# UMM AL-QURA UNIVERSITY

College of Computer and Information Systems

**Computer Engineering Department**

**1403322**

**Computer Communication System**

**Lab Manual**

Student Name: ___________________________________

Student ID: ______________________________________

Section: ________________ Group: __________________

Session (Fall / Spring / Summer) ____________________

This page is intentionally left blank

# TABLE OF CONTENTS




Prepared By: Engr. Muhammad Saqib & Engr. Muhammad Asif Manzoor 2013


Revised By: ______________________ ____________

Reviewed By: ______________________ ____________

Approved By: ______________________ ____________

# Umm Al-Qura University

Computer Engineering Department

# LABORATORY SAFETY GUIDELINES

## A. General Laboratory Safety Rules

### 1. Personal Safety

- Be familiar with the electrical and fire hazards associated with your workplace.
- Be as careful for the safety of others as for yourself. Think before you act.
- Be tidy and systematic.
- Avoid bulky, loose or trailing clothes. Avoid long loose hair.
- No one is allowed to enter in the lab area bare foot due to increased risk of electric shock.
- Remove metal bracelets, rings or watchstraps when working in the laboratories.
- Avoid working with wet hands and clothing.

### 2. Food, Beverages and Smoking

- Due to the increased risk of electric shock, no drinking of beverages, consumption or storage of any kind of food is allowed in the laboratory.
- Smoking is prohibited in all laboratories in all timings.

### 3. Soldering

- No one is allowed to do soldering in any of the computer engineering laboratories except the graduation project design laboratory.
- Anyone doing soldering in the graduation project design laboratory must wear appropriate apparel, socks, gloves, covered shoes and safety goggles to prevent the possibility of severe burns resulting from the splashing or dripping of hot liquefied solder into the face and eyes or on to the exposed skin on the chest, hands, legs, and feet.
- Students who are not so properly attired for these tasks will NOT be allowed to perform any type of soldering in the graduation project design laboratory.

### 4. Laboratory Operating Hours

- Students are never allowed to work alone in any lab area other than scheduled laboratory operating hours unless either a Lab T/A or Course Instructor is present inside that lab area.
- The laboratory operating hours for students are posted on the entrance doorway and on the notice board of computer engineering department.

### 5. Power Supply Related Safety

- Voltages above **50-VAC** or **120-VDC** are always dangerous.
- Extra precautions should be considered as voltage levels are increased.
- Never make any changes to circuits or mechanical layout without first isolating the circuit by switching off and removing connections to power supplies.

### 6. Laboratory Equipment

- Lab equipment may not be removed from the Computer Engineering lab areas without the permission of the Laboratory Supervisors.
- Laboratory bench equipment (except for some lab bench computers) must be turned off before closing down the lab area for the day.
- Never open (remove cover) of any equipment in the laboratories.
- Never "jump," disable, bypass or otherwise disengage any safety device or feature of any equipment in the laboratories.
- Laboratories shall be locked when unoccupied.

### 7. Waste Management Safety

- Know the correct handling, storage and disposal procedures for batteries, cell, capacitors, inductors and other high energy-storage devices.

## 8. Equipment Safety

- Before equipment is energized ensure, circuit connections and layout have been checked by a Teaching Assistant (TA) and all colleagues in your group has given their consent.
- Experiments left unattended should be isolated from the power supplies. If for a special reason, it must be left on, a barrier and a warning notice are required.
- Equipment found to be faulty in any way should be reported to the lab supervisor and taken out of service until inspected and declared safe.

## 9. Equipment Accessories

- Use extension cords only when necessary and only on a temporary basis.
- Request new outlets if your work requires equipment in an area without an outlet.
- Discard damaged cords, cords that become hot, or cords with exposed wiring.

# B. Electrical and Fire Emergency Responses

## 1. Police, Fire or Medical Emergency

- Use the telephone located in the laboratory area and press **0-996** to notify police, fire, and ambulance for emergency help.
- Everyone present in the laboratory area shall be familiar with the locations and operation of safety and emergency equipment, including but not limited to, fire extinguishers, first aid kits, emergency power off system, fire alarm pull stations, and emergency telephones.

## 2. Electric Shock

- When someone suffers serious electrical shock, he may be knocked unconscious.
- If the victim is still in contact with electrical current, immediately turn off the electrical power source.
- If you cannot disconnect the power source, depress the Emergency Power Off switch.
- Do not touch a victim that is still in contact with a live power source; you could be electrocuted! Have someone call for emergency medical assistance immediately. Administer first-aid, as appropriate.

## 3. Electrical Fire

- If an electrical fire occurs, try to disconnect the electrical power source, if possible.
- If the fire is small and you are not in immediate danger; and if you have been properly trained in fighting fires, use the correct type of fire extinguisher to extinguish the fire.
- When in doubt, push in the Emergency Power Off button.
- **NEVER use water to extinguish an electrical fire**.

## 4. Emergency Power Off

- Every lab is equipped with an Emergency Power off System.
- When this switch is depressed, electrical power to the lab will shut off, except for lights.
- Only authorized personnel are permitted to reset power once the Emergency Power Off system has been engaged.

## 5. Building Evacuation in Emergency

- Everyone present in the laboratory should be familiar to emergency exits & way out plans.
- Use the nearest exit doorway from lab area closest to the stairwell to exit the building.
- Follow the Emergency Exit Signs posted in the hallways. Do not use elevators.
- Lab Teaching Assistants (T/As) or Instructor shall make sure all persons are out of the laboratory area and follow the directions posted at each doorway to the laboratory area.

The above general laboratory safety rules are designed to safeguard you and your co-workers, fellow students and colleagues and are a minimum requirement for individuals working in the computer engineering laboratories at Umm Al-Qura University, Makkah Al-Mukarramah. Specialized training and rules may apply depending on type and scope of activities involved.

This page is intentionally left blank

# Experiment # 1: Introduction

**736 93 DATA SOURCE / PARITY GENERATOR**

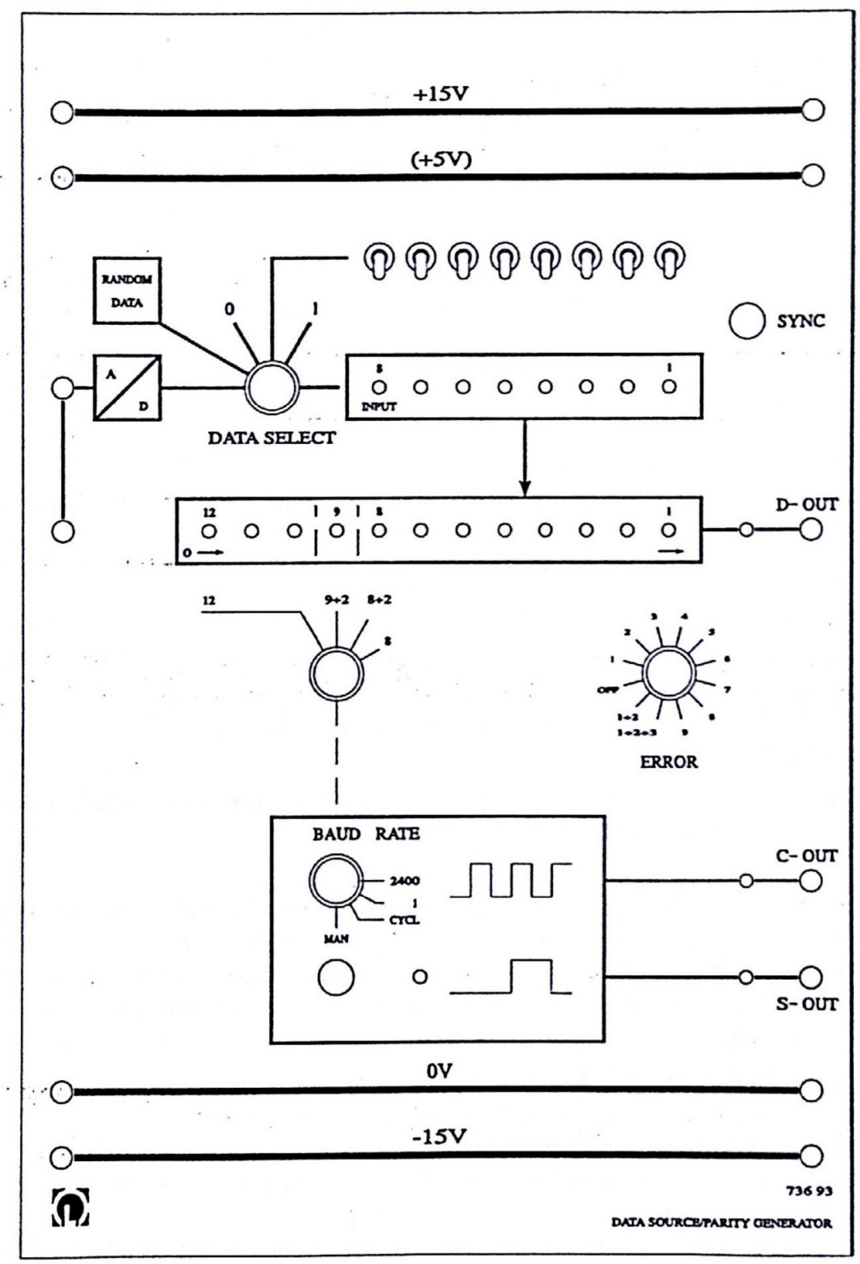



Fig. EQ-1 Front panel 736 93

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

*Baudrate:*

2400 : Transmission with 2400 bit/s, i.e. for a word length of

| | | | |
|---|---|---|---|
| 8 | : | 300 | Words/s |
| 8+2 | : | 240 | Words/s |
| 9+2 | : | 218 | Words/s |
| 12 | : | 200 | Words/s |

The next data word to be transmitted (if necessary including the check bit(s)) is already determined during the transmission of the current word.

1 : Continuous transmission with 1 bit/s.

Cycl : Transmission of a word with 1 bit/s.
The transmission is started by pressing the pushbutton under the baud rate switch. The LED next to this switch lights up, when the transmission of the last word is concluded and thus requesting the start of the next transmission.

Man : In the manual mode the clock pulse is directly determined by the push button:
button in idle state => C-Out = 0
button pressed => C-Out = 1
To indicate this mode the LED beneath the button blinks with 2 Hz.

In these three slow modes the next data word (if necessary, including check bit(s)) to be transferred is determined just before the transmission starts.

When the rotary switch is set to position 2400 bit/s, the transmission continues with the current bit. Conversely, when the switch is changed from the position 2400 bit/s to one of the slower modes the transmission of the current word is completed first.

*Synchronisation:*

There are different synchronisation mechanisms depending on the transmission mode:

Asynchronous modes (8+2 / 9+2):

SYNC button is pressed: D-Out immediately goes to a fixed level (1 = MARK)
C-Out provides clock pulses (when in manual mode too)
S-Out is 0

After a maximum of 10...11 clock pulses the actual word is completed (from the receiver's point of view) and it has to synchronise again with the next start bit.

SYNC button is released: start with a new word (start bit)

Synchronous modes (8 / 12):

SYNC button is pressed: D-Out immediately attains a fixed level (1 = MARK)
C-Out provides clock pulses (when in manual mode too)
S-Out is 0

In this state the display (data sink) is to be reset (using Reset or mode switchover) to internal synchronisation so that is falls out of the synchronisation state and begins searching for the synchronous byte again.

SYNC pushbutton is released: Transmission starts with the initial transmission of the synchronous byte 16h, then it continues in normal mode.

*Bit error creation:*

Using the rotary switch ERROR one or more bits can be inverted in data word transmitted (incl. parity bits) to simulate a faulty transmission. The start bit in the asynchronous mode is not affected! When the error switch is in position "9", the stop bit will be inverted in the mode 8+2 or the parity bit in the mode 9+2. The last two positions of the switch create a 3-fold error (bits 1, 2 and 3) or a 2-fold error (bits 1 and 2), respectively. In the "OFF" position (far left position) of the switch no errors are created.

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

*Outputs:*

The output signals each have a 4-mm socket with corresponding LED display:

D-Out : serial data output

C-Out : clock output (bit clock)

S-Out : synchronous output (word clock)

The synchronous output supplies high level during the first clock pulse of a transmission:

asynchronous modes (8+2 / 9+2) : during the start bit

synchronous modes (8 / 12) : during the 1st bit

The states of these three outputs are indicated with one LED each. (High => LED lights up). The outputs are TTL compatible (L = 0 V, H = 5 V)

*Power supply:*

The training panel is supplied with ±15 V.

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

## 736 92 DISPLAY / PARITY CHECK INDICATOR

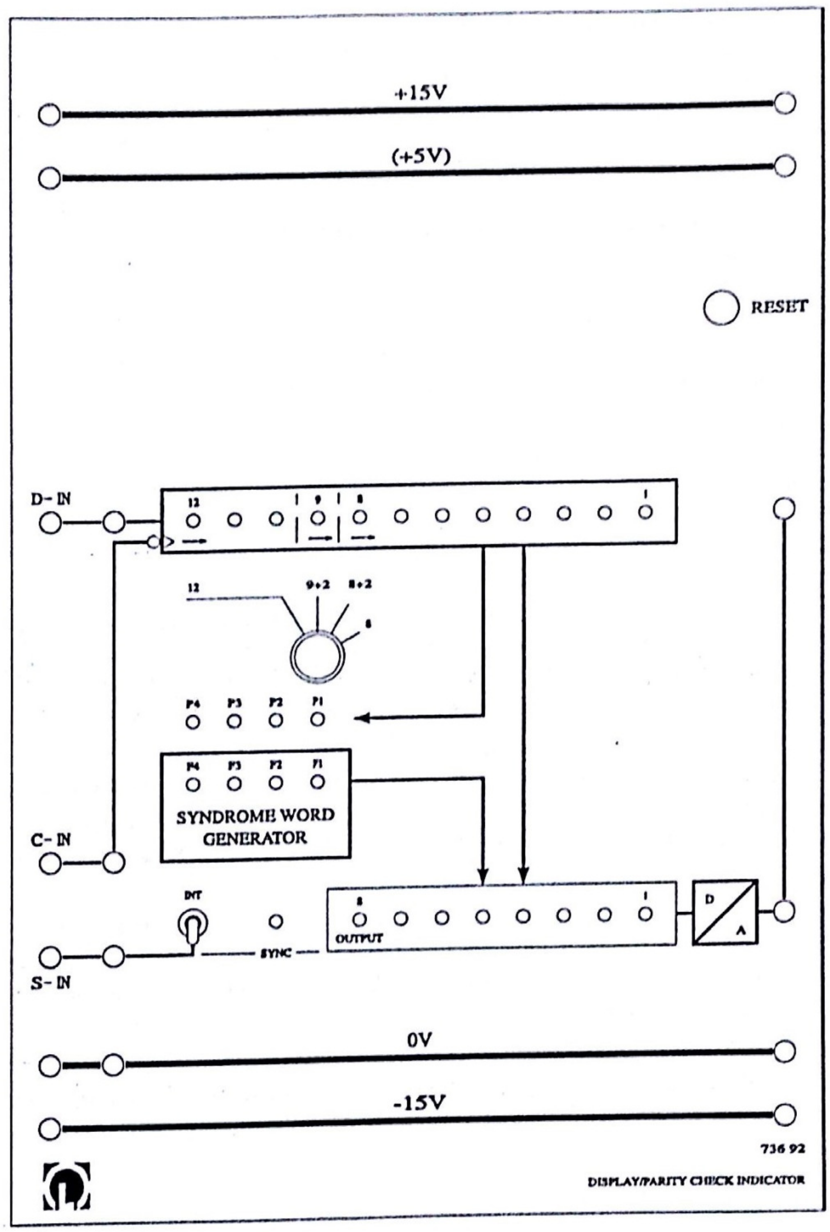



Fig. EQ-2 Front panel 736 92

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

The training panel DISPLAY / PARITY CHECK INDICATOR consists of the following block elements:

*Inputs:*
Two 4mm sockets for each of the following signals:
D-In : serial data
C-In : bit clock
S-In : synchronous signal, word clock
All inputs are TTL compatible and voltage proof up to ±42 V.

*Shift register:*
The shift register for the serial input of the data words is displayed by a row of 12 LEDs. Depending on the word length used, varying multiples of steps are employed. The 8 right bits are used for recording the original data word received from the data source. Bit 9 is the parity bit for the transmission mode with 1 parity bit. Bits 9...12 constitute the 4 check bits in the 12 bit transmission mode. Thus the input of the shift register is bit 12, 9 or 8, depending on the actual word length. After successful transmission the 8 data bits are transferred to the output register (where necessary including error correction).

*Transmission mode:*
4 different word lengths in 2 synchronisation modes are possible:
8 : 8 bit synchronous. After each 8 clock pulses a new transmission starts.
8+2 : 8 bit asynchronous. The transmission of a word starts with a start bit (always "0") and stops with a stop bit (always "1"). So a total transmission needs 10 clock pulses.
9+2 : 9 bit asynchronous. Here a parity bit is transmitted after the 8 data bits. Even parity is used, i.e. the total number of 1-bits is even (parity bit included, but without start and stop bit).
12 : 12 bit synchronous. In addition to the 8 data bits 4 check bits are transferred.

*Error indication in the mode 9+2 (asynchronous + parity bit):*
The parity bit is newly calculated from the 8 transferred data bits and indicated with the LED P1. If this parity bit does not coincide with the transferred parity bit (bit 9 in the shift register), an error has occurred. In this case the LED F1 in the field SYNDROME WORD GENERATOR lights up.

*Error correction in the 12 bit mode:*
Again the 4 parity bits are newly calculated from the 8 transferred data bits (LED's P1...P4). The syndrome word (LED's F1...F4 in the field SYNDROME WORD GENERATOR) is determined by a bit by bit comparison of the transferred bits with the newly calculated parity bits (XOR function). With the help of the syndrome word, 1-bit errors can be corrected and multiple bit errors can be detected.

*Synchronisation:*
The LED SYNC indicates successful synchronisation. The toggle switch located next to this LED is used to switch either to internal or external synchronisation. Note the differences between synchronous and asynchronous modes:

Asynchronous modes (8+2 and 9+2):

*Internal* (switch is in the upper position):
Word synchronisation is carried out again via the start bit (Recognition of a falling edge). The following bits are sampled using the clock of a free running oscillator. In order for this method to function properly, the receiver must be able to recognize the baud rate of the transmitter. Therefore, this mode only works at 2400 bit/s. The inputs C-In and S-In are not needed.

*External* (switch set to the lower position):
Again the word synchronisation is provided by the start bit. But now the signal from the input C-In is used as the bit clock, so that baud rates other than 2400 bit/s are possible.

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

Synchronous modes (8 and 12):

Sampling of single bits is always controlled by the signal C-In. The differences are in the word synchronisation:

*Internal* (set the switch to the upper position):

In the non-synchronised state (LED SYNC off) after the sampling of each bit a test is carried out to check whether the synchronous character (16h = 00010110b) is in the shift register (including the correct check bits in the 12 bit mode). Once the synchronous character has been recognized, the next bit represents the first one of the following word.

*External* (set switch to the lower position):

Here the synchronous signal at input S-In indicates the beginning of a new data word.

The pushbutton RESET is used to introduce a new synchronisation cycle. In the synchronous modes with internal synchronisation, a new synchronous byte (16h) has to be transmitted from the data source.

*Output register:*
Once successfully synchronised, the 8 data bits are transferred from the shift register to the output register (where necessary with error correction). This register is represented by the 8 LED's in the OUTPUT field.

*Analog output:*
The data word in the output register is additionally available as a voltage at the analog output (D/A): The voltage range is -10.24 V ... +10.16 V. The two's complement format is used to represent negative voltages:

| | | | | | | |
|---|---|---|---|---|---|---|
| -10.24 V | : | 10000000b | = | 80h | = | 128 (-128) |
| -10.00 V | : | 10000011b | = | 83h | = | 131 (-125) |
| 0.00 V | : | 00000000b | = | 00h | = | 0 |
| +10.00 V | : | 01111101b | = | 7Dh | = | 125 |
| +10.16 V | : | 01111111b | = | 7Fh | = | 127 |

*Power supply:*
The training panel is supplied with ±15 V.

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
| --- | --- | --- |

**736 94 AMI/HDB3 CODER**

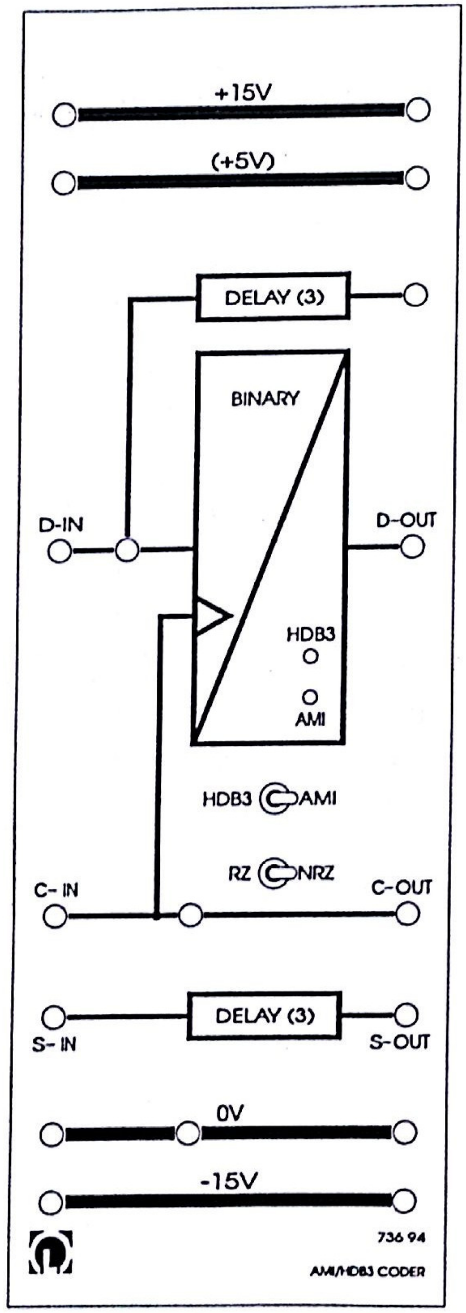



Fig. EQ-3 Front panel 736 94

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

The AMI/HDB3 CODER creates the channel codes AMI and HDB3.

*Inputs:*
D-In : serial data
C-In : bit clock
S-In : synchronous signal, word clock
The inputs are TTL compatible and voltage proof up to ±42 V.

*Outputs:*
D-Out : coded output signal (0 V, ±5 V), 3 clock pulses delayed with respect to D-In
C-Out : connected to C-In
S-Out : 3 clock pulses delayed with respect to S-In
In order to improve the signal representation of the oscilloscope, there is an additional output providing the signal D-In delayed by 3 clock pulses. This outputs C-Out and S-Out are TTL compatible.

*Operating elements:*
The upper toggle switch is used to select the code:
AMI : Alternate Mark Inversion
HDB3 : High Density Bipolar code of the order 3
The selected code is additionally indicated by one of two LED's.

The lower toggle switch selects the signal form:
NRZ : Non Return to Zero (signal is constant for the duration of the whole clock pulse)
RZ : Return to Zero (signal is 0 as long as the clock signal C-In is 0)

*Power supply:*
The training panel is supplied with ±15 V.

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

**736 91 AMI/HDB3 DECODER**

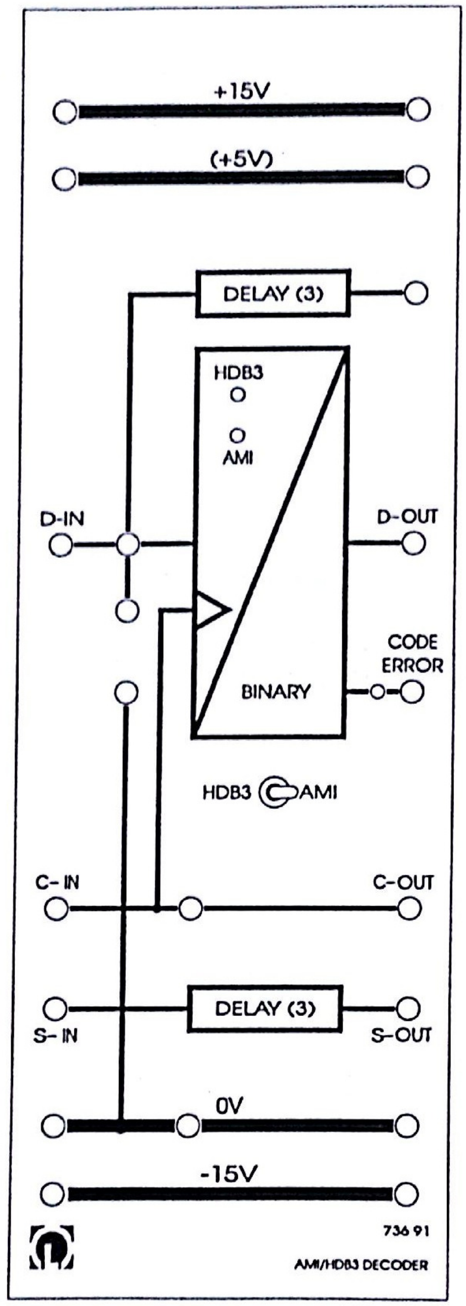



Fig. EQ-4 Front panel 736 91

| TPS 7.2.5.1 | Baseband Data Transmission Methods | EQ |
|---|---|---|

The AMI/HDB3 DECODER translates the channel codes created by the corresponding coder to unipolar NRZ signals.

*Inputs:*
D-In : serial data (coded, 0 V, ±5 V)
C-In : bit clock
S-In : synchronous signal, word clock
The inputs C-In and S-In are TTL compatible. All inputs are voltage proof up to ±42 V.

*Outputs:*
D-Out : decoded output signal, 3 clock pulses delayed with respect to D-In
C-Out : connected to C-In
S-Out : 3 clock pulses delayed with respect to S-In
CODE-ERROR : This output indicates violations of the code rules in the signal D-In. Additional indication with 1 LED.
There is an additional output providing the signal D-In delayed by 3 clock pulses. This output can be used when then signals are presented on an oscilloscope. All outputs are TTL compatible.

*Operating elements:*
The toggle switch is used to select the code:
AMI : Alternate Mark Inversion
HDB3 : High Density Bipolar code of order 3
The selected code is additionally indicated by one of two LED's.

The decoder automatically supports the signal forms:
NRZ : Non Return to Zero,
RZ : Return to Zero
without switchover being required.

*Power supply:*
The training panel is supplied with ±15 V.

# Experiment # 2: Serial Data Transmission

**2-1: Experiment objective**

Experiment 1 demonstrates the processes involved in parallel/serial conversion with the aid of a shift register and the relationships between the clock signal, the data signal and, the synchronous signal in term of time.

**2-2: Equipment list**

1 Power supply unit ±15 V/1 A

1 Training panel DATA SOURCE PARITY-GENERATOR (73693)

1 Oscilloscope

Diverse bridging plugs and cables

**2-3: Experiment set-up**

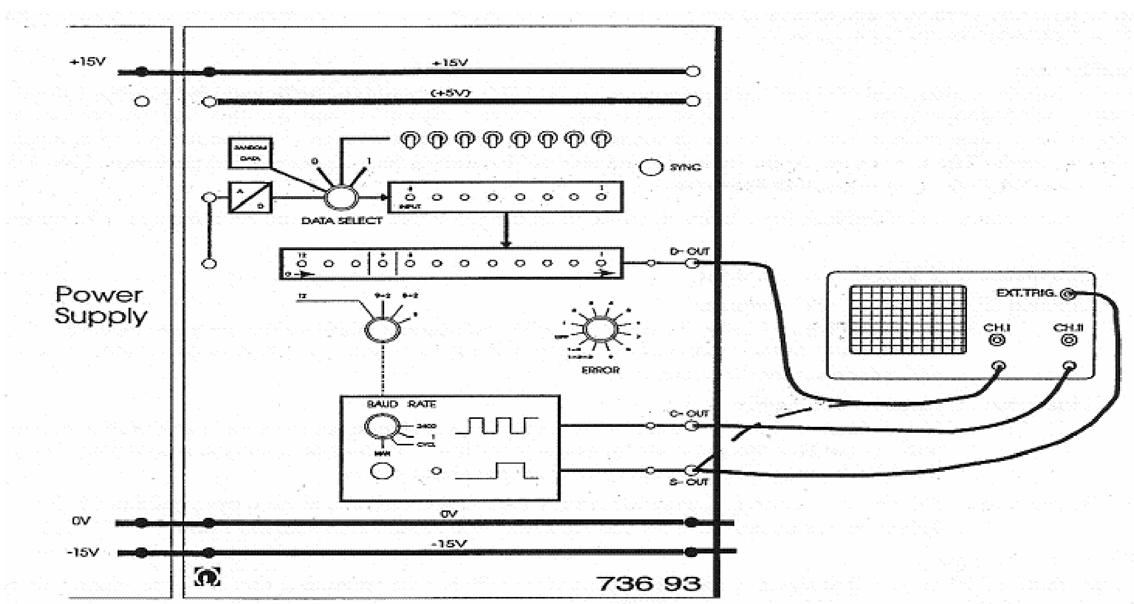



Figure 2-1: Experiment set-up for experiment 2

## 2-4: Experiment procedure

*2-4-1: Section a) Serial output with manual clock pulse*

- Set up the experiment as specified in Figure 2-1 and switch on the power supply. The oscilloscope is first used in the second part of the experiment.
- Set the switches on the training panel DATA SOURCE/PARITY GENERATOR to the following positions:

| | | |
|---|---|---|
| DATA SELECT | : | manual input |
| Input switches | : | any setting, but not all switches in the same position |
| Mode | : | 8 bit synchronous |
| BAUD RATE | : | MAN |
| ERROR | : | OFF |

- In the box labeled BAUD RATE on, the front panel, the LED on the right next to the switch begins blinking at approx. 2 Hz. This indicates that in this mode the clock signal is directly affected by the pushbutton. Please press this button a couple of times!
- When the pushbutton is in the rest state, the clock line (C-Out) is set to "0"(the LED next to the socket C-Out is not lit). When the pushbutton is activated C-Out, is set to "l" and the appropriate LED lights up.
- Press the pushbutton until the LED at the output S-Out (synchronous signal) lights up. Upon the next press of the button the LED at S-Out goes out again.
- Please press the pushbutton 5 more times. When the pushbutton is pressed a sixth time pay close attention to the shift register: When the pushbutton is released the contents of the input register (Input) is entered into the shift register. This is now the status before commencing to transmit it a data word. This same status can also be reached when the system is switched off briefly and then back on again.
- If you continue to press the pushbutton, please observe when the contents of the input of the shift register changes and when each new data bit is applied to the serial output D-Out.

*2-4-2: Section b) Serial output with 2400 bit/s*

- Connect the oscilloscope as specified in Figure 2-1 and carry out the settings according to the specifications in section titled "Introductory Notes". Set the Baud rate switch to the 2400 bit/s position.
- In the lower section of the oscilloscope screen the clock signal appears as a square wave with an amplitude of approx. 5V and a period duration of approx. 417µs (1/2400 s). The data signal is displayed in the upper section. Using the variable time-base deflection of the oscilloscope, the image can be set so that precisely 10 clock pulses are visible on the screen. Then a clock pulse is exactly one division unit in length. Please make a sketch of the screen in Figure 3-2w on the worksheet.
- Now connect channel 1 of the oscilloscope to the synchronous signal S-Out instead of: the data signal D-Out. Observe where the synchronous signal possesses the "l" level. Make a sketch of the screen in Figure 3-3w on the worksheet. Now reconnect channel 1 to the output D-Out.
- Alter the bit pattern to be transmitted using the 8 input switches. Observe how the data signal changes. Where for example does the bit 8 appear (toggle switch to the far left) on the screen? Please complete the sentence on the worksheet.

**2-5: Evaluation (worksheet)**
***Section a)***
Questions

1) How long is the synchronous signal?

____________________________________________

2) At which edge of the clock signal can the status of line data line or synchronous cline change?

____________________________________________

____________________________________________

***Section b)***

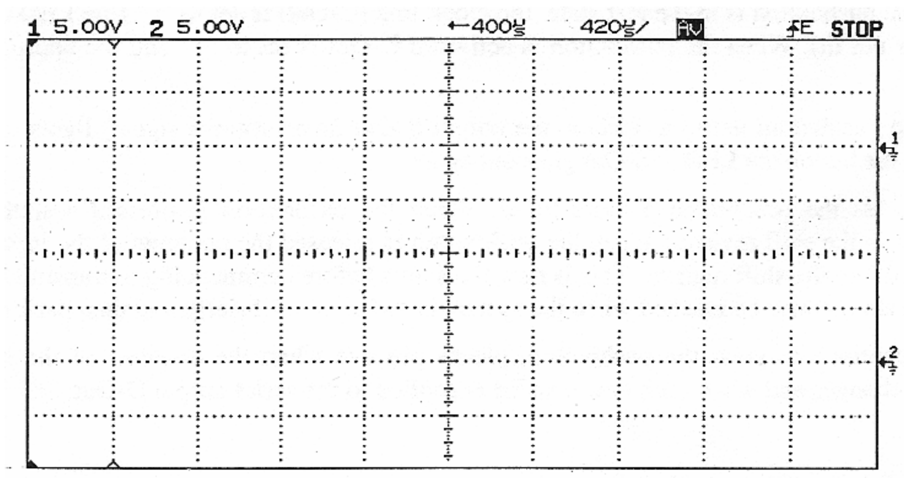



**Figure 2-2w:** Data signal (above) and clock signal (below) during synchronous transmission with 8 bit code word length

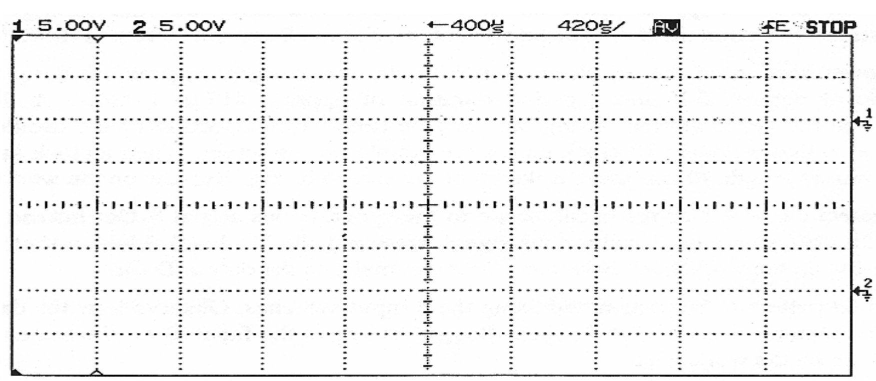



**Figure 2-3w:** Synchronous signal (above) and clock signal (below) during synchronous transmission with 8 bit code word length

Please complete the following sentence:

Bit 8 (toggle switch to the far left) appears on the oscilloscope screen to the far_________

# Experiment # 3: Synchronization Methods

### 3-1: Experiment objective

In Experiment 2, we investigate the various synchronization procedures used in serial transmission. Here, both synchronous as well as asynchronous methods are used in two different operating modes.

### 3-2: Equipment list

1 Power supply unit I15 V/1 A

1 Training panel DATASOURCE/PARITY GENERATOR(73693)

1 Training panel DISPLAY /PARITY CHECK INDICATOR (736 92)

1 Oscilloscope

Diverse bridging plugs and cables

### 3-3: Experiment set-up

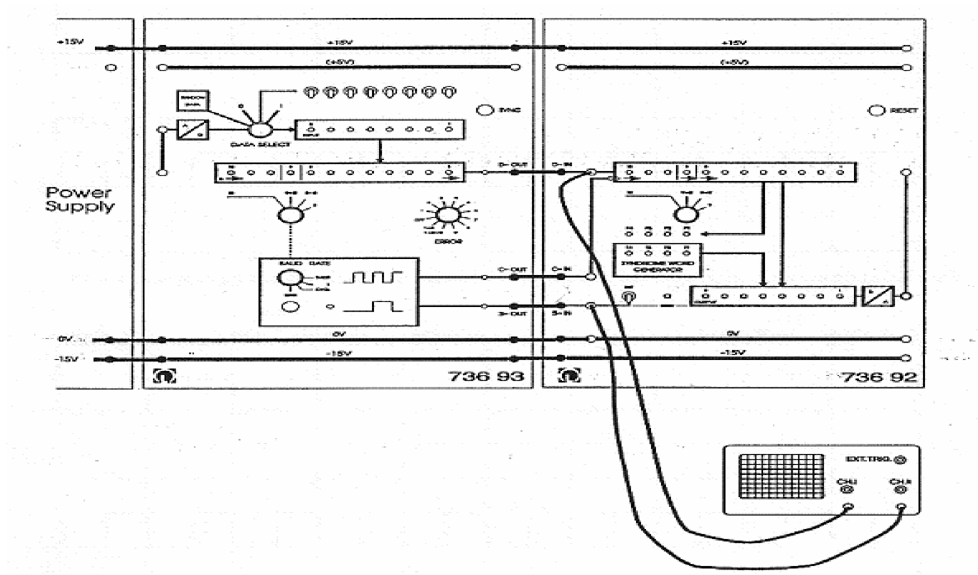



**Figure 3-1: Experiment set-up for experiment 3.**

## 3-4: Experiment procedure

### *3-4-1: Section a) Synchronous external*

Set up the experiment as specified in Fig. 3-1

- Set the switches on the training panel DATA SOURCE / PARITY GENERATOR to the following positions:

  DATA SELECT: manual input
  Input switches: any setting, but not all switches in the same position
  Mode: 8 bit synchronous
  BAUD DATE: Cycle (1 cycle with 1
  bit/s) ERROR: OFF

- Set the switches on the training panel DISPLAY/PARITY CHECK INDICATOR to the following positions:

  Mode: 8 bit synchronous
  Toggle switches: in the down position (external synchronization)

- Switch on the power supply. in the box labeled BAUD RATE, the LED lights up in order to indicate, that a new cycle (transmission of a code word) can be started by pressing the pushbutton located next to it. The LED labeled SYNC on the DISPLAYIPARITY CHECK INDICATOR panel remains dark (if not, please briefly press the RESET-button) thus indicating that the receiver is presently not synchronized.
- Start a transmission cycle and observe when the SYNC-LED lights up.
- Carry out several transmissions with, different bit patterns each time while at the same time observing the receiver's shift register.
- Also reset the receiver once in the middle of a transmission.
- Reset the baudrate switch to 2400 bit/so The oscilloscope is triggered to the synchronous signal (channel 2). Set the bit pattern to 00111011 and make a sketch of the oscilloscope screen in Figure 3-2w.

### *3-4-2: Section b) Synchronous internal*

- Set the SYNC switch on the DISPLAY / ARITY GHECK INDICATOR to INT (internal synchronization).The input S-In is no longer used now. Check this by removing the bridging plug at S- In: The synchronization is maintained until you press the

RESET pushbutton.

**Note**: The synchronized operating state is maintained in synchronous operating mode with internal synchronization until the receiver is forced to operate in the non-synchronized state.

- Briefly press the RESET button on the panel DATA SOURCE/PARITY GENERATOR when the receiver is in the non-synchronized state. Immediately after the pushbutton is released synchronization is restored.
- In order to be able to investigate these processes more closely, set the Baud rate switch back to the CYCLE position. Please set the bit pattern to 00010110. Begin transmission after resetting the receiver.
- After the synchronous character 00010110 has been received completely, the receiver immediately switches back to the synchronized state.
- Try to transmit other bit patterns as well (pressing the RESET button each time beforehand), apparently synchronization only functions with the synchronous character 00010110 (= 16 h = ASCII- character SYN, ASCII stands for American Standard Code for Information Interchange)
- Now set the bit pattern to 01011000. Set the DATA SELECT switch temporarily to the position I (all bits set to "1"). With the BAUD RATE switch set to MAN , now keep pressing the clock button until only "1" bits appear in the transmission and receiver registers. Then reset the DATA SELECT switch to "manual input". After the receiver has been reset activate the clock pushbutton several times while observing the receiver.
- The receiver is synchronized after 10 clock pulses, but this time not at the correct location; Transmit several different bit patterns, in order to detect the displacement between the input and the output data word.
- Now set the Baud rate switch to position 1 (l bit/s) Press the SYNC button and keep it pressed down until the receiver's shift register is filled with "1" bits when the SYNC

  Push button is pressed down the data line D-OUT is set to "l' while the clock pulse line C-OUT continues to supply clock pulses) If necessary reset the receiver After the SYNC button is released a synchronous character is automatically transmitted in the synchronous operating mode so that the receiver is now correctly synchronized.

*3-4-3: Section c) Asynchronous external*

- Set the switches on the training panel DATA SOURCFJPARITY GENERATOR to the following positions:

DATA SELECT: manual input

Input switches: any given position, but not all in the same position

Mode: 8+2 bit asynchronous

BAUD RATE: MAN (manually generated clock pulse)

ERROR: OFF

- Set the switches on the training panel DISPLAYIPARITYCHECK INDIGATOR to the following positions:

Mode: 8+2 bit asynchronous

Toggle switch: downwards (external synchronization)

- The SYNC switch is not related as it is in the case of asynchronous operation to word synchronization, but to bit synchronization. The input S-In is not used and thus remains unconnected.
- Briefly press the SYNC pushbutton on the training panel DATA SOURCE /PARITY GENERATOR in order to set the transmitter to a defined starting status.
- The data line D-Out is now set to the" 1" signal level (rest state). The start bit ('0" level) is now transmitted with the first clock pulse .This however does not yet alter the contents of the shift register. As the start bit does -not come from the shift register but is generated independently thereof. The receiver recognizes the start bit by the falling edge of the clock pulse (release of the clock pushbutton) and is thus synchronized.
- The 8 data bits are transmitted with the subsequent 8 clock cycles. But the receiver first recognizes the data as valid when the stop bit ("1" level) is also correctly recognized. Only then is the transfer carried out from the shift register into the data output register.
- Now reset the BAUD RATE switch back to 2400 bit/so Set the bit pattern to 00111011 and make a sketch of the oscilloscope screen in Figure 4-3w. Make a note of the individual bits of the data signal with START. STOP and 1...8.

*3-4-4: Section d) Asynchronous internal*

- Set the SYNC-switch on the training panel DISPLAYIPARITY CHECK

INDICATOR to INT (internal). This brings about the real asynchronous operating mode which is produced without external clock line. In order to verify-this please remove the connection to the clock input C-In. The start bit is used here not only for word but also bit synchronization. The receiver now uses its own clock generator which is newly synchronized with each start bit but which otherwise runs freely Naturally the nominal transmission rate must be known to c the receiver due to the fact that any deviation between the transmitting clock and the receiving clock may only be so large that the last data bit can still be reliably recognized .Thus this mode only functions at 2400, bit/s on the training panel DISPLAYIPARITY CHECK INDICATOR!

- Set bit pattern with one or several '0" bits Briefly press the RESET button of the receiver and closely observe whether the data is transmitted correctly Most likely a Joss of synchronization will occur in some cases.
- By pressing down the SYNC pushbutton the data line D Out is set to 1 The receiver (supposedly) recognizes a stop bit after several clock pulses at the latest and then waits for the next start bit. When the SYNC pushbutton is released this always causes a new data word'(and thus commencing with a start bit) to be started. This is how the correct synchronization is always reestablished.

## 3-5: Evaluation Experiment 2 (worksheet)

*3-5-1: Section a) Synchronous external Questions*

1) How is the start of a new code word recognized in synchronous operating mode with external synchronization?

______________________________________________

______________________________________________

2) When are the contents of the shift register recognized as a valid code word?

______________________________________________

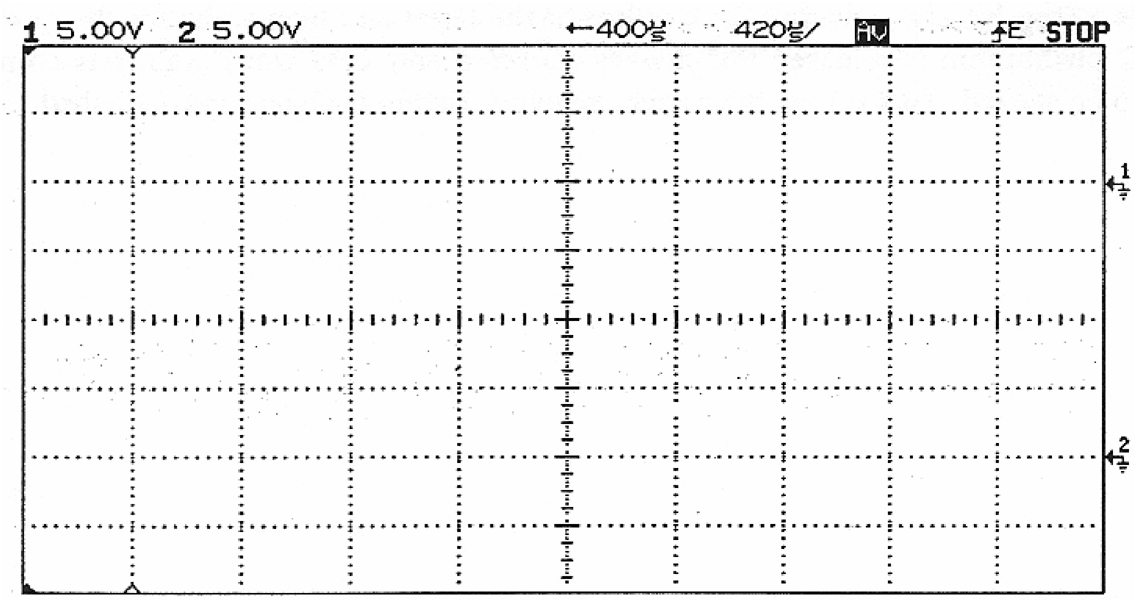



**Figure 3-2w:** Data signal (above) and synchronous signal (below)

*3-5-2: Section b) Synchronous internal*

Questions

1) Why did the receiver become synchronized at the wrong place in the experiment above?

______________________________________________

______________________________________________

2) What is the reason for filling the shift register of the receiver with "1" bits before synchronization (e.g. by keeping the data line set to "1")?

*3-5-3: Section c) Asynchronous external*

Question

1) How is the start of a new code word recognized in asynchronous operating mode?

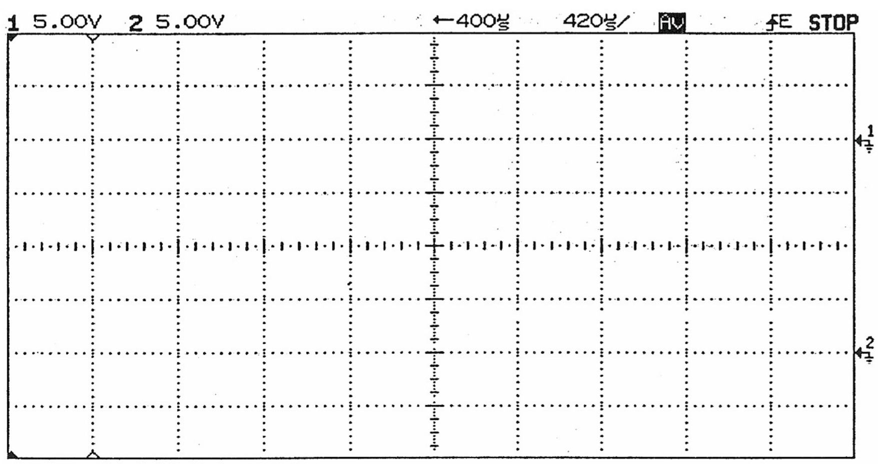



Figure 3-3w: Data signal (above) and synchronous signal (below)

*3-5-4: Section d) Asynchronous internal*

Question

1) Why is the stop bit needed?

# Experiment # 4: Bit Error Recognition and Correction

**4-1: Experiment objective**

In this experiment we investigate the generation of parity bits (even parity) as well as the recognition of one bit errors through parity comparison. In order to carry out recognition and correction of l-bit errors a transmission of 4 check bits is undertaken.

**4-2: Equipment list**

1 Power supply unit ±15 V /1 A

1 Training panel DATA SOURCE/PARITY-GENERATOR (736 93)

1 Training panel DISPLAY /PARITY CHECK INDICATOR (736 92)

1 Oscilloscope

Diverse bridging plugs and cables

**4-3: Experiment set-up**

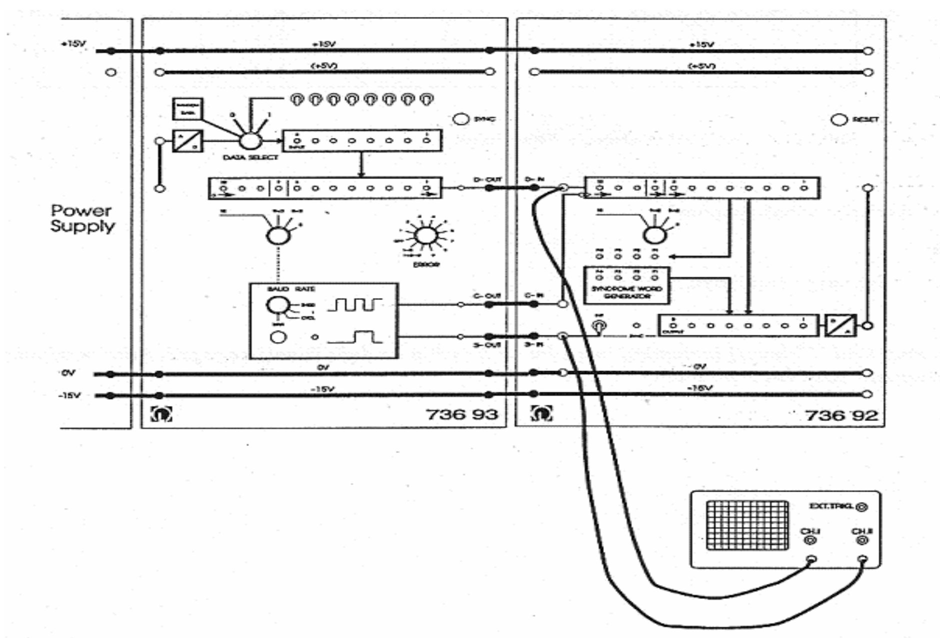



**Figure 4-1: Experiment set-up for experiment 4**

**4-4: Experiment procedure**

*4-4-1 Section a) Generation of parity bits and recognition of one bit errors through check bit comparison*

- Set the switches on the training panel DATASOURCE/PARITY GENERATOR to the following positions:

| | | |
|---|---|---|
| DATA SELECT | : | manual input |
| Input switches | : | all switches set to "0"(downwards) |
| Mode | : | 9 + 2 bit (asynchronous, 1 parity |
| bit) BAUD RATE | : | 2400 bit/s |
| ERROR | : | OFF |

- Also set the mode switches on the training panel DISPLAY /PARITY, CHECK INDICATOR to 9+2. The SYNC switch is set to external as this prevents the shift register from running through all the bits.
- Switch on all of the bits of the data word one after the other while at the same time observing bit 9 in the shift register. This bit is the so-called parity bit.
- With any given input bit pattern count the number of' "1" bits in the shift register. They must always be an even number because the so-called even parity is being used here.
- The transmitted bit pattern including the parity bit appears in the shift register of the training panel DISPLAY/PARITY CHECK INDICATOR. The parity bit is calculated anew based on the transmitted 8 data bits and indicated with the LED Pl. If the transmitted parity bit does not have the same value as the newly calculated one, a 1-bit transmission error has occurred. The comparison of the two parity bits is carried out via an XOR logic operation. The result of the logic operation is indicated in the box SYNDROME WORD GENERATOR by the LED Fl.
- The ERROR switch on the training panel DATA SOURCE/PARITY GENERATOR is used to generate specific bit errors, whereby the original bit is simply inverted. Here, both single bits 1…..9 as well as bit combinations 1+2+3 and 1+2 can be corrupted. The start
  bit in asynchronous transmission is not affected by the ERROR switch. If present, bit 9 is either at a stop bit or a parity bit.
- Set the ERROR switch to various positions and observe the effects on the receiver's

shift register (the transmitter's shift register remains unchanged). This error generation can also be observed on the screen of the oscilloscope.

- Set any given' single error and if desired, alter the data word to be transmitted. All single errors are recognized and displayed via the LED Fl.
- Set the ERROR switch" to the position 1+2 (far right stop). Double errors cannot be detected. This applies generally for every even number of errors.
- Set the ERROR switch to the position 1+2+3 (one setting back). In contrast triple errors can be recognized just like each uneven number of errors.
- In position 9 of the ERROR switch the parity bit itself is destroyed. The error is at so recognized.

*4-4-2 Section b) Generation of 4 check bits during* 12 *bit transmission*

*Recognition and correction of] bit errors, recognition of multiple errors*

- Set the switch on the training panel DATASOURCE/PARITY GENERATOR the following positions:

| | | |
|---|---|---|
| DATA SELECT | : | manual input |
| Input switches | : | all switches to "0" (downwards) |
| Mode | : | 12 bit (synchronous, 4 parity |
| bits) BAUD RATE | : | 2400 bit/s |
| ERROR | : | OFF |

- Also set the mode switch on the training panel DISPLAY/PARITY CHECK INDICATOR to the position 12. The SYNC switch is set to external, in order to avoid synchronization problems for faulty data bits.
- Set one data bit to "1" respectively (all other bits remain set to "0") and observe here which parity bits change. The bits 9...12 in the shift register correspond to the parity bits Pl…..P4. Fill out the Table 3-lw found on the worksheet.
- Answer question 1 on the worksheet (Section b) Check your results by setting the bit combination 10010010 and reading off the parity bits.
- The operating procedure in the receiver initially corresponds to that of the bit error recognition with only one parity bit, but with the exception here that now for every four bits transmitted there is a new calculation and comparison carried out. The

four resulting bits of the Comparison operation form the so- 'called syndrome word. They are displayed in the box SYNDROME WORD GENERATOR. As long as there is no transmission error all of the four bits of the syndrome word must have the va1ue (LEDs remain unlit)

- Answer question 2 on the worksheet (Section b) Test your result by setting different bit patterns and errors and observing the changes occurring to the syndrome word.
- Fill in the missing values in the Table 4-2w .The first section of the Table is arranged according to error positions the second part according to syndrome words.

## 4-5: Evaluation Experiment 3 (worksheet)

*Section a) Questions*

1) What value does the parity bit have for the input bit pattern 10010010?

______________________________________________

2) What happens when only the parity bit itself is faulty during transmission?

______________________________________________

3) Why can an even number of bit errors in-a code word not be recognized through the use of only one parity bit?

______________________________________________

Data bits

| | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 |
|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | |
| P2 | | | | | | | | |
| P3 | | | | | | | | |
| P4 | | | | | | | | |

**Table 4-1w:** Arrangement check bits/data bits

Questions

1) Which values do the 4 parity bits have in the data word 100100101 Use the Table 5-1w to answer this question?

PI: P2: P3: P4:

2) On what is the generated syndrome word dependent? Please mark the correct answer? ( ) on the transmitted bit pattern
( ) on the fault position

| Fault position | Syndrome word | | | | Syndrome word | | | | Fault position |
|---|---|---|---|---|---|---|---|---|---|
| | F4 | F3 | F2 | F1 | F4 | F3 | F2 | F1 | |
| No error | | | | | 0 | 0 | 0 | 0 | |
| D1 | | | | | 0 | 0 | 0 | 1 | |
| D2 | | | | | 0 | 0 | 1 | 0 | P2 |
| D3 | | | | | 0 | 0 | 1 | 1 | |
| D4 | | | | | 0 | 1 | 0 | 0 | P3 |
| D5 | | | | | 0 | 1 | 0 | 1 | |
| D6 | | | | | 0 | 1 | 1 | 0 | |
| D7 | | | | | 0 | 1 | 1 | 1 | |
| D8 | | | | | 1 | 0 | 0 | 0 | P4 |
| P1 | | | | | 1 | 0 | 0 | 1 | |
| P2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | |
| P3 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | |
| P4 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| Double error | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | Double error |
| " | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | " |
| " | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | " |

**Table 4-2w:** Arrangement fault position/syndrome word

# Experiment # 5: AMI Coding and Decoding

### 5-1: Experiment Objective

Here we investigate the structure and features of the AMI code (AMI = Alternate Mark Inversion). The difference between the RZ and NRZ formats are also demonstrated (RZ=Return to Zero, NRZ=Non Return to Zero.

### 5-2: List of Equipment

1 Power supply ±I5 VII A

1 Training panel DATA SOURCE/P ARITY GENERA TOR.(736 93)

1 Training panel DISPLAY/PARITY CHECK INDICATOR {736 92)

1 Training panel AMI/HDB3 CODER.(736 94)

1 Training panel AMI/HDB3DECODER (73691)

1 Oscilloscope

Diverse bridging plugs and cables

### 5-3: Experiment Set-up

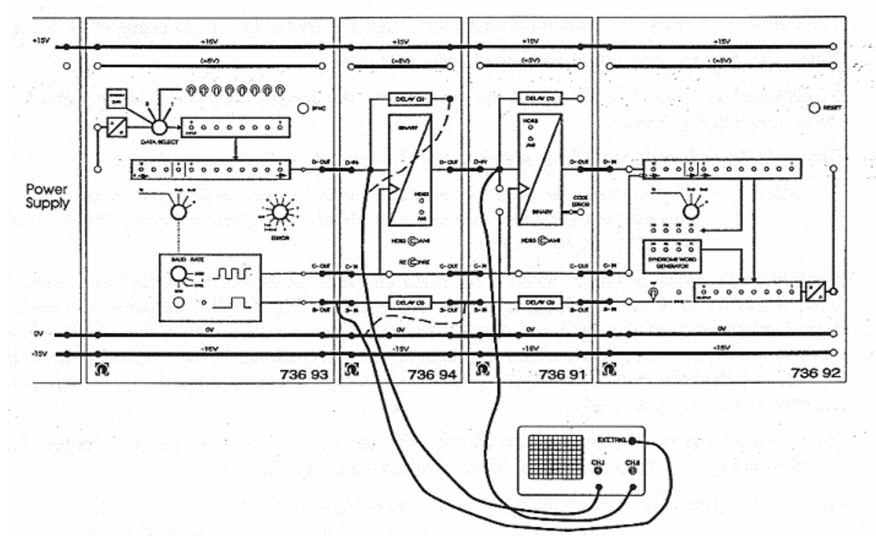



**Figure 5-1:** Experiment set-up for experiment 5.

## 5-4: Experiment Procedure

- Set up the experiment as specified in Figure 6-1, and switch on the power supply unit.
- Set the switches on the training panel DATA SOURCE/PARITY GENPRATOR to the following positions:

| | | |
|---|---|---|
| DATA SELECT | : | manual input |
| Input switches | : | all switches set to "0" |
| (downwards) Mode | : | 8 bit synchronous |
| BAUD RATE | : | 2400 bit/s |
| ERROR | : | OFF |

- Set the switch on the training panel DISPLAY/PARITY CHECK INDICATOR to the following positions:

| | | |
|---|---|---|
| Mode | : | 8 bit synchronous |
| Toggle switches | : | downwards (external synchronization) |

- The toggle switches on the training panels AMI/HDB3 coder and decoder are all pointing to the right (position AMI or NRZ).
- The oscilloscope is initially connected in accordance with the unbroken connection lines sketched in Figure 5-1.
- The oscillograph shows "0" level for both the input signal as well as the output signal of the coder.

  **Note:** "0" bits are transmitted unchanged in the AMI code.
- Switch the bits 1 and 2 of the data, input word to "1" (bit pattern 00000011) Make a sketch of the oscillograph in Figure 5-2w.

  **Note:** In the AMI code the "1" bits are transmitted alternately as "+1" and "-1" bits.

  Attention : In your sketch the coder output signal can be inverted with respect to the value specified in the solution due to the fact that the initial status (coding of the "I" as "+1" or "-1 ") is totally random.

- For technical reasons the output signal of the coder (and also the decoder) is shifted with respect to the input signal by three clock pulses, as can be deduced from Figure 5-2w. In order to recognize the logical relationship of the input .and output bits more

  clearly on the oscilloscope both devices are equipped with one output each where the input signal also shifted by three clock pulses - can be tapped. In order to be able to maintain the time relationship between the data and synchronous signal, the latter must also be delayed.

- Now connect the oscilloscope accordance with the connections outlined with dashed lines as specified in Figure 5-1. Make a sketch of the oscillograph in Figure 5-3w.
- Also try out different bit patterns and observe the output signal of the coder.
- Switch the lower toggle switch on the coder to RZ. Set to a bit pattern of 00011111 and make a sketch of the oscillograph in Figure 5-4w. Here too the output signal of your sketch may be inverted with respect to the example from the solution, see above.
- Instead of the points on the coder, display the corresponding ones on the decoder on the oscilloscope and check whether the original signal can be reproduced.

**5-5: Evaluation Experiment (Worksheet)**

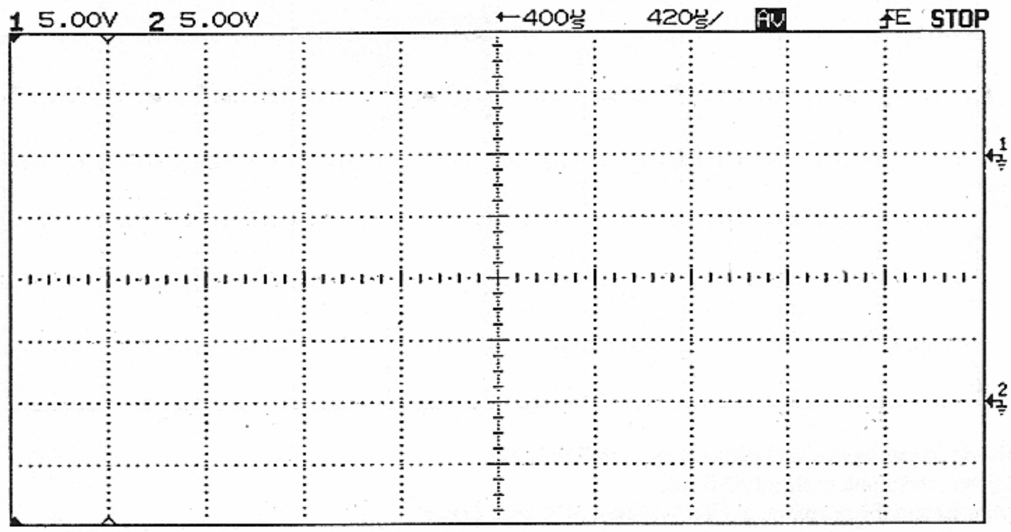



**Figure 5-2w:** above : coder input signal,
Below: coder output signal AMI NRZ
Triggering to the normal undelayed synchronous signal

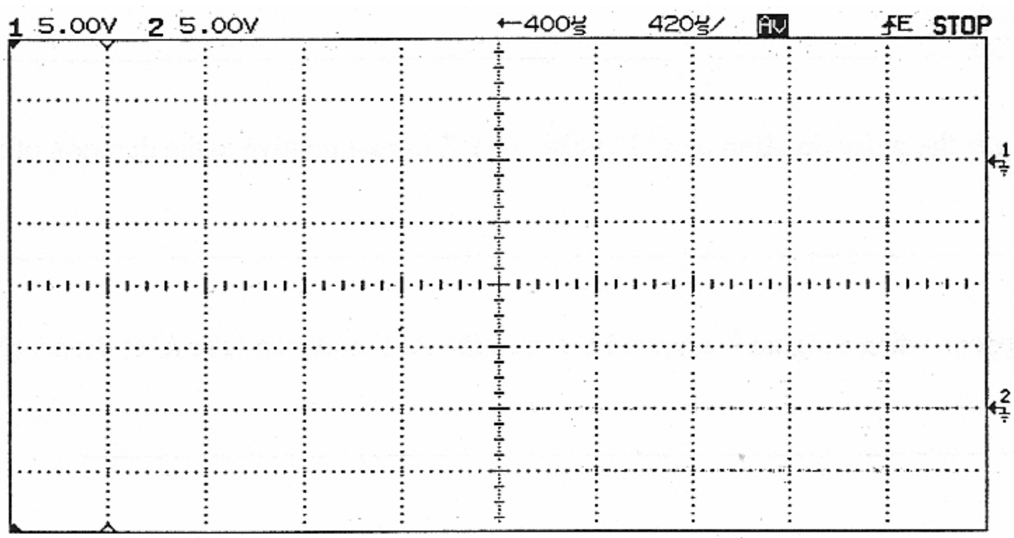



**Figure 5-3w:** Above: coder input signal (delayed by 3 clock pulses)
Below: coder output signal AMI NRZ
Triggering to the synchronous signal delayed by 3 clock pulses

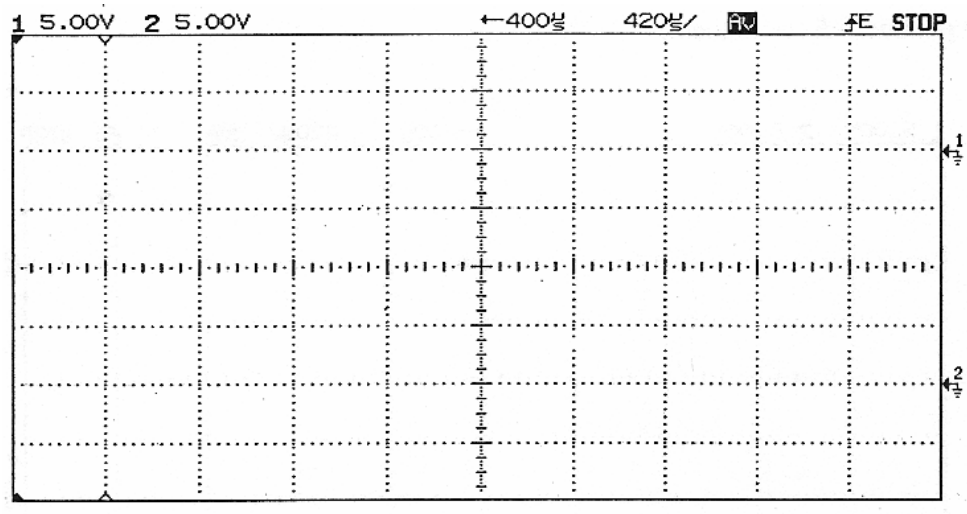



**Figure 5-4w:** Above: coder input signal (delayed by 3 clock pulses) Below: coder output signal AMI RZ, Triggering to the synchronous signal delayed by 3 clock pulses

**Questions**

1) Where on average is the DC component of an AMI signal located?

---

2) How long is the pulse duration of a it “1" pulse for RZ format relative to the duration of clock cycle?

---

3) What happens when a signal is inverted between the AMI coder and the AMI decoder?

---

# Experiment # 6: High Density Bipolar Code of Order 3 (HDB3 code)

**6-1: Experiment Objective**

Here, investigate the structure and features of theHDB3code (HDB = High Density Bipolar Code of Order 3).

**6-2: List of Equipment**

1 Power supply unit ±I5 VII A

1 Training panel. DATA SOURCE/PARITY GENERATOR *036* 93)

1 Training panel DATA DISPLAY /PARITYCHECK,INDICATOR(736 92)

1 Training panelAMI/HDB3 CODER (736 94)

1 Training panel AMI/HDB3 DECODER (73691)

1 Oscilloscope

Diverse bridging plugs and cables

**6-3: Experiment Set-up**

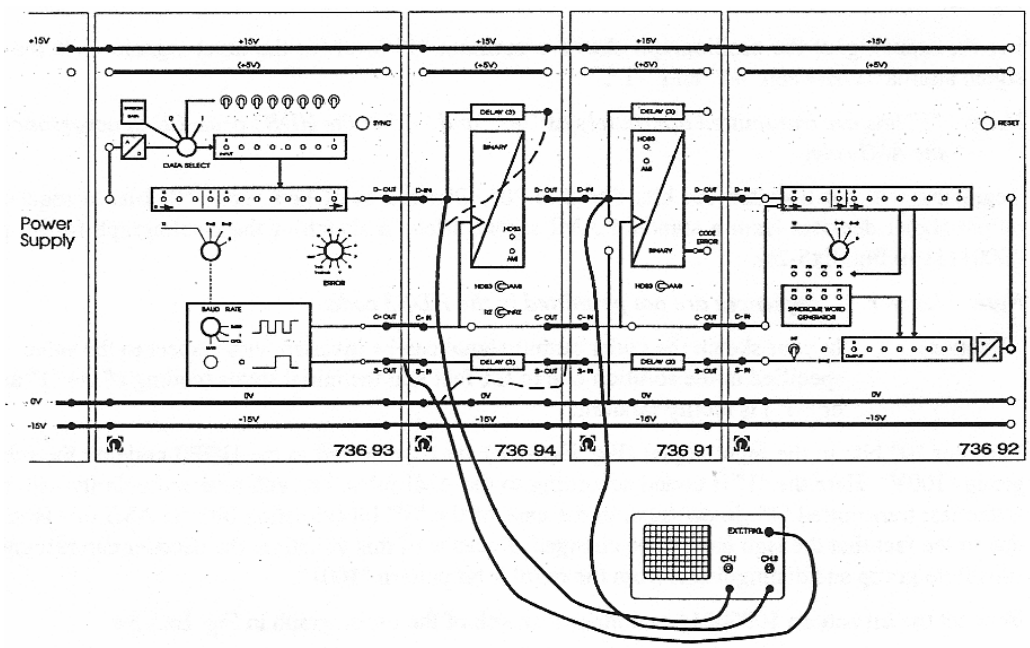



**Figure 6-1 Experiment set-up for experiment 6**

### 6-4: Experiment procedure

- Set up the experiment as specified in Figure 6-1 and switch on the power supply unit.

- Set the switches on the training panel DATA SOURCE/PARITY GENERATOR to the following positions:

| | | |
|---|---|---|
| DATA SELECT | : | manual input |
| Input switches | : | all switches set to "1" (upwards) |
| Mode | : | 8 bit synchronous BAUD RATE : 2400 bit/s |
| ERROR | : | OFF |

- Set the switch on the training panel DISPLAY/PARITY CHECK, INDICA TOR to the following positions:

| | | |
|---|---|---|
| Mode | : | 8 bit synchronous |
| Toggle switch | : | downwards (external synchronization) |

- The toggle switches on the training panels AMl/HDB3 code and decoder are set to HDB3 position and NRZ.

- The oscilloscope is initially connected in accordance with the dashed connections lines sketched in Figure6-1.
- For the input-signal the oscillograph shows a constant " 1" level for the input signal, while the output signal alternates between "+ 1" and "-1 ".

**Note**: If bits are transmitted alternately as "+1"and "-1"in theHDB3 code, i.e. in accordance with the AMI role.

- Beginning with bit 8 switch the bits 8 to 5 to "0". Observe as of how many "O" bits in sequence the output signal deviates from a standard AMI signal. Make a sketch of the oscillograph for bit pattern 00001111 in Figure 6-2w.

**Note**: Longer "0"sequences are not permitted in the HDB code.

Attention: In your sketch the coder output signal can be inverted with respect to the value specified in the solution due to the fact that the initial status (coding of the "1" as "+l" or "-1") is totally random.

- The four "0" bits in the input signal (Figure 7-2w) are represented in thecHDB3, code by the substitute group "100V. Here the" 1" is coded according to the AMI rules, i.e.

with reversed polarity with respect to the last transmitted "l ". In contrast, in the case of the "V" bit (violation bit) the AMI rule is violated, due .to the fact that the sign has not be changed. Because of this Violation, the decoder can recognize the substitute group and distinguish it from the regular bit pattern "1001 ".

- Now set thebitpattern10000011 and make a sketch of the oscillograph in Figure 6-3w.
- The first four "0" bits are represented by the substitute group, but this time by the "000V" combination. The fifth "0" bit is transmitted as "0" as is normal.
- Try out various other bit patterns and observe their effect on the coder output signal.
- Switch the lower toggle switch on the coder to RZ. Set to a bit pattern of 00000000 and sketch the oscillograph on Fig. 6-4w. Here too the output signal may be inverted with respect to the one provided in the sample solution, see above.
- The violation bit also alternates of its own accord (see Figure 6-4w). This ensures that the entire HDB3 signal is also on average AC only just like an AMI signal.
- Instead of the points on the coder, display the corresponding ones on the decoder on the oscilloscope and check whether the original signal can be reproduced.

**6-5: Evaluation Experiment (Worksheet)**

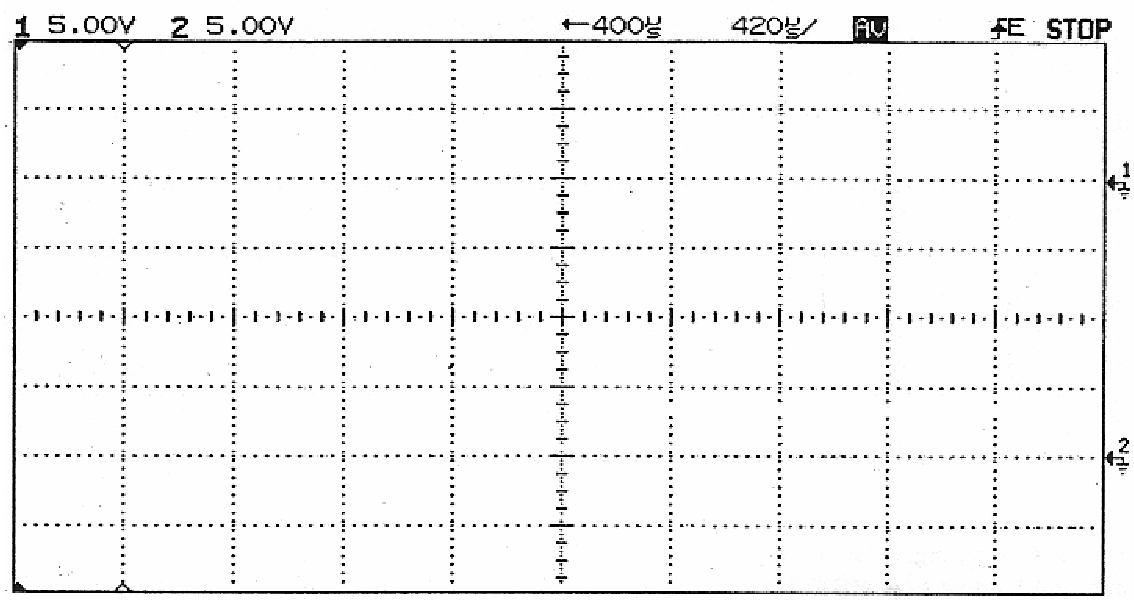



**Figure: 6-2w** Above: coder input signal (delayed by 3 clock pulses),
Below: coder output signal HDB3 NRZ substitute group 100V Triggering to the synchronous signal delayed by 3 clock cycles

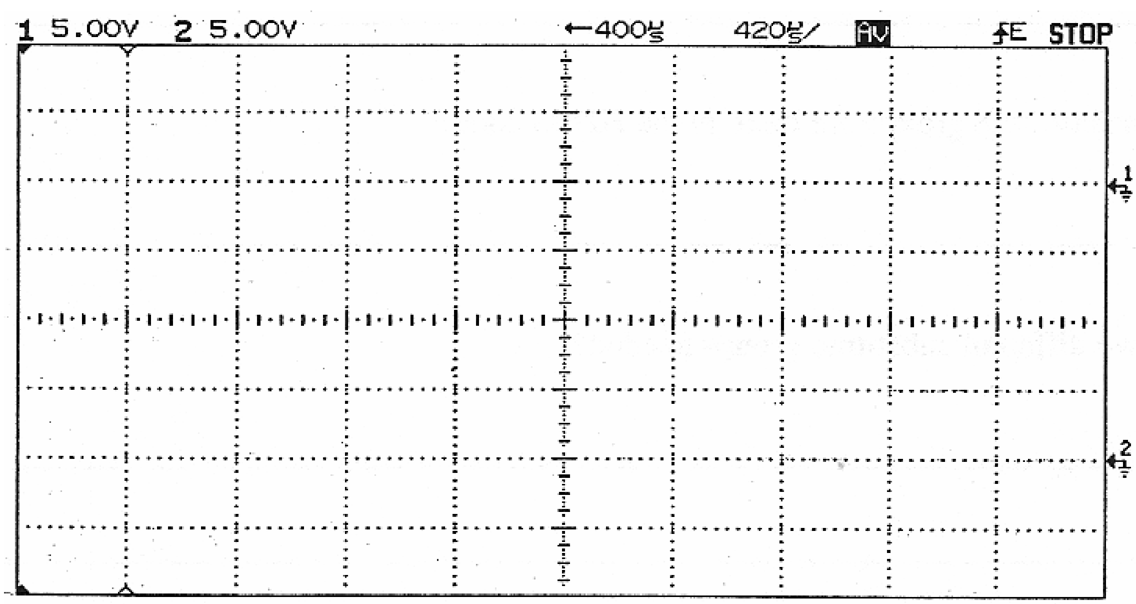



**Figure: 6-3w** Above: coder input signal (delayed by 3 clock pulses)
Below: coder output signal HDB3 NRZ, substitute group 000V Triggering to the synchronous signal delayed by 3 clock pulses

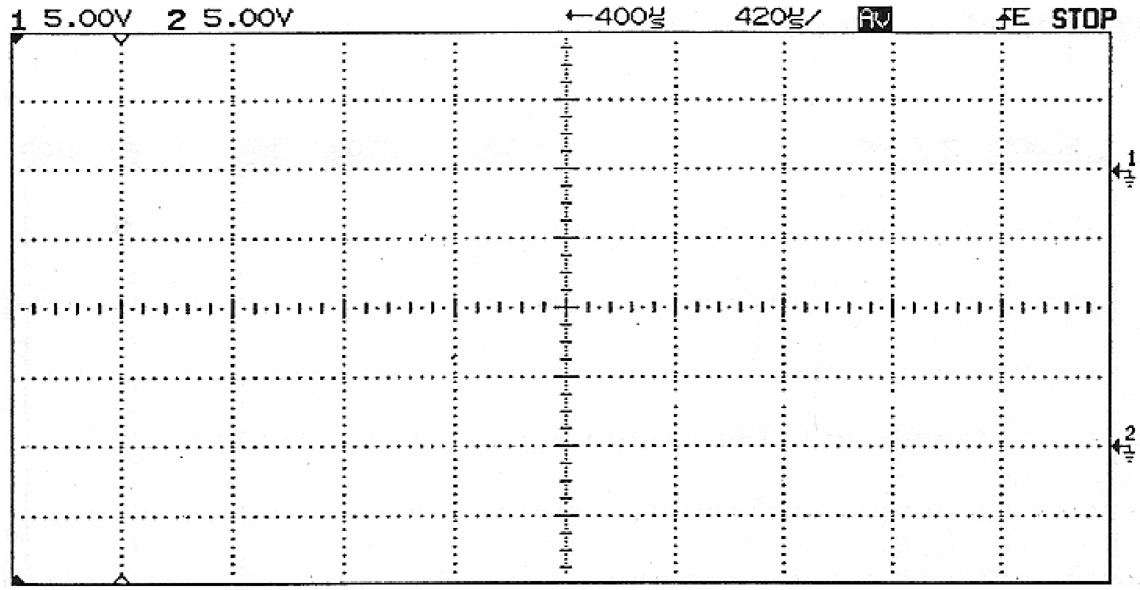



**Figure: 6-4w** Above: coder input signal (all bits 0)
Below: coder output signal HDB3 RZ,
Triggering to the synchronous signal delayed by 3 clock pulses

**Questions**

l) How many “0” bits in sequence are permitted in the output signal of the HDB3 code?

______________________________________________________________

2) Which two substitute groups are there in the HDB3 code?

______________________________________________________________

3) Why are two different substitute groups needed?

______________________________________________________________

4) Three "1" bits have, been transmitted since the last substitute group. Now four “0”bits follow in the input signal. Check off which bit combination is used to substitute for them!

( ) 100V ( ) 100V

5) Is the inversion of the signal without any negative effects in the HDB3 code as well?

______________________________________________________________

# Experiment # 7: Code Error (Caused by Coding with Incorrect Format)

**7-1: Experiment objective**

Here we investigate under what conditions code errors can be detected, When the coder and decoder are not operating with the same format. For this the output CODE ERROR of the AMI/HDB3 decoder is used.

**7-2: List of Equipment**

1 Power supply unit ±15 V/1 A

1 Training panel. DATA SOURCE/PARITY GENERATOR (73693)

1 Training panel DATADISPLAYIPARITY CHECK INDICATOR (736 92)

1 Training panel AMI/HDB3COPER(736 94)

1 Training panel AMI/HDB3 DEC~DER (736 91)

1 Oscilloscope

Diverse bridging plugs and cables

**7-3: Experiment set-up**

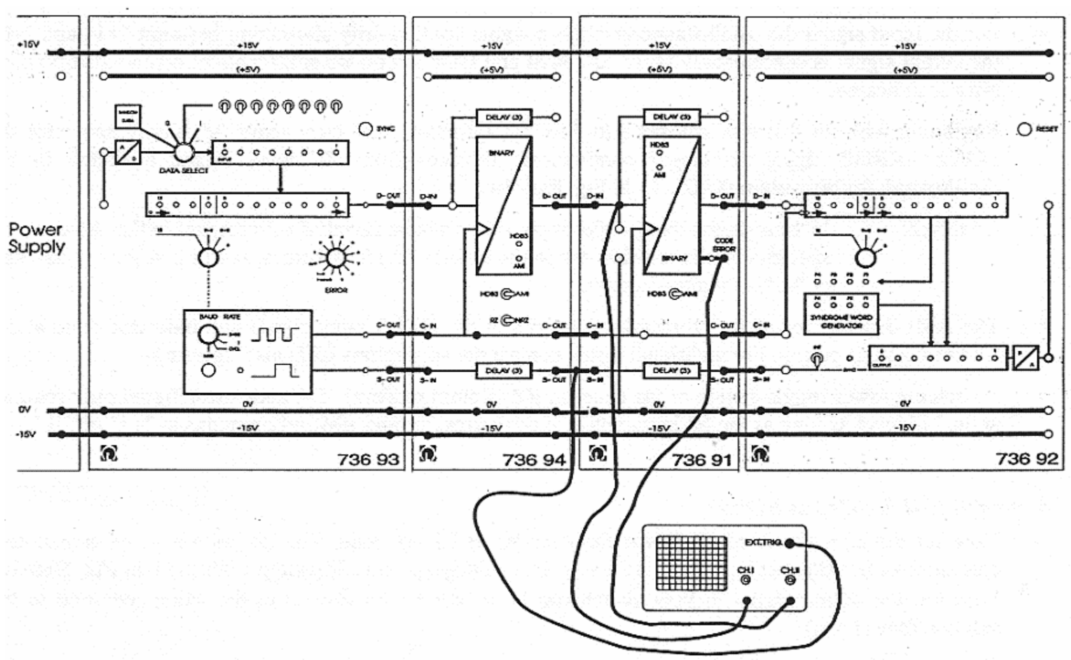



**Figure 7-1:** Experiment set-up for experiment 7.

## 7-4: Experiment procedure

*7-4-1: Section a) HDB3-code decoded as AMI*

- Set up the experiment as specified in Figure 7-1 and switch on the power supply unit.
- Set the switches on the training panel DATA SOURCE / PARITY GENERATOR to the following positions:

| | |
|---|---|
| DATA SELECT: | manual input |
| Input switches : | all switches set to "1" (upwards) Mode : S bit synchronous |
| BAUD RATE : | 2400 bits |
| ERROR: | OFF |

- Set the switch, on the training panel DISPLAY/PARITY CHECK INDICATOR to the following positions:

| | |
|---|---|
| Mode : | 8 bit synchronous |
| Toggle switch : | downwards (external synchronization) |

- The toggle switches on the training panels AMI/HDB3 Coder are set to HDB3 and NRZ position, whereas the toggle switch on the AMI/HDB3 DECODER is set to AMI!
- For the input signal the oscilloscope displays a signal continuously alternating between "+1" and "-1", the output signal is continuously set to "0". AMI and HDB3 code are still identical here so that no code error is indicated.
- Beginning with bit switch the bits 8 to 5 to "0" Observe as of how many "0" bits in sequence the CODE ERROR signal no longer continuously deviates . from "0" level Make a sketch of the oscillograph for bit pattern 00001111 in Figure 7-2w.

  Attention : In your sketch the decoder input signal can be inverted with respect to the value specified in the solution due to the fact that the initial status (coding of the "l" as "+1" or "-1 ") is totally random.
- The AMI decoder recognizes the code violations in the HDB3 input signal and indicates them at the CODE ERROR output. For additional visual control the appropriate LED also lights up.
- Switch the lower toggle switch of the coder to RZ (Return to Zero ). The code error signal only remains at the "l" level as long as the bit in violation of the corresponding AMI rule remains at "+ 1" or "-1".

*7-4-2: Section b) AMI decoded as HDB3*

- Now set the coder to AMI-NRZ and the decoder to HDB3 code. The bit pattern to be transmitted continues to be 00001111. Make a sketch of the oscillograph for bit pattern 00001111 in Figure 7-3w.
- The HDB3 decoder signals at the code error output when more than three "0" bits in the input signal follow in sequence, as this indicates that the code rules for the HDB3 code have been violated.
- Also try other bit patterns and observe the CODE ERROR signal.
- Switch the lower toggle switch of the coder to RZ (Return to Zero). Now the CODE ERROR signal remains at the "1" level for the entire duration of the bit, because here not a "1" signal, but an incorrect "0" signal is the cause of the error.

**7-5: Evaluation Experiment**

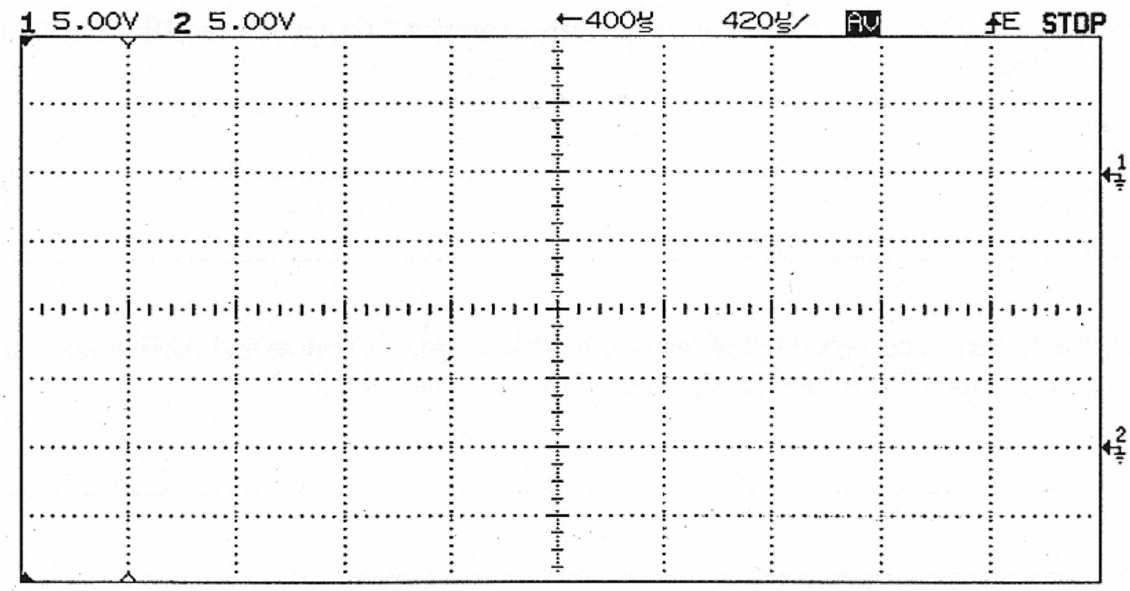



**Figure 7-2w:** above : Decoder input signal (HDB3/ NRZ)
Below : CODE ERROR output (for AMI
decoding) Triggering to the synchronous signal

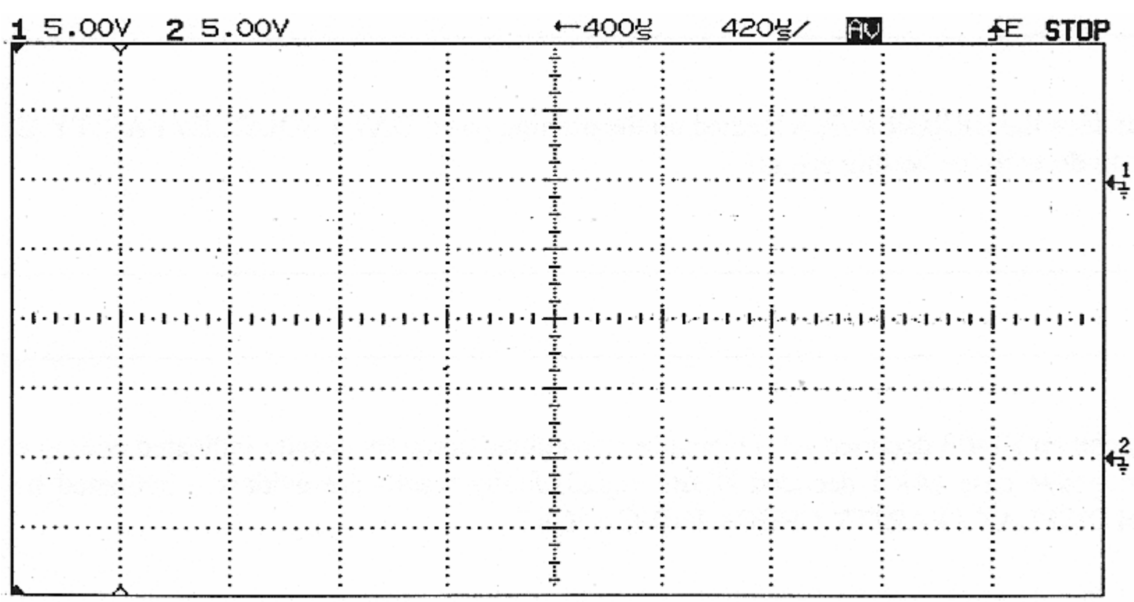



**Figure 7-3w:** above : Decoder input signal (AMI *I* NRZ)
below: CODE ERROR output (for HOB3
decoding) Triggering to the synchronous signal

Questions

1) Every how many bit leads to a coding errorwhei1aconstant "0"signal coded HDB3 is transmitted and AMI-decoded?

______________________________________________

2) When the bit 'sequence 11000000 of the original NRZ signal is transmitted AMI-coded what does the bit sequence of the CODE *ERROR* signal in HDB3 decoding look like?

______________________________________________

3) In contrast to most of the other experiments why is the original (undelayed) input signal displayed on the oscilloscope?

______________________________________________

______________________________________________

______________________________________________

4) What does the ERROR switch located on the training panel DATA SOURCE / PARITY GENERATOR have to do with the coding errors?

______________________________________________

______________________________________________

______________________________________________

5) Why can anHDB3decoded AMI signal be transmitted correctly despite indicated coding error ,while in the opposite case (AMI decoded HDB3 signal) faulty results are evidently indicated by the training panel DISPLAY *I* PARITY CHECK INDICATOR?

______________________________________________

______________________________________________

______________________________________________

______________________________________________

# Experiment # 8: Introduction to MATLAB

**8.1: Introduction:**

MATLAB is commercial “MATrix LABoratory” package by Mathworks, which operates as an interactive programming language environment with graphical output.

In engineering MATLAB is among popular programming languages, due to its interactive interface, reliable algorithmic foundation, fully extensible environment and computational speed.

## 8.2. MATLAB basic operations:

### 8.2.1. Entering and running MATLAB:

On system running Microsoft windows double click on MATLAB icon to launch MATLAB, a command window will appear with a prompt “>>” you are now in MATLAB.

### 8.2.2. Leaving MATLAB:

A MATLAB session is simply terminated by typing the following in MATLAB prompt.

>> Quit or

>> exit

### 8.2.3. Online Help:

Online help is available from the MATLAB prompt both general and scientific commands. We can type the following in command prompt for help.

>>help

>>help demo

## 8.3. Desktop tools in MATLAB:

### 8.3.1. Command window:

In command window we type commands, view program variables; we can clear all the variables by typing Clear in the command Window.

### 8.3.2. Command History:

We can view the the past commands and save a whole session by using “diary”.

## 8.4. Variables:

MATLAB is a case sensitive that is ‘a’ is not the same as ‘A’.

MATLAB has a built in variables like ‘**pi’,’ans’** and **‘eps’**.The variable **‘ans’** will keep the track of the last output which was not assigned to another variable**.**

### 8.4.1. Variable assignment:

The equality sign is used to assign values to the variables.

For example

```
>>x=3

>>y=x^2
```

Output can be suppressed by putting a semicolon to the command lines.

```
>>x=3;

>>y=x^2;
```

### 8.4.2. Active variables:

Who is the active variable in MATLAB.

### 8.4.3. Removing variables:

Clear x and clear are used in MATLAB as removing the variables.

### 8.4.4. Saving and restoring variables:

Save filename, load file name are used for saving and restoring.

### 8.4.5. Variable arithmetic’s:

Operators Precedence:

Let we have an equation like:

2+3*4^2

The following is the operator precedence in MATLAB:

"^, * or /, + or -."

So the ans of above equation will be:

>>ans= 50

### 8.4.6. Command line editing:

In MATLAB the arrows keys allows "command line editing".

## 8.5. Built in MATLAB functions:

**Table 8-1** shows the basic built in MATLAB functions.

| function | Meaning | Example |
|---|---|---|
| Sin | Sine | Sin(pi)=0.0 |
| Cos | Cosine | Cos(pi)=1.0 |
| Tang | Tangent | Tan(pi/4)=1.0 |
| Exp | Exponential | Exp(1.0)=2.7183 |
| Log | Natural log | Log(2.7183)=1.0 |

**Table 8-1**

## 8.6. Matrices:

The matrices can be created in MATLAB by the following commands:

>>A=[1 2 3;3 4 5;3 4 7]; or

>>A=[1,2,3;4,5,6;3,4,7]

The matrix element located in the ith row and jth column of A can be excess by usual way:

>>A (1, 2), A (2, 3)

Matrices can easily be modified by:

>>A (2, 3) =10;

### 8.6.1. Built in matrix functions:

**Table 8-2** shows the built in matrix functions:

| Function | Description |
|---|---|
| Diag | Return diagonal M.E as a vector |
| Eye | Identity matrix |
| Magic | Magic square |
| Ones | Matrix of ones |
| Rand | Randomly generated matrix |
| Zeros | Matrix of zeros |

**Table 8-2**

### 8.6.2. Matrix operations:

+ 'Addition'

− 'Subtraction'

/ 'Division'

, 'Transpose'

^ 'Power'

* 'Multiplication'

./ 'element by element division.

.^ 'element by element power.

### 8.7. Round floating point number to integers:

Round, ceilling, floor, etc can easily done in MATLAB on

Matrices. The function used for these are the following:

Round (f); ceil (f); floor (f)

**8.8. Relational and logical operators:**

**Table 8-3** shows relational and logical operators which are

Supported by MATLAB:

| Operators | Description |
|---|---|
| == | Equal |
| ~= | Not equal |
| < | Less then |
| <= | Less then or equal to |
| > | Greater then |
| >= | Greater then or equal to |
| & | And |
| \| | Or |
| ~ | Not |

**Table 8-3**

**8.9. Branching construct:**

The following branching construct are supported by MATLAB:

**8.9.1. If — end constructs:**

If <condition>,

<Program>

End

**8.9.2. If – else if end construct:**

If <coondition1>

<program1>

Else if <program2>

End

**8.9.3. If – else end construct:**

If <condition1>

<program1>

Else

<program2>

End

**8.9.4. Looping construct:**

We can use for loop, while loop and nested for loop in MATLAB just like c or c++.

**8.10. MATLAB M-file:**

MATLAB commands can be run from one file without having to enter each command on command prompt. In order to use them in a program they are save first. For this purpose program should be written in the edititor.

**8.11. MATLAB graphics:**

In MATLAB we can use multiple graphs, multiple plots, three dimensional plots etc.

The following **figure 8-1** shows multiple graphs:

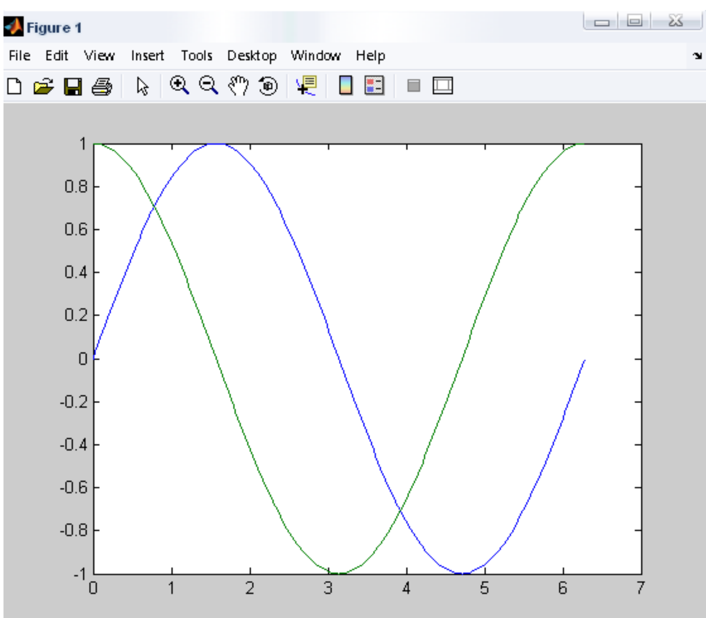



**Figure 8-1**

**Figure 8-2** shows the multiple plots on a single graph.

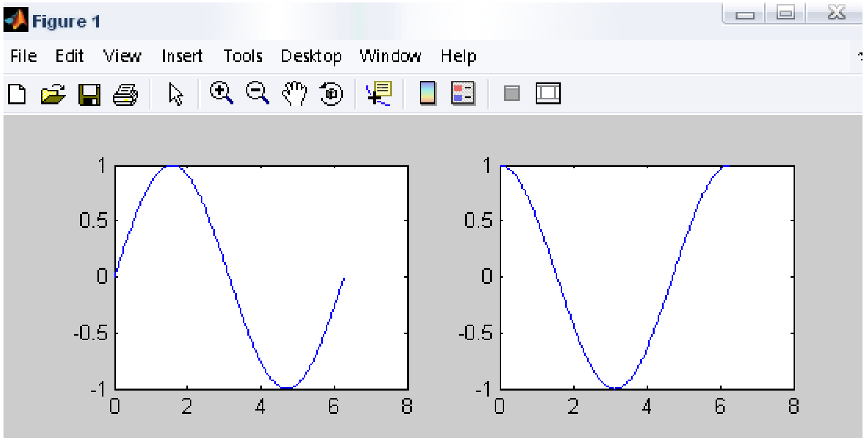



**Figure 8-2**

**Figure 8-3** shows multiple graphs on a single graph

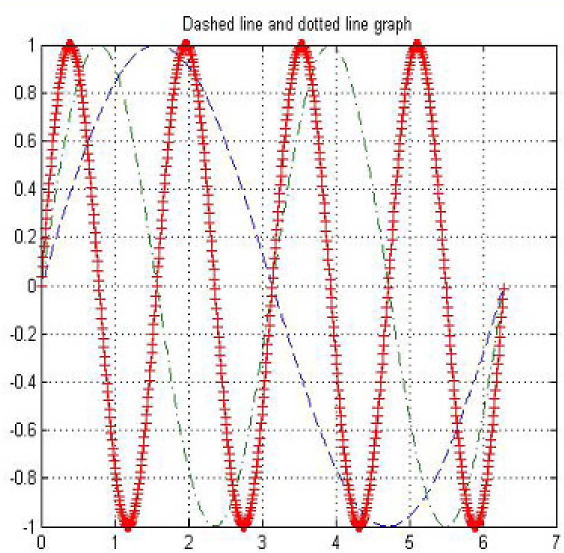



**Figure 8-3**

### 8.11.1. Graph functions:

**Table 8-4** describes different graph functions and there description.

| Functions | Description |
|---|---|
| Plot(x,y) | Linear plot |
| Plot(x,y1,x,y2) | Multiple plots on same graph |
| Mesh(z) | 3-D graph |
| Stem(x) | Discrete plot |
| X-label('x axis label') | Add x-axis label |
| Y-label('y axis label') | Add y-axis label |
| Title('title of plot') | Title of the plot |
| Subplot(m,n,p) | Divide figure of window |
| Hold | Hold current graph in the figure |
| Zoom | Allow zoom in/out using mouse |
| Pause | Wait for user response |

**Table 8-4**

# Experiment # 9: Study the effects of transmission impairments on a digital signal

## OBJECTIVES OF THE LAB

----------------------------------------------------------------------------------------------------

In this lab, we will cover the following topics:

- How to generate a simple digital signal in MATLAB?
- Effect of thermal noise on the signal generated
- Effect of delay distortion on a digital signal
- Effect of attenuation on a digital

----------------------------------------------------------------------------------------------------

## 9.1 DIGITAL SIGNAL GENERATION

A digital signal is one in which the signal intensity maintains a constant level for some period of time and then changes to another constant level. However, with all its sudden changes, digital signal is a composite signal having infinite number of frequencies.

Digital signal can be both periodic and aperiodic. Periodic digital signal can be generated using Fourier Synthesis of Square Wave, while aperiodic digital signal can be represented through Fourier Transform.

### 9.1.1 Synthesis of Square Wave

The Square Wave for one cycle can be represented mathematically as:

$$x(t) = \begin{cases} 1 & ,0 \le t < \frac{T}{2} \\ -1 & ,\frac{T}{2} \le t < T \end{cases}$$

The Complex Amplitude is given by:

$$x(t) = \begin{cases} \frac{4}{j\pi k} & ,k : odd \\ 0 & ,k : even \end{cases}$$

For a fundamental frequency of f=1/T=10Hz, only the odd frequencies harmonics i.e. ±10, ±30,

±50 Hz etc. make up the spectrum.

**-------------------------------------------TASK 01--------------------------------------**

Design a periodic digital signal having 25 harmonics with fundamental frequency of ½ from the following signal:

$$x(t) = \frac{4}{\pi k}\sin(2\pi f t)$$

**-------------------------------------------TASK 02-------------------------------------**

Using MATLAB built-in function "square", generate a square wave of 20 Hz. For details about mentioned function, type: help square in MATLAB command window.

## 9.2 TRANSMISSION IMPAIRMENTS

When a signal is transmitted through a transmission medium, various impairments can corrupt or distort the received signal. These impairments include thermal noise, attenuation, delay distortion, intermodulation noise, impulse noise, etc.

### 9.2.1 Effect of Thermal or White Gaussian Noise on a signal

Thermal noise is due to thermal agitation of electrons. It is present in all electronic devices and transmission media. It is uniformly distributed across the bandwidth used in communication system and hence is often referred as white noise. It cannot be eliminated completely, thus places an upper bound on communications system performance.

To generate white noise in MATLAB, following commands are used:

==================================================

| | |
|---|---|
| **wgn(m, n, p)** | Generates an m-by-n matrix of white noise with output power 'p' specified in dBW by default. |
| **wgn(m, n, p, output type)** | Same as above except output type specifies whether the noise is 'real' or 'complex' |
| **wgn(m, n, p, imp, output type)** | Same as above but now power is generated across the load whose impedance is expressed in ohms |
| **awgn(x, SNR)** | Adds white noise in a vector x; where SNR is signal-to-noise ratio per sample in dB |
| **awgn(x, SNR, 'measured')** | Same as above except that awgn measures the power of x before adding noise |

==================================================

## Example – Generating Real White Noise

To generate a row vector of length 100 containing real white Gaussian noise of power 0 dBW, following command is used:

```
>> y1 = wgn(1, 100, 0);
```

## Example – Generating Complex White Noise

To generate a row vector of length 100 containing complex white Gaussian noise, where each component has a noise power of 0 dBW, type in MATLAB following command

```
>> y2 = wgn(1, 100, 0, 'complex');
```

**--------------------------------------------TASK 03---------------------------------------**

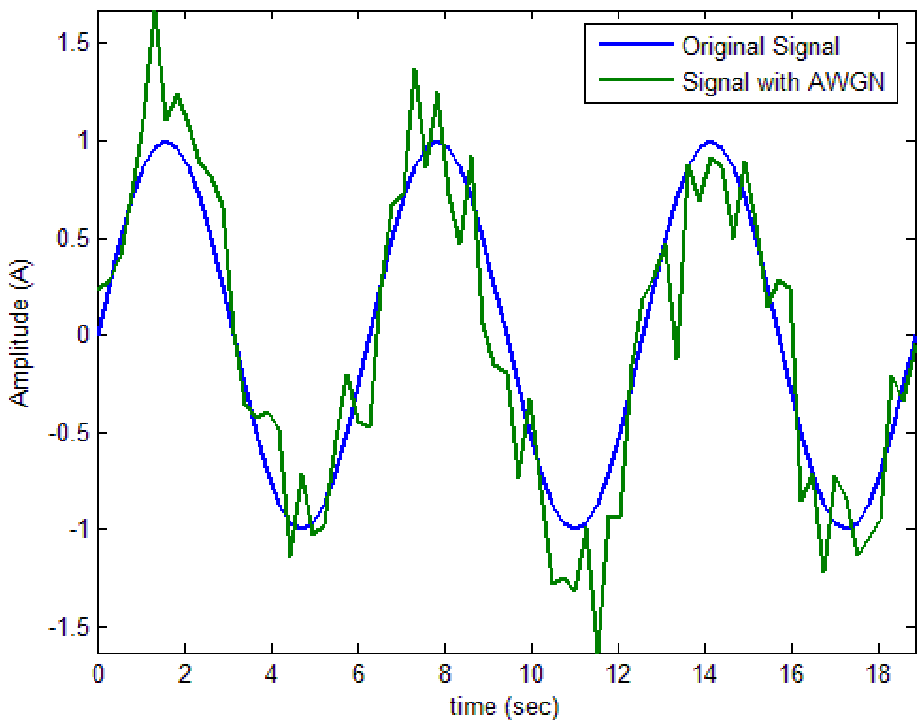
Generate a column vector of length 512 containing real noise of power 3 dBW across a 50 ohm load.

## Example – Add White Noise in a Sinusoidal Signal

Following piece of code adds white Gaussian noise to a sinusoid of 3 Hz such that each sample has SNR of 10 db. It then plots the original and noisy signals.

```
clc
clear all

t = 0:pi/12:6*pi;

x = sin(t);                          % create sinusoidal signal

y = awgn(x, 10);                     % add white Gaussian noise
plot(t, x, t, y, 'lineWidth', 2);    % plot both signals
xlabel('time (sec)');
ylabel('Amplitude (A) ');

legend('Original Signal ', 'Signal with AWGN ');
```

axis tight;




**--------------------------------------TASK 04------------------------------------------**

Add measured white noise of 15 db to square wave generated in task 01. Plot the original and noisy signals on the same graph. Comment on the result.

### 9.2.2 Effect of Delay Distortion on a signal

Delay distortion occurs because propagation velocity of a signal varies with frequency. Considering a band-limited signal, velocity is highest near the center frequency and falls off toward the two edges of the band. As a result, various frequency components of a signal arrive at the receiver at different time, resulting in phase shifts between the different frequencies.

This effect is referred to as delay distortion because the received signal is distorted due to varying delays experienced at its component frequencies.

**--------------------------------------TASK 05-------------------------------------------**

Given the fundamental frequency of 0.375 Hz, effects of delay distortion can be observed by doing following steps:

a) Take three harmonics (including fundamental) and generate a square wave.
b) Now add the phase of 0, 45, & -30 to 1st, 2nd, & 3rd harmonic respectively. Regenerate the square using these harmonics.
c) Plot the two waves. Also, comment on the result.

### 9.2.3 Effect of Attenuation on a signal

Signal strength falls off with the distance it travels on any transmission medium. This property is generally termed as attenuation. For guided media, it is exponential and expressed in units of decibels/distance. Mathematically, decibel is a measure of the ratio between two signal levels.

Attenuation i.e. power loss can be calculated as

$$L_{dB} = -10 \log_{10}\left(\frac{P_{out}}{P_{in}}\right) = 10 \log_{10}\left(\frac{P_{in}}{P_{out}}\right)$$

Example – Measuring Amount of Attenuation in a Sinusoidal Signal

Suppose a sinusoid travels through a transmission medium and its power is reduced to one-half. This means that $P_{out} = ½ P_{in}$. Following MATLAB code determines the attenuation or power loss encountered by the transmitted signal:

```
clc
clear all
t = 0:pi/12:6*pi;
x_t = sin(t);                    % create sinusoidal signal for transmission
Ex_t = sum(x_t .* conj(x_t));    % measure transmitted signal energy
P_in = Ex_t/length(t);           % measure transmitted signal power
x_r = 0.5*sin(t);                % reduce sinusoid strength to ½
Ex_r = sum(x_r .* conj(x_r));    % measure received signal energy
P_out = Ex_r/length(t);          % measure received signal power

L = -10*log10(P_out/P_in)        % calculate attenuation or power loss
```

SAMPLE

RUN

```
L = 6.0206
```

**------------------------------------------TASK 06------------------------------------------**

Write MATLAB code that calculates amount of attenuation if signal power (of sinusoid created in above example) is reduced to 1/3 of original signal's power.

# Experiment # 10: Analog Signal Generation using Digital Data in MATLAB

## OBJECTIVES OF THE LAB

----------------------------------------------------------------------------------------------------

In this lab, we will cover the following topics:

- Understand MATLAB program for ASK & BFSK and test it for different user inputs
- Implement MFSK, BPSK, QPSK signals

----------------------------------------------------------------------------------------------------

## 10.1 AMPLITUDE SHIFT KEYING (ASK)

In ASK, the two binary values are represented by two different amplitudes of the carrier frequency. Commonly, one of the amplitudes is zero, i.e. one binary digit is represented by the presence, at constant amplitude, of the carrier; while the other is represented by the absence of the carrier. The resulting transmitted signal for one bit time is

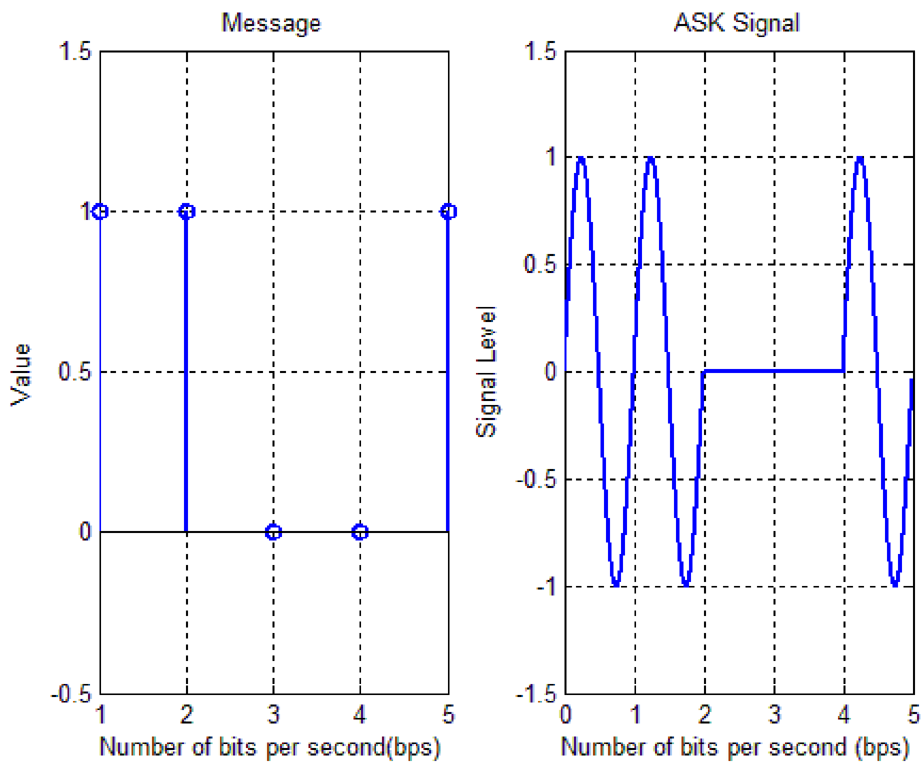
$$\boxed{s(t) = \begin{cases} A\cos(2\pi f_c t) & ,binary\ \ 1 \\ 0 & ,binary\ \ 0 \end{cases}}$$

where carrier signal is $A\cos(2\pi f_c t)$.

## MATLAB Code for Amplitude Shift Keying (ASK):

```
clc
clear all
% taking input message from user
message = input('Enter binary data with spaces eg: [1 1 0 0 1] : ');
f_one = 1;                      % frequency of sinusoid for binary-1
 fs = 1000;                     % sampling frequency
t = [0:1/fs:1];                 % time duration for 1-pulse x =
sin(2*pi*f_one*t);              % sinusoid for binary-1
y = zeros(1,length(t));         % Zero level signal for binary-0
% generating timing vector
tim = t; for(i=1:length(message)-1)
    tim = [tim i+t];
end
% creating ASK signal ask_sig =[];
for i=1:length(message)
    if(message(i) == 0)
                    ask_sig = [ask_sig y];
    else
                ask_sig = [ask_sig x];
   end
 end
% plotting ASK signal w.r.t timing vector figure;
subplot(121);
stem(message, 'linewidth', 2);
 title('Message');
xlabel('Number of bits per second (bps)');
ylabel('Value');
axis([1 length(message) -0.5 1.5]);
grid on; subplot(122);
plot(tim(1,1:(length(message)*fs)),ask_sig(1,1:(length(message)*fs)),'linew idth', 1.5);
title('ASK Signal');
xlabel(' Number of bits per second (bps)');
ylabel('Signal level');
axis([0 length(message) -1.5 1.5]); grid on;
```

Output Signal:




**----------------------------------------TASK 01----------------------------------------**

Write MATLAB code for Binary Phase Shift Keying (BPSK) using the following equation

$$
s(t) = \begin{cases} A\cos(2\pi f_c t) & ,binary \quad 0 \\ A\cos(2\pi f_c t + \pi) & ,binary \quad 1 \end{cases}
$$

Test it for following for different user inputs.

**----------------------------------------TASK 02----------------------------------------**

Write MATLAB code for Quadrature Phase Shift Keying (QPSK) using the equation 5.7 given in Stallings Book. (Pg No. 146)

## 10.2 BINARY FREQUENCY SHIFT KEYING (BFSK)

In BFSK, the two binary values are represented by two different frequencies near the carrier frequency. The resulting transmitted signal for one bit time is

$$\boxed{s(t) = \begin{cases} A\cos(2\pi f_1 t) & ,binary \quad 1 \\ A\cos(2\pi f_2 t) & ,binary \quad 0 \end{cases}}$$

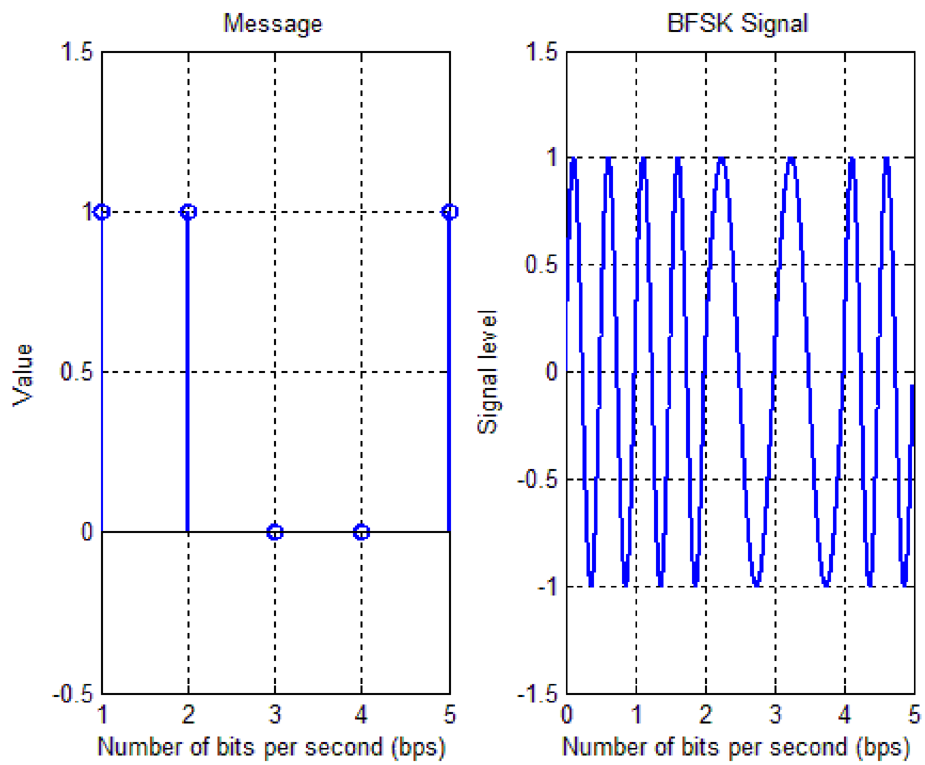
where $f_1$ and $f_2$ are typically offset from the carrier frequency $f_c$ by equal but opposite amounts.

MATLAB Code for Binary Frequency Shift Keying (BFSK):

```
clc
clear all

% taking input message from user
message = input('Enter binary data with spaces eg: [1 1 0 0 1] : ');

f1 = 1;                     % frequency for binary-0
f2 = 2;                     % frequency for binary-1
fs = 1000;                  % sampling frequency
t = [0:1/fs:1];                     % time duration for 1-pulse
x = sin(2*pi*f2*t);                 % sinusoid representing binary-1
y = sin(2*pi*f1*t);                 % sinusoid representing binary-0
   % generating timing vector
   tim = t; for(i=1:length(message)-1)
   tim = [tim i+t];
end

% creating BFSK signal fsk_sig = [];
for i=1:length(message)
   if(message(i) == 0)
      fsk_sig = [fsk_sig y];
```

```
        else
            fsk_sig = [fsk_sig x];
    end
    end

    % plotting BFSK signal w.r.t timing vector
     figure;
    subplot(121);
    stem(message, 'linewidth', 2); title('Message'); xlabel('Number of bits (bps)');
    ylabel('Value');
    axis([1 length(message) -0.5 1.5]);
    grid on; subplot(122);
    plot(tim(1,1:(length(message)*1000)),fsk_sig(1,1:(length(message)*1000))
    , 'linewidth', 1.5);
    title('BFSK Signal');
    xlabel('Number of bits (bps)'); ylabel('Signal
    level');
    axis([0 length(message) -1.5 1.5]);
    grid on;
```

Output Signal:




## ---------------------------------------TASK 03-----------------------------------------

Write MATLAB code that generates Multiple Frequency Shift Keying (MFSK) signal using the following equation

$$s(t) = \{A \cos(2\pi f_i t) \quad ,1 \leq i \leq M$$

where

$f_i = f_c + (2i - 1 - M) f_d$

$f_c$ = carrier frequency

$f_d$ = difference frequency

M = # of different signal elements = $2^L$

L = # of bits per signal element