by software. The synchro frame signal *SYNCOUT* delimits the first generated byte and is regenerated at the beginning of each new frame. The frame length is defined in the *RxRsFrameL* register, thus allowing to determine the beginning of each new frame.

4.8. Global synchronization

Before starting data process the PLL must be locked. According to the input clock bit frequency, it can be necessary to program the *RxVitFreqIn* register.

The received data are first processed by the de–puncturing block, no matter their value.
Then, the depunctured data go to Viterbi decoder. Viterbi counts the errors found in this data stream. The Synchro–Bit block is in charge of collecting the error number over a programmable averaging period (*RxVitSearch*), and comparing it to a programmable threshold (*RxVitThresholdxx*) with respect to the puncturing rate used for the de–puncturing block.

This rate is programmable *(RxVitRate)* in manual or semi–automatic mode, or searched automatically in automatic mode. If the error number is greater than the threshold, there is no synchronization, and Synchro–Bit goes one state ahead in its state machine. There are several states for each puncturing rate since Synchro–Bit is also in charge of resolving $\pi/2$ phase shift ambiguities as well as the ambiguity whether the de–puncturing process starts in the right phase. An interrupt is generated at the end of the averaging period (*SBTE)* if the error number is greater than the threshold.
When the synchro bit is found (interrupt *SBF*), which happens when the error number is lower than the threshold, the search for a synchronization word (either 47h or its complemented value B8h for DVBS standard) starts in the Synchro–Frame block. However, the frame synchronization may not happen on the first synchro word found, since 47h or B8h may simply be a useful data in the stream. *RxSyncWctl* register is used to program the number of consecutive synchro words encountered at the right place (every n bytes, where n is the frame length programmed in *RxRsFrameL*) needed to confirm and synchronize the decoder, allowing 0, 1 or 2 mismatching bits in the ”confirming” words *(RxSyncWCtl)*. In the same way, the desynchronization conditions can be programmed in *RxDeSyncWCtl* register: the number of consecutive non–synchro words found every n bytes to desynchronize the decoder, and the number of error bits allowed. Moreover, *RxSyncWctl* register also contains a programmable time out value representing the maximum duration of the synchro word research after Viterbi synchro bit is found. If so, an interrupt is generated: *TiO*. It has to be noticed that too many errors in Reed–Solomon decoder and in the descrambler can have the synchro frame desynchronize. These respective interrupts (*RSSL* and *DDSC*) may be masked in *RxDeSyncWCtl* register. The interrupt indicating a synchro frame lock is *LCK*. The one for a synchro frame loss is *SYL*. All interrupts can be read from registers *RxVitRSInt0* and *RxVitRSInt1*.
The de–interleaver block starts operating at reception of a synchro word coming from Synchro–Frame.
The Reed–Solomon decoder is able to correct up to 8 bytes. If Reed–Solomon detects more than 8 error bytes, it does not correct the error frame and it raises an error flag (interrupt *RSER*). After a certain number of consecutive error frames (programmable in the *RxRSDeSyncFN* register), Reed–Solomon is said to be desynchronized (interrupt RSSL), and may ask the Synchro–Frame to start another synchronization search.
Due to Reed–Solomon encoding, each transmitted frame contains 16 check symbols in addition to the useful data. The descrambler is in charge of separating the useful data from RS check symbols. The BurstSyncOut signal is used to indicate the valid data. Since the descrambler simulates the process of energy dispersal within transport packets, it looks for B8h synchro words to start operating. However, it outputs only 47h synchro words at the beginning of each frame. If it does not find a B8h synchro word at the expected place, it sets the interrupt *DDSC* to warn the Synchro Frame block that it is desynchronized.

5. Register Mapping and Description

5.1. Register Mapping

The following control and status registers purpose is to synchronize the Viterbi decoder onto the right phase and puncturing rate as well as to monitor and tune the data stream quality and parameters.

Table 5: Address Mapping

Address	Name		Access	Reset Signal	Reset Value
0x01	RxVitControl	Viterbi control	R/W	$\overline{\text{RESETB}}$	0x00
0x02 to 0x06	RxVitThreshold	Viterbi threshold	R/W	$\overline{\text{RESETB}}$	0x00
0x07	RxVitSearch	Viterbi averaging period	R/W	$\overline{\text{RESETB}}$	0x00
0x08 / 0x09	RxVitMonitor	Viterbi monitoring period	R/W	$\overline{\text{RESETB}}$	0x00
0x0A / 0x0B	RxVitError	Viterbi error rate	R	$\overline{\text{RESETB}}$,, Restart & write to RxVit-Monitor	0x00
0x0C / 0x0D	RxVitErrorSync	Viterbi synchro error	R	$\overline{\text{RESETB}}$ & Restart	0x00
0x0E	RxVitRate	Depuncturing rate	R/W	$\overline{\text{RESETB}}$	0x01
0x0F	RxVitFreqIn	Viterbi input frequency	R/W	$\overline{\text{RESETB}}$	0x00
0x10	RxSyncWord	Synchro word	R/W	$\overline{\text{RESETB}}$	0x47
0x11	RxSyncWCtl	Synchro word control	R/W	$\overline{\text{RESETB}}$	0x01
0x12	RxDeSyncWCtl	Out of sync word control	R/W	$\overline{\text{RESETB}}$	0x01
0x13	RxSyncStat	Synchro / Out of sync status	R	$\overline{\text{RESETB}}$ & Restart	0x00
0x14	RxVitSyncCompWord	Synchro complemented word	R/W	$\overline{\text{RESETB}}$	0x88
0x20 / 0x21	RxVitRSInt	Interrupt status	R	$\overline{\text{RESETB}}$ & Read RxVitRSInt	0x00
0x22 / 0x23	RxVitRSMask	Interrupt mask	R/W	$\overline{\text{RESETB}}$	0xFF / 0x3F
0x24	RxVitStatus	Viterbi status	R	$\overline{\text{RESETB}}$ & Restart	0x00
0x30	RxRSFrameL	Reed Solomon frame length	R/W	$\overline{\text{RESETB}}$	0xCC
0x31 / 0x32	RxRSError	Reed Solomon bit error count	R	$\overline{\text{RESETB}}$ & Restart	0x00
0x33	RxRSUncorrect	RS uncorrected frame count	R	$\overline{\text{RESETB}}$ & Read RxRSUncorrect	0x00
0x34	RxRSFrameNb	Reed Solomon max frame count	R/W	$\overline{\text{RESETB}}$	0x00
0x35	RxRSDeSyncFN	Reed Solomon out of sync frame	R/W	$\overline{\text{RESETB}}$	0x1F
0x40	RxRSVitAct	Viterbi & RS activity control	R/W	$\overline{\text{RESETB}}$	0xFF

5.2. Register Description

5.2.1. Viterbi Control

RxVitControl Register address = **0x01** – Reset Value = **0x00**

7	6	5	4	3	2	1	0
BURSTM	**A/M**	**FRE**	**RST**	**SEC**	**S/O**	**PSK**	**H/S**

H/S – Hard/Soft mode:

0 = Soft decision (3 bits detection); 1 = Hard decision using IEXT and QEXT MSB only.

PSK – Modulation mode:

0 = QPSK (two inputs: IEXT and QEXT); 1 = BPSK (IEXT single input)

SEC–S/O:

00 = Offset Sign magnitude ;
01 = Offset Binary
10 = Two's Complement
11 = unused

RST – Software RESET:

0 = Inactive; 1 = Allows the device to restart the synchronization phase

A/M & FRE – Automatic / Manual and Freeze flags:

A/M = 0 & FRE = 0: Automatic mode. Successive enabled code rates are tried with all possible phases, until the system is locked and the block synchro is found; this is the default setup (reset mode).

A/M = 0 & FRE = 1: The code rate is frozen; if no synchro is found, the phase is swept but not the code rate.

A/M = 1: Manual mode. The system is forced to a code rate (that you have previously defined), ignoring the time–out register and the BER value. In this mode, each FRE–bit 0 to 1 transition increments the phase, allowing full control of the synchronization by an external microcontroller, keeping the tuning giving the lowest BER.

BURSTM – Burst Mode:

0 = Inactive (default); 1 = Active.

5.2.2. Viterbi Threshold

RxVitThreshold12 RA[7:0] Register address = **0x02** – Reset value = **0x00**
RxVitThreshold23 RB[7:0] Register address = **0x03** – Reset value = **0x00**
RxVitThreshold34 RC[7:0] Register address = **0x04** – Reset value = **0x00**
RxVitThreshold56 RD[7:0] Register address = **0x05** – Reset value = **0x00**
RxVitThreshold78 RE[7:0] Register address = **0x06** – Reset value = **0x00**

7	6	5	4	3	2	1	0
RA7	**RA6**	**RA5**	**RA4**	**RA3**	**RA2**	**RA1**	**RA0**
RB7	**RB6**	**RB5**	**RB4**	**RB3**	**RB2**	**RB1**	**RB0**
RC7	**RC6**	**RC5**	**RC4**	**RC3**	**RC2**	**RC1**	**RC0**
RD7	**RD6**	**RD5**	**RD4**	**RD3**	**RD2**	**RD1**	**RD0**
RE7	**RE6**	**RE5**	**RE4**	**RE3**	**RE2**	**RE1**	**RE0**

RA [7:0], RB [7:0], RC [7:0], RD [7:0], RE [7:0] – Bit Error Rate Threshold:
Within each register, bits 7 to 0 represent a BER threshold (the average number of errors occuring during an averaging period). The current BER is compared to this programmable threshold. If it is greater, another phase (or another rate) is tried until the proper rate is found (Viterbi is synchronized).

Maximum threshold values:

SN [5:0] = 1xxxxx => Threshold = RX[7:0] & ”0000000”. .128 –> 32640
SN [5:0] = 01xxxx => Threshold = RX[7:0] & ”000000” . . . 64 –> 16320
SN [5:0] = 001xxx => Threshold = RX[7:0] & ”00000” . . . . 32 –> 8160
SN [5:0] = 0001xx => Threshold = RX[7:0] & ”0000” . . . . . 16 –> 4080
SN [5:0] = 00001x => Threshold = RX[7:0] & ”000” . . . . . 8 –> 2040
SN [5:0] = 00000x => Threshold = RX[7:0] & ”00” . . . . . 4 –> 1020

Note:
SN[5:0] word purpose is described in the RxVitSearch register paragraph.
RX[7:0] & ”00” means that two zeros are appended to the register value, giving for this example a minimum and a maximum threshold values of 4 (R[7:0] = 0x01) and 1020 (R[7:0] = 0xFF). RX[7:0] stands for RA[7:0] to RE[7:0].

5.2.3. Viterbi Search

RxVitSearch Register address = **0x07** – Reset Value = **0x00**

7	6	5	4	3	2	1	0
0	**0**	**SN5**	**SN4**	**SN3**	**SN2**	**SN1**	**SN0**

SN [5:0] – Averaging Period:

The averaging period register defines the number of incoming bits **used by the device** to calculate the BER.

0x3F –> 64K bits required to calculate the BER
0x3E –> 63K bits required to calculate the BER
...
...
0x02 –> 3K bits required to calculate the BER
0x01 –> 2K bits required to calculate the BER
0x00 –> 1K bits required to calculate the BER

5.2.4. Viterbi Monitoring

RxVitMonitor0 Register address = **0x08** – Reset value = **0x00**

7	6	5	4	3	2	1	0
Nb7	**Nb6**	**Nb5**	**Nb4**	**Nb3**	**Nb2**	**Nb1**	**Nb0**

RxVitMonitor1 Register address = **0x09** – Reset value = **0x00**

0	**0**	**Nb13**	**Nb12**	**Nb11**	**Nb10**	**Nb9**	**Nb8**

Nb [13:0] – Monitoring Period:

The monitoring period register defines the number of incoming bits required for BER statistical calculation **done by the user**. This value is a multiple of 1Kbits. Writing into this register starts a new monitoring period. At the end of this period, an interrupt may be generated (if enabled by the RxVitRSMask1 register MEMP bit).

0x3FFF –> 16384K bits required to calculate the BER
0x3FFE –> 16383K bits required to calculate the BER
...
...
0x0002 –> 3K bits required to calculate the BER
0x0001 –> 2K bits required to calculate the BER
0x0000 –> 1K bits required to calculate the BER

5.2.5. Viterbi Error Rate (Read Only)

RxVitErrorMsb Register address = **0x0A** – Reset value = **0x00**

7	6	5	4	3	2	1	0
ERV15	ERV14	ERV13	ERV12	ERV11	ERV10	ERV9	ERV8

RxVitErrorLsb Register address = **0x0B** – Reset value = **0x00**

ERV7	ERV6	ERV5	ERV4	ERV3	ERV2	ERV1	ERV0

ERV [15:0] – Number of Error Bits:

This 16–bits counter gives the number of error bits found along a monitoring period. It may be used for an off–line BER calculation done by the user.

Whenever a monitoring period completes, this counter is frozen; it is reset to zero when a monitoring period starts.

5.2.6. Viterbi Error Synchro (Read Only)

RxVitErrorSync0 Register address = **0x0C** – Reset value = **0x00**

7	6	5	4	3	2	1	0
ERS7	ERS6	ERS5	ERS4	ERS3	ERS2	ERS1	ERS0

RxVitErrorSync1 Register address = **0x0D** – Reset value = **0x00**

0	0	ERS13	ERS12	ERS11	ERS10	ERS9	ERS8

ERS [13:0] – Number of Incoming Bits:

This 14–bits counter gives the number of **Kbits** received during a monitoring period before the BER value reaches the BER threshold (leading to a synchronization loss). It may be used to locate the context in which the synchronization loss occurred.

In conjunction with the RxVitError register, it may also be used to tune the BER threshold.

5.2.7. Code Rate

RxVitRate Register address = **0x0E** – Reset value = **0x00**

7	6	5	4	3	2	1	0
PBFrz	PBRst	0	E4	E3	E2	E1	E0

E [4:0] – Enable code Rate:

This field sets the Viterbi decoder code rate when working in **manual mode** or **semi–automatic mode**.

0x01 –> code rate = 1/2
0x02 –> code rate = 2/3
0x04 –> code rate = 3/4
0x08 –> code rate = 5/6
0x10 –> code rate = 7/8

Only one bit should be set to 1. Any other E[4:0] contents leads to a 1/2 code rate selection.

PBRst : Burst mode depuncturing Restart bit:

0 = The incoming IEXT/QEXT data stream is continuously depunctured even if BURSTSYNCIN input pin is held to zero.
1 = Stops the depuncturing process whenever BURSTSYNCIN input pin is set to zero (no valid burst data) and restarts on a ai/bj pair supposed to be the a1/b1 one. See the burst mode functional description.

PBFrz : Burst mode dePuncturing Freeze bit:

0 = The incoming IEXT/QEXT data stream is continuously depunctured even if BURSTSYNCIN input pin is held to zero.
1 = Stops the depuncturing process whenever BURSTSYNCIN input pin is set to zero (no valid burst data) and restarts where it stopped. See the burst mode functional description.

5.2.8. Viterbi Input Frequency

RxVitFreqIn Register address = **0x0F** – Reset value = **0x00**

7	6	5	4	3	2	1	0
VCOC	**FREQ6**	**FREQ5**	**FREQ4**	**FREQ3**	**FREQ2**	**FREQ1**	**FREQ0**

FREQ [6:0] – PLL External DividerValue in BPSK mode:

$$FREQ\ [6:0] = Int\ [\frac{33.15}{FInBit}] - 1$$

FREQ [6:0] – PLL External DividerValue in QPSK mode:

$$FREQ\ [6:0] = Int\ [\frac{16.57}{FInBit}] - 1$$

FInBit : Input frequency on ***CKBITEXT*** input pin:

$$1MHz < FInBit < 10MHz$$

In Automatic Mode the default value should be: VCOC = 0

VCOC – VCO Control:
This bit determines the PLL VCO to be used
0 = VCO2 is selected (output frequency range: 40 MHz to 130 MHz)
1 = VCO1 is selected (output frequency range: 130 MHz to 230 MHz)

5.2.9. Synchro Word

RxSyncWord Register address = **0x10** – Reset value = **0x47**

7	6	5	4	3	2	1	0
SYW7	**SYW6**	**SYW5**	**SYW4**	**SYW3**	**SYW2**	**SYW1**	**SYW0**

SYW [7:0] – Synchronization Word:
This register sets the Synchronization Word recognized by the Viterbi module during synchro and lock phases. The Viterbi synchronization module looks for SYW [7:0] and its complemented value every SCWth block (see RxVitSyncCompWord register).

5.2.10. Synchro Word Control

RxSyncWCtl Register address = **0x11** – Reset value = **0x01**

7	6	5	4	3	2	1	0
TOV1	TOV0	0	MSY1	MSY0	SY2	SY1	SY0

SY [2:0] – Number of Consecutive Synchronization Words:
SY [2:0] is the number of consecutive Synchronization Words to be found to synchronize the decoder.

MSY [1:0] – Maximum Number of Mismatching Bits:
MSY [1:0] is the maximum number of mismatching bits allowed to declare a match when comparing data stream to the expected Synchronization Word. (0, 1 or 2 mismatching bits maximum. If MSY = 11, the number of allowed mismatching bits is set to 0).

TOV [1:0] – Time Out Value:
The Time Out Value programs the maximum duration of the Synchronization Word search after Viterbi synchro bit is found. If no Synchronization Word is found within this duration, an interrupt is generated and the external system must change parameters like the BER Threshold or the MSY value.

11 = Time out (bit period) = 8 x SCW [3:0] (see RxVitSyncCompWord register for SCW[3:0] definition).
10 = Time out (bit period) = 4 x SCW [3:0]
01 = Time out (bit period) = 3 x SCW [3:0]
00 = Time out (bit period) = 2 x SCW [3:0]

5.2.11. Loss of sync Word Control

RxDeSyncWCtl Register address = **0x12** – Reset value = **0x01**

7	6	5	4	3	2	1	0
DRSM	DSCM	0	MDS1	MDS0	DSY2	DSY1	DSY0

DSY [2:0] – Number of consecutive lost synchronization words:
DSY [2:0] is the number of consecutive not expected synchronization words required to go out of sync.

MDS [1:0] – Maximum number of mismatching bits:
MDS [1:0] is the maximum number of mismatching bits allowed to declare a match when comparing data stream to the expected synchronization word (0, 1 or 2 bits maximum. If MDS = 11 the number of mismatching bits is set to 0).

DSCM – Descrambler out of sync mask:
0 = Descrambler out of sync unmasked; 1 = Descrambler out of sync masked

DRSM – Reed–Solomon out of sync mask:
0 = Reed –Solomon out of sync unmasked; 1 = Reed –Solomon out of sync masked

5.2.12. Synchro / Out of sync Status (Read Only)

RxSyncStat Register address = **0x13** – Reset value = **0x00**

7	6	5	4	3	2	1	0
DSW3	DSW2	DSW1	DSW0	DSV3	DSV2	DSV1	DSV0

DSV [3:0] – Number of out of sync bits:
DSV [3:0] is a counter incremented each time the Viterbi looses sync after the first lock.

DSW [3:0] – Number of out of sync words:
DSW [3:0] is a counter incremented each time the Synchronization Word mechanism looses sync after the first lock.

5.2.13. Complemented SyncWord

RxVitSyncCompWord Register address = **0x14** – Reset value = **0x88**

7	6	5	4	3	2	1	0
CENA	**SAPI**	**0**	**0**	**SCW3**	**SCW2**	**SCW1**	**SCW0**

SCW [3:0] – Complemented Synchro Word Position:
The Complemented Synchro Word periodically appears after SCW frames.

SAPI – Correct Pi Ambiguity in Burst Mode:
0 = No correction; 1 = π ambiguity correction enabled in burst mode

CENA – Complemented Sync Word Reseach Enable:
0 = Complemented sync word search disabled; 1 = Complemented sync word search enabled.

5.2.14. Interrupt Status (Read Only)

RxVitRSInt0 Register address = **0x20** – Reset value = **0x00**

7	6	5	4	3	2	1	0
ODFC	**TiO**	**SBF**	**LCK**	**SYL**	**RSSL**	**RSER**	**TEBRS**

RxVitRSInt1 Register address = **0x21** – Reset value = **0x00**

7	6	5	4	3	2	1	0
0	**0**	**DDSC**	**ODBC**	**SBTE**	**EAP**	**MECO**	**EMP**

TEBRS – Reed–Solomon Monitoring Period Finished
RSER – Reed–Solomon Found More Than 9 Bytes in Error
RSSL – Reed–Solomon out of sync
SYL – Synchro Lost
LCK – Synchro Word Found
SBF – Synchro Bit Found
TiO – Time Out Period Without finding synchro word
ODFC – Overflow out of sync Frame Counter
EMP – Viterbi Monitoring Period Finished
MECO – Monitoring Errors Counter Overflow
EAP – Viterbi Averaging Period Finished
SBTE– Synchro Bit Threshold Error
ODBC – Synchro Bit Lost Counter Overflow
DDSC – Descrambler out of sync
The above bits give information about the interrupt generation event. A bit set to 1 shows the interrupt origin.

5.2.15. Interrupt Mask

RxVitRSMask0 Register address = **0x22** – Reset value = **0xFF**

7	6	5	4	3	2	1	0
MODFC	**MTO**	**MPRF**	**MLCK**	**MSYL**	**MRSSL**	**MRSER**	**MTEBRS**

RxVitRSMask1 Register address = **0x23** – Reset value = **0x3F**

7	6	5	4	3	2	1	0
0	**0**	**MDDS**	**MODBC**	**MSBTE**	**MEAP**	**MMECO**	**MEMP**

MTEBRS – Reed–Solomon Monitoring Period Finished
MRSER – Reed–Solomon Found More Than 9 Bytes in Error
MRSSL – Reed–Solomon out of Sync
MSYL – Synchro Lost
MLCK – Synchro Word Found
MPRF – Synchro Bit Found
MTO – Time Out Period Without finding Synchro Word
MODFC – Overflow out of sync Frame Counter
MEMP – Viterbi Monitoring Period Finished
MMECO – Monitoring Errors Counter Overflow
MEAP – Viterbi Averaging Period Finished
MSBTE – Synchro Bit Threshold Error
MODBC – Synchro Bit Lost Counter Overflow
MDDS – Descrambler out of sync
When set to 1, the above bits mask the corresponding interrupts.

5.2.16. Viterbi Status (Read Only)

RxVitStatus Register address = **0x24** – Reset value = **0x00**

7	6	5	4	3	2	1	0
PLU	**EMP**	**TiO**	**SBF**	**LCK**	**PR2**	**PR1**	**PR0**

PR [2:0] – Current Code Rate :
In automatic mode, PR [2:0] indicates the current code rate according to the following table:
100 = code rate 7/8
011 = code rate 5/6
010 = code rate 3/4
001 = code rate 2/3
000 = code rate 1/2

LCK – Synchro Word Found:
0 = Searching for Synchro Word; 1 = Synchro word found and locked**.**

SBF – Synchro Bit Found:
SBF indicates the code rate search status. This bit is irrelevant in manual mode.
0 = Searching for Synchro bit; 1 = Synchro bit found and locked**.**

TiO – Time Out Period Without SWF:
0 = (see conditions for TiO = 1)
1 = When Synchro Bit is found and no Synchro Word is found within the Time Out duration.

EMP – Viterbi End of Monitoring Period:

0 = Monitoring Period not finished.
1 = Monitoring Period finished. A write access to the RxVitMonitor registers resets this bit.

PLU – PLL Lock/Unlock:
0 = Pll unlocked; 1 = Pll locked**.**

5.2.17. Reed Solomon Frame Length

RxRSFrameL Register address = **0x30** – Reset value = **0xCC**

7	6	5	4	3	2	1	0
FL7	**FL6**	**FL5**	**FL4**	**FL3**	**FL2**	**FL1**	**FL0**

FL [7:0] – Frame Length:
The Reed Solomon decoder is based on the original RS (255, 239, T = 8) code, and supports the following shortening K = 34 to 239. This also includes the DVB specification of RS (204, 188, T = 8). The default value is 204 (0xCC), and might be overwritten by values from 50 (34 + 16) to 255 (239 + 16).

5.2.18. Reed–Solomon Bit Error Count (Read Only)

RxRSErrLsb Register address = **0x31** – Reset value = **0x00**

7	6	5	4	3	2	1	0
E7	**E6**	**E5**	**E4**	**E3**	**E2**	**E1**	**E0**

RxRSErrMsb Register address = **0x32** – Reset value = **0x00**

7	6	5	4	3	2	1	0
E15	**E14**	**E13**	**E12**	**E11**	**E10**	**E9**	**E8**

E [15:0] – RS Bit Error Count:
E [15:0] stores the value of the RS bit error count used to calculate the Reed–Solomon BER. If there are more than 8 bytes on error in a frame, the bit error count is not incremented.
If there are 8 bytes on error in a frame with 8 bits on error in each byte, the bit error count is not incremented; this is a bug that will be fixed in the next version.

5.2.19. Reed–Solomon Uncorrected Frames Count (Read Only)

RxRSUncorrect Register address = **0x33** – Reset value = **0x00**

7	6	5	4	3	2	1	0
UNF7	**UNF6**	**UNF5**	**UNF4**	**UNF3**	**UNF2**	**UNF1**	**UNF0**

UNF [7:0] – UnCorrected Frame Number:
UNF [7:0] is the RS Frame error counter used to calculate the Reed–Solomon BER. UNF error count is incremented each time there are more than 8 bytes on error in a frame.

5.2.20. Reed Solomon Maximum Frame Count

RxRSFrameNb Register address = **0x34** – Reset value = **0x00**

7	6	5	4	3	2	1	0
FN7	FN6	FN5	FN4	FN3	FN2	FN1	FN0

FN [7:0] – Maximum Frame Count:
FN [7:0] bits are the 8–Msb preset value of a 10–bit counter used to define the RS monitoring period duration. At the end of this time, the RxRSErr and RxRSUncorrect registers are updated and an interrupt (TEBRS) is generated. Writing into this register starts a new RS monitoring period.

5.2.21. Reed–Solomon out of Sync Frame

RxRSDeSyncFN Register address = **0x35** – Reset value = **0x1F**

7	6	5	4	3	2	1	0
WER	0	0	DRSY4	DRSY3	DRSY2	DRSY1	DRSY0

DRSY [4:0] – Reed–Solomon out of sync Frame Number:
DRSY[4:0] is the number of consecutive Reed–Solomon erroneous frames required to indicate sync loss.

WER – Write Error Bit:

0 = Data is not affected.

1 = If a non–correctable error occurs, the ”transport_error_indicator” (Msb of the first byte following the synchro byte) is forced to ’1’ after descrambling.

5.2.22. Viterbi and Reed–Solomon Activity Control

RxRSVitAct Register address = **0x40** – Reset value = **0xFF**

7	6	5	4	3	2	1	0
SYBE	**EDE**	**RSE**	**DIE**	**SYE**	**VE**	**DPE**	**PLLE**

PLLE – PLL Enable:
0 = PLL disabled, the clock rate equals external clock rate and the PLL input clock PLL is held low.
1 = PLL enabled, the clock rate is generated by the PLL.

DPE – DePuncturing Enable:
0 = Data is not affected; 1 = Input data stream is depunctured.

VE – Viterbi Enable:
0 = Data is not affected; 1 = Input data stream is corrected by the Viterbi decoder.

SYE – Synchro Enable:
0 = Parallel data come from external pin (IEXT, QEXT, TESTIERA, TESTQERA).
1 = Serial flow is converted to parallel data by Synchro.

DIE – De–Interleaver Enable:
0 = Data is not affected; 1 = Input data stream is deinterleaved.

RSE – Reed–Solomon Enable:
0 = Data is not affected; 1 = Errors are corrected.

EDE – Energy Dispersal removal Enable:
0 = Data is not affected; 1 = Output data from RS is descrambled.

SYBE – Synchro Bit Enable:
0 = The Synchro bit found is generated to allow other blocks to work (except Viterbi decoder).
1 = The Viterbi error processing is enabled.

6. Signals Descriptions and Specifications

6.1. Signals Descriptions

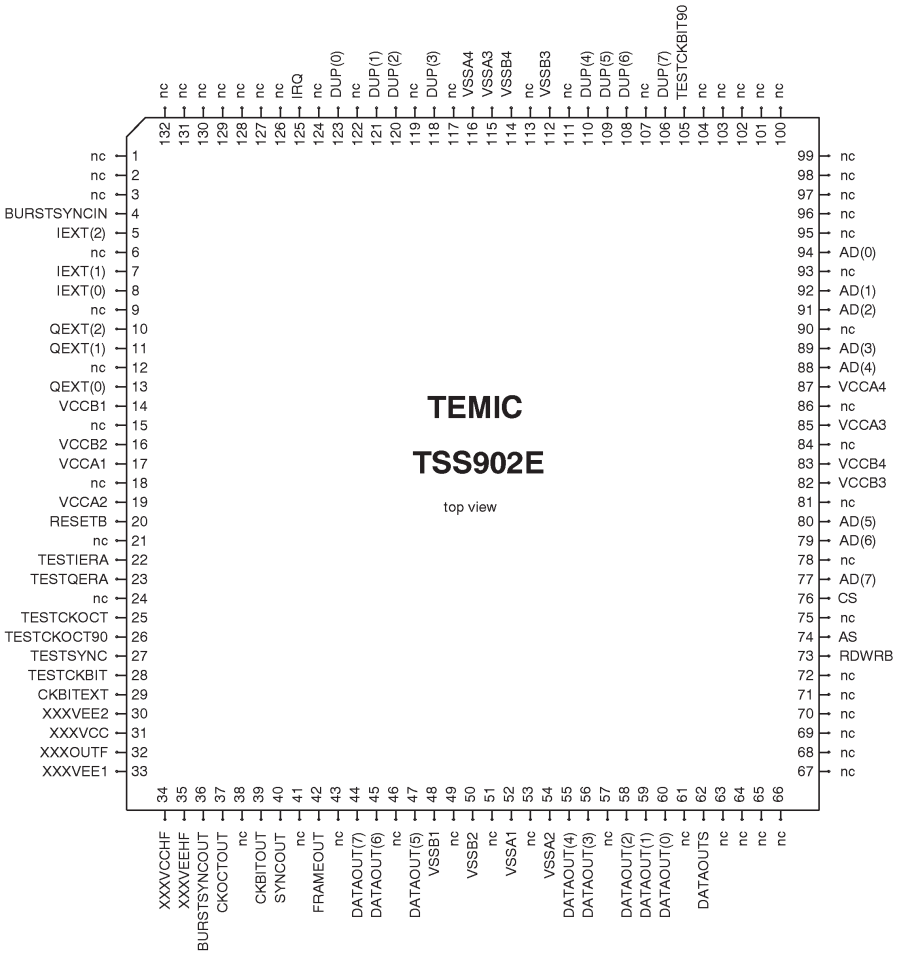
Name	Active State or Edge	Description
MICROPROCESSOR SIGNALS		
CS	Low	Chip select input.
AS	↓ edge	Address latch input.
RDWRB	Low / High	Read (high) / Write (low) signal.
AD[7:0]		Address bus.
DUP[7:0]		Bi–directional data bus.
RESETB	Low	Reset. When active, forces all blocks into a known state. Must be used as specified in the AC parameters section.
IRQ	Low	Interrupt output. May be used by the external system for sync monitoring purpose.
DEMODULATOR SIGNALS		
IEXT[2:0]		3–bit soft decision input. Use IEXT[2] for hard decision input.
QEXT[2:0]		3–bit soft decision input. Use QEXT[2] for hard decision input.
BURSTSYNCIN	High	Burst mode control pin. When high, indicates an incoming data burst. When low, indicates irrelevant data on IEXT and QEXT pins in burst mode.
PLL SIGNALS		
CKBITEXT	↑ edge	Symbol clock for IEXT and QEXT.
XXXOUTF		PLL external RC filter connection.
OUTPUT SIGNALS		
DATAOUT[7:0]		Data output bus. May be read on the CKOCTOUT output clock rising edge.
CKOCTOUT	↓ edge	Parallel data output sampling signal.
SYNCOUT	High	Active whenever a frame sync word is found.
FRAMEOUT	High	Active whenever a multi–frame sync word is found.
DATAOUTS		Serial data output. May be read on the CKBITOUT output clock rising edge.
CKBITOUT		Serial data output sampling signal.
BURSTSYNCOUT	High	Active whenever a multi–frame sync word is found; remains high for a frame length duration.
TEST SIGNALS		
TESTIERA		Depuncturing block I test input.
TESTQERA		Depuncturing block Q test input.
TESTCKOCT	↓ edge	Byte–format data test clock.
TESTCKOCT90	↓ edge	Byte–format data test clock (Pi/2 delayed from TESTCKOCT).
TESTSYNC	High	Indicates a sync word occurrence.
TESTCKBIT	↑ edge	In test mode, emulates the depunctured data clock.
TESTCKBIT90	↑ edge	In test mode, emulates the depunctured data clock (Pi/2 delayed from TESTCKBIT).

Name	Active State or Edge	Description
DEPUNCTURING SIGNALS		
IEXT[2:0]		3–bit soft decision input. Use IEXT[2] for hard decision input.
QEXT[2:0]		3–bit soft decision input. Use QEXT[2] for hard decision input.
BURSTSYNCIN	High	Burst mode control pin. When high, indicates an incoming data burst. When low, indicates irrelevant data on IEXT and QEXT pins in burst mode.
CKBITEXT	↑ edge	Symbol clock for IEXT and QEXT.
TESTCKBIT	↑ edge	In test mode without PLL, emulates the depunctured data clock.
DATAOUT[7:5]		represent Iout[2:0] with added bits according to the code rate
DATAOUT[4:2]		represent Qout[2:0] with added bits according to the code rate
DATAOUT[1]		when this output equal 1, Iout are added bits (EraseI)
DATAOUT[0]		when this output equal 1, Qout are added bits (EraseQ)
CKBITOUT		Serial data output sampling signal.
VITERBI SIGNALS		
IEXT[2:0]		3–bit soft decision input (with or without added bits according to the usage of Depuncturing block). Use IEXT[2] for hard decision input.
QEXT[2:0]		3–bit soft decision input (with or without adding bits according to the usage of Depuncturing block). Use QEXT[2] for hard decision input.
TESTIERA	High	indicates the I adding bits if the depuncturing block is not used
TESTQERA	High	indicates the Q adding bits if the depuncturing block is not used
BURSTSYNCIN	High	If the depuncturing block is not used, the activity on this signal indicates an incoming data. otherwise it indicates irrelevant data on IEXT and QEXT pins. With the Depuncturing block this signal value makes sense only if the Burst mode is programmed.
TESTCKBIT	↑ edge	Witout PLL, emulates the depunctured data clock.
TESTCKBIT90	↑ edge	Witout PLL, emulates the depunctured data clock (Pi/2 delayed from TESTCKBIT)
CKBITEXT	↑ edge	External clock bit needed if the depuncturing block is used. Its frequency depends on the usage of the depuncturing block.
DATAOUT[0]		Viterbi serial output.
CKBITOUT		Serial data output sampling signal.
SYNCHRO_BIT SIGNALS (without VITERBI and DEPUNCTURING blocks)		
BURSTSYNCIN	High	When high, indicates an incoming data.
TESTSYNC	High	When high, emulates a Viterbi error.
TESTCKBIT	↑ edge	External clock bit needed if the Pll is not used
CKBITEXT	↑ edge	External clock bit needed if the Pll is used
SYNCHRO_FRAME SIGNALS (without VITERBI, DEPUNCTURING and SYNCHRO_BIT blocks)		
IEXT[2]		Input serial data.
TESTCKBIT	↑ edge	External clock bit needed if the Pll is not used
CKBITEXT	↑ edge	External clock bit needed if the Pll is used
BURSTSYNCIN	High	When high, indicates a valid incoming data in Burst mode.
TESTSYNC	High	When high, indicates an incoming data in Burst mode. This signal must be set to one 6 bits before the real valid data and set to zero 6 bits before the burst end.
DATAOUT[7:0]		Parallel Data output bus. May be read on the CKOCTOUT output clock rising edge.
DATAOUTS		Serial data output. May be read on the CKBITOUT output clock rising edge.
SYNCOUT	High	Active whenever a frame sync word is found or during the first frame word in burst mode according to the frame length.
BURSTSYNCOUT	High	Active whenever a synchro–frame is locked or when DATAOUT is valid in burst mode.

Name	Active State or Edge	Description
CKOCTOUT		Parallel data output sampling signal.
CKBITOUT		Serial data output sampling signal.
CONVOLUTIONAL DEINTERLEAVER SIGNALS		
IEXT[2:0]		Input DeInterleaver parallel data[7:5], IEXT[2] => Input DeInterleaver serial data if Synchro_Frame is used.
QEXT[2:0]		Input DeInterleaver parallel data[4:2]
TESTIERA		Input DeInterleaver parallel data[1]
TESTQERA		Input DeInterleaver parallel data[0]
CKBITEXT	↑ edge	External clock bit needed if the Pll is not used
TESTCKBIT	↑ edge	External clock bit needed if the Pll is used
TESTCKOCT	↑ edge	External clock byte needed if Synchro_frame is not used
TESTSYNC	High	Emulates the input frame sync word if Synchro_frame is not used
DATAOUT[7:0]		Parallel Data output bus. May be read on the CKOCTOUT output clock rising edge.
DATAOUTS		Serial data output. May be read on the CKBITOUT output clock rising edge.
SYNCOUT	High	Active whenever a frame sync word is transmitted to the output
BURSTSYNCOUT	High	Active whenever a complete convolutional Interleaving sequence is processed and all output data are valid
CKOCTOUT		Parallel data output sampling signal.
CKBITOUT		Serial data output sampling signal.
REED–SOLOMON DECODER SIGNALS		
IEXT[2:0]		Input Reed–Solomon parallel data[7:5], IEXT[2] => Input Reed–Solomon serial data if Synchro_Frame is used.
QEXT[2:0]		Input Reed–Solomon parallel data[4:2]
TESTIERA		Input Reed–Solomon parallel data[1]
TESTQERA		Input Reed–Solomon parallel data[0]
CKBITEXT	↑ edge	External clock bit needed if the Pll is used and Synchro_Frame is used.
TESTCKBIT	↑ edge	External clock bit needed if the Pll is not used and Synchro_Frame is used.
TESTCKOCT	↑ edge	External clock byte needed if Synchro_frame is not used
TESTCKOCT90	↑ edge	External clock byte needed if Synchro_frame is not used (Pi/2 delayed from TESTCKOCT)
TESTSYNC	High	Emulates the input frame sync word if Synchro_frame is not used according to the Frame lenght.
BURSTSYNCIN	High	When high, indicates a valid incoming Frame.
DATAOUT[7:0]		Parallel Data output bus. May be read on the CKOCTOUT output clock rising edge.
DATAOUTS		Serial data output. May be read on the CKBITOUT output clock rising edge only if an input clock bit is used.
SYNCOUT		Active whenever a frame sync word is transmitted to the output
BURSTSYNCOUT		When high, indicates a valid output Frame.
CKOCTOUT		Parallel data output sampling signal.
CKBITOUT		Serial data output sampling signal.
DESCRAMBLER SIGNALS		
IEXT[2:0]		Input Descrambler parallel data[7:5], IEXT[2] => Input Descrambler serial data if Synchro_Frame is used.
QEXT[2:0]		Input Descrambler parallel data[4:2]
TESTIERA		Input Descrambler parallel data[1]
TESTQERA		Input Descrambler parallel data[0]

Name	Active State or Edge	Description
CKBITEXT	↑ edge	External clock bit needed if the Pll is used and Synchro_Frame is used.
TESTCKBIT	↑ edge	External clock bit needed if the Pll is not used and Synchro_Frame is used.
TESTCKOCT	↑ edge	External clock byte needed if Synchro_frame is not used
TESTSYNC	High	Emulates the input frame sync word if Synchro_frame is not used according to the Frame lenght.
DATAOUT[7:0]		Parallel Data output bus. May be read on the CKOCTOUT output clock rising edge.
DATAOUTS		Serial data output. May be read on the CKBITOUT output clock rising edge only if an input clock bit is used.
SYNCOUT	High	Active whenever a frame sync word is transmitted to the output
BURSTSYNCOUT	High	Active whenever a multi–frame sync word is found; remains high for a frame length duration without Reed–Solomon syndrome (Active during 188 byte) .
FRAMEOUT	High	Active whenever a multi–frame sync word is found.
CKOCTOUT		Parallel data output sampling signal.
CKBITOUT		Serial data output sampling signal.

6.2. Pinout Description



7. Programming and Operation Modes

This chapter describes how to operate the TSS902E from the programmer's point of view. Most of the TSS902E registers are made of several fields. In the following descriptions, the concerned registers and fields names will be mentioned as well as recommended values. You should read the Register Description section anyway since some registers particularities like unvalid registers values are not covered in this chapter.

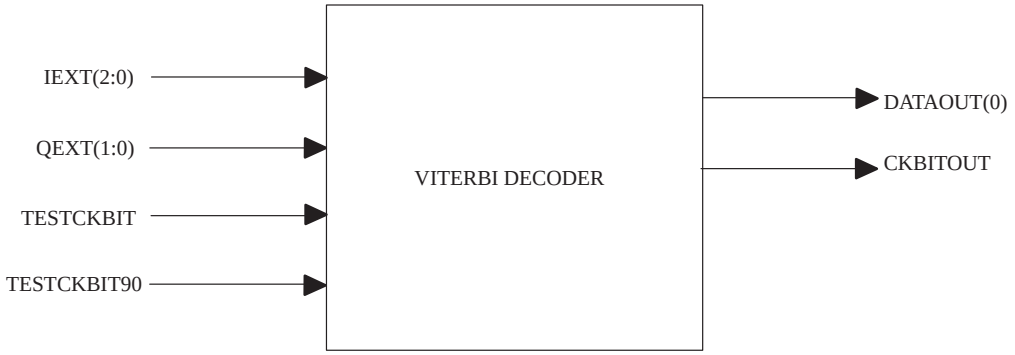
7.1. Device Activity Control

In order to give great flexibility to a system designer, each functional block may be by–passed. You will probably start the TSS902E setup by enabling the blocks you want to be activated. The following sections show how to configure the device in order to activate some blocks only, or the whole circuit.

7.1.1. Using the Viterbi Decoder Only

Working with the Viterbi decoder only requires three conditions:

- The rate is 1/2; i.e. there is no data puncturing.
- QPSK mode is used
  When running in BPSK mode, the I/Q data stream coming to the IEXT pins with the I1 – Q1 – I2 – Q2 – I3 – Q3 sequence is demultiplexed by the depuncturing block; consequently, using the BPSK mode requires the depuncturing block to be enabled.
- For the Viterbi decoder to run alone properly, you must be able to provide it with two stable clock signals.
  If you can not guarantee the clocks stability, you must use the depuncturing block which compensates the clocks jitter.

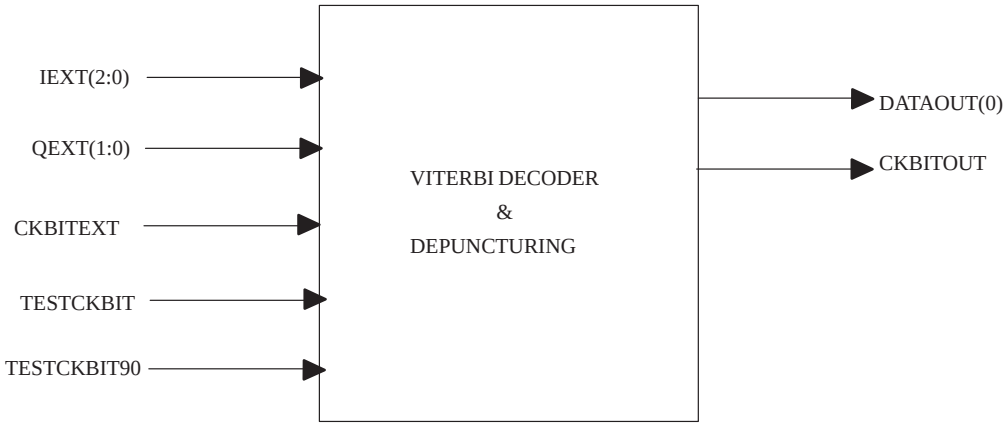


IEXT/QEXT data is clocked on the TESTCKBIT rising edge; there is a Pi/2 phase delay between TESTCKBIT and TESTCKBIT90.

The others device pins must be set as follows:
- TESTIERA, TESTQERA and TESTSYNC connected to VSS (Ground).
- CKBITEXT, TESTCKOCT and TESTCKOCT90 connected to VCC.

RXRSVITACT register = 0X84

7.1.2. Using the Viterbi Decoder And The Depuncturing Block – Without PLL



The CKBITEXT clock is the IEXT and QEXT symbols clock. These signals timing relationship is shown in section 8.6.
The TESTCKBIT clock frequency depends on the BPSK/QPSK mode and the puncturing rate as follows:

- BPSK
  TESTCKBIT_freq = CKBITEXT_freq x Punct_Rate
  If Punct_Rate = 1/2, you must generate a TESTCKBIT clock with a frequency of half the CKBITEXT one.
  In BPSK mode, the data stream comes on IEXT(2:0) pins only; the QEXT(2:0) must be connected to ground.
- QPSK
  TESTCKBIT_freq = CKBITEXT_freq x Punct_Rate x 2
  If Punct_Rate = 1/2, TESTCKBIT_freq = CKBITEXT_freq

The TESTCKBIT90 clock frequency is the same than the TESTCKBIT one; there is only a Pi/2 phase shift between them.

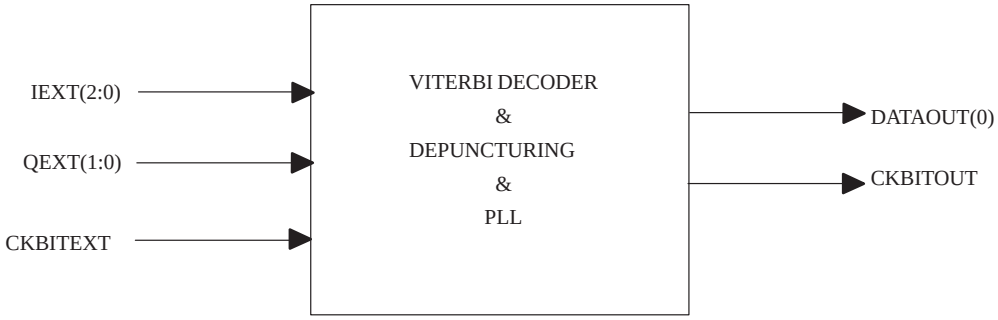
The others device pins must be set as follows:

- TESTIERA, TESTQERA and TESTSYNC connected to VSS (Ground).
- TESTCKOCT and TESTCKOCT90 connected to VCC.

RXRSVITACT register = 0X86

7.1.3. Using the Viterbi Decoder And The Depuncturing Block – With PLL

Using the TSS902E with its PLL is much more simple since you do not need to generate the TESTCKBIT and TESTCKBIT90 clocks.



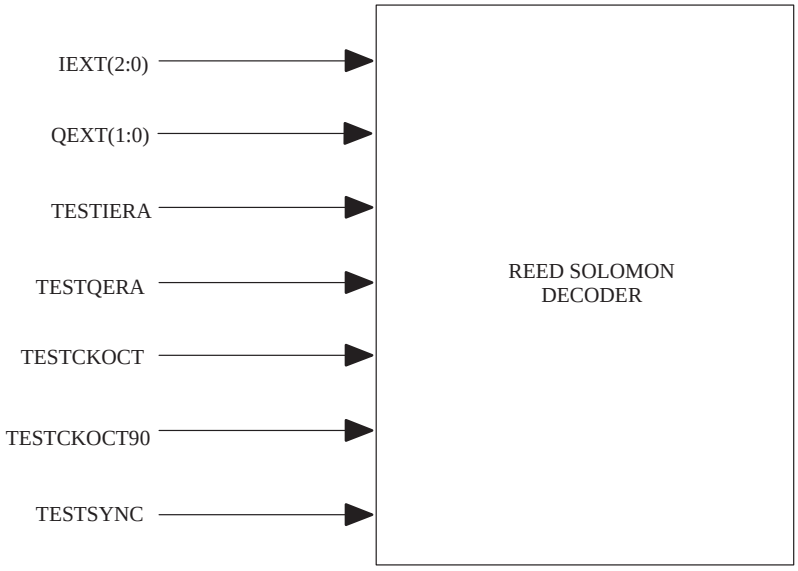
The CKBITEXT clock is the IEXT and QEXT symbols clock. These signals timing relationship is shown in section 8.6.

The others device pins must be set as follows:

- TESTIERA, TESTQERA and TESTSYNC connected to VSS (Ground).
- TESTCKOCT, TESTCKOCT90, TESTCKBIT and TESTCKBIT90 connected to VCC.

RXRSVITACT register = 0X87

7.1.4. Using the Reed Solomon Decoder only



The Reed–Solomon decoder input data stream is made of 8–bit parallel data coming on (from MSB to LSB): IEXT(2:0), QEXT(2:0), TESTIERA and TESTQERA pins.
The data stream is clocked on TESTCKOCT rising edge; there is a Pi/2 phase delay between TESTCKOCT and TESTCKOCT90 clocks.
The TESTSYNC signal must be supplied to mark each sync byte occurence.

7.2. Setting the Sync and Sync Loss Criterions

We address here the programming of the RXSYNCWCTL and RXDESYNCWCTL registers.
The most important function for the evaluation of data from the MPEG–2 Transport Stream (TS) is the sync

acquisition. The actual synchronization of the TS depends on the number of correct sync bytes necessary for the device to synchronize and on the number of distorded sync bytes which the device can not cope with.
Five consecutive correct sync bytes are usually considered as sufficient for sync acquisition and two or more consecutive corrupted sync bytes should indicate sync loss.

- RXSYNCWCTL – SY[2:0] = 5
- RXDESYNCWCTL – DSY[2:0] = 2

Mismatching bits are allowed during the sync bytes search; you can define the maximum allowed number of mismatching bits when comparing the data stream to the expected sync byte. Let us take an example: the expected sync byte is 0x47; if you allow one mismatching bit, then the following bytes will be considered as valid sync bytes: 0x46, 0x45, 0x43, 0x4F, 0x57, 0x67, 0x07 and 0x97.
Although you can set the number of mismatching bits to 0, 1 or 2, it is recommended to set it to zero so that the device does not get synced on wrong data.

- RXSYNCWCTL – MSY[1:0] = 0

Once the Frame sync is got, the device may loose it whenever the expected sync byte can not be found.
A mismatching bits mechanism also exists for sync loss conditions. Let us suppose that we expect a 0x47 value as sync byte. If the device reads 0x46, it will consider it as a valid sync byte if you have set the number of allowed mismatching bits to one. In that case, the device will remain synced. A value of one is a good trade–off for sync loss conditions.

- RXDESYNCWCTL – MDS[1:0] = 1

A time out value may be defined to program the maximum duration of the sync word search after the bit sync is found. Once the time out duration is elapsed, an interrupt is generated and the bit sync state machine goes one step beyond. The interrupt may be used by the external system to change the BER threshold, for example.
This parameter is not significant for the device to synchronize properly.

- RXSYNCWCTL – TOV[1:0] = 0

We can now summarize these registers setup with:

- RXSYNCWCTL = 0x05
- RXDESYNCWCTL = 0x0A

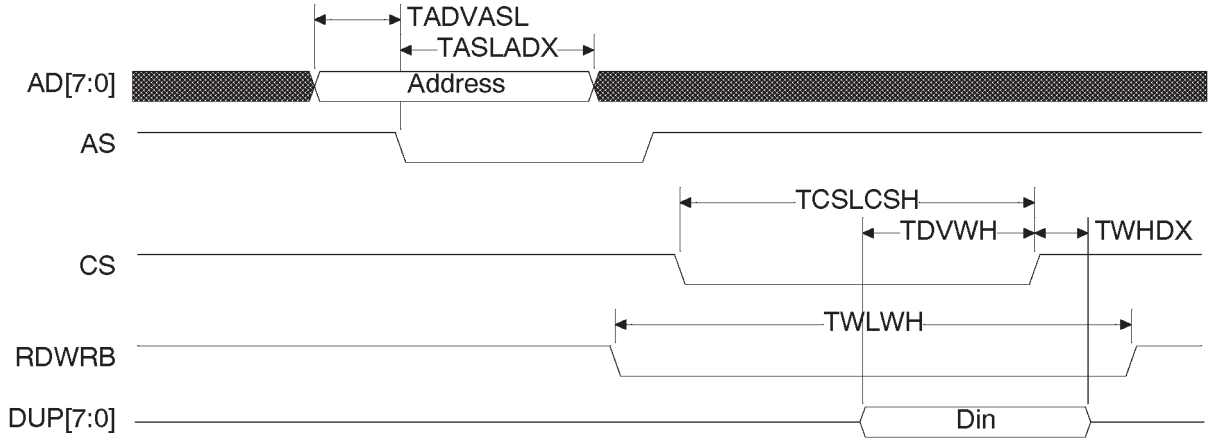
8. AC Characteristics

8.1. Reset Timing

PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TRSTLRSTH	RESETB minimum pulse width	10		usec

The Resetb signal is asynchronous.

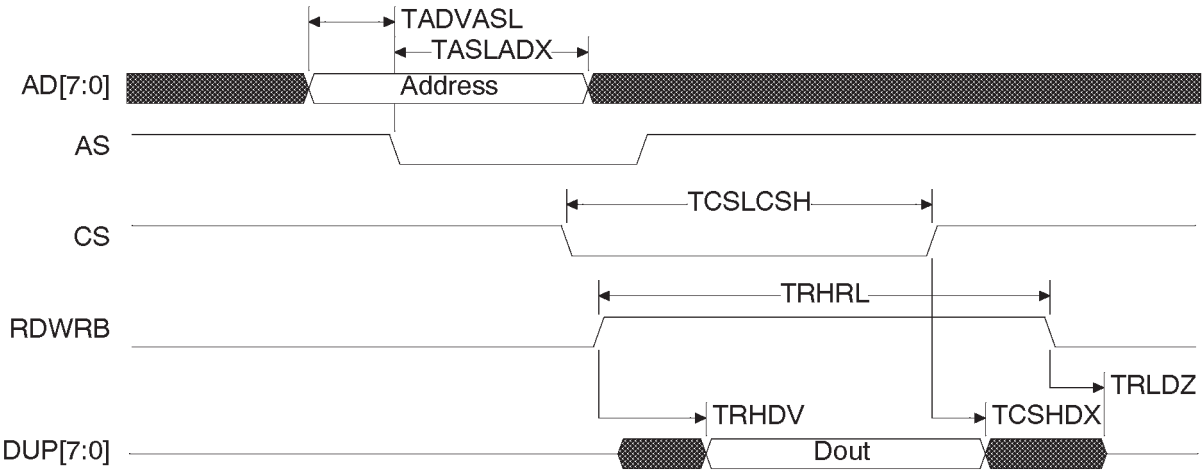
8.2. Microprocessor Write Timing



PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TADVASL	Address setup to AS low	5		nsec
TASLADX	Address hold from AS low	5		nsec
TCSLCSH	Chip Select minimum pulse width	20		nsec
TWLWH	Write signal minimum pulse width	20		nsec
TDVWH	Data setup to end of write (CS high or RDWRB high)	10		nsec
TWHDX	Data hold from end of write (CS high or RDWRB high)			nsec

The device internal write condition is: CS low and RDWRB low.

8.3. Microprocessor Read Timing

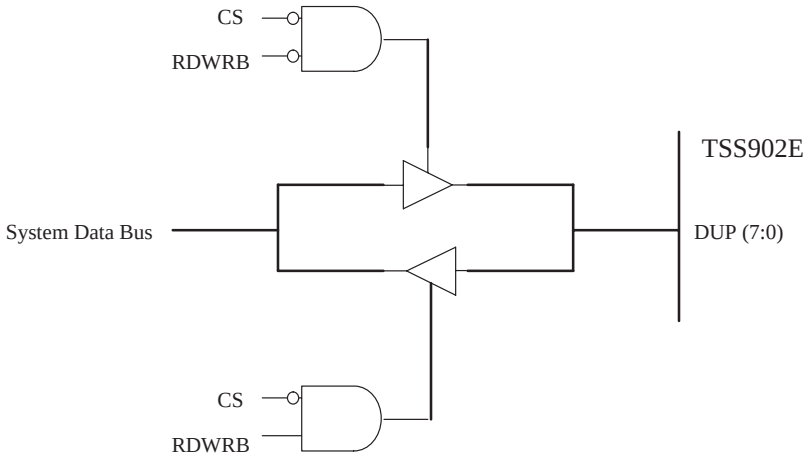


PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TADVASL	Address setup to AS low	5		nsec
TASLADX	Address hold from AS low	5		nsec
TCSLCSH	Chip Select minimum pulse width	20		nsec
TRHRL	Read signal minimum pulse width	20		nsec
TRHDV	Read start to output valid		40	nsec
TCSHDX	End of read to invalid output			nsec
TRLDZ	End of read to high impedance			nsec

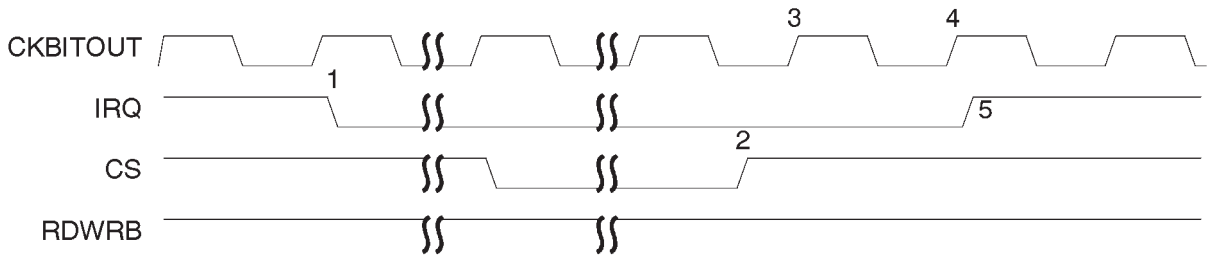
The device internal read condition is: CS low and RDWRB high.

8.4. Microprocessor data bus interface

In systems in which the TSS902E DUP data bus is shared with others devices, it is mandatory to isolate the TSS902E DUP bus by inserting an octal bidirectional buffer between the device DUP bus and the external system microcontroller data bus, as shown in the following figure:



8.5. Interrupt Timing

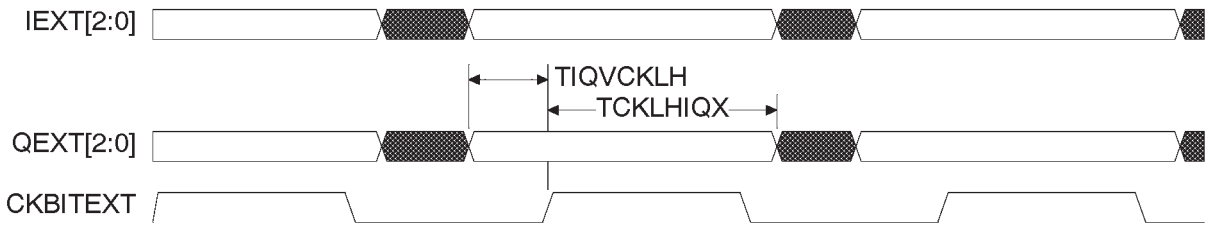


After IRQ output goes down, the microcontroller starts the interrupt registers read operation (*RxVitRSInt0* & *RxVitRSInt1*) in order to find the interrupt origin. Both registers contents must be read and stored; reading these registers clears them. As long as these registers remain uncleared, the IRQ signal stays at low level.

A CS rising edge transition indicates an end of read. Two CKBITOUT rising edges after end of read, the IRQ signal returns to high level.

The microcontroller must guarantee 4 clock periods between two interrupt registers reading.

8.6. IEXT[2:0] and QEXT[2:0] Inputs Timing

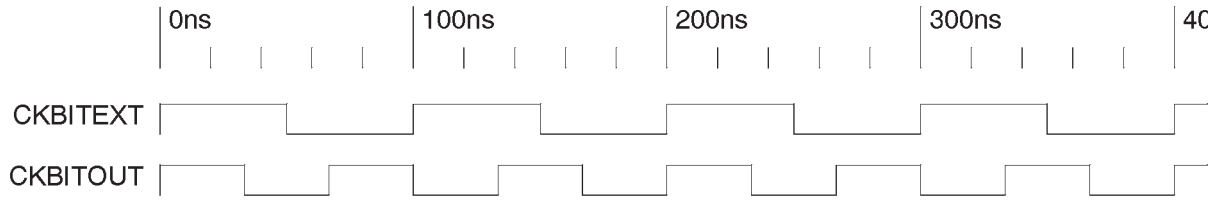


PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TIQVCKLH	IEXT & QEXT setup to CKBITEXT high	5		nsec
TCKLHIQX	IEXT & QEXT hold from CKBITEXT high	5		nsec

8.7. Outputs Timing

PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TCKDATA	Delay from CKOCTOUT clock falling edge to DATAOUT.	5		nsec
TCKSYNCOUT	Delay from CKOCTOUT clock falling edge to SYNCOUT.	5		
TCKFRAMEOUT	Delay from CKOCTOUT clock falling edge to FRAMEOUT.	5		
TCKBURSTOUT	Delay from CKOCTOUT clock falling edge to BURSTSYNCOUT.	5		
TCKDATAS	Delay from CKBITOUT clock falling edge to DATAOUTS.	5		nsec

8.8. PLL block

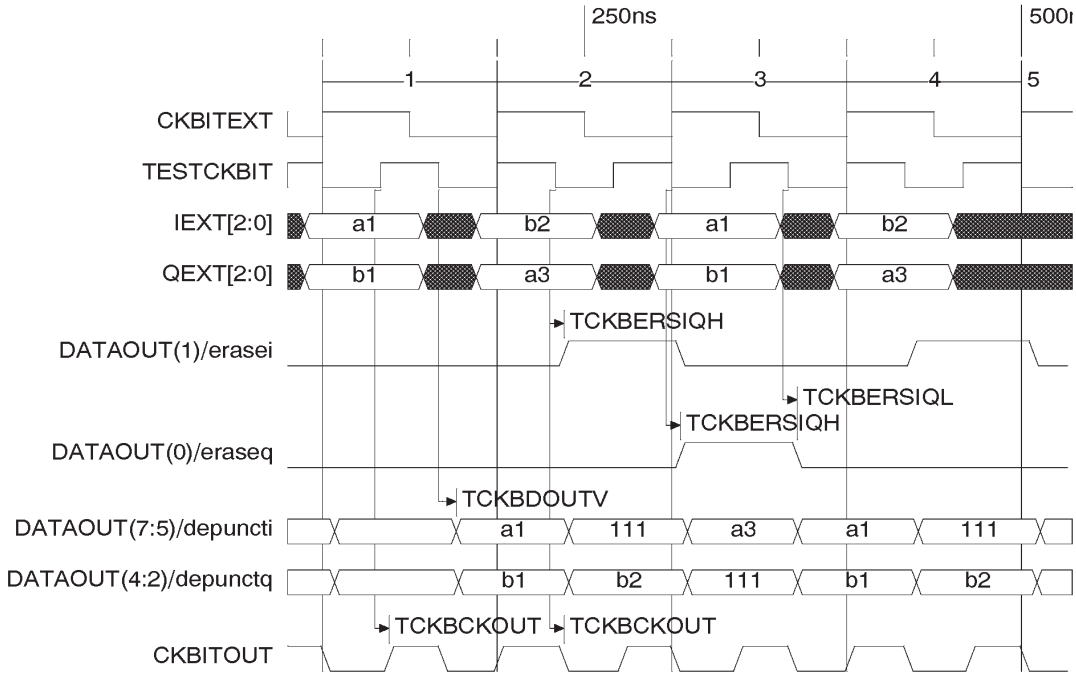


PLL block functionality requirements:

- Enable the PLL block by setting the *RxRSVitAct* register *PLLE* bit.
- The *RxVitFreqIn* register *FREQ[6:0]* and *VCOC* bits must be set as explained in the Registers section.
- The code rate must be defined through the *RxVitRate* register and the manual mode must be selected through the *RxVitControl* one.
  The above timing diagram requires *FREQ[6:0]* to be set to 00H and *VCOC* to 0; the code rate is 3/4 (*RxVitRate* = 04H).

The internally generated depuncturing clock can be observed at the CKBITOUT pin.

8.9. Depuncturing block

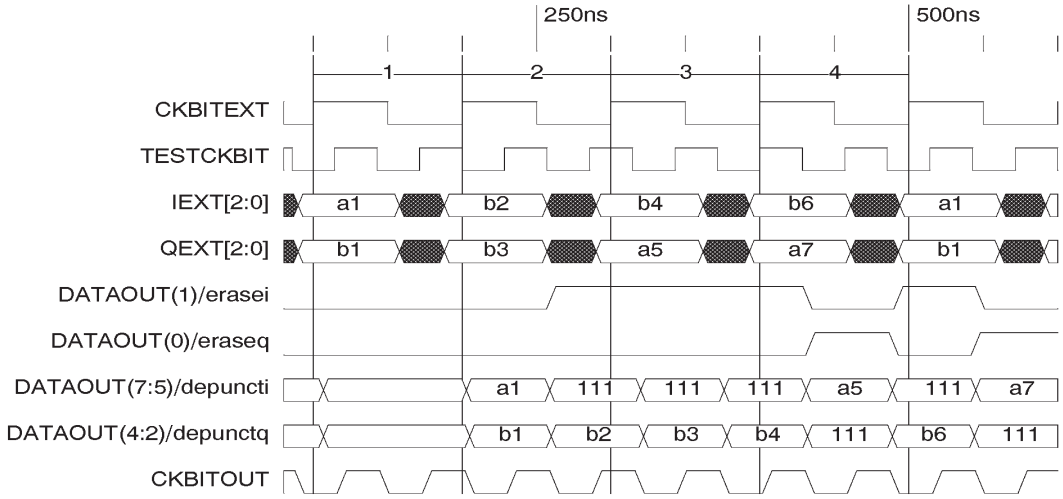


PARAMETER / EVENT	DEFINITION	MIN	MAX	UNITS
TCKBERSIQH	Delay from TESTCKBIT clock falling edge to erasei/eraseq rising edge.		20	nsec
TCKBERSIQL	Delay from TESTCKBIT clock falling edge to erasei/eraseq falling edge.		20	nsec
TCKBDOUTV	Delay from TESTCKBIT clock falling edge to depunctured output data valid.		20	nsec
TCKBCKOUT	Delay from TESTCKBIT clock falling edge to CKBITOUT falling edge.			nsec

Depuncturing block test requirements:

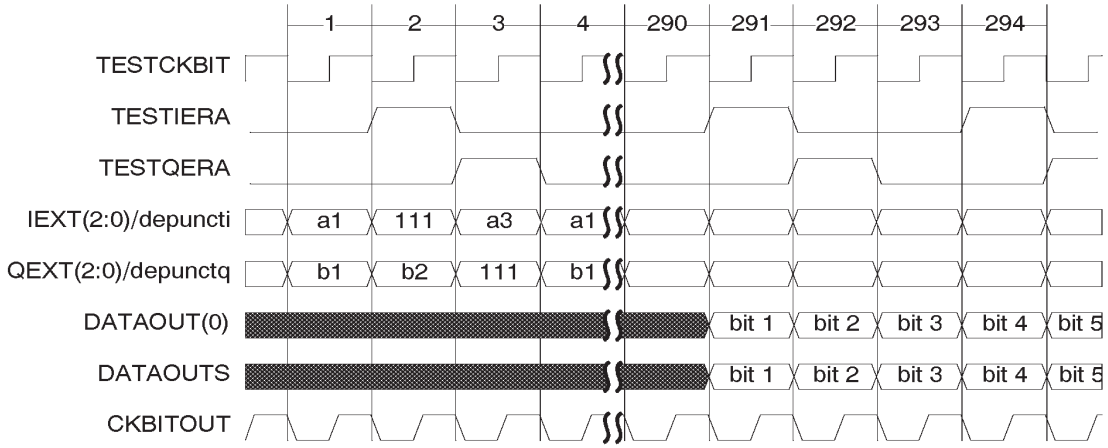
- Enable the Depuncturing block by setting the *RxRSVitAct* register *DPE* bit.
- The code rate must be defined through the *RxVitRate* register and the manual mode must be selected through the *RxVitControl* one. If the code rate is different to 1/2 the Synchro–bit must be enable (set *SYBE*)
- The TESTCKBIT clock frequency and phase must be set properly, according to the selected code rate.

In the above timing diagram the depuncturing rate is 3/4 (RxVitRate = 04H).



The above timing diagram shows a Depuncturing block test sequence using a 7/8 rate.

8.10. Viterbi decoder

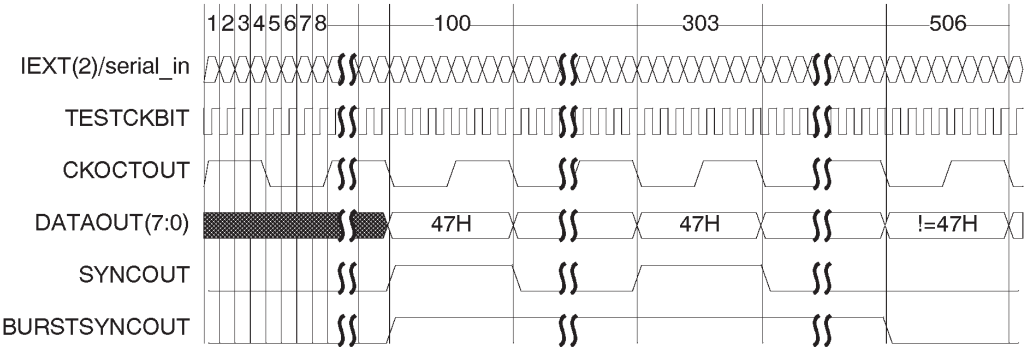


Viterbi decoder test requirements:

- Enable the Viterbi decoder by setting the *RxRSVitAct* register *VE* bit.
  In the above sequence, we assume that a 3/4 code rate is used; the incoming data is already depunctured and sequenced by the TESTCKBIT clock.
- The TESTIERA and TESTQERA inputs must be controlled accordingly to the 3/4 rate.
- In Viterbi test mode, the decoded bits can be found on DATAOUT(0) or DATAOUTS outputs after 290 TESTCKBIT clock periods (the Viterbi decoder latency).
- The CKBITOUT output clock must be checked.

An other possible Viterbi test sequence is to feed IEXT and QEXT with data of code rate 1/2 (no puncturing); in that case, the TESTIERA and TESTQERA inputs must be held low continuously.

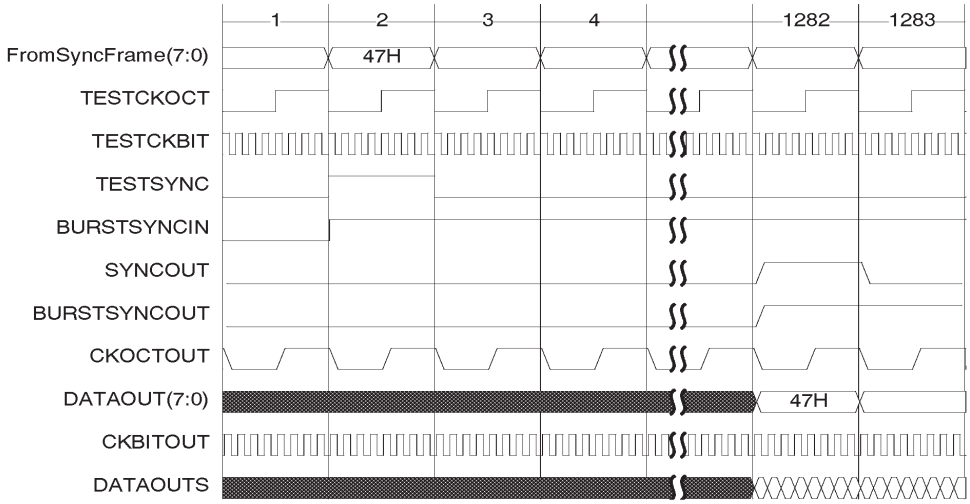
8.11. Synchro–Frame block



Synchro–Frame block test requirements:

- Enable the Synchro–Frame block by setting the *RxRSVitAct* register *SYE* bit.
  In the above timing diagram, the incoming data onto IEXT(2) is supposed to be the Viterbi decoder output, clocked by the TESTCKBIT signal.
- The *RxRSFrameL* register is set to its default hexadecimal CC value (a 204 bytes frame length).
- The *RxSyncWord* register is set to its default hexadecimal 47 value.
- The output data can be read onto DATAOUT(7:0) on CKOCTOUT clock falling edge.
- The SYNCOUT signal goes high whenever the sync word is found.
- The BURSTSYNCOUT signal goes high and remains high as long as the sync word is found.

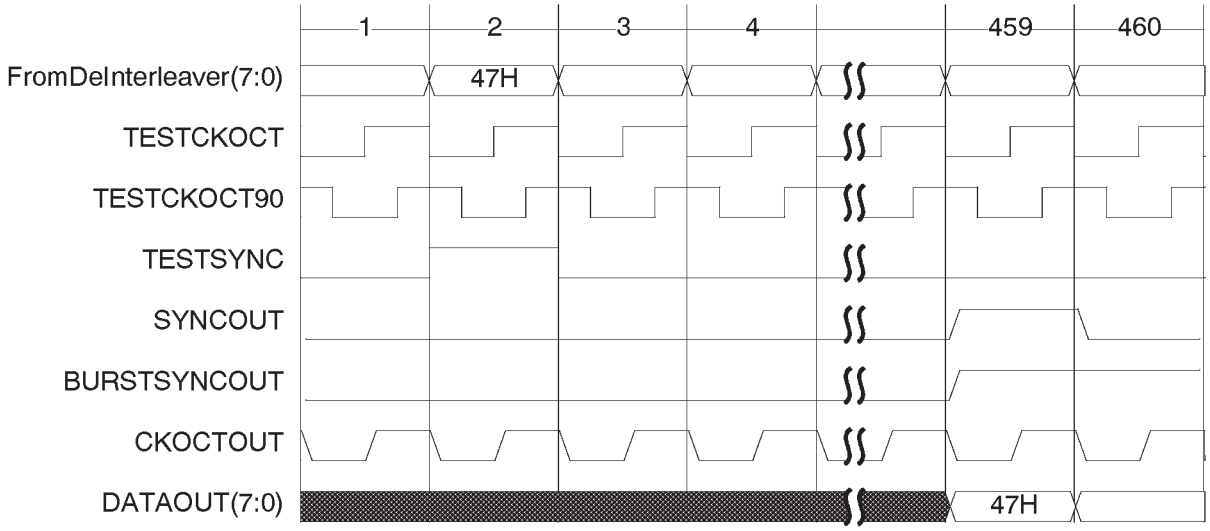
8.12. De–Interleaver block



De–Interleaver block test requirements:

- Enable the de–interleaver block by setting the *RxRSVitAct* register *DIE* bit.
  In the above timing diagram, the incoming data FromSyncFrame(7:0) is supposed to be the Synchro–Frame block output and must be valid on the TESTCKOCT clock falling edge. In test mode, FromSyncFrame(7:0) is made, from MSB to LSB of: IEXT(2:0), QEXT(2:0), TESTIERA and TESTQERA.
- There is a Pi/2 delay between TESTCKOCT and TESTCKOCT90 clocks.
- The *RxRSFrameL* register is set to its default hexadecimal CC value (a 204 bytes frame length).
- The *RxSyncWord* register is set to its default hexadecimal 47 value.
- The TESTSYNCIN input must be high whenever the sync byte is present onto the input bus.
- If sync words periodically appear into the input data stream, the BURSTSYNCIN input must be kept high.
- The output data can be read onto DATAOUT(7:0) on CKOCTOUT clock falling edge after the de–interleaver block latency (1280 TESTCKOCT clock periods). The output data may also be read onto the DATAOUTS serial output on CKBITOUT falling edge.

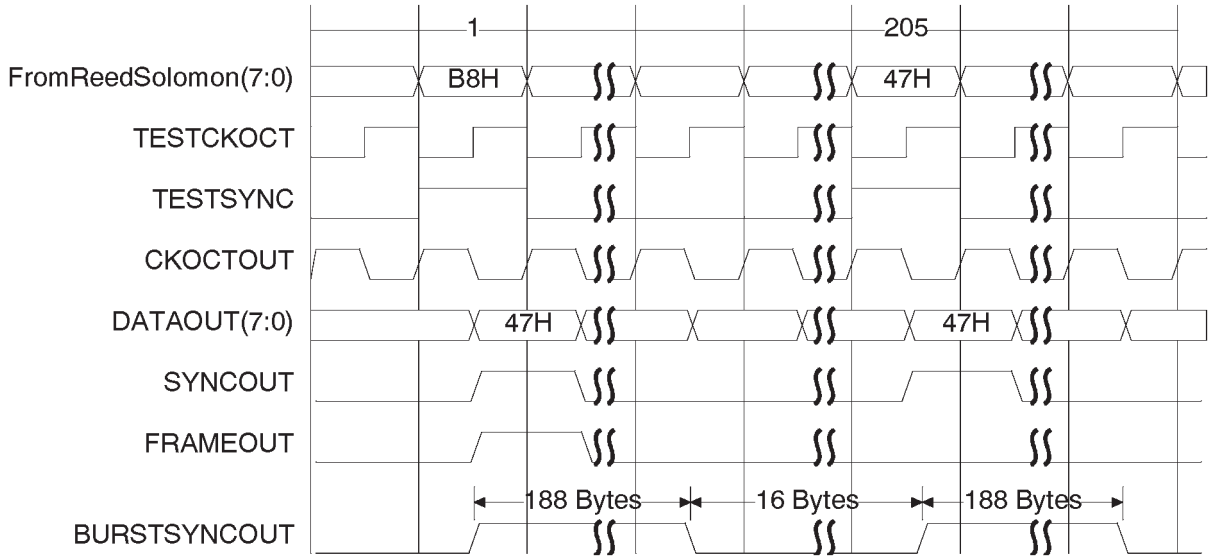
8.13. Reed–Solomon decoder



Reed–Solomon decoder test requirements:

- Enable the Reed–Solomon block by setting the *RxRSVitAct* register *RSE* bit.
  In the above timing diagram, the incoming data FromDeInterleaver(7:0) is supposed to be the De–Interleaver block output and must be valid on the TESTCKOCT clock falling edge. In test mode, FromDeInterleaver(7:0) is made, from MSB to LSB of: IEXT(2:0), QEXT(2:0), TESTIERA and TESTQERA.
- There is a Pi/2 delay between TESTCKOCT and TESTCKOCT90 clocks.
- The *RxRSFrameL* register is set to its default hexadecimal CC value (a 204 bytes frame length).
- The *RxSyncWord* register is set to its default hexadecimal 47 value.
- The TESTSYNCIN input must be high whenever the sync byte is present onto the input bus.
- If sync words periodically appear into the input data stream, the BURSTSYNCIN input must be kept high.
- The output data can be read onto DATAOUT(7:0) on CKOCTOUT clock falling edge after the Reed–Solomon decoder latency (2 x FrameLength + 50bytes).

8.14. Descrambler block



Descrambler block test requirements:

- Enable the Descrambler block by setting the RxRSVitAct register EDE bit.
  In the above timing diagram, the incoming data FromReedSolomon(7:0) is supposed to be the Reed–Solomon decoder output and must be valid on the TESTCKOCT clock falling edge. In test mode, FromReedSolomon(7:0) is made, from MSB to LSB of: IEXT(2:0), QEXT(2:0), TESTIERA and TESTQERA.
- There is a Pi/2 delay between TESTCKOCT and TESTCKOCT90 clocks.
- The RxRSFrameL register is set to its default hexadecimal CC value (a 204 bytes frame length).
- The RxSyncWord register is set to its default hexadecimal 47 value.
- The TESTSYNCIN input must be high whenever the sync byte or its complemented value are present onto the input bus.
- If sync words periodically appear into the input data stream, the BURSTSYNCIN input must be kept high.
- The output data can be read onto DATAOUT(7:0) on CKOCTOUT clock falling edge after the Descrambler block latency (half a TESTCKOCT period).
- The BURSTSYNCOUT signal goes high on B8H occurence and remains high for a 188 bytes duration.

9. Absolute Maximum Rating and Operating Conditions

9.1. Absolute Maximum Rating

Table 6: Absolute Maximum Ratings

• Storage Temperature	–65 to +150°C
• Voltage on any other Pin to VSS	–0.5 to +6.5 V
• I_{OL} per I/O Pin	
• Power Dissipation	3 W

9.2. Operating Conditions

Table 7: Operating Conditions

• Ambient Temperature Under Bias	
Space	–55 to +125°C
Military	–55 to +125°C
• VDD	
High Speed versions	4.5 to 5.5 V

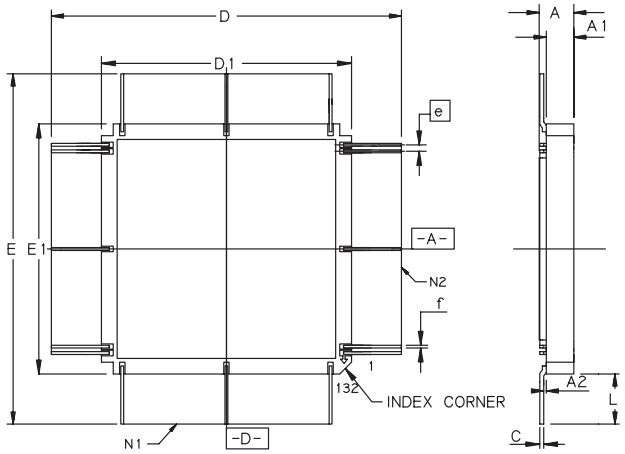
Note:
Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "operating conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

10. DC Characteristics – Space and Military Ranges

Table 8: DC Characteristics at 5V +10%, T_A= –55 to +125°C

Symbol	Parameter	Min	Typical	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	–0.5		0.2VDD - 0.1	V	
V_{IH}	Input high Voltage	0.2VDD + 0.9		VDD + 0.5	V	
V_{OL}	Output Low Voltage			0.4 0.4	V	I_{OL}= 3 mA I_{OL}= 6 mA
V_{OH}	Output high Voltage	2.4			V	I_{OH}= –10 µA
I_{IL}	Input Leakage Current No pull–up/pull–down With pull–up With pull–down		 –50	 +/– 5 +360	 µA	
I_{OZ}	3–State Output Leakage Current		+/– 1	+/– 5	µA	
ICCOP	Operating Current Viterbi block Reed–Solomon block		 20.5 2.2		 mA/MHz	
ICCSB	Standby Current		20		mA	

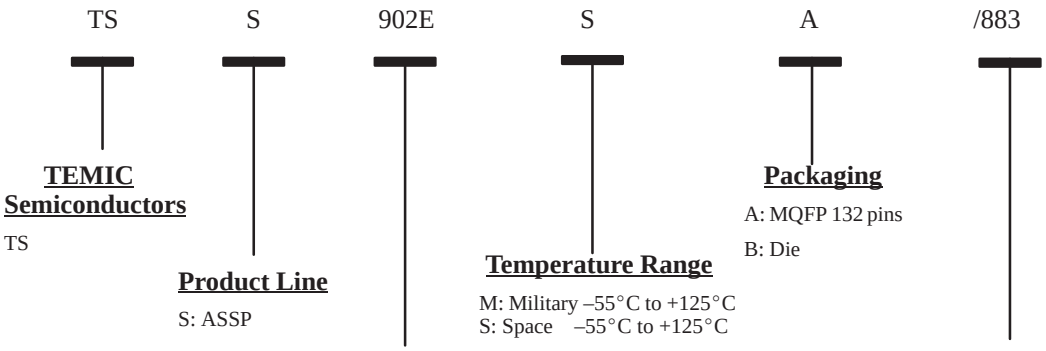
Package



	mm		inch	
	Min	Max	Min	Max
A	2.36	2.82	.093	.111
C	0.152 TYP.		.006 TYP.	
D	37.00	39.38	1.457	1.550
D1	24.00	24.38	.945	.960
E	37.00	39.38	1.457	1.550
E1	24.00	24.38	.945	.960
e	0.635 BSC		.025 BSC	
f	0.200 REF		.008 REF	
A1	1.47	1.83	.058	.072
A2	0.203 REF		.008 REF	
L	6.50	7.50	.256	.295
N1	33		33	
N2	33		33	

Figure 2: 132 pins MQFPF

Ordering Information



Example: TSS902E
S A /883
FRANCE
yyww Lot Nb

The information contained herein is subject to change without notice. No responsibility is assumed by TEMIC for using this publication and/or circuits described herein : nor for any possible infringements of patents or other rights of third parties which may result from its use.