VA08V Features

- 16, 32, 64 or 256 states (memory *m* = 4, 5, 6 or 8, constraint lengths 5, 6, 7 or 9) Viterbi decoder
- Up to 392 MHz internal clock
- Up to 39.2 Mbit/s for 16, 32 or 64 states or 11.5 Mbit/s with 256 states
- Rate 1/2, 1/3 or 1/4 (inputs can be punctured for higher rates)
- Optional or standard code polynomials
- 6–bit received signed magnitude data
- Optional block decoding with or without tail
- Estimated channel bit error outputs
- Optional serial or parallel data input
- Optional automatic coded symbol synchronisation for rate 1/2 QPSK and rate 1/2 to 1/4 BPSK
- 1100 6–input LUTs. 1 or 2 18KB BlockRAMs.
- Asynchronous logic free design
- Free simulation software
- Available as VHDL core for Xilinx FPGAs under SignOnce IP License. ASIC, Altera, Lattice and Microsemi cores available on request.

Introduction

The VA08V is a 16, 32, 64 or 256 state error control decoder using the maximum likelihood Viterbi algorithm. The decoder is designed for maximum flexibility, allowing it to decode various communications standards, as well as custom coding solutions.

The VA08V uses eight add–compare–select (ACS) circuits in parallel up to 8 times for 16, 32 and 64 states and 32 times for 256 state convolutional codes. A single external 2Kx16 synchronous RAM (implemented with two 2Kx8 BlockRAMs) is used to perform the traceback. Smaller memories can be used with 16, 32, or 64 state only operation. In synchronous operation, 10 clock cycles are required per decoded bit for 16, 32, or 64 states or 34 clock cycles for 256 states. Asynchronous operation requires 11 or 35 clock cycles.

Figure 1 shows the schematic symbol for the VA08V 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 performance achieved with various Xilinx parts (parallel input and no automatic synchronisation). T_{cp} is the minimum clock period over recommended operating conditions. These performance figures may change due to device utilisation and configuration.

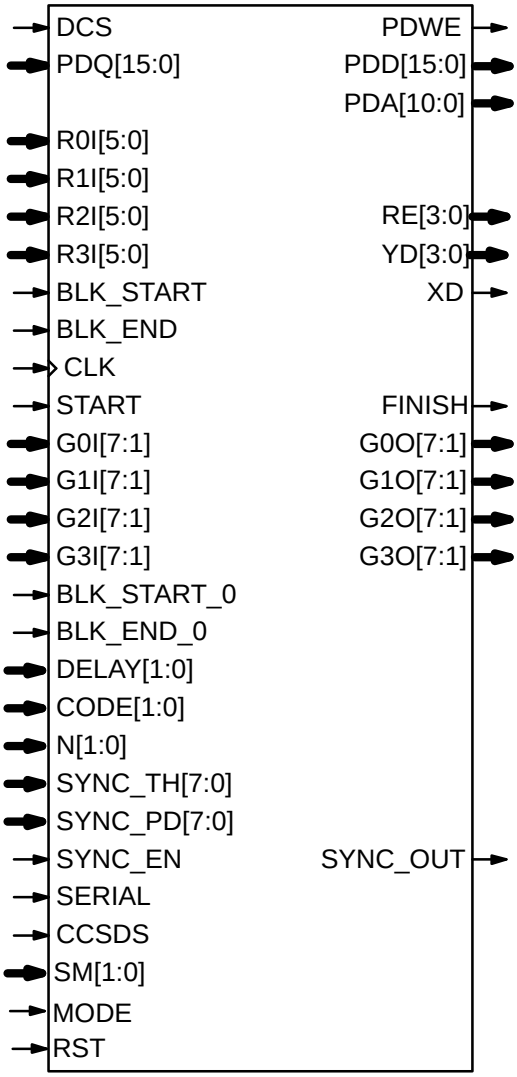


Figure 1: VA08V schematic symbol.

Signal Descriptions

BLK_START	Block Start
BLK_START_0	Start in State Zero
BLK_END	Block End
BLK_END_0	End in State Zero
CCSDS	Invert Sign Bit of R1I
CLK	System Clock

Table 1: Performance of Xilinx parts.

Xilinx Part	T_{cp} (ns)	Data Rate* (Mbit/s)	
		m=4,5,6	*m*=8
XC5VLX30–1	4.351	22.9	6.7
XC5VLX30–2	3.749	26.6	7.8
XC5VLX30–3	3.344	29.9	8.7
XC6VLX75T–1	3.731	26.8	7.8
XC6VLX75T–2	3.161	25.6	7.5
XC6VLX75T–3	2.801	31.6	9.3
XC7Z015–1	5.176	19.3	5.6
XC7Z015–2	4.231	23.6	6.9
XC7Z015–3	3.741	26.7	7.8
XC7A35T–1	5.098	19.6	5.7
XC7A35T–2	4.169	23.9	7.0
XC7A35T–3	3.707	26.9	7.9
XC7K70T–1	3.340	29.9	8.8
XC7K70T–2	2.701	37.0	10.8
XC7K70T–3	2.545	39.2	11.5

*Synchronous operation

CODE	Code Select (see Table 2) 0 = 3GPP™ Polynomials 1 = 3GPP2 Polynomials 2,3 = Use G0I to G3I
DCS	Decoder Chip Select
DELAY	Decoder Delay Select 0 = 64 + *m* 1 = 128 + *m* 2 = 256 + *m* (*m* ≤ 6)
FINISH	Decoder Finish
G0I–G3I	Code Polynomial Input
G0O–G3O	Code Polynomial Output
MODE	Maximum Rate Select 0 = Rates 1/2 to 1/3 1 = Rates 1/2 to 1/4
N	Code Rate 2 = Rate 1/2 3 = Rate 1/3 0 = Rate 1/4
PDA	Path Decision Address
PDD	Path Decision Data
PDQ	Path Decision Input
PDWE	Path Decision Write Enable
R0I–R3I	Received Data
RE	Estimated Symbol Error
RST	Synchronous Reset
SERIAL	0 = Parallel Input (R0I to R3I) 1 = Serial Input (R0I only)

Table 2: Convolutional Codes.

CODE [1:0]	SM [1:0]	N[1:0]	g0	g1	g2	g3
0X	00	10	23	35	–	–
0X	00	11	25	33	37	–
0X	00	00	23	35	25	37
0X	01	10	51	67	–	–
0X	01	11	51	67	75	–
0X	01	00	51	55	67	77
00	10	10	133	171	–	–
00	10	11	133	171	165	–
00	10	00	173	167	135	111
00	11	10	561	753	–	–
00	11	11	557	663	711	–
00	11	00	473	513	671	765
01	10	10	171	133	–	–
01	10	11	171	133	165	–
01	10	00	173	167	135	111
01	11	10	753	561	–	–
01	11	11	557	663	711	–
01	11	00	765	671	513	473
1X	XX	XX	G0I	G1I	G2I	G3I

SM	Code State Select 0 = 16 states (*m* = 4) 1 = 32 states (*m* = 5) 2 = 64 states (*m* = 6) 3 = 256 states (*m* = 8)
START	Decoder Start
SYNC_EN	Synchronisation Enable
SYNC_OUT	Synchronisation Otput SIgnal
SYNC_PD	Synchronisation Period (1 to 255)
SYNC_TH	Synchronisation Threshold (1 to 255)
XD	Decoded Data Output
YD	Decoded Symbol Output

Code Selection

Figure 2 gives a block diagram of a 256 state (*m* = 8) non–systematic encoder. To decode 256 state encoded data, select SM=3. X is the data input and Y0 to Y3 are the coded outputs. GiIj = $g_i^j \in \{0,1\}$, $0 \le i \le 3$, $1 \le j \le 7$, correspond to the code polynomial coefficients which are used by the decoder.

The encoder polynomials are defined as

$$g_i(D) = 1 + g_i^1 D + g_i^2 D^2 + ... + g_i^7 D^7 + D^8 \quad (1)$$

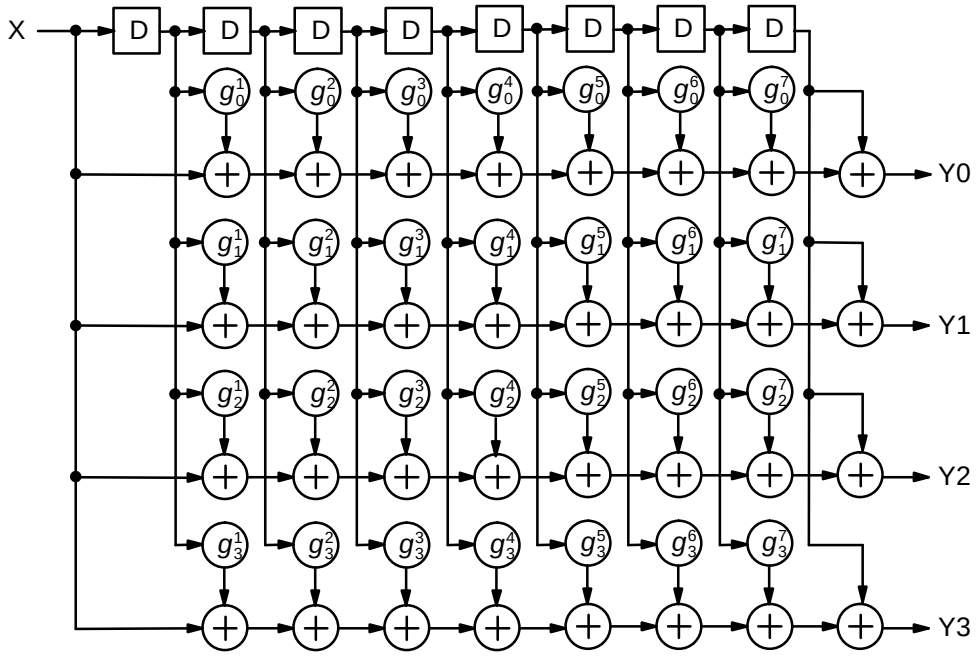


Figure 2: 256 state non–systematic convolutional encoder.

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., $g_0 = 561_8 = 101110001_2 \equiv g_0(D) = 1 + D^2 + D^3 + D^4 + D^8$. This corresponds to G0I[7:1] = 0001110_2.

When CODE[1] = 1, the code polynomials input to G0I[7:1] to G3I[7:1] are used. The input N[1:0] is also used to deselect the inputs for the various rates. That is, R2I[4:0] is internally grounded for rate 1/2 and R3I[4:0] is internally grounded for rate 1/2 and 1/3.

The 3GPP™ [1] and 3GPP2 [2] convolutional code standards are selected by CODE = 0 and 1, respectively. The codes are given in Table 2 in octal notation. G0O[7:1] to G3O[7:1] reflect the code polynomials that are selected.

Figure 3 shows the 64 state (m = 6) encoder. To decode 64 state encoded data, select SM = 2. To simplify decoder complexity the code polynomials are given by

$$g_i(D) = 1 + g_i^1 D + g_i^2 D^2 + g_i^3 D^3 + g_i^6 D^4 + g_i^7 D^5 + D^6 \quad (2)$$

Note that GiI4 and GiI5 must be set to zero when 64 state mode is selected. For example, if g0 = 171, then G0I[7:1] = 0000111. Table 2 shows the 64 state codes selected for CODE = 0 or 1, corresponding to standard rate 1/2 and 1/3 convolutional codes. For rate 1/4, a code was selected from [3].

Similarly, we have

$$g_i(D) = 1 + g_i^1 D + g_i^2 D^2 + g_i^6 D^3 + g_i^7 D^4 + D^5 \quad (3)$$

for 32 state codes (SM = 1, GiI3 to GiI5 equal to zero) and

$$g_i(D) = 1 + g_i^1 D + g_i^6 D^2 + g_i^7 D^3 + D^4 \quad (4)$$

for 16 state codes (SM = 0, GiI2 to GiI5 equal to zero).

Viterbi Decoder

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

Theory of Operation

The Viterbi decoding algorithm [4] finds the most likely transmitted sequence given the received noisy sequence.

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

$$R_k^i = A((1 - 2y_k^i)/\sqrt{m} + n_k^i) \quad (5)$$

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. Figure 4 shows the signal sets for BPSK and QPSK. We have

$$\sigma^2 = \left(2mR\frac{E_b}{N_0}\right)^{-1} \quad (6)$$

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).

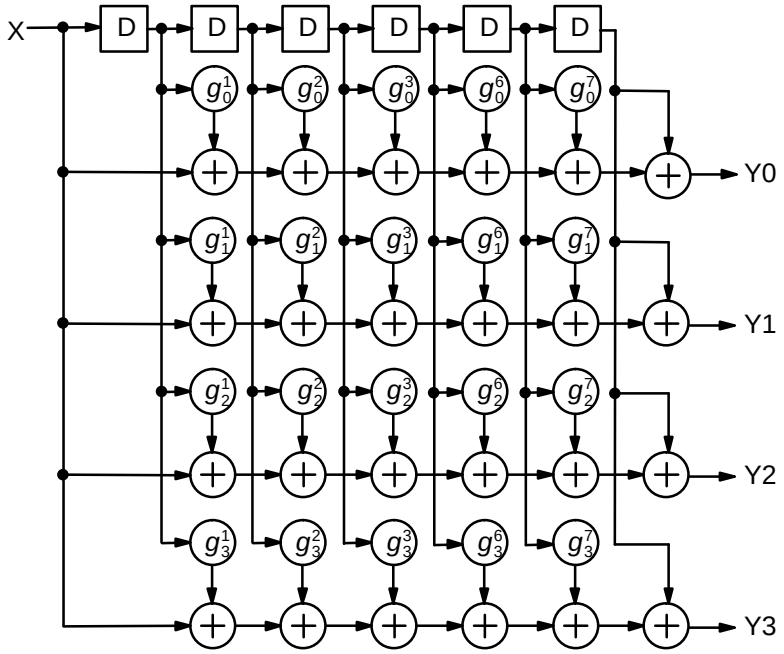


Figure 3: 64 state non–systematic convolutional encoder.

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 equal to d_k.

Due to quantisation and limiting effects the value of *A* should also be adjusted according to the received signal to noise ratio. A program called *cmap* for calculating the optimum values of *A* is included with the cores.

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 15 (which has a range between 15 and 16).

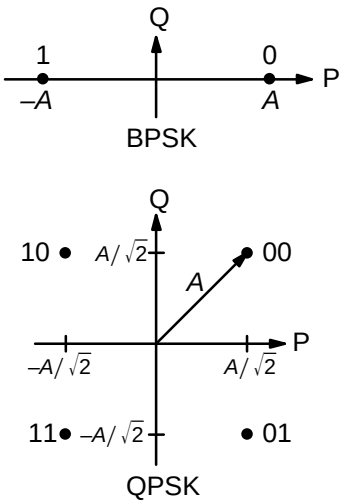


Figure 4: BPSK and QPSK signal sets.

Example 1: Rate 1/3 BPSK code operating at E_b/N_0 = 3 dB. From (6) we have σ^2 = 0.75178. Using cmap we have that *A* = 4.11.

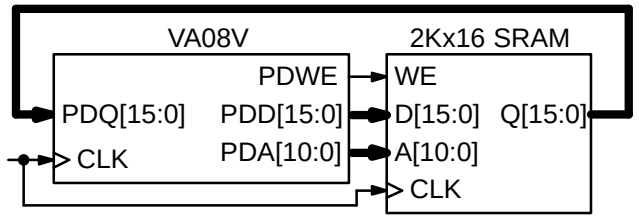
Example 2: Rate 1/2 QPSK code operating at E_b/N_0 = 4 dB. From (6) we have σ^2 = 0.19905. Using cmap we have that *A* = 27.66. Note that the amplitude in each dimension is $A/\sqrt{2}$ = 19.56.

Decoder Operation

The VA08V uses a finite traceback memory and is thus able to continuously decode data. The traceback depth is determined by DELAY and SM. Table 3 gives the minimum decoding depth, maximum decoding depth, and decoder delay for various combinations of DELAY and SM.

The path decision RAM is not included in the VA08V core. A 2Kx16 synchronous RAM is required and can be constructed from two RAMB16_S9 BlockRAMs. PDWE, PDA[10:0], PDD[15:0] and PDQ[15:0] are the path decision write enable, address, data in and data out signals of the RAM. Figure 5 shows how to connect a 2Kx16 synchronous RAM to these signals. This allows the path decision memory to be reused for other applications, e.g., the interleaver memory of a 3G turbo decoder.

If 64 state mode is only selected then the external RAM can be reduced to size 1Kx16. Address PDA10 is not used, so the address bus is PDA[9:0]. This can be implemented with one RAMB16_S18 BlockRAM. For 32 and 16 state



Inputs				Outputs	
WE	CLK	A[10:0]	D[15:0]	Q[15:0]	RAM(A)
0	↑	A	X	RAM(A)	RAM(A)
1	↑	A	D	X	D

Figure 5: Path decision RAM schematic and truth table.

codes, PDA[10:9] and PDA[10:8] are not used, respectively.

Table 3: Decoding depth and delay.

SM	DELAY	Min Depth	Max Depth	Delay
0	0	33	48	68
0	1	65	96	132
0	2	129	192	260
1	0	33	48	69
1	1	65	96	133
1	2	129	192	261
2	0	33	48	70
2	1	65	96	134
2	2	129	192	262
3	0	57	60	72
3	1	113	120	136

The depth address is given by PDA[6:0] for 256 state codes and by PDA[7:0] for 16, 32 and 64 state codes. For DELAY = 0 and SM = 3, PDA6 is not used. For SM < 3, PDA7 is not used for DELAY = 1 and PDA[7:6] are not used for DELAY = 0. The traceback memory can be correspondingly reduced for these configurations.

The VA08V uses eight ACS circuits in parallel. Thus, 32 clock cycles are required to perform 256 ACS operations or 8 clock cycles for 64 ACS operations. For 16 and 32 states, 8 clock cycles are still used even though the ACS circuits only require 2 and 4 clock cycles, respectively. This allows the minimum traceback depth to remain the same for 16, 32 and 64 states. An additional 2 clock cycle overhead is also required.

The decoder uses a rising edge detector circuit at the START input to start decoding the received data. If the high period of the START input is greater than the CLK period, the decoder will start decoding. To detect the next rising transition, the START input must be low for a least one CLK period.

This allows the decoder to be operated in synchronous or asynchronous operation. Synchronous operation requires 10 clock cycles per decoded bit for 16, 32 or 64 states or 34 clock cycles per bit for 256 states. Asynchronous operation requires 11 or 35 clock cycles per decoded bit, respectively.

Figure 6 shows the relationship between the START input and R0I, R1I, R2I, R3I, BLK_START, and BLK_END. In synchronous operation, these inputs must be valid from $2T_{cp}–T_{dsu}$ to $2T_{cp}+T_{dhd}$ after the rising edge of START (T_{cp}, T_{dsu}, and T_{dhd} are the decoder clock period, setup time, and hold time, respectively).

In asynchronous operation these signals must be valid from $T_{cp}–T_{dsu}$ to $2T_{cp}+T_{dhd}$ after the rising

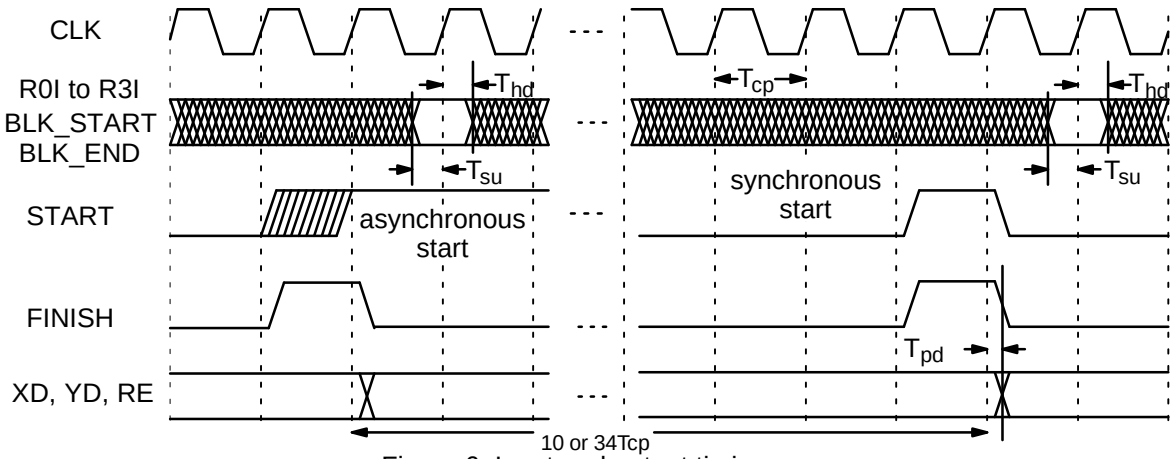


Figure 6: Input and output timing.

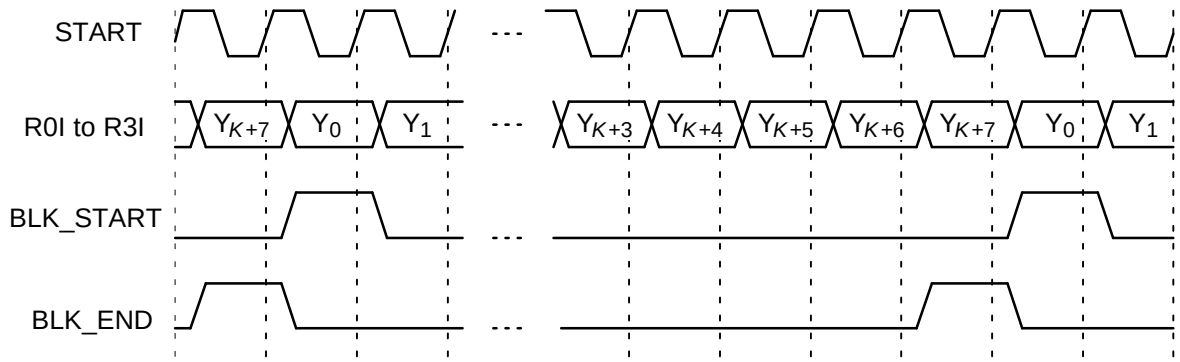


Figure 7: Decoder Block Timing (m = 8)

edge of START. Data must therefore change within one clock cycle after the rising edge of START.

The FINISH output goes high during the last clock cycle of the decoding operation. In continuous synchronous operation, the rising edges of START and FINISH should be coincident.

The decoded output XD, the re–encoded outputs YD[3:0] and estimated channel BER outputs RE[3:0] changes when FINISH goes low. RE[3:0] are obtained by exclusive ORing the appropriately delayed sign bit of the inputs with YD[3:0]. At low BER, these outputs can be used to give a good estimate of the channel BER.

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 BLK_START goes high the state metrics of the Viterbi algorithm algorithm are initialised to their appropriate starting values. WHEN BLK_END goes high, the traceback starts in the appropriate state. Figure 7 illustrates the timing for block encoded data for input data of length K terminated with an m = 8 symbol tail.

Data Format

The decoder uses 6–bit signed magnitude quantisation for R0I to R3I. Table 4 shows the 6–bit quantisation ranges. Note that 0 and 32 indicate the central dead zone and have the same range. Note that most analog to digital to digital (A/D) converters do not have a central dead zone. For maximum performance, we recommend that 7–bit A/Ds are used with the output converted to 6–bit so that the appropriate ranges are obtained.

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.

Table 4: 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

Punctured Code Operation

Manual puncturing can be performed by forcing R0I[4:0] to R3I[4:0] low. For example, rate 2/3 can be obtained by puncturing a rate 1/2 code with puncturing patterns of 11 for R0I and 10 for R1I. That is, R0I is not punctured, while R1I is forced low every other decoded bit.

Mode Selection

To minimise the decoder complexity, the MODE input can be used with the schematic symbols to select only those rates that are expected to be used. When MODE is low, the decoder can decode rate 1/2 and 1/3 codes. When MODE is high, the decoder can decode rate 1/2, 1/3, and

1/4 codes. MODE should only be connected to GND or VCC.

Serial Operation

When the SERIAL input is high, the decoder uses an internal serial to parallel converter to convert serially received data, for example in BPSK modulation, into parallel received data. The received clock should be input to START. This clock must not be divided down and must be equal to the received symbol rate. The received data must be valid from one to two CLK cycles after the rising edge of START and be input to R0I only.

The data corresponding to Y0 to Y3 is assumed to be received in this order. For example for rate 1/2, the data for Y0 is received first, followed by the data for Y1.

Note that FINISH only goes high at the end of each received code symbol. This consists of *n* received data symbols for a rate 1/*n* code.

Due to the serial to parallel operation, the decoder delay increases by *n*–1 received data periods.

Automatic Synchronisation

When SYNC_EN is high, automatic synchronisation to the coded symbol is enabled. This counts the number of state metric normalisations within the decoder. If the count exceeds the synchronisation threshold (SYNC_TH) before the end of the synchronisation period (SYNC_PD), SYNC_OUT will go high for one code symbol period. The normalisation counter is then disabled for one SYNC_PD, to allow the decoder to settle to its new synchronisation state. A new count is then started. If the threshold is not exceeded at the end of the SYNC_PD, the normalisation and period counters are reset, and a new count is started.

When SYNC_EN is high, SYNC_OUT is internally used by the decoder to change the synchronisation state. For serial operation, the code symbol period is increased by one received data period. This is performed only once, and causes the serial to parallel conversion to load one received data period later. With rate 1/2 Gray mapped QPSK operation, the received data is rotated by 0 or 90°, depending on the synchronisation state (0 or 1).

Note that if SYNC_EN is low, SYNC_OUT is not disabled (however, the internal synchronisation state is not allowed to change). This allows SYNC_OUT to be externally used to control the synchronisation state of an external synchronisation circuit.

With rate 1/2 operation at an E_b/N_0 of 4.2 dB (corresponding to a BER of 8.3×10^{-6} with DELAY = 1), the state metric normalisation rate is about 6.7×10^{-3}. When the decoder is out is sync though, the normalisation rate increases to about 4.7×10^{-2}. Thus, with a SYNC_PD of 128, a SYNC_TH of $128 \times 4.7 \times 10^{-2} = 6$ should provide a robust threshold. The average count when in sync is less than one. This should ensure that the decoder does not lose sync when it is in sync and that it quickly synchronises when out of sync.

CCSDS Operation

The VA08V core can also be used to decode the CCSDS [5] rate 1/2 64 state convolutional code. The CCSDS code is selected with CODE = 1, SM = 2, and N = 2. The CCSDS code also inverts Y1 to aid with demodulation. When the CCSDS input to VA08V is high, the sign bit of the received data corresponding to Y1 is also inverted, allowing the decoder to decode CCSDS transmitted data.

Note that there is no differential encoder used for the CCSDS encoder. Since the code used is 180° rotationally invariant (either for BPSK or Gray mapped QPSK), this implies the synchronisation circuit can not detect 180° rotations. This implies that the decoded data could be inverted due to a 180° rotation. This will need to be externally detected, so that non–inverted data is used.

Other inputs

The decode chip select (DCS) input is used to put the Viterbi decoder in a low power mode when low. Note that this is not a clock enable input. The decoder state is lost when DCS goes low.

The RST input when high synchronously forces all flip–flops low. This is useful for VHDL simulations where flip–flops are initially in an unknown state. The BLK_START signal should be high for the first received data in order to initialise the state metrics. The decoded output will be unknown until the unknown data in RAM is flushed out. The length of the unknown output data should be equal to the decoder delay.

Example

We give an example of how the VA08V can be used as a continuous rate 1/2 256 state QPSK decoder. The decoder is operated asynchronously with automatic synchronisation, with the received data clock input to START.

Figure 8 shows the VA08V configuration. The code used is $g_0 = 561_8 = 101110001_2 \rightarrow$ G0I[7:1] = 0001110_2 and $g_1 = 753_8 = 111101011_2 \rightarrow$ G1I[7:1] = 1010111_2. Since the code is invariant to

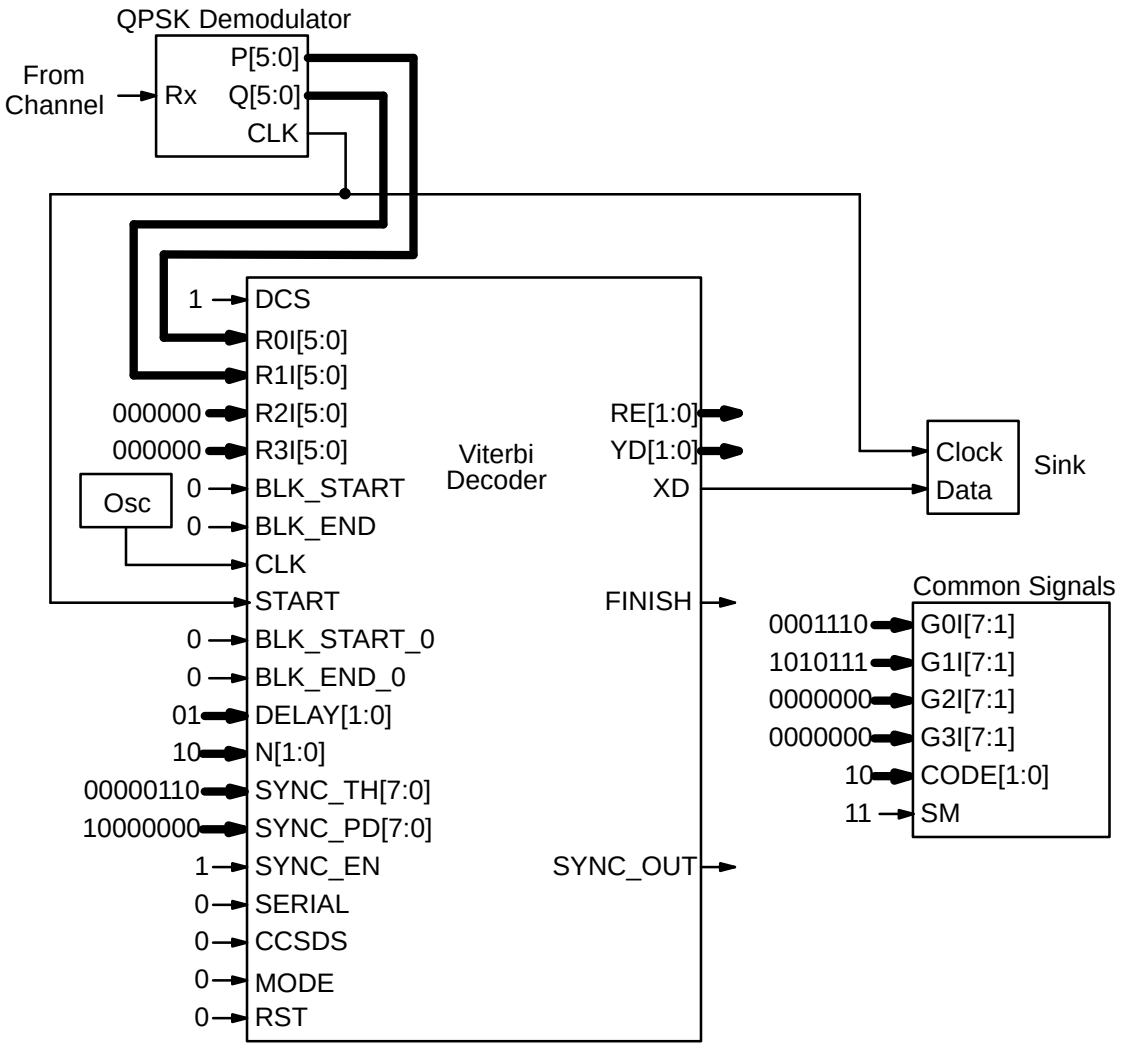


Figure 8: Block diagram of rate 1/2 QPSK codec.

180° phase rotations, differential encoding and decoding can be used if desired.

Simulation Software

Free software for simulating the VA08V Viterbi decoder in additive white Gaussian noise (AWGN) is available by sending an email to info@sworld.com.au with “va08vsim request” in the subject header. The software uses an exact functional simulation of the VA08V Viterbi decoder, including all quantisation and limiting effects.

Figure 9 shows the performance obtained for the standard rate 1/2 64 state convolutional code decoded by the VA08V Viterbi decoder with DELAY = 1, SM = 2, N = 2 and CODE = 1. The performance with both 3–bit and 6–bit input quantisation is shown.

Ordering Information

SW–VA08V–SOS (SignOnce Site License)
SW–VA08V–SOP (SignOnce Project License)
SW–VA08V–VHD (VHDL ASIC License)

All licenses include Xilinx VHDL cores. 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 version 5.2.0 Release 5, Sep. 2002.

[2] Third Generation Partnership Project 2 (3GPP2), “Physical layer standard for cdma2000 spread spectrum systems, Revision D,” 3GPP2 C.S0002–D Version 1.0, 13 Feb. 2004.

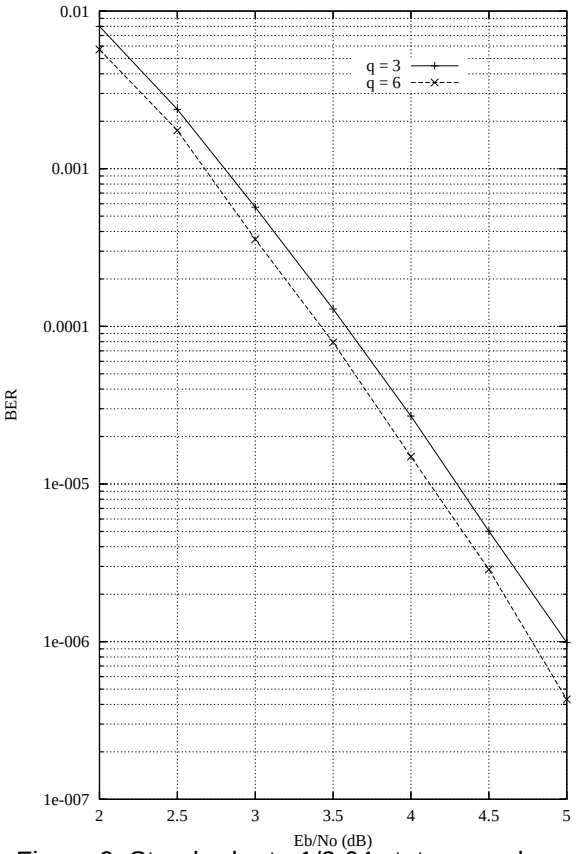


Figure 9: Standard rate 1/2 64 state convolutional code decoded by VA08V performance.

[3] P. J. Lee, "New short constraint length, rate 1/*N* convolutional codes which minimize the required SNR for given desired bit error rates," *IEEE Trans. Commun.*, vol. COM–33, pp. 171–177, Feb. 1985.

[4] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT–13, pp. 260–269, Apr. 1967.

[5] Consultive Committee for Space Data Systems (CCSDS), "Recommendations for space data system standards: Telemetry channel coding," CCSDS 101.0–B–6 Blue Book, Oct. 2002.

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

Version History

- 0.3 14 May 2001. VA08 preliminary product specification. 256 state only.
- 0.4 28 May 2001. Updated Virtex–E performance and complexity. Added CODE[2:0] input, G0O[7:1] to G3O[7:1] outputs and description for G0I[7:1] to G3I[7:1] inputs.
- 0.42 21 July 2001. Changed R0E to R3E outputs to RE[3:0]. Added number of BlockRAMs used in memory and path decision RAM schematic figure. Increased decoder speed.
- 0.70 26 May 2002. First official release. Changed name to VA08V. Increased Virtex–E speed and complexity. Added Virtex–II performance and complexity. Added SM input for 64 state decoder option. Deleted BIT and MCS file description.
- 1.0 29 June 2004. Deleted ENC08V convolutional encoder description. Added Spartan–3 performance and complexity. Changed DELAY input to DELAY[1:0] to allow 262 bit delay option for 64 state decoding.
- 1.01 18 January 2005. Updated Virtex–E, Virtex–II and Spartan–3 complexity.
- 1.02 21 February 2005. Added description for using smaller path decision memory.
- 1.03 26 May 2005. Added Virtex–II Pro and Virtex–4 performance and complexity.
- 1.10 26 April 2007. Added N[1:0] input for code rate selection, SYNC_TH[7:0], SYNC_PD[7:0], SYNC_EN inputs and SYNC_OUT output for automatic synchronisation, SERIAL input for

serial operation and CCSDS input space standard input option.

- 1.20 25 March 2008. Changed SM input to SM[1:0] for 16 and 32 state decoder options. Changed CODE[2:0] input to CODE[1:0] to allow more internal code polynomial selections using N[1:0] and SM[1:0]. Deleted Virtex–E and Spartan–II performance and complexity. Updated Virtex–II Pro, Spartan–3 and Virtex–4 complexity.
- 1.24 7 July 2008. Changed R0I[3:0] to R4I[3:0] four bits inputs to R0I[5:0] to R4I[5:0] six–bit inputs. Added Virtex–5 performance and complexity. Updated Virtex–II Pro, Spartan–3 and Virtex–4 performance and complexity. Added optional internal codes for rate 1/3 64 and 256 state decoder.
- 1.27 21 May 2009. Registered R0I[5:0] to R3I[5:0] inputs. Corrected VA08M name to VA08V.
- 1.28 18 November 2010. Deleted Virtex–II Pro performance and complexity. Improved Virtex–5 performance. Added Virtex–6 performance. Updated Virtex–4 and Virtex–5 complexity. Added description for using less than 6–bit quantisation.
- 1.29 21 January 2011. Added Version History. Corrected PDA[10:0] description for reduced path decision memory sizes.
- 1.30 10 August 2011. Corrected minimum decoding depth in Table 3.
- 1.31 10 April 2017. Added SYNC_OUT in Signal Descriptions. Added clarifications for automatic synchronisation and block operation. Deleted Spartan 3, Virtex 4 and Spartan 6 performance. Added Zync 7, Artix 7 and Kintex 7 performance. Updated Virtex 5 and Virtex 6 performance. Updated complexity. Deleted university license and EDIF core.
- 1.32 13 April 2017. Updated speed and complexity.