Communication Basics

Serialization, Bit Synchronization, Physical Aspects of Transmission, Transmission Frame, Frame Synchronization, Error Control

Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

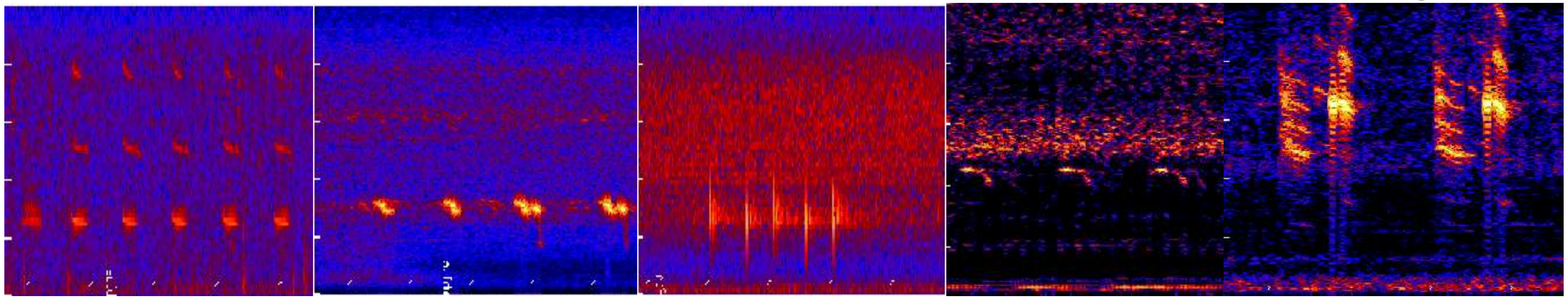
Information

- ***What is* information?**
  - Represented and carried by symbols
  - Recognized by receiver (hopefully)
  - *Interpretation* is the key…

Symbols

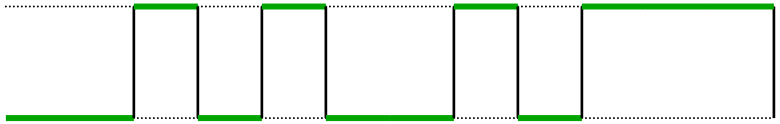
- **Symbols (may) represent information**
  - Voice patterns (Speech)
  - Sign language, Pictograms 🌎 🚹 🍽 🛏
  - Scripture
  - Voltage and current levels
  - Light pulses

Blue Whale Sonograms

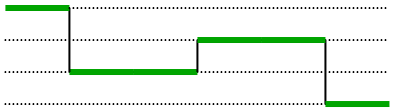


Representation of Symbols for Information Processing, Storage and Exchange

- **In the context of computer systems and data communication**
- **Discrete levels = "Digital"**
  - Resistant against noise
- **How many levels?**
  - Binary (easiest)
    - Bit (binary digit), values 0 and 1
  - M-ary: *More information per time unit!*

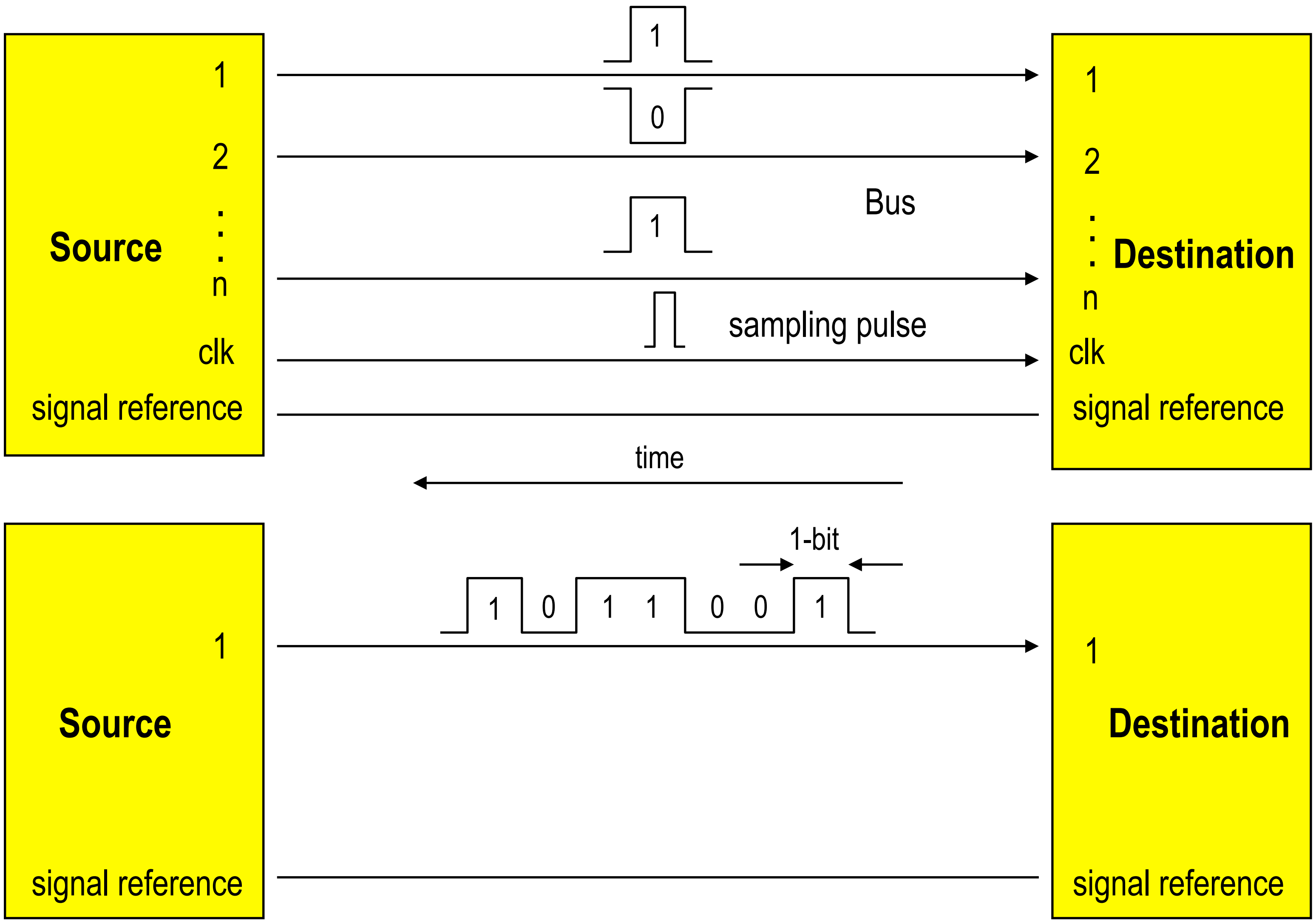


Binary

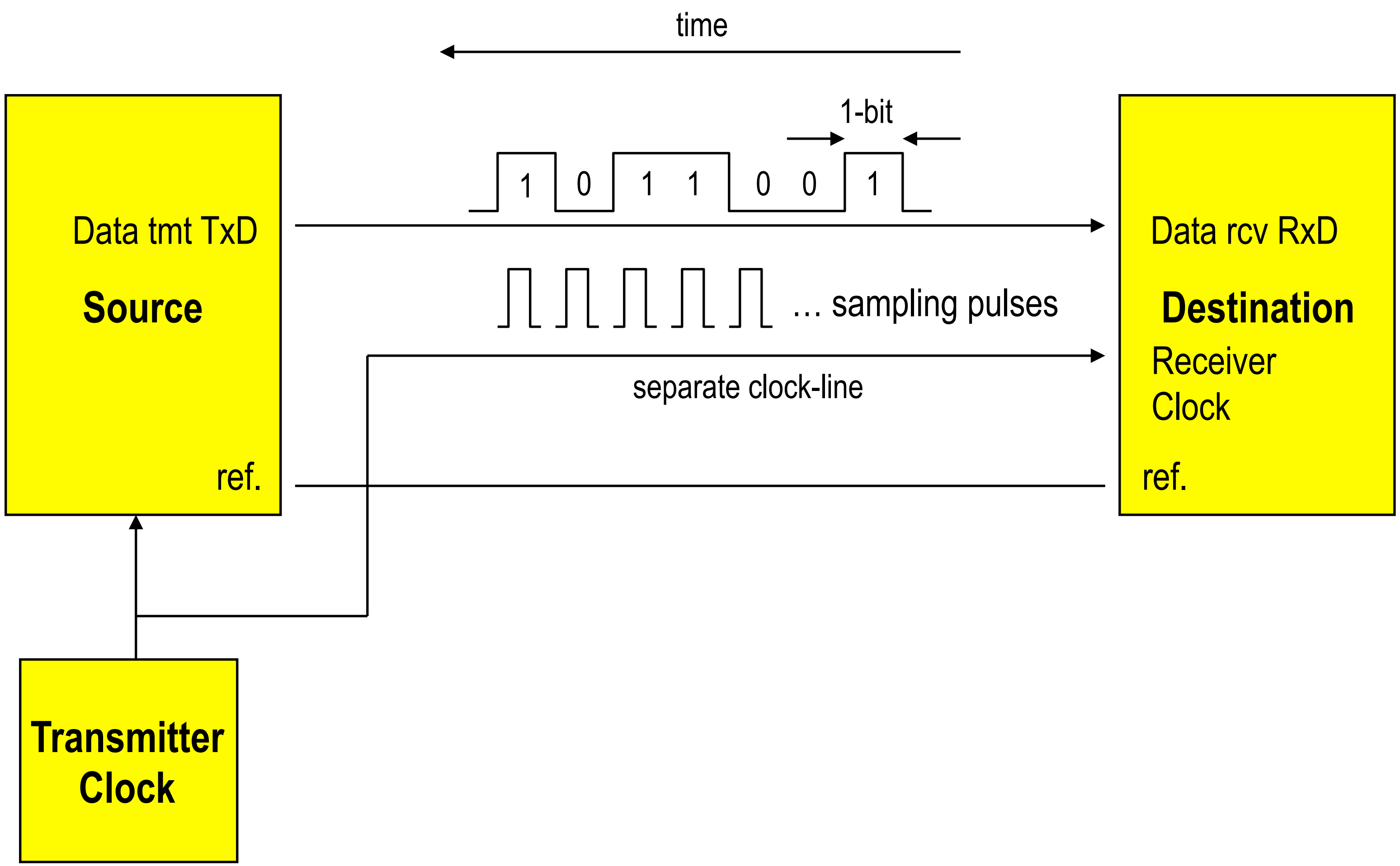


M-ary
(here 4 levels, e. g. ISDN)

Transmission of Information: Parallel versus Serial



Separate Clock-Line ?

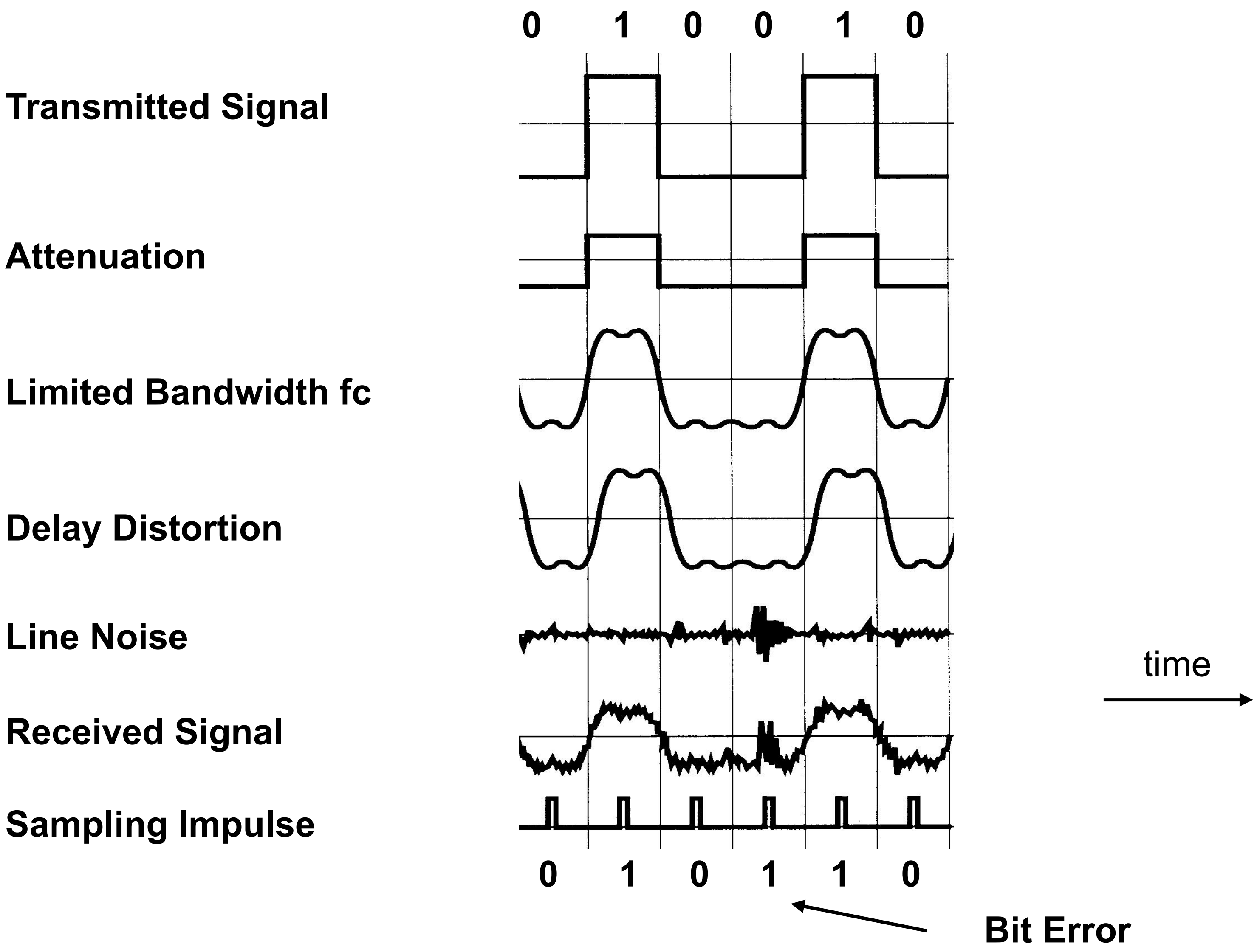


Parallel versus Serial

- **Parallel transmission**
  - Multiple data wires (fast)
  - Explicit clocking wire
  - Simple synchronization but not cost-effective
  - Only useful for small distances

- **Serial transmission**
  - Only one wire (-pair)
  - No clocking wire
  - Most important for data communication for long distances
  - Bit (clock) synchronization is necessary

What Happens To A Signal On The Wire?

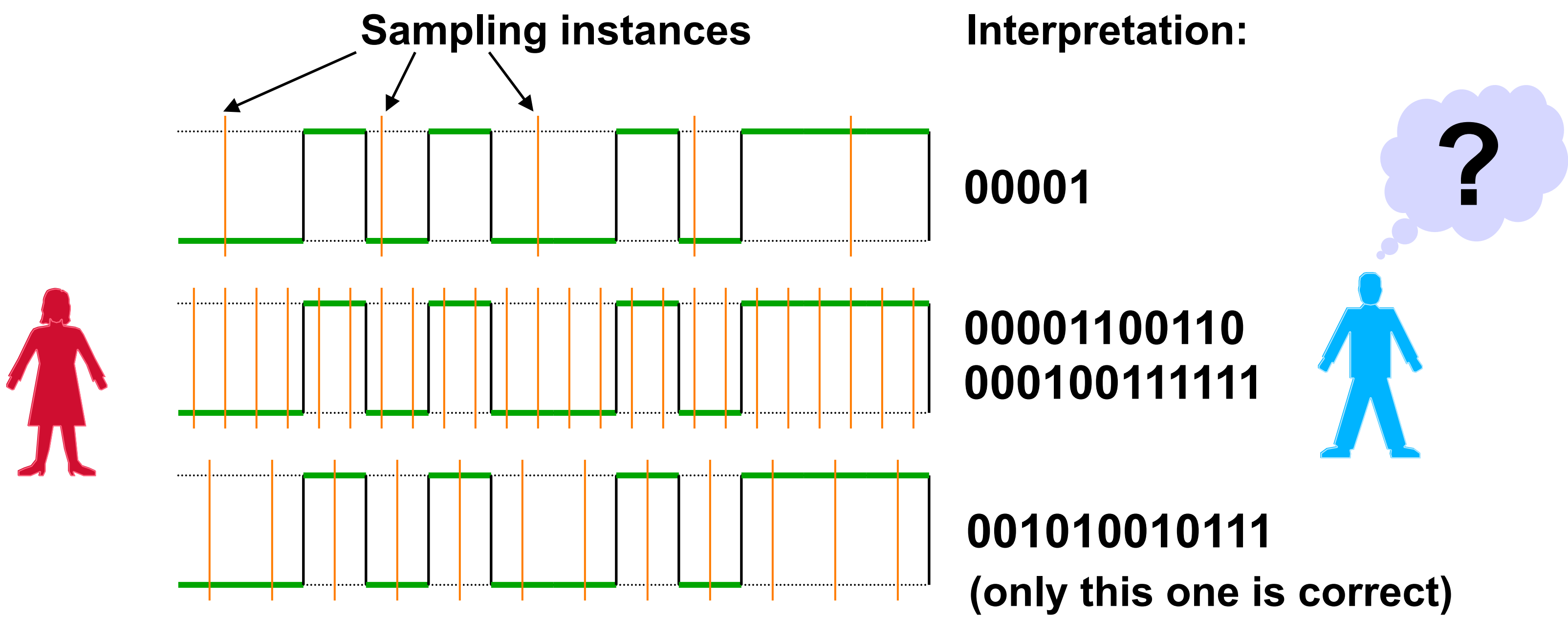


Agenda

- Introduction
- <u>Bit Synchronization</u>
  - Asynchronous
  - Synchronous
- Physical Aspects
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- Transmission Frame
  - Generic Format
  - Frame Synchronization
  - Error Control

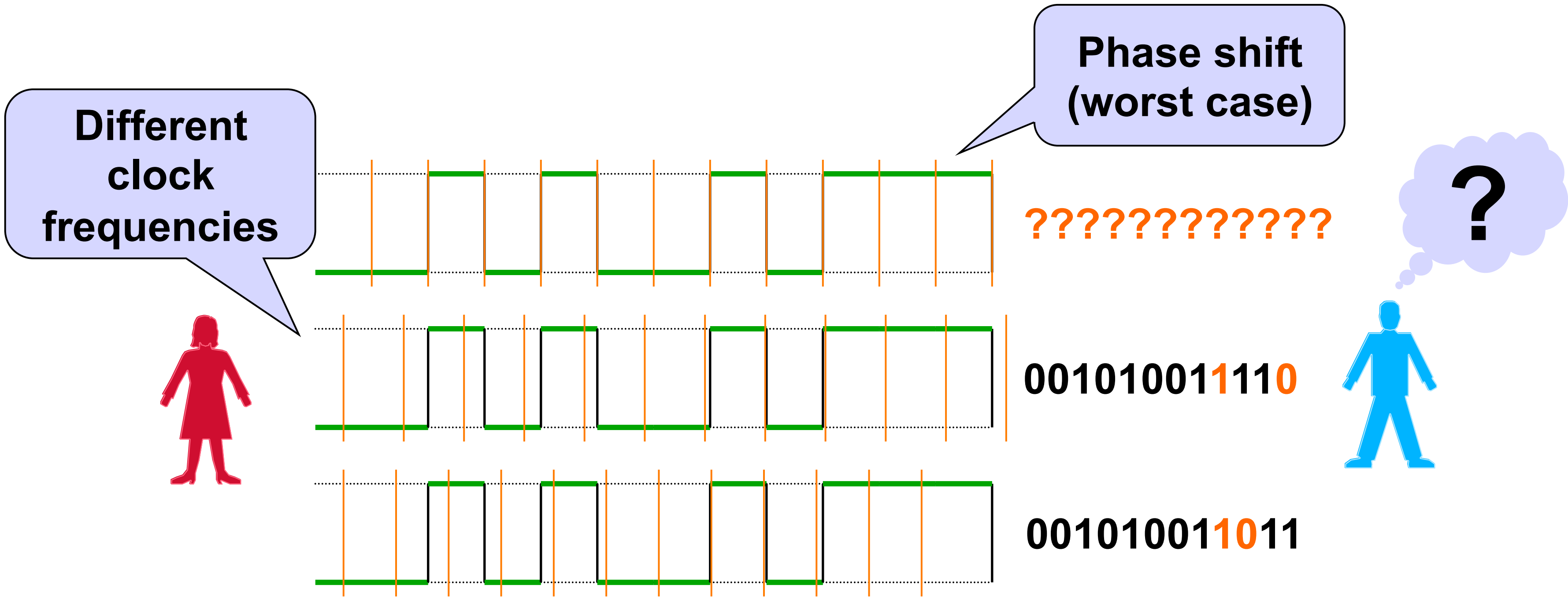
Synchronization

- **Sender sends symbol after symbol...**
- ***When* should receiver pick the signal samples?**
  - => Receiver must sync with sender's clock !

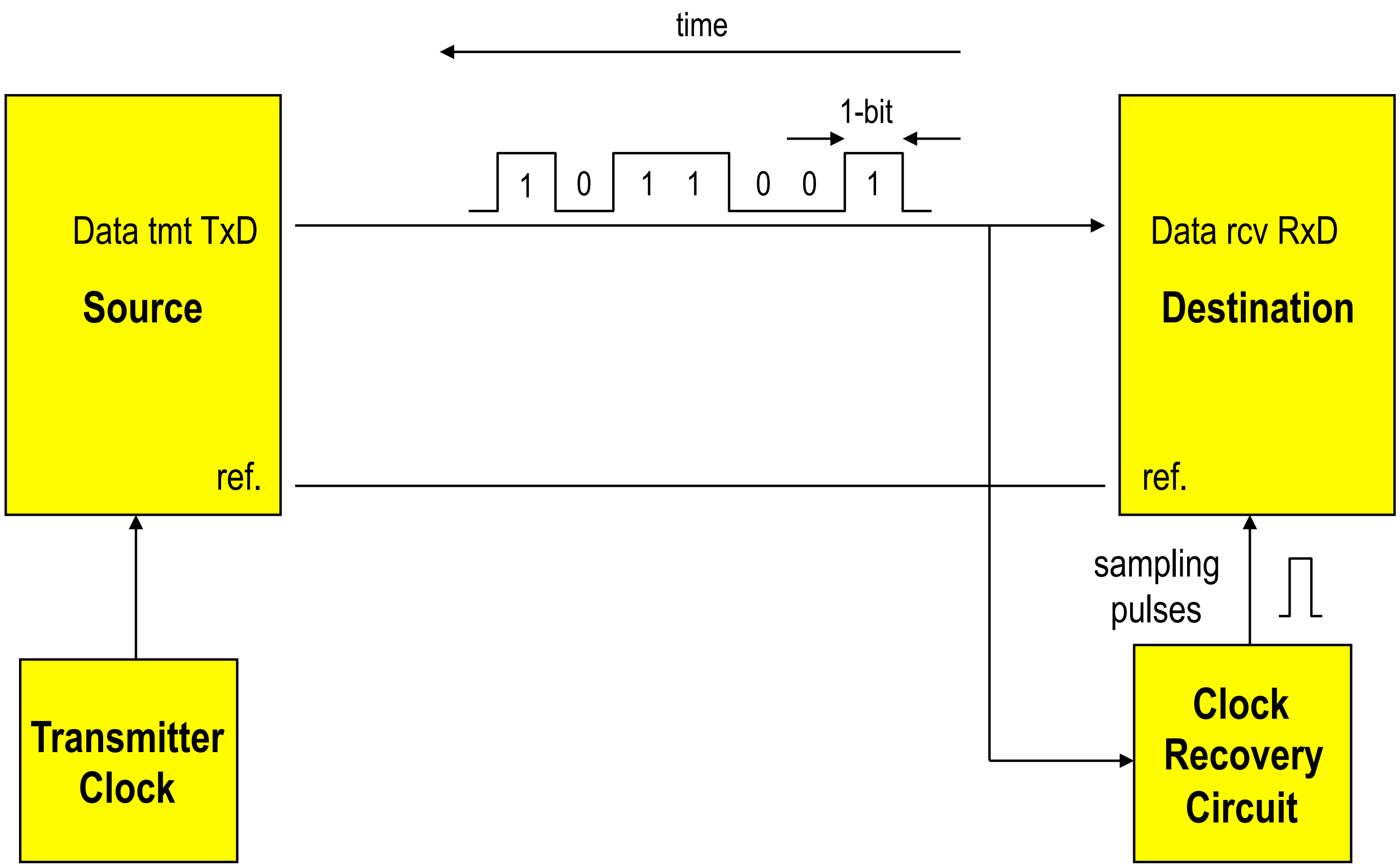


Synchronization

- **In reality, two independent clocks are NEVER precisely synchronous**
  - We always have a frequency shift
  - But we must also care for phase shifts



Bit (Clock) Synchronization Receiver Side

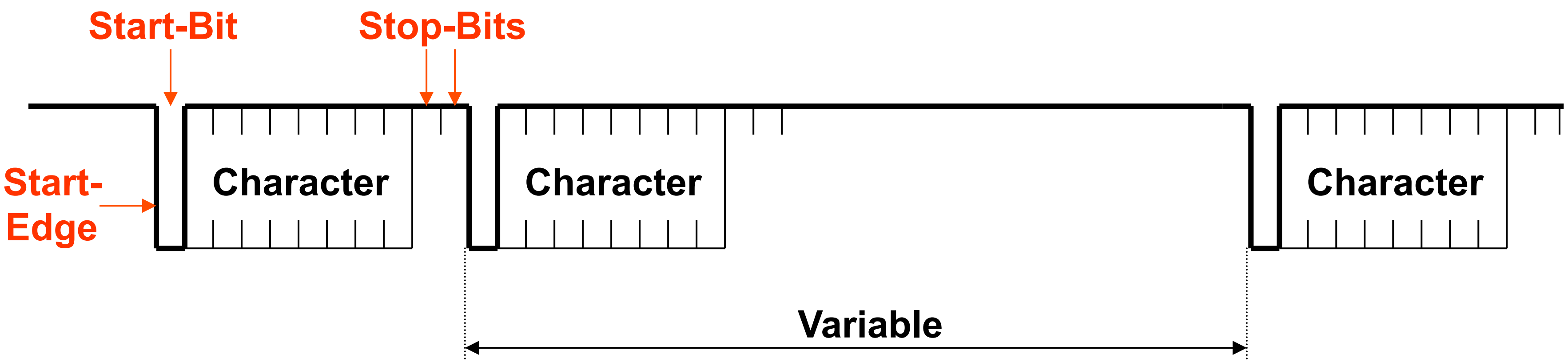


Agenda

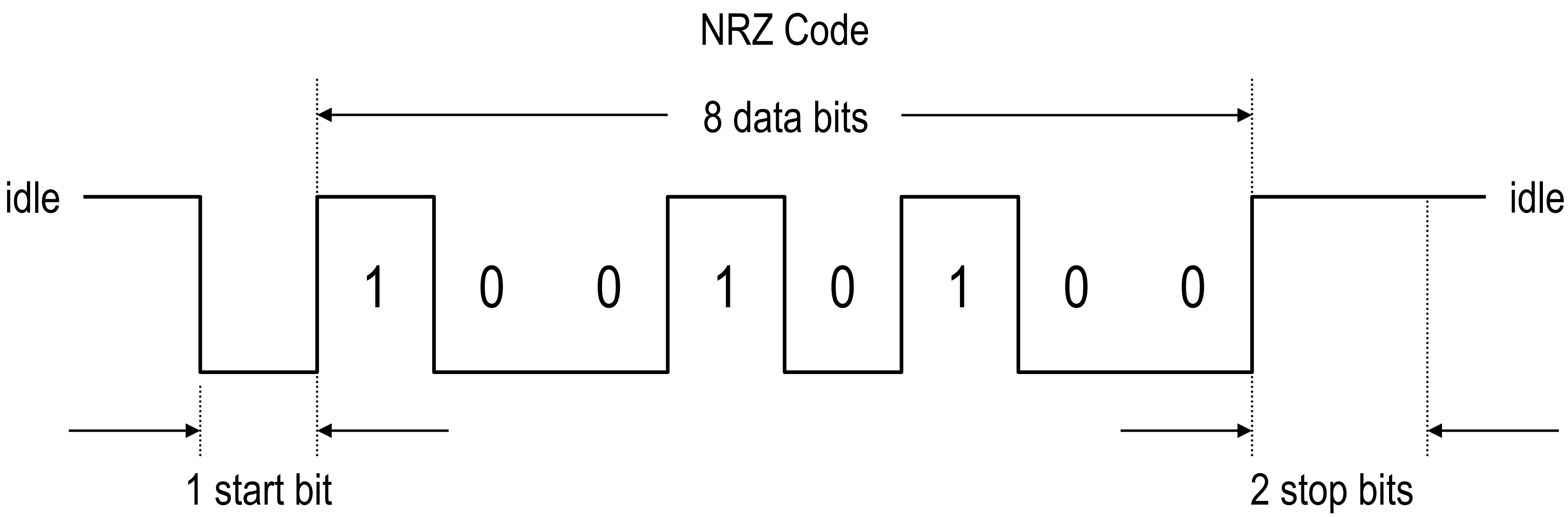
- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

Asynchronous Transmission

- **Independent clocks at transmitter and receiver**
  - Oversampling at the receiver: Much faster than bit rate
- **Only phase is synchronized**
  - Using Start-bits and Stop-bits
  - Variable intervals between characters
  - Synchronicity only during transmission of a data word
- **Inefficient**
  - 8 bits data need additional 3 bits for bit synchronization

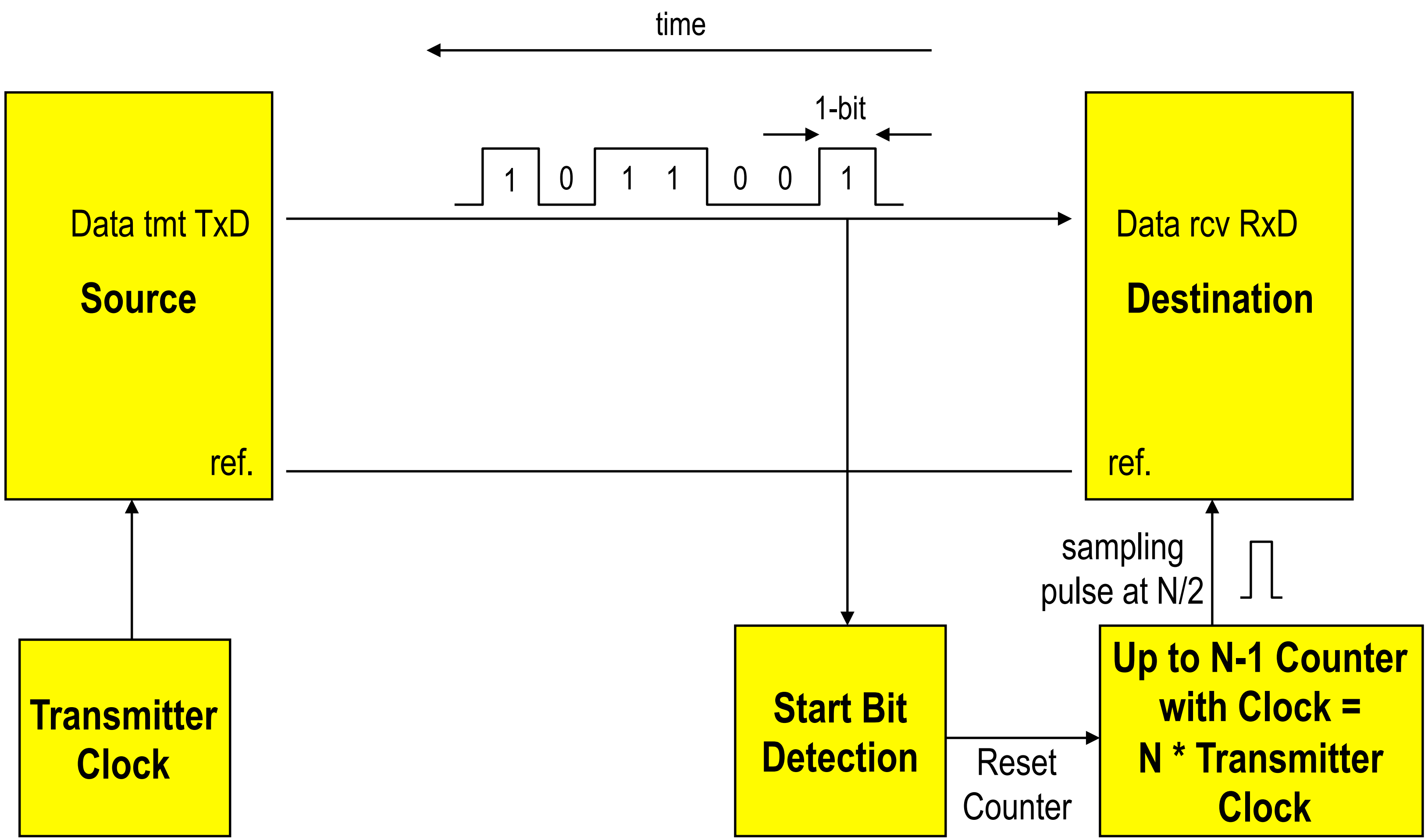


Data Word Framing by Start / Stop Bits



- NRZ (non return to zero) describes the encoding of bits where level 1 refers to logical 1 and level 0 refers to logical 0
- Idle .... no data is transmitted, no change of signal level

Bit Synchronization Circuit Asynchronous



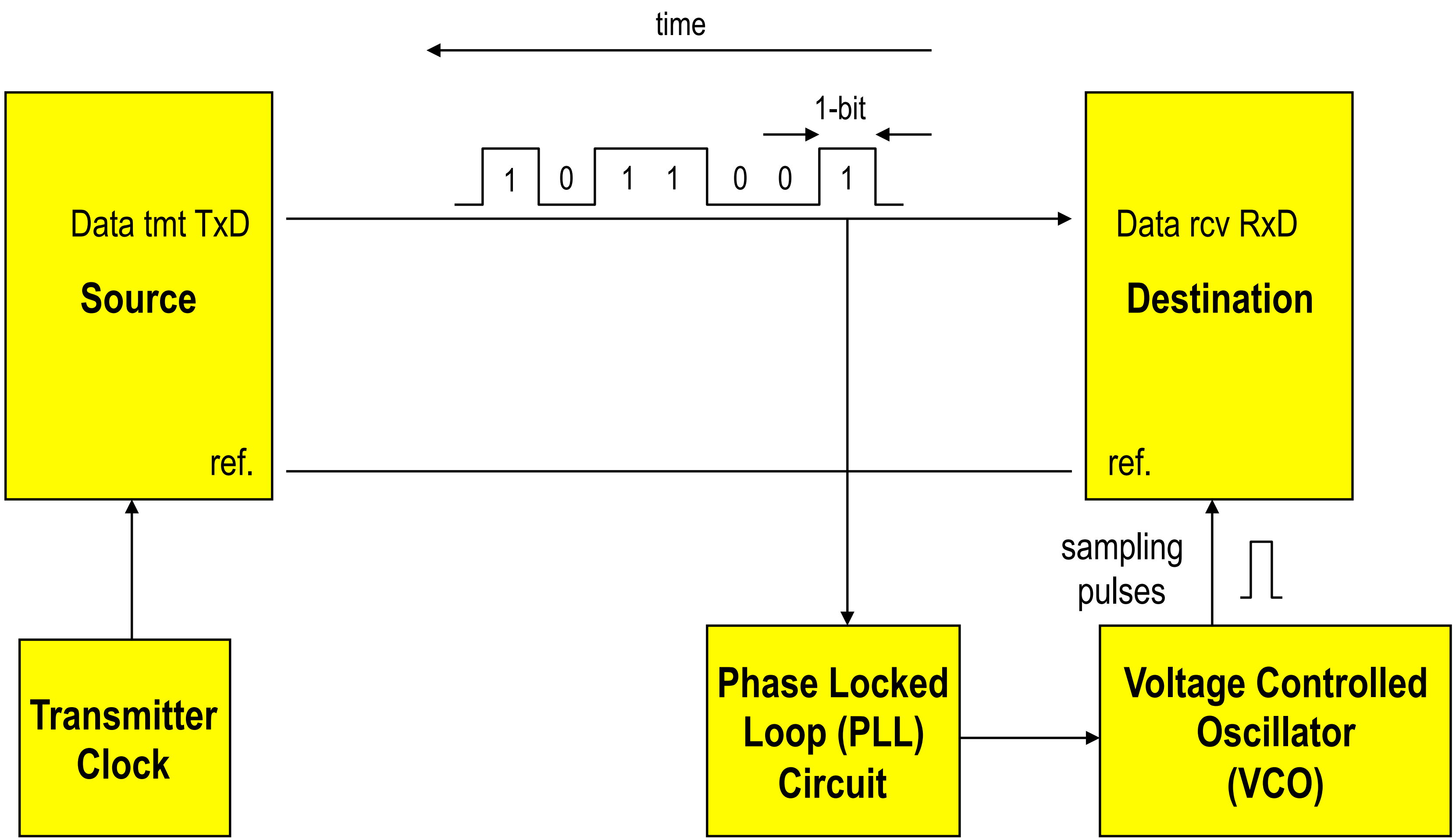
Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

Synchronous Transmission

- **Synchronized clocks**
  - Most important today!
  - Phase and Frequency synchronized
- **Receiver uses a Phased Locked Loop (PLL) control circuit**
  - Requires frequent signal changes
  - => *Coding* or *Scrambling* of data necessary to avoid long sequences without signal changes
    - Encoding / Scrambling at the sender side
    - Decoding / Descrambling at the receiver side
- **Continuous data stream possible**
  - Large frames possible (theoretically endless)
  - Receiver remains synchronized
  - Typically each frame starts with a short "training sequence" aka "preamble" for the PLL to lock in (e. g. 64 bits)

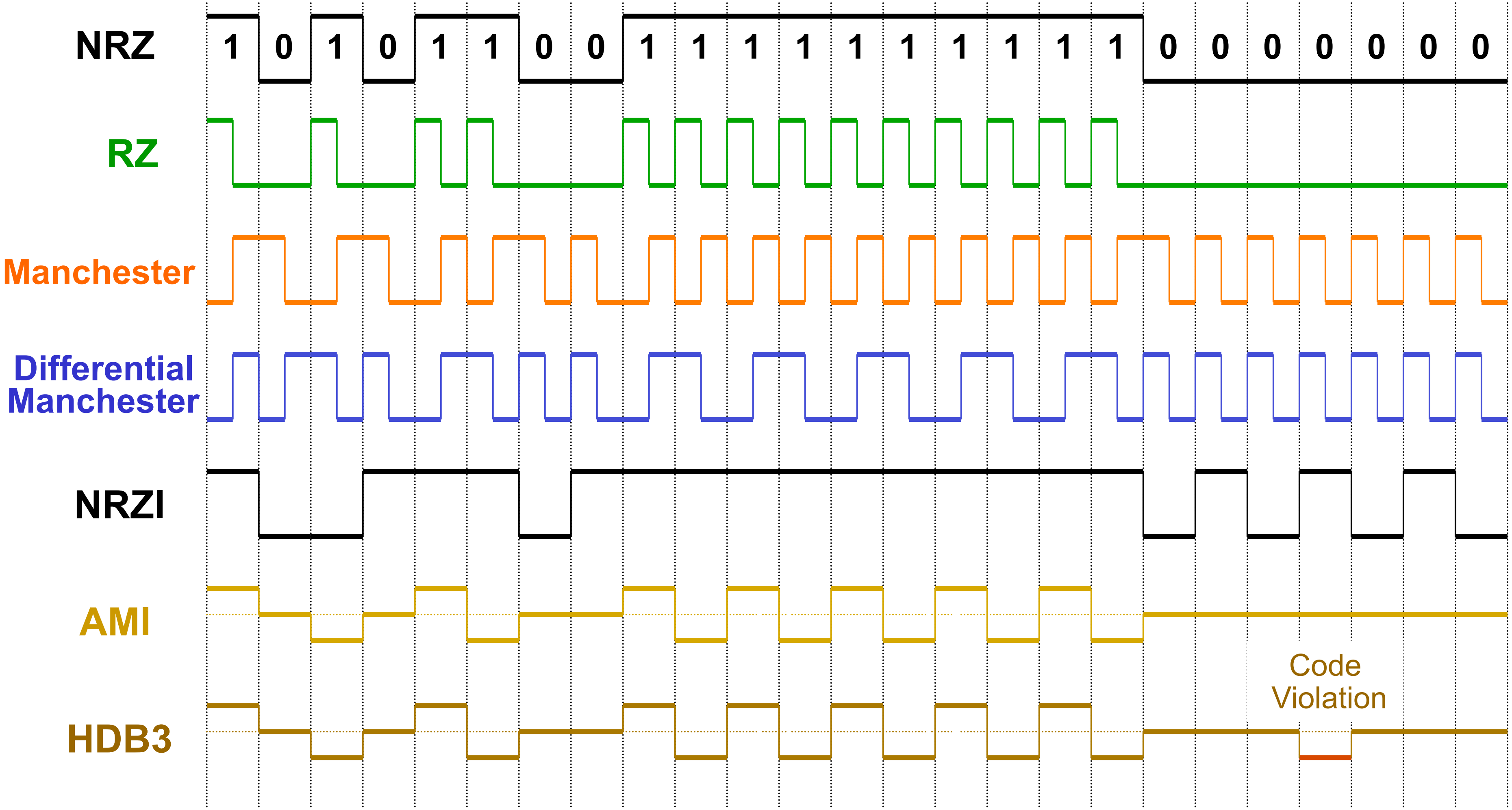
Bit Synchronization Circuit Synchronous



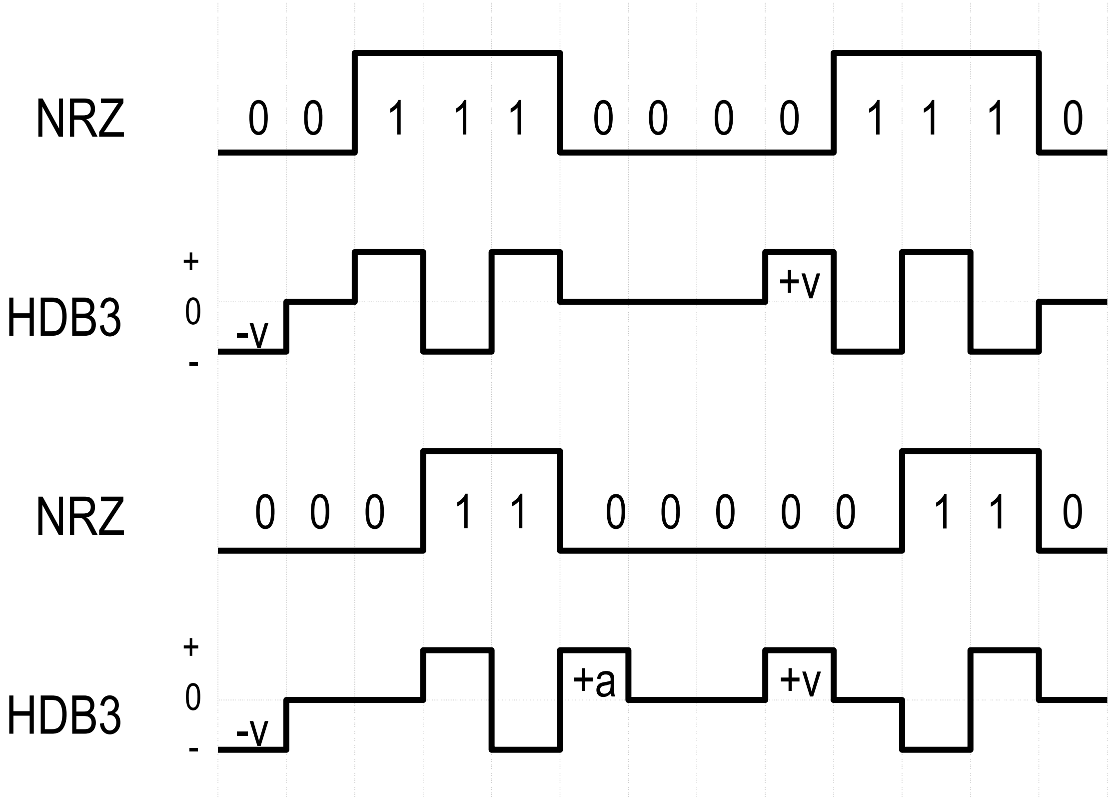
Synchronous Transmission

- **Bit synchronization depends on sufficient signal changes within the bit stream**
  - For long series of 0s or 1s simple NRZ encoding is not able to provide this changes
- **Two basic methods are used to guarantee signal changes**
  - <u>Encoding of bits that every bit contains a signal change</u>
    - Manchester-code (Biphase code), Differential-Manchester-code, commonly used in a LANs
  - <u>Encoding of bits in such a way that there are enough signal changes in the bit stream</u>
    - NRZI (with bitstuffing), RZ and AMI (with scrambler)
    - HDB3 (with code violations), commonly used in a WANs

Line Coding Examples

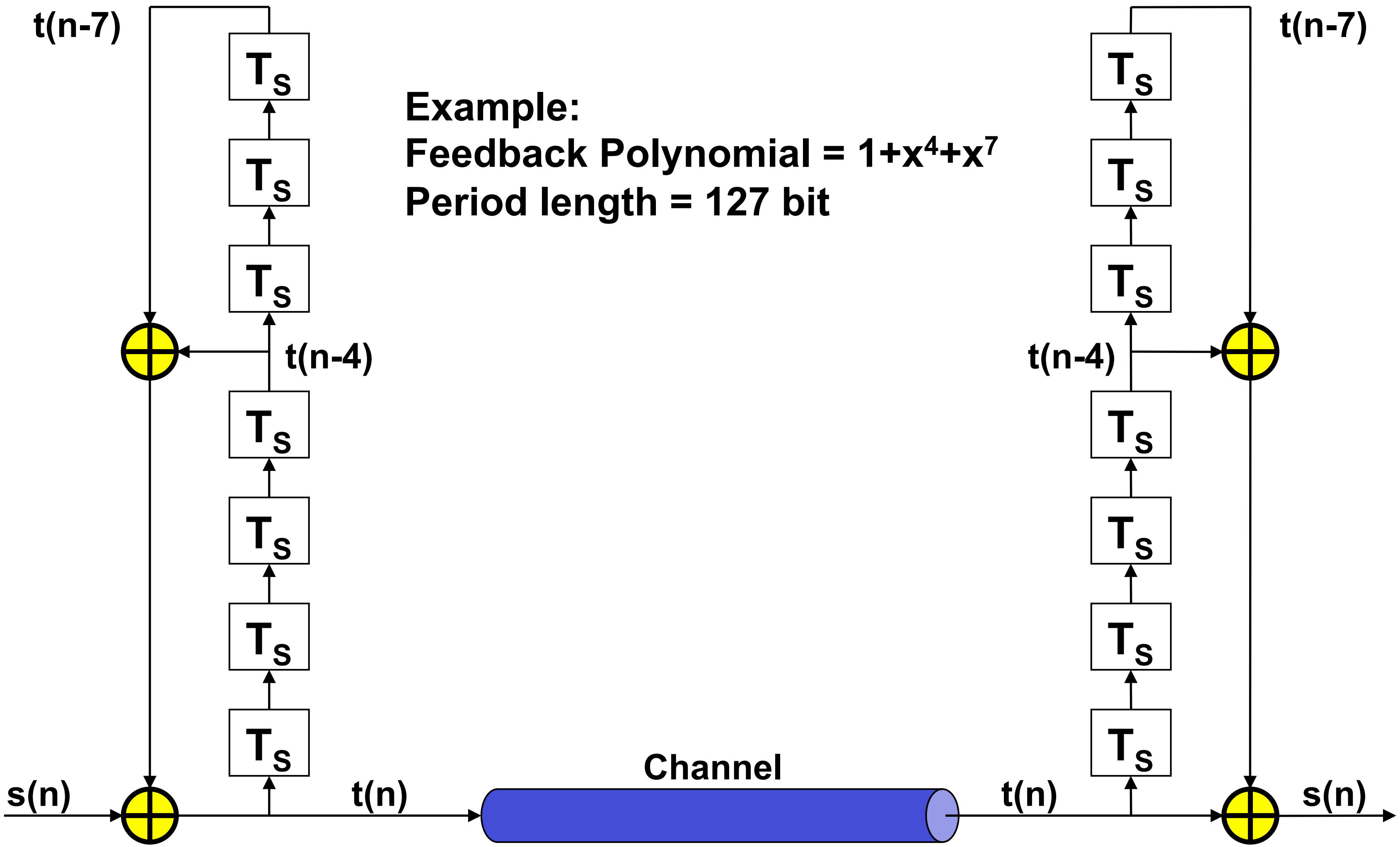


HDB3 (High Density Bipolar 3) Code



	polarity of last pulse	amount of pulses since last violation odd	even
bit pattern	plus minus	0 0 0 +V 0 0 0 -V	-A 0 0 -V +A 0 0 +V

How Does a Scrambler Circuit Look Like?



Example:
Feedback Polynomial = $1+x^4+x^7$
Period length = 127 bit

Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

Theoretical Basis for Data Transmission

- **How can a digital signal be represented?**
  - Fourier analysis proves that any periodic function g(t) with period T can be constructed by summing a (infinite in case of rectangle pulses) number of sinus and cosines functions

$$g(t) = (1/2)c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$

  - With f = 1/T and a_n and b_n as amplitudes of the n^{th} harmonics and c as the dc component
  - Such a decomposition is called Fourier series

Fourier Coefficients

- **How can the values of *c*, a_n and b_n be computed?**

$$c = (2/T)\int_0^T g(t)dt$$

$$a_n = (2/T)\int_0^T g(t)\sin(2\pi nft)dt$$

$$b_n = (2/T)\int_0^T g(t)\cos(2\pi nft)dt$$

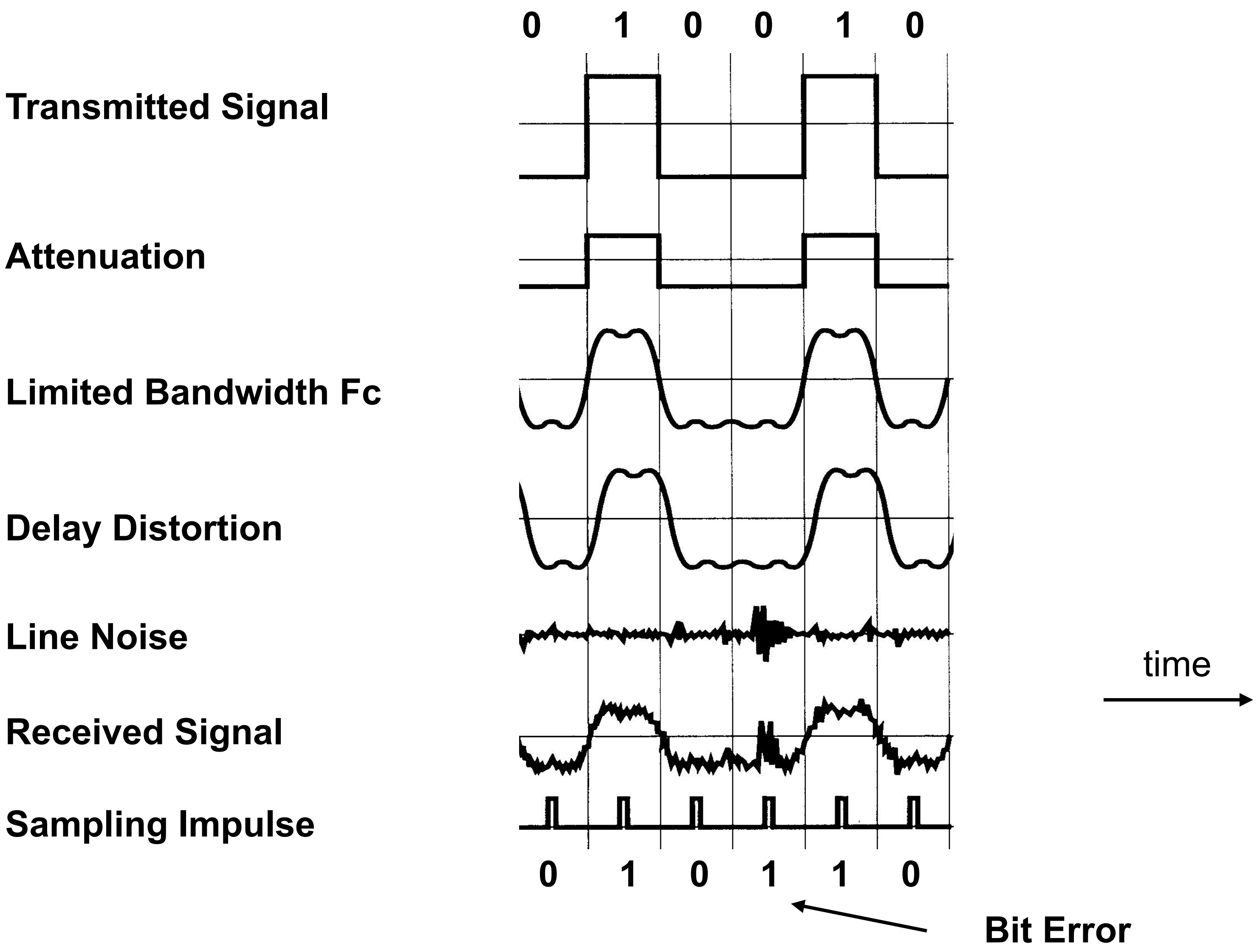
Imperfect Real Data Transmission

1. **No transmission systems can transmit signals without losing some power (attenuation)**

2. **No transmission systems can transmit different Fourier components with the same speed (delay distortion)**

3. **No transmission systems is free from noise**

That Happens To A Signal !!!



Real Data Transmission

- **In real transmission systems**
  - The original signal will be attenuated, distorted and influenced by noise when traversing the transmission line
- **By increasing the bit rate**
  - Bit synchronization even in middle of a bit becomes more and more difficult because of these impairments
  - Above a certain rate bit synchronization will be impossible
- **Relationship**
  - Between bandwidth Fc, line length and maximum achievable bit rate on a certain transmission line (system)

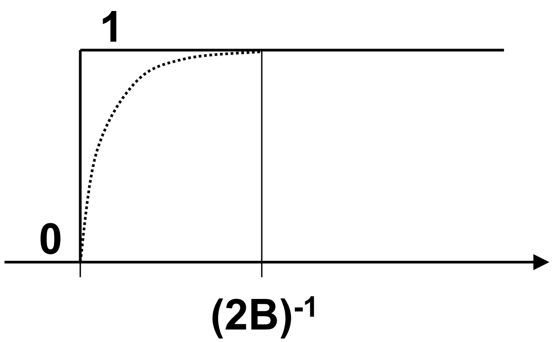
Maximal Information Rate (Theoretical)

- **What is the maximal information rate of an ideal (noiseless) but bandwidth limited transmission channel ?**
  - Nyquist law: $R = 2 * B * \log_2 V$
    - valid for a noiseless channel
    - R ... maximum bit rate (bits/sec)
    - B ... bandwidth range of a bandwidth limited transmission
    - V ... number of signal levels (e.g. 2 for binary transmission)
  - example analogue telephone line
    - B = 3000 Hz (range 400 – 3400 Hz)
    - R = 6000 bits/sec for V = 2
    - R = 18000 bits/sec for V = 8

Nyquist Law Rationale

- **Maximal data rate proportional to channel-bandwidth B**
  - Raise time of Heavyside T=1/(2B)
  - So the maximum rate is R=2B, also called the Nyquist Rate
  - Note: We assume an ideal channel here – without noise!

- **Bandwidth decreases with cable length**
  - As a dirty rule of thumb: BW × Length ≅ const
  - But note that the reality is much more complex
  - Solitons are remarkable exceptions…



Maximum signal rate: At least the amplitude must be reached

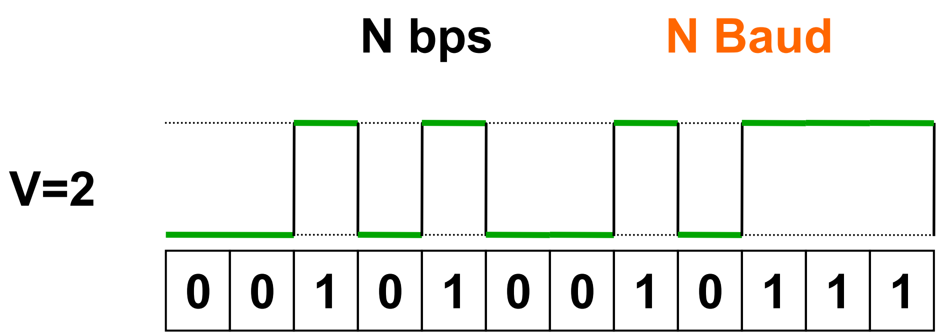
Bitrate versus Baud

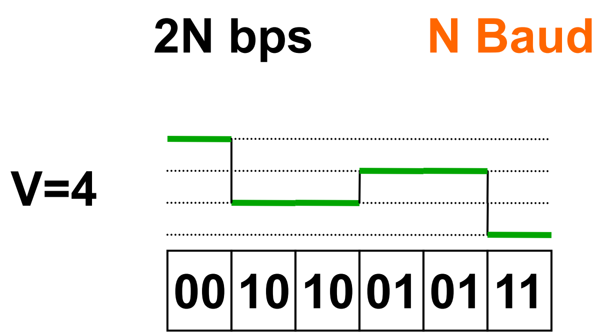
- **The rate of changes of a symbol**
  - is called signaling rate $\mathbf{R_s}$ or Symbol Rate
  - is measured in Baud
- **The rate of bits transported**
  - is called bit rate $\mathbf{R_i}$ or Information Rate
  - and is measured in bit/sec (bps)

- $\mathbf{R_i = R_s * \log_2 V}$
  - V … number of signal levels

- $\mathbf{R_i = R_s}$
  - for binary transmission where V = 2

- **The goal is to send many (=as much as possible) bits per symbol**
  - => QAM (see next slides)





Maximal Information Rate (Reality)

- **What about a real channel? What is the maximum achievable information rate in presence of noise?**
  - Disturbance caused by crosstalk, impulse noise, thermal or white noise

- **Answer by C. E. Shannon in 1948**
  - Even when noise is present, information can be transmitted *without errors* when the information rate is below the *channel capacity C*
  - Channel capacity depends only on channel bandwidth and SNR (signal to noise ratio)
  - max R = C = B * $\log_2$ (1+S/N)
    - S ... signal power, N ... noise power
    - SNR ... measured in decibel (db)
    - SNR = 10 * $\log_{10}$ S/N
  - example analogue telephone line
    - B = 3000 Hz
    - SNR = 30 db means 30 = 10 * $\log_{10}$ (S/N) -> S/N = 1000
    - max R = 3000 * $\log_2$ (1+1000) = 3000 * (9,967226259)
    - max R = approximately 29902 bits/sec

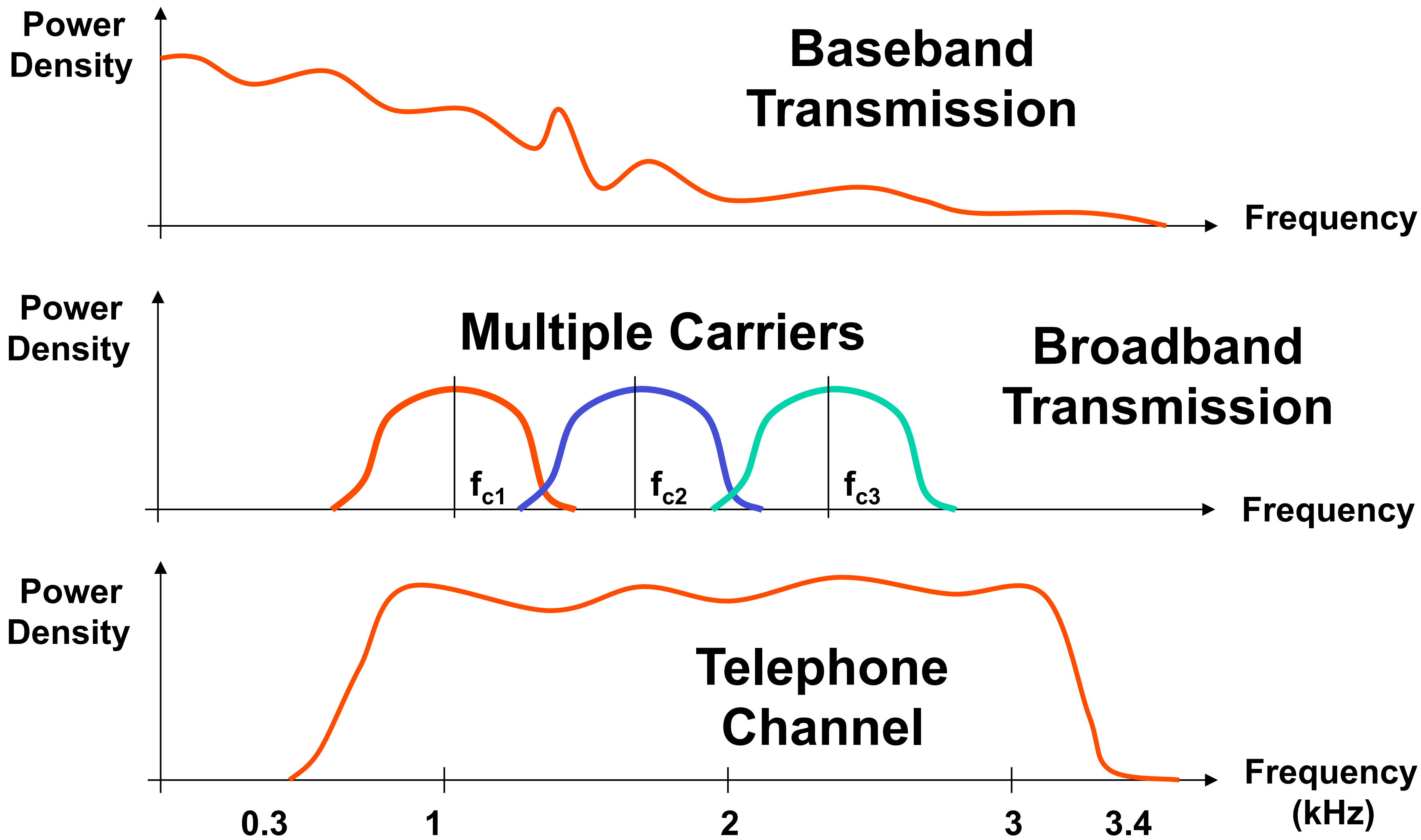
Agenda

- **Introduction**
- **Bit Synchronization**
  – Asynchronous
  – Synchronous
- **Physical Aspects**
  – Mathematical Background
  – Communication Channel / Modulation
  – Serialization / Propagation Delay
- **Transmission Frame**
  – Generic Format
  – Frame Synchronization
  – Error Control

Communication Channels

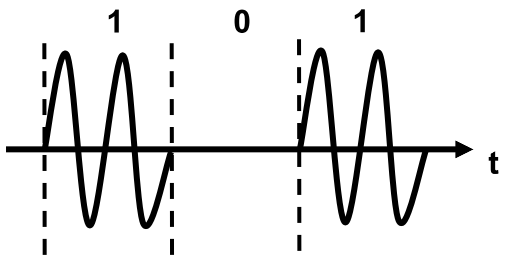
- **Usually *Low-Pass* behavior**
  - Higher frequencies are more attenuated than lower
- **Baseband transmission**
  - Signal without a dedicated carrier
  - Example: LAN technologies (Ethernet etc)
- **Carrierband / Narrowband transmission**
  - The baseband signal modulates a carrier to match special channel properties
- **Broadband transmission**
  - Different baseband signals modulate different carriers
  - Medium can be shared for many users / channels e. g. WLAN and cable networks

Channel Utilization Examples

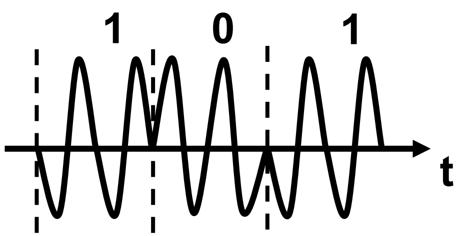


Analogue Modulation Overview

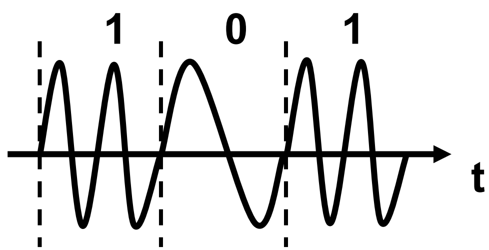
- **EVERY transmission is analogue – but there are different methods to put a base-band signal onto a high-frequency carrier**
- **The most simple (and oldest) is ASK**
  – The illustrated ASK method is simple "On-Off-Keying" (OOK)
- **FSK and PSK are called "angle-modulation" methods (nonlinear => spectrum shape is changed!)**
- **For digital transmission, almost always QAM is used**
  – The BER of BPSK is 3 dB better than for simple OOK



Amplitude Shift Keying (ASK)



Phase Shift Keying (PSK)



Frequency Shift Keying (FSK)

$$g(t) = A_t \cdot \cos(2\pi f_t t + \varphi_t)$$

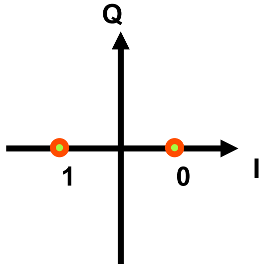
These three parameters can be modulated

QAM: Idea

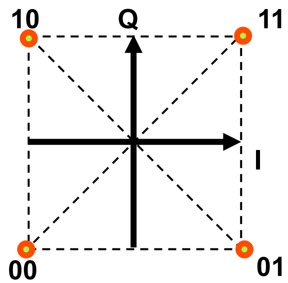
- **"Quadrature Amplitude Modulation"**
- **Idea:**
  1. Separate bits in groups of words (e. g. of 6 bits in case of QAM-64)
  2. Assign a dedicated pair of Amplitude and phase to each word (A,φ)
  3. Create the complex amplitude $Ae^{j\varphi}$
  4. Create the signal $\mathrm{Re}\{Ae^{j\varphi}\, e^{j\omega t}\}$
     $= A\,(\cos\varphi \cos\omega t - \sin\varphi \sin\omega t)$ which represents one (of the 64) QAM symbols
  5. Receiver can reconstruct (A,φ)

QAM: Symbol Diagrams

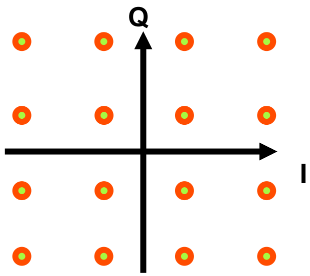
Standard PSK



Quadrature PSK (QPSK)

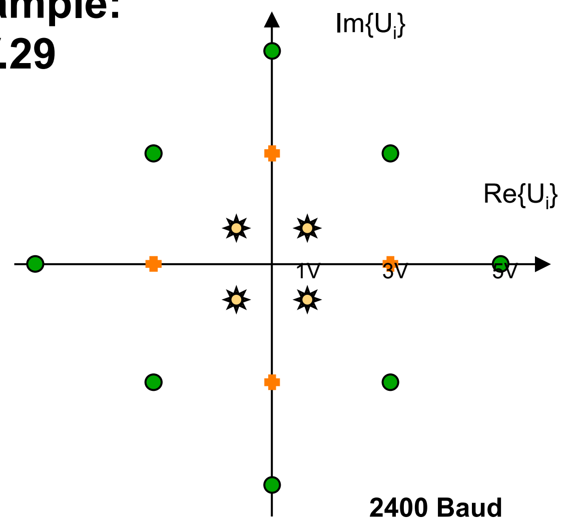


16-QAM



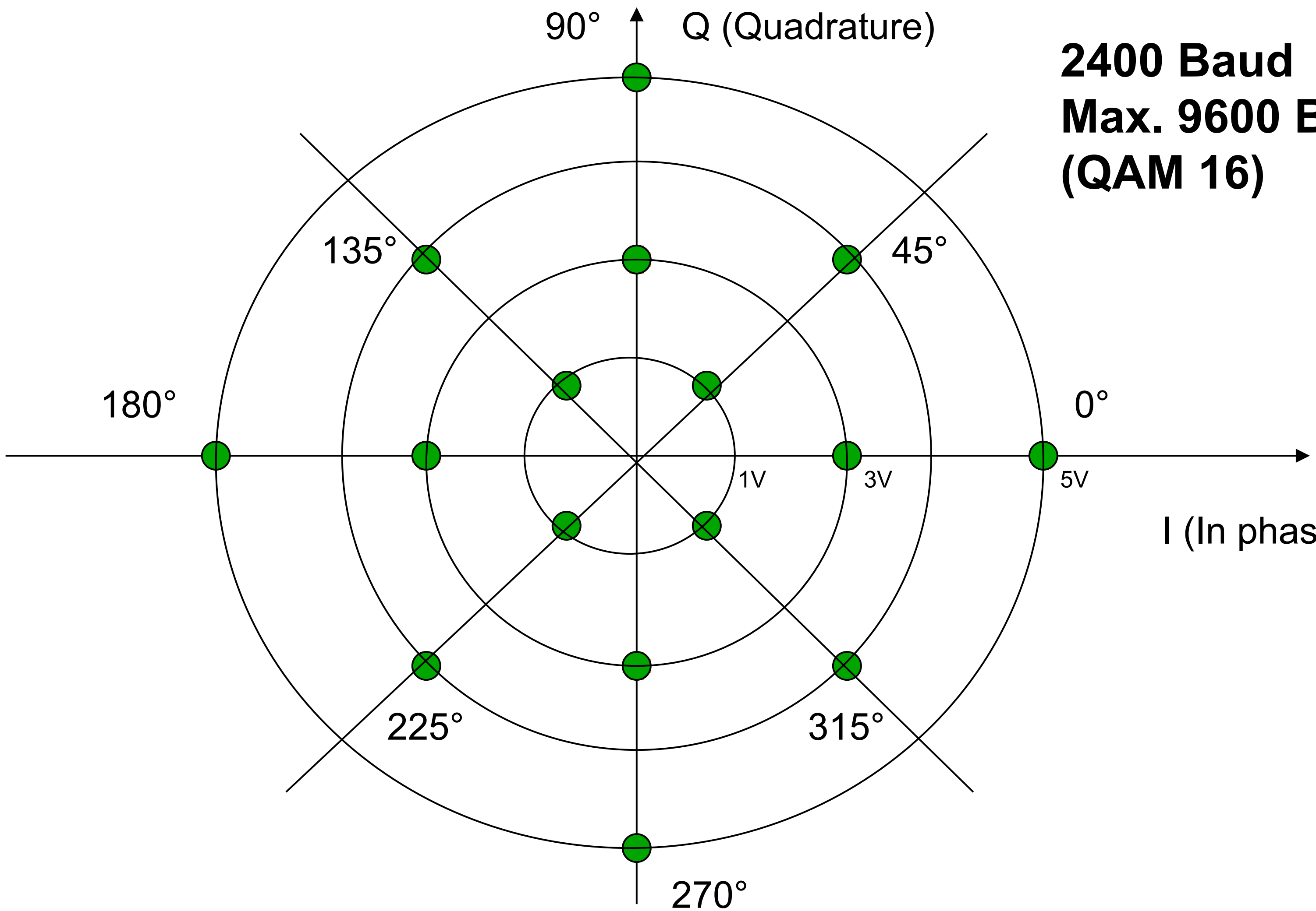
Other example: Modem V.29

- For noisy and distorted channels 4800 bit/s
- For better channels 7200 bit/s
- For even better channels 9600 bit/s



2400 Baud
Max. 9600 Bit/s

Modem: V.29 (QAM) for TELCO Lines

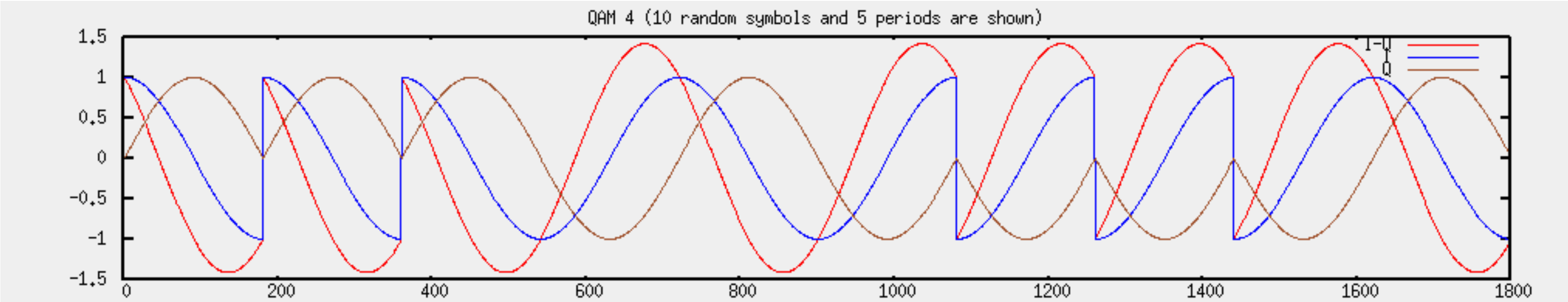


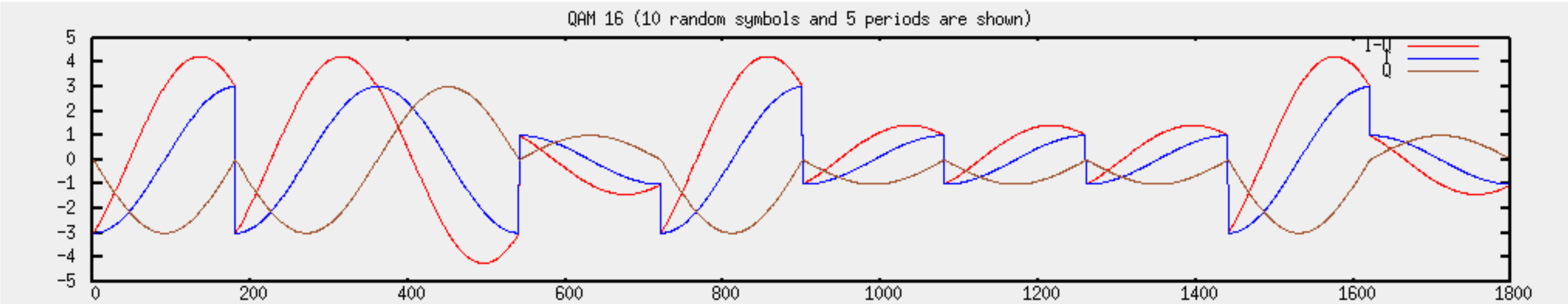
2400 Baud
Max. 9600 Bit/s
(QAM 16)

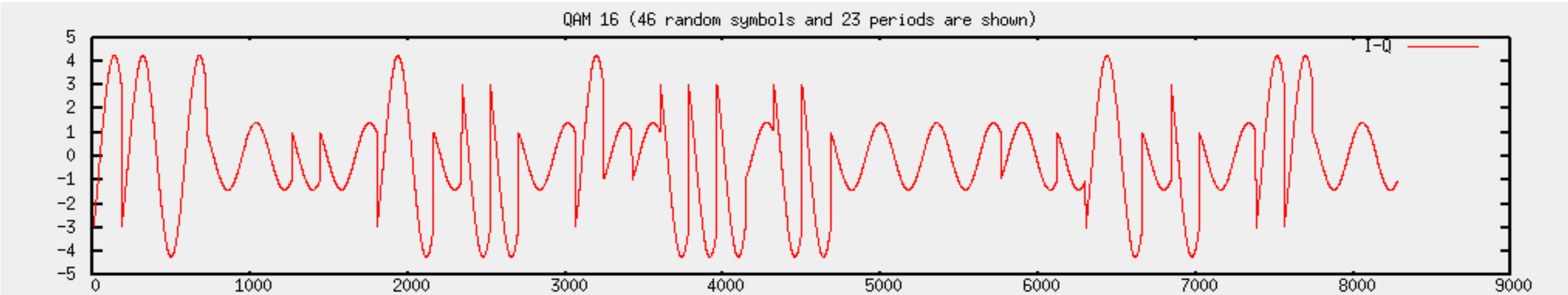
Example QAM Applications

- **One symbol represents a bit pattern**
  – Given N symbols, each represent ld(N) bits
- **Modems (Telco – 200-3500Hz limited),**
- **1000BaseT (Gigabit Ethernet)**
- **WiMAX, GSM, …**
- **WLAN 802.11a and 802.11g:**
  – BPSK @ 6 and 9 Mbps
  – QPSK @ 12 and 18 Mbps
  – 16-QAM @ 24 and 36 Mbps
  – 64-QAM @ 48 and 54 Mbps

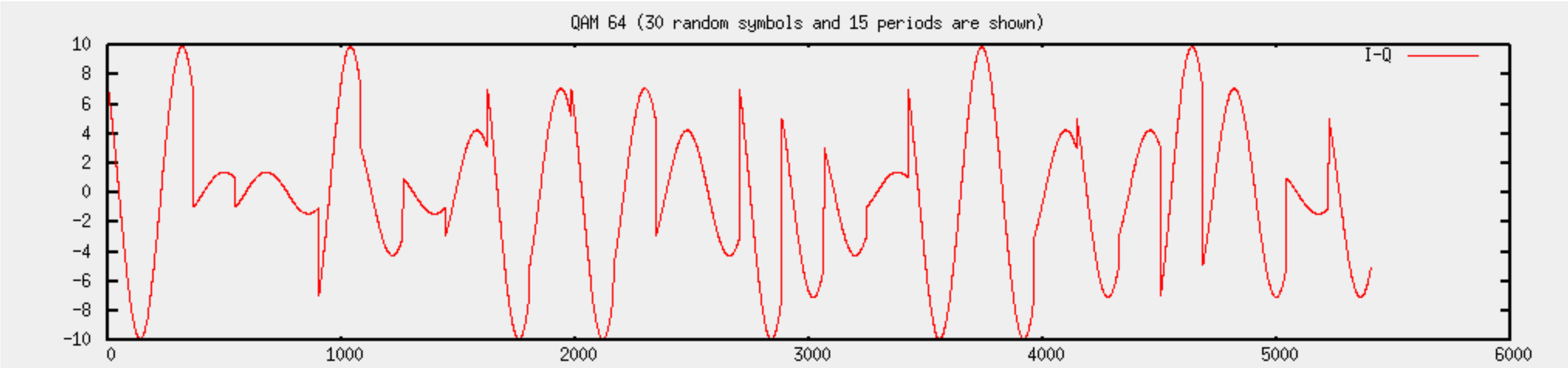
QAM Example Symbols (1)

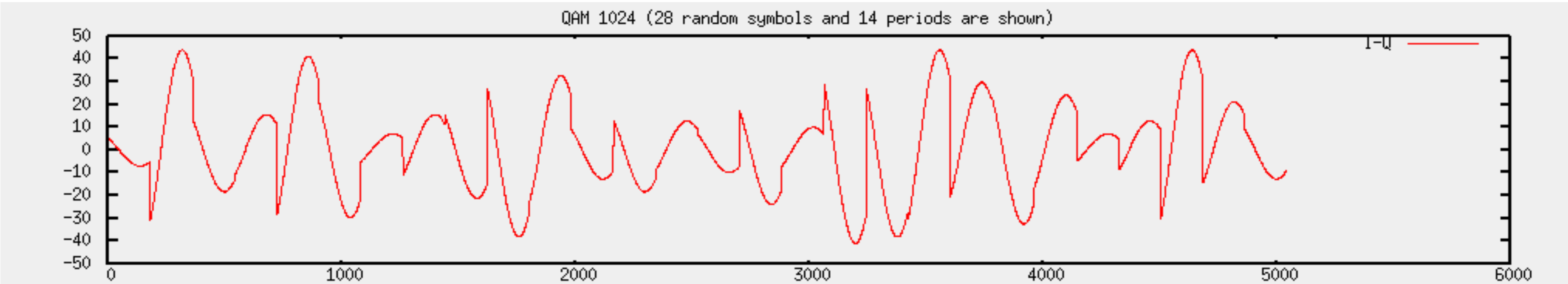


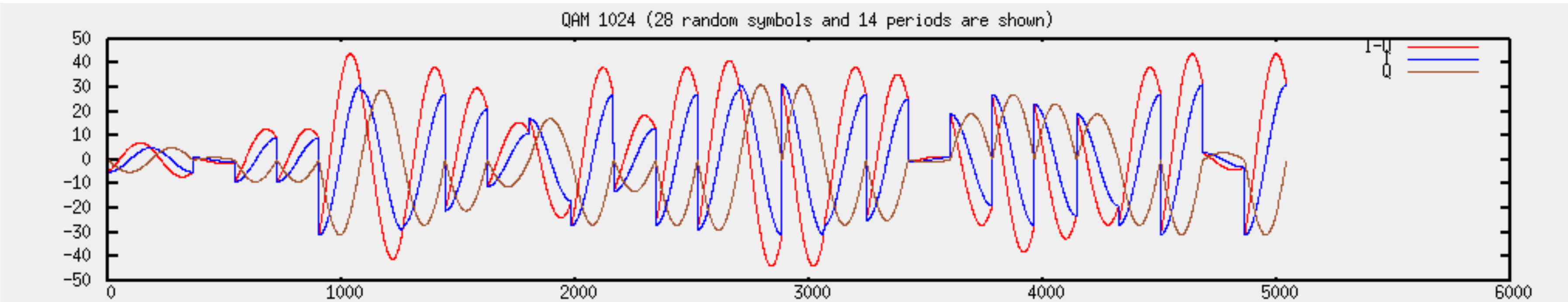




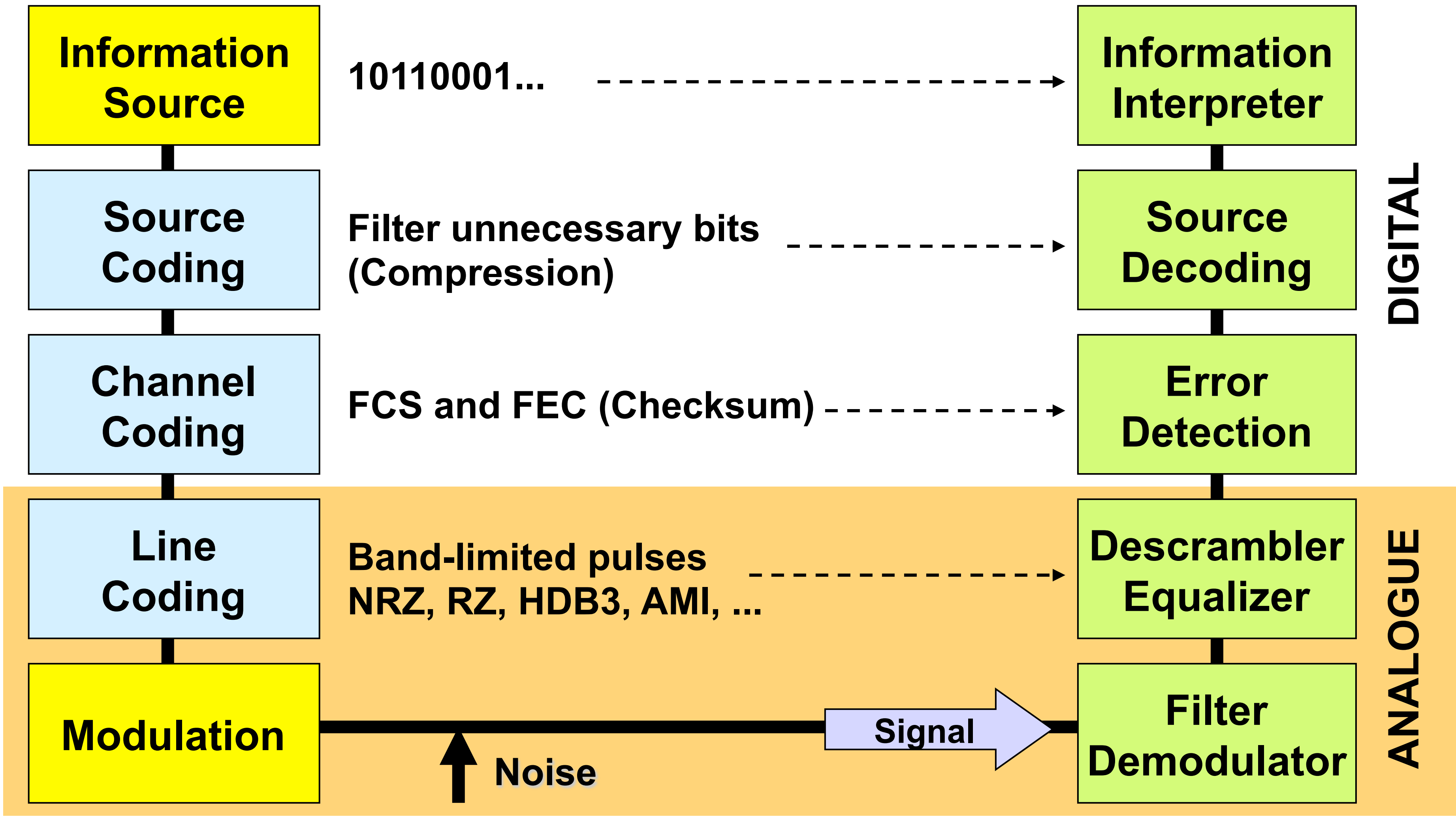
QAM Example Symbols (2)







Transmission System Summary



Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

Time to Transmit A Given Number Of Bytes

<u>Serialization Delay</u> (in ms) = [(Number of Bytes * 8) / (Bitrate in sec)] * 1000

	Bitrate		9,6 kbit/s	48 kbit/s	128 kbit/s	2,048 Mbit/s	10 Mbit/s	100 Mbit/s	155 Mbit/s	622 Mbit/s	1 Gigabit/s
	Number of Byte		Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})	Delay in msec (10^{-3})
Bit	0,125		0,104167	0,020833	0,007813	0,000488	0,000100	0,000010	0,000006	0,000002	0,000001
Byte	1		0,833333	0,166667	0,062500	0,003906	0,000800	0,000080	0,000052	0,000013	0,000008
PCM-30	32		26,666667	5,333333	2,000000	0,125000	0,025600	0,002560	0,001652	0,000412	0,000256
ATM cell	53		44,166667	8,833333	3,312500	0,207031	0,042400	0,004240	0,002735	0,000682	0,000424
Ethernet	64		53,333333	10,666667	4,000000	0,250000	0,051200	0,005120	0,003303	0,000823	0,000512
X.25	256		213,333333	42,666667	16,000000	1,000000	0,204800	0,020480	0,013213	0,003293	0,002048
IP	576		480,000000	96,000000	36,000000	2,250000	0,460800	0,046080	0,029729	0,007408	0,004608
Ethernet	1.518		1.265,000000	253,000000	94,875000	5,929688	1,214400	0,121440	0,078348	0,019524	0,012144
FR	8.192		6.826,666667	1.365,333333	512,000000	32,000000	6,553600	0,655360	0,422813	0,105363	0,065536
TCP	65.534		54.611,666667	10.922,333333	4.095,875000	255,992188	52,427200	5,242720	3,382400	0,842881	0,524272

1kbit/s = 1000 bit/s !!!
1KByte = 1024 Byte !!!

Propagation (Signal) Delay

Tp = **<u>Propagation Delay</u>** (in ms) = [(Distance in m) / (velocity in m/sec)] * 1000

			v=200.000km/s	v=300.000km/s
	Distance		**Delay in msec (10^{-3})**	**Delay in msec (10^{-3})**
CPU Bus	**10 cm**		**0,0000005**	**0,0000003**
	1 m		**0,0000050**	**0,0000033**
RS232, V24/V.28	**15 m**		**0,0000750**	**0,0000500**
LAN, Copper, RJ45	**100 m**		**0,0005000**	**0,0003333**
LAN, FO, X.21/V.11-V.10	**1 km**		**0,0050000**	**0,0033333**
Local Subscriber Line	**2,5 km**		**0,0125000**	**0,0083333**
WAN Link Repeater	**10 km**		**0,0500000**	**0,0333333**
WAN Link Repeater	**100 km**		**0,5000000**	**0,3333333**
WAN FO Link Repeater	**1.000 km**		**5,0000000**	**3,3333333**
WAN FO Link Repeater	**10.000 km**		**50,0000000**	**33,3333333**
Satellite Link	**40.000 km**		**200,0000000**	**133,3333333**
Satellite Link	**50.000 km**		**250,0000000**	**166,6666667**
	100.000 km		**500,0000000**	**333,3333333**
	300.000 km		**1500,0000000**	**1000,0000000**

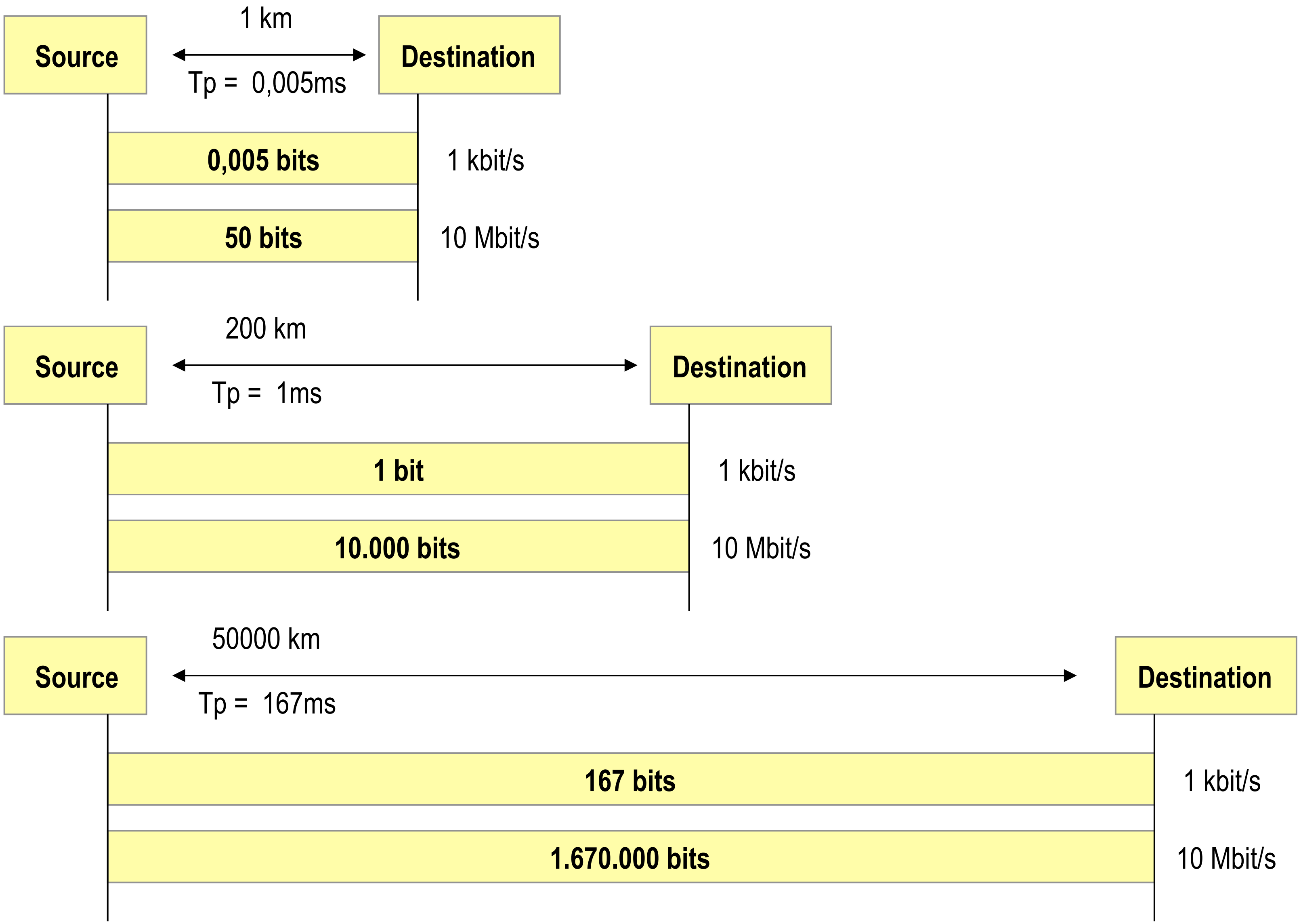
Total Delay (for a block of bits)
= Serialization Delay + Propagation Delay + (Switching Delay)

How Long Is A Bit?

Length (in m) = [(1 / (bitrate per sec)] * [(velocity in m/sec)]

	Bitrate		Bit Length in meter	Bit Length in meter
Analogue Modem	9,6 kbit/s		20833,33	31250,00
Analogue Modem	48 kbit/s		4166,67	6250,00
DS0	64 kbit/s		3125,00	4687,50
ISDN (2B)	128 kbit/s		1562,50	2343,75
PCM-30, E1	2,048 Mbit/s		97,66	146,48
Token Ring 4	4 Mbit/s		50,00	75,00
Ethernet	10 Mbit/s		20,00	30,00
Token Ring16	16 Mbit/s		12,50	18,75
Fast Ethernet, FDDI	100 Mbit/s		2,00	3,00
ATM STM1, OC-3	155 Mbit/s		1,29	1,94
ATM STM4, OC-12	622 Mbit/s		0,32	0,48
Gigabit Ethernet	1 Gigabit/s		0,20	0,30
OC-48	2,5 Gigabit/s		0,08	0,12
10 Gigabit Ethernet	10 Gigabit/s		0,02	0,03
			Copper	LWL - Free Space
			200.000 km /sec	300.000 km / sec

Propagation Delay And Number Of Bits On A Given Link



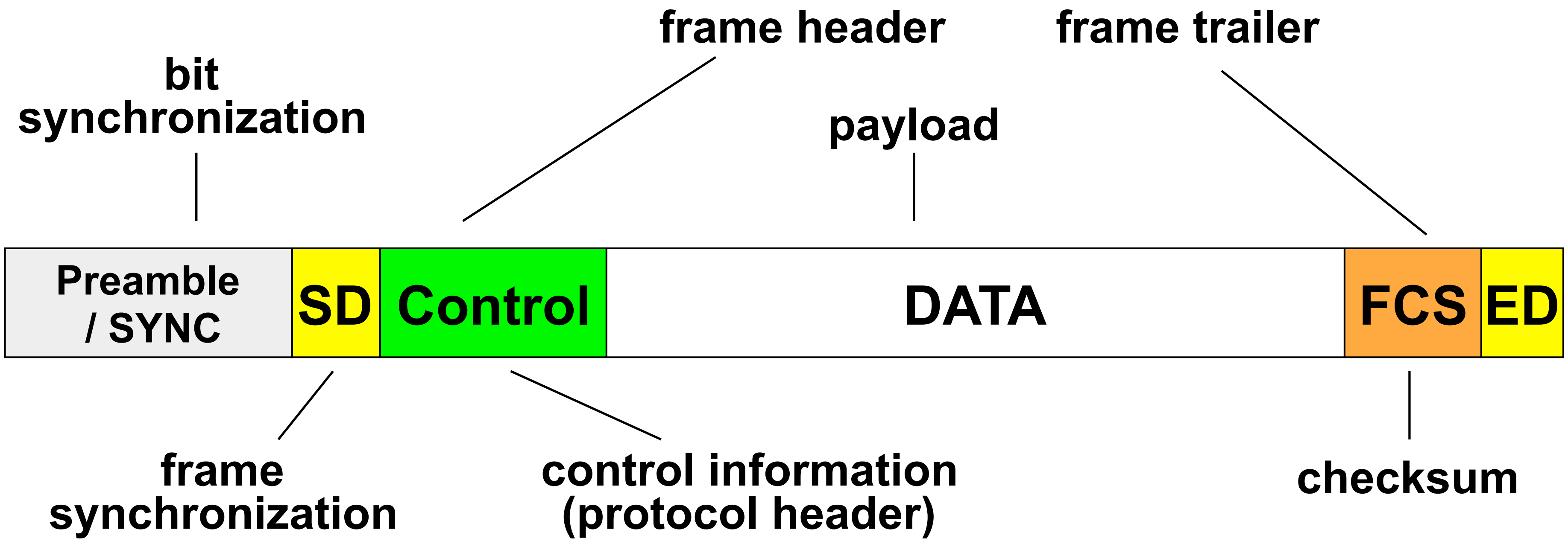
Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **<u>Transmission Frame</u>**
  - <u>Generic Format</u>
  - Frame Synchronization
  - Error Control

Requirements & Facts Serial Transmission System

- **Information between systems is exchanged in blocks of bits**
  - Every block is carried in as so called transmission frames
- **The recognition of the beginning and the end of a block in the received bit stream is necessary**
  - Frame synchronization
- **Errors on physical lines may lead to damage of digital information**
  - 0 becomes 1 and vice versa
  - The longer the block the higher the probability for an error
- **Methods necessary for error checking**
  - Frame protection
  - Error detection and recovery

Generic Frame Format



SYNC - Sync Pattern
SD - Starting Delimiter
ED - Ending Delimiter
FCS - Frame Check Sequence

Preamble

- **Preamble / SYNC is a special bit pattern**
  - Used for bit synchronization after an idle period (Preamble)
  - Can be used as fill pattern during idle times to keep the receiver clock synchronized (SYNC)
- **Enables PLL synchronization**
  - Typically a 0101010...-pattern
  - Example: 8 Byte preamble in Ethernet frames

Preamble / SYNC	**SD**	**Control**	**DATA**	**FCS**	**ED**

Control Field

- **Is used for implementing protocol procedures**
- **Contains information such as**
  - Frame type, protocol type
    - Data, Ack, Nack, Connect, Disconnect, Reset, etc.
    - IP, IPX, AppleTalk, etc.
  - Sequence numbers for identification of frame sequence
    - Necessary for error recovery and flow control with connection oriented services
  - Address information of source and destination in case of a multipoint line
  - Frame length, etc.

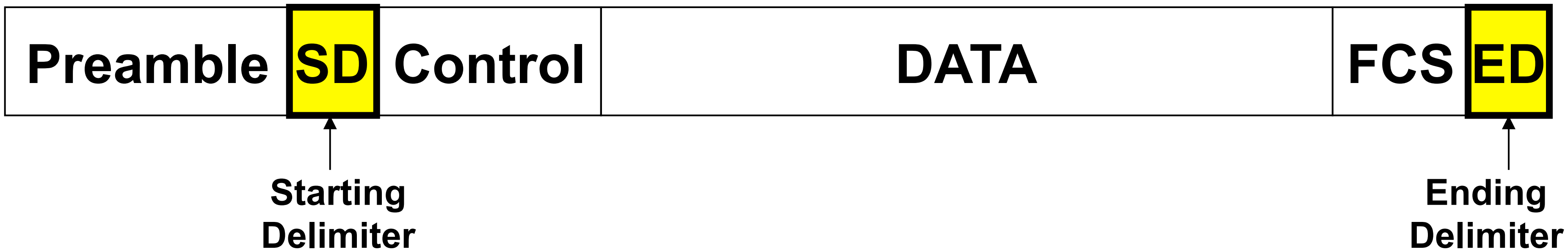
SYNC	**SD**	**Control**	**DATA**	**FCS**	**ED**

Agenda

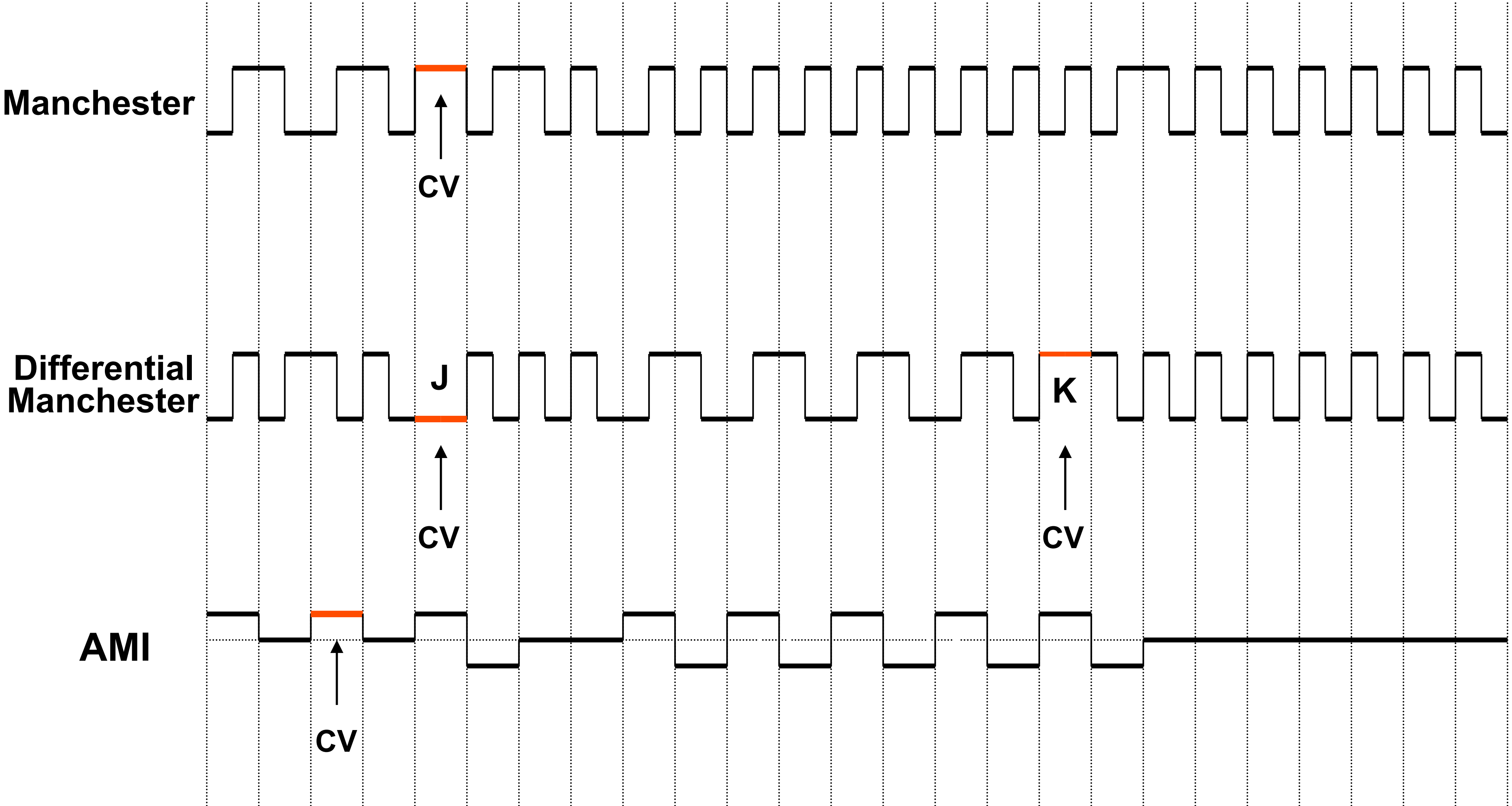
- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **<u>Transmission Frame</u>**
  - Generic Format
  - <u>Frame Synchronization</u>
  - Error Control

Frame Synchronization

- **Beginning and ending of a frame is indicated by SD and ED symbols**
  - Bit-patterns or code-violations
  - Length-field can replace ED (802.3)
  - Idle-line can replace ED (Ethernet)
- **Also called "Framing"**

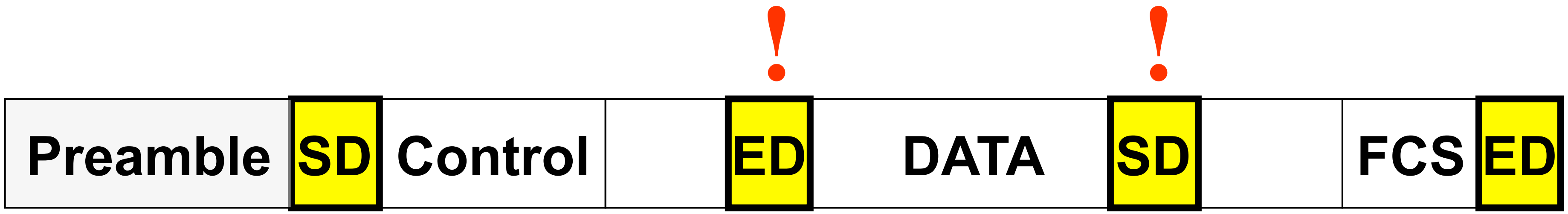


Examples For Code Violations



Protocol Transparence

- **What, if delimiter symbols SD, ED occur within frame?**
- **Solution:**
  - Byte-Stuffing
  - Bit-Stuffing



Character-Oriented Transmission
ASCII-Code

American Standard Code for Information Interchange

Bit Positions			7 6 5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
4	3	2	1								
0	0	0	0	Nul	DLE	SP	0	@	P	\	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	“	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	`	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	l
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

Transmission Control | Format Control | Printable Character | Information Separator | Others

Byte-Stuffing

- **Some character-oriented protocols divide data stream into frames**
  - Old technique, not so important today
  - e.g. IBM BSC (Binary Synchronous Control) protocol

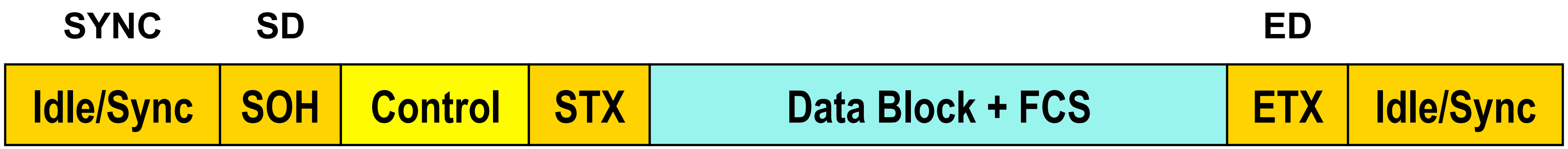
- **Data Link Escape (DLE) character indicates special meaning of next character**

Data to send:

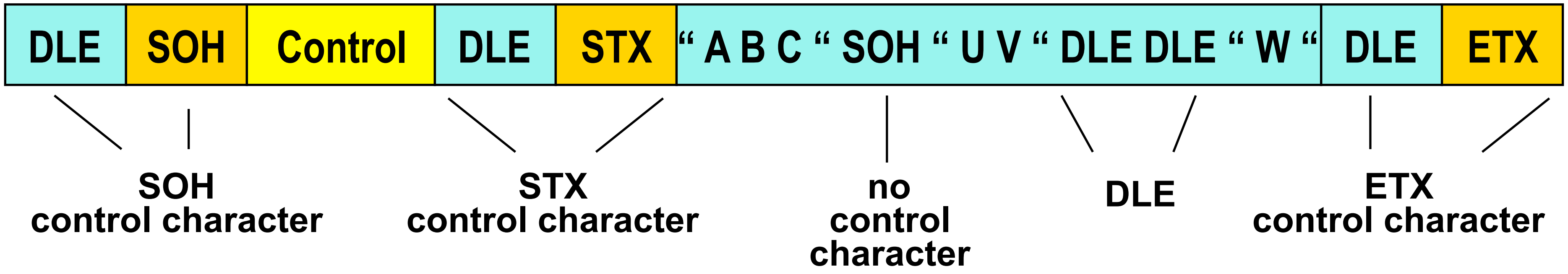
A B C DLE E F G ETX H I STX H

DLE	STX	A B C DLE DLE E F G ETX H I STX H	DLE	ETX

Character Based Transmission With And Without Protocol Transparence



Transmission in non-data transparent mode; control character *not* allowed in data block



Transmission in data-transparent mode with byte-stuffing; control character allowed in data block

Byte-stuffing: DLE inside data portion will be doubled by sender; receiver deletes this doubled DLE

Bit-Stuffing

- **Used in bit-oriented protocols**
  - Used by most protocols
  - Bits represent smallest transmission unit
- **HDLC-like framing: 01111110-pattern**
- **Rule:**
  - Transmitter-HW inserts a zero after five ones
  - Receiver rejects each zero after five ones

Data to send:

0100111110001111111100101100110

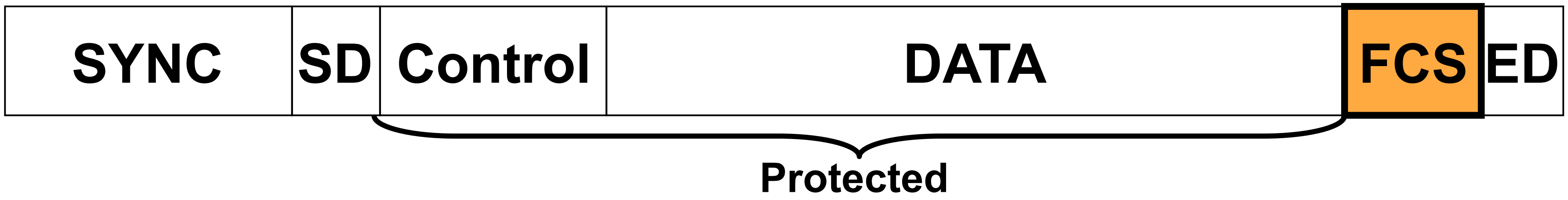
01111110 | **01001111100000111110110010110011** | **01111110**

Agenda

- **Introduction**
- **Bit Synchronization**
  - Asynchronous
  - Synchronous
- **Physical Aspects**
  - Mathematical Background
  - Communication Channel / Modulation
  - Serialization / Propagation Delay
- **Transmission Frame**
  - Generic Format
  - Frame Synchronization
  - Error Control

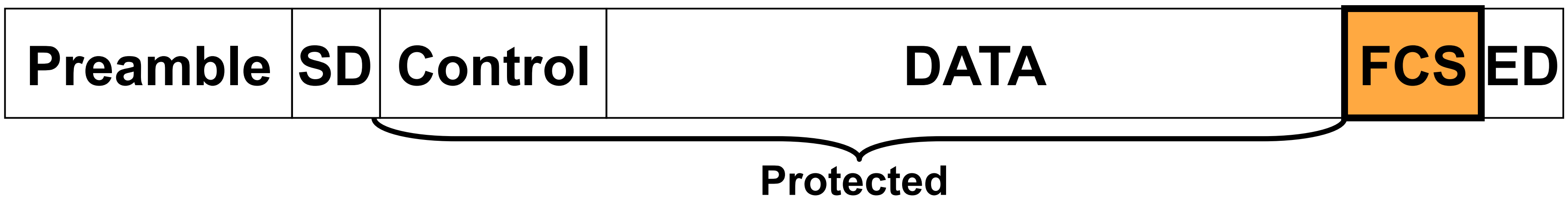
Error Control

- **Focus on error detection**
  - Include enough redundant information with each block of date to enable receiver to detect only errors occurred -> error detecting codes -> Frame Check Sequence
  - After error detection a retransmission of frame is initiated through protocol feedback to the sender
    - Area of ARQ-techniques
    - Feedback Error Control



Frame Protection

- **A frame check sequence (FCS) protects the integrity of our frame**
  - From Sunspots, Mobile-Phones, Noise, Heisenberg and others
- **FCS is calculated upon data bits**
  - Different methods based on mathematical efforts: Parity, Checksum, CRC
- **Receiver compares its own calculation with FCS**



FCS Methods

- **Parity**
  - Even (1001110111) or odd (1001110110) parity bits
  - Examples: Asynchronous character-transmission and memory protection
- **Checksum**
  - Module 2 sum without carry bit (XOR operation)
  - Many variations and improvements
  - Examples: TCP and IP Checksums

Checksum Example: ISBN

- **100% Protection against**
  - Single incorrect digits
  - Permutation of two digits
- **Method:**
  - 10 digits, 9 data + 1 checksum
  - Each digit weighted with factors 1-9
  - Checksum = Sum modulo 11
  - If checksum=10 then use "X"

ISBN 0-13-086388-2

0*1+1*2+3*3+0*4+8*5+6*6+3*7+8*8+8*9 = 244

244 modulo 11 = 2

Cyclic Redundancy Check

- **CRC is one of the strongest methods**
- **Bases on polynomial-codes**
  - Protected bits are used as coefficients of polynomial
  - This polynomial is mod 2 divided by a generator-polynomial
  - The rest is the CRC-Checksum
  - Bit error burst with a maximal length of generator-polynomial are detected 100%
- **Several standardized generator-polynomials**
  - CRC-16: $x^{16}+x^{15}+x^{2}+1$
  - CRC-CCITT: $x^{16}+x^{12}+x^{5}+1$

Error Control

- **Focus on error correction**
  - Include enough redundant information with to enable receiver to correct errors occurred -> error correcting codes ECC (important -> “Hamming Distance”)
  - Forward Error Control (FEC)
  - Required for "extreme" conditions
    - High BER (Bit Error Rate), EMR
    - Long delays, space links
  - Examples: Reed-Solomon codes, Hamming-codes