PCE04CH Features

- 16 state CCSDS compatible turbo encoder
- Rate 1/2, 1/3, 1/4 and 1/6
- Interleaver sizes from 1784 to 16056 bits in multiples of 1784.
- Includes sync marker, optional pseudo–randomiser and ping–pong input memory
- Up to 706 MHz internal clock
- Up to 346 Mbit/s encoding speed
- Serial continuous encoded data out
- 254 or 216 6–input LUTs with 1 or 2 18kB RAMB18s, respectively
- Asynchronous logic free design
- Available as EDIF and VHDL core for Xilinx FPGAs under SignOnce IP License. ASIC, Intel/Altera, Lattice and Microsemi/Actel cores available on request.

Introduction

The PCE04CH is a 16 state CCSDS [1] compatible turbo encoder. Interleaver sizes from 1784 to 16056 bits in multiples of 1784 can be implemented. Turbo code rates of R = 1/2, 1/3, 1/4 and 1/6 can be selected. The un–interleaved data is terminated with a tail using both data and parity information. The interleaved data is terminated with a tail using parity data only. The input block and interleaver size is K. The number of coded bits is $n(K+4)$ where the nominal code rate is $1/n$.

To allow a continuous output stream, a ping–pong input memory is used to buffer the input data to be encoded. The output consists of $32n$ sync marker bits, followed by $n(K+4)$ optionally pseudo–randomised coded bits. Total transmitted length is $n(K+36)$.

Figure 1 shows the schematic symbol for the PCE04CH encoder. The EDIF core can be used with Xilinx Foundation or Integrated Software Environment (ISE) software. The VHDL core can be used with Xilinx ISE or Vivado software. Custom VHDL cores can be used in ASIC designs.

Table 1 shows the performance achieved for various Xilinx parts for K = 1784 and MODE = 1. T_{cp} is the minimum clock period over recommended operating conditions. These performance figures may change due to device utilisation and configuration. Note that Zynq devices up to XC7Z020 and from XC7Z030 use programmable logic equivalent to Artix–7 and Kintex–7 devices, respectively.

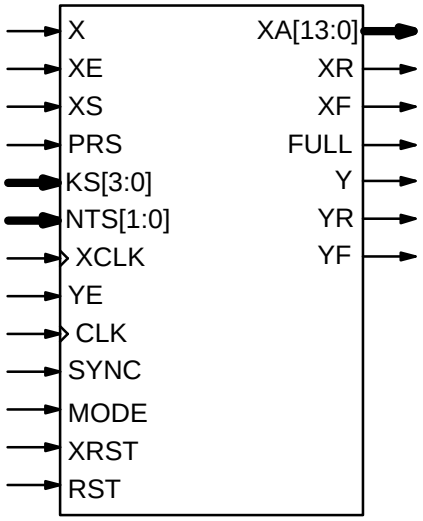

Figure 1: PCE04CH schematic symbol.

Table 1: Example performance

Part	T_{cp} (ns)	Speed (Mbit/s)			
		1/2	1/3	1/4	1/6
XC7S6C–1	4.956	98.8	65.9	49.4	32.9
XC7S6C–2	4.064	120.5	80.3	60.2	40.1
XC7A12T–1	4.854	100.9	67.3	50.4	33.6
XC7A12T–2	3.991	122.8	81.8	61.4	40.9
XC7A12T–3	3.569	137.3	91.5	68.6	45.7
XC7K70T–1	3.053	160.5	107.0	80.2	53.5
XC7K70T–2	2.434	201.3	134.2	100.6	67.1
XC7K70T–3	2.267	216.1	144.1	108.0	72.0
XCKU035–1	2.597	188.7	125.8	94.3	62.9
XCKU035–2	2.147	228.2	152.1	114.1	76.0
XCKU035–3	1.896	258.4	172.3	129.2	86.1
XCKU3P–1	1.833	267.3	178.2	133.6	89.1
XCKU3P–2	1.527	320.9	213.9	160.4	106.9
XCKU3P–3	1.416	346.1	230.7	173.0	115.3

Signal Descriptions

CLK Encoder Clock
FULL Encoder Full (new data not accepted)
KS Data Length Select
0 = Length 1784 (CCSDS)

	1 = Length 3568 (CCSDS)
	2 = Length 7136 (CCSDS)
	3 = Length 8920 (CCSDS)
	4 = Length 5352
	5 = Length 10704
	6 = Length 12488
	7 = Length 14272
	8 = Length 16056
MODE	0 = CCSDS interleaver (1784 to 8920)
	1 = large interleaver (1784 to 16056)
NTS	Turbo Code Rate Select (0 to 3)
	0 = Rate 1/2
	1 = Rate 1/3
	2 = Rate 1/4
	3 = Rate 1/6
PRS	Pseudo Randomiser Select
RST	Synchronous Reset for CLK
X	Data In
XA	Data In Address
XCLK	Data In Clock
XE	Data in Enable
XF	Data in Finish
XR	Data In Ready
XRST	Synchronous Reset for XCLK
XS	Data in Start
SYNC	XLCK and CLK equal
Y	Data Out
YE	Data Out Enable
YF	Data Out Finish
YR	Data Out Ready

Table 2 gives the encoder complexity with MODE. With MODE = 0, a 2KX9 RAMB18 is used for the input memory. This requires dividing the read address by 9, which increases the number of 6–input LUTs as well as decreasing the encoder speed by about 20%.

Note that SYNC and MODE are “soft” inouts and should not be connected to input pins or logic. These inputs are designed to minimise encoder complexity for the configuration selected.

Table 2: Encoder complexity

MODE	LUT	RAMB18
0	254	1
1	216	2

Encoder Operation

Figure 2 gives a block diagram of the PCE04CH CCSDS turbo encoder. X is the input the turbo encoder. $Y[5:0] = Y^5$ down to Y^0 is the six bit parallel output from the encoder. Y^0 corresponds to the K data sequence and $m = 4$ bit tail, where m is the encoder memory. Y^1, Y^2 and Y^3 correspond to the parity output from encoding X. Y^4 and Y^5 correspond to the parity output from encoding the interleaved sequence X'. Each sequence is of length $K+m$.

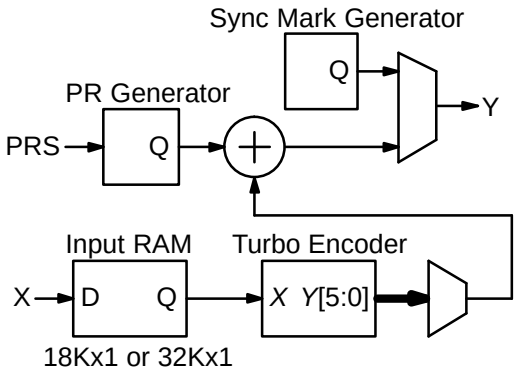

Figure 2: PCE04CH turbo encoder

The data to the turbo encoder is input from the Input RAM in the sequence $X_k\ X_{I(k)}$ where X_k is the data at time k from 0 to K–1 and $I(k)$ is the interleaved address. Table 3 shows the output sequence for the various code rates. For rate 1/3, 1/4 and 1/6, k is incremented by one from 0 to K–1. For rate 1/2, k is incremented by two.

Note that the output for all rates is output in one continuous stream. The encoder does not pause in outputting the data, unless YE is held low. The tail bits are output in the same sequence as the data sequence.

Table 3: Output sequence

Rate	Sequence
1/2	$Y_k^0\ Y_k^1\ Y_{k+1}^0\ Y_{k+1}^4$
1/3	$Y_k^0\ Y_k^1\ Y_k^4$
1/4	$Y_k^0\ Y_k^2\ Y_k^3\ Y_k^4$
1/6	$Y_k^0\ Y_k^1\ Y_k^2\ Y_k^3\ Y_k^4\ Y_k^5$

Encoding

The input data X is written one bit at a time into one half of the Input RAM. The other half of the memory has input data read into the encoder in each clock cycle. Input MODE can be to select either an 18Kx1 (MODE = 0) or 32Kx1 (MODE = 1) input memory. To minimise complexity MODE should not be connected to logic or input pins.

The FULL output indicates when the encoder can accept data. When high this indicates that new data must not be input to the encoder in the next clock cycle. That is, XS and XE must remain low while FULL is high.

If FULL is low, the data start XS signal is used to start the encoder. The data enable input XE must also go high when XS goes high to read the

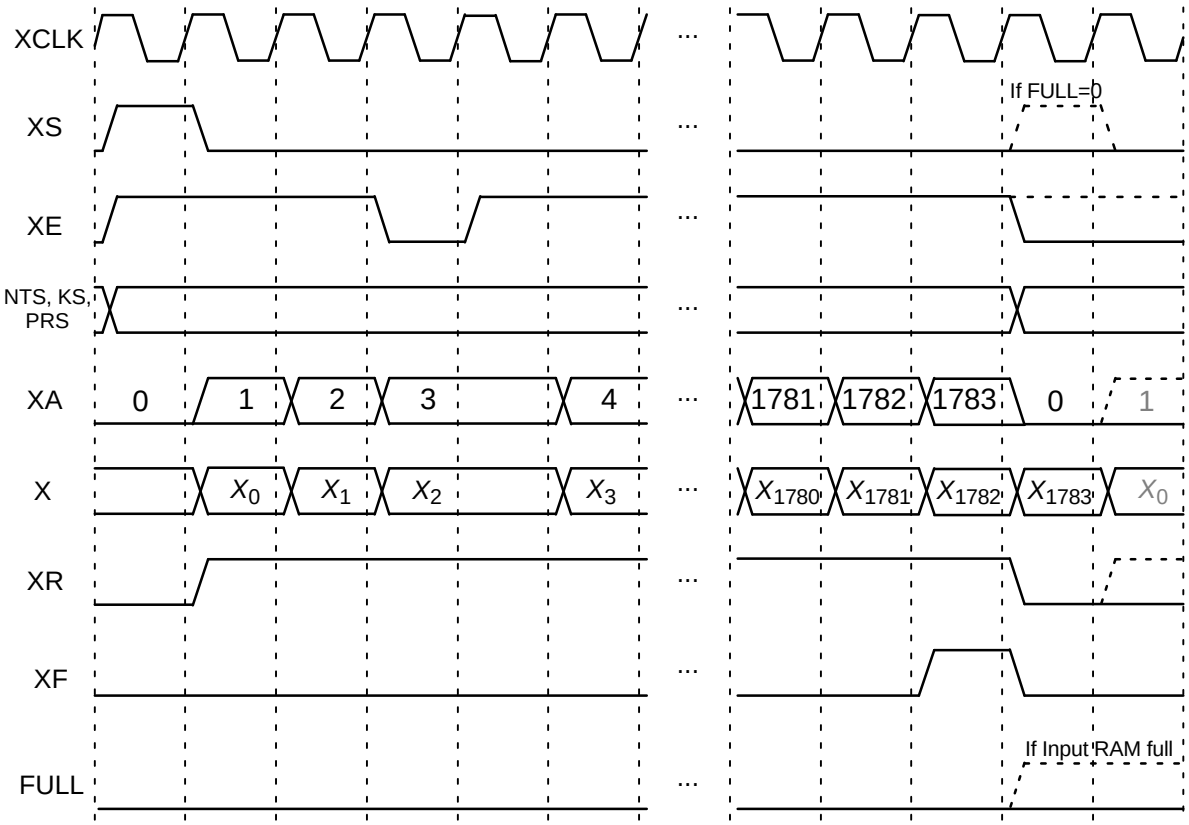

Figure 3: Turbo encoder input timing (K = 1784).

first received data. XE must only go high once for each input data bit. This means XE can only be high for K clock cycles.

The data ready output XR will go high to indicated that XE can now go high so as to input the next data bit. XR will stay high until one clock cycle before the last data is input. XS and XR can be ORed together to form XE if the input data is stored in an external memory.

To ensure that a continuous output is formed, The total time to input the data must not exceed the time to output an encoded stream. That is $LT_X < n(K+36)T_C$ where $L \geq K$ is the number of clock cycles that XS and XR are high, T_X is the XCLK period and T_C is the CLK period. If XCLK and CLK are the same, a simple way to ensure this is that XE must not remain low for each data bit more than n–1 clock cycles.

Valid data must be input one clock cycle after XE goes high. A data address output XA[13:0] is provided for reading data from an external synchronous read input memory. Data read from the input memory must be held if XE goes low as shown in Figure 3.

The data finish output XF will go high at the end of each block while XA = K–1. If XE goes low at XA = K–1, XF will remain high. XF will go low only after XE goes high.

If the other half of the Input RAM is available, FULL will remain low, indicating that the next block may be input. If both halves of the RAM are full, then FULL will go high. FULL will not go low again until one of the halves of the RAM becomes available. If FULL is low, XS and XE can go high. To ensure a continuous output, data should be input as soon as possible after FULL goes low.

Inputs X, XS, XE, PRS, NTS[1:0], and KS[3:0] must be synchronous to XCLK. Outputs XA[13:0], XR, XF and FULL are synchronous to XCLK. Internal encoding uses CLK. If XCLK and CLK are equal to each other in both clock period and phase, then SYNC can equal 1. This reduces the encoder input time by one clock cycle. If XCLK and CLK are not equal, then SYNC must equal 0. SYNC should not be connected to logic or input pins.

The input data can be input in any code order. That is, it is not necessary to wait for the encoder to output the last block of one code before changing to another code. If changing the code, the encoder parameters NTS[1:0], KS[3:0] and PRS must stay constant from the time XS goes high to until after XF goes high.

Figure 3 illustrates the encoder input timing. Each input X_i, $0 \leq i \leq K$–1 represents a data bit at time i. XS is shown going high again for the case

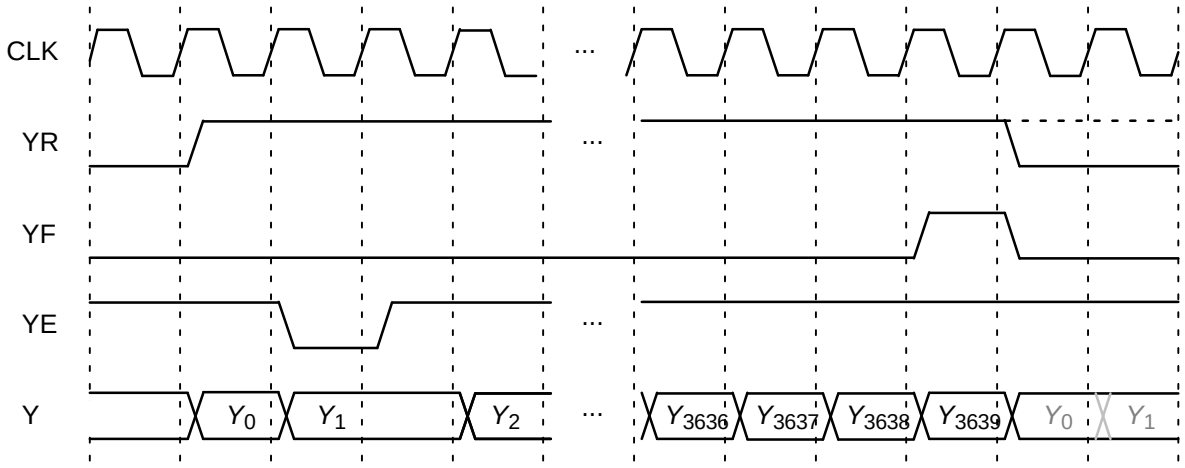

Figure 4: Turbo encoder output timing (R = 1/2, K =1784).

where FULL = 0. Note that for SYNC = 0, FULL will go high for at least four XCLK cycles after XF goes high. If one half of the input memory is then available, FULL will go low. Otherwise, FULL will remain high.

Figure 4 illustrates the encoder output timing. The signal YR goes high when data is ready to be output. The input YE can be used to hold all output signals. If YE is high, YR is high for $n(K+36)$ CLK cycles. Output YF goes high at the end of the last coded bit. Outputs Y, YR and YF are synchronous to CLK. Input YE must be synchronous to CLK.

Scrambling

After encoding, the encoded data can be optionally scrambled using a pseudo random sequence which repeats every 255 clock cycles. The pseudo randomiser select PRS input can be used to enable (PRS=1) or disable (PRS=0) scrambling.

The scrambler generator is an 8–bit right shift register with initial contents of all ones at the start of each encoded block. The output is $e(D) = D^8u(D)$ where $u(D) = Du(D) + D^3u(D) + D^5u(D) + D^8u(D) = e_0 + e_1D + e_2D^2 + ... + e_{N-1}D^{N-1}$, where $N = n(K+4)$ is the number of coded bits. When $e_i = 1$ the encoded data is inverted, otherwise the data is unchanged.

Sync Marker

Before the encoded data is transmitted a sync marker is attached. This sync marker is of length $32n$ and is not scrambled using the pseudo randomiser. Table 4 gives the sync marker sequence in hexadecimal. The left most bit is transmitted first.

Encoder Speed

The average input data rate f_e is

Table 4: Sync Marker Sequence

Rate	Sequence
1/2	034776C7272895B0
1/3	25D5C0CE8990F6C9461BF79C
1/4	034776C7272895B0
	FCB88938D8D76A4F
1/6	25D5C0CE8990F6C9461BF79C
	DA2A3F31766F0936B9E40863

$$f_e = \frac{F_cK}{n(K + 36)} \quad (1)$$

where n is the number of coded bits (2, 3, 4 or 6) and $F_c = 1/T_c$ is the encoder CLK speed.

Encoder Delay

The total encoder delay can be separated into two parts. This is the input memory delay T_i and encoder delay T_e. Each delay is equal to

$$T_i = (K + 1)T_x \quad (2)$$

$$T_e = (3 - S_x)T_c \quad (3)$$

where T_x is the XCLK period, S_x = SYNC and T_c is the CLK period.

The above delays assume that XE is high in the minimum time and that the encoder parameters do not change between encoded blocks. When encoding multiple blocks with varying code parameters, the delay is more difficult to predict. When SYNC is low the actual encoder delay will vary from $T_e - T_c$ to T_e.

Ordering Information

SW–PCE04CH–SOP (SignOnce Project License)
SW–PCE04CH–SOS (SignOnce Site License)
SW–PCE04CH–VHD (VHDL ASIC License)

All licenses include EDIF and VHDL cores. The SignOnce and ASIC licenses allows unlimited instantiations. The EDIF core can be used for Virtex–2, Spartan–3 and Virtex–4 with Foundation or ISE software. The VHDL core can be used for Virtex–5, Spartan–6, Virtex–6, 7–Series, UltraScale and UltraScale+ with ISE or Vivado software.

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] Consultative Committee for Space Data Systems, “Recommendation for space data system standards: TM synchronization and channel coding,” CCSDS 131.0–B–3, Blue Book, Sep. 2017.

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 fax +61 8 7117 1416

Revision History

- 0.00 25 Feb. 2021. Preliminary product specification.
- 0.01 1 Mar. 2021. Added YE input. KS order changed.
- 1.00 26 Mar. 2021. First release. Changed XSYNC to SYNC and NS[1:0] to NTS[1:0].
- 1.01 30 Jul. 2021. Reduced number of LUTs for MODE = 1. One RAMB18 for MODE = 0. Updated encoder speeds.
- 1.02 5 May 2022. Added XRST input.
- 1.03 4 Aug. 2023. Added EDIF core.