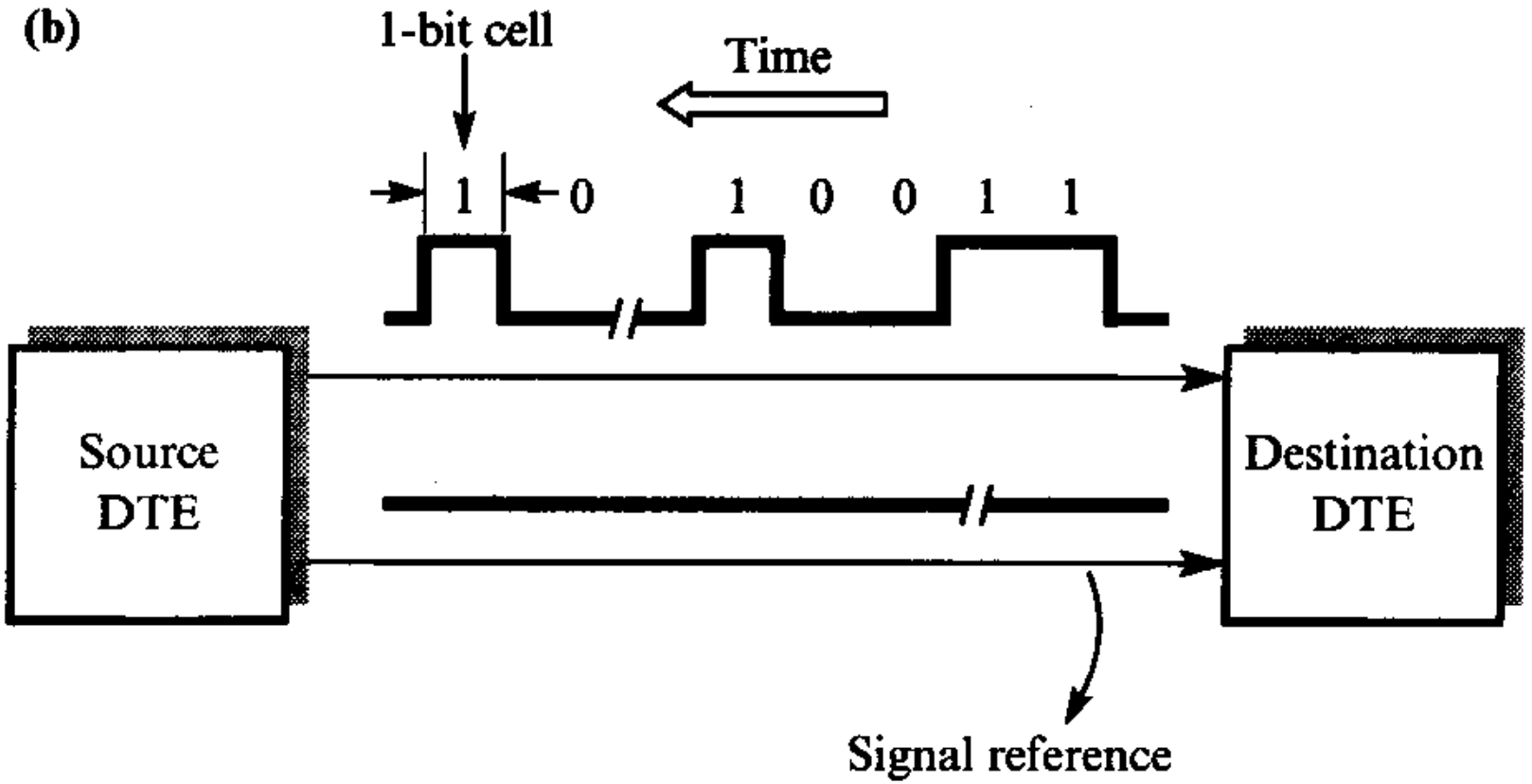
DATA TRANSMISSION

- Most digital messages are longer than just a few bits.
- It is neither practical nor economic to transfer all bits of a long message simultaneously.
- The message is broken into smaller parts and transmitted sequentially.
- Bit-serial transmission conveys a message one bit at a time through a channel.
- Each bit represents a part of the message.

BIT-SERIAL DATA TRANSMISSION

- The individual bits are then reassembled at the destination to compose the message.
- In general, one channel will pass only one bit at a time.
- Thus, bit-serial transmission is necessary in data communications if only a single channel is available.
- Bit-serial transmission is normally just called serial transmission and is the chosen communications method in many computer peripherals.

BIT-SERIAL DATA TRANSMISSION

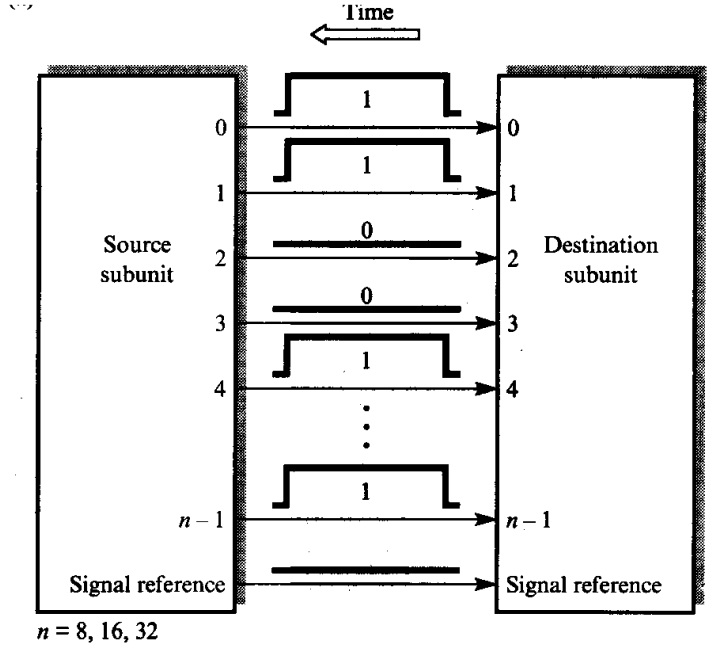


BYTE-SERIAL DATA TRANSMISSION

- Byte-serial transmission conveys eight bits at a time through eight parallel channels.
- Although the raw transfer rate is eight times faster than in bit-serial transmission, eight channels are needed, and the cost may be as much as eight times higher to transmit the message.
- When distances are short, both techniques may be feasible and economic to use parallel channels in return for high data rates.
- The popular Centronics printer interface is a case where byte-serial transmission is used.

BYTE TRANSMISSION

- As another example, it is common practice to use a 16-bit-wide data bus to transfer data between a microprocessor and memory chips.
- This provides the equivalent of 16 parallel channels.
- On the other hand, when communicating with a timesharing system over a modem, only a single channel is available, and bit-serial transmission is required.



SYNCHRONIZATION

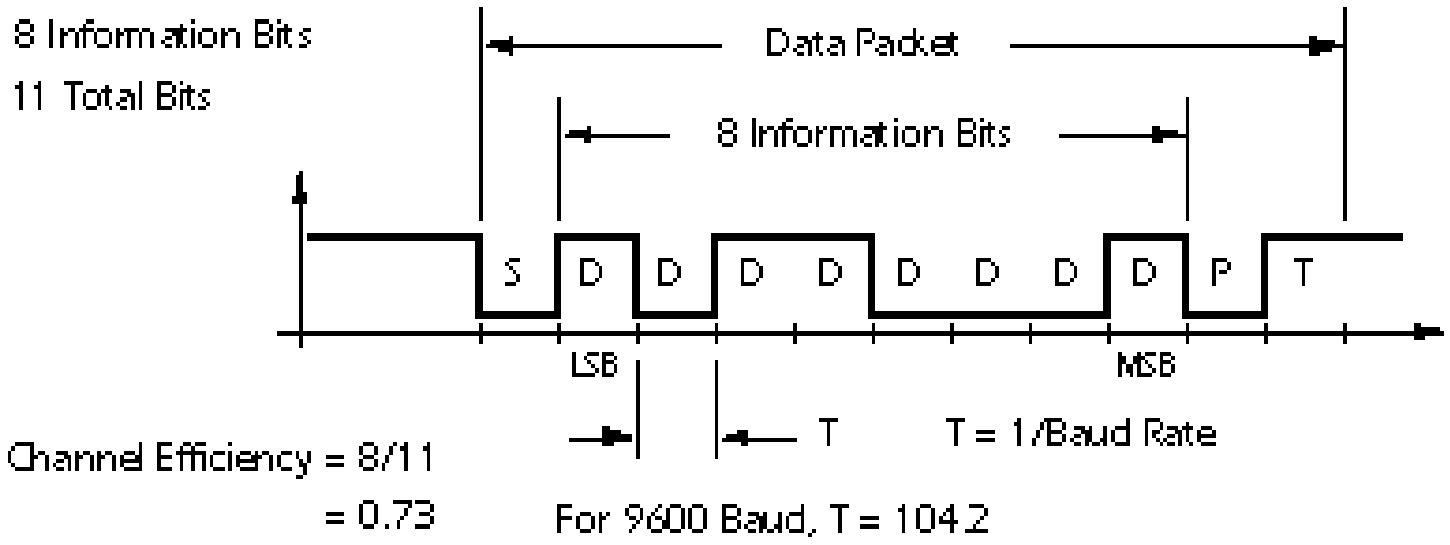
Two types of transmission:

- Asynchronous
- Synchronous

Asynchronous Transmission

In asynchronous transmission, each character is transmitted separately with separate synchronization information. This type of transmission is often used in situation when characters may be generated at random intervals, such as when a user types at a terminal. In asynchronous transmission, all of the bits that comprise a character are framed and then sent as a single transmission string.

ASYNCHRONOUS TRANSMISSION



ASYNCHRONOUS TRANSMISSION

Start and Stop Bit Framing

The clocks of the transmitter and the receiver are not continually synchronized. But the receiver needs to know when the character begins and ends. For this reason, the character's bit string is framed with start and stop bits. The start bit reset the receiver's clock so that it matches the transmitter's. The clock only needs to be accurate enough to stay in sync for the next 8 to 11 ticks. At least one stop bit is added to mark the end of the character and allow recognition of the next start bit.

ASYNCHRONOUS TRANSMISSION

Error Detection

One way to detect errors in asynchronous transmission is to add an extra bit, called a parity bit, to the end of each character in a frame.

Summary:

- Uncomplicated and inexpensive but slow and
- Overhead for each character.

SYNCHRONOUS TRANSMISSION

In some application it is necessary for large blocks of data, such as the contents of a disk file, to be transmitted. Synchronous transmission is more efficient method of transmitting large block of data. The data are usually buffered and transmitted as an entire message or frame. For this reason, clocks on both sides must maintain synchronization during transmission. This is accomplished in one of two ways:

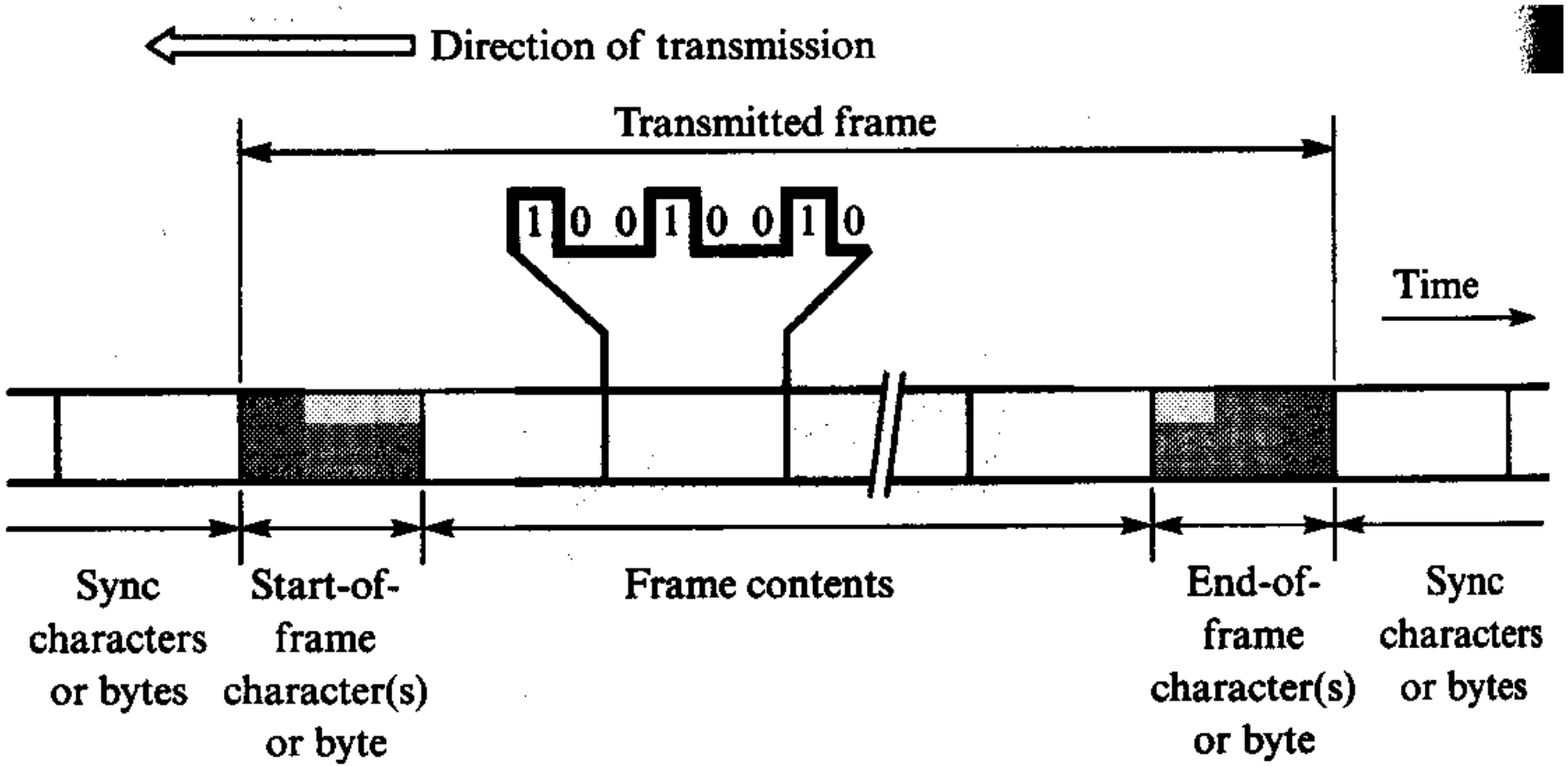
1. A separate synchronizing signal, a clock, can be constantly transmitted
2. Clocking can be included in the data signal

SYNCHRONOUS TRANSMISSION

Error Detection

Cyclic Redundancy Check (CRC) is often used in synchronous transmission. The CRC process subjects the block of data to an algorithm that computes a result based on its contents. This result is appended to the block prior to transmission. On the receiving side, the same algorithm is used and the result is compared with the CRC field. Any difference is assumed to mean frame damage during transmission and retransmission is repeated.

More efficient, higher speed and improved error detection, But expensive and complex transmitter/receiver circuitry.


Direction of transmission
Transmitted frame
1 0 0 1 0 0 1 0
Time
Sync characters or bytes
Start-of-frame character(s) or byte
Frame contents
End-of-frame character(s) or byte
Sync characters or bytes

ERROR DETECTION METHODS

- Noise and momentary electrical disturbances may cause data to be changed as it passes through a communications channel.
- If the receiver fails to detect this, the received message will be incorrect, resulting in possibly serious consequences.
- As a first line of defense against data errors, they must be detected.
- If an error can be flagged, it might be possible to request that the faulty packet be resent, or to at least prevent the flawed data from being taken as correct.
- If sufficient redundant information is sent, one- or two-bit errors may be corrected by hardware within the receiver before the corrupted data ever reaches its destination

ERROR DETECTION METHODS

- A parity bit is added to a data packet for the purpose of error detection.
- In the even-parity convention, the value of the parity bit is chosen so that the total number of '1' digits in the combined data plus parity packet is an even number.
- Upon receipt of the packet, the parity needed for the data is recomputed by local hardware and compared to the parity bit received with the data.
- If any bit has changed state, the parity will not match, and an error will have been detected.
- In fact, if an odd number of bits (not just one) have been altered, the parity will not match.
- If an even number of bits have been reversed, the parity will match even though an error has occurred.
- However, a statistical analysis of data communication errors has shown that a single-bit error is much more probable than a multibit error in the presence of random noise.
- Thus, parity is a reliable method of error detection.

PARITY METHOD

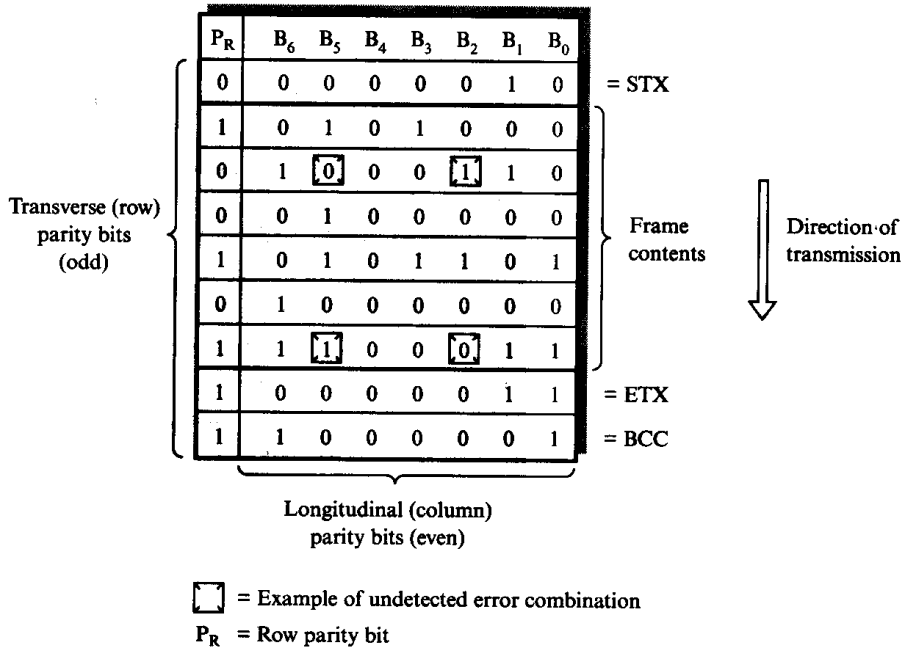
Even-Parity Computation

Data	Parity Bit
1 0 1 1 0 0 0 1	0
1 0 0 0 0 1 1 0	1

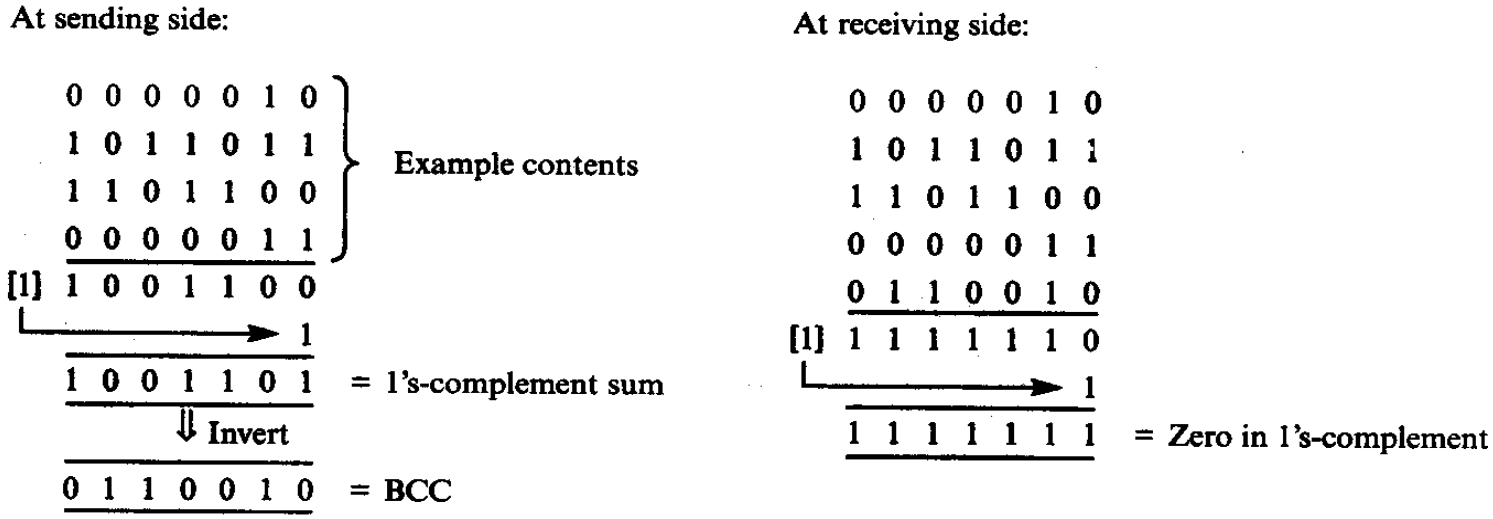
BLOCK SUM CHECK

- Another approach to error detection involves the computation of a checksum. In this case, the packets that constitute a message are added arithmetically.
- A checksum number is appended to the packet sequence so that the sum of data plus checksum is zero.
- When received, the packet sequence may be added, along with the checksum, by a local microprocessor. If the sum is nonzero, an error has occurred.
- As long as the sum is zero, it is highly unlikely (but not impossible) that any data has been corrupted during transmission.

BLOCK SUM CHECK: ROW AND COLUMN PARITY BITS



BLOCK SUM CHECK: 1'S COMPLEMENT SUM



CYCLIC REDUNDANCY CHECK

- With Synchronous Transmissions, the transmitter uses an algorithm to calculate a CRC value, which is appended to the frame.
- The receiver uses the same algorithm, recalculates the CRC and compares it to the CRC in the frame.
- If the values match, it is almost certain there is no errors.

OPERATORS

The main operators are as follows:

1.	AND	&
2.	OR	|
3.	XOR	^|
4.	Left Bit Shift	<<
5.	Right Bit Shift	>>

AND OPERATOR

Bitwise operator AND returns a 1 if both operand bits are 1.

1 & 1 = 1
0 & 1 = 0
1 & 0 = 0
0 & 0 = 0

OR OPERATOR

Bitwise operator OR returns a 1 if at least one operand is a 1.

1 | 1 = 1
0 | 1 = 1
1 | 0 = 1
0 | 0 = 0

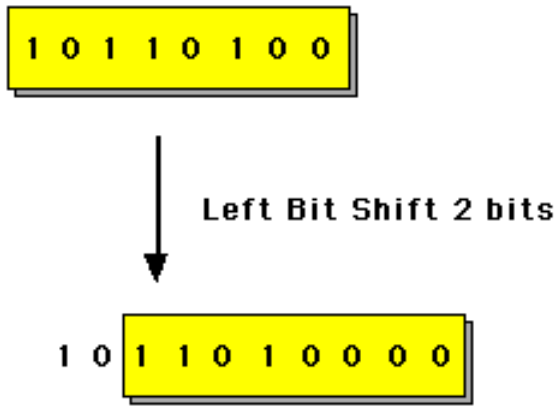
XOR

Bitwise XOR (exclusive OR) returns a 1 if only one of the operands is a 1.

1 ^ 1 = 0
0 ^ 1 = 1
1 ^ 0 = 1
0 ^ 0 = 0

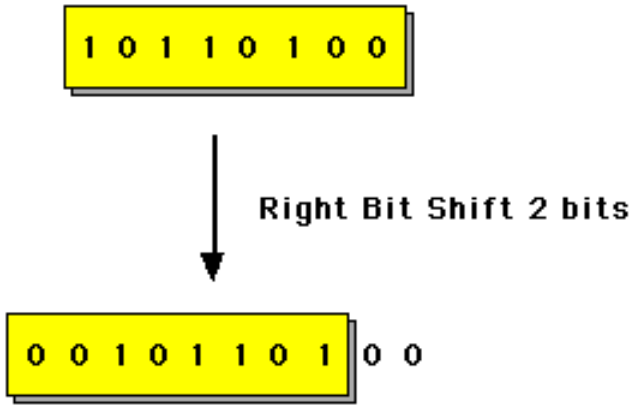
LEFT BIT SHIFT

Left Bit Shift moves the bit values left to a specified position. Bits to the left are moved out of focus and the new bits on the right are set to zero.

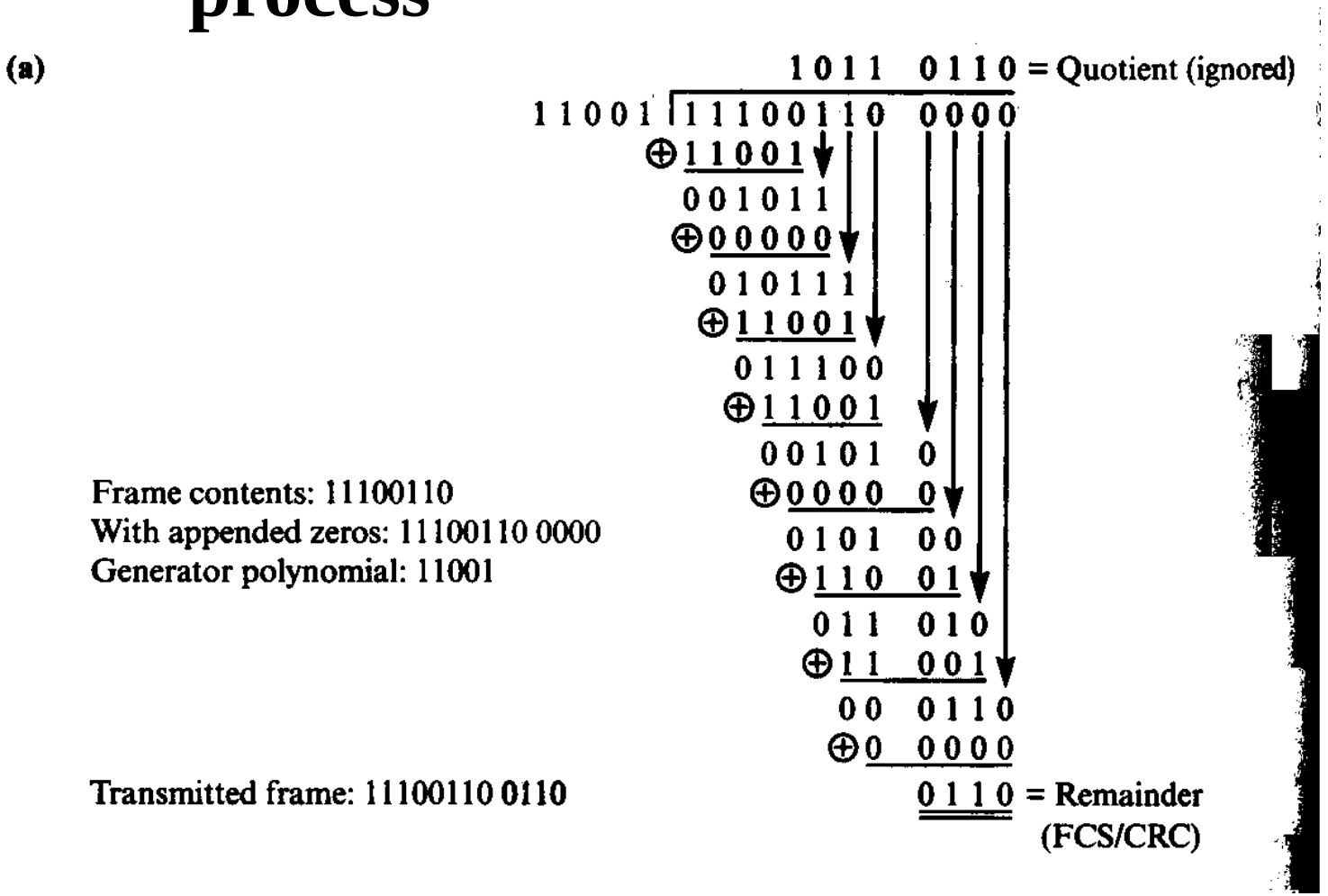


RIGHT BIT SHIFT

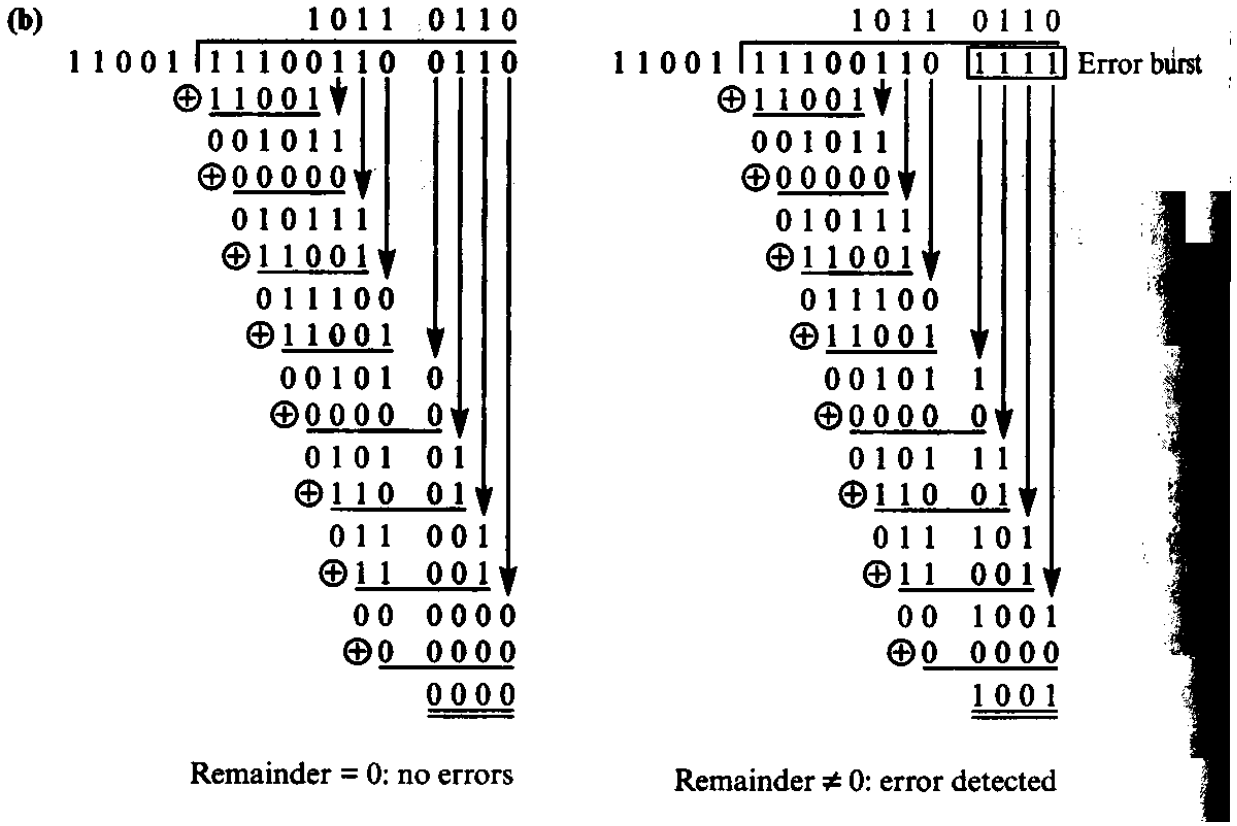
Right Bit Shift move the bit values right to a specified position. Bits to the right are moved out of focus and the new bits on the left are set to zero.



The Frame Check Sequence (FCS) generation process



The Frame Check Sequence (FCS) checking process



ERROR CORRECTION

- Errors may not only be detected, but also corrected if additional code is added to a packet sequence.
- If the error probability is high or if it is not possible to request retransmission, this may be worth doing.
- However, including error-correcting code in a transmission lowers channel efficiency, and results in a noticeable drop in channel throughput.