VA04D Features

- 16 state (memory *m* = 4, constraint length 5) tail biting Viterbi decoder
- Rate 1/5 (inputs can be punctured for higher rates)
- Optional or standard DVB–S2/DVB–S2X code polynomials
- Data length *K* from 4 to 32 bits
- Up to 382 MHz internal clock
- Up to 46 Mbit/s decoding speed (*K* = 16)
- 6–bit received signed magnitude data
- 1315 6–input LUTs
- 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.

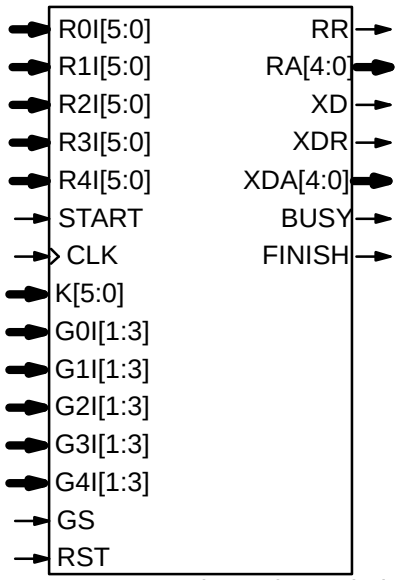

Figure 1: VA04D schematic symbol.

Introduction

The VA04D is a 16 state tail–biting error control decoder using the maximum likelihood Viterbi algorithm. The decoder is designed to decode the DVB–S2 [1] or DVB–S2X [2] standard rate 1/5 tail biting convolutional code. External code inputs and input data puncturing allow other 16 state tail biting codes to be decoded with data length *K* from four to 32 bits.

To reduce complexity with little performance degradation, the VA04D uses only a single Viterbi decoder with 16 add–compare–select (ACS) circuits working in parallel. A single external *K*x30 synchronous RAM is used for the input data.

The input data is read for 2*L*+*K* clock cycles, where *K* is the data length and *L* = 32 is the window training length. The decoder inputs the data in reverse order modulo *K* so as to minimise the decoder delay. The last *L*+*K* path decisions are stored in memory where a traceback is performed, taking an additional *L*+*K* clock cycles. The last *K* bits of the traceback are output as the decoded data. A pipeline delay of *D* = 5 clock cycles gives a total decoding time of 3*L*+2*K*+D = 101+2*K* clock cycles.

Figure 1 shows the schematic symbol for the VA04D 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. T_{cp} is the minimum clock period over recommended operating conditions. These performance figures may change due to device utilisation and configuration.

Signal Descriptions

BUSY	Decoder Busy
CLK	System Clock
FINISH	Decoder Finish
G0I–G5I	External Code
GS	External Code Select
	0 = DVB–S2/S2X polynomials
	1 = Use G0I to G4I
K	Data Length (4–32)
R0I–R4I	Received Data
RA	Received Data Address
RR	Received Data Ready
RST	Synchronous Reset
START	Decoder Start
XD	Decoded Data Output
XDA	Decoded Data Address
XDR	Decoded Data Ready

Code

Figure 2 gives a block diagram of a rate 1/5 16 state (*m* = 4) non–systematic encoder. X is the data input and Y0 to Y4 are the coded outputs.

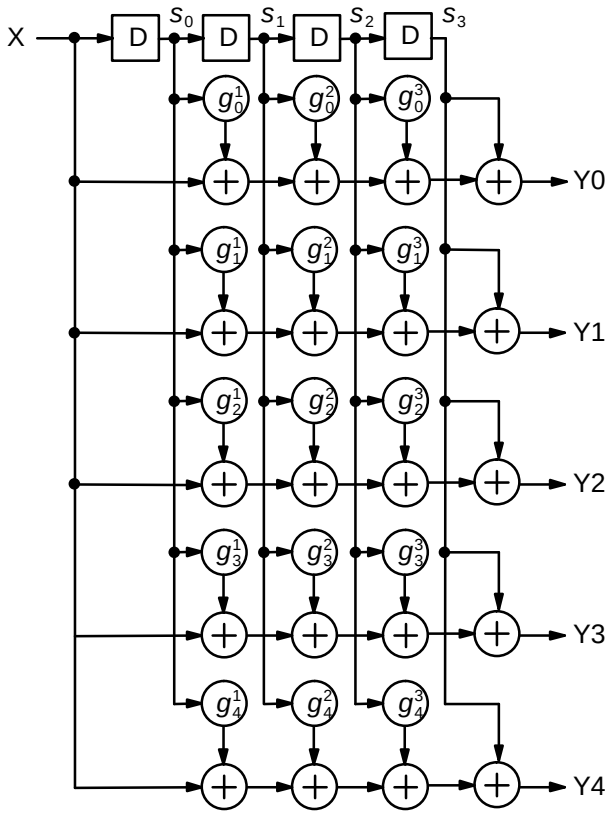

Figure 2: 16 state non–systematic convolutional encoder.

The code polynomial coefficients are GiIj = $g_i^j \in \{0, 1\}$, $0 \le i \le 4$, $1 \le j \le 3$.

Table 1: Performance of Xilinx parts.

Xilinx Part	T_{cp} (ns)	Data Rate (Mbit/s)		
		***K*=8**	***K*=16**	***K*=32**
XC5VLX30–1	4.572	14.9	26.3	42.4
XC5VLX30–2	3.914	17.4	30.7	49.5
XC5VLX30–3	3.480	19.6	34.5	55.7
XC6VLX75T–1	3.876	17.6	31.0	50.0
XC6VLX75T–2	3.424	19.9	35.1	56.6
XC6VLX75T–3	3.093	22.1	38.8	62.7
XC7Z010–1	5.554	12.3	21.6	34.9
XC7Z010–2	4.592	14.8	26.1	42.2
XC7Z010–3	4.103	16.6	29.3	47.2
XC7A35T–1	5.476	12.4	21.9	35.4
XC7A35T–2	4.496	15.2	26.7	43.1
XC7A35T–3	3.999	17.0	30.0	48.4
XC7K70T–1	3.502	19.5	34.3	55.3
XC7K70T–2	2.825	24.2	42.5	68.6
XC7K70T–3	2.612	26.1	46.0	74.2

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., $g_4 = 31_8 = 11001_2 \equiv g_4(D) = 1 + D + D^4$. This corresponds to G4I[1:3] = 100_2.

The DVB–S2/DVB–S2X standard is selected when GS = 0. It has code polynomials $g_0 = 25_8$, $g_1 = 27_8$, $g_2 = 33_8$, $g_3 = 37_8$ and $g_4 = 31_8$. When GS = 1, the external code inputs G0I to G4I are selected.

Tail biting is achieved by initialising the encoder shift register (without transmitting any coded bits) with the last $m = 4$ bits of the K data bits so that $s_3 = x_{K-4}$, $s_2 = x_{K-3}$, $s_1 = x_{K-2}$ and $s_0 = x_{K-1}$, where (s_3,s_2,s_1,s_0) is the encoder state and x_0 to x_{K-1} is the length K input data. The K data bits are then input to produce the $5K$ coded bits. No tail bits are transmitted.

Viterbi Decoder

The Viterbi decoder is designed to efficiently decode short length tail biting convolutional codes.

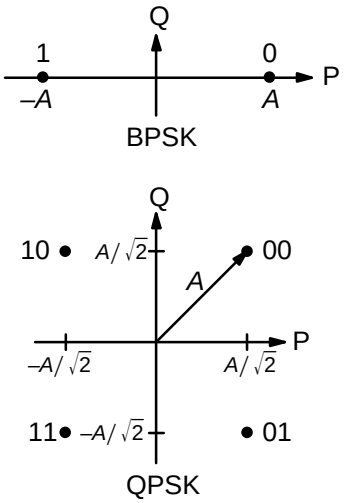

Figure 3: BPSK and QPSK signal sets.

Theory of Operation

The Viterbi decoding algorithm [3] 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

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

where A is the signal amplitude, $y_k^i \in \{0, 1\}$, i = 0 to 4 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 3 shows the signal sets for BPSK and QPSK. We have

$$\sigma^2 = \left(2mR\frac{E_b}{N_0}\right)^{-1} \quad (3)$$

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^i in two's complement notation is equal to y_k^i.

The value of A directly corresponds to the 6–bit signed magnitude inputs. The 6–bit inputs have 63 quantisation regions with a central dead zone. The quantisation regions are labelled from –31 to +31.

Due to quantisation and limiting effects the value of A should be adjusted according to the received signal to noise ratio. For example, for rate 1/5, we recommend that A = 10.7 be used. This value of A lies in quantisation region 11 (which has a range between 10.5 and 11.5).

Example 1: Rate 1/5 BPSK code operating at E_b/N_0 = 3.5 dB. From (3) we have σ^2 = 1.1167.

Decoder Operation

The optimum maximum likelihood decoder for a tail biting convolutional code requires 2^m = 16 Viterbi decoders for each of the 2^m identical start and end states of the code. The sequence with the smallest state metric (SM) is then chosen as the decoded sequence.

To reduce decoder complexity, a suboptimal algorithm is used. The input data is first input for L training symbols, followed by K symbols (the main sequence) and then L post training symbols. The training symbols ensure that the SMs are close to their correct values at the start of the main sequence. The post training symbols are used to ensure reliable path decisions are available at the end of the main sequence. For a large enough L, little performance degradation is achieved compared to the optimal algorithm.

If the symbols are input in a forward sequence, the traceback operation will output the decoded bits in a reverse sequence. To output the decoded bits in a forward sequence then requires a small output memory and an additional delay of K clock cycles. To avoid this reversing step, we instead input the data in reverse sequence. The traceback will then output the data in a forward sequence. A reverse trellis is used, which is obtained by time reversing the code polynomials, for example 10111 becomes 11101.

If the main sequence is input in reverse order as R_{K-1} down to R_0, then the traceback is output as X_0 to X_{m-1}. The L training and post training input symbols are then added to the main sequence in groups of K symbols, with the first symbol having address K–1+L mod K = L–1 mod K. For example, for K = 16, the first symbol has address 31 mod 16 = 15.

Figure 4 illustrates the Viterbi decoder input timing for K = 16. 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 Kx30. The received data ready signal RR goes high to indicate the data to be read from the address given by RA[4:0]. The BUSY signal remains high during decoding. The START signal is ignored during decoding, except for the last decoded bit that is output.

Figure 5 illustrates the Viterbi decoder output timing for K = 16. The decoded output XD is output

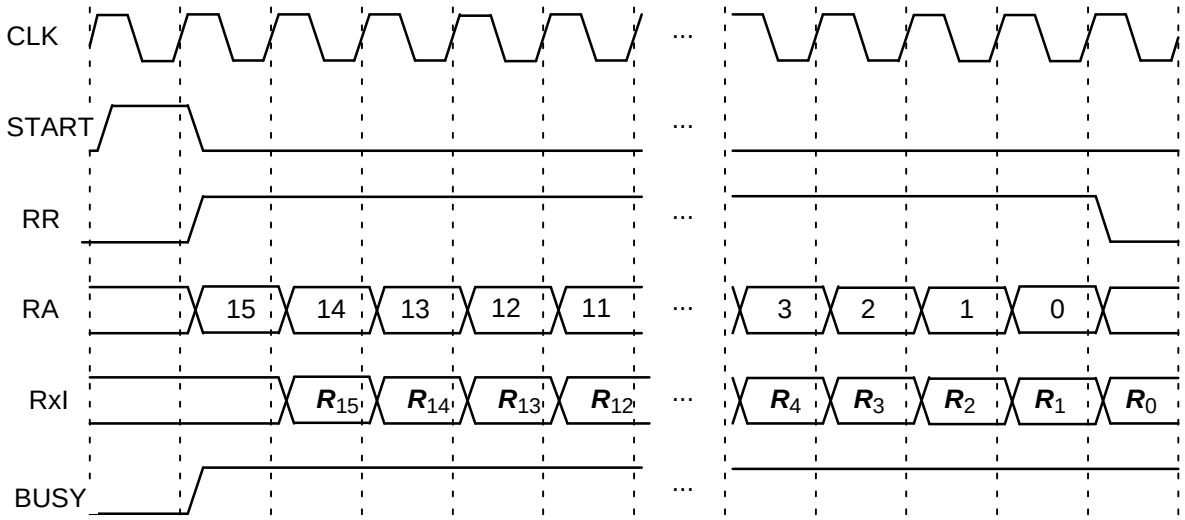

Figure 4: Viterbi Decoder Input Timing (*K* = 16).

while XDR is high with XDA[4:0] indicating the bit address. FINISH goes high for the last decoded bit.

Data Format

The decoder uses 6–bit signed magnitude quantisation for R0I to R4I. Table 2 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 (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).

Table 2: Quantisation for R0I to R4I.

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

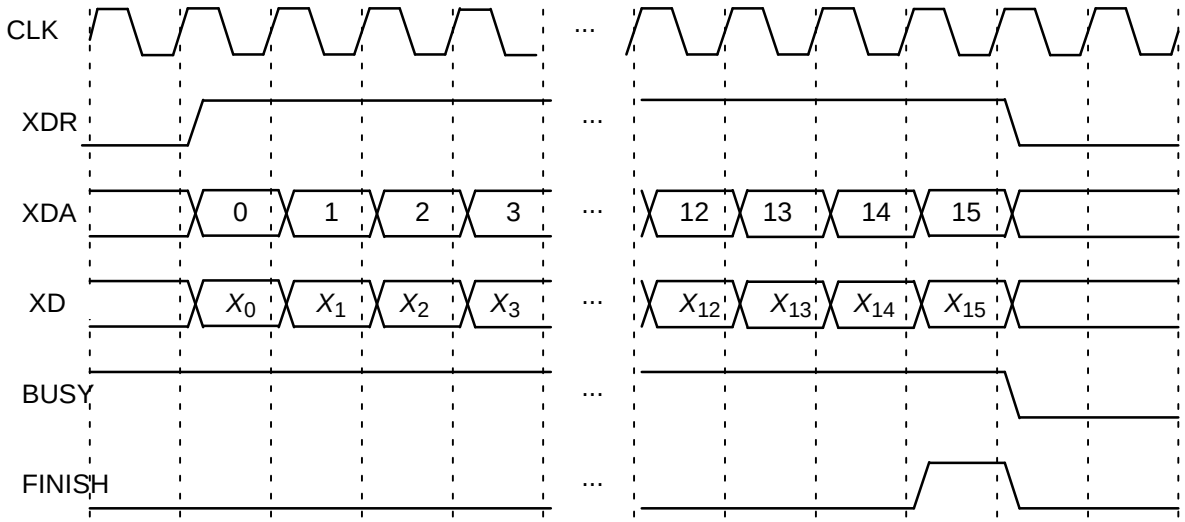

Figure 5: Viterbi Decoder Output Timing (*K* = 16).

Punctured Code Operation

Manual puncturing can be performed by forcing R0I[4:0] to R4I[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, R1I is forced low every other decoded bit and R2I to R4I are always punctured.

Other Inputs

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.

Decoder Speed

The decoding speed is given by

$$f_d = \frac{F_d}{2 + (3L + D)/K} \qquad (4)$$

where F_d is the internal clock speed, K is the data length, $L = 32$ is the training length and $D = 5$ is the pipeline delay. For example, if $K = 16$ and $F_d = 300$ MHz, the decoding speed is 36.0 Mbit/s.

Simulation Software

Free software for simulating the VA04D Viterbi decoder in additive white Gaussian noise (AWGN) is available by sending an email to info@sworld.com.au with “va04dsim request” in the subject header. The software uses an exact functional simulation of the VA04D Viterbi decoder, including all quantisation and limiting effects.

Figure 6 shows the bit error rate (BER) and frame error rate (FER) performance obtained for the standard rate 1/5 16 state tail biting convolutional code decoded by the VA04D Viterbi decoder for $K = 16$ and $A = 10.7$. No puncturing is performed.

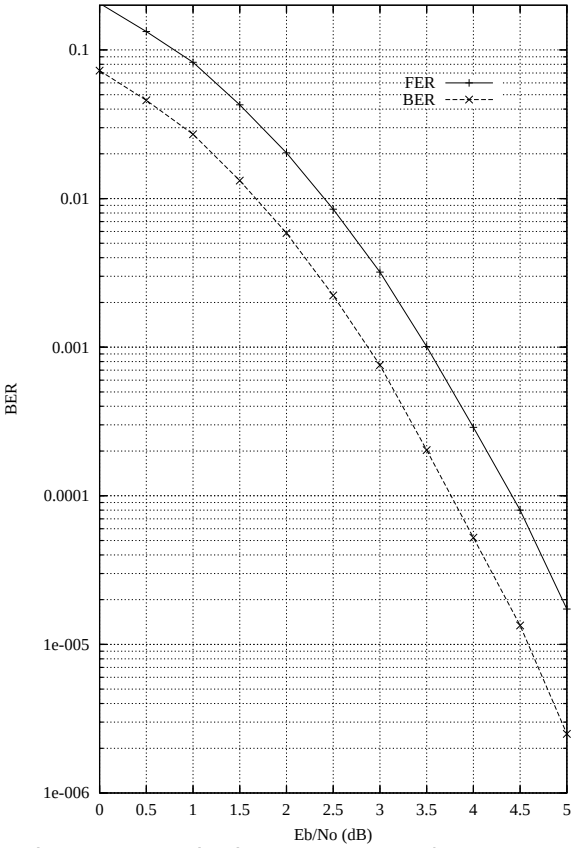

Figure 6: Standard 16 state, rate 1/5, $K = 16$ tail biting convolutional code performance.

Ordering Information

SW–VA04D–SOP (SignOnce Project License)
SW–VA04D–SOS (SignOnce Site License)
SW–VA04D–VHD (VHDL ASIC License)

All licenses are perpetual and include Xilinx VHDL cores, unlimited instantiations, free updates for one year and free lifetime support. SOP allows the core to be used for a specified project. SOS allows unlimited projects for a specified development site. VHD includes a VHDL core customised for your ASIC.

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] ETSI, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 1: DVB–S2,” ETSI EN 302 307–1 V1.4.1, Nov. 2014.

[2] ETSI, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 2: DVB–S2 Extensions (DVB–S2X),” ETSI EN 302 307–2 V1.1.1, Feb. 2015.

[3] 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.

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

Version History

- 0.00 28 July 2017. VA04D preliminary product specification.
- 0.01 7 August 2017. Deleted DCS input. Added GS and G0I to G5I inputs. Increased range of *K* from 8–16 bits to 4–32 bits.
- 1.00 18 August 2017. First release. Added decoder complexity and performance values. Updated channel performance figure. Decreased pipeline delay from *D* = 6 to *D* = 5. Corrected RA[4:0] start address.