
Chord™ Signaling Primer

A multi-wire signaling approach that is a generalization of differential signaling.

©2019 Kandou Bus



Build: May 18, 2020

Contents

Background	2
Differential signaling	2
Chord signaling	3
Signaling technique comparison	4
PAM signaling	5
Previous multiwire signaling approaches	5
ENRZ and EP3L signaling	8
Comparison	10
Chordal codes	13
Summary	17

Background

Companies connect chips in electronic devices via multiple wires. This is because the required throughput immediately saturates the available bandwidth of one, (or even a few) wires.

Chord signaling coordinates the traffic on these already connected wires to better combat noise. This increases the signal to noise ratio (SNR), which in turn both:

- Increases power efficiency, and
- Increases transmission speed

It is a generalization of already existing differential signaling technology, which is becoming obsolete due to faster communication requirements over more challenging channels.

Chord signaling theory defaults to differential signaling when transmitting data over two wires. When we increase the number of wires however, we are able to keep the same advantages of differential signaling whilst improving pin efficiency.

Differential signaling

Differential signaling sends one bit of information over two wires. One wire carries the signal, the other carries the inverted signal. These are known as complementary voltage signals. A comparator at the receiving end of the signals validates these signals.

This is half as efficient as transmitting one bit of information over each wire. This means that we need to double the transmission frequency, or that that differential pin-efficiency is 1/2.

This type of signaling is desirable for high speed links however, as it:

1. Combats common mode noise and electromagnetic interference (EMI) as the wire value sum across the two wires is zero.
2. Eliminates (mostly) simultaneous switching output (SSO) noise as the positive flow on the wires is always the same, regardless of the bit transmitted.
3. Eliminates reference-noise as the signal is detected using a reference-less comparator.
4. Reduces intersymbol-interference as the comparator slicer input signals only has two complementary values.
5. Rejects common mode noise as if the same value is added to the wires, this common value is canceled by the comparator.
6. Improves decision feedback equalization (DFE) as the comparator output is the transmitted bit.

Chord signaling

With chord signaling we can send more bits of data over more wires whilst keeping all of the above desirable properties. For example, with our CNRZ chord signaling method, we can send five bits of data over six wires. This is the signaling method used in our Glasswing products.

Chord signaling is only part of the solution however. Apart from the signaling method itself, it requires an integrated approach that includes the circuits, architectures, practical simulations, and diagnostic applications. At Kandou Bus, we construct, analyze, and implement chord signaling in practical products.

See below for a comparison of our chord signaling versus other similar schemes:

Signaling	Pin-efficiency	1	2	3	4	5	6
NRZ	0.5	o	o	o	o	o	o
PAM-4	1	o	x	x	x	o	-
PAM-X	≤ 1	o	o	o	o	o	x
Binary classical multiwire signaling	< 1	o	o	o	o	o	x
Non-binary classical multiwire signaling	≤ 1	o	o	o	x	o	x
Reference-less chord signaling	Approx. 1	o	o	o	o	o	o
Referenced chord signaling	Approx. 1	o	-	x	x	o	-

Signaling technique comparison

In this section we compare different signaling techniques on four wires against each other. The channel model for these comparisons is extracted using connectors, packages, and PCB traces and has the response as follows:

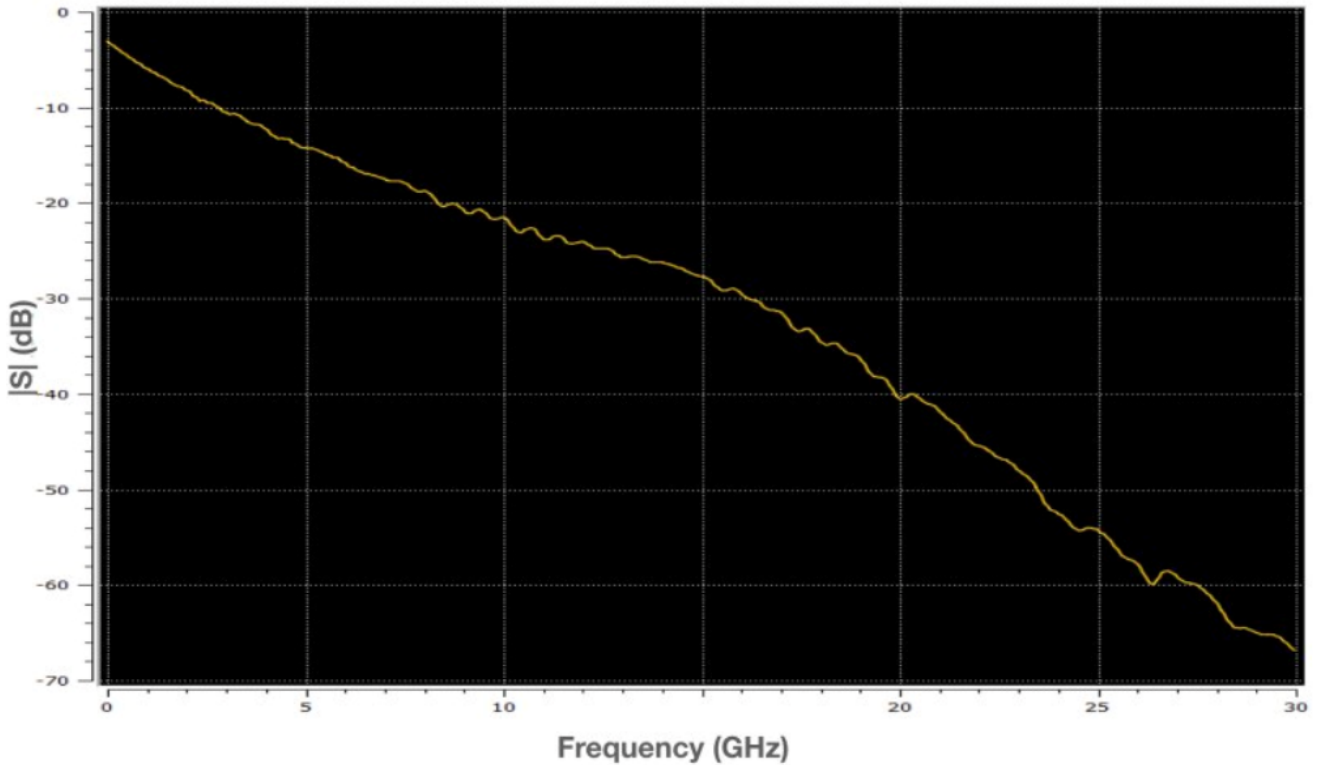


Figure 1: The parameters below apply for all simulations. For each signaling method in the comparison, we record the parameters of the worst eye diagram (in terms of vertical and horizontal opening).

- Throughput of 14 Gbps per wire
- Continuous linear time equalization up to 12 dB, optimized
- TX Finite Impulse Response (FIR) with one pre-cursor and one post-cursor tap, optimized
- Five taps of decision feedback equalization (DFE).
 - Note that some of the non-chord signaling schemes described below do not have efficient DFE architectures to our knowledge. Nevertheless, we assume perfect DFE equalization to give those schemes the benefit of the doubt.
- 600 mV single-ended peak-to-peak transmit amplitude

PAM signaling

The most straightforward way to increase the throughput on communication wires is to use multiple levels on a pair of wires. This technique is commonly called *Pulse Amplitude Modulation* (PAM). If there are n levels used, then the scheme is more precisely called PAM- x . For example,

- Using differential PAM-4 signaling we can transmit two bits on a pair of wires.
- Using differential PAM-8 signaling we can transmit three bits on a pair of wires.
- Using differential PAM-16 signaling we can transmit four bits on a pair of wires.

Signaling	Baud rate (GBaud)	Horiz. opening (ps)	Vert. opening (mV)	ISI ratio
PAM-2	28	27	63	1
PAM-4	14	26	50	3
PAM-8	9.33	20	19	7
PAM-16	7	11	9	15

PAM signaling

Lowering the baud-rate does not necessarily improve the performance. This is due to inter-symbol interference (ISI) and the concept of ISI-ratio. This is a Figure of Merit (FoM) describing the susceptibility of a signaling scheme to ISI. The ISI-ratio is a number ≥ 1 , and the larger it is, the more susceptible is the signaling scheme to ISI. It is also a proxy for the size of the alphabet at the input of the slicer for the given comparators. The ISI-ratio of PAM- X signaling is $X-1$, for any $X \geq 2$.

Previous multiwire signaling approaches

Almost all previous multiwire signaling schemes use a type of coding that David Slepian introduced in his 1965 *Permutation Modulation* paper. A permutation modulation code consists of all the permutations of a base vector (a_1, \dots, a_n) . If the coordinates of the transmitted vector are perturbed independently by additive white Gaussian noise, then Slepian proves that the *maximum likelihood decoder* for this code is the one that ranks the entries and computes from that the permutation that transforms the base vector to the received one. For example, if the base vector is $[1, 0, 0, -1]$ and the received vector is $[0.1, -1.4, 0.3, 0.9]$, then the ranking information recovers the vector $[0, -1, 0, 1]$.

The permutation modulation code generated by the base vector (a_1, \dots, a_n) is denoted by $PM([a_1, \dots, a_n])$. The number of codewords of this code is $n!/c_1! c_2! \dots c_t!$ if the base vector has t distinct coordinates, each appearing with multiplicity c_1, c_2, \dots, c_t . For example, the number of codewords of $PM([1, 0, 0, -1])$ is $4!/2!1!1! = 12$.

We can compute the complete ranking in a variety of ways, though for chip-to-chip communication the easiest is to compute all pairwise comparisons of wire values and deducing the ranking from that. Care is required when comparing two values that are equal in the base vector as the outcome at the receiver can be anything, even if the channel is perfectly noiseless. This is because the behavior of electronic comparison circuit is undefined when the incoming values are too close. In previous literature, these comparisons produce “don’t care” results. This means that the algorithm which computes the ranking should not take into account the outcome of this comparison.

For example, suppose that we are using PM([1,0,0,-1]) together with all six pairwise comparators. For now, we denote a comparator comparing wires with indices a and b by a:b. If the value on wire a is greater than that on wire b, then we denote this by a +, if it is smaller by a -, and if they are equal, then we denote it by X. The following table summarizes the results:

Codeword	0:1	0:2	0:3	1:2	1:3	2:3
[1,0,0,-1]	+	+	+	x	+	+
[1,0,-1,0]	+	+	+	+	x	-
[1,-1,0,0]	+	+	+	-	-	x
[0,1,0,-1]	-	x	+	+	+	+
[0,1,-1,0]	-	+	x	+	+	-
[0,0,1,-1]	x	-	+	-	+	+
[-1,0,0,1]	-	-	-	x	-	-
[-1,0,1,0]	-	-	-	-	x	+
[-1,1,0,0]	-	-	-	+	+	x
[0,-1,0,1]	+	x	-	-	-	-
[0,-1,1,0]	+	-	x	-	-	+
[0,0,-1,1]	x	+	-	+	-	-

Table notes:

- The sum of the wire values across all four wires is 0.
- The positive flow on the wires is always the same.
 - This is true for all permutation modulation codes satisfying the previous condition.
- The signal is detected using reference-less comparators.
- The comparators eliminate common mode noise across all four wires.
- Any two codewords have the property that they differ in at least one none-x position.

- This means that the comparators are able to distinguish the codewords.
- If you leave out any of the comparators, then this condition is not satisfied anymore. (Try it!)
 - For example, if you leave out the last comparator 2:3, then the first two codewords cannot be distinguished anymore since they only differ in positions where one of them has an x.

Permutation modulation codes can increase the throughput per wire compared to differential signaling. For example, since $PM([1,0,0,-1])$ has 12 codewords, it can send 3.5 bits over an interface of four wires, which is 75% more than differential signaling. The largest permutation modulation code on four wires has 24 codewords, hence can send 4.5 bits over an interface with 4 wires. This is 2.25 times more data than differential signaling. The code is $PM([1,1/3,-1/3,-1])$, and has quaternary values on the wires like PAM-4 signaling.

Signaling	Baud rate (GBaud)	Horiz. opening (ps)	Vert. opening (mV)	ISI ratio
$PM([1,1,-1,-1])$	22.4	18	41	1
$PM([1,0,0,-1])$	16	28	36	2
$PM([1,-1,1/3,-1/3])$	12.44	18	25	3

The main issue with permutation modulation codes for chip-to-chip communication is their susceptibility to ISI noise, and possibly crosstalk (depending on the routing of the communication wires).

Where does the ISI come from? This is because at the input of the slicers within the comparators, values of different magnitudes are present. The ratio between the largest and the smallest possible magnitudes is essentially the ISI-ratio of the comparator. We show how the ISI-ratio is calculated without going into the mathematical details of the derivation of this algorithm:

Think of a comparator as a circuit that takes the difference of the values on its two legs, followed by a slicer which compares the difference to a preset value (and determines the bit). Thinking this way, each slicer sees at its input the values -2, -1, 0, 1, and 2, wherein the value 0 is not really taken into account for the decision. For example, the comparator 1:2 sees the values ± 2 on the codewords $\pm[1,-1,0,0]$, the values ± 1 on the codewords $\pm[1,0,-1,0]$, and the value 0 on the codeword $[0,0,-1,1]$. The ratio of the magnitudes it sees is therefore $2/1=2$. This number is the ISI-ratio of this comparator. By symmetry all comparators of $PM([1,0,0,-1])$ have the same ISI-ratio 2.

It may seem that there would have been an easier way to look at the ISI effect on this code: the symbols on the wires are ternary, and hence we expect to see the ISI caused by ternary signaling. It turns out that for PM-codes this answer is accidentally true. In fact, the theory of Chord signaling™ reveals that the number of levels on the wires does not have a direct effect on the ISI. It is only the number of levels present at the slicer, and more importantly the ratio between the largest and the smallest magnitude, which is important. The next section provides some more examples.

ENRZ and EP3L signaling

The ENRZ code consists of the codewords $\pm[-1,1/3,1/3,1/3]$, $\pm[1/3,-1,1/3,1/3]$, $\pm[1/3,1/3,-1,1/3]$, and $\pm[1/3,1/3,1/3,-1]$. The comparators are $\text{av}(0,1):\text{av}(2,3)$, $\text{av}(0,2):\text{av}(1,3)$ and $\text{av}(0,3):\text{av}(1,2)$, where $\text{av}(a,b)$ is the average value of values on wires a and b. The code has eight codewords, so we can send three bits in every UI. This means that the pin-efficiency of this code is 0.75, which is 50% more than that of NRZ. On the other hand, it is only 75% of the pin-efficiency of PAM-4 signaling. So, without further investigation, we expect ENRZ to perform worse than PAM-4.

However, this code has a very interesting property: Its ISI-ratio is 1 (whereas that of PAM-4 is 3). To see this, we compute the values of the comparators before the slicer; that is, we compute the difference of the values at the two legs of the comparators for all the codewords, the same way we did that for the code $\text{PM}([1,0,0,-1])$ above.

Codeword	$\text{av}(0,1)-\text{av}(2,3)$	$\text{av}(0,2)-\text{av}(1,3)$	$\text{av}(0,3)-\text{av}(1,2)$
$[-1,1/3,1/3,1/3]$	-2/3	-2/3	-2/3
$[1/3,-1,1/3,1/3]$	-2/3	+2/3	+2/3
$[1/3,1/3,-1,1/3]$	+2/3	-2/3	+2/3
$[1/3,1/3,1/3,-1]$	+2/3	+2/3	-2/3
$-[-1,1/3,1/3,1/3]$	+2/3	+2/3	+2/3
$-[1/3,-1,1/3,1/3]$	+2/3	-2/3	-2/3
$-[1/3,1/3,-1,1/3]$	-2/3	+2/3	-2/3
$-[1/3,1/3,1/3,-1]$	-2/3	-2/3	+2/3

Interestingly, for every comparator, the magnitude of the values seen before the slicer is the same regardless of the codeword. This is not the case for PAM-4 signaling, or in fact for any PAM-X signaling with $X > 2$. This property is key to the superior ISI performance of ENRZ and manifests itself in the superior horizontal opening of the eye, as we will see later.

A bird's eye view of the architecture ENRZ is given in the figure below. The values on the wires are quaternary, whereas the values before the slicer are binary. The comparators directly produce the output bits, without further decoding.

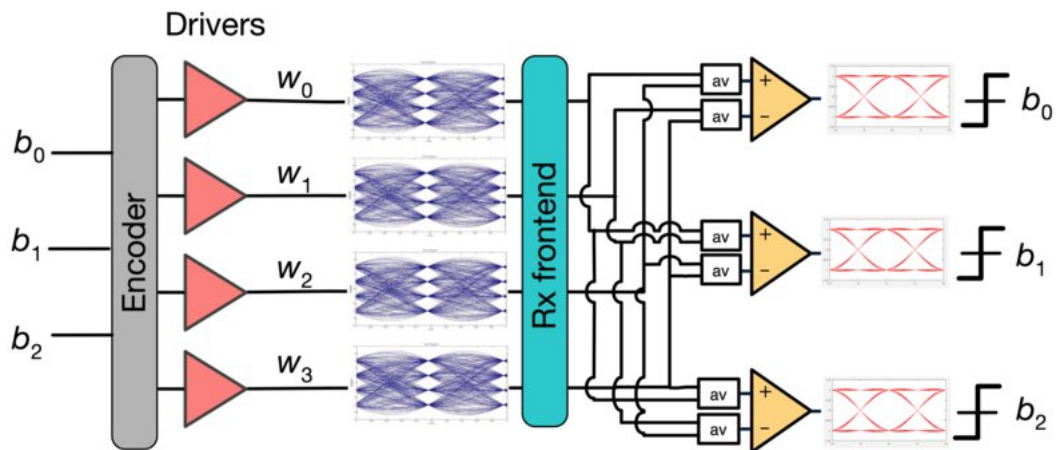


Figure 2: ENRZ architecture

High level architecture of a link based on ENRZ signaling. Note that while the wires carry quaternary signals, only binary signals are observed at the input to the slicers, leading to superior ISI performance.

Therefore, ENRZ satisfies all the properties of differential signaling stated above:

1. The sum of wire values across the wires is zero: YES
2. The positive flow on the wires is always the same, regardless of the bit transmitted: YES, it is +1.
3. The signal is detected using reference-less comparators: YES.
4. The signals at the input of the comparator slicer have only two complementary values: YES.
5. The comparator eliminates common mode noise: YES, a common mode on all 4 wires is eliminated by the comparators.
6. The output of the comparators are the transmitted bits: YES, no further decoding is necessary.

ENRZ is not just limited to NRZ. For any PAM-X signaling scheme, there is an equivalent EPAM-X signaling on four wires, yielding 1.5 times the throughput of PAM-X while keeping the ISI-ratio the same. For example, Kandou's EP3L code consists of 16 codewords with five levels, and can be decoded using the same comparators as ENRZ, followed by a PAM-3 slicer. A bird's eye view of the architecture of this link is given below:

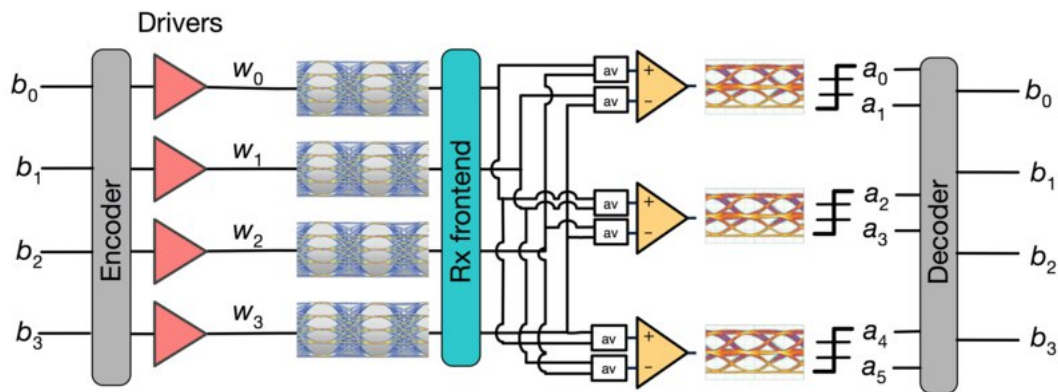


Figure 3: EP3L architecture

High level architecture of a link based on EP3L signaling. Note that while the wires carry five-level signals, ternary signals are observed at the input to the slicers. These are detected via PAM-3 slicers, each providing two bits. The final four bits are calculated by a very efficient decoder. The pin-efficiency of this system is 1.

The ISI-ratio of EP3L is 2, while it has the exact same pin-efficiency as PAM-4. This means that the eye diagrams associated with this signaling are expected to produce a larger horizontal opening. Indeed, this is the case, as can be seen in the following table.

Signaling	Baud rate (GBaud)	Horiz. opening (ps)	Vert. opening (mV)	ISI-ratio
ENRZ	18.75	44	40	1
EP3L	14	35	36	2

The additional “6 dB” version of ENRZ is obtained by adding a 6 dB gain stage to the RX frontend, effectively changing the averaging to a summation. Since ENRZ does not have any references, there are no linearity issues associated with this gain stage. This is not true for other signaling schemes in which the slicer has references, such as PAM-X, or EP3L.

Comparison

The following figure summarizes the above results. The horizontal axis corresponds to the horizontal opening (in pico-seconds) and the vertical axis corresponds to the vertical opening (in mV). Both ENRZ and EP3L signaling schemes exhibit the best horizontal opening. In addition, ENRZ exhibits also the best vertical opening. In particular, both ENRZ and EP3L are better than PAM-4 signaling in this case.

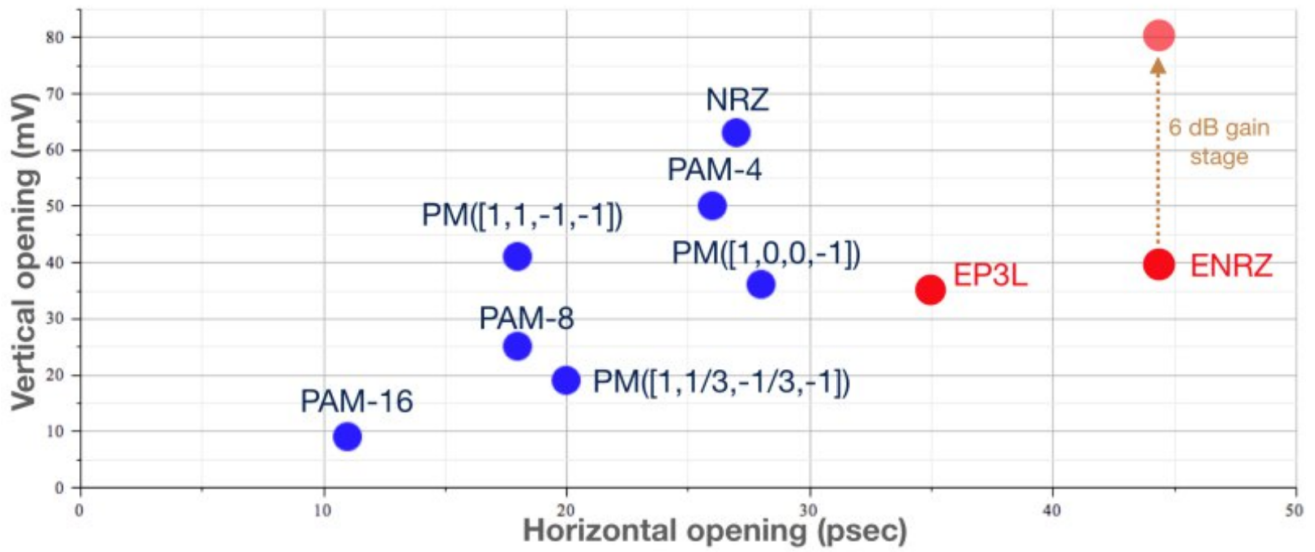


Figure 4: Horizontal and vertical eye openings for the examples above.

The figure does not show the relative advantage of ENRZ for a large range of throughputs. In fact, if we sweep the per-wire-speed from 7 Gbps all the way to 28 Gbps (corresponding to the speeds of the CEI-56G standards), then we obtain the following tables:

Throughput per wire	NRZ	ENRZ	PAM-4	EP3L
7 Gbps	177 mV	228 mV	95 mV	87 mV
14 Gbps	63 mV	80 mV	50 mV	37 mV
21 Gbps	13 mV	38 mV	23 mV	20 mV
28 Gbps	0.6 mV	14 mV	10 mV	9 mV

Throughput per wire	NRZ	ENRZ	PAM-4	EP3L
7 Gbps	64 ps	95 ps	65 ps	82 ps
14 Gbps	27 ps	44 ps	26 ps	35 ps
21 Gbps	11 ps	24 ps	13 ps	19 ps
28 Gbps	1 ps	13 ps	6 ps	10 ps

As we see, ENRZ has the best performance among these signaling methods. For high speed, its horizontal opening is almost twice that of PAM-4 even though its fundamental frequency is higher. Moreover, since ENRZ is

inherently a binary signaling method, it is much better compatible with advanced equalization schemes such as decision feedback equalization than PAM-4 is.

A graphical representation of this table is given by the following two figures:

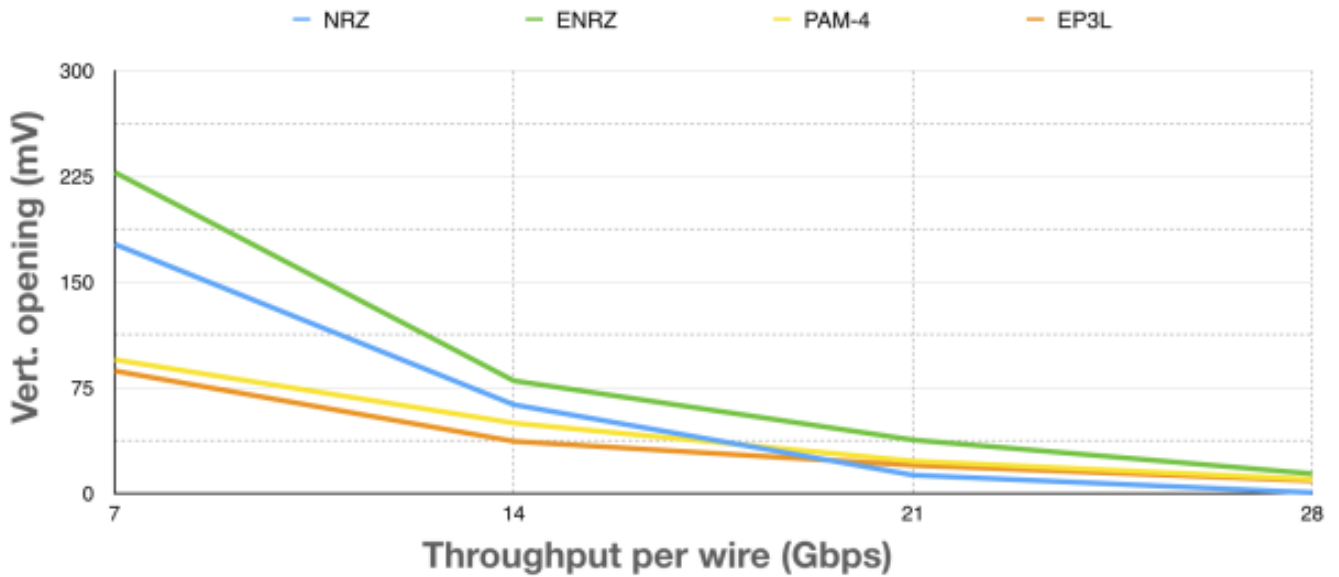


Figure 5: Vertical opening of various signaling schemes as a function of the throughput per wire.

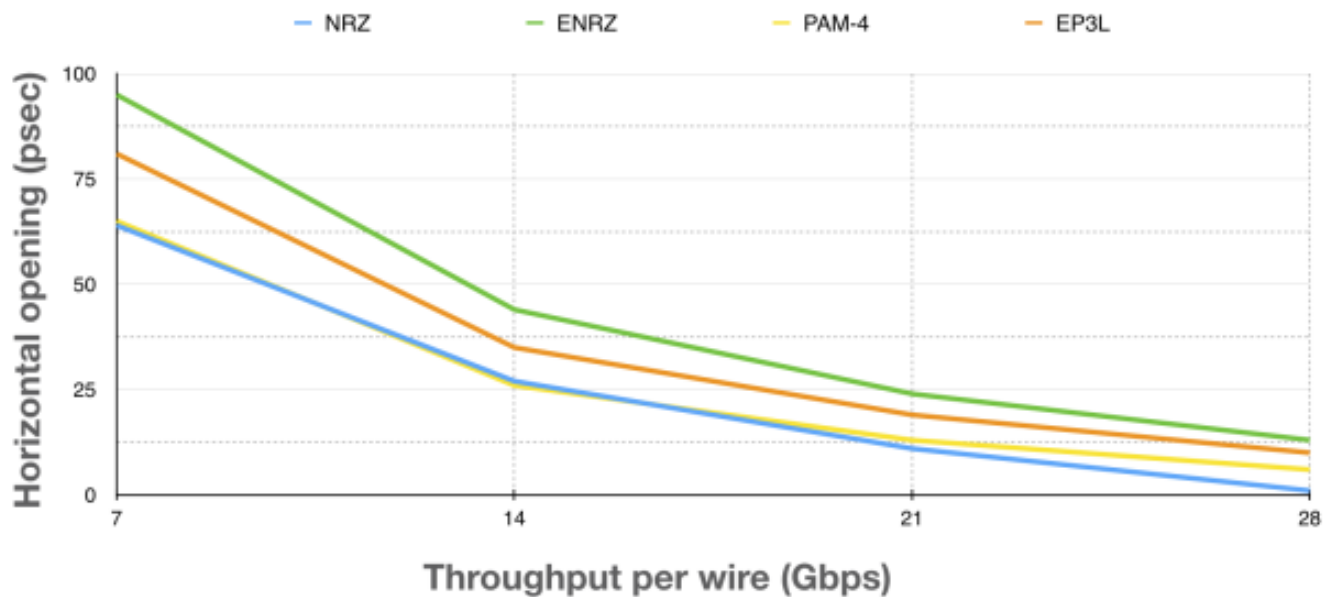


Figure 6: Horizontal opening of various signaling schemes as a function of the throughput per wire.

Chordal codes

To answer this, we need to clarify what a chordal code is. We find that the best way of describing a chordal code is a geometric view: in a first approximation, a chordal code consists of a number of points in the n -dimensional space with coordinates between -1 and $+1$, together with a number of affine hyperplanes that separate these points. The hyperplanes correspond to the comparators (and if they are affine, then they need references), and the points correspond to the codewords that are transmitted on the collection of n wires. We didn't really articulate exactly what it means for the comparators to separate the points; an exact definition will make things a bit more complicated, as it needs to take care of situations that we denoted by "don't cares" above. But for now, perhaps the figure on the left is a good way of thinking of hyperplanes separating the points: the hyperplanes define "chambers", and we should not have two points in the same chamber. In the figure on the left, the chordal code has 24 codewords, and 6 comparators. Everything is in three dimensions, so this corresponds to transmitting 24 codewords (4.5 bits) on three wires, and detecting them with 6 comparators. Though it is not at all clear from this description, this figure is completely equivalent to the code $PM([1,1/3,-1/3,-1])$. The theory of chordal coding, which we are not going to lay down here because of its complexity, provides the equivalence in a very natural way.

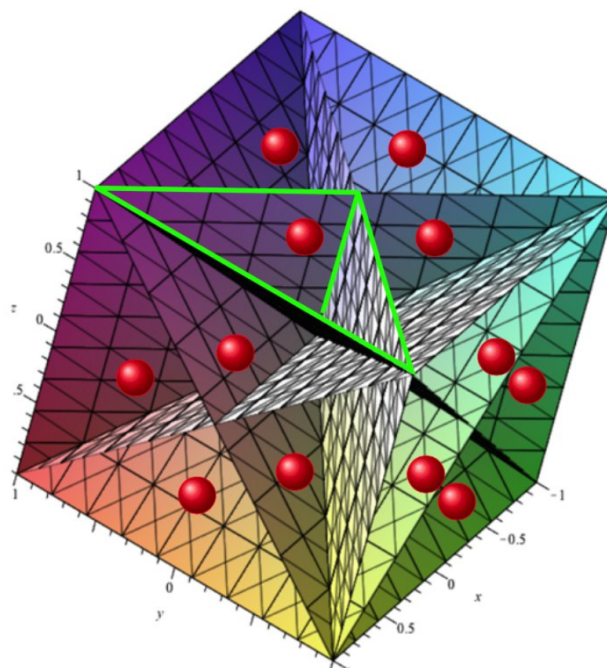


Figure 7: 24 points in the three dimensional space separated by six planes

The geometric view also lets us describe what we mean by the ISI-ratio of a comparator: it is the ratio of the distance of the closest point to the comparator and the distance of the furthest point to that comparator. As can be seen in the above figure, NRZ has ISI-ratio 1: both points of its code have the same distance from its comparator. The situation is more difficult for PAM-4, and this is because the definition of ISI-ratio has to be

modified when there are comparators with references. In that case, the ISI-ratio of any comparator is the same as the one passing through 0, i.e., the one that has no reference. With this modified definition, it can be seen that the ISI-ratio of PAM-4 is 3: the top left point has 3 times the distance from the middle line as any of the two inner points.

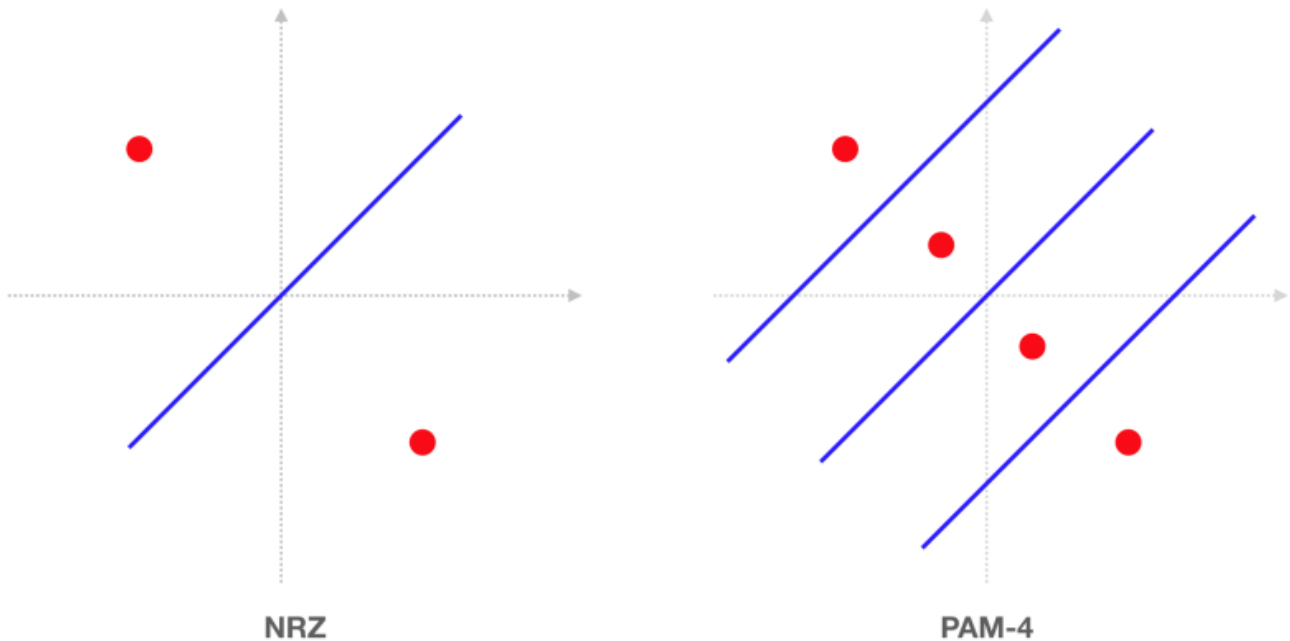


Figure 8: Geometric view of NRZ and PAM-4

The geometric view creates new chordal codes which have not been investigated before, like the following two-wire codes with 6 and 8 codewords (and 3 and 4 comparators, respectively):

As can be seen, the code C6 has ISI-ratio 2, since for every one of its comparators the closest point to the comparator has distance $1/2$, whereas the furthest point has distance 1. In fact, the full theory of chordal codes shows that this code is functionally equivalent to the code $PM([1,0,-1])$; The code C8 has ISI-ratio $1+\sqrt{2}$, since the closest point to every comparator has distance $\sin(\pi/8)$ and the furthest point has distance $\sin(5\pi/8)$, and the ratio $\sin(5\pi/8)/\sin(\pi/8)$ turns out to be $1+\sqrt{2}$. In particular, this example shows that the ISI-ratio does not have to be an integer (not even a rational number).

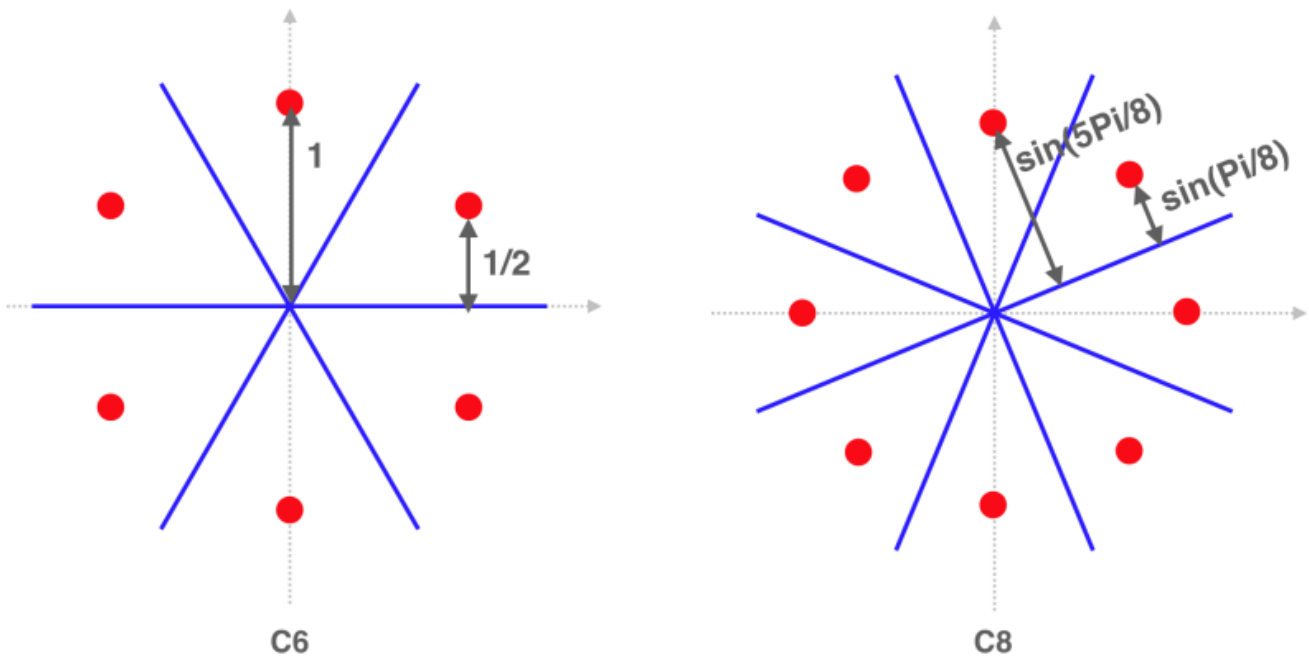


Figure 9: The codes C6 and C8 and their ISI-ratio computation.

Many of the desired properties of a chordal code can be described succinctly using the geometric language.

1. The sum of wire values is zero. This means that the coordinates of the points forming the codewords should sum to zero.
2. The positive flow on the wires is always the same, regardless of the bit transmitted. Under the assumption that the previous condition is valid, this means that the L1-norm of the points should always be the same.
3. The signal is detected using reference-less comparators. This means that the hyperplanes have to pass through the origin.
4. The signals at the input of the comparator slicer have only two complementary values. This means that the ISI-ratio should be 1 for all the comparators.
5. The comparator eliminates common mode noise. This means that the comparators (hyperplanes) should be orthogonal to the vector $(1, 1, \dots, 1)$.
6. The output of the comparators are the transmitted bits. This means that the number of codewords should be 2 to the power of the number of comparators.

Except for the NRZ code above, this section did not constitute a single chordal code satisfying all these constraints. We end this section with one such example which we call the P3-code. The codewords are $\pm[1, 0, -1]$ and $\pm[0, 1, -1]$, so there are four codewords (=2 bits) transmitted on three wires. The pin-efficiency of this code is thus $2/3$. The comparators are 0:1 and $\text{av}(0, 1):2$. The first comparator evaluates to values ± 1 and the second to values ± 1.5 , so both have ISI-ratio 1. A geometric picture of this code as well as a high-level architecture of the system using it is given below.

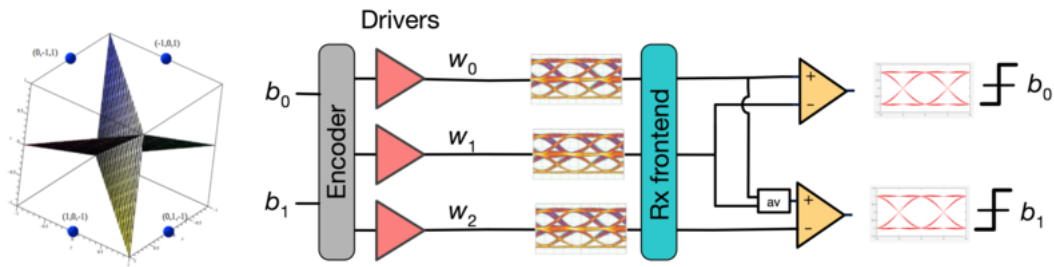


Figure 10: Geometric representation of the P3-code and a high level architecture of the an associated Chord Signaling system. Both comparators see binary values at the input of their associated slicers.

This code is also detectable with the comparators 0:1, and 0:2, which seems to be easier than the comparators given in the figure. Here is a figure of the code and its associated architecture:

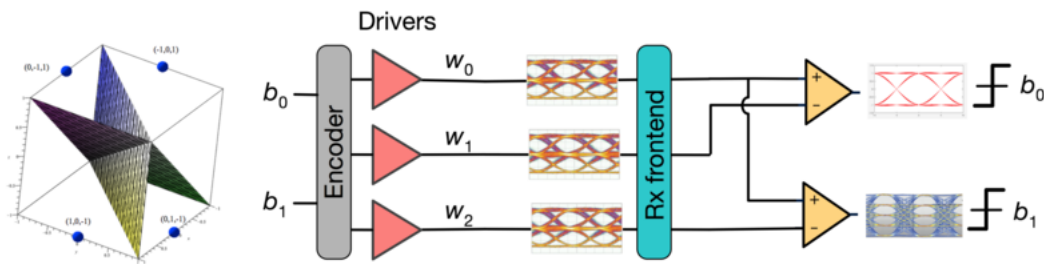


Figure 11: Geometric representation of a code with simpler comparators and a high level architecture of the associated Chord Signaling system. Even though the second comparator is simpler, it does not see binary values at the input of its slicer.

The first comparator in this code is the same as that of the P3 code. The second, however, is different. It has ISI-ratio 2. One can already see from the geometric picture that this code is not as good. The final proof is in the eye diagrams. For an example channel and the same settings for the two signaling systems, we obtain the following eye diagrams using KEYETM.

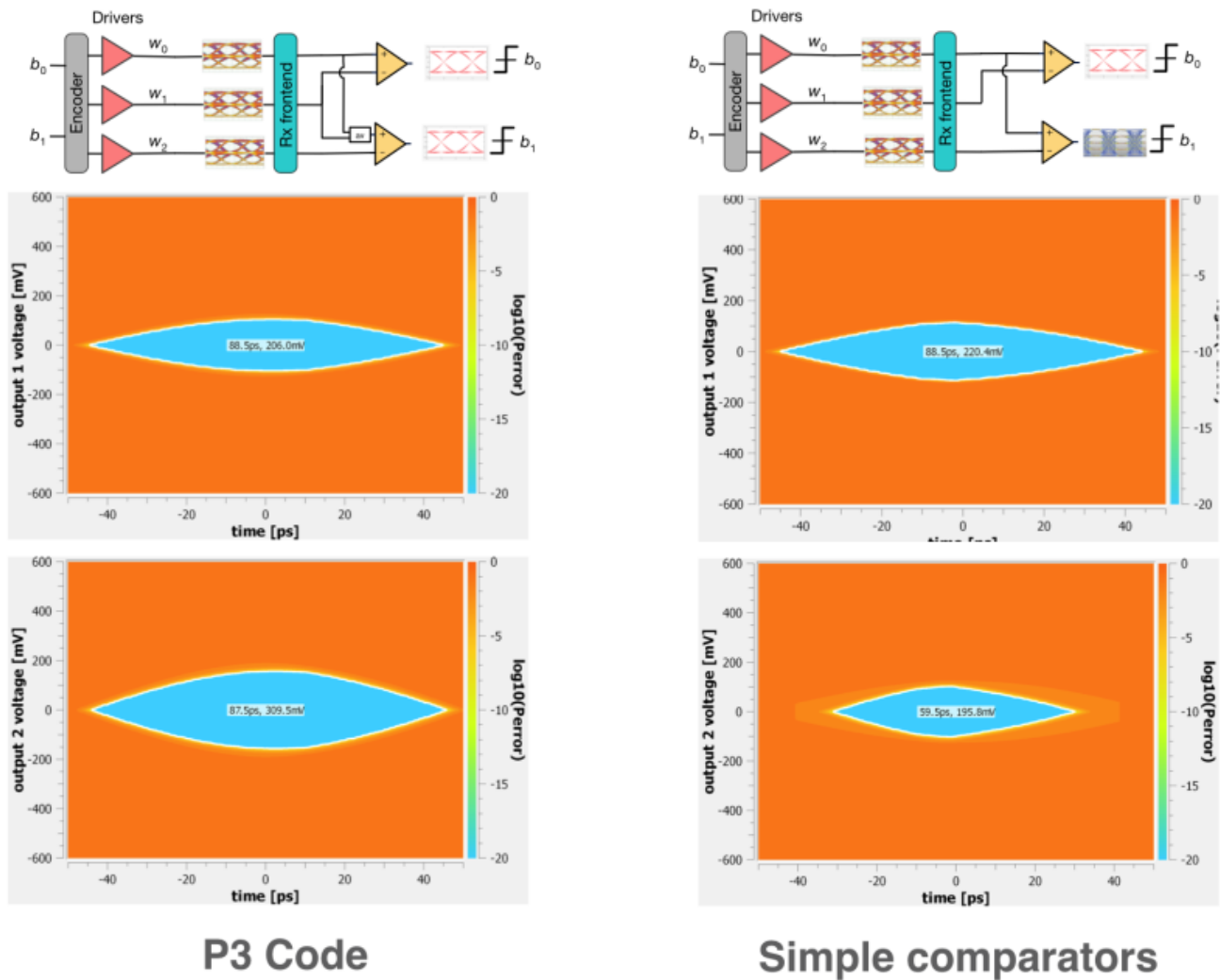


Figure 12: Comparison of the P3 code and the code with a simpler second comparator. As can be seen, this simpler comparator performs worse, because its ISI-ratio is 2 instead of 1.

The scales are the same in all these eye diagrams. Clearly, the comparator with an ISI-ratio of 2 leads to a worse eye than the second comparator of the P3 code. The choice of the comparators for a chordal code therefore matters a lot.

Summary

The simplest answer to this question is: Chord Signaling is a signaling method which uses a chordal code for transmission on detection of signals on multiple wires.

However, this is only part of the story. Chord Signaling doesn't only consist of the signaling method itself, which is a vast area at the intersection of information theory and discrete mathematics, but also of the circuits,

architectures, and an array of practical simulation and diagnostic applications developed by Kandou Bus to construct, analyze, and implement Chord Signaling in practical products. For example, the simulations leading to the eye diagrams for the signaling schemes in the above sections were all done using Kandou's in-house statistical eye diagram tool KEYE™. The time domain counterpart of this tool called TAU™ was used extensively during the design of GW28-125-USR. The code for this product, called CNRZ-5, was designed using a set of proprietary tools and algorithms for the design of chordal codes with given constraints.

Kandou Bus is constantly adding to these tools. The story goes on!