MAP04T Features

- 16 state soft–in–soft–out (SISO) maximum a posteriori (MAP) triple interleaved error control decoder and systematic recursive convolutional encoder
- Up to 95 Mbit/s decoding speed
- Rate 1/2, 1/3, or 1/4 with optional punctured inputs
- 6–bit received data, 8–bit soft–in and soft–out data for information and parity bits for all rates
- 8–bit branch metric inputs for rate 1/2
- Optional code polynomials
- Optional block decoding with or without tail
- Optional max–log–MAP or log–MAP algorithm with 9 or 17 programmable SNR's
- Continuous sliding block algorithm with sliding block lengths of 32 or 64
- Low decoding delay (400 or 784 CLK cycles)
- No external RAM required
- Asynchronous logic free design
- Ideal for iterative decoding of CCSDS turbo codes
- Available as BIT/MCS files for download into Xilinx Virtex field programmable gate arrays (FPGA) or EDIF/VHDL core

Introduction

The MAP04T is a 16 state very high speed maximum a posteriori (MAP) soft–in–soft–out (SISO) triple interleaved error control decoders with log–likelihood–ratio outputs for both the data and parity bits. A MAP decoder finds the most likely information bit to have been transmitted given a received noisy or distorted sequence, thus minimising the bit error rate (BER). This is unlike a Viterbi decoder which finds the most likely coded sequence. At low BERs, the MAP and Viterbi algorithms provide almost identical performance. However, at high BERs such as that experienced in iterative "turbo" decoders, MAP decoders can perform approximately 1 dB better than soft–output Viterbi decoder algorithms. A MAP decoder also inherently provides a soft output.

To achieve maximum decoding speed, the MAP04T uses three stages of internal pipelining. This results in a single MAP decoder that is equivalent to three MAP decoders operating in parallel in a triple interleaved structure.

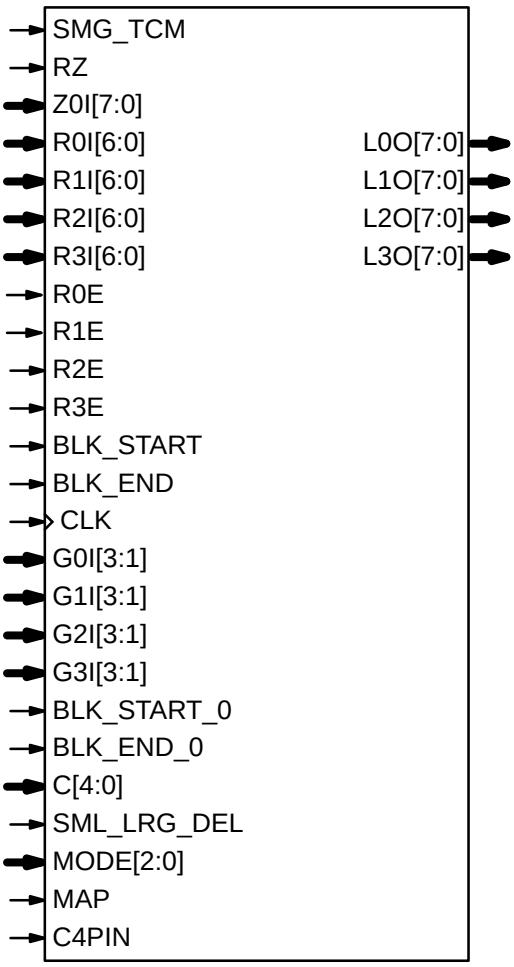

Figure 1: MAP04T schematic symbol.

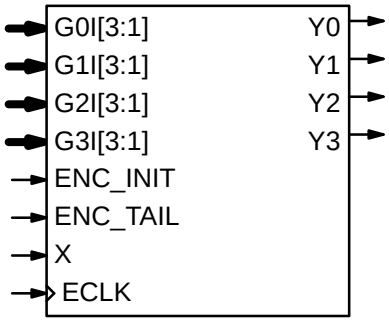

Figure 2: ENC04T schematic symbol.

Figures 1 and 2 show the schematic symbols for the MAP04T decoder and ENC04T encoder,

respectively. These are the symbols that are used to compile various BIT files for download into Xilinx FPGA's. These symbols, along with an EDIF core for the ENC04T are freely available for Foundation schematics from [1]. Table 1 shows the various Xilinx parts currently supported. Support for other Xilinx Virtex parts are available on request. Consult the Xilinx data book for a list of devices, speeds, and packages currently available.

Table 1: Xilinx parts currently supported*.

MAP Decoder/ Xilinx Part	Speed (Mbit/s)/ Speed Grade		
MAP04T/XCV400	56.8/–4	65.3/–5	73.2/–6
MAP04T/XCV400E	76.3/–6	85.3/–7	95.6/–8

*Mode 1H, C4PIN low, MAP high

Encoder

Figure 3 gives a block diagram of the triple interleaved 16 state systematic recursive encoder available in the ENC04T. The input and output D FF's are not shown. X is the data input and Y0 to Y3 are the coded outputs. GiIj = $g_i^j \in \{0, 1\}$, $0 \leq i \leq 3$, $1 \leq j \leq 3$, correspond to the code polynomial coefficients which are also used by the decoder. Data is clocked during the low to high transition of ECLK. Note that due to three stages of internal pipelining in the MAP decoder, each delay element (D^3) is equivalent to a length three shift register.

The encoder polynomials are defined as

$$g_i(D) = 1 + g_i^1 D + g_i^2 D^2 + g_i^3 D^3 + D^4 \quad (1)$$

where D is the delay operator and + indicates modulo–2 (exclusive OR) addition. It is usual practice to express the coefficients in octal notation, e.g., $g_0 = 23_8 = 10011_2 \equiv g_0(D) = 1 + D^3 + D^4$. This corresponds to G0I[3:1] = 100_2.

Figure 4 shows the timing diagram for encoding a block of data of length N, including 12 tail bits (the subscripts indicate the time index). The encoder starts and ends in state 0. The ENC_INIT signal is used to indicate the start of the block and initialises the encoder state to 0 when high and during a low to high transition of ECLK. The ENC_TAIL signal is used to force the final encoder state to zero after being held high for 12 ECLK cycles. While ENC_TAIL is held high, the multiplexer in Figure 3 is in the "Tail" position, i.e., the input data is not selected.

For continuous data operation ENC_INIT and ENC_TAIL should be held low.

The CCSDS turbo code standard [2] has G0I[3:1] = 100_2, G1I[3:1] = 101_2, G2I[3:1] = 010_2, and G3I[3:1] = 111_2.

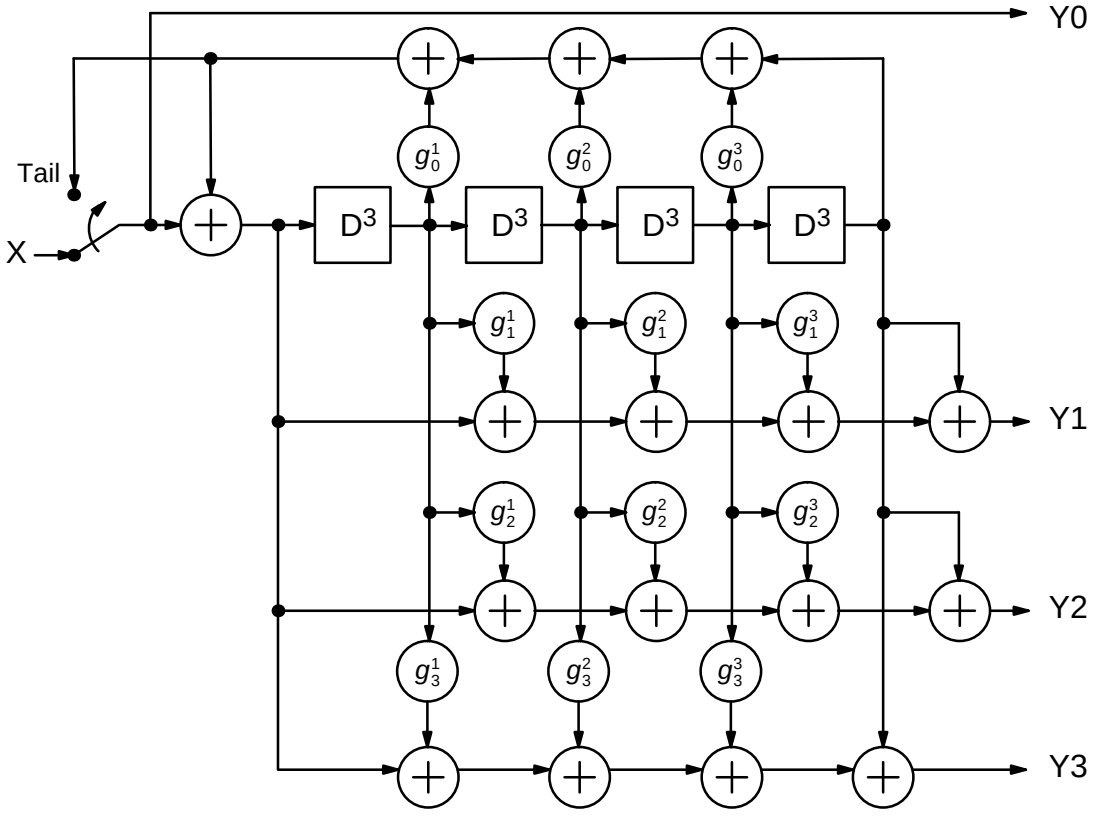

Figure 3: ENC04T triple interleaved 16 state systematic recursive convolutional encoder.

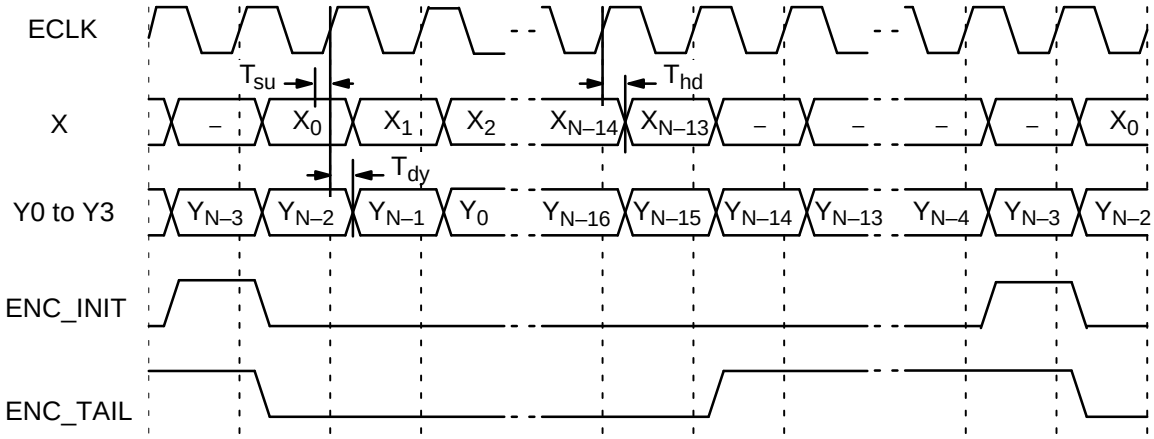

Figure 4: ENC04T Encoder Timing

MAP Decoder

The MAP decoder is designed to be very flexible and can be operated in either continuous or block mode.

Theory of Operation

The MAP decoding algorithm [4] finds the following likelihood ratio:

$$\lambda_k^0 = \frac{P_r(d_k = 1|R_1^N)}{P_r(d_k = 0|R_1^N)}, \tag{2}$$

where d_k is the encoded data bit at time k (equivalent to X in Figure 3) and R_1^N is the received data from time 1 to N. The basic algorithm for an additive white Gaussian noise (AWGN) channel involves exponentials, multiply–add, and division operations. By taking the negative logarithm of (2), the exponentials disappear, the multiplies becomes additions, the divisions become subtractions, and the additions become the E operand defined below:

$$a \text{ E } b = \min(a, b) - f(|a - b|) \tag{3}$$

where

$$f(z) = c\ln(1 + e^{-z/c}) \tag{4}$$

and c is a constant dependent on the signal amplitude and noise variance. The function $f(z)$ is implemented as look up tables within the decoder. When C4PIN and the MAP pin are both high, there are 18 integer values of c, from 0 to 17 via the C inputs. C should be equal to the closest integer to c. For example, if c = 7.442, then C[4:0] = 7 ≡ 00111_2. Values of C greater than 17 are limited to 17. Due to quantisation effects, C = 1 is equivalent to C = 0.

When C = 0, $f(z)$ = 0 and the sub–MAP (also known as max–log–MAP) algorithm is implemented [5]. The sub–MAP algorithm does not require knowledge of the signal to noise ratio (SNR), but does not perform as well as the MAP algorithm. The MAP pin when low can also be used to select sub–MAP operation.

Note that when a change in C is detected on a low–to–high transition of CLK, the decoder takes 67 CLK cycles for the new look–up tables to be loaded. If a change in C is detected during loading, the whole operation starts again.

When C4PIN and C4 are both grounded, the maximum value of C is 9. This reduces the decoder complexity. The decoder takes 35 CLK cycles for the new look–up tables to be loaded. Note that C4PIN and MAP are soft inputs and should not be connected to a pin.

We define the log–likelihood ratio as

$$L_k^0 = -c\ln\lambda_k^0. \tag{5}$$

For binary phase shift keying (BPSK) or quadrature phase shift keying (QPSK) modulation the received signal is described by

$$R_k^i = A((1 - 2y_k^i)/\sqrt{m} + n_k^i) \tag{6}$$

where A is the signal amplitude, $y_k^i \in \{0, 1\}$, i = 0 to 3 correspond to the coded bits, m = 1 for BPSK or m = 2 for QPSK, and n_k^i is a Gaussian distributed random variable with zero mean and normalised variance σ^2. For a systematic code $y_k^0 = d_k$. Figure 5 shows the signal sets for BPSK and QPSK. We have

$$\sigma^2 = \left(2mR\frac{E_b}{N_0}\right)^{-1} \tag{7}$$

where E_b/N_0 is the energy per bit to single sided noise density ratio and $R = k/n$ is the code rate (k is the number of information bits and n is the number of coded bits).

Since a zero is transmitted as $+A/\sqrt{m}$ and a one is transmitted as $-A/\sqrt{m}$ the sign bit of a noiseless R_k^0 in two's complement notation is

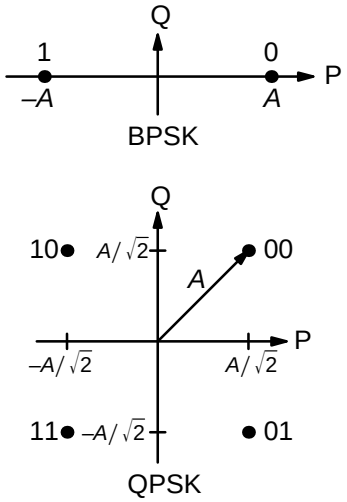

Figure 5: BPSK and QPSK signal sets.

equal to d_k. Similarly, (2) is formed so that the sign bit of L_k is equal to the estimate of d_k, i.e.,

$$\hat{d}_k = \begin{cases} 0 & : L_k \geq 0 \\ 1 & : L_k < 0 \end{cases}. \qquad (8)$$

For optimal performance with BPSK or QPSK, the constant c should be adjusted such that

$$c = \sigma^2 \sqrt{m} A/2. \qquad (9)$$

Due to quantisation and limiting effects the value of A should also be adjusted according to the received signal to noise ratio. A program for calculating the optimum values of A and c is freely available from [3].

The value of A directly corresponds to the 6–bit signed magnitude inputs (described in more detail later). 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 16 (which has a range between 15.5 and 16.5).

Example 1: Rate 1/3 BPSK code operating at E_b/N_0 = 0.3 dB. From (7) we have σ^2 = 1.39988. Assuming c = 11 we have from (9) that A = 15.7.

Example 2: Rate 1/2 QPSK code operating at E_b/N_0 = 0.8 dB. From (7) we have σ^2 = 0.41588. Assuming c = 7 we have from (9) that A = 23.8. Note that the amplitude in each dimension is $A/\sqrt{2}$ = 16.8.

A priori information corresponding to the information bit can also be input to the MAP decoder. This input corresponds to

$$Z_k^0 = -c \ln\left(\frac{P_r(d_k = 1)}{P_r(d_k = 0)}\right), \qquad (10)$$

and is Z0I in the MAP04T. The output L_k^0 (corresponding to L0O) can be expressed as

$$L_k^0 = Z_k^0 + R_k^0 + Z_k^{0\prime}, \qquad (11)$$

where $Z_k^{0\prime}$ is the extrinsic information.

The MAP04T also provides a log ratio output for the parity bits y_k^j, $1 \leq j \leq 3$ (equivalent to Yj in Figure 3). We define this as

$$L_k^j = -c \ln \frac{P_r(c_k = j|R_1^N)}{P_r(c_k = 0|R_1^N)} \qquad (12)$$

which corresponds to LjO. Similarly to L_k^0, we can express L_k^j as

$$L_k^j = Z_k^j + R_k^j + Z_k^{j\prime}, \qquad (13)$$

where Z_k^j is the *a priori* information for y_k^j, R_k^j is the received noisy parity symbol (RjI), and $Z_k^{1\prime}$ is the extrinsic information for y_k^1.

Decoder Operation

The MAP04T is a sliding block decoder [6] and is thus able to continuously decode data (just like a Viterbi decoder). When SML_LRG_DEL is low or high the sliding block length (L) is 32 or 64, respectively. The decoding delay is 400 or 784 CLK cycles. For best performance an L of 64 should be chosen, especially when punctured codes are used. For minimum delay, L of 32 can be chosen.

The MAP04T also uses a unique triple interleaved architecture to maximise decoder speed. The equivalent decoder architecture is shown in Figure 6. Internally, the decoder is implemented as a single MAP decoder, but with three stages of pipelining. This allows a decoder speed that is nearly three times of what can normally be achieved with only a moderate increase in complexity.

The encoded and decoded streams are equivalent to three completely independent interleaved streams. Each equivalent MAP decoder has a decoding delay of $133\,1/3$ or $261\,1/3$ bits.

Block Operation

The decoder is also able to decode blocks of data with the same low decoder delay as in continuous mode. The static inputs BLK_START_0 and BLK_END_0 when high indicate whether the block starts or ends in state 0. When low the decoder assumes that the block starts or ends in an unknown state. The signals BLK_START and BLK_END indicate the start and end of the block in time. When these signals go high the forward and reverse state metrics of the MAP algorithm are initialised to their appropriate starting values. Figure 7 illustrates the timing for block encoded data. Note that only one of the triple interleaved schemes is using the BLK_START and BLK_END signals in this example.

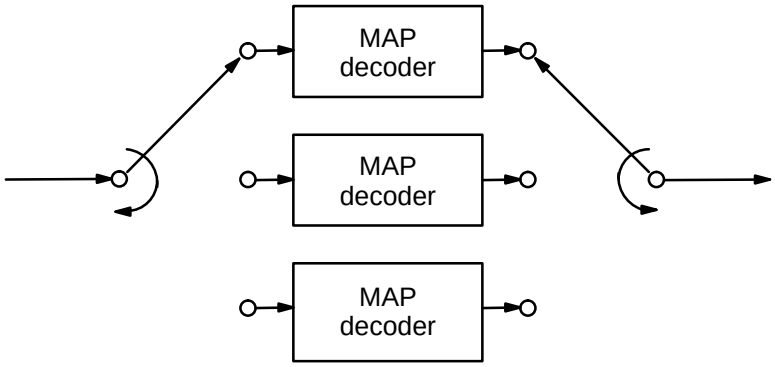

Figure 6: Equivalent triple interleaved MAP decoder.

Block codes with four parity bits can also be decoded. For these codes Y1 to Y3 are not used and R1I to R3I are set to 0. The divisor polynomial g_0 should be a primitive polynomial such as $g_0 = 1 + D^3 + D^4$ (G0I[3:1] = 100_2). Block encoding and decoding is performed as normal (the encoder and decoder states must be made to both start and end in state zero). The (15,11) BCH codes are block codes that can be near–optimally decoded. Shorter or longer block lengths are also possible.

Data Formats

When RZ is low, the seven bit received data R0I to R3I can be in signed magnitude or two's complement format. In signed magnitude format (SMG_TCM low), the most significant bit (i.e., RiI6, i = 0 to 3) corresponds to the sign bit and the remaining bits represent the magnitude. When SMG_TCM is high the input data is in two's complement format. Table 2 gives a partial listing of the range represented by each quantised value.

Internally, the decoder converts the seven bit R0I to R3I values to six bit signed magnitude quantisation with a central dead zone. Table 3 shows the six bit internal quantisation ranges. For external six bit signed magnitude data with central dead zone quantisation, the data is input to RiI[6:1] with RiI0 = 0. For six bit two's complement data with central dead quantisation, the data is input to RiI[6:1] with RiI0 = RiI6.

Table 2: Quantisation for R0I to R3I.

Sign Mag.	Two's Comp	Range
63	63	63↔∞
62	62	62↔63
⋮	⋮	⋮
1	1	1↔2
0	0	0↔1
64	127	–1↔0
65	126	–2↔–1
⋮	⋮	⋮
126	65	–63↔–62
127	64	–∞↔–63

Internally, Z0I is added to R0I to produce an eight bit signed magnitude value. When RZ is low, the seven bit R1I to R3I inputs are converted to six bit signed magnitude values with a central dead zone.

When RZ is high RiE,RiI[6:0] for i = 1 to 3 are equal to eight bit two's complement inputs with notation Z1I[7:0] to Z3I[7:0]. Internally, Z1I to Z3I are converted to eight bit signed magnitude values. This allows the decoder to iteratively decode serially concatenated convolutional codes. In this

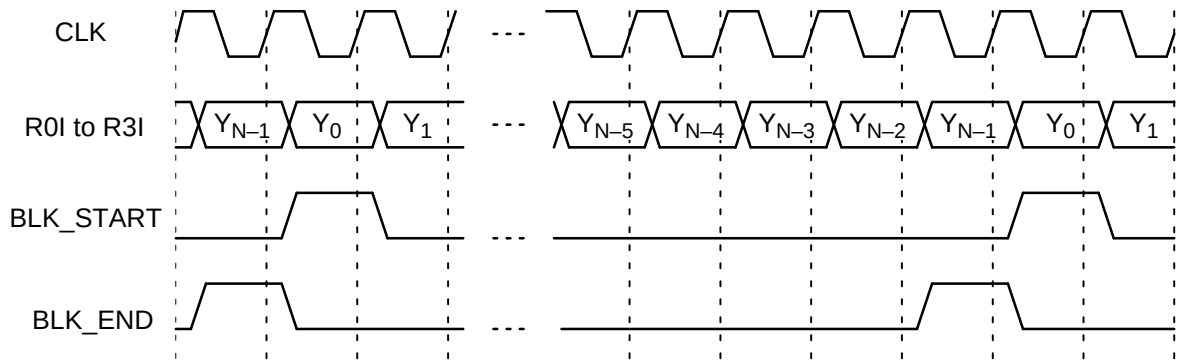

Figure 7: Decoder Block Timing

case, the inner decoder has RZ low and the outer decoder has RZ high. For iteratively decoding parallel concatenated codes, RZ is low for both the inner and outer decoders.

Table 3: Internal quantisation for R0I to R3I.

Decimal	Binary	Range
31	011111	30.5↔ ∞
30	011110	29.5↔30.5
⋮	⋮	⋮
2	000010	1.5↔2.5
1	000001	0.5↔1.5
0	000000	–0.5↔0.5
32	100000	–0.5↔0.5
33	100001	–1.5↔–0.5
34	100010	–2.5↔–1.5
⋮	⋮	⋮
62	111110	–30.5↔–29.5
63	111111	– ∞ ↔–30.5

The eight bit Z0I to Z3I inputs and L0O to L3O outputs are in two's complement arithmetic. Table 4 gives a partial listing of the range represented by each quantised value.

Table 4: Quantisation for Z0I to Z3I and L0O to L3O.

Decimal	Binary	Range
127	01111111	127↔ ∞
126	01111110	126↔127
⋮	⋮	⋮
1	00000001	1↔2
0	00000000	0↔1
255	11111111	–1↔0
254	11111110	–2↔–1
⋮	⋮	⋮
129	10000001	–127↔–126
128	10000000	– ∞ ↔–127

For rate 1/2 codes and MODE = 7 (see next section), RiE,RiI[6:0] for i = 0 to 3 are equal to eight bit branch metric (BM) inputs with notation BM0I[7:0] to BM3I[7:0]. RZ, SMG_TCM, and Z0I inputs are not used. The BM inputs correspond to the BM's for symbols $2y^1 + y^0$, e.g., the BM for $y^1 = 1$ and $y^0 = 0$ corresponds to BM2I. The BM's range in value from 0 to 255. The lower the BM value, the more likely the symbol. To maximise performance, the smallest BM should be subtracted from all four BM's (implying that the smallest BM will then be equal to zero).

The branch metric inputs can be used to decode signal sets other than BPSK and QPSK, e.g., rate 1/2 16QAM. For these signal sets, the branch metrics are not linear to the received signal. Non–linear computations or look–up tables can be used to calculate the branch metrics for these signal sets.

Punctured Code Operation

When RZ is low, the input signals R0E to R3E are used to enable the received data inputs R0I to R3I, respectively. Data is erased when R0E to R3E are low for use in punctured code decoding. Manual puncturing can be performed by forcing R0I[5:0] to R3I[5:0] low.

Example 3: Rate 1/2 code punctured to rate 2/3. In this case the systematic bit is not punctured (R0E high) while the parity bit is punctured every second bit (R1E alternates between high and low). R1E is equal to CLK divided by six. The phase of R1E needs to match the received data so that when R1E is low, the parity information is punctured.

Mode Selection

To minimise the decoder complexity, the MODE[2:0] inputs can be used with the schematic symbols to select only those rates and input types that are expected to be used. Table 5 lists the ten possible modes of operation for the MAP04T.

Table 5: Mode selection (MAP high, C4PIN low)

Mode	*n*	Input	Slices (L)	Slices (H)	Block RAM
0	1	R	2994	3038	11
1	2	R	3021	3065	14
2	3	R	3141	3185	17
3	4	R	3501	3545	20
4	2	R,Z	3878	3962	14
5	3	R,Z	4879	5003	17
6	4	R,Z	6120	6284	20
7	2	BM	2973	3017	20
8	4,2	R,BM	3549	3593	20
9	4,2	R,Z,BM	6168	6332	20

For modes 0 to 7, MODE[2:0] is directly equal to the mode number, e.g., for mode 4, MODE[2:0] = 100_2. These inputs should be connected to internal VCC and GND supplies. Connecting the inputs to pads will result in excessive configurable logic block (CLB) usage and decreased decoder speed.

Table 6: Inputs and outputs selected for BIT/MCS files for different modes

Inputs and Outputs	Mode									
	0	1	2	3	4	5	6	7	8	9
BLK_END, BLK_END_0, BLK_START, BLK_START_0, CLK	●	●	●	●	●	●	●	●	●	●
RESET, ECLK, ENC_INIT, ENC_TAIL, X	●	●	●	●	●	●	●	●	●	●
SML_LRG_DEL*, C[4:0]†	●	●	●	●	●	●	●	●	●	●
R0E, R0I[6:0], Y0, L0O[7:0]	●	●	●	●	●	●	●	●	●	●
R1E, R1I[6:0], Y1		●	●	●	●	●	●	●	●	●
R2E, R2I[6:0]			●	●		●	●	●	●	●
Y2			●	●		●	●		●	●
R3E, R3I[6:0]				●			●	●	●	●
Y3				●			●		●	●
L1O[7:0], RZ					●	●	●			●
L2O[7:0]						●	●			●
L3O[7:0]							●			●
MODE0/MODE2									●	●
SMG_TCM, Z0I[7:0]	●	●	●	●	●	●	●		●	●

* Not connected for d = L; † C[4:0] not connected if MAP low, C4 not connected if C4PIN low

For mode 8, MODE2 should be connected to an input pad while MODE1 and MODE0 should be connected to VCC. MODE2 can then be used to select between R inputs with $n = 4$ (MODE2 low) and BM inputs with $n = 2$ (MODE2 high).

For mode 9, MODE0 should be connected to an input pad while MODE1 and MODE2 should be connected to VCC. MODE0 can then be used to select between R or Z inputs with $n = 4$ (MODE0 low) and BM inputs with $n = 2$ (MODE0 high).

For R inputs, only L0O is output. For R,Z inputs all the outputs are provided corresponding to the number of inputs. Table 6 shows the various inputs and outputs that are available for BIT/MCS files for the ten different modes. Both the encoder and decoder use the same G inputs.

(L) indicates the estimated number of slices used for both the encoder and decoder with $L = 32$ only. (H) indicates the estimated number of slices with $L = 32$ or 64.

Other inputs that should not be connected to pins are C4PIN and MAP. These pins are used to trade off various options with decoder complexity.

The MAP pin is used to select whether the internal lookup tables are used. When MAP is high, normal MAP decoding is performed. When MAP is low, sub–MAP operation is selected. In this case, C[4:0] and C4PIN are not used. If MAP is high and C4PIN is low then C4 must not be connected to a pin. If C4PIN is high then C4 should be connected to a pin. Table 7 shows the change in slices for these configurations compared to Table 5.

Table 7: Change in slices

Mode	MAP04T	
	MAP=L	MAP=H C4PIN=H
0–3,7,8	–966	+681
4	–1341	+906
5	–1716	+1251
6,9	–2091	+1536

If the code selection inputs G0I to G3I are connected to pins, decoder complexity increases by approximately 10–20%.

Decoder Performance

Figure 8 plots BER versus E_b/N_0 of the MAP04T in continuous decoding operation at code rate 1/2 with code polynomials $g_0 = 37_8$ and $g_1 = 21_8$. A BPSK amplitude A of 12 was used from –2 to –1 dB and A = 16 was used from –0.5 to 3 dB. The value of c used at each operating point was the optimum value quantised to the nearest integer. The “software MAP” curve is a computer simulation of the MAP decoding algorithm with no quantisation. As can be seen, the MAP04T achieves near optimum performance with an implementation loss in the hundredths of a dB.

Notice that the sub–MAP decoder has almost identical performance at low BER. However, at high BER, where iterative decoders initially operate, the MAP algorithm performs significantly better than the sub–MAP algorithm.

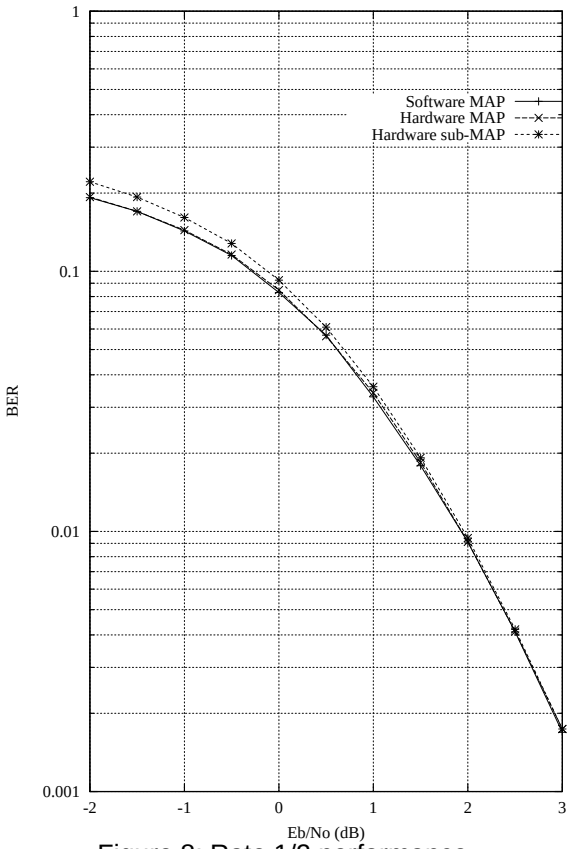

Figure 8: Rate 1/2 performance.

Example

In this section we give an example of how the MAP04T can be used as a continuous rate 1/2 QPSK encoder and decoder. Note that the MAP04T does not perform any synchronisation. This needs to be performed external to the chip.

A simple synchronisation circuit could monitor the output magnitude of the decoder output. The average magnitude will be higher in the synchronised state compared to the unsynchronised states.

Figure 9 shows how the MAP04T1H14C4 (1H indicates the mode, 14 indicates the code, and C4 indicates that pin C4 is used) can be configured for continuous rate 1/2 QPSK operation. The code used is and $g_1 = 31_8 = 11001_2 \rightarrow$ G1I[3:1] = 001_2 = 1, $g_0 = 23_8 = 10011_2 \rightarrow$ G0I[3:1] = 100_2 = 4 (which is 180° rotationally invariant). Note that unconnected inputs are pulled down to ground. Since the code is invariant to 180° phase rotations, differential encoding and decoding can be used if desired.

The demodulator output is assumed to be in two's complement form. An *L* of 64 is used for best performance. The information and parity bits are assumed to be equally likely implying Z0I = 0. Unconnected inputs are pulled to ground.

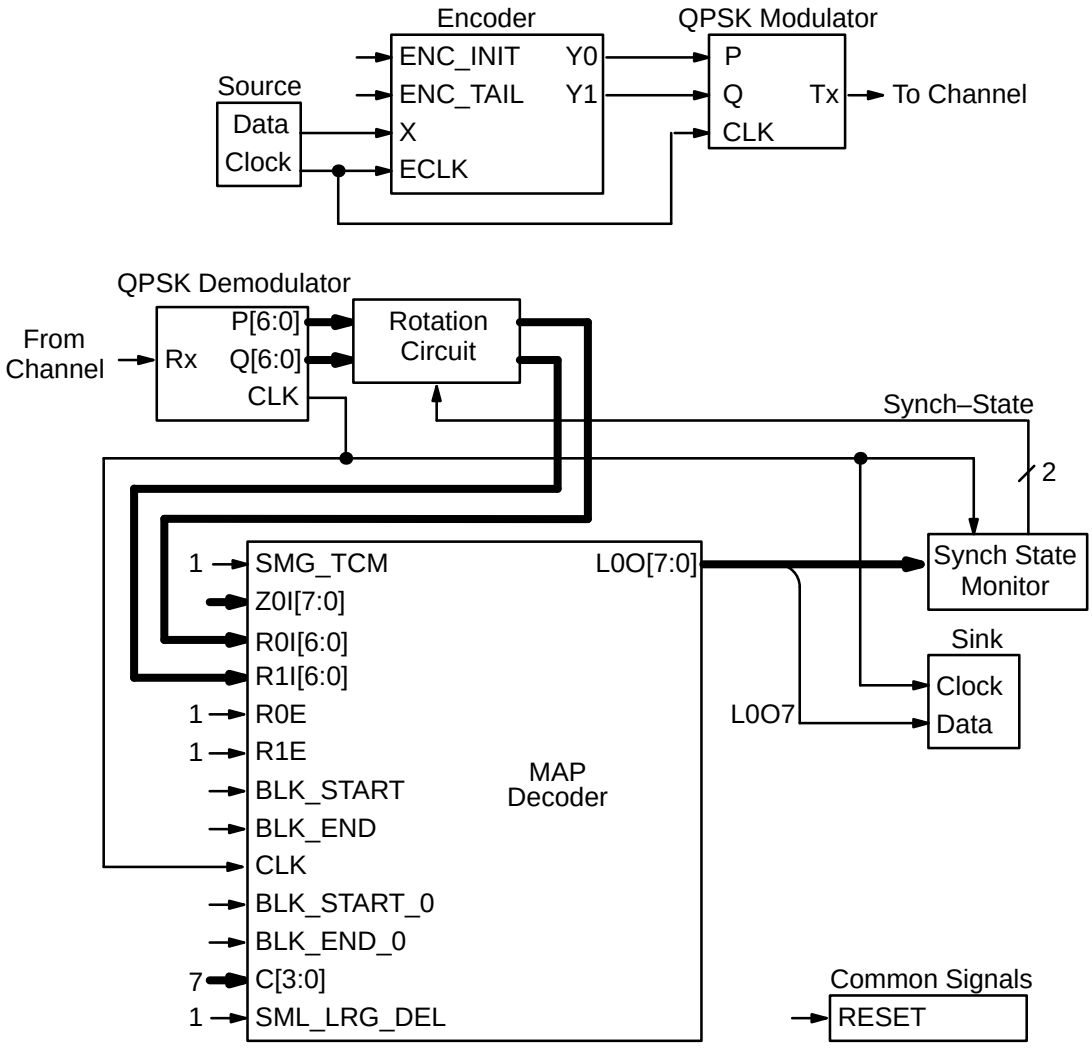

Figure 9: Block diagram of rate 1/2 QPSK codec.

BIT and MCS Files

The BIT file is what is downloaded into the FPGA on startup. The Xilinx data book explains how this can be achieved. The MCS files can be used to program serial PROMs which can then be used to program the FPGA on powerup.

For the BIT files, the encoder and decoder are reset when RESET (not shown in schematic symbols) goes high. Except for clock inputs, all the inputs for the encoder and decoder have pulldown resistors.

Configuration

This section gives the options that were selected for Virtex and Virtex–E devices.

Configuration

- Configuration Rate: 4 MHz
- Configuration Pins –
  Configuration Clk: PullUp
  M0: PullUp
  M1: PullUp
  M2: PullUp
  Program: PullUp
  Done: PullUp
  Unused Pins: PullUp
- JTAG Pins –
  TCK: PullUp
  TDI: PullUp
  TDO: Float
  TMS: PullUp
- Produce ASCII Configuration File: No
- Enable the User ID Code: Yes
  Code: 53570101_H (MAP04T01)
  53570102_H (MAP04T02)
- Enable .bit File Compression: No

Startup

- Start–Up clock: CCLK
- Output Events –
  Done: C4
  Enable Outputs: C5
  Release Set/Reset: C6
  Release Write Enable: C6
  Release DLL: No Wait
- Enable Internal Done Pipe: No

Readback

- Configuration Mode: JTAG
- Security –
  Enable Readback and Reconfiguration: Yes
  Generate Readback Bit Stream: No
  Disable Readback: No
  Disable Readback and Reconfiguration: No

Switching Characteristics

The following switching characteristics reflect worse–case values over the recommended operating conditions. The values are expressed in units of nanoseconds. All values are preliminary.

SW–MAP04T1H14C4–XCV400–6HQ240C		**–4**		**–5**		**–6**	
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Signal setup before CLK, ECLK	T_{su}	2.0		1.8		1.6	
Signal hold after CLK, ECLK	T_{hd}	0		0		0	
Signal delay after CLK, ECLK	T_{dy}		3.5		3.2		2.9
ECLK period	T_{ecp}	8.68		7.54		6.73	
CLK period	T_{cp}	17.62		15.31		13.67	
CLK low	T_{clh}	3.1		2.7		2.4	
CLK high	T_{wps}	3.1		2.7		2.4	

SW–MAP04T1H14C4–XCV400E–8PQ240C		**–6**		**–7**		**–8**	
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Signal setup before CLK, ECLK	T_{su}	1.5		1.4		1.3	
Signal hold after CLK, ECLK	T_{hd}	0		0		0	
Signal delay after CLK, ECLK	T_{dy}		2.9		2.8		2.4
ECLK period	T_{ecp}	5.39		4.82		4.30	
CLK period	T_{cp}	13.11		11.72		10.46	
CLK low	T_{clh}	2.4		2.1		1.9	
CLK high	T_{wps}	2.4		2.1		1.9	

Pinouts (Virtex/Virtex–E)

Pad Name	HQ240/PQ240
BLK_END	P208
BLK_END_0	P149
BLK_START	P100
BLK_START_0	P147
CLK	P92
C0	P52
C1	P50
C2	P65
C3	P49
C4	
ECLK	P89
ENC_INIT	P84
ENC_TAIL	P74
L0O0	P186
L0O1	P175
L0O2	P199
L0O3	P192
L0O4	P191
L0O5	P195
L0O6	P194
L0O7	P193
L1O0	
L1O1	
L1O2	
L1O3	
L1O4	
L1O5	
L1O6	
L1O7	
R0E	P40
R0I0	P35
R0I1	P39
R0I2	P36
R0I3	P66
R0I4	P67
R0I5	P68
R0I6	P27
R1E	P224
R1I0	P222
R1I1	P223
R1I2	P218
R1I3	P217
R1I4	P221
R1I5	P220
R1I6	P228
RESET	P238
RZ	
SMG_TCM	P229
SML_LRG_DEL	P168
X	P80
Y0	P78
Y1	P79
Z0I0	P70
Z0I1	P71
Z0I2	P72
Z0I3	P38
Z0I4	P33
Z0I5	P34
Z0I6	P31
Z0I7	P26

Please consult the Xilinx data book for power and configuration pinouts.

Packages

UCF No.	Mode	Xilinx Part No.
MAP04T01	1H14C4	XCV400–6HQ240C
MAP04T02	1H14C4	XCV400E–8PQ240C

Other Xilinx parts and modes are also available. All pinouts are upward compatible. See the Xilinx data book for pinouts of other packages. Other pinouts can be ordered from *Small World Communications*.

Ordering Information

SW–MAP04T–UNL for EDIF/VHDL core yearly unlimited license

SW–MAP04T–BAS–*n* for EDIF/VHDL core basic license

SW–MAP04T–*md*C*c*–*p*–*n* for BIT/MCS file

n = number of instantiations

m = mode (0 to 9)

d = L (*L* = 32) or H (*L* = 32 or 64)

c = no. of C pins (0 for LOGMAP low,
4 for LOGMAP high and C4PIN low, or
5 for LOGMAP high and C4PIN high)

p = Xilinx part no. (e.g., XCV400E–8PQ240C)

Please indicate how many instantiations you wish to license. An instantiation is considered to be an integrated circuit that uses or is derived from our software in the device's programming or manufacture. License costs per instantiation decrease with increasing number of instantiations. Yearly, unlimited instantiation licenses are also available. Note that *Small World Communications* only provides software and does not provide the actual devices themselves. Please contact *Small*

World Communications for a quote on the number of instantiations and type of license you require.

References

[1] Small World Communications, "Sworld Foundation symbols and sample EDIF files," v4.0, Jan. 2002. www.sworld.com.au/software/sworld.zip

[2] Consultive Committee for Space Data Systems, "Recommendation for space and data system standards: Telemetry channel coding," CCSDS 101.0–B–4 Blue Book, May 1999.

[3] Small World Communications, "MAP decoder constant program," v1.04, Jan. 2001. www.sworld.com.au/software/cmap.zip

[4] 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.

[5] 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.

[6] S. S. Pietrobon, "Implementation and performance of a turbo/MAP decoder," *Int. J. Satellite Commun.*, vol. 16, pp. 23–46, Jan.–Feb. 1998.

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 8332 3177