PCD03V Features

Turbo Decoder

- 8 state 3GPP™ (UMTS and LTE) and 3GPP2 (cdma2000/1xEV–DV Release D and 1xEV–DO Release B) compatible
- Rate 1/2, 1/3, 1/4 or 1/5
- 40 to 5114 (3GPP™ UMTS), 40 to 6144 (3GPP™ LTE) or 17 to 21504 (3GPP2) bit interleaver
- Optional external interleaver parameters for 3GPP™ LTE and programmable row coefficients for 3GPP2 interleaver
- Up to 309 MHz internal clock
- Up to 29.4 Mbit/s with 5 decoder iterations
- 6–bit signed magnitude or two's complement input data
- Optional log–MAP or max–log–MAP constituent decoder algorithms
- Up to 128 iterations in 1/2 iteration steps.
- Optional power efficient early stopping
- Optional extrinsic information scaling and limiting
- Estimated channel error output
- Implement one, two, three, or four different standards from the one core
- Free simulation software

Viterbi Decoder (Optional)

- 16, 32, 64 or 256 states (constraint lengths 5, 6, 7 or 9, encoder memory m = 4, 5, 6 or 8)
- Rate 1/2, 1/3 or 1/4
- Data lengths from 1 to 32768–m bits with tail termination of m bits
- Optional tail biting decoding from m to 2048 data bits
- Up to 7.1 Mbit/s (256 state) or 24.4 Mbit/s (16, 32 or 64 states)
- 6–bit signed magnitude or two's complement input data
- Estimated channel error output

- Available as VHDL core for Xilinx FPGAs under SignOnce IP License. ASIC, Altera, Lattice and Microsemi cores available on request.

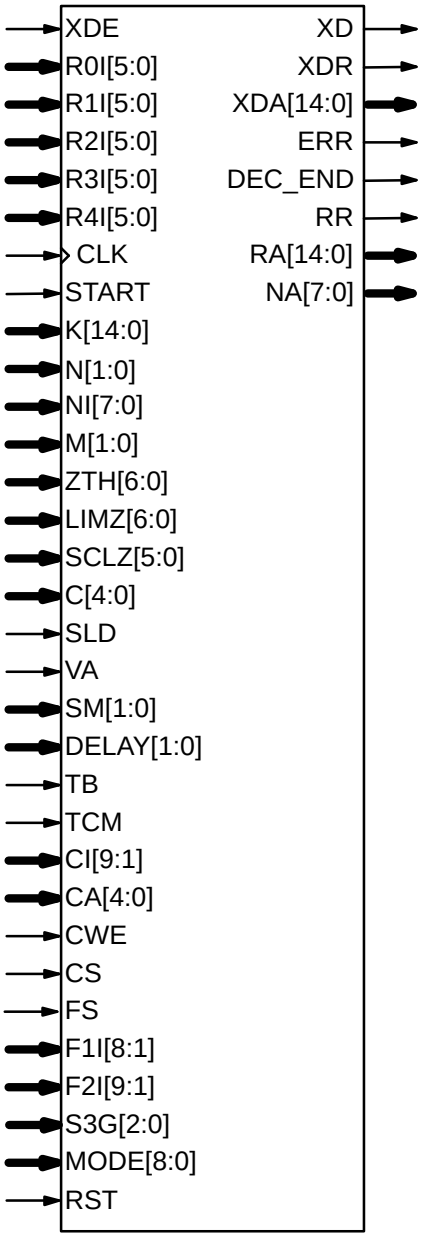

Figure 1: PCD03V schematic symbol.

Introduction

The PCD03V is a fully compatible 3GPP™ (UMTS [1] and LTE [2]) and 3GPP2 (cdma2000/1xEV–DV Release D [3] and 1xEV–DO Release B [4]) error control decoder. The 3GPP™ and 3GPP2 turbo codes have a number of similarities, but also important differences which affect their implementation. The biggest similarity is that they use the same constituent 8 state systematic recursive convolutional code. The interleavers of 3GPP™ UMTS and 3GPP2 also use a similar row/

column architecture, but are quite different in their complexity. 3GPP™ LTE uses a simple quadratic permutation interleaver. The 3GPP™ and 3GPP2 codes also have different tails.

The 3GPP™ UMTS interleaver has either 5, 10 or 20 rows and a number of columns equal to p–1, p or p+1, where p is a prime number from 7 to 257. The 3GPP2 interleaver has 32 rows and 2^n columns, where n ranges from 2 to 10.

For 3GPP™ UMTS, the use of prime numbers and other complexities implies that a number of parameters need to be calculated before the interleaver can be used. When the block length is changed, the interleaver parameters are automatically calculated using an efficient internal circuit. This occurs during the first half iteration of decoding. Thus, decoding can start immediately, without having to wait for the parameters to be calculated. The block length can range from 40 to 5114 bits.

For 3GPP2, the use of powers of 2 greatly simplifies parameter calculation. The block length can range from 17 to 21504 bits, although the 3GPP2/1xEV–DV standard only uses 23 specific interleavers (186, 378, 402, 570, 762, 786, 1146, 1530, 1554, 2298, 3066, 3090, 4602, 4626, 6138, 6162, 9210, 9234, 12282, 12306, 15378, 18450 and 20730 bits). The forward link for 1xEV–DO uses eight different interleavers (122, 250, 506, 1018, 2042, 3066, 4090, and 5114 bits). The reverse link for 1xEV–DO uses 12 different interleavers (122, 250, 506, 762, 1018, 1530, 2042, 3066, 4090, 6138, 8186, and 12282 bits). For interleaver sizes less than 65, the parameter n is fixed at 2 (4 columns).

For 3GPP™ LTE, there are 188 interleaver sizes ranging from 40 to 6144 bits. Two parameters f_1 and f_2 are used by the interleaver. All interleaver sizes from 40 to 504 bits that are a multiple of 8, 512 to 1008 bits that are a multiple of 16, 1024 to 2016 bits that are multiple of 32, and 2048 to 6144 bits that are a multiple of 64 are specified.

For 3GPP™ only a code rate of 1/3 is specified. However, the PCD03V can also optionally decode rate 1/2, 1/4 and 1/5 turbo codes. For 3GPP2, rate 1/2, 1/3, 1/4 and 1/5 are specified.

For 3GPP™, each tail of the two constituent encoders are terminated using all the data and parity bits (for a total of 12 bits for rate 1/3). For rate 1/2 the PCD03V uses the same tail as for rate 1/3. For rate 1/4 and 1/5, a total of 18 tail bits are used.

For 3GPP2, the number of tail bits is equal to 6n, for a rate 1/n code. For rate 1/2, the tails for 3GPP™ and 3GPP2 are determined in the same way. For rate 1/3 and 1/4 though, data bits are repeated to make up for the additional tail bits in the 3GPP2 standard. For rate 1/5 the data bit is repeated three times for 1xEV–DV and two times for 1xEV–DO. The second parity bit is repeated two times for 1xEV–DO.

The MAP03V MAP decoder core is used with the PCD03V core to iteratively decode the 3GPP™ or 3GPP2 turbo code. The Log–MAP algorithm for maximum performance or the max–log–MAP algorithm for minimum complexity can be selected. The sliding block algorithm is used with sliding block lengths of 32 or 64. Six–bit quantisation is used for maximum performance. The extrinsic information can be scaled and limited with each half iteration, improving performance with max–log–MAP decoding.

The VA08V Viterbi decoder core is used with the PCD03V core to decode the 3GPP™ and 3GPP2 convolutional codes. The decoder shares its traceback memory with the internal interleaver memory of the turbo decoder, minimising complexity. Minimum traceback depths of 33, 65 or 129 bits for 16, 32 and 64 states, or depths of 57, 113 or 225 bits for 256 states can be selected. 6–bit quantisation is used.

The turbo decoder can achieve up to 29.4 Mbit/s with 5 iterations and max–log–MAP decoding using a 309 MHz internal clock (K = 5114). Log–MAP decoding decreases speed by about 32%. Optional early stopping allows the decoder to greatly reduce power consumption with little degradation in performance. The Viterbi decoder can achieve up to 7.1 Mbit/s with 256 states (K = 504) and 24.4 Mbit/s with 16, 32 or 64 states (K = 506).

Figure 1 shows the schematic symbol for the PCD03V decoder. The VHDL core can be used with Xilinx Integrated Software Environment (ISE) or Vivado software to implement the core in Xilinx FPGA's.

Table 1 shows the resources used for Kintex–7. Resources for Virtex–5, Virtex–6 and other 7–series devices are similar to that for Kintex–7. The MODE[8:0] inputs can be used to select various decoder implementations. The input/output memory is not included. Only one global clock is used. No other resources are used. If not specified, the turbo decoder uses small–log–MAP.

Table 2 shows the performance achieved with various Xilinx parts. T_{cp} is the minimum clock period over recommended operating conditions. These performance figures may change due to device utilisation and configuration.

Table 1: Resources used.

Configuration	MAP	Turbo Rates	Viterbi Rates	6–Input LUTs	18KB Block RAMS
UMTS	Max	1/3	–	2538	3
UMTS	Small	1/3	–	3232	3
UMTS	Large	1/3	–	3374	3
UMTS & Viterbi	Small	1/3	1/2–1/3	3987	3
1xEV–DV	Small	1/2–1/5	–	3231	11
1xEV–DV & Viterbi	Small	1/2–1/5	1/2–1/4	4106	11
UMTS/1xEV–DV	Small	1/2–1/5	–	3774	11
UMTS/1xEV–DV & Viterbi	Small	1/2–1/5	1/2–1/4	4653	11
LTE	Small	1/3	–	2731	3
LTE & Viterbi	Small	1/3	1/3	3507	3

Table 2: Performance of Xilinx parts.

Xilinx Part	T_{cp} (ns)	Turbo* Mbit/s	K=9 Mbit/s	K=7 Mbit/s
XC5VLX30–1	5.923	16.0	3.9	13.3
XC5VLX30–2	5.082	18.7	4.5	15.5
XC5VLX30–3	4.527	20.9	5.1	17.4
XC6VLX75T–1	5.045	18.8	4.5	15.6
XC6VLX75T–2	4.366	21.7	5.3	18.1
XC6VLX75T–3	3.965	23.9	5.8	19.9
XC7Z015–1	6.484	14.6	3.5	12.1
XC7Z015–2	5.293	17.9	4.3	14.9
XC7Z015–3	4.695	20.2	4.9	16.8
XC7A35T–1	6.418	14.8	3.6	12.3
XC7A35T–2	5.293	17.9	4.3	14.9
XC7A35T–3	4.699	20.2	4.9	16.8
XC7K70T–1	4.313	22.0	5.3	18.3
XC7K70T–2	3.483	27.2	6.6	22.6
XC7K70T–3	3.232	29.4	7.1	24.4

*Max–log–MAP, 5 iterations, K = 5114, SLD = 1

Signal Descriptions

C MAP Decoder Constant
0–9 (MODE1 = 0)
0–17 (MODE1 = 1)

CA 3GGP2 row constant address (0 to 31)

CI 3GPP2 row constant (0 to 511)

CS 3GPP2 row constant select
0 = Select internal row constants
1 = Select programmed row constants

CWE 3GPP2 row constant write enable

CLK System Clock

DEC_END Decode End Signal

DELAY Viterbi Decoder Delay
0 = delay 64+m (TB = 0)
1 = delay 128+m (TB = 0)
2 = delay 256+m (TB = 0, MODE[6:5] > 1)
0 = delay 128+m (TB = 1)
1 = delay 256+m (TB = 1)
2 = delay 512+m (TB = 1, MODE[6:5] > 1)
m = encoder memory (4, 5, 6 or 8)

ERR Estimated Error

F1I LTE external parameter 1 (0 to 255)
F1I = f_1 div 2.

F2I LTE external parameter 2 (0 to 511)
F2I = f_2 div 2.

FS LTE external parameter select
0 = Select internal parameters
1 = Select external parameters

K Interleaver Length
40–5114 for 3GPP™ UMTS
40–6144 for 3GPP™ LTE
17–21504 for 3GPP2

LIMZ Extrinsic Information Limit (1–127)

M Early Stopping Mode
0 = no early stopping
1 = early stop at odd half iteration
2 = early stop at even half iteration
3 = early stop at any half iteration

MODE Implementation Mode (see Table 3)

N Code Rate
0 = rate 1/4
1 = rate 1/5 (turbo only)
2 = rate 1/2
3 = rate 1/3

NA Half Iteration Number (0–255)

NI Number of Half Iterations (0–255)
NI = 2I–1 where I is number of iterations

R0I–R4I Received Data

RA Received Data Address

RR Received Data Ready

RST Synchronous Reset

S3G[1:0] Turbo Decoder Standard Select
0 = 3GPP™ UMTS
1 = 3GPP™ LTE

2 = 3GPP2/1xEV–DV
3 = 3GPP2/1xEV–DO

S3G[2] Viterbi Decoder Standard Select
0 = 3GPP™
1 = 3GPP2

SCLZ Extrinsic Information Scale (1–32)

SLD MAP Decoder Delay
0 = delay 138
1 = delay 266

SM Viterbi Decoder State Select
0 = 16 state (constraint length 5)
1 = 32 state (constraint length 6)
2 = 64 state (constraint length 7)
3 = 256 state (constraint length 9)

START Decoder Start

TB Tail Biting Select for Viterbi decoder
0 = Teminated with *m* symbol tail
1 = Tail biting decoding

TCM Input data type select
0 = Signed Magnitude
1 = Two's Complement

VA Viterbi Decoder Select
0 = turbo decoder
1 = Viterbi decoder

XD Decoded Data

XDA Decoded Data Address

XDE Decoded Data Enable

XDR Decoded Data Ready

ZTH Early Stopping Threshold (1–127)

Table 3 describes each of the MODE[8:0] inputs that are used to select various decoder implementations. Note that MODE[8:0] are "soft" inputs and should not be connected to input pins or logic. These inputs are designed to minimise decoder complexity for the configuration selected.

Note that UMTS, LTE, 1xEV–DO and 1xEV–DV have maximum interleaver sizes of 5K, 6K, 12K and 20.25K, respectively. However, sizes of 6K, 12K and 22K are provided since the Xilinx BlockRAM step size is 2K with 8–bit words.

Table 4 gives the various pin connections for the decoder modes. SM[1:0] = 3 and TB = 0 to select the 256 state Viterbi decoder for 3GPP™ UMTS and 3GPP2. SM[1:0] = 2 and TB = 1 to select the 64 state tail biting Viterbi decoder for 3GPP™ LTE.

Turbo Decoder Parameters

For optimal performance, the maximum a posteriori (MAP) [5] constituent decoder is used which is dependent on the signal to noise ratio (SNR). Unlike other turbo decoders with suboptimum soft–in–soft–in (SISO) decoders, using the MAP (or specifically the log–MAP [6]) algorithm can provide up to 0.5 dB coding gain at low SNRs. Log–MAP operation is enabled when MODE0 is high.

Table 3: MODE selection

Input	Description
MODE0	0 = max–log–MAP 1 = log–MAP
MODE1	0 = small log–MAP (C4 = 0) 1 = large log–MAP
MODE2	0 = rate 1/2–1/3 turbo 1 = rate 1/2–1/5 turbo
MODE3	0 = turbo decoder 1 = turbo and Viterbi decoder
MODE4	0 = rate 1/2–1/3 Viterbi 1 = rate 1/2–1/4 Viterbi
MODE[6:5]	0 = 4K Interleaver 1 = 6K Interleaver (3GPP™) 2 = 12K Interleaver (1xEV–DO) 3 = 22K Interleaver (1xEV–DV)
MODE[8:7]	0 = Other implementation 1 = 3GPP2/UMTS Implementation 2 = 3GPP2/LTE Implementation 3 = 3GPP™ Implementation

Table 4: Decoder mode

Mode	S3G [1:0]	VA	MODE [6:2]	K[14:12]	N[1:0]
3GPP™ UMTS	00	0	01000	00,PIN	11
3GPP™ UMTS VA	00	PIN	01010	00,PIN	1,PIN
3GPP™ LTE	01	0	01000	00,PIN	11
3GPP™ LTE VA	01	PIN	01010	00,PIN	11
1xEV–DV	10	0	11001	PIN	PIN
1xEV–DV VA	10	PIN	11111	PIN	PIN
1xEV–DO Forward	11	0	01001	00,PIN	PIN
1xEV–DO Reverse	11	0	10001	0,PIN	PIN

With binary phase shift keying (BPSK, $m = 1$) or quadrature phase shift keying (QPSK, $m = 2$) modulation (see Figure 2) the decoder constant C should be adjusted such that

$$C = A\sigma^2\sqrt{m}/2. \quad (1)$$

where A is the signal amplitude and σ^2 is the normalised noise variance given by

$$\sigma^2 = 1/(2mRE_b/N_0). \quad (2)$$

E_b/N_0 is the energy per bit to single sided noise density ratio and $R = 1/(n+6(\lfloor n/2 \rfloor+1)/K)$ for 3GPP™, $R = 1/(n+6n/K)$ for 3GPP2, K is the data length, and $n = 2–5$. C should be rounded to the

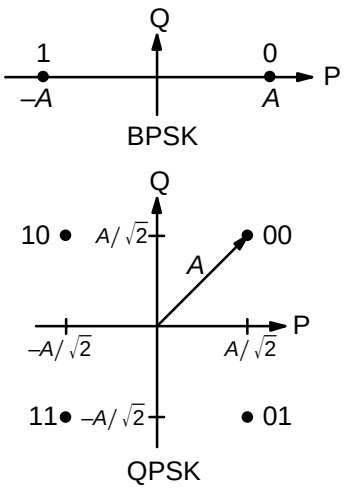

Figure 2: BPSK and QPSK signal sets.

nearest integer and limited to be no higher than 17 with MODE1 high and 9 with MODE1 low. Max–log–MAP [6] operation occurs when C = 0. Due to quantisation effects, C = 1 is equivalent to C = 0. Max–Log–MAP operation is also enabled when MODE0 is low.

For each code (with a particular block size, rate and number of iterations), there will be a minimum E_b/N_0 where the maximum acceptable BER or FER is achieved. The value of C should be chosen for this E_b/N_0. This value of C can be kept constant for all E_b/N_0 values for this code. For higher values of E_b/N_0, there will be neglible degradation in performance, even though C will be higher than optimal [7]. For lower E_b/N_0 values, there could be up to a few tenths of a dB degradation, since C will be lower than optimal. However, this should not have much impact since the BER or FER will already be above the maximum acceptable level anyway.

For fading channels the value of A and σ^2 should be averaged across the block to determine the average value of C. Each received value r_k should then be scaled by $(A\sigma^2)/(A_k\sigma_k^2)$ where A_k and σ_k^2 are the amplitude and normalised variance of r_k. Note that this scaling should be performed for both the log–MAP and max–log–MAP algorithms for optimal performance.

The value of A directly corresponds to the 6–bit signed magnitude inputs (shown in Table 5). The 6–bit inputs have 63 quantisation regions with a central dead zone. The quantisation regions are labelled from –31 to +31. For example, one could have A = 15.7. This value of A lies in quantisation region 15 (which has a range between 15 and 16).

Table 5: Quantisation for R0I to R4I.

Signed Magnitude		Two's Complement		Range
Dec.	**Binary**	**Dec.**	**Binary**	
31	011111	31	011111	30.5↔∞
30	011110	30	011110	29.5↔30.5
⋮	⋮	⋮	⋮	⋮
2	000010	2	000010	1.5↔2.5
1	000001	1	000001	0.5↔1.5
0	000000	0	000000	–0.5↔0.5
33	100001	63	111111	–1.5↔–0.5
34	100010	62	111110	–2.5↔–1.5
⋮	⋮	⋮	⋮	⋮
62	111110	34	100010	–30.5↔–29.5
63	111111	33	100001	–∞↔–30.5

Since most analogue to digital (A/D) converters do not have a central dead zone, a 7–bit A/D should be used and then converted to 6–bit as shown in the table. This allows maximum performance to be achieved.

For signed magnitude inputs a decimal value of 32 has a magnitude of 0 (equivalent to –0). For two's complement inputs, if 32 is input, this will be internally limited to 33 (equivalant to –31).

For input data quantised to less than 6–bits, the data should be mapped into the most significant bit positions of the input, the next bit equal to 1 and the remaining least significant bits tied low. For example, for 3–bit received data R0T[2:0], where R0T[2] is the sign bit, we have R0I[5:3] = R0T[2:0] and R0I[2:0] = 4 in decimal (100 in binary). For punctured input data, all bits must be zero, e.g., R1I[5:0] = 0.

Due to quantisation and limiting effects the value of A should also be adjusted according to the received signal to noise ratio.

Example 1: Rate 1/3 BPSK code operating at E_b/N_0 = 0.3 dB. From (2) we have σ^2 = 1.39988. Assuming A = 16 we have from (1) that C = 11 to the nearest integer.

Figure 3 gives a block diagram of the PCD03V eight state turbo decoder. The number of turbo decoder half–iterations is given by NI, ranging from 0 to 255. NI = 2I–1 where I is the number of iterations. This is equivalent to 0.5 to 128 iterations. The decoder initially starts at half iteration NA = 0, increasing by one until NI is reached or an earlier time if early stopping is enabled. The NA output can be used to select LIMZ and SCLZ values, especially for max–log–MAP decoding.

The turbo decoder speed f_d is given by

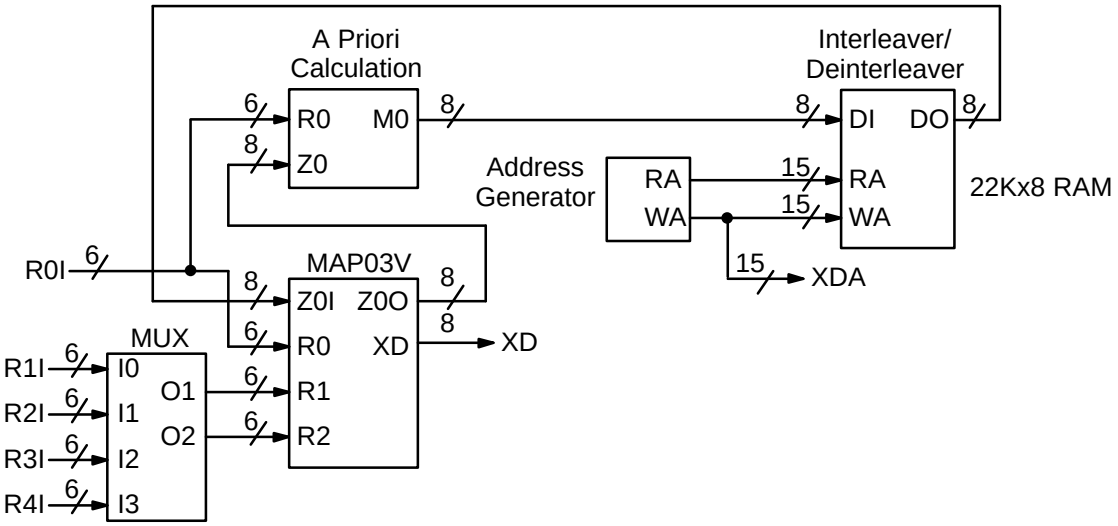

Figure 3: Simplified block diagram of PCD03V eight state turbo decoder.

$$f_d = \frac{F_d}{(NI + 1)(1 + L/K)} \qquad (3)$$

where F_d is the CLK frequency and L is the MAP decoder delay in bits (equal to either 138 or 266). The two delays indicate the sliding block length used in the MAP decoder, either 32 or 64, respectively. For short block lengths L = 138 should be used to increase decoder speed, while L = 266 should be used for larger block sizes to increase performance. This parameter can be selected with the SLD input.

For example, if F_d = 100 MHz and I = 5 (NI = 9) the decoder speed ranges from 2.2 Mbit/s for K = 40 and L = 138 to 9.5 Mbit/s for K = 5114 and L = 266.

An important parameter is LIMZ, the limit factors for the extrinsic information. Extrinsic information is the "correction" term that the MAP decoder determines from the received data and *a priori* information. It is used used as *a priori* information for the next MAP decoding or half iteration. By limiting the correction term, we can prevent the decoder from making decisions too early, which improves decoder performance.

The limit factor LIMZ should vary between 1 and 127, although we recommend that 96 be used.

Another parameter that can used to adjust decoder performance is SCLZ which ranges from 1 to 32. The extrinsic information is scaled by SCLZ/32. Thus, when SCLZ = 32, no scaling is performed. For log–MAP decoding we recommend SCLZ = 29. For max–log–MAP decoding we recommend SCLZ = 23. The NA output can be used to adjust LIMZ and SCLZ with the number of iterations for optimum performance.

There are four decoder operation modes given by M. Mode M = 0 decodes a received block with a fixed number of iterations (given by NI). Modes 1 to 3 are various early stopping algorithms. Early stopping is used to stop the decoder from iterating further once it has estimated there are zero errors in the block. Mode 1 will stop decoding after an odd number of half–iterations. Mode 2 will stop decoding after an even number of half iterations. Mode 3 will stop after either an odd or even number of half iterations. Further details are given in the next section.

3GPP2 Interleaver Programming

The PCD03V turbo decoder allows the option of programming the row constants that are used in the 3GPP2 interleaver. There are 32 constants, all of them being odd in value. Note that if K is the interleaver size, the maximum constant value must be less than 2^n, where $n = \lceil \log_2 K \rceil - 5$.

The maximum value of n is 10, so each constant can be represented by a 10 bit value. However, since all the parameters are odd, this implies the least significant bit (lsb) is always equal to one. Thus, only the nine most significant bits (msb) should be input, with the lsb being ignored. This is why the constant input CI[9:1] does not include the lsb CI0. For example, if the constant is 349, then the lsb should be deleted and $\lfloor 349/2 \rfloor = 174$ be input to CI[9:1].

Figure 4 shows an example of programming the 3GPP2 interleaver parameters. During the low to high transition of CLK, if CWE is high, the value at CI[9:1] is programmed into the internal memory at address location CA[4:0].

Note that K or CS can be changed in any order before, during or after programming the row con-

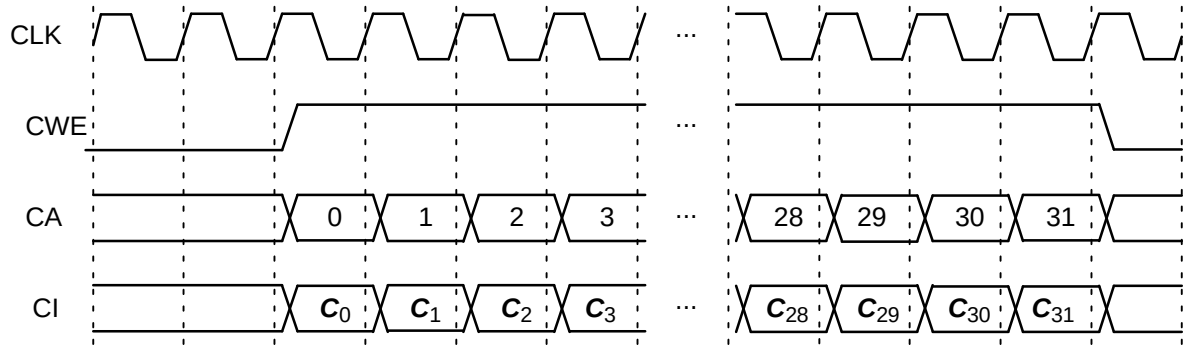

Figure 4: 3GPP2 Parameter Programming.

stants. As long as the correct *K* and CS are input to the decoder before decoding begins, the decoder will use the selected parameters. If CS is low, the internal 3GPP2 standard parameters are selected. If CS is high, the externally programmed parameters are selected.

3GPP™ LTE Interleaver

There are 188 standard interleaver sizes from 40 to 6144 bits. To select the internal parameters, set FS low and input the data length into K[12:0]. The decoder will automatically select the parameters for that length. Note that the only valid lengths are from 40 to 504 bits that are a multiple of 8, 512 to 1008 bits that are a multiple of 16, 1024 to 2016 bits that are multiple of 32, and 2048 to 6144 bits that are a multiple of 64. Other interleaver lengths will cause incorrect operation.

To input external interleavber parameters, set FS high. Any length from 40 to 6144 bits can be input, provided that *K* is a multiple of 8. Parameter F1I[8:1] is equal to f_1 divided by two. That is, the least significant bit of f_1 is deleted since it is always equal to one due to f_1 being odd. Similarly, parameter F2I[9:1] is equal to f_2 divided by two since f_2 is always even. For correct operation, $f_1 < 512$, $f_2 < 1024$, and $f_1+f_2 < 1024$.

Turbo Decoder Operation

After the START signal is sent, the decoder will read the received data at the CLK speed. It is assumed that the received data is stored in a synchronous read RAM of size (*K*+6)x6*n*, *n* = 2 to 5. The received data ready signal RR goes high to indicated the data to be read from the address given by RA[14:0]. Tables 6 and 7 illustrates which data is stored for address 0 to *K*–1 for the main data, *K* to *K*+2 for the first tail, and *K*+3 to *K*+5 for the second tail, for 3GPP™ and 3GPP2, respectively. The entries for the table indicate which encoded data output is selected, X, Y1 and Y2 for the first encoder and X′, Y1′ and Y2′ for the second encoder. The code polynomials are $g^0(D) = 1+D^2+D^3$ (13 in octal), $g^1(D) = 1+D+D^3$ (15) and $g^2(D) = 1+D+D^2+D^3$ (17). For rate 1/2 and 1/4 the main data is punctured, which is why two entries are shown.

Table 6: Input data format (3GPP™)

Rate	Data	Main	Tail 1	Tail 2
1/2	R0I	X X	X	X′
	R1I	Y1 Y1′	Y1	Y1′
1/3	R0I	X	X	X′
	R1I	Y1	Y1	Y1′
	R2I	Y1′	0	0
1/4	R0I	X X	X	X′
	R1I	Y1 Y1	Y1	Y1′
	R2I	Y2 Y1′	Y2	Y2′
	R3I	Y2′ Y2′	0	0
1/5	R0I	X	X	X′
	R1I	Y1	Y1	Y1′
	R2I	Y2	Y2	Y2′
	R3I	Y1′	0	0
	R4I	Y2′	0	0

The decoder then iteratively decodes the received data for *NI*+1 half iterations, rereading the received data for each half iteration for *K*+3 CLK cycles. The signal RR goes high for *K*+3 clock cycles while data is being output. Figure 5 illustrates the decoder timing where the data is input on the first half iteration.

The START signal is ignored during decoding, except for the last decoded bit that is output. A synchronous reset is also provided. All flip flops in the turbo decoder are reset during a low to high transition of CLK while RST is high.

The decoded block is output during the last half–iteration. The signal XDR goes high for *K* CLK cycles while the block is output. If *NI* is even, the block is output in sequential order. For *NI* odd, the block is output in interleaved order. To deinterleave the block, the output XDA[14:0] can be used

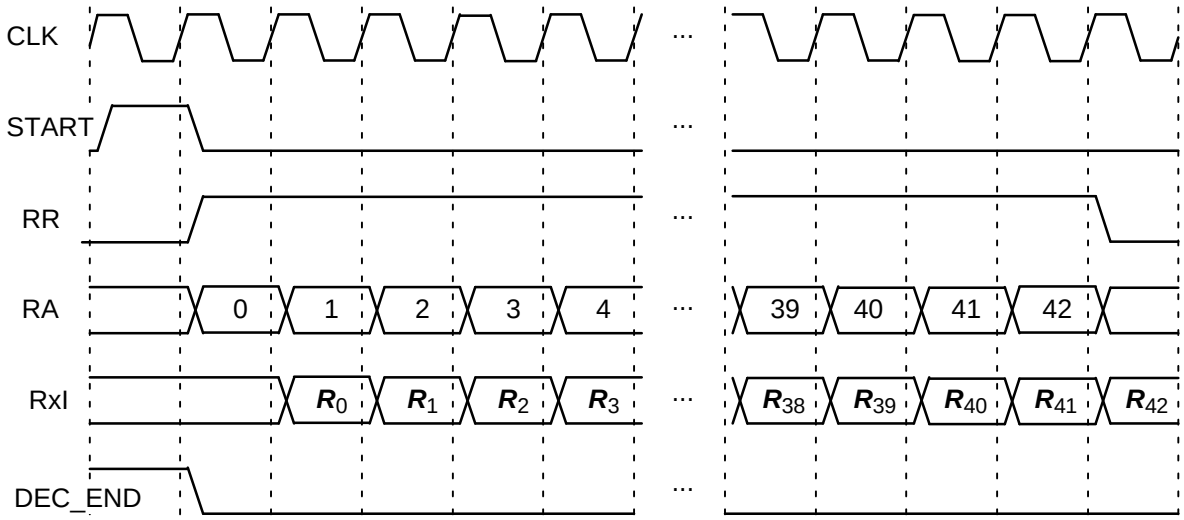

Figure 5: Turbo Decoder Input Timing (K = 40).

as the write address to a buffer RAM. After the block has been written to the buffer RAM, the decoded block can be sequentially read from the buffer RAM.

The signal ERR is a channel error estimator output. For even *NI*, it is the exclusive OR of XD and the sign bit of the input R0I. For odd *NI*, it is the exclusive OR of XD re–encoded to give the first parity bit and the sign bit of the input corresponding to Y1′ (this is because R0I is input in sequential order, not in the interleaved order of the output). If the output of the MAP decoder has zero errors, then this gives an approximation of the channel bit error rate (BER). Since Y1′ is punctured for rate 1/2 and 1/4, the number of bits counted is $\lfloor K/2 \rfloor$.

The DEC_END signal is low during decoding. At the end of decoding, DEC_END goes high. Figure 6 illustrates the decoder timing where data is output on the last half iteration. After startup, the maximum number of clock cycles for decoding is (*NI*+1)(*K*+*L*)+1.

During the last half iteration the decoded data is stored into the interleaver memory. Once decoding has been completed, the input XDE can be used to sequentially clock the decoded data from from the interleaver memory (regardless of the

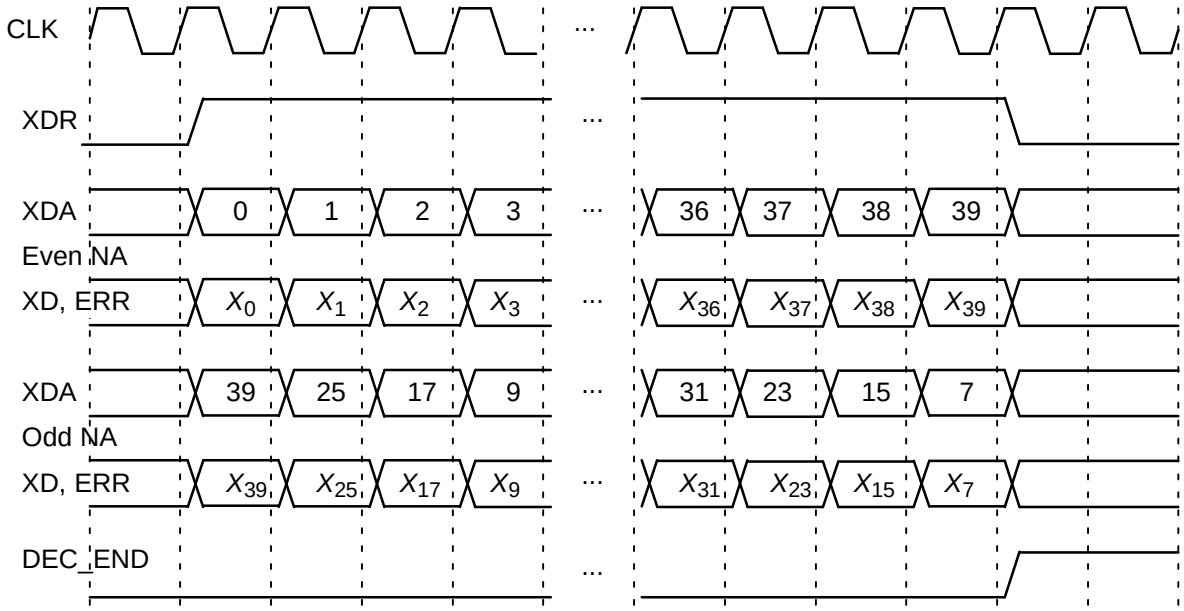

Figure 6: Turbo Decoder Output Timing (K = 40).

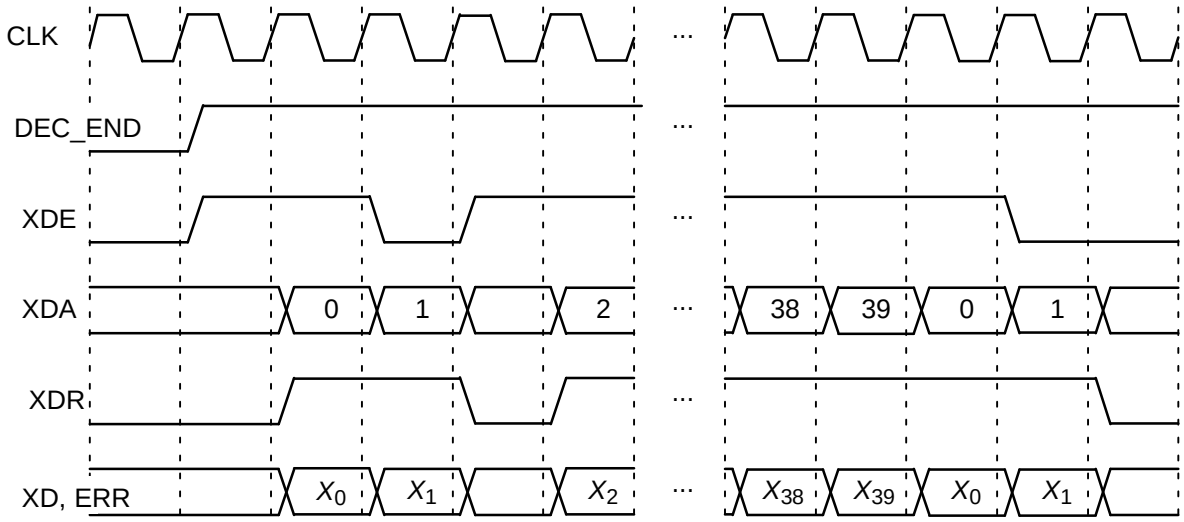

Figure 7: XDE Timing (K = 40).

number of iterations). XDE is disabled while the decoder is iterating. Figure 7 shows the decoder timing when XDE is used.

Table 7: Input data format (3GPP2)

Rate	Data	Main	Tail 1	Tail 2
1/2	R0I	X X	X	X′
	R1I	Y1 Y1′	Y1	Y1′
1/3	R0I	X	X	X′
	R1I	Y1	X	X′
	R2I	Y1′	Y1	Y1′
1/4	R0I	X X	X	X′
	R1I	Y1 Y1	X	X′
	R2I	Y2 Y1′	Y1	Y1′
	R3I	Y2′ Y2′	Y2	Y2′
1/5	R0I	X	X	X′
1xEV–DV	R1I	Y1	X	X′
	R2I	Y2	X	X′
	R3I	Y1′	Y1	Y1′
	R4I	Y2′	Y2	Y2′
1/5	R0I	X	X	X′
1xEV–DO	R1I	Y1	X	X′
	R2I	Y2	Y1	Y1′
	R3I	Y1′	Y2	Y2′
	R4I	Y2′	Y2	Y2′

The output ERR is also output when XDE goes high. The outputs RA and RR are used to read the sign bit of R0I which is exclusive–ORed with XD to give ERR.

The early stopping algorithm uses the magnitude of the extrinsic information to determine when to stop. As the decoder iterates, the magnitudes generally increases in value as the decoder becomes more confident in its decision. By comparing the smallest magnitude of a block with threshold ZTH, we can decide when to stop. If the smallest magnitude is greater than ZTH, i.e., not equal or less than ZTH, the decoder will stop iterating if early stopping has been enabled.

Since the last half iteration is used to store the decoded data into the interleaver memory, the decoder performs an extra half iteration once the threshold has been exceeded.

Increasing ZTH will increase the average number of iterations and decrease the BER. Decreasing ZTH will decrease the average number of iterations and increase the BER. In general, higher values of SNR will decrease the number of iterations. A value of ZTH = 23 was found to give a good trade off between the average number of iterations and BER performance.

For high SNR operation early stopping can lead to significantly reduced power consumption, since most blocks will be decoded in one or two iterations.

Simulation Software

Free software for simulating the PCD03V turbo decoder in additive white Gaussian noise (AWGN) or with external data is available by sending an email to info@sworld.com.au with “pcd03vsim request” in the subject header. The software uses an exact functional simulation of the PCD03V turbo decoder, including all quantisation and limiting effects.

After unzipping pcd03vsim.zip, there should be pcd03vsim.exe and code.txt. The file code.txt contains the parameters for running pcd03vsim. These parameters are

`m` Constituent code (CC) memory (2 to 4)
`nt` Number of turbo code outputs (2 to 5)
`g0` Divisor polynomial of CC in octal notation
`g1` 1st numerator polynomial of CC
`g2` 2nd numerator polynomial of CC
`EbNomin` Minimum E_b/N_0 (in dB)
`EbNomax` Maximum E_b/N_0 (in dB)
`EbNoinc` E_b/N_0 increment (in dB)
`optC` Input scaling parameter (0.37 for rate 1/3)
`ferrmax` Number of frame errors to count
`Pfmin` Minimum frame error rate (FER)
`Pbmin` Minimum bit error rate (BER)
`NI` Number of half iterations–1 (0 to 255)
`SLD` MAP decoder delay select (0 or 1)
`LIMZ` Extrinsic information limit (1 to 127)
`SCLZ` Extrinsic information scale (1 to 32)
`M` Stopping mode (0 to 4)
`ZTH` Extrinsic info. threshold (0 to 127)
`q` Number of quantisation bits (1 to 6)
`LOGMAP` Log–MAP decoding (MODE0, 0 or 1)
`C4PIN` Use five–bit *C* (MODE1, 0 or 1)
`enter_C` Use external value of C (y or n)
`C` *C* (0 to 17)
`K` Block length (40 to 5114 for 3GPP™ or 17 to 21504 for 3GPP2)
`S3G` Standard select (0 to 3)
`CS` Use external 3GPP2 interleaver parameters (0 or 1)
`fs` LTE interleaver parameter select (0 or 1)
`f1` LTE interleaver parameter 1 (1 to 511)
`f2` LTE interleaver parameter 2 (0 to 1022)
`state` State file (0 to 2)
`s1` Seed 1 (1 to 2147483562)
`s2` Seed 2 (1 to 2147483398)
`out_screen` Output data to screen (y or n)
`read_x` Use external information data (y or n)
`read_r` Use external received data (y or n)
`out_dir` Output directory
`in_dir` Input directory

Note that g0, g1 and g2 are given in octal notation, e.g., g0 = 13 $\equiv 1011_2 \equiv 1+D^2+D^3$. For the 3GPP™ standard, m = 3, nt = 3, g0 = 13 and g1 = 15 (g2 is not used). The nominal turbo code rate is 1/nt.

An optional Genie aided stopping mode can be selected by setting M = 4. This will stop the decoder from further iterations when the Genie has detected there are no errors compared to the transmitted data. This allows a lower performance bound to be simulated, allowing fast simulations for various configurations at low bit error rates.

The parameter `optC` is used to determine the "optimum" values of *A* and *C*. The value of *A* is

$$A = \frac{\text{optC}(2^{q-1} - 1)}{\text{mag}(\sigma)} \tag{4}$$

where σ^2 is the normalised noise variance given by (2) and mag(σ) is the normalising magnitude resulting from an auto–gain control (AGC) circuit. With log–MAP decoding, we recommend using `optC` = 0.37 for rate 1/3 and `optC` = 0.58 for rate 1/2. We have

$$\text{mag}(\sigma) = \sigma\sqrt{\frac{2}{\pi}}\exp\left(\frac{-1}{2\sigma^2}\right) + 1 - 2Q\left(\frac{1}{\sigma}\right) \tag{5}$$

where $Q(x)$ is the error function given by

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-t^2}{2}\right)dt. \tag{6}$$

For the "optimum" *A*, if `enter_C` = n, we round the value of *C* given by (1) to the nearest integer. If LOGMAP = MODE0 = 0 then *C* is forced to 0. If LOGMAP = 1 and C4PIN = MODE1 = 0, *C* is limited to a maximum value of 9. If LOGMAP = 1 and C4PIN = 1, *C* is limited to a maximum value of 17. If `enter_C` = y, a constant value of *C* is used for all SNR values. For the recommended `optC` values, *C* = 6 can be used for all SNR values with little degradation in performance.

The simulation will increase E_b/N_0 (in dB) in `EbNoinc` increments from `EbNomin` until `EbNomax` is reached or the frame error rate (FER) is below or equal to `Pfmin` or the bit error rate (BER) is below or equal to `Pbmin`. Each simulation point continues until the number of frame errors is equal to `ferrmax`. If `ferrmax` = 0, then only one frame is simulated.

When the simulation is finished the output is given in, for example, file k512.dat, where *K* = 512. The first line gives the E_b/N_0 (`Eb/No`), the number of frames (`num`), the number of bit errors in the frame (`err`), the total number of bit errors (`berr`), the total number of frame errors (`ferr`), the average number of iterations (`na`), the average BER (`Pb`) and the average FER (`Pf`). Following this, the number of iterations, `na`, `berr`, `ferr`, `Pb`, and `Pf` are given for each half iteration.

The following file was used to give the simulation results shown in Figure 8 with log–MAP decoding. Auto–stopping was used with up to 10 iterations. When iterating is stopped early, the `nasum` (`2*num*na`), `berr` and `ferr` results at stopping are copied for each half iteration to the maximum iteration number. This allows the performance to be obtained at each half iteraion. Figure 9 shows the average number of iterations with E_b/N_0. Figure 10 shows the log–MAP performance with *NI* = 19, *n* = 2 and 3, and *K* = 256, 512 and 1024.

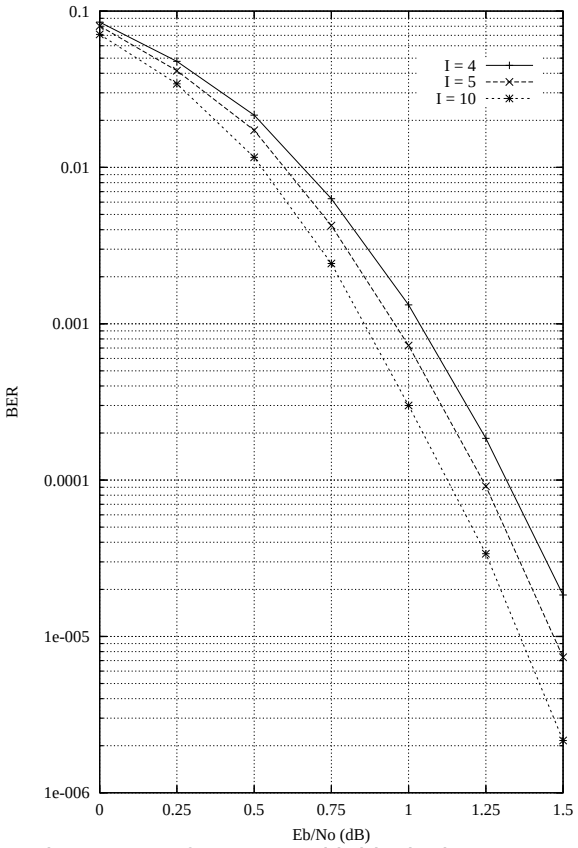

Figure 8: Performance with block size 512, 3GPP™ and auto–stopping (ZTH = 23).

```
{m nt g0 g1 g2}
 3 3  13 15 17
{EbNomin EbNomax EbNoinc optC}
 0.0     1.5     0.25    0.37
{ferrmax Pfmin Pbmin}
 128     1e-99 1e-5
{NI  SLD LIMZ  SCLZ  M  ZTH}
 19  1   96    29    3  23
{q  LOGMAP  C4PIN  enter_C  C}
 6  1       1      y        6
{K   S3G  CS  fs  f1  f2}
 512 0    0   0   3   10
{state s1    s2     out_screen}
```

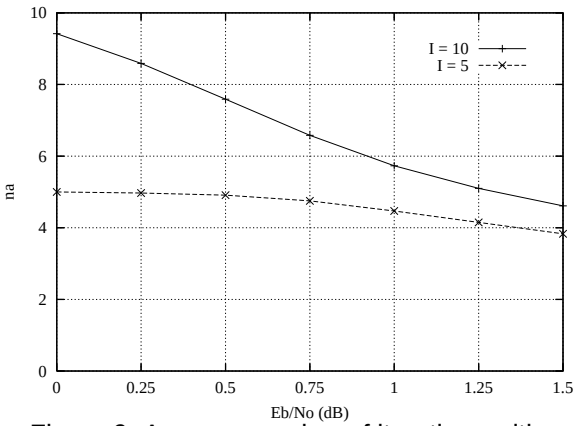

Figure 9: Average number of iterations with block size 512 and auto–stopping (ZTH = 23).

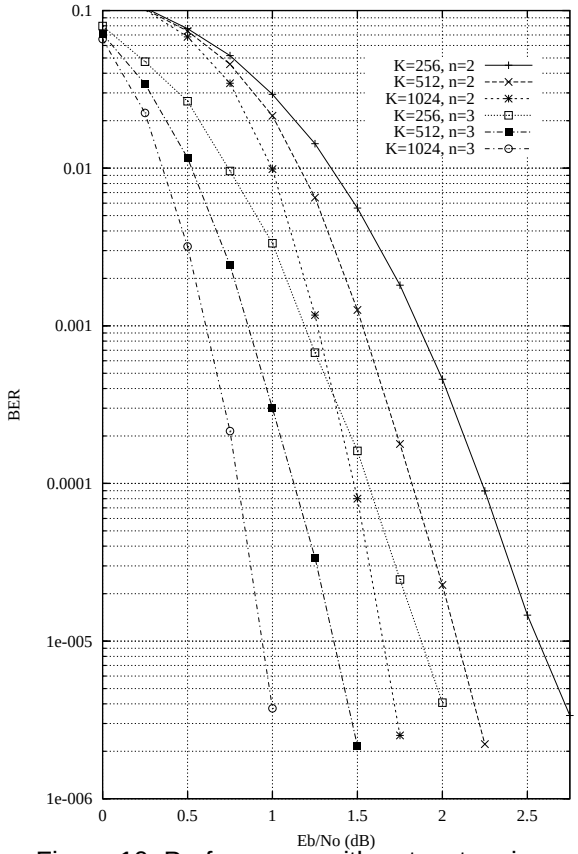

Figure 10: Performance with auto–stopping and 10 iterations.

```
 0     12345 67890  y
{read_x  read_r  out_dir  in_dir}
 n       n       output   input
```

The `state` input can be used to continue the simulation after the simulation has been stopped, e.g., by the program being closed or your computer crashing. For normal simulations, `state` = 0. While the program is running, the simulation state is alternatively written into State1.dat and State2.dat. Two state files are used in case the program stops while writing data into one file. To continue the simulation after the program is stopped follow these instructions:

1) Copy the state files State1.dat and State2.dat. This ensures you can restart the program if a mistake is made in configuring code.txt.

2) Examine the state files and choose one that isn't corrupted.

3) Change the state parameter to 1 if State1.dat is used or 2 if State2.dat is used.

4) Restart the simulion. The output will be appended to the existing k(K).dat file.

5) After the simulation has been completed, make sure that `state` is changed back to 0.

The software can also be used to encode and decode external data. To encode a block

`X_(K).dat` in the directory given by `in_dir`, set `read_x` to `y`, e.g., `X_512.dat` in directory `input` (each line contains one bit of data). The encoded stream `Y_(K).dat` will be output to the directory given by `out_dir`, e.g., `Y_512.dat` to directory `output`.

To decode data, place the received block of data in file `R_(K).dat` in directory `in_dir` and set `read_r` to `y`. The decoded data is output to `XD_(K).dat` in directory `out_dir`. For 3GPP™ `R_(K).dat` has in each line R[i,j], i = 0 to nt–1 from j = 0 to *K*–1 and then i = 0 to n–1 from j = *K* to *K*+2*m*–1. For 3GPP2 `R_(K).dat` has in each line R[i,j], i = 0 to nt–1 from j = 0 to *K*+2*m*–1. e.g., for nt = 3 the first three lines could be

–31 1 –25
–31 12 9
11 31 31

The input data is of the form

R[i,j] = A*(1–2*Y[i,j]+N[i,j])

where A is the signal amplitude, Y[i,j] is the coded bit, and N[i,j] is white Gaussian noise with zero mean and normalised variance σ^2. The magnitude of R[i,j] should be rounded to the nearest integer and be no greater than 2^{q-1}–1. If `read_r` = `y`, then *C* is externally input via `c`.

For 3GPP2, external interleaver parameters can be used. Set `cs = 1` and place the 32 interleaver parameters, one parameter per line, in file `c.dat` in directory `in_dir`. The full value should be input, and not the value with the lsb removed.

Viterbi Decoder Operation

The Viterbi decoder is operated in a similar way to the turbo decoder. The START signal is used to start decoding, using RR and RA to read the received data. For rate 1/2 operation, R2I to R4I are not used. For rate 1/3 operation R3I to R4I are not used. For rate 1/4 R4I is not used.

The input SM selects 16, 32, 64 or 256 states (constraint lengths 5, 6, 7 or 9, memory *m* = 4, 5, 6 or 8). For TB = 0, the input DELAY = 0, 1 and 2 selects a delay of *D* = 64+*m*, 128+*m* and 256+*m*, respectively. The number of states is equal to 2^m and the constraint length is equal to *m*+1. For TB = 1, the delay is *D* = 128+*m,* 256+*m*, and 512+*m* for DELAY = 0, 1, and 2, respectively. This assumes that the encoder start state is equal to the last *m* bits of the data block. Note that MODE[6:5] > 1 for 256 states and DELAY = 2.

Table 8 shows the codes selected with the number of states, code rate, and 3G standard (selected by S3G[2]).

Table 8: Convolutional Codes.

S3G [2]	SM [1:0]	N[1:0]	g0	g1	g2	g3
X	00	10	23	35	–	–
X	00	11	25	33	37	–
X	00	00	23	35	25	37
X	01	10	51	67	–	–
X	01	11	51	67	75	–
X	01	00	51	55	67	77
0	10	10	133	171	–	–
0	10	11	133	171	165	–
0	10	00	173	167	135	111
0	11	10	561	753	–	–
0	11	11	557	663	711	–
0	11	00	473	513	671	765
1	10	10	171	133	–	–
1	10	11	171	133	165	–
1	10	00	173	167	135	111
1	11	10	753	561	–	–
1	11	11	557	663	711	–
1	11	00	765	671	513	473

For TB = 0, the decoder first inputs the received data from address 0 to *K*–1. The tail is then input from address *K* to *K*+*m*–1. After a decoding delay, the decoded data is output to XD. XDR goes high for one clock cycle at the beginning of each decoded bit. XDA goes from address 0 to *K*–1 as the decoded data is output.

For TB = 1, the input sequence is more complicated. There is an additional delay of *T* = 64, 128 or 256 for DELAY = 0, 1 or 2, respectively. The data is input for *T* symbols from address –*T* mod *K* to *K*–1, for *K* symbols from address 0 to *K*–1, and then for *T* symbols from address 0 to *T*–1 mod *K*. The decoder automatically calculates the correct address for RA, so the data only needs to be stored from address 0 to *K*–1. Data lengths available for tail biting decoding are restricted from *m* to 2048 bits.

The output ERR is the exclusive–OR of the sign bit of R0I with the corresponding re–encoded decoded output bit. This allows an estimate of the channel BER.

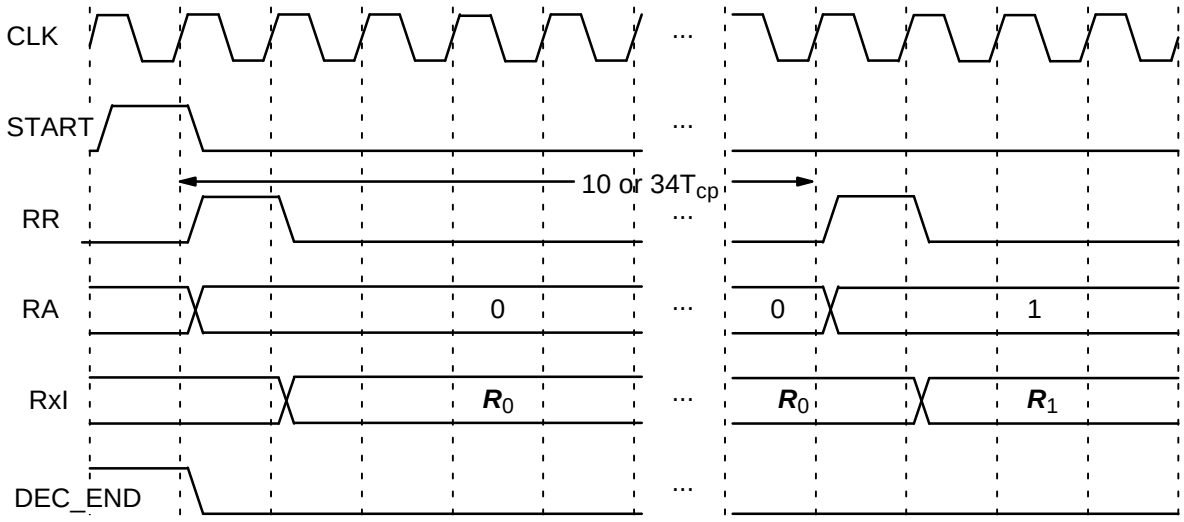

Figure 11: Viterbi Decoder Input Timing.

Figure 11 shows the Viterbi decoder input timing. Two clock cycles are used to start decoding, with each decoded bit taking 10 clock cycles for 16, 32 or 64 states or 34 clock cycles with 256 states.

Figure 12 shows the Viterbi decoder output timing. The input XDE is not used either during or after Viterbi decoding.

The decoding speed is given by

$$f_d = \frac{F_d}{N_c(1 + D/K) + 2/K} \qquad (7)$$

where F_d is the internal clock speed, N_c is the number of decoder clock cycles (10 or 34) and D is the Viterbi decoder delay in bits. For example, if TB = 0, K = 504 (the maximum in 3GPP™), D = 136 (SM = 3, DELAY = 1), N_c = 34 (SM = 3) and F_d = 100 MHz, decoding speed is 2.3 Mbit/s.

Ordering Information

SW–PCD03V–SOS (SignOnce Site License)
SW–PCD03V–SOP (SignOnce Project License)
SW–PCD03V–VHD (VHDL ASIC License)

All licenses include Xilinx VHDL cores. The above licenses do no include the Viterbi decoder which must be ordered separately (see the VA08V data sheet). The SignOnce and ASIC licenses allows unlimited instantiations and free updates for one year.

Note that *Small World Communications* only provides software and does not provide the actual devices themselves. Please contact *Small World Communications* for a quote.

References

[1] Third Generation Partnership Project (3GPP), “Universal mobile telecommunications system (UMTS); Multiplexing and channel coding (FDD),” 3GPP TS 25.212 V5.2.0 Release 5, Sep. 2002.

[2] Third Generation Partnership Project (3GPP), “Evolved universal terrestrial radio access (E–UTRA); Multiplexing and channel

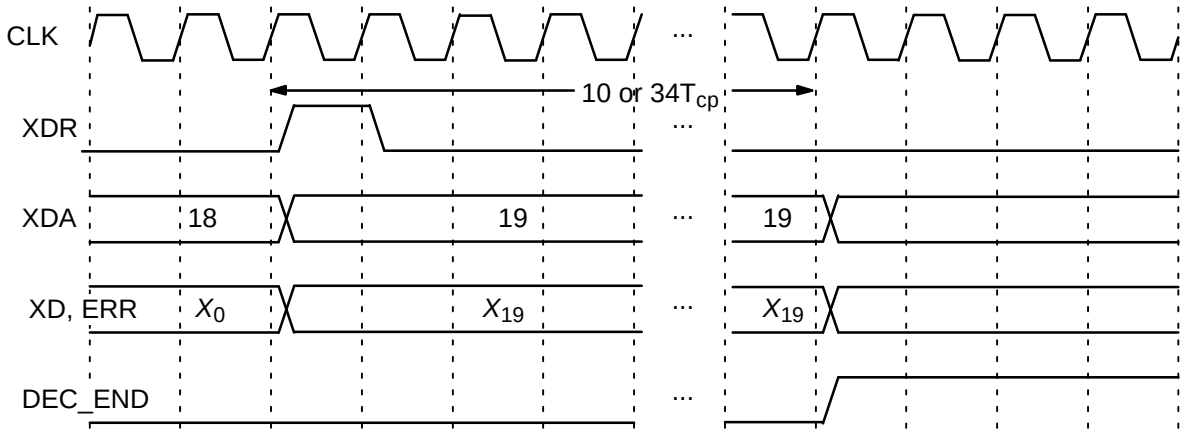

Figure 12: Viterbi Decoder Output Timing (K = 20).

coding," 3GPP TS 36.212 V8.1.0 Release 8, Nov. 2007.
[3] Third Generation Partnership Project 2 (3GPP2), "Physical layer standard for cdma2000 spread spectrum systems, Release D," 3GPP2 C.S0002–D Version 2.0, Sep. 2005.
[4] Third Generation Partnership Project 2 (3GPP2), "cdma2000 high rate packet data air interface specification, Release B," 3GPP2 C.S0024–B Version 2.0, Mar. 2007.
[5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT–20, pp. 284–287, Mar. 1974.
[6] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub–optimal MAP decoding algorithms operating in the log domain," *ICC'95*, Seattle, WA, USA, pp. 1009–1013, June 1995.
[7] M. C. Reed and J. A. Asenstorfer, "A novel variance estimator for turbo–code decoding," *Int. Conf. on Telecommun.*, Melbourne, Australia, pp. 173–178, Apr. 1997.

Small World Communications, 6 First Avenue, Payneham South SA 5070, Australia.
info@sworld.com.au ph. +61 8 8332 0319
http://www.sworld.com.au

Version History

- 0.0 23 Nov. 2000. PCD03 preliminary product specification
- 0.4 1 Mar. 2001. Added C4PIN and LOGMAP input. Added XDA[12:0] and ERR output. Deleted output decoded data option. Added MAP decoder delay option. Corrected interleaver procedure. Added decoder speed and complexity for Virtex–E. Added simulation software and performance curves for K=600. Added BIT and MCS files section. Defined MAP decoder delay as 136 or 264.
- 0.5 31 May 2001. Added START and RST input. Corrected number of bits in microprocessor address input MA[1:0] to MA[5:0]. Updated speed and complexity for Virtex–E. Reduced number of decoder clock cycles by one. Moved turbo pascal procedures to Appendix.
- 0.6 5 Aug. 2001. Added 256 state Viterbi decoder option. Added XDE input.
- 0.62 22 Sep. 2001. Added external information and received data input options for simulation software.
- 0.63 8 Apr. 2002. Corrected 2.5 iteration performance figure.
- 0.70 25 June 2002. Deleted microprocessor interface. Added rate 1/2, 1/4 and 1/5 code rate and extrinsic information scaling options. Added automatic generation of interleaver parameters. Added 64 state and rate 1/4 Viterbi decoder options. Added performance figure for K = 200, 500 and 1000 for rate 1/2 and 1/3. Deleted BIT and MCS files section and Appendix. Added France Telecom contact information.
- 1.00 13 Jan. 2003. Official release. Added 3GPP2 1xEV–DV option. Added interleave size options of 4K, 5K, 20K and 21K. Added Virtex–II performance and complexity.
- 1.12 12 June 2003. Added interleaver lengths in Signal Descriptions. Corrected Table 2 title. Corrected turbo code polynomial descriptions.
- 1.2 19 Aug. 2003. Updated decoder speed and complexity. Corrected explanation of inputs for Viterbi decoder.
- 1.4 8 June 2004. Updated Virtex–E and Virtex–II and added Spartan–3 speed and complexity. Deleted rate 1/5 1xEV–DV resources used and decoder mode. Decreased 3GPP2 interleaver minimum number of columns from 16 to 8 and interleaver length from 33 to 17. Added VHDL

ASIC core availability. Changed licensing options to from Yearly Unlimited and Basic to SignOnce Site, SignOnce Project, University and VHDL ASIC. Added rate 1/2 Viterbi decoder option for 3GPP™/VA Decoder Mode. Corrected 3GPP2 tail for 1xEV–DV.

- 1.41 18 Jan. 2005. Changed SCLZ recommendation for log–MAP and max–log–MAP from 32 and 26 to 29 and 23.
- 1.42 19 May 2005. Added Virtex–II Pro and Virtex–4 speed and complexity.
- 1.43 10 June 2005. Added description of received data file for simulation software.
- 1.5 5 May 2006. Added 3GPP2 1xEV–DO turbo decoder option. Corrected Virtex–4 Xilinx part description.
- 1.52 4 April 2007. Updated Virtex–II and Virtex–4 complexity. Deleted 1xEV–DV and 1xEV–DO descriptions in MODE selection.
- 1.60 29 Mar. 2008. Added tail biting, code selection and 32 and 64 state Viterbi decoder options. Added two's complement input option. Added 3GPP2 programmable interleaver parameter option. Deleted Virtex–E and Virtex–II speed and complexity.
- 1.7 24 Apr. 2008. Added 3GPP™ LTE/E–UTRA turbo decoder option. Added Actel, Lattice and Altera FPGA availability. Changed 5K and 21K interleaver options to 6K and 22K.
- 1.72 6 May 2008. Changed LTE/E–UTRA description to LTE.
- 1.74 4 July 2008. Changed Viterbi decoder input from 4 to 6 bits. Deleted Altera FPGA availability. Updated Spartan–3, Virtex–II Pro and Virtex–4 and added Virtex–5 speed. Complexity changed from Virtex–II Pro and Spartan–3/Virtex–4 to Virtex–4 with updated values. Corrected reference to Viterbi decoder output timing.
- 1.76 12 Sep. 2008. Added MODE[8:7] for 3GPP2/UMTS and 3GPP2/LTE implementation options. Changed DELAY[1:0] input in schematic symbol from pin to bus. Added 1xEV–DV description for resources used. Added Virtex–5 complexity.
- 1.77 9 Oct. 2008. Updated speed and complexity.
- 1.80 8 Apr. 2009. Increased max–log–MAP decoder speed from 28% to 44% increase over log–MAP. Increased number of clock cycles per half iteration by one. Updated complexity. Added Altera FPGA availability.
- 1.82 17 June 2010. Deleted Virtex–II Pro and added Virtex–6 and Spartan–6 speed. Corrected MODE input for 1xEV–DO decoder mode and two's complement decimal representation for input quantisation. Deleted 3GPP™ and 3GPP2 description in resources used. Redefined MAP decoder delay to include start delay of two clock cycles.
- 1.83 8 Dec. 2010. Changed 20K interleaver option to 12K. Replaced two BlockRAMs used in UMTS interleaver generator with four 256x8 slice memories. Improved Virtex–5, Virtex–6 and Spartan–6 speed. Updated Virtex–4 and Virtex–5 complexity. Changed simulation software description from pcd03sim to pcd03vsim. Added version history.
- 1.84 25 Jan. 2011. Updated input data format. Corrected decoding time. Added Genie aided early stopping mode for BER simulation software.
- 1.85 10 Aug. 2011. Corrected description of data input addressing for tail biting Viterbi decoding.
- 1.86 22 Dec. 2016. In Table 1 deleted Virtex 4 LUTs and updated 6–Input LUTs resources. In Table 2 deleted Spartan 3, Spartan 6 and Virtex 4 performance. Added Artix–7, Kintex–7 and Zynq–7 performance. Changed turbo decoder performance from log–MAP to max–log–MAP. Replaced Figures 8 and 9 with K = 512 performance. Replaced Figure 10 with K = 256, 512 and 1024 performance.
- 1.87 18 Apr. 2017. Updated resources used.
- 1.88 28 Apr. 2017. Updated resources used.
- 1.89 2 June 2017. Updated performance and resources used.
- 1.90 14 July 2017. Updated performance and resources used.
- 1.91 8 Nov. 2017. Updated performance and resources used. Decreased startup delay by one clock cycle. Added 3GPP™ implementation mode for MODE[8:7]. Increased Viterbi decoder tail biting data range. Reordered simulation parameters.
- 1.92 14 Nov. 2017. Corrected D for Viterbi decoding and TB=1.
- 1.93 1 Dec. 2017. Updated speed and complexity.
- 1.94 27 Dec. 2017. Updated complexity with Viterbi decoder.
- 1.95 22 Apr. 2018. Added SM = 3 and DELAY = 2 Viterbi decoding for CODE[6:5] > 1. Updated speed and complexity.