Course 8: Bit synchronization

Agenda

- Introduction: a synchronization-related systems view
- Step-by-step synchronized systems (fully asynchronous systems)
- Bit-level synchronized systems
- Fully synchronized systems

Introduction

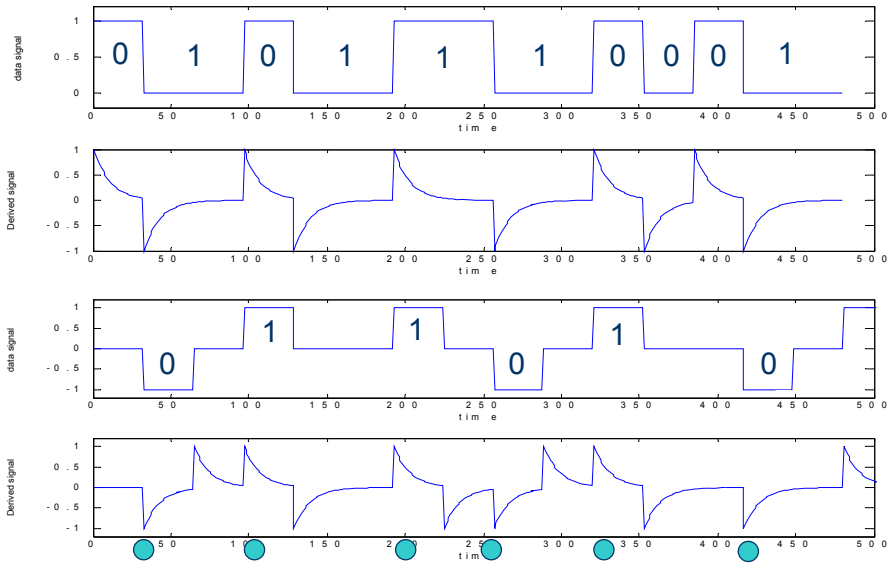
- Synchronous systems:
  - every bit has the same duration, T
  - The same time interval is preserved between any two consecutive bits (multiple of T)
- Bit-level synchronized systems
  - Referred to as “start/stop” asynchronous systems
  - Synchronization realized at bit level
  - No synchronization between two successive characters (words)
- Step-by-step synchronized systems
  - Fully asynchronous systems
  - No synchronization neither at bit, nor at character level

Principles

- No clock signal needed at reception
- The signal corresponding to every bit must have at least one transition
- Synchronization ensured from the data signal
- The system can be considered as “self-synchronized”
- For high-data rates, asynchronous transmission is not recommended

Examples

Legend:
- pulses used for detection



Comments on the previous figure

- Two examples of step-by-step synchronized systems are given
- Bit intervals are not equal, nor the time between two successive bits
- In the first example, every pulse in the derived signal indicates that a new bit was transmitted
  - By measuring the time-interval between two pulses, “0” or “1” is decided
  - Short duration between pulses means “0”, long duration means “1”
- In the second example, every pair of pulses in the derived signal corresponds to a bit
  - The odd-index pulses are used for detection
  - Negative odd pulse means “0”, positive means “1”

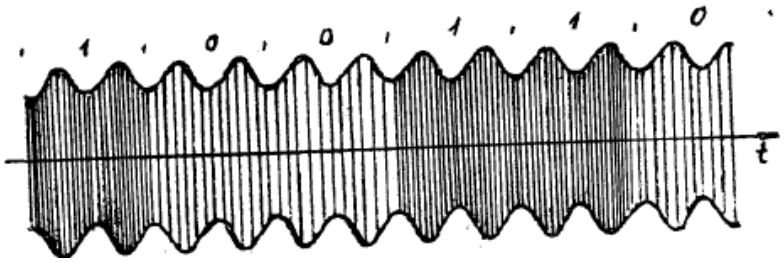
Fully synchronous systems

- Same time interval between two consecutive bits
- For detection, the signal must be sampled at the “right” times (e.g. : where there is no ISI)
- Locally generated clock needed at receiver
- This clock must be synchronous with the time basis used by the transmitter
- Example of free-ISI transmission using sinc pulse, on the last slide

Requirements

- The timing information must be transmitted
- Synchronization types:
  - Independent synchronization: separate, dedicated clocking information transmitted along with the signal
  - Dependent synchronization: the clocking information incorporated in the data (e.g. transitions)
- Synchronization guaranteed during the transmission breaks (independent synchronization)
- Special Sync character must be transmitted (dependent synchronization)

In-channel synchronization



- FM signal for data transmission
- AM used for synchronization (see the envelope)
- SNR is impacted negatively, since some energy is consumed only for synchronization purposes

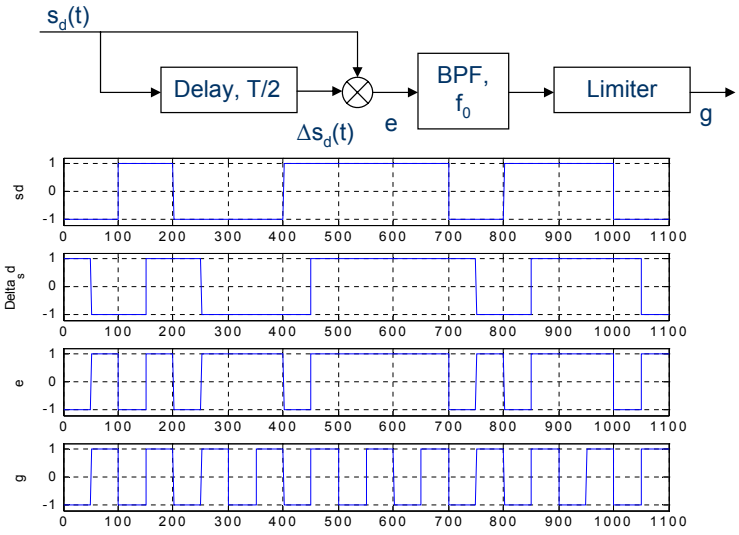
Out-of-channel synchronization

- Synchronization signal transmitted out of band
- Parasite modulation avoided
- Major drawback ☹: the synchronization signal is differently affected by the channel (compared to the data signal)
- Supplementary efforts required to compensate the frequency selectivity

Dependent synchronization

- Synchronization information extracted from the data signal
- Statistically speaking, the number of transitions is not high enough
- Synchronization of the receiver's time basis is based on a local oscillator
- Principle: a frequency component of 1/T (T=symbol time) must be extracted from the received spectrum

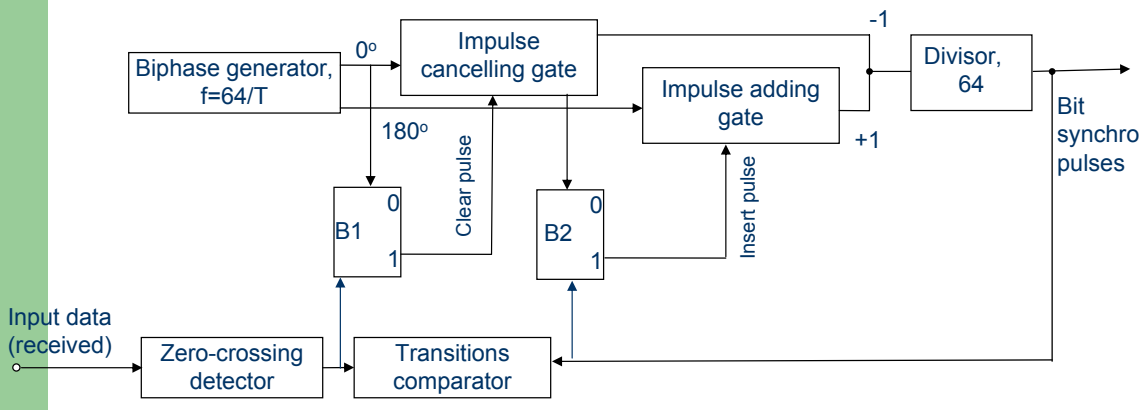
Dependent synchronization scheme



Comments on the previous scheme

- The scheme allows to extract a clock with the frequency 1/T
- By multiplication (the signal and its delayed version) the number of transitions doubles, easing the synchronization
- The FTB will extract the 1/T component, which, by limitation, is transformed into a rectangular train of pulses
- The signal g(t) can be seen as the time basis for the receiver

Digital approach: one-step synchronizer

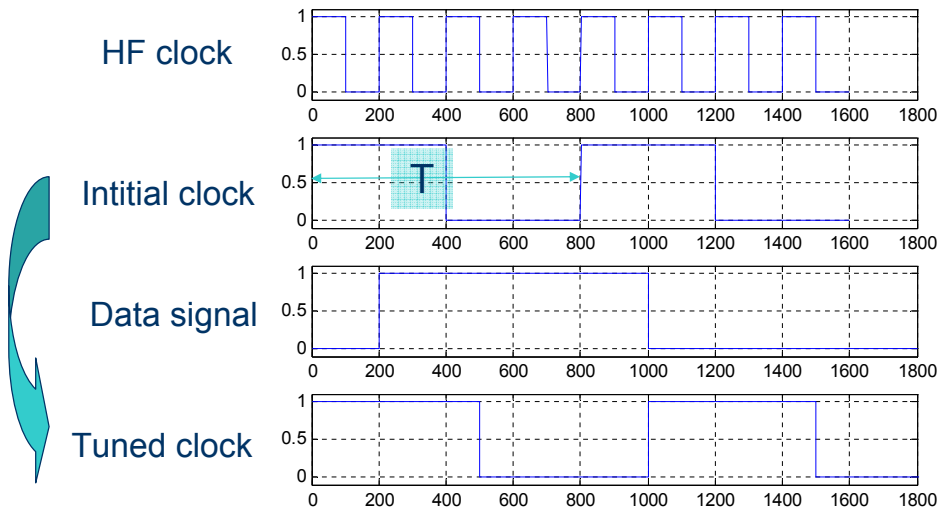


Input: data signal
Output: clock signal synchronized with data

One-step synchronizer: comments on the scheme

- Transition time instant compared to the clock transition
- If there is a time shift between them, the receiver's time basis is tuned
- The scheme uses a higher "resolution" clock (clock frequency: 64/T)
- Frequency divisors are used to tune the clock frequency

Graphical example (one-step synchronizer)



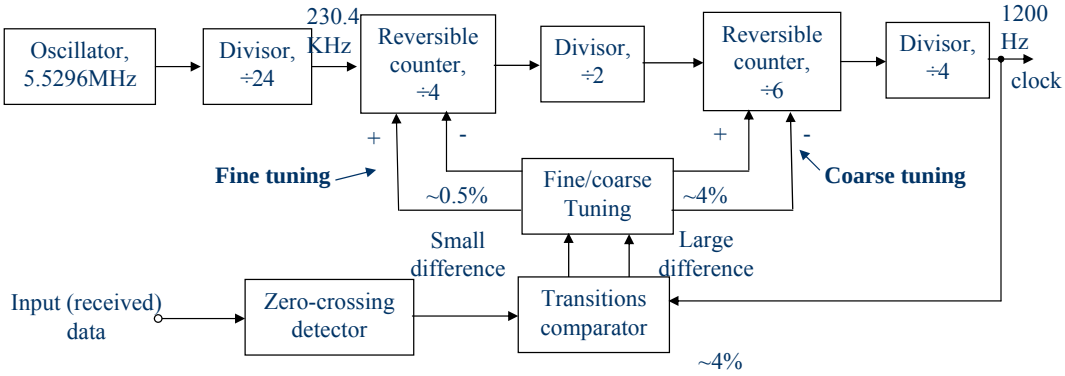
Comments on the previous figure

- For graphical illustration purposes, the high-frequency clock has the frequency 4/T
- If the data transition occurs after the clock transition (detection made by transitions comparator), one pulse is deleted (from the HF clock)
- After the divisor, the clock signal frequency will be lower
- For the opposite scenario (data transition occurring after the clock signal transition), one pulse is added
- This determines a higher frequency of the clock signal

Frequency divisor

- The principle of frequency divisor is very important for the understanding of the previous scheme
- A M-times frequency divisor, counts the pulses from the “high frequency” input signal and generate an output impulse when the counter reaches to M
- By deleting an impulse from the HF input signal, the counter will “saturate” later and the output pulse will be issued accordingly (see the figure from slide 16)
- By adding an impulse to the HF signal, the counter will saturate earlier, and the output pulse will be generated faster
- “The insert/clear” impulse operations are the key points of the synchronization circuit

Two-step synchronization



Input: data signal
Output: clock signal synchronized with data

Two-step synchronization

- Principle: the high and the low synchronization errors in the receiver's clock treated differently
- If the synchronization error is high, a coarse tuning takes place
  - An impulse is added/cleared, having a weight of 1/24 (≈4%)
  - The weight is given by the multipliers (4x6=24)
- If the synchronization error is small, a fine tuning takes place
  - Impulse added/cleared, having the weight 1/192
  - The weight results from 4x2x6x4=192

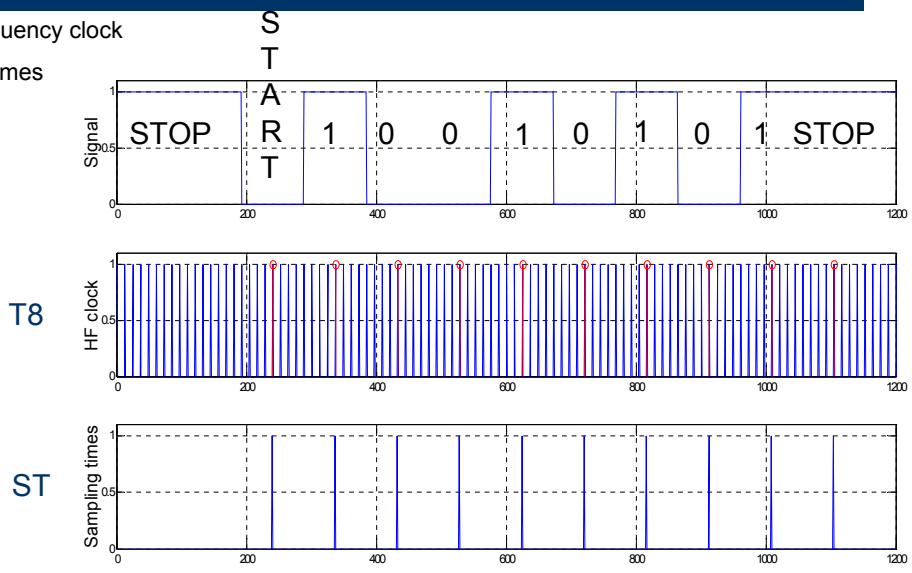
Start-stop asynchronous systems

- The break between two consecutive symbols has a lower limit, but does not have an upper limit
- During the breaks, a continuous voltage level exists on the line
- When transmission begins, a special START “character” is transmitted (NO signal)
- After the start, the bits composing the byte are transmitted, eventually followed by a parity check bit
- Inside of the byte: synchronization is needed !!!

Graphical illustration

T8- high frequency clock

ST-sample times



Comments on the previous figure

- The high-frequency clock (T8) has a period T_{HF}=T/8 (T is the bit duration)
  - Its frequency is eight times the bit rate
- After three consecutive zero-value samples, the fourth T8 pulse is considered as test pulse for the START message
- The sampling pulses (SP) are provided by a ÷8 divisor
- After 8 data signal sampling pulses, the 9th is considered as test pulse for the stop character

Fully synchronized systems